

repository.ub.ac.id

**PERAMALAN KONSUMSI BAHAN BAKAR MINYAK (BBM) DI
INDONESIA DENGAN METODE *HYBRID* ALGORITMA
GENETIKA DAN *SUPPORT VECTOR REGRESSION***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Rofiqoh Ainun Zakiyah

NIM: 125150201111071



PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
JURUSAN INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

PERAMALAN KONSUMSI BAHAN BAKAR MINYAK (BBM) DI INDONESIA DENGAN
METODE *HYBRID* ALGORITMA GENETIKA DAN *SUPPORT VECTOR REGRESSION*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Rofiqoh Ainun Zakiyah

NIM: 125150201111071

Skripsi ini telah diuji dan dinyatakan lulus pada
21 Juli 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Imam Cholissodin, S.Si., M.Kom

NIK: 201201 850719 1 001

Dian Eka Ratnawati, S.Si., M.Kom

NIP: 19730619 200212 2 001

Mengetahui
Ketua Jurusan Informatika

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

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 21 Juli 2016



Rofiqoh Ainun Zakiyah

NIM: 125150201111071

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat, rahmat, ridho dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“Peramalan Konsumsi Bahan Bakar Minyak (BBM) Di Indonesia Dengan Metode Hybrid Algoritma Genetika Dan Support Vector Regression”** sebagai salah satu persyaratan untuk menyelesaikan studi di Jurusan Informatika, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

Penulis menyadari bahwa tugas akhir ini dapat terselesaikan berkat bantuan, petunjuk, bimbingan dan dukungan dari berbagai pihak yang telah banyak membantu proses penyelesaian tugas akhir ini. Oleh karena itu penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada :

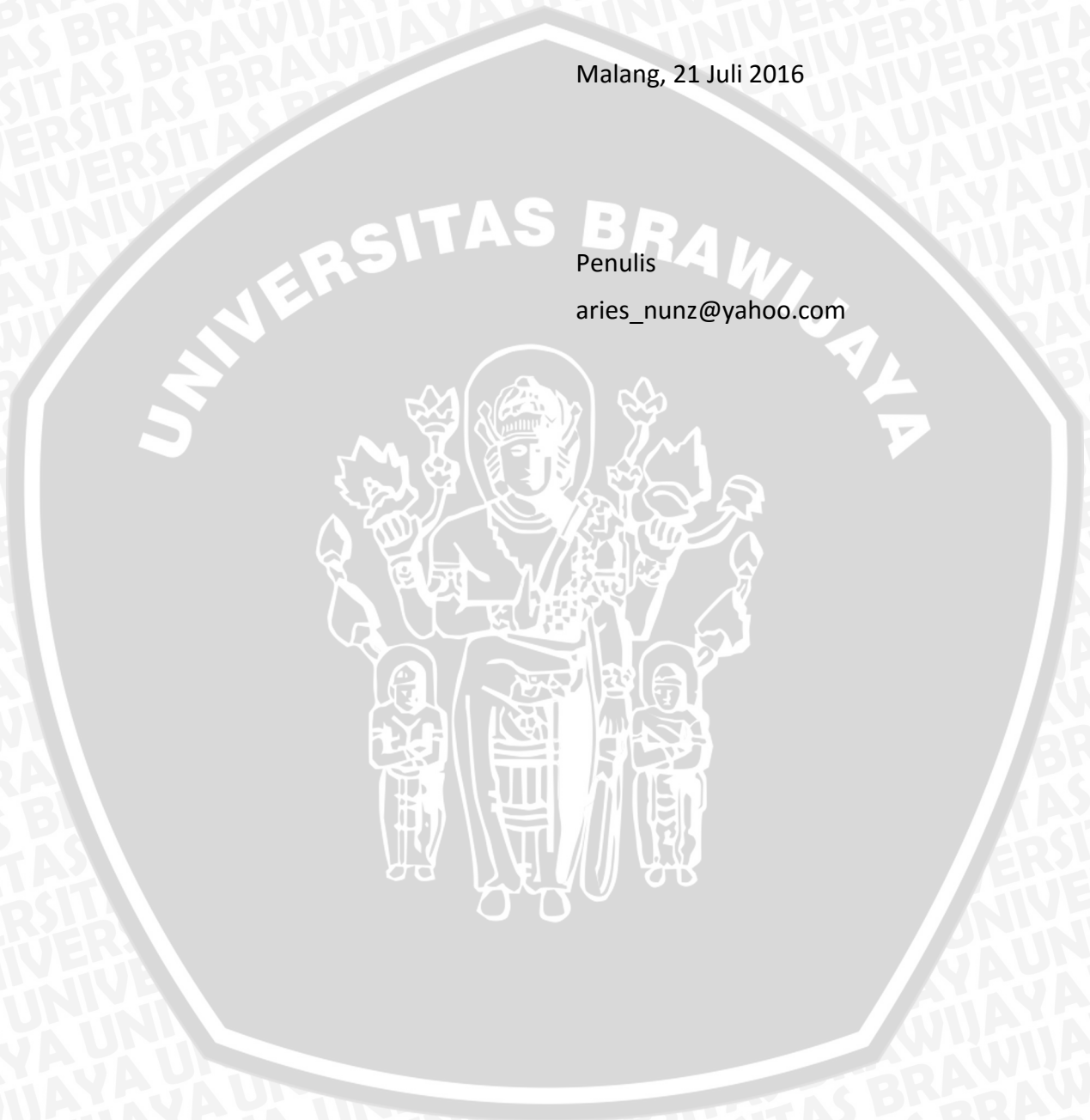
1. Imam Cholissodin, S.Si, M.Kom selaku pembimbing I dan Dian Eka Ratnawati, S.Si, M.Kom selaku pembimbing II yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. M. Tanzil Furqon, S.Kom, M.CompSc selaku penguji I dan Wijaya Kurniawan, S.T, M.T selaku penguji II yang telah menguji dan memberikan saran untuk skripsi saya sehingga skripsi saya bisa terselesaikan dengan lebih baik.
3. Orang tua penulis, Bapak Achmad Rusydi dan Ibu Fauzatur Rohillah yang tak henti-hentinya memberikan dukungan moril dan materil.
4. Saudara penulis, Nur Chanifah yang telah memberikan motivasi dan semangat demi terselesaikannya skripsi ini.
5. Seluruh dosen Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
6. Sahabat serta saudara penulis, Hikmah Maharani Iqomatul Haq yang senantiasa menemani, bertukar ilmu, serta memberikan semangat dan motivasi dalam penyelesaian skripsi.
7. Sahabat penulis Tasnima, Akmilatul, Tenika, Mega, Christine, Tanti, Adit, Imam dan Krisbianto yang selalu menemani, bertukar ilmu, semangat dan motivasi kepada penulis selama masa studi dan penyusunan skripsi.
8. Teman-teman IF-F terimakasih atas kebersamaan serta bantuan yang telah diberikan dan kita lakukan bersama dari semester 1 hingga semester 8 ini.
9. Teman-teman seperjuangan Fakultas Ilmu Komputer angkatan 2012 yang telah memberikan bantuan selama masa studi hingga penyelesaian skripsi ini.
10. Dan semua pihak yang tidak bisa disebutkan satu per satu. Terima kasih atas segala bantuannya.

Penulis sadar bahwa skripsi ini masih banyak kekurangan, oleh karena itu kritik dan saran yang bersifat membangun sangat diharapkan untuk menyempurnakan skripsi ini. Penulis berharap skripsi ini dapat bermanfaat khususnya bagi diri sendiri dan bagi semua pihak.

Malang, 21 Juli 2016

Penulis

aries_nunz@yahoo.com



ABSTRAK

Rofiqoh Ainun Zakiyah. 2016. Peramalan Konsumsi Bahan Bakar Minyak (BBM) Di Indonesia Dengan Metode *Hybrid* Algoritma Genetika Dan *Support Vector Regression*. Program Studi Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing : Imam Cholissodin, S.Si, M.Kom dan Dian Eka Ratnawati, S.Si, M.Kom.

Saat ini kebutuhan yang sangat penting untuk dipenuhi agar berlangsungnya berbagai macam kegiatan sehari-hari adalah bahan bakar minyak. Jika terjadi keterbatasan dalam memenuhi kebutuhan bahan bakar minyak dapat mengakibatkan kegiatan produksi, distribusi dan konsumsi menjadi terlambat. Hal tersebut bisa mengakibatkan terjadinya *demand for oil* yang meningkat, harga minyak internasional menjadi berubah-ubah dan gas dalam negeri yang melambung tinggi. Dengan melakukan peramalan konsumsi bahan bakar minyak akan sangat berguna untuk kelangsungan kegiatan sehari-hari dari berbagai sektor dan menghindari adanya kesenjangan negatif antara jumlah produksi dan konsumsi bahan bakar minyak. Banyak metode peramalan yang telah digunakan untuk meramalkan data yang bersifat non linier seperti *Support Vector Regression* (SVR) yang terbukti pada beberapa kasus peramalan dapat memberikan hasil peramalan yang baik, seperti pada peramalan konsumsi energi harian dan peramalan kebutuhan bahan bakar premium. Namun, untuk hasil yang lebih optimal dapat digunakan metode optimasi seperti *Hybrid* Algoritma Genetika. Penelitian mengenai penggunaan metode optimasi *Hybrid* Algoritma Genetika untuk peramalan menggunakan SVR pernah digunakan pada peramalan beban listrik dan terbukti metode tersebut dapat menghasilkan nilai *error* yang kecil. Sesuai dengan hasil pengujian yang dilakukan menggunakan data yang didapatkan dari *website* www.bp.com berupa jumlah konsumsi BBM di Indonesia dari tahun 1965-2014, metode SVR yang dioptimasi dengan HGA dapat menghasilkan MAPE sebesar 3.64 hingga 4.65.

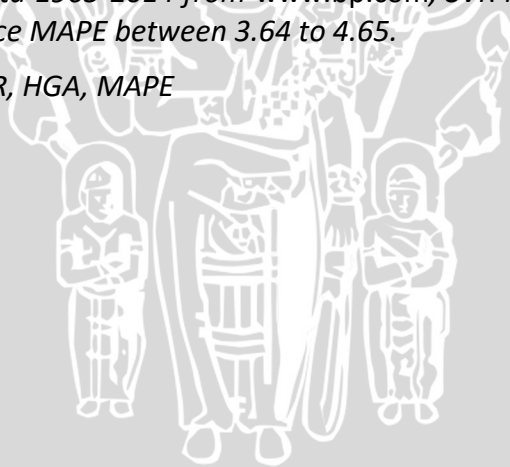
Kata Kunci : Bahan bakar minyak, SVR, HGA, MAPE.

ABSTRACT

Rofiqoh Ainun Zakiyah. 2016. Support Vector Regression and Hybrid Genetic Algorithm for oil consumption forecasting. Minor Thesis of Informatics, Program of Information Technology and Computer Science, Brawijaya University, Malang. Advisor : Imam Cholissodin, S.Si, M.Kom and Dian Eka Ratnawati, S.Si, M.Kom.

Currently the most important thing to accomplish daily activities is fuel oil. If there are some trouble related fuel oil needs, it can affect production, distribution, and consumption. This kind of trouble can result in demand for oil increased, the international oil prices become volatile and domestic gas soared. By forecasting the consumption of fuel oil will be really useful for daily activities in various sectors and avoid the negative gap between the production and consumption of fuel oil. There are so many forecasting methods that have been used to predict non-linear data such as Support Vector Regression (SVR) which is proven a good result in some forecasting cases like daily energy consumption forecasting and premium fuel needs prediction. However, optimization methods can be used for more optimal result such as Hybrid Genetic Algorithm. Hybrid Genetic Algorithm optimization for forecasting using the SVR has been used in electric load forecasting research and produce a small error value. According to the result of testing performance using fuel oil consumption data 1965-2014 from www.bp.com, SVR methods with HGA optimization can produce MAPE between 3.64 to 4.65.

Keywords : Fuel Oil, SVR, HGA, MAPE



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvi
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.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Bahan Bakar Minyak	8
2.2.1 Produksi BBM.....	8
2.2.2 Kebutuhan BBM	9
2.3 Algoritma Genetika	9
2.3.1 Representasi Kromosom	10
2.3.2 Nilai <i>Fitness</i>	10
2.3.3 Seleksi.....	10
2.3.4 <i>Crossover</i>	10
2.3.5 Mutasi	11
2.4 <i>Support Vector Regression</i>	11
2.4.1 Model SVR.....	11
2.4.2 Algoritma <i>Sequential Learning</i>	11

2.5 Hybrid Algoritma Genetika	13
2.6 Metode Kernel	13
2.7 Normalisasi Data	14
2.8 Nilai Evaluasi	14
2.9 <i>Hybrid Algoritma Genetika dan Support Vector Regression</i>	15
2.9.1 Representasi Kromosom	15
2.9.2 Operator Genetika	16
2.9.3 Fungsi <i>Fitness</i>	18
BAB 3 METODOLOGI PENELITIAN	19
3.1 Studi Literatur	20
3.2 Analisa Kebutuhan	20
3.3 Pengumpulan Data	20
3.4 Perancangan Sistem	21
3.5 Implementasi	21
3.6 Perancangan Uji Coba	21
3.7 Pengambilan Kesimpulan	21
3.8 Penyelesaian Masalah Menggunakan HGA-SVR	21
3.8.2 Penyelesaian dengan SVR	24
3.8.3 Penyelesaian dengan <i>Hybrid Algoritma Genetika</i>	36
3.8.4 Proses Seleksi	43
BAB 4 PERANCANGAN SISTEM	45
4.1 Perancangan <i>User Interface</i>	45
4.1.1 Perancangan Halaman Peramalan	45
4.1.2 Perancangan Halaman Peramalan	46
4.2 Perancangan Uji Coba dan Evaluasi	47
4.2.1 Uji Coba Jumlah Iterasi SVR	47
4.2.2 Uji Coba Jumlah Generasi HGA	48
4.2.3 Uji Coba Ukuran Populasi	48
4.2.4 Uji Coba <i>Crossover Rate</i>	49
4.2.5 Uji Coba <i>Mutation Rate</i>	49
4.2.6 Uji Coba <i>Range Parameter KP1</i>	50
4.2.7 Uji Coba <i>Range Parameter KP2</i>	50



4.2.8 Uji Coba <i>Range</i> Parameter λ	51
4.2.9 Uji Coba <i>Range</i> Parameter γ	51
4.2.10 Uji Coba <i>Range</i> Parameter C	52
BAB 5 IMPLEMENTASI SISTEM	53
5.1 Implementasi Program	53
5.1.1 Implementasi Proses Normalisasi Data	53
5.1.2 Implementasi Proses <i>Sequential Learning</i>	53
5.1.3 Implementasi Proses Denormalisasi Data	55
5.1.4 Implementasi Proses Peramalan dan <i>Error Rate</i>	55
5.1.5 Implementasi Proses Representasi Kromosom	56
5.1.6 Implementasi Proses <i>Crossover</i>	57
5.1.7 Implementasi Proses Mutasi	59
5.1.8 Implementasi Proses Seleksi	59
5.2 Implementasi Antarmuka	61
5.2.1 Halaman Proses <i>Training</i>	61
5.2.2 Halaman Proses <i>Testing</i>	61
BAB 6 PENGUJIAN DAN ANALISIS	63
6.1 Hasil dan Analisa Uji Coba Jumlah Iterasi SVR	63
6.2 Hasil dan Analisa Uji Coba Jumlah Generasi HGA	64
6.3 Hasil dan Analisa Uji Coba Ukuran Populasi	66
6.4 Hasil dan Analisa Uji Coba <i>Crossover Rate</i>	67
6.5 Hasil dan Analisa Uji Coba <i>Mutation Rate</i>	68
6.6 Hasil dan Analisa Uji Coba <i>Range</i> Parameter $KP1$	69
6.7 Hasil dan Analisa Uji Coba <i>Range</i> Parameter $KP2$	71
6.8 Hasil dan Analisa Uji Coba <i>Range</i> Parameter λ	72
6.9 Hasil dan Analisa Uji Coba <i>Range</i> Parameter γ	73
6.10 Hasil dan Analisa Uji Coba <i>Range</i> Parameter C	74
BAB 7 Penutup	77
7.1 Kesimpulan	77
7.2 Saran	77
DAFTAR PUSTAKA	78

LAMPIRAN A DATA KONSUMSI BBM PERIODE 1965-2014 80

LAMPIRAN B VISUALISASI HASIL PENGUJIAN 81



DAFTAR TABEL

Tabel 2.1 Hasil Perbandingan Metode Optimasi	6
Tabel 2.2 Tabel Kajian Pustaka	6
Tabel 3.1 Tabel Data Konsumsi BBM	22
Tabel 3.2 Tabel Jumlah Kosumsi BBM.....	24
Tabel 3.3 Tabel Data Latih.....	25
Tabel 3.4 Tabel Data Uji	25
Tabel 3.5 Hasil Normalisasi Data Latih.....	26
Tabel 3.6 Hasil Normalisasi Data Uji	27
Tabel 3.7 Tabel Nilai Parameter SVR.....	28
Tabel 3.8 Hasil Perhitungan Jarak Data Latih.....	29
Tabel 3.9 Hasil Komputasi Matriks Kernel	30
Tabel 3.10 Hasil Komputasi Matriks Rij.....	31
Tabel 3.11 Nilai <i>Error</i> Iterasi 10.....	31
Tabel 3.12 Nilai $\delta\alpha_i$ dan $\delta\alpha_i^*$ Iterasi 10.....	32
Tabel 3.13 Nilai α_i dan α_i^* Iterasi 10	32
Tabel 3.14 Nilai $f(x)$ Iterasi 10	33
Tabel 3.15 Nilai $f(x)$ Iterasi 10	34
Tabel 3.16 Hasil Denormalisasi Data Latih	35
Tabel 3.17 Hasil Denormalisasi Data Uji	35
Tabel 3.18 Nilai error rate MAPE	35
Tabel 3.19 Tabel Representasi Kromosom	38
Tabel 3.20 Data Populasi Awal.....	38
Tabel 3.21 Tabel Pembangkitan σ	41
Tabel 3.22 Hasil <i>Crossover</i>	41
Tabel 3.23 Pembangkitan Nilai r	42
Tabel 3.24 Hasil Mutasi.....	43
Tabel 3.25 Hasil Perhitungan <i>Fitness</i>	44
Tabel 3.26 Hasil Evaluasi	44
Tabel 3.27 Hasil Seleksi	44

Tabel 4.1 Rancangan Uji Coba Jumlah Iterasi SVR	47
Tabel 4.2 Rancangan Uji Coba Jumlah Generasi HGA.....	48
Tabel 4.3 Rancangan Uji Coba Ukuran Populasi	48
Tabel 4.4 Rancangan Uji Coba <i>Crossover Rate</i> dan <i>Mutation Rate</i>	49
Tabel 4.5 Rancangan Uji Coba <i>Mutation Rate</i>	49
Tabel 4.6 Rancang Uji Coba <i>Range</i> Parameter <i>KP1</i>	50
Tabel 4.7 Rancang Uji Coba <i>Range</i> Parameter <i>KP2</i>	50
Tabel 4.8 Rancang Uji Coba <i>Range</i> Parameter λ	51
Tabel 4.9 Rancang Uji Coba <i>Range</i> Parameter γ	51
Tabel 4.10 Rancang Uji Coba <i>Range</i> Parameter <i>C</i>	52
Tabel 6.1 Rancangan Uji Coba Jumlah Iterasi SVR	63
Tabel 6.2 Hasil Uji Coba Jumlah Generasi HGA.....	65
Tabel 6.3 Hasil Uji Coba Ukuran Populasi	66
Tabel 6.4 Hasil Uji Coba <i>Crossover Rate</i>	67
Tabel 6.5 Hasil Uji Coba <i>Mutation Rate</i>	69
Tabel 6.6 Hasil Uji Coba <i>Range</i> Parameter <i>KP1</i>	70
Tabel 6.7 Hasil Uji Coba <i>Range</i> Parameter <i>KP2</i>	71
Tabel 6.8 Hasil Uji Coba <i>Range</i> Parameter λ	72
Tabel 6.9 Hasil Uji Coba <i>Range</i> Parameter γ	74
Tabel 6.10 Hasil Uji Coba <i>Range</i> Parameter <i>C</i>	75

DAFTAR GAMBAR

Gambar 2.1 Grafik Konsumsi dan Produksi	8
Gambar 2.2 Struktur Gen Pada HGA.....	16
Gambar 2.3 Operator Genetika HGA	17
Gambar 3.1 Diagram Alir Metode Penelitian.....	19
Gambar 3.2 Blok Diagram Proses HGA-SVR.....	23
Gambar 3.3 Diagram Alir Proses Normalisasi Data BBM	26
Gambar 3.4 Diagram Alir <i>Sequential Learning</i> SVR.....	27
Gambar 3.5 Diagram Alir Proses Hitung Matriks Kernel.....	29
Gambar 3.6 Diagram Alir Matriks R_{ij}	30
Gambar 3.7 Diagram Alir Cek Maksimum.....	33
Gambar 3.8 Proses Denormalisasi Data.....	34
Gambar 3.9 Proses HGA.....	36
Gambar 3.10 Representasi Kromosom HGA.....	37
Gambar 3.11 Flowchart Proses Crossover	39
Gambar 3.12 Flowchart Proses Adewuya's Crossover	40
Gambar 3.13 Diagram Alir Proses Mutasi.....	42
Gambar 3.14 Subproses mutasi.....	42
Gambar 3.15 Diagram Alir Hitung <i>fitness</i>	43
Gambar 4.1 Perancangan Halaman Perancangan	45
Gambar 4.2 Perancangan Halaman Perancangan	46
Gambar 5.1 Halaman proses <i>training</i>	61
Gambar 5.2 Halaman proses testing.....	62
Gambar 6.1 Grafik Uji Iterasi SVR	64
Gambar 6.2 Grafik Waktu Komputasi Jumlah Iterasi SVR	64
Gambar 6.3 Grafik Uji Generasi HGA	65
Gambar 6.4 Grafik Uji Populasi	67
Gambar 6.5 Grafik Uji Crossover Rate	68
Gambar 6.6 Grafik Uji Mutation Rate	69
Gambar 6.7 Grafik Uji Range Parameter $KP1$	70
Gambar 6.8 Grafik Uji Range Parameter $KP2$	72

Gambar 6.9 Grafik Uji Range Parameter λ 73

Gambar 6.10 Grafik Uji Range Parameter γ 74

Gambar 6.11 Grafik Uji Range Parameter C..... 76



DAFTAR LAMPIRAN

LAMPIRAN A DATA KONSUMSI BBM PERIODE 1965-2014	80
A.1 Data Konsumsi BBM Periode 1965-2014	80
LAMPIRAN B VISUALISASI HASIL PENGUJIAN	81
B.1 Nilai Parameter Optimal untuk Peramalan Konsumsi Bahan Bakar Minyak di Indonesia Dengan Metode <i>Hybrid</i> Algen - SVR.....	81
B.2 Nilai α^* dan α Optimal untuk Fungsi Regresi pada SVR	81
B.3 Visualisasi Hasil Uji Coba Jumlah Iterasi SVR.....	82
B.4 Visualisasi Hasil Uji Coba Banyak Generasi.....	82
B.5 Visualisasi Hasil Uji Coba Ukuran Populasi	83
B.6 Visualisasi Hasil Uji Coba <i>Crossover Rate</i>	83
B.7 Visualisasi Hasil Uji Coba <i>Mutation Rate</i>	84
B.8 Visualisasi Hasil Uji Coba Range Parameter <i>KP1</i>	84
B.9 Visualisasi Hasil Uji Coba Range Parameter <i>KP2</i>	85
B.10 Visualisasi Hasil Uji Coba Range Parameter <i>Lambda</i> (λ)	85
B.11 Visualisasi Hasil Uji Coba Range Parameter <i>Gamma</i> (γ).....	86
B.12 Visualisasi Hasil Uji Coba Range Parameter <i>C</i>	86



BAB 1 PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan.

1.1 Latar belakang

Bahan bakar minyak merupakan salah satu kebutuhan yang sangat penting untuk menjalankan berbagai macam aktivitas dalam kehidupan sehari-hari. Banyaknya masyarakat yang telah membeli kendaraan untuk menunjang aktivitas mereka mengakibatkan cadangan jumlah produksi bahan bakar minyak menjadi menipis. Pertamina telah mengimpor minyak pada harga internasional dan menjual pada konsumen dengan harga bersubsidi, jika terjadi keterbatasan dalam memenuhi kebutuhan bahan bakar minyak dapat mengakibatkan kegiatan produksi, distribusi dan konsumsi menjadi terlambat. Hal tersebut bisa mengakibatkan terjadinya *demand for oil* yang meningkat, harga minyak internasional menjadi berubah-ubah dan gas dalam negeri yang melambung tinggi. Menurut data dari *Statistical Review of World Energy 2015*, jumlah produksi minyak di Indonesia beberapa tahun terakhir mengalami penurunan dari tahun 2010 sebesar 1003.01758575342 barrels, 2011 sebesar 952.284732613698, hingga 2014 sebesar 852.267507441095. Sedangkan jumlah konsumsi minyak di Indonesia beberapa bulan terakhir mengalami peningkatan dari tahun 2008 hingga 2014 (Desvina & Siddiq, 2011).

Berdasarkan penjelasan mengenai jumlah produksi dan jumlah konsumsi minyak di Indonesia, bisa dikatakan bahwa Indonesia saat ini mengalami kesenjangan negatif yakni jumlah konsumsi minyak lebih tinggi daripada jumlah produksi minyak. Sumber energi terbarukan yang ada di Indonesia seperti biodiesel dari tanaman jarak pagar dapat dimanfaatkan untuk mengurangi jumlah konsumsi bahan bakar minyak di Indonesia. Namun, sumber energi tersebut mengakibatkan adanya penumpukan CO₂ yang akan meningkatkan temperatur udara bumi. Selain itu, dengan meningkatkan harga BBM untuk mengurangi jumlah konsumsi BBM mengakibatkan kendala ekonomi bagi masyarakat miskin. Oleh karena itu, seiring dengan berlakunya kebijakan mengenai bahan bakar baru dan kenaikan harga BBM, hendaknya dibuat sebuah pemodelan untuk peramalan jumlah konsumsi BBM di Indonesia. Hal ini bertujuan agar masyarakat dan pemerintah mampu menyeimbangkan jumlah produksi BBM, jumlah konsumsi BBM, produksi bahan bakar baru, maupun kenaikan harga BBM.

Penelitian yang telah dilakukan terkait dengan permasalahan konsumsi minyak adalah penelitian yang dilakukan oleh Oktavianis Kusumaningrum (2012). Pada penelitian tersebut didapatkan hasil bahwa metode Hibrida dapat memprediksi lebih baik dibandingkan metode tunggal. Kombinasi metode ARIMA dan *Neural Network* dapat bekerja sama dengan baik untuk menghasilkan peramalan lebih baik dengan nilai *root mean square error* (RMSE) yang kecil dibandingkan dengan

model ARIMA dan NN secara individu. Nilai RMSE yang didapatkan pada *out-sample* untuk pemilihan ramalan terbaik selama 12 bulan dengan metode ARIMA-NN adalah 988, metode ARIMA sebesar 1379, sedangkan metode NN menghasilkan 1108 (Kusumaningrum & Suhartono, 2012).

Penelitian kedua terkait metode *Support Vector Regression* telah dilakukan oleh E Ying (2012) dengan objek penelitian pada perekrutan siswa. Hasil penelitian mengenai perbandingan jumlah *error* antara SVR dan BP *Neural Network* yang menunjukkan bahwa dengan penelitian menggunakan SVR bisa menghasilkan hasil yang paling mendekati dengan objek penelitian (Ying, 2012).

Penelitian ketiga terkait metode *Hybrid Algoritma Genetika* dilakukan oleh Mithun M. Bhaskar (2011). Hasil penelitian yang telah dilakukan menunjukkan bahwa dengan menggunakan metode *Hybrid Algoritma Genetika* dapat menghasilkan nilai parameter yang optimal dibandingkan dengan menggunakan metode optimasi lain seperti yang ada didalam penelitian yakni algoritma genetik sederhana (GA), algoritma genetik adaptif (AGA), evolusi diferensial (DE), optimasi partikel swarm (PSO) dan pencarian harmoni berbasis music (MBHS) (Bhaskar & Maheswarapu, 2011).

Penelitian yang terakhir yakni terkait dengan metode *Support Vector Regression* dan *Hybrid Algoritma Genetika* telah dilakukan oleh Chih-Hung Wu (2009). Penelitian tersebut terfokus pada pendekatan metode yang digunakan yakni penggabungan antara metode HGA-SVR dalam melakukan peramalan. Hasil yang didapatkan dari penelitian tersebut adalah perbandingan nilai *root mean square error* (RMSE) metode GA-SVR dan HGA-SVR yakni 9.46 : 7.73. Nilai RMSE yang dihasilkan menunjukkan bahwa peramalan menggunakan HGA-SVR dapat menghasilkan nilai *error rate* peramalan yang rendah (Wu, Tzeng, & Lin, 2009).

Berdasarkan pada penelitian yang terakhir yakni peramalan menggunakan metode HGA-SVR yang mendapatkan hasil terbaik, maka sebuah sistem peramalan akan dirancang menggunakan HGA-SVR dengan objek konsumsi BBM. Penggunaan metode *Support Vector Regression* pada penelitian ini didasarkan pada kemampuannya dalam menangani data dimensi tinggi pada fase *training* dan memiliki keakuratan yang tinggi (Pasolli et al., 2012) (Ying, 2012). Metode HGA digunakan karena metode tersebut dapat memberikan hasil yang optimum dan mengurangi waktu dalam pencarian solusi (Amelia, Regasari, & Priharsari, 2014).

1.2 Rumusan masalah

Berdasarkan uraian latar belakang tersebut, maka dirumuskan masalah sebagai berikut:

1. Bagaimana mengimplementasikan optimasi *Hybrid Algoritma Genetika* dan metode *Support Vector Regression* untuk peramalan konsumsi BBM?
2. Berapa nilai *error rate* peramalan yang dapat dihasilkan oleh metode *Hybrid Algoritma Genetika* dan *Support Vector Regression* (SVR)?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Untuk mengimplementasikan optimasi *Hybrid* Algoritma Genetika dan metode *Support Vector Regression* untuk peramalan konsumsi BBM.
2. Untuk mengetahui nilai *error rate* peramalan yang dapat dihasilkan oleh metode *Hybrid* Algoritma Genetika dan *Support Vector Regression* (SVR).

1.4 Manfaat

Penelitian ini diharapkan dapat bermanfaat untuk berbagai pihak. Manfaat dari penelitian ini adalah sebagai berikut:

- Bagi Penulis
Penulis mendapatkan pemahaman tentang penggunaan metode *Hybrid* Algoritma Genetika dan *Support Vector Regression* untuk peramalan konsumsi BBM di Indonesia.
- Bagi Masyarakat
Memberikan solusi kepada masyarakat Indonesia dalam menghadapi siklus konsumsi BBM diwaktu yang akan datang.
- Bagi Pemerintah
Memberikan solusi kepada pemerintah Negara Indonesia agar dapat lebih bijaksana dalam memutuskan kebijakan produksi BBM dalam negeri kedepannya.

1.5 Batasan masalah

Agar pembahasan penelitian ini tidak menyimpang dari apa yang telah dirumuskan, maka diperlukan batasan-batasan. Batasan-batasan dalam penelitian ini adalah :

1. Obyek yang digunakan adalah konsumsi bahan bakar minyak di indonesia tahun 1965-2014 yang didapatkan dari *website* resmi *Statistical Review of World Energy* www.bp.com. Data yang dibuat acuan konsumsi bahan bakar minyak di indonesia tahun 1965-2014.
2. *Output* yang dihasilkan oleh sistem adalah *value forecasting error*, dengan asumsi semakin kecil *value forecasting error* maka hasil peramalan data tersebut semakin baik.

1.6 Sistematika pembahasan

Untuk mencapai tujuan yang sudah dikemukakan diatas, maka sistematika penulisan yang disusun dalam penelitian ini adalah sebagai berikut:

BAB I : Pendahuluan

Bab ini menjelaskan mengenai latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan.

BAB II : Landasan Kepustakaan

Bab ini menjelaskan kajian pustaka dan dasar teori yang menjadi acuan dalam penulisan skripsi yang diperoleh dari beberapa literatur, jurnal, dan halaman website.

BAB III : Metodologi Penelitian

Bab ini berisi metode atau langkah-langkah yang digunakan dalam penelitian yang terdiri dari studi literatur, analisa kebutuhan, pengumpulan data, perancangan perangkat lunak, implementasi, perancangan uji coba, dan pengambilan keputusan.

BAB IV : Perancangan Sistem

Bab ini berisi perancangan yang terdiri dari perancangan antarmuka, serta perancangan uji coba dan evaluasi.

BAB V : Implementasi Sistem

Bab ini berisi penjelasan tentang teknis implementasi, batasan-batasan implementasi, serta algoritma operasi yang digunakan untuk mengembangkan sistem.

BAB VI : Pengujian dan Analisis

Bab ini menjelaskan tentang tingkat akurasi dan analisa hasil terhadap metode yang digunakan.

BAB VII : Penutup

Bab ini berisi kesimpulan dan saran atas kekurangan di dalamnya sehingga dapat dikembangkan menjadi penelitian yang lebih baik.

BAB 2 LANDASAN KEPUSTAKAAN

Secara umum bab ini berisi kajian pustaka dan dasar teori yang digunakan dalam penelitian. Dasar teori difokuskan pada bahan bakar minyak sebagai objek penelitian, metode *hybrid* algoritma genetika dan *Support Vector Regression* sebagai algoritma yang akan diimplementasikan dalam penelitian.

2.1 Kajian Pustaka

Dalam penulisan skripsi ini, pada bagian Kajian Pustaka akan dijabarkan mengenai penelitian – penelitian yang relevan terhadap judul yang diajukan. Penelitian yang dikaji meliputi pemilihan fitur dan optimasi terhadap SVR hingga kinerja algoritma yang dipengaruhi oleh parameter SVR dan kajian mengenai HGA sebagai metode yang digunakan sebagai metode optimasi yang handal.

Untuk objek peramalan dan prediksi kebutuhan bahan bakar telah dibahas pada penelitian sebelumnya. Salah satunya adalah penelitian yang dengan judul "*Peramalan Kebutuhan Bahan Bakar Premium di Depot Ampenan dengan Metode Hibrida Arima Neural Network untuk Optimasi Persediaan*". Penelitian tersebut dilakukan karena kebutuhan wilayah dan permintaan BBM yang cenderung tidak konstan. Penelitian tersebut menunjukkan bahwa kombinasi metode ARIMA dan *Neural Network* dapat menghasilkan peramalan yang bagus dengan nilai RMSE yang kecil dibandingkan model ARIMA dan NN secara individu. Nilai RMSE yang didapatkan untuk peramalan konsumsi minyak selama 12 bulan dengan metode ARIMA-NN adalah 988, metode ARIMA sebesar 1379, sedangkan metode NN menghasilkan 1108 (Kusumaningrum & Suhartono, 2012).

Salah satu penelitian yang membahas tentang peramalan dengan metode *Support Vector Regression* adalah penelitian dengan judul "*Application of Support Vector Regression Algorithm in Colleges Recruiting Students Prediction*". Permasalahan nonlinier tersebut dimodelkan dengan metode *Support Vector Regression* dan *Backpropagation Neural Network*. Hasil perbandingan jumlah peramalan error antara metode SVR dan BP *Neural Network* menunjukkan bahwa metode SVR memiliki tingkat peramalan yang paling mendekati data sesungguhnya dibandingkan BP *neural network* (Ying, 2012).

Penelitian ketiga terkait metode *Hybrid* Algoritma Genetika dilakukan oleh Mithun M. Bhaskar (2011) untuk optimasi aliran daya. Hasil penelitian yang telah dilakukan menunjukkan bahwa dengan menggunakan metode *Hybrid* Algoritma Genetika dapat menghasilkan nilai parameter yang optimal dibandingkan dengan menggunakan metode optimasi lain seperti yang ada didalam penelitian yakni algoritma genetik sederhana (GA), algoritma genetik adaptif (AGA), evolusi diferensial (DE), optimasi partikel swarm (PSO) dan pencarian harmoni berbasis music (MBHS). Tabel 2.1 menunjukkan hasil dari perbandingan metode optimasi yang digunakan (Bhaskar & Maheswarapu, 2011).

Tabel 2.1 Hasil Perbandingan Metode Optimasi

Metode	Biaya Bahan Bakar	Kerugian	Running time/iterasi	Iterasi
GA	802.359	9.60	0.488	40
PSO	802.667	9.59	1.737	70
DE	801.114	9.54	1.236	46
AGA	802.700	9.44	1.25	24
HGA	800.801	9.23	0.620	37

Penelitian yang terakhir yakni terkait dengan metode *Support Vector Regression* dan *Hybrid Algoritma Genetika* telah dilakukan oleh Chih-Hung Wu (2009). Penelitian tersebut menghasilkan perbandingan dari nilai RMSE metode GA-SVR dan HGA-SVR yakni 9.46 : 7.73. Kernel yang digunakan pada GA-SVR adalah *Radial Basis Function* sedangkan pada HGA-SVR menggunakan *Polynomial* kernel. HGA-SVR menggunakan Poly kernel karena sebelumnya fungsi kernel beserta parameter dari masing-masing kernel sudah mengalami proses optimasi untuk mendapatkan nilai yang optimal (Wu, Tzeng, & Lin, 2009).

Pada penelitian ini, akan digunakan metode *Hybrid Algoritma Genetika* dan *Support Vector Regression* sebagai solusi untuk melakukan peramalan konsumsi bahan bakar minyak (BBM). Model regresi nonlinier yang dihasilkan nantinya akan digunakan menjadi sebuah persamaan yang digunakan untuk mencari peramalan konsumsi BBM di tahun berikutnya. Perbandingan objek dan metode penelitian dari masing-masing referensi akan ditunjukkan pada Tabel 2.2.

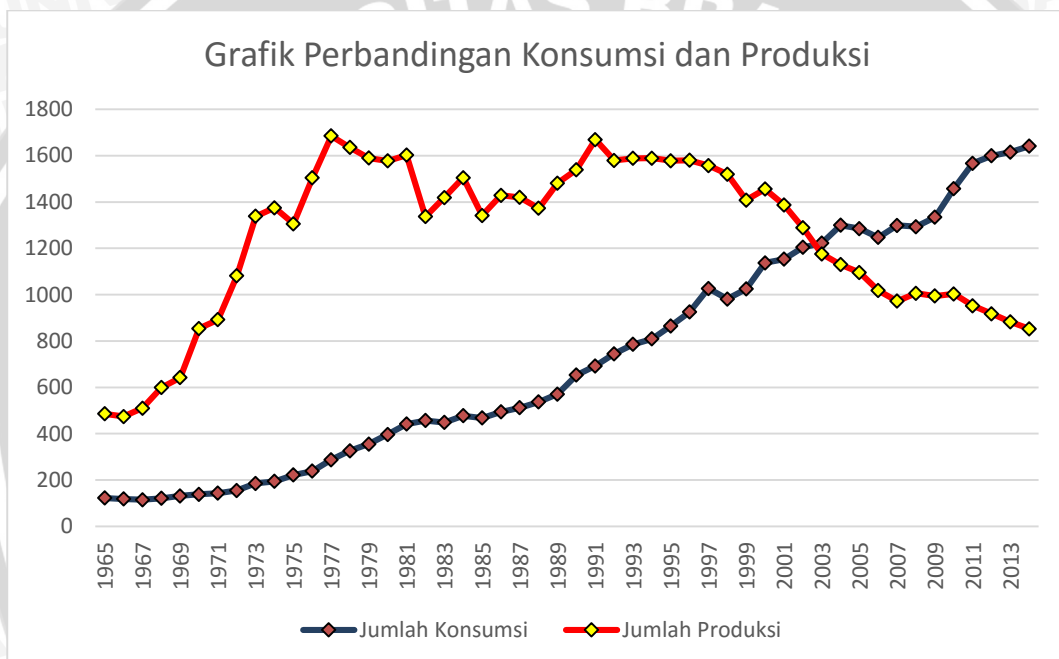
Tabel 2.2 Tabel Kajian Pustaka

No	Judul	Objek	Metode	Hasil
1.	Peramalan Kebutuhan Bahan Bakar Premium di Depot Ampenan dengan Metode Hibrida Arima-Neural Network untuk Optimasi Persediaan (Kusumaningrum & Suhartono, 2012)	Jumlah konsumsi bahan bakar	Metode Hibrida Arima-Neural Network	Hasil perbandingan model ARIMA, Neural Network dan Hibrida ARIMA-NN menunjukkan bahwa model Hibrida ARIMA-NN menghasilkan nilai RMSE sebesar 988 terkecil dibandingkan dengan model tunggal yaitu ARIMA sebesar 1379 dan NN sebesar 1108.

2.	<i>Application of support vector regression algorithm in colleges recruiting students prediction</i> (Ying, 2012)	Data mahasiswa dari provinsi sichuan dari tahun 2000 sampai 2008	<i>Support Vector Regression</i>	Penerapan metode SVR yang telah digunakan mendapatkan hasil peramalan dengan akurasi yang tinggi dibandingkan dengan BP <i>neural network</i> .
3.	<i>A Hybrid Genetic Algorithm Approach for Optimal Power Flow</i> (Bhaskar & Maheswarapu, 2011)	Aliran daya	<i>Hybrid Algoritma Genetika</i>	Penggunaan metode HGA merupakan algoritma cepat dan mampu menghasilkan nilai parameter yang optimal dibanding Algoritma Genetik Adaptif (AGA) dan <i>Particle Swam Optimization</i> (PSO).
4.	<i>A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression</i> (Wu, Tzeng, & Lin, 2009)	Penggunaan Listrik	<i>Hybrid Algoritma Genetika dan Support Vector Regression</i>	Penelitian tersebut menghasilkan perbandingan dari nilai RMSE metode GA-SVR dan HGA-SVR yakni 9.46 : 7.73
5.	Usulan Penulis: Peramalan konsumsi bahan bakar minyak (BBM) dengan metode <i>Hybrid Algoritma Genetika dan Support Vector Regression</i>	Konsumsi BBM	<i>Hybrid Algoritma Genetika dan Support Vector Regression</i>	Proses algoritma yang digunakan dalam sistem mampu menghasilkan peramalan dengan nilai MAPE terkecil.

2.2 Bahan Bakar Minyak

Bahan Bakar Minyak (BBM) merupakan salah satu kebutuhan yang harus terpenuhi dalam kehidupan sehari-hari baik rumah tangga ataupun dunia kerja. Seiring dengan meningkatnya aktivitas pembangunan diberbagai sektor yang ditunjukkan dengan peningkatan mobilitas barang maupun orang mengakibatkan kebutuhan bahan bakar minyak terus meningkat (Sunoto, 2006). Indonesia resmi menjadi negara pengimpor minyak sejak tahun 2004. Hal ini dikarenakan menurunnya tingkat produksi minyak disatu sisi sementara disisi lainnya tingkat konsumsi minyak terus meningkat yang ditunjukkan pada Gambar 2.1. Salah satu produk bahan bakar minyak (BBM) yang paling krusial adalah premium. Premium merupakan satu dari tiga produk bahan bakar bersubsidi yang paling banyak diminati oleh masyarakat (Farizal et al., 2014).



Gambar 2.1 Grafik Konsumsi dan Produksi

2.2.1 Produksi BBM

Berdasarkan teknologi yang ada sebelum tahun 1970, kilang yang dibangun adalah kilang yang memiliki kemampuan rendah (*low processing*) dimana kilang yang dibangun pada saat itu hanya digunakan untuk mengolah minyak ringan. Sedangkan setelah tahun 1970 kilang yang dibangun merupakan kilang dengan kemampuan tinggi (*high processing*) dengan tujuan untuk mengolah minyak berat yang berasal dari sumur local maupun dari timur tengah. Sebagian besar kilang yang dimiliki Indonesia saat ini berumur 30 tahun mengingat selama 20 tahun Indonesia tidak melakukan penambahan kilang minyak baru di Indonesia.

Berdasarkan keadaan kilang yang ada di Indonesia, pemerintah saat ini mulai merencanakan untuk membatasi BBM bersubsidi. Disamping itu, pemerintah juga harus mempertimbangkan untuk *impor* BBM non subsidi karena BBM non subsidi yang di produksi di Indonesia terbatas tiap tahunnya. Dengan ketersediaan infrastruktur kilang nasional saat ini, produksi BBM oleh nasional baru dapat memenuhi sekitar 56% kebutuhan BBM nasional. Ditambah dengan teknologi kilang yang sudah tua mengakibatkan efisiensi kilang semakin lama semakin menurun. Kondisi tersebut dapat mengakibatkan ketergantungan Indonesia terhadap impor BBM semakin besar dan cenderung merugikan Indonesia mengingat harga BBM impor yang dibeli oleh Indonesia cukup mahal.

Rencana pembangunan kilang sebetulnya sudah dilontarkan sejak 8 tahun lalu, tepatnya Desember 2005. Namun sampai saat ini belum ada satu pun kilang minyak baru yang berhasil dibangun. Mundurnya rencana pembangunan kilang di Indonesia banyak disebabkan oleh kecilnya insentif yang dapat diberikan oleh Pemerintah kepada investor disamping masalah lain seperti sulitnya pembebasan lahan.

2.2.2 Kebutuhan BBM

Peningkatan kebutuhan BBM tertinggi terjadi pada sektor transportasi, hal ini diperkirakan disebabkan karena peningkatan jumlah kendaraan yang cukup tinggi, peningkatan mobilitas perjalanan karena jarak tempat tinggal yang semakin menjauh dari tempat kerja, kemacetan yang semakin padat, ditambah harga BBM yang cenderung masih murah. Peningkatan penggunaan BBM juga terjadi untuk sektor pembangkit akibat masih adanya beberapa pembangkit yang seharusnya menggunakan gas masih kesulitan untuk mendapatkan bahan bakar gas sehingga terpaksa masih menggunakan BBM.

Untuk mengatasi over kuota dan mengendalikan konsumsi BBM bersubsidi, pada tahun 2011 Pemerintah telah menyusun sejumlah program-program antara lain: pembatasan kategori pengguna BBM bersubsidi serta pembatasan volume secara bertahap dan pengendalian penggunaan BBM bersubsidi melalui sistem distribusi dan penyempurnaan regulasi. Akan tetapi hingga tahun 2012 program-program tersebut masih belum dapat berjalan optimal. Hal lain yang perlu mendapat perhatian adalah pengawasan dan pengendalian Badan Usaha pelaksana penyalur BBM bersubsidi baik itu Pertamina, maupun Badan Usaha lainnya.

2.3 Algoritma Genetika

Algoritma genetika (*Genetic Algorithms*) merupakan tipe algoritma evolusi yang paling populer. Algoritma genetika berkembang seiring dengan perkembangan teknologi informasi yang sangat pesat. Kemampuannya dalam menyelesaikan berbagai masalah kompleks, mengakibatkan algoritma ini banyak digunakan dalam berbagai bidang yang mengalami permasalahan optimasi dan model matematika yang sulit direpresentasikan.

Proses dalam algoritma genetika dimulai dengan tahap inisialisasi, yaitu menciptakan individu – individu secara acak yang memiliki susunan gen (kromosom) tertentu. Kromosom ini mewakili solusi dari permasalahan. Tahap selanjutnya adalah reproduksi untuk menghasilkan *offspring* dari individu yang ada dipopulasi. Setelah reproduksi akan lahir individu baru sehingga jumlah individu bertambah. Setiap kromosom mempunyai *fitness*, makin besar *fitness* makin baik kromosom tersebut untuk dijadikan solusi. Tahap menghitung *fitness* ini disebut tahap evaluasi. Tahap akhir adalah seleksi yaitu memilih individu dari himpunan populasi dan *offspring*. Individu hasil seleksi dipertahankan hidup pada generasi berikutnya (Rahmi et al., 2015).

2.3.1 Representasi Kromosom

Kromosom tersusun dari sejumlah gen yang merepresentasikan variabel-variabel solusi. Representasi kromosom pada Algoritma Genetika disertai dengan fungsi *fitness* yang menyatakan kebaikan dari solusi. Representasi kromosom diperlukan untuk menjelaskan individu dalam populasi. Setiap kromosom tersusun atas urutan gen dari suatu alfabet yang dapat terdiri dari biner (0 dan 1), *floating point*, *integer*, simbol-simbol, matriks, dan lain sebagainya (Mahmudy, 2006).

2.3.2 Nilai *Fitness*

Nilai *fitness* adalah nilai yang menyatakan baik tidaknya suatu individu. Nilai *fitness* digunakan sebagai acuan dalam mencapai nilai optimal pada algoritma genetika yang bertujuan untuk mendapatkan nilai *fitness* paling tinggi.

2.3.3 Seleksi

Seleksi merupakan proses untuk memilih individu yang akan menjadi induk pada proses reproduksi. Langkah pertama yang dilakukan adalah mencari nilai *fitness* yang akan digunakan pada tahap berikutnya. Semakin tinggi nilai *fitness* suatu individu maka semakin besar pula kemungkinannya untuk terpilih (Mahmudy, 2006). Masing-masing individu akan memiliki probabilitas yang tergantung pada nilai objektif masing-masing individu terhadap semua individu yang ada pada wadah seleksi tersebut.

2.3.4 *Crossover*

Crossover (kawin silang) merupakan operator dari algoritma genetika yang melibatkan dua induk untuk membentuk suatu kromosom yang baru. *Crossover* dilakukan untuk menghasilkan titik baru yang siap diuji. Individu dipilih secara acak untuk dilakukan *crossing*. Namun jika *crossover* tidak dilakukan, maka nilai yang dimiliki induk akan diturunkan kepada kromosom turunannya. Prinsip dari *crossover* adalah melakukan operasi (pertukaran, aritmatika) pada gen-gen yang bersesuaian dari dua induk untuk menghasilkan individu baru (Mahmudy, 2006) (Mahmudy, 2008).

2.3.5 Mutasi

Mutasi gen merupakan operator pada algoritma genetika yang berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Peluang mutasi didefinisikan sebagai preentase jumlah total gen yang mengendalikan banyaknya gen baru yang muncul pada populasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin dievaluasi. Tetapi jika terlalu besar kemungkinan kemiripan dengan induknya akan hilang (Mahmudy, 2006) (Mahmudy, 2008).

2.4 Support Vector Regression

Support Vector Regression (SVR) adalah sebuah model nonlinier yang digunakan untuk menyelesaikan permasalahan regresi. Proses pelatihan SVR mirip dengan memecahkan masalah pemrograman kuadrat linier terkendala yang menyediakan nilai optimal yang unik dan tidak ada masalah minimum lokal (Alwee et al., 2013). Melalui pemetaan nonlinier, SVR mampu melakukan pemetaan fitur ruang dimensi standar menjadi fitur ruang berdimensi tinggi (Li & Suohai, 2013)

2.4.1 Model SVR

Nilai *bias* dapat diformulasikan ke dalam bentuk lain yaitu dengan menggunakan penambahan dimensi untuk vektor masukan tambahan λ ditunjukkan pada Persamaan 2.1 (Vijayakumar, 1999).

$$f(x) = \sum_{i=1}^l (\alpha_i^* - \alpha_i)(K(x_i, x) + \lambda^2). \quad (2.1)$$

Keterangan :

- $(\alpha_i^* - \alpha_i)$. merupakan *lagrange multiplier*
- $K(x_i, x)$ merupakan matriks kernel
- λ merupakan variabel skalar

Terjadi perbedaan pendefinisian *Lagrange multiplier* (α dan α^*) pada fungsi tidak linier yang digunakan Vijayakumar (1999) dan Smola & Scholkopf (2004). Hal ini terjadi karena bergantung pada inisialisasi letak α dan α^* pada grafik fungsi *loss* SVR.

2.4.2 Algoritma *Sequential Learning*

Pada setiap perhitungan fungsi SVR, terdapat proses *sequential learning*. Algoritma *sequential* untuk regresi dapat dilihat di bawah ini (Vijayakumar & Wu, 1999) :

1. Inisialisasi $\alpha_i = 0, \alpha_i^* = 0$. Kemudian dilakukan perhitungan matriks *hessian* yang ditunjukkan pada Persamaan 2.2.

$$[R]_{ij} = K(x_i, x_j) + \lambda^2 \text{ untuk } i, j = 1, \dots, l \quad (2.2)$$

Keterangan:

R_{ij} = Matriks *Hessian*

λ^2 = Variabel Skalar

l = Banyaknya data

2. Untuk masing-masing data training $i, j = 1, \dots, l$, lakukan

2.1 Hitung nilai *error* menggunakan Persamaan 2.3.

$$E_i = y_i - \sum_{j=1}^l (\alpha_j^* - \alpha_j) R_{ij} \quad (2.3)$$

Keterangan:

y_i = Nilai kelas *support vector* (berupa nilai aktual data uji)

E_i = Nilai *error* ke- i

2.2 Hitung $\delta\alpha_i$ dan $\delta\alpha_i^*$ menggunakan Persamaan 2.4 dan 2.5.

$$\delta\alpha_i^* = \min\{\max[\gamma(E_i - \varepsilon), -\alpha_i^*], C - \alpha_i^*\} \quad (2.4)$$

$$\delta\alpha_i = \min\{\max[\gamma(-E_i - \varepsilon), -\alpha_i], C - \alpha_i\} \quad (2.5)$$

Keterangan:

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

ε = Nilai epsilon

C = Nilai kompleksitas

γ = Nilai *learning rate* yang didapatkan dari Persamaan 2.6.

$$\gamma = \frac{\text{konstanta learning rate (cLR)}}{\max(\text{diagonal matriks } R_{ij})} \quad (2.6)$$

2.3 Update nilai α_i dan α_i^* dengan Persamaan 2.7 dan 2.8.

$$\alpha_i^* = \alpha_i^* + \delta\alpha_i^* \quad (2.7)$$

$$\alpha_i = \alpha_i + \delta\alpha_i \quad (2.8)$$

3. Jika proses telah mencapai pada iterasi yang telah ditentukan atau $\max|\delta\alpha_i^*| < \varepsilon$ dan $\max|\delta\alpha_i| < \varepsilon$, maka algoritma selesai. Jika tidak maka ulangi langkah kedua.

4. Lakukan pengujian dengan fungsi regresi seperti yang telah dijelaskan pada Persamaan 2.1.

5. Selesai.

2.5 Hybrid Algoritma Genetika

Banyak penelitian yang telah membuktikan bahwa standar algoritma algoritma genetika sering kali lebih mengarah pada solusi lokal optimum, khususnya pada masalah-masalah yang cukup besar. Atas dasar tersebut, beberapa peneliti mengkombinasikan metode-metode heuristik lain kedalam algoritma genetika atau biasa disebut *hybrid* algoritma genetika dengan harapan mampu meningkatkan kinerja algoritma genetika. *Hybrid* algoritma genetika merupakan gabungan dari algoritma genetik dengan pencarian lokal untuk menghasilkan solusi yang lebih optimal dengan waktu komputasi yang lebih cepat (Sempena, 2010).

Hibridisasi *hybrid* algoritma genetika dengan teknik *local search* menghasilkan *memetic algorithms*. Algoritma *hill-climbing* merupakan teknik *local search* sederhana yang bias dipakai pada optimasi fungsi tanpa kendala (Mahmudy, Marian, & Luong, 2014). Proses hibridisasi dengan algoritma meta-heuristik juga bias dilakukan misalnya hibridisasi *real-coded* GA (RCGA) dengan *variable neighbourhoods search* (VNS) untuk penyelesaian permasalahan *part type selection* dan *machine learning* pada *flexible manufacturing system* (FMS) (Mahmudy W. , 2013).

2.6 Metode Kernel

Metode kernel merupakan sebuah teknik yang bisa digunakan untuk membangun sebuah fitur-fitur untuk permasalahan nonlinier yakni memetakan *input space* dan *feature space*. Metode kernel yang sering digunakan adalah (Wu et al., 2009):

- *Radial Basis Function* (RBF) kernel.

Perhitungan kernel RBF akan dijabarkan pada Persamaan 2.9.

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (2.9)$$

Dimana σ^2 merupakan varian *gaussian* kernel.

- *Polinomial* kernel.

Rumus perhitungan kernel polynomial akan ditunjukkan pada Persamaan 2.10.

$$k(x_i, x_j) = (t + x_i^T x_j)^d \quad (2.10)$$

Dimana d merupakan derajat polinomial, t merupakan parameter polinomial.

- *Linear* kernel.

Persamaan 2.11 merupakan rumus yang digunakan pada *Linear* kernel.

$$k(x_i, x_j) = x_i^T x_j \quad (2.11)$$

2.7 Normalisasi Data

Proses *preprocessing* data yang dilakukan pada Peramalan konsumsi BBM berupa normalisasi data. Hal ini bertujuan untuk standarisasi semua data yang digunakan dalam perhitungan sehingga data berada pada jarak tertentu (Patro et al., 2015). Proses normalisasi yang digunakan pada penelitian ini adalah *Min-Max normalization*. Perhitungan *Min-Max normalization* ditunjukkan pada Persamaan 2.12:

$$x' = \frac{(x - x_{\min})}{x_{\max} - x_{\min}} \quad (2.12)$$

Keterangan:

- x' = Hasil normalisasi data yang nilainya berkisar antara 0 dan 1
- x = Nilai data yang akan dinormalisasi
- x_{\max} = Nilai maksimum dari dataset yang digunakan
- x_{\min} = Nilai minimum dari dataset yang digunakan

Sedangkan untuk proses denormalisasi data adalah proses untuk mengembalikan nilai data menjadi nilai riil (seperti keadaan sebelum dinormalisasi) yaitu dengan mencari nilai x . Proses pencarian nilai x dapat dijabarkan melalui Persamaan 2.13 yang merupakan *reverse* dari Persamaan 2.12.

$$x = (x'(x_{\max} - x_{\min})) + x_{\min} \quad (2.13)$$

2.8 Nilai Evaluasi

Dalam regresi ada beberapa ukuran *error* yang sering dipakai untuk menilai performansi suatu fungsi prediksi. Nilai evaluasi digunakan jika proses peramalan sudah mencapai batas iterasi, iterasi bisa dikatakan bagus ketika nilai evaluasi yang didapatkan adalah paling kecil mendekati 0. Jika y_i menyatakan nilai prediksi untuk data ke- i dan \hat{y}_i adalah nilai *output* aktual data ke- i dan m adalah banyaknya data, maka beberapa ukuran *error* yang sering dipakai adalah (Sari, 2009):

1. *Mean squared error* (MSE).

Rumus yang digunakan untuk menghitung nilai MSE dapat dilihat pada Persamaan 2.14.

$$MSE = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 \quad (2.14)$$

2. *Mean absolute deviation (MAD).*

Rumus yang digunakan untuk menghitung nilai MAD dapat dilihat pada Persamaan 2.15.

$$MAD = \frac{\sum_{i=1}^m |\hat{y}_i - y_i|}{m} \quad (2.15)$$

3. *Mean absolute percentage error (MAPE).*

Rumus yang digunakan untuk menghitung nilai MAPE dapat dilihat pada Persamaan 2.16 yang didapatkan dari nilai APE yang ditunjukkan pada Persamaan 2.17.

$$MAPE = \frac{\sum_{i=1}^m APE_i}{m} \quad (2.16)$$

Dimana

$$APE = \frac{\sum_{i=1}^m |\hat{y}_i - y_i|}{m} \times 100 \quad (2.17)$$

Dengan melihat salah satu atau lebih ukuran error di atas, dapat dipilih metode mana yang terbaik untuk suatu set data. MAPE sering digunakan karena MAPE tetap mempertahankan konvergen selama evolusi generasi (Wu et al., 2009). Untuk rentang nilai MAPE adalah $0 \leq MAPE \leq +\infty$ (Makridakis & Hibon, 1995).

2.9 Hybrid Algoritma Genetika dan *Support Vector Regression*

Pemodelan HGA-SVR yang akan digunakan yakni tipe dari fungsi kernel dan nilai parameter SVR akan dioptimasi secara dinamis dengan mengimplementasikan proses berevolusi, dan model SVR akan berperan sebagai peramal dengan nilai-nilai optimal yang telah dihasilkan (Wu et al., 2009). Proses *hybrid* algoritma genetika yang diimplementasikan pada penelitian ini berbeda dengan pengertian *hybrid* algoritma genetika yang sudah dijelaskan pada subbab 2.5. Proses *hybrid* pada penelitian kali ini dilakukan pada representasi kromosom yang akan dijelaskan pada subbab 2.9.1.

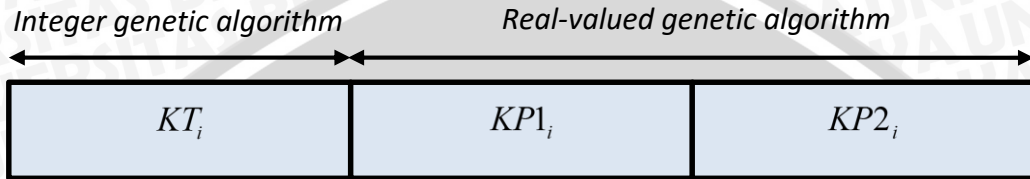
2.9.1 Representasi Kromosom

Representasi kromosom pada *hybrid* algoritma genetika bisa dibilang mudah, karena semua parameter yang ada dan tipe dari kernel yang akan dioptimasi bisa langsung direpresentasikan menjadi bentuk kromosom. Sehingga kromosom

tersebut dapat direpresentasikan seperti pada himpunan Persamaan 2.18 (Wu et al., 2009) :

$$X = \{KT_i, KP1_i, KP2_i\} \quad (2.18)$$

dimana $KP1_i$ dan $KP2_i$ merupakan nilai dari parameter fungsi kernel sedangkan KT_i merupakan kode dari tipe fungsi kernel. Representasi kromosom akan ditunjukkan pada Gambar 2.2.



Gambar 2.2 Struktur Gen Pada HGA

2.9.2 Operator Genetika

Pada pemodelan *hybrid* algoritma genetika dan *support vector regression* yang diimplementasikan menggunakan metode seleksi *elitism*. Pada metode seleksi *elitism* dapat menghasilkan kromosom yang bisa digunakan untuk generasi selanjutnya ditempatkan di proses perkawinan pada proses *crossover* dan mutasi. Setelah kromosom pasangan dipilih, satu atau lebih posisi kromosom akan dipilih secara acak untuk diproses lagi. Kromosom yang baru akan dikombinasikan dengan kromosom yang sudah terbentuk sebelumnya untuk menjadi generasi yang baru. Metode *crossover* yang digunakan adalah *adewuya's crossover* yang disajikan pada Persamaan 2.19, 2.20, dan 2.21 (Wu, Tzeng, Goo, & Fang, 2007).

$$X_1^{old} = \{x_{11}, x_{12}, \dots, x_{1n}\}, X_2^{old} = \{x_{21}, x_{22}, \dots, x_{2n}\} \quad (2.19)$$

Perubahan jarak dekat :

$$\begin{aligned} X_1^{new} &= X_1^{old} + \sigma(X_1^{old} - X_2^{old}) \\ X_2^{new} &= X_2^{old} - \sigma(X_1^{old} - X_2^{old}) \end{aligned} \quad (2.20)$$

Perubahan jarak jauh :

$$\begin{aligned} X_1^{new} &= X_1^{old} + \sigma(X_2^{old} - X_1^{old}) \\ X_2^{new} &= X_2^{old} - \sigma(X_2^{old} - X_1^{old}) \end{aligned} \quad (2.21)$$

X_1^{old} dan X_2^{old} merupakan representasi dari nilai kromosom sebelum proses *crossover*. X_1^{new} dan X_2^{new} merupakan representasi nilai kromosom baru yang

telah mengalami proses *crossover*. Sedangkan σ merepresentasikan bilangan acak dengan batas (0,1).

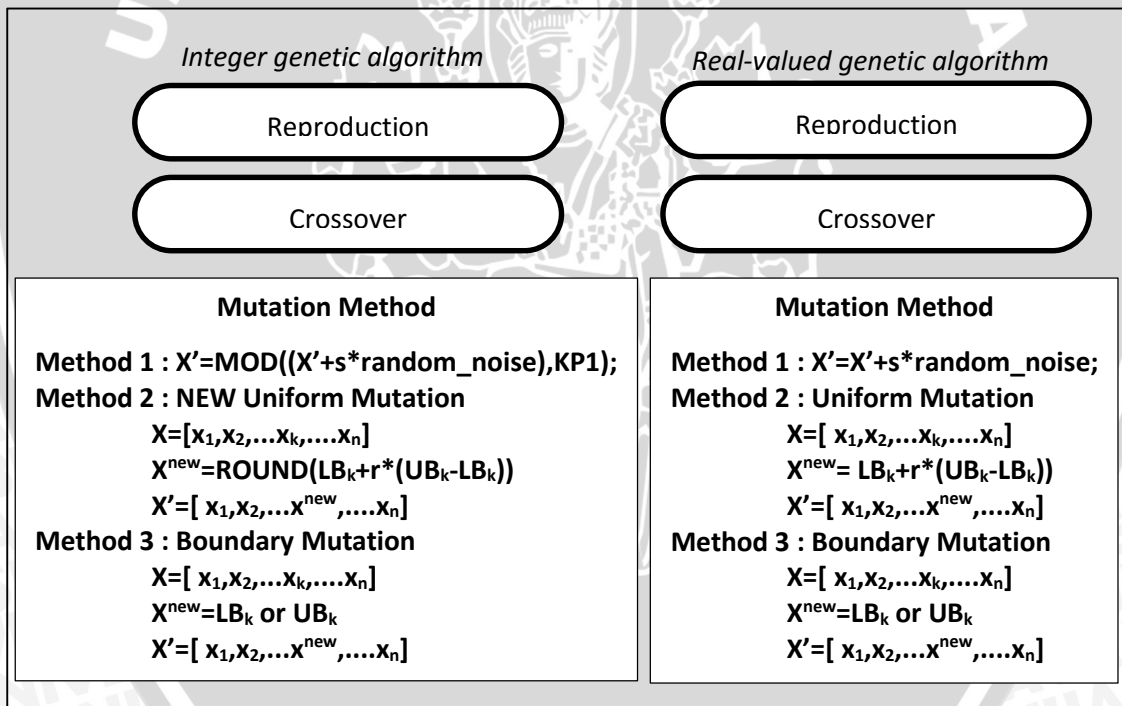
Proses mutasi dalam model ini disajikan pada Persamaan 2.22, 2.23, dan 2.24 (Wu et al., 2009):

$$X^{old} = \{x_1, x_2, \dots, x_n\} \quad (2.22)$$

$$X_k^{new} = LB_k + r \times (UB_k - LB_k) \quad (2.23)$$

$$X^{new} = \{x_1, x_2, x_k^{new}, \dots, x_n\} \quad (2.24)$$

Dimana n merupakan banyak parameter, r merepresentasikan bilangan acak dengan batas (0,1), dan k merupakan lokasi mutasi. *Lower Bound (LB)* dan *Upper Bound (UB)* merupakan parameter batas bawah dan batas atas. Sedangkan untuk LB_k dan UB_k merupakan parameter batas bawah dan batas atas pada lokasi k . X^{old} merepresentasikan individu sebelum proses mutasi dan X^{new} merepresentasikan individu baru setelah proses mutasi. Operator genetika yang baru akan ditunjukkan pada Gambar 2.3.



Gambar 2.3 Operator Genetika HGA

Berikut ini merupakan keterangan dari Gambar 2.3 :

- $KP1$ = Nilai parameter ke-1 kernel.
- r = Nilai random dengan batas (0,1).
- LB_k = Nilai batas bawah parameter.

UB_k = Nilai batas atas parameter.

2.9.3 Fungsi *Fitness*

Kualitas dari masing-masing kromosom yang terbentuk akan dinilai kelayakannya dengan menggunakan fungsi *fitness* sebelum mencari nilai optimal untuk parameter SVR (Mahmudy et al., 2014) (Wu et al., 2009). Beberapa indikator telah digunakan untuk menghitung mengevaluasi keakuratan model misalnya MAPE dan RMSE. Fungsi *fitness* didefinisikan pada Persamaan 2.25 sebagai berikut (J., Neto, & Ferreira, 2009) (Alves, Pereira, Silva, & Ferreira, 2013) :

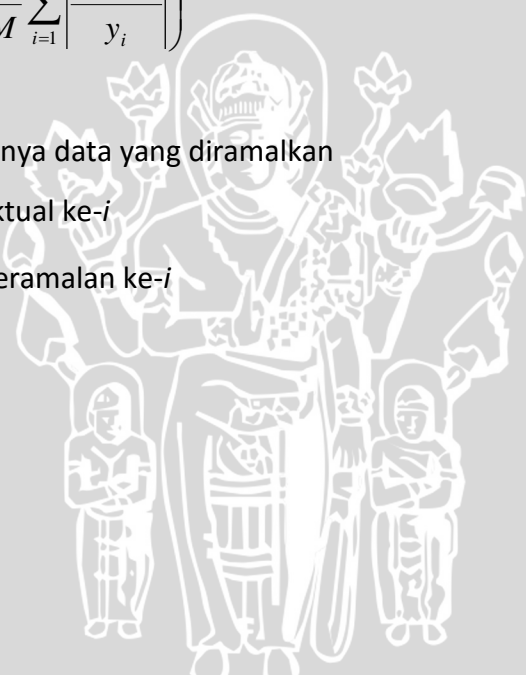
$$fitness = \frac{1}{1 + MAPE} \quad (2.25)$$
$$1 + \left(100 * \frac{1}{M} \sum_{i=1}^M \left| \frac{y_i - \hat{y}_i}{y_i} \right| \right)$$

Keterangan :

M : Banyaknya data yang diramalkan

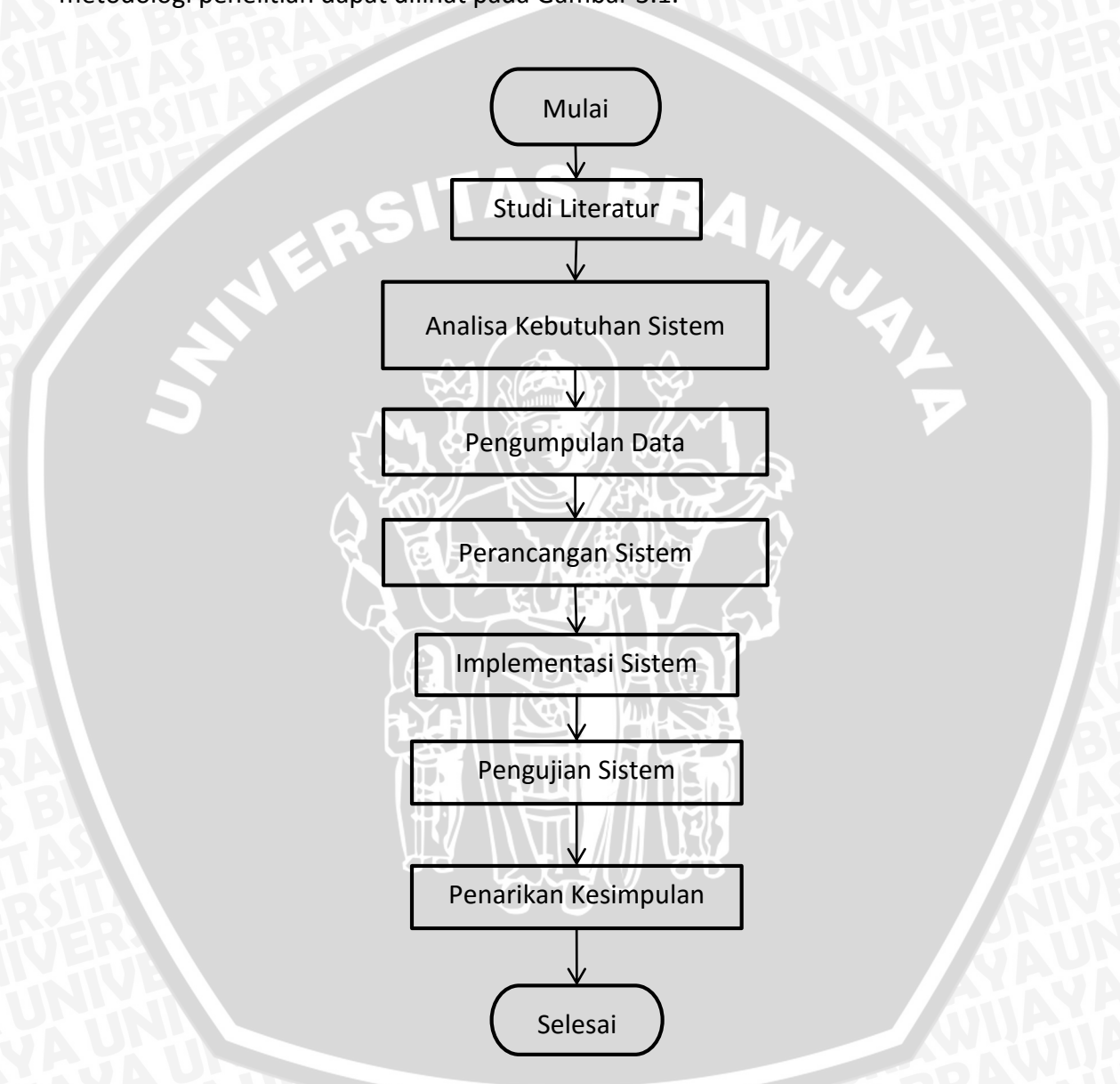
y_i : Data aktual ke- i

\hat{y}_i : Hasil peramalan ke- i



BAB 3 METODOLOGI PENELITIAN

Pada bab ini akan dibahas mengenai beberapa hal, yaitu tahapan penelitian yang akan dilakukan, kebutuhan sistem yang akan dibuat dan penyelesaian masalah menggunakan implementasi metode *Support Vector Regression* yang dioptimasi menggunakan metode *Hybrid* Algoritma Genetika. Tahapan metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi Literatur

Mempelajari literatur dari beberapa bidang ilmu yang berhubungan dengan pembuatan sistem peramalan konsumsi BBM, diantaranya :

- Metode *Support Vector Regression*
- Metode *Hybrid Algoritma Genetika*
- Rekayasa Perangkat Lunak
- Pemrograman dengan menggunakan bahasa Java
- Proses Pengujian Sistem
- Konsumsi BBM.

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

3.2 Analisa Kebutuhan

Analisis kebutuhan bertujuan untuk menganalisis dan mendapatkan semua kebutuhan yang diperlukan Sistem peramalan konsumsi bahan bakar minyak dengan metode *hybrid algoritma genetika* dan *support vector regression*. Analisis kebutuhan disesuaikan dengan variabel penelitian, lokasi penelitian, dan kebutuhan penelitian.

Spesifikasi kebutuhan perangkat yang digunakan dalam pembuatan sistem pendukung keputusan ini meliputi:

- a. Spesifikasi Kebutuhan *Hardware*
 - Komputer / Laptop
- b. Spesifikasi Kebutuhan *Software*
 - Microsoft Windows 7 sebagai operating system.
 - JAVA merupakan bahasa pemrograman untuk menjalankan aplikasi developer
 - Menggunakan Netbeans 8.0 sebagai editor untuk membuat aplikasi
 - Enterprise Architecture sebagai aplikasi untuk membuat diagram
 - MySQL sebagai server DBMS (*Database Management System*).

3.3 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang didapatkan melalui *website* <http://www.bp.com/en/global/corporate/about-bp/energy-economics/statistical-review-of-world-energy.html>. Data tersebut berupa jumlah konsumsi BBM di seluruh dunia pada tahun 1965-2014, namun pada penelitian ini yang digunakan adalah jumlah konsumsi BBM yang ada di Indonesia saja.

3.4 Perancangan Sistem

Perancangan sistem dilakukan untuk mempermudah implementasi, pengujian serta analisis. Sistem akan dibangun menggunakan bahasa pemrograman Java beserta antarmuka yang dapat memudahkan pengguna untuk menggunakan sistem yang akan dibangun.

1. Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna dapat memudahkan pengguna menggunakan sistem yang akan dibangun.

2. Perancangan Pengujian

Perancangan pengujian sistem yang dilakukan meliputi pengujian jumlah iterasi SVR, generasi HGA, populasi, nilai *crossover rate*, nilai *mutation rate*, range parameter $KP1$, $KP2$, λ , γ , dan C .

3.5 Implementasi

Pada tahap ini akan dijelaskan bagaimana implemetasi dalam pembuatan peramalan konsumsi bahan bakar minyak (BBM) di Indonesia dengan metode *hybrid* algoritma genetika dan *support vector regression* dengan mengacu pada perancangan sistem. Impelementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman Java dan *tools* pendukung lainnya.

3.6 Perancangan Uji Coba

Pada tahap ini dilakukan pengujian kerja sistem yang telah dibuat untuk mengetahui bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dan kebutuhan. Pengujian yang dilakukan meliputi pengujian iterasi SVR, generasi HGA, nilai *crossover rate*, nilai *mutation rate*, dan juga *range* parameter yang dioptimasi.

3.7 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah sesuai tahapan perancangan, implementasi, dan pengujian sistem aplikasi peramalan konsumsi bahan bakar minyak (BBM) di Indonesia dengan metode *hybrid* algoritma genetika dan *support vector regression* telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan evaluasi terhadap sistem yang dibangun.

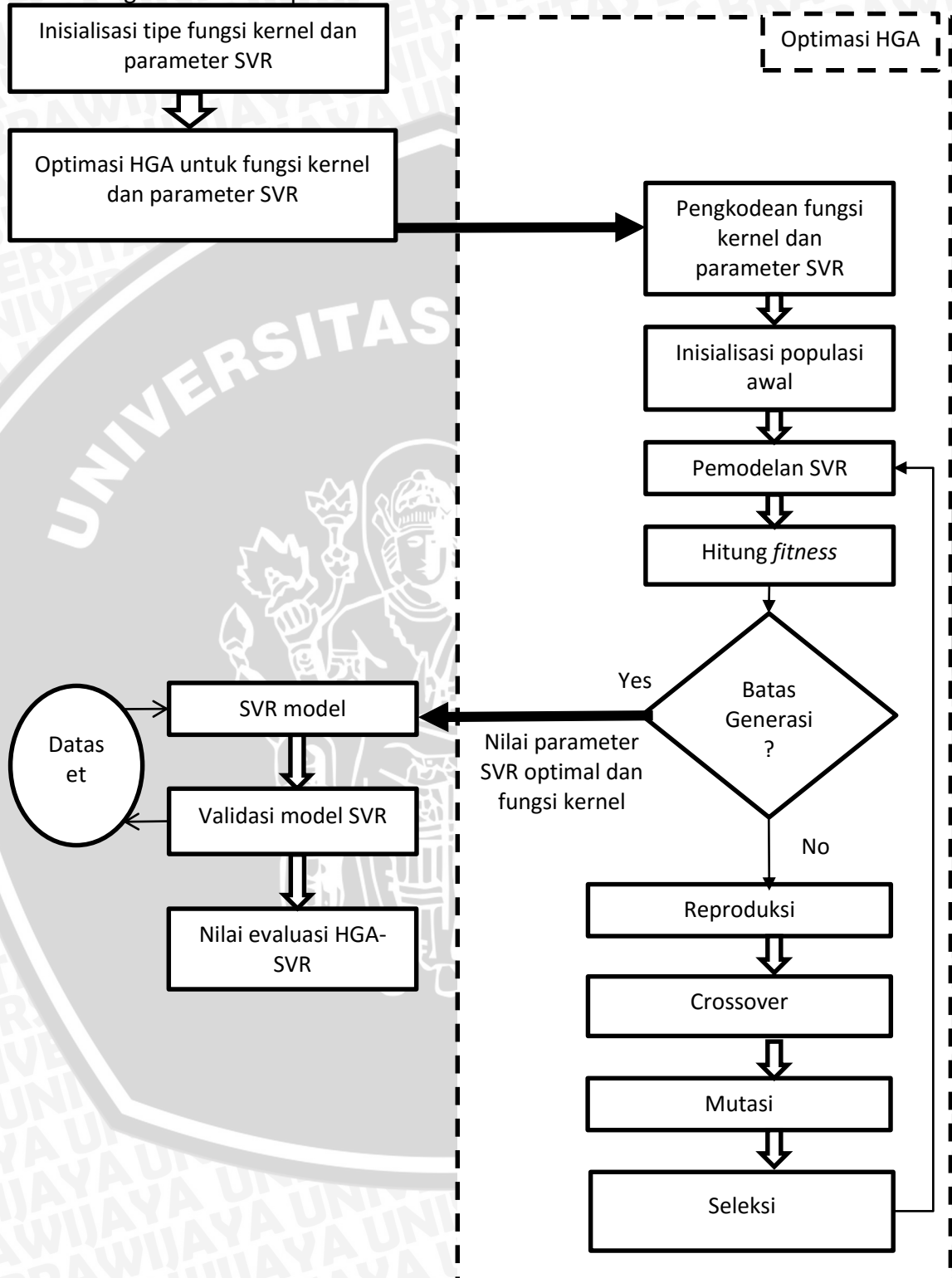
3.8 Penyelesaian Masalah Menggunakan HGA-SVR

Peramalan konsumsi BBM dilakukan dengan membentuk model regresi, yang selanjutnya dilakukan pengujian untuk mendapatkan *error rate*. *Error rate* tersebut mengindikasikan bagaimana akurasi model regresi yang dibentuk atau biasa disebut dengan nilai evaluasi. Tujuan utamanya adalah meminimalkan tingkat *error* dari peramalan yang dilakukan. Pada studi kasus kali ini, digunakan data konsumsi BBM pada tahun 1965-2014 yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 Tabel Data Konsumsi BBM

Tahun	Jumlah Konsumsi BBM	Tahun	Jumlah Konsumsi BBM
1965	122.4476712	2001	1152.913425
1966	117.9256986	2002	1204.439038
1967	114.7659178	2003	1222.237686
1968	121.1098361	2004	1299.186962
1969	131.1056438	2005	1284.890503
1970	138.2172603	2006	1247.184651
1971	142.9839452	2007	1298.935538
1972	154.8656831	2008	1293.616438
1973	185.0575342	2009	1333.74248
1974	193.982411	2010	1457.756157
1975	221.9059726	2011	1566.807712
1976	238.0966667	2012	1599.388411
1977	287.407589	2013	1614.96582
1978	326.0261096	2014	1640.990397
1979	354.9598082		
1980	396.4958197		
1981	441.5163836		
1982	456.6725479		
1983	448.4044932		
1984	477.3640164		
1985	467.9943014		
1986	494.1211233		
1987	511.9309041		
1988	537.0831967		
1989	570.7151781		
1990	653.2235068		
1991	692.1147945		
1992	745.0984153		
1993	785.8611233		
1994	808.9757534		
1995	864.6745753		
1996	924.8253279		
1997	1026.491863		
1998	980.7497808		
1999	1024.682849		
2000	1136.86376		

Berdasarkan data konsumsi BBM diatas, dapat ditentukan data latih yang akan digunakan untuk pelatihan SVR sesuai dengan langkah-langkah yang terdapat di dalam blok diagram HGA-SVR pada Gambar 3.2 berikut ini :



Gambar 3.2 Blok Diagram Proses HGA-SVR

Penjelasan :

1. Inisialisasi fungsi kernel, parameter fungsi kernel, dan parameter fungsi kernel yang akan dioptimasi.
2. Inisialisasi jumlah generasi, popsize, *crossover rate*, *mutation rate*, serta range nilai dari masing-masing parameter.
3. Tentukan nilai individu dari masing-masing parameter yang dioptimasi sebanyak popsize yang telah ditentukan.
4. Proses reproduksi ada 2 yakni *crossover* dengan menggunakan metode *adewuya's crossover* dan mutasi dengan menggunakan metode *new uniform mutation*.
5. Setelah proses reproduksi selesai, maka dilakukan proses hitung *fitness* dengan pemodelan SVR untuk melakukan proses selanjutnya.
6. Metode *elitism selection* digunakan untuk mencari nilai *fitness* tertinggi.
7. Individu yang memiliki *fitness* tertinggi sebanyak popsize akan menjadi parent populasi yang selanjutnya untuk generasi berikutnya.
 - o Subproses SVR :
 - Inisialisasi data aktual untuk peramalan data selanjutnya.
 - Perhitungan matriks kernel dilakukan sesuai dengan parameter yang telah diinisialisasi pada proses HGA
 - Perhitungan matriks R_{ij} .
 - Perhitungan nilai *error*.
 - Perhitungan $\delta\alpha_i$ dan $\delta\alpha_i^*$.
 - Update α_i dan α_i^* .
 - Pengujian model untuk mendapatkan nilai peramalan.

3.8.2 Penyelesaian dengan SVR

Peramalan konsumsi BBM dilakukan untuk membentuk model regresi, yang kemudian akan diuji untuk mendapatkan *error rate*. *Error rate* tersebut mengindikasikan bagaimana akurasi model regresi yang dibentuk atau biasa disebut dengan nilai evaluasi. Tujuan utamanya adalah meminimalkan tingkat *error* dari peramalan yang dilakukan.

1. Memilih Fitur yang Digunakan
Data konsumsi yang digunakan adalah mulai dari tahun 1975-1986. Untuk menentukan jumlah konsumsi BBM yang terjadi pada bulan berikutnya akan diformulasikan pada Tabel 3.2 dengan menggunakan data 4 tahun sebelumnya.

Tabel 3.2 Tabel Jumlah Kosumsi BBM

No	Tahun	X ₁	X ₂	X ₃	X ₄	Y
1	1979	238.0966667	287.407589	326.0261096	354.9598082	396.4958197

2	1980	287.407589	326.0261096	354.9598082	396.4958197	441.5163836
3	1981	326.0261096	354.9598082	396.4958197	441.5163836	456.6725479
4	1982	354.9598082	396.4958197	441.5163836	456.6725479	448.4044932
5	1983	396.4958197	441.5163836	456.6725479	448.4044932	477.3640164
6	1984	441.5163836	456.6725479	448.4044932	477.3640164	467.9943014
7	1985	456.6725479	448.4044932	477.3640164	467.9943014	494.1211233
8	1986	448.4044932	477.3640164	467.9943014	494.1211233	511.9309041
9	1987	477.3640164	467.9943014	494.1211233	511.9309041	537.0831967
10	1988	467.9943014	494.1211233	511.9309041	537.0831967	570.7151781

2. Menentukan Data Latih dan Data Uji

Dalam proses SVR, harus ditentukan data yang akan digunakan sebagai data latih dan data uji. Untuk studi kasus ini, digunakan data latih dan data uji yang sama sebanyak 5 data sesuai dengan Tabel 3.2 yang akan disajikan pada Tabel 3.3 dan Tabel 3.4 berikut:

Tabel 3.3 Tabel Data Latih

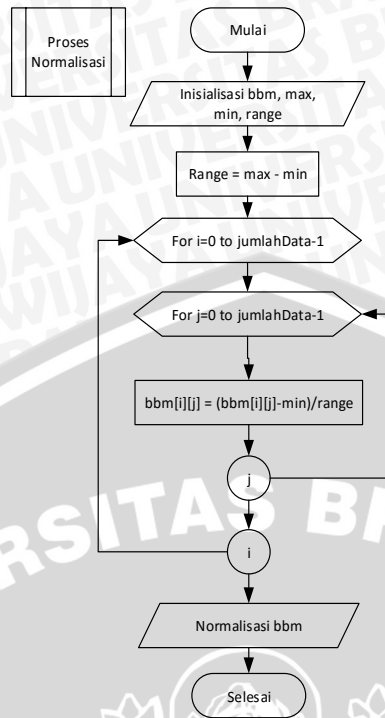
No	Tahun	X ₁	X ₂	X ₃	X ₄	Y
1	1979	238.0966667	287.407589	326.0261096	354.9598082	396.4958197
2	1980	287.407589	326.0261096	354.9598082	396.4958197	441.5163836
3	1981	326.0261096	354.9598082	396.4958197	441.5163836	456.6725479
4	1982	354.9598082	396.4958197	441.5163836	456.6725479	448.4044932
5	1983	396.4958197	441.5163836	456.6725479	448.4044932	477.3640164

Tabel 3.4 Tabel Data Uji

No	Tahun	X ₁	X ₂	X ₃	X ₄	Y
1	1984	441.5163836	456.6725479	448.4044932	477.3640164	467.9943014
2	1985	456.6725479	448.4044932	477.3640164	467.9943014	494.1211233
3	1986	448.4044932	477.3640164	467.9943014	494.1211233	511.9309041
4	1987	477.3640164	467.9943014	494.1211233	511.9309041	537.0831967
5	1988	467.9943014	494.1211233	511.9309041	537.0831967	570.7151781

3. Melakukan normalisasi terhadap data latih dan data uji

Proses normalisasi merupakan proses yang dilakukan untuk melakukan standarisasi terhadap semua data. Proses normalisasi yang digunakan pada penelitian ini adalah *Min-Max normalization*. Diagram alir proses normalisasi akan dijelaskan pada Gambar 3.3 Diagram Alir Proses Normalisasi Data .



Gambar 3.3 Diagram Alir Proses Normalisasi Data BBM

Pada perhitungan normalisasi, nilai *range* didapat dari nilai maksimum *dataset* dikurangi nilai minimum *dataset* pada setiap fiturnya. Berikut ini adalah contoh perhitungan hasil normalisasi nilai data latih pertama untuk X1.

- X = 238.0966667
- X_{min} = 114.7659178
- X_{max} = 1640.990397

Setelah diketahui nilai maksimum dan minimum dari data BBM tersebut maka selanjutnya dilakukan perhitungan menggunakan Persamaan 2.12.

$$x' = \frac{(238.0966667 - 114.7659178)}{1640.990397 - 114.7659178} = 0.080808$$

Sehingga hasil normalisasi untuk data latih akan disajikan pada Tabel 3.5. Sedangkan hasil normalisasi untuk data uji akan disajikan pada Tabel 3.6.

Tabel 3.5 Hasil Normalisasi Data Latih

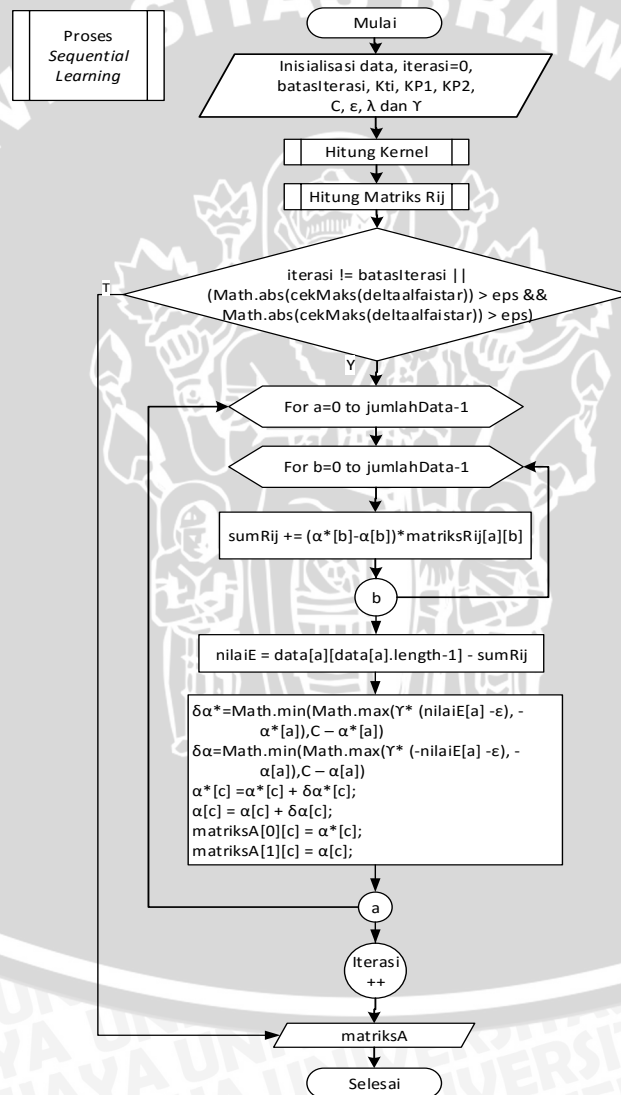
No	Tahun	X1	X2	X3	X4	Y
1	1979	0.080808	0.113117	0.13842	0.157378	0.184593
2	1980	0.113117	0.13842	0.157378	0.184593	0.214091
3	1981	0.13842	0.157378	0.184593	0.214091	0.224021
4	1982	0.157378	0.184593	0.214091	0.224021	0.218604
5	1983	0.184593	0.214091	0.224021	0.218604	0.237578

Tabel 3.6 Hasil Normalisasi Data Uji

No	Tahun	X1	X2	X3	X4	Y
1	1984	0.214091	0.224021	0.218604	0.237578	0.231439
2	1985	0.224021	0.218604	0.237578	0.231439	0.248558
3	1986	0.218604	0.237578	0.231439	0.248558	0.260227
4	1987	0.237578	0.231439	0.248558	0.260227	0.276707
5	1988	0.231439	0.248558	0.260227	0.276707	0.298743

4. Proses *Sequential Learning*

Proses *Sequential learning* merupakan proses yang ada untuk melakukan regresi dari harga BBM. Adapun diagram alir dari proses *sequential learning* dijabarkan seperti pada Gambar 3.4 berikut :



Gambar 3.4 Diagram Alir *Sequential Learning SVR*



➤ Langkah Pertama

Inisialisasi nilai parameter SVR. Berikut beberapa parameter yang digunakan dalam perhitungan metode SVR, yang akan disajikan pada Tabel 3.7 dibawah ini :

Tabel 3.7 Tabel Nilai Parameter SVR

KT_i	$KP1_i$	$KP2_i$	λ	γ	C
2	0.728245716	0.028925063	0.195320536	0.80608222	0.151884162

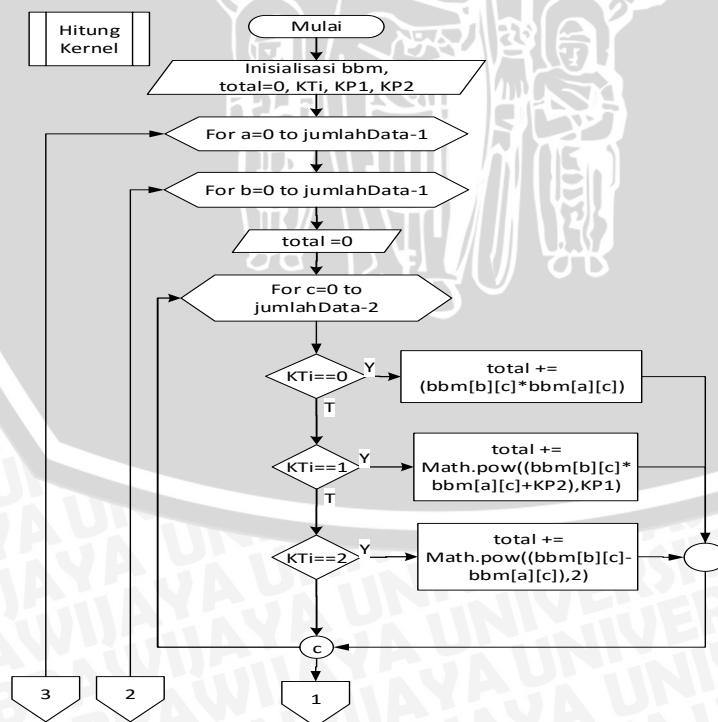
KT_i merupakan tipe kernel ke- i , $KP1_i$ merupakan parameter 1 kernel ke- i , $KP2_i$ merupakan parameter 2 kernel ke- i , λ merupakan nilai lambda, γ merupakan nilai gamma, dan C merupakan nilai kompleksitas,

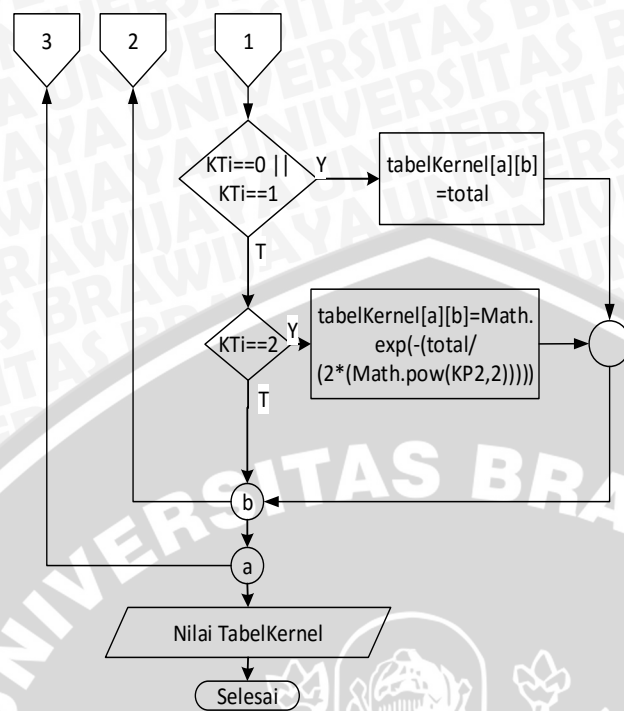
➤ Langkah Kedua

Inisialisasi nilai α_i dan α_i^* dengan nilai 0 sebanyak jumlah data latih.

➤ Langkah Ketiga

Membangun model regresi, berikut ini contoh perhitungan data latih ke-1 dengan data latih ke-5. Pertama, menghitung nilai matriks kernel menggunakan tipe kernel yang didasarkan pada inisialisasi parameter yang dihasilkan pada optimasi HGA. Tipe kernel yang digunakan adalah linear kernel, polinomial kernel, dan RBF kernel. Berikut ini merupakan tahapan proses perhitungan matriks kernel yang ditunjukkan pada Gambar 3.5 Diagram Alir Proses Hitung Matriks Kernel menggunakan RBF kernel berdasarkan hasil optimasi HGA.





Gambar 3.5 Diagram Alir Proses Hitung Matriks Kernel

Berikut ini contoh perhitungan data latih ke-1 dan data latih ke-2. Pertama, menghitung jarak antar data. Pada Tabel 3.8 berisi hasil perhitungan jarak data latih 1 sampai 5 :

$$\| x_1 - x_2 \|^2 = (0.113116828 - 0.080807739)^2 + (0.138420131 - 0.113116828)^2 + (0.157377826 - 0.138420131)^2 + (0.184592703 - 0.157377826)^2$$

Tabel 3.8 Hasil Perhitungan Jarak Data Latih

Data ke-i	1	2	3	4	5
1	0	0.002784178	0.01062648	0.021139152	0.032043212
2	0.002784178	0	0.002610433	0.008861898	0.016432934
3	0.01062648	0.002610433	0	0.00206879	0.006923231
4	0.021139152	0.008861898	0.00206879	0	0.001738743
5	0.032043212	0.016432934	0.006923231	0.001738743	0

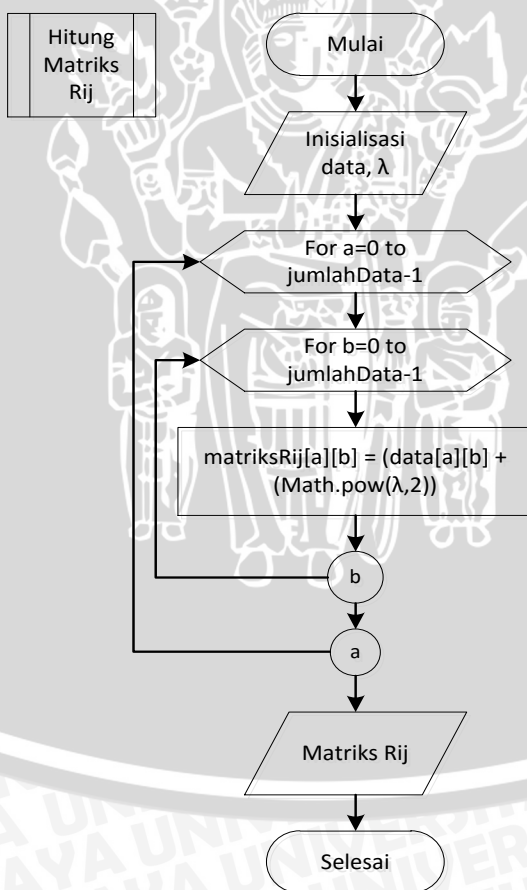
Setelah melakukan proses menghitung jarak data latih maka dilakukan proses perhitungan matriks kernel. Tabel 3.9 Berisi matriks kernel data latih 1-5 :

Tabel 3.9 Hasil Komputasi Matriks Kernel

Data ke- <i>i</i>	1	2	3	4	5
1	1	0.18940528	0.0017458	3.26232E-06	4.82479E-09
2	0.189405285	1	0.210129067	0.00501159	5.43229E-05
3	0.0017458	0.21012906	1	0.290445369	0.015963895
4	3.26232E-06	0.00501159	0.290445369	1	0.353773319
5	4.82479E-09	5.43229E-05	0.015963895	0.353773319	1

➤ Langkah Keempat

Membentuk model regresi berdasarkan jarak data yang telah diperoleh sebelumnya. Berikut ini contoh perhitungan menggunakan data latih ke-1 dan data latih ke-5. Tabel 3.10 berisi hasil model regresi yang telah dibentuk, fungsi kernel yang digunakan merupakan fungsi kernel RBF. Tahapan proses perhitungan matriks R_{ij} dapat dilihat pada Gambar 3.6 Diagram Alir Matriks R_{ij} .



Gambar 3.6 Diagram Alir Matriks R_{ij}

Berikut ini merupakan contoh perhitungan matriks Rij data latih 1 dan data latih 2 berdasarkan Persamaan 2.2.

$$[R]_{1,2} = K(x_1, x_2) + \lambda^2 = \exp\left(-\frac{0}{2 \times (0.028925063)^2}\right) + 0.195320536^2$$

$$= 1.03815011$$

Tabel 3.10 Hasil Komputasi Matriks Rij

Rij	1	2	3	4	5
1	1.038150112	0.227555397	0.039895912	0.038153374	0.038150116
2	0.227555397	1.038150112	0.248279178	0.043161702	0.038204434
3	0.039895912	0.248279178	1.038150112	0.32859548	0.054114006
4	0.038153374	0.043161702	0.32859548	1.038150112	0.391923431
5	0.038150116	0.038204434	0.054114006	0.391923431	1.038150112

➤ Langkah Kelima

Langkah selanjutnya adalah melakukan perhitungan E sesuai dengan Persamaan 2.3. Berikut ini merupakan contoh perhitungan nilai E sehingga didapatkan hasil nilai E pada iterasi 10 pelatihan SVR yang akan disajikan pada Tabel 3.11 Nilai Error Iterasi sebagai berikut:

$$E_1 = 0.184592703 - ((0.151884162 - 0) * 1.03815011$$

$$- (0.151884162 - 0) * 0.2275554$$

$$- (0.151884162 - 0) * 0.03989591$$

$$- (0.151884162 - 0) * 0.03815337$$

$$- (0.151884162 - 0) * 0.03815012$$

$$= -0.025296767$$

Tabel 3.11 Nilai Error Iterasi 10

E
-0.025296767
-0.028217824
-0.035554108
-0.060860575
0.000556855

➤ Langkah Keenam

Setelah mendapatkan nilai E, maka pada iterasi tersebut harus dilakukan perhitungan nilai $\delta\alpha_i$ dan $\delta\alpha_i^*$ sesuai dengan Persamaan 2.4 dan



Persamaan 2.5 sehingga akan dihasilkan nilai seperti pada Tabel 3.12 berikut ini :

$$\begin{aligned} \delta\alpha_i^* &= (\min(\max(0.80608222 * (-0.025296767 - 0.0000001), -0.151884162), \\ &0.15188 - 0.151884162)) \\ &= -0.020391355 \end{aligned}$$

$$\begin{aligned} \delta\alpha_i &= (\min(\max(0.80608222 * (0.025296767 - 0.0000001), -0), 0.15188 - 0)) \\ &= 0.020391193 \end{aligned}$$

Tabel 3.12 Nilai $\delta\alpha_i$ dan $\delta\alpha_i^*$ Iterasi 10

$\delta\alpha_i^*$	$\delta\alpha_i$
-0.020391355	0.020391193
-0.022745967	0.022745806
-0.028659615	0.028659454
-0.049058708	0.049058547
0	0

➤ Langkah Ketujuh

Ketika sudah dilakukan perhitungan $\delta\alpha_i^*$ dan $\delta\alpha_i$, langkah selanjutnya adalah melakukan pembaruan nilai α_i^* dan α_i yang nantinya akan digunakan pada iterasi selanjutnya sesuai dengan Persamaan 2.7 dan Persamaan 2.8 sehingga menghasilkan nilai seperti yang ada pada Tabel 3.13.

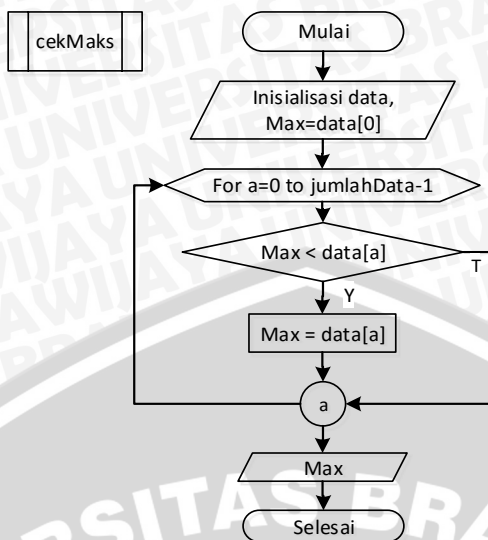
Tabel 3.13 Nilai α_i dan α_i^* Iterasi 10

α_i^*	α_i
0.131492808	0.020391193
0.129138195	0.022745806
0.123224547	0.028659454
0.102825454	0.049058547
0.151884162	0

➤ Langkah Kedelapan

Melakukan pencarian nilai maksimum dari $\delta\alpha_i^*$ dan $\delta\alpha_i$ untuk melakukan pengecekan kondisi berhenti pada proses *sequential learning*. Diagram alir dari proses pencarian maksimum nilai akan dijabarkan pada Gambar 3.7.





Gambar 3.7 Diagram Alir Cek Maksimum

5. Proses Pengujian Model Regresi

Untuk melakukan pengujian Regresi maka dilakukan perhitungan sesuai dengan Persamaan 2.1, sehingga nilai peramalan untuk data 5 akan dijabarkan pada perhitungan di bawah ini. Tabel 3.19 merupakan nilai peramalan pada iterasi ke 10.

$$\begin{aligned}
 f(x) &= ((0.131492808 - 0.020391193) * 1.03815011) + \\
 &((0.129138195 - 0.022745806) * 0.2275554) + \\
 &((0.123224547 - 0.028659454) * 0.03989591) + \\
 &((0.102825454 - 0.049058547) * 0.03815337) + \\
 &((0.151884162 - 0) * 0.03815012) \\
 &= 0.151169
 \end{aligned}$$

Tabel 3.14 Nilai $f(x)$ Iterasi 10

Data Latih ke- <i>i</i>	Nilai Aktual BBM	$f(x)$
1	0.184592703	0.151169
2	0.214090699	0.167335
3	0.224021194	0.154907
4	0.218603869	0.15525
5	0.237578484	0.192172

Dengan langkah yang sama seperti pengujian data latih, pengujian juga dilakukan terhadap data uji dimana sebelum masuk ke proses perhitungan $F(x)$ dilakukan perhitungan jarak data latih dan data uji untuk menghasilkan

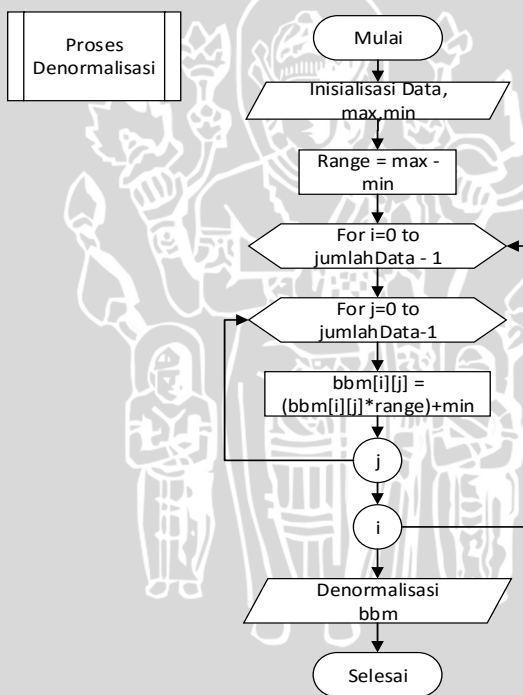
Matriks *Hessian* Data Uji, sehingga hasil peramalan untuk Data Uji dapat dilihat pada Tabel 3.15.

Tabel 3.15 Nilai $f(x)$ Iterasi 10

Data Latih ke- <i>i</i>	Nilai Aktual BBM	$f(x)$
1	0.669690374	0.111208709
2	0.680206301	0.107752701
3	0.713966481	0.096586839
4	0.725628362	0.054849206
5	0.776046421	0.000930311

6. Proses Denormalisasi Hasil Peramalan

Untuk menguji hasil peramalan sebelumnya harus dilakukan denormalisasi terhadap nilai – nilai peramalan yang sudah didapat, proses denormalisasi akan dijelaskan pada Gambar 3.8.



Gambar 3.8 Proses Denormalisasi Data

Maka nilai denormalisasi untuk hasil peramalan data latih pertama akan dijabarkan sesuai dengan Persamaan 2.13.

$$x = (0.15116886 \times (1640.990397 - 114.7659178)) + 114.7659178 = 345.4835382$$

Sehingga nilai hasil normalisasi untuk peramalan data uji dan data latih akan ditunjukkan pada Tabel 3.16 dan Tabel 3.17.

Tabel 3.16 Hasil Denormalisasi Data Latih

Data Latih ke- <i>i</i>	F(x)	F(x) Denormalisasi	Nilai Aktual
1	0.15116886	345.4835382	396.4958197
2	0.16733491	370.1565483	441.5163836
3	0.1549069	351.1886223	456.6725479
4	0.15524972	351.7118461	448.4044932
5	0.19217157	408.0628685	477.3640164

Tabel 3.17 Hasil Denormalisasi Data Uji

Data Uji ke- <i>i</i>	F(x)	F(x) Denormalisasi	Nilai Aktual
1	0.111208709	284.4953712	467.9943014
2	0.107752701	279.2207276	494.1211233
3	0.096586839	262.1791163	511.9309041
4	0.054849206	198.478118	537.0831967
5	0.000930311	116.185781	570.7151781

7. Proses Menghitung *Error Rate*

Tahapan terakhir adalah menghitung tingkat *error* atau biasa disebut *error rate*, dalam skripsi ini digunakan nilai *Mean absolute percentage error* (MAPE sesuai dengan Persamaan 2.14. Berikut ini merupakan contoh perhitungan *value error rate* MAPE dan hasilnya akan ditunjukkan pada Tabel 3.18.

$$\begin{aligned}
 MAPE &= \frac{1}{5} * \left(\left(\text{abs} \left(\frac{396.4958197 - 345.483538}{345.483538} \right) \right) + \left(\text{abs} \left(\frac{441.5163836 - 370.156548}{370.156548} \right) \right) + \left(\text{abs} \left(\frac{456.6725479 - 351.188622}{351.188622} \right) \right) + \left(\text{abs} \left(\frac{448.4044932 - 351.711846}{351.711846} \right) \right) + \left(\text{abs} \left(\frac{477.3640164 - 408.062869}{408.062869} \right) \right) \right) * 100 \\
 &= 17.64155407
 \end{aligned}$$

Tabel 3.18 Nilai error rate MAPE

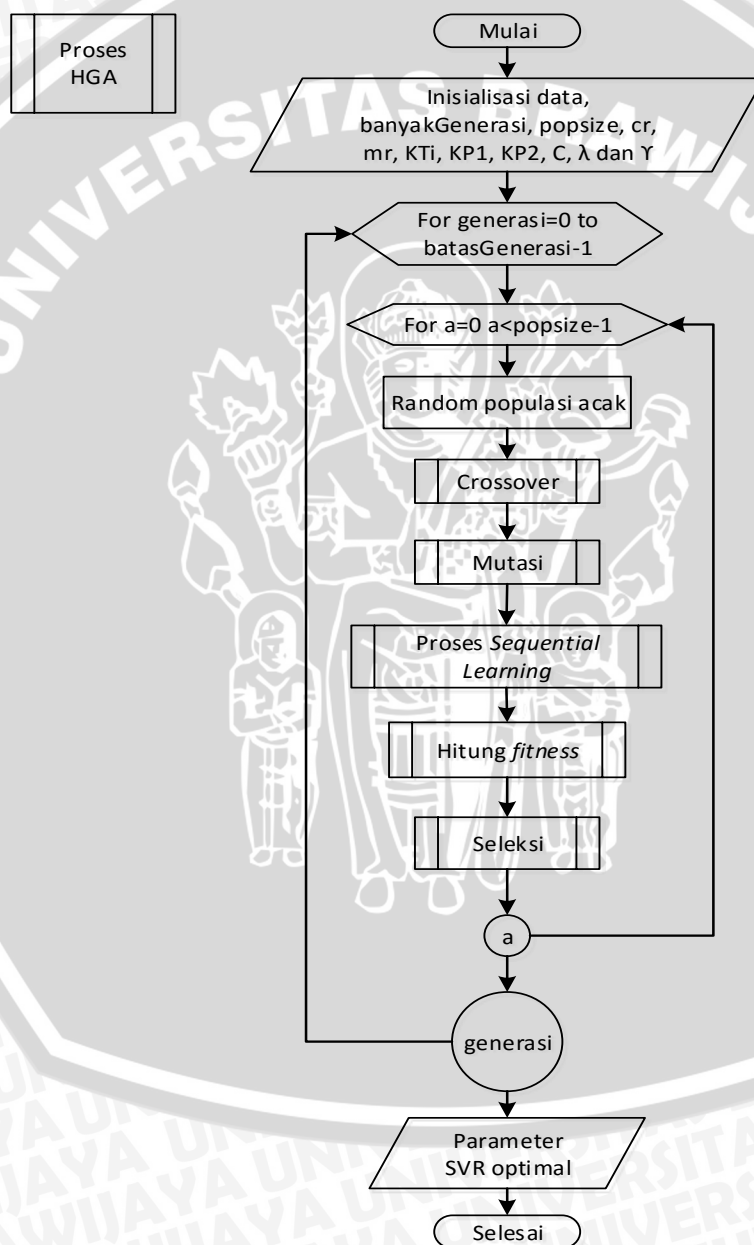
NILAI	(Konsumsi-f(x))	Abs((Konsumsi-f(x))/Konsumsi)
	51.01228147	0.128657804
	71.35983528	0.161624433
	105.4839256	0.230983724
	96.692647	0.215637106



	69.30114789	0.145174637
AVERAGE		0.176415541
MAPE		17.64155407
FITNESS		0.053643596

3.8.3 Penyelesaian dengan *Hybrid* Algoritma Genetika

Pada subbab ini akan dibahas proses pemodelan regresi pada konsumsi BBM dengan menggunakan *Hybrid* Algoritma Genetika yang akan ditunjukkan pada Gambar 3.9.



Gambar 3.9 Proses HGA

Berikut penjabaran *flowchart* yang telah di buat :

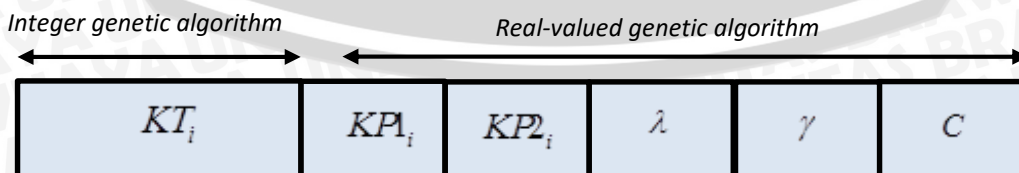
1. Data konsumsi BBM mulai tahun 1965 sampai 2014.
2. Inisialisasi Parameter awal :
 - Jumlah ukuran populasi (*popSize*)
 - Jumlah generasi
 - *Crossover rate* (*cr*)
 - *Mutation rate* (*mr*)
3. Data konsumsi BBM diolah dalam bentuk fungsi SVR
4. *Generate* populasi awal sebanyak jumlah populasi yang telah ditentukan
5. Membuat populasi baru dengan menggunakan proses berikut sebanyak jumlah generasi yang ditentukan :
 - Melakukan proses reproduksi dengan proses crossover, metode yang digunakan adalah *adewuya's crossover*.
 - Proses mutasi dengan metode *new uniform mutation* berdasarkan *mr* yang telah ditentukan pula sebelumnya.
 - Masing-masing kromosom dihitung nilai *fitnessnya*
 - Proses evaluasi dari individu awal (*parent*) dan *offspring*
 - Melakukan proses seleksi dengan metode *elitism selection* untuk menentukan individu sebanyak ukuran populasi awal pada generasi selanjutnya.
6. Hasil akhir adalah kromosom terbaik pada akhir generasi, serta menghasilkan regresi yang baru.

Pada setiap proses yang dideskripsikan pada *flowchart* sebelumnya terdapat *predefined process* dimana proses tersebut akan dijabarkan lebih detail dan disertai dengan perhitungan manualnya. Berikut insialisasi parameter awalnya :

- Jumlah Generasi = 1
- Ukuran Populasi (*popSize*) = 5
- *Crossover rate* (*cr*) = 0,4
- *Mutation rate* (*mr*) = 0,4

3.8.3.1 Representasi Kromosom dan Perhitungan *Fitness*

Kromosom tersusun atas bilangan real yang menyatakan koefisien regresi nilai konsumsi BBM pada tahun sebelumnya. Panjang kromosom dalam 1 individu adalah 6 karena pada metode yang dilakukan fungsi kernel dan parameter yang ada pada fungsi kernel akan di optimasi. Representasi kromosom yang baru akan ditunjukkan pada Gambar 3.10.



Gambar 3.10 Representasi Kromosom HGA

Dimana KT_i merupakan fungsi kernel yang digunakan, $KP1_i$ merupakan parameter 1 untuk fungsi kernel ke- i , $KP2_i$ merupakan parameter 2 untuk fungsi kernel ke- i , λ merupakan variabel skalar pada parameter SVR, γ merupakan gamma pada parameter SVR sedangkan C merupakan nilai kompleksitas pada parameter SVR. Untuk detail inialisasi parameter pada representasi kromosom yang baru akan ditunjukkan pada Tabel 3.19.

Tabel 3.19 Tabel Representasi Kromosom

KT_i	$KP1_i$	$KP2_i$	λ	γ	C
0	0.330353708	0.160166511	0.334964789	0.719712262	0.728245713

3.8.3.2 Inialisasi Populasi Awal

Inialisasi populasi awal merupakan pembentukan kromosom-kromosom sebanyak jumlah populasi yang telah ditentukan pada awal pembuatan parameter. Ukuran populasi atau sering disebut dengan *popSize* pada permasalahan kali ini dibuat sebanyak 5 populasi. Representasi kromosom awal akan ditunjukkan pada Tabel 3.20.

Tabel 3.20 Data Populasi Awal

PARENT	KROMOSOM					
	KT_i	$KP1_i$	$KP2_i$	λ	γ	C
P1	0	0.330354	0.160167	0.334965	0.719712	0.728246
P2	0	0.850688	0.452119	0.680675	0.905636	0.028925
P3	1	0.579092	0.856376	0.640981	0.114154	0.195321
P4	1	0.157052	0.631789	0.759962	0.222317	0.806082
P5	2	0.728246	0.028925	0.195321	0.806082	0.151884

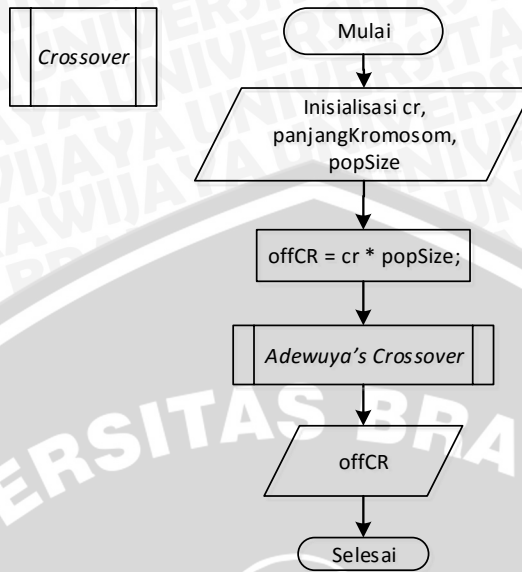
3.8.3.3 Proses Reproduksi

Proses reproduksi merupakan proses individu untuk menghasilkan keturunan baru, pada proses reproduksi dibagi lagi menjadi dua metode yaitu kawin silang atau *crossover* dan mutasi, kedua metode tersebut akan menghasilkan anak atau *offspring*.

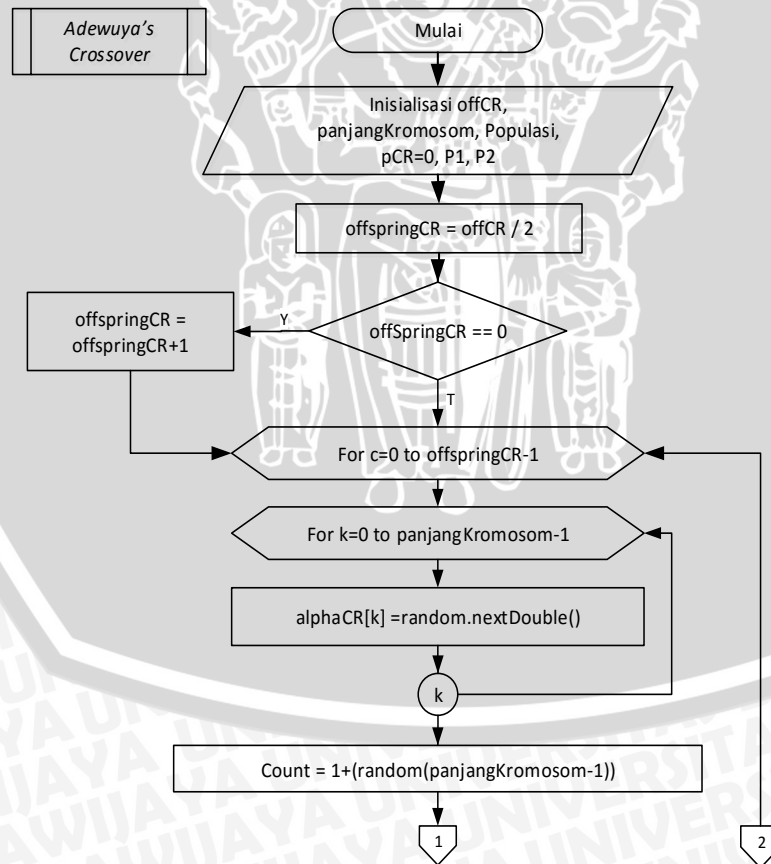
a. Pindah Silang (Crossover)

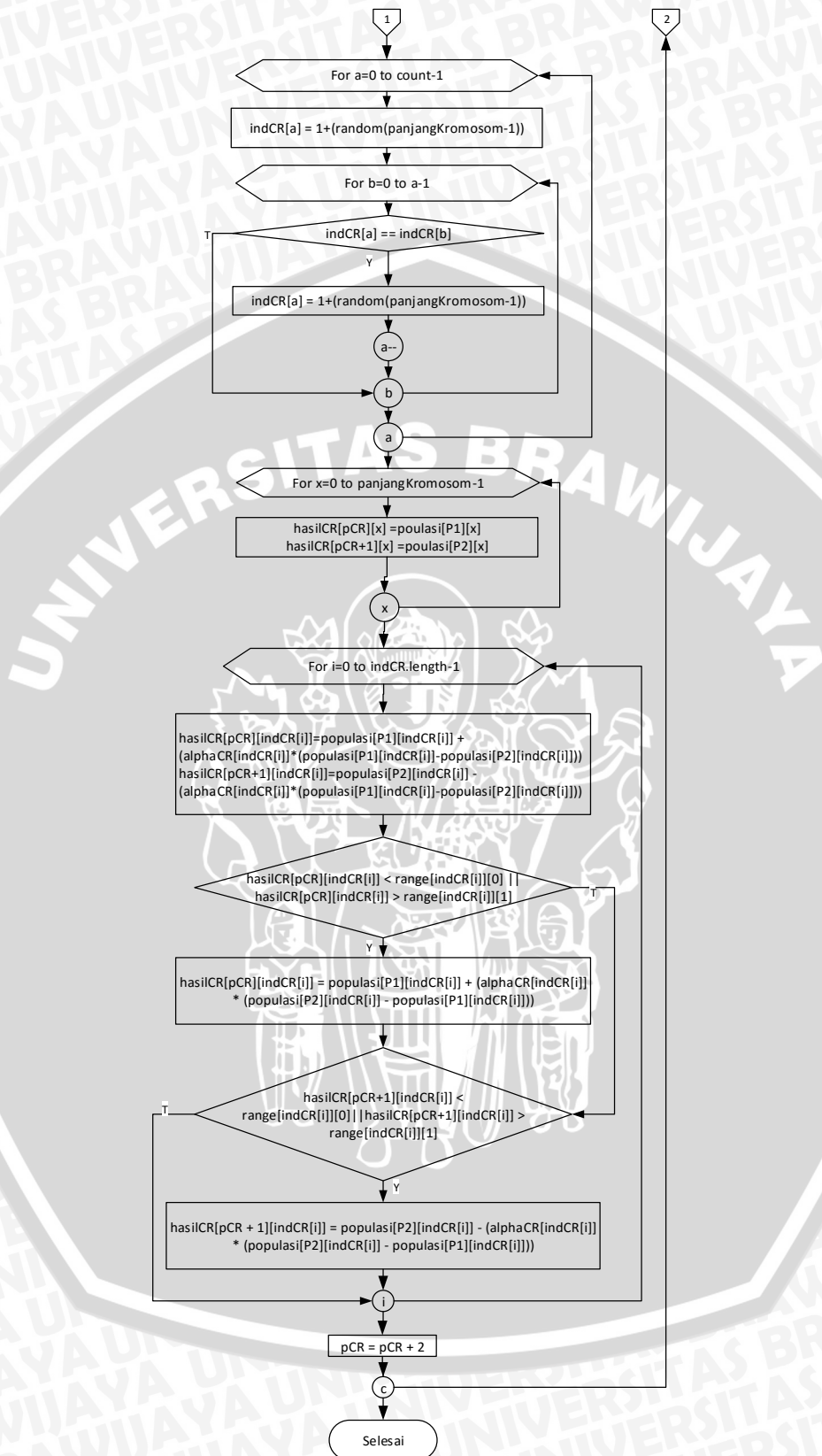
Pada kasus ini metode *crossover* yang dipilih adalah metode *adewuya's crossover*, yaitu dengan memilih dua induk secara *random* dari populasi yang telah dibentuk. Populasi awal sebanyak 5 individu dengan *crossover rate* (*cr*) sebesar 0,4 maka *offspring* yang dihasilkan adalah $0,4 \times 5 = 2$. Akan dijelaskan proses *crossover* dalam diagram alir *flowchart* pada Gambar 3.3. Pada *flowchart* yang ditunjukkan dalam Gambar 3.11 terdapat subproses yaitu subproses *crossover*. Pada diagram alir

Gambar 3.12 akan dijelaskan mengenai metode *adewuya's crossover* yang digunakan dalam penelitian ini.



Gambar 3.11 Flowchart Proses Crossover





Gambar 3.12 Flowchart Proses Adewuya's Crossover



Proses *adewuya's crossover* akan dijabarkan pada perhitungan manual sebagai berikut :

1. σ dibangkitkan secara *random* pada interval [0,1] yang ditunjukkan pada Tabel 3.21.

Tabel 3.21 Tabel Pembangkitan σ

PEMBANGKIT NILAI σ					
σ_1	σ_2	σ_3	σ_4	σ_5	σ_6
0.65001751	0.915117956	0.818165793	0.73539895	0.888114403	0.979119621

2. Tentukan 2 individu secara acak untuk menjadi parent yang akan menghasilkan anak *crossover*.
3. Tentukan banyaknya kromosom yang akan diproses. Maksimal banyaknya kromosom yang akan diproses adalah 5 yakni kromosom ke-2 sampai kromosom ke-6.
4. Hitung masing-masing kromosom yang terpilih menggunakan Persamaan 2.20. Berikut ini merupakan contoh perhitungannya:

$$X_1^{new} = 0.330353708 + 0.915117956(0.330353708 - 0.850687581) = -0.14581316$$

$$X_2^{new} = 0.850687581 - 0.915117956(0.330353708 - 0.850687581) = 1.32685445$$

5. Jika nilai kromosom yang baru kurang dari parameter batas bawah dan lebih dari batas atas maka kromosom tersebut akan diproses kembali sesuai dengan Persamaan 2.21. Berikut ini merupakan contoh perhitungan manualnya :

$$X_1^{new} = 0.330353708 + 0.915117956(0.850687581 - 0.330353708) = 0.806520578$$

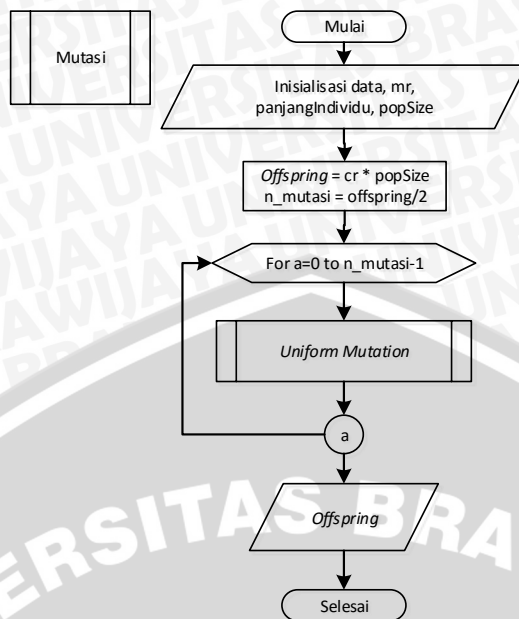
6. Hasil individu yang didapatkan dari proses *crossover* akan ditunjukkan pada Tabel 3.22.

Tabel 3.22 Hasil *Crossover*

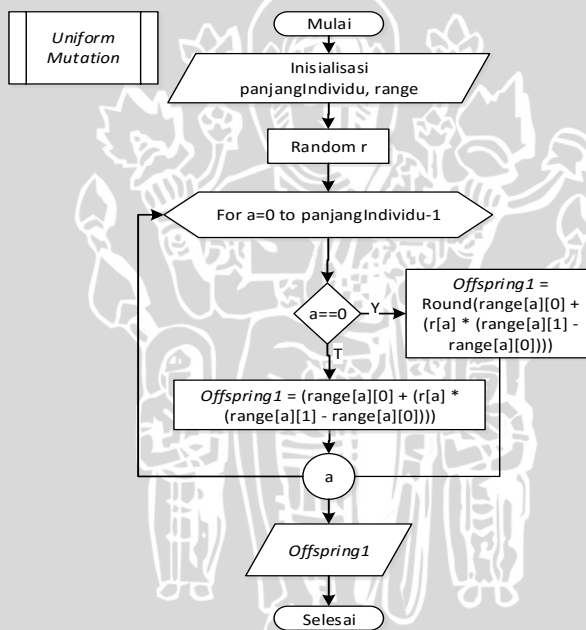
Hasil <i>Crossover</i>						
P1	0	0.330353708	0.160166511	0.334964789	0.719712262	0.728245713
P2	0	0.850687581	0.452119291	0.680674764	0.905635642	0.028925063
C1	0	0.806520578	0.399032289	0.080730037	0.55459103	1.412964282
C2	0	1.32685445	0.69098507	0.934909516	1.070756874	0.713643632

b. Mutasi

Metode mutasi yang digunakan adalah *new uniform mutation*. Dengan *mutation rate* sebesar 0,4 maka banyaknya *offspring* yang dihasilkan adalah $0,4 \times 5 = 2$. Berikut flowchart untuk melakukan proses mutasi akan ditunjukkan pada Gambar 3.13 dan subproses mutasi pada Gambar 3.14.



Gambar 3.13 Diagram Alir Proses Mutasi



Gambar 3.14 Subproses mutasi

r dibangkitkan secara *random* pada interval $[0,1]$ yang ditunjukkan pada Tabel 3.23 Pembangkitan Nilai r .

Tabel 3.23 Pembangkitan Nilai r

PEMBANGKIT NILAI r					
r1	r2	r3	r4	r5	r6
0.0565726	0.221127208	0.661352633	0.839039423	0.712192036	0.149233662

Setelah nilai r dibangkitkan, proses mutasi dilanjutkan dengan menghitung nilai kromosom baru untuk kromosom ke-2 sampai ke-6 menggunakan Persamaan 2.23 sedangkan untuk kromosom ke-1 menggunakan Persamaan *new uniform mutation* yang terdapat pada Gambar 2.3 dan hasilnya akan yang ditunjukkan pada Tabel 3.24. Berikut ini merupakan contoh perhitungan manual untuk parameter ke-2 sampai ke-6 :

$$C3 = 0.01 + 0.839039423 * (100 - (0.1)) = 83.90555193$$

Berikut ini adalah contoh perhitungan manual parameter ke-1.

$$C3 = Round(0 + 0.0565726 * (2 - (0))) = 0$$

Tabel 3.24 Hasil Mutasi

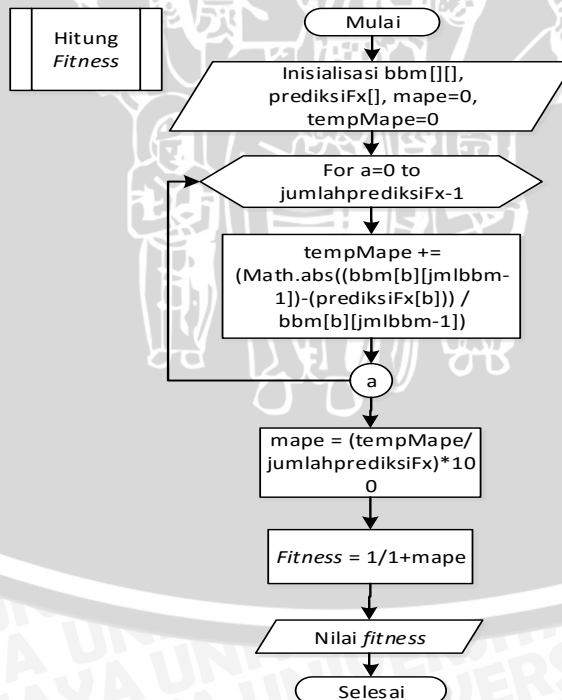
Hasil Mutasi						
C3	0	22.12050949	66.13526329	83.90555193	712.1920362	149.2421695
C4	1	83.3960306	9.767722503	42.48143051	8.954802621	71.00475775

3.8.4 Proses Seleksi

Sebelum melakukan proses seleksi, dilakukan proses perhitungan nilai *fitness* berdasarkan pada Persamaan 2.25.

$$fitness = \frac{1}{1 + 2.294856226} = 0.30350338$$

Hasil perhitungan *fitness* dari pembentukan kromosom sebelumnya dapat dilihat pada Tabel 3.25 dan tahapannya bisa dilihat pada Gambar 3.15.



Gambar 3.15 Diagram Alir Hitung *fitness*

Tabel 3.25 Hasil Perhitungan *Fitness*

KROMOSOM						<i>fitness</i>
KT_i	$KP1_i$	$KP2_i$	λ	γ	C	
0	0.330354	0.160167	0.334965	0.719712	0.728246	0.303503

Proses seleksi ini bertujuan untuk mempertahankan individu yang mempunyai nilai *fitness* yang tinggi agar dapat hidup untuk generasi selanjutnya. Pada penelitian ini metode seleksi yang akan digunakan adalah metode *elitism selection*. Hasil akhir berupa hasil pengurutan mulai dari idividu yang memiliki nilai *fitness* paling besar hingga paling kecil seperti yang ditunjukkan pada Tabel 3.26. Hasil seleksi individu terbaik ditampilkan pada Tabel 3.27.

Tabel 3.26 Hasil Evaluasi

PARENT	KROMOSOM						<i>Fitness</i>
	KT_i	$KP1_i$	$KP2_i$	λ	γ	C	
P1	0	0.3304	0.1602	0.3350	0.7197	0.7282	0.3035
P2	0	0.8507	0.4521	0.6807	0.9056	0.0289	0.0217
P3	1	0.5791	0.8564	0.6410	0.1142	0.1953	0.0007
P4	1	0.1571	0.6318	0.7600	0.2223	0.8061	0.0002
P5	2	0.7282	0.0289	0.1953	0.8061	0.1519	0.0536
C1	0	0.8065	0.3990	0.0807	0.5546	1.4130	0.1199
C2	0	1.3269	0.6910	0.9349	1.0708	0.7136	0.0008
C3	0	22.1205	66.1353	83.9056	712.1920	149.2422	0.0000
C4	1	83.3960	9.7677	42.4814	8.9548	71.0048	0.0000

Tabel 3.27 Hasil Seleksi

PARENT	KROMOSOM						<i>Fitness</i>
	KT_i	$KP1_i$	$KP2_i$	λ	γ	C	
P1	0	0.3304	0.1602	0.3350	0.7197	0.7282	0.3035
C1	0	0.8065	0.3990	0.0807	0.5546	1.4130	0.1199
P5	2	0.7282	0.0289	0.1953	0.8061	0.1519	0.0536
P2	0	0.8507	0.4521	0.6807	0.9056	0.0289	0.0217
C2	0	1.3269	0.6910	0.9349	1.0708	0.7136	0.0008

BAB 4 PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem, yaitu berupa perancangan antarmuka pengguna, perancangan pengujian sistem.

4.1 Perancangan *User Interface*

Pada rancangan *user interface*, aplikasi akan dibuat dua halaman yaitu halaman pertama untuk halaman optimasi, sedangkan halaman kedua digunakan untuk menginput parameter HGA-SVR serta hasil peramalan yang didapat.

4.1.1 Perancangan Halaman Peramalan

Pada rancangan *user interface* untuk halaman peramalan, digunakan untuk memasukkan inputan parameter Algoritma SVR dan HGA, batas – batas parameter yang digunakan, serta perolehan hasil peramalan yang telah diproses. Perancangan halaman parameter algoritma dan hasil peramalan dapat dilihat pada Gambar 4.1.

The screenshot shows a user interface for parameter input. It features a title box (1) at the top center. Below it, there are two sections: 'Parameter' and 'Range Parameter'. The 'Parameter' section includes five text input fields (2-6) for Iterasi SVR, Generasi, PopSize, CR, and MR. The 'Range Parameter' section includes five pairs of text input fields (7-11) for KP1, KP2, Lambda, Gamma, and C. A large empty rectangular area (13) is positioned to the right of these inputs. At the bottom, there are two text input fields (14 and 15) labeled 'MAPE'.

Gambar 4.1 Perancangan Halaman Perancangan

Keterangan:

1. Judul aplikasi.
2. *Textfield* untuk memasukkan jumlah iterasi maksimum SVR.
3. *Textfield* untuk memasukkan banyaknya generasi HGA.
4. *Textfield* untuk memasukkan ukuran banyaknya *PopSize*.

5. *Textfield* untuk memasukkan nilai *Crossover Rate*.
6. *Textfield* untuk memasukkan nilai *Mutation Rate*.
7. *Textfield* untuk memasukkan nilai *range* parameter *KP1*.
8. *Textfield* untuk memasukkan nilai *range* parameter *KP2*.
9. *Textfield* untuk memasukkan nilai *range* parameter λ .
10. *Textfield* untuk memasukkan nilai *range* parameter γ .
11. *Textfield* untuk memasukkan nilai *range* parameter *C*.
12. *Button* proses untuk melakukan proses data menggunakan parameter yang sudah diisikan pada *textfield*.
13. Halaman Hasil proses *Learning* data.
14. *Textpane* untuk menampilkan nilai *error rate*.
15. *Button* proses testing untuk masuk ke halaman testing.

4.1.2 Perancangan Halaman Peramalan

Pada rancangan *user interface* untuk halaman testing, digunakan untuk memasukkan nilai parameter yang optimal serta untuk mendapatkan hasil peramalan. Perancangan halaman parameter algoritma dan hasil peramalan dapat dilihat pada Gambar 4.2.

The image shows a user interface for parameter input and result display. It consists of the following elements:

- 1**: A title box at the top center.
- Parameter**: A section header on the left side.
- 2-9**: Input fields for parameters: Iterasi SVR, KT, KP1, KP2, Lambda, Gamma, C, and Epsilon.
- 11**: A button for processing the data.
- 10**: A large text area for displaying the results.
- MAPE**: A label for the Mean Absolute Percentage Error, with an input field **12** next to it.

Gambar 4.2 Perancangan Halaman Perancangan

Keterangan:

1. Judul aplikasi.

2. *Textfield* untuk memasukkan jumlah iterasi maksimum SVR.
3. *Textfield* untuk memasukkan parameter *KT*.
4. *Textfield* untuk memasukkan nilai parameter *KP1*.
5. *Textfield* untuk memasukkan nilai parameter *KP2*.
6. *Textfield* untuk memasukkan nilai parameter λ .
7. *Textfield* untuk memasukkan nilai parameter γ .
8. *Textfield* untuk memasukkan nilai parameter *C*.
9. *Textfield* nilai default parameter Epsilon.
10. Halaman Hasil proses *Learning* data.
11. *Button* proses untuk melakukan proses data menggunakan parameter yang sudah diisikan pada *textfield*.
12. *Textpane* untuk menampilkan hasil nilai *error rate* yang dihasilkan.

4.2 Perancangan Uji Coba dan Evaluasi

Perancangan uji coba dan evaluasi disebabkan karena tidak adanya metode yang pasti untuk melakukan peramalan konsumsi BBM serta menentukan parameter algoritma HGA-SVR. Oleh karena itu untuk melakukan evaluasi sistem yang akan dibuat maka akan dilakukan uji coba antara lain:

1. Uji coba jumlah iterasi SVR yang optimal.
2. Uji coba jumlah generasi HGA yang optimal.
3. Uji coba banyaknya populasi yang optimal.
4. Uji coba nilai *crossover rate* terbaik agar menghasilkan peramalan yang optimal.
5. Uji coba nilai *mutation rate* terbaik agar menghasilkan peramalan yang optimal.
6. Uji coba range nilai parameter *KP1* pada HGA-SVR.
7. Uji coba range nilai parameter *KP2* pada HGA-SVR.
8. Uji coba range nilai parameter λ pada HGA-SVR.
9. Uji coba range nilai parameter γ pada HGA-SVR.
10. Uji coba range nilai parameter *C* pada HGA-SVR.

4.2.1 Uji Coba Jumlah Iterasi SVR

Uji coba banyaknya jumlah iterasi SVR digunakan untuk mengetahui jumlah iterasi yang paling optimal pada SVR untuk memberikan hasil peramalan yang paling baik. Pengujian iterasi SVR dilakukan sebanyak 10 kali. Rancangan uji coba untuk mengetahui jumlah iterasi SVR yang paling optimal dapat dilihat pada Tabel 4.1.

Tabel 4.1 Rancangan Uji Coba Jumlah Iterasi SVR

Jumlah Iterasi SVR	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
10											

Jumlah Iterasi SVR	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
50											
100											
500											
1000											
5000											
10000											
50000											
100000											

4.2.2 Uji Coba Jumlah Generasi HGA

Uji coba banyaknya jumlah generasi HGA digunakan untuk mengetahui jumlah generasi yang paling optimal pada HGA untuk memberikan hasil peramalan yang paling baik. Pengujian generasi HGA dilakukan sebanyak 10. Rancangan uji coba untuk mengetahui jumlah generasi HGA yang paling optimal dapat dilihat pada Tabel 4.2.

Tabel 4.2 Rancangan Uji Coba Jumlah Generasi HGA

Jumlah generasi	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
50											
65											
80											
95											
110											
125											
140											
155											
170											

4.2.3 Uji Coba Ukuran Populasi

Uji coba ukuran populasi digunakan untuk mencari banyaknya populasi yang bisa menghasilkan nilai parameter optimal dalam peramalan. Rancangan uji coba ukuran populasi ditunjukkan pada Tabel 4.3.

Tabel 4.3 Rancangan Uji Coba Ukuran Populasi

Banyak populasi	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
20											
25											
30											

Banyak populasi	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
35											
40											
45											
50											
55											
60											
65											
70											

4.2.4 Uji Coba *Crossover Rate*

Uji coba *crossover rate* dilakukan untuk mengetahui nilai *cr* terbaik agar mendapatkan hasil koefisien untuk mendapatkan hasil prediksi konsumsi BBM yang optimal. Uji coba kombinasi *cr* ini menggunakan nilai *cr* yang berbeda dalam range 0 – 1. Rancangan uji coba nilai *crossover rate* ditunjukkan pada Tabel 4.4.

Tabel 4.4 Rancangan Uji Coba *Crossover Rate* dan *Mutation Rate*

Nilai <i>cr</i>	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
0											
0,1											
0,2											
0,3											
0,4											
0,5											
0,6											
0,7											
0,8											
0,9											
1											

4.2.5 Uji Coba *Mutation Rate*

Uji coba *mutation rate* dilakukan untuk mengetahui nilai *mr* terbaik agar mendapatkan hasil koefisien untuk mendapatkan hasil prediksi konsumsi BBM yang optimal. Uji coba *mr* ini menggunakan nilai *mr* yang berbeda dalam range 0 – 1. Rancangan uji coba nilai *mutation rate* ditunjukkan pada Tabel 4.5.

Tabel 4.5 Rancangan Uji Coba *Mutation Rate*

Nilai <i>mr</i>	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	

0												
0,1												
0,2												
0,3												
0,4												
0,5												
0,6												
0,7												
0,8												
0,9												
1												

4.2.6 Uji Coba Range Parameter KP1

Uji coba range parameter KP1 merupakan uji coba yang dilakukan untuk mengetahui seberapa panjang nilai range KP1 yang dapat menghasilkan nilai koefisien yang optimal. Rancang uji coba range parameter KP1 ditunjukkan pada Tabel 4.6.

Tabel 4.6 Rancang Uji Coba Range Parameter KP1

Kombinasi		Nilai Fitness Percobaan ke-i										Rata – rata Fitness
Min	Max	1	2	3	4	5	6	7	8	9	10	
0.0001	10											
	100											
	1000											
0.01	10											
	100											
	1000											
0	10											
	100											
	1000											

4.2.7 Uji Coba Range Parameter KP2

Uji coba range parameter KP2 merupakan uji coba yang dilakukan untuk mengetahui seberapa panjang nilai range KP2 yang dapat menghasilkan nilai koefisien yang optimal. Rancang uji coba range parameter KP2 ditunjukkan pada Tabel 4.7.

Tabel 4.7 Rancang Uji Coba Range Parameter KP2

Kombinasi		Nilai Fitness Percobaan ke-i										Rata – rata Fitness
Min	Max	1	2	3	4	5	6	7	8	9	10	
0.0001	10											

	100											
	1000											
0.01	10											
	100											
	1000											
0	10											
	100											
	1000											

4.2.8 Uji Coba Range Parameter λ

Uji coba *range* parameter λ merupakan uji coba yang dilakukan untuk mengetahui seberapa panjang nilai *range* λ yang dapat menghasilkan nilai koefisien yang optimal. Rancang uji coba *range* parameter λ ditunjukkan pada Tabel 4.8.

Tabel 4.8 Rancang Uji Coba Range Parameter λ

Kombinasi		Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
Min	Max	1	2	3	4	5	6	7	8	9	10	
0.0001	1											
	10											
	100											
0.001	1											
	10											
	100											
0.01	1											
	10											
	100											

4.2.9 Uji Coba Range Parameter γ

Uji coba *range* parameter γ merupakan uji coba yang dilakukan untuk mengetahui seberapa panjang nilai *range* γ yang dapat menghasilkan nilai koefisien yang optimal. Rancang uji coba *range* parameter γ ditunjukkan pada Tabel 4.9.

Tabel 4.9 Rancang Uji Coba Range Parameter γ

Kombinasi		Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
Min	Max	1	2	3	4	5	6	7	8	9	10	
0.0001	10											
	100											
	1000											

0.01	10												
	100												
	1000												
0	10												
	100												
	1000												

4.2.10 Uji Coba Range Parameter C

Uji coba *range* parameter C merupakan uji coba yang dilakukan untuk mengetahui seberapa panjang nilai *range* C yang dapat menghasilkan nilai koefisien yang optimal. Rancang uji coba *range* parameter C ditunjukkan pada Tabel 4.10.

Tabel 4.10 Rancang Uji Coba Range Parameter C

Kombinasi		Nilai <i>Fitness</i> Percobaan ke- i										Rata – rata <i>Fitness</i>	
Min	Max	1	2	3	4	5	6	7	8	9	10		
0.0001	1												
	10												
	100												
0.001	10												
	10												
	100												
0.01	1												
	10												
	100												

BAB 5 IMPLEMENTASI SISTEM

Pada bagian ini menjelaskan tentang implementasi sistem yang diperoleh dari bab sebelumnya yaitu perancangan, yang meliputi perancangan algoritma dan perancangan antarmuka.

5.1 Implementasi Program

Berdasarkan perancangan yang telah dibahas pada Bab 4, maka akan dibahas mengenai implementasi program sesuai dengan perancangan yang telah dibuat. Sistem diimplementasikan menggunakan bahasa pemrograman Java dengan aplikasi *Netbeans* versi 8.0.2.

5.1.1 Implementasi Proses Normalisasi Data

Proses ini diawali dengan melakukan normalisasi terhadap seluruh data yang akan diolah untuk standarisasi data, proses keseluruhan normalisasi dapat dilihat pada Source Code 5.1.

```

1 public void setNormalisasi() {
2     range = maxData - minData;
3     for (int i = 0; i < BBM.length; i++) {
4         for (int j = 0; j < BBM[i].length; j++) {
5             BBM[i][j] = (BBM[i][j] - minData) / range;
6         }
7     }
8 }

```

Source Code 5.1 Proses Normalisasi Data

Penjelasan dari Source Code 5.1 adalah sebagai berikut :

1. Baris 2 merupakan proses perhitungan *range* data
2. Baris 5 merupakan proses normalisasi menggunakan *min-max normalization*

5.1.2 Implementasi Proses *Sequential Learning*

Implementasi proses *sequential learning* pada Source Code 5.2.

```

1 public double[][] hitungKernel() {
2     double[][] tabelKernel = new
3     double[BBM.length][BBM.length];
4
5     for (int a = 0; a < BBM.length; a++) {
6         for (int b = 0; b < BBM.length; b++) {
7             double temp = 0;
8             for (int c = 0; c < BBM[a].length - 1; c++)
9             {
10                if (Kti == 0) {
11                    temp += (BBM[a][c] * BBM[b][c]);
12                }
13                if (Kti == 1) {
14                    temp += (Math.pow((BBM[b][c] *
15                    BBM[a][c] + KP2), KP1));
16                }
17            }
18            tabelKernel[a][b] = temp;
19        }
20    }
21    return tabelKernel;
22 }

```

```

13         }
14         if (Kti == 2) {
15             temp += (Math.pow((BBM[b][c] -
BBM[a][c]), 2));
16         }
17     }
18     if (Kti == 0 || Kti == 1) {
19         tabelKernel[a][b] = temp;
20     }
21     if (Kti == 2) {
22         tabelKernel[a][b] = Math.exp(-(temp / (2
* (Math.pow(KP2, 2)))));
23     }
24 }
25 }
26 return tabelKernel;
27 }
28 public double[][] hitungRij() {
29     double tabelKernel[][] = hitungKernel();
30     double[][] matriksRij = new
double[tabelKernel.length][tabelKernel.length];
31
32     for (int a = 0; a < tabelKernel.length; a++) {
33         for (int b = 0; b < tabelKernel.length; b++) {
34             matriksRij[a][b] = (tabelKernel[a][b] +
(Math.pow(lamda, 2)));
35         }
36     }
37     return matriksRij;
38 }
39 public double[][] seqLearning(int batasIterasi) {
40     double[][] tabelKernel = getTabelKernel();
41     double[][] matriksRij = getMatriksRij();
42     double[] nilaiE = new double[bbm.length];
43     double[] alfai = new double[bbm.length];
44     double[] alfaistar = new double[bbm.length];
45     double[] deltaalfai = new double[bbm.length];
46     double[] deltaalfaistar = new double[bbm.length];
47     double[][] matriksA = new
double[bbm.length][bbm.length];
48
49     int iterasi = 0;
50     double maxDAstar = deltaalfaistar[0];
51     double maxDA = deltaalfai[0];
52     while (iterasi != batasIterasi ||
(Math.abs(cekMaks(deltaalfaistar)) > eps &&
Math.abs(cekMaks(deltaalfaistar)) > eps)) {
53         for (int a = 0; a < bbm.length; a++) {
54             double sumRij = 0;
55             for (int b = 0; b < bbm.length; b++) {
56                 sumRij = sumRij + ((alfaistar[b] -
alfai[b]) * (matriksRij[b][1]));
57             }
58             nilaiE[a] = (bbm[a][bbm[a].length - 1]) -
sumRij;
59             deltaalfaistar[a] = Math.min(Math.max(gamma
* (nilaiE[a] - eps), -alfaistar[a]), com - alfaistar[a]);

```

```

60         deltaalfai[a] = Math.min(Math.max(gamma * (-
nilaiE[a] - eps), -alfai[a]), com - alfai[a]);
61         alfaistar[a] = alfaistar[a] +
deltaalfaistar[a];
62         alfai[a] = alfai[a] + deltaalfai[a];
63         matriksA[0][a] = alfaistar[a];
64         matriksA[1][a] = alfai[a];
65     }
66     iterasi++;
67 }
68 return matriksA;
69 }

```

Source Code 5.2 Proses Sequential Learning

Penjelasan dari Source Code 5.2 adalah sebagai berikut :

1. Baris 1-27 merupakan proses perhitungan Kernel data uji.
2. Baris 28-38 merupakan proses perhitungan Matriks Hessian SVR.
3. Baris 57 merupakan proses perhitungan nilai F(x).
4. Baris 59 merupakan proses perhitungan nilai E.
5. Baris 59-60 merupakan proses perhitungan $\delta\alpha^*$ dan $\delta\alpha$.
6. Baris 61-62 merupakan proses perhitungan α^* dan α .
7. Baris 63-64 merupakan proses memasukkan nilai α^* dan α dalam sebuah Matriks.

5.1.3 Implementasi Proses Denormalisasi Data

Proses denormalisasi data akan ditunjukkan pada Source Code 5.3.

```

1 public void setDeNorm() {
2     range = maxData - minData;
3     for (int i = 0; i < BBM.length; i++) {
4         for (int j = 0; j < BBM[i].length; j++) {
5             BBM[i][j] = (BBM[i][j] * range) + minData;
6         }
7     }
8 }

```

Source Code 5.3 Proses Denormalisasi Data

Penjelasan dari Source Code 5.3 adalah sebagai berikut :

1. Baris 2 merupakan proses perhitungan *range* data
2. Baris 5 merupakan proses denormalisasi yang didapat dari *reverse min-max normalization*.

5.1.4 Implementasi Proses Peramalan dan Error Rate

Implementasi proses peramalan dan *Error Rate* akan ditunjukkan pada Source Code 5.4.

```

1 public double getMapeTest(int KTi, double kp1, double kp2,
2 double lamda, double gamma, double com) {
3     this.KTi = KTi;
4     this.KP1 = kp1;
5     this.KP2 = kp2;
6     this.lamda = lamda;

```

```

6      this.com = com;
7      this.gamma = gamma;
8      double[][] tabelKernel = hitungKernelTest();
9      double[][] matriksRij = hitungRijTest();
10     double[][] matriksA = svrproses(batasIterasi);
11     double[] fx = new double[BBM_test.length];
12     double[] fx_denorm = new double[BBM_test.length];
13
14     for (int i = 0; i < fx.length; i++) {
15         double tempFx = 0;
16         for (int j = 0; j < fx.length; j++) {
17             tempFx += (matriksA[0][j] - matriksA[1][j])
18             * matriksRij[i][j];
19         }
20         fx[i] = tempFx;
21         fx[i] = (fx[i] * range) + minData;
22         fx_denorm[i] = fx[i];
23     }
24     setDeNormTest();
25     double tempMape = 0;
26     for (int b = 0; b < fx.length; b++) {
27         tempMape +=
28         (Math.abs((BBM_test[b][BBM_test[b].length - 1]) - (fx[b])) /
29         BBM_test[b][BBM_test[b].length - 1]);
30     }
31     setNormalisasiTest();
32     mape = (tempMape / fx.length) * 100;
33     return mape;

```

Source Code 5.4 Proses Peramalan dan *Error Rate*

Penjelasan dari Source Code 5.4 adalah sebagai berikut :

1. Baris 2-7 merupakan inisialisasi parameter proses.
2. Baris 8 merupakan proses pemanggilan fungsi untuk menghitung matriks kernel.
3. Baris 9 merupakan proses pemanggilan fungsi untuk menghitung matriks *hessian*.
4. Baris 10 merupakan proses pemanggilan fungsi proses *sequential learning* untuk mendapatkan nilai matriks alfa.
5. Baris 14-22 merupakan proses pengujian regresi.
6. Baris 24-29 merupakan proses perhitungan nilai *error rate*.

5.1.5 Implementasi Proses Representasi Kromosom

Pembangkitan populasi awal dilakukan dengan merepresentasikan kromosom berdasarkan hasil pembangkitan *popSize* yang diinputkan oleh *user* serta data konsumsi BBM yang sebelumnya telah di-load. Proses representasi kromosom serta pembangkitan populasi awal ditunjukkan pada Source Code 5.5.

```

1     public void setKromosom(){
2         populasi = new double[popSize][panjangIndividu];
3         for (int a = 0; a < popSize; a++) {
4             for (int b = 0; b < panjangIndividu; b++) {

```



```

5         if (b == 0) {
6             this.populasi[a][b] = random.nextInt(3);
7         } else {
8             this.populasi[a][b] =
9             (random.nextDouble() * (range[b][1] - range[b][0])) +
10            range[b][0];
11         }
12     }

```

Source Code 5.5 Proses Representasi Kromosom

Penjelasan dari Source Code 5.5 adalah sebagai berikut :

1. Baris 2 merupakan inisialisasi populasi.
2. Baris 3-8 merupakan proses random untuk masing-masing kromosom pada tiap individu sebanyak popSize.

5.1.6 Implementasi Proses Crossover

Pada Source Code 5.6 merupakan penjabaran dari implementasi proses *crossover*.

```

1 public double[][] adewuyaCrossover(double[][] populasi, int
2   offCR) {
3     int offspringCR = offCR // 2;
4
5     if (offspringCR == 0) {
6         offspringCR = offspringCR + 1;
7         System.out.println(offspringCR);
8     }
9     double hasilCrossover[][] = new double[(int)
10    offCR][panjangKromosom];
11    int pCR = 0;
12    for (int c = 0; c < offspringCR; c++) {
13        double alphaCR[] = new double[panjangKromosom];
14        for (int k = 0; k < panjangKromosom; k++) {
15            alphaCR[k] = random.nextDouble();
16        }
17
18        int IndexP1, IndexP2;
19        do {
20            IndexP1 = random.nextInt(popSize);
21            IndexP2 = random.nextInt(popSize);
22        } while (IndexP1 == IndexP2);
23
24        int count = 1 + (random.nextInt(panjangKromosom
25        - 1)); //random banyak kromosom yang akan diproses
26        int[] indCrossed = new int[count];
27
28        for (int a = 0; a < count; a++) {
29            indCrossed[a] = 1 +
30            (random.nextInt(panjangKromosom - 1));
31        }
32        for (int a = 0; a < indCrossed.length; a++) {
33            for (int b = 0; b < a; b++) {

```

```

30         if (indCrossed[a] == indCrossed[b]) {
31             indCrossed[a] = 1 +
32             (random.nextInt (panjangKromosom - 1));
33             a--;
34         }
35     }
36     for (int x = 0; x < panjangKromosom; x++) {
37         hasilCrossover[pCR][x] =
38         populasi[IndexP1][x];
39         hasilCrossover[pCR + 1][x] =
40         populasi[IndexP2][x];
41         for (int i = 0; i < indCrossed.length; i++) {
42             hasilCrossover[pCR][indCrossed[i]] =
43             populasi[IndexP1][indCrossed[i]] + (alphaCR[indCrossed[i]]
44             * (populasi[IndexP1][indCrossed[i]] -
45             populasi[IndexP2][indCrossed[i]]));
46             hasilCrossover[pCR + 1][indCrossed[i]] =
47             populasi[IndexP2][indCrossed[i]] - (alphaCR[indCrossed[i]]
48             * (populasi[IndexP1][indCrossed[i]] -
49             populasi[IndexP2][indCrossed[i]]));
50             if (hasilCrossover[pCR][indCrossed[i]] <
51             range[indCrossed[i]][0] ||
52             hasilCrossover[pCR][indCrossed[i]] >
53             range[indCrossed[i]][1]) {
54                 hasilCrossover[pCR][indCrossed[i]] =
55                 populasi[IndexP1][indCrossed[i]] + (alphaCR[indCrossed[i]]
56                 * (populasi[IndexP2][indCrossed[i]] -
57                 populasi[IndexP1][indCrossed[i]]));
58                 } else if (hasilCrossover[pCR +
59                 1][indCrossed[i]] <= range[indCrossed[i]][0] ||
60                 hasilCrossover[pCR + 1][indCrossed[i]] >
61                 range[indCrossed[i]][1]) {
62                     hasilCrossover[pCR + 1][indCrossed[i]]
63                     = populasi[IndexP2][indCrossed[i]] -
64                     (alphaCR[indCrossed[i]] * (populasi[IndexP2][indCrossed[i]]
65                     - populasi[IndexP1][indCrossed[i]]));
66                 }
67             if (hasilCrossover[pCR][indCrossed[i]] <
68             range[indCrossed[i]][0] ||
69             hasilCrossover[pCR][indCrossed[i]] >
70             range[indCrossed[i]][1]) {
71                 hasilCrossover[pCR][indCrossed[i]] =
72                 populasi[IndexP1][indCrossed[i]];
73                 } else if (hasilCrossover[pCR +
74                 1][indCrossed[i]] < range[indCrossed[i]][0] ||
75                 hasilCrossover[pCR + 1][indCrossed[i]] >
76                 range[indCrossed[i]][1]) {
77                     hasilCrossover[pCR + 1][indCrossed[i]]
78                     = populasi[IndexP2][indCrossed[i]];
79                 }
80             }
81         }
82     }
83 }

```

Source Code 5.6 Proses Crossover

Penjelasan dari Source Code 5.6 adalah sebagai berikut :

1. Baris 2-3 merupakan inisialisasi parameter proses.

2. Baris 5-7 merupakan proses random untuk nilai α pada *crossover*.
3. Baris 11-14 merupakan proses untuk random index untuk parent *crossover*.
4. Baris 16-51 merupakan proses *crossover*.

5.1.7 Implementasi Proses Mutasi

Proses mutasi akan dijabarkan pada Source Code 5.7.

```

1      public double[][] uniformMutation(double[][] populasi)
2      {
3          double hasil[][] = new double[1][panjangIndividu];
4          double rmut[] = new double[panjangIndividu];
5          for (int b = 0; b < panjangIndividu; b++) {
6              rmut[b] = random.nextDouble();
7          }
8
9          for (int i = 0; i < panjangIndividu; i++) {
10             if (i == 0) {
11                 hasil[0][i] = Math.round(range[i][0] +
12 (rmut[i] * (range[i][1] - range[i][0])));
13             } else {
14                 hasil[0][i] = range[i][0] + (rmut[i] *
15 (range[i][1] - range[i][0]));
16             }
17             if (hasil[0][i] < range[i][0]) {
18                 hasil[0][i] = range[i][0];
19             } else if (hasil[0][i] > range[i][1]) {
20                 hasil[0][i] = range[i][1];
21             }
22             return hasil;
23         }
24     }

```

Source Code 5.7 Proses Mutasi

Penjelasan dari Source Code 5.7 adalah sebagai berikut :

1. Baris 2-3 merupakan inisialisasi parameter pada proses.
2. Baris 5-7 merupakan proses random nilai r pada mutasi.
3. Baris 9-20 merupakan proses mutasi.

5.1.8 Implementasi Proses Seleksi

Pembangkitan populasi awal dilakukan dengan merepresentasikan kromosom berdasarkan hasil pembangkitan *popSize* yang diinputkan oleh *user* serta data konsumsi BBM yang sebelumnya telah di-load. Proses representasi kromosom serta pembangkitan populasi awal ditunjukkan pada Source Code 5.8.

```

1      public double[][] seleksi(double[][] hasilPopulasi) {
2          double hasilSeleksi[][] = new
3          double[popSize][panjangIndividu];
4          double mape;
5          double fitness[] = new double[hasilPopulasi.length];
6          int[] IndexAwalPopulasiGabungan = new
7          int[hasilPopulasi.length];

```

```

6         for (int i = 0; i < hasilPopulasi.length; i++) {
7             IndexAwalPopulasiGabungan[i] = i;
8         }
9         for (int a = 0; a < hasilPopulasi.length; a++) {
10            fitness[a] = getFitness((int)
11            hasilPopulasi[a][0], hasilPopulasi[a][1],
12            hasilPopulasi[a][2], hasilPopulasi[a][3],
13            hasilPopulasi[a][4], hasilPopulasi[a][5]);
14        }
15        for (int i = 0; i < hasilPopulasi.length; i++) {
16            double TempNilaiFitness = fitness[i];
17            int TempIndexAwalPopulasiGabungan =
18            IndexAwalPopulasiGabungan[i];
19            double[][] TempHasil = new
20            double[1][panjangIndividu];
21            for (int j = (i + 1); j < hasilPopulasi.length;
22            j++) {
23                if (fitness[j] > TempNilaiFitness) {
24                    TempNilaiFitness = fitness[j];
25                    fitness[j] = fitness[i];
26                    fitness[i] = TempNilaiFitness;
27                    TempIndexAwalPopulasiGabungan =
28                    IndexAwalPopulasiGabungan[j];
29                    IndexAwalPopulasiGabungan[j] =
30                    IndexAwalPopulasiGabungan[i];
31                    IndexAwalPopulasiGabungan[i] =
32                    TempIndexAwalPopulasiGabungan;
33                    System.arraycopy(hasilPopulasi[j], 0,
34                    TempHasil[0], 0, panjangIndividu);
35                    System.arraycopy(hasilPopulasi[i], 0,
36                    hasilPopulasi[j], 0, panjangIndividu);
37                    System.arraycopy(TempHasil[0], 0,
38                    hasilPopulasi[i], 0, panjangIndividu);
39                }
40            }
41        }
42        for (int a = 0; a < populasi.length; a++) {
43            for (int b = 0; b < panjangIndividu; b++) {
44                populasi[a][b] = hasilPopulasi[a][b];
45                bestParent[0][b] = populasi[0][b];
46                setbestParent(bestParent, fitness[0]);
47            }
48        }
49        return hasilSeleksi;
50    }

```

Source Code 5.8 Proses Seleksi

Penjelasan dari Source Code 5.8 adalah sebagai berikut :

1. Baris 2-5 merupakan inialisasi parameter proses.
2. Baris 6-8 merupakan proses inialisasi index untuk populasi gabungan.
3. Baris 9-11 merupakan proses pemanggilan fungsi *getFitness* untuk mendapatkan nilai *fitness* pada masing-masing individu.
4. Baris 13-31 merupakan proses seleksi.

5. Baris 32-37 merupakan proses penentuan populasi untuk generasi selanjutnya dan menentukan *bestParent* pada generasi.

5.2 Implementasi Antarmuka

Sub bab Implementasi Antarmuka merupakan pembahasan mengenai hasil implementasi antarmuka yang sebelumnya dirancang pada Bab 4.

5.2.1 Halaman Proses *Training*

Halaman proses *training* digunakan untuk memasukkan inputan parameter Algoritma SVR dan HGA, batas – batas parameter yang digunakan, serta perolehan hasil peramalan yang telah diproses. Halaman proses *training* dapat dilihat pada Gambar 5.1.

The screenshot shows a software window titled "PERAMALAN KONSUMSI BAHAN BAKAR MINYAK DI INDONESIA DENGAN MENGGUNAKAN HYBRID ALGORITMA GENETIKA DAN SUPPORT VECTOR REGRESSION". The interface includes a "Parameter" section with input fields for Iterasi SVR, Generasi HGA, PopSize, Crossover Rate, and Mutation Rate. Below this is a "Range Parameter" section with input fields for KT, KP1, KP2, λ, γ, and C. A "Proses Training" button is located to the right of the parameter fields. The main area features a tabbed interface with three tabs: "Data Awal", "Hasil Peramalan", and "Parameter Terbaik". The "Hasil Peramalan" tab is active, showing a table with two columns: "Tahun" and "Jumlah Konsumsi". At the bottom of the window, there is a "MAPE" input field and a "Proses Testing" button.

Gambar 5.1 Halaman proses *training*

5.2.2 Halaman Proses *Testing*

Halaman proses *testing* digunakan untuk memasukkan nilai parameter yang optimal serta untuk mendapatkan hasil peramalan. Perancangan halaman parameter algoritma dan hasil peramalan dapat dilihat pada Gambar 5.2.

**PERAMALAN KONSUMSI BAHAN BAKAR MINYAK DI INDONESIA
DENGAN MENGGUNAKAN HYBRID ALGORITMA GENETIKA DAN SUPPORT VECTOR REGRESSION**

Parameter

Iterasi SVR

KT

KP1

KP2

λ

γ

C

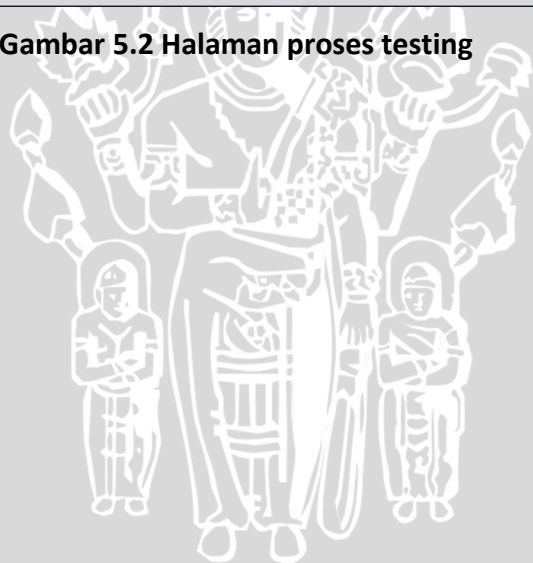
ϵ

Proses Testing

Data Aktual Testing		Hasil Peramalan Testing	
Tahun		Jumlah Konsumsi	

MAPE

Gambar 5.2 Halaman proses testing



BAB 6 PENGUJIAN DAN ANALISIS

Bab ini berisi hasil pengujian serta analisis dari hasil uji coba yang telah dilakukan dalam peramalan konsumsi BBM menggunakan metode HGA-SVR. Pada pengujian peramalan konsumsi BBM digunakan data *training* sebanyak 34 data yang diambil mulai tahun 1969-2002 dan data *testing* sebanyak 11 data yang diambil mulai tahun 2003-2013.

6.1 Hasil dan Analisa Uji Coba Jumlah Iterasi SVR

Untuk pengujian pada jumlah iterasi SVR yang digunakan adalah 10, 50, 100, 500, 1000, 5000, 10000, 50000, dan 100000. Detail parameter yang digunakan pada uji coba iterasi SVR adalah sebagai berikut :

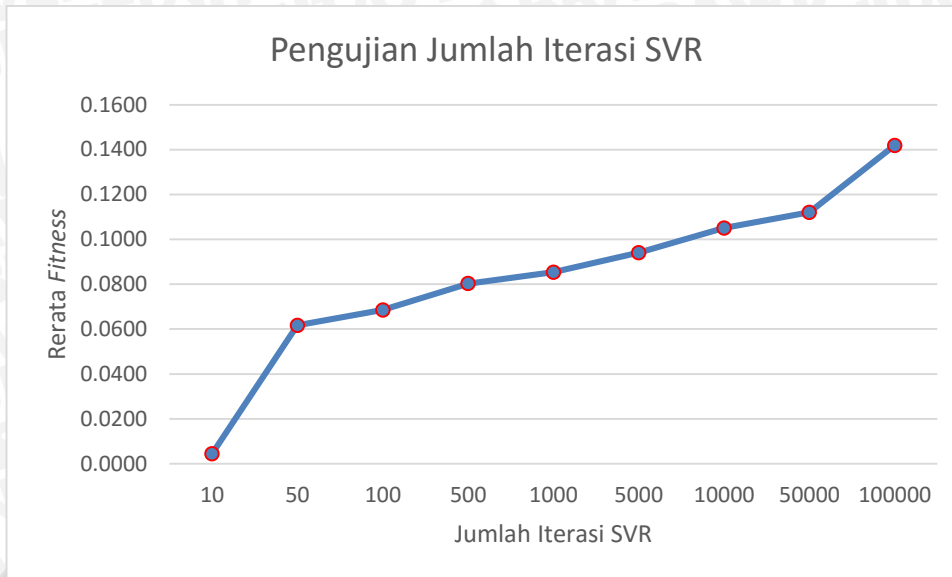
- a. Generasi : 155
- b. *PopSize* : 50
- c. *Crossover Rate* : 0.4
- d. *Mutation Rate* : 0.6
- e. Range parameter *KTi* : 0 - 2
- f. Range parameter *KP1* : 0 - 100
- g. Range parameter *KP2* : 0 - 100
- h. Range parameter λ : 0.0001 - 10
- i. Range parameter γ : 0 - 1000
- j. Range parameter *C* : 0.01 - 10

Hasil pengujian jumlah iterasi SVR yang digunakan dapat dilihat pada Tabel 6.1.

Tabel 6.1 Rancangan Uji Coba Jumlah Iterasi SVR

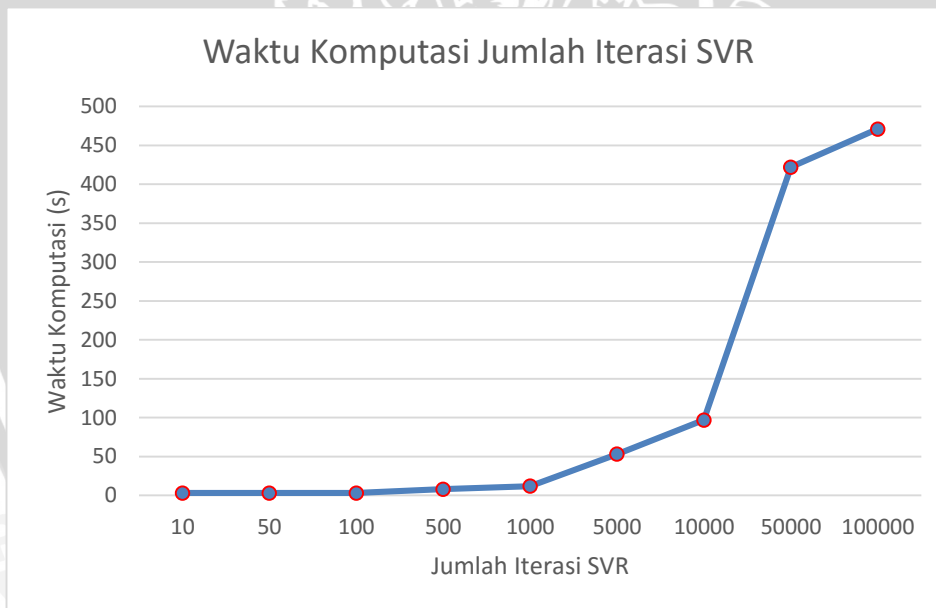
Jumlah Iterasi SVR	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
10	0.0055	0.0028	0.0055	0.0055	0.0056	0.0055	0.0055	0.0012	0.0055	0.0015	0.0044
50	0.0811	0.0053	0.1040	0.0915	0.0014	0.1145	0.1136	0.0014	0.1025	0.0013	0.0617
100	0.1046	0.0106	0.1140	0.1031	0.1096	0.0115	0.1041	0.0021	0.1246	0.0012	0.0685
500	0.1243	0.0270	0.1043	0.0922	0.1092	0.0245	0.1053	0.1005	0.0304	0.0852	0.0803
1000	0.0653	0.1029	0.0683	0.0631	0.1096	0.0652	0.0735	0.1399	0.0647	0.1015	0.0854
5000	0.1038	0.0653	0.0655	0.1145	0.0619	0.1418	0.0655	0.1423	0.1141	0.0659	0.0941
10000	0.1088	0.1353	0.1036	0.1254	0.1032	0.1118	0.0355	0.1123	0.1114	0.1036	0.1051
50000	0.1053	0.1495	0.1083	0.1031	0.1041	0.1052	0.1135	0.1223	0.1047	0.1047	0.1121
100000	0.1463	0.1485	0.1395	0.1453	0.1181	0.1454	0.1412	0.1497	0.1458	0.1391	0.1419

Berdasarkan grafik hasil pengujian pada Gambar 6.1 dengan data *training* dan data *testing* yang bervariasi, didapatkan nilai paling optimal untuk peramalan pada iterasi sebanyak 100000. Tentunya semakin banyak iterasi yang digunakan, nilai α dan α^* yang lebih baik dan menghasilkan nilai peramalan yang cukup baik.



Gambar 6.1 Grafik Uji Iterasi SVR

Gambar 6.2 menunjukkan waktu komputasi yang digunakan dengan jumlah iterasi tertentu, penggunaan iterasi SVR yang cukup besar memakan waktu komputasi yang cukup lama, walaupun dengan jaminan hasil peramalan yang cukup akurat.



Gambar 6.2 Grafik Waktu Komputasi Jumlah Iterasi SVR

6.2 Hasil dan Analisa Uji Coba Jumlah Generasi HGA

Untuk pengujian pada jumlah generasi HGA yang digunakan adalah 50, 65, 80, 95, 110, 125, 140, 155, dan 170. Detail parameter yang digunakan pada uji coba generasi HGA adalah sebagai berikut :

- a. Iterasi SVR : 1000

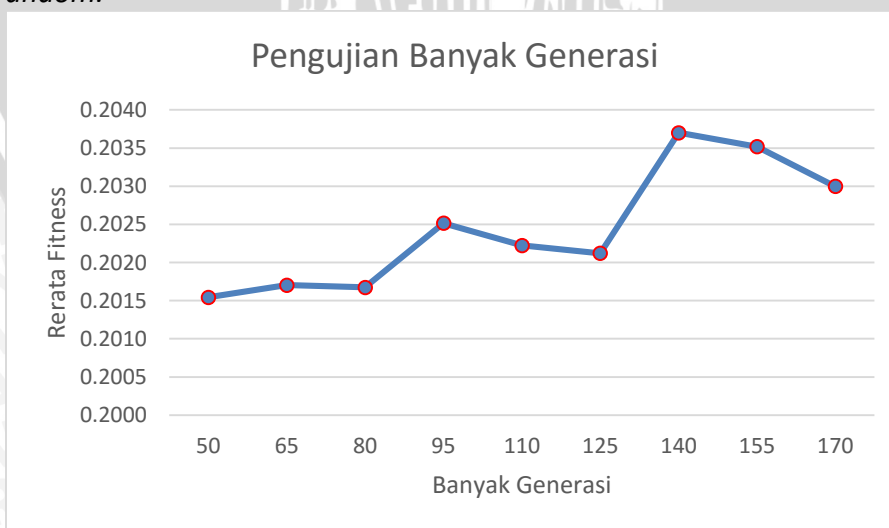
- b. *PopSize* : 50
- c. *Crossover Rate* : 0.4
- d. *Mutation Rate* : 0.6
- e. Range parameter *KTi* : 0 - 2
- f. Range parameter *KP1* : 0 - 100
- g. Range parameter *KP2* : 0 - 100
- h. Range parameter λ : 0.0001 - 10
- i. Range parameter γ : 0 - 1000
- j. Range parameter *C* : 0.01 - 10

Hasil pengujian jumlah iterasi SVR yang digunakan dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Uji Coba Jumlah Generasi HGA

Jumlah generasi	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
50	0.2013	0.2019	0.2014	0.2012	0.2008	0.2016	0.2044	0.1988	0.2038	0.2004	0.2015
65	0.2010	0.2014	0.2010	0.2012	0.2063	0.2015	0.2007	0.2014	0.2014	0.2012	0.2017
80	0.2015	0.2045	0.2011	0.2012	0.2014	0.2014	0.2016	0.2014	0.2014	0.2013	0.2017
95	0.2019	0.2014	0.2064	0.2014	0.2014	0.2014	0.2024	0.2014	0.2013	0.2064	0.2025
110	0.2051	0.2014	0.2014	0.2014	0.2014	0.2064	0.2014	0.2013	0.2014	0.2012	0.2022
125	0.2040	0.2014	0.2014	0.2014	0.2014	0.2013	0.2014	0.2012	0.2064	0.2014	0.2021
140	0.2014	0.2036	0.2064	0.2064	0.2014	0.2014	0.2064	0.2064	0.2025	0.2014	0.2037
155	0.2014	0.2014	0.2119	0.2010	0.2027	0.2014	0.2064	0.2013	0.2014	0.2064	0.2035
170	0.2012	0.2016	0.2025	0.2014	0.2014	0.2064	0.2014	0.2064	0.2014	0.2064	0.2030

Berdasarkan grafik hasil pengujian pada Gambar 6.3 dengan data *training* dan data *testing* yang bervariasi, didapatkan nilai paling optimal untuk peramalan pada generasi sebanyak 140. Jumlah generasi yang terlalu sedikit dapat mengakibatkan kromosom yang terbentuk terlalu sedikit sehingga kromosom baru yang terbentuk tidak mengalami banyak perubahan. Sedangkan jumlah generasi yang terlalu banyak tidak memberikan jaminan bahwa kromosom yang dihasilkan akan mengalami peningkatan. Hal ini dikarenakan sifat inisialisasi kromosom awal secara *random*.



Gambar 6.3 Grafik Uji Generasi HGA

6.3 Hasil dan Analisa Uji Coba Ukuran Populasi

Untuk pengujian pada ukuran populasi yang digunakan adalah 20, 25, 30, 35, 40, 45, 50, 55, 60, 65 dan 70. Detail parameter yang digunakan pada uji coba populasi adalah sebagai berikut :

- a. Iterasi SVR : 1000
- b. Generasi : 155
- c. *Crossover Rate* : 0.4
- d. *Mutation Rate* : 0.6
- e. Range parameter *KTi* : 0 - 2
- f. Range parameter *KP1* : 0 - 100
- g. Range parameter *KP2* : 0 - 100
- h. Range parameter λ : 0.0001 - 10
- i. Range parameter γ : 0 - 1000
- j. Range parameter *C* : 0.01 - 10

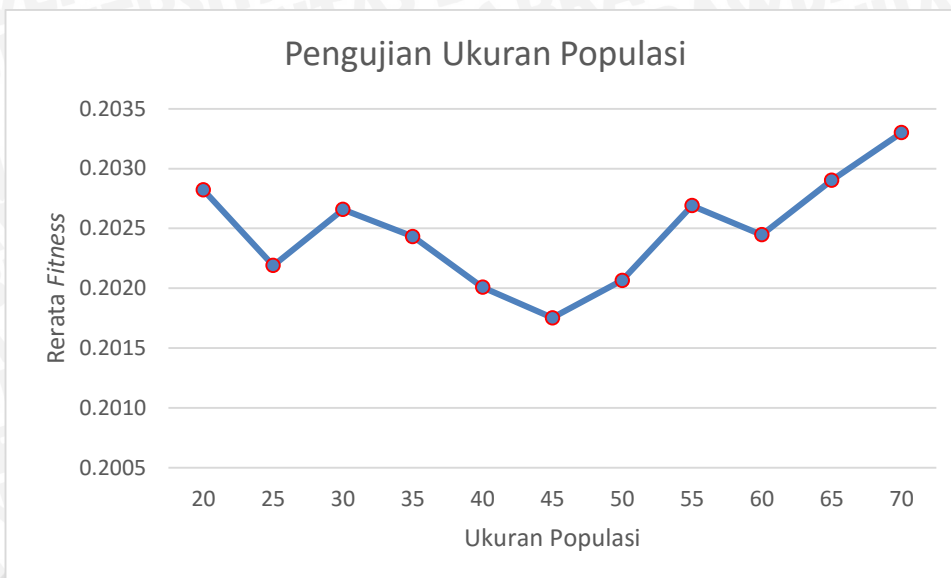
Hasil uji coba ukuran populasi ditunjukkan pada Tabel 6.3.

Tabel 6.3 Hasil Uji Coba Ukuran Populasi

Banyak populasi	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
20	0.2064	0.2013	0.2013	0.2062	0.2013	0.2063	0.2014	0.2014	0.2014	0.2013	0.2028
25	0.2014	0.2013	0.2047	0.2014	0.2014	0.2014	0.2013	0.2064	0.2013	0.2014	0.2022
30	0.2025	0.2012	0.2061	0.2010	0.2013	0.2024	0.2027	0.2064	0.2016	0.2014	0.2027
35	0.2014	0.2023	0.2012	0.2014	0.2014	0.2013	0.2014	0.2064	0.2061	0.2014	0.2024
40	0.2014	0.2012	0.2014	0.2013	0.2014	0.2014	0.2064	0.2013	0.2021	0.2022	0.2020
45	0.2054	0.2012	0.2014	0.2012	0.2016	0.2013	0.2014	0.2013	0.2014	0.2014	0.2018
50	0.2014	0.2014	0.2034	0.2014	0.2013	0.2014	0.2014	0.2014	0.2063	0.2013	0.2021
55	0.2014	0.2037	0.2014	0.2064	0.2063	0.2014	0.2014	0.2014	0.2023	0.2014	0.2027
60	0.2014	0.2030	0.2012	0.2064	0.2058	0.2014	0.2012	0.2014	0.2014	0.2014	0.2024
65	0.2014	0.2013	0.2036	0.2014	0.2014	0.2013	0.2012	0.2121	0.2014	0.2040	0.2029
70	0.2034	0.2014	0.2014	0.2014	0.2021	0.2022	0.2079	0.2062	0.2051	0.2019	0.2033

Berdasarkan grafik hasil pengujian pada Gambar 6.4 dengan data *training* dan data *testing* yang bervariasi, didapatkan nilai paling optimal untuk peramalan pada populasi sebanyak 70 karena pada ukuran populasi sebesar 70 dengan kelipatan 5 pada ukuran populasi selanjutnya tidak mengalami kenaikan nilai rata-rata *fitness* dan cenderung stabil.

Semakin besar ukuran populasi maka akan menghasilkan nilai *fitness* semakin besar juga, namun setelah batas tertentu (pada kasus ini sebesar 70) kenaikan yang didapatkan semakin tidak signifikan. Pembatasan ukuran populasi perlu dilakukan karena semakin banyak ukuran populasi maka nilai *fitness* yang didapatkan cenderung stabil.



Gambar 6.4 Grafik Uji Populasi

6.4 Hasil dan Analisa Uji Coba Crossover Rate

Untuk pengujian pada uji coba *crossover rate* yang digunakan adalah 0 hingga 1 dengan kelipatan 0.1. Detail parameter yang digunakan pada uji coba *crossover rate* adalah sebagai berikut :

- a. Iterasi SVR : 1000
- b. Generasi : 155
- c. *PopSize* : 50
- d. *Mutation Rate* : 0.6
- e. Range parameter *KTi* : 0 - 2
- f. Range parameter *KP1* : 0 - 100
- g. Range parameter *KP2* : 0 - 100
- h. Range parameter λ : 0.0001 - 10
- i. Range parameter γ : 0 - 1000
- j. Range parameter *C* : 0.01 - 10

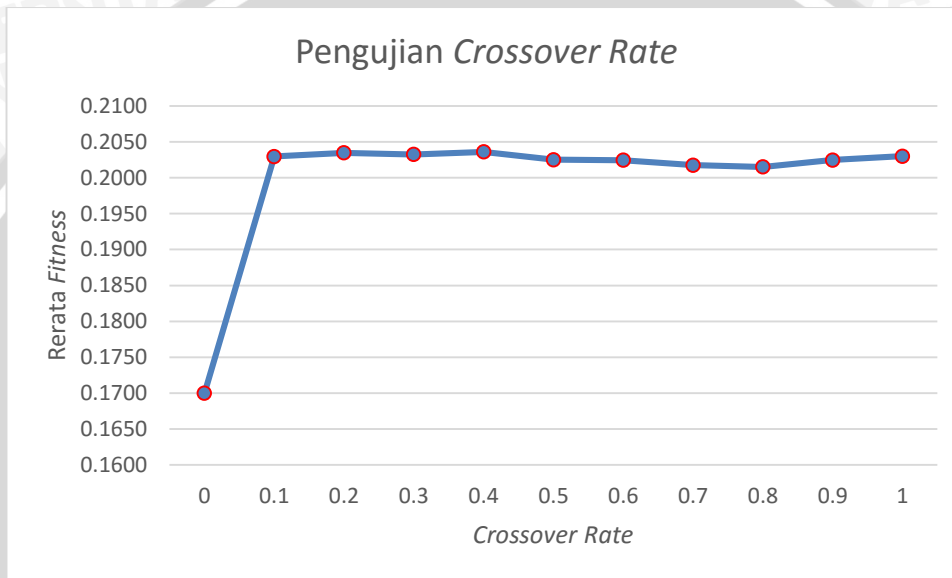
Hasil uji coba nilai *crossover rate* ditunjukkan pada Tabel 6.4.

Tabel 6.4 Hasil Uji Coba Crossover Rate

Nilai <i>cr</i>	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
0	0.1924	0.1457	0.2052	0.1943	0.1827	0.1635	0.0835	0.1550	0.1898	0.1877	0.1700
0,1	0.2012	0.2011	0.2052	0.2012	0.2010	0.2056	0.2012	0.2062	0.2058	0.2013	0.2030
0,2	0.2014	0.2013	0.2035	0.2012	0.2064	0.2014	0.2063	0.2065	0.2062	0.2006	0.2035
0,3	0.2019	0.2121	0.2064	0.2030	0.2024	0.2014	0.2014	0.2014	0.2014	0.2012	0.2033
0,4	0.2014	0.2063	0.2014	0.2014	0.2064	0.2064	0.2064	0.2038	0.2014	0.2014	0.2036
0,5	0.2014	0.2014	0.2014	0.2034	0.2064	0.2014	0.2014	0.2014	0.2057	0.2014	0.2025
0,6	0.2064	0.2014	0.2033	0.2014	0.2052	0.2014	0.2014	0.2014	0.2013	0.2014	0.2024

0,7	0.2026	0.2014	0.2014	0.2012	0.2014	0.2014	0.2014	0.2014	0.2019	0.2035	0.2018
0,8	0.2014	0.2014	0.2019	0.2014	0.2014	0.2014	0.2014	0.2012	0.2024	0.2014	0.2015
0,9	0.2013	0.2014	0.2013	0.2014	0.2077	0.2012	0.2014	0.2014	0.2064	0.2014	0.2025
1	0.2014	0.2014	0.2014	0.2102	0.2033	0.2014	0.2019	0.2064	0.2014	0.2014	0.2030

Berdasarkan grafik hasil pengujian pada Gambar 6.5 dengan data *training* dan data *testing* yang bervariasi, didapatkan nilai paling optimal untuk peramalan pada *crossover rate* sebesar 0.4. *Crossover rate* yang terlalu tinggi dapat mengakibatkan algoritma genetika tidak mampu memperluas area pencarian. Hal ini dapat mengakibatkan kromosom yang dihasilkan pada generasi berikutnya tidak mengalami kenaikan nilai *fitness*.



Gambar 6.5 Grafik Uji Crossover Rate

6.5 Hasil dan Analisa Uji Coba Mutation Rate

Untuk pengujian pada uji coba *mutation rate* yang digunakan adalah 0 hingga 1 dengan kelipatan 0.1. Detail parameter yang digunakan pada uji coba *mutation rate* adalah sebagai berikut :

- a. Iterasi SVR : 1000
- b. Generasi : 155
- c. *PopSize* : 50
- d. *Crossover Rate* : 0.4
- e. Range parameter *KT_i* : 0 - 2
- f. Range parameter *KP₁* : 0 - 100
- g. Range parameter *KP₂* : 0 - 100
- h. Range parameter λ : 0.0001 - 10
- i. Range parameter γ : 0 - 1000
- j. Range parameter *C* : 0.01 - 10

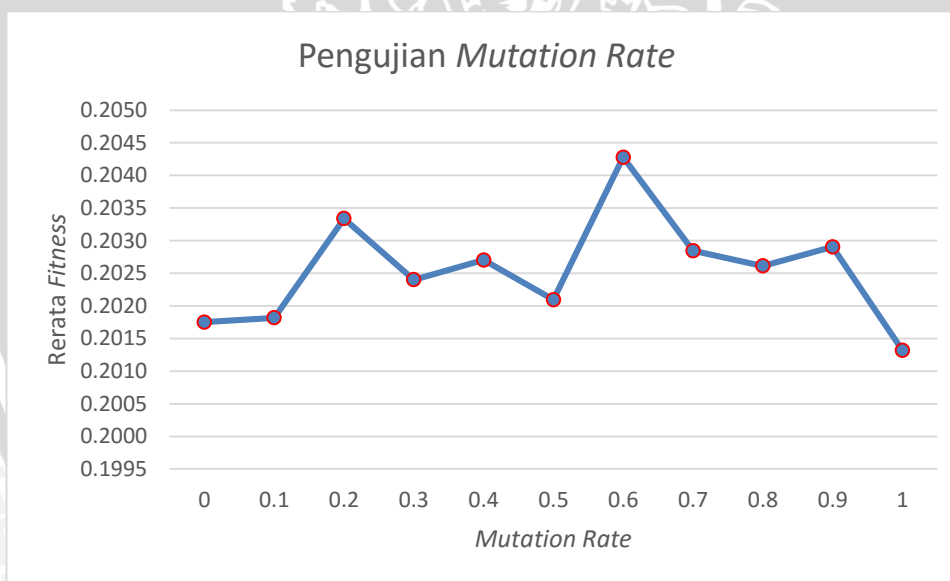
Hasil uji coba nilai *mutation rate* ditunjukkan pada Tabel 6.5.



Tabel 6.5 Hasil Uji Coba Mutation Rate

Nilai <i>mr</i>	Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
0	0.2052	0.2014	0.2014	0.2014	0.2014	0.2012	0.2014	0.2014	0.2014	0.2014	0.2018
0,1	0.2014	0.2014	0.2014	0.2057	0.2013	0.2014	0.2014	0.2014	0.2014	0.2014	0.2018
0,2	0.2077	0.2064	0.2014	0.2062	0.2014	0.2025	0.2014	0.2014	0.2037	0.2014	0.2033
0,3	0.2025	0.2054	0.2062	0.2014	0.2014	0.2014	0.2014	0.2014	0.2025	0.2005	0.2024
0,4	0.2064	0.2014	0.2058	0.2014	0.2013	0.2014	0.2020	0.2046	0.2014	0.2014	0.2027
0,5	0.2014	0.2014	0.2021	0.2025	0.2014	0.2014	0.2014	0.2023	0.2014	0.2057	0.2021
0,6	0.2064	0.2014	0.2058	0.2048	0.2014	0.2064	0.2021	0.2067	0.2014	0.2064	0.2043
0,7	0.2064	0.2013	0.2014	0.2064	0.2063	0.2014	0.2001	0.2025	0.2014	0.2014	0.2028
0,8	0.2051	0.2014	0.2036	0.2014	0.2013	0.2013	0.2028	0.2014	0.2014	0.2065	0.2026
0,9	0.2014	0.2014	0.2019	0.2064	0.2013	0.2064	0.2014	0.2064	0.2012	0.2014	0.2029
1	0.2014	0.2013	0.2014	0.2013	0.1996	0.2014	0.2014	0.2014	0.2014	0.2026	0.2013

Berdasarkan grafik hasil pengujian pada Gambar 6.6 dengan data *training* dan data *testing* yang bervariasi, didapatkan nilai paling optimal untuk peramalan pada *mutation rate* sebesar 0.6. *Mutation rate* yang terlalu tinggi dapat mengakibatkan algoritma genetika akan bekerja seperti *random search* dan tidak mampu mengeksplorasi daerah pencarian secara efektif.



Gambar 6.6 Grafik Uji Mutation Rate

6.6 Hasil dan Analisa Uji Coba Range Parameter KPI

Untuk pengujian pada nilai *range* parameter yang digunakan berdasarkan pada Tabel 4.6. Detail parameter yang digunakan adalah sebagai berikut :

- a. Iterasi SVR : 1000
- b. Generasi : 155
- c. *PopSize* : 50

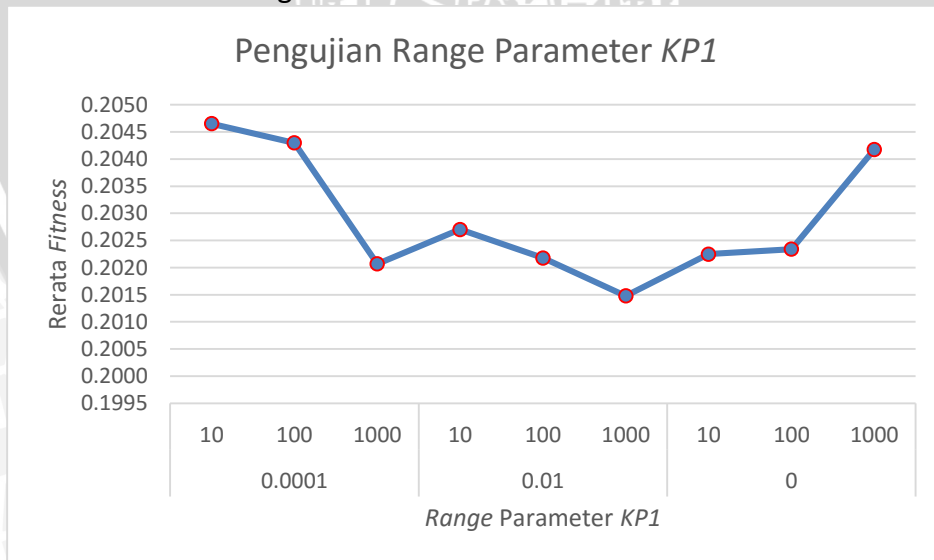
- d. *Crossover Rate* : 0.4
- e. *Mutation Rate* : 0.6
- f. Range parameter *KTi* : 1 - 2
- g. Range parameter *KP2* : 0 - 100
- h. Range parameter λ : 0.0001 - 10
- i. Range parameter γ : 0 - 1000
- j. Range parameter *C* : 0.01 - 10

Hasil uji coba *range* parameter *KP1* ditunjukkan pada Tabel 6.6.

Tabel 6.6 Hasil Uji Coba Range Parameter *KP1*

Kombinasi		Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
Min	Max	1	2	3	4	5	6	7	8	9	10	
0.0001	10	0.2064	0.2014	0.2014	0.2069	0.2014	0.2038	0.2014	0.2149	0.2075	0.2014	0.2047
	100	0.1999	0.2064	0.2064	0.2014	0.2042	0.2143	0.2014	0.2014	0.2014	0.2064	0.2043
	1000	0.2014	0.2044	0.2012	0.2012	0.2012	0.2014	0.2013	0.2014	0.2062	0.2012	0.2021
0.01	10	0.2036	0.2014	0.2064	0.2014	0.2013	0.2026	0.2013	0.2020	0.2058	0.2014	0.2027
	100	0.2057	0.2014	0.2035	0.2014	0.2014	0.2014	0.2044	0.1992	0.2014	0.2021	0.2022
	1000	0.2014	0.2013	0.2013	0.2013	0.2013	0.2011	0.2030	0.2014	0.2014	0.2014	0.2015
0	10	0.2014	0.2014	0.2014	0.2014	0.2054	0.2014	0.2014	0.2011	0.2064	0.2013	0.2022
	100	0.2027	0.2014	0.2014	0.2014	0.2018	0.2064	0.2022	0.2014	0.2013	0.2034	0.2023
	1000	0.2023	0.2064	0.2014	0.2063	0.2069	0.2063	0.2014	0.2014	0.2064	0.2030	0.2042

Berdasarkan grafik hasil pengujian pada Gambar 6.7 dengan data *training* dan data *testing* yang bervariasi, didapatkan nilai paling optimal untuk peramalan pada *range* 0.0001 - 10. Parameter *KP1* mempengaruhi proses penjumlahan pada *polynomial kernel*, apabila *KP1* terlalu besar maka hasil parameter yang didapatkan belum tentu bagus.



Gambar 6.7 Grafik Uji Range Parameter *KP1*

6.7 Hasil dan Analisa Uji Coba *Range* Parameter *KP2*

Untuk pengujian pada nilai *range* parameter yang digunakan berdasarkan pada Tabel 4.7. Detail parameter yang digunakan adalah sebagai berikut :

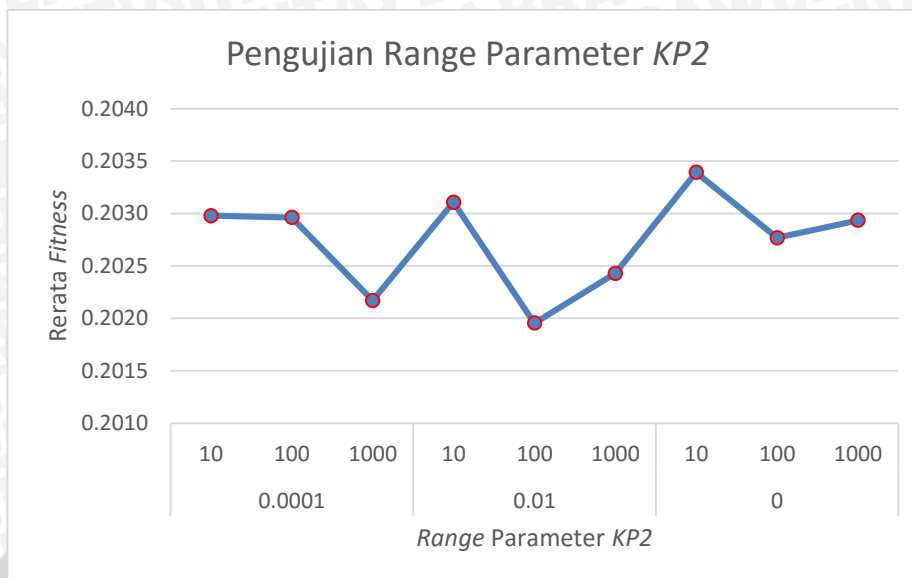
- a. Iterasi SVR : 1000
- b. Generasi : 155
- c. *PopSize* : 50
- d. *Crossover Rate* : 0.4
- e. *Mutation Rate* : 0.6
- f. Range parameter *KTi* : 2
- g. Range parameter *KP1* : 0 - 100
- h. Range parameter λ : 0.0001 - 10
- i. Range parameter γ : 0 - 1000
- j. Range parameter *C* : 0.01 - 10

Hasil uji coba *range* parameter *KP2* ditunjukkan pada Tabel 6.7.

Tabel 6.7 Hasil Uji Coba *Range* Parameter *KP2*

Kombinasi		Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
Min	Max	1	2	3	4	5	6	7	8	9	10	
0.0001	10	0.2012	0.2014	0.2014	0.2095	0.2014	0.2064	0.2045	0.2014	0.2014	0.2014	0.2030
	100	0.2061	0.2064	0.2013	0.2014	0.2014	0.2014	0.2014	0.2026	0.2014	0.2064	0.2030
	1000	0.2013	0.2014	0.2013	0.2045	0.2014	0.2014	0.2014	0.2013	0.2064	0.2014	0.2022
0.01	10	0.2013	0.2014	0.2038	0.2014	0.2014	0.2014	0.2064	0.2064	0.2014	0.2064	0.2031
	100	0.2063	0.2014	0.2014	0.2022	0.2014	0.2014	0.2014	0.2014	0.2014	0.2013	0.2020
	1000	0.2014	0.2013	0.2013	0.2063	0.2013	0.2014	0.2014	0.2053	0.2014	0.2033	0.2024
0	10	0.2026	0.2043	0.2018	0.2047	0.2014	0.2064	0.2038	0.2012	0.2064	0.2014	0.2034
	100	0.2014	0.2013	0.2144	0.2012	0.2014	0.2023	0.2014	0.2014	0.2018	0.2011	0.2028
	1000	0.2064	0.2017	0.2014	0.2011	0.2012	0.2059	0.2025	0.2064	0.2014	0.2014	0.2029

Berdasarkan grafik hasil pengujian pada Gambar 6.8 dengan data *training* dan data *testing* yang bervariasi, didapatkan nilai paling optimal untuk peramalan pada *range* 0-10. Parameter *KP2* memberikan pengaruh terhadap polynomial kernel sebagai degree dan terhadap RBF kernel sebagai σ . Degree pada polynomial dapat meningkatkan fleksibilitas pada batas deviasi yang berujung pada nilai *error rate* yang dihasilkan. Sedangkan σ menentukan kedekatan antara 2 titik sehingga dapat menentukan tingkat konsistensi terhadap batas deviasi. Nilai degree dan σ yang terlalu besar bias mengakibatkan sulitnya menemukan tingkat konsistensi terhadap batas deviasi.



Gambar 6.8 Grafik Uji Range Parameter $KP2$

6.8 Hasil dan Analisa Uji Coba Range Parameter λ

Untuk pengujian pada nilai *range* parameter yang digunakan berdasarkan pada Tabel 4.8. Detail parameter yang digunakan adalah sebagai berikut :

- a. Iterasi SVR : 1000
- b. Generasi : 155
- c. *PopSize* : 50
- d. *Crossover Rate* : 0.4
- e. *Mutation Rate* : 0.6
- f. Range parameter KTi : 0 - 2
- g. Range parameter $KP1$: 0 - 100
- h. Range parameter $KP2$: 0 - 100
- i. Range parameter γ : 0 - 1000
- j. Range parameter C : 0.01 - 10

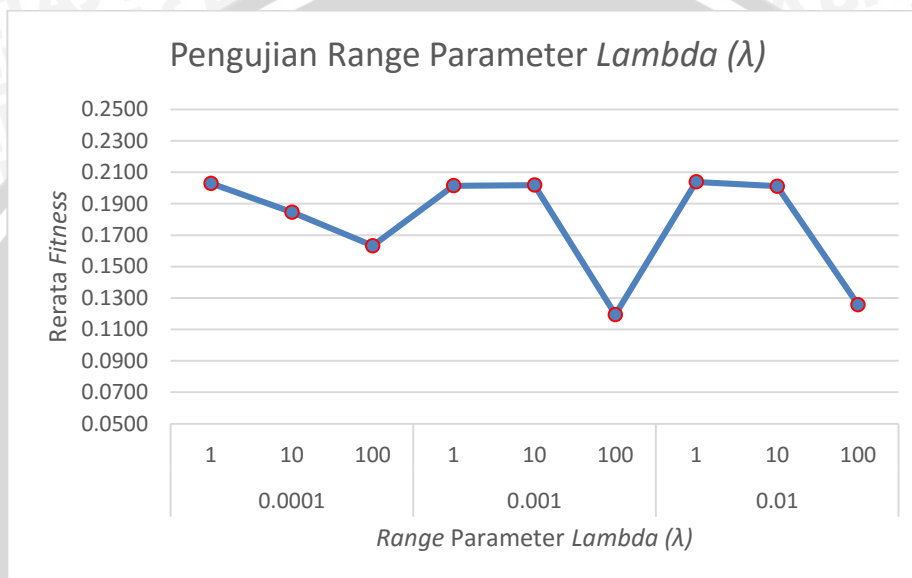
Hasil uji coba *range* parameter λ ditunjukkan pada Tabel 6.8.

Tabel 6.8 Hasil Uji Coba Range Parameter λ

Kombinasi		Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
Min	Max	1	2	3	4	5	6	7	8	9	10	
0.0001	1	0.2064	0.2014	0.2014	0.2014	0.2026	0.2014	0.2014	0.2064	0.2064	0.2011	0.2030
	10	0.2014	0.2014	0.0281	0.2014	0.2013	0.2073	0.2014	0.2014	0.2014	0.2014	0.1846
	100	0.2014	0.2013	0.2014	0.2014	0.0108	0.2014	0.2014	0.2014	0.0108	0.2014	0.1633
0.001	1	0.2014	0.2013	0.2013	0.2014	0.2013	0.2013	0.2014	0.2018	0.2014	0.2014	0.2014
	10	0.2014	0.2014	0.2064	0.2014	0.2014	0.2014	0.2013	0.2014	0.2014	0.2013	0.2019
	100	0.0108	0.2014	0.2014	0.0108	0.2014	0.0108	0.1441	0.2014	0.2014	0.0108	0.1194
0.01	1	0.2014	0.2146	0.2013	0.2014	0.2014	0.2032	0.2014	0.2064	0.2014	0.2064	0.2039

10	0.2014	0.2020	0.2064	0.2007	0.2014	0.2014	0.1911	0.2044	0.2014	0.2014	0.2011
100	0.2014	0.2014	0.2074	0.0108	0.2014	0.2014	0.0108	0.0108	0.2014	0.0108	0.1257

Berdasarkan grafik hasil pengujian pada Gambar 6.9 dengan data *training* dan data *testing* yang bervariasi, didapatkan nilai paling optimal untuk peramalan pada *range* 0.01 - 1. Semakin besar nilai dari λ akan memberikan nilai evaluasi yang baik pada regresi, namun waktu komputasi yang dibutuhkan akan lebih lama karena lambatnya kecepatan konvergensi dan proses *learning* yang kurang stabil. Hal ini dikarenakan pengaruh nilai λ dapat meningkatkan *input space* sehingga tidak akan terjadi konvergensi dini dan menjaga proses *learning* agar tetap stabil.



Gambar 6.9 Grafik Uji Range Parameter λ

6.9 Hasil dan Analisa Uji Coba Range Parameter γ

Untuk pengujian pada nilai *range* parameter yang digunakan berdasarkan pada Tabel 4.9. Detail parameter yang digunakan adalah sebagai berikut :

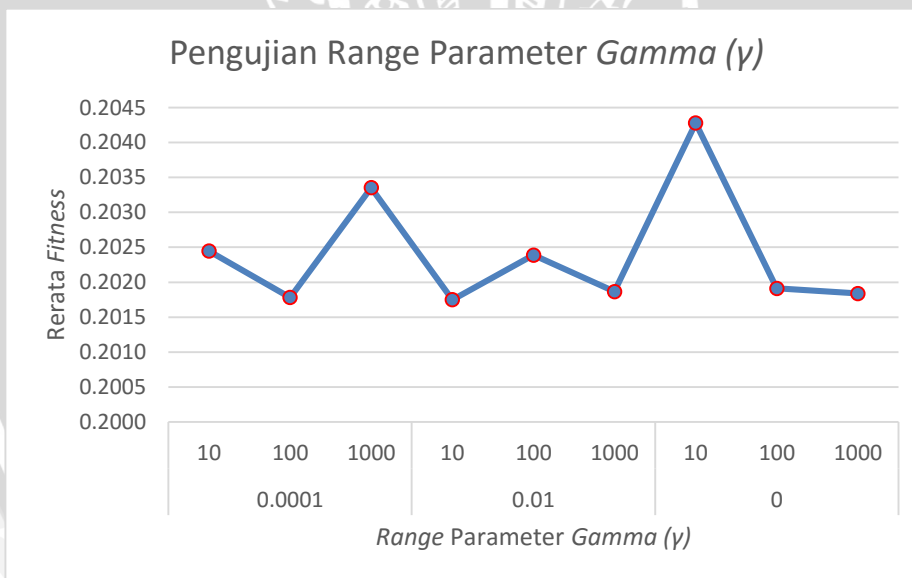
- a. Iterasi SVR : 1000
- b. Generasi : 155
- c. *PopSize* : 50
- d. *Crossover Rate* : 0.4
- e. *Mutation Rate* : 0.6
- f. Range parameter *KTi* : 0 - 2
- g. Range parameter *KP1* : 0 - 100
- h. Range parameter *KP2* : 0 - 100
- i. Range parameter λ : 0.0001 - 10
- j. Range parameter *C* : 0.01 - 10

Hasil uji coba *range* parameter γ ditunjukkan pada Tabel 6.9.

Tabel 6.9 Hasil Uji Coba Range Parameter γ

Kombinasi		Nilai Fitness Percobaan ke- <i>i</i>										Rata – rata Fitness
Min	Max	1	2	3	4	5	6	7	8	9	10	
0.0001	10	0.2012	0.2014	0.2014	0.2027	0.2061	0.2063	0.2014	0.2014	0.2014	0.2012	0.2024
	100	0.2014	0.2012	0.2014	0.2027	0.2031	0.2014	0.2024	0.2014	0.2014	0.2014	0.2018
	1000	0.2014	0.2064	0.2013	0.2013	0.2014	0.2063	0.2064	0.2064	0.2014	0.2014	0.2034
0.01	10	0.2013	0.2014	0.2013	0.2014	0.2025	0.2014	0.2024	0.2014	0.2032	0.2014	0.2017
	100	0.2014	0.2064	0.2023	0.2014	0.2014	0.2014	0.2014	0.2014	0.2055	0.2014	0.2024
	1000	0.2064	0.2014	0.2014	0.2014	0.2014	0.2014	0.2014	0.2014	0.2013	0.2013	0.2019
0	10	0.2120	0.2014	0.2064	0.2064	0.2014	0.2012	0.2014	0.2014	0.2050	0.2064	0.2043
	100	0.2014	0.2025	0.2014	0.2013	0.2013	0.2014	0.2025	0.1997	0.2064	0.2014	0.2019
	1000	0.2012	0.2012	0.2014	0.2006	0.2014	0.2060	0.2025	0.2014	0.2014	0.2014	0.2018

Berdasarkan grafik hasil pengujian pada Gambar 6.10 dengan data *training* dan data *testing* yang bervariasi, didapatkan nilai paling optimal untuk peramalan pada *range* 0 - 10. Semakin kecil nilai γ , maka proses *learning* akan berlangsung lebih lama namun hasil yang didapat lebih konvergen. Hal ini dikarenakan pengaruh nilai γ yang mengontrol kecepatan proses *learning* dalam mencapai hasil yang konvergen.



Gambar 6.10 Grafik Uji Range Parameter γ

6.10 Hasil dan Analisa Uji Coba Range Parameter C

Untuk pengujian pada nilai *range* parameter yang digunakan berdasarkan pada Tabel 4.10. Detail parameter yang digunakan adalah sebagai berikut :

- a. Iterasi SVR : 1000
- b. Generasi : 155
- c. PopSize : 50

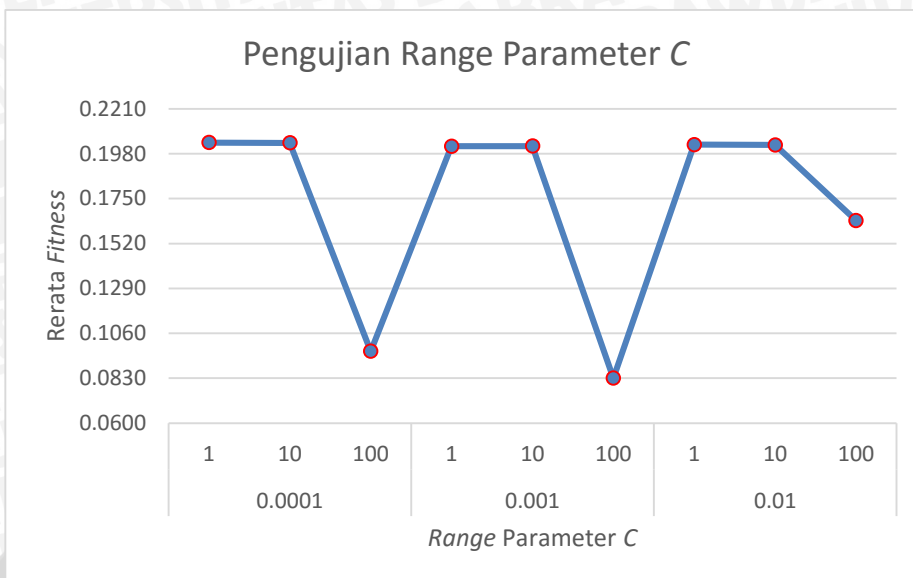
- d. *Crossover Rate* : 0.4
- e. *Mutation Rate* : 0.6
- f. Range parameter *KT_i* : 0 - 2
- g. Range parameter *KP₁* : 0 - 100
- h. Range parameter *KP₂* : 0 - 100
- i. Range parameter λ : 0.0001 - 10
- j. Range parameter γ : 0 - 1000

Hasil uji coba *range* parameter *C* ditunjukkan pada Tabel 6.10.

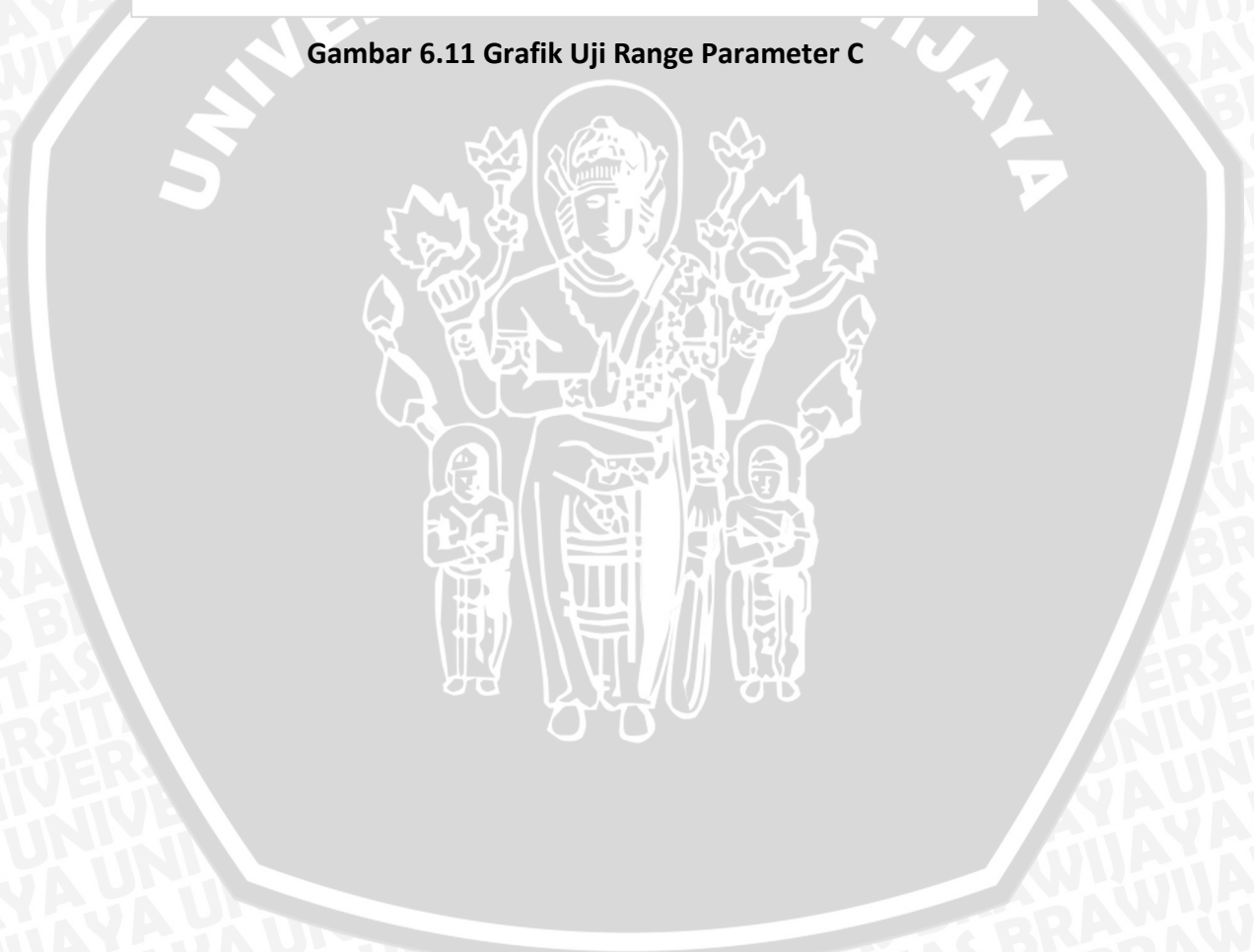
Tabel 6.10 Hasil Uji Coba *Range* Parameter *C*

Kombinasi		Nilai <i>Fitness</i> Percobaan ke- <i>i</i>										Rata – rata <i>Fitness</i>
Min	Max	1	2	3	4	5	6	7	8	9	10	
0.0001	1	0.2086	0.2013	0.1994	0.2061	0.2023	0.2061	0.2037	0.2043	0.2043	0.2012	0.2038
	10	0.2013	0.2013	0.2014	0.2157	0.2014	0.2014	0.2039	0.2013	0.2014	0.2063	0.2035
	100	0.2044	0.2014	0.0922	0.2060	0.0108	0.0108	0.0155	0.2049	0.0108	0.0108	0.0968
0.001	1	0.2025	0.2052	0.2014	0.2013	0.2013	0.2014	0.2014	0.2014	0.2013	0.2014	0.2018
	10	0.2014	0.2014	0.2014	0.2023	0.2014	0.2013	0.2064	0.2013	0.2013	0.2014	0.2019
	100	0.0108	0.0108	0.0108	0.2013	0.0108	0.2014	0.0108	0.2012	0.0108	0.1614	0.0830
0.01	1	0.2014	0.2064	0.2022	0.2014	0.2024	0.2014	0.2013	0.2012	0.2023	0.2064	0.2026
	10	0.2014	0.2013	0.2014	0.2014	0.2014	0.2064	0.2014	0.2014	0.2074	0.2013	0.2025
	100	0.2042	0.2014	0.2042	0.2014	0.2013	0.2014	0.0108	0.2014	0.0108	0.2013	0.1638

Berdasarkan grafik hasil pengujian pada Gambar 6.11 dengan data *training* dan data *testing* yang bervariasi, didapatkan nilai paling optimal untuk peramalan pada *range* 0.0001 - 1. Nilai *C* yang ada pada *range* tersebut terbukti menghasilkan nilai paling optimal dibandingkan dengan batas lainnya. Konstanta *C* dijadikan sebagai variabel yang menampung nilai pelanggaran toleransi berupa batas atas deviasi terhadap fungsi regresi, semakin tinggi nilai *C* yang didapat memberikan indikasi bahwa semakin jauh garis regresi yang dibentuk terhadap batas deviasi yang ada, sedangkan semakin kecil nilai *C* yang didapat menandakan garis regresi yang terbentuk semakin mendekati dengan batas deviasi yang ada.



Gambar 6.11 Grafik Uji Range Parameter C



BAB 7 PENUTUP

Bab ini membahas kesimpulan dan saran terkait penelitian peramalan jumlah konsumsi BBM menggunakan metode *Hybrid* Algoritma Genetika dan *Support Vector Regression*.

7.1 Kesimpulan

Berikut adalah kesimpulan yang dapat diambil dari hasil yang telah didapatkan dari perancangan, implementasi dan pengujian yang telah dilakukan

1. Dalam penelitian ini terdapat 6 parameter yang dioptimasi menggunakan HGA diantaranya yaitu tipe kernel, nilai parameter kernel ke-1, nilai parameter kernel ke-2, nilai lambda, nilai gamma, dan nilai kompleksitas. Panjang data yang digunakan dalam SVR adalah sebanyak 45 data yang berisi data harga BBM dalam rentang waktu pertahun. Nilai optimal dari parameter tersebut akan didapatkan dari proses optimasi dengan metode *Hybrid* Algoritma Genetika. Selanjutnya nilai parameter yang dihasilkan akan digunakan untuk parameter metode *Support Vector Regression* sehingga akan dihasilkan nilai evaluasi peramalan.
2. Koefisien yang dihasilkan oleh sistem menghasilkan nilai error yang cukup rendah dan pada sistem ini algoritma HGA ini dapat memberikan solusi optimal dalam mengoptimasi peramalan harga BBM menggunakan metode *Support Vector Regression* yaitu dengan menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) sebesar 2.8737 pada iterasi SVR sebesar 50 dengan parameter diantaranya $KT=0$ yang merupakan linear kernel, $KP1=86.5562$, $KP2=31.8421$, $\lambda=1.8414$, $\gamma=73.3125$, $C=3.0701$.

7.2 Saran

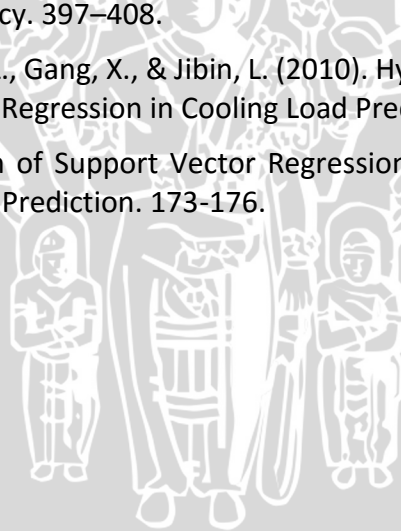
Berdasarkan kesimpulan diatas, ada beberapa *point* yang perlu diperhatikan jika pembaca ingin melakukan penelitian lebih lanjut.

1. Pada penelitian berikutnya peneliti bisa menambahkan tipe fungsi kernel seperti kernel *laplacian*, kernel *multiquadratic*, atau kernel *sigmoid*. Hal ini bertujuan agar proses memetakan *input space* dan *feature space* bias lebih bervariasi, sehingga dapat menghasilkan nilai peramalan dengan nilai *error rate* yang kecil.
2. Penelitian yang selanjutnya, peneliti bisa menggunakan metode SVR untuk *multitarget forecasting*. Hal ini bertujuan untuk memudahkan peneliti dalam meramalkan untuk beberapa waktu sekaligus.

DAFTAR PUSTAKA

- Alves, G. I., Pereira, E. J., Silva, D. A., & Ferreira, T. A. (2013). Data Envelopment Analysis for Selection of the Fitness Function in Evolutionary Algorithms Applied to Time Series Forecasting Problem.
- Alwee, R., Shamsuddin, S. M., & 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. 2013.
- Amelia, Regasari, R., & Priharsari, D. (2014). Implementasi Hybrid Algoritma Genetika Untuk Penjadwalan Auditor Pada Audit Internal Mutu Universitas Brawijaya. 4.
- Bhaskar, M. M., & Maheswarapu, S. (2011). A Hybrid Genetic Algorithm Approach for Optimal Power Flow. 9(1693-6930), 211-216.
- Desvina, A. P., & Siddiq, A. R. (2011). Pendekatan Peramalan Tingkat Konsumsi Minyak Di Indonesia Dengan Menggunakan Metode Box-Jenkins. 9.
- Farizal, Rachman, A., & Rasyid, H. A. (2014). Model Peramalan Konsumsi Bahan Bakar Jenis Premium Di Indonesia Dengan Regresi Linier Berganda. 13(1412-6869).
- J., A. R., Neto, P. S., & Ferreira, T. A. (2009). A Prime Step in The Time Series Forecasting With Hybrid Methods: The Fitness Function Choice.
- Kusumaningrum, O., & Suhartono, H. (2012). Peramalan Kebutuhan Bahan Bakar Premium di Depot Ampean dengan Metode Hibrida Arima-Neural Network untuk Optimasi Persediaan. 1(2301-928X).
- Mahmudy, W. (2013). *Algoritma Evousi*.
- Mahmudy, W. F. (2006). Penerapan Algoritma Genetika Pada Optimasi Model Penugasan. 10.
- Mahmudy, W. F. (2008). Optimasi Fungsi tanpa Kendala Menggunakan Algoritma dengan Kromosom biner dan perbaikan Kromosom hill-climbing. 4.
- Mahmudy, W. F., Marian, R. M., & Luong, L. H. (2014). Hybrid Genetic Algorithms for Part Type Selection and Machine Loading Problems with Alternative Production Plans in Flexible Manufacturing System. 8.
- Makridakis, S., & Hibon, M. (1995). Evaluating Accuracy (or Error) Measure.
- Pasolli, L., Notarnicola, C., & Bruzzone, L. (2012). Multi-Objective Parameter Optimization in Support Vector Regression: General Formulation and Application to the Retrieval of Soil Moisture From Remote Sensing Data. 5.

- Rahmi, A., Mahmudy, W. F., & Setiawan, B. D. (2015). Prediksi Harga Saham Berdasarkan Data Historis Menggunakan Model Regresi yang Dibangun Dengan Algoritma Genetik. 5.
- Sari, D. P. (2009). Analisis Performansi Support Vector Regression dalam Memprediksi Bonus Tahunan Karyawan. *IV*.
- Sempena, S. (2010). Algoritma Genetik Hibrida dalam Penyelesaian Job-shop Scheduling Problem.
- Sunoto. (2006). Analisis Kebijakan Subsidi Bagi Nelayan Akibat Kenaikan Harga Bahan Bakar Minyak (Bbm). *XVII(1410-8828)*, 57-66.
- Vijayakumar, S., & Wu, S. (1999). Sequential Support Vector Classifiers and Regression. 610-619.
- Wu, C.-H., Tzeng, G.-H., & Lin, R.-H. (2009). A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression.
- Wu, C.-H., Tzeng, G.-H., & Lin, R.-H. (2009). A Novel Hybrid Genetiz Algorithm For Kernel Function And Parameter Optimization in Support Vector Regression.
- Wu, C.-H., Tzeng, G.-H., Goo, Y.-J., & Fang, W.-C. (2007). A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. 397–408.
- Xuemei, L., Lixing, D., Yan, L., Gang, X., & Jibin, L. (2010). Hybrid Genetic Algorithm and Support Vector Regression in Cooling Load Prediction.
- Ying, E. (2012). Application of Support Vector Regression Algorithm in Colleges Recruiting Students Prediction. 173-176.



LAMPIRAN A DATA KONSUMSI BBM PERIODE 1965-2014

A.1 Data Konsumsi BBM Periode 1965-2014

Tahun	Jumlah Konsumsi BBM	Tahun	Jumlah Konsumsi BBM
1965	122.4476712	1991	692.1147945
1966	117.9256986	1992	745.0984153
1967	114.7659178	1993	785.8611233
1968	121.1098361	1994	808.9757534
1969	131.1056438	1995	864.6745753
1970	138.2172603	1996	924.8253279
1971	142.9839452	1997	1026.491863
1972	154.8656831	1998	980.7497808
1973	185.0575342	1999	1024.682849
1974	193.982411	2000	1136.86376
1975	221.9059726	2001	1152.913425
1976	238.0966667	2002	1204.439038
1977	287.407589	2003	1222.237686
1978	326.0261096	2004	1299.186962
1979	354.9598082	2005	1284.890503
1980	396.4958197	2006	1247.184651
1981	441.5163836	2007	1298.935538
1982	456.6725479	2008	1293.616438
1983	448.4044932	2009	1333.74248
1984	477.3640164	2010	1457.756157
1985	467.9943014	2011	1566.807712
1986	494.1211233	2012	1599.388411
1987	511.9309041	2013	1614.96582
1988	537.0831967	2014	1640.990397
1989	570.7151781		
1990	653.2235068		

Data yang ada pada tabel diatas merupakan data konsumsi BBM di Indonesia periode 1965-2014 dengan data pertahun.

LAMPIRAN B VISUALISASI HASIL PENGUJIAN

B.1 Nilai Parameter Optimal untuk Peramalan Konsumsi Bahan Bakar Minyak di Indonesia Dengan Metode *Hybrid Algen - SVR*

Hasil berikut merupakan nilai yang optimal untuk masing-masing variabel yang ada pada metode *Hybrid Algen* dan metode SVR setelah beberapa uji coba yang dilakukan:

1. Batas pencarian parameter KTi : 0-2
2. Batas pencarian parameter $KP1$: 0.0001 – 10
3. Batas pencarian parameter $KP2$: 0 - 10
4. Batas pencarian parameter λ : 0.01 - 1
5. Batas pencarian parameter γ : 0 – 10
6. Batas pencarian parameter C : 0.0001 – 1
7. Jumlah iterasi SVR : 100000
8. Jumlah generasi : 140
9. Jumlah popSize : 70
10. *Crossover rate* : 0.4
11. *Mutation rate* : 0.6
12. Jumlah periode peramalan : 11
13. Nilai KTi : 0
14. Nilai $KP1$: 9.540703036304205
15. Nilai $KP2$: 7.67495716407897
16. Nilai λ : 0.0416260590237002
17. Nilai γ : 5.99395504812782
18. Nilai C : 0.3615393392147428

Hasil prakiraan menggunakan parameter diatas menghasilkan nilai MAPE sebesar 3.708677.

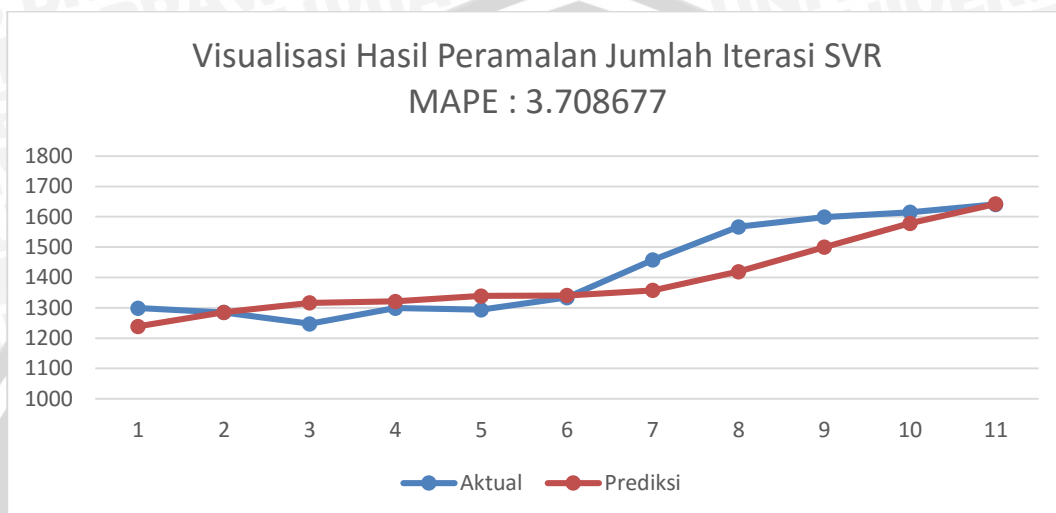
B.2 Nilai α^* dan α Optimal untuk Fungsi Regresi pada SVR

Hasil berikut merupakan nilai α^* dan α yang optimal untuk peramalan SVR yang didapatkan pada iterasi terakhir yaitu 1000000:

NO	α^*	α	NO	α^*	α	NO	α^*	α
1	0	0.361539	13	0.361539	0	25	0.361539	0
2	0	0.361539	14	0.361539	0	26	0.361539	0
3	0	0.361539	15	0.361539	0	27	0.361539	0
4	0	0.361539	16	0.361539	0	28	0.361539	0
5	0	0.361539	17	0.361539	0	29	0.361539	0
6	0.237392	0.124136	18	0.361539	0	30	0.361539	0
7	0.361539	0	19	0.361539	0	31	0.361539	0
8	0.361539	0	20	0.361539	0	32	0.361539	0

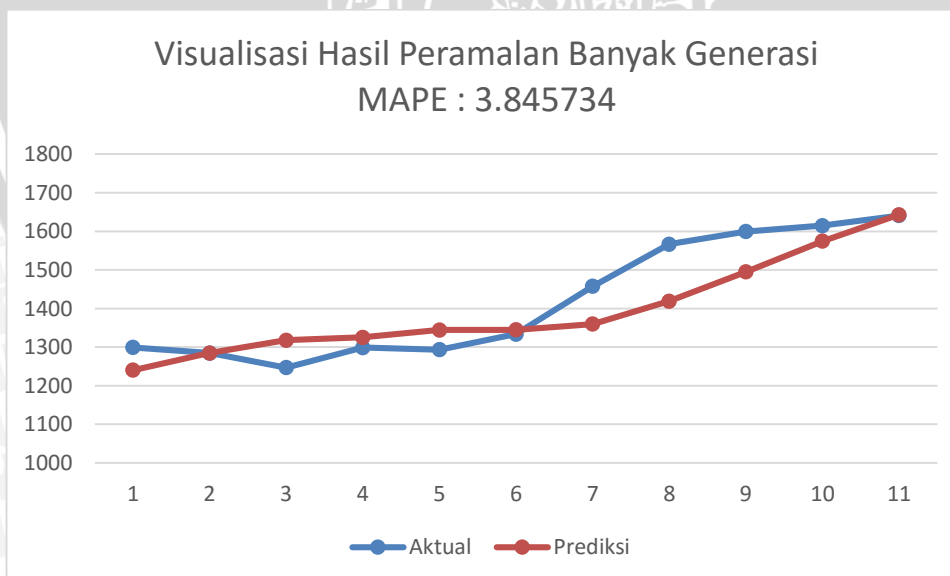
9	0.361539	0	21	0.361539	0	33	0.361539	0
10	0.361539	0	22	0.361539	0	34	0.361539	0
11	0.361539	0	23	0.361539	0			
12	0.361539	0	24	0.361539	0			

B.3 Visualisasi Hasil Uji Coba Jumlah Iterasi SVR



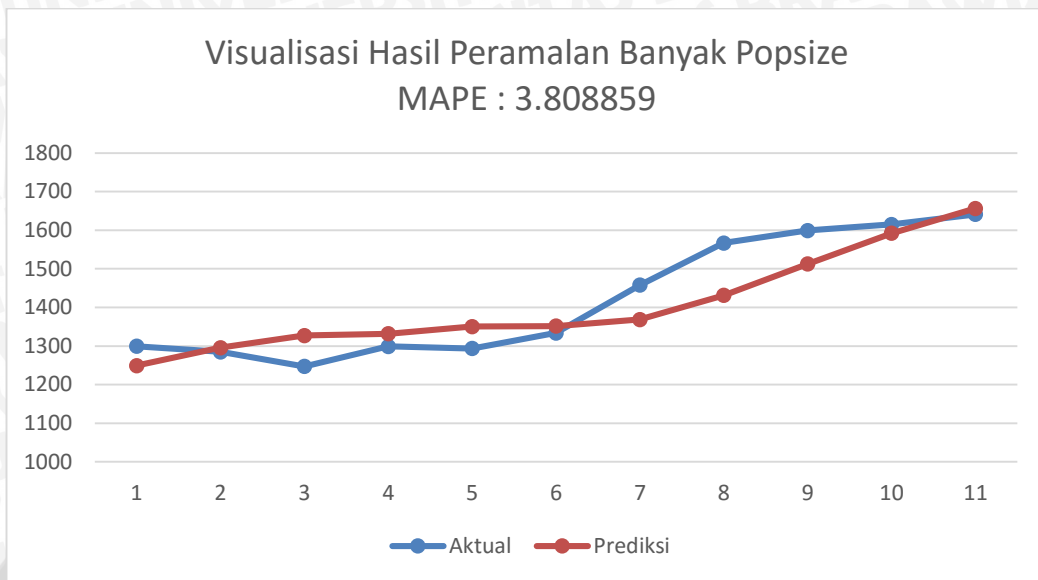
Grafik diatas merupakan visualisasi hasil peramalan jumlah iterasi SVR yang terbaik dari 9 variasi dan 10 kali percobaan dengan MAPE sebesar 3.643668, yaitu menggunakan iterasi SVR sebesar 100000.

B.4 Visualisasi Hasil Uji Coba Banyak Generasi



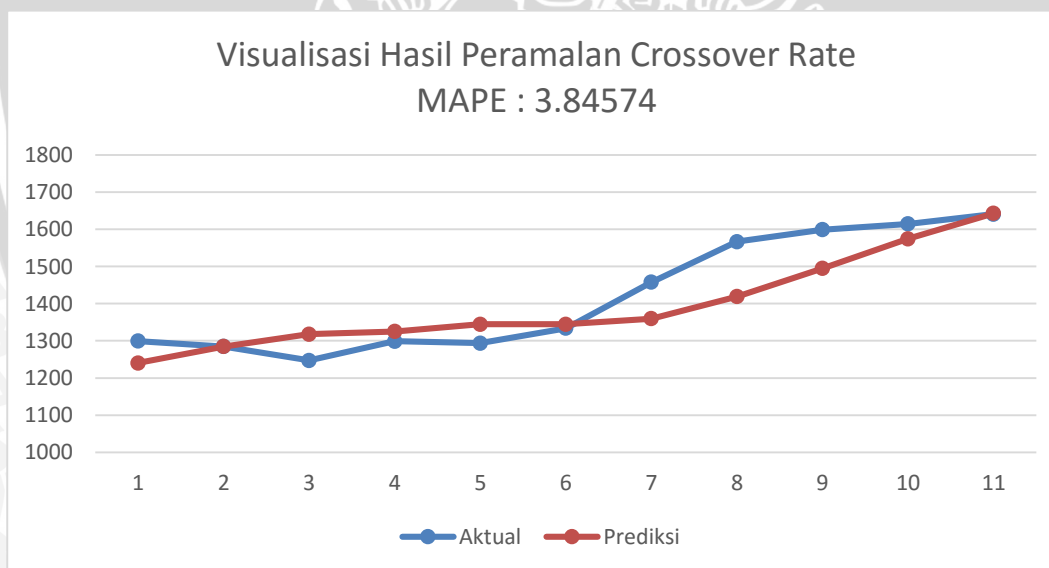
Grafik diatas merupakan visualisasi hasil peramalan banyak generasi yang terbaik dari 11 variasi dan 10 kali percobaan dengan MAPE sebesar 3.845734, yaitu menggunakan generasi sebanyak 140.

B.5 Visualisasi Hasil Uji Coba Ukuran Populasi



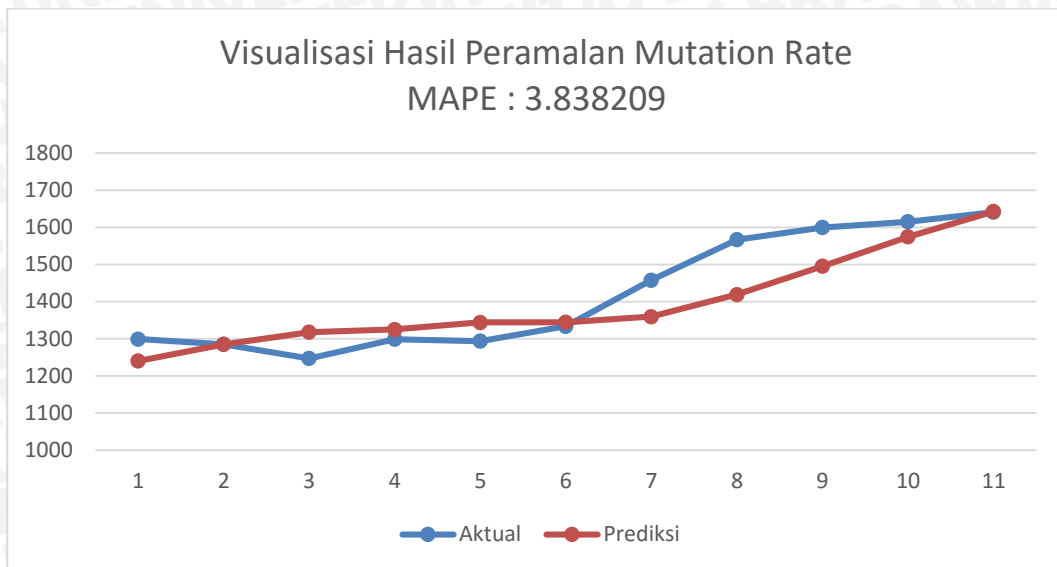
Grafik diatas merupakan visualisasi hasil peramalan banyak *popsize* yang terbaik dari 11 variasi dan 10 kali percobaan dengan MAPE sebesar 3.808859, yaitu menggunakan *popsize* sebanyak 70.

B.6 Visualisasi Hasil Uji Coba Crossover Rate



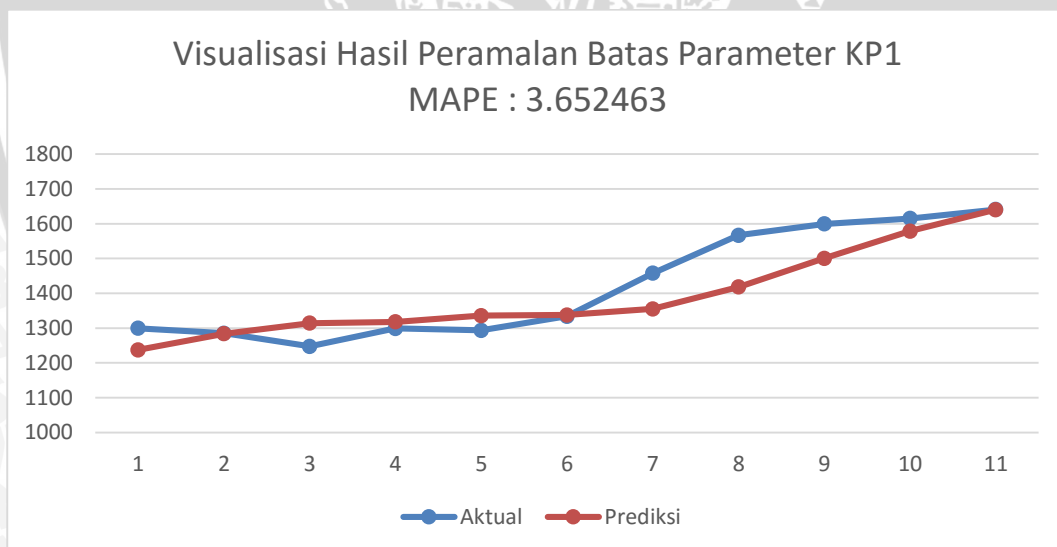
Grafik diatas merupakan visualisasi hasil peramalan *crossover rate* yang terbaik dari 11 variasi dan 10 kali percobaan dengan MAPE sebesar 3.84574, yaitu menggunakan *crossover rate* sebesar 0.4.

B.7 Visualisasi Hasil Uji Coba Mutation Rate



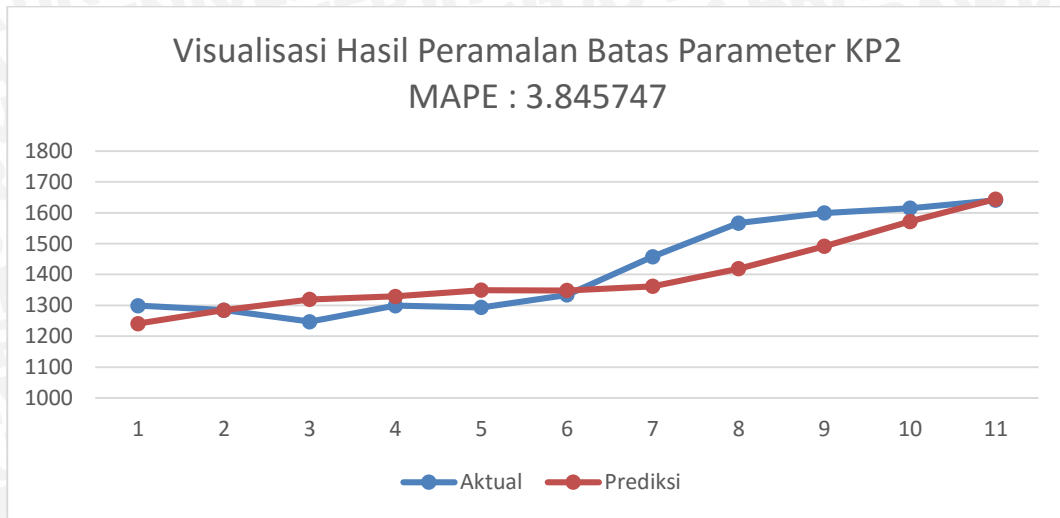
Grafik diatas merupakan visualisasi hasil peramalan *mutation rate* yang terbaik dari 11 variasi dan 10 kali percobaan dengan MAPE sebesar 3.838209, yaitu menggunakan *mutation rate* sebesar 0.6.

B.8 Visualisasi Hasil Uji Coba Range Parameter KP1



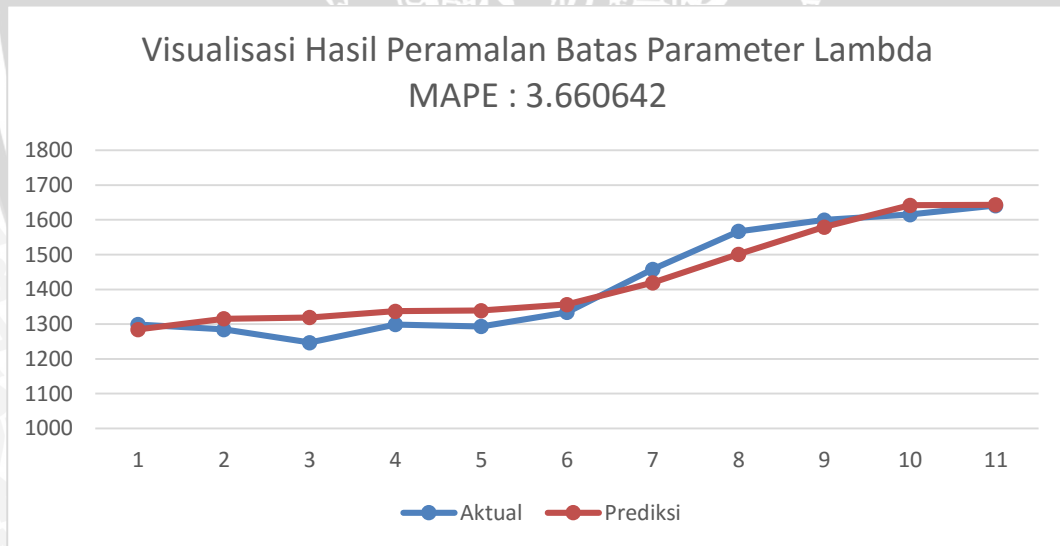
Grafik diatas merupakan visualisasi hasil peramalan range parameter *KP1* yang terbaik dari 9 variasi dan 10 kali percobaan dengan MAPE sebesar 3.652463, yaitu menggunakan batas atas *KP1* sebesar 0.0001 dan batas atas sebesar 10.

B.9 Visualisasi Hasil Uji Coba Range Parameter KP2



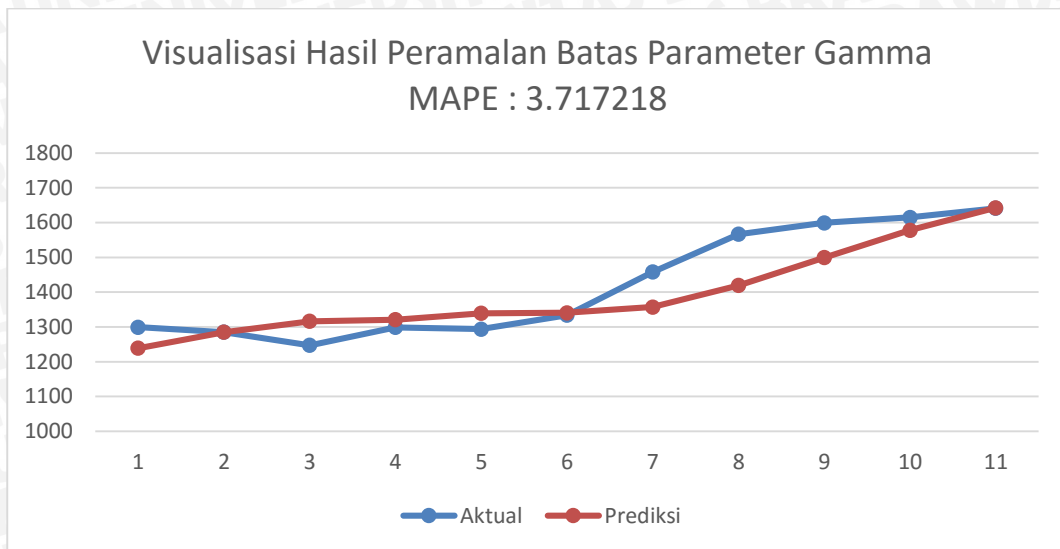
Grafik diatas merupakan visualisasi hasil peramalan range parameter $KP2$ yang terbaik dari 9 variasi dan 10 kali percobaan dengan MAPE sebesar 3.845747, yaitu menggunakan batas atas $KP2$ sebesar 0 dan batas atas sebesar 10.

B.10 Visualisasi Hasil Uji Coba Range Parameter Lambda (λ)



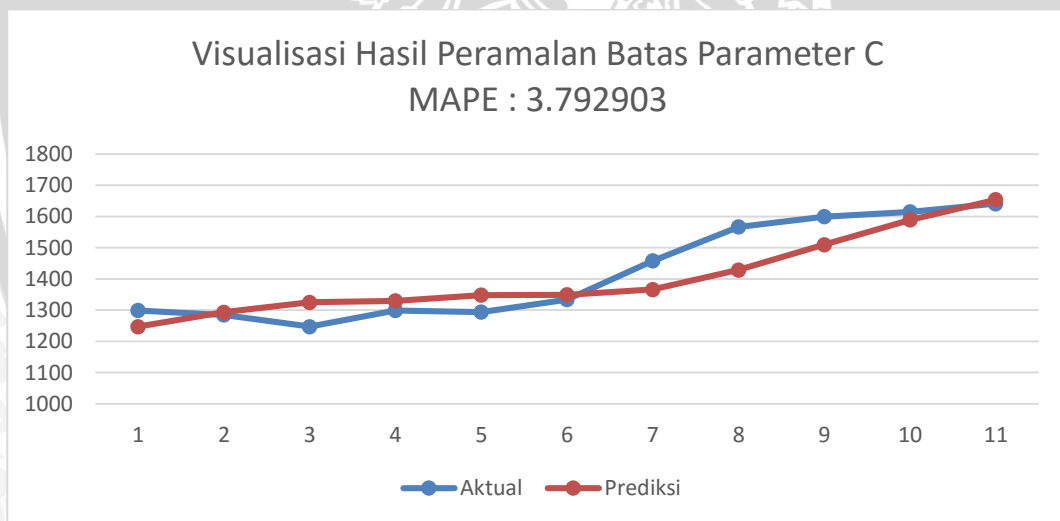
Grafik diatas merupakan visualisasi hasil peramalan batas parameter λ yang terbaik dari 9 variasi dan 10 kali percobaan dengan MAPE sebesar 3.660642, yaitu menggunakan batas atas λ sebesar 0.01 dan batas atas sebesar 1.

B.11 Visualisasi Hasil Uji Coba *Range* Parameter γ



Grafik diatas merupakan visualisasi hasil peramalan batas parameter γ yang terbaik dari 9 variasi dan 10 kali percobaan dengan MAPE sebesar 3.717218, yaitu menggunakan batas atas γ sebesar 0 dan batas atas sebesar 10.

B.12 Visualisasi Hasil Uji Coba *Range* Parameter C



Grafik diatas merupakan visualisasi hasil peramalan batas parameter C yang terbaik dari 9 variasi dan 10 kali percobaan dengan MAPE sebesar 3.792903, yaitu menggunakan batas atas C sebesar 0.0001 dan batas atas sebesar 1.