

**OPTIMASI FUZZY TIME SERIES UNTUK PREDIKSI JUMLAH
PRODUKSI SAGA LEATHER FASHION MENGGUNAKAN
METODE ALGORITME GENETIKA**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Aditya Chandra Nurhakim

NIM: 135150207111034



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

OPTIMASI FUZZY TIME SERIES UNTUK PREDIKSI JUMLAH PRODUKSI SAGA
LEATHER FASHION MENGGUNAKAN METODE ALGORITME GENETIKA

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Aditya Chandra Nurhakim
NIM: 135150207111034

Skripsi ini telah diuji dan dinyatakan lulus pada
15 April 2019
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2

Budi Darma Setiawan, S.Kom, M.Cs
NIP: 19841015 201404 1 002

Candra Dewi, S.Kom, M.Sc
NIP: 19771114 200312 2 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

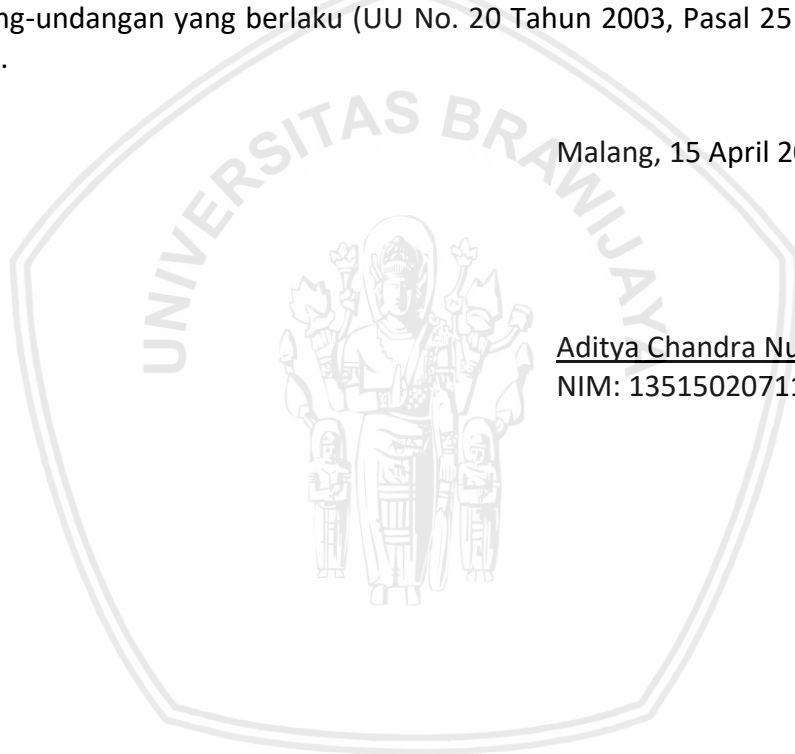
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan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

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, 15 April 2019

Aditya Chandra Nurhakim

NIM: 135150207111034



PRAKATA

Dengan mengucapkan syukur kehadiran Allah SWT, atas rahmat dan karunia-Nya. Sehingga penulis dapat menyelesaikan skripsi ini yang berjudul “Optimasi Fuzzy Time Series Untuk Prediksi Jumlah Produksi SAGA Leather Fashion Menggunakan Metode Algoritme Genetika”. Tujuan penulisan skripsi ini, untuk memenuhi sebagian persyaratan guna memperoleh gelar Sarjana Komputer pada Fakultas Ilmu Komputer Universitas Brawijawa.

Penyelesaian skripsi ini tidak terlepas dari bantuan berbagai pihak, sehingga pada kesempatan ini penulis ingin mengucapkan terima kasih dari hati yang terdalam untuk bantuan moril dan materil yang telah diberikan kepada penulis, terutama kepada yang saya hormati:

1. Prof. Dr. Ir. Nuhfil Hanani AR., MS. selaku Rektor Universitas Brawijaya
2. Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer
3. Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika
4. Budi Darma Setiawan, S.Kom, M.Cs dan Candra Dewi, S.Kom, M.Sc. selaku pembimbing penulis yang selalu mengarahkan dengan sabar dalam menyelesaikan skripsi ini.
5. Teristimewa Orang Tua dan adik-adik penulis yang selalu mendoakan, mendukung dan selalu percaya sehingga penulis dapat menyelesaikan skripsi ini. Serta yang teristimewa Novirra Dwi Asri yang selalu setia memberikan motivasi dan arahan untuk penulis.
6. Teman-teman penulis yang berada di Bali, Filkom maupun di Malang. Penulis ingin mengucapkan terima kasih yang sebesar-besarnya atas bantuan yang telah diberikan kepada penulis.

Akhir kata penulis menyadari skripsi ini masih memiliki banyak kekurangan. Oleh karena itu, penulis memohon saran dan kritik untuk membantu dalam menyempurnakan skripsi ini.

Malang, 16 Maret 2019

Penulis

adit.candra00@gmail.com

ABSTRAK

Aditya Chandra Nurhakim, Optimasi Fuzzy Time Series Untuk Prediksi Jumlah Produksi SAGA Leather Fashion Menggunakan Metode Algoritme Genetika.

Pembimbing: Budi Darma Setiawan, S.Kom, M.Cs dan Candra Dewi, S.Kom, M.Sc.

Fashion sudah menjadi salah satu hal yang menjadi perhatian bagi setiap golongan masyarakat. Kulit hewan (*leather*) merupakan bahan baku produk *fashion* yang sangat diminati saat ini. Berbeda dengan produk berbahan dasar kain yang tidak tahan lama, bahan kulit dapat bertahan pada kurun waktu 10-15 tahun lamanya. SAGA *leather fashion* yang telah berdiri dari tahun 1999, telah memproduksi lebih dari 1000 produk kulit seperti jaket, tas, dompet, sepatu, sabuk dan topi dari berbagai macam kulit seperti kulit sapi, kulit kambing dan kulit domba sesuai dengan permintaan yang ada, yang menyebabkan produksi setiap bulan berbeda-beda dan mengakibatkan tidak optimalnya sisa bahan baku (kulit) yang telah digunakan. Oleh sebab itu, dengan memprediksi jumlah barang yang akan diproduksi pada bulan berikutnya, hasil prediksi digunakan untuk memproduksi barang dengan menggunakan sisa kulit pada bulan sebelumnya. Produk yang diprediksi antara lain produk topi, sabuk, dompet, tas, jaket dan sepatu. Prediksi pada penelitian ini menggunakan metode *fuzzy time series* yang dioptimasi menggunakan metode algoritme genetika.

Hasil analisis dari penelitian ini yaitu, jumlah popsize mempengaruhi nilai *fitness* yang dihasilkan. Semakin banyak jumlah popsize yang digunakan, maka semakin tinggi kemungkinan untuk menghasilkan individu terbaik pada generasi awal dikarenakan *offspring* yang dihasilkan bervariasi untuk dipilih dalam generasi selanjutnya. Tetapi, jika jumlah popsize terlalu banyak maka waktu komputasi yang dibutuhkan pun semakin lama. Sedangkan untuk nilai kombinasi *cr* dan *mr*, menghasilkan individu terbaik ketika nilai *cr* lebih besar dari nilai *mr* dikarenakan nilai kromosom *offspring* yang dihasilkan dari proses *crossover* memiliki perbedaan yang cukup besar dengan *parentnya*, yang mempengaruhi nilai *fitness* yang dihasilkan. Pada penelitian ini menghasilkan rata-rata nilai RMSE (*Root Mean Square Error*) sebesar 4.481 dan rata-rata tingkat akurasi sebesar 92.97% dari masing-masing produk yang telah diprediksi.

Kata kunci: *fuzzy time series*, algoritme genetika, kulit, *fashion*, RMSE, akurasi

ABSTRACT

Aditya Chandra Nurhakim, Optimization Fuzzy Time Series For Forecasting Production of SAGA Leather Fashion Using Genetic Algorithm.

Supervisors: Budi Darma Setiawan, S.Kom, M.Cs and Candra Dewi, S.Kom, M.Sc.

Fashion has become one of the things that attract attention for every society. Leather is the raw material for fashion products that are in great demand today. Unlike fabric-based products that are not durable, leather can last for 10-15 years. SAGA leather fashion, which was established in 1999, has produced more than 1000 leather products such as jackets, bags, wallets, shoes, belts and hats from various types of leather such as cowhide, goat skin and sheep skin according to the request, which causes production every month varies and results in not optimal of residual raw material (skin) that has been used. Therefore, by predicting the number of items to be produced in the following month, the prediction results are used to produce goods using the remaining skin in the previous month. The products predicted include caps, belts, wallets, bags, jackets and shoes. Prediction in this study uses fuzzy time series method which is optimized using the genetic algorithm method.

The analysis of this study are, the amount of popsize affecting the fitness value produced. The more number of popsizes used, the higher the possibility of producing the best individuals in the first generation because the resulting offspring varies to be chosen in the next generation. However, if the number of popsize is too much then the computational time needed is getting longer. Whereas for the combination value of cr and mr , it produces the best individual when the cr value is greater than the value of mr because the offspring chromosome value generated from the crossover process has a considerable difference with the parent, which affects the fitness value produced. In this study the average RMSE (Root Mean Square Error) value was 4,481 and the average level of accuracy was 92.97% of each predicted product.

Keywords: *fuzzy time series, genetic algorithm, leather, fashion, RMSE, accuracy*

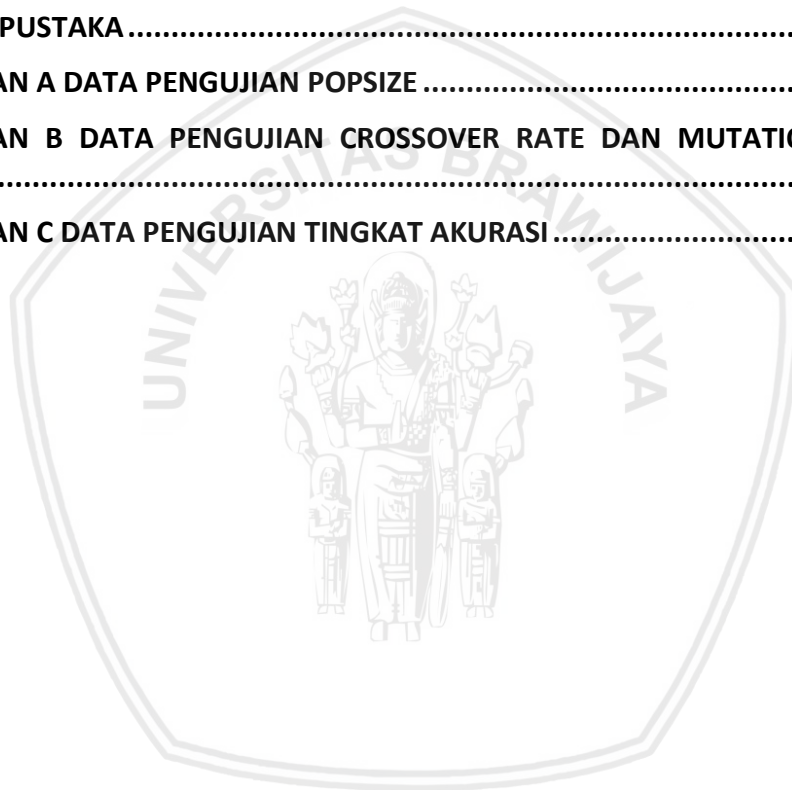
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiii
DAFTAR PERSAMAAN	xv
DAFTAR KODE PROGRAM	xvi
DAFTAR LAMPIRAN.....	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN.....	4
2.1 Kajian Pustaka	4
2.2 Dasar Teori	5
2.2.1 SAGA Leather Fashion	5
2.2.2 Prediksi	6
2.2.3 Logika Fuzzy.....	6
2.2.4 Fuzzy Time Series	6
2.2.5 Algoritme Genetika	8
2.2.6 RMSE & RMSPE	10
BAB 3 METODOLOGI	11
3.1 Tipe Penelitian	11
3.2 Pengumpulan Data	11

3.3 Implementasi	12
3.4 Peralatan Pendukung.....	13
3.5 Perancangan Pengujian	13
3.6 Analisis Hasil	14
BAB 4 ALGORITME	15
4.1 Universe of Discourse	16
4.2 Inisialisasi Populasi.....	18
4.3 Crossover	20
4.4 Mutation	23
4.5 Inisialisasi Sub-Himpunan	24
4.6 Fuzzy Time Series	25
4.7 Fuzzyfikasi	26
4.8 FLR.....	28
4.9 FLRG	29
4.10 Defuzzyfikasi	30
4.11 RMSE & Fitness	31
4.12 Seleksi	32
4.13 Perhitungan Manual	33
4.13.1 Inisialisasi Populasi.....	33
4.13.2 Crossover.....	34
4.13.3 Mutation	34
4.13.4 Inisialisasi Sub-Himpunan	35
4.13.5 Fuzzyfikasi	35
4.13.6 FLR & FLRG	36
4.13.7 Defuzzyfikasi.....	38
4.13.8 Prediksi	38
4.13.9 RMSE & Fitness	39
4.13.10 Seleksi.....	41
4.14 Perancangan Antarmuka	41
BAB 5 IMPLEMENTASI	43
5.1 Ruang Lingkup Pengujian	43
5.1.1 Perangkat Keras	43

5.1.2 Perangkat Lunak.....	43
5.2 Implementasi Algoritme	43
5.2.1 Universe of Discourse	43
5.2.2 Inisialisasi Populasi.....	44
5.2.3 Crossover.....	45
5.2.4 Mutation	46
5.2.5 Sub-Himpunan.....	47
5.2.6 Fuzzyfikasi	48
5.2.7 FLR	48
5.2.8 FLRG	49
5.2.9 Defuzzyfikasi.....	49
5.2.10 RMSE & Fitness	50
5.2.11 Seleksi.....	51
5.3 Hasil Implementasi	52
BAB 6 PENGUJIAN DAN ANALISIS	54
6.1 Pengujian Popsize	54
6.1.1 Pengujian Produk Topi	54
6.1.2 Pengujian Produk Sabuk	54
6.1.3 Pengujian Produk Dompot	55
6.1.4 Pengujian Produk Tas.....	56
6.1.5 Pengujian Produk Jaket	56
6.1.6 Pengujian Produk Sepatu	57
6.2 Pengujian Cr & Mr.....	58
6.2.1 Pengujian Produk Topi	58
6.2.2 Pengujian Produk Sabuk	59
6.2.3 Pengujian Produk Dompot	59
6.2.4 Pengujian Produk Tas.....	60
6.2.5 Pengujian Produk Jaket	61
6.2.6 Pengujian Produk Sepatu	61
6.3 Pengujian Tingkat Akurasi.....	62
6.3.1 Tingkat Akurasi Produk Topi	63
6.3.2 Tingkat Akurasi Produk Sabuk.....	63

6.3.3 Tingkat Akurasi Produk Dompot	64
6.3.4 Tingkat Akurasi Produk Tas	64
6.3.5 Tingkat Akurasi Produk Jacket	65
6.3.6 Tingkat Akurasi Produk Sepatu	65
6.4 Hasil Analisis	66
BAB 7 PENUTUP	67
7.1 Kesimpulan.....	67
7.2 Saran	67
DAFTAR PUSTAKA	68
LAMPIRAN A DATA PENGUJIAN POPSIZE	70
LAMPIRAN B DATA PENGUJIAN CROSSOVER RATE DAN MUTATION RATE.....	74
LAMPIRAN C DATA PENGUJIAN TINGKAT AKURASI	78



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	5
Tabel 3.1 Data Produksi	12
Tabel 3.2 Pengujian Popsiz	13
Tabel 3.3 Pengujian C_r dan M_r	14
Tabel 3.4 Pengujian Tingkat Akurasi	14
Tabel 4.1 Inisialisasi Populasi Awal	33
Tabel 4.2 <i>Parent Crossover</i>	34
Tabel 4.3 <i>Child Crossover</i>	34
Tabel 4.4 <i>Parent Mutation</i>	34
Tabel 4.5 <i>Child Mutation</i>	34
Tabel 4.6 Populasi Baru	35
Tabel 4.7 Inisialisasi Sub-Himpunan	35
Tabel 4.8 <i>Fuzzyfikasi</i>	36
Tabel 4.9 FLR	37
Tabel 4.10 FLRG	37
Tabel 4.11 <i>Defuzzyfikasi</i>	38
Tabel 4.12 Hasil Prediksi	38
Tabel 4.13 Kuadrat <i>Error</i>	39
Tabel 4.14 Hasil RMSE	40
Tabel 4.15 Hasil <i>Fitness</i>	40
Tabel 4.16 Hasil Seleksi	41
Tabel 5.1 Spesifikasi Perangkat Keras	43
Tabel 5.2 Spesifikasi Perangkat Lunak	43
Tabel 6.1 Jumlah Popsiz Optimal	58
Tabel 6.2 Parameter Optimal Algoritme Genetika	62
Tabel 6.3 Tingkat Akurasi Topi	63
Tabel 6.4 Tingkat Akurasi Sabuk	63
Tabel 6.5 Tingkat Akurasi Dompot	64
Tabel 6.6 Tingkat Akurasi Tas	64
Tabel 6.7 Tingkat Akurasi Jaket	65

Tabel 6.8 Tingkat Akurasi Sepatu 65
Tabel 6.9 Hasil Tingkat Akurasi 66



DAFTAR GAMBAR

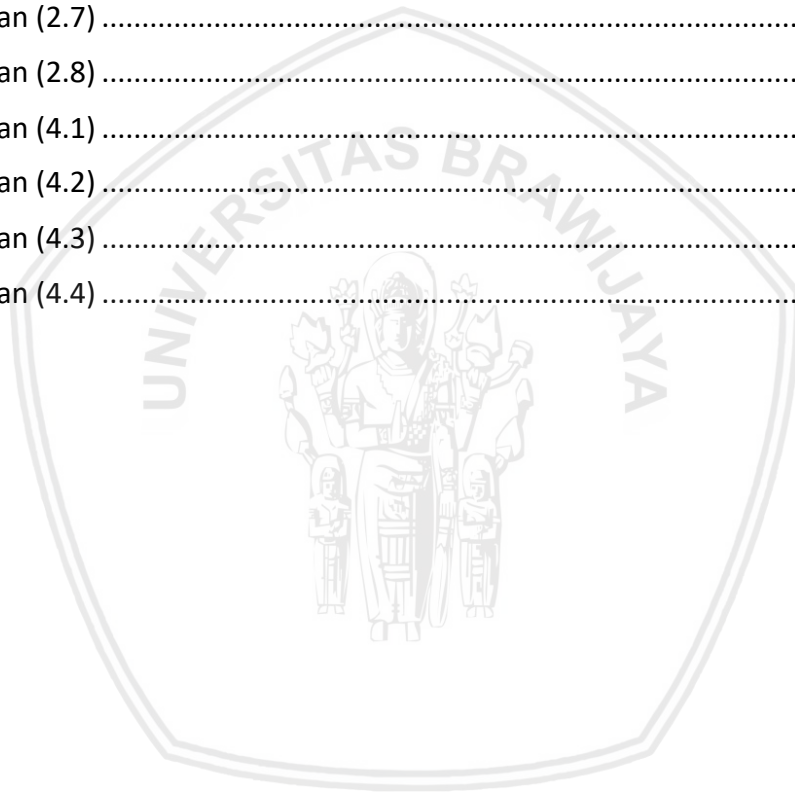
Gambar 2.1 <i>Parent One-Cut Point Crossover</i>	9
Gambar 2.2 <i>Child One-Cut Point Crossover</i>	9
Gambar 2.3 <i>Parent Random Mutation</i>	9
Gambar 2.4 <i>Child Random Mutation</i>	9
Gambar 3.1 Tahapan Penelitian.....	11
Gambar 4.1 Diagram Alir Optimasi <i>Fuzzy Time Series</i>	16
Gambar 4.2 Diagram Alir <i>Universe of Discourse</i>	17
Gambar 4.3 Diagram Alir Inisialisasi Populasi	18
Gambar 4.4 Diagram Alir Urut Populasi.....	19
Gambar 4.5 Diagram Alir <i>Crossover</i>	20
Gambar 4.6 Diagram Alir <i>One-Cut Point Crossover</i>	21
Gambar 4.7 Diagram Alir Urut <i>Crossover</i>	22
Gambar 4.8 Diagram Alir <i>Mutation</i>	24
Gambar 4.9 Diagram Alir Sub-Himpunan.....	25
Gambar 4.10 Diagram Alir <i>Fuzzy Time Series</i>	26
Gambar 4.11 Diagram Alir <i>Fuzzyfikasi</i>	28
Gambar 4.12 Diagram Alir FLR	28
Gambar 4.13 Diagram Alir FLRG.....	29
Gambar 4.14 Diagram Alir <i>Defuzzyfikasi</i>	30
Gambar 4.15 Diagram Alir RMSE & <i>Fitness</i>	31
Gambar 4.16 Diagram Alir Seleksi.....	32
Gambar 4.17 Halaman Awal Proses Algoritme Genetika	41
Gambar 4.18 Halaman Proses <i>Fuzzy Time Series</i>	42
Gambar 4.19 Halaman Hasil Prediksi	42
Gambar 5.1 Hasil Proses Algoritme Genetika	52
Gambar 5.2 Hasil Proses <i>Fuzzy Time Series</i>	52
Gambar 5.3 Hasil Prediksi	53
Gambar 6.1 Pengujian Popsiize Topi.....	54
Gambar 6.2 Pengujian Popsiize Sabuk	55
Gambar 6.3 Pengujian Popsiize Dompot	55

Gambar 6.4 Pengujian Popsize Tas	56
Gambar 6.5 Pengujian Popsize Jaket	57
Gambar 6.6 Pengujian Popsize Sepatu	57
Gambar 6.7 Pengujian Cr & Mr Topi	59
Gambar 6.8 Pengujian Cr & Mr Sabuk	59
Gambar 6.9 Pengujian Cr & Mr Dompok	60
Gambar 6.10 Pengujian Cr & Mr Tas.....	60
Gambar 6.11 Pengujian Cr & Mr Jaket.....	61
Gambar 6.12 Pengujian Cr & Mr Sepatu.....	62
Gambar 6.13 Perbandingan Tingkat Akurasi	66



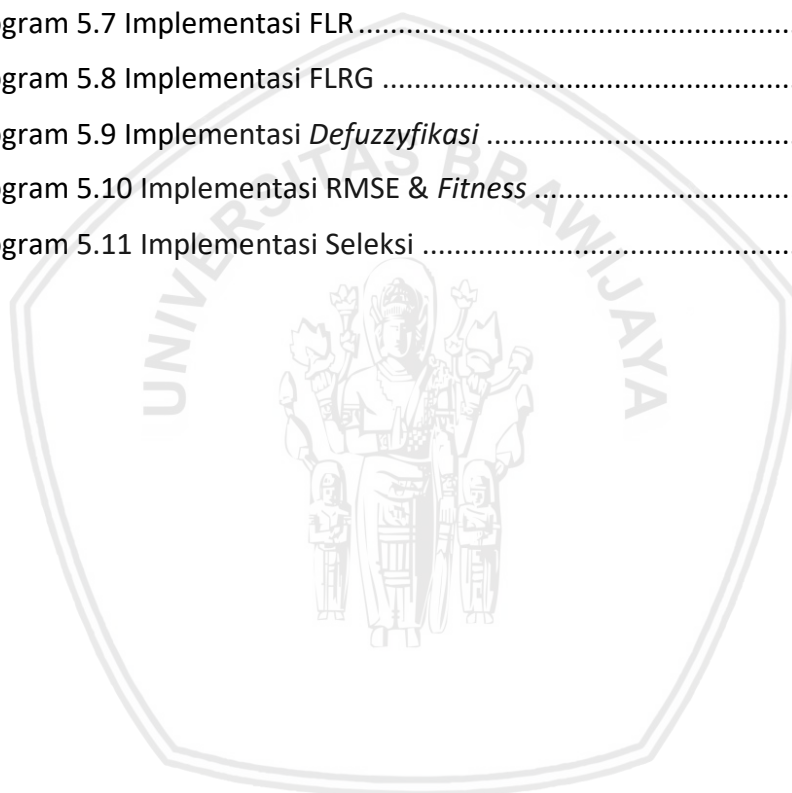
DAFTAR PERSAMAAN

Persamaan (2.1)	7
Persamaan (2.2)	7
Persamaan (2.3)	7
Persamaan (2.4)	8
Persamaan (2.5)	8
Persamaan (2.6)	10
Persamaan (2.7)	10
Persamaan (2.8)	10
Persamaan (4.1)	37
Persamaan (4.2)	38
Persamaan (4.3)	39
Persamaan (4.4)	40



DAFTAR KODE PROGRAM

Kode Program 5.1 Implementasi <i>Universe Of Discourse</i>	44
Kode Program 5.2 Implementasi Inisialisasi Populasi.....	45
Kode Program 5.3 Implementasi <i>Crossover</i>	46
Kode Program 5.4 Implementasi <i>Mutation</i>	47
Kode Program 5.5 Implementasi Sub-Himpunan	47
Kode Program 5.6 Implementasi <i>Fuzzyfikasi</i>	48
Kode Program 5.7 Implementasi FLR.....	48
Kode Program 5.8 Implementasi FLRG	49
Kode Program 5.9 Implementasi <i>Defuzzyfikasi</i>	50
Kode Program 5.10 Implementasi RMSE & <i>Fitness</i>	51
Kode Program 5.11 Implementasi Seleksi	51



DAFTAR LAMPIRAN

LAMPIRAN A DATA PENGUJIAN POPSIZE	70
A.1 Produk Topi.....	70
A.2 Produk Sabuk	70
A.3 Produk Dompot	71
A.4 Produk Tas	72
A.5 Produk Jacket	72
A.6 Produk Sepatu	73
LAMPIRAN B DATA PENGUJIAN CROSSOVER RATE DAN MUTATION RATE.....	74
B.1 Produk Topi.....	74
B.2 Produk Sabuk	74
B.3 Produk Dompot.....	75
B.4 Produk Tas	76
B.5 Produk Jacket	76
B.6 Produk Sepatu	77
LAMPIRAN C DATA PENGUJIAN TINGKAT AKURASI	78
C.1 Produk Topi.....	78
C.2 Produk Sabuk	78
C.3 Produk Dompot.....	79
C.4 Produk Tas	79
C.5 Produk Jacket	80
C.6 Produk Sepatu.....	80

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Fashion sudah menjadi salah satu hal yang menjadi perhatian bagi setiap golongan masyarakat. *Fashion* tidak lagi hanya sekedar kebutuhan sekunder, melainkan telah menjadi suatu kebutuhan primer yang ada di setiap kehidupan bermasyarakat. *Fashion* dan pakaian, mungkin adalah cara paling penting dimana hubungan sosial antar manusia dibangun, dialami dan dipahami (Barnard, 1996). *Fashion* saat ini, tidak hanya pada pakaian tetapi juga merambah ke aksesoris seperti tas, sabuk, dompet dan topi. Bahan baku produk *fashion* sangat beragam mulai dari kain seperti katun, denim, polyester dan masih banyak lagi. Kulit hewan (*leather*) juga merupakan bahan baku produk *fashion* yang sangat diminati saat ini. Kulit hewan memiliki berbagai macam keuntungan tersendiri yaitu tahan lama dan tahan segala cuaca.

Berbeda dengan produk berbahan dasar kain yang tidak tahan lama, bahan kulit dapat bertahan pada kurun waktu 10-15 tahun lamanya. Produk yang memakai bahan kulit (asli) memiliki kesan yang mewah dikarenakan harga yang ditawarkan lebih mahal dibandingkan dengan produk berbahan kain dengan model produk yang sama. Bahan kulit membutuhkan perawatan berkala untuk menjaga kualitasnya agar tetap lentur dan tahan lama. Produk yang memakai bahan dasar kulit sebagai bahan bakunya, membutuhkan ongkos lebih dalam hal pembuatan dan juga pembelian bahan baku.

SAGA *leather fashion* yang telah berdiri dari tahun 1999, telah memproduksi lebih dari 1000 produk kulit seperti jaket, tas, dompet, sepatu, sabuk dan topi dari berbagai macam kulit seperti kulit sapi, kulit kambing dan kulit domba. SAGA *leather fashion* memproduksi barang sesuai dengan permintaan yang ada, yang menyebabkan produksi setiap bulan berbeda-beda dan mengakibatkan tidak optimalnya sisa bahan baku (kulit) yang telah digunakan, dimana sisa kulit tersebut dapat digunakan untuk memproduksi barang-barang yang dalam proses pembuatannya membutuhkan bahan kulit yang relatif kecil/sedikit. Oleh sebab itu dibutuhkan suatu upaya untuk mengatasi masalah sisa bahan baku tersebut, yaitu dengan memprediksi jumlah barang yang akan diproduksi pada bulan berikutnya, kemudian dari hasil prediksi tersebut digunakan untuk memproduksi barang dengan menggunakan sisa kulit pada bulan sebelumnya. Prediksi pada penelitian ini menggunakan metode *fuzzy time series*, dimana metode *fuzzy time series* merupakan metode logika *fuzzy* yang mengacu pada data historis untuk memprediksi nilai yang akan didapatkan untuk jenjang waktu kedepannya.

Penelitian-penelitian yang dilakukan sebelum penelitian ini, mengoptimasi *fuzzy time series* menggunakan metode algoritme genetika dan mendapatkan hasil prediksi yang baik dan nilai *error* yang relatif rendah. Algoritme genetika merupakan metode optimasi yang dapat menyelesaikan masalah yang rumit dan kompleks dengan waktu relatif singkat, karena algoritme genetika dalam pengoperasiannya dapat menggunakan berbagai macam data, seperti data

numerik dan data yg menggunakan fungsi analitis (Haupt & Haupt, 2004). Salah satu penelitian yang menggunakan algoritme genetika yaitu untuk mengoptimasi *fuzzy time series* dalam memprediksi kepadatan lalu lintas di jalan tol (Wicaksana, et al., 2017). Dalam penelitian tersebut, penulis menggunakan data kendaraan yang melintasi jalan tol pada Januari 2016 sampai dengan Desember 2016 untuk memprediksi jumlah kendaraan yang akan melintas pada Januari 2017. Dan juga penelitian yang membahas tentang prediksi produksi gula pasir untuk bulan berikutnya menggunakan *fuzzy time series* yang dioptimasi dengan algoritme genetika (Ridhwan, et al., 2017).

Berdasarkan uraian diatas, penulis ingin melakukan penelitian untuk memprediksi jumlah barang yang akan diproduksi untuk bulan berikutnya pada SAGA *leather fashion* menggunakan metode *fuzzy time series*, dan untuk menghasilkan hasil prediksi yang optimal, digunakan metode algoritme genetika dalam menentukan batas interval pada metode *fuzzy time series*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, dapat disimpulkan rumusan masalah sebagai berikut:

1. Bagaimana hasil prediksi menggunakan *fuzzy time series* yang dioptimasi dengan algoritme genetika pada SAGA *leather fashion*?
2. Bagaimana nilai parameter algoritme genetika yang sesuai untuk mendapatkan hasil yang optimal?
3. Bagaimana nilai *error* yang dihasilkan setelah dioptimasi menggunakan algoritme genetika?

1.3 Tujuan

Berdasarkan rumusan masalah diatas, dapat disimpulkan tujuan dari penelitian ini adalah:

1. Menerapkan metode *fuzzy time series* yang dioptimasi dengan algoritme genetika untuk memprediksi jumlah produksi barang SAGA *leather fashion*.
2. Mengetahui nilai parameter algoritme genetika yang sesuai untuk mendapatkan hasil yang optimal.
3. Mengetahui nilai *error* yang dihasilkan setelah dioptimasi menggunakan algoritme genetika.

1.4 Manfaat

Berdasarkan tujuan diatas, dapat disimpulkan manfaat dari penelitian ini adalah mampu memprediksi jumlah barang yang akan diproduksi SAGA *leather fashion* untuk bulan kedepannya, untuk mengoptimalkan sisa kulit yang telah digunakan.

1.5 Batasan Masalah

1. Data yang digunakan untuk penelitian diperoleh dari SAGA *leather fashion* berupa jaket, tas, dompet, sepatu, topi dan sabuk selama rentang waktu 24 bulan.
2. Metode algoritme genetika yang digunakan yaitu *one-cut point crossover*, *random mutation* dan seleksi *elitism*.

1.6 Sistematika Pembahasan

Bagian ini berisi struktur skripsi mulai dari bab pendahuluan sampai bab penutup dan deskripsi singkat dari masing-masing bab. Diharapkan bagian ini dapat membantu pembaca dalam memahami sistematika pembahasan isi dalam skripsi ini.

BAB I PENDAHULUAN

Pada bab ini berisikan latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika pembahasan.

BAB II LANDASAN KEPUSTAKAAN

Pada bab ini membahas tentang kajian pustaka yg berisikan penelitian-penelitian sebelum penelitian ini dan membahas tentang dasar teori penelitian ini.

BAB III METODOLOGI

Pada bab ini membahas langkah-langkah penulis dalam menyelesaikan penelitian menggunakan algoritme genetika untuk optimasi peningkatan laba.

BAB IV ALGORITME

Pada bab ini menjelaskan perhitungan manual dalam menyelesaikan masalah pada penelitian ini.

BAB V IMPLEMENTASI

Pada bab ini menjelaskan hasil dari algoritme secara terperinci dalam menyelesaikan masalah prediksi jumlah produksi SAGA *leather fashion*.

BAB VI PENGUJIAN DAN ANALISIS

Pada bab ini menampilkan hasil pengujian algoritme yang menampilkan solusi terbaik yang akan digunakan.

BAB VII PENUTUP

Pada bab ini berisikan kesimpulan dan saran tentang apa yang dihasilkan oleh penelitian ini.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Sebelum penulis melakukan penelitian ini terdapat beberapa penelitian-penelitian sebelumnya yang membahas tentang prediksi menggunakan *fuzzy time series* yang dioptimasi menggunakan algoritme genetika antara lain.

Pertama adalah penelitian yang dilakukan oleh Andhi Surya Wicaksana pada tahun 2017. Dari hasil penelitian dapat disimpulkan bahwa algoritme genetika yang digunakan untuk mengoptimasi metode perhitungan *fuzzy time series*, mendapatkan hasil prediksi volume kendaraan yang lebih akurat dalam proses pembagian interval. Untuk hasil uji coba prediksi volume kendaraan pada bulan Januari 2017 dengan algoritme genetika, hasil prediksi yaitu 4.202.565 unit kendaraan. Nilai *popsize* yang digunakan yaitu 125, *cr* (*crossover rate*) berupa 0,9 dan *mr* (*mutation rate*) berupa 0,1 dengan iterasi pada generasi 2000. Sedangkan untuk pengujian tingkat hasil *error* pada bulan Oktober 2016 sampai dengan Maret 2017, menghasilkan persentase *error* 16,66% menggunakan metode *Average Forecasting Error Rate* (AFER). Untuk hasil *fitness* didapatkan nilai 0,03030303 yang menunjukkan hasil tersebut sangat mendekati nilai uji berupa 4.202.532 unit. Sedangkan untuk prediksi tanpa optimasi menghasilkan 4.624.384 unit kendaraan.

Kedua adalah penelitian yang dilakukan oleh Tahajuda Mandariansah pada tahun 2017. Penelitian ini berhasil mendapatkan nilai *fitness* sebesar 21,1814 menggunakan jumlah interval 7, kombinasi *cr* dan *mr* masing-masing 0,9 dan 0,5, berdasarkan 1050 populasi dan generasi sebesar 100. Algoritme *fuzzy time series*, membentuk titik tengah interval yang digunakan untuk batas nilai kromosom. Kemudian teknik yang digunakan untuk reproduksi kromosom yaitu *extended intermediete crossover* dan *random mutation*, sedangkan kombinasi *elitism* dan *roulette wheel* digunakan untuk proses seleksi masing-masing 75% dan 25%. Penelitian ini mempunyai nilai *error* sebesar 4,7211%, yang didapatkan dengan menggunakan metode AFER. Nilai *error* tersebut lebih baik dari penelitian Andhi Surya Wicaksana (2017).

Ketiga adalah penelitian yang dilakukan oleh Khaira Istiqara pada tahun 2017. Pada penelitian ini, parameter-parameter yang digunakan yaitu kromosom berjumlah 60, kombinasi nilai *cr* dan *mr* berupa 0,4 dan 0,2, ukuran populasi sebesar 360 dan jumlah generasi 550. Dari parameter tersebut, metode algoritme genetika yang digunakan yaitu *one-cut point crossover*, *uniform mutation* dan *elitism selection*. Proses perhitungan nilai *error* menggunakan metode *Mean Absolute Percentage Error* (MAPE) menghasilkan *error* sebesar 2,266776%, dimana nilai tersebut termasuk dalam kategori rendah.

Keempat adalah penelitian yang dilakukan oleh Afif Ridhwan pada tahun 2017. Dari penelitian ini, penulis mendapatkan hasil *error* MAPE sebesar 2,07%. Hasil tersebut didapat berdasarkan nilai *popsize* = 50, kombinasi *cr* = 0.7 *mr* = 0.3,

dan iterasi maksimum = 250. Hasil *error* tersebut lebih rendah dari penelitian Khaira Istiqara (2017) yang menggunakan metode perhitungan yang sama.

Berdasarkan penelitian-penelitian yang telah diuraikan diatas, penulis ingin melakukan penelitian untuk memprediksi jumlah produksi barang SAGA *leather fashion* menggunakan *fuzzy time series*, yang kemudian dioptimasi menggunakan metode algoritme genetika.

Tabel 2.1 Kajian Pustaka

Judul	Tahun	Saran
Algoritme genetika untuk optimasi <i>fuzzy time series</i> dalam memprediksi kepadatan lalu lintas di jalan tol (Andhi Surya).	2017	Lebih baik untuk melakukan observasi langsung guna mendapatkan hasil data yang lebih baik.
Optimasi <i>fuzzy time series</i> untuk peramalan kebutuhan hidup layak kota kediri dengan menggunakan algoritme genetika (Tahajuda).	2017	Menambahkan beberapa faktor pendukung untuk memaksimalkan nilai hasil peramalan.
Prediksi kebutuhan air PDAM Kota Malang menggunakan metode <i>fuzzy time series</i> dengan algoritme genetika (Khaira).	2017	Untuk menghasilkan prediksi yang lebih baik, disarankan menggunakan metode algoritme genetika yang berbeda.
Peramalan produksi gula pasir menggunakan <i>fuzzy time series</i> dengan optimasi algoritma genetika (Afif).	2017	Menggunakan jumlah data yang lebih banyak dalam kombinasi <i>fuzzy time series</i> dan algoritme genetika.

2.2 Dasar Teori

Pada sub-bab ini dijelaskan secara rinci tentang teori-teori yang digunakan untuk mendukung peneliti dalam menyelesaikan masalah pada penelitian ini.

2.2.1 SAGA Leather Fashion

SAGA *leather fashion* merupakan sebuah butik yang terletak di provinsi Bali, yang memproduksi berbagai macam barang berbahan baku kulit. Kulit merupakan barang produksi, yang dihasilkan dari keterampilan penyamak kulit (*tanner*) dari kulit binatang, reptil, ikan dan burung (NIIR Board, 2011). Kulit hewan sebelum digunakan dalam produksi, membutuhkan berbagai proses agar memiliki kualitas yang bagus dan siap menjadi bahan baku untuk diproduksi. Barang yang diproduksi berupa jaket, tas, dompet, sepatu, sabuk dan topi sedangkan kulit yang digunakan untuk bahan baku yaitu kulit sapi, kulit kambing dan kulit domba.

2.2.2 Prediksi

Prediksi merupakan suatu ilmu/metode yang memberikan gagasan atau nilai yang akan terjadi di waktu yang akan datang. Prediksi melibatkan suatu estimasi atau perhitungan untuk masa depan yang diharapkan (Kenneth, 2015). Dalam prediksi, diperlukan data historis yang digunakan untuk menemukan sebuah gagasan atau nilai di masa depan.

2.2.3 Logika Fuzzy

Logika *fuzzy* merupakan sebuah metode yang mempresentasikan setiap keadaan dalam interval $[0,1]$ (Zadeh, 1965), yang berbeda dengan logika boolean dimana sebuah hasil dibatasi dengan 0 dan 1. Oleh karena itu muncul lah logika fuzzy untuk mencegah suatu makna yang ambigu. Logika *fuzzy* dapat mengatasi suatu konsep yang sering ditemukan dalam kehidupan sehari-hari, seperti panas, sangat panas, dingin dan sangat dingin. Logika *fuzzy* menggunakan variabel linguistik dan menekankan pada tingkat kebenaran, seperti pemikiran manusia (Kayacan, 2016).

2.2.4 Fuzzy Time Series

Fuzzy Time Series (FTS) merupakan sebuah metode yang dapat digunakan untuk menyelesaikan masalah prediksi menggunakan data historis yang bernilai linguistik (Chen,1995). Data historis tersebut dapat membentuk suatu pola, untuk kemudian dilakukan prediksi berdasarkan pola-pola yang terdapat pada data yang digunakan. Metode FTS ini dikembangkan untuk kebutuhan proses dinamik yang mangandalkan data dari variabel linguistik (Hong, 2005), variabel linguistik merupakan variabel yang memiliki nilai berupa kata. Pada tahun 1993, Song & Chissom melakukan penelitian untuk memprediksi pendaftaran pada Universitas Alabama. Persamaan yang digunakan dalam penelitian tersebut yaitu:

Jika $Y(t)$ ($t = \dots, 0, 1, 2, \dots$) bagian dari himpunan R , yang merupakan *universe of discourse* (himpunan semesta) dimana set *fuzzy* $f_i(t)$ ($i = 1, 2, \dots$) sudah didefinisikan dan $F(t)$ adalah bagian dari himpunan $f_i(t)$ ($i = 1, 2, \dots$). Maka pada $Y(t)$ ($t = \dots, 0, 1, 2, \dots$) terdapat $F(t)$ yang dinyatakan sebagai *fuzzy time series*.

Dari persamaan diatas, $F(t)$ dapat disimpulkan sebagai variabel linguistik dan $f_i(t)$ ($i = 1, 2, \dots$) yang kemungkinan sebagai nilai linguistik dari $F(t)$. Karena nilai dari $F(t)$ dapat berbeda pada tiap waktu, $F(t)$ merupakan fungsi waktu dari t . Karena *universe of discoure* dapat berbeda tiap waktu, maka $F(t)$ juga dapat berbeda tiap waktu. Jika $F(t)$ hanya disebabkan oleh $F(t - 1)$, maka hubungan ini dipresentasikan menjadi $F(t - 1) \rightarrow F(t)$.

Dari persamaan diatas, *fuzzy time series* dapat dibedakan menjadi:

- *Time-variant fuzzy time series*
 $F(t) \rightarrow F(t - 1)$ tidak sama dengan $F(t - 1) \rightarrow F(t - 2)$
- *Time-invariant fuzzy time series*
 $F(t) \rightarrow F(t - 1)$ sama dengan $F(t - 1) \rightarrow F(t - 2)$

1. Prediksi fuzzy time series

Pada penelitian sebelumnya yang dilakukan oleh Song & Chissom (1993), mereka memprediksi jumlah pendaftaran pada Universitas Alabama menggunakan persamaan:

$$A = A_{i-1} \circ R \tag{2.1}$$

A_{i-1} merupakan pendaftaran pada tahun $i-1$ yang direpresentasikan dengan himpunan fuzzy, A_i merupakan prediksi pendaftaran yang direpresentasikan dengan himpunan fuzzy, “ \circ ” merupakan operator *max-min* dan R merupakan relasi fuzzy yang menunjukkan hubungan fuzzy antara fuzzy time series. Metode prediksi ini membutuhkan komputasi yang rumit untuk mendapatkan relasi fuzzy R jika bernilai sangat besar (Chen, 1996).

Oleh karena itu Chen (1996) melakukan penelitian dengan menggunakan metode baru pada proses prediksi dalam fuzzy time series sebagai berikut:

1. Membagi universe of discourse

$$U = [D_{min} - D_1, D_{max} + D_2] \tag{2.2}$$

menjadi lebih panjang dan memiliki panjang interval yang sama u_1, u_2, \dots, u_k . U didefinisikan sebagai *universe of discourse*, dimana D_{min} dan D_{max} merupakan nilai maksimum dan minimal dari data histori. Sedangkan D_1 dan D_2 merupakan bilangan positif yang dapat ditentukan.

2. Mendefinisikan himpunan fuzzy menjadi A_1, A_2, \dots, A_k sebagai variabel linguistik, dimana himpunan fuzzy A_1, A_2, \dots, A_k pada universe of discourse didefinisikan dengan persamaan:

$$\begin{aligned} A_1 &= a_{11}/u_1 + a_{12}/u_2 + \dots + a_{1k}/u_k, \\ A_2 &= a_{21}/u_1 + a_{22}/u_2 + \dots + a_{2k}/u_k \end{aligned} \tag{2.3}$$

⋮

$$A_m = a_{m1}/u_1 + a_{m2}/u_2 + \dots + a_{mk}/u_k$$

a_{ij} merupakan derajat fungsi keanggotaan yang bernilai pada interval 0 dan 1, dimana $1 \leq i \leq m$ dan $1 \leq j \leq k$. Proses *fuzzyfikasi* pada data aktual $[i]$ yaitu dengan menentukan apakah i termasuk dalam interval u_k , dan jika nilai derajat keanggotaan u_k tertinggi pada A_k , maka *fuzzyfikasi* i dapat ditentukan sebagai A_k .

3. Mendapatkan Fuzzy Logical Relationship (FLR) yaitu $A_i \rightarrow A_j$, dimana A_i merupakan fuzzyfikasi data i dan A_j adalah $i + 1$. A_i direpresentasikan sebagai *current state* dan A_j direpresentasikan sebagai *next state*.

4. Mengelompokkan hasil FLR yang telah didapatkan sebelumnya menjadi Fuzzy Logical Relationship Group (FLRG).

5. Kalkulasi hasil dari prediksi dengan beberapa prinsip yaitu:

- Jika hasil *fuzzyfikasi* i adalah A_i , dan hanya ada satu FLR di dalam FLRG yang *current state* dari FLR adalah A_i , seperti:



$$A_i \rightarrow A_j \quad (2.4)$$

Dimana A_i dan A_j adalah himpunan *fuzzy* dan nilai tertinggi derajat keanggotaan dari A_j terdapat pada interval u_j , dan *median* dari u_j adalah m_j , maka hasil tahun prediksi $i + 1$ adalah m_j ,

- Jika hasil *fuzzyfikasi* tahun i adalah A_i , dan ada beberapa FLR di dalam FLRG yang *current state* dari FLR adalah A_i , seperti:

$$\begin{aligned} A_i &\rightarrow A_{j1}, \\ A_i &\rightarrow A_{j2}, \\ &\vdots \\ A_i &\rightarrow A_{jk}. \end{aligned} \quad (2.5)$$

Dimana $A_i, A_{j1}, A_{j2}, \dots, A_{jk}$ adalah himpunan *fuzzy* dan nilai tertinggi derajat keanggotaan dari $A_{j1}, A_{j2}, \dots, A_{jk}$ terdapat pada interval u_1, u_2, \dots, u_k , dan *median* dari u_1, u_2, \dots, u_k adalah m_1, m_2, \dots, m_k , maka hasil dari tahun prediksi $i + 1$ adalah $(m_1 + m_2 + \dots + m_k)/k$.

- Jika hasil *fuzzyfikasi* tahun i adalah A_i dan tidak terdapat FLRG yang *current state* adalah A_i , dimana nilai tertinggi derajat keanggotaan dari A_i terdapat pada interval u_i dan *median* dari u_i adalah m_i , maka hasil tahun prediksi $i + 1$ adalah m_i .

2.2.5 Algoritme Genetika

Algoritme genetika (*genetic algorithm*) merupakan sub-set dari algoritme evolusi, yang merupakan teknik optimasi berbasis seleksi alam. Pada teori evolusi, individu-individu dalam suatu populasi menghasilkan *offspring* (keturunan) yang lebih baik dari induknya. Algoritme genetika sendiri telah digunakan dalam dua cara, yaitu sebagai sebuah teknik untuk menyelesaikan masalah teknologi dan menyederhanakan model ilmiah untuk bisa menjawab pertanyaan tentang alam (Mitchell, 1998). Algoritme genetika sangat populer untuk menyelesaikan masalah-masalah optimasi, dikarenakan algoritme genetika dapat melakukan pencarian dengan radius yang luas dan tidak tergantung pada daerah optimum lokal (Mahmudy, 2015). Alur proses pada algoritme genetika yaitu sebagai berikut:

1. Inisialisasi

Merupakan tahapan awal yang dilakukan dalam menyelesaikan masalah menggunakan algoritme genetika, dengan cara membangkitkan angka random yang disebut dengan kromosom.

2. Reproduksi

Merupakan tahapan yang dilakukan setelah populasi awal dibangkitkan. Reproduksi merupakan proses untuk membentuk individu-individu baru (*offspring*) dari induk (*parent*). Pada proses untuk membentuk *offspring* tersebut, terdapat dua cara yaitu dengan *crossover* dan *mutation*.

Crossover dilakukan dengan secara acak mengambil dua *parent* dari populasi untuk ditukar nilai dari kromosomnya, sedangkan *mutation* dilakukan dengan secara acak mengambil sebuah *parent* dari populasi untuk diubah nilai dari kromosomnya. Metode *crossover* yang digunakan dalam penelitian ini adalah *one-cut point crossover* dikarenakan pada penelitian sejenis mendapatkan hasil yang optimal. Pada *crossover* ini, kromosom diberi tanda (*cut point*) untuk kemudian ditukarkan nilai antar dua *parent*. Pada Gambar 2.1, merupakan contoh *one-cut point crossover* yang dilakukan oleh Istiqara, et al. (2017).

P ₁	1460082	1484194	1618045	1628659	1634739	1635333	1682896	1790173
	1799061	1829026	1893993	1952642	1976014	1983040	2001933	2152659
	2261425	2330503	2350621					
P ₃	1409634	1422696	1502909	1506285	1533303	1575276	1600001	1634849
	1780903	1796801	1815578	1824030	1840183	2004382	2108949	2241182
	2366000	2387108	2391940					

Gambar 2.1 Parent One-Cut Point Crossover

Cut point berada diantara nilai gen ke-10 dan nilai ke-11, kemudian nilai gen ke-1 sampai ke-10 pada P₁ ditukarkan dengan nilai gen pada P₃ yang menghasilkan C₁ seperti pada Gambar 2.2.

C ₁	1409634	1422696	1502909	1506285	1533303	1575276	1600001	1634849
	1780903	1796801	1893993	1952642	1976014	1983040	2001933	2152659
	2261425	2330503	2350621					

Gambar 2.2 Child One-Cut Point Crossover

Sedangkan untuk jenis *mutation* yang digunakan adalah *random mutation*. Pada *mutation* ini, diambil sebuah nilai pada kromosom yang disebut dengan gen, yang kemudian akan diubah nilai gen tersebut. Pada Gambar 2.3, merupakan contoh *random mutation* yang dilakukan oleh Istiqara, et al. (2017).

P ₂	1484316	1494533	1585013	1655655	1656069	1702098	1773172	1872123
	2040909	2057830	2058752	2101842	2189983	2226703	2243157	2294671
	2309276	2385891	2387294					

Gambar 2.3 Parent Random Mutation

Sebelum melakukan *mutation*, terdapat aturan yaitu nilai *mutation* didapatkan secara acak dengan nilai gen sebelumnya sebagai batas bawah dan nilai setelahnya sebagai batas atas. Pada P₂ diatas nilai gen yang akan mengalami *mutation* yaitu gen ke-5, dimana nilai *mutation* harus berada pada interval nilai gen-4 dan gen ke-6. Pada Gambar 2.4, merupakan hasil dari *mutation* pada P₂.

C ₂	1484316	1494533	1585013	1655655	1665952	1702098	1773172	1872123
	2040909	2057830	2058752	2101842	2189983	2226703	2243157	2294671
	2309276	2385891	2387294					

Gambar 2.4 Child Random Mutation

3. Evaluasi

Merupakan tahapan untuk menentukan baik atau tidaknya suatu kromosom, yang dinamakan dengan nilai kebugaran (*fitness*). *Fitness* didapatkan dengan melakukan perhitungan berulang-ulang pada populasi sampai didapatkan nilai

fitness minimal yang telah ditentukan. Penentuan nilai pada permasalahan maksimum, dapat menggunakan Persamaan 2.6 (Mahmudy, 2015).

$$fitness = f(x) \quad (2.6)$$

4. Seleksi

Merupakan tahapan pemilihan individu-individu terbaik dari tahapan reproduksi yang kemudian individu-individu tersebut akan menjadi induk pada generasi berikutnya. Nilai *fitness* memiliki pengaruh dalam melakukan seleksi, dikarenakan semakin besar nilai *fitness* maka kemungkinan terpilihnya individu terbaik semakin besar, yang akan membuat generasi berikutnya menjadi lebih baik dari sebelumnya. Pada penelitian ini menggunakan seleksi *elitism*, dimana pada metode ini individu yang memiliki *fitness* tinggi memiliki kemungkinan untuk dapat terpilih. Tetapi nilai *fitness* pada individu tersebut dapat mengalami kerusakan dan mengakibatkan nilai *fitness* individu tersebut mengalami penurunan, oleh sebab itu individu yang memiliki nilai *fitness* terbaik dalam populasi harus melakukan pengkopian kromosom agar nilai *fitness* tetap bertahan dalam menghadapi proses evolusi (Dharmaputra, 2012).

2.2.6 RMSE & RMSPE

Untuk menghitung nilai *error*, pada penelitian ini menggunakan metode *Root Mean Square Error* (RMSE). Dari 145 pakar prediksi yang diberikan pertanyaan tentang metode *error* apa yang sering digunakan ketika menghitung keakuratan metode prediksi, mayoritas pakar tersebut memilih RMSE sebagai metode untuk menghitung *error* (Armstrong & Collopy, 1992). Pada perhitungan RMSE, yaitu nilai data aktual dikurangi nilai prediksi yang kemudian dikuadratkan, dan rata-rata hasil kuadrat tersebut diakarkan.

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (x_i - y_i)^2}{n}} \quad (2.7)$$

Sedangkan untuk mengetahui tingkat akurasi yang dihasilkan, menggunakan metode *Root Mean Square Percentage Error* (RMSPE).

$$RMSPE = \sqrt{\frac{\sum_{t=1}^n \left(\frac{x_i - y_i}{x_i}\right)^2}{n}} \times 100 \quad (2.8)$$

Pada metode RMSPE, nilai data aktual dikurangi nilai prediksi yang kemudian nilai tersebut dibagi nilai data aktual, hasil tersebut dikuadratkan dan rata-rata hasil kuadrat tersebut diakarkan. Setelah didapatkan hasil pengakaran tersebut, kemudian dikalikan 100.

Keterangan:

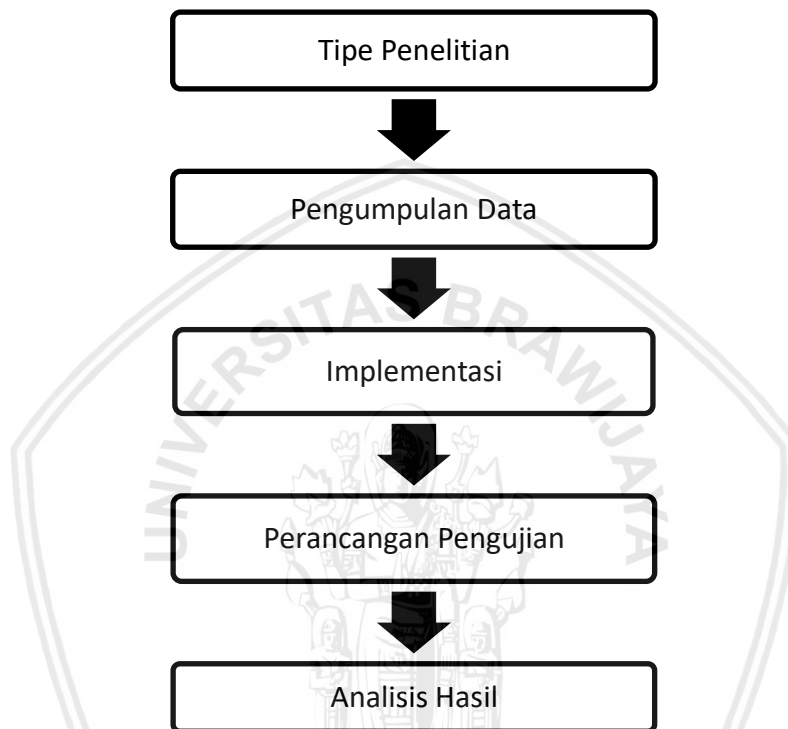
x_i = data aktual

y_i = data prediksi

n = banyaknya data prediksi

BAB 3 METODOLOGI

Bab ini berisikan penjelasan tahapan-tahapan yang dilakukan dalam penelitian ini, meliputi tipe penelitian, pengumpulan data, implementasi, perancangan pengujian dan analisis hasil. Tahapan penelitian dapat ditunjukkan pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.1 Tipe Penelitian

Tipe penelitian pada penelitian ini adalah non-implementatif, yang berfokus pada analisis dalam menyelesaikan masalah yang dihadapi. Tipe penelitian ini dibagi menjadi 2 yaitu:

- Deskriptif

Pada penelitian ini, aspek yang diutamakan yaitu menjelaskan hasil yang diperoleh dari berbagai percobaan pada suatu objek dalam penelitian

3.2 Pengumpulan Data

Penelitian ini, menggunakan data dari *SAGA leather fashion*. Data yang didapatkan berupa jumlah produk yang dihasilkan dalam kurun waktu 24 bulan. Pada Tabel 3.1, ditampilkan data produksi dari *SAGA leather fashion*.

Tabel 3.1 Data Produksi

Tahun	Bulan	Sapi		Kambing		Domba	
		Topi	Sabuk	Dompot	Tas	Jaket	Sepatu
2016	Januari	47	57	139	65	72	36
	Ferbuari	45	52	163	73	66	37
	Maret	43	59	153	70	75	39
	April	34	53	141	57	73	37
	Mei	47	68	152	69	76	47
	Juni	40	63	147	74	84	43
	Juli	47	64	142	71	81	41
	Agustus	45	53	148	74	78	40
	September	45	53	151	64	73	43
	Oktober	46	83	139	58	69	42
	November	47	70	142	56	82	36
	Desember	32	77	150	73	68	41
2017	Januari	44	55	162	62	77	45
	Ferbuari	48	57	137	62	78	36
	Maret	34	72	141	66	82	41
	April	38	69	151	61	82	44
	Mei	41	82	135	64	68	44
	Juni	33	62	134	66	79	43
	Juli	39	81	135	71	72	37
	Agustus	40	80	155	60	85	37
	September	36	74	146	56	82	38
	Oktober	38	75	149	61	69	36
	November	36	54	144	73	79	45
	Desember	42	65	144	69	84	43

3.3 Implementasi

a. Algoritme

Pada penelitian ini, algoritme yang digunakan meliputi:

- Fuzzy time series

Metode untuk memprediksi apa yang akan terjadi pada waktu kedepannya menggunakan data historis.

- Algoritme genetika

Metode perhitungan berbasis seleksi alam dimana sebuah individu bereproduksi untuk menghasilkan keturunan yang lebih baik dari induknya.

b. Penerapan metode

Pada penelitian ini, akan mengimplementasikan algoritme genetika pada metode *fuzzy time series* untuk prediksi jumlah produksi pada SAGA *leather fashion*. Data yang diinputkan pada sistem ini yaitu generasi, popsiize, *cr*, *mr*

dan juga memilih produk yang akan diprediksi. Sedangkan untuk output yang dihasilkan berupa hasil prediksi beserta *error* yang dihasilkan dari populasi tersebut.

3.4 Peralatan Pendukung

- a. Perangkat keras
 - MacBook Pro 13-inch
 - Processor 2,7 GHz Intel Core i7
 - RAM 4GB 1067 MHz DDR3
 - Graphics Intel HD 3000 384 MB
- b. Perangkat lunak
 - MacOS High Sierra
 - Netbeans 8.2

3.5 Perancangan Pengujian

Penelitian ini, melakukan pengujian pada jumlah popsize, nilai kombinasi *cr* dan *mr*, dan tingkat akurasi.

1. Pengujian popsize

Pengujian ukuran popsize pada penelitian ini, dibutuhkan untuk mengetahui pengaruh jumlah popsize pada nilai *fitness*. Tabel uji coba pengujian jumlah popsize ditampilkan pada Tabel 3.2.

Tabel 3.2 Pengujian Popsiz

Generasi	Jumlah Popsiz									
	10	20	30	40	50	60	70	80	90	100
5										
10										
15										
20										
25										
30										
35										
40										
45										
50										
Waktu Komputasi										

2. Pengujian nilai kombinasi *cr* dan *mr*

Pengujian nilai *cr* (*crossover rate*) dan *mr* (*mutation rate*) pada penelitian ini, dibutuhkan untuk mengetahui pengaruh banyaknya *offspring* yang dihasilkan pada nilai *fitness*. Tabel uji coba pengujian kombinasi *cr* dan *mr* ditampilkan pada Tabel 3.3.

Tabel 3.3 Pengujian Cr dan Mr

Generasi	Kombinasi Cr : Mr									
	0.1: 1	0.2: 0.9	0.3: 0.8	0.4: 0.7	0.5: 0.6	0.6: 0.5	0.7: 0.4	0.8: 0.3	0.9: 0.2	1: 0.1
5										
10										
15										
20										
25										
30										
35										
40										
45										
50										
Waktu Komputasi										

3. Pengujian tingkat akurasi

Pengujian tingkat akurasi pada penelitian ini, dibutuhkan untuk mengetahui ketepatan hasil prediksi yang diperoleh dari individu terbaik menggunakan parameter algoritme genetika yang optimal. Tabel uji coba pengujian tingkat akurasi ditampilkan pada Tabel 3.4.

Tabel 3.4 Pengujian Tingkat Akurasi

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
...
Jumlah					
RMSPE					
Akurasi					

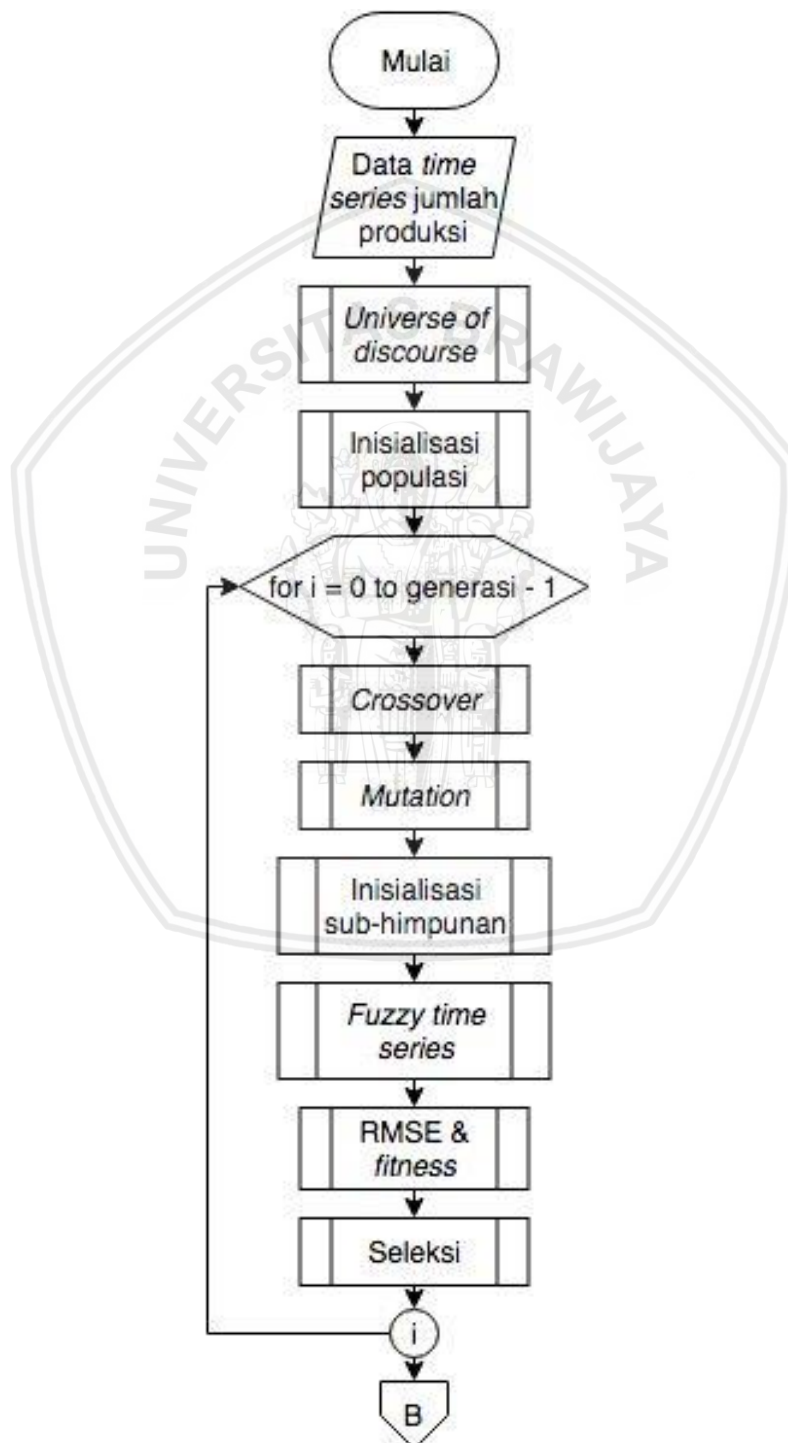
3.6 Analisis Hasil

Tahapan akhir yang dilakukan pada penelitian ini yaitu analisis hasil, dimana setelah berbagai pengujian yang telah dilakukan mendapatkan hasil dari penelitian berupa kesimpulan, dan saran untuk penelitian selanjutnya.



BAB 4 ALGORITME

Bab ini berisikan tahapan perhitungan manual yang dilakukan dalam optimasi *fuzzy time series* untuk prediksi jumlah produksi SAGA *leather fashion* menggunakan metode algoritme genetika. Tahapan yang dilakukan dalam proses perhitungan manual penelitian ini, ditampilkan dalam flowchart pada Gambar 4.1.





Gambar 4.1 Diagram Alir Optimasi *Fuzzy Time Series* dengan Algoritme Genetika

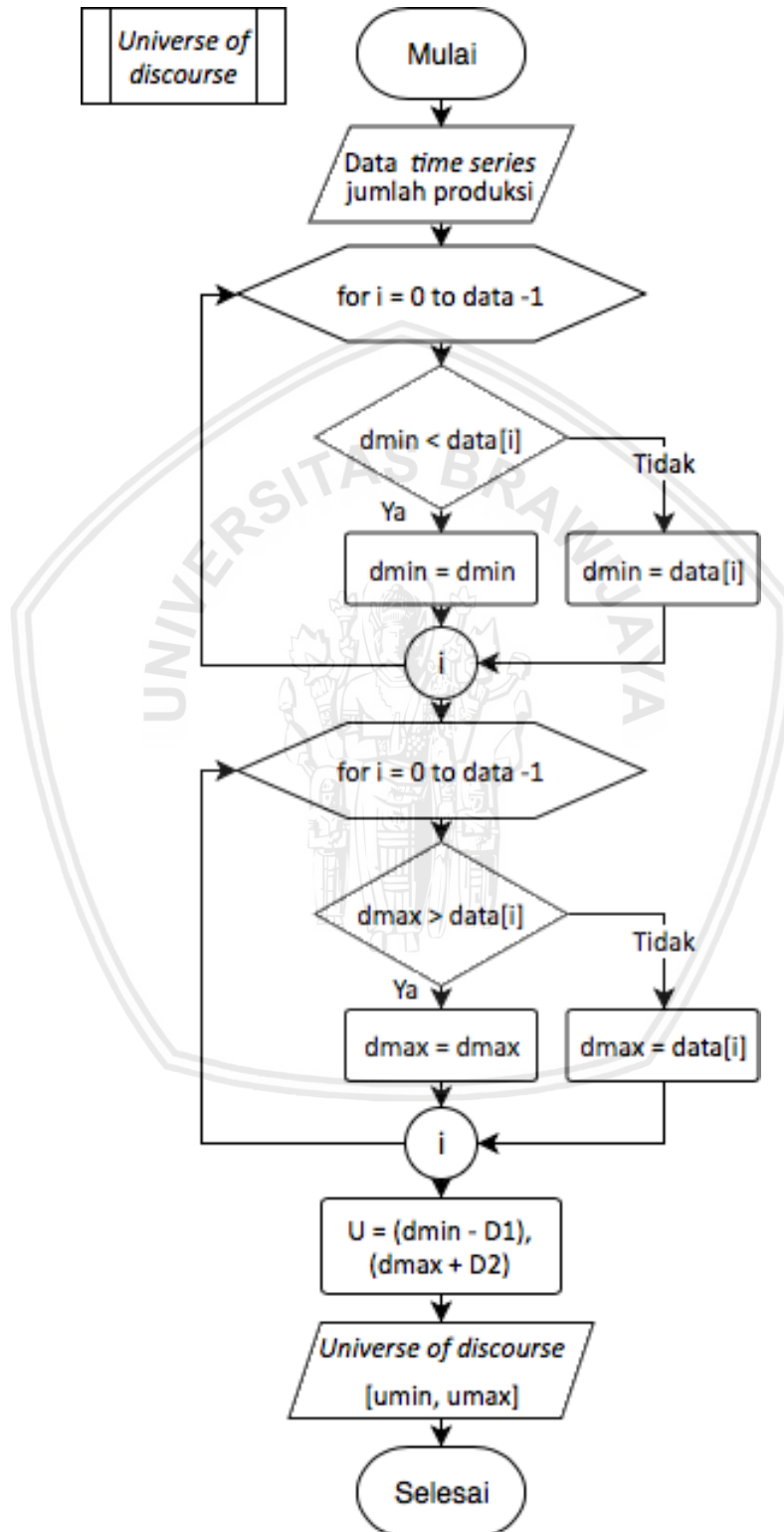
Hal pertama yang dilakukan dalam proses optimasi *fuzzy time series* dengan algoritme genetika yaitu menentukan *universe of discourse* dari data *time series*, dengan rentang waktu dari Januari 2016 sampai Desember 2017. Setelah didapatkan *universe of discourse*, selanjutnya inisialisasi populasi dengan dibangkitkan nilai individu secara acak dalam interval *universe of discourse*. Kemudian individu-individu tersebut melalui tahap reproduksi, berupa *crossover* dan *mutation*. Setelah melalui tahap reproduksi dan menghasilkan individu-individu baru, kemudian masuk ke tahap *fuzzy time series* yang didalamnya terdapat proses *fuzzyfikasi*, *Fuzzy Logical Relationship (FLR)*, *Fuzzy Logical Relationship Group (FLRG)* dan *defuzzyfikasi*. Dari proses-proses tersebut kemudian didapatkan hasil prediksi yang kemudian ditentukan hasil *error* yang dihasilkan dengan menggunakan metode *Root Mean Square Error (RMSE)*, dan menghitung *fitness* dari nilai RMSE yang dihasilkan.

4.1 Universe of Discourse

Tahapan-tahapan dalam proses *universe of discourse* pada Gambar 4.2 dapat diuraikan sebagai berikut:

1. Mendapatkan data *time series* jumlah produksi SAGA *leather fashion*.
2. Melakukan perulangan [i] sampai dengan banyak data yang telah didapatkan.
3. Memberikan kondisi apabila nilai variabel *dmin* kurang dari data[i] maka nilai *dmin* tetap, sedangkan jika tidak maka nilai *dmin* sama dengan nilai data[i], kemudian keluar dari perulangan.
4. Melakukan perulangan [i] sampai dengan banyak data yang telah didapatkan.
5. Memberikan kondisi apabila nilai variabel *dmax* lebih dari data[i] maka nilai *dmax* tetap, sedangkan jika tidak maka nilai *dmax* sama dengan nilai data[i], kemudian keluar dari perulangan.
6. Mendapatkan nilai *universe of discourse* dengan persamaan $U = [dmin - D1], [dmax + D2]$, dimana nilai D1 dan D2 dapat ditentukan.

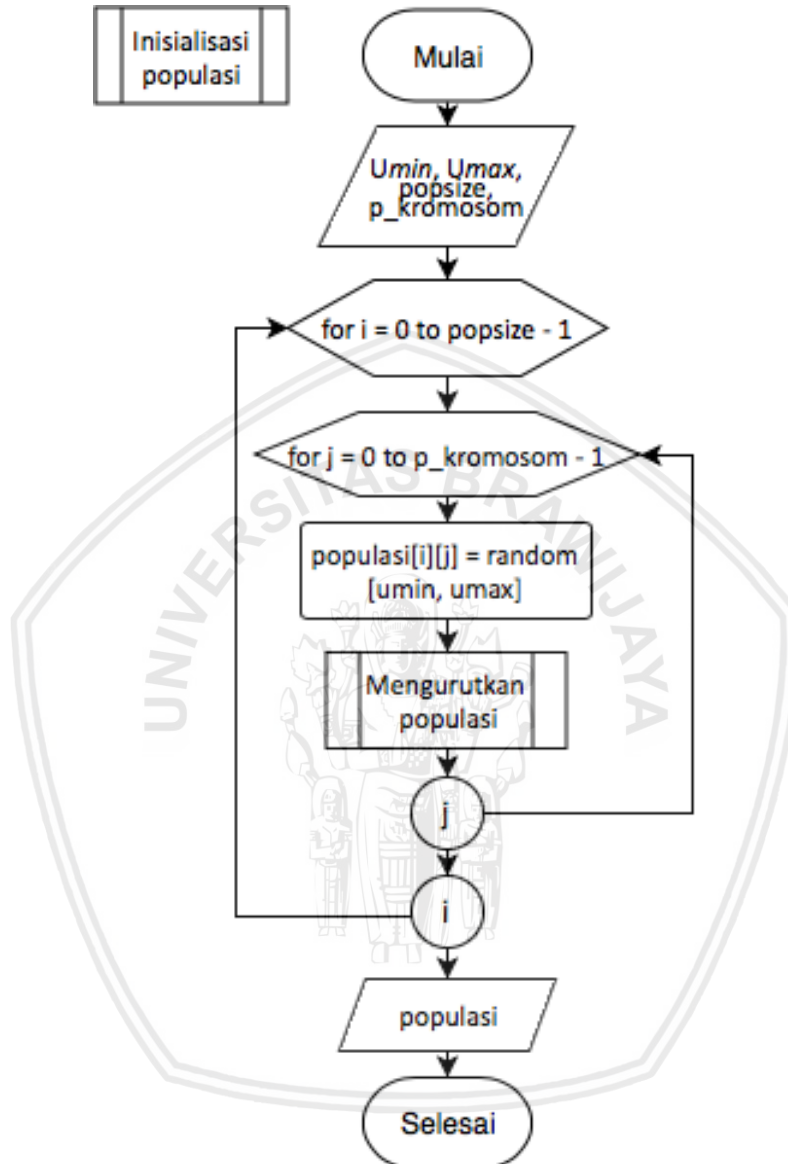
7. Setelah melakukan berbagai proses diatas, kemudian didapatkan nilai umin sebagai batas bawah dan umax sebagai batas atas dalam *universe of discourse*.



Gambar 4.2 Diagram Alir *Universe of Discourse*

4.2 Inisialisasi Populasi

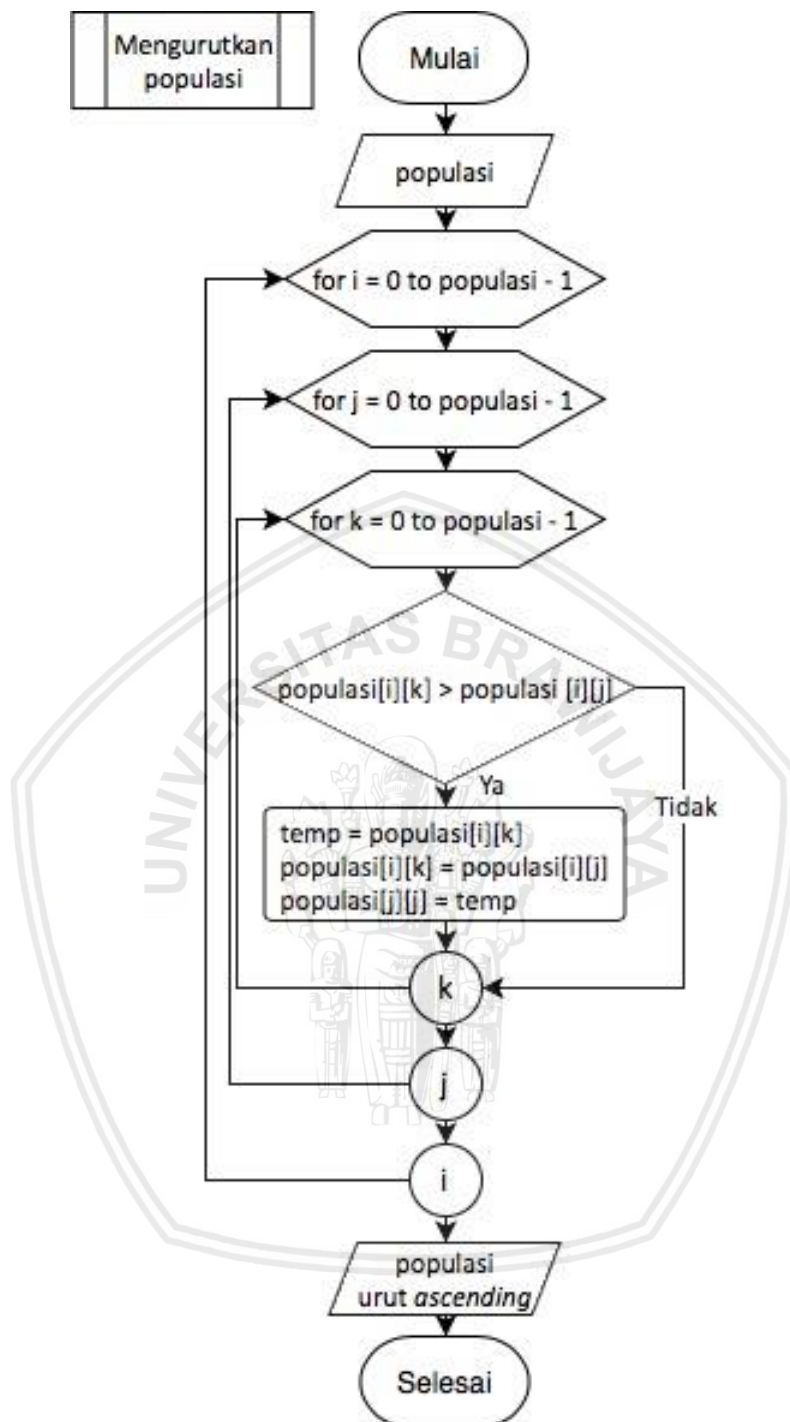
Tahapan-tahapan dalam proses inisialisasi populasi ditampilkan pada Gambar 4.3.



Gambar 4.3 Diagram Alir Inisialisasi Populasi

Penjelasan pada Gambar 4.3 yaitu sebagai berikut:

1. Mendapatkan jumlah populasi awal (popsize), panjang kromosom ($p_kromosom$) dan *universe of discourse*.
2. Melakukan perulangan [i] sampai dengan popsize yang telah didapatkan.
3. Melakukan perulangan [j] sampai dengan $p_kromosom$ yang telah didapatkan.
4. Membangkitkan variabel populasi secara acak dalam interval umin dan umax.
5. Setelah nilai populasi dibangkitkan secara acak, kemudian nilai tersebut diurutkan dari nilai terkecil ke terbesar. Proses mengurutkan ditampilkan pada Gambar 4.4.



Gambar 4.4 Diagram Alir Urut Populasi

Tahapan-tahapan dalam proses mengurutkan populasi pada Gambar 4.4 dapat diuraikan sebagai berikut:

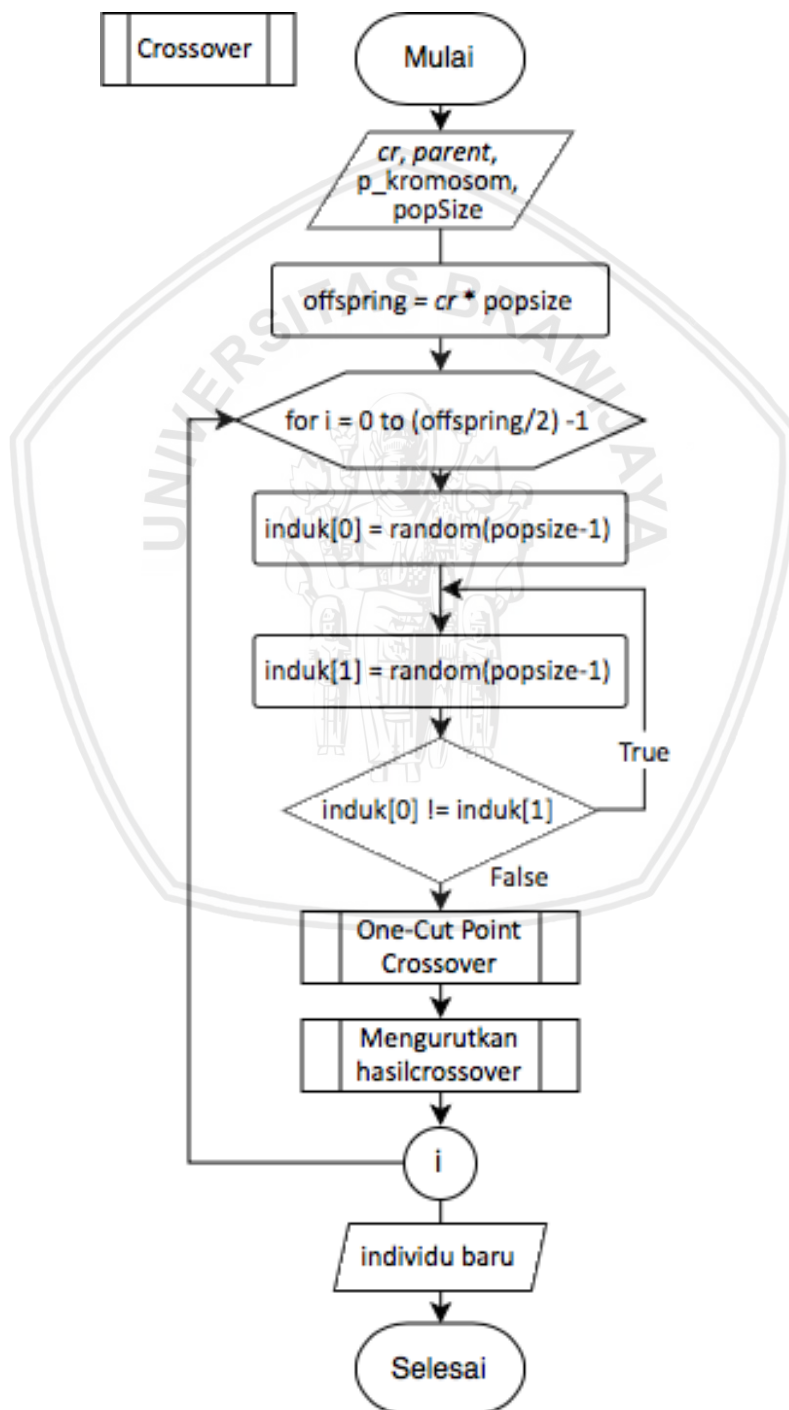
1. Mendapatkan nilai populasi.
2. Melakukan tiga perulangan sampai dengan populasi dengan variabel [i], [j] dan [k].
3. Memberikan kondisi apabila nilai populasi[i][k] lebih dari populasi [i][j] maka nilai variabel temp sama dengan populasi[i][k], populasi[i][k] sama dengan

populasi[i][j] dan populasi[i][j] sama dengan temp, jika tidak maka masuk kembali ke perulangan.

- Setelah melakukan proses diatas, kemudian didapatkan nilai populasi yang urut dari nilai terkecil ke nilai terbesar.

4.3 Crossover

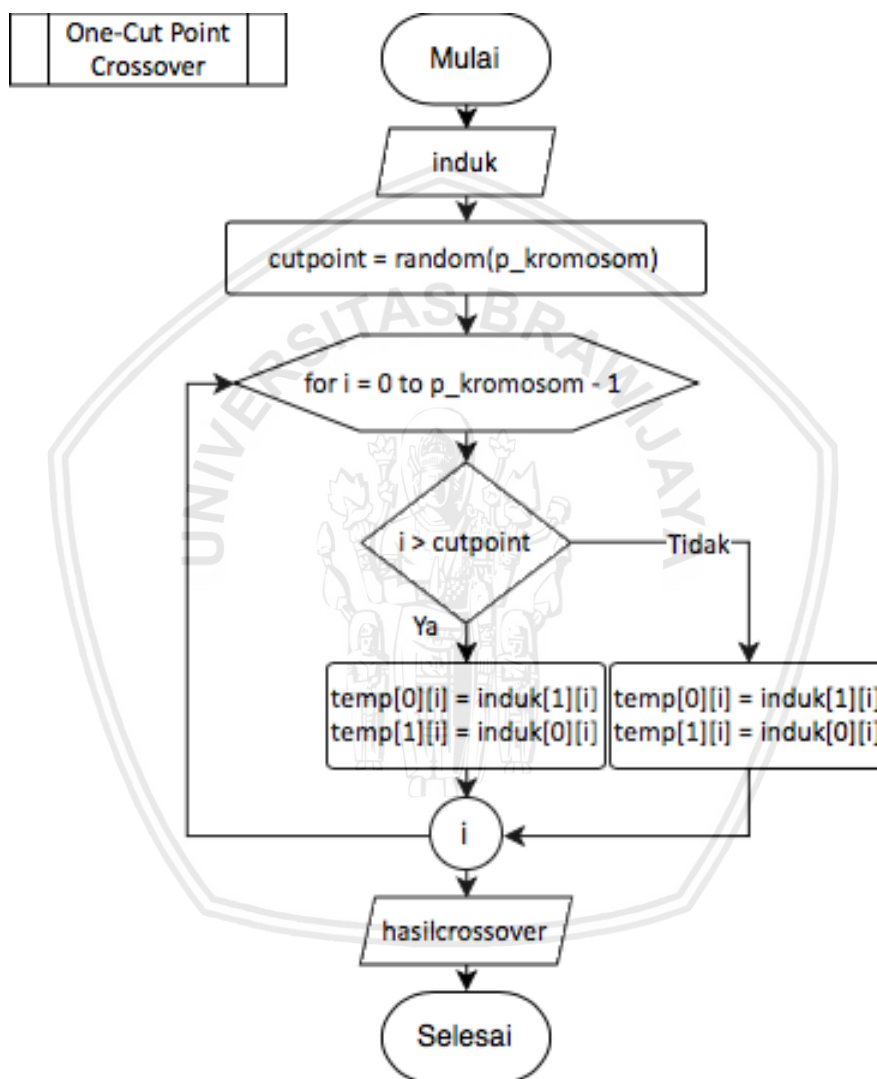
Tahapan-tahapan dalam proses *crossover* ditampilkan pada Gambar 4.5.



Gambar 4.5 Diagram Alir *Crossover*

Penjelasan pada Gambar 4.5 yaitu sebagai berikut:

1. Menginputkan *crossover rate* (*cr*) dengan interval 0 sampai 1.
2. Menghitung *offspring* (individu yang dihasilkan dalam proses reproduksi) dengan persamaan *cr* dikalikan dengan *popsiz*.
3. Melakukan perulangan [*i*] sampai dengan jumlah *offspring* yang didapatkan.
4. Menentukan induk yang akan mengalami proses *crossover*, kemudian masuk ke proses *one-cut point crossover* yang ditampilkan pada Gambar 4.6.



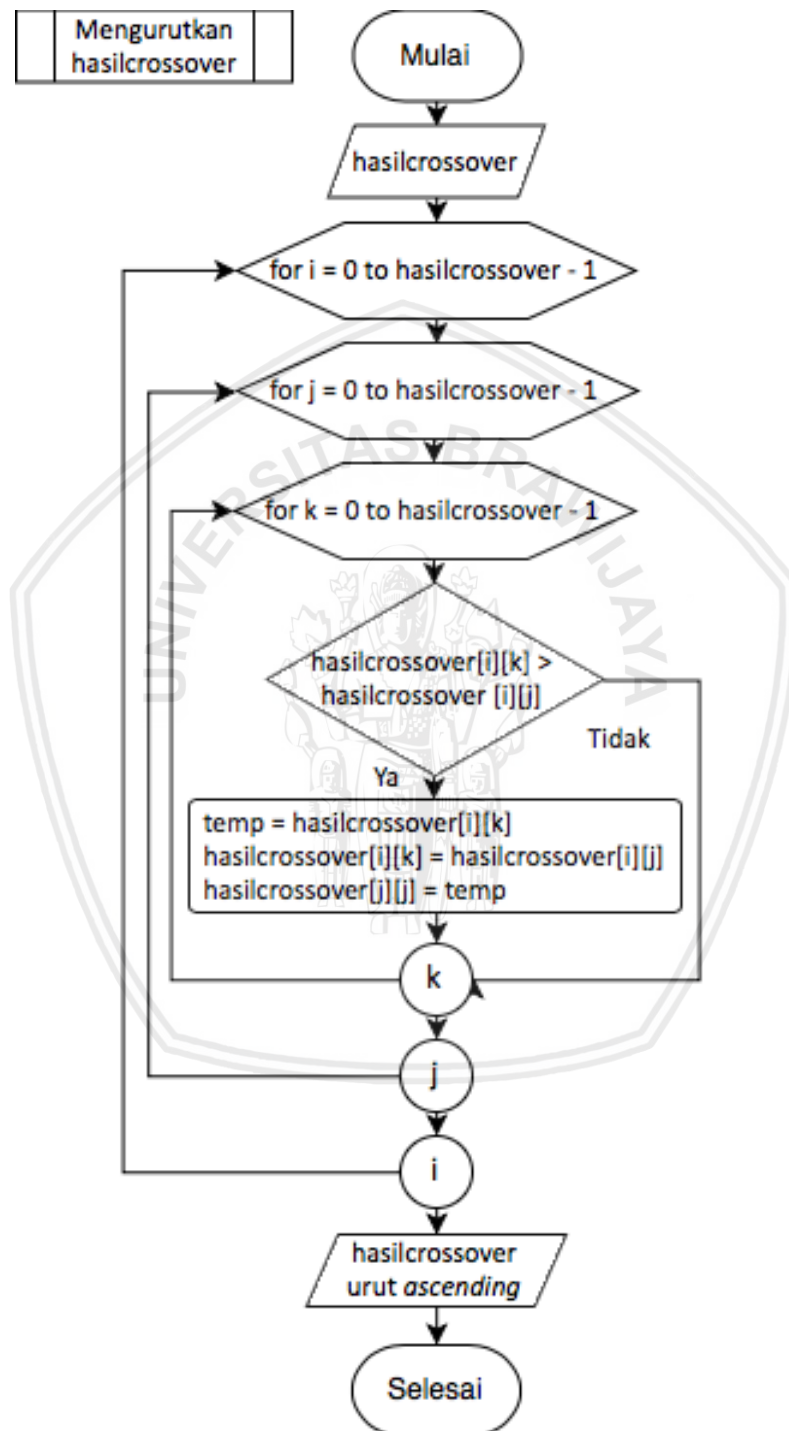
Gambar 4.6 Diagram Alir One-Cut Point Crossover

Tahapan-tahapan dalam proses *one-cut point crossover* pada Gambar 4.6 dapat diuraikan sebagai berikut:

1. Mendapatkan induk yang akan melalui proses *crossover*.
2. Menentukan *cutpoint* secara acak.
3. Melakukan perulangan [*i*] sampai dengan *p_kromosom*.
4. Menukarkan nilai antar induk sesuai dengan *cutpoint* yang telah ditentukan.



- Setelah melakukan berbagai proses diatas, kemudian didapatkan hasil *crossover*, yang kemudian hasil *crossover* tersebut diurutkan dari nilai terkecil ke terbesar. Proses mengurutkan hasil *crossover* ditampilkan pada Gambar 4.7.



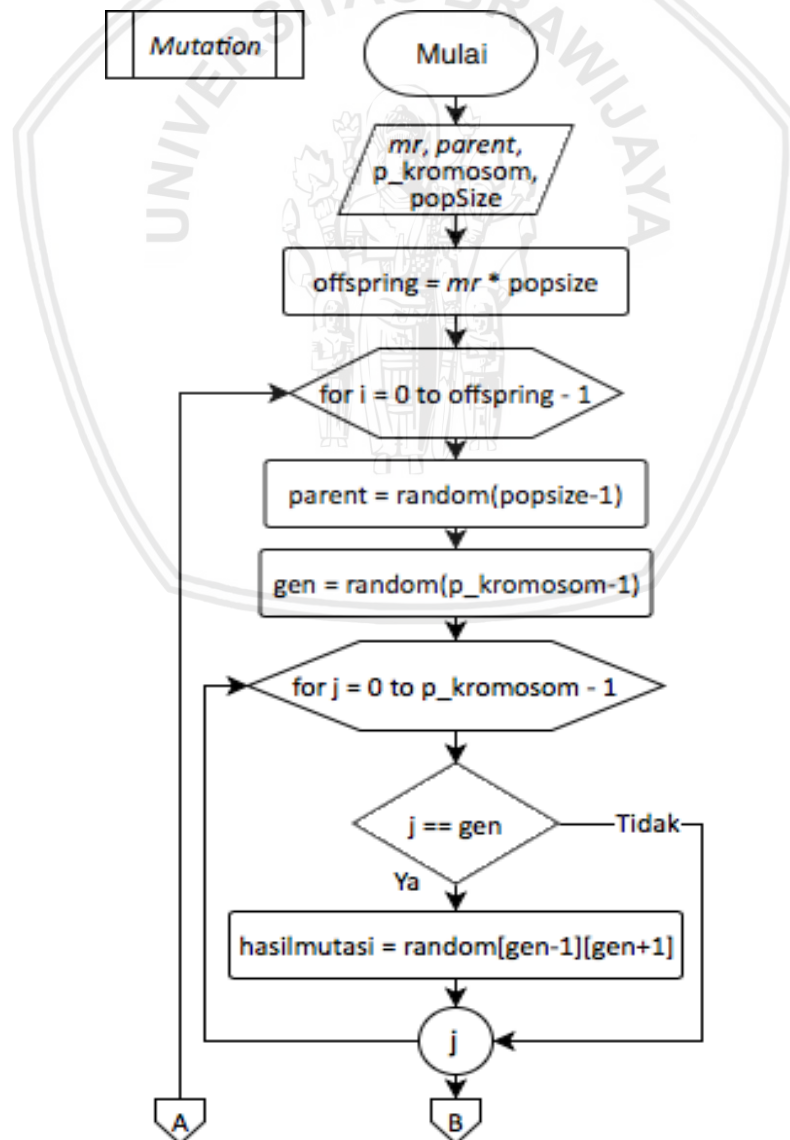
Gambar 4.7 Diagram Alir Urut *Crossover*

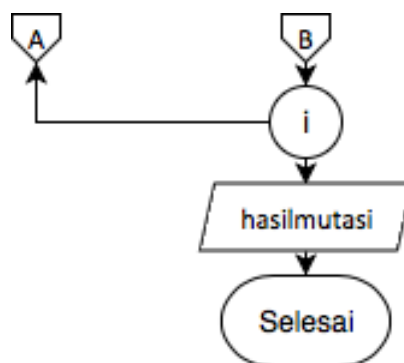
Tahapan-tahapan dalam proses mengurutkan hasil *crossover* pada Gambar 4.7 dapat diuraikan sebagai berikut:

1. Mendapatkan hasil *crossover*.
2. Melakukan tiga perulangan sampai dengan hasil *crossover* dengan variabel [i], [j] dan [k].
3. Memberikan kondisi apabila nilai hasil*crossover*[i][k] lebih dari hasil*crossover*[i][j] maka nilai variabel temp sama dengan hasil*crossover*[i][k], hasil*crossover*[i][k] sama dengan hasil*crossover*[i][j] dan hasil*crossover*[i][j] sama dengan temp, jika tidak maka masuk kembali ke perulangan.
4. Setelah melakukan proses diatas, kemudian didapatkan nilai hasil *crossover* yang urut dari nilai terkecil ke nilai terbesar.

4.4 Mutation

Tahapan-tahapan dalam proses *mutation* ditampilkan pada Gambar 4.8.





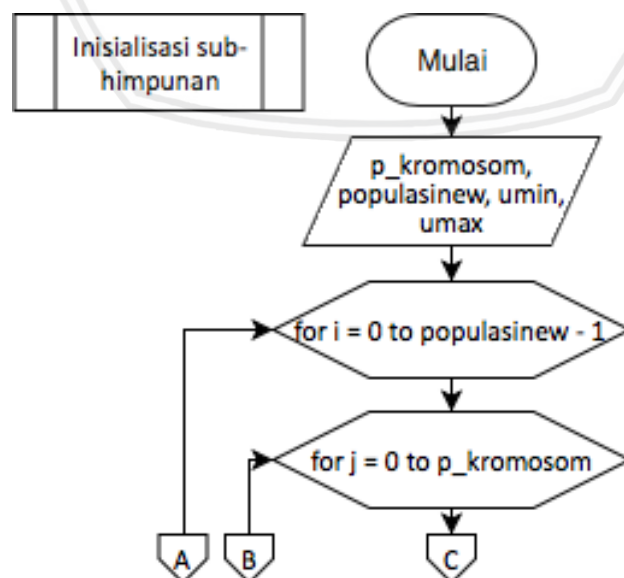
Gambar 4.8 Diagram Alir Mutation

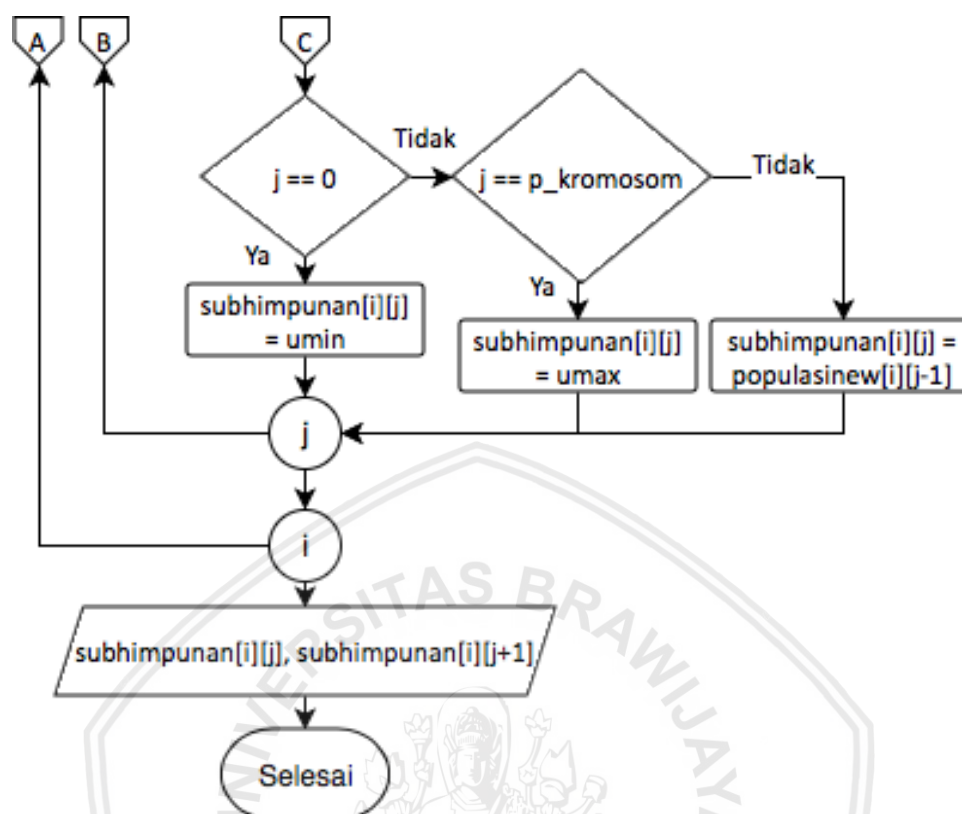
Penjelasan pada Gambar 4.8 yaitu sebagai berikut:

1. Menginputkan *mutation rate* (*mr*) dengan interval 0 sampai 1.
2. Menghitung *offspring* (individu yang dihasilkan dalam proses reproduksi) dengan persamaan *mr* dikalikan dengan *popSize*.
3. Melakukan perulangan [*i*] sampai dengan jumlah *offspring* yang didapatkan.
4. Menentukan induk dan gen yang akan mengalami proses *mutation* secara acak.
5. Melakukan perulangan [*j*] sampai dengan *p_kromosom*.
6. Mendapatkan hasil *mutation* secara acak dalam interval nilai gen sebelumnya sebagai batas bawah dan nilai gen setelahnya sebagai batas atas.
7. Setelah melakukan berbagai proses diatas, kemudian didapatkan hasil *mutation*.

4.5 Inisialisasi Sub-Himpunan

Tahapan-tahapan dalam proses inisialisasi sub-himpunan ditampilkan pada Gambar 4.9.





Gambar 4.9 Diagram Alir Sub-Himpunan

Penjelasan pada Gambar 4.9 yaitu sebagai berikut:

1. Mendapatkan $p_kromosom$, $populasinew$ (populasi baru setelah proses reproduksi), $umin$ dan $umax$ dari proses sebelumnya.
2. Melakukan perulangan $[i]$ sampai dengan jumlah $populasinew$ dan perulangan $[j]$ sampai dengan $p_kromosom$.
3. Memberikan kondisi jika nilai j sama dengan 0 maka subhimpunan sama dengan $umin$, sedangkan jika nilai j sama dengan nilai maksimal $p_kromosom$ maka subhimpunan sama dengan $umax$, dan jika tidak maka sub-himpunan sama dengan $populasinew[j + 1]$.
4. Setelah melakukan berbagai proses diatas, kemudian didapatkan sub-himpunan yang direpresentasikan sebagai u_1, u_2, \dots, u_j .

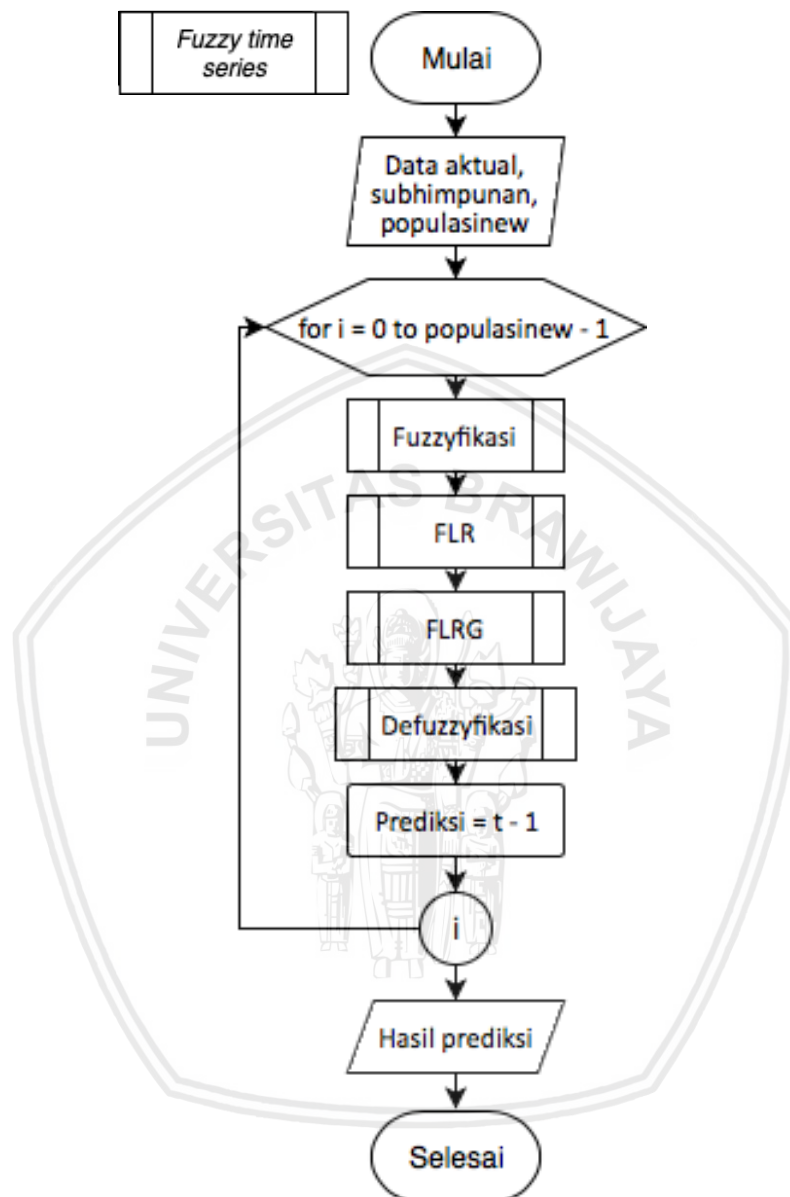
4.6 Fuzzy Time Series

Tahapan-tahapan dalam proses *fuzzy time series* pada Gambar 4.10 dapat diuraikan sebagai berikut:

1. Mendapatkan data aktual, sub-himpunan dan populasi dari proses sebelumnya.
2. Melakukan proses *fuzzyfikasi*, FLR, FLRG, *defuzzyfikasi* dan kemudian proses prediksi dengan persamaan $t - 1$, dimana t sebagai bulan yang diprediksi.



3. Melakukan perulangan sampai dengan populasi yang didapatkan.
4. Setelah melakukan berbagai proses diatas, kemudian didapatkan hasil prediksi.



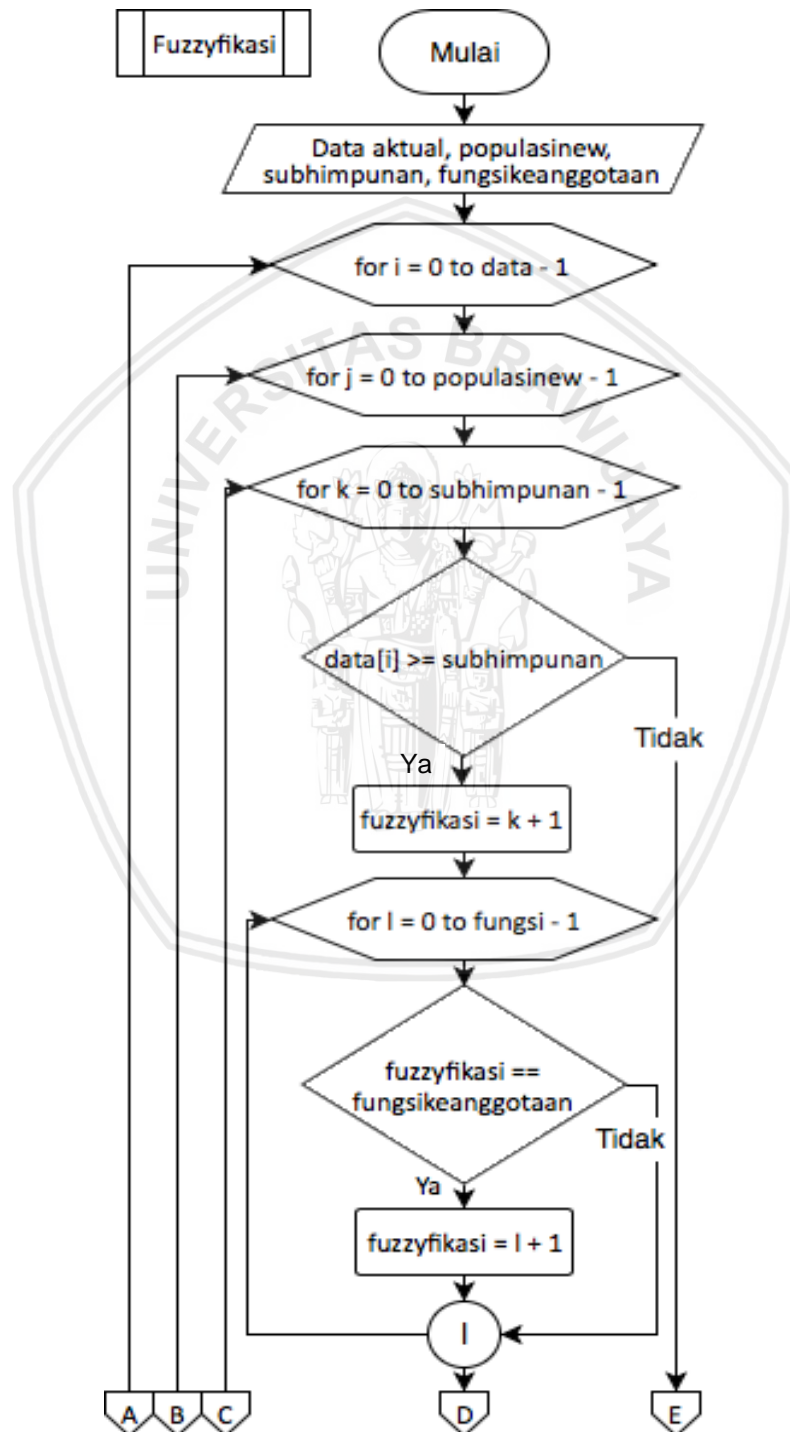
Gambar 4.10 Diagram Alir *Fuzzy Time Series*

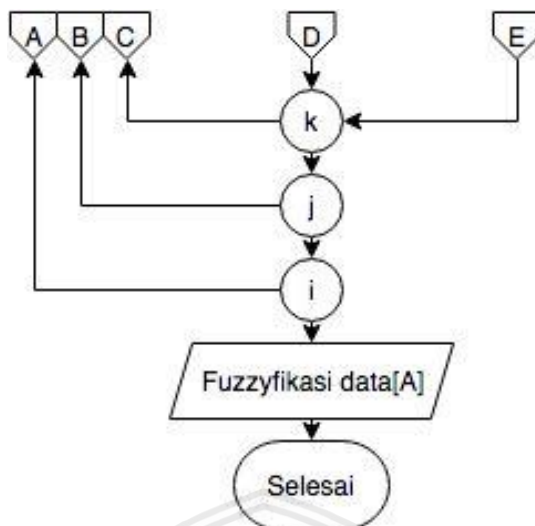
4.7 Fuzzyfikasi

Tahapan-tahapan dalam proses *fuzzyfikasi* seperti pada Gambar 4.11, dapat diuraikan sebagai berikut:

1. Mendapatkan data aktual, subhimpunan, populasi dan fungsi keanggotaan dari proses sebelumnya.
2. Melakukan perulangan [i] sampai dengan populasinew, perulangan [j] sampai dengan subhimpunan dan perulangan [k] sampai dengan fungsi.

3. Memberikan kondisi apabila $data[i]$ lebih dari atau sama dengan subhimpunan $[j]$, maka *fuzzyfikasi* sama dengan $[j] + 1$, jika tidak maka masuk kembali ke perulangan $[j]$.
4. Apabila *fuzzyfikasi* sama dengan fungsikeanggotaan, maka *fuzzyfikasi* sama dengan $[k] + 1$, jika tidak maka masuk kembali ke perulangan $[k]$.
5. Setelah melakukan berbagai proses diatas, kemudian didapatkan hasil data *fuzzyfikasi* yang direpresentasikan sebagai A.



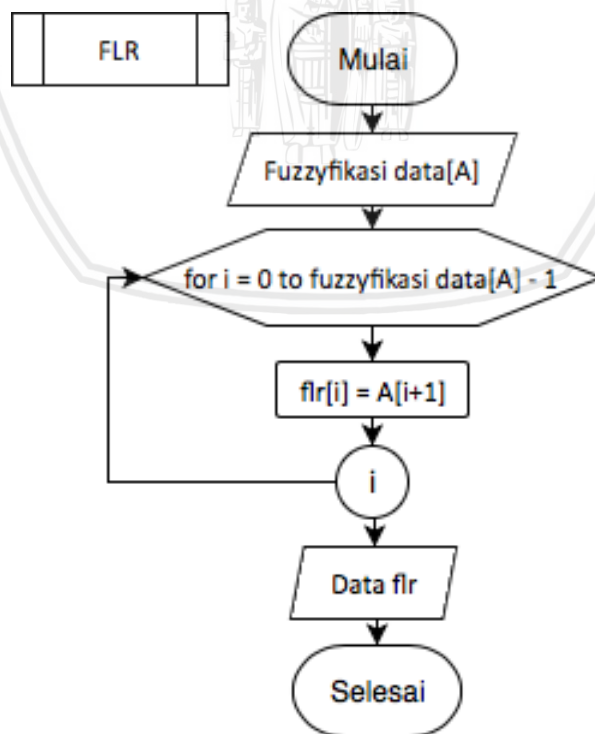


Gambar 4.11 Diagram Alir Fuzzyfikasi

4.8 FLR

Tahapan-tahapan dalam proses FLR (*Fuzzy Logical Relationship*) seperti pada Gambar 4.12, dapat diuraikan sebagai berikut:

1. Mendapatkan data *fuzzyfikasi* dari proses sebelumnya.
2. Melakukan perulangan sampai dengan *fuzzyfikasi*.
3. Menginisialisasi variabel flr sama dengan *fuzzyfikasi*[i + 1].
4. Setelah melakukan berbagai proses diatas, kemudian didapatkan hasil berupa data flr.

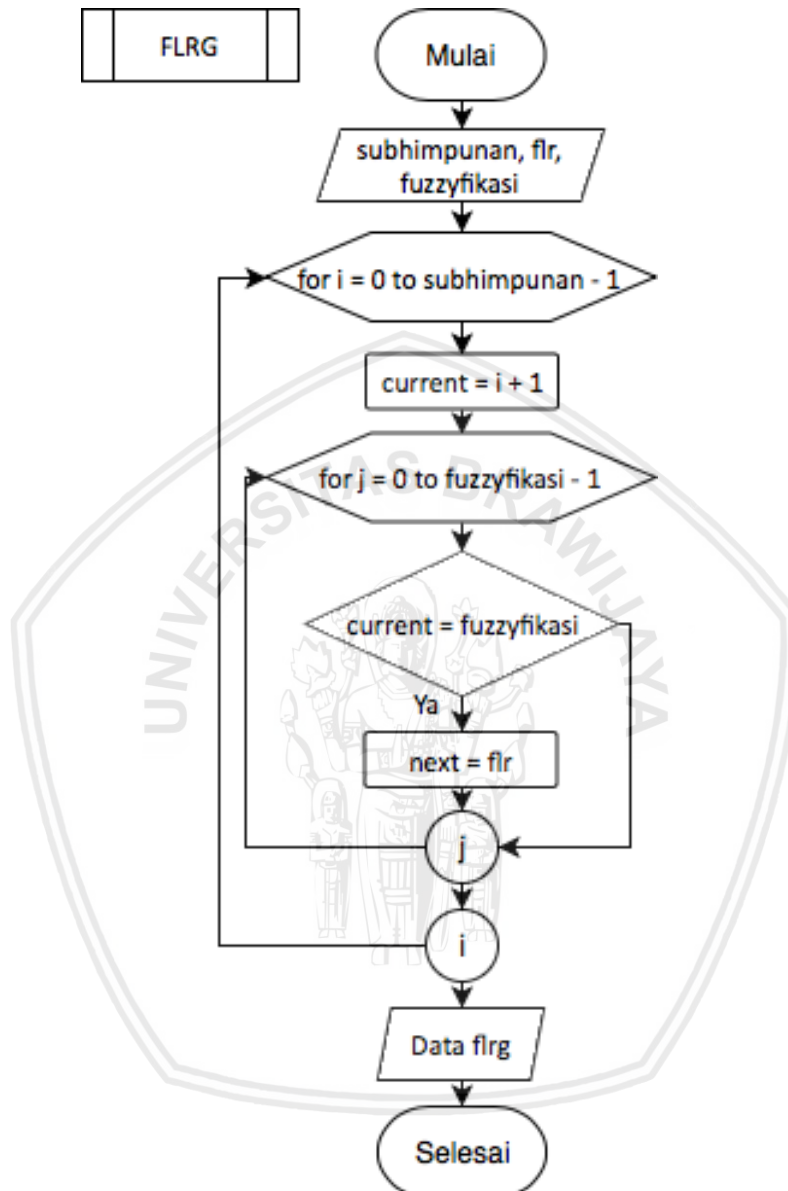


Gambar 4.12 Diagram Alir FLR



4.9 FLRG

Tahapan-tahapan dalam proses FLRG (*Fuzzy Logical Relationship Group*) ditampilkan pada Gambar 4.13.



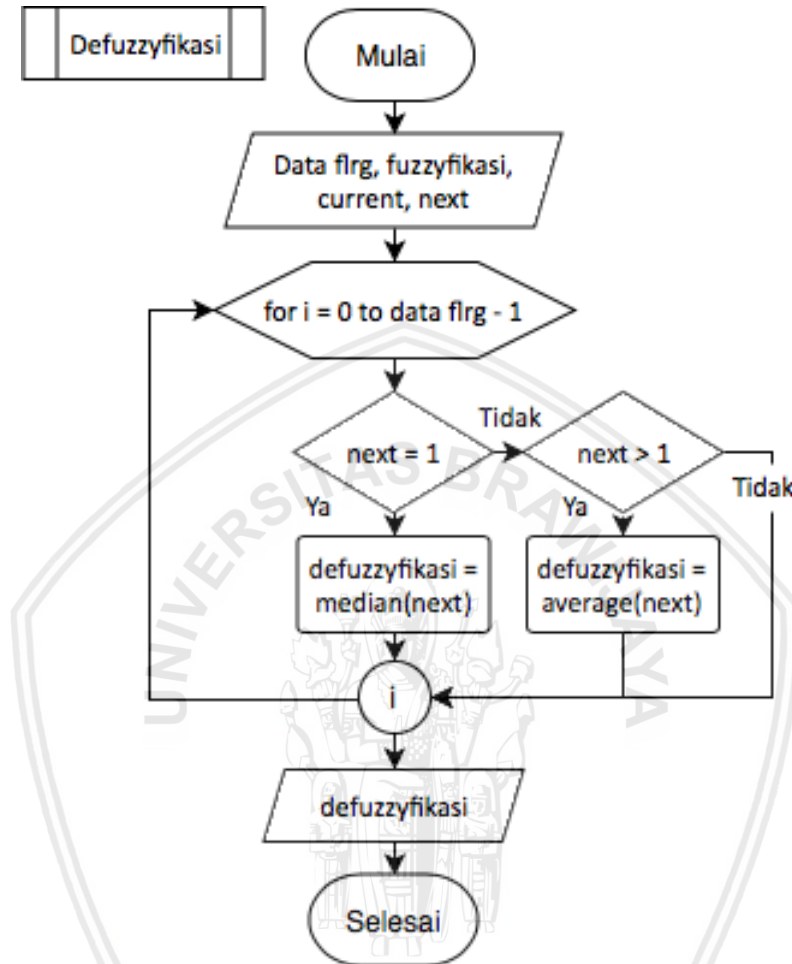
Gambar 4.13 Diagram Alir FLRG

Penjelasan pada Gambar 4.13 yaitu sebagai berikut:

1. Mendapatkan data subhimpunan, flr dan *fuzzyfikasi* dari proses sebelumnya.
2. Melakukan perulangan [i] sampai subhimpunan.
3. Menginisialisasi variabel *current* sama dengan $i + 1$.
4. Melakukan perulangan [j] sampai dengan *fuzzyfikasi*.
5. Memberikan kondisi apabila nilai *current* sama dengan *fuzzyfikasi* maka variabel *next* sama dengan flr, jika tidak maka kembali ke perulangan [j].
6. Setelah melakukan berbagai proses diatas, kemudian didapatkan hasil berupa data flrg.

4.10 Defuzzyfikasi

Tahapan-tahapan dalam proses *defuzzyfikasi* ditampilkan pada Gambar 4.14.



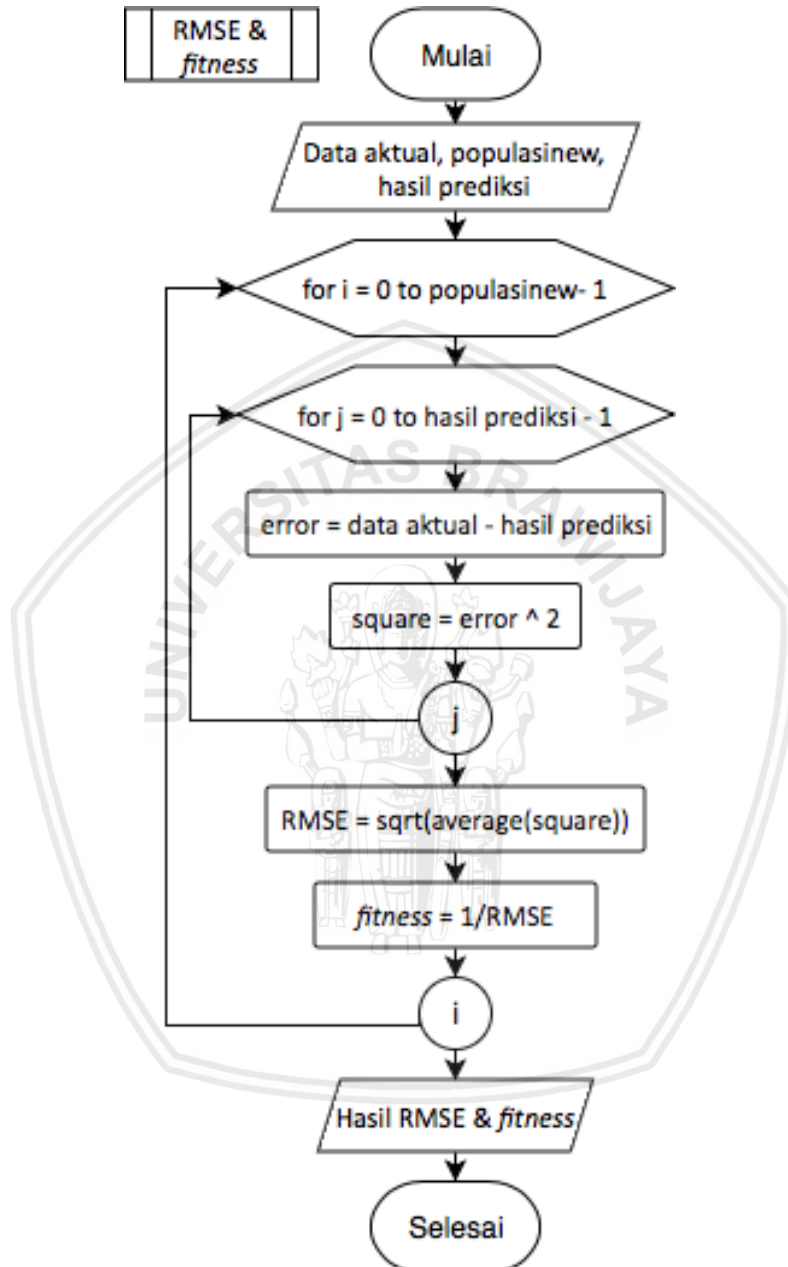
Gambar 4.14 Diagram Alir *Defuzzyfikasi*

Penjelasan pada Gambar 4.14 yaitu sebagai berikut:

1. Mendapatkan data *flrg*, *fuzzyfikasi*, *current*, dan *next* dari proses sebelumnya.
2. Melakukan perulangan [i] sampai dengan data *flrg*.
3. Memberi kondisi apabila terdapat hanya ada satu variabel *next* pada variabel *current* maka nilai *defuzzyfikasi* sama dengan nilai tengah dari variabel *next* tersebut.
4. Memberi kondisi apabila terdapat lebih dari satu variabel *next* pada variabel *current* maka nilai *defuzzyfikasi* sama dengan rata-rata dari variabel-variabel *next* tersebut, jika tidak maka masuk kembali ke perulangan [i].

4.11 RMSE & Fitness

Tahapan-tahapan proses RMSE & *fitness* ditampilkan pada Gambar 4.15.



Gambar 4.15 Diagram Alir RMSE & *Fitness*

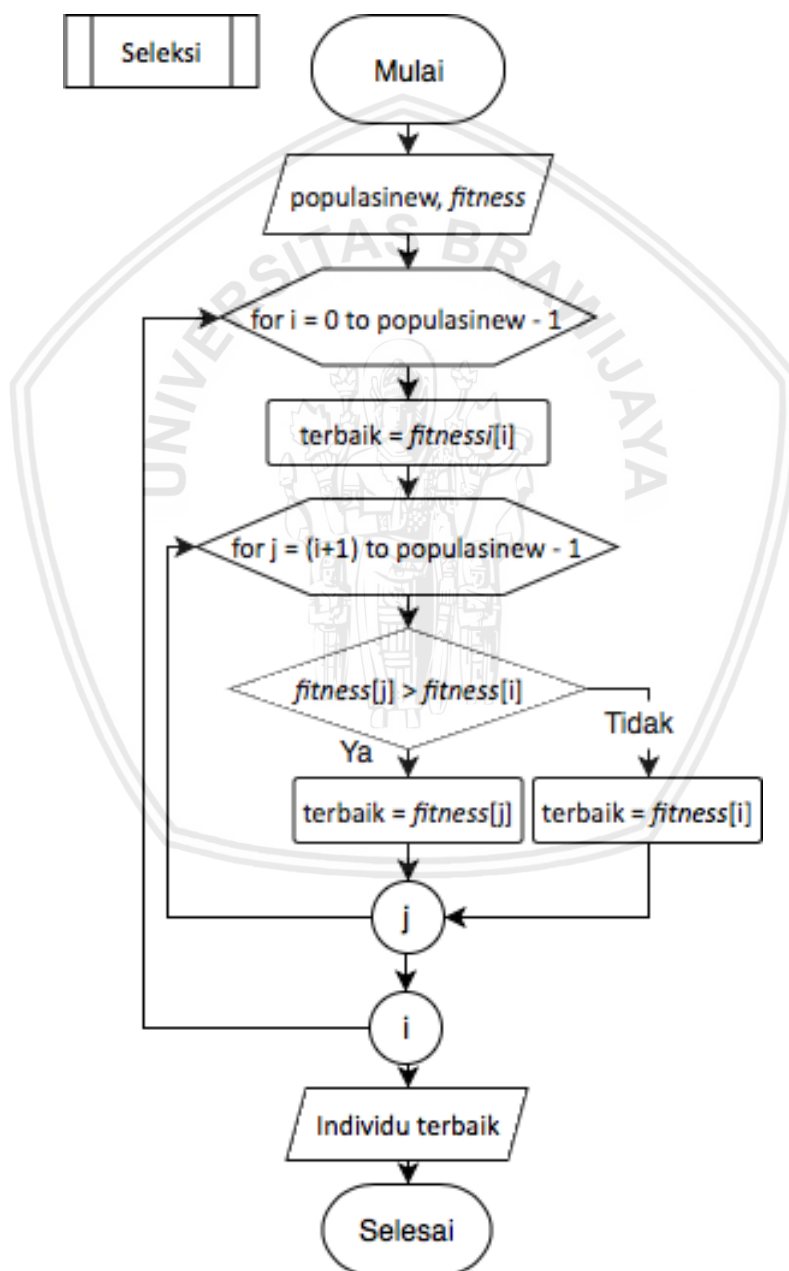
Penjelasan pada Gambar 4.15 yaitu sebagai berikut

1. Mendapatkan data aktual, populasinew dan hasil prediksi.
2. Melakukan perulangan [i] sampai dengan populasi new.
3. Melakukan perulangan [j] sampai dengan prediksi.
4. Menghitung nilai *error* dengan persamaan data aktual – hasil prediksi.
5. Menghitung mengkuadratkan hasil *error* yang telah diperoleh.

6. Menghitung RMSE dengan persamaan mengakarkan hasil rata-rata dari hasil kuadrat yang telah diperoleh.
7. Menghitung nilai *fitness* dengan persamaan $1/\text{RMSE}$.
8. Setelah melakukan berbagai proses diatas, didapatkan hasil berupa nilai RMSE & *fitness*.

4.12 Seleksi

Tahapan-tahapan dalam proses seleksi ditampilkan pada Gambar 4.13.



Gambar 4.16 Diagram Alir Seleksi

Penjelasan pada Gambar 4.16 yaitu sebagai berikut:

1. Mendapatkan populasi dan hasil *fitness* dari proses sebelumnya.
2. Melakukan perulangan [i] sampai dengan populasinew.
3. Membangkitkan variabel terbaik = *fitness*[i].
4. Melakukan perulangan [j] sampai dengan populasinew.
5. Memberi kondisi apabila *fitness*[j] lebih dari *fitness*[i], maka variabel terbaik = *fitness*[j] dan apabila tidak maka variabel terbaik = *fitness*[i].
6. Setelah melakukan berbagai proses diatas, kemudian didapatkan hasil berupa individu terbaik.

4.13 Perhitungan Manual

Pada perhitungan manual berikut, diberikan contoh perhitungan menggunakan jenis produk sabuk kulit sapi dengan parameter algoritme genetika sebagai berikut:

Ukuran populasi awal	=	3
Panjang kromosom	=	7
Banyak generasi	=	1
<i>Crossover rate</i>	=	0,7
<i>Mutation rate</i>	=	0,4

Tahap awal yang dilakukan yaitu menentukan *universe of discourse* dengan persamaan $U = [Dmin - D1], [Dmax + D2]$, yang kemudian mendapatkan hasil *Umin* sebagai batas bawah dan *Umax* sebagai batas atas pada *universe of discourse*. Untuk nilai *Dmin* dan *Dmax* masing-masing 53 dan 83, sedangkan untuk nilai *D1* dan *D2* masing-masing 3 dan 2. Kemudian didapatkan *universe of discourse* yaitu *Umin* = 50 dan *Umax* 85.

4.13.1 Inisialisasi Populasi

Tahapan selanjutnya yaitu inisialisasi populasi awal dimana jumlah populasi telah ditentukan sebelumnya dengan membangkitkan nilai kromosom secara acak, dengan nilai kromosom didapatkan dalam interval *universe of discourse* yang telah didapatkan pada proses sebelumnya. Pada Tabel 4.1 ditampilkan populasi awal dari produk sabuk kulit sapi.

Tabel 4.1 Inisialisasi Populasi Awal

Individu	Kromosom						
P1	52	59	66	70	77	79	80
P2	52	53	57	58	59	74	76
P3	51	52	64	67	75	77	79

Nilai kromosom yang didapatkan dari interval *universe of discourse* diurutkan dari nilai terkecil ke nilai terbesar. Dari nilai-nilai kromosom tersebut akan digunakan menjadi pembatas dalam membentuk sub-himpunan.



4.13.2 Crossover

Tahapan selanjutnya yaitu proses *crossover*, dimana dalam penelitian ini menggunakan teknik *one cut-point crossover*. Untuk *offspring* yang dihasilkan didapatkan dengan $cr * \text{popsize}$ yaitu $0,7 * 3 = 2,1$ yang hasilnya dibulatkan menjadi 2.

Tabel 4.2 Parent Crossover

Individu	Kromosom						
P2	52	53	57	58	59	74	76
P3	51	52	64	67	75	77	79

Cut | Point

Pada Tabel 4.2 dipilih P2 dan P3 sebagai *parent* yang akan di *crossover*, dimana *cutpoint* berada pada titik diantara gen ke-4 dan ke-5.

Tabel 4.3 Child Crossover

Individu	Kromosom						
C1	52	53	57	58	75	77	79
C2	51	52	59	64	67	74	76

Cut | Point

Setelah melalui proses *crossover* dimana nilai gen ke-6 sampai ke-8 ditukarkan nilainya antara 2 *parent* yang telah terpilih sebelumnya. Kemudian setelah di *crossover* nilai tersebut diurutkan dari yang terkecil ke nilai terbesar. Seperti pada Tabel 4.3, kedua *parent* tersebut menghasilkan 2 *child* yang sesuai dengan persamaan *offspring* yang telah diperhitungkan sebelumnya.

4.13.3 Mutation

Tahapan selanjutnya yaitu proses *mutation*, dimana dalam penelitian ini menggunakan teknik *random mutation*. Untuk *offspring* yang dihasilkan didapatkan dengan $mr * \text{popsize}$ yaitu $0,2 * 3 = 0,6$ yang hasilnya dibulatkan menjadi 1.

Tabel 4.4 Parent Mutation

Individu	Kromosom						
P1	52	59	66	70	77	79	80

Pada Tabel 4.4, dipilih P1 sebagai *parent* yang akan di *mutation*. Pada P1 tersebut, diambil nilai kromosom pada gen ke-2.

Tabel 4.5 Child Mutation

Individu	Kromosom						
C3	52	54	66	70	77	79	80

Pada Tabel 4.5, nilai gen ke-2 diubah nilainya secara acak dengan aturan lebih dari nilai gen sebelumnya dan kurang dari nilai gen setelahnya.

Setelah melakukan proses *crossover* dan *mutation* diatas, didapatkan populasi baru yang ditampilkan pada Tabel 4.6.

Tabel 4.6 Populasi Baru

Individu	Kromosom						
P1	52	59	66	70	77	79	80
P2	52	53	57	58	59	74	76
P3	51	52	64	67	75	77	79
C1	52	53	57	58	75	77	79
C2	51	52	59	64	67	74	76
C3	52	54	66	70	77	79	80

Tabel 4.6 menampilkan populasi baru pada produk sabuk hasil dari proses *crossover* dan *mutation*.

4.13.4 Inisialisasi Sub-Himpunan

Setelah didapatkan populasi baru dari produk sabuk, langkah selanjutnya yaitu inisialisasi sub-himpunan dimana nilai gen dari tiap kromosom dijadikan sebagai acuan dalam menentukan batas dari tiap sub-himpunan, dan U_{min} sebagai batas bawah sedangkan U_{max} sebagai batas atas dalam sub-himpunan. Pada proses *fuzzy time series* ini dipilih P1 sebagai contoh perhitungan, pada Tabel 4.7 ditampilkan sub-himpunan dari P1.

$$U_{min} = 50, U_{max} = 85$$

Tabel 4.7 Inisialisasi Sub-Himpunan

Sub-Himpunan	Batas Bawah	Batas Atas
U1	50	<52
U2	52	<59
U3	59	<60
U4	66	<70
U5	70	<77
U6	77	<79
U7	79	<80
U8	80	85

4.13.5 Fuzzyfikasi

Setelah didapatkan sub-himpunan, tahap selanjutnya yaitu proses *fuzzyfikasi* dimana setiap sub-himpunan diubah menjadi variabel linguistik. Untuk mendapatkan variabel linguistiknya menggunakan Persamaan 2.3, yang menghasilkan fungsi keanggotaan sebagai berikut:

$$A1 = 1/U_1 + 0.5/U_2 + 0/U_3 + 0/U_4 + 0/U_5 + 0/U_6 + 0/U_7 + 0/U_8 + 0/U_9 + 0/U_{10}$$

$$A2 = 0.5/U_1 + 1/U_2 + 0.5/U_3 + 0/U_4 + 0/U_5 + 0/U_6 + 0/U_7 + 0/U_8 + 0/U_9 + 0/U_{10}$$

⋮

$$A9 = 0/U_1 + 0/U_2 + 0/U_3 + 0/U_4 + 0/U_5 + 0/U_6 + 0/U_7 + 0.5/U_8 + 1/U_9 + 0.5/U_{10},$$

$$A10 = 0/U_1 + 0/U_2 + 0/U_3 + 0/U_4 + 0/U_5 + 0/U_6 + 0/U_7 + 0/U_8 + 0.5/U_9 + 1/U_{10}.$$

Pada fungsi keanggotaan diatas, terdapat variabel linguistik A1, A2, ..., A10 dimana jumlah tersebut sama dengan jumlah sub-himpunan yang telah diinisialisasi sebelumnya. Kemudian setelah mendapatkan variabel linguistiknya yaitu menentukan data produksi masuk dalam interval sub-himpunan U. sebagai contoh untuk data produksi bulan Januari 2016 yaitu 57, dimana nilai tersebut masuk kedalam interval sub-himpunan U2, kemudian U2 pada fungsi keanggotaan memiliki nilai 1 pada variabel linguistik A2. Jadi untuk *fuzzyfikasi* bulan Januari 2016 yaitu A2.

Tabel 4.8 Fuzzyfikasi

Tahun	Bulan	Produksi	Fuzzyfikasi
2016	Januari	57	A2
	Ferbuari	52	A2
	Maret	59	A3
	April	53	A2
	Mei	68	A4
	Juni	63	A3
	Juli	64	A3
	Agustus	53	A2
	September	53	A2
	Oktober	83	A8
	November	70	A5
	Desember	77	A6
2017	Januari	55	A2
	Ferbuari	57	A2
	Maret	72	A5
	April	69	A4
	Mei	82	A8
	Juni	62	A3
	Juli	81	A8
	Agustus	80	A8
	September	74	A5
	Oktober	75	A5
	November	54	A2
	Desember	65	A3

Pada Tabel 4.8, ditampilkan hasil *fuzzyfikasi* pada P1 dengan data yang digunakan seperti pada Tabel 3.1 yaitu data produk sabuk kulit sapi.

4.13.6 FLR & FLRG

Setelah didapatkan hasil *fuzzyfikasi*, maka tahapan yang dilakukan setelahnya yaitu menentukan FLR dan FLRG dari hasil *fuzzyfikasi* tersebut.



Tabel 4.9 FLR

Tahun	Bulan	Current	FLR
2016	Januari	A2	A2->A2
	Februari	A2	A2->A3
	Maret	A3	A3->A2
	April	A2	A2->A4
	Mei	A4	A4->A3
	Juni	A3	A3->A3
	Juli	A3	A3->A2
	Agustus	A2	A2->A2
	September	A2	A2->A8
	Oktober	A8	A8->A5
	November	A5	A5->A6
	Desember	A6	A6->A2
2017	Januari	A2	A2->A2
	Februari	A2	A2->A5
	Maret	A5	A5->A4
	April	A4	A4->A8
	Mei	A8	A8->A3
	Juni	A3	A3->A8
	Juli	A8	A8->A8
	Agustus	A8	A8->A5
	September	A5	A5->A5
	Oktober	A5	A5->A2
	November	A2	A2->A3
	Desember	A3	-

Pada Tabel 4.9, untuk mendapatkan FLR dari hasil *fuzzyfikasi*, menggunakan persamaan:

$$A_i \rightarrow A_{i+1} \tag{4.1}$$

A_i merupakan hasil *fuzzyfikasi* yang direpresentasikan dengan *current state* dan A_{i+1} adalah *next state*. Sebagai contoh untuk *fuzzyfikasi* pada bulan Oktober 2016 adalah A8 dan *fuzzyfikasi* pada bulan November 2016 adalah A5, jadi hasil FLRnya adalah A8->A5, dimana A8 sebagai *current state* dan A5 sebagai *next state*.

Tabel 4.10 FLRG

Current	Next
A1	-
A2	A2,A3,A4,A8,A5
A3	A2,A3,A8
A4	A3,A8
A5	A6,A4,A5,A2
A6	A2
A7	-
A8	A5,A3,A8



Dari Tabel 4.10, FLRG merupakan kumpulan *next state* dari *current state* yang sama. Sebagai contoh pada nilai *current state* A8 memiliki FLR A8->A5, A8->A3 dan A8->A8 maka FLRGnya yaitu A8 -> A5, A3, A8.

4.13.7 Defuzzyfikasi

Setelah didapatkan FLRG, langkah selanjutnya yaitu *defuzzyfikasi* FLRG yang telah didapatkan sebelumnya untuk kemudian menjadi acuan hasil prediksi penelitian ini. Langkah pertama seperti yang telah dijelaskan pada sub-bab 2.2.5 yaitu, mendapatkan *median* yang direpresentasikan sebagai m_i dari masing-masing *next state* dalam FLRG. Kemudian jika terdapat lebih dari 1 *next state*, maka untuk mendapatkan *defuzzyfikasi A* menggunakan persamaan:

$$A_i = (m_1, m_2, \dots, m_k) / k \tag{4.2}$$

Tabel 4.11 Defuzzyfikasi

Current	Next	Defuzzyfikasi
A1	A1	51
A2	A2,A3,A4,A8,A5	68.4
A3	A2,A3,A8	66.8333
A4	A3,A8	72.5
A5	A6,A4,A5,A2	68.75
A6	A2	55.5
A7	A7	79.5
A8	A5,A3,A8	72.8333

Pada Tabel 4.11 ditampilkan hasil dari *defuzzyfikasi*, sebagai contoh untuk perhitungan *defuzzyfikasi* A2 menggunakan Persamaan 4.2, yaitu sebagai berikut:

$$\begin{aligned} A_2 &= m_2 + m_3 + m_4 + m_8 + m_5 / 5 \\ &= 55.5 + 62.5 + 68 + 82.5 + 73.5 / 5 \\ &= 68.4 \end{aligned}$$

4.13.8 Prediksi

Setelah melewati tahap *defuzzyfikasi*, lalu tahap selanjutnya yaitu prediksi. Untuk mendapatkan nilai prediksi digunakan persamaan $t - 1$, dimana t merupakan waktu data aktual. Sebagai contoh untuk prediksi pada bulan Juni 2017 digunakan hasil *defuzzyfikasi* dari bulan Mei 2017, dimana pada bulan tersebut termasuk dalam *fuzzyfikasi* A8 yang hasil *defuzzyfikasi* nya adalah 72.8333. Jadi hasil prediksi untuk bulan Juni 2017 yaitu 73 unit sabuk, dimana dari hasil *defuzzyfikasi* tersebut dibulatkan ke atas. Pada Tabel 4.12 ditampilkan hasil dari prediksi.

Tabel 4.12 Hasil Prediksi

Tahun	Bulan	Data Aktual (a)	Prediksi (b)	Error (a - b)
2016	Januari	57	-	-
	Ferbuari	52	69	-17
	Maret	59	69	-10

Tahun	Bulan	Data Aktual (a)	Prediksi (b)	Error (a – b)
	April	53	67	-14
	Mei	68	69	-1
	Juni	63	73	-10
	Juli	64	67	-3
	Agustus	53	67	-14
	September	53	2	51
	Oktober	83	69	14
	November	70	73	-3
2017	Desember	77	69	8
	Januari	55	56	-1
	Ferbuari	57	69	-12
	Maret	72	69	3
	April	69	69	0
	Mei	82	73	9
	Juni	62	73	-11
	Juli	81	67	14
	Agustus	80	73	7
	September	74	73	1
	Oktober	75	69	6
	November	54	69	-15
2018	Desember	65	69	-4
Januari	-	67	-	

Dari Tabel 4.12, ditampilkan juga hasil *error* yang didapatkan dari hasil prediksi. Untuk mendapatkan hasil *error* menggunakan persamaan:

$$\text{error} = \text{data aktual} - \text{hasil prediksi} \quad (4.3)$$

Sebagai contoh hasil *error* yang didapatkan pada Desember 2017 yaitu -4, dimana data aktual 65 dan hasil prediksi 69.

4.13.9 RMSE & Fitness

Setelah didapatkan hasil prediksi, kemudian menghitung RMSE yang dihasilkan dari prediksi tersebut. Sebagai contoh untuk mendapatkan nilai RMSE, yaitu nilai *error* dikuadratkan yang hasil kuadratnya ditampilkan pada Tabel 4.13.

Tabel 4.13 Kuadrat Error

Tahun	Bulan	Error	Kuadrat
2016	Januari	-	-
	Ferbuari	-17	289
	Maret	-10	100
	April	-14	196
	Mei	-1	1
	Juni	-10	100
	Juli	-3	9
	Agustus	-14	196
	September	51	2601
	Oktober	14	196
	November	-3	9
	Desember	8	64
2017	Januari	-1	1

Tahun	Bulan	Error	Kuadrat
	Ferbuari	-12	144
	Maret	3	9
	April	0	0
	Mei	9	81
	Juni	-11	121
	Juli	14	196
	Agustus	7	49
	September	1	1
	Oktober	6	36
	November	-15	225
	Desember	-4	16
Jumlah			4640

Kemudian dari hasil kuadrat pada Tabel 4.13, dihitung nilai RMSE dengan menggunakan Persamaan 2.7 sebagai berikut:

$$RMSE = \sqrt{\frac{4640}{23}}$$

$$= 14.20349008$$

Dari hasil tersebut, didapatkan nilai RMSE untuk P1 yaitu 14.20349008.

Tabel 4.14 Hasil RMSE

Individu	RMSE
P1	14.2034901
P2	10.1660133
P3	8.31708561
C1	9.61882213
C2	9.62785813
C3	9.9060807

Pada Tabel 4.14, ditampilkan hasil RMSE yang diperoleh dalam populasi. Dari hasil RMSE diatas, kemudian menghitung nilai *fitness* dengan menggunakan persamaan:

$$fitness = \frac{1}{RMSE} \tag{4.4}$$

Tabel 4.15 Hasil Fitness

Individu	Fitness
P1	0.07041
P2	0.09837
P3	0.12023
C1	0.10396
C2	0.10387
C3	0.10095

Pada Tabel 4.15, ditampilkan hasil *fitness* yang diperoleh dalam populasi yang didapatkan menggunakan Persamaan 4.4.



4.13.10 Seleksi

Tahap terakhir dalam penelitian ini yaitu seleksi, dimana dalam penelitian ini menggunakan teknik seleksi *elitism*. Teknik ini memilih individu dengan *fitness* terbaik dalam populasi untuk kemudian menjadi *parent* pada generasi berikutnya. Pada Tabel 4.16, ditampilkan individu-individu yang telah mengalami seleksi sesuai dengan jumlah popsize.

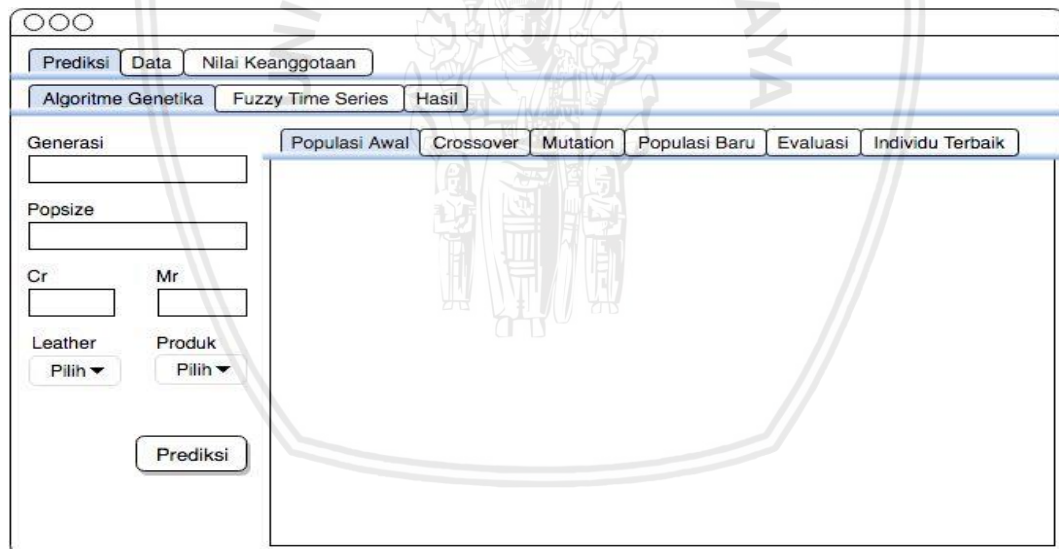
Tabel 4.16 Hasil Seleksi

Individu	RMSE	Fitness
P3	8.31708561	0.12023
C1	9.61882213	0.10396
C2	9.62785813	0.10387

Dari Tabel 4.16, terpilih individu P3, C1 dan C2 dengan nilai *fitness* tertinggi dari populasi.

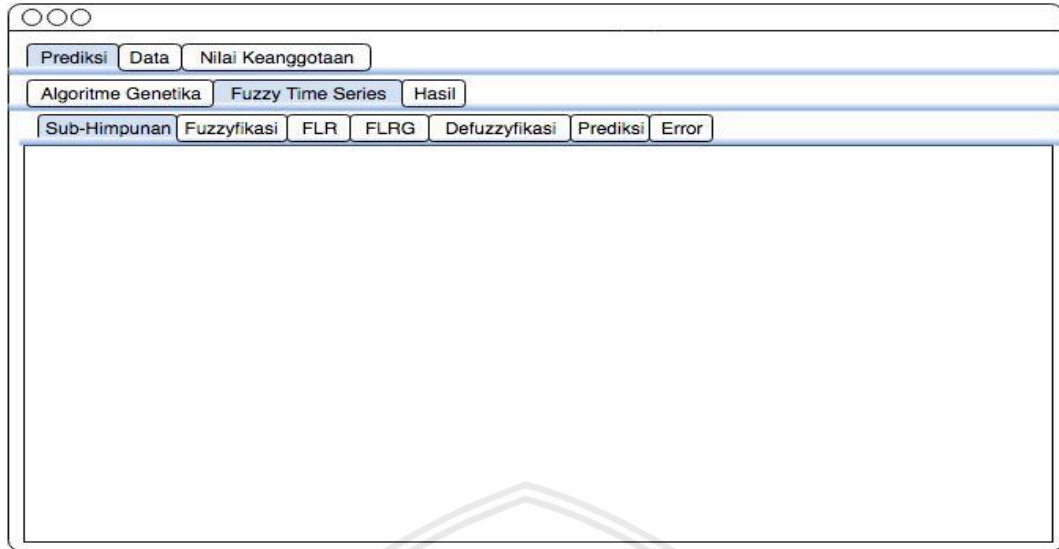
4.14 Perancangan Antarmuka

Pada bab ini ditampilkan perancangan antarmuka dalam implementasi algoritme pada penelitian ini.



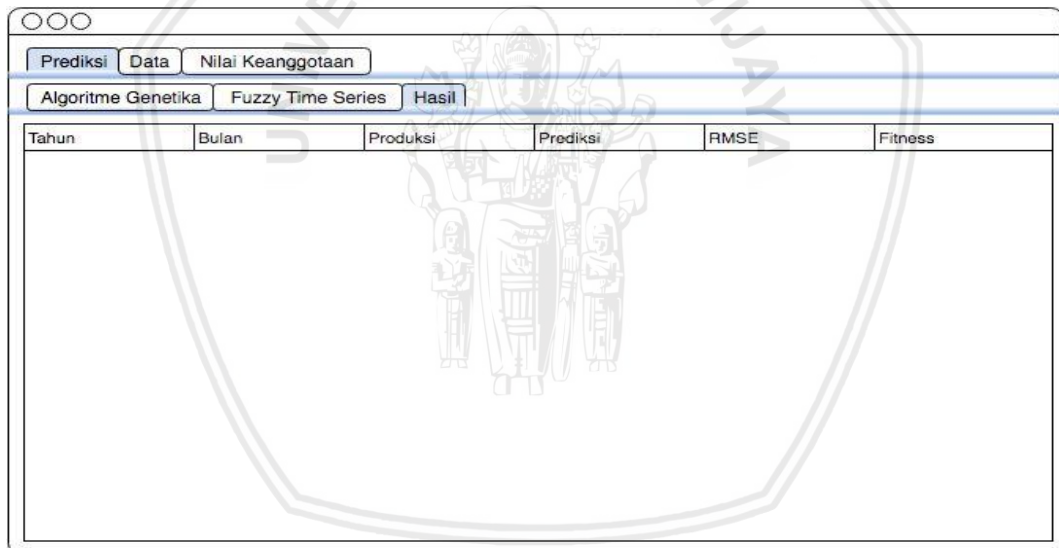
Gambar 4.17 Halaman Awal Proses Algoritme Genetika

Pada Gambar 4.17, ditampilkan halaman awal pada proses algoritme genetika, dimana terdapat 4 *textfield* yang masing-masing untuk menginputkan jumlah generasi, jumlah popsize (populasi awal), *cr* (*crossover rate*) dan *mr* (*mutation rate*). Dan juga terdapat 2 *combobox* yang masing-masing untuk memilih kulit dan produk yang akan diprediksi.



Gambar 4.19 Halaman Proses *Fuzzy Time Series*

Pada Gambar 4.18, ditampilkan halaman dari proses *fuzzy time series* yang meliputi sub-himpunan, *fuzzyfikasi*, FLR (*fuzzy logical relationship*), FLRG (*fuzzy logical relationship group*), *defuzzyfikasi*, prediksi dan error.



Gambar 4.18 Halaman Hasil Prediksi

Pada Gambar 4.19, ditampilkan halaman hasil prediksi dari individu terbaik yang dihasilkan dari proses algoritme genetika.

BAB 5 IMPLEMENTASI

Bab ini berisikan tentang implementasi algoritme dalam menyelesaikan permasalahan pada penelitian ini, yang berisikan spesifikasi sistem dan implementasi algoritme.

5.1 Ruang Lingkup Pengujian

Dalam penelitian ini, dibutuhkan spesifikasi ruang lingkup pengujian yang terdiri dari perangkat keras dan perangkat lunak.

5.1.1 Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pengujian, ditampilkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
Model	Macbook Pro 13-inch
Processor	Processor 2,7 GHz Intel Core i7
Memori	RAM 4GB 1067 MHz DDR3

5.1.2 Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengujian, ditampilkan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

Komponen	Spesifikasi
Sistem Operasi	High Sierra
Pemrograman	Java
Aplikasi	Netbeans 8.2

5.2 Implementasi Algoritme

Pada bab ini, dijelaskan tentang implementasi algoritme dalam membantu menyelesaikan masalah pada penelitian ini, dimana yang dijelaskan yaitu algoritme *universe of discourse*, inialisasi populasi, *crossover*, *mutation*, inialisasi sub-himpunan, *fuzzyfikasi*, FLR (*Fuzzy Logical Relationship*), FLRG (*Fuzzy Logical Relationship Group*), *defuzzyfikasi*, RMSE (*Root Mean Square Error*), *fitness* dan seleksi.

5.2.1 Universe of Discourse

Nilai *universe of discourse* digunakan sebagai acuan dalam membangkitkan nilai populasi awal, pada Kode Program 5.1 ditampilkan implementasi dari *universe of discourse*.

Algoritme 1: Universe of Discourse	
1	for (int i = 2; i < dataa.length; i++) {
2	if (dmin < dataa[i][item]) {
3	dmin = dmin;
4	} else {
5	dmin = dataa[i][item];
6	}
7	}
8	for (int i = 2; i < dataa.length; i++) {
9	if (dmax > dataa[i][item]) {
10	dmax = dmax;
11	} else {
12	dmax = dataa[i][item];
13	}
14	}
15	umin = dmin - d1;
16	umax = dmax + d2;

Kode Program 5.1 Implementasi *Universe Of Discourse*

Penjelasan pada Kode Program 5.1 sebagai berikut:

1. Baris 1 – 7 mencari nilai terendah dalam data.
2. Baris 8 – 14 mencari nilai tertinggi dalam data.
3. Baris 15 – 16 menghitung *universe of discourse*.

5.2.2 Inisialisasi Populasi

Inisialisasi populasi dibangkitkan dalam interval *universe of discourse* yang telah didapatkan sebelumnya, dimana panjang tiap kromosom telah ditentukan dengan panjangnya yaitu 7. Pada Kode Program 5.2 ditampilkan implementasi dari inisialisasi populasi.

Algoritme 2: Inisialisasi Populasi	
1	for (int i = 0; i < popsize; i++) {
2	for (int j = 0; j < panjangkromosom; j++) {
3	if (j == 0) {
4	populasi[i][j] = (int) (Math.random() * ((umax - 1) - (umin + 1) + 1) + (umin + 1);
5	} else {
6	temp1 = (int) (Math.random() * ((umax - 1) - (umin + 1) + 1) + (umin + 1);
7	for (int k = 0; k < j; k++) {
8	if (temp1 == populasi[i][k]) {
9	j = j - 1;
10	break;
11	} else {
12	populasi[i][j] = temp1;
13	}
14	}
15	}
16	}
17	}
18	for (int i = 0; i < populasi.length; i++) {
19	for (int j = 0; j < populasi[0].length; j++) {
20	for (int k = 0; k < populasi[0].length; k++) {
21	if (populasi[i][k] > populasi[i][j]) {
22	int temp = populasi[i][k];
23	populasi[i][k] = populasi[i][j];

Algoritme 2: Inisialisasi Populasi	
24	populasi[i][j] = temp;
25	}
26	}
27	}
28	}

Kode Program 5.2 Implementasi Inisialisasi Populasi

Penjelasan pada Kode Program 5.2 sebagai berikut:

1. Baris 1 – 6 membangkitkan nilai kromosom pada populasi awal secara acak dengan *umin* sebagai batas bawah dan *umax* sebagai batas atas.
2. Baris 7 – 17 mencegah munculnya nilai gen yang sama dalam satu kromosom.
3. Baris 18 – 28 melakukan pengurutan nilai kromosom dari terkecil ke terbesar.

5.2.3 Crossover

Proses *crossover* dilakukan setelah didapatkan populasi awal, dimana nilai kromosom ditukar antar individu dengan batasan yang disebut dengan *cut-point*. Pada Kode Program 5.3 ditampilkan implementasi dari *crossover*.

Algoritme 3: Crossover	
1	for (int i = 0; i < offspring; i = i + 2) {
2	boolean sama = true;
3	induk[0] = r.nextInt(popsize - 1);
4	while (sama) {
5	induk[1] = r.nextInt(popsize - 1);
6	if (induk[0] != induk[1]) {
7	sama = false;
8	}
9	}
10	cutpoint = r.nextInt(panjangkromosom - 1) + 1;
11	for (int k = 0; k < panjangkromosom; k++) {
12	if (k > cutpoint) {
13	temp[0][k] = indukc[1][k];
14	temp[1][k] = indukc[0][k];
15	} else {
16	temp[0][k] = indukc[0][k];
17	temp[1][k] = indukc[1][k];
18	}
19	}
20	for (int j = 0; j < hasilcrossover.length; j++) {
21	for (int k = 0; k < hasilcrossover[0].length; k++) {
22	for (int l = 0; l < hasilcrossover[0].length; l++) {
23	if (hasilcrossover[j][l] > hasilcrossover[j][k])
24	{
25	int x = hasilcrossover[j][l];
26	hasilcrossover[j][l] = hasilcrossover[j][k];
27	hasilcrossover[j][k] = x;
28	}
29	}
30	}
31	for (int i = 0; i < hasilcrossover.length; i++) {
32	for (int j = 0; j < hasilcrossover[i].length; j++) {
33	for (int l = j; l < hasilcrossover[i].length; l++){
34	if (j != l) {
35	if (hasilcrossover[i][j] ==

Algoritme 3: Crossover

```

36     hasilcrossover[i][l]) {
37         if (j == 0) {
38             hasilcrossover[i][j] = (int)
39                 (Math.random() * ((hasilcrossover[i][j +
40                 1] - 1) - (umin + 1) + 1)) + (umin + 1);
41         } else if (j == (hasilcrossover[i].length -
42         1)) {
43             hasilcrossover[i][j] = (int)
44                 (Math.random() * ((umax - 1) -
45                 (hasilcrossover[i][j - 1] + 1) + 1)) +
46                 (hasilcrossover[i][j - 1] + 1);
47         } else {
48             hasilcrossover[i][j] = (int)
49                 (Math.random() * ((hasilcrossover[i][j +
50                 1] - 1) - (hasilcrossover[i][j - 1] + 1)
51                 + 1)) + (hasilcrossover[i][j - 1] + 1);
52         }
53     }
54 }
55 }
56 }
57 }
58 }

```

Kode Program 5.3 Implementasi *Crossover*

Penjelasan pada Kode Program 5.3 sebagai berikut:

1. Baris 1 – 9 menentukan induk secara acak dari populasi awal yang akan mengalami proses *crossover*.
2. Baris 10 menentukan *cutpoint* secara acak dari panjang kromosom induk yang terpilih.
3. Baris 11 – 19 melakukan proses *crossover*.
4. Baris 20 – 30 melakukan pengurutan nilai kromosom hasil *crossover* dari terkecil ke terbesar.
5. Baris 31 – 47 melakukan perulangan jika pada hasil *crossover* terdapat nilai gen yang sama, nilai gen tersebut mengalami *mutation* dengan kondisi hasil *mutation* tersebut lebih dari nilai gen sebelumnya dan kurang dari nilai gen setelahnya.

5.2.4 Mutation

Pada proses *mutation*, diambil nilai sebuah gen pada kromosom yang kemudian nilai gen tersebut diubah secara acak dengan kondisi hasil *mutation* tersebut lebih dari nilai gen sebelumnya dan kurang dari nilai gen setelahnya. Pada Kode Program 5.4 ditampilkan implementasi dari *mutation*.

Algoritme 4: Mutation

```

1   for (int i = 0; i < offspring; i++) {
2       int p = r.nextInt(popsize - 1);
3       int gen = r.nextInt(panjangkromosom - 1) + 1;
4       for (int j = 0; j < panjangkromosom; j++) {
5           if (j == gen) {
6               if (j == 0) {
7                   hasilmutasi[i][j] = (int) (Math.random() *
8                   ((populasi[p][j + 1] - 1) - (umin + 1) + 1)) +

```

Algoritme 4: Mutation

```

      (umin + 1);
8      } else if (j == 8) {
9          hasilmutasi[i][j] = (int) (Math.random() *
              ((umax - 1) - (populasi[p][j - 1] + 1) + 1)) +
              (populasi[p][j - 1] + 1);
10         } else {
11             hasilmutasi[i][j] = (int) (Math.random() *
                  ((populasi[p][j + 1] - 1) - (populasi[p][j - 1]
                      + 1) + 1)) + (populasi[p][j - 1] + 1);
12         }
13     } else {
14         hasilmutasi[i][j] = populasi[p][j];
15     }
16 }
17 }

```

Kode Program 5.4 Implementasi Mutation

Penjelasan pada Kode Program 5.4 sebagai berikut:

1. Baris 2 menentukan induk secara acak dari populasi awal yang akan mengalami proses *mutation*.
2. Baris 3 menentukan gen secara acak pada kromosom induk yang terpilih.
3. Baris 4 – 17 melakukan proses *mutation* pada gen dengan kondisi hasil *mutation* tersebut lebih dari nilai gen sebelumnya dan kurang dari nilai gen setelahnya.

5.2.5 Sub-Himpunan

Pada proses sub-himpunan, nilai kromosom pada populasi baru hasil dari reproduksi digunakan dalam membentuk sub-himpunan dan pada setiap individu membentuk sub-himpunan yang terbagi menjadi u_1, u_2, \dots, u_8 , dimana sub-himpunan tersebut didapatkan dari jumlah kromosom ditambah satu. Pada Kode Program 5.5 ditampilkan implementasi dari sub-himpunan.

Algoritme 5: Sub-Himpunan

```

1  for (int i = 0; i < populasinew.length; i++) {
2      for (int j = 0; j < panjangkromosom + 2; j++) {
3          if (j == 0) {
4              subhimpunan[i][j] = umin;
5          }
6          if (j == 8) {
7              subhimpunan[i][j] = umax;
8          }
9          if (j != 0 && j != 8) {
10             subhimpunan[i][j] = populasinew[i][j - 1];
11         }
12     }
13 }

```

Kode Program 5.5 Implementasi Sub-Himpunan

Penjelasan pada Kode Program 5.5 sebagai berikut:

1. Baris 3 – 5 menginisialisasi nilai sub-himpunan awal yang diambil dari nilai *umin*.
2. Baris 6 – 8 menginisialisasi nilai sub-himpunan akhir yang diambil dari nilai *umax*.

- Baris 9 – 11 menginisialisasi nilai sub-himpunan yang diambil dari nilai kromosom pada populasi.

5.2.6 Fuzzyfikasi

Pada proses *fuzzyfikasi*, data jumlah produksi dimasukkan kedalam sub-himpunan yang termasuk dalam interval yang telah didefinisikan sebelumnya. Pada Kode Program 5.6 ditampilkan implementasi dari *fuzzyfikasi*.

Algoritme 6: Fuzzyfikasi	
1	for (int i = 0; i < dataa.length; i++) { //10
2	for (int j = 0; j < subhimpunan.length; j++) {
3	for (int k = 0; k < subhimpunan[0].length; k++) {
4	if (subhimpunan[j][k] <= dataa[i][item] && dataa[i][item] < subhimpunan[j][k + 1]) {
5	fuzzyfikasi[i][j] = k + 1;
6	for (int l = 0; l < fungsi.length; l++) {
7	if (fungsi[l][k] == 1) {
8	fuzzyfikasi[i][j] = l + 1;
9	}
10	}
11	}
12	}
13	}
14	}

Kode Program 5.6 Implementasi *Fuzzyfikasi*

Penjelasan pada Kode Program 5.6 sebagai berikut:

- Baris 1 – 3 melakukan perulangan pada jumlah data dan sub-himpunan.
- Baris 4 – 5 menentukan apakah data jumlah produksi termasuk ke dalam interval sub-himpunan u_1 sampai u_8 yang telah didapatkan sebelumnya.
- Baris 6 – 9 menentukan variabel linguistik dari sub-himpunan yang terpilih pada tabel fungsi yang bernilai 1.

5.2.7 FLR

Pada proses FLR (*Fuzzy Logical Relationship*), hasil *fuzzyfikasi* yang telah didapatkan kemudian ditentukan hubungan atau nilai *fuzzyfikasi* setelahnya yang didefinisikan sebagai *next state*. Pada Kode Program 5.7 ditampilkan implementasi dari FLR.

Algoritme 7: FLR	
1	for (int i = 0; i < fuzzyfikasi.length - 1; i++) {
2	for (int j = 0; j < fuzzyfikasi[0].length; j++) {
3	flr[i][j] = fuzzyfikasi[i + 1][j];
4	}
5	}

Kode Program 5.7 Implementasi FLR

Penjelasan pada Kode Program 5.7 sebagai berikut:

- Baris 1 – 2 melakukan perulangan pada data *fuzzyfikasi*.
- Baris 3 mendefinisikan FLR = *fuzzyfikasi* [i + 1] sebagai *next state* dan *fuzzyfikasi* [i] sebagai *current state*.

5.2.8 FLRG

Pada proses FLRG (*Fuzzy Logical Relationship Group*), *next state* dari FLR yang memiliki *current state* yang sama dikelompokkan menjadi satu. Pada Kode Program 5.8 ditampilkan implementasi dari FLRG.

```
Algoritme 8: FLRG
1  for (int k = 0; k < fuzzyfikasi.length; k++) {
2      if (current == fuzzyfikasi[k][r]) {
3          next = flr[k][r];
4          count++;
5          if (list.size() == 0) {
6              list.add(next);
7          } else {
8              for (int l = 0; l < list.size(); l++) {
9                  if (next != list.get(l)) {
10                     tempnext = next;
11                 } else {
12                     tempnext = 0;
13                     l = list.size();
14                 }
15             }
16             if (tempnext > 0) {
17                 list.add(next);
18             }
19         }
20     } else {
21         if (count == 0 && k == fuzzyfikasi.length - 2) {
22             next = current;
23             list.add(current);
24             tempnext = 0;
25         }
26     }
27 }
```

Kode Program 5.8 Implementasi FLRG

Penjelasan pada Kode Program 5.8 sebagai berikut:

1. Baris 2 – 3 mendapatkan *next state* dari proses FLR yang telah didefinisikan sebelumnya.
2. Baris 5 – 19 mengelompokkan *next state* pada *current state* yang sama.
3. Baris 21 – 25 mendefinisikan jika tidak terdapat *next state*.

5.2.9 Defuzzyfikasi

Pada proses *defuzzyfikasi*, variabel linguistik *current state* pada FLRG diubah menjadi bilangan dimana untuk mendapatkannya menggunakan *next state* yang terdapat pada *current state*. Pada Kode Program 5.9 ditampilkan implementasi dari *defuzzyfikasi*.

```
Algoritme 9: Defuzzyfikasi
1  if (current == next) {
2      if (current == (panjangkromosom + 1)) {
3          median = (subhimpunan[r][next - 2] +
4                  subhimpunan[r][next - 1]) / 2;
5          average = median;
6          defuzzyfikasi[j][r] = average;
7      }
8  }
```

Algoritme 9: Defuzzyfikasi	
7	if (next < (panjangkromosom + 1)) {
8	median = (subhimpunan[r][next - 1] +
	subhimpunan[r][next]) / 2;
9	average = median;
10	defuzzyfikasi[j][r] = average;
11	}
12	} else {
13	for (int l = 0; l < list.size(); l++) {
14	if (list.get(l) == (panjangkromosom + 1)) {
15	median += (subhimpunan[r][list.get(l) - 2] +
	subhimpunan[r][list.get(l) - 1]) / 2;
16	}
17	if (list.get(l) > 0 && list.get(l) <
	(panjangkromosom + 1)) {
18	median += (subhimpunan[r][list.get(l) - 1] +
	subhimpunan[r][list.get(l)]) / 2;
19	}
20	}
21	average = median / list.size();
22	defuzzyfikasi[j][r] = average;
23	}

Kode Program 5.9 Implementasi Defuzzyfikasi

Penjelasan pada Kode Program 5.9 sebagai berikut:

1. Baris 1 – 6 melakukan perhitungan jika terdapat hanya satu *next state* pada *current state*.
2. Baris 7 – 11 melakukan perhitungan jika tidak terdapat *next state* pada *current state*.
3. Baris 12 – 20 melakukan perhitungan jika terdapat *next state* lebih dari satu pada *current state*.

5.2.10 RMSE & Fitness

Pada perhitungan RMSE (*Root Mean Square Error*) dan *fitness*, data jumlah produksi dikurangi dengan hasil prediksi untuk mendapatkan nilai *error* yang dihasilkan dari penelitian ini. Pada Kode Program 5.10 ditampilkan implementasi dari RMSE dan *fitness*.

Algoritme 10: RMSE & Fitness	
1	for (int i = 0; i < error.length - 1; i++) {
2	for (int j = 0; j < error[0].length; j++) {
3	error[i + 1][j] = dataa[i + 1][item] -
	(int)Math.ceil(prediksi[i + 1][j]);
4	square[i + 1][j] = (int)Math.pow(error[i+1][j], 2);
5	}
6	}
7	for (int i = 0; i < square[0].length; i++) {
8	temp = 0;
9	for (int j = 0; j < square.length; j++) {
10	temp += square[j][i];
11	}
12	mean[i] = (temp / square.length - 1);
13	root[i] = Math.sqrt(mean[i]);
14	}
15	for (int i = 0; i < root.length; i++) {

Algoritme 10: RMSE & Fitness	
16	fitness[i] = 1 / root[i];
17	}

Kode Program 5.10 Implementasi RMSE & Fitness

Penjelasan pada Kode Program 5.10 sebagai berikut:

1. Baris 2 – 3 menghitung nilai *error* dengan mengurang data aktual jumlah produksi dengan data hasil prediksi.
2. Baris 4 – 5 menghitung nilai kuadrat dari nilai *error* yang telah dihasilkan.
3. Baris 12 menghitung nilai rata-rata dari nilai kuadrat yang telah dihasilkan.
4. Baris 13 mengakarkan nilai rata-rata yang telah dihasilkan untuk mendapatkan nilai RMSE.
5. Baris 15 – 17 menghitung nilai *fitness* dengan cara 1 dibagi dengan nilai RMSE yang telah dihasilkan.

5.2.11 Seleksi

Pada proses seleksi, individu dengan nilai *fitness* terbaik pada populasi diurutkan yang berjumlah sama dengan popsize, yang kemudian individu tersebut digunakan sebagai induk pada generasi berikutnya. Pada Kode Program 5.11 ditampilkan implementasi dari seleksi.

Algoritme 11: Seleksi	
1	for (int i = 0; i < fitness.length - 1; i++) {
2	for (int j = 0; j < fitness.length - i - 1; j++) {
3	if (fitness[j] < fitness[j + 1]) {
4	temp = fitness[j];
5	fitness[j] = fitness[j + 1];
6	fitness[j + 1] = temp;
7	for (int k = 0; k < panjangkromosom; k++) {
8	temp2 = populasinew[j][k];
9	populasinew[j][k] = populasinew[j + 1][k];
10	populasinew[j + 1][k] = temp2;
11	}
12	}
13	}
14	}
15	for (int i = 0; i < popsize; i++) {
16	for (int j = 0; j < panjangkromosom; j++) {
17	populasi[i][j] = populasinew[i][j];
18	}
19	}

Kode Program 5.11 Implementasi Seleksi

Penjelasan pada Kode Program 5.11 sebagai berikut:

1. Baris 1 – 14 mengurutkan nilai *fitness* terbesar ke terkecil.
2. Baris 15 – 19 mengambil nilai *fitness* tertinggi sebanyak popsize.



5.3 Hasil Implementasi

Pada bab ini, menampilkan hasil dari implementasi algoritme berupa gambar dari halaman proses algoritme genetika, *fuzzy time series* dan hasil prediksi.

Generasi ke-	Kromosom	Fitness
Generasi ke-1	36 37 39 41 44 47 49	0.27107642486025224
Generasi ke-2	36 37 39 41 44 45 47	0.2741594402766581
Generasi ke-3	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-4	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-5	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-6	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-7	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-8	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-9	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-10	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-11	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-12	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-13	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-14	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-15	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-16	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-17	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-18	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-19	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-20	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-21	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-22	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-23	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-24	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-25	36 37 39 41 44 47 48	0.29078999901889985
Generasi ke-26	36 37 39 41 44 47 48	0.29078999901889985

Gambar 5.1 Hasil Proses Algoritme Genetika

Pada Gambar 5.1, ditampilkan halaman hasil dari individu terbaik pada proses algoritme genetika.

Generasi k...	Individu	U1	U2	U3	U4	U5	U6	U7	U8
Generasi k...	P1	[30, 31]	[31, 34]	[34, 37]	[37, 39]	[39, 40]	[40, 44]	[44, 46]	[46, 50]
	P2	[30, 35]	[35, 39]	[39, 40]	[40, 41]	[41, 42]	[42, 44]	[44, 47]	[47, 50]
	P3	[30, 32]	[32, 38]	[38, 40]	[40, 42]	[42, 44]	[44, 46]	[46, 47]	[47, 50]
	P4	[30, 35]	[35, 36]	[36, 37]	[37, 39]	[39, 43]	[43, 47]	[47, 49]	[49, 50]
	P5	[30, 31]	[31, 37]	[37, 43]	[43, 44]	[44, 45]	[45, 47]	[47, 48]	[48, 50]
	P6	[30, 31]	[31, 33]	[33, 34]	[34, 36]	[36, 40]	[40, 44]	[44, 46]	[46, 50]
	P7	[30, 33]	[33, 34]	[34, 36]	[36, 41]	[41, 46]	[46, 48]	[48, 49]	[49, 50]
	P8	[30, 32]	[32, 33]	[33, 42]	[42, 45]	[45, 46]	[46, 48]	[48, 49]	[49, 50]
	P9	[30, 35]	[35, 39]	[39, 41]	[41, 43]	[43, 44]	[44, 45]	[45, 46]	[46, 50]
	P10	[30, 36]	[36, 37]	[37, 39]	[39, 41]	[41, 44]	[44, 47]	[47, 49]	[49, 50]
	P11	[30, 31]	[31, 32]	[32, 39]	[39, 40]	[40, 41]	[41, 44]	[44, 45]	[45, 50]
	P12	[30, 32]	[32, 36]	[36, 38]	[38, 40]	[40, 45]	[45, 47]	[47, 48]	[48, 50]
	P13	[30, 31]	[31, 32]	[32, 33]	[33, 35]	[35, 41]	[41, 44]	[44, 45]	[45, 50]
	P14	[30, 34]	[34, 38]	[38, 39]	[39, 42]	[42, 44]	[44, 46]	[46, 48]	[48, 50]
	P15	[30, 37]	[37, 39]	[39, 42]	[42, 44]	[44, 46]	[46, 47]	[47, 48]	[48, 50]
	P16	[30, 32]	[32, 39]	[39, 43]	[43, 44]	[44, 46]	[46, 48]	[48, 49]	[49, 50]
	P17	[30, 31]	[31, 33]	[33, 34]	[34, 39]	[39, 42]	[42, 47]	[47, 49]	[49, 50]
	P18	[30, 36]	[36, 37]	[37, 40]	[40, 41]	[41, 42]	[42, 44]	[44, 46]	[46, 50]
	P19	[30, 34]	[34, 35]	[35, 37]	[37, 40]	[40, 44]	[44, 47]	[47, 48]	[48, 50]
	P20	[30, 31]	[31, 33]	[33, 38]	[38, 40]	[40, 42]	[42, 47]	[47, 49]	[49, 50]
	P21	[30, 31]	[31, 32]	[32, 34]	[34, 35]	[35, 37]	[37, 42]	[42, 49]	[49, 50]
	P22	[30, 36]	[36, 39]	[39, 43]	[43, 46]	[46, 47]	[47, 48]	[48, 49]	[49, 50]
	P23	[30, 31]	[31, 33]	[33, 40]	[40, 41]	[41, 43]	[43, 44]	[44, 46]	[46, 50]
	P24	[30, 32]	[32, 36]	[36, 37]	[37, 38]	[38, 40]	[40, 41]	[41, 47]	[47, 50]
	P25	[30, 35]	[35, 38]	[38, 39]	[39, 40]	[40, 45]	[45, 46]	[46, 48]	[48, 50]
	P26	[30, 31]	[31, 33]	[33, 35]	[35, 39]	[39, 41]	[41, 44]	[44, 49]	[49, 50]
	P27	[30, 31]	[31, 32]	[32, 34]	[34, 35]	[35, 37]	[37, 40]	[40, 44]	[44, 50]

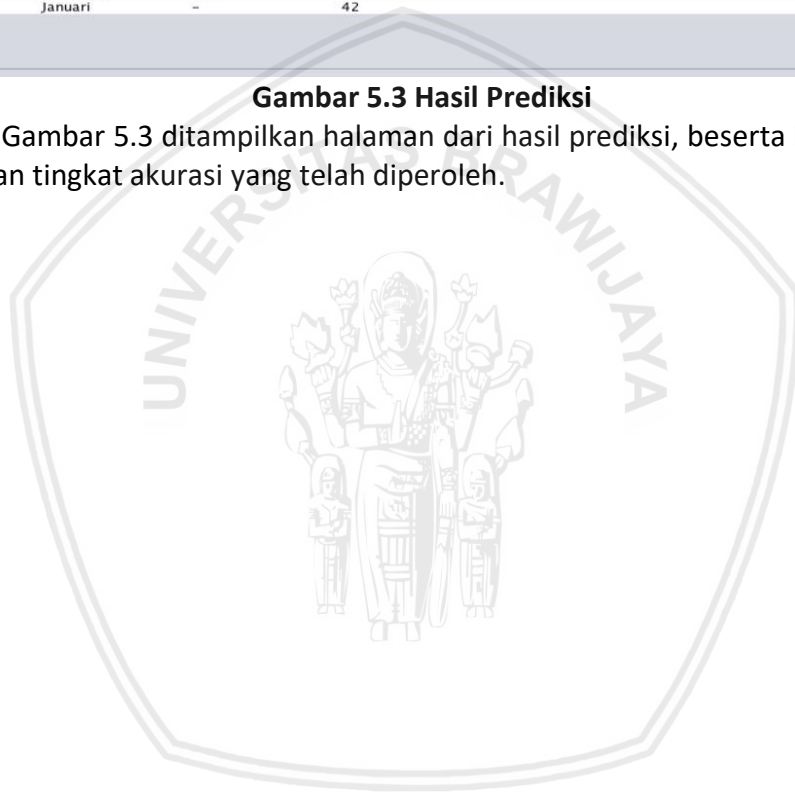
Gambar 5.2 Hasil Proses Fuzzy Time Series

Pada Gambar 5.2 ditampilkan halaman dari sub-himpunan pada proses *fuzzy time series*.

Prediksi		Data				
Algoritme Genetika		Fuzzy Time Series	Hasil			
Tahun	Bulan	Produksi	Prediksi	RMSE	Fitness	Akurasi
2016	Januari	47	-	3.71249359536...	0.26936073404...	90.16%
	Februari	45	40			
	Maret	43	45			
	April	34	33			
	Mei	47	43			
	Juni	40	40			
	Juli	47	42			
	Agustus	45	40			
	September	45	45			
	Oktober	46	45			
	November	47	45			
	Desember	32	40			
2017	Januari	44	43			
	Februari	48	45			
	Maret	34	40			
	April	38	43			
	Mei	41	40			
	Juni	33	33			
	Juli	39	43			
	Agustus	40	42			
	September	36	42			
	Oktober	38	40			
	November	36	40			
	Desember	42	40			
2018	Januari	-	42			

Gambar 5.3 Hasil Prediksi

Pada Gambar 5.3 ditampilkan halaman dari hasil prediksi, beserta hasil RMSE, *fitness* dan tingkat akurasi yang telah diperoleh.



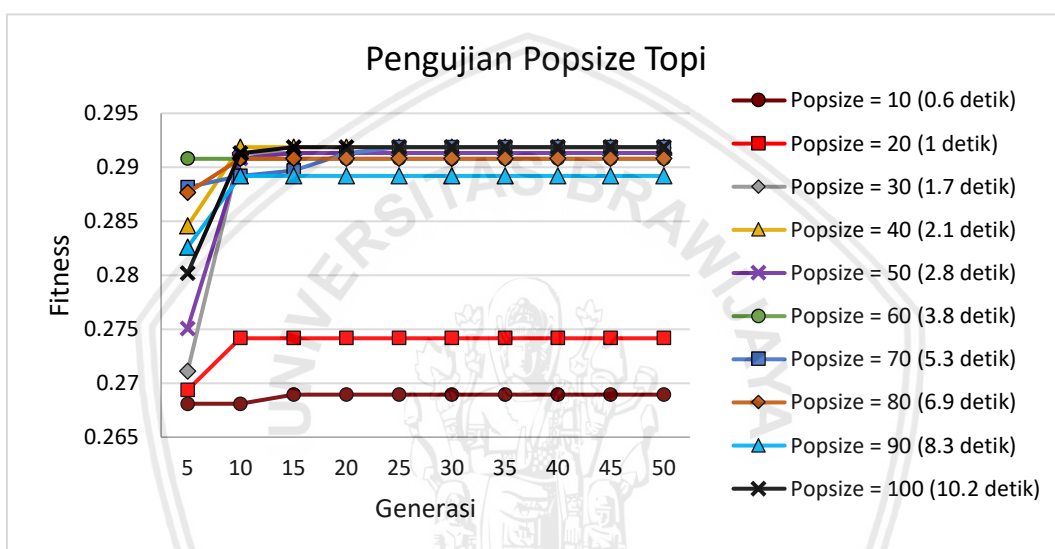
BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian Popsize

Pada pengujian popsize, parameter-parameter algoritme genetika yang digunakan yaitu jumlah generasi = 50 sedangkan untuk nilai kombinasi cr dan mr = 0.5 dan 0.5.

6.1.1 Pengujian Produk Topi

Pada Gambar 6.1, ditampilkan grafik hasil dari pengujian jumlah popsize pada produk topi.

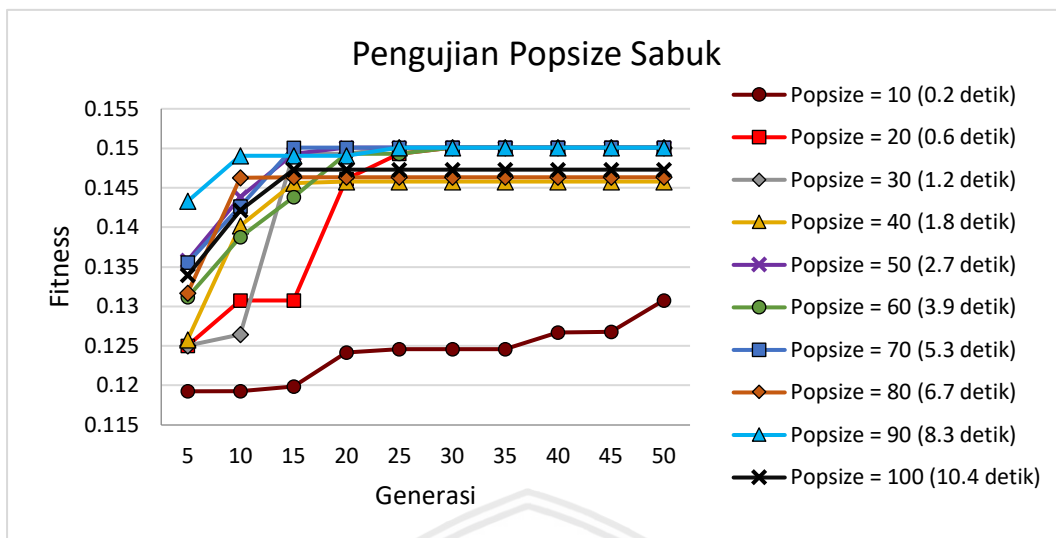


Gambar 6.1 Pengujian Popsize Topi

Pada grafik hasil Gambar 6.1, menunjukkan pengujian dengan popsize = 10 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.268095132 dan kemudian grafik mengalami konvergen pada generasi ke-15 dengan nilai *fitness* sebesar 0.268936877. Sedangkan pada jumlah popsize = 40, menghasilkan nilai *fitness* terbesar sejak dari generasi ke-10 yang bernilai 0.291865012 dan membutuhkan waktu komputasi yaitu 2.1 detik. Pada grafik hasil diatas, dapat dilihat bahwa terdapat 5 grafik yang mengalami konvergen sejak dari generasi ke-10 yaitu pada jumlah popsize = 20, 30, 40, 80 dan 90, sedangkan pada jumlah popsize = 60 menghasilkan nilai *fitness* sebesar 0.290789999 pada setiap generasinya. Dari pengujian pada produk topi ini, dapat disimpulkan bahwa pengujian dengan jumlah popsize kurang dari 30, menghasilkan nilai *fitness* yang rendah.

6.1.2 Pengujian Produk Sabuk

Pada Gambar 6.2, ditampilkan grafik hasil dari pengujian jumlah popsize pada produk sabuk.

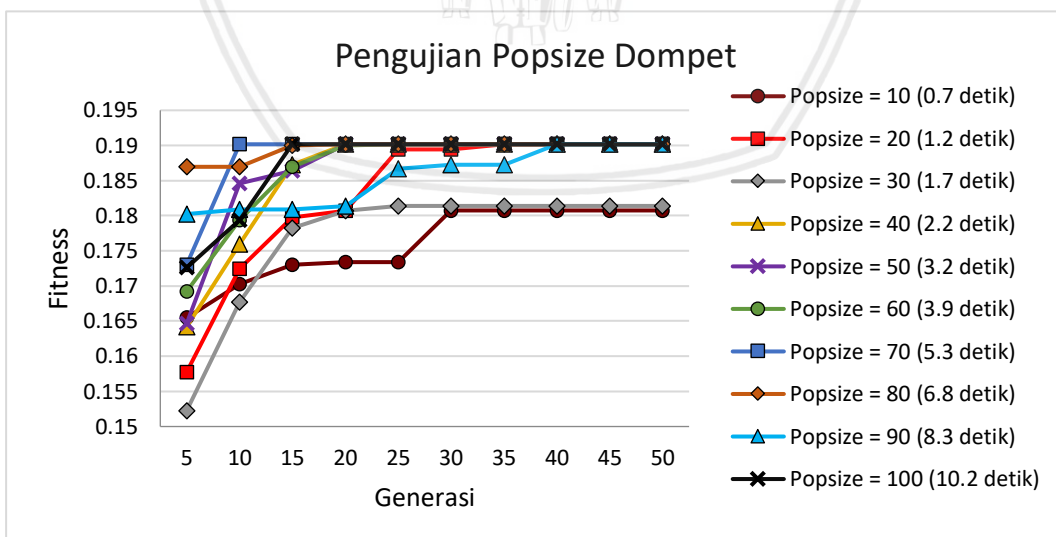


Gambar 6.2 Pengujian Popsiz Sabuk

Pada grafik hasil Gambar 6.2, menunjukkan pengujian dengan popsize = 10 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.119263873 dan nilai *fitness* terbesar dari jumlah popsize ini yaitu pada generasi ke-25 yang bernilai 0.13076839. Sedangkan pada jumlah popsize = 70, menghasilkan nilai *fitness* terbesar sejak dari generasi ke-15 yang bernilai 0.150089754 dan membutuhkan waktu komputasi yaitu 5.3 detik. Pada grafik hasil diatas, dapat dilihat bahwa terdapat 3 grafik yang mengalami konvergen sejak dari generasi ke-15 yaitu pada jumlah popsize = 70, 80 dan 100.

6.1.3 Pengujian Produk Dompot

Pada Gambar 6.3, ditampilkan grafik hasil dari pengujian jumlah popsize pada produk dompet.



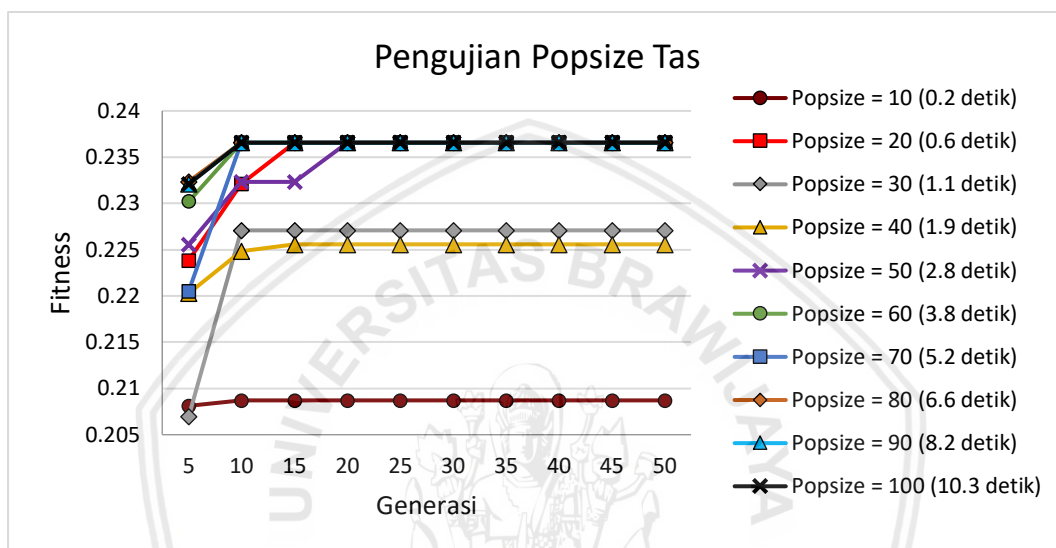
Gambar 6.3 Pengujian Popsiz Dompot

Pada grafik hasil Gambar 6.3, menunjukkan pengujian dengan popsize = 30 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.152267803 dan grafik mengalami konvergen pada generasi ke-25 dengan nilai *fitness*

sebesar 0.181395008. Sedangkan pada jumlah popsize = 70, menghasilkan nilai *fitness* terbesar sejak dari generasi ke-10 yang bernilai 0.19016709 dan membutuhkan waktu komputasi yaitu 5.3 detik. Pada grafik hasil diatas, dapat dilihat bahwa hanya pada jumlah popsize = 10 dan 30, yang tidak menghasilkan nilai *fitness* terbesar.

6.1.4 Pengujian Produk Tas

Pada Gambar 6.4, ditampilkan grafik hasil dari pengujian jumlah popsize pada produk tas.

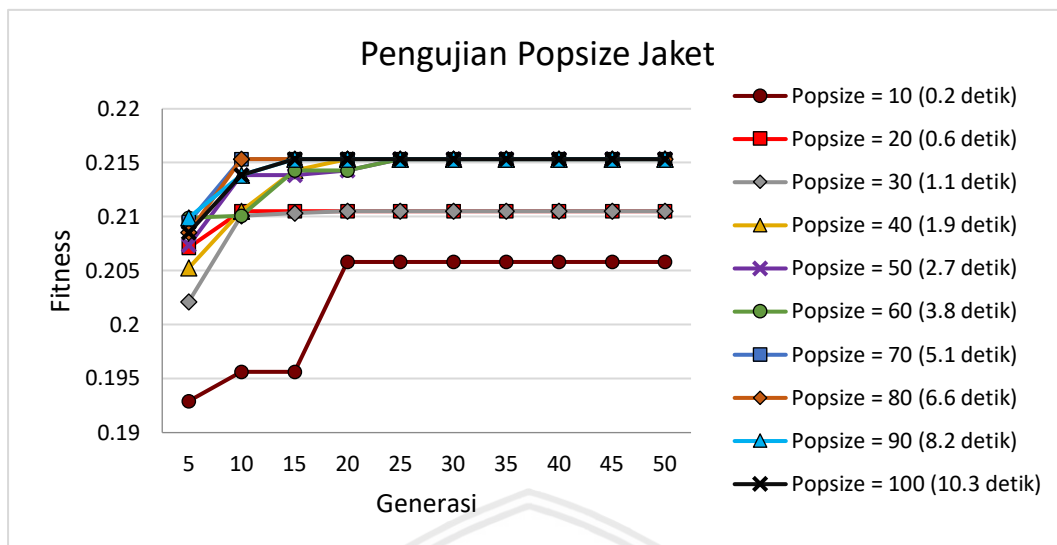


Gambar 6.4 Pengujian Popsiz Tas

Pada grafik hasil Gambar 6.4, menunjukkan pengujian dengan popsize = 30 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.206955406 dan grafik mengalami konvergen pada generasi ke-10 dengan nilai *fitness* sebesar 0.227089204. Sedangkan pada jumlah popsize = 60, menghasilkan nilai *fitness* terbesar sejak dari generasi ke-10 yang bernilai 0.236560924 dan membutuhkan waktu komputasi yaitu 3.8 detik. Pada grafik hasil diatas, dapat dilihat bahwa selain jumlah popsize = 60, terdapat juga 4 grafik yang menghasikan nilai *fitness* sejak dari generasi ke-10, yaitu pada jumlah popsize = 70, 80, 90 dan 100. Dari pengujian pada produk tas ini, dapat disimpulkan bahwa pengujian dengan jumlah popsize = 10, menghasilkan nilai *fitness* yang rendah sebesar 0.208711777 yang konvergen pada generasi ke-10.

6.1.5 Pengujian Produk Jaket

Pada Gambar 6.5, ditampilkan grafik hasil dari pengujian jumlah popsize pada produk jaket.

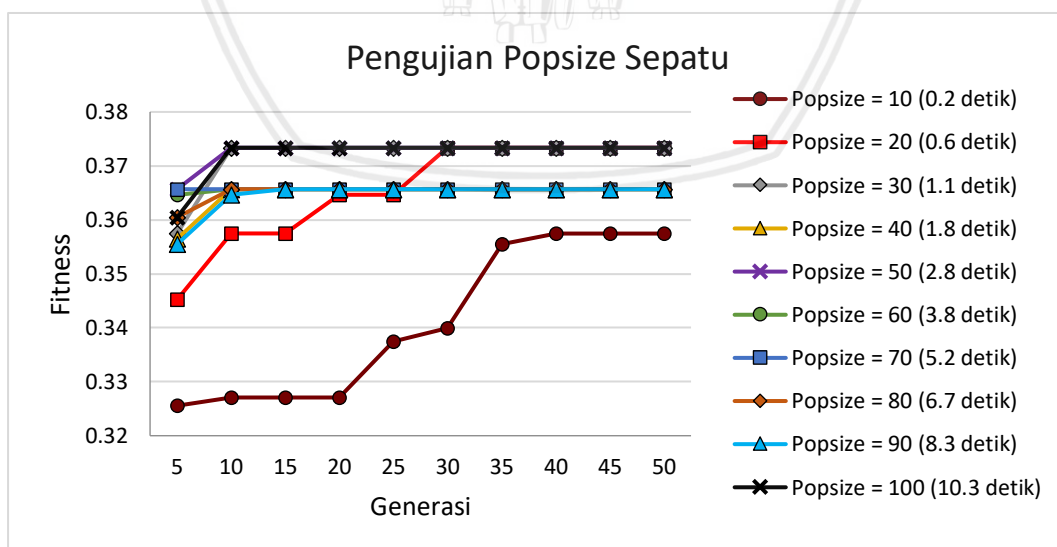


Gambar 6.5 Pengujian Popsize Jaket

Pada grafik hasil Gambar 6.5, menunjukkan pengujian dengan popsize = 10 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.192916636 dan grafik mengalami konvergen pada generasi ke-20 dengan nilai *fitness* sebesar 0.20580883. Sedangkan pada jumlah popsize = 70 dan 80, menghasilkan nilai *fitness* terbesar sejak dari generasi ke-10 yang bernilai 0.215339192 dan masing-masing membutuhkan waktu komputasi yaitu 5.1 dan 6.6 detik. Pada grafik hasil diatas, dapat dilihat bahwa pada jumlah popsize = 20 dan 30, grafik mengalami konvergen pada nilai *fitness* sebesar 0.210513639.

6.1.6 Pengujian Produk Sepatu

Pada Gambar 6.6, ditampilkan grafik hasil dari pengujian jumlah popsize pada produk jaket.



Gambar 6.6 Pengujian Popsize Sepatu

Pada grafik hasil Gambar 6.6, menunjukkan pengujian dengan popsize = 10 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.325562257 dan grafik mengalami konvergen pada generasi ke-40 dengan nilai *fitness*

sebesar 0.357460176. Sedangkan pada jumlah popsize = 30, 50 dan 100, menghasilkan nilai *fitness* terbesar sejak dari generasi ke-10 yang bernilai 0.373354978 dan masing-masing membutuhkan waktu komputasi yaitu 1.1, 2.8 dan 10.3 detik. Pada grafik hasil diatas, dapat dilihat bahwa pada jumlah popsize = 40, 60, 70, 80 dan 90, grafik mengalami konvergen pada nilai *fitness* sebesar 0.365678725.

Dari hasil pengujian jumlah popsize pada masing-masing produk, dapat disimpulkan bahwa jumlah popsize = 10 menghasilkan nilai *fitness* yang kecil. Kemudian secara bertahap nilai *fitness* mengalami kenaikan seiring dengan generasi dan bertambahnya jumlah popsize yang digunakan dalam pengujian. Untuk hasil lebih lengkap berupa tabel pada pengujian jumlah popsize dapat dilihat pada Lampiran A.

6.2 Pengujian Cr & Mr

Pada pengujian kombinasi nilai *cr* (*crossover rate*) dan *mr* (*mutation rate*), digunakan perbandingan nilai *cr* dan *mr* yaitu 0.1:1 sampai 1:0.1. Nilai *cr* dan *mr* merupakan nilai dengan interval 0 sampai dengan 1 yang digunakan untuk menentukan *offspring* yang dihasilkan dari proses *crossover* dan *mutation*. Dari pengujian jumlah popsize yang telah dilakukan, dapat diambil kesimpulan untuk jumlah popsize optimal yang akan digunakan pada pengujian ini ditampilkan pada Tabel 6.1.

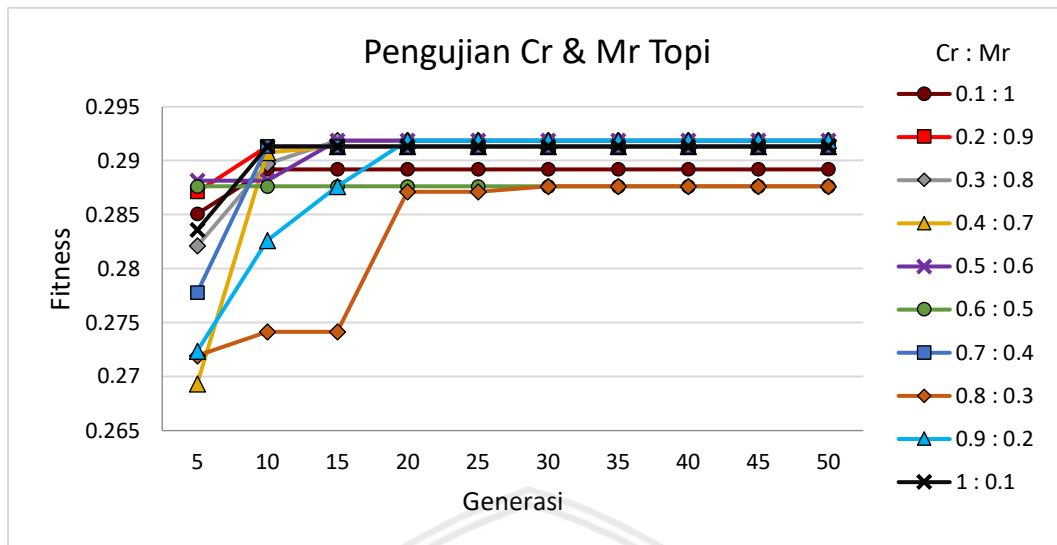
Tabel 6.1 Jumlah Popsi Optimal

Produk	Popsi	Waktu Komputasi
Topi	40	2.1 detik
Sabuk	70	5.3 detik
Dompot	70	5.3 detik
Tas	60	3.8 detik
Jaket	70	5.1 detik
Sepatu	30	1.1 detik

Jumlah popsize pada Gambar 6.1, didapatkan dari pengujian yang menghasilkan nilai *fitness* terbesar sejak dari generasi awal dan waktu komputasi yang digunakan singkat.

6.2.1 Pengujian Produk Topi

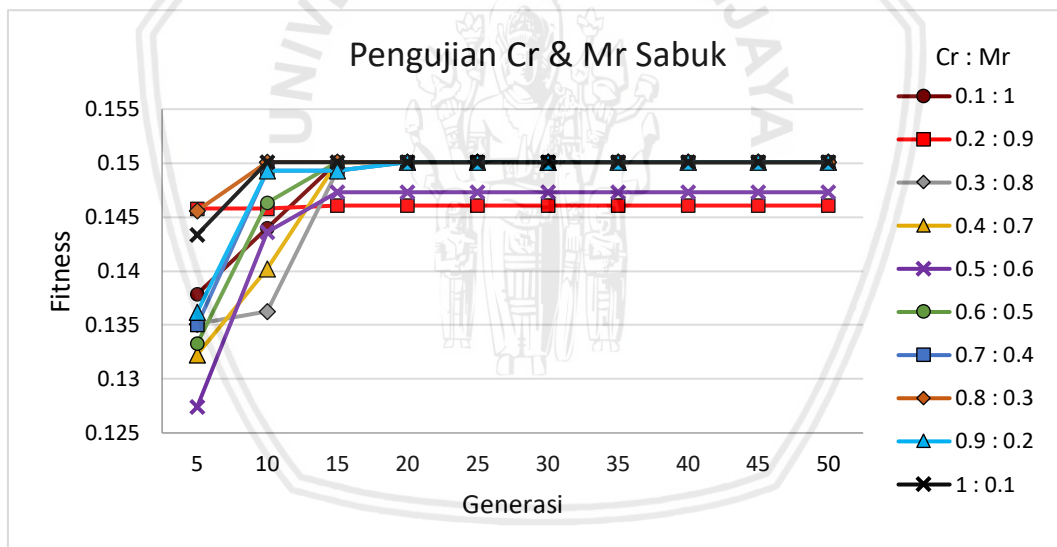
Pada Gambar 6.7, ditampilkan grafik hasil dari pengujian kombinasi nilai *cr* & *mr* pada produk topi yang menunjukkan pengujian dengan kombinasi 0.4:0.7 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.285082593 dan grafik mengalami konvergen pada generasi ke-15 dengan nilai *fitness* sebesar 0.289199522. Sedangkan pada nilai kombinasi 0.3:0.8 dan 0.5:0.6, menghasilkan nilai *fitness* terbesar di generasi ke-15 yang bernilai 0.291865012. Kemudian hanya pada kombinasi 0.6:0.5 yang menghasilkan nilai *fitness* sebesar 0.28763486 pada setiap generasinya.



Gambar 6.7 Pengujian Cr & Mr Topi

6.2.2 Pengujian Produk Sabuk

Pada Gambar 6.8, ditampilkan grafik hasil dari pengujian nilai kombinasi *cr* & *mr* pada produk sabuk.

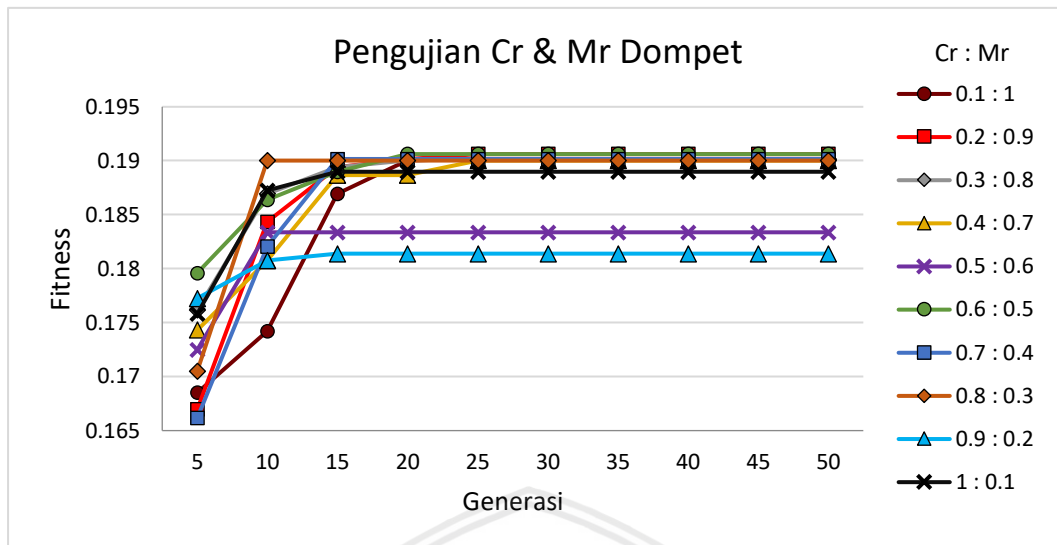


Gambar 6.8 Pengujian Cr & Mr Sabuk

Pada grafik hasil Gambar 6.8, menunjukkan pengujian dengan kombinasi 0.5:0.6 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.127402806 dan grafik mengalami konvergen pada generasi ke-15 dengan nilai *fitness* sebesar 0.147302794. Sedangkan pada nilai kombinasi 0.8:0.3 dan 1:0.1, menghasilkan nilai *fitness* terbesar di generasi ke-10 yang bernilai 0.150089754. Kemudian hanya pada nilai kombinasi 0.2:0.9 dan 0.5:0.6 yang tidak menghasilkan nilai *fitness* terbesar.

6.2.3 Pengujian Produk Dompot

Pada Gambar 6.9, ditampilkan grafik hasil dari pengujian nilai kombinasi *cr* & *mr* pada produk dompet.

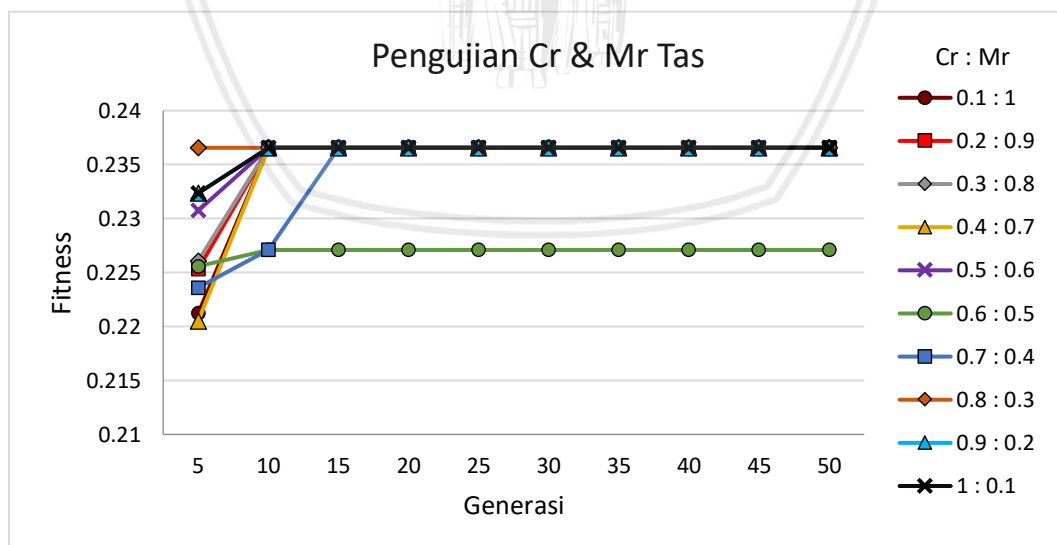


Gambar 6.9 Pengujian Cr & Mr Dompot

Pada grafik hasil Gambar 6.9, menunjukkan pengujian dengan kombinasi 0.7:0.4 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.166165714 dan grafik mengalami konvergen pada generasi ke-15 dengan nilai *fitness* sebesar 0.19016709. Sedangkan pada nilai kombinasi 0.6:0.5, menghasilkan nilai *fitness* terbesar di generasi ke-20 yang bernilai 0.19061719. Kemudian nilai kombinasi lain yang menghasilkan nilai *fitness* terbesar yaitu pada kombinasi 0.2:0.9 di generasi ke-25.

6.2.4 Pengujian Produk Tas

Pada Gambar 6.10, ditampilkan grafik hasil dari pengujian nilai kombinasi *cr* & *mr* pada produk tas.



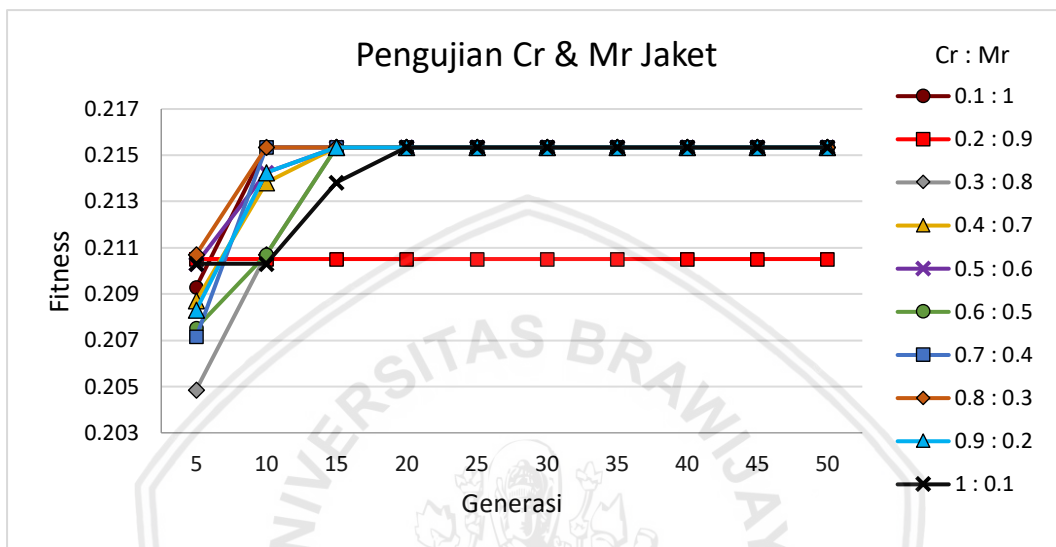
Gambar 6.10 Pengujian Cr & Mr Tas

Pada grafik hasil Gambar 6.10, menunjukkan pengujian dengan kombinasi 0.4:0.7 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.220512568 dan grafik mengalami konvergen pada generasi ke-10 yang menghasilkan nilai *fitness* terbesar yang bernilai 0.236560924. Kemudian pada

kombinasi 0.8:0.3 menghasilkan nilai *fitness* terbesar sejak dari generasi ke-5, sedangkan hanya pada nilai kombinasi 0.6:0.5 yang konvergen pada nilai *fitness* sebesar 0.227089204.

6.2.5 Pengujian Produk Jacket

Pada Gambar 6.11, ditampilkan grafik hasil dari pengujian nilai kombinasi *cr* & *mr* produk jacket.

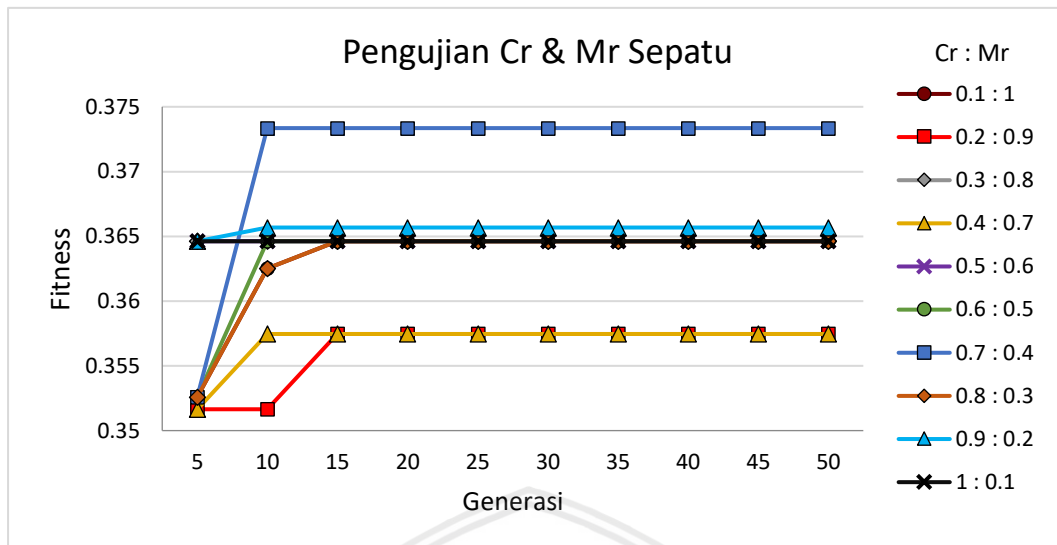


Gambar 6.11 Pengujian Cr & Mr Jacket

Pada grafik hasil Gambar 6.11, menunjukkan pengujian dengan kombinasi 0.3:0.8 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.204867769 dan grafik mengalami konvergen pada generasi ke-15 dengan nilai *fitness* sebesar 0.215339192. Sedangkan pada nilai kombinasi 0.1:1, 0.7:0.4 dan 0.8:0.3, menghasilkan nilai *fitness* terbesar di generasi ke-10 yang bernilai 0.215339192. Kemudian hanya pada kombinasi 0.2:0.9 yang menghasilkan nilai *fitness* sebesar 0.210513639 pada setiap generasinya.

6.2.6 Pengujian Produk Sepatu

Pada Gambar 6.12, menampilkan grafik hasil dari pengujian kombinasi nilai *cr* & *mr* pada produk sepatu yang menunjukkan pengujian dengan kombinasi 0.2:0.9 dan 0.4:0.7 menghasilkan nilai *fitness* terkecil pada generasi ke-5 yaitu sebesar 0.351647429 dan masing-masing grafik mengalami konvergen pada generasi ke-10 dan ke-15 dengan nilai *fitness* sebesar 0.357460176. Sedangkan hanya pada nilai kombinasi 0.7:0.4 yang menghasilkan nilai *fitness* terbesar di generasi ke-10 yang bernilai 0.373354978.



Gambar 6.12 Penguujian Cr & Mr Sepatu

Dari hasil pengujian nilai kombinasi *cr* & *mr* pada masing-masing produk, dapat dilihat bahwa nilai kombinasi 0.4:0.7 menghasilkan nilai *fitness* terkecil pada 3 pengujian yaitu pada produk topi, tas dan sepatu. Sedangkan nilai kombinasi 0.8:0.3 menghasilkan nilai *fitness* terbesar pada 3 pengujian yaitu pada produk sabuk, tas dan jaket. Untuk hasil lebih lengkap berupa tabel pada pengujian nilai kombinasi *cr* & *mr* dapat dilihat pada Lampiran B.

6.3 Pengujian Tingkat Akurasi

Dari nilai kombinasi *cr* dan *mr* yang telah dilakukan, dapat diambil kesimpulan untuk nilai parameter algoritme genetika yang optimal pada setiap produk ditampilkan pada Tabel 6.2.

Tabel 6.2 Parameter Optimal Algoritme Genetika

Produk	Popsi	Cr : Mr
Topi	40	0.5 : 0.6
Sabuk	70	0.8 : 0.3
Dompot	70	0.6 : 0.5
Tas	60	0.8 : 0.3
Jaket	70	0.7 : 0.4
Sepatu	30	0.7 : 0.4

Nilai kombinasi *cr* dan *mr* pada Tabel 6.2, didapatkan dari pengujian yang menghasilkan nilai *fitness* terbesar sejak dari generasi awal dan juga nilai *cr* dan *mr* yang berdekatan. Setelah didapatkan parameter optimal algoritme genetika pada masing-masing produk, kemudian dilakukan pengujian tingkat akurasi dengan menggunakan parameter pada Tabel 6.2 dengan menggunakan metode RMSPE (*Root Mean Square Percentage Error*) pada Persamaan 2.8 dan untuk mendapatkan hasil akurasi, yaitu $1 - \text{RMSPE}$.

6.3.1 Tingkat Akurasi Produk Topi

Pada Tabel 6.3, menampilkan hasil prediksi dan tingkat akurasi pada produk topi.

Tabel 6.3 Tingkat Akurasi Topi

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Ferbuari	45	40	0.11111111	0.01234568
	Maret	43	45	-0.0465116	0.00216333
	April	34	33	0.02941176	0.00086505
...
2017	Oktober	38	40	-0.0526316	0.00277008
	November	36	40	-0.11111111	0.01234568
	Desember	42	40	0.04761905	0.00226757
Jumlah					0.183756467

Pada Tabel 6.3 setelah didapatkan jumlah dari hasil kuadrat *error*, untuk mendapatkan tingkat akurasi dihitung RMSPE sebagai berikut:

$$\begin{aligned} \text{RMSPE} &= \sqrt{\frac{0.183756467}{23}} \times 100 \\ &= 8.94\% \end{aligned}$$

Dari perhitungan diatas, didapatkan hasil RMSPE sebesar 8.94%. Kemudian menghitung tingkat akurasi yaitu 1 – 8.94%, yang menghasilkan tingkat akurasi sebesar 91.06%.

6.3.2 Tingkat Akurasi Produk Sabuk

Pada Tabel 6.4, menampilkan hasil prediksi dan tingkat akurasi pada produk sabuk.

Tabel 6.4 Tingkat Akurasi Sabuk

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Ferbuari	52	61	-0.1730769	0.02995562
	Maret	59	65	-0.1016949	0.01034186
	April	53	61	-0.1509434	0.02278391
...
2017	Oktober	75	77	-0.0266667	0.00071111
	November	54	55	-0.0185185	0.00034294
	Desember	65	65	0	0
Jumlah					0.24447727

Pada Tabel 6.4 setelah didapatkan jumlah dari hasil kuadrat *error*, untuk mendapatkan tingkat akurasi dihitung RMSPE sebagai berikut:

$$\begin{aligned} \text{RMSPE} &= \sqrt{\frac{0.24447727}{23}} \times 100 \\ &= 10.31\% \end{aligned}$$

Dari perhitungan diatas, didapatkan hasil RMSPE sebesar 10.31%. Kemudian menghitung tingkat akurasi yaitu 1 – 10.31%, yang menghasilkan tingkat akurasi sebesar 89.69%.

6.3.3 Tingkat Akurasi Produk Dompot

Pada Tabel 6.5, menampilkan hasil prediksi dan tingkat akurasi pada produk dompot.

Tabel 6.5 Tingkat Akurasi Dompot

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Ferbuari	163	151	0.07361963	0.00541985
	Maret	153	154	-0.0065359	0.000042719
	April	141	144	-0.0212766	0.00045269
...
2017	Oktober	149	146	0.02013423	0.00040539
	November	144	146	-0.0138889	0.0001929
	Desember	144	146	-0.0138889	0.0001929
Jumlah					0.02911136

Pada Tabel 6.5 setelah didapatkan jumlah dari hasil kuadrat *error*, untuk mendapatkan tingkat akurasi dihitung RMSPE sebagai berikut:

$$\begin{aligned} \text{RMSPE} &= \sqrt{\frac{0.02911136}{23}} \times 100 \\ &= 3.56\% \end{aligned}$$

Dari perhitungan diatas, didapatkan hasil RMSPE sebesar 3.56%. Kemudian menghitung tingkat akurasi yaitu 1 – 3.56%, yang menghasilkan tingkat akurasi sebesar 96.44%.

6.3.4 Tingkat Akurasi Produk Tas

Pada Tabel 6.6, menampilkan hasil prediksi dan tingkat akurasi pada produk tas.

Tabel 6.6 Tingkat Akurasi Tas

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Ferbuari	73	67	0.08219178	0.00675549
	Maret	70	67	0.04285714	0.00183673
	April	57	56	0.01754386	0.00030779
...
2017	Oktober	61	68	-0.1147541	0.0131685
	November	73	68	0.06849315	0.00469131
	Desember	69	67	0.02898551	0.00084016
Jumlah					0.09569713

Pada Tabel 6.6 setelah didapatkan jumlah dari hasil kuadrat *error*, untuk mendapatkan tingkat akurasi dihitung RMSPE sebagai berikut:

$$\begin{aligned} \text{RMSPE} &= \sqrt{\frac{0.09569713}{23}} \times 100 \\ &= 6.45\% \end{aligned}$$

Dari perhitungan diatas, didapatkan hasil RMSPE sebesar 6.45%. Kemudian menghitung tingkat akurasi yaitu 1 – 6.45%, yang menghasilkan tingkat akurasi sebesar 93.55%.

6.3.5 Tingkat Akurasi Produk Jacket

Pada Tabel 6.7, menampilkan hasil prediksi dan tingkat akurasi pada produk jacket.

Tabel 6.7 Tingkat Akurasi Jacket

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Ferbuari	66	77	-0.1666667	0.02777778
	Maret	75	77	-0.0266667	0.00071111
	April	73	73	0	0
...
2017	Oktober	69	73	-0.057971	0.00336064
	November	79	81	-0.0253165	0.00064092
	Desember	84	79	0.05952381	0.00354308
Jumlah					0.09205015

Pada Tabel 6.7 setelah didapatkan jumlah dari hasil kuadrat *error*, untuk mendapatkan tingkat akurasi dihitung RMSPE sebagai berikut:

$$\begin{aligned} \text{RMSPE} &= \sqrt{\frac{0.09205015}{23}} \times 100 \\ &= 6.33\% \end{aligned}$$

Dari perhitungan diatas, didapatkan hasil RMSPE sebesar 6.33%. Kemudian menghitung tingkat akurasi yaitu 1 – 6.33%, yang menghasilkan tingkat akurasi sebesar 93.67%.

6.3.6 Tingkat Akurasi Produk Sepatu

Pada Tabel 6.8, menampilkan hasil prediksi dan tingkat akurasi pada produk sepatu.

Tabel 6.8 Tingkat Akurasi Sepatu

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Ferbuari	37	41	-0.1081081	0.01168736
	Maret	39	41	-0.0512821	0.00262985
	April	37	36	0.02702703	0.00073046
...
2017	Oktober	36	36	0	0
	November	45	41	0.08888889	0.00790123
	Desember	43	40	0.06976744	0.0048675
Jumlah					0.10025869

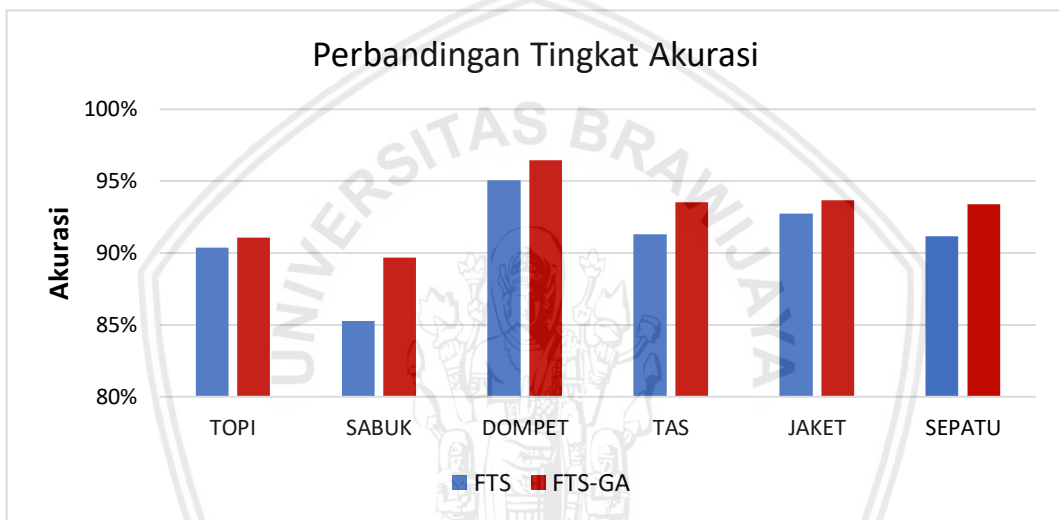
Pada Tabel 6.8 setelah didapatkan jumlah dari hasil kuadrat *error*, untuk mendapatkan tingkat akurasi dihitung RMSPE sebagai berikut:

$$\begin{aligned} \text{RMSPE} &= \sqrt{\frac{0.10025869}{23}} \times 100 \\ &= 6.60\% \end{aligned}$$

Dari perhitungan diatas, didapatkan hasil RMSPE sebesar 6.60%. Kemudian menghitung tingkat akurasi yaitu 1 – 6.60%, yang menghasilkan tingkat akurasi sebesar 93.40%.

6.4 Hasil Analisis

Dari pengujian yang telah dilakukan, dapat dilihat bahwa jumlah popsize mempengaruhi nilai *fitness* yang dihasilkan. Semakin banyak jumlah popsize yang digunakan, maka semakin tinggi kemungkinan untuk menghasilkan individu terbaik pada generasi awal dikarenakan *offspring* yang dihasilkan bervariasi untuk dipilih dalam generasi selanjutnya. Tetapi, jika jumlah popsize terlalu banyak maka waktu komputasi yang dibutuhkan pun semakin lama. Sedangkan untuk nilai kombinasi *cr* dan *mr*, menghasilkan individu terbaik ketika nilai *cr* lebih besar dari nilai *mr* dikarenakan nilai kromosom *offspring* yang dihasilkan dari proses *crossover* memiliki perbedaan yang cukup besar dengan *parentnya*, yang mempengaruhi nilai *fitness* yang dihasilkan. Kemudian dari individu terbaik tersebut menghasilkan tingkat akurasi diatas 90% (kecuali pada produk sabuk).



Gambar 6.13 Perbandingan Tingkat Akurasi

Pada Gambar 6.13, ditampilkan grafik hasil perbandingan tingkat akurasi menggunakan FTS (*Fuzzy Time Series*) dan FTS-GA (*Fuzzy Time Series - Genetic Algorithm*) dari pengujian pada bulan Februari 2016 sampai Desember 2017. Pada Tabel 6.9 ditampilkan tabel dari perbandingan tingkat akurasi.

Tabel 6.9 Hasil Tingkat Akurasi

	TOPI	SABUK	DOMPET	TAS	JAKET	SEPATU
FTS	90.38%	85.29%	95.06%	91.29%	92.72%	91.17%
FTS-GA	91.06%	89.69%	96.44%	93.55%	93.67%	93.40%

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan pada optimasi algoritme genetika dalam memprediksi jumlah produksi SAGA *leather fashion* menggunakan *fuzzy time series*, dapat diambil kesimpulan sebagai berikut:

1. Pada penelitian ini, menghasilkan prediksi yang sangat baik dengan tingkat akurasi rata-rata sebesar 92.97% dari masing- masing produk.
2. Pada penelitian ini, nilai parameter algoritme genetika yang optimal sebagai berikut: produk topi jumlah popsize = 40, nilai kombinasi cr dan mr = 0.5 dan 0.6, waktu komputasi = 2.1 detik yang menghasilkan nilai *fitness* sebesar 0.291865012, produk sabuk jumlah popsize = 70, nilai kombinasi cr dan mr = 0.8 dan 0.3, waktu komputasi = 5.3 detik yang menghasilkan nilai *fitness* sebesar 0.150089754, produk dompet jumlah popsize = 70, nilai kombinasi cr dan mr = 0.6 dan 0.5, waktu komputasi = 5.3 detik yang menghasilkan nilai *fitness* sebesar 0.19061719, produk tas jumlah popsize = 60, nilai kombinasi cr dan mr = 0.8 dan 0.3, waktu komputasi = 3.8 detik yang menghasilkan nilai *fitness* sebesar 0.236560924. produk jaket jumlah popsize = 70, nilai kombinasi cr dan mr = 0.7 dan 0.4, waktu komputasi - 5.1 detik yang menghasilkan nilai *fitness* sebesar 0.215339192 dan produk sepatu jumlah popsize = 30, nilai kombinasi cr dan mr = 0.7 dan 0.4, waktu komputasi = 1.1 detik yang menghasilkan nilai *fitness* sebesar 0.373354978.
3. *Fuzzy time series* yang dioptimasi menggunakan algoritme genetika menghasilkan nilai rata-rata *error* dari masing-masing produk yang lebih kecil sebesar 4.480755246 dibandingkan dengan hanya menggunakan *fuzzy time series* sebesar 5.764766752.

7.2 Saran

Berdasarkan hasil kesimpulan yang telah didapatkan pada optimasi algoritme genetika dalam memprediksi jumlah produksi SAGA *leather fashion* menggunakan *fuzzy time series*, saran yang dapat diberikan untuk penelitian selanjutnya yaitu:

1. Mencoba menggunakan metode *crossover*, *mutation* dan seleksi pada algoritme genetika lainnya, untuk mengetahui perbedaan nilai *fitness* yang dihasilkan.

DAFTAR PUSTAKA

- Aladag, C. H., Yolcu, U., Egrioglu, E. & Bas, E., 2014. Fuzzy Lagged Variable Selection in Fuzzy Time Series with Genetic Algorithms. *Applied Soft Computing*, Volume 22, pp. 465-473.
- Armstrong, J. S. & Collopy, F., 1992. *International Journal of Forecasting*. Error measures for generalizing about forecasting methods: Empirical comparisons, 08(1), pp. 69-80.
- Barnard, M., 1996. *Fashion as Communication*. Routledge: s.n.
- Cai, Q., Zhang, D., Wu, B. & Leung, S. C., 2013. A Novel Stock Forecasting Model based on Fuzzy Time Series and Genetic Algorithm. *Procedia Computer Science*, Volume 18, pp. 1155-1162.
- Chen, M., 1996. Forecasting Enrollments Based on Fuzzy Time Series. *Fuzzy Sets and Systems*, Volume 81, pp. 311-319.
- Dharmaputra, B., 2012. *Sistem Penjadwalan Mata Kuliah Dengan Algoritme Genetika*.
- Engineers, N. B. o. C., 2011. *Leather Processing & Tanning Technology Handbook*. s.l.:NIIR Project Consultancy Services.
- Evans, C., 1997. Fashion of Communication. *Journal of Design History*, 1 January, 10(2), pp. 231-233.
- Haupt, R. L. & Haupt, S. E., 2004. *Practical Genetic Algorithm*. 2nd Edition penyunt. s.l.:J. Wiley.
- Hendariningrum, R. & Susilo, E., 2008. Fashion Dan Gaya Hidup: Identitas Dan Komunikasi. *Jurnal Ilmu Komunikasi FISIP UPN*, 6(2), pp. 25-32.
- Ibrahim, A. M., 2004. *Fuzzy Logic for Embedded Systems Applications Chapter 2: Fuzzy Sets*. s.l.:s.n.
- Istiqara, K., Furqon, M. T. & Indriati, I., 2017. Prediksi Kebutuhan Air PDAM Kota Malang Menggunakan Metode Fuzzy Time Series Dengan Algoritma Genetika. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(1), pp. 133-142.
- Kayacan, E. & Khanesar, M. A., 2016. *Fuzzy Neural Networks for Real Time Control Applications*. s.l.:s.n.
- Land, K. C., 2015. Forecasting. Dalam: J. D. Wright, penyunt. *International Encyclopedia of the Social & Behavioral Sciences*. 2nd Edition penyunt. s.l.:s.n., pp. 316-323.
- Mahmudy, W. F., 2015. *Dasar-dasar Algoritme Evolusi*. Malang: FILKOM UB.
- Mandariansah, T., Setiawan, B. D. & Wihandika, R. C., 2017. Optimasi Fuzzy Time Series Untuk Peramalan Kebutuhan Hidup Layak Kota Kediri Dengan

- Menggunakan Algoritme Genetika. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(5), pp. 1823-1832.
- Mitchell, M., 1998. *An Introduction to Genetic Algorithms*. s.l.:MIT Press.
- Pykacz, J., 2007. Quantum Structures and Fuzzy Set Theory. Dalam: K. Engesser, D. M. Gabbay & D. Lehmann, penyunt. *Handbook of Quantum Logic and Quantum Structures*. s.l.:s.n., pp. 55-74.
- Ridhwan, A., Ratnawati, D. E. & Rahayudi, B., 2017. Peramalan Produksi Gula Pasir Menggunakan Fuzzy Time Series Dengan Optimasi Algoritma Genetika (Studi Kasus PG Candi Baru Sidoarjo). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(8), pp. 2542-2548.
- Rob J. Hyndman, A. B. K., 2006. Another Look at Measures of Forecast Accuracy. *International Journal of Forecasting*, 22(4), p. 679–688.
- Song, Q. & Chissom, B. S., 1993. Fuzzy Time Series and Its Models. *Fuzzy Sets and Systems*, Volume 54, pp. 269-277.
- Wicaksana, A. S., Setiawan, B. D. & Rahayudi, B., 2017. Algoritma Genetika Untuk Optimasi Fuzzy Time Series Dalam Memprediksi Kepadatan Lalu Lintas di Jalan Tol. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(3), pp. 1063-1071.
- Zadeh, L., 1965. Fuzzy Sets. *Information and Control*, 8(3), pp. 338-353.

LAMPIRAN A DATA PENGUJIAN POPSIZE

A.1 Produk Topi

Generasi	Jumlah Popsi									
	10	20	30	40	50	60	70	80	90	100
5	0.2680951	0.2693607	0.2710764	0.2845802	0.2750598	0.29079	0.2881536	0.2876349	0.2825971	0.280175
10	0.2680951	0.2741594	0.291326	0.291865	0.29079	0.29079	0.2891995	0.29079	0.2891995	0.291326
15	0.2689369	0.2741594	0.291326	0.291865	0.291326	0.29079	0.2897268	0.29079	0.2891995	0.291865
20	0.2689369	0.2741594	0.291326	0.291865	0.291326	0.29079	0.291326	0.29079	0.2891995	0.291865
25	0.2689369	0.2741594	0.291326	0.291865	0.291326	0.29079	0.291865	0.29079	0.2891995	0.291865
30	0.2689369	0.2741594	0.291326	0.291865	0.291326	0.29079	0.291865	0.29079	0.2891995	0.291865
35	0.2689369	0.2741594	0.291326	0.291865	0.291326	0.29079	0.291865	0.29079	0.2891995	0.291865
40	0.2689369	0.2741594	0.291326	0.291865	0.291326	0.29079	0.291865	0.29079	0.2891995	0.291865
45	0.2689369	0.2741594	0.291326	0.291865	0.291326	0.29079	0.291865	0.29079	0.2891995	0.291865
50	0.2689369	0.2741594	0.291326	0.291865	0.291326	0.29079	0.291865	0.29079	0.2891995	0.291865
Waktu Komputasi (detik)	0.6	1	1.7	2.1	2.8	3.8	5.3	6.9	8.3	10.2

A.2 Produk Sabuk

Generasi	Jumlah Popsi									
	10	20	30	40	50	60	70	80	90	100
5	0.1192639	0.1250425	0.1250425	0.1258146	0.1358097	0.131159	0.1355382	0.1317019	0.1433029	0.133943
10	0.1193008	0.1307684	0.1264252	0.1402074	0.1438174	0.1387911	0.1426674	0.1462715	0.1490712	0.142102
15	0.1198958	0.1307684	0.1492877	0.1455959	0.1492877	0.1438174	0.1500898	0.1463396	0.1490712	0.147302
20	0.124201	0.1461356	0.1492877	0.1457976	0.1500898	0.1492877	0.1500898	0.1463396	0.1490712	0.147302
25	0.1245776	0.1492877	0.1500898	0.1457976	0.1500898	0.1492877	0.1500898	0.1463396	0.1500898	0.147302

Generasi	Jumlah Popsi									
	10	20	30	40	50	60	70	80	90	100
30	0.1245776	0.1500898	0.1500898	0.1457976	0.1500898	0.1500898	0.1500898	0.1463396	0.1500898	0.147302
35	0.1245776	0.1500898	0.1500898	0.1457976	0.1500898	0.1500898	0.1500898	0.1463396	0.1500898	0.147302
40	0.1267338	0.1500898	0.1500898	0.1457976	0.1500898	0.1500898	0.1500898	0.1463396	0.1500898	0.147302
45	0.1267781	0.1500898	0.1500898	0.1457976	0.1500898	0.1500898	0.1500898	0.1463396	0.1500898	0.147302
50	0.1307684	0.1500898	0.1500898	0.1457976	0.1500898	0.1500898	0.1500898	0.1463396	0.1500898	0.147302
Waktu Komputasi (detik)	0.2	0.6	1.2	1.8	2.7	3.9	5.3	6.7	8.3	10.4

A.3 Produk Dompot

Generasi	Jumlah Popsi									
	10	20	30	40	50	60	70	80	90	100
5	0.1654719	0.1577713	0.1522678	0.1642061	0.1646895	0.1692412	0.1730547	0.186961	0.1802384	0.172605
10	0.170305	0.1724941	0.1676823	0.175942	0.1845916	0.1793539	0.1901671	0.186961	0.1808782	0.179353
15	0.1730547	0.1797314	0.1782357	0.1872458	0.1863953	0.186961	0.1901671	0.1900178	0.1808782	0.190167
20	0.1733937	0.1807497	0.1807497	0.1901671	0.1901671	0.1900178	0.1901671	0.1901671	0.181395	0.190167
25	0.1733937	0.189424	0.181395	0.1901671	0.1901671	0.1901671	0.1901671	0.1901671	0.1866775	0.190167
30	0.1807497	0.189424	0.181395	0.1901671	0.1901671	0.1901671	0.1901671	0.1901671	0.1872458	0.190167
35	0.1807497	0.1901671	0.181395	0.1901671	0.1901671	0.1901671	0.1901671	0.1901671	0.1872458	0.190167
40	0.1807497	0.1901671	0.181395	0.1901671	0.1901671	0.1901671	0.1901671	0.1901671	0.1901671	0.190167
45	0.1807497	0.1901671	0.181395	0.1901671	0.1901671	0.1901671	0.1901671	0.1901671	0.1901671	0.190167
50	0.1807497	0.1901671	0.181395	0.1901671	0.1901671	0.1901671	0.1901671	0.1901671	0.1901671	0.190167
Waktu Komputasi (detik)	0.7	1.2	1.7	2.2	3.2	3.9	5.3	6.8	8.3	10.2

A.4 Produk Tas

Generasi	Jumlah Popsi									
	10	20	30	40	50	60	70	80	90	100
5	0.2081214	0.2238502	0.2069554	0.2202798	0.2255769	0.2302073	0.2205126	0.2323588	0.2320866	0.232086
10	0.2087118	0.2320866	0.2270892	0.224832	0.2323588	0.2365609	0.2365609	0.2365609	0.2365609	0.236560
15	0.2087118	0.2365609	0.2270892	0.2255769	0.2323588	0.2365609	0.2365609	0.2365609	0.2365609	0.236560
20	0.2087118	0.2365609	0.2270892	0.2255769	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.236560
25	0.2087118	0.2365609	0.2270892	0.2255769	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.236560
30	0.2087118	0.2365609	0.2270892	0.2255769	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.236560
35	0.2087118	0.2365609	0.2270892	0.2255769	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.236560
40	0.2087118	0.2365609	0.2270892	0.2255769	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.236560
45	0.2087118	0.2365609	0.2270892	0.2255769	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.236560
50	0.2087118	0.2365609	0.2270892	0.2255769	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.236560
Waktu Komputasi (detik)	0.2	0.6	1.1	1.9	2.8	3.8	5.2	6.6	8.2	10.3

A.5 Produk Jaket

Generasi	Jumlah Popsi									
	10	20	30	40	50	60	70	80	90	100
5	0.1929166	0.2071484	0.2021202	0.2052426	0.2073419	0.2099078	0.2095069	0.2085144	0.2099078	0.208514
10	0.195626	0.2105136	0.2101092	0.2105136	0.2138356	0.2101092	0.2153392	0.2153392	0.2138356	0.213835
15	0.195626	0.2105136	0.2103111	0.214262	0.2138356	0.214262	0.2153392	0.2153392	0.2153392	0.215339
20	0.2058088	0.2105136	0.2105136	0.2153392	0.214262	0.214262	0.2153392	0.2153392	0.2153392	0.215339
25	0.2058088	0.2105136	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339
30	0.2058088	0.2105136	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339
35	0.2058088	0.2105136	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339

Generasi	Jumlah Popsize									
	10	20	30	40	50	60	70	80	90	100
40	0.2058088	0.2105136	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339
45	0.2058088	0.2105136	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339
50	0.2058088	0.2105136	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339
Waktu Komputasi (detik)	0.2	0.6	1.1	1.9	2.7	3.8	5.1	6.6	8.2	10.3

A.6 Produk Sepatu

Generasi	Jumlah Popsize									
	10	20	30	40	50	60	70	80	90	100
5	0.3255623	0.3452115	0.3574602	0.3564713	0.3656787	0.3646203	0.3656787	0.3604768	0.3554907	0.360476
10	0.327073	0.3574602	0.373355	0.3656787	0.373355	0.3656787	0.3656787	0.3656787	0.3646203	0.373355
15	0.327073	0.3574602	0.373355	0.3656787	0.373355	0.3656787	0.3656787	0.3656787	0.3656787	0.373355
20	0.327073	0.3646203	0.373355	0.3656787	0.373355	0.3656787	0.3656787	0.3656787	0.3656787	0.373355
25	0.3374335	0.3646203	0.373355	0.3656787	0.373355	0.3656787	0.3656787	0.3656787	0.3656787	0.373355
30	0.3399675	0.373355	0.373355	0.3656787	0.373355	0.3656787	0.3656787	0.3656787	0.3656787	0.373355
35	0.3554907	0.373355	0.373355	0.3656787	0.373355	0.3656787	0.3656787	0.3656787	0.3656787	0.373355
40	0.3574602	0.373355	0.373355	0.3656787	0.373355	0.3656787	0.3656787	0.3656787	0.3656787	0.373355
45	0.3574602	0.373355	0.373355	0.3656787	0.373355	0.3656787	0.3656787	0.3656787	0.3656787	0.373355
50	0.3574602	0.373355	0.373355	0.3656787	0.373355	0.3656787	0.3656787	0.3656787	0.3656787	0.373355
Waktu Komputasi (detik)	0.2	0.6	1.1	1.8	2.8	3.8	5.2	6.7	8.3	10.3

LAMPIRAN B DATA PENGUJIAN CROSSOVER RATE DAN MUTATION RATE

B.1 Produk Topi

Generasi	Kombinasi cr : mr									
	0.1 : 1	0.2 : 0.9	0.3 : 0.8	0.4 : 0.7	0.5 : 0.6	0.6 : 0.5	0.7 : 0.4	0.8 : 0.3	0.9 : 0.2	1 : 0.1
5	0.2850826	0.2871189	0.2821077	0.2693607	0.2881536	0.2876349	0.2778151	0.2719467	0.2723849	0.283583
10	0.2891995	0.291326	0.2897268	0.29079	0.2881536	0.2876349	0.291326	0.2741594	0.2825971	0.291326
15	0.2891995	0.291326	0.291865	0.291326	0.291865	0.2876349	0.291326	0.2741594	0.2876349	0.291326
20	0.2891995	0.291326	0.291865	0.291326	0.291865	0.2876349	0.291326	0.2871189	0.291865	0.291326
25	0.2891995	0.291326	0.291865	0.291326	0.291865	0.2876349	0.291326	0.2871189	0.291865	0.291326
30	0.2891995	0.291326	0.291865	0.291326	0.291865	0.2876349	0.291326	0.2876349	0.291865	0.291326
35	0.2891995	0.291326	0.291865	0.291326	0.291865	0.2876349	0.291326	0.2876349	0.291865	0.291326
40	0.2891995	0.291326	0.291865	0.291326	0.291865	0.2876349	0.291326	0.2876349	0.291865	0.291326
45	0.2891995	0.291326	0.291865	0.291326	0.291865	0.2876349	0.291326	0.2876349	0.291865	0.291326
50	0.2891995	0.291326	0.291865	0.291326	0.291865	0.2876349	0.291326	0.2876349	0.291865	0.291326
Waktu Komputasi	2.1 detik									

B.2 Produk Sabuk

Generasi	Kombinasi cr : mr									
	0.1 : 1	0.2 : 0.9	0.3 : 0.8	0.4 : 0.7	0.5 : 0.6	0.6 : 0.5	0.7 : 0.4	0.8 : 0.3	0.9 : 0.2	1 : 0.1
5	0.1378705	0.1457976	0.1351072	0.1322516	0.1274028	0.133269	0.1350001	0.1455959	0.1361925	0.143366
10	0.1440118	0.1457976	0.1362474	0.1402074	0.1436238	0.1463396	0.1492877	0.1500898	0.1492877	0.150089
15	0.1500898	0.1460678	0.1492877	0.1500898	0.1473028	0.1500898	0.1492877	0.1500898	0.1492877	0.150089
20	0.1500898	0.1460678	0.1500898	0.1500898	0.1473028	0.1500898	0.1500898	0.1500898	0.1500898	0.150089
25	0.1500898	0.1460678	0.1500898	0.1500898	0.1473028	0.1500898	0.1500898	0.1500898	0.1500898	0.150089
30	0.1500898	0.1460678	0.1500898	0.1500898	0.1473028	0.1500898	0.1500898	0.1500898	0.1500898	0.150089

Generasi	Kombinasi cr : mr									
	0.1 : 1	0.2 : 0.9	0.3 : 0.8	0.4 : 0.7	0.5 : 0.6	0.6 : 0.5	0.7 : 0.4	0.8 : 0.3	0.9 : 0.2	1 : 0.1
35	0.1500898	0.1460678	0.1500898	0.1500898	0.1473028	0.1500898	0.1500898	0.1500898	0.1500898	0.1500898
40	0.1500898	0.1460678	0.1500898	0.1500898	0.1473028	0.1500898	0.1500898	0.1500898	0.1500898	0.1500898
45	0.1500898	0.1460678	0.1500898	0.1500898	0.1473028	0.1500898	0.1500898	0.1500898	0.1500898	0.1500898
50	0.1500898	0.1460678	0.1500898	0.1500898	0.1473028	0.1500898	0.1500898	0.1500898	0.1500898	0.1500898
Waktu Komputasi	5.3 detik									

B.3 Produk Dompert

Generasi	Kombinasi cr : mr									
	0.1 : 1	0.2 : 0.9	0.3 : 0.8	0.4 : 0.7	0.5 : 0.6	0.6 : 0.5	0.7 : 0.4	0.8 : 0.3	0.9 : 0.2	1 : 0.1
5	0.1685083	0.1669694	0.1761793	0.1743075	0.1724941	0.1796053	0.1661657	0.1705201	0.177259	0.175823
10	0.1741924	0.1843188	0.186961	0.1807497	0.1833732	0.1863953	0.1820473	0.1900178	0.1807497	0.187245
15	0.186961	0.189424	0.189424	0.1886895	0.1833732	0.1889822	0.1901671	0.1900178	0.181395	0.188982
20	0.1900178	0.1901671	0.1900178	0.1886895	0.1833732	0.1906172	0.1901671	0.1900178	0.181395	0.188982
25	0.1901671	0.1906172	0.1901671	0.1900178	0.1833732	0.1906172	0.1901671	0.1900178	0.181395	0.188982
30	0.1901671	0.1906172	0.1901671	0.1900178	0.1833732	0.1906172	0.1901671	0.1900178	0.181395	0.188982
35	0.1901671	0.1906172	0.1901671	0.1900178	0.1833732	0.1906172	0.1901671	0.1900178	0.181395	0.188982
40	0.1901671	0.1906172	0.1901671	0.1900178	0.1833732	0.1906172	0.1901671	0.1900178	0.181395	0.188982
45	0.1901671	0.1906172	0.1901671	0.1900178	0.1833732	0.1906172	0.1901671	0.1900178	0.181395	0.188982
50	0.1901671	0.1906172	0.1901671	0.1900178	0.1833732	0.1906172	0.1901671	0.1900178	0.181395	0.188982
Waktu Komputasi	5.3 detik									

B.4 Produk Tas

Generasi	Kombinasi cr : mr									
	0.1 : 1	0.2 : 0.9	0.3 : 0.8	0.4 : 0.7	0.5 : 0.6	0.6 : 0.5	0.7 : 0.4	0.8 : 0.3	0.9 : 0.2	1 : 0.1
5	0.2212152	0.2253278	0.2260777	0.2205126	0.2307396	0.2255769	0.2236068	0.2365609	0.2323588	0.232358
10	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.2270892	0.2270892	0.2365609	0.2365609	0.236560
15	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.2270892	0.2365609	0.2365609	0.2365609	0.236560
20	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.2270892	0.2365609	0.2365609	0.2365609	0.236560
25	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.2270892	0.2365609	0.2365609	0.2365609	0.236560
30	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.2270892	0.2365609	0.2365609	0.2365609	0.236560
35	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.2270892	0.2365609	0.2365609	0.2365609	0.236560
40	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.2270892	0.2365609	0.2365609	0.2365609	0.236560
45	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.2270892	0.2365609	0.2365609	0.2365609	0.236560
50	0.2365609	0.2365609	0.2365609	0.2365609	0.2365609	0.2270892	0.2365609	0.2365609	0.2365609	0.236560
Waktu Komputasi	3.8 detik									

B.5 Produk Jaket

Generasi	Kombinasi cr : mr									
	0.1 : 1	0.2 : 0.9	0.3 : 0.8	0.4 : 0.7	0.5 : 0.6	0.6 : 0.5	0.7 : 0.4	0.8 : 0.3	0.9 : 0.2	1 : 0.1
5	0.2093072	0.2105136	0.2048678	0.2087118	0.2103111	0.2075359	0.2071484	0.2107167	0.2083176	0.210311
10	0.2153392	0.2105136	0.2107167	0.2138356	0.214262	0.2107167	0.2153392	0.2153392	0.214262	0.210311
15	0.2153392	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.213835
20	0.2153392	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339
25	0.2153392	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339
30	0.2153392	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339
35	0.2153392	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339
40	0.2153392	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.215339

Generasi	Kombinasi cr : mr									
	0.1 : 1	0.2 : 0.9	0.3 : 0.8	0.4 : 0.7	0.5 : 0.6	0.6 : 0.5	0.7 : 0.4	0.8 : 0.3	0.9 : 0.2	1 : 0.1
45	0.2153392	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392
50	0.2153392	0.2105136	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392	0.2153392
Waktu Komputasi	5.1 detik									

B.6 Produk Sepatu

Generasi	Kombinasi cr : mr									
	0.1 : 1	0.2 : 0.9	0.3 : 0.8	0.4 : 0.7	0.5 : 0.6	0.6 : 0.5	0.7 : 0.4	0.8 : 0.3	0.9 : 0.2	1 : 0.1
5	0.3525965	0.3516474	0.3646203	0.3516474	0.3646203	0.3525965	0.3525965	0.3525965	0.3646203	0.364620
10	0.3625308	0.3516474	0.3646203	0.3574602	0.3646203	0.3646203	0.373355	0.3625308	0.3656787	0.364620
15	0.3646203	0.3574602	0.3646203	0.3574602	0.3646203	0.3646203	0.373355	0.3646203	0.3656787	0.364620
20	0.3646203	0.3574602	0.3646203	0.3574602	0.3646203	0.3646203	0.373355	0.3646203	0.3656787	0.364620
25	0.3646203	0.3574602	0.3646203	0.3574602	0.3646203	0.3646203	0.373355	0.3646203	0.3656787	0.364620
30	0.3646203	0.3574602	0.3646203	0.3574602	0.3646203	0.3646203	0.373355	0.3646203	0.3656787	0.364620
35	0.3646203	0.3574602	0.3646203	0.3574602	0.3646203	0.3646203	0.373355	0.3646203	0.3656787	0.364620
40	0.3646203	0.3574602	0.3646203	0.3574602	0.3646203	0.3646203	0.373355	0.3646203	0.3656787	0.364620
45	0.3646203	0.3574602	0.3646203	0.3574602	0.3646203	0.3646203	0.373355	0.3646203	0.3656787	0.364620
50	0.3646203	0.3574602	0.3646203	0.3574602	0.3646203	0.3646203	0.373355	0.3646203	0.3656787	0.364620
Waktu Komputasi	1.1 detik									

LAMPIRAN C DATA PENGUJIAN TINGKAT AKURASI

C.1 Produk Topi

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Januari	47	-	-	-
	Ferbuari	45	40	0.11111111	0.01234568
	Maret	43	45	-0.0465116	0.00216333
	April	34	33	0.02941176	0.00086505
	Mei	47	42	0.10638298	0.01131734
	Juni	40	40	0	0
	Juli	47	42	0.10638298	0.01131734
	Agustus	45	40	0.11111111	0.01234568
	September	45	45	0	0
	Oktober	46	45	0.02173913	0.00047259
	November	47	45	0.04255319	0.00181077
	Desember	32	40	-0.25	0.0625
2017	Januari	44	42	0.04545455	0.00206612
	Ferbuari	48	49	-0.0208333	0.00043403
	Maret	34	33	0.02941176	0.00086505
	April	38	42	-0.1052632	0.01108033
	Mei	41	40	0.02439024	0.00059488
	Juni	33	33	0	0
	Juli	39	42	-0.0769231	0.00591716
	Agustus	40	42	-0.05	0.0025
	September	36	42	-0.1666667	0.02777778
	Oktober	38	40	-0.0526316	0.00277008
	November	36	40	-0.11111111	0.01234568
	Desember	42	40	0.04761905	0.00226757
2018	Januari	-	42	-	-
Jumlah					0.18375647
RMSPE					8.94%
Akurasi					91.06%

C.2 Produk Sabuk

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Januari	57	-	-	-
	Ferbuari	52	61	-0.1730769	0.02995562
	Maret	59	65	-0.1016949	0.01034186
	April	53	61	-0.1509434	0.02278391
	Mei	68	65	0.04411765	0.00194637
	Juni	63	59	0.06349206	0.00403124
	Juli	64	59	0.078125	0.00610352
	Agustus	53	59	-0.1132075	0.01281595
	September	53	65	-0.2264151	0.05126379
	Oktober	83	65	0.21686747	0.0470315
	November	70	67	0.04285714	0.00183673
	Desember	77	77	0	0
2017	Januari	55	55	0	0
	Ferbuari	57	61	-0.0701754	0.00492459
	Maret	72	61	0.15277778	0.02334105
	April	69	77	-0.115942	0.01344255
	Mei	82	77	0.06097561	0.00371802
	Juni	62	67	-0.0806452	0.00650364
	Juli	81	80	0.01234568	0.00015242
	Agustus	80	76	0.05	0.0025
	September	74	76	-0.027027	0.00073046
	Oktober	75	77	-0.0266667	0.00071111
	November	54	55	-0.0185185	0.00034294
	Desember	65	65	0	0
2018	Januari	-	65	-	-
Jumlah					0.24447727
RMSPE					10.31%
Akurasi					89.69%

C.3 Produk Dompot

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Januari	139	-	-	-
	Februari	163	151	0.07361963	0.00541985
	Maret	153	154	-0.0065359	0.000042719
	April	141	144	-0.0212766	0.000452694
	Mei	152	151	0.00657895	4.32825E-05
	Juni	147	144	0.02040816	0.000416493
	Juli	142	146	-0.028169	0.000793493
	Agustus	148	151	-0.0202703	0.000410884
	September	151	146	0.03311258	0.001096443
	Oktober	139	138	0.00719424	5.17572E-05
	November	142	151	-0.0633803	0.00401706
	Desember	150	151	-0.0066667	4.44444E-05
2017	Januari	162	160	0.01234568	0.000152416
	Februari	137	134	0.02189781	0.000479514
	Maret	141	143	-0.0141844	0.000201197
	April	151	151	0	0
	Mei	135	138	-0.0222222	0.000493827
	Juni	134	143	-0.0671642	0.004511027
	Juli	135	143	-0.0592593	0.00351166
	Agustus	155	143	0.07741935	0.005993757
	September	146	144	0.01369863	0.000187652
	Oktober	149	146	0.02013423	0.000405387
	November	144	146	-0.0138889	0.000192901
	Desember	144	146	-0.0138889	0.000192901
2018	Januari	-	144	-	-
		Jumlah		0.02911136	
		RMSPE		3.56%	
		Akurasi		96.44%	

C.4 Produk Tas

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Januari	65	-	-	-
	Februari	73	67	0.08219178	0.00675549
	Maret	70	67	0.04285714	0.00183673
	April	57	56	0.01754386	0.00030779
	Mei	69	68	0.01449275	0.00021004
	Juni	74	73	0.01351351	0.00018262
	Juli	71	67	0.05633803	0.00317397
	Agustus	74	67	0.09459459	0.00894814
	September	64	67	-0.046875	0.00219727
	Oktober	58	63	-0.0862069	0.00743163
	November	56	56	0	0
	Desember	73	68	0.06849315	0.00469131
2017	Januari	62	67	-0.0806452	0.00650364
	Februari	62	63	-0.016129	0.00026015
	Maret	66	63	0.04545455	0.00206612
	April	61	67	-0.0983607	0.00967482
	Mei	64	68	-0.0625	0.00390625
	Juni	66	63	0.04545455	0.00206612
	Juli	71	67	0.05633803	0.00317397
	Agustus	60	67	-0.1166667	0.01361111
	September	56	56	0	0
	Oktober	61	68	-0.1147541	0.0131685
	November	73	68	0.06849315	0.00469131
	Desember	69	67	0.02898551	0.00084016
2018	Januari	-	69	-	-
		Jumlah		0.09569713	
		RMSPE		6.45%	
		Akurasi		93.55%	

C.5 Produk Jaket

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Januari	72	-	-	-
	Februari	66	77	-0.1666667	0.02777778
	Maret	75	77	-0.0266667	0.00071111
	April	73	73	0	0
	Mei	76	73	0.03947368	0.00155817
	Juni	84	86	-0.0238095	0.00056689
	Juli	81	81	0	0
	Agustus	78	79	-0.0128205	0.00016437
	September	73	79	-0.0821918	0.00675549
	Oktober	69	73	-0.057971	0.00336064
	November	82	81	0.01219512	0.00014872
	Desember	68	73	-0.0735294	0.00540657
2017	Januari	77	77	0	0
	Februari	78	79	-0.0128205	0.00016437
	Maret	82	79	0.03658537	0.00133849
	April	82	73	0.1097561	0.0120464
	Mei	68	73	-0.0735294	0.00540657
	Juni	79	77	0.02531646	0.00064092
	Juli	72	79	-0.0972222	0.00945216
	Agustus	85	77	0.09411765	0.00885813
	September	82	81	0.01219512	0.00014872
	Oktober	69	73	-0.057971	0.00336064
	November	79	81	-0.0253165	0.00064092
	Desember	84	79	0.05952381	0.00354308
2018	Januari	-	84	-	-
		Jumlah		0.09205015	
		RMSPE		6.33%	
		Akurasi		93.67%	

C.6 Produk Sepatu

Tahun	Bulan	Data Aktual (a)	Data Prediksi (b)	Error $\left(\frac{a-b}{a}\right)$	Error ²
2016	Januari	36	-	-	-
	Februari	37	41	-0.1081081	0.01168736
	Maret	39	41	-0.0512821	0.00262985
	April	37	36	0.02702703	0.00073046
	Mei	47	41	0.12765957	0.01629697
	Juni	43	40	0.06976744	0.0048675
	Juli	41	40	0.02439024	0.00059488
	Agustus	40	44	-0.1	0.01
	September	43	44	-0.0232558	0.00054083
	Oktober	42	40	0.04761905	0.00226757
	November	36	36	0	0
	Desember	41	41	0	0
2017	Januari	45	44	0.02222222	0.00049383
	Februari	36	40	-0.11111111	0.01234568
	Maret	41	41	0	0
	April	44	44	0	0
	Mei	44	44	0	0
	Juni	43	44	-0.0232558	0.00054083
	Juli	37	40	-0.0810811	0.00657414
	Agustus	37	41	-0.1081081	0.01168736
	September	38	41	-0.0789474	0.00623269
	Oktober	36	36	0	0
	November	45	41	0.08888889	0.00790123
	Desember	43	40	0.06976744	0.0048675
2018	Januari	-	43	-	-
		Jumlah		0.10025869	
		RMSPE		6.60%	
		Akurasi		93.40%	