

# PERAMALAN RUNUT WAKTU CURAH HUJAN MENGUNAKAN METODE “SVR – ANT CO”

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Voni Tri Astutik  
NIM: 125150201111058



PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2017

## PENGESAHAN

PERAMALAN RUNUT WAKTU CURAH HUJAN  
MENGUNAKAN METODE "SVR-ANT CO"

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Voni Tri Astutik

NIM: 125150201111058

Skripsi ini telah diuji dan dinyatakan lulus pada  
20 Januari 2017

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Arief Andy Soebroto, S.T., M.Kom

NIP: 1972045 1999903 1 002

Hartanto S.T., M.M

NIP: 1973048 199503 1 001

Mengetahui  
Ketua Jurusan Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 16 Januari 2015



Voni Tri Asrutik

NIM: 125150201111058

## KATA PENGANTAR

Bagian ini memuat pernyataan resmi untuk menyampaikan rasa terima kasih penulis Alhamdulillah. Puji syukur penulis panjatkan kepada Allah SWT atas rahmat serta ridho-Nya sehingga skripsi yang berjudul *Peramalan Runut Waktu Curah Hujan Menggunakan Metode "SVR – ANT CO"* ini dapat terselesaikan. Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Arief Andy Soebroto, S.T., M.Kom selaku dosen pembimbing skripsi yang telah banyak membantu, membimbing, dan mengarahkan sehingga skripsi ini dapat selesai.
2. Bapak Hartanto, S.T., M.M. selaku pihak BMKG Stasiun Klimatologi Karangploso Malang sekaligus dosen pembimbing skripsi yang telah banyak membantu, membimbing, dan mengarahkan sehingga skripsi ini dapat selesai.
3. Bapak Ahmad Luthfi, S.ST. selaku pihak BMKG Stasiun Klimatologi Karangploso Malang yang juga turut ikut membantu dan mengarahkan sehingga dapat menyelesaikan skripsi ini.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku ketua jurusan teknik informatika.
5. Ibu Indriati, S.T.,M.T. selaku dosen penasehat akademik yang selalu memberikan nasehat selama menempuh masa studi.
6. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya memberikan semangat kepada penulis, serta senantiasa tiada hentinya memberikan doa dan semangat demi terselesainya skripsi ini.
7. Seluruh civitas akademika Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 10 Januari 2016

Penulis

Voni29@gmail.com

## ABSTRAK

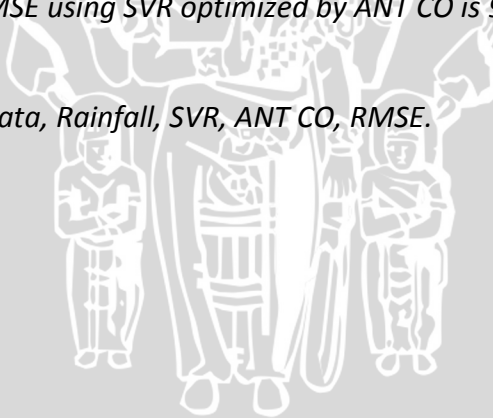
Curah hujan merupakan salah satu faktor penentu perubahan iklim yang terjadi di suatu wilayah. Curah hujan memiliki pola yang tidak dapat dipastikan, sedangkan curah hujan merupakan salah satu unsur iklim yang berpengaruh di berbagai sektor kehidupan seperti manajemen sumber daya air. Untuk dapat mendukung penentuan cuaca dan iklim disuatu wilayah pada waktu tertentu maka diperlukan peramalan curah hujan untuk menentukan curah hujan yang akan terjadi diwaktu mendatang. Banyak metode peramalan yang telah digunakan untuk meramalkan data runut waktu seperti *Support Vector Regression* (SVR) yang terbukti pada beberapa kasus dapat memberikan hasil peramalan yang baik. Metode *Support Vector Regression* (SVR) merupakan *machine learning* pengembangan dari metode *Support Vector Machine* (SVM) yang dapat melakukan peramalan bersifat non linier. Untuk meningkatkan akurasi regresi metode SVR diperlukan pemilihan parameter SVR yang tepat. Parameter tersebut adalah  $c$ LR, kompleksitas (C), epsilon ( $\epsilon$ ), lambda ( $\lambda$ ), degree (d), gamma kernel ( $\gamma_k$ ). Metode optimasi seperti *Ant Colony Optimization* (ANT CO) dapat melakukan optimasi pemilihan parameter metode. Dengan digunakannya ANT CO maka menghasilkan *error rate* yang lebih rendah dan meningkatnya nilai akurasi metode SVR. Hasil dari pengujian yang dilakukan menggunakan data dari Badan Meteorologi Klimatologi dan Geofisika (BMKG) yang berupa data curah hujan pada tahun 2001 hingga 2015 wilayah Poncokusumo. Berdasarkan evaluasi kinerja peramalan runut waktu curah hujan diwilayah Poncokusumo menggunakan metode SVR-ANT CO mengacu pada nilai *Root Mean Square Error* (RMSE). Metode SVR yang dioptimasi menggunakan ANT CO dapat menghasilkan nilai RMSE sebesar 9.482.

**Kata kunci:** Data Runut Waktu, Curah Hujan,SVR, ANT CO, RMSE.

## ABSTRACT

Rainfall is one of the determining factors of climate change in a region. Rainfall has a pattern that can not be ascertained, in other hand rainfall is one of the climate factor that affect various sectors in social life such as the management of water resource. To be able make determination of the weather and climate in certain region and time, forecasting of the rainfall is needed. Many forecasting methods that have been used to predict the time series data such as Support Vector Regression (SVR) which is proven in some cases can provide better forecasting results. Support Vector Regression (SVR) is the development methods of machine learning Support Vector Machine (SVM) which can perform non linear forecasting. To improve the accuracy of regression methods required good parameter selection. The parameter that need the optimization are cLR, Complexity (C), epsilon ( $\epsilon$ ), lambda ( $\lambda$ ), degree (d), gamma kernel ( $\gamma_k$ ). Optimization methods such as Ant Colony Optimization (ANT CO) can optimize the choice of method parameters. The ANT CO can make significantly lower error rate and increased accuracy values SVR methods. Results of testing conducted using data from the Meteorology, Climatology and Geophysics (BMKG) in the form of rainfall data from 2001 to 2015 in Poncokusumo region. Based on the performance evaluation of time-series forecasting of rainfall in the region Poncokusumo using SVR-ANT CO refers to the value Root Mean Square Error (RMSE). The value of RMSE using SVR optimized by ANT CO is 9.482.

**Keyword:** Time Series Data, Rainfall, SVR, ANT CO, RMSE.



## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xiv
DAFTAR PERSAMAAN.....	xix
DAFTAR LAMPIRAN .....	xxi
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	4
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>6</b>
2.1 Kajian pustaka.....	6
2.2 Curah hujan.....	84
2.2.1 Pola Curah Hujan Indonesia.....	84
2.2.2 <i>Standardized Precipitation Index (SPI)</i> .....	86
2.2.3 Fenomena yang mempengaruhi Iklim/Musim di Indonesia .....	86
2.3 Data Runut Waktu .....	87
2.3.1 Pola Data Runut Waktu.....	88
2.3.2 <i>Preprocessing</i> Data Runut Waktu .....	90
2.4 Peramalan .....	91
2.5 <i>Support Vector Regression (SVR)</i> .....	93
2.5.1 Fungsi Kernel.....	98
2.5.2 Metode Sequensial SVR .....	99

2.6 <i>Ant Colony Optimization</i> (ANT CO) .....	101
2.6.1 Karakteristik .....	101
2.6.2 Algoritma ANT CO .....	102
2.7 Metode SVR-ANT CO.....	<b>Error! Bookmark not defined.</b>
2.8 Evaluasi kinerja .....	102
<b>BAB 3 METODOLOGI .....</b>	<b>105</b>
3.1 Studi literatur.....	106
3.2 Pengumpulan data.....	106
3.3 Analisa kebutuhan .....	106
3.4 Perancangan Sistem.....	107
3.4.1 Model perancangan sistem.....	107
3.4.2 Arsitektur Sistem Peramalan .....	108
3.5 Implementasi .....	109
3.6 Analis dan pengujian sistem .....	109
3.7 Kesimpulan.....	110
<b>BAB 4 PERANCANGAN.....</b>	<b>111</b>
4.1 Peramalan .....	112
4.1.1 Definisi Masalah .....	112
4.1.2 Pengumpulan Data.....	<b>Error! Bookmark not defined.</b>
4.1.3 Analisis Data .....	<b>Error! Bookmark not defined.</b>
4.1.4 Seleksi dan Validasi Model.....	<b>Error! Bookmark not defined.</b>
4.2 Perancangan Antarmuka .....	112
4.3 Perancangan Pengujian .....	113
4.3.1 Perancangan Skenario Pengujian Iterasi SVR .....	114
4.3.2 Perancangan Skenario Pengujian Jumlah Iterasi ANT CO .....	114
4.3.3 Perancangan Skenario Pengujian Jumlah Semut.....	115
4.3.4 Perancangan Skenario Pengujian Batas Nilai Parameter.....	115
4.3.5 Perancangan Skenario Pengujian Variasi Variabel Lag .....	118
4.3.6 Perancangan Skenario Pengujian Variasi Data Latih.....	116
<b>BAB 5 Implementasi .....</b>	<b>119</b>
5.1 Spesifikasi Sistem .....	119
5.1.1 Spesifikasi Perangkat Keras.....	119



5.1.2 Spesifikasi Perangkat Lunak .....	120
5.1.3 Batasan-Batasan Sistem .....	120
5.2 Implementasi Algoritma .....	120
5.2.1 Implementasi Algoritma SVR .....	120
5.2.2 Implementasi Algoritma ANT CO .....	130
5.3 Implementasi Antarmuka .....	138
<b>BAB 6 Pengujian dan analisis .....</b>	<b>141</b>
6.1 Hasil dan Analisis Uji Coba Jumlah Iterasi Pelatihan SVR .....	141
6.2 Hasil dan Analisis Uji Coba Pengujian SVR .....	142
6.3 Hasil dan Analisis Uji Coba Jumlah Iterasi ANT CO .....	143
6.4 Hasil dan Analisis Uji Coba Jumlah Semut pada ANT CO .....	144
6.5 Hasil dan Analisis Uji Coba Batas Nilai Parameter SVR.....	145
6.5.1 Hasil dan Analisis Uji Coba Batas Nilai Parameter $cLR$ .....	145
6.5.2 Hasil dan Analisis Uji Coba Batas Nilai Parameter $C$ .....	147
6.5.3 Hasil dan Analisis Uji Coba Batas Nilai Parameter $\epsilon$ .....	148
6.5.4 Hasil dan Analisis Uji Coba Batas Nilai Parameter $\lambda$ .....	149
6.5.5 Hasil dan Analisis Uji Coba Batas Nilai Parameter $d$ .....	150
6.5.6 Hasil dan Analisis Uji Coba Batas Nilai Parameter $\gamma_k$ .....	151
6.6 Hasil dan analisis uji coba pengujian SVR-ANT CO .....	152
6.7 Perbandingan Hasil dan Analisis Uji Coba Pengujian SVR dan Pengujian SVR-ANT CO .....	153
6.8 Hasil dan Analisis Uji Coba Pengujian Variasi Periode Waktu ( <i>Lag Time</i> ) SVR-ANT CO .....	153
6.9 Hasil dan Analisis Uji Coba Variasi Jumlah Data Latih SVR-ANT CO ..	154
6.10 Analisis Global.....	155
<b>BAB 7 PENUTUP .....</b>	<b>157</b>
7.1 Kesimpulan.....	157
7.2 Saran .....	158
<b>DAFTAR PUSTAKA.....</b>	<b>159</b>
<b>LAMPIRAN A Data Curah Hujan Dasarian Berdasarkan Runut Waktu.....</b>	<b>168</b>



## DAFTAR TABEL

Tabel 2.1 Kajian Pustaka .....	8
Tabel 2.2 Signifikasnsi MAPE.....	103
Tabel 4.1 Data curah hujan dasarian tahun 2000-2015 wilayah Poncokusumo Malang.....	<b>Error! Bookmark not defined.</b>
Tabel 4.2 Data curah hujan dasarian tahun 2000-2015 wilayah Tangkilsari Malang .....	<b>Error! Bookmark not defined.</b>
Tabel 4.3 Rekap data runut waktu Wilayah Poncokusumo.....	<b>Error! Bookmark not defined.</b>
Tabel 4.4 Data dasarian Ke-1 Bulan Januari Tahun 2011 sampai dengan Dasarian Ke-1 Bulan Mei Tahun 2011 .....	<b>Error! Bookmark not defined.</b>
Tabel 4.5 Pola data sekuensial (horizontal) curah hujan Wilayah Poncokusumo Malang.....	<b>Error! Bookmark not defined.</b>
Tabel 4.6 Data Bulan Januari Dasarian Ke-2 Tahun 2000-2012.....	<b>Error! Bookmark not defined.</b>
Tabel 4.7 Pola data paralel (vertikal) curah hujan Wilayah Poncokusumo Malang .....	<b>Error! Bookmark not defined.</b>
Tabel 4.8 Rekap data dasarian Wilayah Tangkilsari.....	<b>Error! Bookmark not defined.</b>
Tabel 4.9 Data Dasarian ke-2 Bulan November Tahun 2013 sampai Dasarian ke-2 Bulan Maret Tahun 2014 .....	<b>Error! Bookmark not defined.</b>
Tabel 4.10 Pola data sekuensial (horizontal) curah hujan Wilayah Tangkilsari Malang.....	<b>Error! Bookmark not defined.</b>
Tabel 4.11 Data Bulan Februari Dasarian ke-1 pada Tahun 2000-2012 .....	<b>Error! Bookmark not defined.</b>
Tabel 4.12 Data Bulan Februari dasarian ke-1 pada Tahun 2000-2012 Tangkilsari .....	<b>Error! Bookmark not defined.</b>
Tabel 4.13 Tahapan proses SVR-ANT CO .....	<b>Error! Bookmark not defined.</b>
Tabel 4.14 Nilai parameter ANT CO .....	<b>Error! Bookmark not defined.</b>
Tabel 4.15 Batas parameter SVR dan kernel anova.....	<b>Error! Bookmark not defined.</b>
Tabel 4.16 Inisialisasi batas parameter SVR dan kernel anova.....	<b>Error! Bookmark not defined.</b>
Tabel 4.17 Nilai <i>pheromone</i> awal.....	<b>Error! Bookmark not defined.</b>
Tabel 4.18 Nilai Probabilitas iterasi ke-1.....	<b>Error! Bookmark not defined.</b>
Tabel 4.19 Nilai <i>pheromone update</i> lokal ke-1 .....	<b>Error! Bookmark not defined.</b>



Tabel 4.20 Nilai Probabilitas iterasi ke-2.....**Error! Bookmark not defined.**

Tabel 4.21 Seleksi Roulette Wheel.....**Error! Bookmark not defined.**

Tabel 4.22 Nilai *pheromone update* lokal ke-2 .....**Error! Bookmark not defined.**

Tabel 4.23 Node semut .....**Error! Bookmark not defined.**

Tabel 4.24 Tabel data curah hujan Pocokusumo .....**Error! Bookmark not defined.**

Tabel 4.25 Data latih dan data uji .....**Error! Bookmark not defined.**

Tabel 4.26 Range normalisasi .....**Error! Bookmark not defined.**

Tabel 4.27 Hasil Normalisasi Data Latih .....**Error! Bookmark not defined.**

Tabel 4.28 Inisialisasi parameter SVR dan kernel anova**Error! Bookmark not defined.**

Tabel 4.29 Inisialisasi nilai  $\alpha$  dan  $\alpha^*$  .....**Error! Bookmark not defined.**

Tabel 4.30 Hasil perhitungan kernel anova .....**Error! Bookmark not defined.**

Tabel 4.31 Matrik Hessian (Model Regresi) .....**Error! Bookmark not defined.**

Tabel 4.32 Nilai  $E_i$  iterasi ke-1 .....**Error! Bookmark not defined.**

Tabel 4.33 Nilai  $\delta\alpha^*$  dan  $\delta\alpha$  Iterasi ke-1 .....**Error! Bookmark not defined.**

Tabel 4.34 nilai  $\alpha^*$  dan  $\alpha$  Iterasi ke-1.....**Error! Bookmark not defined.**

Tabel 4.35 Nilai  $E_i$  iterasi ke-2 .....**Error! Bookmark not defined.**

Tabel 4.36 Nilai  $\delta\alpha_i^*$  dan  $\delta\alpha_i$  Iterasi ke-2 .....**Error! Bookmark not defined.**

Tabel 4.37 nilai  $\alpha^*$  dan  $\alpha$  Iterasi ke-2.....**Error! Bookmark not defined.**

Tabel 4.38 Nilai  $E_i$  iterasi ke-300 .....**Error! Bookmark not defined.**

Tabel 4.39 Nilai  $\delta\alpha_i^*$  dan  $\delta\alpha_i$  Iterasi ke-300 .....**Error! Bookmark not defined.**

Tabel 4.40 nilai  $\alpha^*$  dan  $\alpha$  Iterasi ke-300.....**Error! Bookmark not defined.**

Tabel 4.41 Hasil peramalan data latih.....**Error! Bookmark not defined.**

Tabel 4.42 Denormalisasi Data Latih.....**Error! Bookmark not defined.**

Tabel 4.43 Memperbarui *pheromone global*.....**Error! Bookmark not defined.**

Tabel 4.44 Data uji .....**Error! Bookmark not defined.**

Tabel 4.45 Hasil perhitungan kernel *anova* pada Data Uji**Error! Bookmark not defined.**

Tabel 4.46 *Matriks Hessian* (Model Regresi) pada Data Uji**Error! Bookmark not defined.**

Tabel 4.47 Hasil Peramalan data Uji .....**Error! Bookmark not defined.**

Tabel 4.48 Denormalisasi Data uji.....**Error! Bookmark not defined.**

Tabel 4.49 Rancangan Uji Coba iterasi SVR..... 114



Tabel 4.50 Pengujian SVR.....	114
Tabel 4.51 Rancangan Uji Coba iterasi ANT CO .....	114
Tabel 4.52 Rancangan Uji Coba Jumlah Semut.....	115
Tabel 4.53 Rancangan Pengujian Batas Parameter $cLR$ .....	115
Tabel 4.54 Rancangan Pengujian Batas Parameter $C$ .....	115
Tabel 4.55 Rancangan Pengujian Batas Parameter $\epsilon$ .....	116
Tabel 4.56 Rancangan Pengujian Batas Parameter $\lambda$ .....	116
Tabel 4.57 Rancangan Pengujian Batas Parameter $d$ .....	116
Tabel 4.58 Rancangan Pengujian Batas Parameter $\gamma_k$ .....	116
Tabel 4.59 Pengujian SVR-ANT CO .....	117
Tabel 4.60 Perbandingan Pengujian SVR dan SVR-ANT CO .....	117
Tabel 4.61 Rancangan Pengujian Variasi Variabel Lag.....	118
Tabel 4.62 Rancangan Pengujian Variasi Data Latih .....	118
Tabel 5.1 Spesifikasi Perangkat Keras .....	119
Tabel 5.2 Spesifikasi Perangkat Lunak .....	120
Tabel 6.1 Hasil uji coba jumlah iterasi SVR .....	142
Tabel 6.2 Hasil Uji Coba Pengujian SVR.....	143
Tabel 6.3 Hasil uji coba jumlah iterasi ANT CO .....	144
Tabel 6.4 Hasil uji coba jumlah semut .....	145
Tabel 6.5 Hasil uji coba batas nilai parameter $cLR$ .....	146
Tabel 6.6 Hasil uji coba batas nilai parameter $C$ .....	147
Tabel 6.7 Hasil uji coba batas nilai parameter $\epsilon$ .....	148
Tabel 6.8 Hasil uji coba batas nilai parameter $\lambda$ .....	149
Tabel 6.9 Hasil uji coba batas nilai parameter $d$ .....	150
Tabel 6.10 Hasil uji coba batas nilai parameter $\gamma_k$ .....	151
Tabel 6.11 Pengujian SVR-ANT CO .....	152
Tabel 6.12 Hasil perbandingan uji coba pengujian SVR dan SVR-ANT CO.....	153
Tabel 6.13 Hasil uji coba pengujian variasi variabel lag.....	154
Tabel 6.14 Hasil uji coba variasi jumlah data latih.....	155
Tabel 7.1 Data curah hujan dasarian tahun 2000-2014 wilayah Karangploso Malang.....	168



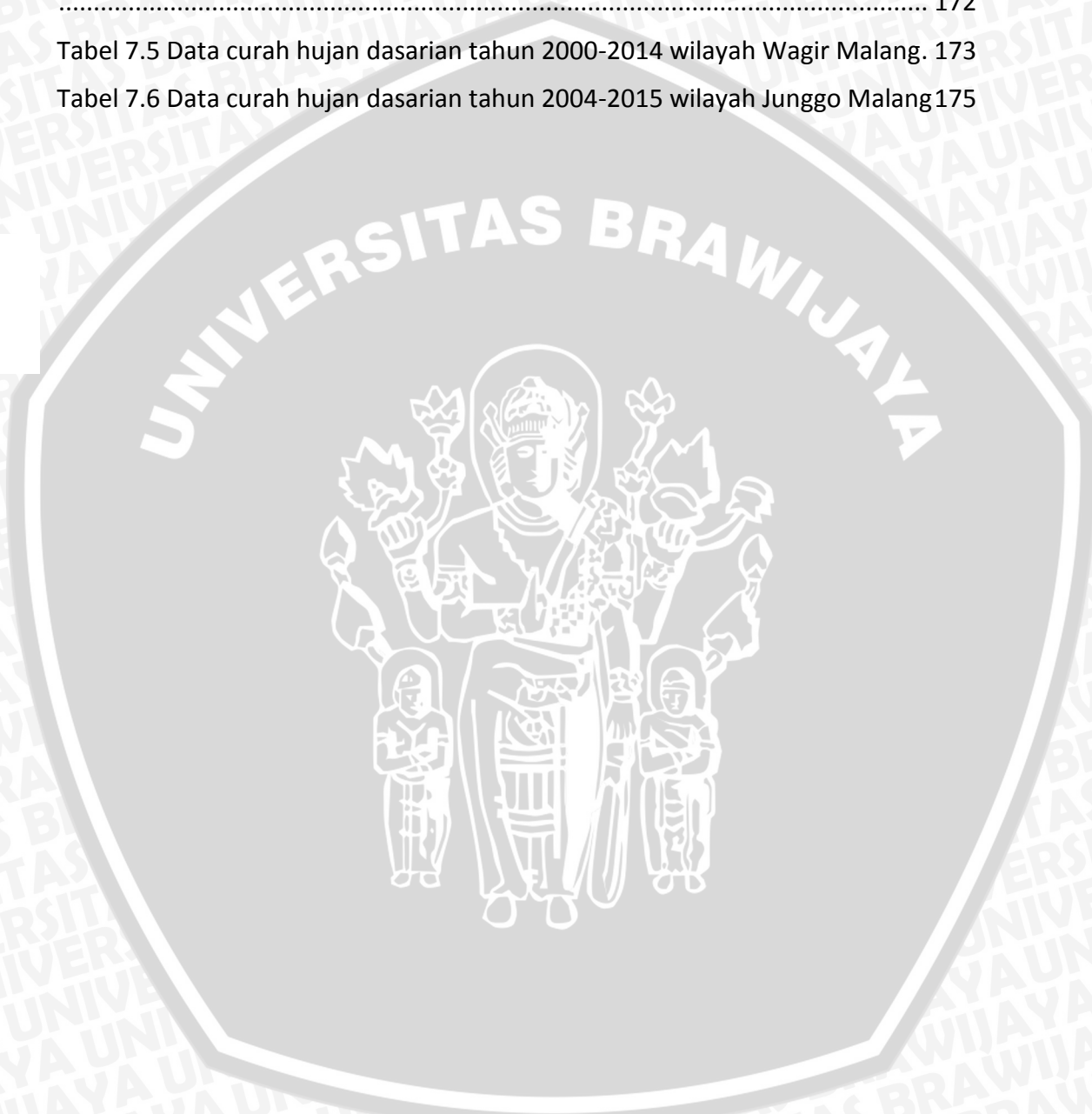
Tabel 7.2 Data curah hujan dasarian tahun 2000-2014 wilayah Poncokusumo Malang..... 169

Tabel 7.3 Data curah hujan dasarian tahun 2000-2014 wilayah Pujon Malang. 170

Tabel 7.4 Data curah hujan dasarian tahun 2000-2014 wilayah Tangkilsari Malang ..... 172

Tabel 7.5 Data curah hujan dasarian tahun 2000-2014 wilayah Wagir Malang. 173

Tabel 7.6 Data curah hujan dasarian tahun 2004-2015 wilayah Junggo Malang 175



## DAFTAR GAMBAR

Gambar 2.1 Pola curah hujan di Indonesia .....	85
Gambar 2.2 Pola stasioner .....	88
Gambar 2.3 Pola trend .....	88
Gambar 2.4 Pola musiman .....	89
Gambar 2.5 Pola siklis .....	89
Gambar 2.6 Proses peramalan .....	91
Gambar 2.7 Hyperplane .....	94
Gambar 2.8 $\epsilon$ -insentive loss function pada SVR .....	95
Gambar 2.9 Penggunaan kernel .....	98
Gambar 2.10 Metode ANT CO .....	101
Gambar 3.1 Diagram blok metodologi penelitian .....	105
Gambar 3.2 Pengumpulan data curah hujan .....	106
Gambar 3.3 Model perancangan sistem .....	107
Gambar 3.4 Arsitektur peramalan runut waktu curah hujan menggunakan metode SVR-ACO .....	108
Gambar 3.5 Implementasi sistem .....	109
Gambar 4.1 Diagram blok perancangan .....	111
Gambar 4.2 Grafik Rekap Curah Hujan Wilayah Poncokusumo	<b>Error! Bookmark not defined.</b>
Gambar 4.3 Grafik data 0 curah hujan dasarian Tahun 2000-2015 Wilayah Poncokusumo Malang.....	<b>Error! Bookmark not defined.</b>
Gambar 4.4 Grafik pola data sekuensial (horizontal) Wilayah Poncokusumo Malang.....	<b>Error! Bookmark not defined.</b>
Gambar 4.5 Grafik bentuk pola data paralel (vertikal) Wilayah Poncokusumo Malang.....	<b>Error! Bookmark not defined.</b>
Gambar 4.6 Grafik Rekap Curah Hujan Wilayah Tangkilsari	<b>Error! Bookmark not defined.</b>
Gambar 4.7 Grafik analisis data 0 curah hujan dasarian Tahun 2000-2015 Wilayah Tangkilsari Malang .....	<b>Error! Bookmark not defined.</b>
Gambar 4.8 Grafik pola data sekuensial (horizontal) Wilayah Tangkilsari Malang .....	<b>Error! Bookmark not defined.</b>

Gambar 4.9 Grafik bentuk pola data Paralel wilayah Tangkilsari Malang .....**Error! Bookmark not defined.**

Gambar 4.10 Diagram alir tahapan proses SVR-ANT CO**Error! Bookmark not defined.**

Gambar 4.11 Diagram alir tahapan pencarian node semut**Error! Bookmark not defined.**

Gambar 4.12 Diagram alir algoritma pencarian node semut**Error! Bookmark not defined.**

Gambar 4.13 *Pseudocode* algoritma pencarian node semut**Error! Bookmark not defined.**

Gambar 4.14 Diagram alir tahapan memperbaiki *pheromone* lokal .....**Error! Bookmark not defined.**

Gambar 4.15 Diagram alir algoritma memperbaiki *pheromone* lokal .....**Error! Bookmark not defined.**

Gambar 4.16 *Pseudocode* algoritma memperbaiki *pheromone* lokal.....**Error! Bookmark not defined.**

Gambar 4.17 Seleksi Roulette Wheel .....**Error! Bookmark not defined.**

Gambar 4.18 Penggambaran jalur semut .....**Error! Bookmark not defined.**

Gambar 4.19 Diagram alir tahapan normalisasi .....**Error! Bookmark not defined.**

Gambar 4.20 Diagram alir algoritma normalisasi .....**Error! Bookmark not defined.**

Gambar 4.21 *Pseudocode* algoritma normalisasi .....**Error! Bookmark not defined.**

Gambar 4.22 Diagram alir tahapan pelatihan SVR ...**Error! Bookmark not defined.**

Gambar 4.23 Diagram alir tahapan Matrik Hessian data latih**Error! Bookmark not defined.**

Gambar 4.24 Diagram alir algoritma Matrik Hessian data latih**Error! Bookmark not defined.**

Gambar 4.25 *Pseudocode* algoritma Matrik Hessian**Error! Bookmark not defined.**

Gambar 4.26 Diagram alir tahapan menghitung nilai *lagrange multiplier* ( $\alpha$ ,  $\alpha^*$ ) .....**Error! Bookmark not defined.**

Gambar 4.27 Diagram alir algoritma menghitung nilai *lagrange multiplier* ( $\alpha$ ,  $\alpha^*$ ) .....**Error! Bookmark not defined.**

Gambar 4.28 *Pseudocode* menghitung nilai *lagrange multiplier* ( $\alpha$ ,  $\alpha^*$ ).....**Error! Bookmark not defined.**

Gambar 4.29 Diagram alir tahapan menghitung regresi data latih .....**Error! Bookmark not defined.**



Gambar 4.30 Diagram alir algoritma menghitung regresi data latih.....**Error! Bookmark not defined.**

Gambar 4.31 *Pseudocode* algoritma menghitung regresi data latih.....**Error! Bookmark not defined.**

Gambar 4.32 Diagram alir tahapan evaluasi kinerja peramalan**Error! Bookmark not defined.**

Gambar 4.33 Diagram alir algoritma evaluasi kinerja peramalan**Error! Bookmark not defined.**

Gambar 4.34 *Pseudocode* algoritma evaluasi kinerja peramalan**Error! Bookmark not defined.**

Gambar 4.35 Diagram Alir tahapan memperbaiki *pheromone* global.....**Error! Bookmark not defined.**

Gambar 4.36 Diagram Alir algoritma memperbaiki *pheromone* global.....**Error! Bookmark not defined.**

Gambar 4.37 *Pseudocode* algoritma memperbaiki *pheromone* global .....**Error! Bookmark not defined.**

Gambar 4.38 Diagram alir tahapan pengujian SVR...**Error! Bookmark not defined.**

Gambar 4.39 Diagram alir tahapan Matrik Hessian data uji**Error! Bookmark not defined.**

Gambar 4.40 Diagram alir algoritma Matrik Hessian data uji**Error! Bookmark not defined.**

Gambar 4.41 *Pseudocode* algoritma Matrik Hessian**Error! Bookmark not defined.**

Gambar 4.42 Diagram Alir tahapan regresi data uji .**Error! Bookmark not defined.**

Gambar 4.43 Diagram alir algoritma regresi data uji**Error! Bookmark not defined.**

Gambar 4.44 *Pseudocode* algoritma regresi data uji **Error! Bookmark not defined.**

Gambar 4.45 Diagram alir proses evaluasi kinerja data uji**Error! Bookmark not defined.**

Gambar 4.46 Diagram alir algoritma evaluasi kinerja data uji**Error! Bookmark not defined.**

Gambar 4.47 *Pseudocode* algoritma evaluasi kinerja data uji**Error! Bookmark not defined.**

Gambar 4.48 Perancangan antarmuka ..... 113

Gambar 5.1 Implementasi..... 119

Gambar 5.2 Implementasi algoritma proses pengambilan data ..... 121

Gambar 5.3 Implementasi algoritma proses pencarian nilai terkecil..... 121





Gambar 5.4 Implementasi algoritma proses pencarian nilai terbesar.....	122
Gambar 5.5 Implementasi algoritma proses normalisasi data .....	122
Gambar 5.6 Implementasi algoritma proses pencarian nilai kernel.....	123
Gambar 5.7 Implementasi algoritma proses pencarian nilai Matriks Hessian ...	123
Gambar 5.8 Implementasi algoritma proses <i>sequential learning</i> .....	124
Gambar 5.9 Implementasi algoritma proses pencarian nilai $E$ .....	124
Gambar 5.10 Implementasi algoritma proses pencarian nilai $\vartheta\alpha^*$ .....	125
Gambar 5.11 Implementasi algoritma proses pencarian nilai $\vartheta\alpha$ .....	125
Gambar 5.12 Implementasi algoritma proses pencarian nilai $\alpha^*$ .....	125
Gambar 5.13 Implementasi algoritma proses pencarian nilai $\alpha$ .....	126
Gambar 5.14 Implementasi algoritma proses perhitungan nilai $f(x)$ .....	126
Gambar 5.15 Implementasi algoritma proses perhitungan nilai denormalisasi	126
Gambar 5.16 Implementasi algoritma proses perhitungan nilai rmse data latih	127
Gambar 5.17 Implementasi algoritma proses perhitungan nilai rmse data uji..	128
Gambar 5.18 Implementasi algoritma proses perhitungan matriks hessian testing .....	128
Gambar 5.19 Implementasi algoritma proses perhitungan nilai kernel.....	129
Gambar 5.20 Implementasi algoritma proses perhitungan nilai denormalisasi	129
Gambar 5.21 Implementasi algoritma proses perhitungan nilai RMSE data uji.	129
Gambar 5.22 Implementasi algoritma perhitungan <i>pheromone alpha</i> dan <i>beta</i> .....	130
Gambar 5.23 Implementasi algoritma perhitungan total <i>pheromone</i> .....	131
Gambar 5.24 Implementasi algoritma perhitungan <i>argument max</i> .....	131
Gambar 5.25 Implementasi algoritma memperbaiki <i>pheromone</i> lokal.....	132
Gambar 5.26 Implementasi algoritma perhitungan probabilitas .....	133
Gambar 5.27 Implementasi algoritma perhitungan probabilitas kumulatif.....	133
Gambar 5.28 Implementasi algoritma perhitungan index seleksi.....	134
Gambar 5.29 Implementasi algoritma perhitungan memperbaiki <i>pheromone global</i> .....	135
Gambar 5.30 Implementasi algoritma perhitungan konversi nilai cLR .....	135
Gambar 5.31 Implementasi algoritma perhitungan konversi nilai kompleksitas	136
Gambar 5.32 Implementasi algoritma perhitungan konversi nilai <i>epsilon</i> .....	136



Gambar 5.33 Implementasi algoritma perhitungan konversi nilai $\lambda$ .....	137
Gambar 5.34 Implementasi algoritma perhitungan konversi nilai $degree$ .....	137
Gambar 5.35 Implementasi algoritma perhitungan konversi nilai $gamma\ kernel$ .....	138
Gambar 5.36 Implementasi antarmuka panel pertama .....	138
Gambar 5.37 Implementasi antarmuka panel kedua .....	139
Gambar 5.38 Implementasi antarmuka panel ketiga .....	139
Gambar 5.39 Implementasi antarmuka panel keempat.....	140
Gambar 6.1 Pengujian dan analisis .....	141
Gambar 6.2 Grafik hasil uji coba jumlah iterasi SVR.....	142
Gambar 6.3 Grafik hasil uji coba jumlah iterasi ANT CO.....	144
Gambar 6.4 Grafik hasil uji coba jumlah semut.....	145
Gambar 6.5 Grafik hasil uji coba batas parameter $cLR$ .....	146
Gambar 6.6 Grafik hasil uji coba batas parameter $C$ .....	147
Gambar 6.7 Grafik hasil uji coba batas parameter $\epsilon$ .....	148
Gambar 6.8 Grafik hasil uji coba batas parameter $\lambda$ .....	150
Gambar 6.9 Grafik pengujian batas parameter $d$ .....	151
Gambar 6.10 Grafik pengujian batas parameter $\gamma_k$ .....	152
Gambar 6.11 Grafik pengujian variasi variabel lag .....	154
Gambar 6.12 Grafik pengujian variasi data latih .....	155

## DAFTAR PERSAMAAN

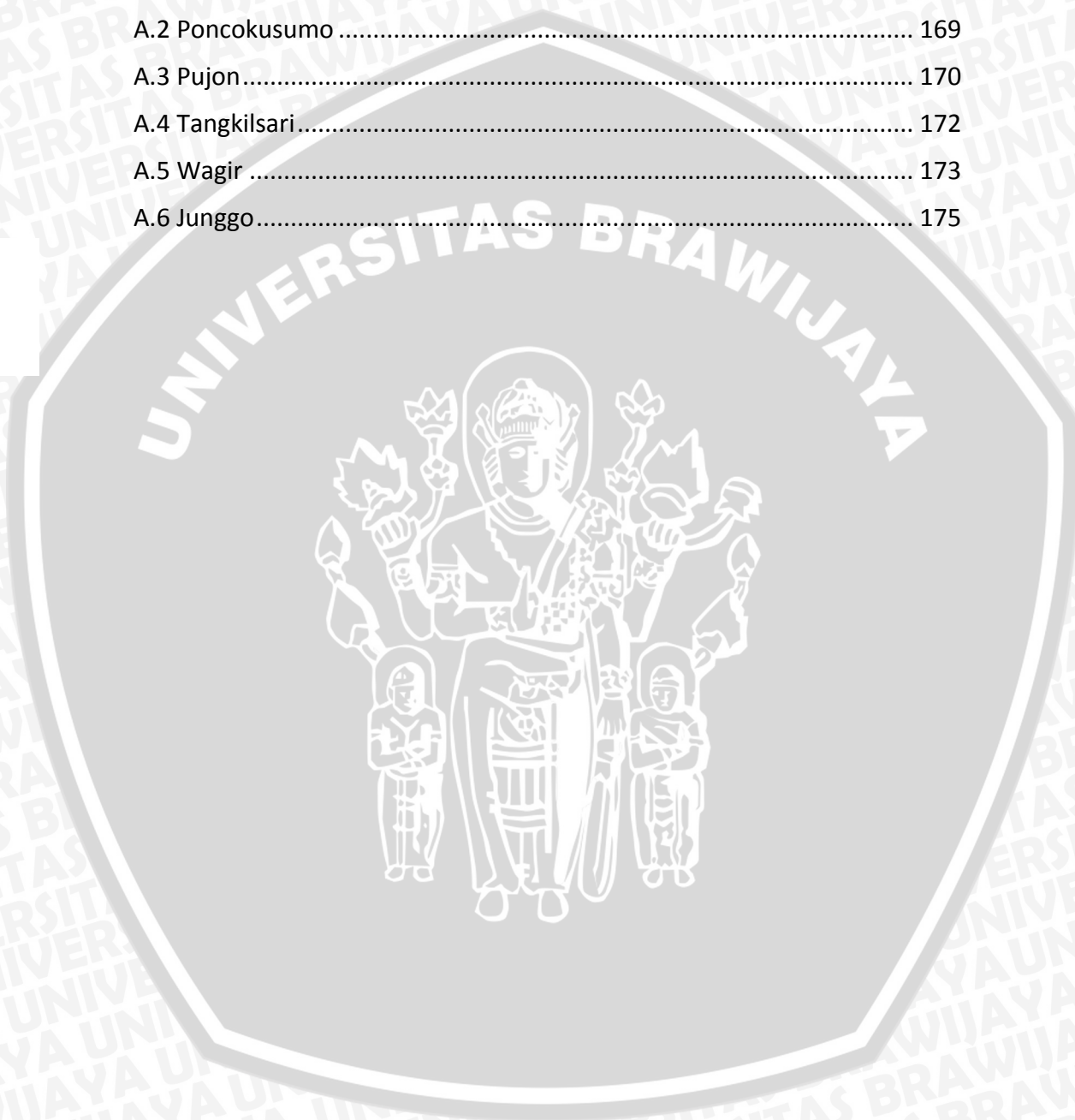
Persamaan (2.1) Normalisasi <i>Min-Max</i> .....	91
Persamaan (2.2) Denomalisasi <i>Min-Max</i> .....	91
Persamaan (2.3) Fungsi dasar regresi linier .....	94
Persamaan (2.4) Estimasi koefisien .....	95
Persamaan (2.5) Transformasi estimasi koefisien .....	95
Persamaan (2.6) Konsep <i>lagrangian multipliers</i> .....	96
Persamaan (2.7) Fungsi regresi non-linier .....	96
Persamaan (2.8) Fungsi regresi SVR.....	97
Persamaan (2.9) Fungsi regresi SVR dengan kernel.....	97
Persamaan (2.10) Fungsi regresi SVR usulan Vijayakumar .....	97
Persamaan (2.11) Kernel Linier .....	98
Persamaan (2.12) Kernel <i>Gaussian Radial Basis Function</i> (RBF).....	98
Persamaan (2.13) Kernel <i>analysis of variance</i> (ANOVA) RBF.....	99
Persamaan (2.14) Matrik Hessian .....	99
Persamaan (2.15) Perhitungan nilai kesalahan error proses pelatihan .....	100
Persamaan (2.16) Perhitungan nilai $\delta\alpha^*$ .....	100
Persamaan (2.17) Perhitungan nilai $\delta\alpha$ .....	100
Persamaan (2.18) Perhitungan nilai $\alpha$ .....	100
Persamaan (2.19) Perhitungan nilai $\alpha^*$ .....	100
Persamaan (2.20) Fungsi regresi SVR.....	100
Persamaan (2.21) Probabilitas perpindahan semut ( <i>argument max</i> ) ANT CO <b>Error!</b> <b>Bookmark not defined.</b>	
Persamaan (2.22) Probabilitas perpindahan semut ( <i>roulette wheel</i> ) ANT CO <b>Error!</b> <b>Bookmark not defined.</b>	
Persamaan (2.23) Memperbarui <i>pheromone</i> lokal ANT-CO <b>Error!</b> <b>Bookmark not defined.</b>	
Persamaan (2.24) Memperbarui <i>pheromone</i> global ANT-CO <b>Error!</b> <b>Bookmark not defined.</b>	
Persamaan (2.25) Perhitungan $\Delta\tau$ ANT-CO .....	<b>Error! Bookmark not defined.</b>
Persamaan (2.26) <i>Mean square error</i> (MSE) .....	103
Persamaan (2.27) <i>Mean Absolute Percentage Error</i> (MAPE) .....	103

Persamaan (2.28) *Mean Absolute Error* (MAE)..... 103  
Persamaan (2.29) *Root Mean square error* (RMSE)..... 104



## DAFTAR LAMPIRAN

LAMPIRAN A Data Curah Hujan Dasarian Berdasarkan Runut Waktu.....	168
A.1 Karangploso .....	168
A.2 Poncokusumo .....	169
A.3 Pujon.....	170
A.4 Tangkilsari.....	172
A.5 Wagir .....	173
A.6 Junggo.....	175



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Curah hujan merupakan salah satu kunci dari variabel hidrologi yang menghubungkan antara atmosfer dan proses yang terjadi pada permukaan tanah. Hasil dari aktivitas kompleks yang terjadi karena adanya fenomena pada atmosfer secara global menentukan curah hujan. Secara klimatologi wilayah Indonesia terdapat 407 pola iklim yang mempengaruhi curah hujan di Indonesia. Menurut Aldrian & Djamil (2008), Indonesia merupakan negara maritim yang memiliki curah hujan yang tinggi dengan rata-rata curah hujan 2700mm per tahun. Pulau Jawa sendiri pada tahun 1991-2005 memiliki rata-rata curah hujan yang mencapai 2300mm. Curah hujan memiliki pola yang tidak dapat dipastikan, sedangkan curah hujan merupakan salah satu unsur iklim yang berpengaruh di berbagai sektor kehidupan seperti manajemen sumber daya air. Untuk dapat menentukan pengelolaan sumber daya air yang tepat maka dibutuhkan suatu teknologi peramalan runut waktu curah hujan yang akurat dan terperinci (Chang, et al., 2013) (BMKG, 2015) (Pawitan & Haryani, 2011).

Peramalan runut waktu dapat dirancang dengan menggunakan dua pendekatan yaitu pendekatan statistik dan pendekatan *artificial intelligence* (AI). Peramalan dengan pendekatan statistik menggunakan metode *single variable* yaitu meramalkan dengan data variabel tunggal. Pada peramalan yang memiliki variabel banyak dalam pendekatan statistik dapat menggunakan metode regresi. Peramalan dengan pendekatan statistik yang menggunakan metode *time series* umumnya merupakan model linier. Metode regresi merupakan metode yang memanfaatkan relasi dan variasi dari variabel, namun karena adanya ketergantungan suatu variabel dengan variabel lain mengakibatkan eksekusi metode yang menghabiskan waktu (Abraham & Ledolter, 2005) (Yaseen, et al., 2015).

Pada peramalan kasus nonlinier dapat menggunakan pendekatan AI karena AI telah menunjukkan kemampuan dalam memprediksi dan memodelkan fenomena dalam peramalan. Ada beberapa metode yang digunakan dalam AI untuk peramalan. Salah satu metode yang digunakan sistem cerdas dalam peramalan adalah *support vector regression* (SVR). SVR merupakan bagian atau cabang dari SVM yang diterapkan dalam menyelesaikan kasus regresi dan permasalahan runut waktu. Penentuan parameter SVR sangat menentukan akurasi dari SVR. Untuk meningkatkan akurasi SVR dibutuhkan suatu metode optimasi. Metode optimasi yang digunakan pada SVR diantaranya adalah *Genetic Algorithm* (GA), *Particle Swarm Optimization* (PSO), *Ant Colony Optimization* (ANT CO), *Bee Colony*, *Firefly* dan metode optimasi lainnya. Dalam optimasi SVR, beberapa algoritma evolusi seperti algoritma genetika dan *Simulated Annealing* telah digunakan untuk menentukan seleksi parameter, namun algoritma tersebut masih terjebak di lokal optimum maka dari itu digunakannya ANT CO. ANT CO memiliki keunggulan pada metode pembelajaran dan pencarian

sehingga dapat memberikan hasil yang baik pada permasalahan optimasi (Yaseen, et al., 2015) (Hong, et al., 2011).

Banyak penelitian tentang peramalan curah hujan yang telah dilakukan sebelumnya salah satunya penelitian yang dilakukan oleh He dkk (2015) untuk meramalkan curah hujan yang menerapkan kecerdasan dengan menggunakan metode gabungan *Hybrid Wavelet Neural Network Mode* (HWNN) dan *Particle Swarm Optimization* (PSO). Penelitian ini dilakukan di Australia yang memiliki anomali cuaca yang terjadi setiap bulannya pada 255 stasiun pengukur hujan yang ada di Australia. Metode HWNN yang digunakan dalam penelitian menunjukkan pengurangan prediksi yang salah dengan persentase curah hujan yang diabaikan sebesar 29.3% dan peramalan dengan taksiran yang terlalu tinggi sebesar 147% (He, et al., 2015)

Selain menggunakan metode HWNN, pada penelitian peramalan dapat menggunakan metode lainnya seperti penelitian yang telah dilakukan oleh Hong, dkk (2011). Wei-Chiang Hong dkk menggunakan metode SVR dengan *Continuous Ant Colony Optimization* (CACO) untuk memperkirakan arus lalu lintas antar kota yang dilakukan di Kota Taiwan. Pada penelitian ini acuan dalam menentukan indeks kesalahan peramalan menggunakan *normalized mean square error* (NRMSE). Hasil dari penelitian menunjukkan bahwa peramalan menggunakan SVRCACO lebih baik dengan menghasilkan NRMSE sebesar 0.2632 daripada penelitian sebelumnya dengan metode SARIMA yang menghasilkan nilai NRMSE sebesar 0.303. Dari penelitian, dengan menggunakan metode SVRCACO memberikan nilai *error* yang lebih kecil daripada menggunakan metode SARIMA

Berdasarkan pemaparan mengenai peramalan curah hujan dan uraian singkat tentang metode SVR-ANT CO, maka dirancang sebuah sistem peramalan yang menggabungkan keberhasilan objek penelitian dan keberhasilan metode yaitu Peramalan Runut Waktu Curah Hujan Menggunakan Metode "SVR - ANT CO". Penggunaan metode *hybrid* diharapkan dapat memberikan nilai dengan akurasi yang tinggi dan dapat diinterpretasikan di dalam peramalan. Model SVR dipilih karena telah terbukti berhasil untuk menyelesaikan masalah regresi nonlinier dan permasalahan runut waktu. Digunakan optimasi dengan menggunakan ANT CO diharapkan dapat mengatasi kelemahan metode SVR yaitu menentukan nilai parameter SVR, salah satu parameter tersebut adalah parameter *epsilon* ( $\epsilon$ ) yang dapat mempengaruhi tingkat akurasi pada metode SVR. Sistem ini diharapkan akan dapat membantu mengatasi permasalahan untuk peramalan curah hujan yang tidak menentu. (Hung & Hong, 2009).

## 1.2 Rumusan masalah

Berdasarkan paparan latar belakang tersebut, maka rumusan masalah yang bisa dikaji adalah sebagai berikut:

1. Bagaimana merancang sistem peramalan runut waktu curah hujan menggunakan metode SVR- ANT CO?
2. Bagaimana mengimplementasikan metode SVR-ANT CO ke dalam sistem untuk peramalan runut waktu curah hujan?

3. Bagaimana hasil evaluasi peramalan runtu waktu curah hujan menggunakan SVR-ANT CO?

### 1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Menerapkan metode SVR dengan optimasi ANT CO untuk peramalan runtu waktu curah hujan.
2. Menguji hasil evaluasi dalam menerapkan metode SVR-ANT CO untuk peramalan runtu waktu curah hujan.
3. Mendapatkan hasil peramalan curah hujan yang tepat dengan menggunakan metode yang diusulkan dalam penelitian ini.

### 1.4 Manfaat

Manfaat penelitian ini adalah:

1. Dapat lebih memahami tentang implementasi sistem peramalan curah hujan dengan metode SVR dan ANT CO.
2. Dapat menerapkan ilmu yang telah didapat terutama tentang sistem peramalan curah hujan.
3. Mempermudah pengguna dalam menentukan peramalan curah hujan yang akurat.

### 1.5 Batasan masalah

Pada bagian ini merupakan batasan-batasan masalah yang diterapkan pada penelitian adalah sebagai berikut:

1. Data curah hujan yang digunakan, diperoleh dari Stasiun klimatologi BMKG Karangploso Malang di wilayah curah hujan di Poncokusumo Kabupaten Malang Jawa Timur, Indonesia dengan rentang waktu tahun 2000-2015.
2. Data curah hujan yang digunakan untuk peramalan adalah data curah hujan per dasarian (data selama 10 hari) dan menggunakan satuan millimeter per dasarian.
3. Data curah hujan yang digunakan untuk proses peramalan yaitu data curah hujan pada wilayah tertentu dan model data berdasarkan hasil analisis data yang dilakukan pada penelitian ini.
4. Menggunakan metode *Support Vector Regression* dengan optimasi *Ant Colont Optimization* (SVR-ANT CO).
5. *Kernel trick* yang digunakan pada metode SVR adalah *Analysis of Variance* (ANOVA) *Radial Basis Function* (RBF) *kernel*.
6. Pengujian akurasi metode yang digunakan pada penelitian ini terhadap data aktual berdasarkan hasil indikator dari *Root Mean Square Error* (RMSE).



## 1.6 Sistematika pembahasan

Untuk mencapai tujuan yang diharapkan, maka sistematika pembahasan yang disusun dalam tugas akhir ini adalah sebagai berikut:

### BAB 1 Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, sistematika penulisan dan waktu pengerjaan yang digunakan untuk merancang sistem Peramalan Runut Waktu Curah Hujan dengan Menggunakan Metode SVR-ANT CO.

### BAB 2 Landasan Kepustakaan

Tinjauan Pustaka menjelaskan tentang kajian pustaka terkait dengan penelitian yang telah ada seperti penelitian tentang peramalan serta metode SVR dan ANT CO. Dasar teori yang diperlukan untuk mendukung penelitian ini adalah peramalan, definisi curah hujan, metode SVR, metode ANT CO, dan metode SVR-ANT CO.

### BAB 3 Metodologi

Membahas tentang algoritma yang digunakan untuk perancangan peramalan runut waktu curah hujan yaitu SVR dan ANT CO dan langkah kerja yang dilakukan penulis dalam penulisan tugas akhir

### BAB 4 Perancangan

Membahas tentang bagaimana perancangan peramalan runut waktu curah hujan menggunakan metode SVR-ANT CO, bagaimana perancangan antarmuka dan perancangan pengujian. Perancangan peramalan terdiri dari definisi masalah peramalan, pengumpulan data, analisis data, seleksi dan validasi data

### BAB 5 Implementasi

Membahas tentang implementasi peramalan runut waktu curah hujan menggunakan metode SVR-ANT CO, bagaimana antarmuka dari sistem dan *source code* untuk pengembangan sistem.

### BAB 6 Pengujian dan Analisis

Membahas evaluasi dan kinerja dengan melakukan pengujian Blackbox dan pengujian akurasi pada sistem dan analisis yang telah direalisasikan dan telah memenuhi kriteria.

### BAB 7 Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian peramalan runut waktu curah hujan yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut.

UNIVERSITAS BRAWIJAYA



## BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepastakaan terdiri atas kajian pustaka dan dasar teori. Kajian pustaka adalah pembahasan tentang penelitian yang sudah dilakukan dan yang diusulkan untuk penelitian berikutnya. Dasar teori adalah pembahasan tentang teori yang diperlukan untuk penelitian seperti curah hujan, data runut waktu, peramalan, Metode SVR, metode ANT CO, dan metode evaluasi kinerja..

### 2.1 Kajian pustaka

Kajian pustaka ini bertujuan untuk membandingkan usulan penelitian dengan judul “Peramalan Runut Waktu Curah Hujan Dengan Menggunakan Metode SVR-ANT CO” dengan beberapa penelitian sebelumnya yang terkait.

Pada peramalan sebelumnya terutama peramalan data runut waktu pada kasus hidrologi, metode yang paling banyak digunakan adalah metode dengan pendekatan kecerdasan buatan. Penggunaan kecerdasan buatan pada peramalan terutama dalam pengetahuan di bidang hidrologi seperti peramalan cuaca dan curah hujan semakin berkembang. Banyak metode seperti optimasi matematika, logika, klasifikasi, *statistical learning*, dan peluang telah digunakan secara luas terutama pada bidang peramalan. Ada empat subkategori penggunaan kecerdasan buatan terutama pada bidang penelitian hidrologi yaitu Pendekatan klasifikasi dan *machine learning*, *Fuzzy set*, *Evolutionary computation* (EC) dan *Wavelet conjunction model* (Yaseen, et al., 2015).

Penelitian selanjutnya yang dilakukan oleh Hong, dkk (2011). Penggunaan metode SVR dengan *continuous* ANT CO digunakan untuk memperkirakan arus lalu lintas antar kota yang dilakukan di Kota Taiwan pada bulan Februari 2005 sampai bulan Maret 2005. ANT CO digunakan untuk menentukan nilai tiga parameter dari model SVR. Karena ANT CO digunakan untuk mendiskritkan nilai kontinyu dari parameter SVR maka dalam penelitian ini disebutkan menggunakan *continuous* ANT CO (CACO). Parameter yang dioptimasi adalah  $C$ ,  $\epsilon$  dan  $\sigma$  yang mempengaruhi hasil dari peramalan dengan menggunakan SVR. Hasil dari penelitian menunjukkan bahwa peramalan menggunakan SVRCACO lebih baik dengan menghasilkan NRMSE sebesar 0.2632% daripada penelitian sebelumnya dengan metode SARIMA yang menghasilkan nilai NRMSE sebesar 0.303%. Dari penelitian tersebut dapat disimpulkan bahwa dengan penerapan metode SVR dengan optimasi CACO dapat menghasilkan hasil optimal dalam menyelesaikan masalah peramalan arus lalu lintas (Hong, et al., 2011).

Penelitian lainnya yang dilakukan Ju & Hong (2013) untuk meramalkan sistem elektrik pada peramalan dengan metode SVR. Pada penelitian ini metode optimasi untuk meningkatkan performa dari peramalan yang digunakan adalah *chaotic gravitational search algorithm* (CGSA). CGSA digunakan untuk mengoptimasi tiga parameter SVR yang terjebak lokal optimal yaitu  $C$ ,  $\epsilon$  dan parameter Gaussian RBF kernel  $\sigma$ . *Mean absolute percentage error* (MAPE) digunakan untuk membandingkan hasil penelitian ini dan penelitian sebelumnya

yang telah dilakukan. Hasil dari penelitian ini menunjukkan nilai MAPE 2.587% menggunakan metode SVRCGSA dan hasil MAPE dari metode pembandingan yaitu ARIMA adalah 6.044%, hal ini menunjukkan tingkat error pada metode SVRCGSA lebih kecil dari pada metode ARIMA walaupun memakan waktu lebih lama daripada metode ARIMA. Dari penelitian tersebut dapat disimpulkan bahwa dengan penerapan metode SVR dengan optimasi CGSA dapat menghasilkan hasil optimal dalam menyelesaikan permasalahan (Ju & Hong, 2013).

Penelitian yang dilakukan oleh Geng, dkk (2015) pada peramalan beban arus listrik. Metode yang digunakan adalah *Support Vector Regression-Chaotic Cloud Simulated Annealing* (SVRCCSA) untuk peramalan beban listrik di China. Data yang digunakan dalam penelitian adalah data historis beban listrik di Cina utara dengan data dari bulan Januari 2004 sampai bulan April 2009. Penelitian yang dilakukan oleh Jing Geng dkk bertujuan untuk mencari kombinasi parameter SVR terbaik. Kombinasi parameter yang dicari adalah parameter  $\sigma$ ,  $C$ ,  $\epsilon$  dengan nilai kombinasi terbaik 342.43, 891.67, 11.626 yang menghasilkan nilai MAPE sebesar 4.389%. Dari penelitian tersebut dapat disimpulkan bahwa dengan penerapan metode SVR dengan optimasi CCSA dapat menghasilkan hasil optimal dalam menyelesaikan permasalahan (Geng, et al., 2015).

Pada penelitian peramalan oleh Liu & Wang (2015) untuk peramalan lalu lintas. Metode yang digunakan dalam penelitian ini adalah SVR yang dioptimasi menggunakan *global artificial fish swarm algorithm* (GAFSA). Data yang digunakan adalah data lalu lintas dari 1 Maret 2012 pukul 0:00 sampai tanggal 13 Maret 2012 pukul 13:00. Data diperoleh dengan mengoleksi data tiap jam dari <http://news.ntu.edu.tw/stat>. Dari 10 kali percobaan GAFSA-SVR yang memiliki akurasi paling dekat dengan nilai rata-rata dengan nilai MAE 0.939%, RMSE 1.153% dan akurasi sebesar 89.852%. Dari penelitian tersebut dapat disimpulkan bahwa dengan penerapan metode SVR dengan optimasi GAFSA dapat menghasilkan hasil optimal dalam menyelesaikan permasalahan (Liu & Wang, 2015).

Penelitian peramalan oleh Hong dkk (2011) yang digunakan untuk meramalkan rencana wisata. Data yang digunakan adalah data kedatangan turis di Barbados pada tahun 1956-2003. Metode yang digunakan dalam peramalan adalah SVR dengan *Chaotic Genetic Algorithm*. Penelitian ini bertujuan untuk menentukan kombinasi parameter SVR yang terbaik. Hasil dari penelitian merupakan nilai error metode yang diuji dengan metode pembandingan lebih rendah pada kasus kedatangan turis dengan nilai MAPE 5.04%. Dari penelitian tersebut dapat disimpulkan bahwa dengan penerapan metode SVR dengan optimasi CGA dapat menghasilkan hasil optimal dalam menyelesaikan permasalahan (Hong, et al., 2011).

Penelitian peramalan oleh Hong dkk (2013) untuk meramalkan siklus ekonomi. Data yang digunakan adalah data bulanan beban elektrik di China dari bulan Januari 2004 sampai bulan April 2009. Metode peramalan yang dipakai adalah SVR dengan *Chaotic Genetic Algorithm*. Penelitian ini bertujuan untuk menentukan kombinasi parameter SVR yang terbaik. Hasil dari penelitian

menunjukkan siklus ekonomi dengan nilai MAPE dengan menggunakan SVR dengan *chaotic genetic algorithm* lebih rendah dari metode pembandingan yaitu sebesar 2.695%. Dari penelitian tersebut dapat disimpulkan bahwa dengan penerapan metode SVR dengan optimasi CGA dapat menghasilkan hasil optimal dalam menyelesaikan permasalahan (Hong, et al., 2013).

Penelitian peramalan oleh Hong dkk (2010) untuk meramalkan keminatan pada ponsel 3G di Taiwan. Data yang digunakan dalam penelitian adalah data dari *subscriber* ponsel 3G dari Chunghwa Telcon Co. pada tahun 2006-2008. Metode yang digunakan pada penelitian adalah SVR dengan *Hybrid Evolutionary Algorithms* GA-SA. Dalam penelitian ini dicari kombinasi terbaik dari parameter SVR. Dari penelitian menghasilkan MAPE menggunakan SVRGA nilai 3.108% dengan nilai parameter  $(\sigma; C; \epsilon) = (0.3959; 1492; 54.855)$  dan SVRGA-SA memiliki solusi lebih baik nilai  $(\sigma; C; \epsilon) = (0.9010; 7681; 21.242)$  dan nilai MAPE=0.720%. Dari penelitian tersebut dapat disimpulkan bahwa dengan penerapan metode SVR dengan optimasi SA-GA dapat menghasilkan hasil optimal dalam menyelesaikan permasalahan (Hong, et al., 2010).

Dari penelitian yang telah dipaparkan, SVR telah banyak digunakan pada berbagai bidang peramalan seperti bidang sosial, ekonomi dan ilmiah. Akan tetapi SVR masih belum bisa menghasilkan nilai akurasi yang baik, hal tersebut dikarenakan parameter yang ada pada metode SVR. Maka dari itu dibutuhkan sebuah optimasi untuk menentukan kombinasi parameter yang tepat sehingga dapat menghasilkan akurasi yang tinggi. Beberapa metode yang digunakan untuk mengoptimasi SVR yaitu GA, PSO, ANT CO, *Bee*, *Firefly* dan sebagainya. Pada penelitian ini akan dirancang sistem peramalan tentang curah hujan dengan metode SVR yang dioptimasi dengan menggunakan metode ANT CO yaitu "Peramalan Runut Waktu Curah Hujan Menggunakan Metode "SVR - ANT CO"". ANT CO memiliki keunggulan pada metode pembelajaran dan pencarian sehingga dapat memberikan hasil yang baik dan memberikan hasil peramalan yang optimal. Perbandingan penelitian tugas akhir dan penelitian-penelitian peramalan sebelumnya disajikan pada Tabel 2.1

Tabel 2.1 Kajian Pustaka

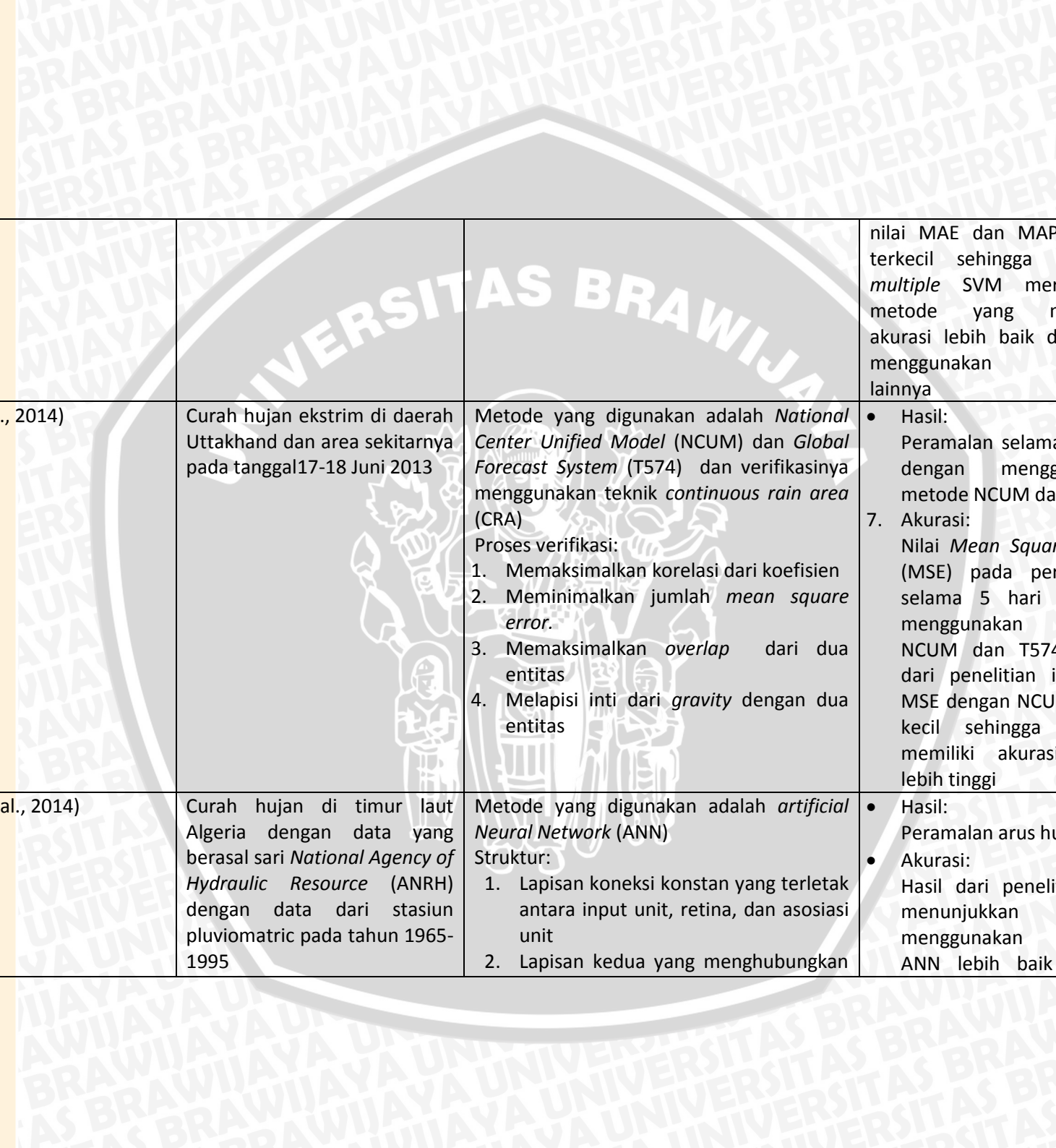
No	Peneliti	Objek	Metode(Proses)	Hasil
1	(Hong, et al., 2010)	Peramalan keminatan pada ponsel 3G. data dari <i>subscriber</i> ponsel 3G dari Chunghwa Telcon Co. pada tahun 2006-2008	Metode yang digunakan SVR dengan <i>Hybrid Evolutionary Algorithms</i> GA-SA Proses: <ol style="list-style-type: none"> <li>1. Inisialisasi</li> <li>2. Evaluasi <i>fitness</i></li> <li>3. Operasi seleksi</li> <li>4. <i>Crossover</i></li> <li>5. Mutasi</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Kombinasi parameter terbaik</li> <li>• Akurasi: Hasil MAPE menggunakan SVRGA nilai 3.108&amp; dengan nilai parameter (<math>\sigma</math>; C ; <math>\epsilon</math>) = (0.3959 ; 1492 ; 54.855) dan SVRGA-SA memiliki solusi lebih baik nilai (<math>\sigma</math>; C ; <math>\epsilon</math>) = (0.9010 ; 7681 ; 21.242) dan nilai MAPE=0.720%</li> </ul>
2	(Hong, et al., 2011)	Peramalan rencana pariwisata. Dengan data kedatangan turis di Barbados pada tahun 1956-2003	Metode yang digunakan SVR dengan <i>Chaotic Genetic Algoritm</i> Proses: <ol style="list-style-type: none"> <li>1. Inisialisasi parameter</li> <li>2. Mendefinisikan jumlah populasi</li> <li>3. Perhitungan <i>fitness</i></li> <li>4. Seleksi <i>parent</i></li> <li>5. <i>Crossover</i></li> <li>6. Mutasi dengan <i>Annealing Chaotic</i></li> </ol>	Nilai error metode yang diuji dengan metode pembandingan lebih rendah pada kasus kedatangan turis dengan nilai MAPE= 5.04%

3	(Hong, et al., 2011)	<p>Arus lalu lintas di Taiwan pada bulan Februari 2005 sampai bulan Maret 2005</p> <p>Data input:</p> <ul style="list-style-type: none"> <li>• <i>Morning peak period</i> (jam 06.00 sampai 10.00)</li> <li>• <i>Evening peak period</i> (jam 16.00 sampai 20.00)</li> </ul> <p>Data latih:</p> <ul style="list-style-type: none"> <li>• <i>Morning peak period</i> :25 jam</li> <li>• <i>Evening peak period</i>: 60 jam</li> </ul> <p>Data validasi:</p> <ul style="list-style-type: none"> <li>• <i>Morning peak period</i> :10 jam</li> <li>• <i>Evening peak period</i>: 15 jam</li> </ul> <p>Data uji:</p> <ul style="list-style-type: none"> <li>• <i>Morning peak period</i> :10 jam</li> <li>• <i>Evening peak period</i>: 15 jam</li> </ul>	<p>Metode: SVR <i>with continuous</i> ACO</p> <p>Proses:</p> <ol style="list-style-type: none"> <li>1. Inisialisasi awal parameter SVR yaitu <math>C</math>, <math>\epsilon</math> dan <math>\sigma</math>.</li> <li>2. Memberikan tugas kepada setiap semut dalam koloni</li> <li>3. Menentukan jumlah semut dan menghitung jarak antar kota</li> <li>4. Pemberhentian iterasi ketika iterasi maksimum telah tercapai atau masuk ke langkah selanjutnya</li> <li>5. Menghitung peluang kunjungan semut</li> <li>6. Pemberhentian iterasi ke II jika semut sudah menyelesaikan jalur antara sumber makanan dan sarang melalui semua kota dan menghasilkan jarak terpendek sebagai solusi</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Hasil peramalan arus lalu lintas di Taiwan</li> <li>• Akurasi: nilai NRSME sebagai berikut <ul style="list-style-type: none"> <li><i>Morning peak period</i> <ul style="list-style-type: none"> <li>- SARIMA = 0.3039%</li> <li>- SVRCACO = 0.2632%</li> </ul> </li> <li><i>Evening peak period</i> <ul style="list-style-type: none"> <li>- SARIMA = 0.1821%</li> <li>- SVRCACO = 0.1033%</li> </ul> </li> </ul> </li> </ul>
4	(Ju & Hong, 2013)	<p>Sistem elektrik</p> <p>Input:</p> <p>Nilai parameter SVR yaitu <math>C</math>, <math>\epsilon</math> dan <math>\sigma</math>.</p>	<p>Metode: SVR-CGSA</p> <p>Proses:</p> <ol style="list-style-type: none"> <li>1. Inisialisasi awal parameter SVR yaitu <math>C</math>, <math>\epsilon</math> dan <math>\sigma</math>.</li> <li>2. Menghitung nilai <i>error</i> dalam peramalan dari semua pasangan agen</li> <li>3. <i>Update</i> status dan kecepatan dan juga posisi dari setiap pasangan agen</li> <li>4. Mencapai iterasi maksimal CGSA</li> <li>5. Melakukan proses <i>chaotic algoritma search</i></li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Menghasilkan peramalan permintaan listrik bulanan yang lebih unggul</li> <li>• Akurasi: SVRCGSA <ol style="list-style-type: none"> <li>1. MAPE = 2.587%</li> <li>2. MAPE = 6.044%,</li> </ol> Hal ini menunjukkan </li> </ul>

			<ol style="list-style-type: none"> <li>6. Menghitung iterasi <i>chaotic</i> berikutnya</li> <li>7. Melakukan transformasi untuk mendapatkan parameter</li> <li>8. Menghitung nilai objektif yang baru</li> <li>9. Menghentikan kriteria jika ada akurasi yang lebih kecil dari nilai indeks atau iterasi maksimal sudah tercapai</li> </ol>	tingkat error pada metode SVRCGSA lebih kecil dari pada metode ARIMA walaupun memakan waktu lebih lama daripada metode ARIMA
5	(Lin, et al., 2013)	<p>Peramalan curah hujan badai dengan input yang memiliki kombinasi yang optimal dengan parameter yang digunakan</p> <ol style="list-style-type: none"> <li>1. Tekanan udara (P)</li> <li>2. Temperatur udara (T)</li> <li>3. Kecepatan angin (W)</li> <li>4. Arah angin (D)</li> <li>5. Durasi sinar matahari (S)</li> </ol>	<p>Metode yang digunakan adalah <i>multi-objective genetic algorithm</i> (MOGA) <i>Support Vector Machine</i> (SVM).</p> <p>Proses:</p> <ol style="list-style-type: none"> <li>1. <i>Encoding</i> pada kromosom</li> <li>2. Menentukan jumlah populasi</li> <li>3. Perhitungan fungsi dan nilai fitness</li> <li>4. Penggunaan strategi <i>elitism</i></li> <li>5. Melakukan terminasi tes</li> <li>6. Seleksi</li> <li>7. <i>Crossover</i></li> <li>8. Mutasi</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Hasil kombinasi masukan yang paling optimal yaitu P, T, W, D, S, dan R</li> <li>• Akurasi: -</li> </ul>
6	(Pan, et al., 2013)	<p>Peramalan banjir dengan curah hujan di Taiwan. Data yang digunakan berasal dari hulu sungai Tsengwen dengan luas 481km<sup>2</sup>, curah hujan tahunan rata-rata 2.700mm dan aliran sungai rata-rata tahunan 29,0 m<sup>3</sup>s</p>	<p>Metode yang digunakan ANN.</p> <p>Proses:</p> <ol style="list-style-type: none"> <li>1. inialisasi parameter</li> <li>2. Perhitungan ANN</li> <li>3. Menghitung rata-rata curah hujan</li> <li>4. Menghitung efektivitas curah hujan</li> <li>5. Ruang NN untuk proses curah hujan</li> <li>6. Evaluasi performa</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Perhitungan nilai error pada penelitian dengan tiga model observasi</li> <li>• Akurasi: -</li> </ul>



7	(Hong, et al., 2013)	Peramalan siklus ekonomi. Dengan data bulanan beban elektrik di China dari Januari 2004 sampai April 2009	Metode yang digunakan SVR dengan <i>chaotic genetic algorithm</i> Proses: 1. Inisialisasi parameter 2. Mendefinisikan jumlah populasi 3. Perhitungan <i>fitness</i> 4. Seleksi <i>parent</i> 5. <i>Crossover</i> Mutasi dengan Annealing chaotic	<ul style="list-style-type: none"> <li>• Hasil: Hasil peramalan siklus ekonomi</li> <li>• Akurasi: Nilai MAPE dengan menggunakan SVR dengan <i>chaotic genetic algorithm</i> lebih rendah yaitu sebesar 2.695%</li> </ul>
8	(Yan & Chowdhury, 2014)	<i>Electricity market clearing price (MCP)</i> Input: PJM <i>interconnected market</i> dengan data pada bulan Januari sampai Desember 2009	Metode: <i>Multiple SVM</i> Proses: 1. Inisialisasi data input 2. Klasifikasi data input 3. Melakukan prediksi menggunakan SVM pada variabel <i>peak price, high price, medium price, low price</i> 4. Menghitung nilai MAE dan MAPE	<ul style="list-style-type: none"> <li>• Hasil: Perbandingan hasil peramalan dengan menggunakan SVM</li> <li>• Akurasi:             <ol style="list-style-type: none"> <li>3. <i>single SVM</i> <ul style="list-style-type: none"> <li>• MAE 7.2523%</li> <li>• MAPE 15.6454%</li> </ul> </li> <li>4. <i>single LSSVM</i> <ul style="list-style-type: none"> <li>• MAE 7.9003%</li> <li>• MAPE 16.2610%</li> </ul> </li> <li>5. <i>hybrid LSSVM-ARMAX</i> <ul style="list-style-type: none"> <li>• MAE 11.0989%</li> <li>• MAPE 13.9709%</li> </ul> </li> <li>6. <i>multiple SVM</i> <ul style="list-style-type: none"> <li>• MAE = 5.6958%</li> <li>• MAPE=1.3875%</li> </ul> </li> </ol> </li> </ul> <p><i>multiple SVM</i> menghasilkan</p>



				nilai MAE dan MAPE yang terkecil sehingga metode <i>multiple SVM</i> merupakan metode yang memiliki akurasi lebih baik daripada menggunakan metode lainnya
9	(Dube, et al., 2014)	Curah hujan ekstrim di daerah Uttakhand dan area sekitarnya pada tanggal 17-18 Juni 2013	<p>Metode yang digunakan adalah <i>National Center Unified Model (NCUM)</i> dan <i>Global Forecast System (T574)</i> dan verifikasi menggunakan teknik <i>continuous rain area (CRA)</i></p> <p>Proses verifikasi:</p> <ol style="list-style-type: none"> <li>1. Memaksimalkan korelasi dari koefisien</li> <li>2. Meminimalkan jumlah <i>mean square error</i>.</li> <li>3. Memaksimalkan <i>overlap</i> dari dua entitas</li> <li>4. Melapisi inti dari <i>gravity</i> dengan dua entitas</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Peramalan selama 5 hari dengan menggunakan metode NCUM dan T574.</li> <li>7. Akurasi: Nilai <i>Mean Square Error (MSE)</i> pada peramalan selama 5 hari dengan menggunakan metode NCUM dan T574. Hasil dari penelitian ini nilai MSE dengan NCUM lebih kecil sehingga NCUM memiliki akurasi yang lebih tinggi</li> </ul>
10	(Dounia, et al., 2014)	Curah hujan di timur laut Algeria dengan data yang berasal dari <i>National Agency of Hydraulic Resource (ANRH)</i> dengan data dari stasiun pluviometric pada tahun 1965-1995	<p>Metode yang digunakan adalah <i>artificial Neural Network (ANN)</i></p> <p>Struktur:</p> <ol style="list-style-type: none"> <li>1. Lapisan koneksi konstan yang terletak antara input unit, retina, dan asosiasi unit</li> <li>2. Lapisan kedua yang menghubungkan</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Peramalan arus hujan</li> <li>• Akurasi: Hasil dari penelitian ini menunjukkan dengan menggunakan metode ANN lebih baik dalam</li> </ul>

			unit asosiasi dan unit jawaban	memodelkan arus hujan dengan nilai Nash sebesar 95.5% dibandingkan dengan model pembanding yaitu GR2M dengan nilai Nash 87.3%
11	(Abbot & Marohasy, 2014)	Curah hujan di daerah Queensland, Australia dengan input: <ol style="list-style-type: none"> <li>1. Variable lokal</li> <li>8. Curah hujan</li> <li>9. Temperatur maksimum</li> <li>10. Temperatur minimum</li> <li>11. Tekanan atmosfer</li> <li>2. Indeks iklim</li> <li>12. SOI</li> <li>13. SST1</li> <li>14. SST2</li> <li>3. Tambahan indeks iklim</li> <li>15. Nino</li> <li>16. SAM</li> <li>17. DMI</li> <li>18. IPO</li> </ol>	Metode yang digunakan <i>artificial neural network</i> (ANN) Proses: <ol style="list-style-type: none"> <li>1. <i>Pre-screening</i> variable input</li> <li>2. Menguji kombinasi variabel</li> <li>3. Membandingkan output model ANN dengan output model POAMA</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil:                              Dari penelitian metode peramalan dengan ANN dapat meramal musim panas yang kering pada tahun 2001-2002 dan 2002-2003, secara akurat meramal musim panas yang lembab pada tahun 2008-2008 dan 2008-2009 namun tidak dapat meramal dengan baik musim panas lembab pada tahun 1999-2000 dan 2003-2004</li> <li>• Akurasi:                              -</li> </ul>
12	(Löwe, et al., 2014)	Observasi curah hujan dengan menggunakan alat pengukur hujan dan radar cuaca untuk peramalan operasional aliran air di perkotaan. Dengan	Metode yang digunakan adalah <i>stochastic flow forecasting</i> dengan menggunakan alat pengukur hujan dan radar cuaca. Proses: <ol style="list-style-type: none"> <li>1. Pencarian mean area</li> </ol>	Observasi menggunakan menghasilkan ramalan dengan meningkatkan peluang peramalan aliran air daripada menggunakan alat

		periode data dari tanggal 25 Juni 2010 sampai 6 September 2010	2. DAS terpadu 3. Distributed subcatchment	pengukur hujan
13	(Schepen & Wang, 2014)	Peramalan curah hujan di Australia dengan periode dari tahun 1980-2010 dengan tiga lokasi tangkapan yaitu Burdekin (36,260 km <sup>2</sup> ), Namoi (2532 km <sup>2</sup> ), Acheron (639 km <sup>2</sup> ).	Ada tiga metode yang digunakan yaitu <i>Bayesian joint probability</i> , <i>Bayesian model</i> dan <i>Schaake Shuffle</i> . Proses: 1. Bayesian joint probability 2. Menghitung nilai rata-rata <i>Bayesian</i> 3. <i>sequencing</i>	<ul style="list-style-type: none"> <li>• Hasil: Menghasilkan <i>ensemble time series</i> dan dengan bias yang kecil pada peramalan</li> <li>• Akurasi: -</li> </ul>
14	(He, et al., 2015)	Curah hujan dari Australian Bureau of Meteorology Data input: Data curah hujan dari 255 stasiun di Australia Data latih : Tahun 1959-1998 Data Validasi: Tahun 1989-1998 Data Uji : data bulan Januari 1990 sampai Desember 2008	Metode: HWNN Proses: 1. Melakukan standarisasi rangkaian anomali curah hujan 2. Mendekomposisi rangkaian yang berpotensi dengan menggunakan MODWT 3. Memilih <i>subset</i> pada setiap level dekomposisi dari input kandidat <i>predictor</i> dengan menggunakan modifikasi algoritma PMI 4. Menentukan nilai optimal dari jaringan yang tersembunyi pada ANN pada level dekomposisi yang berbeda dengan menggunakan algoritma PSO 5. Prakiraan anomali curah hujan bulanan dengan membuat model	<ul style="list-style-type: none"> <li>• Hasil: Hasil peramalan curah hujan di Australia</li> <li>• Akurasi: Metode peramalan HWNN mengurangi nilai <i>error absolute</i> sebesar 0.16-0.92 dan meningkatkan nilai efisiensi <i>Nash-Sutcliffe</i> sebesar 0.17-1.80 dibandingkan menggunakan model pembanding yaitu <i>Artificial Neural Network (ANN)</i>.</li> </ul>

			<p>yang optimal dan menghitung semua peramalan pada subseri <i>anomaly</i></p> <p>6. Mendapatkan output dari peramalan dengan <i>inverse transform</i> dari standarisasi anomali bulanan</p>	
15	(Geng, et al., 2015)	Data historis beban listrik di Cina utara dengan data dari bulan Januari 2004 sampai bulan April 2009	<p>Metode yang digunakan <i>Support Vector Regression-Chaotic Cloud Simulated Annealing</i> (SVRCCSA)</p> <p>Proses:</p> <ol style="list-style-type: none"> <li>1. Inisialisasi parameter SVR ke variabel chaotic</li> <li>2. Menentukan kondisi Y dari <i>cloud generator</i></li> <li>3. Implementasi CSA</li> <li>4. <i>Testing</i></li> <li>5. Solusi</li> <li>6. Pengurangan temperatur</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Peramalan beban listrik dengan kombinasi parameter SVR terbaik</li> <li>• Akurasi: Perhitungan MAPE untuk menentukan tingkat akurasi. SVRCCSA memiliki nilai MAPE sebesar 4.389% dengan nilai parameter <math>\sigma</math>, <math>C</math>, <math>\epsilon</math> adalah 342.43, 891.67, 11.626.</li> </ul>
16	(Liu & Wang, 2015)	Data lalu lintas dari 1 Maret 2012 pukul 0:00 sampai tanggal 13 Maret 2012 pukul 13:00 dengan mengoleksi data tiap jam dari <a href="http://news.ntu.edu.tw/stat">http://news.ntu.edu.tw/stat</a>	<p>Metode yang digunakan adalah GAFSA-SVR</p> <p>Proses:</p> <ol style="list-style-type: none"> <li>1. Pengumpulan lalu lintas data jaringan</li> <li>2. Memprediksi verifikasi</li> <li>3. Penetapan <i>sample set</i></li> <li>4. Membandingkan dengan model lain dan mengatur parameter</li> <li>5. Evaluasi kriteria</li> <li>6. Evaluasi hasil</li> <li>7. Analisis hasil</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Menghasilkan peramalan lalu lintas</li> <li>• Akurasi: Dari 10 kali percobaan GAFSA-SVR yang memiliki akurasi paling dekat dengan nilai rata-rata dengan nilai MAE = 0.939%, RMSE= 1.153%</li> </ul>

				dan akurasi sebesar 89.852%
17	(Yaseen, et al., 2015)	Aliran sungai yang telah diteliti sebelumnya	<p>Ada empat metode utama untuk peramalan aliran sungai yaitu</p> <ol style="list-style-type: none"> <li>1. <i>Artificial neural network</i> (ANN)</li> <li>2. <i>Support vector machine</i> (SVM)</li> <li>3. Logika Fuzzy</li> <li>4. <i>Evolutionary Computing</i></li> <li>5. <i>Wavelet-complementary</i></li> </ol> <p>Proses:</p> <ol style="list-style-type: none"> <li>1. <i>Preposesing</i> data</li> <li>2. Memilih variabel input yang paling sesuai</li> <li>3. Memilih model yang paling akurat</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Review dari jurnal-jurnal tentang aliran sungai selama periode tahun 2000-2015 dan mengidentifikasi penelitian ke depan dengan menggunakan metode yang inovatif seperti <i>Fast Orthogonal Search</i> (FOS) dan <i>Swarm Intelligence</i> (SI)</li> <li>• Akurasi: -</li> </ul>
18	(Marohasy & Abbot, 2015)	Curah hujan dengan model temperatur time series pada data curah hujan bulanan di Cape Otway, Australia 70% dari data digunakan untuk training, 10% data untuk testing	<p>Metode yang digunakan adalah <i>Artificial Neural Network</i> yang digunakan untuk memaksimalkan kualitas data pada <i>training</i> dan validasi.</p> <p>Proses:</p> <ol style="list-style-type: none"> <li>1. Penentuan temperatur <i>time series</i></li> <li>2. <i>Embedded</i> dengan <i>multi dimentional array</i></li> <li>3. Menghitung <i>root mean square error</i></li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Hasil dari penelitian ini menunjukkan bahwa pada peramalan curah hujan bulanan tidak ada input variable yang bertanggungjawab dengan memberikan hasil terbaik pada</li> </ul>

			<p>(RMSE)</p> <ol style="list-style-type: none"> <li>Menghitung <i>mean absolute error</i> (MAE)</li> <li>Menghitung korelasi pearson</li> <li>Membandingkan RMSE, MAE dan korelasi pearson menggunakan <i>one-way ANOVA</i></li> </ol>	<p>peramalan.</p> <ul style="list-style-type: none"> <li>Akurasi: -</li> </ul>
19	(Zhou, et al., 2015)	<p>Level racun <i>phenol</i> pada <i>Photobacterium Phosphoreum</i> dengan pengujian 18 tipe <i>phenol</i> dan 13 desain <i>phenol</i></p>	<p>Metode yang digunakan QSAR dengan menggunakan tiga <i>descriptor</i> dan SVR.</p> <p>Proses:</p> <ol style="list-style-type: none"> <li>Memilih metode seleksi <i>descriptor</i> yang dipakai</li> <li>Validasi model SVR internal dan eksternal</li> <li>Validasi stabilitas dari model terbaik</li> <li>Desain <i>phenol</i> dan prediksi</li> <li>Interpretasi model</li> </ol>	<ul style="list-style-type: none"> <li>Hasil: Menentukan <i>descriptor</i> SVR terbaik</li> <li>Akurasi: -</li> </ul>
20	(Urraca, et al., 2016)	<p>Radiasi surya,dengan data set dari stasiun dari <i>Spanish Agency for Irrigation in Agriculture</i> (SIAR) dengan variabel sebagai berikut:</p> <ol style="list-style-type: none"> <li>Curah hujan (R)</li> <li>Kecepatan angin (WS)</li> <li>Temperatur (T)</li> <li>Kelembaban relative (RH)</li> </ol>	<p>Metode yang digunakan adalah <i>support vector regression</i> dan <i>random forest</i> beserta regresi linier klasik dan KNN. Dan algoritma genetic yang digunakan dalam untuk proses data <i>training</i></p> <p>Proses:</p> <ol style="list-style-type: none"> <li>Menetapkan model, dengan cara melakukan data latih selama periode tahun yang berbeda</li> <li><i>Moving</i> model, ada dua pendekatan yaitu melalui <i>instance</i> sebelumnya dan <i>instance</i> yang mirip. Cara</li> </ol>	<ul style="list-style-type: none"> <li>Hasil: Hasil dari penelitian ini berupa perbandingan nilai MAE dan kemampuan peramalan dari SVM, KNN, dan <i>linier regression</i> pada data latih dengan jumlah <i>instance</i> yang berbeda.</li> <li>Akurasi: -</li> </ul>

			instance sebelumnya adalah dengan melatih dengan <i>sample</i> sebelumnya sampai ke <i>sample</i> uji. Sedangkan instance yang mirip dengan melatih data yang paling dekat sebesar $n$ terhadap <i>sample</i> dan <i>sample</i> uji.	
21	Peramalan Runut Waktu Curah Hujan Menggunakan Metode "SVR - Ant CO"	Data curah hujan BMKG dengan rentang waktu 2000-2015. Wilayah yang digunakan adalah wilayah Poncokusumo Malang Data yang digunakan adalah data dasarian (persepuluh hari)	<p>Metode: SVR -ANT CO</p> <p>Proses:</p> <ol style="list-style-type: none"> <li>1. Inialisasi awal parameter ANT CO</li> <li>2. Menentukan node semut             <ul style="list-style-type: none"> <li>- Menentukan node demut pertama (<math>r</math>)</li> <li>- Menentukan node selanjutnya (<math>u</math>)</li> <li>- Memperbarui <i>pheromone</i> local</li> </ul> </li> <li>3. Konversi nilai parameter SVR</li> <li>4. Pelatihan SVR</li> <li>5. Memperbarui <i>pheromone</i> global</li> <li>6. Pengujian SVR</li> </ol>	<ul style="list-style-type: none"> <li>• Hasil: Peramalan runut waktu curah hujan</li> <li>• Akurasi: -RMSE= 9.482</li> </ul>



## 2.2 Curah hujan

Suatu proses fisis yang dihasilkan oleh fenomena cuaca disebut dengan hujan. Cuaca sendiri adalah suatu sistem yang kompleks sehingga sulit untuk memprakirakan secara tepat. Definisi dari curah hujan adalah ketinggian air hujan yang terkumpul dalam penakar hujan pada tempat yang datar, tidak menyerap, tidak meresap dan tidak mengalir. Curah hujan merupakan salah satu unsur iklim yang berpengaruh di berbagai sektor kehidupan seperti pertanian, manajemen sumber daya air dan manajemen risiko bencana. Pola distribusi hujan serta besarnya curah hujan yang terjadi pada suatu tempat berperan dalam penentuan dampak dari akibat hujan. Satuan yang digunakan dalam mengukur curah hujan adalah milimeter (mm). Unsur hujan pada setiap satu milimeter artinya dalam luasan satu meter persegi pada tempat yang datar tertampung air hujan setinggi satu milimeter air hujan sebanyak satu liter (Tukidi, 2010) (Estiningtyas, et al., 2007) (BMKG, 2012).

Perbandingan antara jumlah curah hujan selama rentang waktu yang ditetapkan seperti dalam satu periode musim hujan atau satu periode musim kemarau dengan jumlah curah normal hujan yaitu yang dihitung selama 30 tahun periode menentukan sifat hujan. Ada 3 kategori sifat hujan yaitu (BMKG, 2012)

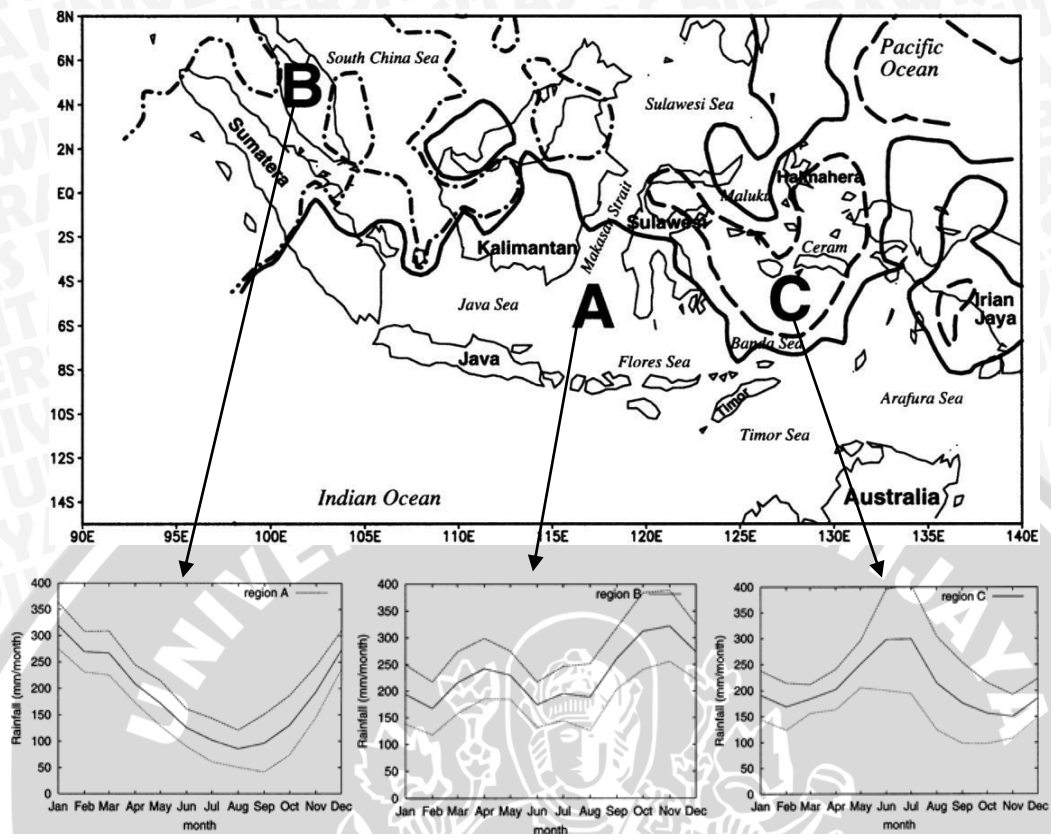
1. Di atas Normal (AN), jika curah hujan lebih dari 115% terhadap rata-ratanya.
2. Normal (N), jika nilai curah hujan antara 85%-115% terhadap rata-rata.
3. Dibawah Normal (BN), jika nilai curah hujan kurang dari 85% terhadap nilai rata-ratanya.

Adapun istilah-istilah normal curah hujan yang dipaparkan oleh Badan Meteorologi Klimatologi dan Geofisika (BMKG) (BMKG, 2015) yaitu

1. Rata-rata Curah Hujan Bulanan  
Nilai rata-rata curah hujan masing-masing bulan dengan periode minimal 10 tahun.
2. Provisional Normal Curah Hujan  
Nilai rata-rata curah hujan masing-masing bulan dengan periode waktu yang dapat ditentukan secara bebas dan diisyaratkan minimal 10 tahun.
3. Normal Curah Hujan Bulanan  
Nilai rata-rata curah hujan masing-masing bulan selama periode 30 tahun.
4. Standar Normal Curah Hujan Bulanan  
Nilai rata-rata curah hujan pada masing-masing bulan selama 30 tahun, dimulai dari tahun 1901 s/d 1930, 1931 s/d 1960, 1961 s/d 1990, 1991 s/d 2010 dan seterusnya.

### 2.2.1 Pola Curah Hujan Indonesia

Menurut Aldrian dan Dwi Susanto (2003) Indonesia memiliki tiga pola iklim utama yang dilihat berdasarkan terjadinya hujan dalam periode satu tahun. Tiga pola curah hujan dijelaskan pada Gambar 4.1 yang memberikan gambaran pola curah hujan berdasarkan rata-rata tahunan di beberapa wilayah Indonesia.



**Gambar 2.1 Pola curah hujan di Indonesia**

Sumber: (Aldrian & Dwi Susanto, 2003).

1. Pola Monsunal

Bersifat unimodial yaitu memiliki satu puncak musim hujan pada akhir tahun dan awal tahun selanjutnya yaitu biasanya terjadi pada bulan Desember sampai bulan Februari. Enam bulan selanjutnya merupakan musim pancaroba yaitu tiga bulan musim peralihan dari musim hujan ke musim kemarau dan tiga bulan berikutnya peralihan dari musim kemarau ke musim hujan. Musim kering biasa terjadi pada pertengahan tahun yaitu bulan Juli, Juni dan Agustus. Pola monsunial dominan terjadi di Pulau Sumatera bagian selatan, Kalimantan selatan dan tengah, Jawa, Bali, Nusa Tenggara dan sebagian di Papua.

2. Pola ekuatorial

Bersifat bimodial yaitu memiliki dua musim puncak hujan yang terjadi pada bulan Maret dan Oktober atau saat terjadi ekinoks. Pola ini biasanya terjadi di Pulau Sumatera bagian tengah dan utara dan Kalimantan bagian utara.

3. Pola lokal

Bersifat unimodial, hampir sama dengan monsunial tetapi bentuknya berlawanan. Dengan demikian pola lokal terjadi saat pertengahan tahun. Pola ini biasanya terjadi pada Pulau Maluku, Sulawesi dan beberapa wilayah Papua.

### 2.2.2 Standardized Precipitation Index (SPI)

Dalam suatu periode waktu yang panjang (satu bulan, dua bulan, tiga bulan dan seterusnya) digunakan sebuah indeks yang menentukan penyimpangan curah hujan terhadap normalnya, indeks tersebut disebut dengan *Standardized Precipitation Index* (SPI). Perhitungan nilai SPI menggunakan metode statistik probalistik distribusi *gamma*. Nilai SPI menentukan tingkat kekeringan dan kebasahan dengan kategori sebagai berikut:

1. Tingkat Kekeringan
  - a. Sangat Kering : jika nilai SPI  $\leq -2.00$  dengan probabilitas 2.3%
  - b. Kering : jika nilai SPI  $-1.50$  s/d  $-1.99$  dengan probabilitas 4.4%
  - c. Agak Kering : jika nilai SPI  $-1.00$  s/d  $-1.49$  dengan probabilitas 9.2%
2. Tingkat Normal
  - a. Normal : jika nilai SPI  $-0.99$  s/d  $0.99$  dengan probabilitas 68.2%
3. Tingkat Kebasahan
  - a. Sangat Basah : jika nilai SPI  $\geq 2.00$  dengan probabilitas 2.3%
  - b. Basah : jika nilai SPI  $1.50$  s/d  $1.99$  dengan probabilitas 4.4%
  - c. Agak Basah : jika nilai SPI  $1.00$  s/d  $1.49$  dengan probabilitas 9.2%

Kekeringan meteorologis merupakan berkurangnya curah hujan dari keadaan normalnya dalam jangka waktu yang panjang (satu bulan, dua bulan, tiga bulan dan seterusnya). Untuk menghitung nilai SPI sendiri menggunakan data curah hujan selama tiga bulanan (BMKG, 2015).

### 2.2.3 Fenomena yang mempengaruhi Iklim/Musim di Indonesia

Fenomena yang mempengaruhi iklim dan musim pada wilayah Indonesia dibagi menjadi dua yaitu fenomena global dan fenomena lokal (BMKG, 2015).

#### 2.2.3.1 Fenomena Global

1. El Nino dan La Nina
 

El Nino merupakan fenomena global dari sistem interaksi lautan atmosfer yang ditandai dengan memanasnya suhu muka laut di Ekuador Pasifik Tengah atau anomali suhu muka laut di daerah tersebut positif yaitu lebih panas dari rata-ratanya. Kebalikan dari El Nino adalah La Nina. La Nina ditandai dengan anomali suhu muka laut negatif atau lebih dingin dari rata-ratanya. Pengaruh El Nino di wilayah Indonesia terjadi bila kondisi suhu perairan Indonesia cukup dingin yang diikuti dengan berkurangnya curah hujan secara drastis. Sedangkan La Nina menyebabkan curah hujan di Indonesia meningkat bila diikuti dengan menghangatnya suhu muka laut di perairan Indonesia. Namun fenomena El Nino dan La Nina tidak berpengaruh di seluruh wilayah Indonesia, hanya daerah-daerah tertentu yang menerima dampak pengaruhnya (BMKG, 2015).
2. Dipole Mode
 

Fenomena interaksi laut dan atmosfer di Samudra Hindia yang dihitung berdasarkan perbedaan nilai atau selisih antara anomali suhu muka laut perairan pantai timur Afrika dengan perairan di sebelah barat Sumatera

merupakan definisi dari Dipole Mode. Sedangkan perbedaan anomali suhu muka laut disebut sebagai Dipole Mode Indeks (DMI) (BMKG, 2015).

3. Madden Agustusan Oscillation (MAO)

Osilasi aktivitas pertumbuhan awan-awan sepanjang jalur dimulai dengan perairan Afrika Timur hingga perairan Pasifik bagian barat (utara Papua) mengindikasikan MAO. Periode dari peristiwa osilasi sendiri relatif pendek yaitu sekitar 30 sampai 50 hari (BMKG, 2015).

### 2.2.3.2 Fenomena Lokal

1. Sirkulasi Monsun Asia-Australia

Penentu sirkulasi angin di Indonesia adalah pola perbedaan tekanan udara di Australia dan Asia. Pola tekanan udara tersebut mengikuti pola peredaran matahari dalam setahun yang mengakibatkan sirkulasi angin di Indonesia membentuk pola monsun. Pola monsun sendiri adalah sirkulasi angin yang mengalami perubahan arah setiap setengah tahun sekali. Pola angin barat terjadi karena adanya tekanan tinggi di Asia yang berkaitan dengan berlangsungnya musim hujan di Indonesia. Pola angin timur/tenggara terjadi karena adanya tekanan tinggi di Australia yang berkaitan dengan berlangsungnya musim kemarau di Indonesia (BMKG, 2015).

2. Daerah Pertemuan Angin antar Tropis (*Inter Tropical Convergence Zone/ITCZ*)

Indonesia merupakan wilayah yang berada di daerah khatulistiwa yang menyebabkan wilayah Indonesia yang dilewati oleh ITCZ yang pada umumnya berpotensi pada terjadinya pertumbuhan awan-awan. ITCZ sendiri merupakan daerah tekanan rendah yang memanjang dari barat ke timur dengan posisi selalu berubah mengikuti pergerakan posisi matahari ke arah utara dan selatan khatulistiwa (BMKG, 2015).

3. Suhu Permukaan Laut di Wilayah Perairan Indonesia

Salah satu indikator banyak sedikitnya kandungan uap air di atmosfer yang berkaitan dengan proses pembentukan awan di wilayah perairan Indonesia adalah kondisi suhu permukaan laut. Jika suhu muka laut dingin, maka akan berpotensi mengurangi kandungan uap air di atmosfer dan sebaliknya jika suhu permukaan laut panas berpotensi menambah kandungan uap air di atmosfer (BMKG, 2015).

## 2.3 Data Runut Waktu

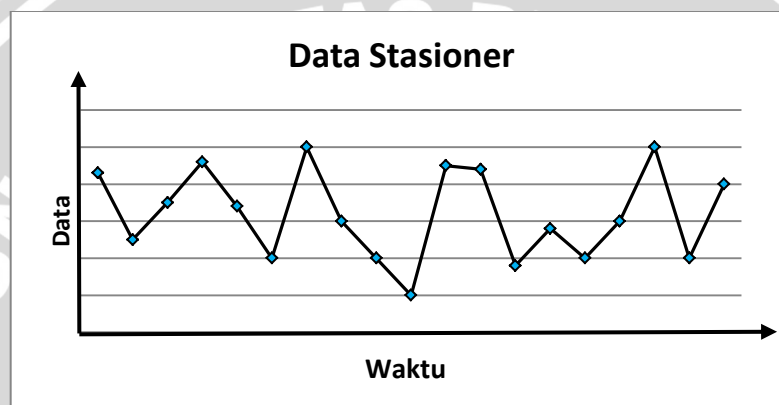
Data merupakan bahan baku yang digunakan dalam proses peramalan. Dalam peramalan data yang paling umum digunakan adalah data historis dari objek. Data historis digunakan karena tidak berubah secara drastis secara tiba-tiba. Data historis dikenal juga sebagai data *time series* yang merupakan deretan hasil yang dicatat secara berurutan dalam jangka waktu. Jangka waktu yang umum digunakan dalam data runut waktu adalah jam, harian, mingguan, bulanan, tahunan dan interval lainnya yang bisa digunakan. Grafik runut waktu disebut juga dengan *time series plot*, dengan menggunakan grafik dapat menentukan pola data runut waktu yang digunakan. (Montgomery, et al., 2008).

### 2.3.1 Pola Data Runut Waktu

Ada beberapa karakteristik atau pola dari data runut waktu seperti, stasioner, musiman, tren dan siklis. Analisis pola data runut waktu diperlukan untuk mempermudah metode peramalan yang tepat (Palit & Popovic, 2006).

#### 1. Pola Horizontal atau stasioner

Pola horizontal terjadi apabila dari waktu ke waktu nilai rata-rata dan variasi data konstan tidak mengalami perubahan. Struktur data stasioner ditunjukkan dengan pola yang datar, tidak memiliki tren atau musiman, variasi yang rendah dan struktur data stasioner ada peningkatan dan penurunan suatu nilai konstan secara konsisten dari waktu ke waktu. Pola data horizontal atau stasioner memiliki korelasi seperti pada Gambar 2.2

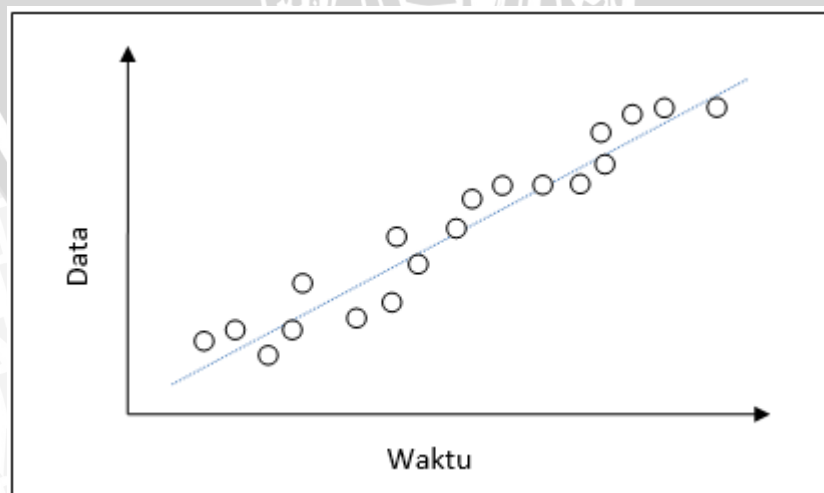


Gambar 2.2 Pola stasioner

Sumber: (Palit & Popovic, 2006)

#### 2. Pola Trend

Pola *Trend* terjadi apabila dalam suatu data terdapat gerakan naik dan turun dalam waktu jangka panjang yang diperoleh dari rata-rata perubahan waktu yang memiliki nilai mulus (*smooth*). Pola data tren digambarkan pada Gambar 2.3

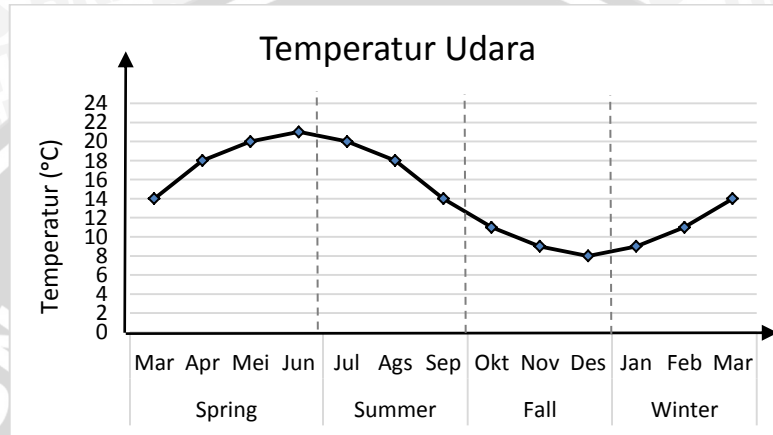


Gambar 2.3 Pola trend

Sumber: (Palit & Popovic, 2006)

3. Pola Musiman

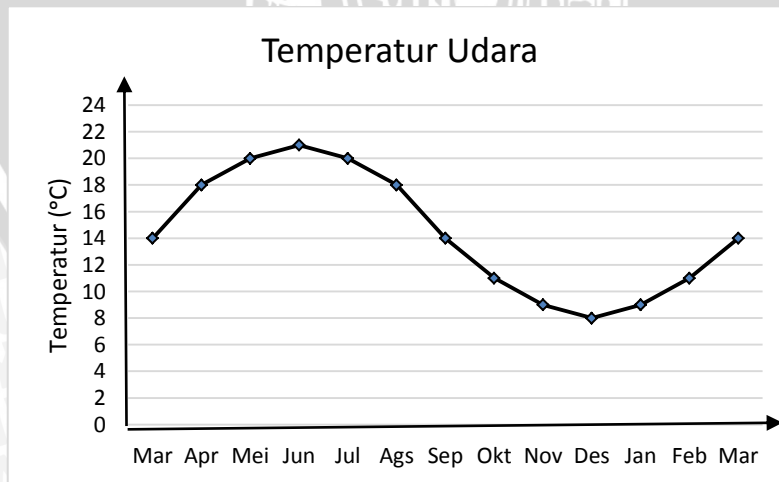
Pola musiman terjadi apabila faktor musiman mempengaruhi suatu nilai data. Misalnya bulanan pada suatu tahun tertentu, atau hari-hari pada minggu tertentu. Pola data musiman mempunyai pola musim yang berulang dari periode ke periode berikutnya. Tujuan utama dari analisis data musiman adalah untuk mendeteksi karakter dari fluktuasi periodik pada grafik seperti pada Gambar 2.4 yang merupakan pola musiman dari data temperatur udara di negara Swedia dan Finlandia.



**Gambar 2.4 Pola musiman**  
 Sumber: (Musyafiroh & Musyafiroh, 2014)

4. Pola Siklis

Pola siklis memiliki pola yang cenderung tidak stabil dimana nilai data terjadi gelombang naik dan turun pada interval waktu yang tetap. Perbedaan dari pola siklis dan musiman adalah pola siklis memiliki panjang periode yang konstan dan berulang secara periodik dengan teratur. Pola siklis ditunjukkan oleh Gambar 2.5 dengan panjang rata-rata periode siklis lebih panjang dari periode musiman dan besarnya siklis lebih bervariasi daripada pola musiman.



**Gambar 2.5 Pola siklis**  
 Sumber: (Musyafiroh & Musyafiroh, 2014)

5. Random



Pola acak data yang disebabkan situasi yang tidak biasa dan tidak mengikuti pola yang jelas. Pola random tidak memiliki periode yang pasti, bahkan terjadi secara acak yang berulang tanpa memperhatikan periode tertentu.

### 2.3.2 Preprocessing Data Runut Waktu

*Preprocessing* data adalah tahap awalan dalam pengolahan data dan mempersiapkan data. Persiapan data yang baik akan menunjang kualitas hasil dari keluaran data yang telah diproses. Data dapat dikatakan berkualitas jika memiliki akurasi data, kelengkapan data, konsistensi data, data terbaharui, tingkat kepercayaan pada data, mudah dipahami. Ada beberapa teknik data *preprocessing*. Untuk meningkatkan akurasi dan efisiensi dari suatu algoritma dapat dilakukan teknik *preprocessing* diantaranya adalah sebagai berikut (Han, et al., 2011).

1. Pembersihan data berguna untuk merapikan dan memperbaiki ketidak konsistenan data.
2. Integrasi data merupakan tahap penggabungan data dari berbagai sumber
3. Reduksi data bertujuan untuk mengurangi ukuran data seperti jumlah data, fitur, dan pengelompokan data.
4. Transformasi data bertujuan untuk mengefisienkan data sehingga pola data dapat mudah untuk dipahami.

Dari beberapa teknik data *preprocessing* yang telah dijelaskan sebelumnya dapat diterapkan pada pengolahan awal data runut waktu sesuai kebutuhan pengguna. Berikut merupakan hal yang perlu diperhatikan dalam pengolahan awal data runut waktu (Montgomery, et al., 2015).

1. Apakah ada data yang hilang?
2. Apakah data melebihi rentang yang diharapkan?
3. Apakah ada potensi *outlier* atau nilai-nilai yang tidak biasa?

Menurut Palit dan Popovic (2006) hal terpenting dalam pengolahan awal data runut waktu meliputi penataan data dan normalisasi data. Penataan data diperlukan untuk membedakan antara data pelatihan dan data pengujian. Sedangkan normalisasi data merupakan salah satu strategi dalam transformasi data yang dapat diterapkan pada ruang *input* pada data runut waktu (Crone, et al., 2006). Normalisasi data merupakan proses penyederhanaan data ke dalam skala kecil yang telah ditentukan. Dalam penentuan batas skala tersebut dapat ditentukan antara -1 sampai 1 atau 0 sampai 1, dan batas skala lainnya. Ada banyak teknik dalam normalisasi data, salah satunya adalah teknik normalisasi *min-max*. Teknik normalisasi *min-max* merupakan transformasi linear dari data asli. Normalisasi *min-max* akan menemukan kesalahan jika nilai masukan berada jauh dari nilai rentang data asli dikarenakan pada teknik ini akan mempertahankan hubungan antara nilai-nilai data asli (Han, et al., 2011). Berikut teknik normalisasi *min-max* ditunjukkan pada Persamaan (2.1)

$$v'_i = \frac{v_i - \min_A}{\max_A - \min_A} (\text{new\_max}_A - \text{new\_min}_A) + \text{new\_min}_A \quad (2.1)$$

Dimana:

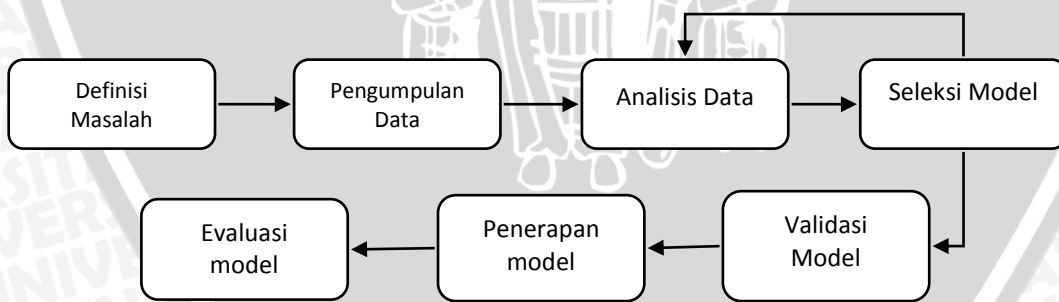
- $v$  : nilai normal
- $A$  : Atribut
- $\min_A$  : nilai minimum pada atribut  $A$
- $\max_A$  : nilai maksimum pada atribut  $A$
- $\text{new\_min}_A$  : nilai minimum baru pada nilai normal
- $\text{new\_max}_A$  : nilai maksimum baru pada nilai normal

Kemudian untuk denormalisasi data dapat menggunakan Persamaan (2.2) berikut

$$v_i = \frac{v'_i - \text{new\_min}_A}{(\text{new\_max}_A - \text{new\_min}_A)} (\max_A - \min_A) + \min_A \quad (2.2)$$

## 2.4 Peramalan

Ramalan adalah sebuah prediksi kejadian dimasa mendatang dan kejadian-kejadian lainnya. Seperti yang dikatakan oleh Neil Bohr membuat prediksi yang bagus tidaklah mudah. Peramalan merupakan permasalahan yang penting yang menjangkau berbagai bidang baik itu bidang bisnis, industri, pemerintahan, ekonomi, lingkungan ilmiah, kesehatan, sosial, politik dan keuangan (Montgomery, et al., 2008). Peramalan sendiri merupakan bagian dari aktivitas manajemen pembuatan keputusan. Kebutuhan peramalan sendiri semakin meningkat dengan adanya usaha untuk mengurangi ketergantungan kesempatan dan terasa menjadi lebih terlatih dengan lingkungannya (Makridakis, et al., 1997). Ada tujuh tahapan proses pada sistem peramalan yang ditunjukkan pada Gambar 2.6.



**Gambar 2.6** Proses peramalan

Sumber: (Montgomery, et al., 2008)

### 1. Definisi masalah

Meningkatkan pemahaman bagaimana ramalan akan digunakan sesuai dengan harapan pengguna sistem ramalan. Tahap definisi masalah sangat menentukan keberhasilan suatu model peramalan. Tahap ini memberikan peluang pada pembuat keputusan tentang interval yang akan digunakan dalam peramalan yang sesuai dengan kebutuhan pengguna.



2. Pengumpulan data  
Pengumpulan data histori atau variabel yang akan diprediksi termasuk dengan informasi historis dari prediksi variabel yang berpotensi. Dalam tahap ini dibutuhkan perencanaan bagaimana kumpulan data dan penyimpanannya ditangani sehingga keandalan dan integritas dari data dapat dipertahankan
3. Analisis data  
Pada data runtut waktu analisis digunakan untuk mengenali pola data yang digunakan seperti data tren, musiman atau siklis. Dengan menentukan pola data dapat memberikan informasi jenis model dan metode peramalan kuantitatif yang dapat digunakan
4. Seleksi dan penyesuaian model  
Memilih satu atau lebih model peramalan dan menyesuaikannya dengan data. Penyesuaian yang dimaksud adalah menghitung parameter model yang belum diketahui sehingga dapat menentukan kandidat model yang sesuai untuk peramalan yang digunakan
5. Validasi model  
Validasi model dapat dilakukan dengan membagi data menjadi dua bagian yaitu bagian untuk dilatih dan bagian untuk diuji. Hal ini akan menentukan bagaimana kinerja dari model peramalan yang digunakan saat data baru ditambahkan dan dapat memberikan hasil yang baik
6. Penerapan model peramalan  
Memastikan pengguna mengerti bagaimana menggunakan model yang meramal setiap waktu. Pada tahap ini juga memastikan sumber data dan informasi lainnya yang dibutuhkan akan selalu tersedia.
7. Evaluasi performa model peramalan  
Memastikan model yang digunakan masih memberikan performa yang baik, salah satu cara untuk memantau performa dari model peramalan adalah dengan membuat grafik kontrol dari nilai *error* yang dihasilkan sistem secara rutin.

Peramalan sendiri terlibat dengan perkiraan dan perhitungan dari kejadian masa depan yang diharapkan atau bentuk perkembangan dari model konseptual. Permasalahan dalam peramalan di klasifikasi menjadi jangka pendek, jangka menengah, dan jangka panjang. Peramalan jangka pendek memprediksi kejadian dalam dalam beberapa periode waktu ke depan seperti hari, minggu dan bulan. Peramalan jangka menengah memprediksi satu atau dua tahun mendatang dan peramalan jangka panjang bertahun-tahun ke depan. Peramalan jangka pendek dan menengah digunakan dalam operasi manajemen untuk pendanaan dan untuk pengembangan proyek, sedangkan peramalan jangka panjang digunakan untuk kasus seperti pembuatan rencana strategi. Ada empat metode peramalan yang telah dikembangkan dan digunakan secara luas yaitu: (Montgomery, et al., 2008) (Land, 2015)

1. Pendekatan indikator utama

Menurut National Bureau of Economic Research (NBER) dan Wesley Mitchell (1927) peramalan dengan menggunakan pendekatan indikator utama paling luas dikembangkan dan diaplikasikan untuk peramalan dibidang ekonomi dan bisnis. Pada pendekatan indikator ekonomi fokus pada menemukan penjelasan, dan digunakannya untuk meramalkan siklus dari bisnis (Land, 2015).

2. Model formal

Model peramalan formal berdasarkan perhitungan matematika dan statistik dengan perhitungan parameter secara empiris (Land, 2015).

3. Analisis secara kuantitatif

Metode kuantitatif merupakan metode yang memerlukan informasi dari masa lalu yang dikuantitatifkan dalam bentuk numerik. Ada dua jenis analisis kuantitatif yaitu (Makridakis, et al., 1997)

- *Time series*, yaitu metode yang digunakan untuk analisis data berupa deret waktu
- *Explanatory*, yaitu metode yang digunakan untuk menganalisa pola hubungan sebab-akibat

4. Analisis secara kualitatif

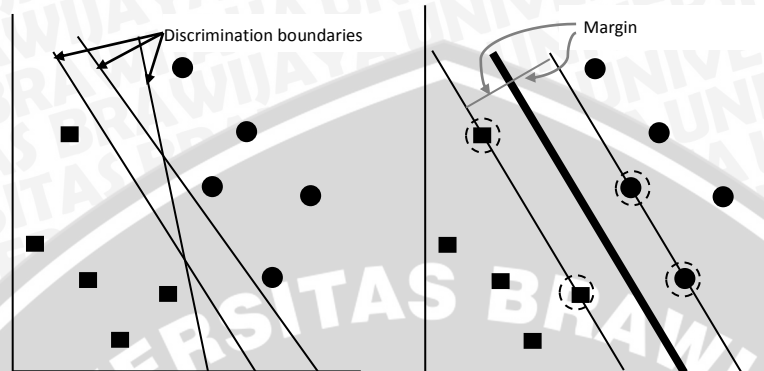
Pada metode peramalan dengan pendekatan analisis secara kualitatif ini digunakan pada peramalan fitur umum yang biasa digunakan di masyarakat dalam bentuk analisis teori dan skenario (Land, 2015). Pada metode kualitatif informasi kuantitatif yang didapat hanya sedikit, bahkan tidak ada namun pengetahuan kualitatif ada. Metode kualitatif didasarkan pada pemikiran yang intuitif, perkiraan yang logis, dan informasi yang telah diperoleh sebelumnya (Makridakis, et al., 1997).

Pada peramalan agar menghasilkan ramalan yang memiliki obyektifitas yang tinggi maka diperlukan sebuah teknik peramalan. Teknik peramalan merupakan cara memperkirakan secara sistematis dan pragmatis atas data yang relevan dimasa lalu. Dengan menggunakan teknik peramalan yang baik diharapkan dapat membuat suatu peramalan yang memberikan cara pengerjaan teratur dan terarah sehingga dapat memberikan analisa yang lebih baik dan meningkatkan tingkat kepercayaan karena dapat diuji dan dibuktikan secara ilmiah (Makridakis, et al., 1997).

## 2.5 Support Vector Regression (SVR)

SVR merupakan bagian atau cabang dari SVM. SVR adalah penerapan dari SVM dalam dalam menyelesaikan kasus regresi. Menurut (Santosa, 2007 ) dalam kasus klasifikasi pada SVM output data dalam bilangan bulat atau diskrit, sedangkan untuk SVR output data berupa bilangan riil atau kontinyu. Dengan menggunakan konsep  *$\epsilon$ -insentive loss function*, yang dikemukakan oleh Vapnik SVM dapat digeneralisasi untuk melakukan regresi. Konsep dari SVM merupakan cara untuk mencari *hyperlane* terbaik yang berfungsi sebagai pemisah pada dua kelas input. *Hyperlane* terbaik dapat diukur dengan menggunakan *margin* dari

*hyperlane* tersebut dan mencari titik maksimal dari titik *margin* tersebut. *Margin* merupakan jarak antara *hyperplane* dengan data terdekat dari masing-masing kelas. Data yang terdekat dari *margin* disebut dengan *super vector*. Pada Gambar 2.7 memperlihatkan beberapa data yang menggunakan data dari dua kelas dengan Kelas-1 disimbolkan persegi dan Kelas+1 disimbolkan dengan lingkaran.



**Gambar 2.7 Hyperplane**

Sumber: (Furi, et al., 2015)

Pada Gambar 2.7 sebelah kiri memperlihatkan garis-garis alternatif sebagai pemisah antara dua kelas. Sedangkan pada gambar sebelah kanan terdapat garis tebal yang menunjukkan *hyperplane* terbaik. Titik-titik pada lingkaran bergaris putus adalah *suport vector*. SVR yang merupakan adaptasi dari SVM memiliki kemampuan untuk mengatasi permasalahan *overfitting*, sehingga dapat menghasilkan fungsi dengan tingkat kesalahan yang lebih kecil sehingga dapat menghasilkan peramalan yang bagus. Pada SVR permasalahan regresi non-linier ruang input berdimensi rendah ditransformasikan ke dalam ruang fitur berdimensi tinggi (Furi, et al., 2015).

Misalkan jika ada  $l$  set data training,  $(x_i, y_i), i = 1, \dots, l$  dengan data input  $x = (x_1, x_2, \dots, x_l)$  dan output  $y = \{y_1, \dots, y_l\} \subseteq \mathbb{R}$ . Untuk menemukan fungsi  $f(x)$  dengan standar deviasi yang paling besar  $\epsilon$  dan target aktual  $y_i$  pada semua data latih dapat menggunakan persamaan untuk regresi linier dimana nilai  $\epsilon$  adalah 0 yang ditunjukkan oleh Persamaan (2.3) (Wirawan, 2011)

$$f(x) = w^T \varphi(x) + b \tag{2.3}$$

Dimana :

$\varphi(x)$  = fungsi yang memetakan  $x$  dalam suatu dimensi

$w^T$  = vektor pembobot

$b$  = bias

$x$  =vektor input

Agar mendapat generalisasi yang baik untuk fungsi regresi  $f(x)$  dapat dilakukan dengan meminimalkan norm dari  $w$  dan  $b$ . Estimasi koefisien dengan meminimalkan fungsi resiko dilakukan dengan Persamaan (2.4)

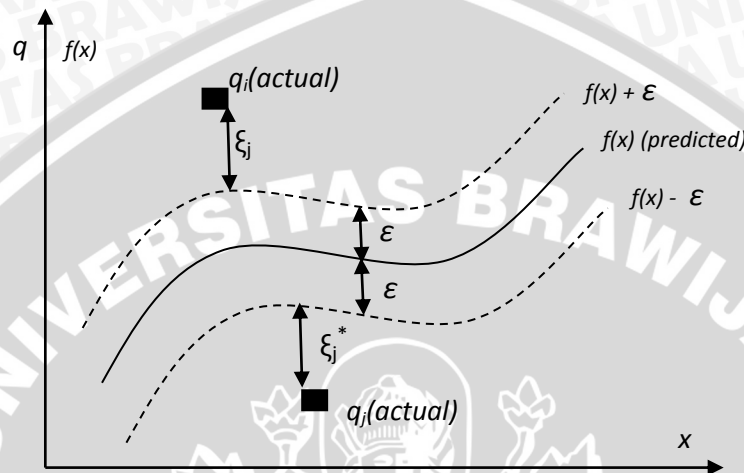


$$R(C) = \frac{1}{2} \|w\|^2 + C \frac{1}{l} \sum_{i=1}^l L_e(y_i, f(x_i)q_i) \quad (2.4)$$

Dimana:

$L_e(y_i, f(x_i)q_i) = \epsilon$ -insentive loss function

$\frac{1}{2} \|w\|^2$  = ketentuan regularisasi yang mengontrol kompleksitas fungsi



**Gambar 2.8**  $\epsilon$ -insentive loss function pada SVR

Sumber: (Lu, et al., 2009)

$C$  = kontanta yang mengatur regularisasi

$\epsilon$ -insentive loss function merupakan parameter yang merepresentasikan radius dari tube yang terletak di sekitar fungsi regresi ( $f(x)$ ). Nilai  $\epsilon$  mengatur batas error dari data set pada kelasnya masing-masing. Nilai  $C$  menjadi nilai penalti dataset yang memiliki nilai error lebih besar dari  $\epsilon$ . Setiap kesalahan yang nilainya lebih besar dari  $\epsilon$  maka akan dikenakan penalti sebesar  $C$ . semakin kecil nilai  $C$  maka semakin toleran sistem terhadap kesalahan. Pada Gambar 2.2 terdapat slack variable ( $\xi$ ),  $\xi_i$  dan  $\xi_i^*$ , dengan  $i=1, \dots, n$  yang digunakan untuk mengukur deviasi dari batas  $\epsilon$ -insentive, yang merepresentasikan jarak antara data dan nilai batas zona  $\epsilon$ -insentive. Dengan menggunakan slack variable Persamaan (2.4) ditransformasikan menjadi

$$R_{reg}(f) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \quad (2.5)$$

Dengan syarat

$$q_i - (w\phi(x_i)) - b \leq \epsilon + \xi_i$$

$$(w\phi(x_i)) + b - q_i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* = 0 \text{ untuk } i = 1, \dots, n$$

Dimana

$R_{reg}(f)$  = Transformasi estimasi koefisien

$\|w\|^2$  = ketentuan regularisasi yang mengontrol kompleksitas fungsi

$C$  = nilai kompleksitas

$\xi_i, \xi_i^*$  = *slack variable*, jarak antara nilai asli dan batasan terdekat dari  $\varepsilon$ -tube

Untuk mempermudah dalam menyelesaikan permasalahan optimasi konveks pada Persamaan (2.4), persamaan tersebut bisa ditransformasikan ke dalam *dual space* dengan menerapkan Lagrangian *Multipliers* dan kondisi Karush-Kuhn-Tucker (KKT) (Kavousi-Fard, et al., 2014). Kondisi KKT bertujuan untuk mengatur nilai bias dalam membantu *Lagrange* memaksimalkan nilai agar tidak melebihi batas  $\varepsilon$  (Smola & Schölkopf, 2004). Berikut merupakan formulasi untuk optimasi *dual problem* dengan menggunakan *quadratic programming* yang ditunjukkan pada Persamaan (2.6).

$$\text{Maks } \left\{ \begin{array}{l} -\frac{1}{2} \sum_{i,j=0}^{\ell} (a_i - a_i^*)(a_j - a_j^*) \langle x_i \bullet x_j \rangle \\ -\varepsilon \sum_{i=1}^{\ell} (a_i + a_i^*) + \sum_{i=1}^{\ell} y_i (a_i - a_i^*) \end{array} \right\} \quad (2.6)$$

Dengan syarat  $\sum_{i=1}^{\ell} (a_i - a_i^*) = 0$  dan  $a_i, a_i^* \in [0, C]$

Dimana

$\alpha_i^*, \alpha_i$  = Nilai *Lagrange Multiplier*

$x_i$  = Nilai dari fitur data yang akan digunakan untuk menentukan peramalan

$x_j$  = Nilai dari fitur data pelatihan ke- $j$

$\varepsilon$  = Nilai kerugian

$C$  = Nilai Kompleksitas

Dari Persamaan (2.6) diperoleh persamaan regresi non linier yang ditunjukkan pada Persamaan (2.7)

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) (x \bullet x_i) + b \quad (2.7)$$

Dimana

$\alpha_i^*, \alpha_i$  = Nilai *Lagrange Multiplier*

$x$  = Nilai dari fitur data yang akan digunakan untuk menentukan peramalan

$x_i$  = Nilai dari fitur data pelatihan ke- $i$

$b$  = Nilai bias

$C$  = Nilai Kompleksitas

Dengan menggunakan pemodelan SVR, antara variabel *input* dan data *output* menjadi tidak linier. Sehingga dikembangkan metode SVR non linier dengan memetakan variabel *input* ke dalam ruang fitur berdimensi tinggi  $F$  yang direpresentasikan menjadi  $\varphi: R^d \rightarrow F$  sehingga membentuk fungsi regresi non linier yang ditunjukkan pada Persamaan (2.8) (Furi, et al., 2015).

$$f(x) = \sum_{i=1}^{\lambda} (\alpha_i^* - \alpha_i)(\varphi(x) \bullet \varphi(x_i)) + b \quad (2.8)$$

Dimana

$\varphi(x)$  = fungsi yang memetakan dalam suatu dimensi

$\alpha_i^*, \alpha_i$  = Nilai *Lagrange Multiplier*

$b$  = Nilai bias

Untuk membantu mengatasi permasalahan dalam memetakan data non linier, maka dapat diselesaikan dengan fungsi *kernel* yang sesuai dengan *dot product* untuk vektor inputan di ruang fitur berdimensi tinggi (Basak & Patranabis, 2007). Sehingga untuk menyederhanakan fungsi regresi pada Persamaan (2.8) maka fungsi untuk SVR regresi non linier ditunjukkan pada Persamaan (2.9)

$$F(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) - K(x_i x_j) + b \quad (2.9)$$

Dimana

$\alpha_i^*, \alpha_i$  = Nilai *Lagrange Multiplier*

$K(x_i, x)$  = fungsi kernel.

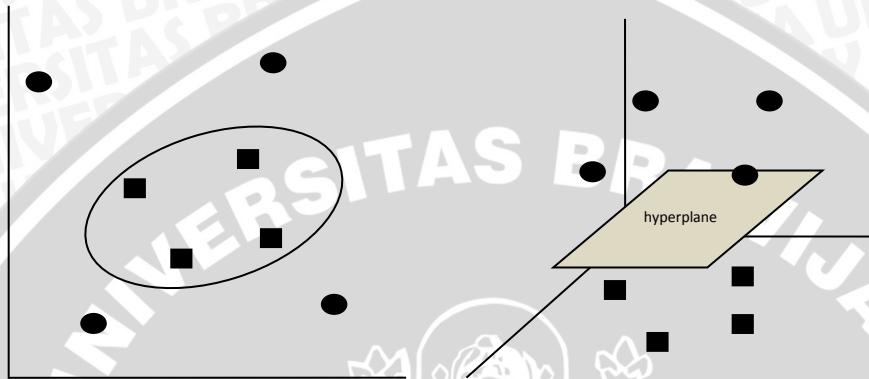
Berdasarkan persamaan (2.8) SVR regresi non linier, Vijayakumar dan Wu (1999) memberikan alternatif dengan memodifikasi formulasi yang bertujuan untuk mendapatkan proses implementasi pada *support vectors* yang lebih sederhana dan cepat. Modifikasi formulasi dilakukan pada nilai bias  $b$  dengan menambahkan dimensi pada pola vektor seperti  $x' = (x_1, \dots, x_N, \lambda)$  dan menghubungkan nilai  $b$  pada vektor bobot  $w' = (w_1, \dots, w_N, b/\lambda)$  dimana  $\lambda$  adalah variabel skalar. Sehingga didapatkan alternatif dengan cara mengganti nilai dari minimum  $\|w\|^2$  dengan meminimalkan  $\|w'\|^2 = \|w\|^2 + b^2/\lambda^2$ . Oleh karena itu, variabel  $b$  pada persamaan (2.9) diganti dengan variabel skalar  $\lambda$  yang ditunjukkan pada persamaan (2.10)

$$F(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) - K(x_i x_j) + \lambda^2 \quad (2.10)$$

Penggunaan alternatif dengan mengganti variabel  $b$  dengan variabel skalar  $\lambda$  pada metode SVR dapat memberikan konseptual yang sederhana dan kemudahan dalam penerapan metode SVR dengan menggunakan algoritma *sequential learning* (Vijayakumar & Wu, 1999).

### 2.5.1 Fungsi Kernel

Pada *machine learning* terdapat istilah metode *kernel* fungsi kernel digunakan dalam menyelesaikan permasalahan non-linear. Untuk menjaga topologi dari data maka pemetaan tersebut dilakukan, yang berada dua data yang berjarak dekat pada input space berjarak dekat juga pada *feature space* begitu juga sebaliknya. Untuk memecahkan masalah linear dalam ruang berdimensi tinggi, dapat dilakukan dengan mengganti *linear product* ( $x_1$  dan  $x_2$ ) dengan fungsi kernel yang sesuai (Furi, et al., 2015).



**Gambar 2.9 Penggunaan kernel**

Sumber: (Furi, et al., 2015)

Pada Gambar 2.9 di sisi kiri merupakan representasi data pada ruang dimensi dua yang tidak bisa dipisahkan secara linear. Pada sisi kanan terlihat fungsi  $\phi$  yang memetakan data kedalam ruang fitur berdimensi baru dengan dimensi lebih tinggi yaitu 3, dimana kedua kelas dapat dipisahkan secara linear oleh sebuah *hyperplane* (Furi, et al., 2015). Contoh metode kernel yang digunakan dengan dalam penelitian ini adalah sebagai berikut:

1. *Linear kernel*

Merupakan *kernel* dasar yang paling sederhana jika dibandingkan dengan fungsi *kernel* lainnya (Wirawan, 2011). Fungsi *Linear kernel* hanya melakukan pemetaan  $\phi$  secara linier seperti pada Persamaan (2.11).

$$k(x, x') = (x \cdot x') = (\phi(x) \cdot \phi(x')) \quad (2.11)$$

2. *Gaussian kernel / Radial Basis Function (RBF)*

RBF mempunyai ciri dimana dalam perumusan fungsinya menggunakan *squared exponential* (Williams & Rasmussen, 2006). Gaussian RBF *kernel* merupakan *kernel* yang populer untuk diterapkan pada metode SVM. Namun diperlukan penyesuaian terhadap lebarnya parameter  $\sigma$  yang akan digunakan (Schölkopf & Smola, 2002). Fungsi dari *gaussian kernel* seperti pada persamaan (2.12)

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (2.12)$$

3. *Anova Kernel*

ANOVA atau *analysis of variance* sebagai teknik statistik untuk menganalisa interaksi berbagai macam atribut. Kernel *Anova Radial Basis kernel* dapat menyelesaikan permasalahan non linear yang dapat memecahkan masalah linear menjadi ruang dimensi tinggi yaitu dengan mengganti *inner product* ( $x_i$  dan  $x_j$ ) untuk menggambarkan *dot product* di ruang fitur (Rasmussen & Williams, 2006). Fungsi ANOVA RBF kernel seperti pada Persamaan (2.13).

$$K(x_i, x_j) = \left( \sum_{i,j=1}^N e^{(-\gamma(x_i - x_j)^2)} \right)^d \quad (2.13)$$

Dimana terdapat parameter  $d > 0$  dan  $\gamma \geq 0$  yang mempengaruhi kinerja ANOVA RBF *kernel*. Ketika parameter  $d$  bernilai 1 maka memiliki dalam komputasi bersifat linier dan berpangkat 1. Sedangkan ketika parameter  $d$  bernilai 2 terdapat dua interaksi yang berpangkat kuadrat. Dan dengan alasan yang sama, ketika parameter  $d$  bernilai 3 memiliki tiga interaksi kesamaan kubik (Berk, et al., 2014).

### 2.5.2 Metode Sequential SVR

Pada proses pelatihan metode *Support Vector Machine* (SVM) untuk menangani kasus klasifikasi dan regresi digunakan *Quadratic Programming* (QP). *Quadratic Programming* (QP) digunakan untuk mendapatkan *hyperplane* yang optimum (Bertsimas & Shioda, 2007). Namun *Quadratic Programming* memiliki kelemahan yaitu proses perhitungan yang kompleks sehingga memakan waktu yang banyak dan memungkinkan terjadinya ketidakstabilan secara numerik. Oleh karena itu, Vijayakumar dan Wu (1999) memberikan alternatif untuk menyelesaikan permasalahan *Quadratic Programming* yaitu dengan menggunakan Algoritma sekuensial pada pelatihan metode SVM dalam menangani kasus klasifikasi dan regresi (Vijayakumar & Wu, 1999).

Algoritma sekuensial pada pelatihan SVM dalam menangani kasus regresi yang lebih dikenal dengan *Support Vector Regression* (SVR), merupakan metode yang memberikan solusi yang optimal, mempercepat proses iterasi serta memberikan urutan yang lebih cepat dibandingkan menggunakan solusi konvensional (Vijayakumar & Wu, 1999). Langkah-langkah penggunaan metode sekuensial adalah sebagai berikut

1. Langkah pertama adalah melakukan inisialisasi terhadap parameter yang digunakan, yaitu  $C, cLR, \varepsilon, \lambda, d, \gamma_k$  dan iterasi maksimum.
2. Langkah berikutnya adalah inisialisasi  $\alpha_i^*$  dan  $\alpha_i$  (diberikan nilai 0 untuk inisialisasi awal keduanya)
3. Menghitung Matrik Hessian dengan Persamaan (2.14)

$$R_{ij} = (K(x_i, x) + \lambda^2) \text{ untuk } ij = 1, \dots, l \quad (2.14)$$

Keterangan:

$R_{ij}$  = Matriks Hessian

$\lambda^2$  = Variabel Skalar



$l$  = Banyaknya data  
 $K(x_i, x)$  = Kernel

4. Untuk setiap *training point*, lakukan langkah (a), (b) dan (c) dari 1 hingga  $l$   
 Langkah 1. Menghitung nilai *error* dengan Persamaan (2.15)

$$E_i = y_i - \sum_{j=1}^l (\alpha_j^* - \alpha_j) R_{ij} \quad (2.15)$$

Keterangan:

$E_i$  = Nilai *error* ke- $i$   
 $y_i$  = data aktual

- Langkah 2. Menghitung delta alpha dan delta alpha star dengan Persamaan (2.16) dan (2.17)

$$\delta \alpha_i^* = \min \{ \max [\gamma(E_i - \varepsilon), -\alpha_i^*], C - \alpha_i^* \} \quad (2.16)$$

$$\delta \alpha_i = \min \{ \max [\gamma(-E_i - \varepsilon), -\alpha_i], C - \alpha_i \} \quad (2.17)$$

Keterangan:

$\delta \alpha_i^*, \delta \alpha_i$  = variabel tunggal, bukan bentuk dari perkalian  $\delta$  dan  $\alpha_i$  atau perkalian  $\delta$  dan  $\alpha_i^*$

$\varepsilon$  = Nilai Epsilon

$C$  = Nilai kompleksitas

$\gamma$  = Nilai *Learning Rate* yang didapatkan dari  $\frac{cLR}{\text{maks}(\text{diagonal Matrik Hessian})}$

- Langkah 3. Menghitung alpha dan apha star dengan Persamaan (2.18) dan (2.19)

$$\alpha_i = \alpha_i + \delta \alpha_i \quad (2.18)$$

$$\alpha_i^* = \alpha_i^* + \delta \alpha_i^* \quad (2.19)$$

5. Kembali pada langkah ketiga, sampai pada kondisi iterasi maksimum atau  $\max(|\delta \alpha_i|) < \varepsilon$  dan  $\max(|\delta \alpha_i^*|) < \varepsilon$

6. Disebut *support vector* jika suatu data memenuhi persyaratan  $(\alpha_i^* - \alpha_i) \neq 0$ , hal ini dapat diartikan bahwa kedua variabel tersebut memiliki nilai yang berbeda.

7. Menggunakan fungsi regresi yang dijabarkan pada Persamaan (2.20)

$$f(x) = \sum_{i=1}^l (\alpha_i^* - \alpha_i) (K(x_i, x) + \lambda^2) \quad (2.20)$$

Keterangan:

$f(x)$  = fungsi regresi

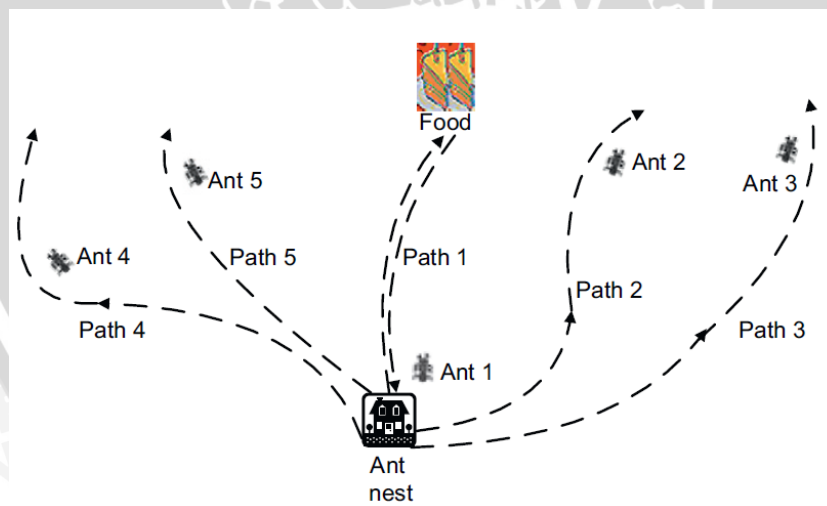
$\alpha_i, \alpha_i'$  = *Lagrange multiplier*

$K(x_i, x)$  = kernel

8. Selesai

## 2.6 Ant Colony Optimization (ANT CO)

ANT CO diperkenalkan pertama kali oleh Dorigo dan Stutzle pada awal tahun 1990 yang berdasarkan perilaku dari koloni semut. ANT CO adalah strategi optimasi yang bersifat *stochastic*. Secara umum ANT CO adalah populasi agen semut secara independen yang bergerak dalam lingkungannya yang merepresentasikan ruang solusi terhadap permasalahan. Komunikasi tak langsung dari semut dalam suatu koloni menggunakan jejak *pheromone* pada permukaan tanah untuk mencari jalur terpendek antara sumber makanan dan sarang. Semut lain merasakan adanya *pheromone* dan sangat tertarik untuk menandai jalur tersebut dengan menambahkan *pheromone* ke dalam jalur. *Pheromone* merupakan zat yang mudah menguap seiring berjalannya waktu. Pada jalur terpanjang *pheromone* lebih cepat menguap dan *pheromone* pada jalur pendek ditambahkan dengan sangat cepat (Ariyasingha & Fernando, 2015) (Boryczka & Kozak, 2015) (Nia, et al., 2014). Pada Gambar 2.10 menunjukkan proses bagaimana semut menemukan jalur terpendek antara sarang dan sumber makanan. Pertama, setiap semut meninggalkan sarangnya untuk mencari sumber makanan secara acak. Saat semut berkeliling, semut akan meninggalkan jejak *pheromone* yang dapat dideteksi oleh semut lainnya. Ketika menemukan sumber makanan semut akan mengambil makanan dan kembali ke sarang dengan mengikuti jejak *pheromone* sebelumnya dan menambahkan *pheromone* pada jalur yang sama. Grup kedua semut pergi meninggalkan sarang untuk mencari makanan, semut dari grup kedua dapat mendeteksi bahwa *pheromone* pada *Path 1* lebih banyak daripada *Path* lainnya. Dengan ini semut dapat menentukan jalur terpendek dari sarang ke sumber makanan (Hong, et al., 2011).



**Gambar 2.10 Metode ANT CO**

Sumber: (Hong, et al., 2011)

### 2.6.1 Karakteristik

Karakteristik dari metode ANT CO adalah sebagai berikut (Putra & Fatichah, n.d.)

1. Menggunakan interaksi agen semut dimana masing-masing semut hanya mampu melakukan tugas sederhana untuk menghasilkan solusi.

2. Menggunakan informasi yang diperoleh dari iterasi sebelumnya yaitu *pheromone* untuk menentukan hasil selanjutnya
3. Penambahan *pheromone* merupakan mekanisme *feedback* pada suatu node dan mekanisme *pheromone evaporation* untuk mengurangi *pheromone*.

### 2.6.2 Algoritma ANT CO

Langkah-langkah algoritma semut adalah sebagai berikut (Dorigo, et al., 2006) (Putra & Fatichah, n.d.)

1. Inisialisasi awal sistem ANT CO  
Menentukan posisi dan jumlah semut dan melakukan inisialisasi nilai awal atribut.
2. Konstruksi solusi semut  
Setiap semut diletakkan pada setiap *vertex* secara acak, selanjutnya semut secara bergantian berjalan menelusuri *vertex* yang terhubung kepadanya dan membentuk *feature subset*.
3. Menghitung nilai *pheromone* setiap semut  
Semut terbaik ditentukan dengan semut yang memiliki *pheromone* terbanyak. Pada iterasi pertama, semut terbaik adalah semut dengan *pheromone* terbanyak pada iterasi tersebut sedangkan iterasi selanjutnya semut terbaik adalah semut dengan *pheromone* terbanyak pada iterasi tersebut dibandingkan dengan semut terbaik sebelumnya. *Feature subset* terbaik adalah *feature subset* yang dihasilkan oleh semut terbaik.
4. Memeriksa jumlah iterasi  
Apabila jumlah iterasi telah mencapai jumlah iterasi yang telah didefinisikan sebelumnya maka aplikasi dihentikan dan *feature subset* terbaik yang diperoleh pada langkah 4 dihasilkan sebagai *output* apabila tidak maka dilanjutkan ke langkah 6.
5. Perubahan *pheromone*  
Setelah semua semut menghasilkan *feature subset* dan *pheromone*-nya masing-masing maka setiap semut meletakkan *pheromone* pada semua *vertex* yang dilaluinya. Perubahan *pheromone* tersebut dinamakan perubahan *pheromone* lokal. Sedangkan pembaruan *pheromone* pada *feature subset* terbaik dinamakan perubahan *pheromone global*.
6. *Generate* populasi baru  
Semua semut di *vertex* dihapus kemudian melakukan proses *generate* populasi semut baru sesuai dengan inisialisasi awal dan kembali ke langkah 3.

### 2.7 Evaluasi kinerja

Dalam pengukuran hasil peramalan, tidak ada peramalan yang memiliki tingkat akurasi sebesar 100%, sehingga untuk mengukur sebuah metode peramalan adalah dengan menggunakan tingkat kesalahan yang terjadi. Semakin kecil tingkat kesalahan maka semakin baik metode peramalan tersebut (Montgomery, et al., 2008).

1. Rata-rata kuadrat kesalahan (*Mean Square Error* = MSE)

MSE dihitung dengan menjumlahkan kuadrat semua kesalahan peramalan pada setiap periode dan membaginya dengan jumlah periode peramalan (Hyndman & Koehler, 2006). Secara sistematis MSE dirumuskan pada Persamaan (2.26)

$$MSE = \frac{\sum_{k=0}^n (A_t - F_t)^2}{n} \tag{2.21}$$

Keterangan:

$A_t$  = permintaan aktual

$F_t$  = permintaan hasil peramalan

$n$  = jumlah data

2. Rata-rata persentase kesalahan absolute (*Mean Absolute Percentage Error* = MAPE)

MAPE adalah perhitungan dengan perbandingan persentase perbedaan nilai rata-rata *absolute* antara nilai peramalan dengan nilai yang sebenarnya terjadi yang dinyatakan dengan Persamaan (2.27) (Hutasuhut, et al., 2014).

$$MAPE = \left( \frac{1}{n} \right) \sum_{i=1}^n \left| \frac{F_t - A_t}{A_t} \right| \tag{2.22}$$

Keterangan:

$A_t$  = permintaan aktual

$F_t$  = permintaan hasil peramalan

$n$  = jumlah data

Hasil nilai persentase yang diperoleh menunjukkan kemampuan model peramalan dengan nilai yang semakin rendah menunjukkan kemampuan model peramalan yang semakin baik. Kemampuan model peramalan menggunakan MAPE dikategorikan pada Tabel 2.2 berikut

**Tabel 2.2 Signifikasnsi MAPE**

MAPE	Signifikan
<10%	Kemampuan peramalan sangat baik
10-20%	Kemampuan peramalan baik
20-50%	Kemampuan peramalan layak
>50%	Kemampuan peramalan buruk

Sumber : (Chang, et al., 2007)

3. *Mean Absolute Error* (MAE)

Dalam membandingkan metode peramalan pada satu jenis data yang sama, MAE lebih populer karena mudah untuk dipahami dalam penghitungan (Hyndman & Athanasopoulos, 2014).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_t - \hat{y}_t| \tag{2.23}$$



Mean Square Error (MSE) pada Persamaan (2.28) merupakan jenis pengukuran nilai *error* berdasarkan rata-rata yang dikuadratkan.

#### 4. Root Mean Absolute Error (RMSE)

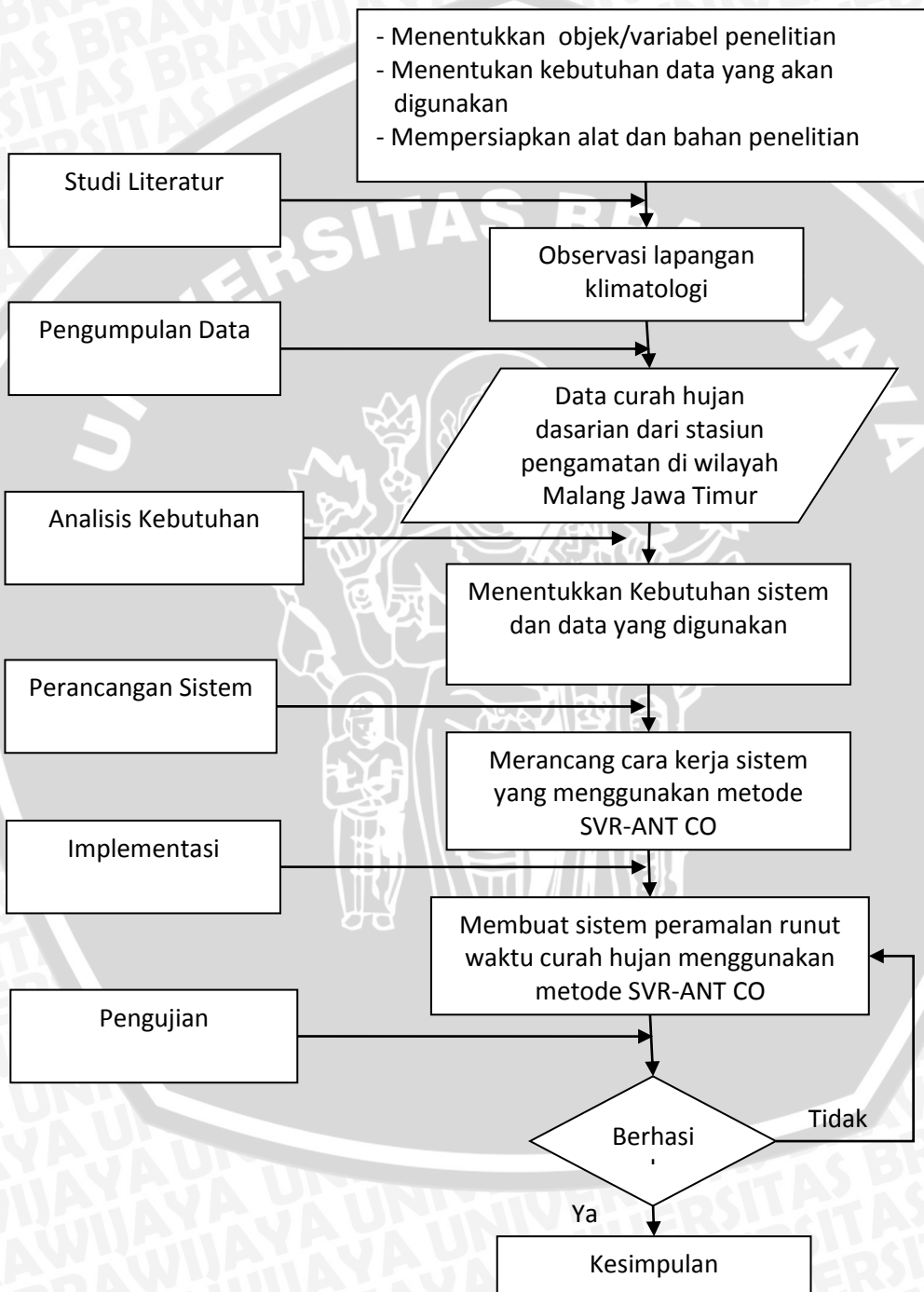
Perumusan RMSE dan MSE sangat populer digunakan karena termasuk dalam teori statistika. RMSE lebih sering digunakan daripada MSE pada data dengan skala yang sama. Metode RMSE juga sensitif terhadap penyimpangan data atau *outlier* selain itu RMSE memiliki sifat yang bervariasi saat melakukan evaluasi peramalan di waktu yang berbeda, hal ini karena sifat nonlinear terdapat pada pola dataset yang digunakan (Hyndman & Koehler, 2006). Metode RMSE memiliki hasil evaluasi yang baik, sehingga dapat ditafsirkan sebagai standar evaluasi kinerja peramalan (Levenbach, 2015). Pengakaran terhadap MSE menjadi *Root Mean Square Error* (RMSE) yang ditunjukkan pada Persamaan (29)

$$RMSE = \sqrt{MSE} \quad (2.24)$$



### BAB 3 METODOLOGI

Metodologi penelitian menjelaskan metode yang digunakan dalam pembuatan peramalan curah hujan dengan menggunakan metode *support vector regression* (SVR) dan *ant colony optimization* (ANT CO). Tahapan metodologi penelitian dapat dilihat sebagai berikut



Gambar 3.1 Diagram blok metodologi penelitian



### 3.1 Studi literatur

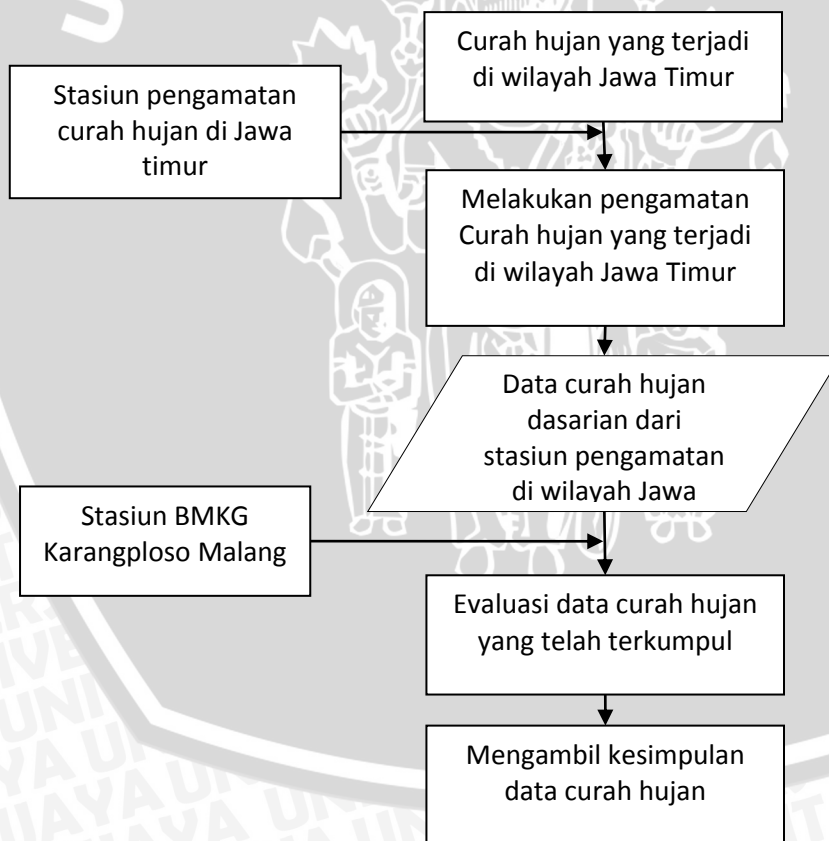
Mempelajari literatur dari beberapa bidang ilmu yang berhubungan dengan pembuatan peramalan curah hujan, metode *Support Vector Regression* (SVR) dan *Ant Colony optimization* (ANT CO), diantaranya:

1. Peramalan
2. Curah hujan
3. Metode SVR
4. Metode ANT CO
5. Rekayasa Perangkat Lunak

Literatur tersebut didapatkan dari buku, jurnal, e-book, penelitian sebelumnya dan dokumentasi proyek

### 3.2 Pengumpulan data

Pengumpulan data dilakukan di Badan Meteorologi Klimatologi dan Geofisika di daerah Karangploso, Malang. Data pada penelitian ini menggunakan data runtu waktu curah hujan harian. Data-data tersebut akan digunakan untuk menghitung tingkat akurasi sistem yang akan dibangun.



Gambar 3.2 Pengumpulan data curah hujan

### 3.3 Analisa kebutuhan

Analisa kebutuhan dilakukan dengan menentukan kebutuhan apa saja yang dibutuhkan untuk membangun sistem peramalan cuaca. Berikut ini kebutuhan



yang digunakan dalam pembuatan peramalan curah hujan dengan menggunakan metode *Support Vector Regression* (SVR) dan *Ant Colony Optimization* (ANT CO):

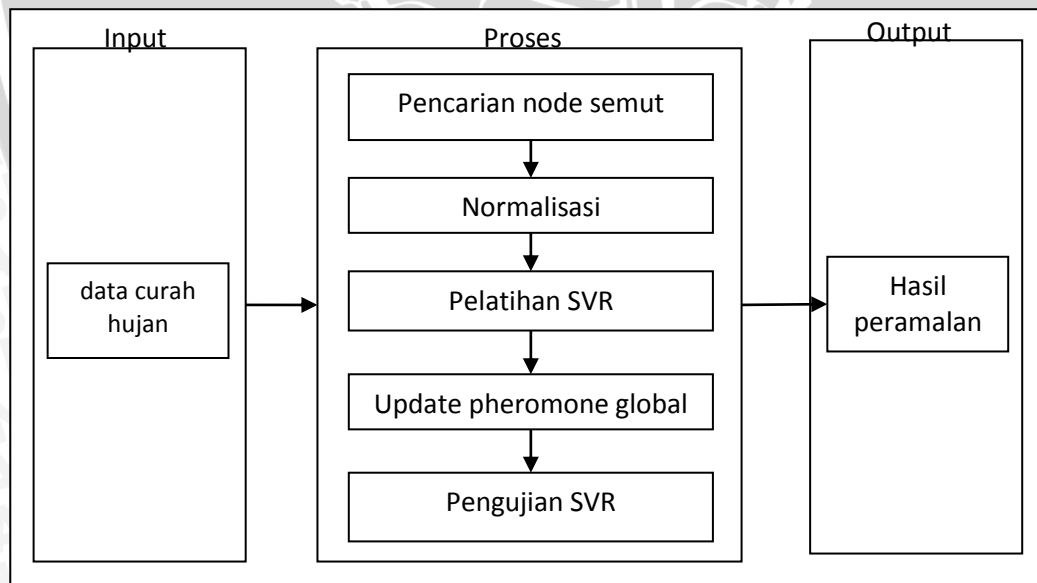
1. Kebutuhan perangkat keras, meliputi:
  - Prosesor : Intel® Core™2 Duo CPU T6600 2.20 GHz
  - RAM : 2.00 GB
  - VGA : Mobile Intel(R) 4 Series Express Chipset Family
  - Storage : 298.09 GB
2. Kebutuhan perangkat lunak, meliputi:
  - Sistem operasi : Windows 7 Ultimate
  - Editor Pemrograman : Java Netbeans 8.0
3. Data yang dibutuhkan, meliputi:
  - Data dasarian runut waktu curah hujan wilayah Poncokusumo Malang dengan rentang tahun 2000-2015

### 3.4 Perancangan Sistem

Perancangan sistem dibuat untuk rancangan kerja dari sistem secara menyeluruh, lebih baik dari segi model maupun segi arsitektur untuk mempermudah implementasi dan pengujian. Langkah kerja dalam sistem disesuaikan dengan arsitektur dari peramalan.

#### 3.4.1 Model perancangan sistem

Model perancangan sistem menjelaskan mengenai cara kerja sistem secara terstruktur mulai dari input yang dimasukkan hingga mendapatkan hasil. Diagram



model perancangan sistem dapat dilihat pada Gambar 3.3

Pada Gambar 3.2 terdiri atas tiga proses utama, yaitu:

1. Input



Input pada sistem peramalan ini adalah data curah hujan yang diperoleh dari Badan Meteorologi Klimatologi dan Geofisika (BMKG) di Karangploso, Malang.

2. Proses

Proses perhitungan pada penelitian ini menggunakan metode *Support Vector Regression* (SVR) yang dioptimasi dengan menggunakan metode *Ant Colony Optimization* (ANT CO). Langkah-langkah perhitungan dengan menggunakan metode *Support Vector Regression-Ant Colony Optimization* (SVR-ANT CO):

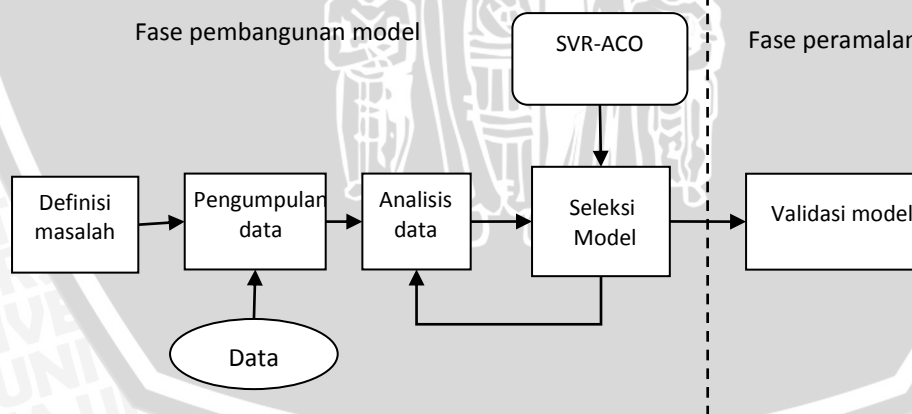
- a. Pencarian node: menentukan nilai parameter SVR dengan menggunakan metode optimasi ANT CO.
- b. Normalisasi: proses preprocessing data dengan menggunakan normalisasi sesuai dengan skala yang digunakan.
- c. Pelatihan SVR: melakukan proses pelatihan SVR dengan menggunakan nilai parameter dari metode ANT CO.
- d. Update pheromone global: proses memperbarui nilai pheromon pada semut.
- e. Pengujian SVR: melakukan proses peramalan dengan menggunakan parameter terbaik dari proses pelatihan SVR.

3. Output

Hasil peramalan dari curah hujan pada wilayah Poncokusumo.

3.4.2 Arsitektur Sistem Peramalan

Arsitektur sistem peramalan curah hujan dengan metode *Support Vector Regression-Ant Colony Optimization* (SVR-ANT CO) dapat dilihat pada Gambar 3.4



**Gambar 3.4 Arsitektur peramalan runtu waktu curah hujan menggunakan metode SVR-ACO**

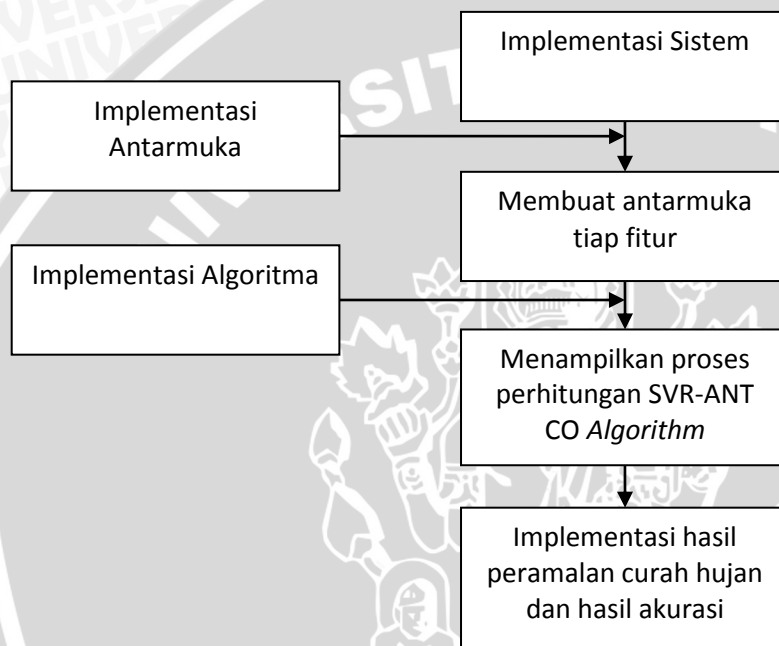
Gambar 3.4 menjelaskan arsitektur sistem peramalan, fase pembangunan model berisi definisi masalah kemudian pengumpulan data dengan analisis yang dilanjutkan dengan seleksi model untuk peramalan yaitu SVR-ANT CO. Sistem menggunakan metode *Support Vector Regression-Ant Colony Optimization* (SVR-



ANT CO). Sistem peramalan menggunakan Metode SVR kemudian dioptimasi dengan menggunakan ANT CO. Setelah fase pembangunan model dilakukan fase peramalan dengan melakukan validasi model kemudian menerapkan model yang telah dipilih dan melakukan evaluasi kinerja untuk melihat bagaimana hasil peramalan menggunakan data dan model yang dipilih.

### 3.5 Implementasi

Implementasi sistem adalah fase membangun sistem yang mengacu pada perancangan peramalan dan menerapkan hal yang telah didapatkan dalam proses



studi literatur.

**Gambar 3.5 Implementasi sistem**

Fase-fase yang ada dalam proses implementasi antara lain:

1. Implementasi antarmuka dari setiap fitur
2. Implementasi algoritma, melakukan proses perhitungan dengan metode SVR yang telah dioptimasi dengan metode ANT CO kedalam bahasa pemrograman
3. Hasil implementasi yang menghasilkan sistem peramalan runut waktu curah hujan menggunakan metode SVR-ANT CO.

### 3.6 Analisis dan pengujian sistem

Analisis dan pengujian sistem dilakukan untuk mengetahui apakah sistem berjalan dengan baik dan sesuai dengan spesifikasi kebutuhan yang telah diterapkan dapat berjalan dengan baik pada peramalan runut waktu curah hujan menggunakan metode SVR-ANT CO. Uji coba yang dilakukan untuk mengevaluasi yang dilakukan antara lain:

1. Uji coba pengujian iterasi SVR
2. Uji coba pengujian SVR
3. Uji coba jumlah iterasi ANT CO
4. Uji coba jumlah semut
5. Uji coba untuk menentukan batas parameter metode SVR yang paling optimal
6. Uji coba pengujian SVR-ANT CO
7. Uji coba pengujian jumlah variasi variabel lag SVR-ANT CO
8. Uji coba pengujian variasi data latih SVR-ANT CO
9. Uji coba nilai *error* dan akurasi yang dihasilkan menggunakan metode SVR-ANT CO dengan menggunakan persamaan RMSE

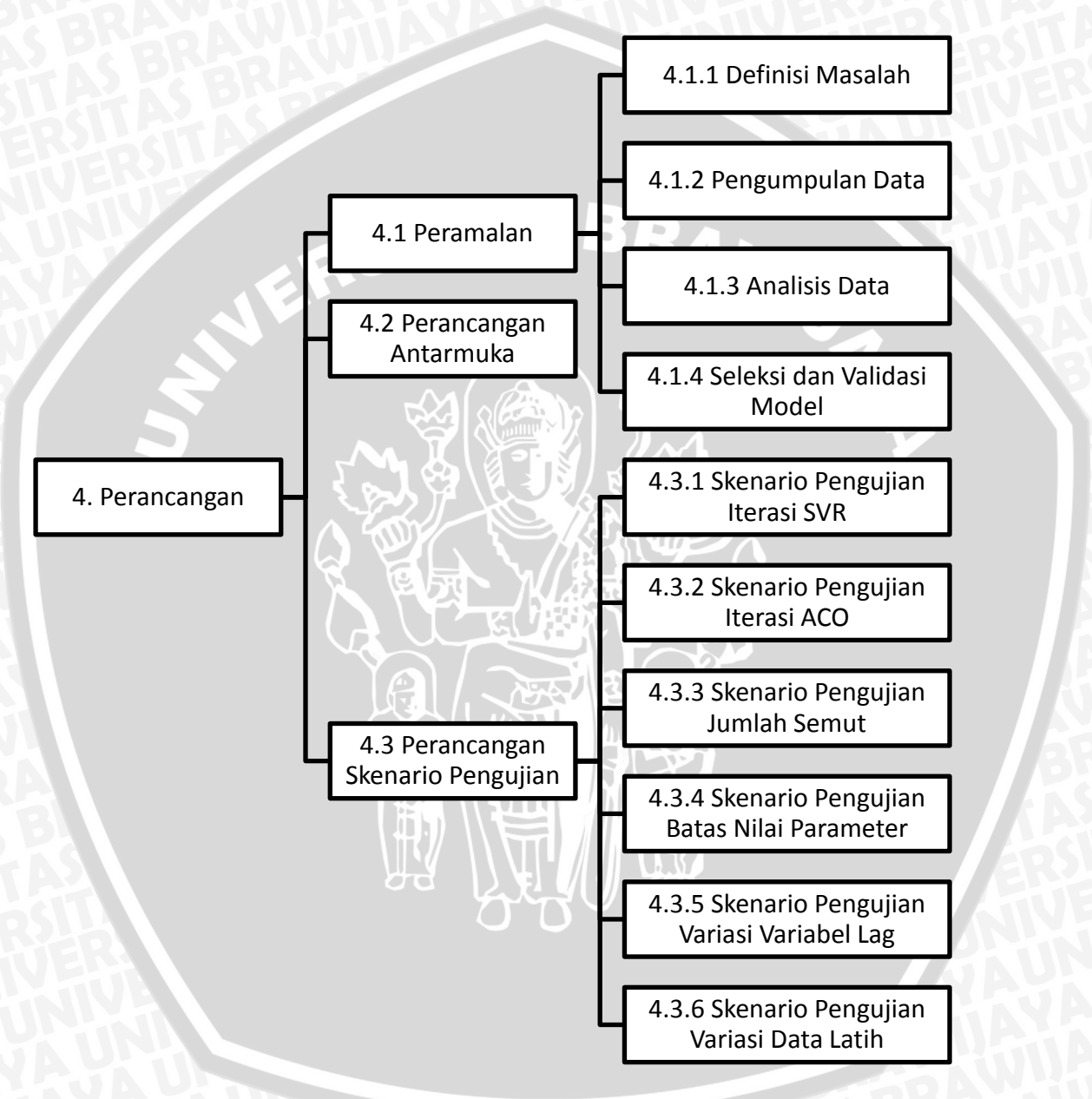
### 3.7 Kesimpulan

Kesimpulan dilakukan setelah semua tahap perancangan, implementasi dan pengujian metode yang diterapkan sudah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis metode sistem peramalan runtu waktu curah hujan menggunakan metode SVR-ANT CO. Kesimpulan yang disusun merupakan hasil pengujian dan analisis yang mampu menjawab rumusan masalah yang telah dibuat. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan yang terjadi serta memberikan pertimbangan untuk penelitian selanjutnya baik penelitian mengenai curah hujan maupun tentang penggunaan metode SVR-ANT CO.



## BAB 4 PERANCANGAN

Pada bab perancangan akan menjelaskan bagaimana perancangan peramalan menggunakan metode SVR dan ANT CO beserta perancangan antarmuka dan skenario pengujian. Tahapan dari perancangan ditunjukkan pada Gambar 4.1



Gambar 4.1 Diagram blok perancangan

## 4.1 Peramalan

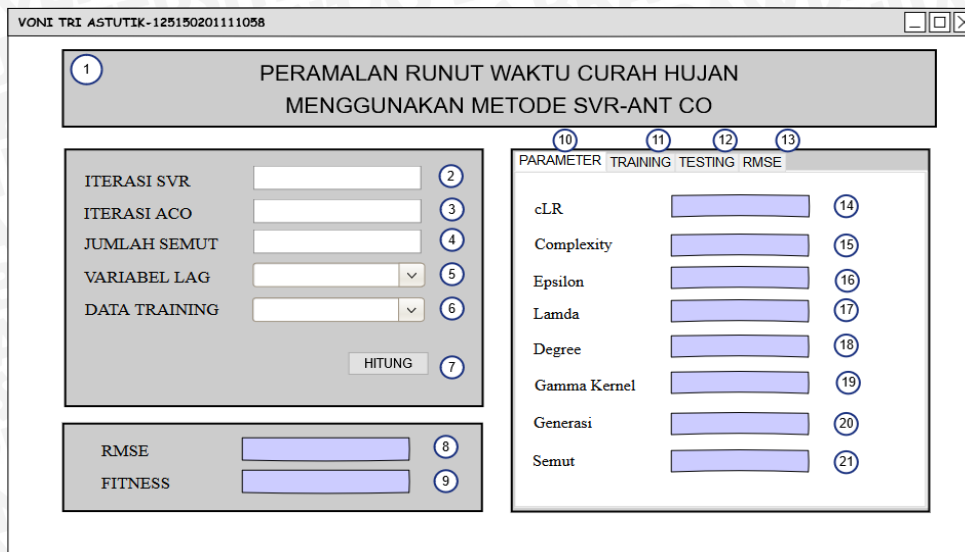
Perancangan sistem peramalan menjelaskan tahapan dari sistem peramalan, tahapan peramalan yang dilakukan pada penelitian ini yaitu definisi masalah, pengumpulan data, analisis data, seleksi model dan validasi model. Pada tahapan definisi masalah yang merupakan penjabaran masalah yang menyebabkan dibutuhkannya sistem peramalan. Kedua pengumpulan data yang dibutuhkan dalam peramalan kemudian menganalisis data yang telah dikumpulkan. Tahap berikutnya adalah pemilihan model peramalan kemudian melakukan validasi model yang telah dipilih.

### 4.1.1 Definisi Masalah

Tujuan dari definisi masalah adalah untuk membantu dalam penentuan metode peramalan yang digunakan dan bagaimana analisis data yang akan mempengaruhi hasil dari peramalan. Pada penelitian ini masalah peramalan yang dibahas adalah peramalan runtu waktu curah hujan. Secara klimatologi wilayah Indonesia terdapat 407 pola iklim yang terjadi mempengaruhi curah hujan di Indonesia. Menurut Aldrian & Djamil (2008), Indonesia merupakan negara maritim yang memiliki curah hujan yang tinggi dengan rata-rata curah hujan 2700mm per tahun. Curah hujan memiliki pola yang tidak dapat dipastikan, sedangkan curah hujan merupakan faktor penentu pengelolaan sumber daya air. Jika pengelolaan sumber daya air dilakukan dengan baik akan berdampak dengan ketersediaan sumber daya air di masyarakat terutama pada sektor pertanian. Namun jika sumber daya air tidak dapat dikelola dengan baik maka dapat menimbulkan bencana seperti bencana banjir atau kekeringan. Untuk dapat menentukan pengelolaan sumber daya air yang tepat maka dibutuhkan suatu teknologi peramalan runtu waktu curah hujan yang akurat dan terperinci.

## 4.2 Perancangan Antarmuka

Pada perancangan antarmuka untuk halaman utama berisi *input* parameter yang akan digunakan untuk perhitungan data latih, halaman ini terdapat nama aplikasi, tombol untuk melanjutkan proses perhitungan data latih. Perancangan halaman utama dapat dilihat pada Gambar 4.48



Gambar 4.2 Perancangan antarmuka

Keterangan :

1. Nama aplikasi.
2. *Textbox* untuk mengisi nilai jumlah iterasi SVR.
3. *Textbox* untuk mengisi nilai jumlah iterasi ANT-CO.
4. *Textbox* untuk mengisi nilai jumlah semut.
5. *Combo box* untuk memilih nilai jumlah variabel lag.
6. *Combo box* untuk memilih nilai jumlah data latih.
7. *Button* untuk melakukan proses perhitungan.
8. *Label* untuk menampilkan nilai RMSE data uji.
9. *Label* untuk menampilkan nilai *fitnes* data uji.
10. *Tab panel* untuk pindah ke panel parameter.
11. *Tab panel* untuk pindah ke panel *training*.
12. *Tab panel* untuk pindah ke panel *testing*.
13. *Tab panel* untuk pindah ke panel RMSE.
14. *Label* untuk menampilkan nilai parameter *cLR*.
15. *Label* untuk menampilkan nilai parameter kompleksitas.
16. *Label* untuk menampilkan nilai parameter *epsilon*.
17. *Label* untuk menampilkan nilai parameter *lamda*.
18. *Label* untuk menampilkan nilai parameter *degree*.
19. *Label* untuk menampilkan nilai parameter *gamma kernel*.
20. *Label* untuk menampilkan nilai generasi terbaik.
21. *Label* untuk menampilkan nilai semut terbaik.

### 4.3 Perancangan Pengujian

Dalam menentukan parameter terbaik dari metode SVR-ANT CO dibutuhkan suatu perancangan skenario pengujian untuk mendapatkan nilai parameter yang optimal. Uji cobatersebut antara lain:

1. Pengujian untuk menentukan jumlah iterasi SVR yang optimal.
2. Pengujian untuk menentukan jumlah iterasi ANT CO yang optimal.

3. Pengujian untuk menentukan jumlah semut yang optimal.
4. Pengujian untuk menentukan batas nilai parameter SVR yang optimal.
5. Pengujian untuk menentukan variasi variabel lag.
6. Pengujian untuk menentukan variasi jumlah data latih.

#### 4.3.1 Perancangan Skenario Pengujian Iterasi Pelatihan SVR

Pengujian iterasi SVR yaitu uji coba yang digunakan untuk mengetahui jumlah iterasi yang dibutuhkan untuk mendapatkan nilai akurasi yang optimal. Rancangan pengujian jumlah iterasi SVR terdapat pada Tabel.45

**Tabel 4.1 Rancangan Uji Coba iterasi SVR**

Jumlah Iterasi	Nilai RMSE Percobaan Ke-i										Rerata RMSE
	1	2	3	4	5	6	7	8	9	10	
10											
100											
500											
1000											

#### 4.3.2 Perancangan Skenario Pengujian SVR

Uji coba pengujian SVR yaitu uji coba yang digunakan untuk membandingkan hasil fitness pada iterasi SVR sebelum dilakukan optimasi dengan metode ANT CO yang menggunakan parameter dari Tabel 4.16. Rancangan pengujian SVR terdapat pada Tabel 4.50 berikut:

**Tabel 4.2 Pengujian SVR**

Percobaan ke-	RMSE
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
RMSE terbaik	

#### 4.3.3 Perancangan Skenario Pengujian Jumlah Iterasi ANT CO

Uji coba banyaknya iterasi ANT CO yaitu uji coba yang digunakan untuk mengetahui jumlah iterasi yang dibutuhkan untuk mendapatkan nilai parameter yang optimal. Rancangan pengujian jumlah iterasi ANT CO terdapat pada Tabel.51

**Tabel 4.3 Rancangan Uji Coba iterasi ANT CO**

Jumlah Iterasi	Nilai RMSE Percobaan Ke-i										Rerata RMSE
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											



50												
100												
150												
200												

#### 4.3.4 Perancangan Skenario Pengujian Jumlah Semut

Pengujian jumlah semut yaitu uji coba yang digunakan untuk mengetahui banyaknya jumlah semut yang diperlukan untuk mendapatkan nilai parameter SVR yang optimal. Rancangan pengujian jumlah semut dapat dilihat pada Tabel.52

**Tabel 4.4 Rancangan Uji Coba Jumlah Semut**

Jumlah Semut	Nilai RMSE Percobaan Ke-i										Rerata RMSE	
	1	2	3	4	5	6	7	8	9	10		
5												
10												
15												
20												
25												
30												

#### 4.3.5 Perancangan Skenario Pengujian Batas Nilai Parameter

Pengujian batas nilai parameter adalah uji coba untuk mengetahui panjang nilai *range* ruang pencarian dimensi agar dapat menghasilkan kombinasi parameter SVR yang optimal. Dalam rancangan uji coba batas nilai parameter *cLR* pada Tabel 4.53, untuk rancangan uji coba batas nilai parameter kompleksitas (*C*) pada Tabel 4.54, untuk rancangan uji coba batas nilai parameter *epsilon* pada Tabel 4.55, untuk rancangan uji coba batas nilai parameter *lamda* pada Tabel.56, untuk rancangan uji coba batas nilai parameter *degree* pada Tabel 4.57, untuk rancangan uji coba batas nilai parameter *gamma kernel* pada Tabel 4.58.

**Tabel 4.5 Rancangan Pengujian Batas Parameter cLR**

Batas Nilai Min <i>cLR</i>	Batas Nilai Max <i>cLR</i>	Nilai RMSE Percobaan ke-i										Rerata RMSE
		1	2	3	4	5	6	7	8	9	10	
0.001	0.1											
0.1	0.2											
0.2	0.3											
0.3	0.4											
0.4	0.5											
0.5	0.6											
0.6	0.7											
0.7	0.8											
0.8	0.9											
0.9	0.999											

**Tabel 4.6 Rancangan Pengujian Batas Parameter C**

Batas Nilai Min C	Batas Nilai Max C	Nilai RMSE Percobaan ke-i										Rerata RMSE
		1	2	3	4	5	6	7	8	9	10	
1	10											
10	100											



Batas Nilai	Batas Nilai	Nilai RMSE Percobaan ke-i										Rerata	
100	500												
500	1000												

**Tabel 4.7 Rancangan Pengujian Batas Parameter  $\epsilon$**

Batas Nilai Min $\epsilon$	Batas Nilai Max $\epsilon$	Nilai RMSE Percobaan ke-i										Rerata RMSE	
		1	2	3	4	5	6	7	8	9	10		
0.000001	0.00000999												
0.00001	0.0000999												
0.0001	0.000999												
0.001	0.0099												
0.01	0.099												

**Tabel 4.8 Rancangan Pengujian Batas Parameter  $\lambda$**

Batas Nilai Min $\lambda$	Batas Nilai Max $\lambda$	Nilai RMSE Percobaan ke-i										Rerata RMSE	
		1	2	3	4	5	6	7	8	9	10		
0.001	0.1												
0.1	0.2												
0.2	0.3												
0.3	0.4												
0.4	0.5												
0.5	0.6												
0.6	0.7												
0.7	0.8												
0.8	0.9												
0.9	0.999												

**Tabel 4.9 Rancangan Pengujian Batas Parameter  $d$**

Batas Nilai Min $d$	Batas Nilai Max $d$	Nilai RMSE Percobaan ke-i										Rerata RMSE	
		1	2	3	4	5	6	7	8	9	10		
1	2												
2	3												
3	4												
4	5												
5	6												
6	7												
7	8												
8	9												
9	9.999												

**Tabel 4.10 Rancangan Pengujian Batas Parameter  $\gamma_k$**

Batas Nilai Min $\gamma$	Batas Nilai Max $\gamma$	Nilai RMSE Percobaan ke-i										Rerata RMSE	
		1	2	3	4	5	6	7	8	9	10		
1	10												
10	100												
100	500												



Batas Nilai	Batas Nilai	Nilai RMSE Percobaan ke-i								Rerata
500	1000									

#### 4.3.6 Perancangan Skenario Pengujian SVR-ANT CO

Uji coba pengujian SVR-ANT CO yaitu uji coba yang digunakan untuk membandingkan hasil RMSE pada iterasi SVR tanpa optimasi ANT CO pada subbab 4.3.2 Tabel 4.50 dengan RMSE iterasi SVR yang menggunakan batas parameter yang telah ditentukan dengan optimasi ANT CO. Rancangan pengujian SVR-ANT CO terdapat pada Tabel 4.59

**Tabel 4.11 Pengujian SVR-ANT CO**

Percobaan ke-	RMSE
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
RMSE Terbaik	

#### 4.3.7 Perancangan Skenario Perbandingan Pengujian SVR dan Pengujian SVR-ANT CO

Uji coba perbandingan pengujian SVR dengan pengujian SVR-ANT CO yaitu uji coba yang digunakan untuk mengetahui nilai *error* yang terkecil dari kedua pengujian tersebut. Rancangan perbandingan pengujian SVR dengan pengujian SVR-ANT CO terdapat pada Tabel 4.60 berikut:

**Tabel 4.12 Perbandingan Pengujian SVR dan SVR-ANT CO**

Percobaan ke-	RMSE SVR	RMSE SVR-ANT CO
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
RMSE terbaik		



#### 4.3.8 Perancangan Skenario Pengujian Variasi Variabel Lag

Pengujian variasi variabel lag yaitu uji coba yang digunakan untuk mengetahui jumlah variabel lag yang dibutuhkan untuk mendapatkan nilai akurasi yang optimal. Rancangan pengujian jumlah variasi variabel lag terdapat pada Tabel 4.61

**Tabel 4.13 Rancangan Pengujian Variasi Variabel Lag**

Variabel Lag	Nilai RMSE Percobaan ke- <i>i</i>										Rerata RMSE
	1	2	3	4	5	6	7	8	9	10	
2											
3											
4											
5											
6											
7											
8											

#### 4.3.9 Perancangan Skenario Pengujian Variasi Data Latih

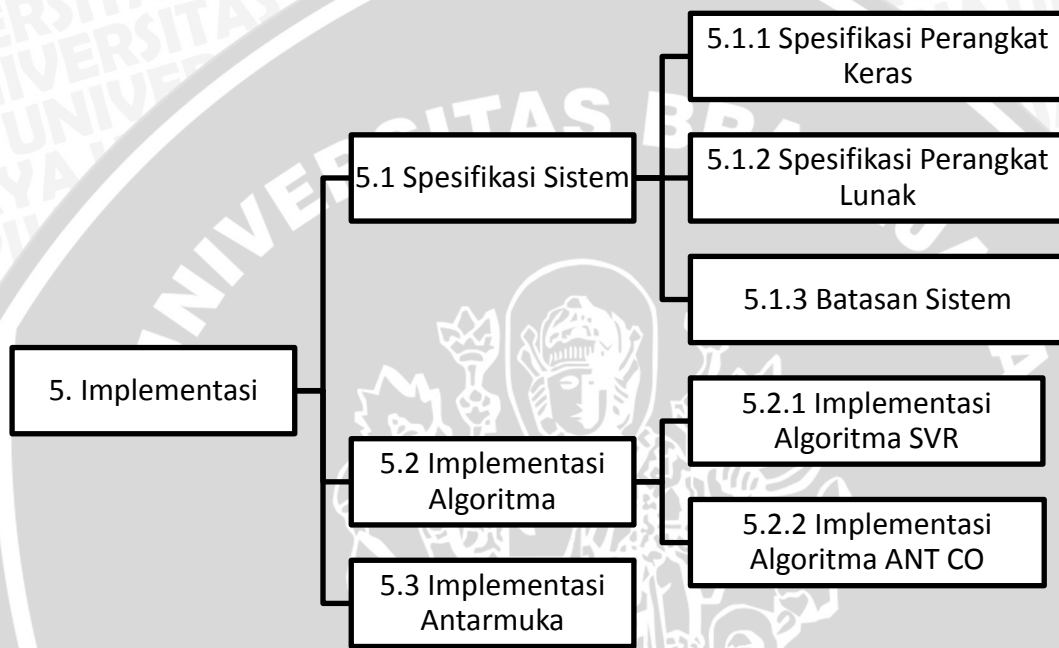
Pengujian variasi data latih yaitu uji coba yang digunakan untuk mengetahui jumlah data latih yang dibutuhkan untuk mendapatkan nilai akurasi yang optimal. Rancangan pengujian jumlah variasi data latih terdapat pada Tabel 4.62

**Tabel 4.14 Rancangan Pengujian Variasi Data Latih**

Variasi Data Latih	Nilai Fitness Percobaan ke- <i>i</i>										Rerata RMSE
	1	2	3	4	5	6	7	8	9	10	
2											
3											
4											
5											
6											
7											
8											

## BAB 5 IMPLEMENTASI

Pada bab ini membahas tentang implementasi dari perancangan sistem peramalan runtu waktu curah hujan menggunakan metode *Support Vector Regression* dan *Ant Colony Optimization (SVR-ANT CO)*. Pembahasan implementasi meliputi beberapa bagian *antara* lain spesifikasi sistem, batasan sistem, implementasi algoritma dan implementasi antarmuka yang ditunjukkan pada Gambar 5.1.



Gambar 5.1 Implementasi

### 5.1 Spesifikasi Sistem

Spesifikasi sistem ini didasarkan dari hasil perancangan yang telah diuraikan pada bab metodologi. Pada bagian implementasi sistem ini dibahas tentang bagaimana spesifikasi dari sistem dan batasan-batasan sistem dalam implementasi yang ada. Lingkungan implementasi terdiri dari lingkungan perangkat keras dan lingkungan perangkat lunak.

#### 5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan pada pengembangan sistem ini dijelaskan pada Tabel 5.1

Tabel 5.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	Intel(R) Core(TM) i5 CPU M 380 @ 2.53GHz (4 CPUs), ~2.5GHz

Memori (RAM)	2048 MB
Harddisk	500 GB

### 5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan pada pengembangan sistem ini dijelaskan pada Tabel 5.2

**Tabel 5.2 Spesifikasi Perangkat Lunak**

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 10
Bahasa Pemrograman	Java
Tools Pemrograman	Netbeans 8.0.2

### 5.1.3 Batasan-Batasan Sistem

Berikut beberapa batasan yang membatasi pengembangan sistem ini:

1. *Input* yang diterima sistem adalah berupa data curah hujan
2. *Output* yang diberikan berupa peramalan curah hujan dalam beberapa tahun kedepan
3. Pengembangan sistem dilakukan dengan menggunakan bahasa pemrograman Java
4. Algoritma yang diimplementasikan adalah *Support Vector Regression (SVR)* dan *Ant Colony Optimization (ANT CO)*.

## 5.2 Implementasi Algoritma

Implementasi algoritma dibagi menjadi 2 proses utama, yaitu implementasi algoritma *Support Vector Regression* yang terdiri dari 20 utama proses dan *Ant Colony Optimization* yang memiliki proses utama.

### 5.2.1 Implementasi Algoritma SVR

Implementasi algoritma SVR terdiri dari beberapa 20 bagian yaitu

1. Proses pengambilan data

Fungsi `setData` digunakan untuk pengambilan data dalam bentuk `.txt` yang kemudian akan dipakai dalam proses peramalan. Data yang digunakan akan dirubah ke dalam *array* dua dimensi. Proses pengambilan data dapat ditunjukkan pada Gambar 5.2

```

1 public double[][] setData() {
2     data = new double[banyakData][banyakLag];
3     try {
4         File file = new File(dataCh);
5         BufferedReader br = new BufferedReader(new
6             FileReader(file));
7         String line;
8         int j = 0;
9         while ((line = br.readLine()) != null) {

```

```

10     int start = line.indexOf("");
11     String sub = line.substring(start);
12     String[] vertex = sub.split(",");
13     for (int k = 0; k < vertex.length; k++) {
14         data[j][k] = Double.parseDouble(vertex[k]);
15     }
16     j++;
17 }
18 br.close();
19 } catch (Exception e) {
20     System.out.println("Error: " + e.getMessage());
21 }
22 return data;
23 }

```

**Gambar 5.2 Implementasi algoritma proses pengambilan data**

Penjelasan Gambar 5.2 adalah sebagai berikut :

- Baris 1 merupakan inialisaii nama *method*
  - Baris 2 merupakan inialisasi variabel data.
  - Baris 4-5 merupakan pencarian file yang akan dijadikan data peramalan.
  - Baris 7-17 merupakan fungsi perulangan untuk merubah nilai data ke dalam bentuk *array*.
  - Baris 18-21 merupakan fungsi yang akan menampilkan pesan bila fungsi pada baris 3-17 tidak.
  - Baris 22 merupakan proses pengambilan nilai data.
2. Proses pencarian nilai terkecil

Fungsi `getMinData` digunakan mencari nilai terkecil data. Proses pencarian nilai terkecil dapat dilihat pada Gambar 5.3

```

1 private double getMinData() {
2     minData = this.data[0][0];
3     for (int i = 0; i < data.length; i++) {
4         for (int j = 0; j < data[0].length; j++) {
5             if (data[i][j] < minData) {
6                 minData = data[i][j];
7             }
8         }
9     }
10    return minData;
11 }

```

**Gambar 5.3 Implementasi algoritma proses pencarian nilai terkecil**

Penjelasan Gambar 5.3 adalah sebagai berikut :

- Baris 1 merupakan inialisasi nama *method*
- Baris 2 merupakan inialisasi variabel data
- Baris 3-9 merupakan proses pencarian data terkecil
- Baris 10 merupakan proses pengembalian nilai `minData`

### 3. Proses pencarian nilai terbesar

Fungsi `getMaksData` digunakan mencari nilai terbesar data. Proses pencarian nilai terbesar dapat dilihat pada Gambar 5.4

```

1 private double getMaksData() {
2     double maksData = this.data[0][0];
3     for (int i = 0; i < data.length; i++) {
4         for (int j = 0; j < data[0].length; j++) {
5             if (data[i][j] > maksData) {
6                 maksData = data[i][j];
7             }
8         }
9     }
10    return maksData;
11 }

```

**Gambar 5.4 Implementasi algoritma proses pencarian nilai terbesar**

Penjelasan Gambar 5.4 adalah sebagai berikut :

- Baris 2 merupakan inisialisasi variabel data
- Baris 3-9 merupakan proses pencarian data terbesar
- Baris 10 merupakan proses pengembalian nilai `maksData`

### 4. Proses normalisasi data

Fungsi normalisasi digunakan untuk proses normalisasi terhadap seluruh data yang akan diolah. Proses normalisasi dapat dilihat pada Gambar 5.5

```

1 public void normalisasi() {
2     normalisasi = new
3     double[data.length][data[0].length];
4     minData = getMinData();
5     range = getMaksData() - getMinData();
6     for (int i = 0; i < data.length; i++) {
7         for (int j = 0; j < data[0].length; j++) {
8             normalisasi[i][j] = (data[i][j] - minData) /
9             range;
10        }
11    }
12 }
13 }

```

**Gambar 5.5 Implementasi algoritma proses normalisasi data**

Penjelasan Gambar 5.5 sebagai berikut :

- Baris 2-5 merupakan inisialisasi variabel data.
- Baris 6-9 merupakan proses perulangan dalam melakukan normalisasi data.

### 5. Proses perhitungan kernel

Fungsi `anovaKernel` digunakan untuk menghitung nilai kernel dengan kernel anova yang digunakan. Proses perhitungan kernel dapat dilihat pada Gambar 5.6

```

1 public void anovaKernel() {
2     anovaKernel = new
3     double[banyakTraining][banyakTraining];
4     for (int i = 0; i < anovaKernel.length; i++) {
5         for (int j = 0; j < anovaKernel[0].length; j++)
6         {
7             double value = 0;
8             for (int k = 0; k < banyakLag - 1; k++) {
9                 value += Math.exp(-gammaKernel *
10 (Math.pow(normalisasi[i][k] - normalisasi[j][k], 2)));
11             }
12             value = Math.pow(value, degree);
13             anovaKernel[i][j] = value;
14         }
15     }
16 }

```

**Gambar 5.6 Implementasi algoritma proses pencarian nilai kernel**

Penjelasan Gambar 5.6 adalah sebagai berikut :

- a. Baris 2 merupakan inisialisasi variabel data
  - b. Baris 4-16 merupakan proses perulangan perhitungan kernel anova
6. Proses perhitungan Matriks Hessian  
Fungsi matriksHesian digunakan untuk mencari nilai matriks hessian dari data yang didapat dari proses perhitungan nilai kernel. Proses perhitungan matriks hessian dapat dilihat pada Gambar 5.7

```

1 public void matriksHessian() {
2     matriksHessian = new
3     double[banyakTraining][banyakTraining];
4     for (int i = 0; i < matriksHessian.length; i++) {
5         for (int j = 0; j < matriksHessian[0].length;
6 j++) {
7             matriksHessian[i][j] = anovaKernel[i][j] +
8 Math.pow(lambda, 2);
9         }
10     }
11 }
12 }
13 }

```

**Gambar 5.7 Implementasi algoritma proses pencarian nilai Matriks Hessian**

Penjelasan Gambar 5.7 adalah sebagai berikut :

- a. Baris 2 merupakan inisialisasi variabel data
  - b. Baris 4-13 merupakan proses perulangan perhitungan matriks hessian dengan menambah nilai kernel yang telah didapat dengan lamda yang dikuadratkan
7. Proses *sequential learning*  
Fungsi *sekuensialLearning* digunakan untuk mencari nilai  $E$ ,  $\alpha$ ,  $\alpha^*$ ,  $\partial\alpha$ , dan  $\partial\alpha^*$ . Proses perhitungan *sequential learning* dapat dilihat pada Gambar 5.8



```

1 public void sekuensialLearning() {
2     nilaiE = new double[banyakTraining];
3     deltaAlphaStar = new double[banyakTraining];
4     deltaAlpha = new double[banyakTraining];
5     alphaStar = new double[banyakTraining];
6     alpha = new double[banyakTraining];
7     gamma = cLR / matriksHessian[0][0];
8     for (int i = 0; i < banyakIterasi; i++) {
9         hitungE();
10        hitungDeltaAlphaStar();
11        hitungDeltaAlpha();
12        hitungAlphaStar();
13        hitungAlpha();
14    }
15    }
16 }

```

**Gambar 5.8 Implementasi algoritma proses *sequential learning***

Penjelasan Gambar 5.8 adalah sebagai berikut :

- Baris 2-7 merupakan inisialisasi variabel data
- Baris 8-13 merupakan proses perulangan pencarian nilai  $E$  dengan memanggil method `hitungE()`,  $\alpha$  dengan memanggil method `hitungAlpha()`,  $\alpha^*$  dengan memanggil method `hitungAlphaStar()`,  $\partial\alpha$  dengan memanggil method `hitungDeltaAlphaStar()`, dan  $\partial\alpha^*$  dengan memanggil method `hitungDeltaAlphaStar()`.

#### 8. Proses perhitungan nilai $E$

Fungsi `hitungE` digunakan untuk mencari nilai  $E$  dari data latih. nilai kernel dengan kernel anova yang digunakan. Proses perhitungan kernel dapat dilihat pada Gambar 5.9

```

1 private void hitungE() {
2     for (int i = 0; i < banyakTraining; i++) {
3         double aktual =
4         normalisasi[i][normalisasi[0].length - 1];
5         double value_e = 0;
6         for (int m = 0; m < banyakTraining; m++) {
7             value_e += (alphaStar[m] - alpha[m]) *
8             matriksHessian[i][m];
9
10        }
11        nilaiE[i] = aktual - value_e;
12    }
13 }
14 }

```

**Gambar 5.9 Implementasi algoritma proses pencarian nilai  $E$**

Penjelasan Gambar 5.9 adalah sebagai berikut :

- Baris 2-4 merupakan perulangan untuk mencari nilai aktual
- Baris 6-12 merupakan perulangan untuk menghitung variabel `value_e`

- c. Baris 11 merupakan perhitungan nilai  $E$  dengan mengurangi nilai aktual dan  $value\_e$  yang telah didapat sebelumnya.

9. Proses perhitungan nilai  $\vartheta\alpha^*$

Fungsi `hitungDeltaAlphaStar` digunakan untuk mencari nilai  $\vartheta\alpha^*$  dari data latih. Proses perhitungan  $\vartheta\alpha^*$  dapat dilihat pada Gambar 5.10

```

1 private void hitungDeltaAlphaStar() {
2     for (int i = 0; i < banyakTraining; i++) {
3         deltaAlphaStar[i] = Math.min(Math.max(gamma *
4 (nilaiE[i] - epsilon), -alphaStar[i]), kompleksitas -
5 alphaStar[i]);
6     }
7 }
8 }
9 
```

**Gambar 5.10 Implementasi algoritma proses pencarian nilai  $\vartheta\alpha^*$**

Penjelasan Gambar 5.10 adalah sebagai berikut :

- a. Baris 2-6 merupakan perulangan untuk mencari nilai variabel `deltaAlphaStar`

10. Proses perhitungan nilai  $\vartheta\alpha$

Fungsi `hitungDeltaAlpha` digunakan untuk mencari nilai  $\vartheta\alpha$  dari data latih. Proses perhitungan  $\vartheta\alpha$  dapat dilihat pada Gambar 5.11

```

1 private void hitungDeltaAlpha() {
2     for (int i = 0; i < banyakTraining; i++) {
3         deltaAlpha[i] = Math.min(Math.max(gamma * (-
4 nilaiE[i] - epsilon), -alpha[i]), kompleksitas - alpha[i]);
5     }
6 }
7 
```

**Gambar 5.11 Implementasi algoritma proses pencarian nilai  $\vartheta\alpha$**

Penjelasan Gambar 5.11 adalah sebagai berikut :

- a. Baris 2-4 merupakan perulangan untuk mencari nilai variabel `deltaAlpha`

11. Proses perhitungan nilai  $\alpha^*$

Fungsi `hitungAlphaStar` digunakan untuk mencari nilai  $\alpha^*$  dari data latih. Proses perhitungan  $\alpha^*$  dapat dilihat pada Gambar 5.12

```

1 private void hitungAlphaStar() {
2     for (int i = 0; i < banyakTraining; i++) {
3         alphaStar[i] = deltaAlphaStar[i] + alphaStar[i];
4     }
5 }
6 
```

**Gambar 5.12 Implementasi algoritma proses pencarian nilai  $\alpha^*$**

Penjelasan Gambar 5.12 adalah sebagai berikut :

- a. Baris 2-4 merupakan perulangan untuk mencari nilai variabel `alphaStar`

12. Proses perhitungan nilai  $\alpha$

Fungsi `hitungAlpha` digunakan untuk mencari nilai  $\alpha$  dari data latih. Proses perhitungan  $\alpha$  dapat dilihat pada Gambar 5.13

```

1 private void hitungAlpha() {
2     for (int i = 0; i < banyakTraining; i++) {
3         alpha[i] = deltaAlpha[i] + alpha[i];
4     }
5 }
6 }
7 }

```

**Gambar 5.13 Implementasi algoritma proses pencarian nilai  $\alpha$**

Penjelasan Gambar 5.13 adalah sebagai berikut :

a. Baris 2-4 merupakan perulangan untuk mencari nilai variabel `alpha`

### 13. Proses perhitungan $f(x)$

Fungsi `hitungfx_training` digunakan untuk mencari nilai  $f(x)$  dari data latih.

Proses perhitungan  $f(x)$  dapat dilihat pada Gambar 5.14

```

1 private void hitungfx_training() {
2     fx_training = new double[banyakTraining];
3     for (int i = 0; i < banyakTraining; i++) {
4         double value_fx = 0;
5         for (int k = 0; k < banyakTraining; k++) {
6             value_fx += (alphaStar[k] - alpha[k]) *
7 matriksHessian[i][k];
8         }
9         fx_training[i] = value_fx;
10    }
11 }

```

**Gambar 5.14 Implementasi algoritma proses perhitungan nilai  $f(x)$**

Penjelasan Gambar 5.14 adalah sebagai berikut :

- Baris 2 merupakan inisialisasi variabel data
- Baris 3-9 merupakan proses perulangan pencarian nilai `value_fx` dan nilai `fx_training` adalah nilai `value_fx`.

### 14. Proses perhitungan denormalisasi

Fungsi `hitungDenorm_training` digunakan untuk mencari nilai denormalisasi data latih. Proses perhitungan denormalisasi dapat dilihat pada Gambar 5.15

```

1 private void hitungDenorm_training() {
2     denorm = new double[banyakTraining];
3     for (int i = 0; i < banyakTraining; i++) {
4         denorm[i] = (fx_training[i] * range) + minData;
5     }
6 }

```

**Gambar 5.15 Implementasi algoritma proses perhitungan nilai denormalisasi**

Penjelasan Gambar 5.15 adalah sebagai berikut :

- Baris 2 merupakan inisialisasi variabel data
- Baris 3-9 merupakan proses perulangan pencarian nilai denormalisasi data.

15. Proses perhitungan RMSE data latih

Fungsi `getRMSETraining` digunakan untuk mencari nilai RMSE data latih. Proses perhitungan RMSE dapat dilihat pada Gambar 5.16

```
1 public double getRMSETraining() {
2     double value = 0;
3     for (int i = 0; i < banyakTraining; i++) {
4         value += Math.pow(data[i][data[0].length - 1] -
5 denorm[i], 2);}
6     double MSE = value / banyakTraining;
7     double RMSE = Math.sqrt(MSE);
8     return RMSE; }
```

**Gambar 5.16 Implementasi algoritma proses perhitungan nilai rmse data latih**

Penjelasan Gambar 5.16 adalah sebagai berikut :

- Baris 2 merupakan inisialisasi variabel data
- Baris 3-5 merupakan proses perulangan pencarian nilai `value`.
- Baris 6 merupakan proses perhitungan MSE
- Baris 7 merupakan proses perhitungan RMSE
- Baris 8 merupakan pengembalian nilai RMSE

16. Proses perhitungan data uji SVR

Fungsi `testingSVR` digunakan untuk melakukan proses testing. Proses testing SVR dapat dilihat pada Gambar 5.17

```
1 public void testingSVR() {
2     int banyak_testing = banyakData - banyakTraining;
3     fx_testing = new double[banyak_testing];
4     double[] data_forecast = new double[banyakLag - 1];
5     for (int i = 0; i < data_forecast.length; i++) {
6         data_forecast[i] =
7 normalisasi[normalisasi.length - 1][i + 1];
8     }
9
10    for (int i = 0; i < fx_testing.length; i++) {
11        double hessian_testing[] = new
12 double[banyakTraining];
13        hessian_testing =
14 getHessianTesting(data_forecast);
15        double peramalan = 0;
16        for (int j = 0; j < banyakTraining; j++) {
17            peramalan += (alphaStar[j] - alpha[j]) *
18 hessian_testing[j];
19        }
20        fx_testing[i] = peramalan;
21
22        for (int j = 0; j < data_forecast.length - 1;
23 j++) {
24            data_forecast[j] = data_forecast[j + 1];
25        }
26        data_forecast[data_forecast.length - 1] =
27 peramalan;
```

```

28     }
29
30     }

```

**Gambar 5.17 Implementasi algoritma proses perhitungan nilai rmse data uji**

Penjelasan Gambar 5.17 adalah sebagai berikut :

- Baris 2-4 merupakan inisialisasi variabel data.
- Baris 5-8 merupakan proses perulangan pencarian data\_forecast.
- Baris 10-14 merupakan proses perulangan untuk mencari matrik hessian.
- Baris 15-20 merupakan proses perulangan pencarian  $f(x)$ .
- Baris 22-30 merupakan proses perulangan pencarian data peramalan

#### 17. Proses perhitungan Matriks Hessian testing

Fungsi `getHessianTesting` digunakan untuk mencari nilai matriks hessian dari data uji. Proses perhitungan matriks hessian testing dapat dilihat pada Gambar 5.18

```

1 private double[] getHessianTesting(double[] data_forecast) {
2     double[] hessian = new double[banyakTraining];
3     double[] anovaTesting = new double[banyakTraining];
4     anovaTesting = getAnovaTesting(data_forecast);
5     for (int i = 0; i < banyakTraining; i++) {
6         hessian[i] = anovaTesting[i] + Math.pow(lambda,
7     2);
8     }
9     return hessian;
10 }

```

**Gambar 5.18 Implementasi algoritma proses perhitungan matriks hessian testing**

Penjelasan Gambar 5.18 adalah sebagai berikut :

- Baris 2-4 merupakan inisialisasi variabel data
- Baris 4-10 merupakan proses perulangan perhitungan matriks hessian dengan menambah nilai kernel testing yang telah didapat dengan lamda yang dikuadratkan.

#### 18. Proses perhitungan kernel testing

Fungsi `getAnovaTesting` digunakan untuk menghitung nilai kernel dengan kernel anova untuk data uji. Proses perhitungan kernel testing dapat dilihat pada Gambar 5.19

```

1 private double[] getAnovaTesting(double[] data_forecast) {
2     double[] anovaTesting = new double[banyakTraining];
3     for (int i = 0; i < banyakTraining; i++) {
4         double value = 0;
5         for (int k = 0; k < banyakLag - 1; k++) {
6             value += Math.exp(-gammaKernel *
7 (Math.pow(normalisasi[i][k] - data_forecast[k], 2)));
8         }
9         value = Math.pow(value, degree);
10
11         anovaTesting[i] = value;

```

```

12     }
13     return anovaTesting;
14 }

```

**Gambar 5.19 Implementasi algoritma proses perhitungan nilai kernel**

Penjelasan Gambar 5.19 adalah sebagai berikut :

- a. Baris 2 merupakan inisialisasi variabel data
- b. Baris 4-16 merupakan proses perulangan perhitungan kernel anova menggunakan data hasil dari testing SVR

19. Proses perhitungan denormalisasi testing

Fungsi hitungDenorm\_testing digunakan untuk mencari nilai denormalisasi data latih. Proses perhitungan denormalisasi data uji dapat dilihat pada Gambar 5.20

```

1 private void hitungDenormTesting() {
2     denorm_testing = new double[banyakData -
3 banyakTraining];
4     for (int i = 0; i < fx_testing.length; i++) {
5         denorm_testing[i] = (fx_testing[i] * range) +
6 minData;
7     }
8 }

```

**Gambar 5.20 Implementasi algoritma proses perhitungan nilai denormalisasi**

Penjelasan Gambar 5.20 adalah sebagai berikut :

- a. Baris 2 merupakan inisialisasi variabel data
- b. Baris 3-8 merupakan proses perulangan pencarian nilai denormalisasi data uji.

20. Proses perhitungan RMSE testing

Fungsi getRMSETesting digunakan untuk mencari nilai RMSE data uji. Proses perhitungan RMSE dapat dilihat pada Gambar 5.21

```

1 public double getRMSETesting() {
2     hitungDenormTesting();
3     double value = 0;
4     for (int i = 0; i < fx_testing.length; i++) {
5         value += Math.pow(data[i
6 banyakTraining][data[0].length - 1] - denorm_testing[i], 2);
7     }
8     double MSE = value / fx_testing.length;
9     double RMSE = Math.sqrt(MSE);
10    return RMSE;
11 }

```

**Gambar 5.21 Implementasi algoritma proses perhitungan nilai RMSE data uji**

Penjelasan Gambar 5.21 adalah sebagai berikut :

- a. Baris 2 pemanggilan fungsi hitungDenormTesting()
- b. Baris 3 inisialisasi variabel
- c. Baris 4-7 merupakan proses perulangan pencarian nilai value.
- d. Baris 8 merupakan proses perhitungan MSE



- e. Baris 9 merupakan proses perhitungan RMSE
- f. Baris 10 merupakan pengembalian nilai RMSE

### 5.2.2 Implementasi Algoritma ANT CO

Implementasi algoritma ANT CO terdiri dari beberapa bagian 14 yaitu

#### 1. Fungsi Perhitungan Pheromone $\alpha$ dan $\beta$

Fungsi ini berguna untuk perhitungan nilai pheromone  $\alpha$  dan  $\beta$  untuk mencari matriks pencarian jalur *node ant*. Proses perhitungan Pheromone  $\alpha$  dan  $\beta$  dapat dilihat pada Gambar 5.22

```

1 public double[][] alphaBeta(double[][] data, double alpha,
2 double beta, int pjpgDt) {
3     double[][] result = new double[pjpgDt][pjpgDt];
4     double[][] tAlphaBeta = new double[pjpgDt][pjpgDt];
5     for (int i = 0; i < pjpgDt; i++) {
6         for (int j = 0; j < pjpgDt; j++) {
7             tAlphaBeta[i][j] = (Math.pow(data[i][j],
8 alpha)) * (Math.pow(data[i][j], beta));
9             result[i][j] = tAlphaBeta[i][j];
10        }
11    }
12    return result;
13 }

```

**Gambar 5.22 Implementasi algoritma perhitungan *pheromone alpha* dan *beta***

Penjelasan Gambar 5.22 adalah sebagai berikut :

- a. Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki empat variabel parameter
  - b. Baris 3-4 merupakan inialisasi variabel
  - c. Baris 5-11 merupakan fungsi perulangan untuk menghitung seluruh nilai *pheromone*
  - d. Baris 12-13 merupakan fungsi pengembalian nilai *result*
- #### 2. Fungsi Perhitungan Total *Pheromone* $\alpha$ dan $\beta$
- Fungsi ini berguna untuk menghitung total *pheromone*  $\alpha$  dan  $\beta$  pada setiap baris matriks. Proses perhitungan total *Pheromone*  $\alpha$  dan  $\beta$  dapat dilihat pada Gambar 5.23

```

1 public double[] totalPheromoneRU(double[][] alphaBeta, int
2 pjpgDt) {
3     double[] result = new double[pjpgDt];
4     double[] totalPru = new double[pjpgDt];
5     double tempPru;
6     for (int i = 0; i < pjpgDt; i++) {
7         tempPru = 0.0;
8         for (int j = 0; j < pjpgDt; j++) {
9             tempPru += alphaBeta[i][j];
10        }
11        totalPru[i] = tempPru;
12        result[i] = totalPru[i];

```

```

13     }
14     return result;
15 }

```

**Gambar 5.23 Implementasi algoritma perhitungan total *pheromone***

Penjelasan3 adalah sebagai berikut :

- Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki dua variabel parameter
  - Baris 3-5 merupakan inialisasi variabel
  - Baris 6-13 merupakan fungsi perulangan untuk menghitung total nilai *pheromone* setiap baris
  - Baris 14-15 merupakan fungsi pengembalian nilai *result*
3. Fungsi Perhitungan *Argument Max*  
 Fungsi ini berguna untuk menghitung hasil maksimal dari setiap baris *pheromone*. Proses perhitungan *argument max* dapat dilihat pada Gambar 5.24

```

1 public int maxTauRU(int pJgDt, double[][] alphaBeta, int r,
2 int ja, int iterAco) {
3     double maxTauRU, tempMaxPru;
4     int result = 0, u = 0, indexMax = 0;
5     maxTauRU = alphaBeta[r][0];
6     for (int i = 0; i < pJgDt; i++) {
7         if (maxTauRU < alphaBeta[r][i]) {
8             tempMaxPru = alphaBeta[r][i];
9             maxTauRU = tempMaxPru;
10            indexMax = i;
11            u = indexMax;
12            result = u;
13        } else {
14            if (maxTauRU == alphaBeta[r][i]) {
15                int[] data = {indexMax, i};
16                randomArray(data);
17                tempMaxPru = alphaBeta[r][data[0]];
18                maxTauRU = tempMaxPru;
19                indexMax = data[0];
20                u = indexMax;
21                result = u;
22            } else {
23                maxTauRU = maxTauRU;
24                indexMax = indexMax;
25                u = indexMax;
26                result = u;
27            }
28        }
29    }
30    return result;
31 }

```

**Gambar 5.24 Implementasi algoritma perhitungan *argument max***

Penjelasan Gambar 2.24 adalah sebagai berikut :



- Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki lima variabel parameter
  - Baris 3-5 merupakan inialisasi variabel yang digunakan dalam *method*
  - Baris 6-28 merupakan perhitungan *argument max*, dimana *id pheromone* yang paling optimal dijadikan node selanjutnya *ant*
  - Baris 29- 30 merupakan fungsi pengembalian nilai *result*
4. Fungsi Perhitungan *Update Pheromone Local*  
Fungsi ini berguna untuk memperbarui nilai *pheromone* yang telah terpilih menjadi *node ant*. Proses perhitungan *update pheromone local* dapat dilihat pada Gambar 5.25

```

1 public double[][] updatePheromoneLocal(double[][] data, int
2 r, int u, double rho, double Tij, int pjpgDt) {
3     double[][] result = new double[pjpgDt][pjpgDt];
4     for (int i = 0; i < pjpgDt; i++) {
5         for (int j = 0; j < pjpgDt; j++) {
6             if ((i == r) && (j == u)) {
7                 data[i][j] = ((1 - rho) * data[i][j]) +
8 (rho * Tij);
9                 result[i][j] = data[i][j];
10            } else {
11                data[i][j] = data[i][j];
12                result[i][j] = data[i][j];
13            }
14        }
15    }
16    return result;
17 }

```

**Gambar 5.25 Implementasi algoritma memperbarui *pheromone* lokal**

Penjelasan Gambar 5.25 adalah sebagai berikut :

- Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki enam variabel parameter
  - Baris 3 merupakan inialisasi variabel yang digunakan dalam *method*
  - Baris 4-15 merupakan fungsi perulangan untuk *update* nilai *pheromone* seluruh *ant*.
  - Baris 16-17 merupakan fungsi pengembalian nilai *result*
4. Fungsi Perhitungan Probabilitas  
Fungsi ini berguna untuk menghitung nilai probabilitas setiap *pheromone*. Proses perhitungan probabilitas dapat dilihat pada Gambar 5.26

```

1 public double[] probabilitas(int pjpgDt, double[][] alphaBeta,
2 double[] totalPru, int r) {
3     double[] result = new double[pjpgDt];
4     double prob[] = new double[pjpgDt];
5     for (int i = 0; i < pjpgDt; i++) {
6         prob[i] = alphaBeta[r][i] / totalPru[r];
7         result[i] = prob[i];
8     }

```

```

9      return result;
10     }

```

**Gambar 5.26 Implementasi algoritma perhitungan probabilitas**

Penjelasan Gambar 5.26 adalah sebagai berikut :

- a. Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki empat variabel parameter
- b. Baris 3-4 merupakan inialisasi variabel yang digunakan dalam *method*
- c. Baris 5-8 merupakan fungsi perulangan untuk perhitungan probabilitas
- d. Baris 9-10 merupakan fungsi pengembalian nilai *result*

6. Fungsi Perhitungan Probabilitas Kumulatif

Fungsi ini berguna untuk menghitung total nilai probabilitas kumulatif *pheromone*. Proses perhitungan probabilitas kumulatif dapat dilihat pada Gambar 5.27

```

1  public double[] probabilitasCum(int pjpgDt, double[]
2  probabilitas) {
3      double[] result = new double[pjpgDt];
4      double probCum[] = new double[pjpgDt];
5      double tempProb = 0.00;
6      for (int i = 0; i < pjpgDt; i++) {
7          tempProb += probabilitas[i];
8          probCum[i] = tempProb;
9          result[i] = probCum[i];
10     }
11     return result;
12 }

```

**Gambar 5.27 Implementasi algoritma perhitungan probabilitas kumulatif**

Penjelasan Gambar 5.27 adalah sebagai berikut :

- a. Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki dua variabel parameter
- b. Baris 3-5 merupakan inialisasi variabel yang digunakan dalam *method*
- c. Baris 6-10 merupakan fungsi perulangan untuk perhitungan probabilitas kumulatif
- d. Baris 11-12 merupakan fungsi pengembalian nilai *result*

7. Fungsi Perhitungan Index Seleksi

Fungsi ini berguna untuk menghitung index seleksi *node ant* yang baru/ Proses perhitungan index seleksi dapat dilihat pada Gambar 5.28

```

1  public int indexSeleksi(double[] probabilitasCum, int ja, int
2  iterAco, int a, int NCMMax, int jmlAnt, int jmlNodeAnt) {
3      Scanner input = new Scanner(System.in);
4      int result = 0, probSeleksi;
5      double[][][] nilaiSeleksi = new
6  double[NCMMax][jmlAnt][jmlNodeAnt];
7      nilaiSeleksi[iterAco][ja][a] = (double)
8  (Math.random() * 1);
9      for (int i = 0; i < probabilitasCum.length; i++) {

```



```

10         if (nilaiSeleksi[iterAco][ja][a] <
11     probabilitasCum[i]) {
12             probSeleksi = i;
13             result = probSeleksi;
14             break;
15         }
16     }
17     return result;
18 }

```

**Gambar 5.28 Implementasi algoritma perhitungan index seleksi**

Penjelasan Gambar 5.28 adalah sebagai berikut :

- Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki tujuh variabel parameter
- Baris 3-4 merupakan inialisasi variabel
- Baris 5-8 merupakan inialisasi nilai seleksi menggunakan fungsi *random*
- Baris 9-16 merupakan fungsi perulangan untuk menentukan index nilai seleksi
- Baris 17-18 merupakan fungsi pengembalian nilai *result*

#### 1. Fungsi Perhitungan memperbarui *Pheromone Global*

Fungsi ini berguna untuk memperbarui nilai *pheromone* secara *global*. Proses perhitungan *update pheromone global* dapat dilihat pada Gambar 5.29

```

1 public double[][][] updatePheromoneGlobal(double[][][]
2 data, double nmse, double delta, double[][][] indexX,
3 double[][][] indexY, int antUpdate, int jmlNodeAnt, int
4 iterAco, int NCmax, int ordinate, int jmlAnt) {
5     double[][][] result = new
6 double[NCmax][jmlAnt][ordinate + 1][ordinate + 1];
7     double konstanta = 1;
8     for (int i = 0; i < (ordinate + 1); i++) {
9         for (int j = 0; j < (ordinate + 1); j++) {
10             data[iterAco][antUpdate][i][j] = ((konstanta
11 - delta) * data[iterAco][antUpdate][i][j]) + (delta * 0);
12             result[iterAco][antUpdate][i][j] =
13 data[iterAco][antUpdate][i][j];
14             int count = 0;
15             for (int l = 0; l < jmlNodeAnt - 1; l++) {
16                 if (i == indexX[iterAco][antUpdate][l] &&
17 j == indexY[iterAco][antUpdate][l]) {
18                     count++;
19                     if (count == 1) {
20                         data[iterAco][antUpdate][i][j] =
21 data[iterAco][antUpdate][i][j] + (delta * (konstanta /
22 nmse));
23                         result[iterAco][antUpdate][i][j]
24 = data[iterAco][antUpdate][i][j];
25                     }
26                     if (count > 1) {
27                         data[iterAco][antUpdate][i][j] =

```

```

28 ((konstanta - delta) * data[iterAco][antUpdate][i][j]) +
29 (delta * (konstanta / nmse));
30         result[iterAco][antUpdate][i][j]
31 = data[iterAco][antUpdate][i][j];
32     }
33 }
34 }
35 }}
36     return result;
37 }

```

**Gambar 5.29 Implementasi algoritma perhitungan memperbarui *pheromone global***

Penjelasan Gambar 2.29 adalah sebagai berikut :

- Baris 1-4 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki sebelas variabel parameter
  - Baris 5-7 merupakan inialisasi variabel
  - Baris 8-36 merupakan fungsi perulangan untuk *update* nilai *pheromone* keseluruhan
  - Baris 37-38 merupakan fungsi pengembalian nilai *result*
2. Fungsi Perhitungan Konversi Nilai cLR  
Fungsi ini berguna untuk mengkonversi nilai ANT CO menjadi parameter kompleksitas SVR. Proses perhitungan konversi nilai cLR dapat dilihat pada Gambar 5.30

```

1 public double clr(double[][][] node, int ja, int iterAco, int
2 rangeClr) {
3     double result = 0, parClr;
4     parClr = ((node[iterAco][ja][0] * (rangeClr / 10)) +
5 (node[iterAco][ja][1] * (rangeClr / 100)) +
6 (node[iterAco][ja][2] * (rangeClr / 1000))) / rangeClr;
7     result = parClr;
8     return result; }

```

**Gambar 5.30 Implementasi algoritma perhitungan konversi nilai cLR**

Penjelasan Gambar 5.30 adalah sebagai berikut :

- Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki empat variabel parameter
  - Baris 3 merupakan inialisasi variabel
  - Baris 4-7 merupakan perhitungan konversi nilai cLR
  - Baris 8 merupakan fungsi pengembalian nilai *result method*
3. Fungsi Perhitungan Konversi Nilai Kompleksitas (*C*)  
Fungsi ini berguna untuk mengkonversi nilai ANT CO menjadi parameter kompleksitas SVR. Proses perhitungan konversi nilai kompleksitas (*C*) dapat dilihat pada Gambar 5.31

```

1 public double c(double[][][] node, int ja, int iterAco, int
2 rangeC) {
3     double result = 0, parC;
4     parC = (node[iterAco][ja][3] * (rangeC / 10)) +
5     (node[iterAco][ja][4] * (rangeC / 100)) +
6     (node[iterAco][ja][5] * (rangeC / 1000)) ;
7     result = parC;
8     return result;}

```

**Gambar 5.31 Implementasi algoritma perhitungan konversi nilai kompleksitas**

Penjelasan Gambar 5.31 adalah sebagai berikut :

- a. Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki empat variabel parameter
  - b. Baris 3 merupakan inialisasi variabel
  - c. Baris 4-7 merupakan perhitungan konversi nilai kompleksitas
  - d. Baris 8 merupakan fungsi pengembalian nilai *result method*
4. Fungsi Perhitungan Konversi Nilai *Epsilon* ( $\epsilon$ )

Fungsi ini berguna untuk mengkonversi nilai ANT CO menjadi parameter *epsilon* SVR. Proses perhitungan konversi nilai *Epsilon* ( $\epsilon$ ) dapat dilihat pada Gambar 5.32

```

1 public double epsilon(double[][][] node, int ja, int iterAco,
2 int rangeEps) {
3     double result = 0, parEps, eps;
4     parEps = ((node[iterAco][ja][6] * (rangeEps / 10)) +
5     (node[iterAco][ja][7] * (rangeEps / 100)) +
6     (node[iterAco][ja][8] * (rangeEps / 1000)) +
7     node[iterAco][ja][11])/10000;
8     result = parEps;
9     return result;
10 }

```

**Gambar 5.32 Implementasi algoritma perhitungan konversi nilai *epsilon***

Penjelasan Gambar 5.32 adalah sebagai berikut :

- a. Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki empat variabel parameter
  - b. Baris 3 merupakan inialisasi variabel
  - c. Baris 4-8 merupakan perhitungan konversi nilai *epsilon*
  - d. Baris 9-10 merupakan fungsi pengembalian nilai *result method*
5. Fungsi Perhitungan Konversi Nilai *Lamda* ( $\lambda$ )

Fungsi ini berguna untuk mengkonversi nilai ANT CO menjadi parameter *epsilon* SVR. Proses perhitungan konversi nilai *Lamda* ( $\lambda$ ) dapat dilihat pada Gambar 5.33

```

1 public double lamda(double[][][] node, int ja, int iterAco,
2 int rangeLamda) {
3     double result = 0, parLamda;
4     parLamda = ((node[iterAco][ja][9] * (rangeLamda / 10))
5 + (node[iterAco][ja][10] * (rangeLamda / 100)) +
6 (node[iterAco][ja][11] * (rangeLamda / 1000)) ) / rangeLamda;
7     result = parLamda;
8     return result;
9 }

```

**Gambar 5.33 Implementasi algoritma perhitungan konversi nilai *lamda***

Penjelasan Gambar 5.33 adalah sebagai berikut :

- a. Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki empat variabel parameter
  - b. Baris 3 merupakan inialisasi variabel
  - c. Baris 4-7 merupakan perhitungan konversi nilai *lamda*
  - d. Baris 8-9 merupakan fungsi pengembalian nilai *result*
6. Fungsi Perhitungan Konversi Nilai *Degree (d)*

Fungsi ini berguna untuk mengkonversi nilai ANT CO menjadi parameter *epsilon SVR*. Proses perhitungan konversi nilai *Degree (d)* dapat dilihat pada Gambar 5.34

```

1 public double degree(double[][][] node, int ja, int iterAco,
2 int rangeDegree) {
3     double result = 0, parDegree;
4     parDegree = (((node[iterAco][ja][12] * (rangeDegree /
5 10)) + (node[iterAco][ja][13] * (rangeDegree / 100)) +
6 (node[iterAco][ja][14] * (rangeDegree / 1000)))/100);
7     result = parDegree;
8     return result;
9 }

```

**Gambar 5.34 Implementasi algoritma perhitungan konversi nilai *degree***

Penjelasan Gambar 5.34 adalah sebagai berikut :

- a. Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki empat variabel parameter
  - b. Baris 3 merupakan inialisasi variabel
  - c. Baris 4-7 merupakan perhitungan konversi nilai *degree*
  - d. Baris 8-9 merupakan fungsi pengembalian nilai *result*
7. Fungsi Perhitungan Konversi Nilai *Gamma Kernel ( $\gamma$ )*

Fungsi ini berguna untuk mengkonversi nilai ANT CO menjadi parameter *epsilon SVR*. Proses perhitungan konversi nilai *Gamma Kernel ( $\gamma$ )* dapat dilihat pada Gambar 5.35

```

1 public double gamma(double[][][] node, int ja, int iterAco,
2 int rangeGamma) {
3     double result = 0, parGamma;
4     parGamma = ((node[iterAco][ja][15] * (rangeGamma /
5 10)) + (node[iterAco][ja][16] * (rangeGamma / 100)) +
6 (node[iterAco][ja][17] * (rangeGamma / 1000)) )/1;
7     result = parGamma;
8     return result;
9 }

```

**Gambar 5.35 Implementasi algoritma perhitungan konversi nilai *gamma kernel***

Penjelasan Gambar 5.35 adalah sebagai berikut :

- Baris 1-2 merupakan inialisasi nama *method*, dimana pada *method* ini memiliki empat variabel parameter
- Baris 3 merupakan inialisasi variabel
- Baris 4-7 merupakan perhitungan konversi nilai *gamma kernel*
- Baris 8-9 merupakan fungsi pengembalian nilai *result*

### 5.3 Implementasi Antarmuka

Implementasi antarmuka merupakan hasil dari pembahasan yang terdapat pada bab 4 mengenai perancangan. Pada halaman utama sistem pengguna harus memasukkan nilai iterasi SVR, iterasi ANT CO, jumlah semut, jumlah variabel lag, dan jumlah data training. Hasil dari peramalan akan ditampilkan pada panel pertama yang menunjukkan nilai parameter terbaik, semut terbaik dan generasi terbaik. Berikut pada Gambar 5.36 merupakan antarmuka dari halaman utama sistem dan hasil peramalan pada panel pertama.



**Gambar 5.36 Implementasi antarmuka panel pertama**

Pada panel kedua menampilkan hasil dari pelatihan dengan menggunakan parameter yang ditampilkan pada panel pertama. Berikut merupakan antarmuka dari hasil peramalan pada panel kedua yaitu hasil dari pelatihan yang ditunjukkan pada Gambar 5.37.

No	Aktual	Peramalan
1	88.0	87.99752274998407
2	77.0	76.999999999999955
3	7.0	7.002477250167098
4	48.0	48.0000000000004434
5	223.0	223.00000000001612
6	80.0	79.999999999999315
7	51.0	50.997522749949205
8	112.0	111.99752274984154

**Gambar 5.37 Implementasi antarmuka panel kedua**

Pada panel ketiga menampilkan hasil dari peramalan data uji dengan menggunakan parameter yang ditampilkan pada panel pertama. Berikut merupakan antarmuka dari hasil peramalan pada panel ketiga yaitu hasil dari pengujian yang ditunjukkan pada Gambar 5.38.

No	Aktual	Peramalan
1	53.0	54.131259978737624
2	77.0	98.90857526270709
3	85.0	95.90947922084231

**Gambar 5.38 Implementasi antarmuka panel ketiga**



Pada panel keempat menampilkan nilai RMSE tiap semut dan tiap generasinya. Berikut merupakan antarmuka dari hasil peramalan pada panel keempat yaitu nilai RMSE pada Gambar 5.39.

VONI TRI ASTUTIK-125150201111058

### PERAMALAN RUNUT WAKTU CURAH HUJAN MENGUNAKAN METODE SVR-ANT CO

ITERASI SVR :

ITERASI ANT CO :

JUMLAH SEMUT :

VARIABLE LAG :

DATA TRAINING :

**RMSE** : 14.145460473843245

**FITNESS** : 0.06602638471950298

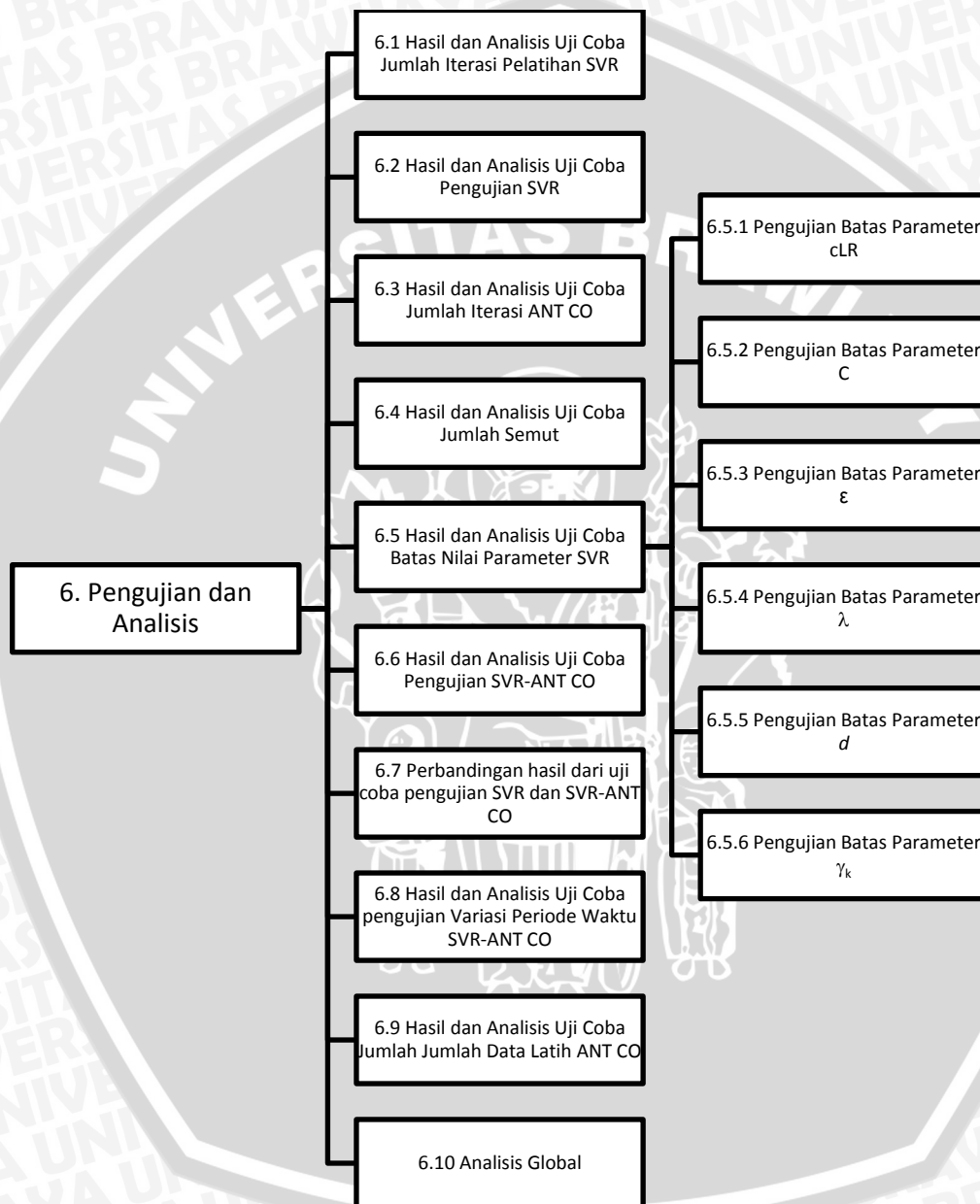
PARAMETER		TRAINING	TESTING	RMSE
No	Generasi	ANT		RMSE
241	9	1		0.057
242	9	2		0.002
243	9	3		0.002
244	9	4		0.002
245	9	5		0.001
246	9	6		0.002
247	9	7		0.002
248	9	8		0.002
249	9	9		0.003
250	9	10		0.003
251	9	11		0.003
252	9	12		0.001
253	9	13		0.001
254	9	14		0.002
255	9	15		0.002

Gambar 5.39 Implementasi antarmuka panel keempat



## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini membahas mengenai tahapan pengujian dan analisis dari implementasi algoritma *Support Vector Regression* (SVR) yang dioptimasi oleh algoritma *Ant Colony Optimization* (ANT CO) untuk meramalkan data runtu waktu curah hujan. Berikut Tahapan dari pengujian ditunjukkan pada Gambar 6.1



Gambar 6.1 Pengujian dan analisis

### 6.1 Hasil dan Analisis Uji Coba Jumlah Iterasi Pelatihan SVR

Uji coba yang dilakukan adalah menguji jumlah iterasi SVR. Data yang digunakan untuk uji coba jumlah iterasi SVR menggunakan data latih pada Tabel 4.16 dengan variabel *lag time* yang terdiri dari  $X_1$ ,  $X_2$ ,  $X_3$  dan  $Y$  merupakan data

aktual. Jumlah iterasi yang yang digunakan uji coba yaitu 10, 100, 500 dan 1000. Parameter-parameter yang digunakan untuk uji coba jumlah iterasi SVR adalah sebagai berikut:

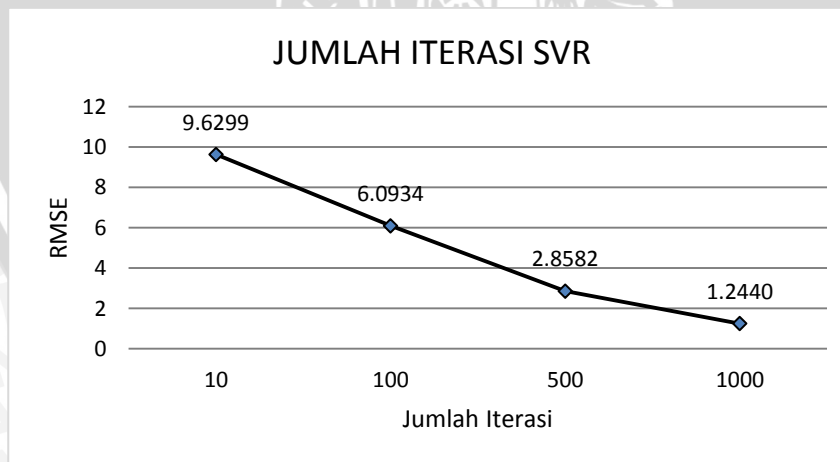
- Batas nilai parameter  $cLR$  : 0.01 – 1
- Batas nilai parameter  $C$  : 10 - 1000
- Batas nilai parameter  $\epsilon$  : 0.0001 – 0.01
- Batas nilai parameter  $\lambda$  : 0.01 - 1
- Batas nilai parameter  $degree (d)$  : 1- 10
- Batas nilai parameter  $gamma$  kernel ( $\gamma_k$ ): 1 – 100
- Panjang periode waktu ( $lag\ time$ ) : 3
- Jumlah data latih : 7
- Jumlah data uji : 3

Hasil uji coba jumlah iterasi SVR dapat diamati pada Tabel 6.1 berikut.

**Tabel 6.1 Hasil uji coba jumlah iterasi SVR**

Jumlah Iterasi	Nilai RMSE Percobaan ke- $i$										Rerata RMSE
	1	2	3	4	5	6	7	8	9	10	
10	2.5970	1.1766	1.6412	2.1787	0.5601	1.1269	54.5704	2.2494	28.6534	1.5448	9.6299
100	1.5147	1.5063	32.4636	1.4729	17.1018	0.8820	0.6475	1.7341	1.7098	1.9011	6.0934
500	1.2865	1.3396	1.5318	0.4107	0.4108	1.4444	2.0044	18.9546	0.7294	0.4701	2.8582
<b>1000</b>	<b>0.8945</b>	<b>1.0288</b>	<b>1.8405</b>	<b>0.8004</b>	<b>1.8111</b>	<b>1.4591</b>	<b>1.8679</b>	<b>0.5059</b>	<b>1.7878</b>	<b>0.4442</b>	<b>1.2440</b>

Berdasarkan hasil uji coba jumlah iterasi SVR pada Tabel 6.1 rerata RMSE yang memiliki nilai terkecil adalah pada iterasi ke-1000 dengan nilai 1.2440 sehingga dapat disimpulkan semakin banyak jumlah iterasi pada SVR, semakin baik nilai *lagrange multiplier* ( $\alpha, \alpha^*$ ) dan akan menghasilkan nilai peramalan yang mendekati nilai data aktual. Grafik hasil rerata fitness RMSE uji coba jumlah iterasi SVR dapat dilihat pada Gambar 6.2.



**Gambar 6.2 Grafik hasil uji coba jumlah iterasi SVR**

## 6.2 Hasil dan Analisis Uji Coba Pengujian SVR

Uji coba yang dilakukan merupakan pengujian SVR dengan menggunakan hasil uji coba iterasi SVR sebelumnya. Pengujian SVR menggunakan batas parameter sesuai penelitian yang dilakukan dengan mengambil batas parameter pada tabel

4.16. Hasil pengujian ditunjukkan pada Tabel 6.2. Parameter yang digunakan sebagai berikut :

- Batas nilai parameter  $cLR$  : 0.01 – 1
- Batas nilai parameter  $C$  : 10 - 1000
- Batas nilai parameter  $\epsilon$  : 0.0001 – 0.01
- Batas nilai parameter  $\lambda$  : 0.01 - 1
- Batas nilai parameter  $degree (d)$  : 1- 10
- Batas nilai parameter  $gamma$  kernel ( $\gamma_k$ ): 1 – 100
- Iterasi maksimum SVR :1000
- Panjang periode waktu ( $lag\ time$ ) : 3
- Jumlah data latih : 7
- Jumlah data uji : 3

**Tabel 6.2 Hasil Uji Coba Pengujian SVR**

Percobaan ke-	RMSE
1	77.7890
2	77.6031
3	77.4854
4	77.1265
5	76.4928
6	77.7948
7	77.7698
8	77.7859
9	77.7953
10	77.5130
<b>RMSE terbaik</b>	<b>76.4928</b>

Berdasarkan Tabel 6.2 hasil uji coba pada pengujian SVR menghasilkan nilai RMSE terbaik sebesar 76.4928 yang didapatkan dari percobaan ke 5.

### 6.3 Hasil dan Analisis Uji Coba Jumlah Iterasi ANT CO

Uji coba iterasi ANT CO memiliki 7 variasi jumlah iterasi yaitu 10, 20, 30, 50, 100 dan 200. Uji coba dilakukan sebanyak sepuluh kali, tiap variasi kemudian diambil nilai rerata RMSE dari sepuluh kali percobaan. Parameter yang digunakan sebagai berikut :

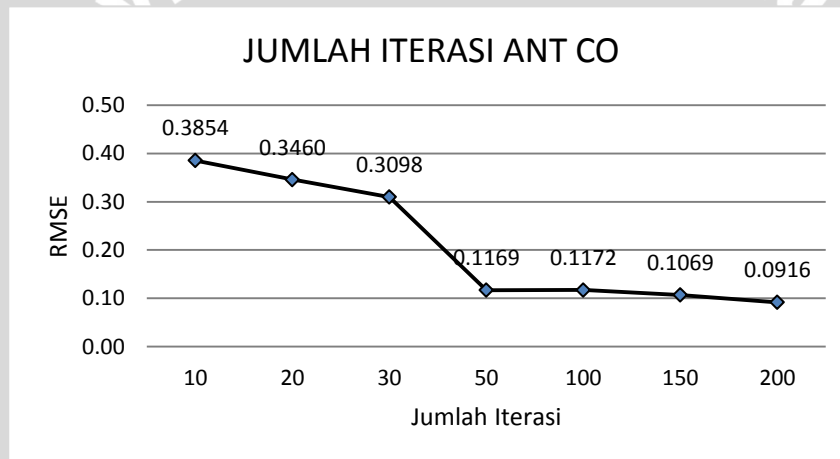
- Batas nilai parameter  $cLR$  : 0.01 – 1
- Batas nilai parameter  $C$  : 10 - 1000
- Batas nilai parameter  $\epsilon$  : 0.0001 – 0.01
- Batas nilai parameter  $\lambda$  : 0.01 - 1
- Batas nilai parameter  $degree (d)$  : 1- 10
- Batas nilai parameter  $gamma$  kernel ( $\gamma_k$ ): 1 – 100
- Jumlah semut : 7
- Iterasi maksimum SVR :1000
- Panjang periode waktu ( $lag\ time$ ) : 3
- Jumlah data latih : 7
- Jumlah data uji : 3

Hasil uji coba jumlah iterasi ANT CO dapat diamati pada Tabel 6.3 berikut.

**Tabel 6.3 Hasil uji coba jumlah iterasi ANT CO**

Jumlah Iterasi	Nilai RMSE Percobaan ke- <i>i</i>										Rerata RMSE
	1	2	3	4	5	6	7	8	9	10	
10	0.2925	0.2464	0.1565	0.2025	0.2340	1.0197	0.9422	0.2903	0.1575	0.3125	0.3854
20	0.5413	0.3988	0.2655	0.2759	0.9439	0.0532	0.0675	0.4500	0.2430	0.2209	0.3460
30	0.2633	0.0045	0.3128	0.2049	0.0405	0.0234	0.2512	1.9210	0.0675	0.0090	0.3098
50	0.2841	0.0062	0.2475	0.0473	0.0713	0.0022	0.0017	0.1688	0.2588	0.0810	0.1169
100	0.2519	0.0113	0.0625	0.2408	0.0900	0.0230	0.0293	0.2250	0.2267	0.0113	0.1172
150	0.2880	0.0968	0.1575	0.0361	0.0585	0.2610	0.0270	0.0250	0.1125	0.0062	0.1069
<b>200</b>	<b>0.0450</b>	<b>0.1778</b>	<b>0.1643</b>	<b>0.0450</b>	<b>0.0399</b>	<b>0.0315</b>	<b>0.1193</b>	<b>0.1468</b>	<b>0.1373</b>	<b>0.0089</b>	<b>0.0916</b>

Berdasarkan Tabel 6.3 hasil uji coba iterasi ANT CO memiliki nilai rerata RMSE terendah pada iterasi ke 200. Pada iterasi ke 200 hasil dari rerata RMSE sebesar 0.0916. Grafik hasil rerata RMSE uji coba jumlah iterasi ANT CO ditunjukkan pada Gambar 6.3.



**Gambar 6.3 Grafik hasil uji coba jumlah iterasi ANT CO**

#### 6.4 Hasil dan Analisis Uji Coba Jumlah Semut pada ANT CO

Uji coba jumlah semut pada ANT CO memiliki 6 variasi jumlah semut yaitu 5, 10, 15, 20, 25, dan 30. Uji coba dilakukan sebanyak sepuluh kali, tiap variasi kemudian diambil nilai rerata RMSE dari sepuluh kali percobaan. Parameter yang digunakan sebagai berikut :

- Batas nilai parameter *CLR* : 0.01 – 1
- Batas nilai parameter *C* : 10 - 1000
- Batas nilai parameter  $\epsilon$  : 0.0001 – 0.01
- Batas nilai parameter  $\lambda$  : 0.01 - 1
- Batas nilai parameter *degree (d)* : 1- 10
- Batas nilai parameter *gamma kernel ( $\gamma_k$ )* : 1 - 100
- Iterasi ANT CO : 200
- Iterasi maksimum SVR :1000
- Panjang periode waktu (*lag time*) : 3
- Jumlah data latihan : 7

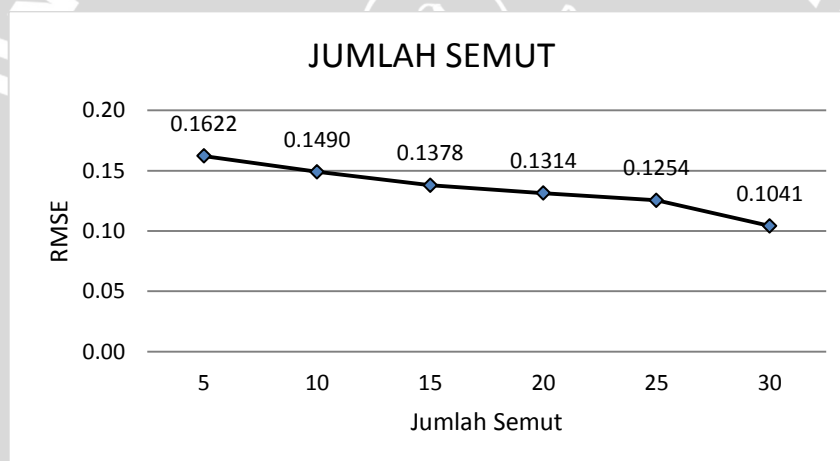
- Jumlah data uji : 3

Hasil uji coba jumlah semut pada ANT CO dapat diamati pada Tabel 6.4 berikut.

**Tabel 6.4 Hasil uji coba jumlah semut**

Jumlah Semut	Nilai RMSE Percobaan ke- <i>i</i>										Rerata RMSE
	1	2	3	4	5	6	7	8	9	10	
5	0.2925	0.0968	0.2475	0.2880	0.1148	0.0405	0.2250	0.1058	0.1553	0.0556	0.1622
10	0.0270	0.2700	0.2565	0.1305	0.0788	0.0473	0.2543	0.1148	0.2250	0.0855	0.1490
15	0.1508	0.2475	0.0046	0.2402	0.0045	0.1875	0.3105	0.0062	0.0981	0.1283	0.1378
20	0.0354	0.0021	0.0190	0.2475	0.2115	0.2723	0.2948	0.0450	0.0225	0.1643	0.1314
25	0.0395	0.0562	0.2760	0.1080	0.2790	0.0247	0.1125	0.2138	0.0225	0.1215	0.1254
30	<b>0.2588</b>	<b>0.0271</b>	<b>0.0042</b>	<b>0.1035</b>	<b>0.0652</b>	<b>0.0450</b>	<b>0.0518</b>	<b>0.2948</b>	<b>0.1710</b>	<b>0.0203</b>	<b>0.1041</b>

Berdasarkan Tabel 6.4 hasil uji coba jumlah semut pada ANT CO memiliki nilai rerata RMSE terendah pada jumlah semut 30. Pada jumlah semut 30 hasil dari rerata RMSE sebesar 0.1041. Grafik hasil rerata RMSE uji coba jumlah semut ditunjukkan pada Gambar 6.4.



**Gambar 6.4 Grafik hasil uji coba jumlah semut**

## 6.5 Hasil dan Analisis Uji Coba Batas Nilai Parameter SVR

Pengujian batas nilai parameter terdiri dari 6 bagian yaitu hasil dan analisis uji coba parameter  $cLR$ , hasil dan analisis uji coba parameter Kompleksitas ( $C$ ), hasil dan analisis uji coba parameter Epsilon ( $\epsilon$ ), hasil dan analisis uji coba parameter Lamda ( $\lambda$ ), hasil dan analisis uji coba parameter Degree ( $d$ ) dan hasil dan analisis uji coba parameter Gamma Kernel ( $\gamma_k$ ).

### 6.5.1 Hasil dan Analisis Uji Coba Batas Nilai Parameter $cLR$

Pengujian batas nilai parameter  $cLR$  dilakukan sebanyak sepuluh variasi batasan. Uji coba dilakukan sebanyak sepuluh kali, tiap variasi kemudian diambil nilai rerata RMSE dari sepuluh kali percobaan. Berikut detail parameter yang digunakan.

- Batas nilai parameter  $C$  : 10 - 1000

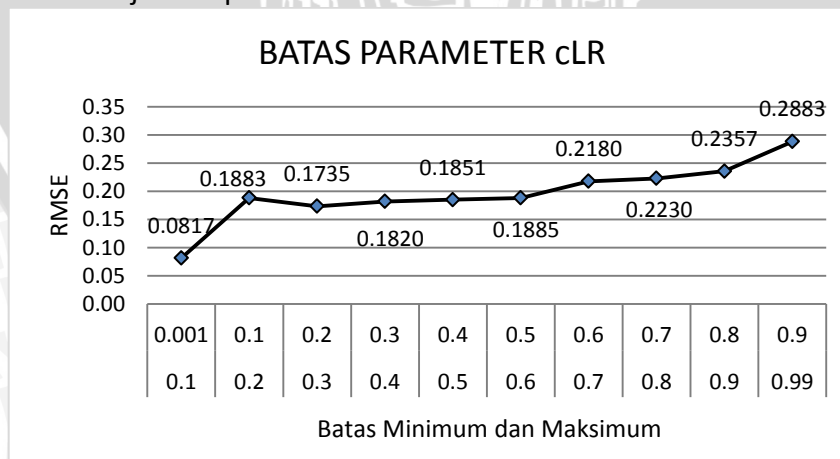
- Batas nilai parameter  $\epsilon$  : 0.0001 – 0.01
- Batas nilai parameter  $\lambda$  : 0.01 - 1
- Batas nilai parameter *degree* ( $d$ ) : 1- 10
- Batas nilai parameter *gamma* kernel ( $\gamma_k$ ): 1 - 100
- Iterasi maksimum SVR :1000
- Iterasi ANT CO : 200
- Jumlah semut : 30
- Panjang periode waktu (*lag time*) : 3
- Jumlah data latih : 7
- Jumlah data uji : 3

Hasil uji coba batas nilai parameter *cLR* dapat diamati pada Tabel 6.5 berikut.

**Tabel 6.5 Hasil uji coba batas nilai parameter *cLR***

Batas Min <i>cLR</i>	Batas Max <i>cLR</i>	Nilai RMSE Percobaan ke- <i>i</i>										Rerata RMSE
		1	2	3	4	5	6	7	8	9	10	
<b>0.001</b>	<b>0.1</b>	<b>0.2380</b>	<b>0.1553</b>	<b>0.0135</b>	<b>0.0225</b>	<b>0.0608</b>	<b>0.0022</b>	<b>0.0083</b>	<b>0.2310</b>	<b>0.0518</b>	<b>0.0333</b>	<b>0.0817</b>
0.11	0.2	0.2025	0.0135	0.2543	0.2835	0.2138	0.3083	0.2768	0.1328	0.1710	0.0270	0.1883
0.21	0.3	0.2813	0.2475	0.1643	0.0720	0.0630	0.1688	0.2475	0.2948	0.0405	0.1553	0.1735
0.31	0.4	0.0180	0.2475	0.2858	0.0517	0.0495	0.2745	0.2700	0.2722	0.1642	0.1868	0.1820
0.41	0.5	0.2520	0.2475	0.2833	0.0630	0.0135	0.2340	0.2092	0.2790	0.0540	0.2158	0.1851
0.51	0.6	0.2655	0.2700	0.1667	0.2655	0.0877	0.0293	0.2565	0.1093	0.1485	0.2858	0.1885
0.61	0.7	0.1880	0.2070	0.0248	0.1521	0.2700	0.2688	0.2970	0.2618	0.0720	0.4382	0.2180
0.71	0.8	0.0202	0.2594	0.2896	0.1964	0.1260	0.2722	0.2903	0.2452	0.2273	0.3038	0.2230
0.81	0.9	0.0585	0.2531	0.2655	0.2671	0.2587	0.2731	0.2250	0.2163	0.2813	0.2587	0.2357
0.91	0.99	0.5721	0.5850	0.0900	0.2585	0.2880	0.2880	0.2250	0.2475	0.3038	0.0248	0.2883

Berdasarkan Tabel 6.5 hasil uji coba batas nilai parameter *cLR* memiliki nilai rerata RMSE terendah pada batas minimum 0.001 dan maksimum 0.1 yang menghasilkan nilai rerata RMSE sebesar 0.0817, sehingga parameter *Learning Rate* (*cLR*) dengan batas nilai minimum 0.001 sampai dengan batas maksimal 0.1 dapat mengoptimalkan hasil peramalan. Grafik hasil rerata RMSE uji coba batas nilai parameter *cLR* ditunjukkan pada Gambar 6.5.



**Gambar 6.5 Grafik hasil uji coba batas parameter *cLR***

### 6.5.2 Hasil dan Analisis Uji Coba Batas Nilai Parameter C

Pengujian batas nilai parameter  $C$  dilakukan sebanyak empat variasi batasan. Uji coba dilakukan sebanyak sepuluh kali, tiap variasi kemudian diambil nilai rerata RMSE dari sepuluh kali percobaan. Berikut detail parameter yang digunakan.

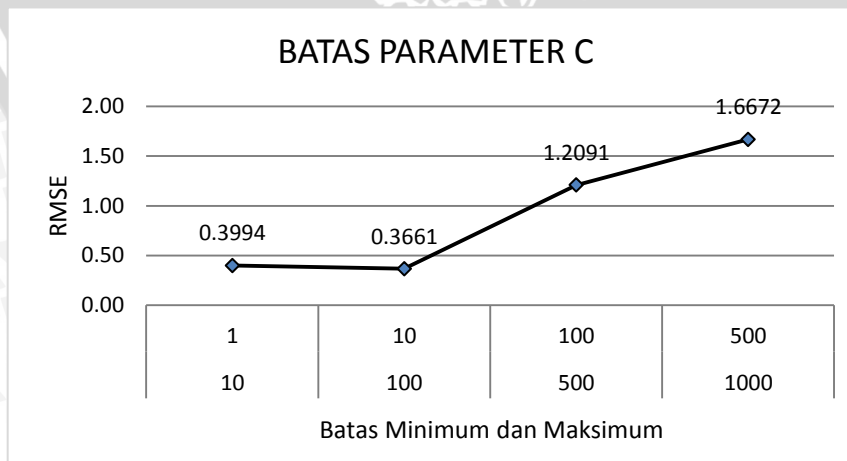
- Batas nilai parameter  $cLR$  : 0.001-0.1
- Batas nilai parameter  $\epsilon$  : 0.0001 – 0.01
- Batas nilai parameter  $\lambda$  : 0.01 - 1
- Batas nilai parameter  $degree (d)$  : 1- 10
- Batas nilai parameter  $gamma$  kernel ( $\gamma_k$ ): 1 - 100
- Iterasi maksimum SVR :1000
- Iterasi ANT CO : 200
- Jumlah semut : 30
- Panjang periode waktu ( $lag\ time$ ) : 3
- Jumlah data latih : 7
- Jumlah data uji : 3

Hasil uji coba jumlah batas nilai parameter  $C$  dapat diamati pada Tabel 6.6 berikut.

**Tabel 6.6 Hasil uji coba batas nilai parameter C**

Batas Min C	Batas Max C	Nilai RMSE Percobaan ke- $i$										Rerata RMSE
		1	2	3	4	5	6	7	8	9	10	
1	10	0.3141	0.3175	0.3753	0.1660	0.3613	0.3967	0.1480	0.3078	1.4462	0.1605	0.3994
11	100	<b>0.1955</b>	<b>0.7794</b>	<b>0.5443</b>	<b>0.3229</b>	<b>0.3406</b>	<b>0.1225</b>	<b>0.3432</b>	<b>0.3298</b>	<b>0.3283</b>	<b>0.3544</b>	<b>0.3661</b>
101	500	0.0542	1.7558	1.1777	0.0419	0.1004	1.6776	2.5356	2.2228	0.0601	2.4647	1.2091
501	1000	2.6997	2.8950	0.0542	1.0895	2.4893	2.6588	2.4741	0.4039	0.5017	1.4057	1.6672

Berdasarkan Tabel 6.6 hasil uji coba batas nilai parameter  $C$  memiliki nilai rerata RMSE terendah pada batas minimum 10 dan maksimum 100 yang menghasilkan nilai rerata RMSE sebesar 0.3661. Parameter  $C$  dengan batas nilai minimum 10 sampai dengan batas maksimal 100 merupakan batas optimal parameter  $C$  untuk peramalan curah hujan. Berdasarkan hasil uji batas nilai parameter  $C$ , maka data rerata RMSE batas nilai parameter  $C$  disajikan dalam bentuk grafik pada Gambar 6.6 berikut



**Gambar 6.6 Grafik hasil uji coba batas parameter C**



### 6.5.3 Hasil dan Analisis Uji Coba Batas Nilai Parameter $\epsilon$

Pengujian batas nilai parameter  $\epsilon$  dilakukan sebanyak lima variasi batasan. Uji coba dilakukan sebanyak sepuluh kali, tiap variasi kemudian diambil nilai rerata RMSE dari sepuluh kali percobaan. Berikut detail parameter yang digunakan.

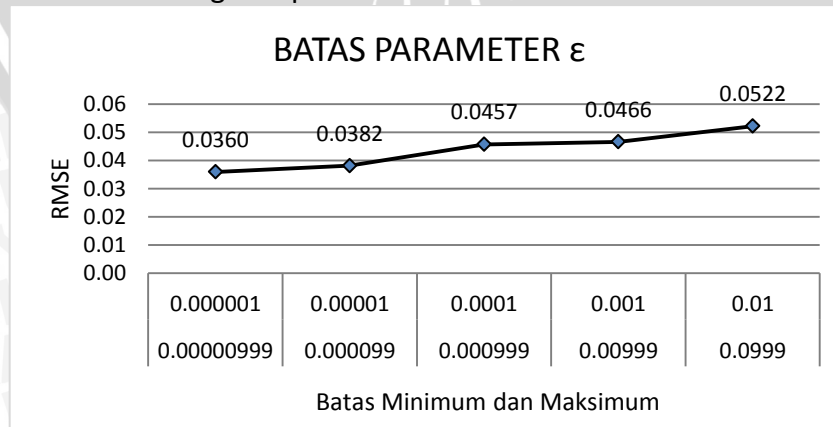
- Batas nilai parameter cLR : 0.001-0.1
- Batas nilai parameter C : 11-100
- Batas nilai parameter  $\lambda$  : 0.01 - 1
- Batas nilai parameter *degree* ( $d$ ) : 1- 10
- Batas nilai parameter *gamma* kernel ( $\gamma_k$ ): 1 - 100
- Iterasi maksimum SVR :1000
- Iterasi ANT CO : 200
- Jumlah semut : 30
- Panjang periode waktu (*lag time*) : 3
- Jumlah data latih : 7
- Jumlah data uji : 3

Hasil uji coba jumlah batas nilai parameter  $\epsilon$  ditunjukkan pada Tabel 6.7 berikut.

**Tabel 6.7 Hasil uji coba batas nilai parameter  $\epsilon$**

Batas Min $\epsilon$	Batas Max $\epsilon$	Nilai RMSE Percobaan ke- $i$										Rerata RMSE
		1	2	3	4	5	6	7	8	9	10	
0.000001	0.00000999	0.0339	0.0376	0.0375	0.0389	0.0351	0.0380	0.0349	0.0356	0.0351	0.0332	0.0360
0.00001	0.0000999	0.0343	0.0343	0.0361	0.0409	0.0405	0.0407	0.0378	0.0378	0.0404	0.0389	0.0382
0.0001	0.000999	0.0495	0.0432	0.0428	0.0459	0.0453	0.0440	0.0430	0.0510	0.0481	0.0441	0.0457
0.001	0.00999	0.0472	0.0538	0.0437	0.0430	0.0428	0.0539	0.0539	0.0439	0.0405	0.0438	0.0466
0.01	0.0999	0.0682	0.0587	0.0447	0.0611	0.0457	0.0553	0.0485	0.0474	0.0425	0.0497	0.0522

Berdasarkan Tabel 6.7 hasil uji coba batas nilai parameter  $\epsilon$  memiliki nilai rerata RMSE terendah pada batas minimum 0.000001 dan maksimum 0.00000999 yang menghasilkan nilai rerata RMSE sebesar 0.0360. Parameter  $\epsilon$  dengan batas nilai minimum 0.000001 sampai dengan batas maksimal 0.00000999 merupakan batas optimal parameter  $\epsilon$  untuk peramalan curah hujan. Berdasarkan hasil uji batas nilai parameter  $\epsilon$ , maka data rerata RMSE batas nilai parameter  $\epsilon$  ditunjukkan dalam bentuk grafik pada Gambar 6.7 berikut



**Gambar 6.7 Grafik hasil uji coba batas parameter  $\epsilon$**

### 6.5.4 Hasil dan Analisis Uji Coba Batas Nilai Parameter $\lambda$

Pengujian batas nilai parameter  $\lambda$  dilakukan sebanyak sepuluh variasi batasan. Uji coba dilakukan sebanyak sepuluh kali, tiap variasi kemudian diambil nilai rerata RMSE dari sepuluh kali percobaan. Berikut detail parameter yang digunakan.

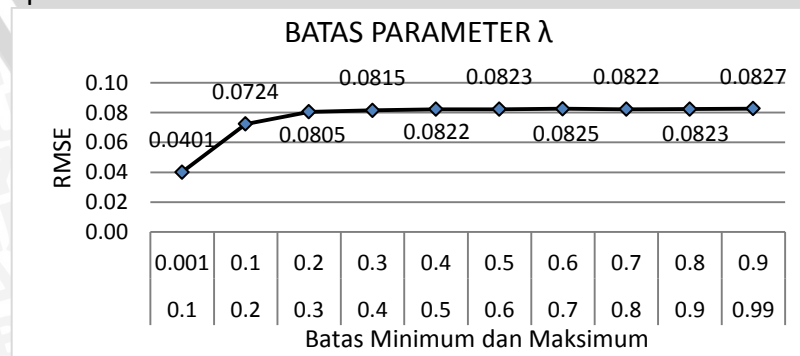
- Batas nilai parameter  $cLR$  : 0.001-0.1
- Batas nilai parameter  $C$  : 11-100
- Batas nilai parameter  $\epsilon$  : 0.000001 – 0.00000999
- Batas nilai parameter  $degree (d)$  : 1- 10
- Batas nilai parameter  $gamma$  kernel ( $\gamma_k$ ): 1 - 100
- Iterasi maksimum SVR :1000
- Iterasi ANT CO : 200
- Jumlah semut : 30
- Panjang periode waktu ( $lag\ time$ ) : 3
- Jumlah data latih : 7
- Jumlah data uji : 3

Hasil uji coba batas nilai parameter  $\lambda$  dapat diamati pada Tabel 6.8 berikut.

**Tabel 6.8 Hasil uji coba batas nilai parameter  $\lambda$**

Batas Min $\lambda$	Batas Max $\lambda$	Nilai RMSE Percobaan ke- $i$										Rerata RMSE
		1	2	3	4	5	6	7	8	9	10	
0.001	0.1	0.0421	0.0401	0.0401	0.0394	0.0390	0.0406	0.0404	0.0402	0.0394	0.0397	0.0401
0.1	0.2	0.0726	0.0718	0.0724	0.0725	0.0711	0.0731	0.0705	0.0741	0.0735	0.0725	0.0724
0.2	0.3	0.0801	0.0811	0.0798	0.0808	0.0803	0.0801	0.0811	0.0810	0.0804	0.0809	0.0805
0.3	0.4	0.0817	0.0816	0.0814	0.0817	0.0814	0.0813	0.0811	0.0816	0.0815	0.0815	0.0815
0.4	0.5	0.0818	0.0822	0.0824	0.0823	0.0822	0.0823	0.0833	0.0816	0.0825	0.0819	0.0822
0.5	0.6	0.0825	0.0822	0.0820	0.0822	0.0824	0.0826	0.0820	0.0825	0.0824	0.0816	0.0823
0.6	0.7	0.0825	0.0820	0.0826	0.0825	0.0829	0.0825	0.0827	0.0826	0.0819	0.0832	0.0825
0.7	0.8	0.0824	0.0821	0.0825	0.0821	0.0823	0.0826	0.0819	0.0824	0.0820	0.0821	0.0822
0.8	0.9	0.0823	0.0827	0.0819	0.0824	0.0829	0.0820	0.0821	0.0823	0.0822	0.0822	0.0823
0.9	0.99	0.0828	0.0826	0.0827	0.0823	0.0823	0.0828	0.0831	0.0828	0.0832	0.0825	0.0827

Berdasarkan Tabel 6.8 hasil uji coba batas nilai parameter  $\lambda$  memiliki nilai rerata RMSE terendah pada batas minimum 0.001 dan maksimum 0.1 yang menghasilkan nilai rerata RMSE sebesar 0.0401. Parameter  $\lambda$  dengan batas minimum 0.001 sampai dengan batas maksimal 0.1 merupakan batas optimal parameter  $\lambda$  untuk peramalan curah hujan. Berdasarkan hasil uji batas nilai parameter  $\lambda$ , maka data rerata RMSE batas nilai parameter  $\lambda$  disajikan dalam bentuk grafik pada Gambar 6.8.



Gambar 6.8 Grafik hasil uji coba batas parameter  $\lambda$

6.5.5 Hasil dan Analisis Uji Coba Batas Nilai Parameter  $d$

Pengujian batas nilai parameter *degree* ( $d$ ) dilakukan sebanyak sembilan variasi batasan. Uji coba dilakukan sebanyak sepuluh kali, tiap variasi kemudian diambil nilai rerata RMSE dari sepuluh percobaan. Berikut detail parameter yang digunakan.

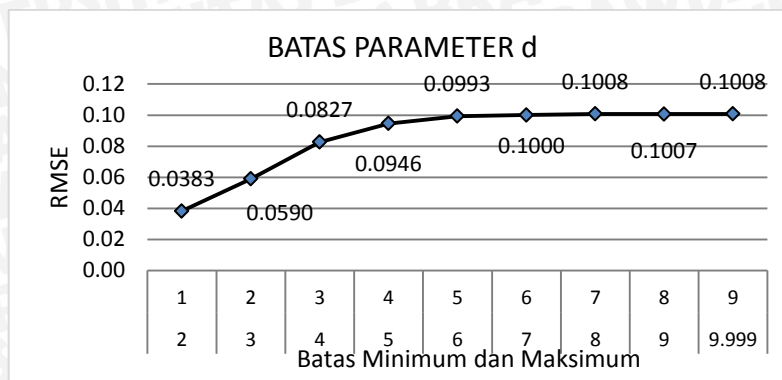
- Batas nilai parameter *CLR* : 0.001-0.1
- Batas nilai parameter *C* : 11-100
- Batas nilai parameter  $\epsilon$  : 0.000001 – 0.00000999
- Batas nilai parameter  $\lambda$  : 0.01- 0.1
- Batas nilai parameter *gamma* kernel ( $\gamma_k$ ): 1 - 100
- Iterasi maksimum SVR :1000
- Iterasi ANT CO : 200
- Jumlah semut : 30
- Panjang periode waktu (*lag time*) : 3
- Jumlah data latih : 7
- Jumlah data uji : 3

Hasil uji coba batas nilai parameter  $d$  dapat diamati pada Tabel 6.9 berikut.

Tabel 6.9 Hasil uji coba batas nilai parameter  $d$

Batas Min <i>degree</i>	Batas Max <i>degree</i>	Nilai RMSE Percobaan ke- <i>i</i>										Rerata RMSE
		1	2	3	4	5	6	7	8	9	10	
1	2	0.0352	0.0387	0.0395	0.0379	0.0385	0.0391	0.0388	0.0381	0.0388	0.0381	0.0383
2	3	0.0586	0.0583	0.0588	0.0594	0.0592	0.0587	0.0597	0.0584	0.0593	0.0598	0.0590
3	4	0.0825	0.0825	0.0820	0.0829	0.0817	0.0841	0.0828	0.0835	0.0821	0.0829	0.0827
4	5	0.0956	0.0943	0.0939	0.0941	0.0944	0.0951	0.0939	0.0949	0.0943	0.0952	0.0946
5	6	0.0990	0.0998	0.0994	0.0994	0.0993	0.0992	0.0993	0.0990	0.0992	0.0996	0.0993
6	7	0.1000	0.1001	0.1008	0.0998	0.1001	0.0999	0.0995	0.1003	0.1003	0.0996	0.1000
7	8	0.1005	0.1011	0.1003	0.1012	0.1010	0.1012	0.1007	0.1000	0.1008	0.1013	0.1008
8	9	0.1008	0.1008	0.1010	0.1006	0.1001	0.0998	0.1012	0.1006	0.1013	0.1010	0.1007
9	9.999	0.1018	0.1001	0.1012	0.1006	0.1008	0.1005	0.1020	0.1002	0.1006	0.1005	0.1008

Berdasarkan Tabel 6.9 hasil uji coba batas nilai parameter  $d$  memiliki nilai rerata RMSE terendah pada batas minimum 1 dan maksimum 2 yang menghasilkan nilai rerata RMSE sebesar 0.0383. Parameter  $d$  dengan batas nilai minimum 1 sampai dengan batas maksimal 2 merupakan batas optimal parameter  $d$  untuk perhitungan kernel anova. Berdasarkan hasil uji batas nilai parameter  $d$ , maka data rerata RMSE batas nilai parameter  $d$  disajikan dalam bentuk grafik pada Gambar 6.9



Gambar 6.9 Grafik pengujian batas parameter  $d$

### 6.5.6 Hasil dan Analisis Uji Coba Batas Nilai Parameter $\gamma_k$

Pengujian batas nilai parameter  $\gamma_k$  dilakukan sebanyak empat variasi batasan. Uji coba dilakukan sebanyak sepuluh kali, tiap variasi kemudian diambil nilai rerata RMSE dari sepuluh kali percobaan. Berikut detail parameter yang digunakan.

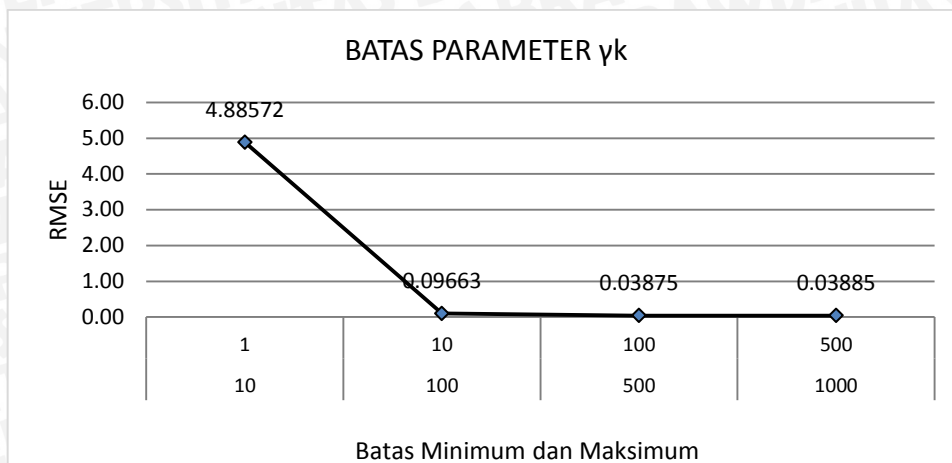
- Batas nilai parameter  $cLR$  : 0.001-0.1
- Batas nilai parameter  $C$  : 11-100
- Batas nilai parameter  $\epsilon$  : 0.000001 – 0.00000999
- Batas nilai parameter  $\lambda$  : 0.01- 0.1
- Batas nilai parameter  $degree (d)$  : 1 - 2
- Iterasi maksimum SVR :1000
- Iterasi ANT CO : 200
- Jumlah semut : 30
- Panjang periode waktu ( $lag\ time$ ) : 3
- Jumlah data latih : 7
- Jumlah data uji : 3

Hasil uji coba jumlah batas nilai parameter  $\gamma_k$  dapat diamati pada Tabel 6.10 berikut.

Tabel 6.10 Hasil uji coba batas nilai parameter  $\gamma_k$

Batas Min $\gamma_k$	Batas Max $\gamma_k$	Nilai RMSE Percobaan ke- $i$										Rerata RMSE
		1	2	3	4	5	6	7	8	9	10	
1	10	4.87339	4.77969	4.91793	4.88550	4.86566	4.88427	5.01702	4.84656	4.73884	5.04837	4.88572
11	100	0.09650	0.09660	0.09730	0.09617	0.09457	0.09784	0.09724	0.09901	0.09614	0.09498	0.09663
<b>100</b>	<b>500</b>	<b>0.03849</b>	<b>0.03851</b>	<b>0.03877</b>	<b>0.03894</b>	<b>0.03860</b>	<b>0.03838</b>	<b>0.03931</b>	<b>0.03842</b>	<b>0.03897</b>	<b>0.03913</b>	<b>0.03875</b>
500	1000	0.03814	0.03920	0.03831	0.03892	0.03939	0.03907	0.03843	0.03870	0.03926	0.03903	0.03885

Berdasarkan Tabel 6.10 hasil uji coba batas nilai parameter  $\gamma_k$  memiliki nilai rerata RMSE terendah pada batas minimum 100 dan maksimum 500 yang menghasilkan nilai rerata RMSE sebesar 0.03875. Parameter  $\gamma_k$  dengan batas nilai minimum 100 sampai dengan batas maksimal 500 merupakan batas optimal parameter  $\gamma_k$  untuk perhitungan kernel anova Berdasarkan hasil uji batas nilai parameter  $\gamma_k$ , maka data rerata RMSE batas nilai parameter  $\gamma_k$  disajikan dalam bentuk grafik pada Gambar 6.10 berikut



Gambar 6.10 Grafik pengujian batas parameter  $\gamma_k$

### 6.6 Hasil dan analisis uji coba pengujian SVR-ANT CO

Uji coba yang dilakukan adalah melakukan pengujian menggunakan metode SVR-ANT CO dengan parameter-parameter SVR yang dioptimasi, iterasi maksimum dari SVR dan ANT CO dan jumlah semut dari hasil pengujian yang terbaik. Parameter-parameter yang digunakan untuk uji coba pengujian SVR-ANT CO adalah sebagai berikut:

- Batas nilai parameter  $cLR$  : 0.001-0.1
- Batas nilai parameter  $C$  : 11-100
- Batas nilai parameter  $\epsilon$  : 0.000001 – 0.00000999
- Batas nilai parameter  $\lambda$  : 0.01- 0.1
- Batas nilai parameter  $degree (d)$  : 1 – 2
- Batas nilai parameter  $gamma$  kernel ( $\gamma_k$ ): 100-500
- Iterasi maksimum SVR :1000
- Iterasi ANT CO : 200
- Jumlah semut : 30
- Panjang periode waktu ( $lag\ time$ ) : 3
- Jumlah data latihan : 7
- Jumlah data uji : 3

Tabel 6.11 Pengujian SVR-ANT CO

Percobaan ke-	RMSE
1	46.5517
2	50.4654
3	36.0594
4	50.3254
5	54.5206
6	53.6125
7	57.2681
8	46.3915

9	51.7327
10	47.0575
<b>RMSE Terbaik</b>	<b>36.0594</b>

Berdasarkan Tabel 6.11 hasil uji coba pada pengujian SVR-ANT CO dengan menggunakan parameter terbaik dari hasil pengujian batas parameter menghasilkan nilai RMSE terbaik sebesar 36.0594 yang diperoleh dari percobaan ke 3.

## 6.7 Perbandingan Hasil dan Analisis Uji Coba Pengujian SVR dan Pengujian SVR-ANT CO

Perbandingan antara hasil dari uji coba pengujian SVR dengan pengujian parameter hasil optimasi SVR-ANT CO dilakukan untuk mengetahui nilai RMSE yang terkecil dari kedua pengujian tersebut. Hasil Perbandingan antara hasil dari uji coba pengujian SVR dengan pengujian parameter hasil optimasi SVR-ANT CO berdasarkan nilai RMSE ditunjukkan pada Tabel 6.12.

**Tabel 6.12 Hasil perbandingan uji coba pengujian SVR dan SVR-ANT CO**

Percobaan ke-	PENGUJIAN SVR	PENGUJIAN SVR-ANT CO
	RMSE	RMSE
1	77.7890	46.5517
2	77.6031	50.4654
3	77.4854	36.0594
4	77.1265	50.3254
5	76.4928	54.5206
6	77.7948	53.6125
7	77.7698	57.2681
8	77.7859	46.3915
9	77.7953	51.7327
10	77.5130	47.0575
<b>RMSE terbaik</b>	<b>76.4928</b>	<b>36.0594</b>

Berdasarkan Tabel 6.12 dapat disimpulkan nilai RMSE dari peramalan curah hujan di Daerah Poncokusumo pada dasarian 2 bulan Januari dari tahun 2000-2012 menggunakan metode SVR-ANT CO lebih baik dengan nilai RMSE terbaik 36.0594 dibandingkan dengan menggunakan metode SVR yang memiliki nilai RMSE terbaik 76.4928

## 6.8 Hasil dan Analisis Uji Coba Pengujian Variasi Periode Waktu (*Lag Time*) SVR-ANT CO

Pengujian variabel lag dilakukan sebanyak tujuh variasi batasan. Uji coba dilakukan sebanyak sepuluh kali, tiap variasi kemudian diambil nilai rerata RMSE dari sepuluh kali percobaan. Berikut detail data dan parameter yang digunakan.

- Batas nilai parameter  $cLR$  : 0.001-0.1
- Batas nilai parameter  $C$  : 11-100
- Batas nilai parameter  $\epsilon$  : 0.000001 – 0.00000999
- Batas nilai parameter  $\lambda$  : 0.01- 0.1
- Batas nilai parameter *degree* ( $d$ ) : 1 – 2

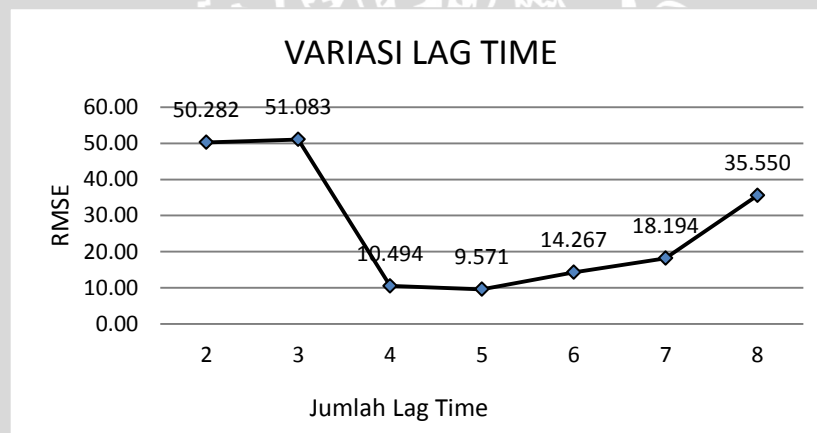
- Batas nilai parameter  $\gamma$  kernel ( $\gamma_k$ ): 100-500
- Iterasi maksimum SVR :1000
- Iterasi ANT CO : 200
- Jumlah semut : 30
- Jumlah data uji : 3

Hasil uji coba variasi variabel lag dapat diamati pada Tabel 6.10 berikut.

**Tabel 6.13 Hasil uji coba pengujian variasi variabel lag**

Variabel Lag	Nilai RMSE Percobaan ke- <i>i</i>										Rerata RMSE
	1	2	3	4	5	6	7	8	9	10	
2	50.282	54.249	65.667	57.480	59.241	55.180	50.813	51.356	52.191	55.180	50.282
3	51.083	60.728	59.976	64.359	50.546	61.112	62.291	13.286	61.500	43.643	51.083
4	10.494	17.149	31.680	16.361	24.773	14.699	10.820	32.670	39.323	18.011	10.494
<b>5</b>	<b>9.571</b>	<b>9.941</b>	<b>12.228</b>	<b>11.092</b>	<b>37.610</b>	<b>39.650</b>	<b>10.976</b>	<b>12.245</b>	<b>38.526</b>	<b>13.837</b>	<b>9.571</b>
6	14.267	21.727	14.244	18.084	15.474	13.948	13.749	14.974	13.728	15.393	14.267
7	18.194	28.851	24.510	17.051	29.769	15.722	15.807	15.367	15.779	18.455	18.194
8	36.351	32.818	37.808	31.203	30.757	30.333	30.957	43.893	44.649	36.733	35.550

Berdasarkan Tabel 6.10 hasil uji coba variasi variabel lag memiliki nilai rerata RMSE terendah pada jumlah lag sebesar 6 yang menghasilkan nilai rerata RMSE sebesar 9.5721. Berdasarkan hasil uji variasi variabel lag, maka data rerata RMSE variasi variabel lag disajikan dalam bentuk grafik pada Gambar 6.9 berikut



**Gambar 6.11 Grafik pengujian variasi variabel lag**

### 6.9 Hasil dan Analisis Uji Coba Variasi Jumlah Data Latih SVR-ANT CO

Pengujian variasi jumlah data latih dilakukan sebanyak tujuh variasi batasan. Uji coba dilakukan sebanyak sepuluh kali, tiap variasi kemudian diambil nilai rerata RMSE dari sepuluh kali percobaan. Berikut detail data dan parameter yang digunakan.

- Batas nilai parameter  $CLR$  : 0.001-0.1
- Batas nilai parameter  $C$  : 11-100
- Batas nilai parameter  $\epsilon$  : 0.000001 – 0.00000999
- Batas nilai parameter  $\lambda$  : 0.01- 0.1
- Batas nilai parameter  $degree (d)$  : 1 – 2
- Batas nilai parameter  $\gamma$  kernel ( $\gamma_k$ ): 100-500

- Iterasi maksimum SVR :1000
- Iterasi ANT CO : 200
- Jumlah semut : 30
- Panjang periode waktu (*lag time*) : 5

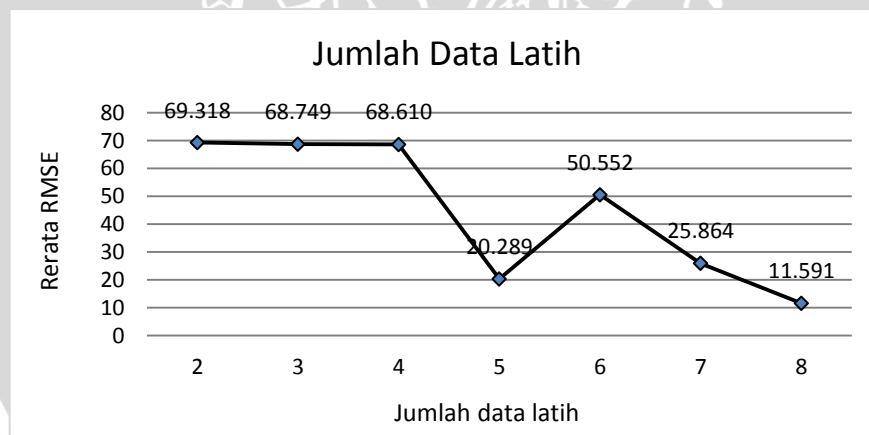
Hasil uji coba variasi variabel lag dapat diamati pada Tabel 6.10 berikut.

Hasil uji coba variasi data latih dapat diamati pada Tabel 6.11 berikut.

**Tabel 6.14 Hasil uji coba variasi jumlah data latih**

Variasi Data Latih	Nilai RMSE Percobaan ke- <i>i</i>										Rerata RMSE
	1	2	3	4	5	6	7	8	9	10	
2	63.935	57.480	75.923	79.000	59.976	78.365	72.529	58.524	68.444	79.000	69.318
3	68.444	68.930	68.444	68.444	69.922	67.966	69.922	69.423	69.423	66.568	68.749
4	67.966	66.568	67.493	69.423	69.423	68.444	70.429	70.429	67.966	67.966	68.610
5	21.371	19.492	18.802	22.585	17.975	18.011	22.753	18.084	21.883	21.936	20.289
6	46.170	65.667	73.627	52.763	45.296	35.496	47.544	49.251	42.860	46.847	50.552
7	17.975	18.011	22.753	18.084	35.496	47.544	11.755	33.602	28.240	25.178	25.864
8	<b>9.482</b>	<b>12.423</b>	<b>9.858</b>	<b>11.953</b>	<b>11.469</b>	<b>10.862</b>	<b>9.753</b>	<b>18.417</b>	<b>11.107</b>	<b>10.587</b>	<b>11.591</b>

Berdasarkan Tabel 6.11 hasil uji coba variasi data latih memiliki nilai rerata RMSE terendah pada jumlah data latih sebesar 8 yang menghasilkan nilai rerata RMSE sebesar 11.591. Berdasarkan hasil uji variasi data latih, maka data rerata RMSE variasi data latih disajikan dalam bentuk grafik pada Gambar 6.11 berikut



**Gambar 6.12 Grafik pengujian variasi data latih**

## 6.10 Analisis Global

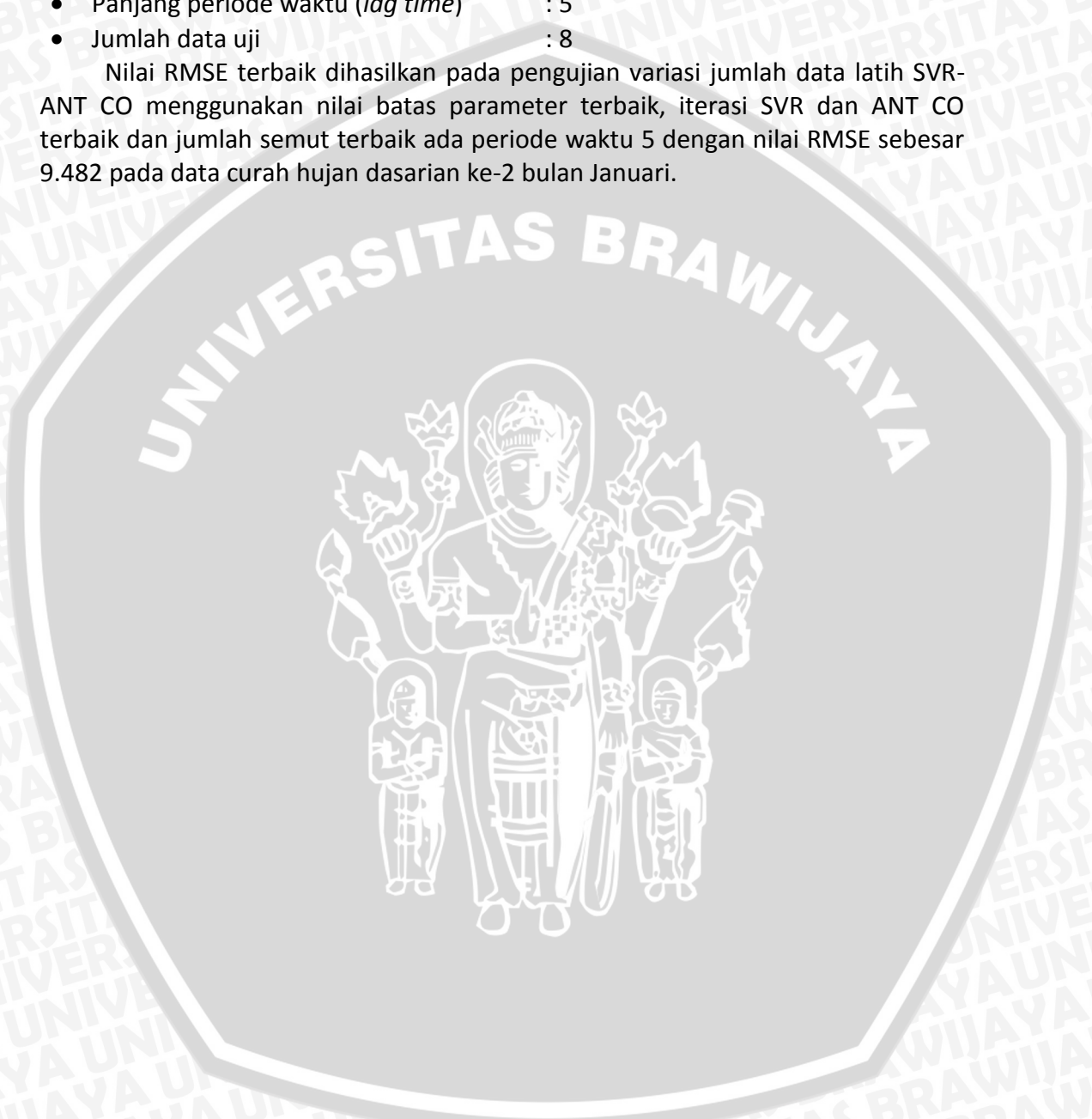
Berdasarkan hasil keseluruhan pengujian yang telah dilakukan, berikut parameter terbaik yang digunakan pada peramalan dasarian curah hujan dasarian ke-2 bulan Januari di wilayah Poncokusumo

- Iterasi maksimum SVR :1000
- Iterasi ANT CO : 200
- Jumlah semut : 30
- Batas nilai parameter *cLR* : 0.001-0.1



- Batas nilai parameter  $C$  : 11-100
- Batas nilai parameter  $\epsilon$  : 0.000001 – 0.00000999
- Batas nilai parameter  $\lambda$  : 0.01- 0.1
- Batas nilai parameter *degree* ( $d$ ) : 1 – 2
- Batas nilai parameter *gamma* kernel ( $\gamma_k$ ): 100-500
- Panjang periode waktu (*lag time*) : 5
- Jumlah data uji : 8

Nilai RMSE terbaik dihasilkan pada pengujian variasi jumlah data latih SVR-ANT CO menggunakan nilai batas parameter terbaik, iterasi SVR dan ANT CO terbaik dan jumlah semut terbaik ada periode waktu 5 dengan nilai RMSE sebesar 9.482 pada data curah hujan dasarian ke-2 bulan Januari.



## BAB 7 PENUTUP

Bab ini membahas kesimpulan dan saran terkait penelitian runut waktu curah hujan menggunakan metode *Support Vector Regression* dengan *Ant Colony Optimization*.

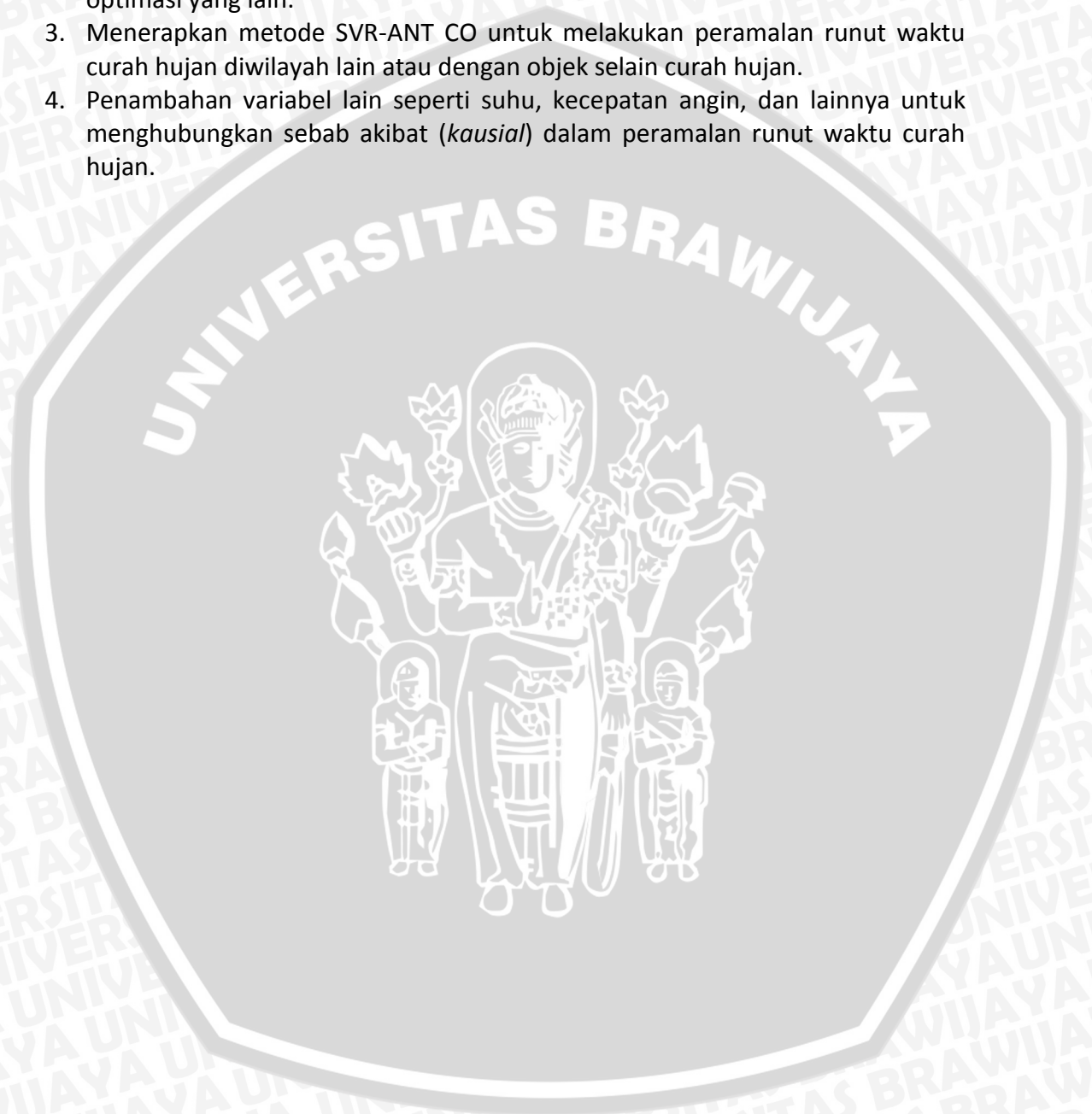
### 7.1 Kesimpulan

Berdasarkan hasil penelitian mengenai implementasi *Ant Colony Optimization* (ANT CO) untuk optimasi peramalan data runut waktu curah hujan menggunakan *Support Vector Regression* (SVR) dapat ditarik beberapa kesimpulan yaitu sebagai berikut:

1. Peramalan runut waktu curah hujan diperlukan strategi peramalan untuk dapat melakukan peramalan terstruktur dengan baik. Strategi peramalan meliputi definisi masalah, pengumpulan data, analisis data, pemilihan metode ramalan (SVR-ANT CO) dan validasi model ramalan. Dalam melakukan peramalan runut waktu curah hujan secara kuantitatif dapat menerapkan strategi peramalan kedalam sebuah sistem berupa perangkat lunak untuk mempermudah dan mempercepat waktu dalam hal komputasi
2. Algoritma *Ant Colony Optimization* (ANT CO) dapat diimplementasikan untuk optimasi peramalan data runut waktu curah hujan menggunakan metode *Support Vector Regression* (SVR) dengan menemukan koefisien-koefisien terbaik untuk setiap parameter SVR yang dioptimasi. Dalam penelitian ini terdapat enam parameter metode SVR yang dioptimasi menggunakan ACO diantaranya yaitu nilai  $cLR$ , nilai kompleksitas ( $C$ ), nilai  $epsilon$  ( $\epsilon$ ), nilai  $lamda$  ( $\lambda$ ), nilai  $degree$  ( $d$ ), dan nilai  $gamma$  kernel ( $\gamma_k$ ). Adapun batas rentang nilai yang dibangkitkan ANT CO untuk setiap parameter tersebut berbeda-beda, batas rentang untuk nilai  $cLR$  yaitu antara 0.001 sampai 0.1, batas nilai parameter  $C$  antara 11 sampai 100, batas nilai parameter  $\epsilon$  antara 0.000001 sampai 0.00000999, batas nilai parameter  $\lambda$  antara 0.01 sampai 0.1, batas nilai parameter  $degree$  antara 1 sampai 2, batas nilai parameter  $\gamma_k$  antara 100 sampai 500. Variasi variabel lag yang digunakan adalah 5 dengan jumlah data latih sebanyak 8 data.
3. Evaluasi kinerja peramalan runut waktu curah hujan menggunakan metode SVR-ANT CO mengacu pada nilai RMSE. Nilai evaluasi kinerja peramalan diperoleh dari hasil perbandingan nilai aktual dengan hasil ramalan. Semakin rendah nilai RMSE, maka semakin baik kinerja peramalan. Berdasarkan hasil pengujian, metode SVR-ANT CO dapat melakukan peramalan runut waktu curah hujan pada bulan Januari Dasarian ke-2 dengan nilai RMSE terendah sebesar 9.482 .

## 7.2 Saran

1. Penambahan data runtu waktu curah hujan, semakin banyak data runtu waktu akan berpengaruh pada proses pelatihan SVR dalam mengenali pola data curah hujan.
2. Mengoptimalkan parameter metode SVR dengan strategi atau algoritma optimasi yang lain.
3. Menerapkan metode SVR-ANT CO untuk melakukan peramalan runtu waktu curah hujan di wilayah lain atau dengan objek selain curah hujan.
4. Penambahan variabel lain seperti suhu, kecepatan angin, dan lainnya untuk menghubungkan sebab akibat (*kausal*) dalam peramalan runtu waktu curah hujan.



## DAFTAR PUSTAKA

- Abbot, J. & Marohasy, J., 2014. Input selection and optimisation for monthly rainfall forecasting in Queensland, Australia, using artificial neural networks. *Atmospheric Research*, Volume 138, pp. 166-178.
- Abraham, B. & Ledolter, J., 2005. *Statistical Methods for Forecasting*. New jersey: John Wiley & Sons, Inc.
- Adhani, G., Buono, A. & Faqih, A., 2013. Support Vector Regression modelling for rainfall prediction in dry season based on Southern Oscillation Index and NINO3. 4. *In Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, September. pp. 315-320.
- Adhani, G., Buono, A. & Faqih, A., 2014. Optimization of Support Vector Regression using Genetic Algorithm and Particle Swarm Optimization for Rainfall Prediction in Dry Season. *ELKOMNIKA Indonesian Journal of Electrical Engineering*, November, Volume 12(11), pp. 7912-7919.
- Aldrian, E. & Djamil, Y. S., 2008. Spatio-temporal climatic change of. *International Journal of Climatology*, Volume 28, pp. 435-448.
- Aldrian, E. & Dwi Susanto, R., 2003. Identification of three dominant rainfall regions within Indonesia and their relationship to sea surface temperature. *International Journal of Climatology*, Volume 23(12), pp. 1435-1452.
- Alwee, R., Hj Shamsuddin, S. & Sallehuddin, R., 2013. Hybrid support vector regression and autoregressive integrated moving average models improved by particle swarm optimization for property crime rates forecasting with economic indicators. *The Scientific World Journal*, 8 May. Volume 2013.
- Ariyasingha, I. & Fernando, T., 2015. Performance analysis of the multi-objective ant colony optimization algorithms for the traveling salesman problem. *Swarm and Evolutionary Computation*, Volume 23, pp. 11-26.
- Atole, D. P., Kapgate, D. & Longadge, R., 2015. Statistical Modeling for Rainfall Prediction using Data Mining Technique. *ABHIYANTRIKI: An International Journal of Engineering & Technology*, January, Volume 2, pp. 35-39.
- Balamurugan, M. & R.S, S., 2014. A Survey On Rainfall Prediction Using Data Mining. *International Journal of Computer Science and Mobile Applications*, Februari, 2(2), pp. 84-88.
- Basak, D. P. S. & Patranabis, D., 2007. Support vector regression. *Neural Information Processing-Letters and Reviews*, Volume 11(10), pp. 203-224.
- Ben-Hur, A. & Weston, J., 2010. A user's guide to support vector machines. *Data mining techniques for the life sciences*, pp. 223-239.
- Berk, R. et al., 2014. Using regression kernels to forecast a failure to appear in court. 5 September.

Bertsimas, D. & Shioda, R., 2007. Classification and Regression via Integer Optimization. *OPERATIONSRESEARCH*, Volume 55, pp. 252-271.

Bi, L., Tsimhoni, O. & Liu, Y., 2011. Using the Support Vector Regression Approach. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART A: SYSTEMS AND HUMANS*, MAY. Volume 41( 3).

BMKG Stasiun Klimatologi Karangploso, 2015. Analisis Hujan November 2015 dan Prakiraan Hujan Januari - Maret 2016. In: *Analisis Hujan November 2015 dan Prakiraan Hujan Januari - Maret 2016*. Malang(Jawa Timur): s.n., p. 52.

BMKG Stasiun Klimatologi Karangploso, 2015. Prakiraan Musim Hujan Tahun 2015/2016 Provinsi Jawa Timur. In: *Prakiraan Musim Hujan Tahun 2015/2016 Provinsi Jawa Timur*. Malang(Jawa Timur): s.n., p. 40.

BMKG, 2012. *Daftar Istilah Klimatologi*. [Online] Available at: <http://www.bmkg.go.id/daftar-istilah-musim> [Accessed 28 Januari 2016].

Boryczka, U. & Kozak, J., 2015. Enhancing the effectiveness of Ant Colony Decision Tree algorithms by co-learning. *Applied Soft Computing*, Volume 30, pp. 166-178.

Burbidge, R. & Buxton, B., n.d. An Introduction to Support Vector Machines for Data Mining. *Computer Science Dept., UCL, Gower Street, WC1E 6BT, UK.*

Cahyani, M. P., Dewi, C. & Yudistira, N., 2014. Implementasi Algoritma Al-Alaoui Backpropagation dalam peramalan curah hujan. *Repositori Jurnal Mahasiswa PTIK UB*, Volume 3(2).

Chang, F.-J. et al., 2013. Watershed rainfall forecasting using neuro-fuzzy networks with the assimilation of multi-sensor information. *Journal of Hydrology*, 508(2014), p. 374-384.

Chang, P.-C., Wang, Y.-W. & Liu, C.-H., 2007. The development of a weighted evolving fuzzy neural network for PCB sales forecasting. *Expert Systems with Applications*, p. 86-96.

Chen, N., Lu, W., Yang, J. & Li, G., 2004. *Support vector machine in chemistry*. s.l.:World Scientific Publishing Co. Pte. Ltd.

Cortes, C. & Vapnik, V., 1995. Support-vector networks. *Machine learning*, , Volume 20(3), pp. 273-297..

Crone, S., Guajardo, J. & Weber, R., 2006. The impact of preprocessing on support vector regression and neural networks in time series prediction. *DMIN*, pp. 37-44.

Crone, S., Lessmann, S. & Pietsch, S., 2006. Forecasting with Computational Intelligence - An Evaluation of Support Vector Regression and Artificial Neural Networks for Time Series Prediction. *International Joint Conference on Neural Networks*, 16-21 July .pp. 3159-3166.

Deng, N., Tian, Y. & Zhang, C., 2012. *Support vector machines: optimization based theory, algorithms, and extensions*. Minneapolis, Minnesota, U.S.A.: CRC press.

Department Of Treasury, 2008. Forecasting accuracy of the act budget estimates. May.

Diani, K., Setiawan, S. & Suhartono, S., 2013. Pemodelan VAR-NN dan GSTAR-NN untuk Peramalan Curah Hujan di Kabupaten Malang. *Jurnal Sains dan Seni ITS*, Volume 2(1), pp. D31-D36.

Dorigo, M., Birattari, M. & Stutzle, T., 2006. Ant Colony Optimization : Artificial Ants as a Computational Intelligence Technique. *IEEE Computational*, Volume X, pp. 28-39.

Dounia, M., Sabr, D. & Yassine, D., 2014. Rainfall- Rain off Modeling Using Artificial Neural Network. *APCBEE Procedia*, Volume 10, pp. 251-256.

Drucker, H. et al., 1997. Support Vector Regression Machines. *Advances in neural information processing systems*, p. 155–161.

Dube, A. et al., 2014. Forecasting the heavy rainfall during Himalayan flooding—June 2013. *Weather and Climate Extremes*, Volume 4, pp. 22-34.

El Margae, S., Sanae, B., Mounir, A. & Youssef, F., 2014. Traffic Sign Recognition based on multi-block LBP features using SVM with normalization. *Intelligent Systems: Theories and Applications (SITA-14), 2014 9th International Conference*, May.pp. 1-7.

Engelbrecht, A. P., 2007. *Computational Intelligence: An Introduction 2nd ed.* 2 ed. s.l.:West Sussex: John Wiley & Sons Ltd.

Estiningtyas, W., Ramadhani, F. & Aldrian, E., 2007. Analisis Korelasi Curah Hujan dan Suhu Permukaan Laut Wilayah Indonesia, Serta Implikasinya Untuk Prakiraan Curah Hujan (Studi kasus Kabupaten Cilacap). *J. Agromet Indonesia* , 21(2), pp. 46-60.

Fan, G.-F., Peng, L.-L., Hong, W.-C. & Sun, F., 2016. Electric load forecasting by the SVR model with differential empirical mode decomposition and auto regression. *Neurocomputing*, Volume 173, pp. 958-970.

Furi, R. P., Jondri & Saepudin, D., 2015. Prediksi Financial Time Series Menggunakan Independent Component Analysis dan Support Vector Regression Studi Kasus : IHSB dan JII. *Fakultas Informatika Telkom University, Bandung*.

Geng, J., Huang, M.-L., Li, M.-W. & Hong, W.-C., 2015. Hybridization of seasonal chaotic cloud simulated annealing algorithm in a SVR-based load forecasting model. *Neurocomputing*, Volume 151, p. 1362–1373.

Gunn, S., 1998. Support vector machines for classification and regression. *ISIS technical report*, 10 May. Volume 14.

Han, J., Kamber, M. & Pei, J., 2011. Data mining: concepts and techniques. In: 3, ed. *The Morgan Kaufmann Series in Data Management Systems*. s.l.:Elsevier.

Hasan, N., Nath, N. & Rasel, R., 2015. A support vector regression model for forecasting rainfall. *Electrical Information and Communication Technology (EICT), 2015 2nd International Conference on*, December..pp. 554-559.

He, X., Guan, H. & Qin, i., 2015. A hybrid wavelet neural network model with mutual information and particle swarm optimization for forecasting monthly rainfall. *ournal of Hydrology*, 527(2015), p. 88–100.

Hong, W. C., Dong, Y., Chen, L. Y. & Lai, C. Y., 2010. Taiwanese 3G Mobile Phone Demand Forecasting by SVR with Hybrid Evolutionary Algorithms. *Expert Systems with Applications*, Volume 37, p. 452–4462.

Hong, W. C., Dong, Y., Chen, L. Y. & Wei, S. Y., 2011. SVR with Hybrid Chaotic Genetic Algorithms For Tourism Demand Forecasting. *Applie d Sof t Computing*, Volume 11, p. 1881–1890.

Hong, W. C. et al., 2013. Cyclic Electric Load Forecasting by Seasonal SVR with Chaotic Genetic Algorithm. *Electrical Power and Energy Systems*, Volume 44, p. 604–614.

Hong, W.-C., Chen, Y.-F., Chen, P.-W. & Yeh, Y.-H., 2007. Continuous ant colony optimization algorithms in a support vector regression based financial forecasting model. *IEEE*.

Hong, W.-C., Dong, Y., Zheng, F. & Lai, C.-Y., 2011. Forecasting Urban Traffic Flow by SVR with Continuous ACO. *Applied Mathematical Modelling*, Volume 35 , p. 1282–1291.

Hong, W. & Pai, P., 2007. Potential assessment of the support vector regression technique in rainfall forecasting. *Water Resources Management*, Volume 21(2), pp. 495-513.

Hu, L., Che, X. & Cheng, X. 2., 2010. Bandwidth prediction based on nu-support vector regression and parallel hybrid particle swarm optimization. *International Journal of Computational Intelligence Systems*, April, Volume 3(1), pp. 70-83.

Hung, W.-M. & Hong, W.-C., 2009. Application of SVR with improved ant colony optimization algorithms in exchange rate forecasting. *Control and Cybernetics*, 29(3), pp. 863-891.

Hutasuhut, A. H., Anggraeni, W. & Tyasnurita, R., 2014. Pembuatan Aplikasi Pendukung Keputusan Untuk Peramalan Persediaan Bahan Baku Produksi Plastik Blowing dan Inject Menggunakan Metode ARIMA (Autoregressive Integrated Moving Average) Di CV. Asia. *JURNAL TEKNIK POMITS*, 3(4), pp. 169-174.

Hyndman, R. J. & Athanasopoulos, G., 2014. *Forecasting: principles and practice*. s.l.:s.n.

Hyndman, R. & Koehler, A., 2006. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4), pp. 679-688.

Indrabayu, et al., 2012. Prediksi Curah Hujan Dengan Fuzzy Logic. *Prosiding Hasil Penelitian Fakultas Teknik*, Desember. Volume 6.

Ju, F.-Y. & Hong, W.-C., 2013. Application of seasonal SVR with chaotic gravitational search algorithm in electricity forecasting. *Applied Mathematical Modelling*, Volume 37, p. 9643–9651.

Junaedi, H., Budiarto, H., Maryati, I. & Melani, Y., 2011. DATA TRANSFORMATION PADA DATA MINING. *Prosiding Konferensi Nasional "Inovasi dalam Desain dan Teknologi" - IDEaTech*.

Kavousi-Fard, A., Samet, H. & Marzbani, F., 2014. A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short Term Load Forecasting. *Expert Systems with Applications*, p. 6047–6056.

Kennedy, J. & Eberhart, R., 1995. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, Nov/Dec, Volume 4, pp. 1942 - 1948.

Kulkarni, S. R. & Harman, G., 2011. Statistical Learning Theory: A Tutorial.

Land, K. C., 2015. Forecasting. In: *International Encyclopedia of the Social & Behavioral Sciences*. North Carolina: Elsevier, pp. 316-323.

ling Chen, H. et al., 2014. Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy,. *Applied Mathematics and Computation*, July, Volume 239, pp. 180-197.

Lin, G. F., Jhong, C. B. & Chang, C. C., 2013. Development of an effective data-driven model for hourly typhoon rainfall forecasting. *Journal of Hydrology*, Volume 495, pp. 52-63.

Liu, Y. & Wang, R. X., 2015. Study on network traffic forecast model of SVR optimized by GAFSA. *Chaos, Solitons and Fractals*, Volume 000, pp. 1-7.

Lixing, D., Jinhu, L., Xuemei, L. & lanlan, L., 2010. Support Vector Regression and Ant Colony Optimization for HVAC Cooling Load Prediction. *IEEE*, pp. 537-51.

Löwe, R. et al., 2014. Probabilistic online runoff forecasting for urban catchments using inputs from rain gauges as well as statically and dynamically adjusted weather radar. *Journal of Hydrology*, Volume 512, p. 397–407.

Lu, C.-J., Lee, T.-S. & Chiu, C.-C., 2009. Financial time series forecasting using independent component analysis and support vector regression. *Elsevier*, Volume 47, p. 115–125.

Lu, K. & Wang, L., 2011. A Novel Nonlinear Combination Model Based on Support Vector Machine for Rainfall Prediction. *Computational Sciences and Optimization (CSO), 2011 Fourth International Joint Conference on*, 15-19 April, pp. 1343-1346..

Lu, Z., Liang, L., Song, G. & Wang, S., 2010.. Polychotomous kernel Fisher discriminant via top-down induction of binary tree. *Computers & Mathematics with Applications*, Volume 60(3), pp. 511-519.

Makridakis, S. G., Wheelwright, S. C. & Hyndman, R. J., 1997. *Forecasting Methods and Applications*. 3rd ed. s.l.:John Wiley & Sons, Inc..

Marohasy, J. & Abbot, J., 2015. Assessing the quality of eight different maximum temperature time series as inputs when using arti fi cial neural networks to forecast monthly rainfall at Cape Otway, Australia. *Atmospheric Research*, Volume 166, p. 141 –149.



Ma, X., Zhang, Y. & Wang, Y., 2015. Performance evaluation of kernel functions based on grid search for support vector regression. In *Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2015 IEEE 7th International Conference on, July.pp. 283-288.

Media Indonesia, 2013. *Ratusan Rumah di Malang Diterjang Puting Beliung*. s.l.:s.n.

Mekanik, F., Imteaz, M., Gato-Trinidad, S. & Elmahdi, A., 2013. Multiple regression and Artificial Neural Network for long-term rainfall forecasting using large scale climate modes. *Journal of Hydrology* , 503 (2013), p. 11–21.

Montgomery, D. C., Jennings, C. L. & Kulahci, M., 2008. *Introduction to Time Series Analysis and Forecasting*. New Jersey: ohn Wiley & Sons, Inc..

Montgomery, D., Jennings, C. & Kulahci, M., 2015. *Introduction to time series analysis and forecasting*. Hoboken(New Jersey): John Wiley & Sons.

Muflich, F., 2011. Pengembangan Sistem Informasi Berbasis Web Untuk Pengelolaan Data Penelitian Dan Pengabdian Kepada Masyarakat (P2M) Di Fakultas Teknik Universitas Sebelas Maret.

Mujiasih, S., 2011. Pemanfaatan Data Mining Untuk Prakiraan Cuaca. *Jurnal Meteorologi dan Geofisika*, 2 September, Volume 12, pp. 189 - 195.

Nayak, D. R., Mahapatra, A. & Mishra, P., 2013. A Survey on Rainfall Prediction using Artificial Neural Network. *International Journal of Computer Applications*, Juni, Volume 72, pp. 32-40.

Nia, A. R., Far, M. H. & Niaki, S. T. A., 2014. A fuzzy vendor managed inventory of multi-item economic order quantity model under shortage: An ant colony optimization algorithm. *Int. J. Production Economics*, Volume 155, pp. 259-271.

Nugroho, A. S., Witarto, A. B. & Handoko, D., 2003. *Support Vector Machine : Teori dan Aplikasinya dalam Bioinformatika*. [Online] Available at: <http://www.llmuKomputer.Com> [Accessed 27 Januari 2016].

Pai, P. & Hong, W., 2007. A recurrent support vector regression model in rainfall forecasting. *Hydrological Processes*, Volume 21(6), pp. 819-827.

Palit, A. & Popovic, D., 2006. *Computational intelligence in time series forecasting*. s.l.:Springer Science & Business Media.

Pan, T. Y. et al., 2013. Improvement of watershed flood forecasting by typhoon rainfall climate model with an ANN-based southwest monsoon rainfall enhancement. *Journal of Hydrology*, Volume 506, pp. 90-100.

Pawitan, H. & Haryani, G. S., 2011. Water Resources, Sustainability and Societal Livelihoods in Indonesia. *Ecohydrology & Hydrobiology*, 11(3-4), pp. 231-243.

Prahutama, A. & Yasin, H., 2015. Prediction of Weekly Rainfall in Semarang City Use Support Vector Regression (SVR) with Quadratic Loss Function. *International Journal of Science and Engineering*, July, Volume 9(1), pp. 13-16.

Putra, Y. A. & Fatichah, C., n.d. Implementasi Metode Ant Colony Optimization untuk Pemilihan Fitur Pada Kategorisasi Dokumen Teks. *Teknologi Informasi, Institut Teknologi Sepuluh Nopember*.

Qahwaji, R. & Colak, T., 2007. Automatic Short-Term Solar Flare Prediction Using Machine Learning And Sunspot Associations. *Department of Electronic Imaging and Media Communications*.

Ramadhani, A., Farmadi, A. & Budiman, I., 2014. Clustering Data Cuaca Untuk Pengenalan Pola Perioditas Iklim Wilayah Pelaihari Dengan Metode Fuzzy C-Means. *Jurnal Teknologi dan Industri*, Juni, Volume 3, pp. 57-64.

Rasmussen, C. & Williams, C., 2006. *Gaussian processes for machine learning*. s.l.:The MIT Press.

Rini, D., Shamsuddin, S. & Yuhani, S., 2011. Particle swarm optimization: technique, system and challenges. *International Journal of Computer Applications*, January, Volume 14(1), pp. 19-26.

Salcedo-Sanz, S. et al., 2011. Short term wind speed prediction based on evolutionary support vector regression algorithms. *Expert Systems with Applications*, April, Volume 8(4), pp. 4052-4057.

Santosa, B. & Willy, P., 2011. *Metoda Metaheuristik, Konsep dan Implementasi*. Surabaya: Graha Ilmu.

Sanusi, S., Buono, A., Sitanggang, I. & Faqih, A., 2014. Downscaling Modeling Using Support Vector Regression for Rainfall Prediction. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, August, Volume 12(8), pp. 6423-6430.

Schepen, A. & Wang, Q., 2014. Ensemble forecasts of monthly catchment rainfall out to long lead times by post-processing coupled general circulation model output. *Journal of Hydrology*, Volume 519, p. 2920–2931.

Schölkopf, B. & Smola, A., 2002. Support Vector Machines and Kernel Algorithms.

Shao, B., Liu, J., Huang, Z. & Li, R., 2011. A parallel particle swarm optimization algorithm for reference stations distribution. *Journal of Software*, July, Volume 6(7), pp. 1281-1288.

Shi, Y. & Eberhart, R., 1998. A Modified Particle Swarm Optimizer. May. pp. 69-73.

Shoba, G. & Shobha, G., 2014. Rainfall Prediction Using Data Mining techniques: A Survey. *International Journal Of Engineering And Computer Science*, Mei, 3(5), pp. 6206-6211.

Silver, N., 2012. *The signal and the noise: Why so many predictions fail-but some don't*. s.l.:Penguin.

Smola, A. & Schölkopf, B., 2004. A tutorial on support vector regression.. *Statistics and computing*, Volume 14(3), pp. 199-222.

Somvanshi, V. et al., 2006. Modelling and prediction of rainfall using artificial neural network and ARIMA techniques. *J. Ind. Geophys. Union*, Volume 10(2), pp. 141-151.

Tim Pusat Bahasa Depdiknas, 2008. *Kamus Bahasa Indonesia*. Jakarta: Pusat Bahasa Departemen Pendidikan Nasional.

T. & Musyafiroh, E., 2014. Penerapan Metode Exponential Smoothing Untuk Prediksi Jumlah Penjualan Air Minum Dalam Kemasan. *JUTISI*, Volume Vol. 3, p. 465 – 526.

Toksari, M. D., 2016. A hybrid algorithm of Ant Colony Optimization (ACO) and Iterated Local Search (ILS) for estimating electricity domestic consumption: Case of Turkey. *Electrical Power and Energy Systems*, Volume 78, pp. 776-782.

Trafalis, T., Santosa, B. & Richman, M., 2005. Learning networks in rainfall estimation. *Computational Management Science*, Volume 2(3), pp. 229-251.

Tsai, P., Hong, W., Chen, C. & Chen, C., 2014. Parallel Particle Swarm Optimization Optimized LDI Controller with Lyapunov Stability Criterion for Nonlinear Structural Systems. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, Volume 8(5), pp. 858-862..

Tukidi, 2010. Karakter Curah Hujan Di Indonesia. *Jurusan Geografi FIS UNNES*, 7(2), pp. 136-145.

Urraca, R. et al., 2016. Smart baseline models for solar irradiation forecasting. *Energy Conversion and Management*, Volume 108, p. 539–548.

Valverde, M., Araujo, E. & Velho, H. C., 2014. Neural network and fuzzy logic statistical downscaling of atmospheric circulation-type specific weather pattern for rainfall forecasting. *Applied Soft Computing*, Volume 22, p. 681–694.

Vapnik, V., Golowich, S. & Smola, A., 1996. Support vector method for function approximation, regression estimation, and signal processing. *In Advances in neural information processing systems 9*.

Vapnik, V. N., 1998. *Statistical Learning Theory*. New York: JOHN WILEY & SONS, INC.

Vert, J., Tsuda, K. & Schölkopf, B., 2004. A primer on kernel methods. *Kernel Methods in Computational Biology*, pp. 35-70.

Vijayakumar, S. & Wu, S., 1999. Sequential Support Vector Classifiers and Regression. *proc. International Conference on Soft Computing*, pp. 610-619.

Vijayakumar, S. & Wu, S., 1999. Sequential Support Vector Classifiers and Regression. *IJA/SOCO*, June, pp. 610-619.

Vikram, P. & Veer, P., 2011. Rainfall forecasting using Nonlinear SVM based on PSO. *International Journal of Computer Science and Information Technologies*, Volume 2(5), pp. 2309-2313.

Wei, C., 2012. Wavelet Support Vector Machines for Forecasting Precipitation in Tropical Cyclones: comparisons with GSVM, Regression, and MM5. *Weather and Forecasting*, April, Volume 27(2), pp. 438-450.

Williams, C. & Rasmussen, C., 2006. *Gaussian processes for machine learning*. s.l.:the MIT Press.

Wirawan, R. S., 2011. *Perbandingan Peramalan Permintaan Antara Artificial Neural Network dan Support Vector Regreeion Dengan Metode Tradisional*. Depok, Fakultas Teknik, Universitas Indonesia.

Xu, L. & Zhang, F., 2007. Parallel Particle Swarm Optimization for Attribute Reduction. *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, Volume 1, pp. 770-775.

Xun W., An Y.B. & R., J., 2015. Application of Parallel Particle Swarm Optimize Support Vector Machine Model Based on Hadoop Framework in the Analysis of Railway Passenger Flow Data in China. *Chemical Engineering Transactions*, Volume 46, pp. 367-372.

Yan, X. & Chowdhury, N. A., 2014. Mid-term electricity market clearing price forecasting: A multiple SVM approach. *Electrical Power and Energy Systems*, Volume 58, pp. 206-214.

Yao, B., Hu, P., Zhang, M. & Jin, M., 2014. A support vector machine with the tabu search algorithm for freeway incident detection. *International Journal of Applied Mathematics and Computer Science*, Volume 24(2), pp. 397-404.

Yaseen, Z. M. et al., 2015. Artificial intelligence based models for stream-flow forecasting:2000-2015. *Journal of Hydrology*, Volume 530, pp. 829-844.

Zhao, S. & Wang, L., 2010. Support Vector Regression Based on Particle Swarm Optimization for Rainfall. *Third International Joint Conference on Computational Science and Optimization*, May, Volume 2, pp. 484-487.

Zhou, W. et al., 2015. Nonlinear QSAR Models with High-Dimensional Descriptor Selection and SVR Improve Toxicity Prediction and Evaluation of Phenols on Photobacterium Phosphoreum. *Chemometrics and Intelligent Laboratory Systems*, Volume 145, pp. 30-38.

## LAMPIRAN A Data Curah Hujan Dasarian Berdasarkan Runut Waktu

### A.1 Karangploso

Tabel 7.1 Data curah hujan dasarian tahun 2000-2014 wilayah Karangploso Malang

Tahun	Januari			Februari			Maret			April			Mei			Juni		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2000	49	76	96	69	64	84	14	87	196	32	70	13	20	0	0	0	0	0
2001	140	0	130	167	77	24	67	188	133	17	19	0	143	0	0	143	12	0
2002	82	95	279	163	74	51	40	23	47	78	56	5	0	16	0	0	0	0
2003	76	15	89	70	183	109	91	138	0	63	0	0	35	64	0	0	0	0
2004	109	63	217	107	69	106	139	181	145	24	0	0	48	29	43	0	0	0
2005	65	127	165	7	0	53	223	48	125	85	0	0	0	0	0	0	0	16
2006	213	70	93	135	90	165	56	96	116	129	115	25	88	6	45	42	0	0
2007	9	85	42	104	52	33	20	86	60	90	91	32	0	7	0	6	0	10
2008	40	0	107	107	38	129	127	93	229	18	15	11	28	0	0	0	2	0
2009	96	69	144	123	34	0	59	5	14	26	24	8	27	52	28	1	0	0
2010	138	66	179	116	116	81	326	36	101	176	0	0	143	25	132	25	3	0
2011	49	48	87	174	66	22	132	89	204	212	198	81	154	93	22	0	0	0
2012	111	77	52	174	177	149	108	59	187	25	0	11	24	0	0	0	0	20
2013	74	57	156	49	111	96	85	88	92	89	93	0	0	47	47	75	27	0
2014	177	29	144	48	77	0	124	84	0	93	37	0	11	32	187	0	0	0
Tahun	Juli			Agustus			September			Oktober			November			Desember		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2000	0	26	0	0	0	0	0	0	114	47	175	23	138	196	22	0	93	11
2001	0	0	0	0	0	0	0	0	0	71	21	140	104	93	19	44	93	72
2002	0	0	0	0	0	0	0	0	0	0	0	9	14	155	8	153	0	259
2003	0	0	0	0	0	0	0	0	0	0	0	12	0	138	96	205	15	39

2004	0	0	0	0	0	0	0	0	0	0	0	38	0	72	86	205	109	23	142
2005	72	0	0	0	0	0	0	0	0	0	0	97	33	0	62	75	11	81	108
2006	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	16	59	140
2007	7	0	0	0	0	0	0	0	0	29	0	0	0	206	0	0	78	101	185
2008	0	0	0	0	0	0	3	0	0	27	0	0	0	66	30	92	47	168	11
2009	30	0	0	0	0	0	0	3	0	0	23	0	0	80	54	69	12	102	
2010	22	42	18	0	46	41	61	77	103	85	83	92	294	46	176	0	0	0	
2011	0	0	0	0	0	0	0	0	0	0	0	0	39	187	26	113	0	0	
2012	0	0	0	0	0	0	0	0	0	25	82	0	0	121	77	115	124	168	
2013	57	3	27	0	0	0	0	0	0	0	0	0	96	31	53	63	368	152	72
2014	0	20	0	10	0	0	0	0	0	0	0	0	96	187	26	113	118	141	141

## A.2 Poncokusumo

Tabel 7.2 Data curah hujan dasarian tahun 2000-2014 wilayah Poncokusumo Malang

Tahun	Januari			Februari			Maret			April			Mei			Juni		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2000	152	90	236	334	110	87	146	198	286	130	201	203	69	43	56	0	54	0
2001	298	149	244	279	116	13	43	183	228	133	112	9	74	0	10	80	28	32
2002	69	101	224	148	167	109	145	113	202	112	145	50	0	39	0	0	0	0
2003	259	51	286	89	149	93	170	29	63	141	48	0	101	27	0	0	0	0
2004	161	232	232	68	235	158	97	252	99	10	23	0	55	11	40	23	0	0
2005	83	88	114	56	136	187	204	99	78	106	57	0	0	0	0	0	32	77
2006	279	77	132	89	162	118	122	134	97	56	112	42	67	52	82	6	0	0
2007	10	7	51	94	203	61	46	125	205	59	64	28	0	18	25	11	0	13
2008	149	48	51	80	18	29	103	161	188	36	0	9	48	0	0	6	0	0
2009	75	223	255	57	257	183	113	78	65	169	205	90	13	74	67	0	0	0
2010	148	80	217	219	146	81	114	47	119	168	194	223	81	46	29	29	34	57

2011	55	51	91	164	53	19	105	80	116	172	64	78	95	32	23	0	0	0
2012	214	112	76	114	115	78	210	94	39	33	56	18	28	30	0	0	0	0
2013	59	53	226	193	221	46	172	34	49	63	122	89	0	25	78	193	0	14
2014	251	77	71	73	23	59	18	199	60	22	135	72	19	7	5	0	22	17
Tahun	Juli			Agustus			September			Oktober			November			Desember		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2000	0	0	0	0	0	114	0	0	0	0	147	138	205	258	199	0	72	26
2001	0	8	28	0	0	0	0	0	0	29	72	153	0	149	86	147	61	84
2002	0	0	0	0	0	0	0	0	0	0	0	0	92	78	194	100	261	345
2003	0	0	0	0	0	0	0	0	0	11	0	40	28	208	180	177	105	198
2004	0	0	0	0	0	0	0	22	0	0	0	59	98	26	241	197	152	244
2005	23	2	0	7	0	0	0	27	0	0	81	42	26	119	59	134	137	194
2006	0	0	0	0	0	0	0	0	0	0	0	0	22	21	31	94	96	167
2007	0	0	0	0	0	0	0	0	0	0	0	49	134	0	10	166	157	254
2008	0	0	0	0	0	0	0	0	0	47	51	83	102	181	66	240	164	36
2009	0	0	0	0	0	0	0	0	0	7	0	0	16	66	28	19	0	57
2010	16	53	42	0	0	53	59	115	153	42	7	184	179	60	128	126	119	40
2011	0	0	0	0	0	0	0	0	0	0	0	55	102	124	57	92	85	252
2012	0	0	0	0	0	0	0	0	0	4	38	17	5	113	94	105	98	143
2013	8	28	12	0	0	0	0	0	0	0	27	22	23	51	64	30	279	141
2014	0	24	0	0	0	0	0	0	0	0	27	22	102	124	57	138	97	228

### A.3 Pujon

Tabel 7.3 Data curah hujan dasarian tahun 2000-2014 wilayah Pujon Malang

Tahun	Januari			Februari			Maret			April			Mei			Juni		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2000	95	71	253	117	9	133	95	91	107	102	77	37	74	0	12	34	0	15

2001	125	76	131	314	246	52	117	87	155	133	88	35	27	0	12	73	0	0
2002	82	80	259	275	209	115	30	130	80	61	165	30	0	0	0	0	0	0
2003	75	3	95	237	226	170	121	135	10	5	0	0	96	30	0	0	6	0
2004	48	71	263	239	107	124	158	233	126	0	28	35	135	12	150	0	0	0
2005	36	59	50	40	16	75	101	36	66	238	46	0	5	0	0	0	0	79
2006	106	218	154	53	159	66	81	63	102	40	143	116	36	52	142	0	0	0
2007	53	0	73	165	77	201	95	67	144	109	88	43	0	22	13	9	0	17
2008	249	53	141	153	198	106	250	208	136	130	31	23	70	0	0	0	0	0
2009	80	246	146	303	105	162	72	24	122	6	39	61	0	125	82	0	43	0
2010	169	284	87	178	100	165	166	45	128	94	166	65	71	77	156	61	52	0
2011	146	103	139	115	26	87	132	23	215	80	131	54	177	63	0	0	0	12
2012	177	80	128	180	163	148	73	119	71	40	21	30	46	0	0	0	0	13
2013	208	352	288	62	173	192	75	129	103	287	156	0	12	13	137	45	61	25
2014	176	235	342	300	39	174	72	104	80	39	74	99	0	23	30	0	9	12
Tahun	Juli			Agustus			September			Oktober			November			Desember		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2000	0	0	0	0	0	0	0	0	0	128	45	115	114	173	0	100	0	0
2001	0	34	0	0	0	0	0	22	8	122	78	144	62	190	7	46	44	86
2002	0	0	0	0	0	0	0	0	0	0	0	0	0	72	127	110	63	206
2003	0	0	0	0	0	0	0	0	0	15	24	10	224	217	171	73	51	0
2004	18	6	0	0	0	0	0	18	0	0	0	7	25	12	128	197	84	162
2005	23	0	0	0	15	0	0	5	36	0	63	89	0	81	72	98	125	174
2006	0	0	12	0	0	0	0	0	0	0	0	0	17	0	0	2	38	275
2007	36	0	0	0	1	0	12	0	0	37	0	65	188	0	18	171	252	416
2008	0	0	0	0	0	0	21	0	0	79	24	56	61	38	154	79	97	0
2009	0	0	0	0	0	0	0	10	0	0	39	28	14	85	88	25	16	143
2010	15	26	0	21	5	64	76	66	95	45	65	118	163	0	292	225	86	55
2011	0	5	0	0	0	0	0	0	0	0	3	52	193	109	71	109	40	184
2012	0	0	0	0	0	0	0	0	0	38	30	10	5	204	135	112	114	186



2013	55	96	38	0	0	8	0	0	0	38	0	56	29	222	74,5	322	202	152
2014	0	0	49	9	0	0	0	0	0	38	0	56	193	109	71	149	184	223

#### A.4 Tangkilsari

Tabel 7.4 Data curah hujan dasarian tahun 2000-2014 wilayah Tangkilsari Malang

Tahun	Januari			Februari			Maret			April			Mei			Juni		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2000	147	46	100	99	79	118	127	80	184	28	99	64	88	62	97	28	62	25
2001	229	21	196	147	133	29	22	82	177	50	50	7	85	0	0	49	118	0
2002	72	30	277	82	94	65	141	175	202	109	176	10	46	0	0	18	0	0
2003	125	23	214	44	72	42	142	131	89	70	14	63	51	11	0	6	0	17
2004	0	0	0	62	133	215	86	205	65	17	18	60	19	0	24	0	0	0
2005	31	97	23	38	151	109	101	24	0	143	148	15	0	0	8	0	0	121
2006	178	39	89	134	70	19	0	0	0	69	132	0	127	23	84	0	0	0
2007	24	20	53	63	31	0	22	76	146	228	82	62	0	53	48	53	0	14
2008	43	19	86	75	4	98	122	108	206	105	0	0	19	0	0	9	24	0
2009	44	59	145	86	153	189	43	54	31	91	54	0	0	42	38	0	7	0
2010	162	31	199	172	201	52	96	63	55	55	109	208	110	52	43	8	3	16
2011	79	45	115	82	39	15	0	119	88	77	81	40	49	20	0	5	0	0
2012	110	134	39	31	70	43	219	45	0	109	68	29	0	19	0	0	0	0
2013	129	53	102	85	103	78	157	123	22	107	141	56	0	115	117	158	31	54
2014	66	69	98	100	23	60	68	88	30	63	77	119	27	10	78	49	38	0
Tahun	Juli			Agustus			September			Oktober			November			Desember		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2000	0	0	0	0	0	30	0	0	36	0	169	133	95	43	46	0	113	70
2001	9	0	0	0	0	0	0	0	0	53	34	135	60	0	26	148	0	74
2002	0	0	0	0	0	0	0	0	0	0	8	0	90	53	98	18	121	260

2003	0	0	0	0	0	0	0	0	0	41	0	3	41	126	262	262	109	102
2004	0	14	0	0	0	0	0	51	0	0	1	7	26	6	188	205	0	149
2005	0	0	0	0	0	0	0	17	1	0	48	0	37	11	202	120	116	259
2006	0	0	0	0	0	0	0	0	0	0	0	0	0	26	33	86	66	182
2007	0	0	0	0	0	0	0	0	0	0	0	34	114	27	23	107	111	279
2008	0	0	0	0	0	0	0	0	0	18	12	12	110	60	52	61	0	71
2009	0	0	6	0	0	0	0	23	0	0	2	25	2	56	84	0	0	87
2010	17	0	14	0	0	54	2	104	63	65	12	158	263	42	82	186	0	19
2011	0	0	0	0	0	0	0	0	0	0	0	66	125	72	55	28	54	128
2012	0	0	0	0	0	0	0	0	0	0	0	0	0	51	54	251	142	195
2013	34	1	10	0	0	0	0	0	0	0	15	25	8	80	74	252	287	136
2014	0	0	0	0	0	0	0	0	0	0	15	25	125	72	55	166	92	143

### A.5 Wagir

Tabel 7.5 Data curah hujan dasarian tahun 2000-2014 wilayah Wagir Malang

Tahun	Januari			Februari			Maret			April			Mei			Juni		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2000	120	114	96	131	88	165	79	109	241	60	122	66	54	51	66	86	0	0
2001	293	65	124	163	71	0	70	170	186	65	93	42	40	0	35	80	80	0
2002	203	65	253	85	115	75	45	199	245	138	78	0	61	0	0	0	0	0
2003	240	128	276	10	160	63	136	130	19	73	79	0	64	80	15	25	0	0
2004	60	177	276	121	234	205	175	175	111	15	55	97	46	0	51	0	25	0
2005	31	175	51	56	209	182	140	52	195	100	65	0	0	0	0	65	100	0
2006	240	132	157	167	115	93	132	194	137	62	118	26	120	0	62	11	0	0
2007	10	41	63	158	52	85	73	193	143	65	112	137	0	70	0	102	0	71
2008	78	52	143	94	18	161	145	281	211	144	38	0	103	2	3	0	32	0
2009	128	114	248	99	150	349	150	24	85	49	155	106	74	192	188	0	33	0

2010	305	193	479	138	217	85	234	331	55	141	550	503	366	92	172	16	32	18	
2011	114	48	123	175	27	12	211	146	252	145	177	93	191	81	2	15	0	0	
2012	144	269	58	121	106	134	180	50	60	117	33	30	86	15	0	44	0	0	
2013	100	104	152	43	225	67	147	99	120	153	150	0	14	31	142	158	32	47	
2014	313	57	153	179	40	166	34	299	65	140	107	93	9	33	66	2	2	25	
Tahun	Juli			Agustus			September			Oktober			November			Desember			
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	
2000	0	0	0	0	0	67	0	0	90	0	242	156	306	203	102	0	0	50	
2001	0	0	90	0	0	0	0	25	30	224	51	111	50	131	53	100	122	160	
2002	0	0	0	0	0	0	0	0	0	0	0	25	34	43	96	94	146	313	
2003	0	0	0	0	0	0	10	0	0	21	0	20	0	80	234	418	86	135	
2004	0	0	0	0	0	0	0	40	0	0	0	28	67	32	356	153	84	211	
2005	37	21	0	21	0	0	0	0	36	0	82	53	11	69	197	132	190	414	
2006	0	0	0	0	0	0	0	0	0	0	0	0	29	0	80	54	137	295	
2007	0	25	0	0	0	0	0	0	0	0	0	31	252	60	18	155	154	298	
2008	0	0	0	0	0	0	14	0	0	70	58	74	99	171	92	87	160	113	
2009	0	0	9	0	0	0	0	5	0	0	0	14	15	29	88	100	7	301	
2010	11	13	9	0	8	107	33	155	141	94	94	91	279	70	95	151	115	55	
2011	0	0	0	0	0	0	0	5	0	0	7	36	225	109	127	40	148	209	
2012	0	4	0	0	0	0	0	0	0	0	0	0	46	125	69	90	252	121	
2013	10	34	10	0	0	0	0	0	0	0	0	0	70	45	77	256	176	260	131
2014	0	37	0	0	0	0	0	0	0	0	0	0	70	225	109	127	75	149	142

A.6 Junggo

Tabel 7.6 Data curah hujan dasarian tahun 2004-2015 wilayah Junggo Malang

Tahun	Januari			Februari			Maret			April			Mei			Juni		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2003	195	17	82	71	150	64	167	75	62	167	0	22	132	21	0	0	4	0
2004	26	200	149	126	184	259	76	264	32	4	2	0	6	43	38	0	21	0
2005	24	53	59	0	184	143	130	62	76	74	93	76	0	0	0	0	0	117
2006	354	79	109	86	183	81	65	198	123	85	358	52	48	5	180	0	0	0
2007	32	55	66	181	109	153	98	84	134	135	95	106	0	2	8	2	2	1
2008	133	47	135	95	80	98	174	274	198	164	0	19	62	15	2	6	1	0
2009	83	125	181	163	75	200	136	115	58	38	61	35	28	70	42	5	0	0
2010	196	104	115	151	161	144	188	18	76	65	244	121	103	24	100	0	64	4
2011	77	57	148	66	14	34	100	10	229	110	249	13	136	77	7	0	4	28
2012	144	167	70	131	77	149	159	73	162	24	27	24	35	8	4	0	2	5
2013	76	130	213	55	126	50	98	130	159	112	144	0	7	8	142	120	22	53
2014	157	118	65	103	25	126	101	140	19	49	75	45	17	13	24	4	57	10
Tahun	Juli			Agustus			September			Oktober			November			Desember		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
2003	0	0	0	0	0	0	0	13	0	20	0	0	54	105	254	186	70	63
2004	0	5	0	0	0	0	0	45	3	0	0	6	88	9	185	264	117	173
2005	101	57	0	0	0	0	0	80	8	2	69	48	0	26	85	293	235	205
2006	0	0	0	0	0	0	0	0	0	0	0	0	44	54	19	48	100	186
2007	2	1	0	4	3	0	0	0	0	25	0	47	141	6	23	132	85	285
2008	0	0	0	0	0	45	8	0	0	87	12	98	30	40	160	187	225	12
2009	0	0	0	0	0	0	0	46	0	0	38	87	5	62	203	5	55	141
2010	26	11	24	17	9	102	48	81	93	59	145	58	146	1	32	181	65	60
2011	0	13	0	0	0	0	0	0	0	2	0	47	217	159	74	42	102	193
2012	0	17	0	0	0	0	0	0	0	18	28	11	38	145	132	149	217	144

2013	26	36	11	13	0	14	0	0	0	0	0	0	91	88	87	231	229	55
2014	29	32	6	17	18	1	0	0	0	0	0	2	0	76	162	212	118	127

