

PERAMALAN RUNUT WAKTU CURAH HUJAN MENGUNAKAN METODE "SVR - PPSO"

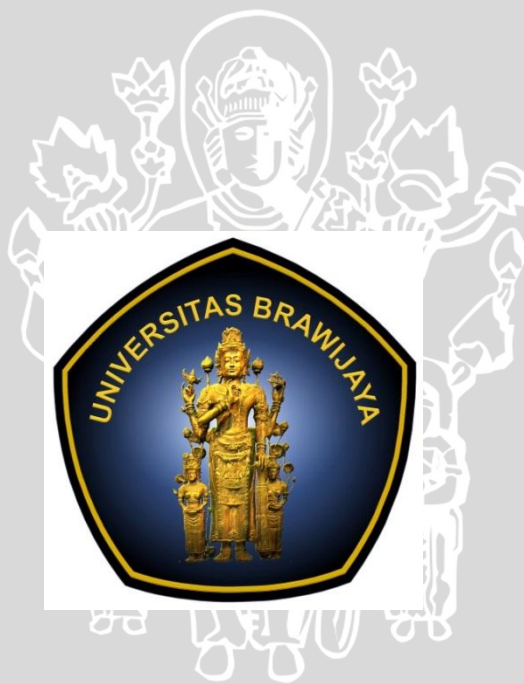
SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Januar Anas Fauzi Syams

NIM: 125150207111020



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017**

PENGESAHAN

PERAMALAN RUNUT WAKTU CURAH HUJAN
MENGUNAKAN METODE "SVR - PPSO"

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:

Januar Anas Fauzi Syams

NIM: 125150207111020

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: 19720425 199903 1 002

Hartanto, S.T., M.M.

NIP: 19730418 199503 1 001

Mengetahui

Ketua Jurusan Teknik Informatika

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

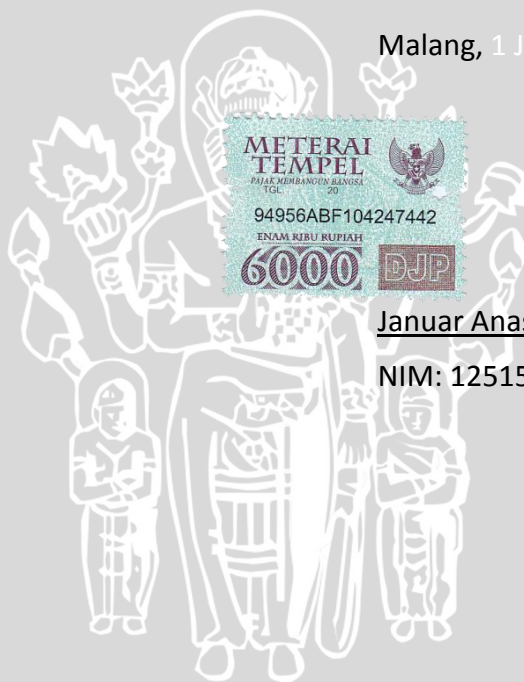
NIP: 19710518 200312 1 001

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 di dalam 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, 1 Juni 2016



Januar Anas Fauzi Syams

NIM: 125150207111020

KATA PENGANTAR

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 - PPSO" 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. Kedua orang tua dan keluarga besar Syams atas segala nasehat, kasih sayang, perhatian dan kesabarannya, serta senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
2. Bapak Arief Andy Soebroto, S.T, M.Kom selaku dosen pembimbing skripsi yang telah banyak membantu, membimbing, dan mengarahkan sehingga skripsi ini dapat selesai.
3. 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.
4. Bapak Ahmad Luthfi, S.ST. selaku pihak BMKG Stasiun Klimatologi Karangploso Malang yang juga turut ikut membantu dan mengarahkan sehingga dapat menyelesaikan skripsi ini.
5. Bapak Agus Wahyu Widodo, S.T, M.Cs dan Ibu Candra Dewi, S.Kom, M.Sc selaku penguji yang telah memberikan masukan dan saran sehingga skripsi ini menjadi lebih baik.
6. Bapak Wibisono Sukmo Wardhono, S.T, M.T selaku dosen penasehat akademik yang selalu memberikan nasehat selama menempuh masa studi.
7. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku ketua jurusan teknik informatika.
8. Seluruh tetangga rumah, saudara dekat maupun jauh satu turunan umat manusia, burung peliharaan "Bejo", serta kawan maupun lawan sepermainan dan lainnya atas dukungan, masukan dan semangat yang diberikan kepada penulis sehingga terselesaikannya skripsi ini.
9. 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, 20 Juli 2016

Penulis
Januaranas21@gmail.com

ABSTRAK

Informasi cuaca dapat digunakan masyarakat untuk mengantisipasi kejadian buruk yang menimbulkan bencana, sehingga merugikan jiwa dan harta. Cuaca merupakan kondisi sementara rata-rata udara yang menjadi hal utama untuk penentuan iklim di suatu wilayah berupa rata-rata cuaca pada jangka waktu tertentu, sedangkan curah hujan menjadi unsur yang berpengaruh terhadap perubahan iklim di suatu wilayah. Curah hujan merupakan fenomena yang rumit dan bersifat non linier. Selain itu, curah hujan juga menjadi variabel penting untuk peramalan cuaca yang dapat mempengaruhi kehidupan. Oleh sebab itu, diperlukan peramalan curah hujan yang akan terjadi di waktu mendatang untuk dapat mendukung penentuan cuaca dan iklim di suatu wilayah pada waktu tertentu. Peramalan dapat dilakukan berdasarkan pendekatan secara objektif dengan melakukan komputasi matematika pada data historis runut waktu. Metode *Support Vector Regression* (SVR) merupakan *machine learning* pengembangan dari metode *Support Vector Machine* (SVM) yang dapat melakukan peramalan objek bersifat non linier. Untuk meningkatkan akurasi regresi metode SVR diperlukan penentuan nilai parameter SVR yang tepat, seperti parameter C dan ϵ . Metode *Particle Swarm Optimization* (PSO) yang terinspirasi dari tingkah laku sekumpulan makhluk hidup dapat melakukan optimasi parameter metode SVR dan strategi *Parallel* PSO (PPSO) dapat meningkatkan pencarian solusi optimum secara global. Berdasarkan evaluasi kinerja peramalan runut waktu curah hujan di wilayah Poncokusumo menggunakan metode SVR-PPSO yang mengacu pada nilai *Root Mean Square Error* (RMSE). Metode SVR-PPSO dapat melakukan peramalan runut waktu curah hujan tahunan dengan nilai RMSE 41.51560, hujan musiman dengan nilai RMSE 56.11680, dan hujan tahun ke tahun dengan nilai RMSE 23.35970.

Kata kunci: Peramalan Runut Waktu, Curah Hujan, *Support Vector Regression*, *Parallel Particle Swarm Optimization*

ABSTRACT

Weather information can be used by the community to anticipate any disasters. The weather is a temporary condition of the average air that the main thing for the determination of the climate in a region such as the average weather at a certain period, while rainfall into elements that influence climate change in the region. Rainfall is a complex phenomenon and non linear. In addition, rainfall is also becoming an important variable for weather forecasting which could affect life. Therefore, it is necessary that rainfall forecasting will occur in the future be able to support the determination of the weather and climate in an area at a particular time. Forecasting can be done based on an objective approach to compute the math on historical data time series. Method of Support Vector Regression (SVR) is the development of methods of machine learning Support Vector Machine (SVM) to forecasting the object is non linear. To improve the accuracy of regression methods necessary to determine SVR appropriate parameter values, such as the parameters C and ϵ . Particle Swarm Optimization (PSO) method which is inspired by the behavior of a set of creatures could perform parameter optimization methods and strategies SVR Parallel PSO (PPSO) can improve search optimum solution globally. Based on the performance evaluation of time-series forecasting rainfall in Poncokusumo region using the SVR-PPSO which refers to the value Root Mean Square Error (RMSE). SVR-PPSO method can perform time-series forecasting annual rainfall with RMSE value 41.51560, seasonal precipitation with RMSE value 56.11680, and rain to the year with RMSE value 23.35970.

Keywords: Forecasting time series, Rainfall, Support Vector Regression, Parallel Particle Swarm Optimization



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR PERSAMAAN.....	xx
DAFTAR KODE PROGRAM	xxiii
DAFTAR LAMPIRAN	xxiv
BAB 1 PENDAHULUAN.....	1
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
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Peramalan	37
2.2.1 Langkah-Langkah Peramalan	38
2.2.2 Metode Peramalan	41
2.2.3 Jangka Waktu Peramalan	43
2.2.4 Evaluasi Kinerja Peramalan	44
2.3 Data Runut Waktu	47
2.3.1 Analisis Data Runut Waktu.....	48
2.3.2 Karakteristik Data Runut Waktu	49
2.3.3 Pengolahan Awal Data Runut Waktu	52
2.4 Curah Hujan	54

2.4.1 Dampak Curah Hujan	55
2.4.2 Pengukuran Curah Hujan	57
2.4.3 Intensitas Curah Hujan	59
2.4.4 Normal Curah Hujan	59
2.4.5 Pola Curah Hujan Indonesia	60
2.5 <i>Support Vector Regression</i> (SVR)	61
2.5.1 <i>Kernel Trick</i>	67
2.5.2 Algoritma Sekuensial SVR	72
2.5.3 Normalisasi Ruang Fitur SVR	75
2.6 <i>Particle Swarm Optimization</i> (PSO)	77
2.6.1 Algoritma PSO	78
2.6.2 Variasi PSO	82
2.6.3 <i>Parallel PSO</i> (PPSO)	91
BAB 3 METODOLOGI	94
3.1 Studi Literatur	94
3.2 Analisis Kebutuhan	95
3.2.1 Kebutuhan Sistem Komputer	95
3.2.2 Kebutuhan Data Penelitian	96
3.3 Perancangan	96
3.4 Implementasi Sistem	97
3.5 Pengujian dan Pembahasan	97
3.6 Penarikan Kesimpulan	98
BAB 4 PERANCANGAN	99
4.1 Formulasi Strategi Peramalan	100
4.1.1 Definisi Masalah	100
4.1.2 Pengumpulan Data	101
4.1.3 Analisis Data	106
4.1.4 Pemilihan Data	109
4.1.5 Pemilihan Metode	115
4.1.6 Pemodelan	116
4.1.7 Validasi Model	116
4.1.8 Penerapan Model	117

4.1.9 Pengamatan Peramalan	117
4.2 Formulasi Metode SVR-PPSO.....	118
4.2.1 Proses Konfigurasi Data	126
4.2.2 Proses Pelatihan SVR.....	126
4.2.3 Proses Pengujian SVR.....	149
4.2.4 Proses PPSO	162
4.3 Perancangan Sistem.....	193
4.3.1 Perancangan Basis Data	194
4.3.3 Perancangan Antarmuka Pengguna.....	195
4.5 Perancangan Skenario Pengujian	206
4.5.1 Rancangan Uji Coba Batas Parameter SVR	206
4.5.2 Rancangan Uji Coba Jumlah Iterasi SVR.....	209
4.5.3 Rancangan Uji Coba Jumlah Partikel PPSO	210
4.5.4 Rancangan Uji Coba Jumlah Iterasi PPSO	210
4.5.5 Rancangan Uji Coba Jumlah <i>Slave</i> PSO	211
4.5.6 Rancangan Uji Coba Panjang Periode Runut Waktu (<i>Lag Time</i>).....	211
4.5.7 Rancangan Uji Coba Peramalan Runut Waktu Curah Hujan.....	212
4.5.8 Rancangan Uji Coba Perbandingan Peramalan Runut Waktu Curah Hujan Menggunakan Metode SVR, SVR-PPSO, dan Regresi Linier Berganda	215
BAB 5 IMPLEMENTASI SISTEM	216
5.1 Batasan Implementasi	216
5.2 Implementasi Program	217
5.2.1 Implementasi Proses Konfigurasi Data	217
5.2.2 Implementasi Proses Pelatihan SVR.....	218
5.2.3 Implementasi Proses Pengujian SVR.....	222
5.2.4 Implementasi Proses Strategi Paralel PSO	225
5.2.5 Implementasi Proses PSO	227
BAB 6 PENGUJIAN	232
6.1 Hasil dan Pembahasan Uji Coba Batas Parameter SVR	232
6.1.1 Hasil dan Pembahasan Uji Coba Batas Parameter γ	233
6.1.2 Hasil dan Pembahasan Uji Coba Batas Parameter λ	234

6.1.3 Hasil dan Pembahasan Uji Coba Batas Parameter C.....	235
6.1.4 Hasil dan Pembahasan Uji Coba Batas Parameter ϵ	236
6.1.5 Hasil dan Pembahasan Uji Coba Batas Parameter γ_k <i>kernel</i>	237
6.1.6 Hasil dan Pembahasan Uji Coba Batas Parameter d <i>Kernel</i>	238
6.2 Hasil dan Pembahasan Uji Coba Jumlah Iterasi SVR.....	239
6.3 Hasil dan Pembahasan Uji Coba Jumlah Partikel PPSO	241
6.4 Hasil dan Pembahasan Uji Coba Jumlah Iterasi PPSO	243
6.5 Hasil dan Pembahasan Uji Coba Jumlah <i>Slave</i> PSO	245
6.6 Hasil dan Pembahasan Uji Coba Panjang Periode Runut Waktu (<i>Lag Time</i>)	246
6.7 Hasil dan Pembahasan Uji Coba Peramalan Runut Waktu Curah Hujan	248
6.7.1 Hasil dan Pembahasan Uji Coba Pengaruh Parameter ANOVA RBF <i>Kernel</i>	248
6.7.2 Hasil dan Pembahasan Uji Coba Peramalan Runut Waktu Curah Hujan Tahunan	253
6.7.3 Hasil dan Pembahasan Uji Coba Peramalan Runut Waktu Curah Hujan Musiman	254
6.7.4 Hasil dan Pembahasan Uji Coba Peramalan Runut Waktu Curah Hujan Tahun ke Tahun	256
6.8 Hasil dan Pembahasan Uji Coba Perbandingan Peramalan Runut Waktu Curah Hujan Menggunakan Metode SVR, SVR-PPSO, dan Regresi Linier Berganda	257
BAB 7 PENUTUP	260
7.1 Kesimpulan.....	260
7.2 Saran	261
DAFTAR PUSTAKA.....	262
LAMPIRAN A DATA CURAH HUJAN HARIAN WILAYAH KARANGPLOSO	271
A.1 Data curah hujan harian wilayah Karangploso tahun 2010	271
A.2 Data curah hujan harian wilayah Karangploso tahun 2011	272
A.3 Data curah hujan harian wilayah Karangploso tahun 2015	273
LAMPIRAN B HASIL PENGUJIAN	274
B.1 Hasil Uji Coba Batas Parameter γ	274
B.2 Hasil Uji Coba Batas Parameter λ	275

B.3 Hasil Uji Coba Batas Parameter C	276
B.4 Hasil Uji Coba Batas Parameter ϵ	277
B.5 Hasil Uji Coba Batas Parameter γ_k <i>Kernel</i>	278
B.6 Hasil Uji Coba Batas Parameter d <i>Kernel</i>	279
B.7 Hasil Uji Coba Jumlah Iterasi SVR	279
B.8 Hasil Uji Coba Jumlah Partikel PPSO	280
B.9 Hasil Uji Coba Jumlah Iterasi PPSO	281
B.10 Hasil Uji Coba Jumlah <i>Slave</i> PSO	282
B.11 Hasil Uji Coba Panjang Periode Runut Waktu (<i>Lag Time</i>)	283
B.12 Hasil Uji Coba Pengaruh Parameter ANOVA RBF <i>Kernel</i>	284
B.13 Hasil Uji Coba Peramalan Runut Waktu Curah Hujan Tahunan	285
B.14 Hasil uji Coba Peramalan Runut Waktu Curah Hujan Musiman	286
B.15 Hasil uji Coba Peramalan Runut Waktu Curah Hujan Tahun ke Tahun	287
B.16 Hasil uji Coba Perbandingan Peramalan Runut Waktu Curah Hujan Menggunakan Metode SVR, SVR-PPSO, dan Regresi Linier Berganda	287



DAFTAR TABEL

Tabel 2.1 Perbandingan penelitian penulis dengan penelitian terkait sebelumnya	11
Tabel 2.2 Data curah hujan bulanan tahun 2000 - 2001 Karangploso Malang	47
Tabel 2.3 Periode pola musiman.....	50
Tabel 2.4 Dampak curah hujan pada musim hujan dan musim kemarau	56
Tabel 2.5 Intensitas curah hujan	59
Tabel 3.1 Spesifikasi perangkat keras personal komputer	95
Tabel 3.2 Spesifikasi perangkat lunak personal komputer	96
Tabel 4.1 Data curah hujan di wilayah Karangploso tahun 2000-2015	102
Tabel 4.2 Data curah hujan di wilayah Karangploso tahun 2000-2015 (Lanjutan)	103
Tabel 4.3 Data curah hujan di wilayah Poncokusumo tahun 2000-2015	104
Tabel 4.4 Data curah hujan di wilayah Poncokusumo tahun 2000-2015 (Lanjutan)	105
Tabel 4.5 Contoh pemilihan data model pola horizontal	111
Tabel 4.6 Pemilihan data runut waktu curah hujan tahun 2006 Poncokusumo menggunakan pola horizontal	112
Tabel 4.7 Pemilihan data latih dan data uji runut waktu curah hujan tahun 2006 Poncokusumo menggunakan pola horizontal dengan 3 <i>lag time</i>	113
Tabel 4.8 Contoh pemilihan data model pola vertikal.....	113
Tabel 4.9 Pemilihan data latih dan data uji runut waktu curah hujan tahun 2000-2012 Poncokusumo menggunakan pola vertikal dengan 3 <i>lag time</i>	114
Tabel 4.10 Pemilihan data latih dan data uji runut waktu curah hujan tahun 2006 Poncokusumo menggunakan pola horizontal dengan 3 <i>lag time</i>	115
Tabel 4.11 Tahapan proses metode SVR-PPSO.....	119
Tabel 4.12 Inisialisasi data latih dan data uji	126
Tabel 4.13 Batas parameter SVR dan ANOVA RBF <i>Kernel</i>	128
Tabel 4.14 Inisialisasi nilai parameter SVR dan ANOVA RBF <i>Kernel</i>	128
Tabel 4.15 Inisialisasi awal nilai α_i^* dan α_i	128
Tabel 4.16 Hasil perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih	131
Tabel 4.17 Hasil perhitungan normalisasi pada ruang fitur.....	134
Tabel 4.18 Hasil perhitungan matriks Hessien Rij proses pelatihan.....	136

Tabel 4.19 Nilai E_i iterasi ke 1	141
Tabel 4.20 Nilai $\delta\alpha_i^*$ dan $\delta\alpha_i$ iterasi ke 1	142
Tabel 4.21 Nilai α_i^* dan α_i iterasi ke 1	142
Tabel 4.22 Nilai $\delta\alpha_i^*$ dan $\delta\alpha_i$ iterasi ke 2	143
Tabel 4.23 Nilai α_i^* dan α_i iterasi ke 2	143
Tabel 4.24 Nilai $\delta\alpha_i^*$ dan $\delta\alpha_i$ iterasi ke 113	143
Tabel 4.25 Nilai α_i^* dan α_i iterasi ke 113	144
Tabel 4.26 Hasil perhitungan regresi data latih	146
Tabel 4.27 Hasil perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih dan data uji	152
Tabel 4.28 Hasil perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih dan data uji (lanjutan)	153
Tabel 4.29 Hasil perhitungan normalisasi pada ruang fitur	155
Tabel 4.30 Hasil perhitungan matriks Hessian Rij proses pengujian	157
Tabel 4.31 Hasil perhitungan matriks Hessian Rij proses pengujian	158
Tabel 4.32 Hasil perhitungan regresi data uji	161
Tabel 4.33 Batasan dimensi partikel individu	168
Tabel 4.34 Inisialisasi awal posisi dan kecepatan partikel	173
Tabel 4.35 Evaluasi fitness partikel	176
Tabel 4.36 Kecepatan partikel pada iterasi ke 1	181
Tabel 4.37 Pembatasan kecepatan partikel pada iterasi ke 1	182
Tabel 4.38 Posisi partikel pada iterasi ke 1	184
Tabel 4.39 Pembatasan posisi partikel pada iterasi ke 1	185
Tabel 4.40 Perbarui evaluasi fitness partikel pada iterasi ke 1	187
Tabel 4.41 Penentuan keberhasilan partikel pada iterasi ke 1	188
Tabel 4.42 Pengurutan individu terbaik dari <i>slave</i> PSO	192
Tabel 4.43 Perancangan basis data	194
Tabel 4.44 Keterangan komponen daftar pilihan konfigurasi sistem	196
Tabel 4.45 Keterangan komponen panel konfigurasi data ramalan	196
Tabel 4.46 Keterangan komponen panel konfigurasi data ramalan (lanjutan) ..	197

Tabel 4.47 Keterangan komponen panel konfigurasi data ramalan.....	197
Tabel 4.48 Keterangan komponen panel konfigurasi data ramalan.....	197
Tabel 4.49 Keterangan komponen panel tab data ramalan	198
Tabel 4.50 Keterangan komponen panel tab proses SVR-PPSO	199
Tabel 4.51 Keterangan komponen panel tab proses SVR-PPSO (lanjutan)	200
Tabel 4.52 Keterangan komponen panel tab hasil pelatihan SVR.....	201
Tabel 4.53 Keterangan komponen panel tab hasil pelatihan SVR (lanjutan)	202
Tabel 4.54 Keterangan komponen panel tab hasil pengujian SVR	203
Tabel 4.55 Keterangan komponen panel tab hasil pengujian SVR (lanjutan)	204
Tabel 4.56 Keterangan komponen panel tab hasil ramalan	205
Tabel 4.57 Rancangan uji coba batas parameter γ	206
Tabel 4.58 Rancangan uji coba batas parameter γ (lanjutan)	207
Tabel 4.59 Rancangan uji coba batas parameter λ	207
Tabel 4.60 Rancangan uji coba batas parameter C.....	207
Tabel 4.61 Rancangan uji coba batas parameter C (lanjutan)	208
Tabel 4.62 Rancangan uji coba batas parameter ϵ	208
Tabel 4.63 Rancangan uji coba batas parameter γ_k <i>kernel</i>	208
Tabel 4.64 Rancangan uji coba batas parameter γ_k <i>kernel</i> (lanjutan).....	209
Tabel 4.65 Rancangan uji coba batas parameter d <i>kernel</i>	209
Tabel 4.66 Rancangan uji coba jumlah iterasi SVR	209
Tabel 4.67 Rancangan uji coba jumlah iterasi SVR (lanjutan).....	210
Tabel 4.68 Rancangan uji coba jumlah partikel PPSO.....	210
Tabel 4.69 Rancangan uji coba jumlah iterasi PPSO	211
Tabel 4.70 Rancangan uji coba jumlah <i>slave</i> PSO	211
Tabel 4.71 Rancangan uji coba panjang periode runut waktu (<i>lag time</i>).....	212
Tabel 4.72 Rancangan uji coba pengaruh batas parameter ANOVA RBF <i>kernel</i>	212
Tabel 4.73 Rancangan uji coba peramalan runut waktu curah hujan tahunan..	213
Tabel 4.74 Rancangan uji coba peramalan runut waktu curah hujan musiman	214
Tabel 4.75 Rancangan uji coba variasi runut waktu ramalan pola vertikal	214
Tabel 4.76 Rancangan uji coba perbandingan peramalan runut waktu curah hujan menggunakan metode SVR, SVR-PPSO, dan regresi linier berganda	215

DAFTAR GAMBAR

Gambar 2.1 Langkah-langkah peramalan	38
Gambar 2.2 Proses peramalan.....	38
Gambar 2.3 Konseptual bingkai kerja sistem peramalan	40
Gambar 2.4 Jenis metode peramalan	41
Gambar 2.5 Contoh kasus sederhana peramalan runut waktu	48
Gambar 2.6 Plot data curah hujan bulanan Karangploso tahun 2000	49
Gambar 2.7 Hasil penjualan produk farmasi selama mingguan	49
Gambar 2.8 Populasi penduduk di U.S.A., 1790-1990.....	50
Gambar 2.9 Angka kematian karena kecelakaan U.S.A., 1973–1978.....	51
Gambar 2.10 Permasalahan deteksi sinyal	52
Gambar 2.11 Pola curah hujan di Indonesia	60
Gambar 2.12 Grafik pengaturan <i>margin loss</i> pada SVR linier dan fungsi $\mathcal{E}$ - <i>insentive loss function</i>	63
Gambar 2.13 Pemetaan Φ data ke dimensi tinggi pada ruang fitur untuk kasus klasifikasi	68
Gambar 2.14 Diagram alir algoritma sekuensial SVR untuk proses pelatihan	73
Gambar 2.15 Normalisasi ruang input dan pemetaan ke ruang fitur.....	76
Gambar 2.16 Ilustrasi pencarian solusi menggunakan metode PSO	81
Gambar 2.17 Algoritma PSO	82
Gambar 2.18 Ilustrasi pengendalian partikel pada ruang pencarian.....	89
Gambar 2.19 Variasi algoritma PSO	91
Gambar 2.20 Ilustrasi strategi paralel PSO	92
Gambar 2.21 Algoritma paralel PSO	93
Gambar 3.1 Diagram alir metodologi penelitian	94
Gambar 4.1 Struktur perancangan	99
Gambar 4.2 Strategi peramalan.....	100
Gambar 4.3 Pola curah hujan di wilayah Karangploso	107
Gambar 4.4 Pola curah hujan di wilayah Poncokusumo	108
Gambar 4.5 Pemilihan data latih dan data uji berdasarkan runut waktu	110
Gambar 4.6 Pola data horizontal tahun 2006 Poncokusumo	112

Gambar 4.7 Pola data vertikal tahun 2000-2012 Poncokusumo	114
Gambar 4.8 Diagram alir tahapan proses metode SVR-PPSO	118
Gambar 4.9 Diagram alir tahapan proses pelatihan SVR.....	127
Gambar 4.10 Diagram alir tahapan proses perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih	129
Gambar 4.11 Diagram alir algoritma proses perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih	130
Gambar 4.12 Rancangan algoritma proses perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih	131
Gambar 4.13 Diagram alir tahapan proses perhitungan normalisasi data latih.	132
Gambar 4.14 Diagram alir algoritma proses perhitungan normalisasi data latih	133
Gambar 4.15 Rancangan algoritma proses perhitungan normalisasi data latih	133
Gambar 4.16 Diagram alir tahapan proses perhitungan matriks Hessian $[R]_{ij}$ data latih.....	134
Gambar 4.17 Diagram alir algoritma proses perhitungan matriks Hessian $[R]_{ij}$ data latih.....	135
Gambar 4.18 Rancangan algoritma proses perhitungan matriks Hessian $[R]_{ij}$ data latih.....	136
Gambar 4.19 Diagram alir tahapan proses perhitungan nilai α_i^* dan α_i	137
Gambar 4.20 Diagram alir algoritma proses perhitungan nilai α_i^* dan α_i	139
Gambar 4.21 Rancangan algoritma proses perhitungan nilai α_i^* dan α_i	140
Gambar 4.22 Diagram alir tahapan proses perhitungan regresi data latih	144
Gambar 4.23 Diagram alir algoritma proses perhitungan regresi data latih.....	145
Gambar 4.24 Rancangan algoritma proses perhitungan regresi data latih.....	146
Gambar 4.25 Diagram alir tahapan proses evaluasi kinerja peramalan.....	147
Gambar 4.26 Diagram alir algoritma proses evaluasi kinerja peramalan.....	148
Gambar 4.27 Rancangan algoritma proses evaluasi kinerja peramalan	148
Gambar 4.28 Diagram alir tahapan proses pengujian SVR	149
Gambar 4.29 Diagram alir tahapan proses perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih dan data uji	150

Gambar 4.30 Diagram alir algoritma proses perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih dan data uji	151
Gambar 4.31 Rancangan algoritma proses perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih dan data uji	152
Gambar 4.32 Diagram alir tahapan proses perhitungan normalisasi data uji....	153
Gambar 4.33 Diagram alir algoritma proses perhitungan normalisasi data uji..	154
Gambar 4.34 Rancangan algoritma proses perhitungan normalisasi data uji....	155
Gambar 4.35 Diagram alir tahapan proses perhitungan matriks Hessian data uji	156
Gambar 4.36 Diagram alir algoritma proses perhitungan matriks Hessian $[R]_{ij}$ data uji.....	157
Gambar 4.37 Rancangan algoritma proses perhitungan matriks Hessian $[R]_{ij}$ data uji.....	157
Gambar 4.38 Diagram alir tahapan proses perhitungan regresi data uji	158
Gambar 4.39 Diagram alir algoritma proses perhitungan regresi data uji	160
Gambar 4.40 Rancangan algoritma proses perhitungan regresi data uji.....	160
Gambar 4.41 Diagram alir tahapan proses <i>slave</i> PSO.....	163
Gambar 4.42 Diagram alir algoritma proses <i>slave</i> PSO	164
Gambar 4.43 Rancangan algoritma proses <i>slave</i> PSO	164
Gambar 4.44 Diagram alir tahapan proses <i>master</i> PSO	165
Gambar 4.45 Diagram alir algoritma proses <i>master</i> PSO	166
Gambar 4.46 Rancangan algoritma proses <i>master</i> PSO	167
Gambar 4.47 Diagram alir tahapan proses PSO.....	169
Gambar 4.48 Diagram alir algoritma proses PSO	171
Gambar 4.49 Rancangan algoritma proses PSO	172
Gambar 4.50 Diagram alir tahapan proses evaluasi <i>fitness</i> tiap partikel	173
Gambar 4.51 Diagram alir algoritma proses evaluasi <i>fitness</i> tiap partikel.....	174
Gambar 4.52 Rancangan algoritma proses evaluasi <i>fitness</i> tiap partikel.....	174
Gambar 4.53 Diagram alir tahapan proses perhitungan <i>fitness</i>	175
Gambar 4.54 Diagram alir algoritma proses perhitungan <i>fitness</i> partikel	175
Gambar 4.55 Rancangan algoritma proses perhitungan <i>fitness</i> partikel	176
Gambar 4.56 Diagram alir tahapan proses penentuan <i>global best</i>	177

Gambar 4.57 Diagram alir algoritma proses penentuan <i>global best</i>	178
Gambar 4.58 Rancangan algoritma proses penentuan <i>global best</i>	178
Gambar 4.59 Diagram alir tahapan proses perhitungan kecepatan partikel	179
Gambar 4.60 Diagram alir proses perhitungan kecepatan partikel pada dimensi y	180
Gambar 4.61 Rancangan algoritma proses perhitungan kecepatan partikel	181
Gambar 4.62 Diagram alir tahapan proses perhitungan posisi partikel	182
Gambar 4.63 Diagram alir algoritma proses perhitungan posisi partikel pada dimensi y	183
Gambar 4.64 Rancangan algoritma proses perhitungan posisi partikel.....	184
Gambar 4.65 Diagram alir tahapan proses penentuan <i>local best</i>	185
Gambar 4.66 Diagram alir algoritma proses penentuan <i>local best</i>	186
Gambar 4.67 Rancangan algoritma proses penentuan <i>local best</i>	187
Gambar 4.68 Diagram alir pengurutan partikel <i>slave</i> PSO	189
Gambar 4.69 Diagram alir algoritma proses pengurutan partikel <i>slave</i> PSO	191
Gambar 4.70 Rancangan algoritma proses pengurutan partikel <i>slave</i> PSO	192
Gambar 4.71 Model alur sistem peramalan runut waktu curah hujan menggunakan metode SVR-PPSO	193
Gambar 4.72 Tampilan awal sistem.....	195
Gambar 4.73 Detail tampilan daftar pilihan konfigurasi sistem	196
Gambar 4.74 Detail tampilan panel tab data ramalan	198
Gambar 4.75 Detail tampilan panel tab proses SVR-PPSO	199
Gambar 4.76 Detail Tampilan panel tab hasil pelatihan SVR dengan pratinjau panel tab hasil ramalan.....	201
Gambar 4.77 Detail tampilan panel tab hasil pelatihan SVR dengan pratinjau panel tab nilai α_i^* dan α_i	201
Gambar 4.78 Detail tampilan panel tab hasil pelatihan SVR dengan pratinjau panel tab hasil ramalan.....	203
Gambar 4.79 Detail tampilan panel tab hasil pelatihan SVR dengan pratinjau panel tab hasil konversi curah hujan bulanan	203
Gambar 4.80 Detail tampilan panel tab hasil ramalan	205
Gambar 5.1 Struktur implementasi sistem	216
Gambar 6.1 Struktur Pengujian.....	232

Gambar 6.2 Grafik hasil uji coba batas parameter γ	233
Gambar 6.3 Grafik hasil uji coba batas parameter λ	234
Gambar 6.4 Grafik hasil uji coba batas parameter C	235
Gambar 6.5 Grafik hasil uji coba batas parameter ε	237
Gambar 6.6 Grafik hasil uji coba batas parameter γ_k <i>kernel</i>	238
Gambar 6.7 Grafik hasil uji coba batas parameter d <i>kernel</i>	239
Gambar 6.8 Grafik hasil uji coba jumlah iterasi SVR.....	240
Gambar 6.9 Grafik waktu eksekusi berdasarkan jumlah iterasi SVR	241
Gambar 6.10 Grafik hasil uji coba jumlah partikel PPSO	242
Gambar 6.11 Grafik waktu eksekusi berdasarkan jumlah partikel PPSO	243
Gambar 6.12 Grafik hasil uji coba jumlah iterasi PPSO	244
Gambar 6.13 Grafik waktu eksekusi berdasarkan jumlah iterasi PPSO.....	244
Gambar 6.14 Grafik hasil uji coba jumlah <i>slave</i> PSO	245
Gambar 6.15 Grafik waktu eksekusi berdasarkan jumlah <i>slave</i> PSO.....	246
Gambar 6.16 Grafik hasil uji coba panjang periode runut waktu (<i>lag time</i>)	247
Gambar 6.17 Grafik hasil uji coba γ_k 0.0001 – 0.1 dan d 1.....	249
Gambar 6.18 Grafik hasil uji coba γ_k 0.0001 – 0.1 dan d 2.....	249
Gambar 6.19 Grafik hasil uji coba γ_k 0.0001 – 0.1 dan d 3.....	250
Gambar 6.20 Grafik hasil uji coba γ_k 1 - 100 dan d 1.....	251
Gambar 6.21 Grafik hasil uji coba γ_k 1 - 100 dan d 2.....	251
Gambar 6.22 Grafik hasil uji coba γ_k 0.0001 – 0.1 dan d 3.....	252
Gambar 6.23 Grafik hasil peramalan runut waktu curah hujan tahun 2015.....	254
Gambar 6.24 Grafik hasil peramalan runut waktu curah hujan musiman tahun 2014 – 2015.....	255
Gambar 6.25 Grafik hasil peramalan runut waktu curah hujan dasarian 2 tahun 2013, 2014, dan 2015.....	257
Gambar 6.26 Grafik hasil perbandingan peramalan runut waktu curah hujan menggunakan metode SVR, SVR-PPSO, dan regresi linier berganda	259

DAFTAR PERSAMAAN

Persamaan (2.1) Konsep evaluasi kinerja peramalan	44
Persamaan (2.2) <i>Mean Error</i> (ME)	45
Persamaan (2.3) <i>Mean Absolute Error</i> (MAE)	45
Persamaan (2.4) <i>Mean Square Error</i> (MSE)	46
Persamaan (2.5) <i>Root Mean Square Error</i> (RMSE)	46
Persamaan (2.6) <i>Mean Absolute Percentage Error</i> (MAPE)	46
Persamaan (2.7) Normalisasi <i>Min-max</i>	54
Persamaan (2.8) Pengukuran banyaknya curah hujan	57
Persamaan (2.9) Pengukuran kepadatan hujan	57
Persamaan (2.10) Pengukuran intensitas hujan	59
Persamaan (2.11) Contoh pengukuran intensitas hujan	59
Persamaan (2.12) Contoh set data latih	61
Persamaan (2.13) Fungsi dasar regresi linier	62
Persamaan (2.14) <i>Convex optimization problem</i>	62
Persamaan (2.15) Pengembangan <i>convex optimization problem</i>	63
Persamaan (2.16) Penggunaan <i>ϵ-insentive loss function</i>	63
Persamaan (2.17) Konsep <i>lagrangian multipliers</i>	64
Persamaan (2.18) Konsep vektor bobot W	64
Persamaan (2.19) Fungsi <i>Support Vector Regression</i> (SVR)	65
Persamaan (2.20) Penentuan nilai b	65
Persamaan (2.21) Penentuan nilai b	65
Persamaan (2.22) Konsep vektor bobot W untuk kasus non linier	65
Persamaan (2.23) Fungsi SVR untuk kasus non linier	66
Persamaan (2.24) Fungsi SVR usulan Vijayakumar	66
Persamaan (2.25) Fungsi <i>linear kernel</i>	69
Persamaan (2.26) <i>Homogeneous polynomial kernel</i>	69
Persamaan (2.27) <i>Non-homogeneous polynomial kernel</i>	69
Persamaan (2.28) Pengaturan nilai parameter C pada <i>polynomial kernel</i>	70
Persamaan (2.29) Fungsi <i>gaussian kernel</i> menggunakan σ	70
Persamaan (2.30) Fungsi <i>gaussian kernel</i> menggunakan γ	70

Persamaan (2.31) Fungsi <i>exponential kernel</i>	71
Persamaan (2.32) Fungsi <i>laplacian kernel</i>	71
Persamaan (2.33) Fungsi ANOVA RBF <i>kernel</i>	71
Persamaan (2.34) Matriks <i>Hessian</i> $[R]_{ij}$	74
Persamaan (2.35) Perhitungan nilai kesalahan error proses pelatihan.....	74
Persamaan (2.36) Perhitungan nilai $\delta\alpha_i^*$	74
Persamaan (2.37) Perhitungan nilai $\delta\alpha_i$	74
Persamaan (2.38) Batas nilai <i>learning rate</i>	75
Persamaan (2.39) Perhitungan nilai α_i^*	75
Persamaan (2.40) Perhitungan nilai α_i	75
Persamaan (2.41) Normalisasi pada ruang fitur SVR	76
Persamaan (2.42) Normalisasi pada ruang fitur SVR	76
Persamaan (2.43) Normalisasi ANOVA RBF <i>kernel</i>	77
Persamaan (2.44) Inisialisasi awal posisi partikel	79
Persamaan (2.45) Inisialisasi awal kecepatan partikel.....	79
Persamaan (2.46) Pembatas dimensi posisi partikel	79
Persamaan (2.47) Pembaruan kecepatan partikel	80
Persamaan (2.48) Penentuan <i>local best</i>	80
Persamaan (2.49) Penentuan <i>global best</i>	80
Persamaan (2.50) Perhitungan perpindahan posisi partikel	81
Persamaan (2.51) Pembatasan kecepatan partikel	83
Persamaan (2.52) Pembatasan kecepatan partikel menggunakan δ	84
Persamaan (2.53) Alternatif Pembatasan kecepatan partikel menggunakan δ	84
Persamaan (2.54) Pembaruan kecepatan partikel menggunakan W	84
Persamaan (2.55) Penentuan variasi nilai bobot inersia W	85
Persamaan (2.56) Penentuan probabilitas keberhasilan partikel	87
Persamaan (2.57) Perhitungan <i>fitness</i> partikel.....	87
Persamaan (2.58) Perhitungan presentase tingkat keberhasilan partikel	87
Persamaan (2.59) Indikator dalam penyesuaian nilai bobot inersia $w(t+1)$	88
Persamaan (2.60) <i>Adaptive Inertia Weight</i> PSO (AIWPSO)	88

Persamaan (2.61) Pengendalian partikel menggunakan metode *nearest* 89

Persamaan (2.62) Pengendalian partikel menggunakan metode *random back* .. 90



DAFTAR KODE PROGRAM

Kode Program 5.1 Implementasi proses konfigurasi data	217
Kode Program 5.2 Implementasi proses perhitungan matriks Hessian data latih	218
Kode Program 5.3 Implementasi proses perhitungan normalisasi data latih	218
Kode Program 5.4 Implementasi proses perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih	219
Kode Program 5.5 Implementasi proses perhitungan nilai α_i^* dan α_i	220
Kode Program 5.6 Implementasi proses perhitungan regresi data latih.....	221
Kode Program 5.7 Implementasi proses evaluasi kinerja peramalan	221
Kode Program 5.8 Implementasi proses perhitungan regresi data uji.....	222
Kode Program 5.9 Implementasi proses perhitungan matriks Hessian data uji	223
Kode Program 5.10 Implementasi proses perhitungan normalisasi data uji	224
Kode Program 5.11 Implementasi proses perhitungan ANOVA RBF <i>kernel</i> menggunakan data latih dan data uji	224
Kode Program 5.12 Proses konfigurasi <i>slave</i> PSO.....	225
Kode Program 5.13 Implementasi proses <i>slave</i> PSO	226
Kode Program 5.14 Implementasi proses pengurutan partikel <i>slave</i> PSO	226
Kode Program 5.15 Implementasi proses <i>master</i> PSO	227
Kode Program 5.16 Implementasi proses PSO	228
Kode Program 5.17 Implementasi proses evaluasi <i>fitness</i> tiap partikel.....	228
Kode Program 5.18 Implementasi proses penentuan <i>global best</i>	229
Kode Program 5.19 Implementasi proses perhitungan kecepatan partikel pada dimensi y	229
Kode Program 5.20 Implementasi proses perhitungan posisi partikel.....	230
Kode Program 5.21 Implementasi proses perhitungan <i>local best</i>	230
Kode Program 5.22 Implementasi proses perhitungan <i>fitness</i> partikel	231

DAFTAR LAMPIRAN

LAMPIRAN A DATA CURAH HUJAN HARIAN WILAYAH KARANGPLOSO 271

LAMPIRAN B HASIL PENGUJIAN 274



BAB 1 PENDAHULUAN

Pada bab ini membahas pendahuluan yang terdiri dari latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

1.1 Latar Belakang

Informasi cuaca dapat digunakan masyarakat untuk mengantisipasi kejadian buruk yang menimbulkan bencana, sehingga merugikan jiwa dan harta (BMKG Stasiun Klimatologi Karangploso, 2015a). Seperti kejadian hujan lebat disertai angin puting beliung di wilayah kecamatan Karangploso Malang pada tanggal 28 januari 2013 yang menyebabkan sekitar 500 rumah warga mengalami kerusakan dan beberapa korban luka (Media Indonesia, 2013). Berdasarkan pengamatan Badan Meteorologi Klimatologi dan Geofisika (BMKG) stasiun klimatologi Karangploso Malang, curah hujan di wilayah Karangploso tercatat di atas 50 milimeter sejak awal dasarian (10 hari) pada tanggal 1-10 desember 2012 sampai dasarian 3 bulan januari 2013. Banyaknya curah hujan yang terjadi pada satu dasarian secara berkelanjutan di atas 50 milimeter dapat ditetapkan sebagai musim hujan (BMKG Stasiun Klimatologi Karangploso, 2015b). Menurut bupati malang Rendra Kresna wilayah kecamatan Karangploso, Poncokusumo, Wagir, dan Tumpang merupakan daerah rawan terjadi puting beliung pada saat musim hujan (Sukarelawati, 2012).

Cuaca merupakan kondisi sementara rata-rata udara yang menjadi hal utama untuk penentuan iklim di suatu wilayah berupa rata-rata cuaca pada jangka waktu tertentu, sedangkan curah hujan menjadi unsur yang berpengaruh terhadap perubahan iklim di suatu wilayah (Ramadhani, et al., 2014). Selain itu, curah hujan juga menjadi variabel penting untuk peramalan cuaca yang dapat mempengaruhi kehidupan (Nayak, et al., 2013). Oleh sebab itu, perlu dilakukan peramalan curah hujan yang akan terjadi di waktu mendatang untuk dapat mendukung penentuan cuaca dan iklim di suatu wilayah pada waktu tertentu (Montgomery, et al., 2015).

Pada dasarnya ramalan merupakan upaya untuk mengetahui terhadap peristiwa atau kejadian di masa depan yang akan terjadi (Montgomery, et al., 2015). Peramalan dapat dilakukan berdasarkan pendekatan secara objektif dengan melakukan komputasi matematika pada data historis runut waktu (Singh, 2016). Menurut Somvanshi, et al. (2006) iklim dan curah hujan merupakan fenomena yang rumit dan bersifat non linier. Dengan menerapkan kecerdasan buatan seperti metode Jaringan Syaraf Tiruan (JST) dapat digunakan untuk memprediksi objek yang bersifat non linier dan menjadi alternatif yang menarik dari metode statistika (Somvanshi, et al., 2006). Pada penelitian Somvanshi, et al. (2006) memodelkan dan memprediksi pola perilaku fenomena curah hujan menggunakan metode JST yang memiliki nilai evaluasi *mean absolute error* (MAE) sebesar 0.9198, sedangkan penggunaan metode *Autoregressive Integrated Moving Average* (ARIMA) sebesar 24.3888.

Terdapat penelitian lain yang mengevaluasi kinerja peramalan runut waktu pada data hasil penjualan barang bulanan baik model linier dan non linier menggunakan metode *Support Vector Regression* (SVR), *Multilayer Perceptrons* (MLP) dan metode statistika (Crone, et al., 2006b). Berdasarkan penelitian Crone, et al. (2006a) membuktikan bahwa metode SVR bernilai *mean ranks* 1.07 memiliki peluang baik untuk peramalan *time series* model nonlinier dibandingkan metode JST yang bernilai *mean ranks* 1.87, serta mengungguli dari metode statistika yang hanya memiliki akurasi terbaik bernilai *mean ranks* 1.67 pada model linier (Crone, et al., 2006b).

Metode SVR merupakan pengembangan dari metode *Support Vector Machine* (SVM) untuk memecahkan masalah regresi yang bersifat non linier (Alwee, et al., 2013). Metode SVR menggunakan dasar teori statistika yang dikembangkan oleh Vapnik dan koleganya sebagai metode alternatif selain JST (Smola & Schölkopf, 2004). Untuk meningkatkan akurasi regresi SVR diperlukan penentuan nilai parameter yang tepat (Zhao & Wang, 2010). Zhao dan Wang (2010) menerapkan optimasi parameter SVR menggunakan algoritma *Particle Swarm Optimization* (PSO) dapat meningkatkan akurasi peramalan curah hujan di Guangxi China dengan nilai *Mean Absolute Percentage Error* (MAPE) 3.85%.

Algoritma PSO diperkenalkan oleh Kennedy dan Eberhart (1995) yang terinspirasi dari tingkah laku sekumpulan makhluk hidup. Algoritma PSO sering diterapkan pada permasalahan optimasi, sehingga terus dikembangkan maupun dimodifikasi (Rini, et al., 2011). Seperti penelitian yang dilakukan oleh Xun W., et al. (2015) menerapkan strategi *Parallel Particle Swarm Optimization* (PPSO) untuk optimasi parameter (C, σ^2) pada metode SVR untuk meramalkan arus penumpang kereta api di China berdasarkan data tahunan 1987-2010. Metode SVR-PPSO secara efektif dapat meningkatkan akurasi prediksi dibandingkan dengan metode SVR-PSO. Dengan menerapkan strategi *Parallel* PSO (PPSO) dapat meningkatkan pencarian solusi global optimum (Deep, et al., 2010)

Berdasarkan pembahasan dari penelitian sebelumnya, metode SVR-PPSO memiliki peluang yang baik diterapkan untuk peramalan curah hujan. Dengan berlatar belakang demikian, sehingga penulis mengajukan penelitian untuk peramalan runut waktu curah hujan salah satu wilayah kecamatan di kabupaten Malang menggunakan metode SVR-PPSO.

1.2 Rumusan Masalah

Dengan berlatar belakang demikian dapat diperoleh rumusan masalah yang dapat dijadikan sebagai pedoman dalam penelitian ini adalah sebagai berikut:

1. Bagaimana melakukan peramalan runut waktu curah hujan?
2. Bagaimana menerapkan metode SVR-PPSO untuk peramalan runut waktu curah hujan?
3. Bagaimana kinerja peramalan peramalan runut waktu curah hujan menggunakan metode SVR-PPSO?

1.3 Tujuan

Penelitian ini memiliki tujuan pencapaian diantaranya:

1. Melakukan peramalan runut waktu curah hujan.
2. Menerapkan metode SVR-PPSO untuk peramalan runut waktu curah hujan.
3. Mengetahui kinerja peramalan runut waktu curah hujan menggunakan metode SVR-PPSO.

1.4 Manfaat

Manfaat pada penelitian ini diharapkan agar dapat menambah wawasan penulis dalam melakukan peramalan runut waktu curah hujan menggunakan metode SVR-PPSO. Dengan adanya penelitian ini sehingga dapat menghasilkan manfaat positif kepada masyarakat umum khususnya diruang lingkup riset penelitian, pengetahuan dan teknologi.

1.5 Batasan Masalah

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

1. Data curah hujan diperoleh dari Badan Meteorologi Klimatologi dan Geofisika (BMKG) stasiun klimatologi Karangploso Malang Jawa Timur, Indonesia.
2. Data curah hujan yang dibutuhkan meliputi wilayah di Karangploso dan Poncokusumo Malang Jawa Timur, Indonesia dengan rentang waktu tahun 2000-2015 pada masing-masing wilayah.
3. Data curah hujan yang digunakan dengan rentang waktu per dasarian selama 10 hari, sehingga selama per bulan terdapat 3 dasarian dan per tahun terdapat 36 dasarian.
4. Data curah hujan yang digunakan dengan satuan millimeter per dasarian.
5. Data curah hujan dipilih dan digunakan pada wilayah tertentu berdasarkan hasil analisis yang dilakukan pada penelitian ini.
6. Peramalan runut waktu curah hujan menggunakan menggunakan metode *epsilon* - SVR.
7. *Kernel trick* yang digunakan pada metode SVR adalah *Analysis of Variance* (ANOVA) *Radial Basis Function* (RBF) *kernel*.
8. Algoritma pelatihan metode SVR menggunakan *sequential* SVR yang diusulkan oleh Vijayakumar dan Wu (1999).
9. Evaluasi kinerja peramalan mengacu pada nilai kesalahan (*error*) berdasarkan nilai *Root Mean Square Error* (RMSE).
10. Optimasi parameter SVR yang digunakan menggunakan strategi metode PPSO.
11. Varian metode PSO yang digunakan merupakan variasi PSO terhadap parameter pembatas dimensi partikel dan bobot inersia (*Inertia Weight*).
12. Strategi paralel diterapkan pada metode PSO yang dapat melakukan komputasi berjalan secara bersamaan.

1.6 Sistematika Pembahasan

Pada penyusunan skripsi ini terdapat penulisan yang terstruktur seperti berikut:

- **BAB 1 PENDAHULUAN**

Menguraikan hal umum terkait dengan peramalan runtu waktu curah hujan menggunakan metode SVR-PPSO secara sistematis dan berkesinambungan. Penulisan dimulai dari latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika penulisan.

- **BAB 2 LANDASAN KEPUSTAKAAN**

Membahas kajian pustaka dan dasar teori peramalan runtu waktu curah hujan menggunakan metode SVR-PPSO berdasarkan referensi-referensi yang saling berkaitan dalam penelitian ini.

- **BAB 3 METODOLOGI PENELITIAN**

Menguraikan langkah-langkah penelitian yang akan dilakukan secara umum dalam menerapkan metode SVR-PPSO untuk peramalan runtu waktu curah hujan.

- **BAB 4 PERANCANGAN**

Membahas mengenai perancangan strategi peramalan dan sistem yang akan dibangun dalam implementasi metode SVR-PPSO untuk peramalan runtu waktu curah hujan. Pada bab ini berisi pembahasan, seperti formulasi strategi peramalan dan metode, perancangan tampilan antarmuka pengguna, algoritma dan alur program sistem yang akan diimplementasikan.

- **BAB 5 IMPLEMENTASI SISTEM**

Membahas mengenai implementasi sistem berdasarkan hasil perancangan sistem yang telah dilakukan. Pada bab implementasi secara mendasar akan membahas komponen utama dari algoritma metode SVR-PPSO menggunakan bahasa pemrograman java berdasarkan perancangan sistem.

- **BAB 6 PENGUJIAN**

Menguraikan tentang pengujian sistem yang dapat melakukan peramalan curah hujan menggunakan metode SVR-PPSO. Serta menguraikan pembahasan hasil pengujian terhadap sistem yang dibangun dalam peramalan runtu waktu curah hujan.

- **BAB 7 PENUTUP**

Pada bab terakhir ini menguraikan kesimpulan yang diperoleh terhadap penelitian ini dan menyertakan saran yang dapat digunakan untuk pengembangan penelitian terkait selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisi kajian pustaka terhadap penelitian-penelitian terkait yang berhubungan dengan penelitian dan pembahasan tentang dasar teori yang dapat mendukung dalam penyusunan penelitian ini.

2.1 Kajian Pustaka

Terdapat penelitian-penelitian sebelumnya yang berkaitan dengan penelitian ini. Beberapa sebagian penelitian sebelumnya secara hal umum telah dibahas pada bab pendahuluan latar belakang. Dengan membandingkan dari penelitian sebelumnya diperoleh persamaan dan perbedaan terhadap penelitian yang diajukan penulis. Sehingga dapat digunakan sebagai indikator pembeda dari penelitian sebelumnya. Berikut merupakan daftar penelitian terkait sebelumnya yang dikaji pada penelitian ini.

Berdasarkan kajian penelitian terkait sebelumnya terdapat penggunaan metode jaringan syaraf tiruan yang dilakukan oleh Cahyani, et al. (2014) untuk peramalan curah hujan. Dengan menggunakan Algoritma *Al-Alaoui Backpropagation* terdapat nilai *Mean Square Error* (MSE) 0.00313611. Data curah hujan yang digunakan pada penelitian tersebut pada periode tahun januari 2011 sampai juni 2012 yang diperoleh dari Badan Meteorologi Klimatologi dan Geofisika (BMKG) Malang. Berdasarkan hasil penelitian tersebut metode jaringan syaraf tiruan memiliki akurasi 100% terhadap pengujian pada bulan juli saat musim kemarau (Cahyani, et al., 2014).

Penelitian terkait lainnya juga menggunakan jaringan syaraf tiruan yang dihibridisasi dengan metode statistika. Seperti penelitian yang dilakukan oleh Diani, et al. (2013) dengan melakukan pemodelan jaringan syaraf tiruan berdasarkan metode *Generalized Autoregressive* (GSTAR) dan *Vector Autoregressive* (VAR). Data curah hujan yang digunakan di Tumpang, Pujon, dan Batu kabupaten Malang periode tahun 1996 sampai Pebruari 2012 yang diperoleh dari BMKG Malang. Berdasarkan penelitian tersebut metode *Vector Autoregressive Neural Network* (VAR-NN) dapat meramalkan hasil terbaik dibandingkan metode lainnya dengan menggunakan 1 neuron dalam hidden layer. Akurasi berdasarkan nilai evaluasi *Root Mean Square Error* (RMSE) terendah metode *Generalized Autoregressive Neural Network* (GSTAR-NN) yang bernilai 4.590 pada peramalan curah hujan di Batu, Malang (Diani, et al., 2013).

Penelitian lain menerapkan penggunaan metode kecerdasan buatan maupun metode statistika untuk melakukan peramalan. Seperti penelitian yang dilakukan oleh Trafalis, et al. (2005) untuk peramalan curah hujan memanfaatkan data *time series* curah hujan yang diperoleh dari stasiun pengukur hujan di Chandler, Oklahoma Mesonet untuk tahun kalender 1998. Dengan menggunakan metode *Artificial Neural Networks* (ANNs), *Support Vector Regression* (SVR), *Least-Squares Support Vector Regression* (LS-SVR), *Linear Regression* (LR) dan *Rain Rate* (RR) formula. Berdasarkan akurasi yang diperoleh menggunakan rumus MSE metode

ANNs bernilai 0.3383, LR bernilai 0.4665, SVR bernilai 0.3099, LS-SVR bernilai 0.2709, dan RR bernilai 0.3728 (Trafalis, et al., 2005).

Terdapat penelitian lain yang dilakukan oleh Alwee, et al. (2013) melakukan peramalan tingkat kejahatan berdasarkan indikator ekonomi di Amerika Serikat. Dengan menggunakan metode gabungan SVR dan *Autoregressive Integrated Moving Average* (ARIMA) dengan optimasi parameter menggunakan algoritma *Particle Swarm Optimization* (PSO). Berdasarkan penelitian tersebut terdapat hipotesis terhadap metode gabungan model PSOSVR-PSOARIMA yang dapat menghasilkan kesalahan kecil dibandingkan dengan model individu dan model hibrida PSOSVR-ARIMA. Akurasi di dapat menggunakan perhitungan rumus statistika RMSE yang menghasilkan PSOSVR-PSOARIMA bernilai *error* 10.973 dan PSOSVR-ARIMA bernilai *error* 11.704 (Alwee, et al., 2013).

Pada penelitian yang telah dijelaskan sebelumnya membuktikan penggunaan metode SVR untuk peramalan memiliki kinerja yang baik dengan tingkat *error* yang rendah. Oleh sebab itu, metode SVR sering digunakan untuk melakukan peramalan. Seperti penelitian lain yang dilakukan Prahutama dan Yasin (2015) melakukan peramalan curah hujan mingguan di kota Semarang. Pada penelitian tersebut menggunakan metode SVR dan *quadratic loss function* dengan *linear kernel*, *polynomial*, dan *Radial Basis Function* (RBF). Data yang digunakan merupakan data tahunan curah hujan di Semarang selama periode tahun 2009 sampai 2013. Berdasarkan data curah hujan mingguan di kota Semarang menunjukkan tren yang fluktuatif, non stasioner. Namun, pada metode SVR dengan menggunakan *kernel polynomial* sebesar 71.61% mampu meramalkan curah hujan mingguan dengan tepat (Prahutama & Yasin, 2015).

Metode SVM kasus regresi juga dapat meramalkan hujan didaerah tropis topan dengan melakukan analisis data dari 73 daerah tropis topan yang mempengaruhi daerah aliran sungai pada tahun 1990-2007 di Shimen Reservoir, Taiwan (Wei, 2012). Analisis berupa karakteristik iklim, trek topan dan jumlah curah hujan per jam di Shimen Reservoir, Taiwan. Dengan melakukan perbandingan metode *Mesoscale Model* (MM5), *Gaussian Kernel SVMs* (GSVMs), *Wavelet Kernel SVMs* (WSVMs) dan *linear regressions* (*regression*). Penelitian yang dilakukan Wei (2012) menyatakan metode WSVM dapat memberikan peramalan yang baik dibandingkan metode lainnya yang diteliti. Berdasarkan analisis bahwa 5 model menghasilkan peramalan curah hujan yang kumulatif, sedangkan model statistika cenderung terlalu rendah. Akurasi berdasarkan ilai evaluasi RMSE metode GSVM bernilai 0.288, WSVM bernilai 0.257 dan *regression* bernilai 0.296 (Wei, 2012).

Metode SVR juga dapat melakukan pemodelan *downscaling* untuk peramalan curah hujan (Sanusi, et al., 2014). Dengan menggunakan data GCM sebagai penjelasan variabel yang diperoleh dari *Climate Information Tool Kit* (CLIK) APEC *Climate Center* (APCC) untuk membangun model prediksi pada 3 bulan yang berbeda: Mei, Juni, dan Juli dari tahun 1982-2008 (27 tahun), serta menggunakan data lokal curah hujan (digunakan sebagai variabel respon) diperoleh dari nilai rata-rata musiman curah hujan setiap stasiun curah hujan di Indramayu pada 15 stasiun pengamatan. Data pengamatan curah hujan yang digunakan di Indramayu

pada 3 bulan: Mei, Juni, Juli dari tahun 1982-2008 (27 tahun) setiap stasiun hujan. Berdasarkan korelasi dan nilai *error* RMSE pada fungsi *kernel* linear lebih baik daripada menggunakan fungsi *kernel* RBF. Terdapat korelasi terbaik pada peramalan curah hujan di Cikedung dengan nilai korelasi 0.744 dan nilai *error* RMSE 23.937. *Kernel* Linier menghasilkan prediksi yang baik daripada RBF *kernel* (Sanusi, et al., 2014).

Berdasarkan penelitian yang telah dijelaskan sebelumnya bahwa metode SVR sering digunakan untuk melakukan peramalan curah hujan. Namun, akurasi yang diperoleh terkadang masih kurang baik dan perlu adanya optimasi parameter pada fungsi *kernel* dengan menggunakan teknik algoritma optimasi lain (Sanusi, et al., 2014). Seperti penelitian yang dilakukan oleh Pai dan Hong (2007) melakukan peramalan curah hujan menggunakan metode SVR dan *Recurrent* SVR (RSVR) yang dioptimasi menggunakan algoritma genetika, serta perbandingan dengan metode *General Regression Neural Networks* (GRNN). Metode SVR terbukti memiliki hasil peramalan yang akurat berdasarkan perhitungan rumus *Mean Absolute Percentage Error* (MAPE). Metode RSVRG bernilai 1.381, metode SVRG bernilai 1.559 dan metode GRNN bernilai 1.654 (Pai & Hong, 2007).

Terdapat optimasi parameter SVR dengan algoritma *tabu search* untuk deteksi insiden pada jalan tol atau jalan bebas hambatan (Yao, et al., 2014). Dengan menggunakan data jalan tol dari kota Shenyang ke Dalian dan jalan tol utama di Cina daratan dan Total panjang adalah 400 km. Serta data jalan tol Heda di provinsi Liaoning dari Jixi ke Mudanjiang dengan total panjang 1.390 km. Berdasarkan penelitian tersebut algoritma *tabu search* secara efektif dapat mengoptimasi parameter terbaik pada metode SVR, sehingga mengungguli dari metode *Artificial Neural Networks* (ANNs). Akurasi didapatkan berdasarkan perhitungan *Mean Time-To-Detect* (MTTD) metode SVR bernilai 3.52, sedangkan metode ANNs bernilai 4.63 (Yao, et al., 2014).

Optimasi parameter SVR terus dikembangkan untuk menghasilkan yang terbaik, sehingga banyak variasi dalam penggunaan algoritma optimasi. Seperti penelitian yang dilakukan Adhani, et al. (2013) melakukan optimasi parameter SVR menggunakan algoritma *grid search* untuk pemodelan peramalan curah hujan. Dengan menggunakan data curah hujan selama musim kering di Indramayu menggunakan *Southern Oscillation Index* (SOI), NINO 3.4 SST dan data aktual pada jangka tahun 1970-2010. Data SOI and NINO SST diperoleh dari Australian *Bureau of Meteorology* (BOM), sedangkan data aktual curah hujan diperoleh BMKG stasiun cuaca di kabupaten Indramayu. Pada penelitian Adhani, et al. (2013) dilakukan pula terhadap penggunaan *kernel* metode SVR. Berdasarkan hasil penelitian tersebut penerapan metode SVR dengan optimasi algoritma *grid search* dan penggunaan *kernel* RBF dapat meramalkan curah hujan pada musim kemarau dengan korelasi tertinggi nilai koefisien, dengan nilai terendah *Normalized Root Mean Square Error* (NRMSE) bernilai 1.73. Dibandingkan dengan *kernel polynomial* memiliki kinerja dengan koefisien korelasi rendah dan nilai kesalahan yang tinggi bernilai 357.54, sedangkan *kernel* linier memiliki nilai kesalahan 4.37 (Adhani, et al., 2013).

Terdapat penggunaan algoritma optimasi lain untuk optimasi parameter SVR menggunakan algoritma *Simulated Annealing* (SA) (Hong & Pai, 2007). Dengan menggunakan data curah hujan pada daerah aliran sungai di Wu-Tu, Taiwan utara. Data curah hujan diperoleh Volume per jam (dari bulan agustus 1985 sampai bulan gustus 1997) diperoleh dari tiga alat pengukur hujan. Metode SVR-SA dibandingkan metode *Recurrent Neural Network* (RNN) memiliki nilai validasi *Normalized Mean Squared Error* (NMSE) yang rendah bernilai 0.3963. Selain itu, dapat melakukan peramalan curah hujan selama musim topan dengan tepat. Dengan parameter terbaik σ bernilai 0.05206, C bernilai 28.867 dan ϵ bernilai 0.87811 (Hong & Pai, 2007).

Penelitian lain yang menerapkan optimasi metode SVR dengan *Ant Colony Optimization* (ACO) menggunakan *kernel Radial Basis Function* (RBF) (Irwanti, et al., 2016). Pada penelitian tersebut mengoptimalkan parameter SVR kompleksitas, *epsilon* dan parameter *kernel* RBF yaitu sigma dilakukan optimasi. Pada penelitian tersebut membuktikan metode SVR-ACO dapat melakukan peramalan harga Bahan Bakar Minyak (BBM) yang menghasilkan nilai *error* NMSE rendah dengan rentang 0.108 sampai 0.28 (Irwanti, et al., 2016).

Menurut penelitian Hong dan Pai (2007) yang telah dijelaskan sebelumnya menyarankan untuk menggunakan algoritma optmasi lainnya untuk optimasi parameter SVR. Salah satu saran Hong dan Pai (2007) seperti penggunaan algoritma PSO. Terdapat peneltian terkait penggunaan algoritma PSO untuk optimasi metode SVR untuk peramalan curah hujan di Guangxi China (Zhao & Wang, 2010). Data yang digunakan merupakan data curah hujan bulanan di Guangxi China selama tahun 1954–2008. Berdasarkan penelitian Zhao dan Wang (2010) dapat digunakan sebagai metode alternatif untuk peramalan curah hujan dengan akurasi peramalan yang lebih tinggi serta meningkatkan kualitas peramalan. Akurasi berdasarkan perhitungan MAPE untuk metode SVR bernilai 26.46% dan metode SVR-PSO bernilai 3.85% (Zhao & Wang, 2010). Penelitian serupa yang dilakukan oleh Zhao dan Wang (2010) juga dilakukan oleh Vikram dan Veer (2011) untuk melakukan peramalan curah hujan di Andhra Pradesh. Data yang digunakan merupakan data curah hujan bulanan di Andhra Pradesh selama tahun 2006–2010. Berdasarkan penelitian tersebut menyatakan metode SVR-PSO dapat digunakan sebagai metode alternatif untuk peramalan curah hujan untuk mendapatkan akurasi peramalan yang lebih tinggi serta meningkatkan kualitas prediksi (Vikram & Veer, 2011).

Penggunaan metode SVR-PSO tidak hanya untuk peramalan curah hujan. Terdapat penelitian lain menggunakan metode SVR-PSO untuk memprediksi kecepatan angin (Salcedo-Sanz, et al., 2011). Dengan menggunakan data kecepatan angin mulai bulan januari sampai bulan juni 2006 pada turbin angin yang diperoleh dari peternakan angin di La Fuensanta, Albacete, Spanyol. Pada penelitian Salcedo-Sanz, et al. (2011) selain menggunakan metode SVR-PSO juga menggunakan SVR dengan optimasi *Evolutionary Programming* (EP) sebagai komparasi. Berdasarkan penerapan metode tersebut, didapatkan akurasi perhitungan berdasarkan *Mean Absolute Error* (MAE) di global model GFS pada

Turbine 15. Metode SVR-PSO menunjukkan kinerja yang sangat baik untuk prediksi kecepatan angin yang bernilai 1.782335, sedangkan SVR-EP bernilai 1.792161 (Salcedo-Sanz, et al., 2011).

Begitu juga penelitian yang dilakukan Novitasari, et al. (2016) dengan mengkombinasikan metode *Simulated Annealing* (SA) dengan metode PSO dengan kasus yang sama. Pada metode SAPSO-SVR memiliki kecepatan untuk mendapatkan konvergensi dan memiliki waktu komputasi yang tinggi dibandingkan metode PSO-SVR. Metode SAPSO-SVR memiliki nilai *cost* 0.5575 dengan waktu eksekusi 50238 ms, sedangkan metode PSO-SVR memiliki nilai *cost* 0.5981 dengan waktu eksekusi 83756 ms (Novitasari, et al., 2016).

Berdasarkan penelitian sebelumnya, algoritma PSO sering digunakan untuk kasus optimasi, sehingga terus dikembangkan maupun dimodifikasi (Rini, et al., 2011). Terdapat penelitian lain yang melakukan variasi metode PSO untuk optimasi metode SVR. Salah satunya penelitian yang dilakukan Adhani, et al. (2014) dengan menggunakan metode gabungan *Genetic Algorithm* dengan PSO (GAPSO) untuk optimasi parameter SVR untuk peramalan curah hujan. Dengan menggunakan data data Indian Ocean Dipole (IOD), NINO3.4, *Sea Surface Temperature Anomaly* (SSTA) dan Hasil observasi curah hujan dari 15 stasiun di Indramayu periode tahun 1979-2008. Penerapan Metode SVR menggunakan *kernel* RBF yang menghasilkan metode gabungan SVR-GAPSO memiliki koefisien korelasi tertinggi 0,87 dan RMSE nilai kesalahan dari 11.53 di stasiun Bulak. Sementara itu, nilai kesalahan peramalan di stasiun Cikedung memiliki nilai *error* NMRSE terendah 0.78 dan koefisien korelasi 9.01. Hal tersebut disebabkan oleh bentuk fungsi yang tidak sesuai dengan data, atau berbagai parameter yang salah dikumpulkan saat optimasi terjadi (Adhani, et al., 2014). Sementara itu, pada penelitian Soebroto, et al. (2015) menggunakan metode TVIWPSO untuk optimasi parameter SVR dalam prediksi tinggi muka air (TMA). Berdasarkan penelitian Soebroto, et al. (2015) hasil evaluasi terbaik diperoleh pada data bulan juni 2007 memiliki nilai kesalahan MAE terkecil bernilai 0.00755.

Terdapat variasi algoritma PSO untuk optimasi parameter metode SVM pada kasus klasifikasi (Chen, et al., 2014). Pada penelitian yang dilakukan oleh ling Chen, et al. (2014) menggunakan metode SVM klasifikasi dengan optimasi *Parallel Time Variant Particle Swarm Optimization* (PTVPSO), PSO dan *grid search*. Dengan menggunakan sebanyak 30 data set dari database *UCI Machine Learning* dan *StatLog* dikomparasi pada permasalahan klasifikasi. Berdasarkan penelitian tersebut membuktikan metode PTVPSO-SVM dengan akurasi 98,64% dapat meningkatkan akurasi prediksi yang tinggi serta mengurangi waktu komputasi. Dibandingkan dengan metode PSO-SVM dengan akurasi 97,29%, sedangkan metode *Grid Search* (GRID-SVM) 96,42%. Oleh sebab itu, metode PTVPSO-SVM dapat melakukan pemilihan fitur dan parameter yang optimal pada SVM (Chen, et al., 2014).

Berdasarkan penelitian yang telah dijelaskan sebelumnya optimasi parameter metode SVM baik kasus regresi maupun klasifikasi dapat mempengaruhi akurasi

metode SVM dalam menyelesaikan masalah. Terdapat penelitian terkait memparalelkan metode PSO atau *Parallel Particle Swarm Optimization* (PPSO) untuk optimasi parameter SVR untuk meramalkan arus penumpang kereta api di China (Xun W., et al., 2015). Dengan menggunakan data arus penumpang kereta api di China selama periode tahun 1987 sampai 2010. Berdasarkan penelitian tersebut metode SVR-PPSO dapat meningkatkan akurasi prediksi dibandingkan dengan metode SVR-PSO, serta dapat meningkatkan eksekusi waktu komputasi lebih cepat. Untuk metode PPSO tercatat kurang lebih 50 menit pada node ke 8, sedangkan metode PSO tercatat PSO = kurang lebih 1 jam (Xun W., et al., 2015).

Menurut Xun W., et al. (2015) dalam penggunaan metode PPSO dapat mengantisipasi metode PSO terjebak pada nilai lokal optimal yang berlebihan serta meningkatkan waktu eksekusi proses optimasi. Pada penelitian terkait lainnya metode PPSO sering diterapkan untuk permasalahan optimasi, seperti optimasi pencarian referensi stasiun distribusi pada sektor penerbangan (Shao, et al., 2011). Pada proses optimasi tersebut data diperoleh dari Stasiun referensi yang mengumpulkan pengukuran dari *Global Positioning System* (GPS) dan SBAS satelit untuk menentukan *User Differential Range Error* (UDRE) serta data koreksi. Berdasarkan hasil penelitian tersebut membuktikan metode PPSO dapat menghindari terjebak pada lokal optimasi. PPSO dapat membantu menemukan referensi yang baik stasiun distribusi di SBAS, dan meningkatkan akurasi UDRE, meningkatkan ketersediaan SBAS dan meningkatkan integritas serta keamanan penerbangan. Akurasi didapatkan berdasarkan nilai *fitness* untuk metode PPSO bernilai 337.53, PSO (Local optimization) bernilai 694.71 dan PSO bernilai 347.80 (Shao, et al., 2011).

Penelitian terkait penggunaan metode PPSO lainnya terdapat optimasi pengurangan atribut menggunakan *Dataset Car Test Result* (CRT) dan 5 *dataset* dengan atribut diskrit dari UCI *repository* (Xu & Zhang, 2007). Pada penelitian tersebut membuktikan metode PPSO memiliki kinerja yang lebih baik dengan dataset yang lebih besar. Dengan waktu komputasi waktu metode PPSO bernilai 8.253 detik yang jauh lebih cepat dibandingkan PSO bernilai 17.150 detik. Dengan menggunakan data CTR hasil eksekusi untuk menemukan solusi memiliki waktu selisih yang berbeda (Xu & Zhang, 2007).

Menurut penelitian sebelumnya penggunaan metode SVR dapat digunakan untuk peramalan curah hujan dengan baik. Begitu juga, dengan adanya optimasi parameter SVR dengan metode optimasi *parallel* PSO (PPSO) yang dapat mengoptimalkan parameter SVR sehingga dapat memaksimalkan kinerja metode SVR untuk melakukan peramalan. Berdasarkan penelitian terkait, penelitian ini akan menerapkan metode SVR dengan optimasi PPSO untuk meramalkan curah hujan di beberapa wilayah kecamatan kabupaten Malang. Peramalan curah hujan dilakukan berdasarkan data runut waktu curah hujan pada rentang waktu tahun 2000 sampai 2015. Data curah hujan dengan satuan milimeter per dasarian (10 hari).

Berikut merupakan penelitian-penelitian terkait yang disajikan dalam bentuk tabel agar mudah dipahami ditunjukkan pada Tabel 2.1.

Tabel 2.1 Perbandingan penelitian penulis dengan penelitian terkait sebelumnya

No.	Peneliti	Objek	Metode	Hasil
1.	Wei (2012)	Analisis data dari 73 daerah tropis topan yang mempengaruhi daerah aliran sungai pada tahun 1990-2007 di Shimen Reservoir, Taiwan. Analisis karakteristik iklim, trek topan dan jumlah curah hujan per jam di SHimen Reservoir, Taiwan.	<i>Mesoscale Model</i> (MM5), <i>Gaussian Kernel SVMs</i> (GSVMs), <i>Wavelet Kernel SVMs</i> (WSVMs) dan <i>linear regressions</i> (regression). Proses: 1. Pengumpulan, <i>preprocessing</i> data dan statistik. 2. Temukan fitur dan kasus dirancang. 3. Pembangunan model.	Metode WSVM memiliki rata BIAS dan nilai-nilai ETS dekat dengan 1,0 memberikan prediksi yang lebih baik daripada Metode GSVM dan model regresi. Serta membandingkan model MM5 untuk memvalidasi numerik pada Topan Zeb (1998) dan Nari (2001). Berdasarkan analisis bahwa 5 model menghasilkan peramalan curah hujan yang kumulatif, sedangkan model statistika cenderung terlalu rendah. Akurasi berdasarkan ilai evaluasi RMSE. Akurasi: • GSVM = 0.288 • WSVM = 0.257 • Regression = 0.296

2.	Diani, et al. (2013)	Data curah hujan di Kabupaten Malang periode tahun 1996 sampai Pebruari 2012 yang diperoleh dari Badan Meteorologi Klimatologi dan Geofisika (BMKG) Malang. Lokasi ketinggian curah hujan di Tumpang, Pujon, dan Batu.	<i>Vector Autoregressive</i> (VAR), <i>Generalized Autoregressive</i> (GSTAR), <i>Vector Autoregressive Neural Network</i> (VAR-NN) dan <i>Generalized Autoregressive Neural Network</i> (GSTAR-NN). Dengan melakukan pemodelan jaringan syaraf tiruan berdasarkan metode GSTAR dan VAR.	Metode VAR-NN dapat meramalkan hasil terbaik dibandingkan metode lainnya dengan menggunakan 1 neuron dalam hidden layer. Akurasi berdasarkan nilai evaluasi RMSE pada peramalan curah hujan di Batu, Malang. Akurasi: • VAR-NN = 4.590 • GSTAR-NN = 4.821 • VARIMA = 4.719 • GSTAR = 4.840
3.	Cahyani, et al. (2014)	Data curah hujan harian sebanyak 304 data pada periode bulan januari 2011 sampai januari 2012 diperoleh dari Badan Meteorologi Klimatologi dan Geofisika (BMKG) Malang. Data curah hujan yang terbaik terdapat pada	Algoritma <i>Al-Alaoui Backpropagation</i> Proses: 1. Inisialisasi bobot dengan bilangan acak nilai kecil. 2. Cek kondisi terpenuhi atau belum.	Dengan menggunakan Algoritma <i>Al-Alaoui Backpropagation</i> terdapat nilai MSE 0.00313611 dan memiliki akurasi 100% terhadap pengujian pada bulan juli saat musim kemarau.

		periode tahun januari 2011 sampai juni 2012.	3. Tiap pasang training dilakukan langka 3 sampai 8. 4. Tiap masukan menerima sinyal input x_i dan diteruskan ke seluruh hidden layer. 5. Semua keluaran tersembunyi dihitung. 6. Semua sinyal input yang sudah terbobot dijumlahkan. 7. Hitung faktor δ unit keluaran. 8. Hitung factor δ layer tersembunyi. 9. Hitung perubahan bobot garis yang akan menjadi keluaran.	
4.	Trafalis, et al. (2005)	Data curah hujan dari Chandler, Oklahoma Mesonet Stasiun pengukur	Metode <i>Artificial Neural Networks</i> (ANNs), standard <i>Support Vector Regression</i> (SVR), <i>Least-Squares Support Vector Regression</i> (LS-SVR),	Metode SVR memiliki kinerja yang baik dibandingkan teknik lainnya untuk deteksi curah hujan. Akurasi berdasarkan rerata proses

		hujan untuk tahun kalender 1998.	<i>Linear Regression</i> (LR) dan <i>Rain Rate</i> (RR) formula. Proses: 1. Inisialisasi data dan parameter. 2. Proses <i>sequential learning</i>. 3. Proses pengujian model regresi. 4. Proses perhitungan <i>error</i>. 5. <i>Error rate</i>.	<i>testing Mean Square Error</i> (MSE). Akurasi: • ANNs = 0.3383 • LR = 0.4665 • SVR = 0.3099 • LS-SVR = 0.2709 • RR = 0.3728
5.	Hong dan Pai (2007)	Data curah hujan pada daerah Aliran Sungai di Wu-Tu, Taiwan utara. Data curah hujan diperoleh Volume per jam (dari Agustus 1985 sampai Agustus 1997) dari tiga alat pengukur hujan.	<i>Support Vector Regression</i> (SVR) dengan optimasi parameter algoritma <i>Simulated Annealing</i> (SA). Proses: 1. Inisialisasi suhu dan kandidat untuk optimas parameter SVR. 2. Membuat langkah acak untuk mendapatkan sistem.	Berdasarkan optimasi parameter metode <i>Holt-Winters</i> (HW), <i>Exponential Smoothing</i> (SHW) dan <i>Recurrent Neural Network</i> (RNN), parameter model SVRSA memiliki nilai validasi <i>Normalized Mean Squared Error</i> (NMSE) rendah untuk peramalan curah hujan selama musim topan yang tepat. Selain itu, dapat dikembangkan dengan

			3. cek pengujian awal. 4. cek kesemimbangan. 5. set suhu reduksi. 6. Pendinginan.	menggunakan teknik optimasi lainnya, seperti <i>Particle Swarm Optimization</i>(PSO). Akurasi: SVRSA model • $\sigma = 0.05206$, • $C = 28.867$ • $\epsilon = 0.87811$ • nilai <i>error</i> NMSE = 0.3963
6.	Pai dan Hong (2007)	Data curah hujan selama musim topan pada daerah aliran sungai Wu-Tu di Taiwan utara.	<i>Support Vector Regression</i> (SVRG) dan recurrent SVR (RSVRG) dioptimasi menggunakan algoritma genetika, serta dikomparasi dengan metode <i>General Regression Neural Networks</i> (GRNN). Proses: 1. Inisialisasi data <i>training</i>, <i>testing</i> dan validasi. 2. Percobaan peramalan menggunakan SVR.	Penggunaan metode RSVRG mencapai hasil peramalan yang lebih akurat dibandingkan dengan SVRG dan GRNN model. Akurasi berdasarkan perhitungan <i>Mean Absolute Percentage Error</i> (MAPE). Akurasi: • RSVRG = 1.381 • SVRG = 1.559 • GRNN = 1.654

			3. Kalkulasi <i>training error</i>. 4. Seleksi SVR parameter dengan algoritma genetika. 5. Seleksi SVR model dengan tingkat validasi <i>error</i> terkecil. 6. Testing nilai <i>error</i>.	
7.	Zhao dan Wang (2010)	Data curah hujan bulanan di Guangxi China selama tahun 1954–2008.	<i>Support Vector Regression</i> (SVR) dengan optimasi parameter menggunakan algoritma <i>Particle Swarm Optimization</i> (PSO). Proses: 1. Inisialisasi data berupa data <i>testing</i>, data validasi dan data <i>training</i>. 2. Inisialisasi populasi, posisi dan kecepatan. 3. <i>Training</i> parameter SVR secara <i>partial</i>. 4. Kalkulasi <i>fitness</i> terbaik.	Metode SVR-PSO dapat digunakan sebagai metode alternatif untuk peramalan curah hujan dengan akurasi peramalan yang lebih tinggi serta meningkatkan kualitas prediksi. Akurasi berdasarkan perhitungan <i>Mean Absolute Percentage Error</i> (MAPE). Akurasi: • SVR 26.46% • SVR-PSO 3.85%

			5. Perbarui kecepatan. 6. Perbarui posisi. 7. Cek kondisi apakah sudah memenuhi kriteria. 8. <i>Training</i> parameter SVR secara <i>fully</i> . 9. <i>Testing</i> data menggunakan parameter SVR yang telah dikombinasikan.	
8.	Vikram dan Veer (2011)	Data curah hujan bulanan di Andhra Pradesh selama tahun 2006–2010.	<i>Support Vector Regression</i> (SVR) dengan optimasi parameter menggunakan algoritma <i>Particle Swarm Optimization</i> (PSO). Proses: 1. Inisialisasi data dan parameter SVR serta optimasi PSO. 2. Perhitungan kecepatan. 3. Pembaruan posisi. 4. Proses perhitungan <i>error</i>.	Metode SVR-PSO dapat digunakan sebagai metode alternatif untuk peramalan curah hujan dengan akurasi peramalan yang lebih tinggi serta meningkatkan kualitas prediksi.

			5. Parameter optimal.	
9.	Salcedo-Sanz, et al. (2011)	Data kecepatan angin dari bulan januari sampai bulan juni 2006 pada turbin angin dari peternakan angin di La Fuensanta, Albacete, Spanyol.	<i>Support Vector Regression</i> (SVR) dengan optimasi <i>Evolutionary Programming</i> (EP) dan <i>Particle Swarm Optimization</i> (PSO). Proses: <i>Evolutionary programming</i> 1. Inialisasi populasi tiap individu. 2. Evaluasi nilai <i>fitness</i> tiap individu. 3. Tiap parent melakukan perkawinan. 4. Kalkulasi nilai <i>fitness</i> anak berdasarkan tiap hasil persilangan. 5. Melakukan komparasi dari <i>parent</i> dan anaknya.	Dari kedua metode SVR-EP dan SVR-PSO menunjukkan kinerja yang sangat baik untuk memprediksi kecepatan angin dibandingkan model penelitian sebelumnya <i>Multi-Layer Perceptron</i> (MLP). Akurasi berdasarkan perhitungan <i>Mean Absolute Error</i> (MAE) di global model GFS pada Turbine 15. Akurasi: • EP-SVMr = 1.792161 • PSO-SVMr = 1.782335

			6. Pemilihan individu terbaik untuk generasi selanjutnya. 7. Berhenti apabila memenuhi kriteria, jika belum kembali ke langkah 3. <i>Particle Swarm Optimization</i> 1. Inisialisasi data dan parameter SVR serta PSO. 2. Perhitungan kecepatan. 3. Pembaruan posisi. 4. Proses perhitungan <i>error</i> . 5. Parameter optimal.	
10.	Adhani, et al. (2013)	Data curah hujan selama musim kering di Indramayu menggunakan <i>Southern Oscillation Index</i> (SOI), NINO 3.4 SST dan data aktual pada jangka tahun 1970-2010. Data SOI and NINO SST diperoleh dari	<i>Support Vector Regression</i> (SVR) dengan optimasi parameter algoritma <i>grid search</i> . Serta pengujian <i>kernel Radial Basis Function</i> (RBF), <i>Polynomial</i> dan <i>Linear</i> . Proses:	<i>Support Vector Regresi</i> (SVR) dengan optimasi algoritma <i>grid search</i> dan menggunakan <i>kernel Radial Basis Function</i> (RBF) untuk meramalkan curah hujan pada musim kemarau dengan korelasi tertinggi nilai koefisien, dan

		Australian Bureau of Meteorology (BOM), sedangkan data observasi curah hujan diperoleh Badan Meteorologi Klimatologi dan Geofisika (BMKG) stasiun cuaca di kabupaten Indramayu.	1. Identifikasi masalah dan formulasi. 2. Pengambilan data. 3. <i>Training</i> data. 4. Clustering terhadap <i>kernel linear</i>, <i>polynomial</i> dan RBF. 5. Optimasi parameter menggunakan <i>grid search</i>. 6. <i>Testing</i> data. 7. <i>Clustering</i>. 8. Peramalan. 9. Analisis dan evaluasi.	nilai terendah NRMSE menggunakan SOI dan data Nino 3.4 SST. Dibandingkan dengan <i>kernel polynomial</i> memiliki kinerja dengan koefisien korelasi rendah dan nilai kesalahan <i>Normalized Root Mean Square Error</i> (NRMSE) yang tinggi. Akurasi: • RBF = 1.73 • Linear = 4.37 • Polynomial = 357.54
11.	Alwee, et al. (2013)	Data tingkat kejahatan: - Data tahunan tingkat kejahatan - Indeks harga konsumen (pakaian) - Produk domestik bruto di Amerika Serikat	Metode gabungan <i>Support Vector Regression</i> (SVR) dan <i>Autoregressive Integrated Moving Average</i> (ARIMA) dengan optimasi parameter menggunakan algoritma <i>Particle Swarm Optimization</i> (PSO).	Metode gabungan model PSOSVR-PSOARIMA, dapat menghasilkan kesalahan kecil dibandingkan dengan model individu dan model hibrida PSOSVR-ARIMA. Model hibrida diusulkan dapat diterapkan untuk peramalan tingkat kejahatan. Akurasi

		- Tingkat pengangguran (umur 20 - 24 tahun) pada tahun 1960-2009	Proses: 1. Pembagian data training dan data testing. 2. Inisialisasi posisi dan kecepatan particle. 3. Training SVR model <i>k-fold cross-validation</i>. 4. Evaluasi nilai fitness tiap partikel. 5. Perbarui <i>pbest</i> dan <i>gbest</i>. 6. Kalkulasi nilai peramalan SVR. 7. Kalkulasi SVR model. 8. Identifikasi model ARIMA. 9. inisialisasi posisi dan kecepatan partikel secara acak. 10. Evaluasi nilai fitness tiap partikel. 11. Perbarui <i>pbest</i> dan <i>gbest</i>.	berdasarkan perhitungan <i>Root Mean Square Error</i> (RMSE). Akurasi: • ARIMA = 72.371 • PSOARIMA = 72.125 • PSOSVR = 12.332 • PSOSVR ARIMA = 11.704 • PSOSVR PSOARIMA = 10.973
--	--	---	--	--

			12. kalkulasi nilai peramalan ARIMA. 13. Integrasi nilai peramalan SVR ARIMA.	
12.	Yao, et al. (2014)	- Data jalan bebas hambatan (tol) Shenda dari kota Shenyang ke Dalian dan jalan bebas hambatan (tol) utama di Cina daratan dan Total panjang adalah 400 km. - Data jalan bebas hambatan (tol) Heda di provinsi Liaoning. dari Jixi ke Mudanjiang dan total panjang 1.390 km	<i>Support Vector Regression</i> (SVR) dengan optimasi parameter menggunakan algoritma <i>tabu search</i>. Proses: 1. <i>Input</i> data untuk diidentifikasi SVM. 2. Normalisasi data. 3. Inisialisasi solusi secara acak untuk optimasi parameter ($C, \epsilon, dan \sigma$). 4. Evaluasi solusi berdasarkan tetangga terdekat dengan jumlah minimal. 5. Komputasi dengan SVM yang telah dioptimasi.	Algoritma <i>tabu search</i> efektif dapat memberikan nilai parameter terbaik untuk SVM, sehingga mengungguli dari <i>Artificial Neural Networks</i> (ANNs). SVM adalah alat yang ampuh untuk mendeteksi insiden di jalan bebas hambatan (tol). Akurasi berdasarkan perhitungan <i>Mean Time-To-Detect</i> (MTTD). Akurasi: • SVM = 3.52 • ANN = 4.63

13.	Sanusi, et al. (2014)	- Data GCM (digunakan sebagai memperjelas variabel) diperoleh dari <i>Climate Information Tool Kit (CLIK) APEC Climate Center (APCC)</i> untuk membangun model prediksi pada 3 bulan yang berbeda: Mei, Juni, dan Juli (MJJ) dari tahun 1982-2008 (27 tahun). - Data curah hujan (digunakan sebagai variabel respon) diperoleh dari nilai rata-rata musiman curah hujan setiap stasiun curah hujan di Indramayu pada 15 stasiun pengamatan. Data pengamatan curah hujan digunakan 3 bulan: Mei, Juni, Juli (MJJ) dari tahun 1982-2008 (27 tahun) di setiap stasiun hujan.	<i>Support Vector Regression (SVR)</i> dengan menggunakan <i>kernel Radial Basis Function (RBF)</i> dan <i>Linear</i>. 1. Pengumpulan data curah hujan. 2. Distribusi data training dan testing dari data GCM dan data aktual. 3. Menggunakan <i>9-fold cross validation</i>. 4. Analisis data menggunakan <i>Prescription Cost Analysis (PCA)</i>. 5. <i>SVR training</i> dan <i>testing</i>. 6. Hasil peramalan.	Model GCM dapat digunakan untuk prediksi curah hujan menggunakan metode SVR. Berdasarkan korelasi dan nilai <i>error RMSE</i> pada fungsi <i>kernel</i> linier lebih baik daripada menggunakan fungsi <i>kernel RBF</i> . Akurasi masih belum baik dan perlu adanya optimasi parameter pada fungsi <i>kernel</i> dengan teknik optimasi lain. terdapat korelasi terbaik pada peramalan curah hujan di Cikedung dengan nilai korelasi 0.744 dan nilai <i>error RMSE</i> 23.937. <i>Kernel</i> Linier menghasilkan prediksi yang baik daripada <i>RBF kernel</i> .
14.	Adhani, et al. (2014)	Data Indian Ocean Dipole (IOD), NINO3.4, Sea Surface	<i>Support Vector Regression (SVR)</i> dengan optimasi	Penerapan metode SVR-GAPSO dengan koefisien

		Temperature Anomaly (SSTA) data dan Hasil observasi curah hujan dari 15 stasiun di Indramayu periode tahun 1979-2008.	parameter kombinasi algoritma <i>Genetic Algorithm</i> dan <i>Particle Swarm Optimization</i> (GAPSO), serta menggunakan <i>kernel Radial Basis Function</i> (RBF). Proses: 1. Identifikasi masalah dan formulasi. 2. Pengambilan data. 3. Training data. 4. Clustering terhadap <i>kernel linear</i>. polynomial dan RBF. 5. Optimasi parameter menggunakan GAPSO. 6. <i>testing</i> data. 7. <i>clustering</i>. 8. Peramalan. 9. Analisis dan evaluasi	korelasi tertinggi 0,87 dan RMSE nilai kesalahan dari 11.53 di stasiun Bulak. Stasiun Cikedung memiliki nilai <i>error</i> NMRSE terendah 0,78 dan koefisien korelasi 9,01. Hal tersebut disebabkan oleh bentuk fungsi yang tidak sesuai dengan data, atau berbagai parameter yang salah dikumpulkan saat optimasi terjadi.
15.	Chen, et al. (2014)	Sebanyak 30 data set dari database UCI <i>Machine</i>	<i>Support Vector Machine</i> (SVM) dengan optimasi Parallel Time	Dengan menggunakan metode PTVPSO-SVM dapat

		<i>Learning</i> dan <i>StatLog</i> dikomparasi pada permasalahan klasifikasi.	Variant <i>Particle Swarm Optimization</i> (PTVPSO). Proses: 1. <i>Encode</i> partikel dengan 2 dimensi. 2. inisialisasi individu pada populasi dengan bilangan acak. 3. Training SVM model dengan fitur seleksi dan optimasi parameter menggunakan PSO. 4. Menambah iterasi pada optimasi PSO. 5. Menambah jumlah populasi serta perbarui posisi dan kecepatan partikel. 6. Melakukan pencarian solusi secara <i>continue</i>. 7. Training SVM model dengan mengkalkulasi	meningkatkan akurasi prediksi yang tinggi serta mengurangi waktu komputasi. Dibandingkan dengan PSO-SVM dan <i>grid search</i> (GRID-SVM) metode PTVPSO-SVM dapat melakukan pemilihan fitur dan parameter yang optimal pada SVM. Berdasarkan studi kasus data Wisconsin PTV. Akurasi: • PTVPSO-SVM = 98.64% • PSO-SVM = 97.29% • GRID-SVM = 96.42%.
--	--	--	--	---

			<i>fitness</i> terbaik berdasarkan partikel. 8. Perbarui <i>fitness</i> optimal dan personal optimal. 9. Perbarui global optimal <i>fitness</i> dan global optimal partikel. 10. Mendapatkan parameter optimal dan fitur terbaik berdasarkan partikel terbaik (global optimal).	
16.	Prahotama dan Yasin (2015)	Data tahunan curah hujan di Semarang selama 2009-2013	Support Vector Regression (SVR) dan <i>Quadratic Loss Function</i> dengan menggunakan <i>kernel linear</i>, <i>polynomial</i>, dan <i>Radial Basis Function</i> (RBF). Proses: 1. Inisialisasi variable prediksi. 2. Menggunakan fungsi <i>kernel</i>.	Berdasarkan data curah hujan mingguan di kota Semarang menunjukkan tren yang fluktuatif, non stasioner. Namun, SVR menggunakan <i>kernel polynomial</i> sebesar 71.61% mampu meramalkan curah hujan mingguan dengan tepat.

			3. Mencari parameter yang terbaik dengan menggunakan <i>kernel</i>. 4. Menghitung berat <i>vector</i> dan bias. 5. Menghitung data <i>training</i> dan data <i>testing</i>. 6. Menghitung <i>R-Square</i> data <i>training</i> dan data <i>testing</i>. 7. Prediksi curah hujan dengan data <i>training</i> dibagi data <i>testing</i>.	
17.	Xun W., et al. (2015)	Data arus penumpang kereta api di China selama tahun 1987-2010.	<i>Support Vector Regression</i> (SVR) dengan optimasi parameter menggunakan algoritma <i>Parallel Particle Swarm Optimization</i> (PPSO). Proses: 1. Pengumpulan data. 2. Inisialisasi hadoop <i>platform</i>.	Metode PPSO-SVR secara efektif dapat meningkatkan akurasi prediksi dibandingkan dengan metode PSO-SVR dan SVR basik, sehingga tingkat eksekusi waktu komputasi lebih cepat. Akurasi:

			3. Inisialisasi konfigurasi PPSO untuk optimasi parameter SVM.	• PPSO = kurang lebih 50 menit pada node ke 8. • PSO = kurang lebih 1 jam.
18.	Soebroto, et al. (2015)	Data Tinggi Muka Air (TMA) dimulai tahun 2006 hingga 2010 pada bulan Juni 2006, Desember 2006, Juni 2007, Desember 2007, Januari 2008, Juli 2008, Januari 2009, Juli 2009, Januari 2010 dan bulan Februari 2010. Dari data tersebut, selanjutnya dilakukan normalisasi.	<i>Support Vector Regression (SVR) dan Time Variant Inertia Weight Particle Swarm optimization (TVIWPSO) dengan kernel Radial Basis Function (RBF).</i> Proses: 1. Inisialisasi parameter SVR pada 5 populasi dan setiap populasi berisi 8 partikel. 2. Menaikkan jumlah iterasi TVIWPSO. 3. Perhitungan <i>kernel</i> RBF <i>timevalue</i>. 4. Perhitungan <i>Kernel</i> <i>Matrisk</i> <i>Rij</i>. 5. Menaikkan jumlah iterasi SVR.	Metode SVR yang dioptimasi menggunakan metode TVIWPSO dapat melakukan prediksi Tinggi Muka Air (TMA) dengan baik. Hasil evaluasi terbaik diperoleh pada data bulan juni 2007 dengan nilai kesalahan MAE terkecil bernilai 0.00755. Namun, dalam proses iterasi pada tahap pelatihan dan pengujian terbaik pada iterasi ke 100000 dengan MAE bernilai 0.142659943.

			6. Perhitungan E_i. 7. Perhitungan $F(x)$. 8. Cek kondisi iterasi maksimal SVR (iterasi kecil), jika belum maksimal maka melakukan perulangan kembali. 9. Perhitungan MAE untuk mencari Gbest. 10. Perhitungan <i>Velocity</i>. 11. Perhitungan posisi. 12. Seleksi Fitness Value. 13. Cek kondisi iterasi maksimal PSO, jika belum maksimal maka melakukan perulangan kembali. 14. Ambil hasil partikel dengan parameter SVR terbaik.	
--	--	--	--	--

			15. Menaikkan jumlah iterasi SVR. 16. Perhitungan E_i. 17. Perhitungan $F(x)$. 18. Cek kondisi iterasi maksimal SVR (iterasi besar), jika belum maksimal maka melakukan perulangan kembali. 19. Hasil peramalan TMA.	
19.	Novitasari, et al. (2016)	Data yang digunakan merupakan dataset proyek perangkat lunak Desharnais pada tahun 1983-1987 sebanyak 77 data proyek. Dengan 1 variabel dependen <i>Effort</i> dan 7 variabel independen yaitu <i>TeamExp</i>, <i>ManagerExp</i>, <i>Transactions</i>, <i>Entities</i>, <i>PointsAdjust</i>, <i>Envergure</i>, dan <i>PointsNonAdjust</i>.	<i>Particle Swarm Optimization</i> (PSO) dengan kombinasi metode <i>Simulated Annealing</i> (SA) untuk optimasi <i>Support Vector Regression</i> (SVR). Proses: 1. Inisialisasi parameter metode PSO, SA dan data yang akan digunakan. 2. Normalisasi data.	Metode SA dapat mengimprovisasi metode PSO serta dapat mengoptimalkan parameter SVR dengan baik. Pada hasil komparasi metode SAPSO-SVR dengan metode PSO-SVR. Metode SAPSO-SVR lebih cepat mendapatkan konvergensi dan memiliki waktu komputasi yang tinggi dibandingkan metode PSO-SVR. Akurasi:

			14. Hasil optimasi parameter SVR dan fitur masukan.	
20.	Xu dan Zhang (2007)	Pengurangan atribut menggunakan <i>dataset Car Test Result (CRT)</i> dan 5 <i>dataset</i> dengan atribut diskrit dari UCI <i>repository</i> .	<i>Parallel Particle Swarm Optimization (PPSO)</i> Proses: 1. Hitung inti relatif kondisi atribut relatif terhadap atribut keputusan. 2. Mengatur jumlah partikel dan maksimum iterasi <i>itermax</i>, membagi seluruh populasi dalam beberapa sub-populasi, dan mendistribusikan prosesor <i>slave</i> untuk setiap partikel 3. prosesor <i>Guru</i> mengirimkan pesan mulai setiap budak prosesor, menginisialisasi partikel dengan Informasi 4. Setiap prosesor <i>slave</i> mengeksekusi PPSO untuk	PPSO untuk atribut pengurangan memiliki kinerja yang lebih baik dengan dataset yang lebih besar. Dengan waktu komputasi waktu jauh lebih sedikit cepat dibandingkan PSO. Dengan menggunakan data CTR hasil eksekusi untuk menemukan solusi memiliki waktu selisih yang berbeda. Akurasi: • PSO = 17.150 detik • Paralel PSO = 8.253 detik.

			atribut pengurangan secara independen. 5. Jika jumlah iterasi mampu menjadi dibagi dengan sepuluh persis, prosesor <i>slave</i> akan mengirimkan mereka partikel pesawat ke prosesor utama. 6. Partikel terburuk diganti dengan yang diterima satu. 7. Jika iterasi selesai maka berakhir dan ambil partikel terbaik global, sebaliknya lanjutkan ke langkah 4.	
21.	Shao, et al. (2011)	Optimasi pencarian referensi stasiun distribusi pada sektor penerbangan. Stasiun referensi mengumpulkan pengukuran dari <i>Global Positioning System</i> (GPS) dan SBAS satelit untuk menentukan	<i>Parallel Particle Swarm Optimization</i> (PPSO) Proses: 1. Buka slot <i>thread</i> 2. Inisialisasi populasi beserta partikel.	Algoritma PPSO dapat menghindari terjebak pada lokal optimasi. PPSO dapat membantu menemukan referensi yang baik stasiun distribusi di SBAS, dan meningkatkan akurasi UDRE, meningkatkan ketersediaan

		<i>user differential range error</i> (UDRE) serta data koreksi.	- inisialisasi kecepatan tiap partikel. - kalkulasi nilai <i>fitness</i> tiap partikel. - pembelajaran <i>local optimal</i>. - pencocokan posisi global <i>optimal</i>. - perbarui kecepatan dan posisi. - pencatatan global <i>optimal</i>.	SBAS dan meningkatkan integritas serta keamanan penerbangan. Akurasi didapatkan berdasarkan nilai <i>fitness</i>. Akurasi: - PPSO = 337.53 - PSO (Local optimization) = 694.71 - PSO = 347.80.
22.	Tsai, et al. (2014)	Optimasi LDI Controller	<i>Parallel Particle Swarm Optimization</i> (PPSO), Fuzzy Inferensi Sistem (Takagi dan Sugeno), jaringan syaraf tiruan dan <i>Lyapunov Stability Criterion</i>. Proses: - Inisialisasi populasi. - Evaluasi nilai <i>fitness</i>.	PPSO melakukan pencarian P matriks umum untuk memenuhi kriteria stabilitas sistem nonlinear. Berdasarkan kriteria yang dihasilkan, <i>fuzzy</i> desain controller, dengan teknik LMI, dapat digunakan untuk menstabilkan sistem <i>fuzzy</i> sesuai yang diusulkan.

			5. Perpindahan partikel berdasarkan posisi dan kecepatan. 6. Komunikasi dan evaluasi antar partikel. 7. Pengakhiran iterasi dan pencatatan posisi partikel terbaik. 8. Parameter SVR terbaik digunakan untuk dilakukan proses pelatihan. 9. Selanjutnya, proses pengujian peramalan curah hujan. 10. Pencatatan nilai evaluasi kinerja peramalan.	
--	--	--	--	--

Sumber: Alwee, et al. (2013), Trafalis, et al. (2005) Zhao dan Wang (2010), Vikram dan Veer (2011) Salcedo-Sanz, et al. (2011), Yao, et al., (2014), Sanusi, et al. (2014), Adhani, et al. (2013), Adhani, et al. (2014), Hong dan Pai (2007), Prahutama dan Yasin (2015), ling Chen, et al. (2014), Shao, et al. (2011), Xu dan Zhang (2007), Tsai, et al. (2014) Diani, et al. (2013), Cahyani, et al. (2014), Wei (2012), Xun W., et al. (2015), Novitasari, et al. (2016), Soebroto, et al. (2015)

2.2 Peramalan

Ramalan merupakan prediksi terhadap berberapa peristiwa atau kejadian di masa depan (Montgomery, et al., 2015). Lain halnya, dengan definisi prediksi menurut kamus besar bahasa Indonesia merupakan ramalan (Tim Pusat Bahasa Depdiknas, 2008). Secara umum bertujuan pendekatan terhadap kenyataan objek yang terjadi sebenarnya, berdasarkan pengetahuan kejadian-kejadian objek yang telah terjadi (Silver, 2012). Pada bidang gempa, prediksi menyatakan definitif yang spesifik kapan dan di mana gempa akan terjadi, sedangkan ramalan menyatakan probabilistik terhadap suatu objek kondisi belum terjadi pada skala waktu yang lebih lama (Silver, 2012).

Peramalan merupakan suatu kegiatan untuk meramalkan atau melakukan ramalan. Istilah peramalan memiliki keterkaitan erat dengan istilah prediksi (Palit & Popovic, 2006). Istilah prediksi dan peramalan sering digunakan beberapa penelitian yang mengartikan berbeda pada kedua istilah tersebut (Silver, 2012). Menurut Palit dan Popovic (2006), beberapa penelitian dapat menentukan nilai masa depan berdasarkan kenyataan data koleksi yang telah diobservasi dengan mengansumsikan istilah prediksi bukan peramalan. Istilah prediksi biasa digunakan pada sistem kontrol, sedangkan peramalan lebih cenderung dan berhubungan dengan permasalahan analisis *time series* (Palit & Popovic, 2006).

Menurut Heizer dan Render (2011), peramalan adalah perpaduan antara seni dan ilmu untuk memprediksi peristiwa masa depan. Peramalan dapat memanfaatkan data historis dan memprediksi ke masa depan dengan menggunakan perhitungan matematika. Peramalan dapat dilakukan secara subjektif bahkan intuitif atau dapat melakukan kombinasi diantara keduanya dengan menggunakan hasil perhitungan matematika disertai keputusan subjektif yang tepat dari peramal.

Peramalan diperlukan, jika terdapat ketidakpastian tentang masa depan yang tidak dapat dikendalikan (Armstrong, 2001). Menurut Abraham dan Ledolter (2009), kejadian masa depan merupakan suatu hal yang tidak pasti dan terkadang membuat hasil peramalan menjadi kurang tepat. Oleh karena itu, tujuan dari peramalan berusaha untuk mengurangi kesalahan dalam meramalkan dan meningkatkan kualitas hasil ramalan.

Peramalan dapat menjadi hal yang penting disegala aspek kehidupan, dalam kasus individual seperti karir atau kesuksesan seseorang dimasa depan (Armstrong, 2001). Selain itu, peramalan menjadi permasalahan penting dalam berbagai bidang seperti bisnis dan industri, pemerintahan, ekonomi, kedokteran, keuangan, politik, ilmu sosial dan ilmu lingkungan (Montgomery, et al., 2015). Dengan adanya peramalan dapat dijadikan sebagai faktor penting dalam suatu perencanaan dan membantu untuk mendukung keputusan. Oleh karena itu, diperlukan pemahaman tahap-tahap peramalan beserta pemilihan metode yang sesuai untuk dapat menghasilkan ramalan akurat.

kabupaten Malang. Metode RMSE juga pernah digunakan untuk menguji aplikasi HyBMG versi 2.0 untuk melakukan peramalan curah hujan (Sonjaya, et al., 2009). Metode RMSE memiliki hasil evaluasi yang baik, sehingga dapat ditafsirkan sebagai standar evaluasi kinerja untuk peramalan (Levenbach, 2015).

2.3 Data Runut Waktu

Data sebagai bahan baku dalam proses pemodelan dan peramalan (Montgomery, et al., 2015). Sebagian besar masalah peramalan mengacu waktu pada data historis objek kasus yang diangkat (Montgomery, et al., 2015). istilah data historis tersebut lebih sering dikenal dengan data runut waktu (*time series*). Data runut waktu merupakan deretan hasil observasi yang dicatat secara berurutan pada jangka waktu tertentu (jam, harian, mingguan, bulanan, dan tahunan) (Box, et al., 2015).

Data runut waktu dapat diasumsikan sebagai X_t , sedangkan t merupakan waktu tercatatnya data. Deretan data runut waktu juga dapat diasumsikan $\{X_1, X_2, \dots, X_n\}$, sedangkan n merupakan bilangan waktu catatan terakhir pada data runut waktu. Apabila akan meramalkan pada waktu selanjutnya, maka dapat diasumsikan X_{t+1} sebagai waktu kedepan sebagai variabel ramalan, sedangkan waktu sebelumnya dapat diasumsikan X_{t-1} sebagai data pada waktu mundur kebelakang (Box, et al., 2015). Pada Tabel 2.2 merupakan contoh data runut waktu yang diperoleh dari oleh Badan Meteorologi Klimatologi dan Geofisika (BMKG) stasiun klimatologi Karangploso Malang.

Tabel 2.2 Data curah hujan bulanan tahun 2000 - 2001 Karangploso Malang

Tahun	2000	2001
Januari	221	270
Februari	217	268
Maret	297	388
April	115	36
Mei	20	143
Juni	0	155
Juli	26	0
Agustus	0	0
September	114	0
Oktober	245	232
November	356	216
Desmber	104	209

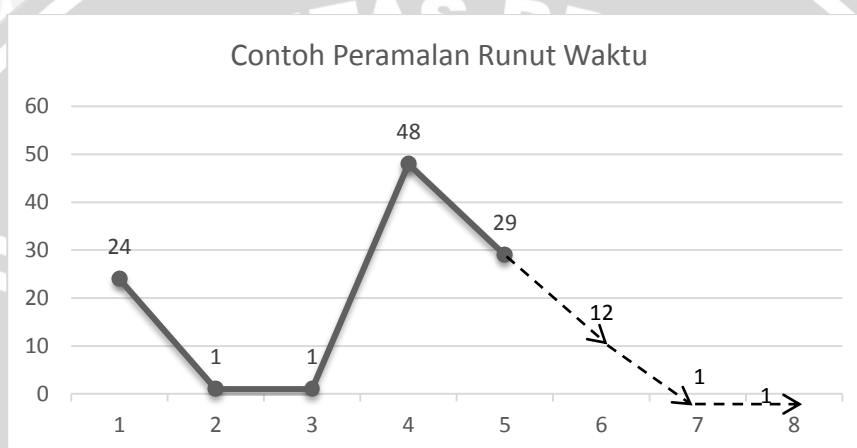
Sumber: BMKG Stasiun Klimatologi Karangploso Malang

Pada Tabel 2.2 merupakan data runut waktu hasil pencatatan fenomena alam terjadinya curah hujan dengan satuan millimeter (MM) yang disajikan dalam periode waktu bulanan selama tahun 2000 sampai dengan tahun 2006. Dengan

adanya data runut waktu dapat mempermudah untuk analisis dan meramalkan kejadian selanjutnya yang akan terjadi kedepannya.

2.3.1 Analisis Data Runut Waktu

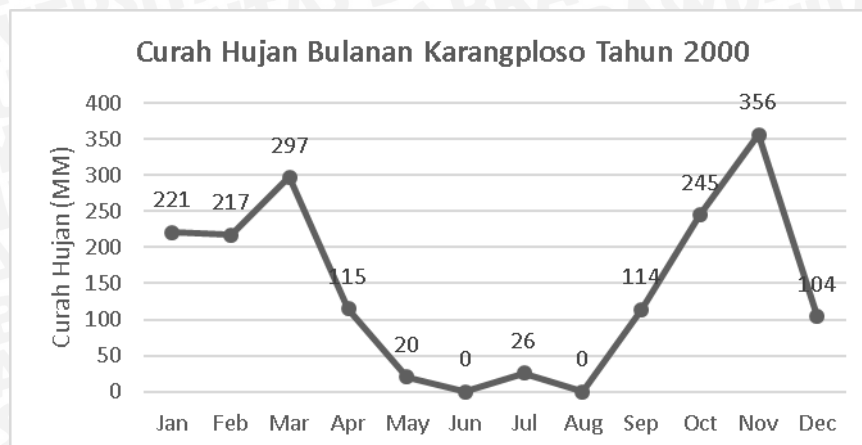
Pada hakekatnya, data runut waktu memiliki ciri khas sebuah pendekatan terhadap hasil observasi yang saling ketergantungan (Box, et al., 2015). Oleh karena itu, diperlukan analisis data runut waktu yang bertujuan untuk menemukan pola dan dapat melakukan ekstrapolasi dengan meneruskan pola selanjutnya ke masa depan (Singh, 2016). Analisis tersebut meliputi identifikasi untuk dapat meningkatkan prediktor dalam memprediksi serta menyesuaikan hubungan variabel pada model yang secara sederhana dapat diasumsikan poin peramalan seperti Gambar 2.5 (Singh, 2016; Montgomery, et al., 2015).



Gambar 2.5 Contoh kasus sederhana peramalan runut waktu

Pada Gambar 2.5, dimisalkan terdapat model data runut waktu hasil observasi dengan garis lurus tidak putus sampai waktu ke lima yang disebut dengan panjang periode waktu (*lag time*) pada angka {24,1,1,48,29} dan dilakukan ekstrapolasi ke nilai selanjutnya pada waktu ke enam pada angka {12} dengan garis putus disertai panah. Ekstrapolasi dapat dilakukan secara keberlanjutan sesuai kebutuhan berdasarkan analisis pola data historis sebelumnya.

Pada pendekatan peramalan, menurut Palit dan Popovic (2006) menggambar (*plotting*) data runut waktu dalam bentuk grafik sangat disarankan pada langkah awal analisis runut waktu. Hal tersebut akan membantu untuk memenuhi proses peramalan. Selain itu dapat menentukan dari sisi mana peramalan akan menjadi yang terbaik. Pada Gambar 2.6 merupakan contoh sebagian gambaran data curah hujan yang direpresentasikan dalam bentuk grafik.



Gambar 2.6 Plot data curah hujan bulanan Karangploso tahun 2000

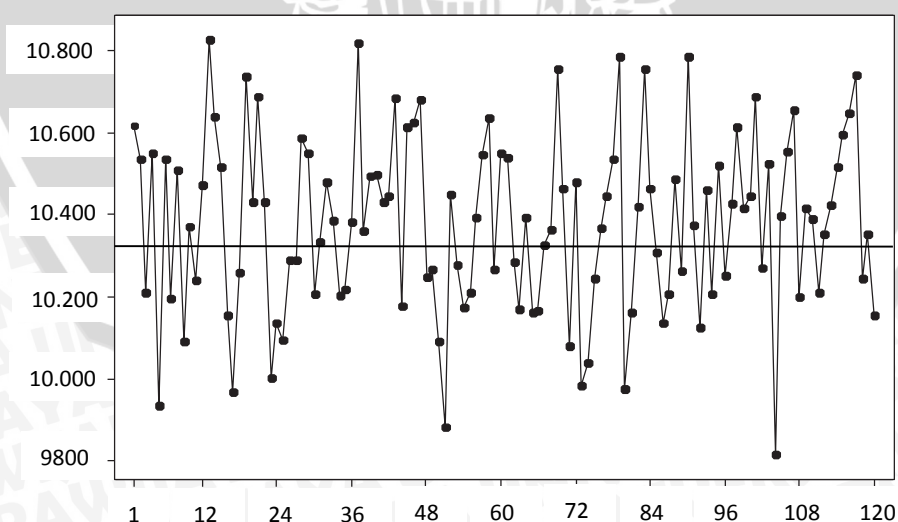
Sumber: BMKG Stasiun Klimatologi Karangploso Malang

Berdasarkan hasil *plotting* dalam bentuk grafik maka dapat diketahui pola data runut waktu yang akan dibahas secara khusus pada sub bab 2.3.2. Selain itu, juga dapat diketahui batas atas-bawah, data *outlier*, dan pergerakan perubahan data secara tajam pada data runut waktu yang dianalisis (Brockwell & Davis, 2002).

2.3.2 Karakteristik Data Runut Waktu

Data runut waktu memiliki karakteristik yang berbeda-beda atau bahkan sama pada kasus tertentu seperti data kasus curah hujan musiman. Berikut merupakan karakteristik pola data runut waktu berdasarkan hasil *plotting* dalam bentuk grafik (Heizer dan Render, 2011; Makridakis, et al., 1998).

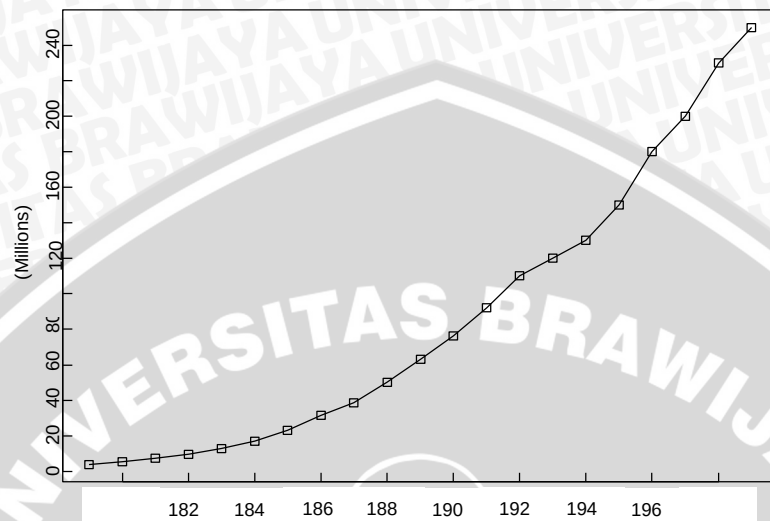
1. *Horizontal* merupakan pola dengan bentuk yang cenderung seimbang secara mendatar. Pada pola ini berfluktuatif sekitar rata-rata yang konstan dan bersifat stabil atau stasioner seperti pada Gambar 2.7 (Makridakis, et al., 1998).



Gambar 2.7 Hasil penjualan produk farmasi selama mingguan

Sumber: Montgomery, et al. (2015)

2. *Trend* merupakan pola yang berkecenderungan. Pola ini memiliki pergerakan ke atas atau ke bawah secara bertahap pada data dari waktu ke waktu (Heizer & Render, 2011). Dengan kata lain terdapat peningkatan atau penurunan kecenderungan tren pada data seperti Gambar 2.8 (Makridakis, et al., 1998).



Gambar 2.8 Populasi penduduk di U.S.A., 1790-1990.

Sumber: Brockwell dan Davis (2002)

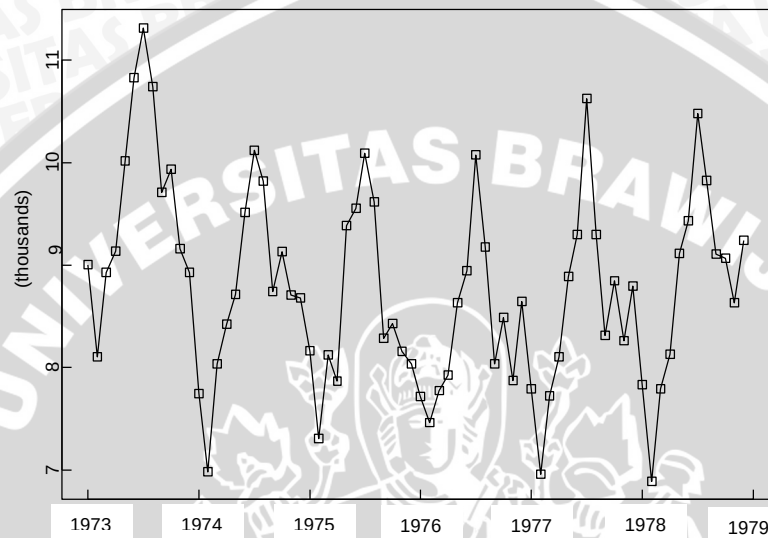
3. *Seasonality* merupakan pola musiman dengan data yang bersifat berulang pada beberapa periode hari, minggu, bulan, atau tahun (Heizer & Render, 2011). Dengan kata lain pola musiman memiliki faktor perubahan secara rutin pada waktu-waktu tertentu dan memiliki beberapa variasi pada kasus tertentu (Makridakis, et al., 1998; Montgomery, et al., 2015). Pola musiman dapat dilihat pada Gambar 2.6 sebelumnya yang terdapat beberapa waktu terjadinya hujan di wilayah Karangploso di tahun 2000. Secara umum periode waktu pola musiman dapat diketahui berdasarkan banyaknya jumlah musim seperti pada Tabel 2.3.

Tabel 2.3 Periode pola musiman

Periode Waktu	Satuan Waktu	Jumlah Musim
Mingguan	Harian	7
Dasarian	Harian	8-11
Bulanan	Harian	28-31
Bulanan	Dasarian	3
Bulanan	Mingguan	4-4 $\frac{1}{2}$
Tahunan	Dasarian	36
Tahunan	Pertiga Bulan	4
Tahunan	Bulanan	12
Tahunan	Mingguan	52

Sumber: Heizer dan Render (2011), dan BMKG Stasiun Klimatologi Karangploso Malang

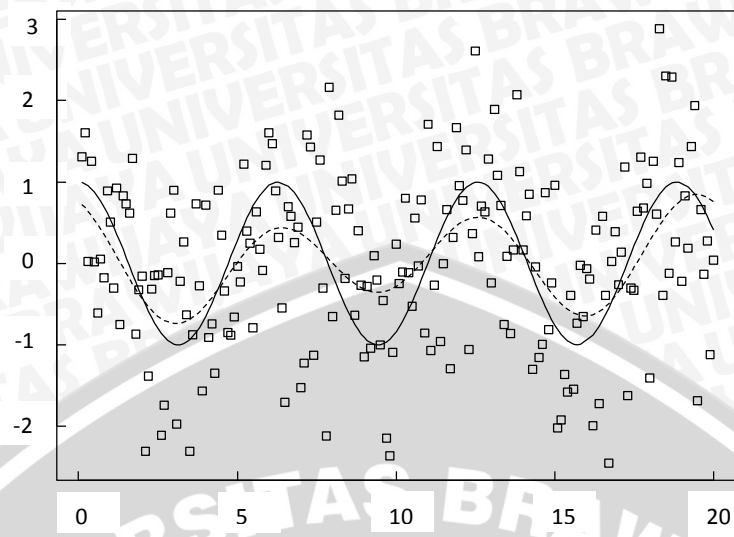
4. *Cycles* merupakan pola siklis bentuk data yang dapat naik dan turun tidak berdasarkan periode waktu yang tentu. Pola siklis terbentuk biasanya pada tiap tahun berganti (Heizer & Render, 2011). Perbedaan utama antara pola siklis dan musiman adalah pada pola siklis memiliki panjang periode yang konstan dan berulang secara periodik teratur, sedangkan persamaanya adalah sama-sama memiliki periode panjangnya waktu. Namun, panjang rata-rata periode siklis biasanya lebih lama dari musiman dan besarnya siklis biasanya lebih bervariasi daripada musiman seperti pada Gambar 2.9 (Makridakis, et al., 1998).



Gambar 2.9 Angka kematian karena kecelakaan U.S.A., 1973–1978

Sumber: Brockwell dan Davis (2002)

5. *Random* merupakan pola acak pada data yang disebabkan oleh situasi kondisi yang tidak biasa dan tidak mengikuti pola yang jelas. Oleh sebab itu, pada pola ini terdapat kemungkinan untuk sulit diprediksi (Heizer & Render, 2011). Pada pola acak ini tidak memiliki periode yang pasti, bahkan terjadi secara acak yang diulang beberapa kali tanpa memperhatikan periode tertentu seperti pada Gambar 2.10 (Palit & Popovic, 2006).



Gambar 2.10 Permasalahan deteksi sinyal

Sumber: Brockwell dan Davis (2002)

Terdapat jenis pola pada data runtut waktu yang telah dijelaskan sebelumnya dan menjadi karakteristik tersendiri. Menurut Palit dan Popovic (2006) karakteristik utama pada data runtut waktu adalah stasioneritas, linieritas, tren, dan musiman. Pada karakteristik tren dan musiman telah dijelaskan pada paragraf sebelumnya. Berikut merupakan penjelasan tambahan mengenai stasioneritas dan linieritas.

1. Stasioneritas berkaitan dengan nilai rata-rata dan varians pada data runtut waktu. Pada kedua hal tersebut, memiliki nilai yang konstan atau stabil dari waktu ke waktu. Serta kovarians antara X_t dan X_{t-1} memiliki nilai jarak selisih yang saling ketergantungan dan tidak berubah dari waktu ke waktu (Palit & Popovic, 2006).
2. Linieritas berkaitan dengan nilai data runtut waktu yang bergantung atau berhubungan pada keadaan nilai pada data itu sendiri. Oleh sebab itu, dari keadaan tersebut yang akan menentukan pola data runtut waktu tersebut.

Terkadang terdapat kesulitan untuk mengetahui tingkat linieritas yang linier atau non linier. Pada model data non linier memiliki kemungkinan banyaknya variasi dan intensitas data yang berbeda-beda. Menurut Palit dan Popovic (2006) pada kasus bidang akuntansi untuk mengetahui linieritas data dapat menggunakan model linier terlebih dahulu. Apabila model linier tidak sesuai, maka selanjutnya menggunakan model non linier untuk diuji hasilnya.

2.3.3 Pengolahan Awal Data Runtut Waktu

Secara umum sebelum melakukan proses pengolahan data, terdapat pra proses pengolahan awal pada data. Hal tersebut perlu diterapkan juga pada data runtut waktu. Tahap ini lebih populer disebut dengan *preprocessing* data yang merupakan tahapan awal untuk mempersiapkan data. Persiapan data yang baik diperlukan karena kualitas *input* data dapat mempengaruhi kualitas hasil *output* data (Junaedi, et al., 2011). Data yang berkualitas meliputi akurasi data,

kelengkapan data, konsistensi data, data selalu terupdate, kepercayaan data, dan mudah dipahami (Han, et al., 2011). Ada beberapa teknik data *preprocessing* yang dapat meningkatkan akurasi dan efisiensi algoritma diantaranya seperti berikut (Han, et al., 2011).

1. Pembersihan data berguna untuk merapikan dan memperbaiki ketidakkonsistenan 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?

Disisi lain, menurut Han, et al. (2011) data dunia nyata cenderung tidak lengkap, tidak rapi, dan tidak konsisten. Dalam penelitian ini menggunakan data runut waktu curah hujan, sehingga diperlukan penanganan seperti data tidak lengkap atau mungkin adanya data kosong karena bermasalah (*missing value*). Dalam penanganan *missing value* dapat diabaikan atau dengan mengisi nilai secara manual berdasarkan nilai probabilitas, nilai tengah, dan nilai rata-rata (Han, et al., 2011). Merapikan data merupakan penggantian nilai data yang tidak biasa dengan melakukan penghalusan atau menggantikan dengan nilai data yang wajar. Dalam merapikan data dapat menggunakan metode *binning*, klastering dan regresi, sedangkan untuk konsistensi data dapat dilakukan secara manual dengan menyamakan representasi data yang akan digunakan (Han, et al., 2011).

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 untuk transformasi data yang dapat diterapkan pada ruang *input* pada data runut waktu (Crone, et al., 2006a). Normalisasi data merupakan proses penyederhanaan data kedalam skala kecil yang telah ditentukan. Dalam penentuan batas skala tersebut dapat ditentukan antara -1 sampai 1 atau 0 sampai 1, dan batas skala lainnya (Han, et al., 2011). Terdapat beberapa metode normalisasi data yang dapat digunakan, salah satunya metode *Min-max* pada persamaan (2.7). (Han, et al., 2011).

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A \quad (2.7)$$

Dimana,

v : nilai normal

A : Atribut

$\min_A$: nilai minimum pada atribut A

$\max_A$: nilai maksimum pada atribut A

new_min_A : nilai minimum baru pada nilai normal

new_max_A : nilai maksimum baru pada nilai normal

Metode *min-max* menghubungkan nilai-nilai pada data awal berdasarkan skala yang ditentukan, sehingga dapat meminimalisir kesalahan pada data masukan yang berada di luar batas rentang pada data asli dengan atribut A (Han, et al., 2011). Proses normalisasi dapat menyederhanakan data awal atau data asli menjadi kecil, serta dapat meratakan kesamaan data dan *input* data yang terbaru (Han, et al., 2011). Tahap normalisasi data dilakukan sebelum proses pelatihan dan pengujian untuk memastikan jarak masing-masing data tidak terlalu jauh (El Margae, et al., 2014).

2.4 Curah Hujan

Didalam siklus hidrologi, jika adanya udara berupa uap air dari laut yang bergerak ke atas sehingga menjadi awan dan suhu awan tersebut mendekati titik embun maka terjadi proses pengembunan uap air yang disebut dengan kondensasi (*condensation*) (Agus & Hartati, 2011). Selanjutnya, hasil pengembunan tersebut akan mengendap di udara atau jatuh dari awan turun ke permukaan bumi yang disebut dengan presipitasi (*precipitation*) (Agus & Hartati, 2011). Sebelum jatuhnya presipitasi ke permukaan bumi berupa partikel air atau Kristal es yang menguap terlebih dahulu disebut dengan *virga* atau *stalaktit* (Tjasyono HK, 2012a).

Presipitasi berawal dari adanya awan yang memiliki beberapa situasi kondisi tertentu. Presipitasi dapat berupa air dalam bentuk cair yaitu hujan atau padat yaitu batu es hujan atau salju yang jatuh ke permukaan bumi (Tjasyono HK, 2012a). Menurut Agus dan Hartati (2011) presipitasi berdasarkan arahnya, seperti arah horizontal dapat berupa embun, kabut, dan sebagainya, sedangkan arah vertikal dapat berupa hujan (hujan batu es atau salju). Berikut merupakan sifat umum hujan yang berkaitan dengan jenis awan (Wirjohamidjojo & Swarinoto, 2007).

1. Hujan dengan intensitas atau kelembatan tinggi dengan durasi waktu yang singkat pada umumnya dapat diperkirakan hujan berasal dari awan jenis *kumulonimbus* atau *cumulus*.
2. Hujan dengan intensitas atau kelembatan rendah pada umumnya berasal dari awan yang sejenis *stratus*.
3. Hujan durasi waktu lama pada umumnya berasal dari awan *altostratus* tebal atau dari *nimbostratus*.

Hujan merupakan salah satu bentuk presipitasi yang sering terjadi di bumi (Tjasyono HK, 2012a). Dalam arti lain, hujan merupakan hasil dari penguapan dari awan sehingga turun ke permukaan bumi dapat berupa bentuk air, salju dan es (Indrabayu, et al., 2012). Menurut (Wirjohamidjojo & Swarinoto, 2007) berdasarkan ciri-ciri hujan yang turun ke permukaan bumi hujan diberi nama sebagai berikut.

1. Hujan batu berupa benda pada es batu secara umum berasal dari awan kumulonimbus yang tinggi.
2. Hujan gerimis berupa butiran air kecil yang berasal dari awan stratus atau altostratus.
3. Hujan deras (*Shower*) biasanya berlangsung dengan cepat yang berasal dari awan kumulonimbus.
4. Hujan jujuh (*Continues rain*) merupakan hujan yang turun lebat dan berlangsung secara kontinu yang berasal dari awan altostratus tebal atau nimbostratus.
5. Hujan suri merupakan hujan yang turun hanya sedikit saja sehingga memungkinkan tidak dapat terukur.
6. Hujan merah merupakan hujan yang terdapat kandungan debu atau zat bahan lain sehingga memberikan efek warna air kemerahan.

Berdasarkan wilayah di Indonesia presipitasi biasanya dimaksud dengan curah hujan (Tjasyono HK, 2012a). Hujan yang turun ke permukaan bumi dapat diukur berdasarkan volume air yang tertampung dalam satuan luas wadah, sedangkan hasil pengukuran tersebut disebut dengan curah hujan (Manullang & Tamba, 2013). Curah dalam arti dasar yang berarti banyaknya tumpahan air yang turun, sehingga curah hujan merupakan banyaknya hujan yang tercurah atau turun di suatu tempat atau limpahan hujan (Tim Pusat Bahasa Depdiknas, 2008).

Dalam arti lain, curah hujan merupakan hasil pengamatan dari perhitungan banyaknya air hujan yang jatuh ke permukaan bumi berdasarkan ketinggian terkumpulnya air dalam penakaran dengan satuan waktu millimeter (MM) (Indrabayu, et al., 2012). Sehubungan hal tersebut, penakar curah hujan harus memenuhi syarat bahwa tidak adanya proses penguapan, peresapan atau mengalir di luar wadah penakar (Manullang & Tamba, 2013).

2.4.1 Dampak Curah Hujan

Curah hujan menjadi salah satu faktor utama dalam meteorologi yang dapat menyebabkan banjir (Tjasyono HK, 2012b). Terdapat faktor lain selain meteorologi yang berasal dari sifat-sifat tanah di wilayahnya seperti tanah miring, tanah gundul bahkan pendangkalan sungai. Berikut merupakan beberapa jenis hujan yang sering terjadi secara umum (Tjasyono HK, 2012b).

1. Hujan gerimis disebut dengan *drizzle* berdasarkan penakaran curah hujan biasanya pada hujan ini memiliki intensitas curah hujan yang diameternya kecil kurang dari 1 mm. Pada hujan ini secara perlahan memiliki kesempatan untuk dapat meresap ke dalam permukaan tanah.

2. Hujan deras disebut dengan *shower* berdasarkan penakaran curah hujan biasanya pada hujan ini memiliki intensitas curah hujan yang tinggi diameternya antara 4 sampai 6 mm. Pada hujan ini secara cepat dengan intensitas besar akan menutup permukaan tanah, sehingga banyaknya air hujan yang jatuh kepermukaan tanah tidak sempat meresap ke dalam tanah dan memungkinkan terjadinya banjir.

Meskipun curah hujan dapat menimbulkan hal negatif bagi masyarakat, di sisi lain curah hujan memiliki manfaat positif bagi masyarakat seperti bidang pertanian dan hidrologi. Tergantung pada pemanfaatannya, beberapa studi kasus curah hujan dengan intensitas tinggi memiliki peluang energi yang dapat menghasilkan energi listrik. Oleh karena itu, diperlukan tingkat kesadaran yang tinggi akan pengelolaan dan pemanfaatan turunya hujan ke permukaan bumi (Tjasyono HK, 2012b). Berdasarkan dampak curah hujan yang terjadi dapat dikategorikan seperti berikut (Tjasyono HK, 2012b).

1. Dampak secara langsung memungkinkan dampaknya dapat dirasakan cepat dan lambat. Secara umum dampak secara langsung terjadi karena adanya fenomena alam yang ekstrim, seperti curah hujan yang lebat berlangsung secara terus menerus sehingga menimbulkan banjir dan tanah longsor.
2. Dampak secara tidak langsung memungkinkan dampaknya dapat dirasakan pada waktu kedepan yang dipengaruhi oleh fenomena alam yang secara terus-menerus berlangsung, seperti tidak adanya hujan sehingga dapat menimbulkan kekeringan pada tanaman.

Berdasarkan kategori sebelumnya, sehingga dapat diperhatikan beberapa kejadian dampak terhadap curah hujan yang telah terjadi disuatu wilayah. Pada Tabel 2.4 merupakan beberapa dampak curah hujan yang terjadi beserta upaya atau solusinya di bidang pertanian berdasarkan musim hujan dan musim kemarau.

Tabel 2.4 Dampak curah hujan pada musim hujan dan musim kemarau

Musim Hujan		Musim Kemarau	
Keuntungan (1)	Kerugian (2)	Keuntungan (3)	Kerugian (4)
Memenuhi kebutuhan air pada tanaman.	Jika terlalu berlebihan dapat menyebabkan banjir.	Tidak ada banjir.	Jika jumlah hujan sedikit dapat menyebabkan kekeringan pada tanaman.
Upaya atau Solusi			
✓ (1)(2)(4) Sistem waduk dan irigasi. ✓ (2)(4) Waspada dengan memanfaatkan informasi cuaca untuk mengatur antisipasi terjadinya banjir dan kekeringan.			

Sumber: Wirjohamidjojo dan Swarinoto (2007)

2.4.2 Pengukuran Curah Hujan

Pada proses pengukuran curah hujan diperlukan sebuah alat untuk dapat menangkap curah hujan yang terjadi di suatu tempat. Penakar curah hujan merupakan instrumental yang berguna untuk menampung curah hujan, sehingga diketahui jumlah curah hujan menggunakan satuan waktu tertentu (Manullang & Tamba, 2013). Alat penakar curah hujan dipasang pada tempat yang bebas tanpa halangan agar dapat murni menampung air yang jatuh dari awan (Wirjohamidjojo & Swarinoto, 2007). Pengukuran curah hujan yang tertampung akan memiliki volume dan dibagi dengan luasnya corong penakar, sehingga diperoleh tebal atau tingginya curah hujan berupa air yang tertampung dengan satuan milimeter (MM) (Manullang & Tamba, 2013). Secara umum alat penakar curah hujan dikategorikan secara konvensional (manual) dan otomatis (Muliantara, et al., 2015).

1. Penakar curah hujan secara konvensional atau manual tidak dapat otomatis mencatat hasil penakaran curah hujan (*non recording*). Salah satu alat penakar curah hujan ini adalah tipe observatorium yang berbentuk silinder (Endriyanto & Ihsan, 2011).
2. Penakar curah hujan otomatis dapat mencatat sendiri hasil penakaran curah hujan (*recording*). Salah satu alat penakar curah hujan ini adalah tipe hellman (Muliantara, et al., 2015).

Dalam pengukuran banyaknya curah hujan yang jatuh ke permukaan bumi pada selang waktu tertentu secara umum menggunakan satuan mm atau inci. Dapat diasumsikan jika terdapat 1 mm curah hujan pada waktu tertentu maka dapat diartikan banyaknya curah hujan yang tertampung pada wadah seluas 1 m² sama dengan 1 liter air yang dijabarkan pada persamaan (2.8) (Wirjohamidjojo & Swarinoto, 2007).

$$1 \text{ m}^2 \times 1 \text{ mm} = 1.000.000 \text{ mm}^3 = 1 \text{ liter} \quad (2.8)$$

Terdapat pengukuran kepadatan hujan yang menyatakan seringnya terjadi hujan atau banyaknya hari hujan pada waktu tertentu, semisal pada kurun waktu satu bulan yang terdapat 30 hari. Dengan demikian dapat diasumsikan, apabila terjadi 10 hari hujan dengan menggunakan persamaan (2.9) sehingga diketahui adanya kepadatan hujan lebih dari 0.3 mm (Wirjohamidjojo & Swarinoto, 2007).

$$D = \frac{h}{B} \quad (2.9)$$

Dimana,

D : kepadatan hujan

h : banyaknya hari hujan

B : total banyaknya satuan hari

Curah hujan kumulatif merupakan terkumpulnya curah hujan pada rentang waktu tertentu (BMKG Stasiun Klimatologi Karangploso, 2015b). Perhitungan curah hujan kumulatif diperoleh dengan menjumlahkan nilai curah hujan pertama hingga nilai curah hujan berikutnya secara berlanjut pada batas waktu yang ditentukan (Wirjohamidjojo & Swarinoto, 2007). Batas waktu dalam pengukuran

merupakan pencatatan durasi lamanya waktu hujan yang turun ke permukaan bumi dalam satuan waktu (menit, jam, hari dan satuan waktu lainnya) (Agus & Hartati, 2011). Batas waktu lainnya seperti harian, dasarian, bulanan dan musiman sesuai zona musim yang ditentukan. Dasarian merupakan rentang waktu yang dimulai selama 10 hari, selama satu bulan terdapat 3 dasarian dimulai tanggal 1 sampai 10, tanggal 11 sampai 20, dan tanggal 21 hingga akhir bulan (BMKG Stasiun Klimatologi Karangploso, 2015b).

Zona musim merupakan cakupan wilayah yang memiliki musim serupa seperti musim hujan dan kemarau. Berbeda dengan cakupan daerah administrasi pemerintahan seperti kabupaten atau kota. Lain halnya, dengan cakupan wilayah yang kurang jelas pola hujannya untuk diketahui termasuk musim hujan atau kemarau disebut non zona musim (BMKG Stasiun Klimatologi Karangploso, 2015b). Berdasarkan hujan di wilayah tertentu dapat ditafsirkan memiliki sifat hujan seperti berikut (Wirjohamidjojo & Swarinoto, 2007).

1. Hujan di cakupan wilayah lintang tengah dan tinggi:
 - Hujan dapat berupa bentuk es atau salju.
 - Intensitas hujan relatif rendah dengan durasi waktu yang lama.
 - Hujan terjadi di daerah perenggan dan siklon.
2. Hujan di cakupan wilayah tropik:
 - Hujan sangat jarang dalam bentuk es hanya dalam bentuk air.
 - Intensitas hujan relative besar dengan durasi waktu yang pendek.
 - Hujan di daerah pesisir sering terjadi ketika malam hari atau menjelang pagi hari.
 - Hujan di atas daratan pada umumnya biasa terjadi pada waktu siang atau sore hari.
 - Hujan di daerah pegunungan biasanya terjadi pada sore hari atau menjelang malam hari.

Dengan adanya pengukuran curah hujan sehingga dapat diperoleh informasi tentang terjadinya hujan yang terjadi di suatu wilayah. Selain itu, dapat dianalisis dan diketahui sifat dasar hujan yang terjadi berdasarkan beberapa faktor tertentu. Berikut merupakan faktor-faktor yang menjadi sifat dasar hujan yang perlu diukur (Wirjohamidjojo & Swarinoto, 2007).

1. Jumlah atau banyaknya air hujan yang tertampung selama pengamatan dengan satuan milimeter (MM). Curah hujan 1 mm yang tertampung pada wadah permukaan datar seluas 1 m² dan sama dengan sebanyak 1 liter.
2. Intensitas atau kelembatan hujan merupakan jumlah atau banyaknya hujan turun pada satuan waktu tertentu seperti mm/detik.
3. Hari hujan merupakan jumlah atau banyaknya hari dengan terjadinya turunnya hujan pada waktu tertentu seperti hari per minggu atau per bulan. Biasanya hari hujan dapat ditetapkan hari hujan apabila terjadi hujan minimal sebanyak 0.5 mm.

2.4.3 Intensitas Curah Hujan

Cuaca atau iklim yang ekstrim terjadi apabila terdapat penyimpangan terhadap nilai rata-rata atau batas ambang meteorologi disuatu wilayah (BMKG Stasiun Klimatologi Karangploso, 2015a). Pengukuran tersebut berdasarkan intensitas hujan yaitu banyaknya curah hujan yang turun pada saat waktu hujan. Pengukuran intensitas hujan dapat diasumsikan pada persamaan (2.10) (Wirjohamidjojo & Swarinoto, 2007).

$$I = \frac{r}{t} \quad (2.10)$$

Dimana,

I : intensitas hujan

r : banyaknya curah hujan (MM)

t : periode waktu terjadinya hujan (menit).

Sebagai contoh studi kasus terjadi hujan selama 5 menit dan berdasarkan pengukurannya setinggi 15 mm maka intensitasnya 3 mm/menit seperti pada (2.11) (Wirjohamidjojo & Swarinoto, 2007).

$$I = \frac{15}{3} = 3 \text{ (mm/menit)} \quad (2.11)$$

Dalam pengukuran intensitas hujan dapat dilakukan pada satuan durasi tertentu selain contoh studi kasus di atas seperti (mm/jam, cm/hari, dan lainnya) (Agus & Hartati, 2011). Pada Tabel 2.5 merupakan ketentuan BMKG secara umum tentang batas ambang cuaca atau iklim ekstrim dilihat dari intensitas curah hujan yang terjadi pada suatu wilayah (BMKG Stasiun Klimatologi Karangploso, 2015a).

Tabel 2.5 Intensitas curah hujan

Curah Hujan Harian	Curah Hujan Bulanan	Intensitas	Tingkat Rawan Banjir
< 20 MM	< 300 MM	Ringan	Rendah
20 – 50 MM	300 – 500 MM	Sedang	Menengah
50 – 100 MM		Lebat	
> 100 MM	> 500 MM	Sangat Lebat	Tinggi

Sumber: BMKG Stasiun Klimatologi Karangploso (2015a)

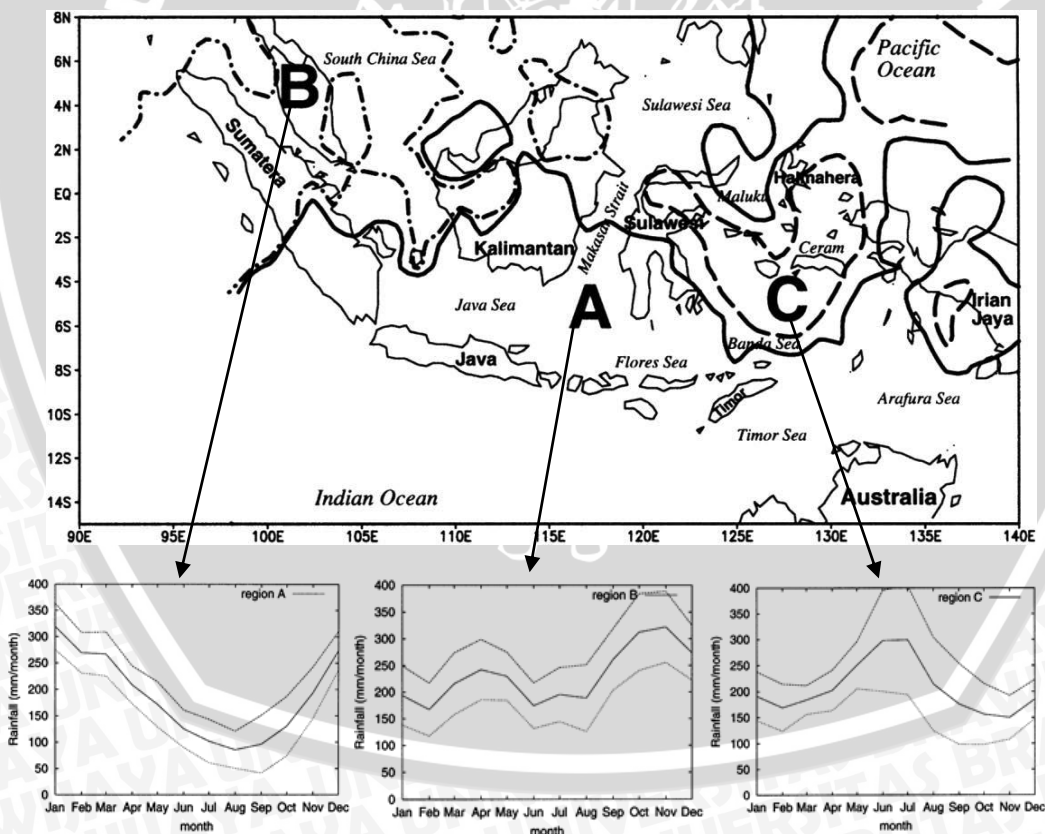
2.4.4 Normal Curah Hujan

Berdasarkan waktu yang bervariasi untuk menganalisis curah hujan di suatu tempat. Berguna untuk melakukan analisis perbandingan curah hujan yang terjadi, sehingga dihasilkan sifat-sifat hujan. Berikut merupakan normal curah hujan berdasarkan rentan waktu (BMKG Stasiun Klimatologi Karangploso, 2015a).

1. Rata-rata curah hujan bulanan merupakan nilai rata-rata curah hujan pada tiap-tiap bulan dengan periode minimal 10 tahun.
2. Provisional normal curah hujan bulanan merupakan nilai rata-rata curah hujan pada tiap-tiap bulan dengan periode waktu yang dapat ditentukan secara bebas dan disyaratkan minimal 10 tahun.
3. Normal curah hujan bulanan merupakan nilai rata-rata curah hujan pada tiap-tiap bulan selama periode 30 tahun.
4. Standar normal curah hujan bulanan adalah nilai rata-rata curah hujan pada tiap-tiap bulan berkelipatan atau selama periode 30 tahun, seperti dimulai dari tahun 1901 sampai dengan 1930 dan tahun seterusnya.

2.4.5 Pola Curah Hujan Indonesia

Menurut Aldrian dan Dwi Susanto (2003) terdapat 3 pola iklim utama di negara Indonesia, jika dilihat berdasarkan terjadinya hujan selama setahun. Dengan kata lain, dengan adanya hujan yang terjadi dapat diketahui pola curah hujan. Berikut merupakan pola curah hujan berdasarkan rata-rata tahunan terjadinya hujan di beberapa wilayah, sehingga dapat dikategorikan tipe curah hujannya pada Gambar 2.11 beserta penjelasannya (Aldrian & Dwi Susanto, 2003).



Gambar 2.11 Pola curah hujan di Indonesia

Sumber: Aldrian dan Dwi Susanto (2003)

- A. Pola Monsunal bersifat unimodial yaitu terdapat satu puncak musim hujan pada akhir tahun dan awal tahun selanjutnya, biasanya pada bulan desember sampai februari. Pada enam bulan selanjutnya merupakan musim pancaroba yaitu tiga bulan perubahan dari musim hujan ke musim kemarau dan tiga bulan sisanya perubahan dari musim kemarau ke musim hujan. Musim kering biasanya terjadi pada pertengahan tahun biasanya bulan juni, juli, dan agustus. Pola ini biasanya dominan terjadi di pulau Sumatra bagian selatan, Kalimantan selatan dan tengah, Jawa, Bali, Nusa Tenggara dan beberapa sebagian papua.
- B. Pola Ekuatorial bersifat bimodial yaitu terdapat dua puncak musim hujan yang biasanya terjadi pada bulan maret dan oktober atau pada saat terjadi ekinoks. Pola ini biasanya dominan terjadi di pulau Sumatra bagian tengah dan Utara dan Kalimantan bagian utara.
- C. Pola Lokal bersifat unimodial hampir sama dengan pola monsual, namun bentuknya berlawanan. Dengan kata lain, musim hujan pada pola ini biasanya terjadi pertengahan tahun. Pola ini biasanya dominan terjadi di pulau Maluku, Sulawesi dan beberapa sebagian papua.

Dalam menetapkan permulaan awal musim hujan dapat berdasarkan pengukuran banyaknya curah hujan di mulai awal dasarian di atas 50 milimeter diikuti dasarian selanjutnya yang semakin meningkat jumlahnya. Dibandingkan dengan normalnya, musim hujan dapat terjadi lebih awal, sama, bahkan mundur (BMKG Stasiun Klimatologi Karangploso, 2015b).

2.5 Support Vector Regression (SVR)

Support Vector Machine (SVM) merupakan *machine learning* yang diperkenalkan oleh Vladimir Naumovich Vapnik pada tahun 1995 untuk mengatasi permasalahan klasifikasi (Cortes & Vapnik, 1995). Metode SVM dikembangkan untuk dapat mengatasi kasus regresi yang dikenal dengan metode *Support Vector Regression* (SVR) (Drucker, et al., 1997). Pada dasarnya metode SVR menerapkan prinsip dari metode SVM. Konseptual metode SVM berawal dari masukan vektor yang dipetakan secara non linier ke dalam ruang fitur (*feature space*) berdimensi tinggi. Di dalam ruang fitur tersebut ditunjukkan sebagai permukaan keputusan linier yang dibentuk (Cortes & Vapnik, 1995).

Dengan memanfaatkan penggunaan *$\mathcal{E}$ -insentive loss function* yang dikemukakan oleh Vapnik (1995), sehingga metode SVR dapat menangani permasalahan kasus regresi dengan baik (Hasan, et al., 2015). Metode SVR menggunakan prinsip *Structural Risk Minimisation* (SRM) untuk dapat meminimalkan batas yang diharapkan dengan nilai kesalahan (*error*) yang rendah pada proses pelatihan data (Gunn, 1998). Dalam penggunaan metode SVR dimisalkan terdapat satu set data latih (2.12).

$$\{(x_1, y_1), \dots, (x_\ell, y_\ell)\} \subset X \times R \quad (2.12)$$

Dimana,

X : ruang pada vektor dari variabel masukan

ℓ : banyaknya data

Contohnya, seperti $X = R^d$ dengan d sebagai total banyaknya fitur data dimensi yang digunakan. Pada contoh pemanfaatan data dimensi tersebut dapat digunakan seperti data *time series* curah hujan berdasarkan runut waktu tertentu. Tujuan dari metode $\mathcal{E}$ -insentive SVR untuk mencari fungsi $f(x)$ yang sebagian besar masuk dalam ruang lingkup nilai deviasi $\mathcal{E}$ berdasarkan nilai aktual y_i pada data latih, dan saat yang sama untuk dapat memungkinkan kondisi *flat*. Dengan kata lain tidak mempedulikan nilai kesalahan selama fungsi tersebut jika terdapat kurang dari nilai deviasi $\mathcal{E}$, namun tidak akan diterima jika setiap nilai penyimpangan terlalu besar. Hal tersebut penting untuk meminimalisir kehilangan nilai aktual (Smola & Schölkopf, 2004). Didalam *feature space* menggunakan bentuk fungsi dasar regresi linier seperti persamaan (2.13).

$$f(x) = \langle w, x \rangle + b \text{ dengan } w \in R, b \in R \quad (2.13)$$

Dimana,

$f(x)$: representasi prediksi

w : vektor bobot

b : merupakan nilai konstan (bias)

Kondisi *flatness* pada persamaan (2.13) bermaksud dengan mencari nilai vektor bobot w yang minimal. Salah satu cara untuk memastikan dalam meminimalkan nilai bobot w menggunakan konsep pencarian solusi optimum secara cembung (*convex optimization problem*) (Smola & Schölkopf, 2004).

$$\begin{aligned} &\text{Meminimalkan } \frac{1}{2} \|w\|^2 \\ &\text{Dengan syarat } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \mathcal{E} \\ \langle w, x_i \rangle + b - y_i \leq \mathcal{E} \end{cases} \end{aligned} \quad (2.14)$$

Dimana,

b : bias

y_i : data aktual

x_i : data prediktor

$\mathcal{E}$: nilai batas toleransi

w : vektor pembobot

Pada persamaan (2.14) dapat diasumsikan bahwa kondisi fungsi f nilai aktual terhadap variabel pasangan (x_i, y_i) tidak kurang lebih dari nilai $\mathcal{E}$ yang telah ditentukan. Dalam penggunaan konsep pencarian solusi optimum secara cembung (*convex optimization problem*) diharapkan dapat menyelesaikan permasalahan dengan mudah (Smola & Schölkopf, 2004). Namun, terkadang terdapat ketidakmampuan atau kesalahan pada fungsi tersebut dengan adanya data di luar pada nilai $\mathcal{E}$ yang telah ditentukan (Bi, et al., 2011). Dengan

penggunaan variabel *slack* ξ_i dan ξ_i^* dapat mengatasi kendala ketidakmampuan pada permasalahan *convex optimization*. Berikut merupakan formula persamaan yang dinyatakan oleh Vapnik (1995).

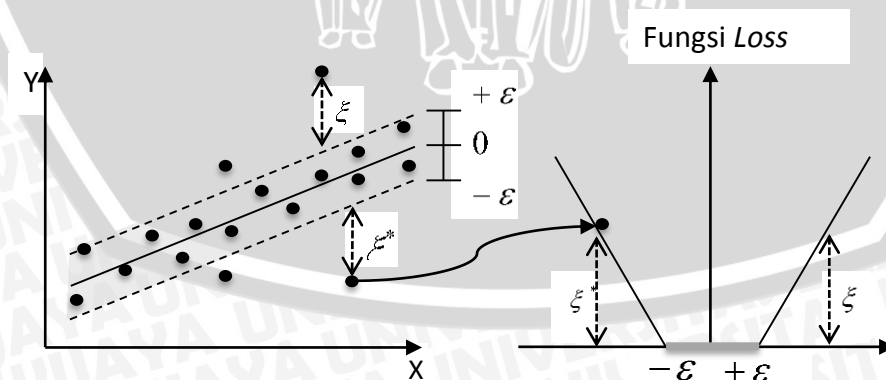
$$\begin{aligned} &\text{Meminimalkan } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*) \\ &\text{Dengan syarat } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, C \geq 0 \end{cases} \end{aligned} \quad (2.15)$$

Dimana,

- b : bias
- y : data aktual
- x : data prediktor
- ε : nilai batas toleransi
- w : vektor pembobot
- C : kompleksitas
- ξ_i, ξ_i^* : variabel *slack*

Pada variabel C merupakan konstanta penting untuk menentukan batas kompleksitas dalam pencarian fungsi f yang rata (*flatness*) dan jumlah deviasi nilai *error* pada pelatihan yang tidak melebihi batas toleransi dari nilai ε (Smola & Schölkopf, 2004). Pada permasalahan tersebut berkaitan dengan penggunaan ε -*insentive loss function* sesuai dengan persyaratan pada (2.16) dan dapat diilustrasikan pada Gambar 2.12.

$$|\xi|_{\varepsilon} = \begin{cases} 0 & \text{jika } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{lainnya} \end{cases} \quad (2.16)$$



Gambar 2.12 Grafik pengaturan *margin loss* pada SVR linier dan fungsi ε -*insentive loss function*

Sumber: Schölkopf dan Smola (2002a).

Penggunaan konsep pencarian solusi optimum secara cembung (*convex optimization problem*) untuk menyelesaikan permasalahan utama pada formulasi (2.15) dapat menyelesaikan permasalahan secara berganda (Bi, et al., 2011). Dengan menggunakan konsep *lagrangian multipliers* dapat memecahkan permasalahan masalah ganda (Lu & Wang, 2011).

$$\text{Memaksimalkan } \left\{ \begin{array}{l} -\frac{1}{2} \sum_{i,j=0}^{\ell} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(x_i \bullet x_j) \\ -\varepsilon \sum_{i=1}^{\ell} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{\ell} y_i (\alpha_i - \alpha_i^*) \end{array} \right\} \quad (2.17)$$

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

Dimana,

α_i, α_i^* : *lagrangian multipliers*

y : data aktual

x : data prediktor

ε : nilai batas toleransi

C : kompleksitas

Pada dua variabel α_i dan α_i^* merupakan *lagrangian multipliers* pada data latih ke i dengan beberapa syarat yang bertujuan untuk memaksimalkan jarak toleransi data. Dengan menggunakan *quadratic programming* pada persamaan (2.17) dapat menyelesaikan permasalahan ganda (Smola & Schölkopf, 2004). Didalam kasus klasifikasi hanya terdapat beberapa koefisien dari nilai $(\alpha_i - \alpha_i^*)$ yang tidak bernilai nol, sedangkan titik-titik data yang telah sesuai dalam batasnya disebut dengan *support vector* (Vijayakumar & Wu, 1999). Berdasarkan pembahasan persamaan sebelumnya, sehingga menghasilkan nilai vektor bobot W yang dapat dirumuskan pada persamaan (2.18).

$$w = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) x_i \quad (2.18)$$

Dimana,

α_i, α_i^* : *lagrangian multipliers*

x : data prediktor

Pada persamaan (2.18) dapat disebut *support vector expansion*. Terdapat variabel W sebagai nilai vektor bobot yang dapat menggambarkan kombinasi linier terhadap pola proses pemodelan atau pelatihan X_i . Kompleksitas terhadap fungsi yang direpresentasikan oleh *support vector* merupakan nilai independen pada dimensi ruang masukan X dan bergantung pada banyaknya jumlah *support vector* (Smola & Schölkopf, 2004). Dengan menggabungkan persamaan (2.13), sehingga didapatkan persamaan fungsi seperti (2.19).

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \quad (2.19)$$

Dimana,

α_i, α_i^* : lagrangian multipliers

x : data prediktor

b : bias

Dalam mengevaluasi fungsi $f(x)$ yang menggunakan persamaan (2.19), sehingga tidak perlu menghitung W secara rinci. Meskipun W dapat memungkinkan melakukan komputasi secara efisien pada kondisi linier. Pada perhitungan nilai b dapat diformulasikan seperti (2.20) dan (2.21) (Basak & Patranabis, 2007).

$$b = y_i - \langle w, x_i \rangle - \varepsilon \quad \text{untuk } \alpha_i \in (0, C) \quad (2.20)$$

$$b = y_i - \langle w, x_i \rangle + \varepsilon \quad \text{untuk } \alpha_i^* \in (0, C) \quad (2.21)$$

Dimana,

α_i, α_i^* : lagrangian multipliers

b : bias

y : data aktual

x : data prediktor

ε : nilai batas toleransi

W : vektor pembobot

C : kompleksitas

Sebab itu, variabel α_i, α_i^* bernilai nol untuk mengikuti kondisi KKT. Oleh karena itu, tidak ada atau bahkan jarang dalam pengembangan terhadap variabel W untuk data X_i . Beberapa contoh koefisien yang ada dalam batas toleransi yang diminimalkan disebut dengan *Support Vectors* (Basak & Patranabis, 2007).

Berdasarkan beberapa kasus terdapat nilai masukan dan keluaran yang bersifat non linier (Bi, et al., 2011). Metode SVR dapat dikembangkan untuk kasus non linier dengan memetakan variabel masukan pada pelatihan X_i yang dilakukan oleh $\Phi: X \rightarrow F$ ke dalam beberapa ruang fitur F dan menerapkannya pada metode SVR basis dasar linier (Basak & Patranabis, 2007). Dengan mengembangkan persamaan (2.18) dan (2.19) sehingga menjadi bentuk persamaan non linier (2.22) dan (2.23).

$$w = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \phi(x_i) \quad (2.22)$$

dan

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) k \langle x_i, x \rangle + b \quad (2.23)$$

Dimana,

- ϕ : pemetaan ke ruang fitur
- k : *kernel* untuk pemetaan
- α_i, α_i^* : *lagrangian multipliers*
- X : data prediktor
- b : bias

Terdapat perbedaan dengan dasar regresi linier pada penentuan nilai vektor bobot W secara rinci yang tidak perlu digunakan. Pada kasus non linier, masalah optimasi disesuaikan dengan mencari fungsi yang *flatnees* pada ruang fitur bukan pada ruang masukan. Oleh karena itu, konsep dasar SVR melakukan pendekatan dalam memecahkan masalah tersebut menggunakan persamaan (2.23) sebagai fungsi regresi (Basak & Patranabis, 2007). Pada persamaan (2.23) terdapat fungsi *kernel* $k \langle x_i, x \rangle$ yang mendefinisikan sebagai linier pada titik hasil atau *dot product* yang dipetakan secara non linier pada ruang fitur misalnya seperti $k \langle x_i, x \rangle = \phi(x_i) \phi(x)$ (Basak & Patranabis, 2007). Dalam penggunaan fungsi *kernel* pada persamaan (2.23) dibahas secara khusus pada sub bab 2.5.1.

Pada persamaan fungsi regresi (2.23) Vijayakumar dan Wu (1999) mengusulkan alternatif dalam menentukan nilai bias b yang dapat ditentukan komputasi secara rinci. Dengan menggunakan dimensi tambahan pada pola vektor seperti $x' = (x_1, \dots, x_N, \lambda)$ serta menghubungkan nilai bias ke dalam vektor bobot $w' = (w_1, \dots, w_N, b/\lambda)$. Pada dimensi tambahan tersebut, terdapat variabel λ sebagai konstanta skalar. Seperti pada persamaan (2.15) memiliki konsep berusaha untuk memaksimalkan margin meskipun sedikit berbeda dari permasalahan yang sebenarnya. Dalam memaksimalkan $\frac{2}{\|w\|}$ atau meminimalkan $\|w^2\|$, Vijayakumar dan Wu (1999) mengusulkan penyederhanaan dengan meminimalkan $\|w'\|^2 = \|w\|^2 + b^2 / \lambda^2$, sehingga dapat menggantikan variabel b pada persamaan (2.23) menjadi λ^2 seperti pada persamaan (2.24).

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) k \langle x_i, x \rangle + \lambda^2 \quad (2.24)$$

Dimana,

- λ^2 : Variabel skalar
- k : *kernel* untuk pemetaan
- α_i, α_i^* : *lagrangian multipliers*
- X : data prediktor

Menurut Vijayakumar dan Wu (1999) pada pengaturan nilai parameter λ yang terlalu besar biasanya akan mengakibatkan ketidakstabilan dan lambatnya konvergensi dalam proses pelatihan. Oleh sebab itu, pengaturan nilai parameter λ perlu disesuaikan dengan sama besarnya nilai vektor masukan. Namun, ketika menggunakan *kernel* non linier belum tentu mempengaruhi konvergensi. Hal tersebut sangat bergantung pada *kernel* yang digunakan. Dengan memberikan pengaturan nilai nol pada parameter λ atau seperti mengabaikan parameter bias di dalam ruang fitur berdimensi tinggi akan memberikan hasil yang baik pada data objek kasus dunia nyata (Vijayakumar & Wu, 1999).

Dengan menggunakan variabel λ^2 sebagai alternatif b dalam penerapan metode SVR dapat menggunakan algoritma sekuensial pelatihan SVR yang diusulkan oleh Vijayakumar dan Wu (1999), sehingga memudahkan dalam penerapan metode SVR dengan konseptual sederhana untuk dapat menyelesaikan komputasi yang kompleks. algoritma sekuensial pelatihan SVR terbukti dapat mengoptimalkan kinerja metode SVM baik kasus klasifikasi maupun regresi (Vijayakumar & Wu, 1999).

Dalam penggunaan metode SVR yang telah dijelaskan sebelumnya, terdapat parameter ε yang dapat berfungsi dengan baik apabila telah dilakukan pendekatan terhadap besar kecilnya nilai parameter ε yang sesuai (Schölkopf, et al., 1998). Dengan adanya permasalahan tersebut, Schölkopf, et al., (1998) mengembangkan metode ε -SVR yang dapat menyesuaikan nilai parameter ε secara otomatis untuk dapat meningkatkan akurasi. Pengembangan tersebut menghasilkan metode V -SVR, sehingga parameter V menjadi batas atas dalam poin *error* atau batas bawah *support vector* pada poin ε -insensitive tube SVR. Oleh karena itu, besar kecilnya nilai parameter ε akan ditentukan oleh besar kecilnya nilai parameter V [0,1] (Basak & Patranabis, 2007).

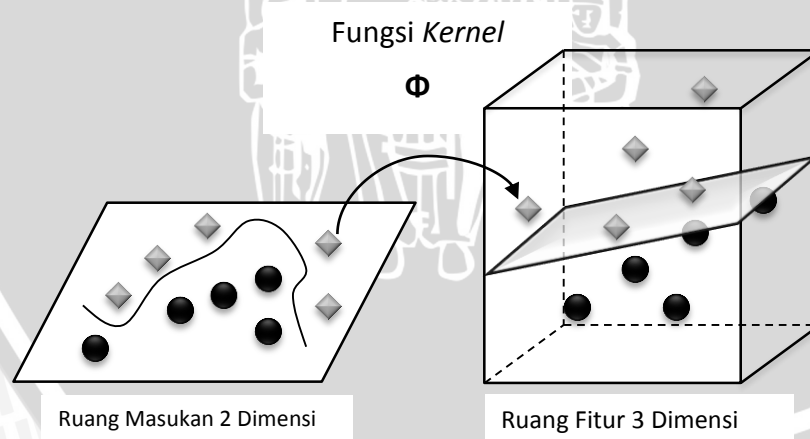
Menurut Chang dan Lin (2002) nilai parameter ε memang tidak seharusnya bernilai terlalu kecil karena akan berakibat *overfitting*. *Overfitting* merupakan kondisi tahap pelatihan yang sangat baik, namun buruk pada tahap pengujian pada data yang baru. Secara intuitif berhubungan dengan nilai parameter ε dan dipengaruhi oleh nilai target y . Oleh sebab itu, alternatif lain untuk mengatasi permasalahan ε -SVR dengan menormalisasikan nilai target sebelum tahap pelatihan. Dimisalkan terdapat nilai target yang berjarak $[-1, +1]$, secara efektif jarak parameter ε dinormalisasikan dengan jarak [0,1] (Chang & Lin, 2002). Hal tersebut memiliki kesamaan dengan konsep parameter V pada V -SVR yang dapat memudahkan dalam penyesuaian parameter ε (Chang & Lin, 2002). Disamping normalisasi pada nilai target juga diperlukan untuk dapat menyesuaikan nilai parameter C dengan kompleksitas nilai target yang digunakan pada metode ε -SVR (Chang & Lin, 2002).

2.5.1 Kernel Trick

Istilah *kernel* berasal dari bahasa asing yang berarti inti, pada arti lain sebagai biji yang terkandung dalam buah (Sewell, 2007). Menurut kamus komputer terbitan Microsoft Press (2002), istilah *kernel* merupakan inti dari sebuah sistem

operasi serta bagian dari sistem yang berperan penting dalam pengelolaan memori, file dan perangkat keras, pengelolaan waktu dan tanggal, eksekusi aplikasi, dan mengalokasikan sumber daya sistem (Microsoft Press, 2002). Dengan kata lain *kernel* pada sistem operasi menjembatani antara perangkat lunak dengan perangkat keras. Lain halnya, pada komputer grafik juga menggunakan istilah *kernel* yang biasa disebut dengan *Graphical Kernel System* (GKS). GKS berfungsi untuk menjembatani antara programmer dengan aplikasi yang menggunakan sebuah standar komputer grafik untuk membuat, memanipulasi, dan menampilkan grafik pada komputer (Microsoft Press, 2002). Namun, sebelumnya istilah *kernel* pertama kali digunakan dalam bidang matematika sebagai persamaan integral dan fungsi lainnya (Sewell, 2007).

Pada *machine learning* juga terdapat istilah metode *kernel* yang berfungsi memetakan data pada ruang fitur dimensi yang tinggi berupaya meningkatkan komputasi secara linier (Genton, 2001). *Support Vector Machine* (SVM) merupakan salah satu metode *machine learning* yang menerapkan metode *kernel* untuk menangani masalah klasifikasi, regresi, estimasi, dan klasterisasi (Genton, 2001). Dengan menggunakan fungsi *kernel* $k(x, x')$ seperti pada persamaan (2.24) pada metode SVM untuk kasus regresi yang digunakan pada penelitian ini. Pada fungsi *kernel* $k(x, x') = \phi(x)\phi(x')$ melakukan pemetaan menggunakan simbol ϕ terhadap *dot product* secara non linier pada ruang fitur berdimensi tinggi (Basak & Patranabis, 2007). Pendekatan tersebut sering disebut dengan istilah *kernel trick* (Schölkopf & Smola, 2002a). *Kernel trick* berfungsi menjembatani antara non linier dan linier, sehingga model linier pada ruang fitur dapat menyesuaikan kasus data model non linier dari ruang input (Sewell, 2007). *Kernel Trick* dapat diilustrasikan pada kasus klasifikasi metode SVM seperti Gambar 2.13.



Gambar 2.13 Pemetaan Φ data ke dimensi tinggi pada ruang fitur untuk kasus klasifikasi

Sumber: Kaundal, et al. (2006).

Pada Gambar 2.13 merupakan ilustrasi dalam usaha pemisahan atau klasifikasi data melalui ruang fitur 3 dimensi. Pada ruang fitur 3 dimensi melakukan pemetaan secara linier dan adaptasi pemisahan data dari ruang masukan 2

dimensi secara non linier. Hal tersebut merupakan fungsi dari penggunaan *kernel trick* untuk dapat menangani kasus non linier. Dalam penggunaan *kernel trick* pada metode SVM terdapat beberapa jenis fungsi *kernel* yang sering digunakan seperti *linear kernel*, *polynomial kernel* dan *gaussian radial basis function kernel* (Lu & Wang, 2011). Berikut merupakan penjelasan fungsi *kernel* yang dapat diterapkan pada metode SVM.

1. *Linear kernel* merupakan *kernel* dasar yang paling sederhana jika dibandingkan dengan fungsi *kernel* lainnya. Fungsi *linear kernel* hanya melakukan pemetaan ϕ secara linier seperti pada persamaan (2.25).

$$k(x, x') = (x \cdot x') = (\phi(x) \cdot \phi(x')) \quad (2.25)$$

Dimana,

ϕ : pemetaan ke ruang fitur

$x \cdot x'$: vektor data

Meskipun hanya dengan melakukan pemetaan secara sederhana tanpa adanya parameter, seperti pada persamaan (2.25) masih tergolong dalam fungsi *kernel* (Deng, et al., 2012). Pada *linear kernel* tersebut dapat diterapkan pada metode SVM dengan data sederhana yang bersifat linier.

2. *Polynomial kernel* memiliki kesamaan terhadap *linier kernel* dengan penambahan parameter $d \geq 0$ sebagai pemangkatan yang bernilai bilangan bulat positif. Pada persamaan (2.26) disebut dengan *homogeneous polynomial kernel* (Deng, et al., 2012).

$$k(x, x') = (x \cdot x')^d \quad (2.26)$$

Dimana,

$x \cdot x'$: vektor data

d : pangkat (*degree*)

Terdapat parameter *degree- d polynomial* melakukan pemetaan pada ruang fitur yaitu $\{x_1^2, x_1x_2, x_2^2\}$ (Vert, et al., 2004). Kesamaan terhadap *linier kernel* pada persamaan (2.25) terletak pada parameter d , jika bernilai 1 (Ben-Hur & Weston, 2010).

$$k(x, x') = ((x \cdot x') + c)^d \quad (2.27)$$

Dimana,

$x \cdot x'$: vektor data

c : kompleksitas

d : pangkat (*degree*)

Pada persamaan (2.27) disebut dengan *non-homogeneous polynomial kernel*. Terdapat penambahan parameter $c \geq 0$ konstanta yang melakukan pemetaan pada ruang fitur dengan 2 parameter menjadi seperti $\{1, x_1, x_2, x_1^2, x_1x_2, x_2^2\}$ (Vert, et al., 2004). Namun, pada beberapa penelitian mengatur nilai parameter

c dengan nilai 1, seperti pada persamaan (2.27) (Ma, et al., 2015; Ben-Hur & Weston, 2010; Deng, et al., 2012; Chen, et al., 2004).

$$k(x, x') = ((x \cdot x') + 1)^d \quad (2.28)$$

Dimana,

$x \cdot x'$: vektor data

d : pangkat (*degree*)

3. *Gaussian kernel* lebih dikenal dengan *Radial Basis Function* (RBF) yang memiliki ciri khas pada persamaan fungsinya menggunakan *squared exponential* (Rasmussen & Williams, 2006). *Gaussian RBF kernel* merupakan *kernel* yang populer untuk diterapkan pada metode SVM. Namun, diperlukan penyesuaian terhadap lebarnya parameter σ yang akan digunakan (Schölkopf & Smola, 2002b). Fungsi dari *gaussian kernel* seperti pada persamaan (2.29).

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (2.29)$$

Dimana,

$x_i - x_j$: vektor data

σ : sigma

Pada fungsi *gaussian kernel* terdapat parameter $\sigma > 0$ yang berperan penting pada kinerja *kernel*. Apabila parameter σ bernilai kecil akan melakukan pendekatan maksimal terhadap *support vector* atau data dan menjadi hal yang kompleks (Vert, et al., 2004). Sebaliknya, jika nilai parameter ini terlalu tinggi menyebabkan *kernel* cenderung bersifat lebih ke linier dan mengurangi kemampuan dalam memetakan data pada dimensi tinggi (Appati, et al., 2014). Akibatnya, dalam mengatur parameter $\sigma > 0$ menjadi hal yang sensitive pada proses pelatihan (Appati, et al., 2014). Terdapat persamaan (2.30) alternative lain pada *gaussian kernel*.

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (2.30)$$

Dimana,

$x_i - x_j$: vektor data

γ : gamma

Pada persamaan (2.30) menggunakan parameter γ sebagai rumus alternatif $\frac{1}{2\sigma^2}$. Variabel $\gamma > 0$ juga berperan penting terhadap kinerja *Gaussian kernel* seperti peran pada parameter d *kernel polynomial*. Tingkat kesensitifan parameter γ berbanding terbalik dengan parameter σ . Apabila pengaturan nilai parameter γ semakin kecil, cenderung mendekati sifat linier dan

sebaliknya (Ben-Hur & Weston, 2010). Terdapat fungsi *kernel* lain yang termasuk dalam kategori RBF *kernel*, seperti *exponential kernel* pada persamaan (2.31).

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{2\sigma^2}\right) \quad (2.31)$$

Dimana,

$x_i - x_j$: vektor data

σ : sigma

Pada fungsi *exponential kernel* terdapat hubungan yang sama terhadap *gaussian kernel* dan membedakan dengan menghilangkan kuadrat. Begitu juga dengan *laplacian kernel* yang termasuk kategori RBF *kernel* seperti pada persamaan (2.32) (Fadel, et al., 2016).

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{\sigma}\right) \quad (2.32)$$

Dimana,

$x_i - x_j$: vektor data

σ : sigma

Pada *laplacian kernel* ini yang menjadi dasar terhadap persamaan *gaussian kernel* (Fadel, et al., 2016). *Gaussian kernel* dan *laplacian kernel* dapat digunakan tanpa adanya pengetahuan tentang data yang akan digunakan (Karatzoglou, et al., 2006). *Gaussian kernel*, *laplacian kernel* dan *exponential kernel* juga terbukti memiliki akurasi yang baik pada metode SVM untuk pengenalan pola (Fadel, et al., 2016). Terdapat fungsi *kernel* lain yang masih termasuk dalam kategori RBF *kernel* yaitu ANOVA RBF *kernel* yang menggunakan ciri khas *squared exponential*. ANOVA RBF *kernel* memiliki notasi dasar yang sama dengan ANOVA RBF *kernel* (Berk, et al., 2014). Pada dasarnya ANOVA RBF singkatan dari *analysis of variance* sebagai teknik statistik untuk menganalisis interaksi berbagai macam atribut (Rasmussen & Williams, 2006). ANOVA RBF *kernel* dapat menyelesaikan permasalahan regresi dengan baik (Karatzoglou, et al., 2006). Fungsi ANOVA RBF *kernel* seperti pada persamaan (2.33).

$$K(x_i, x_j) = \left(\sum_{i,j=1}^N \exp(-\gamma(x_i - x_j)^2) \right)^d \quad (2.33)$$

Dimana,

$x_i - x_j$: vektor data

d : pangkat (*degree*)

γ : gamma

N : banyaknya data

Terdapat parameter $d > 0$ dan $\gamma \geq 0$ yang mempengaruhi kinerja ANOVA RBF *kernel*. Apabila parameter d bernilai 1 yang berarti pangkat 1 dan proses komputasi bersifat linier, sedangkan ketika parameter d bernilai 2 menjadi pangkat kuadrat bersifat non linier. Apabila parameter d bernilai 3 memiliki tiga menjadi kesamaan kubik (Berk, et al., 2014). Prinsip ANOVA RBF *kernel* pada parameter d memiliki kesamaan dengan parameter d pada *polynomial kernel*.

Pada penjelasan di atas sebelumnya merupakan beberapa contoh fungsi *kernel* secara umum sering diterapkan pada metode SVM. Pemilihan fungsi *kernel* dapat mempengaruhi kinerja pada algoritma yang digunakan (Sewell, 2007). Pada penelitian ini, menerapkan ANOVA RBF *kernel* untuk kasus regresi pada peramalan runtu waktu curah hujan. ANOVA RBF *kernel* terbukti dapat menyelesaikan permasalahan estimasi proyek pengembangan perangkat lunak dengan baik (Prabhakar & Dutta, 2013).

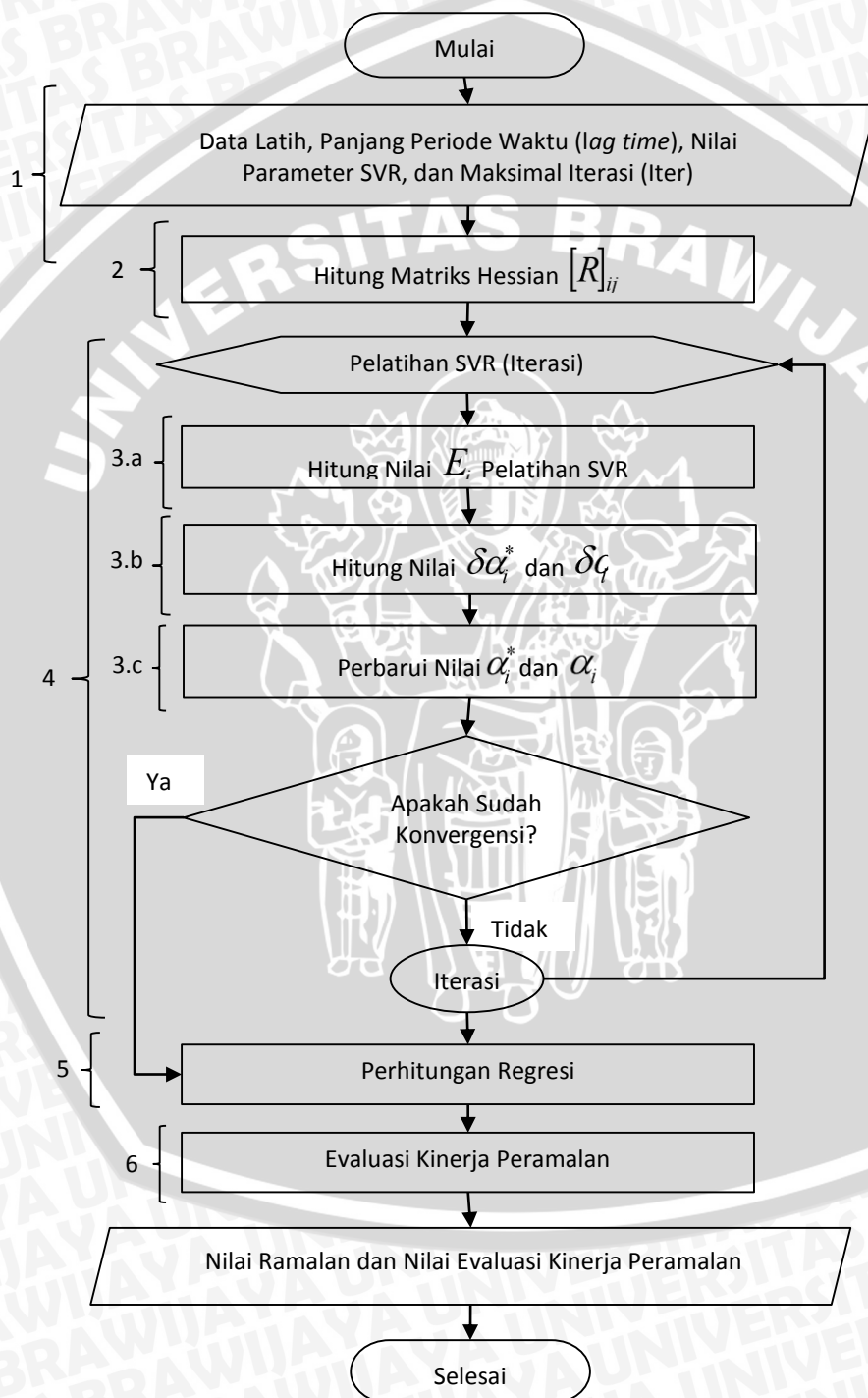
2.5.2 Algoritma Sekuensial SVR

Pada proses pelatihan metode *Support Vector Machine* (SVM) kasus klasifikasi dan regresi dapat diselesaikan dengan *Quadratic Programming* (QP). QP berfungsi untuk menemukan *hyperplane* atau garis pemisah yang optimal pada metode SVM. Namun, dalam penyelesaian oleh QP cukup kompleks dan banyak waktu (Cholissodin, et al., 2014). Meskipun hanya menggunakan dataset yang berjumlah tidak terlalu banyak, QP juga akan membatasi kinerja metode SVM seperti pada perhitungan perkalian matriks yang diproses dalam memori pada tahap pelatihan (Karatzoglou, et al., 2006).

Berdasarkan permasalahan QP dalam implementasi metode SVM, sehingga diperlukan algoritma alternatif untuk dapat melakukan komputasi cepat dan tidak terlalu kompleks pada pelatihan metode SVM (Laskov, et al., 2006). Algoritma alternatif pada proses pelatihan metode SVM yang telah ada seperti *Sequential Minimal Optimization* (SMO) (Platt, 1998), chunking (Osuna, et al., 1997) dan *simpleSVM* (Vishwanathan, et al., 2003). Meskipun telah ada algoritma alternatif untuk dapat diterapkan, namun tidak terlalu sering digunakan dalam implementasi pelatihan metode SVM (Loiacono, et al., 2007). Menurut Loiacono, et al., (2007) dalam review penelitian Laskov, et al., (2006) yang menganalisis algoritma alternatif pada pelatihan metode SVM, menjelaskan hal tersebut dikarenakan belum adanya standar dalam penggunaan algoritma tambahan serta agak rumit dan sulit untuk dapat diterapkan pada metode SVM dengan tujuan mendapatkan kinerja komputasi yang baik.

Pada penerapan metode SVM, Vijayakumar dan Wu (1999) mengusulkan algoritma sekuensial pada proses pelatihan metode SVM baik untuk kasus klasifikasi maupun regresi. Algoritma sekuensial merupakan langkah-langkah berupa skema sederhana untuk kemudahan dalam penerapan serta mengoptimalkan kinerja metode SVM. Algoritma sekuensial dikembangkan

sebagai solusi dalam optimasi dan alternatif QP (Vijayakumar & Wu, 1999). Pada penelitian ini, bertujuan untuk menyelesaikan kasus regresi menggunakan metode SVM yang lebih dikenal dengan *Support Vector Regression* (SVR). Pada Gambar 2.14 merupakan urutan langkah-langkah algoritma sekuensial untuk proses pelatihan metode SVR yang diusulkan oleh Vijayakumar dan Wu (1999) beserta beberapa proses tambahan.



Gambar 2.14 Diagram alir algoritma sekuensial SVR untuk proses pelatihan

Berikut merupakan penjelasan pada Gambar 2.14 berupa urutan langkah-langkah algoritma sekuensial untuk proses pelatihan metode SVR yang diusulkan oleh Vijayakumar dan Wu (1999) beserta beberapa proses tambahan.

1. Inisialisasi nilai awal $\alpha_i = \alpha_i^* = 0$ serta masukan parameter SVR yang akan digunakan seperti $\lambda = 0$, $C = 1000$, $\varepsilon = 0.0001$, $\gamma = 0.1$ (*learning rate SVR*), $\gamma = 1$ (*learning rate ANOVA RBF kernel*), $d = 1$ (kuadrat ANOVA RBF kernel) dan maksimum iterasi selama proses pelatihan metode SVR.
2. Selanjutnya menghitung matriks *Hessian* $[R]_{ij}$ pada persamaan (2.34).

$$[R]_{ij} = (K(x_i, x_j) + \lambda^2) \text{ untuk } i, j = 1, \dots, l \quad (2.34)$$

Dimana,

$[R]_{ij}$: matriks *Hessian*

K : *kernel trick*

(x_i, x_j) : data ke $i, j = 1, \dots, l$

λ^2 : variabel skalar

l : banyaknya data

3. Pada setiap iterasi proses pelatihan awal dari i sampai selesai l dilakukan perhitungan (3.a), (3.b) dan (3.c).
 - a) Perhitungan nilai kesalahan *error* proses pelatihan dengan menggunakan persamaan (2.35).

$$E_i = y_i - \sum_{j=1}^l (\alpha_i^* - \alpha_i) R_{ij} \quad (2.35)$$

Dimana,

E_i : nilai *error* pada pelatihan ke i

y_i : data aktual ke i

α_i dan α_i^* : dual variable alpha (*lagrangian multiplier*)

- b) Perhitungan nilai $\delta\alpha_i^*$ dan $\delta\alpha_i$ setelah didapatkan nilai E_i dengan menggunakan persamaan (2.36) dan (2.37).

$$\delta\alpha_i^* = \min \{ \max [\gamma(E_i - \varepsilon), -\alpha_i^*], C - \alpha_i^* \} \quad (2.36)$$

$$\delta\alpha_i = \min \{ \max [\gamma(-E_i - \varepsilon), -\alpha_i], C - \alpha_i \} \quad (2.37)$$

Dimana,

$\delta\alpha_i^*$ dan $\delta\alpha_i$: delta alpha yang bukan dari hasil perkalian δ dengan variabel

α_i dan α_i^*

γ : nilai *learning rate* (LR)

Pada nilai *learning rate* dengan batasan maksimal menggunakan perhitungan dari hasil pembagian nilai konstanta *learning rate* (cLR) dengan nilai maksimal secara diagonal dari matriks Hessian Rij pada persamaan (2.38). Menurut Vijayakumar dan Wu (1999) pada cLR dapat diinisialisasi awal dengan nilai 2 atau 1.9.

$$\gamma \leq \frac{\text{Konstanta } \textit{learning rate} \text{ (cLR)}}{\text{Maksimal(diagonal matriks hessian atau } [R]_{ii} \text{)}} \quad (2.38)$$

- c) Memperbarui nilai α_i^* dan α_i setelah langkah mendapatkan nilai $\delta\alpha_i^*$ dan $\delta\alpha_i$ dengan menggunakan persamaan (2.39) dan (2.40).

$$\alpha_i^* = \alpha_i^* + \delta\alpha_i^* \quad (2.39)$$

$$\alpha_i = \alpha_i + \delta\alpha_i \quad (2.40)$$

Dimana,

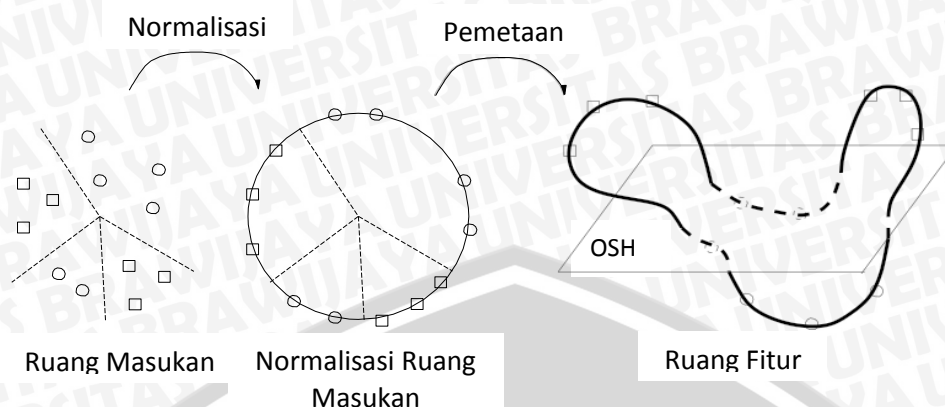
$\delta\alpha_i^*$ dan $\delta\alpha_i$: delta alpha yang bukan dari hasil perkalian $\mathcal{S}$ dengan variabel α_i dan α_i^* : dual variable alpha (*lagrangian multiplier*)

4. Ulangi langkah ketiga sampai pada batas maksimum iterasi atau telah mengalami konvergensi dengan syarat nilai $\text{maksimal}(|\delta\alpha_i^*|) < \varepsilon$ dan $\text{maksimal}(|\delta\alpha_i|) < \varepsilon$ selama proses pelatihan metode SVR.
5. Setelah mengalami konvergensi atau pemberhentian. Selanjutnya proses perhitungan regresi menggunakan persamaan (2.24), sehingga didapatkan hasil peramalan pada proses pelatihan.
6. Selanjutnya, melakukan evaluasi kinerja hasil peramalan pada proses pelatihan menggunakan persamaan (2.5).

2.5.3 Normalisasi Ruang Fitur SVR

Pada penggunaan metode SVR terdapat sebuah proses *preprocessing* data yang dapat mempengaruhi akurasi peramalan (Crone, et al., 2006a). Normalisasi data merupakan salah satu strategi dalam *transformation* data yang dapat diterapkan pada ruang *input* metode SVR untuk peramalan (Crone, et al., 2006a). Normalisasi pada ruang masukan telah dijelaskan sebelumnya pada pengolahan awal data runut waktu pada sub bab 2.3.3.

Pada dasarnya algoritma SVM ditujukan untuk menemukan *Optimal Separating Hyperplane* (OSH) terbaik didalam ruang fitur. Dengan melakukan pemetaan secara non linier pada ruang input yang dinormalisasi terdapat pengaruh dalam beberapa kasus sehingga menyebabkan hilangnya normalisasi pada skalanya seperti Gambar 2.15 (Graf & Borer, 2001).



Gambar 2.15 Normalisasi ruang input dan pemetaan ke ruang fitur

Sumber: Graf dan Borer (2001)

Permasalahan tersebut dapat diatasi dengan melakukan normalisasi di dalam ruang fitur sebagai alternatif normalisasi di dalam ruang masukan. Normalisasi di dalam ruang fitur sudah tidak termasuk ruang lingkup proses *preprocessing* data karena tidak diterapkan pada awal masukan data (Graf & Borer, 2001). Dengan mendefinisikan ulang fungsi *kernel* yang digunakan metode SVR menyerupai seperti konsep normalisasi di dalam ruang masukan. Dapat diasumsikan terdapat fungsi *kernel* $k(x, x')$ dan mendefinisikan fungsi *kernel* yang dapat melakukan normalisasi seperti persamaan (2.41) (Ben-Hur & Weston, 2010) atau persamaan (2.42) (Rasmussen & Williams, 2006).

$$\bar{K}(x, x') = \frac{K(x, x')}{\sqrt{K(x, x)K(x', x')}} \quad (2.41)$$

$$\bar{K}(x, x') = \frac{K(x, x')}{\sqrt{K(x, x)}\sqrt{K(x', x')}} \quad (2.42)$$

Dimana,

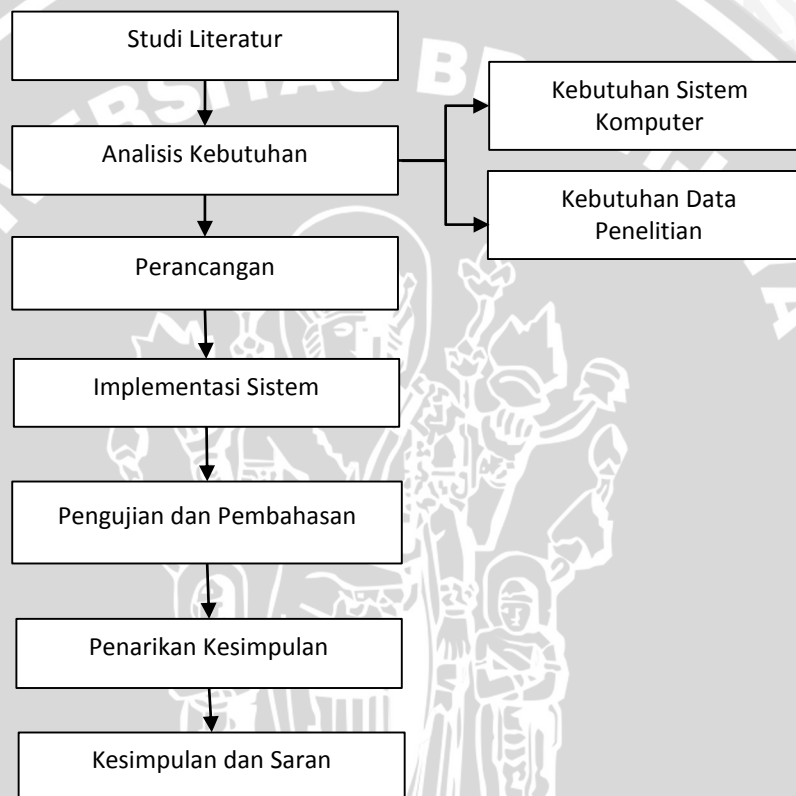
K : *kernel*

$x \cdot x'$: vektor data

Berdasarkan persamaan (2.41) atau (2.42) dapat diperoleh skala maksimal $\bar{k}(x, x')=1$ di dalam ruang fitur. Dalam mendefinisikan fungsi ANOVA RBF *kernel* pada persamaan (2.33) menggunakan saah satu definisi normalisasi pada ruang fitur persamaan (2.41), sehingga menjadi seperti pada persamaan (2.43).

BAB 3 METODOLOGI

Pada bab ini berisi tahapan dalam menyelesaikan penelitian yang berjudul peramalan runut waktu curah hujan menggunakan metode *Support Vector Regression* dan *Parallel Particle Swarm Optimization* (SVR-PPSO). Pada penelitian ini, termasuk dalam tipe penelitian perancangan (*design*) yang akan menghasilkan sebuah *prototype* berupa perangkat lunak untuk dapat melakukan peramalan runut waktu curah hujan. Berikut merupakan gambaran metodologi penelitian yang akan dilakukan dalam bentuk diagram alir ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram alir metodologi penelitian

3.1 Studi Literatur

Pada penelitian ini, memerlukan studi literatur dari dasar teori secara khusus telah dibahas pada BAB 2. Dasar teori diperoleh berdasarkan referensi yang diperoleh dari artikel, buku, jurnal, konferensi serta penelitian-penelitian terkait baik nasional maupun internasional dan bimbingan secara langsung dari dosen pembimbing dalam proses penelitian ini. Berikut merupakan dasar teori yang dibutuhkan sebagai pendukung penelitian ini antara lain:

1. Peramalan
2. Data runut waktu
3. Objek curah hujan

4. Metode *Support Vector Regression* (SVR) untuk peramalan
5. Metode *Parallel Particle Swarm Optimization* (PPSO) untuk optimasi

Studi literatur yang telah diperoleh akan digunakan sebagai pedoman pengetahuan dasar dalam melakukan analisis, perancangan, implementasi, dan pengujian dalam tahap-tahap penelitian ini.

3.2 Analisis Kebutuhan

Analisis kebutuhan merupakan langkah awal dalam penelitian yang harus tercukupi. Pada penelitian ini bertipe perancangan (*design*), sehingga analisis kebutuhan diperlukan untuk dapat mendukung seluruh tahap-tahap penelitian dalam perancangan, implementasi, dan pengujian objek dengan metode yang digunakan. Adapun kebutuhan yang harus terpenuhi dalam penelitian ini yaitu kebutuhan sistem komputer dan data penelitian.

3.2.1 Kebutuhan Sistem Komputer

Pada penelitian ini, peramalan runut waktu curah hujan menggunakan metode "SVR - PPSO" dibutuhkan sebuah sistem. Oleh karena itu, pencapaian kebutuhan sistem komputer haruslah mencukupi untuk mendukung berjalannya penelitian yang optimal. Terdapat kebutuhan sistem komputer baik perangkat keras dan perangkat lunak untuk dapat mendukung berjalannya penelitian yang akan dilakukan. Adapun sistem komputer yang digunakan terdiri dari perangkat keras dan perangkat lunak. Pada Tabel 3.1 merupakan spesifikasi perangkat keras, sedangkan pada Tabel 3.2 merupakan spesifikasi perangkat lunak yang digunakan penulis untuk memenuhi kebutuhan dalam penelitian ini.

Tabel 3.1 Spesifikasi perangkat keras personal komputer

Nama Komponen	Spesifikasi
<i>Motherboard</i>	ECS H61H2-M2 (V1.0)
<i>Processor</i>	Intel® Core™ i7-2600K (3.40 GHz, Cache 8 MB)
	<i>Heatsink CPU Cooler Fan</i> (Socket LGA 1155)
Memori (RAM)	VENOM RX DDR3 (2 × 2GB) PC10600 1333Mhz
<i>Power Supply</i>	PowerFul seri P-Ful5 600W Sata <i>Double Forward</i>
Media Penyimpanan (<i>Hard disk</i>)	Seagate Barracuda 7200.10 (160GB SATA)
Layar Monitor	17.0" LG Flatron E1600 <i>Display</i> (1360 x 768)
Mouse	Rexus Mouse Gaming USB RXM-X2 2400dpi
Keyboard	A4tech KM-720

Tabel 3.2 Spesifikasi perangkat lunak personal komputer

Nama Komponen	Spesifikasi
Sistem Operasi	Microsoft Windows 7 Ultimate 64-bit (6.1, Build 7600)
Editor Dokumentasi	Microsoft Office 2016
Editor Pemrograman	NetBeans IDE 8.1
Browser	Mozilla Firefox 48.0.2
Paket Pendukung Sistem	Driver Hardware
	Java Development Kit (JDK) 1.8.0

3.2.2 Kebutuhan Data Penelitian

Kebutuhan data diperlukan untuk mendukung objek penelitian dalam peramalan runtu waktu curah hujan. Data curah hujan yang digunakan merupakan data sekunder berdasarkan hasil pengamatan dari Badan Meteorologi Klimatologi dan Geofisika (BMKG) stasiun klimatologi Karangploso Malang.

Permohonan data ditujukan kepada pihak Badan Meteorologi Klimatologi dan Geofisika (BMKG) stasiun klimatologi Karangploso Malang. Dengan adanya permohonan data diharapkan untuk mendapatkan data sesuai kebutuhan pada penelitian ini. Berikut merupakan spesifikasi data yang dibutuhkan dalam penelitian ini.

1. Data curah hujan di wilayah Karangploso dan Poncokusumo Malang Jawa Timur, Indonesia dengan rentang waktu tahun 2000-2015.
2. Data curah hujan dengan rentang waktu per dasarian selama 10 hari, sehingga selama sebulan terdapat 3 dasarian.
3. Data curah hujan dengan satuan millimeter per dasarian.

3.3 Perancangan

Tahap perancangan bermaksud untuk merancang dalam menyelesaikan permasalahan. Dalam penyelesaian permasalahan pada penelitian ini dibutuhkan sebuah sistem berupa perangkat lunak berbasis *prototype*. Maka tujuan dari tahap perancangan dapat digunakan sebagai acuan dalam tahap implementasi dan pengujian. Berikut merupakan tahap perancangan secara umum yang akan dilakukan dan dibahas secara khusus pada bab 4 perancangan.

1. Formulasi strategi peramalan dilakukan dengan menyusun strategi peramalan dalam menyelesaikan permasalahan objek curah hujan. Dengan demikian, didapatkan solusi dari permasalahan penelitian ini serta menghasilkan langkah-langkah penyelesaian permasalahan peramalan objek curah hujan yang dapat diterapkan.

2. Formulasi metode SVR-PPSO merupakan penjabaran perhitungan metode secara manual, sehingga dapat diketahui alur proses metode berupa rancangan diagram alir algoritma yang dapat diterapkan.
3. Perancangan sistem berupa kontruksi dari sisi *back end* dan *front end* sistem yang akan diimplementasikan. Pada sisi *back end* sistem utama mengacu pada hasil rancangan diagram alir algoritma yang telah dijabarkan pada formulasi metode SVR-PPSO untuk dapat melakukan peramalan. Pada sisi *front end* sistem berupa berupa rancangan antarmuka pengguna untuk mempermudah pengguna dalam menggunakan sistem yang diimplementasikan.
4. Perancangan skenario pengujian terdiri dari pengujian metode dan data yang digunakan. Secara umum pengujian metode berupa uji coba metode SVR dan PPSO, serta uji variasi data runut waktu curah hujan yang digunakan. Dalam perancangan skenario pengujian ditujukan untuk mempermudah dalam tahap pengujian.

Selain ditujukan untuk mempermudah dalam implementasi sistem, tahap perancangan dapat mempermudah dalam tahap pengujian sistem. Pada tahap perancangan skenario pengujian akan diterapkan secara khusus pada bab pengujian.

3.4 Implementasi Sistem

Implementasi merupakan tahap lanjutan dari perancangan sistem peramalan runut waktu curah hujan menggunakan metode metode SVR-PPSO. Pada tahap implementasi melakukan penyesuaian dan mengacu dari hasil perancangan sistem yang telah dilakukan. Implementasi sistem akan dilakukan dan dibahas secara khusus pada 4.5.8 dengan beberapa batasan yang telah ditentukan. Implementasi program merupakan penerapan algoritma metode SVR-PPSO menggunakan bahasa pemrograman java. Pada proses implementasi diperlukan bantuan sistem komputer yang telah dijelaskan pada sub 3.2.1. Implementasi program dilakukan untuk menghasilkan sebuah sistem berbasis *prototype* yang dapat meramalkan data curah hujan. Setelah melakukan implementasi sistem, selanjutnya dapat dilakukan pengujian dan evaluasi untuk mengetahui kemampuan sistem dalam menyelesaikan permasalahan.

3.5 Pengujian dan Pembahasan

Pengujian dan pembahasan dilakukan terhadap sistem yang telah diimplementasikan bertujuan untuk mengetahui kemampuan sistem dalam meramalkan curah hujan. Terdapat pengujian metode SVR-PPSO yang diterapkan dalam sistem dan penggunaan data runut waktu curah hujan sebagai objek sistem. Pengujian dan pembahasan akan dilakukan secara khusus pada BAB 6 menggunakan hasil perancangan skenario pengujian. Uji coba dilakukan berdasarkan hasil perancangan skenario pengujian. Dalam melakukan uji coba mengacu pada hasil nilai keluaran dari sistem menggunakan rumus statistika yaitu *Root Mean Square Error* (RMSE) dan *fitness*. *Fitness* ditunjukkan sebagai konfigurasi optimal dengan mengacu terhadap nilai *error* yang dihasilkan. Dengan adanya

tahap pengujian dan pembahasan, sehingga mempermudah untuk menarik kesimpulan terhadap penyelesaian masalah peramalan runut waktu curah hujan menggunakan metode SVR-PPSO.

3.6 Penarikan Kesimpulan

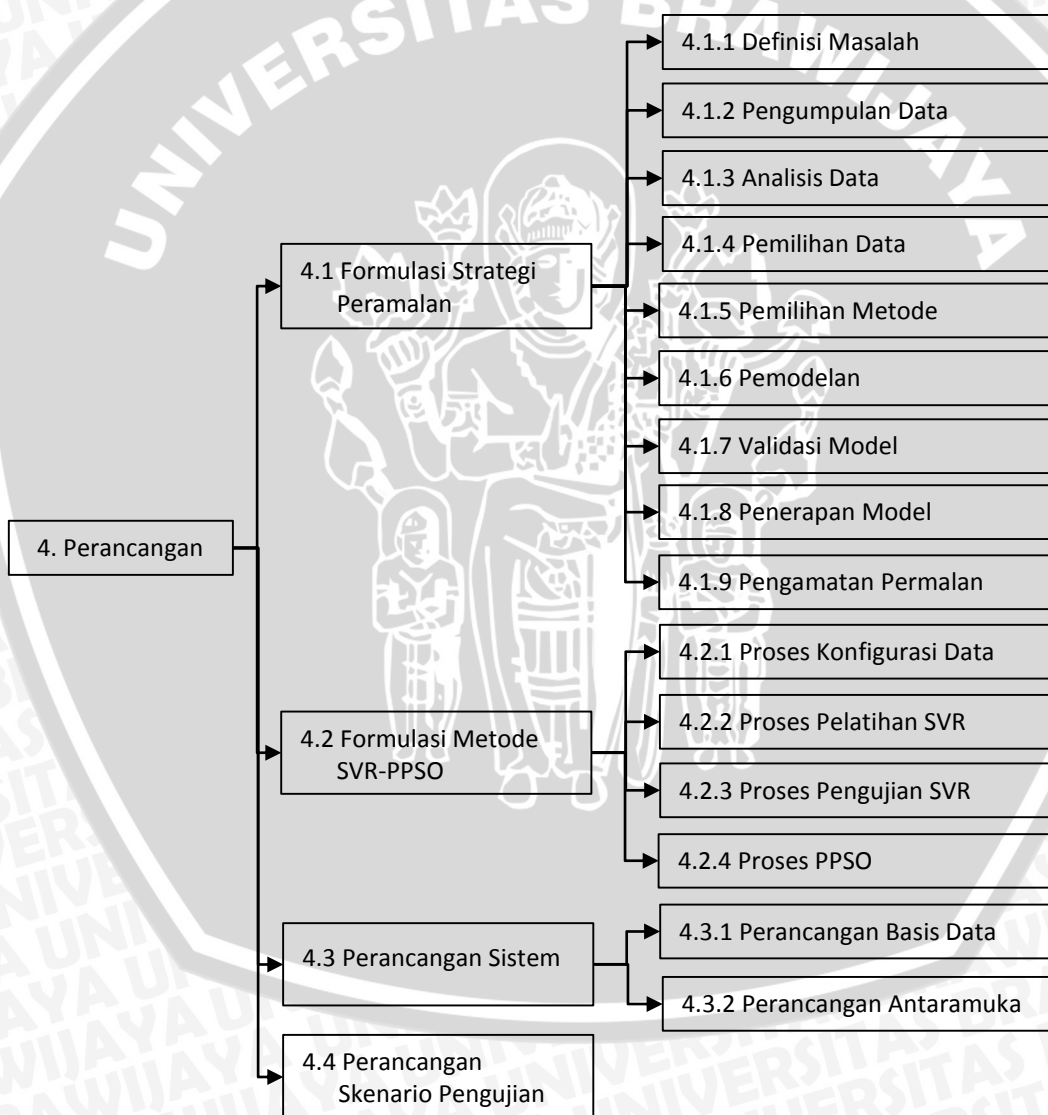
Tahap penelitian dilakukan mulai dari studi literatur sampai pengujian dan pembahasan. Apabila seluruh tahap tersebut telah dilakukan, sehingga dapat diperoleh kesimpulan berdasarkan hasil pengujian yang dilakukan terhadap sistem peramalan runut waktu curah hujan menggunakan metode SVR-PPSO berbasis prototype. Kesimpulan dapat diperoleh secara valid berdasarkan bukti hasil pengujian yang dilakukan. Selanjutnya, dapat dilakukan penarikan kesimpulan untuk menjawab rumusan masalah yang telah dirumuskan.

Selain penarikan kesimpulan, terdapat masukan dari penulis atau saran dalam melakukan penelitian ini seperti kekurangan maupun kesalahan yang terjadi. Berdasarkan saran tersebut, dapat dilakukan pengembangan ataupun perbaikan pada penelitian selanjutnya. Perbaikan dari metode SVR-PPSO maupun objek curah hujan yang diangkat. Dengan harapan kedepan penelitian selanjutnya dapat melakukan penelitian yang lebih baik dari penelitian sebelumnya.



BAB 4 PERANCANGAN

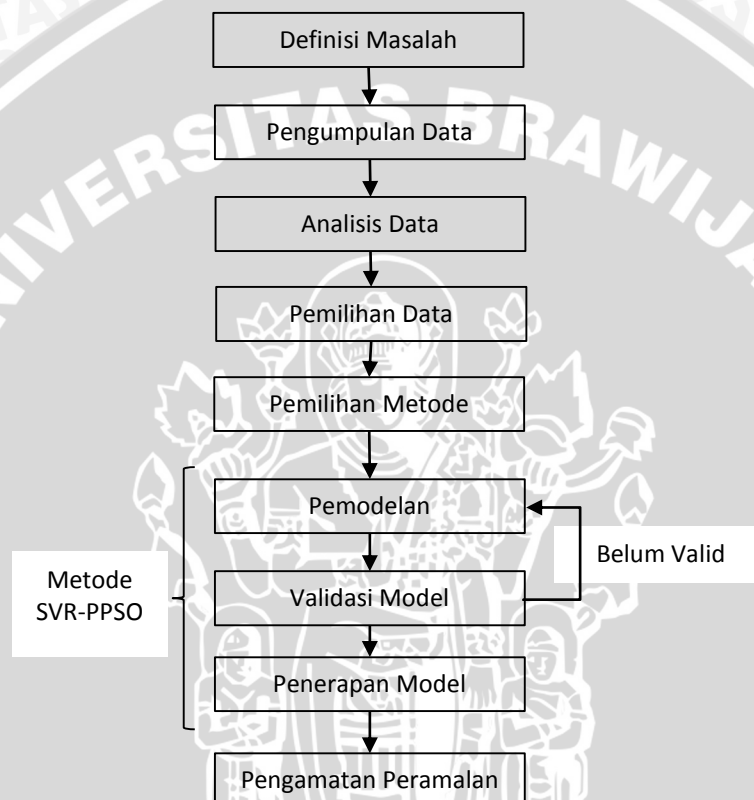
Pada bab ini merupakan tahap perancangan (*design*) yang menghasilkan rancangan sistem berupa perangkat lunak yang dapat melakukan peramalan runtu waktu curah hujan menggunakan metode *Support Vector Regression* dan *Parallel Particle Swarm Optimization* (SVR-PPSO). Pada tahap perancangan diperlukan formulasi strategi permasalahan, perancangan sistem dan implementasi sistem. Perancangan skenario pengujian juga dilakukan untuk memudahkan peneliti dalam mengevaluasi kinerja peramalan metode SVR-PPSO. Berikut merupakan gambaran secara umum tahap perancangan yang akan dilakukan dalam bentuk diagram struktur yang ditunjukkan pada Gambar 4.1.



Gambar 4.1 Struktur perancangan

4.1 Formulasi Strategi Peramalan

Formulasi permasalahan diperlukan untuk dapat memenuhi tujuan pada penelitian yang dilakukan dengan baik. Formulasi permasalahan pada penelitian ini berupa formulasi strategi peramalan dengan bentuk urutan proses peramalan yang tersusun. Formulasi strategi peramalan dilakukan berdasarkan dasar teori terkait dengan proses peramalan. Dalam memformulasikan strategi peramalan ini mengacu pada rumusan masalah serta membatasi masalah yang akan diselesaikan. Pada Gambar 4.2 merupakan strategi peramalan yang akan diterapkan pada penelitian ini.



Gambar 4.2 Strategi peramalan

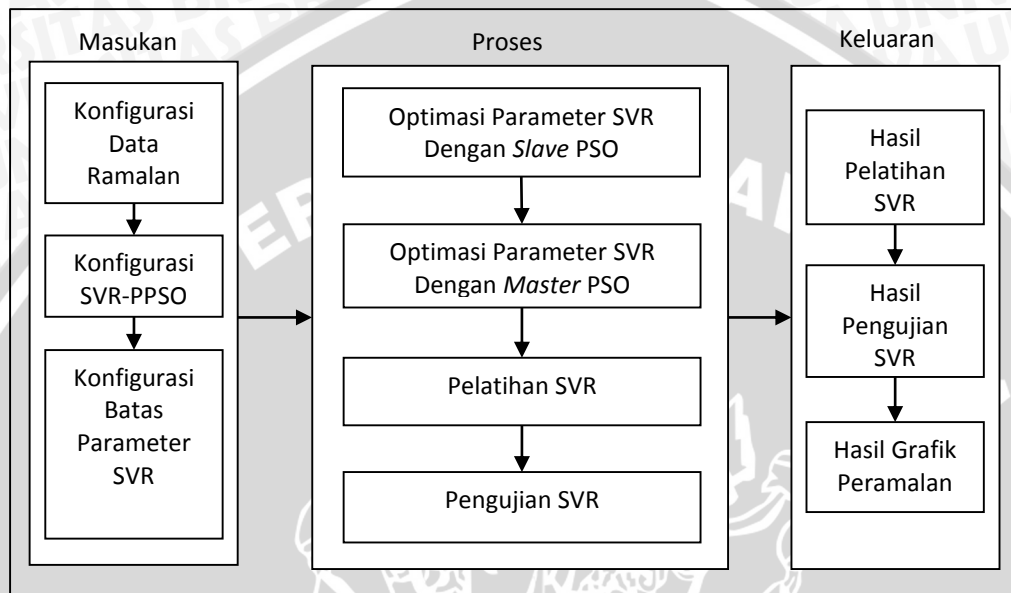
Sumber: Abraham dan Ledolter (2005), Armstrong (2001), dan Montgomery, et al. (2015)

4.1.1 Definisi Masalah

Pada sub bab 1.1 secara umum telah dibahas mengenai pentingnya unsur curah hujan yang dapat mempengaruhi kehidupan. Curah hujan yang terjadi pada waktu yang berbeda dapat dikategorikan dalam musim hujan dan musim kering. Beberapa dampak positif dan negatif terjadinya curah hujan di musim tertentu yang telah dibahas pada sub bab 2.4.1. Berdasarkan hal tersebut, sehingga diperlukan peramalan runut waktu curah hujan untuk dapat menghindari terjadinya dampak negatif.

4.3 Perancangan Sistem

Pada penelitian ini, diperlukan perancangan sebuah sistem yang menerapkan metode SVR-PPSO untuk melakukan peramalan runut waktu curah hujan. Secara umum sistem memiliki tahap masukan, proses, dan keluaran yang dapat berjalan dengan adanya pengguna. Dengan melakukan perancangan sistem, sehingga memudahkan dalam implementasi. Pada Gambar 4.71 merupakan model alur sistem berdasarkan formulasi strategi peramalan pada sub bab 4.1 dan formulasi metode SVR-PPSO pada sub bab 4.2 yang telah dijelaskan sebelumnya.



Gambar 4.71 Model alur sistem peramalan runut waktu curah hujan menggunakan metode SVR-PPSO

Pada Gambar 4.71 merupakan model alur sistem peramalan runut waktu curah hujan menggunakan metode SVR-PPSO. Pada alur sistem terdapat tahapan yang terdiri dari masukan, proses, dan keluaran. Berikut merupakan penjelasan Gambar 4.71 berdasarkan tahapan alur sistem.

1. Masukan merupakan tahap awal sebelum proses yang memerlukan masukan dari pengguna. Masukan dari pengguna berupa masukan konfigurasi data ramalan, konfigurasi metode SVR-PPSO, dan konfigurasi batas parameter SVR. Pada tahap masukan ini, pengguna tidak memerlukan memasukkan data ramalan karena data runut waktu curah hujan tersimpan dalam basis data.
2. Proses merupakan tahap utama yang menerapkan strategi peramalan menggunakan metode SVR-PPSO. Terdapat proses optimasi parameter SVR menggunakan strategi paralel PSO yaitu *slave* PSO dan *master* PSO pada proses pelatihan SVR. Setelah didapatkan parameter terbaik, dilakukan pelatihan SVR kembali dan dilanjutkan dengan pengujian SVR sebagai peramalan runut waktu curah hujan.
3. Keluaran merupakan tahap akhir berupa hasil dari tahap proses sebelumnya. Hasil akhir berupa teks dan gambar yang merepresentasikan hasil peramalan runut waktu curah hujan menggunakan metode SVR-PPSO. Hasil utama dalam

bentuk teks berupa nilai evaluasi kinerja peramalan, sedangkan dalam bentuk gambar berupa grafik perbandingan hasil ramalan dengan nilai aktual pada data runut waktu curah hujan.

Berdasarkan model alur sistem pada Gambar 4.71, sehingga diperlukan yang sebuah antarmuka pengguna untuk dapat berinteraksi dengan sistem yang dibahas secara khusus pada sub bab 4.3.2. Selain itu juga diperlukan perancangan basis data untuk menyimpan data runut waktu curah hujan yang dibahas secara khusus pada sub bab 4.3.1.

4.3.1 Perancangan Basis Data

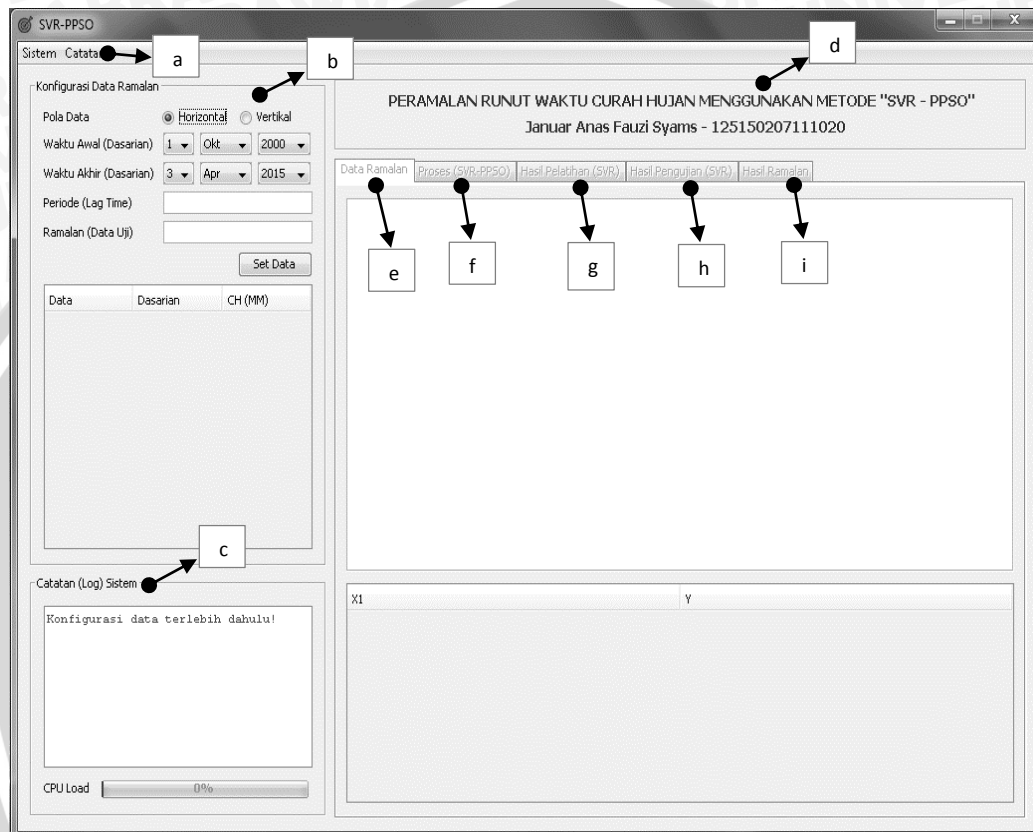
Perancangan basis data pada sistem yang akan dibuat menggunakan *SQLite* dan hanya ada satu tabel data runut waktu curah hujan. Pada tabel tersebut terdapat 5 kolom dengan isi data yang berbeda. Pada Tabel 4.43 berikut merupakan perancangan basis data.

Tabel 4.43 Perancangan basis data

No	Nama Kolom	Tipe Kolom	Keterangan
1	rowid	Integer	Nomor urutan data
2	dasarian	Integer	Waktu dasarian
3	bulan	Varchar	Waktu bulan
4	tahun	Integer	Waktu tahun
5	ch	Double	Curah Hujan (MM)

4.3.3 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna pada sistem dilakukan secara langsung di perangkat lunak NetBeans IDE menggunakan bahasa pemrograman java. Pada perancangan halaman antarmuka pengguna secara menyeluruh menjadi satu halaman. Perancangan halaman tersebut ditujukan untuk mempermudah pengguna dalam menggunakan sistem. Pada Gambar 4.72 merupakan hasil rancangan tampilan awal sistem.

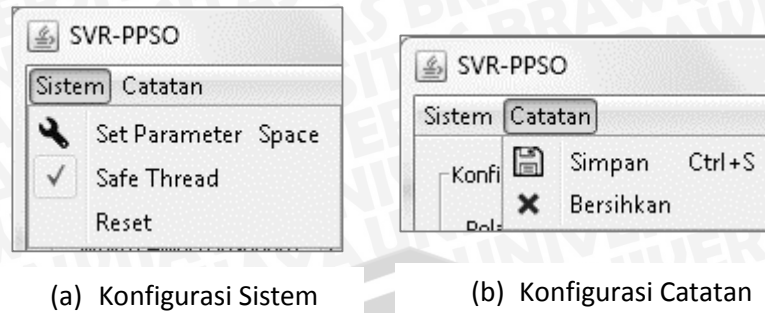


Gambar 4.72 Tampilan awal sistem

Pada Gambar 4.72 merupakan tampilan awal sistem yang terdiri dari daftar pilihan atau opsi konfigurasi sistem, panel konfigurasi data, panel catatan (log) sistem, panel judul penelitian, panel tab data ramalan, panel tab proses SVR-PPSO, panel tab hasil pelatihan SVR, panel tab hasil pengujian SVR, dan panel tab hasil ramalan. Pada tampilan antarmuka daftar pilihan konfigurasi sistem dan panel akan dibahas komponen yang digunakan secara detail pada sub bab selanjutnya.

4.3.3.1 Rancangan Daftar Pilihan Konfigurasi Sistem

Pada Gambar 4.72a merupakan tampilan daftar pilihan atau opsi konfigurasi sistem menggunakan komponen dasar JMenuBar. Pada tampilan daftar pilihan konfigurasi sistem berfungsi untuk mengatur kerja sistem sebelum proses berjalan dan manajemen catatan hasil kerja sistem. Pada Gambar 4.73 merupakan detail tampilan daftar pilihan konfigurasi sistem ketika dipilih oleh pengguna.



Gambar 4.73 Detail tampilan daftar pilihan konfigurasi sistem

Pada Gambar 4.73a merupakan tampilan daftar pilihan dalam menkonfigurasi sistem. Selanjutnya, pada Gambar 4.73b merupakan pilhan dalam manajemen catatan hasil kerja sistem pada Gambar 4.73b. Pada Tabel 4.44 merupakan keterangan komponen yang digunakan untuk tampilan daftar pilihan konfigurasi sistem.

Tabel 4.44 Keterangan komponen daftar pilihan konfigurasi sistem

Type Komponen	Fungsi Komponen
JMenuBar	Menampilkan daftar pilihan konfigurasi sistem.
JMenu	Menampilkan pilihan konfigurasi sistem.
	Menampilkan pilihan dalam manajemen catatan hasil kerja sistem.
JMenuItem	Mengatur inisialisasi contoh parameter awal pada konfigurasi metode SVR-PPSO.
	Reset sistem seperti awal.
	Menyimpan catatan hasil kerja sistem.
	Menghapus catatan hasil kerja sistem.
JCheckBoxMenuItem	Mengatur sistem pada mode <i>safe thread</i> atau tidak.

4.3.3.2 Rancangan Panel Konfigurasi Data Ramalan

Pada Gambar 4.72b merupakan tampilan panel konfigurasi data ramalan menggunakan komponen dasar JPanel. Pada tampilan panel konfigurasi data ramalan berfungsi untuk pengolahan awal data runut waktu sebelum proses peramalan. Pada Tabel 4.45 dan Tabel 4.46 merupakan keterangan komponen yang digunakan untuk tampilan panel konfigurasi data ramalan.

Tabel 4.45 Keterangan komponen panel konfigurasi data ramalan

Type Komponen	Fungsi Komponen
JPanel	Menampilkan panel konfigurasi data ramalan.
JLabel	Menampilkan label teks pola data.
	Menampilkan label teks waktu awal (dasarian).
	Menampilkan label teks waktu akhir (dasarian).
	Menampilkan label teks periode (lag time).
	Menampilkan label teks ramalan (data uji).
JRadioButton	Masukan pilihan pola data secara horizontal.

Tabel 4.46 Keterangan komponen panel konfigurasi data ramalan (lanjutan)

Type Komponen	Fungsi Komponen
JRadioButton	Masukan pilihan pola data secara vertikal.
JComboBox	Masukan dasarian pada waktu awal (dasarian).
	Masukan bulan pada waktu awal (dasarian).
	Masukan tahun pada waktu awal (dasarian).
	Masukan dasarian pada waktu akhir (dasarian).
	Masukan bulan pada waktu akhir (dasarian).
JFormattedTextField	Masukan periode (<i>lag time</i>).
	Masukan ramalan (data uji).
JButton	Proses konfigurasi data ramalan.
JTable	Menampilkan data curah hujan berdasarkan runut waktu yang telah konfigurasi.

4.3.3.3 Rancangan Panel Catatan Sistem

Pada Gambar 4.72c merupakan tampilan panel catatan sistem menggunakan komponen dasar JPanel. Pada tampilan panel catatan sistem berfungsi untuk menampilkan catatan sistem selama proses penggunaan sistem. Pada Tabel 4.47 merupakan keterangan komponen yang digunakan untuk tampilan panel catatan sistem.

Tabel 4.47 Keterangan komponen panel konfigurasi data ramalan

Type Komponen	Fungsi Komponen
JPanel	Menampilkan panel catatan sistem.
JTextArea	Menampilkan hasil catatan sistem.
JLabel	Menampilkan label teks CPU <i>load</i> .
JProgressBar	Menampilkan presentase CPU <i>load</i> .

4.3.3.4 Rancangan Panel Judul Penelitian

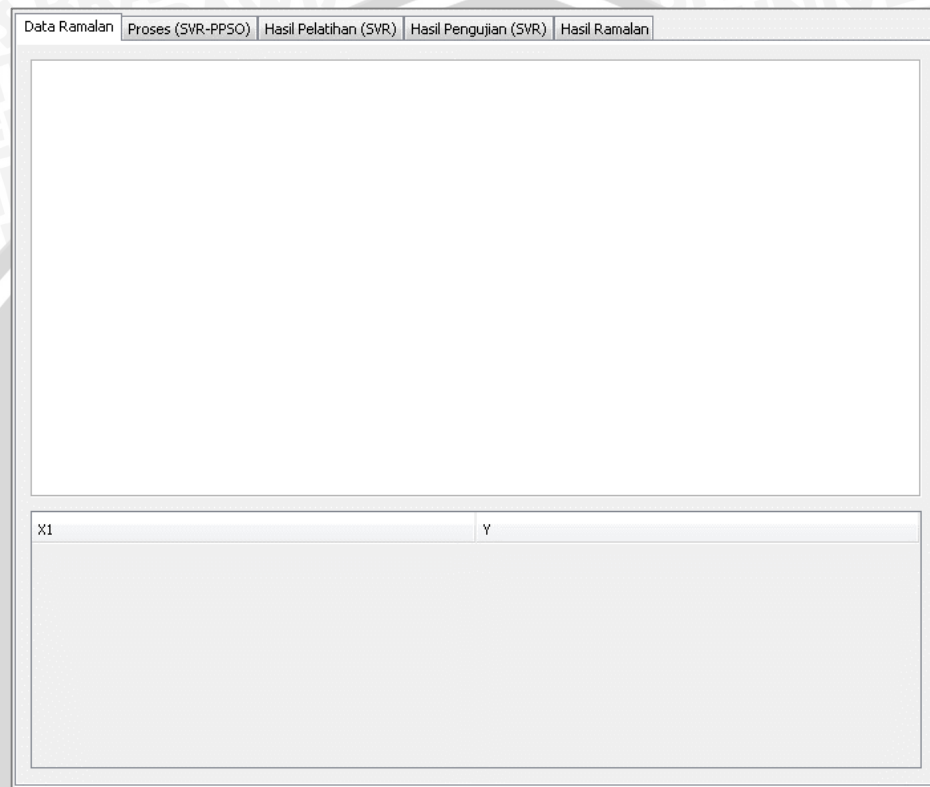
Pada Gambar 4.72d merupakan tampilan panel judul penelitian menggunakan komponen dasar JPanel. Pada tampilan panel judul penelitian berfungsi untuk menampilkan judul penelitian ini. Pada Tabel 4.48 merupakan keterangan komponen yang digunakan untuk tampilan panel judul penelitian.

Tabel 4.48 Keterangan komponen panel konfigurasi data ramalan

Type Komponen	Fungsi Komponen
JPanel	Menampilkan panel judul penelitian.
JLabel	Menampilkan label teks judul penelitian ini.
	Menampilkan label teks nama dan nim penulis dalam penelitian ini.

4.3.3.5 Rancangan Panel Tab Data Ramalan

Pada Gambar 4.72e merupakan tampilan panel tab data ramalan yang berisikan komponen dasar JPanel. Pada tampilan panel tab data ramalan berfungsi untuk menampilkan grafik data latih dan data uji berdasarkan konfigurasi data yang telah dilakukan oleh pengguna, serta menampilkan model regresi data peramalan. Pada Gambar 4.74 merupakan detail tampilan panel tab data ramalan sebelum data dikonfigurasi oleh pengguna.



Gambar 4.74 Detail tampilan panel tab data ramalan

Pada Gambar 4.74 detail tampilan panel tab data ramalan terdapat JPanel, JTable, dan beberapa komponen antarmuka tambahan yang diterapkan. Komponen tambahan berguna untuk menampilkan grafik menggunakan library JfreeChart versi 1.0.19. Pada Tabel 4.68 merupakan keterangan komponen yang digunakan untuk tampilan panel tab data ramalan.

Tabel 4.49 Keterangan komponen panel tab data ramalan

Type Komponen	Fungsi Komponen
JPanel	Menampilkan grafik konfigurasi data latih dan data uji sebelum proses peramalan.
JTable	Menampilkan model regresi data peramalan yang telah dikonfigurasi oleh pengguna.

4.3.3.6 Rancangan Panel Tab Proses SVR-PPSO

Pada Gambar 4.72f merupakan tampilan panel tab proses SVR-PPSO yang berisikan komponen dasar JPanel. Pada tampilan panel tab proses SVR-PPSO berfungsi untuk melakukan konfigurasi metode SVR-PPSO dalam proses optimasi parameter SVR dan peramalan. Pada Gambar 4.75 merupakan detail tampilan panel tab proses SVR-PPSO yang memerlukan masukan dari pengguna.

Gambar 4.75 Detail tampilan panel tab proses SVR-PPSO

Pada Gambar 4.75 detail tampilan panel tab proses SVR-PPSO terdapat 2 panel dan panel tab. Panel tab yang ada didalam tampilan panel tab proses SVR-PPSO bertujuan untuk menampilkan partikel dari metode PSO. Pada Tabel 4.50 dan Tabel 4.51 merupakan keterangan komponen yang digunakan untuk tampilan panel tab proses SVR-PPSO.

Tabel 4.50 Keterangan komponen panel tab proses SVR-PPSO

Type Komponen	Fungsi Komponen
JPanel	Menampilkan konfigurasi SVR-PPSO
	Menampilkan konfigurasi Batas Parameter SVR.
JLabel	Menampilkan label teks jumlah slave PSO.
	Menampilkan label teks jumlah partikel PSO.
	Menampilkan label teks maksimal iterasi PSO.
	Menampilkan label teks maksimal iterasi SVR.
	Menampilkan label teks target fitness.

Tabel 4.51 Keterangan komponen panel tab proses SVR-PPSO (lanjutan)

Type Komponen	Fungsi Komponen
JLabel	Menampilkan label teks batas minimal.
	Menampilkan label teks batas maksimal.
	Menampilkan label teks Gamma (γ).
	Menampilkan label teks Lambda (λ).
	Menampilkan label teks Complexity (c).
	Menampilkan label teks Epsilon (ϵ).
	Menampilkan label teks Gamma (γ) <i>kernel</i> .
	Menampilkan label teks Degree (d) <i>kernel</i> .
JFormattedTextField	Masukan jumlah slave PSO.
	Masukan jumlah partikel PSO.
	Masukan maksimal iterasi PSO.
	Masukan maksimal iterasi SVR.
	Masukan target fitness.
	Masukan batas minimal Gamma (γ).
	Masukan batas minimal Lambda (λ).
	Masukan batas minimal Complexity (c).
	Masukan batas minimal Epsilon (ϵ).
	Masukan batas minimal Gamma (γ) <i>kernel</i> .
	Masukan batas minimal Degree (d) <i>kernel</i> .
	Masukan batas maksimal Gamma (γ).
	Masukan batas maksimal Lambda (λ).
	Masukan batas maksimal Complexity (c).
	Masukan batas maksimal Epsilon ϵ .
	Masukan batas maksimal Gamma (γ) <i>kernel</i> .
	Masukan batas maksimal Degree (d) <i>kernel</i> .
JRadioButton	Masukan pilihan target <i>fitness</i> pada hasil pelatihan (SVR).
	Masukan pilihan target <i>fitness</i> pada hasil pengujian (SVR).
JButton	Proses metode PPSO untuk optimasi parameter SVR.
	Proses pemberhentian metode PPSO dalam optimasi parameter SVR.
JTabbedPane	Menampilkan partikel metode PPSO.
JTable	Menampilkan partikel awal <i>slave</i> PSO.
	Menampilkan slave partikel <i>slave</i> PSO.
	Menampilkan elite partikel <i>slave</i> PSO.
	Menampilkan master partikel PSO.

4.3.3.7 Rancangan Panel Tab Hasil Pelatihan SVR

Pada Gambar 4.72g merupakan tampilan panel tab hasil pelatihan SVR yang berisikan komponen dasar JPanel. Pada tampilan panel tab hasil pelatihan SVR berfungsi untuk menampilkan hasil ramalan terhadap data latih menggunakan

metode SVR. Pada Gambar 4.77 dan Gambar 4.77 merupakan detail tampilan panel tab hasil pelatihan SVR setelah proses SVR-PPSO.

Gambar 4.76 Detail Tampilan panel tab hasil pelatihan SVR dengan pratinjau panel tab hasil ramalan

Gambar 4.77 Detail tampilan panel tab hasil pelatihan SVR dengan pratinjau panel tab nilai α_i^* dan α_i

Pada Gambar 4.77 dan Gambar 4.77 detail tampilan panel tab hasil pelatihan SVR terdapat 2 panel dan panel tab. Panel tab yang ada didalam tampilan panel tab hasil pelatihan SVR bertujuan untuk menampilkan hasil ramalan pelatihan pada Gambar 4.77 dan nilai α_i^* dan α_i pada Gambar 4.77. Pada Tabel 4.52 dan Tabel 4.53 merupakan keterangan komponen yang digunakan untuk tampilan panel tab hasil pelatihan SVR.

Tabel 4.52 Keterangan komponen panel tab hasil pelatihan SVR

Type Komponen	Fungsi Komponen
JPanel	Menampilkan parameter SVR (pelatihan).
	Menampilkan evaluasi kinerja peramalan (pelatihan).
JLabel	Masukan label teks Gamma (γ).
	Masukan label teks Lambda (λ).
	Masukan label teks kompleksitas (c).

Tabel 4.53 Keterangan komponen panel tab hasil pelatihan SVR (lanjutan)

Type Komponen	Fungsi Komponen
	Masukan label teks Epsilon (ϵ).
	Masukan label teks Gamma (γ) <i>kernel</i> .
	Masukan label teks Degree (d) <i>kernel</i> .
	Menampilkan label teks nilai RMSE.
JFormattedTextField	Menampilkan Gamma (γ).
	Menampilkan Lambda (λ).
	Menampilkan kompleksitas (c).
	Menampilkan Epsilon (ϵ).
	Menampilkan Gamma (γ) <i>kernel</i> .
	Menampilkan Degree (d) <i>kernel</i> .
	Menampilkan nilai RMSE.
	Menampilkan nilai fitness.
	Menampilkan γ .
JTabbedPane	Menampilkan hasil pelatihan SVR.
JTable	Menampilkan hasil ramalan pelatihan SVR.
	Menampilkan nilai α_i^* dan α_i
JPanel	Menampilkan parameter SVR (pelatihan).
	Menampilkan evaluasi kinerja peramalan (pelatihan).
JLabel	Masukan label teks Gamma (γ).
	Masukan label teks Lambda (λ).
	Masukan label teks kompleksitas (c).
	Masukan label teks Epsilon (ϵ).
	Masukan label teks Gamma (γ) <i>kernel</i> .
	Masukan label teks Degree (d) <i>kernel</i> .
	Menampilkan label teks nilai RMSE.
JFormattedTextField	Menampilkan Gamma (γ).
	Menampilkan Lambda (λ).
	Menampilkan kompleksitas (c).
	Menampilkan Epsilon (ϵ).
	Menampilkan Gamma (γ) <i>kernel</i> .
	Menampilkan Degree (d) <i>kernel</i> .
	Menampilkan nilai RMSE.
	Menampilkan nilai fitness.
	Menampilkan γ .
JTabbedPane	Menampilkan hasil pelatihan SVR.
JTable	Menampilkan hasil ramalan pelatihan SVR.
	Menampilkan nilai α_i^* dan α_i

4.3.3.8 Rancangan Panel Tab Hasil Pengujian SVR

Pada Gambar 4.72h merupakan tampilan panel tab hasil pengujian SVR yang berisikan komponen dasar JPanel. Pada tampilan panel tab hasil pengujian SVR berfungsi untuk menampilkan hasil ramalan terhadap data uji menggunakan metode SVR. Pada Gambar 4.78 dan Gambar 4.79 merupakan detail tampilan panel tab hasil pengujian SVR setelah proses SVR-PPSO.

The screenshot shows a software interface with five tabs: 'Data Ramalan', 'Proses (SVR-PPSO)', 'Hasil Pelatihan (SVR)', 'Hasil Pengujian (SVR)', and 'Hasil Ramalan'. The 'Hasil Pelatihan (SVR)' tab is active. It contains two main sections: 'Parameter SVR dan Kernel (Pengujian)' and 'Evaluasi Kinerja Peramalan (Pengujian)'. The first section has input fields for Gamma (γ), Lambda (λ), Kompleksitas (C), Epsilon (ϵ), Gamma (γ) kernel, and Degree (d) kernel. The second section has input fields for Nilai RMSE and Nilai Fitness. On the right, there is a table titled 'Hasil Ramalan (Pengujian)' with a sub-tab 'Konversi CH Bulanan'. The table has columns: 'Data', 'Dasarian', 'Data Aktual (Uji)', and 'Hasil Ramalan (...)'. The table is currently empty.

Gambar 4.78 Detail tampilan panel tab hasil pelatihan SVR dengan pratinjau panel tab hasil ramalan

The screenshot shows the same software interface as Gambar 4.78, but with the 'Hasil Pelatihan (SVR)' tab active. The 'Parameter SVR dan Kernel (Pengujian)' and 'Evaluasi Kinerja Peramalan (Pengujian)' sections are identical. The 'Hasil Ramalan (Pengujian)' sub-tab is now 'Konversi CH Bulanan'. The table has columns: 'Bulan', 'Aktual', 'Intensitas', 'Ramalan', and 'Intensitas'. The table is currently empty.

Gambar 4.79 Detail tampilan panel tab hasil pelatihan SVR dengan pratinjau panel tab hasil konversi curah hujan bulanan

Pada Gambar 4.78 dan Gambar 4.79 detail tampilan panel tab hasil pengujian SVR terdapat 2 panel dan panel tab. Panel tab yang ada didalam tampilan panel tab hasil pengujian SVR bertujuan untuk menampilkan hasil ramalan pengujian pada Gambar 4.78 dan hasil konversi curah hujan bulanan pada Gambar 4.79. Pada Tabel 4.54 dan Tabel 4.55 merupakan keterangan komponen yang digunakan untuk tampilan panel tab hasil pelatihan SVR.

Tabel 4.54 Keterangan komponen panel tab hasil pengujian SVR

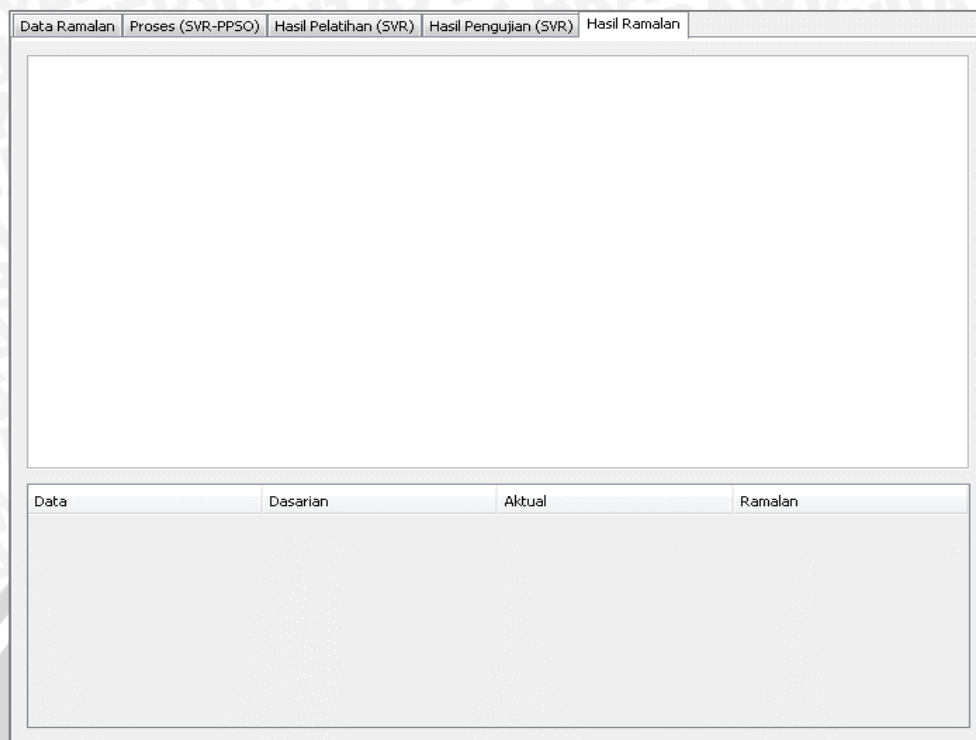
Type Komponen	Fungsi Komponen
JPanel	Menampilkan parameter SVR (pengujian).
	Menampilkan evaluasi kinerja peramalan (pengujian).

Tabel 4.55 Keterangan komponen panel tab hasil pengujian SVR (lanjutan)

Type Komponen	Fungsi Komponen
JLabel	Masukan label teks Gamma (γ).
	Masukan label teks Lambda (λ).
	Masukan label teks kompleksitas (c).
	Masukan label teks Epsilon (ϵ).
	Masukan label teks Gamma (γ) <i>kernel</i> .
	Masukan label teks Degree (d) <i>kernel</i> .
	Menampilkan label teks nilai RMSE.
JFormattedTextField	Menampilkan Gamma (γ).
	Menampilkan Lambda (λ).
	Menampilkan kompleksitas (c).
	Menampilkan Epsilon (ϵ).
	Menampilkan Gamma (γ) <i>kernel</i> .
	Menampilkan Degree (d) <i>kernel</i> .
	Menampilkan nilai RMSE.
	Menampilkan nilai fitness.
JTabbedPane	Menampilkan hasil pelatihan SVR.
JTable	Menampilkan hasil ramalan pengujian SVR.
	Menampilkan hasil konversi curah hujan bulanan.

4.3.3.9 Rancangan Panel Tab Hasil Ramalan

Pada Gambar 4.72i merupakan tampilan panel tab hasil ramalan yang berisikan komponen dasar JPanel. Pada tampilan panel tab hasil ramalan berfungsi untuk menampilkan grafik hasil ramalan data latih dan data uji berdasarkan konfigurasi data yang telah dilakukan oleh pengguna, serta menampilkan perbandingan hasil ramalan dengan nilai aktual. Pada Gambar 4.80 merupakan detail tampilan panel tab hasil ramalan sebelum dilakukan proses peramalan menggunakan metode SVR-PPSO.



Gambar 4.80 Detail tampilan panel tab hasil ramalan

Pada Gambar 4.80 detail tampilan panel tab hasil ramalan terdapat JPanel, JTable, dan beberapa komponen antarmuka tambahan yang diterapkan. Komponen tambahan berguna untuk menampilkan grafik menggunakan library JfreeChart versi 1.0.19. Pada Tabel 4.56 merupakan keterangan komponen yang digunakan untuk tampilan panel tab hasil ramalan.

Tabel 4.56 Keterangan komponen panel tab hasil ramalan

Type Komponen	Fungsi Komponen
JPanel	Menampilkan grafik hasil ramalan data latih dan data uji setelah proses peramalan.
JTable	Menampilkan perbandingan hasil ramalan dengan nilai aktual.

4.5 Perancangan Skenario Pengujian

Perancangan skenario pengujian diperlukan untuk mempermudah melakukan pengujian metode SVR-PPSO dalam peramalan runut waktu curah hujan. Pada penggunaan metode SVR-PPSO terdapat beberapa konfigurasi dari pengguna terhadap sistem yang dapat mempengaruhi hasil peramalan. Berdasarkan hal tersebut, sehingga diperlukan uji coba terhadap konfigurasi metode SVR-PPSO yang dimasukkan oleh pengguna. Berikut merupakan scenario pengujian secara umum terhadap metode SVR-PPSO yang telah diimplementasikan.

1. Uji coba batas parameter SVR.
2. Uji coba jumlah partikel PPSO.
3. Uji coba jumlah iterasi PPSO.
4. Uji coba jumlah *slave* PSO.
5. Uji coba panjang periode runut waktu (*lag time*).
6. Uji coba peramalan runut waktu.

Uji coba dilakukan dengan masukan konfigurasi data, metode SVR-PPSO, dan batas parameter SVR dari pengguna terhadap sistem yang telah diimplementasikan. Masing-masing uji coba dari beberapa skenario dilakukan sebanyak 10 kali, sehingga diperoleh rerata berdasarkan percobaan yang dilakukan sebanyak 10 kali tersebut. Tiap uji coba yang dilakukan mengacu pada evaluasi kinerja peramalan berupa nilai RMSE, nilai *fitness* yang diperoleh dari nilai RMSE, dan waktu eksekusi dalam hitungan detik.

4.5.1 Rancangan Uji Coba Batas Parameter SVR

Uji coba batas parameter SVR ditujukan untuk membatasi ruang dimensi partikel dalam pencarian solusi agar dapat menghasilkan kombinasi parameter SVR yang optimal pada proses pelatihan. Dasar pemilihan batas minimal dan maksimal parameter SVR mengadopsi dari hasil uji coba beberapa penelitian lain yang terkait dan telah dilakukan sebelumnya pada Tabel 4.13.

4.5.1.1 Rancangan Uji Coba Batas Parameter γ

Berdasarkan Tabel 4.13 dapat menggunakan batas parameter γ minimal lebih dari nilai 0 dan maksimal menggunakan persamaan (2.38) pada SVR. Namun, pada penelitian ini memperluas uji coba dengan membatasi parameter γ minimal bernilai 0.0001 dan maksimal bernilai 100. Pada Tabel 4.57 dan Tabel 4.62 merupakan rancangan uji coba batas parameter γ .

Tabel 4.57 Rancangan uji coba batas parameter γ

Batas Min γ	Batas Maks γ	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0.0001	0.001											
	0.01											
	0.1											
0.001	0.01											
	0.1											

Tabel 4.58 Rancangan uji coba batas parameter γ (lanjutan)

Batas Min γ	Batas Maks γ	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0.01	1											
	0.1											
	1											
	10											
0.1	1											
	10											
	100											

4.5.1.2 Rancangan Uji Coba Batas Parameter λ

Berdasarkan Tabel 4.13 dapat menggunakan batas parameter λ minimal bernilai 0 dan maksimal bernilai 5 pada SVR. Pada Tabel 4.59 merupakan rancangan uji coba batas parameter λ .

Tabel 4.59 Rancangan uji coba batas parameter λ

Batas Min λ	Batas Maks λ	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0	0											
	0.0001											
	0.001											
0.0001	0.001											
	0.01											
	0.1											
0.001	0.01											
	0.1											
	1											
0.01	0.1											
	1											
	5											

4.5.1.3 Rancangan Uji Coba Batas Parameter c

Berdasarkan Tabel 4.13 dapat menggunakan batas parameter c minimal bernilai 0.001 dan maksimal bernilai 1000 pada SVR. Pada Tabel 4.60 dan Tabel 4.61 merupakan rancangan uji coba batas parameter c .

Tabel 4.60 Rancangan uji coba batas parameter C

Batas Min c	Batas Maks c	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0.001	0.01											
	0.1											
	1											
0.01	0.1											
	1											

Tabel 4.61 Rancangan uji coba batas parameter C (lanjutan)

Batas Min c	Batas Maks c	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
1	10											
	10											
	100											
	1000											
10	100											
	1000											
100	1000											

4.5.1.4 Rancangan Uji Coba Batas Parameter ϵ

Berdasarkan Tabel 4.13 dapat menggunakan batas parameter ϵ minimal bernilai 0 dan maksimal bernilai 100. Namun, pada penelitian ini memperluas uji coba dengan membatasi parameter ϵ minimal bernilai 0.0001 dan maksimal bernilai 100. Pada Tabel 4.62 merupakan rancangan uji coba batas parameter ϵ .

Tabel 4.62 Rancangan uji coba batas parameter ϵ

Batas Min ϵ	Batas Maks ϵ	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0.0001	0.001											
	0.01											
	0.1											
0.001	0.01											
	0.1											
	1											
0.01	0.1											
	1											
	10											
0.1	1											
	10											
	100											

4.5.1.5 Rancangan Uji Coba Batas Parameter γ_k kernel

Berdasarkan Tabel 4.13 dapat menggunakan batas parameter γ_k kernel minimal bernilai 0.1 dan maksimal bernilai 100. Namun, pada penelitian ini melakukan uji coba dengan membatasi parameter γ_k kernel minimal bernilai 0.0001 dan maksimal bernilai 100. Pada Tabel 4.63 dan Tabel 4.64 merupakan rancangan uji coba batas parameter γ_k kernel.

Tabel 4.63 Rancangan uji coba batas parameter γ_k kernel

Batas Min γ_k kernel	Batas Maks γ_k kernel	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0.0001	0.001											
	0.01											

Tabel 4.64 Rancangan uji coba batas parameter γ kernel (lanjutan)

Batas Min γ kernel	Batas Maks γ kernel	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0.001	0.1											
	0.01											
	0.1											
	1											
0.01	0.1											
	1											
	10											
0.1	1											
	10											
	100											
1	10											
	100											
10	100											

4.5.1.6 Rancangan Uji Coba Batas Parameter d Kernel

Berdasarkan Tabel 4.13 dapat menggunakan batas parameter d kernel minimal bernilai 1 dan maksimal bernilai 3 pada SVR. Pada Tabel 4.65 merupakan rancangan uji coba batas parameter d kernel.

Tabel 4.65 Rancangan uji coba batas parameter d kernel

Batas Min d Kernel	Batas Maks d Kernel	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
1	1											
	2											
	3											
2	2											
	3											
3	3											

4.5.2 Rancangan Uji Coba Jumlah Iterasi SVR

Uji coba banyaknya jumlah iterasi SVR ditujukan untuk dapat melakukan penyelesaian masalah dengan melakukan pemodelan ramalan disetiap iterasi yang menghasilkan ramalan data latih pada proses pelatihan SVR yang optimal. Pada Tabel 4.66 dan Tabel 4.73 merupakan rancangan uji coba jumlah iterasi SVR.

Tabel 4.66 Rancangan uji coba jumlah iterasi SVR

Jumlah Iterasi SVR	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>	Rerata Waktu Eksekusi (detik)
	1	2	3	4	5	6	7	8	9	10		
10												
50												

Tabel 4.67 Rancangan uji coba jumlah iterasi SVR (lanjutan)

Jumlah Iterasi SVR	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>	Rerata Waktu Eksekusi (detik)
	1	2	3	4	5	6	7	8	9	10		
100												
500												
1000												
5000												
10000												
50000												
100000												
500000												
1000000												

4.5.3 Rancangan Uji Coba Jumlah Partikel PPSO

Uji coba banyaknya jumlah partikel PPSO ditujukan untuk jumlah partikel pada *slave* PSO dan *master* PSO yang dapat melakukan pencarian solusi agar dapat menghasilkan kombinasi parameter SVR yang optimal. Pada Tabel 4.68 merupakan rancangan uji coba jumlah partikel PPSO.

Tabel 4.68 Rancangan uji coba jumlah partikel PPSO

Jumlah Partikel	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>	Rerata Waktu Eksekusi (detik)
	1	2	3	4	5	6	7	8	9	10		
5												
10												
15												
20												
25												
30												
35												
40												
45												
50												

4.5.4 Rancangan Uji Coba Jumlah Iterasi PPSO

Uji coba banyaknya jumlah iterasi PPSO ditujukan untuk dapat melakukan penyelesaian masalah dengan mengubah atau memperbarui posisi dan kecepatan tiap partikel sehingga menghasilkan kombinasi parameter SVR yang optimal. Pengaturan iterasi PPSO berlaku pada *slave* PSO dan *master* PSO dalam pencarian solusi optimal. Pada Tabel 4.69 merupakan rancangan uji coba jumlah iterasi PPSO.

Tabel 4.69 Rancangan uji coba jumlah iterasi PPSO

Jumlah Iterasi PPSO	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>	Rerata Waktu Eksekusi (detik)
	1	2	3	4	5	6	7	8	9	10		
10												
20												
30												
40												
50												
60												
70												
80												
90												
100												

4.5.5 Rancangan Uji Coba Jumlah *Slave* PSO

Uji coba banyaknya jumlah *slave* PSO ditujukan untuk dapat melakukan penyelesaian masalah dalam pencarian solusi optimum pada kawanan partikel berbasis populasi ganda secara paralel atau waktu bersamaan. Pada Tabel 4.70 merupakan rancangan uji coba jumlah *slave* PSO.

Tabel 4.70 Rancangan uji coba jumlah *slave* PSO

Jumlah <i>Slave</i> PSO	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>	Rerata Waktu Eksekusi (detik)
	1	2	3	4	5	6	7	8	9	10		
0												
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												

4.5.6 Rancangan Uji Coba Panjang Periode Runut Waktu (*Lag Time*)

Uji coba panjang periode runut waktu (*lag time*) ditujukan untuk dapat melakukan peramalan yang optimal. Dalam penentuan panjang periode runut waktu (*lag time*) diperlukan pengetahuan mendasar mengenai data runut waktu yang akan digunakan seperti data latih dan data uji. Pada Tabel 4.71 merupakan rancangan uji coba panjang periode runut waktu (*lag time*).

Tabel 4.71 Rancangan uji coba panjang periode runut waktu (*lag time*)

Panjang Periode Runut Waktu (<i>Lag Time</i>)	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
1											
3											
6											
9											
12											
15											
18											
21											
24											
27											
30											
33											
36											

4.5.7 Rancangan Uji Coba Peramalan Runut Waktu Curah Hujan

Uji coba peramalan runut waktu ditujukan untuk melakukan pengujian metode SVR-PPSO menggunakan data runut waktu curah hujan. Pada konfigurasi data diperlukan pengetahuan dasar mengenai data runut waktu yang akan digunakan. Sebelum uji coba peramalan runut waktu curah hujan dilakukan uji coba pengaruh parameter ANOVA RBF *kernel* terlebih dahulu untuk mendapatkan konfigurasi parameter yang tepat. Selanjutnya dilakukan uji coba peramalan runut waktu curah hujan tahunan, musiman, dan tahun ke tahun.

4.5.7.1 Rancangan Uji Coba Pengaruh Parameter ANOVA RBF *Kernel*

Berdasarkan Tabel 4.13 dalam beberapa penelitian mengatur nilai parameter ANOVA RBF *kernel* dengan konsisten pada parameter γ_k *kernel* dan d . Pada Tabel 4.72 merupakan rancangan uji coba pengaruh batas parameter ANOVA RBF *kernel*.

Tabel 4.72 Rancangan uji coba pengaruh batas parameter ANOVA RBF *kernel*

Batas Min γ_k <i>kernel</i>	Batas Maks γ_k <i>kernel</i>	Batas Min d <i>Kernel</i>	Batas Maks d <i>Kernel</i>	Evaluasi Pelatihan (SVR-PPSO)		Evaluasi Pengujian (SVR)	
				RMSE	<i>fitness</i>	RMSE	<i>fitness</i>
1	100	1	1				
		2	2				
		3	3				
0.0001	0.1	1	1				
		2	2				
		3	3				

4.5.7.2 Rancangan Uji Coba Peramalan Runut Waktu Curah Hujan Tahunan

Uji coba peramalan runut waktu curah hujan tahunan menggunakan pola horizontal yang telah dibahas pada sub bab 4.1.4.1 sebelumnya. Uji coba ini ditujukan untuk menguji metode SVR-PPSO dalam melakukan peramalan runut waktu curah hujan data tahunan. Dalam penentuan variasi runut waktu diperlukan pengetahuan mendasar mengenai data runut waktu yang akan digunakan. Pada Tabel 4.73 merupakan rancangan uji coba peramalan runut waktu curah hujan tahunan menggunakan pola horizontal.

Tabel 4.73 Rancangan uji coba peramalan runut waktu curah hujan tahunan

Runut Waktu Dasarian		Evaluasi Pelatihan (SVR-PPSO)		Evaluasi Pengujian (SVR)	
Awal	Akhir	RMSE	<i>fitness</i>	RMSE	<i>fitness</i>
1-jan-2000	3-des-2002				
	3-des-2003				
	3-des-2004				
	3-des-2005				
	3-des-2006				
	3-des-2007				
	3-des-2008				
	3-des-2009				
	3-des-2010				
	3-des-2011				
	3-des-2012				
	3-des-2013				
	3-des-2014				
	3-des-2015				

4.5.7.3 Rancangan Uji Coba Peramalan Runut Waktu Curah Hujan Musiman

Uji coba peramalan runut waktu curah hujan musiman berdasarkan hasil analisis musim hujan di wilayah Poncokusumo pada sub bab 4.1.3.2. Dalam peramalan runut waktu musiman menggunakan pola horizontal yang telah dibahas pada sub bab 4.1.4.1 sebelumnya. Uji coba ini ditujukan untuk menguji metode SVR-PPSO dalam melakukan peramalan runut waktu curah hujan musiman. Dalam penentuan variasi runut waktu diperlukan pengetahuan mendasar mengenai data runut waktu yang akan digunakan. Pada Tabel 4.74 merupakan rancangan uji coba peramalan runut waktu curah hujan musiman menggunakan pola horizontal.

Tabel 4.74 Rancangan uji coba peramalan runut waktu curah hujan musiman

Runut Waktu Dasarian		Evaluasi Pelatihan (SVR-PPSO)		Evaluasi Pengujian (SVR)	
Awal	Akhir	RMSE	<i>fitness</i>	RMSE	<i>fitness</i>
1-Okt-00	3-Apr-02				
	3-Apr-03				
	3-Apr-04				
	3-Apr-05				
	3-Apr-06				
	3-Apr-07				
	3-Apr-08				
	3-Apr-09				
	3-Apr-10				
	3-Apr-11				
	3-Apr-12				
	3-Apr-13				
	3-Apr-14				
	3-Apr-15				

4.5.7.4 Rancangan Uji Coba Peramalan Runut Waktu Curah Hujan Tahun ke Tahun

Uji coba peramalan runut waktu curah hujan tahun ke tahun menggunakan pola vertikal yang telah dibahas pada sub bab 4.1.4.2 sebelumnya. Dalam penentuan variasi runut waktu diperlukan pengetahuan mendasar mengenai data runut waktu yang akan digunakan. Pada Tabel 4.75 merupakan rancangan uji coba peramalan runut waktu curah hujan tahun ke tahun menggunakan pola vertikal yang mewakili musim hujan pada bulan januari dan desember.

Tabel 4.75 Rancangan uji coba variasi runut waktu ramalan pola vertikal

Runut Waktu Dasarian		<i>(Lag Time)</i>	Ramalan (Data Uji)	Evaluasi Pelatihan (SVR-PPSO)		Evaluasi Pengujian (SVR)	
Awal	Akhir			RMSE	<i>fitness</i>	RMSE	<i>fitness</i>
2-jan-2000	2-jan-2015	1	1				
		3	3				
		5	5				
2-des-2000	2-agt-2015	1	1				
		3	3				
		5	5				

4.5.8 Rancangan Uji Coba Perbandingan Peramalan Runut Waktu Curah Hujan Menggunakan Metode SVR, SVR-PPSO, dan Regresi Linier Berganda

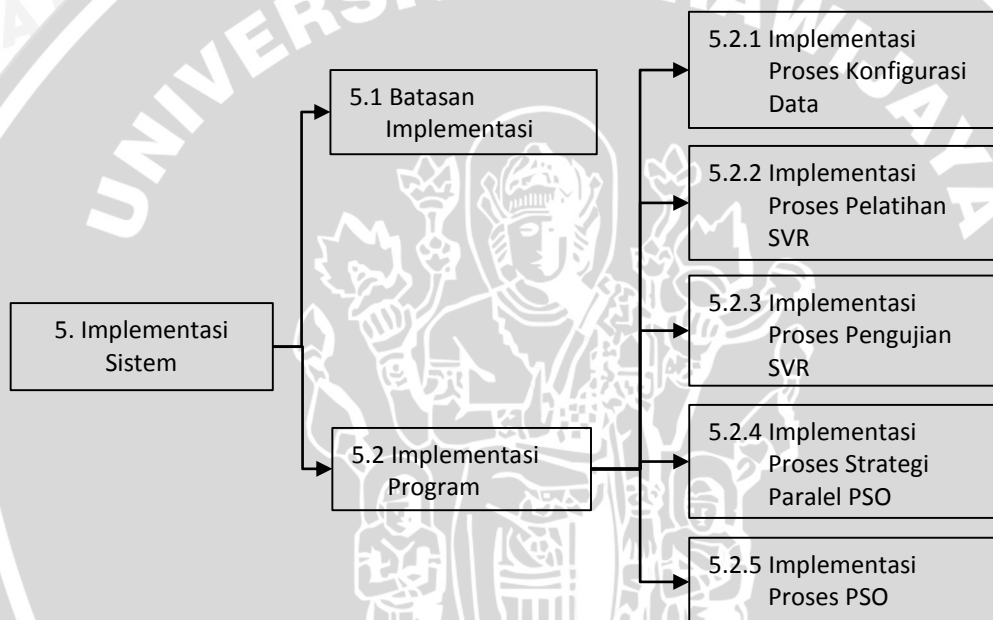
Uji coba perbandingan peramalan runut waktu curah hujan ditujukan untuk mengetahui masing-masing hasil evaluasi kinerja peramalan menggunakan metode SVR, SVR-PPSO, dan regresi linier berganda. Pada perbandingan metode SVR dilakukan tanpa adanya metode optimasi parameter oleh PPSO, sama halnya dengan metode regresi linier berganda. Pada perbandingan ini, menerapkan uji coba peramalan runut waktu curah hujan tahunan menggunakan pola horizontal yang telah dibahas pada sub bab 4.1.4.1 sebelumnya. Dengan data runut waktu dasarian 1 januari 2000 sampai dasarian 3 desember 2014 sebagai data latih, sedangkan data runut waktu dasarian 1 januari 2015 sampai dasarian 3 desember 2015 sebagai data uji. Berdasarkan peramalan runut waktu curah hujan tahunan menggunakan pola horizontal, maka menggunakan panjang periode runut waktu (*lag time*) bernilai 36. Pada Tabel 4.76 merupakan rancangan uji coba perbandingan peramalan runut waktu curah hujan menggunakan metode SVR, SVR-PPSO, dan regresi linier berganda.

Tabel 4.76 Rancangan uji coba perbandingan peramalan runut waktu curah hujan menggunakan metode SVR, SVR-PPSO, dan regresi linier berganda

Metode	Evaluasi Pelatihan		Evaluasi Pengujian	
	RMSE	<i>fitness</i>	RMSE	<i>fitness</i>
SVR				
SVR-PPSO				
Regresi Linier Berganda				

BAB 5 IMPLEMENTASI SISTEM

Pada bab ini merupakan tahap implementasi sistem yang menerapkan hasil perancangan. Implementasi sistem dilakukan untuk menghasilkan sebuah sistem berupa perangkat lunak berbasis *prototype* yang dapat melakukan peramalan runut waktu curah hujan menggunakan metode *Support Vector Regression* dan *Parallel Particle Swarm Optimization* (SVR-PPSO). Pada proses implementasi sistem terdapat batasan implementasi yang bertujuan untuk memperjelas dan mencegah pelebaran masalah dari ruang lingkup penelitian ini. Pada bab Implementasi sistem akan membahas implementasi program menggunakan bahasa pemrograman java dengan bantuan sistem komputer yang telah dijelaskan pada sub 3.2.1. Berikut merupakan gambaran secara umum tahap implementasi sistem dalam bentuk diagram struktur yang ditunjukkan pada Gambar 5.1.



Gambar 5.1 Struktur implementasi sistem

5.1 Batasan Implementasi

Pada proses implementasi diperlukan pembatasan dengan menyesuaikan hasil perancangan yang telah dilakukan sebelumnya. Batasan implementasi ditujukan untuk memperjelas dan mencegah adanya pelebaran masalah dari ruang lingkup penelitian ini. Beberapa batasan implementasi yang diterapkan dalam implementasi sistem pada penelitian ini sebagai berikut.

1. Sistem yang diimplementasikan berupa perangkat lunak berbasis *prototype* yang perlu dilakukan beberapa pengujian lebih lanjut.
2. Sistem dibangun berdasarkan ruang lingkup perangkat lunak berbasis *desktop* dengan menggunakan bahasa pemrograman java.

3. Data yang digunakan dalam sistem disimpan secara sederhana dalam *Database Management System* (DBMS) SQLite.
4. Data yang disimpan dalam DBMS SQLite adalah data runut waktu curah hujan di wilayah Poncokusumo.
5. Data yang disimpan dalam DBMS SQLite hanya dapat dibaca tanpa ada fitur manajemen data seperti ubah dan hapus data.
6. Masukan dari pengguna yang dapat diolah oleh sistem berupa angka dan opsi pilihan terhadap konfigurasi data dan metode SVR-PPSO.
7. Proses paralel yang dapat dilakukan oleh sistem memiliki batasan menyesuaikan dengan spesifikasi sistem komputer pengguna.
8. Keluaran yang dihasilkan oleh sistem berupa angka dalam bentuk teks, catatan (*log*) sistem berupa teks dan gambar grafik.

5.2 Implementasi Program

Pada implementasi program mengacu hasil perancangan sebelumnya yang telah dilakukan formulasi metode SVR-PPSO pada sub bab 4.2. Dalam implementasi program terdapat proses konfigurasi data, proses pelatihan SVR, proses pengujian SVR, proses strategi paralel PSO, dan proses PSO.

5.2.1 Implementasi Proses Konfigurasi Data

Proses konfigurasi data merupakan tahap awal yang harus dilakukan pada sistem sebelum melakukan proses peramalan. Proses ini berfungsi untuk melakukan menginisialisasi data pada tahap awal diantaranya data latih dan data uji berdasarkan panjang periode (*lag time*), dan mengatur panjang periode ramalan. Implementasi proses konfigurasi data ditunjukkan pada Kode Program 5.1.

Kode Program 5.1 Implementasi proses konfigurasi data

```

1 public static void konfigurasiData(double[] dataAwal, int lagTime,
2   int ramalan) {
3     double[][] setDataLatih = new double[(dataAwal.length -
4   lagTime) - ramalan][lagTime + 1];
5     double[][] setDataUji = new double[ramalan][lagTime + 1];
6     for (int i = 0; i < setDataLatih.length; i++) {
7       for (int j = 0; j < setDataLatih[i].length; j++) {
8         setDataLatih[i][j] = dataAwal[i + j];
9       }
10    }
11    for (int i = 0; i < setDataUji.length; i++) {
12      for (int j = 0; j < setDataUji[i].length; j++) {
13        setDataUji[i][j] = dataAwal[(dataAwal.length -
14   lagTime) - ramalan) + (i + j)];
15      }
16    }
17    dataLatih = setDataLatih;
18    dataUji = setDataUji;
19  }

```

Penjelasan Kode Program 5.1:

1. Baris 3-5 merupakan proses inisialisasi variabel data latih dan data uji berdasarkan masukan pengguna.

2. Baris 6-10 merupakan proses pengisian data ke dalam variabel data latih.
3. Baris 11-16 merupakan proses pengisian data ke dalam variabel data uji.
4. Baris 18-19 merupakan proses pengambilan data sebagai data latih dan data uji.

5.2.2 Implementasi Proses Pelatihan SVR

Proses pelatihan SVR merupakan tahap pemodelan ramalan berdasarkan data latih yang digunakan. Proses ini berperan penting dalam proses peramalan yang terdiri dari beberapa sub proses perhitungan. Dalam implementasi proses pelatihan SVR mengacu kepada perancangan yang telah dijelaskan sebelumnya pada sub bab 4.2.2.

5.2.2.1 Implementasi Proses Perhitungan Matriks Hessian Data Latih

Proses perhitungan matriks Hessian data latih merupakan bagian dari tahap pelatihan data ramalan menggunakan metode SVR. Proses ini dilakukan setelah data latih dinormalisasi pada ruang fitur. Implementasi proses perhitungan matriks Hessian data latih ditunjukkan pada Kode Program 5.2.

Kode Program 5.2 Implementasi proses perhitungan matriks Hessian data latih

```

1 public double[][] getRIJLatih() {
2     double MatriksRijLatih[][] = new
3     double[dataLatih.length][dataLatih.length];
4     double normData[][] = getNormDataLatih();
5     for (int i = 0; i < normData[0].length; i++) {
6         for (int j = 0; j < normData[0].length; j++) {
7             MatriksRijLatih[i][j] = normData[i][j] +
8             Math.pow(lambda, 2);
9         }
10    }
11    return MatriksRijLatih;
12 }
```

Penjelasan Kode Program 5.2:

1. Baris 2-4 merupakan proses inisialisasi variabel matriks Hessian data latih dan normalisasi data latih.
2. Baris 5-10 merupakan proses perhitungan matriks Hessian data latih menggunakan persamaan (2.34) berdasarkan data latih yang telah dinormalisasi pada ruang fitur.

5.2.2.2 Implementasi Proses Perhitungan Normalisasi Data Latih

Proses perhitungan normalisasi data latih pada ruang fitur merupakan bagian dari tahap pelatihan data ramalan menggunakan metode SVR. Proses ini dilakukan setelah perhitungan ANOVA RBF *kernel* menggunakan data latih. Implementasi proses perhitungan normalisasi data latih pada ruang fitur ditunjukkan pada Kode Program 5.3.

Kode Program 5.3 Implementasi proses perhitungan normalisasi data latih

```

1 public double[][] getNormDataLatih() {
2     double data[][] = getJarakDataLatihANOVA RBF();
3 }
```

```

4      double normData[][] = new
5      double[data.length][data.length];
6      for (int i = 0; i < data.length; i++) {
7          for (int j = 0; j < data.length; j++) {
8              normData[i][j] = data[i][j] / Math.sqrt(data[i][i]
9              * data[j][j]);
10         }
11     }
12     return normData;
13 }

```

Penjelasan Kode Program 5.3:

1. Baris 2-4 merupakan proses inialisasi variabel jarak data latih dan normalisasi data latih.
2. Baris 5-10 merupakan proses perhitungan normalisasi data latih pada ruang fitur yang menerapkan persamaan (2.43) berdasarkan hasil perhitungan ANOVA RBF *kernel* menggunakan data latih.

5.2.2.3 Implementasi Proses Perhitungan ANOVA RBF *Kernel* Menggunakan Data Latih

Proses perhitungan ANOVA RBF *kernel* menggunakan data latih merupakan bagian dari tahap pelatihan data ramalan. Oleh karena itu, dalam proses ini membutuhkan data latih yang telah dikonfigurasi pada proses sebelumnya. Implementasi proses perhitungan normalisasi data latih pada ruang fitur ditunjukkan pada Kode Program 5.4.

Kode Program 5.4 Implementasi proses perhitungan ANOVA RBF *kernel* menggunakan data latih

```

1  public double[][] getJarakDataLatihANOVA RBF() {
2      double data[][] = new
3      double[dataLatih.length][dataLatih.length];
4      for (int i = 0; i < dataLatih.length; i++) {
5          for (int j = 0; j < dataLatih.length; j++) {
6              double total = 0;
7              for (int k = 0; k < dataLatih[i].length - 1; k++)
8              {
9                  total += Math.exp(-gammaAnova *
10 Math.pow(dataLatih[j][k] - dataLatih[i][k], 2));
11              }
12              data[i][j] = Math.pow(total, degreeAnova);
13          }
14      }
15      return data;
16  }

```

Penjelasan Kode Program 5.4:

1. Baris 2-3 merupakan proses inialisasi variabel data latih.
2. Baris 4-14 merupakan proses perhitungan ANOVA RBF *kernel* menggunakan data latih yang menerapkan persamaan (2.33).

5.2.2.4 Implementasi Proses Perhitungan Nilai α_i^* dan α_i

Proses perhitungan nilai α_i^* dan α_i merupakan bagian inti dari tahap pelatihan data ramalan. Proses ini dilakukan setelah diperoleh hasil perhitungan matriks

Hessian data latih. Implementasi proses perhitungan nilai α_i^* dan α_i ditunjukkan pada Kode Program 5.5.

Kode Program 5.5 Implementasi proses perhitungan nilai α_i^* dan α_i

```

1 public double[][] getMatriksAlfa(int maxIter, double RijLatih[][])
2 {
3     double a[] = new double[dataLatih.length];
4     double aStar[] = new double[dataLatih.length];
5     double deltaAStar[] = new double[dataLatih.length];
6     double deltaAI[] = new double[dataLatih.length];
7     double NilaiError[] = new double[dataLatih.length];
8     double matriksA[][] = new double[2][dataLatih.length];
9     for (int iterasi = 0; iterasi <= maxIter; iterasi++) {
10         double MaxDelta = 0;
11         for (int i = 0; i < dataLatih.length; i++) {
12             double jumlahRij = 0;
13             for (int j = 0; j < RijLatih[i].length; j++) {
14                 jumlahRij = jumlahRij + (aStar[j] - a[j]) *
15                 RijLatih[j][i];
16             }
17             NilaiError[i] = (dataLatih[i][dataLatih[i].length
18 - 1]) - jumlahRij;
19             deltaAStar[i] = Math.min(Math.max(gamma *
20 (NilaiError[i] - epsilon), -aStar[i]), complexity - aStar[i]);
21             deltaAI[i] = Math.min(Math.max(gamma * (-
22 NilaiError[i] - epsilon), -a[i]), complexity - a[i]);
23             MaxDelta = Double.max(MaxDelta,
24 Double.max(deltaAStar[i], deltaAI[i]));
25         }
26         for (int k = 0; k < dataLatih.length; k++) {
27             aStar[k] = deltaAStar[k] + aStar[k];
28             a[k] = deltaAI[k] + a[k];
29         }
30         for (int m = 0; m < dataLatih.length; m++) {
31             matriksA[0][m] = aStar[m];
32             matriksA[1][m] = a[m];
33         }
34         if (MaxDelta < epsilon) {
35             break;
36         }
37     }
38     this.matriksA = matriksA;
39     return matriksA;
40 }
41

```

Penjelasan Kode Program 5.5:

1. Baris 3-8 merupakan proses inisialisasi variabel alfa, delta alfa, nilai E_i dan matriks α_i^* dan α_i .
2. Baris 11-26 merupakan proses pelatihan SVR berdasarkan perhitungan (3.a), (3.b) dan (3.c) pada algoritma sekuensial SVR yang telah dibahas pada sub bab 2.5.2.
3. Baris 27-30 merupakan proses perhitungan nilai $\delta\alpha_i^*$ dan $\delta\alpha_i$ menggunakan persamaan (2.36) dan (2.37) berdasarkan nilai E_i menggunakan persamaan (2.35).

4. Baris 31-34 merupakan proses pembaruan nilai a_i^* dan a_i menggunakan persamaan (2.39) dan (2.40) berdasarkan hasil perhitungan nilai δa_i^* dan δa_i sebelumnya.
5. Baris 35-37 merupakan proses pengecekan kondisi apakah proses pelatihan SVR telah mencapai konvergensi atau belum berdasarkan catatan nilai maksimal dari nilai δa_i^* dan δa_i . Jika sudah mencapai konvergensi, maka proses iterasi pelatihan SVR berhenti.

5.2.2.5 Implementasi Proses Perhitungan Regresi Data Latih

Proses perhitungan regresi data latih merupakan bagian dari tahap pelatihan data ramalan. Proses ini dilakukan setelah diperoleh hasil ramalan data latih dan nilai a_i^* dan a_i dalam proses pelatihan. Implementasi proses perhitungan regresi data latih ditunjukkan pada Kode Program 5.6.

Kode Program 5.6 Implementasi proses perhitungan regresi data latih

```

1 public double[] getFxFatih(double matriksA[], double getRij[][])
2 {
3     double fx[] = new double[dataLatih.length];
4     fxLatih = new double[fx.length];
5     for (int i = 0; i < fx.length; i++) {
6         double prediksi = 0;
7         for (int j = 0; j < getRij[i].length; j++) {
8             prediksi = prediksi + (matriksA[0][j] -
9 matriksA[1][j]) * getRij[i][j];
10        }
11        fx[i] = prediksi;
12        fxLatih[i] = fx[i];
13    }
14    return fx;
15 }
```

Penjelasan Kode Program 5.6:

1. Baris 3-4 merupakan proses inisialisasi variabel perhitungan regresi sesuai dengan banyaknya data latih.
2. Baris 8-9 merupakan proses penjumlahan dari hasil matriks Rij data latih dikalikan dengan inisialisasi awal nilai a_i^* dan a_i yang telah diperoleh pada proses perhitungan sebelumnya.
3. Baris 11-12 merupakan proses pengisian hasil perhitungan regresi ke dalam variabel untuk dapat digunakan pada proses evaluasi kinerja peramalan.

5.2.2.6 Implementasi Proses Evaluasi Kinerja Peramalan

Proses evaluasi kinerja peramalan merupakan bagian dari tahap pelatihan dan pengujian data ramalan. Proses ini dilakukan setelah diperoleh hasil ramalan data dan nilai aktual sebagai pembanding. Implementasi proses evaluasi kinerja peramalan ditunjukkan pada Kode Program 5.7.

Kode Program 5.7 Implementasi proses evaluasi kinerja peramalan

```

1 public double getRMSE(double fx[], double dataAktual[][]) {
```

```

2      double RMSE, jumlahRMSE = 0;
3      for (int i = 0; i < fx.length; i++) {
4          jumlahMSE +=
5      Math.pow((dataAktual[i][dataAktual[i].length - 1] - fx[i]), 2);
6      }
7      RMSE = Math.sqrt(jumlahRMSE / fx.length);
8      return RMSE;
9  }

```

Penjelasan Kode Program 5.7:

1. Baris 2 merupakan proses inisialisasi variabel nilai kesalahan RMSE.
2. Baris 4-3 merupakan proses evaluasi kinerja peramalan menggunakan Mean Square Error (MSE) pada persamaan (2.4).
3. Baris 7 merupakan proses evaluasi kinerja peramalan dengan pengakaran terhadap MSE menjadi *Root Mean Square Error* (RMSE) menggunakan persamaan (2.5) dan mengisi nilai hasil perhitungan ke dalam variabel RMSE.

5.2.3 Implementasi Proses Pengujian SVR

Proses pengujian SVR merupakan tahap peramalan berdasarkan pemodelan data latih yang telah dilakukan sebelumnya. Proses ini dilakukan setelah proses pelatihan SVR menggunakan data uji yang berbeda pada data latih. Dalam proses pengujian SVR terdiri dari beberapa sub proses perhitungan. Dalam implementasi proses pengujian SVR mengacu kepada perancangan yang telah dijelaskan sebelumnya pada sub bab 4.2.3.

5.2.3.1 Implementasi Proses Perhitungan Regresi Data Uji

Proses perhitungan perhitungan regresi data uji merupakan bagian dari tahap pengujian data ramalan. Proses ini dilakukan setelah diperoleh hasil pemodelan data latih dan nilai α_i^* dan α_i dalam proses pelatihan. Terdapat beberapa perbedaan proses perhitungan regresi data uji dengan data latih dalam pola perubahan data. Implementasi proses perhitungan regresi data latih ditunjukkan pada Kode Program 5.8.

Kode Program 5.8 Implementasi proses perhitungan regresi data uji

```

1  public double[] getFxUji(double matriksA[][]) {
2      double fx[] = new double[dataUji.length];
3      fxUji = new double[fx.length];
4      for (int i = 0; i < dataUji.length; i++) {
5          double prediksi = 0;
6          double SekuensialRIJ[] = getRIJUji(dataUji[i]);
7          for (int j = 0; j < SekuensialRIJ.length; j++) {
8              prediksi = prediksi + (matriksA[0][j] -
9      matriksA[1][j]) * SekuensialRIJ[j];
10         }
11         for (int x = i; x < dataUji.length - 1; x++) {
12             for (int y = 0; y < dataUji[0].length - 1; y++) {
13                 dataUji[x + 1][y] = dataUji[x][y + 1];
14             }
15             dataUji[x + 1][dataUji[0].length - 2] = prediksi;
16         }
17         fx[i] = prediksi;
18         fxUji[i] = fx[i];
19     }

```

20	return fx;
21	}

Penjelasan Kode Program 5.8:

1. Baris 2-3 merupakan proses inisialisasi variabel perhitungan regresi sesuai dengan banyaknya data uji.
2. Baris 8-9 merupakan proses penjumlahan dari hasil matriks Rij data latih dan data uji dikalikan dengan inisialisasi awal nilai a_i^* dan a_i yang telah diperoleh pada proses pelatihan SVR.
3. Baris 13 merupakan proses perubahan atau pergeseran prediktor pada data uji berdasarkan hasil peramalan pengujian SVR dengan model *autoregressive*.
4. Baris 15 merupakan proses perubahan nilai aktual pada data uji berdasarkan hasil peramalan pengujian SVR dengan model *autoregressive*.
5. Baris 17-18 merupakan proses pengisian hasil perhitungan regresi ke dalam variabel untuk dapat digunakan pada proses evaluasi kinerja peramalan.

5.2.3.2 Implementasi Proses Perhitungan Matriks Hessian Data Uji

Proses perhitungan matriks Hessian data uji merupakan bagian dari tahap pengujian data ramalan menggunakan metode SVR. Proses ini dilakukan setelah data uji dinormalisasi pada ruang fitur. Terdapat penambahan data latih pada proses perhitungan matriks Hessian data uji, sehingga terdapat perbedaan dalam proses perhitungan matriks Hessian data latih. Implementasi proses perhitungan matriks Hessian data uji ditunjukkan pada Kode Program 5.9.

Kode Program 5.9 Implementasi proses perhitungan matriks Hessian data uji

1	public double[] getRIJUji(double dataSekuensial[]) {
2	double MatriksRijUji[] = new double[dataLatih.length];
3	double normData[] = getNormDataUji(dataSekuensial);
4	for (int i = 0; i < dataLatih.length; i++) {
5	MatriksRijUji[i] = normData[i] + Math.pow(lambda, 2);
6	}
7	return MatriksRijUji;
8	}

Penjelasan Kode Program 5.9:

1. Baris 2-3 merupakan proses inisialisasi variabel matriks Hessian data uji berdasarkan data latih dan normalisasi data latih.
2. Baris 4-6 merupakan proses perhitungan matriks Hessian data uji menggunakan persamaan (2.34) berdasarkan data uji yang telah dinormalisasi pada ruang fitur.

5.2.3.3 Implementasi Proses Perhitungan Normalisasi Data Uji

Proses perhitungan normalisasi data uji pada ruang fitur merupakan bagian dari tahap pengujian data ramalan menggunakan metode SVR. Proses ini dilakukan setelah perhitungan ANOVA RBF *kernel* menggunakan data latih dan data uji. Terdapat penambahan data latih pada proses normalisasi data uji pada ruang fitur, sehingga terdapat perbedaan dalam proses perhitungan normalisasi

data latih pada ruang fitur. Implementasi proses perhitungan normalisasi data uji pada ruang fitur ditunjukkan pada Kode Program 5.10.

Kode Program 5.10 Implementasi proses perhitungan normalisasi data uji

```

1 public double[] getNormDataUji(double[] dataSekuensial) {
2     double data[] = getJarakDataLatihUjiANOVA
3     RBF(dataSekuensial);
4     double dataLat[][] = getJarakDataLatihANOVA RBF();
5     double normData[] = new double[dataLatih.length];
6     for (int i = 0; i < dataLatih.length; i++) {
7         normData[i] = data[i] / Math.sqrt(dataLat[i][i] *
8         dataLat[i][i]);
9     }
10    return normData;
11 }

```

Penjelasan Kode Program 5.10:

1. Baris 2-4 merupakan proses inialisasi variabel jarak data latih dan data uji, dan normalisasi data latih.
2. Baris 5-8 merupakan proses perhitungan normalisasi data latih pada ruang fitur menggunakan persamaan (2.43) berdasarkan matriks hasil perhitungan ANOVA RBF *kernel* menggunakan data uji dan data latih.

5.2.3.4 Implementasi Proses Perhitungan ANOVA RBFK *kernel* Menggunakan Data Latih dan Data Uji

Proses perhitungan ANOVA RBF *kernel* menggunakan data latih dan data uji menggunakan ANOVA RBF *kernel* merupakan bagian dari tahap pengujian data ramalan. Oleh karena itu, dalam proses ini membutuhkan data latih dan data uji yang telah dikonfigurasi pada proses sebelumnya. Terdapat penambahan data latih pada proses perhitungan ANOVA RBF *kernel* menggunakan data latih dan data uji, sehingga terdapat perbedaan dalam proses perhitungan perhitungan ANOVA RBF *kernel* menggunakan data latih Implementasi proses perhitungan normalisasi data latih pada ruang fitur ditunjukkan pada Kode Program 5.11.

Kode Program 5.11 Implementasi proses perhitungan ANOVA RBF *kernel* menggunakan data latih dan data uji

```

1 public double[] getJarakDataLatihUjiANOVA RBF(double[]
2 dataSekuensial) {
3     double data[] = new double[dataLatih.length];
4     for (int i = 0; i < dataLatih.length; i++) {
5         double total = 0;
6         for (int j = 0; j < dataSekuensial.length - 1; j++) {
7             total += Math.exp(-gammaAnova *
8             Math.pow(dataLatih[i][j] - dataSekuensial[j], 2));
9         }
10        data[i] = Math.pow(total, degreeAnova);
11    }
12    return data;
13 }

```

Penjelasan Kode Program 5.11:

1. Baris 3 merupakan proses inialisasi variabel data latih dan data uji.

- Baris 4-11 merupakan proses perhitungan ANOVA RBF *kernel* menggunakan data latih dan data uji yang menerapkan persamaan (2.33).

5.2.4 Implementasi Proses Strategi Paralel PSO

Proses strategi paralel PSO merupakan tahap awal dalam optimasi parameter SVR untuk konfigurasi komputasi PSO secara paralel. Proses ini dilakukan pada awal pertama kali setelah konfigurasi data untuk mengoptimalkan hasil pemodelan data latih pada proses pelatihan SVR. Dalam proses strategi paralel PSO SVR terdiri dari beberapa sub proses. Dalam implementasi proses strategi paralel PSO mengacu kepada perancangan yang telah dijelaskan sebelumnya pada sub bab 4.2.4.

5.2.4.1 Implementasi Proses Konfigurasi *Slave* PSO

Proses konfigurasi *slave* PSO merupakan bagian dari tahap awal untuk persiapan optimasi parameter SVR secara paralel. Proses ini berfungsi untuk mengatur seberapa banyak proses komputasi PSO secara paralel dalam waktu bersamaan. Implementasi proses konfigurasi *slave* PSO ditunjukkan pada Kode Program 5.12.

Kode Program 5.12 Proses konfigurasi *slave* PSO

```

1 public void slavePSO() throws InterruptedException {
2     if (jumSlave != 0) {
3         Parallel parallel[] = new Parallel[jumSlave];
4         for (int i = 0; i < parallel.length; i++) {
5             parallel[i] = new Parallel(jumSlave, jumPart,
6 iterPSO);
7             Thread thread = new Thread(parallel[i]);
8             thread.setDaemon(false);
9             thread.setName("SLAVE PSO " + (i + 1));
10            thread.start();
11        }
12        synchronized (SlavePartikel) {
13            SlavePartikel.wait();
14        }
15    }
16 }

```

Penjelasan Kode Program 5.12:

- Baris 3-11 merupakan proses pembuatan *thread* baru untuk *slave* PSO berdasarkan konfigurasi jumlah *slave*.
- Baris 10 memulai proses optimasi parameter SVR oleh *slave* PSO.
- Baris 12-14 merupakan proses sinkronisasi untuk menunggu proses optimasi *slave* PSO selesai dengan mengacu pada kesesuaian jumlah partikel *slave* PSO.

5.2.4.2 Implementasi Proses *Slave* PSO

Proses *slave* PSO merupakan tahap optimasi parameter SVR secara paralel. Proses ini berfungsi untuk melakukan proses komputasi PSO secara paralel dalam waktu bersamaan. Implementasi proses *slave* PSO ditunjukkan pada Kode Program 5.13.

Kode Program 5.13 Implementasi proses *slave* PSO

```

1 public void run() {
2     AIWPSO slavePSO = new skripsi.AIWPSO(jumPart, iterPSO);
3     slavePSO.Optimasi();
4     PartikelAwal.addAll(slavePSO.tempArrPartikel);
5     SlavePartikel.addAll(slavePSO.arrPartikel);
6     synchronized (SlavePartikel) {
7         if (SlavePartikel.size() == (jumSlave * jumPart))
8         {
9             SlavePartikel.notifyAll();
10        }
11    }
12 }

```

Penjelasan Kode Program 5.13:

1. Baris 3-11 merupakan proses pembuatan *thread* baru untuk *slave* PSO berdasarkan konfigurasi jumlah *slave*.
2. Baris 2-3 merupakan proses optimasi PSO berdasarkan konfigurasi *slave* PSO.
3. Baris 4-5 merupakan proses pengisian partikel berdasarkan proses *slave* PSO.
4. Baris 6-11 merupakan proses sinkronisasi memberikan notifikasi kepada system apabila jumlah partikel *slave* PSO sesuai dengan dikonfigurasi sebelumnya.

5.2.4.3 Implementasi Proses Pengurutan Partikel *Slave* PSO

Proses pengurutan partikel *slave* PSO merupakan tahap awal sebelum optimasi parameter SVR oleh *master* PSO. Proses ini berfungsi untuk melakukan proses pengambilan partikel terbaik berdasarkan *fitness* sebagai individu awal *master* PSO. Implementasi proses pengurutan partikel *slave* PSO ditunjukkan pada Kode Program 5.14.

Kode Program 5.14 Implementasi proses pengurutan partikel *slave* PSO

```

1 public void setElitePartikel() {
2     List<Partikel> tempPartikel = new ArrayList<>();
3     Partikel temp, getMax;
4     tempPartikel.addAll(SlavePartikel);
5     int jmax, u = tempPartikel.size() - 1;
6     for (int i = 0; i < tempPartikel.size(); i++) {
7         jmax = 0;
8         for (int j = 1; j <= u; j++) {
9             if ((tempPartikel.get(j).fitness) <
10 (tempPartikel.get(jmax).fitness)) {
11                 jmax = j;
12             }
13         }
14         temp = tempPartikel.get(u);
15         getMax = tempPartikel.get(jmax);
16         tempPartikel.set(u, getMax);
17         tempPartikel.set(jmax, temp);
18         u--;
19     }
20     double upperHalf = Math.round(tempPartikel.size() / 2);
21     for (int k = 0; k < upperHalf; k++) {
22         ElitePartikel.add(k, tempPartikel.get(k));
23     }
24 }

```

Penjelasan Kode Program 5.14:

1. Baris 2-4 merupakan proses penyalinan partikel *slave* PSO ke dalam variabel partikel sementara untuk dilakukan pengurutan partikel berdasarkan *fitness*.
2. Baris 6-19 merupakan proses pertukaran posisi partikel berdasarkan *fitness* tertinggi yang dimiliki masing-masing partikel.
3. Baris 20-23 merupakan proses proses penyalinan partikel *slave* PSO yang telah urut ke dalam variabel partikel *elite* atau partikel terbaik.

5.2.4.4 Implementasi Proses *Master* PSO

Proses *master* PSO merupakan tahap optimasi parameter SVR menggunakan individu awal terbaik dari *slave* PSO. Proses ini berfungsi untuk melakukan proses optimasi parameter SVR secara global optimum. Implementasi proses *master* PSO ditunjukkan pada Kode Program 5.15.

Kode Program 5.15 Implementasi proses *master* PSO

```

1 public void masterPSO() throws InterruptedException {
2     AIWPSO masterPSO = null;
3     if (jumSlave == 0) {
4         masterPSO = new skripsi.AIWPSO(jumPart, iterPSO);
5         masterPSO.Optimasi();
6         PartikelAwal.addAll(masterPSO.tempArrPartikel);
7         MasterPartikel.addAll(masterPSO.arrPartikel);
8         gBest = masterPSO.gBest;
9     } else if (jumSlave != 0) {
10        masterPSO = new skripsi.AIWPSO(ElitePartikel,
11 iterPSO);
12        masterPSO.Optimasi();
13        MasterPartikel.addAll(masterPSO.arrPartikel);
14        gBest = masterPSO.gBest;
15    }
16 }

```

Penjelasan Kode Program 5.15:

1. Baris 2 merupakan proses permulaan *master* PSO.
2. Baris 3 merupakan proses pengecekan kondisi apabila jumlah slave PSO bernilai 0 maka dilakukan proses master PSO dengan menginisialisasi individu awal secara acak (random) menggunakan proses pada baris 4-8.
3. Sebaliknya, apabila jumlah slave PSO tidak atau lebih dari nilai 0 maka dilakukan proses master PSO dengan partikel elite dari proses slave PSO menggunakan proses pada baris 10-14.
4. Baris 5 dan 12 merupakan proses PSO dalam optimasi parameter SVR.
5. Baris 6-7 dan 13 merupakan proses pengisian partikel berdasarkan proses PSO yang telah dilakukan.
6. Baris 8 dan 14 merupakan proses pengambilan partikel terbaik secara keseluruhan (*global*).

5.2.5 Implementasi Proses PSO

Proses PSO merupakan tahapan utama yang dilakukan oleh *slave* dan *master* PSO dalam optimasi parameter SVR. Proses ini berfungsi untuk melakukan proses optimasi parameter SVR. Implementasi proses PSO terdapat sub proses dipanggil melalui method yang ditunjukkan pada Kode Program 5.16.

Kode Program 5.16 Implementasi proses PSO

```

1 public void Optimasi() {
2     EvaluasiFitness(arrPartikel);
3     setGlobalBest(arrPartikel);
4     for (int i = 0; i <= iterPSO; i++) {
5         for (int j = 0; j < arrPartikel.size(); j++) {
6             hitungVKecepatan(arrPartikel.get(j),
7                 arrPartikel.get(gBest));
8             hitungXPosisi(arrPartikel.get(j));
9             setLocalBest(arrPartikel.get(j));
10            setGlobalBest(arrPartikel);
11        }
12        double persentase = suksesCount / arrPartikel.size();
13        weight = ((wmax - wmin) * persentase) + wmin;
14        suksesCount = 0;
15    }
16 }

```

Penjelasan Kode Program 5.16:

1. Baris 2 merupakan proses evaluasi *fitness* tiap partikel.
2. Baris 3 dan 10 merupakan proses penentuan *global best*.
3. Secara menyeluruh baris 4-15 merupakan proses optimasi mengacu pada algoritma PSO seperti pada Gambar 2.19 yang telah dijelaskan sebelumnya.
4. Baris 12-14 merupakan proses adaptasi terhadap nilai bobot inersia menggunakan persamaan (2.55) berdasarkan tingkat kesuksesan partikel dalam pencarian solusi optimum.

5.2.5.1 Implementasi Proses Evaluasi *Fitness* Tiap Partikel

Proses evaluasi *fitness* tiap partikel merupakan bagian tahap proses PSO dalam optimasi parameter SVR. Proses ini berfungsi untuk mengetahui bagaimana baik penerapan konfigurasi parameter SVR yang dihasilkan oleh nilai RMSE dan dikonversi menjadi nilai *fitness* partikel. Implementasi proses evaluasi *fitness* tiap partikel ditunjukkan pada Kode Program 5.17.

Kode Program 5.17 Implementasi proses evaluasi *fitness* tiap partikel

```

1 public void EvaluasiFitness(List<Partikel> arrPartikel) {
2     for (int i = 0; i < arrPartikel.size(); i++) {
3         arrPartikel.get(i).fitness =
4         cekFitness(arrPartikel.get(i));
5     }
6 }

```

Penjelasan Kode Program 5.17:

1. Baris 2-4 merupakan sub proses method lain untuk melakukan perhitungan evaluasi *fitness* tiap partikel sesuai banyaknya partikel.

5.2.5.2 Implementasi Proses Penentuan *Global Best*

Proses penentuan *global best* merupakan bagian tahap proses PSO dalam optimasi parameter SVR. Proses ini berfungsi untuk mengetahui partikel terbaik dalam pencarian solusi dalam mengkonfigurasi parameter SVR berdasarkan nilai *fitness*. Implementasi proses penentuan *global best* ditunjukkan pada Kode Program 5.18.

Kode Program 5.18 Implementasi proses penentuan *global best*

```

1 public void setGlobalBest(List<Partikel> arrPartikel) {
2     double awal = arrPartikel.get(0).fitness;
3     for (int j = 0; j < arrPartikel.size(); j++) {
4         if (arrPartikel.get(j).fitness > awal) {
5             awal = arrPartikel.get(j).fitness;
6             gBest = j;
7         }
8     }
9 }

```

Penjelasan Kode Program 5.18:

1. Baris 2 merupakan proses pengambilan nilai *fitness* partikel pada posisi awal.
2. Baris 3-8 merupakan proses perbandingan *fitness* partikel diposisi awal dan selanjutnya.
3. Baris 5-6 merupakan proses pencatatan posisi partikel dengan *fitness* tertinggi.

5.2.5.3 Implementasi Proses Perhitungan Kecepatan Partikel

Proses perhitungan kecepatan partikel merupakan bagian tahap proses PSO dalam optimasi parameter SVR. Proses ini berfungsi untuk mengatur kecepatan partikel dalam pencarian solusi optimum parameter SVR berdasarkan nilai *fitness*. Implementasi proses perhitungan kecepatan partikel secara umum ditunjukkan pada Kode Program 5.19.

Kode Program 5.19 Implementasi proses perhitungan kecepatan partikel pada dimensi y

```

1 public void hitungVKecepatan(Partikel partikel, Partikel acuan) {
2     partikel.vGamma = weight * partikel.vGamma + c1 * r1 *
3     (partikel.gamma - partikel.xGamma) + c2 * r2 * (acuan.xGamma -
4     partikel.xGamma);
5     if (partikel.vGamma < Delta * (gammaMin - gammaMax)) {
6         partikel.vGamma = Delta * (gammaMin - gammaMax);
7     } else if (partikel.vGamma > Delta * (gammaMax -
8     gammaMin)) {
9         partikel.vGamma = Delta * (gammaMax - gammaMin);
10    }
11 }

```

Penjelasan Kode Program 5.19:

1. Baris 2-4 merupakan proses perhitungan kecepatan partikel menggunakan persamaan (2.54).
2. Baris 5-10 merupakan proses pembatasan kecepatan partikel menggunakan pengkondisian persamaan (2.51).

5.2.5.4 Implementasi Proses Perhitungan Posisi Partikel

Proses perhitungan posisi partikel merupakan bagian tahap proses PSO dalam optimasi parameter SVR. Proses ini berfungsi untuk mengatur partikel partikel dalam pencarian solusi optimum parameter SVR berdasarkan nilai *fitness*. Implementasi proses perhitungan partikel partikel secara umum ditunjukkan pada Kode Program 5.20.

Kode Program 5.20 Implementasi proses perhitungan posisi partikel

```

1 public void hitungXPosisi(Partikel partikel) {
2     partikel.xGamma = partikel.xGamma + partikel.vGamma;
3     if (partikel.xGamma < gammaMin) {
4         partikel.xGamma = gammaMin + ((-
5 ThreadLocalRandom.current().nextDouble(1)) * partikel.vGamma);
6     } else if (partikel.xGamma > gammaMax) {
7         partikel.xGamma = gammaMax + ((-
8 ThreadLocalRandom.current().nextDouble(1)) * partikel.vGamma);
9     }
10 }

```

Penjelasan Kode Program 5.20:

1. Baris 2 merupakan proses perhitungan posisi partikel menggunakan persamaan (2.50).
2. Baris 3-9 merupakan proses pembatasan posisi partikel menggunakan pengkondisian persamaan (2.62).

5.2.5.5 Implementasi Proses Perhitungan *Local Best*

Proses perhitungan *local best* merupakan bagian tahap proses PSO dalam optimasi parameter SVR. Proses ini berfungsi untuk menentukan apakah diperlukan perpindahan posisi atau tetap berdasarkan nilai *fitness*. Implementasi proses perhitungan *local best* secara umum ditunjukkan pada Kode Program 5.21.

Kode Program 5.21 Implementasi proses perhitungan *local best*

```

1 public void setLocalBest(Partikel partikel) {
2     Partikel XposisiBaru = new Partikel(partikel.xGamma,
3 partikel.xLambda, partikel.xComplexity, partikel.xEpsilon,
4 partikel.xGammaAnova, Math.round(partikel.xDegreeAnova));
5     double fitnessXposisi = cekFitness(XposisiBaru);
6     if (fitnessXposisi > partikel.fitness) {
7         suksesCount = suksesCount + 1;
8         partikel.gamma = XposisiBaru.gamma;
9         partikel.lambda = XposisiBaru.lambda;
10        partikel.complexity = XposisiBaru.complexity;
11        partikel.epsilon = XposisiBaru.epsilon;
12        partikel.gammaAnova = XposisiBaru.gammaAnova;
13        partikel.degreeAnova =
14 Math.round(XposisiBaru.degreeAnova);
15        partikel.fitness = fitnessXposisi;
16    }
17 }

```

Penjelasan Kode Program 5.21:

1. Baris 2-5 merupakan proses perhitungan *fitness* partikel.
2. Baris 6-16 secara umum merupakan proses penentuan individu *personal / local best* berdasarkan pengkondisian pada persamaan (2.48).
3. Baris 7 merupakan proses pengkondisian apabila tercatat ada perubahan atau perkembangan terhadap *local best*, maka dilakukan pencatatan atau penambahan poin kesuksesan dalam pencarian solusi seperti pada persamaan (2.56).
4. Baris 8-15 merupakan proses perubahan posisi berdasarkan kondisi *local best*.

5.2.5.6 Implementasi Proses Perhitungan *Fitness* Partikel

Proses perhitungan *fitness* partikel merupakan bagian tahap proses PSO dalam optimasi parameter SVR. Proses ini berfungsi sebagai acuan individu saat pencarian solusi optimum. Implementasi proses perhitungan *local best* secara umum ditunjukkan pada Kode Program 5.22.

Kode Program 5.22 Implementasi proses perhitungan *fitness* partikel

```

1 public double cekFitness(Partikel partikel) {
2     double rmse = svr.getLatih(partikel.gamma,
3     partikel.lambda, partikel.complexity, partikel.epsilon,
4     partikel.gammaAnova, Math.round(partikel.degreeAnova));
5     double fitness = 1 / (1 + rmse);
6     return fitness;
7 }

```

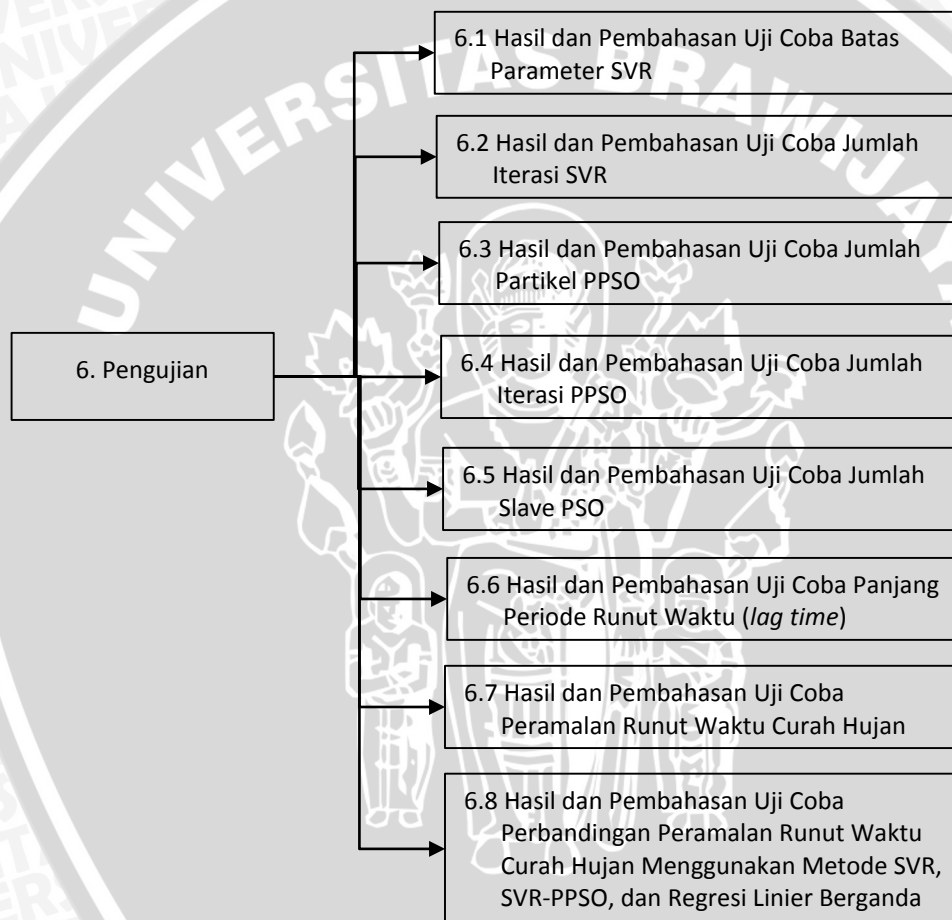
Penjelasan Kode Program 5.22:

1. Baris 2-4 merupakan proses pelatihan SVR, sehingga menghasilkan nilai RMSE.
2. Baris 5 merupakan perhitungan *fitness* berdasarkan nilai RMSE yang dihasilkan pelatihan SVR menggunakan persamaan (2.57).



BAB 6 PENGUJIAN

Pada bab ini merupakan pembahasan mengenai hasil uji coba terhadap sistem perangkat lunak berbasis *prototype* yang dapat melakukan peramalan runut waktu curah hujan menggunakan metode *Support Vector Regression* dan *Parallel Particle Swarm Optimization* (SVR-PPSO). Pada proses uji coba dilakukan berdasarkan perancangan skenario pengujian pada sub bab 4.4 dengan bantuan sistem komputer yang telah dijelaskan pada sub 3.2.1. Berikut merupakan gambaran secara umum tahap pengujian dalam bentuk diagram struktur yang ditunjukkan pada Gambar 6.1.



Gambar 6.1 Struktur Pengujian

6.1 Hasil dan Pembahasan Uji Coba Batas Parameter SVR

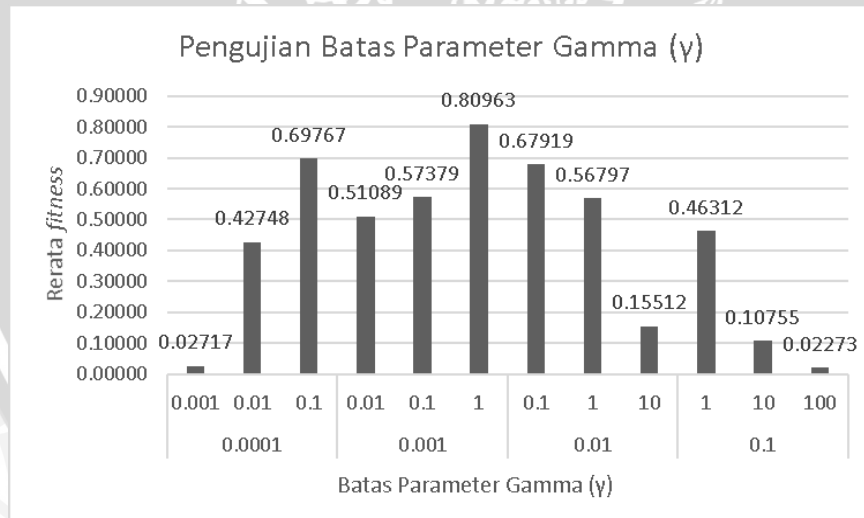
Uji coba batas parameter SVR ditujukan untuk membatasi dimensi partikel dalam pencarian solusi agar dapat menghasilkan kombinasi parameter SVR yang optimal pada proses pelatihan. Pengujian batas parameter SVR diantaranya γ , λ , c , ϵ , γ_k kernel, dan d kernel. Uji coba batas parameter SVR dilakukan pada proses pelatihan SVR menggunakan data runut waktu curah hujan dasarian 1 bulan januari 2000 sampai dasarian 3 bulan desember 2001 pola data horizontal.

6.1.1 Hasil dan Pembahasan Uji Coba Batas Parameter γ

Uji coba awal dilakukan pada batas parameter γ berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba batas parameter γ yang dilakukan dengan batasan nilai 0.0001 - 100. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba batas parameter γ .

- | | |
|---|----------------|
| a. Panjang periode (<i>lag time</i>) | : 18 |
| b. Ramalan (data uji) | : 3 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |
| f. Jumlah <i>slave</i> PSO | : 2 |
| g. Batas parameter γ | : 0.0001 - 100 |
| h. Batas parameter λ | : 0 - 5 |
| i. Batas parameter c | : 0.001 - 1000 |
| j. Batas parameter ε | : 0.0001 - 100 |
| k. Batas parameter γ_k <i>kernel</i> | : 0.0001 - 100 |
| l. Batas parameter d <i>kernel</i> | : 1 - 3 |

Pengujian batas parameter γ dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian. Pada lampiran B.1 merupakan hasil uji coba batas parameter γ secara detail, sedangkan pada Gambar 6.2 merupakan hasil uji coba batas parameter γ yang direpresentasikan dalam bentuk grafik.



Gambar 6.2 Grafik hasil uji coba batas parameter γ

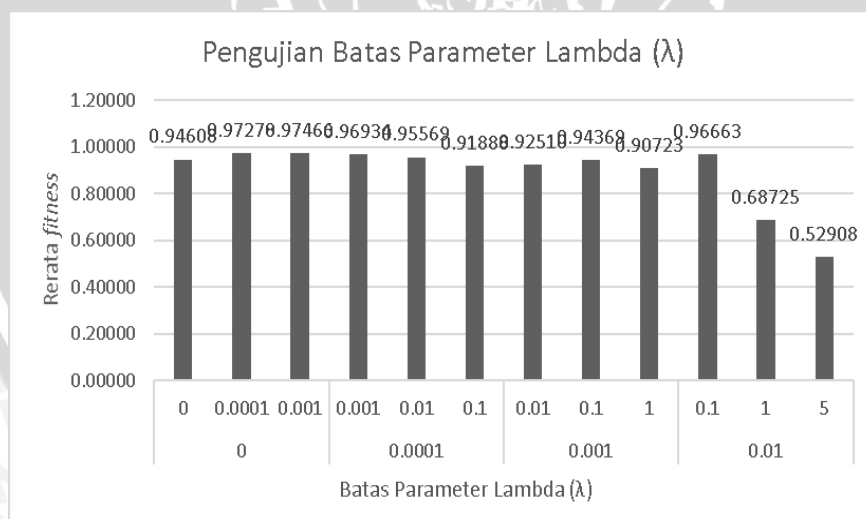
Pada Gambar 6.2 dapat diketahui batas parameter γ terbaik antara 0.001 – 1 berdasarkan nilai *fitness* tertinggi yang bernilai 0.80963. Besaran nilai parameter γ akan mempengaruhi tingkat konvergen pada proses pelatihan SVR (Vijayakumar & Wu, 1999). Hal tersebut terbukti berdasarkan hasil uji coba yang telah dilakukan dengan menggunakan batas parameter γ terlalu besar dan terlalu kecil mempengaruhi nilai *fitness*.

6.1.2 Hasil dan Pembahasan Uji Coba Batas Parameter λ

Uji coba batas parameter λ berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba batas parameter λ yang dilakukan dengan batasan nilai 0 - 5. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba batas parameter λ .

- | | |
|---|----------------|
| a. Panjang periode (<i>lag time</i>) | : 18 |
| b. Ramalan (data uji) | : 3 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |
| f. Jumlah <i>slave</i> PSO | : 2 |
| g. Batas parameter γ | : 0.001 - 1 |
| h. Batas parameter λ | : 0 - 5 |
| i. Batas parameter c | : 0.001 - 1000 |
| j. Batas parameter ε | : 0.0001 - 100 |
| k. Batas parameter γ_k <i>kernel</i> | : 0.0001 - 100 |
| l. Batas parameter d <i>kernel</i> | : 1 - 3 |

Pengujian batas parameter λ dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter γ yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian. Pada lampiran B.2 merupakan hasil uji coba batas parameter γ secara detail, sedangkan pada Gambar 6.3 merupakan hasil uji coba batas parameter λ yang direpresentasikan dalam bentuk grafik.



Gambar 6.3 Grafik hasil uji coba batas parameter λ

Pada Gambar 6.3 dapat diketahui batas parameter λ terbaik dengan batasan 0 - 0, 0 - 0.0001, 0 - 0.001, 0.0001 - 0.001, 0.0001 - 0.01, 0.0001 - 0.1, 0.001 - 0.01, 0.001 - 0.1, 0.001 - 1, dan 0.01 - 0.1. Secara umum batas parameter λ terbaik dapat diperoleh antara nilai 0 - 0.1 yang memiliki rerata nilai *fitness* di atas 0.9. Besaran nilai parameter λ akan mempengaruhi kecepatan konvergen dan ketidakstabilan pada proses pelatihan SVR (Vijayakumar & Wu, 1999). Hal tersebut

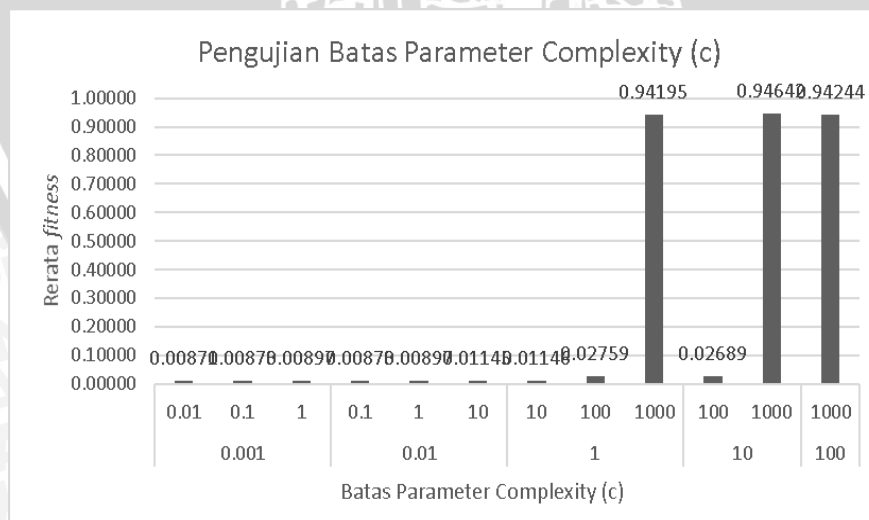
terbukti berdasarkan hasil uji coba yang telah dilakukan dengan menggunakan batas parameter λ yang terlalu besar mempengaruhi nilai *fitness* yang kurang baik.

6.1.3 Hasil dan Pembahasan Uji Coba Batas Parameter C

Uji coba batas parameter C berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba batas parameter C yang dilakukan dengan batasan nilai 0.001 - 1000. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba batas parameter C.

- | | |
|---|----------------|
| a. Panjang periode (<i>lag time</i>) | : 18 |
| b. Ramalan (data uji) | : 3 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |
| f. Jumlah <i>slave</i> PSO | : 2 |
| g. Batas parameter γ | : 0.001 - 1 |
| h. Batas parameter λ | : 0 - 0.1 |
| i. Batas parameter c | : 0.001 - 1000 |
| j. Batas parameter ε | : 0.0001 - 100 |
| k. Batas parameter γ_k <i>kernel</i> | : 0.0001 - 100 |
| l. Batas parameter d <i>kernel</i> | : 1 - 3 |

Pengujian batas parameter C dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter γ dan λ yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian. Pada lampiran B.3 merupakan hasil uji coba batas C secara detail, sedangkan pada Gambar 6.4 merupakan hasil uji coba batas parameter λ yang direpresentasikan dalam bentuk grafik.



Gambar 6.4 Grafik hasil uji coba batas parameter C

Pada Gambar 6.4 dapat diketahui batas parameter C terbaik dengan batasan 1 - 1000, 10 - 1000, dan 100 - 1000. Secara umum batas parameter C terbaik

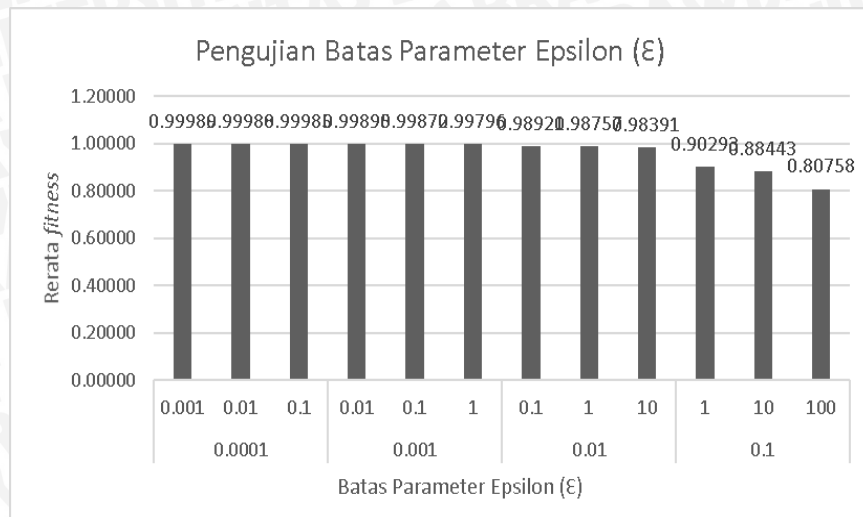
diperoleh antara nilai 1 – 1000 yang memiliki rerata nilai *fitness* di atas 0.9. Besaran nilai parameter C berfungsi sebagai variabel penting pembatas kompleksitas dalam pencarian fungsi f yang rata (*flatness*) (Smola & Schölkopf, 2004). Hal tersebut terbukti berdasarkan hasil uji coba yang telah dilakukan dengan menggunakan batas parameter C yang terlalu kecil mempengaruhi nilai *fitness* yang kurang baik.

6.1.4 Hasil dan Pembahasan Uji Coba Batas Parameter ϵ

Uji coba batas parameter ϵ berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba batas parameter ϵ yang dilakukan dengan batasan nilai 0.0001 - 100. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba batas parameter ϵ .

- | | |
|---|----------------|
| a. Panjang periode (<i>lag time</i>) | : 18 |
| b. Ramalan (data uji) | : 3 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |
| f. Jumlah <i>slave</i> PSO | : 2 |
| g. Batas parameter γ | : 0.001 - 1 |
| h. Batas parameter λ | : 0 - 0.1 |
| i. Batas parameter c | : 1 - 1000 |
| j. Batas parameter ϵ | : 0.0001 - 100 |
| k. Batas parameter γ_k <i>kernel</i> | : 0.0001 - 100 |
| l. Batas parameter d <i>kernel</i> | : 1 - 3 |

Pengujian batas parameter ϵ dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter C , γ , dan λ yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian. Pada lampiran B.4 merupakan hasil uji coba batas ϵ secara detail, sedangkan pada Gambar 6.5 merupakan hasil uji coba batas parameter ϵ yang direpresentasikan dalam bentuk grafik.



Gambar 6.5 Grafik hasil uji coba batas parameter ϵ

Pada Gambar 6.5 dapat diketahui batas parameter ϵ terbaik dengan batasan 0.0001 – 0.001, 0.0001 – 0.01, 0.0001 – 0.1, 0.001 – 0.01, 0.001 – 0.1, 0.001 – 1, 0.01 – 0.1, 0.01 – 0.1, 0.01 – 1, 0.01 – 10, dan 0.1 - 1. Namun, pada batasan 0.01 – 1, 0.01 – 10, dan 0.1 – 1 cenderung mendekati nilai batas minimal. Oleh karena itu, dapat digunakan batas parameter ϵ secara umum antara nilai 0.0001 – 0.1 yang sudah cukup memiliki rerata nilai *fitness* di atas 0.9. Besaran nilai parameter ϵ berfungsi sebagai batas toleransi *support vector* yang rata (*flatness*) seperti pada Gambar 2.12. Apabila nilai parameter ϵ terlalu besar dapat mempengaruhi tingkat kompleksitas terhadap fungsi yang direpresentasikan oleh *support vector* (Smola & Schölkopf, 2004). Selain itu, nilai parameter ϵ menjadi acuan konvergensi dengan syarat nilai $\max(|\delta a_i^*|) < \epsilon$ dan $\max(|\delta a_i|) < \epsilon$ selama proses pelatihan SVR. Hal tersebut terbukti berdasarkan hasil uji coba yang telah dilakukan dengan menggunakan batas parameter ϵ yang terlalu besar mempengaruhi nilai *fitness* yang kurang baik.

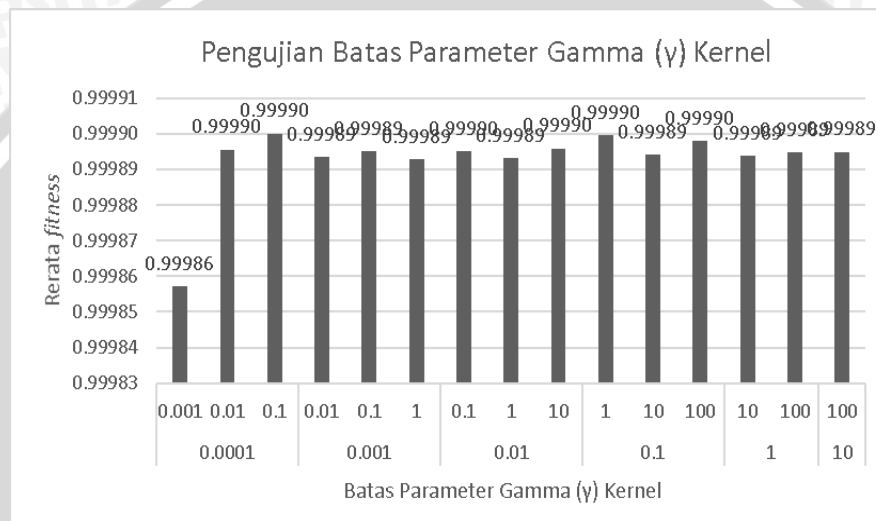
6.1.5 Hasil dan Pembahasan Uji Coba Batas Parameter γ *kernel*

Uji coba batas parameter γ *kernel* berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba batas parameter γ *kernel* yang dilakukan dengan batasan nilai 0.0001 - 100. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba batas parameter γ *kernel*.

- | | |
|--|-------------|
| a. Panjang periode (<i>lag time</i>) | : 18 |
| b. Ramalan (data uji) | : 3 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |
| f. Jumlah <i>slave</i> PSO | : 2 |
| g. Batas parameter γ | : 0.001 - 1 |
| h. Batas parameter λ | : 0 – 0.1 |
| i. Batas parameter c | : 1 - 1000 |

- j. Batas parameter ε : 0.0001 – 0.1
- k. Batas parameter γ *kernel* : 0.0001 - 100
- l. Batas parameter d *kernel* : 1 - 3

Pengujian batas parameter γ *kernel* dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter ε , c , γ , dan λ yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian. Pada lampiran B.5 merupakan hasil uji coba batas γ *kernel* secara detail, sedangkan pada Gambar 6.6 merupakan hasil uji coba batas parameter γ *kernel* yang direpresentasikan dalam bentuk grafik.



Gambar 6.6 Grafik hasil uji coba batas parameter γ *kernel*

Pada Gambar 6.6 dapat diketahui batas parameter γ *kernel* dapat menggunakan banyak variasi kombinasi batas. Namun, secara umum apabila menggunakan batasan nilai 0.0001 - 100 cenderung mendekati nilai batas maksimal. Oleh karena itu, dapat digunakan batas parameter γ *kernel* secara umum antara nilai 0.0001 – 0.1 atau 1 – 100 yang sudah cukup memiliki rerata nilai *fitness* di atas 0.9. Besaran nilai parameter γ *kernel* dapat mempengaruhi kinerja ANOVA RBF *kernel* dalam melakukan proses pemetaan pada ruang fitur. Namun, berdasarkan uji coba dapat menggunakan banyak variasi kombinasi batas parameter γ *kernel* dengan baik pada proses pelatihan SVR. Oleh sebab itu, perlu dilakukan uji coba lanjut pengaruh batas parameter γ *kernel* terhadap hasil peramalan pada proses pengujian SVR.

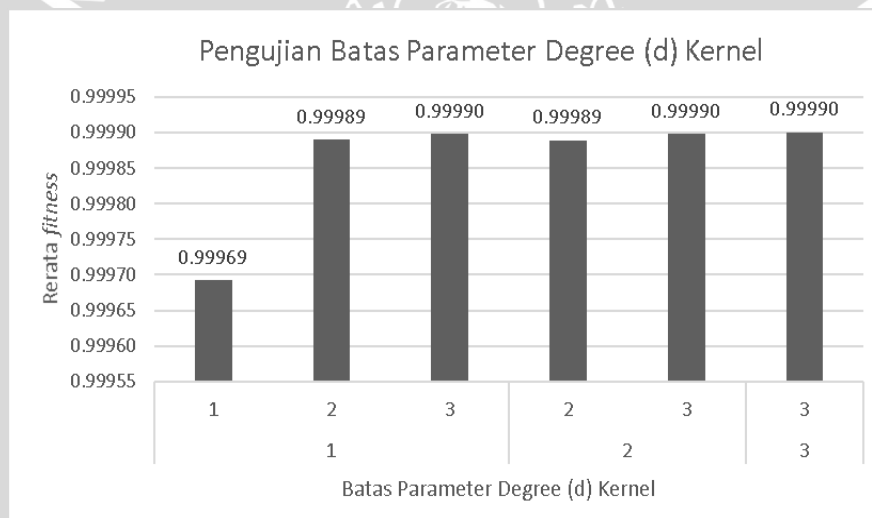
6.1.6 Hasil dan Pembahasan Uji Coba Batas Parameter d *Kernel*

Uji coba batas parameter d *kernel* berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba batas parameter d *kernel* yang dilakukan dengan batasan nilai 1 – 3. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba batas parameter d *kernel*.

- a. Panjang periode (*lag time*) : 18
- b. Ramalan (data uji) : 3

- c. Jumlah iterasi SVR : 1000
- d. Jumlah Partikel PPSO : 10
- e. Maksimal iterasi PPSO : 100
- f. Jumlah *slave* PSO : 2
- g. Batas parameter γ : 0.001 - 1
- h. Batas parameter λ : 0 - 0.1
- i. Batas parameter c : 1 - 1000
- j. Batas parameter ε : 0.0001 - 0.1
- k. Batas parameter γ_k *kernel* : 0.0001 - 0.1
- l. Batas parameter d *kernel* : 1 - 3

Pengujian batas parameter d *kernel* dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter γ_k *kernel*, ε , c , γ , dan λ yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian. Pada lampiran B.6 merupakan hasil uji coba batas d *kernel* secara detail, sedangkan pada Gambar 6.7 merupakan hasil uji coba batas parameter d *kernel* yang direpresentasikan dalam bentuk grafik.



Gambar 6.7 Grafik hasil uji coba batas parameter d *kernel*

Pada Gambar 6.7 dapat diketahui batas parameter d *kernel* dapat menggunakan kombinasi batas secara umum antara 1 - 3 yang memiliki rerata nilai *fitness* di atas 0.9. Besaran nilai parameter d *kernel* dapat mempengaruhi kinerja ANOVA RBF *kernel* dalam melakukan proses pemetaan pada ruang fitur. Namun, berdasarkan uji coba dapat menggunakan kombinasi batas parameter d *kernel* secara umum antara 1 - 3 dengan baik pada proses pelatihan SVR. Oleh sebab itu, perlu dilakukan uji coba lanjut batas parameter d *kernel* secara detail terhadap hasil peramalan pada proses pengujian SVR.

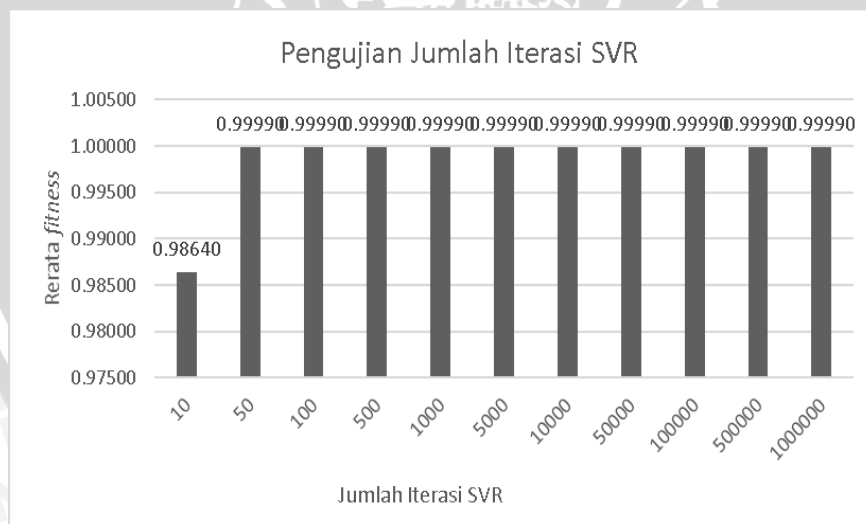
6.2 Hasil dan Pembahasan Uji Coba Jumlah Iterasi SVR

Uji coba jumlah iterasi SVR berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba jumlah iterasi SVR yang dilakukan sebanyak

10 - 1000000 pada proses pelatihan SVR menggunakan data runut waktu curah hujan dasarian 1 bulan januari 2000 sampai dasarian 3 bulan desember 2001 pola data horizontal. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba jumlah iterasi SVR.

- | | |
|---|----------------|
| a. Panjang periode (<i>lag time</i>) | : 18 |
| b. Ramalan (data uji) | : 3 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |
| f. Jumlah <i>slave</i> PSO | : 2 |
| g. Batas parameter γ | : 0.001 - 1 |
| h. Batas parameter λ | : 0 - 0.1 |
| i. Batas parameter c | : 1 - 1000 |
| j. Batas parameter ϵ | : 0.0001 - 0.1 |
| k. Batas parameter γ_k <i>kernel</i> | : 0.0001 - 0.1 |
| l. Batas parameter d <i>kernel</i> | : 1 - 3 |

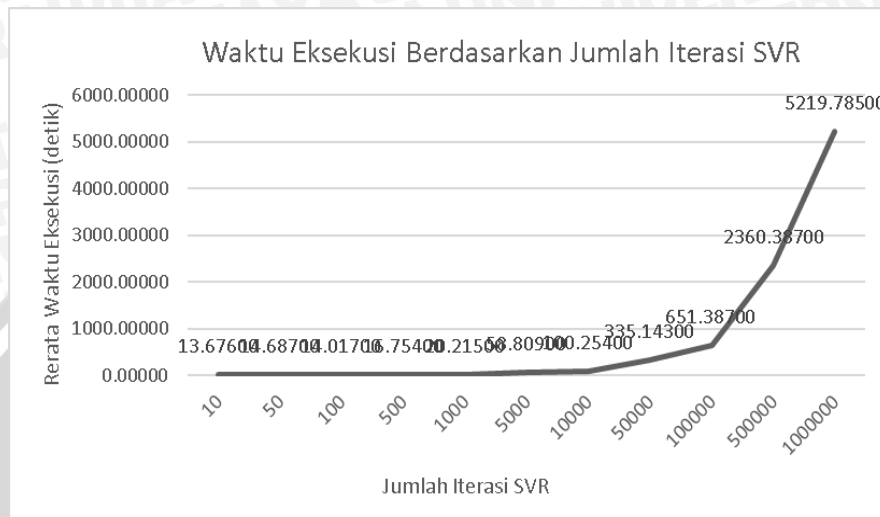
Pengujian jumlah iterasi SVR dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter SVR dan konfigurasi SVR-PPSO yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian. Pada lampiran B.7 merupakan hasil uji coba jumlah iterasi SVR secara detail, sedangkan pada Gambar 6.8 merupakan hasil uji coba jumlah iterasi SVR yang direpresentasikan dalam bentuk grafik.



Gambar 6.8 Grafik hasil uji coba jumlah iterasi SVR

Pada Gambar 6.8 dapat diketahui jumlah iterasi SVR dapat menggunakan jumlah iterasi sebesar 10 sudah memiliki rerata nilai *fitness* di atas 0.9. Hal tersebut terjadi dikarenakan PSO yang memaksakan parameter γ bernilai besar untuk mempercepat konvergensi ini hanya dengan 10 iterasi. Namun, dengan menggunakan jumlah iterasi SVR rendah belum tentu mendapatkan proses pelatihan SVR yang optimal. Jumlah iterasi SVR sebenarnya tidak terlalu

mempengaruhi nilai *fitness* karena konvergensi diutamakan mengacu pada nilai parameter ε dengan syarat nilai $\max(|\delta a_i^*|) < \varepsilon$ dan $\max(|\delta a_i|) < \varepsilon$ selama proses pelatihan SVR. Namun, besarnya jumlah iterasi SVR akan berpengaruh terhadap waktu komputasi proses pelatihan SVR yang dapat divisualisasikan secara grafik pada Gambar 6.9.



Gambar 6.9 Grafik waktu eksekusi berdasarkan jumlah iterasi SVR

Pada Gambar 6.9 dapat diketahui jumlah iterasi SVR yang digunakan akan mempengaruhi waktu eksekusi yang lebih lama. Meskipun menggunakan jumlah iterasi SVR yang besar akan tetap memiliki nilai *fitness* tinggi. Oleh sebab itu, penentuan jumlah iterasi SVR diperlukan batasan hemat waktu agar tidak terlalu lama memakan waktu dalam proses pelatihan SVR. Dalam penentuan jumlah iterasi SVR dapat menggunakan iterasi sebesar 1000 yang sudah cukup mendapatkan nilai *fitness* tinggi dan tidak terlalu lama memakan waktu eksekusi pada proses pelatihan SVR.

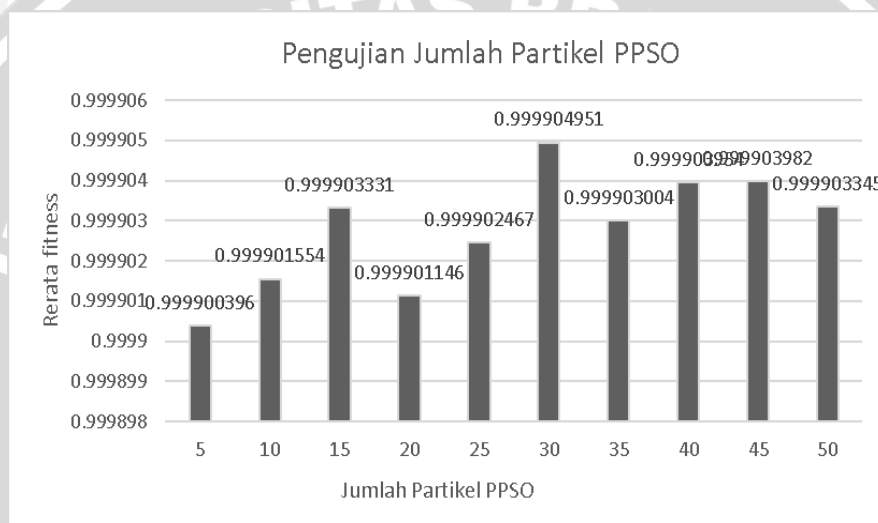
6.3 Hasil dan Pembahasan Uji Coba Jumlah Partikel PPSO

Uji coba jumlah partikel PPSO berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba jumlah partikel PPSO yang dilakukan berkelipatan 5 yaitu 5, 10, 15, 20, 25, 30, 35, 40, 45, dan 50 pada proses pelatihan SVR menggunakan data runut waktu curah hujan dasarian 1 bulan januari 2000 sampai dasarian 3 bulan desember 2001 pola data horizontal. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba jumlah partikel PPSO.

- | | |
|--|-------------|
| a. Panjang periode (<i>lag time</i>) | : 18 |
| b. Ramalan (data uji) | : 3 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |
| f. Jumlah <i>slave</i> PSO | : 2 |
| g. Batas parameter γ | : 0.001 - 1 |

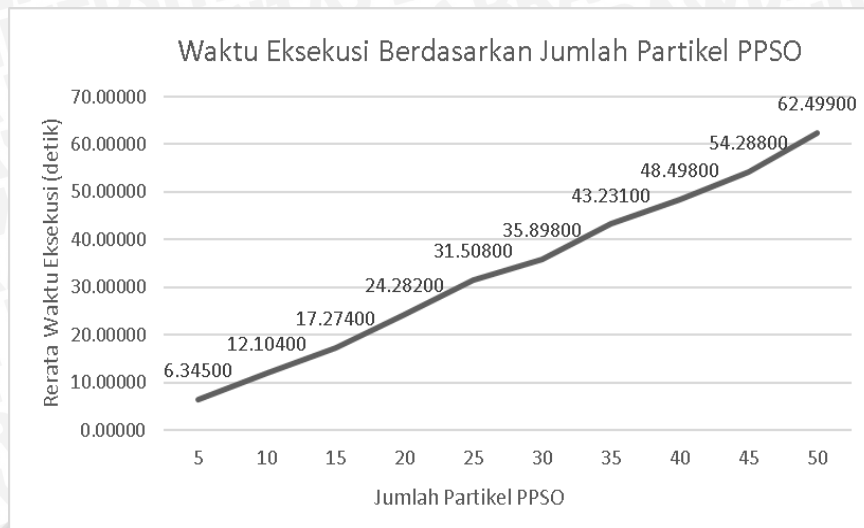
- h. Batas parameter λ : 0 – 0.1
- i. Batas parameter c : 1 - 1000
- j. Batas parameter ε : 0.0001 – 0.1
- k. Batas parameter γ_k *kernel* : 0.0001 – 0.1
- l. Batas parameter d *kernel* : 1 - 3

Pengujian jumlah partikel PPSO dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter SVR dan konfigurasi SVR-PPSO yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian. Pada lampiran B.8 merupakan hasil uji coba jumlah partikel PPSO secara detail, sedangkan pada Gambar 6.10 merupakan hasil uji coba jumlah partikel PPSO yang direpresentasikan dalam bentuk grafik.



Gambar 6.10 Grafik hasil uji coba jumlah partikel PPSO

Pada Gambar 6.10 dapat diketahui penggunaan jumlah partikel PPSO sebesar 5 partikel sudah cukup dapat memiliki rerata nilai *fitness* di atas 0.9. Semakin banyak partikel, maka semakin banyak peluang untuk dapat melakukan pencarian solusi dalam optimasi parameter SVR pada proses pelatihan. Namun, semakin banyak jumlah partikel akan mempengaruhi waktu eksekusi yang dapat divisualisasikan secara grafik pada Gambar 6.11.



Gambar 6.11 Grafik waktu eksekusi berdasarkan jumlah partikel PPSO

Pada Gambar 6.11 dapat diketahui besarnya jumlah partikel PPSO akan mempengaruhi waktu eksekusi proses optimasi parameter SVR. Dengan demikian, diperlukan penentuan jumlah partikel PPSO sesuai kebutuhan dan keterbatasan sumber daya.

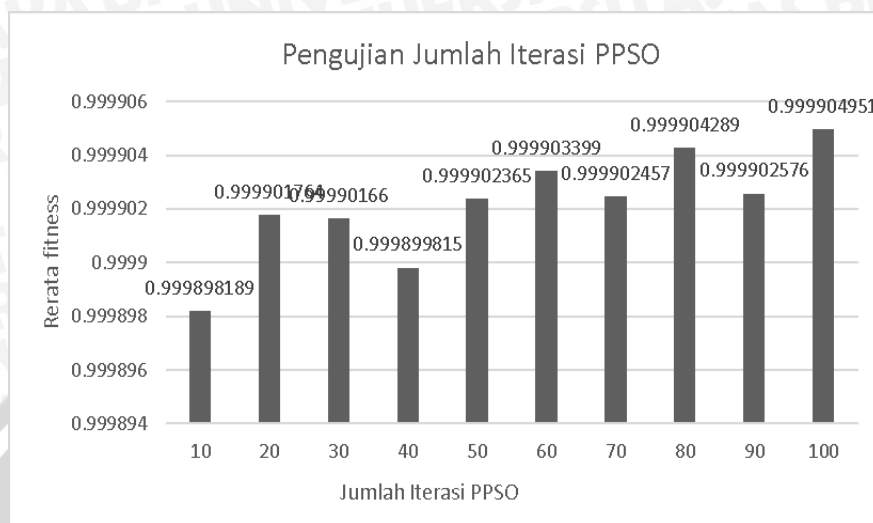
6.4 Hasil dan Pembahasan Uji Coba Jumlah Iterasi PPSO

Uji coba jumlah iterasi PPSO berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba jumlah iterasi PPSO yang dilakukan berkelipatan 10 yaitu 10, 20, 30, 40, 50, 60, 70, 80, 90, dan 100 pada proses pelatihan SVR menggunakan data runut waktu curah hujan dasarian 1 bulan januari 2000 sampai dasarian 3 bulan desember 2001 pola data horizontal. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba jumlah iterasi PPSO.

- | | |
|---|----------------|
| a. Panjang periode (<i>lag time</i>) | : 18 |
| b. Ramalan (data uji) | : 3 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |
| f. Jumlah <i>slave</i> PSO | : 2 |
| g. Batas parameter γ | : 0.001 - 1 |
| h. Batas parameter λ | : 0 - 0.1 |
| i. Batas parameter c | : 1 - 1000 |
| j. Batas parameter ε | : 0.0001 - 0.1 |
| k. Batas parameter γ_k <i>kernel</i> | : 0.0001 - 0.1 |
| l. Batas parameter d <i>kernel</i> | : 1 - 3 |

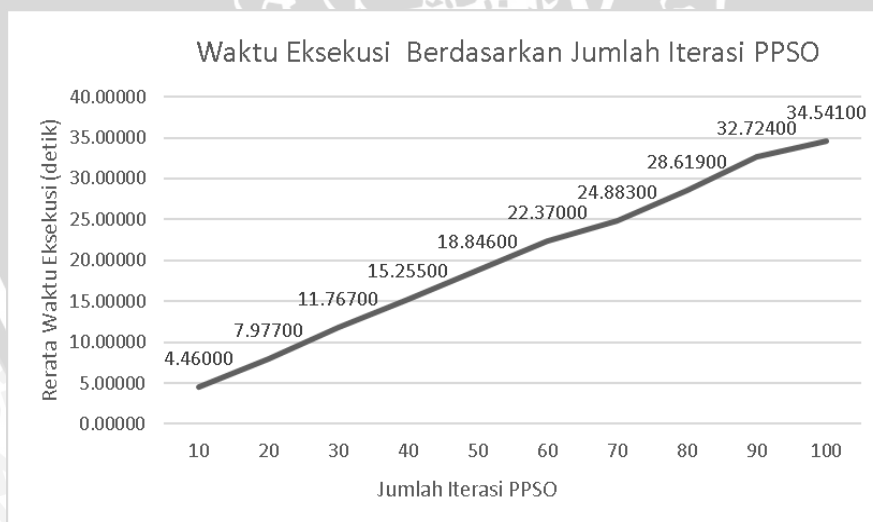
Pengujian jumlah iterasi PPSO dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter SVR dan konfigurasi SVR-PPSO yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian.

Pada lampiran B.9 merupakan hasil uji coba jumlah iterasi PPSO secara detail, sedangkan pada Gambar 6.12 merupakan hasil uji coba jumlah iterasi PPSO yang direpresentasikan dalam bentuk grafik.



Gambar 6.12 Grafik hasil uji coba jumlah iterasi PPSO

Pada Gambar 6.12 dapat diketahui jumlah iterasi PPSO dapat menggunakan jumlah iterasi sebesar 10 sudah memiliki rerata nilai *fitness* di atas 0.9. Hal tersebut terjadi dikarenakan PSO yang memaksakan parameter γ bernilai besar untuk mempercepat konvergensi dini pelatihan SVR. Namun, semakin banyak jumlah iterasi PPSO akan mempengaruhi waktu eksekusi yang dapat divisualisasikan secara grafik pada Gambar 6.13.



Gambar 6.13 Grafik waktu eksekusi berdasarkan jumlah iterasi PPSO

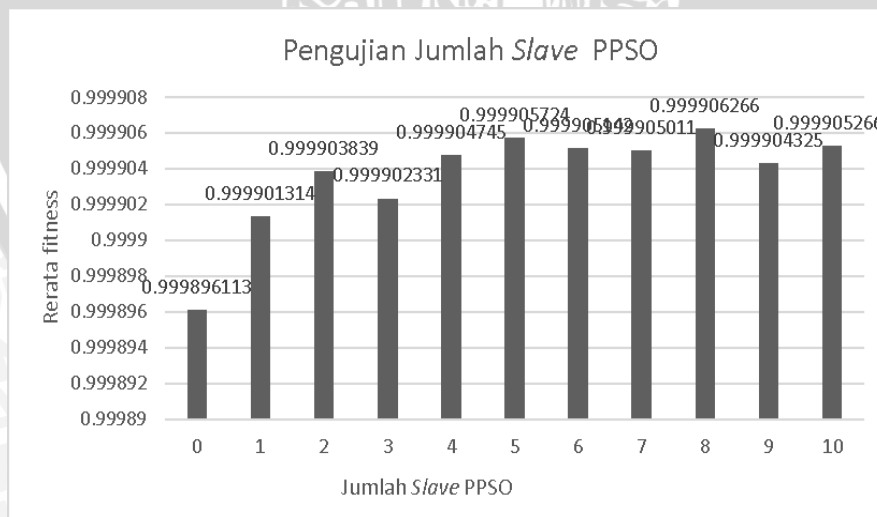
Pada Gambar 6.13 dapat diketahui besarnya jumlah iterasi PPSO akan mempengaruhi waktu eksekusi proses optimasi parameter SVR. Dengan demikian, diperlukan penentuan jumlah iterasi PPSO sesuai kebutuhan dan keterbatasan sumber daya.

6.5 Hasil dan Pembahasan Uji Coba Jumlah *Slave* PSO

Uji coba jumlah *slave* PSO berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba jumlah *slave* PSO yang dilakukan tanpa *slave* PSO atau 0 sampai dengan 10 *slave* PSO pada proses pelatihan SVR menggunakan data runut waktu curah hujan dasarian 1 bulan januari 2000 sampai dasarian 3 bulan desember 2001 pola data horizontal. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba jumlah *slave* PSO.

- a. Panjang periode (*lag time*) : 18
- b. Ramalan (data uji) : 3
- c. Jumlah iterasi SVR : 1000
- d. Jumlah Partikel PPSO : 10
- e. Maksimal iterasi PPSO : 100
- f. Jumlah *slave* PSO : 2
- g. Batas parameter γ : 0.001 - 1
- h. Batas parameter λ : 0 - 0.1
- i. Batas parameter c : 1 - 1000
- j. Batas parameter ε : 0.0001 - 0.1
- k. Batas parameter γ_k *kernel* : 0.0001 - 0.1
- l. Batas parameter d *kernel* : 1 - 3

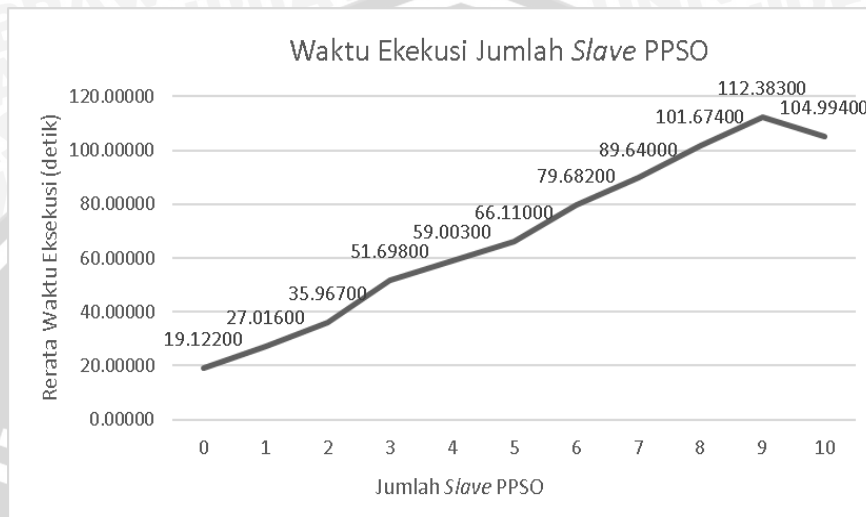
Pengujian jumlah *slave* PSO dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter SVR dan konfigurasi SVR-PPSO yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian. Pada lampiran B.10 merupakan hasil uji coba jumlah *slave* PSO secara detail, sedangkan pada Gambar 6.14 merupakan hasil uji coba jumlah *slave* PSO yang direpresentasikan dalam bentuk grafik.



Gambar 6.14 Grafik hasil uji coba jumlah *slave* PSO

Pada Gambar 6.14 dapat diketahui jumlah *slave* PSO lebih didominasi di atas nilai 2 untuk mendapatkan rerata nilai *fitness* di atas 0.9. Namun, secara umum

penggunaan tanpa *slave* PSO juga sudah cukup memiliki rerata nilai *fitness* yang baik di atas 0.9. *Slave* PSO bertujuan untuk dapat melakukan strategi paralel PSO yang dapat meningkatkan pencarian solusi optimum secara global (Deep, et al., 2010). Hal tersebut terbukti dengan hasil uji coba penggunaan *slave* PSO dapat meningkatkan solusi optimum berdasarkan nilai *fitness*. Namun, semakin banyak jumlah *slave* PSO akan mempengaruhi waktu eksekusi yang dapat divisualisasikan secara grafik pada Gambar 6.15.



Gambar 6.15 Grafik waktu eksekusi berdasarkan jumlah *slave* PSO

Pada Gambar 6.15 dapat diketahui besarnya jumlah *slave* PSO akan mempengaruhi waktu eksekusi proses optimasi parameter SVR. Berdasarkan teori strategi paralel PSO seharusnya dapat mempercepat proses komputasi (Deep, et al., 2010). Namun, berdasarkan hasil uji coba semakin banyak jumlah *slave* PSO cukup banyak memakan waktu akan tetapi dapat mengoptimalkan pencarian solusi secara global (Deep, et al., 2010). Strategi paralel PSO yang diterapkan pada penelitian ini terdapat *master* PSO yang melanjutkan komputasi pencarian solusi berdasarkan individu dari *slave* PSO. Oleh karena itu, pada proses *master* PSO memerlukan cukup banyak waktu. Dengan demikian, diperlukan penentuan jumlah *slave* PSO sesuai kebutuhan dan keterbatasan sumber daya.

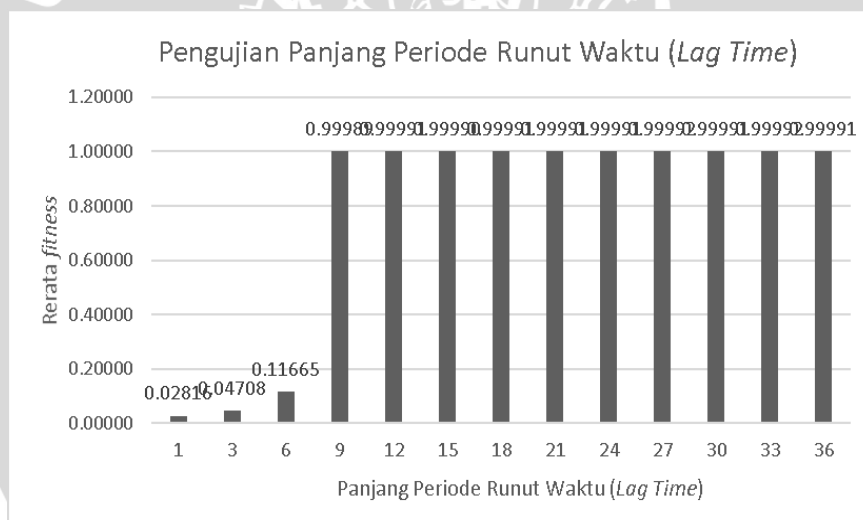
6.6 Hasil dan Pembahasan Uji Coba Panjang Periode Runut Waktu (*Lag Time*)

Uji coba panjang periode runut waktu (*lag time*) berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba panjang periode runut waktu (*lag time*) yang dilakukan dimulai dari 1 dengan berkelipatan 3 yaitu 1, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, dan 36 pada proses pelatihan SVR menggunakan data runut waktu curah hujan dasarian 1 bulan januari 2000 sampai dasarian 3 bulan desember 2001 pola data horizontal. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba panjang periode runut waktu (*lag time*).

- a. Panjang periode (*lag time*) : 18

- b. Ramalan (data uji) : 3
- c. Jumlah iterasi SVR : 1000
- d. Jumlah Partikel PPSO : 10
- e. Maksimal iterasi PPSO : 100
- f. Jumlah *slave* PSO : 2
- g. Batas parameter γ : 0.001 - 1
- h. Batas parameter λ : 0 - 0.1
- i. Batas parameter c) : 1 - 1000
- j. Batas parameter ϵ : 0.0001 - 0.1
- k. Batas parameter γ_k *kernel* : 0.0001 - 0.1
- l. Batas parameter d *kernel* : 1 - 3

Pengujian panjang periode runut waktu (*lag time*) dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter SVR dan konfigurasi SVR-PPSO yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba dilakukan sebanyak sepuluh kali berdasarkan perancangan skenario pengujian. Pada lampiran B.11 merupakan hasil uji coba panjang periode runut waktu (*lag time*) secara detail, sedangkan pada Gambar 6.16 merupakan hasil uji coba panjang periode runut waktu (*lag time*) yang direpresentasikan dalam bentuk grafik.



Gambar 6.16 Grafik hasil uji coba panjang periode runut waktu (*lag time*)

Pada Gambar 6.16 dapat diketahui periode runut waktu (*lag time*) dapat menggunakan panjang 9, 12, 15, 18, 21, 24, 27, 30, 33, dan 36 yang memiliki rerata nilai fitness di atas 0.9. Panjang periode runut waktu (*lag time*) bernilai 9 artinya dasarian saat ini dianalisis berdasarkan dasarian 2 bulan yang lalu. Lain halnya, dengan 36 *lag time* dapat berarti dasarian saat ini dianalisis berdasarkan dasarian yang sama pada tahun lalu karena satu tahun terdapat 36 dasarian. Konfigurasi panjang periode runut waktu (*lag time*) telah dibahas pada sub bab 4.2.1 sebelumnya.

6.7 Hasil dan Pembahasan Uji Coba Peramalan Runut Waktu Curah Hujan

Uji coba peramalan runut waktu ditujukan untuk melakukan pengujian metode SVR-PPSO menggunakan variasi data runut waktu curah hujan. Pada konfigurasi data diperlukan pengetahuan dasar mengenai data runut waktu yang akan digunakan. Sebelum dilakukan uji coba peramalan runut waktu curah hujan dilakukan uji coba pengaruh parameter ANOVA RBF *kernel* terlebih dahulu untuk mendapatkan konfigurasi parameter yang tepat. Selanjutnya dilakukan uji coba peramalan runut waktu curah hujan tahunan, musiman, dan tahun ke tahun.

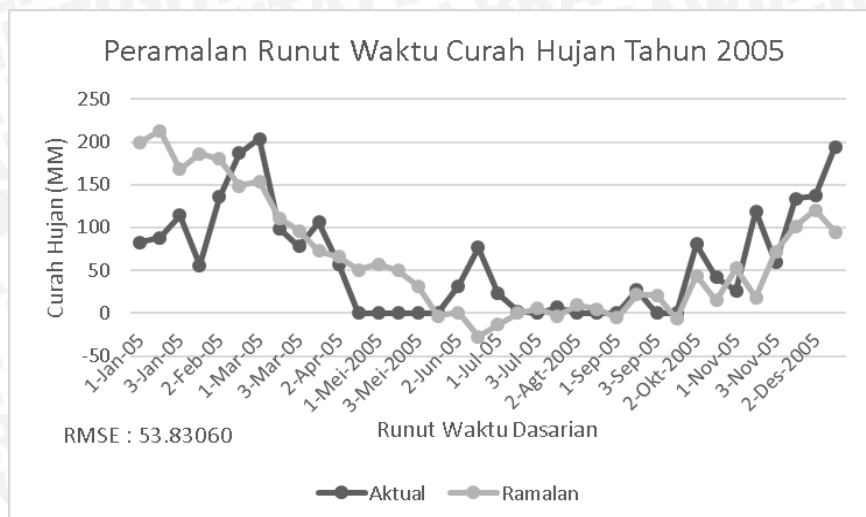
6.7.1 Hasil dan Pembahasan Uji Coba Pengaruh Parameter ANOVA RBF *Kernel*

Uji coba pengaruh batas parameter ANOVA RBF *kernel* berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba pengaruh batas parameter ANOVA RBF *kernel* yang dilakukan pada proses pelatihan SVR menggunakan data runut waktu curah hujan dasarian 1 bulan januari 2000 sampai dasarian 3 bulan desember 2005 pola data horizontal. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba pengaruh batas parameter ANOVA RBF *kernel*.

- | | |
|--|----------------|
| a. Panjang periode (<i>lag time</i>) | : 36 |
| b. Ramalan (data uji) | : 36 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |
| f. Jumlah <i>slave</i> PSO | : 2 |
| g. Batas parameter γ | : 0.001 - 1 |
| h. Batas parameter λ | : 0 - 0.1 |
| i. Batas parameter c | : 1 - 1000 |
| j. Batas parameter ϵ | : 0.0001 - 0.1 |

Pengujian pengaruh batas parameter ANOVA RBF *kernel* dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter SVR dan konfigurasi SVR-PPSO yang didapatkan dari hasil uji coba sebelumnya. Setiap uji coba menerapkan konfigurasi parameter γ *kernel* dan d *kernel* yang bervariasi. Pada lampiran B.12 merupakan hasil uji coba pengaruh batas parameter ANOVA RBF *kernel* secara detail, sedangkan pada Gambar 6.17 merupakan hasil uji coba pengaruh batas parameter ANOVA RBF *kernel* yang direpresentasikan dalam bentuk grafik dengan konfigurasi parameter γ *kernel* dan d *kernel* seperti berikut.

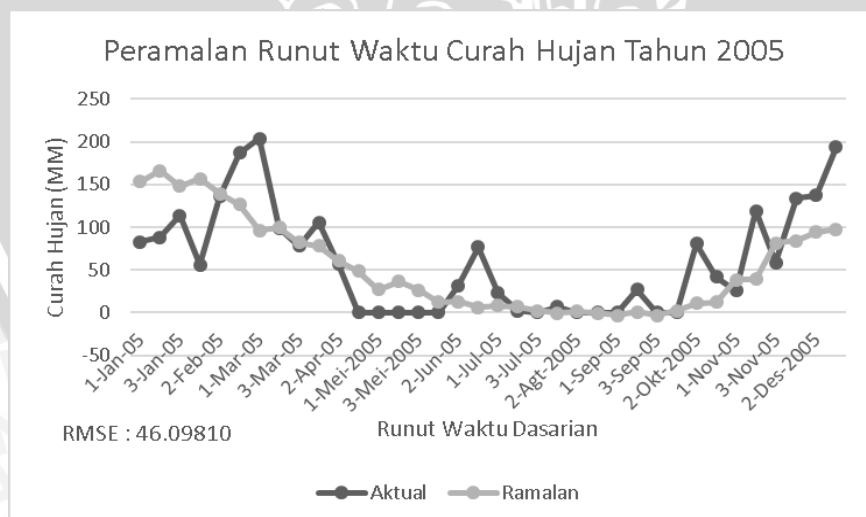
- | | |
|---|----------------|
| a. Batas parameter γ <i>kernel</i> | : 0.0001 - 0.1 |
| b. Batas parameter d <i>kernel</i> | : 1 - 1 |



Gambar 6.17 Grafik hasil uji coba γ_k 0.0001 – 0.1 dan d 1

Pada Gambar 6.17 menunjukkan ramalan menggunakan konfigurasi parameter γ_k *kernel* bernilai 0.0001 – 0.1 dan d *kernel* konsisten bernilai 1 dapat meramalkan curah hujan dengan nilai RMSE 53.83060 pada proses pengujian SVR. Dapat dilihat pada Gambar 6.17 ramalan yang dihasilkan mampu mengikuti pola nilai aktual runut waktu curah hujan secara fluktuatif. Hasil uji coba pengaruh batas parameter ANOVA RBF *kernel* selanjutnya dapat dilihat pada Gambar 6.18 yang direpresentasikan dalam bentuk grafik dengan konfigurasi parameter γ_k *kernel* dan d *kernel* seperti berikut.

- Batas parameter γ_k *kernel* : 0.0001 – 0.1
- Batas parameter d *kernel* : 2 – 2

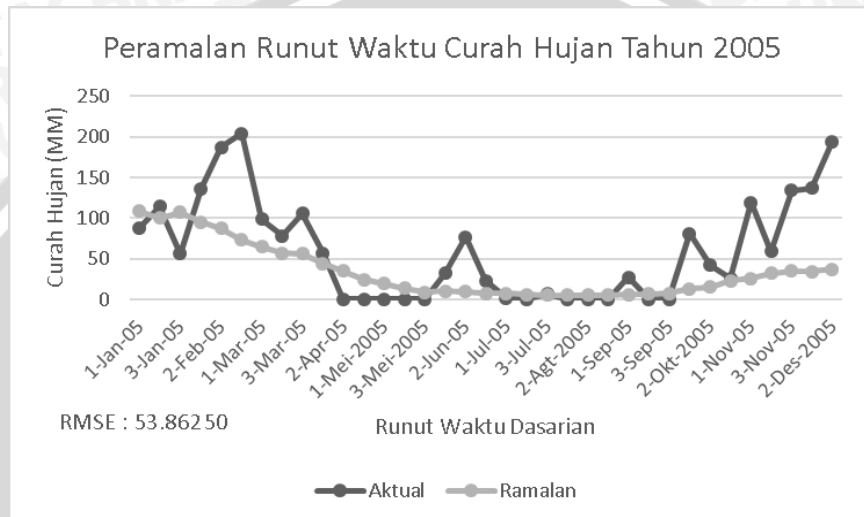


Gambar 6.18 Grafik hasil uji coba γ_k 0.0001 – 0.1 dan d 2

Pada Gambar 6.18 menunjukkan ramalan menggunakan konfigurasi parameter γ_k *kernel* bernilai 0.0001 – 0.1 dan d *kernel* konsisten bernilai 2 dapat meramalkan curah hujan dengan nilai RMSE 46.09810 pada proses pengujian SVR. Dapat dilihat pada Gambar 6.18 ramalan yang dihasilkan mampu mengikuti pola

nilai aktual runut waktu curah hujan. Namun, tingkat linieritas berkurang tidak dapat mengikuti pola nilai aktual runut waktu curah hujan secara fluktuatif. Hal tersebut dipengaruhi oleh konfigurasi parameter d *kernel* yang konsisten bernilai 2. Hasil uji coba pengaruh batas parameter ANOVA RBF *kernel* selanjutnya dapat dilihat pada Gambar 6.19 yang direpresentasikan dalam bentuk grafik dengan konfigurasi parameter yk *kernel* dan d *kernel* seperti berikut.

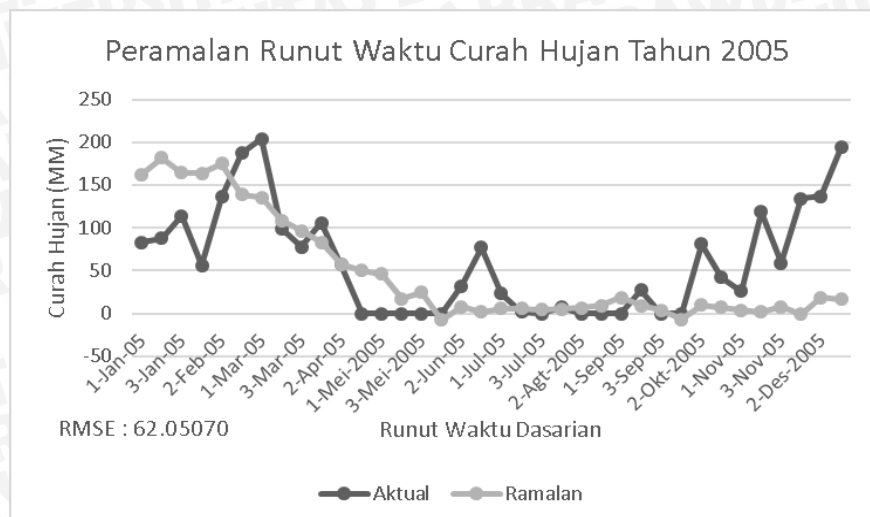
- Batas parameter yk *kernel* : 0.0001 – 0.1
- Batas parameter d *kernel* : 3 – 3



Gambar 6.19 Grafik hasil uji coba yk 0.0001 – 0.1 dan d 3

Pada Gambar 6.19 menunjukkan ramalan menggunakan konfigurasi parameter yk *kernel* bernilai 0.0001 – 0.1 dan d *kernel* konsisten bernilai 3 dapat meramalkan curah hujan dengan nilai RMSE 53.86250 pada proses pengujian SVR. Dapat dilihat pada Gambar 6.19 ramalan yang dihasilkan mampu mengikuti pola nilai aktual runut waktu curah hujan. Namun, tingkat linieritas berkurang tidak dapat mengikuti pola nilai aktual runut waktu curah hujan secara fluktuatif dan cenderung mendatar. Hal tersebut dipengaruhi oleh konfigurasi parameter d *kernel* yang konsisten bernilai 3. Hasil uji coba pengaruh batas parameter ANOVA RBF *kernel* selanjutnya dapat dilihat pada Gambar 6.20 yang direpresentasikan dalam bentuk grafik dengan konfigurasi parameter yk *kernel* dan d *kernel* seperti berikut.

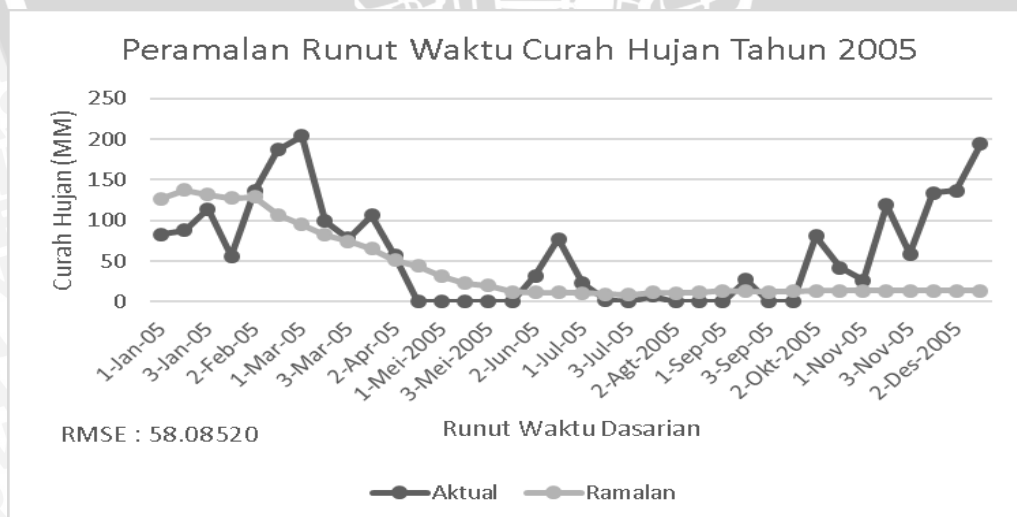
- Batas parameter yk *kernel* : 1 – 100
- Batas parameter d *kernel* : 1 – 1



Gambar 6.20 Grafik hasil uji coba γ_k 1 - 100 dan d 1

Pada Gambar 6.20 menunjukkan ramalan menggunakan konfigurasi parameter γ_k *kernel* bernilai 1 - 100 dan d *kernel* konsisten bernilai 1 dapat meramalkan curah hujan dengan nilai RMSE 62.05070 pada proses pengujian SVR. Selain itu, ramalan yang dihasilkan mampu mengikuti pola nilai aktual runut waktu curah hujan secara fluktuatif. Namun, terdapat ketidakmampuan dalam meramalkan runut waktu curah hujan akhir tahun yang cenderung mendatar tidak seperti pola hasil ramalan pada Gambar 6.17. Hal tersebut dipengaruhi oleh konfigurasi parameter γ_k *kernel* bernilai 1 - 100 dan d *kernel* yang konsisten bernilai 1. Hasil uji coba pengaruh batas parameter ANOVA RBF *kernel* selanjutnya dapat dilihat pada Gambar 6.21 yang direpresentasikan dalam bentuk grafik dengan konfigurasi parameter γ_k *kernel* dan d *kernel* seperti berikut.

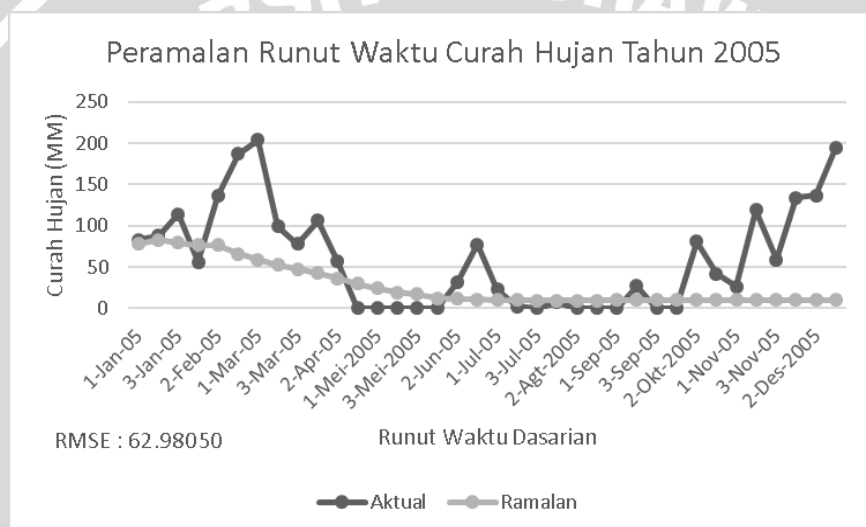
- Batas parameter γ_k *kernel* : 1 - 100
- Batas parameter d *kernel* : 2 - 2



Gambar 6.21 Grafik hasil uji coba γ_k 1 - 100 dan d 2

Pada Gambar 6.21 menunjukkan hasil ramalan dengan menggunakan konfigurasi parameter γ *kernel* bernilai 1 - 100 dan d *kernel* konsisten bernilai 2 dapat meramalkan curah hujan dengan nilai RMSE 58.08520 pada proses pengujian SVR. Dapat dilihat pada Gambar 6.21 ramalan yang dihasilkan mampu mengikuti pola nilai aktual runut waktu curah hujan. Namun, tingkat linieritas berkurang tidak dapat mengikuti pola nilai aktual runut waktu curah hujan secara fluktuatif dan cenderung mendatar. Hal tersebut dipengaruhi oleh konfigurasi parameter γ *kernel* bernilai 1 - 100 dan d *kernel* yang konsisten bernilai 2. Hasil uji coba pengaruh batas parameter ANOVA RBF *kernel* selanjutnya dapat dilihat pada Gambar 6.22 yang direpresentasikan dalam bentuk grafik dengan konfigurasi parameter γ *kernel* dan d *kernel* seperti berikut.

- Batas parameter γ *kernel* : 0.0001 – 0.1
- Batas parameter d *kernel* : 3 – 3



Gambar 6.22 Grafik hasil uji coba γ 0.0001 – 0.1 dan d 3

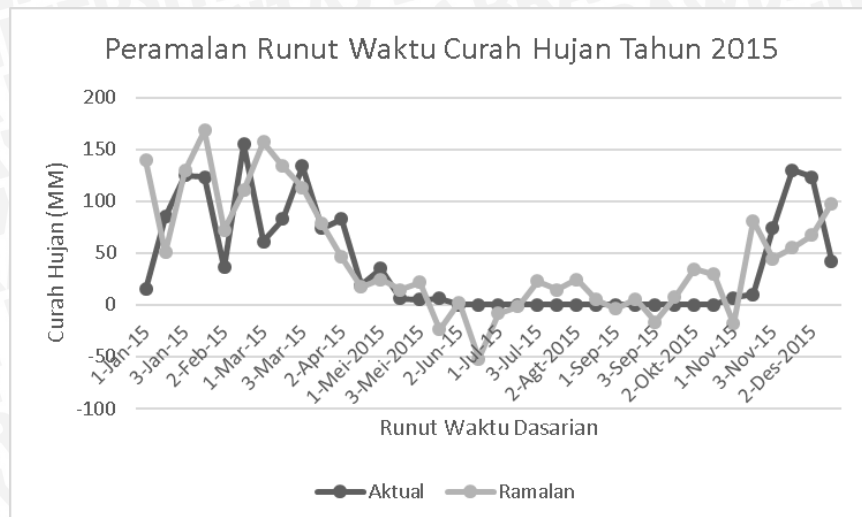
Pada Gambar 6.22 menunjukkan ramalan menggunakan konfigurasi parameter γ *kernel* bernilai 1 - 100 dan d *kernel* konsisten bernilai 2 dapat meramalkan curah hujan dengan nilai RMSE 62.98050 pada proses pengujian SVR. Dapat dilihat pada Gambar 6.22 ramalan yang dihasilkan mampu mengikuti pola nilai aktual runut waktu curah hujan. Namun, tingkat linieritas berkurang tidak dapat mengikuti pola nilai aktual runut waktu curah hujan secara fluktuatif dan cenderung mendatar. Hal tersebut dipengaruhi oleh konfigurasi parameter γ *kernel* bernilai 1 - 100 dan d *kernel* yang konsisten bernilai 3. Dengan demikian, berdasarkan hasil uji coba yang dilakukan sebaiknya konfigurasi parameter γ *kernel* bernilai 0.0001 – 0.1. Pada pengaturan parameter d *kernel* dapat konsisten bernilai 1, 2, dan 3 sesuai pola data yang akan diramalkan. Dengan kata lain, parameter d *kernel* tidak perlu dioptimasi menggunakan PSO pada proses pelatihan SVR.

6.7.2 Hasil dan Pembahasan Uji Coba Peramalan Runut Waktu Curah Hujan Tahunan

Uji coba peramalan runut waktu curah hujan tahunan berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba peramalan runut waktu curah hujan tahunan menggunakan data runut waktu curah hujan dasarian 1 bulan januari 2000 sampai dasarian 3 bulan desember 2015 pola data horizontal. Uji coba peramalan runut waktu curah hujan tahunan dimulai dari ramalan tahun 2002 secara bertahap sampai tahun 2015. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba peramalan runut waktu curah hujan tahunan.

- | | |
|---|----------------|
| a. Panjang periode (<i>lag time</i>) | : 36 |
| b. Ramalan (data uji) | : 36 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |
| f. Jumlah <i>slave</i> PSO | : 2 |
| g. Batas parameter γ | : 0.001 - 1 |
| h. Batas parameter λ | : 0 - 0.1 |
| i. Batas parameter c | : 1 - 1000 |
| j. Batas parameter ε | : 0.0001 - 0.1 |
| k. Batas parameter γ_k <i>kernel</i> | : 0.0001 - 0.1 |
| l. Batas parameter d <i>kernel</i> | : 1 - 1 |

Pengujian peramalan runut waktu curah hujan tahunan dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter SVR dan konfigurasi SVR-PPSO yang didapatkan dari hasil uji coba sebelumnya. Pada lampiran B.13 merupakan hasil uji coba peramalan runut waktu curah hujan tahunan secara detail. Berdasarkan hasil uji coba yang dilakukan, SVR-PPSO dapat melakukan peramalan runut waktu curah hujan tahunan dengan nilai RMSE terendah 41.51560 pada ramalan runut waktu curah hujan tahun 2015. Pada Gambar 6.23 merupakan hasil uji coba peramalan runut waktu curah hujan tahun 2015 yang direpresentasikan dalam bentuk grafik.



Gambar 6.23 Grafik hasil peramalan runut waktu curah hujan tahun 2015

Pada Gambar 6.23 menunjukkan SVR-PPSO dapat meramalkan runut waktu curah hujan tahun 2015 dengan nilai RMSE 41.51560 pada proses pengujian SVR. Selain itu, ramalan yang dihasilkan SVR-PPSO mampu mengikuti pola nilai aktual runut waktu curah hujan tahun 2015 secara fluktuatif. Berikut merupakan parameter SVR-PPSO terbaik yang digunakan dalam peramalan runut waktu curah hujan tahun 2015 di wilayah Poncokusumo.

- | | |
|--------------------------------|-------------|
| a. Parameter γ | : 0.02056 |
| b. Parameter λ | : 0.00983 |
| c. Parameter c | : 916.88057 |
| d. Parameter ϵ | : 0.02761 |
| e. Parameter γ_k kernel | : 0.06210 |
| f. Parameter d kernel | : 1 |

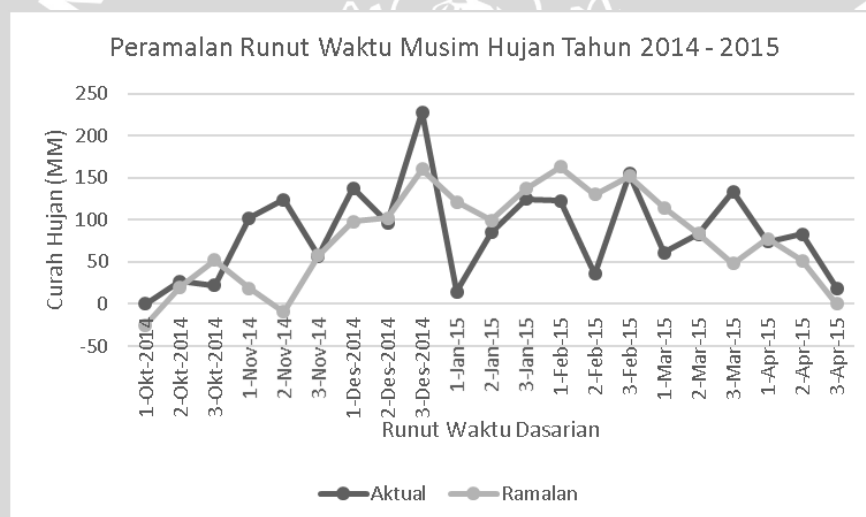
6.7.3 Hasil dan Pembahasan Uji Coba Peramalan Runut Waktu Curah Hujan Musiman

Uji coba peramalan runut waktu curah hujan musiman berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba peramalan runut waktu curah hujan musiman menggunakan data runut waktu curah hujan dasarian 1 bulan oktober 2000 sampai dasarian 3 bulan april 2015 pola data horizontal. Uji coba peramalan runut waktu curah hujan musiman dimulai dari ramalan musim hujan dasarian bulan april tahun 2002 secara bertahap sampai bulan april tahun 2015. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba peramalan runut waktu curah hujan musiman.

- | | |
|--|--------|
| a. Panjang periode (<i>lag time</i>) | : 21 |
| b. Ramalan (data uji) | : 21 |
| c. Jumlah iterasi SVR | : 1000 |
| d. Jumlah Partikel PPSO | : 10 |
| e. Maksimal iterasi PPSO | : 100 |

- f. Jumlah *slave* PSO : 2
- g. Batas parameter γ : 0.001 - 1
- h. Batas parameter λ : 0 - 0.1
- i. Batas parameter c : 1 - 1000
- j. Batas parameter ε : 0.0001 - 0.1
- k. Batas parameter γ_k *kernel* : 0.0001 - 0.1
- l. Batas parameter d *kernel* : 1 - 1

Pengujian peramalan runut waktu curah hujan musiman dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter SVR dan konfigurasi SVR-PPSO yang didapatkan dari hasil uji coba sebelumnya. Pada lampiran B.14 merupakan hasil uji coba peramalan runut waktu curah hujan musiman secara detail. Berdasarkan hasil uji coba yang dilakukan, SVR-PPSO dapat melakukan peramalan runut waktu curah hujan musiman dengan nilai RMSE terendah 56.11680 pada ramalan runut waktu curah hujan musiman tahun 2014-2015. Pada Gambar 6.24 merupakan hasil uji coba peramalan runut waktu curah hujan musiman tahun 2015 yang direpresentasikan dalam bentuk grafik.



Gambar 6.24 Grafik hasil peramalan runut waktu curah hujan musiman tahun 2014 – 2015

Pada Gambar 6.24 menunjukkan SVR-PPSO dapat meramalkan runut waktu curah hujan musiman tahun 2014-2015 dengan nilai RMSE 56.11680 pada proses pengujian SVR. Selain itu, ramalan yang dihasilkan SVR-PPSO mampu mengikuti pola nilai aktual runut waktu curah hujan tahun 2014-2015 secara fluktuatif. Berikut merupakan parameter SVR-PPSO terbaik yang digunakan dalam peramalan runut waktu curah hujan musiman tahun 2015 di wilayah Poncokusumo.

- a. Parameter γ : 0.01012
- b. Parameter λ : 0.05700
- c. Parameter c : 655.09746
- d. Parameter ε : 0.01843

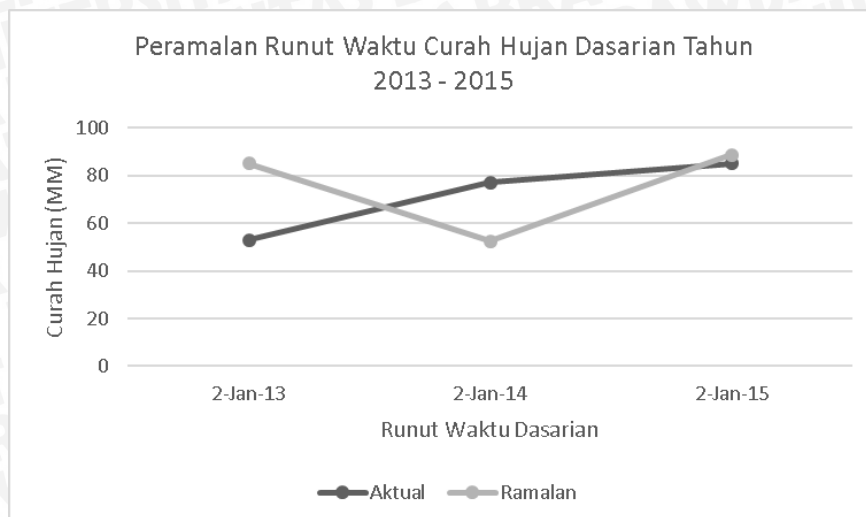
- e. Parameter γ_k kernel : 0.09977
 f. Parameter d kernel : 1

6.7.4 Hasil dan Pembahasan Uji Coba Peramalan Runut Waktu Curah Hujan Tahun ke Tahun

Uji coba peramalan runut waktu curah hujan tahun ke tahun berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba peramalan runut waktu curah hujan tahun ke tahun menggunakan pola data vertikal yang mewakili musim hujan pada dasarian 2 bulan januari dan desember tahun 2000 sampai tahun 2015. Uji coba peramalan runut waktu curah hujan musiman dimulai dari ramalan dasarian 2 bulan januari tahun 2012 sampai dasarian 2 bulan januari tahun 2015 dan begitu juga dengan bulan desember. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba peramalan runut waktu curah hujan tahun ke tahun.

- a. Panjang periode (*lag time*) : 1, 3, dan 5
 b. Ramalan (data uji) : 1, 3, dan 5
 c. Jumlah iterasi SVR : 1000
 d. Jumlah Partikel PPSO : 10
 e. Maksimal iterasi PPSO : 100
 f. Jumlah *slave* PSO : 2
 g. Batas parameter γ : 0.001 - 1
 h. Batas parameter λ : 0 - 0.1
 i. Batas parameter c : 1 - 1000
 j. Batas parameter ϵ : 0.0001 - 0.1
 k. Batas parameter γ_k kernel : 0.0001 - 0.1
 l. Batas parameter d kernel : 1 - 1

Pengujian peramalan runut waktu curah hujan tahun ke tahun dilakukan berdasarkan konfigurasi awal parameter SVR-PPSO. Selain itu, digunakan batas parameter SVR dan konfigurasi SVR-PPSO yang didapatkan dari hasil uji coba sebelumnya. Pada lampiran B.15 merupakan hasil uji coba peramalan runut waktu curah hujan tahun ke tahun secara detail. Berdasarkan hasil uji coba yang dilakukan, SVR-PPSO dapat melakukan peramalan runut waktu curah hujan tahun ke tahun dengan nilai RMSE terendah 23.35970 pada ramalan runut waktu curah hujan dasarian 2 tahun 2013, 2014, dan 2015. Pada Gambar 6.25 merupakan hasil uji coba peramalan runut waktu curah hujan dasarian 2 tahun 2013, 2014, dan 2015 yang direpresentasikan dalam bentuk grafik.



Gambar 6.25 Grafik hasil peramalan runut waktu curah hujan dasarian 2 tahun 2013, 2014, dan 2015

Pada Gambar 6.25 menunjukkan SVR-PPSO dapat meramalkan runut waktu curah hujan dasarian 2 tahun 2013, 2014, dan 2015 dengan nilai RMSE 23.35970 pada proses pengujian SVR. Selain itu, ramalan yang dihasilkan SVR-PPSO mampu mendekati nilai aktual runut waktu curah hujan dasarian 2 tahun 2013, 2014, dan 2015. Berikut merupakan parameter SVR-PPSO terbaik yang digunakan dalam peramalan runut waktu curah hujan dasarian 2 tahun 2013, 2014, dan 2015 di wilayah Poncokusumo.

- | | |
|--------------------------------|-------------|
| a. Parameter γ | : 0.50930 |
| b. Parameter λ | : 0.03781 |
| c. Parameter c | : 734.70757 |
| d. Parameter ϵ | : 0.00014 |
| e. Parameter γ_k kernel | : 0.03313 |
| f. Parameter d kernel | : 1 |

6.8 Hasil dan Pembahasan Uji Coba Perbandingan Peramalan Runut Waktu Curah Hujan Menggunakan Metode SVR, SVR-PPSO, dan Regresi Linier Berganda

Uji coba perbandingan peramalan runut waktu curah hujan menggunakan metode SVR, SVR-PPSO, dan regresi linier berganda berdasarkan perancangan skenario pengujian yang telah dilakukan sebelumnya. Uji coba perbandingan peramalan runut waktu curah hujan tahunan menggunakan data runut waktu curah hujan dasarian 1 bulan januari 2000 sampai dasarian 3 bulan desember 2015 pola data horizontal. Uji coba peramalan runut waktu curah hujan tahunan dimulai dari ramalan tahun 2002 secara bertahap sampai tahun 2015. Berikut merupakan konfigurasi awal parameter SVR-PPSO untuk uji coba peramalan runut waktu curah hujan tahunan.

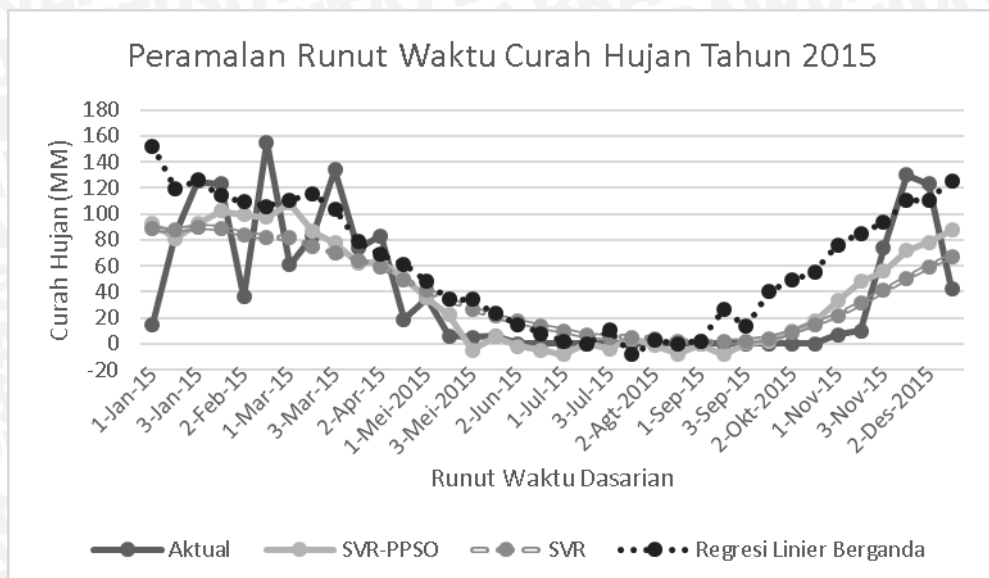
- | | |
|--|------|
| a. Panjang periode (<i>lag time</i>) | : 36 |
| b. Ramalan (data uji) | : 36 |

- c. Jumlah iterasi SVR : 1000
- d. Jumlah Partikel PPSO : 10
- e. Maksimal iterasi PPSO : 100
- f. Jumlah *slave* PSO : 2
- g. Batas parameter γ : 0.001 - 1
- h. Batas parameter λ : 0 - 0.1
- i. Batas parameter c : 1 - 1000
- j. Batas parameter ε : 0.0001 - 0.1
- k. Batas parameter γ_k *kernel* : 0.0001 - 0.1
- l. Batas parameter d *kernel* : 1 - 1

Pengujian peramalan runut waktu curah hujan tahunan pada metode SVR-PPSO dilakukan berdasarkan konfigurasi awal parameter dan batas parameter SVR yang didapatkan dari hasil uji coba sebelumnya. Pada konfigurasi awal metode SVR dilakukan tanpa adanya metode optimasi parameter oleh PPSO, sama halnya dengan metode regresi linier berganda. Berikut merupakan konfigurasi awal parameter metode SVR yang dapat diasumsikan berdasarkan penelitian sebelumnya pada Tabel 4.13.

- a. Panjang periode (*lag time*) : 36
- b. Ramalan (data uji) : 36
- c. Jumlah iterasi SVR : 1000
- d. Batas parameter γ : 0.001
- e. Batas parameter λ : 0
- f. Batas parameter c : 1000
- g. Batas parameter ε : 0.0001
- h. Batas parameter γ_k *kernel* : 0.01
- i. Batas parameter d *kernel* : 2

Selain konfigurasi awal SVR-PPSO dan SVR, konfigurasi awal pada metode regresi linier berganda hanya terdapat pada konfigurasi panjang periode (*lag time*) bernilai 36 dan panjang ramalan (data uji) bernilai 36. Pada lampiran B.16 merupakan hasil uji coba peramalan runut waktu curah hujan tahunan secara detail, sedangkan pada Gambar 6.26 merupakan hasil uji coba perbandingan peramalan runut waktu curah hujan menggunakan metode SVR, SVR-PPSO, dan regresi linier berganda yang direpresentasikan dalam bentuk grafik.



Gambar 6.26 Grafik hasil perbandingan peramalan runut waktu curah hujan menggunakan metode SVR, SVR-PPSO, dan regresi linier berganda

Pada Gambar 6.26 menunjukkan metode SVR, SVR-PPSO, dan regresi linier berganda dapat meramalkan runut waktu curah hujan tahun 2015 pada proses pengujian. Berdasarkan hasil uji coba yang dilakukan, SVR-PPSO dapat melakukan peramalan runut waktu curah hujan di tahun 2015 dengan nilai RMSE 30.1386 lebih rendah dari metode SVR dengan nilai RMSE 32.0153 dan metode regresi linier berganda dengan nilai RMSE 41.9851.

BAB 7 PENUTUP

Pada bab ini memuat kesimpulan terhadap penelitian ini yang diperoleh dari hasil perancangan, implementasi sistem, dan pengujian. Serta saran dari penulis untuk pengembangan selanjutnya terkait peramalan runut waktu curah hujan menggunakan metode *Support Vector Regression* dan *Parallel Particle Swarm Optimization* (SVR-PPSO).

7.1 Kesimpulan

Berdasarkan hasil penelitian peramalan runut waktu curah hujan menggunakan metode SVR-PPSO, maka dapat diperoleh kesimpulan 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-PPSO), pemodelan ramalan, validasi model ramalan, penerapan model ramalan, dan pengamatan hasil 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 komputasi.
2. Penerapan metode SVR-PPSO untuk peramalan diperlukan data runut waktu curah hujan yang terbagi menjadi data latih dan data uji. Strategi PPSO melakukan optimasi parameter SVR menggunakan konsep sekumpulan individu yang memiliki dimensi nilai parameter SVR yaitu γ , λ , c , ϵ , γ_k *kernel*, dan d *kernel*. Tiap individu melakukan evaluasi *fitness* yang diperoleh dari nilai *Root Mean Square Error* (RMSE) pada proses pelatihan SVR berdasarkan dimensi individu pada batas yang telah ditentukan. Berdasarkan nilai *fitness*, tiap individu bertukar informasi untuk mendapatkan *fitness* terbaik dengan pembaruan kecepatan dan perpindahan posisi dimensi sampai pada batas waktu yang telah ditentukan. Individu dengan *fitness* terbaik merupakan parameter SVR yang optimal untuk digunakan pada proses pengujian SVR. Pengujian SVR merupakan proses utama untuk melakukan peramalan runut waktu curah hujan.
3. Evaluasi kinerja peramalan runut waktu curah hujan menggunakan metode SVR-PPSO 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. Untuk mendapatkan kinerja peramalan runut waktu curah hujan yang baik diperlukan konfigurasi metode SVR-PPSO yang tepat. Berdasarkan hasil pengujian, metode SVR-PPSO dapat melakukan peramalan runut waktu curah hujan tahun 2015 dengan nilai RMSE 41.51560, peramalan runut waktu curah hujan musiman tahun 2014 – 2015 dengan nilai RMSE 56.11680, dan peramalan runut waktu curah hujan dasarian 2 tahun 2013, 2014, dan 2015 dengan nilai RMSE 23.35970.

7.2 Saran

Berdasarkan hasil penelitian peramalan runut waktu curah hujan menggunakan metode SVR-PPSO, terdapat saran yang dapat digunakan untuk pengembangan selanjutnya sebagai berikut:

1. Tidak perlu mengoptimalkan parameter *d kernel*. Cukup dengan mengatur nilai parameter *d kernel* secara konsisten bernilai 1, 2, dan 3 sesuai dengan pola data yang digunakan.
2. Menggunakan *kernel trick* lain pada metode SVR untuk melakukan pemetaan pada ruang fitur.
3. Penambahan data runut waktu curah hujan, semakin banyak data runut waktu akan berpengaruh pada proses pelatihan SVR dalam mengenali pola data curah hujan.
4. Mengoptimalkan parameter metode SVR dengan strategi atau algoritma optimasi yang lain.
5. Menerapkan metode SVR-PPSO untuk melakukan peramalan runut waktu curah hujan di wilayah lain atau dengan objek selain curah hujan.
6. Penambahan variabel lain seperti suhu, kecepatan angin, dan lainnya untuk menghubungkan sebab akibat (*kausal*) dalam peramalan runut waktu curah hujan.
7. Menggunakan strategi peramalan lain untuk dapat melakukan peramalan lebih baik.



DAFTAR PUSTAKA

- Abraham, B. & Ledolter, J., 2005. Statistical methods for forecasting. s.l.:John Wiley & Sons.
- 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. TELKOMNIKA Indonesian Journal of Electrical Engineering, November, Volume 12(11), pp. 7912-7919.
- 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.
- Agarwal, D. & Sharma, D., 2016. Experimental Study on Bound Handling Techniques for Multi-objective Particle Swarm Optimization. Innovations in Bio-Inspired Computing and Applications, Volume 424, pp. 555-564.
- Agus, I. & Hartati, H., 2011. Uji Kesesuaian Chi-Kuadrat Data Hujan Das Batang Kuranji Kota Padang. REKAYASA SIPIL, Volume 7(2), pp. 99-111.
- 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.
- Appati, J., Gogovi, G. & Fosu, G., 2014. On the Selection of Appropriate Kernel Function for SVM in Face Recognition. International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4(3).
- Armstrong, J. e., 2001. Principles of forecasting: a handbook for researchers and practitioners. s.l., KLUWER ACADEMIC PUBLISHERS.
- Asalla, S. et al., 2016. Restoring Mitochondrial Function: A Small Molecule-mediated Approach to Enhance Glucose Stimulated Insulin Secretion in Cholesterol Accumulated Pancreatic beta cells. Scientific reports, Volume 6, p. 27513.
- Bagheripour, P., Gholami, A., Asoodeh, M. & Vaezzadeh-Asadi, M., 2015. Support vector regression based determination of shear wave velocity. Journal of Petroleum Science and Engineering, Volume 124, pp. 95-99.

- 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.
- 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, 2015a. 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, 2015b. 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.
- Box, G., Jenkins, G., Reinsel, G. & Ljung, G., 2015. *Time series analysis: forecasting and control*. 5 ed. s.l.:John Wiley & Sons.
- Brockwell, P. & Davis, R., 2002. *Introduction to Time Series and Forecasting*. 2 ed. s.l.:Springer Science & Business Media.
- Cahyani, M. P., Dewi, C. & Yudistira, N., 2014. Implementasi Algoritma Al-Alaoui Backpropagation dalam peramalan curah hujan. *Repositori Jurnal Mahasiswa PTIIK UB*, Volume 3(2).
- Chang, C. & Lin, C., 2002. Training v-support vector regression: theory and algorithms. *Neural Computation*, 1 August, Volume 14(8), pp. 1959-1977.
- Chen, H. I. 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.
- Chen, N., Lu, W., Yang, J. & Li, G., 2004. *Support vector machine in chemistry*. s.l.:World Scientific Publishing Co. Pte. Ltd.
- Chen, W., Hong, Y., Jun, D. & Jiulong, J., 2013. Research on Parameter Selection of Support Vector Regression. In *Applied Mechanics and Materials*, Volume 344, pp. 219-225.
- Cholissodin, I., Kurniawati, M., Indriati & Arwani, I., 2014. Classification of campus e-complaint documents using Directed Acyclic Graph Multi-class SVM based on analytic hierarchy process. n *Advanced Computer Science and Information Systems (ICACSIS)*, 2014 International Conference on, 18-19 October.pp. 247-253.
- Clerc, M., 2006. Confinements and biases in particle swarm optimisation.

- Cortes, C. & Vapnik, V., 1995. Support-vector networks. Machine learning, , Volume 20(3), pp. 273-297..
- Crone, S., Guajardo, J. & Weber, R., 2006a. 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., 2006b. 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.
- Deep, K., Sharma, S. & Pant, M., 2010. Modified Parallel Particle Swarm Optimization for Global Optimization. In Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on, September.pp. 1451-1458.
- 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.
- Drucker, H. et al., 1997. Support Vector Regression Machines. Advances in neural information processing systems, p. 155–161.
- Eberhart, R. C. & Shi, Y., 2001. Particle swarm optimization: developments, applications and resources. In evolutionary computation, 2001. Proceedings of the 2001 Congress on, Volume 1, pp. 81-86.
- Eberhart, R. & Kennedy, J., 1995. A New Optimizer Using Particle Swarm Theory. In Proceedings of the sixth international symposium on micro machine and human science, Volume 1, pp. 39-43.
- 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.
- Endriyanto & Ihsan, F., 2011. Teknik pengamatan curah hujan di stasiun klimatologi kebun percobaan Cukurgondang, Pasuruan. Buletin Teknik Pertanian, Volume 16(2), pp. 61-63.
- Engelbrecht, A. P., 2007. Computational Intelligence: An Introduction 2nd ed. 2 ed. s.l.:West Sussex: John Wiley & Sons Ltd.
- Fadel, S. et al., 2016. Investigating the Effect of Different Kernel Functions on the Performance of SVM for Recognizing Arabic Characters. International Journal

- of Advanced Computer Science and Applications, Volume 1(7), pp. pp.446-450.
- Genton, M., 2001. Classes of kernels for machine learning: a statistics perspective. The Journal of Machine Learning Research, Volume 2, pp. 299-312.
- Graf, A. & Borer, S., 2001. Normalization in support vector machines. In Pattern Recognition, pp. 277-282.
- Gunn, S., 1998. Support vector machines for classification and regression. ISIS technical report, 10 May. Volume 14.
- Guo, W., Chen, G. & Feng, X., 2006. A new strategy of acceleration coefficients for particle swarm optimization. In 2006 10th International Conference on Computer Supported Cooperative Work in Design, pp. 1-5.
- 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.
- Heizer, J. & Render, B., 2011. Operations Management. In: 10th ed. s.l.:s.n.
- Helwig, S., Branke, J. & Mostaghim, S., 2013. Experimental Analysis of Bound Handling Techniques in Particle Swarm Optimization. IEEE Transactions on Evolutionary computation, Volume 17(2), pp. 259 - 271.
- 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.
- 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.
- Irwanti, Y. C., Cholissodin, I. & Ridok, A., 2016. Peramalan Harga Bbm Menggunakan Metode Support Vector Regression Dengan Ant Colony Optimization. Repositori Jurnal Mahasiswa PTIIK UB, Volume 7(26).
- ISO/IEC/IEEE 42010:2011(E), 2011. Systems and software engineering -- Architecture description. s.l., IEEE, pp. 1-46.
- Ivanciuc, O., 2002. Support vector machine identification of the aquatic toxicity mechanism of organic compounds. Internet Electron. J. Mol., pp. 157-172.
- Jakubcová, M., Máca, P. & Pech, P., 2014. A comparison of selected modifications of the particle swarm optimization algorithm. Journal of Applied Mathematics, pp. 1-.

- Junaedi, H., Budianto, H., Maryati, I. & Melani, Y., 2011. DATA TRANSFORMATION PADA DATA MINING. Prosiding Konferensi Nasional "Inovasi dalam Desain dan Teknologi" - IDeaTech.
- Karatzoglou, A., Meyer, D. & Hornik, K., 2006. Support vector machines in R. Journal of statistical software, Volume 15(9), pp. 1-28.
- Kaundal, R., Kapoor, A. & Raghava, G., 2006. Machine learning techniques in disease forecasting: a case study on rice blast prediction. BMC bioinformatics,, Volume 7(1), p. 1.
- Kennedy, J. & Eberhart, R., 1995. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks, Nov/Dec, Volume 4, pp. 1942 - 1948.
- Laskov, P., Gehl, C., Krüger, S. & Müller, K., 2006. Incremental support vector learning: Analysis, implementation and applications. The Journal of Machine Learning Research, Volume 1909-1936, pp. 1909-1936.
- Levenbach, H., 2015. Taming Uncertainty: All You Need to Know About Measuring Forecast Accuracy, But Are Afraid to Ask. [Online] Available at: http://cpdftraining.org/downloads/Levenbach_Accuracy2015.pdf [Accessed 26 04 2016].
- Loiacono, D., Marelli, A. & Lanzi, P., 2007. Support vector regression for classifier prediction. s.l., s.n., pp. 1806-1813.
- 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..
- Makridakis, S., Wheelwright, S. & Hyndman, R., 1998. Forecasting: Methods and Applications. 3 ed. s.l.:Wiley.
- Manullang, V. & Tamba, T., 2013. Modifikasi Penakar Hujan Otomatis Tipe Tipping Bucket Dengan Hall Effect Sensor ATS276. Saintia Fisika, Volume 6.
- 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.
- Microsoft Press, 2002. Microsoft Computer Dictionary, Fifth Edition 5th. In: WA, USA: Microsoft Press Redmond, WA, USA ©2002, p. 704.
- Montgomery, D., Jennings, C. & Kulahci, M., 2015. Introduction to time series analysis and forecasting. Hoboken(New Jersey): John Wiley & Sons.

- Mori, H. & Ohmi, M., 2005. Probabilistic short-term load forecasting with Gaussian processes. In Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems, November.p. 6.
- Muliantara, A., Sanjaya ER, N. & Widiartha, I., 2015. Perancangan Alat Ukur Ketinggian Curah Hujan Otomatis Berbasis Mikrokontroler. Jurnal Ilmu Komputer, Volume 8(2).
- 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.
- Nickabadi, A., Ebadzadeh, M. & Safabakhsh, R., 2011. A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing, Volume 11(4), pp. 3658-3670.
- Novitasari, D., Cholissodin, I. & Mahmudy, W., 2016. Hybridizing PSO With SA for Optimizing SVR Applied to Software Effort Estimation. TELKOMNIKA (Telecommunication Computing Electronics and Control), Volume 14(1).
- Osuna, E., Freund, R. & Girosi, F., 1997. An improved training algorithm for support vector machines. In Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop, pp. 276-285.
- 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.
- Platt, J., 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. 21 April.
- Ponnambalam, S., Jawahar, N. & Chandrasekaran, S., 2009. Discrete particle swarm optimization algorithm for flowshop scheduling. INTECH Open Access Publisher.
- Prabhakar, D. & Dutta, M., 2013. Prediction of software effort using artificial neural network and support vector machine. International Journal of Advanced Research in Computer Science and Software Engineering, March, 3(3), pp. 40-46.
- Prahotama, 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.
- Prognose, Ö., Woschnagg, E. & Cipan, J., 2004. Evaluating Forecast Accuracy. [Online]
Available at: <http://homepage.univie.ac.at/robert.kunst/procip.pdf>
[Accessed 26 04 2016].

- 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. & Yuhaziz, 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.
- Schölkopf, B., Bartlett, P., Smola, A. & Williamson, R., 1998. Shrinking the Tube: A New Support Vector Regression Algorithm. *Advances in neural information processing systems*, pp. 330-336.
- Schölkopf, B. & Smola, A., 2002a. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge, Massachusetts London, England: MIT press.
- Schölkopf, B. & Smola, A., 2002b. *Support Vector Machines and Kernel Algorithms*.
- Sewell, M., 2007. *Kernel methods*. London: University of London, Department of Computer Science.
- Shahzad, F. et al., 2009. Opposition-based particle swarm optimization with velocity clamping (OVCPSO).. In *Advances in Computational Intelligence*, pp. 339-348.
- 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., 1998a. A Modified Particle Swarm Optimizer. May.pp. 69-73.
- Shi, Y. & Eberhart, R., 1998b. Parameter Selection in Particle Swarm Optimization. In *International Conference on Evolutionary Programming*, March.pp. 591-600.
- Shi, Y. & Eberhart, R., 1999. Empirical study of particle swarm optimization. In *Evolutionary Computation*, 1999. CEC 99. Proceedings of the 1999 Congress on, Volume 3.

- Silver, N., 2012. The signal and the noise: Why so many predictions fail-but some don't. s.l.:Penguin.
- Singh, P., 2016. Applications of Soft Computing in Time Series Forecasting: Simulation and Modeling Techniques. s.l.:Springer.
- Smola, A. & Schölkopf, B., 2004. A tutorial on support vector regression.. Statistics and computing, Volume 14(3), pp. 199-222.
- Soebroto, A. et al., 2015. Prediksi Tinggi Muka Air (TMA) Untuk Deteksi Dini Bencana Banjir Menggunakan SVR-TVIWPSO. Jurnal Teknologi Informasi dan Ilmu Komputer, Volume 2(2).
- 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.
- Sonjaya, I. et al., 2009. Uji Aplikasi HyBMG Versi 2.0 Untuk Prakiraan Curah Hujan. Buletin Meteorologi Klimatologi dan Geofisika, September, 5(3), pp. 323-339.
- Sukarelawati, E., 2012. Bupati Malang Instruksikan Warga Waspada Puting Beliung. [Online] Available at: <http://www.antarajatim.com/lihat/berita/81037/bupati-malang-instruksikan-warga-waspada-puting-beliung> [Accessed 17 07 2016].
- Tim Pusat Bahasa Depdiknas, 2008. Kamus Bahasa Indonesia. Jakarta: Pusat Bahasa Departemen Pendidikan Nasional.
- Tjasyono HK, B., 2012a. Mikrofisika Awan dan Hujan. Jakarta(Jakarta): Badan Meteorologi Klimatologi dan Geofisika.
- Tjasyono HK, B., 2012b. Meteorologi Indonesia 1, Karakteristik dan Sirkulasi Atmosfer. Jakarta: Badan Meteorologi Klimatologi dan Geofisika.
- 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..
- 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. IIA/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.

- Vishwanathan, S., Smola, A. & Murty, M., 2003. SimpleSVM. s.l., s.n.
- Wang, D., Wang, M. & Qiao, X., 2009. Support vector machines regression and modeling of greenhouse environment. *Computers and electronics in agriculture*, Volume 66(1), pp. 46-52.
- Wang, W., Xu, Z., Lu, W. & Zhang, X., 2003. Determination of the spread parameter in the Gaussian kernel for classification and regression. *Neurocomputing*, Volume 55(3), pp. 643-663.
- Wei, C., 2012. Wavelet Support Vector Machines for Forecasting Precipitation in Tropical Cyclones: omparisons with GSVM, Regression, and MM5. *Weather and Forecasting*, April, Volume 27(2), pp. 438-450.
- Wirjohamidjojo, S. & Swarinoto, Y., 2007. *Praktek Meteorologi Pertanian*. Jakarta: Badan Meteorologi Klimatologi dan Geofisika.
- 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.
- 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.
- 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.

LAMPIRAN A DATA CURAH HUJAN HARIAN WILAYAH KARANGPLOSO

A.1 Data curah hujan harian wilayah Karangploso tahun 2010

Tanggal	Tahun 2010											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	25	25	0	1	0	0	0	0	9	23	45	9999
2	0	41	20	1	0	10	0	0	0	10	39	9999
3	0	3	15	5	22.0	0	0	0	0	9	37	9999
4	0	5	20	7	33	7	0	0	0	8	0	9999
5	0	0	190	50	25	0	22	0	16	0	0	9999
6	69	7	21	57	35	2	0	0	22	0	0	9999
7	9	28	3	5	0	6	0	0	0	1	5	9999
8	5	0	56	10	17	0	0	0	0	6	69	9999
9	0	7	1	35	0	0	0	0	14	28	99	9999
10	30	0	0	5	11	0	0	0	0	0	0	9999
11	1	15	0	0	3	0	0	0	16	0	0	9999
12	10	4	0	0	2	0	40	0	20	2	0	9999
13	1	10	0	0	2	0	1	16	0	3	0	9999
14	4	0	0	0	3	0	1	0	0	29	0	9999
15	6	12	15	0	1	3	0	0	6	0	0	9999
16	3	36	16	0	3	0	0	0	18	0	0	9999
17	35	1	0	0	2	0	0	30	0	7	4	9999
18	4	37	0	0	0	0	0	0	4	42	18	9999
19	1	1	0	0	5	0	0	0	13	0	0	9999
20	1	0	5	0	4	0	0	0	0	0	24	9999
21	10	17	20	0	2	0	0	26	8	24	10	9999
22	16	0	0	0	10	0	0	0	0	10	34	9999
23	8	5	3	0	4	0	0	0	10	15	23	9999
24	10	8	18	0	20	0	0	11	12	12	18	9999
25	30	16	0	0	1	0	0	4	9	0	15	9999
26	29	19	5	0	45	0	3	0	23	6	19	9999
27	14	10	0	0	35	0	0	0	0	3	4	9999
28	15	6	38	0	0	0	15	0	25	4	6	9999
29	31	0	1	0	0	0	0	0	3	0	46	9999
30	0	0	6	0	0	0	0	0	13	18	1	9999
31	16	0	10	0	15	0	0	0	0	0	0	9999

Sumber: BMKG Stasiun Klimatologi Karangploso Malang

A.2 Data curah hujan harian wilayah Karangploso tahun 2011

Tanggal	Tahun 2011											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	18	16	7	12	16	0	0	0	0	0	5	9999
2	1	48	12	14	17	0	0	0	0	0	10	9999
3	0	36	14	10	12	0	0	0	0	0	3	9999
4	0	22	0	19	21	0	0	0	0	0	6	9999
5	8	12	9	24	26	0	0	0	0	0	0	9999
6	0	8	12	29	19	0	0	0	0	0	3	9999
7	4	9	13	21	31	0	0	0	0	0	23	9999
8	0	10	21	24	0	0	0	0	0	0	91	9999
9	0	0	30	28	0	0	0	0	0	0	21	9999
10	18	13	14	31	12	0	0	0	0	0	25	9999
11	19	14	8	27	8	0	0	0	0	0	1	9999
12	8	16	10	20	11	0	0	0	0	0	1	9999
13	0	0	0	24	18	0	0	0	0	0	0	9999
14	0	0	7	28	17	0	0	0	0	0	1	9999
15	0	21	9	18	0	0	0	0	0	0	3	9999
16	0	0	13	15	0	0	0	0	0	0	7	9999
17	0	8	16	22	9	0	0	0	0	0	0	9999
18	9	7	17	4	6	0	0	0	0	0	0	9999
19	8	0	3	21	5	0	0	0	0	0	0	9999
20	4	0	6	19	19	0	0	0	0	0	13	9999
21	9	8	0	45	22	0	0	0	0	5	1	9999
22	11	0	12	20	0	0	0	0	0	0	3	9999
23	7	0	18	16	0	0	0	0	0	0	15	9999
24	6	0	37	0	0	0	0	0	0	0	45	9999
25	8	0	26	0	0	0	0	0	0	0	26	9999
26	4	7	28	0	0	0	0	0	0	0	0	9999
27	9	3	19	0	0	0	0	0	0	0	0	9999
28	0	4	21	0	0	0	0	0	0	15	23	9999
29	7	0	25	0	0	0	0	0	0	0	0	9999
30	12	0	10	0	0	0	0	0	0	9	0	9999
31	14	0	8	0	0	0	0	0	0	10	0	9999

Sumber: BMKG Stasiun Klimatologi Karangploso Malang

A.3 Data curah hujan harian wilayah Karangploso tahun 2015

Tanggal	Tahun 2015											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	9999	5	19	0	42	0	0	0	0	0	0	29
2	9999	5	8	0	53	0	0	0	0	0	0	3
3	9999	1	2	6	34	0	0	0	0	0	10	3
4	9999	0	23	0	0	0	0	0	0	0	0	0
5	9999	38	0	0	7	0	0	0	0	0	0	0
6	9999	31	10	18	0	0	0	0	0	0	0	0
7	9999	6	5	0	9	0	0	0	0	0	0	0
8	9999	11	9	0	12	0	0	0	0	0	0	0
9	9999	2	2	0	0	0	0	0	0	0	3	0
10	9999	20	0	0	4	0	0	0	0	0	0	0
11	9999	11	0	4	3	0	0	0	0	0	8	67
12	9999	66	4	24	6	0	0	0	0	0	54	0
13	9999	0	7	21	0	0	0	0	0	0	0	6
14	9999	16	0	0	0	0	0	0	0	0	0	1
15	9999	0	21	0	0	0	0	0	0	0	0	13
16	9999	0	16	0	0	0	0	0	0	0	0	0
17	9999	0	5	0	0	0	0	0	0	0	0	6
18	9999	0	10	0	0	0	0	0	0	0	3	9
19	9999	30	29	0	0	0	0	0	0	0	0	28
20	9999	8	0	0	0	0	0	0	0	0	0	1
21	9999	0	0	0	0	0	0	0	0	0	0	10
22	9999	0	0	0	0	0	0	0	0	0	0	0
23	9999	1	1	0	0	0	0	0	0	0	0	7
24	9999	18	18	0	0	0	0	0	0	0	0	0
25	9999	4	4	1	0	0	0	0	0	0	0	8
26	9999	0	0	2	0	0	0	0	0	0	0	0
27	9999	0	0	0	0	0	0	0	0	0	0	0
28	9999	0	0	18	0	0	0	0	0	0	0	0
29	9999		0	0	0	0	0	0	0	0	0	8
30	9999		0	0	0	0	0	0	0	0	0	0
31	9999		0		0		0	0		0		4

Sumber: BMKG Stasiun Klimatologi Karangploso Malang

LAMPIRAN B HASIL PENGUJIAN

B.1 Hasil Uji Coba Batas Parameter γ

Batas Min γ	Batas Maks γ	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0.0001	0.001	0.03162	0.02935	0.02898	0.02524	0.02633	0.03060	0.02765	0.02366	0.03081	0.01751	0.02717
	0.01	0.29843	0.24996	0.76778	0.01151	0.37463	0.06771	0.63590	0.73402	0.58094	0.55387	0.42748
	0.1	0.42056	0.65744	0.67805	0.73607	0.45954	0.64170	0.77058	0.84310	0.83443	0.93519	0.69767
0.001	0.01	0.59762	0.35560	0.33746	0.51996	0.93744	0.59637	0.42773	0.58218	0.47852	0.27603	0.51089
	0.1	0.45363	0.54445	0.72450	0.78652	0.74228	0.73509	0.85691	0.34358	0.44130	0.10966	0.57379
	1	0.78883	0.60377	0.74664	0.89953	0.95967	0.95598	0.78637	0.92375	0.81204	0.61968	0.80963
0.01	0.1	0.98776	0.91688	0.94117	0.68003	0.84902	0.66309	0.84220	0.41509	0.20911	0.28761	0.67919
	1	0.78403	0.38130	0.17721	0.62658	0.74908	0.64456	0.97953	0.83750	0.04310	0.45677	0.56797
	10	0.04241	0.01640	0.01749	0.46684	0.09907	0.81583	0.03061	0.02081	0.02357	0.01813	0.15512
0.1	1	0.54127	0.75129	0.59909	0.03853	0.64985	0.03340	0.12750	0.91700	0.20550	0.76777	0.46312
	10	0.05956	0.80969	0.02639	0.02328	0.01839	0.03867	0.02564	0.03832	0.01866	0.01694	0.10755
	100	0.02645	0.05450	0.01930	0.01706	0.01992	0.02319	0.01666	0.01682	0.01849	0.01491	0.02273

B.2 Hasil Uji Coba Batas Parameter λ

Batas Min λ	Batas Maks λ	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0	0	0.95467	0.96521	0.95906	0.99455	0.99264	0.97890	0.96024	0.91919	0.83647	0.89988	0.94608
	0.0001	0.93532	0.99651	0.94131	0.99518	0.96159	0.98685	0.98071	0.98518	0.98390	0.96118	0.97278
	0.001	0.99957	0.88744	0.99090	0.98867	0.96268	0.98499	0.98899	0.99297	0.96446	0.98595	0.97466
0.0001	0.001	0.93856	0.99596	0.98077	0.97875	0.95026	0.98969	0.98343	0.98952	0.92378	0.96270	0.96934
	0.01	0.91231	0.96519	0.99870	0.98842	0.98607	0.97717	0.97520	0.89234	0.99493	0.86655	0.95569
	0.1	0.87422	0.99390	0.98485	0.94787	0.99397	0.95060	0.88544	0.80369	0.91174	0.84250	0.91888
0.001	0.01	0.99517	0.99738	0.91666	0.76935	0.99507	0.98850	0.84077	0.97885	0.93176	0.83753	0.92510
	0.1	0.96361	0.98571	0.99748	0.89948	0.95027	0.89455	0.97452	0.97590	0.94515	0.85028	0.94369
	1	0.97204	0.57674	0.79062	0.99705	0.86280	0.96906	0.91787	0.99351	0.99801	0.99464	0.90723
0.01	0.1	0.94949	0.98572	0.89571	0.99249	0.99018	0.99363	0.92281	0.99300	0.99670	0.94653	0.96663
	1	0.48315	0.34683	0.78954	0.86720	0.68055	0.64885	0.85127	0.61459	0.70176	0.68725	0.68725
	5	0.15772	0.38524	0.52523	0.43879	0.95073	0.81470	0.52026	0.83363	0.18439	0.48014	0.52908

B.3 Hasil Uji Coba Batas Parameter C

Batas Min c	Batas Maks c	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0.001	0.01	0.00871	0.00871	0.00871	0.00871	0.00871	0.00871	0.00871	0.00871	0.00871	0.00871	0.00871
	0.1	0.00873	0.00874	0.00873	0.00873	0.00873	0.00873	0.00873	0.00873	0.00873	0.00873	0.00873
	1	0.00897	0.00897	0.00897	0.00897	0.00897	0.00897	0.00897	0.00897	0.00896	0.00897	0.00897
0.01	0.1	0.00873	0.00873	0.00873	0.00873	0.00873	0.00873	0.00873	0.00873	0.00873	0.00873	0.00873
	1	0.00897	0.00897	0.00896	0.00897	0.00897	0.00897	0.00897	0.00900	0.00897	0.00897	0.00897
	10	0.01146	0.01147	0.01147	0.01144	0.01148	0.01144	0.01131	0.01147	0.01149	0.01146	0.01145
1	10	0.01148	0.01148	0.01149	0.01137	0.01143	0.01139	0.01149	0.01149	0.01148	0.01149	0.01146
	100	0.02954	0.02911	0.02928	0.02520	0.02545	0.02397	0.02960	0.02804	0.02851	0.02721	0.02759
	1000	0.95582	0.99338	0.99183	0.96964	0.99529	0.95266	0.96012	0.89289	0.83215	0.87570	0.94195
10	100	0.02546	0.02812	0.02772	0.02826	0.02513	0.02543	0.02900	0.02536	0.02522	0.02919	0.02689
	1000	0.99667	0.86463	0.88151	0.89716	0.95679	0.99218	0.97211	0.98764	0.96765	0.94786	0.94642
100	1000	0.87918	0.99483	0.99413	0.97203	0.96705	0.97806	0.85871	0.99245	0.82398	0.96399	0.94244

B.4 Hasil Uji Coba Batas Parameter ϵ

Batas Min ϵ	Batas Maks ϵ	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0.0001	0.001	0.99989	0.99989	0.99990	0.99990	0.99989	0.99990	0.99990	0.99990	0.99989	0.99990	0.99989
	0.01	0.99988	0.99989	0.99990	0.99986	0.99988	0.99988	0.99989	0.99989	0.99988	0.99989	0.99988
	0.1	0.99986	0.99986	0.99988	0.99984	0.99985	0.99986	0.99988	0.99986	0.99986	0.99973	0.99985
0.001	0.01	0.99897	0.99901	0.99898	0.99882	0.99888	0.99896	0.99904	0.99882	0.99902	0.99902	0.99895
	0.1	0.99866	0.99884	0.99877	0.99857	0.99888	0.99858	0.99878	0.99866	0.99850	0.99895	0.99872
	1	0.99801	0.99858	0.99856	0.99808	0.99774	0.99860	0.99801	0.99671	0.99823	0.99711	0.99796
0.01	0.1	0.98988	0.98917	0.98983	0.98870	0.98871	0.98972	0.98897	0.98978	0.98908	0.98823	0.98921
	1	0.98923	0.98774	0.98956	0.98909	0.98695	0.98767	0.98788	0.98747	0.98592	0.98419	0.98757
	10	0.98536	0.98442	0.98729	0.98306	0.98092	0.98322	0.98418	0.98143	0.98396	0.98530	0.98391
0.1	1	0.90360	0.90384	0.90785	0.90780	0.90819	0.90459	0.89729	0.90617	0.89268	0.89726	0.90293
	10	0.90422	0.89709	0.90183	0.89303	0.88275	0.88344	0.87442	0.88209	0.89725	0.82817	0.88443
	100	0.86154	0.82346	0.70482	0.84812	0.77965	0.76373	0.75790	0.86734	0.77754	0.89167	0.80758

B.5 Hasil Uji Coba Batas Parameter γ_k Kernel

Batas Min γ_k kernel	Batas Maks γ_k kernel	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0.0001	0.001	0.99987	0.99988	0.99988	0.99987	0.99988	0.99977	0.99982	0.99986	0.99987	0.99987	0.99986
	0.01	0.99988	0.99990	0.99990	0.99990	0.99990	0.99989	0.99990	0.99989	0.99990	0.99988	0.99990
	0.1	0.99990	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990
0.001	0.01	0.99990	0.99990	0.99990	0.99989	0.99990	0.99990	0.99990	0.99987	0.99989	0.99989	0.99989
	0.1	0.99990	0.99990	0.99989	0.99989	0.99988	0.99989	0.99990	0.99989	0.99990	0.99990	0.99989
	1	0.99990	0.99989	0.99989	0.99989	0.99990	0.99989	0.99989	0.99988	0.99990	0.99990	0.99989
0.01	0.1	0.99989	0.99989	0.99989	0.99990	0.99990	0.99989	0.99989	0.99990	0.99990	0.99990	0.99990
	1	0.99989	0.99989	0.99990	0.99989	0.99989	0.99989	0.99990	0.99989	0.99989	0.99989	0.99989
	10	0.99989	0.99990	0.99989	0.99989	0.99990	0.99989	0.99989	0.99990	0.99990	0.99990	0.99990
0.1	1	0.99990	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990
	10	0.99990	0.99990	0.99990	0.99990	0.99989	0.99988	0.99989	0.99989	0.99990	0.99989	0.99989
	100	0.99990	0.99990	0.99990	0.99990	0.99989	0.99989	0.99989	0.99990	0.99990	0.99990	0.99990
1	10	0.99990	0.99990	0.99989	0.99990	0.99990	0.99989	0.99989	0.99989	0.99989	0.99989	0.99989
	100	0.99989	0.99989	0.99989	0.99990	0.99990	0.99990	0.99989	0.99989	0.99990	0.99990	0.99989
10	100	0.99990	0.99990	0.99988	0.99988	0.99990	0.99990	0.99990	0.99989	0.99989	0.99990	0.99989

B.6 Hasil Uji Coba Batas Parameter d Kernel

Batas Min d Kernel	Batas Maks d Kernel	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
1	1	0.99979	0.99979	0.99978	0.99979	0.99881	0.99979	0.99979	0.99980	0.99979	0.99978	0.99969
	2	0.99988	0.99989	0.99990	0.99989	0.99989	0.99989	0.99989	0.99989	0.99989	0.99989	0.99989
	3	0.99990	0.99989	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990
2	2	0.99989	0.99989	0.99990	0.99989	0.99989	0.99989	0.99989	0.99988	0.99989	0.99989	0.99989
	3	0.99989	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	0.99989	0.99990
3	3	0.99990	0.99990	0.99990	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990

B.7 Hasil Uji Coba Jumlah Iterasi SVR

Jumlah Iterasi SVR	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>	Rerata Waktu Eksekusi (detik)
	1	2	3	4	5	6	7	8	9	10		
10	0.98633	0.98640	0.98643	0.98639	0.98631	0.98646	0.98645	0.98629	0.98646	0.98646	0.98640	13.67600
50	0.99988	0.99990	0.99990	0.99990	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	0.99990	14.68700
100	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	14.01700
500	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99989	0.99990	0.99990	0.99990	0.99990	16.75400
1000	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	0.99989	0.99990	20.21500
5000	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99989	0.99989	0.99990	58.80900
10000	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99989	0.99991	0.99990	100.25400
50000	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	335.14300
100000	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99989	0.99990	651.38700
500000	0.99989	0.99990	0.99990	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	0.99989	0.99990	2360.38700
1000000	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	5219.78500

B.8 Hasil Uji Coba Jumlah Partikel PPSO

Jumlah Partikel	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>	Rerata Waktu Eksekusi (detik)
	1	2	3	4	5	6	7	8	9	10		
5	0.99989	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	6.34500
10	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	12.10400
15	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	17.27400
20	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	24.28200
25	0.99990	0.99990	0.99990	0.99991	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	31.50800
30	0.99990	0.99991	0.99991	0.99990	0.99990	0.99991	0.99990	0.99990	0.99991	0.99990	0.99990	35.89800
35	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	43.23100
40	0.99990	0.99990	0.99990	0.99991	0.99991	0.99990	0.99991	0.99990	0.99991	0.99990	0.99990	48.49800
45	0.99990	0.99991	0.99991	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	54.28800
50	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	62.49900

B.9 Hasil Uji Coba Jumlah Iterasi PPSO

Jumlah Iterasi PPSO	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>	Rerata Waktu Eksekusi (detik)
	1	2	3	4	5	6	7	8	9	10		
10	0.99989	0.99990	0.99990	0.99989	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	4.46000
20	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	7.97700
30	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	11.76700
40	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	15.25500
50	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	18.84600
60	0.99990	0.99990	0.99990	0.99990	0.99991	0.99991	0.99990	0.99991	0.99991	0.99990	0.99990	22.37000
70	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	24.88300
80	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99991	0.99990	0.99990	28.61900
90	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	32.72400
100	0.99990	0.99991	0.99991	0.99990	0.99990	0.99991	0.99990	0.99990	0.99991	0.99990	0.99990	34.54100

B.10 Hasil Uji Coba Jumlah *Slave* PSO

Jumlah <i>Slave</i> PSO	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>	Rerata Waktu Eksekusi (detik)
	1	2	3	4	5	6	7	8	9	10		
0	0.99990	0.99990	0.99990	0.99990	0.99989	0.99990	0.99988	0.99989	0.99990	0.99990	0.99990	19.12200
1	0.99990	0.99990	0.99990	0.99990	0.99989	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	27.01600
2	0.99991	0.99991	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	35.96700
3	0.99990	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99990	51.69800
4	0.99991	0.99990	0.99990	0.99990	0.99991	0.99991	0.99991	0.99990	0.99990	0.99990	0.99990	59.00300
5	0.99991	0.99991	0.99991	0.99991	0.99990	0.99991	0.99990	0.99990	0.99991	0.99990	0.99991	66.11000
6	0.99990	0.99991	0.99991	0.99991	0.99991	0.99990	0.99990	0.99991	0.99990	0.99990	0.99991	79.68200
7	0.99990	0.99990	0.99990	0.99991	0.99991	0.99990	0.99991	0.99991	0.99990	0.99991	0.99991	89.64000
8	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991	0.99990	0.99990	0.99991	0.99991	0.99991	101.67400
9	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	0.99990	0.99991	0.99990	0.99990	0.99990	112.38300
10	0.99991	0.99991	0.99991	0.99990	0.99990	0.99991	0.99990	0.99990	0.99991	0.99990	0.99991	104.99400

B.11 Hasil Uji Coba Panjang Periode Runut Waktu (*Lag Time*)

Panjang Periode Runut Waktu (<i>Lag Time</i>)	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rerata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
1	0.02816	0.02818	0.02817	0.02815	0.02820	0.02819	0.02811	0.02817	0.02816	0.02815	0.02816
3	0.04708	0.04708	0.04708	0.04708	0.04708	0.04708	0.04708	0.04708	0.04708	0.04708	0.04708
6	0.11665	0.11665	0.11665	0.11665	0.11665	0.11665	0.11665	0.11665	0.11665	0.11665	0.11665
9	0.99989	0.99989	0.99989	0.99989	0.99989	0.99989	0.99990	0.99989	0.99989	0.99989	0.99989
12	0.99991	0.99991	0.99990	0.99991	0.99990	0.99990	0.99991	0.99991	0.99991	0.99991	0.99991
15	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990	0.99990
18	0.99991	0.99991	0.99991	0.99991	0.99990	0.99991	0.99990	0.99991	0.99990	0.99991	0.99991
21	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991
24	0.99991	0.99992	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991
27	0.99992	0.99992	0.99991	0.99991	0.99991	0.99991	0.99992	0.99992	0.99992	0.99992	0.99992
30	0.99992	0.99991	0.99991	0.99991	0.99991	0.99992	0.99991	0.99991	0.99991	0.99991	0.99991
33	0.99992	0.99992	0.99992	0.99992	0.99992	0.99992	0.99992	0.99991	0.99991	0.99992	0.99992
36	0.99991	0.99991	0.99991	0.99991	0.99992	0.99991	0.99991	0.99991	0.99991	0.99991	0.99991

B.12 Hasil Uji Coba Pengaruh Parameter ANOVA RBF Kernel

Batas Min yk kernel	Batas Maks yk kernel	Batas Min d Kernel	Batas Maks d Kernel	Evaluasi Pelatihan (SVR-PPSO)		Evaluasi Pengujian (SVR)	
				RMSE	fitness	RMSE	fitness
0.0001	0.1	1	1	0.00260	0.99740	53.83060	0.01820
		2	2	0.00020	0.99980	46.09810	0.02120
		3	3	0.00020	0.99980	53.86250	0.01820
1	100	1	1	0.00110	0.99890	62.05070	0.01590
		2	2	0.00030	0.99970	58.08520	0.01690
		3	3	0.00020	0.99980	62.98050	0.01560

B.13 Hasil Uji Coba Peramalan Runut Waktu Curah Hujan Tahunan

Runut Waktu Dasarian		Evaluasi Pelatihan (SVR-PPSO)		Evaluasi Pengujian (SVR)	
Awal	Akhir	RMSE	<i>fitness</i>	RMSE	<i>fitness</i>
1-jan-2000	3-des-2002	0.00020	0.99980	75.81430	0.01300
	3-des-2003	0.00040	0.99960	72.74140	0.01360
	3-des-2004	0.00150	0.99850	58.56040	0.01680
	3-des-2005	0.00900	0.99110	55.86560	0.01760
	3-des-2006	0.00310	0.99690	42.16280	0.02320
	3-des-2007	0.00130	0.99870	79.40440	0.01240
	3-des-2008	0.00320	0.99680	85.28070	0.01160
	3-des-2009	0.01390	0.98630	61.67930	0.01600
	3-des-2010	0.03110	0.96980	76.24160	0.01290
	3-des-2011	0.27890	0.78190	60.24260	0.01630
	3-des-2012	0.07860	0.92710	48.98360	0.02000
	3-des-2013	0.13260	0.88290	78.71800	0.01250
	3-des-2014	0.35740	0.73670	59.53880	0.01650
	3-des-2015	0.39790	0.71530	41.51560	0.02350

B.14 Hasil uji Coba Peramalan Runut Waktu Curah Hujan Musiman

Runut Waktu Dasarian		Evaluasi Pelatihan (SVR-PPSO)		Evaluasi Pengujian (SVR)	
Awal	Akhir	RMSE	<i>fitness</i>	RMSE	<i>fitness</i>
1-Okt-00	3-Apr-02	0.0002	0.99980	111.67500	0.00890
	3-Apr-03	0.0002	0.99980	74.92210	0.01320
	3-Apr-04	0.0016	0.99840	64.80740	0.01520
	3-Apr-05	0.085	0.92160	64.82300	0.01520
	3-Apr-06	0.0045	0.99550	57.02350	0.01720
	3-Apr-07	0.0954	0.91290	77.10800	0.01280
	3-Apr-08	0.1222	0.89110	80.07520	0.01230
	3-Apr-09	0.5764	0.63430	72.86060	0.01350
	3-Apr-10	1.2681	0.44090	75.20470	0.01310
	3-Apr-11	3.3496	0.22990	71.51210	0.01380
	3-Apr-12	12.9939	0.07150	61.06330	0.01610
	3-Apr-13	11.0781	0.08280	67.06840	0.01470
	3-Apr-14	5.0994	0.16390	82.25940	0.01200
	3-Apr-15	8.7955	0.10210	56.11680	0.01750

B.15 Hasil uji Coba Peramalan Runut Waktu Curah Hujan Tahun ke Tahun

Runut Waktu Dasarian		(Lag Time)	Ramalan (Data Uji)	Evaluasi Pelatihan (SVR-PPSO)		Evaluasi Pengujian (SVR)	
Awal	Akhir			RMSE	<i>fitness</i>	RMSE	<i>fitness</i>
2-jan-2000	2-jan-2015	1	1	22.68460	0.04220	77.86520	0.01270
		3	3	0.00010	0.99990	23.35970	0.04110
		5	5	0.00010	0.99990	55.13900	0.01780
2-des-2000	2-agt-2015	1	1	0.00020	0.99980	99.75110	0.00990
		3	3	0.00010	0.99990	147.17440	0.00670
		5	5	0.00010	0.99990	137.98070	0.00720

B.16 Hasil uji Coba Perbandingan Peramalan Runut Waktu Curah Hujan Menggunakan Metode SVR, SVR-PPSO, dan Regresi Linier Berganda

Metode	Evaluasi Pelatihan		Evaluasi Pengujian	
	RMSE	<i>fitness</i>	RMSE	<i>fitness</i>
SVR	54.4653	0.018	32.0153	0.0303
SVR-PPSO	51.874	0.0189	30.1386	0.0321
Regresi Linier Berganda	51.4313	0.0191	41.9851	0.0233