TUGAS AKHIR – SS 1451501

PENERAPAN MODEL ARIMA, *SUPPORT VECTOR REGRESSION* (SVR), DAN *GENETIC ALGORTIHM – SUPPORT VECTOR REGRESSION* (GA-SVR) UNTUK MERAMALKAN PENJUALAN MOBIL TOYOTA DI INDONESIA

DESSYANA RATNA WIDYASARI
NRP 06211940005008

Dosen Pembimbing
Irhamah, M.Si., Ph.D

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2021

TUGAS AKHIR – SS 1451501

PENERAPAN MODEL ARIMA, *SUPPORT VECTOR REGRESSION* (SVR), DAN *GENETIC ALGORTIHM – SUPPORT VECTOR REGRESSION* (GA-SVR) UNTUK MERAMALKAN PENJUALAN MOBIL TOYOTA DI INDONESIA

DESSYANA RATNA WIDYASARI
NRP 06211940005008

Dosen Pembimbing
Irhamah, M.Si., Ph.D

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2021

FINAL PROJECT – SS 1451501

ARIMA, *SUPPORT VECTOR REGRESSION* (SVR), AND GENETIC ALGORITHM – SUPPORT VECTOR REGRESSION (GA-SVR) MODEL FOR TOYOTA INDONESIA CAR SALES FORECASTING

DESSYANA RATNA WIDYASARI
NRP 06211940005008

Supervisor
Irhamah, M.Si., Ph.D

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF SCIENCE AND DATA ANALYTICS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2021

LEMBAR PENGESAHAN

PENERAPAN MODEL *ARIMA*, *SUPPORT VECTOR REGRESSION* (SVR), DAN *GENETIC ALGORTIHM – SUPPORT VECTOR REGRESSION* (GA-SVR) UNTUK MERAMALKAN PENJUALAN MOBIL TOYOTA DI INDONESIA

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Statistika
pada
Program Studi Sarjana Departemen Statistika
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember

Oleh:
DESSYANA RATNA WIDYASARI
NRP 06211940005008

Disetujui oleh Pembimbing:
Irhamah, M.Si., Ph.D ( )
NIP. 19780406 200112 2 002

Mengetahui,
Kepala Departemen Statistika
Fakultas Sains dan Analitika Data ITS

Dr. Dra. Kartika Fithriasari, M.Si
NIP. 19691212 199303 002

SURABAYA, AGUSTUS 2021

(*Halaman ini sengaja dikosongkan*)

PENERAPAN MODEL *ARIMA, SUPPORT VECTOR REGRESSION* (SVR), DAN *GENETIC ALGORTIHM – SUPPORT VECTOR REGRESSION* (GA-SVR) UNTUK MERAMALKAN PENJUALAN MOBIL TOYOTA DI INDONESIA

Nama Mahasiswa : Dessyana Ratna Widyasari
NRP : 06211940005008
Departemen : Statistika
Dosen Pembimbing : Irhamah, M.Si., Ph.D

Abstrak

Penjualan mobil Toyota menduduki peringkat teratas di pasar mobil Indonesia dengan share lebih dari 30% selama beberapa tahun terakhir. Hingga tahun 2019, penjualan mobil Toyota cenderung meningkat, namun mulai tahun 2020, penjualan mobil Toyota merosot tajam dibandingkan tahun-tahun sebelumnya. Hal ini merupakan dampak dari pandemi COVID-19 yang menyebar pada kuartal pertama tahun 2020. Data penjualan mobil umumnya bersifat non linier, sehingga peramalan menggunakan ARIMA akan menghasilkan error yang cukup besar. Penggunaan model SVR (Support Vector Regression) dalam peramalan memiliki performansi yang baik dalam mengatasi permasalahan time series dan data non linier. Pada penelitian ini digunakan model ARIMA dan SVR untuk meramalkan penjualan mobil Toyota. Dalam pemilihan parameter SVR, digunakan metode GA agar diperoleh hasil ramalan yang lebih baik. Berdasarkan hasil analisis yang telah dilakukan, model yang menghasilkan prediksi terbaik adalah SVR dengan optimasi algoritma genetika (GA-SVR) karena memiliki nilai RMSE in sample, RMSE out sample, dan SMAPE out sample yang paling kecil dibandingkan model lainnya. Sehingga, metode ini sudah tepat digunakan untuk meramalkan penjualan mobil Toyota.

Kata Kunci : ARIMA, Genetic Algorithm, Peramalan Penjualan, Support Vector Regression (SVR), Toyota.

(*Halaman ini sengaja dikosongkan*)

ARIMA, SUPPORT VECTOR REGRESSION (SVR), AND GENETIC ALGORITHM – SUPPORT VECTOR REGRESSION (GA-SVR) MODEL FOR TOYOTA INDONESIA CAR SALES FORECASTING

Name : **Dessyana Ratna Widyasari**
Student's Number : **06211940005008**
Department : **Statistics**
Supervisor : **Irhamah, M.Si., Ph.D**

Abstract

Toyota has occupied the highest rank regarding car selling within Indonesian car market with percentage of share passes beyond 30% in the past few years. The selling rate had increased gradually until the year of 2019, but it was dramatically declined in 2020. COVID-19 outbreak emergence in the first quartile of 2020 became the pivotal cause of the Toyota's falling rate. Generally, car selling data is non-linear, and thus the employment of ARIMA would meet fair amount of forecasting errors. The usage of SVR (Support Vector Regression) in forecasting contains excellent performance to solve trouble in time series and non-linear data. This research employed ARIMA model and SVR for forecasting Toyota car selling. In terms of SVR parameter selection, GA method was utilized in order to obtain more reasonable result. According to analysis, the best prediction is resulted from SVR model with genetic algorithm optimization (GA-SVR) as it resulted the smallest value in RMSE in sample, RMSE out sample, and SMAPE out sample compared to other models. Hence, this method is appropriate for forecasting Toyota car selling.

Kata Kunci : ***ARIMA, Genetic Algorithm, Support Vector Regression (SVR), Sales Forecasting, Toyota.***

(*Halaman ini sengaja dikosongkan*)

KATA PENGANTAR

Segala puji bagi Allah SWT Tuhan Semesta Alam yang memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul:

"**Penerapan Model ARIMA, *Support Vector Regression* (SVR), dan *Genetic Algortihm – Support Vector Regression* (GA-SVR) Untuk Meramalkan Penjualan Mobil Toyota Di Indonesia"**

Terselesaikannya Tugas Akhir ini tidak lepas dari bantuan, bimbingan dan dukungan berbagai pihak, sehingga penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibu Irhamah, M.Si., P.hD. selaku dosen pembimbing yang senantiasa sabar dalam membimbing dan mengarahkan penulis dari awal hingga tugas akhir ini terselesaikan dengan baik. Beliau juga tak henti-hentinya mendoakan dan memberikan saran kepada penulis selama penulis duduk di bangku perkuliahan.
2. Prof. Drs. Nur Iriawan, M.Ikom, Ph.D dan Ibu Adatul Mukarromah, S.Si., M.Si. selaku dosen penguji yang telah memberikan saran dan masukan sehingga Tugas Akhir ini tersusun dengan baik.
3. Ibu Dr. Dra. Kartika Fithriasari, M.Si selaku Kepala Departemen dan Ibu Dr. Santi Wulan Purnami, S.Si., M.Si selaku Sekretaris Departemen I Statistika FSAD ITS yang membantu secara administratif dalam proses penyusunan tugas akhir.
4. Seluruh dosen Departemen Statistika FSAD ITS yang telah mengajarkan ilmu yang sangat bermanfaat ketika penulis duduk di bangku perkuliahan. Tak lupa juga terima kasih penulis kepada seluruh staf Departemen Statistika FSAD ITS.
5. Kedua orang tua, kedua adik, serta keluarga besar penulis yang selalu mendoakan, memberikan motivasi, dan dukungan kepada penulis dalam menyelesaikan Tugas Akhir.

6. Seluruh pihak yang telah banyak membantu penulis dan tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran diharapkan dari semua pihak untuk tahap pengembangan selanjutnya. Semoga Tugas Akhir ini akan bermanfaat untuk menambah wawasan keilmuan bagi semua pihak.

Surabaya, Juli 2021

Penulis

DAFTAR ISI

HALAMAN JUDUL ..... **i**
LEMBAR PENGESAHAN ..... **v**
Abstrak ..... **vii**
Abstract ..... **ix**
KATA PENGANTAR ..... **xi**
DAFTAR ISI ..... **xiii**
DAFTAR TABEL ..... **xv**
DAFTAR GAMBAR ..... **xvii**
DAFTAR LAMPIRAN ..... **xix**
1. BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Perumusan Masalah 5
1.3 Tujuan 6
1.4 Manfaat 6
1.5 Batasan Masalah 7
2. BAB II TINJAUAN PUSTAKA **9**
2.1 Model ARIMA (*Autoregressive Integrated Moving Average*) 9
2.1.1 Stasioneritas Model ARIMA 10
2.1.2 Identifikasi Model ARIMA 12
2.1.3 Estimasi Parameter 14
2.1.4 Uji Signifikansi Parameter 15
2.1.5 Uji Diagnostik Residual 16
2.2 Deteksi *Outlier* 17
2.3 Uji Linieritas Teräsvirta 18
2.4 Model *Support Vector Regression* (SVR) 19
2.5 Kernel *Radial Basis Function* 22
2.6 Algoritma Genetika (GA) 23

2.7 Kriteria Pemilihan Model Terbaik 26
2.8 Agen Tunggal Pemegang Merek (ATPM) 26
3. BAB III METODOLOGI PENELITIAN 29
3.1 Sumber Data 29
3.2 Variabel Penelitian dan Struktur Data 29
3.3 Langkah Penelitian..... 30
3.4 Diagram Alir 34
4. BAB IV ANALISIS DAN PEMBAHASAN 39
4.1 Karakteristik Data Penjualan Mobil di Indonesia 39
4.2 Pemodelan Penjualan Mobil Toyota Menggunakan ARIMA 44
4.2.1 Identifikasi Pola Penjualan Mobil Toyota 44
4.2.2 Identifikasi Model ARIMA 47
4.2.3 Pemilihan Model Terbaik ARIMA..... 53
4.3 Pemodelan Penjualan Mobil Toyota Menggunakan *Support Vector Regression* 55
4.4 Pemodelan Penjualan Mobil Toyota Menggunakan *Support Vector Regression* dengan Optimasi Algoritma Genetika (GA-SVR)..... 58
4.5 Peramalan Penjualan Toyota Menggunakan Model Terbaik 62
5. BAB V KESIMPULAN DAN SARAN 67
5.1 Kesimpulan 67
5.2 Saran 68
DAFTAR PUSTAKA 69
LAMPIRAN..... 74
BIODATA PENULIS 107

DAFTAR TABEL

Tabel 2.1 Transformasi Box-Cox 12
Tabel 2.2 Identifikasi Model ARIMA..... 14
Tabel 3.1 Struktur Data 29
Tabel 4.1 Deskripsi Penjualan Mobil di Indonesia 2011 hingga 2021..... 42
Tabel 4.2 Deskripsi Penjualan Mobil Toyota di Indonesia 42
Tabel 4.3 Hasil Estimasi Dugaan Model ARIMA 48
Tabel 4.4 Diagnostic Checking pada Residual Model ARIMA . 49
Tabel 4.5 Diagnostic Checking pada Residual Model ARIMA Outlier 51
Tabel 4.6 Pemilihan Model Terbaik ARIMA 54
Tabel 4.7 Input Lag SVR..... 56
Tabel 4.8 Range Parameter..... 57
Tabel 4.9 Parameter Optimum Masing-masing Input SVR 57
Tabel 4.10 Parameter Optimum GA-SVR 61
Tabel 4.11 Pemilihan Model Terbaik untuk Meramalkan Penjualan Toyota 63
Tabel 4.12 Peramalan Penjualan Toyota 12 Bulan Mendatang.. 64

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi ε-Insensitive Loss Function 20
Gambar 2.2 ATPM di Indonesia 27
Gambar 3.1 Diagram Alir Penelitian 34
Gambar 3.2 Diagram Alir Tahapan ARIMA..... 35
Gambar 3.3 Diagram Alir Tahapan SVR 36
Gambar 3.4 Diagram Alir Tahapan GA-SVR 37
Gambar 4.1 Market Share Penjualan Merek Mobil di Indonesia periode 2011 hingga 2020 39
Gambar 4.2 Perbandingan Market Share Merek-merek mobil di Indonesia dibagi dalam Dua Periode 40
Gambar 4.3 Perbandingan Penjualan Toyota dengan Merek Lainnya 2011 hingga 2021 41
Gambar 4.4 Market Share Penjualan Mobil di Indonesia Tahun 2020 43
Gambar 4.5 Komposisi Penjualan Mobil Toyota Tahun 2020 .. 44
Gambar 4.6 Time Series Plot Penjualan Mobil Toyota 45
Gambar 4.7 Transformasi Box-Cox 46
Gambar 4.8 ACF Plot Penjualan Mobil Toyota 46
Gambar 4.9 (a) Time Series Plot Penjualan Mobil Toyota dan **(b)** Plot ACF Hasil Regular Differencing orde 1 (d = 1) 47
Gambar 4.10 Plot PACF data Penjualan Mobil Toyota Setelah Regular Differencing Orde 1 (d = 1) 48
Gambar 4.11 Time Series Plot perbandingan Model ARIMA data In Sample 53
Gambar 4.12 Time Series Plot perbandingan Model ARIMA data Out Sample 54
Gambar 4.13 Time Series Plot Model SVR Terbaik pada Data **(a)** In Sample dan **(b)** Out Sample 58
Gambar 4.14 Ilustrasi Bentuk Kromosom..... 59
Gambar 4.15 Ilustrasi Pindah Silang..... 60

Gambar 4.16 Nilai Fitness Konvergen pada Input f(Y_{t-1}, Y_{t-2}) 60
Gambar 4.17 Time Series Plot Model GA-SVR Terbaik pada Data **(a)** In Sample dan **(b)** Out Sample 62
Gambar 4.18 Perbandingan Model Terbaik dengan Data *Out Sample* 63
Gambar 4.19 Hasil Ramalan Penjualan Toyota 12 Bulan Kedepan 65

DAFTAR LAMPIRAN

Lampiran 1 Data Penjualan Toyota 74
Lampiran 2. Syntax dan Output Uji ADF 76
Lampiran 3. Syntax Model ARIMA 77
Lampiran 4. *Output* Model ARIMA 80
Lampiran 5. Penguraian Model ARIMA 86
Lampiran 6. Komponen *Outlier* Masing-masing Model ARIMA 89
Lampiran 7. *Syntax* Model ARIMA *Outlier* 92
Lampiran 8. *Output* Model ARIMA *Outlier* 94
Lampiran 9. Pengutaian Model ARIMA *Outlier* 97
Lampiran 10. *Syntax* dan *Output* Uji *Teräsvirta* 99
Lampiran 11. *Syntax SVR* 100
Lampiran 12. *Syntax* GA-SVR 101
Lampiran 13. *Output GA* 102
Lampiran 14. Pernyataan Keaslian Data 105

(*Halaman ini sengaja dikosongkan*)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Manusia dituntut untuk bekerja serba cepat dan tepat di era globalisasi ini. Tingkat mobilitas yang tinggi memaksa seseorang untuk menggunakan sarana transportasi guna mendukung aktivitas sehari-hari. Transportasi bukan lagi hal yang mewah, namun telah menjadi kebutuhan primer setiap manusia. Pembangunan jalan dan infrastuktur yang sedang digalakkan pemerintah juga turut mendorong masyarakat untuk memiliki kendaraan pribadi. Sektor transportasi merupakan salah satu sektor yang memberikan andil besar terhadap perekonomian di Indonesia, dimana hingga tahun 2019, sektor perdangangan besar dan eceran; reparasi mobil dan sepeda motor memberikan kontribusi tertinggi yaitu 13% dari total nilai PDRB atas dasar harga konstan. Selain itu, subsektor perdangangan mobil, sepeda motor, dan reparasinya menyumbang sebesar 2,5% dari total nilai PDRB tahun 2019 (BPS, 2019). Menurut laporan dari OICA (*Organisation Internationale des Constructeurs d'Automobiles*), Indonesia merupakan negara dengan penjualan mobil terbanyak keenam se-Asia Pasifik pada tahun 2019. Laporan dari *ASEAN Automotive Federation* juga menunjukkan bahwa Indonesia merupakan negara dengan penjualan mobil tertinggi di ASEAN dengan penjualan melebihi 1 juta unit, atau dengan kata lain, Indonesia menguasai 32,3% pangsa pasar di ASEAN. Berdasarkan data dari Gabungan Industri Kendaraan Bermotor Indonesia (GAIKINDO), penjualan mobil di Indonesia cenderung mengalami peningkatan dari tahun 2011 hingga 2018. Jumlah mobil yang terjual pada tahun 2018 sebanyak 1,152 juta unit atau naik sebesar 7,2% jika dibandingkan tahun 2017.

Agen tunggal pemegang merek atau disingkat ATPM merupakan perusahaan nasional yang ditunjuk oleh perusahaan manufaktur pemilik merek dan berperan untuk melakukan produksi hingga distribusi mobil. Salah satu ATPM terbesar di

Indonesia adalah PT. Toyota Astra Motor yang menaungi merek Toyota. Pasar mobil di Indonesia sebesar lebih dari 60% didominasi oleh 3 *brand* besar yaitu Toyota, Daihatsu, dan Honda. Selama beberapa tahun terakhir, lebih dari sepertiga pangsa pasar mobil di Indonesia dikuasai oleh Toyota. Hingga tahun 2020, *market share* Toyota tetap menguasai peringkat teratas dengan *share* 31,2%, kemudian disusul Daihatsu dengan *share* 17,3%, dan Honda sebesar 13,7%. Ketiga *brand* tersebut selalu bersaing di peringkat teratas *market share* penjualan mobil di Indonesia. Penjualan mobil di Indonesia masih didominasi oleh tipe MPV (*multipurpose vehicle*). Masyarakat memilih mobil tipe MPV karena harganya terjangkau, memiliki fitur yang lengkap, dan kapasitas yang besar. Toyota tetap mempertahankan posisinya di urutan tertatas penjualan mobil tipe MPV. Segmen MPV menyumbang penjualan Toyota sebesar 31,8% berkat kontribusi Avanza-Veloz, Rush, dan Innova (Yolandha, 2021).

Pandemi COVID-19 yang mulai menyebar di Indonesia pada kuartal pertama 2020 memberikan dampak buruk di berbagai sektor salah satunya perekonomian. Menurut laporan dari Badan Pusat Statistik yang dilansir dari Kompas.com, pertumbuhan ekonomi di Indonesia pada kuartal II tahun 2020 sebesar minus 5,32 persen. Pertumbuhan ekonomi pada kuartal I 2020 hanya sebesar 2,97 persen, turun drastis dari 5,02 persen pada periode yang sama di tahun 2019. Ketidakstabilan ekonomi secara langsung berpengaruh terhadap industri otomotif. Dampak pandemi membuat penjualan mobil terus menurun, bahkan hingga akhir tahun 2020, jumlah penjualan mobil nasional hanya sebanyak 532 ribu unit atau turun 48,35% dibandingkan tahun 2019 yang mencapai 1,03 juta unit. Penurunan penjualan yang sangat drastis membuat ATPM merugi karena kejadian pandemi merupakan kejadian yang tidak dapat dikendalikan dan tidak dapat diperkirakan sampai kapan dampaknya terhadap perusahaan. Penurunan penjualan mobil nasional terjadi mulai bulan Maret 2020 dan berada dititik terendah pada bulan Mei 2020 dengan total penjualan hanya 17 ribu unit saja. Dampak ini juga dirasakan oleh

Toyota dimana total penjualan mobil di tahun 2020 hanya 182.655 unit atau turun 44,8% dibanding penjualan pada periode yang sama tahun sebelumnya. Pemerosotan penjualan yang tajam terjadi pada bulan Mei 2020 dimana unit terjual hanya 6.727 unit atau turun 78,8% dibandingkan periode yang sama tahun sebelumnya. Hal ini menjadi sebuah pukulan keras kepada ATPM Toyota karena dampak yang ditimbulkan sangat besar dan relatif sulit untuk mempertahankan keseimbangan *supply* dan *demand* di masa pandemi. Hal ini mendorong perusahaan untuk melakukan *forecasting* penjualan untuk masa mendatang agar kerugian dapat diminimalkan.

Model ARIMA merupakan model yang sering digunakan untuk peramalan, namun model ini hanya bisa menangkap pola linier dan tidak bisa menangkap dengan mudah pola-pola nonlinier (Pai & Lin, 2005). Penelitian oleh Noverianto (2018) menyatakan bahwa data penjualan mobil di Indonesia cenderung memiliki pola nonlinier sehingga peramalan menggunakan metode ARIMA akan menghasilkan *error* yang cukup besar. Metode peramalan yang dapat digunakan pada data non linier contohnya adalah *Artificial Neural Network* (ANN) dan *Support Vector Machine* (SVM). Zhang (2003) melakukan pemodelan beban listrik jangka pendek yang mengandung unsur nonlinier menggunakan *hybrid* ARIMA-ANN dan menghasilkan bahwa peramalan ARIMA memiliki *error* yang cukup besar dan akurasi yang rendah. Model ANN dan SVM memiliki kemampuan baik dalam meramalkan bagian-bagian sensitif seperti nonlinier (Nie dkk., 2012), namun model ANN tidak dapat secara akurat meramalkan bagian-bagian dasar data linier karena fokusnya hanya bagian *fitting* nonlinier dan tidak memiliki kepresisian hasil ramalan yang baik.

Support Vector Machine (SVM) dicetuskan oleh Vapnik di tahun 1995. Metode ini memiliki prinsip meminimalkan *upper bound* dari *generalization error* dan membangkitkan fungsi regresi dengan menerapkan sekumpulan fungsi linier yang berdimensi tinggi. SVM dapat diterapkan pada klasifikasi dan regresi, jika SVM digunakan untuk menyelesaikan permasalahan regresi maka

disebut *Support Vector Regression* (SVR). SVR merupakan model yang sangat efektif untuk mempelajari sampel kecil dan dapat mengatasi permasalahan dimensi data (Liu, dkk., 2019). Keunggulan lain SVR adalah tidak memerlukan asumsi-asumsi khusus dan dapat mengatasi data nonlinier menggunakan trik *kernel*.

Peramalan menggunakan SVR pernah dilakukan oleh Pai & Lin (2005) melakukan peramalan harga saham dengan membandingkan metode ARIMA dan SVR diperoleh bahwa model SVR menghasilkan *error* terkecil. Penelitian yang sama dilakukan oleh Nie dkk. (2012) untuk meramalkan penggunaan listrik di Cina dihasilkan bahwa model SVR menghasilkan MAPE sebesar 4%, sedangkan model ARIMA 4,5%. Maldonado, Gonzales & Crone (2019) memodelkan beban listrik jangka pendek dan menghasilkan bahwa model SVR dapat menggambarkan model non linier secara fleksibel karena adanya peran dari fungsi kernel serta lebih *robust* dari *neural networks*.

Penelitian ini akan menggunakan kernel *Radial Basis Function* (RBF) karena umum digunakan dalam SVR dan menghasilkan *output* yang lebih baik (Thurnhofer-Hemsi dkk., 2020). Terdapat tiga parameter dalam RBF, yaitu C (*cost*), γ (*gamma*), ε (*epsilon*), dimana nilai optimum parameter tersebut dicari menggunakan berbagai metode seperti *grid search* dan Algoritma Genetika (*Genetic Algorithm*). Algoritma Genetika digunakan untuk mengoptimasi parameter dengan ruang lingkup yang lebih besar dan serta lebih efisien dalam megoptimasi jumlah parameter yang lebih banyak (Machairas, Tsangrassoulis, & Axarli, 2014).

Penelitian mengenai SVR dengan optimasi GA pernah dilakukan oleh Suci (2017) yaitu meramalkan curah hujan di Kabupaten Bojonegoro dan menghasilkan bahwa model SVR – *Genetic Algorithm* memberikan hasil *error* yang lebih kecil dibandingkan ARIMA dan SVR. Pai (2006) meramalkan reliabilitas sistem menggunakan SVM dengan optimasi GA dalam memilih parameter dan diperoleh bahwa metode ini lebih baik

daripada ARIMA. Penelitian lain oleh Guo, Li, dan Zhang (2012) yaitu meramalkan harga minyak menggunakan SVM-GA menghasilkan RMSE yang lebih rendah dibandingkan model SVM tanpa optimasi. Harafani dan Wahono (2015) memodelkan peristiwa kebakaran hutan menggunakan beberapa model regresi menunjukkan bahwa SVM-GA memiliki akurasi yang lebih baik dibandingkan model-model regresi lainnya. Penelitian tentang peramalan penjualan mobil menggunakan metode GA-SVR pernah diteliti oleh Yuan (2012) dan menghasilkan bahwa GA-SVR memiliki akurasi yang lebih baik daripada *neural networks* dan metode konvensional lainnya serta tetap memiliki performansi yang baik meskipun terjadi intervensi seperti bencana alam.

Berdasarkan uraian diatas, penelitian ini akan berfokus menggunakan model ARIMA dan SVR dengan menggunakan optimasi *genetic algorithm* (GA) untuk meramalkan penjualan mobil Toyota di Indonesia. Harapannya, hasil penelitian ini dapat memberikan informasi tambahan kepada ATPM Toyota untuk memperkirakan berapa jumlah penjualan mobil yang sesuai dengan kebutuhan pasar di masa mendatang.

1.2 Perumusan Masalah

Toyota menguasai sepertiga pangsa pasar mobil di Indonesia selama beberapa tahun terakhir. Penjualan mobil Toyota cenderung bervariasi dari tahun ke tahun, namun pada tahun 2020, penurunan yang terjadi mencapai 44,8% diakibatkan oleh pandemi COVID-19 yang mulai menyebar di kuartal pertama. Toyota mengalami penurunan yang paling drastis jika dibandingkan dengan merek-merek mobil lainnya. Hal ini membuat ATPM Toyota mengalami kerugian dan perlu menyusun ulang strategi pemasaran baru sesuai dengan kondisi permintaan dimasa pandemi. Peramalan penjualan diperlukan ATPM Toyota untuk memperkirakan berapa unit yang harus dijual agar dapat meminimumkan *cost* akibat kelebihan produksi. Model ARIMA merupakan model yang sering digunakan untuk peramalan, namun model ini hanya mampu menangkap pola-pola linier data saja.

Data penjualan mobil umumnya bersifat nonlinier, sehingga akan dilakukan pemodelan nonlinier salah satunya menggunakan metode SVR (*Support Vector Regression*). Untuk mengatasi permasalahan optimasi parameter pada SVR, digunakan metode GA (Algoritma Genetika) agar diperoleh model yang paling baik dengan *error* yang rendah.

Berdasarkan uraian diatas, akan dilakukan peramalan penjualan mobil Toyota menggunakan metode ARIMA, SVR, dan GA-SVR. Ketiga model tersebut akan dibandingkan menggunakan kriteria RMSE untuk *in sample* dan sMAPE *out sample*. Model terbaik akan digunakan untuk meramalkan penjualan Toyota 12 bulan kedepan.

1.3 Tujuan

Berdasarkan uraian dari rumusan masalah diatas, tujuan yang ingin dicapai pada penelitian ini adalah sebagai berikut.

1. Mendapatkan gambaran tentang karakteristik data penjualan mobil Toyota di Indonesia.
2. Mendapatkan model ARIMA, SVR, dan GA-SVR untuk meramalkan penjualan mobil Toyota.
3. Mendapatkan model terbaik dari tiga metode peramalan yaitu ARIMA, SVR, dan GA-SVR untuk meramalkan penjualan mobil Toyota.
4. Mendapatkan nilai ramalan penjualan mobil Toyota 12 bulan mendatang berdasarkan model terbaik yang terpilih.

1.4 Manfaat

Manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut.

1. Bagi ATPM Toyota sebagai dasar informasi dalam membuat strategi penjualan sesuai permintaan pasar di kala pandemi sehingga dapat memperkirakan berapa jumlah mobil yang harus dijual untuk meminimalkan kerugian.
2. Sebagai informasi bagi pihak-pihak terkait dan dapat dijadikan referensi tambahan untuk penelitian selanjutnya.

3. Sebagai informasi tambahan mengenai penjualan Toyota kepada GAIKINDO sebagai bahan untuk membuat kebijakan produksi mobil demi memenuhi kebutuhan konsumen.

1.5 Batasan Masalah

Penelitian ini dibatasi dengan menggunakan data *retail sales* atau penjualan dari *dealer* ke konsumen yang berasal dari laporan GAIKINDO periode Januari 2011 hingga Februari 2021. Data yang akan dianalisis adalah penjualan mobil merek Toyota dari semua tipe mobil menggunakan metode ARIMA, SVR, dan GA-SVR. Penggunaan *Genetic Algorithm* dimaksudkan untuk optimasi parameter *Support Vector Regression* agar diperoleh nilai RMSE dan sMAPE yang minimum. Beberapa *software* yang digunakan untuk analisis pada penelitian ini adalah Ms. Excel, Minitab, Tableau, SAS On Demand, dan R Studio.

(*Halaman ini sengaja dikosongkan*)

BAB II
TINJAUAN PUSTAKA

2.1 Model ARIMA (*Autoregressive Integrated Moving Average*)

Model AR (*Autoregressive*) dan MA (*Moving Average*) merupakan model yang sudah sering digunakan secara luas dibidang peramalan. Model ini bersifat linier dan hanya dapat digunakan untuk meramalkan data-data stasioner. Dalam penerapan *time series* terutama di bidang ekonomi dan bisnis, sering ditemukan data yang tidak stasioner. Ketidakstasioneran tersebut dapat berupa mean yang tidak konstan, varians, ataupun keduanya. Model ARIMA diperkenalkan pertama kali oleh Box & Jenkins pada tahun 1970 dan merupakan model yang paling populer untuk peramalan. Model ARIMA merupakan model gabungan antara AR dan MA yang telah stasioner melalui proses *differencing*. Model ARIMA memiliki tiga koefisien yaitu *p* sebagai operator AR, *d* yang menujukkan operator *differencing*, dan *q* sebagai operator MA. Berikut ini adalah bentuk umum dari model ARIMA (*p*, *d*, *q*).

$$\phi_p(B)(1-B)^d \dot{Y}_t = \theta_0 + \theta_q(B)a_t, \qquad (2.1)$$

dimana:

θ_0 = Koefisien tren deterministik

$\phi_p(B)$ = Operator AR $(1-\phi_1 B - ... - \phi_p B^p)$

$\theta_q(B)$ = Operator MA $(1-\theta_1 B - ... - \theta_q B^q)$

$(1-B)^d$ = Operator *differencing*

a_t = Komponen *error*

$\dot{Y}_t$ = data *time series* pada waktu ke-*t* dengan $\dot{Y}_t = Y_t - \mu$ (*t* = 1, 2,..., *n*).

Dalam banyak kasus, terutama dalam deret waktu bidang ekonomi dan bisnis, biasanya mengandung fenomena perulangan setelah periode waktu tertentu atau disebut musiman. Jika dianggap

Y_t mengandung variasi seasonal dan *fit* dengan model ARIMA nonseasonal seperti pada persamaan (2.1), maka sangat jelas hasilnya tidak akan memenuhi asumsi *white noise*. Oleh karena itu, dikenal model musiman sebagai berikut.

$$\Phi_P(B^S)(1-B^S)^D \dot{Y}_t = \Theta_Q(B^S)a_t, \qquad (2.2)$$

dimana:

$\Phi_P(B^S)$ = $1-\Phi_1 B^S - \Phi_2 B^{2S} .. - \Phi_P B^{PS}$

$\Theta_Q(B^S)$ = $1-\Theta_1 B^S - \Theta_2 B^{2S} .. - \Theta_Q B^{QS}$.

S = Orde musiman

Gabungan dari persamaan (2.1) dan (2.2) menghasilkan model yang dikenal sebagai model ARIMA musiman multiplikatif Box-Jenkins dengan persamaan

$$\Phi_P(B^S)\phi_p(B)(1-B)^d(1-B^S)^D \dot{Y}_t = \theta_q(B)\Theta_Q(B^S)a_t.$$

(2.3)

Agar lebih mudah, biasanya $\phi_p(B)$ dan $\theta_q(B)$ disebut model AR dan MA reguler, sedangkan $\Phi_P(B^S)$ dan $\Theta_Q(B^S)$ disebut model AR dan MA seasonal. Model (2.3) biasanya dilambangkan dengan ARIMA$(p, d, q)\times(P, D, Q)_s$ (Wei, 2006).

2.1.1 Stasioneritas Model ARIMA

Model ARIMA dikatakan stasioner jika mean dan varians tidak dipengaruhi oleh waktu tertentu. Beberapa indikator untuk mengetahui adanya ketidakstasioneran data dalam mean adalah sebagai berikut.

1. Pola ACF (*Autocorrelation Fuction*) turun secara lambat, bisa bentuk naik atau turun.
2. Mean *level* yang berubah-ubah.
3. Parameter model Y_t, tidak memenuhi syarat kestasioneran. Salah satu metode untuk menguji adanya ketidakstasioneran data adalah menggunakan uji ADF (*Augmented Dickey Fuller*) yang merupakan pengembangan dari uji DF (*Dickey-Fuller*).

Uji ADF menguji hipotesis nol bahwa *unit root* ada dalam sampel deret waktu. Hipotesis alternatif dapat berbeda-beda bergantung pada jenis uji yang digunakan, namun biasanya hipotesis alternatif adalah stasioner dalam tren. Prosedur uji untuk uji ADF adalah mendapatkan estimasi model pada persamaan (2.4) (Gujarati, 2003)

$$\Delta Y_t = \beta_0 + \beta_1 t + \gamma Y_{t-1} + \sum_{i=1}^{p-1} \delta_i \Delta Y_{t-i} + a_t, \qquad (2.4)$$

Dimana β_0 merupakan konstanta, koefisien β_1 dari tren waktu, dan p adalah orde *lag* dari proses *autoregressive*, a adalah *error* murni *white noise* dan $\Delta Y_{t-1} = Y_{t-1} - Y_{t-2}$. Jika $\beta_0 = 0$ dan $\beta_1 = 0$, maka model (2.4) sesuai dengan model *random walk*, dan jika $\beta_1 = 0$, maka model tersebut dikatakan *random walk* dengan *drift.*

Algoritma uji ADF diawali dengan mendapatkan estimasi dari $\hat{\gamma}$ dari persamaan (2.4) kemudian dilanjutkan dengan menguji apakah $\hat{\gamma} = 0$ yang artinya data tidak stasioner dalam mean menggunakan statistik uji sebagai berikut.

$$\text{ADF} = \frac{\hat{\gamma}}{SE(\hat{\gamma})}. \qquad (2.5)$$

Nilai kritis yang digunakan sama dengan uji DF karena uji ADF mengikuti distribusi yang sama dengan statistik DF yaitu t. H_0 ditolak jika nilai ADF > $t_{\alpha/2;\ (n\ -\ p)}$ dengan n adalah banyak pengamatan dan p banyaknya parameter atau *P value* < α.

Menurut Wei (2006), model nonstasioner dalam mean dapat dijadikan stasioner melalui proses *differencing* pada data. dengan kata lain deret $\{Y_t\}$ tidak stasioner tapi hasil dari *differecing* orde ke d untuk semua bilangan $d \geq 1$ adalah stasioner. Sebagai contoh, *differecing* order ke d mengikuti fenomena *white noise* jika:

$$(1-B)^d Y_t = a_t. \qquad (2.6)$$

Jika dimisalkan bahwa $d = 1$ sehingga diperoleh,

$$(1-B)Y_t = a_t, \qquad (2.7)$$

atau

$$Y_t = Y_{t-1} + a_t. \qquad (2.8)$$

Salah satu metode untuk mestasionerkan varians pada data adalah transformasi Box-Cox yang diperkenalkan pada tahun 1964. Transformasi ini digunakan untuk memodifikasi bentuk distribusi dari data sehingga dapat menjadi lebih normal dan stasioner dalam varians (Box & Cox, 1964). Transformasi Box-Cox memiliki bentuk matematis sebagai berikut.

$$T(Y_t) = \frac{Y_t^{\lambda} - 1}{\lambda}, \lambda \neq 0, \qquad (2.9)$$

dimana λ merupakan parameter transformasi. Secara umum, nilai standar untuk λ adalah -2, -1,5, -1, …, 2 digunakan untuk menentukan bentuk transformasi. Pendugaan parameter λ dapat dicari dengan metode kemungkinan maksimum (*Maximum Likelohood Estimation*) dimana nilai λ dipilih berdasarkan jumlah kuadrat sisaan paling minimum. Setiap nilai λ memiliki rumus transformasi yang berbeda.

Tabel 2.1 Transformasi *Box-Cox*

λ	**Transformasi**
-1	$1/Y_t$
-0.5	$1/\sqrt{Y_t}$
0	$\ln(Y_t)$
0.5	$\sqrt{Y_t}$
1	Y_t

2.1.2 Identifikasi Model ARIMA

Identifikasi model ARIMA dilakukan dengan melihat plot ACF (*Autocorellation Function*) dan PACF (*Partial Autocorrelation Function*). ACF menggambarkan korelasi dari Y_t dan Y_{t-k} dari proses yang sama, dipisahkan hanya dengan k lag waktu. Pada proses stasioner $\{Y_t\}$ dengan $E(Y_t) = \mu$ dan Var(Y_t) = ($Y_t - \mu$) = yang konstan serta kovarians Cov(Y_t, Y_{t-k}), dimana

merupakan fungsi dari $|t - s|$, kovarians antara Y_t dan Y_{t-k} dapat dituliskan sebagai berikut (Wei, 2006).

$$\gamma_k = \mathrm{Cov}(Y_t, Y_{t+k}) = E[(Y_t - \mu)(Y_{t+k} - \mu)]. \qquad (2.10)$$

Korelasi antara Y_t dan Y_{t-k} adalah

$$\rho_k = \frac{\mathrm{Cov}(Y_t, Y_{t+k})}{\sqrt{\mathrm{Var}(Y_t)}\sqrt{\mathrm{Var}(Y_{t+k})}} = \frac{\gamma_k}{\gamma_0}. \qquad (2.11)$$

keterangan:

k = order *lag* (k = 0, 1, 2, …, K)

γ_k = *Autocovariance function* pada *lag* ke- k

ρ_k = *Autocorellation Function* pada *lag* ke-k

Y_t = Data aktual pada waktu t.

Sebagai tambahan untuk autokorelasi antara Y_t dan Y_{t-k} dapat pula diidentifikasi korelasi antara Y_t dan Y_{t-k} setelah variabel yang menyebabkan dependensi bersama yaitu Y_{t+1}, Y_{t+2}, …., Y_{t+1} dikeluarkan. Korelasi bersyarat dapat dituliskan sebagai berikut.

$$\mathrm{Corr}(Y_t, Y_{t+k} \mid Y_{t+1}, Y_{t+2}, ..., Y_{t+k-1}). \qquad (2.12)$$

Persamaan (2.11) biasanya disebut dengan *Partial Autocorrelation* dalam analisis deret waktu. Selain ACF, PACF antara Y_t dan Y_{t-k} dapat dituliskan sebagai berikut.

$$P_k = \frac{\mathrm{Cov}[(Y_t - \hat{Y}_t), (Y_{t+k} - \hat{Y}_{t+k})}{\sqrt{\mathrm{var}(Y_t - \hat{Y}_t)}\sqrt{\mathrm{var}(Y_{t+k} - \hat{Y}_{t+k})}}, \qquad (2.13)$$

dimana:

P_k = *Partial Autocorrelation Function*

$\hat{Y}_t$ = Hasil ramalan pada waktu t.

Identifikasi model ARIMA dapat dijelaskan pada Tabel 2.2 sebagai berikut (Cryer & Chan, 2008).

Tabel 2.2 Identifikasi Model ARIMA

	ACF	PACF
AR(*p*)	*Tails off*	*Cuts off* setelah *lag p*
MA(*q*)	*Cuts off* setelah *lag q*	*Tails off*
ARMA(*p*, *q*)	*Tails off*	*Tails off*

2.1.3 Estimasi Parameter

Terdapat beberapa cara untuk menaksir parameter model ARIMA yaitu metode *moment*, metode *conditional least squares*, metode *maximum likelihood estimation*, dan metode *nonlinear estimation*. Metode estimasi yang digunakan pada penelitian ini adalah *conditional least square* (CLS) karena metode ini paling sering digunakan. Metode CLS memiliki prinsip meminimumkan *error* kuadrat atau SSE (*sum square error*).

Dimisalkan jika ada model AR(1) dengan persamaan $Y_t - \mu = \phi(Y_t - \mu) + a_t$, dimana model ini dapat dilihat sebagai model regresi dengan variabel prediktor Y_{t-1} dan respon variabel Y_t. Metode *least square* meminimumkan SSE dengan cara *differencing*. Berikut ini adalah nilai *error* dari hasil estimasi AR(1) (Cryer & Chan, 2008).

$$a_t = (Y_t - \mu) - \phi(Y_{t-1} - \mu). \qquad (2.14)$$

Karena $Y_1, Y_2, \ldots, Y_n$ sudah diobservasi, maka penjumlahan dapat dilakukan penjumlahan dari $t = 2$ hingga $t = n$. Nilai SSE dapat dituliskan sebagai berikut.

$$S(\phi, \mu) = \sum_{t=2}^{n} a_t^{\,2} = \sum_{t=2}^{n} \left[(Y_t - \mu) - \phi(Y_{t-1} - \mu)\right]^2. \qquad (2.15)$$

Untuk meminimumkan *error* kuadrat, persamaaan (2.15) dilakukan dengan menurunkan terhadap μ dan ϕ. Berikut ini adalah hasil penurunan terhadap μ.

$$\frac{\partial S(\phi, \mu)}{\partial \mu} = \sum_{t=2}^{n} 2\left[(Y_t - \mu) - \phi(Y_{t-1} - \mu)\right](-1 + \phi) = 0. \qquad (2.16)$$

Sehingga diperoleh estimasi parameter untuk μ dari model AR(1) yang dituliskan sebagai berikut.

$$\hat{\mu} = \frac{1}{(n-1)(1-\phi)}\left[\sum_{t=2}^{n} Y_t - \phi\sum_{t=2}^{n} Y_{t-1}\right]. \qquad (2.17)$$

Jika n sangat besar, maka hasil estimasi pada persamaan (2.17) dapat dituliskan sebagai berikut.

$$\frac{1}{(n-1)}\sum_{t=2}^{n} Y_t \approx \frac{1}{(n-1)}\sum_{t=2}^{n} Y_{t-1} \approx \bar{Y}, \qquad (2.18)$$

sehingga,

$$\hat{\mu} = \frac{1}{1-\phi}(1-\phi)\bar{Y} = \bar{Y}. \qquad (2.19)$$

Menggunakan cara yang sama, dengan menurunkan persamaan (2.15) terhadap ϕ diperoleh hasil sebagai berikut.

$$\frac{\partial S}{\partial \phi} = \sum_{t=2}^{n} 2\left[(Y_t - \bar{Y}) - \phi(Y_{t-1} - \bar{Y})\right](Y_{t-1} - \bar{Y}) = 0. \qquad (2.20)$$

Akan diperoleh hasil estimasi ϕ sebagai berikut.

$$\hat{\phi} = \frac{\sum_{t=2}^{n}(Y_t - \bar{Y})(Y_{t-1} - \bar{Y})}{\sum_{t=2}^{n}\left(Y_{t-1} - \bar{Y}\right)^2}. \qquad (2.21)$$

2.1.4 Uji Signifikansi Parameter

Pengujian signifikansi parameter digunakan untuk menguji apakah parameter pada model ARIMA sudah signifikan. Sebagai ilustrasi, jika dimisalkan terdapat model AR(1), maka hipotesis nol pada uji signifikansi parameter adalah H_0: $\phi = 0$. Jika hipotesis nol gagal ditolak, menandakan bahwa tidak ada hubungan linier antara variabel prediktor (Y_{t-k}) dan respon (Y_t) atau sebaliknya. Jika H_0: $\phi = 0$ ditolak atau dengan kata lain $\phi \neq 0$, maka variabel prediktor menjelaskan variabilitas dari variabel respon. Untuk menguji hipotesis ini, digunakan statistik uji sebagai berikut.

$$t = \frac{\hat{\phi}}{SE(\hat{\phi})}. \qquad (2.22)$$

$\hat{\phi}$ merupakan estimasi dari ϕ pada model AR(1). H_0 ditolak jika $|t| > t_{\alpha/2;\,(n-p)}$ dengan n adalah banyak pengamatan dan p banyaknya parameter atau *P value* < α yang artinya bahwa parameter signifikan terhadap model (Montgomery, Peck, & Vining, 2012).

2.1.5 Uji Diagnostik Residual

Setelah tahap pemodelan ARIMA, dilakukan uji diagnostik residual meliputi uji asumsi *white noise* dan uji asumsi distribusi normal. Model yang memenuhi kedua asumsi tersebut merupakan model terbaik dan dapat digunakan untuk proses peramalan. Uji distribusi normal menggunakan uji Kolomgorov-Smirnov.

1. Uji Distribusi Normal

   Uji distribusi normal yang digunakan adalah uji Kolmogorov-Smirnov (KS). Uji ini digunakan untuk menguji apakah sebuah variabel $F_n(x)$ mengikuti suatu distribusi yang telah ditentukan dari sebuah populasi $F_0(x)$. Distribusi yang telah ditentukan tidak selalu normal, namun dalam penelitian ini, uji KS berfokus pada distribusi normal. Hipotesis nol menguji apakah nilai $D_{max} = 0$ dengan menggunakan statistik uji sebagai berikut (Daniel, 1989).

$$D_n = \underset{x}{Sup}\left|F_n(x) - F_0(x)\right|, \qquad (2.23)$$

   dimana:

   $F_n(x)$ = Fungsi distribusi peluang kumulatif pengamatan dari data yang diobservasi

   $F_0(x)$ = Fungsi distibusi peluang kumulatif dugaan.

   Hipotesis nol ditolak jika jika $D_n > D_{(1-\alpha,\,n)}$ atau *P-Value* < α.

2. Uji Asumsi *White Noise Ljung Box*

   Uji Ljung-Box merupakan uji statistik untuk memeriksa apakah terdapat kelompok autokorelasi dari sebuah deret waktu yang tidak sama dengan nol. Hipotesis nol yang diuji adalah $\rho_1 = \rho_2 = \ldots = \rho_k$ atau tidak ada korelasi serial pada residual, sedangkan hipotesis alternatif adalah minimal satu

pasang residual memiliki korelasi serial ($\rho_k = 0$). Statistik uji Q dituliskan sebagai berikut (Ljung & Box, 1978)

$$Q = n(n+2)\sum_{k=1}^{K}\frac{r_k(\hat{a})^2}{(n-k)}. \qquad (2.24)$$

keterangan:

$r_k(\hat{a})$ = autokorelasi residual a pada lag ke-k

Dibawah hipotesis nol, statistik Q mengikuti distribusi $\chi^2_{(K-p)}$ dimana p merupakan banyak parameter pada model. Pada taraf signifikan α, H_0 ditolak jika $Q > \chi^2_{(K-p)}$ atau *p-value* < α.

2.2 Deteksi *Outlier*

Data *time series* terkadang memiliki *outlier* yang diakibatkan peristiwa-peristiwa yang tak teduga misalnya gejolak ekonomi, kebijakan pemerintah, libur nasional, pemilihan presiden, wabah penyakit dan lain-lain. Pengaruh dari peristiwa tersebut menyebabkan penurunan atau kenaikan nilai yang cukup tajam sehingga menyebabkan data menjadi *outlier*. Hal tersebut menjadi masalah dalam analisis deret waktu sehingga pengaruh *outlier* harus dihilangkan. Menurut Wei (2006), terdapat dua model *outlier* yang biasa dikenal sebagai *outlier additive* dan *innovational*. Berikut ini adalah bentuk *outlier additive*.

$$Y_t = \begin{cases} X_t & ,t \neq T \\ X_t + \omega, & t = T \end{cases} \qquad (2.25)$$

$$= X_t + \omega I_t^{(T)} \qquad (2.26)$$

$$= \frac{\theta(B)}{\phi(B)}a_t + \omega I_t^{(T)}. \qquad (2.27)$$

dimana:

$$I_t^{(T)} = \begin{cases} 1, t = T \\ 0, t \neq T \end{cases}. \qquad (2.28)$$

keterangan:

Y_t = Data aktual pada waktu t

X_t = Deret tanpa *outlier*
ω = Komponen *outlier*
I_t = Indikator variabel yang menjelaskan ada atau tidaknya *outlier* pada waktu ke-*t*.

Selanjutnya adalah model *innovational outlier* yang didefinisikan sebagai berikut.

$$Y_t = X_t + \frac{\theta(B)}{\phi(B)}\omega I_t^{\ (T)} \tag{2.29}$$

$$= \frac{\theta(B)}{\phi(B)}(a_t + \omega I_t^{\ (T)}). \tag{2.30}$$

Berdasarkan model yang telah didefinisikan, dapat diketahui perbedaan dari *additive outlier* dan *innovational outlier*. *Additive outlier* adalah tipe *outlier* yang hanya mempengaruhi observasi ke *T* yaitu Y_T, sedangkan *innovational outlier* adalah tipe *outlier* yang mempengaruhi semua observasi Y_T, Y_{T+1}, … diluar waktu *T* melalui sistem yang dijelaskan oleh $\theta(B)/\phi(B)$. Model umum *outlier* dengan *k outlier* adalah sebagai berikut.

$$Y_t = \sum_{j=1}^{k} \omega_j v_j(B) I_t^{\ (T)} + X_t. \tag{2.31}$$

Selain kedua jenis *outlier* diatas, terdapat pula *level-shift outlier* dimana semua pengamatan muncul setelah *outlier* pindah ke *level* baru. Berbeda dengan *outlier* aditif, *level-shift outlier* mempengaruhi banyak pengamatan dan memiliki efek yang permanen (Tsay, 1988).

2.3 Uji Linieritas Teräsvirta

Salah satu uji yang sering digunakan untuk mendeteksi adanya pola nonlinier dalam data adalah uji Teräsvirta yang perkenalkan oleh Teräsvirta, Lin, & Granger pada tahun 1993. Uji ini termasuk dalam uji kelompok *Lagrange Multiplier* yang dikembangkan dari model *neural network*. Uji Teräsvirta dilakukan dengan cara regresi polinomial dan menguji suku-suku polinomialnya. Jika dimisalkan ada dua variabel yaitu Y_{t-1} dan Y_{t-2}, maka model yang diuji adalah sebagai berikut.

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \delta_1 Y_{t-1}{}^2 + \delta_2 Y_{t-2}{}^2 + \delta_3 Y_{t-1} Y_{t-2} + \delta_4 Y_{t-1}{}^3 + \delta_5 Y_{t-2}{}^3 + \delta_6 Y_{t-1}{}^2 Y_{t-2} + \delta_7 Y_{t-1} Y_{t-2}{}^2. \quad (2.32)$$

Lalu menguji hipotesis nol bahwa parameter pada regresi polinomial (δ) bersifat linier ($\delta_1 = \delta_2 = \delta_3 = \delta_4 = \delta_5 = \delta_6$), sedangkan hipotesis alternatif adalah salah satu parameter tidak linier. Statistik uji yang digunakan adalah nilai F dari hasil pemodelan regresi polinomial sebagai berikut.

$$F = \frac{MSR}{MSE}. \quad (2.33)$$

H_0 ditolak jika nilai $F > F_{(k,\ n-k-1)}$ dengan k adalah banyaknya variabel bebas dan P *value* < α.

2.4 Model *Support Vector Regression* (SVR)

Support Vector Machine pertama kali diperkenalkan oleh Vapnik di tahun 1995. Model ini termasuk dikategori *supervised model* dan dapat digunakan untuk klasifikasi dan regresi (Muller & Guido, 2018). Model SVM memberikan dampak yang besar pada penelitian karena kemampuannya untuk menyelesaikan masalah nonlinier dan *dataset* berdimensi tinggi (Vapnik, 1998). SVM dapat diperluas untuk melakukan pendekatan fungsi regresi dengan menerapkan beberapa fungsi linier yang berdimensi tinggi yang disebut *Support Vector Regression* (SVR).

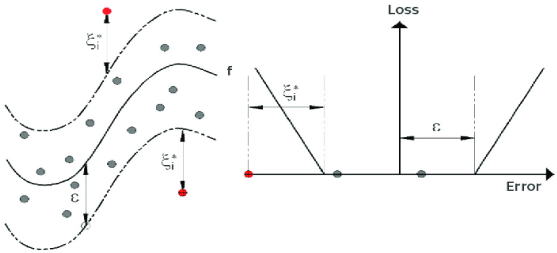
SVR menggunakan konsep *stuructural risk minimization* atau estimasi suatu fungsi dengan meminimalkan batas *generalization error* sehingga dapat mengatasi *overfitting*. Model SVR memberikan fleksibilitas untuk mendefinisikan berapa banyak *error* yang dapat diterima dalam model dan dapat mencari garis atau *hyperplane* yang cocok untuk *fitting* data (Sharp, 2020).

Jika terdapat $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)\} \subset X \times \mathbb{R}$, dimana X melambangkan ruang dari pola input seperti $X = \mathbb{R}^d$. Konsep dasar SVR adalah menentukan fungsi $f(x)$ yang memiliki deviasi ε terbanyak dari target aktual y_i sehingga menghasilkan prediksi yang mendekati nilai sebenarnya (Smola & Scholkopf, 2004). Berikut ini adalah deskripsi dari fungsi regresi SVR.

$$f(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x}) + b, \text{ dengan } \mathbf{w} \in \mathrm{X} \text{ dan } b \in \mathbb{R}. \quad (2.34)$$

Dimana **w** merupakan vektor pembobot yang akan diatur, *b* merupakan koefisien, dan $\varphi(\mathbf{x})$ merupakan fungsi pemeta nonlinier. Tujuan dari SVR ialah mencari fungsi $f(x)$ yang memiliki deviasi ε terbanyak dari target y_i untuk semua data *training* dan membuat kurva persamaan (2.34) menjadi selandai mungkin.

Dalam SVR, dikenal *ε-insensitive loss function* untuk mengukur resiko empiris atau selisih target dengan hasil estimasi. Oleh karena itu, parameter ε perlu diatur sebagai batas toleransi kesalahan hasil estimasi. Berikut merupakan ilustrasi dari *ε-insensitive loss function*.



Gambar 2.1 Ilustrasi *ε-Insensitive Loss Function*

$$L_\varepsilon(y_i, f(x_i)) = \begin{cases} |y_i - f(x_i)| - \varepsilon, & |y_i - f(x_i)| \geq \varepsilon \\ 0 & \text{lainnya} \end{cases} \qquad (2.35)$$

Gambar 2.1 mengilustrasikan bahwa terdapat titik-titik yang berada diluar daerah yang diarsir akan berkontribusi pada *cost*. Hal ini akan menjadi permasalahan pada klasifikasi, sehingga digunakan variabel *slack* (ξ_i, ξ_i*) untuk merepresentasikan jarak dari nilai aktual ke nilai batas yang sesuai dari *ε-tube*. Variabel *slack* merupakan variabel yang yang ditambahkan ke model kendala matematis untuk mengkonversikan pertidaksamaan (≤) menjadi persamaan (=).

Jika dibadingkan dengan OLS, tujuan SVR adalah meminimalkan *norm Euclidean* dari **w** bukan *error* kuadrat seperti OLS. SVR mencari *hyperplane* optimum yang memiliki batasan

maksimum antara sampel positif dan negatif terdekat yang dapat dituliskan sebagai berikut.

$$\text{minimumkan} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*), \tag{2.36}$$

$$\text{Dengan batasan} = \begin{cases} \mathrm{y}_i - f(\mathbf{x}_i) - \xi_i \le \varepsilon \\ f(\mathbf{x}_i) - \mathrm{y}_i - \xi_i^* \le \varepsilon \end{cases}, \tag{2.37}$$

dimana i = 1,2,3, … , n dan ξ_i, ξ_i^* > 0.

ξ_i , ξ_i * merupakan variabel *slack* yang dapat dilihat pada Gambar 2.1. Parameter C mengontrol pertukaran antara keempirisan dan kesalahan generalisasi (Schölkopf, Smola, & Bach, 2002). Parameter lain yang harus diatur secara presisi adalah ε. Solusi untuk persamaan (2.36) dan (2.37) dapat diselesaikan secara mudah menggunakan formula dual. Metode dualisasi standar yang digunakan adalah Langrange *multipliers* yang dideskripsikan oleh Fletcher (1989). Fungsi Langrange untuk mengoptimasi persamaan (2.38) adalah sebagai berikut (Guo, dkk., 2014).

$$L(\mathbf{w}, b, \xi_i, \beta_\mathrm{i}) = \frac{\|\mathbf{w}\|^2}{2} + C\left(\sum_{i=1}^{n}(\xi_i + \xi_i^*)\right) - \sum_{i=1}^{n}\beta_i[\mathbf{w}^T\varphi(\mathbf{x}_i) + f(\mathbf{x}_i) + \xi_i - \mathrm{y}_i], \tag{2.38}$$

Dimana β_i merupakan koefisien Lagrange dengan $0 \le \beta_i \le C$ dan $0 \le \beta_i^* \le C$ (Ma, dkk., 2019).

Persamaan turunan parsial dari setiap variabel dapat didefinisikan sebagai berikut.

$$\frac{\partial L}{\partial \mathbf{w}} = 0, \quad \frac{\partial L}{\partial \mathrm{y}_i} = 0, \quad \frac{\partial L}{\partial \xi_i} = 0, \quad \frac{\partial L}{\partial \beta_i} = 0. \tag{2.39}$$

Berdasarkan dari kondisi Mercer, fungsi kernel dapat didefinisikan sebagai berikut (Zhu, Lin, & Zhang, 2005).

$$K(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i)^T f(\mathbf{x}_j), \tag{2.40}$$

dan kondisi pembatas dari koefisien Lagrange didefinisikan sebagai

$$\sum_{i=1}^{n}(\beta_i - \beta_i^*) = 0, \tag{2.41}$$

dengan vektor bobot optimal *hyperplane* yang diharapkan adalah

$$\mathbf{w}^* = \sum_{i=1}^{n}(\beta_i - \beta_i^*) f(\mathbf{x}) = 0. \tag{2.42}$$

Fungsi regresi dari *sample space* dapat dituliskan sebagai berikut.

$$f(\mathbf{x}, \beta_i, \beta_i^*) = \sum_{i=1}^{n}(\beta_i - \beta_i^*) K(\mathbf{x}_i, \mathbf{x}_j) + b. \tag{2.43}$$

Penggunaan kernel memungkinkan untuk melakukan perhitungan SVR tanpa secara eksplisit memetakan kedalam *feature space*. Pemilihan fungsi kernel untuk suatu tipe data masih menjadi masalah untuk diselesaikan (Amari & Wu, 1999). Jika suatu fungsi memenuhi kondisi Mercer, maka penggunaan kernel harus dipertimbangkan (Vapnik, 1995). Pada penelitian ini akan digunakan kernel *radial basis function.*

2.5 Kernel *Radial Basis Function*

Pada model SVR dikenal istilah *kernel* yang merupakan suatu fungsi untuk memetakan data berdimensi kecil ke dimensi yang lebih besar. Terdapat beberapa *kernel* yang umum digunakan dalam SVR seperti *linear kernel*, *polynomial kernel*, *radial basis function*, dan *sigmoid* (Hsu, Chang, & Lin, 2016). Nilai *kernel* sama dengan *inner product* dari dua vektor $\mathbf{x}_i$ dan $\mathbf{x}_j$ dalam ruang fitur $\phi(\mathbf{x}_i)$ dan $\phi(\mathbf{x}_j)$ seperti $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) * \phi(\mathbf{x}_j)$. Penelitian ini berfokus menggunakan *kernel radial basis function* (RBF) atau dikenal sebagai *Gaussian Kernel* karena performanya yang cukup baik (Ali & A.Smith, 2006). *Kernel* RBF dapat didefinisikan sebagai berikut.

$$\exp(\gamma - \|\mathbf{x}_i - \mathbf{x}_j\|^2). \tag{2.44}$$

Dimana $\gamma > 0$ merupakan parameter yang mengontrol panjang *kernel* (Schölkopf, Smola, & Bach, 2002).

Support Vector Machine sangat sensitif dengan parameter (Wang dkk., 2007), Terdapat 3 parameter utama yang menyeimbagkan antara *training error* dan *generalization*. Berikut

ini adalah penjelasan masing-masing parameter (Lanlan, dkk., 2015).

a. Faktor *C* atau *Cost*, merupakan mendeskripsikan dejarat dari *punishment error*. Jika nilai *C* semakin tinggi, maka *fitting* semakin tinggi sehingga *training error* dan *test error* semakin rendah. Namun jika *C* terlalu besar, maka akan terjadi fenomena *overfitting* dan akan meningkatkan *error* prediksi.
b. Parameter ε, jika semakin besar maka akurasi *fitting* akan semakin kecil, jumlah *support vector* akan semakin rendah, dan kompleksitas akurasi dari model akan berkurang.
c. Parameter γ, jika semakin kecil maka *fitting radial basis function* akan semakin buruk, jika semakin besar maka memiliki performa *fitting* yang baik namun akan menyebabkan kemampuan generalisasi yang buruk.

Dalam mencari parameter terbaik, dapat digunakan metode *grid search* atau mencoba pasangan nilai setiap parameter. Pada umumnya, *range* parameter *C* yang sering digunakan adalah $C = \{10^{-5}, 10^{-4}, 10^{-3}, \ldots, 10^{4}, 10^{5}\}$ (Ring & Eskofier, 2016) atau $C = \{2^{-5}, 2^{-3}, \ldots, 2^{13}, 2^{15}\}$; parameter $\gamma = \{2^{-15}, 2^{-13}, \ldots, 2^{5}, 2^{3}\}$ (Hsu, Chang, & Lin, 2016), dan parameter epsilon $\varepsilon = \{0{,}01; 0{,}1; 0{,}5; 1\}$ (Maldonado, Gonzales, & Crone, 2019)

2.6 Algoritma Genetika (GA)

Optimasi parameter pada SVR dilakukan dengan banyak cara, salah satunya adalah manual *grid search* yang paling banyak digunakan. Namun, pendekatan *grid search* akan semakin rumit jika *search space* meningkat. Selain itu, pendekatan ini memakan waktu yang cukup lama dan tidak selalu mendapatkan *hyperparameter* terbaik untuk model. Pendekatan lainnya yang dikenal untuk mengatasi permasalahan ini adalah *genetic algorithm* (GA). GA merupakan salah satu algoritma optimasi yang diajukan oleh Holland tahun 1975 dimana ternspirasi dari konsep evolusi biologi yaitu Teori Darwin (Tao, dkk., 2020).

Algoritma Genetika secara random membangkitkan beberapa set dari *initial solution*, kemudian menjadi grup, lalu

mulai proses pencarian. Tiap individu dalam populasi merupakan solusi dari permasalahan yang disebut kromosom. Kromosom tersebut dalam iterasi selanjutnya akan berkembang disebut genetika. Algoritma genetika yang paling utama adalah melewati proses *selection, crossover,* dan *mutation* untuk membagkitkan kromosom generasi selanjutnya yang disebut *offspring* (Lanlan dkk., 2015). *Offspring* menggantikan populasi lama dan membentuk populasi baru di generasi selanjutnya dengan empat operator sebagai berikut.

a. Operator *Selection* digunakan untuk menyeleksi kromosom yang baik untuk bereproduksi. Berdasarkan nilai *fitness function*, kromosom yang memiliki nilai *fitness* lebih tinggi akan menghasilkan *offspring* pada generasi selanjutnya dengan metode *roulette wheel* untuk menentukan apakah sebuah kromosom dapat *survive* ke generasi selanjutnya.

b. Operator *Crossover* digunakan secara random untuk menukar (pindah silang) gen diantara dua kromosom. Jika $S_1 = \{s_{11}, s_{12}, \ldots, s_{1n}\}$, $S_2 = \{s_{21}, s_{22}, \ldots, s_{2n}\}$, merupakan dua kromosom, pilih sebuah bilangan random $0 \leq r \leq n$, maka S_3, S_4 merupakan *offspring* dari *crossover* (S_1, S_2),

$$S_3 = \{s_i \mid \text{jika} \quad i \leq r, s_i \in S_1, \text{lainnya} \quad s_i \in S_2\}$$

$$S_4 = \{s_i \mid \text{jika} \quad i \leq r, s_i \in S_2, \text{lainnya} \quad s_i \in S_1\}.$$

c. Operator *mutation* mengikuti *crossover* untuk menentukan apakah sebuah kromosom dapat bermutasi pada generasi selanjutnya. Jika dimisalkan ada kromosom $S_1 = \{s_{11}, s_{12}, \ldots, s_{1n}\}$, pilih bilangan random random $0 \leq r \leq n$, maka S_3 merupakan mutasi dari S_1, $S_3 = \{ s_i \mid$ jika $i \neq r$, maka $s_i = s_{1,i}$,lainnya s_i = random $(s_{1i})\}$.

d. Operator *elitism* digunakan untuk menentukan individu-individu yang pantas untuk bertahan dalam generasi yang akan datang karena *offspring* yang dihasilkan identik dengan *parent.*

Kromosom yang baik atau buruk diukur menggunakan *fitness value*. Sebuah *fitness function* digunakan untuk memeriksa

performansi dari setiap kromosom untuk mencari nilai optimum dari parameter SVR. Terdapat beberapa cara untuk mengevaluasi akurasi dari hasil prediksi permasalahan *time series* seperti RMSE dan MAPE (Yuan, 2012). Pada penelitian ini, nilai *fitness* akan dihitung menggunakan sMAPE. Jika nilai sMAPE semakin kecil, maka model ramalan akan semakin baik. Langkah-langkah dasar dari GA untuk mengoptimasi parameter SVR adalah sebagai berikut:

a. Menentukan parameter C, ε, dan γ bisa dalam bentuk *range* serta membangkitkan *initial population*.
b. Hitung *mean square error function* untuk menentukan nilai *fitness*.
c. Menentukan individu terbaik dalam populasi ketika nilai *fitness* memenuhi kondisi berhenti, kondisi berhenti secara umum ketika sudah mencapai *error* yang sesuai spesifikasi.
d. Jika belum mencapai kondisi berhenti, proses seleksi, *crossover*, dan *mutation* akan memproduksi populasi baru, dan iterasi berulang mulai langkah (b).
e. Mengulangi langkah (b) sampai (d) hingga *training* berakhir
f. Diperoleh hasil parameter yang optimum.

Pengaturan parameter pada *genetic algorithm* mengikuti prinsip-prinsip sebagai berikut.

a. Jika jumlah populasi semakin besar, akan memperbesar terjadinya fenomena konvergensi prematur sehingga membuat evaluasi *fitness* naik tajam, dan kecepatan konvergensi secara signifikan berkurang. Pada penelitian ini, akan dibatasi menggunakan 100 populasi.
b. Peluang *crossover* (P_c) yang semakin rendah memungkinkan terjadinya pemblokiran pencarian dan merusak model asli. Pada penelitian ini, akan digunakan P_c sebesar 0,8.
c. Peluang mutasi (P_m) yang semakin kecil akan membuat beberapa gen kehilangan informasi secara *premature* tidak dapat dipulihkan, dan peluang mutasi yang terlalu tinggi akan menjadikan pencarian semakin random. Pada penelitian ini akan digunakan adalah 0,01.

2.7 Kriteria Pemilihan Model Terbaik

Model terbaik dipilih berdasarkan nilai *error* terkecil yang diukur menggunakan beberapa kriteria seperti RMSE (*Root mean square error*) untuk *in sample* dan sMAPE (*Symmetric Mean Absolute Percentage Error*) untuk *out sample*. Jika kedua nilai tersebut semakin kecil, maka model semakin baik. Berikut ini adalah rumus untuk menghitung RMSE dan sMAPE (Pai & Lin, 2005).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(Y_t - \hat{Y}_t\right)^2}. \tag{2.45}$$

$$\text{sMAPE} = \frac{1}{N}\sum_{l=1}^{N}\left|\frac{(Y_l - \hat{Y}_l)}{(Y_l + \hat{Y}_l)/2}\right| \times 100. \tag{2.46}$$

keterangan:

$\hat{Y}_t$ = Data ramalan *in sample* pada waktu *t*

Y_t = Data aktual *in sample* pada waktu ke *t*

n = Jumlah pengamatan *in sample*

$\hat{Y}_l$ = Data ramalan *out sample* pada waktu *t*

Y_l = Data aktual *out sample* pada waktu ke *t*

N = jumlah pengamatan *out sample.*

2.8 Agen Tunggal Pemegang Merek (ATPM)

Agen Tunggal Pemegang Merek adalah perorangan atau badan usaha yang ditunjuk oleh pabrik pemilik merek tertentu untuk melakukan penjualan partai besar atas barang pabrik tersebut (Anonim, 2013). Ketentuan mengenai pengaturan ATPM saat ini diatur pada Peraturan Kementerian Perdagangan Nomor 11/M-DAG/PER/3/2006 tentang Ketentuan Dan Tata Cara Penerbitan Surat Tanda Pendaftaran Agen atau Distributor Barang Dan/Atau Jasa, Kitab Undang-Undang Hukum Perdata, dan Ketentuan Undang-undang tentang hukum persaingan di Indonesia yang mempunyai keterkaitan dengan Agen Tunggal Pemegang Merek (ATPM) (A., Surya, dkk., 2013).

Gambar 2.2 ATPM di Indonesia

Dalam konteks otomotif, ATPM berperan untuk memproduksi, mengimpor, memasarkan, mendistribusikan, dan melayani *aftersales.* Beberapa contoh ATPM di Indonesia adalah PT. Toyota Astra Motor yang menaungi merek Toyota, PT. Astra Daihatsu Motor yang menaungi merek Daihatsu, dan PT. Honda Prospect Motor yang menaungi merek Honda.

Perwakilan tiap industri pada ATPM tergabung dalam organisasi bernama GAIKINDO (Gabungan Industri Kendaraan Bermotor Indonesia). Tiap tahunnya GAIKINDO merilis data penjualan dari ATPM di Indonesia. Terdapat dua jenis penjualan yaitu *retail sales* dan *wholesales*. *Retail sales* merupakan penjualan mobil dari diler ke konsumen sedangkan *wholesales* merupakan penjualan mobil dari gudang menuju diler. Data *retail* lebih relevan digunakan karena menggambarkan penjualan sesungguhnya.

(*Halaman ini sengaja dikosongkan*)

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan pada penelitian ini adalah data sekunder yaitu data penjualan mobil di Indonesia yang diunduh dari laman resmi GAIKINDO (gaikindo.or.id). Penelitian ini menggunakan data bulanan *retail sales* Toyota pada periode Januari 2011 hingga Februari 2021 sebanyak 122 data. Data dibagi menjadi dua bagian yaitu *training* dan *testing*, dimana periode data *training* yaitu Januari 2011 hingga Juni 2020 sebanyak 114 data, sedangkan periode data *testing* yaitu Juli 2020 hingga Februari 2021 sebanyak 8 data.

3.2 Variabel Penelitian dan Struktur Data

Variabel yang digunakan adalah penjualan mobil Toyota (Y_t) dalam satuan unit. Tabel 3.1 menunjukkan struktur data pada penelitian ini.

Tabel 3.1 Struktur Data

Tahun	Bulan	Toyota (Y_t)	Keterangan
2011	Januari	Y_1	*In sample* (*Training*)
	Februari	Y_2	
	Maret	Y_3	
	⋮	⋮	
	Desember	Y_{12}	
2012	Januari	Y_{13}	
	Februari	Y_{14}	
	Maret	Y_{15}	
	⋮	⋮	
	Desember	Y_{24}	
⋮	⋮	⋮	

Tabel 3.1 Struktur Data (*Lanjutan*)

Tahun	Bulan	Toyota (Y_t)	Keterangan
2019	Januari	Y_{97}	*In sample* (*Training*)
	Februari	Y_{98}	
	Maret	Y_{99}	
	⋮	⋮	
	Desember	Y_{108}	
2020	Januari	Y_{109}	
	⋮	⋮	
	Juni	Y_{114}	
	Juli	Y_{115}	*Out sample* (*Testing*)
	⋮	⋮	
	Desember	Y_{120}	
2021	Januari	Y_{121}	
	Februari	Y_{122}	

3.3 Langkah Penelitian

Langkah-langkah pada penelitian ini adalah sebagai berikut.

1. Melakukan analisis deskriptif pada data penjualan mobil Toyota di Indonesia.
2. Analisis menggunakan ARIMA
   a. Membagi data kedalam dua bagian yaitu *in sample* dan *out sample*.
   b. Mengidentifikasi pola dari data penjualan mobil Toyota dengan menggunakan *time series plot*.
   c. Identifikasi stasioneritas data penjualan mobil Toyota dalam mean maupun varians.
   d. Melakukan identifikasi model ARIMA dugaan menggunakan plot ACF dan PACF dari data penjualan mobil Toyota.
   e. Estimasi parameter berdasarkan pendugaan model dari plot ACF dan PACF kemudian melakukan uji signifikansi parameter.
   f. Melakukan *diagnostic checking* pada model ARIMA meliputi uji normalitas dan uji *white noise*.

g. Identifikasi pengamatan *outlier* pada masing-masing model ARIMA yang tidak memenuhi asumsi *white noise* dan berdistribusi normal.
h. Melakukan pemodelan ARIMA dengan menambahkan variabel *outlier*.
i. Estimasi parameter model ARIMA *outlier*.
j. Melakukan *diagnostic checking* atau pengecekan asumsi dari model yang memiliki parameter signifikan meliputi uji distribusi normal dan uji *white noise*.
k. Melakukan peramalan menggunakan model ARIMA *outlier* yang signifikan dan memenuhi asumsi.
l. Melakukan evaluasi model menggunakan kriteria RMSE untuk *in sample* dan sMAPE untuk *out sample* lalu membandingkan performa masing-masing model.
m. Pemilihan model terbaik ARIMA berdasarkan RMSE dan sMAPE terkecil.

3. Uji linieritas Teräsvirta
   a. Meregresikan Y_t sebagai variabel respon (y) dengan lag-lag signifikan (Y_{t-k}) berdasarkan plot PACF dari pemodelan ARIMA sebagai variabel prediktor (x) dengan tambahan variabel yaitu suku-suku polinomial seperti pada contoh persamaan (2.32).
   b. Menghitung nilai F dan membandingkan dengan nilai *P-value*.
   c. Membuat keputusan dan kesimpulan hasil pengujian.
4. Analisis menggunakan SVR (*Support Vector Regression*)
   a. Membagi kedalam dua bagian yaitu *in sample* dan *out sample*.
   b. Menyusun model SVR untuk meramalkan penjualan Toyota dengan variabel respon (y) adalah penjualan Toyota aktual (Y_t) dan variabel prediktor (x) adalah lag-lag penjualan Toyota (Y_{t-k}) dari hasil penguraian model ARIMA yang memiliki parameter signifikan pada langkah 2.

c. Menentukan range nilai parameter pada kernel *radial basis* yaitu *C*, ε, dan γ untuk optimasi *hyperplane* pada data *in sample* hingga diperoleh yang optimum menggunakan *grid search*.
d. Melakukan pemodelan SVR berdasarkan *range* kernel yang telah ditentukan.
e. Melakukan pemeriksaan solusi dari model SVR, model optimum diperoleh ketika nilai RMSE dan sMAPE minimum.
f. Melakukan peramalan menggunakan model SVR.
g. Melakukan evaluasi model menggunakan kriteria RMSE untuk *in sample* dan sMAPE untuk *out sample* serta membandingkan performa masing-masing model.
h. Pemilihan model terbaik SVR berdasarkan RMSE dan MAPE terkecil.

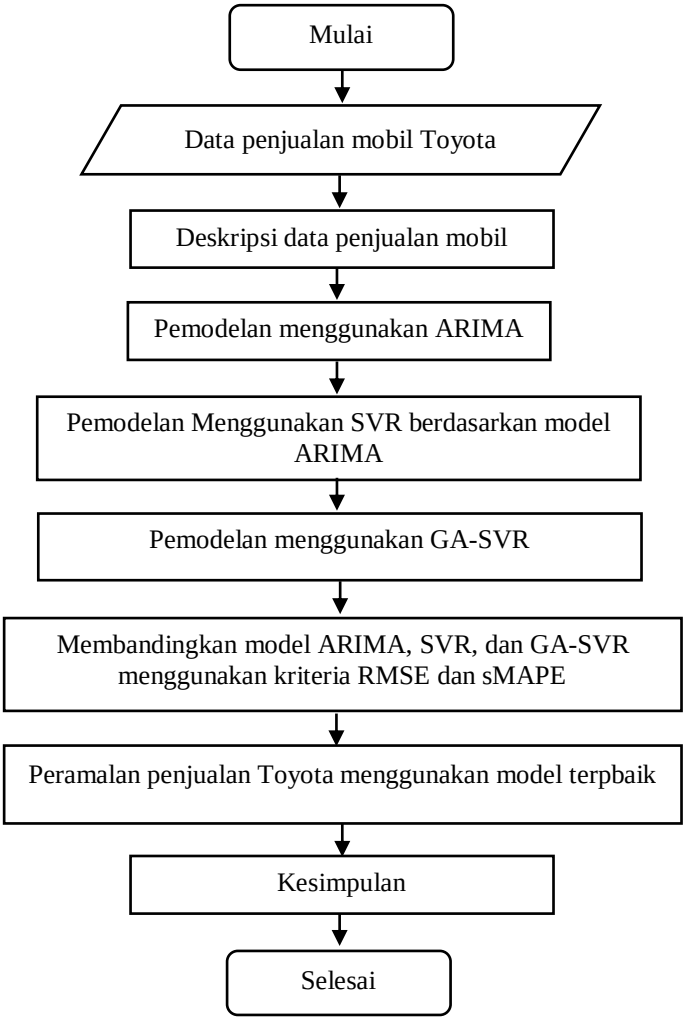
5. Analisis menggunakan GA-SVR (*Genetic Algorithm - Support Vector Regression*)
   a. Membagi data *in sample* dan *out sample.*
   b. Menginisiasi parameter awal yaitu C, γ, dan ε yang diperoleh dari pemodelan SVR pada langkah 4.
   c. Menyusun kromosom dengan membangkitkan populasi awal sebanyak 100, maksimum iterasi sebanyak 20, peluang mutasi sebesar 0,01; dan peluang *crossover* sebesar 0,8.
   d. Menentukan *fitness function* yang digunakan untuk menghitung nilai *fitness*. Pada penelitian ini akan digunakan nilai sMAPE sebagai nilai *fitness*.
   e. Melakukan seleksi kromosom sebanyak 100 dari 100 *parents* menggunakan metode *roulette wheel* untuk menentukan apakah sebuah kromosom dapat *survive* ke generasi selanjutnya.
   f. Melakukan pindah silang atau *crossover* pada individu yang terpilih sebagai orang tua dan mendapatkan individu baru.

g. Melakukan mutasi pada individu dengan mengganti minimal salah satu parameter dengan nilai baru.
h. Melakukan proses *elitism*.
i. Membangkitkan populasi *offspring* parameter baru setelah melalui proses seleksi, *crossover*, *mutation*, dan *elitism*.
j. Ulagi langkah 5c hingga 5h hingga parameter C, γ, dan ε sudah memenuhi kriteria nilai *fitness* terbaik. Solusi dikatakan telah mencapai kriteria apabila nilai sMAPE telah konvergen.
k. Melakukan pemodelan SVR menggunakan parameter yang telah dioptimasi menggunakan GA.
l. Melakukan evaluasi model menggunakan kriteria RMSE untuk *in sample* dan sMAPE untuk *out sample* serta membandingkan performa masing-masing model.
m. Pemilihan model terbaik SVR berdasarkan RMSE dan sMAPE terkecil.

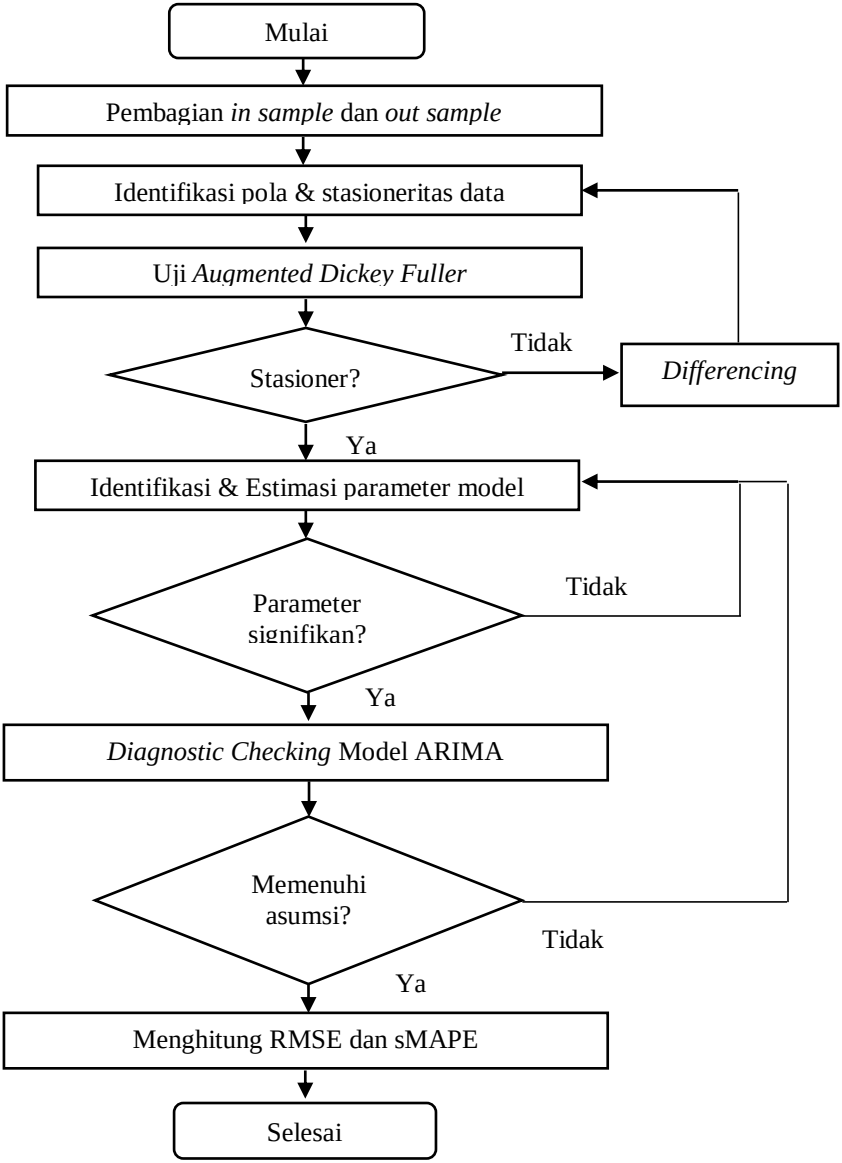
6. Membandingkan ketiga metode tersebut yaitu ARIMA, SVR, dan model GA-SVR menggunakan kriteria RMSE dan sMAPE.
7. Pemilihan model terbaik berdasarkan nilai RMSE dan sMAPE terkecil.
8. Membuat ramalan penjualan mobil 12 bulan kedepan berdasarkan model terbaik mulai Maret 2021 hingga Februari 2022.
9. Menarik kesimpulan dan saran berdasarkan hasil analisis.

3.4 Diagram Alir

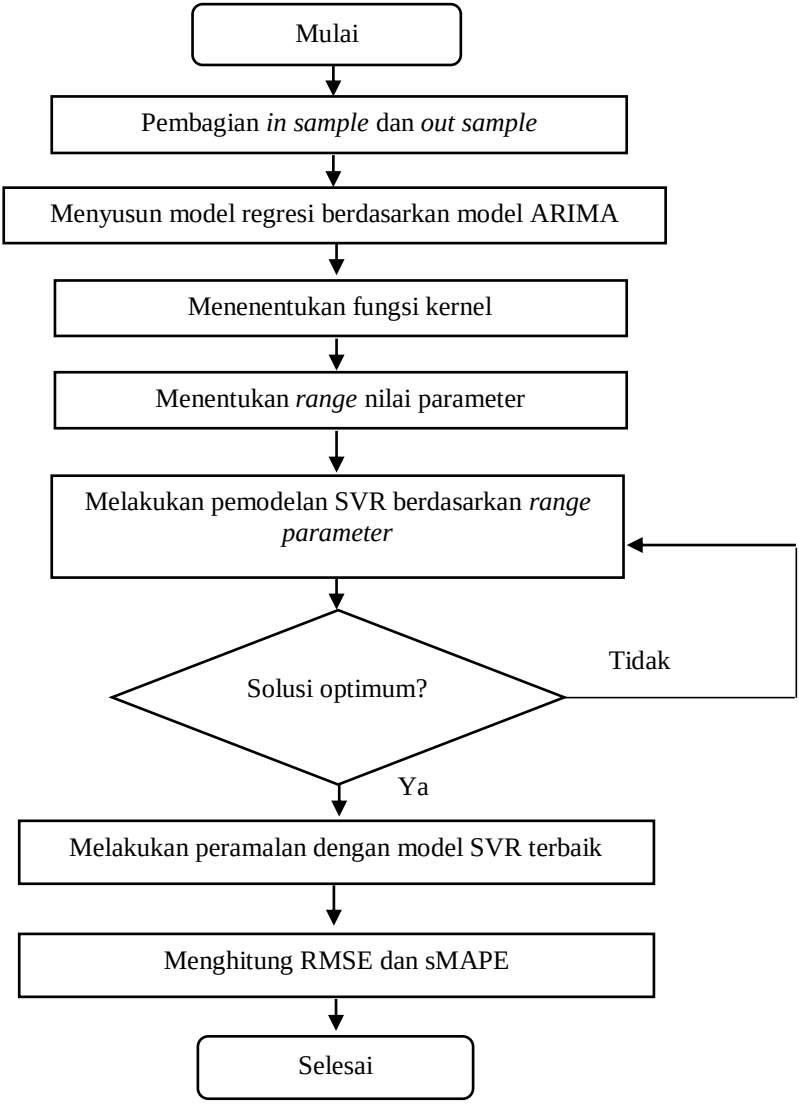
Berikut ini adalah diagram alir dari penelitian.



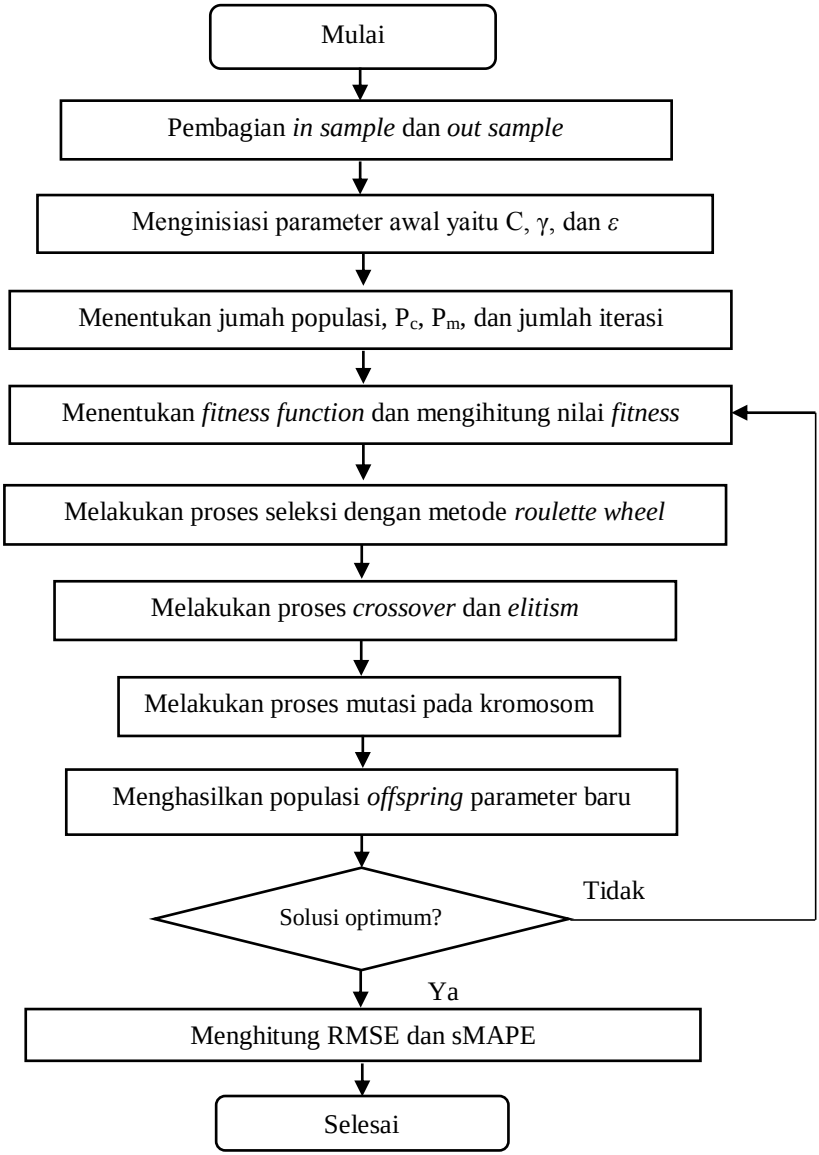
Gambar 3.1 Diagram Alir Penelitian



Gambar 3.2 Diagram Alir Tahapan ARIMA



Gambar 3.3 Diagram Alir Tahapan SVR



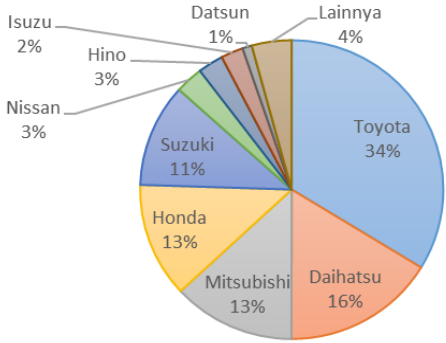
Gambar 3.4 Diagram Alir Tahapan GA-SVR

(*Halaman ini sengaja dikosongkan*)

BAB IV
ANALISIS DAN PEMBAHASAN

4.1 Karakteristik Data Penjualan Mobil di Indonesia

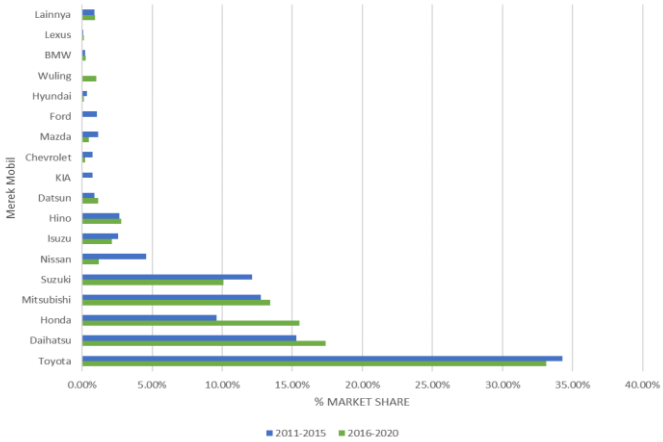
Penjualan mobil di Indonesia dikuasai oleh mobil-mobil buatan Jepang seperti Toyota, Daihatsu, Honda, Mitsubishi, dan Suzuki. Mobil produksi Eropa seperti Mazda, Ford, BMW, dan Mercedes-Benz memiliki *share* yang sangat kecil, bahkan kurang dari 1% selama 10 tahun terakhir, Gambar 4.1 menunjukkan perbandingan rata-rata *market share* bulanan beberapa merek mobil di Indonesia periode 2011 hingga 2020.



Gambar 4.1 *Market Share* Penjualan Merek Mobil di Indonesia periode 2011 hingga 2020

Gambar 4.1 memberikan informasi bahwa Toyota menguasai *market* mobil di Indonesia dengan *share* sebesar 34%, kemudian merek seperti Daihatsu, Mitshubishi, Honda dan Suzuki bersaing cukup ketat karena persentase *share* antara keempat merek tersebut memiliki selisih yang cukup kecil.

Persaingan pasar mobil di Indonesia cukup ketat, bahkan tak jarang terdapat ATPM yang gulung tikar karena sudah tidak mampu bersaning. Gambar 4.2 menjelaskan perkembangan *market share* merek-merek mobil di Indonesia yang dibagi dalam 2 periode 5 tahunan yaitu 2011 hingga 2020 dan 2016 hingga 2020.

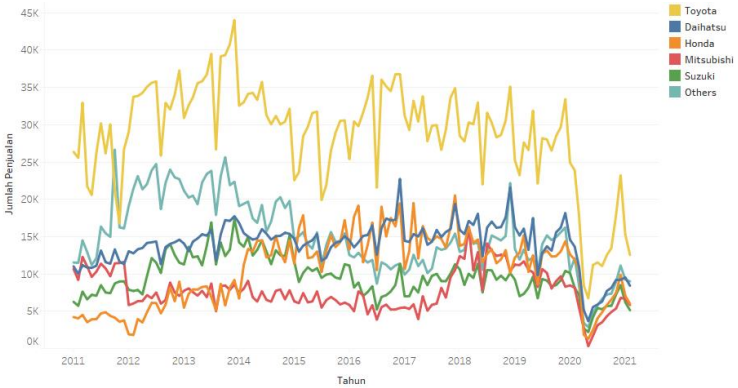


Gambar 4.2 Perbandingan *Market Share* Merek-merek mobil di Indonesia dibagi dalam Dua Periode

Berdasarkan informasi dari Gambar 4.2, Toyota memiliki rata-rata *market share* bulanan sebesar 34,37% pada periode 2011 hingga 2015, atau dengan kata lain, Toyota menguasai lebih dari sepertiga penjualan mobil nasional dengan unit terjual sebanyak 1,86 juta unit dari total *market* periode tersebut yaitu 5,43 unit. Selain Toyota, beberapa merek lainnya seperti Daihatsu, Honda, Mitsubishi, dan Suzuki juga memiliki *share* diatas 10%. Jika dibandingkan dengan periode 2016 hingga 2020, Toyota mengalami penurunan *share* menjadi 33,11% dari total penjualan sebanyak 4,91 juta unit. Hal ini disebabkan oleh penjualan mobil nasional yang merosot tajam di sepanjang tahun 2020 akibat dampak dari pandemi COVID-19. Meskipun begitu, merek-merek kompetitor Toyota seperti Daihatsu, Honda, dan Mitsubishi cenderung mengalami kenaikan *market share* pada periode 2016 hingga 2020.

Terdapat dua merek yang berhenti produksi di periode 2016 hingga 2020 yaitu KIA yang mulai berhenti menjual produknya tahun 2017, dan juga Ford yang telah menutup pabriknya di tahun 2018. Terlihat bahwa pada Gambar 4.1, *market share* kedua merek mobil tersebut turun mendekati 0% di periode 2016 hingga 2020.

Selain beberapa produk yang telah diskontinu, terdapat pula merek mobil baru yang masuk di Indonesia yaitu Wuling. Walaupun baru memulai penjualan di tahun 2017, Wuling berhasil menyaingi merek-merek yang telah lama ada di Indonesia seperti Datsun dan Nissan.



Gambar 4.3 Perbandingan Penjualan Toyota dengan Merek Lainnya 2011 hingga 2021

Gambar 4.3 memperlihatkan penjualan mobil Toyota bulanan dibandingkan dengan merek-merek besar di Indonesia yaitu Daihatsu, Honda, Mitsubishi, dan Suzuki menggunakan Tableau. Kategori mobil *others* meliputi semua merek mobil diluar kelima merek tersebut yang terdaftar di GAIKINDO. Penjualan mobil bulanan di Indonesia periode 2011 hingga 2021 cenderung memiliki pola yang fluktuatif. Gambar 4.2 memperlihatkan bahwa penjualan mobil Toyota selalu memimpin penjualan mobil nasional pada periode 2011 hingga 2021. Terjadi penurunan penjualan mobil nasional yang sangat tajam di bulan Mei 2020 yang akibat dari penyebaran pandemi COVID-19 di kuartal pertama 2020. Dampak ini dirasakan semua ATPM tak terkecuali Toyota.

Sebelum terjadi pandemi, penjualan mobil Toyota memang cenderung tidak stabil, terlihat bahwa penjualan naik dan turun secara tidak teratur. Penjualan tertinggi Toyota mencapai 44.029

unit pada bulan Desember 2013, sedangkan yang terendah pada bulan Mei 2020 yaitu sebanyak 6.272 unit. Penjualan merek lain seperti Daihatsu dan Suzuki cenderung stabil dari tahun ke tahun, sedangkan Mitsubishi dan Honda cenderung mengalami tren kenaikan. Berikut ini adalah rincian penjualan 5 merek mobil terbesar di Indonesia.

Tabel 4.1 Deskripsi Penjualan Mobil di Indonesia 2011 hingga 2021

Merek	Mean	St. Dev	Minimum	Median	Maksimum
Toyota	28759	6993	6727	30061	44029
Honda	10568	4702	1291	11759	20523
Daihatsu	13981	3048	3673	14466	22733
Mitsubishi	7776	2670	313	7467	16302
Suzuki	9566	2741	2205	9322	17339
Lainnya	15029	4929	2874	14443	26651

Berdasarkan Tabel 4.1, dapat diketahui bahwa nilai minimum penjualan semua merek memiliki rentang yang sangat besar dari nilai maksimum, sehingga dapat diketahui bahwa dampak COVID-19 membuat penjualan mobil mengalami penurunan yang sangat tajam. Nilai standar deviasi dari penjualan mobil Toyota merupakan yang paling tinggi dibandingkan merek lainnya, sehingga dapat dikatakan bahwa variasi jumlah penjualan mobil Toyota cenderung tinggi atau dengan kata lain, penjualan memiliki pola yang sangat fluktuatif. Selain itu, persentase penurunan yang dialami oleh Toyota sepanjang tahun 2020 merupakan yang tertinggi jika dibandingkan merek lainnya. Berikut ini adalah rincian penjualan mobil Toyota selama 10 tahun terakhir.

Tabel 4.2 Deskripsi Penjualan Mobil Toyota di Indonesia

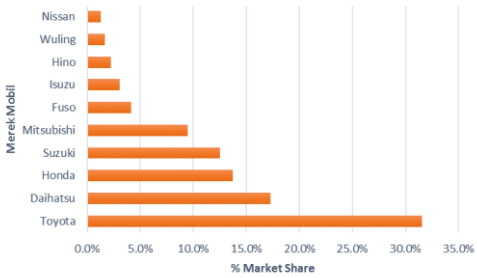
Tahun	Mean	Median	Std. Dev	Minimum	Maksimum
2011	25368	32943	4616	16771	32943
2012	33293	37258	3133	25857	37258
2013	36234	44029	4774	26679	44029
2014	32341	35733	1827	30042	35733
2015	27162	31746	4146	19902	31746
2016	32350	36771	4861	21563	36771
2017	30835	34888	2569	26621	34888

Tabel 4.2 Deskripsi Penjualan Mobil Toyota di Indonesia (*Lanjutan*)

Tahun	Mean	Median	Std. Dev	Minumum	Maksimum
2018	29672	35107	3200	22002	35107
2019	27584	33420	3226	22123	33420
2020	15222	24928	6198	6727	24928

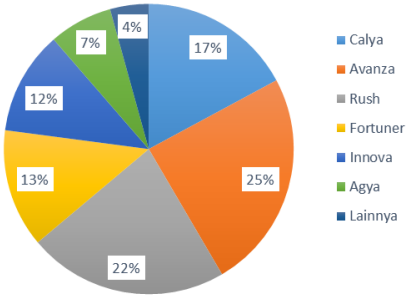
Berdasarkan rincian penjualan Toyota pada Tabel 4.2, rata-rata penjualan mobil Toyota pada tahun 2020 hanya sebanyak 15.222 unit, atau dengan kata lain penjualan tahun 2020 merupakan yang paling rendah selama 10 tahun terakhir. Penjualan terendah terjadi pada bulan Mei 2020 dengan unit terjual hanya 6.727, sedangkan penjualan paling tinggi sebesar 24.928 unit pada bulan Januari 2020, dimana pada saat itu pandemi COVID-19 belum menyebar di Indonesia. Penurunan penjualan yang dialami Toyota sepanjang tahun 2020 mencapai 44,8% jika dibandingkan pada tahun 2019. Selain itu, standar deviasi yang tinggi pada tahun 2020 mengindikasikan adanya variasi penjualan yang cukup besar selama tahun 2020.

Meskipun mengalami penurunan yang sangat tajam, Toyota tetap menduduki posisi pertama *market share* penjualan *retail* mobil di Indonesia di tahun 2020 dengan *share* sebesar 31,6% atau sepertiga pangsa pasar di Indonesia, sedangkan di posisi kedua disusul oleh Daihatsu dengan *share* sebesar 17,3%. Rincian *market share* penjualan mobil di Indonesia tahun 2020 dijelaskan pada Gambar 4.4.



Gambar 4.4 *Market Share* Penjualan Mobil di Indonesia Tahun 2020

Tingginya angka penjualan mobil Toyota berkat kontribusi dari penjualan mobil tipe *passenger car* atau mobil penumpang. Sebanyak 76,8% dari total penjualan mobil di Indonesia tahun 2020 dikuasai oleh mobil *passenger car* dan Toyota merajai pasar mobil penumpang dengan mengandalkan penjualan tipe MPV dan SUV seperti Avanza, Calya, Rush, dan Fortuner yang dijelaskan pada Gambar 4.5.



Gambar 4.5 Komposisi Penjualan Mobil Toyota Tahun 2020

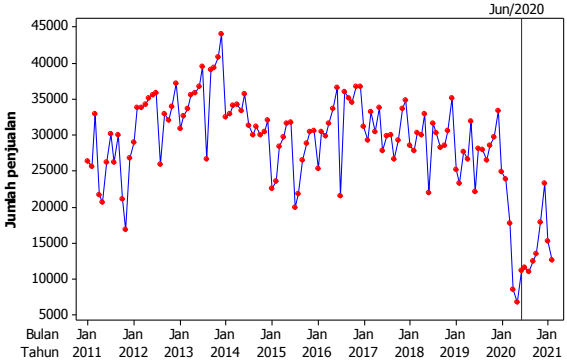
4.2 Pemodelan Penjualan Mobil Toyota Menggunakan ARIMA

Pada subbab ini akan dibahas pemodelan dan peramalan penjualan mobil Toyota menggunakan ARIMA (*Autoregressive Integrated Moving Average*). Langkah-langkah pemodelan secara umum meliputi pemeriksaan pola data, identifikasi model ARIMA, pemilihan model terbaik, dan peramalan penjualan mobil Toyota 12 bulan kedepan.

4.2.1 Identifikasi Pola Penjualan Mobil Toyota

Sebelum melakukan analisis menggunakan ARIMA, terlebih dahulu dilakukan identifikasi pola data penjualan mobil Toyota menggunakan *time series plot*. Identifikasi pola dari data bertujuan untuk mengetahui apakah data penjualan mobil telah stasioner secara visual. Data dibagi menjadi dua bagian yaitu *in sample* dimulai dari Januari 2011 hingga Juni 2020, sedangkan *out*

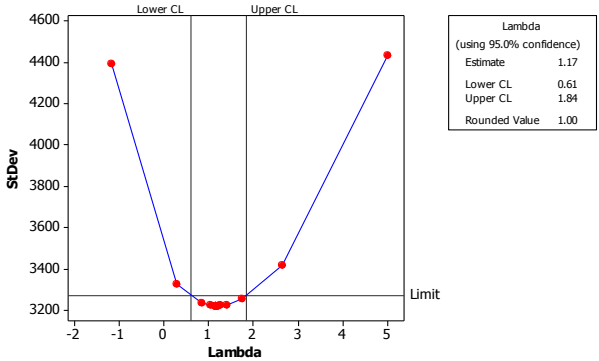
sample dimulai dari Juli 2020 hingga Februari 2021. Gambar 4.6 menunjukkan *time series plot* dari penjualan mobil Toyota.



Gambar 4.6 *Time Series Plot* Penjualan Mobil Toyota

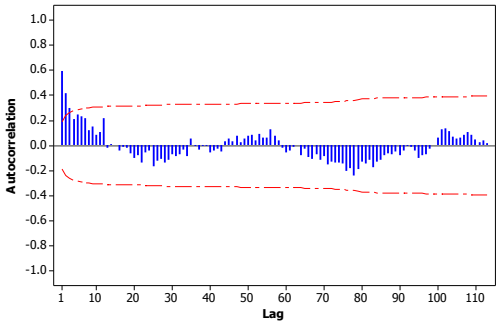
Berdasarkan Gambar 4.6, dapat diketahui bahwa penjualan mobil Toyota memiliki pola yang fluktuatif dan tidak terdapat pola-pola musiman (seasonal). Jika dilihat secara visual, data cenderung tidak stabil karena penjualan sangat variatif dari tahun ke tahun. Selain itu, terdapat data penjualan yang sangat tinggi yaitu pada bulan Desember 2013 dan paling rendah pada bulan Mei 2020. Meskipun secara visual data cenderung tidak stasioner, perlu dilakukan pembuktian dengan menggunakan analisis yang sesuai untuk memeriksa apakah data benar-benar tidak stasioner dalam mean maupun varians.

Untuk menstasionerkan varians data, dapat dilakukan transformasi Box-Cox. Gambar 4.7 memberikan informasi bahwa nilai *rounded value lambda* (λ) pada plot Box-Cox bernilai 1.00 sehingga data tidak perlu ditransformasi dan dapat dikatakan bahwa data telah stasioner dalam varians.



Gambar 4.7 Transformasi *Box-Cox*

Selanjutnya adalah pemeriksaan stasioneritas data dalam mean menggunakan Plot ACF (*Autocorrelation Function*). Berikut ini adalah plot ACF data penjulan Toyota.

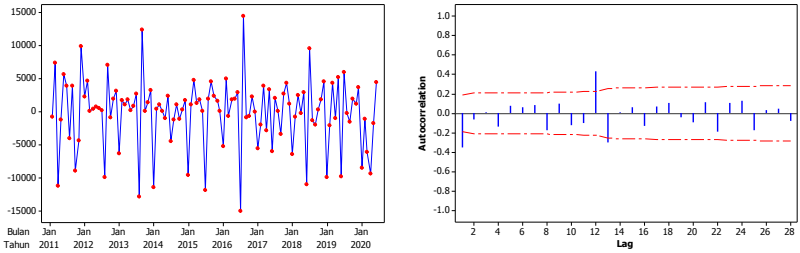


Gambar 4.8 ACF Plot Penjualan Mobil Toyota

Berdasarkan Gambar 4.8, plot ACF memiliki bentuk *cuts off* pada tiga *lag* pertama ($\alpha = 5\%$) . Meskipun begitu, plot ACF membentuk pola sinusoidal, sehingga masih belum dapat diidentifikasi apakah data stasioner dalam mean. Oleh karena itu, perlu diperkuat dengan menggunakan uji ADF (*Augmented Dickey Fuller*)

Berdasarkan hasil pengujian yang dirujuk pada Lampiran 2, diperoleh hasil bahwa *p-value* sebesar 0,7197 yang lebih besar dari taraf signifikan α =0,05 sehingga H_0 gagal ditolak, yang artinya

data penjualan mobil Toyota tidak stasioner dalam mean. Berdasarkan hal tersebut, dilakukan *differencing* untuk mengatasi masalah stasioneritas data. Berikut ini adalah plot *time series* hasil *differencing* orde pertama.



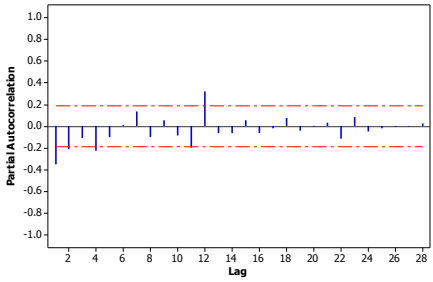
(a) **(b)**

Gambar 4.9 (a) *Time Series Plot* Penjualan Mobil Toyota dan **(b)** Plot ACF Hasil *Regular Differencing* orde 1 (d = 1)

Hasil *differencing* pada Gambar 4.9 (a) menunjukkan data memiliki fluktuasi yang rendah jika dibandingkan pada Gambar 4.6. Jika dilihat dari plot ACF pada Gambar 4.9 (b), data penjualan mobil telah stasioner karena *lag-lag* pada plot memiliki pola *cuts off* di *lag* 1, 12, dan 13 pada taraf α = 5%. Hal ini diperkuat pula dengan hasil uji ADF yang dirujuk dari Lampiran 2, dimana nilai p sebesar 0,01 bernilai lebih kecil dari taraf signifikan α = 0,05, sehingga diputuskan tolak H_0 yang berarti data penjualan Toyota telah stasioner.

4.2.2 Identifikasi Model ARIMA

Data penjualan mobil Toyota telah stasioner dengan *differencing*, sehingga dapat dilanjutkan pemodelan ARIMA. Plot ACF dan PACF digunakan untuk mengidentifikasi dugaan model ARIMA yang terbentuk berdasarkan *lag-lag* yang signifikan. Berdasarkan Gambar 4.9(b), pada plot ACF terdapat *lag* signifikan yaitu *lag* 1, *lag* 12, dan *lag* 13.



Gambar 4.10 Plot PACF data Penjualan Mobil Toyota Setelah Regular *Differencing* Orde 1 (d = 1)

Berdasarkan Gambar 4.10 plot PACF memiliki *lag* signifikan yaitu *lag* 1, *lag* 2, *lag* 4, dan *lag* 12. Berdasarkan *lag-lag* signifikan pada ACF dan PACF, diperoleh dugaan-dugaan model ARIMA dari *output* SAS yang dirangkum pada dari Lampiran 3 sebagai berikut.

Tabel 4.3 Hasil Estimasi Dugaan Model ARIMA

Model	Estimasi Parameter	SE. Estimasi	Nilai *p*
ARIMA(1,1,0)	$\hat{\phi}_1$ = -0,34923	0,08888	0,0001*
ARIMA(2,1,0)	$\hat{\phi}_1$ = -0,42115 $\hat{\phi}_2$ = -0,20747	0,09313 0,09318	0,0001* 0,0280
ARIMA([12],1,0)	$\hat{\phi}_{12}$ = 0,47765	0,08839	0,0001*
ARIMA(0,1,1)	$\hat{\theta}_1$ = 0,54193	0,08160	0,0001*
ARIMA(0,1,[12])	$\hat{\theta}_{12}$ = -0,40595	0,09231	0,0001*
ARIMA(0,1,[1,12,13])	$\hat{\theta}_1$ = 0,45798 $\hat{\theta}_{12}$ = -0,35036 $\hat{\theta}_{13}$ = 0,28501	0,08459 0,10067 0,10427	0,0001* 0,0007* 0,0073*
ARIMA(1,1,0)(1,0,0)^{12}	$\hat{\phi}_1$ = -0,30149 $\hat{\Phi}_{12}$ = 0,44196	0,08043 0,08419	0,0003* 0,0001*

Tabel 4.3 Hasil Estimasi Dugaan Model ARIMA (*Lanjutan)*

Model	Estimasi Parameter	SE. Estimasi	Nilai *p*
ARIMA(1,1,0)(0,0,1)^{12}	$\hat{\phi}_1$ = -0,31786	0,09507	<0,00001*
	$\hat{\Theta}_1$ = -0,40244	0,09133	0,0007*
ARIMA(0,1,1)(1,0,0)^{12}	$\hat{\theta}_1$ = 0,44587	0,08520	<0,0001*
	$\hat{\Phi}_1$ = 0,43292	0,09461	<0,0001*
ARIMA(0,1,1)(0,0,1)^{12}	$\hat{\theta}_1$ =0,37143	0,08122	<0,0001*
	$\hat{\Theta}_1$ =-0,36104	0,08728	<0,0001*

*) signifikan pada taraf α = 0,05

Hipotesis nol menguji bahwa parameter memiliki pengaruh signifikan terhadap Y_t. Dengan menggunakan taraf signifikan α = 0,05, hasil estimasi parameter pada Tabel 4.3 telah signifikan karena $p < 0,05$. Setelah mendapatkan dugaan model ARIMA yang signifikan, dilakukan *diagnostic checking* pada residual yang meliputi asumsi distribusi normal dan asumsi *white noise*. Tabel 4.4 merupakan hasil *diagnostic checking* pada residual ARIMA.

Tabel 4.4 *Diagnostic Checking* pada Residual Model ARIMA

Model	Uji *White Noise*		Uji Normalitas
	Lag	Nilai *p*	Nilai *p*
ARIMA(1,1,0)	6	0,0388	<0,0100*
	12	0,0001*	
	18	0,0001*	
	24	0,0001*	
ARIMA(2,1,0)	6	0,0623	<0,0100*
	12	0,0005*	
	18	0,0010*	
	24	0,0016*	
ARIMA([12],1,0)	6	0,0402*	0,0343*
	12	0,1531	
	18	0,2683	
	24	0,5208	

Tabel 4.4 *Diagnostic Checking* pada Residual Model ARIMA (*Lanjutan*)

Model	**Uji *White Noise***		**Uji Normalitas**
	Lag	**Nilai *p***	**Nilai *p***
ARIMA(0,1,1)	6	0,3124	<0,0100*
	12	0,0073*	
	18	0,0108*	
	24	0,0158*	
ARIMA(0,1,[12])	6	0,0297*	0,0159*
	12	0,0524	
	18	0,0661	
	24	0,0962	
ARIMA(0,1,[1,12,13])	6	0,2733	<0,0100*
	12	0,5844	
	18	0,7843	
	24	0,9035	
ARIMA(1,1,0)(1,0,0)^{12}	6	**0,1522**	**0,1111**
	12	**0,6162**	
	18	**0,8440**	
	24	**0,9746**	
ARIMA(1,1,0)(0,0,1)^{12}	6	0,1498	0,0498*
	12	0,4211	
	18	0,4415	
	24	0,5268	
ARIMA(0,1,1)(1,0,0)^{12}	6	0,4354	0,0249*
	12	0,8543	
	18	0,9017	
	24	0,9811	
ARIMA(0,1,1)(0,0,1)^{12}	6	0,3655	<0,0100*
	12	0,5478	
	18	0,2783	
	24	0,3696	

*) signifikan pada taraf $\alpha = 0,05$

Berdasarkan Tabel 4.4, dari semua model yang memiliki parameter signifikan, hanya terdapat satu model yang memenuhi asumsi residual *white noise* dan berdistribusi normal yaitu ARIMA(1,1,0)(1,0,0)^{12}. Terdapat empat model yang memenuhi asumsi *white noise*, namun tidak memenuhi asumsi distribusi

normal yaitu ARIMA (0,1,[1,12,13]), ARIMA(1,1,0)(0,0,1)^{12}, ARIMA(0,1,1)(1,0,0)^{12}, dan ARIMA(0,1,1)(0,0,1)^{12} , sisanya tidak memenuhi baik asumsi normalitas maupun *white noise*.

Pemodelan ARIMA dilanjutkan dengan deteksi pengamatan *outlier* untuk mengatasi permasalahan asumsi yang tidak terpenuhi. Pada penelitian ini, akan diamati dua tipe *outlier* yaitu *level shift* dan *additive* sebanyak 10 pengamatan. Pemodelan ARIMA *outlier* menggunakan semua model kecuali model ARIMA(1,1,0)(1,0,0)^{12} karena sudah memenuhi asumsi residual berdistribusi normal dan *white noise*. Rincian pengamatan *outlier* dari masing-masing model ARIMA yang signifikan dirujuk pada Lampiran 6.

Setelah mendapatkan pengamatan-pengamatan *outlier*, masing-masing model ARIMA ditambahkan variabel *dummy* yang merepresentasikan pengamatan *outlier* (I_t), kemudian dilanjutkan estimasi model ARIMA *outlier* yang dirujuk pada Lampiran 8. Selanjutnya adalah memeriksa asumsi residual *white noise* dan berdistribusi normal yang dirangkum pada Tabel 4.5 sebagai berikut.

Tabel 4.5 *Diagnostic Checking* pada Residual Model ARIMA *Outlier*

Model	**Uji *White Noise***		**Uji Normalitas**
	Lag	**Nilai *p***	**Nilai *p***
ARIMA(1,1,0) + $\omega\ I_t^{(T)}$	6	0,2785	<0,0100*
	12	0,1003	
	18	0,0940	
	24	0,0030*	
ARIMA(2,1,0) + $\omega\ I_t^{(T)}$	6	0,4280	<0,0100*
	12	0,2080	
	18	0,2050	
	24	0,0219*	
ARIMA([12],1,0) + $\omega\ I_t^{(T)}$	6	0,4698	0,0317*
	12	0,5830	
	18	0,6542	
	24	0,5346	

Tabel 4.5 *Diagnostic Checking* pada Residual Model ARIMA *Outlier* (Lanjutan)

Model	Uji Ljung Box		Uji Normalitas
	Lag	**Nilai *p***	**Nilai *p***
ARIMA(0,1,1) + $\omega\ I_t^{(T)}$	6	0,0848	0,0275*
	12	0,0152*	
	18	0,0241*	
	24	0,0021*	
ARIMA(0,1,[12]) + $\omega\ I_t^{(T)}$	6	**0,6316**	**0,0655**
	12	**0,6274**	
	18	**0,7152**	
	24	**0,2228**	
ARIMA(0,1,[1,12,13]) + $\omega\ I_t^{(T)}$	6	**0,4114**	**>0,1500**
	12	**0,3718**	
	18	**0,5869**	
	24	**0,4028**	
ARIMA(1,1,0)(0,0,1)^{12} + $\omega\ I_t^{(T)}$	6	**0,6892**	**>0,1500**
	12	**0,4390**	
	18	**0,2391**	
	24	**0,0508**	
ARIMA(0,1,1)(1,0,0)^{12} + $\omega\ I_t^{(T)}$	6	0,6302	0,0421*
	12	0,8114	
	18	0,5813	
	24	0,7898	
ARIMA(0,1,1)(0,0,1)^{12} + $\omega\ I_t^{(T)}$	6	0,2880	0,0209*
	12	0,2222	
	18	0,2392	
	24	0,0576	

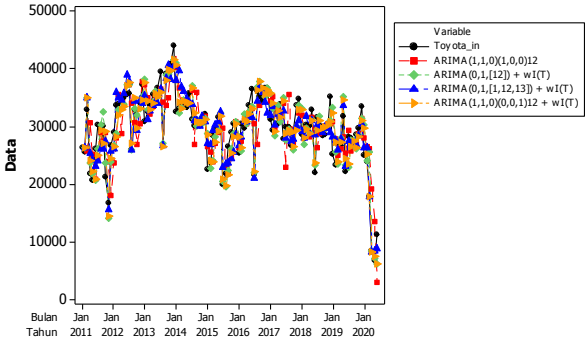
*) signifikan pada taraf α = 0,05

Tabel 4.5 menjelaskan bahwa terdapat tiga model yang memenuhi asumsi residual *white noise* dan berditsribusi normal yaitu ARIMA (0,1,[12]) + $\omega I_t^{(T)}$, ARIMA(0,1,[1,12,13]) + $\omega I_t^{(T)}$, dan ARIMA(1,1,0)(0,0,1)^{12} + $\omega I_t^{(T)}$. Penguraian keempat model tersebut dirujuk pada Lampiran 9.

Model-model tersebut akan dibandingkan apakah hasil ramalannya memiliki akurasi yang tinggi. Pemilihan model terbaik akan dibahas pada sub subbab selanjutnya.

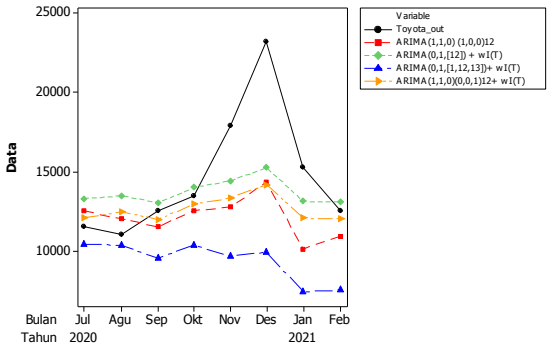
4.2.3 Pemilihan Model Terbaik ARIMA

Model terbaik ARIMA akan dipilih jika hasil ramalannya dapat merepesentasikan data sebenarnya. Jika nilai RMSE dan sMAPE semakin rendah, maka model ARIMA akan semakin akurat. Berdasarkan pemodelan yang telah dilakukan, model ARIMA yang terpilih adalah ARIMA(1,1,0)(1,0,0)^{12}, ARIMA(0,1,[12]) + $\omega I_t^{(T)}$, ARIMA(0,1,[1,12,13]) + $\omega I_t^{(T)}$, dan ARIMA(1,1,0)(0,0,1)^{12} + $\omega I_t^{(T)}$. Berkut ini adalah perbandingan akurasi model ARIMA yang terpilih pada data *in sample* secara visual.



Gambar 4.11 *Time Series Plot* perbandingan Model ARIMA data *In Sample*

Berdasarkan Gambar 4.11, dapat diketahui bahwa keempat model menghasilkan prediksi dengan akurasi yang cukup tinggi karena hasilnya mampu menggambarkan pola data aktual secara baik. Selanjutnya adalah pemeriksaan hasil ramalan pada data *out sample* sebagai berikut.



Gambar 4.12 *Time Series Plot* perbandingan Model ARIMA data *Out Sample*

Gambar 4.12 memperlihatkan bahwa hasil ramalan keempat model ARIMA tidak dapat menggambarkan pola data aktual dengan akurat. Namun dapat dilihat bahwa model ARIMA (0,1,[12]) + $\omega I_t^{(T)}$ memiliki performansi yang lebih baik dalam meramalkan jika dibandingkan model lainnya karena mendekati data aktual.

Meskipun hasil ramalan dari data *out sample* secara visual memiliki *error* yang cukup tinggi, pemilihan model terbaik tetap dilakukan menggunakan kriteria *error* yaitu RMSE dan sMAPE. Berikut ini adalah perbandingan dari kriteria *error* pada dua model ARIMA.

Tabel 4.6 Pemilihan Model Terbaik ARIMA

Model	RMSE *In sample*	RMSE *Out sample*	sMAPE *Out sample*
ARIMA(1,1,0)(1,0,0)^{12}	4311,401	4147,155	28,841
ARIMA(0,1,[12]) + $\omega I_t^{(T)}$	**2941,179**	**3353,528**	**15,578**
ARIMA(0,1,[1,12,13]) + $\omega I_t^{(T)}$	2632,845	6620,715	40,916
ARIMA(1,1,0)(0,0,1)^{12} + $\omega I_t^{(T)}$	2941,113	3799,417	16,217

Berdasarkan Tabel 4.6, dapat diketahui bahwa nilai RMSE untuk data *in sample* paling kecil dimiliki oleh model ARIMA (0,1,[1,12,13]) + $\omega I_t^{(T)}$, sehingga model tersebut merupakan model

terbaik untuk data *in sample*, sedangkan untuk data *out sample*, model ARIMA(0,1,[12]) + $\omega I_t^{(T)}$ merupakan model terbaik karena memiliki nilai RMSE dan sMAPE *out sample* yang paling rendah dibandingkan model lainnya. Model ARIMA(0,1,[12]) + $\omega I_t^{(T)}$ dapat dituliskan sebagai berikut.

$$(1-B)Y_t = (1-\theta_{12}B^{12})a_t + \omega I_t^{(T)}$$

$$Y_t - Y_{t-1} = (1-\theta_{12}B^{12})a_t + \omega I_t^{(T)}$$

$$Y_t - Y_{t-1} = a_t - \theta_{12}a_{t-12} + \omega I_t^{(T)}$$

$$Y_t = Y_{t-1} + a_t + 0,40206a_{t-12} + \omega_{AO}I_{67}^{(67)} + \omega_{AO}I_{32}^{(32)} + \omega_{AO}I_3^{(3)} + \omega_{AO}I_{101}^{(101)}$$
$$+\omega_{AO}I_{90}^{(90)} + \omega_{AO}I_{11}^{(11)} + \omega_{AO}I_{20}^{(20)} + \omega_{LS}I_{55}^{(55)} + \omega_{LS}I_{112}^{(112)} + \omega_{AO}I_9^{(9)}$$

$$Y_t = Y_{t-1} + a_t + 0,40206a_{t-12} - 14380,2I_{67}^{(67)} - 12377,7I_{32}^{(32)} + 9337,4I_3^{(3)} + 7952,9I_{101}^{(101)}$$
$$-6670,1I_{90}^{(90)} - 7005,5I_{11}^{(11)} - 8199,3I_{20}^{(20)} - 8764,1I_{55}^{(55)} - 8764,1I_{112}^{(112)} + 6376,6I_9^{(9)}$$

Penjualan Toyota pada bulan ke *t* dipengaruhi pada penjualan pada bulan sebelumnya (*t*-1) dan dipengaruhi pula dengan pengamatan-pengamatan *outlier*.

4.3 Pemodelan Penjualan Mobil Toyota Menggunakan *Support Vector Regression*

Setelah memodelkan penjualan mobil Toyota menggunakan ARIMA, dilanjutkan dengan pemodelan menggunakan *Support Vector Regression* (SVR). Sebelum melakukan pemodelan, terlebih dahulu dilakukan uji linieritas Teräsvirta untuk mengetahui apakah data penjualan mobil Toyota bersifat linier atau tidak. Berdasarkan plot PACF pada Gambar 4.10, terdapat *lag* yang keluar batas signifikansi yaitu *lag* 1, *lag* 2, *lag* 4, dan *lag* 12. Oleh karena itu, lag-lag tersebut dijadikan input untuk uji Teräsvirta.

Berdasarkan hasil pengujian Teräsvirta pada Lampiran 9 diperoleh hasil bahwa nilai *p* sebesar $4,91 \times 10^{-7}$ yang bernilai lebih kecil daripada taraf signifikan $\alpha = 0,05$ sehingga dapat dikatakan bahwa data penjualan Toyota bersifat non linier. Dengan demikian, pemodelan SVR dapat dilanjutkan karena data penjualan mobil Toyota memiliki bersifat non linier.

Input pemodelan SVR pada penelitian ini dibatasi dari *lag-lag* dari hasil penguraian model ARIMA pada Tabel 4.3 yang memiliki parameter signifikan tanpa melihat asumsi residual berdistribusi normal dan *white* noise. Terdapat 10 model yang memiliki parameter yang signifikan, sebagai ilustrasi, berikut merupakan penguraian model ARIMA (1,1,0).

$$(1-\phi_1 B^1)\left[(1-B)Y_t\right] = a_t$$

$$(1-\phi_1 B^1)(Y_t - Y_{t-1}) = a_t$$

$$Y_t - Y_{t-1} - \phi_1 Y_{t-1} + \phi_1 Y_{t-2} = a_t$$

$$Y_t - Y_{t-1} - \phi_1 (Y_{t-1} - Y_{t-2}) = a_t$$

$$Y_t = Y_{t-1} - 0,34923(Y_{t-1} - Y_{t-2}) + a_t \,.$$

Berdasarkan penguraian model ARIMA(1,1,0), diperoleh input *lag* yaitu Y_{t-1} dan Y_{t-2} , kemudian dengan cara yang sama, dilakukan penguraian model lainnya sehingga diperoleh input *lag* untuk pemodelan SVR yang dirangkum pada Tabel 4.7.

Tabel 4.7 Input *Lag* SVR

Input ke-	Model	Input *Lag*
1	ARIMA(1,1,0)	Y_{t-1} Y_{t-2}
2	ARIMA(2,1,0)	Y_{t-1} , Y_{t-2} , Y_{t-3}
3	ARIMA([12],1,0) ARIMA(0,1,1)(1,0,0)^{12} ARIMA(0,1,1)(0,0,1)^{12}	Y_{t-1} , Y_{t-12} , Y_{t-13}
4	ARIMA (0,1,1) ARIMA (0,1,[12]) ARIMA (0,1,[1,12,13])	Y_{t-1}
5	ARIMA(1,1,0)(1,0,0)^{12} ARIMA(1,1,0)(0,0,1)^{12}	Y_{t-1} , Y_{t-2} , Y_{t-12} , Y_{t-13}

Metode SVR dengan kernel RBF (*Radial Basis Function*) memiliki 3 parameter yaitu γ, *C*, dan *ε*. Parameter terbaik pada fungsi kernel ditentukan dengan mencoba beberapa kemungkinan nilai pada rentang tertentu untuk membuat *hyperplane*. Nilai *range*

parameter yang optimum ditandai dengan nilai RMSE dan sMAPE paling kecil. Tabel 4.9 menunjukkan rentang nilai parameter yang digunakan.

Tabel 4.8 *Range* Parameter

Parameter	***Range***
ε	0,01; 0,1
γ	2^{-11}; 2^{-9}; 2^{-7}; …; 2^{-1}
C	2^{-5}; 2^{0}; 2^{5}; 2^{10} ,2^{15}

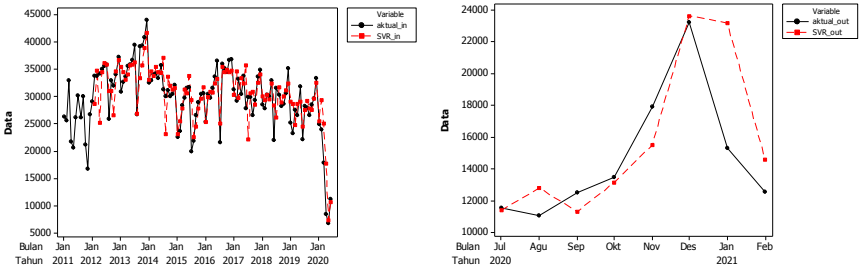
Pada penentuan parameter terbaik, dilakukan metode *grid search* atau mencoba kombinasi dari beberapa nilai parameter pada *range* tertentu. Tabel 4.10 merupakan rangkuman hasil pencarian parameter ε, γ dan C yang optimum pada masing-masing input.

Tabel 4.9 Parameter Optimum Masing-masing Input SVR

Model Input	ε	γ	**C**	**RMSE *in sample***	**RMSE *out sample***	**sMAPE *out Sample***
$f(Y_{t-1}, Y_{t-2})$	0,1	2^{-1}	2^{5}	3694,06	5244,71	24,40
	0,01	2^{-3}	2^{5}	4568,25	5529,89	24,33
	0,01	2^{-1}	2^{5}	3743,88	4938,53	22,68
$f(Y_{t-1}, Y_{t-2}, Y_{t-3})$	0,1	2^{-11}	2^{15}	4577,56	4844,33	23,54
	0,01	2^{-11}	2^{10}	4812,40	4652,17	23,61
$f(Y_{t-1}, Y_{t-12}, Y_{t-13})$	0,1	2^{-7}	2^{15}	3276,289	5095,44	23,43
	0,1	2^{-5}	2^{5}	3519,501	4346,84	20,00
$f(Y_{t-1})$	0,1	2^{-3}	2^{5}	4729,943	4952,43	18,05
$f(Y_{t-1}, Y_{t-2}, Y_{t-12}, Y_{t-13})$	**0,1**	**2^{-5}**	**2^{5}**	**3209,847**	**3101,66**	**12,57**
	0,01	2^{-7}	2^{10}	3391,724	3674,54	15,56
	0,01	2^{-5}	2^{5}	3241,960	3114,95	15,06

Berdasarkan Tabel 4.9, diperoleh bahwa model terbaik SVR pada peramalan penjualan Toyota adalah input $f\ (Y_{t-1}, Y_{t-2}, Y_{t-12}, Y_{t-13})$ dengan parameter ε = 0,1; $\gamma = 2^{-5}$; dan $C = 2^{5}$ karena menghasilkan nilai RMSE dan SMAPE baik *in sample* maupun *out sample* paling rendah dibandingkan model

input lainnya. Berikut ini adalah model terbaik SVR yang digambarkan secara visual menggunakan *time series plot*.



(a) **(b)**

Gambar 4.13 *Time Series Plot* Model SVR Terbaik pada Data **(a)** *In Sample* dan **(b)** *Out Sample*

Berdasarkan Gambar 4.13 (a), dapat dilihat bahwa hasil ramalan input $f\ (Y_{t-1,}\ Y_{t-2,}\ Y_{t-12,}\ Y_{t-13})$ memiliki akurasi yang baik dalam meramalkan penjualan Toyota karena plot hasil ramalan dapat menggambarkan data aktual, selain itu pada Gambar 4.13b, dapat dilihat bahwa hasil ramalan *out sample* juga dapat menggambarkan pola data aktual dengan baik. Oleh karena itu, dapat disimpulkan bahwa model SVR input $f\,(Y_{t-1,}\ Y_{t-2,}\ Y_{t-12,}\ Y_{t-13})$ merupakan model terbaik.

4.4 Pemodelan Penjualan Mobil Toyota Menggunakan *Support Vector Regression* dengan Optimasi Algoritma Genetika (GA-SVR)

Pada subbab sebelumnya, telah diperoleh parameter SVR yang menghasilkan nilai RMSE dan sMAPE yang minimum menggunakan metode *grid search*. Agar memperoleh nilai *error* yang lebih optimum, dapat dilakukan optimasi pada parameter SVR salah satunya menggunakan metode GA (*Genetic Algorithm*). Pada subbab ini, analisis akan dilakukan optimasi menggunakan input SVR seperti pada Tabel 4.8.

Langkah pertama pada GA adalah mendefnisikan variabel yang digunakan yaitu jumlah populasi (kromosom) yaitu sebanyak 100, batas iterasi sebanyak 20, dan peluang terjadinya pindah silang. Kromosom yang dibangkitkan memiliki tiga gen yaitu parameter model SVR yang terdiri atas C, γ, dan ε. Sebagai ilustrasi akan digunakan input $f(Y_{t-1,}\, Y_{t-2,})$ dengan parameter terbaik diperoleh $\varepsilon = 0{,}01$; $\gamma = 2^{-1}$; dan $C = 2^{5}$.

ε	γ	C

→

0,01	2^{-1}	2^{5}

Gambar 4.14 Ilustrasi Bentuk Kromosom

Nilai parameter yang telah diperoleh akan dijadikan *initial value* pada GA-SVR kemudian akan di *generate* sebanyak jumlah populasi sehingga terbentuk 100 kromosom dengan nilai yang bervariasi.

Kromosom yang baik ditentukan dari nilai *fitness function*. Pada penelitian ini, digunakan nilai sMAPE. Jika nilai sMAPE semakin kecil, maka kromosom dikatakan baik dan dapat bertahan hidup. Tahap selanjutnya adalah *selection* menggunakan metode *roulette wheel* untuk menentukan orang tua kromosom mana yang melakukan pemindahan silang atau *crossover*. Pada tahap ini, diperoleh nilai *fitness* pada masing-masing kromosom sebagai penentu apakah sebuah kromosom terpilih sebagai calon orang tua di generasi selanjutnya.

Tahap selanjutnya adalah *crossover* yaitu pindah silang. Kromosom yang terpilih sebagai calon orang tua akan diberikan bilangan random $U(0,1)$. Berikut ini adalah ilustrasi dari pindah silang pada kromosom.

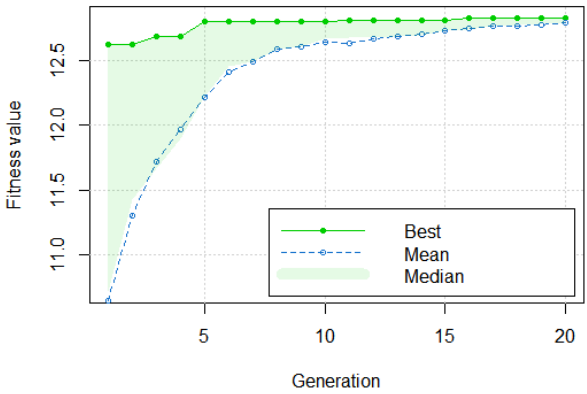
Parent 1	ε_1	γ_1	C_1
Parent 2	ε_2	γ_2	C_2

offspring 1	ε_1	γ_2	C_2
offspring 2	ε_2	γ_1	C_1

Gambar 4.15 Ilustrasi Pindah Silang

Proses pindah silang terjadi jika bila bilangan random bernilai lebih kecil dari peluang pindah silang (P_c = 0,8).

Proses selanjutnya adalah mutasi kromosom dengan peluang mutasi (P_m) sebesar 0,1. Kromosom yang telah melakukan *crossover*, akan dimutasi dengan cara memberikan bilangan random $U(0,1)$ dan jika nilai bilangan random kurang dari P_m, maka salah satu nilai pada kromosom tersebut akan dimutasi dengan bilangan random. Tahap terakhir dari GA *generate* generasi baru sebanyak 20 iterasi hingga diperoleh nilai parameter yang konvergen. Nilai parameter yang terbaik pada generasi sebelumnya disimpan pada proses *elitism*, sehingga di generasi selanjutnya akan diperoleh nilai parameter dengan nilai *fitness* terbaik. Gambar 4.16 menunjukkan bahwa nilai sMAPE sudah konvergen hingga iterasi/generasi ke 20.



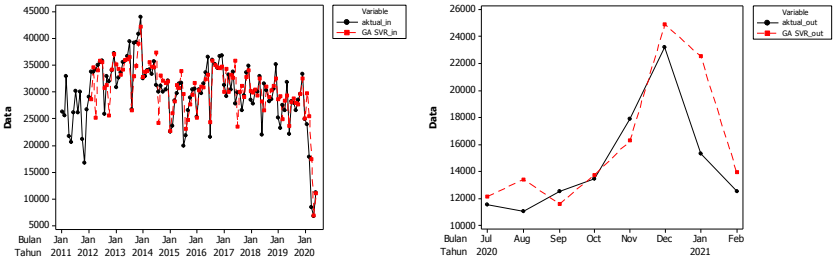
Gambar 4.16 Nilai *Fitness* Konvergen pada Input $f(Y_{t-1}, Y_{t-2})$

Dari rangkaian proses GA yang dilakukan, diperoleh parameter optimum untuk input $f(Y_{t-1,}\, Y_{t-2,})$ adalah C = 47,90; γ = 0,8738; dan ε = 0,0159. Proses optimasi GA dilakukan pada model input lainnya dan nilai optimum parameter dirangkum pada Tabel 4.10 sebagai berikut.

Tabel 4.10 Parameter Optimum GA-SVR

Input	ε	γ	C	**RMSE *in sample***	**RMSE *out sample***	**sMAPE *out Sample***
$f(Y_{t-1,}Y_{t-2})$	0,0159	0,8738	47,90	3595,98	4549,41	24,16
$f(Y_{t-1,}Y_{t-2,}\, Y_{t-3})$	0,0131	0,00046	948,13	4811,94	4643,216	23,56
$f(Y_{t-1,}Y_{t-12,}\, Y_{t-13})$	0,1054	0,0292	54,32	3490,92	4882,032	23,25
$f(Y_{t-1})$	0,1102	0,12179	46,16	4713,91	4852,721	17,21
$f(Y_{t-1,}Y_{t-2,}\, Y_{t-12,}, Y_{t-13})$	**0,0205**	**0,03029**	**53,73**	**3166,40**	**2891,106**	**12,44**

Berdasarkan Tabel 4.10, optimasi GA pada parameter SVR memberikan hasil nilai RMSE dan sMAPE lebih rendah jika dibandingkan dengan parameter sebelum dioptimasi pada Tabel 4.10. Selain itu, model terbaik hasil optimasi GA-SVR adalah input $f\,(Y_{t-1,}\, Y_{t-2,}\, Y_{t-12,}\, Y_{t-13})$ karena nilai kriteria *error* baik *in sample* maupun *out sample* paling rendah diantara model lainnya. Berikut ini adalah visualisasi model terbaik GA-SVR dibandingkan dengan data aktual.



(a) **(b)**

Gambar 4.17 *Time Series Plot* Model GA-SVR Terbaik pada Data **(a)** *In Sample* dan **(b)** O*ut Sample*

Berdasarkan Gambar 4.17a, dapat diketahui bahwa hasil ramalan pada model input $f\ (Y_{t-1,}\ Y_{t-2,}\ Y_{t-12,}\ Y_{t-13})$ memiliki performa yang cukup baik karena mampu menggambarkan pola data *in sample* maupun *out sample*. Jika dibandingkan pada model SVR dan ARIMA, hasil ramalan GA-SVR memberikan hasil yang lebih baik dan *error* yang lebih rendah.

4.5 Peramalan Penjualan Toyota Menggunakan Model Terbaik

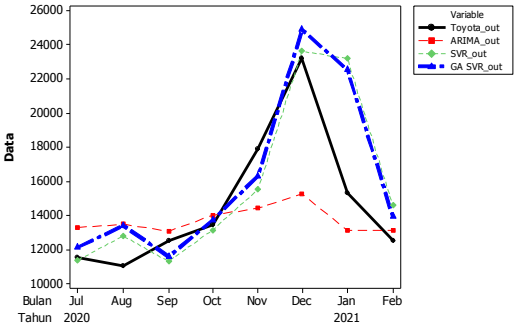
Setelah melakukan pemodelan menggunakan tiga metode yaitu ARIMA, SVR, dan GA-SVR, dilakukan perbandingan nilai RMSE *in sample*, RMSE *out sample* dan sMAPE *out sample* dari model terbaik masing-masing metode untuk mencari model yang paling baik untuk meramalkan penjualan Toyota 12 bulan mendatang. Tabel 4.11 merupakan perbandingan performansi dari ketiga model.

Tabel 4.11 Pemilihan Model Terbaik untuk Meramalkan Penjualan Toyota

Metode	**Model**	**RMSE *in sample***	**RMSE *out sample***	**sMAPE *out Sample***
ARIMA	ARIMA([0,1,[12]) + *outlier*	2941,179	3353,528	15,578
SVR	$f(Y_{t-1}, Y_{t-2}, Y_{t-12}, Y_{t-13})$	3209,847	3101,66	12,57
GA-SVR	$f(Y_{t-1}, Y_{t-2}, Y_{t-12}, Y_{t-13})$	**3166,40**	**2891,106**	**12,44**

Pemilihan model terbaik didasarkan pada nilai RMSE dan sMAPE *out sample* yang paling rendah. Tabel 4.12 memberikan informasi bahwa model terbaik yang terpilih untuk meramalkan penjualan Toyota adalah GA-SVR dengan input $f\ (Y_{t-1}, Y_{t-2}, Y_{t-12}, Y_{t-13})$ karena memiliki nilai RMSE dan sMAPE *out sample* paling rendah dibandingkan model lainnya.

Optimasi GA yang telah dilakukan pada parameter SVR menghasilkan nilai parameter yang lebih optimum ditandai dengan nilai sMAPE yang lebih rendah dibandingkan model SVR. Gambar 4.18 merupakan perbandingan ketiga model pada data *out sample*.



Gambar 4.18 Perbandingan Model Terbaik dengan Data *Out Sample*

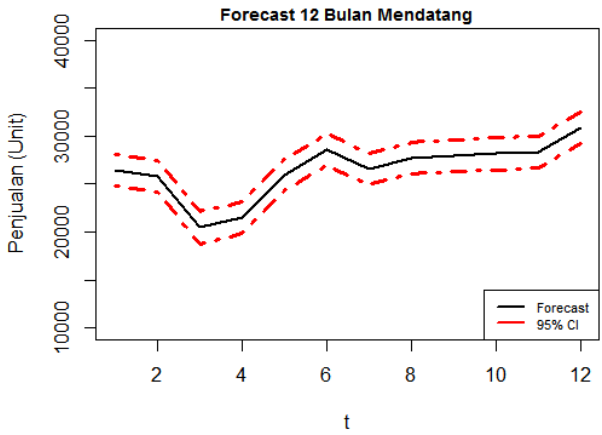
Dari Gambar 4.18, dapat dilihat bahwa hasil ramalan $f\ (Y_{t-1,}\ Y_{t-2,}\ Y_{t-12,}\ Y_{t-13})$ dapat merepresentasikan data aktual secara baik. Sehingga dari penelitian ini dapat disimpulkan bahwa penggunaan GA digunakan untuk optimasi parameter SVR dapat menghasilkan akurasi hasil ramalan yang lebih baik daripada SVR sebelum dioptimasi. Model ini juga tetap memiliki performansi yang baik meskipun terdapat kejadian yang tidak diinginkan seperti pandemi COVID-19.

Model terbaik yang terpilih, yaitu GA-SVR input $f\ (Y_{t-1,}\ Y_{t-2,}\ Y_{t-12,}\ Y_{t-13})$ digunakan untuk meramalkan penjualan mobil Toyota 12 periode kedepan mulai Maret 2021 hingga Februari 2022. Tabel 4.12 merupakan hasil ramalan penjualan Toyota 12 bulan mendatang.

Tabel 4.12 Peramalan Penjualan Toyota 12 Bulan Mendatang

Bulan	Penjualan	Bulan	Penjualan
Maret 2021	26446,05	September 2021	26533,79
April 2021	25801,9	Oktober 2021	27695,83
Mei 2021	20442,34	November 2021	27937,36
Juni 2021	21529,39	Desember 2021	28166,38
Juli 2021	25937,13	Januari 2022	28339,92
Agustus 2021	28617,16	Februari 2022	30821,86

Hasil ramalan pada Tabel 12 menunjukkan bahwa penjualan mulai naik pada bulan Maret 2021, namun turun pada bulan Mei 2021. Berikut merupakan hasil ramalan penjualan Toyota secara visual.



Gambar 4.19 Hasil Ramalan Penjualan Toyota 12 Bulan Kedepan

Gambar 4.19 menunjukkan bahwa hasil ramalan untuk 12 bulan mendatang naik kembali secara *gradual* setelah mengalami penurunan pada Mei 2021 pada bulan Juni 2021 hingga bulan Februari 2022 mencapai 30.809 unit.

(*Halaman ini sengaja dikosongkan*)

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis peramalan penjualan Toyota yang telah dilakukan, dapat ditarik kesimpulan yang menjawab tujuan penelitian sebagai berikut.

1. Penjualan mobil Toyota periode 2011 hingga 2019 cenderung naik dan turun, namun penurunan jumlah penjualan yang terjadi selama tahun 2020 akibat pandemi COVID-19 mencatat rekor yang terburuk selama 10 tahun terakhir. Penurunan yang sangat drastis membuat *market share* Toyota ikut turun. Meskipun begitu, Toyota tetap memimpin pasar mobil di Indonesia dengan mengandalkan penjualan *passenger car* seperti Avanza, Calya, dan Rush.
2. Model terbaik untuk meramalkan penjualan Toyota menggunakan model ARIMA diperoleh yaitu ARIMA(0, 1, [12]) + $\omega I_t^{(T)}$, sedangkan model terbaik pada pemodelan SVR dan GA-SVR diperoleh input $f(Y_{t-1}, Y_{t-2}, Y_{t-12}, Y_{t-13})$.
3. Jika ketiga model dibandingkan, penggunaan GA pada optimasi parameter SVR menghasilkan model dengan performansi terbaik karena memiliki RMSE dan sMAPE pada data *in sample* maupun data *out sample* yang lebih kecil dibandingkan model SVR sebelum dioptimasi. Penggunaan model ARIMA pada pemodelan penjualan Toyota menghasilkan hasil ramalan *in sample* yang cukup akurat namun hasil ramalan pada data *out sample* masih kurang baik. Dengan demikian, model GA-SVR merupakan model yang paling tepat untuk meramalkan penjualan Toyota.
4. Berdasarkan hasil ramalan penjualan Toyota 12 bulan mendatang, dapat diketahui bahwa penjualan mulai dapat beradaptasi menuju normal setelah mengalami penurunan di kala pandemi COVID-19 sepanjang tahun 2020.

5.2 Saran

Agar penelitian ini menjadi lebih baik kedepannya, peneliti merekomendasikan beberapa hal sebagai berikut.

1. Untuk penelitian selanjutnya

   Penelitian selanjutnya disarankan untuk menggunakan metode-metode peramalan lainnya yang cocok dengan data non linier. Selain itu, dapat pula menambahkan variabel prediktor pada model SVR yang tidak terbatas hanya *lag-lag* signifikan pada model ARIMA saja, namun faktor-faktor lain yang mungkin ikut berpengaruh terhadap pernjualan mobil Toyota. Contoh faktor-faktor yang mempengaruhi penjualan mobil Toyota adalah jumlah penjualan mobil merek lain, pertumbuhan ekonomi, harga bahan bakar, dan lain-lain. Pada optimasi parameter, disarankan untuk mencoba jenis optimasi lainnya selain algoritma genetika untuk mendapatkan model yang lebih baik lagi.

2. Untuk ATPM Toyota

   Pihak ATPM dapat menjadikan penelitian ini sebagai informasi mengenai penjualan di masa mendatang. Jika perusahaan berniat untuk meningkatkan penjualan, maka perlu adanya perencanaan yang matang terkait kebutuhan bahan, berapa jumlah unit yang akan diproduksi, dan rencana *budgeting* agar kebutuhan konsumen dapat terpenuhi.

DAFTAR PUSTAKA

A., P., Surya, O., Ardianti, F., F., S., & Dwi, B. (2013). Tinjauan Tentang Keberadaan Agen Tunggal Pemegang Merek (Atpm) Di Industri Otomotif Indonesia. *Privat Law*.

AAF. (2020, Februari). *Number of motor vehicles sold in the ASEAN region 2019, by country*. Retrieved from Statista: https://www.statista.com/statistics/583382/asean-motor-vehicle-sales-by-country/

Ali, S., & A.Smith, K. (2006). On learning algorithm selection for classification. *Applied Soft Computing Vol 6 (2)*, 119-138.

Amari, S., & Wu, S. (1999). Improving support vector machine classifiers by modifying kernel functions. *Neural Networks 12 (6)*, 783-789.

Anonim. (2013). *Apa yang Disebut Agen Tunggal Pemegang Merek?* Retrieved from Bibliotika: https://www.bibliotika.com/2016/02/apa-yang-disebut-agen-tunggal-pemegang.html

Box, G. E., & Cox, D. R. (1964). An Analysis of Transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 211-243.

Box, G., & Jenkins, G. (1970). *Time Seriess Analysis: Forecasting and Control.* San Francisco: Holden-Day.

BPS. (2019). *Statistik Indonesia 2020*. Jakarta: Badan Pusat Statistik.

Cryer, J. D., & Chan, K.-S. (2008). *Time Series Analysis With Applications in R.* New York: Springer Science + Business Media.

Daniel, W. W. (1989). *Applied Nonparametric Statistics.* Wadsworth Pub Co.

Fletcher. (1989). *Practical Methods of Optimization.* New York: John Wiley and Sons.

Gujarati, D. N. (2003). *Basic Econometrics 4th Edition.* New York: McGraw-Hill.

Guo, M., Zhao, Y., & Xiang, J. (2014). Review of object detection methods based on SVM. *Control Decision 29 (02)*, 193-200.

Guo, X., Li, D., & Zhang, A. (2012). Improved support vector machine oil price forecast model based on genetic algorithm optimization parameters. *AASRI Procedia 1*, 525-530.

Harafani, H., & Wahono, R. S. (2015). Optimasi Parameter pada Support Vector Machine Berbasis Algoritma Genetika untuk Estimasi Kebakaran Hutan. *Journal of Intelligent System*.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2016). A Practical Guide to Support Vector Classification.

Kompas. (2020, Agustus 5). *Pertumbuhan Ekonomi RI Minus 5,32 Persen pada Kuartal II-2020*. Retrieved from Kompas.com: https://money.kompas.com/read/2020/08/05/120854826/pertumbuhan-ekonomi-ri-minus-532-persen-pada-kuartal-ii-2020

Lanlan, Z., Juyang, L., Qilin, Z., & Yudong, W. (2015). Using Genetic Algorithm to Optimize Parameters of Support Vector Machine and Its Application in Material Fatigue Life Prediction. *Advances in Natural Science*, 21-26.

Liu, H., Mi, X., Li, Y., Duan, Z., & Xu, Y. (2019). Smart wind speed deep learning based multi-step forecasting model using singular spectrum analysis, convolutional Gated Recurrent Unit network and Support Vector Regression. *Renewable Energy 143*, 842-854.

Ljung, G. M., & Box, G. E. (1978). On a measure of lack of fit in time series models. *Biometrika*, 553-564.

Ma, W., Zhang, X., Xin, Y., & Li, S. (2019). Study on short-term network forecasting based on SVM-MFA. *Journal of Visual Communication and Image Representation 65*.

Machairas, V., Tsangrassoulis, A., & Axarli, K. (2014). Algoritmhs for Optimization of Bulding Design: A Review. *Renewable and Sustainable Energy Reviews (1364)*, 101-112.

Maldonado, S., Gonzales, A., & Crone, S. (2019). Automatic time series analysis for electric load forecasting via support vector regression. *Applied Soft Computing Journal 83*.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis 5th Ed.* New Jersey: John Wiley & Sons.

Muller, A. C., & Guido, S. (2018). *Introduction to Machine Learning with Python.* USA: O'Reilly Media.

Nie, H., Liu, G., Liu, X., & Wang, Y. (2012). Hybrid of ARIMA and SVMs for Short-Term Load Forecasting. *Energy Procedia*, 1455-1460.

Noverianto, R. N. (2018). *Peramalan Penjualan Mobil di Indonesia dengan Model Hybrid ARIMAX-Deep Learning Neural Network dengan Hierarcy Time Series.* Surabaya: Institut Teknologi Sepuluh Nopember.

OICA. (2020, Maret). *Number of passenger cars sold in the Asia Pacific region 2019, by country or region*. Retrieved from Statista: https://www.statista.com/statistics/584904/asia-pacific-passenger-car-sales-by-country/

Ordóñez, C., Lasheras, F. S., Roca-Pardiñas, J., & Juez, F. J. (2019). A hybrid ARIMA-SVM model for the study of the remaining uselful life of aircraft engines. *Journal of Computational and Applied Mathematics*, 184-191.

Pai, P.-F. (2006). System reliability forecasting by support vector machines with genetic algorithms. *Mathematical and Computer Modelling 43*, 262-274.

Pai, P.-F., & Lin, C.-S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega 33*, 497-505.

Ring, M., & Eskofier, B. M. (2016). An approximation of the Gaussian RBF kernel for efficient classification with SVMs. *Pattern Recognition Letters 84*, 107-113.

Schölkopf, B., Smola, A. J., & Bach, F. (2002). *Learning with Kernels.* Massachusetts: MIT Press.

Sharp, T. (2020, Maret 4). *An Introduction to Support Vector Regression (SVR)*. Retrieved from Towards Data Science: https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2

Smola, A. J., & Scholkopf, B. (2004). A Tutorial on Support Vector Regression. *Statistics and Computing*, 199-222.

Suci, K. W. (2017). *Peramalan Curah Hujan Sebagai Pendukung Kalender Tanam Padi di Kabupaten Bojonegoro Menggunakan Metode ARIMA, Support Vector Regression, dan Genetic Algorithm-Support Vector Regression.* Surabaya: Institut Teknologi Sepuluh Nopember.

Tao, H., Al-Sulttani, A. O., Ameen, A. M., Ali, Z. H., Ansari, N. A., Salih, S. Q., & Mostafa, R. R. (2020). Training and Testing Data Division Influence on Hybrid Machine Learnig Model Process: Application of River Flow Forecasting. *Hindawi Complexity*.

Teräsvirta, T., Lin, C. F., & Granger, C. W. (1993). Power of The Neural Network Linearity Test. *Journal of Time Series Analysis*, 159-171.

Thurnhofer-Hemsi, K., López-Rubio, E., Molina-Cabello, M. A., & Najarian, K. (2020). Radial basis function kernel optimization for Support Vector Machine classifiers. *IEEE Transactions on Neural Networks and Learning Systems*.

Tsay, R. S. (1988). Outliers, level shifts, and variance changes in time series. *Journal of Forecasting*, 1-20. Retrieved from IBM Knowledge Center.

Vapnik, V. (1995). *The Nature of Statistics Learning Theory.* New York: Springer.

Vapnik, V. (1998). *Statistical Learning Theory.* USA: Wiley.

Wang, J., Jing, Q. B., & Cao, L. L. (2007). Support vector regression facing multi input and multi output system. *Journal of Tsinghua University 47*, 1737-1741.

Wei, W. W. (2006). *Time Series Analysis: Univariate and Multivariate Methods.* USA: Pearson Education.

Yolandha, F. (2021, Januari 15). *Toyota Puncaki Penjualan Mobil 2020, Ini Tipe Paling Laris*. Retrieved from Republika: https://www.republika.co.id/berita/qmybur370/toyota-puncaki-penjualan-mobil-2020-ini-tipe-paling-laris

Yuan, F.-C. (2012). Parameters Optimization Using Genetic Algorithm in Support Vector Regression for Sales Volume Forecasting. *Applied Mathematics Vol 3*, 1480-1486.

Zhang, G. P. (2003). Time Series Forecasting using a Hybrid ARIMA and Neural Network Model. *Neurocomputing 50*, 159-175.

Zhu, Z., Lin, M., & Zhang, S. (2005). Short-term load forecasting based on wavelet transform and support vector machine. *Microcomputer Application 29 (13)*, 60-64.

LAMPIRAN

Lampiran 1 Data Penjualan Toyota

t	Tahun-bulan	Y_t	t	Tahun-bulan	Y_t
1	11-Jan	26355	31	13-Jul	39497
2	11-Feb	25560	32	13-Aug	26679
3	11-Mar	32943	33	13-Sep	39162
4	11-Apr	21729	34	13-Oct	39334
5	11-May	20582	35	13-Nov	40801
6	11-Jun	26216	36	13-Dec	44029
7	11-Jul	30184	37	14-Jan	32545
8	11-Aug	26151	38	14-Feb	33011
9	11-Sep	30041	39	14-Mar	34120
10	11-Oct	21160	40	14-Apr	34271
11	11-Nov	16771	41	14-May	33333
12	11-Dec	26725	42	14-Jun	35733
13	12-Jan	29010	43	14-Jul	31282
14	12-Feb	33752	44	14-Aug	30045
15	12-Mar	33847	45	14-Sep	31172
16	12-Apr	34270	46	14-Oct	30042
17	12-May	35073	47	14-Nov	30406
18	12-Jun	35618	48	14-Dec	32135
19	12-Jul	35806	49	15-Jan	22555
20	12-Aug	25857	50	15-Feb	23637
21	12-Sep	32928	51	15-Mar	28435
22	12-Oct	32052	52	15-Apr	29713
23	12-Nov	34040	53	15-May	31577
24	12-Dec	37258	54	15-Jun	31746
25	13-Jan	30901	55	15-Jul	19902
26	13-Feb	32617	56	15-Aug	21888
27	13-Mar	33670	57	15-Sep	26514
28	13-Apr	35579	58	15-Oct	28880
29	13-May	35812	59	15-Nov	30513
30	13-Jun	36727	60	15-Dec	30579

Lampiran 1. Data Penjualan Toyota (Lanjutan)

t	Tahun-bulan	Y_t	t	Tahun-bulan	Y_t
61	16-Jan	25381	92	18-Aug	30273
62	16-Feb	30440	93	18-Sep	28304
63	16-Mar	29803	94	18-Oct	28622
64	16-Apr	31626	95	18-Nov	30533
65	16-May	33624	96	18-Dec	35107
66	16-Jun	36564	97	19-Jan	25248
67	16-Jul	21563	98	19-Feb	23226
68	16-Aug	36041	99	19-Mar	27608
69	16-Sep	35137	100	19-Apr	26611
70	16-Oct	34499	101	19-May	31871
71	16-Nov	36755	102	19-Jun	22123
72	16-Dec	36771	103	19-Jul	28179
73	17-Jan	31226	104	19-Aug	28022
74	17-Feb	29281	105	19-Sep	26529
75	17-Mar	33212	106	19-Oct	28495
76	17-Apr	30407	107	19-Nov	29672
77	17-May	33809	108	19-Dec	33420
78	17-Jun	27789	109	20-Jan	24928
79	17-Jul	29868	110	20-Feb	23884
80	17-Aug	29949	111	20-Mar	17787
81	17-Sep	26621	112	20-Apr	8443
82	17-Oct	29321	113	20-May	6727
83	17-Nov	33644	114	20-Jun	11196
84	17-Dec	34888	115	20-Jul	11531
85	18-Jan	28495	116	20-Aug	11057
86	18-Feb	27774	117	20-Sep	12523
87	18-Mar	30293	118	20-Oct	13466
88	18-Apr	30076	119	20-Nov	17908
89	18-May	32979	120	20-Dec	23215
90	18-Jun	22002	121	21-Jan	15292
91	18-Jul	31605	122	21-Feb	12537

Lampiran 2. *Syntax* dan *Output* Uji ADF

```
#Cek stasioneritas
trunc((length(toyota_in)-1)^(1/3))
adf.test(toyota_in)


#DIFFERENCING ORDE 1
dyt_in=diff(toyota_in,lag=1, differences = 1)

#Cek stasioneritas setelah differencing
trunc((length(toyota_in)-1)^(1/3))
adf.test(toyota_in)
```

```
> trunc((length(toyota_in)-1)^(1/3))
[1] 4
> adf.test(toyota_in)

        Augmented Dickey-Fuller Test

data:  toyota_in
Dickey-Fuller = -1.6552, Lag order = 4, p-value = 0.7197
alternative hypothesis: stationary

>
>
> #DIFFERENCING ORDE 1
> dyt_in=diff(toyota_in,lag=1, differences = 1)
>
> #Cek stasioneritas setelah differencing
> trunc((length(dyt_in)-1)^(1/3))
[1] 4
> adf.test(dyt_in)

        Augmented Dickey-Fuller Test

data:  dyt_in
Dickey-Fuller = -7.1377, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(dyt_in) : p-value smaller than printed p-value
```

Lampiran 3. *Syntax* Model ARIMA

```
proc arima data=toyota_in;
identify var=toyota(1);
run;
/* model 1
estimate p=(1) q=(0)
noconstant
method=cls;
outlier maxnum=10;
forecast out=ramalan lead=8 printall;
run;
*/
/* model 2
estimate p=(2) q=(0)
noconstant
method=cls;
outlier maxnum=10;
forecast out=ramalan lead=8 printall;
run;
*/
/* model 3
estimate p=(12) q=(0)
noconstant
method=cls;
outlier maxnum=10;
forecast out=ramalan lead=8 printall;
run;
*/
/* model 4
estimate p=(0) q=(1)
noconstant
method=cls;
outlier maxnum=10;
forecast out=ramalan lead=8 printall;
run;
*/
```

Lampiran 3. *Syntax* Model ARIMA (Lanjutan)

```
/* model 5
estimate p=(0) q=(12)
noconstant
method=cls;
outlier maxnum=10;
forecast out=ramalan lead=8 printall;
run;
*/

/* model 6
estimate p=(0) q=(1,12,13)
noconstant
method=cls;
outlier maxnum=10;
forecast out=ramalan lead=8 printall;
run;
*/

/* model 7
estimate p=(1,12) q=(0)
noconstant
method=cls;
outlier maxnum=10;
forecast out=ramalan lead=8 printall;
run;
*/

/* model 8
estimate p=(1) q=(12)
noconstant
method=cls;
outlier maxnum=10;
forecast out=ramalan lead=8 printall;
run;
*/
```

Lampiran 3. *Syntax* Model ARIMA (Lanjutan)

```
/* model 9
estimate p=(12) q=(1)
noconstant
method=cls;
outlier maxnum=10;
forecast out=ramalan lead=8 printall;
run;
*/

/* model 10
estimate p=(0) q=(1,12)
noconstant
method=cls;
outlier maxnum=10;
forecast out=ramalan lead=8 printall;
run;
*/

proc univariate data=ramalan normal;
var residual;
run;

proc export
data=ramalan
outfile='/home/u58770675/ramalan.csv'
dbms=csv
replace;
run;
```

Lampiran 4. *Output* Model ARIMA

a. ARIMA(1,1,0)

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
AR1,1	-0.34923	0.08888	-3.93	0.0001	1

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	11.72	5	0.0388	-0.072	-0.209	-0.060	-0.139	0.071	0.149
12	37.65	11	<.0001	0.062	-0.144	0.015	-0.149	-0.002	0.395
18	49.23	17	<.0001	-0.195	-0.079	0.030	-0.117	0.080	0.148
24	58.55	23	<.0001	-0.052	-0.088	0.040	-0.148	0.112	0.135

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.942902	Pr < W	0.0001
Kolmogorov-Smirnov	D	0.157805	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.490494	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	2.554274	Pr > A-Sq	<0.0050

b. ARIMA(2,1,0)

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
AR1,1	-0.42115	0.09313	-4.52	<.0001	1
AR1,2	-0.20747	0.09318	-2.23	0.0280	2

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	8.95	4	0.0623	-0.020	-0.076	-0.163	-0.139	0.094	0.118
12	31.39	10	0.0005	0.100	-0.151	-0.031	-0.105	-0.000	0.362
18	39.40	16	0.0010	-0.165	-0.048	-0.042	-0.082	0.091	0.118
24	46.77	22	0.0016	-0.008	-0.111	0.018	-0.126	0.107	0.107

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.939958	Pr < W	<0.0001
Kolmogorov-Smirnov	D	0.1395	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.501908	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	2.655476	Pr > A-Sq	<0.0050

c. ARIMA([12],1,0)

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
AR1,1	0.47765	0.08839	5.40	<.0001	12

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	11.63	5	0.0402	-0.294	-0.038	-0.098	-0.038	0.035	0.002
12	15.69	11	0.1531	0.085	-0.124	0.087	-0.031	0.005	0.038
18	20.12	17	0.2683	-0.139	0.042	0.051	-0.068	0.068	0.027
24	21.99	23	0.5208	-0.053	-0.024	0.054	-0.068	0.047	-0.001

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.979392	Pr < W	0.0784
Kolmogorov-Smirnov	D	0.087166	Pr > D	0.0343
Cramer-von Mises	W-Sq	0.152879	Pr > W-Sq	0.0222
Anderson-Darling	A-Sq	0.924095	Pr > A-Sq	0.0197

d. ARIMA(0,1,1)

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	0.54193	0.08160	6.64	<.0001	1

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	5.94	5	0.3124	0.066	-0.061	-0.074	-0.136	0.064	0.116
12	25.67	11	0.0073	0.072	-0.133	-0.006	-0.113	0.004	0.345
18	33.13	17	0.0108	-0.159	-0.055	-0.006	-0.094	0.074	0.116
24	39.89	23	0.0158	-0.029	-0.091	0.019	-0.130	0.093	0.110

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.939367	Pr < W	<0.0001
Kolmogorov-Smirnov	D	0.141731	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.462491	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	2.510004	Pr > A-Sq	<0.0050

e. ARIMA(0,1,[12])

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	-0.40595	0.09231	-4.40	<.0001	12

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	12.40	5	0.0297	-0.311	-0.046	-0.061	-0.053	0.015	0.030
12	19.52	11	0.0524	0.101	-0.149	0.103	-0.079	0.018	0.087
18	26.48	17	0.0661	-0.149	0.025	0.081	-0.116	0.081	0.056
24	32.19	23	0.0962	-0.065	-0.033	0.062	-0.088	-0.004	0.151

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.979311	Pr < W	0.0772
Kolmogorov-Smirnov	D	0.093794	Pr > D	0.0159
Cramer-von Mises	W-Sq	0.15502	Pr > W-Sq	0.0211
Anderson-Darling	A-Sq	0.913523	Pr > A-Sq	0.0207

f. ARIMA(0,1,[1,12,13])

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	0.45798	0.08459	5.41	<.0001	1
MA1,2	-0.35036	0.10067	-3.48	0.0007	12
MA1,3	0.28501	0.10427	2.73	0.0073	13

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	3.89	3	0.2733	0.052	-0.060	-0.120	-0.086	0.006	0.069
12	7.51	9	0.5844	0.088	-0.096	0.030	-0.086	-0.008	0.060
18	10.54	15	0.7843	-0.051	0.014	0.045	-0.084	0.075	0.071
24	13.14	21	0.9035	-0.052	-0.046	0.011	-0.044	0.022	0.104

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.966776	Pr < W	0.0065
Kolmogorov-Smirnov	D	0.10226	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.235941	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	1.360677	Pr > A-Sq	<0.0050

g. ARIMA(1,1,0)(1,0,0)^{12}

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
AR1,1	-0.30149	0.08043	-3.75	0.0003	1
AR1,2	0.44196	0.08419	5.25	<.0001	12

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	6.71	4	0.1522	-0.043	-0.160	-0.151	-0.053	0.032	0.055
12	8.13	10	0.6162	0.066	-0.079	0.018	-0.009	-0.019	0.009
18	10.42	16	0.8440	-0.025	-0.024	0.035	-0.074	0.071	0.064
24	11.01	22	0.9746	-0.056	-0.018	-0.009	-0.018	0.019	-0.008

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.980682	Pr < W	0.1018
Kolmogorov-Smirnov	D	0.075625	Pr > D	0.1111
Cramer-von Mises	W-Sq	0.116534	Pr > W-Sq	0.0703
Anderson-Darling	A-Sq	0.717904	Pr > A-Sq	0.0621

h. ARIMA(1,1,0)(0,0,1)^{12}

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	-0.40244	0.09507	-4.23	<.0001	12
AR1,1	-0.31786	0.09133	-3.48	0.0007	1

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	6.75	4	0.1498	-0.050	-0.181	-0.097	-0.074	0.013	0.083
12	10.22	10	0.4211	0.078	-0.113	0.043	-0.053	0.023	0.062
18	16.17	16	0.4415	-0.138	0.002	0.067	-0.088	0.082	0.081
24	20.90	22	0.5268	-0.075	-0.042	0.034	-0.087	0.015	0.129

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.978014	Pr < W	0.0594
Kolmogorov-Smirnov	D	0.08369	Pr > D	0.0498
Cramer-von Mises	W-Sq	0.155647	Pr > W-Sq	0.0208
Anderson-Darling	A-Sq	0.941611	Pr > A-Sq	0.0181

i. ARIMA(0,1,1)(1,0,0)^{12}

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	0.44587	0.08520	5.23	<.0001	1
AR1,1	0.43292	0.09461	4.58	<.0001	12

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	3.79	4	0.4353	0.058	-0.061	-0.129	-0.078	0.024	0.043
12	5.51	10	0.8543	0.066	-0.082	0.044	-0.024	-0.008	0.016
18	9.27	16	0.9017	-0.138	-0.003	0.037	-0.034	0.070	0.045
24	10.51	22	0.9811	-0.050	-0.040	0.021	-0.059	0.023	-0.017

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.973678	Pr < W	0.0249
Kolmogorov-Smirnov	D	0.089287	Pr > D	0.0249
Cramer-von Mises	W-Sq	0.190246	Pr > W-Sq	0.0072
Anderson-Darling	A-Sq	1.12508	Pr > A-Sq	0.0061

j. ARIMA(0,1,1)(0,0,1)^{12}

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	0.37143	0.08122	4.57	<.0001	1
MA1,2	-0.36104	0.08728	-4.14	<.0001	12

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	4.31	4	0.3655	-0.025	-0.097	-0.097	-0.104	0.031	0.072
12	8.83	10	0.5478	0.072	-0.115	0.061	-0.040	0.015	0.110
18	18.82	16	0.2783	-0.241	-0.031	0.059	-0.050	0.078	0.070
24	23.58	22	0.3696	-0.052	-0.053	0.040	-0.128	0.023	0.097

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.970579	Pr < W	0.0135
Kolmogorov-Smirnov	D	0.102745	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.249756	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	1.424399	Pr > A-Sq	<0.0050

Lampiran 5. Penguraian Model ARIMA

a. ARIMA(1,1,0)

$$(1-\phi_1 B^1)\left[(1-B)Y_t\right]=0$$

$$(1-\phi_1 B^1)(Y_t - Y_{t-1})=0$$

$$Y_t - Y_{t-1} - \phi_1 Y_{t-1} + \phi_1 Y_{t-2} = 0$$

$$Y_t - Y_{t-1} - \phi_1 (Y_{t-1} - Y_{t-2}) = 0$$

$$Y_t = Y_{t-1} - 0,34923(Y_{t-1} - Y_{t-2})$$

b. ARIMA(2,1,0)

$$(1-\phi_1 B^1)\left[(1-B)Y_t\right]=0$$

$$(1-\phi_1 B^1 - \phi_2 B^2)(Y_t - Y_{t-1})=0$$

$$Y_t - Y_{t-1} - \phi_1 Y_{t-1} + \phi_1 Y_{t-2} - \phi_2 Y_{t-2} + \phi_2 Y_{t-3} = 0$$

$$Y_t - Y_{t-1} - \phi_1 (Y_{t-1} - Y_{t-2}) - \phi_2 (Y_{t-2} - Y_{t-3}) = 0$$

$$Y_t = Y_{t-1} - 0,42115(Y_{t-1} - Y_{t-2}) - 0,20747(Y_{t-2} - Y_{t-3})$$

c. ARIMA([12],1,0)

$$(1-\phi_{12} B^{12})\left[(1-B)Y_t\right]=0$$

$$(1-\phi_{12} B^{12})(Y_t - Y_{t-1})=0$$

$$Y_t - Y_{t-1} - \phi_{12} Y_{t-12} + \phi_{12} Y_{t-13} = 0$$

$$Y_t - Y_{t-1} - \phi_{12} (Y_{t-12} - Y_{t-13}) = 0$$

$$Y_t = Y_{t-1} + 0,47765(Y_{t-12} - Y_{t-13})$$

d. ARIMA(0,1,1)

$$(1-B)Y_t = (1-\theta_1 B)a_t$$

$$Y_t - Y_{t-1} = (1-\theta_1 B)a_t$$

$$Y_t - Y_{t-1} = a_t - \theta_1 a_{t-1}$$

$$Y_t = Y_{t-1} + a_t - 0,54193 a_{t-1}$$

Lampiran 5. Penguraian Model ARIMA (*Lanjutan*)

e. ARIMA(0,1,[12])

$$(1-B)Y_t = (1-\theta_{12}B^{12})a_t$$
$$Y_t - Y_{t-1} = (1-\theta_{12}B^{12})a_t$$
$$Y_t - Y_{t-1} = a_t - \theta_{12}a_{t-12}$$
$$Y_t = Y_{t-1} + a_t + 0,40595a_{t-12}$$

f. ARIMA(0,1,[1,12,13])

$$(1-B)Y_t = (1-\theta_1 B - \theta_{12}B^{12} - \theta_{13}B^{13})a_t$$
$$Y_t - Y_{t-1} = (1-\theta_1 B - \theta_{12}B^{12} - \theta_{13}B^{13})a_t$$
$$Y_t - Y_{t-1} = a_t - \theta_1 a_{t-1} - \theta_{12}a_{t-12} - \theta_{13}a_{t-13}$$
$$Y_t = Y_{t-1} + a_t - 0,45798a_{t-1} + 0,35036a_{t-12} - 0,28501a_{t-13}$$

g. ARIMA(1,1,0)(1,0,0)^{12}

$$(1-\phi_1 B^1)(1-\Phi_1 B^{12})(1-B)Y_t = 0$$
$$(1-\phi_1 B^1)(1-\Phi_1 B^{12})(Y_t - Y_{t-1}) = 0$$
$$(1-\Phi_1 B^{12} - \phi_1 B^1 + \phi_1\Phi_1 B^{13})(Y_t - Y_{t-1}) = 0$$
$$Y_t - Y_{t-1} - \Phi_1 Y_{t-12} + \Phi_1 Y_{t-13} - \phi_1 Y_{t-1} + \phi_1 Y_{t-2} + \phi_1\Phi_1 Y_{t-13} - \phi_1\Phi_1 Y_{t-14} = 0$$
$$Y_t - Y_{t-1} - \phi_1(Y_{t-1} - Y_{t-2}) - \Phi_1(Y_{t-12} - Y_{t-13}) + \phi_1\Phi_1(Y_{t-13} - Y_{t-14}) = 0$$
$$Y_t = Y_{t-1} - 0,30149(Y_{t-1} - Y_{t-2}) + 0,44196(Y_{t-12} - Y_{t-13}) + 0,13324(Y_{t-13} - Y_{t-14})$$

h. ARIMA(1,1,0)(0,0,1)^{12}

$$(1-\phi_1 B^1)(1-B^{12})(1-B)Y_t = (1-\Theta_1 B^{12})a_t$$
$$(1-\phi_1 B^1)(1-B^{12})(Y_t - Y_{t-1}) = (1-\Theta_1 B^{12})a_t$$
$$(1-B^{12} - \phi_1 B^1 + \phi_1 B^{13})(Y_t - Y_{t-1}) = (1-\Theta_1 B^{12})a_t$$
$$Y_t - Y_{t-1} - Y_{t-12} + Y_{t-13} - \phi_1 Y_{t-1} + \phi_1 Y_{t-2} + \phi_1 Y_{t-13} - \phi_1 Y_{t-14} = a_t - \Theta_1 a_{t-12}$$
$$Y_t - Y_{t-1} - (Y_{t-12} - Y_{t-13}) - \phi_1(Y_{t-1} - Y_{t-2}) + \phi_1(Y_{t-13} - Y_{t-14}) = a_t - \Theta_1 a_{t-12}$$
$$Y_t = Y_{t-1} + (Y_{t-1} - Y_{t-2}) - 0,31786(Y_{t-12} - Y_{t-13}) + 0,31786(Y_{t-13} - Y_{t-14}) + a_t - 0,40244a_{t-12}$$

Lampiran 5. Penguraian Model ARIMA (*Lanjutan*)

i. ARIMA(0,1,1)(1,0,0)^{12}

$$(1-\Phi_1 B^{12})(1-B)Y_t = (1-\theta_1 B_t)a_t$$

$$(1-\Phi_1 B^{12})(Y_t - Y_{t-1}) = (1-\theta_1 B_t)a_t$$

$$Y_t + Y_{t-1} - \Phi_1 Y_{t-12} + \Phi_1 Y_{t-13} = a_t - \theta_1 a_{t-1}$$

$$Y_t + Y_{t-1} - \Phi_1 (Y_{t-12} - Y_{t-13}) = a_t - \theta_1 a_{t-1}$$

$$Y_t = Y_{t-1} + 0,44587(Y_{t-12} - Y_{t-13}) + a_t - 0,43292 a_{t-1}$$

j. ARIMA(0,1,1)(0,0,1)^{12}

$$(1-B^{12})(1-B)Y_t = (1-\theta_1 B)(1-\Theta_1 B^{12})a_t$$

$$(1-B^{12})(Y_t - Y_{t-1}) = (1-\theta_1 B)(1-\Theta_1 B^{12})a_t$$

$$Y_t - Y_{t-1} - Y_{t-12} + Y_{t-13} = (1-\Theta_1 B^{12} - \theta_1 B + \theta_1 \Theta_1 B^{13})a_t$$

$$Y_t - Y_{t-1} - (Y_{t-12} - Y_{t-13}) = a_t - \Theta_1 a_{t-12} - \theta_1 a_{t-1} + \theta_1 \Theta_1 a_{t-13}$$

$$Y_t = Y_{t-1} + (Y_{t-12} - Y_{t-13}) + a_t - \Theta_1 a_{t-12} - \theta_1 a_{t-1} + \theta_1 \Theta_1 a_{t-13}$$

$$Y_t = Y_{t-1} + (Y_{t-12} - Y_{t-13}) + a_t + 0,36104 a_{t-12} - 0,37143 a_{t-1} - 0,1341 a_{t-13}$$

Lampiran 6. Komponen *Outlier* Masing-masing Model ARIMA

Model	Observasi	Tipe *Outlier*
ARIMA(1,1,0)	67	*Additive*
	32	*Addditive*
	112	*Shift*
	11	*Additive*
	55	*Shift*
	3	*Additive*
	90	*Additive*
	37	*Shift*
	20	*Additive*
	96	*Additive*
ARIMA(2,1,0)	67	*Additive*
	32	*Additive*
	111	*Shift*
	11	*Additive*
	90	*Additive*
	55	*Shift*
	37	*Shift*
	3	*Additive*
	20	*Additive*
	96	*Additive*
ARIMA([12],1,0)	67	*Additive*
	32	*Additive*
	3	*Additive*
	55	*Shift*
	11	*Additive*
	113	*Additive*
	112	*Shift*
	90	*Additive*
	20	*Additive*
	10	*Shift*

Lampiran 6. Komponen *Outlier* Masing-masing Model ARIMA

Model	Observasi	Tipe *Outlier*
ARIMA(0,1,1)	112	*Shift*
	67	*Additive*
	32	*Additive*
	11	*Additive*
	37	*Shift*
	90	*Additive*
	3	*Additive*
	20	*Additive*
	55	*Shift*
	109	*Shift*
ARIMA(0,1,[12])	67	*Additive*
	32	*Additive*
	3	*Additive*
	101	*Additive*
	90	*Additive*
	11	*Additive*
	20	*Additive*
	55	*Shift*
	112	*Shuft*
	9	*Additive*
ARIMA(0,1,[1,12,13])	111	*Shift*
	67	*Additive*
	11	*Additive*
	32	*Additive*
	3	*Additive*
	55	*Shift*
	101	*Additive*
	20	*Additive*
	14	*Shift*
	112	*Shift*

Lampiran 6. Komponen *Outlier* Masing-masing Model ARIMA

Model	Observasi	Tipe *Outlier*
ARIMA(1,1,0)(0,0,1)^{12}	67	*Additive*
	112	*Shift*
	32	*Additive*
	11	*Additive*
	3	*Additive*
	55	*Shift*
	20	*Additive*
	101	*Additive*
	90	*Additive*
	111	*Shift*
ARIMA(0,1,1)(1,0,0)^{12}	112	*Shift*
	67	*Additive*
	11	*Additive*
	32	*Additive*
	3	*Additive*
	55	*Shift*
	20	*Additive*
	111	*Shift*
	114	*Additive*
	57	*shift*
ARIMA(0,1,1)(0,0,1)^{12}	67	*Additive*
	112	*Shift*
	32	*Additive*
	11	*Additive*
	3	*Additive*
	20	*Additive*
	55	*Additive*
	56	*Additive*
	90	*Additive*
	102	*Additive*

Lampiran 7. *Syntax* Model ARIMA *Outlier*

```
data outlier;
input toyota;
datalines;
26355
25560
32943
21729
:
28495
29672
33420
24928
23884
17787
8443
6727
11196
.
.
.
.
.
.
.
.
;
data outlier;
set outlier;
/* to be changed*/
if _n_= 67 then ao67 =1;else ao67=0;
if _n_= 32 then ao32 =1;else ao32=0;
if _n_>= 111 then ls111 =1;else ls111=0;
if _n_= 11 then ao11 =1;else ao11=0;
if _n_= 90 then ao90 =1;else ao90=0;
if _n_>=55 then ls55 =1;else ls55=0;
if _n_>= 37 then ls37 =1;else ls37=0;
if _n_= 3 then ao3 =1;else ao3=0;
if _n_= 20 then ao20 =1;else ao20=0;
if _n_= 96 then ao96 =1;else ao96=0;
```

Lampiran 7. *Syntax* Model ARIMA *Outlier* (Lanjutan)

```
/* to be changed*/
proc arima data=outlier;
identify var=toyota(1) crosscorr=(ao67(1) ao32(1) ls111(1)
ao11(1) ao90(1) ls55(1) ls37(1) ao3(1) ao20(1) ao96(1));
run;

/* to be changed*/
estimate p=(1,2) q=0 noint
input=(ao67 ao32 ls111 ao11 ao90 ls55 ls37 ao3 ao20 ao96)
method=cls;
forecast out=ramalan lead=8 printall;
run;

proc univariate data=ramalan normal;
var residual;
run;

proc export
data=ramalan
outfile='/home/u58770675/ramoutlier.csv'
dbms=csv
replace;
run:
```

Lampiran 8. *Output* Model ARIMA *Outlier*

a. ARIMA(0,1,[12])

Conditional Least Squares Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MA1,1	-0.40206	0.10919	-3.68	0.0004	12	toyota	0
NUM1	-14380.2	2111.3	-6.81	<.0001	0	ao67	0
NUM2	-12377.7	2190.1	-5.65	<.0001	0	ao32	0
NUM3	9337.4	2004.3	4.66	<.0001	0	ao3	0
NUM4	7952.9	2074.4	3.83	0.0002	0	ao101	0
NUM5	-6670.1	2147.6	-3.11	0.0025	0	ao90	0
NUM6	-7005.5	2004.3	-3.50	0.0007	0	ao11	0
NUM7	-8199.3	2229.1	-3.68	0.0004	0	ao20	0
NUM8	-8985.6	3087.1	-2.91	0.0044	0	ls55	0
NUM9	-8764.1	3107.3	-2.82	0.0058	0	ls112	0
NUM10	6376.6	2046.1	3.12	0.0024	0	ao9	0

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	3.45	5	0.6316	-0.081	-0.095	-0.098	0.051	-0.003	-0.038
12	8.94	11	0.6274	0.084	-0.035	0.005	-0.135	-0.030	0.127
18	13.31	17	0.7152	-0.136	-0.090	-0.023	-0.011	0.071	-0.032
24	27.82	23	0.2228	-0.034	0.005	-0.010	-0.094	-0.066	0.292

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.960287	Pr < W	0.0020
Kolmogorov-Smirnov	D	0.081442	Pr > D	0.0655
Cramer-von Mises	W-Sq	0.163769	Pr > W-Sq	0.0168
Anderson-Darling	A-Sq	1.117471	Pr > A-Sq	0.0064

b. ARIMA(0,1,[1,12,13])

Conditional Least Squares Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MA1,1	0.67625	0.07100	9.52	<.0001	1	toyota	0
MA1,2	-0.33682	0.08813	-3.82	0.0002	12	toyota	0
MA1,3	0.66057	0.08180	8.08	<.0001	13	toyota	0
NUM1	-11251.0	2857.5	-3.94	0.0002	0	ls111	0
NUM2	-11690.6	2220.2	-5.27	<.0001	0	ao67	0
NUM3	-9756.2	2038.9	-4.79	<.0001	0	ao11	0
NUM4	-12088.2	2357.0	-5.13	<.0001	0	ao32	0
NUM5	8818.7	2032.1	4.34	<.0001	0	ao3	0
NUM6	-7327.3	1619.2	-4.53	<.0001	0	ls55	0
NUM7	6542.0	2230.6	2.93	0.0042	0	ao101	0
NUM8	-7346.0	2260.0	-3.25	0.0016	0	ao20	0
NUM9	9104.0	1316.5	6.92	<.0001	0	ls14	0
NUM10	-7995.5	2919.2	-2.74	0.0073	0	ls112	0

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2.87	3	0.4114	0.079	-0.032	-0.113	0.055	0.035	-0.013
12	9.74	9	0.3718	0.049	-0.040	-0.034	-0.192	0.010	0.110
18	13.20	15	0.5869	0.059	-0.118	-0.011	-0.056	-0.006	-0.074
24	21.94	21	0.4028	-0.090	0.010	-0.114	-0.031	0.086	0.177

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.9905	Pr < W	0.6231
Kolmogorov-Smirnov	D	0.054637	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.046649	Pr > W-Sq	>0.2500
Anderson-Darling	A-Sq	0.296742	Pr > A-Sq	>0.2500

c. ARIMA(1,1,0)(0,0,1)^{12}

Conditional Least Squares Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MA1,1	-0.39740	0.10444	-3.81	0.0002	12	toyota	0
AR1,1	-0.20966	0.10661	-1.97	0.0520	1	toyota	0
NUM1	-13472.6	2341.8	-5.75	<.0001	0	ao67	0
NUM2	-8924.1	3127.3	-2.85	0.0052	0	ls112	0
NUM3	-12020.7	2423.2	-4.96	<.0001	0	ao32	0
NUM4	9192.3	2210.6	4.16	<.0001	0	ao3	0
NUM5	-8498.4	2324.8	-3.66	0.0004	0	ao11	0
NUM6	-8773.1	3017.5	-2.91	0.0045	0	ls55	0
NUM7	-7444.3	2420.8	-3.08	0.0027	0	ao20	0
NUM8	7046.8	2274.7	3.10	0.0025	0	ao101	0
NUM9	-5961.4	2332.4	-2.56	0.0121	0	ao90	0
NUM10	-7739.7	3140.8	-2.46	0.0154	0	ls111	0

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2.25	4	0.6892	-0.019	-0.105	-0.069	0.004	0.047	-0.026
12	10.02	10	0.4390	0.085	0.042	-0.021	-0.196	-0.021	0.115
18	19.59	16	0.2391	-0.147	-0.218	0.009	-0.013	0.059	-0.023
24	33.86	22	0.0508	-0.057	0.004	-0.057	-0.084	-0.020	0.290

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.969609	Pr < W	0.0112
Kolmogorov-Smirnov	D	0.057214	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.09704	Pr > W-Sq	0.1251
Anderson-Darling	A-Sq	0.753158	Pr > A-Sq	0.0487

Lampiran 9. Pengutaian Model ARIMA *Outlier*

a. ARIMA(0,1,[12])

$$(1-B)Y_t = (1-\theta_{12}B^{12})a_t + \omega I_t^{(T)}$$

$$Y_t - Y_{t-1} = (1-\theta_{12}B^{12})a_t + \omega I_t^{(T)}$$

$$Y_t - Y_{t-1} = a_t - \theta_{12}a_{t-12} + \omega I_t^{(T)}$$

$$Y_t = Y_{t-1} + a_t + 0,40206a_{t-12} + \omega_{AO}I_{67}^{(67)} + \omega_{AO}I_{32}^{(32)} + \omega_{AO}I_3^{(3)} + \omega_{AO}I_{101}^{(101)}$$
$$+ \omega_{AO}I_{90}^{(90)} + \omega_{AO}I_{11}^{(11)} + \omega_{AO}I_{20}^{(20)} + \omega_{LS}I_{55}^{(55)} + \omega_{LS}I_{112}^{(112)} + \omega_{AO}I_9^{(9)}$$

$$Y_t = Y_{t-1} + a_t + 0,40206a_{t-12} - 14380,2I_{67}^{(67)} - 12377,7I_{32}^{(32)} + 9337,4I_3^{(3)} + 7952,9I_{101}^{(101)}$$
$$- 6670,1I_{90}^{(90)} - 7005,5I_{11}^{(11)} - 8199,3I_{20}^{(20)} - 8764,1I_{55}^{(55)} - 8764,1I_{112}^{(112)} + 6376,6I_9^{(9)}$$

b. ARIMA(0,1,[1,12,13])

$$(1-B)Y_t = (1-\theta_1 B - \theta_{12}B^{12} - \theta_{13}B^{13})a_t + \omega I_t^{(T)}$$

$$Y_t - Y_{t-1} = (1-\theta_1 B - \theta_{12}B^{12} - \theta_{13}B^{13})a_t + \omega I_t^{(T)}$$

$$Y_t - Y_{t-1} = a_t - \theta_1 a_{t-1} - \theta_{12}a_{t-12} - \theta_{13}a_{t-13} + \omega I_t^{(T)}$$

$$Y_t = Y_{t-1} + a_t - 0,67625a_{t-1} + 0,33682a_{t-12} - 0,66057a_{t-13}$$

$$Y_t = Y_{t-1} + a_t - 0,67625a_{t-1} + 0,33682a_{t-12} - 0,66057a_{t-13} + \omega_{LS}I_{111}^{(111)} + \omega_{AO}I_{67}^{(67)} +$$
$$\omega_{AO}I_{11}^{(11)} + \omega_{AO}I_{32}^{(32)} + \omega_{AO}I_3^{(3)} + \omega_{LS}I_{55}^{(55)} + \omega_{AO}I_{101}^{(101)} + \omega_{AO}I_{20}^{(20)} + \omega_{LS}I_{14}^{(14)}$$
$$+ \omega_{AO}I_{112}^{(112)}$$

$$Y_t = Y_{t-1} + a_t - 0,67625a_{t-1} + 0,33682a_{t-12} - 0,66057a_{t-13} - 11251I_{111}^{(111)} - 11690,6I_{67}^{(67)} +$$
$$9756,2I_{11}^{(11)} - 12088,2I_{32}^{(32)} + 8818,7I_3^{(3)} - 7327,3I_{55}^{(55)} + 6542I_{101}^{(101)} - 7346I_{20}^{(20)}$$
$$+ 9104I_{14}^{(14)} - 7995,5I_{112}^{(112)}$$

c. ARIMA(1,1,0)(0,0,1)^{12}

$$(1-\phi_1 B^1)(1-B^{12})(1-B)Y_t = (1-\Theta_1 B^{12})a_t + \omega I_t^{(T)}$$

$$(1-\phi_1 B^1)(1-B^{12})(Y_t - Y_{t-1}) = (1-\Theta_1 B^{12})a_t + \omega I_t^{(T)}$$

$$(1-B^{12} - \phi_1 B^1 + \phi_1 B^{13})(Y_t - Y_{t-1}) = (1-\Theta_1 B^{12})a_t + \omega I_t^{(T)}$$

$$Y_t - Y_{t-1} - (Y_{t-12} - Y_{t-13}) - \phi_1(Y_{t-1} - Y_{t-2}) + \phi_1(Y_{t-13} - Y_{t-14}) = a_t - \Theta_1 a_{t-12}$$

$$Y_t = Y_{t-1} + (Y_{t-1} - Y_{t-2}) - 0,31786(Y_{t-12} - Y_{t-13}) + 0,31786(Y_{t-13} - Y_{t-14}) + a_t - 0,40244a_{t-12} + \omega I_t^{(T)}$$

Lampiran 9. Penguraian Model ARIMA *Outlier* (*Lanjutan*)

$$Y_t = Y_{t-1} + (Y_{t-1} - Y_{t-2}) - 0,31786(Y_{t-12} - Y_{t-13}) + 0,31786(Y_{t-13} - Y_{t-14}) + a_t - 0,40244a_{t-12} +$$
$$\omega_{AO} I_{67}^{(67)} + \omega_{LS} I_{112}^{(112)} + \omega_{AO} I_{32}^{(32)} + \omega_{AO} I_{3}^{(3)} + \omega_{AO} I_{11}^{(11)} + \omega_{LS} I_{55}^{(55)}$$
$$+ \omega_{AO} I_{20}^{(20)} + \omega_{AO} I_{101}^{(67)} + \omega_{AO} I_{90}^{(90)} + \omega_{LS} I_{111}^{(111)}$$

$$Y_t = Y_{t-1} + (Y_{t-1} - Y_{t-2}) - 0,31786(Y_{t-12} - Y_{t-13}) + 0,31786(Y_{t-13} - Y_{t-14}) + a_t - 0,40244a_{t-12} +$$
$$-13472,6I_{67}^{(67)} - 8924,1I_{112}^{(112)} - 12020,7I_{32}^{(32)} + 9192,3I_{3}^{(3)} - 8498,4I_{11}^{(11)} - 8773,1I_{55}^{(55)}$$
$$-7444,3I_{20}^{(20)} + 7046,8I_{101}^{(67)} - 5961,4I_{90}^{(90)} - 7739,7I_{111}^{(111)}$$

Lampiran 10. *Syntax* dan *Output* Uji *Teräsvirta*

```
library(tseries)
data=read.csv("C:/Users/user/OneDrive/Documents/TA/Data
udah rapi/ujitrvt.csv",header=TRUE,sep=",")
data
y=data$yt
x=data.frame(data$yt.1,data$yt.2,data$yt.4,data$yt.12)
terasvirta.test(x, y, type = c("Chisq","F"),
                scale = TRUE)
```

```
        Teraesvirta Neural Network Test

data:  x and y
X-squared = 84.155, df = 30, p-value = 4.91e-07
```

Lampiran 11. *Syntax SVR*

```
library(e1071)
library(tseries)
library(caret)
library(lattice)
library(kernlab)
data=read.csv("C:/Users/user/OneDrive/Documents/TA/Analisi
s TA New/SVR/svrmodel.csv",header=T,sep=",")

#to be changed
train=data[c(1:112),]
test=data[-c(1:112),]

#SVR (to be changed)
tune.out <- tune(svm, yt~., data = train, kernel =
"radial",
                 ranges = list(epsilon=0.1, cost = 2^10,
                               gamma = 2^-1))
tunedmodel=tune.out$best.model
tunedmodel
forecast.in=predict(tunedmodel,train)
forecast.out=predict(tunedmodel,test)
forecast1=(forecast.in)
forecast2=(forecast.out)
#Kriteria Error
testing=((test[,1]))
training=((train[,1]))
isi1=testing-forecast2
isi2=training-forecast1
isi3=training-mean(training)

rmse.in=sqrt(mean(isi2^2))
rmse.out=sqrt(mean(isi1^2))
smape.out=(sum(abs(testing-
forecast2)/((abs(testing)+abs(forecast2))/2))/length(testi
ng))*100

rmse.in
rmse.out
smape.out
```

Lampiran 12. *Syntax* GA-SVR

```
library(e1071)
library(GA)
library(forecast)
data=read.csv("C:/Users/user/OneDrive/Documents/TA/Analisis TA
New/SVR/svrmodel4.csv",header=T,sep=",")
#to be changed
train=data[c(1:112),]
test=data[-c(1:112),]

#Fitness Function (to be maximized)
#Parameter vector x is: (cost, gamma, epsilon)
fitnessFunc = function(x){
  #Retrieve the SVM parameters
  cost = x[1]
  gamma = x[2]
  epsilon = x[3]
  #Train
  model =
svm(yt~.,data=train,cost=cost,gamma=gamma,epsilon=epsilon,type="e
ps-regression",kernel="radial")
  #Test
  smape = mean(abs((predict(model, newdata=train) -
train$yt)/((abs(predict(model,newdata=train))+abs(train$yt))/2)))
  return(1/smape)
}

#Range ofparameter values to be changed
#Parameters are: (cost, gamma, epsilon)
#input 1done
theta_min = c(cost = 20, gamma= 0.01, epsilon =0.01)
theta_max = c(cost = 50, gamma = 0.9, epsilon = 0.15)

result = ga(type = "real-valued", fitness = fitnessFunc,
lower=theta_min,upper=theta_max,seed = 123, elitism = 5, maxiter
= 20, popSize = 100, run = 30, pcrossover=0.8,pmutation =
0.01,parallel = parallel::detectCores())
summary(result)
plot(result)
```

Lampiran 11. Syntax GA-SVR (*Lanjutan*)

```
#to be changed
modpred=svm(yt~.,data=train,cost=56.19923,gamma=0.0299922,epsilo
n=0.01980299)

forecast.in=predict(modpred,train)
forecast.out=predict(modpred,test)
forecast1=forecast.in
forecast2=forecast.out
zt=(test[,1])
xt=(train[,1])
isi1=xt-forecast1
isi2=zt-forecast2
rmse.in=sqrt(mean(isi1^2))
rmse.out=sqrt(mean(isi2^2))
smape.out=(sum(abs(zt-
forecast2)/((abs(zt)+abs(forecast2))/2))/length(zt))*100

rmse.in
rmse.out
smape.out
```

Lampiran 13. *Output GA*

a. Input 1

```
-- Genetic Algorithm -------------------

GA settings:
Type                  =  real-valued
Population size       =  100
Number of generations =  20
Elitism               =  5
Crossover probability =  0.8
Mutation probability  =  0.01
Search domain =
      cost gamma epsilon
lower   20  0.01    0.01
upper   50  0.90    0.15

GA results:
Iterations             = 20
Fitness function value = 12.82769
Solution =
        cost     gamma    epsilon
[1,] 47.8059 0.8738328 0.01594115
```

b. Input 2

```
-- Genetic Algorithm -------------------

GA settings:
Type                  =  real-valued
Population size       =  100
Number of generations =  20
Elitism               =  5
Crossover probability =  0.8
Mutation probability  =  0.01
Search domain =
       cost        gamma epsilon
lower  1024 0.0002441406    0.01
upper 32768 0.0004882812    0.15

GA results:
Iterations             = 20
Fitness function value = 8.352621
Solution =
         cost        gamma    epsilon
[1,] 30799.49 0.0004641684 0.02020093
```

c. Input 3

```
-- Genetic Algorithm -------------------

GA settings:
Type                  =  real-valued
Population size       =  100
Number of generations =  20
Elitism               =  5
Crossover probability =  0.8
Mutation probability  =  0.01
Search domain =
       cost     gamma epsilon
lower   10 0.0078125     0.1
upper   60 0.0312500     0.2

GA results:
Iterations             = 20
Fitness function value = 11.6673
Solution =
         cost      gamma   epsilon
[1,] 54.32345 0.02925738 0.1054629
```

d. Input 4

```
-- Genetic Algorithm -------------------

GA settings:
Type                  =  real-valued
Population size       =  100
Number of generations =  20
Elitism               =  5
Crossover probability =  0.8
Mutation probability  =  0.01
Search domain =
      cost     gamma epsilon
lower   10 0.0078125    0.01
upper   60 0.0312500    0.15

GA results:
Iterations             = 20
Fitness function value = 12.94538
Solution =
         cost      gamma    epsilon
[1,] 53.73739 0.03029315 0.02059771
```

e. Input 5

```
-- Genetic Algorithm -------------------

GA settings:
Type                  =  real-valued
Population size       =  100
Number of generations =  20
Elitism               =  5
Crossover probability =  0.8
Mutation probability  =  0.01
Search domain =
      cost  gamma epsilon
lower   10 0.0625    0.10
upper   50 0.1250    0.15

GA results:
Iterations             = 20
Fitness function value = 8.363336
Solution =
         cost     gamma  epsilon
[1,] 46.16816 0.1217955 0.110236
```

Lampiran 14. Pernyataan Keaslian Data

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika FSAD ITS:

Nama : Dessyana Ratna Widyasari

NRP : 06211940005008

menyatakan bahwa data yang digunakan dalam Tugas Akhir ini merupakan data sekunder yang diambil dari publikasi yaitu:

Sumber : gaikindo.or.id

Keterangan : Data *retail sales* mobil berdasarkan *brand* periode 2011 hingga 2021.

Surat pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data, maka saya siap menerima sanksi sesuai aturan yang berlaku.

Mengetahui,

Pembimbing Tugas Akhir

Irhamah, M.Si., Ph.D
NIP.19780406 200112 2 002

Surabaya, 4 Juli 2021

Dessyana Ratna W
NRP.06211940005008

BIODATA PENULIS

Penulis yang dilahirkan di Surabaya, 03 Desember 1996 memiliki nama lengkap Dessyana Ratna Widyasari. Pendidikan formal yang ditempuh penulis adalah SDN Cilandak Barat 06 Pagi Jakarta Selatan, SMPN 85 Jakarta Selatan, SMP Negeri 12 Surabaya, dan SMAN 15 Surabaya. Penulis diterima di Departemen Statistika Bisnis (Prodi DIII) dan melanjutkan ke jenjang S-1 Statistika pada tahun 2019. Semasa kuliah penulis aktif dalam beberapa organisasi dan jabatan yang diemban diantaranya, Staff Media Informasi HIMADATA-ITS (2016-2017), Staff Divisi Event UKM Fotografi ITS Surabaya (2016-2017), Staf Ahli Divisi *Hunting* UKM Fotografi ITS Surabaya (2017-2018), dan Ketua Biro Kampanye Kreatif Departemen Kominfo HIMADATA-ITS. Selain itu, penulis pernah aktif di beberapa kegiatan seperti Pekan Raya Statistika ITS, ITS Goes Global, dan Commtech Integrated Intiative, dan beberapa kegiatan fotografi di dalam ITS maupun di luar ITS. Segala kritik dan saran yang membangun, serta bagi pembaca yang ingin berdiskusi lebih lanjut dengan penulis mengenai segala sesuatu yang berkaitan dengan Tugas Akhir ini dapat disampaikan melalui nomor HP: 081554604996 dan email: dessyana15@mhs.statistika.its.ac.id