

**PENERAPAN MODEL *HYBRID AUTOREGRESSIVE
INTEGRATED MOVING AVERAGE* DENGAN *SUPPORT
VECTOR REGRESSION (ARIMA-SVR)* PADA PERAMALAN
PENJUALAN KLINKER SEMEN DI PT X**

SKRIPSI

oleh:

**DEFIT FITRI YULIANA
175090500111004**



**PROGRAM STUDI SARJANA STATISTIKA
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2021**





(Halaman ini sengaja dikosongkan)

**PENERAPAN MODEL *HYBRID AUTOREGRESSIVE
INTEGRATED MOVING AVERAGE* DENGAN *SUPPORT
VECTOR REGRESSION (ARIMA-SVR)* PADA PERAMALAN
PENJUALAN KLINKER SEMEN DI PT X**

SKRIPSI

oleh:

**DEFIT FITRI YULIANA
175090500111004**



**PROGRAM STUDI SARJANA STATISTIKA
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2021**





(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN SKRIPSI

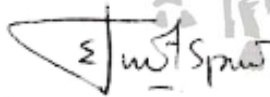
PENERAPAN MODEL *HYBRID AUTOREGRESSIVE INTEGRATED MOVING AVERAGE* DENGAN *SUPPORT VECTOR REGRESSION* (ARIMA-SVR) PADA PERAMALAN PENJUALAN KLINKER SEMEN DI PT X

Oleh:

**DEFIT FITRI YULIANA
175090500111004**

Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 5 Juli 2021
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Statistika

Dosen Pembimbing



Dr. Eni Sumarminingsih, S.Si., MM.
NIP. 197705152002122009

Mengetahui,
Ketua Jurusan Statistika
Fakultas MIPA
Universitas Brawijaya



Rahma Fitriani, S.Si., M.Sc., Ph.D.
NIP. 197603281999032001





(Halaman ini sengaja dikosongkan)

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

NAMA : Defit Fitri Yuliana

NIM : 175090500111004

PROGRAM STUDI : Statistika

SKRIPSI BERJUDUL :

PENERAPAN MODEL *HYBRID AUTOREGRESSIVE INTEGRATED MOVING AVERAGE* DENGAN *SUPPORT VECTOR REGRESSION* (ARIMA-SVR) PADA PERAMALAN PENJUALAN KLINKER SEMEN DI PT X

Dengan ini menyatakan bahwa:

1. Isi dari skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termasuk di isi dan tertulis pada daftar pustaka dalam Skripsi ini.
2. Apabila di kemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung risiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 5 Juli 2021

Yang menyatakan,



Defit Fitri Yuliana

NIM. 175090500111004



(Halaman ini sengaja dikosongkan)

PENERAPAN MODEL *HYBRID AUTOREGRESSIVE INTEGRATED MOVING AVERAGE* DENGAN *SUPPORT VECTOR REGRESSION* (ARIMA-SVR) PADA PERAMALAN PENJUALAN KLINKER SEMEN DI PT X

ABSTRAK

Peramalan deret waktu dapat dilakukan dengan berbagai metode statistika, salah satu metode statistika yang sering digunakan adalah *Autoregressive Integrated Moving Average* (ARIMA) yang merupakan model linier. Model linier belum cukup jika masih ada struktur korelasi linier yang tersisa di residual sehingga model tersebut masih belum cukup. Oleh karena itu, dalam penelitian ini agar diperoleh akurasi yang baik, dilakukan penerapan model peramalan *hybrid* ARIMA-SVR untuk peramalan penjualan karena peramalan penjualan merupakan bagian penting dalam perencanaan produksi. Penentuan model ARIMA berdasarkan identifikasi pola data dan stasioneritas data terhadap ragam maupun rata-rata. Adapun pemodelan SVR menggunakan input residual dari model linier yang diperoleh dan fungsi kernel yang digunakan yaitu RBF. Terdapat beberapa parameter dalam fungsi kernel yang perlu dioptimalisasi di mana dalam penelitian ini menggunakan metode optimalisasi *grid search*. Model linier yang digunakan dalam penelitian ini yaitu model ARIMA(1,1,0), sehingga akan diperoleh model *hybrid* ARIMA(1,1,0)-SVR. Dari model *hybrid* tersebut menghasilkan nilai MAPE kurang dari 10% yaitu sebesar 7.56% pada data *training* dan 9.02% pada data *testing*. Model *hybrid* ARIMA(1,1,0)-SVR pada data *training* maupun data *testing* memiliki nilai MAPE lebih kecil daripada model tunggal (ARIMA). Oleh karena itu, peramalan penjualan klinker menggunakan model *hybrid* memiliki akurasi lebih baik daripada model tunggal.

Kata kunci: ARIMA-SVR, *hybrid*, *testing*, *training*





(Halaman ini sengaja dikosongkan)

IMPLEMENTATION OF HYBRID AUTOREGRESSIVE INTEGRATED MOVING AVERAGE WITH SUPPORT VECTOR REGRESSION MODEL (ARIMA-SVR) FOR SALE'S FORECASTING OF KLINKER IN PT X

ABSTRACT

Time series forecasting use various statistical. One of the statistical methods that is often used for forecasting is the Autoregressive Integrated Moving Average (ARIMA) that produces a linear model. A linear model is not sufficient if there are still linear correlation structures left in the residuals. So in this research, to get good accuracy use hybrid ARIMA-SVR model for sales forecasting because sales forecasting is an important part of production planning. The ARIMA model was form based on the identification of the data pattern and the stationarity of the variety and average. The SVR modeling uses residual input from the linear model and the kernel function used is RBF. There are several parameters in the kernel function that need to be optimized, which in this study uses the grid search optimization method. The linear model used in this study are ARIMA(1,1,0) so we will get the ARIMA(1,1,0)-SVR model. From the hybrid model, the MAPE value is less than 10%, which is 7.56% for training data and 9.02% for testing data. The ARIMA(1,1,0)-SVR model on the training data and testing data has a smaller MAPE value than the single model (ARIMA). Therefore, klinker sales forecasting using the hybrid model has better accuracy than the single model.

Keywords: ARIMA-SVR, hybrid, testing, training





(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Alhamdulillah puji syukur ke hadirat Allah SWT atas berkah dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Penerapan Model *Hybrid Autoregressive Integrated Moving Average* dengan *Support Vector Regression* (ARIMA-SVR) pada Peramalan Penjualan Klinker Semen di PT X”. terselesaikannya skripsi ini tidak terlepas dari bantuan banyak pihak yang telah memberikan masukan kepada penulis. Untuk ini penulis mengucapkan banyak terima kasih kepada:

1. Dr. Eni Sumarminingsih, S.Si., MM. selaku dosen pembimbing atas dukungan, waktu, dan bimbingan yang telah diberikan.
2. Achmad Efendi, S.Si., M.Sc., Ph.D. selaku Ketua Program Studi Statistika Universitas Brawijaya dan dosen penguji I atas bimbingan dan saran yang telah diberikan.
3. Nurjannah, S.Si., M.Phil., Ph.D. selaku dosen penguji II atas bimbingan dan saran yang telah diberikan.
4. Rahma Fitriani, S.Si., M.Sc., Ph.D. selaku Ketua Jurusan Statistika Universitas Brawijaya.
5. Bapak dan Ibu dosen Statistika Fakultas MIPA Universitas Brawijaya atas ilmu yang telah diberikan.
6. Ibu, Bapak, Bibi, Paman, Kakek, Nenek serta seluruh keluarga besar atas doa, motivasi, dukungan, dan kasih sayang yang tiada henti.
7. Teman seperjuangan Agis, Michul, Riska, Risma, Salwa, Ardy, Dimas, Dinda, dan teman-teman Statistika 2017 atas doa, dukungan, kebersamaan, bantuan, dan memberikan semangat selama kuliah.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, baik dari materi maupun teknik penyajiannya. Oleh karena itu, diharapkan adanya saran dan kritik yang membangun dari semua pihak yang bertujuan untuk kesempurnaan skripsi ini. Semoga skripsi ini dapat memberi manfaat dan sumbangan ilmiah yang sebesar-besarnya bagi pembaca. Terima kasih.



Malang,

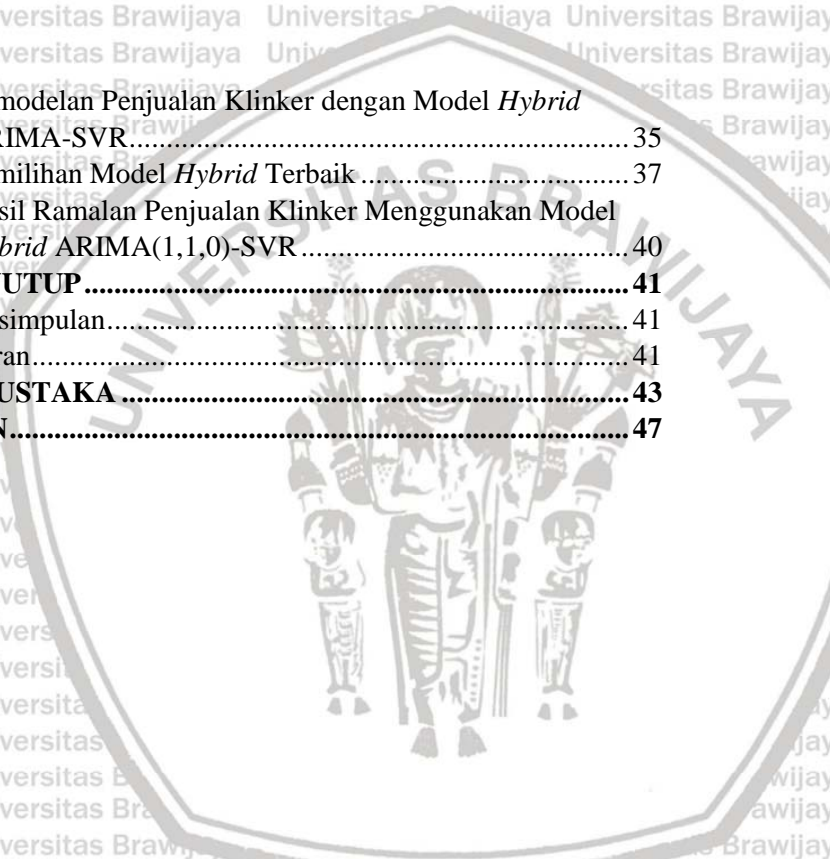
Penulis

DAFTAR ISI

	Hal.
HALAMAN JUDUL	i
LEMBAR PENGESAHAN SKRIPSI.....	iii
LEMBAR PERNYATAAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvii
DAFTAR LAMPIRAN	xix
BAB I PENDAHULUAN	1
1.1 Rumusan Masalah	3
1.2 Tujuan Penelitian.....	4
1.3 Manfaat Penelitian.....	4
1.4 Batasan Masalah.....	4
BAB II TINJAUAN PUSTAKA	5
2.1 Analisis Deret Waktu	5
2.2 Stasioneritas Deret Waktu	5
2.3 <i>Model Autoregressive Intgrated Moving Average</i> (ARIMA).....	7
2.5 Uji Nonlinieritas	13
2.6 Model <i>Support Vector Regression</i> (SVR).....	14
2.7 Model <i>Hybrid</i>	18
2.8 Keباikan Model	19
2.9 Peramalan Penjualan	19
2.10 Klinker Semen.....	20
BAB III METODE PENELITIAN.....	23
3.1 Sumber Data	23
3.2 Metode Penelitian.....	23
3.3 Diagram Alir	26
BAB IV HASIL DAN PEMBAHASAN	29
4.1 Identifikasi Pola Data	29
4.2 Pemodelan Penjualan Kinker dengan Model ARIMA	30



4.3	Pemodelan Penjualan Klinker dengan Model <i>Hybrid</i> ARIMA-SVR.....	35
4.4	Pemilihan Model <i>Hybrid</i> Terbaik.....	37
4.5	Hasil Ramalan Penjualan Klinker Menggunakan Model <i>Hybrid</i> ARIMA(1,1,0)-SVR.....	40
BAB V PENUTUP		41
5.1	Kesimpulan.....	41
5.2	Saran.....	41
DAFTAR PUSTAKA		43
LAMPIRAN		47



DAFTAR TABEL

Tabel 2.1 Nilai Transformasi <i>Box-Cox</i>	6
Tabel 2.2 Pola Teoritik ACF dan PACF dari Orde ARMA	10
Tabel 4.1 Hasil Transformasi <i>Box-Cox</i>	30
Tabel 4.2 Pola Hasil Uji ADF	31
Tabel 4.3 Pendugaan dan Uji Signifikansi Parameter Model ARIMA (<i>p, d, q</i>).....	32
Tabel 4.4 Hasil Uji <i>Ljung-Box</i> dan Uji <i>Jarque Bera</i>	33
Tabel 4.5 Hasil Pengujian Asumsi	34
Tabel 4.6 Hasil Uji <i>Ramsey RESET</i>	35
Tabel 4.7 Hasil Optimasi Parameter	37
Tabel 4.8 Nilai MAPE Model <i>Hybrid</i> pada Data <i>Training</i> dan <i>Testing</i>	37
Tabel 4.9 Data Aktual dan Hasil Prediksi pada Data <i>Testing</i>	39





(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Batas <i>error</i> SVR	15
Gambar 3.1 Diagram Alir Pembentukan Model ARIMA (p,d,q)	26
Gambar 3.2 Diagram Alir Pembentukan Model <i>hybrid</i>	28
Gambar 4.1 Plot Data Penjualan Klinker	29
Gambar 4.2 Hasil Plot PACF dan ACF	31
Gambar 4.3 Plot Data Aktual dan Prediksi Model <i>Hybrid</i> pada Data <i>Training</i>	38
Gambar 4.4 Plot Data Aktual dan Prediksi Model <i>Hybrid</i> pada Data <i>Testing</i>	39



(Halaman ini sengaja dikosongkan)

DAFTAR LAMPIRAN

Lampiran 1 Data Jumlah Penjualan Klinker Bulan Januari 2016 sampai Juni 2020	47
Lampiran 2 Uji <i>Augmented Dicky Fuller</i> (ADF)	48
Lampiran 3 Pendugaan dan Uji Signifikansi Parameter Model ARIMA (p, d, q)	49
Lampiran 4 Uji Diagnostik Model	51
Lampiran 5 Uji Nonlinieritas <i>Ramsey RESET</i>	53
Lampiran 6 Residual Model ARIMA(0,1,1).....	54
Lampiran 7 Residual Model ARIMA(1,1,0).....	55
Lampiran 8 <i>Source Code</i> Pemodelan ARIMA (p, d, q) Menggunakan <i>Software Rstudio</i> 3.6.2.....	56
Lampiran 9 <i>Source Code</i> Pemodelan <i>Hybrid</i> ARIMA(0,1,1)-SVR Menggunakan <i>Software Rstudio</i> 3.6.2	58
Lampiran 10 <i>Source Code</i> Pemodelan <i>Hybrid</i> ARIMA(1,1,0)-SVR Menggunakan <i>Software Rstudio</i> 3.6.2	61
Lampiran 11 <i>Source Code</i> Ramalan N-Periode ke depan Menggunakan Model ARIMA(1,1,0)-SVR	66







(Halaman ini sengaja dikosongkan)



BAB I PENDAHULUAN

1.1 Latar Belakang

Analisis deret waktu merupakan analisis yang dilakukan pada data yang tersusun berdasarkan waktu dengan interval yang sama. Analisis deret waktu sering diterapkan dalam bidang ekonomi maupun industri. Menurut Cryer dan Chan (2008), salah satu tujuan analisis data deret waktu adalah untuk peramalan di masa mendatang yang dapat digunakan dalam pengambilan keputusan. Metode peramalan deret waktu merupakan metode yang saat ini banyak dikembangkan yaitu dengan menggunakan data masa lalu untuk dijadikan dasar peramalan untuk masa depan.

Peramalan deret waktu dapat dilakukan dengan berbagai metode statistika. Salah satu metode statistika yang sangat sering digunakan dalam peramalan data deret waktu adalah *Autoregressive Integrated Moving Average* (ARIMA) yang dikembangkan oleh Box dan Jenkins. ARIMA merupakan metode peramalan linier jangka pendek yang memiliki akurasi sangat baik untuk data deret waktu tidak stasioner (Munarsih, 2011). Beberapa penelitian telah menggunakan metode ARIMA dalam melakukan peramalan, salah satunya yaitu penelitian yang dilakukan oleh Sen dkk., (2016) tentang penerapan ARIMA untuk peramalan konsumsi energi dan emisi GRK. Berdasarkan model yang dihasilkan, ARIMA merupakan metode peramalan yang menghasilkan model linier. Model linier belum cukup jika masih ada struktur korelasi linier yang tersisa di residual. Namun, pola nonlinier dalam data tidak dapat terdeteksi oleh analisis residual. Faktanya, saat ini tidak ada statistik diagnostik umum untuk hubungan autokorelasi nonlinier. Dengan demikian, meskipun suatu model telah lolos pemeriksaan diagnostik, model tersebut masih belum cukup karena hubungan nonlinier belum dimodelkan dengan tepat (Zhang, 2003). Selain itu, peramalan menggunakan metode linier pada data nonlinier akan menghasilkan akurasi rendah. Oleh karena itu, dalam penelitian ini dilakukan pemodelan menggunakan model linier dan model nonlinier sekaligus. Adapun model nonlinier yang digunakan yaitu model *Support Vector Regression* (SVR).

Model *Support Vector Regression* (SVR) merupakan salah satu modifikasi *Support Vector Machine* (SVM) yang digunakan untuk pendekatan regresi. SVR merupakan metode *supervised learning* yang



dapat dilatih untuk mendapatkan fungsi dari data masukan dan data keluaran yang merepresentasikan suatu nilai yang akan diprediksi. Konsep dasar SVR adalah memaksimalkan *hyperplane* untuk mendapatkan data yang akan menjadi *support vector*. Menurut Yasin dkk., (2014) keunggulan SVR adalah mampu mengatasi permasalahan *overfitting*. Adapun penelitian yang dilakukan oleh Yasin pada tahun 2014 yaitu memprediksi harga saham menggunakan metode SVR dengan algoritma *grid search* menghasilkan akurasi peramalan yang tinggi.

Adapun metode yang dapat menghasilkan model peramalan linier dan nonlinier sekaligus yaitu metode kombinasi. Kombinasi metode peramalan dari dua model individu yang saling melengkapi disebut sebagai model peramalan *hybrid* (Zhang, 2003). Model peramalan *hybrid* dapat memodelkan secara linier maupun nonlinier, sehingga dihasilkan model peramalan yang dapat menangkap pola linier dan nonlinier sekaligus. Beberapa penelitian telah dilakukan untuk pengembangan model kombinasi yang digunakan dalam peramalan deret waktu. Penelitian yang telah dilakukan oleh Faruk (2010), menyebutkan bahwa metode peramalan menggunakan model *hybrid* jaringan syaraf tiruan dengan ARIMA menghasilkan akurasi lebih baik daripada model tunggal. Hal ini juga disebutkan dalam hasil kompetisi M3 bahwa metode peramalan dengan model kombinasi akan menghasilkan akurasi lebih baik daripada metode peramalan satu model (Makridakis dan Hibon, 2000). Sehingga dalam penelitian ini dilakukan penerapan model peramalan *hybrid* ARIMA-SVR. ARIMA merupakan metode linier dan SVR merupakan metode nonlinier untuk peramalan penjualan.

Peramalan penjualan merupakan bagian penting dalam perencanaan produksi. Adanya perencanaan produksi dapat membantu perusahaan dalam menentukan penjualan produk di waktu mendatang sehingga dapat mengantisipasi kekurangan maupun kelebihan stok produk. Peramalan penjualan produk dalam penelitian ini dilakukan pada industri bahan bangunan semen khususnya dalam penjualan klinker. Klinker merupakan salah satu bahan baku semen atau bisa disebut produk semen setengah jadi yang dihasilkan dalam proses pembakaran dengan susunan senyawa kimia tertentu pada suhu 1400°C. Klinker digunakan untuk proses produksi semen oleh perusahaan juga dilakukan penjualan klinker ke luar negeri. Permasalahan yang sering dialami perusahaan yaitu sering terjadi



kekurangan stok klinker sehingga perusahaan harus melakukan pembelian klinker dari perusahaan lain. Hal ini terjadi karena belum terdapat peramalan penjualan klinker oleh perusahaan maupun peneliti lain. Penentuan penjualan klinker hanya berdasarkan jumlah rilis klinker ekspor pada tahun sebelumnya dimana sering terjadi ketidaksesuaian Rencana Kerja dan Anggaran Perusahaan (RKAP) dengan rilis aktual klinker yang dilakukan oleh perusahaan. Adapun penelitian yang telah dilakukan oleh Mahardhika dan Susanto (2017), yaitu melakukan peramalan produksi klinker menggunakan metode *exponential smoothing*. Selain itu, peramalan yang banyak dilakukan yaitu peramalan penjualan semen salah satunya adalah penelitian yang dilakukan oleh Rahmadayanti dkk. (2015) yang melakukan peramalan penjualan semen menggunakan metode ARIMA dan *exponential smoothing*. Peramalan penjualan semen merupakan peramalan yang secara tidak langsung mencakup peramalan produksi klinker yang digunakan produksi semen oleh perusahaan. Sedangkan peramalan produksi klinker mencakup peramalan produksi klinker baik yang digunakan untuk produksi semen maupun yang akan dijual ke perusahaan lain. Sehingga dalam hal ini akan terjadi tumpang tindih untuk peramalan produksi klinker yang akan digunakan perusahaan untuk produksi semen. Dengan demikian dalam penelitian ini dilakukan peramalan penjualan klinker untuk mengetahui jumlah klinker yang dijual sehingga dapat memproduksi klinker dengan kuantitas yang tepat berdasarkan data rilis klinker tahun sebelumnya.

Berdasarkan uraian permasalahan tersebut, perlu dilakukan peramalan penjualan klinker dengan hasil peramalan sedekat mungkin dengan nilai aktual. Penjualan klinker diasumsikan tidak dipengaruhi oleh faktor lain. Hal ini terjadi karena jumlah penjualan klinker mengalami perubahan sewaktu-waktu. Oleh karena itu, dalam penelitian ini dilakukan penerapan model *hybrid* ARIMA-SVR pada peramalan penjualan klinker di PT X. Sehingga model yang dihasilkan memiliki akurasi yang baik dan dapat menjelaskan hasil peramalan secara linier maupun nonlinier melalui model *hybrid* yang terbentuk.

1.1 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, adapun rumusan masalah dalam penelitian ini yaitu sebagai berikut.

1. Bagaimana model peramalan penjualan klinker di PT X menggunakan metode *hybrid* ARIMA-SVR?



2. Bagaimana akurasi model peramalan *hybrid* ARIMA-SVR dalam meramalkan penjualan klinker di PT X?

1.2 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut.

1. Memperoleh model *hybrid* ARIMA-SVR untuk peramalan penjualan klinker di PT X.
2. Mengetahui akurasi model *hybrid* ARIMA-SVR yang diperoleh.
3. Memperoleh hasil peramalan penjualan klinker di PT X untuk satu periode ke depan.

1.3 Manfaat Penelitian

Adapun manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut.

1. Menghasilkan model *hybrid* ARIMA-SVR dengan akurasi yang baik.
2. Menjadi solusi alternatif bagi perusahaan dalam peramalan penjualan klinker.
3. Mengembangkan wawasan ilmu pengetahuan mengenai metode peramalan *hybrid* bagi peneliti.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Penjualan klinker pada penelitian ini diasumsikan tidak dipengaruhi faktor lain sehingga hanya dipengaruhi oleh waktu.
2. Data yang digunakan dalam peramalan penjualan klinker pada PT X yaitu data rilis penjualan klinker pada bulan Januari 2016 hingga bulan Juni 2020.

BAB II TINJAUAN PUSTAKA

2.1 Analisis Deret Waktu

Data deret waktu merupakan serangkaian pengamatan yang diambil secara berurutan berdasarkan interval waktu yang tetap. Menurut Box dan Jenkins (1976), deret waktu adalah sekelompok nilai pengamatan yang diperoleh pada titik waktu berbeda dengan selang waktu yang sama dan diasumsikan saling berhubungan satu sama lain. Hal ini menunjukkan bahwa data deret waktu adalah sekumpulan nilai pengamatan yang memiliki keterkaitan antara kejadian masa kini terhadap masa lalu atau lebih yang diukur selama kurun waktu tertentu, berdasarkan waktu dengan interval sama.

Analisis data deret waktu merupakan metode analisis yang mempelajari data deret waktu, untuk membuat peramalan (prediksi). Peramalan deret waktu merupakan pembentukan model untuk memprediksi nilai di waktu mendatang berdasar peristiwa yang telah terjadi. Di bidang industri, data deret waktu digunakan sebagai bahan acuan pembuatan keputusan. Adanya perencanaan produksi dapat membantu perusahaan dalam menentukan kebutuhan produk di waktu mendatang.

Analisis deret waktu dilakukan dengan mempertimbangkan pola kecenderungan data. Menurut Makridakis (1999), pola data deret waktu dibedakan menjadi empat yaitu sebagai berikut.

1. Pola horizontal, terjadi bilamana data berfluktuasi di sekitar rata-rata (stasioner terhadap rata-rata).
2. Pola musiman, terjadi bilamana suatu deret dipengaruhi faktor musiman (misal kuartal tahun tertentu, bulanan atau hari-hari pada minggu tertentu).
3. Pola siklis, terjadi apabila data dipengaruhi oleh fluktuasi ekonomi jangka panjang yang berhubungan dengan siklus bisnis.
4. Pola tren, terjadi bilamana data mengandung kenaikan atau penurunan sekuler jangka panjang.

2.2 Stasioneritas Deret Waktu

Stasioneritas data deret waktu merupakan asumsi yang harus dipenuhi sebelum melakukan analisis deret waktu. Data stasioner adalah data deret waktu bersifat konstan sepanjang waktu dan tidak ada perubahan nilai tengah seiring dengan penambahan waktu.



Menurut Box-jenkins (1994), *time series* yang bersifat stasioner memiliki rata-rata dan ragam yang tidak berpengaruh terhadap pengamatan.

Data deret waktu disebut stasioner kuat jika sebaran peubah Y_t sama dengan sebaran peubah Y_{t+k} untuk setiap nilai t dan k , dan disebut stasioner lemah jika rata-rata hitung konstan $E(Y_t) = \mu$ dan autokorelasi merupakan fungsi dari lag $\rho_k = f(k)$. Kestasioneran data merupakan kondisi yang diperlukan dalam analisis data deret waktu agar ketidaktepatan model dapat diperkecil. Sehingga pada data tidak stasioner harus dilakukan transformasi stasioneritas. Adapun stasioneritas data dalam deret waktu ada dua macam yaitu stasioner terhadap ragam dan stasioner terhadap rata-rata.

2.2.1 Stasioner Terhadap Ragam

Data deret waktu dikatakan stasioner terhadap ragam apabila data berfluktuasi dengan ragam konstan dari waktu ke waktu. Pada umumnya agar ragam homogen dapat dilakukan transformasi *Box-Cox*. Model transformasi yang sering digunakan adalah transformasi yang dikenalkan oleh Box dan Cox (1964) sebagai berikut.

$$T(Y_t) = \begin{cases} \frac{Y_t^\lambda - 1}{\lambda} & \text{for } \lambda \neq 0 \\ \log(Y_t) & \text{for } \lambda = 0 \end{cases} \quad (2.1)$$

Dengan λ merupakan nilai parameter transformasi (Wei, 2006).

Bentuk transformasi *Box-Cox* yang sering digunakan untuk beberapa nilai estimasi seperti pada Tabel 2.1.

Tabel 2.1 Nilai Transformasi *Box-Cox*

Nilai λ	-1.0	-0.5	0	0.5	1.0
Transformasi	$\frac{1}{Y_t}$	$\frac{1}{\sqrt{Y_t}}$	$\ln Y_t$	$\sqrt{Y_t}$	Y_t

(Wei, 2006).

2.2.2 Stasioneritas Terhadap Rata-Rata

Data deret waktu dikatakan stasioner terhadap rata-rata apabila data berfluktuasi disekitar nilai rata-rata yang konstan. Kestasioneran data deret waktu terhadap rata-rata dapat diketahui melalui uji *Dickey Fuller* berdasarkan persamaan dan hipotesis sebagai berikut. (Wei, 2006):



$$\nabla Y_t = \delta Y_{t-1} + \alpha_t \quad (2.2)$$

Dimana $\delta = (\phi - 1)$ dan $\nabla Y_t = Y_t - Y_{t-1}$

$H_0: \delta = 1$ (data tidak stasioner)

vs

$H_1: \delta < 1$ (data stasioner)

Statistik uji T (*Dickey Fuller test*):

$$T = \frac{\hat{\delta}-1}{SE_{\hat{\delta}}} \sim t_{n-1}^{\alpha} \quad (2.3)$$

Keterangan:

$\hat{\delta}$: penduga parameter *drift*

$SE_{\hat{\delta}}$: salah baku penduga $\hat{\delta}$

Y_t : observasi pada waktu ke- t

Y_{t-1} : observasi pada waktu ke- $(t-1)$

∇Y_t : data hasil pembedaan pertama pada waktu ke- t

Tolak H_0 jika, $P(t_{n-1} > T) < \alpha$ atau $T > t_{n-1}^{\alpha}$, dikatakan bahwa data sudah stasioner terhadap rata-rata. Untuk mengatasi deret waktu yang tidak stasioner dalam rata-rata dapat dilakukan pembedaan atau sering disebut dengan *pembedaan* berdasarkan proses *backward shift* dengan operator B yang didefinisikan sebagai berikut.

$$B^z Y_t = Y_{t-z}, \text{ untuk } t \text{ dan } z = 1, 2, \dots, n \text{ di mana } t > z \quad (2.4)$$

Adapun pembedaan pertama ($d=1$) yaitu seperti persamaan (2.5).

$$\Delta Y_t = Y_t - Y_{t-1} = Y_t - B Y_t = (1 - B) Y_t \quad (2.5)$$

dan bentuk pembedaan kedua ($d=2$) yaitu seperti persamaan (2.6).

$$\begin{aligned} \Delta^2 Y_t &= \Delta Y_t - \Delta Y_{t-1} = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) \\ &= Y_t - 2Y_{t-1} + Y_{t-2} = Y_t - 2B Y_t + B^2 Y_t \\ &= (1 - 2B + B^2) Y_t = (1 - B)^2 Y_t \end{aligned} \quad (2.6)$$

Orde d adalah banyak pembedaan yang dilakukan untuk mencapai kestasioneran. Secara umum bentuk pembedaan orde ke- d untuk mencapai stasioneritas seperti pada persamaan (2.7)

$$\Delta^d Y_t = (1 - B)^d Y_t \quad (2.7)$$

Jika terhadap data sudah dilakukan sekali proses pembedaan namun masih tidak stasioner, maka akan dilakukan proses pembedaan kedua. (Cryer dan Chan, 2008).

2.3 Model Autoregressive Integrated Moving Average (ARIMA)

Deret waktu Y_t merupakan model ARIMA jika terdapat diferensiasi yaitu $W_t = \Delta^d Y_t$ (Cryer dan Chan, 2008). Model ARIMA dengan orde (p, d, q) merupakan gabungan dari model *Autoregressive*



(AR) yang memiliki orde (p) dan model *Moving Average* (MA) yang memiliki orde (q) dengan integrasi orde (d). Model AR (p) adalah model dimana deret waktu Y_t merupakan fungsi dari data di masa lalu, yakni $Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}$. Adapun persamaan AR (p) seperti pada persamaan (2.8).

$$\phi(B)Y_t = e_t \text{ atau}$$

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + e_t \quad (2.8)$$

$$\text{Dengan } \phi(B) = (1 - \phi_1 B^1 - \phi_2 B^2 - \dots - \phi_p B^p) \quad (2.9)$$

Model MA (q) adalah model yang digunakan untuk memprediksi Y_t sebagai fungsi dari sisaan atau error dimasa lalu (*past forecast error*).

Persamaan MA (q) seperti pada persamaan (2.10).

$$\theta(B)e_t = Y_t \text{ atau}$$

$$Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} \quad (2.10)$$

$$\text{Dengan } \theta(B) = (1 - \theta_1 B^1 - \theta_2 B^2 - \dots - \theta_q B^q) \quad (2.11)$$

Dalam Cryer dan Chan (2008), bentuk umum ARIMA seperti pada persamaan (2.12).

$$\phi(B)(1 - B)^d Y_t = \theta(B)e_t \quad (2.12)$$

AR (p) stasioner dengan operator

$$\phi(B) = (1 - \phi_1 B^1 - \phi_2 B^2 - \dots - \phi_p B^p) \quad (2.13)$$

dan MA (q) *invertible* dengan operator

$$\theta(B) = (1 - \theta_1 B^1 - \theta_2 B^2 - \dots - \theta_q B^q) \quad (2.14)$$

Keterangan:

ϕ : parameter *Autoregressive* (AR(p))

θ : parameter *Moving Average* (MA(q))

p : orde *Autoregressive* (AR)

d : orde *pembedaan*

q : orde *Moving Average* (MA)

e_t : sisaan acak pada waktu ke- t (*white noise*)

2.4 Pembentukan Model ARIMA

Adapun tahapan pembentukan model ARIMA adalah sebagai berikut.

2.4.1 Identifikasi Model

Apabila data deret waktu telah stasioner, maka dapat dilakukan pembentukan model ARIMA dengan identifikasi model. Proses pengidentifikasian model ARIMA dapat dilakukan dengan cara sebagai berikut.

1. Fungsi Autokorelasi (ACF)

Fungsi autokorelasi digunakan untuk menjelaskan korelasi Y_t dan Y_{t+k} dari suatu proses yang sama dan hanya terpisah oleh *lag* waktu ke- k (Wei, 2006). Adapun korelasi Y_t dan Y_{t+k} seperti pada persamaan (2.15).

$$\rho_k = \frac{\text{cov}(Y_t, Y_{t+k})}{\sqrt{\text{var}(Y_t)}\sqrt{\text{var}(Y_{t+k})}} \quad (2.15)$$

Dimana ρ_k adalah nilai autokorelasi pada *lag* waktu ke- k dengan $k = 1, 2, \dots$, $\text{Cov}(Y_t, Y_{t+k}) = E(Y_t - \mu)(Y_{t+k} - \mu) = \gamma_k$, dan $\text{Var}(Y_t) = \text{Var}(Y_{t+k}) = \gamma_0$. Sehingga fungsi autokorelasi secara umum dapat dituliskan seperti pada persamaan (2.16).

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0} = \frac{\sum_{t=1}^{n-k} (Y_t - \bar{Y})(Y_{t+k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}, \quad k=1, 2, \dots, n \quad (2.16)$$

Keterangan:

$\hat{\rho}_k$: koefisien autokorelasi sampel pada *lag* k

Y_t : pengamatan pada periode waktu ke- t

Y_{t+k} : pengamatan pada periode waktu ke- $(t+k)$

\bar{Y} : rata-rata pengamatan Y_t

2. Fungsi Autokorelasi Parsial (PACF)

Autokorelasi parsial mengukur keeratan hubungan linier antara Y_t dan Y_{t+k} setelah pengaruh dari *lag* 1, 2, . . . , $k-1$ dihilangkan. Adapun fungsi autokorelasi parsial ditunjukkan seperti pada persamaan (2.17).

$$P_k = \frac{\text{cov}[(Y_t - \hat{Y}_t), (Y_{t+k} - \hat{Y}_{t+k})]}{\sqrt{\text{var}(Y_t - \hat{Y}_t)}\sqrt{\text{var}(Y_{t+k} - \hat{Y}_{t+k})}} \quad (2.17)$$

Dengan

$$\hat{Y}_t = \beta_1 Y_{t+1} + \beta_2 Y_{t+2} + \dots + \beta_{k-1} Y_{t+k-1}$$

$$\hat{Y}_{t+k} = \alpha_1 Y_{t+k-1} + \alpha_2 Y_{t+k-2} + \dots + \alpha_{k-1} Y_{t+1}$$

Adapun fungsi PACF sampel dapat dituliskan seperti pada persamaan (2.18).

$$\hat{\phi}_{k,k} = \frac{\hat{\rho}_k - \sum_{j=1}^{k-1} \hat{\phi}_{k-1,j} \hat{\rho}_{k-j}}{1 - \sum_{j=1}^{k-1} \hat{\phi}_{k-1,j} \hat{\rho}_j} \quad (2.18)$$

Dengan $\hat{\phi}_{k,j} = \hat{\phi}_{k-1,j} - \hat{\phi}_{kk}, \hat{\phi}_{k-1,k-j}$ dan $j = 1, 2, \dots, k-1$.

Untuk menentukan ARIMA orde p dan q dapat digunakan dasar teoritik yang terdapat pada Tabel 2.2.

Table 2.2 Pola Teoritik ACF dan PACF dari Orde ARMA

	ACF	PACF
AR(p)	Turun secara eksponensial atau gelombang sinus (<i>dies down</i>)	Terpotong setelah lag ke p (<i>cut off</i>)
MA(q)	Terpotong setelah lag ke q (<i>cut off</i>)	Turun secara eksponensial atau gelombang sinus (<i>dies down</i>)
ARMA(p, q)	Turun secara eksponensial (<i>dies down</i>) setelah lag ($q-p$)	Turun secara eksponensial (<i>dies down</i>) setelah lag ($p-q$)

Sumber: Wei (2006)

2.4.2 Pendugaan Parameter Model

Pendugaan parameter dilakukan untuk memperoleh nilai dari setiap parameter dalam model ARIMA. Menurut Cryer dan Chan (2008), terdapat beberapa metode untuk menduga parameter-parameter tersebut, yaitu metode momen, metode *least square*, dan metode *maximum likelihood*. Pendugaan parameter model pada penelitian ini menggunakan metode *maximum likelihood*. Adapun pendugaan parameter model AR (1) menggunakan metode *maximum likelihood* seperti pada persamaan (2.19).

$$L(\phi, \mu, \sigma_e^2) = (2\pi\sigma_e^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma_e^2} \sum_{t=1}^n e_t^2\right) \quad (2.19)$$

Misalkan $(Y = Y_1, Y_2, \dots, Y_n)$ dan asumsikan bahwa $Y_* = (Y_{1-p}, \dots, Y_{-1}, Y_0)$ dan $e_* = (e_{1-p}, \dots, e_{-1}, e_0)$ diketahui. Sehingga fungsi *log likelihood* bersyarat yang digunakan sebagai berikut.

$$\ln L_* = -\frac{n}{2} \ln(2\pi) (\sigma_e^2)^{-\frac{n}{2}} S_*(\phi, \mu, \theta) \quad (2.20)$$

Dimana $S_*(\phi, \mu, \theta) = \sum_{t=1}^n e_t^2(\phi, \mu, \theta | Y_*, e_*, Y)$ merupakan fungsi kuadrat bersyarat (*conditional sum of square function*). (Wei, 2006).

Setelah diperoleh nilai pendugaan parameter-parameter model ARIMA, selanjutnya adalah melakukan pengujian signifikansi parameter yang diperoleh. Hipotesis yang digunakan untuk melakukan pengujian signifikansi parameter model *Autoregressive* (AR) adalah sebagai berikut.



$$H_0: \phi_j = 0$$

vs

$$H_1: \phi_j \neq 0$$

Statistik uji yang digunakan adalah seperti pada persamaan (2.21).

$$t_{hitung} = \frac{\hat{\phi}_j}{SE(\hat{\phi}_j)} \quad (2.21)$$

Keterangan:

$\hat{\phi}_j$: parameter model AR

$SE(\hat{\phi}_j)$: *standard error* dari parameter model AR

Kriteria penolakan untuk uji signifikansi parameter AR adalah menolak H_0 jika nilai statistik uji $|t_{hitung}| > t_{\frac{\alpha}{2},(n-p)}$, dengan n merupakan banyaknya pengamatan dan p merupakan banyaknya parameter yang diestimasi.

Adapun hipotesis yang digunakan untuk melakukan pengujian signifikansi parameter model *Moving Average* (MA) adalah sebagai berikut.

$$H_0: \theta_j = 0$$

vs

$$H_1: \theta_j \neq 0$$

Statistik uji yang digunakan adalah seperti pada persamaan (2.22).

$$t_{hitung} = \frac{\hat{\theta}_j}{SE(\hat{\theta}_j)} \quad (2.22)$$

Keterangan:

$\hat{\theta}_j$: parameter model MA

$SE(\hat{\theta}_j)$: *standard error* dari parameter model MA

Kriteria penolakan untuk uji signifikansi parameter MA adalah menolak H_0 jika nilai statistik uji $|t_{hitung}| > t_{\frac{\alpha}{2},(n-p)}$, dengan n merupakan banyaknya pengamatan dan p merupakan banyaknya parameter yang diestimasi.

2.4.3 Diagnostik Model

Perlu dilakukan pemeriksaan diagnostik (*diagnostic checking*) setelah dilakukan pendugaan parameter agar model siap digunakan untuk peramalan. Diagnostik model dilakukan dengan memeriksa atau menguji model yang telah dispesifikasi secara benar atau telah dipilih orde p , d , dan q dengan benar. Menurut Wei (2006), dalam melakukan pengujian asumsi *white noise* dapat dilakukan dengan melihat plot



ACF dari sisaan atau menggunakan pengujian *Ljung-Box*. Sisaan dapat dikatakan memenuhi asumsi *white noise* jika tidak terdapat nilai ACF yang signifikan pada sisaan. Sama halnya dengan uji *Ljung-Box*, pengujian ini dilakukan dengan menggunakan autokorelasi dari sisaan sampel. Hipotesis yang digunakan dalam uji *Ljung-Box* adalah sebagai berikut.

$$H_0 : \rho_1 = \rho_2 = \dots = \rho_k = 0 \text{ (white noise)}$$

vs

H_1 : minimal ada satu nilai $\rho_k \neq 0$ dengan $k = 1, 2, \dots, k$ (tidak *white noise*)

Menurut Cryer dan Chan (2008), statistik uji yang digunakan adalah Q yang dapat dihitung menggunakan persamaan (2.23).

$$Q = n(n+2) \sum_{k=1}^K \frac{\hat{\rho}_k^2}{n-k} \sim \chi_{(K-p-q)}^2 \quad (2.23)$$

Keterangan:

$\hat{\rho}_k$: penduga autokorelasi residual pada lag ke- k

k : banyaknya autokorelasi yang diuji

p : banyaknya parameter AR pada model

q : banyaknya parameter MA pada model

n : banyaknya pengamatan

Statistik uji Q mendekati distribusi *chi-square* dengan derajat bebas $(K-p-q)$. Kriteria penolakan untuk uji Q adalah menolak H_0 jika

$$Q > \chi_{(\alpha, K-p-q)}^2$$

Selain pengujian asumsi sisaan bersifat *white noise*, dilakukan juga pengujian asumsi kenormalan sisaan (uji normalitas sisaan). Uji normalitas dilakukan untuk mengetahui apakah sisaan dari model yang didapatkan berdistribusi normal atau tidak. Salah satu uji normalitas terhadap sisaan yang dapat digunakan adalah uji statistik *Jarque Bera Test (JB)*, dengan hipotesis yang digunakan sebagai berikut:

$$H_0 : \varepsilon_i \sim N(\mu, \sigma^2) \text{ Sisaan mengikuti distribusi normal}$$

vs

$$H_1 : \varepsilon_i \not\sim N(\mu, \sigma^2) \text{ Sisaan tidak mengikuti distribusi normal}$$

Statistik uji yang digunakan adalah JB yang dapat dihitung menggunakan persamaan (2.24).

$$JB = \frac{s^2}{6/n} + \frac{(R^2-3)^2}{24/n} \quad (2.24)$$

Kriteria pengambilan keputusan yaitu membandingkan nilai statistik uji (JB) dengan titik kritis dari uji JB . Terima H_0 apabila nilai $JB <$

$\chi^2_{(2)}$ atau jika $p\text{-value} > (\alpha=0.05)$, yang artinya sisaan mengikuti distribusi normal.

2.4.4 Pemilihan Model Terbaik

Pemilihan model terbaik dapat dilakukan dengan menggunakan metode *Akaike's Information Criteria* (AIC) dengan memilih nilai yang terkecil (Enders, 2004). AIC dapat didefinisikan seperti persamaan berikut.

$$\begin{aligned} AIC &= n \ln(\text{jumlah kuadrat sisaan}) 2p \\ &= n \ln\left(\frac{\sum_{t=1}^n e_t^2}{n}\right) 2p \end{aligned} \quad (2.25)$$

Keterangan:

n : Banyaknya pengamatan yang ada dalam pendugaan parameter.

p : Banyaknya parameter dalam model dengan intersep ($p+q+1$) dan tanpa intersep ($p+q$).

e_t : Sisaan model.

2.4.5 Peramalan Model ARIMA

Peramalan adalah kegiatan memperkirakan atau kegiatan memprediksikan apa yang akan terjadi pada masa yang akan datang. Peramalan model ARIMA (p,d,q) dilakukan berdasarkan model yang sesuai dengan data deret waktu yang digunakan. Peramalan model ARIMA (p,d,q) disajikan dalam persamaan (2.26).

$$\begin{aligned} \hat{Y}_t(l) &= \phi_1 \hat{Y}_t(l-1) + \phi_2 \hat{Y}_t(l-2) + \dots + \phi_p \hat{Y}_t(l-p) + \theta_0 \\ &\quad - \theta_1 E(e_{t+l-1} | Y_1, Y_2, \dots, Y_t) - \theta_2 E(e_{t+l-2} | Y_1, Y_2, \dots, Y_t) \\ &\quad - \dots - \theta_q E(e_{t+l-q} | Y_1, Y_2, \dots, Y_t) \end{aligned} \quad (2.26)$$

dengan,

$$E(e_{t+j} | Y_1, Y_2, \dots, Y_t) = \begin{cases} 0 & \text{untuk } j > 0 \\ e_{t+j} & \text{untuk } j \leq 0 \end{cases} \quad (2.27)$$

Keterangan:

Y_t : Pengamatan pada waktu ke- t

e_t : Galat pada waktu ke- t

2.5 Uji Nonlinieritas

Sebelum melakukan pemodelan linier, perlu dilakukan pengujian apakah data tersebut merupakan data nonlinier atau tidak, sehingga metode yang digunakan sesuai dengan data. Terdapat beberapa uji



nonlinieritas yang dapat digunakan, akan tetapi dalam penelitian ini pengujian nonlinieritas dilakukan menggunakan uji *Ramsey RESET*.

RESET tes pertama kali diperkenalkan oleh Ramsey pada 1969 yang berawal dari ide bahwa jika tidak terdapat nonlinieritas maka berbagai transformasi nonlinier dari $f_t = \tilde{X}'_t \hat{\theta}$ tidak memberikan manfaat untuk menyatakan y_t (Kim dkk, 2004). Prosedur uji pada *RESET* tes dapat dijelaskan sebagai berikut :

(i) Regresikan y_t pada \tilde{X}'_t sehingga diperoleh model linier $y_t = f_t + \hat{\epsilon}_t$, dimana $f_t = \tilde{X}'_t \hat{\theta}$ (2.28)

(ii) Tambahkan model linier dalam bentuk $\hat{\epsilon}_t = a_2 f_t^2 + \dots + a_k f_t^k + v_t$ untuk $k \geq 2$ sehingga diperoleh model alternatif

$y_t = \tilde{X}'_t \hat{\theta} + a_2 f_t^2 + \dots + a_k f_t^k + v_t$ untuk $k \geq 2$ (2.29)

(iii) Test dilakukan dengan menguji hipotesis berikut.

$H_0: a_2 = \dots = a_k = 0$

V_s

$H_1: a_k \neq 0$

Jika $\hat{\epsilon} = (\hat{\epsilon}_1, \dots, \hat{\epsilon}_n)$ adalah nilai-nilai residual prediksi dari model linier pada (2.28) dan $\hat{v} = (\hat{v}_1, \dots, \hat{v}_n)$ adalah residual dari model alternatif pada (2.29) maka statistik ujinya adalah sebagai berikut.

$$RESET = \frac{[(\hat{\epsilon}'\hat{\epsilon} - \hat{v}'\hat{v})/(k-1)]}{[(\hat{v}'\hat{v})/(n-k)]} \quad (2.30)$$

H_0 ditolak jika $RESET > F(k-1, n-k)$, sehingga data nonlinier. (Warsito, B., Ispriyanti, D., 2004).

2.6 Model Support Vector Regression (SVR)

2.6.1 Support Vector Regression (SVR)

SVR merupakan suatu model yang dikembangkan dari SVM dalam pemodelan regresi sehingga SVR mampu mengatasi masalah regresi. Berdasarkan prinsip *structural risk minimization* (SRM), SVR mengestimasi suatu fungsi untuk menemukan *hyperplane* terbaik yang dapat memisahkan data dan memperkecil jarak antara *hyperplane* dengan data (Pai dan Lin, 2005). Adapun tujuan SVR yaitu menghasilkan suatu fungsi dengan tingkat kesalahan paling kecil untuk menghasilkan suatu prediksi yang bagus. Konsep dasar SVR untuk regresi adalah mengenalkan fungsi kernel, memetakan data input ke dalam ruang fitur berdimensi tinggi dengan pemetaan



nonlinier dan kemudian melakukan regresi linier di ruang fitur (Shabri dan Suhartono, 2012). Fungsi regresi dalam SVR merupakan fungsi dengan batasan deviasi tertentu yang diperoleh dari data *training* sehingga dapat menghasilkan prediksi yang mendekati target aktual. Adapun fungsi regresi pada SVR dirumuskan sebagai berikut.

$$f(x) = \mathbf{w}^T \varphi(x) + b \quad (2.31)$$

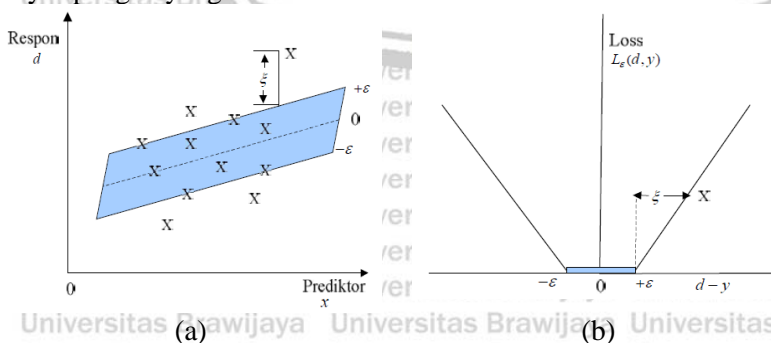
Dimana \mathbf{w} adalah vektor pembobot, $\varphi(x)$ adalah fungsi yang memetakan secara nonlinier x dari ruang input ke dalam suatu ruang fungsi berdimensi tinggi, dan b adalah bias. Koefisien \mathbf{w} dan b diestimasi dengan cara meminimalkan fungsi risiko yang di definisikan dalam persamaan (2.32).

$$R(f(x)) = C \sum_{i=1}^T L_{\varepsilon}(\mathbf{y}_i, f(\mathbf{x}_i)) + \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.32)$$

Dimana

$$L_{\varepsilon}(\mathbf{y}_i, f(\mathbf{x}_i)) = \begin{cases} -\varepsilon |f(\mathbf{x}_i) - \mathbf{y}_i|, & |f(\mathbf{x}_i) - \mathbf{y}_i| \geq \varepsilon \\ 0, & |f(\mathbf{x}_i) - \mathbf{y}_i| < \varepsilon \end{cases} \quad (2.33)$$

L_{ε} merupakan ε - insensitive loss function, $\frac{1}{2} \|\mathbf{w}\|^2$ merupakan vektor pembobot norm, \mathbf{y}_i vektor dari nilai sebenarnya, C dan ε merupakan parameter yang sudah ditentukan. C adalah konstanta yang mengevaluasi *trade-off* antara kompleksitas model dan sejauh mana penyimpangan yang lebih besar dari ε ditoleransi.



Gambar 2.1 Batas *error* SVR (Santosa, 2013)

Berdasarkan Gambar 2.1 (a) dapat diketahui bahwa sampai dengan garis batas *error* ε , nilai *error* dianggap sama dengan 0, sedangkan di luar batas tersebut nilai *error* akan dihitung sebagai ξ_i atau ξ_i^* . ξ_i, ξ_i^*



merupakan positif *slack* variabel, yang mewakili jarak dari nilai sebenarnya ke nilai batas yang sesuai dari ε . Solusi pada masalah ini adalah dari sebuah garis yang memiliki toleransi terhadap error seperti pada Gambar 2.1 (b).

Fungsi f dalam pemodelan SVR, diasumsikan dapat mengaproksimasi semua titik (x_i, y_i) dengan presisi ε . Suatu titik disebut *feasible* apabila semua titik berada dalam rentang $f \pm \varepsilon$. Sedangkan kondisi *infeasible* ketika beberapa titik yang mungkin keluar dari rentang $f \pm \varepsilon$. Untuk mengatasi masalah *infeasible constraint* maka dapat dilakukan dengan menambahkan variabel *slack* pada titik-titik yang *infeasible*. Sehingga optimasi pada persamaan (2.31) yaitu dapat dilihat pada persamaan (2.34).

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (2.34)$$

dengan kendala:

$$\mathbf{y}_i - \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) - b \leq \varepsilon + \xi_i \quad (2.35)$$

$$\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b - \mathbf{y}_i \leq \varepsilon + \xi_i^*, \text{ dan} \quad (2.36)$$

$$\xi_i, \xi_i^* \geq 0, i = 1, 2, \dots, \lambda. \quad (2.37)$$

Untuk mendapatkan solusi persamaan (2.33) maka dapat menggunakan *primal langrangian* dalam bentuk sebagai berikut.

$$L(\mathbf{w}, b, \xi, \xi^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*) = \frac{1}{2} \|\mathbf{w}\|^2 + C (\sum_{i=1}^n (\xi_i, \xi_i^*)) - \sum_{i=1}^n \alpha_i^* [\mathbf{w}\boldsymbol{\varphi}(\mathbf{x}_i) + b - \mathbf{y}_i + \varepsilon + \xi_i^*] - \sum_{i=1}^n \alpha_i [\mathbf{y}_i - \mathbf{w}\boldsymbol{\varphi}(\mathbf{x}_i) - b + \varepsilon + \xi_i] - \sum_{i=1}^n (\beta_i \xi_i + \beta_i^* \xi_i^*) \quad (2.38)$$

Persamaan di atas diminimalkan dengan melakukan turunan parsial L_p terhadap variabel primal $\mathbf{w}, b, \xi, \xi^*$ sehingga menghasilkan persamaan dual untuk problem optimasi dari SVR sebagai berikut.

$$L_p(\alpha_i, \alpha_i^*) = \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) - \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) - \frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) x_i, x_j$$

dan dimaksimalkan dalam bentuk *langrangian multiplier* nonnegatif $\alpha_i, \alpha_i^*, \beta_i, \beta_i^*$. Dalam formulasi dual, problem SVR seperti pada persamaan berikut.

$$\max -\frac{1}{2} \sum_{i=1}^{\lambda} \sum_{j=1}^{\lambda} (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(x_i, x_j) + \sum_{i=1}^{\lambda} (\alpha_i - \alpha_i^*) y_i - \varepsilon \sum_{i=1}^{\lambda} (\alpha_i - \alpha_i^*) \quad (2.39)$$



Dengan kendala:

$$\sum_{i=1}^{\lambda} (\alpha_i - \alpha_i^*) = 0$$

$$0 \leq \alpha_i \leq C, i = 1, 2, \dots, \lambda$$

$$0 \leq \alpha_i^* \leq C, i = 1, 2, \dots, \lambda$$

Berdasarkan persamaan dual pada persamaan diperoleh solusi optimal dari kendala estimasi parameter w dalam bentuk koefisien *langrange* α_i dan α_i^* yang dapat ditulis kembali seperti persamaan (2.40).

$$w = \sum_{i=1}^n (\alpha_i - \alpha_i^*) x_i \quad (2.40)$$

Sehingga fungsi regresi SVR untuk kasus linier sebagai berikut.

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) (x_i, x_j) + b \quad (2.41)$$

Adapun SVR untuk kasus nonlinier memberikan pendekatan alternatif dengan cara melakukan pemetaan data x dari *input space* ke *feature space* dengan dimensi yang lebih tinggi melalui fungsi φ sehingga fungsi regresi nonlinier dapat dituliskan sebagai berikut.

$$f(x) = \sum_{i=1}^{\lambda} (\alpha_i - \alpha_i^*) K(x_i, x_j) + b \quad (2.42)$$

Dimana $K(x_i, x_j)$ adalah *dot-product* kernel yang didefinisikan sebagai $K(x_i, x_j) = \varphi(x_i) \varphi(x_j)$.

2.6.2 Fungsi Kernel

Beberapa metode dalam analisis data mining banyak menggunakan fungsi linier. Fungsi kernel bertujuan untuk memetakan data dari ruang input ke ruang fitur (ruang berdimensi lebih tinggi). Banyak kasus di dunia nyata merupakan kasus yang nonlinier, sehingga untuk mengatasinya dengan cara mentransformasi data ke dalam dimensi ruang yang lebih tinggi. SVM dapat digunakan pada data nonlinier dengan menggunakan pendekatan kernel sehingga dapat dipisahkan secara linier pada *feature space* yang baru. Fungsi kernel yang digunakan pada metode SVR adalah sebagai berikut.

1. Linier: $x^T x$ (2.43)

2. Polinomial: $(x^T x + 1)^n$ (2.44)

3. *Radial Basis Function (RBF)*: $\exp(-\gamma \|x - x_i\|^2)$ (2.45)

Pemilihan fungsi kernel yang tepat merupakan hal yang penting karena menentukan *feature space*. Dalam penelitian ini menggunakan fungsi kernel *Radial Basis Function (RBF)* pada model SVR. SVR dengan fungsi kernel RBF memiliki tiga parameter yang harus



ditentukan yaitu C (*cost*), γ (*gamma*), dan ε (*epsilon*). Metode yang digunakan untuk mencari parameter optimal pada penelitian ini adalah metode *grid search*.

2.6.3 Algoritma *Grid-Search*

Algoritma *grid search* merupakan salah satu algoritma untuk menentukan parameter optimal pada model SVR. Konsep kerja algoritma *grid search* yaitu dengan membagi jangkauan parameter yang akan dioptimalkan ke dalam *grid* dan melintasi semua titik untuk mendapatkan parameter yang optimal. Dalam penerapannya, algoritma *grid search* memerlukan metrik kinerja sebagai pemandu, biasanya diukur dengan *cross validation* pada data *training* (Hsu dkk, 2004). Oleh karena itu disarankan untuk mencoba beberapa variasi pasangan parameter pada *hyperplane* SVR untuk menentukan *hyperplane* dengan akurasi tinggi. Pasangan parameter yang menghasilkan akurasi terbaik yang didapatkan dari uji *cross-validation* merupakan parameter yang optimal. Parameter optimal tersebut yang selanjutnya digunakan untuk model SVR terbaik. Setelah itu, model SVR tersebut digunakan untuk memprediksi data *testing* untuk mendapatkan generalisasi tingkat akurasi model.

Menurut Leidiyana (2013), metode pengujian standar yang dapat dilakukan untuk memprediksi *error rate* adalah *cross-validation*. Membagi data *training* secara random ke dalam beberapa bagian dengan perbandingan yang sama kemudian *error rate* dihitung bagian demi bagian, selanjutnya hitung rata-rata seluruh *error rate* untuk mendapatkan *error rate* secara keseluruhan.

2.7 Model *Hybrid*

Perilaku permintaan klinker ekspor tidak mudah untuk diprediksi. Oleh karena itu, peramalan menggunakan metode *hybrid* yang memiliki kemampuan pemodelan linier dan nonlinier merupakan metode terbaik untuk meramalkan penjualan klinker ekspor. Pendekatan model linier ARIMA belum cukup jika masih ada struktur korelasi linier yang tersisa di residual karena hubungan nonlinier belum dimodelkan dengan tepat (Zhang, 2003). Oleh karena itu, model *hybrid* dilakukan untuk memodelkan pola linier dan nonlinier secara keseluruhan yang dapat meningkatkan kinerja peramalan (Chen dkk, 2007).

Adapun bentuk umum model *hybrid* sebagai berikut (Zhang, 2003).



$$Y_t = L_t + N_t + e_t \quad (2.46)$$

Dimana L_t adalah komponen linier dan N_t adalah komponen nonlinier dari model *hybrid*. Langkah pemodelan *hybrid* yaitu pertama data dimodelkan dengan model linier ARIMA kemudian diperoleh residual pada waktu ke- t dari model linier, sebagai berikut.

$$N_t = Y_t - \hat{L}_t \quad (2.47)$$

Dimana \hat{L}_t menunjukkan nilai ramalan model linier pada waktu ke- t . kemudian N_t dimodelkan menggunakan model nonlinier SVR, sehingga hubungan nonlinier dapat tertangkap. Dengan p input, model nonlinier dari residual menjadi sebagai berikut.

$$N_t = f(N_{t-1}, N_{t-2}, \dots, N_{t-p}) + e_t \quad (2.48)$$

dengan f adalah fungsi nonlinier yang ditentukan oleh model nonlinier dan e_t adalah kesalahan acak. Jika model f yang terbentuk tidak sesuai maka e_t tidak akan sepenuhnya acak, sehingga identifikasi model adalah faktor yang sangat penting. Oleh karena itu, ramalan gabungannya adalah sebagai berikut.

$$\hat{Y}_t = \hat{L}_t + \hat{N}_t \quad (2.49)$$

Dimana \hat{N}_t adalah nilai peramalan pada persamaan (2.48).

2.8 Kebaikan Model

Evaluasi model dilakukan untuk mengetahui seberapa baik model yang terbentuk berdasarkan residual yang dihasilkan. Kriteria pemilihan model terbaik yang digunakan adalah berdasarkan nilai *Mean Absolute Percentage Error* (MAPE) untuk data *training* dan data *testing*. Model terbaik didapatkan jika nilai MAPE paling kecil diantara model yang ada, hal ini sesuai dengan tujuan dari peramalan, yaitu untuk memperoleh angka ramalan dengan kesalahan sekecil-kecilnya. Besarnya nilai MAPE dapat dihitung dengan formula sebagai berikut (Wei, 2006):

$$\text{MAPE} = \left(\frac{1}{L} \sum_{l=1}^L \left| \frac{Y_{n+l} - \hat{Y}_{n(l)}}{Y_{n+l}} \right| \right) 100\% \quad (2.50)$$

dengan n adalah banyaknya data *training* (*in sample*), L adalah banyaknya ramalan data *testing* (*out sample*) yang dilakukan, Y_{n+l} adalah data aktual dan $\hat{Y}_{n(l)}$ adalah data hasil ramalan.

2.9 Peramalan Penjualan

Peramalan penjualan merupakan suatu usaha perencanaan strategis yang bertujuan untuk optimalisasi kebutuhan dan keinginan pembeli. Penjualan merupakan suatu transaksi yang bertujuan untuk



mendapatkan suatu keuntungan dan merupakan suatu jantung dari suatu perusahaan (Himayati, 2008). Banyak faktor yang mempengaruhi penjualan yang berasal dari eksternal perusahaan maupun internal perusahaan.

Penjualan merupakan proses penting dalam suatu perusahaan. Setiap kegiatan atau proses produksi berpengaruh sangat besar terhadap kegiatan penjualan. Kegiatan produksi dilakukan berdasarkan perencanaan produksi. Perencanaan produksi meliputi perencanaan jumlah barang yang akan dijual pada masa yang akan datang. Jumlah barang atau jasa yang akan dijual pada masa mendatang mampu diperkirakan melalui peramalan (*forecasting*).

Peramalan merupakan suatu teknik yang digunakan untuk memprediksi atau meperkirakan suatu nilai pada masa mendatang menggunakan data masa lalu maupun data saat ini. Peramalan yang sering digunakan untuk keperluan perencanaan produksi pada suatu perusahaan adalah peramalan penjualan. Adanya peramalan penjualan dalam suatu perencanaan produksi dapat membantu memperkirakan jumlah barang atau jasa yang akan dijual sehingga proses produksi dapat disesuaikan dengan hasil peramalan penjualan yang diperoleh. Adapun persamaan peramalan penjualan menggunakan model *hybrid* ARIMA(1,1,0)-SVR dapat dilihat pada persamaan (2.51).

$$\hat{Y}_t(l) = \hat{L}_t(l) + \hat{N}_t(l) \quad (2.51)$$

$$\hat{Y}_t(l) = [(1 + \phi)\hat{Y}_t(l-1) - \phi\hat{Y}_t(l-2) + \theta_0] + [C \sum_{i=1}^l \exp(-\gamma \|e_{ti} - e_{ti}^*\|^2) + b] \quad (2.52)$$

Dimana

$$e_{ti} = [e_{t+i-1}, e_{t+i-2}, \dots, e_{t+i-p}]$$

$$e_{ti}^* = [e_{t+i-2}, e_{t+i-3}, \dots, e_{t+i-(p+1)}]$$

$\hat{Y}_t(l)$: Hasil ramalan model *Hybrid*

$\hat{L}_t(l)$: Hasil ramalan model ARIMA

$\hat{N}_t(l)$: Hasil ramalan model SVR

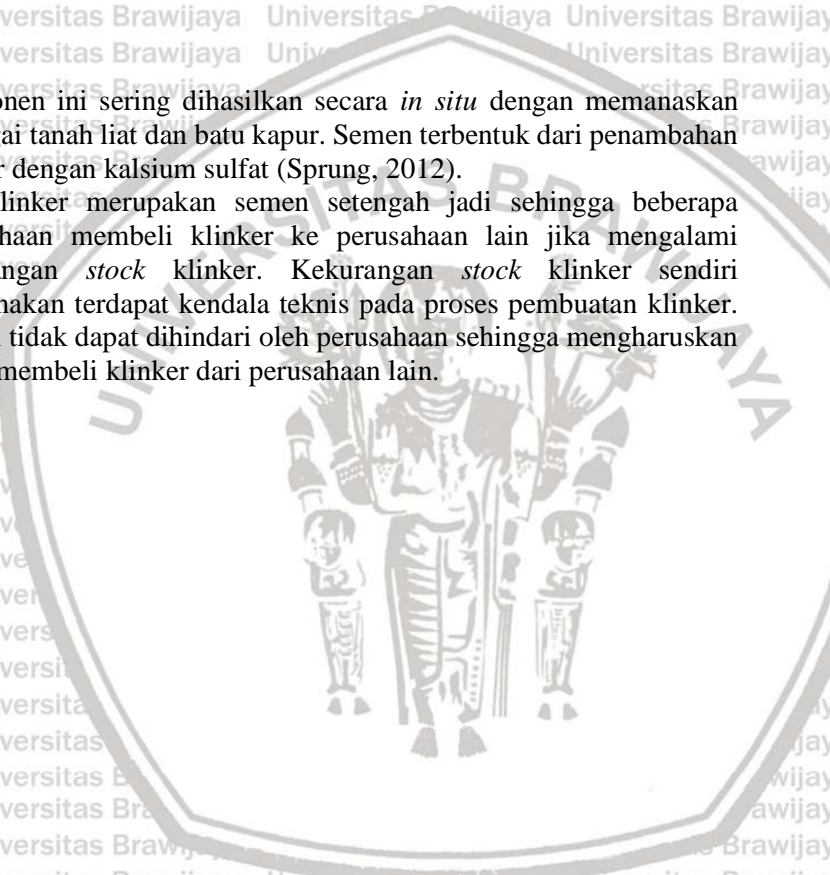
2.10 Klinker Semen

Klinker merupakan bahan baku semen yang dihasilkan dari proses pembakaran dengan suhu 1400°C. Klinker terdiri dari berbagai kalsium silikat termasuk *alite* dan *belite*. *Tricalcium aluminate* dan *kalsium aluminoferrite* adalah komponen umum lainnya. Komponen-



komponen ini sering dihasilkan secara *in situ* dengan memanaskan berbagai tanah liat dan batu kapur. Semen terbentuk dari penambahan klinker dengan kalsium sulfat (Sprung, 2012).

Klinker merupakan semen setengah jadi sehingga beberapa perusahaan membeli klinker ke perusahaan lain jika mengalami kekurangan *stock* klinker. Kekurangan *stock* klinker sendiri dikarenakan terdapat kendala teknis pada proses pembuatan klinker. Hal ini tidak dapat dihindari oleh perusahaan sehingga mengharuskan untuk membeli klinker dari perusahaan lain.





(Halaman ini sengaja dikosongkan)

BAB III METODE PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yaitu data penjualan klinker per bulan pada Januari tahun 2016 hingga Juni tahun 2020. Data diperoleh dari PT X sebanyak 54 pengamatan. Adapun analisis dalam penelitian ini secara keseluruhan dilakukan dengan bantuan *software Rstudio* 3.6.2 untuk perhitungan statistika dan menggunakan *Microsoft Excel* untuk penyusunan data.

3.2 Metode Penelitian

Metode analisis data pada penelitian ini dilakukan dengan beberapa tahap. Tahap pertama dilakukan pemodelan data menggunakan model linier ARIMA. Selanjutnya tahap kedua dilakukan pemodelan SVR pada residual hasil ramalan model linier. Setelah memperoleh model yang bisa menangkap pola linier dan nonlinier data maka dapat dilakukan peramalan menggunakan metode *hybrid* ARIMA-SVR. Adapun langkah-langkah dalam peramalan model *hybrid* ARIMA-SVR adalah sebagai berikut.

3.2.1 Pembentukan Model ARIMA (p,d,q)

Berikut merupakan langkah-langkah pembentukan model ARIMA (p,d,q).

1. Membuat plot data penjualan klinker untuk mengetahui pola data.
2. Membagi data menjadi data *training* dan data *testing*. Dimana data *training* merupakan data penjualan klinker pada Januari 2016 hingga bulan November 2019. Sedangkan data *testing* merupakan data penjualan klinker pada bulan Desember 2019 hingga bulan Juni 2020.
3. Menguji kestasioneran terhadap ragam dengan melihat λ (*rounded value*) pada *Box Cox plot*, jika tidak bernilai satu maka data tidak stasioner terhadap ragam sehingga perlu dilakukan transformasi *Box-Cox* seperti pada persamaan (2.1).
4. Menguji kestasioneran terhadap rata-rata menggunakan uji *Dickey Fuller* dengan statistik uji pada persamaan (2.3), jika data tidak stasioner terhadap rata-rata maka perlu distasionerkan dengan melakukan diferensiasi menggunakan persamaan (2.5) untuk diferensiasi orde ke- d .



5. Melakukan identifikasi model ARIMA (p,d,q) dengan melihat plot ACF dan PACF untuk menentukan orde (p,d,q) sesuai persamaan (2.16) dan (2.18).
6. Melakukan pendugaan parameter pada model tentatif yang terbentuk menggunakan metode *maximum likelihood* sesuai pada persamaan (2.20).
7. Kemudian setelah dilakukan pendugaan parameter, selanjutnya dilakukan uji signifikansi parameter secara parsial menggunakan uji t. Adapun statistik uji yang digunakan sesuai pada persamaan (2.21) untuk uji signifikansi parameter AR(p) dan persamaan (2.22) untuk uji signifikansi parameter MA(q).
8. Melakukan pengujian terhadap sisaan yang bersifat *white noise* menggunakan pengujian *Ljung-Box* dengan statistik uji pada persamaan (2.23).
9. Melakukan pengujian normalitas terhadap sisaan model menggunakan uji *Jarque Bera* dengan statistik uji sesuai pada persamaan (2.24).
10. Menghitung nilai AIC sesuai persamaan (2.25) untuk memperoleh model linier terbaik yang akan digunakan sebagai model *hybrid*.
11. Menghitung *fitted value* dari model ARIMA yang signifikan yaitu model ARIMA(0,1,1), ARIMA(1,1,0).
12. Menghitung residual dari model ARIMA yang diperoleh untuk digunakan sebagai input model SVR.

3.2.2 Pembentukan Model *hybrid* ARIMA-SVR

Berikut merupakan langkah-langkah pembentukan model *hybrid* ARIMA-SVR.

1. Sebelum dilakukan pemodelan SVR, maka dilakukan pengujian nonlinieritas pada residual model ARIMA sesuai sub bab 2.5.
2. Pemodelan SVR dilakukan dengan menggunakan input dari residual model ARIMA yang diperoleh, dalam penelitian ini menggunakan model ARIMA dengan semua parameter yang signifikan yaitu model ARIMA(0,1,1) dan ARIMA(1,1,0).
3. Penentuan fungsi kernel agar dapat menentukan nilai-nilai parameter *gamma*, *cost* (C), dan *epsilon* untuk optimasi *hyperplane* pada data *training*.

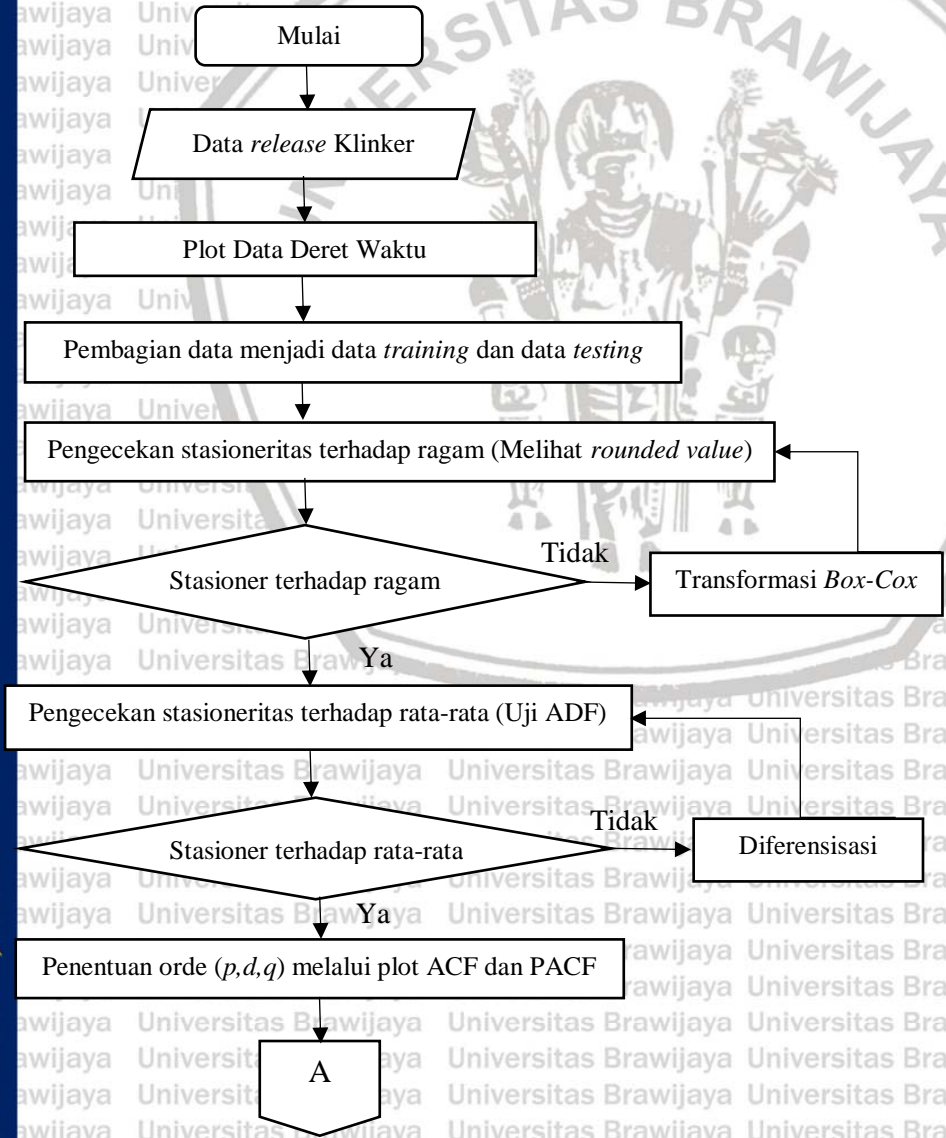


4. Pemilihan parameter terbaik menggunakan metode *grid search* sesuai penjelasan pada anak sub bab (2.6.3) untuk memperoleh model SVR optimal.
5. Membentuk model *hybrid* ARIMA-SVR yang merupakan gabungan model ARIMA dan model SVR yang diperoleh.
6. Pemilihan model *hybrid* terbaik dengan melihat nilai MAPE yang terkecil pada data *training* maupun data *testing*.
7. Peramalan penjualan klinker satu periode ke depan yaitu penjualan klinker bulan Juli 2020 menggunakan model *hybrid*.

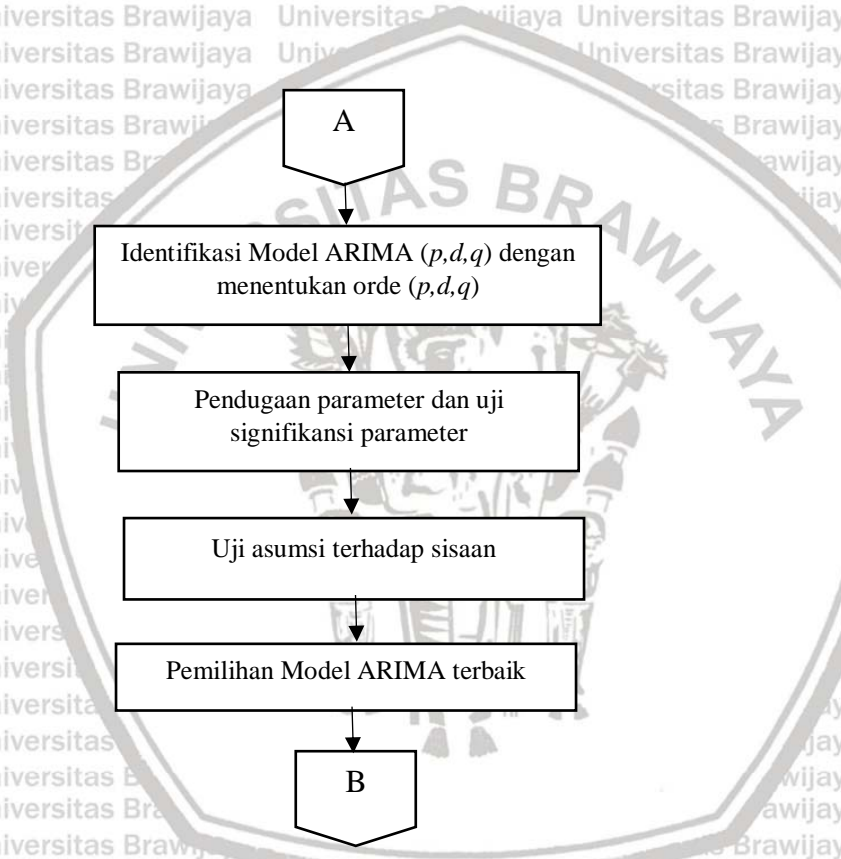


3.1 Diagram Alir

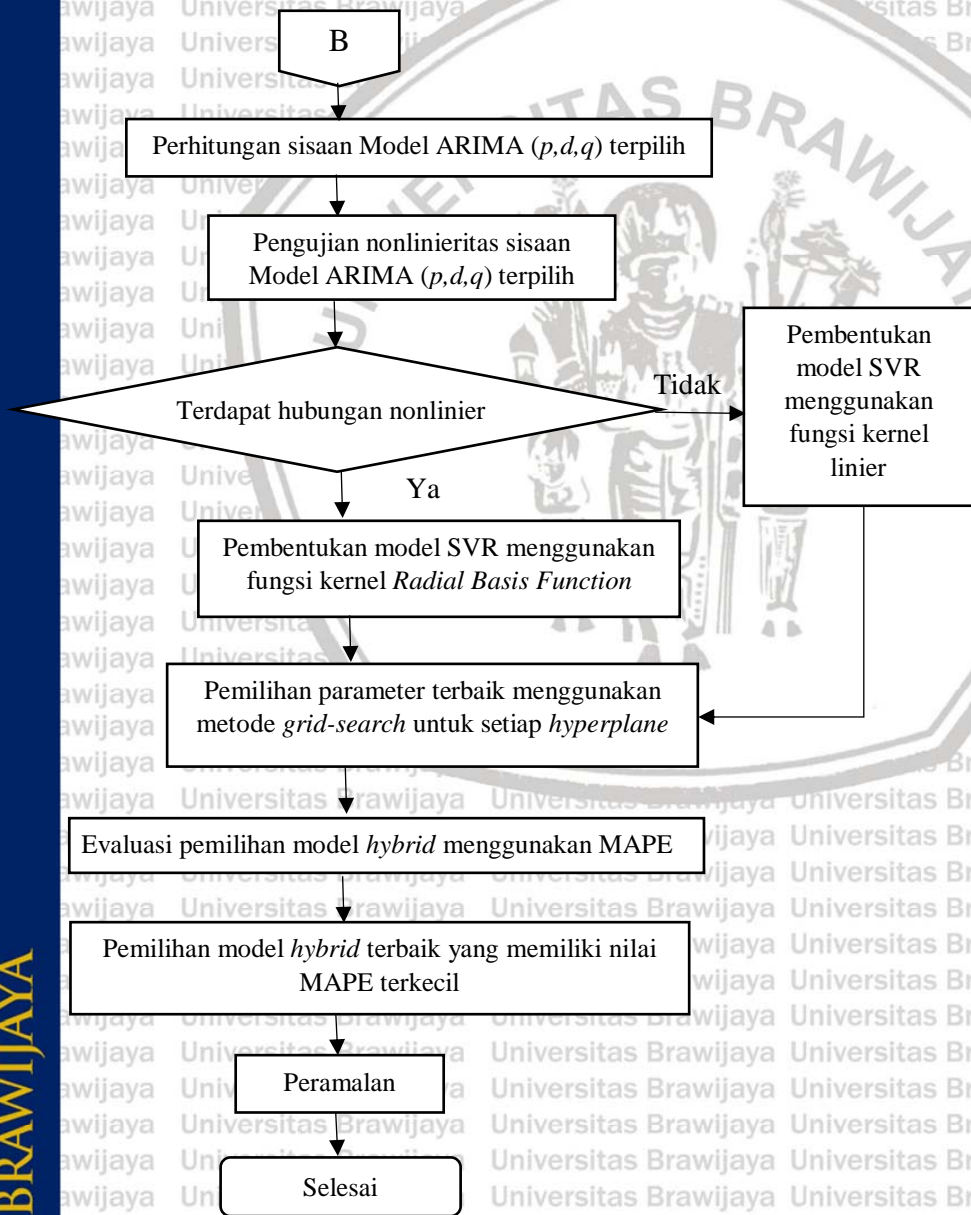
Adapun tahapan dalam peramalan *hybrid* ARIMA-SVR untuk meramalkan penjualan klinker yaitu sebagai berikut.



Gambar 3.1 Diagram Alir Pembentukan Model ARIMA (p,d,q)



Gambar 3.1 Diagram Alir Pembentukan Model ARIMA (p,d,q) (Lanjutan)

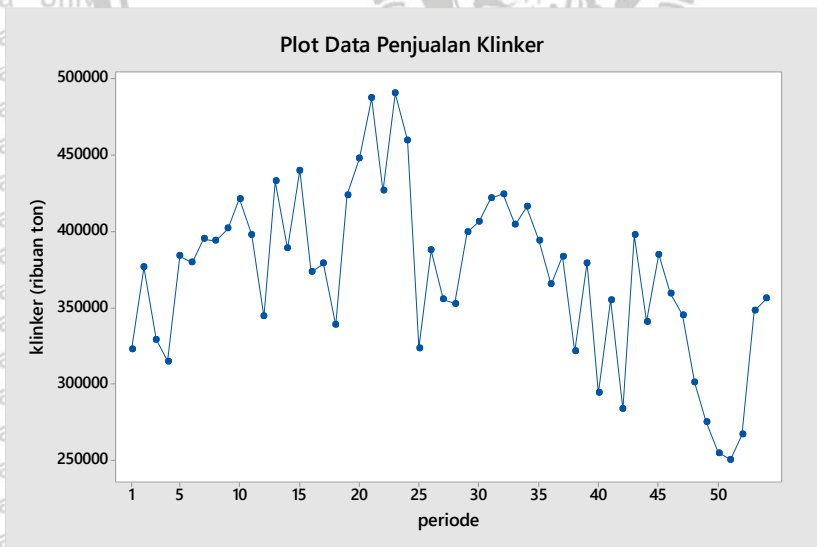


Gambar 3.2 Diagram Alir Pembentukan Model *hybrid* ARIMA-SVR

BAB IV HASIL DAN PEMBAHASAN

4.1 Identifikasi Pola Data

Tahap pertama yang harus dilakukan dalam melakukan analisis deret waktu adalah identifikasi pola data. Identifikasi pola data dilakukan untuk mengetahui gambaran secara umum data yang akan dianalisis. Identifikasi pola data dapat dilakukan dengan membuat plot data deret waktu. Adapun plot deret waktu dari data penjualan klinker bulanan di PT X pada bulan Januari 2016 hingga Juni 2020 dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 Plot Data Penjualan Klinker

Berdasarkan Gambar 4.1 dapat dilihat bahwa data penjualan klinker PT X memiliki pola *trend* kuadratik, dimana jumlah penjualan klinker mengalami kenaikan tertinggi yang terjadi pada periode ke-23 yaitu bulan November 2017. Permintaan ekspor klinker meningkat pada bulan November karena pada musim penghujan permintaan semen domestik menurun sehingga pabrik memiliki persediaan klinker yang cukup untuk dilakukan ekspor. Berdasarkan identifikasi pola data penjualan klinker yang memiliki pola *trend* kuadratik dan

hanya terdiri dari satu variabel deret waktu sehingga metode peramalan yang digunakan adalah model ARIMA.

4.2 Pemodelan Penjualan Klinker dengan Model ARIMA

Pemodelan ARIMA dilakukan dengan beberapa tahap yaitu pengujian stasioneritas data, identifikasi model, pendugaan parameter, diagnostik model, dan pemilihan model terbaik. Adapun penjelasan langkah-langkah pembentukan model ARIMA (p,d,q) yaitu pada penjelasan berikut.

4.2.1 Pengujian Stasioneritas Data

Asumsi yang harus dipenuhi sebelum melakukan analisis deret waktu adalah stasioneritas data. Pengujian stasioneritas perlu dilakukan untuk mengetahui apakah data memiliki penyebaran yang stabil atau tidak yaitu data bersifat konstan sepanjang waktu dan tidak ada perubahan nilai tengah seiring dengan pertambahan waktu. Terdapat dua asumsi stasioneritas data deret waktu yaitu stasioneritas terhadap ragam dan stasioneritas terhadap rata-rata.

Data deret waktu dikatakan stasioner terhadap ragam apabila data berfluktuasi dengan ragam konstan dari waktu ke waktu. Stasioneritas data terhadap ragam dapat dilihat dari *rounded value* (λ), apabila nilai λ sama dengan satu atau angka satu terletak di dalam selang, maka dapat dikatakan bahwa data telah stasioner terhadap ragam. Pada data penjualan klinker diperoleh λ sebesar 1.999924 sehingga perlu dilakukan transformasi *Box-Cox* sesuai persamaan (2.1). Adapun transformasi yang dilakukan dapat dilihat pada Tabel 4.1 berikut.

Tabel 4.1 Hasil Transformasi *Box-Cox*

Transformasi	Lambda (λ)
$Y_t^{(1.999924)}$	0.8808438
$Y_t^{(0.8808438)}$	1.048259

Berdasarkan Tabel 4.1 dapat diketahui bahwa data penjualan klinker pada transformasi pertama menghasilkan *rounded value* (λ) sebesar 0.8808438 ($\lambda < 1$) sehingga perlu dilakukan transformasi kedua. Pada transformasi yang kedua menghasilkan *rounded value* (λ) sebesar 1.048259 ($\lambda > 1$) maka dapat disimpulkan data telah stasioner terhadap ragam.

Selanjutnya dilakukan pengujian stasioneritas data terhadap rata-rata menggunakan uji *Augmented Dicky Fuller* (ADF) sesuai pada



persamaan (2.2). Data penjualan klinker dapat dikatakan stasioner apabila hipotesis nol pada uji ADF ditolak, dengan kriteria jika $p\text{-value} < (\alpha = 0.05)$. Adapun hipotesis dan hasil uji ADF dari data penjualan klinker secara lengkap terdapat pada Lampiran 2 dan ditunjukkan secara ringkas pada Tabel 4.2 sebagai berikut.

$H_0: \delta = 1$ (data tidak stasioner) vs

$H_1: \delta < 1$ (data stasioner)

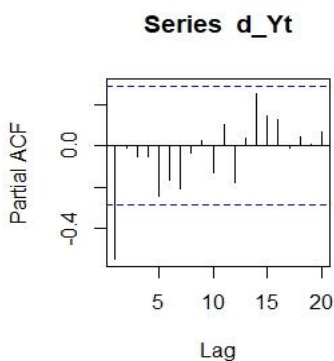
Tabel 4.2 Hasil Uji ADF

Data	P-value	Keterangan
Y_t	0.06137	Tidak stasioner
Pembedaan pertama	0.03035	Stasioner

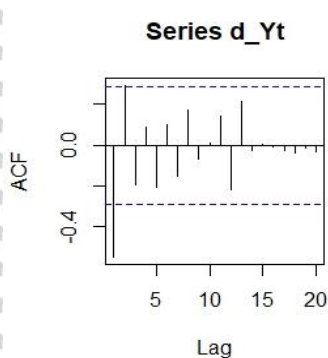
Berdasarkan Tabel 4.2 dapat diketahui bahwa data penjualan klinker sebelum dilakukan *pembedaan* tidak stasioner terhadap rata-rata karena $p\text{-value}$ lebih dari 0.05, sehingga data perlu dilakukan *pembedaan*. Setelah dilakukan *pembedaan* satu kali, diperoleh $p\text{-value}$ kurang dari 0.05 sehingga data sudah stasioner terhadap rata-rata.

4.2.2 Identifikasi Model ARIMA (p, d, q)

Tahap awal dalam membentuk model ARIMA (p, d, q) adalah identifikasi model dengan melihat plot PACF dan ACF terhadap *lag* pada data stasioner sesuai dengan subbab (2.4.1). Untuk menentukan orde (p, d, q) dapat menggunakan acuan pada Tabel 2.2. Adapun plot PACF dan ACF berturut-turut dapat dilihat pada Gambar 4.2 dan Gambar 4.3 berikut.



Gambar 4.2 (a) Hasil Plot PACF



Gambar 4.2 (b) Hasil Plot ACF



Berdasarkan Gambar 4.2 (a) dan (b) dapat diketahui bahwa nilai PACF menunjukkan *lag* yang keluar dari batas (nyata) adalah *lag* ke-1 sedangkan nilai ACF menunjukkan *lag* yang keluar dari batas (nyata) adalah *lag* ke-1 dan 2 pada data hasil *pembedaan* pertama sehingga diperoleh orde $p=1$; $d=1$; $q=2$. Selanjutnya untuk memperoleh model terbaik maka dibentuk model tentatif yaitu ARIMA(1,1,2), ARIMA(1,1,1), ARIMA(0,1,2), ARIMA(0,1,1), dan ARIMA(1,1,0).

4.2.3 Pendugaan Parameter

Setelah diperoleh model tentatif dari hasil identifikasi model ARIMA menggunakan plot PACF dan ACF, selanjutnya dapat dilakukan pendugaan parameter dan uji signifikansi parameter. Pendugaan parameter menggunakan metode *Maximum Likelihood Estimation* (MLE) sesuai pada persamaan (2.20). Adapun ringkasan hasil pendugaan dan uji signifikansi parameter model tentatif dapat dilihat pada Tabel 4.3 dan hasil secara lengkap dapat dilihat pada Lampiran 3.

Tabel 4.3 Pendugaan dan Uji Signifikansi Parameter Model ARIMA (p, d, q)

Model	Parameter	Penduga	<i>p-value</i>	Keterangan
ARIMA(1,1,2)	ϕ_1	-0.7689	0.0244	Signifikan
	θ_1	0.1988	0.5942	Tidak Signifikan
	θ_2	-0.2219	0.5204	Tidak Signifikan
ARIMA(1,1,1)	ϕ_1	-0.5379	0.0241	Signifikan
	θ_1	-0.0168	0.9540	Tidak Signifikan
ARIMA(0,1,2)	θ_1	-0.5042	0.0012	Signifikan
	θ_2	0.2004	0.4204	Tidak Signifikan
ARIMA(0,1,1)	θ_1	-0.5939	0.0003	Signifikan
ARIMA(1,1,0)	ϕ_1	-0.5496	0.0000	Signifikan

Berdasarkan Tabel 4.3 dapat diketahui bahwa model yang semua parameternya signifikan yaitu model ARIMA(0,1,1) dan

ARIMA(1,1,0) karena parameter pada setiap model memiliki p -value lebih kecil dari $\alpha = 0.05$.

4.2.4 Diagnostik Model

Diagnostik model dilakukan untuk memeriksa apakah sisaan berdistribusi normal dan bersifat saling bebas (*white noise*) dengan demikian diagnostik model dapat memastikan bahwa model layak. Pengujian sisaan bersifat *white noise* dilakukan menggunakan uji *Ljung-Box* sesuai persamaan (2.23) dan pengujian normalitas sisaan dilakukan menggunakan uji *Jarque Bera* sesuai persamaan (2.24). Adapun hipotesis dan pengujian secara lengkap dapat dilihat pada Lampiran 4 dan ringkasan hasil pengujian dapat dilihat pada Tabel 4.4 sebagai berikut.

Hipotesis uji *Ljung-Box*

$H_0 : \rho_1 = \rho_2 = \dots = \rho_k = 0$ (*white noise*) vs

H_1 : minimal ada satu nilai $\rho_k \neq 0$ dengan $k = 1, 2, \dots, k$ (tidak *white noise*)

Hipotesis uji *Jarque Bera*

$H_0 : \varepsilon_i \sim N(\mu, \sigma^2)$ Sisaan mengikuti distribusi normal

vs

$H_1 : \varepsilon_i \not\sim N(\mu, \sigma^2)$ Sisaan tidak mengikuti distribusi normal

Tabel 4.4 Hasil Uji *Ljung-Box* dan Uji *Jarque Bera*

Model	Uji <i>Ljung-Box</i>		Uji <i>Jarque Bera</i>	
	p -value	Keterangan	p -value	Keterangan
ARIMA(1,1,2)	0.2604	<i>White noise</i>	0.0000	Tidak Normal
ARIMA(1,1,1)	0.4543	<i>White noise</i>	0.0000	Tidak Normal
ARIMA(0,1,2)	0.2888	<i>White noise</i>	0.0143	Tidak Normal
ARIMA(0,1,1)	0.1747	<i>White noise</i>	0.1580	Normal
ARIMA(1,1,0)	0.5769	<i>White noise</i>	0.0000	Tidak Normal

Berdasarkan Tabel 4.4 pada uji *Ljung-Box* dari lima model tentatif semua memenuhi asumsi *white noise* karena semua model mempunyai p -value lebih besar dari $\alpha = 0.05$ sehingga sisaan bersifat



white noise atau saling bebas. Selain itu pada uji *Jarque Bera* hanya satu model yang memenuhi asumsi normalitas yaitu model ARIMA(0,1,1) sedangkan empat model lainnya tidak memenuhi asumsi normalitas.

4.2.5 Pemilihan Model Terbaik

Pemilihan model terbaik dari delapan model tentatif yang dihasilkan yaitu berdasarkan hasil uji signifikansi parameter, diagnostik model, dan nilai AIC yang dihasilkan. Adapun ringkasan hasil pengujian asumsi yang telah dilakukan dapat dilihat pada Tabel 4.5 berikut.

Tabel 4.5 Hasil Pengujian Asumsi

Model	Parameter	Normalitas	White noise	AIC
ARIMA(1,1,2)	ϕ_1 (Signifikan)	Tidak Normal	White noise	2024.55
	θ_1 (Tidak Signifikan)			
	θ_2 (Tidak Signifikan)			
ARIMA(1,1,1)	ϕ_1 (Signifikan)	Tidak Normal	White noise	2024.79
	θ_1 (Tidak Signifikan)			
ARIMA(0,1,2)	θ_1 (Signifikan)	Tidak Normal	White noise	2026.40
	θ_2 (Tidak Signifikan)			
ARIMA(0,1,1)	θ_1 (Signifikan)	Normal	White noise	2022.79
ARIMA(1,1,0)	ϕ_1 (Signifikan)	Tidak Normal	White noise	2022.80

Berdasarkan Tabel 4.5 dapat dilihat bahwa model yang memiliki parameter signifikan adalah ARIMA(0,1,1) dan ARIMA(1,1,0). Adapun model yang memiliki nilai AIC terkecil adalah ARIMA(0,1,1). Sedangkan model ARIMA(1,1,0) memiliki selisih nilai AIC sangat kecil dengan model ARIMA(0,1,1) sehingga kedua model tersebut dapat dilakukan pemodelan *hybrid*.



4.3 Pemodelan Penjualan Klinker dengan Model *Hybrid* ARIMA-SVR

Pemodelan *hybrid* ARIMA-SVR dilakukan menggunakan model ARIMA yang memiliki parameter signifikan yaitu model ARIMA(0,1,1) dan ARIMA(1,1,0). Adapun langkah-langkah pembentukan model *hybrid* ARIMA-SVR yaitu pada penjelasan berikut.

4.3.1 Retransformasi Model ARIMA

Pemodelan *hybrid* menggunakan input residual model ARIMA yang diperoleh sehingga perlu dilakukan perhitungan residual dari model ARIMA. Sebelum dilakukan perhitungan residual, maka dilakukan retransformasi terhadap *fitted value* model ARIMA yang terbentuk. Retransformasi dilakukan untuk mengembalikan hasil transformasi *Box-Cox* yang dilakukan pada pembentukan model ARIMA. Adapun retransformasi dilakukan melalui *invers Box-Cox* menggunakan *rounded value* (λ) hasil transformasi pertama yaitu sebesar 0.8808438 kemudian dilanjutkan menggunakan *rounded value* (λ) sebelum dilakukan transformasi yaitu sebesar 1.999924.

4.3.2 Hasil Pengujian Nonlinieritas

Sebelum pembentukan model *hybrid* ARIMA-SVR, maka dilakukan pengujian nonlinieritas untuk mengetahui apakah terdapat hubungan nonlinier pada residual model ARIMA. Uji nonlinieritas menggunakan uji *Ramsey RESET* dengan input residual model linier. Pengujian nonlinieritas dilakukan pada model linier yang diperoleh yaitu ARIMA(0,1,1) dan ARIMA(1,1,0). Adapun ringkasan hasil uji nonlinieritas data penjualan klinker dapat dilihat pada Tabel 4.6 dan hasil pengujian secara lengkap terdapat pada Lampiran 5.

Tabel 4.6 Hasil Uji *Ramsey RESET*

Model	<i>p-value</i>	Keterangan
ARIMA(0,1,1)	0.0429	Linier
ARIMA(1,1,0)	0.2443	Linier

Berdasarkan Tabel 4.6 diperoleh hasil uji *Ramsey RESET* data penjualan klinker pada model ARIMA(0,1,1) dan ARIMA(1,1,0) terima H_0 yaitu *p-value* lebih besar dari ($\alpha = 0.05$) dimana



menunjukkan bahwa data linier. Meskipun hasil uji *Ramsey RESET* menunjukkan bahwa tidak terdapat hubungan nonlinier pada residual, model tersebut masih belum cukup karena masih terdapat korelasi linier yang ada pada residual sehingga hubungan nonlinier belum dimodelkan dengan tepat (Zhang, 2003). Dengan demikian, pada penelitian ini pembentukan model *hybrid* tetap dilakukan untuk menangkap hubungan nonlinier pada residual yang tidak terdeteksi pada pengujian *Ramsey RESET* sehingga dapat menghasilkan model yang baik. Adapun model yang baik yaitu model yang dekat dengan data aktual atau yang menghasilkan nilai akurasi tinggi.

4.3.3 Pembentukan Model *Hybrid* ARIMA-SVR

Model *hybrid* yang digunakan adalah model *Support Vector Regression* (SVR) karena model tersebut dapat digunakan baik pada data yang memiliki hubungan linier maupun nonlinier pada residual. Adapun pemodelan SVR menggunakan input residual model linier dan *lag*-nya. Penentuan *lag* yang digunakan sebagai input model SVR berdasarkan plot PACF yang signifikan. Berdasarkan plot PACF, *lag* yang signifikan adalah *lag* ke-1 sehingga input model SVR yaitu menggunakan residual dan *lag* ke-1. Residual masing-masing model tentatif yang diperoleh dapat dilihat pada Lampiran 6 dan Lampiran 7.

Pemodelan SVR dilakukan menggunakan fungsi kernel. Adapun fungsi kernel yang digunakan dalam penelitian ini yaitu *Radial Basis Function* (RBF) sesuai persamaan (2.45). Terdapat beberapa parameter dalam fungsi kernel RBF yaitu *epsilon*, *cost* (C), dan *gamma*. Parameter tersebut perlu dioptimalkan agar mendapatkan hasil estimasi yang optimum. Metode optimasi yang digunakan adalah metode *grid search* sesuai penjelasan pada anak sub bab (2.6.3) untuk memperoleh kombinasi ketiga parameter dengan nilai MAPE yang terkecil. Parameter *epsilon* untuk semua model tentatif yang signifikan digunakan rentang nilai antara 0.0001 sampai dengan 0.001 dengan selisih 0.0001. Parameter *cost* untuk semua model tentatif yang signifikan digunakan rentang nilai antara 300 sampai dengan 600 dengan selisih 100. Parameter *gamma* untuk semua model tentatif yang signifikan digunakan rentang nilai antara 0.5 sampai dengan 0.7 dengan selisih 0.1. Adapun hasil optimasi parameter dan nilai MAPE yang diperoleh model dengan kombinasi parameter yang ditentukan dapat dilihat pada Tabel 4.7.



Tabel 4.7 Hasil Optimasi Parameter

Model	<i>Epsilon</i>	<i>Cost</i>	<i>Gamma</i>	MAPE
ARIMA(0,1,1)-SVR	0.0002	300	0.7	8.15%
ARIMA(1,1,0)-SVR	0.0002	300	0.7	7.56%

Berdasarkan Tabel 4.7 terlihat bahwa kombinasi parameter untuk model ARIMA(0,1,1) dan ARIMA(1,1,0) yaitu parameter *epsilon* sebesar 0.0002 yang menunjukkan bahwa batas rentang toleransi kesalahan atau lebar zona-*insensitive* sebesar 0.0002. Parameter *cost* sebesar 300 yang berarti bahwa batas atas deviasi dapat ditoleransi dengan parameter *cost* sebesar 300. Parameter *gamma* sebesar 0.7 yang berarti bahwa data *training* dipetakan dalam ruang fitur yaitu sebesar 0.7. Kombinasi ketiga parameter tersebut menghasilkan nilai MAPE pada model ARIMA(0,1,1)-SVR sebesar 8.15% dan ARIMA(1,1,0)-SVR sebesar 7.56%.

4.4 Pemilihan Model *Hybrid* Terbaik

Pemilihan model *hybrid* terbaik dalam penelitian ini menggunakan kriteria nilai MAPE. Model yang baik adalah model yang menghasilkan nilai MAPE kurang dari 10% baik pada data *testing* maupun data *training*. Adapun ringkasan nilai MAPE model *hybrid* dapat dilihat pada Tabel 4.8 berikut.

Tabel 4.8 Nilai MAPE Model *Hybrid* pada Data *Training* dan *Testing*

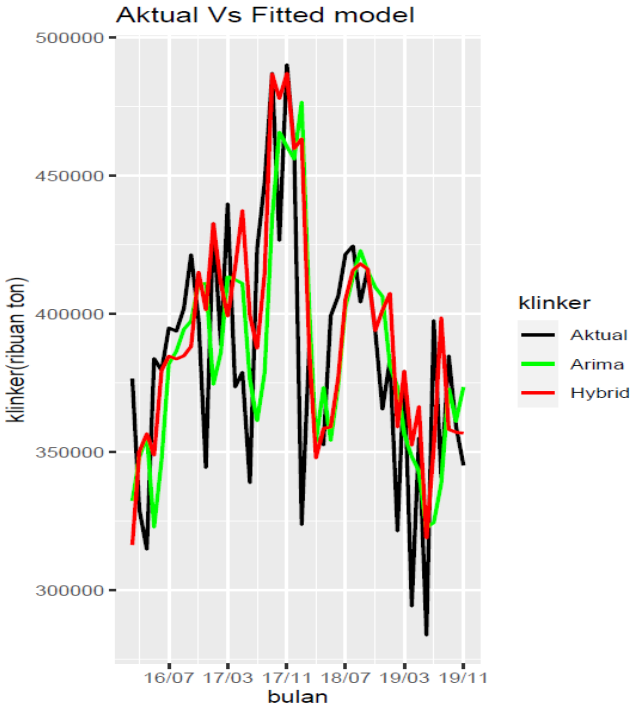
Model	<i>Training</i>	<i>Testing</i>
ARIMA(0,1,1)	9.17%	9.11%
ARIMA(0,1,1)-SVR	8.15%	9.66%
ARIMA(1,1,0)	8.30%	9.16%
ARIMA(1,1,0)-SVR	7.56%	9.02%

Berdasarkan Tabel 4.8 dapat diketahui bahwa pada data *training* model ARIMA(0,1,1)-SVR yang terbentuk memiliki nilai MAPE kurang dari 10% dan memiliki nilai MAPE lebih kecil daripada model ARIMA. Namun, pada data *testing* nilai MAPE tidak lebih kecil dari model ARIMA sehingga model ARIMA(0,1,1)-SVR tidak menunjukkan performansi lebih baik daripada model ARIMA.

Sedangkan model ARIMA(1,1,0)-SVR pada data *training* maupun data *testing* memiliki nilai MAPE kurang dari 10%. Selain

itu, model ARIMA(1,1,0)-SVR baik pada data *training* maupun data *testing* memiliki nilai MAPE lebih kecil daripada model ARIMA, hal ini menunjukkan bahwa model *hybrid* memiliki akurasi sangat baik untuk digunakan peramalan daripada model tunggal.

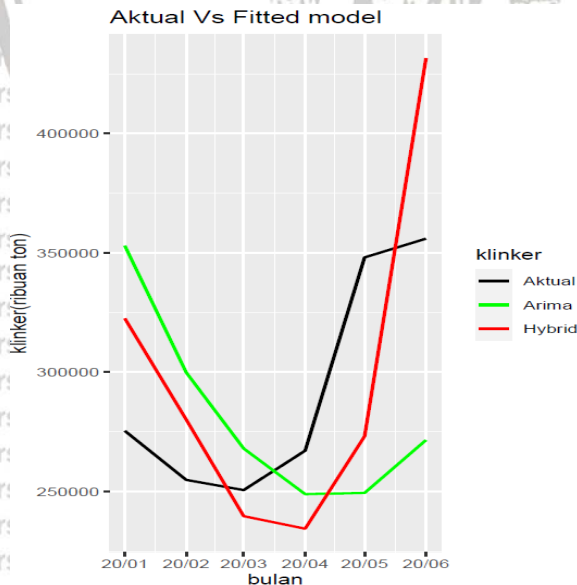
Dengan demikian, model *hybrid* terbaik adalah ARIMA(1,1,0)-SVR untuk meramalkan jumlah penjualan klinker. Adapun plot data aktual dan prediksi untuk model ARIMA(1,1,0) dan model ARIMA(1,1,0)-SVR baik data *training* maupun data *testing* masing-masing dapat dilihat pada Gambar 4.3 dan Gambar 4.4 serta hasil ramalan data *testing* dapat dilihat pada Tabel 4.9



Gambar 4.3 Plot Data Aktual dan Prediksi Model *Hybrid* pada Data *Training*

Tabel 4.9 Data Aktual dan Hasil Prediksi pada Data *Testing*

Bulan	Data Aktual (Ribuan Ton)	Hasil Prediksi ARIMA (Ribuan Ton)	Hasil Prediksi ARIMA-SVR (Ribuan Ton)
Januari 2020	275410.6	353049.7	322572.7
Februari 2020	254832	299889.9	280147.5
Maret 2020	250566	267990.7	239631.4
April 2020	267083	248867.5	234371.9
Mei 2020	348091.2	249367.1	273119.4
Juni 2020	355955	271549.5	431655.1



Gambar 4.4 Plot Data Aktual dan Prediksi Model *Hybrid* pada Data *Testing*

Berdasarkan Gambar 4.3 dan Gambar 4.4 dapat dilihat bahwa plot data hasil prediksi yang paling mendekati data aktual adalah plot hasil prediksi model *hybrid* ARIMA(1,1,0)-SVR. Hasil plot prediksi data *testing* baik pada model ARIMA maupun model *hybrid* sensitive apabila terjadi penurunan dan kenaikan pada data aktual. Hal ini dapat



dilihat pada plot data *testing* ketika terjadi penurunan atau kenaikan nilai aktual maka diikuti penurunan atau kenaikan pada data prediksi secara drastis yaitu penurunan pada bulan Januari dan kenaikan pada bulan April. Sehingga hasil prediksi bulan Juni baik pada model ARIMA maupun model *hybrid* memiliki selisih sangat jauh dengan nilai aktual, tetapi prediksi model *hybrid* lebih dekat dengan nilai aktual daripada model ARIMA. Dengan demikian model *hybrid* merupakan model terbaik yang dapat digunakan untuk meramalkan jumlah penjualan klinker. Hasil penelitian ini menunjukkan bahwa prediksi penjualan klinker pada saat ini dipengaruhi oleh penjualan klinker satu periode sebelumnya. Adapun jumlah penjualan klinker pada Juni 2020 diprediksi menggunakan model *hybrid* yaitu sebesar 431655.1×10^3 Ton.

4.5 Hasil Ramalan Penjualan Klinker Menggunakan Model *Hybrid* ARIMA(1,1,0)-SVR

Setelah diperoleh model *hybrid* terbaik, selanjutnya dilakukan peramalan jumlah penjualan klinker satu periode ke depan yaitu bulan Juli 2020 menggunakan model *hybrid* ARIMA-SVR. Adapun hasil ramalan penjualan klinker untuk bulan Juli 2020 yaitu sebesar 376640.9×10^3 Ton. Hasil ramalan ini menunjukkan bahwa terdapat penurunan penjualan klinker pada bulan Juli 2020 jika dibandingkan dengan penjualan bulan sebelumnya. Dengan demikian perusahaan dapat mengurangi jumlah produksi klinker yang akan diekspor sehingga tidak terjadi kelebihan stok klinker pada perusahaan.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pembahasan yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut.

1. Model *hybrid* yang terbentuk yaitu model ARIMA(1,1,0)-SVR dengan menggunakan input residual dan residual *lag* ke-1 model linier ARIMA(1,1,0). Adapun estimasi parameter model nonlinier SVR dengan menggunakan kombinasi parameter *epsilon* sebesar 0.0002, parameter *cost* sebesar 300, dan parameter *gamma* sebesar 0.7.
2. Model ARIMA(1,1,0)-SVR menghasilkan nilai MAPE kurang dari 10% yaitu sebesar 7.56% pada data *training* dan 9.02% pada data *testing*. Model *hybrid* ARIMA(1,1,0)-SVR pada data *training* maupun data *testing* memiliki nilai MAPE lebih kecil daripada model tunggal (ARIMA). Dengan demikian peramalan penjualan klinker menggunakan model *hybrid* memiliki akurasi lebih baik daripada model tunggal.
3. Hasil peramalan jumlah penjualan klinker satu periode ke depan yaitu bulan Juli 2020 sebesar 376640.9×10^3 Ton. Hasil ramalan ini menunjukkan bahwa terdapat penurunan penjualan klinker pada bulan Juli 2020 jika dibandingkan dengan penjualan bulan sebelumnya.

5.2 Saran

Adapun saran untuk penelitian selanjutnya yaitu pada penelitian ini tidak memperhatikan adanya faktor lain yang mempengaruhi jumlah penjualan klinker. Oleh karena itu, untuk penelitian selanjutnya dapat menambahkan variabel eksogen yaitu dengan menambahkan faktor inflasi atau faktor ekonomi lain yang dapat mempengaruhi jumlah penjualan klinker.





(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- Box, G. E. P., dan Jenkins, G. M. (1976). *Time series analysis: forecasting and control*. San Francisco: Holden-Day.
- Box, G. E. P., dan Cox, D. R. (1964). An Analysis of Transformation. *Journal of The Royal Statistical Society*. 26(2): 211-252.
- Chen, K. Y., dan Wang, C. H. (2007). A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan. *Expert Systems with Applications*, 32(1), 254–264.
- Cryer, J. D., and Chan, K. 2008. *Time Series Analysis*. USA: Springer.
- Enders, W. 2004. *Applied Econometric Time Series*. John Wiley and Son, Inc.: New York.
- Faruk, D.O. 2010. A hybrid neural network and ARIMA model for water quality time series prediction. *Eng App Intelligence*. 23: 586-594.
- Himayati. 2008. *Eksplorasi Zahir Accounting*. Jakarta: Elex Media Komputindo.
- Hsu, C.W., et al., 2004. A Practical Guide to Support Vektor Classification, Department of Computer Science and Information Engineering, National Taiwan University.
- Khashei, M., dan Bijari, M. (2010). An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with applications*. 37(1): 479–489.
- Kim, T.H., Lee., Y.S. and Newbold, P., 2004. *Spurious Nonlinear Regressions in Econometrics*, working paper, School of Economics, University of Nottingham, Nottingham NG7 2RD, UK.
- Leidiyana, H., 2013. Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Risiko Kredit Kepemilikan Kendaraan Bermotor, Jurnal Penelitian Ilmu Komputer, *Sistem Embedded & Logic*. 1(1): 65-76.
- Mahardhika, A. D. dan Susanto, N., 2017. Peramalan Perencanaan Produksi Terak Dengan Metode Exponential Smoothing With Trend pada PT. Semen Indonesia (Persero) Tbk, "Jurnal Teknik Industri, Universitas Diponegoro. 6(1):1-10.
- Makridakis, S., Wheelwright, S.C., dan Mcgee, V.E. 1999. *Metode dan Aplikasi Peramalan*, Edisi kedua. Alih bahasa Ir. Hari Suminto. Binarupa Aksara: Jakarta.



- Makridakis, S., dan Hibon, M. (2000). The M3-Competition: results, conclusions and implications. *International journal of forecasting*. 16(4): 451–476.
- Munarsih, E., 2011. Penerapan Model ARIMA Neural Network Hybrid untuk Peramalan Time Series. Thesis. S2 Matematika FMIPA Universitas Gadjah Mada. Yogyakarta.
- Pai, P. F., dan Lin, C. S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*. 33(6): 497–505.
- Rahamdayanti, R., Susilo, B., dan Puspitaningrum, D., 2015. Perbandingan Keakuratan Metode Autoregressive Integrated Moving Average dan Exponential Smoothing pada Peramalan Penjualan Semen. *Jurnal Rekursif*. Fakultas Teknik Universitas Bengkulu. 3(1): 23-36.
- Ramsey. 1969. Tests for Specification Errors in Classical Linear Least Squares Regression Analysis, *Journal of the Royal Statistical Society*, vol. *Series B*, no. 31, pp. 350-371.
- Ren, L., and Glasure, Y. (2009). Aplicableity of the Revised Mean Absolute Percentage Errors (MAPE) Approach to Some Popular Normal and Non-Normal Independent Time Series. *Advances in Economic Research*, 15: 409-420
- Sen, P., Roy, M., and Pal, P., 2016 Application of ARIMA for Forecasting Energy Consumption and CHG Emission: a Case Study of an Indian Pig Iron Manufacturing Organization. *Energy Journal*. 116: 1031-1038.
- Shabri, A., dan Suhartono. (2012). Streamflow forecasting using least-squares support vector machines. *Hydrological Sciences Journal*. 57(7): 1275–1293.
- Sprung. 2012 "Cement" dalam Ullmann's Encyclopedia of Industrial Chemistry, Wiley-VCH, Weinheim.
- Warsito, D. I. 2004. Uji Linearitas Data Time Series dengan RESET Test, *Jurnal Matematika dan Komputer*. 7(3): 36-44, Desember 2004, ISSN: 1410-8518. (Online). Jurusan matematika FMIPA Universitas Diponegoro. (<https://media.neliti.com/media/publications/117191-ID-uji-linieritas-data-time-series-dengan-r.pdf>), diakses pada 10 Desember 2020.
- Wei, W. W. S. (2006). *Time series analysis: univariate and multivariate methods*. Pearson Addison Wesley.



Yasin, H., Prahutama, A., & Utami, T. (2014). Prediksi Harga Saham Menggunakan Support Vector Regression dengan Algoritma Grid Search. *Media Statistika*. 7(1).

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*. 50: 159–175.





(Halaman ini sengaja dikosongkan)

LAMPIRAN

Lampiran 1. Data Jumlah Penjualan Klinker Bulan Januari 2016 sampai Juni 2020.

Bulan	klinker	Bulan	klinker
Januari 2016	323061	April 2018	352609
Februari 2016	376641	Mei 2018	399278
Maret 2016	329242	Juni 2018	406466
April 2016	314804	Juli 2018	421585
Mei 2016	383862	Agustus 2018	424636
Juni 2016	379522	September 2018	404284
Juli 2016	394945	Oktober 2018	416223
Agustus 2016	393762	November 2018	394010
September 2016	401962	Desember 2018	365502
Oktober 2016	421488	Januari 2019	383537
November 2016	397878	Februari 2019	321403
Desember 2016	344365	Maret 2019	379297
Januari 2017	432750	April 2019	294212
Februari 2017	388802	Mei 2019	355136
Maret 2017	439908	Juni 2019	283697
April 2017	373564	Juli 2019	397612
Mei 2017	378889	Agustus 2019	340800
Juni 2017	338888	September 2019	384859
Juli 2017	423724	Oktober 2019	359406
Agustus 2017	447767	November 2019	345172
September 2017	487159	Desember 2019	300867
Oktober 2017	426601	Januari 2020	275410.6
November 2017	490284	Februari 2020	254832
Desember 2017	459802	Maret 2020	250566
Januari 2018	323727	April 2020	267083
Februari 2018	387620	Mei 2020	348091.2
Maret 2018	355465	Juni 2020	355955



Lampiran 2. Uji *Augmented Dickey Fuller* (ADF)

```
> adf.test(train2)
```

Augmented Dickey-Fuller Test

```
data: train2  
Dickey-Fuller = -3.4417, Lag order = 3, p-value = 0.0613  
7  
alternative hypothesis: stationary
```

```
> adf.test(diff1)
```

Augmented Dickey-Fuller Test

```
data: diff1  
Dickey-Fuller = -3.7637, Lag order = 3, p-value = 0.0303  
5  
alternative hypothesis: stationary
```



Lampiran 3. Pendugaan dan Uji Signifikansi Parameter Model

ARIMA (p, d, q)

```
> #====PENDUGAAN PARAMETER DAN PENGUJIAN HIPOTESIS====#  
> library(forecast)  
> m1 = arima(train2, c(1,1,2), include.mean = T);m1
```

Call:

```
arima(x = train2, order = c(1, 1, 2), include.mean = T)
```

Coefficients:

	ar1	ma1	ma2
	-0.7689	0.1988	-0.2219
s.e.	0.3418	0.3732	0.3454

```
sigma^2 estimated as 6.626e+17: log likelihood = -1009.27, aic = 2026.55
```

```
> m2 = Arima(train2, c(1,1,1), include.mean = T);m2
```

Series: train2

ARIMA(1,1,1)

Coefficients:

	ar1	ma1
	-0.5379	-0.0168
s.e.	0.2386	0.2918

```
sigma^2 estimated as 6.97e+17: log likelihood=-1009.4  
AIC=2024.79 AICc=2025.37 BIC=2030.28
```

```
> m3 = Arima(train2, c(0,1,2), include.mean = T);m3
```

Series: train2

ARIMA(0,1,2)

Coefficients:

	ma1	ma2
	-0.5042	0.2004
s.e.	0.1557	0.2487

```
sigma^2 estimated as 7.231e+17: log likelihood=-1010.2  
AIC=2026.4 AICc=2026.97 BIC=2031.88
```

```
> m4 = arima(train2, c(0,1,1), include.mean = T);m4
```

Call:

```
arima(x = train2, order = c(0, 1, 1), include.mean = T)
```

Coefficients:

	ma1
	-0.5939
s.e.	0.1657

```
sigma^2 estimated as 6.952e+17: log likelihood = -1010.4,  
aic = 2024.79
```


Lampiran 3. Pendugaan dan Uji Signifikansi Parameter Model ARI
MA (p, d, q) (Lanjutan)

```
> m5 = Arima(train2, c(1,1,0), include.mean = T);m5  
Series: train2  
ARIMA(1,1,0)
```

```
Coefficients:  
          ar1  
      -0.5496  
s.e.      0.1217
```

```
sigma^2 estimated as 6.816e+17: log likelihood=-1009.4  
AIC=2022.8  AICC=2023.08  BIC=2026.46
```



Lampiran 4. Uji Diagnostik Model

```
> #=====UJI WHITE NOISE=====#  
> coeftest(m1)
```

```
z test of coefficients:
```

	Estimate	Std. Error	z value	Pr(> z)	
ar1	-0.76893	0.34180	-2.2496	0.02447	*
ma1	0.19876	0.37315	0.5327	0.59426	
ma2	-0.22195	0.34536	-0.6427	0.52045	

```
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
' 1
```

```
> coeftest(m2)
```

```
z test of coefficients:
```

	Estimate	Std. Error	z value	Pr(> z)	
ar1	-0.537937	0.238553	-2.2550	0.02413	*
ma1	-0.016798	0.291761	-0.0576	0.95409	

```
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
' 1
```

```
> coeftest(m3)
```

```
z test of coefficients:
```

	Estimate	Std. Error	z value	Pr(> z)	
ma1	-0.50417	0.15571	-3.2378	0.001204	**
ma2	0.20038	0.24870	0.8057	0.420411	

```
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
' 1
```

```
> coeftest(m4)
```

```
z test of coefficients:
```

	Estimate	Std. Error	z value	Pr(> z)	
ma1	-0.59386	0.16572	-3.5835	0.000339	***

```
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
' 1
```



Lampiran 4. Uji Diagnostik Model (Lanjutan)

```
> coeftest(m5)
```

```
z test of coefficients:
```

```
      Estimate Std. Error z value Pr(>|z|)
ar1 -0.54965    0.12167  -4.5174 6.262e-06 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> #=====UJI NORMALITAS GALAT=====
```

```
> library(tseries)
```

```
> jarque.bera.test(m1$residuals)
```

```
      Jarque Bera Test
```

```
data: m1$residuals
```

```
X-squared = 20.972, df = 2, p-value = 2.792e-05
```

```
> jarque.bera.test(m2$residuals)
```

```
      Jarque Bera Test
```

```
data: m2$residuals
```

```
X-squared = 19.614, df = 2, p-value = 5.507e-05
```

```
> jarque.bera.test(m3$residuals)
```

```
      Jarque Bera Test
```

```
data: m3$residuals
```

```
X-squared = 8.4919, df = 2, p-value = 0.01432
```

```
> jarque.bera.test(m4$residuals)
```

```
      Jarque Bera Test
```

```
data: m4$residuals
```

```
X-squared = 3.6905, df = 2, p-value = 0.158
```

```
> jarque.bera.test(m5$residuals)
```

```
      Jarque Bera Test
```

```
data: m5$residuals
```

```
X-squared = 20.288, df = 2, p-value = 3.931e-05
```



Lampiran 5 Hail Uji Nonlinieritas Ramsey RESET

```
M4= MODEL ARIMA(0,1,1)
```

```
> resettest(train~et.m4-et1.m4, power=29, type="regressor", data=N)
```

```
RESET test
```

```
data: train ~ et.m4 - et1.m4
```

```
RESET = 1.5871, df1 = 2, df2 = 42, p-value = 0.2165
```

```
M5= MODEL ARIMA(1,1,0)
```

```
> resettest(train~yt1.m5+et.m5, power=2, type="regressor", data=N)
```

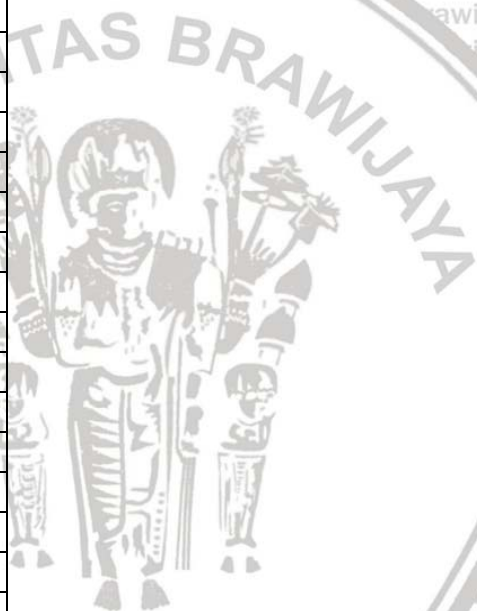
```
RESET test
```

```
data: train ~ yt1.m5 + et.m5
```

```
RESET = 1.9494, df1 = 2, df2 = 41, p-value = 0.1553
```

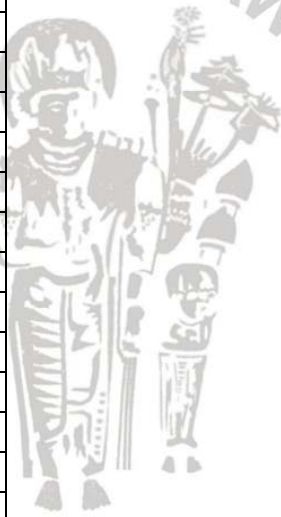
Lampiran 6. Residual Model ARIMA (0,1,1)

No	Residual	No	Residual
1	1.83E+02	25	-1.36E+05
2	4.57E+04	26	-2.11E+04
3	-2.36E+04	27	-4.48E+04
4	-2.76E+04	28	-2.99E+04
5	5.24E+04	29	2.87E+04
6	2.59E+04	30	2.40E+04
7	3.06E+04	31	2.92E+04
8	1.67E+04	32	2.02E+04
9	1.81E+04	33	-8.43E+03
10	3.02E+04	34	6.91E+03
11	-5.90E+03	35	-1.81E+04
12	-5.70E+04	36	-3.93E+04
13	5.37E+04	37	-5.69E+03
14	-1.27E+04	38	-6.55E+04
15	4.35E+04	39	1.79E+04
16	-4.09E+04	40	-7.45E+04
17	-1.94E+04	41	1.51E+04
18	-5.16E+04	42	-6.25E+04
19	5.35E+04	43	7.57E+04
20	5.52E+04	44	-1.34E+04
21	7.15E+04	45	3.61E+04
22	-1.92E+04	46	-4.36E+03
23	5.22E+04	47	-1.68E+04
24	-9.88E+00		



Lampiran 7. Residual Model ARIMA (1,1,0)

No	Residual	No	Residual
1	183.4278	25	-153013
2	44312.1	26	-15277.3
3	-18726.6	27	1874.364
4	-41087.2	28	-20791.1
5	61061.13	29	45097.75
6	32321.61	30	32290.55
7	13033.43	31	19058.09
8	7236.083	32	11308.78
9	7548.739	33	-18677.5
10	24017.79	34	658.4419
11	-12965.4	35	-15683.5
12	-66618.3	36	-40831.1
13	58249.06	37	2164.533
14	2722.742	38	-52303.9
15	26509.34	39	22718.12
16	-38851.3	40	-54170.5
17	-32153.6	41	12156.09
18	-37081.3	42	-39037.2
19	62431.43	43	73158.04
20	68879.98	44	2171.905
21	52481.84	45	12014.08
22	-39220.4	46	-1743.18
23	29645.27	47	-28387.1
24	3683.154		



Lampiran 8. *Source Code* Pemodelan ARIMA (p, d, q)

Menggunakan *Software Rstudio* 3.6.2

```
#=====INPUT DATA=====#
library(readxl)
data.clinker=read.csv("E:/SKRIPSI/data.skrripsi/clinker_bulan.csv",header=TRUE,sep=",");data.clinker
clinker=data.klinker$clinker;clinker
plot.ts(clinker,col="darkblue",lwd = 2.5)
library(TSA)
library(mgcv)
library(locfit)
train=cbind(data.clinker[1:47,]);train
test=cbind(data.clinker[48:54,]);test

#=====STASIONERITAS DATA TERHADAP RAGAM=====#
library(forecast)
BoxCox.lambda(train[,2])
train1=BoxCox(train[,2],1.999924);train1
BoxCox.lambda(train1)
train2=BoxCox(train1, 0.8808438);train2
BoxCox.lambda(train2)

#=====STASIONERITAS DATA TERHADAP RATA-RATA=====#
library(tseries)
adf.test(train2)
diff1=diff(train2);diff1
adf.test(diff1)
acf(diff1)
list(acf(diff1))
pacf(diff1)
list(pacf(diff1))

#=====PENDUGAAN PARAMETER DAN PENGUJIAN HIPOTESIS=====#
library(forecast)
m1 = arima(train2, c(1,1,2), include.mean = T);m1
m2 = Arima(train2, c(1,1,1), include.mean = T);m2
m3 = Arima(train2, c(0,1,2), include.mean = T);m3
m4 = arima(train2, c(0,1,1), include.mean = T);m4
m5 = Arima(train2, c(1,1,0), include.mean = T);m5
```



Lampiran 8. *Source Code* Pemodelan ARIMA (p, d, q) Menggunakan *Software Rstudio* 3.6.2 (Lanjutan)

```
library(lmtest)
coefest(m1)
coefest(m2)
coefest(m3)
coefest(m4)
coefest(m5)
```

```
#####UJI WHITE NOISE#####
```

```
checkresiduals(m1)
checkresiduals(m2)
checkresiduals(m3)
checkresiduals(m4)
checkresiduals(m5)
```

```
#####UJI NORMALITAS GALAT#####
```

```
library(tseries)
jarque.bera.test(m1$residuals)
jarque.bera.test(m2$residuals)
jarque.bera.test(m3$residuals)
jarque.bera.test(m4$residuals)
jarque.bera.test(m5$residuals)
```



Lampiran 9. *Source Code* Pemodelan Hybrid ARIMA (0,1,1)-SVR
Menggunakan *Software Rstudio* 3.6.2

```
#####INPUT DATA#####  
library(readxl)  
data.clinker=read.csv("E:/SKRIPSI/data.skrripsi/+  
clinker_bulan.csv",header=TRUE,sep=",");data.clinker  
clinker=data.klinker$clinker;clinker  
#####MODEL HYBRID ARIMA(0,1,1)-SVR TRAINING#####  
train=cbind(data.clinker[1:47,]);train  
#####ARIMA#####  
#m4  
m4 = Arima(train2, c(0,1,1), include.mean = T);m4  
Yfits =fitted(m4);Yfits  
Yfitsinv1 =InvBoxCox(Yfits, lambda=0.8808438);Yfitsinv1  
Yfitsinv2 =InvBoxCox(Yfitsinv1, lambda=1.999924);Yfitsinv2  
Yfit.arima.trainm4  
=as.numeric(Yfitsinv2[2:47]);Yfit.arima.trainm4  
Yfit.arima.m4=as.numeric(Yfitsinv2);Yfit.arima.m4  
library(MLmetrics)  
mape.arima.train = MAPE(Yfitsinv2,train[,2]);mape.arima.train  
#####UJI NONLINIERITAS#####  
library(Hmisc)  
et.m4 =train[,2]-as.numeric(Yfitsinv2);et.m4  
et1.m4 =Lag(et.m4);et1.m4 #lag 1 et  
N =data.frame(train[,2],et.m4,et1.m4);N  
library(lmtest)  
resetttest(train[,2]~et.m4-et1.m4, power=29, type="regressor",  
data=N)  
#####SVR#####  
et.m4 =train[,2]-as.numeric(Yfitsinv2);et.m4  
library(quantmod)  
et1.m4 =Lag(et.m4);et1.m4 #lag 1 et  
data.trainm4 =data.frame(cbind(et.m4,et1.m4));data.trainm4  
xtrain=data.trainm4[2:47,];xtrain  
library(e1071)  
tuning =tune(svm, et.m4~, data=data.trainm4,  
type='eps-regression', kernel='radial',
```



Lampiran 9. *Source Code* Pemodelan Hybrid ARIMA (0,1,1)-SVR Menggunakan *Software Rstudio* 3.6.2 (Lanjutan)

```
ranges= list(epsilon=seq(0.0001,0.001,0.0001),
cost=seq(300,600,100), gamma=seq(0.5,0.7,0.1))
summary(tuning)
svr.opt.trainm4 =svm(et.m4~,data=data.trainm4,
type = "eps-regression",kernel = "radial",
gamma = 0.7, cost = 300,
epsilon=0.0002);svr.opt.trainm4
Ysvr.trainm4 =predict(svr.opt.trainm4,xtrain) ;Ysvr.trainm4

#####HYBRID#####
fit.hybrid.trainm4=Yfit.arima.trainm4+as.numeric(Ysvr.trainm
4);fit.hybrid.trainm4
library(MLmetrics)
mape.hybrid.trainm4 =
MAPE(fit.hybrid.trainm4,train[2:47,2]);mape.hybrid.trainm4

=====PERAMALAN MODEL HYBRID DATA TESTING=====
test=cbind(data.clinker[48:54,]);test
###ARIMA###
library(forecast)
test1=BoxCox(test[,2],1.999924);test1
test2=BoxCox(test1,0.8808438);test2
m4t = Arima(test2, c(0,1,1), include.mean = T);m4t
Yfits =c(forecast(m4,h=1)$mean,fitted(m4t)[-1]);Yfits
Yfitsinv1 =InvBoxCox(Yfits, lambda=0.8808438);Yfitsinv1
Yfitsinv2 =InvBoxCox(Yfitsinv1, lambda=1.999924);Yfitsinv2
arima.test=as.numeric(Yfitsinv2[2:7])
Lampiran 9. Source Code Pemodelan Hybrid ARIMA (0,1,1)-SVR
Menggunakan Software Rstudio 3.6.2 (Lanjutan)

mape.arima.testm4=
MAPE(as.numeric(Yfitsinv2),test[,2]);mape.arima.testm4
###SVR###
library(Hmisc)
at.m4=test[,2]-as.numeric(Yfitsinv2);at.m4
at1.m4=Lag(at.m4);at1.m4 #data residual lag 1
data.testm4=data.frame(cbind(at.m4,at1.m4));data.testm4
xtest=data.testm4[2:7,];xtest
```



Lampiran 9. *Source Code* Pemodelan Hybrid ARIMA (0,1,1)-SVR
Menggunakan *Software Rstudio* 3.6.2 (Lanjutan)

```
library(e1071)
model.svr.testm4 = svm(et.m4~,data=data.trainm4,
                        type = "eps-regression",
                        kernel = "radial",
                        gamma = 0.7, cost = 300, epsilon=0.0002)
summary(model.svr.testm4)
fitsvr.testm4=predict(model.svr.testm4,newdata=xtest);fitsvr.t
estm4
###MODEL HYBRID###
fit.hybrid.testm4
=arima.test+as.numeric(fitsvr.testm4[1:6]);fit.hybrid.testm4
library(MLmetrics)
mape.hybrid.testm4=
MAPE(fit.hybrid.testm4,test[2:7,2]);mape.hybrid.testm4
```

Lampiran 10. *Source Code* Pemodelan Hybrid ARIMA (1,1,0)-SVR
Menggunakan *Software Rstudio* 3.6.2

```
#####INPUT DATA#####  
library(readxl)  
data.clinker=read.csv("E:/SKRIPSI/data.skripsi/  
clinker_bulan.csv",header=TRUE,sep=",");data.clinker  
clinker=data.clinker$klinker;clinker  
#####MODEL HYBRID ARIMA(1,1,0)-SVR TRAINING#####  
train=cbind(data.clinker[1:47,]);train  
#####ARIMA#####  
#m5  
m5 = Arima(train2, c(1,1,0), include.mean = T);m5  
Yfits =fitted(m5);Yfits  
Yfitsinv1 =InvBoxCox(Yfits, lambda=0.8808438);Yfitsinv1  
Yfitsinv2 =InvBoxCox(Yfitsinv1, lambda=1.999924);Yfitsinv2  
Yfit.arima.trainm5 =as.numeric(Yfitsinv2[2:47]);Yfit.arima.trainm5  
Yfit.arima.m5=as.numeric(Yfitsinv2);Yfit.arima.m5  
library(MLmetrics)  
mape.arima.train = MAPE(Yfitsinv2,train[,2]);mape.arima.train  
#####UJI NONLINIERITAS#####  
library(Hmisc)  
et.m5 =train[,2]-as.numeric(Yfitsinv2);et.m5  
yt1.m5 =Lag(train);yt1.m5 #lag 1 train  
N =data.frame(train[,2],yt1.m5,et.m5);N  
library(lmtest)  
resettest(train~yt1.m5+et.m5, power=2, type="regressor", data=N)  
#####SVR#####  
library(quantmod)  
et.m5 =train[,2]-as.numeric(Yfitsinv2);et.m5  
et1.m5 =Lag(et.m5);et1.m5 #lag 1 et  
data.trainm5 =data.frame(cbind(et.m5,et1.m5));data.trainm5  
xtrain=data.trainm5[2:47,];xtrain  
library(e1071)
```



Lampiran 10. *Source Code* Pemodelan Hybrid ARIMA (1,1,0)-SVR
Menggunakan *Software Rstudio* 3.6.2 (Lanjutan)

```
tuning = tune(svm, et.m5~., data=data.trainm4,  
            type='eps regression', kernel='radial',  
            ranges= list(epsilon=seq(0.0001,0.001,0.0001),  
                        cost=seq(300,600,100), gamma=seq(0.5,0.7,0.1))  
summary(tuning)  
svr.opt.trainm5 = svm(et.m5~.,data=data.trainm5,  
                    type = "eps-regression",kernel = "radial",  
                    gamma = 0.7, cost = 300,  
                    epsilon=0.0002);svr.opt.trainm5  
Ysvr.trainm5 = predict(svr.opt.trainm5,xtrain) ; Ysvr.trainm5  
  
#####HYBRID#####  
fit.hybrid.trainm5=Yfit.arima.trainm5+as.numeric(Ysvr.trainm5);fit.  
hybrid.trainm5  
library(MLmetrics)  
mape.hybrid.trainm5 =  
MAPE(fit.hybrid.trainm5,train[2:47,2]);mape.hybrid.trainm5  
  
###Plot Aktual Vs Fitted Model Hybrid Training###  
library(ggplot2)  
library(tidyr)  
library(dplyr)  
library(lubridate)  
klinker=c(train[2:47,2]);klinker  
date=c(train[,1]);m  
Bulan=ym(date[2:47])  
a=data.frame(Bulan,klinker,Yfit.arima.trainm5, fit.hybrid.trainm5);a  
str(a)  
a %>%  
  select(Bulan, klinker, Yfit.arima.trainm5, fit.hybrid.trainm5) %>%  
  head(n = 10)  
ggplot(data = a, mapping = aes(x = Bulan,group=3)) +  
  geom_line(aes(y = klinker, color = "Aktual"), size = 0.8) +
```



Lampiran 10. *Source Code* Pemodelan Hybrid ARIMA (1,1,0)-SVR Menggunakan *Software Rstudio* 3.6.2 (Lanjutan)

```
geom_line(aes(y = Yfit.arima.trainm5, color = "Arima"), size = 0.8)
+
geom_line(aes(y = fit.hybrid.trainm5, color = "Hybrid"), size = 0.8)
+
ggtitle("Aktual Vs Fitted model")+
xlab('bulan')+
ylab('klinker(ribuan ton)')+
scale_x_date(date_breaks = "8 month", date_labels = "%y/%m")+
scale_color_manual(name = "klinker", # judul legend
                    values = c("Aktual" = "black", "Arima" = "green",
                               "Hybrid" = "red"))

#=====PERAMALAN MODEL HYBRID DATA TESTING=====#
test=cbind(data.clinker[48:54,]);test
###ARIMA####
library(forecast)
test1=BoxCox(test[,2],1.999924);test1
test2=BoxCox(test1,0.8808438);test2
m5t = Arima(test2, c(1,1,0), include.mean = T);m5t
Yfits =c(forecast(m5,h=1)$mean,fitted(m5t)[-1]);Yfits
Yfitsinv1 =InvBoxCox(Yfits, lambda=0.8808438);Yfitsinv1
Yfitsinv2 =InvBoxCox(Yfitsinv1, lambda=1.999924);Yfitsinv2
Yfit.arima.test=as.numeric(Yfitsinv2[2:7]);arima.test
mape.arima.testm5=
MAPE(as.numeric(Yfitsinv2),test[,2]);mape.arima.testm5

###SVR###
library(Hmisc)
at.m5=test[,2]-as.numeric(Yfitsinv2);at.m5
at1.m5=Lag(at.m5);at1.m5 #data residual lag 1
data.testm5=data.frame(cbind(at.m5,at1.m5));data.testm5
xtest=data.testm5[2:7,];xtest
library(e1071)
```



Lampiran 10. *Source Code* Pemodelan Hybrid ARIMA (1,1,0)-SVR
Menggunakan *Software Rstudio* 3.6.2 (Lanjutan)

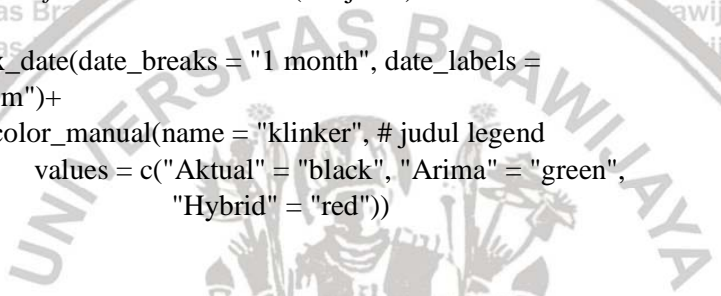
```
model.svr.testm5 = svm(et.m5~.,data=data.trainm5,
type = "eps-regression",
kernel = "radial", gamma = 0.7,
cost = 300, epsilon=0.0002)
summary(model.svr.testm5)
fitsvr.testm5=predict(model.svr.testm5,newdata=xtest);fitsvr.testm5

####MODEL HYBRID###
fit.hybrid.testm5
=arima.test+as.numeric(fitsvr.testm5[1:6]);fit.hybrid.testm5library(
MLmetrics)
mape.hybrid.testm5=
MAPE(fit.hybrid.testm5,test[2:7,2]);mape.hybrid.testm5

####Plot Aktual Vs Fitted Model Hybrid Testing####
klinker=c(test[2:7,2]);klinker
date=c(test[,1]);date
Bulan=ym(date[2:7])
b=data.frame(Bulan, klinker, Yfit.arima.testm5, fit.hybrid.testm5);b
str(b)
b%>%
select(Bulan, klinker, Yfit.arima.testm5, fit.hybrid.testm5)%>%
head(n = 10)
ggplot(data = b, mapping = aes(x = Bulan,group=3)) +
geom_line(aes(y = klinker, color = "Aktual"), size = 0.8) +
geom_line(aes(y = Yfit.arima.testm5,color = "Arima"), size =
0.8) +
geom_line(aes(y = fit.hybrid.testm5, color = "Hybrid"), size =
0.8) +
ggtitle("Aktual Vs Fitted model")+
xlab('bulan')+
ylab('klinker(ribuan ton)'+
```


Lampiran 10. *Source Code* Pemodelan Hybrid ARIMA (1,1,0)-SVR
Menggunakan *Software Rstudio* 3.6.2 (Lanjutan)

```
scale_x_date(date_breaks = "1 month", date_labels =  
"%y/%m")+  
scale_color_manual(name = "klinker", # judul legend  
values = c("Aktual" = "black", "Arima" = "green",  
"Hybrid" = "red"))
```



Lampiran 11. *Source Code* Ramalan n-periode ke depan
Menggunakan Model ARIMA(1,1,0)-SVR

```
#-----PERAMALAN N-PERIODE KE DEPAN-----#
###ARIMA###
library(forecast)
clinker1=BoxCox(clinker,1.999924);clinker1
clinker2=BoxCox(clinker1,0.8808438);clinker2
cl = Arima(clinker2, c(1,1,0), include.mean = T);cl
Yfits =c(forecast(cl,h=1)$mean);Yfits
Yfitsinv1 =InvBoxCox(Yfits, lambda=0.8808438);Yfitsinv1
Yfitsinv2 =InvBoxCox(Yfitsinv1, lambda=1.999924);Yfitsinv2
fore.arima=forecast(Yfitsinv2,1);fore.arima

###SVR###
library(Hmisc)
nt=clinker-as.numeric(Yfitsinv2);nt
nt
acf(nt)
nt1=Lag(nt);nt1 #data residual lag 1
data.cl=data.frame(cbind(nt,nt1));data.cl
library(e1071)
model.svr.cl = svm(nt~,data=data.cl, type = "eps-regression",
kernel = "radial", gamma = 0.7, cost = 300,
epsilon=0.0002)
m=1
for(i in (length(nt)):(length(nt)+m))
xt=c(nt,rep(0,m))
{
Xtest=matrix(xt[i-1],byrow=FALSE)
xt[i]=predict(model.svr.cl,Xtest)
}
fore.SVR=xt[(length(nt)):(length(nt)+m)];fore.SVR

###MODEL HYBRID###
fore.arima.svr=fore.arima+fore.SVR;fore.arima.svr
```