

**PERAMALAN HARGA MINYAK GORENG DENGAN PENDEKATAN
*K-MEDOIDS TIME SERIES CLUSTERING***

SKRIPSI



**Muhammad Aldani Zen
1907016007**

**PROGRAM STUDI S1 STATISTIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS MULAWARMAN
SAMARINDA
2023**

**PERAMALAN HARGA MINYAK GORENG DENGAN PENDEKATAN
*K-MEDOIDS TIME SERIES CLUSTERING***

SKRIPSI

**Diajukan kepada
Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Mulawarman untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Statistika**

Oleh :

**Muhammad Aldani Zen
1907016007**

**PROGRAM STUDI S1 STATISTIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS MULAWARMAN
SAMARINDA
2023**

HALAMAN PENGESAHAN

Skripsi Sarjana berjudul **Peramalan Harga Minyak Goreng dengan Pendekatan *K-Medoids Time Series Clustering*** oleh **Muhammad Aldani Zen** telah dipertahankan di depan Dewan Penguji pada tanggal 8 Februari 2023.

SUSUNAN TIM PEMBIMBING

Menyetujui,

Pembimbing I,



Dr. Sri Wahyuningsih, M.Si
NIP. 19690413 200012 2 001

Pembimbing II,



Andrea Tri Rian Dani, M.Stat
NIP. 19980116 202203 1 018

Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Mulawarman



Dr. Dra. Hj. Ratna Kusuma, M.Si
NIP. 19630416 198903 2 001

PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa dalam Skripsi yang berjudul “Peramalan Harga Minyak Goreng dengan Pendekatan *K-Medoids Time Series Clustering*” tidak terdapat karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi manapun. Sepanjang pengetahuan saya, tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam Daftar Pustaka.

Demikian pernyataan ini dibuat dengan sebenar-benarnya. Saya sanggup menerima konsekuensi akademik dikemudian hari apabila pernyataan yang dibuat tidak benar.

Samarinda, 8 Februari 2023



Muhammad Aldari Zen

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH**

Saya yang bertanda tangan di bawah ini :

Nama : Muhammad Aldani Zen
NIM : 1907016007
Program Studi/Jurusan : Statistika/Matematika
Judul Skripsi : Peramalan Harga Minyak Goreng dengan
Pendekatan *K-Medoids Time Series Clustering*

Menyatakan bahwa saya menyetujui untuk memberikan hak menyimpan, mengalihmediakan/mengalihformatkan, mengelola dalam bentuk pangkalan data, mempublikasikan untuk kepentingan akademis kepada Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Mulawarman, tanpa perlu meminta izin dari saya selama tetap mencantumkan nama saya sebagai peneliti/pencipta.

Demikian surat pernyataan ini saya buat dengan sesungguhnya dan semoga dapat digunakan sebagaimana mestinya.

Samarinda, 8 Februari 2023

Yang membuat pernyataan



Muhammad Aldani Zen
NIM. 1907016007

ABSTRAK

Data harga minyak goreng seluruh Provinsi di Indonesia merupakan jenis data runtun waktu multivariabel di mana mengarah pada kondisi *big data*. Konsep dari *big data* cenderung memiliki sifat kompleksitas yang tinggi dalam pemrosesan data runtun waktu ketika melakukan proses identifikasi model yang sesuai untuk tiap runtun waktu. *Time series clustering* merupakan salah satu metode yang dapat digunakan dalam pemrosesan data runtun waktu multivariabel. Komponen yang digunakan pada *time series clustering* adalah *K-Medoids* dalam proses pengelompokan dan ARIMA dalam proses pemodelan dan peramalan. Tujuan dari penelitian ini adalah memperoleh nilai *K* optimal berdasarkan koefisien *silhouette*, memperoleh hasil pengelompokan menggunakan algoritma *K-Medoids*, mendapatkan model ARIMA terbaik untuk setiap klaster, dan memperoleh akurasi peramalan untuk setiap Provinsi di Indonesia. Pada penelitian ini, ukuran kemiripan yang digunakan adalah jarak *dynamic time warping*. Berdasarkan hasil analisis, diperoleh nilai *K* optimal sebesar 2 klaster dengan 19 Provinsi bergabung pada klaster 1 dan 15 Provinsi bergabung pada klaster 2. Penentuan objek *prototype* dilakukan dengan menghitung rata-rata data runtun waktu dari suatu klaster, sehingga didapatkan model ARIMA terbaik untuk klaster 1 adalah ARIMA (1,1,0) dan klaster 2 adalah ARIMA (1,1,0). Akurasi peramalan dari data harga minyak goreng setiap Provinsi di Indonesia memberikan rata-rata MAPE pada klaster 1 sebesar 1,47% untuk data *in-sample* dan sebesar 9,42% untuk data *out-sample*. Pada klaster 2 memberikan rata-rata MAPE sebesar 1,60% untuk data *in-sample* dan sebesar 15,20% untuk data *out-sample*.

Kata Kunci: ARIMA, *K-medoids*, minyak goreng, *silhouette coefficient*, *time series clustering*.

ABSTRACT

Cooking oil price data for all provinces in Indonesia is a type of multivariable time series data that leads to big data conditions. The concept of big data tends to have high complexity in processing time series data when identifying the appropriate model for each time series. Time series clustering is a method that can be used in processing multivariable time series data. The components used in time series clustering are K-Medoids in the clustering process and ARIMA in the modeling and forecasting process. The purpose of this study is to obtain optimal K values based on silhouette coefficients, obtain clustering results using the K-Medoids algorithm, obtain the best ARIMA model for each cluster, and obtain forecasting accuracy for each province in Indonesia. In this study, the similarity measure used is the dynamic time-warping distance. Based on the results of the analysis, the optimal K value is obtained for 2 clusters with 19 Provinces joining cluster 1 and 15 Provinces joining cluster 2. Determination of the prototype object is done by calculating the average time series data from a cluster, so that the best ARIMA model is obtained for cluster 1 is ARIMA (1,1,0) and cluster 2 is ARIMA (1,1,0). Forecasting accuracy from cooking oil price data for each province in Indonesia gives an average MAPE in cluster 1 of 1.47% for in-sample data and 9.42% for out-sample data. Cluster 2 gives an average MAPE of 1.60% for in-sample data and 15.20% for out-sample data.

Keywords: ARIMA, cooking oil, K-medoids, silhouette coefficient, time series clustering.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji dan syukur kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Penelitian ini dibuat oleh penulis untuk memenuhi kewajiban penulis sebagai mahasiswa program studi S1 Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Mulawarman. Skripsi ini berjudul “Peramalan Harga Minyak Goreng dengan Pendekatan *K-Medoids Time Series Clustering*” yang penulis kerjakan untuk memenuhi syarat kelulusan.

Dalam menyelesaikan skripsi ini, penulis banyak mendapat bantuan dan dukungan dari berbagai pihak. Untuk itu, penulis mengucapkan banyak terima kasih kepada:

1. Ibu Dr. Sri Wahyuningsih, M.Si., selaku Dosen Pembimbing 1 dan Bapak Andrea Tri Rian Dani, M.Stat., selaku Dosen Pembimbing 2 yang sangat banyak memberikan waktu, tenaga, bantuan, masukan, dan bimbingan serta motivasi kepada penulis dalam menyelesaikan penelitian ini.
2. Ibu Meiliyani Siringoringo, M.Si., selaku Dosen Penguji 1 dan Ibu Ika Purnamasari, M.Si., selaku Dosen Penguji 2 yang telah memberikan arahan, kritikan, dan masukan yang membangun demi kesempurnaan penelitian ini.
3. Bapak Ibu Dosen di Jurusan Matematika yang telah memberikan ilmu pengetahuan selama penulis menempuh pendidikan di program studi Statistika Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.
4. Kedua orang tua tercinta Bapak Subliansyah Zen dan Ibu Lily Hayati yang senantiasa memberikan dukungan dan doa yang tak terhingga kepada penulis dalam mengerjakan penelitian ini.
5. Kedua saudara tersayang Liliana Zen dan Muhammad Denizar Zen yang telah menghibur penulis ketika mengalami kesulitan dalam mengerjakan penelitian ini.

6. Teman-teman dekat Risa Kristia, Fifi Yupita, Robiatul Fauziah, Dini Elizabeth, Isna Alima, Noviani Nur Azizah, M. Reyfi Syahnaz, dan Muhammad Yafi yang selalu menghibur dan memberikan semangat serta menjadi pendengar kepada penulis ketika mengalami kesulitan dan kebingungan dalam mengerjakan penelitian ini.
7. Seluruh teman-teman Statistika 2019 yang tidak dapat penulis sebutkan satu persatu, terima kasih atas segala bantuan, doa, motivasi, dan dukungan dalam proses pengerjaan penelitian ini.

Dalam melakukan penyusunan skripsi ini, penulis menyadari terdapat banyak kekurangan. Untuk itu, penulis mohon maaf yang sebesar-besarnya atas segala kekurangan dan kesalahan yang terdapat dalam skripsi ini, serta penulis mengharapkan adanya kritik dan saran untuk penulis. Akhir kata, penulis ucapkan terima kasih atas perhatiannya.

Wassalamu 'alaikum Warahmatullahi Wabarakatuh

Samarinda, 8 Februari 2023

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
PERNYATAAN KEASLIAN SKRIPSI.....	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xv
DAFTAR SIMBOL	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Batasan Masalah.....	5
1.3 Rumusan Masalah	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	6
BAB 2 TINJAUAN PUSTAKA.....	7
2.1 <i>Data Mining</i>	7
2.2 Analisis Klaster	9
2.3 Standarisasi Data	10
2.4 Ukuran Kemiripan dan Ketidakmiripan Objek	11
2.5 Metode <i>K-Medoids Clustering</i>	13
2.6 Evaluasi <i>Clustering</i>	14
2.7 Peramalan.....	15
2.8 Analisis Runtun Waktu	16
2.9 Stasioneritas Data.....	19

2.9.1 Stasioneritas dalam Variansi	20
2.9.2 Stasioneritas dalam Rata-rata	21
2.10 Fungsi Otokorelasi dan Otokorelasi Parsial	24
2.11 Model Runtun Waktu Box-Jenkins	27
2.12 Estimasi Parameter Model ARIMA	31
2.13 Pengujian Signifikansi Parameter	33
2.14 Pemeriksaan Diagnostik Model	33
2.15 Ukuran Akurasi Peramalan	35
2.16 Harga Minyak Goreng.....	36
BAB 3 METODOLOGI PENELITIAN	38
3.1 Rancangan Penelitian	38
3.2 Waktu dan Tempat Penelitian	38
3.3 Populasi dan Sampel Penelitian	39
3.4 Teknik Sampling	39
3.5 Variabel Penelitian dan Teknik Pengumpulan Data	39
3.6 Teknik Analisis Data.....	39
BAB 4 HASIL DAN PEMBAHASAN	45
4.1 Data Harga Minyak Goreng	45
4.2 Statistika Deskriptif.....	45
4.3 <i>K-Medoids Time Series Clustering</i>	47
4.3.1 Analisis Klaster	47
4.3.2 Pemodelan ARIMA.....	59
4.3.3 Peramalan Harga Minyak Goreng di Indonesia	81
BAB 5 PENUTUP.....	102
5.1 Kesimpulan	102
5.2 Saran.....	103
DAFTAR PUSTAKA	104
LAMPIRAN.....	107
RIWAYAT HIDUP	186

DAFTAR TABEL

	Halaman
Tabel 2.1 Beberapa Nilai λ dengan Transformasi Box-Cox	21
Tabel 2.2 Pola Teoritis FOK dan FOKP dari Proses Data yang Stasioner	29
Tabel 2.3 Interpretasi Nilai MAPE	36
Tabel 4.1 Data Harga Minyak Goreng pada 34 Provinsi di Indonesia	45
Tabel 4.2 Standarisasi Data	49
Tabel 4.3 Informasi <i>Medoids</i> Iterasi ke-1	50
Tabel 4.4 Hasil Perhitungan Jarak DTW Iterasi ke-1	52
Tabel 4.5 Informasi Kandidat <i>Medoids</i> Baru Iterasi ke-2	54
Tabel 4.6 Hasil Perhitungan Jarak DTW Iterasi ke-2	54
Tabel 4.7 Hasil Perhitungan Koefisien <i>Silhouette</i> untuk $K = 2$	57
Tabel 4.8 Hasil Kluster	58
Tabel 4.9 Informasi Objek <i>Prototype</i>	59
Tabel 4.10 Hasil Penaksiran dan Pengujian Signifikansi Parameter Model ARIMA Sementara untuk Klaster 1	68
Tabel 4.11 Hasil Pengujian Normalitas <i>Error</i> Klaster 1	69
Tabel 4.12 Hasil Penaksiran dan Pengujian Signifikansi Parameter Model ARIMA Sementara untuk Klaster 2	78
Tabel 4.13 Hasil Pengujian Normalitas <i>Error</i> Klaster 2	79
Tabel 4.14 Model ARIMA Layak untuk Setiap Klaster	81
Tabel 4.15 Prediksi Data Harga Minyak Goreng Klaster 1 Model ARIMA (0,1,1)	82
Tabel 4.16 Peramalan Data <i>Out-Sample</i> Harga Minyak Goreng Klaster 1 Model ARIMA (0,1,1)	83
Tabel 4.17 Nilai Akurasi Model ARIMA (0,1,1) Klaster 1	84
Tabel 4.18 Peramalan Data Harga Minyak Goreng Klaster 1 untuk Model ARIMA (0,1,1)	84
Tabel 4.19 Prediksi Data Harga Minyak Goreng Klaster 1 Model ARIMA (0,1,2)	86

Tabel 4.20	Peramalan Data <i>Out-Sample</i> Harga Minyak Goreng Klaster 1 Model ARIMA (0,1,2).....	87
Tabel 4.21	Nilai Akurasi Model ARIMA (0,1,2) Klaster 1	87
Tabel 4.22	Peramalan Data Harga Minyak Goreng Klaster 1 untuk Model ARIMA (0,1,2).....	88
Tabel 4.23	Prediksi Data Harga Minyak Goreng Klaster 1 Model ARIMA (1,1,0)	89
Tabel 4.24	Peramalan Data <i>Out-Sample</i> Harga Minyak Goreng Klaster 1 Model ARIMA (1,1,0).....	90
Tabel 4.25	Nilai Akurasi Model ARIMA (1,1,0) Klaster 1	90
Tabel 4.26	Peramalan Data Harga Minyak Goreng Klaster 1 untuk Model ARIMA (1,1,0).....	91
Tabel 4.27	Rekapitulasi Nilai Akurasi Peramalan Seluruh Model Klaster 1.....	92
Tabel 4.28	Nilai Akurasi Peramalan Tingkat Individu (Provinsi) Klaster 1.....	93
Tabel 4.29	Prediksi Data Harga Minyak Goreng Klaster 2 Model ARIMA (0,1,1)	94
Tabel 4.30	Peramalan Data <i>Out-Sample</i> Harga Minyak Goreng Klaster 2 Model ARIMA (0,1,1).....	95
Tabel 4.31	Nilai Akurasi Model ARIMA (0,1,1) Klaster 2	95
Tabel 4.32	Peramalan Data Harga Minyak Goreng Klaster 2 untuk Model ARIMA (0,1,1).....	96
Tabel 4.33	Prediksi Data Harga Minyak Goreng Klaster 2 Model ARIMA (1,1,0)	97
Tabel 4.34	Peramalan Data <i>Out-Sample</i> Harga Minyak Goreng Klaster 2 Model ARIMA (1,1,0).....	98
Tabel 4.35	Nilai Akurasi Model ARIMA (1,1,0) Klaster 2	98
Tabel 4.36	Peramalan Data Harga Minyak Goreng Klaster 2 untuk Model ARIMA (1,1,0).....	99
Tabel 4.37	Rekapitulasi Nilai Akurasi Peramalan Seluruh Model Klaster 2...	100
Tabel 4.38	Nilai Akurasi Peramalan Tingkat Individu (Provinsi) Klaster 2....	101

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Jarak <i>euclidean</i> (kiri) dan jarak DTW (kanan)	12
Gambar 2.2 Pola data horizontal.....	18
Gambar 2.3 Pola data <i>trend</i>	18
Gambar 2.4 Pola data siklis	19
Gambar 2.5 Pola data musiman	19
Gambar 3.1 Rancangan penelitian	38
Gambar 3.2 Tahapan analisis data	43
Gambar 4.1 (a) <i>Spatial mapping</i> Oktober 2017; (b) <i>Spatial mapping</i> Oktober 2022.....	46
Gambar 4.2 <i>Barcode</i> untuk mengakses <i>spatial mapping</i> bergerak.....	46
Gambar 4.3 Grafik nilai koefisien <i>silhouette</i>	57
Gambar 4.4 <i>Spatial mapping</i> hasil kluster	58
Gambar 4.5 (a) <i>Multiple plot</i> runtun waktu kluster 1; (b) <i>Multiplot</i> runtun waktu kluster 2	61
Gambar 4.6 Grafik runtun waktu $Y(1)_t$ bulan Oktober 2017 – Januari 2022....	62
Gambar 4.7 Grafik runtun waktu setelah transformasi $(Y(1)_t)^*$	64
Gambar 4.8 (a) Grafik FOK $Y(1)_t^*$; (b) Grafik FOKP $Y(1)_t^*$	65
Gambar 4.9 (a) Grafik FOK $Y(1)_t^*$ setelah <i>differencing</i> ; (b) Grafik FOKP $Y(1)_t^*$ setelah <i>differencing</i>	66
Gambar 4.10 Grafik Ljung-Box model ARIMA (0,1,1) untuk kluster 1	70
Gambar 4.11 Grafik Ljung-Box model ARIMA (0,1,2) untuk kluster 1	71
Gambar 4.12 Grafik Ljung-Box model ARIMA (1,1,0) untuk kluster 1	71
Gambar 4.13 Grafik runtun waktu $Y(2)_t$ bulan Oktober 2017 – Januari 2022 ..	72
Gambar 4.14 Grafik runtun waktu setelah transformasi $(Y(2)_t)^*$	74
Gambar 4.15 (a) Grafik FOK $Y(2)_t^*$; (b) Grafik FOKP $Y(2)_t^*$	75

Gambar 4.16 (a) Grafik FOK $Y(2)_t^*$ setelah *differencing* ; (b) Grafik FOKP

$Y(2)_t^*$ setelah *differencing* 76

Gambar 4.17 Grafik Ljung-Box model ARIMA (0,1,1) untuk klaster 2 80

Gambar 4.18 Grafik Ljung-Box model ARIMA (1,1,0) untuk klaster 2 81

Gambar 4.19 Data aktual vs prediksi klaster 1 untuk model ARIMA (0,1,1) 85

Gambar 4.20 Data aktual vs prediksi klaster 1 untuk model ARIMA (0,1,2) 88

Gambar 4.21 Data aktual vs prediksi klaster 1 untuk model ARIMA (1,1,0) 91

Gambar 4.22 Data aktual vs prediksi klaster 2 untuk model ARIMA (0,1,1) 96

Gambar 4.23 Data aktual vs prediksi klaster 2 untuk model ARIMA (1,1,0) 99

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Data Harga Minyak Goreng pada 34 Provinsi di Indonesia	108
Lampiran 2. <i>Command R</i> untuk Eksplorasi Data	116
Lampiran 3. Tabel Hasil Standarisasi Data	117
Lampiran 4. <i>Command R</i> Contoh Perhitungan Jarak <i>Dynamic Time Warping</i>	124
Lampiran 5. <i>Command R</i> untuk Jarak <i>Dynamic Time Warping</i> Keseluruhan	133
Lampiran 6. <i>Command R</i> Algoritma <i>K-Medoids Clustering</i> dengan $K = 2$...	136
Lampiran 7. <i>Command R</i> Algoritma <i>K-Medoids Clustering</i> dengan $K = 3$...	137
Lampiran 8. <i>Command R</i> Algoritma <i>K-Medoids Clustering</i> dengan $K = 4$...	138
Lampiran 9. <i>Command R</i> Algoritma <i>K-Medoids Clustering</i> dengan $K = 5$...	139
Lampiran 10. <i>Command R</i> Algoritma <i>K-Medoids Clustering</i> dengan $K = 6$..	140
Lampiran 11. <i>Command R</i> Algoritma <i>K-Medoids Clustering</i> dengan $K = 7$..	141
Lampiran 12. <i>Command R</i> Algoritma <i>K-Medoids Clustering</i> dengan $K = 8$..	142
Lampiran 13. <i>Command R</i> Algoritma <i>K-Medoids Clustering</i> dengan $K = 9$..	143
Lampiran 14. <i>Command R</i> Algoritma <i>K-Medoids Clustering</i> dengan $K = 10$	144
Lampiran 15. <i>Command R</i> untuk Koefisien <i>Silhouette</i>	145
Lampiran 16. <i>Command R</i> untuk Hasil Klaster	146
Lampiran 17. Visualisasi <i>Spatial Mapping</i> untuk Hasil Klaster	147
Lampiran 18. Data Objek <i>Prototype</i> Setiap Klaster	148
Lampiran 19. <i>Command R</i> untuk Visualisasi Objek <i>Prototype</i> Klaster 1	149
Lampiran 20. <i>Command R</i> untuk Visualisasi Objek <i>Prototype</i> Klaster 2	150
Lampiran 21. <i>Command R</i> untuk Identifikasi Awal Model ARIMA Klaster 1	151
Lampiran 22. <i>Command R</i> untuk Identifikasi Orde dan Penaksiran Parameter Model ARIMA Klaster 1	153
Lampiran 23. <i>Command R</i> untuk <i>Diagnostic Checking</i> Model ARIMA Klaster 1	154
Lampiran 24. <i>P-value</i> Pengujian Ljung-Box untuk Semua Model Klaster 1 ...	155

Lampiran 25. <i>Command R</i> untuk Identifikasi Awal Model ARIMA Klaster 2	156
Lampiran 26. <i>Command R</i> untuk Identifikasi Orde dan Penaksiran Parameter Model ARIMA Klaster 2	158
Lampiran 27. <i>Command R</i> untuk <i>Diagnostic Checking</i> Model ARIMA Klaster 2	159
Lampiran 28. <i>P-value</i> Pengujian Ljung-Box untuk Semua Model Klaster 2	160
Lampiran 29. <i>Command R</i> untuk Pembangkitan Nilai <i>Error</i>	161
Lampiran 30. Data Aktual dan Prediksi untuk Objek <i>Prototype</i> Klaster 1 Model ARIMA (0,1,1)	162
Lampiran 31. Data Aktual dan Prediksi untuk Objek <i>Prototype</i> Klaster 1 Model ARIMA (0,1,2)	163
Lampiran 32. Data Aktual dan Prediksi untuk Objek <i>Prototype</i> Klaster 1 Model ARIMA (1,1,0)	164
Lampiran 33. Data Aktual dan Prediksi untuk Objek <i>Prototype</i> Klaster 2 Model ARIMA (0,1,1)	165
Lampiran 34. Data Aktual dan Prediksi untuk Objek <i>Prototype</i> Klaster 2 model ARIMA (1,1,0)	166
Lampiran 35. Plot Aktual dan Prediksi Setiap Provinsi di Klaster 1	167
Lampiran 36. Plot Aktual dan Prediksi Setiap Provinsi di Klaster 2	170
Lampiran 37. <i>Command R</i> untuk Nilai MAPE Objek <i>Prototype</i> Klaster 1	172
Lampiran 38. <i>Command R</i> untuk Nilai MAPE Objek <i>Prototype</i> Klaster 2	174
Lampiran 39. <i>Command R</i> untuk Nilai MAPE Setiap Provinsi di Klaster 1	175
Lampiran 40. <i>Command R</i> untuk Nilai MAPE Setiap Provinsi di Klaster 2	179
Lampiran 41. Tabel Harga Kritis D dalam Tes Satu Sampel Kolmogorov-Smirnov	183
Lampiran 42. Tabel Harga Kritis <i>t</i>	184
Lampiran 43. Tabel Harga Kritis <i>Chi-Square</i> (χ^2)	185

DAFTAR SIMBOL

Simbol	Arti
Z_t	Data pada waktu ke- t
n	Banyaknya data
$\bar{Z}$	Rata-rata dari Z_t
S_Z	Deviasi standar dari Z_t
$\tilde{Z}_t$	Standarisasi data dari Z_t
$Z(1)_t$	Data observasi pertama pada waktu ke- t dengan $Z(1)_t = Z_1, Z_2, Z_3, \dots, Z_i$ di mana $i = 1, 2, 3, \dots, n$
$Z(2)_t$	Data observasi kedua pada waktu ke- t dengan $Z(2)_t = Z_1, Z_2, Z_3, \dots, Z_j$ di mana $j = 1, 2, 3, \dots, m$
$\mathbf{C}$	Matriks jarak <i>dynamic time warping</i>
c_{ij}	Elemen matriks $\mathbf{C}$ pada baris ke- i dan kolom ke- j
w_{ij}	Nilai mutlak dari selisih antara Z_i dengan Z_j
P	Sekumpulan dari <i>warping path</i>
s	Panjang <i>warping path</i>
$d_{DTW}(Z(1)_t, Z(2)_t)$	Jarak <i>dynamic time warping</i> data $Z(1)_t$ dengan data $Z(2)_t$
K	Banyak klaster dalam metode <i>K-Medoids</i>
C_i	Klaster ke- i
E	Jumlah kesalahan absolut
N	Jumlah objek yang bergabung dalam klaster ke- i
S	Selisih fungsi objektif lama dengan fungsi objektif yang baru
a_i	Rata-rata jarak objek ke- i dengan semua data pada satu klaster yang sama
b_i	Rata-rata jarak objek ke- i dengan semua data pada satu klaster yang berbeda

$SC(i)$	Koefisien <i>silhouette</i>
$Y(1)_t$	Data objek <i>prototype</i> pertama pada waktu ke- t
$Y(2)_t$	Data objek <i>prototype</i> kedua pada waktu ke- t
Z_t^*	Transformasi data dari Z_t
d	Orde <i>differencing</i> (pembedaan)
B	Operator <i>backshift</i>
∇^d	<i>Differencing</i> orde ke- d
δ	Koefisien regresi untuk <i>lag</i>
γ	Koefisien regresi untuk <i>lag</i> pembedaan
e_t	<i>Error</i> pada waktu ke- t
τ	Statistik uji ADF
$\hat{\delta}$	Nilai estimasi dari δ
$SE(\hat{\delta})$	<i>Standard error</i> dari $\hat{\delta}$
α	Taraf signifikansi
n_p	Banyaknya parameter
μ	Rata-rata
ρ_k	Koefisien otokorelasi pada <i>lag-k</i>
γ_k	Koefisien otokovariansi pada <i>lag-k</i>
k	Selisih waktu (<i>lag</i>)
ϕ_{kk}	Koefisien otokorelasi parsial pada <i>lag-k</i>
p	Orde untuk <i>autoregressive</i> (AR)
q	Orde untuk <i>moving average</i> (MA)
ϕ_p	Parameter <i>autoregressive</i> orde ke- p
θ_q	Parameter <i>moving average</i> orde ke- q
L	Fungsi <i>likelihood</i>
l	Fungsi <i>log-likelihood</i>
π	Phi = 3,141592654
$\hat{\sigma}_e^2$	Estimasi <i>maximum likelihood</i> variansi <i>error</i>
t	Statistik uji t
ω	Pemisalan suatu parameter model ARIMA (ϕ, θ)

$\hat{\omega}$	Nilai estimasi dari ω
$SE(\hat{\omega})$	<i>Standard error</i> dari nilai estimasi ω
$F_0(Z)$	Fungsi distribusi kumulatif
$F_n(Z)$	Fungsi distribusi empiris
D	Statistik uji Kolmogorov-Smirnov
Q^*	Statistik uji Ljung-Box
$\hat{p}_{e_k}$	Estimator fungsi otokorelasi data <i>error</i> pada <i>lag</i> k
k^*	<i>Lag</i> maksimum
χ^2	Distribusi <i>chi-square</i>
$\hat{Z}_t$	Prediksi dan peramalan pada periode ke- t

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Peramalan (*forecasting*) merupakan serangkaian proses dalam analisis statistika untuk memprediksi suatu kejadian di masa yang akan datang. Peramalan dilakukan dengan mencermati data masa lalu sehingga memperoleh pola tertentu yang dapat diestimasi pada masa mendatang. Analisis runtun waktu merupakan salah satu metode peramalan yang dapat diimplementasikan untuk meramalkan serangkaian data runtun waktu (Aswi & Sukarna, 2006). Menurut Sisimi (2018), model peramalan runtun waktu yang paling populer adalah model *Autoregressive Integrated Moving Average* (ARIMA) oleh George Box dan Gwilym Jenkins yang mengembangkan model tersebut pada tahun 1970. Konsep dari model ARIMA yaitu menentukan hubungan statistik yang baik antar peubah yang diramalkan dengan nilai masa lampau peubah tersebut sehingga model tersebut dapat dilakukan untuk peramalan dengan hasil ramalan yang akurat.

Seiring berkembangnya kemajuan teknologi yang semakin canggih, masyarakat dituntut untuk bertransformasi menjadi serba digital. Teknologi informasi merupakan salah satu perkembangan teknologi yang cukup pesat. Informasi digital yang diterima baik terstruktur maupun tidak terstruktur akan secara terus menerus tersimpan dan terkumpul dengan jumlah yang sangat besar. Penggunaan pembelajaran *data mining* banyak digunakan dalam membantu mengelola data dalam jumlah yang cukup besar.

Data mining merupakan salah satu metode untuk menemukan wawasan yang menarik dari kumpulan basis data dengan ukuran yang sangat besar. Proses dari *data mining* yaitu mengekstraksi dan mengidentifikasi wawasan tersembunyi yang tersimpan dalam basis data besar (*big data*) sehingga memiliki wawasan baru yang dapat dipahami dengan mudah (Turban, 2005). Menurut Yudistira (2021), konsep dari *big data* cenderung memiliki sifat kompleksitas yang tinggi karena struktur dari data besar beragam dan menjadi tantangan dalam pemrosesan data. Masalah akan menjadi lebih serius, ketika *big data* kemudian dikaitkan

dengan data runtun waktu, dimana terjadi kompleksitas komputasi yang tinggi dalam mengidentifikasi model peramalan runtun waktu, salah satunya adalah model ARIMA. Proses identifikasi model dengan menentukan orde ARIMA yang sesuai dari kondisi *big data* akan menyulitkan peneliti terhadap efisiensi waktu, tenaga, biaya serta kompleksitas dari bahasa pemrograman yang dibangun karena perlu mengidentifikasi model runtun waktu secara individual tiap runtun waktu. Pemilihan suatu algoritma yang sesuai menjadi sangat penting dalam menangani permasalahan *big data* dalam pemrosesan data runtun waktu salah satunya dengan menggunakan analisis klaster.

Konsep dari analisis klaster adalah mempartisi sekumpulan objek pengamatan ke dalam suatu himpunan. Himpunan-himpunan yang terbentuk disebut dengan klaster. Objek-objek di dalam klaster tersebut adalah objek yang memiliki kemiripan satu sama lain namun berbeda dengan objek yang berada pada klaster lainnya. Pengelompokan menjadi hal yang berguna karena dapat mengarah pada penemuan kelompok yang sebelumnya tidak diketahui di dalam data (Han dkk., 2012). Menurut Prasetyo (2012), terdapat dua jenis analisis klaster berdasarkan strukturnya yaitu *hierarchical clustering* dan *non-hierarchical clustering*. Perbedaan dari kedua teknik klaster ini terletak pada proses penentuan jumlah kelompok. Pada penelitian ini hanya terfokus pada *non-hierarchical clustering*. Metode *non-hierarchical clustering* dimulai dengan menentukan jumlah klaster yang akan dipartisi, sehingga dalam pembagian *set* data ke dalam sejumlah kelompok tidak tumpang tindih antara satu objek individu dengan objek individu yang lainnya. Salah satu contoh dari metode ini adalah *K-Means*.

Algoritma pengelompokan *K-Means* pada *non-hierarchical clustering* memiliki kelemahan pada data *outlier*. Menurut Han dan Kamber (2006), *outlier* merupakan objek data yang menyimpang secara signifikan dari objek lainnya. Suatu objek dengan nilai yang berbeda dari distribusi data secara substansial akan menunjukkan penyimpangan dari distribusi data. Seiring berkembangnya teknik komputasi dari berbagai penelitian yang dilakukan, teknik *Partitioning Around Method* (PAM) atau *K-Medoids* muncul untuk mengatasi kelemahan dari *K-Means* yang sensitif terhadap *outlier*. Konsep dari teknik *K-Medoids* adalah

menggunakan salah satu objek sebagai *medoids* atau pusat dalam mewakili sebuah kelompok kemudian menempatkan objek yang menjadi *non-medoids* ke dalam kelompok yang paling dekat dengan *medoids* tersebut. Analisis kluster terus berkembang dengan penerapannya yang tidak hanya terbatas pada satu tipe data.

Tipe data yang banyak digunakan dalam analisis kluster adalah data *cross-section*, namun pada kondisi tertentu terkadang dituntut dalam memproses data dengan tipe data yang lain salah satunya adalah tipe data runtun waktu. Menurut Dani dkk. (2019), algoritma pengelompokan tidak hanya spesifik pada tipe data *cross-section*, data runtun waktu dapat juga diimplementasikan pada analisis kluster dengan algoritma yang sedikit berbeda dengan data *cross-section*. *Time series clustering* adalah teknik pendekatan yang dapat digunakan dalam pemrosesan data runtun waktu multivariabel. Konsep dari *time series clustering* adalah mengelompokkan berdasarkan kemiripan dari pola masing-masing runtun waktu, sehingga objek runtun waktu yang memiliki kemiripan pola yang sama akan cenderung membentuk suatu himpunan. Teknik *time series clustering* semakin berkembang dengan digunakannya jarak-jarak pengukuran kemiripan yang sesuai dengan karakteristik data runtun waktu. Jarak yang digunakan dalam mengukur kemiripan dua data runtun waktu pada penelitian ini adalah jarak *dynamic time warping* di mana memiliki kelebihan yaitu dapat mengukur dua runtun waktu dengan panjang runtun waktu yang berbeda.

Beberapa penelitian yang telah dilakukan sebelumnya menggunakan metode *time series clustering*, yaitu penelitian yang dilakukan oleh Wijaya dan Ngatini (2020) dengan melakukan pemodelan harga beras menggunakan ARIMA untuk level individu dan level kelompok. Hasil dari penelitian tersebut menunjukkan pemodelan level kelompok lebih efisien dibandingkan pemodelan level individu dengan akurasi rata-rata MAPE sebesar 3,52% untuk level individu dan MAPE pada klaster 1 dan klaster 2 berturut-turut sebesar 1,97% dan 2,12% untuk level kelompok. Penelitian lain dilakukan oleh Novidianto dan Dani (2020) dengan mengelompokkan data runtun waktu kasus aktif COVID-19 menurut Provinsi di Indonesia. Hasil penelitian menunjukkan terbentuknya 3 klaster di mana Provinsi Jakarta dan Jawa Timur merupakan klaster dengan kasus aktif

COVID-19 yang tinggi. Selanjutnya penelitian yang dilakukan oleh Astuti dkk. (2022) tentang pengelompokan dan peramalan kasus COVID-19 menggunakan model ARIMA di Indonesia. Hasil penelitian menunjukkan terbentuknya 2 klaster dengan hasil peramalan masing-masing klaster cenderung konstan.

Teknik *time series clustering* dapat diimplementasikan untuk menganalisa masalah-masalah dalam bidang sektor ekonomi sebagai perancangan kebijakan pemerintah dalam mengatasi masalah-masalah yang terjadi. Ketidakstabilan harga bahan pokok merupakan salah satu masalah yang dihadapi dalam sektor ekonomi khususnya pada minyak goreng.

Minyak goreng merupakan salah satu komoditas pokok pangan yang strategis di Indonesia. Tingginya konsumsi minyak goreng akan menyebabkan peningkatan permintaan bahan pokok tersebut setiap tahunnya. Menurut Irnawati (2021), ketika permintaan (*demand*) suatu barang meningkat namun tidak diikuti dengan *supply* yang meningkat maka akan menyebabkan ketidakseimbangan dan akan berdampak pada kenaikan harga barang. Berdasarkan data yang disajikan oleh Pusat Informasi Harga Pangan Strategis Nasional (PIHPS) per 26 Agustus 2022, harga minyak goreng untuk pasar tradisional masih tergolong tinggi hingga mencapai Rp 20.200 per liter. Puncak harga minyak goreng tertinggi pernah mencapai hingga Rp 24.650 per liter pada tanggal 22 April 2022. Terbatasnya persediaan pasar nasional dan meningkatnya permintaan dari pasar internasional menyebabkan harga minyak goreng melambung tinggi. Sedangkan harga minyak goreng sebelumnya telah diatur dalam Peraturan Menteri Perdagangan Republik Indonesia No. 22 yang menjelaskan bahwa Harga Eceran Tertinggi (HET) dari minyak goreng berada pada rentang Rp 11.000 hingga Rp 13.000 per liter. Pengendalian harga minyak goreng berada pada *range* HET yang telah diatur sebelumnya menjadi hal yang sangat penting dalam menjaga stabilitas ekonomi di Indonesia.

Berdasarkan uraian latar belakang tersebut dan penelitian-penelitian terkait sebelumnya, peneliti tertarik untuk melakukan penelitian menggunakan peramalan analisis runtun waktu dengan pendekatan *time series clustering* yang berjudul

“Peramalan Harga Minyak Goreng dengan Pendekatan *K-Medoids Time Series Clustering*”.

1.2 Batasan Masalah

Batasan masalah yang digunakan agar penelitian ini dapat berjalan secara terarah serta dapat mencapai tujuan yang diinginkan adalah sebagai berikut:

1. Nilai parameter K pada *K-Medoids Clustering Time Series* yang digunakan adalah $K = 2, 3, 4, \dots, 10$.
2. Pada proses peramalan data dibagi ke dalam proporsi 85% data *in-sample* dan 15% data *out-sample*.
3. Model peramalan yang digunakan adalah ARIMA.

1.3 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Berapa nilai K optimal berdasarkan koefisien *silhouette* dalam proses pengelompokan menggunakan algoritma *K-Medoids*?
2. Bagaimana hasil pengelompokan dengan menggunakan algoritma *K-Medoids*?
3. Apa model ARIMA terbaik untuk setiap klaster yang terbentuk dari data harga minyak goreng di Indonesia?
4. Bagaimana akurasi peramalan dari data harga minyak goreng setiap Provinsi di Indonesia dengan menggunakan model ARIMA terbaik setiap klaster?

1.4 Tujuan Penelitian

Sesuai dengan permasalahan yang telah dipaparkan sebelumnya, maka tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Untuk memperoleh nilai K optimal berdasarkan koefisien *silhouette* dalam proses pengelompokan menggunakan algoritma *K-Medoids*.

2. Untuk memperoleh hasil pengelompokan dengan menggunakan algoritma *K-Medoids*.
3. Untuk mendapatkan model ARIMA terbaik untuk setiap klaster yang terbentuk dari data harga minyak goreng di Indonesia.
4. Untuk memperoleh akurasi peramalan dari data harga minyak goreng setiap Provinsi di Indonesia dengan menggunakan model ARIMA terbaik setiap klaster.

1.5 Manfaat Penelitian

Manfaat yang diharapkan peneliti dari pemodelan harga minyak goreng dengan pendekatan *K-Medoids Time Series Clustering* dalam penelitian ini adalah:

1. Bagi mahasiswa dapat menambah pengetahuan dan menjadi bahan referensi untuk penelitian selanjutnya yang berkaitan dengan analisis runtun waktu dan *data mining*.
2. Bagi masyarakat dapat menjadi sumber informasi terkait fenomena minyak goreng.
3. Bagi pihak pemerintah diharapkan dapat menjadi pertimbangan dalam perancangan kebijakan ekonomi pasar khususnya minyak goreng.

BAB 2

TINJAUAN PUSTAKA

2.1 *Data Mining*

Data mining adalah suatu proses dalam menemukan pola yang menarik dan pengetahuan dari kumpulan data yang besar. Kumpulan data tersebut dapat berupa basis data, data *warehouse* atau tempat penyimpanan data (Han dkk., 2012). *Data mining* merupakan informasi baru yang diekstraksi dari sejumlah data besar, digunakan dalam pengambilan suatu keputusan. Tujuan utama dari proses *data mining* adalah mengekstrak atau mengambil intisari pengetahuan dari sekumpulan data sehingga diperoleh struktur data yang dapat dipahami manusia serta meliputi basis data dan manajemen data, pra pemrosesan data, pertimbangan kompleksitas, pasca pemrosesan terhadap struktur yang ditemukan, visualisasi, dan *online updating* (Suyanto, 2019). Proses dalam penemuan pengetahuan dari basis data yang besar terdiri dari lima langkah yaitu (Han dkk., 2012):

1. *Data cleaning and integration*, pembersihan data dilakukan bertujuan untuk menghilangkan *noise* data serta data yang tidak konsisten. Selanjutnya integrasi data merupakan proses dalam menggabungkan sejumlah data dari berbagai macam sumber menjadi satu kesatuan yang tersimpan.
2. *Data selection and transformation*, data yang relevan dengan analisis diambil dari basis data. Kemudian melakukan transformasi data di mana data diubah ke dalam bentuk yang sesuai untuk penambangan.
3. *Data mining*, proses yang sangat penting yaitu menerapkan metode yang cerdas untuk mengekstrak pola data dari kumpulan basis data yang besar.
4. *Data evaluation and presentation*, setelah menemukan pola data dari proses ekstraksi data, maka hasil tersebut perlu divisualisasikan dan merepresentasikan pengetahuan tersebut dalam bentuk yang sederhana dan mudah mengerti kepada pengguna.

Menurut Prasetyo (2012), terdapat tiga langkah utama dalam *data mining* yaitu eksplorasi data, membangun model serta melakukan validasi, dan

penerapannya. Berdasarkan tugas yang dapat dilakukan *data mining* dibagi menjadi beberapa kelompok, yaitu:

1. Model prediktif

Model prediktif erat kaitannya dengan pembuatan sebuah model yang dapat melakukan pemetaan dari setiap himpunan variabel ke setiap targetnya, kemudian menggunakan model tersebut untuk memberikan nilai target pada himpunan yang baru yang didapat.

2. Analisis klaster

Analisis klaster atau kelompok digunakan untuk pengelompokan (*clustering*) identifikasi data yang memiliki karakteristik tertentu.

3. Analisis asosiasi

Analisis asosiasi sering disebut dengan analisis keranjang pasar, analisis ini bertujuan untuk mengidentifikasi objek produk yang kemungkinan dibeli konsumen bersamaan dengan produk lain.

4. Deteksi anomali

Deteksi anomali erat kaitannya dengan data *outlier* di mana pengamatan sebuah data dari sejumlah data yang secara signifikan mempunyai karakteristik yang berbeda dari sisa data yang lain.

Suyanto (2019) menjelaskan kegunaan *data mining* yang dibagi menjadi dua yaitu deskriptif dan prediktif. Secara deskriptif *data mining* digunakan untuk menemukan pola data dari sekumpulan basis data dan dapat dimengerti manusia yang menjelaskan karakteristik data. Secara prediktif *data mining* digunakan untuk membangun sebuah model pengetahuan yang akan digunakan untuk proses prediksi. Menurut Santoso dkk. (2021), terdapat tiga metode utama pembelajaran dalam teknik *data mining* yaitu:

1. *Supervised learning*, teknik *data mining* yang prosesnya memerlukan pelatihan untuk mengenali pola data yang telah tersedia. Konsep *supervised learning* melatih algoritma untuk mengenali pola dari data set yang nantinya akan diterapkan sebagai acuan dalam memprediksi data baru.
2. *Unsupervised learning*, teknik ini kebalikan dari *supervised learning*, di mana dalam prosesnya tidak memerlukan pelatihan. *Unsupervised learning* adalah

teknik *data mining* yang bersifat prediktif di mana akan berguna untuk mengelompokan atau mengkategorikan data.

3. *Reinforcement learning*, penggunaan teknik *data mining* yang melakukan pelatihan algoritma dengan sistem *trial and error*. Konsep dari teknik ini adalah berinteraksi dengan lingkungannya dan mengamati konsekuensi atas tindakannya sebagai tanggapan atas penghargaan maupun hukuman yang diterima. Informasi yang dihasilkan dari setiap interaksi dengan lingkungan, diterapkan *reinforcement learning* untuk memperbarui pengetahuannya (Andreanus & Kurniawan, 2017).

2.2 Analisis Klaster

Analisis klaster adalah salah satu teknik dalam *data mining* yang diimplementasikan untuk proses penggerombolan himpunan objek ke dalam suatu ke dalam suatu kelompok hingga objek-objek yang memiliki persamaan karakteristik tertinggi akan bergabung di suatu kelompok tersebut, namun memiliki ketidakmiripan yang tinggi dengan objek-objek yang bergabung di kelompok lainnya (Suyanto, 2019).

Menurut Prasetyo (2012), tujuan digunakannya analisis klaster adalah agar objek-objek yang bergabung dalam sebuah kelompok merupakan objek-objek yang memiliki kemiripan atau berhubungan satu sama lain dan berbeda atau tidak berhubungan dengan objek di kelompok yang lain. Santosa (2007) menjelaskan kelompok yang baik adalah kelompok yang memiliki ciri-ciri sebagai berikut:

1. Homogenitas (kesamaan) yang tinggi antar anggota kelompok (*within cluster*).
2. Heterogenitas (perbedaan) yang tinggi antar kelompok yang satu dengan kelompok yang lainnya (*between cluster*).

Analisis klaster berdasarkan strukturnya terdapat dua metode pengelompokan, yaitu *hierarchical clustering* (klaster hirarki) dan *non-hierarchical clustering* (klaster non-hirarki). Pengelompokan hirarki pada dasarnya bekerja dengan menggabungkan objek individu dengan objek atau kelompok yang lainnya secara berurutan seperti pohon menggunakan dendogram. Satu data tunggal dapat membentuk kelompok sendiri, dua atau lebih kelompok

kecil dapat bergabung menjadi sebuah kelompok besar secara terus menerus di mana ada tingkatan yang jelas antar objek dari yang memiliki kemiripan tertinggi hingga tidak memiliki kemiripan. Metode yang terdapat dalam klaster hirarki adalah *single linkage*, *complete linkage*, dan *average linkage*. Metode yang kedua adalah klaster non-hirarki, metode ini disebut juga metode *partitioning* di mana dalam prosesnya membagi set data ke dalam sejumlah kelompok yang tidak tumpang tindih antara satu kelompok dengan kelompok yang lain. Setiap objek hanya menjadi anggota satu kelompok. Metode yang termasuk dalam klaster non-hirarki adalah *K-means* dan *K-medoids clustering*. Perbedaan yang jelas dari kedua metode adalah proses penentuan jumlah klaster. Pada klaster non-hirarki dalam penentuan jumlah klaster ditentukan terlebih dahulu sesuai yang diinginkan, sedangkan pada klaster hirarki dilakukan diakhir dalam penentuan jumlah kelompok menyesuaikan pada tingkat yang sesuai untuk mengelompokkan objek menjadi beberapa kelompok (Prasetyo, 2012).

2.3 Standarisasi Data

Teknik standarisasi sangat penting dalam proses analisis klaster. Nilai-nilai atribut data yang berbeda-beda rentangnya seringkali perlu dilakukan standarisasi agar proses pengelompokan tidak bias. Standarisasi akan membuat atribut memiliki bobot yang sama sehingga memperkecil perbedaan antar kelompok. Salah satu teknik standarisasi yang dapat digunakan adalah standarisasi *Z-Score* di mana metode tersebut sering disebut pula dengan *zero-mean* karena dalam prosesnya atribut ditransformasi ke dalam data standar berdasarkan nilai rata-rata sama dengan nol serta deviasi standar sama dengan satu (Suyanto, 2019).

Misalkan terdapat data runtun waktu $Z_t = Z_1, Z_2, Z_3 \dots, Z_n$ akan dilakukan proses standarisasi dengan menghitung nilai rata-rata dan deviasi standar. Formulasi dari rata-rata runtun waktu dituliskan pada Persamaan (2.1).

$$\bar{Z} = \left(\frac{1}{n}\right) \sum_{t=1}^n Z_t \quad (2.1)$$

sedangkan untuk deviasi standarnya dituliskan pada Persamaan (2.2)

$$S_Z = \sqrt{\frac{1}{n-1} \sum_{t=1}^n (Z_t - \bar{Z})^2} \quad (2.2)$$

standarisasi data runtun waktu Z_t dituliskan pada Persamaan (2.3).

$$\tilde{Z}_t = \frac{Z_t - \bar{Z}}{S_Z} \quad (2.3)$$

Dengan:

Z_t : data runtun waktu Z untuk periode waktu ke- t

n : banyaknya periode waktu

$\bar{Z}$: rata-rata dari Z_t

S_Z : deviasi standar dari Z_t

$\tilde{Z}_t$: standarisasi data dari Z_t

2.4 Ukuran Kemiripan dan Ketidakmiripan Objek

Pada umumnya proses dalam membentuk klaster didasarkan pada kedekatan antar objek yang satu dengan objek lainnya. Ukuran kemiripan (*similarity*) atau ketidakmiripan (*dissimilarity*) merupakan tahap awal pengelompokan di mana sebuah ukuran untuk kesesuaian diantara objek-objek yang akan dibentuk menjadi sebuah kelompok. Objek-objek yang berada pada satu kelompok akan memiliki jarak yang dekat, sedangkan objek yang berada pada kelompok yang berbeda akan memiliki jarak yang jauh (Suyanto, 2019). Terdapat sejumlah formula untuk mengukur jarak antara dua objek runtun waktu, salah satunya adalah satunya adalah *dynamic time warping*.

Dynamic time warping adalah salah satu ukuran kemiripan yang terkenal dalam mengukur jarak antar objek runtun waktu. Kelebihan dari *dynamic time warping* dapat mengukur objek runtun waktu dengan periode yang tidak sama panjangnya. Proses *dynamic time warping* menggunakan teknik pemrograman untuk menemukan jalur (*warping path*) yang mungkin dan memilih salah satu yang menghasilkan jarak minimum antara dua runtun waktu menggunakan matriks jarak, di mana masing-masing elemen dalam matriks adalah jarak

kumulatif dari minimal tiga tetangga terdekatnya. Ilustrasi dari *dynamic time warping* disajikan pada Gambar 2.1.



(Sumber: Guna dkk., 2014)

Gambar 2.1 Jarak euclidean (kiri) dan jarak DTW (kanan)

Misalkan terdapat dua set data runtun waktu $Z(1)_t = Z_1, Z_2, Z_3, \dots, Z_i$ dan $Z(2)_t = Z_1, Z_2, Z_3, \dots, Z_j$ di mana $i = 1, 2, 3, \dots, n$ dan $j = 1, 2, 3, \dots, m$ akan membentuk sebuah matriks yang disebut dengan matriks C berukuran $m \times n$ dengan elemen ke- (i, j) adalah selisih antara $Z(1)_i$ dengan $Z(2)_j$, kemudian ditambah dengan tiga elemen dengan nilai minimum yang berdekatan $\{c_{(i-1)(j-1)}, c_{(i-1)j}, c_{i(j-1)}\}$, di mana $0 < i \leq n$ dan $0 < j \leq m$. Matriks C untuk elemen ke- i dan ke- j adalah sebagai berikut:

$$c_{ij} = w_{ij} + \min \{c_{(i-1)(j-1)}, c_{(i-1)j}, c_{i(j-1)}\} \quad (2.4)$$

dengan w_{ij} adalah nilai mutlak dari selisih antara $Z(1)_i$ dengan $Z(2)_j$. Jarak *dynamic time warping* dari kedua set runtun waktu tersebut adalah jarak minimum dari *warping path* yang mungkin dan dapat dirumuskan sebagai

$$d_{DTW}(Z(1)_t, Z(2)_t) = \min_{w \in P} \left(\sqrt{\sum_{i,j=1}^s c_{ij}} \right) \quad (2.5)$$

dengan P adalah sekumpulan dari semua *warping path* yang mungkin, c_{ij} adalah elemen (i, j) pada *warping path*, dan s adalah panjang dari *warping path* (Chotirat & Ratanamahatana, 2007).

2.5 Metode *K-Medoids Clustering*

Menurut Han dkk. (2012), kelemahan metode pengelompokan *K-Means* adalah sensitif terhadap data *outlier*. *Outlier* terjadi ketika suatu data berada jauh dari sekumpulan data lainnya. Dengan demikian, ketika objek tersebut ditujukan ke sebuah klaster, maka akan mendistorsi nilai-rata-rata dari klaster tersebut. Alih-alih mengambil nilai rata-rata dari objek-objek di dalam sebuah klaster sebagai titik acuan, seseorang dapat mengambil objek aktual untuk mewakili klaster dengan menggunakan satu objek representatif setiap klaster. Metode ini disebut dengan *K-Medoids* untuk mengatasi kelemahan *K-Means*.

Partitioning Around Method (PAM) adalah salah satu metode *clustering* tipe *K-Medoids*. Konsep PAM adalah menentukan K klaster pada N objek dengan terlebih dahulu menentukan objek representatif (*medoids*) untuk setiap klaster. Setelah pemilihan awal *K-Medoids*, metode tersebut berulang kali mencoba untuk membuat pilihan yang lebih baik dari *medoids* dengan menganalisis semua kemungkinan pasangan objek sedemikian rupa sehingga satu objek adalah *medoids* dan tidak untuk yang lainnya. Ukuran kualitas pengelompokan terbaik dihitung untuk setiap kombinasi tersebut, pilihan terbaik dari titik dalam satu iterasi dipilih sebagai *medoids* untuk iterasi berikutnya.

Definisi *K-Medoids* adalah metode pengelompokan yang menggunakan objek representatif (*medoids*) sebagai titik pusat atau *centroid*. Setiap objek yang tersisa dikelompokkan dengan objek representatif yang paling mirip. Dengan demikian metode partisi dapat dilakukan dengan meminimalkan jumlah ketidakmiripan antara setiap objek p dan objek representatif terdekat, yaitu menggunakan jumlah kesalahan absolut. Kesalahan absolut didefinisikan sebagai

$$E = \sum_{i=1}^K \sum_{p \in C_i} \text{dist}(p, o_i) \quad (2.6)$$

di mana E adalah jumlah kesalahan absolut untuk semua objek p dalam himpunan data, dan o_i adalah objek representatif klaster C_i .

Langkah-langkah dari algoritma *K-Medoids* adalah sebagai berikut:

1. Memilih K objek pada sekumpulan N objek sebagai objek representatif (*medoids*).

2. Menghitung jarak *dynamic time warping* untuk setiap objek terhadap masing-masing *medoids*.
3. Menetapkan setiap objek ke klaster yang sesuai berdasarkan jarak *medoids* terdekat dan menghitung fungsi objektif yang merupakan jumlah jarak dari semua objek ke objek *medoids* terdekat.
4. Memilih salah satu objek dari masing-masing klaster (*non-medoids*) sebagai kandidat *medoids* yang baru.
5. Menghitung kembali jarak *dynamic time warping* untuk setiap objek terhadap masing-masing *non-medoids*.
6. Menetapkan setiap objek ke klaster yang sesuai berdasarkan jarak *non-medoids* terdekat dan menghitung fungsi objektif baru yang merupakan jumlah jarak dari semua objek ke objek *non-medoids* terdekat.
7. Menghitung selisih dari fungsi objektif baru dengan fungsi objektif yang lama, jika selisih dari fungsi objektif bernilai negatif ($S < 0$) dalam artian fungsi objektif baru memiliki total jarak yang lebih kecil, maka *medoids* sebelumnya diganti dengan *medoids* baru.
8. Mengulangi langkah 4 hingga langkah 7 sampai menghasilkan total jarak atau fungsi objektif yang minimum

2.6 Evaluasi *Clustering*

Evaluasi *clustering* dapat mengetahui ukuran seberapa baik suatu data dikelompokkan. Koefisien *silhouette* salah satu metode evaluasi yang merupakan penggabungan dari metode *cohesion* dan *separation* sehingga dapat melihat kualitas dan kekuatan klaster. Koefisien *silhouette* memberikan informasi sesuai dengan jumlah klaster pada *dataset*. Langkah-langkah dalam menghitung koefisien *silhouette* adalah sebagai berikut:

1. Menghitung rata-rata jarak dari suatu objek ke-*i* dengan semua objek yang berada pada satu klaster yang sama.
2. Menghitung rata-rata jarak suatu objek ke-*i* dengan semua objek yang berada pada klaster yang berbeda.
3. Menghitung nilai koefisien *silhouette* dengan formula pada Persamaan (2.7).

$$SC(i) = \frac{b_i - a_i}{\max\{a_i, b_i\}}, \quad (2.7)$$

dengan:

a_i : rata-rata jarak objek ke- i dengan semua data pada satu klaster yang sama

b_i : rata-rata jarak objek ke- i dengan semua objek pada klaster yang berbeda

Nilai koefisien *silhouette* berkisar antara -1 hingga 1. Hasil pengelompokan dikatakan baik jika nilai koefisien *silhouette* yang dimiliki sebesar 1, artinya objek ke- i sudah bergabung dalam klaster yang tepat. Jika nilai koefisien *silhouette* sebesar -1, artinya struktur klaster yang dihasilkan tidak baik. Koefisien *silhouette* yang digunakan adalah koefisien *silhouette* global di mana rata-rata dari setiap $SC(i)$ (Kaufman & Rousseeuw, 1990).

2.7 Peramalan

Peramalan (*forecasting*) merupakan salah satu analisis runtun waktu yang mendukung dalam proses pengambilan keputusan. Menurut Aswi dan Sukarna (2006), peramalan adalah suatu langkah untuk memperkirakan apa yang akan terjadi pada masa yang akan datang dengan mempertimbangkan data masa lampau maupun data saat ini. Peramalan merupakan bagian integral dari kegiatan pengambilan keputusan, sebab efektif atau tidaknya suatu keputusan umumnya bergantung pada beberapa faktor.

Tujuan dari peramalan adalah sebagai penduga perubahan yang akan terjadi dan dilakukan untuk menghadapi kondisi yang tidak pasti dengan cara meminimumkan kesalahan dalam meramalkan melalui pengukuran tingkat kesalahan dalam meramal. Peramalan terjadi karena adanya jangka waktu antar kebutuhan mendatang dengan peristiwa itu sendiri (Makridakis dkk., 1999).

Metode yang digunakan dalam peramalan dapat berpengaruh terhadap hasil ramalan yang diperoleh. Berdasarkan metodenya, peramalan dibagi menjadi dua kategori utama, yaitu metode kualitatif dan metode kuantitatif. Metode peramalan kualitatif banyak menuntut analisis yang didasarkan pada pemikiran

intuitif, pemikiran logis, dan informasi yang telah diperoleh peneliti sebelumnya. Sedangkan pada peramalan kuantitatif memerlukan informasi yang telah terjadi pada masa lampau yang dikuantitatifkan dalam bentuk data numerik, sehingga dapat diramalkan menggunakan perhitungan statistika dan matematika. Model pada metode peramalan kuantitatif terbagi menjadi dua, yaitu model regresi dan model runtun waktu. Model regresi yaitu model yang melibatkan dan menguji variabel yang diduga mempengaruhi variabel dependen dengan tujuan menemukan bentuk hubungan tersebut dan menggunakannya sebagai estimasi nilai variabel dependen dari variabel independen. Sedangkan untuk model runtun waktu melibatkan data historis dalam meramalkan kondisi masa yang akan datang dengan mengesktrapolasikan pola tersebut ke masa yang akan datang (Makridakis dkk., 1999).

Peramalan kuantitatif dapat diterapkan ketika terdapat situasi sebagai diantaranya :

1. Terdapat informasi masa lalu
2. Informasi tersebut dapat dikuantitatifkan dalam bentuk data numerik
3. Dapat diasumsikan bahwa aspek pola masa lalu akan terus berlanjut di masa mendatang.

Menurut Ramdhani (2014), peramalan dapat diklasifikasikan menjadi 3 kategori berdasarkan dari segi jangka waktu ramalan yang disusun, yaitu:

- a. Peramalan jangka pendek, yaitu peramalan dengan jangka waktu kurang atau sama dengan 3 periode.
- b. Peramalan jangka menengah, yaitu peramalan dengan jangka waktu antara 3 periode sampai dengan 18 periode.
- c. Peramalan jangka panjang, yaitu peramalan dengan jangka waktu lebih dari 18 periode.

2.8 Analisis Runtun Waktu

Makridakis dkk. (1999) menjelaskan analisis runtun waktu adalah suatu prosedur dalam statistika yang digunakan untuk meramalkan struktur probabilitas keadaan yang akan datang guna pengambilan suatu keputusan. Analisis runtun

waktu didasarkan pada pengamatan sekarang (Z_t) dan dipengaruhi oleh satu atau beberapa pengamatan sebelumnya (Z_{t-n}). Menurut Aswi dan Sukarna (2006), analisis runtun waktu merupakan salah satu metode statistika yang berupaya untuk meramalkan kondisi masa yang akan datang dengan memperhatikan data historis dan mengekstrapolasikan pola tersebut ke masa yang akan datang.

Data runtun waktu merupakan sekumpulan data berupa nilai pengamatan yang diukur selama periode waktu tertentu berdasarkan interval waktu yang tetap (Wei, 2006). Menurut Makridakis dkk. (1999), data runtun waktu berhubungan dengan data statistik yang dicatat dalam interval waktu, seperti penjualan, harga, persediaan, produksi, nilai tukar (*kurs*), harga saham dan lain sebagainya.

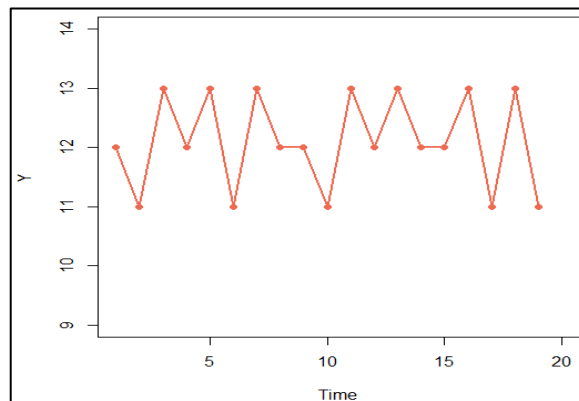
Suatu urutan pengamatan memiliki model runtun waktu jika memenuhi asumsi berikut (Aswi & Sukarna, 2006):

1. Interval waktu dapat dinyatakan dalam satuan waktu yang sama (identik).
2. Adanya ketergantungan atau hubungan antara kejadian masa mendatang terhadap masa sebelumnya atau lebih dikenal dengan istilah adanya otokorelasi antara suatu variabel pada waktu tertentu dengan variabel itu sendiri pada waktu-waktu sebelumnya.
3. Data masa depan mengikuti pola data yang terjadi di masa lalu.

Data runtun waktu yang terjadi di masa lalu digunakan dalam proses peramalan dengan meramalkan pola yang terbentuk dari data runtun waktu tersebut ke masa yang akan datang. Makridakis dkk. (1999) menjelaskan 4 jenis pola data runtun waktu, yaitu:

1. Pola Horizontal (H)

Pola horizontal terjadi ketika data berfluktuasi di sekitar rata-rata yang konstan. Artinya pola data tersebut tidak menunjukkan peningkatan atau penurunan secara signifikan selama waktu tertentu. Pola data horizontal diilustrasikan pada Gambar 2.2.

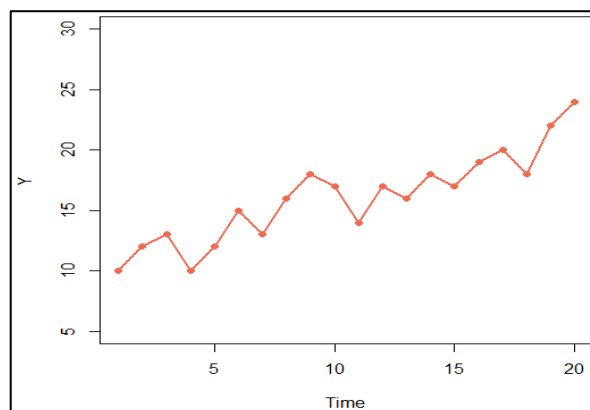


Sumber: Simulasi *software R*

Gambar 2.2 Pola data horizontal

2. Pola *Trend* (T)

Pola *trend* terjadi ketika dalam data terdapat kenaikan atau penurunan sekuler jangka panjang dalam data. Pola data *trend* diilustrasikan pada Gambar 2.3.

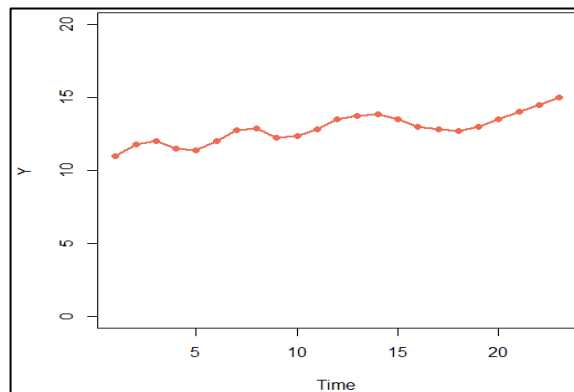


Sumber: Simulasi *software R*

Gambar 2.3 Pola data *trend*

3. Pola Siklis (C)

Pola siklis terjadi ketika data dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis. Pola data siklis diilustrasikan pada Gambar 2.4.

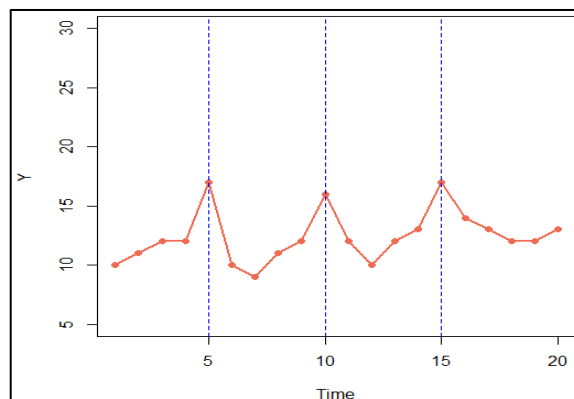


Sumber: Simulasi *software R*

Gambar 2.4 Pola data siklis

4. Pola Musiman (S)

Pola musiman terjadi ketika suatu runtun waktu dipengaruhi oleh faktor musiman. Komponen musiman ditunjukkan pada suatu pola perubahan yang berulang secara periodik. Pola data musiman diilustrasikan pada Gambar 2.5.



Sumber: Simulasi *software R*

Gambar 2.5 Pola data musiman

2.9 Stasioneritas Data

Asumsi data yang stasioner pada beberapa metode peramalan harus dipenuhi dalam pembentukan model analisis runtun waktu. Stasioneritas mempunyai makna bahwa tidak terdapat perubahan yang drastis pada data, artinya fluktuasi data berada di sekitar suatu nilai rata-rata yang konstan, tidak tergantung pada waktu dan variansi dari fluktuasi tersebut. Peramalan dapat dilakukan apabila kondisi data sudah stasioner. Data yang tidak stasioner mempunyai tiga

kondisi, yaitu tidak stasioner dalam rata-rata, tidak stasioner dalam variansi ataupun tidak stasioner keduanya (Makridakis dkk., 1999).

Aswi dan Sukarna (2006) menjelaskan konsep stasioner secara statistika dapat diuraikan sebagai berikut:

- a. Apabila suatu data runtun waktu dibuat diagramnya dan kemudian tidak ada perubahan rata-rata yang jelas dari waktu ke waktu, dikatakan bahwa runtun data tersebut stasioner pada rata-rata.
- b. Apabila grafik runtun waktu tidak memperlihatkan adanya perubahan varians yang jelas dari waktu ke waktu, dapat dikatakan runtun data tersebut stasioner dalam varians.
- c. Apabila rata-rata mengalami perubahan dari waktu ke waktu dengan kata lain menyimpang dengan beberapa pola siklus kecenderungan (*trend-cycle*), maka runtun waktu tersebut mempunyai rata-rata yang tidak stasioner.
- d. Apabila rata-rata suatu data runtun waktu menyimpang (berubah setiap waktu) dan varians tidak konstan setiap waktu, runtun waktu tersebut memiliki rata-rata dan varians yang tidak stasioner.

2.9.1 Stasioneritas dalam Variansi

Suatu data runtun waktu dikatakan stasioner dalam variansi jika data berfluktuasi secara konstan terhadap variansi atau tidak berubah-ubah dari waktu ke waktu. Teknik dalam melihat kestasioneran data dalam variansi dapat dilihat dalam bentuk visual dengan melihat perubahan pola dan fluktuasi data dari waktu ke waktu menggunakan grafik runtun waktu (Makridakis dkk., 1999).

Aswi dan Sukarna (2006) menjelaskan cara mengatasi data yang tidak stasioner dengan melakukan transformasi pada data untuk membuat data tersebut menjadi stasioner. Transformasi data adalah suatu cara untuk mengubah skala pengukuran data asli menjadi bentuk lain sehingga data dapat memenuhi asumsi-asumsi yang mendasari suatu analisis. Transformasi tersebut dinamakan transformasi pangkat (*power transformation*) yang dikenalkan oleh Box dan Cox (1964). Secara umum transformasi yang digunakan:

$$Z_t^* = \begin{cases} \frac{Z_t^\lambda - 1}{\lambda} & , \lambda \neq 0 \\ \ln Z_t & , \lambda = 0 \end{cases} \quad (2.8)$$

di mana Z_t adalah data runtun waktu dan λ adalah parameter transformasi. Beberapa penggunaan nilai λ serta kaitannya dengan transformasinya disajikan pada Tabel 2.1.

Tabel 2.1 Beberapa Nilai λ dengan Transformasi Box-Cox

Nilai λ	Transformasi
-1	$\frac{1}{Z_t}$
-0,5	$\frac{1}{\sqrt{Z_t}}$
0	$\ln Z_t$
0,5	$\sqrt{Z_t}$
1	Z_t

Sumber: Aswi & Sukarna (2006)

Ketika menstabilkan variansi terdapat beberapa ketentuan, yaitu:

1. Transformasi boleh dilakukan hanya untuk Z_t yang positif ($Z_t > 0$).
2. Transformasi dilakukan sebelum melakukan pembedaan dan pemodelan runtun waktu.
3. Nilai λ dipilih berdasarkan *Sum Square Error* (SSE) dari runtun waktu hasil transformasi. Nilai SSE terkecil memberikan hasil variansi yang konstan.
4. Transformasi tidak hanya menstabilkan variansi, tetapi juga dapat menormalkan distribusi.

2.9.2 Stasioneritas dalam Rata-rata

Menurut Makridakis dkk. (1999), suatu data runtun waktu dikatakan stasioner dalam rata-rata jika data berfluktuasi di sekitar nilai rata-rata yang konstan, tidak tergantung pada waktu dan variansi dari fluktuasi tersebut. Stasioner terhadap rata-rata dapat dilihat dalam bentuk visual dengan melihat pergerakan dari runtun waktu menggunakan grafik runtun waktu. Apabila dilihat

dari grafik fungsi otokorelasi (FOK), maka nilai-nilai otokorelasi dari data stasioner akan turun menjadi nol sesudah *lag* waktu kedua atau ketiga.

Apabila data tidak stasioner dalam rata-rata, maka dapat dikonversikan menjadi runtun data stasioner melalui *differencing*, yaitu runtun data asli diganti dengan selisih. Jumlah *differencing* yang dilakukan untuk mencapai stasioner dinotasikan sebagai d . Bentuk *differencing* pertama ($d=1$) dituliskan pada Persamaan (2.9).

$$BZ_t = Z_{t-1} \quad (2.9)$$

Operator *backshift* pada Persamaan (2.9) mempunyai pengaruh menggeser data 1 periode ke belakang, sehingga jika digunakan dua notasi B akan menggeser data tersebut 2 periode ke belakang dituliskan pada Persamaan (2.10).

$$B(BZ_t) = B^2Z_t = Z_{t-2} \quad (2.10)$$

Apabila suatu data runtun waktu tidak stasioner, maka data tersebut dapat dibuat menjadi stasioner dengan melakukan *differencing* dituliskan pada Persamaan (2.11).

$$\nabla = Z_t - Z_{t-1} \quad (2.11)$$

Operator *backshift* (B) menurut Persamaan (2.11) dapat ditulis kembali menjadi:

$$\nabla = Z_t - BZ_t = (1 - B)Z_t \quad (2.12)$$

Pada Persamaan (2.12), *differencing* pertama dinyatakan oleh $(1 - B)Z_t$. Sama halnya apabila *differencing* kedua, yaitu *differencing* pertama dari *differencing* pertama sebelumnya harus dihitung, sehingga:

$$\begin{aligned} \nabla^2 &= (Z_t - Z_{t-1}) - (Z_{t-1} - Z_{t-2}) \\ &= Z_t - 2Z_{t-1} + Z_{t-2} \\ &= (1 - 2B + B^2)Z_t \\ &= (1 - B)^2Z_t \end{aligned} \quad (2.13)$$

Pada Persamaan (2.13), *differencing* kedua dinyatakan oleh $(1 - B)^2Z_t$. Sedangkan secara umum *differencing* orde ke- d untuk menghasilkan suatu proses baru yang stasioner, misalnya ∇^d dituliskan pada Persamaan (2.14).

$$\nabla^d = (1 - B)^dZ_t \quad (2.14)$$

Proses *differencing* dapat dilakukan hingga hasil *differencing* menunjukkan kondisi stasioner terhadap rata-rata dan grafik FOK menurun mendekati nol sesudah *lag* waktu kedua atau ketiga (Hanke & Wichern, 2003).

Pengecekan kestasioneran data runtun waktu terhadap rata-rata dapat pula menggunakan pengujian hipotesis akar unit (*unit root test*) atau uji *Augmented Dickey-Fuller* (ADF). Menurut Widarjono (2007), uji ADF adalah salah satu uji yang paling sering digunakan dalam pengujian stasioneritas dari data, yakni dengan melihat apakah di dalam model terdapat *unit root* atau tidak. Data dikatakan stasioner apabila data berpola horizontal, tidak mengandung komponen *tren* dengan keragaman yang konstan serta tidak terdapat fluktuasi. Uji ADF memiliki persamaan model yang dituliskan pada Persamaan (2.15).

$$\nabla Z_t = \delta Z_{t-1} + \sum_{i=2}^p \gamma_i \nabla Z_{t-1} + e_t \quad (2.15)$$

dengan:

∇Z_t : *differencing* pertama dari Z_t

δ : koefisien regresi untuk *lag*

γ : koefisien regresi untuk *lag* perbedaan

e_t : *error* pada waktu ke- t

Hipotesis yang digunakan pada pengujian ADF adalah sebagai berikut:

Hipotesis

$H_0 : \delta = 0$ (Data tidak stasioner dalam rata-rata)

$H_1 : \delta \neq 0$ (Data stasioner dalam rata-rata)

Taraf Signifikansi

$\alpha = 5\%$

Statistik Uji

$$\tau = \frac{\hat{\delta}}{SE(\hat{\delta})} \quad (2.16)$$

dengan,

$\hat{\delta}$: nilai estimasi dari δ

$SE(\hat{\delta})$: *standard error* dari $\hat{\delta}$

Daerah Penolakan

H_0 ditolak jika $|\tau| > |\tau_{\alpha;db}|$, di mana $\tau_{\alpha;db}$ adalah nilai tabel *MacKinnon* yang memiliki derajat bebas $(db) = n - n_p$, dengan n adalah banyaknya data dan n_p adalah banyaknya parameter atau menolak H_0 jika $p - value < \alpha$.

2.10 Fungsi Otokorelasi dan Otokorelasi Parsial

Apabila kondisi stasioner baik dalam rata-rata maupun dalam variansi telah terpenuhi, identifikasi model dengan membuat grafik fungsi otokorelasi (FOK) adalah tahapan selanjutnya yang dilakukan. Grafik FOK dapat digunakan dalam mengidentifikasi model dan kestasioneran data. Selain grafik FOK, terdapat pula grafik fungsi otokorelasi parsial (FOKP). FOKP membantu menetapkan model ARIMA untuk peramalan, sehingga identifikasi model untuk pemodelan data runtun waktu memerlukan perhitungan dan penggambaran dari hasil fungsi FOK dan FOKP. Melalui perhitungan FOK dan FOKP maka dapat ditentukan model AR atau model MA dengan orde tertentu dari data yang sedang dianalisis (Aswi & Sukarna, 2006).

2.10.1 Fungsi Otokorelasi (FOK)

Fungsi Otokorelasi (FOK) adalah alat utama dalam menentukan model yang cocok pada suatu data observasi. FOK berfungsi mengukur korelasi antar pengamatan dengan *lag* k . Proses stasioner suatu data runtun waktu Z_t diperoleh rata-rata $E(Z_t) = \mu$ dan variansi $var(Z_t) = E(Z_t - \mu)^2 = \sigma^2$ yang konstan dan kovariansi $Cov(Z_t, Z_{t+k})$ yang fungsinya hanya pada perbedaan waktu. Sehingga, dapat ditulis kovariansi antara Z_t dan Z_{t+k} adalah:

$$\gamma_k = cov(Z_t, Z_{t+k}) = E(Z_t - \mu)(Z_{t+k} - \mu) \quad (2.17)$$

dan korelasi antara Z_t dan Z_{t+k} dituliskan pada Persamaan (2.18).

$$\rho_k = \frac{cov(Z_t, Z_{t+k})}{\sqrt{var(Z_t)}\sqrt{var(Z_{t+k})}} = \frac{\gamma_k}{\gamma_0} \quad (2.18)$$

dengan notasi $var(Z_t) = var(Z_{t+k}) = \gamma_0$. Sebagai fungsi k , γ_k disebut fungsi otokovariansi dan ρ_k disebut fungsi otokorelasi (FOK). Pada analisis runtun

waktu, γ_k dan ρ_k menggambarkan kovariansi dan korelasi antara Z_t dan Z_{t+k} dari proses yang sama, hanya dipisahkan oleh *lag* ke- k (Wei, 2006).

Menurut Aswi dan Sukarna (2006), koefisien otokorelasi diuji untuk menentukan apakah terdapat perbedaan nilai yang signifikan dengan nol atau tidak. Koefisien otokorelasi ρ_k dapat diestimasi dengan persamaan (2.19) berikut:

$$\hat{\rho}_k = \frac{\sum_{t=1}^{n-k} (Z_t - \bar{Z})(Z_{t+k} - \bar{Z})}{\sum_{t=1}^n (Z_t - \bar{Z})^2}, k = 0, 1, 2, \dots \quad (2.19)$$

dengan:

ρ_k : koefisien otokorelasi pada *lag*- k

k : selisih waktu (*lag*)

n : banyaknya data

$\bar{Z}$: rata-rata dari Z_t

Z_t : data pada waktu ke- t

Z_{t+k} : data pada waktu ke- $t+k$

Grafik FOK dapat digunakan sebagai alat untuk mengidentifikasi kestasioneran data. Jika grafik FOK cenderung turun lambat atau turun lambat atau turun secara linier, maka dapat disimpulkan data belum stasioner dalam rata-rata.

2.10.2 Fungsi Otokorelasi Parsial (FOKP)

Aswi dan Sukarna (2006) mendefinisikan Fungsi Otokorelasi Parsial (FOKP) sebagai alat untuk mengukur tingkat keeratan antara Z_t dan Z_{t+k} , apabila pengaruh dari *lag* waktu 1,2,3,... dan seterusnya sampai $k - 1$ dianggap terpisah. FOKP mengidentifikasi model yang sesuai dengan data pengamatan dengan menunjukkan besarnya korelasi parsial antara pengamatan pada waktu ke t (Z_t) dengan pengamatan pada waktu-waktu setelahnya (Z_{t+k}).

FOKP ditulis dengan notasi ϕ_{kk} dan didefinisikan sebagai berikut:

$$\phi_{kk} = \text{corr}(Z_t, Z_{t+k} | Z_{t-1}, Z_{t-2}, \dots, Z_{t-k+1}) \quad (2.20)$$

Nilai ϕ_{kk} dapat ditentukan melalui persamaan Yule Walker pada *lag* ke- k sebagai

$$\rho_j = \phi_{k1}\rho_{j-1} + \phi_{k2}\rho_{j-2} + \dots + \phi_{kk}\rho_{j-k} \quad (2.21)$$

Sehingga untuk dengan $j=1,2,3,\dots,k$ akan berlaku persamaan sebagai berikut:

$$\begin{aligned}
\rho_1 &= \phi_{k1}\rho_0 + \phi_{k2}\rho_1 + \cdots + \phi_{kk}\rho_{k-1} \\
\rho_2 &= \phi_{k1}\rho_1 + \phi_{k2}\rho_0 + \cdots + \phi_{kk}\rho_{k-2} \\
&\vdots \\
\rho_k &= \phi_{k1}\rho_{k-1} + \phi_{k2}\rho_{k-2} + \cdots + \phi_{kk}\rho_0
\end{aligned} \tag{2.22}$$

Sistem persamaan diatas dapat diubah dalam bentuk matriks sebagai berikut:

$$\begin{bmatrix} 1 & \rho_1 & \rho_2 & \rho_3 & \cdots & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \rho_{k-4} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \phi_{k1} \\ \phi_{k2} \\ \vdots \\ \phi_{kk} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_k \end{bmatrix} \tag{2.23}$$

dengan menggunakan metode Cramer, untuk $k = 1$ diperoleh:

$$\phi_{11} = \rho_1 \tag{2.24}$$

untuk $k = 2$,

$$\phi_{22} = \frac{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & \rho_2 \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{vmatrix}} \tag{2.25}$$

untuk $k = 3$,

$$\phi_{33} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_2 \\ \rho_2 & \rho_1 & \rho_3 \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{vmatrix}} \tag{2.26}$$

dan seterusnya sehingga nantinya diperoleh nilai fungsi otokorelasi parsial untuk lag k dan $j = 1, 2, 3, \dots, k$ sebagai berikut:

$$\phi_{kk} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \rho_3 & \cdots & \rho_1 \\ \rho_1 & 1 & \rho_1 & \rho_2 & \cdots & \rho_2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \rho_{k-4} & \cdots & \rho_k \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \rho_3 & \cdots & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \rho_{k-4} & \cdots & 1 \end{vmatrix}} \tag{2.27}$$

Metode yang lebih efisien untuk menyelesaikan persamaan Yule Walker adalah sebagai berikut:

$$\hat{\phi}_{kk} = \frac{\hat{\rho}_k - \sum_{j=1}^{k-1} \hat{\phi}_{k-1,j} \hat{\rho}_{k-j}}{1 - \sum_{j=1}^{k-1} \hat{\phi}_{k-1,j} \hat{\rho}_j} \quad (2.28)$$

dengan

$$\hat{\phi}_{kj} = \hat{\phi}_{k-1,j} - \hat{\phi}_{kk} \hat{\phi}_{k-1,k-j}, \text{ untuk } j = 1, 2, \dots, k-1 \quad (2.29)$$

2.11 Model Runtun Waktu Box-Jenkins

Model *Autoregressive Integrated Moving Average* (ARIMA) telah dipelajari secara mendalam oleh George Box dan Gwilym Jenkins pada tahun 1976, kedua nama tersebut sering disinonimkan dengan proses ARIMA yang diterapkan untuk analisis runtun waktu, peramalan, dan pengendalian. Model ARIMA terdiri dari dua aspek, yaitu aspek *autoregressive* dan *moving average* (rata-rata bergerak). Secara umum, model ARIMA dituliskan dengan notasi ARIMA (p, d, q) , dimana p menyatakan orde dari proses *autoregressive* (AR), d menyatakan pembedaan (*differencing*), dan q menyatakan dari proses *moving average* (MA) (Aswi & Sukarna, 2006).

2.11.1 Proses Autoregressive

Proses *autoregressive* sesuai dengan namanya adalah proses hasil regresi dengan dirinya sendiri. Bentuk umum dari proses *autoregressive* tingkat p atau AR(p) dituliskan pada Persamaan (2.30).

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + e_t \quad (2.30)$$

Persamaan (2.30) menunjukkan bahwa nilai sekarang suatu proses menghubungkan dengan nilai-nilai sebelumnya pada selang waktu. Dalam hal ini diasumsikan e_t adalah independen dengan $Z_{t-1}, Z_{t-2}, \dots$. Jadi dapat dipandang Z_t diregresikan pada p nilai Z yang lalu. Persamaan (2.30) biasanya juga ditulis dalam bentuk

$$\phi_p(B)Z_t = e_t \quad (2.31)$$

dengan $\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$ dan B dikenal sebagai operator *backshift* (Salamah dkk., 2003).

2.11.2 Proses *Moving Average*

Bentuk umum dari *moving average* tingkat q atau $MA(q)$ didefinisikan

$$Z_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} \quad (2.32)$$

di mana e_t adalah independen dan berdistribusi normal dengan rata-rata nol, variansi σ_a^2 dan B adalah operator *backshift*, Persamaan (2.32) dapat ditulis dalam bentuk

$$Z_t = \theta_q(B) e_t \quad (2.33)$$

dengan $\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$ yang dikenal sebagai operator $MA(q)$. *Mean* dan *varians* dari model $MA(q)$ selanjutnya dapat dihitung yaitu

$$E(Z_t) = E(e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q}) = 0 \quad (2.34)$$

dan

$$Var(Z_t) = Var(e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q}) \quad (2.35)$$

untuk q berhingga, maka proses MA akan selalu stasioner.

(Salamah dkk., 2003).

2.11.3 Proses *Autoregressive Moving Average*

Suatu perluasan dari model AR dan MA adalah model campuran yang berbentuk

$$Z_t = \phi_1 Z_{t-1} + \dots + \phi_p Z_{t-p} + e_t - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} \quad (2.36)$$

yang dinamakan model $ARMA(p, q)$. Model ini dapat dituliskan pada Persamaan (2.37).

$$\phi_p(B) Z_t = \theta_q(B) e_t \quad (2.37)$$

Syarat-syarat stasioneritas dan invertibilitas memerlukan akar-akar dari $\phi(B) = 0$ dan $\theta(B) = 0$ terletak di luar lingkaran satuan. Dengan mengambil ekspektasi persamaan (2.36) diperoleh $E(Z_t) = 0$ (Salamah dkk., 2003).

Jika diketahui suatu data *time series* yang telah stasioner, langkah-langkah berikut merupakan tahapan yang bermanfaat untuk melakukan identifikasi model dugaan sementara yang sesuai sebagai berikut:

1. Membuat grafik runtun waktu

Menurut Salamah dkk. (2003), pada setiap analisis runtun waktu, tahap pertama yang harus dilakukan adalah membuat *plot* data. Melalui *plot* data

dapat diketahui apakah data mengandung *trend*, musiman, *outlier*, variansi tidak konstan, atau fenomena nonformal dan nonstasioner yang lain. Dalam kasus data runtun waktu yang stasioner, maka fenomena-fenomena ini tidak akan ditemukan.

2. Menghitung dan mencocokkan sampel FOK dan FOKP dengan bentuk-bentuk teoritiknya

Melalui identifikasi grafik FOK dan FOKP dari data yang telah stasioner, dapat menentukan orde p dan orde q dari model ARMA pada suatu data runtun waktu. Pola teoritis FOK dan FOKP dari proses data runtun waktu yang telah stasioner disajikan pada tabel 2.2 berikut:

Tabel 2.2 Pola Teoritis FOK dan FOKP dari Proses Data yang Stasioner

Proses	FOK	FOKP
$AR(p)$	<i>Dies Down</i> (Turun cepat secara eksponensial/sinusoidal)	<i>Cut off after lag p</i> (Terputus setelah <i>lag p</i>)
$MA(q)$	<i>Cut off after lag q</i> (Terputus setelah <i>lag q</i>)	<i>Dies Down</i> (Turun cepat secara eksponensial/sinusoidal)
$ARMA(p,q)$	<i>Dies Down</i> (Turun cepat secara eksponensial/sinusoidal)	<i>Dies Down</i> (Turun cepat secara eksponensial/sinusoidal)
$AR(p)$ atau $MA(q)$	<i>Cut off after lag q</i> (Terputus setelah <i>lag q</i>)	<i>Cut off after lag p</i> (Terputus setelah <i>lag p</i>)
<i>White Noise</i> (<i>Random</i>)	Tidak ada yang signifikan (tidak ada yang keluar batas)	Tidak ada yang signifikan (tidak ada yang keluar batas)

Sumber: Suhartono (2008)

Berdasarkan Tabel 2.2 dapat dijelaskan sebagai berikut:

- a. Jika grafik FOK menurun secara bertahap menuju ke-0 dan grafik FOKP menuju ke-0 setelah *lag p*, maka dugaan modelnya adalah $AR(p)$.

- b. Jika grafik FOK menuju ke-0 setelah *lag p*, dan grafik FOKP menurun secara bertahap menuju ke-0, maka dugaan modelnya adalah MA(*q*).
- c. Jika grafik FOK dan grafik FOKP menurun secara bertahap menuju ke-0, maka dugaan modelnya adalah ARMA (*p,q*).
- d. Jika grafik FOK dan grafik FOKP menuju ke-0 setelah *lag p* dan *q*, maka dugaan modelnya adalah AR(*p*) atau MA(*q*).
- e. Jika grafik FOK dan grafik FOKP tidak ada yang signifikan, maka dugaan modelnya adalah *white noise (random)*

2.11.4 Proses *Autoregressive Integrated Moving Average*

Model ARIMA(*p, d, q*) adalah gabungan model AR(*p*) dan MA(*q*) dengan pola data yang tidak stasioner dan dilakukan pembedaan dengan orde *d*. Suatu runtun waktu yang dihasilkan oleh proses ARIMA(*p, d, q*) dapat dinyatakan dalam bentuk observasi yang lalu, sesatan yang lalu dan sekarang. Proses ARIMA digambarkan dengan dimensi *p, d*, dan *q*. Jadi ARIMA(*p, d, q*) berarti suatu runtun waktu nonstasioner yang setelah diambil selisih ke *d* menjadi stasioner yang mempunyai model *autoregressive* tingkat *p* dan *moving average* tingkat *q* (Salamah dkk., 2003).

Suatu proses (Z_t) dikatakan mengikuti model ARIMA yang non-stasioner dalam rata-rata jika ada orde *d* ($d \geq 1$). Model umum untuk ARIMA(*p, d, q*) dituliskan pada Persamaan (2.38).

$$(1 - \phi_1 B - \dots - \phi_p B^p)(1 - B)^d Z_t = (1 - \theta_1 B - \dots - \theta_q B^q) e_t \quad (2.38)$$

Data runtun waktu tidak dapat langsung digunakan untuk mendapatkan model ARIMA, tetapi terlebih dahulu data tersebut distasionerkan (Aswi & Sukarna, 2006).

Pada dasarnya identifikasi model ARIMA adalah sama dengan identifikasi pada model ARMA. Perbedaan mendasar terletak pada tahap (a) di mana grafik runtun waktu menunjukkan data tidak stasioner, sehingga perlu dilakukan transformasi (jika variansi tidak konstan) dan *differencing* (jika *mean* tidak konstan) terlebih dahulu (Salamah dkk., 2003).

2.12 Estimasi Parameter Model Autoregressive Integrated Moving Average

Jika model awal ARIMA (p, d, q) telah diperoleh, selanjutnya parameter dari model tersebut ditaksir, sehingga diperoleh besaran koefisien model. Terdapat beberapa metode dalam menaksir parameter ARIMA Box-Jenkins, yaitu metode *moment*, metode *least square*, metode *maximum likelihood*, metode *unconditional least square*, dan metode *nonlinear estimation* (Aswi & Sukarna, 2006). Pada penelitian ini penaksiran parameter menggunakan metode *maximum likelihood*.

Metode *maximum likelihood* (ML) telah banyak digunakan dalam mengestimasi parameter (Wei, 2006). Salamah dkk. (2003) menjelaskan teknik estimasi ML diterapkan dengan membuat asumsi tentang bentuk fungsi probabilitas dari data yang teramati. Metode ini dimulai dengan mengasumsikan bahwa *error* (e_t) berdistribusi normal ($e_t \sim iid N(0, \sigma_e^2)$).

Fungsi kepadatan probabilitas (FKP) suatu *error* (e_t) adalah sebagai berikut:

$$f(e_t | \sigma_e^2) = (2\pi\sigma_e^2)^{-1/2} \exp\left(-\frac{e_t^2}{2\sigma_e^2}\right) \quad (2.39)$$

Karena *error* saling bebas, maka distribusi bersamanya dituliskan pada Persamaan (2.40).

$$f(e_1, e_2, \dots, e_n | \sigma_e^2) = (2\pi\sigma_e^2)^{-n/2} \exp\left(-\frac{1}{2\sigma_e^2} \sum_{t=1}^n e_t^2\right) \quad (2.40)$$

Setiap e_t dapat dinyatakan dalam bentuk observasi Z , parameter-parameter ϕ , θ dan σ_e^2 , serta *error-error* sebelumnya, yaitu:

$$e_t = Z_t - \phi_1 Z_{t-1} - \dots - \phi_p Z_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} \quad (2.41)$$

Persamaan (2.41) dapat dipandang sebagai hubungan berulang antara e_t yang berurutan, jika diketahui parameter-parameter dan observasi Z_t . Akibatnya nilai setiap e_t dapat dihitung sebagai fungsi parameter dan observasi.

Selanjutnya dengan mensubstitusikan Persamaan (2.41) ke dalam Persamaan (2.40) akan diperoleh fungsi kepadatan bersama Z sebagai berikut:

$$f(Z|\phi, \theta, \sigma_e^2) = (2\pi\sigma_e^2)^{-n/2} \exp\left(-\frac{1}{2\sigma_e^2} \sum_{t=1}^n (Z_t - \phi_1 Z_{t-1} - \dots - \phi_p Z_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q})^2\right) \quad (2.42)$$

Maka fungsi *likelihood* untuk parameter-parameternya jika diketahui data observasi adalah sebagai berikut:

$$L(\phi, \theta, \sigma_e^2|Z) = (2\pi\sigma_e^2)^{-n/2} \exp\left(-\frac{1}{2\sigma_e^2} S(\phi, \theta)\right) \quad (2.43)$$

dengan,

$$S(\phi, \theta) = \sum_{t=1}^n (Z_t - \phi_1 Z_{t-1} - \dots - \phi_p Z_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q})^2 \quad (2.44)$$

Karena pada akhirnya hanya akan mengarah pada besar *relative likelihood*, maka cukup jika hanya memandang $\ln$ *likelihood* yang diberikan dengan rumus sebagai berikut:

$$l(\phi, \theta, \sigma_e^2) = -\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma_e^2 - \frac{1}{2\sigma_e^2} S(\phi, \theta) \quad (2.45)$$

Dapat dilihat bahwa parameter-parameter ϕ dan θ hanya masuk dalam bagian jumlah kuadrat fungsi *likelihood*, dengan demikian untuk memaksimumkan *likelihood* kita hanya perlu meminimumkan fungsi jumlah kuadrat untuk seluruh nilai parameter-parameter itu. Setelah estimasi ML parameter-parameter itu diperoleh, dengan mudah dapat ditunjukkan bahwa estimasi ML parameter-parameter itu diperoleh, dengan mudah dapat ditunjukkan bahwa estimasi ML untuk σ_e^2 dituliskan pada Persamaan (2.46).

$$\hat{\sigma}_e^2 = \frac{\hat{S}(\hat{\phi}, \hat{\theta})}{n} \quad (2.46)$$

Pada penaksiran parameter model dengan ML *estimation* tidak dapat diselesaikan secara analisis untuk mendapatkan hasil eksplisit. Oleh karena itu untuk memperoleh nilai estimasi parameternya digunakan iterasi dengan metode Newton-Raphson (Salamah dkk., 2003).

2.13 Pengujian Signifikansi Parameter

Menurut Aswi dan Sukarna (2006), model ARIMA yang baik dalam menggambarkan suatu kejadian adalah model yang salah satunya menunjukkan bahwa penaksiran parameternya signifikan berbeda dengan nol.

Misalkan ω adalah suatu parameter model ARIMA (ϕ, θ) dan $\hat{\omega}$ adalah estimator dari parameter tersebut, serta $SE(\hat{\omega})$ adalah *standard error* dari estimator ω , tahapan dari pengujian signifikansi parameter adalah sebagai berikut:
Hipotesis

$H_0 : \omega = 0$ (Parameter tidak signifikan pada model)

$H_1 : \omega \neq 0$ (Parameter signifikan pada model)

Taraf Signifikansi

$\alpha = 5\%$

Statistik Uji

$$t_{hit} = \frac{\hat{\omega}}{SE(\hat{\omega})} \quad (2.47)$$

Daerah Penolakan

H_0 ditolak jika $|t_{hit}| > t_{\alpha/2, db}$ atau $p - value < \alpha$ di mana $t_{\alpha/2, db}$ diperoleh dari nilai tabel kritis t dengan α adalah taraf signifikansi dan db adalah derajat bebas di mana $db = n - n_p$, dengan n_p adalah banyaknya parameter (Salamah dkk., 2003).

2.14 Pemeriksaan Diagnostik Model

Pemeriksaan diagnostik model dilakukan untuk memeriksa apakah model yang telah diestimasi sudah sesuai dengan data yang ada. Apabila dijumpai penyimpangan yang cukup serius, maka harus merumuskan kembali model yang baru, yang selanjutnya dilakukan estimasi dan verifikasi. Melalui verifikasi akan diketahui bagaimana model yang kurang cocok dimodifikasi menjadi model baru (Salamah dkk., 2003). Proses pemeriksaan diagnostik model terbagi menjadi dua, antara lain pengujian *error* berdistribusi normal dan pengujian independensi *error*.

2.14.1 Pengujian *Error* Berdistribusi Normal

Uji asumsi ini bertujuan untuk mengetahui apakah data *error* telah memenuhi asumsi kenormalan. Salah satu uji kenormalan adalah menggunakan uji *Kolmogorov-Smirnov* dengan rumusan hipotesis:

Hipotesis

$H_0 : F_0(Z) = F_n(Z)$ (Fungsi distribusi kumulatif = fungsi distribusi empiris)

$H_1 : F_0(Z) \neq F_n(Z)$ (Fungsi distribusi kumulatif $\neq$ fungsi distribusi empiris)

Taraf Signifikansi

$\alpha = 5\%$

Statistik Uji

$$D_{hit} = \sup \{|F_0(Z) - F_n(Z)|\} \quad (2.48)$$

dengan,

$F_0(Z)$: fungsi distribusi kumulatif

$F_n(Z)$: fungsi distribusi empiris

Daerah Penolakan

H_0 ditolak jika $D_{hit} > D_{\alpha,n}$ atau $p - value < \alpha$ di mana $D_{\alpha,n}$ diperoleh dari nilai tabel kritis D dalam uji satu sampel *Kolmogorov-Smirnov*, dengan α adalah taraf signifikansi dan n adalah banyaknya data (Aswi & Sukarna, 2006).

2.14.2 Pengujian Independensi *Error*

Pengujian terhadap *error* dapat dilakukan secara individu ataupun bersama-sama. Pengujian secara individu dapat dilakukan jika diketahui distribusi dari estimasi *error* secara umum mendekati normal dengan rata-rata nol. Sedangkan pengujian secara bersama dapat menggunakan uji *Ljung-Box* yang dapat dituliskan sebagai berikut:

Hipotesis

$H_0 : \rho_1 = \rho_2 = \dots = \rho_k = 0$

(Tidak terdapat korelasi antar *error*)

H_1 : minimal ada satu $\rho_j \neq 0$, untuk $j = 1, 2, 3, \dots, k$

(Terjadi korelasi antar *error*)

Taraf Signifikansi

$\alpha = 5\%$

Statistik Uji

$$Q^* = n(n+2) \sum_{k=1}^{k^*} \frac{\hat{\rho}_k^2}{(n-k)} \quad (2.49)$$

di mana,

n : banyaknya data

k : nilai *lag*

k^* : *lag* maksimum

$\hat{\rho}_{e_k}$: estimator fungsi otokorelasi data *error* pada *lag* k

Daerah Penolakan

H_0 ditolak jika $Q^* > \chi_{\alpha, db}^2$ atau $p\text{-value} < \alpha$ di mana $\chi_{\alpha, db}^2$ diperoleh dari nilai tabel kritis *chi-square* dengan α adalah taraf signifikansi dan db adalah derajat bebas di mana $db = k^*$, dengan k^* adalah maksimum *lag* (Ljung & Box, 1978).

2.15 Ukuran Akurasi Peramalan

Pada model data runtun waktu, ada kemungkinan terdapat beberapa model yang sesuai yaitu semua parameter signifikan dan *error* memenuhi asumsi *white noise*. Penentuan model terbaik dari beberapa model yang telah memenuhi syarat tersebut, dibutuhkan kriteria terbaik dan akurat (Aswi & Sukarna, 2006). Ukuran akurasi peramalan yang akan digunakan untuk pemilihan model ARIMA yang terbaik pada penelitian ini adalah *Akaike's Information Criterion* (AIC) dan *Mean Absolute Percentage Error* (MAPE).

2.15.1 Akaike's Information Criterion (AIC)

Wei (2006) menjelaskan *Akaike's Information Criterion* (AIC) adalah suatu kriteria pemilihan model terbaik yang diperkenalkan oleh Akaike pada tahun 1973 dengan mempertimbangkan banyaknya parameter dalam model. Kriteria AIC dituliskan pada Persamaan (2.50).

$$AIC = n \ln \hat{\sigma}_e^2 + 2n_p \quad (2.50)$$

dengan

$$\hat{\sigma}_e^2 = \sum_{t=1}^n \frac{e_t^2}{n} \quad (2.51)$$

di mana n adalah banyaknya data dan n_p adalah banyaknya parameter model.

2.15.2 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) dihitung menggunakan kesalahan absolut pada tiap periode dibagi dengan nilai observasi yang nyata untuk periode itu. Kemudian, merata-rata kesalahan persentase absolut tersebut. Pendekatan ini berguna ketika ukuran atau besar ramalan variabel itu penting dalam mengevaluasi ketetapan ramalan. MAPE mengindikasikan seberapa besar kesalahan dalam meramal yang dibandingkan dengan nilai nyata. Formulasi MAPE dituliskan pada Persamaan (2.52).

$$MAPE = \frac{1}{n} \left[\sum_{t=1}^n \frac{|e_t|}{Z_t} \times 100\% \right] = \frac{1}{n} \left[\sum_{t=1}^n \frac{|Z_t - \hat{Z}_t|}{Z_t} \times 100\% \right] \quad (2.52)$$

di mana nilai $\hat{Z}_t$ adalah nilai prediksi peramalan. Semakin kecil nilai MAPE, semakin mendekati nilai sebenarnya dengan nilai estimasi atau metode yang dipilih merupakan metode terbaik (Makridakis dkk., 1999). Interpretasi dari nilai MAPE untuk melihat seberapa akurat prediksi dapat dilihat pada Tabel 2.3.

Tabel 2.3 Interpretasi Nilai MAPE

MAPE (%)	Interpretasi
<10	Prediksi sangat akurat
10 – 20	Prediksi akurat
20 – 50	Prediksi cukup akurat
>50	Prediksi tidak akurat

Sumber : David (1982)

2.16 Harga Minyak Goreng

Minyak goreng adalah bahan pangan dengan komposisi utama trigliserida yang berasal dari bahan nabati, dengan atau tanpa perubahan kimiawi, termasuk pendinginan dan telah melalui proses rafinasi atau pemurnian yang digunakan untuk menggoreng (BSN, 2013). Menurut Sartika (2009), jenis minyak yang

umumnya dipakai untuk menggoreng adalah minyak nabati seperti minyak sawit, minyak kacang tanah, minyak wijen dan sebagainya. Minyak goreng jenis ini mengandung sekitar 80% asam lemak tak jenuh jenis asam oleat dan linoleate.

Menurut Kemendag (2021), di Indonesia minyak goreng yang paling sering digunakan adalah minyak goreng sawit. Kondisi ini disebabkan karena Indonesia merupakan negara penghasil sawit. Saat ini konsumsi minyak goreng sawit meningkat, baik untuk kebutuhan dalam negeri maupun ekspor karena tumbuhnya industri jasa boga dan perubahan gaya hidup masyarakat yang didukung oleh perbaikan tingkat ekonomi.

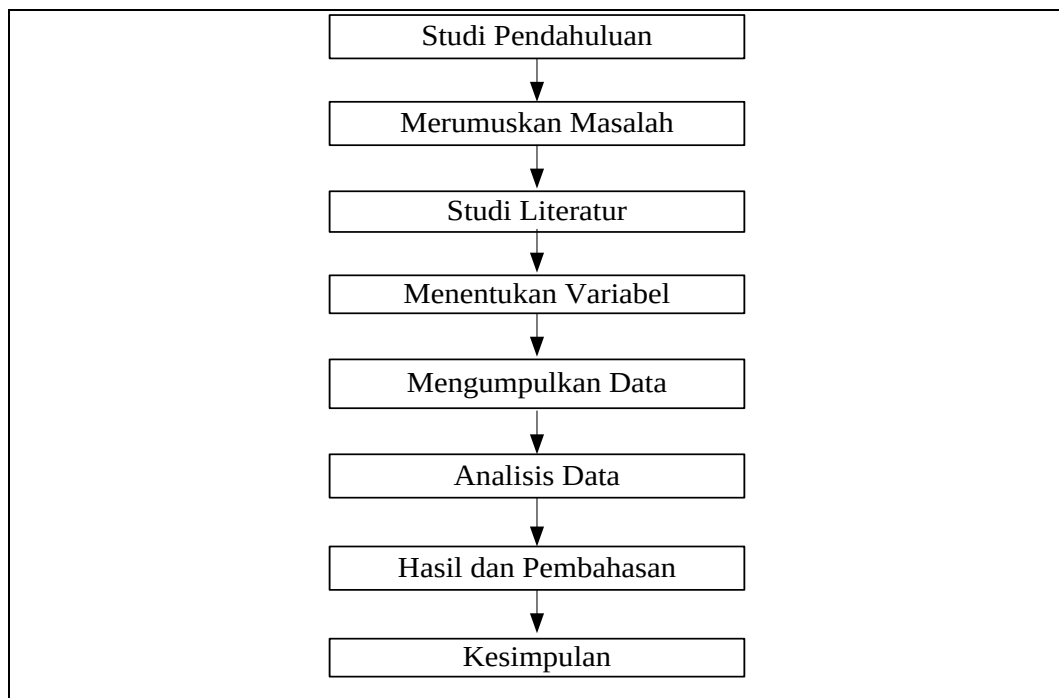
Peraturan Menteri Perdagangan Republik Indonesia No. 22 menjelaskan bahwa Harga Eceran Tertinggi (HET) dari minyak goreng berada pada rentang Rp 11.000 hingga Rp 13.000 per liter. Saat ini minyak goreng menjadi perhatian serius baik dari masyarakat maupun pelaku pasar. Minyak goreng menjadi salah satu komoditas pokok pangan yang strategis di Indonesia. Tingginya konsumsi minyak goreng sehingga menyebabkan peningkatan permintaan bahan pokok tersebut setiap tahunnya. Menurut Irnawati (2021), ketika permintaan (*demand*) suatu barang meningkat namun tidak diikuti dengan *supply* yang meningkat maka akan menyebabkan ketidakseimbangan dan akan berdampak pada kenaikan harga barang. Terbatasnya persediaan pasar nasional dan meningkatnya permintaan dari pasar internasional menyebabkan harga minyak goreng melambung tinggi. Pusat Informasi Harga Pangan Strategis Nasional (PIHPS) menyajikan data per 26 Agustus 2022, harga minyak goreng untuk pasar tradisional di Indonesia masih tergolong tinggi hingga mencapai Rp 20.200 per liter. Ketidakstabilan harga minyak goreng tersebut menjadi suatu hal yang menarik bagi peneliti untuk mengkaji lebih lanjut terkait permasalahan harga minyak goreng.

BAB 3

METODOLOGI PENELITIAN

3.1 Rancangan Penelitian

Secara umum penelitian ini terdiri dari rancangan yang disajikan pada Gambar 3.1.



Gambar 3.1 Rancangan penelitian

Rancangan penelitian yang digunakan adalah rancangan penelitian kausal komparatif yang bersifat *ex post facto*, yang artinya data dikumpulkan setelah semua kejadian yang dipersoalkan berlangsung. Penelitian ini dilakukan dengan mengumpulkan data bulanan harga minyak goreng di Indonesia dari bulan Oktober 2017 sampai dengan Oktober 2022 sebagai objek penelitian.

3.2 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan mulai pada bulan November 2022 sampai dengan Februari 2023. Pengolahan data dilakukan di Laboratorium Statistika Ekonomi dan Bisnis, Gedung Prof. Sudrajat S. U., Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Mulawarman.

3.3 Populasi dan Sampel Penelitian

Populasi dalam penelitian ini adalah harga minyak goreng di Indonesia yang tersedia pada situs resmi Pusat Informasi Harga Pangan Nasional. Sampel yang digunakan dalam penelitian ini adalah harga minyak goreng di Indonesia dari bulan Oktober 2017 sampai dengan Oktober 2022 dengan satuan rupiah.

3.4 Teknik Sampling

Teknik *sampling* pada penelitian ini menggunakan *purposive sampling*. Teknik *purposive sampling* didasarkan pada pertimbangan tertentu. Pertimbangan peneliti menggunakan sampel tersebut adalah ketersediaan data terbaru yang disajikan pada situs resmi Pusat Informasi Harga Pangan Nasional yaitu dari bulan Oktober 2017 sampai dengan Oktober 2022.

3.5 Variabel Penelitian dan Teknik Pengumpulan Data

Variabel yang digunakan dalam penelitian ini adalah harga minyak goreng pada 34 Provinsi di Indonesia. Teknik pengumpulan data yang digunakan dalam penelitian ini adalah data sekunder yaitu data runtun waktu bulanan harga minyak goreng di pasar tradisional 34 Provinsi di Indonesia bulan Oktober 2017 sampai dengan Oktober 2022 yang diperoleh melalui situs resmi Pusat Informasi Harga Pangan Nasional (<https://hargapangan.id/>).

3.6 Teknik Analisis Data

Teknik analisis data dalam penelitian ini menggunakan algoritma *K-Medoids* dan dilanjutkan dengan model peramalan ARIMA dengan pendekatan *time series clustering*. Pada penelitian ini menggunakan bantuan komputasi *software R*. Tahapan yang dilakukan dalam analisis data adalah sebagai berikut:

1. Eksplorasi data harga minyak goreng untuk mengetahui gambaran umum dan perkembangan data harga minyak goreng dari Bulan Oktober 2017 sampai dengan Oktober 2022 tiap provinsi yang divisualisasikan dengan membuat peta (*spatial mapping*) bergerak di Indonesia
2. Melakukan standarisasi data dengan menggunakan Persamaan (2.3)

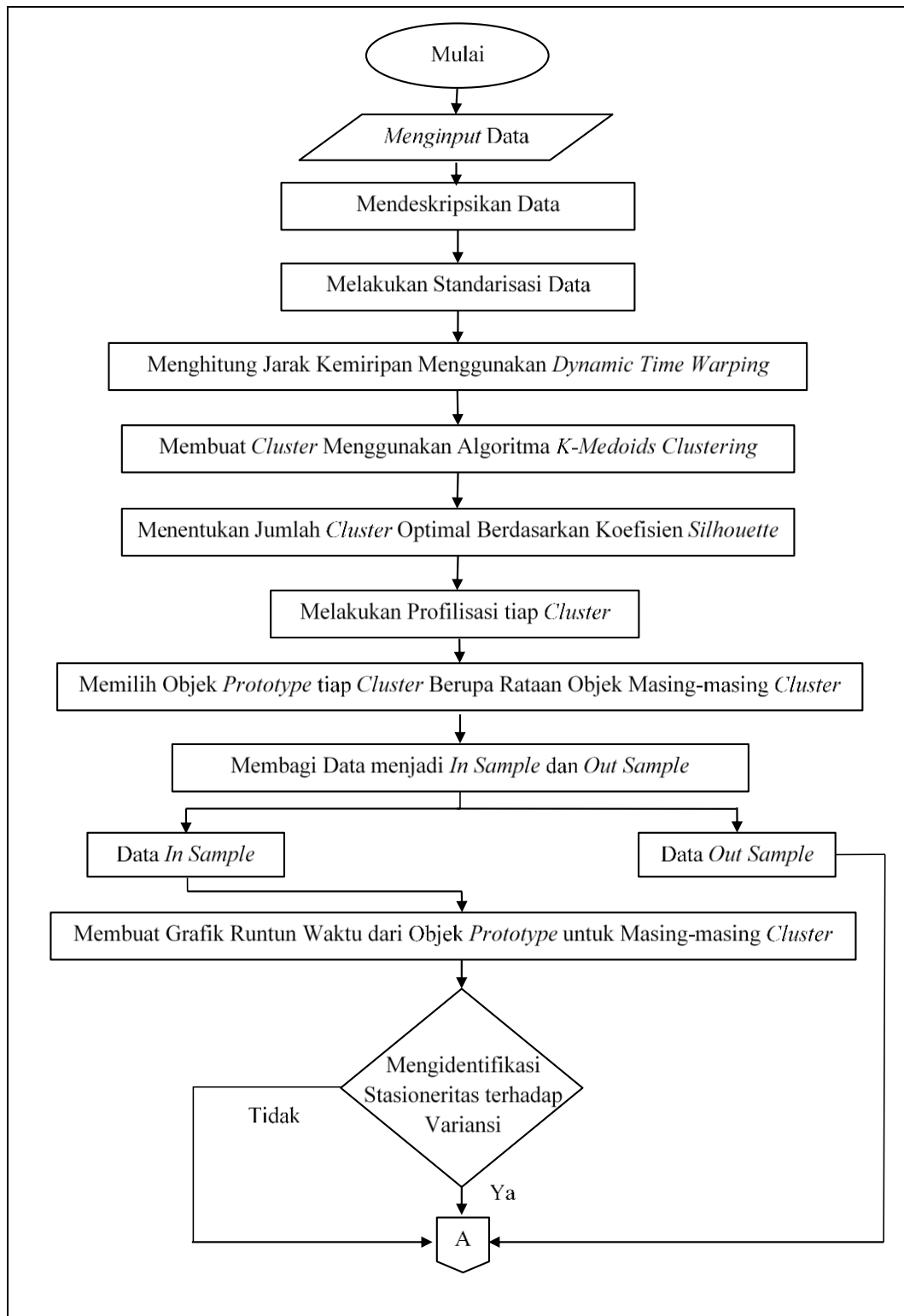
3. Melakukan pengelompokan data dengan mengimplementasikan algoritma *K-Medoids Clustering* untuk setiap $K = 2, 3, 4, \dots, 10$ dengan tahapan sebagai berikut:
 - a. Memilih sebanyak K objek sebagai *medoids*, di mana objek dalam penelitian ini adalah tiap provinsi di Indonesia.
 - b. Menghitung jarak *dynamic time warping* setiap masing-masing objek ke *medoids* yang telah dipilih berdasarkan Persamaan (2.5).
 - c. Menetapkan objek ke klaster terdekat berdasarkan jarak ke *medoids* terdekat dan menghitung fungsi objektif yang merupakan total jarak dari semua objek ke objek *medoids* terdekat
 - d. Memilih salah satu objek dari masing-masing klaster sebagai kandidat *medoids* (*non-medoids*) yang baru.
 - e. Menghitung kembali jarak *dynamic time warping* setiap masing-masing objek ke *non-medoids*.
 - f. Menetapkan setiap objek ke klaster yang sesuai berdasarkan jarak *non-medoids* terdekat dan menghitung fungsi objektif baru yang merupakan jumlah jarak dari semua objek ke objek *non-medoids* terdekat.
 - g. Menghitung simpangan (S) dengan menghitung selisih antara fungsi objektif baru dengan fungsi objektif lama. Jika nilai $S < 0$ dalam artian fungsi objektif baru memiliki total jarak yang lebih kecil, maka *medoids* awal diganti dengan *medoids* yang baru.
 - h. Mengulangi langkah (d) hingga langkah (g) sampai menghasilkan total jarak atau fungsi objektif yang minimum.
4. Menentukan jumlah K optimal menggunakan koefisien *silhouette* berdasarkan Persamaan (2.7).
5. Melakukan profilisasi klaster berdasarkan jumlah K optimal dan menginterpretasikan hasil klaster yang divisualisasikan dalam *spatial mapping* berdasarkan provinsi yang bergabung tiap klasternya.
6. Menentukan objek *prototype* yang nantinya dapat mewakili dan merepresentasikan anggotanya dalam satu klaster. Penentuan objek *prototype* dilakukan dengan menghitung rata-rata data runtun waktu dari suatu klaster.

7. Membagi data harga minyak goreng di Indonesia (baik data aktual keseluruhan objek maupun objek *prototype*) menjadi data *in-sample* dan data *out-sample* dengan proporsi pembagiannya adalah 85:15. Data *in-sample* adalah data runtun waktu dengan periode yang dimulai dari bulan Oktober 2017 hingga Januari 2022 yang digunakan sebagai pemodelan ARIMA. Untuk data *out-sample* dimulai pada bulan Februari 2022 sampai dengan Oktober 2022 yang digunakan sebagai validasi peramalan.
8. Membuat grafik runtun waktu setiap objek *prototype* dari setiap klaster yang terbentuk untuk melihat pola runtun waktu yang terbentuk, sehingga pola-pola yang terbentuk dapat menjadi pertimbangan dalam proses analisis selanjutnya.
9. Melakukan peramalan analisis runtun waktu harga minyak goreng menggunakan objek *prototype* yang telah dibangun menggunakan model ARIMA dengan tahapan sebagai berikut:
 - a. Mengidentifikasi stasioneritas data untuk setiap objek *prototype*. Pemodelan ARIMA dapat dilakukan jika stasioneritas data terpenuhi baik stasioneritas dalam variansi maupun dalam rata-rata. Pendeteksian stasioneritas dalam variansi dapat menggunakan transformasi Box-Cox berdasarkan nilai λ . Selanjutnya mengidentifikasi stasioneritas dalam rata-rata dengan membuat grafik FOK dan FOKP. Identifikasi stasioneritas dalam rata-rata juga dapat dilakukan pengujian hipotesis dengan menggunakan uji *Augmented Dickey-Fuller* (ADF) pada Persamaan (2.16). Jika data runtun waktu tidak stasioner dalam rata-rata, maka perlu dilakukan *differencing* pada data sampai data hasil *differencing* menunjukkan kondisi stasioner pada rata-rata.
 - b. Mengidentifikasi model ARIMA berdasarkan grafik FOK dan FOKP dengan pola teoritisnya dari data yang telah stasioner berdasarkan Tabel 2.2.
 - c. Mengestimasi parameter model ARIMA. Pengestimasian parameter model menggunakan *maximum likelihood estimation* dengan bantuan komputasi *software* R untuk mengurangi kompleksitas dalam perhitungan.
 - d. Melakukan pengujian signifikansi parameter. Nilai estimasi parameter yang diperoleh melalui tahap sebelumnya akan dilakukan uji signifikansi

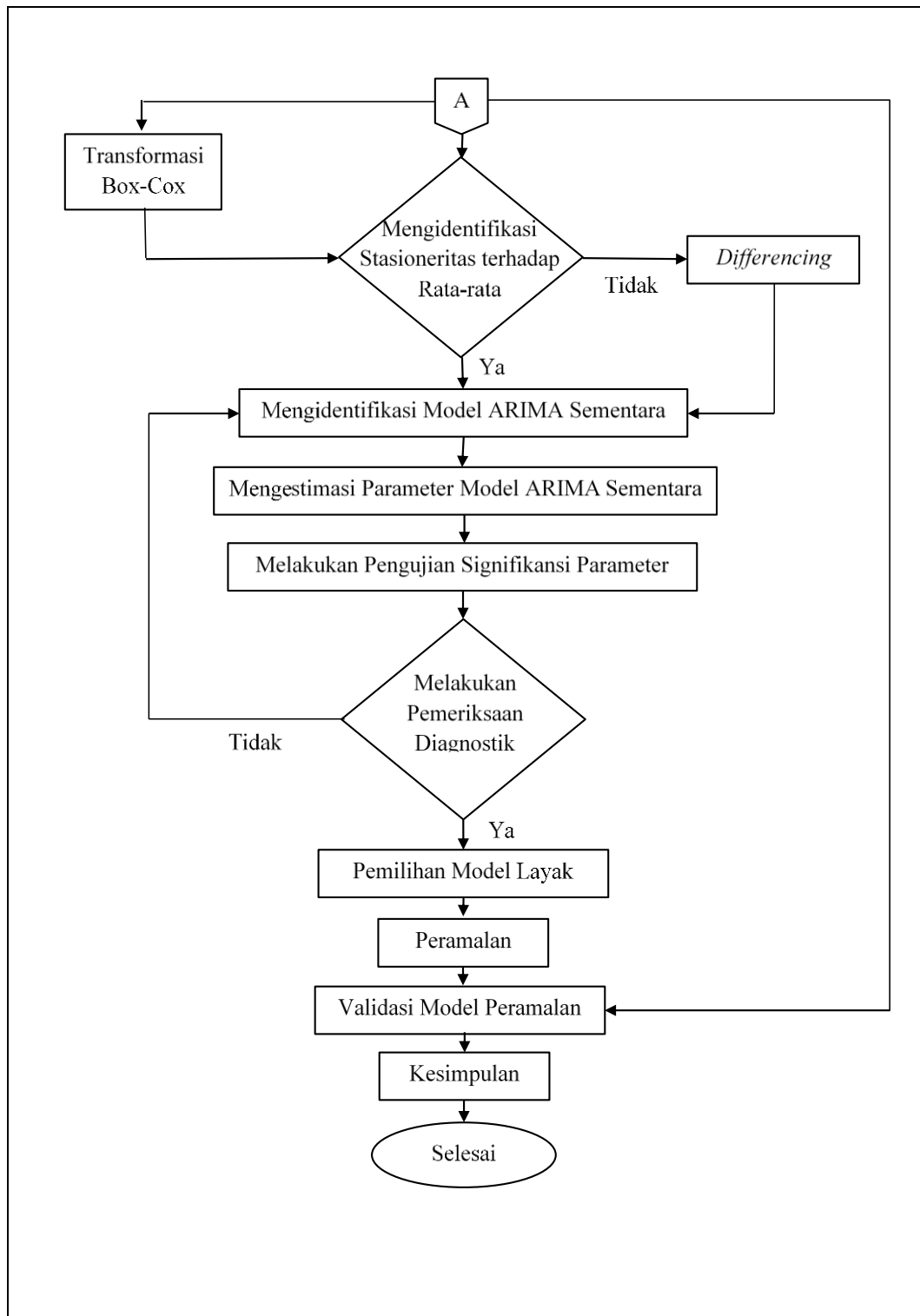
parameter menggunakan statistik uji *t*. Tujuannya adalah untuk mengetahui apakah suatu parameter model layak dimasukkan dalam model.

- e. Melakukan pemeriksaan diagnostik, proses pemeriksaan diagnostik meliputi dua pengujian, yaitu pengujian independensi *error* dan *error* berdistribusi normal. Uji independensi *error* menggunakan *Ljung-Box*, tujuannya adalah untuk mengetahui korelasi antar *error*. Uji normalitas *error* menggunakan Kolmogorov Smirnov, tujuannya adalah untuk mengetahui apakah *error* berdistribusi normal.
- f. Pemilihan model ARIMA yang layak, jika model ARIMA yang parameter-nya signifikan dan lolos tahap pemeriksaan diagnostik, maka model ARIMA tersebut layak digunakan sebagai model peramalan.
- g. Melakukan peramalan untuk periode ke depan, di mana peramalan dibagi menjadi 2 kategori yaitu peramalan tingkat klaster dan peramalan tingkat individu. Peramalan tingkat klaster yaitu meramalkan objek *prototype* setiap klaster menggunakan model ARIMA layak yang terbentuk. Kemudian peramalan tingkat individu yaitu meramalkan semua objek yang bergabung di suatu klaster menggunakan model ARIMA yang mempunyai nilai akurasi peramalan yang minimum dari *prototype* setiap klaster yang terbentuk. Peramalan dilakukan sebanyak 15 periode (9 periode sebagai validasi peramalan untuk data *out-sample* dan 6 periode untuk peramalan ke depan) Model ARIMA dari setiap klaster yang terbentuk dan memiliki nilai akurasi peramalan minimum adalah model yang nantinya digunakan untuk kesimpulan dalam model peramalan.
- h. Pengukuran ketepatan hasil peramalan, ukuran ketepatan hasil ramalan menggunakan AIC dan MAPE pada data *in-sample* dan hanya menggunakan MAPE untuk data *out-sample* berdasarkan Persamaan (2.50) dan Persamaan (2.52).

Selengkapnya tahapan analisis data peramalan harga minyak goreng di Indonesia dengan menggunakan pendekatan *K-Medoids time series clustering* disajikan pada Gambar 3.2.



Gambar 3.2 Tahapan analisis data



Gambar 3.2 Tahapan analisis data (lanjutan)

BAB 4

HASIL DAN PEMBAHASAN

4.1 Data Harga Minyak Goreng

Data harga minyak goreng di pasar tradisional pada 34 Provinsi di Indonesia dalam satuan rupiah per liter dari bulan Oktober 2017 sampai dengan bulan Oktober 2022 disajikan pada Tabel 4.1.

Tabel 4.1 Data Harga Minyak Goreng pada 34 Provinsi di Indonesia

Tahun	Bulan	t	Unit Observasi (Provinsi)				
			$Z(1)_t$	$Z(2)_t$	$Z(3)_t$	...	$Z(34)_t$
2017	Oktober	1	14.300	13.500	13.500	...	16.450
	November	2	14.450	13.450	13.350	...	16.350
	Desember	3	14.450	13.500	13.200	...	16.400
2018	Januari	4	14.400	13.400	13.150	...	16.450
	Februari	5	14.350	13.250	12.950	...	16.250
	Maret	6	14.350	13.250	12.850	...	16.250
⋮		⋮	⋮	⋮	⋮	⋮	⋮
2022	Oktober	61	18.750	18.600	17.700	...	27.900

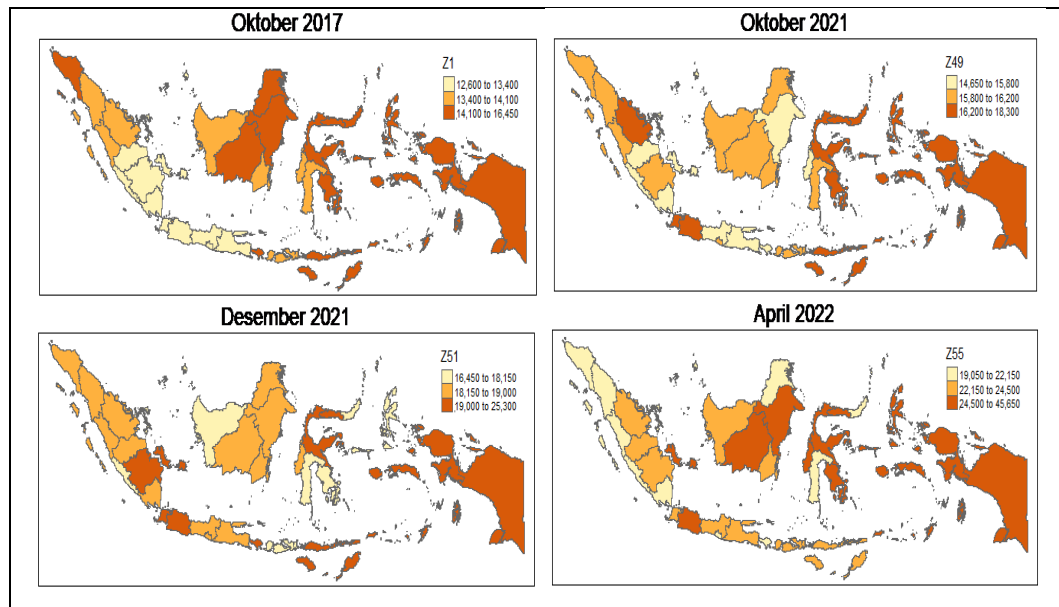
Sumber : Pusat Informasi Harga Pangan Strategis Nasional

Data selengkapnya untuk harga minyak goreng pada 34 Provinsi di Indonesia dapat dilihat pada Lampiran 1.

4.2 Statistika Deskriptif

Sebelum melakukan analisis lebih lanjut, pembahasan akan diawali dengan mengeksplorasi data penelitian sebagai gambaran awal dan mengetahui karakteristik dari data tersebut yang akan dilakukan peramalan dengan pendekatan *K-medoids time series clustering*. Eksplorasi data dilakukan dengan memvisualisasikan data menggunakan *spatial mapping* bergerak. Tujuannya adalah melihat perkembangan harga minyak goreng dari bulan ke bulan di setiap Provinsi. Penggunaan *spatial mapping* bergerak akan mengetahui perubahan-perubahan yang terjadi setiap bulannya pada harga minyak goreng di setiap

Provinsi. Sebagai contoh untuk *spatial mapping* untuk bulan Oktober 2017 dan Oktober 2022 ditampilkan pada Gambar 4.1



Gambar 4.1 (a) *Spatial mapping* Oktober 2017; (b) *Spatial mapping* Oktober 2022

Sedangkan untuk *spatial mapping* bergerak dari Bulan Oktober 2017 sampai dengan Oktober 2022 dapat diakses melalui *barcode* yang ditampilkan pada Gambar 4.2 berikut dengan cara melakukan *scanning* pada *barcode*.



Gambar 4.2 *Barcode* untuk mengakses *spatial mapping* bergerak

Berdasarkan *spatial mapping* yang ditampilkan pada Gambar 4.1 dapat dilihat pergerakan harga minyak goreng tertinggi sebagian besar terletak pada wilayah Indonesia bagian Timur, sedangkan untuk wilayah Indonesia bagian Barat dan Tengah sebagian besar cenderung memiliki harga minyak goreng yang lebih

rendah dibandingkan dengan Indonesia bagian Timur. Seiring berjalannya waktu, sampai dengan tahun 2022 seluruh Provinsi di Indonesia perlahan-lahan mengalami kenaikan harga minyak goreng. Hal ini dapat dilihat pada warna setiap wilayah yang semakin gelap dan rentang yang ditampilkan semakin tinggi. Berdasarkan data yang disajikan oleh PIHPS, bulan April 2022 merupakan puncak harga tertinggi minyak goreng dengan rentang tertingginya mencapai Rp 24.500 sampai dengan Rp 45.650.

4.3 *K-Medoids Time Series Clustering*

Metode pengelompokan runtun waktu merupakan pendekatan yang digunakan dalam *data mining* untuk mengeksplorasi data runtun waktu. Tahapan dalam proses pengelompokan runtun waktu dengan menggunakan *K-Medoids time series clustering* adalah sebagai berikut :

4.3.1 Analisis Klaster

Proses dari pengelompokan runtun waktu data harga minyak goreng adalah sebagai berikut :

1. Standarisasi data

Data yang digunakan dalam penelitian ini adalah data bulanan harga minyak goreng di pasar tradisional di Indonesia yang terdiri dari 34 Provinsi di Indonesia berdasarkan Tabel 4.1. Standarisasi data bertujuan untuk membuat semua unit observasi berada pada rentang yang sama dan memperkecil perbedaan nilai antar unit observasi. Sebagai contoh akan dilakukan perhitungan standarisasi untuk $Z(1)_t$ yaitu data harga minyak goreng Provinsi Aceh berdasarkan data yang disajikan pada Lampiran 1.

Terdapat tiga langkah dalam menghitung standarisasi data, langkah pertama yaitu menghitung rata-rata dari data harga minyak goreng Provinsi Aceh ($Z(1)_t$) berdasarkan Persamaan (2.1), sehingga diperoleh

$$\bar{Z}(1)_t = \frac{1}{61} \sum_{t=1}^{61} (14.300 + 14.550 + 14.450 + \dots + 18.750)$$

$$= \frac{1}{61} (921.550) = 15.107,38$$

Langkah selanjutnya yaitu menghitung deviasi standar dari data harga minyak goreng Provinsi Aceh ($Z(1)_t$) berdasarkan Persamaan (2.2), sehingga diperoleh

$$\begin{aligned} S_{Z(1)_t} &= \sqrt{\left(\frac{1}{61-1}\right) \sum_{t=1}^{61} (14.300 - 15.107,38)^2 + \dots + (18.750 - 15.107,38)^2} \\ &= \sqrt{\left(\frac{1}{60}\right) (391.064.180,3)} = 2.306,02 \end{aligned}$$

Setelah nilai rata-rata dan deviasi standar dari data harga minyak goreng Provinsi Aceh ($Z(1)_t$) telah diketahui, selanjutnya adalah melakukan standarisasi data dengan menggunakan Persamaan (2.3), sehingga diperoleh

Untuk $t = 1$,

$$\tilde{Z}(1)_1 = \frac{14.300 - 15.107,38}{2.306,02} = -0,35$$

Untuk $t = 2$,

$$\tilde{Z}(1)_2 = \frac{14.550 - 15.107,38}{2.306,02} = -0,24$$

Untuk $t = 3$,

$$\tilde{Z}(1)_3 = \frac{14.450 - 15.107,38}{2.306,02} = -0,28$$

dengan cara yang sama perhitungan dilakukan untuk $t = 4$ sampai dengan $t = 61$, sehingga diperoleh

$$\tilde{Z}(1)_{61} = \frac{18.750 - 15.107,38}{2.306,02} = 1,57$$

Standarisasi data untuk unit observasi lainnya dapat dihitung dengan formula perhitungan yang sama dengan $Z(1)_t$. Data harga minyak goreng 34 Provinsi di Indonesia yang telah distandarisasi disajikan pada Tabel 4.2.

Tabel 4.2 Standarisasi Data

t	$\tilde{Z}(1)_t$	$\tilde{Z}(2)_t$	$\tilde{Z}(3)_t$	$\tilde{Z}(4)_t$	$\tilde{Z}(5)_t$	$\tilde{Z}(6)_t$	$\dots$	$\tilde{Z}(34)_t$
1	-0,35	-0,37	-0,25	-0,29	-0,52	-0,31	$\dots$	-0,35
2	-0,24	-0,38	-0,30	-0,29	-0,46	-0,28	$\dots$	-0,37
3	-0,28	-0,37	-0,36	-0,33	-0,46	-0,26	$\dots$	-0,36
4	-0,30	-0,40	-0,38	-0,36	-0,46	-0,48	$\dots$	-0,35
5	-0,32	-0,45	-0,45	-0,45	-0,46	-0,48	$\dots$	-0,40
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
61	1,57	1,38	1,29	0,83	0,62	1,04	$\dots$	2,00

Sumber: Lampiran 3

Hasil standarisasi data berdasarkan Tabel 4.2 kemudian disusun dalam bentuk matriks yang disebut dengan matriks **O** berukuran 61×34 untuk memudahkan dalam langkah penelitian selanjutnya.

$$\mathbf{O} = \begin{matrix} & \begin{matrix} \tilde{Z}(1)_t & \tilde{Z}(2)_t & \tilde{Z}(3)_t & \tilde{Z}(4)_t & \tilde{Z}(5)_t & \tilde{Z}(6)_t & \dots & \tilde{Z}(34)_t \end{matrix} \\ \begin{matrix} -0,35 & -0,37 & -0,25 & -0,29 & -0,52 & -0,31 & \dots & -0,35 \\ -0,24 & -0,38 & -0,30 & -0,29 & -0,46 & -0,28 & \dots & -0,37 \\ -0,28 & -0,37 & -0,36 & -0,33 & -0,46 & -0,26 & \dots & -0,36 \\ -0,30 & -0,40 & -0,38 & -0,36 & -0,46 & -0,48 & \dots & -0,35 \\ -0,32 & -0,45 & -0,45 & -0,45 & -0,46 & -0,48 & \dots & -0,40 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1,57 & 1,38 & 1,29 & 0,83 & 0,62 & 1,04 & \dots & 2,00 \end{matrix} \end{matrix}$$

2. Pengelompokan menggunakan algoritma *K-Medoids*

Algoritma *K-Medoids* merupakan salah satu prosedur pengelompokan dengan meminimalkan jumlah jarak antara setiap objek dengan objek *medoids* terdekat berdasarkan Persamaan (2.6). Objek pada penelitian ini yaitu 34 Provinsi di Indonesia. Penelitian ini menggunakan jumlah kluster $K = 2, 3, 4, \dots, 10$ yang kemudian akan ditentukan K optimal dalam mengelompokkan data harga minyak goreng. Sebagai contoh berikut merupakan proses pengelompokan data harga minyak goreng menjadi 2 kluster.

Langkah pertama yaitu memilih 2 objek pada 34 Provinsi di Indonesia sebagai objek *medoids*. Proses pemilihan objek *medoids* bersifat acak. Provinsi

Aceh ($\tilde{Z}(1)_t$) dan Provinsi Sulawesi Tengah ($\tilde{Z}(28)_t$) dipilih sebagai *medoids* pada iterasi ke-1. Informasi dari kedua *medoids* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Informasi *Medoids* Iterasi ke-1

t	1	2	3	...	61
Aceh ($\tilde{Z}(1)_t$)	-0,35	-0,24	-0,28	...	1,57
Sulawesi Tengah ($\tilde{Z}(28)_t$)	-0,46	-0,52	-0,47	...	1,34

Kemudian menghitung nilai jarak setiap objek dengan *medoids* awal menggunakan jarak *dynamic time warping* (DTW). Perhitungan jarak DTW dilakukan untuk *medoids* pertama dengan semua Provinsi kemudian dilanjutkan dengan perhitungan untuk *medoids* kedua dengan semua Provinsi. Sebagai contoh berikut merupakan perhitungan mencari nilai jarak DTW untuk Provinsi Sumatera Utara ($\tilde{Z}(2)_t$) terhadap objek *medoids* pertama yaitu Provinsi Aceh ($\tilde{Z}(1)_t$). Langkah perhitungan jarak DTW dimulai dengan menyusun data hasil standarisasi $\tilde{Z}(1)_t$ menjadi bentuk baris dimulai dari kiri ke kanan, sedangkan untuk $\tilde{Z}(2)_t$ menjadi bentuk kolom dimulai dari bawah ke atas. Kemudian membuat matriks $\mathbf{C}$ berukuran 61×61 dengan setiap elemen matriksnya dapat dihitung menggunakan Persamaan (2.4) dan Persamaan (2.5) sebagai berikut:

$$c_{ij} = w_{ij} + \min \begin{cases} c_{(i-1)(j-1)} \\ c_{(i-1)j} \\ c_{i(j-1)} \end{cases} ; i, j = 1, 2, 3, \dots, 61$$

sehingga,

$$c_{11} = w_{11} + \min \begin{cases} c_{(1-1)(1-1)} = 0 \\ c_{(1-1)1} = 0 \\ c_{1(1-1)} = 0 \end{cases}$$

dengan $w_{11} = |-0,35 - (-0,37)| = 0,02$, sehingga diperoleh elemen matriks c_{11} adalah 0,02, kemudian

$$c_{21} = w_{21} + \min \begin{cases} c_{(2-1)(1-1)} = 0 \\ c_{(2-1)1} = 0,02 \\ c_{2(1-1)} = 0 \end{cases}$$

dengan $w_{21} = |-0,24 - (-0,37)| = 0,13$, sehingga diperoleh elemen matriks c_{21} adalah 0,13, kemudian

$$c_{31} = w_{31} + \min \begin{cases} c_{(3-1)(1-1)} = 0 \\ c_{(3-1)1} = 0,12 \\ c_{3(1-1)} = 0 \end{cases}$$

dengan $w_{31} = |-0,28 - (-0,37)| = 0,09$, sehingga diperoleh elemen matriks c_{31} adalah 0,09. Perhitungan elemen matriks seterusnya dilakukan dengan cara yang sama sampai dengan perhitungan elemen matriks c_{6161} . Matriks $\mathbf{C}$ berukuran 61×61 dengan setiap elemen matriksnya diperoleh sebagai berikut:

$\tilde{Z}(2)_t$									
↑ 1,38	1,73	3,37	5,04	6,74	8,46	10,17	...	4,46	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	
-0,45	0,11	0,27	0,36	0,42	0,35	0,35	...	9,89	
-0,40	0,06	0,19	0,27	0,31	0,22	0,22	...	7,85	
-0,37	0,02	0,15	0,24	0,21	0,15	0,15	...	5,87	
-0,38	0,04	0,17	0,19	0,15	0,10	0,10	...	3,91	
-0,37	0,02	0,13	0,09	0,06	0,04	0,04	...	1,95	
	-0,35	-0,24	-0,28	-0,30	-0,32	-0,32	...	1,57	$\tilde{Z}(1)_t$

Sumber: Lampiran 4

Selanjutnya adalah mencari sekumpulan dari semua *warping path* yang mungkin berdasarkan matriks $\mathbf{C}$ dimulai dari elemen matriks kiri bawah atau yang disebut dengan matriks c_{11} sampai dengan elemen matriks kanan atas atau yang disebut dengan matriks c_{6161} . Adapun *warping path* yang mungkin dari matriks $\mathbf{C}$ adalah sebagai berikut:

$$\begin{aligned}
P_1 &= \{c_{11}, c_{21}, c_{31}, c_{41}, c_{51}, c_{61}, \dots, c_{6161}\} \\
P_2 &= \{c_{11}, c_{21}, c_{31}, c_{32}, c_{33}, c_{34}, \dots, c_{6161}\} \\
P_3 &= \{c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, \dots, c_{6161}\} \\
&\vdots \\
P_k &= \{c_{11}, \dots, \dots, \dots, \dots, \dots, \dots, \dots, c_{6161}\}
\end{aligned}$$

Ketika semua *warping path* yang mungkin telah terbentuk, selanjutnya menghitung jarak DTW menggunakan Persamaan (2.5). Jarak DTW yang digunakan adalah jarak yang memiliki *warping path* yang minimum dari akar kuadrat hasil penjumlahan setiap elemen ke- (i,j) . Proses DTW menggunakan teknik pemrograman yaitu *software* R untuk menemukan *warping path* yang menghasilkan jarak minimum untuk $\tilde{Z}(2)_t$ terhadap $\tilde{Z}(1)_t$. Jarak DTW tersebut yang nantinya akan menjadi jarak antar kedua objek yang digunakan. Berdasarkan *software* R pada Lampiran 4 diperoleh jarak DTW untuk $\tilde{Z}(2)_t$ terhadap $\tilde{Z}(1)_t$ adalah sebesar 6,58. Perhitungan jarak DTW untuk unit observasi lainnya dilakukan dengan cara yang sama, hingga diperoleh jarak DTW untuk $\tilde{Z}(34)_t$ terhadap $\tilde{Z}(1)_t$ adalah sebesar 13,26.

Setelah menghitung jarak DTW untuk semua objek terhadap *medoids* berdasarkan Persamaan (2.5), selanjutnya adalah mengalokasikan setiap objek ke *medoids* terdekatnya dan menghitung fungsi objektif yang merupakan total jarak objek ke *medoids* terdekat, sehingga hasil nilai jarak serta informasi kluster pada iterasi pertama disajikan pada Tabel 4.4.

Tabel 4.4 Hasil Perhitungan Jarak DTW Iterasi ke-1

Provinsi	Jarak ke <i>medoids</i>		Terdekat	Kluster
	K1	K2		
Aceh	0,00	13,87	0,00	1
Sumatera Utara	6,59	12,02	6,59	1
Sumatera Barat	11,73	10,39	10,39	2
Riau	9,29	9,70	9,29	1
Kepulauan Riau	16,57	10,58	10,58	2
Jambi	12,49	9,31	9,31	2
Bengkulu	14,63	12,64	12,64	2
Sumatera Selatan	13,14	9,76	9,76	2

Tabel 4.4 Hasil Perhitungan Jarak DTW Iterasi ke-1 (lanjutan)

Provinsi	Jarak ke <i>medoids</i>		Terdekat	Klaster
	K1	K2		
Kepulauan Bangka Belitung	15,41	10,01	10,01	2
Lampung	6,97	13,18	6,97	1
Banten	9,23	9,65	9,23	1
Jawa Barat	12,09	8,34	8,34	2
DKI Jakarta	9,02	11,34	9,02	1
Jawa Tengah	11,08	8,90	8,90	2
DI Yogyakarta	9,58	11,87	9,58	1
Jawa Timur	11,24	7,44	7,44	2
Bali	6,97	10,46	6,97	1
Nusa Tenggara Barat	11,59	8,55	8,55	2
Nusa Tenggara Timur	13,36	10,37	10,37	2
Kalimantan Barat	10,20	7,08	7,08	2
Kalimantan Selatan	6,85	9,52	6,85	1
Kalimantan Tengah	11,48	9,57	9,57	2
Kalimantan Timur	14,16	9,30	9,30	2
Kalimantan Utara	11,69	9,66	9,66	2
Gorontalo	12,28	10,18	10,18	2
Sulawesi Selatan	8,14	11,20	8,14	1
Sulawesi Tenggara	20,27	14,30	14,30	2
Sulawesi Tengah	13,87	0,00	0,00	2
Sulawesi Utara	11,18	11,22	11,18	1
Sulawesi Barat	13,21	8,19	8,19	2
Maluku	12,67	8,51	8,51	2
Maluku Utara	14,38	14,18	14,18	2
Papua	11,55	11,07	11,07	2
Papua Barat	13,26	12,46	12,46	2
Fungsi Objektif			304,62	

Sumber: Lampiran 5

Berdasarkan Tabel 4.4, sebagai contoh untuk Kepulauan Bangka Belitung memperoleh jarak DTW ke *medoids* pertama sebesar 15,41 dan ke *medoids* kedua sebesar 10,01, sehingga Kepulauan Bangka Belitung bergabung ke klaster 2 dikarenakan memiliki jarak lebih kecil ke *medoids* kedua dibandingkan ke *medoids* pertama. Setelah memperoleh hasil jarak dari setiap objek pada iterasi ke-1, maka selanjutnya memilih salah satu objek dari masing-masing klaster sebagai kandidat *medoids* yang baru. Kandidat *medoids* baru untuk iterasi ke-2 disajikan pada Tabel 4.5.

Tabel 4.5 Informasi Kandidat *Medoids* Baru Iterasi ke-2

t	1	2	3	...	61
DKI Jakarta ($\tilde{Z}(13)_t$)	-0,51	-0,56	-0,59	...	1,24
Nusa Tenggara Barat ($\tilde{Z}(18)_t$)	-0,57	-0,57	-0,59	...	1,33

Kemudian menghitung kembali nilai jarak terdekat dengan kandidat *medoids* baru (*non-medoids*) menggunakan jarak DTW dengan proses yang sama seperti pada iterasi ke-1. Setelah diperoleh jarak DTW untuk semua objek terhadap kandidat *medoids* baru, selanjutnya dihitung kembali fungsi objektif yang baru dan hasil nilai jarak DTW untuk semua objek ke kandidat *medoids* baru pada iterasi ke-2 dapat dilihat pada Tabel 4.6.

Tabel 4.6 Hasil Perhitungan Jarak DTW Iterasi ke-2

Provinsi	Jarak ke <i>medoids</i>		Terdekat	Klaster
	K1	K2		
Aceh	9,02	11,59	9,02	1
Sumatera Utara	6,94	10,28	6,94	1
Sumatera Barat	8,52	10,35	8,52	1
Riau	8,86	10,14	8,86	1
Kepulauan Riau	14,07	12,30	12,30	2
Jambi	8,89	8,97	8,89	1
Bengkulu	11,22	12,01	11,22	1
Sumatera Selatan	11,02	11,60	11,02	1
Kepulauan Bangka Belitung	15,49	11,81	11,81	2
Lampung	7,52	11,11	7,52	1
Banten	6,73	9,75	6,73	1
Jawa Barat	6,99	7,92	6,99	1
DKI Jakarta	0,00	7,70	0,00	1
Jawa Tengah	5,94	7,34	5,94	1
DI Yogyakarta	5,71	9,50	5,71	1
Jawa Timur	7,65	7,88	7,65	1
Bali	7,28	8,40	7,28	1
Nusa Tenggara Barat	7,70	0,00	0,00	2
Nusa Tenggara Timur	11,17	10,02	10,02	2
Kalimantan Barat	9,48	7,91	7,91	2
Kalimantan Selatan	5,56	7,89	5,56	1
Kalimantan Tengah	11,14	7,36	7,36	2
Kalimantan Timur	13,29	9,75	9,75	2
Kalimantan Utara	8,54	5,93	5,93	2
Gorontalo	8,38	7,61	7,61	2
Sulawesi Selatan	7,83	8,34	7,83	1

Tabel 4.6 Hasil Perhitungan Jarak DTW Iterasi ke-2 (lanjutan)

Provinsi	Jarak ke <i>medoids</i>		Terdekat	Klaster
	K1	K2		
Sulawesi Tenggara	22,91	19,43	19,43	2
Sulawesi Tengah	11,34	8,55	8,55	2
Sulawesi Utara	10,24	9,13	9,13	2
Sulawesi Barat	11,26	9,19	9,19	2
Maluku	12,18	9,90	9,90	2
Maluku Utara	16,00	14,85	14,85	2
Papua	13,81	10,85	10,85	2
Papua Barat	15,95	13,41	13,41	2
Fungsi Objektif			293,66	

Sumber: Lampiran 5

Jarak antara iterasi ke-1 dan iterasi ke-2 telah diperoleh, selanjutnya menghitung selisih antara fungsi objektif baru dengan fungsi objektif lama. Berdasarkan Tabel 4.4 diperoleh fungsi objektif lama sebesar 304,62 dan berdasarkan Tabel 4.6 diperoleh fungsi objektif baru sebesar 293,66. Artinya, selisih antara fungsi objektif baru dengan fungsi objektif lama menghasilkan selisih kurang dari nol ($S < 0$), sehingga *medoids* awal diganti dengan *medoids* yang baru. Proses iterasi dilanjutkan dengan *medoids* yang baru sampai diperoleh fungsi objektif yang paling minimum. Penyelesaian algoritma *K-medoids* sampai diperoleh *medoids* yang optimal menggunakan bantuan *software* R. Berdasarkan *software* R pada Lampiran 6 diperoleh *medoids* yang optimal adalah Provinsi Banten ($\tilde{Z}(11)_i$) sebagai *medoids* klaster pertama dan Provinsi Kalimantan Timur ($\tilde{Z}(23)_i$) sebagai *medoids* klaster kedua dengan fungsi objektif sebesar 288,09.

Proses pengelompokan dilakukan dengan cara yang sama untuk $K = 3, 4, 5, \dots, 10$. Perbedaan hanya terletak pada penentuan banyaknya *medoids*. Semakin banyaknya klaster maka banyaknya *medoids* juga akan semakin besar. Hasil pengelompokan untuk $K = 3, 4, 5, \dots, 10$ dapat dilihat pada Lampiran 7 sampai dengan Lampiran 14. Pada penelitian ini telah dicoba ketika pengelompokan dengan tidak menggunakan standarisasi data dan mengakibatkan pengelompokan yang bias. Jumlah objek-objek yang bergabung pada tiap klaster

cenderung tidak seimbang dan objek-objek yang memiliki perbedaan nilai diantara objek lainnya cenderung membentuk suatu klaster sendiri.

3. Pemilihan banyak klaster optimal

Setelah memperoleh hasil pengelompokan untuk $K = 2, 3, 4, \dots, 10$ maka selanjutnya adalah menentukan nilai K optimal dalam proses pengelompokan menggunakan algoritma *K-medoids*. Evaluasi klaster yang digunakan pada penelitian ini adalah koefisien *silhouette* berdasarkan Persamaan (2.7).

Sebagai contoh akan dilakukan perhitungan koefisien *silhouette* Provinsi Aceh untuk $K = 2$. Langkah pertama adalah menghitung nilai a_1 yaitu rata-rata jarak DTW dari Provinsi Aceh dengan semua objek yang berada pada satu klaster dengan Provinsi Aceh. Berdasarkan hasil pengelompokan untuk $K = 2$ pada Lampiran 5 diperoleh terdapat 18 Provinsi lainnya yang berada pada satu klaster dengan Provinsi Aceh. Matriks jarak DTW dapat dilihat pada Lampiran 4, sehingga perhitungannya adalah sebagai berikut:

$$\begin{aligned} a_{Aceh} &= \frac{1}{18} \times [6,59 + 11,73 + 9,29 + \dots + 12,28 + 8,14] \\ &= 9,91 \end{aligned}$$

Selanjutnya menghitung nilai b_1 yaitu rata-rata jarak DTW dari Provinsi Aceh dengan semua objek yang berada pada klaster yang berbeda, sehingga perhitungannya adalah sebagai berikut:

$$\begin{aligned} b_{Aceh} &= \frac{1}{15} \times [16,57 + 14,63 + 15,41 + \dots + 11,55 + 13,26] \\ &= 13,85 \end{aligned}$$

Nilai a_i dan b_i telah diperoleh, selanjutnya adalah menghitung koefisien *silhouette* berdasarkan Persamaan (2.7).

$$\begin{aligned} SC_{Aceh} &= \frac{b_{Aceh} - a_{Aceh}}{\max(a_{Aceh}, b_{Aceh})} \\ &= \frac{13,85 - 9,91}{13,85} \\ &= 0,28 \end{aligned}$$

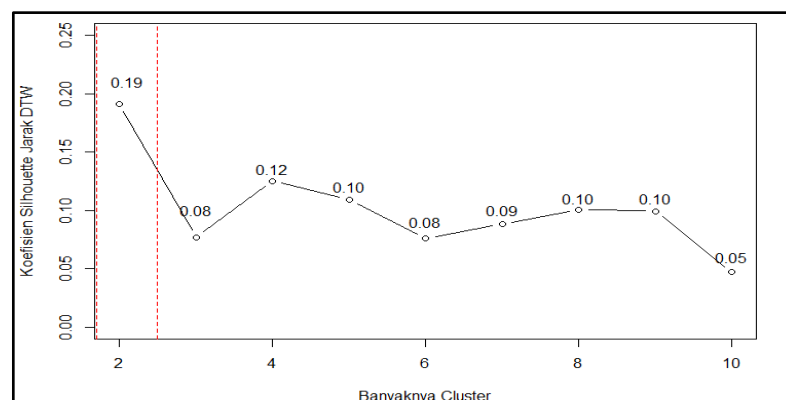
Koefisien *silhouette* untuk Provinsi lainnya dapat dihitung dengan formula yang sama dengan Provinsi Aceh. Nilai koefisien *silhouette* untuk tiap Provinsi disajikan pada Tabel 4.7.

Tabel 4.7 Hasil Perhitungan Koefisien *Silhouette* untuk $K = 2$

Provinsi	Koefisien <i>Silhouette</i>
Aceh	0,28
Sumatera Utara	0,39
Sumatera Barat	0,36
Riau	0,32
Kepulauan Riau	-0,04
⋮	⋮
Papua	0,12
Papua Barat	0,09
Rata-rata	0,19

Sumber: Lampiran 15

Koefisien *silhouette* yang digunakan adalah koefisien *silhouette* global di mana rata-rata koefisien *silhouette* tiap Provinsi, sehingga berdasarkan Tabel 4.7 nilai koefisien *silhouette* global untuk $K = 2$ adalah sebesar 0,19. Perhitungan dilakukan dengan cara yang sama untuk $K = 3, 4, 5, \dots, 10$. Nilai koefisien *silhouette* untuk semua K dalam menentukan banyak kluster yang optimal dapat dilihat pada Lampiran 15 dan ditampilkan pada Gambar 4.3.



Gambar 4.3 Grafik nilai koefisien *silhouette*

Nilai K optimal terlihat dari nilai koefisien *silhouette* terbesar. Berdasarkan Gambar 4.3, dapat dilihat untuk kasus ini nilai K yang optimal dalam mengelompokkan harga minyak goreng di Indonesia adalah 2 kluster dengan nilai koefisien *silhouette* terbesar yaitu 0,19.

4. Pengelompokan dan interpretasi hasil klaster

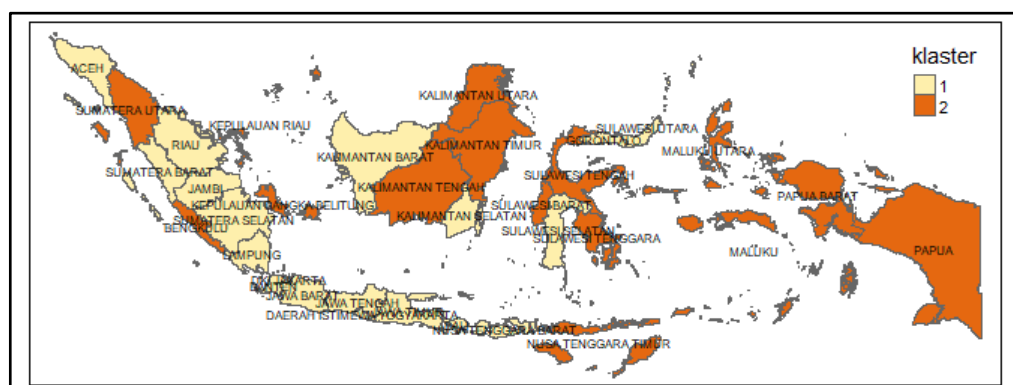
Setelah memperoleh pengelompokan yang optimal, langkah selanjutnya yaitu melakukan pengelompokan dan interpretasi hasil klaster berdasarkan banyak klaster yang optimal. Pada proses peramalan harga minyak goreng di Indonesia menggunakan pendekatan *K-medoids time series clustering* dengan jarak yang digunakan adalah DTW dengan jumlah klaster optimal sebanyak 2 klaster, sehingga diperoleh hasil pengelompokan yang dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil Klaster

Klaster	Provinsi	Jumlah
1	Aceh, Sumatera Utara, Sumatera Barat, Riau, Jambi, Sumatera Selatan, Lampung, Banten, Jawa Barat, DKI Jakarta, Jawa Tengah, DI Yogyakarta, Jawa Timur, Bali, Nusa Tenggara Barat, Kalimantan Barat, Kalimantan Selatan, Gorontalo, Sulawesi Selatan	19
2	Kepulauan Riau, Bengkulu, Kepulauan Bangka Belitung, Nusa Tenggara Timur, Kalimantan Tengah, Kalimantan Timur, Kalimantan Utara, Sulawesi Tenggara, Sulawesi Tengah, Sulawesi Utara, Sulawesi Barat, Maluku, Maluku Utara, Papua, Papua Barat	15
Total		34

Sumber: Lampiran 6

Visualisasi dari hasil klaster pada Tabel 4.8 dapat dilihat pada *spatial mapping* yang ditampilkan pada Gambar 4.4.



Gambar 4.4 *Spatial mapping* hasil klaster

Berdasarkan Gambar 4.4 dapat diketahui hasil pengelompokan harga minyak goreng pada 34 Provinsi di Indonesia. Pada klaster 1 terdapat 19 Provinsi yang bergabung dan dapat dilihat pada wilayah yang memiliki warna lebih terang.

Klaster 1 adalah klaster dengan harga minyak goreng dibawah klaster 2. Selanjutnya pada klaster 2 terdapat 15 Provinsi yang bergabung dan dapat dilihat pada wilayah yang memiliki warna lebih gelap. Klaster 2 adalah klaster dengan harga minyak goreng tertinggi. Visualisasi *spatial mapping* dari hasil klaster lebih jelasnya dapat dilihat pada Lampiran 17.

4.3.2 Pemodelan ARIMA

Hasil pengelompokan dari data harga minyak goreng di Indonesia telah diperoleh. Berdasarkan Gambar 4.4 diperoleh hasil pengelompokan sebanyak 2 klaster. Selanjutnya adalah menentukan objek *prototype* yang nantinya dapat mewakili dan merepresentasikan anggotanya dalam satu klaster. Pada penelitian ini akan terbentuk 2 objek *prototype* dikarenakan banyaknya klaster yang terbentuk sebanyak 2 klaster. Penentuan objek *prototype* dilakukan dengan menghitung rata-rata runtun waktu dari suatu klaster. Berdasarkan profilisasi klaster pada Tabel 4.8, maka perhitungan untuk memperoleh objek *prototype* untuk setiap klaster dapat dilihat pada Tabel 4.9 sebagai berikut:

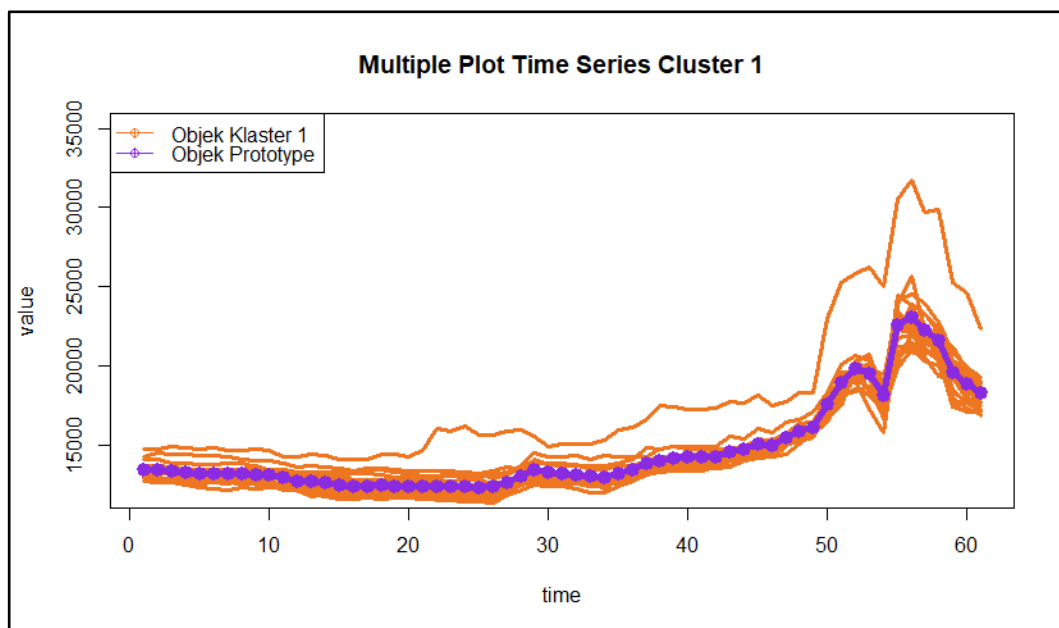
Tabel 4.9 Informasi Objek *Prototype*

Klaster	Objek	<i>t</i>				
		1	2	3	...	61
1	$Z(1)_t$	14.300	14.550	14.450	...	18.750
	$Z(2)_t$	13.500	13.450	13.500	...	18.600
	$Z(3)_t$	13.500	13.350	13.200	...	17.700
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$Z(26)_t$	13.400	13.400	13.400	...	18.850
Prototype	$Y(1)_t$	13.476	13.455	13.418	...	18.334
2	$Z(5)_t$	12.600	12.750	12.750	...	15.400
	$Z(7)_t$	12.600	12.850	12.800	...	17.500
	$Z(9)_t$	12.850	12.800	12.800	...	16.350
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$Z(34)_t$	16.450	16.350	16.400	...	27.900
Prototype	$Y(2)_t$	14.600	14.513	14.530	...	20.747

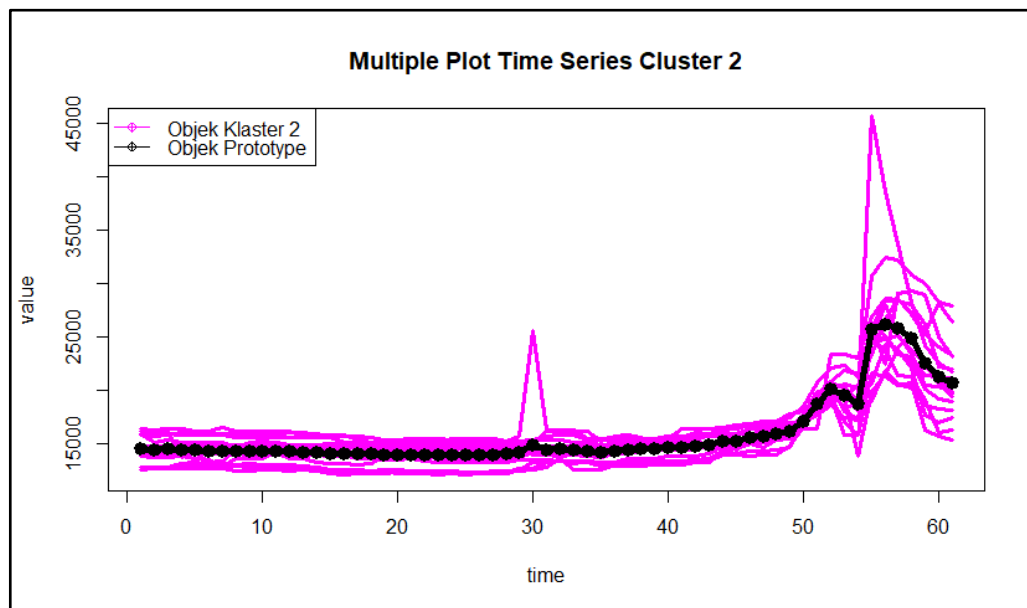
Sumber: Lampiran 18

Berdasarkan Tabel 4.9, objek *prototype* yang terbentuk dari klaster 1 adalah rata-rata dari semua objek runtun waktu yang berada pada klaster 1 yang dapat dimisalkan dengan notasi $Y(1)_t$, sedangkan untuk objek *prototype* yang terbentuk dari klaster 2 adalah rata-rata dari semua objek runtun waktu yang berada pada klaster 2 yang dapat dimisalkan dengan notasi $Y(2)_t$.

Kedua objek *prototype* inilah yang nantinya akan dilakukan pemodelan ARIMA untuk mewakili seluruh anggota dari masing-masing klaster, sehingga nantinya anggota dari setiap klaster hanya akan menggunakan model ARIMA dari objek *prototype* untuk melakukan proses peramalan. Dengan demikian konsep dari peramalan dengan pendekatan *time series clustering* akan bermanfaat karena tidak harus memodelkan ARIMA untuk tiap Provinsi. Visualisasi dari penentuan objek *prototype* untuk klaster 1 dan klaster 2 dapat dilihat pada Gambar 4.5



(a)



(b)

Gambar 4.5 (a) *Multiple plot* runtun waktu klaster 1; (b) *Multiplot* runtun waktu klaster 2

Berdasarkan Gambar 4.5 dapat dilihat runtun waktu untuk objek *prototype* yang akan dilakukan pemodelan ARIMA. Model ARIMA yang terbentuk nantinya akan sebanyak klaster yang optimal, artinya pada penelitian ini model ARIMA yang akan terbentuk sebanyak 2 model.

1. Membagi data *in-sample* dan *out-sample*

Sebelum melakukan pemodelan ARIMA, data harga minyak goreng (baik data aktual maupun objek *prototype*) dibagi menjadi data *in-sample* dan data *out-sample* dengan proporsi pembagian 85:15, yaitu 85% data *in-sample* dan 15% data *out-sample*. Adapun hasil pembagian data *in-sample* dan data *out-sample* dapat dilihat pada perhitungan sebagai berikut:

$$\begin{aligned}
 \text{Jumlah data } in\text{-sample} &= \text{Proporsi data } in\text{-sample} \times N \\
 &= 85\% \times 61 \\
 &= 51,85 \approx 52 \\
 \text{Jumlah data } out\text{-sample} &= N - \text{jumlah data } in\text{-sample} \\
 &= 61 - 52 \\
 &= 9
 \end{aligned}$$

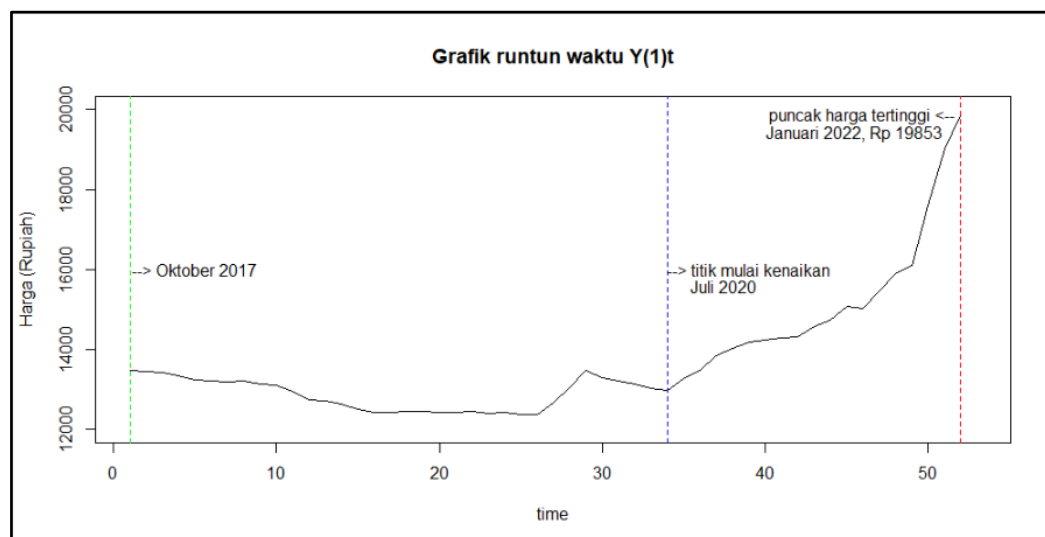
Berdasarkan perhitungan, sebanyak 52 data pertama (Oktober 2017 sampai dengan Januari 2022) menjadi data *in-sample*. Data *in-sample* yang nantinya digunakan untuk membuat model. Sedangkan sebanyak 9 data terakhir (Februari 2022 sampai dengan Oktober 2022) menjadi data *out-sample*. Data *out-sample* yang nantinya akan digunakan untuk melakukan proses validasi model peramalan.

2. Model ARIMA untuk klaster 1

Objek *prototype* yang terbentuk dari klaster 1 adalah rata-rata dari semua objek runtun waktu yang berada pada klaster 1 berdasarkan Tabel 4.8 yang dapat dimisalkan dengan notasi $Y(1)_t$. Tahapan-tahapan dalam proses pembentukan model ARIMA dari objek *prototype* adalah sebagai berikut:

a) Eksplorasi data runtun waktu

Sebelum melangkah lebih lanjut, terlebih dahulu akan dilakukan eksplorasi data runtun waktu dengan membuat grafik runtun waktu pada $Y(1)_t$ yaitu objek *prototype* klaster 1 dari bulan Oktober 2017 sampai dengan Januari 2022 untuk melihat gambaran dari perubahan, pola serta fluktuasi yang terbentuk dari nilai $Y(1)_t$. Grafik runtun waktu dari $Y(1)_t$ ditampilkan pada Gambar 4.6.



Gambar 4.6 Grafik runtun waktu $Y(1)_t$ bulan Oktober 2017 – Januari 2022

Berdasarkan Gambar 4.6, dapat dilihat bahwa nilai tertinggi terjadi pada periode terakhir atau pada saat $t = 52$ (bulan Januari 2022) yaitu sebesar Rp 19.853. Pada awal-awal periode hingga $t = 33$ (bulan Juni 2020) rata-rata harga minyak goreng cenderung stabil. Kemudian terjadi peningkatan rata-rata harga minyak goreng dari $t = 34$ (bulan Juli 2020) sampai dengan $t = 52$ (bulan Januari 2022). Dikutip dari <https://industri.kontan.co.id/>, Gabungan Industri Minyak Nabati Indonesia (GIMNI) memaparkan bahwa harga sawit mulai tak terbandung sejak bulan Mei 2020 di mana dari tahun 2020 hingga saat ini terjadi kenaikan harga sawit hingga 100% dan menyebabkan terjadinya kenaikan harga minyak goreng di dalam negeri, sehingga pada grafik runtun waktu membentuk pola *trend* naik.

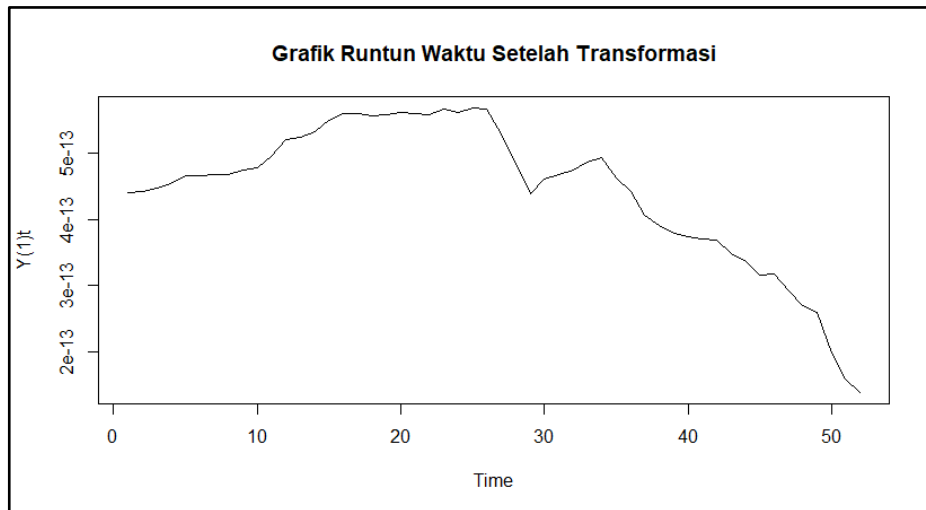
b) Stasioneritas Data

Eksplorasi data runtun waktu telah dilakukan dan berdasarkan Gambar 4.6 dapat dilihat terjadi peningkatan yang cukup drastis rata-rata harga minyak goreng dari $t = 35$ (bulan Agustus 2020) sampai dengan $t = 52$ (bulan Januari 2022) sehingga hal ini menyebabkan terjadinya perubahan variansi dari waktu ke waktu dan tidak berfluktuasi pada nilai rata-ratanya sehingga mengindikasikan $Y(1)_t$ tidak stasioner baik dalam variansi maupun rata-rata, sehingga perlu terlebih dahulu distasionerkan. Identifikasi stasioneritas untuk $Y(1)_t$ adalah sebagai berikut:

1. Stasioneritas dalam variansi

Berdasarkan Gambar 4.6, secara visual sedikit sulit untuk menentukan apakah data belum atau telah stasioner dalam variansi. Solusinya adalah pemeriksaan stasioneritas dalam variansi menggunakan transformasi Box-Cox untuk mengestimasi nilai λ . Hasil estimasi nilai λ yang diperoleh untuk $Y(1)_t$ adalah sebesar -2,99 (Lampiran 21). Dengan demikian nilai λ sebesar -2,99 menunjukkan data tidak stasioner dalam variansi karena nilai λ tersebut tidak mendekati 1, sehingga perlu dilakukan transformasi. Formulasi dari transformasi adalah memangkatkan data $Y(1)_t$ dengan nilai λ sebesar -2,99. Setelah data ditransformasi menggunakan transformasi pangkat, akan dilakukan pemeriksaan

kembali dengan melihat grafik runtun waktu dari data transformasi $Y(1)_t$ yang ditampilkan pada Gambar 4.7.

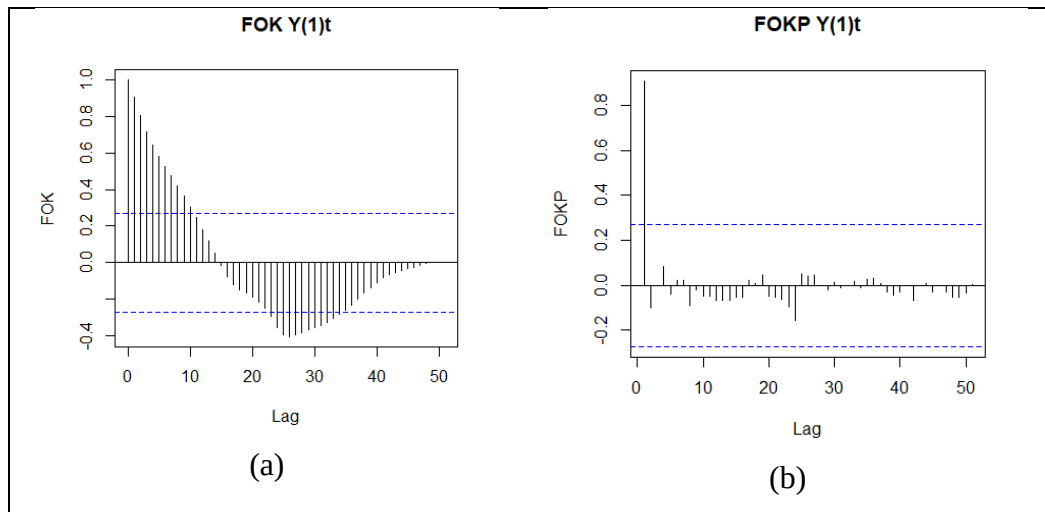


Gambar 4.7 Grafik runtun waktu setelah transformasi ($Y(1)_t^*$)

Setelah dilakukan transformasi pangkat, diperoleh estimasi nilai λ dari data $Y(1)_t$ sebesar 1,32 (Lampiran 21). Hal tersebut dapat diindikasikan data $Y(1)_t$ yang ditransformasi telah stasioner dalam variansi.

2. Stasioneritas dalam rata-rata

Setelah stasioneritas data dalam variansi telah terpenuhi, maka dilanjutkan dengan identifikasi stasioneritas dalam rata-rata. Berdasarkan Gambar 4.7 mengindikasikan bahwa data belum stasioner dalam rata-rata, hal ini dikarenakan data cenderung berfluktuatif tidak di sekitar nilai rata-rata yang konstan dari waktu ke waktu. Identifikasi stasioneritas dalam rata-rata dapat dilakukan dengan melihat grafik fungsi otokorelasi (FOK), kemudian untuk lebih meyakinkan dilanjutkan dengan melakukan pengujian hipotesis menggunakan uji *Augmented* Dickey-Fuller (ADF). FOK dan FOKP untuk $Y(1)_t^*$ ditampilkan pada Gambar 4.8.



Gambar 4.8 (a) Grafik FOK $Y(1)_t^*$; (b) Grafik FOKP $Y(1)_t^*$

Berdasarkan Gambar 4.8 (a) dapat dilihat bahwa nilai-nilai otokorelasi pada *lag-lag* awal untuk grafik FOK menurun lambat secara eksponensial atau membentuk pola *dies down*, sehingga berdasarkan grafik FOK tersebut mengindikasikan data $Y(1)_t^*$ belum stasioner dalam rata-rata. Hal ini menjadi lebih jelas ketika melakukan pengujian hipotesis menggunakan uji ADF pada Persamaan (2.16).

Hipotesis

$H_0 : \delta = 0$ (Data $Y(1)_t^*$ tidak stasioner dalam rata-rata)

$H_1 : \delta \neq 0$ (Data $Y(1)_t^*$ stasioner dalam rata-rata)

Taraf Signifikansi

$$\alpha = 5\%$$

Statistik Uji

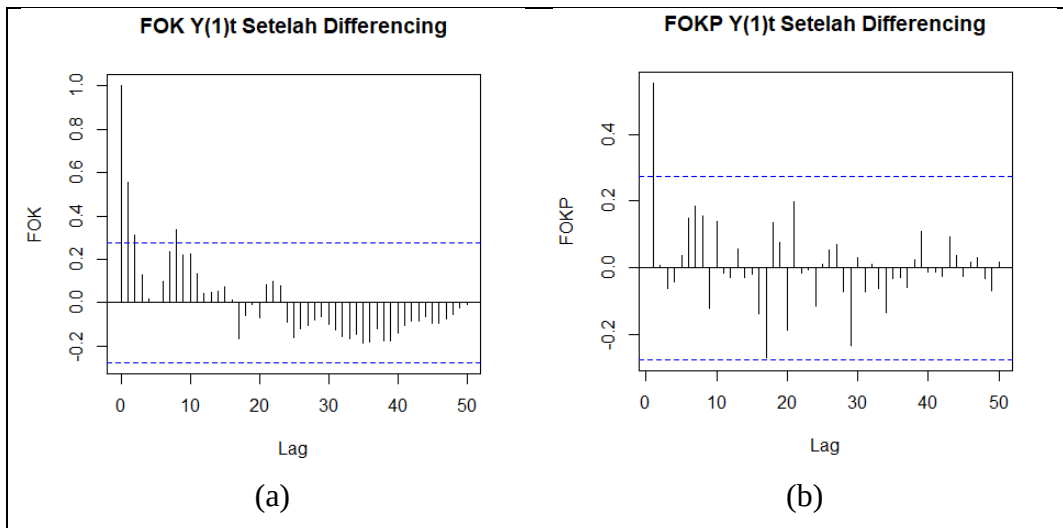
$$\tau = \frac{\hat{\delta}}{SE(\hat{\delta})} = 1,03 ; p - value = 0,30$$

Daerah Penolakan

$$H_0 \text{ ditolak jika } |\tau| > |\tau_{0,05;50}| = 1,95 \text{ atau } p - value < \alpha$$

Berdasarkan pengujian menunjukkan nilai $|\tau| = 1,03 < |\tau_{0,05;50}| = 1,95$ atau $p - value = 0,30 > 0,05$ (Lampiran 20), sehingga diputuskan H_0 gagal ditolak. Artinya data belum stasioner dalam rata-rata, sehingga perlu

distasionerkan terlebih dahulu dengan melakukan *differencing* orde 1 ($d = 1$) menggunakan Persamaan (2.14). Ketika data telah dilakukan *differencing*, maka akan diidentifikasi kembali dengan membentuk dan melihat pola dari FOK yang ditampilkan pada Gambar 4.9 (a).



Gambar 4.9 (a) Grafik FOK $Y(1)_t^*$ setelah *differencing* ; (b) Grafik FOKP $Y(1)_t^*$ setelah *differencing*

Berdasarkan Gambar 4.9 (a), terlihat bahwa grafik FOK mengalami *cut off* setelah *lag* 2, dikarenakan hal tersebut ditandai dengan nilai fungsi otokorelasi yang mengalami penurunan secara cepat menuju nol. Hal ini menjadi lebih jelas ketika melakukan pengujian hipotesis kembali menggunakan uji ADF untuk $Y(1)_t^*$ yang telah di *differencing*.

Hipotesis

$H_0 : \delta = 0$ (Data *differencing* $Y(1)_t^*$ tidak stasioner dalam rata-rata)

$H_1 : \delta \neq 0$ (Data *differencing* $Y(1)_t^*$ stasioner dalam rata-rata)

Taraf Signifikansi

$\alpha = 5\%$

Statistik Uji

$$\tau = \frac{\hat{\delta}}{SE(\hat{\delta})} = 2,76 ; p - value = 0,00$$

Daerah Penolakan

H_0 ditolak jika $|\tau| > |\tau_{0,05;50}| = 1,95$ atau $p - value < \alpha$.

Berdasarkan pengujian menunjukkan nilai $|\tau| = 2,76 > |\tau_{0,05;50}| = 1,95$ atau $p - value = 0,00 < 0,05$ (Lampiran 21), sehingga diputuskan H_0 ditolak. Artinya data *differencing* $Y(1)_t^*$ telah stasioner dalam rata-rata.

c) Identifikasi model ARIMA sementara

Stasioneritas data telah terpenuhi, selanjutnya adalah menentukan orde model ARIMA sementara yang dapat dilihat melalui grafik FOK dan FOKP serta kemudian mencocokkan pola dari kedua grafik tersebut dengan pola teoritiknya berdasarkan Tabel 2.2. Berdasarkan Gambar 4.9 (a), grafik FOK mengalami *cut off* setelah *lag* 2 sehingga orde untuk MA adalah 2. Sedangkan berdasarkan Gambar 4.9 (b), grafik FOKP mengalami *cut off* setelah *lag* 1 sehingga orde untuk AR adalah 1 dengan orde untuk *differencing* adalah 1. Pola teoritik pada Tabel 2.2 menjelaskan ketika FOK dan FOKP mengalami *cut off*, maka kemungkinan model yang akan terbentuk adalah AR (p) atau MA (q), sehingga kombinasi model ARIMA sementara yang terbentuk untuk model ARIMA klaster 1 adalah ARIMA (0,1,1), ARIMA (0,1,2), dan ARIMA (1,1,0).

Kombinasi model ARIMA sementara yang terbentuk dapat dituliskan menggunakan Persamaan (2.38) sebagai berikut:

1. Model ARIMA (0,1,1) dapat dituliskan menjadi

$$Y(1)_t = Y(1)_{t-1} + e(1)_t - \theta_1 e(1)_{t-1}$$

2. Model ARIMA (0,1,2) dapat dituliskan menjadi

$$Y(1)_t = Y(1)_{t-1} + e(1)_t - \theta_1 e(1)_{t-1} - \theta_2 e(1)_{t-2}$$

3. Model ARIMA (1,1,0) dapat dituliskan menjadi

$$Y(1)_t = Y(1)_{t-1} + \phi_1 Y(1)_{t-1} - \phi_1 Y(1)_{t-2} + e(1)_t$$

d) Estimasi dan pengujian signifikansi parameter

Sebelumnya telah dibentuk kombinasi model ARIMA sementara, maka langkah selanjutnya adalah melakukan penaksiran dan pengujian signifikansi parameter dari semua model sementara. Pada penelitian ini, metode estimasi yang digunakan adalah *maximum likelihood estimation* (MLE) dan pengujian

signifikansi parameter menggunakan Persamaan (2.47). Hasil Pengujian signifikansi parameter untuk seluruh model estimasi ARIMA sebagai berikut:

Hipotesis

$H_0 : \omega = \{\phi, \theta\} = 0$ (Parameter tidak signifikan pada model)

$H_1 : \omega = \{\phi, \theta\} \neq 0$ (Parameter signifikan pada model)

Taraf Signifikansi

$\alpha = 5\%$

Statistik Uji

$$t_{hit} = \frac{\hat{\omega}}{SE(\hat{\omega})}$$

Daerah Penolakan

H_0 ditolak jika $|t_{hit}| > t_{0,025,db}$ atau $p - value < \alpha$

Hasil estimasi parameter dan pengujian signifikansi parameter model ARIMA sementara untuk klaster 1 disajikan pada Tabel 4.10.

Tabel 4.10 Hasil Penaksiran dan Pengujian Signifikansi Parameter Model ARIMA Sementara untuk Klaster 1

Model	Estimasi	t_{hitung}	db	$t_{(0,025;db)}$	$P-value$	Keputusan
ARIMA (0,1,1)*	$\hat{\theta}_1 = 0,46$	4,65	50	2,00	0,00	H_0 ditolak
ARIMA (0,1,2)*	$\hat{\theta}_1 = 0,53$	4,09	49	2,01	0,00	H_0 ditolak
	$\hat{\theta}_2 = 0,28$	2,15		2,01	0,03	H_0 ditolak
ARIMA (1,1,0)*	$\hat{\phi}_1 = 0,59$	5,31	50	2,00	0,00	H_0 ditolak

Sumber: Lampiran 22

Keterangan: (*) Model dengan semua parameter yang signifikan

Berdasarkan Tabel 4.10, dapat diketahui ketiga model ARIMA sementara dapat digunakan untuk analisis selanjutnya yaitu ARIMA (0,1,1), ARIMA (0,1,2), dan ARIMA (1,1,0) karena semua parameter di dalam model tersebut signifikan.

e) Pemeriksaan diagnostik

Sebelumnya telah diketahui terdapat dua model ARIMA sementara yang semua parameternya signifikan, langkah selanjutnya adalah melakukan pemeriksaan diagnostik untuk membuktikan bahwa model-model tersebut apakah

cukup memadai untuk digunakan pada proses peramalan. Proses pemeriksaan diagnostik model terbagi menjadi dua, antara lain pengujian *error* berdistribusi normal dan pengujian independensi *error*.

1. Normalitas *error*

Uji asumsi ini bertujuan untuk mengetahui apakah data *error* telah memenuhi asumsi normalitas. Pengujian yang digunakan adalah uji Kolmogorov-Smirnov. Pengujian hipotesis untuk normalitas *error* adalah sebagai berikut:

Hipotesis

$$H_0 : F_0(Z) = F_n(Z) \text{ (Fungsi distribusi kumulatif = fungsi distribusi empiris)}$$

$$H_1 : F_0(Z) \neq F_n(Z) \text{ (Fungsi distribusi kumulatif} \neq \text{fungsi distribusi empiris)}$$

Taraf Signifikansi

$$\alpha = 5\%$$

Statistik Uji

$$D_{hit} = \sup \{|F_0(Z) - F_n(Z)|\}$$

Daerah Penolakan

$$H_0 \text{ ditolak jika } D_{hit} > D_{0,05,52} \text{ atau } p\text{-value} < \alpha$$

Hasil pengujian normalitas *error* disajikan pada Tabel 4.11.

Tabel 4.11 Hasil Pengujian Normalitas *Error* Kluster 1

Model	D_{hit}	$D_{0,05,52}$	$P\text{-value}$	Keputusan
ARIMA (0,1,1)	0,15	0,18	0,14	H_0 gagal ditolak
ARIMA (0,1,2)	0,17	0,18	0,06	H_0 gagal ditolak
ARIMA (1,1,0)	0,17	0,18	0,08	H_0 gagal ditolak

Sumber: Lampiran 23

Berdasarkan Tabel 4.11, dapat dilihat bahwa ketiga model yaitu ARIMA (0,1,1), ARIMA (0,1,2) dan ARIMA (1,1,0) telah memenuhi asumsi *error* berdistribusi normal.

2. Independensi *error*

Uji independensi *error* dilakukan untuk mengetahui apakah *error* antar *lag* saling berkorelasi atau tidak dengan menggunakan uji Ljung-Box. Pengujian hipotesis untuk independensi *error* adalah sebagai berikut:

Hipotesis

$$H_0 : \rho_1 = \rho_2 = \dots = \rho_{51} = 0$$

(Tidak terdapat korelasi antar *error*)

$$H_1 : \text{minimal ada satu } \rho_j \neq 0, \text{ untuk } j = 1, 2, 3, \dots, 51$$

(Terdapat korelasi antar *error*)

Taraf Signifikansi

$$\alpha = 5\%$$

Statistik Uji

$$Q^* = n(n+2) \sum_{k=1}^{k^*} \frac{\hat{\rho}_k^2}{(n-k)}$$

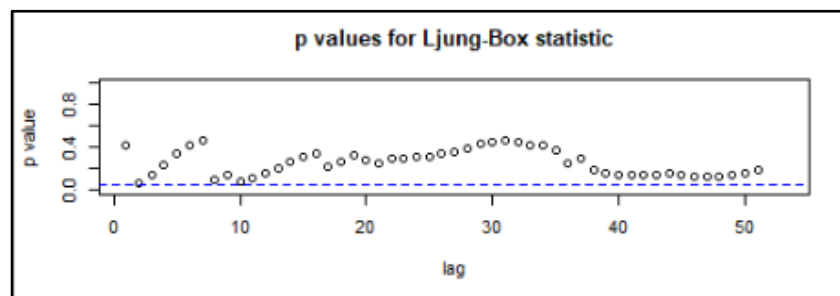
Daerah Penolakan

$$H_0 \text{ ditolak jika } Q^* > \chi_{\alpha, db}^2 \text{ atau } p\text{-value} < \alpha$$

Hasil dari pengujian independensi *error* (Lampiran 24) untuk setiap model divisualisasikan menggunakan grafik dari nilai *p-value* pada pengujian Ljung-Box sebagai berikut:

- Model ARIMA (0,1,1)

Nilai *p-value* grafik Ljung-Box untuk model ARIMA (0,1,1) dapat dilihat pada Gambar 4.10.

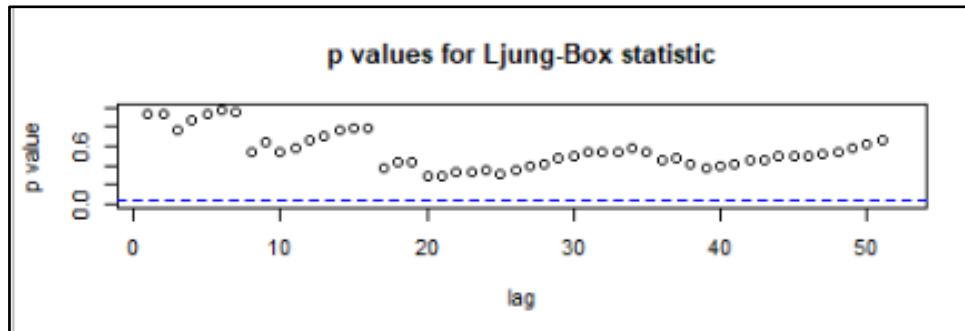


Gambar 4.10 Grafik Ljung-Box model ARIMA (0,1,1) untuk klaster 1

Berdasarkan Gambar 4.10 dapat dilihat bahwa *error* dari model ARIMA (0,1,1) memenuhi asumsi independensi *error*. Hal tersebut dikarenakan untuk setiap *lag* memiliki nilai *p-value* lebih besar dari 0,05 yang ditandai dengan semua nilai yang berada di atas garis batas 0,05.

- Model ARIMA (0,1,2)

Nilai *p-value* grafik Ljung-Box untuk model ARIMA (0,1,2) dapat dilihat pada Gambar 4.11

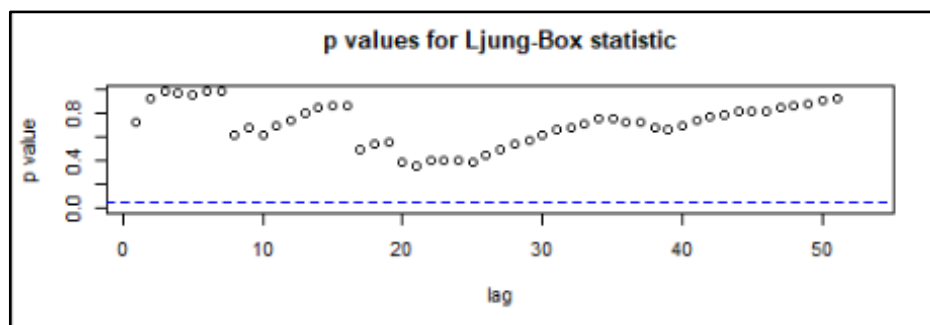


Gambar 4.11 Grafik Ljung-Box model ARIMA (0,1,2) untuk klaster 1

Berdasarkan Gambar 4.11 dapat dilihat bahwa *error* dari model ARIMA (0,1,2) memenuhi asumsi independensi *error*. Hal tersebut dikarenakan untuk setiap *lag* memiliki nilai *p-value* lebih besar dari 0,05 yang ditandai dengan semua nilai yang berada di atas garis batas 0,05.

- Model ARIMA (1,1,0)

Nilai *p-value* grafik Ljung-Box untuk model ARIMA (1,1,0) dapat dilihat pada Gambar 4.12



Gambar 4.12 Grafik Ljung-Box model ARIMA (1,1,0) untuk klaster 1

Berdasarkan Gambar 4.12 dapat dilihat bahwa *error* dari model ARIMA (1,1,0) memenuhi asumsi independensi *error*. Hal tersebut dikarenakan untuk setiap *lag* memiliki nilai *p-value* lebih besar dari 0,05 yang ditandai dengan semua nilai yang berada di atas garis batas 0,05.

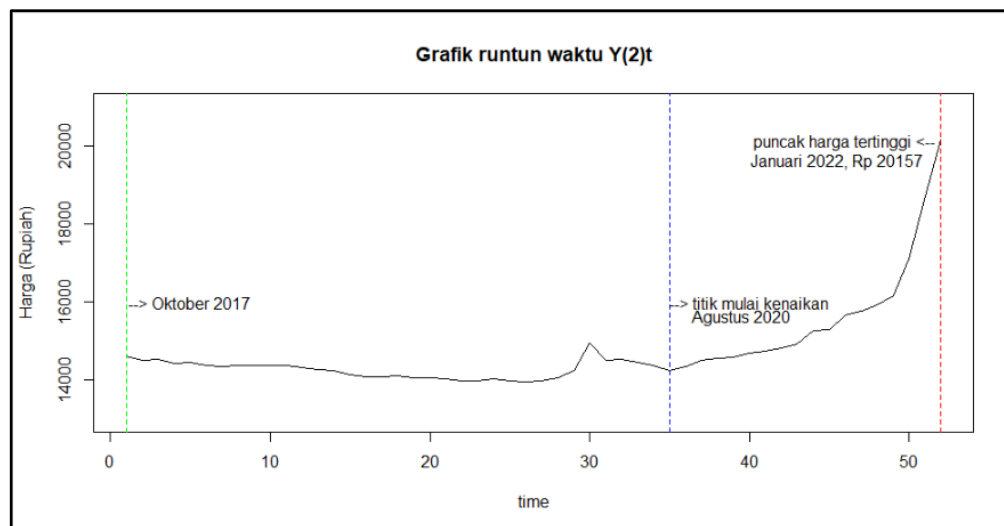
Berdasarkan kedua uji pemeriksaan diagnostik yang telah dilakukan yaitu uji normalitas *error* dan uji independensi *error* maka dapat disimpulkan ketiga model yaitu ARIMA (0,1,1), ARIMA (0,1,2), dan ARIMA (1,1,0) dinyatakan layak digunakan untuk melanjutkan ke tahap peramalan harga minyak goreng di Indonesia untuk klaster 1.

3. Model ARIMA untuk klaster 2

Objek *prototype* yang terbentuk dari klaster 2 adalah rata-rata dari semua objek runtun waktu yang berada pada klaster 2 berdasarkan Tabel 4.8 yang dapat dimisalkan dengan notasi $Y(2)_t$. Tahapan-tahapan dalam proses pembentukan model ARIMA dari objek *prototype* adalah sebagai berikut:

a) Eksplorasi data runtun waktu

Sebelum melangkah lebih lanjut, terlebih dahulu akan dilakukan eksplorasi data runtun waktu dengan membuat grafik runtun waktu pada $Y(2)_t$ yaitu objek *prototype* klaster 2 dari bulan Oktober 2017 sampai dengan Januari 2022 untuk melihat gambaran dari perubahan, pola serta fluktuasi yang terbentuk dari nilai $Y(2)_t$. Grafik runtun waktu dari $Y(2)_t$ ditampilkan pada Gambar 4.13.



Gambar 4.13 Grafik runtun waktu $Y(2)_t$ bulan Oktober 2017 – Januari 2022

Berdasarkan Gambar 4.13, dapat dilihat bahwa nilai tertinggi terjadi pada periode terakhir atau pada saat $t = 52$ (bulan Januari 2022) yaitu sebesar Rp 20157. Pada

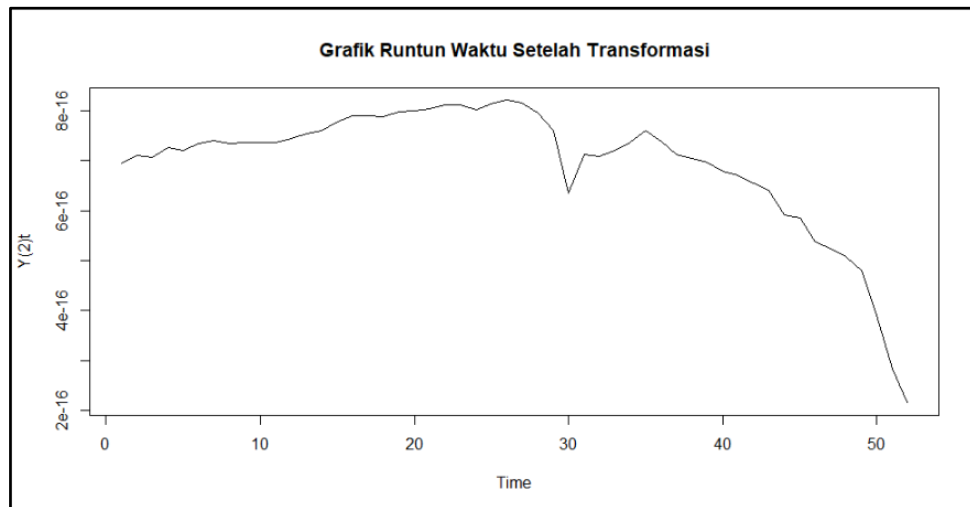
awal-awal periode hingga $t = 34$ (bulan Juli 2020) rata-rata harga minyak goreng cenderung stabil. Kemudian terjadi peningkatan rata-rata harga minyak goreng dari $t = 35$ (bulan Agustus 2020) sampai dengan $t = 52$ (bulan Januari 2022). Dikutip dari <https://industri.kontan.co.id/>, Gabungan Industri Minyak Nabati Indonesia (GIMNI) memaparkan bahwa harga sawit mulai tak terbendung sejak bulan Mei 2020 di mana dari tahun 2020 hingga saat ini terjadi kenaikan harga sawit hingga 100% dan menyebabkan terjadinya kenaikan harga minyak goreng di dalam negeri, sehingga pada grafik runtun waktu membentuk pola *trend* naik.

b) Stasioneritas Data

Eksplorasi data runtun waktu telah dilakukan dan berdasarkan Gambar 4.13 dapat dilihat terjadi peningkatan yang cukup drastis rata-rata harga minyak goreng dari $t = 35$ (bulan Agustus 2020) sampai dengan $t = 52$ (bulan Januari 2022) sehingga hal ini menyebabkan terjadinya perubahan variansi dari waktu ke waktu dan tidak berfluktuasi pada nilai rata-ratanya sehingga mengindikasikan $Y(2)_t$ tidak stasioner baik dalam variansi maupun rata-rata, sehingga perlu terlebih dahulu distasionerkan. Identifikasi stasioneritas untuk $Y(2)_t$ adalah sebagai berikut:

1. Stasioneritas dalam variansi

Berdasarkan Gambar 4.13, secara visual sedikit sulit untuk menentukan apakah data belum atau telah stasioner dalam variansi. Solusinya adalah pemeriksaan stasioneritas dalam variansi menggunakan transformasi Box-Cox untuk mengestimasi nilai λ . Hasil estimasi nilai λ yang diperoleh untuk $Y(2)_t$ adalah sebesar -3,64 (Lampiran 25). Dengan demikian nilai λ sebesar -3,64 menunjukkan data tidak stasioner dalam variansi karena nilai λ tersebut tidak mendekati 1, sehingga perlu dilakukan transformasi. Formulasi dari transformasi adalah mengangkat data $Y(2)_t$ dengan nilai λ sebesar -3,64. Setelah data ditransformasi menggunakan transformasi pangkat, akan dilakukan pemeriksaan kembali dengan melihat grafik runtun waktu dari data transformasi $Y(2)_t$ yang ditampilkan pada Gambar 4.14.

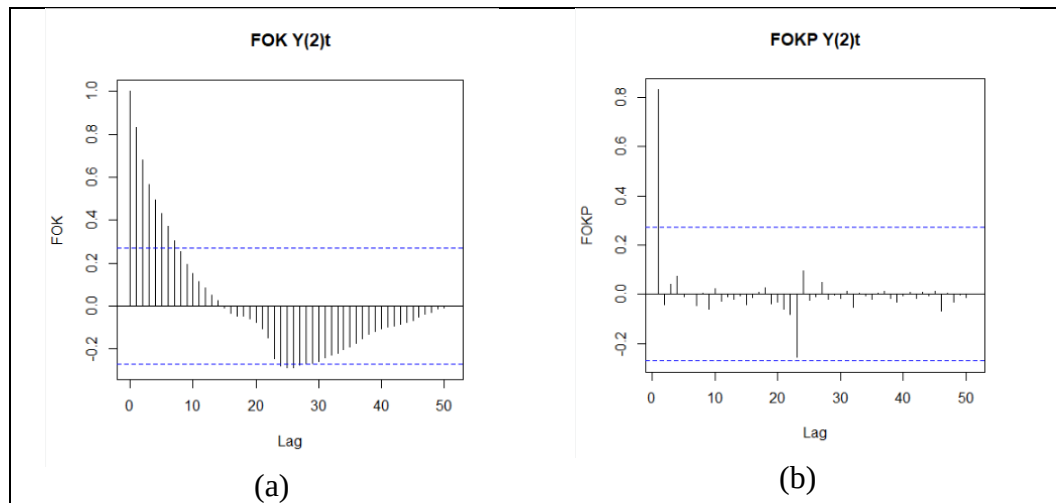


Gambar 4.14 Grafik runtun waktu setelah transformasi ($Y(2)_t^*$)

Setelah dilakukan transformasi pangkat, diperoleh estimasi nilai λ dari data $Y(2)_t$ sebesar 1,01 yang artinya telah mendekati 1 (Lampiran 25). Hal tersebut dapat diindikasikan data $Y(2)_t$ yang ditransformasi telah stasioner dalam variansi.

2. Stasioneritas dalam rata-rata

Setelah stasioneritas data dalam variansi telah terpenuhi, maka dilanjutkan dengan identifikasi stasioneritas dalam rata-rata. Berdasarkan Gambar 4.14 mengindikasikan bahwa data belum stasioner dalam rata-rata, hal ini dikarenakan data cenderung berfluktuatif tidak di sekitar nilai rata-rata yang konstan dari waktu ke waktu. Identifikasi stasioneritas dalam rata-rata dapat dilakukan dengan melihat grafik fungsi otokorelasi (FOK), kemudian untuk lebih meyakinkan dilanjutkan dengan melakukan pengujian hipotesis menggunakan uji *Augmented* Dickey-Fuller (ADF). FOK dan FOKP untuk $Y(2)_t^*$ ditampilkan pada Gambar 4.15.



Gambar 4.15 (a) Grafik FOK $Y(2)_t^*$; (b) Grafik FOKP $Y(2)_t^*$

Berdasarkan Gambar 4.15 (a) dapat dilihat bahwa nilai-nilai otokorelasi pada *lag-lag* awal untuk grafik FOK menurun lambat secara eksponensial atau membentuk pola *dies down*, sehingga berdasarkan grafik FOK tersebut mengindikasikan data $Y(2)_t^*$ belum stasioner dalam rata-rata. Hal ini menjadi lebih jelas ketika melakukan pengujian hipotesis menggunakan uji ADF pada Persamaan (2.16).

Hipotesis

$H_0 : \delta = 0$ (Data $Y(2)_t^*$ tidak stasioner dalam rata-rata)

$H_1 : \delta \neq 0$ (Data $Y(2)_t^*$ stasioner dalam rata-rata)

Taraf Signifikansi

$\alpha = 5\%$

Statistik Uji

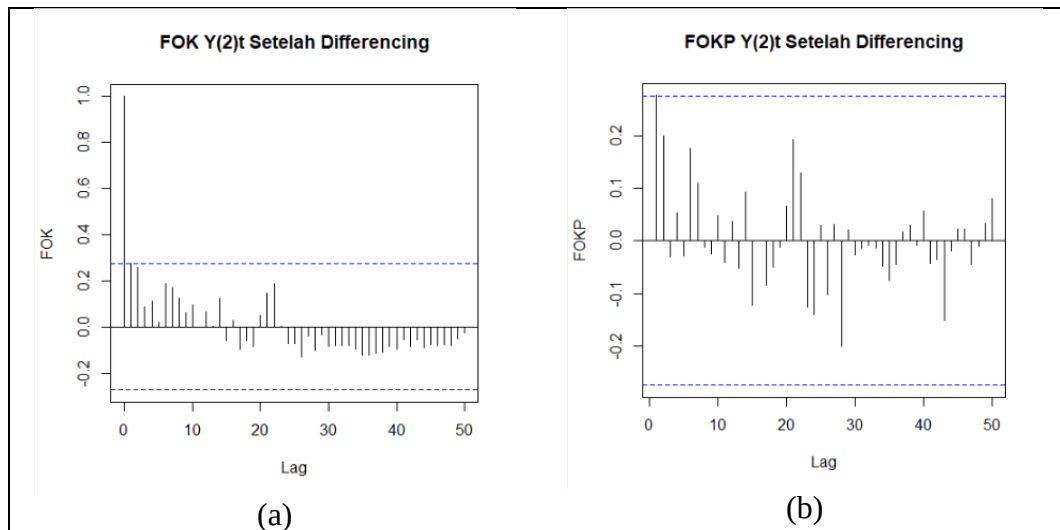
$$\tau = \frac{\hat{\delta}}{SE(\hat{\delta})} = 1,22 ; p - value = 0,22$$

Daerah Penolakan

H_0 ditolak jika $|\tau| > |\tau_{0,05;50}| = 1,95$ atau $p - value < \alpha$

Berdasarkan pengujian menunjukkan nilai $|\tau| = 1,22 < |\tau_{0,05;50}| = 1,95$ atau $p - value = 0,22 > 0,05$ (Lampiran 25), sehingga diputuskan H_0 gagal ditolak. Artinya data belum stasioner dalam rata-rata, sehingga perlu

distasionerkan terlebih dahulu dengan melakukan *differencing* orde 1 ($d = 1$) menggunakan Persamaan (2.14). Ketika data telah dilakukan *differencing*, maka akan diidentifikasi kembali dengan membentuk dan melihat pola dari FOK yang ditampilkan pada Gambar 4.16 (a).



Gambar 4.16 (a) Grafik FOK $Y(2)_t^*$ setelah *differencing* ; (b) Grafik FOKP $Y(2)_t^*$ setelah *differencing*

Berdasarkan Gambar 4.16 (a), terlihat bahwa grafik FOK mengalami *cut off* setelah *lag* 1, dikarenakan hal tersebut ditandai dengan nilai fungsi otokorelasi yang mengalami penurunan secara cepat menuju nol. Hal ini menjadi lebih jelas ketika melakukan pengujian hipotesis kembali menggunakan uji ADF untuk $Y(2)_t^*$ yang telah di *differencing*.

Hipotesis

$H_0 : \delta = 0$ (Data *differencing* $Y(2)_t^*$ tidak stasioner dalam rata-rata)

$H_1 : \delta \neq 0$ (Data *differencing* $Y(2)_t^*$ stasioner dalam rata-rata)

Taraf Signifikansi

$$\alpha = 5\%$$

Statistik Uji

$$\tau = \frac{\hat{\delta}}{SE(\hat{\delta})} = 2,11 ; p - value = 0,03$$

Daerah Penolakan

H_0 ditolak jika $|\tau| > |\tau_{0,05;50}| = 1,95$ atau $p - value < \alpha$.

Berdasarkan pengujian menunjukkan nilai $|\tau| = 2,11 > |\tau_{0,05;50}| = 1,95$ atau $p - value = 0,03 < 0,05$ (Lampiran 25), sehingga diputuskan H_0 ditolak. yang artinya data *differencing* $Y(2)_t^*$ telah stasioner dalam rata-rata dan proses pemodelan ARIMA dapat dilanjutkan ke tahap berikutnya.

c) Identifikasi model ARIMA sementara

Stasioneritas data telah terpenuhi, selanjutnya adalah menentukan orde model ARIMA sementara (orde AR dan MA) yang dapat dilihat melalui grafik FOK dan FOKP serta kemudian mencocokkan pola dari kedua grafik tersebut dengan pola teoritiknya berdasarkan Tabel 2.2. Berdasarkan Gambar 4.16 (a), grafik FOK mengalami *cut off* setelah *lag* 1 sehingga orde untuk MA adalah 1. Sedangkan berdasarkan Gambar 4.16 (b), grafik FOKP mengalami *cut off* setelah *lag* 1 sehingga orde untuk AR adalah 1 dengan orde untuk *differencing* adalah 1. Pola teoritik pada Tabel 2.2 menjelaskan ketika FOK dan FOKP mengalami *cut off*, maka kemungkinan model yang akan terbentuk adalah AR (p) atau MA (q), sehingga kombinasi model ARIMA sementara yang terbentuk untuk model ARIMA klaster 2 adalah ARIMA (0,1,1) dan ARIMA (1,1,0).

Kombinasi model ARIMA sementara yang terbentuk dapat dituliskan menggunakan Persamaan (2.38) sebagai berikut:

1. Model ARIMA (0,1,1) dapat dituliskan menjadi

$$Y(2)_t = Y(2)_{t-1} + e(2)_t - \theta_1 e(2)_{t-1}$$

2. Model ARIMA (1,1,0) dapat dituliskan menjadi

$$Y(2)_t = Y(2)_{t-1} + \phi_1 Y(2)_{t-1} - \phi_1 Y(2)_{t-2} + e(2)_t$$

d) Estimasi dan pengujian signifikansi parameter

Sebelumnya telah dibentuk kombinasi model ARIMA sementara, maka langkah selanjutnya adalah melakukan penaksiran dan pengujian signifikansi parameter dari semua model sementara. Pada penelitian ini, metode estimasi yang digunakan adalah MLE dan pengujian signifikansi parameter menggunakan

Persamaan (2.47). Hasil Pengujian signifikansi parameter untuk seluruh model estimasi ARIMA sebagai berikut:

Hipotesis

$H_0 : \omega = \{\phi, \theta\} = 0$ (Parameter tidak signifikan pada model)

$H_1 : \omega = \{\phi, \theta\} \neq 0$ (Parameter signifikan pada model)

Taraf Signifikansi

$\alpha = 5\%$

Statistik Uji

$$t_{hit} = \frac{\hat{\omega}}{SE(\hat{\omega})}$$

Daerah Penolakan

H_0 ditolak jika $|t_{hit}| > t_{0,025,db}$

Hasil estimasi dan pengujian signifikansi parameter disajikan pada Tabel 4.12.

Tabel 4.12 Hasil Penaksiran dan Pengujian Signifikansi Parameter Model ARIMA Sementara untuk Klaster 2

Model	Estimasi	t_{hitung}	db	$t_{(0,025;db)}$	P-value	Keputusan
ARIMA (0,1,1)*	$\hat{\theta}_1 = 0,22$	2,05	50	2,00	0,03	H_0 ditolak
ARIMA (1,1,0)*	$\hat{\phi}_1 = 0,59$	2,55	50	2,00	0,01	H_0 ditolak

Sumber: Lampiran 26

Keterangan: (*) Model dengan semua parameter yang signifikan

Berdasarkan Tabel 4.12, dapat diketahui kedua model ARIMA sementara dapat digunakan untuk analisis selanjutnya yaitu ARIMA (0,1,1) dan ARIMA (1,1,0) karena semua parameter di dalam model tersebut signifikan.

e) Pemeriksaan diagnostik

Sebelumnya telah diketahui kedua model ARIMA sementara yang semua parameternya signifikan, langkah selanjutnya adalah melakukan pemeriksaan diagnostik untuk membuktikan bahwa model-model tersebut apakah cukup memadai untuk digunakan pada proses peramalan. Proses pemeriksaan diagnostik model terbagi menjadi dua, antara lain pengujian *error* berdistribusi normal dan pengujian independensi *error*.

1. Normalitas *error*

Uji asumsi ini bertujuan untuk mengetahui apakah data *error* telah memenuhi asumsi normalitas. Pada penelitian ini pengujian yang digunakan menggunakan uji Kolmogorov-Smirnov. Pengujian hipotesis untuk normalitas *error* adalah sebagai berikut:

Hipotesis

$H_0 : F_0(Z) = F_n(Z)$ (Fungsi distribusi kumulatif = fungsi distribusi empiris)

$H_1 : F_0(Z) \neq F_n(Z)$ (Fungsi distribusi kumulatif $\neq$ fungsi distribusi empiris)

Taraf Signifikansi

$$\alpha = 5\%$$

Statistik Uji

$$D_{hit} = \sup \{|F_0(Z) - F_n(Z)|\}$$

Daerah Penolakan

$$H_0 \text{ ditolak jika } D_{hit} > D_{0,05,52} \text{ atau } p\text{-value} < \alpha$$

Hasil pengujian normalitas *error* parameter untuk klaster 2 disajikan pada Tabel 4.13.

Tabel 4.13 Hasil Pengujian Normalitas *Error* Klaster 2

Model	D_{hit}	$D_{0,05,52}$	$P\text{-value}$	Keputusan
ARIMA (0,1,1)	0,17	0,18	0,08	H_0 gagal ditolak
ARIMA (1,1,0)	0,17	0,18	0,06	H_0 gagal ditolak

Sumber: Lampiran 25

Berdasarkan Tabel 4.13, dapat dilihat bahwa kedua model baik ARIMA (0,1,1) maupun ARIMA (1,1,0) telah memenuhi asumsi *error* berdistribusi normal.

2. Independensi *error*

Uji independensi *error* dilakukan untuk mengetahui apakah *error* antar *lag* saling berkorelasi atau tidak dengan menggunakan uji Ljung-Box. Pengujian hipotesis untuk independensi *error* adalah sebagai berikut:

Hipotesis

$$H_0 : \rho_1 = \rho_2 = \dots = \rho_{51} = 0$$

(Tidak terdapat korelasi antar *error*)

H_1 : minimal ada satu $\rho_j \neq 0$, untuk $j = 1, 2, 3, \dots, 51$

(Terdapat korelasi antar *error*)

Taraf Signifikansi

$\alpha = 5\%$

Statistik Uji

$$Q^* = n(n+2) \sum_{k=1}^{k^*} \frac{\hat{\rho}_k^2}{(n-k)}$$

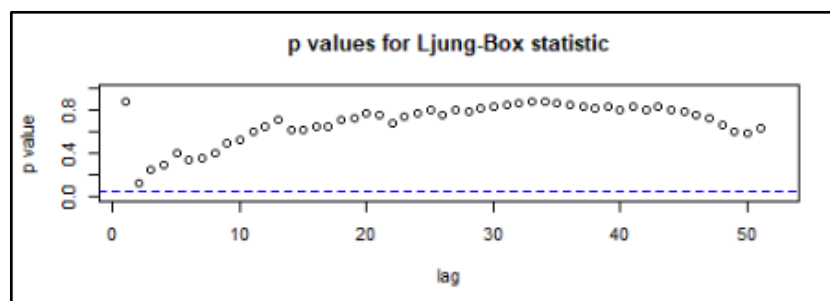
Daerah Penolakan

H_0 ditolak jika $Q^* > \chi_{\alpha, db}^2$ atau $p\text{-value} < \alpha$

Hasil dari pengujian independensi *error* (Lampiran 28) untuk setiap model divisualisasikan menggunakan grafik dari nilai $p\text{-value}$ pada pengujian Ljung-Box sebagai berikut:

- Model ARIMA (0,1,1)

Nilai $p\text{-value}$ grafik Ljung-Box untuk model ARIMA (0,1,1) dapat dilihat pada Gambar 4.17

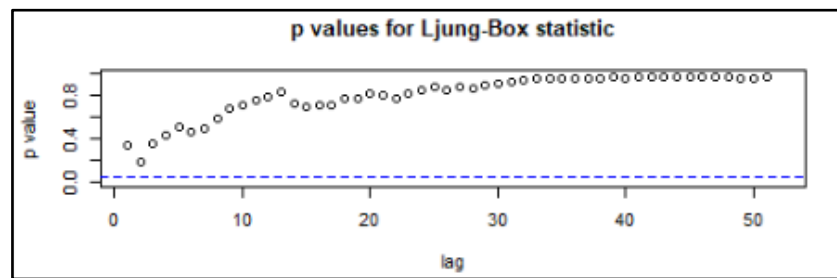


Gambar 4.17 Grafik Ljung-Box model ARIMA (0,1,1) untuk kluster 2

Berdasarkan Gambar 4.17 dapat dilihat bahwa *error* dari model ARIMA (0,1,1) memenuhi asumsi independensi *error*. Hal tersebut dikarenakan untuk setiap *lag* memiliki nilai $p\text{-value}$ lebih besar dari 0,05 yang ditandai dengan semua nilai yang berada di atas garis batas 0,05.

- Model ARIMA (0,1,1)

Nilai $p\text{-value}$ grafik Ljung-Box untuk model ARIMA (1,1,0) dapat dilihat pada Gambar 4.18.



Gambar 4.18 Grafik Ljung-Box model ARIMA (1,1,0) untuk klaster 2

Berdasarkan Gambar 4.18 dapat dilihat bahwa *error* dari model ARIMA (1,1,0) memenuhi asumsi independensi *error*. Hal tersebut dikarenakan untuk setiap *lag* memiliki nilai *p-value* lebih besar dari 0,05 yang ditandai dengan semua nilai yang berada di atas garis batas 0,05. Kedua uji pemeriksaan diagnostik yang telah dilakukan yaitu uji normalitas *error* dan uji independensi *error*, maka dapat disimpulkan kedua model yaitu ARIMA (0,1,1) dan ARIMA (1,1,0) dinyatakan layak digunakan untuk melanjutkan ke tahap peramalan harga minyak goreng di Indonesia untuk klaster 2.

4.3.3 Peramalan Harga Minyak Goreng di Indonesia

Setelah dilakukan pemodelan ARIMA untuk masing-masing klaster, tahap selanjutnya adalah melakukan peramalan untuk 15 periode ke depan yaitu 9 periode untuk data *out-sample* dan 6 periode untuk peramalan periode kedepan. Data *out-sample* pada penelitian ini terhitung dari bulan Februari 2022 sampai dengan Oktober 2022 yaitu sebanyak 9 periode. Berikut merupakan hasil dari identifikasi model ARIMA yang layak untuk setiap klaster yang disajikan pada Tabel 4.14

Tabel 4.14 Model ARIMA Layak untuk Setiap Klaster

Klaster	Model ARIMA	
1	ARIMA (0,1,1)	$Y(1)_t = Y(1)_{t-1} + e(1)_t - \hat{\theta}_1 e(1)_{t-1}$
	ARIMA (0,1,2)	$Y(1)_t = Y(1)_{t-1} + e(1)_t - \hat{\theta}_1 e(1)_{t-1} - \hat{\theta}_2 e(1)_{t-2}$
	ARIMA (1,1,0)	$Y(1)_t = Y(1)_{t-1} + \hat{\phi}_1 Y(1)_{t-1} - \hat{\phi}_1 Y(1)_{t-2} + e(1)_t$
2	ARIMA (0,1,1)	$Y(2)_t = Y(2)_{t-1} + e(2)_t - \hat{\theta}_1 e(2)_{t-1}$
	ARIMA (1,1,0)	$Y(2)_t = Y(2)_{t-1} + \hat{\phi}_1 Y(2)_{t-1} - \hat{\phi}_1 Y(2)_{t-2} + e(2)_t$

1. Klaster 1

Berdasarkan Tabel 4.14, dapat diketahui bahwa pada klaster 1 terdapat 3 model layak yang dapat digunakan untuk peramalan. Ketiga model layak yaitu model ARIMA (0,1,1), ARIMA (0,1,2), dan ARIMA (1,1,0) akan diramalkan pada tingkat klaster dan memilih salah satu dari kedua model tersebut yang memiliki ukuran akurasi peramalan terkecil untuk meramalkan anggota-anggota yang bergabung dalam satu klaster. Ukuran akurasi peramalan yang digunakan adalah AIC dan MAPE.

a. Peramalan model ARIMA (0,1,1)

Model ARIMA (0,1,1) dapat dituliskan $Y(1)_t = Y(1)_{t-1} + e(1)_t - \hat{\theta}_1 e(1)_{t-1}$ dengan nilai koefisien untuk parameter berdasarkan Tabel 4.9 diperoleh

$$Y(1)_t = Y(1)_{t-1} + e(1)_t - 0,46e(1)_{t-1}$$

Proses untuk menghitung nilai dari $\hat{Y}(1)_1$ tidak memiliki informasi $Y(1)_0$ maka akan diberikan contoh untuk mencari nilai dari $\hat{Y}(1)_2$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(1)_2 &= Y(1)_1 + e(1)_2 - 0,46e(1)_1 \\ &= 13.476 + e(1)_2 - ((0,46)(-4,50)) \\ &= 13.474 \text{ (Berdasarkan perhitungan software R)}\end{aligned}$$

Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(1)_{52}$, sehingga diperoleh hasil prediksi yang disajikan pada Tabel 4.15.

Tabel 4.15 Prediksi Data Harga Minyak Goreng Klaster 1 Model ARIMA (0,1,1)

t	$Y(1)_t$ (Rupiah/Liter)	$\hat{Y}(1)_t$ (Rupiah/Liter)
1	13.476	13.480
2	13.455	13.474
3	13.418	13.446
4	13.345	13.404
$\vdots$	$\vdots$	$\vdots$
52	19.853	19.273

Sumber: Lampiran 30

Selanjutnya adalah menghitung peramalan untuk data *out-sample*. Adapun sebelum dilakukan perhitungan, terlebih dahulu dilakukan pembangkitan secara *random* data *error* sebanyak 67 data yang bersifat *white noise* dan berdistribusi $N(82,41; 263,98)$ berdasarkan data aktual dan prediksi dari data *in-sample* pada Tabel 4.15. Hasil bangkitan data dapat dilihat pada Lampiran 29. Sebagai contoh akan dicari nilai dari $\hat{Y}(1)_{53}$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(1)_{53} &= Y(1)_{52} + e(1)_{53} - 0,46e(1)_{52} \\ &= 19.853 + e(1)_{53} - ((0,46)(579,80)) \\ &= 19.586\end{aligned}$$

Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(1)_{61}$, sehingga diperoleh hasil peramalan data *out-sample* yang disajikan pada Tabel 4.16

Tabel 4.16 Peramalan Data *Out-Sample* Harga Minyak Goreng Klaster 1 Model ARIMA (0,1,1)

t	$Y(1)_t$ (Rupiah/Liter)	$\hat{Y}(1)_t$ (Rupiah/Liter)
53	19.555	19.586
54	18.155	19.398
55	22.589	19.325
56	23.063	19.169
57	22.271	19.134
58	21.642	18.903
59	19.582	18.923
60	18.897	19.012
61	18.334	18.858

Sumber: Lampiran 30

Berdasarkan Tabel 4.16 terdapat dua informasi yaitu data aktual dan data hasil peramalan. Perhitungan dimulai dengan prediksi ($t = 1$ hingga $t = 52$) kemudian dilanjutkan dengan peramalan sebanyak data *out-sample* ($t = 53$ hingga $t = 61$) menggunakan model ARIMA (0,1,1). Selanjutnya menghitung akurasi dari model peramalan dari model ARIMA (0,1,1) dan diperoleh akurasi peramalan yang disajikan pada Tabel 4.17

Tabel 4.17 Nilai Akurasi Model ARIMA (0,1,1) Klaster 1

Data	Jenis Akurasi	Nilai Akurasi
<i>In-sample</i>	AIC	-3.087,35
	MAPE	1,00%
<i>Out-sample</i>	MAPE	7,98%

Sumber: Lampiran 22 & Lampiran 37

Langkah selanjutnya akan dilakukan peramalan 6 periode ke depan untuk harga minyak goreng klaster 1 menggunakan model ARIMA (0,1,1). Sebagai contoh akan dicari nilai dari $\hat{Y}(1)_{62}$, maka perhitungannya adalah

$$\begin{aligned}
 \hat{Y}(1)_{62} &= Y(1)_{61} + e(1)_{62} - 0,46e(1)_{61} \\
 &= 18.858 + e(1)_{53} - 0,46(131,91) \\
 &= 18.797
 \end{aligned}$$

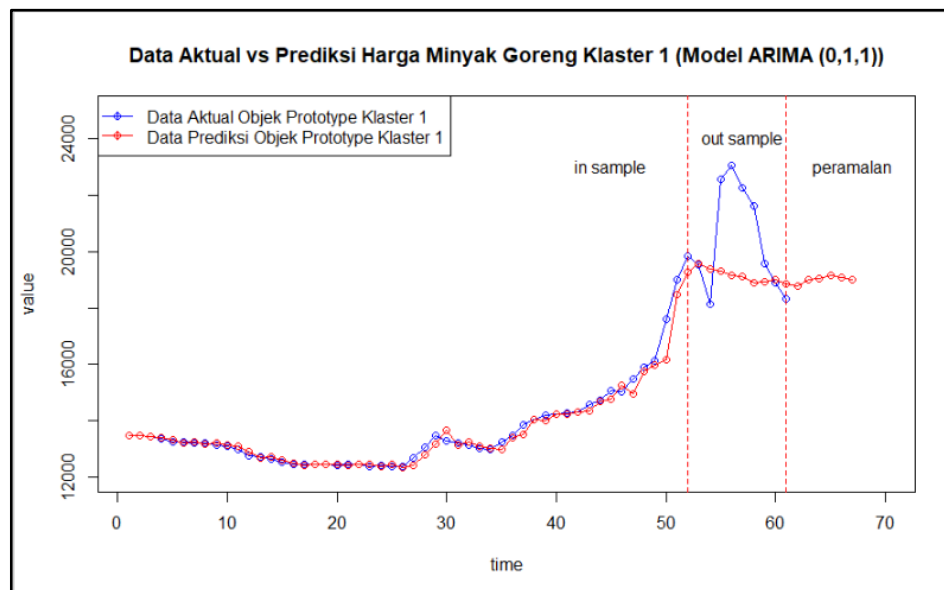
Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(1)_{67}$, sehingga diperoleh hasil peramalan yang disajikan pada Tabel 4.18

Tabel 4.18 Peramalan Data Harga Minyak Goreng Klaster 1 untuk Model ARIMA (0,1,1)

<i>t</i>	Bulan	$\hat{Y}(1)_t$ (Rupiah/Liter)
62	November 2022	18.797
63	Desember 2022	19.001
64	Januari 2023	19.045
65	Februari 2023	19.173
66	Maret 2023	19.081
67	April 2023	19.019

Sumber: Lampiran 30

Selanjutnya grafik data aktual dan peramalan untuk klaster 1 ($Y(1)_t$) untuk data *in-sample*, *out-sample*, dan hasil ramalan model ARIMA (0,1,1) ditampilkan pada Gambar 4.19.



Gambar 4.19 Data aktual vs prediksi klaster 1 untuk model ARIMA (0,1,1)

Berdasarkan Gambar 4.19 dapat dilihat bahwa grafik runtun waktu untuk data *in-sample* nilai prediksinya hampir mengikuti pola data aktual dengan tingkat akurasi peramalan menggunakan MAPE sebesar 1,00%. Kemudian untuk data *out-sample* periode pertama hampir mengikuti pola data aktualnya, namun selanjutnya karena ada fluktuasi dari data aktual sehingga menyebabkan nilai ramalannya tidak terlalu mengikuti pola data aktual dengan tingkat akurasi peramalan menggunakan MAPE sebesar 7,98%. Selanjutnya hasil peramalan rata-rata harga minyak goreng klaster 1 untuk 6 periode ke depan, yaitu November 2022 sampai dengan April 2023 menunjukkan nilai-nilai yang relatif konstan walaupun terdapat perubahan-perubahan yang kecil yaitu sekitar Rp 19.000, sehingga secara generalisasi maka objek-objek yang berada pada klaster 1 juga akan menghasilkan ramalan yang relatif konstan.

b. Peramalan model ARIMA (0,1,2)

Model ARIMA (0,1,2) untuk klaster 1 dapat dituliskan $Y(1)_t = Y(1)_{t-1} + e(1)_t - \hat{\theta}_1 e(1)_{t-1} - \hat{\theta}_2 e(1)_{t-2}$ dengan nilai koefisien untuk parameter berdasarkan Tabel 4.9 diperoleh

$$Y(1)_t = Y(1)_{t-1} + e(1)_t - 0,53e(1)_{t-1} - 0,28e(1)_{t-2}$$

Proses untuk menghitung nilai dari $\hat{Y}(1)_1$ dan $\hat{Y}(1)_2$ tidak memiliki informasi $Y(1)_0$ dan $e(1)_0$ maka akan diberikan contoh untuk mencari nilai dari $\hat{Y}(1)_3$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(1)_3 &= Y(1)_2 + e(1)_3 - 0,53e(1)_2 - 0,28e(1)_1 \\ &= 13.455 + e(1)_3 - ((0,53)(-4,50)) - ((0,28)(-17,93)) \\ &= 13.444 \text{ (Berdasarkan perhitungan software R)}\end{aligned}$$

Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(1)_{52}$, sehingga diperoleh hasil prediksi yang disajikan pada Tabel 4.19

Tabel 4.19 Prediksi Data Harga Minyak Goreng Klaster 1 Model ARIMA (0,1,2)

t	$Y(1)_t$ (Rupiah/Liter)	$\hat{Y}(1)_t$ (Rupiah/Liter)
1	13.476	13.480
2	13.455	13.472
3	13.418	13.444
4	13.345	13.399
$\vdots$	$\vdots$	$\vdots$
52	19.853	19.977

Sumber: Lampiran 31

Selanjutnya adalah menghitung peramalan untuk data *out-sample*. Adapun sebelum dilakukan perhitungan, terlebih dahulu dilakukan pembangkitan secara *random* data *error* sebanyak 67 data yang bersifat *white noise* dan berdistribusi $N(58,38; 250,07)$ berdasarkan data aktual dan prediksi dari data *in-sample* pada Tabel 4.19. Hasil bangkitan data dapat dilihat pada Lampiran 29. Sebagai contoh akan dicari nilai dari $\hat{Y}(1)_{53}$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(1)_{53} &= Y(1)_{52} + e(1)_{53} - 0,53e(1)_{52} - 0,28e(1)_{51} \\ &= 19.853 + e(1)_{53} - 0,53(-124,90) - 0,28(379,09) \\ &= 19.813\end{aligned}$$

Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(1)_{61}$, sehingga diperoleh hasil peramalan data *out-sample* yang disajikan pada Tabel 4.20.

Tabel 4.20 Peramalan Data *Out-Sample* Harga Minyak Goreng Klaster 1 Model ARIMA (0,1,2)

t	$Y(1)_t$ (Rupiah/Liter)	$\hat{Y}(1)_t$ (Rupiah/Liter)
53	19.555	19.813
54	18.155	19.651
55	22.589	19.549
56	23.063	19.499
57	22.271	19.471
58	21.642	19.535
59	19.582	19.342
60	18.897	19.096
61	18.334	18.824

Sumber: Lampiran 31

Berdasarkan Tabel 4.17 terdapat dua informasi yaitu data aktual dan data hasil peramalan. Perhitungan dimulai dengan prediksi ($t = 1$ hingga $t = 52$) kemudian dilanjutkan dengan peramalan sebanyak data *out-sample* ($t = 53$ hingga $t = 61$) menggunakan model ARIMA (0,1,2). Berdasarkan hasil prediksi dan peramalan dari model ARIMA (0,1,2) yang dapat dilihat pada Lampiran 31 diperoleh akurasi peramalan yang disajikan pada Tabel 4.21

Tabel 4.21 Nilai Akurasi Model ARIMA (0,1,2) Klaster 1

Data	Jenis Akurasi	Nilai Akurasi
<i>In-sample</i>	AIC	-3.089,91
	MAPE	0,87%
<i>Out-sample</i>	MAPE	7,30%

Sumber: Lampiran 22 & Lampiran 37

Langkah selanjutnya akan dilakukan peramalan 6 periode ke depan untuk harga minyak goreng klaster 1 menggunakan model ARIMA (0,1,2). Sebagai contoh akan dicari nilai dari $\hat{Y}(1)_{62}$, maka perhitungannya adalah

$$\begin{aligned}
 \hat{Y}(1)_{62} &= \hat{Y}(1)_{61} + e(1)_{62} - 0,53e(1)_{61} - 0,28e(1)_{60} \\
 &= 18.824 + e(1)_{62} - 0,53(432,99) - 0,28(388,47) \\
 &= 18.945
 \end{aligned}$$

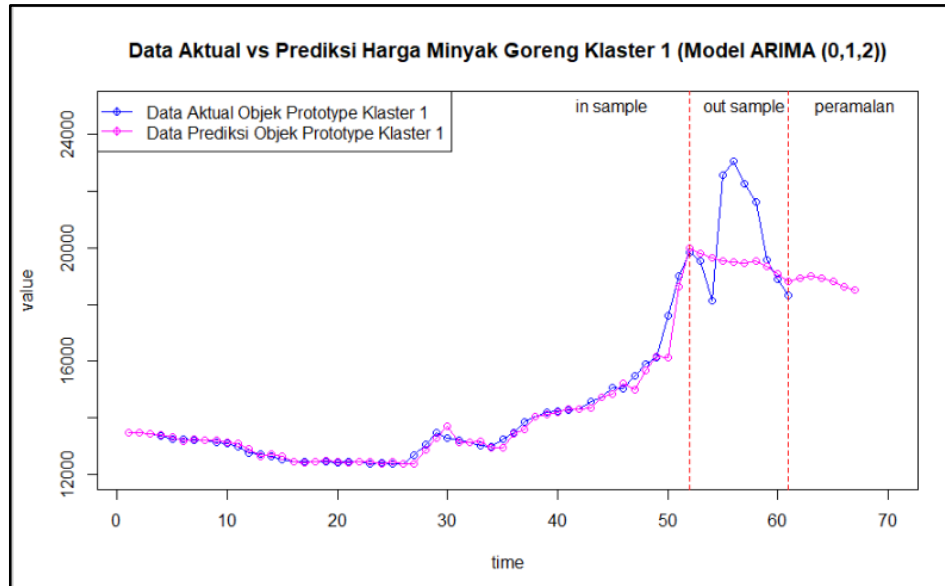
Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(1)_{67}$, sehingga diperoleh hasil peramalan yang disajikan pada Tabel 4.22

Tabel 4.22 Peramalan Data Harga Minyak Goreng Klaster 1 untuk Model ARIMA (0,1,2)

t	Bulan	$\hat{Y}(1)_t$ (Rupiah/Liter)
62	November 2022	18.945
63	Desember 2022	18.994
64	Januari 2023	18.944
65	Februari 2023	18.822
66	Maret 2023	18.616
67	April 2023	18.523

Sumber: Lampiran 31

Selanjutnya grafik data aktual dan peramalan untuk klaster 1 ($Y(1)_t$) untuk data *in-sample*, *out-sample*, dan hasil ramalan model ARIMA (0,1,2) ditampilkan pada Gambar 4.20



Gambar 4.20 Data aktual vs prediksi klaster 1 untuk model ARIMA (0,1,2)

Berdasarkan Gambar 4.20 dapat dilihat bahwa grafik runtun waktu untuk data *in-sample* nilai prediksinya hampir mengikuti pola data aktual dengan tingkat akurasi peramalan menggunakan MAPE sebesar 0,87%. Kemudian untuk data *out-sample* pada periode pertama mengikuti pola data aktualnya, namun

selanjutnya karena ada fluktuasi dari data aktual sehingga menyebabkan nilai ramalannya tidak terlalu mengikuti pola data aktual dengan tingkat akurasi peramalan menggunakan MAPE sebesar 7,30%. Selanjutnya hasil peramalan rata-rata harga minyak goreng klaster 1 untuk 6 periode ke depan, yaitu November 2022 sampai dengan April 2023 menunjukkan nilai-nilai yang relatif menurun hingga sebesar Rp 18.523, sehingga secara generalisasi maka objek-objek yang berada pada klaster 1 juga akan menghasilkan ramalan yang relatif menurun.

c. Peramalan model ARIMA (1,1,0)

Model ARIMA (1,1,0) dapat dituliskan

$Y(1)_t = Y(1)_{t-1} + \hat{\phi}_1 Y(1)_{t-1} - \hat{\phi}_1 Y(1)_{t-2} + e(1)_t$ dengan nilai koefisien untuk parameter berdasarkan Tabel 4.9 diperoleh

$$Y(1)_t = Y(1)_{t-1} + 0,59Y(1)_{t-1} - 0,59Y(1)_{t-2} + e(1)_t$$

Proses untuk menghitung nilai dari $\hat{Y}(1)_1$ dan $\hat{Y}(1)_2$ tidak memiliki informasi $Y(1)_0$ maka akan diberikan contoh untuk mencari nilai dari $\hat{Y}(1)_3$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(1)_3 &= Y(1)_2 + 0,59Y(1)_2 - 0,59Y(1)_1 + e(1)_3 \\ &= 13.455 + 0,59(13.455) - 0,59(13.476) + e(1)_3 \\ &= 13.442 \text{ (Berdasarkan perhitungan software R)}\end{aligned}$$

Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(1)_{52}$, sehingga diperoleh hasil prediksi yang disajikan pada Tabel 4.23

Tabel 4.23 Prediksi Data Harga Minyak Goreng Klaster 1 Model ARIMA (1,1,0)

t	$Y(1)_t$ (Rupiah/Liter)	$\hat{Y}(1)_t$ (Rupiah/Liter)
1	13.476	13.480
2	13.455	13.471
3	13.418	13.442
4	13.345	13.396
$\vdots$	$\vdots$	$\vdots$
52	19.853	20.080

Sumber: Lampiran 32

Selanjutnya adalah menghitung peramalan untuk data *out-sample*. Sebagai contoh akan dicari nilai dari $\hat{Y}(1)_{53}$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(1)_{53} &= Y(1)_{52} + 0,59Y(1)_{52} - 0,59Y(1)_{51} + e(1)_{53} \\ &= 19.853 + 0,59(19.853) - 0,59(19.003) + e(1)_{53} \\ &= 20.433 \text{ (Berdasarkan perhitungan software R)}\end{aligned}$$

Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(1)_{61}$, sehingga diperoleh hasil peramalan data *out-sample* yang disajikan pada Tabel 4.24

Tabel 4.24 Peramalan Data *Out-Sample* Harga Minyak Goreng Klaster 1 Model ARIMA (1,1,0)

t	$Y(1)_t$ (Rupiah/Liter)	$\hat{Y}(1)_t$ (Rupiah/Liter)
53	19.555	20.433
54	18.155	20.811
55	22.589	21.047
56	23.063	21.192
57	22.271	21.280
58	21.642	21.333
59	19.582	21.364
60	18.897	21.382
61	18.334	21.393

Sumber: Lampiran 32

Berdasarkan Tabel 4.24 terdapat dua informasi yaitu data aktual dan data hasil peramalan. Perhitungan dimulai dengan prediksi ($t = 1$ hingga $t = 52$) kemudian dilanjutkan dengan peramalan sebanyak data *out-sample* ($t = 53$ hingga $t = 61$) menggunakan model ARIMA (1,1,0). Berdasarkan hasil prediksi dan peramalan dari model ARIMA (1,1,0) yang dapat dilihat pada Lampiran 32 diperoleh akurasi peramalan yang disajikan pada Tabel 4.25

Tabel 4.25 Nilai Akurasi Model ARIMA (1,1,0) Klaster 1

Data	Jenis Akurasi	Nilai Akurasi
<i>In-sample</i>	AIC	-3.093,77
	MAPE	0,85%
<i>Out-sample</i>	MAPE	8,76%

Sumber: Lampiran 22 & Lampiran 37

Langkah selanjutnya akan dilakukan peramalan 6 periode ke depan untuk harga minyak goreng klaster 1 menggunakan model ARIMA (1,1,0). Sebagai contoh akan dicari nilai dari $\hat{Y}(1)_{62}$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(1)_{62} &= \hat{Y}(1)_{61} + 0,59\hat{Y}(1)_{61} - 0,59\hat{Y}(1)_{60} + e(1)_{62} \\ &= 18.334 + 0,59(18.334) - 0,59(18.897) + e(1)_{62} \\ &= 21.400 \text{ (Berdasarkan perhitungan software R)}\end{aligned}$$

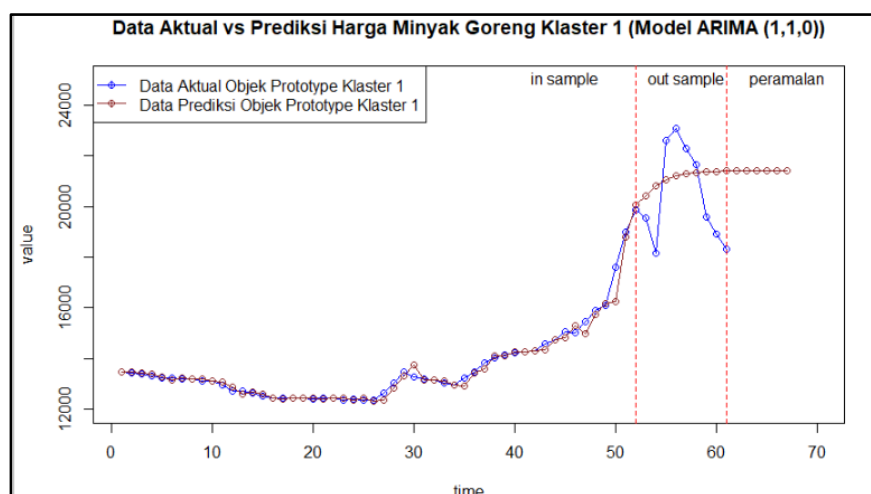
Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(1)_{67}$, sehingga diperoleh hasil peramalan yang disajikan pada Tabel 4.26

Tabel 4.26 Peramalan Data Harga Minyak Goreng Klaster 1 untuk Model ARIMA (1,1,0)

t	Bulan	$\hat{Y}(1)_t$ (Rupiah/Liter)
62	November 2022	21.400
63	Desember 2022	21.404
64	Januari 2023	21.406
65	Februari 2023	21.407
66	Maret 2023	21.408
67	April 2023	21.409

Sumber: Lampiran 32

Selanjutnya grafik data aktual dan peramalan untuk klaster 1 ($Y(1)_t$) untuk data *in-sample*, *out-sample*, dan hasil ramalan model ARIMA (1,1,0) ditampilkan pada Gambar 4.21



Gambar 4.21 Data aktual vs prediksi klaster 1 untuk model ARIMA (1,1,0)

Berdasarkan Gambar 4.21 dapat dilihat bahwa grafik runtun waktu untuk data *in-sample* nilai prediksinya hampir mengikuti pola data aktual dengan tingkat akurasi peramalan menggunakan MAPE sebesar 0,85%. Kemudian untuk data *out-sample* pada periode pertama mengikuti pola data aktualnya, namun selanjutnya karena ada fluktuasi dari data aktual sehingga menyebabkan nilai ramalannya tidak terlalu mengikuti pola data aktual dengan tingkat akurasi peramalan menggunakan MAPE sebesar 8,76%. Selanjutnya hasil peramalan rata-rata harga minyak goreng klaster 1 untuk 6 periode ke depan, yaitu November 2022 sampai dengan April 2023 menunjukkan nilai-nilai yang relatif naik mencapai Rp 21.409 walaupun tidak terlalu signifikan, sehingga secara generalisasi maka objek-objek yang berada pada klaster 1 juga akan menghasilkan ramalan yang relatif naik.

Selanjutnya dilakukan peramalan untuk tiap dengan menggunakan model ARIMA dari objek *prototype* klaster 1. Model yang digunakan adalah model ARIMA yang memiliki nilai akurasi peramalan minimum. Rekapitulasi nilai akurasi peramalan dari ketiga model disajikan pada Tabel 4.27.

Tabel 4.27 Rekapitulasi Nilai Akurasi Peramalan Seluruh Model Klaster 1

Model	AIC	MAPE (%)	
		<i>In-sample</i>	<i>Out-sample</i>
ARIMA (0,1,1)	-3.087,35	1,00	7,98
ARIMA (0,1,2)	-3.089,91	0,87	7,30
ARIMA (1,1,0)	-3.093,77	0,85	8,76

Berdasarkan Tabel 4.27 dapat disimpulkan bahwa model ARIMA (1,1,0) yang akan digunakan dalam meramalkan objek runtun waktu yang bergabung di klaster 1, dikarenakan model tersebut memiliki nilai akurasi peramalan yang paling minimum diantara ketiga model. Grafik ramalan dari semua Provinsi yang bergabung di klaster 1 dapat dilihat pada Lampiran 31. Evaluasi peramalan dari semua Provinsi disajikan pada Tabel 4.28.

Tabel 4.28 Nilai Akurasi Peramalan Tingkat Individu (Provinsi) Klaster 1

Provinsi	MAPE (%)	
	<i>In-sample</i>	<i>Out-sample</i>
Aceh	1,38	14,08
Sumatera Utara	1,08	4,58
Sumatera Barat	1,77	11,29
Riau	1,55	9,82
Jambi	1,43	10,31
Sumatera Selatan	1,48	10,54
Lampung	1,45	9,45
Banten	1,82	8,50
Jawa Barat	1,28	10,92
DKI Jakarta	1,55	7,31
Jawa Tengah	1,38	10,42
DI Yogyakarta	1,50	7,90
Jawa Timur	1,36	12,23
Bali	1,18	6,79
Nusa Tenggara Barat	1,14	9,01
Kalimantan Barat	1,71	7,90
Kalimantan Selatan	1,21	12,23
Gorontalo	2,45	6,79
Sulawesi Selatan	1,24	9,01

Sumber: Lampiran 39

Berdasarkan Tabel 4.28, dapat diketahui bahwa dengan menggunakan model peramalan ARIMA (1,1,0) dari objek *prototype* klaster 1 untuk seluruh Provinsi yang bergabung di klaster 1 memiliki nilai MAPE yang relatif kecil baik untuk *in-sample* yang memiliki nilai prediksi yang sangat akurat maupun *out-sample* yang memiliki nilai prediksi yang baik. Namun dari informasi tersebut dapat diketahui bahwa peramalan untuk semua Provinsi yang bergabung di klaster 1 dapat menggunakan model peramalan dari objek *prototype* saja tanpa harus memodelkan peramalan untuk masing-masing Provinsi.

2. Klaster 2

Berdasarkan Tabel 4.14, dapat diketahui bahwa pada klaster 2 terdapat 2 model layak yang dapat digunakan untuk peramalan. Kedua model layak yaitu model ARIMA (0,1,1) dan ARIMA (1,1,0) akan diramalkan pada tingkat klaster

dan memilih salah satu dari kedua model tersebut yang memiliki ukuran akurasi peramalan terkecil untuk meramalkan anggota-anggota yang bergabung dalam satu klaster. Ukuran akurasi peramalan yang digunakan adalah AIC dan MAPE.

a. Peramalan model ARIMA (0,1,1)

Model ARIMA (0,1,1) dapat dituliskan $Y(2)_t = Y(2)_{t-1} + e(2)_t - \hat{\theta}_1 e(2)_{t-1}$

dengan nilai koefisien untuk parameter berdasarkan Tabel 4.12 diperoleh

$$Y(2)_t = Y(2)_{t-1} + e(2)_t - 0,22e(2)_{t-1}$$

Proses untuk menghitung nilai dari $\hat{Y}(2)_1$ tidak memiliki informasi $Y(2)_0$ maka akan diberikan contoh untuk mencari nilai dari $\hat{Y}(2)_2$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(2)_2 &= Y(2)_1 + e(2)_2 - 0,22e(2)_1 \\ &= 14.600 + e(2)_2 - ((0,22)(-4,01)) \\ &= 13.597 \text{ (Berdasarkan perhitungan software R)}\end{aligned}$$

Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(2)_{52}$, sehingga diperoleh hasil prediksi yang disajikan pada Tabel 4.29

Tabel 4.29 Prediksi Data Harga Minyak Goreng Klaster 2 Model ARIMA (0,1,1)

t	$Y(1)_t$ (Rupiah/Liter)	$\hat{Y}(1)_t$ (Rupiah/Liter)
1	14.600	14.604
2	14.513	14.597
3	14.530	14.494
4	14.417	14.537
$\vdots$	$\vdots$	$\vdots$
52	20.157	19.074

Sumber: Lampiran 33

Selanjutnya adalah menghitung peramalan untuk data *out-sample*. Adapun sebelum dilakukan perhitungan, terlebih dahulu dilakukan pembangkitan secara *random* data *error* sebanyak 67 data yang bersifat *white noise* dan berdistribusi $N(88,32; 305,77)$ berdasarkan data aktual dan prediksi dari data *in-sample* pada

Tabel 4.29. Hasil bangkitan data dapat dilihat pada Lampiran 29. Sebagai contoh akan dicari nilai dari $\hat{Y}(2)_{53}$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(2)_{53} &= Y(2)_{52} + e(2)_{53} - 0,22e(2)_{52} \\ &= 20.157 + e(1)_{53} - ((0,22)(1.082,64)) \\ &= 19.918\end{aligned}$$

Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(2)_{61}$, sehingga diperoleh hasil peramalan data *out-sample* yang disajikan pada Tabel 4.30.

Tabel 4.30 Peramalan Data *Out-Sample* Harga Minyak Goreng Klaster 2 Model ARIMA (0,1,1)

t	$Y(2)_t$ (Rupiah/Liter)	$\hat{Y}(2)_t$ (Rupiah/Liter)
53	19.567	19.918
54	18.710	19.868
55	25.670	19.865
56	26.213	19.898
57	25.853	19.940
58	24.893	19.916
59	22.580	19.853
60	21.340	19.814
61	20.747	19.676

Sumber: Lampiran 33

Berdasarkan Tabel 4.30 terdapat dua informasi yaitu data aktual dan data hasil peramalan. Perhitungan dimulai dengan prediksi ($t = 1$ hingga $t = 52$) kemudian dilanjutkan dengan peramalan sebanyak data *out-sample* ($t = 53$ hingga $t = 61$) menggunakan model ARIMA (0,1,1). Berdasarkan hasil prediksi dan peramalan dari model ARIMA (0,1,1) yang dapat dilihat pada Lampiran 33 diperoleh akurasi peramalan yang disajikan pada Tabel 4.31

Tabel 4.31 Nilai Akurasi Model ARIMA (0,1,1) Klaster 2

Data	Jenis Akurasi	Nilai Akurasi
<i>In-sample</i>	AIC	-3.723,29
	MAPE	0,95%
<i>Out-sample</i>	MAPE	13,54%

Sumber: Lampiran 26 & Lampiran 38

Langkah selanjutnya akan dilakukan peramalan 6 periode ke depan untuk harga minyak goreng klaster 2 menggunakan model ARIMA (0,1,1). Sebagai contoh akan dicari nilai dari $\hat{Y}(2)_{62}$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(2)_{62} &= \hat{Y}(2)_{61} + e(2)_{62} - 0,22e(2)_{61} \\ &= 19.676 + e(2)_{62} - 0,22(119,23) \\ &= 19.650\end{aligned}$$

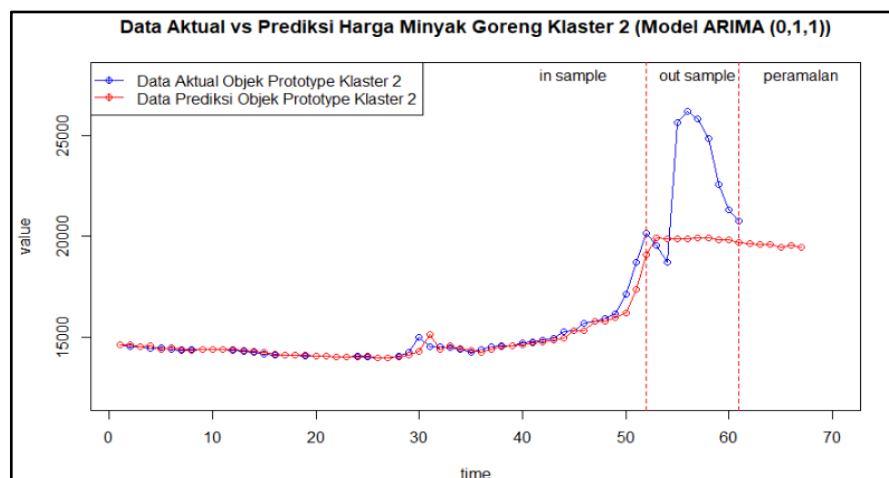
Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(2)_{67}$, sehingga diperoleh hasil peramalan yang disajikan pada Tabel 4.32

Tabel 4.32 Peramalan Data Harga Minyak Goreng Klaster 2 untuk Model ARIMA (0,1,1)

t	Bulan	$\hat{Y}(1)_t$ (Peramalan) (Rupiah)
62	November 2022	19.650
63	Desember 2022	19.601
64	Januari 2023	19.604
65	Februari 2023	19.453
66	Maret 2023	19.559
67	April 2023	19.477

Sumber: Lampiran 33

Selanjutnya grafik data aktual dan peramalan untuk klaster 2 ($Y(2)_t$) untuk data *in-sample*, *out-sample*, dan hasil ramalan model ARIMA (0,1,1) ditampilkan pada Gambar 4.22



Gambar 4.22 Data aktual vs prediksi klaster 2 untuk model ARIMA (0,1,1)

Berdasarkan Gambar 4.22 dapat dilihat bahwa grafik runtun waktu untuk data *in-sample* nilai prediksinya hampir mengikuti pola data aktual dengan tingkat akurasi peramalan menggunakan MAPE sebesar 0,95%. Kemudian untuk data *out-sample* nilai ramalannya tidak terlalu mengikuti pola data aktual dengan tingkat akurasi peramalan menggunakan MAPE sebesar 13,54%. Selanjutnya hasil peramalan rata-rata harga minyak goreng klaster 2 untuk 6 periode ke depan, yaitu November 2022 sampai dengan April 2023 menunjukkan nilai-nilai yang relatif konstan sekitar Rp 19.000, sehingga secara generalisasi maka objek-objek yang berada pada klaster 2 juga akan menghasilkan ramalan yang relatif konstan.

b. Peramalan model ARIMA (1,1,0)

Model ARIMA (1,1,0) dapat dituliskan $Y(2)_t = Y(2)_{t-1} + \hat{\phi}_1 Y(2)_{t-1} - \hat{\phi}_1 Y(2)_{t-2} + e(2)_t$ dengan nilai koefisien untuk parameter berdasarkan Tabel 4.12 diperoleh

$$Y(2)_t = Y(2)_{t-1} + 0,34Y(2)_{t-1} - 0,34Y(2)_{t-2} + e(2)_t$$

Proses untuk menghitung nilai dari $\hat{Y}(2)_1$ dan $\hat{Y}(2)_2$ tidak memiliki informasi $Y(2)_0$ maka akan diberikan contoh untuk mencari nilai dari $\hat{Y}(2)_3$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(2)_3 &= Y(2)_2 + 0,34Y(2)_2 - 0,34Y(2)_1 + e(2)_3 \\ &= 14.513 + 0,34(14.513) - 0,34(14.600) + e(1)_3 \\ &= 14.483 \text{ (Berdasarkan perhitungan software R)}\end{aligned}$$

Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(2)_{52}$, sehingga diperoleh hasil prediksi yang disajikan pada Tabel 4.33

Tabel 4.33 Prediksi Data Harga Minyak Goreng Klaster 2 Model ARIMA (1,1,0)

t	$Y(1)_t$ (Rupiah/Liter)	$\hat{Y}(1)_t$ (Rupiah/Liter)
1	14.600	14.604
2	14.513	14.594
3	14.530	14.483
$\vdots$	$\vdots$	$\vdots$
52	20.157	19.437

Sumber: Lampiran 34

Selanjutnya adalah menghitung peramalan untuk data *out-sample*. Sebagai contoh akan dicari nilai dari $\hat{Y}(2)_{53}$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(2)_{53} &= Y(2)_{52} + 0,34Y(2)_{52} - 0,34Y(2)_{51} + e(2)_t \\ &= 20.157 + 0,34(18.697) - 0,34(17.113) + e(2)_t \\ &= 20.806 \text{ (Berdasarkan perhitungan software R)}\end{aligned}$$

Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(2)_{61}$, sehingga diperoleh hasil peramalan data *out-sample* yang disajikan pada Tabel 4.34.

Tabel 4.34 Peramalan Data *Out-Sample* Harga Minyak Goreng Klaster 2 Model ARIMA (1,1,0)

t	$Y(2)_t$ (Rupiah/Liter)	$\hat{Y}(2)_t$ (Rupiah/Liter)
53	19.567	20.806
54	18.710	21.056
55	25.670	21.146
56	26.213	21.177
57	25.853	21.188
58	24.893	21.192
59	22.580	21.193
60	21.340	21.194
61	20.747	21.194

Sumber: Lampiran 34

Berdasarkan Tabel 4.34 terdapat dua informasi yaitu data aktual dan data hasil peramalan. Perhitungan dimulai dengan prediksi ($t = 1$ hingga $t = 52$) kemudian dilanjutkan dengan peramalan sebanyak data *out-sample* ($t = 53$ hingga $t = 61$) menggunakan model ARIMA (1,1,0). Berdasarkan hasil prediksi dan peramalan dari model ARIMA (1,1,0) yang dapat dilihat pada Lampiran 34 diperoleh akurasi peramalan yang disajikan pada Tabel 4.35.

Tabel 4.35 Nilai Akurasi Model ARIMA (1,1,0) Klaster 2

Data	Jenis Akurasi	Nilai Akurasi
<i>In-sample</i>	AIC	-3.725,53
	MAPE	0,89%
<i>Out-sample</i>	MAPE	10,84%

Sumber: Lampiran 26 & Lampiran 38

Langkah selanjutnya akan dilakukan peramalan 6 periode ke depan untuk harga minyak goreng klaster 1 menggunakan model ARIMA (1,1,0). Sebagai contoh akan dicari nilai dari $\hat{Y}(2)_{62}$, maka perhitungannya adalah

$$\begin{aligned}\hat{Y}(2)_{62} &= \hat{Y}(2)_{61} + 0,34\hat{Y}(2)_{61} - 0,34\hat{Y}(2)_{60} + e(2)_t \\ &= 21.194 + 0,34(21.994) - 0,34(21.994) + e(2)_t \\ &= 21.194 \text{ (Berdasarkan perhitungan software R)}\end{aligned}$$

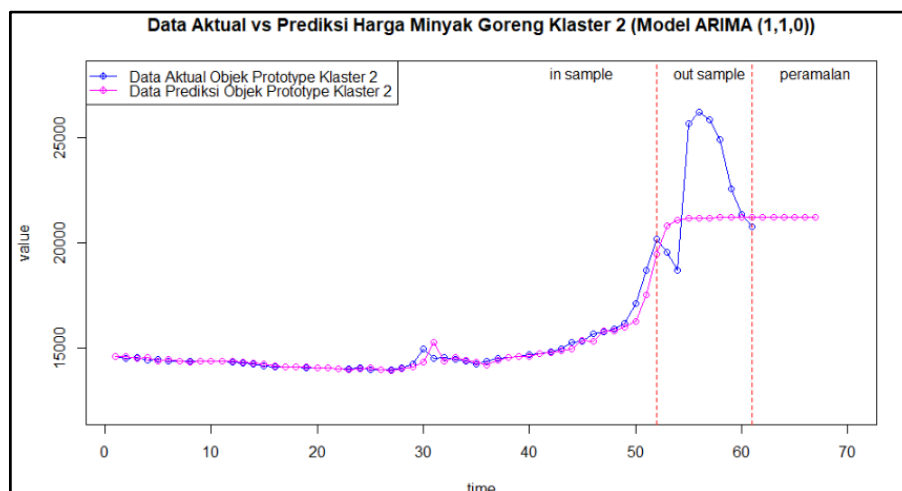
Perhitungan terus dilanjutkan dengan cara yang sama hingga $\hat{Y}(2)_{67}$, sehingga diperoleh hasil peramalan yang disajikan pada Tabel 4.36.

Tabel 4.36 Peramalan Data Harga Minyak Goreng Klaster 2 untuk Model ARIMA (1,1,0)

t	Bulan	$\hat{Y}(2)_t$ (Rupiah/Liter)
62	November 2022	21.194
63	Desember 2022	21.194
64	Januari 2023	21.194
65	Februari 2023	21.194
66	Maret 2023	21.194
67	April 2023	21.194

Sumber: Lampiran 34

Selanjutnya grafik data aktual dan peramalan untuk klaster 2 ($Y(2)_t$) untuk data *in-sample*, *out-sample*, dan hasil ramalan model ARIMA (1,1,0) ditampilkan pada Gambar 4.23.



Gambar 4.23 Data aktual vs prediksi klaster 2 untuk model ARIMA (1,1,0)

Berdasarkan Gambar 4.23 dapat dilihat bahwa grafik runtun waktu untuk data *in-sample* nilai prediksinya hampir mengikuti pola data aktual dengan tingkat akurasi peramalan menggunakan MAPE sebesar 0,89%. Kemudian untuk data *out-sample* pada periode pertama mengikuti pola data aktualnya, namun selanjutnya karena ada fluktuasi dari data aktual sehingga menyebabkan nilai ramalannya tidak terlalu mengikuti pola data aktual dengan tingkat akurasi peramalan menggunakan MAPE sebesar 10,84%. Selanjutnya hasil peramalan rata-rata harga minyak goreng klaster 2 untuk 6 periode ke depan, yaitu November 2022 sampai dengan April 2023 menunjukkan nilai-nilai yang relatif naik mencapai Rp 21.194 walaupun setelahnya menunjukkan nilai-nilai yang konstan, sehingga secara generalisasi maka objek-objek yang berada pada klaster 2 juga akan menghasilkan ramalan yang relatif naik dan setelahnya menunjukkan nilai-nilai yang konstan.

Selanjutnya akan dilakukan peramalan untuk tingkat individu (Provinsi) dengan menggunakan model ARIMA dari objek *prototype* klaster 2. Model yang digunakan adalah model ARIMA yang memiliki nilai akurasi peramalan minimum. Rekapitulasi nilai akurasi peramalan dari ketiga model baik dari data *in-sample* maupun dari data *out-sample* disajikan pada Tabel 4.37.

Tabel 4.37 Rekapitulasi Nilai Akurasi Peramalan Seluruh Model Klaster 2

Model	AIC	MAPE (%)	
		<i>In-sample</i>	<i>Out-sample</i>
ARIMA (0,1,1)	-3.723,29	0,95	13,54
ARIMA (1,1,0)	-3.725,53	0,89	10,84

Berdasarkan Tabel 4.37 dapat disimpulkan bahwa model ARIMA (1,1,0) yang akan digunakan dalam meramalkan objek runtun waktu yang bergabung di klaster 2, dikarenakan model tersebut memiliki nilai akurasi peramalan yang paling minimum diantara kedua model. Grafik ramalan dari semua objek yang bergabung di klaster 2 dapat dilihat pada Lampiran 35, sehingga evaluasi peramalan menggunakan MAPE dari seluruh objek yang bergabung di klaster 2 dengan menggunakan model dari objek *prototype* terbaik yaitu ARIMA (1,1,0) disajikan pada Tabel 4.38.

Tabel 4.38 Nilai Akurasi Peramalan Tingkat Individu (Provinsi) Klaster 2

Provinsi	MAPE (%)	
	<i>In-sample</i>	<i>Out-sample</i>
Kepulauan Riau	1,51	15,45
Bengkulu	1,37	13,10
Kepulauan Bangka Belitung	1,54	17,49
Nusa Tenggara Timur	0,88	14,75
Kalimantan Tengah	1,15	7,37
Kalimantan Timur	1,85	10,81
Kalimantan Utara	1,32	7,51
Sulawesi Tenggara	1,02	27,50
Sulawesi Tengah	1,72	12,87
Sulawesi Utara	1,26	6,07
Sulawesi Barat	1,65	11,43
Maluku	1,06	11,65
Maluku Utara	1,77	37,33
Papua	1,85	14,90
Papua Barat	4,11	19,82

Sumber: Lampiran 40

Berdasarkan Tabel 4.38, dapat diketahui bahwa dengan menggunakan model peramalan ARIMA (1,1,0) dari objek *prototype* klaster 2 untuk seluruh Provinsi yang bergabung di klaster 2 memiliki nilai MAPE yang relatif kecil baik untuk *in-sample* yang memiliki nilai prediksi yang sangat akurat maupun *out-sample* yang memiliki nilai prediksi yang baik walaupun terdapat beberapa Provinsi yang memiliki nilai akurasi prediksi yang cenderung besar. Namun dari informasi tersebut dapat diketahui bahwa peramalan untuk semua Provinsi yang bergabung di klaster 2 dapat menggunakan model peramalan dari objek *prototype* saja tanpa harus memodelkan peramalan untuk masing-masing Provinsi.

BAB 5

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil analisis data dan pembahasan, maka diperoleh kesimpulan sebagai berikut:

1. Nilai K optimal dalam proses pengelompokan data harga minyak goreng di Indonesia menggunakan algoritma *K-Medoids* adalah 2 klaster dengan nilai koefisien *silhouette* yang diperoleh sebesar 0,19.
2. Hasil pengelompokan data harga minyak goreng di Indonesia menggunakan algoritma *K-Medoids* dengan jarak *dynamic time warping* yaitu terdapat 19 Provinsi yang bergabung pada klaster 1 di mana klaster dengan harga minyak goreng dibawah klaster 2 dan terdapat 15 Provinsi yang bergabung pada klaster 2 di mana klaster dengan harga minyak goreng tertinggi.
3. Model ARIMA terbaik untuk setiap klaster yang terbentuk dari data harga minyak goreng di Indonesia adalah sebagai berikut:
 - a. Klaster 1
Model ARIMA terbaik yang diperoleh dari klaster 1 adalah ARIMA (1,1,0). Persamaan model tersebut dapat ditulis sebagai berikut:
$$Y(1)_t = Y(1)_{t-1} + 0,59Y(1)_{t-1} - 0,59Y(1)_{t-2} + e(1)_t$$
 - b. Klaster 2
Model ARIMA terbaik yang diperoleh dari klaster 2 adalah ARIMA (1,1,0). Persamaan model tersebut dapat ditulis sebagai berikut:
$$Y(2)_t = Y(2)_{t-1} + 0,34Y(2)_{t-1} - 0,34Y(2)_{t-2} + e(2)_t$$
4. Akurasi peramalan dari data harga minyak goreng setiap Provinsi di Indonesia dengan menggunakan model ARIMA terbaik untuk klaster 1 cenderung memberikan hasil MAPE yang sangat akurat baik untuk *in-sample* (dengan rata-rata MAPE sebesar 1,47%) maupun *out-sample* (dengan rata-rata MAPE sebesar 9,42%), Akurasi peramalan klaster 2 juga

cenderung memberikan hasil MAPE yang sangat akurat untuk *in-sample* (dengan rata-rata MAPE sebesar 1,60%) dan MAPE yang akurat untuk *out-sample* (dengan rata-rata MAPE sebesar 15,20%). Beberapa Provinsi memiliki MAPE yang cenderung besar. Namun dari informasi tersebut dapat diketahui bahwa peramalan untuk semua Provinsi yang bergabung di setiap klaster dapat menggunakan model peramalan dari objek *prototype* saja tanpa harus memodelkan peramalan untuk masing-masing Provinsi,

5.2 Saran

Berdasarkan hasil analisis menggunakan *K-medoids Time Series Clustering* saran peneliti untuk penelitian selanjutnya mengenai metode *time series clustering* adalah sebagai berikut:

1. Koefisien *silhouette* yang dihasilkan untuk menentukan *K* optimal tergolong relatif kecil sehingga untuk penelitian selanjutnya mengenai metode *time series clustering* dapat menggunakan optimasi *K-Medoids Time Series Clustering* dengan *Principle Component Analysis* (PCA).
2. Menggunakan ukuran pemusatan lainnya untuk menentukan objek *prototype* seperti median atau modus.
3. Menggunakan algoritma pengelompokan lainnya seperti *C-Means* atau *Fuzzy K-Medoids*.
4. Menggunakan model peramalan lainnya dengan mempertimbangkan pola data seperti *Seasonal Autoregressive Integrated Moving Average* (SARIMA) dan *Neural Network* (NN), atau model *hybrid* seperti *Autoregressive Integrated Moving Average-Neural Network* (ARIMA-NN) atau *Singular Spectrum Analysis-Neural Network* (SSA-NN).

DAFTAR PUSTAKA

- Andreanus, J., & Kurniawan, A. (2017). Sejarah, Teori Dasar dan Penerapan Reinforcement Learning: Sebuah Tinjauan Pustaka. *Jurnal Telematika*, 12(2).
- Astuti, D., Hartanti, D. Y., Nurhayanti, S. T., & Fransiska, H. (2022). Clustering and Forecasting of Covid-19 Data in Indonesia. *Jurnal Matematika, Statistika Dan Komputasi*, 18(3), 324–335.
- Aswi, & Sukarna. (2006). *Analisis Deret Waktu: Teori dan Aplikasi*. Makassar: Andira Publisher.
- BSN. (2013). *SNI 3741:2013. Minyak Goreng*. Jakarta: Badan Standarisasi Nasional.
- Chotirat, V. N., & Ratanamahatana, A. (2007). On Clustering Multimedia Time Series Data Using K-Means and Dynamic Time Warping. *2007 International Conference on Multimedia and Ubiquitous Engineering*.
- Dani, A. T. R., Wahyuningsih, S., & Rizki, N. A. (2019). Penerapan Hierarchical Clustering Metode Agglomerative pada Data Runtun Waktu. *Jambura Journal of Mathematics*, 1(2), 64–78.
- David, L. C. (1982). *Industrial and Business Forecasting Methods: A practical guide to exponential smoothing and curve fitting*. London: Butterworth Scientific.
- Guna, J., Stojmenova, E., Lugmayr, A., Humar, I., & Pogačnik, M. (2014). User identification approach based on simple gestures. *Multimedia Tools and Applications*, 71(1), 179–194.
- Han, J., & Kamber, M. (2006). *Data Mining: Concept and Techniques Second Edition*. San Fransisco.
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concept and Techniques Third Edition*. Waltham.
- Hanke, J. E., & Wichern, D. W. (2003). *Business Forecasting Eight Edition*. New Jersey: Pearson Prentice, Inc.
- Irnawati. (2021). *Proyeksi Ketersediaan dan Kebutuhan Minyak Goreng Indonesia Tahun 2021*. Jakarta: Repositori Publikasi Kementerian Pertanian Republik Indonesia.
- Kaufman, L., & Rousseeuw, P. J. (1990). *Finding Groups in Data An Introduction to Cluster Analysis*. New Jersey: John Wiley & Sons Inc Publication.

- Kemendag. (2021). *Profil Komoditas Minyak Goreng*. Jakarta: Kementerian Perdagangan Republik Indonesia.
- Kontan. (2022). CPO Naik Tajam Sejak 2020, Pengusaha Dukung Kebijakan Minyak Goreng Satu Harga. <https://industri.kontan.co.id/news/cpo-naik-tajam-sejak-2020-pengusaha-dukung-kebijakan-minyak-goreng-satu-harga>. Diakses pada 23 Januari 2023.
- Ljung, G. & Box, G. (1978). On a Measure of Lack of Fit in Time Series Models. *Biometrika*, 65(2), 297–303.
- Makridakis, S., Wheelwright, S. C., & McGee, V. E. (1999). *Metode dan Aplikasi Peramalan Jilid 1*. Jakarta: Binarupa Aksara.
- Novidianto, R., & Dani, A. T. R. (2020). Analisis Kluster Kasus Aktif COVID-19 Menurut Provinsi di Indonesia Berdasarkan Data Deret Waktu. *Jurnal Aplikasi Statistika & Komputasi Statistik*, 12, 15–24.
- Prasetyo, E. (2012). *Konsep dan Aplikasi menggunakan MATLAB*. Yogyakarta: Andi Offset.
- Ramdhani, M. A. (2014). *Manajemen Operasi*. Bandung: CV. Pustaka Setia.
- Salamah, M., Suhartono, & Wulandari, S. P. (2003). *Analisis Time Series*. Surabaya: Fakultas Matematika dan Ilmu Pengetahuan Alam ITS.
- Santosa, B. (2007). *Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.
- Santoso, P., Abijono, H., & Anggreini, N. L. (2021). Algoritma Supervised Learning dan Unsupervised Learning dalam Pengolahan Data. *Unira Malang*, 4(2).
- Sartika, R. A. D. (2009). Pengaruh Suhu dan Lama Proses Menggoreng (Deep Frying) Terhadap Pembentukan Asam Lemak Trans. *Makara Sains*, 13(1), 23–28.
- Sismi, M. Y. D. (2018). Perbandingan Prediksi Harga Saham PT.BRI, Tbk dengan Metode ARIMA dan Moving Average. *Prosiding Seminar Nasional Mahasiswa Unimus*, 1, 351–360.
- Suhartono. (2008). *Analisis Data Statistik dengan R*. Surabaya: Lab Statistika Komputasi ITS.
- Suyanto. (2019). *Data Mining untuk Klasifikasi dan Klusterisasi Data Edisi Revisi*. Bandung: Informatika.
- Turban, E. (2005). *Decision Support Systems and Intelligent Systems*. Yogyakarta: Andi Offset.

- Wei, W. W. S. (2006). *Time Series Analysis: Univariate and Multivariate Methods Second Edition*. New York: Pearson Education.
- Widarjono, A. (2007). *Ekonometrika: Teori dan Aplikasi untuk Ekonomi dan Bisnis*. Yogyakarta: Ekonisia.
- Wijaya, S. U., & Ngatini, N. N. (2020). Pengembangan Pemodelan Harga Beras di Wilayah Indonesia Bagian Barat dengan Pendekatan Clustering Time Series. *Limits: Journal of Mathematics and Its Applications*, 17(1), 51–66.
- Yudistira, N. (2021). Peran Big Data dan Deep Learning untuk Menyelesaikan Permasalahan Secara Komprehensif. *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, 11(2), 78–89.

LAMPIRAN

Lampiran 1. Data Harga Minyak Goreng pada 34 Provinsi di Indonesia

Provinsi	2017			2018				
	Oktober	November	Desember	Januari	Februari	Maret	April	Mei
Aceh	14.300	14.550	14.450	14.400	14.350	14.350	14.300	14.100
Sumatera Utara	13.500	13.450	13.500	13.400	13.250	13.250	13.150	13.200
Sumatera Barat	13.500	13.350	13.200	13.150	12.950	12.850	12.800	12.750
Riau	13.750	13.750	13.650	13.550	13.300	13.350	13.350	13.350
Kepulauan Riau	12.600	12.750	12.750	12.750	12.750	12.750	12.750	12.500
Jambi	13.100	13.200	13.250	12.600	12.600	12.650	12.700	12.700
Bengkulu	12.600	12.850	12.800	12.900	13.100	13.050	12.850	12.950
Sumatera Selatan	13.250	13.200	13.200	13.200	13.150	13.050	13.000	13.000
K. Bangka Belitung	12.850	12.800	12.800	13.150	13.250	13.050	13.150	13.100
Lampung	13.300	13.350	13.250	13.350	13.150	13.100	13.100	13.100
Banten	12.900	12.800	12.700	12.500	12.350	12.250	12.150	12.300
Jawa Barat	13.350	13.250	13.250	13.200	12.950	12.950	13.000	12.950
DKI Jakarta	13.700	13.550	13.450	13.500	13.450	13.450	13.450	13.450
Jawa Tengah	12.900	12.800	13.000	13.000	12.650	12.650	12.550	12.800
DI Yogyakarta	12.750	12.700	12.650	12.650	12.650	12.800	12.750	12.750
Jawa Timur	13.000	12.950	12.850	12.800	12.750	12.650	12.700	12.550
Bali	14.100	14.100	13.900	13.900	13.800	13.800	13.850	13.850
Nusa Tenggara Barat	13.450	13.450	13.400	13.250	13.200	13.250	13.250	13.300
Nusa Tenggara Timur	14.850	14.800	14.800	14.900	14.900	14.900	14.900	14.900
Kalimantan Barat	13.550	13.550	13.500	13.400	13.300	13.350	13.400	13.300
Kalimantan Selatan	13.500	13.500	13.450	13.400	13.350	13.350	13.350	13.350
Kalimantan Tengah	14.500	14.350	14.250	14.350	14.200	14.200	14.150	14.100
Kalimantan Timur	16.200	16.200	16.200	13.750	14.850	14.850	14.850	14.800
Kalimantan Utara	14.100	14.100	14.000	14.100	14.150	14.250	14.150	14.150
Gorontalo	14.750	14.750	14.900	14.850	14.750	14.850	14.650	14.650
Sulawesi Selatan	13.400	13.400	13.400	13.450	13.450	13.400	13.350	13.450
Sulawesi Tenggara	14.750	14.500	14.550	14.400	14.100	14.100	14.100	14.100
Sulawesi Tengah	14.100	13.900	14.050	14.050	14.200	13.250	13.400	14.450
Sulawesi Utara	14.450	14.500	14.400	14.450	14.350	14.400	14.450	14.450
Sulawesi Barat	13.850	13.750	13.800	13.900	13.650	13.550	12.950	13.650
Maluku	15.800	15.750	15.800	16.050	15.950	15.950	15.950	15.950
Maluku Utara	15.950	15.950	15.950	15.850	15.850	15.850	15.850	15.600
Papua	15.950	15.150	15.400	15.200	15.200	15.200	15.200	14.700
Papua Barat	16.450	16.350	16.400	16.450	16.250	16.250	16.500	16.200

Lampiran 1. Data Harga Minyak Goreng pada 34 Provinsi di Indonesia (lanjutan)

Provinsi	2018							2019
	Juni	Juli	Agustus	September	Oktober	November	Desember	Januari
Aceh	14.000	14.000	13.900	13.600	13.700	13.650	13.550	13.400
Sumatera Utara	13.100	13.150	13.050	12.600	12.400	12.400	12.350	12.250
Sumatera Barat	12.750	12.700	12.450	12.250	12.250	12.100	11.750	11.600
Riau	13.300	13.400	13.100	13.050	13.050	12.850	12.700	12.400
Kepulauan Riau	12.500	12.600	12.550	12.500	12.600	12.450	12.200	12.150
Jambi	12.700	12.650	12.600	12.400	11.900	11.800	11.650	11.650
Bengkulu	12.800	12.850	12.750	12.700	12.650	12.650	12.600	12.550
Sumatera Selatan	13.000	13.100	13.000	12.400	12.400	12.350	12.050	12.100
K. Bangka Belitung	13.150	13.050	13.050	13.050	12.900	12.800	12.650	12.450
Lampung	12.500	12.500	12.200	12.150	11.750	11.900	11.750	11.700
Banten	12.250	12.350	12.300	12.200	12.050	11.900	11.650	11.600
Jawa Barat	12.950	12.950	12.850	12.650	12.600	12.500	12.450	12.200
DKI Jakarta	13.450	13.450	13.450	13.300	13.300	13.300	13.050	12.850
Jawa Tengah	12.550	12.600	12.450	12.350	12.300	12.150	12.050	12.050
DI Yogyakarta	12.800	12.650	12.500	12.500	12.250	12.200	12.150	12.100
Jawa Timur	12.550	12.400	12.250	12.150	12.050	12.000	11.950	11.850
Bali	13.600	13.450	13.450	13.250	13.250	13.300	13.300	13.300
Nusa Tenggara Barat	13.250	13.150	13.200	13.150	13.150	13.050	13.050	13.250
Nusa Tenggara Timur	14.900	14.900	14.800	14.800	14.800	14.800	14.800	14.850
Kalimantan Barat	13.300	13.300	13.200	12.100	13.100	12.800	12.700	12.650
Kalimantan Selatan	13.250	13.250	13.250	13.000	12.900	12.900	12.800	12.700
Kalimantan Tengah	14.050	14.200	14.100	14.150	14.100	13.950	14.100	14.000
Kalimantan Timur	14.750	14.700	14.750	14.750	14.650	14.650	14.500	14.550
Kalimantan Utara	14.150	14.150	14.000	13.900	13.900	13.950	13.900	13.850
Gorontalo	14.800	14.650	14.250	14.300	14.400	14.350	14.150	14.150
Sulawesi Selatan	13.700	13.400	12.850	12.800	12.800	12.850	12.700	12.600
Sulawesi Tenggara	14.100	14.100	14.100	14.150	14.150	14.150	13.850	14.100
Sulawesi Tengah	14.000	14.000	13.950	14.150	14.050	14.300	14.100	13.950
Sulawesi Utara	14.450	14.350	14.300	14.300	14.350	14.200	14.250	14.050
Sulawesi Barat	13.650	13.650	14.300	13.650	13.600	13.350	13.200	13.200
Maluku	15.950	15.950	15.900	16.050	16.050	15.850	15.850	15.800
Maluku Utara	15.600	15.600	15.600	15.600	15.600	15.600	15.600	15.100
Papua	15.200	15.250	15.100	14.950	14.900	14.900	15.050	15.000
Papua Barat	16.200	16.200	16.200	16.200	15.750	15.950	15.600	15.750

Lampiran 1. Data Harga Minyak Goreng pada 34 Provinsi di Indonesia (lanjutan)

Provinsi	2019							
	Februari	Maret	April	Mei	Juni	Juli	Agustus	September
Aceh	13.550	13.550	13.450	13.400	13.400	13.350	13.350	13.200
Sumatera Utara	12.300	12.300	12.300	12.300	12.400	12.200	12.150	12.150
Sumatera Barat	11.700	11.750	11.750	11.750	11.750	11.700	11.550	11.700
Riau	12.300	12.300	12.400	12.350	12.350	12.300	12.200	12.100
Kepulauan Riau	12.300	12.300	12.300	12.350	12.350	12.250	12.150	12.200
Jambi	11.750	11.800	11.800	11.800	11.800	11.800	11.600	11.600
Bengkulu	12.550	12.450	12.450	12.250	12.300	12.250	12.200	12.200
Sumatera Selatan	12.150	12.200	12.150	12.000	12.150	12.150	12.100	12.150
K. Bangka Belitung	12.450	12.500	12.550	12.550	12.450	12.450	12.500	12.500
Lampung	11.700	11.750	11.700	11.700	11.650	11.650	11.600	11.600
Banten	11.700	11.650	11.700	11.900	11.650	11.600	11.500	11.800
Jawa Barat	12.200	12.250	12.250	12.300	12.350	12.250	12.200	12.250
DKI Jakarta	12.800	12.850	12.850	12.850	12.850	12.850	12.750	12.800
Jawa Tengah	11.750	11.750	11.900	11.850	11.850	11.800	11.750	11.750
DI Yogyakarta	12.100	11.900	11.900	11.900	11.900	11.550	11.550	11.450
Jawa Timur	11.900	11.950	12.000	11.850	12.000	11.900	11.900	11.850
Bali	13.400	13.450	13.300	13.200	13.050	13.050	13.200	13.300
Nusa Tenggara Barat	13.200	13.150	13.000	13.000	13.100	12.950	13.050	13.100
Nusa Tenggara Timur	14.850	14.950	14.950	15.050	15.050	14.850	15.000	15.000
Kalimantan Barat	12.650	12.650	12.600	12.600	12.550	12.500	12.400	12.500
Kalimantan Selatan	12.500	12.450	12.400	12.350	12.200	12.250	12.100	12.050
Kalimantan Tengah	14.050	13.800	13.750	13.700	13.750	13.700	13.750	13.750
Kalimantan Timur	14.650	14.550	14.550	14.550	14.450	14.450	14.500	14.500
Kalimantan Utara	14.200	14.300	13.950	14.050	13.600	13.850	13.700	13.600
Gorontalo	14.150	14.450	14.450	14.250	14.650	16.050	15.900	16.200
Sulawesi Selatan	12.600	12.600	12.750	12.750	12.700	12.700	12.600	12.450
Sulawesi Tenggara	14.100	14.100	14.100	14.000	14.000	14.000	14.000	14.000
Sulawesi Tengah	13.700	13.700	13.500	13.500	13.650	13.400	13.350	13.350
Sulawesi Utara	14.200	14.350	14.150	14.300	14.400	14.250	14.250	14.700
Sulawesi Barat	13.100	13.150	13.850	13.850	13.900	13.900	13.900	13.700
Maluku	15.400	15.650	15.400	15.400	15.450	15.450	15.450	15.400
Maluku Utara	15.100	15.100	15.100	15.100	15.100	15.100	15.100	15.500
Papua	14.950	14.950	14.900	14.650	14.850	14.800	14.750	14.800
Papua Barat	15.750	15.650	15.250	15.400	15.100	15.100	15.250	15.250

Lampiran 1. Data Harga Minyak Goreng pada 34 Provinsi di Indonesia (lanjutan)

Provinsi	2019			2020				
	Oktober	November	Desember	Januari	Februari	Maret	April	Mei
Aceh	13.200	12.900	13.150	13.600	14.050	13.800	13.800	13.800
Sumatera Utara	12.150	12.200	12.350	12.750	13.050	12.900	12.900	12.800
Sumatera Barat	11.700	11.650	12.200	13.050	13.550	13.200	13.000	13.000
Riau	12.150	12.050	12.550	13.050	13.550	13.400	13.150	13.200
Kepulauan Riau	12.150	12.300	12.450	12.450	12.450	12.600	12.850	13.600
Jambi	11.700	11.700	12.000	12.500	12.750	12.600	12.450	12.250
Bengkulu	12.250	12.250	12.250	12.250	12.550	13.300	13.400	13.400
Sumatera Selatan	12.150	12.100	12.400	13.100	13.400	13.050	13.000	12.850
K. Bangka Belitung	12.350	12.350	12.350	12.400	12.700	12.700	13.300	13.500
Lampung	11.700	11.750	12.100	12.600	12.900	12.850	13.100	12.800
Banten	11.600	11.700	12.150	12.850	13.300	13.050	12.800	12.550
Jawa Barat	12.200	12.200	12.650	12.950	13.500	13.200	13.150	13.200
DKI Jakarta	12.800	12.600	12.750	13.050	14.150	13.800	13.700	13.750
Jawa Tengah	11.750	11.800	12.050	12.600	12.850	12.750	12.600	12.600
DI Yogyakarta	11.450	11.350	11.850	12.150	12.500	12.400	12.400	12.600
Jawa Timur	11.800	11.850	12.150	12.600	12.850	12.650	12.600	12.550
Bali	13.200	13.350	13.600	13.500	13.900	13.850	13.900	13.750
Nusa Tenggara Barat	13.000	13.150	13.300	13.750	14.500	14.300	14.250	14.350
Nusa Tenggara Timur	15.000	15.000	14.800	15.300	15.200	15.400	15.400	15.400
Kalimantan Barat	12.500	12.550	12.750	12.800	13.450	13.150	12.950	12.700
Kalimantan Selatan	11.950	12.050	12.200	12.700	13.450	13.500	13.100	13.050
Kalimantan Tengah	13.700	13.750	13.800	13.950	14.500	14.050	13.950	13.900
Kalimantan Timur	14.550	14.300	14.400	14.500	15.050	14.650	14.250	14.200
Kalimantan Utara	13.600	13.650	13.650	13.650	13.750	14.250	14.800	14.800
Gorontalo	15.650	15.650	15.900	15.950	15.600	14.950	15.100	15.100
Sulawesi Selatan	12.550	12.650	12.650	12.550	13.050	13.000	13.100	12.950
Sulawesi Tenggara	13.950	13.950	14.150	14.000	14.500	14.450	14.450	14.500
Sulawesi Tengah	13.350	13.350	13.450	14.100	14.100	14.200	14.250	14.250
Sulawesi Utara	14.500	14.200	14.200	14.950	14.500	14.350	14.200	14.250
Sulawesi Barat	13.250	13.300	13.250	13.300	13.550	13.550	13.550	13.400
Maluku	15.400	15.150	15.150	15.200	15.250	15.350	15.700	15.850
Maluku Utara	15.500	15.500	15.500	14.850	14.950	14.950	16.250	16.250
Papua	14.800	14.650	14.700	14.750	14.800	14.950	14.800	14.250
Papua Barat	15.300	15.450	15.450	15.250	15.700	25.600	16.300	16.300

Lampiran 1. Data Harga Minyak Goreng pada 34 Provinsi di Indonesia (lanjutan)

Provinsi	2020							2021
	Juni	Juli	Agustus	September	Oktober	November	Desember	Januari
Aceh	13.500	13.450	13.750	13.750	14.000	14.100	14.250	14.400
Sumatera Utara	12.700	12.650	12.900	13.100	13.350	13.500	13.650	14.100
Sumatera Barat	12.600	12.500	13.000	13.250	13.800	14.000	13.900	13.800
Riau	12.850	12.850	13.100	13.400	13.850	13.900	13.950	13.950
Kepulauan Riau	12.650	12.600	12.600	13.050	13.200	13.200	13.300	13.200
Jambi	12.050	12.050	12.400	12.750	13.150	13.300	13.500	13.500
Bengkulu	13.400	13.300	12.950	12.950	12.950	12.950	13.100	13.100
Sumatera Selatan	12.850	12.650	13.100	13.450	13.550	13.600	13.750	13.750
K. Bangka Belitung	13.200	13.400	13.150	13.200	13.300	13.250	13.450	13.450
Lampung	12.550	12.500	12.750	12.900	13.500	13.600	13.650	13.800
Banten	12.700	12.250	12.800	13.150	13.400	13.500	14.200	14.200
Jawa Barat	13.050	12.900	13.250	13.500	14.050	14.250	14.300	14.400
DKI Jakarta	13.700	13.700	13.850	14.100	14.850	14.800	14.950	14.950
Jawa Tengah	12.450	12.450	12.700	12.850	13.450	13.500	13.650	13.750
DI Yogyakarta	12.500	12.400	12.650	12.750	13.550	13.750	13.750	14.150
Jawa Timur	12.900	12.450	12.850	13.100	13.400	13.550	13.650	13.750
Bali	13.550	13.550	13.750	13.750	13.800	14.000	14.200	14.100
Nusa Tenggara Barat	14.150	14.350	14.300	14.250	14.300	14.800	14.900	14.800
Nusa Tenggara Timur	15.150	15.100	14.950	14.900	14.900	15.350	15.500	15.500
Kalimantan Barat	12.550	12.600	12.850	13.200	13.650	13.750	13.950	14.300
Kalimantan Selatan	13.000	13.000	12.900	13.250	13.750	13.850	14.300	14.350
Kalimantan Tengah	13.950	13.850	13.750	14.100	14.150	14.350	14.550	14.650
Kalimantan Timur	14.400	14.100	14.200	14.250	14.600	14.950	14.900	15.150
Kalimantan Utara	14.800	14.950	14.650	14.850	14.900	14.850	15.000	14.900
Gorontalo	15.050	15.350	16.000	16.150	16.600	17.500	17.450	17.300
Sulawesi Selatan	12.900	12.950	13.100	13.200	13.300	13.300	13.400	13.400
Sulawesi Tenggara	14.400	14.500	14.550	14.450	14.600	14.550	14.450	14.400
Sulawesi Tengah	14.350	14.000	14.200	14.350	14.700	14.700	14.700	14.700
Sulawesi Utara	14.150	14.150	14.250	14.250	14.800	14.800	14.900	14.900
Sulawesi Barat	13.450	13.450	13.350	13.350	13.600	13.650	13.650	13.850
Maluku	15.900	15.350	15.450	15.600	15.550	15.550	15.450	15.700
Maluku Utara	16.250	16.250	15.250	15.750	15.850	15.600	15.150	15.650
Papua	14.400	14.900	14.950	15.000	15.000	15.100	15.200	15.700
Papua Barat	16.300	15.550	15.350	15.350	15.350	15.350	15.450	15.450

Lampiran 1. Data Harga Minyak Goreng pada 34 Provinsi di Indonesia (lanjutan)

Provinsi	2021							
	Februari	Maret	April	Mei	Juni	Juli	Agustus	September
Aceh	14.450	14.550	14.600	14.850	15.400	15.250	15.600	16.150
Sumatera Utara	14.250	14.450	14.650	14.800	15.150	15.300	15.700	16.100
Sumatera Barat	14.100	13.950	14.450	14.550	14.900	14.450	15.700	16.000
Riau	14.050	14.100	14.150	14.150	14.650	14.450	14.900	15.950
Kepulauan Riau	13.400	13.400	13.450	13.950	14.000	13.850	14.000	14.000
Jambi	13.600	13.550	13.650	14.000	14.200	14.250	14.950	15.700
Bengkulu	13.100	13.350	13.350	14.400	14.550	15.050	14.700	15.350
Sumatera Selatan	13.750	13.750	14.250	14.450	14.550	14.700	15.350	15.750
K. Bangka Belitung	13.450	13.500	13.600	14.300	14.150	14.350	14.350	14.400
Lampung	13.800	13.750	14.150	14.350	14.400	14.550	15.000	15.450
Banten	14.250	14.250	14.650	14.650	15.100	15.000	15.400	15.900
Jawa Barat	14.350	14.350	14.650	14.900	15.350	15.100	15.750	16.350
DKI Jakarta	14.950	14.950	15.550	15.400	16.050	15.850	16.450	16.650
Jawa Tengah	13.800	13.800	14.050	14.300	14.650	14.550	15.000	15.300
DI Yogyakarta	14.200	14.100	14.250	14.400	15.100	14.900	15.150	15.500
Jawa Timur	13.500	13.700	14.000	14.450	14.700	14.500	15.050	15.250
Bali	14.150	14.250	14.500	14.800	15.100	15.350	15.450	15.500
Nusa Tenggara Barat	14.800	14.800	14.800	14.900	15.000	15.200	15.400	15.600
Nusa Tenggara Timur	15.250	15.400	15.800	16.150	16.600	16.850	16.950	17.100
Kalimantan Barat	14.350	14.350	14.550	14.600	14.850	15.000	15.350	15.700
Kalimantan Selatan	14.350	14.400	14.550	14.800	14.950	15.400	15.500	15.650
Kalimantan Tengah	14.650	14.850	15.050	15.100	15.300	15.450	15.650	15.650
Kalimantan Timur	15.150	15.200	15.000	15.050	15.100	15.250	15.400	15.550
Kalimantan Utara	14.700	14.750	14.800	14.850	15.250	15.600	15.700	15.750
Gorontalo	17.300	17.350	17.750	17.650	18.150	17.500	17.750	18.350
Sulawesi Selatan	13.350	13.650	13.750	14.100	14.250	14.300	14.400	15.050
Sulawesi Tenggara	14.500	14.500	14.900	15.200	15.250	15.850	16.050	16.100
Sulawesi Tengah	14.950	14.950	14.950	15.000	15.000	15.150	15.150	15.600
Sulawesi Utara	14.900	14.900	14.900	15.450	15.950	15.900	16.300	16.950
Sulawesi Barat	13.850	13.850	14.100	14.750	14.750	15.100	15.400	15.450
Maluku	15.700	15.750	15.950	16.350	16.450	16.800	16.800	16.800
Maluku Utara	15.650	16.100	16.000	16.000	16.000	16.450	16.450	16.450
Papua	15.450	15.600	15.700	15.750	14.350	16.400	16.500	16.400
Papua Barat	16.400	16.400	16.400	16.500	16.750	16.900	17.150	17.150

Lampiran 1. Data Harga Minyak Goreng pada 34 Provinsi di Indonesia (lanjutan)

Provinsi	2021			2022				
	Oktober	November	Desember	Januari	Februari*	Maret*	April*	Mei*
Aceh	16.150	17.300	18.950	20.250	20.750	17.950	20.100	20.900
Sumatera Utara	16.150	18.000	18.700	19.150	19.700	19.550	21.200	21.350
Sumatera Barat	16.000	17.850	18.600	19.050	17.300	15.850	21.800	21.950
Riau	16.200	17.050	18.800	19.150	19.250	16.700	22.250	22.150
Kepulauan Riau	14.850	16.700	18.050	19.400	15.800	15.900	21.700	21.350
Jambi	15.700	16.450	18.150	18.400	18.700	18.700	22.550	22.550
Bengkulu	15.500	16.850	17.850	18.900	19.500	13.850	19.850	21.550
Sumatera Selatan	16.000	17.500	19.600	19.600	19.600	16.600	22.900	23.050
K. Bangka Belitung	14.650	17.850	19.850	19.350	17.150	15.900	24.650	23.900
Lampung	15.650	17.250	18.150	19.250	18.450	18.300	20.500	21.250
Banten	16.200	17.850	19.350	19.600	19.800	17.000	22.350	22.300
Jawa Barat	16.500	18.050	19.150	19.850	19.850	17.750	24.500	23.850
DKI Jakarta	17.100	18.300	20.100	20.650	20.300	19.000	22.150	23.900
Jawa Tengah	15.550	17.100	18.550	19.200	18.600	17.450	23.250	22.350
DI Yogyakarta	15.800	17.600	18.150	18.600	18.150	17.250	19.850	21.100
Jawa Timur	15.500	17.550	18.600	19.450	18.650	16.750	23.450	22.650
Bali	15.750	16.700	19.050	19.800	18.500	18.150	20.850	21.500
Nusa Tenggara Barat	15.850	16.750	18.000	19.300	19.500	19.000	24.150	24.550
Nusa Tenggara Timur	17.850	18.100	19.400	20.100	19.450	20.500	23.350	26.600
Kalimantan Barat	15.800	16.800	17.900	20.350	19.700	17.350	23.900	25.700
Kalimantan Selatan	16.000	16.700	18.400	19.750	20.200	18.800	22.600	23.300
Kalimantan Tengah	15.950	16.500	18.950	19.400	19.350	18.750	24.700	25.050
Kalimantan Timur	15.500	16.450	18.600	20.050	18.400	18.800	24.550	24.800
Kalimantan Utara	15.900	16.450	18.700	20.750	20.250	19.100	20.750	24.350
Gorontalo	18.300	23.000	25.300	25.850	26.200	25.000	30.550	31.750
Sulawesi Selatan	15.850	16.650	17.550	19.950	18.350	17.800	20.300	22.050
Sulawesi Tenggara	16.200	16.900	17.900	19.750	19.750	19.100	45.650	38.600
Sulawesi Tengah	16.300	17.550	19.500	20.550	18.050	17.400	24.850	28.000
Sulawesi Utara	17.050	17.100	17.650	18.600	17.500	17.200	19.050	21.900
Sulawesi Barat	15.450	16.950	18.150	19.050	20.150	18.100	26.450	25.150
Maluku	16.800	17.500	19.700	20.800	21.900	21.500	25.850	28.400
Maluku Utara	16.450	16.450	16.450	23.350	23.350	23.000	26.850	28.650
Papua	16.350	16.800	19.000	20.150	20.550	20.200	26.150	22.500
Papua Barat	17.500	18.550	20.700	22.150	22.350	21.350	30.650	32.400

Lampiran 1. Data Harga Minyak Goreng pada 34 Provinsi di Indonesia (lanjutan)

Provinsi	2022				
	Juni*	Juli*	Agustus*	September*	Oktober*
Aceh	20.600	19.350	19.150	19.150	18.750
Sumatera Utara	21.750	21.450	20.150	19.800	18.600
Sumatera Barat	21.050	20.300	17.650	17.750	17.700
Riau	21.500	21.550	20.050	19.200	16.850
Kepulauan Riau	20.450	20.750	16.350	15.750	15.400
Jambi	22.450	21.450	17.600	17.100	17.100
Bengkulu	23.800	22.150	17.750	17.000	17.500
Sumatera Selatan	21.150	20.500	17.400	17.200	17.150
K. Bangka Belitung	21.400	21.250	19.050	16.100	16.350
Lampung	20.750	21.150	19.200	18.600	17.950
Banten	22.100	21.050	19.250	18.200	18.150
Jawa Barat	21.950	21.100	19.250	17.900	17.750
DKI Jakarta	23.350	22.100	21.200	19.750	19.050
Jawa Tengah	21.750	20.850	18.350	17.850	17.500
DI Yogyakarta	20.350	20.000	18.000	17.350	16.950
Jawa Timur	21.500	20.950	18.550	17.650	17.250
Bali	21.000	20.550	19.350	18.000	18.850
Nusa Tenggara Barat	23.950	22.750	20.900	19.950	19.250
Nusa Tenggara Timur	28.550	27.050	25.250	24.100	23.300
Kalimantan Barat	22.400	21.900	20.100	19.450	18.950
Kalimantan Selatan	23.250	22.450	20.300	19.650	19.300
Kalimantan Tengah	23.350	23.100	21.450	21.350	21.150
Kalimantan Timur	25.150	23.800	22.550	20.650	19.450
Kalimantan Utara	24.000	25.350	22.200	21.150	19.650
Gorontalo	29.700	29.850	25.300	24.600	22.400
Sulawesi Selatan	22.600	21.900	20.300	19.900	18.850
Sulawesi Tenggara	33.700	28.200	24.250	22.600	21.800
Sulawesi Tengah	25.200	23.350	20.500	20.350	19.950
Sulawesi Utara	20.650	20.250	18.650	18.250	18.200
Sulawesi Barat	23.450	22.650	20.100	19.250	19.000
Maluku	28.400	28.050	26.150	22.200	22.050
Maluku Utara	28.500	27.350	25.500	28.100	26.350
Papua	29.050	29.350	28.950	25.000	23.150
Papua Barat	32.150	30.750	30.000	28.250	27.900

Sumber : Pusat Informasi Harga Pangan Strategis Nasional

Keterangan : (*) Data *out-sample*

Lampiran 2. Command R untuk Eksplorasi Data

```
> library(units)
> library(tmap)
> library(raster)
> library(sf)
> library(maptools)
> library(ggplot2)

> #Input Shp File
> spIndo = shapefile(file.choose())
> names(spIndo)
> spIndo$PROVINSI

> #Menampilkan Peta Indonesia
> tm_shape(spIndo) + tm_polygons()

> #Panggil Data
> data<-read.csv(file.choose(),header=TRUE)
> data

> #Menampilkan Layout Peta Tiap Variabel
> projection(spIndo) <- CRS("+init=epsg:3395")

> #Menampilkan peta tiap bulan
> spIndo$Z1=data$Z1 #Periode t=1
> tm_shape(spIndo) + tm_polygons(col="Z1",n=3,style="quantile")
+ tm_layout(main.title = "Oktober 2017", title.size =
  1.5,main.title.position="center")
```


Lampiran 3. Tabel Hasil Standarisasi Data

t	1	2	3	4	5	6	7	8	9
$\tilde{Z}(1)_t$	-0,350	-0,242	-0,285	-0,307	-0,328	-0,328	-0,350	-0,437	-0,480
$\tilde{Z}(2)_t$	-0,371	-0,389	-0,371	-0,406	-0,458	-0,458	-0,492	-0,475	-0,509
$\tilde{Z}(3)_t$	-0,252	-0,308	-0,363	-0,382	-0,455	-0,492	-0,511	-0,529	-0,529
$\tilde{Z}(4)_t$	-0,295	-0,295	-0,331	-0,367	-0,458	-0,440	-0,440	-0,440	-0,458
$\tilde{Z}(5)_t$	-0,525	-0,464	-0,464	-0,464	-0,464	-0,464	-0,464	-0,566	-0,566
$\tilde{Z}(6)_t$	-0,315	-0,281	-0,264	-0,485	-0,485	-0,468	-0,451	-0,451	-0,451
$\tilde{Z}(7)_t$	-0,587	-0,494	-0,513	-0,476	-0,402	-0,421	-0,494	-0,457	-0,513
$\tilde{Z}(8)_t$	-0,396	-0,414	-0,414	-0,414	-0,432	-0,467	-0,485	-0,485	-0,485
$\tilde{Z}(9)_t$	-0,519	-0,536	-0,536	-0,415	-0,380	-0,449	-0,415	-0,432	-0,415
$\tilde{Z}(10)_t$	-0,302	-0,284	-0,320	-0,284	-0,356	-0,374	-0,374	-0,374	-0,590
$\tilde{Z}(11)_t$	-0,435	-0,468	-0,500	-0,565	-0,614	-0,647	-0,679	-0,630	-0,647
$\tilde{Z}(12)_t$	-0,435	-0,468	-0,468	-0,484	-0,566	-0,566	-0,550	-0,566	-0,566
$\tilde{Z}(13)_t$	-0,512	-0,561	-0,594	-0,578	-0,594	-0,594	-0,594	-0,594	-0,594
$\tilde{Z}(14)_t$	-0,428	-0,462	-0,393	-0,393	-0,514	-0,514	-0,548	-0,462	-0,548
$\tilde{Z}(15)_t$	-0,485	-0,505	-0,524	-0,524	-0,524	-0,466	-0,485	-0,485	-0,466
$\tilde{Z}(16)_t$	-0,393	-0,410	-0,444	-0,461	-0,478	-0,513	-0,496	-0,547	-0,547
$\tilde{Z}(17)_t$	-0,353	-0,353	-0,438	-0,438	-0,480	-0,480	-0,459	-0,459	-0,565
$\tilde{Z}(18)_t$	-0,576	-0,576	-0,592	-0,641	-0,658	-0,641	-0,641	-0,625	-0,641
$\tilde{Z}(19)_t$	-0,556	-0,570	-0,570	-0,541	-0,541	-0,541	-0,541	-0,541	-0,541
$\tilde{Z}(20)_t$	-0,382	-0,382	-0,398	-0,430	-0,462	-0,446	-0,430	-0,462	-0,462
$\tilde{Z}(21)_t$	-0,408	-0,408	-0,425	-0,441	-0,457	-0,457	-0,457	-0,457	-0,489
$\tilde{Z}(22)_t$	-0,373	-0,423	-0,457	-0,423	-0,474	-0,474	-0,490	-0,507	-0,524
$\tilde{Z}(23)_t$	0,058	0,058	0,058	-0,797	-0,413	-0,413	-0,413	-0,431	-0,448
$\tilde{Z}(24)_t$	-0,540	-0,540	-0,573	-0,540	-0,523	-0,489	-0,523	-0,523	-0,523
$\tilde{Z}(25)_t$	-0,654	-0,654	-0,622	-0,633	-0,654	-0,633	-0,675	-0,675	-0,643
$\tilde{Z}(26)_t$	-0,400	-0,400	-0,400	-0,382	-0,382	-0,400	-0,418	-0,382	-0,292
$\tilde{Z}(27)_t$	-0,319	-0,361	-0,353	-0,378	-0,428	-0,428	-0,428	-0,428	-0,428
$\tilde{Z}(28)_t$	-0,462	-0,524	-0,477	-0,477	-0,431	-0,725	-0,679	-0,354	-0,493
$\tilde{Z}(29)_t$	-0,541	-0,514	-0,567	-0,541	-0,594	-0,567	-0,541	-0,541	-0,541
$\tilde{Z}(30)_t$	-0,416	-0,448	-0,432	-0,400	-0,481	-0,513	-0,706	-0,481	-0,481
$\tilde{Z}(31)_t$	-0,429	-0,444	-0,429	-0,358	-0,387	-0,387	-0,387	-0,387	-0,387
$\tilde{Z}(32)_t$	-0,362	-0,362	-0,362	-0,387	-0,387	-0,387	-0,387	-0,450	-0,450
$\tilde{Z}(33)_t$	-0,212	-0,422	-0,356	-0,409	-0,409	-0,409	-0,409	-0,540	-0,409
$\tilde{Z}(34)_t$	-0,359	-0,380	-0,369	-0,359	-0,400	-0,400	-0,349	-0,411	-0,411

Lampiran 3. Tabel Hasil Standarisasi Data (lanjutan)

t	10	11	12	13	14	15	16	17	18
$\tilde{Z}(1)_t$	-0,480	-0,524	-0,654	-0,610	-0,632	-0,675	-0,740	-0,675	-0,675
$\tilde{Z}(2)_t$	-0,492	-0,527	-0,682	-0,751	-0,751	-0,768	-0,803	-0,786	-0,786
$\tilde{Z}(3)_t$	-0,548	-0,640	-0,714	-0,714	-0,769	-0,899	-0,954	-0,917	-0,899
$\tilde{Z}(4)_t$	-0,422	-0,531	-0,549	-0,549	-0,622	-0,676	-0,785	-0,821	-0,821
$\tilde{Z}(5)_t$	-0,525	-0,546	-0,566	-0,525	-0,587	-0,690	-0,710	-0,649	-0,649
$\tilde{Z}(6)_t$	-0,468	-0,485	-0,553	-0,723	-0,757	-0,808	-0,808	-0,774	-0,757
$\tilde{Z}(7)_t$	-0,494	-0,531	-0,550	-0,568	-0,568	-0,587	-0,605	-0,605	-0,642
$\tilde{Z}(8)_t$	-0,450	-0,485	-0,699	-0,699	-0,717	-0,824	-0,806	-0,788	-0,770
$\tilde{Z}(9)_t$	-0,449	-0,449	-0,449	-0,502	-0,536	-0,589	-0,658	-0,658	-0,641
$\tilde{Z}(10)_t$	-0,590	-0,698	-0,716	-0,861	-0,806	-0,861	-0,879	-0,879	-0,861
$\tilde{Z}(11)_t$	-0,614	-0,630	-0,663	-0,712	-0,761	-0,842	-0,859	-0,826	-0,842
$\tilde{Z}(12)_t$	-0,566	-0,599	-0,665	-0,681	-0,714	-0,731	-0,813	-0,813	-0,796
$\tilde{Z}(13)_t$	-0,594	-0,594	-0,643	-0,643	-0,643	-0,725	-0,791	-0,807	-0,791
$\tilde{Z}(14)_t$	-0,531	-0,582	-0,617	-0,634	-0,686	-0,720	-0,720	-0,823	-0,823
$\tilde{Z}(15)_t$	-0,524	-0,582	-0,582	-0,679	-0,699	-0,718	-0,737	-0,737	-0,815
$\tilde{Z}(16)_t$	-0,598	-0,649	-0,683	-0,718	-0,735	-0,752	-0,786	-0,769	-0,752
$\tilde{Z}(17)_t$	-0,628	-0,628	-0,713	-0,713	-0,691	-0,691	-0,691	-0,649	-0,628
$\tilde{Z}(18)_t$	-0,674	-0,658	-0,674	-0,674	-0,707	-0,707	-0,641	-0,658	-0,674
$\tilde{Z}(19)_t$	-0,541	-0,570	-0,570	-0,570	-0,570	-0,570	-0,556	-0,556	-0,527
$\tilde{Z}(20)_t$	-0,462	-0,494	-0,844	-0,526	-0,621	-0,653	-0,669	-0,669	-0,669
$\tilde{Z}(21)_t$	-0,489	-0,489	-0,571	-0,603	-0,603	-0,635	-0,668	-0,733	-0,749
$\tilde{Z}(22)_t$	-0,474	-0,507	-0,490	-0,507	-0,557	-0,507	-0,540	-0,524	-0,607
$\tilde{Z}(23)_t$	-0,466	-0,448	-0,448	-0,483	-0,483	-0,536	-0,518	-0,483	-0,518
$\tilde{Z}(24)_t$	-0,523	-0,573	-0,607	-0,607	-0,590	-0,607	-0,624	-0,506	-0,472
$\tilde{Z}(25)_t$	-0,675	-0,760	-0,749	-0,728	-0,738	-0,781	-0,781	-0,781	-0,717
$\tilde{Z}(26)_t$	-0,400	-0,599	-0,617	-0,617	-0,599	-0,653	-0,689	-0,689	-0,689
$\tilde{Z}(27)_t$	-0,428	-0,428	-0,419	-0,419	-0,419	-0,470	-0,428	-0,428	-0,428
$\tilde{Z}(28)_t$	-0,493	-0,508	-0,446	-0,477	-0,400	-0,462	-0,508	-0,586	-0,586
$\tilde{Z}(29)_t$	-0,594	-0,621	-0,621	-0,594	-0,675	-0,648	-0,755	-0,675	-0,594
$\tilde{Z}(30)_t$	-0,481	-0,271	-0,481	-0,497	-0,577	-0,625	-0,625	-0,658	-0,642
$\tilde{Z}(31)_t$	-0,387	-0,401	-0,358	-0,358	-0,415	-0,415	-0,429	-0,543	-0,472
$\tilde{Z}(32)_t$	-0,450	-0,450	-0,450	-0,450	-0,450	-0,450	-0,575	-0,575	-0,575
$\tilde{Z}(33)_t$	-0,396	-0,435	-0,474	-0,487	-0,487	-0,448	-0,461	-0,474	-0,474
$\tilde{Z}(34)_t$	-0,411	-0,411	-0,411	-0,504	-0,462	-0,535	-0,504	-0,504	-0,524

Lampiran 3. Tabel Hasil Standarisasi Data (lanjutan)

t	19	20	21	22	23	24	25	26	27
$\tilde{Z}(1)_t$	-0,719	-0,740	-0,740	-0,762	-0,762	-0,827	-0,827	-0,957	-0,849
$\tilde{Z}(2)_t$	-0,786	-0,786	-0,751	-0,820	-0,837	-0,837	-0,837	-0,820	-0,768
$\tilde{Z}(3)_t$	-0,899	-0,899	-0,899	-0,917	-0,972	-0,917	-0,917	-0,935	-0,732
$\tilde{Z}(4)_t$	-0,785	-0,803	-0,803	-0,821	-0,858	-0,894	-0,876	-0,912	-0,731
$\tilde{Z}(5)_t$	-0,649	-0,628	-0,628	-0,669	-0,710	-0,690	-0,710	-0,649	-0,587
$\tilde{Z}(6)_t$	-0,757	-0,757	-0,757	-0,757	-0,825	-0,825	-0,791	-0,791	-0,689
$\tilde{Z}(7)_t$	-0,642	-0,716	-0,697	-0,716	-0,734	-0,734	-0,716	-0,716	-0,716
$\tilde{Z}(8)_t$	-0,788	-0,842	-0,788	-0,788	-0,806	-0,788	-0,788	-0,806	-0,699
$\tilde{Z}(9)_t$	-0,623	-0,623	-0,658	-0,658	-0,641	-0,641	-0,693	-0,693	-0,693
$\tilde{Z}(10)_t$	-0,879	-0,879	-0,897	-0,897	-0,915	-0,915	-0,879	-0,861	-0,734
$\tilde{Z}(11)_t$	-0,826	-0,761	-0,842	-0,859	-0,891	-0,793	-0,859	-0,826	-0,679
$\tilde{Z}(12)_t$	-0,796	-0,780	-0,764	-0,796	-0,813	-0,796	-0,813	-0,813	-0,665
$\tilde{Z}(13)_t$	-0,791	-0,791	-0,791	-0,791	-0,824	-0,807	-0,807	-0,873	-0,824
$\tilde{Z}(14)_t$	-0,771	-0,789	-0,789	-0,806	-0,823	-0,823	-0,823	-0,806	-0,720
$\tilde{Z}(15)_t$	-0,815	-0,815	-0,815	-0,951	-0,951	-0,990	-0,990	-1,028	-0,834
$\tilde{Z}(16)_t$	-0,735	-0,786	-0,735	-0,769	-0,769	-0,786	-0,803	-0,786	-0,683
$\tilde{Z}(17)_t$	-0,691	-0,734	-0,797	-0,797	-0,734	-0,691	-0,734	-0,670	-0,565
$\tilde{Z}(18)_t$	-0,724	-0,724	-0,691	-0,740	-0,707	-0,691	-0,724	-0,674	-0,625
$\tilde{Z}(19)_t$	-0,527	-0,498	-0,498	-0,556	-0,513	-0,513	-0,513	-0,513	-0,570
$\tilde{Z}(20)_t$	-0,685	-0,685	-0,701	-0,717	-0,749	-0,717	-0,717	-0,701	-0,637
$\tilde{Z}(21)_t$	-0,765	-0,781	-0,830	-0,814	-0,863	-0,879	-0,911	-0,879	-0,830
$\tilde{Z}(22)_t$	-0,624	-0,641	-0,624	-0,641	-0,624	-0,624	-0,641	-0,624	-0,607
$\tilde{Z}(23)_t$	-0,518	-0,518	-0,553	-0,553	-0,536	-0,536	-0,518	-0,605	-0,570
$\tilde{Z}(24)_t$	-0,590	-0,557	-0,709	-0,624	-0,675	-0,709	-0,709	-0,692	-0,692
$\tilde{Z}(25)_t$	-0,717	-0,760	-0,675	-0,379	-0,410	-0,347	-0,463	-0,463	-0,410
$\tilde{Z}(26)_t$	-0,635	-0,635	-0,653	-0,653	-0,689	-0,743	-0,707	-0,671	-0,671
$\tilde{Z}(27)_t$	-0,428	-0,445	-0,445	-0,445	-0,445	-0,445	-0,453	-0,453	-0,419
$\tilde{Z}(28)_t$	-0,648	-0,648	-0,601	-0,679	-0,694	-0,694	-0,694	-0,694	-0,663
$\tilde{Z}(29)_t$	-0,701	-0,621	-0,567	-0,648	-0,648	-0,407	-0,514	-0,675	-0,675
$\tilde{Z}(30)_t$	-0,416	-0,416	-0,400	-0,400	-0,400	-0,464	-0,609	-0,593	-0,609
$\tilde{Z}(31)_t$	-0,543	-0,543	-0,529	-0,529	-0,529	-0,543	-0,543	-0,615	-0,615
$\tilde{Z}(32)_t$	-0,575	-0,575	-0,575	-0,575	-0,575	-0,475	-0,475	-0,475	-0,475
$\tilde{Z}(33)_t$	-0,487	-0,553	-0,500	-0,513	-0,527	-0,513	-0,513	-0,553	-0,540
$\tilde{Z}(34)_t$	-0,607	-0,576	-0,638	-0,638	-0,607	-0,607	-0,597	-0,566	-0,566

Lampiran 3. Tabel Hasil Standarisasi Data (lanjutan)

t	28	29	30	31	32	33	34	35	36
$\tilde{Z}(1)_t$	-0,654	-0,459	-0,567	-0,567	-0,567	-0,697	-0,719	-0,589	-0,589
$\tilde{Z}(2)_t$	-0,630	-0,527	-0,579	-0,579	-0,613	-0,648	-0,665	-0,579	-0,509
$\tilde{Z}(3)_t$	-0,419	-0,234	-0,363	-0,437	-0,437	-0,585	-0,622	-0,437	-0,345
$\tilde{Z}(4)_t$	-0,549	-0,367	-0,422	-0,513	-0,494	-0,622	-0,622	-0,531	-0,422
$\tilde{Z}(5)_t$	-0,587	-0,587	-0,525	-0,422	-0,114	-0,505	-0,525	-0,525	-0,340
$\tilde{Z}(6)_t$	-0,519	-0,434	-0,485	-0,536	-0,604	-0,672	-0,672	-0,553	-0,434
$\tilde{Z}(7)_t$	-0,716	-0,605	-0,328	-0,291	-0,291	-0,291	-0,328	-0,457	-0,457
$\tilde{Z}(8)_t$	-0,450	-0,343	-0,467	-0,485	-0,539	-0,539	-0,610	-0,450	-0,325
$\tilde{Z}(9)_t$	-0,676	-0,571	-0,571	-0,362	-0,293	-0,397	-0,328	-0,415	-0,397
$\tilde{Z}(10)_t$	-0,554	-0,446	-0,464	-0,374	-0,482	-0,572	-0,590	-0,500	-0,446
$\tilde{Z}(11)_t$	-0,451	-0,305	-0,386	-0,468	-0,549	-0,500	-0,647	-0,468	-0,354
$\tilde{Z}(12)_t$	-0,566	-0,386	-0,484	-0,501	-0,484	-0,534	-0,583	-0,468	-0,386
$\tilde{Z}(13)_t$	-0,725	-0,365	-0,479	-0,512	-0,496	-0,512	-0,512	-0,463	-0,381
$\tilde{Z}(14)_t$	-0,531	-0,445	-0,479	-0,531	-0,531	-0,582	-0,582	-0,497	-0,445
$\tilde{Z}(15)_t$	-0,718	-0,582	-0,621	-0,621	-0,543	-0,582	-0,621	-0,524	-0,485
$\tilde{Z}(16)_t$	-0,530	-0,444	-0,513	-0,530	-0,547	-0,427	-0,581	-0,444	-0,359
$\tilde{Z}(17)_t$	-0,607	-0,438	-0,459	-0,438	-0,501	-0,586	-0,586	-0,501	-0,501
$\tilde{Z}(18)_t$	-0,477	-0,230	-0,296	-0,313	-0,280	-0,345	-0,280	-0,296	-0,313
$\tilde{Z}(19)_t$	-0,426	-0,455	-0,398	-0,398	-0,398	-0,470	-0,484	-0,527	-0,541
$\tilde{Z}(20)_t$	-0,621	-0,414	-0,510	-0,573	-0,653	-0,701	-0,685	-0,605	-0,494
$\tilde{Z}(21)_t$	-0,668	-0,425	-0,408	-0,538	-0,554	-0,571	-0,571	-0,603	-0,489
$\tilde{Z}(22)_t$	-0,557	-0,373	-0,524	-0,557	-0,574	-0,557	-0,590	-0,624	-0,507
$\tilde{Z}(23)_t$	-0,536	-0,344	-0,483	-0,623	-0,640	-0,570	-0,675	-0,640	-0,623
$\tilde{Z}(24)_t$	-0,692	-0,658	-0,489	-0,302	-0,302	-0,302	-0,251	-0,353	-0,285
$\tilde{Z}(25)_t$	-0,400	-0,474	-0,612	-0,580	-0,580	-0,590	-0,527	-0,389	-0,358
$\tilde{Z}(26)_t$	-0,707	-0,527	-0,545	-0,509	-0,563	-0,581	-0,563	-0,509	-0,472
$\tilde{Z}(27)_t$	-0,445	-0,361	-0,369	-0,369	-0,361	-0,378	-0,361	-0,353	-0,369
$\tilde{Z}(28)_t$	-0,462	-0,462	-0,431	-0,415	-0,415	-0,385	-0,493	-0,431	-0,385
$\tilde{Z}(29)_t$	-0,272	-0,514	-0,594	-0,675	-0,648	-0,701	-0,701	-0,648	-0,648
$\tilde{Z}(30)_t$	-0,593	-0,513	-0,513	-0,513	-0,561	-0,545	-0,545	-0,577	-0,577
$\tilde{Z}(31)_t$	-0,600	-0,586	-0,558	-0,458	-0,415	-0,401	-0,558	-0,529	-0,486
$\tilde{Z}(32)_t$	-0,637	-0,612	-0,612	-0,287	-0,287	-0,287	-0,287	-0,537	-0,412
$\tilde{Z}(33)_t$	-0,527	-0,513	-0,474	-0,513	-0,658	-0,618	-0,487	-0,474	-0,461
$\tilde{Z}(34)_t$	-0,607	-0,514	1,533	-0,390	-0,390	-0,390	-0,545	-0,586	-0,586

Lampiran 3. Tabel Hasil Standarisasi Data (lanjutan)

t	37	38	39	40	41	42	43	44	45
$\tilde{Z}(1)_t$	-0,480	-0,437	-0,372	-0,307	-0,285	-0,242	-0,220	-0,112	0,127
$\tilde{Z}(2)_t$	-0,423	-0,371	-0,320	-0,164	-0,112	-0,043	0,026	0,078	0,198
$\tilde{Z}(3)_t$	-0,142	-0,068	-0,105	-0,142	-0,031	-0,086	0,098	0,135	0,265
$\tilde{Z}(4)_t$	-0,258	-0,240	-0,222	-0,222	-0,186	-0,168	-0,149	-0,149	0,032
$\tilde{Z}(5)_t$	-0,278	-0,278	-0,237	-0,278	-0,196	-0,196	-0,176	0,030	0,051
$\tilde{Z}(6)_t$	-0,298	-0,247	-0,179	-0,179	-0,145	-0,162	-0,128	-0,009	0,059
$\tilde{Z}(7)_t$	-0,457	-0,457	-0,402	-0,402	-0,402	-0,310	-0,310	0,078	0,133
$\tilde{Z}(8)_t$	-0,289	-0,271	-0,218	-0,218	-0,218	-0,218	-0,040	0,032	0,067
$\tilde{Z}(9)_t$	-0,362	-0,380	-0,310	-0,310	-0,310	-0,293	-0,258	-0,014	-0,066
$\tilde{Z}(10)_t$	-0,230	-0,194	-0,176	-0,122	-0,122	-0,140	0,004	0,076	0,094
$\tilde{Z}(11)_t$	-0,272	-0,240	-0,011	-0,011	0,005	0,005	0,135	0,135	0,282
$\tilde{Z}(12)_t$	-0,205	-0,139	-0,123	-0,090	-0,106	-0,106	-0,008	0,074	0,222
$\tilde{Z}(13)_t$	-0,135	-0,152	-0,103	-0,103	-0,103	-0,103	0,094	0,045	0,258
$\tilde{Z}(14)_t$	-0,239	-0,222	-0,170	-0,136	-0,119	-0,119	-0,033	0,053	0,174
$\tilde{Z}(15)_t$	-0,175	-0,097	-0,097	0,058	0,077	0,038	0,097	0,155	0,426
$\tilde{Z}(16)_t$	-0,256	-0,205	-0,171	-0,137	-0,222	-0,154	-0,051	0,103	0,188
$\tilde{Z}(17)_t$	-0,480	-0,395	-0,311	-0,353	-0,332	-0,290	-0,184	-0,057	0,070
$\tilde{Z}(18)_t$	-0,296	-0,132	-0,099	-0,132	-0,132	-0,132	-0,132	-0,099	-0,066
$\tilde{Z}(19)_t$	-0,541	-0,412	-0,369	-0,369	-0,441	-0,398	-0,283	-0,182	-0,053
$\tilde{Z}(20)_t$	-0,350	-0,318	-0,255	-0,143	-0,127	-0,127	-0,063	-0,048	0,032
$\tilde{Z}(21)_t$	-0,327	-0,295	-0,149	-0,133	-0,133	-0,116	-0,068	0,013	0,062
$\tilde{Z}(22)_t$	-0,490	-0,423	-0,357	-0,323	-0,323	-0,256	-0,190	-0,173	-0,106
$\tilde{Z}(23)_t$	-0,501	-0,378	-0,396	-0,309	-0,309	-0,291	-0,361	-0,344	-0,326
$\tilde{Z}(24)_t$	-0,268	-0,285	-0,234	-0,268	-0,336	-0,319	-0,302	-0,285	-0,149
$\tilde{Z}(25)_t$	-0,262	-0,072	-0,082	-0,114	-0,114	-0,104	-0,019	-0,040	0,066
$\tilde{Z}(26)_t$	-0,436	-0,436	-0,400	-0,400	-0,418	-0,310	-0,274	-0,148	-0,094
$\tilde{Z}(27)_t$	-0,344	-0,353	-0,369	-0,378	-0,361	-0,361	-0,294	-0,244	-0,236
$\tilde{Z}(28)_t$	-0,276	-0,276	-0,276	-0,276	-0,199	-0,199	-0,199	-0,183	-0,183
$\tilde{Z}(29)_t$	-0,353	-0,353	-0,299	-0,299	-0,299	-0,299	-0,299	-0,004	0,264
$\tilde{Z}(30)_t$	-0,497	-0,481	-0,481	-0,416	-0,416	-0,416	-0,336	-0,126	-0,126
$\tilde{Z}(31)_t$	-0,501	-0,501	-0,529	-0,458	-0,458	-0,444	-0,387	-0,273	-0,244
$\tilde{Z}(32)_t$	-0,387	-0,450	-0,562	-0,437	-0,437	-0,325	-0,350	-0,350	-0,350
$\tilde{Z}(33)_t$	-0,461	-0,435	-0,409	-0,278	-0,343	-0,304	-0,278	-0,264	-0,631
$\tilde{Z}(34)_t$	-0,586	-0,586	-0,566	-0,566	-0,369	-0,369	-0,369	-0,349	-0,297

Lampiran 3. Tabel Hasil Standarisasi Data (lanjutan)

t	46	47	48	49	50	51	52	53	54
$\tilde{Z}(1)_t$	0,062	0,214	0,452	0,452	0,951	1,666	2,230	2,447	1,233
$\tilde{Z}(2)_t$	0,250	0,388	0,526	0,544	1,183	1,424	1,580	1,770	1,718
$\tilde{Z}(3)_t$	0,098	0,560	0,671	0,671	1,354	1,631	1,797	1,151	0,615
$\tilde{Z}(4)_t$	-0,040	0,123	0,504	0,595	0,904	1,539	1,666	1,703	0,777
$\tilde{Z}(5)_t$	-0,011	0,051	0,051	0,400	1,161	1,717	2,272	0,791	0,832
$\tilde{Z}(6)_t$	0,076	0,314	0,569	0,569	0,825	1,403	1,488	1,590	1,590
$\tilde{Z}(7)_t$	0,318	0,188	0,428	0,484	0,982	1,351	1,739	1,960	-0,125
$\tilde{Z}(8)_t$	0,121	0,352	0,495	0,584	1,119	1,867	1,867	1,867	0,798
$\tilde{Z}(9)_t$	0,003	0,003	0,021	0,108	1,221	1,918	1,744	0,978	0,543
$\tilde{Z}(10)_t$	0,148	0,310	0,472	0,545	1,121	1,445	1,842	1,553	1,499
$\tilde{Z}(11)_t$	0,249	0,379	0,542	0,640	1,178	1,666	1,748	1,813	0,901
$\tilde{Z}(12)_t$	0,140	0,354	0,551	0,600	1,109	1,471	1,701	1,701	1,011
$\tilde{Z}(13)_t$	0,192	0,389	0,454	0,602	0,995	1,585	1,765	1,651	1,225
$\tilde{Z}(14)_t$	0,139	0,294	0,397	0,483	1,015	1,514	1,737	1,531	1,136
$\tilde{Z}(15)_t$	0,349	0,446	0,582	0,698	1,396	1,610	1,784	1,610	1,260
$\tilde{Z}(16)_t$	0,120	0,308	0,376	0,461	1,162	1,521	1,811	1,538	0,889
$\tilde{Z}(17)_t$	0,175	0,218	0,239	0,344	0,746	1,740	2,057	1,507	1,359
$\tilde{Z}(18)_t$	0,000	0,066	0,132	0,214	0,510	0,921	1,349	1,415	1,250
$\tilde{Z}(19)_t$	0,019	0,048	0,091	0,306	0,378	0,751	0,953	0,766	1,067
$\tilde{Z}(20)_t$	0,080	0,191	0,303	0,335	0,654	1,004	1,785	1,578	0,829
$\tilde{Z}(21)_t$	0,208	0,240	0,289	0,403	0,630	1,181	1,619	1,765	1,311
$\tilde{Z}(22)_t$	-0,056	0,011	0,011	0,111	0,295	1,113	1,263	1,246	1,046
$\tilde{Z}(23)_t$	-0,274	-0,221	-0,169	-0,187	0,145	0,895	1,401	0,826	0,965
$\tilde{Z}(24)_t$	-0,031	0,003	0,020	0,071	0,258	1,021	1,717	1,547	1,157
$\tilde{Z}(25)_t$	-0,072	-0,019	0,108	0,097	1,092	1,579	1,695	1,769	1,516
$\tilde{Z}(26)_t$	-0,076	-0,040	0,195	0,483	0,772	1,097	1,962	1,385	1,187
$\tilde{Z}(27)_t$	-0,135	-0,102	-0,093	-0,077	0,040	0,207	0,517	0,517	0,408
$\tilde{Z}(28)_t$	-0,137	-0,137	0,003	0,219	0,606	1,210	1,535	0,761	0,560
$\tilde{Z}(29)_t$	0,237	0,451	0,800	0,853	0,880	1,175	1,685	1,095	0,934
$\tilde{Z}(30)_t$	-0,014	0,083	0,099	0,099	0,582	0,968	1,258	1,612	0,952
$\tilde{Z}(31)_t$	-0,144	-0,144	-0,144	-0,144	0,055	0,682	0,996	1,309	1,195
$\tilde{Z}(32)_t$	-0,237	-0,237	-0,237	-0,237	-0,237	-0,237	1,488	1,488	1,400
$\tilde{Z}(33)_t$	-0,094	-0,068	-0,094	-0,107	0,011	0,587	0,889	0,994	0,902
$\tilde{Z}(34)_t$	-0,266	-0,214	-0,214	-0,142	0,075	0,520	0,820	0,861	0,655

Lampiran 3. Tabel Hasil Standarisasi Data (lanjutan)

t	55	56	57	58	59	60	61
$\tilde{Z}(1)_t$	2,165	2,512	2,382	1,840	1,753	1,753	1,580
$\tilde{Z}(2)_t$	2,287	2,339	2,477	2,374	1,925	1,804	1,390
$\tilde{Z}(3)_t$	2,812	2,868	2,535	2,258	1,280	1,317	1,298
$\tilde{Z}(4)_t$	2,792	2,756	2,520	2,538	1,993	1,685	0,831
$\tilde{Z}(5)_t$	3,218	3,074	2,704	2,827	1,017	0,770	0,626
$\tilde{Z}(6)_t$	2,899	2,899	2,865	2,525	1,216	1,046	1,046
$\tilde{Z}(7)_t$	2,089	2,717	3,547	2,938	1,314	1,037	1,222
$\tilde{Z}(8)_t$	3,043	3,097	2,420	2,188	1,083	1,012	0,994
$\tilde{Z}(9)_t$	3,588	3,327	2,457	2,405	1,639	0,612	0,699
$\tilde{Z}(10)_t$	2,292	2,562	2,382	2,526	1,824	1,607	1,373
$\tilde{Z}(11)_t$	2,644	2,627	2,562	2,220	1,634	1,292	1,275
$\tilde{Z}(12)_t$	3,229	3,015	2,391	2,112	1,504	1,060	1,011
$\tilde{Z}(13)_t$	2,257	2,830	2,650	2,240	1,945	1,470	1,241
$\tilde{Z}(14)_t$	3,129	2,820	2,613	2,304	1,445	1,273	1,153
$\tilde{Z}(15)_t$	2,269	2,754	2,463	2,327	1,551	1,299	1,144
$\tilde{Z}(16)_t$	3,178	2,905	2,512	2,324	1,504	1,196	1,059
$\tilde{Z}(17)_t$	2,501	2,776	2,564	2,374	1,867	1,296	1,655
$\tilde{Z}(18)_t$	2,944	3,076	2,878	2,484	1,875	1,563	1,332
$\tilde{Z}(19)_t$	1,886	2,820	3,380	2,949	2,432	2,102	1,872
$\tilde{Z}(20)_t$	2,916	3,490	2,438	2,279	1,705	1,498	1,339
$\tilde{Z}(21)_t$	2,543	2,770	2,754	2,495	1,797	1,587	1,473
$\tilde{Z}(22)_t$	3,033	3,150	2,582	2,499	1,948	1,914	1,847
$\tilde{Z}(23)_t$	2,972	3,059	3,181	2,710	2,274	1,611	1,192
$\tilde{Z}(24)_t$	1,717	2,938	2,820	3,278	2,209	1,853	1,344
$\tilde{Z}(25)_t$	2,690	2,944	2,510	2,542	1,579	1,431	0,965
$\tilde{Z}(26)_t$	2,088	2,720	2,918	2,666	2,088	1,944	1,565
$\tilde{Z}(27)_t$	4,847	3,668	2,849	1,930	1,269	0,993	0,860
$\tilde{Z}(28)_t$	2,867	3,842	2,975	2,402	1,520	1,473	1,349
$\tilde{Z}(29)_t$	1,926	3,454	2,784	2,569	1,711	1,497	1,470
$\tilde{Z}(30)_t$	3,640	3,222	2,675	2,417	1,596	1,322	1,242
$\tilde{Z}(31)_t$	2,435	3,162	3,162	3,062	2,520	1,395	1,352
$\tilde{Z}(32)_t$	2,363	2,813	2,775	2,488	2,025	2,675	2,238
$\tilde{Z}(33)_t$	2,461	1,505	3,221	3,300	3,195	2,160	1,675
$\tilde{Z}(34)_t$	2,578	2,940	2,888	2,599	2,443	2,082	2,009

Lampiran 4. Command R Contoh Perhitungan Jarak *Dynamic Time Warping*

```

> library(dtw)
> library(dtwclust)
> dataku=read.csv(file.choose(),header=TRUE)
> a1<-dataku$ACEH
> a2<-dataku$SUMATERA.UTARA
> datmat<-matrix(c(a1,a2),byrow=TRUE,nrow=3)
> dtw_matrix = matrix(rep(c(0),3721), nrow=61, ncol=61, byrow =
TRUE)
> dtw_matrix[1,1] = abs(a1[1] - a2[1])
> for (i in 2:61){
+   dtw_matrix[i,1] = abs(a1[i] - a2[1])
+ }
>
> for (j in 2:61){
+   dtw_matrix[1,j] = abs(a1[1] - a2[j])
+ }
> for (i in 2:61){
+   for (j in 2:61){
+     dtw_matrix[i,j] = abs(a1[i] - a2[j]) + min(dtw_matrix[i,j-
1], dtw_matrix[i-1,j], dtw_matrix[i-1,j-1])
+   }
+ }
> dtw_matrix
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 0.02122113 0.03848606 0.02122111 0.05575101 0.1075459 0.1075459 0.1420758
[2,] 0.129632927 0.16811899 0.15085404 0.18538394 0.2717087 0.3235035 0.3580334
[3,] 0.086268201 0.18980135 0.23712224 0.27165214 0.3579769 0.4443016 0.5306264
[4,] 0.064585839 0.14643663 0.21102247 0.31013821 0.4225627 0.5088875 0.6297421
[5,] 0.042903476 0.10307190 0.14597538 0.22340875 0.3526370 0.4818652 0.6456233
[6,] 0.042903476 0.10307190 0.14597538 0.22340875 0.3526370 0.4818652 0.6456233
[7,] 0.02122113 0.05970718 0.08092829 0.13667930 0.2442252 0.3517710 0.4938468
[8,] 0.065508337 0.06946450 0.12521551 0.11190673 0.1327231 0.1535396 0.2088859
[9,] 0.108873063 0.15711645 0.17833756 0.18624989 0.1344550 0.1552715 0.1655211
[10,] 0.108873063 0.20048118 0.26598951 0.25268073 0.1570033 0.1570033 0.1672530
[11,] 0.152237788 0.24384590 0.35271896 0.37038861 0.2229164 0.2229164 0.1883865
[12,] 0.282331964 0.41730480 0.52617786 0.60052103 0.4189236 0.4189236 0.3498638
[13,] 0.238967238 0.46066953 0.65627204 0.73061520 0.5715661 0.5715661 0.4679764
[14,] 0.260649601 0.48235189 0.72131913 0.88239174 0.7458909 0.7458909 0.6077713
[15,] 0.304014326 0.54739898 0.78636622 0.99080355 0.9635805 0.9635805 0.7909310
[16,] 0.369061414 0.65581079 0.91646039 1.12089773 1.2463172 1.2463172 1.0391378
[17,] 0.304014326 0.59076370 0.89477803 1.16426245 1.3385873 1.4640067 1.2222974
[18,] 0.304014326 0.59076370 0.89477803 1.16426245 1.3819520 1.5562769 1.4054571
[19,] 0.347379052 0.63412843 0.93814275 1.20762718 1.4253168 1.6430063 1.6319815
[20,] 0.369061414 0.69917552 1.00318984 1.27267427 1.4903638 1.7080534 1.8801883
[21,] 0.369061414 0.72085788 1.06823693 1.33772136 1.5554109 1.7731005 1.9562602
[22,] 0.390743777 0.74254024 1.11160165 1.42445081 1.6421404 1.8598300 2.0429896
[23,] 0.390743777 0.76422260 1.13328402 1.46781553 1.7288698 1.9465594 2.1297191
[24,] 0.455790865 0.82926969 1.22001347 1.55454498 1.8372816 2.0983359 2.2814956
[25,] 0.455790865 0.89431678 1.28506056 1.64127443 1.9240111 2.2067478 2.4332722
[26,] 0.585885041 1.02441096 1.48020182 1.83641570 2.1408347 2.4235714 2.6717781
[27,] 0.477473228 0.93768151 1.41515473 1.85809806 2.2275642 2.5319832 2.7801900
[28,] 0.282331964 0.54739898 0.82973094 1.07753300 1.2735402 1.4695474 1.6310247
[29,] 0.087190700 0.15711645 0.24430715 0.29696795 0.2978339 0.2986998 0.3323638
[30,] 0.195602513 0.26552826 0.35271896 0.40537976 0.4062457 0.4071117 0.3734477
[31,] 0.195602513 0.37394008 0.46113078 0.51379158 0.5146575 0.5155235 0.4481956
[32,] 0.195602513 0.37394008 0.56954259 0.62220339 0.6230693 0.6239353 0.5229434
[33,] 0.325696689 0.50403425 0.69963677 0.86070938 0.8615753 0.8624413 0.7277855
[34,] 0.347379052 0.65581079 0.85141330 1.01248592 1.1217637 1.1226296 0.9543099
[35,] 0.217284876 0.41730480 0.63458968 0.81734465 0.9483048 1.0792649 1.0507401
[36,] 0.217284876 0.41730480 0.63458968 0.81734465 0.9483048 1.0792649 1.1471703
[37,] 0.108873063 0.20048118 0.30935424 0.38369740 0.4062457 0.4287940 0.4407756
[38,] 0.065508337 0.11375172 0.17926006 0.21023850 0.2310549 0.2518713 0.3072176
[39,] 0.000461249 0.01726495 0.01772620 0.05179485 0.1376584 0.2235219 0.3439153
[40,] 0.064585839 0.08231204 0.08185079 0.11684194 0.2027054 0.2885689 0.4089624
[41,] 0.086268201 0.16811899 0.16811899 0.20264889 0.2894349 0.3752984 0.4956918
[42,] 0.129632927 0.23316608 0.29775192 0.33228182 0.4186066 0.5053926 0.6257860
[43,] 0.151315289 0.29821317 0.38448137 0.48359711 0.5699219 0.6562466 0.7775625
[44,] 0.259727103 0.42830734 0.55794027 0.67873837 0.8296490 0.9159737 1.0368284

```


Lampiran 4. Command R Contoh Perhitungan Jarak Dynamic Time Warping (lanjutan)

[45,]	0.498233092	0.77522514	0.92654043	1.09070326	1.2632962	1.4142068	1.5350615
[46,]	0.433186004	0.88363696	1.20841115	1.39425634	1.6102140	1.7828070	1.9682475
[47,]	0.584962542	1.03541350	1.46859950	1.82790359	2.0655436	2.2815013	2.4886242
[48,]	0.823468532	1.42569602	1.85888203	2.32659793	2.7376969	2.9753369	3.2258245
[49,]	0.823468532	1.66420201	2.24916456	2.71688046	3.2363912	3.6474902	3.9196601
[50,]	1.322162873	2.16289635	2.98636489	3.60585733	4.1253681	4.6448788	5.0905077
[51,]	2.037680840	3.37710866	4.20057720	5.05857563	5.7298629	6.2493737	6.8034143
[52,]	2.601422269	4.65636806	5.97853093	6.83652937	7.7463226	8.4176099	8.9716506
[53,]	2.818245896	5.43693312	7.47461395	8.83130673	9.7411000	10.6508933	11.3567105
[54,]	1.604033587	3.22533212	4.82936571	6.46792920	8.1582875	9.8486459	11.5735341
[55,]	2.536375181	4.15767372	5.76170730	7.40027079	9.0906291	10.7809875	12.5058757
[56,]	2.883292984	5.43693312	7.04096670	8.67953019	10.3698885	12.0602469	13.7851351
[57,]	2.753198808	5.52366257	8.19013192	9.82869541	11.5190538	13.2094121	14.9343003
[58,]	2.211139741	4.43954443	6.65068417	8.89635381	11.1938183	13.4912828	15.5414065
[59,]	2.124410291	4.26608553	6.39049582	8.54943601	10.7601711	12.9709061	15.2161710
[60,]	2.124410291	4.26608553	6.39049582	8.54943601	10.7601711	12.9709061	15.2161710
[61,]	1.950951390	3.91916773	5.87011912	7.85560041	9.8928766	11.9301527	14.0019587
	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]
[1,]	0.1248108	0.1593407	0.1420758	0.1766057	0.3319902	0.4010500	0.4010500
[2,]	0.3580334	0.3925633	0.3925633	0.4270932	0.6170077	0.8414521	0.9105119
[3,]	0.5478913	0.5824213	0.5996862	0.6342161	0.8241306	1.0831048	1.3075492
[4,]	0.6988019	0.7505968	0.7678617	0.8196566	1.0095711	1.2685453	1.5275196
[5,]	0.7762353	0.8798250	0.9143549	0.9661498	1.1733292	1.4323034	1.6912777
[6,]	0.7921165	0.9572584	1.0435832	1.1126430	1.3198224	1.5960616	1.8550358
[7,]	0.6186576	0.7779983	0.9200741	1.0966798	1.4286700	1.7208724	1.9971116
[8,]	0.2469672	0.3195785	0.3749248	0.4648010	0.7100618	1.0243824	1.3387030
[9,]	0.1708045	0.2000510	0.2120326	0.2585441	0.4604402	0.7313960	1.0023519
[10,]	0.1708045	0.2000510	0.2120326	0.2585441	0.4604402	0.7313960	1.0023519
[11,]	0.2159011	0.1849227	0.2163058	0.2151794	0.3737107	0.6013019	0.8288930
[12,]	0.3671287	0.3291351	0.3464000	0.3421268	0.2436166	0.3411135	0.4386104
[13,]	0.4852413	0.4299827	0.4472476	0.4257095	0.3154184	0.3844782	0.4819752
[14,]	0.6250363	0.5525127	0.5697776	0.5309745	0.3655379	0.4345977	0.5036575
[15,]	0.8081959	0.7184074	0.7356724	0.6796043	0.3722927	0.4413525	0.5104123
[16,]	1.0564027	0.9493492	0.9666142	0.8932812	0.4305850	0.3830602	0.3938277
[17,]	1.2395624	1.1152439	1.1325089	1.0419109	0.4373398	0.4588748	0.4588748
[18,]	1.4227220	1.2811387	1.2984036	1.1905407	0.4440946	0.5131544	0.5346894
[19,]	1.6492464	1.4903981	1.5076631	1.3825352	0.4807045	0.4765444	0.5089943
[20,]	1.8974532	1.7213399	1.7386049	1.5962121	0.5389968	0.4873119	0.4873119
[21,]	2.1456600	1.9522817	1.9695467	1.8098889	0.5972891	0.4980794	0.4980794
[22,]	2.2434142	2.2049059	2.2221709	2.0452482	0.6772638	0.5089943	0.5089943
[23,]	2.3301437	2.4575301	2.4747950	2.2806074	0.7572385	0.5199092	0.5199092
[24,]	2.4819202	2.6478150	2.7924663	2.5810137	0.9022602	0.5958711	0.5958711
[25,]	2.6336968	2.7995915	2.9827512	2.8814200	1.0472820	0.6718331	0.6718331
[26,]	2.9155675	3.0814622	3.2646219	3.3119205	1.3223979	0.8778892	0.8778892
[27,]	3.0456617	3.2549211	3.4380808	3.5867106	1.4891020	0.9755335	0.9755335
[28,]	1.8097670	1.9539794	2.1154567	2.2424041	1.5175392	1.0730305	1.0730305
[29,]	0.3487628	0.3996917	0.4333557	0.5015495	0.7251279	1.0177661	1.3104043
[30,]	0.4243766	0.4062457	0.4744396	0.4735736	0.5887402	0.7729666	0.9571930
[31,]	0.4654605	0.4637286	0.4809936	0.5137916	0.5887402	0.7729666	0.9571930
[32,]	0.5402084	0.5212115	0.5384765	0.5212115	0.6289582	0.7729666	0.9571930
[33,]	0.7450504	0.7087886	0.7260536	0.6915237	0.5361391	0.5902713	0.6444036
[34,]	0.9715748	0.9180481	0.9353130	0.8835182	0.5727490	0.5685890	0.6010388
[35,]	1.0680050	0.9972133	1.0144783	0.9454185	0.6662333	0.7311330	0.7311330
[36,]	1.1644352	1.0763786	1.0936436	1.0073188	0.7597175	0.8287773	0.8936770
[37,]	0.4460590	0.4753055	0.4872871	0.5337986	0.7356946	1.0066505	1.0997332
[38,]	0.3452990	0.4179103	0.4732566	0.5631328	0.7790594	1.0500152	1.3209711
[39,]	0.4103461	0.4829574	0.5383037	0.6281799	0.8734407	1.1584270	1.4293829
[40,]	0.5120908	0.6130515	0.6683979	0.7582741	1.0035349	1.3178554	1.6028418
[41,]	0.5988203	0.7364786	0.8201744	0.9100506	1.1553114	1.4696320	1.7839525
[42,]	0.7289144	0.8665728	0.9869662	1.1051919	1.3504527	1.6647732	1.9790938
[43,]	0.8806910	1.0183493	1.1387427	1.2936660	1.5672763	1.8815969	2.1959174
[44,]	1.1408793	1.2785377	1.3989311	1.5538544	1.8641622	2.2068323	2.5211529
[45,]	1.6386512	1.7772320	1.8976254	2.0525487	2.3628566	2.7422243	3.0848943
[46,]	2.0718372	2.2099568	2.3312727	2.4861960	2.7965038	3.1758715	3.5552392
[47,]	2.6567997	2.7949193	2.9157740	3.0716198	3.3819276	3.7612953	4.1406630
[48,]	3.4156824	3.6183878	3.7392425	3.8946271	4.2058574	4.5852251	4.9645927
[49,]	4.1528827	4.3772705	4.5627110	4.7180956	5.0288647	5.4091549	5.7885225
[50,]	5.3454127	5.6131652	5.8202881	6.0402585	6.3510276	6.7308565	7.1111466
[51,]	7.2317782	7.5212131	7.7717007	8.0133535	8.3887084	8.7685373	9.1483662
[52,]	9.5084263	9.9713201	10.2434900	10.5285075	10.9255448	11.3699596	11.7497885
[53,]	11.8934862	12.4647918	12.9104207	13.2171205	13.6575225	14.1236196	14.5680344
[54,]	13.0643338	13.6356394	14.1896801	14.6698388	15.1319232	15.6413850	16.1074821
[55,]	14.2134990	15.7388286	16.2928692	16.8814398	17.5169831	18.0481273	18.5575891
[56,]	15.4927584	17.2349116	18.7429762	19.3315468	20.0755019	20.7801050	21.3112492
[57,]	16.6419236	18.3840768	20.1089651	21.6515596	22.3955147	23.2085296	23.9131327
[58,]	17.2490298	18.9911830	20.7160712	22.4754893	24.1734684	24.9864833	25.7994983

Lampiran 4. Command R Contoh Perhitungan Jarak Dynamic Time Warping (lanjutan)

[45,]	0.498233092	0.77522514	0.92654043	1.09070326	1.2632962	1.4142068	1.5350615
[46,]	0.433186004	0.88363696	1.20841115	1.39425634	1.6102140	1.7828070	1.9682475
[47,]	0.584962542	1.03541350	1.46859950	1.82790359	2.0655436	2.2815013	2.4886242
[48,]	0.823468532	1.42569602	1.85888203	2.32659793	2.7376969	2.9753369	3.2258245
[49,]	0.823468532	1.66420201	2.24916456	2.71688046	3.2363912	3.6474902	3.9196601
[50,]	1.322162873	2.16289635	2.98636489	3.60585733	4.1253681	4.6448788	5.0905077
[51,]	2.037680840	3.37710866	4.20057720	5.05857563	5.7298629	6.2493737	6.8034143
[52,]	2.601422269	4.65636806	5.97853093	6.83652937	7.7463226	8.4176099	8.9716506
[53,]	2.818245896	5.43693312	7.47461395	8.83130673	9.7411000	10.6508933	11.3567105
[54,]	1.604033587	3.22533212	4.82936571	6.46792920	8.1582875	9.8486459	11.5735341
[55,]	2.536375181	4.15767372	5.76170730	7.40027079	9.0906291	10.7809875	12.5058757
[56,]	2.883292984	5.43693312	7.04096670	8.67953019	10.3698885	12.0602469	13.7851351
[57,]	2.753198808	5.52366257	8.19013192	9.82869541	11.5190538	13.2094121	14.9343003
[58,]	2.211139741	4.43954443	6.65068417	8.89635381	11.1938183	13.4912828	15.5414065
[59,]	2.124410291	4.26608553	6.39049582	8.54943601	10.7601711	12.9709061	15.2161710
[60,]	2.124410291	4.26608553	6.39049582	8.54943601	10.7601711	12.9709061	15.2161710
[61,]	1.950951390	3.91916773	5.87011912	7.85560041	9.8928766	11.9301527	14.0019587
	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]
[1,]	0.1248108	0.1593407	0.1420758	0.1766057	0.3319902	0.4010500	0.4010500
[2,]	0.3580334	0.3925633	0.3925633	0.4270932	0.6170077	0.8414521	0.9105119
[3,]	0.5478913	0.5824213	0.5996862	0.6342161	0.8241306	1.0831048	1.3075492
[4,]	0.6988019	0.7505968	0.7678617	0.8196566	1.0095711	1.2685453	1.5275196
[5,]	0.7762353	0.8798250	0.9143549	0.9661498	1.1733292	1.4323034	1.6912777
[6,]	0.7921165	0.9572584	1.0435832	1.1126430	1.3198224	1.5960616	1.8550358
[7,]	0.6186576	0.7779983	0.9200741	1.0966798	1.4286700	1.7208724	1.9971116
[8,]	0.2469672	0.3195785	0.3749248	0.4648010	0.7100618	1.0243824	1.3387030
[9,]	0.1708045	0.2000510	0.2120326	0.2585441	0.4604402	0.7313960	1.0023519
[10,]	0.1708045	0.2000510	0.2120326	0.2585441	0.4604402	0.7313960	1.0023519
[11,]	0.2159011	0.1849227	0.2163058	0.2151794	0.3737107	0.6013019	0.8288930
[12,]	0.3671287	0.3291351	0.3464000	0.3421268	0.2436166	0.3411135	0.4386104
[13,]	0.4852413	0.4299827	0.4472476	0.4257095	0.3154184	0.3844782	0.4819752
[14,]	0.6250363	0.5525127	0.5697776	0.5309745	0.3655379	0.4345977	0.5036575
[15,]	0.8081959	0.7184074	0.7356724	0.6796043	0.3722927	0.4413525	0.5104123
[16,]	1.0564027	0.9493492	0.9666142	0.8932812	0.4305850	0.3830602	0.3938277
[17,]	1.2395624	1.1152439	1.1325089	1.0419109	0.4373398	0.4588748	0.4588748
[18,]	1.4227220	1.2811387	1.2984036	1.1905407	0.4440946	0.5131544	0.5346894
[19,]	1.6492464	1.4903981	1.5076631	1.3825352	0.4807045	0.4765444	0.5089943
[20,]	1.8974532	1.7213399	1.7386049	1.5962121	0.5389968	0.4873119	0.4873119
[21,]	2.1456600	1.9522817	1.9695467	1.8098889	0.5972891	0.4980794	0.4980794
[22,]	2.2434142	2.2049059	2.2221709	2.0452482	0.6772638	0.5089943	0.5089943
[23,]	2.3301437	2.4575301	2.4747950	2.2806074	0.7572385	0.5199092	0.5199092
[24,]	2.4819202	2.6478150	2.7924663	2.5810137	0.9022602	0.5958711	0.5958711
[25,]	2.6336968	2.7995915	2.9827512	2.8814200	1.0472820	0.6718331	0.6718331
[26,]	2.9155675	3.0814622	3.2646219	3.3119205	1.3223979	0.8778892	0.8778892
[27,]	3.0456617	3.2549211	3.4380808	3.5867106	1.4891020	0.9755335	0.9755335
[28,]	1.8097670	1.9539794	2.1154567	2.2424041	1.5175392	1.0730305	1.0730305
[29,]	0.3487628	0.3996917	0.4333557	0.5015495	0.7251279	1.0177661	1.3104043
[30,]	0.4243766	0.4062457	0.4744396	0.4735736	0.5887402	0.7729666	0.9571930
[31,]	0.4654605	0.4637286	0.4809936	0.5137916	0.5887402	0.7729666	0.9571930
[32,]	0.5402084	0.5212115	0.5384765	0.5212115	0.6289582	0.7729666	0.9571930
[33,]	0.7450504	0.7087886	0.7260536	0.6915237	0.5361391	0.5902713	0.6444036
[34,]	0.9715748	0.9180481	0.9353130	0.8835182	0.5727490	0.5685890	0.6010388
[35,]	1.0680050	0.9972133	1.0144783	0.9454185	0.6662333	0.7311330	0.7311330
[36,]	1.1644352	1.0763786	1.0936436	1.0073188	0.7597175	0.8287773	0.8936770
[37,]	0.4460590	0.4753055	0.4872871	0.5337986	0.7356946	1.0066505	1.0997332
[38,]	0.3452990	0.4179103	0.4732566	0.5631328	0.7790594	1.0500152	1.3209711
[39,]	0.4103461	0.4829574	0.5383037	0.6281799	0.8734407	1.1584270	1.4293829
[40,]	0.5120908	0.6130515	0.6683979	0.7582741	1.0035349	1.3178554	1.6028418
[41,]	0.5988203	0.7364786	0.8201744	0.9100506	1.1553114	1.4696320	1.7839525
[42,]	0.7289144	0.8665728	0.9869662	1.1051919	1.3504527	1.6647732	1.9790938
[43,]	0.8806910	1.0183493	1.1387427	1.2936660	1.5672763	1.8815969	2.1959174
[44,]	1.1408793	1.2785377	1.3989311	1.5538544	1.8641622	2.2068323	2.5211529
[45,]	1.6386512	1.7772320	1.8976254	2.0525487	2.3628566	2.7422243	3.0848943
[46,]	2.0718372	2.2099568	2.3312727	2.4861960	2.7965038	3.1758715	3.5552392
[47,]	2.6567997	2.7949193	2.9157740	3.0716198	3.3819276	3.7612953	4.1406630
[48,]	3.4156824	3.6183878	3.7392425	3.8946271	4.2058574	4.5852251	4.9645927
[49,]	4.1528827	4.3772705	4.5627110	4.7180956	5.0288647	5.4091549	5.7885225
[50,]	5.3454127	5.6131652	5.8202881	6.0402585	6.3510276	6.7308565	7.1111466
[51,]	7.2317782	7.5212131	7.7717007	8.0133535	8.3887084	8.7685373	9.1483662
[52,]	9.5084263	9.9713201	10.2434900	10.5285075	10.9255448	11.3699596	11.7497885
[53,]	11.8934862	12.4647918	12.9104207	13.2171205	13.6575225	14.1236196	14.5680344
[54,]	13.0643338	13.6356394	14.1896801	14.6698388	15.1319232	15.6413850	16.1074821
[55,]	14.2134990	15.7388286	16.2928692	16.8814398	17.5169831	18.0481273	18.5575891
[56,]	15.4927584	17.2349116	18.7429762	19.3315468	20.0755019	20.7801050	21.3112492
[57,]	16.6419236	18.3840768	20.1089651	21.6515596	22.3955147	23.2085296	23.9131327
[58,]	17.2490298	18.9911830	20.7160712	22.4754893	24.1734684	24.9864833	25.7994983

Lampiran 4. *Command R* Contoh Perhitungan Jarak *Dynamic Time Warping* (lanjutan)

[59,]	17.4441710	19.5115597	21.2364479	22.9958661	24.9106688	26.6777076	27.4907225
[60,]	17.4441710	19.7067009	21.7568246	23.5162428	25.4310455	27.4149080	29.1819468
[61,]	16.0564998	18.1455708	20.2173769	22.3237128	24.5854333	26.9162136	29.2469939
	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]	[,21]
[1,]	0.4183150	0.4528449	0.4355799	0.4355799	0.4355799	0.4355799	0.4010500
[2,]	0.9277768	0.9795717	0.9795717	0.9795717	0.9795717	0.9795717	0.9105119
[3,]	1.3938739	1.4456688	1.4801987	1.4801987	1.4801987	1.4801987	1.3766090
[4,]	1.7692289	1.8900835	1.9246134	1.9591433	1.9591433	1.9591433	1.8210237
[5,]	1.9675169	2.2437561	2.3473458	2.3818757	2.4164056	2.4164056	2.2437561
[6,]	2.1312750	2.4420441	2.7010184	2.8046081	2.8391380	2.8736679	2.6664885
[7,]	2.2733508	2.5841199	2.8776241	3.1365983	3.2401880	3.2747179	3.0675385
[8,]	1.6702885	2.0364039	2.3852544	2.7341049	3.0829553	3.4318058	3.3818591
[9,]	1.2905727	1.6133234	1.9188091	2.2242949	2.5297806	2.8352664	3.1062222
[10,]	1.2905727	1.6133234	1.9188091	2.2242949	2.5297806	2.8352664	3.1062222
[11,]	1.0737491	1.3531350	1.6152561	1.8773771	2.1394981	2.4016191	2.6292103
[12,]	0.5533723	0.7026641	0.8346910	0.9667178	1.0987447	1.2307715	1.3282685
[13,]	0.5967371	0.7460289	0.8780557	1.0100826	1.1421094	1.2741363	1.3716332
[14,]	0.6184194	0.7677112	0.8997381	1.0317649	1.1637918	1.2958186	1.3933156
[15,]	0.5967371	0.7243465	0.8346910	0.9450355	1.0553800	1.1657245	1.2415390
[16,]	0.4218602	0.4844225	0.5297199	0.5750173	0.6203147	0.6656121	0.6763796
[17,]	0.4869072	0.5494696	0.5947670	0.6400644	0.6853618	0.7306592	0.7414267
[18,]	0.5519543	0.6145167	0.6598141	0.7051115	0.7504089	0.7957063	0.8064738
[19,]	0.5587091	0.6361990	0.6814964	0.7267938	0.7720912	0.8173886	0.8281561
[20,]	0.5153444	0.5779067	0.6232041	0.6685015	0.7137989	0.7590963	0.7698638
[21,]	0.5153444	0.5779067	0.6232041	0.6685015	0.7137989	0.7590963	0.7698638
[22,]	0.5044295	0.5453095	0.5689246	0.5925396	0.6161546	0.6397697	0.6506845
[23,]	0.5107796	0.5453095	0.5689246	0.5925396	0.6161546	0.6397697	0.6506845
[24,]	0.5694766	0.5349467	0.5763788	0.6103566	0.6339716	0.6575867	0.7157316
[25,]	0.6281736	0.5591138	0.5763788	0.6178108	0.6517887	0.6754037	0.7335486
[26,]	0.8169648	0.7133751	0.7306400	0.7479050	0.7893370	0.8233149	0.8814598
[27,]	0.8973442	0.7592246	0.7764895	0.7937545	0.8110194	0.8524515	0.9209592
[28,]	1.0121061	0.9085164	0.8912514	0.9085164	0.9257813	0.9430463	0.9499484
[29,]	1.3220092	1.2529494	1.2184195	1.2184195	1.2356845	1.2529494	1.2356845
[30,]	1.1586844	1.3947056	1.4371758	1.4371758	1.4371758	1.4544408	1.4199109
[31,]	1.1586844	1.3947056	1.6134619	1.6559321	1.6559321	1.6559321	1.6041373
[32,]	1.1586844	1.3947056	1.6134619	1.8322182	1.8746884	1.8746884	1.7883637
[33,]	0.7158007	0.8217278	0.9103899	0.9990520	1.0877142	1.1763763	1.2305085
[34,]	0.6507536	0.7349983	0.8019781	0.8689579	0.9359376	1.0029174	1.0353673
[35,]	0.7808478	0.8650925	0.9320723	0.9990520	1.0660318	1.1330116	1.1654614
[36,]	0.9109420	0.9951867	1.0621665	1.1291462	1.1961260	1.2631057	1.2955556
[37,]	1.1818978	1.2336927	1.3006724	1.3676522	1.4346320	1.5016117	1.5340616
[38,]	1.4313187	1.5480133	1.5825432	1.6495229	1.7165027	1.7834824	1.8159323
[39,]	1.7176037	1.8624812	1.9619108	1.9964407	2.0634205	2.1304003	2.1628501
[40,]	1.8910626	2.2138133	2.3414259	2.4408555	2.4753854	2.5423651	2.5748150
[41,]	2.0862038	2.4089545	2.7144403	2.8420529	2.9414825	2.9760124	3.0084623
[42,]	2.3106793	2.6474605	2.9529463	3.2584320	3.3860446	3.4854742	3.4854742
[43,]	2.5275030	2.8936184	3.2131346	3.5186204	3.8241061	3.9517187	4.0166184
[44,]	2.8527384	3.2188538	3.5677043	3.8872206	4.1927063	4.4981921	4.5912747
[45,]	3.4164798	3.7825952	4.1314457	4.4802962	4.7998125	5.1052982	5.3762541
[46,]	3.9151742	4.2812896	4.6301401	4.9789905	5.3278410	5.6473573	5.9183131
[47,]	4.5372956	4.9317605	5.2806109	5.6294614	5.9783119	6.3271624	6.6121487
[48,]	5.3612254	5.7923879	6.1695878	6.5184383	6.8672888	7.2161392	7.5304598
[49,]	6.1851551	6.6163177	7.0302152	7.4074152	7.7562656	8.1051161	8.4194367
[50,]	7.5077793	7.9389418	8.3528393	8.7667369	9.1439368	9.4927873	9.8071079
[51,]	9.5459213	9.9770839	10.3909814	10.8048790	11.2187766	11.5959765	11.9102971
[52,]	12.1468824	12.5789674	12.9928649	13.4067625	13.8206601	14.2345576	14.5772277
[53,]	14.9651283	15.3967520	15.8115721	16.2254697	16.6393672	17.0532648	17.4326324
[54,]	16.5691618	17.0007856	17.4151444	17.8299645	18.2438621	18.6577596	19.0371273
[55,]	19.0409512	19.5371608	19.9515196	20.3658784	20.7806985	21.1945960	21.5739637
[56,]	21.8379760	22.3558679	22.8348126	23.2491714	23.6635302	24.0783503	24.4577179
[57,]	24.4615419	25.0227985	25.5234256	26.0023702	26.4167290	26.8310878	27.2113780
[58,]	26.5213663	27.1043054	27.6482971	28.1489241	28.6278688	29.0422276	29.4220565
[59,]	28.3210024	29.0774004	29.6430745	30.1870662	30.6876932	31.1666379	31.5464668
[60,]	30.0122267	30.8770365	31.6161695	32.1818436	32.7258353	33.2264623	33.6708771
[61,]	31.5299921	32.3948018	33.2423467	33.9814797	34.5471538	35.0911455	35.5572426
	[,22]	[,23]	[,24]	[,25]	[,26]	[,27]	[,28]
[1,]	0.4701098	0.4873748	0.4873748	0.4873748	0.4701098	0.4183150	0.2801954
[2,]	0.9795717	1.0658964	1.0831614	1.0831614	1.0486315	0.9450418	0.6688026
[3,]	1.4456688	1.5319935	1.6183183	1.6355832	1.5837884	1.4284038	1.0140450
[4,]	1.8900835	1.9764083	2.0627330	2.1490578	2.0972629	1.8900835	1.3376051
[5,]	2.3128159	2.3991407	2.4854654	2.5717902	2.5890551	2.3300809	1.6394828
[6,]	2.7355483	2.8218731	2.9081978	2.9945226	3.0635824	2.7700782	1.9413606
[7,]	3.1365983	3.2229231	3.3092478	3.3955726	3.4646324	3.1883932	2.2215560
[8,]	3.4509189	3.5372437	3.6235684	3.7098932	3.7789530	3.5199787	2.4150219
[9,]	3.4462379	3.8035185	3.8945243	3.9808490	4.0499088	3.8081995	2.5651231
[10,]	3.4462379	3.8035185	4.1607991	4.2518049	4.3208647	4.0964203	2.7152243

Lampiran 4. Command R Contoh Perhitungan Jarak Dynamic Time Warping (lanjutan)

[11,]	2.9258612	3.2397771	3.5536929	3.8676088	4.1642597	4.3412764	2.8219607
[12,]	1.4948252	1.6786469	1.8624686	2.0462903	2.2128471	2.3276090	2.3509667
[13,]	1.5381900	1.7220117	1.9058334	2.0896551	2.2562118	2.3709737	2.3476160
[14,]	1.5598723	1.7436940	1.9275157	2.1113374	2.2778942	2.3926561	2.3492914
[15,]	1.3864134	1.5485528	1.7106921	1.8728315	2.0177058	2.1107854	2.1558254
[16,]	0.7562069	0.8532992	0.9503914	1.0474837	1.1273110	1.1553434	1.2654306
[17,]	0.8212540	0.9183462	1.0154385	1.1125307	1.1923580	1.2203905	1.2003835
[18,]	0.8863011	0.9833933	1.0804856	1.1775778	1.2574051	1.2854376	1.2454235
[19,]	0.9079834	1.0050757	1.1021679	1.1992602	1.2790875	1.3071199	1.3338283
[20,]	0.8496911	0.9467834	1.0438756	1.1409679	1.2207952	1.2488276	1.3589148
[21,]	0.8496911	0.9467834	1.0438756	1.1409679	1.2207952	1.2488276	1.3589148
[22,]	0.7088295	0.7842394	0.8596492	0.9350591	0.9932041	0.9995542	1.1313237
[23,]	0.7088295	0.7842394	0.8596492	0.9350591	0.9932041	0.9995542	1.1313237
[24,]	0.6575867	0.6679495	0.6783123	0.6886751	0.6955772	0.7542742	0.9510908
[25,]	0.6644888	0.6679495	0.6783123	0.6886751	0.6955772	0.7542742	0.9510908
[26,]	0.8014852	0.7842202	0.7876809	0.7980437	0.8256714	0.8843684	1.0811850
[27,]	0.8300697	0.7955398	0.7955398	0.7990004	0.8266282	0.9060508	1.1028674
[28,]	0.9966264	0.9793615	0.9793615	0.9793615	0.9655572	0.9413901	0.9294085
[29,]	1.3116464	1.3583244	1.3583244	1.3583244	1.3272552	1.2512932	1.1011920
[30,]	1.4889707	1.5821976	1.6288756	1.6288756	1.5805414	1.4527846	1.1645638
[31,]	1.6731971	1.7595218	1.8527487	1.8994267	1.8338276	1.6542759	1.2279355
[32,]	1.8574235	1.9437482	2.0300730	2.1232999	2.0871138	1.8557673	1.2913073
[33,]	1.3537005	1.4941575	1.6346145	1.7750715	1.8982635	1.9271644	1.3580297
[34,]	1.1368769	1.2556515	1.3744261	1.4932008	1.5947104	1.6444252	1.4464345
[35,]	1.2669711	1.3857457	1.5045203	1.6232949	1.7248046	1.7745194	1.4881239
[36,]	1.3970653	1.5158399	1.6346145	1.7533891	1.8548988	1.9046136	1.5298133
[37,]	1.6355713	1.7543459	1.8731205	1.9918951	2.0934048	2.1431196	1.6799145
[38,]	1.9174420	2.0362166	2.1549912	2.2737658	2.3752755	2.4249903	1.8733804
[39,]	2.2643598	2.3831344	2.5019090	2.6206836	2.7221933	2.7719081	2.1318934
[40,]	2.6763247	2.7950993	2.9138739	3.0326485	3.1341582	3.1838730	2.4554535
[41,]	3.1099719	3.2287465	3.3475211	3.4662958	3.5678054	3.6175202	2.8006959
[42,]	3.5869839	3.7057585	3.8245331	3.9433077	4.0448174	4.0945322	3.1893031
[43,]	4.0856782	4.2044528	4.3232275	4.4420021	4.5435117	4.5932265	3.5995927
[44,]	4.7252343	4.8115590	4.9303336	5.0491082	5.1506179	5.2003327	4.1182940
[45,]	5.5383965	5.6896210	5.7759458	5.8947204	5.9962300	6.0459448	4.8755014
[46,]	6.2583288	6.4377362	6.5889607	6.6752854	6.7767951	6.8265099	5.5676616
[47,]	6.9521644	7.3094450	7.4888524	7.6400769	7.7091367	7.7588515	6.4115984
[48,]	7.8845060	8.2417866	8.5990672	8.7784746	8.9124341	8.9296991	7.4940412
[49,]	8.8028171	9.1741282	9.5314088	9.8886894	10.0508319	10.1329965	8.5764840
[50,]	10.1904883	10.5911336	10.9624447	11.3197253	11.6597410	11.7700886	10.1576212
[51,]	12.2936775	12.6943228	13.0949681	13.4662792	13.8062949	14.0945157	12.4542763
[52,]	14.9606081	15.3612534	15.7618987	16.1625440	16.5165902	16.8048110	15.3146728
[53,]	17.8443623	18.2450076	18.6456529	19.0462983	19.4296787	19.7319300	18.3918929
[54,]	19.4855547	19.9145495	20.3151949	20.7158402	21.0992206	21.4308061	20.2549008
[55,]	22.0223912	22.4880836	22.9170784	23.3177237	23.7011041	24.0326896	23.0502502
[56,]	24.9061454	25.3718378	25.8375302	26.2665250	26.6499054	26.9814909	26.1925175
[57,]	27.6598055	28.1254979	28.5911903	29.0568827	29.4686126	29.8001981	29.2046905
[58,]	29.8714065	30.3370989	30.8027913	31.2684837	31.7169112	32.0768462	31.6748045
[59,]	31.9953555	32.4619704	32.9276628	33.3933552	33.8417827	34.2384153	34.0581891
[60,]	34.1197658	34.5859195	35.0525344	35.5182268	35.9666542	36.3632869	36.4415736
[61,]	36.0707172	36.5368708	37.0030245	37.4696394	37.9180669	38.3146995	38.5732125
[, 29] [, 30] [, 31] [, 32] [, 33] [, 34] [, 35]							
[1,]	0.1766057	0.2284005	0.2284005	0.2629304	0.2974603	0.3147253	0.2284005
[2,]	0.4616231	0.5134180	0.5652129	0.5997428	0.6688026	0.7205974	0.5652129
[3,]	0.7032759	0.7550708	0.8068656	0.8931904	0.9622502	1.0485749	0.8586605
[4,]	0.9232463	0.9750412	1.0268360	1.1131608	1.2340154	1.3203402	1.1304257
[5,]	1.1215343	1.1733292	1.2251240	1.3114488	1.4323034	1.5704230	1.3805086
[6,]	1.3198224	1.3716172	1.4234121	1.5097368	1.6305915	1.7687111	1.6305915
[7,]	1.4964280	1.5482229	1.6000177	1.6863425	1.8071971	1.9453167	1.8589920
[8,]	1.5863042	1.6380991	1.6898939	1.7762187	1.8970734	2.0351930	2.0006631
[9,]	1.6328157	1.6846106	1.7364054	1.8227302	1.9435848	2.0817045	2.0989694
[10,]	1.6793272	1.7311221	1.7829169	1.8692417	1.9900963	2.1282159	2.1800108
[11,]	1.6824740	1.7342688	1.7860637	1.8723885	1.9932431	2.1313627	2.1831576
[12,]	1.8094214	1.7576266	1.8094214	1.8266864	1.8327791	1.8439513	1.9191039
[13,]	1.8930041	1.7894144	1.7894144	1.7921565	1.8294284	1.8839653	1.8757391
[14,]	1.9982691	1.8428846	1.8428846	1.8083547	1.8077461	1.8406006	1.8940708
[15,]	2.1468989	1.9397195	1.9397195	1.8706597	1.8355212	1.8182562	1.9150912
[16,]	1.4791074	1.6409894	1.8028714	1.9302235	1.9283434	1.8938135	1.9801382
[17,]	1.3490132	1.4458482	1.5426831	1.6049881	1.6327632	1.6432734	1.7401083
[18,]	1.3490132	1.4458482	1.5426831	1.6049881	1.6327632	1.6432734	1.7401083
[19,]	1.4374180	1.4892129	1.5860478	1.6483528	1.6761279	1.6866381	1.7834730
[20,]	1.5475052	1.5993000	1.6510949	1.7133999	1.7411750	1.7516852	1.8485201
[21,]	1.5725917	1.7093872	1.7611821	1.7784470	1.8062221	1.8167323	1.9135672
[22,]	1.3666829	1.5502473	1.7338116	1.8828461	1.8929516	1.9034617	2.0002967
[23,]	1.3666829	1.5502473	1.7338116	1.8828461	1.9973507	1.9901912	2.0870261
[24,]	1.2514971	1.5001086	1.7487201	1.9478932	2.0623978	2.1524779	2.2388027

Lampiran 4. Command R Contoh Perhitungan Jarak Dynamic Time Warping (lanjutan)

[25,]	1.2514971	1.5001086	1.7487201	1.9628016	2.1274448	2.2246845	2.4010894
[26,]	1.3815913	1.6302028	1.8788142	2.0928958	2.2724475	2.4198257	2.6033901
[27,]	1.4032737	1.6518851	1.9004966	2.1145782	2.2941298	2.4564165	2.6901196
[28,]	1.0563559	1.1315084	1.2066610	1.2472837	1.2533764	1.2645486	1.3397012
[29,]	0.9976023	1.1175910	1.2375797	1.3611796	1.4363322	1.4596899	1.3845373
[30,]	1.0378203	1.0091792	1.0207561	1.0668629	1.1474996	1.2454013	1.2569781
[31,]	1.0780382	1.0207561	1.0207561	1.0668629	1.1474996	1.2454013	1.2569781
[32,]	1.1182562	1.0323330	1.0323330	1.0668629	1.1474996	1.2454013	1.2569781
[33,]	1.2885683	1.1508503	1.1508503	1.1163204	1.1163204	1.1485129	1.2670302
[34,]	1.4805628	1.2910499	1.2910499	1.2219901	1.1874602	1.1701953	1.2887126
[35,]	1.5083348	1.3011554	1.3011554	1.2464146	1.2464146	1.2464146	1.1803008
[36,]	1.5500242	1.3112609	1.3112609	1.2708390	1.3053689	1.3226339	1.1904062
[37,]	1.5763248	1.4095672	1.4095672	1.4036752	1.4382051	1.4900000	1.2887126
[38,]	1.6662010	1.5512383	1.5512383	1.5798762	1.6144061	1.6662010	1.4303836
[39,]	1.8211243	1.7579564	1.7579564	1.7924863	1.8556542	1.9074490	1.6371018
[40,]	2.0410947	2.0297217	2.0297217	2.0642516	2.1333114	2.2137442	1.9088670
[41,]	2.2827474	2.3231693	2.3231693	2.3576992	2.4267590	2.5130838	2.2023146
[42,]	2.5677649	2.6195598	2.6599816	2.6945115	2.7635713	2.8498961	2.5391270
[43,]	2.8744648	2.9262596	2.9780545	3.0530062	3.1220660	3.2083908	2.8976217
[44,]	3.2895764	3.3413713	3.3931661	3.4794909	3.5889725	3.6752973	3.3645282
[45,]	3.9431941	3.9949889	4.0467838	4.1331085	4.2539632	4.3807098	4.0699407
[46,]	4.5317646	4.5835595	4.6353543	4.7216791	4.8425337	4.9806533	4.7103061
[47,]	5.2721117	5.3239066	5.3757014	5.4620262	5.5828808	5.7210004	5.5024480
[48,]	6.2509648	6.3027596	6.3545545	6.4408793	6.5617339	6.6998535	6.5330960
[49,]	7.2298179	7.2816127	7.3334076	7.4197323	7.5405870	7.6787066	7.5637439
[50,]	8.7073653	8.7591602	8.8109550	8.8972798	9.0181344	9.1562540	9.0930862
[51,]	10.9004307	10.9522256	11.0040204	11.0903452	11.2111998	11.3493194	11.3379464
[52,]	13.6572375	13.7090324	13.7608272	13.8471520	13.9680066	14.1061262	14.1465481
[53,]	16.6308680	16.6826628	16.7344577	16.8207824	16.9416371	17.0797567	17.1315515
[54,]	18.3902861	18.4420810	18.4938758	18.5802006	18.7010552	18.8391748	18.8909697
[55,]	21.0820459	21.1338407	21.1856356	21.2719603	21.3928150	21.5309346	21.5827294
[56,]	24.1207234	24.1725183	24.2243131	24.3106379	24.4314925	24.5696121	24.6214070
[57,]	27.0293068	27.0811016	27.1328965	27.2192212	27.3400759	27.4781955	27.5299903
[58,]	29.3958311	29.4476259	29.4994208	29.5857455	29.7066002	29.8447198	29.8965146
[59,]	31.6756259	31.7274208	31.7792156	31.8655404	31.9863950	32.1245146	32.1763095
[60,]	33.9554208	34.0072156	34.0590105	34.1453352	34.2661899	34.4043095	34.4561043
[61,]	36.0617567	36.1135515	36.1653464	36.2516712	36.3725258	36.5106454	36.5624403
[1,]	[,36]	[,37]	[,38]	[,39]	[,40]	[,41]	[,42]
[1,]	0.1593407	0.07301596	0.02122111	0.03057374	0.1859583	0.2377531	0.3068129
[2,]	0.4270932	0.25444374	0.15085404	0.09905919	0.1081202	0.2374615	0.4358627
[3,]	0.6514811	0.39250679	0.23712224	0.13353254	0.2199704	0.2808263	0.4792274
[4,]	0.8541865	0.50888748	0.30170808	0.14632352	0.2761261	0.4143588	0.5442745
[5,]	1.0352096	0.60358581	0.34461156	0.15521490	0.3105995	0.4921969	0.6994894
[6,]	1.2162327	0.69828414	0.38751503	0.16410628	0.3194908	0.5266702	0.7773275
[7,]	1.3755734	0.77130010	0.40873614	0.19468001	0.3500646	0.5572440	0.8334832
[8,]	1.4481846	0.78501359	0.47424448	0.31198320	0.4673678	0.6745472	0.9507864
[9,]	1.4774312	0.84209180	0.58311755	0.47265112	0.6280357	0.8352151	1.1114543
[10,]	1.5066777	0.89917001	0.69199061	0.63331903	0.7887036	0.9958830	1.2721222
[11,]	1.5207959	0.99961295	0.84422840	0.83735167	0.9927362	1.1999156	1.4761548
[12,]	1.6650083	1.23015006	1.12656036	1.17147849	1.3268630	1.5340425	1.8102817
[13,]	1.7658559	1.41732245	1.36552760	1.41732245	1.6176251	1.8248045	2.1010438
[14,]	1.8883859	1.62617720	1.62617720	1.67797205	1.8851515	2.1372490	2.4134882
[15,]	2.0542806	1.87839667	1.93019152	1.98198638	2.1891658	2.4481400	2.7692974
[16,]	2.1460330	2.19566324	2.24745809	2.35104779	2.5582272	2.8172015	3.1452355
[17,]	1.9060030	2.15822251	2.46223683	2.60326727	2.8622415	3.1212158	3.4492498
[18,]	1.9060030	2.15822251	2.46223683	2.81804601	3.1144610	3.4252301	3.7532642
[19,]	1.9493678	2.20158723	2.50560156	2.86141074	3.3726045	3.7208143	4.1006432
[20,]	2.0144148	2.26663432	2.57064865	2.92645782	3.4376516	4.0006401	4.4179098
[21,]	2.0794619	2.33168141	2.63569573	2.99150491	3.5026986	4.0656872	4.6977356
[22,]	2.1661914	2.41841086	2.72242519	3.07823436	3.5894281	4.1524167	4.7844651
[23,]	2.2529208	2.50514031	2.80915464	3.16496381	3.6761575	4.2391461	4.8711945
[24,]	2.4046974	2.65691685	2.96093118	3.31674035	3.8279341	4.3909227	5.0229711
[25,]	2.5564739	2.80869339	3.11270771	3.46851689	3.9797106	4.5426992	5.1747476
[26,]	2.8488548	3.09056410	3.39457843	3.75038761	4.2615813	4.8245699	5.4566183
[27,]	2.9427437	3.27453317	3.56803733	3.92384651	4.4350402	4.9980288	5.6300772
[28,]	1.4839135	1.71445064	1.99678260	2.33090942	2.8204208	3.3617270	3.9720930
[29,]	1.3906301	1.42602592	1.51321662	1.65220217	1.9465723	2.2927372	2.7079620
[30,]	1.3144611	1.45826872	1.62162843	1.76061399	2.0549841	2.4011491	2.8163738
[31,]	1.3144611	1.45826872	1.65387123	1.86902580	2.1633959	2.5095609	2.9247856
[32,]	1.3144611	1.45826872	1.65387123	1.90126860	2.2718077	2.6179727	3.0331974
[33,]	1.4445552	1.58836289	1.78396541	2.03136277	2.4341447	2.8564787	3.2717034
[34,]	1.4762897	1.74013943	1.93574195	2.18313931	2.5859212	3.0404980	3.5318918
[35,]	1.2594660	1.42495605	1.64224093	1.91132066	2.3357849	2.8120441	3.3573630
[36,]	1.2594660	1.42495605	1.64224093	1.91132066	2.3357849	2.8120441	3.3573630
[37,]	1.2196528	1.27673098	1.38560404	1.54627196	1.8623244	2.2301717	2.6670789
[38,]	1.2922640	1.23336625	1.29887459	1.41617778	1.6888655	2.0133481	2.4068905

Lampiran 4. Command R Contoh Perhitungan Jarak *Dynamic Time Warping* (lanjutan)

[39,]	1.4299224	1.28469985	1.23382750	1.28608360	1.4937243	1.7531598	2.0816551
[40,]	1.6326278	1.40108054	1.29841334	1.24661849	1.3892121	1.5836005	1.8470487
[41,]	1.8570156	1.53914360	1.38468154	1.28109184	1.3675297	1.5402358	1.7820016
[42,]	2.1247682	1.72057137	1.51431447	1.35892991	1.3586383	1.4879796	1.6863808
[43,]	2.4142031	1.92368151	1.66562976	1.45845035	1.4145024	1.4662973	1.6430161
[44,]	2.8120498	2.23520347	1.92535686	1.66638260	1.4670501	1.4152553	1.4835622
[45,]	3.4484025	2.78523141	2.42358995	2.11282084	1.7581038	1.6545141	1.5854543
[46,]	4.0197081	3.27021227	2.85677596	2.49421199	1.9841104	1.8287259	1.6906063
[47,]	4.7427902	3.90696966	3.44173850	3.02737968	2.3618935	2.1547141	1.9475347
[48,]	5.7043784	4.78223304	4.26520703	3.79905336	2.9781827	2.7192084	2.4429692
[49,]	6.6659665	5.65749642	5.08867556	4.57072704	3.5944718	3.2837027	2.9384037
[50,]	8.1262490	7.03145415	6.41083843	5.84109506	4.7094553	4.3468913	3.9325325
[51,]	10.3020494	9.12092984	8.44851927	7.82698105	6.5399567	6.1255979	5.6421793
[52,]	13.0415913	11.77414696	11.04994154	10.37660847	8.9341996	8.4680459	7.9155675
[53,]	15.9979568	14.64418771	13.86818744	13.14305951	11.5452660	11.0273175	10.4057793
[54,]	17.7401100	16.30001615	15.47222103	14.69529825	12.9421202	12.3723769	11.8817788
[55,]	20.4146048	18.88818618	18.00859621	17.17987858	15.2713160	14.6497778	13.8901200
[56,]	23.4360173	21.82327401	20.89188919	20.01137671	17.9474296	17.2740965	16.4453789
[57,]	26.3273358	24.62826767	23.64508800	22.71278066	20.4934490	19.7683211	18.8705436
[58,]	28.6765951	26.89120226	25.85622774	24.87212555	22.4974093	21.7204865	20.7536493
[59,]	30.9391250	29.06740741	27.98063803	26.94474099	24.4146402	23.5859226	22.5500255
[60,]	33.2016549	31.24361255	30.10504832	29.01735643	26.3318711	25.4513586	24.3464018
[61,]	35.2907259	33.24635879	32.05599971	30.91651297	28.0756431	27.1433357	25.9693191
[,43]	[,44]	[,45]	[,46]	[,47]	[,48]	[,49]	
[1,]	0.3758727	0.42766676	0.5485223	0.6003171	0.7384367	0.8765563	0.8938213
[2,]	0.5742739	0.6951285	0.8677780	1.0404275	1.2303420	1.5065812	1.6619658
[3,]	0.7466883	0.9368944	1.1786037	1.4030481	1.7138172	2.0418512	2.3353554
[4,]	0.8117354	1.1309912	1.4420519	1.7355561	2.0981200	2.5470088	2.8923078
[5,]	0.8984649	1.2177207	1.6578311	2.0206867	2.4523104	2.9529940	3.4191477
[6,]	1.0536798	1.3044501	1.7445606	2.2364659	2.7374410	3.3071844	3.8251329
[7,]	1.1532002	1.4813474	1.8529724	2.3448777	2.9749026	3.6139973	4.2010057
[8,]	1.2960854	1.6675973	2.1165991	2.5400189	3.1700438	3.9381883	4.5945481
[9,]	1.4567533	1.8538472	2.3462137	2.8470104	3.4085498	4.1766943	4.9621038
[10,]	1.6174212	2.0145151	2.5324636	3.0766250	3.7155413	4.4152003	5.2006098
[11,]	1.8214539	2.2185477	2.7364962	3.3062396	3.9885206	4.7655565	5.4824805
[12,]	2.1555807	2.5526745	3.0706230	3.6403664	4.3482294	5.1686300	5.9629308
[13,]	2.4463428	2.8434366	3.3613851	3.9311285	4.6389915	5.4849741	6.3226396
[14,]	2.7587872	3.1558811	3.6738296	4.2435730	4.9514359	5.7974185	6.6606660
[15,]	3.1145964	3.5116903	4.0296388	4.5993821	5.3072451	6.1532277	7.0164752
[16,]	3.5354527	3.9325465	4.4504950	5.0202384	5.7281014	6.5740840	7.4373315
[17,]	3.8463437	4.2883557	4.8063042	5.3760476	6.0839106	6.9298931	7.7931407
[18,]	4.1503580	4.5992467	5.1621134	5.7318568	6.4397197	7.2857023	8.1489498
[19,]	4.4977371	4.9466258	5.5163692	6.1310307	6.8388936	7.6848762	8.5481237
[20,]	4.8667985	5.3156872	5.8854306	6.5069688	7.2597499	8.1057325	8.9689800
[21,]	5.1840651	5.6847486	6.2544920	6.8760302	7.6356880	8.5265887	9.3898363
[22,]	5.4855733	6.0236976	6.6452358	7.2667740	8.0264318	8.9242092	9.8323749
[23,]	5.75723027	6.3252057	6.9841847	7.6575178	8.4171756	9.3149530	10.2299954
[24,]	5.7240792	6.4769823	7.3507400	8.0615138	8.8729665	9.7707439	10.6857863
[25,]	5.8758558	6.6287588	7.5025165	8.4280691	9.2769625	10.2265348	11.1415771
[26,]	6.1577265	6.9106295	7.7843872	8.7099398	9.7736119	10.7606249	11.7274622
[27,]	6.3311854	7.0840884	7.9578461	8.8833987	9.9470708	11.1488626	12.1531405
[28,]	4.6515189	5.3827395	6.2348149	7.1386851	8.1806748	9.3607842	10.5581586
[29,]	3.1922466	3.7283260	4.3852600	5.0939890	5.9408375	6.9258056	7.9280387
[30,]	3.3006584	3.8367378	4.4936719	5.2024008	6.0492493	7.0342174	8.0364505
[31,]	3.4090702	3.9451496	4.6020837	5.3108126	6.1576611	7.1426292	8.1448623
[32,]	3.5174820	4.0535614	4.7104955	5.4192244	6.2660729	7.2510411	8.2532741
[33,]	3.7559880	4.2920674	4.9490015	5.6577304	6.5045789	7.4895470	8.4917801
[34,]	4.0161763	4.5522558	5.2091898	5.9179187	6.7647673	7.7497354	8.7519685
[35,]	3.9717417	4.6379153	5.3392840	6.0480129	6.8948614	7.8798296	8.8820627
[36,]	3.9717417	4.6379153	5.4249436	6.1781071	7.0249556	8.0099237	9.0121568
[37,]	3.1730458	3.7308076	4.4094240	5.1398353	6.0083662	7.0150167	8.0389321
[38,]	2.8694927	3.3838898	4.0191415	4.7061880	5.5313542	6.4946400	7.4751907
[39,]	2.4792102	2.9285602	3.4987648	4.1207642	4.8808833	5.7791220	6.6946256
[40,]	2.1795567	2.5638596	3.0690171	3.6259695	4.3210415	5.1542331	6.0046896
[41,]	2.0928273	2.4554478	2.9389230	3.4741930	4.1475826	4.9590918	5.7878660
[42,]	1.9538417	2.2730975	2.7132079	3.2051132	3.8351381	4.6032826	5.3886921
[43,]	1.8887946	2.1863681	2.6047961	3.0750191	3.6833616	4.4298237	5.1935508
[44,]	1.6209290	1.8100906	2.1201069	2.4819180	2.9818487	3.6198990	4.2752143
[45,]	1.5847015	1.6340458	1.7055561	1.8288612	2.0902860	2.4898303	2.9066396
[46,]	1.6207936	1.6004042	1.7369615	1.8939083	2.1553331	2.5548774	2.9716867
[47,]	1.8086623	1.7364780	1.6156234	1.6521990	1.8268943	2.1397092	2.4697890
[48,]	2.2350370	2.1110578	1.8693485	1.8175537	1.7160097	1.7903186	1.8818925
[49,]	2.6614116	2.4856376	2.1230737	2.0194840	1.7798204	1.7903186	1.8818925
[50,]	3.5864806	3.3589118	2.8754932	2.7201086	2.3423255	2.2042059	2.1974391
[51,]	5.2270676	4.9477039	4.3434307	4.1362512	3.6203485	3.3441093	3.3200776
[52,]	7.4313960	7.1002375	6.3751096	6.1161353	5.4621130	5.0477541	5.0064575

Lampiran 4. Command R Contoh Perhitungan Jarak Dynamic Time Warping (lanjutan)

[53,]	9.8525481	9.4695947	8.6236121	8.3128430	7.5207010	6.9682226	6.9096610
[54,]	11.0594878	10.6247395	9.6579023	9.2953383	8.3650768	7.6744788	7.5986522
[55,]	13.1987691	12.7122260	11.6245341	11.2101753	10.1417941	9.3130765	9.2199850
[56,]	15.6849682	15.1466303	13.9380837	13.4719301	12.2654293	11.2985921	11.1882356
[57,]	18.0410732	17.4509404	16.1215392	15.6035907	14.2589703	13.1540135	13.0263920
[58,]	19.8551190	19.2131914	17.7629355	17.1931922	15.7104522	14.4673758	14.3224894
[59,]	21.5824355	20.8887130	19.3176025	18.6960642	17.0752047	15.6940086	15.5318573
[60,]	23.3097519	22.5642345	20.8722694	20.1989363	18.4399571	16.9206415	16.7412252
[61,]	24.8636094	24.0662972	22.2534774	21.5283495	19.6312507	17.9738154	17.7771342
	[,50]	[,51]	[,52]	[,53]	[,54]	[,55]	[,56]
[1,]	1.532624	1.774334	1.929718	2.119633	2.067838	2.637581	2.689376
[2,]	2.318034	3.198546	3.595640	3.940939	4.027264	4.597007	5.218546
[3,]	3.129543	4.027321	5.063218	5.650226	5.943730	6.599798	7.221336
[4,]	3.824615	4.860512	5.913674	7.139486	7.674699	8.537947	9.245810
[5,]	4.403250	5.577266	6.768548	8.011625	9.185641	10.290598	11.205640
[6,]	4.930090	6.155901	7.485302	8.866498	10.057780	11.801540	12.958292
[7,]	5.357757	6.704423	8.085620	9.604935	10.934336	12.695361	14.490916
[8,]	5.820360	7.218821	8.720871	10.291982	11.759503	13.658647	15.471467
[9,]	6.257267	7.724787	9.278633	10.970598	12.489914	14.527178	16.478117
[10,]	6.624822	8.161695	9.784600	11.528360	13.168530	15.257589	17.346648
[11,]	6.906693	8.572615	10.264872	12.077692	13.769657	15.979570	18.120424
[12,]	7.318658	8.984580	10.805886	12.688058	14.449083	16.710791	18.972500
[13,]	7.755744	9.353180	11.174487	13.185708	15.016084	17.346852	19.660356
[14,]	8.137135	9.811948	11.564769	13.575990	15.535416	17.935536	20.318099
[15,]	8.518526	10.236704	12.066902	14.009637	15.969063	18.498233	20.950147
[16,]	8.939382	10.683142	12.556705	14.576817	16.467758	18.996927	21.577891
[17,]	9.295191	11.038951	12.938096	15.001573	16.969890	19.430574	22.011539
[18,]	9.651001	11.394761	13.293905	15.382964	17.394646	19.932707	22.445186
[19,]	10.050174	11.793934	13.693079	15.782138	17.819402	20.400828	22.990683
[20,]	10.471031	12.214791	14.113935	16.202994	18.240258	20.847266	23.480486
[21,]	10.891887	12.635647	14.534792	16.623851	18.661115	21.268122	23.926925
[22,]	11.334426	13.078186	14.977330	17.066389	19.103653	21.710661	24.369463
[23,]	11.776964	13.520724	15.419869	17.508928	19.546192	22.153200	24.812002
[24,]	12.239632	14.028310	15.927455	18.016514	20.053778	22.660785	25.319588
[25,]	12.695423	14.490978	16.435040	18.524099	20.561363	23.168371	25.827173
[26,]	13.281308	15.076863	17.027802	19.161779	21.199043	23.806051	26.464853
[27,]	13.758781	15.554336	17.505275	19.646129	21.728311	24.335319	26.994121
[28,]	14.239436	16.047223	18.005494	20.128680	22.250071	24.861205	27.528248
[29,]	14.729075	16.541821	18.518951	20.617995	22.779425	25.390238	28.063933
[30,]	15.228747	17.042332	19.038362	21.118647	23.302657	25.914638	28.600345
[31,]	15.738599	17.548644	19.554774	21.634819	23.829069	26.440502	29.127757
[32,]	16.258410	18.060556	20.072331	22.162480	24.361573	26.972621	29.655169
[33,]	16.788216	18.583062	20.600937	22.691193	24.894286	27.505286	30.184775
[34,]	17.328030	19.106174	21.130648	23.220007	25.427000	28.037999	30.714391
[35,]	17.877844	19.639286	21.663760	23.750828	25.959714	28.570712	31.244007
[36,]	18.437658	20.172398	22.196871	24.281649	26.492428	29.103426	31.773623
[37,]	18.997472	20.705510	22.727982	24.812470	27.025142	29.636140	32.303239
[38,]	19.567286	21.238622	23.259093	25.343291	27.557856	30.168854	32.832855
[39,]	20.137100	21.771734	23.790204	25.874102	28.089569	30.701568	33.362471
[40,]	20.706914	22.304846	24.321315	26.404913	28.620283	31.234282	33.892087
[41,]	21.276728	22.837958	24.852426	26.935724	29.151000	31.766996	34.421703
[42,]	21.846542	23.371070	25.383537	27.466535	29.681714	32.299710	34.951319
[43,]	22.416356	23.904182	25.914648	27.997346	30.212428	32.832424	35.480935
[44,]	22.986170	24.437294	26.445759	28.528157	30.743142	33.365138	36.010551
[45,]	23.555984	24.970406	26.976870	29.058968	31.273856	33.897852	36.540167
[46,]	24.125798	25.503518	27.507981	29.589779	31.804570	34.430566	37.069783
[47,]	24.695612	26.036630	28.039092	30.120590	32.335284	34.961280	37.600399
[48,]	25.265426	26.569742	28.570203	30.651701	32.866000	35.492000	38.131015
[49,]	25.835240	27.102854	29.101314	31.182812	33.396714	36.022714	38.661631
[50,]	26.405054	27.635966	29.632425	31.713923	33.927428	36.553428	39.192247
[51,]	26.974868	28.169078	30.163536	32.245034	34.458142	37.084142	39.722863
[52,]	27.544682	28.702190	30.694647	32.776145	34.988856	37.614856	40.253479
[53,]	28.114496	29.235302	31.225758	33.307256	35.519570	38.145570	40.784095
[54,]	28.684310	29.768414	31.756869	33.838367	36.050284	38.676284	41.314711
[55,]	29.254124	30.301526	32.287980	34.369478	36.580998	39.206998	41.845327
[56,]	29.823938	30.834638	32.819091	34.900589	37.111712	39.737712	42.375943
[57,]	30.393752	31.367750	33.350202	35.431700	37.642426	40.268426	42.906559
[58,]	30.963566	31.900862	33.881313	35.962811	38.173140	40.799140	43.437175
[59,]	31.533380	32.433974	34.412424	36.493922	38.703854	41.329854	43.967791
[60,]	32.103194	32.967086	34.943535	37.025033	39.234568	41.860568	44.498407
[61,]	32.673008	33.500198	35.474646	37.556144	39.765282	42.391282	45.029023
	[,58]	[,59]	[,60]	[,61]			
[1,]	2.723906	2.275017	2.154163	1.739804			
[2,]	5.339400	4.441623	4.199913	3.371196			
[3,]	7.998259	6.651593	6.289029	5.045953			
[4,]	10.661536	8.883246	8.399827	6.742392			

Lampiran 4. Command R Contoh Perhitungan Jarak *Dynamic Time Warping* (lanjutan)

```
[5,] 12.707691 11.136581 10.532307 8.460513
[6,] 14.753847 13.389916 12.664788 10.178635
[7,] 16.735360 15.664933 14.818950 11.918439
[8,] 18.596423 18.026680 17.059842 13.744972
[9,] 20.259142 20.431791 19.344099 15.614870
[10,] 21.283057 22.664253 21.628356 17.484768
[11,] 22.333072 23.731533 23.955978 19.398031
[12,] 23.375062 24.911643 26.189249 21.441388
[13,] 24.235567 25.910268 27.325994 23.441380
[14,] 25.065960 26.792455 28.346301 25.463054
[15,] 25.818863 27.666213 29.271854 27.528094
[16,] 26.585019 28.484163 30.210658 29.658180
[17,] 27.217067 29.185271 30.963561 31.723219
[18,] 27.779764 29.817320 31.664670 33.028601
[19,] 28.256776 30.423382 32.340083 33.773074
[20,] 28.755470 30.922076 32.967827 34.470169
[21,] 29.322650 31.420770 33.466521 35.097913
[22,] 29.834136 32.009632 33.986898 35.618290
[23,] 30.302256 32.521118 34.575760 36.138667
[24,] 30.809842 33.054285 35.152292 36.792576
[25,] 31.317428 33.561871 35.685460 37.369108
[26,] 31.955107 34.199551 36.323140 38.032370
[27,] 32.484376 34.728819 36.852408 38.561638
[28,] 32.818502 35.062946 37.186535 38.895765
[29,] 29.206251 31.589680 33.852255 35.700471
[30,] 29.314663 31.698092 33.960667 35.808882
[31,] 29.423075 31.806504 34.069079 35.917294
[32,] 29.531487 31.914916 34.177490 36.025706
[33,] 29.769993 32.153422 34.415996 36.264212
[34,] 30.030181 32.413610 34.676185 36.524400
[35,] 30.160275 32.543704 34.806279 36.654495
[36,] 30.290369 32.673799 34.936373 36.784589
[37,] 29.512286 31.917397 34.201654 36.071552
[38,] 28.558262 30.920009 33.160901 34.987434
[39,] 27.192273 29.488973 31.664818 33.426304
```


Lampiran 5. Command R untuk Jarak Dynamic Time Warping Keseluruhan

```

> library(dtw)
> library(clusterSim)
> library(cluster)
> library(fpc)
> library(tidyverse)
> library(factoextra)
> library(dendextend)
> library(kmed)
> library(writexl)
> standarisasi=read.csv(file.choose(),header=TRUE,sep=",")
> matriks<-cbind(standarisasi[,-1])
> rownames(matriks) = standarisasi$PROV
> datascale<-matriks
> #Dynamic Time Warping Distance
> dtw_distance=dist(datascale,method="DTW")
> dtw_distance<-as.matrix(dtw_distance)
> dtw_distance=as.data.frame(dtw_distance)
> dtw_distance
      1          2          3          4          5          6
2  6.587717
3 11.734999 10.424612
4  9.288661  8.712210  7.794012
5 16.571216 15.168150 14.718072 10.873779
6 12.485603  8.935065  7.926371  7.155165 11.094381
7 14.631814 13.425053 14.139021 14.406620 13.100954 11.237303
8 13.135418 10.017370  9.544450  8.305255  8.491646  7.820225
9 15.412170 14.844000 15.253914 12.527580  9.219551 15.583924
10  6.965916  5.787531  9.691589  7.740107 17.517512  9.702240
11  9.228286  7.542009  6.963939  7.369943 11.497818  8.891444
12 12.087852  8.242304  9.343021  7.252808  9.009571  6.557203
13  9.017982  6.941820  8.516827  8.856533 14.071536  8.889055
14 11.080264  8.255625  8.481941  7.488560 10.028791  6.112226
15  9.583586  6.714331  8.588641  9.187534 13.971257  7.797821
16 11.236200  8.343914  8.640030  7.545536  8.679490  6.321454
17  6.972374  7.068444  9.375960  8.455364 13.527718  8.731011
18 11.589093 10.275939 10.346528 10.144379 12.304042  8.971968
19 13.356769 13.123161 15.993579 12.889700 14.510859 13.209973
20 10.200187  8.957239  9.337464  8.438565 11.450804  8.893665
21  6.854467  5.367432  7.547835  7.412707 13.614560  8.494159
22 11.477037 10.465721 13.646395 10.615121 12.600403 12.699182
23 14.159537 15.097925 13.240935 11.406947 11.251657 11.909064
24 11.694363 10.392958 12.884965 11.661693 11.012354  9.670510
25 12.278098 10.942612 11.673202  9.890976 12.705844 10.050914
26  8.138163  9.038756 12.411393 10.292427 14.716104 10.670442
27 20.269139 20.911479 20.612439 18.986772 17.450026 19.432549
28 13.867548 12.018896 10.387265  9.697933 10.579909  9.307255
29 11.178136 11.423887 13.512265 12.101072 13.460809 11.761630
30 13.213384 11.901278 11.462640 10.730342 10.768879  9.856931
31 12.666697 12.118244 12.162433 11.683895 11.772674 10.916553
32 14.383423 14.521590 18.302158 14.231247 17.754689 15.886656
33 11.546568 13.562435 15.707730 13.116115 13.848150 12.774758
34 13.263443 15.459114 15.880363 12.547683 16.689808 15.914611

```

Lampiran 5. Command R untuk Jarak Dynamic Time Warping Keseluruhan (lanjutan)

	7	8	9	10	11	12
8	12.234169					
9	14.037380	11.062702				
10	14.852795	11.322655	17.037852			
11	11.809330	7.186816	12.253653	7.967763		
12	12.370653	5.873624	10.323498	9.661927	7.423262	
13	11.223491	11.023583	15.492052	7.518353	6.729687	6.993340
14	11.821701	7.257420	12.058415	8.087563	6.484171	4.830936
15	12.126951	10.866714	16.173868	8.201079	7.920113	8.288327
16	12.207556	5.654545	9.572628	9.250683	6.445220	4.414567
17	12.503933	9.839441	13.621145	7.616592	7.394668	9.390708
18	12.007526	11.596694	11.807915	11.105150	9.753415	7.922979
19	12.728382	13.269761	13.387378	14.405122	12.891541	11.826120
20	12.722371	7.776810	9.471195	10.149929	7.559773	7.562178
21	13.124858	10.308849	13.338931	6.407468	5.581934	7.669704
22	15.469452	12.093474	10.386971	13.070015	11.338230	9.942044
23	14.711103	11.709579	11.863239	16.437783	12.308366	10.799141
24	11.089786	11.054504	11.071251	12.599409	9.754744	9.783676
25	12.873200	11.422202	12.597016	10.825808	8.509441	8.332607
26	12.751786	11.446302	13.637136	9.194678	9.662043	9.909185
27	20.816919	18.881383	16.056409	23.044628	22.335650	18.476918
28	12.640640	9.756662	10.011846	13.184927	9.646734	8.343193
29	12.001010	13.445341	11.625800	14.096589	9.232576	10.299272
30	13.781135	10.446076	9.163855	12.939360	10.022706	8.078790
31	11.731300	12.551970	12.498402	12.717803	12.439261	10.534867
32	17.391216	18.192592	16.538849	15.913938	16.736995	17.445650
33	15.376209	12.839101	12.218574	16.247470	14.276920	10.946706
34	18.368241	16.485430	16.505632	16.721594	15.237020	15.170674
	13	14	15	16	17	18
14	5.940526					
15	5.708107	7.022322				
16	7.645990	3.735386	8.350954			
17	7.275676	7.562520	8.246541	8.341943		
18	7.697194	7.342558	9.495451	7.879089	8.399834	
19	11.174689	12.194104	13.727896	12.372794	10.093814	10.015998
20	9.479244	7.337159	9.817444	6.302354	7.564048	7.906554
21	5.560951	6.995895	6.380798	7.124486	5.652042	7.892561
22	11.140496	9.915590	13.704863	9.373596	9.331252	7.356581
23	13.291162	9.985189	14.291755	8.912261	11.156039	9.754594
24	8.543027	8.110083	11.630660	9.354522	8.177120	5.934041
25	8.380979	8.522566	11.018914	8.112506	8.885884	7.611883
26	7.833017	8.792338	10.667629	9.389482	6.845601	8.340337
27	22.914128	19.106270	25.134583	17.946995	19.684540	19.427335
28	11.342412	8.899913	11.870301	7.435800	10.460220	8.546140
29	10.235598	10.214712	11.474563	10.372342	10.210662	9.128223
30	11.257145	8.313255	11.951419	7.839090	10.605460	9.191498
31	12.176383	8.904846	12.312203	10.143679	9.750408	9.895247
32	15.995088	15.908033	18.246464	15.252756	11.744166	14.847866
33	13.805597	11.396583	14.905032	11.670918	11.900010	10.851854
34	15.949448	14.974273	17.633036	14.332514	12.542933	13.407455

Lampiran 5. Command R untuk Jarak *Dynamic Time Warping* Keseluruhan (lanjutan)

	19	20	21	22	23	24
20	10.344913					
21	11.424972	7.021462				
22	7.692279	7.912004	8.752818			
23	11.454249	8.928966	12.323731	8.209476		
24	7.940050	7.840721	9.068842	7.773741	10.175699	
25	12.667979	9.833041	8.802583	12.223146	12.909666	9.138307
26	9.025588	9.211316	7.608210	8.510574	11.459358	9.216152
27	17.445175	17.369848	20.155462	15.566220	15.935785	19.542123
28	10.371528	7.075156	9.522926	9.569590	9.303451	9.662387
29	8.962778	9.659711	9.258577	10.099947	11.769987	8.051149
30	11.071371	7.687317	10.224766	7.927663	8.874944	10.255013
31	10.256327	9.156840	10.091909	7.606282	8.931522	10.470088
32	14.225692	14.604964	13.511983	10.146836	12.277358	12.334581
33	9.134033	9.485804	13.169773	9.267551	9.505835	9.023365
34	10.455230	11.953853	13.293686	8.793479	11.075911	14.623101
	25	26	27	28	29	30
26	11.407884					
27	23.376860	18.933159				
28	10.180303	11.199412	14.303349			
29	10.954155	10.440815	20.123902	11.219487		
30	10.430811	11.225418	13.191385	8.194026	9.533641	
31	13.187356	8.741427	14.558792	8.514613	12.059326	7.699557
32	16.398377	11.760355	20.968051	14.180750	16.720225	15.186889
33	15.144380	10.381722	15.573410	11.073298	11.755434	9.241548
34	18.531654	10.765593	16.895016	12.463752	13.844671	13.234925
	31	32	33			
32	10.771120					
33	7.136318	13.247414				
34	9.840697	11.974838	10.966097			

Lampiran 6. Command R Algoritma K-Medoids Clustering dengan $K = 2$

```

> k2<- pam(dtw_distance,2)
> k2
Medoids:
              ID      ACEH SUMATERA UTARA SUMATERA BARAT
BANTEN          11  9.228286      7.542009      6.963939 ...
KALIMANTAN TIMUR 23 14.159537      15.097925      13.240935 ...
Clustering vector:
              ACEH      SUMATERA UTARA
              1              1
SUMATERA BARAT              RIAU
              1              1
KEPULAUAN RIAU              JAMBI
              2              1
BENGKULU      SUMATERA SELATAN
              2              1
KEPULAUAN BANGKA BELITUNG      LAMPUNG
              2              1
BANTEN              JAWA BARAT
              1              1
DKI JAKARTA      JAWA TENGAH
              1              1
DI YOGYAKARTA      JAWA TIMUR
              1              1
BALI      NUSA TENGGARA BARAT
              1              1
NUSA TENGGARA TIMUR      KALIMANTAN BARAT
              2              1
KALIMANTAN SELATAN      KALIMANTAN TENGAH
              1              2
KALIMANTAN TIMUR      KALIMANTAN UTARA
              2              2
GORONTALO      SULAWESI SELATAN
              1              1
SULAWESI TENGGARA      SULAWESI TENGAH
              2              2
SULAWESI UTARA      SULAWESI BARAT
              2              2
MALUKU      MALUKU UTARA
              2              2
PAPUA      PAPUA BARAT
              2              2

Objective function:
      build      swap
18.55505 17.53629

Available components:
[1] "medoids"      "id.med"      "clustering" "objective"
"isolation"
[6] "clusinfo"      "silinfo"      "diss"      "call"      "data"

```

Lampiran 7. Command R Algoritma K-Medoids Clustering dengan $K = 3$

```

> k3<- pam(dtw_distance,3)
> k3
Medoids:
              ID      ACEH SUMATERA UTARA SUMATERA BARAT
BANTEN         11  9.228286      7.542009      6.963939 ...
PAPUA           33 11.546568     13.562435     15.707730 ...
NUSA TENGGARA BARAT 18 11.589093     10.275939     10.346528 ...
Clustering vector:
              ACEH              SUMATERA UTARA
              1              1
SUMATERA BARAT              RIAU
              1              1
KEPULAUAN RIAU              JAMBI
              2              1
BENGKULU              SUMATERA SELATAN
              2              1
KEPULAUAN BANGKA BELITUNG LAMPUNG
              2              1
BANTEN              JAWA BARAT
              1              1
DKI JAKARTA              JAWA TENGAH
              1              1
DI YOGYAKARTA              JAWA TIMUR
              1              1
BALI              NUSA TENGGARA BARAT
              1              3
NUSA TENGGARA TIMUR              KALIMANTAN BARAT
              2              3
KALIMANTAN SELATAN              KALIMANTAN TENGAH
              1              3
KALIMANTAN TIMUR              KALIMANTAN UTARA
              2              3
GORONTALO              SULAWESI SELATAN
              3              3
SULAWESI TENGGARA              SULAWESI TENGAH
              2              3
SULAWESI UTARA              SULAWESI BARAT
              3              3
MALUKU              MALUKU UTARA
              2              2
PAPUA              PAPUA BARAT
              2              2

Objective function:
  build      swap
16.27547 16.27547

Available components:
[1] "medoids"      "id.med"      "clustering"  "objective"
"isolation"
[6] "clusinfo"    "silinfo"    "diss"        "call"        "data"

```

Lampiran 8. Command R Algoritma K-Medoids Clustering dengan $K = 4$

```

> k4<- pam(dtw_distance,4)
> k4
Medoids:
              ID      ACEH SUMATERA UTARA SUMATERA BARAT
BANTEN          11  9.228286      7.542009      6.963939 ...
KALIMANTAN TIMUR 23 14.159537     15.097925     13.240935 ...
KALIMANTAN UTARA 24 11.694363     10.392958     12.884965 ...
SULAWESI TENGGARA 27 20.269139     20.911479     20.612439 ...
Clustering vector:
              ACEH              SUMATERA UTARA
              1              1
SUMATERA BARAT              RIAU
              1              1
KEPULAUAN RIAU              JAMBI
              2              1
BENGKULU              SUMATERA SELATAN
              3              1
KEPULAUAN BANGKA BELITUNG      LAMPUNG
              2              1
BANTEN              JAWA BARAT
              1              1
DKI JAKARTA              JAWA TENGAH
              1              1
DI YOGYAKARTA              JAWA TIMUR
              1              1
BALI              NUSA TENGGARA BARAT
              1              3
NUSA TENGGARA TIMUR              KALIMANTAN BARAT
              3              3
KALIMANTAN SELATAN              KALIMANTAN TENGAH
              1              2
KALIMANTAN TIMUR              KALIMANTAN UTARA
              2              3
GORONTALO              SULAWESI SELATAN
              1              3
SULAWESI TENGGARA              SULAWESI TENGAH
              4              2
SULAWESI UTARA              SULAWESI BARAT
              3              2
MALUKU              MALUKU UTARA
              2              2
PAPUA              PAPUA BARAT
              2              2

Objective function:
      build      swap
14.90893 14.86619

Available components:
[1] "medoids"      "id.med"      "clustering"  "objective"
"isolation"
[6] "clusinfo"     "silinfo"     "diss"        "call"        "data"

```

Lampiran 9. Command R Algoritma K-Medoids Clustering dengan $K = 5$

```

> k5<- pam(dtw_distance,5)
> k5
Medoids:
              ID      ACEH SUMATERA UTARA SUMATERA BARAT
BANTEN         11  9.228286      7.542009      6.963939 ...
KEPULAUAN RIAU   5 16.571216      15.168150      14.718072 ...
NUSA TENGGARA BARAT 18 11.589093      10.275939      10.346528 ...
PAPUA           33 11.546568      13.562435      15.707730 ...
SULAWESI TENGGARA 27 20.269139      20.911479      20.612439 ...
Clustering vector:
              ACEH              SUMATERA UTARA
              1              1
SUMATERA BARAT              RIAU
              1              1
KEPULAUAN RIAU              JAMBI
              2              1
              BENGKULU      SUMATERA SELATAN
              2              1
KEPULAUAN BANGKA BELITUNG      LAMPUNG
              2              1
              BANTEN              JAWA BARAT
              1              1
              DKI JAKARTA      JAWA TENGAH
              1              1
              DI YOGYAKARTA      JAWA TIMUR
              1              1
              BALI      NUSA TENGGARA BARAT
              1              3
NUSA TENGGARA TIMUR      KALIMANTAN BARAT
              4              3
KALIMANTAN SELATAN      KALIMANTAN TENGAH
              1              3
KALIMANTAN TIMUR      KALIMANTAN UTARA
              4              3
              GORONTALO      SULAWESI SELATAN
              3              3
SULAWESI TENGGARA      SULAWESI TENGAH
              5              3
SULAWESI UTARA      SULAWESI BARAT
              3              3
              MALUKU      MALUKU UTARA
              4              4
              PAPUA      PAPUA BARAT
              4              4

Objective function:
  build      swap
13.82015 13.82015

Available components:
[1] "medoids"      "id.med"      "clustering" "objective"
"isolation"
[6] "clusinfo"    "silinfo"    "diss"      "call"      "data"

```

Lampiran 10. Command R Algoritma K-Medoids Clustering dengan $K = 6$

```

> k6<- pam(dtw_distance,6)
> k6
Medoids:
              ID      ACEH SUMATERA UTARA SUMATERA BARAT
SUMATERA UTARA    2  6.587717      0.000000      10.424612 ...
JAWA BARAT       12 12.087852      8.242304      9.343021 ...
KEPULAUAN RIAU    5 16.571216     15.168150     14.718072 ...
NUSA TENGGARA BARAT 18 11.589093     10.275939     10.346528 ...
PAPUA            33 11.546568     13.562435     15.707730 ...
SULAWESI TENGGARA 27 20.269139     20.911479     20.612439 ...
Clustering vector:
              ACEH      SUMATERA UTARA
              1          1
SUMATERA BARAT          RIAU
              1          2
KEPULAUAN RIAU      JAMBI
              3          2
BENGKULU      SUMATERA SELATAN
              3          2
KEPULAUAN BANGKA BELITUNG      LAMPUNG
              3          1
BANTEN      JAWA BARAT
              2          2
DKI JAKARTA      JAWA TENGAH
              1          2
DI YOGYAKARTA      JAWA TIMUR
              1          2
BALI      NUSA TENGGARA BARAT
              1          4
NUSA TENGGARA TIMUR      KALIMANTAN BARAT
              5          4
KALIMANTAN SELATAN      KALIMANTAN TENGAH
              1          4
KALIMANTAN TIMUR      KALIMANTAN UTARA
              5          4
GORONTALO      SULAWESI SELATAN
              4          4
SULAWESI TENGGARA      SULAWESI TENGAH
              6          4
SULAWESI UTARA      SULAWESI BARAT
              4          4
MALUKU      MALUKU UTARA
              5          5
PAPUA      PAPUA BARAT
              5          5

Objective function:
  build      swap
12.86363 12.51027

Available components:
[1] "medoids"      "id.med"      "clustering" "objective"
"isolation"
[6] "clusinfo"      "silinfo"      "diss"      "call"      "data"

```


Lampiran 11. Command R Algoritma K-Medoids Clustering dengan $K = 7$

```

> k7<- pam(dtw_distance,7)
> k7
Medoids:
              ID      ACEH SUMATERA UTARA SUMATERA BARAT
SUMATERA UTARA    2  6.587717      0.000000      10.424612 ...
JAWA BARAT       12 12.087852      8.242304      9.343021 ...
KEPULAUAN RIAU    5 16.571216     15.168150     14.718072 ...
NUSA TENGGARA BARAT 18 11.589093     10.275939     10.346528 ...
MALUKU           31 12.666697     12.118244     12.162433 ...
SULAWESI TENGGARA 27 20.269139     20.911479     20.612439 ...
MALUKU UTARA     32 14.383423     14.521590     18.302158 ...
Clustering vector:
              ACEH              SUMATERA UTARA
              1              1
SUMATERA BARAT              RIAU
              1              2
KEPULAUAN RIAU              JAMBI
              3              2
              BENGKULU      SUMATERA SELATAN
              3              2
KEPULAUAN BANGKA BELITUNG      LAMPUNG
              3              1
              BANTEN          JAWA BARAT
              2              2
              DKI JAKARTA      JAWA TENGAH
              1              2
              DI YOGYAKARTA      JAWA TIMUR
              1              2
              BALI          NUSA TENGGARA BARAT
              1              4
NUSA TENGGARA TIMUR          KALIMANTAN BARAT
              5              4
KALIMANTAN SELATAN          KALIMANTAN TENGAH
              1              5
KALIMANTAN TIMUR          KALIMANTAN UTARA
              5              4
              GORONTALO      SULAWESI SELATAN
              4              4
SULAWESI TENGGARA          SULAWESI TENGAH
              6              5
SULAWESI UTARA          SULAWESI BARAT
              4              5
              MALUKU          MALUKU UTARA
              5              7
              PAPUA          PAPUA BARAT
              5              7

Objective function:
      build      swap
11.96740 11.53438

Available components:
[1] "medoids"      "id.med"      "clustering"  "objective"
"isolation"
[6] "clusinfo"    "silinfo"    "diss"        "call"        "data"

```

Lampiran 12. Command R Algoritma K-Medoids Clustering dengan $K = 8$

```

> k8<- pam(dtw_distance,8)
> k8
Medoids:
      ID      ACEH SUMATERA UTARA SUMATERA BARAT
SUMATERA UTARA      2  6.587717      0.000000      10.424612 ...
JAWA BARAT          12 12.087852      8.242304      9.343021 ...
KEPULAUAN RIAU      5 16.571216      15.168150      14.718072 ...
BENGKULU            7 14.631814      13.425053      14.139021 ...
NUSA TENGGARA BARAT 18 11.589093      10.275939      10.346528 ...
MALUKU              31 12.666697      12.118244      12.162433 ...
SULAWESI TENGGARA   27 20.269139      20.911479      20.612439 ...
PAPUA BARAT         34 13.263443      15.459114      15.880363 ...
Clustering vector:
      ACEH      SUMATERA UTARA
      1      1
SUMATERA BARAT      RIAU
      1      2
KEPULAUAN RIAU      JAMBI
      3      2
      BENGKULU      SUMATERA SELATAN
      4      2
KEPULAUAN BANGKA BELITUNG      LAMPUNG
      3      1
      BANTEN      JAWA BARAT
      2      2
      DKI JAKARTA      JAWA TENGAH
      1      2
      DI YOGYAKARTA      JAWA TIMUR
      1      2
      BALI      NUSA TENGGARA BARAT
      1      5
NUSA TENGGARA TIMUR      KALIMANTAN BARAT
      6      5
      KALIMANTAN SELATAN      KALIMANTAN TENGAH
      1      6
      KALIMANTAN TIMUR      KALIMANTAN UTARA
      6      5
      GORONTALO      SULAWESI SELATAN
      5      5
      SULAWESI TENGGARA      SULAWESI TENGAH
      7      6
      SULAWESI UTARA      SULAWESI BARAT
      5      6
      MALUKU      MALUKU UTARA
      6      8
      PAPUA      PAPUA BARAT
      6      8
Objective function:
      build      swap
11.11977 10.87139

```

Lampiran 13. Command R Algoritma K-Medoids Clustering dengan $K = 9$

```

> k9<- pam(dtw_distance,9)
> k9
Medoids:
              ID      ACEH SUMATERA UTARA SUMATERA BARAT
SUMATERA UTARA    2  6.587717      0.000000      10.424612 ...
JAWA BARAT       12 12.087852      8.242304      9.343021 ...
KEPULAUAN RIAU   5  16.571216     15.168150     14.718072 ...
BENGKULU         7  14.631814     13.425053     14.139021 ...
NUSA TENGGARA BARAT 18 11.589093     10.275939     10.346528 ...
MALUKU           31 12.666697     12.118244     12.162433 ...
SULAWESI TENGGARA 27 20.269139     20.911479     20.612439 ...
MALUKU UTARA     32 14.383423     14.521590     18.302158 ...
PAPUA BARAT      34 13.263443     15.459114     15.880363 ...
Clustering vector:
              ACEH              SUMATERA UTARA
              1              1
SUMATERA BARAT              RIAU
              1              2
KEPULAUAN RIAU              JAMBI
              3              2
              BENGKULU      SUMATERA SELATAN
              4              2
KEPULAUAN BANGKA BELITUNG      LAMPUNG
              3              1
              BANTEN          JAWA BARAT
              2              2
              DKI JAKARTA      JAWA TENGAH
              1              2
              DI YOGYAKARTA      JAWA TIMUR
              1              2
              BALI              NUSA TENGGARA BARAT
              1              5
NUSA TENGGARA TIMUR      KALIMANTAN BARAT
              6              5
KALIMANTAN SELATAN      KALIMANTAN TENGAH
              1              6
KALIMANTAN TIMUR      KALIMANTAN UTARA
              6              5
              GORONTALO      SULAWESI SELATAN
              5              5
SULAWESI TENGGARA      SULAWESI TENGAH
              7              6
SULAWESI UTARA      SULAWESI BARAT
              5              6
MALUKU              MALUKU UTARA
              6              8
PAPUA              PAPUA BARAT
              6              9

Objective function:
  build      swap
10.45078 10.29163

```

Lampiran 14. Command R Algoritma K-Medoids Clustering dengan $K = 10$

```

> k10<- pam(dtw_distance,10)
> k10
Medoids:
      ID      ACEH SUMATERA UTARA SUMATERA BARAT
SUMATERA UTARA      2  6.587717      0.000000      10.424612 ...
JAWA BARAT          12 12.087852      8.242304      9.343021 ...
KEPULAUAN RIAU       5 16.571216     15.168150     14.718072 ...
BENGKULU             7 14.631814     13.425053     14.139021 ...
KALIMANTAN SELATAN  21  6.854467      5.367432      7.547835 ...
NUSA TENGGARA BARAT 18 11.589093     10.275939     10.346528 ...
MALUKU              31 12.666697     12.118244     12.162433 ...
SULAWESI TENGGARA   27 20.269139     20.911479     20.612439 ...
MALUKU UTARA        32 14.383423     14.521590     18.302158 ...
PAPUA BARAT         34 13.263443     15.459114     15.880363 ...
Clustering vector:
      ACEH      SUMATERA UTARA
      1      1
SUMATERA BARAT      RIAU
      1      2
KEPULAUAN RIAU      JAMBI
      3      2
      BENGKULU      SUMATERA SELATAN
      4      2
KEPULAUAN BANGKA BELITUNG      LAMPUNG
      3      1
      BANTEN      JAWA BARAT
      5      2
      DKI JAKARTA      JAWA TENGAH
      5      2
      DI YOGYAKARTA      JAWA TIMUR
      1      2
      BALI      NUSA TENGGARA BARAT
      5      6
NUSA TENGGARA TIMUR      KALIMANTAN BARAT
      7      6
      KALIMANTAN SELATAN      KALIMANTAN TENGAH
      5      7
      KALIMANTAN TIMUR      KALIMANTAN UTARA
      7      6
      GORONTALO      SULAWESI SELATAN
      6      6
      SULAWESI TENGGARA      SULAWESI TENGAH
      8      7
      SULAWESI UTARA      SULAWESI BARAT
      6      7
      MALUKU      MALUKU UTARA
      7      9
      PAPUA      PAPUA BARAT
      7      10
Objective function:
      build      swap
9.787786 9.729462

```

Lampiran 15. Command R untuk Koefisien Silhouette

```

> #Validitas Klaster Menggunakan Koefisien Silhouette untuk DTW
> sil.dtw<-NULL
> for (k in 2:10){
+   clustering.dtw <-pam(dtw_distance,k)
+   koef.silhdtw <-
mean(silhouette(clustering.dtw$cluster,dmatrix=dtw_distance)[,3])
+   sil.dtw <- rbind(sil.dtw, c(k, koef.silhdtw))
+ }
> sil.dtw
      [,1]      [,2]
[1,]    2 0.19101628
[2,]    3 0.07681848
[3,]    4 0.12519397
[4,]    5 0.10894392
[5,]    6 0.07588228
[6,]    7 0.08798110
[7,]    8 0.10079792
[8,]    9 0.09920083
[9,]   10 0.04710534
> plot(sil.dtw[,1], sil.dtw[,2],
type="b",ylim=c(0,0.25),xlim=c(2,10), xlab="Banyaknya Cluster",
ylab="Koefisien Silhouette Jarak DTW", main="Plot Koefisien
Silhouette")
> text(2.1,0.21,"0.19")
> text(3,0.10,"0.08")
> text(4,0.135,"0.12")
> text(5,0.12,"0.10")
> text(6,0.09,"0.08")
> text(7,0.10,"0.09")
> text(8,0.11,"0.10")
> text(9,0.11,"0.10")
> text(10,0.06,"0.05")
> abline(v=1.7, col="red", lwd=1, lty=2)
> abline(v=2.5, col="red", lwd=1, lty=2)

```

Lampiran 16. *Command R* untuk Hasil Klaster

```
> library(units)
> library(tmap)
> library(raster)
> library(sf)
> library(maptools)
> library(ggplot2)

> #Input Shp File
> spIndo = shapefile(file.choose())
> names(spIndo)
> spIndo$PROVINSI

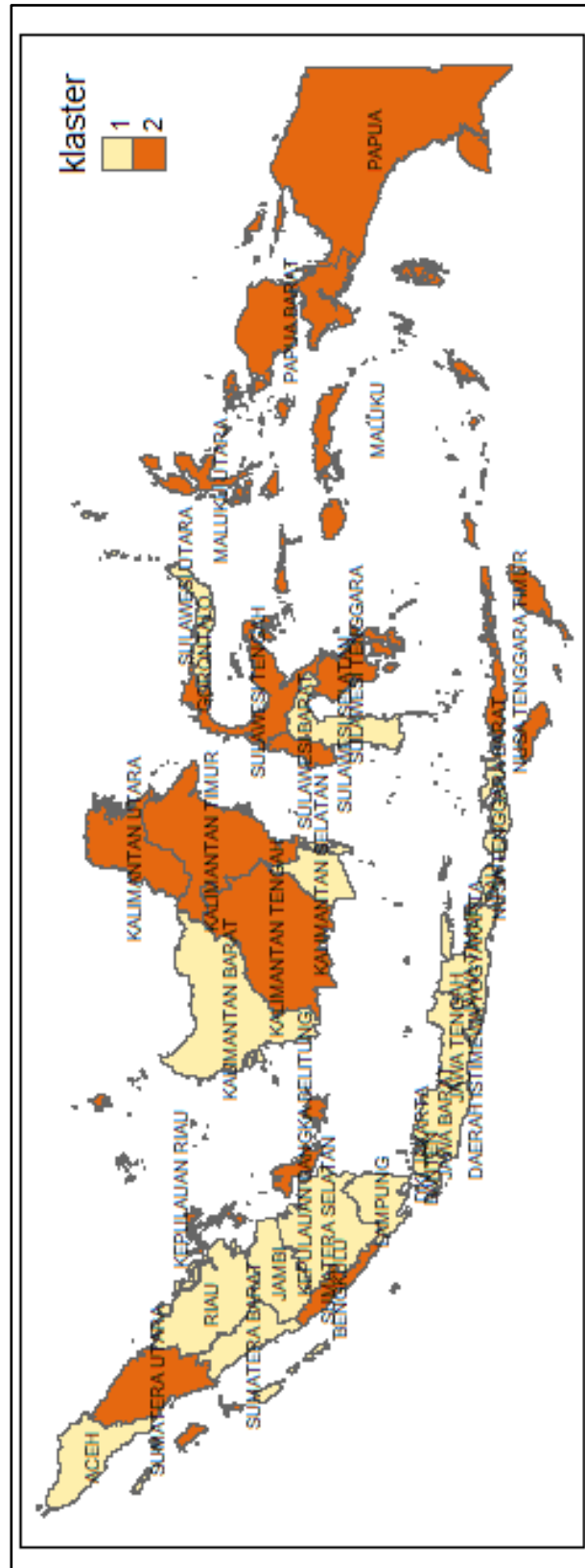
> #Menampilkan Peta Indonesia
> tm_shape(spIndo) + tm_polygons()

> #Panggil Data
> data<-read.csv(file.choose(),header=TRUE)

> #Menampilkan Layout Peta Tiap Variabel
> projection(spIndo) <- CRS("+init=epsg:3395")

> Profilisasi Klaster
> spIndo$klaster=data$profil #Profil
> tm_shape(spIndo) + tm_polygons(col="klaster") +
  tm_layout(main.title = "Profilisasi Klaster", title.size =
    1.5,main.title.position="center")+tm_text(text = "PROVINSI",
    size = 0.5, col = "black")
```

Lampiran 17. Visualisasi *Spatial Mapping* untuk Hasil Klaster



Lampiran 18. Data Objek *Prototype* Setiap Klaster

t	$Y(1)_t$	$Y(2)_t$	t	$Y(1)_t$	$Y(2)_t$	t	$Y(1)_t$	$Y(2)_t$
1	13.476	14.600	22	12.453	13.987	43	14.576	14.930
2	13.455	14.513	23	12.392	13.990	44	14.742	15.253
3	13.418	14.530	24	12.421	14.030	45	15.079	15.297
4	13.345	14.417	25	12.379	13.977	46	15.032	15.663
5	13.232	14.450	26	12.382	13.943	47	15.466	15.770
6	13.229	14.373	27	12.671	13.970	48	15.889	15.913
7	13.203	14.347	28	13.058	14.060	49	16.108	16.153
8	13.205	14.373	29	13.492	14.237	50	17.603	17.113
9	13.147	14.363	30	13.284	14.957	51	19.003	18.697
10	13.111	14.370	31	13.213	14.497	52	19.853	20.157
11	12.963	14.363	32	13.150	14.523	53	19.555*	19.567*
12	12.747	14.327	33	13.032	14.450	54	18.155*	18.710*
13	12.716	14.270	34	12.979	14.363	55	22.589*	25.670*
14	12.650	14.237	35	13.263	14.240	56	23.063*	26.213*
15	12.516	14.150	36	13.466	14.360	57	22.271*	25.853*
16	12.442	14.090	37	13.858	14.497	58	21.642*	24.893*
17	12.442	14.090	38	14.029	14.547	59	19.582*	22.580*
18	12.461	14.100	39	14.176	14.583	60	18.897*	21.340*
19	12.455	14.050	40	14.250	14.687	61	18.334*	20.747*
20	12.426	14.047	41	14.282	14.740			
21	12.439	14.027	42	14.318	14.833			

Keterangan: (*) Data *out-sample*

Lampiran 19. Command R untuk Visualisasi Objek *Prototype* Klaster 1

```

> #CLUSTER 1
> #Input Data
> datacl=read.csv(file.choose(),header=TRUE)
> #Multiple Plot
> xrange <- range(1:61)
> yrange <- range(c(datacl$GORONTALO))
> plot(xrange, yrange, type="n", xlab="time",
+      ylab="value",ylim=c(12000,32000),main="Multiple Plot Time
      Series Cluster 1")
> lines(datacl$ACEH, col='chocolate2', type='l',lwd=3)
> lines(datacl$SUMATERA.UTARA, col='chocolate2', type='l',lwd=3)
> lines(datacl$SUMATERA.BARAT, col='chocolate2', type='l',lwd=3)
> lines(datacl$RIAU, col='chocolate2', type='l',lwd=3)
> lines(datacl$JAMBI, col='chocolate2', type='l',lwd=3)
> lines(datacl$SUMATERA.SELATAN, col='chocolate2',
      type='l',lwd=3)
> lines(datacl$LAMPUNG, col='chocolate2', type='l',lwd=3)
> lines(datacl$BANTEN, col='chocolate2', type='l',lwd=3)
> lines(datacl$JAWA.BARAT, col='chocolate2', type='l',lwd=3)
> lines(datacl$DKI.JAKARTA, col='chocolate2', type='l',lwd=3)
> lines(datacl$JAWA.TENGAH, col='chocolate2', type='l',lwd=3)
> lines(datacl$DI.YOGYAKARTA, col='chocolate2', type='l',lwd=3)
> lines(datacl$JAWA.TIMUR, col='chocolate2', type='l',lwd=3)
> lines(datacl$BALI, col='chocolate2', type='l',lwd=3)
> lines(datacl$NUSA.TENGGARA.BARAT, col='chocolate2',
      type='l',lwd=3)
> lines(datacl$KALIMANTAN.BARAT, col='chocolate2',
      type='l',lwd=3)
> lines(datacl$KALIMANTAN.SELATAN, col='chocolate2',
      type='l',lwd=3)
> lines(datacl$GORONTALO, col='chocolate2', type='l',lwd=3)
> lines(datacl$SULAWESI.SELATAN, col='chocolate2',
      type='l',lwd=3)
> lines(datacl$Average, col='blueviolet', type='o',lwd=3)
> legend("topleft",legend=c("Objek Klaster 1","Objek
      Prototype"),cex=1,lty=1,col=c("chocolate2","blueviolet"),pch=10)

```

Lampiran 20. Command R untuk Visualisasi Objek *Prototype* Klaster 2

```

> #CLUSTER 2
> #Input Data
> datac2=read.csv(file.choose(),header=TRUE)
> #Multiple Plot
> xrange <- range(1:61)
> yrange <- range(c(datac2$NUSA.TENGGARA.TIMUR))
> plot(xrange, yrange, type="n", xlab="time",
+      ylab="value",ylim=c(12000,45000),main="Multiple Plot Time
+      Series Cluster 2")
> lines(datac2$KEPULAUAN.RIAU, col='magenta', type='l',lwd=3)
> lines(datac2$BENGKULU, col='magenta', type='l',lwd=3)
> lines(datac2$KEPULAUAN.BANGKA.BELITUNG, col='magenta',
+      type='l',lwd=3)
> lines(datac2$NUSA.TENGGARA.TIMUR, col='magenta',
+      type='l',lwd=3)
> lines(datac2$KALIMANTAN.TENGAH, col='magenta', type='l',lwd=3)
> lines(datac2$KALIMANTAN.TIMUR, col='magenta', type='l',lwd=3)
> lines(datac2$KALIMANTAN.UTARA, col='magenta', type='l',lwd=3)
> lines(datac2$SULAWESI.TENGGARA, col='magenta', type='l',lwd=3)
> lines(datac2$SULAWESI.TENGAH, col='magenta', type='l',lwd=3)
> lines(datac2$SULAWESI.UTARA, col='magenta', type='l',lwd=3)
> lines(datac2$SULAWESI.BARAT, col='magenta', type='l',lwd=3)
> lines(datac2$MALUKU, col='magenta', type='l',lwd=3)
> lines(datac2$MALUKU.UTARA, col='magenta', type='l',lwd=3)
> lines(datac2$PAPUA, col='magenta', type='l',lwd=3)
> lines(datac2$PAPUA.BARAT, col='magenta', type='l',lwd=3)
> lines(datac2$Average, col='black', type='o',lwd=5)
> legend("topleft",legend=c("Objek Klaster 2","Objek
+      Prototype"),cex=1,lty=1,col=c("magenta","black"),pch=10)

```

Lampiran 21. Command R untuk Identifikasi Awal Model ARIMA Klaster 1

```

> library(MASS)
> library(car)
> library(forecast)
> library(tseries)
> library(quadprog)
> library(zoo)
> library(fracdiff)
> library(fUnitRoots)
> library(lmtest)
> library(writexl)
> library(BETS)
> library(AID)
> library(urca)

> #CLUSTER 1
> #Input Data
> datacl=read.csv(file.choose(),header=TRUE)

> #ARIMA
> #cek stasioner dalam variansi
> boxcox(datacl$Average~1)
> p<-powerTransform(datacl$Average)
> p
Estimated transformation parameter
datacl$Average
      -2.991946
> y<-(datacl$Average^(-2.991946))
> out <- boxcoxnc(y,method="mle",verbose=F,plot=F)
> out$lambda.hat
[1] 1.32

> #cek stasioner dalam rata-rata
> test=ur.df(y)
> summary(test)
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
Test regression none
Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
Errors:
      Min          1Q      Median          3Q          Max
-5.271e-14 -4.374e-15  1.652e-15  7.197e-15  4.946e-14
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
z.lag.1      -0.005102   0.004935  -1.034    0.306
z.diff.lag    0.582187   0.117348   4.961  9.2e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error standard error: 1.587e-14 on 48 degrees of freedom
Multiple R-squared:  0.3702, Adjusted R-squared:  0.3439
F-statistic: 14.11 on 2 and 48 DF,  p-value: 1.517e-05

```

Lampiran 21. Command R untuk Identifikasi Awal Model ARIMA Klaster 1 (lanjutan)

```

Value of test-statistic is: -1.034

Critical values for test statistics:
      1pct  5pct 10pct
taul -2.6 -1.95 -1.61

> datadiff=diff(y,differences=1)
> plot.ts(datadiff)

> #setelah diff
> test2=ur.df(datadiff)
> summary(test2)
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
Test regression none

Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Errors:
      Min          1Q      Median          3Q         Max
-5.297e-14 -6.774e-15 -1.053e-15  5.145e-15  4.916e-14

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
z.lag.1      -0.37541    0.13573  -2.766  0.00809 **
z.diff.lag  -0.05062    0.15265  -0.332  0.74166
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error standard error: 1.619e-14 on 47 degrees of freedom
Multiple R-squared:  0.196, Adjusted R-squared:  0.1618
F-statistic: 5.729 on 2 and 47 DF, p-value: 0.005935

Value of test-statistic is: -2.7659

Critical values for test statistics:
      1pct  5pct 10pct
taul -2.6 -1.95 -1.61

```

Lampiran 22. Command R untuk Identifikasi Orde dan Penaksiran Parameter Model ARIMA Klaster 1

```

> acf(datadiff,56,main="FOK Y(1)t Setelah
Differencing",ylab="FOK")
> pacf(datadiff,56,main="FOKP Y(1)t Setelah
Differencing",ylab="FOKP")
> #Penaksiran dan Pengujian Signifikansi Parameter
> fit1=arima(x=y,order=c(0,1,1))
> coeftest(fit1)

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ma1  0.46757    0.10039  4.6575 3.2e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
AIC = -3087.35

> fit2=arima(x=y,order=c(0,1,2))
> coeftest(fit2)

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ma1  0.53803    0.13137  4.0956 4.211e-05 ***
ma2  0.28318    0.13113  2.1596  0.03081 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
AIC = -3089.91

> fit4=arima(x=y,order=c(1,1,0))
> coeftest(fit4)

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.59107    0.11119  5.3158 1.062e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
AIC = -3093.77

```

Lampiran 23. Command R untuk Diagnostic Checking Model ARIMA Klaster 1

```
#Pemeriksaan diagnostik
#Independensi Error
> fit1=arima(x=y,order=c(0,1,1))
> tsdiag(fit1,53) #Mengeluarkan plot Ljung Box
> fit2=arima(x=y,order=c(0,1,2))
> tsdiag(fit2,52) #Mengeluarkan plot Ljung Box
> fit3=arima(x=y,order=c(0,1,3))
> tsdiag(fit3,53) #Mengeluarkan plot Ljung Box
> fit4=arima(x=y,order=c(1,1,0))
> tsdiag(fit4,53) #Mengeluarkan plot Ljung Box

#Normalitas error
#Mengeluarkan nilai error
> res1=resid(fit1)
> res2=resid(fit2)
> res3=resid(fit3)
> res4=resid(fit4)
#Kolmogorov Smirnov Test
> ks.test(res1,"pnorm",mean(res1),sqrt(var(res1)))

Exact one-sample Kolmogorov-Smirnov test

data:  res1
D = 0.15531, p-value = 0.1462
alternative hypothesis: two-sided

> ks.test(res2,"pnorm",mean(res2),sqrt(var(res2)))

Exact one-sample Kolmogorov-Smirnov test

data:  res2
D = 0.1769, p-value = 0.06794
alternative hypothesis: two-sided

> ks.test(res3,"pnorm",mean(res3),sqrt(var(res3)))

Exact one-sample Kolmogorov-Smirnov test

data:  res3
D = 0.14698, p-value = 0.1913
alternative hypothesis: two-sided

> ks.test(res4,"pnorm",mean(res4),sqrt(var(res4)))

Exact one-sample Kolmogorov-Smirnov test

data:  res4
D = 0.17033, p-value = 0.0867
alternative hypothesis: two-sided
```

Lampiran 24. *P-value* Pengujian Ljung-Box untuk Semua Model Klaster 1

ARIMA (0,1,1)									
<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>
1	0,427	12	0,160	23	0,298	34	0,422	45	0,144
2	0,067	13	0,206	24	0,319	35	0,369	46	0,130
3	0,141	14	0,258	25	0,310	36	0,257	47	0,128
4	0,240	15	0,314	26	0,345	37	0,292	48	0,132
5	0,350	16	0,344	27	0,362	38	0,185	49	0,140
6	0,415	17	0,221	28	0,385	39	0,157	50	0,161
7	0,464	18	0,273	29	0,437	40	0,139	51	0,186
8	0,104	19	0,321	30	0,450	41	0,138		
9	0,148	20	0,277	31	0,471	42	0,149		
10	0,088	21	0,247	32	0,454	43	0,143		
11	0,116	22	0,296	33	0,422	44	0,162		
ARIMA (0,1,2)									
<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>
1	0,922	12	0,652	23	0,327	34	0,588	45	0,504
2	0,935	13	0,708	24	0,346	35	0,538	46	0,489
3	0,774	14	0,773	25	0,308	36	0,448	47	0,515
4	0,872	15	0,796	26	0,359	37	0,482	48	0,545
5	0,936	16	0,790	27	0,388	38	0,416	49	0,576
6	0,972	17	0,382	28	0,419	39	0,370	50	0,615
7	0,953	18	0,434	29	0,469	40	0,390	51	0,653
8	0,544	19	0,442	30	0,501	41	0,423		
9	0,635	20	0,297	31	0,536	42	0,454		
10	0,542	21	0,284	32	0,545	43	0,463		
11	0,587	22	0,323	33	0,541	44	0,506		
ARIMA (1,1,0)									
<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>
1	0,728	12	0,739	23	0,397	34	0,754	45	0,827
2	0,930	13	0,804	24	0,410	35	0,752	46	0,827
3	0,985	14	0,857	25	0,394	36	0,727	47	0,849
4	0,974	15	0,872	26	0,447	37	0,732	48	0,870
5	0,962	16	0,862	27	0,495	38	0,688	49	0,888
6	0,985	17	0,495	28	0,539	39	0,673	50	0,907
7	0,986	18	0,548	29	0,578	40	0,704	51	0,923
8	0,613	19	0,564	30	0,620	41	0,738		
9	0,685	20	0,388	31	0,661	42	0,771		
10	0,626	21	0,360	32	0,687	43	0,791		
11	0,702	22	0,399	33	0,712	44	0,820		

Lampiran 25. Command R untuk Identifikasi Awal Model ARIMA Klaster 2

```

> library(MASS)
> library(car)
> library(forecast)
> library(tseries)
> library(quadprog)
> library(zoo)
> library(fracdiff)
> library(fUnitRoots)
> library(lmtest)
> library(writexl)
> library(BETS)
> library(AID)
> library(urca)

> #CLUSTER 2
> #Input Data
> datac2=read.csv(file.choose(),header=TRUE)
> #ARIMA
> #cek stasioner dalam variansi
> library(AID)
> out <- boxcoxnc(datac2$Average,method="jb",
  verbose=F,plot=F,lambda=seq(-5,5,0.01))
> out$lambda.hat
[1] -3.64
> y<-(datac2$Average^(-3.64))
> out <- boxcoxnc(y,method="jb",
  verbose=F,plot=F,lambda=seq(-25,25,0.01))
> out$lambda.hat
[1] 1.01

> #cek stasioner dalam rata-rata
> test=ur.df(y)
> summary(test)
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
Test regression none

Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
Errors:
      Min          1Q      Median          3Q         Max
-1.075e-16 -9.800e-18  3.316e-18  1.255e-17  1.226e-16
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
z.lag.1      -0.007912   0.006481  -1.221   0.2281
z.diff.lag    0.326943   0.140549   2.326   0.0243 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error standard error: 3.231e-17 on 48 degrees of freedom
Multiple R-squared:  0.1417, Adjusted R-squared:  0.1059
F-statistic: 3.961 on 2 and 48 DF, p-value: 0.02557

```


Lampiran 25. Command R untuk Identifikasi Awal Model ARIMA Klaster 2 (lanjutan)

```

Value of test-statistic is: -1.2209

Critical values for test statistics:
      1pct   5pct 10pct
taul -2.6 -1.95 -1.61

> datadiff=diff(y,differences=1)
> plot.ts(datadiff)
> #setelah diff
> test2=ur.df(datadiff)
> summary(test2)
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
Test regression none

Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
Errors:
      Min          1Q      Median          3Q         Max
-1.089e-16 -1.095e-17 -2.480e-19  6.831e-18  1.236e-16
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
z.lag.1      -0.3903      0.1847  -2.114   0.0399 *
z.diff.lag   -0.3263      0.1588  -2.055   0.0455 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error standard error: 3.174e-17 on 47 degrees of freedom
Multiple R-squared:  0.3602, Adjusted R-squared:  0.3329
F-statistic: 13.23 on 2 and 47 DF, p-value: 2.771e-05

Value of test-statistic is: -2.1139

Critical values for test statistics:
      1pct   5pct 10pct
taul -2.6 -1.95 -1.61

```

Lampiran 26. Command R untuk Identifikasi Orde dan Penaksiran Parameter Model ARIMA Klaster 2

```
> acf(datadiff,56,main="FOK Y(2)t Setelah
Differencing",ylab="FOK")
> pacf(datadiff,56,main="FOKP Y(2)t Setelah
Differencing",ylab="FOKP")

> #Penaksiran dan Pengujian Signifikansi Parameter
> fit1=arima(x=y,order=c(0,1,1))
> coeftest(fit1)

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ma1  0.22504    0.10940   2.057  0.03969 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
AIC = -3723.29

> fit2=arima(x=y,order=c(1,1,0))
> coeftest(fit2)

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.34656    0.13560   2.5557  0.0106 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
AIC = -3725.53
```

Lampiran 27. Command R untuk Diagnostic Checking Model ARIMA Klaster 2

```
#Pemeriksaan diagnostik
#Independensi Error
> fit1=arima(x=y,order=c(0,1,1))
> tsdiag(fit1,53) #Mengeluarkan plot Ljung Box
> fit2=arima(x=y,order=c(1,1,0))
> tsdiag(fit2,53) #Mengeluarkan plot Ljung Box

#Normalitas error
#Mengeluarkan nilai error
> res1=resid(fit1)
> res2=resid(fit2)
#Kolmogorov Smirnov Test
> ks.test(res1,"pnorm",mean(res1),sqrt(var(res1)))

      Exact one-sample Kolmogorov-Smirnov test

data:  res1
D = 0.17082, p-value = 0.08517
alternative hypothesis: two-sided

> ks.test(res2,"pnorm",mean(res2),sqrt(var(res2)))

      Exact one-sample Kolmogorov-Smirnov test

data:  res2
D = 0.17894, p-value = 0.06288
alternative hypothesis: two-sided
```

Lampiran 28. *P-value* Pengujian Ljung-Box untuk Semua Model Klaster 2

ARIMA (0,1,1)									
<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>
1	0,885	12	0,644	23	0,740	34	0,888	45	0,790
2	0,128	13	0,708	24	0,768	35	0,872	46	0,758
3	0,247	14	0,623	25	0,807	36	0,856	47	0,732
4	0,292	15	0,626	26	0,764	37	0,843	48	0,663
5	0,412	16	0,658	27	0,806	38	0,823	49	0,610
6	0,337	17	0,658	28	0,768	39	0,831	50	0,595
7	0,355	18	0,719	29	0,823	40	0,808	51	0,629
8	0,401	19	0,728	30	0,841	41	0,823		
9	0,495	20	0,772	31	0,855	42	0,812		
10	0,526	21	0,762	32	0,869	43	0,831		
11	0,608	22	0,687	33	0,884	44	0,798		
ARIMA (1,1,0)									
<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>	<i>lag</i>	<i>p-value</i>
1	0,350	12	0,797	23	0,824	34	0,960	45	0,976
2	0,196	13	0,835	24	0,850	35	0,960	46	0,975
3	0,353	14	0,733	25	0,883	36	0,961	47	0,974
4	0,429	15	0,700	26	0,858	37	0,963	48	0,969
5	0,519	16	0,708	27	0,882	38	0,962	49	0,963
6	0,466	17	0,718	28	0,872	39	0,969	50	0,966
7	0,502	18	0,775	29	0,900	40	0,965	51	0,973
8	0,584	19	0,767	30	0,916	41	0,973		
9	0,681	20	0,813	31	0,929	42	0,971		
10	0,709	21	0,808	32	0,941	43	0,977		
11	0,766	22	0,780	33	0,953	44	0,974		

Lampiran 29. Command R untuk Pembangkitan Nilai Error

```

> #Bangkitan Error untuk ARIMA (0,1,1) C1
> set.seed(2700)
> n=67
> e1=rnorm(n,82.41,263.98)
> e1
[1] -176.952475 142.128715 180.477386 -370.410109 418.384033
[6] 335.069353 -251.939580 296.835524 774.056063 316.264592
[11] 243.599749 455.139636 56.765079 412.808805 -114.849618
[16] 43.418329 148.285341 -108.804386 102.803585 373.863823
[21] 255.454551 -1.936715 85.536166 -230.386986 105.650200
[26] 38.039967 245.693902 -67.766966 -21.868907 368.771287
[31] 635.362583 169.421002 165.792409 -144.736326 222.332922
[36] -30.633348 679.357894 -98.058230 442.608896 -294.547850
[41] 16.924934 18.188849 -322.440563 -130.313892 89.647588
[46] 8.491131 343.293237 -207.719649 710.161237 -232.246096
[51] -14.660411 -223.145232 408.646771 157.560830 339.161384
[56] 76.352799 502.977290 -43.228425 -194.829287 335.969901
[61] 131.917445 -442.600780 -97.214217 -277.457518 200.977758
[66] 133.099292 -18.919996
> #Bangkitan Error untuk ARIMA (0,1,2) C1
> set.seed(2709)
> n=67
> e2=rnorm(n,58.38,250.07)
> e2
[1] 415.7258141 -497.1245943 78.2455124 -7.1302941 -53.9300705
[6] -461.5834346 73.3204490 262.6105273 224.6953221 125.5679245
[11] 749.6105475 -217.9231290 -303.8477923 -26.1075032 -15.1012372
[16] 143.6671885 92.7299765 186.5535350 411.5494471 -168.3909318
[21] -66.7514574 458.7623487 426.9419476 -0.9501569 370.5704195
[26] -205.8955450 201.9720877 -109.1302841 -5.7996344 -240.4638499
[31] 360.1235119 219.4960458 172.7724085 639.8086108 -181.2550400
[36] 571.8579086 491.7617765 132.4021516 142.8405008 -279.5502156
[41] -9.2754534 181.0557764 501.3489189 -81.8842640 -249.6494665
[46] -105.8324888 -99.6133693 -325.3607643 187.3921216 418.1221220
[51] -197.1401539 248.1750068 370.1887251 -1.6326636 93.8127529
[56] 3.7416539 -121.7707012 427.7402480 237.0938874 388.4737964
[61] -432.9900147 135.4363411 24.1452124 217.5018651 273.4163855
[66] 29.9902022 -90.4414041
> #Bangkitan Error untuk ARIMA (0,1,1) C2
> set.seed(2711)
> n=67
> e3=rnorm(n,88.32,305.77)
> e3
[1] 384.683542 285.068662 568.995289 -444.532073 292.222210
[6] 72.965956 378.822509 144.237916 451.211043 228.034621
[11] -185.509776 -36.174242 102.623024 -440.532362 -349.373704
[16] -7.431306 347.059180 33.605989 194.173449 -277.784662
[21] -264.106033 507.084849 42.107436 102.947478 23.651645
[26] 276.038455 -54.091405 -224.859919 -108.437761 -241.397455
[31] -200.840860 20.241137 -117.196247 475.911907 443.554292
[36] 476.473856 -121.516542 -67.062891 348.561500 86.935482
[41] -473.555249 -431.042678 -99.614789 178.581506 41.875656
[46] -173.334425 228.630610 402.439033 149.574439 -28.548824
[51] 61.865884 101.812896 227.239238 14.964360 -151.283772
[56] -188.660201 109.081661 287.701734 176.240449 624.484285
[61] 119.238897 221.559457 -10.231574 685.424424 -482.480381
[66] 373.147434 -337.965765

```

Lampiran 30. Data Aktual dan Prediksi untuk Objek *Prototype* Klaster 1 Model ARIMA (0,1,1)

t	$Y(1)_t$	$\hat{Y}(1)_t$	t	$Y(1)_t$	$\hat{Y}(1)_t$	t	$Y(1)_t$	$\hat{Y}(1)_t$
1	13.476	13.480,51	24	12.421	12.363,30	47	15.466	14.940,04
2	13.455	13.474,02	25	12.379	12.448,35	48	15.889	15.738,95
3	13.418	13.446,42	26	12.382	12.347,10	49	16.108	15.961,15
4	13.345	13.404,79	27	12.671	12.398,45	50	17.603	16.178,54
5	13.232	13.317,44	28	13.058	12.806,97	51	19.003	18.472,18
6	13.229	13.192,84	29	13.492	13.182,35	52	19.853	19.273,19
7	13.203	13.246,04	30	13.284	13.647,17	53	19.555*	19.586,29*
8	13.205	13.183,07	31	13.213	13.126,84	54	18.155*	19.398,31*
9	13.147	13.215,30	32	13.150	13.254,07	55	22.589*	19.325,83*
10	13.111	13.115,54	33	13.032	13.102,44	56	23.063*	19.169,82*
11	12.963	13.108,88	34	12.979	12.999,58	57	22.271*	19.134,70*
12	12.747	12.896,97	35	13.263	12.969,42	58	21.642*	18.903,33*
13	12.716	12.679,22	36	13.466	13.409,74	59	19.582*	18.923,21*
14	12.650	12.733,34	37	13.858	13.492,63	60	18.897*	19.012,83*
15	12.516	12.611,77	38	14.029	14.043,06	61	18.334*	18.858,29*
16	12.442	12.472,20	39	14.176	14.022,44	62		18.797,61[^]
17	12.442	12.427,98	40	14.250	14.250,15	63		19.001,20[^]
18	12.461	12.448,58	41	14.282	14.249,93	64		19.045,92[^]
19	12.455	12.466,82	42	14.318	14.297,09	65		19.173,55[^]
20	12.426	12.449,49	43	14.576	14.327,82	66		19.081,10[^]
21	12.439	12.415,08	44	14.742	14.698,11	67		19.019,88[^]
22	12.453	12.450,25	45	15.079	14.762,70			
23	12.392	12.454,29	46	15.032	15.236,54			

Keterangan: (*) Data *out-sample* ; ([^]) Peramalan

Lampiran 31. Data Aktual dan Prediksi untuk Objek *Prototype* Klaster 1 Model ARIMA (0,1,2)

t	$Y(1)_t$	$\hat{Y}(1)_t$	t	$Y(1)_t$	$\hat{Y}(1)_t$	t	$Y(1)_t$	$\hat{Y}(1)_t$
1	13.476	13.480,51	24	12.421	12.355,37	47	15.466	15.002,38
2	13.455	13.472,93	25	12.379	12.436,48	48	15.889	15.676,36
3	13.418	13.444,18	26	12.382	12.366,91	49	16.108	16.171,71
4	13.345	13.399,16	27	12.671	12.374	50	17.603	16.139,45
5	13.232	13.309,12	28	13.058	12.848,09	51	19.003	18.623,90
6	13.229	13.176,82	29	13.492	13.281,54	52	19.853	19.977,90
7	13.203	13.235,80	30	13.284	13.684,15	53	19.555*	19.813,05*
8	13.205	13.200,22	31	13.213	13.141,93	54	18.155*	19.651,82*
9	13.147	13.198,33	32	13.150	13.147,73	55	22.589*	19.549,04*
10	13.111	13.121,03	33	13.032	13.171,25	56	23.063*	19.499,77*
11	12.963	13.091,40	34	12.979	12.960,07	57	22.271*	19.471,52*
12	12.747	12.893,30	35	13.263	12.951,35	58	21.642*	19.535,01*
13	12.716	12.638,58	36	13.466	13.449,85	59	19.582*	19.342,41*
14	12.650	12.718,06	37	13.858	13.574,79	60	18.897*	19.096,98*
15	12.516	12.635,54	38	14.029	14.025,93	61	18.334*	18.824,70*
16	12.442	12.435,66	39	14.176	14.119,59	62		18.945,41[^]
17	12.442	12.413,11	40	14.250	14.207,64	63		18.994,87[^]
18	12.461	12.459,46	41	14.282	14.289,59	64		18.944,15[^]
19	12.455	12.470,11	42	14.318	14.290,11	65		18.822,12[^]
20	12.426	12.447,33	43	14.576	14.330,92	66		18.616,30[^]
21	12.439	12.410,37	44	14.742	14.723,87	67		18.523,85[^]
22	12.453	12.448,44	45	15.079	14.827,89			
23	12.392	12.463,65	46	15.032	15.227,22			

Keterangan: (*) Data *out-sample* ; (^) Peramalan

Lampiran 32. Data Aktual dan Prediksi untuk Objek *Prototype* Klaster 1 Model ARIMA (1,1,0)

t	$Y(1)_t$	$\hat{Y}(1)_t$	t	$Y(1)_t$	$\hat{Y}(1)_t$	t	$Y(1)_t$	$\hat{Y}(1)_t$
1	13.476	13.480,51	24	12.421	12.356,50	47	15.466	15.004,49
2	13.455	13.471,93	25	12.379	12.438,27	48	15.889	15.747,52
3	13.418	13.442,65	26	12.382	12.354,44	49	16.108	16.162,03
4	13.345	13.396,32	27	12.671	12.383,77	50	17.603	16.243,28
5	13.232	13.302,59	28	13.058	12.855,12	51	19.003	18.808,41
6	13.229	13.166,97	29	13.492	13.310,41	52	19.853	20.080,23
7	13.203	13.227,23	30	13.284	13.777,58	53	19.555*	20.433,97*
8	13.205	13.187,73	31	13.213	13.166,89	54	18.155*	20.811,30*
9	13.147	13.206,18	32	13.150	13.171,74	55	22.589*	21.047,93*
10	13.111	13.113,19	33	13.032	13.113,32	56	23.063*	21.192,96*
11	12.963	13.089,91	34	12.979	12.964,21	57	22.271*	21.280,59*
12	12.747	12.878,59	35	13.263	12.948,07	58	21.642*	21.333,07*
13	12.716	12.625,86	36	13.466	13.443,08	59	19.582*	21.364,33*
14	12.650	12.697,82	37	13.858	13.592,01	60	18.897*	21.382,90*
15	12.516	12.611,63	38	14.029	14.112,48	61	18.334*	21.393,90*
16	12.442	12.439,40	39	14.176	14.134,14	62		^21.400,42
17	12.442	12.399,07	40	14.250	14.265,84	63		^21.404,27
18	12.461	12.442	41	14.282	14.294,47	64		^21.406,55
19	12.455	12.472,29	42	14.318	14.301,05	65		^21.407,90
20	12.426	12.451,46	43	14.576	14.339,45	66		^21.408,70
21	12.439	12.408,99	44	14.742	14.737,56	67		^21.409,17
22	12.453	12.446,71	45	15.079	14.843,75			
23	12.392	12.461,30	46	15.032	15.293,36			

Keterangan: (*) Data *out-sample* ; (^) Peramalan

Lampiran 33. Data Aktual dan Prediksi untuk Objek *Prototype* Klaster 2 Model ARIMA (0,1,1)

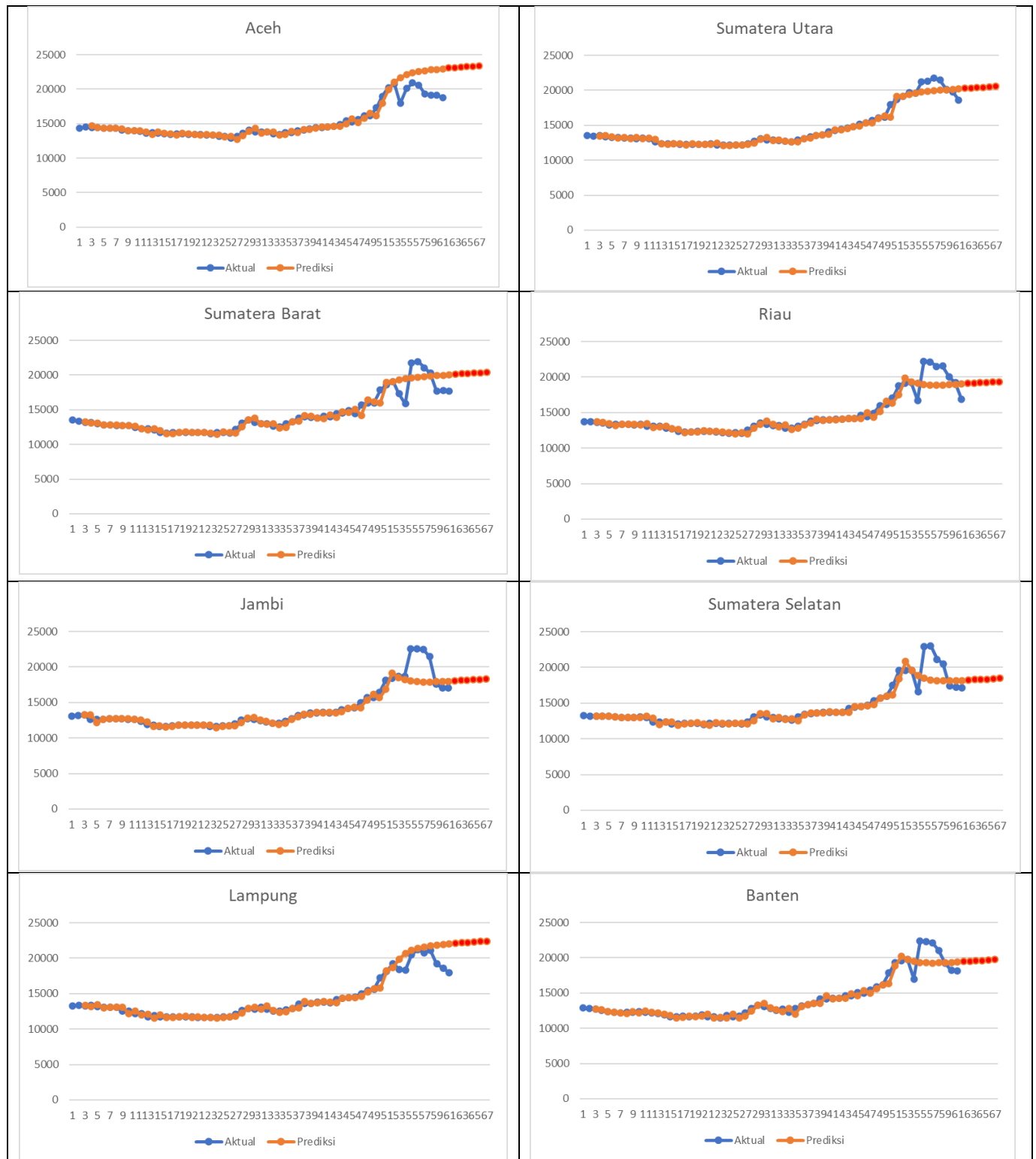
t	$Y(2)_t$	$\hat{Y}(2)_t$	t	$Y(2)_t$	$\hat{Y}(2)_t$	t	$Y(2)_t$	$\hat{Y}(2)_t$
1	14.600	14.604,01	24	14.030	13.992,44	47	15.770	15.752,84
2	14.513	14.597,85	25	13.977	14.038,52	48	15.913	15.773,87
3	14.530	14.494,72	26	13.943	13.963,33	49	16.153	15.945,10
4	14.417	14.537,98	27	13.970	13.938,44	50	17.113	16.201,55
5	14.450	14.390,41	28	14.060	13.977,15	51	18.697	17.353,90
6	14.373	14.463,57	29	14.237	14.078,96	52	20.157	19.074,36
7	14.347	14.352,98	30	14.957	14.273,72	53	19.567*	19.918,82*
8	14.373	14.345,66	31	14.497	15.133,29	54	18.710*	19.868,83*
9	14.363	14.379,19	32	14.523	14.369,90	55	25.670*	19.865,53*
10	14.370	14.359,37	33	14.450	14.558,51	56	26.213*	19.898,82*
11	14.363	14.372,40	34	14.363	14.426,09	57	25.853*	19.940,32*
12	14.327	14.360,89	35	14.240	14.348,98	58	24.893*	19.916,32*
13	14.270	14.319,42	36	14.360	14.216	59	22.580*	19.853,03*
14	14.237	14.258,99	37	14.497	14.393,35	60	21.340*	19.814,26*
15	14.150	14.232,07	38	14.547	14.520,81	61	20.747*	19.676,87*
16	14.090	14.131,83	39	14.583	14.552,92	62		^19.650,64
17	14.090	14.080,67	40	14.687	14.589,81	63		^19.601,89
18	14.100	14.092,10	41	14.740	14.709,29	64		^19.604,14
19	14.050	14.101,78	42	14.833	14.746,95	65		^19.453,35
20	14.047	14.038,47	43	14.930	14.852,69	66		^19.559,50
21	14.027	14.048,92	44	15.253	14.947,66	67		^19.477,40
22	13.987	14.022,09	45	15.297	15.325,83			
23	13.990	13.979,16	46	15.663	15.290,55			

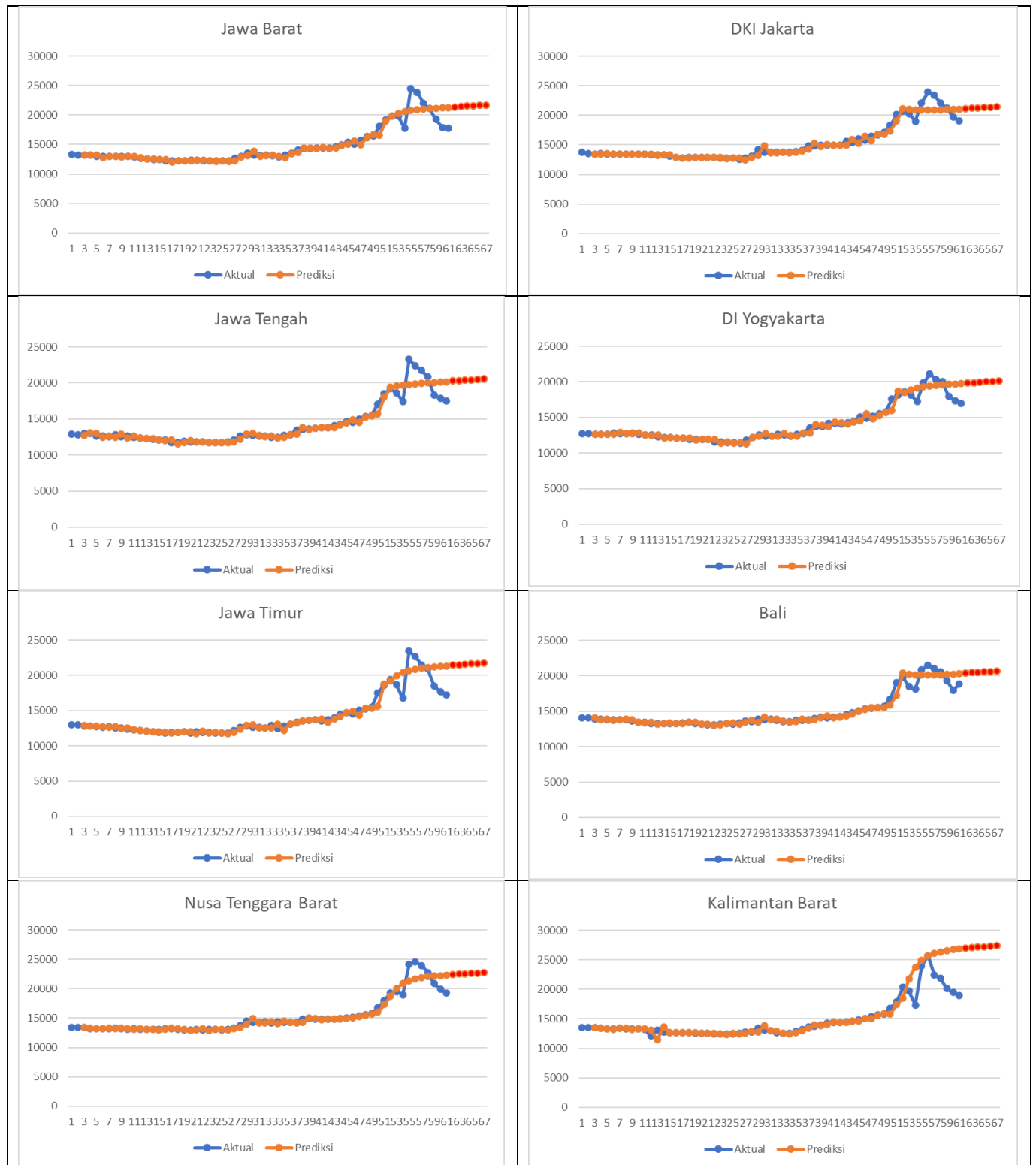
Keterangan: (*) Data *out-sample* ; (^) Peramalan

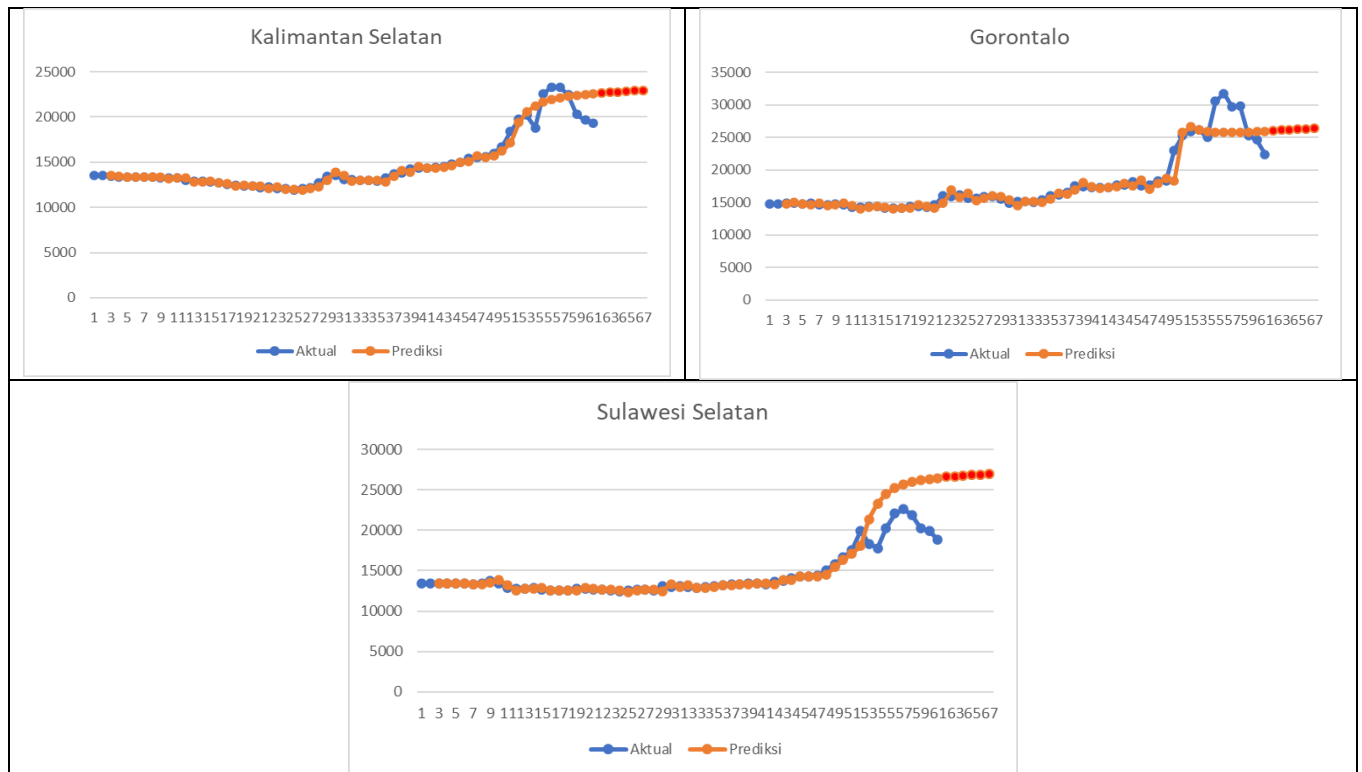
Lampiran 34. Data Aktual dan Prediksi untuk Objek *Prototype* Klaster 2 model ARIMA (1,1,0)

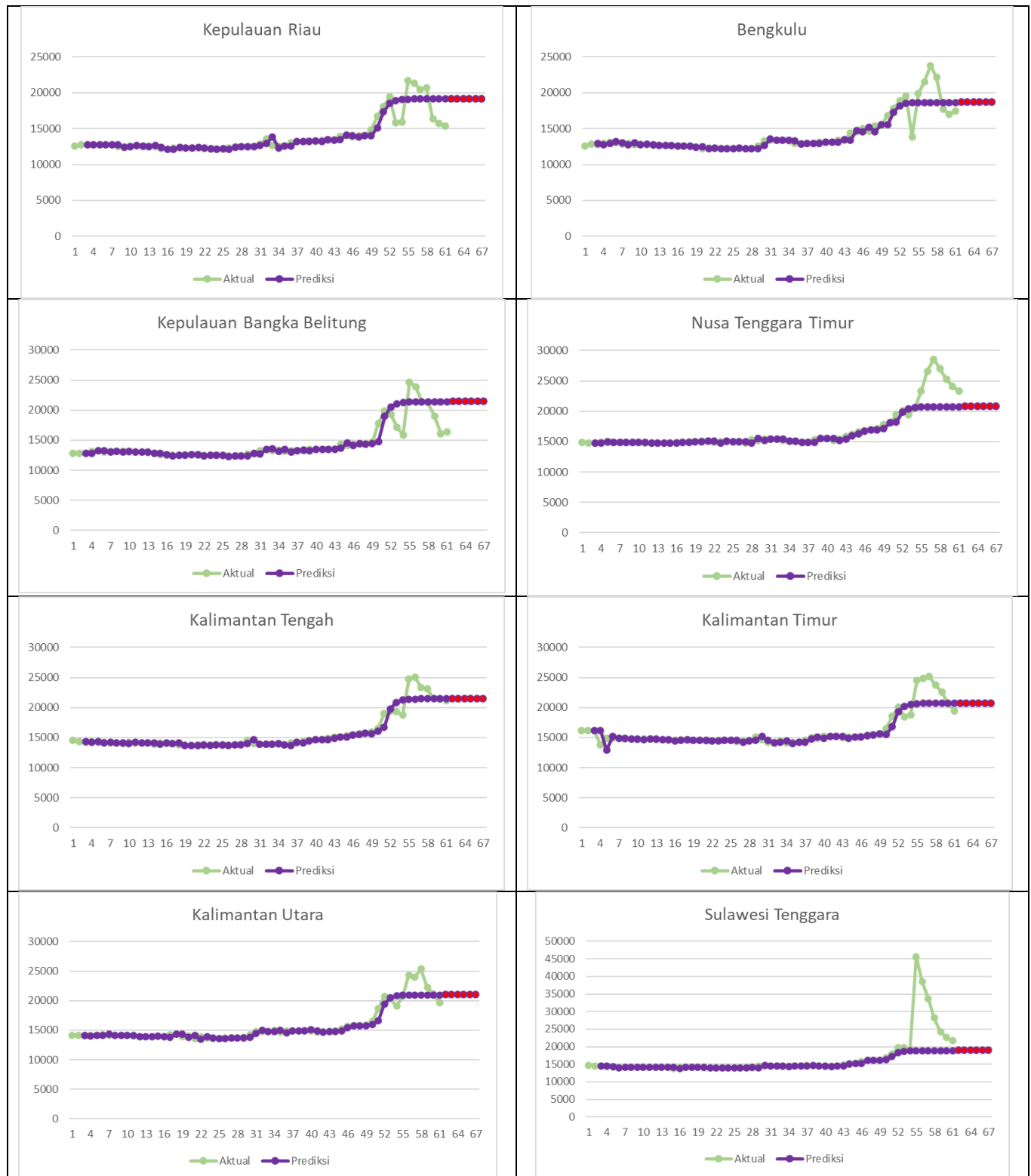
t	$Y(2)_t$	$\hat{Y}(2)_t$	t	$Y(2)_t$	$\hat{Y}(2)_t$	t	$Y(2)_t$	$\hat{Y}(2)_t$
1	14.600	14.604,01	24	14.030	13.991,04	47	15.770	15.799,76
2	14.513	14.594,54	25	13.977	14.043,99	48	15.913	15.807,88
3	14.530	14.483,40	26	13.943	13.958,85	49	16.153	15.963,99
4	14.417	14.535,91	27	13.970	13.931,31	50	17.113	16.240,20
5	14.450	14.378,78	28	14.060	13.979,41	51	18.697	17.515,01
6	14.373	14.461,52	29	14.237	14.091,83	52	20.157	19.437,01
7	14.347	14.346,75	30	14.957	14.300,81	53	19.567*	20.806,97*
8	14.373	14.338,04	31	14.497	15.249,97	54	18.710*	21.056,34*
9	14.363	14.382,06	32	14.523	14.352,08	55	25.670*	21.146,07*
10	14.370	14.359,54	33	14.450	14.532,06	56	26.213*	21.177,58*
11	14.363	14.372,43	34	14.363	14.425,09	57	25.853*	21.188,55*
12	14.327	14.360,58	35	14.240	14.333,41	58	24.893*	21.192,36*
13	14.270	14.314,62	36	14.360	14.198,50	59	22.580*	21.193,68*
14	14.237	14.250,49	37	14.497	14.402,70	60	21.340*	21.194,14*
15	14.150	14.225,65	38	14.547	14.545,92	61	20.747*	21.194,30*
16	14.090	14.120,42	39	14.583	14.564,52	62		^21.194,35
17	14.090	14.069,48	40	14.687	14.595,57	63		^21.194,37
18	14.100	14.090	41	14.740	14.723,86	64		^21.194,38
19	14.050	14.103,47	42	14.833	14.758,58	65		^21.194,38
20	14.047	14.032,86	43	14.930	14.865,87	66		^21.194,38
21	14.027	14.045,96	44	15.253	14.964,31	67		^21.194,38
22	13.987	14.020,10	45	15.297	15.372,82			
23	13.990	13.973,26	46	15.663	15.312,39			

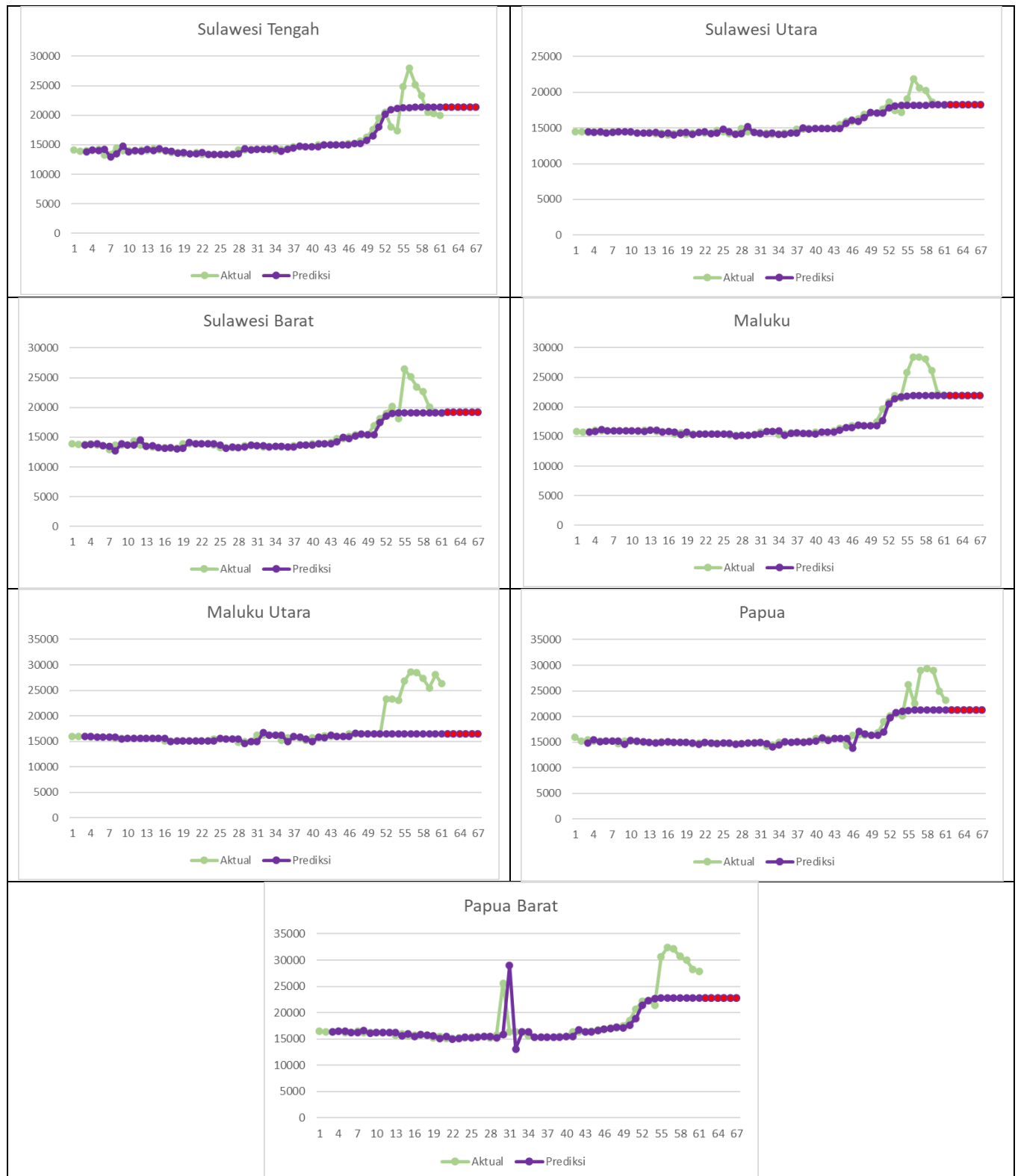
Keterangan: (*) Data *out-sample* ; (^) Peramalan

Lampiran 35. Plot Aktual dan Prediksi Setiap Provinsi di Klaster 1

Lampiran 35. Plot aktual dan prediksi tiap Provinsi di kluster 1 (lanjutan)

Lampiran 35. Plot Aktual dan Prediksi Setiap Provinsi di Klaster 1 (lanjutan)

Lampiran 36. Plot Aktual dan Prediksi Setiap Provinsi di Klaster 2

Lampiran 36. Plot Aktual dan Prediksi Setiap Provinsi di Klaster 2 (lanjutan)

Lampiran 37. Command R untuk Nilai MAPE Objek Prototype Klaster 1

```

> #MAPE FIT 1 (ARIMA (0,1,1))
> act1c1=read.csv(file.choose(),header=TRUE)
> n<-length(act1c1$prediksi)
> #Insample
> PEin1=array(NA, dim=c(n))
> for(i in 1:52){
+   PEin1[i]=abs((act1c1$Average[i]
+     act1c1$prediksi[i])/(act1c1$Average[i]))}
> MAPEin1=mean(PEin1,na.rm=TRUE)*100
> MAPEin1
[1] 1.003213
> #Outsample
> PEout1=array(NA, dim=c(n))
> for(i in 53:61){
+   PEout1[i]=abs((act1c1$Average[i]
+     act1c1$prediksi[i])/(act1c1$Average[i]))}
> MAPEout1=mean(PEout1,na.rm=TRUE)*100
> MAPEout1
[1] 7.989837

> #MAPE FIT 2 (ARIMA (0,1,2))
> act1c1=read.csv(file.choose(),header=TRUE)
> n<-length(act2c1$prediksi)
> #Insample
> PEin2=array(NA, dim=c(n))
> for(i in 1:52){
+   PEin2[i]=abs((act2c1$Average[i]
+     act2c1$prediksi[i])/(act2c1$Average[i]))}
> MAPEin2=mean(PEin2,na.rm=TRUE)*100
> MAPEin2
[1] 0.8748912
> #Outsample
> PEout2=array(NA, dim=c(n))
> for(i in 53:61){
+   PEout2[i]=abs((act2c1$Average[i]
+     act2c1$prediksi[i])/(act2c1$Average[i]))}
> MAPEout2=mean(PEout2,na.rm=TRUE)*100
> MAPEout2
[1] 7.303995

```


Lampiran 37. Command R untuk Nilai MAPE Objek *Prototype* Klaster 1 (lanjutan)

```

> #MAPE FIT 4 (ARIMA (1,1,0))
> act4c1=read.csv(file.choose(),header=TRUE)
> n<-length(act4c1$prediksi)
> #Insample
> PEin4=array(NA, dim=c(n))
> for(i in 1:52){
+   PEin4[i]=abs((act4c1$Average[i]
+     act4c1$prediksi[i])/(act4c1$Average[i]))}
> MAPEin4=mean(PEin4,na.rm=TRUE)*100
> MAPEin4
[1] 0.8493679
> #Outsample
> PEout4=array(NA, dim=c(n))
> for(i in 53:61){
+   PEout4[i]=abs((act4c1$Average[i]
+     act4c1$prediksi[i])/(act4c1$Average[i]))}
> MAPEout4=mean(PEout4,na.rm=TRUE)*100
> MAPEout4
[1] 8.764212

```

Lampiran 38. Command R untuk Nilai MAPE Objek Prototype Klaster 2

```

> #MAPE FIT 1 (ARIMA (0,1,1))
> act1c2=read.csv(file.choose(),header=TRUE)
> n<-length(act1c2$prediksi)
> #Insample
> PEin1=array(NA, dim=c(n))
> for(i in 1:52){
+   PEin1[i]=abs((act1c2$Average[i]
+     act1c2$prediksi[i])/(act1c2$Average[i]))}
> MAPEin1=mean(PEin1,na.rm=TRUE)*100
> MAPEin1
[1] 0.9501537
> #Outsample
> PEout1=array(NA, dim=c(n))
> for(i in 53:61){
+   PEout1[i]=abs((act1c2$Average[i]
+     act1c2$prediksi[i])/(act1c2$Average[i]))}
> MAPEout1=mean(PEout1,na.rm=TRUE)*100
> MAPEout1
[1] 13.54875

> #MAPE FIT 2 (ARIMA (1,1,0))
> act2c2=read.csv(file.choose(),header=TRUE)
> n<-length(act2c2$prediksi)
> #Insample
> PEin2=array(NA, dim=c(n))
> for(i in 1:52){
+   PEin2[i]=abs((act2c2$Average[i]-
+     act2c2$prediksi[i])/(act2c2$Average[i]))}
> MAPEin2=mean(PEin2,na.rm=TRUE)*100
> MAPEin2
[1] 0.890551
> #Outsample
> PEout2=array(NA, dim=c(n))
> for(i in 53:61){
+   PEout2[i]=abs((act2c2$Average[i]-
+     act2c2$prediksi[i])/(act2c2$Average[i]))}
> MAPEout2=mean(PEout2,na.rm=TRUE)*100
> MAPEout2
[1] 10.84423

```

Lampiran 39. Command R untuk Nilai MAPE Setiap Provinsi di Klaster 1

```

> #MAPE level individu klaster 1
> indc1=read.csv(file.choose(),header=TRUE)

> #Insample
> n<-length(indc1$prediksi)
> inPE1=array(NA, dim=c(n))
> inPE2=array(NA, dim=c(n))
> inPE3=array(NA, dim=c(n))
> inPE4=array(NA, dim=c(n))
> inPE5=array(NA, dim=c(n))
> inPE6=array(NA, dim=c(n))
> inPE7=array(NA, dim=c(n))
> inPE8=array(NA, dim=c(n))
> inPE9=array(NA, dim=c(n))
> inPE10=array(NA, dim=c(n))
> inPE11=array(NA, dim=c(n))
> inPE12=array(NA, dim=c(n))
> inPE13=array(NA, dim=c(n))
> inPE14=array(NA, dim=c(n))
> inPE15=array(NA, dim=c(n))
> inPE16=array(NA, dim=c(n))
> inPE17=array(NA, dim=c(n))
> inPE18=array(NA, dim=c(n))
> inPE19=array(NA, dim=c(n))
> for(i in 1:52){
+   inPE1[i]=abs((indc1$ACEH[i]-indc1$pACEH[i])/(indc1$ACEH[i]))
+   inPE2[i]=abs((indc1$SUMATERA.UTARA[i]-
+     indc1$pSUMATERA.UTARA[i])/(indc1$SUMATERA.UTARA[i]))
+   inPE3[i]=abs((indc1$SUMATERA.BARAT[i]-
+     indc1$pSUMATERA.BARAT[i])/(indc1$SUMATERA.BARAT[i]))
+   inPE4[i]=abs((indc1$RIAU[i]-indc1$pRIAU[i])/(indc1$RIAU[i]))
+   inPE5[i]=abs((indc1$JAMBI[i]-
+     indc1$pJAMBI[i])/(indc1$JAMBI[i]))
+   inPE6[i]=abs((indc1$SUMATERA.SELATAN[i]-
+     indc1$pSUMATERA.SELATAN[i])/(indc1$SUMATERA.SELATAN[i]))
+   inPE7[i]=abs((indc1$LAMPUNG[i]-
+     indc1$pLAMPUNG[i])/(indc1$LAMPUNG[i]))
+   inPE8[i]=abs((indc1$BANTEN[i]-
+     indc1$pBANTEN[i])/(indc1$BANTEN[i]))
+   inPE9[i]=abs((indc1$JAWA.BARAT[i]-
+     indc1$pJAWA.BARAT[i])/(indc1$JAWA.BARAT[i]))
+   inPE10[i]=abs((indc1$DKI.JAKARTA[i]-
+     indc1$pDKI.JAKARTA[i])/(indc1$DKI.JAKARTA[i]))
+   inPE11[i]=abs((indc1$JAWA.TENGAH[i]-
+     indc1$pJAWA.TENGAH[i])/(indc1$JAWA.TENGAH[i]))
+   inPE12[i]=abs((indc1$DI.YOGYAKARTA[i]-
+     indc1$pDI.YOGYAKARTA[i])/(indc1$DI.YOGYAKARTA[i]))
+   inPE13[i]=abs((indc1$JAWA.TIMUR[i]-
+     indc1$pJAWA.TIMUR[i])/(indc1$JAWA.TIMUR[i]))
+   inPE14[i]=abs((indc1$BALI[i]-
+     indc1$pBALI[i])/(indc1$BALI[i]))
+   inPE15[i]=abs((indc1$NUSA.TENGGARA.BARAT[i]-
+     indc1$pNUSA.TENGGARA.BARAT[i])/(
+     indc1$NUSA.TENGGARA.BARAT[i]))

```

Lampiran 39. Command R untuk Nilai MAPE Setiap Provinsi di Kluster 1 (lanjutan)

```
+   inPE16[i]=abs((indcl$KALIMANTAN.BARAT[i]-
+   indcl$pKALIMANTAN.BARAT[i])/(indcl$KALIMANTAN.BARAT[i]))
+   inPE17[i]=abs((indcl$KALIMANTAN.SELATAN[i]-
+   indcl$pKALIMANTAN.SELATAN[i])/(indcl$KALIMANTAN.SELATAN[i]))
+   inPE18[i]=abs((indcl$GORONTALO[i]-
+   indcl$pGORONTALO[i])/(indcl$GORONTALO[i]))
+   inPE19[i]=abs((indcl$SULAWESI.SELATAN[i]-
+   indcl$pSULAWESI.SELATAN[i])/(indcl$SULAWESI.SELATAN[i]))}

> inMAPE1=mean(inPE1,na.rm=TRUE)*100
> inMAPE2=mean(inPE2,na.rm=TRUE)*100
> inMAPE3=mean(inPE3,na.rm=TRUE)*100
> inMAPE4=mean(inPE4,na.rm=TRUE)*100
> inMAPE5=mean(inPE5,na.rm=TRUE)*100
> inMAPE6=mean(inPE6,na.rm=TRUE)*100
> inMAPE7=mean(inPE7,na.rm=TRUE)*100
> inMAPE8=mean(inPE8,na.rm=TRUE)*100
> inMAPE9=mean(inPE9,na.rm=TRUE)*100
> inMAPE10=mean(inPE10,na.rm=TRUE)*100
> inMAPE11=mean(inPE11,na.rm=TRUE)*100
> inMAPE12=mean(inPE12,na.rm=TRUE)*100
> inMAPE13=mean(inPE13,na.rm=TRUE)*100
> inMAPE14=mean(inPE14,na.rm=TRUE)*100
> inMAPE15=mean(inPE15,na.rm=TRUE)*100
> inMAPE16=mean(inPE16,na.rm=TRUE)*100
> inMAPE17=mean(inPE17,na.rm=TRUE)*100
> inMAPE18=mean(inPE18,na.rm=TRUE)*100
> inMAPE19=mean(inPE19,na.rm=TRUE)*100

> #Outsample
> outPE1=array(NA, dim=c(n))
> outPE2=array(NA, dim=c(n))
> outPE3=array(NA, dim=c(n))
> outPE4=array(NA, dim=c(n))
> outPE5=array(NA, dim=c(n))
> outPE6=array(NA, dim=c(n))
> outPE7=array(NA, dim=c(n))
> outPE8=array(NA, dim=c(n))
> outPE9=array(NA, dim=c(n))
> outPE10=array(NA, dim=c(n))
> outPE11=array(NA, dim=c(n))
> outPE12=array(NA, dim=c(n))
> outPE13=array(NA, dim=c(n))
> outPE14=array(NA, dim=c(n))
> outPE15=array(NA, dim=c(n))
> outPE16=array(NA, dim=c(n))
> outPE17=array(NA, dim=c(n))
> outPE18=array(NA, dim=c(n))
> outPE19=array(NA, dim=c(n))
```

Lampiran 39. Command R untuk Nilai MAPE Setiap Provinsi di Kluster 1 (lanjutan)

```

> for(i in 53:61){
+   outPE1[i]=abs((indcl$ACEH[i]-
+     indcl$pACEH[i])/(indcl$ACEH[i]))
+   outPE2[i]=abs((indcl$SUMATERA.UTARA[i]-
+     indcl$pSUMATERA.UTARA[i])/(indcl$SUMATERA.UTARA[i]))
+   outPE3[i]=abs((indcl$SUMATERA.BARAT[i]-
+     indcl$pSUMATERA.BARAT[i])/(indcl$SUMATERA.BARAT[i]))
+   outPE4[i]=abs((indcl$RIAU[i]-
+     indcl$pRIAU[i])/(indcl$RIAU[i]))
+   outPE5[i]=abs((indcl$JAMBI[i]-
+     indcl$pJAMBI[i])/(indcl$JAMBI[i]))
+   outPE6[i]=abs((indcl$SUMATERA.SELATAN[i]-
+     indcl$pSUMATERA.SELATAN[i])/(indcl$SUMATERA.SELATAN[i]))
+   outPE7[i]=abs((indcl$LAMPUNG[i]-
+     indcl$pLAMPUNG[i])/(indcl$LAMPUNG[i]))
+   outPE8[i]=abs((indcl$BANTEN[i]-
+     indcl$pBANTEN[i])/(indcl$BANTEN[i]))
+   outPE9[i]=abs((indcl$JAWA.BARAT[i]-
+     indcl$pJAWA.BARAT[i])/(indcl$JAWA.BARAT[i]))
+   outPE10[i]=abs((indcl$DKI.JAKARTA[i]-
+     indcl$pDKI.JAKARTA[i])/(indcl$DKI.JAKARTA[i]))
+   outPE11[i]=abs((indcl$JAWA.TENGAH[i]-
+     indcl$pJAWA.TENGAH[i])/(indcl$JAWA.TENGAH[i]))
+   outPE12[i]=abs((indcl$DI.YOGYAKARTA[i]-
+     indcl$pDI.YOGYAKARTA[i])/(indcl$DI.YOGYAKARTA[i]))
+   outPE13[i]=abs((indcl$JAWA.TIMUR[i]-
+     indcl$pJAWA.TIMUR[i])/(indcl$JAWA.TIMUR[i]))
+   outPE14[i]=abs((indcl$BALI[i]-
+     indcl$pBALI[i])/(indcl$BALI[i]))
+   outPE15[i]=abs((indcl$NUSA.TENGGAH.BARAT[i]-
+     indcl$pNUSA.TENGGAH.BARAT[i])/(
+     indcl$NUSA.TENGGAH.BARAT[i]))
+   outPE16[i]=abs((indcl$KALIMANTAN.BARAT[i]-
+     indcl$pKALIMANTAN.BARAT[i])/(indcl$KALIMANTAN.BARAT[i]))
+   outPE17[i]=abs((indcl$KALIMANTAN.SELATAN[i]-
+     indcl$pKALIMANTAN.SELATAN[i])/(indcl$KALIMANTAN.SELATAN[i]))
+   outPE18[i]=abs((indcl$GORONTALO[i]-
+     indcl$pGORONTALO[i])/(indcl$GORONTALO[i]))
+   outPE19[i]=abs((indcl$SULAWESI.SELATAN[i]-
+     indcl$pSULAWESI.SELATAN[i])/(indcl$SULAWESI.SELATAN[i]))}

> outMAPE1=mean(outPE1,na.rm=TRUE)*100
> outMAPE2=mean(outPE2,na.rm=TRUE)*100
> outMAPE3=mean(outPE3,na.rm=TRUE)*100
> outMAPE4=mean(outPE4,na.rm=TRUE)*100
> outMAPE5=mean(outPE5,na.rm=TRUE)*100
> outMAPE6=mean(outPE6,na.rm=TRUE)*100
> outMAPE7=mean(outPE7,na.rm=TRUE)*100
> outMAPE8=mean(outPE8,na.rm=TRUE)*100
> outMAPE9=mean(outPE9,na.rm=TRUE)*100
> outMAPE10=mean(outPE10,na.rm=TRUE)*100
> outMAPE11=mean(outPE11,na.rm=TRUE)*100
> outMAPE12=mean(outPE12,na.rm=TRUE)*100

```

Lampiran 39. Command R untuk Nilai MAPE Setiap Provinsi di Kluster 1 (lanjutan)

```

> outMAPE13=mean(outPE13,na.rm=TRUE)*100
> outMAPE14=mean(outPE14,na.rm=TRUE)*100
> outMAPE15=mean(outPE15,na.rm=TRUE)*100
> outMAPE16=mean(outPE12,na.rm=TRUE)*100
> outMAPE17=mean(outPE13,na.rm=TRUE)*100
> outMAPE18=mean(outPE14,na.rm=TRUE)*100
> outMAPE19=mean(outPE15,na.rm=TRUE)*100

> provc1<-c("Aceh","Sumatera Utara","Sumatera Barat","Riau",
+           "Jambi","Sumatera Selatan","Lampung",
+           "Banten","Jawa Barat","DKI Jakarta","Jawa Tengah",
+           "DI Yogyakarta","Jawa Timur","Bali",
+           "Nusa Tenggara Barat","Kalimantan Barat",
+           "Kalimantan Selatan","Gorontalo","Sulawesi Selatan")

> MAPEinc1<-c(inMAPE1,inMAPE2,inMAPE3,inMAPE4,inMAPE5,inMAPE6,
+             inMAPE7,inMAPE8,inMAPE9,
+             inMAPE10,inMAPE11,inMAPE12,inMAPE13,inMAPE14,
+             inMAPE15,inMAPE16,inMAPE17,inMAPE18,inMAPE19)

> MAPEoutc1<-c(outMAPE1,outMAPE2,outMAPE3,outMAPE4,outMAPE5,
+              outMAPE6,outMAPE7,outMAPE8,outMAPE9,
+              outMAPE10,outMAPE11,outMAPE12,outMAPE13,
+              outMAPE14,outMAPE15,outMAPE16,outMAPE17,
+              outMAPE18,outMAPE19)

> tabelmapec1=data.frame(provc1,MAPEinc1,MAPEoutc1)
> tabelmapec1

```

	provc1	MAPEinc1	MAPEoutc1
1	Aceh	1.378163	14.078798
2	Sumatera Utara	1.080998	4.580222
3	Sumatera Barat	1.772367	11.287380
4	Riau	1.553806	9.820386
5	Jambi	1.433554	10.310456
6	Sumatera Selatan	1.477772	10.542318
7	Lampung	1.453926	9.450777
8	Banten	1.819086	8.496072
9	Jawa Barat	1.275535	10.924775
10	DKI Jakarta	1.552126	7.309654
11	Jawa Tengah	1.377187	10.421460
12	DI Yogyakarta	1.502685	7.903116
13	Jawa Timur	1.362760	12.228380
14	Bali	1.184314	6.791731
15	Nusa Tenggara Barat	1.143129	9.007944
16	Kalimantan Barat	1.708655	7.903116
17	Kalimantan Selatan	1.212315	12.228380
18	Gorontalo	2.449642	6.791731
19	Sulawesi Selatan	1.237202	9.007944

```

> write_xlsx(tabelmapec1,"D:/Draft Proposal/Data & Sintaks
+           /tabelmapec1.xlsx")

```

Lampiran 40. Command R untuk Nilai MAPE Setiap Provinsi di Klaster 2

```

> #MAPE level individu klaster 2
> indc2=read.csv(file.choose(),header=TRUE)

> #Insample
> n<-length(indc2$prediksi)
> inPE1=array(NA, dim=c(n))
> inPE2=array(NA, dim=c(n))
> inPE3=array(NA, dim=c(n))
> inPE4=array(NA, dim=c(n))
> inPE5=array(NA, dim=c(n))
> inPE6=array(NA, dim=c(n))
> inPE7=array(NA, dim=c(n))
> inPE8=array(NA, dim=c(n))
> inPE9=array(NA, dim=c(n))
> inPE10=array(NA, dim=c(n))
> inPE11=array(NA, dim=c(n))
> inPE12=array(NA, dim=c(n))
> inPE13=array(NA, dim=c(n))
> inPE14=array(NA, dim=c(n))
> inPE15=array(NA, dim=c(n))
> for(i in 1:52){
+   inPE1[i]=abs((indc2$KEPULAUAN.RIAU[i]-
+     indc2$pKEPULAUAN.RIAU[i])/(indc2$KEPULAUAN.RIAU[i]))
+   inPE2[i]=abs((indc2$BENGKULU[i]-
+     indc2$pBENGKULU[i])/(indc2$BENGKULU[i]))
+   inPE3[i]=abs((indc2$KEPULAUAN.BANGKA.BELITUNG[i]-
+     indc2$pKEPULAUAN.BANGKA.BELITUNG[i])/(
+     indc2$KEPULAUAN.BANGKA.BELITUNG[i]))
+   inPE4[i]=abs((indc2$NUSA.TENGGA.TIMUR[i]-
+     indc2$pNUSA.TENGGA.TIMUR[i])/(
+     indc2$NUSA.TENGGA.TIMUR[i]))
+   inPE5[i]=abs((indc2$KALIMANTAN.TENGAH[i]-
+     indc2$pKALIMANTAN.TENGAH[i])/(indc2$KALIMANTAN.TENGAH[i]))
+   inPE6[i]=abs((indc2$KALIMANTAN.TIMUR[i]-
+     indc2$pKALIMANTAN.TIMUR[i])/(indc2$KALIMANTAN.TIMUR[i]))
+   inPE7[i]=abs((indc2$KALIMANTAN.UTARA[i]-
+     indc2$pKALIMANTAN.UTARA[i])/(indc2$KALIMANTAN.UTARA[i]))
+   inPE8[i]=abs((indc2$SULAWESI.TENGGA[i]-
+     indc2$pSULAWESI.TENGGA[i])/(indc2$SULAWESI.TENGGA[i]))
+   inPE9[i]=abs((indc2$SULAWESI.TENGAH[i]-
+     indc2$pSULAWESI.TENGAH[i])/(indc2$SULAWESI.TENGAH[i]))
+   inPE10[i]=abs((indc2$SULAWESI.UTARA[i]-
+     indc2$pSULAWESI.UTARA[i])/(indc2$SULAWESI.UTARA[i]))
+   inPE11[i]=abs((indc2$SULAWESI.BARAT[i]-
+     indc2$pSULAWESI.BARAT[i])/(indc2$SULAWESI.BARAT[i]))
+   inPE12[i]=abs((indc2$MALUKU[i]-
+     indc2$pMALUKU[i])/(indc2$MALUKU[i]))
+   inPE13[i]=abs((indc2$MALUKU.UTARA[i]-
+     indc2$pMALUKU.UTARA[i])/(indc2$MALUKU.UTARA[i]))
+   inPE14[i]=abs((indc2$PAPUA[i]-
+     indc2$pPAPUA[i])/(indc2$PAPUA[i]))
+   inPE15[i]=abs((indc2$PAPUA.BARAT[i]-
+     indc2$pPAPUA.BARAT[i])/(indc2$PAPUA.BARAT[i]))}

```

Lampiran 40. Command R untuk Nilai MAPE Setiap Provinsi di Klaster 2 (lanjutan)

```

> inMAPE1=mean(inPE1,na.rm=TRUE)*100
> inMAPE2=mean(inPE2,na.rm=TRUE)*100
> inMAPE3=mean(inPE3,na.rm=TRUE)*100
> inMAPE4=mean(inPE4,na.rm=TRUE)*100
> inMAPE5=mean(inPE5,na.rm=TRUE)*100
> inMAPE6=mean(inPE6,na.rm=TRUE)*100
> inMAPE7=mean(inPE7,na.rm=TRUE)*100
> inMAPE8=mean(inPE8,na.rm=TRUE)*100
> inMAPE9=mean(inPE9,na.rm=TRUE)*100
> inMAPE10=mean(inPE10,na.rm=TRUE)*100
> inMAPE11=mean(inPE11,na.rm=TRUE)*100
> inMAPE12=mean(inPE12,na.rm=TRUE)*100
> inMAPE13=mean(inPE13,na.rm=TRUE)*100
> inMAPE14=mean(inPE14,na.rm=TRUE)*100
> inMAPE15=mean(inPE15,na.rm=TRUE)*100

> #Outsample
> outPE1=array(NA, dim=c(n))
> outPE2=array(NA, dim=c(n))
> outPE3=array(NA, dim=c(n))
> outPE4=array(NA, dim=c(n))
> outPE5=array(NA, dim=c(n))
> outPE6=array(NA, dim=c(n))
> outPE7=array(NA, dim=c(n))
> outPE8=array(NA, dim=c(n))
> outPE9=array(NA, dim=c(n))
> outPE10=array(NA, dim=c(n))
> outPE11=array(NA, dim=c(n))
> outPE12=array(NA, dim=c(n))
> outPE13=array(NA, dim=c(n))
> outPE14=array(NA, dim=c(n))
> outPE15=array(NA, dim=c(n))

> for(i in 53:61){
+   outPE1[i]=abs((indc2$KEPULAUAN.RIAU[i]-
+     indc2$pKEPULAUAN.RIAU[i])/(indc2$KEPULAUAN.RIAU[i]))
+   outPE2[i]=abs((indc2$BENGKULU[i]-
+     indc2$pBENGKULU[i])/(indc2$BENGKULU[i]))
+   outPE3[i]=abs((indc2$KEPULAUAN.BANGKA.BELITUNG[i]-
+     indc2$pKEPULAUAN.BANGKA.BELITUNG[i])/
+     (indc2$KEPULAUAN.BANGKA.BELITUNG[i]))
+   outPE4[i]=abs((indc2$NUSA.TENGGA.TIMUR[i]-
+     indc2$pNUSA.TENGGA.TIMUR[i])/
+     (indc2$NUSA.TENGGA.TIMUR[i]))
+   outPE5[i]=abs((indc2$KALIMANTAN.TENGAH[i]-
+     indc2$pKALIMANTAN.TENGAH[i])/(indc2$KALIMANTAN.TENGAH[i]))
+   outPE6[i]=abs((indc2$KALIMANTAN.TIMUR[i]-
+     indc2$pKALIMANTAN.TIMUR[i])/(indc2$KALIMANTAN.TIMUR[i]))
+   outPE7[i]=abs((indc2$KALIMANTAN.UTARA[i]-
+     indc2$pKALIMANTAN.UTARA[i])/(indc2$KALIMANTAN.UTARA[i]))
+   outPE8[i]=abs((indc2$SULAWESI.TENGGA[i]-
+     indc2$pSULAWESI.TENGGA[i])/(indc2$SULAWESI.TENGGA[i]))
+   outPE9[i]=abs((indc2$SULAWESI.TENGAH[i]-

```


Lampiran 40. Command R untuk Nilai MAPE Setiap Provinsi di Kluster 2 (lanjutan)

```

      indc2$pSULAWESI.TENGAH[i])/(indc2$SULAWESI.TENGAH[i]))
+   outPE10[i]=abs((indc2$SULAWESI.UTARA[i]-
      indc2$pSULAWESI.UTARA[i])/(indc2$SULAWESI.UTARA[i]))
+   outPE11[i]=abs((indc2$SULAWESI.BARAT[i]-
      indc2$pSULAWESI.BARAT[i])/(indc2$SULAWESI.BARAT[i]))
+   outPE12[i]=abs((indc2$MALUKU[i]-
      indc2$pMALUKU[i])/(indc2$MALUKU[i]))
+   outPE13[i]=abs((indc2$MALUKU.UTARA[i]-
      indc2$pMALUKU.UTARA[i])/(indc2$MALUKU.UTARA[i]))
+   outPE14[i]=abs((indc2$PAPUA[i]-
      indc2$pPAPUA[i])/(indc2$PAPUA[i]))
+   outPE15[i]=abs((indc2$PAPUA.BARAT[i]-
      indc2$pPAPUA.BARAT[i])/(indc2$PAPUA.BARAT[i]))}

> outMAPE1=mean(outPE1,na.rm=TRUE)*100
> outMAPE2=mean(outPE2,na.rm=TRUE)*100
> outMAPE3=mean(outPE3,na.rm=TRUE)*100
> outMAPE4=mean(outPE4,na.rm=TRUE)*100
> outMAPE5=mean(outPE5,na.rm=TRUE)*100
> outMAPE6=mean(outPE6,na.rm=TRUE)*100
> outMAPE7=mean(outPE7,na.rm=TRUE)*100
> outMAPE8=mean(outPE8,na.rm=TRUE)*100
> outMAPE9=mean(outPE9,na.rm=TRUE)*100
> outMAPE10=mean(outPE10,na.rm=TRUE)*100
> outMAPE11=mean(outPE11,na.rm=TRUE)*100
> outMAPE12=mean(outPE12,na.rm=TRUE)*100
> outMAPE13=mean(outPE13,na.rm=TRUE)*100
> outMAPE14=mean(outPE14,na.rm=TRUE)*100
> outMAPE15=mean(outPE15,na.rm=TRUE)*100

> provc2<-c("Kepulauan Riau","Bengkulu",
      "Kepulauan Bangka Belitung","Nusa Tenggara Timur",
+   "Kalimantan Tengah","Kalimantan Timur",
      "Kalimantan Utara","Sulawesi Tenggara",
      "Sulawesi Tengah","Sulawesi Utara","Sulawesi Barat",
      "Maluku","Maluku Utara","Papua","Papua Barat")

> MAPEinc2<-c(inMAPE1,inMAPE2,inMAPE3,inMAPE4,inMAPE5,inMAPE6,
      inMAPE7,inMAPE8,inMAPE9,inMAPE10,inMAPE11,
      inMAPE12,inMAPE13,inMAPE14,inMAPE15)

> MAPEoutc2<-c(outMAPE1,outMAPE2,outMAPE3,outMAPE4,outMAPE5,
      outMAPE6,outMAPE7,outMAPE8,outMAPE9,outMAPE10,
      outMAPE11,outMAPE12,outMAPE13,outMAPE14,
      outMAPE15)

```

Lampiran 40. *Command R* untuk Nilai MAPE Setiap Provinsi di Klaster 2 (lanjutan)

```
> tabelmapec2=data.frame(provc2,MAPEinc2,MAPEoutc2)
> tabelmapec2
```

	provc2	MAPEinc2	MAPEoutc2
1	Kepulauan Riau	1.5073330	15.451872
2	Bengkulu	1.3729122	13.097998
3	Kepulauan Bangka Belitung	1.5446213	17.486101
4	Nusa Tenggara Timur	0.8766825	14.749729
5	Kalimantan Tengah	1.1483235	7.373999
6	Kalimantan Timur	1.8450454	10.814337
7	Kalimantan Utara	1.3231268	7.509803
8	Sulawesi Tenggara	1.0239075	27.495761
9	Sulawesi Tengah	1.7169404	12.874332
10	Sulawesi Utara	1.2593788	6.068496
11	Sulawesi Barat	1.6453020	11.429052
12	Maluku	1.0605844	11.645794
13	Maluku Utara	1.7652016	37.333340
14	Papua	1.8465797	14.901376
15	Papua Barat	4.1053207	19.823740

```
> write_xlsx(tabelmapec2,"D:/Draft Proposal/Data & Sintaks
/tabelmapec2.xlsx")
```

Lampiran 41. Tabel Harga Kritis D dalam Tes Satu Sampel Kolmogorov-Smirnov

N	Tingkat Signifikansi untuk D = maksimum $ F_0(X) - F_N(X) $				
	0,20	0,15	0,10	0,05	0,01
1	0,900	0,925	0,950	0,975	0,995
2	0,684	0,726	0,776	0,842	0,929
3	0,565	0,597	0,642	0,708	0,828
4	0,494	0,525	0,564	0,624	0,733
5	0,446	0,474	0,510	0,565	0,669
6	0,410	0,436	0,470	0,521	0,618
7	0,381	0,405	0,438	0,486	0,577
8	0,358	0,381	0,411	0,457	0,543
9	0,339	0,360	0,388	0,432	0,514
10	0,322	0,342	0,368	0,410	0,490
11	0,307	0,326	0,352	0,391	0,468
12	0,295	0,313	0,338	0,375	0,450
13	0,284	0,302	0,325	0,361	0,433
14	0,274	0,292	0,314	0,349	0,418
15	0,266	0,283	0,304	0,338	0,404
16	0,258	0,274	0,295	0,328	0,392
17	0,250	0,266	0,286	0,318	0,381
18	0,244	0,259	0,278	0,309	0,371
19	0,237	0,252	0,272	0,301	0,363
20	0,231	0,246	0,264	0,294	0,356
25	0,21	0,22	0,24	0,27	0,32
30	0,19	0,20	0,22	0,24	0,29
35	0,18	0,19	0,21	0,23	0,27
n > 35	$\frac{1,07}{\sqrt{n}}$	$\frac{1,14}{\sqrt{n}}$	$\frac{1,22}{\sqrt{n}}$	$\frac{1,36}{\sqrt{n}}$	$\frac{1,63}{\sqrt{n}}$

Sumber : Siegel, Sidney, 1956, Non Parametric Statistics For The Behavioral Sciences, New York : Mc Graw-Hill Book Company.

Lampiran 42. Tabel Harga Kritis t

db	Taraf signifikansi untuk tes satu sisi	db	Taraf signifikansi untuk tes satu sisi	db	Taraf signifikansi untuk tes satu sisi
	$\alpha = 0,025$		$\alpha = 0,025$		$\alpha = 0,025$
	Taraf signifikansi untuk tes dua sisi		Taraf signifikansi untuk tes dua sisi		Taraf signifikansi untuk tes dua sisi
	$\alpha = 0,05$		$\alpha = 0,05$		$\alpha = 0,05$
1	12,706	22	2,074	43	2,017
2	4,303	23	2,069	44	2,015
3	3,182	24	2,064	45	2,014
4	2,776	25	2,060	46	2,013
5	2,571	26	2,056	47	2,012
6	2,447	27	2,052	48	2,011
7	2,365	28	2,048	49	2,010
8	2,306	29	2,045	50	2,009
9	2,262	30	2,042	51	2,008
10	2,228	31	2,040	52	2,007
11	2,201	32	2,037	53	2,006
12	2,179	33	2,035	54	2,005
13	2,160	34	2,032	55	2,004
14	2,145	35	2,030	56	2,003
15	2,131	36	2,028	57	2,002
16	2,120	37	2,026	58	2,002
17	2,110	38	2,024	59	2,001
18	2,101	39	2,023	60	2,000
19	2,093	40	2,021	61	2,000
20	2,086	41	2,020	62	1,999
21	2,080	42	2,018	63	1,998

Sumber: Perhitungan menggunakan *excel* yang dilakukan oleh penulis

Lampiran 43. Tabel Harga Kritis *Chi-Square* (χ^2)

<i>db</i>	$\alpha = 0,05$	<i>db</i>	$\alpha = 0,05$	<i>db</i>	$\alpha = 0,05$
1	3,841	22	33,924	43	59,304
2	5,991	23	35,172	44	60,481
3	7,815	24	36,415	45	61,656
4	9,488	25	37,652	46	62,830
5	11,070	26	38,885	47	64,001
6	12,592	27	40,113	48	65,171
7	14,067	28	41,337	49	66,339
8	15,507	29	42,557	50	67,505
9	16,919	30	43,773	51	68,669
10	18,307	31	44,985	52	69,832
11	19,675	32	46,194	53	70,993
12	21,026	33	47,400	54	72,153
13	22,362	34	48,602	55	73,311
14	23,685	35	49,802	56	74,468
15	24,996	36	50,998	57	75,624
16	26,296	37	52,192	58	76,778
17	27,587	38	53,384	59	77,931
18	28,869	39	54,572	60	79,082
19	30,144	40	55,758	61	80,232
20	31,410	41	56,942	62	81,381
21	32,671	42	58,124	63	82,529

Sumber: Perhitungan menggunakan *excel* yang dilakukan oleh penulis

RIWAYAT HIDUP



Muhammad Aldani Zen, lahir pada tanggal 1 Februari 2001 di Samarinda, Kalimantan Timur. Anak dari tiga bersaudara pasangan Bapak Subliansyah Zen dan Ibu Lily Hayati. Memulai pendidikan formal pada Tahun 2007-2013 di SD Negeri 006 Samarinda Utara. Menempuh pendidikan selanjutnya pada Tahun 2013-2016 di SMP Negeri 22 Samarinda, kemudian melanjutkan pendidikan pada Tahun 2016-2019 di SMA Negeri 2 Samarinda.

Pendidikan perguruan tinggi dimulai pada Tahun 2019 di Universitas Mulawarman, Fakultas Matematika dan Ilmu Pengetahuan Alam, Jurusan Matematika, Program Studi Statistika melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN). Selama menempuh perkuliahan telah aktif dalam kegiatan-kegiatan akademik seperti menjadi asisten praktikum di Jurusan Matematika untuk beberapa mata kuliah wajib dan pilihan. Aktif dalam komunitas *Community Development Speaker* BEM-KM Universitas Mulawarman periode 2020 sebagai anggota Bidang Kreatif, organisasi Himpunan Mahasiswa Statistika Universitas Mulawarman, yakni periode 2021 sebagai Kepala Bidang Kerohanian, lalu pada periode 2022 sebagai Ketua. Selama menjadi mahasiswa juga turut aktif menjadi enumerator survei diantaranya Survei Potensi Parkir di Wilayah Kota Samarinda oleh Dinas Komunikasi dan Informatika Kota Samarinda pada Tahun 2022 dan Survei Pemantauan Harga oleh Bank Indonesia pada Tahun 2023.

Pada Tahun 2022 mengikuti program Kuliah Kerja Nyata (KKN) oleh Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LP2M) Universitas Mulawarman Angkatan 48 di Kelurahan Gunung Lingai, Kecamatan Sungai Pinang, Samarinda, Kalimantan Timur. Kemudian pada Tanggal 1 September melaksanakan Praktik Kerja Lapangan (PKL) di Badan Pusat Statistik Provinsi Kalimantan Timur.