

**PENERAPAN ALGORITMA GENETIKA DALAM
PENCARIAN NILAI PARAMETER UNTUK PERAMALAN
DATA PENJUALAN SECARA *TIME SERIES***

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana Komputer dalam bidang Ilmu Komputer

Oleh :

**DANU PATRIA P. N.
0410960014-96**



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI

PENERAPAN ALGORITMA GENETIKA DALAM PENCARIAN NILAI PARAMETER UNTUK PERAMALAN DATA PENJUALAN SECARA *TIME SERIES*

Oleh:
DANU PATRIA P. N.
0410960014-96

Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 24 Maret 2009
dan dinyatakan memenuhi syarat untuk memperoleh
gelar Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I,

Pembimbing II,

Edy Santoso, S.Si., M.Kom
NIP. 132 304 307

Drs. Achmad Ridok, M.Kom
NIP. 132 090 392

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, MSc.
NIP. 132 126 049

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Danu Patria P. N.
NIM : 0410960014-96
Jurusan : Matematika
Program Studi : Ilmu Komputer
Penulis Tugas Akhir berjudul : Penerapan Algoritma Genetika
Dalam Pencarian Nilai
Parameter Untuk Peramalan
Data Penjualan Secara *Time
Series*

Dengan ini menyatakan bahwa :

1. Isi dari tugas akhir yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Tugas Akhir ini.
2. Apabila dikemudian hari ternyata Tugas Akhir yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 24 Maret 2009

Yang menyatakan,

(Danu Patria P. N.)

NIM. 0410960014-96

UNIVERSITAS BRAWIJAYA



PENERAPAN ALGORITMA GENETIKA DALAM PENCARIAN NILAI PARAMETER UNTUK PERAMALAN DATA PENJUALAN SECARA *TIME SERIES*

ABSTRAK

Peramalan data merupakan suatu hal yang penting di dunia bisnis. Contoh data yang perlu untuk didapatkan hasil ramalannya adalah data penjualan. Salah satu metode peramalan yang ada adalah peramalan secara *time series*, yaitu metode peramalan dengan cara menganalisa data pada waktu yang lalu saja dan mengabaikan faktor-faktor lainnya.

Metode yang umum digunakan untuk menyelesaikan permasalahan peramalan *time series* ini adalah dengan cara memodelkannya secara linier, kemudian dengan memperhatikan data-data dari waktu sebelumnya, dicari koefisien-koefisien model linier yang mampu mewakili perilaku seluruh rangkaian data. Algoritma genetika bisa diterapkan untuk membantu proses pencarian koefisien-koefisien model linier tadi dengan cara merepresentasikannya ke dalam kromosom, kemudian melalui proses evolusi yang baik, bisa didapatkan sebuah koefisien model linier yang optimal. Koefisien model linier ini kemudian diterapkan untuk mendapatkan nilai ramalan data *time series* satu tahap ke depan. Nilai periode peramalan yang digunakan dalam proses peramalan merupakan nilai yang dihasilkan secara otomatis oleh sistem, dan merupakan nilai yang diharapkan memiliki tingkat akurasi terbaik.

Pengujian sistem kemudian dilakukan dengan cara membandingkan data hasil ramalan dengan data aktual. Dari 120 kali pengujian untuk semua jenis data yang ada, didapatkan rata-rata nilai kesalahan sebesar 0.1132%. Hasil pengujian tersebut menunjukkan bahwa data prediksi yang dihasilkan menggunakan metode algoritma genetika ini mampu mendekati data aktual (data sebenarnya) yang dicari dengan baik.

UNIVERSITAS BRAWIJAYA



GENETIC ALGORITHM IMPLEMENTATION IN *TIME SERIES* PARAMETER SEARCHING FOR SALES DATA FORECASTING

ABSTRACT

Data forecasting is something that very important in the business world. One kind of data that need to be forecasted is the sales data. Time series method in forecasting data is a method that will analyze historical data only and ignoring other factors. Hence, this method is appropriate to be applied in such data that cannot be determined exactly what are its influencing factors, like on this sales data.

The common method used to solve this time series forecasting problem is by models it as a linear equation, and then search those linear equation coefficients that can follows the data behavior with considering the data taken from past period. Genetic algorithm can be implemented to find the linear equation coefficients by represent it in a cromosom, and then trough a good evolution process, the optimum linear equation coefficients can be obtained. This value is then applied to the data to get a forecasting result for one period ahead. The forecasting period value that being used in this forecasting process is automatically generated by the system, and this value is expected to shows the best accuracy.

This system is then evaluated by comparing the prediction result with the actual data. From 120 test with all kind of data that provided, the average fault percentage is 0.1132 %. This test result is showed that prediction data created by using genetic algorithm method can approach the actual data exemplary.

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Alhamdulillah Robbil alamin, puji syukur penulis panjatkan kepada Allah SWT atas segala limpahan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini, yang berjudul “Penerapan Algoritma Genetika Dalam Peramalan Data Penjualan Secara *Time Series*”. Sebagai salah satu syarat untuk memperoleh gelar Sarjana dalam bidang Ilmu Komputer

Banyak pihak yang berperan atas terselesaikannya penulisan Skripsi ini. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Edy Santoso, S.Si., M.Kom, selaku pembimbing utama dalam penulisan skripsi.
2. Drs. Achmad Ridok, M.Kom, selaku pembimbing pendamping dalam penulisan skripsi.
3. Wayan Firdaus Mahmudy S.Si, MT selaku Ketua Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.
4. Drs. Marji, MT, selaku pembimbing akademik.
5. Dr. Agus Suryanto, Msc, selaku Ketua Jurusan Matematika, FMIPA Universitas Brawijaya.
6. Segenap Bapak dan Ibu Dosen yang telah mendidik penulis selama menempuh pendidikan di Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.
7. Segenap staf dan karyawan di Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.
8. Orang tua dan keluarga besar penulis atas dukungan materi dan doa restunya.
9. Teman-teman ”Rabu Pagi” atas segala bantuan, masukan dan dukungan semangatnya.
10. Rekan-rekan di Program Studi Ilmu Komputer FMIPA Universitas Brawijaya yang telah banyak memberikan bantuan, kritik dan saran demi kelancaran pelaksanaan penyusunan tugas akhir ini.
11. Semua anggota dan intern di AIESEC Indonesia LC Unibraw. Atas segala dukungan dan inspirasinya.

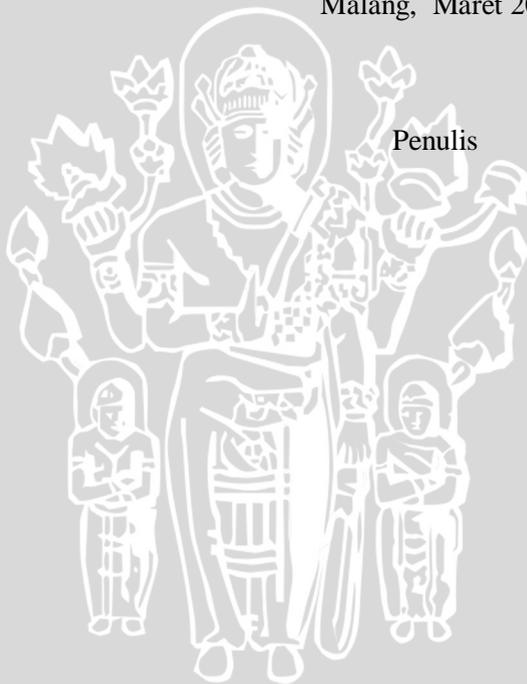
12. Dan semua pihak yang telah membantu dan mendukung dalam penulisan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Akhinya, penulis sadari bahwa masih banyak kekurangan dalam penyusunan skripsi ini, oleh karena itu penulis sangat menghargai saran dan kritik yang membangun demi perbaikan penulisan dan mutu isi skripsi ini untuk kelanjutan penelitian serupa di masa mendatang.

Semoga skripsi ini dapat bermanfaat.

Malang, Maret 2009

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	5
2.1. Konsep Peramalan Data <i>Time Series</i>	5
2.2. Algoritma Genetika	6
2.2.1. Representasi kromosom	7
2.2.2. Fungsi <i>Fitness</i>	8
2.2.3. Seleksi Orangtua	9
2.2.4. Persilangan	10
2.2.5. Mutasi	12
2.2.6. Seleksi Populasi	14
2.3. Peramalan Data <i>Time Series</i> Dengan Algoritma Genetika	14
BAB III METODOLOGI DAN PERANCANGAN	17
3.1. Identifikasi Sistem	17
3.2.	17
Sumber Data	17
3.3. Tahapan Algoritma Genetika	17

3.3.1.	Representasi Kromosom.....	19
3.3.2.	Fungsi <i>Fitness</i>	20
3.3.3.	Seleksi Orang tua	21
3.3.4.	<i>Crossover</i>	22
3.3.5.	Mutasi.....	23
3.3.6.	Seleksi Populasi	24
3.4.	
	Struktur Data.....	25
3.5.	
	Perancangan Tampilan Antar Muka.....	27
3.6.	
	Perancangan Uji Coba	28
3.6.1.	Pengujian Variasi Data Masukan.....	28
3.6.2	Pengujian Variasi Parameter Genetik.....	29
3.7.	Contoh Perhitungan	30
3.7.1	Membangkitkan Populasi Awal	31
3.7.2	Penghitungan Nilai <i>Fitness</i>	31
3.7.3	Seleksi Orang tua	32
3.7.4.	<i>Crossover</i>	33
3.7.5	Mutasi.....	34
3.7.6	Seleksi Populasi	35
3.7.7	Peramalan Data.....	36
 BAB IV IMPLEMENTASI DAN PEMBAHASAN		37
4.1	Lingkungan Implementasi	37
4.1.1	Lingkungan perangkat keras	37
4.1.2	Lingkungan perangkat lunak	37
4.2	Implementasi Program	37
4.2.1.	Inisialisasi Kromosom.....	38
4.2.2.	Perhitungan Nilai <i>Fitness</i>	38
4.2.3.	Mengurutkan Kromosom Sesuai Nilai <i>Fitness</i>	39
4.2.4.	<i>Crossover</i>	40
4.2.5.	Proses Mutasi.....	43
4.2.6.	Perulangan Proses <i>Crossover</i>	43
4.2.7.	Seleksi Proses Mutasi.....	44
4.2.8	Perulangan Proses Genetik	45
4.2.9.	Pengambilan Nilai Prediksi	46
4.2.10.	Pengambilan Nilai Terbaik pada Setiap Nilai k	47
4.3.	Implementasi Antar Muka (Interface).....	48

4.4. Uji Coba Sistem	50
4.4.1 Pengujian Dengan Nilai k yang Berbeda	50
4.4.2 Pengujian Berdasarkan Input Data	52
4.4.3 Pengujian Berdasarkan Input Jumlah Generasi.....	53
4.4.4 Pengujian Berdasarkan Input parameter Genetik.....	56
4.4.5 Pengujian Akurasi Peramalan Secara Simultan	57
BAB V KESIMPULAN DAN SARAN.....	65
5.1 Kesimpulan.....	65
5.2 Saran.....	65



UNIVERSITAS BRAWIJAYA



DAFTAR GAMBAR

Gambar 2.1 Skema umum algoritma genetika	6
Gambar 2.2 Seleksi proporsional menggunakan <i>roulette wheel</i>	9
Gambar 2.3 Penyilangan satu titik.....	10
Gambar 2.4 Penyilangan banyak titik.....	11
Gambar 2.5 Penyilangan seragam.....	11
Gambar 2.6 Mutasi kromosom biner	12
Gambar 2.7 Mutasi dengan membalik nilai.....	13
Gambar 2.8 Mutasi pemilihan nilai acak.....	13
Gambar 3.1 Diagram alir proses algoritma genetika	18
Gambar 3.2 Diagram alir proses pencarian nilai terbaik.....	19
Gambar 3.3 Representasi kromosom	20
Gambar 3.4 Rancangan tampilan antar muka	27
Gambar 4.1 Inisialisasi kromosom awal	38
Gambar 4.2 Hitung <i>Fitness</i>	39
Gambar 4.3 Mengurutkan kromosom sesuai nilai <i>fitness</i>	40
Gambar 4.4 Menghitung daerah <i>roulette wheel</i>	41
Gambar 4.5 Memilih individu yang melakukan persilangan.....	41
Gambar 4.6 Proses persilangan.....	42
Gambar 4.7 Memasukkan kromosom anak ke populasi.....	42
Gambar 4.8 Proses mutasi	43
Gambar 4.9 Perulangan proses persilangan.....	44
Gambar 4.10 Seleksi proses mutasi	44
Gambar 4.11 Perulangan proses genetik.....	46
Gambar 4.12 Menghitung nilai prediksi akhir.....	47
Gambar 4.13 Mencari Nilai <i>k</i> Terbaik	48
Gambar 4.14 Tampilan utama pada <i>browser</i>	49
Gambar 4.15 Tampilan setelah dijalankan	49
Gambar 4.16 Nilai fitness tiap generasi sampai generasi ke-200	52
Gambar 4.17 Nilai fitness tiap generasi sampai generasi ke-300	53
Gambar 4.18 Nilai fitness tiap generasi sampai generasi ke-400	53
Gambar 4.19 Nilai fitness tiap generasi sampai generasi ke-500	53
Gambar 4.20 Nilai fitness tiap generasi sampai generasi ke-600	54
Gambar 4.21 Perbandingan data aktual dengan data hasil prediksi pada data penjualan komoditas manufaktur.....	57
Gambar 4.22 Perbandingan data aktual dengan data hasil prediksi pada data penjualan komoditas tekstil.....	59

Gambar 4.23 Perbandingan data aktual dengan data hasil prediksi pada data penjualan komoditas elektronik 61

UNIVERSITAS BRAWIJAYA



DAFTAR TABEL

Tabel 2.1 Contoh data <i>time series</i>	5
Tabel 2.2 Contoh representasi <i>real</i>	8
Tabel 2.3 Contoh representasi desimal	8
Tabel 2.4 Contoh representasi biner	8
Tabel 3.1 Data Yang akan diproses	25
Tabel 3.2 Contoh penyimpanan data pada matriks	25
Tabel 3.3 Contoh penyimpanan data populasi pada matriks	26
Tabel 3.4 Tabel pengujian variasi data masukan	28
Tabel 3.5 Tabel pengujian variasi jumlah generasi.....	29
Tabel 3.6 Tabel pengujian variasi parameter genetik	29
Tabel 3.7 Contoh data yang akan diproses	30
Tabel 3.8 Populasi awal yang terbentuk.....	31
Tabel 3.9 Penghitungan <i>fitness</i> individu ke-1.....	31
Tabel 3.10 Penghitungan <i>fitness</i> individu ke-2.....	31
Tabel 3.11 Penghitungan <i>fitness</i> individu ke-3.....	32
Tabel 3.12 Penempatan individu dalam <i>roulette wheel</i>	32
Tabel 3.13 Penghitungan <i>fitness</i> anak 1	33
Tabel 3.14 Penghitungan <i>fitness</i> anak 2	34
Tabel 3.15 Penghitungan <i>fitness</i> individu 1 setelah dimutasi.....	35
Tabel 3.16 Populasi setelah proses mutasi	35
Tabel 3.17 Mengurutkan seluruh individu berdasarkan <i>fitness</i>	36
Tabel 3.18 Populasi baru yang terbentuk	36
Tabel 4.1 Perbandingan hasil dengan menggunakan nilai k yang berbeda pada data penjualan komoditas manufaktur	51
Tabel 4.2 Perbandingan hasil dengan menggunakan nilai k yang berbeda pada data penjualan komoditas elektronik	51
Tabel 4.3 Nilai aktual (nilai sebenarnya) yang dicari	52
Tabel 4.4 Persentasi kesalahan peramalan data <i>time series</i> penjualan komoditas manufaktur.....	52
Tabel 4.5 Persentasi kesalahan peramalan data <i>time series</i> penjualan komoditas tekstil.....	53
Tabel 4.6 Persentasi kesalahan peramalan data <i>time series</i> penjualan komoditas elektronik.....	53
Tabel 4.7 Nilai <i>fitness</i> terbaik yang dihasilkan dengan input jumlah generasi yang berbeda	54
Tabel 4.8 Persentasi kesalahan hasil peramalan dengan	

	menggunakan input parameter genetik yang berbeda.....	57
Tabel 4.9	Hasil pengujian 40data secara berurutan pada data penjualan komoditas manufaktur	58
Tabel 4.10	Hasil pengujian 40data secara berurutan pada data penjualan komoditas tekstil.....	60
Tabel 4.11	Hasil pengujian 40data secara berurutan pada data penjualan komoditas elektronik.....	62
Tabel 4.12	Nilai persentasi kesalahan untuk tiap data ujicoba.....	64

UNIVERSITAS BRAWIJAYA



DAFTAR LAMPIRAN

Lampiran 1 Data yang digunakan dalam pengujian..... 67

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



BAB I

PENDAHULUAN

1.1. Latar Belakang

Peramalan dan prediksi data pada dunia bisnis merupakan suatu hal yang penting. Contoh data yang perlu untuk didapatkan hasil ramalannya adalah data penjualan. Hasil peramalan data penjualan yang baik sangat diperlukan sebagai pendukung di dalam penyusunan rencana, strategi dan pengambilan keputusan pemasaran.

Salah satu metode peramalan yang ada adalah peramalan secara *time series* (*time-series forecasting*). Peramalan data *time series* adalah peramalan data dengan menggunakan data pada waktu yang lalu yang berkorelasi untuk mendapatkan data pada waktu yang akan datang. Pada metode ini, yang diperhatikan untuk mendapatkan hasil ramalan adalah data-data pada waktu yang lalu saja dan mengabaikan faktor-faktor lain. Pertimbangan yang digunakan untuk mencari hasil ramalan adalah dengan melihat pada hubungan antar data atau pola data-nya. Sehingga peramalan dengan metode ini ini sesuai untuk diterapkan pada data yang tidak bisa ditentukan secara tepat faktor-faktor yang mempengaruhinya, misalnya seperti pada data penjualan, data jumlah produksi, data jumlah konsumsi dan sebagainya.

Teknik peramalan standar statistik merupakan metode yang banyak digunakan dalam peramalan data *time series*, seperti dengan menggunakan metode *moving average*, regresi dan ARIMA (*Autoregression Integrated Moving Average*). Metode tersebut digunakan untuk mengenali pola data *time series* dan kemudian menerapkannya untuk mendapatkan hasil ramalan. Namun dalam beberapa kasus, pola data *time series* ini sulit ditemukan karena merupakan data yang acak dan tidak teratur. Sehingga dewasa ini telah banyak dikembangkan metode peramalan menggunakan metode-metode *heuristik*.

Algoritma genetika merupakan salah satu metode *heuristik* yang ada dan sering digunakan untuk menyelesaikan masalah-masalah yang kompleks dengan hasil yang relatif akurat dan dengan waktu yang cepat. Algoritma genetika adalah algoritma yang menirukan proses seleksi alamiah (proses evolusi) dalam membentuk individu-individu baru dengan nilai kesesuaian dengan lingkungan (*fitness*) yang lebih

baik. Proses evolusi dalam algoritma genetika dimulai dari proses reproduksi dan mutasi kemudian dilanjutkan dengan proses seleksi. Dalam proses seleksi ini, hanya individu yang memiliki nilai kesesuaian dengan lingkungan yang baik saja yang bisa bertahan, sedangkan yang memiliki nilai kesesuaian yang kurang akan tersingkir dari populasi. Sehingga semakin lama nilai kesesuaian setiap individu dalam populasi akan meningkat.

Dalam peramalan *time series* dengan algoritma genetika, pada intinya adalah dengan cara mengenali perilaku data dengan mencari besar pengaruh data-data dari waktu yang sebelumnya, kemudian menerapkannya untuk memprediksi nilai satu tahap kedepan.

Data hasil proses peramalan yang dihasilkan dengan metode ini kemudian akan di uji coba dengan cara membandingkannya dengan data yang sebenarnya (data aktual), sehingga nilai keakuratan data yang dihasilkan bisa dilihat dan dianalisa.

Pada akhirnya, diharapkan dengan penerapan algoritma genetika dalam peramalan data *time series*, mampu mempelajari perilaku data dengan baik, dan cepat sehingga bisa didapatkan hasil peramalan yang akurat.

Dari latar belakang yang telah dipaparkan tersebut, maka judul yang diambil pada skripsi ini yaitu ***"Penerapan Algoritma Genetika Dalam Pencarian Nilai Parameter Untuk Peramalan Data Penjualan Secara Time Series"***

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan beberapa masalah, antara lain:

1. Bagaimana menerapkan algoritma genetika dalam peramalan data secara *time series*.
2. Bagaimana hasil uji coba penerapan algoritma genetika dalam peramalan data secara *time series*.

1.3. Batasan Masalah

Untuk menghindari meluasnya permasalahan, maka dibatasi pada hal-hal berikut:

1. Peramalan yang dilakukan adalah peramalan untuk satu tahap kedepan saja.

2. Data *time series* yang digunakan dalam uji coba adalah data nyata jumlah penjualan perbulan barang-barang produksi yang didapat dari situs *statsonline*.

1.4. Tujuan Penelitian

Tujuan penulisan tugas akhir ini adalah sebagai berikut:

1. Mengetahui penerapan algoritma genetika dalam peramalan data secara *time series*.
2. Mengetahui hasil uji coba penerapan algoritma genetika dalam peramalan data secara *time series*.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini yaitu untuk menyediakan perangkat lunak peramalan data secara *time series* menggunakan algoritma genetika.

1.6. Metodologi Penelitian

Metodologi yang digunakan dalam penyusunan skripsi ini adalah sebagai berikut:

1. Studi literatur
Mencari dan mempelajari beberapa literatur dan artikel ilmiah yang mengandung teori-teori tentang karakteristik data *time series*, dan konsep algoritma genetika.
2. Perancangan sistem
Membuat rancangan sistem untuk peramalan data time-series menggunakan algoritma genetika.
3. Implementasi sistem
Mengimplementasikan hasil rancangan sistem dengan cara membangun perangkat lunak.
4. Uji coba dan analisa hasil
Menguji coba dan menganalisa hasil implementasi untuk menarik kesimpulan penelitian.

1.7. Sistematika Penulisan

Untuk memberikan gambaran tentang tugas akhir, berikut disajikan garis besar pembahasan dari keseluruhan isi laporan tugas akhir untuk setiap bab.

BAB I : PENDAHULUAN

Bab ini berisi latar belakang penulisan, permasalahan yang ada, batasan masalah, tujuan dan manfaat serta metodologi dan sistematika penulisan tugas akhir.

BAB II : TINJAUAN PUSTAKA

Bab ini berisi penjelasan singkat tentang teori-teori tentang peramalan data *time series*, pergerakan data, dan konsep algoritma genetika.

BAB III : METODOLOGI DAN PERANCANGAN

Bab ini berisi perancangan penerapan metode yang digunakan untuk menyelesaikan masalah peramalan data *time series* menggunakan algoritma genetika

BAB IV : HASIL DAN PEMBAHASAN

Bab ini berisi tentang implementasi sistem, pengujian dan analisa hasil pengujian yang didapatkan.

BAB V : PENUTUP

Bab ini berisi kesimpulan penelitian dan saran-saran untuk pengembangan lebih lanjut.



BAB II TINJAUAN PUSTAKA

2.1 Konsep Peramalan Data *Time Series*

Yang dimaksud dengan data *time series* adalah rangkaian data yang didapatkan dan dicatat secara berurutan dalam selang waktu tertentu yang tetap. Menurut Simangunsong (Simangunsong, 2005), data-data ini merupakan nilai yang saling berkaitan dan tergantung.

Data *time series* misalnya data harga penutupan saham harian dalam satu tahun, data penjualan mobil perbulan dalam dua tahun, data pertumbuhan ekonomi pertahun dalam satu dekade dan sebagainya. Contoh data *time series* ditunjukkan pada tabel 2.1.

Tabel 2.1 Contoh data *time series*

Tanggal	Jumlah
3 April 2007	15.655
4 April 2007	16.754
5 April 2007	16.325
6 April 2007	15.998
...	...
...	...
...	...
20 Mei 2007	16.879
21 Mei 2007	17.141

Cara penulisan data *time series* yang umum digunakan yaitu: Z_{t_1} , Z_{t_2} , Z_{t_3} , ..., Z_{t_n} . Di mana $t_1, t_2, t_3, \dots, t_n$ adalah satuan waktu (hari, bulan, triwulan, tahun).

Dengan menggunakan beberapa metode peramalan, dari data *time series* yang telah diketahui, dapat dilakukan pendekatan untuk mendapatkan nilai data satu tahap atau beberapa tahap ke depan. Menurut Fariza (Fariza,_) , peramalan *time series* adalah model peramalan dengan menggunakan data-data pada waktu yang lalu yang *dependent* atau berkorelasi. Pola data *time series* menggambarkan semua kejadian dan mengabaikan faktor-faktor lain, karena yang dilihat bukan hal-hal yang mempengaruhi fluktuasi data, sehingga

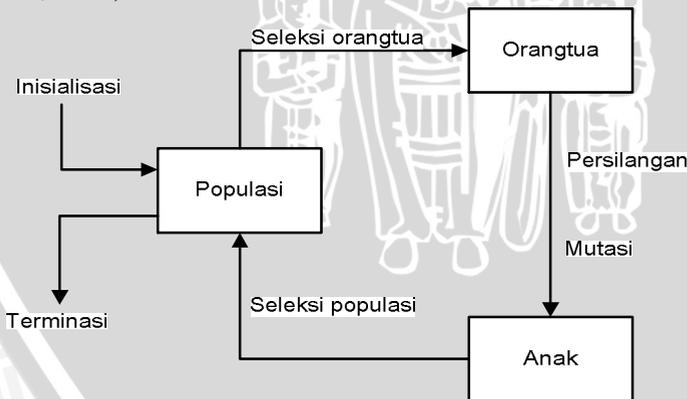
pertimbangan yang digunakan untuk meramal dalam model *time series* ini adalah hubungan antar data.

2.2 Algoritma Genetika

Algoritma genetika merupakan algoritma yang menirukan proses seleksi alamiah (proses evolusi) dalam membentuk individu baru dengan nilai kesesuaian dengan lingkungan (*fitness*) yang lebih baik. Pembentukan individu baru tersebut dilakukan dengan cara reproduksi dan mutasi. Kemudian, dalam proses seleksi alamiah ini, hanya individu yang memiliki nilai *fitness* yang baik saja yang bisa bertahan, sedangkan yang memiliki nilai *fitness* yang kurang akan tersingkir dari populasi. Sehingga semakin lama nilai *fitness* setiap individu dalam populasi akan meningkat.

Dalam algoritma genetika, sebuah variabel solusi direpresentasikan sebagai struktur kromosom pada satu individu dalam populasi. Selanjutnya, melalui proses evolusi tadi, akan didapatkan sebuah individu yang memiliki nilai kesesuaian dengan lingkungan (*fitness*) yang tinggi. Individu inilah yang memiliki kromosom yang membawa variabel solusi yang diharapkan optimal. Secara sederhana, proses algoritma genetika diilustrasikan oleh Gambar 2.1.

Evolusi yang terjadi pada individu-individu dalam populasi digambarkan oleh siklus seleksi orangtua, *crossover* dan mutasi serta elitisme. Evolusi berhenti jika kondisi terminasi sudah terpenuhi (Suyanto, 2008).



ambar 2.1 Skema umum algoritma genetika
Sumber : Suyanto, 2008

Terdapat 6 komponen utama dalam algoritma genetika. Yaitu representasi kromosom, fungsi *fitness*, seleksi orang tua, pindah silang (*crossover*), mutasi dan elitisme.

Lebih spesifik lagi, secara garis besar algoritma genetika dapat dijelaskan dengan algoritma sebagai berikut :

1. [**Inisialisasi**] Membangun populasi secara random sebanyak n kromosom (sesuai dengan masalahnya)
2. [**Fitness**] Evaluasi setiap *fitness* $f(x)$ dari setiap kromosom x pada populasi
3. [**Populasi baru**] Membuat populasi baru dengan mengulang langkah-langkah berikut sampai populasi baru lengkap
 1. [**Seleksi**] Pilih dua kromosom induk dari populasi berdasarkan *fitness*nya (semakin besar *fitness*nya semakin besar kemungkinannya untuk terpilih)
 2. [**Perkawinan silang**] Sesuai dengan besarnya kemungkinan perkawinan silang, induk terpilih disilangkan untuk membentuk anak. Jika tidak ada perkawinan silang, maka anak merupakan salinan dari induknya.
 3. [**Mutasi**] Sesuai dengan besarnya kemungkinan mutasi, anak dimutasi pada setiap lokus (posisi pada kromosom)
 4. [**Penerimaan**] tempatkan anak baru pada populasi baru
4. [**Ganti**] Gunakan populasi yang baru dibentuk untuk proses algoritma selanjutnya.
5. [**Tes**] jika kondisi akhir terpenuhi, berhenti, dan hasilnya adalah solusi terbaik dari populasi saat itu.
6. [**Ulangi**] Ke nomer 2. (Obitko, 1998)

2.2.1 Representasi Kromosom

Langkah awal dalam algoritma genetika adalah dengan merepresentasikan variabel-variabel solusi yang dicari sebagai susunan kromosom. Dalam hal ini satu kromosom harus merepresentasikan satu solusi.

Berbeda dengan teknik pencarian konvensional, algoritma genetik berangkat dari himpunan solusi yang dihasilkan secara acak. Himpunan ini disebut populasi. Sedangkan setiap individu dalam populasi disebut kromosom yang merupakan representasi dari solusi (Gen dan Cheng, 1997).

Terdapat berbagai macam skema untuk merepresentasikan suatu variabel solusi ke dalam bentuk kromosom. Pemilihan skema yang

akan digunakan tergantung kepada masalah yang dihadapi. Beberapa skema representasi kromosom yang umum digunakan antara lain:

- Representasi bilangan *real*. Pada skema ini, nilai gen merupakan bilangan *real* yang berada dalam interval $[0,R]$, dimana R adalah bilangan real positif dan biasanya $R = 1$. Contoh representasi *real* ditunjukkan pada tabel 2.2.

Tabel 2.2 Contoh representasi *real*

G1	G2	G3	G4	G5
0,9876	0,4672	0,0031	0,2365	0,5531

- Representasi bilangan diskrit desimal. Setiap gen bisa bernilai salah satu bilangan bulat dalam interval $[0,R]$. Contoh representasi bilangan desimal ditunjukkan pada tabel 2.3.

Tabel 2.3 Contoh representasi desimal

G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
3	1	7	9	5	9	2	8	4	1

- Representasi bilangan biner. Setiap gen hanya bisa direpresentasikan oleh bilangan biner 0 atau 1. Contoh representasi biner ditunjukkan pada tabel 2.4.

Tabel 2.4 Contoh representasi biner

G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
0	1	0	0	1	1	0	1	0	1

2.2.2 Fungsi *Fitness*

Fungsi *fitness* digunakan untuk menentukan nilai kualitas sebuah rangkaian kromosom tertentu yang dibawa setiap individu. Menurut Suyanto (Suyanto, 2005), di dalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup. Sedangkan yang bernilai *fitness* rendah akan mati.

Pada algoritma genetika untuk masalah optimasi fungsi, maka fungsi *fitness*-nya bisa ditemukan dengan mudah, yaitu fungsi itu sendiri. Namun dalam beberapa kasus lain yang aplikatif, seperti penjadwalan, peramalan, pembelajaran dan sebagainya, diperlukan formula fungsi *fitness* yang lebih kompleks untuk bisa menyesuaikan

dengan beragam masalah tersebut. Seperti dengan menggunakan fungsi objektif dan batasan-batasan.

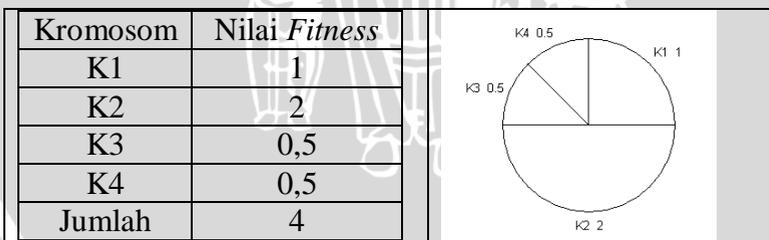
2.2.3 Seleksi Orangtua

Seleksi orang tua dilakukan untuk menentukan individu mana saja yang akan dipilih untuk melakukan rekombinasi. Penentuan ini didasarkan pada nilai *fitness* yang dimiliki individu-individu tersebut.

Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap-tahap seleksi berikutnya (Kusumadewi, 2005).

Metode yang umum digunakan dalam proses seleksi yaitu metode seleksi *fitness* proposional. Pada metode ini, pemilihan individu yang akan menjadi orang tua dilakukan secara proporsional sesuai dengan nilai *fitness*-nya. Nilai *fitness* digunakan untuk menentukan besarnya nilai probabilitas (kemungkinan) untuk terpilih menjadi orang tua. Semakin besar nilai *fitness* individu, maka kemungkinannya untuk terpilih juga semakin besar.

Metode seleksi *fitness* proposional ini dapat diilustrasikan sebagai permainan roda *roulette*, dimana individu yang memiliki nilai *fitness* lebih besar menempati potongan lingkaran yang lebih besar, sehingga kemungkinan untuk terpilih juga menjadi lebih besar. Gambaran seleksi proporsional menggunakan *roulette wheel* diilustrasikan pada Gambar 2.2.



Gambar 2.2 Seleksi proporsional menggunakan *roulette wheel*

Pada Gambar 2.2 dapat dilihat bahwa, kromosom K2 dengan nilai *fitness* paling besar, menempati potongan sebesar setengah lingkaran.

Dengan demikian K2 memiliki peluang sebesar 0,5 (2 dibagi 4) untuk terpilih sebagai orang tua.

2.2.4 Crossover

Salah satu komponen paling penting dalam algoritma genetika adalah *crossover* atau pindah silang. Sebuah kromosom yang mengarah pada solusi yang bagus bisa diperoleh dari proses memindah-silangkan dua buah kromosom (Suyanto, 2005).

Beberapa metode *crossover* yang bisa digunakan antara lain adalah penyilangan satu titik (*single-point crossover*), penyilangan banyak titik (*multi-point crossover*), penyilangan seragam (*uniform crossover*) dan *arithmetic crossover*.

- Penyilangan satu titik (*single-point crossover*).

Pada penyilangan satu titik, kromosom anak dihasilkan dari kombinasi 2 potongan kromosom dari 2 kromosom induknya. Posisi pemotongan kromosom induk bisa didapatkan dari nilai random atau dari nilai yang telah ditentukan sebelumnya. Ilustrasi penyilangan satu titik dapat dilihat pada Gambar 2.3.

Misalkan dari 2 kromosom induk dengan panjang 10:

Induk 1: 0 1 1 1 0 0 1 0 1 1

Induk 2: 1 1 0 1 0 0 0 0 1 1

Posisi penyilangan yang terpilih: 5

Setelah penyilangan, diperoleh kromosom baru:

Anak 1: 0 1 1 1 0 | 0 0 0 1 1

Anak 2: 1 1 0 1 0 | 0 1 0 1 1

Gambar 2.3 Penyilangan satu titik

- Penyilangan banyak titik (*multi-point crossover*).

Pada penyilangan banyak titik, kromosom anak dihasilkan dari n buah potongan kromosom dari 2 induknya. Posisi-posisi pemotongan kromosom induk bisa didapatkan dari nilai random atau nilai yang telah ditentukan sebelumnya. Ilustrasi penyilangan banyak titik dapat dilihat pada Gambar 2.4.

Misalkan dari 2 kromosom induk dengan panjang 12:

Induk 1: 0 1 1 0 1 1 0 0 1 0 1 0

Induk 2: 1 0 1 1 0 0 1 0 1 0 0 1

Posisi penyilangan yang terpilih: 3, 6, 9

Setelah penyilangan, diperoleh kromosom baru:

Anak 1: 0 1 1 | 1 0 0 | 0 0 1 | 0 0 1

Anak 2: 1 0 1 | 0 1 1 | 1 0 1 | 0 1 0

Gambar 2.4 Penyilangan banyak titik

- Penyilangan seragam (*uniform crossover*).
Pada penyilangan seragam, gen pada kromosom induk yang akan diwariskan ke kromosom anak ditentukan secara random dengan probabilitas yang sama. Sebuah *mask* penyilangan dibuat secara random sepanjang panjang kromosom untuk menunjukkan gen milik induk mana yang akan diwariskan kepada anak. Ilustrasi penyilangan seragam dapat dilihat pada Gambar 2.5.

Misalkan dari 2 kromosom induk dengan panjang 10:

Induk 1: 0 1 1 0 1 1 0 0 1 0

Induk 2: 1 0 1 1 0 0 1 0 1 0

Mask penyilangan: 0 1 0 1 0 0 1 1 0 1

Anak 1: 1 0 1 0 0 0 0 0 1 0

Anak 2: 0 0 1 1 1 1 1 0 1 0

Gambar 2.5 Penyilangan seragam

- Penyilangan aritmatika (*Arithmetic crossover*).
Pada representasi kromosom dengan bilangan *real*, tiga model *crossover* yang sebelumnya kurang sesuai untuk diterapkan, karena akan menyebabkan gen-gen pada kromosom dalam populasi kurang bervariasi. Untuk menghindari kondisi tersebut, maka digunakan *arithmetic crossover*. Dengan metode ini, kromosom anak dibangun dari nilai "antara" dari nilai kedua orangtuanya, sehingga bisa menghasilkan nilai yang sangat bervariasi.

Misalkan kedua kromosom orangtua dinyatakan sebagai x_1 dan x_2 , dan gen yang dimutasi pada posisi ke k . Maka kedua anak dihasilkan berdasarkan persamaan (2.1) dan persamaan (2.2).

$$x_1'(k) = r \cdot x_1(k) + (1-r) \cdot x_2(k) \quad (2.1)$$

$$x_2'(k) = r \cdot x_2(k) + (1-r) \cdot x_1(k) \quad (2.2)$$

Nilai r merupakan nilai diantara 0 – 1 yang didapat dari nilai acak atau dari nilai yang telah ditentukan sebelumnya.

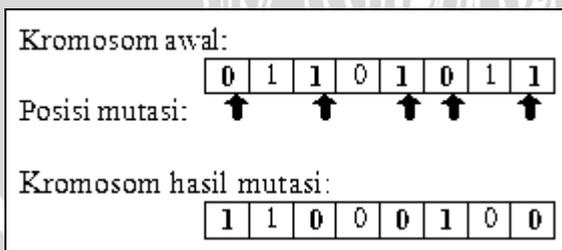
2.2.5 Mutasi

Langkah selanjutnya setelah dilakukan *crossover* yaitu proses mutasi. Tujuan dilakukan proses mutasi adalah untuk menambah keragaman variasi gen dan kromosom di dalam populasi. Tetapi karena sifatnya yang acak, maka individu yang dihasilkan oleh proses mutasi ini tidak dapat dipastikan apakah hasilnya akan mendekati atau menjauhi solusi yang dicari.

Proses mutasi adalah proses untuk mengubah sebuah gen atau lebih di dalam kromosom. Proses mutasi ini dilakukan terhadap kromosom anak yang dihasilkan, bila memenuhi syarat probabilitas mutasi yang telah ditentukan.

Terdapat beberapa metode yang umum digunakan dalam proses mutasi, misalnya:

- Mutasi untuk representasi biner.
Mutasi pada kromosom dengan representasi biner secara sederhana bisa dilakukan dengan cara membalik nilai binernya. Mutasi untuk representasi biner diilustrasikan pada Gambar 2.6.

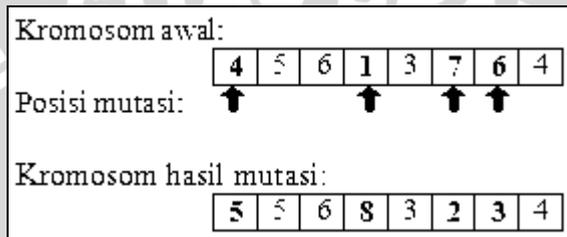


Gambar 2.6 Mutasi kromosom biner

- Mutasi untuk representasi integer.

1. Mutasi dengan membalik nilai integer

Cara ini merupakan perluasan dari mutasi untuk representasi biner. Nilai integer pada gen dibalik diganti dengan nilai kebalikannya. Sebagai contoh jika nilai gen berada pada interval [1, 8], maka gen bernilai 1 dimutasi menjadi 8, gen bernilai 2 menjadi 7 dan seterusnya. Mutasi untuk representasi integer diilustrasikan pada Gambar 2.7.



Gambar 2.7 Mutasi dengan membalik nilai

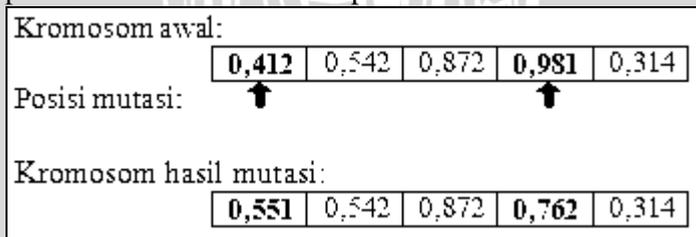
2. Mutasi dengan pemilihan nilai secara acak

Nilai gen yang terpilih untuk dimutasi diganti dengan nilai baru yang dihasilkan secara acak, di dalam interval nilai-nilai yang dibolehkan.

- Mutasi untuk representasi *real*.

1. Mutasi dengan pemilihan nilai secara acak

Metode yang digunakan dalam mutasi ini, sama dengan metode mutasi dengan pemilihan secara acak pada representasi integer. Tetapi dengan nilai gen bertipe real. Mutasi untuk representasi *real* diilustrasikan pada Gambar 2.8.



Gambar 2.8 Mutasi pemilihan nilai acak

2. Mutasi dengan penambahan nilai
Pada metode ini, gen yang dimutasi ditambahkan dengan nilai real positif atau negatif yang nilainya kecil, yang dihasilkan secara acak.

2.2.6 Seleksi Populasi

Menurut Suyanto, di dalam algoritma genetika, suatu populasi dirancang untuk memiliki jumlah individu yang selalu tetap pada setiap generasi (Suyanto, 2008).

Untuk menjaga agar jumlah individu yang ada dalam populasi selalu tetap, maka individu baru yang telah tercipta dari proses *crossover* dan mutasi tidak serta merta masuk ke menjadi anggota populasi, tetapi harus didahului dengan proses seleksi populasi. Individu baru akan masuk menjadi anggota populasi hanya bila nilai *fitness*-nya lebih baik daripada nilai *fitness* satu atau beberapa anggota populasi yang lain. Individu baru tersebut masuk ke menjadi anggota populasi menggantikan individu lain yang nilai *fitness*-nya paling rendah.

Proses ini akan menjamin bahwa individu yang membawa kromosom dengan nilai *fitness* yang terbaik tidak hilang, sementara individu yang memiliki nilai *fitness* rendah akan diganti dengan individu baru yang lebih baik.

2.3 Peramalan Data *Time-Series* Dengan Algoritma Genetika

Metode peramalan *time series* dengan algoritma genetika, pada intinya adalah dengan cara mengenali pola data dari waktu yang sebelumnya, kemudian menerapkannya untuk memprediksi nilai satu tahap kedepan.

Menurut Suyanto (Suyanto, 2008), untuk memprediksi penjualan pada hari H , hanya digunakan hasil-hasil penjualan pada hari sebelumnya, $H-1$, $H-2$, dan seterusnya. Dengan demikian, masalah ini dapat dimodelkan secara linier sebagai berikut:

$$z = a_0 + a_1y_1 + a_2y_2 + \dots + a_ky_k \quad (2.3)$$

Dimana Z adalah nilai prediksi pada waktu ke H yang dicari dan y_1 sampai y_k adalah masukan yang berupa hasil-hasil penjualan pada

hari-hari sebelumnya, $H-1$, $H-2$, ..., $H-k$. Sedangkan a_0 sampai a_k adalah variabel bernilai real.

Untuk mendapat nilai Z yang dicari, maka nilai variabel a_0 sampai a_k harus ditemukan terlebih dahulu. Pada penelitian ini, nilai koefisien pada model linear yang digunakan adalah bilangan real antara 0 dan 1.

Algoritma genetika digunakan untuk menemukan nilai-nilai variabel tersebut. Melalui proses evolusi yang baik, diharapkan bisa didapatkan nilai variabel a_0 sampai a_k yang optimal, sehingga nilai Z yang dihasilkan akan semakin mendekati nilai sebenarnya.



UNIVERSITAS BRAWIJAYA



BAB III METODOLOGI DAN PERANCANGAN

3.1 Identifikasi Sistem

Dalam penelitian ini, akan dibangun sebuah perangkat lunak untuk peramalan data *time series* menggunakan algoritma genetika. Perangkat lunak ini merupakan sistem untuk memprediksi nilai data pada tahap selanjutnya pada rangkaian data yang disusun berurutan sesuai selang waktu yang tetap, dengan cara mencari pola datanya menggunakan algoritma genetika.

Pola data yang dicari, berdasarkan persamaan (2.3), merupakan rangkaian nilai variabel a_0 sampai a_k yang optimal yang digunakan sebagai koefisien untuk mendapatkan nilai ramalan (z).

Secara umum, langkah-langkah yang dilakukan dalam merancang pembangunan sistem ini adalah:

1. Menentukan sumber data yang akan digunakan
2. Menentukan tahapan algoritma genetika
3. Perancangan struktur data
4. Perancangan tampilan antar muka
5. Perancangan uji coba sistem

3.2 Sumber Data

Data yang digunakan dalam penelitian ini merupakan data *time series* berupa data nyata jumlah penjualan perbulan barang-barang komoditas dari bulan Januari 2000 sampai bulan Desember 2007. Data-data tersebut diperoleh dari *Website: Statistics South Africa* (www.statsonline.com).

3.3 Tahapan Algoritma Genetika

Tahapan algoritma genetika digunakan untuk mencari nilai variabel a_0 sampai a_k yang optimal, yang akan digunakan sebagai koefisien untuk mendapatkan nilai ramalan (z). Secara umum tahapan algoritma genetika ini dapat diilustrasikan dengan diagram alir pada Gambar 3.1.

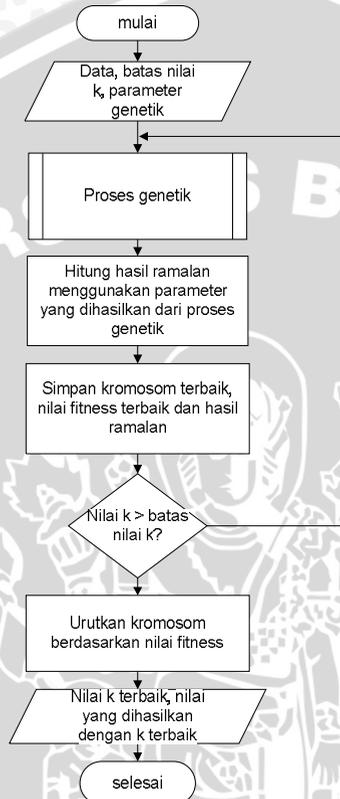


Gambar 3.1 Diagram alir proses algoritma genetika

Proses algoritma genetika tersebut dilakukan untuk menghitung hasil prediksi dengan menggunakan sebuah nilai periode peramalan (k). Untuk bisa mengetahui nilai k yang paling optimal, maka perlu dilakukan proses genetika secara berulang-ulang dengan menggunakan input nilai k yang berbeda-beda.

Untuk setiap nilai k yang dicari ini, akan dicari masing-masing nilai prediksinya secara berurutan dari nilai k terendah, yaitu 2, sampai batas nilai k yang telah ditentukan. Untuk kemudian dicari nilai

berapakah yang merupakan nilai k terbaik. Ilustrasi proses untuk mencari nilai k yang terbaik ditunjukkan pada gambar 3.2.

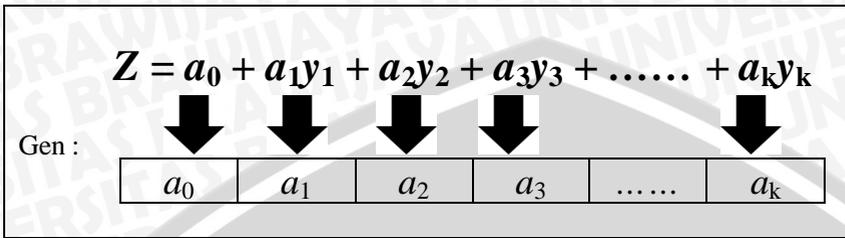


Gambar 3.2 Diagram alir proses pencarian nilai ramalan terbaik

3.3.1 Representasi Kromosom

Dalam algoritma genetika, satu kromosom harus merepresentasikan satu solusi yang dapat diterima. Dalam hal ini, satu kromosom harus merupakan satu model peramalan.

Dari persamaan (2.3), karena dalam hal ini yang dicari adalah nilai variabel dari a_0 sampai a_k yang optimal, maka kromosom merepresentasikan rangkaian bilangan dari a_0 sampai a_k tersebut. Sehingga setiap kromosom dibentuk dari $k+1$ buah gen. Proses ini diilustrasikan pada Gambar 3.3.



Gambar 3.3 Representasi kromosom

Variabel a_0 sampai a_k merupakan bilangan bernilai real antara 0 dan 1, dengan ketelitian hingga 4 angka dibelakang koma. Oleh karena itu, maka untuk masalah ini representasi kromosomnya menggunakan *real-number encoding*.

3.3.2 Fungsi *Fitness*

Suatu hasil peramalan akan dikatakan baik adalah apabila nilai kesalahannya semakin kecil. Solusi akan optimum bila nilai kesalahannya sama dengan 0. Permasalahan seperti ini dikenal dengan permasalahan minimasi. Solusi akan semakin baik, jika nilai *fitness*-nya semakin kecil.

Cara menghitung nilai kesalahan peramalan yaitu dengan menghitung selisih nilai sebenarnya (z) dengan nilai hasil peramalan (z'). Nilai hasil peramalan didapatkan dari persamaan (2.3) dengan koefisien variabel a_0 sampai a_k didapat dari rangkaian kromosom.

Karena proses peramalan dilakukan berulang-ulang sesuai dengan jumlah masukan datanya, maka penghitungan nilai kesalahan peramalan dilakukan juga secara berulang-ulang. Sehingga nilai *fitness* yang digunakan adalah nilai rata-rata kesalahan absolut peramalan (*mean absolute error*). Nilai rata-rata kesalahan absolut peramalan dapat dituliskan seperti pada persamaan (3.1).

$$\frac{1}{n} \sum_{p=1}^n |z_p - z'_p| \tag{3.1}$$

Dari persamaan (3.1), n adalah jumlah proses peramalan, z_p adalah nilai sebenarnya pada proses peramalan ke p dan z'_p adalah nilai hasil ramalan pada proses peramalan ke- p .

Langkah-langkah untuk menghitung nilai fitness sebuah rangkaian kromosom ditunjukkan dengan menggunakan *pseudo code* yang dijabarkan berikut ini.

```
//jumdat menunjukkan jumlah data yang akan diproses
//k menunjukkan nilai tahapan peramalan (K)
//data(n) menunjukkan nilai data ke-n
for i = 0 to ( jumdat - k )
    stop = i + (k-1);
    posisi = k;
    for n = i to stop
        hasilkali = kromosom(posisi) * data(n);
        hasiltot = hasiltot + hasilkali;
        posisi = posisi - 1;
    next n
    hasiltot = hasiltot + kromosom(0);
    selisih = absolute( data(i + k ) - hasiltot );
    hasiltot = 0;
    jumselisih =jumselisih +selisih;
next i
fitness = ( jumselisih / (jumdata - k ) )
```

3.3.3 Seleksi Orang Tua

Proses seleksi orang tua dilakukan untuk menentukan individu mana di dalam populasi yang akan terpilih untuk melakukan *crossover*. Proses ini bisa dilakukan beberapa kali tergantung nilai probabilitas *crossover* yang dimasukkan. Semakin besar nilai probabilitas *crossover*-nya, maka individu yang terpilih untuk melakukan *crossover* akan semakin banyak.

Proses seleksi orang tua ini dilakukan secara proporsional sesuai dengan nilai *fitness*. Hal ini bertujuan untuk memperbesar kemungkinan individu yang memiliki nilai *fitness* yang lebih baik akan terpilih untuk melakukan *crossover*. Metode seleksi proporsional yang dipakai yaitu dengan menggunakan metode *roulette wheel*.

Dalam metode seleksi *roulette wheel*, karena dalam hal ini yang dihadapi adalah permasalahan minimasi, maka langkah pertama yang dilakukan adalah dengan menentukan besarnya nilai invers *fitness* setiap individu. Kemudian langkah selanjutnya yaitu menentukan daerah *roulette* setiap individu yang nilainya proporsional dengan nilai

invers *fitness* yang dimiliki. Semakin besar nilai invers *fitness*-nya, maka daerah *roulette* yang dimilikinya akan semakin besar, sehingga peluang untuk terpilih juga semakin besar.

Langkah yang terakhir yaitu dengan membangkitkan bilangan acak antara 1 sampai 100. Individu yang terpilih menjadi orang tua adalah individu yang memiliki daerah *roulette* yang mencakup bilangan acak tadi.

Dari ilustrasi tadi, maka untuk proses seleksi orang tua ini dapat disusun sebuah *pseudo code* sebagai berikut:

```
//menghitung nilai jumlah invers fitness dalam satu populasi
//fitness(i) menyatakan nilai fitness individu ke-i
for i = 1 to npop
    invers(i) = 1/fitness(i);
    juminv = juminv + invers(i);
next i

//menempatkan individu pada roulette wheel
for j = 1 to npop
    invfitness(i) = 1/fitness(i);
    skala(i) = (invfitness(i)/juminv) * 100;
    jumskala = jumskala + skala(i);
    daerah(i) = jumskala;
next j

//memilih induk dengan memutar roulette wheel
rdm = random(0,100);
for m = 1 to npop
    if ( rdm <= daerah(m) );
        induk = individu(m);
        break for;
next m
```

3.3.4 Crossover

Apabila proses seleksi orang tua telah dilakukan, maka proses berikutnya adalah *crossover* 2 kromosom tersebut. *Crossover* adalah cara mengkombinasikan gen-gen induk untuk menghasilkan keturunan baru.

Karena representasi kromosom yang digunakan adalah representasi *real*, maka metode *crossover* yang digunakan adalah metode *arithmetic-crossover*, menggunakan persamaan (2.1) dan (2.2) dengan nilai r yang berada diantara 0 dan 10 dihasilkan secara random.

Penerapan *arithmetic crossover* dalam *crossover* kromosom orang tua diilustrasikan dengan *pseudo code* berikut:

```
//nilai r dihasilkan dengan membangkitkan bilangan acak
//nilai (i) menunjukan gen ke (i) dalam sebuah rangkaian kromosom
r = random (0,1)
for n = 0 to Pkromosom
    anak1(i) = ( r * induk1(i) ) + ( (1 - r) * induk2(i) );
    anak2(i) = ( (1 - r) * induk1(i) ) + ( r * induk2(i) );
next n
hitungfitness(anak1, anak2);
```

Anak yang dihasilkan dari proses tadi (anak1 dan anak2) merupakan individu baru yang memiliki susunan kromosom berbeda dengan induknya. Oleh karena itu, maka setelah proses *crossover* ini terbentuk, perlu dipanggil lagi fungsi hitung fitness untuk menghitung nilai kesesuaian rangkaian kromosom pada anak1 dan anak2.

3.3.5 Mutasi

Proses mutasi diperlukan dalam algoritma genetika untuk mempercepat proses evolusi dan mencegah konvergensi prematur, yaitu kondisi dimana semua gen bernilai hampir seragam sehingga nilai optimal tidak bisa tercapai.

Proses mutasi tidak dilakukan pada setiap individu pada populasi. Tetapi pada individu-individu yang terpilih saja. Besarnya peluang kromosom individu akan dimutasi atau tidak, ditentukan dari nilai probabilitas mutasi (P_{mut}) yang dimasukkan. Penggunaan probabilitas mutasi untuk menentukan apakah kromosom akan dimutasi pada algoritma genetika bisa diterapkan dengan cara membangkitkan nilai acak antara 0 sampai 1. Bila nilai yang dihasilkan berada di antara nilai 0 dan nilai P_{mut} , maka kromosom tersebut dimutasi, bila diluar itu, maka kromosom tidak dimutasi.

Proses mutasi pada representasi kromosom *real*, dapat dilakukan menggunakan metode penggantian nilai secara acak, dan penambahan

nilai. Dalam penelitian ini, metode yang digunakan adalah metode penggantian nilai secara acak.

Langkah pertama dalam mutasi dengan metode penggantian secara acak yaitu dengan memilih kromosom individu yang akan dimutasi. Setelah itu, maka langkah selanjutnya yaitu mengganti nilai gen-gen tersebut dengan nilai *real* baru. Setelah nilai gen dalam individu yang termutasi berubah, maka perlu dilakukan perhitungan nilai fitness dengan memanggil fungsi fitness. Proses ini dapat diilustrasikan dengan *pseudo code* sebagai berikut:

```
//menentukan apakah sebuah individu akan dimutasi
for h = 0 to npop
  rdm = random(0,1);
  if ( pmut > rdm )
    for i = 0 to Pkromosom
      //mengganti nilai gen ke – i pada individu ke-h dengan
      // nilai baru yang dihasilkan secara random
      genbaru = random(0,1);
      individu(h)(i) = genbaru;
    next i
  hitungfitness( individu(h) )
next h
```

3.3.6 Seleksi Populasi

Dalam algoritma genetika, jumlah individu dalam populasi selalu tetap pada setiap generasi. Maka setelah proses *crossover* dan mutasi yang menciptakan individu baru, perlu dilakukan seleksi populasi untuk menjaga agar individu dengan *fitness* yang paling baik saja yang ada di populasi.

Individu baru akan masuk menjadi anggota populasi hanya bila nilai *fitness*-nya lebih baik daripada nilai *fitness* satu atau beberapa anggota populasi yang lain. Individu baru tersebut masuk ke menjadi anggota populasi menggantikan individu lain yang nilai *fitness*-nya paling rendah.

Untuk itu, maka setelah proses *crossover* dan mutasi selesai dilakukan, semua individu yang terbentuk dimasukkan dulu menjadi anggota populasi, kemudian seluruh anggota populasi tersebut diurutkan sesuai nilai *fitness* yang dimiliki. Setelah itu diambil individu yang terbaik sebanyak nilai jumlah populasi, untuk

membentuk populasi baru. Sedangkan individu yang tidak terpilih untuk membentuk populasi baru, akan dihilangkan. Proses ini dapat diilustrasikan dengan *pseudo code* berikut:

```
//memasukkan anak yang telah dihasilkan ke dalam populasi
for i = 1 to jumanak
    n = (npop + i );
    individu(n) = anak(i);
next i

//mengurutkan (sorting) individu sesuai nilai fitness
for i = 1 to (npop+jumanak)
    for j = npop down to i
        if (individu(j)(f) < individu(j - 1)(f) )
            temp = individu(j);
            individu(j) = individu(j-1);
            individu(j-1) = temp;
    previous j
next i
```

Setelah proses pengurutan dilakukan, maka bisa dikatakan bahwa rangkaian kromosom terbaik dalam satu generasi terdapat pada individu pertama.

Pada iterasi selanjutnya (generasi selanjutnya), individu yang akan diproses adalah individu yang berada pada posisi teratas sebanyak nilai jumlah populasi, sehingga secara otomatis, individu yang berada diluar individu yang terpilih untuk generasi selanjutnya tersebut akan hilang.

3.4 Struktur Data

Struktur data yang digunakan terdiri dari dua buah array. Sebuah array digunakan untuk menyimpan data *time series* yang akan diproses. Dan sebuah array 2 dimensi (matriks) untuk menyimpan kromosom individu-individu dalam populasi

Untuk menyimpan data yang akan diproses, digunakan sebuah matriks berukuran panjang sebanyak jumlah data.

Misalkan data *time series* yang akan diproses ada pada tabel 3.1:

Tabel 3.1 Data yang akan diproses

Periode	Jumlah
Juli 2007	11.149
Agustus 2007	11.640
September 2007	12.018
November 2007	11.948
Desember 2007	12.159
Januari 2008	12.078
Februari 2008	11.436

Dengan $n = 7$, maka data pada matriks ditunjukkan pada tabel 3.2:

Tabel 3.2 Contoh penyimpanan data pada matriks

Data[0]	Data[1]	Data[2]	Data [3]	Data [4]	Data [5]	Data [6]
11.149	11.640	12.018	11.948	12.159	12.078	11.436

Sedangkan untuk menyimpan rangkaian kromosom dalam individu-individu di populasi, disimpan dalam array 2 dimensi (matriks) dengan panjang kolom matriks sebesar $k+2$ dan panjang baris matriks sebesar jumlah populasi yang dimasukkan. Kolom 0 sampai ke k untuk menyimpan gen yang mewakili nilai a_0 sampai a_k , sedangkan kolom ke $k+1$ untuk menyimpan nilai *fitness*-nya dan kolom ke $k+2$ untuk menyimpan daerah roulette. Masing-masing baris digunakan untuk menyimpan rangkaian kromosom tiap individu.

Penggunaan matriks populasi ini dapat diilustrasikan sebagai berikut:

Misalkan jumlah periode ramalan (k) = 3. Dan jumlah individu dalam populasi = 5. Maka data pada matriks populasi seperti ditunjukkan dalam tabel 3.3.

Tabel 3.3 Contoh penyimpanan data populasi pada matriks

	a_0	a_1	a_2	a_3	<i>fitness (MSE)</i>	<i>Daerah roulette</i>
1	0.2364	0.4312	0.4231	0.7634	19.703	31.6511
2	0.4231	0.1323	0.4635	0.6553	19.7918	63.1153
3	0.2341	0.6452	0.3343	0.7687	31.4027	82.9284
4	0.4321	0.3314	0.6545	0.9782	47.7264	96.0125
5	0.1512	0.0442	0.7683	0.4415	156.3254	100

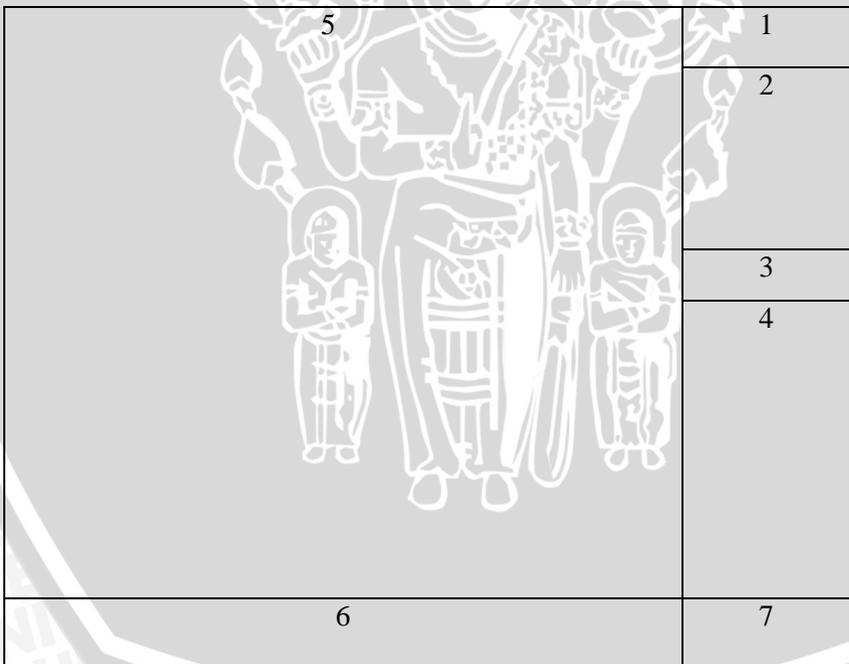
3.5 Perancangan Tampilan Antar Muka

Bagian-bagian antar muka yang dibutuhkan pengguna antara lain adalah:

- Input:
 1. Mengambil data yang akan diproses
 2. Nilai masukan parameter genetik
 3. Tombol untuk memulai proses penghitungan
 4. Tempat data yang akan diproses

- Output:
 5. Hasil iterasi proses genetika
 6. Kromosom terbaik yang dihasilkan
 7. Nilai hasil ramalan

Dari bagian-bagian tersebut, dapat dirancang sebuah tampilan antar muka pengguna seperti pada Gambar 3.4.



Gambar 3.4 Rancangan tampilan antar muka

3.6 Perancangan Uji Coba

Untuk mengevaluasi hasil peramalan yang dihasilkan, maka perlu dilakukan uji coba. Dengan input data, periode ramalan dan parameter genetik yang berbeda, hasil ramalan yang dihasilkan bisa berbeda pula. Oleh karena itu, perlu dilakukan pengujian keberhasilan hasil peramalan dengan berbagai kondisi input yang berbeda.

Parameter keberhasilan data ramalan yang dihasilkan adalah nilai persentasi kesalahan (*error rate*) bila dibandingkan dengan data sebenarnya. Nilai persentasi kesalahan ini bisa diukur dengan persamaan 3.2.

$$ErrorRate = \left(\frac{NilaiSebenarnya - NilaiRamalan}{NilaiSebenarnya} \right) \times 100\% \quad (3.2)$$

Nilai hasil ramalan bisa dikatakan semakin akurat bila nilai kesalahan (*error rate*) semakin kecil.

3.6.1 Pengujian Variasi Data Masukan

Dengan parameter genetik yang sama, tetapi menggunakan jenis data dan jumlah data *time series* yang berbeda, bisa didapatkan hasil nilai *fitness* dan persentasi akurasi yang berlainan. Untuk itu perlu dilakukan pengujian dengan data masukan dan jumlah data yang berbeda-beda. Pengujian berdasarkan data masukan akan dilakukan berdasarkan tabel 3.4.

Nilai yang dicatat pada setiap percobaan merupakan nilai *fitness* terbaik dan nilai persentasi kesalahan yang diukur menggunakan persamaan 3.2.

Tabel 3.4 Tabel pengujian variasi data masukan

Data masukan	Jumlah data				
	10	30	50	70	90
Data 1					
Data 2					
Data 3					

Variasi data yang dimasukkan sebanyak 3 jenis, yaitu data *time series* penjualan komoditas manufaktur, data penjualan komoditas tekstil dan data penjualan komoditas elektronik. Data tersebut

kemudian dimasukkan di dalam sistem secara bergantian masing-masing sebanyak 10, 30, 50, 70 dan 90 data.

Semua ujicoba tersebut dilakukan dengan menggunakan parameter genetik, yaitu jumlah generasi, jumlah individu dalam populasi, nilai probabilitas *crossover* dan nilai probabilitas mutasi yang sama.

3.6.2 Pengujian Variasi Parameter Genetik

Parameter genetik yang merupakan masukan dalam sistem ini adalah jumlah generasi (n), jumlah individu dalam populasi (n_{pop}) dan nilai probabilitas mutasi (P_{mut}). Sedangkan nilai probabilitas *crossover* (P_{cros}) ditetapkan sebesar 0,5.

Perbedaan parameter genetik yang digunakan dapat menyebabkan perbedaan nilai *fitness* dan persentasi akurasi hasil peramalan. Untuk itu perlu dilakukan pengujian dengan parameter genetik yang berbeda-beda.

Pengujian dengan berbagai variasi parameter genetik akan dilakukan berdasarkan tabel 3.5 dan tabel 3.6.

Pengujian pertama dilakukan dengan menggunakan jumlah generasi yang berbeda, yaitu 200, 300, 400, 500 dan 600 generasi. Nilai *fitness* terbaik yang didapatkan pada masing-masing pengujian kemudian dicatat dan ditempatkan pada tabel 3.5. Pengujian ini bermaksud untuk mengetahui pada generasi ke berapakah nilai *fitness* yang terbaik telah tercapai. Nilai yang dicatat pada ujicoba ini adalah nilai *fitness* terbaiknya.

Pada tabel 3.6, dengan 3 variasi masukan jumlah populasi, yaitu 5, 10 dan 20 generasi. Masing-masing jumlah generasi diujicobakan dengan 3 variasi nilai probabilitas *crossover*, yaitu 0,3, 0,8 dan 1, dan dengan 3 variasi nilai probabilitas mutasi, yaitu 0,1; 0,3 dan 0,8. Semua ujicoba tadi dilakukan dengan jumlah data masukan (n) dan jumlah generasi yang sama. Pengujian ini bertujuan untuk mengetahui pengaruh *input* parameter genetik terhadap nilai *fitness* terbaik yang dihasilkan.

Nilai yang dicatat pada setiap percobaan merupakan nilai *fitness* terbaik dan nilai persentasi kesalahan yang diukur menggunakan persamaan 3.2.

Tabel 3.5 Tabel pengujian variasi jumlah generasi

Jumlah generasi	Hasil
200	
300	
400	
500	
600	

Tabel 3.6 Tabel pengujian variasi parameter genetik

N pop (i)	$P_{crs} 0.3$			$P_{crs} 0.8$			$P_{crs} 1$		
	$P_{mut} 0.1$	$P_{mut} 0.3$	$P_{mut} 0.8$	$P_{mut} 0.1$	$P_{mut} 0.3$	$P_{mut} 0.8$	$P_{mut} 0.1$	$P_{mut} 0.3$	$P_{mut} 0.8$
5									
10									
20									

3.7 Contoh Perhitungan

Misalkan dari data *time series* dengan $n = 6$ pada tabel 3.7 akan diramalkan data periode selanjutnya, dengan periode ramalan 2 dan dengan masukan parameter genetik yang digunakan yaitu, jumlah generasi 1, jumlah individu dalam populasi 3 nilai probabilitas *crossover* 0.5 dan nilai probabilitas mutasi 0.5.

Langkah-langkah algoritma genetika yang dilakukan secara berurutan adalah, membangkitkan populasi awal, penghitungan nilai *fitness*, seleksi orang tua, *crossover*, mutasi, dan seleksi populasi, untuk mendapatkan individu dengan nilai kromosom terbaik.

Setelah diapatkan kromosom dengan nilai *fitness* terbaik dari proses algoritma genetika, maka rangkaian kromosom yang merepresentasikan variabel bilangan dari a_0 sampai a_k tersebut digunakan untuk meramalkan data pada periode selanjutnya dengan menggunakan persamaan (2.3).

Dalam contoh perhitungan ini, dimisalkan data yang akan diproses seperti pada tabel 3.7.

Tabel 3.7 Contoh data yang akan diproses

Periode	Jumlah
Januari 2008	560
Februari 2008	551
Maret 2008	564
April 2008	586
Mei 2008	613
Juni 2008	598

3.7.1 Membangkitkan Populasi Awal

Dalam algoritma genetika, kromosom individu-individu pada populasi awal dibangkitkan dari bilangan acak.

Kromosom individu yang dibangkitkan sebanyak jumlah populasi (n_{pop}) yaitu 3. Misalnya populasi awal yang terbentuk seperti pada tabel 3.8.

Tabel 3.8 Populasi awal yang terbentuk

	a_0	a_1	a_2
Individu ke-1 :	0.5542	0.2131	0.8021
Individu ke-2 :	0.6730	0.1085	0.4239
Individu ke-3 :	0.0519	0.6541	0.2552

3.7.2 Penghitungan Nilai *Fitness*

Dari populasi yang terbentuk sebelumnya, kemudian masing-masing individu dihitung nilai *fitness*-nya. Penghitungan nilai *fitness* individu ke-1 ditunjukkan pada tabel 3.9, individu ke-2 pada tabel 3.10 dan individu ke 3 pada tabel 3.11. Nilai z' didapatkan dari persamaan (2.3).

Tabel 3.9 Perhitungan *fitness* individu ke-1

y_2 (H-2)	y_1 (H-1)	z (H)	$y_1 \times a_1$	$y_2 \times a_2$	z'	$ z-z' $
560	551	564	119.34	441.96	561.85	2.15
551	564	586	117.42	452.38	570.35	15.65
564	586	613	120.19	470.03	590.78	22.22
586	613	598	124.87	491.69	617.12	19.12
$ z-z' $ total						59.14
<i>fitness</i> ($(z-z' /total) / j-k$) :						14.78

Tabel 3.10 Perhitungan *Fitness* individu ke-2

y_2 (H-2)	y_1 (H-1)	z (H)	$y_1 \times a_1$	$y_2 \times a_2$	z'	$ z-z' $
560	551	564	60.76	233.56	294.99	269.01
551	564	586	59.78	239.08	299.53	286.47
564	586	613	61.19	248.41	310.27	302.73
586	613	598	63.58	259.85	324.10	273.9
$ z-z' $ total						1132.11
<i>fitness</i> ($(z-z' /total) / j-k$)						283.03

Tabel 3.11 Perhitungan *fitness* individu ke-3

y_2 (H-2)	y_1 (H-1)	z (H)	$y_1 \times a_1$	$y_2 \times a_2$	z'	$ z-z' $
560	551	564	366.29	140.62	506.96	57.04
551	564	586	360.41	143.93	504.40	143.98
564	586	613	368.91	149.55	518.51	94.5
586	613	598	383.30	156.44	539.79	58.21
$ z-z' $ total						353.73
<i>fitness</i> ($(z-z' /total) / j-k$)						88.43

3.7.3 Seleksi Orang Tua

Jumlah probabilitas mutasi yang dimasukkan adalah 0.5. Berarti bila jumlah individu adalah 3, maka kemungkinan individu yang terpilih sebagai orang tua sebanyak 1.5 individu. Karena untuk proses *crossover* diperlukan paling tidak 2 individu, sehingga dalam seleksi orang tua ini hanya ada dua kemungkinan, yaitu terpilih 2 individu sebagai orang tua atau 0 individu (tidak ada yang terpilih sebagai orang tua) . Disini misalnya yang terpilih adalah kemungkinan pertama, yaitu akan diseleksi 2 individu unuk melakukan *crossover*.

Seleksi orang tua dilakukan dengan menggunakan metode *roulette wheel*. Langkah pertama yang dilakukan yaitu dengan mengurutkan individu-individu dalam populasi sesuai nilai *fitness*, kemudian masing-masing individu tersebut dihitung daerah *roulette*-nya. Penempatan individu dalam *roulette wheel* ditunjukkan pada tabel 3.12.

Tabel 3.12 Penempatan individu dalam *roulette wheel*

	Kromosom Individu	<i>fitness</i>	invers <i>fitness</i>	skala	daerah <i>roulette</i>
1	0,2131 0,8021 0,5542	14.78	0.0676	82.03	1 – 82.03
3	0,6541 0,2552 0,0519	88.43	0.0113	13.71	82.04 – 95.75
2	0,1085 0,4239 0,6730	283.03	0.0035	4.24	95.76 – 100

Selanjutnya, untuk menentukan individu mana yang akan melakukan *crossover*, dibangkitkan 2 bilangan acak antara 0 sampai 100. Misalkan bilangan acak yang dihasilkan adalah 14 dan 85, maka individu yang terpilih menjadi orang tua yaitu individu ke-1 dan individu ke-3.

3.7.4 *Crossover*

Setelah individu yang menjadi orang tua telah dipilih, maka langkah selanjutnya yaitu melakukan *crossover* kromosom-kromosom tersebut.

Crossover dilakukan dengan metode *arithmetic crossover* dengan nilai r dihasilkan dari bilangan acak. Misalkan nilai r yang dihasilkan adalah 0.2, maka proses *crossover* yang dilakukan adalah sebagai berikut.

Induk 1: 0.5542 0.2131 0.8021

Induk 2: 0.0519 0.6541 0.2552

Crossover dengan $r = 0,2$

Anak 1₍₁₎ : $0.2 * 0.5542 + 0.8 * 0.0519 = 0.1523$

Anak 2₍₁₎ : $0.2 * 0.0519 + 0.8 * 0.5542 = 0.4551$

Anak 1₍₂₎ : $0.2 * 0.2131 + 0.8 * 0.6541 = 0.5658$

Anak 2₍₂₎ : $0.2 * 0.6541 + 0.8 * 0.2131 = 0.3308$

Anak 1₍₃₎ : $0.2 * 0.8021 + 0.8 * 0.2552 = 0.3645$

Anak 2₍₃₎ : $0.2 * 0.2552 + 0.8 * 0.8021 = 0.6920$

Dari perhitungan, dapat diperoleh kromosom anak:

Anak 1: 0.1523 0.5031 0.3645

Anak 2: 0.4551 0.3308 0.6920

Proses penghitungan *fitness* kromosom Anak 1 dan Anak 2 ditunjukkan pada tabel 3.13 dan tabel 3.14.

Tabel 3.13 Penghitungan *fitness* Anak 1

y_2 (H-2)	y_1 (H-1)	z (H)	$y_1 \times a_1$	$y_2 \times a_2$	z'	$ z-z' $
560	551	564	311.76	483.45	795.64	231.64
551	564	586	319.11	475.68	795.22	209.22
564	586	613	331.56	486.90	818.89	205.89
586	613	598	346.84	505.89	771.01	173.01
$ z-z' $ total						819.76
$fitness ((z-z' /total) / j-k)$						204.94

Tabel 3.14 Penghitungan *fitness* Anak 2

y_2 (H-2)	y_1 (H-1)	z (H)	$y_1 \times a_1$	$y_2 \times a_2$	z'	$ z-z' $
560	551	564	182.27	387.52	570.24	6.24
551	564	586	186.57	381.29	568.32	17.68
564	586	613	193.85	390.29	584.59	28.41
586	613	598	202.78	405.51	608.74	10.74
$ z-z' $ total						63.07
$fitness ((z-z' /total) / j-k)$:						15.77

Dari perhitungan pada tabel 3.19 dan tabel 3.20, didapatkan nilai *fitness* Anak 1 adalah 204.94 dan nilai *fitness* Anak 2 adalah 15.77.

3.7.5 Mutasi

Langkah pertama dalam proses mutasi yaitu dengan melakukan pengecekan apakah rangkaian kromosom pada individu akan dimutasi. Caranya dengan membangkitkan bilangan acak antara 0 sampai 1. Proses mutasi akan dilakukan bila bilangan acak yang muncul berada dibawah nilai probabilitas mutasi yang dimasukan.

Nilai probabilitas mutasi yang dimasukan adalah 0.2. Misalkan bilangan acak yang muncul untuk Individu 1 sampai 3 adalah 0.75; 0.14; 0.73 Anak 1 adalah 0.25 dan untuk Anak 2 adalah 0.87, maka proses mutasi akan dilakukan terhadap individu 2 saja. Tidak dilakukan mutasi pada individu 1, individu 3 atau pada anak 1 dan anak 2.

Proses mutasi dilakukan dengan metode penggantian nilai gen-gen dalam kromosom, dengan nilai baru yang dihasilkan secara acak. Proses mutasi pada individu ke-1, misalnya dihasilkan kromosom

Kromosom individu 2 : 0,1085 0,4239 0,6730

Kromosom individu 2 setelah dimutasi : **0.5412 0.2918 0.7373**

Maka nilai *fitness*-nya dapat dihitung dengan proses seperti yang terdapat pada tabel 3.15

Tabel 3.15 Penghitungan *fitness* individu ke-1 setelah dimutasi

y_2 (H-2)	y_1 (H-1)	z (H)	$y_1 \times a_1$	$y_2 \times a_2$	z'	$ z-z' $
560	551	564	160.78	412.89	574.21	10.21
551	564	586	164.57	406.25	571.36	14.64
564	586	613	170.99	415.84	587.37	25.63
586	613	598	178.87	432.06	611.47	13.47
$ z-z' $ total						63.95
<i>fitness</i> ($(z-z' /total) / j-k$)						15.98

Setelah proses mutasi ini, maka susunan individu di dalam populasi menjadi seperti yang terdapat pada tabel 3.16.

Tabel 3.16 populasi setelah proses mutasi

	Kromosom Individu			<i>fitness</i>
Individu 1	0,2131	0,8021	0,5542	14.78
Individu 3	0,6541	0,2552	0,0519	88.43
Individu 2	0.5412	0.2918	0.7373	15.98
Anak 1	0.1523	0.5031	0.3645	204.94
Anak 2	0.4551	0.3308	0.6920	15.77

3.7.6 Seleksi Populasi

Dari proses *crossover* dan mutasi yang telah dilakukan sebelumnya, telah tercipta 2 rangkaian kromosom (individu) baru. Yaitu anak 1 dan anak 2. Individu baru yang terbentuk ini, kemudian akan dirurutkan sesuai *fitness*nya dengan seluruh individu yang lain seperti pada table 3.17.

Selanjutnya, karena di dalam populasi baru harus selalu berisi 3 individu, maka 2 individu dengan nilai *fitness* paling rendah, yaitu individu 3 dan anak 1 tidak terseleksi sebagai anggota populasi baru.

Anggota populasi baru setelah dilakukan seleksi populasi ditunjukkan pada tabel 3.18.

Tabel 3.17 Mengurutkan seluruh individu berdasarkan *fitness*

	a_0	a_1	a_2	<i>Fitness</i>
Individu 1 :	0.5542	0.2131	0.8021	14.78
Anak 2:	0.4551	0.3308	0.6920	15.77
Individu 2 :	0.5412	0.2918	0.7373	15.98
Individu 3 :	0.6541	0.2552	0.0519	88.43
Anak 1 :	0.1523	0.5031	0.3645	204.94

Tabel 3.18 Populasi baru yang terbentuk

	a_0	a_1	a_2	<i>fitness</i>
Individu 1 :	0.5542	0.2131	0.8021	14.78
Individu 2 :	0.4551	0.3308	0.6920	15.77
Individu 3 :	0.4318	0.5658	0.8633	204.94

Setelah proses pengurutan dilakukan, akan selalu didapatkan rangkaian kromosom terbaik pada individu ke-1.

Maka pada generasi ke-1 ini didapatkan kromosom terbaik yaitu, 0.5542 0.2131 0.8021. Dengan nilai *fitness* 14.78.

3.7.7 Peramalan Data

Dari proses algoritma genetika yang telah dilakukan, didapatkan kromosom terbaik yaitu: 0.5542 0.2131 0.8021. Nilai-nilai ini kemudian digunakan sebagai variabel bilangan dari a_0 sampai a_2 untuk mendapatkan nilai ramalan data periode selanjutnya.

Dari tabel 3.12, data periode selanjutnya yang dicari adalah data pada tanggal 7 Januari 2008. Dengan menggunakan persamaan (2.3) hasil ramalan (z') dapat diperoleh dengan proses berikut ini.

$$z' = a_0 + a_1y_1 + a_2y_2$$

$$z' = 0.5542 + (598*0.2131) + (613*0.8021)$$

$$z' = 619.12$$

Maka, nilai data yang akan muncul pada tanggal 7 Januari 2008 dengan nilai $k = 2$ diramalkan sebesar 619.12.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini akan dijelaskan implementasi seluruh proses yang sudah dirancang pada bab sebelumnya disertai dengan analisa data yang dihasilkan.

4.1 Lingkungan Implementasi

Dalam pengembangan sistem peramalan data *time series* menggunakan algoritma genetika ini, digunakan lingkungan perangkat keras dan perangkat lunak tertentu.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem peramalan data *time series* menggunakan algoritma genetika ini adalah

1. Prosesor Intel T5300 1.73 GHz
2. Memori 1536 Mbytes
3. Harddisk dengan kapasitas 80 GB
4. Monitor 14.4"
5. Keyboard
6. Mouse

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem peramalan data *time series* menggunakan algoritma genetika ini adalah

1. Sistem Operasi Windows XP
2. PHP 5.1.4
3. Apache versi Windows 5.1.3a
4. Macromedia Dreamweaver 8

4.2 Implementasi Program

Dari analisa dan perancangan sistem yang terdapat pada sub bab 3.1, maka pada sub bab ini akan dijelaskan implementasi proses-proses tersebut secara lebih rinci.

4.2.1 Inisialisasi Kromosom

Dalam algoritma genetika, tahap awal yang dilakukan adalah membangkitkan rangkaian kromosom sebanyak jumlah individu dalam populasi. Rangkaian kromosom yang dibangkitkan adalah susunan bilangan acak dari 0 sampai 1 sepanjang nilai periode peramalan, dengan ketelitian hingga 4 angka dibelakang koma. Fungsi untuk membangkitkan susunan kromosom ini dapat dilihat pada Gambar 4.1

```
1 function inisialisasi($krmin, $npop, $k)
2 {
3     global $krmin;
4     for ($j=1; $j <= $npop; $j++)
5     {
6         print ("<BR>Individu ke $j :<BR>");
7         for ($i = 0; $i <= $k; $i++)
8         {
9             $random = mt_rand(0, 10000);
10            $random = $random/10000;
11            $krmin[$j][$i] = $random;
12        }
13    }
14
15    return $krmin;
16 }
```

Gambar 4.1 Inisialisasi kromosom awal

Rangkaian kromosom yang dihasilkan kemudian disimpan di dalam matriks “\$krmin”. Dengan setiap baris matriks untuk menyimpan satu individu dan setiap kolom matriks untuk menyimpan nilai setiap gen-nya.

4.2.2 Perhitungan Nilai *Fitness*

Nilai *fitness* digunakan untuk menentukan nilai kebaikan sebuah individu. Pada sistem ini, *fitness* dihitung dengan cara menghitung nilai rata-rata kesalahan absolut peramalan (*mean absolute error*).

Nilai *fitness* yang dihasilkan dari proses penghitungan ini kemudian disimpan di dalam kolom ke $k+1$ di setiap baris individu.

```

1 function hitungfitness($krmin, $h, $j, $ardata, $k)
2 {
3     $stemp = array();
4     $arselisih = array();
5
6     for($i = 0; $i < ($j - ($k)); $i++)
7     {
8         $t=0;
9         $stop = $i+($k-1);
10        $bobot = $k;
11        for ($n = $i; $n <= ($stop); $n++)
12        {
13            $stemp = $krmin[$h+1][$bobot] ;
14            $hasils = $ardata[$n] * $stemp;
15
16            $z += $hasils;
17            $bobot --;
18        }
19
20        $z += $krmin[$h+1][0];
21        $arselisih[$i] = abs($ardata[$i+$k]-$z);
22        $z=0;
23        $sum += $selisihs[$i];
24    }
25
26    $fitness = ($sum / ($j - $k));
27    $fitness = round($fitness,4);
28    $krmin[$h+1][($k+1)]= $fitness;
29
30    $sum = 0;
31    return $krmin;
32 }

```

Gambar 4.2 Hitung *fitness*

4.2.3 Mengurutkan Kromosom Sesuai Nilai *Fitness*

Setelah nilai *fitness* tiap individu telah didapatkan, maka langkah selanjutnya yaitu mengurutkan kromosom-kromosom di dalam populasi sesuai dengan nilai *fitness*-nya. Karena fungsi *fitness* yang digunakan merupakan fungsi minimasi, maka pengurutan dilakukan secara ascending (dari nilai terkecil ke nilai yang lebih besar).

Pada sistem ini metode pengurutan yang digunakan adalah metode *bubble sort*, dengan menggunakan variabel “\$stemp” bertipe array untuk tempat penyimpanan sementara. Fungsi untuk mengurutkan kromosom bisa dilihat pada Gambar 4.3.

```

1 function urutfitness($krmin, $npop, $k)
2 {
3     $temp = array();
4
5     for ($i = 1 ; $i <= $npop ; $i++)
6     {
7         for ($j = $npop; $j >= $i ; $j--)
8         {
9             if($krmin[$j][($k+1)] < $krmin[$j-1][($k+1)])
10            {
11                for($m = 0; $m <= ($k+1); $m++)
12                {
13                    $temp[$m] = $krmin[$j][$m];
14                    $krmin[$j][$m] = $krmin[$j-1][$m];
15                    $krmin[$j-1][$m] = $temp[$m];
16                }
17            }
18        }
19    }
20
21    return $krmin;
22 }

```

Gambar 4.3 Mengurutkan kromosom sesuai nilai *fitness*

4.2.4 Crossover

Sebelum proses *crossover* dilakukan, terlebih dahulu diperlukan proses seleksi untuk menentukan individu mana yang akan terpilih untuk melakukan proses *crossover*. Di dalam sistem ini metode seleksi yang dipakai yaitu metode *roulette wheel*.

Gambar 4.4 menunjukkan proses penghitungan daerah *roulette* untuk masing-masing kromosom, sesuai dengan nilai *fitness* yang dimilikinya. Nilai daerah *roulette* tadi kemudian disimpan di indeks ke $k+2$ di dalam setiap baris individu. Sehingga misalnya untuk individu ke- n , maka nilai daerah *roulette*-nya disimpan pada array “\$krmin” di dalam indeks ke “[n][k+2]”.

Setelah nilai daerah *roulette* untuk masing-masing individu telah didapatkan, maka proses selanjutnya yaitu membangkitkan bilangan acak (“\$rdm”). Individu yang terpilih untuk melakukan *crossover* adalah individu yang memiliki daerah *roulette* yang mencakup nilai acak tadi.

```

1 function roulette($krmin, $npop, $k)
2 {
3     $juminv = 0;
4
5     for ($i=1; $i<=$npop ; $i++)
6     {
7         $x = $krmin[$i][$k+1];
8         $inv = round ((1/$x), 4);
9         $juminv += $inv;
10    }
11
12    for ($i=1; $i<=$npop ; $i++)
13    {
14        $x = $krmin[$i][$k+1];
15        $inv = round ((1/$x), 4);
16        $skala = round((($inv/$juminv) *100), 4);
17        $jskala += $skala;
18        $krmin[$i][$k+2] = $jskala;
19    }
20
21    return $krmin;
22 }

```

Gambar 4.4 Menghitung daerah *roulette wheel*

Proses pemilihan individu yang akan melakukan *crossover* ini ditunjukkan pada Gambar 4.5. Indeks individu yang terpilih kemudian disimpan pada variabel “\$kros”.

```

1 function slkrs($krmin, $npop, $k)
2 {
3     $rdm = mt_rand(1,10000);
4     $rdm = ($rdm/100);
5
6     $kros = 1;
7     for ($j = 1; $j <= $npop; $j++)
8     {
9         if ($rdm <= $krmin[$j][($k+2)])
10        {
11            $kros = $j;
12            break;
13        }
14    }
15
16    return $kros;
17 }

```

Gambar 4.5 Memilih individu yang akan melakukan *crossover*

Karena dalam sistem ini menggunakan kromosom bernilai *real* (representasi *real*), maka metode *crossover* yang dipakai adalah

metode aritmatik. Nilai r yang digunakan merupakan bilangan acak antara 0.01 sampai 0.99. Fungsi untuk melakukan proses *crossover* aritmatik ini ditunjukkan pada Gambar 4.6.

```

1 function crossover($krmin,$k,$anak,$kros1,$kros2,$i,$jcrs)
2 {
3     $r = mt_rand(1, 99);
4     $r = ($r/100);
5
6     for ($m = 0; $m < ($k+1) ; $m++)
7     {
8         $nilai1 = ( $r * ($krmin[$kros1][$m]) + (1 - $r)
9             * ($krmin[$kros2][$m]) );
10        $anak[($i+1)][$m] = round ($nilai1, 4);
11
12        $nilai2 = ( (1 - $r) * ($krmin[$kros1][$m]) + $r
13            * ($krmin[$kros2][$m]) );
14        $anak[($i+2)][$m] = round ($nilai2, 4);
15    }
16    return $anak;

```

Gambar 4.6 Proses *crossover*

Proses *crossover* ini akan mengambil input berupa 2 individu yang telah terpilih dari proses seleksi sebelumnya dan kemudian akan membentuk 2 kromosom anak. Kromosom anak yang terbentuk tadi selanjutnya disimpan di dalam matriks “\$anak”, dengan setiap baris pada matriks untuk menyimpan rangkaian kromosom setiap anak yang terbentuk.

```

1 function masukpop($anak,$krmin,$jcrs,$npop,$k,$jumlahanak)
2 {
3     for ( $i =1; $i <= $jumlahanak ; $i++)
4     {
5         for ($j = 0; $j < ($k+2); $j++)
6         {
7             $x = ($npop + $i);
8             $temp = $krmin[($npop+$i)][$j];
9             $krmin[$x][$j] = $anak[$i][$j];
10            $anak[$i][$j] = $temp;
11        }
12    }
13
14    return $krmin;
15 }

```

Gambar 4.7 Memasukkan kromosom anak ke populasi

4.2.5 Proses Mutasi

Proses mutasi dilakukan dengan menggunakan metode penggantian nilai gen dengan nilai lain yang dibangkitkan secara acak. Fungsi untuk melakukan mutasi dengan penggantian secara acak ini ditunjukkan pada Gambar 4.8.

```
1 function mutasi($k,$krmin,$jumdat,$i)
2 {
3     for ($m = 0 ; $m <= $k; $m++)
4     {
5         $random = mt_rand(0, 10000);
6         $random = $random/10000;
7         $krmin[$i][$m] = $random;
8         print ("Kromosom ke $i , , $m= ".$krmin[$i][$m]);
9     }
10
11     return $krmin;
12 }
```

Gambar 4.8 Proses mutasi

Fungsi mutasi ini mengambil masukan berupa “\$krmin” sebagai matriks penyimpanan kromosom-kromosom individu di dalam populasi, dan “\$i” sebagai indeks individu yang akan dimutasi. Nilai gen baru yang dihasilkan dari proses ini kemudian disimpan kembali ke dalam matriks “\$krmin” dengan menindih nilai gen yang sebelumnya.

4.2.6 Perulangan Proses *Crossover*

Proses ini dilakukan untuk menentukan berapa banyak individu yang akan melakukan proses *crossover* dalam setiap generasi. Jumlah individu yang akan melakukan *crossover* tadi bergantung pada nilai probabilitas *crossover* yang dimasukkan. Semakin besar nilai probabilitas *crossover*-nya, maka kemungkinan jumlah individu yang akan melakukan proses *crossover* akan semakin banyak.

Proses ini akan memanggil fungsi *slkcrs* untuk memilih individu yang akan melakukan *crossover*, memanggil fungsi *crossover* untuk melakukan proses *crossover*, kemudian memanggil fungsi *hitungfitness* untuk menghitung nilai *fitness* kromosom anak yang telah terbentuk. Proses perulangan *crossover* ini dapat dilihat pada Gambar 4.9.

```

1 function drivecrs($npop,$probc,$krmin,$k,$anak,$kros1,
2     $kros2, $jumanak, $jcrs, $jumanak, $jumdat, $ardata)
3 {
4     $jumanak = 0;
5     for ($h = 1 ; $h <= ($npop/2) ; $h++)
6     {
7         $rdm = mt_rand(0,100);
8         $rdm = ($rdm/100);
9         if ($probc > $rdm)
10        {
11            $kros1 = slkcrs($krmin, $npop, $k);
12            do ($kros2 = slkcrs ($krmin, $npop, $k));
13                while ($kros2 == $kros1);
14            $anak = crossover($krmin, $k, $anak, $kros1,
15                $kros2,$jumanak, $jcrs);
16            $anak = hitungfitness($anak, $jumanak, $jumdat,
17                $ardata, $k);
18            $anak = hitungfitness($anak, ($jumanak+1),
19                $jumdat,$ardata, $k);
20            $jumanak = $jumanak+2;
21        }
22    }
23 }

```

Gambar 4.9 Perulangan proses *crossover*

4.2.7 Seleksi Proses Mutasi

Proses ini dilakukan untuk menseleksi apakah sebuah individu akan dimutasi atau tidak. Bila sebuah individu setelah diseleksi ternyata memenuhi nilai probabilitas mutasi, maka akan dilakukan proses mutasi pada individu tersebut. Proses seleksi individu untuk dimutasi ini dapat dilihat pada Gambar 4.10.

```

1 function drivemut($npop, $jumanak, $probm, $krmin, $k,
2     $jumdat $ardata)
3 {
4     for ($i = 2; $i <= ($npop+$jumanak); $i++)
5     {
6         $ran = mt_rand(0,100);
7         $ran = ($ran/100);
8         if ($ran <= $probm)
9         {
10            print("<BR>Individu yg mutasi : ke- $i<BR>");
11            $krmin = mutasi($k, $krmin, $jumdat, $i);
12            $krmin = hitungfitness($krmin, ($i-1),$jumdat,
13                $ardata, $k);
14        }
15    }
16 }

```

Gambar 4.10 Seleksi proses mutasi

Pada tahap ini, proses seleksi individu yang akan melakukan mutasi dimulai dari individu kedua sampai individu terakhir. Hal ini dilakukan untuk menjaga agar nilai kromosom individu pertama, yang merupakan individu dengan nilai *fitness* terbaik tidak berubah karena proses mutasi.

Proses mutasi yang dilakukan akan mengubah nilai kromosom pada individu, sehingga pada setiap proses pemanggilan fungsi mutasi perlu dilakukan proses penghitungan kembali nilai *fitness* dengan cara memanggil fungsi *hitungfitness*.

4.2.8 Perulangan Proses Genetik

Fungsi “*loopgen*” digunakan untuk melakukan perulangan proses genetik seperti perhitungan nilai *fitness*, pengurutan individu sesuai nilai *fitness*, proses *crossover*, dan proses *mutasi*.

Di dalam proses ini dipanggil fungsi “*inisialisasi*”, untuk proses membangkikan populasi awal, kemudian fungsi “*hitungfitness*” untuk menghitung nilai *fitness*-nya. Setelah generasi awal terbentuk, maka dilakukan proses *looping* sebanyak jumlah generasi.

Pada proses *looping* sebanyak jumlah generasi ini, terdapat pemanggilan fungsi “*urutfitness*” untuk mengurutkan individu berdasarkan nilai *fitness*-nya, fungsi “*roulette*” untuk menghitung daerah *roulette* tiap individu, fungsi “*drivecrs*” untuk melakukan proses seleksi orangtua dan proses persilangan, dan fungsi “*drivemut*” untuk melakukan proses seleksi individu yang akan dimutasi dan proses mutasi itu sendiri.

Kode program untuk proses perulangan proses genetik ini ditunjukkan pada gambar 4.11.

```

1 function loopgen($ardata, $krmin, $npop, $k, $ngen, $probc,
2     $probm)
3     {
4         $krmin = inisialisasi($krmin, $npop, $k);
5         //hitung fitness populasi awal
6         for ($h = 0; $h<$npop; $h++)
7             {
8                 $krmin = hitungfitness($krmin, $h, $jumdat, $ardata,
9                     $k);
10            }
11        //looping proses genetik dari generasi ke 1 sampai ngen
12        for ($g=1; $g <= $ngen; $g++)
13            {
14                //mengurutkan individu berdasarkan fitness
15                $krmin = urutfitness($krmin, $npop, $k);
16
17                //memanggil fungsi roulette
18                $krmin = roulette($krmin, $npop, $k);
19
20                $jumanak = 0;
21                $anak =drivecrs($npop,$probc,$krmin,$k,$anak,
22                    $kros1,$kros2, $jumanak, $jcrs, $jumanak, $jumdat,
23                    $ardata);
24
25                $krmin = masukpop($anak, $krmin, $jcrs, $npop, $k,
26                    $jumanak);
27
28                $krmin = drivemut($npop, $jumanak, $probm, $krmin,
29                    $k, $jumdat $ardata);
30
31                $krmin = urutfitness($krmin,($npop+$jumanak), $k);
32
33                $sterbaik[$g] = $krmin[1][($k+1)];
34            }
35        return array($krmin, $ardata, $jumdat, $sterbaik);
36    }

```

Gambar 4.11 Perulangan proses genetik

4.2.9 Pengambilan Nilai Prediksi

Setelah proses iterasi sebanyak jumlah generasi untuk satu nilai k selesai dilakukan, maka kromosom terbaik yang dihasilkan adalah rangkaian kromosom yang terdapat pada individu pertama. Rangkaian kromosom ini kemudian digunakan sebagai koefisien untuk menghitung prediksi nilai selanjutnya berdasarkan persamaan 2.3.

Proses penghitungan nilai prediksi ini ditunjukkan pada Gambar 4.12. Variabel “\$ind” pada listing program tersebut menunjukkan

jumlah data yang dimasukkan. Sehingga “\$ardata[\$ind]” berisi data pada posisi yang terakhir.

Kode program pada baris ke-10 di gambar 4.12 bertujuan untuk menambahkan nilai a_0 pada hasil perhitungan perkalian antara koefisien data (yang berada pada rangkaian kromosom) dengan data aktual yang ada.

```
1 function ambilhasil($k,$ind, $krmin,$ardata)
2 {
3     for ($i = 0; $i < $k; $i++)
4     {
5         $max += ($krmin[1][$i] * $ardata[$ind]);
6         print ("<BR>". $krmin[1][$i]. " X ". $ardata[$ind]);
7         $ind--;
8     }
9
10    $max += $krmin[1][0];
11    return $max;
12 }
```

Gambar 4.12 Menghitung nilai prediksi akhir

4.2.10 Pengambilan Nilai Terbaik pada Setiap Nilai k

Untuk setiap nilai k yang akan dihitung, akan diproses dengan proses genetika yang berbeda pula. Untuk itu diperlukan proses perulangan (iterasi) seluruh proses genetika untuk tiap nilai k yang dicari.

Pada setiap iterasi ini, nilai *fitness*, nilai hasil prediksi dan rangkaian kromosom yang terbaik disimpan di dalam matriks “\$best”. Kemudian setelah semua proses iterasi selesai dilakukan, maka akan diambil hasil ramalan dan rangkaian kromosom yang memiliki nilai *fitness* terbaik yang tersimpan di dalam matriks “\$best” tadi dan kemudian ditetapkan sebagai hasil ramalan dan rangkaian kromosom terbaik.

Proses perulangan untuk tiap nilai k dan proses pengambilan nilai *fitness* terbaik dari semua hasil tiap proses iterasi ditunjukkan pada Gambar 4.13.

```

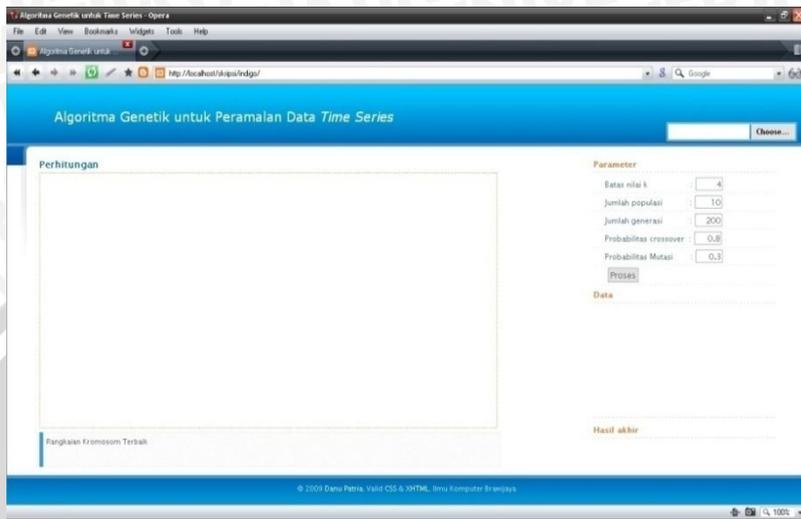
1  for ($k = 2; $k <= $batas; $k++)
2  {
3      //fungsi loopgen merupakan fungsi bertugas untuk
4      //melakukan proses genetik
5      list($krmin,$ardata, $jumdat, $sterbaik) =
6      loopgen($sumber, $ardata, $krmin, $npop, $k, $ngen,
7      $probc, $probm);
8
9      $sind = ($jumdat-1);
10     $smax = ambilhasil($b,$sind,$krmin,$ardata);
11
12     $best[0][$idx] = $k; //simpan nilai k - nya
13     $best[1][$idx] = $krmin[1][($k+1)]; //simpan fitness
14     $best[2][$idx] = $smax; //seimpan nilai ramalan
15
16     $idx +=1;
17     //simpan rangkaian kromosomnya
18     for($t = 0; $t<=$b; $t++)
19     {
20         $krbest[$k][$t] = $krmin[1][$t];
21     }
22 }
23
24 for($y = 0; $y < $idx; $y++)
25 {
26     for($z = ($idx-1); $z >= $y; $z--)
27     {
28         if($best[1][$z] < $best[1][($z-1)])
29         {
30             for ($x=0; $x<3; $x++)
31             {
32                 $temp = $best[$x][$z];
33                 $best[$x][$z] = $best[$x][($z-1)];
34                 $best[$x][($z-1)] = $temp;
35             }
36         }
37     }
38 }
39
40 print("nilai k terbaik = ".$best[0][0]
41     ."fitness terbaik = ".$best[1][0]
42     ."nilai ramalan terbaik = ".$best[2][0].");

```

Gambar 4.13 Mencari nilai *k* terbaik

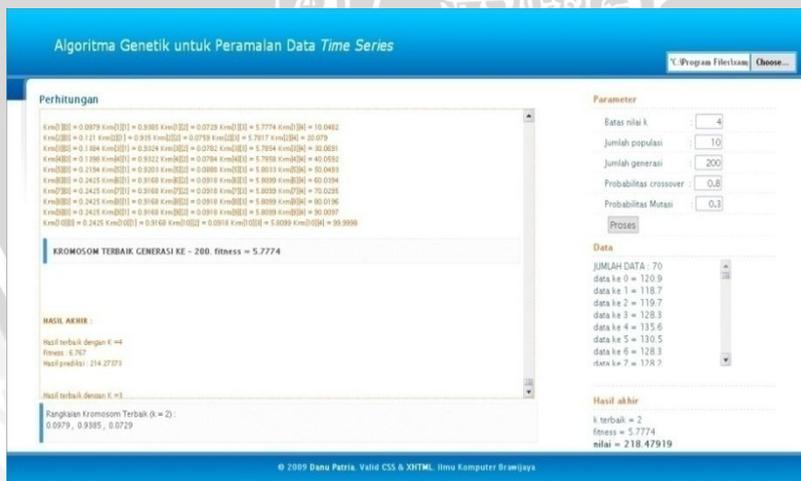
4.3 Implementasi Antarmuka (Interface)

Berdasarkan rancangan antarmuka yang terdapat pada sub bab 3.4. Maka pada sub bab ini akan dijelaskan hasil antarmuka untuk penerapan algoritma genetika dalam peramalan data penjualan secara *time series* ini.



Gambar 4.14 Tampilan utama pada browser

Gambar 4.14 adalah tampilan utama ketika sistem dijalankan menggunakan browser Opera versi 9.61. User kemudian diminta untuk memilih sumber data yang akan diproses, kemudian memasukkan parameter genetik yang diminta, yaitu batas nilai k , jumlah individu populasi, jumlah generasi, nilai probabilitas *crossover* dan probabilitas mutasi.



Gambar 4.15 Tampilan setelah dijalankan

Setelah dijalankan, sistem kemudian akan melakukan perhitungan secara algoritma genetik yang hasilnya akan ditampilkan kembali ke dalam browser seperti pada Gambar 4.15.

4.4 Uji Coba Sistem

Dari sistem yang telah dibuat sebelumnya, maka pada sub bab ini akan dijabarkan pembahasan mengenai proses pengujian yang telah dilakukan pada sistem dan hasil proses pengujian tersebut.

4.4.1 Pengujian Hasil Dengan Nilai k yang Berbeda

Pada tahap ini akan dibandingkan nilai hasil ramalan yang dihasilkan dengan menggunakan nilai k optimal yang didapatkan sistem dengan nilai hasil ramalan yang dihasilkan dengan menggunakan nilai k yang lain.

Dilakukan 2 kali pengujian dengan menggunakan jenis data input yang berbeda, yaitu data *time series* penjualan komoditas manufaktur dan data *time series* penjualan komoditas elektronik, masing-masing sebanyak 100 data. Input parameter genetik yang digunakan yaitu jumlah generasi sebanyak 400 generasi, jumlah populasi 10, probabilitas *crossover* 0.8, dan probabilitas mutasi 0.5.

Nilai aktual untuk data *time series* penjualan komoditas manufaktur adalah 117 dan untuk data *time series* penjualan komoditas elektronik adalah 96.5. Nilai persentasi kesalahan dihitung menggunakan persamaan 2.3.

Hasil pengujian dengan menggunakan data *time series* penjualan komoditas manufaktur disajikan pada tabel 4.1, dan untuk data *time series* penjualan komoditas elektronik disajikan pada tabel 4.2. Nilai yang dicetak tebal merupakan nilai k terbaik yang didapatkan sistem. Pada data pertama nilai k terbaik yang dihasilkan adalah 4, dan pada data yang kedua adalah 5.

Tabel 4.1 Perbandingan hasil dengan menggunakan nilai k yang berbeda pada data penjualan komoditas manufaktur

Nilai k	Nilai <i>fitness</i>	Nilai hasil	Persentasi kesalahan
2	5.9508	116.549	0.388
3	5.5248	116.125	0.7478
4	5.3537	116.338	0.712
5	6.0647	115.168	1.57
6	5.8367	115.205	1.534
7	6.2579	112.59	3.801
8	8.7334	119.28	1.965
9	12.6279	122.51	4.741
10	15.8627	130.06	2.637

Tabel 4.2 Perbandingan hasil dengan menggunakan nilai k yang berbeda pada penjualan komoditas elektronik

Nilai k	Nilai <i>fitness</i>	Nilai hasil	Persentasi kesalahan
2	8.2295	103.524	7.278
3	7.1185	102.142	5.846
4	7.1054	99.784	3.403
5	7.0172	97.98	1.533
6	7.3607	96.948	0.982
7	7.3668	97.232	0.758
8	17.1083	114.591	8.384
9	63.5492	120.57	24.94
10	84.6368	136.434	41.37

Nilai k terbaik yang dihasilkan sistem merupakan hasil yang memiliki nilai *fitness* terbaik. Sehingga diharapkan nilai ramalan yang dihasilkan memiliki akurasi lebih baik. Bisa dilihat dari hasil pengujian bahwa hasil ramalan dengan menggunakan nilai k terbaik relatif lebih mendekati nilai aktual. Walaupun ada beberapa hasil peramalan dengan menggunakan nilai k lain yang memiliki akurasi lebih baik.

4.4.2 Pengujian Berdasarkan Jumlah Data Input

Pada tahap ini akan disajikan hasil pengujian sistem dengan menggunakan data input yang berbeda-beda, yaitu data *time series* penjualan komoditas manufaktur, data penjualan komoditas tekstil dan data penjualan komoditas elektronik. Nilai aktual (nilai sebenarnya) yang dicari untuk data-data tersebut dapat dilihat pada tabel 4.3

Tabel 4.3 Nilai aktual (nilai sebenarnya) yang dicari

Jenis data	Nilai aktual
Data penjualan komoditas manufaktur	115.5
Data penjualan komoditas tekstil	98.5
Data penjualan komoditas elektronik	104.4

Setiap jenis data tadi kemudian diujikan dengan memasukkan 10 data, 30 data, 50 data, 70 data dan 90 data. Masing-masing pengujian dilakukan dengan menggunakan batas k dengan nilai 6, dan parameter genetik yang sama, yaitu jumlah populasi 10, jumlah generasi 400, probabilitas *crossover* 0.8, dan probabilitas mutasi 0.5.

Hasil pengujian berdasarkan data-data input tersebut disajikan pada tabel 4.4, tabel 4.5 dan tabel 4.6.

Nilai persentasi kesalahan yang dicantumkan pada masing-masing tabel dihitung dengan menggunakan persamaan 2.3.

Tabel 4.4 Persentasi kesalahan peramalan data *time series* penjualan komoditas manufaktur

Jumlah data	K terbaik	Nilai <i>fitness</i> (MAE)	Nilai hasil	Persentasi kesalahan
10	4	5.2015	116.69	1.03 %
30	4	5.126	116.38	0.761 %
50	4	5.5615	115.78	1.242 %
70	4	5.458	116.31	0.701 %
90	4	5.0472	116.48	0.848 %

Tabel 4.5 Persentasi kesalahan peramalan data *time series* penjualan komoditas tekstil

Jumlah data	K terbaik	Nilai <i>fitness</i> (MAE)	Nilai hasil	Persentasi kesalahan
10	3	7.836	92.551	6.04 %
30	3	9.149	89.614	9.021 %
50	4	9.528	86.682	11.99 %
70	4	9.859	84.181	14.53 %
90	3	10.62	86.303	12.38 %

Tabel 4.6 Persentasi kesalahan peramalan data *time series* penjualan komoditas elektronik

Jumlah data	K terbaik	Nilai <i>fitness</i> (MAE)	Nilai hasil	Persentasi kesalahan
10	5	4.348	99.411	4.77 %
30	3	7.804	105.42	0.97 %
50	5	7.18	102.09	2.21 %
70	3	6.807	102.65	1.67 %
90	6	6.969	102.016	2.28 %

Dari pengujian tersebut didapatkan bahwa pada masing-masing jenis data, jumlah data yang dimasukkan untuk pengujian memiliki pengaruh yang berbeda-beda terhadap hasil peramalan. Hal ini disebabkan karena perbedaan karakteristik masing-masing data yang digunakan untuk pengujian.

4.4.3 Pengujian Berdasarkan Input Jumlah Generasi

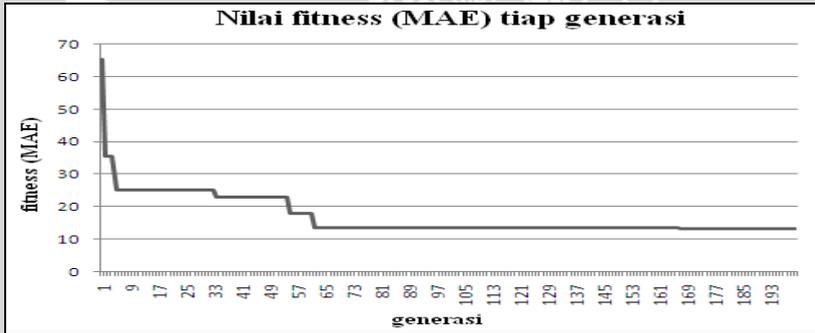
Pada tabel 4.7 disajikan perbandingan nilai *fitness* terbaik yang dihasilkan sistem dengan menggunakan input parameter jumlah generasi yang berbeda-beda, yaitu 200 generasi, 300 generasi, 400 generasi, 500 generasi dan 600 generasi. Masing-masing pengujian dilakukan dengan menggunakan data, batas nilai *k* dan parameter genetik lain yang sama, yaitu data *time series* penjualan komoditas tekstil dengan jumlah 90 data, batas nilai *k* = 6, jumlah populasi 10, probabilitas *crossover* 0.8 dan probabilitas mutasi 0.5.

Pengujian ini dilakukan untuk mengetahui pada generasi ke berapakah nilai *fitness* yang optimal telah dicapai.

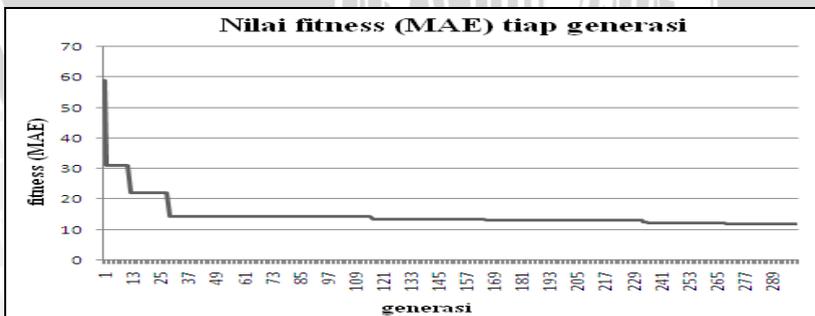
Tabel 4.7 Nilai *fitness* terbaik yang dihasilkan dengan *input* jumlah generasi yang berbeda.

Jumlah generasi	Nilai <i>fitness</i> (MAE) terbaik
200	13.24
300	11.91
400	11.51
500	11.53
600	11.54

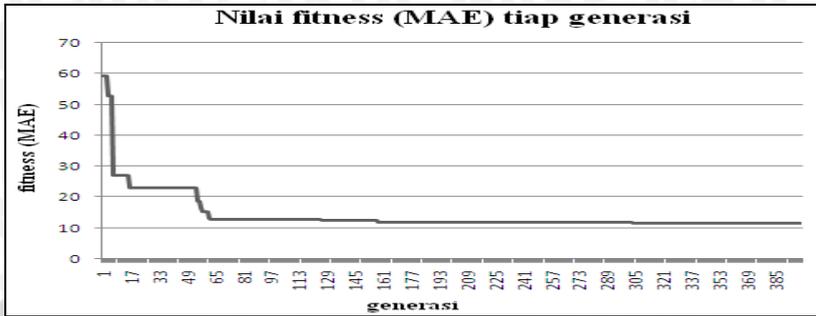
Kemudian pada Gambar 4.16, Gambar 4.17, Gambar 4.18, Gambar 4.19 dan Gambar 4.20 ditunjukkan grafik perkembangan nilai *fitness* terbaik tiap generasi dari generasi pertama hingga ke generasi terakhir masing-masing.



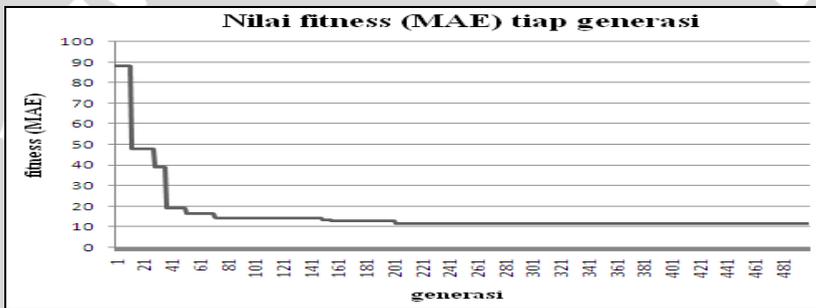
Gambar 4.16 Grafik perkembangan nilai *fitness* terbaik hingga generasi ke 200



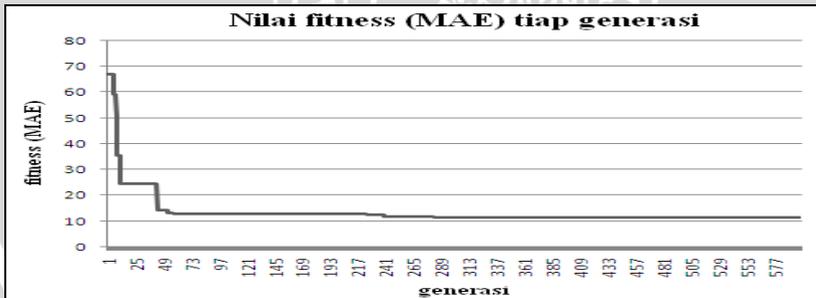
Gambar 4.17 Grafik perkembangan nilai *fitness* terbaik hingga generasi ke 300



Gambar 4.18 Grafik perkembangan nilai *fitness* terbaik hingga generasi ke 400



Gambar 4.19 Grafik perkembangan nilai *fitness* terbaik hingga generasi ke 500



Gambar 4.20 Grafik perkembangan nilai *fitness* terbaik hingga generasi ke 600

Dari grafik perkembangan nilai *fitness* yang telah ditampilkan, bisa dilihat bahwa sebenarnya nilai *fitness* maksimal telah tercapai sebelum

generasi ke-400. Setelah generasi ke-400, perkembangan nilai *fitness* yang terjadi tidak menunjukkan nilai yang signifikan lagi.

Pada seluruh ujicoba tersebut terdapat sedikit perbedaan pada generasi ke berapa nilai *fitness* maksimal didapatkan. Hal ini disebabkan karena pada setiap sistem dijalankan, akan terjadi perbedaan pada saat inialisasi populasi awal, pada proses *crossover* dan pada proses mutasinya.

4.4.4 Pengujian Berdasarkan Input Parameter Genetik

Hasil pengujian dengan menggunakan input parameter genetik yaitu probabilitas *crossover* dan probabilitas mutasi yang berbeda-beda. ditunjukkan pada tabel 4.8. Pada masing-masing proses pengujian tersebut, digunakan data yang sama, yaitu data *time series* penjualan komoditas manufaktur dengan jumlah 90 data, dengan batas nilai $k = 6$ dan dengan jumlah generasi yang sama, yaitu 400 generasi.

Dari data pada tabel 4.8 tersebut, dapat dilihat bahwa nilai *fitness* yang dihasilkan sistem akan menjadi relatif semakin baik bila nilai parameter genetik yang digunakan semakin besar.

Tabel 4.8 Persentasi kesalahan hasil peramalan dengan menggunakan input parameter genetik yang berbeda

N pop	Prob Crs	Prob Mts	K	Nilai <i>fitness</i> (MAE)	Nilai hasil	Persentasi Kesalahan (%)
5	0.3	0.1	4	5.43	117.1	1.38 %
5	0.3	0.3	4	5.21	116.59	0.94 %
5	0.3	0.8	4	5.55	116.35	0.73 %
5	0.8	0.1	4	5.38	116.54	0.9 %
5	0.8	0.3	3	5.75	115.32	0.15 %
5	0.8	0.8	4	5.34	116.58	0.93 %
5	1	0.1	3	5.82	114.11	1.2 %
5	1	0.3	4	5.24	116.67	1.01 %
5	1	0.8	4	5.1	116.26	0.65 %
10	0.3	0.1	3	5.55	115.42	0.06 %
10	0.3	0.3	4	5.45	114.97	0.45 %
10	0.3	0.8	4	5.05	116.33	1.69 %
10	0.8	0.1	4	5.34	117.13	1.41 %
10	0.8	0.3	4	5.29	115.82	0.27 %
10	0.8	0.8	4	5.21	116.65	0.99 %
10	1	0.1	4	5.42	115.98	0.41 %
10	1	0.3	4	5.53	115.01	0.42 %
10	1	0.8	4	5.29	115.85	0.3 %
20	0.3	0.1	4	5.27	115.91	0.35 %
20	0.3	0.3	4	5.26	116.64	0.98 %
20	0.3	0.8	4	5.21	116.08	0.5 %
20	0.8	0.1	4	5.44	115.35	0.12 %
20	0.8	0.3	4	5.47	116.3	0.69 %
20	0.8	0.8	4	5.12	116.28	0.67 %
20	1	0.1	4	5.27	116.71	1.04 %
20	1	0.3	4	5.34	116.78	1.1 %
20	1	0.8	4	5.14	116.29	0.01 %

4.4.5 Pengujian Akurasi Peramalan Secara Simultan

Ujicoba ini dilakukan dengan menggunakan parameter genetik tertentu yaitu jumlah generasi 400, jumlah individu dalam populasi 10, nilai probabilitas *crossover* 0.8, nilai probabilitas mutasi 0.5. Jumlah data yang dimasukkan sebanyak 70 data *history*. Dan batas nilai *k* yang digunakan yaitu 6.

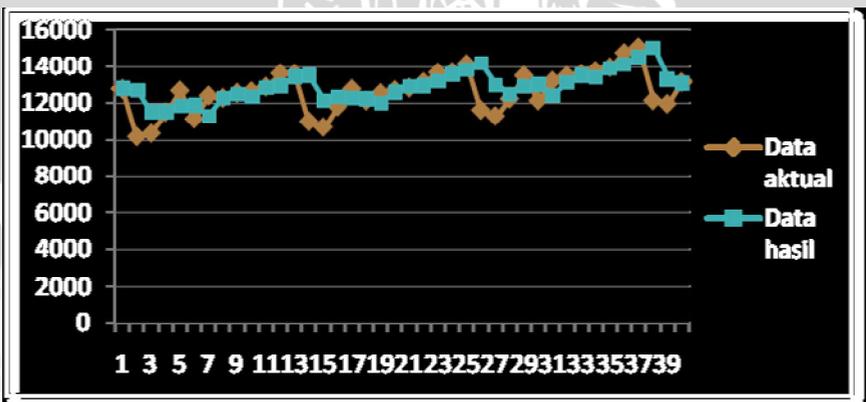
Untuk setiap jenis data, pengujian dilakukan secara simultan sebanyak 40 kali pengujian. Perbandingan data hasil prediksi dan data aktual yang tadi kemudian disajikan dalam bentuk tabel dan grafik.

Hasil pengujian dengan menggunakan data *time series* penjualan komoditas manufaktur ditunjukkan pada tabel 4.9. Dan grafik perbandingan data aktual dan data hasilnya pada gambar 4.21.

Tabel 4.9 Hasil pengujian 40 data secara berurutan pada data penjualan komoditas manufaktur

No.	Nilai Aktual (z)	Nilai hasil ramalan (z')	Selisih ($z - z'$)	Persentasi kesalahan
1	12753.9	12813.84	59.94	0.0047 %
2	10156.5	12695.07	2538.57	0.249945 %
3	10334.1	11537.34	1203.24	0.116434 %
4	11499.6	11537.34	37.74	0.003282 %
5	12665.1	11834.82	830.28	0.065557 %
6	11133.3	11886.99	753.69	0.067697 %
7	12387.6	11303.13	1084.47	0.087545 %
8	12265.5	12279.93	14.43	0.001176 %
9	12543	12485.28	57.72	0.004602 %
10	12620.7	12333.21	287.49	0.022779 %
11	12876	12857.13	18.87	0.001466 %
12	13597.5	12938.16	659.34	0.04849 %
13	13575.3	13435.44	139.86	0.010303 %
14	10989	13504.26	2515.26	0.228889 %
15	10678.2	12138.96	1460.76	0.136798 %
16	11799.3	12339.87	540.57	0.045814 %
17	12753.9	12265.5	488.4	0.038294 %
18	12121.2	12255.51	134.31	0.011081 %
19	12520.8	11986.89	533.91	0.042642 %
20	12687.3	12608.49	78.81	0.006212 %
21	12820.5	12909.3	88.8	0.006926 %
22	13120.2	12939.27	180.93	0.01379 %
23	13641.9	13194.57	447.33	0.032791 %

No.	Nilai Aktual (z)	Nilai hasil ramalan (z')	Selisih ($z - z'$)	Persentasi kesalahan
24	13675.2	13578.63	96.57	0.007062 %
25	14108.1	13827.27	280.83	0.019906 %
26	11610.6	14170.26	2559.66	0.220459 %
27	11288.7	12999.21	1710.51	0.151524 %
28	12221.1	12510.81	289.71	0.023706 %
29	13475.4	12899.31	576.09	0.042751 %
30	12121.2	13016.97	895.77	0.073901 %
31	13197.9	12402.03	795.87	0.060303 %
32	13475.4	13113.54	361.86	0.026853 %
33	13564.2	13504.26	59.94	0.004419 %
34	13741.8	13399.92	341.88	0.024879 %
35	13930.5	13889.43	41.07	0.002948 %
36	14718.6	14122.53	596.07	0.040498 %
37	15029.4	14511.03	518.37	0.03449 %
38	12132.3	15021.63	2889.33	0.238152 %
39	11943.6	13311.12	1367.52	0.114498 %
40	13120.2	13099.11	21.09	0.001607 %
Rata-rata nilai persentasi kesalahan:				0.0583 %



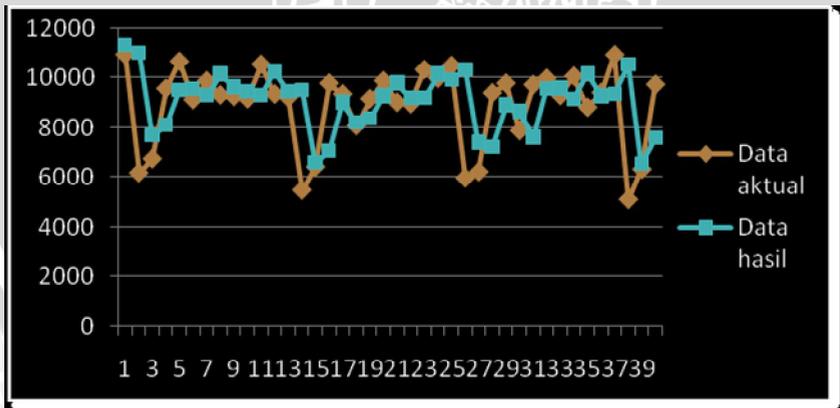
Gambar 4.21 Grafik perbandingan data aktual dengan data hasil prediksi pada data penjualan komoditas manufaktur

Hasil pengujian dengan menggunakan data *time series* penjualan komoditas tekstil ditunjukkan pada tabel 4.10 dan grafik perbandingan data aktual dan data hasilnya pada gambar 4.22.

Tabel 4.10 Hasil pengujian 40 data secara berurutan pada data penjualan komoditas tekstil

No.	Nilai Aktual (z)	Nilai hasil ramalan (z')	Selisih ($z - z'$)	Persentasi kesalahan
1	10944.6	11315.34	370.74	0.033874
2	6171.6	11011.2	4839.6	0.784173
3	6737.7	7671.21	933.51	0.13855
4	9568.2	8093.01	1475.19	0.154176
5	10667.1	9513.81	1153.29	0.108117
6	9124.2	9570.42	446.22	0.048905
7	9879	9290.7	588.3	0.059551
8	9312.9	10147.62	834.72	0.089631
9	9257.4	9637.02	379.62	0.041007
10	9157.5	9453.87	296.37	0.032364
11	10545	9292.92	1252.08	0.118737
12	9346.2	10257.51	911.31	0.097506
13	9213	9439.44	226.44	0.024578
14	5472.3	9492.72	4020.42	0.734686
15	6415.8	6571.2	155.4	0.024221
16	9779.1	7041.84	2737.26	0.279909
17	9346.2	9000.99	345.21	0.036936
18	8069.7	8206.23	136.53	0.016919
19	9157.5	8341.65	815.85	0.089091
20	9879	9276.27	602.73	0.061011
21	9013.2	9780.21	767.01	0.085099
22	8957.7	9187.47	229.77	0.025651
23	10334.1	9203.01	1131.09	0.109452
24	9978.9	10146.51	167.61	0.016796

No.	Nilai Aktual (z)	Nilai hasil ramalan (z')	Selisih ($z - z'$)	Persentasi kesalahan
25	10500.6	9915.63	584.97	0.055708
26	5960.7	10320.78	4360.08	0.731471
27	6216	7393.71	1177.71	0.189464
28	9390.6	7217.22	2173.38	0.231442
29	9779.1	8912.19	866.91	0.088649
30	7869.9	8651.34	781.44	0.099295
31	9723.6	7607.94	2115.66	0.21758
32	9978.9	9541.56	437.34	0.043826
33	9290.7	9574.86	284.16	0.030585
34	10067.7	9133.08	934.62	0.092834
35	8813.4	10152.06	1338.66	0.151889
36	9401.7	9268.5	133.2	0.014168
37	10933.5	9350.64	1582.86	0.144772
38	5094.9	10539.45	5444.55	1.068627
39	6315.9	6544.56	228.66	0.036204
40	9734.7	7561.32	2173.38	0.223261
Rata-rata nilai persentasi kesalahan:				0.1657



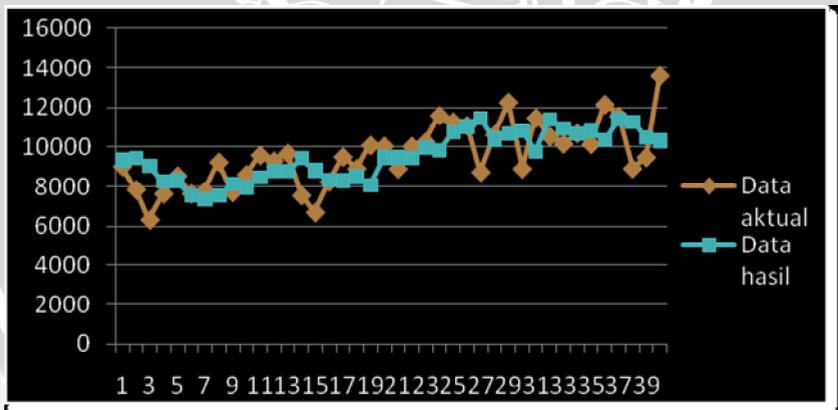
Gambar 4.22 Grafik perbandingan data aktual dengan data hasil prediksi pada data penjualan komoditas tekstil

Hasil pengujian dengan menggunakan data *time series* penjualan komoditas elektronik ditunjukkan pada tabel 4.11 dan grafik perbandingan data aktual dan data hasilnya pada gambar 4.23.

Tabel 4.11 Hasil pengujian 40 data secara berurutan pada data penjualan komoditas elektronik

No.	Nilai Aktual (z)	Nilai hasil ramalan (z')	Selisih ($z - z'$)	Persentasi kesalahan
1	9035.4	9343.98	308.58	0.034152
2	7858.8	9412.8	1554	0.19774
3	6338.1	9050.94	2712.84	0.428021
4	7647.9	8229.54	581.64	0.076052
5	8558.1	8276.16	281.94	0.032944
6	7647.9	7528.02	119.88	0.015675
7	7792.2	7340.43	451.77	0.057977
8	9246.3	7523.58	1722.72	0.186315
9	7714.5	8117.43	402.93	0.05223
10	8624.7	7949.82	674.88	0.07825
11	9601.5	8468.19	1133.31	0.118035
12	9312.9	8767.89	545.01	0.058522
13	9690.3	8764.56	925.74	0.095533
14	7559.1	9428.34	1869.24	0.247283
15	6704.4	8818.95	2114.55	0.315397
16	8247.3	8332.77	85.47	0.010363
17	9512.7	8283.93	1228.77	0.129172
18	8924.4	8485.95	438.45	0.049129
19	10101	8067.48	2033.52	0.201319
20	10056.6	9466.08	590.52	0.05872
21	8913.3	9470.52	557.22	0.062516
22	10045.5	9449.43	596.07	0.059337
23	10278.6	9943.38	335.22	0.032613
24	11610.6	9819.06	1791.54	0.154302
25	11288.7	10731.48	557.22	0.049361

No.	Nilai Aktual (z)	Nilai hasil ramalan (z')	Selisih ($z - z'$)	Persentasi kesalahan
26	11100	11043.39	56.61	0.0051
27	8746.8	11461.86	2715.06	0.310406
28	10633.8	10385.16	248.64	0.023382
29	12276.6	10680.42	1596.18	0.130018
30	8913.3	10801.41	1888.11	0.211831
31	11488.5	9762.45	1726.05	0.150242
32	10556.1	11387.49	831.39	0.078759
33	10178.7	10937.94	759.24	0.074591
34	10711.5	10699.29	12.21	0.00114
35	10156.5	10841.37	684.87	0.067432
36	12165.6	10334.1	1831.5	0.150547
37	11588.4	11403.03	185.37	0.015996
38	8924.4	11263.17	2338.77	0.262065
39	9490.5	10466.19	975.69	0.102807
40	13619.7	10323	3296.7	0.242054
Rata-rata nilai persentasi kesalahan:				0.1156



Gambar 4.23 Grafik perbandingan data aktual dengan data hasil prediksi pada data penjualan komoditas elektronik

Dari tahapan pengujian ini, didapatkan persentasi kesalahan rata-rata untuk setiap jenis data yang diujicobakan disajikan pada tabel 4.12. Yaitu sebesar 0.0583% untuk data penjualan komoditas manufaktur, 0.1657% untuk data penjualan komoditas tekstil dan 0.1156% untuk data penjualan komoditas elektronik. Rata-rata nilai persentasi kesalahan untuk semua data tersebut adalah sebesar 0.1132%

Tabel 4.12 Nilai persentasi kesalahan untuk tiap data ujicoba

Data	Persentasi kesalahan total	Persentasi kesalahan rata-rata
Data penjualan komoditas manufaktur	2.335%	0.0583 %
Data penjualan komoditas tekstil	6.63 %	0.1657 %
Data penjualan komoditas elektronik	4.627 %	0.1156 %
Rata-rata nilai persentasi kesalahan total :		0.1132 %



BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dari skripsi ini adalah:

1. Pada skripsi ini telah diterapkan metode algoritma genetika untuk menyelesaikan permasalahan pencarian nilai parameter untuk peramalan data secara *time series*.
2. Nilai periode peramalan yang digunakan untuk proses peramalan dengan metode ini merupakan nilai yang dihasilkan secara otomatis oleh sistem, dan merupakan nilai yang menghasilkan akurasi data yang relatif lebih baik bila dibandingkan dengan menggunakan nilai yang dimasukan secara manual.
3. Dalam ujicoba secara simultan untuk 40 data *time series* penjualan komoditas manufaktur didapatkan nilai rata-rata persentasi kesalahan sebesar 0.0583 %, untuk data *time series* penjualan komoditas tekstil sebesar 0.1657 % dan untuk data penjualan komoditas elektronik sebesar 0.1156 %. Untuk semua data yang diujicobakan, didapatkan rata-rata nilai kesalahan sebesar 0.1132 %.
4. Hasil pengujian tersebut menunjukkan bahwa data hasil prediksi yang dihasilkan menggunakan parameter dari metode algoritma genetika ini mampu mendekati data aktual (data sebenarnya) yang dicari dengan baik.
5. Parameter genetik yang dimasukkan sebagai input sangat mempengaruhi kinerja sistem. Semakin besar nilai jumlah generasi, jumlah individu dalam populasi, probabilitas mutasi, dan probabilitas *crossover*-nya, maka rata-rata nilai *fitness* yang dicapai akan relatif semakin baik.

5.2 Saran

Saran yang diberikan pada pembuatan skripsi ini:

1. Menerapkan algoritma genetika untuk peramalan data *time series* lainnya, seperti untuk harga saham, nilai tukar mata uang dan lain sebagainya.

2. Sebelum data dimasukkan ke dalam sistem, perlu dilakukan proses *preprocessing* terlebih dulu untuk menghilangkan data yang tidak relevan (*noise data/outlier*). Sehingga akurasi hasil bisa lebih meningkat.
3. Melakukan perbandingan nilai persentasi kesalahan yang dihasilkan menggunakan metode ini dengan metode peramalan data lain, misalnya metode jaringan syaraf tiruan atau logika *fuzzy*.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- Fariza, A., Nur Iman, B., Basuki, A. *Peramalan Penjualan Time Series Menggunakan Hybrid Algoritma Genetika Simulated Annealing*. Politeknik Elektronika Negeri Surabaya. <http://digilib.its.ac.id/detil.php?id=1612&q=hybrid>. Tanggal akses 3 April 2008.
- Gen, M and Cheng, R. *Genetic Algorithms And Engineering Design*. John Wiley & Sons, Inc. New York: 1997.
- Hui, A. *Using genetic Programming to Perform Time-Series Forecasting of Stock Prices*. Standford University. Standford. <http://www.genetic-programming.org/sp2003/Hui.pdf>. Tanggal akses: 5 April 2008
- Kusumadewi, S., Purnomo, H. 2005. *Penyelesaian Masalah Optimasi Dengan Teknik-Teknik Heuristik*. Penerbit Graha Ilmu. Yogyakarta.
- Mendenhall, W., Reinmuth, J.E. 1988. *Statistik Untuk Manajemen dan Ekonomi*. Penerbit Erlangga. Jakarta.
- Obitko, M. 1998. *Introduction to Genetic Algorithm*, <http://cs.felk.cvut.cz/~xobitko/ga/>. Tanggal akses: 28 Februari 2006
- Simangunsong, M. 2006. *Aplikasi Peramalan Multivariate Time-Series Dengan Metode Fungsi Transfer Arch-Garch*. Universitas Brawijaya. Malang.
- Suyanto. 2005. *Algoritma Genetika Dalam Matlab*. Penerbit Andi. Yogyakarta.
- Suyanto. 2008. *Evolutionary Computation. Komputasi Berbasis "Evolusi" dan "Genetika"*. Penerbit Informatika. Bandung.

UNIVERSITAS BRAWIJAYA



LAMPIRAN

Lampiran 1 Data yang digunakan dalam pengujian.

1. Data penjualan komoditas manufaktur:

85.8, 97.2, 98.9, 92.3, 96.9, 99.6, 100, 95.6, 102.3, 106.9, 105.5, 81.8, 83.6, 93.7, 97.9, 90.7, 95.9, 94.8, 98.3, 97.7, 100.9, 107.3, 109.9, 86.3, 85.2, 98.7, 104.8, 89, 99.9, 102.1, 100.1, 102.5, 104.8, 109.8, 113.6, 89.5, 91.1, 102.7, 106.8, 95.3, 100.2, 104, 102.8, 102.3, 103.1, 114.9, 117.6, 92.5, 93.7, 103.6, 108.2, 104.4, 110.3, 106.7, 108.8, 109.8, 111.4, 119.6, 119.8, 92.7, 94.4, 104.3, 108.4, 100, 105.7, 105.3, 108.7, 106.3, 108.7, 116.7, 114.9, 91.5, 93.1, 103.6, 114.1, 100.3, 111.6, 110.5, 113, 113.7, 116, 122.5, 122.3, 99, 96.2, 106.3, 114.9, 109.2, 112.8, 114.3, 115.5, 118.2, 122.9, 123.2, 127.1, 104.6, 101.7, 110.1, 121.4, 109.2, 118.9, 121.4, 122.2, 123.8, 125.5, 132.6, 135.4, 109.3, 107.6, 118.2

2. Data penjualan komoditas tekstil:

71.1, 96, 94.2, 92.8, 96.8, 99.3, 117.6, 104.6, 112.6, 123, 111.7, 69.1, 74.7, 108, 120.8, 92, 104.5, 114.3, 107.4, 105, 107.2, 104.8, 103.8, 57.5, 67, 95.8, 110.2, 98.1, 110.7, 110.9, 109.4, 112.2, 105.8, 112.2, 124.4, 67.2, 71.9, 109.8, 114.7, 111.1, 123.2, 112.1, 115.4, 124.4, 117.8, 124.9, 128.9, 70.3, 85.8, 102.1, 106.6, 92.9, 103.2, 97.5, 98.3, 92.6, 93.5, 106, 85.3, 53.3, 63.5, 90.5, 107, 92.7, 100.9, 97.8, 107.1, 99.6, 98.3, 101.3, 98.6, 55.6, 60.7, 86.2, 96.1, 82.2, 89, 83.9, 83.4, 82.5, 95, 84.2, 83, 49.3, 57.8, 88.1, 84.2, 72.7, 82.5, 89, 81.2, 80.7, 93.1, 89.9, 94.6, 53.7, 56.0, 84.6, 88.1, 70.9, 87.6, 89.9, 83.7, 90.7, 79.4, 84.7, 98.5, 45.9, 56.9, 87.7

3. Data penjualan komoditas elektronik:

77.3, 121, 101.7, 78.9, 89.7, 104, 90.2, 102.2, 95.7, 133.3, 100, 89.8, 97.4, 102.5, 115, 91.1, 82.5, 104.9, 105.8, 90.3, 108.2, 121.7, 83.2, 97.2, 97.6, 105.2, 98.2, 94.7, 101.7, 104.5, 91.1, 102, 101.5, 111.1, 94, 98.6, 85.4, 94.5, 101.1, 92.5, 95.1, 99.5, 109.4, 100.6, 95.2, 108.9, 102.5, 90.7, 100.1, 100.4, 91.3, 99.7, 96.6, 100.5, 92.6, 82, 88.2, 92.4, 101.1, 83.9, 72.4, 86.2, 90.8, 86.8, 85.2, 87.7, 89.3, 89.8, 88.5, 70.7, 81.4, 70.8, 57.1, 68.9, 77.1, 68.9, 70.2, 83.3, 69.5, 77.7, 86.5, 83.9,

87.3, 68.1, 60.4, 74.3, 85.7, 80.4, 91, 90.6, 80.3, 90.5, 92.6, 104.6,
101.7, 100, 78.8, 95.8, 110.6, 80.3, 103.5, 95.1, 91.7, 96.5, 91.5, 109.6,
104.4, 80.4, 85.5, 122.7

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA

