

PEMODELAN REGRESI MENGGUNAKAN ALGORITMA GENETIKA UNTUK PREDIKSI CURAH HUJAN

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Stevanie Amanda Samuel Putri

NIM: 115090607111034



PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

PEMODELAN REGRESI MENGGUNAKAN ALGORITMA GENETIKA UNTUK PREDIKSI
CURAH HUJAN

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Stevanie Amanda Samuel Putri
NIM : 115090607111034

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D.
NIP:19720919 199702 1 001

Mengetahui
Ketua Program Studi Informatika dan Ilmu Komputer

Drs. Marji, M.T.
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 13 Januari 2016

Stevanie Amanda Samuel Putri

NIM: 115090607111034

KATA PENGANTAR

Puji syukur penulis panjatkan atas berkat rahmat Allah SWT sehingga penulis dapat menyelesaikan skripsi yang berjudul “Pemodelan Regresi Menggunakan Algoritma Genetika” sebagai salah satu persyaratan memperoleh gelar Sarjana Komputer di Program Teknologi Informasi dan Ilmu Komputer (PTI IK), Program Studi Informatika/Illmu Komputer, Universitas Brawijaya Malang.

Atas terselesaikannya skripsi ini, penulis mengucapkan terima kasih kepada Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., selaku dosen pembimbing yang telah banyak memberikan pengarahan, ilmu, dan waktunya kepada penulis, segenap bapak dan ibu dosen yang telah mendidik dengan sangat baik selama menempuh pendidikan di PTI IK Universitas Brawijaya, serta staff dan karyawan PTI IK Universitas Brawijaya yang telah banyak membantu keperluan akademik.

Selain itu penulis juga mengucapkan terimakasih kepada orang tua penulis atas semua bantuan dan dukungan yang begitu besar baik dalam bentuk moril maupun materil, serta sahabat penulis yang selalu memberikan semangat dan bantuan dalam pengerjaan skripsi.

Penulis menyadari bahwa skripsi ini tidak terlepas dari kekurangan dan kesalahan. Oleh karena itu, penulis menerima kritik dan saran yang bersifat membangun untuk pengembangan penelitian selanjutnya. Penulis berharap agar skripsi ini dapat memberikan manfaat bagi semua pihak yang berkepentingan.

Malang, 13 Januari 2016

Penulis

stevanieasp@gmail.com

ABSTRAK

Tengger merupakan sebuah kawasan wisata alam di provinsi Jawa Timur yang dikunjungi oleh jutaan wisatawan setiap tahunnya. Adanya Taman Nasional Bromo Tengger Semeru adalah salah satu penarik minat wisatawan baik wisatawan domestik maupun wisatawan mancanegara. Sama dengan daerah-daerah lain di Indonesia, kawasan Tengger memiliki iklim tropis. Pola pergerakan arah angin di wilayah iklim tropis Indonesia setiap enam bulan sekali membuat adanya dua pergantian musim yaitu musim hujan dan musim kemarau. Pergerakan arah angin ini memiliki peranan yang sangat penting dalam mempengaruhi pola curah hujan. Informasi besarnya curah hujan menjadi sangat bermanfaat bagi wisatawan yang hendak berkunjung maupun bagi penduduk asli Tengger untuk mengantisipasi terjadinya peristiwa ekstrem yaitu banjir akibat curah hujan yang tinggi maupun sebaliknya. Oleh karena itu diperlukan suatu estimasi untuk memperkirakan curah hujan di periode berikutnya. Permasalahan ini akan dimodelkan dengan persamaan regresi linear menggunakan algoritma genetika. Representasi kromosom yang digunakan adalah representasi biner dimana kromosom pada sebuah individu yang bernilai 1 variabelnya digunakan untuk prediksi dan apabila kromosom bernilai 0 maka variabelnya tidak digunakan prediksi. Dari pengujian yang dilakukan, sistem mampu memberikan prediksi terbaik pada jumlah populasi 30, jumlah generasi 10, kombinasi cr 0.6 dan mr 0.4, serta jumlah data 20 dasarian. Algoritma genetika mampu memberikan solusi yang optimal dengan menghasilkan jumlah periode lebih sedikit yang digunakan pada prediksi namun dengan nilai *fitness* yang lebih baik.

Kata kunci : regresi linear, algoritma genetika, curah hujan.



ABSTRACT

As a famous natural tourist destination, Tengger which is located in East Java Indonesia able to attract over millions of tourists every year. The existence of this Bromo Tengger Semeru National Park fascinate peoples around the world to travel there. Just like the other province in Indonesia, Tengger also has a tropical climate. The wind movement patterns in a tropical area causes the area come through two seasons that is dry season and rainy season that changes every six months, it also can affects the rainfall patterns. Rainfall amounts information becomes very important for the visitor, tourists, or local Tengger citizens to anticipating extreme events such a hard rainfall or drought condition. Hence, the existence of forecast system to estimate the rainfall amount on the next period is needed. This research implements genetic algorithms to apply linear regression equation. Binary chromosome representation used to represent the set of solutions to get an optimum result. Chromosome representation using binary number state the uses of variable in an individual. If a chromosome in an individual is zero then it will not be used to predict using linear regression. One cut point crossover method used in crossover reproduction process and the mutation process apply invers method. The selection method to get the set of population is Elitism selection which is bundle the parents and offSprings to one and then sort the values of fitness from higher fitness to lowest fitness value. Based on the test analysis, the best parameter values of genetic algorithms is 30 population size, 10 generations, 20 data amounts, crossover rate is 0.6 and mutation rate is 0.4. This set of parameters gave the best result in rainfall forecasting.

Key Words : Genetic Algorithm, Linear Regression, Rainfall.



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah.....	2
1.6 Sistematika Penulisan	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Curah Hujan	5
2.3 Prediksi.....	6
2.4 Regresi Linear.....	6
2.4.1 Regresi Linear Sederhana.....	7
2.4.2 Regresi Linear Berganda.....	7
2.5 Algoritma Genetika	8
2.5.1 Parameter Algoritma Genetika	9
2.5.2 Proses Algoritma Genetika.....	10
2.5.3 Representasi Kromosom	11
2.5.4 Reproduksi	12
2.5.5 Evaluasi.....	13
2.5.6 Seleksi.....	14
2.5.7 Perhitungan <i>Error</i>	14
BAB 3 METODOLOGI PENELITIAN DAN PERANCANGAN.....	16

3.1 Tahapan Penelitian	16
3.2 Teknik Pengumpulan Data	17
3.3 Algoritma yang Digunakan.....	18
3.4 Analisa Kebutuhan Sistem	18
3.4.1 Deskripsi Umum Sistem	18
3.4.2 Daftar Kebutuhan Sistem	18
3.5 Pengujian Algoritma.....	19
BAB 4 PERANCANGAN.....	20
4.1 Formulasi Permasalahan.....	20
4.1.1 Deskripsi Umum Sistem	20
4.1.2 Data yang Digunakan	20
4.1.3 Contoh Penyelesaian Masalah	20
4.2 Siklus Algoritma Genetika.....	23
4.3 Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika	24
4.3.1 Representasi Kromosom	25
4.3.2 Inisialisasi Populasi Awal.....	25
4.3.3 Reproduksi <i>Crossover</i>	29
4.3.4 Reproduksi Mutasi	30
4.3.5 Evaluasi dan Seleksi.....	31
4.4 Perancangan Uji Coba	35
4.4.1 Rancangan Uji Coba Ukuran Populasi (<i>popSize</i>)	35
4.4.2 Rancangan Uji Coba Jumlah Generasi.....	36
4.4.3 Rancangan Uji Coba Kombinasi <i>cr</i> dan <i>mr</i>	36
4.4.4 Rancangan Uji Coba Jumlah Data yang Digunakan.....	37
4.5 Perancangan <i>User interface</i>	37
4.5.1 Rancangan <i>User interface</i> Halaman Data Curah Hujan	37
4.5.2 Rancangan <i>User interface</i> Halaman <i>Parent</i>	38
4.5.3 Rancangan <i>User interface</i> Halaman Generasi Awal.....	39
4.5.4 Rancangan <i>User interface</i> Halaman Generasi Akhir	39
4.5.5 Rancangan <i>User interface</i> Hasil.....	40
4.5.6 Rancangan <i>User interface</i> Halaman Perbandingan	40
BAB 5 IMPLEMENTASI	42



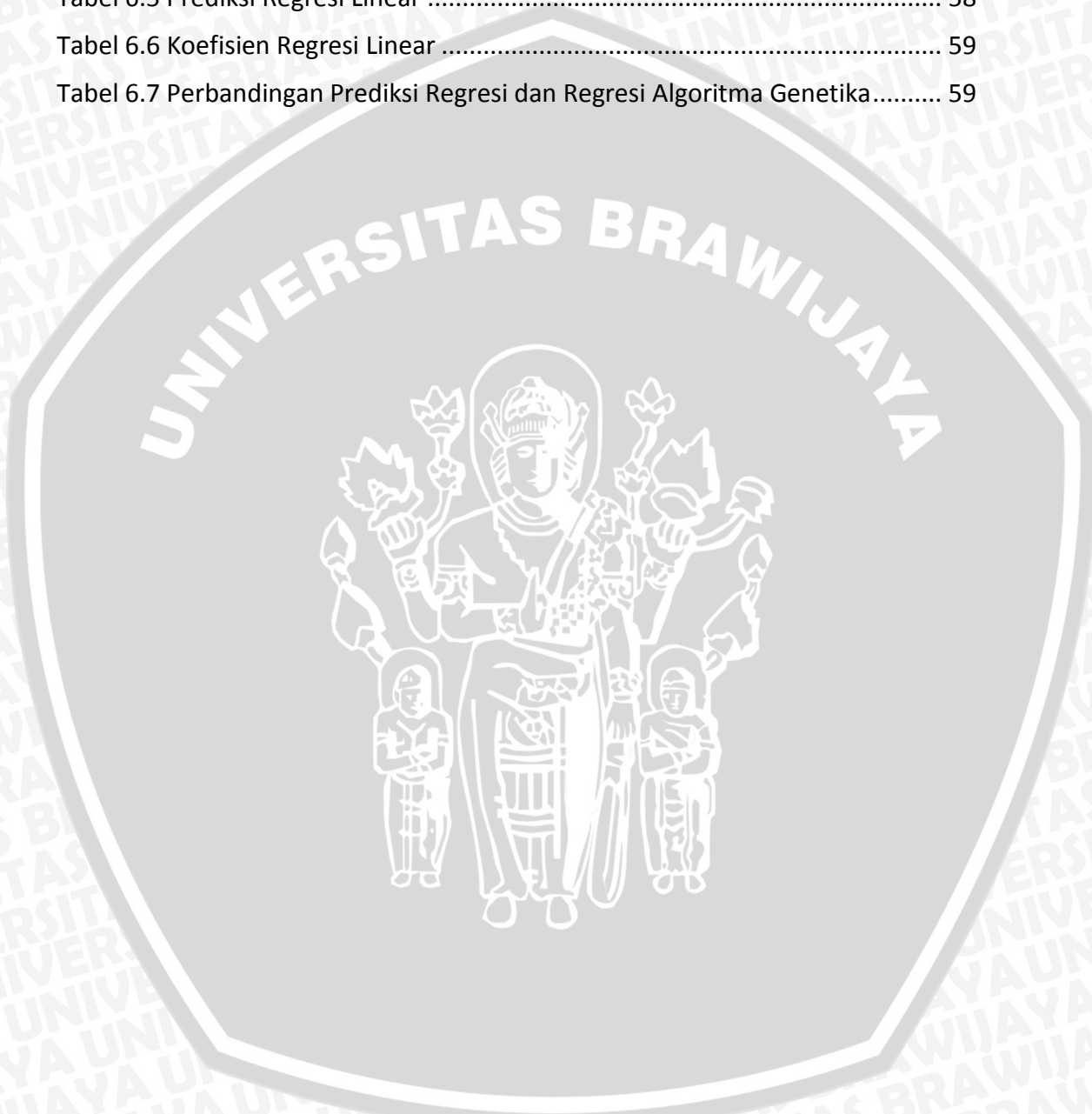
5.1 Implementasi Program	42
5.1.1 Proses Ambil Data	42
5.1.2 Proses Inisialisasi Kromosom Biner	42
5.1.3 Proses Generate Random <i>Parent</i> Awal	43
5.1.4 Proses Perhitungan <i>Error</i> dan <i>Fitness</i>	44
5.1.5 Proses Perhitungan Crossover	45
5.1.6 Proses Perhitungan Mutasi	46
5.1.7 Proses Evaluasi Seleksi	47
5.1.8 Proses Regresi	48
5.2 Implementasi <i>User interface</i>	48
5.2.1 Halaman Data Curah Hujan	48
5.2.2 Halaman Populasi <i>Parent</i>	49
5.2.3 Halaman Generasi Awal	49
5.2.4 Halaman Generasi Akhir	50
5.2.5 Halaman Hasil	51
5.2.6 Halaman Perbandingan	51
BAB 6 PENGUJIAN DAN ANALISIS	53
6.1 Hasil dan Analisa Uji Coba Ukuran Populasi	53
6.2 Hasil dan Analisa Uji Coba Jumlah Generasi	54
6.3 Hasil dan Analisa Uji Coba Kombinasi <i>cr</i> dan <i>mr</i>	55
6.4 Hasil dan Analisa Uji Coba Jumlah Data yang Digunakan	56
6.5 Hasil dan Analisa Perbandingan Hasil Regresi Linear dengan Regresi Menggunakan Algoritma Genetika	57
BAB 7 Penutup	61
7.1 Kesimpulan	61
7.2 Saran	61
DAFTAR PUSTAKA	62



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	4
Tabel 2.2 Contoh Representasi Kromosom Biner	11
Tabel 2.3 Contoh <i>One cut point Crossover</i>	12
Tabel 2.4 Contoh Mutasi	13
Tabel 4.1 Data Curah Hujan Dasarian Stasiun Observasi Tujur 2014	21
Tabel 4.2 Data Curah Hujan 8 Dasarian Sebelumnya	22
Tabel 4.3 Koefisien Regresi Linear	22
Tabel 4.4 Pembentukan Kromosom	25
Tabel 4.5 Representasi Kromosom <i>Parent</i>	26
Tabel 4.6 Data Curah Hujan Untuk Prediksi	26
Tabel 4.7 Proses <i>Fitness P1</i>	27
Tabel 4.8 Koefisien Regresi Individu P1	28
Tabel 4.9 Proses <i>One cut point Crossover</i>	29
Tabel 4.10 <i>OffSpring</i> Hasil Crossover	30
Tabel 4.11 Proses Mutasi Kromosom Biner	30
Tabel 4.12 <i>OffSpring</i> Hasil Mutasi	30
Tabel 4.13 Evaluasi	32
Tabel 4.14 Seleksi	33
Tabel 4.15 Prediksi Regresi Tanpa Algoritma Genetika	34
Tabel 4.16 Rancangan Uji Coba Ukuran Populasi	35
Tabel 4.17 Rancangan Uji Coba Jumlah Generasi	36
Tabel 4.18 Rancangan Uji Coba Kombinasi cr dan mr	36
Tabel 4.19 Rancangan Uji Coba Jumlah Data yang Digunakan	37
Tabel 5.1 Source Code Ambil Data	42
Tabel 5.2 Source Code Inisialisasi Kromosom	43
Tabel 5.3 Source Code Generate Random <i>Parent Awal</i>	43
Tabel 5.4 Source Code <i>Error</i> dan <i>Fitness</i>	44
Tabel 5.5 Source Code Crossover	45
Tabel 5.6 Source Code Mutasi	46
Tabel 5.7 Source Code Evaluasi dan Seleksi	47
Tabel 5.8 Source Code Regresi	48

Tabel 6.1 Hasil Pengujian Ukuran Populasi.....	53
Tabel 6.2 Hasil Pengujian Jumlah Generasi.....	54
Tabel 6.3 Hasil Pengujian Kombinasi Nilai cr dan mr.....	55
Tabel 6.4 Hasil Pengujian Jumlah Data	57
Tabel 6.5 Prediksi Regresi Linear	58
Tabel 6.6 Koefisien Regresi Linear	59
Tabel 6.7 Perbandingan Prediksi Regresi dan Regresi Algoritma Genetika.....	59



DAFTAR GAMBAR

Gambar 2.1 Proses Algoritma Genetika	10
Gambar 2.2 Pseudocode <i>Elitism selection</i>	14
Gambar 3.1 Diagram Alir Penelitian	16
Gambar 4.1 Pseudocode Pemodelan Regresi Algoritma Genetika	24
Gambar 4.2 Diagram Alir Proses Evaluasi	31
Gambar 4.3 Diagram Alir Proses <i>Elitism selection</i>	33
Gambar 4.4 Rancangan <i>User interface</i> Halaman Data Curah Hujan	37
Gambar 4.5 Rancangan <i>User interface</i> Halaman <i>Parent</i>	38
Gambar 4.6 Rancangan <i>User interface</i> Halaman Proses Generasi ke-1	39
Gambar 4.7 Rancangan <i>User interface</i> Halaman Proses Generasi Akhir.....	39
Gambar 4.8 Rancangan <i>User interface</i> Halaman Hasil	40
Gambar 4.9 Rancangan <i>User interface</i> Halaman Perbandingan	40
Gambar 5.1 Implementasi Halaman Data Curah Hujan.....	48
Gambar 5.2 Implementasi Halaman Populasi <i>Parent</i>	49
Gambar 5.3 Implementasi Halaman Generasi Awal	50
Gambar 5.4 Implementasi Halaman Generasi Akhir	51
Gambar 5.5 Implementasi Halaman Hasil	51
Gambar 5.6 Implementasi Halaman Perbandingan.....	52
Gambar 6.1 Grafik Hasil Uji Coba Ukuran Populasi	53
Gambar 6.2 Grafik Hasil Uji Coba Jumlah Generasi	54
Gambar 6.3 Grafik Hasil Uji Coba Kombinasi Nilai c_r dan m_r	56
Gambar 6.4 Grafik Hasil Uji Coba Jumlah Data.....	57
Gambar 6.5 Grafik Perbandingan Hasil Prediksi	60

BAB 1 PENDAHULUAN

1.1 Latar belakang

Tengger merupakan sebuah kawasan wisata alam yang berada di provinsi Jawa Timur yang dikunjungi oleh jutaan wisatawan setiap tahunnya. Adanya Taman Nasional Bromo Tengger Semeru dengan pemandangan alam yang indah adalah salah satu penarik minat wisatawan baik wisatawan domestik maupun wisatawan mancanegara. Sama dengan daerah-daerah lain di Indonesia, kawasan Tengger memiliki iklim tropis. Letak Indonesia yang berada di garis khatulistiwa dan berada di antara benua Asia dan Australia serta Samudra Pasifik dan Samudra Hindia serta berada diantara dua benua mempengaruhi pola pergerakan arah angin di wilayah Indonesia setiap enam bulan sekali sehingga Indonesia mengalami dua pergantian musim yaitu musim hujan dan musim kemarau. Pergerakan arah angin ini memiliki peranan yang sangat penting dalam mempengaruhi pola curah hujan.

Butir-butir air yang jatuh dari awan atau kelompok awan dan mencapai permukaan bumi disebut dengan hujan (Fadholi, 2013). Sedangkan air hujan yang terkumpul pada penakar hujan di tempat yang datar, tidak mengalir dan tidak meresap disebut sebagai curah hujan. Besarnya curah hujan di suatu wilayah tidak dapat ditentukan secara pasti namun dapat diprediksi atau diramalkan. Proses prediksi atau peramalan dapat dilakukan dengan menggunakan data historis/data masa lampau besarnya curah hujan di suatu wilayah untuk memprediksi/meramalkan besarnya curah hujan di wilayah tersebut di masa mendatang.

Informasi besarnya curah hujan menjadi sangat bermanfaat bagi wisatawan yang hendak berkunjung maupun bagi penduduk asli Tengger. Selain itu informasi besarnya curah hujan juga sangat diperlukan untuk mengantisipasi terjadinya peristiwa ekstrem yaitu banjir akibat curah hujan yang tinggi maupun sebaliknya yaitu curah hujan rendah yang menyebabkan kekeringan. Curah hujan yang terlalu tinggi atau terlalu rendah keduanya dapat menyebabkan kerugian secara materi.

Metode-metode yang bisa digunakan untuk memprediksi curah hujan adalah analisis regresi dan jaringan syaraf tiruan. Penelitian menggunakan analisis regresi telah dilakukan oleh Purbaya (2014) yaitu dalam memprediksi berat badan ideal menggunakan analisis regresi linear dengan Fuzzy Mamdani. Pada penelitian ini menunjukkan bahwa metode analisis regresi linear adalah metode terbaik pada penelitian ini dengan nilai MSE sebesar 56,64. Penelitian lainnya dilakukan oleh Apriyanti (2005) yaitu pada optimasi jaringan syaraf tiruan dengan algoritma genetika untuk peramalan hujan. Penelitian tersebut menunjukkan algoritma genetika dapat digunakan sebagai parameter dalam jaringan syaraf tiruan (Apriyanti, 2005).

Penggunaan analisis regresi dalam melakukan prediksi dapat memberikan koefisien-koefisien regresi yang baik, namun tidak dapat menentukan kombinasi-kombinasi variabel yang digunakan untuk memprediksi. Oleh karena itu dibutuhkan sebuah algoritma genetika dalam mendapatkan kombinasi-kombinasi

variabel untuk menghasilkan koefisien regresi yang akan digunakan untuk melakukan prediksi.

Algoritma genetika dapat digunakan untuk membentuk sebuah model regresi yang bisa digunakan untuk memprediksi curah hujan. Penelitian tentang prediksi menggunakan algoritma genetika telah dilakukan sebelumnya yaitu untuk prediksi harga saham [Rahmi, 2015], prediksi konsumsi Kwh beban listrik [Permatasari, 2015] dan lainnya. Pada penelitian prediksi harga saham yang dilakukan oleh Rahmi (2015) menunjukkan bahwa implementasi algoritma genetika mampu menentukan koefisien (kromosom) terbaik yang nantinya menghasilkan hasil prediksi yang mendekati data aslinya. Sedangkan pada penelitian prediksi konsumsi Kwh listrik yang dilakukan oleh Permatasari (2015) menunjukkan bahwa penggunaan metode algoritma genetika memberikan solusi yang lebih baik. Berkaitan dengan uraian sebelumnya maka penulis menentukan judul pada skripsi ini adalah “Pemodelan Regresi Menggunakan Algoritma Genetika Untuk Prediksi Curah Hujan”.

1.2 Rumusan masalah

Berdasarkan uraian latar belakang, maka dapat dirumuskan permasalahan yaitu:

1. Bagaimana pemodelan regresi menggunakan algoritma genetika pada prediksi curah hujan.
2. Bagaimana menentukan nilai parameter algoritma genetika yang tepat.
3. Berapa tingkat error yang dihasilkan pemodelan regresi linear menggunakan metode algoritma genetika untuk prediksi curah hujan.

1.3 Tujuan

Berdasarkan rumusan masalah, tujuan yang ingin dicapai pada penelitian ini adalah:

1. Mengimplementasikan pemodelan regresi menggunakan algoritma genetika pada prediksi curah hujan.
2. Menentukan nilai parameter algoritma genetika yang tepat.
3. Mendapatkan tingkat akurasi yang dihasilkan metode algoritma genetika pada prediksi curah hujan.

1.4 Manfaat

Manfaat yang didapat dari penelitian ini adalah untuk membantu pemerintah setempat maupun sebagai referensi penelitian-penelitian berikutnya mengenai pemodelan regresi yang dibangun dengan algoritma genetika untuk prediksi curah hujan.

1.5 Batasan masalah

Adapun batasan-batasan masalah dalam penelitian ini adalah :

1. Pemodelan regresi yang digunakan pada penelitian ini adalah regresi linear sederhana.
2. Metode yang digunakan untuk memprediksi adalah metode algoritma genetika.
3. Data yang digunakan adalah data curah hujan dasarian yang dikumpulkan pada Penelitian Unggulan Perguruan Tinggi 2015 dengan judul *Sistem Peramalan Iklim dengan GSTAR-SUR Untuk Prediksi Kalender Tanam Kentang*.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan pada penelitian ini adalah :

BAB I PENDAHULUAN

Bab pendahuluan ini berisikan latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan skripsi.

BAB II LANDASAN KEPUSTAKAAN

Menjelaskan mengenai kajian pustaka dan landasan teori yang berhubungan dengan skripsi ini yaitu tentang prediksi, curah hujan, regresi linear, algoritma genetika dan perhitungan *error*.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan mengenai langkah-langkah yang digunakan dalam implementasi metode algoritma genetika dengan pemodelan regresi untuk prediksi curah hujan.

BAB IV PERANCANGAN

Pada bab ini menjelaskan perancangan yang digunakan dalam implementasi metode Algoritma Genetika untuk prediksi curah hujan yang meliputi perancangan *user interface* dan perancangan pengujian.

BAB VI IMPLEMENTASI

Menjelaskan implementasi metode algoritma genetika dengan pemodelan regresi untuk prediksi curah hujan, *user interface* sistem dan *source code* untuk mengembangkan sistem ini.

BAB VI PENGUJIAN DAN ANALISIS

Menjelaskan proses pengujian dan hasil pengujian implementasi metode algoritma genetika dengan pemodelan regresi untuk prediksi curah hujan, serta analisis dari pengujian tersebut.

BAB VII PENUTUP

Bab ini memuat kesimpulan yang diperoleh dari hasil perngujian dan saran untuk pengembang.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Tabel 2.1 Kajian Pustaka

No	Tahun	Judul	Peneliti	Objek	Hasil Penelitian
1	2014	Perbandingan Analisis Regresi Linear Berganda dengan Sistem Inferensi Fuzzy Mamdani dalam Memprediksi Berat Badan Ideal	Rifal Purbaya	Berat Badan	Metode analisis regresi linear adalah metode terbaik pada penelitian ini dengan nilai MSE sebesar 56,64.
2	2005	Optimasi Jaringan Syaraf Tiruan Dengan Algoritma Genetika Untuk Peramalan Curah Hujan	Novi Apriyanti	Curah Hujan	Algoritma Genetika dapat digunakan dalam salah satu parameter dalam Jaringan Syaraf Tiruan.
3	2011	Aplikasi untuk Prediksi Jumlah Mahasiswa Pengambil Mata Kuliah dengan Menggunakan Algoritma Genetika. Studi Kasus : Jurusan Teknik Informatika ITS.	Hafid Hazaki, Joko Lianto Buliali, Anny Yuniarti	Mata Kuliah	Nilai <i>fitness</i> yang dihasilkan dari proses regresi menghasilkan keakuratan yang lebih tinggi dibandingkan data nilai <i>fitness</i> yang tidak melewati proses regresi.

Sumber : (Apriyanti, 2005),(Hazaki, 2011),(Purbaya, 2014).

Kajian pustaka yang digunakan dalam penelitian ini terdiri dari 3 penelitian seperti yang ditunjukkan pada Tabel 2.1. Penelitian pertama dilakukan oleh Purbaya (2014) yang membandingkan analisis regresi linear berganda dengan Sistem Inferensi Fuzzy Mamdani dalam memprediksi berat badan ideal, dimana objek yang digunakan adalah berat badan. Pada penelitian tersebut, peneliti membandingkan hasil MSE yang didapatkan dari hasil prediksi menggunakan metode analisis regresi linear berganda dengan Sistem Inferensi Fuzzy Mamdani. Hasil penelitian tersebut menunjukkan bahwa MSE metode analisis regresi linear

berganda lebih kecil daripada MSE metode Fuzzy Mamdani yaitu 56,54 untuk regresi linear dan 69,45 untuk Fuzzy Mamdani (Purbaya, 2014).

Penelitian kedua dilakukan oleh Apriyanti (2005) yang melakukan optimasi jaringan syaraf tiruan dengan algoritma genetika untuk peramalan curah hujan dimana objek yang digunakan adalah data curah hujan. Pada penelitian ini menunjukkan bahwa algoritma genetika dapat digunakan untuk optimasi salah satu parameter dalam jaringan syaraf tiruan. Hal ini dibuktikan dengan nilai *RMSE* (*Root Mean Square Error*) pada prediksi menggunakan jaringan syaraf tiruan optimasi lebih tinggi daripada nilai *RMSE* menggunakan jaringan syaraf tiruan standar (Apriyanti,2005).

Penelitian ketiga yaitu aplikasi untuk memprediksi jumlah mahasiswa pengambil mata kuliah dengan menggunakan algoritma genetika yang dilakukan oleh Hazaki (2011). Pada penelitian ini objek yang digunakan adalah mata kuliah dengan studi kasus di Jurusan Teknik Informatika ITS. Pada percobaan yang dilakukan pada penelitian ini menunjukkan pengaruh nilai *fitness* yang dihasilkan melalui proses regresi linear menghasilkan keakuratan yang lebih tinggi daripada tanpa melalui proses regresi terlebih dahulu (Hazaki, 2011).

2.2 Curah Hujan

Hujan merupakan salah satu fenomena alam yang sangat dipengaruhi oleh iklim yang keberadaannya sangat penting dalam kehidupan untuk memenuhi kebutuhan air oleh semua makhluk hidup di bumi. Terjadinya hujan disebabkan karena pendinginan suhu udara atau penambahan uap air ke udara. Hujan adalah butir-butir air yang jatuh dari awan atau kelompok awan ke permukaan bumi.

Butiran air yang turun dari awan dengan diameter lebih dari 0,5 mm disebut hujan sedangkan butiran air yang berdiameter 0,5-0,1 mm disebut gerimis (Mukid, 2013). Untuk dapat mencapai permukaan bumi maka butiran air harus memiliki garis tengah lebih dari 200 mikrometer (1 mikrometer = 0,001 cm). Apabila garis tengah kurang dari ukuran tersebut maka butiran air tidak akan mencapai permukaan bumi karena habis saat menguap di atmosfer.

Curah hujan adalah air yang jatuh di permukaan datar selama periode tertentu yang diukur dengan menggunakan alat penakar hujan sehingga dapat diketahui jumlahnya dalam satuan tinggi millimeter (mm). Dalam definisi lain curah hujan merupakan tinggi air hujan yang jatuh di permukaan tanah sebelum meresap ke tanah, mengalir ke permukaan dan menguap. Curah hujan 1 milimeter memiliki arti pada permukaan per satuan luas (m^2) terdapat air hujan setinggi 1 milimeter. Menurut Wilson pada (Mukid,2013) faktor yang mempengaruhi curah hujan adalah kelembaban udara, tekanan udara, temperatur dan kecepatan angin.

Data curah hujan dapat berupa data curah hujan harian, bulanan, tahunan, atau dasarian. Biasanya data curah hujan yang diperoleh oleh stasiun klimatologi dianalisa untuk melihat sebaran yang ada. Analisa ini memiliki tujuan untuk memperkirakan atau menentukan kejadian curah hujan pada masa mendatang berdasarkan data curah hujan sebelumnya. Salah satu manfaat dari mengetahui

data curah hujan di suatu daerah yaitu mengetahui potensi daerah tersebut terhadap bencana alam yang disebabkan oleh faktor hujan.

Data curah hujan dasarian merupakan data curah hujan dengan rentang waktu atau dengan periode sepuluh hari. Oleh karena itu dalam suatu bulan dibagi menjadi tiga dasarian.

1. Dasarian 1 dimulai tanggal 1 hingga tanggal 10
2. Dasarian 2 dimulai tanggal 11 sampai tanggal 20
3. Dasarian 3 dimulai tanggal 21 hingga akhir bulan.

2.3 Prediksi

Prediksi merupakan sebuah proses untuk melakukan perkiraan tentang sesuatu yang mungkin akan terjadi di masa mendatang berdasarkan informasi masa lalu dan sekarang. Terdapat dua metode untuk melakukan prediksi yaitu prediksi kualitatif dan prediksi kuantitatif yang akan dijelaskan berikut (Syafuddin, 2014).

a. Prediksi Kualitatif

Prediksi kualitatif adalah prediksi yang dilakukan berdasarkan data kualitatif masa lalu yang tidak dapat direpresentasikan menjadi suatu angka atau nilai. Hasil prediksi menggunakan teknik prediksi ini sangat bergantung pada pemikiran individu yang menyusunnya. Biasanya pemikiran individu didapatkan dari hasil opini, pengetahuan maupun pengalaman yang dimiliki.

b. Prediksi Kuantitatif

Teknik prediksi kuantitatif merupakan prediksi berdasarkan data kuantitatif masa lalu yang dapat direpresentasikan secara tegas ke dalam bentuk angka atau nilai untuk memprediksikan data di masa mendatang. Penggunaan metode pada prediksi kuantitatif sangat mempengaruhi hasil akhir prediksi karena dengan menggunakan metode yang berbeda akan diperoleh hasil prediksi yang berbeda pula. Selain itu juga mempengaruhi besarnya penyimpangan yang dihasilkan antara data aktual dan data hasil prediksi.

2.4 Regresi Linear

Regresi linear adalah sebuah metode statistika yang digunakan untuk mengetahui bentuk hubungan antara variabel terikat (dependen; respon; Y) dengan satu atau lebih variabel bebas (independen; prediktor; X) (Kurniawan, 2008). Umumnya analisa regresi memiliki manfaat untuk memprediksi atau meramalkan nilai suatu variabel terikat berdasarkan data-data yang ada pada variabel bebas, misalnya meramalkan besarnya curah hujan di masa mendatang berdasarkan data saat ini dan data masa lalu. Manfaat lain dari menggunakan analisa regresi adalah mengetahui hubungan antar variabel yang sedang diteliti dan mengetahui sejauh mana pengaruhnya terhadap variabel lain.

Regresi linear dibagi menjadi dua jenis yaitu regresi linear sederhana dan regresi linear berganda. Pada regresi linear sederhana hanya terdapat satu variabel bebas sedangkan pada regresi linear berganda terdapat lebih dari satu variabel bebas. Pada suatu model regresi terdapat koefisien-koefisien, dimana koefisien pada model regresi adalah nilai duga parameter untuk kondisi sebenarnya.

2.4.1 Regresi Linear Sederhana

Regresi linear sederhana merupakan persamaan dimana satu variabel bebas (independen) dan satu variabel terikat (dependen) yang membentuk suatu model persamaan untuk menunjukkan sebuah hubungan matematis. Variabel terikat (dependen) disimbolkan dengan Y dan variabel bebas disimbolkan dengan X. Dimana variabel X merupakan data actual pada sebuah kasus, sedangkan b merupakan koefisien regresi. Koefisien regresi sendiri adalah jumlah perubahan pada Y yang disebabkan karena perubahan nilai X (Azkiya, 2015).

Persamaan (2-1) berikut adalah persamaan umum persamaan regresi linear sederhana.

$$Y' = a + bX \quad (2-1)$$

Dimana :

Y' = Nilai prediksi variabel terikat (dependen)

X = Variabel bebas (independen)

α = Konstanta

b = Koefisien regresi variabel bebas

2.4.2 Regresi Linear Berganda

Regresi linear berganda merupakan sebuah metode statistika untuk membuat suatu model yang menunjukkan hubungan ketergantungan antara suatu variabel terikat (dependen; Y) dengan lebih dari satu variabel bebas (X_1, X_2, \dots, X_n). Persamaan yang dihasilkan oleh analisis regresi linear berganda dapat digunakan untuk memprediksi nilai suatu variabel Y (respon) berdasarkan beberapa variabel X (prediktor). Pada penelitian ini model regresi yang digunakan adalah regresi linear berganda karena dalam melakukan prediksi curah hujan menggunakan lebih dari satu variabel.

Bentuk umum regresi linear berganda ditunjukkan pada persamaan (2-2) berikut ini.

$$Y' = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_nX_n \quad \dots\dots(2-2)$$

Dimana :

Y' = Variabel terikat

b_0 = Konstanta

$b_1, b_2, b_3, \dots, b_n$ = Parameter koefisien regresi variabel bebas.
 $X_1, X_2, X_3, \dots, X_n$ = Variabel bebas

Misalkan pada model regresi diberikan 2 variabel bebas X_1 dan X_2 maka persamaannya adalah $Y' = b_0 + b_1X_1 + b_2X_2$. Untuk mendapatkan koefisien regresi menggunakan matriks langkahnya adalah :

1. Membentuk matriks A, b, dan g.

$$A = \begin{bmatrix} n & \sum X_{1i} & \sum X_{2i} & \dots & \sum X_{ki} \\ \sum X_{1i} & \sum X_{1i}^2 & \sum X_{1i}X_{2i} & \dots & \sum X_{1i}X_{ki} \\ \dots & \dots & \dots & \dots & \dots \\ \sum X_{ki} & \sum X_{ki}X_{1i} & \sum X_{ki}X_{2i} & \dots & \sum X_{ki}^2 \end{bmatrix}$$

$$b = \begin{bmatrix} b_0 \\ b_1 \\ \dots \\ b_k \end{bmatrix} \quad g = \begin{bmatrix} g_0 = \sum Y_i \\ g_1 = \sum X_{1i}Y_i \\ \dots \\ g_k = \sum X_{ki}Y_i \end{bmatrix}$$

2. Membentuk persamaan normal dalam bentuk $A \cdot b = g$.

$$nb_0 + b_1 \sum X_{1i} + b_2 \sum X_{2i} + \dots + b_k \sum X_{ki} = \sum Y_i$$

$$b_0 \sum X_{1i} + b_1 \sum X_{1i}^2 + b_2 \sum X_{1i}X_{2i} + \dots + b_k \sum X_{1i}X_{ki} = \sum X_{1i}Y_i$$

$$b_0 \sum X_{ki} + b_1 \sum X_{ki}X_{1i} + b_2 \sum X_{ki}X_{2i} + \dots + b_k \sum X_{ki}^2 = \sum X_{ki}Y_i$$

3. Menghitung matriks koefisien b dalam bentuk $b = A^{-1} \cdot g$

$$b_0 = \bar{Y} - b_1\bar{X}_1 - b_2\bar{X}_2$$

$$b_1 = \frac{\left(\sum_{i=1}^n X_2^2\right)\left(\sum_{i=1}^n X_1Y\right) - \left(\sum_{i=1}^n X_1X_2\right)\left(\sum_{i=1}^n X_2Y\right)}{\left(\sum_{i=1}^n X_1^2\right)\left(\sum_{i=1}^n X_2^2\right) - \left(\sum_{i=1}^n X_1X_2\right)^2}$$

$$b_2 = \frac{\left(\sum_{i=1}^n X_1^2\right)\left(\sum_{i=1}^n X_2Y\right) - \left(\sum_{i=1}^n X_1X_2\right)\left(\sum_{i=1}^n X_1Y\right)}{\left(\sum_{i=1}^n X_1^2\right)\left(\sum_{i=1}^n X_2^2\right) - \left(\sum_{i=1}^n X_1X_2\right)^2}$$

2.5 Algoritma Genetika

Algoritma genetika (Genetic Algorithms, GAs) adalah algoritma yang diperkenalkan oleh John Holland pada awal 1970 di Universitas Michigan, New



York. Seiring dengan perkembangan teknologi informasi yang sangat pesat algoritma genetika juga mengalami perkembangan. Hal ini membuat algoritma genetika banyak digunakan dalam menyelesaikan berbagai masalah kompleks pada berbagai bidang seperti pada bidang fisika, biologi, ekonomi, dan masalah optimasi yang memiliki model matematika kompleks atau sulit dibangun.

Algoritma genetika merupakan suatu algoritma pencarian untuk mendapatkan hasil yang paling baik berdasarkan sistem yang natural yaitu proses reproduksi atau perkawinan dan proses seleksi genetika. Pada proses reproduksi kombinasi individu dihasilkan dari himpunan solusi yang dibangkitkan secara acak (*random*). Himpunan solusi ini disebut populasi.

Konsep algoritma genetika mengacu pada teori evolusi Darwin dimana individu yang memiliki kualitas lebih baik pada sebuah populasi lebih memiliki kemampuan untuk bertahan dari seleksi alam dan bertahan hidup (Mahmudy, 2013). Pada sebuah populasi terdapat sejumlah individu yang memiliki peran sebagai induk (*parent*) dan melakukan reproduksi untuk menghasilkan keturunan (*offSpring*). Individu yang memiliki kualitas lebih baik kemungkinan besar akan menghasilkan individu-individu yang lebih baik meskipun nyatanya tidak selalu menghasilkan keturunan yang lebih baik dari induknya.

2.5.1 Parameter Algoritma Genetika

Pada proses kerja algoritma genetika diperlukan beberapa parameter tertentu yang akan mempengaruhi hasil solusi yang diberikan dari permasalahan yang diselesaikan. Beberapa parameter penting tersebut adalah (Mahmudy, 2013).

1. Ukuran Populasi

Ukuran populasi atau biasa disebut dengan *popSize* adalah jumlah kromosom yang terdapat pada sebuah populasi. Semakin besar ukuran sebuah populasi akan membuat pencarian solusi lebih luas tetapi tidak menjamin akan memberikan solusi yang optimal. Apabila ukuran populasi terlalu besar akan membuat proses algoritma genetika menjadi lebih lama. Sebaliknya apabila ukuran populasi terlalu kecil akan membuat pencarian solusi menjadi sempit dan tidak memberikan hasil yang optimal.

2. Jumlah Generasi

Jumlah generasi yang dimaksud adalah jumlah iterasi yang akan dilakukan pada proses evaluasi algoritma genetika. Proses pengujian terhadap jumlah generasi dilakukan beberapa kali agar mendapatkan hasil yang optimal pada penyelesaian suatu masalah.

3. *Crossover rate* (*cr*)

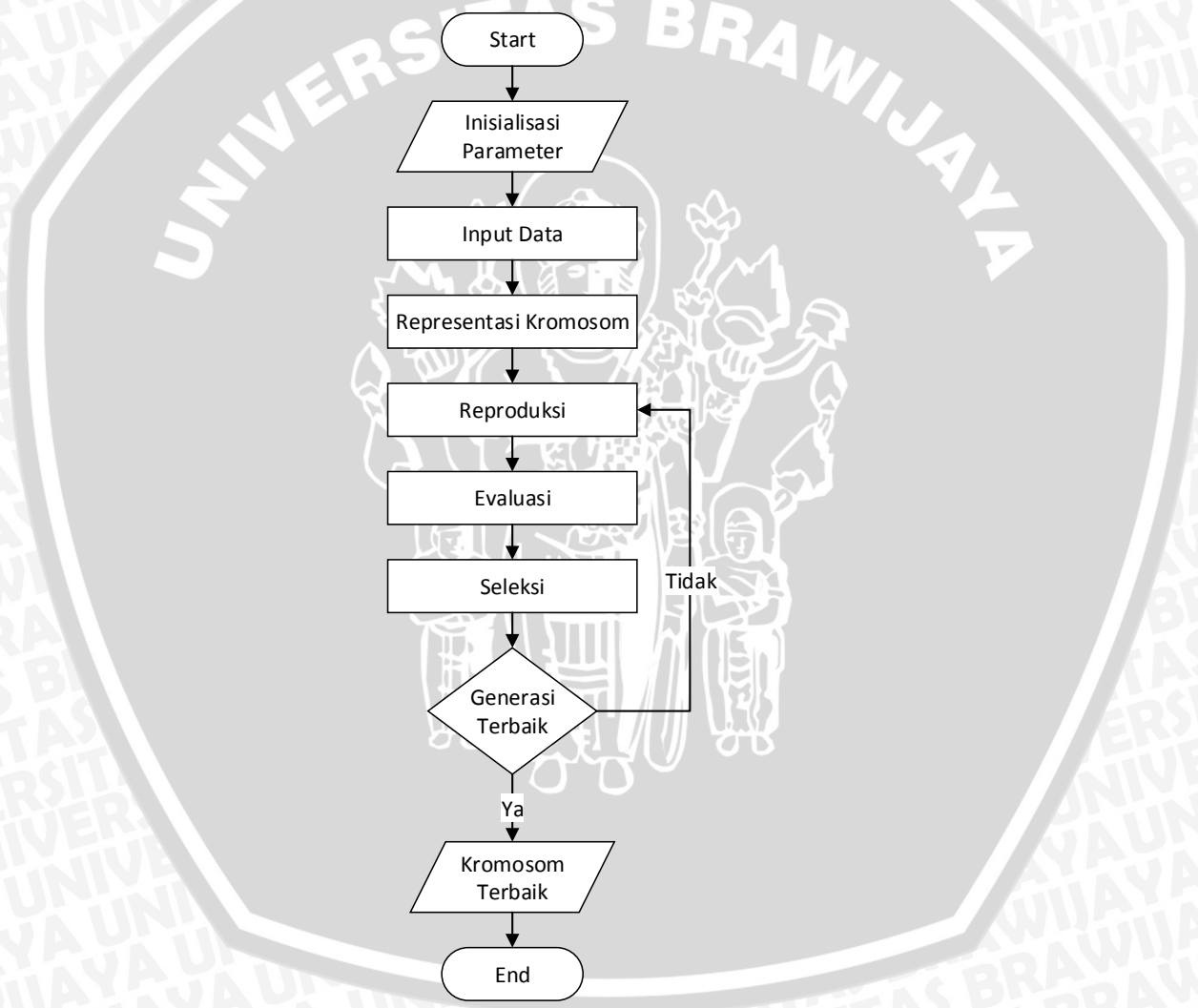
Crossover rate (*cr*) adalah persen total kromosom yang akan melakukan proses *crossover*. *Crossover* sendiri dilakukan untuk mendapatkan kromosom-kromosom baru yang memiliki sifat lebih baik dari kromosom terdahulu. Nilai *cr* berkisar antara 0 hingga 1. Untuk mendapatkan nilai *cr* yang tepat dalam menyelesaikan permasalahan algoritma genetika maka perlu dilakukan beberapa kali pengujian.

4. *Mutation rate* (*mr*)

Mutation rate (mr) digunakan untuk mendapatkan jumlah *offSpring* yang dihasilkan pada proses mutasi. mr sendiri adalah seberapa sering kromosom akan melakukan mutasi dalam proses algoritma genetika. Sama dengan cr , mr biasanya bernilai antara 0 hingga 1. Tidak ada metode khusus dalam menentukan nilai mr oleh karena itu perlu dilakukan beberapa kali pengujian untuk mendapatkan kombinasi yang baik.

Dalam menentukan parameter-parameter algoritma genetika ini tidak terdapat metode yang pasti. Kombinasi parameter yang optimal didapatkan dari serangkaian pengujian yang dilakukan sebelumnya agar mendapatkan solusi yang optimal. Selain itu juga dipengaruhi oleh permasalahan yang sedang diselesaikan.

2.5.2 Proses Algoritma Genetika



Gambar 2.1 Proses Algoritma Genetika

Pada Gambar 2.1 proses awal pada algoritma genetika yaitu inisialisasi parameter dan input data yang digunakan untuk penyelesaian masalah. Parameter yang digunakan pada penyelesaian masalah algoritma genetika

ditentukan secara acak dalam interval tertentu. Proses selanjutnya adalah membuat representasi kromosom yaitu menciptakan individu yang dibangkitkan secara acak yang memiliki susunan gen (*chromosome*) tertentu. Individu-individu tersebut kemudian ditempatkan pada sebuah penampungan yang disebut populasi.

Setelah representasi kromosom dibuat maka proses reproduksi dilakukan yaitu perkawinan antar individu pada populasi untuk menciptakan keturunan (*offspring*). Dengan terciptanya individu baru maka proses reproduksi mengakibatkan jumlah individu pada populasi bertambah. Terdapat dua cara pada proses reproduksi yaitu reproduksi dengan perkawinan silang (*crossover*) dan reproduksi dengan proses mutasi (Mahmudy, 2013).

Proses selanjutnya adalah evaluasi yaitu proses untuk menghitung nilai kebugaran (*fitness*) setiap kromosom baik *offSpring* maupun kromosom induk (*parent*). Semakin besar nilai *fitness* maka semakin baik kromosom tersebut untuk dijadikan calon solusi. Setelah evaluasi proses terakhir adalah seleksi untuk memilih individu dari himpunan populasi dan *offspring* yang terbaik dipertahankan hidup untuk dijadikan *parent* pada generasi selanjutnya. Kromosom dengan nilai *fitness* yang lebih baik memiliki peluang lebih besar untuk terpilih sehingga akan terbentuk generasi yang lebih baik daripada generasi saat ini. Penjelasan lebih detail tentang proses algoritma genetika diijelaskan pada sub-bab berikut ini.

2.5.3 Representasi Kromosom

Representasi kromosom digunakan untuk menyelesaikan suatu masalah menggunakan algoritma genetika yaitu dengan memetakan (*encoding*) solusi dalam masalah tersebut menjadi string kromosom. String kromosom ini tersusun atas sejumlah gen yang menggambarkan variabel-variabel keputusan yang digunakan dalam solusi. Pada representasi kromosom terdapat beberapa bentuk yaitu representasi biner, integer, real code dan permutasi (Mahmudy, 2013). Contoh representasi kromosom biner ditunjukkan pada Tabel 2.2 berikut ini.

Tabel 2.2 Contoh Representasi Kromosom Biner

Parent	Kromosom				
	1	2	3	4	5
P1	1	0	1	1	1

Pada contoh representasi kromosom biner Tabel 2.2, kromosom dibentuk dari pembangkitan nilai random 1 dan 0.

Pembangkitan populasi awal pada algoritma genetika yaitu dengan melakukan pembangkitan sejumlah individu secara acak untuk menjadi solusi sebuah permasalahan. Ukuran populasi (*popSize*) disesuaikan dengan permasalahan yang sedang diselesaikan. Ukuran populasi yang baik adalah yang optimal. Apabila ukuran populasi terlalu kecil maka ruang eksplorasi untuk mendapatkan solusi akan tidak cukup, sedangkan apabila ukuran populasi terlalu besar akan

mengakibatkan proses perhitungan menjadi lebih lama tanpa menjamin akan menghasilkan solusi yang optimal.

2.5.4 Reproduksi

Proses reproduksi merupakan proses yang dilakukan setelah pembangkitan kromosom awal. Reproduksi dilakukan untuk mendapatkan individu baru dari perkawinan antar individu dalam sebuah populasi. Himpunan keturunan yang dihasilkan nantinya akan disimpan pada sebuah penampungan yang disebut *offSpring*. Penjelasan lebih lanjut pada reproduksi perkawinan silang (*crossover*) dan reproduksi dengan proses mutasi (*mutation*) akan dijelaskan pada sub-bab berikut.

2.5.4.1 Crossover

Proses *crossover* dilakukan untuk menambah keanekaragaman individu dalam suatu populasi dengan cara menyilangkan antar string. Pada tahap ini tingkat *crossover* (*crossover rate / cr*) harus ditentukan untuk mengetahui rasio / persen *offSpring* yang dihasilkan dari proses *crossover* dengan cara mengalikan ukuran populasi (*popSize*) dengan nilai *cr*. Sebagai contoh apabila nilai *cr* ditentukan sebesar 0.6 dan ukuran populasi adalah 20 maka akan dihasilkan 12 *OffSpring* dari proses *crossover*.

Metode *crossover* yang digunakan pada penelitian ini adalah *one cut point crossover* untuk menghasilkan *Offspring*. Pada proses *crossover* menggunakan metode *one cut point crossover*, dua induk dipilih secara acak dari populasi kemudian menentukan titik potong yang akan digunakan. Setelah titik potong (*cut point*) ditetapkan maka nilai yang ada pada kromosom/gen setelah titik potong tersebut akan ditukarkan antar *parent* yang melakukan *crossover*. Berikut ini adalah contoh proses *one cut point crossover*.

Tabel 2.3 Contoh One cut point Crossover

Parent	Kromosom									
	1	2	3	4	5	6	7	8	9	10
P6	1	0	0	1	1	1	1	1	0	0
P7	1	0	1	1	0	0	0	1	1	1

Offspring	Kromosom									
	1	2	3	4	5	6	7	8	9	10
C1	1	0	0	1	0	0	0	1	1	1
C2	1	0	1	1	1	1	1	1	0	0

Pada contoh ini induk yang terpilih untuk melakukan *crossover* ke-1 adalah induk P6 dan induk P7 dengan titik potong acak yaitu 4. Setelah dilakukan pemotongan maka nilai pada segmen kromosom setelah titik potong akan ditukar dengan induk lawannya.

2.5.4.2 Mutasi

Proses mutasi dilakukan untuk menghindari terjadinya konvergensi dini dan menggantikan gen yang hilang pada populasi selama proses seleksi sehingga

keberagaman individu dalam populasi tetap terjaga. Jika pada proses reproduksi *crossover* memerlukan dua induk, maka pada reproduksi mutasi hanya memerlukan satu induk saja. Pada tahap ini tingkat mutasi (*mutation rate/mr*) harus ditentukan untuk mengetahui rasio *offSpring* yang dihasilkan dengan cara mengalikan ukuran populasi (*popSize*) dengan nilai *mr*. Biasanya nilai *mr* yaitu pada range 0 mendekati 1. Misalkan nilai *mr* ditentukan sebesar 0,4 dan *popSize* sebesar 20 maka akan dihasilkan 8 *offSpring* dari proses mutasi.

Tabel 2.4 Contoh Mutasi

Parent	P3	1	1	1	1	1	1	1	0	1	0
Offspring	C13	1	1	0	1	1	1	1	0	1	0

Pada contoh reproduksi yang ditunjukkan Tabel 2,4, induk yang terpilih melakukan mutasi adalah induk P3. Random point yang terpilih adalah titik ke-3 sehingga nilai yang ada pada kromosom tersebut nilainya ditukarkan dengan nilai lawannya (*invers*), misalkan pada kromosom tersebut bernilai 0 maka setelah termutasi maka nilainya menjadi 1 dan sebaliknya. Sedangkan untuk kromosom yang tidak mengalami mutasi maka nilai yang dihasilkan pada *offSpring* akan tetap.

2.5.5 Evaluasi

Setelah proses reproduksi selanjutnya adalah melakukan evaluasi. Pada proses evaluasi nilai *fitness* masing-masing individu/kromosom dalam sebuah populasi akan dihitung. *Fitness* adalah suatu ukuran untuk mengukur baik tidaknya solusi yang dinyatakan oleh suatu individu. Dalam algoritma genetika individu yang memiliki nilai *fitness* besar akan menghasilkan solusi terbaik. Pada tahap ini individu dari *parent* dan *offSpring* dikumpulkan menjadi satu untuk kemudian dihitung nilai *fitness* masing-masing kromosom. Fungsi *fitness* yang akan digunakan dalam masalah pemodelan regresi untuk prediksi curah hujan ditunjukkan pada persamaan (2-5).

$$fitness = \frac{10}{\epsilon} + \frac{1}{(\alpha \cdot (n+1))} \tag{2-5}$$

Dimana :

- ϵ = nilai MSE (Mean Square Error) yang dihasilkan
- α = 0,1
- n = jumlah kromosom bernilai 1

Fungsi *fitness* adalah sebuah fungsi untuk mencari nilai *fitness* maksimal yang dihasilkan dan untuk menghindari konvergensi dini. Pada perhitungan nilai *fitness*, konstanta 1 merupakan variabel konstanta yang telah ditentukan untuk menghindari penyebut bernilai 0. Variabel α adalah suatu variabel kontrol agar nilai pembagi yang dihasilkan tidak terlalu besar yang ditentukan sebesar 0.1. Sedangkan nilai n merupakan jumlah kromosom yang digunakan untuk memprediksi pada suatu individu. Dengan adanya jumlah kromosom (n) pada



perhitungan *fitness* maka apabila terdapat dua individu dengan nilai MSE yang sama akan dihasilkan nilai *fitness* yang berbeda.

2.5.6 Seleksi

Proses seleksi adalah proses terakhir pada algoritma genetika yang dilakukan untuk memilih individu dengan nilai *fitness* terbaik dari himpunan populasi dan *offSpring* yang akan dipertahankan hidup pada generasi berikutnya. Nantinya generasi ini akan menjadi *parent* pada generasi selanjutnya. Jumlah individu yang dipertahankan hidup sama dengan jumlah individu pada generasi saat ini, sebagai contoh apabila jumlah individu generasi saat ini adalah 20 maka jumlah individu pada generasi selanjutnya juga 20. Individu-individu yang akan menjadi *parent* pada generasi selanjutnya tidak hanya berasal dari generasi *parent* karena pada proses reproduksi telah dihasilkan *offSpring* yang memiliki kemungkinan memiliki *fitness* tinggi. Semakin besar nilai *fitness* suatu individu pada sebuah populasi maka semakin besar peluangnya untuk dipertahankan hidup.

Beberapa metode yang digunakan dalam seleksi yaitu *roulette wheel*, *binary tournament*, dan *elitism selection*. Metode yang akan digunakan dalam penelitian ini ialah metode *elitism selection*. Metode *elitism selection* memilih *popSize* individu terbaik dari kumpulan individu di populasi (*parent*) dan *offSpring* (Mahmudy, 2013). Individu-individu tersebut diurutkan berdasarkan nilai *fitness* tertinggi hingga *fitness* terendah. Individu-individu dengan nilai *fitness* tertinggi yang terpilih pada proses seleksi akan menjadi induk (*parent*) pada generasi berikutnya.

Namun metode *elitism selection* ini memiliki kelemahan yaitu metode ini tidak memungkinkan bagi individu dengan nilai *fitness* rendah untuk masuk ke populasi berikutnya. Dalam beberapa kasus solusi optimum didapatkan dicapai pada individu dengan *fitness* rendah (Mahmudy, 2013). Pseudocode *elitism selection* ditunjukkan pada Gambar 2.2.

Gambar 2.2 Pseudocode *Elitism selection*

Sumber: Mahmudy (2013)

```
ProcedureElitismSelection
Input
    pop                himpunan individu pada populasi
    popSize            ukuran populasi
    OffSpring          individu anak (offSpring)
Temporary<- Merge (pop, OffSpring)
OrderAscending (Temporary)
pop <- CopyBest (Temporary, popSize)
Output
    pop                himpunan individu baru hasil seleksi
END PROCEDURE
```

2.5.7 Perhitungan Error

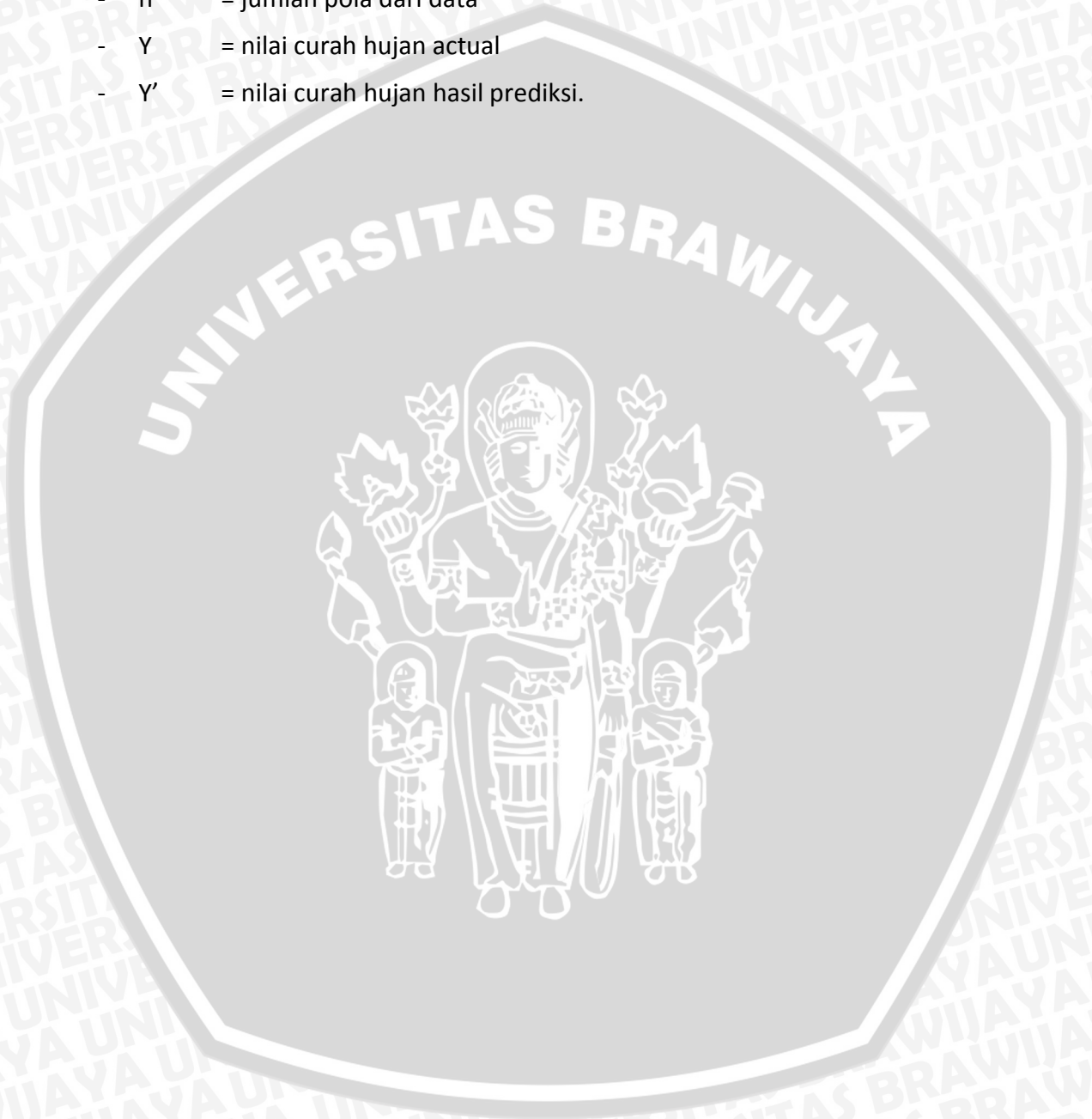
Perhitungan *error* dilakukan untuk mengetahui apakah sebuah metode prediksi mampu digunakan untuk melakukan prediksi dengan baik. Metode perhitungan *error* yang digunakan pada penelitian ini adalah metode *Mean Square Error* (MSE) yaitu sebuah metode analisa untuk membandingkan dan mendapat hasil penyimpangan data hasil prediksi dengan data aktual. Apabila nilai MSE

semakin kecil maka perbedaan antara hasil prediksi dan data aktual semakin kecil. Fungsi persamaan menghitung nilai MSE ditunjukkan pada persamaan (2-6).

$$MSE = \frac{1}{n} \sqrt{\sum_{i=1}^n (Y_i - Y'_i)^2} \quad (2-6)$$

Dimana :

- n = jumlah pola dari data
- Y = nilai curah hujan actual
- Y' = nilai curah hujan hasil prediksi.

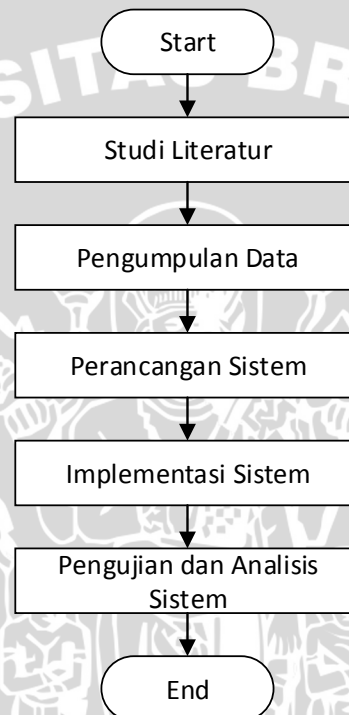


BAB 3 METODOLOGI PENELITIAN DAN PERANCANGAN

Pada bab metodologi penelitian dan perancangan akan dijelaskan tahap penelitian, pengumpulan data, algoritma yang digunakan, analisa kebutuhan sistem, dan pengujian algoritma.

3.1 Tahapan Penelitian

Tahapan pada penelitian ini dimulai dari studi literatur, pengumpulan data, perancangan sistem, implementasi sistem, pengujian dan analisis sistem, dan terakhir adalah kesimpulan dan saran seperti yang ditunjukkan pada Gambar 3.1.



Gambar 3.1Diagram Alir Penelitian

Sumber: Perancangan

1. Studi Literatur

Tahapan utama pada penelitian ini adalah studi literatur yaitu mempelajari dasar-dasar teori dan literatur penelitian-penelitian sebelumnya yang memiliki kaitan dengan penelitian yang akan dilakukan. Sumber literatur dapat berasal dari buku, jurnal, media, internet, hasil penelitian orang lain maupun sumber pustka lainnya yang dapat dipertanggungjawabkan. Adapun literatur pendukung pada penelitian ini yaitu :

1. Curah Hujan.
2. Regresi Linear.
3. Algoritma Genetika.

2. Pengumpulan Data

Tahap kedua pada penelitian ini yaitu pengumpulan data. Data curah hujan yang digunakan diperoleh dari data primer pada Penelitian Unggulan Perguruan Tinggi 2015 dengan judul Sistem Peramalan Iklim dengan GSTAR-SUR Untuk Prediksi Kalender Tanam Kentang dengan tim peneliti yaitu Dr. Ir. Atiek Iriany, MS., Wayan Firdaus Mahmudy, Ph.D., dan Samingun Handoyo, S.Si, M.Cs.

3. Perancangan Sistem

Pada tahap ketiga ini akan dilakukan pembuatan perancangan sistem yang meliputi formulasi permasalahan, siklus algoritma yang digunakan, siklus penyelesaian masalah pemodelan regresi menggunakan algoritma genetika, dan perancangan *user interface*. Pada sub bab formulasi permasalahan akan dijelaskan mengenai deskripsi umum sistem, data yang digunakan, dan contoh permasalahan disertai perhitungannya. Contoh perhitungan manual akan ditampilkan pada sub bab siklus penyelesaian masalah.

4. Implementasi Sistem

Pada tahap ini dilakukan implementasi ke sistem sesuai dengan perancangan yang telah dibuat. Implementasi sistem berupa implementasi lingkungan yang terdiri dari lingkungan *software* dan *hardware*, implementasi algoritma genetika, dan implementasi antarmuka.

5. Pengujian dan Analisis Sistem

Setelah implementasi sistem maka tahap selanjutnya adalah pengujian. Pengujian yang dilakukan sesuai dengan perancangan yang dibuat untuk kemudian dilakukan analisis sistem berdasarkan hasil pengujian yang didapatkan tersebut.

6. Kesimpulan dan Saran

Tahap terakhir pada penelitian ini yaitu proses penarikan kesimpulan dan saran. Kesimpulan yang dibuat merupakan kesimpulan dari hasil penelitian yang dilakukan saat ini yaitu pemodelan regresi menggunakan algoritma genetika untuk prediksi curah hujan. Sedangkan saran dibuat untuk peneliti selanjutnya yang hendak mengembangkan penelitian ini.

3.2 Teknik Pengumpulan Data

Pengumpulan data sampel pada tahap ini dilakukan sebagai acuan untuk menyelesaikan permasalahan penelitian ini yaitu memprediksi curah hujan. Data sampel yang digunakan pada penelitian ini merupakan data curah hujan dengan periode sepuluh harian atau disebut dengan data dasarian yang dikumpulkan pada Penelitian Unggulan Perguruan Tinggi 2015 dengan judul *Sistem Peramalan Iklim dengan GSTAR-SUR Untuk Prediksi Kalender Tanam Kentang*.

3.3 Algoritma yang Digunakan

Pada penelitian ini algoritma yang digunakan adalah algoritma genetika seperti yang telah dijelaskan sebelumnya pada Bab II. Karena berdasarkan penelitian-penelitian lain yang telah dilakukan menunjukkan bahwa penggunaan algoritma genetika dalam prediksi memberikan hasil yang baik. Algoritma yang digunakan diimplementasikan menggunakan bahasa pemrograman Java.

3.4 Analisa Kebutuhan Sistem

Pada tahap ini akan dijelaskan mengenai analisis kebutuhan sistem dan perancangan pada aplikasi prediksi curah hujan menggunakan algoritma genetika. Tahap analisis kebutuhan sistem yang digunakan meliputi deskripsi umum sistem dan daftar kebutuhan sistem.

3.4.1 Deskripsi Umum Sistem

Pada penelitian ini akan dibuat aplikasi yang merupakan implementasi pemodelan regresi linear menggunakan algoritma genetika untuk memprediksi curah hujan. Data curah hujan yang digunakan dalam prediksi ini merupakan data historis curah hujan pada saat ini dan masa sebelumnya. Untuk melakukan proses algoritma pada aplikasi ini maka *user* harus menginputkan nilai dari parameter yang dibutuhkan diantaranya ukuran populasi (*popSize*), nilai *cr*, nilai *mr*, dan jumlah generasi. Besarnya nilai parameter-parameter ini akan berpengaruh pada keoptimalan solusi yang dihasilkan. Setelah proses algoritma dilakukan maka akan menghasilkan estimasi besarnya curah hujan yang mendekati nilai sebenarnya.

3.4.2 Daftar Kebutuhan Sistem

Pada tahap ini menjelaskan hal-hal yang dibutuhkan sistem yang akan dibuat. Berikut ini adalah beberapa hal yang harus dipenuhi untuk menjalankan sistem.

1. Data yang diinputkan oleh *user* berupa parameter awal yaitu :
 - *popSize*
 - *mr*
 - *cr*, dan
 - jumlah generasi.
2. Data curah hujan dasarian pada periode tertentu.

Untuk dapat menjalankan sistem maka lingkungan implementasi yang dibutuhkan adalah.

1. Kebutuhan *Hardware*
 - Laptop Toshiba Core i-3
 - RAM 4 GB
 - DDR3

- VGA 1GB
- 2. Kebutuhan *Software*
 - Sistem operasi yang digunakan yaitu Microsoft Windows 7.0
 - Pengolah dokumentasi dan manual menggunakan Microsoft Office 2013
 - Bahasa pemrograman yang digunakan untuk menjalankan program yaitu Java
 - *Software* untuk membuat aplikasi yaitu NetBeans IDE

3.5 Pengujian Algoritma

Pada sub-bab ini merupakan tahap pengujian terhadap hasil implementasi algoritma genetika yang telah dilakukan sebelumnya. Karena nilai parameter algoritma genetika ditentukan secara random dan tidak terdapat metode khusus maka perlu dilakukan beberapa pengujian pada parameter untuk mendapatkan solusi optimal.

1. Pengujian untuk menentukan jumlah individu pada populasi (*popSize*). Besar *popSize* yang berbeda juga akan memberikan *offSpring* yang berbeda. Apabila ukuran *popSize* makin besar maka *offSpring* yang dihasilkan juga makin banyak.
2. Pengujian untuk mencari banyaknya generasi yang dapat memberikan solusi optimal.
3. Pengujian untuk mendapatkan kombinasi parameter *mutation rate* (*mr*) dan *crossover rate* (*cr*) yang akan menghasilkan solusi optimal. Contoh kombinasi yang digunakan misalnya 0,1 : 0,9, 0,2 : 0,8, 0,7 : 0,3 dan lain-lain. Kombinasi parameter yang berbeda akan menghasilkan *offSpring* yang berbeda pula.
4. Pengujian untuk menentukan jumlah data yang optimal untuk dilakukan prediksi.
5. Pengujian untuk menentukan apakah persamaan regresi baru yang dihasilkan baik digunakan untuk melakukan prediksi. Pada tahap akhir pemodelan regresi menggunakan algoritma genetika akan dihasilkan sebuah persamaan regresi linear ganda yang baru kemudian persamaan ini akan diuji apakah baik atau tidak untuk melakukan prediksi.

BAB 4 PERANCANGAN

4.1 Formulasi Permasalahan

Pada bab formulasi permasalahan dijelaskan mengenai permasalahan yang akan diselesaikan disertai dengan perhitungan manualnya.

4.1.1 Deskripsi Umum Sistem

Data yang diperlukan pada sistem ini yaitu berupa curah hujan dasarian dalam satuan millimeter (mm). Beberapa inputan pada sistem dari *user* juga diperlukan diantaranya nilai *popSize*, jumlah generasi, nilai *cr*, nilai *mr*, dan jumlah data yang akan digunakan dalam proses regresi algoritma genetika. Setelah *user* memasukkan inputan dan data yang dibutuhkan maka sistem akan melakukan pengolahan data kedalam algoritma genetika yang menghasilkan persamaan regresi.

Dalam proses persamaan regresi linear dibutuhkan koefisien regresi yaitu variabel *x* dan variabel *y*. Setelah proses regresi dilakukan selanjutnya yaitu proses algoritma genetika dimulai dari menentukan jumlah gen atau kromosom pada populasi (*popSize*) kemudian dibangkitkan. Setelah gen atau kromosom dibangkitkan maka dilakukan proses reproduksi yakni *crossover* dan mutasi, dimana proses ini memerlukan nilai *cr* dan *mr* yang dimasukkan oleh *user* sebelumnya. Langkah selanjutnya yaitu melakukan iterasi sebanyak jumlah generasi dari masukan *user*.

4.1.2 Data yang Digunakan

1. Data yang digunakan adalah data curah hujan yang diobservasi di daerah Tengger dengan periode sepuluh harian (dasarian). Terdapat 4 lokasi observasi di Tengger yaitu Puspo, Sumber, Tosari dan Tuter. Data dasarian curah hujan yang diperoleh merupakan data periode 2005 hingga 2014.
2. Parameter algoritma genetika yang digunakan yaitu *popSize*, jumlah generasi, *crossover rate* (*cr*), *mutation rate* (*mr*).

4.1.3 Contoh Penyelesaian Masalah

Sebagai contoh, seorang petugas BMKG hendak melakukan prediksi curah hujan di stasiun observasi Tuter dasarian 1 pada Bulan Januari 2015, maka dibutuhkan data dasarian curah hujan periode sebelum-sebelumnya. Untuk melakukan perhitungan manual maka langkah-langkahnya adalah.

1. Menentukan data curah hujan dasarian 8 bulan ke belakang.

Data dasarian yang digunakan pada contoh permasalahan ini adalah data curah hujan tahun 2014 di Stasiun Observasi Tuter yang akan ditampilkan pada Tabel 4.1.

Tabel 4.1 Data Curah Hujan Dasarian Stasiun Observasi Tuter 2014

Stasiun Observasi Tuter Tahun 2014					
Bulan	Dasarian	Curah Hujan	Bulan	Dasarian	Curah Hujan
Januari	1	15.100	Juli	1	0.500
	2	5.200		2	1.900
	3	14.909		3	0.000
Februari	1	14.500	Agustus	1	0.000
	2	1.300		2	0.000
	3	11.375		3	0.000
Maret	1	7.000	September	1	0.000
	2	9.300		2	0.000
	3	9.545		3	0.000
April	1	5.500	Oktober	1	0.000
	2	7.700		2	0.000
	3	0.000		3	0.000
Mei	1	10.800	November	1	3.900
	2	7.600		2	11.600
	3	0.000		3	5.100
Juni	1	7.400	Desember	1	18.600
	2	1.900		2	10.100
	3	0.000		3	21.545

Tabel 4.1 berisi data curah hujan dasarian di stasiun observasi Tuter pada tahun 2014 mulai dari bulan Januari hingga bulan Desember. Dimana setiap bulan dibagi menjadi tiga bagian dasarian. Data dasarian curah hujan tersebut dinyatakan dalam satuan millimeter (mm).

Sebagai contoh untuk melakukan proses perhitungan prediksi curah hujan dasarian 1 di bulan Januari 2015 menggunakan data dari 8 dasarian sebelumnya. Sehingga untuk memprediksi curah hujan dasarian 1 di bulan Januari 2015, data yang digunakan adalah data dasarian 3, 2, 1 bulan Desember, dasarian 3, 2, 1 bulan November, dan dasarian 3, 2 bulan Oktober 2014. Data tersebut kemudian akan diformulasikan pada Tabel 4.2.

Tabel 4.2 Data Curah Hujan 8 Dasarian Sebelumnya

No	Bulan	Dasarian	Y	X1	X2	X3	X4	X5	X6	X7	X8
1	Juni	2	1.90	7.40	0.00	7.60	10.80	0.00	7.70	5.50	9.55
2	Juni	3	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70	5.50
3	Juli	1	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70
4	Juli	2	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00
5	Juli	3	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80
6	Agustus	1	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60
7	Agustus	2	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00
8	Agustus	3	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40
9	September	1	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90
10	September	2	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00
11	September	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50
12	Oktober	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90
13	Oktober	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	Oktober	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	November	1	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	November	2	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	November	3	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00
18	Desember	1	18.60	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00
19	Desember	2	10.10	18.60	5.10	11.60	3.90	0.00	0.00	0.00	0.00
20	Desember	3	21.55	10.10	18.60	5.10	11.60	3.90	0.00	0.00	0.00

2. Persamaan Fungsi Regresi

Dengan menggunakan fitur Data Analysis pada Microsoft Excell dan data yang digunakan adalah Tabel 4.2, pada contoh perhitungan ini Input Y Range yang dimasukkan adalah seluruh data pada kolom Y. Sedangkan Input X Range yang dimasukkan adalah seluruh data nomor 1 hingga nomor 20 pada kolom X1, X2, X3, X4, X5, X6, X7, dan X8. Setelah data input Y Range dan data X Range dimasukkan maka akan dihasilkan koefisien-koefisien yang ditunjukkan Tabel 4.3.

Tabel 4.3 Koefisien Regresi Linear

Coefficients	
Intercept	1.654434
X Variable 1	0.123967
X Variable 2	1.157849
X Variable 3	0.20948
X Variable 4	-0.176908
X Variable 5	-0.639807
X Variable 6	-0.066548
X Variable 7	0.019195
X Variable 8	-0.02447

Tabel 4.3 merupakan tabel yang berisi koefisien regresi yang dihasilkan menggunakan data pada Tabel 4.2. Nilai pada kolom "Intercept" sebesar 1.6544 merupakan koefisien regresi α , kolom "X Variable 1" sebesar 0.1239 adalah koefisien regresi b_1 , kolom "X Variable 2" sebesar 1.1578 adalah koefisien regresi b_2 , kolom "X Variable 3" sebesar 0.2094 adalah koefisien regresi b_3 , kolom "X

Variable 4" sebesar -0.1769 adalah koefisien regresi b_4 , kolom "X Variable 5" sebesar -0.6398 adalah koefisien regresi b_5 , kolom "X Variable 6" sebesar -0.0665 adalah koefisien regresi b_6 , kolom "X Variable 7" sebesar 0.0191 adalah koefisien regresi b_7 dan kolom "X Variable 8" sebesar -0.0244 adalah koefisien regresi b_8 . Dari koefisien regresi yang telah didapatkan maka dibuat sebuah persamaan regresi linear yang ditunjukkan pada persamaan (4-1).

$$Y' = 1.6544 + 0.1239 X_1 + 1.1578 X_2 + 0.2094 X_3 - 0.1769 X_4 - 0.6398 X_5 - 0.0665 X_6 + 0.0191 X_7 - 0.0244 X_8 \quad \dots(4-1)$$

Persamaan ini kemudian digunakan untuk menghitung prediksi curah hujan dasarian 3 Bulan Desember 2015. Berikut ini merupakan contoh perhitungan prediksi menggunakan persamaan regresi yang koefisiennya telah didapatkan dari data dasarian 8 periode sebelumnya.

$$\begin{aligned} Y' &= 1.6544 + 0.1239 (10.10) + 1.1578 (18.60) + 0.2094 (5.10) - \\ & 0.1769 (11.60) - 0.6398 (3.90) - 0.0665 (0.00) + 0.0191(0.00) - \\ & 0.0244 (0.00) \\ &= 20.96 \end{aligned}$$

3. Menghitung Error

Setelah melakukan perhitungan prediksi berdasarkan persamaan regresi maka dihitung nilai *error* dengan menggunakan rumus MSE yang telah dijelaskan sebelumnya. Berikut ini merupakan contoh perhitungan MSE.

$$\begin{aligned} \varepsilon &= \frac{1}{n} \sqrt{\sum_{i=1}^n (Y_i - Y'_i)^2} \\ &= \frac{1}{20} \{(1,90 - 1,61)^2 + (0 - 2,22)^2 + (0,50 + 0,37)^2 + (1,90 - 0,51)^2 + (0 + 2,72)^2 + \\ & (0 - 2,07)^2 + (0 - 1,98)^2 + (0 - 0,85)^2 + (0 - 0,36)^2\} + (0 - 1,54)^2 + (0 - \\ & 1,68)^2 + (0 - 1,61)^2 + (0 - 1,65)^2 + (0 - 1,65)^2 + (3,9 - 1,65)^2 + (11,6 - \\ & 2,14)^2 + (5,1 - 7,61)^2 + (18,6 - 16,53)^2 + (10,10 - 11,61)^2 + (21,55 - 20,96)^2 \\ &= \frac{1}{20} \sqrt{145,10} \\ &= 0,60 \end{aligned}$$

Dari perhitungan persamaan regresi, nilai prediksi curah hujan dasarian 3 Bulan Desember 2014 di Stasiun Observasi Tuter adalah 20,96. Setelah semua data dilakukan perhitungan regresi dan diketahui nilai *error* masing-masing gen kemudian didapatkan total *error*. Setelah itu nilai MSE (*Mean Square Error*) dihitung seperti pada persamaan. Total *error* diakarkan kemudian dibagi jumlah data dan didapatkan nilai MSE 0,60.

4.2 Siklus Algoritma Genetika

Siklus pemodelan regresi dengan algoritma genetika digambarkan pada pseudocode Gambar 4.1.

```

Procedure Regresi Algoritma Genetika
begin
  t= 0
  inisialisasi parameter
  inisialisasi kromosom P (t)
  pemodelan regresi
  while (not termination condition) do
    reproduksi crossover C(t) dari P(t)
    reproduksi mutasi C(t) dari P(t)
    evaluasi fitness C(t) dan P(t)
    seleksi elitism P (t+1) dari P(t) dan C(t)
  t = t+1
  end while
persamaan regresi baru
end

```

Gambar 4.1 Pseudocode Pemodelan Regresi Algoritma Genetika

Sumber: Perancangan

Pada pseudocode regresi algoritma genetika yang ditunjukkan Gambar 4.3, data curah hujan dasarian digunakan sebagai masukan (*input*) pada sistem prediksi curah hujan. Proses pertama yaitu melakukan inisialisasi parameter awal yakni *crossover rate* (*cr*), *mutation rate* (*mr*), jumlah generasi, dan ukuran populasi (*popSize*). Pada tahap ini representasi kromosom dilakukan. Setelah melakukan inisialisasi langkah selanjutnya adalah inisialisasi populasi sejumlah populasi yang sudah ditentukan sebelumnya. Setelah populasi kromosom biner dibangkitkan maka data curah hujan digunakan sebagai input dalam mencari koefisien-koefisien regresi. Proses reproduksi dengan proses *crossover* sebanyak jumlah generasi yang ditentukan berdasarkan nilai *cr* untuk membuat populasi baru. Setelah reproduksi *crossover* proses selanjutnya adalah reproduksi dengan mutasi.

Setelah melalui proses reproduksi, masing-masing kromosom dihitung nilai *fitness*nya untuk kemudian dilakukan evaluasi pada individu awal (*parent*) dan individu *offSpring*. Melakukan proses seleksi dengan menggunakan metode *elitism selection* untuk menentukan individu yang akan menjadi *parent* pada generasi berikutnya. Pada akhir proses algoritma genetika, kromosom terbaik pada suatu generasi dan persamaan regresi linear yang baru akan dihasilkan.

4.3 Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika

Dalam penyelesaiannya algoritma genetika memiliki beberapa proses diantaranya representasi kromosom, proses reproduksi *crossover* dan mutasi, evaluasi dan seleksi. Perhitungan manual dari setiap proses akan dijelaskan pada sub-bab berikut.

4.3.1 Representasi Kromosom

Pada penelitian ini representasi kromosom merupakan representasi biner yang menyatakan periode untuk prediksi curah hujan. Representasi biner hanya bernilai 1 dan 0. Panjang kromosom yaitu sesuai banyaknya variabel yang terlibat dalam perhitungan, yaitu jumlah periode yang diambil kebelakang untuk melakukan prediksi curah hujan di periode selanjutnya. Pada permasalahan ini panjang kromosom yang dibuat pada satu individu adalah 10. Jumlah individu dibuat sejumlah *popSize* yang telah ditentukan pada inialisasi parameter. Berikut ini merupakan contoh pembentukan kromosom yang ditampilkan pada Tabel 4.4.

Tabel 4.4 Pembentukan Kromosom

Individu	Kromosom									
P(t)	1	2	3	4	5	6	7	8	9	10
P1	0	1	0	1	1	0	1	1	1	1

Tabel 4.4 adalah contoh representasi kromosom individu dengan panjang kromosom 10. Kromosom ke-2, 4, 5, 7, 8, 9, 10 memiliki kromosom bernilai 1 yang artinya kromosom tersebut digunakan sebagai variabel dalam prediksi dan perhitungan *fitness* individu tersebut. Sedangkan kromosom lainnya yaitu kromosom ke-1, 3, dan 6 yang bernilai 0 tidak digunakan dalam prediksi dan tidak berperan dalam perhitungan *fitness*.

4.3.2 Inialisasi Populasi Awal

Pada tahap inialisasi dilakukan pembuatan populasi awal sebesar jumlah populasi (*popSize*) yang sudah ditentukan pada inialisasi parameter yaitu 15. Pada permasalahan representasi kromosom yang digunakan adalah representasi biner. Representasi biner hanya berisi nilai 1 dan 0 dimana nilai 1 menyatakan kromosom / gen digunakan untuk memprediksi dan nilai 0 berarti kromosom tersebut tidak digunakan untuk prediksi regresi linear. Sebelum membangkitkan kromosom, langkah awal yang dilakukan adalah inialisasi parameter yaitu :

- *popSize* = 15
- *cr* = 0,5
- *mr* = 0,5
- generasi = 1

Tabel 4.5 Representasi Kromosom Parent

Individu	Kromosom										
	P(t)	1	2	3	4	5	6	7	8	9	10
P1	0	1	0	1	1	0	1	1	1	1	1
P2	1	0	0	0	0	1	0	1	0	1	1
P3	0	0	0	1	0	1	1	0	0	0	0
P4	0	1	0	0	0	0	1	1	1	1	0
P5	1	0	1	0	1	0	0	0	0	1	1
P6	1	1	0	0	0	0	1	1	1	1	0
P7	0	0	1	0	1	1	1	0	1	1	1
P8	1	1	0	0	0	0	1	0	0	0	0
P9	0	1	0	0	0	0	1	0	1	0	0
P10	1	1	1	0	0	0	1	1	1	1	1
P11	1	1	1	0	1	1	1	1	0	0	0
P12	1	1	1	0	1	1	0	0	0	0	1
P13	0	1	0	1	1	1	0	1	1	1	1
P14	1	1	1	1	0	0	0	0	0	0	0
P15	1	0	1	0	1	1	0	1	1	0	0

Tabel 4.5 merupakan representasi kromosom awal yang dibangkitkan pada ukuran populasi (*popSize*) 15 secara random. Individu-individu ini memiliki susunan gen/kromosom yang mewakili solusi dari permasalahan.

Tabel 4.6 Data Curah Hujan Untuk Prediksi

No	Bulan	Dasarian	Y	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
1	Juni	2	1.90	7.40	0.00	7.60	10.80	0.00	7.70	5.50	9.55	9.30	7.00
2	Juni	3	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70	5.50	9.55	9.30
3	Juli	1	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70	5.50	9.55
4	Juli	2	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70	5.50
5	Juli	3	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70
6	Agustus	1	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00
7	Agustus	2	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80
8	Agustus	3	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60
9	September	1	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00
10	September	2	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40
11	September	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90
12	Oktober	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00
13	Oktober	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50
14	Oktober	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90
15	November	1	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	November	2	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	November	3	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	Desember	1	18.60	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	Desember	2	10.10	18.60	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00
20	Desember	3	21.55	10.10	18.60	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00

Tabel 4.6 merupakan tabel data sepanjang 10 kromosom yang digunakan untuk memprediksi besarnya curah hujan dan mendapatkan nilai *fitness*.

Langkah awal sebelum mendapatkan nilai *fitness* P1 adalah mencari nilai *Y'* yang merupakan nilai hasil prediksi curah hujan dasarian dan nilai *error* dari hasil prediksi tersebut. Karena jumlah kromosom yang bernilai 1 pada individu P1



adalah 7, maka data yang digunakan hanyalah data yang kromosomnya bernilai 1 tersebut diantaranya X2, X4, X5, X7, X8, X9, dan X10 seperti yang ditampilkan pada Tabel 4.7.

Tabel 4.7 Proses Fitness P1

No	Bulan	Dasarian	Y	X2	X4	X5	X7	X8	X9	X10	Y'	Error
1	Juni	2	1.90	0.00	10.80	0.00	5.50	9.55	9.30	7.00	0.53	1.8677
2	Juni	3	0.00	7.40	7.60	10.80	7.70	5.50	9.55	9.30	1.90	3.6059
3	Juli	1	0.50	1.90	0.00	7.60	0.00	7.70	5.50	9.55	-2.37	8.2085
4	Juli	2	1.90	0.00	7.40	0.00	10.80	0.00	7.70	5.50	1.12	0.6035
5	Juli	3	0.00	0.50	1.90	7.40	7.60	10.80	0.00	7.70	-0.62	0.3886
6	Agustus	1	0.00	1.90	0.00	1.90	0.00	7.60	10.80	0.00	1.35	1.8256
7	Agustus	2	0.00	0.00	0.50	0.00	7.40	0.00	7.60	10.80	-0.48	0.2342
8	Agustus	3	0.00	0.00	1.90	0.50	1.90	7.40	0.00	7.60	2.00	3.9832
9	September	1	0.00	0.00	0.00	1.90	0.00	1.90	7.40	0.00	-0.29	0.0820
10	September	2	0.00	0.00	0.00	0.00	0.50	0.00	1.90	7.40	0.97	0.9318
11	September	3	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90	2.80	7.8649
12	Oktober	1	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00	2.89	8.3750
13	Oktober	2	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50	2.23	4.9951
14	Oktober	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90	2.52	6.3723
15	November	1	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.89	1.0173
16	November	2	11.60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.89	75.8396
17	November	3	5.10	3.90	0.00	0.00	0.00	0.00	0.00	0.00	7.39	5.2257
18	Desember	1	18.60	11.60	0.00	0.00	0.00	0.00	0.00	0.00	16.26	5.4763
19	Desember	2	10.10	5.10	3.90	0.00	0.00	0.00	0.00	0.00	8.87	1.5084
20	Desember	3	21.55	18.60	11.60	3.90	0.00	0.00	0.00	0.00	22.28	0.5370
											jumlah	138.9425
											akar	11.7874
											MSE	0.5894
											fitness	18.2173

Berdasarkan Tabel 4.7, untuk menghitung nilai prediksi curah hujan dan nilai *fitness* dasarian ke-3 Desember 2014 (No.20) oleh individu P1 maka data yang digunakan diantaranya yaitu :

- X2 = dasarian ke-2 Desember 2014
- X4 = dasarian ke-3 November 2014
- X5 = dasarian ke-2 November 2014
- X7 = dasarian ke-3 Oktober 2014
- X8 = dasarian ke-2 Oktober 2014
- X9 = dasarian ke-1 Oktober 2014
- X10 = dasarian ke-3 September 2014.

Perhitungan Y' menggunakan persamaan regresi $Y' = \alpha + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_nX_n$. Nilai a, b₁, b₂, b₃, b₄, b₅, b₆ dan b₇ individu P1 yang didapatkan dari perhitungan manual data analisis Microsoft Excell adalah :

Tabel 4.8 Koefisien Regresi Individu P1

Coefficients	
Intercept	2.8914
X Variable 1	1.1525
X Variable 2	0.0264
X Variable 3	-0.6038
X Variable 4	0.1267
X Variable 5	0.0789
X Variable 6	-0.2947
X Variable 7	-0.1932

Pada Tabel 4.8 koefisien regresi *Intercept* merupakan koefisien regresi α , kolom “X Variable 1” adalah koefisien regresi b_1 , kolom “X Variable 2” adalah koefisien regresi b_2 , dan kolom “X Variable 7” adalah koefisien regresi b_7 . Untuk menghitung nilai Y' maka dibuat sebuah persamaan dari koefisien regresi Tabel 4.8.

$$Y' = \alpha + b_1X_2 + b_2X_4 + b_3X_5 + b_4X_7 + b_5X_8 + b_6X_9 + b_7X_{10}$$

$$Y' = 2.8914 + (1.1525) 18.6 + (0.0264) 11.6 + (-0.6038) 3.9 + (0.1267) 0.00 + (0.0789) 0.00 + (-0.2947) 0.00 + (-0.1932) 0.00$$

$$Y' = 22.28$$

Setelah nilai Y' didapatkan maka nilai *error* dihitung.

$$error = (Y_i - Y'_i)^2$$

$$= (21.55 - 22.88)^2$$

$$= 0.5370$$

Setiap baris data pada Tabel 4.6 mulai dari dasarian ke-2 Bulan Juni hingga data dasarian ke-3 Bulan Desember 2014 dilakukan prediksi dan akan mendapatkan hasil prediksi beserta nilai *error*nya. Pada dasarian ke-2 Bulan Juni dengan nilai $Y' = 0.53$, dan $Y=1.90$, nilai *error* yang didapatkan adalah 1.8677 dengan cara mengkuadratkan selisih nilai asli (Y) dan nilai hasil prediksi (Y'). Setelah semua nilai prediksi dan *error* didapatkan maka diperoleh jumlah *error* sebesar 138.9425. Nilai *Mean Square Error* (MSE) didapatkan menggunakan rumus pada persamaan (2-6) sebesar 0.5894.

$$MSE = \frac{1}{n} \sqrt{\sum_{i=1}^n (Y_i - Y'_i)^2}$$

$$= \frac{1}{20} \sqrt{138.9425}$$

$$= 0.5894$$

Langkah terakhir adalah mengitung *fitness* individu P1 menggunakan nilai MSE dan jumlah kromosom pada P1 yang sudah diperoleh dari proses sebelumnya.

$$fitness = \frac{10}{\varepsilon} + \frac{1}{(\alpha * (n + 1))}$$

$$= \frac{10}{0.5894} + \frac{1}{(0.1 * (7 + 1))}$$

$$= 16.967 + 1.25$$

fitness = 18.2173

Dari perhitungan *fitness* menggunakan persamaan (2-5) didapatkan nilai *fitness* individu P1 yang memiliki panjang kromosom 7 sebesar 18.2173.

4.3.3 Reproduksi Crossover

Proses reproduksi dengan metode *crossover* dilakukan untuk menghasilkan *offSpring* yang bertujuan mempertahankan keberagaman individu. Jumlah *offSpring* yang dihasilkan pada proses *crossover* dihitung berdasarkan perkalian antara *crossover rate* (*cr*) dan ukuran populasi (*popSize*) yang sudah ditentukan saat inialisasi awal. Metode *crossover* yang digunakan pada representasi biner penelitian ini adalah metode *one cut point crossover*.

Pada contoh persoalan ini nilai *cr* adalah 0,5 dan ukuran *popSize* adalah 15 sehingga jumlah *offSpring* yang akan dihasilkan dari proses *crossover* adalah 8 *offSpring*. Proses *crossover* menggunakan metode *one cut point crossover* yaitu memilih satu titik potong secara acak pada individu *parent* kemudian menukar nilai setelah titik potong tersebut antar *parent*. Titik potong pada setiap pasang induk adalah random dan tidak selalu sama. Tabel 4.8 merupakan proses *crossover* pada populasi awal.

Tabel 4.9 Proses One cut point Crossover

Crossover ke-1, Parent Terpilih : P6 &P7, cut point 4										
P6	1	1	0	0	0	0	1	1	1	0
P7	0	0	1	0	1	1	1	0	1	1
C1	1	1	0	0	1	1	1	0	1	1
C2	0	0	1	0	0	0	1	1	1	0

Pada Tabel 4.9 proses *crossover* terjadi antara dua *parent* yaitu individu P6 dan individu P7 yang dipilih secara random. Individu yang sudah terpilih untuk melakukan reproduksi tidak dapat terpilih lagi. Pada proses *crossover* titik potong (*cut point*) yang terpilih secara random adalah titik 4, sehingga nilai kromosom yang berada setelah titik ke-4 akan mengalami pindah silang. Kromosom setelah titik ke-4 pada individu P6 akan dipindahkan ke titik ke-5 individu *offSpring* C2 dan kromosom setelah titik ke-4 individu P7 akan dipindahkan ke titik ke-5 *offSpring* C1.

Proses *crossover* ini dilakukan sebanyak 4 kali sesuai hasil *offSpring* sehingga menghasilkan individu-individu *offSpring* C1, C2, C3, C4, C5, C6, C7. Dan C8 pada Tabel 4.10.



Tabel 4.10 OffSpring Hasil Crossover

Individu	Kromosom									
C(t)	1	2	3	4	5	6	7	8	9	10
C1	1	1	0	0	1	1	1	0	1	1
C2	0	0	1	0	0	0	1	1	1	0
C3	0	1	0	0	0	0	1	0	0	0
C4	1	1	0	0	0	0	1	1	1	0
C5	1	0	0	0	0	1	0	1	1	0
C6	1	0	1	0	1	1	0	1	0	1
C7	0	1	1	0	1	1	0	0	0	1
C8	1	1	0	1	1	0	1	1	1	1

4.3.4 Reproduksi Mutasi

Sama halnya dengan *crossover*, proses reproduksi mutasi juga bertujuan untuk menghasilkan *offSpring*. Banyaknya *offSpring* yang dihasilkan dari proses mutasi ditentukan oleh besarnya hasil perkalian antara *popSize* dan *mutation rate* (*mr*). Mutasi dilakukan dengan memilih satu induk secara acak dari populasi dengan menukarkan nilai dengan nilai invers pada gen yang terpilih.

Pada contoh persoalan ini *mutation rate* (*mr*) yang digunakan adalah 0,5 dan jumlah populasi adalah 15 sehingga *offSpring* yang dihasilkan dari proses mutasi adalah sebanyak 7 *offSpring*.

Pada representasi kromosom biner, mutasi dilakukan dengan mengganti nilai biner pada kromosom yang terpilih dengan nilai inversnya. Misalkan pada kromosom yang terpilih nilai biner nya adalah 1 kemudian diubah menjadi lawannya yaitu 0. Contoh proses mutasi kromosom ditampilkan pada Tabel 4.10 berikut. Pada mutasi pertama, *parent* yang terpilih untuk melakukan mutasi adalah individu P5 dengan titik mutasi yang terpilih adalah titik ke-3 sehingga dihasilkan individu *offSpring* C9.

Tabel 4.11 Proses Mutasi Kromosom Biner

Mutasi ke-1, point: 3										
P5	1	0	1	0	1	0	0	0	1	1
C9	1	0	0	0	1	0	0	0	1	1

Mutasi dilakukan sebanyak 7 kali sesuai dengan jumlah *offSpring* yang dihasilkan berdasarkan nilai *mr* dan *popSize*. Seluruh individu hasil mutasi akan ditampilkan pada Tabel 4.11.

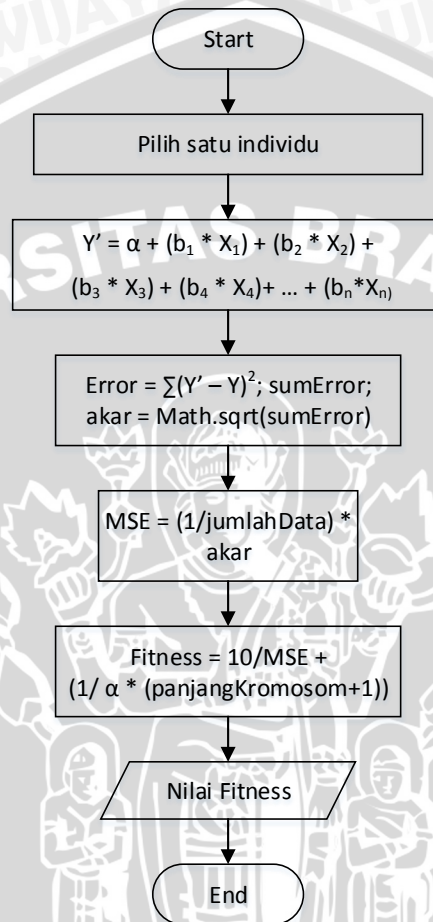
Tabel 4.12 OffSpring Hasil Mutasi

Individu	Kromosom									
C(t)	1	2	3	4	5	6	7	8	9	10
C9	1	0	0	0	1	0	0	0	1	1
C10	0	1	0	1	0	0	1	0	1	0
C11	1	1	1	0	0	1	1	1	1	1
C12	1	0	1	0	1	1	1	1	0	0
C13	0	1	1	0	1	1	0	0	0	1
C14	0	1	0	1	1	1	1	1	1	1
C15	1	1	1	1	0	0	0	1	0	0



4.3.5 Evaluasi dan Seleksi

Dari setiap individu yang terbentuk yaitu individu *parent* dan *offSpring* akan dilakukan proses evaluasi yaitu menghitung kebugaran (*fitness*) tiap individu. Setelah nilai *fitness* semua individu didapatkan maka populasi *parent* dan populasi *offSpring* digabung menjadi satu untuk proses selanjutnya yaitu seleksi. Gambar 4.2 merupakan Gambar Diagram Alir Proses Evaluasi.



Gambar 4.2 Diagram Alir Proses Evaluasi

Sumber: Perancangan

Proses evaluasi menghitung nilai *fitness* dimulai dengan memilih satu individu pada populasi. Langkah kedua yaitu menghitung nilai Y' yaitu nilai prediksi data curah hujan. Nilai Y' akan digunakan untuk mencari nilai *error* prediksi dengan cara mengkuadratkan hasil selisih nilai Y' dan nilai data curah hujan asli periode itu (Y). Setelah semua individu pada populasi dihitung nilai Y' dan nilai *error*nya maka nilai *error* seluruh individu pada populasi akan dijumlah. Jumlah *error* tersebut lalu dicari nilai akarnya. Selanjutnya menghitung nilai MSE (*Mean Square Error*) dengan rumus satu dibagi jumlah individu pada populasi (jumlahData) yaitu 20 dikalikan nilai akar jumlah *error*. Langkah terakhir adalah menghitung *fitness* dengan cara 10 dibagi nilai MSE ditambah 1 dibagi α dikalikan panjang kromosom yang digunakan pada populasi tersebut ditambah 1.

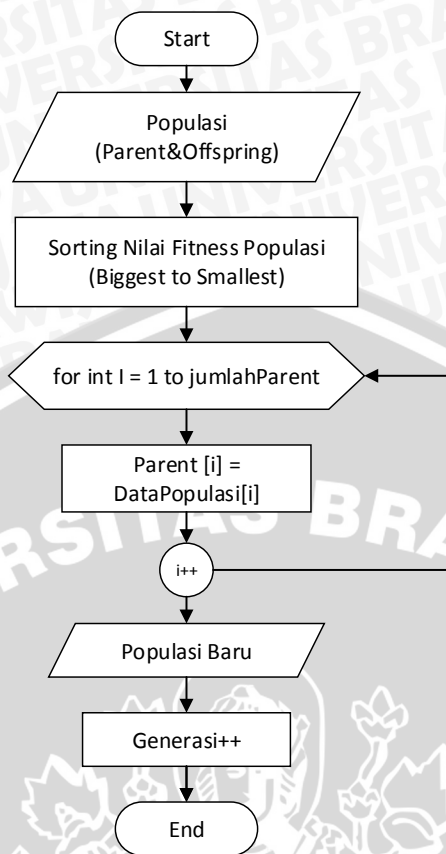
Tabel 4.13 merupakan tabel hasil evaluasi dari *parent* dan seluruh *offSpring* yang dihasilkan dari proses *crossover* dan mutasi sebelumnya. Pada setiap individu dihitung nilai *fitness*nya untuk mengetahui kebugaran individu tersebut kemudian dilakukan proses seleksi. Individu dari *parent* disimbolkan dengan P sedangkan individu hasil reproduksi / *offSpring* disimbolkan dengan C.

Tabel 4.13 Evaluasi

Individu P(t)&C(t)	Kromosom										FITNESS
	1	2	3	4	5	6	7	8	9	10	
P1	0	1	0	1	1	0	1	1	1	1	18.2173
P2	1	0	0	0	0	1	0	1	0	1	11.7250
P3	0	0	0	1	0	1	1	0	0	0	11.7412
P4	0	1	0	0	0	0	1	1	1	0	16.5622
P5	1	0	1	0	1	0	0	0	1	1	11.7993
P6	1	1	0	0	0	0	1	1	1	0	17.1807
P7	0	0	1	0	1	1	1	0	1	1	11.5791
P8	1	1	0	0	0	0	1	0	0	0	16.3587
P9	0	1	0	0	0	0	1	0	1	0	16.4983
P10	1	1	1	0	0	0	1	1	1	1	18.3875
P11	1	1	1	0	1	1	1	1	0	0	11.3270
P12	1	1	1	0	1	1	0	0	0	1	18.2110
P13	0	1	0	1	1	1	0	1	1	1	18.7993
P14	1	1	1	1	0	0	0	0	0	0	15.7699
P15	1	0	1	0	1	1	0	1	1	0	11.5436
C1	1	1	0	0	1	1	1	0	1	1	19.4561
C2	0	0	1	0	0	0	1	1	1	0	11.2898
C3	0	1	0	0	0	0	1	0	0	0	16.1987
C4	1	1	0	0	0	0	1	1	1	0	17.1807
C5	1	0	0	0	0	1	0	1	1	0	11.7403
C6	1	0	1	0	1	1	0	1	0	1	11.8436
C7	0	1	1	0	1	1	0	0	0	1	18.4427
C8	1	1	0	1	1	0	1	1	1	1	18.5007
C9	1	0	0	0	1	0	0	0	1	1	11.8541
C10	0	1	0	1	0	0	1	0	1	0	16.1344
C11	1	1	1	0	0	1	1	1	1	1	18.5978
C12	1	0	1	0	1	1	1	1	0	0	11.4080
C13	0	1	1	0	1	1	0	0	0	1	18.4427
C14	0	1	0	1	1	1	1	1	1	1	18.9509
C15	1	1	1	1	0	0	0	1	0	0	16.5033

Proses selanjutnya yaitu proses seleksi menggunakan metode *elitism selection*. Pada metode *elitism selection* dilakukan pengurutan menurun berdasarkan nilai *fitness* kromosom mulai dari nilai *fitness* tertinggi hingga nilai *fitness* terendah. Metode *elitism selection* ini akan menjamin individu yang terbaik akan selalu lolos (Mahmudy, 2013). Hasil akhir seleksi yaitu individu-individu yang paling baik sejumlah populasi awal untuk dijadikan sebagai generasi selanjutnya. Diagram alir proses seleksi menggunakan metode *elitism selection* akan ditampilkan pada Gambar 4.3.





Gambar 4.3 Diagram Alir Proses *Elitism selection*

Sumber: Perancangan

Setelah proses seleksi didapatkan populasi dengan nilai *fitness* optimal sebesar populasi awal.

Tabel 4.14 Seleksi

P(t-1)	P(t) asal	Fitness
P1	C1	19.4561
P2	C14	18.9509
P3	P13	18.7993
P4	C11	18.5978
P5	C8	18.5007
P6	C13	18.4427
P7	P10	18.3875
P8	P1	18.2173
P9	P12	18.2110
P10	P6	17.1807
P11	C4	17.1807
P12	P4	16.5622
P13	C15	16.5033
P14	P9	16.4983
P15	P8	16.3590

Proses seleksi yang dilakukan menggunakan metode *elitism selection* yaitu memilih satu individu dengan nilai *fitness* terbaik. 15 individu dengan *fitness*

terbaik tersebut akan menjadi *parent*/induk pada generasi selanjutnya. Berdasarkan data Tabel 4.14 pada contoh persoalan ini individu yang memiliki nilai *fitness* tertinggi adalah individu C1 dengan *fitness* sebesar 19.4561. Dari individu C1 ini didapatkan persamaan regresi baru yaitu :

$$Y' = 2.6424 + (0.126) X_1 + (1.0823) X_2 + (-0.478) X_3 + (-0.3303) X_4 + (0.19454) X_5 + (-0.45) X_6 + (-0.248) X_7$$

Karena individu C1 memiliki nilai *fitness* terbaik maka representasi binernya digunakan untuk memprediksi curah hujan menggunakan pemodelan regresi linear. Berikut ini adalah representasi biner individu C1.

C1	1	1	0	0	1	1	1	0	1	1
----	---	---	---	---	---	---	---	---	---	---

Berdasarkan representasi biner individu C1 maka data yang digunakan adalah data 7 periode yaitu dasarian ke-1, 2, 5, 6, 7, 9, dan 10 kebelakang.

Dengan menggunakan jumlah periode yang sama yaitu 7, dilakukan perhitungan regresi linear tanpa menggunakan algoritma genetika yang berarti data dasarian diambil secara urut. Hasil perhitungan tersebut ditampilkan pada Tabel 4.15 berikut.

Tabel 4.15 Prediksi Regresi Tanpa Algoritma Genetika

No	Bulan	Dasarian	Y	X1	X2	X3	X4	X5	X6	X7	Y'	Error
1	Juni	2	1.90	7.40	0.00	7.60	10.80	0.00	-7.70	5.50	1.70	0.0389
2	Juni	3	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70	2.16	4.6768
3	Juli	1	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00	-0.41	0.8268
4	Juli	2	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.42	2.2027
5	Juli	3	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60	-2.61	6.8080
6	Agustus	1	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00	2.15	4.6120
7	Agustus	2	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40	1.96	3.8438
8	Agustus	3	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90	0.99	0.9735
9	September	1	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00	0.33	0.1101
10	September	2	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50	1.49	2.2270
11	September	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90	1.66	2.7713
12	Oktober	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.62	2.6255
13	Oktober	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.62	2.6255
14	Oktober	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.62	2.6255
15	November	1	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.62	5.1968
16	November	2	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00	2.10	90.2821
17	November	3	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00	7.60	6.2251
18	Desember	1	18.60	5.10	11.60	3.90	0.00	0.00	0.00	0.00	16.61	3.9747
19	Desember	2	10.10	18.60	5.10	11.60	3.90	0.00	0.00	0.00	11.57	2.1695
20	Desember	3	21.55	10.10	18.60	5.10	11.60	3.90	0.00	0.00	20.95	0.3605
											jumlah Error	145.1762
											akar	12.0489
											MSE	0.6024
											fitness	17.8490

Nilai *fitness* yang dihasilkan dari pemodelan regresi menggunakan algoritma genetika lebih tinggi dibandingkan nilai *fitness* yang dihasilkan dari pemodelan regresi tanpa pemilihan periode menggunakan algoritma genetika. Peran algoritma genetika pada pemodelan regresi ini adalah untuk menentukan koefisien yang digunakan sehingga nilai prediksi yang dihasilkan lebih baik.

4.4 Perancangan Uji Coba

Parameter yang dibuat saat inialisasi awal tidak memiliki metode khusus untuk menentukannya oleh karena itu diperlukan beberapa kali pengujian untuk mendapat solusi yang paling optimal. Beberapa pengujian yang dilakukan adalah :

1. Pengujian ukuran populasi (*popSize*).
2. Pengujian jumlah generasi.
3. Pengujian kombinasi parameter *cr* dan *mr*.
4. Pengujian jumlah data yang digunakan untuk prediksi.

4.4.1 Rancangan Uji Coba Ukuran Populasi (*popSize*)

Ukuran populasi (*popSize*) yang digunakan pada implementasi algoritma genetika akan mempengaruhi solusi yang dihasilkan. Oleh karena itu perlu dilakukan uji coba untuk mendapatkan ukuran populasi yang akan menghasilkan solusi yang optimal. Pada perancangan uji coba ini ukuran populasi yang digunakan adalah kelipatan 20. Parameter yang digunakan pada uji coba ukuran populasi adalah :

- *popSize* : 10-110
- Jumlah Generasi : 5
- *Crossover Rate* (*cr*) : 0,6
- *Mutation rate* (*mr*) : 0,4

Tabel 4.16 Rancangan Uji Coba Ukuran Populasi

Ukuran Populasi	Nilai <i>fitness</i> percobaan ke-										Rata-rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											
40											
50											
60											
...											
...											
110											

Tabel 4.16 merupakan tabel rancangan untuk melakukan uji coba ukuran populasi yang optimal. Tabel tersebut menampilkan ukuran populasi mulai dari 10 hingga 110, dan nilai *fitness* setiap ukuran populasi yang diuji 10 kali. Selain itu juga terdapat kolom rata-rata *fitness* dari seluruh generasi pada tiap ukuran populasi.

4.4.2 Rancangan Uji Coba Jumlah Generasi

Uji coba jumlah generasi dilakukan untuk mendapatkan jumlah generasi yang akan menghasilkan solusi optimal. Pada uji coba ini jumlah generasi yang digunakan adalah kelipatan 5 dengan batas maksimal jumlah generasi adalah 25. Perancangan tabel uji coba jumlah generasi akan ditampilkan pada Tabel 4.17. Parameter yang digunakan pada uji coba ukuran populasi adalah :

- *popSize* : ukuran populasi terbaik pada uji coba sebelumnya
- Jumlah Generasi : 5-25
- *Crossover Rate* (cr) : 0,6
- *Mutation rate* (mr) : 0,4

Tabel 4.17 Rancangan Uji Coba Jumlah Generasi

Jumlah Generasi	Nilai <i>fitness</i> percobaan ke-										Rata-rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
5											
10											
15											
20											
25											

4.4.3 Rancangan Uji Coba Kombinasi cr dan mr

Pada perancangan uji coba kombinasi nilai *crossover rate* dan *mutation rate* diperlukan beberapa parameter diantaranya :

- *popSize* : ukuran populasi terbaik pada uji coba sebelumnya
- Jumlah Generasi : jumlah generasi terbaik pada uji coba sebelumnya
- *Crossover Rate* (cr) : 0-1
- *Mutation rate* (mr) : 1-0

Tabel 4.18 Rancangan Uji Coba Kombinasi cr dan mr

Kombinasi		Nilai <i>fitness</i> percobaan ke-										Rata-rata <i>Fitness</i>
Cr	mr	1	2	3	4	5	6	7	8	9	10	
0	1											
0,1	0,9											
0,2	0,8											
0,3	0,7											
0,4	0,6											
0,5	0,5											
0,6	0,4											
0,7	0,3											
0,8	0,2											
0,9	0,1											
1	0											



4.4.4 Rancangan Uji Coba Jumlah Data yang Digunakan

Uji coba jumlah data dilakukan untuk mendapatkan jumlah data yang akan menghasilkan solusi optimal. Pada uji coba ini jumlah data yang digunakan adalah 15-45. Perancangan tabel uji coba jumlah data akan ditampilkan pada Tabel 4.19. Parameter yang digunakan pada uji coba ini adalah :

- *popSize* : ukuran populasi terbaik pada uji coba sebelumnya
- Jumlah Generasi : generasi terbaik pada uji coba sebelumnya
- *Crossover Rate* (*cr*) : *cr* terbaik pada uji coba sebelumnya
- *Mutation rate* (*mr*) : *mr* terbaik pada uji coba sebelumnya

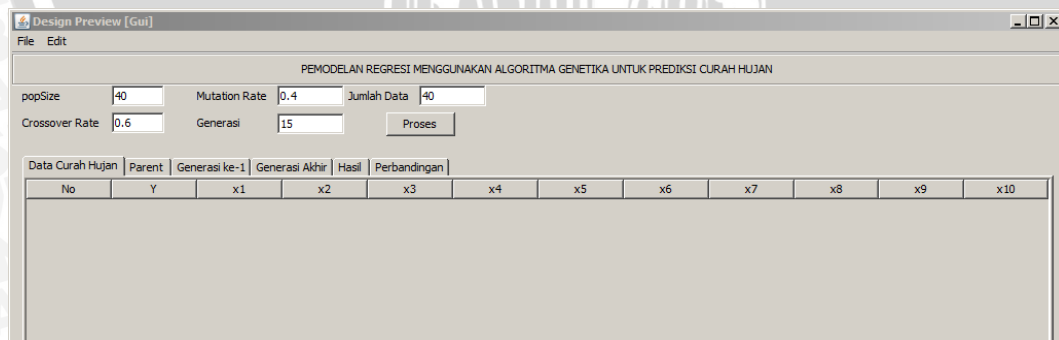
Tabel 4.19 Rancangan Uji Coba Jumlah Data yang Digunakan

Jumlah Data	Nilai <i>fitness</i> percobaan ke-										Rata-rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
5											
10											
15											
20											
25											

4.5 Perancangan *User interface*

Pada perancangan *user interface* aplikasi yang akan dibuat terdapat tiga halaman utama yaitu halaman Home, halaman *Parent* , halaman Generasi Awal, halaman Generasi Akhir, halaman Hasil, dan halaman Perbandingan. Rancangan masing-masing halaman akan dijabarkan pada sub-bab berikut.

4.5.1 Rancangan *User interface* Halaman Data Curah Hujan



Gambar 4.4 Rancangan *User interface* Halaman Data Curah Hujan

Sumber: Perancangan

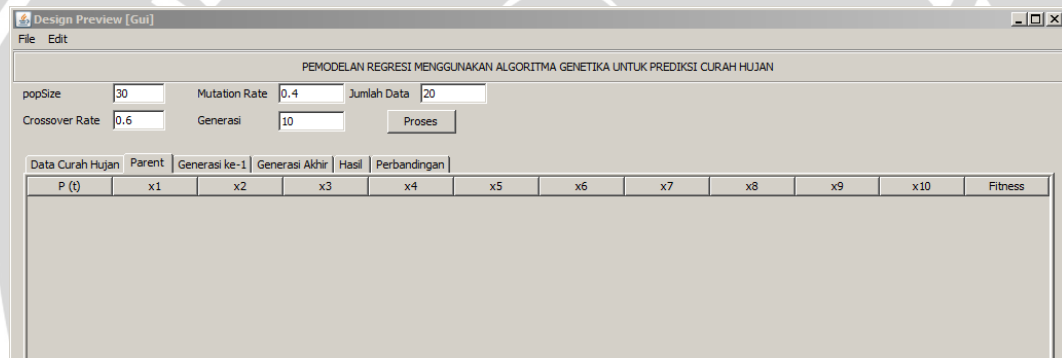
Gambar 4.4 merupakan rancangan *user interface* pada halaman home dari aplikasi yang akan dibuat. Pada bagian atas terdapat label judul aplikasi ini yaitu “Pemodelan Regresi Menggunakan Algoritma Genetika Untuk Prediksi Curah

Hujan”. Kemudian terdapat *textField* untuk menginputkan beberapa parameter yang dibutuhkan untuk menjalankan proses algoritma genetika, yaitu :

1. *TextField* ‘Jumlah Data’ untuk memasukkan jumlah data yang akan digunakan dalam prediksi menggunakan algoritma genetika.
2. *TextField* ‘popSize’ untuk memasukkan jumlah populasi.
3. *TextField* ‘Crossover rate’ untuk memasukkan nilai *crossover rate*.
4. *TextField* ‘MutationRate’ untuk memasukkan nilai *mutation rate*.
5. *TextField* ‘Generasi’ untuk memasukkan jumlah iterasi yang akan dilakukan.
6. Button ‘Proses’ untuk melakukan proses algoritma.

4.5.2 Rancangan *User interface* Halaman *Parent*

Pada perancangan *user interface* halaman *parent*, nilai parameter-parameter yang telah diinputkan dan dijalankan oleh sistem tetap akan ditampilkan untuk memudahkan *user*.



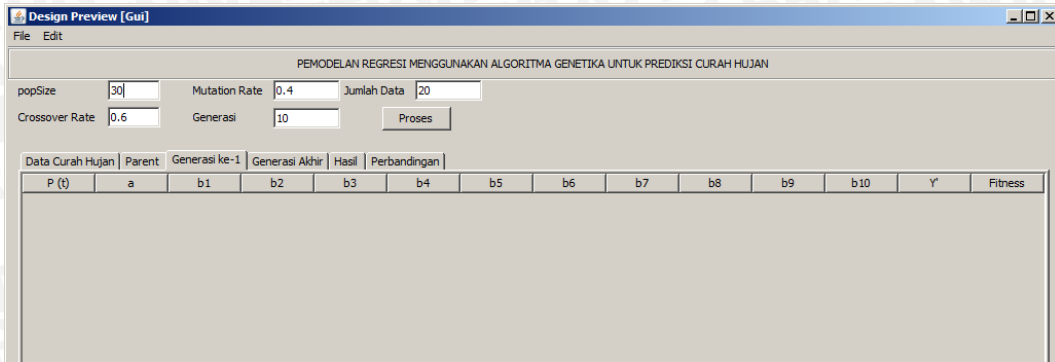
Gambar 4.5 Rancangan *User interface* Halaman *Parent*

Sumber: Perancangan

Gambar 4.5 adalah rancangan *user interface* halaman *Parent* yang akan menampilkan sebuah tabel dengan kolom :

1. Kolom *P(t)* untuk menampilkan keterangan nama individu dengan jumlah individu yang ditampilkan adalah sesuai dengan *popSize* yang diinputkan oleh *user*.
2. Kolom *x1* hingga *x10* menampilkan representasi kromosom biner yang dibangkitkan secara random.
3. Kolom *Fitness* yang menampilkan nilai *Fitness* yang dihasilkan oleh individu pada populasi *parent*.

4.5.3 Rancangan *User interface* Halaman Generasi Awal



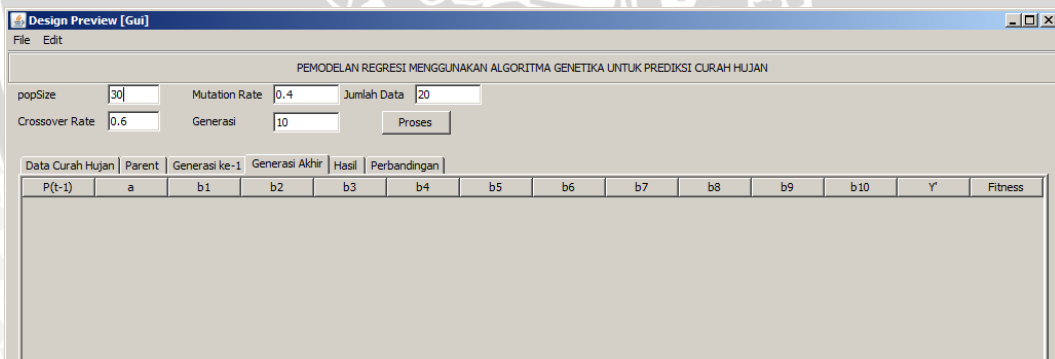
Gambar 4.6 Rancangan *User interface* Halaman Proses Generasi ke-1

Sumber: Perancangan

Gambar 4.6 merupakan gambar rancangan *user interface* aplikasi pada halaman Generasi ke-1. Halaman ini menampilkan tabel dengan beberapa kolom yaitu :

1. Kolom Individu, menampilkan nama individu hasil generasi ke-1.
2. Kolom a, b1, b2, hingga b10 yang menampilkan kromosom hasil seleksi generasi *parent*.
3. Y' menampilkan hasil prediksi tiap individu pada populasi.
4. Kolom *fitness* yang menampilkan nilai kebugaran tiap individu.

4.5.4 Rancangan *User interface* Halaman Generasi Akhir

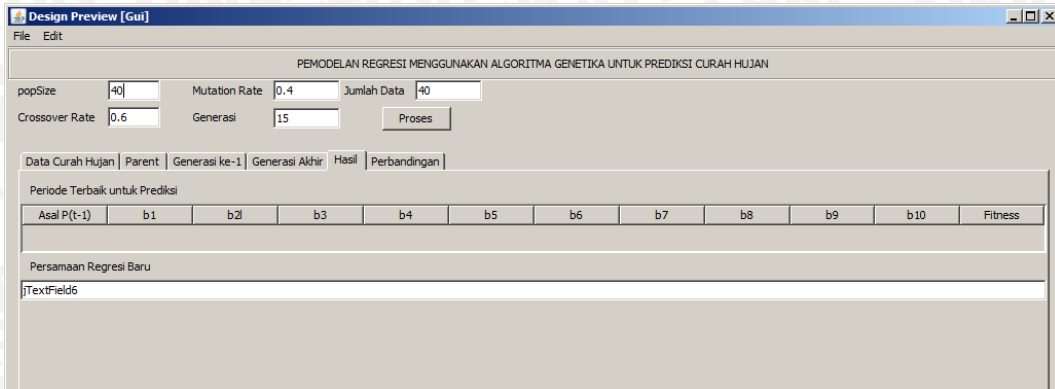


Gambar 4.7 Rancangan *User interface* Halaman Proses Generasi Akhir

Sumber: Perancangan

Rancangan *user interface* proses generasi akhir ditampilkan pada Gambar 4.7. Sama halnya dengan halaman generasi ke-1, pada halaman ini terdapat tabel yang menampilkan individu-individu pada generasi akhir dengan beberapa kolom diantaranya keterangan individu, nilai kromosom a, b1, b2, hingga b10, dan nilai *fitness* masing-masing individu pada populasi.

4.5.5 Rancangan *User interface* Hasil



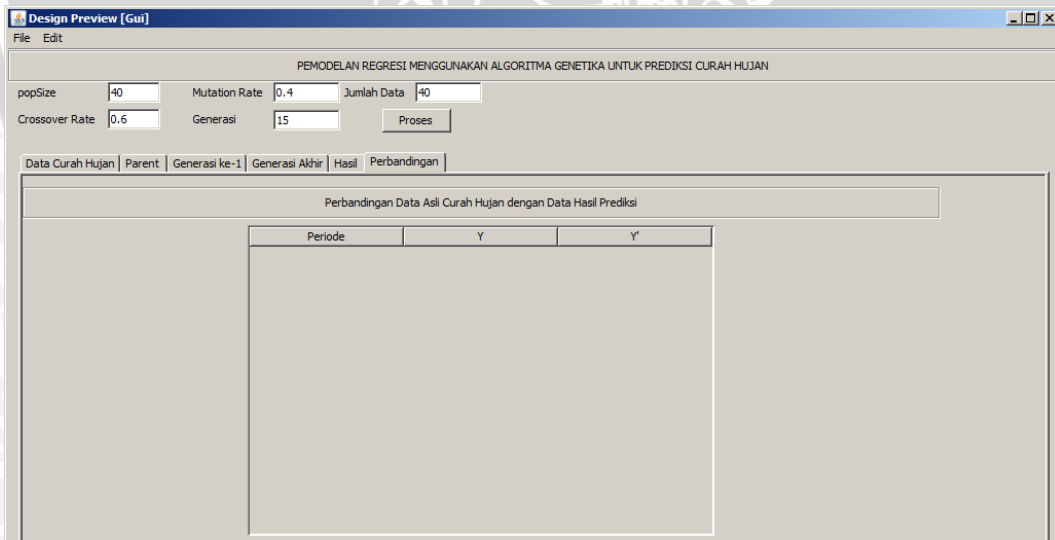
Gambar 4.8 Rancangan *User interface* Halaman Hasil

Sumber: Perancangan

Pada Gambar 4.8 *user interface* halaman 'Hasil' terdapat sebuah tabel dengan beberapa item diantaranya :

1. Tabel 'Periode Terbaik Untuk Prediksi' menampilkan representasi biner dari individu terpilih yang menghasilkan nilai *fitness* tertinggi. Kromosom yang ditampilkan ini dapat dijadikan acuan untuk periode yang digunakan dalam melakukan prediksi menggunakan pemodelan regresi.
2. *TextField* 'Persamaan Regresi Baru' yang akan menampilkan persamaan regresi baru yang dihasilkan dari proses algoritma menggunakan parameter yang telah diinputkan.

4.5.6 Rancangan *User interface* Halaman Perbandingan



Gambar 4.9 Rancangan *User interface* Halaman Perbandingan

Sumber: Perancangan

Pada perancangan *user interface* halaman perbandingan ditampilkan sebuah tabel yang memiliki kolom:

1. Kolom nama periode, yaitu menampilkan dasarian dan nama bulan.
2. Kolom Y yang menampilkan data asli curah hujan secara urut sesuai jumlah data yang diinputkan oleh *user*.
3. Kolom Y' yang menampilkan hasil prediksi yang dihasilkan dari proses algoritma genetika menggunakan sistem.



BAB 5 IMPLEMENTASI

5.1 Implementasi Program

Pada sub-bab berikut akan dijelaskan mengenai struktur data yang digunakan dan source code implementasi algoritma yang digunakan.

5.1.1 Proses Ambil Data

Proses pengambilan data yang digunakan adalah dengan mengambil data yang telah disimpan pada database dengan nama file data_asli.sql. Jumlah data keseluruhan pada database adalah 360, dengan jumlah data per tahun adalah 36. Data curah hujan dasarian yang digunakan dalam penelitian ini adalah data yang diperoleh dari Stasiun Observasi Tukur. Berikut ini merupakan source code proses pengambilan data dari database yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Source Code Ambil Data

```

1  Double[][] getDataFromDB() {
2      double dataDB[] = new double[360];
3      Data = new Double[jumlahData][width + 1];
4      try {
5          rs = st.executeQuery("SELECT * from data_asli");
6          rs.beforeFirst();
7          int no = 0;
8          while (rs.next()) {
9              for (int i = (no * 36); i < 36 * (no + 1); i++) {
10                 dataDB[i] = (rs.getDouble(i - (no * 36) +
11                 1));
12             }
13             no++;
14         }
15     } catch (SQLException ex) {
16     }
17     for (int i = 0; i < jumlahData; i++) {
18         for (int j = 0, k = 360 - jumlahData + i; j < width + 1;
19         j++, k--) {
20             Data[i][j] = dataDB[k];
21         }
22     }
23     return Data;
24 }

```

Fungsi jumlahData adalah variabel yang digunakan untuk menyimpan nilai input parameter jumlah data oleh *user*.

5.1.2 Proses Inialisasi Kromosom Biner

Pembangkitan kromosom awal pada populasi dilakukan berdasarkan ukuran populasi (*popSize*) yang diinputkan oleh *user*. Kromosom dibangkitkan secara random dengan range angka 0 dan 1. Panjang kromosom ditentukan sepanjang 10. Sehingga populasi kromosom dibangkitkan sepanjang jumlah populasi dengan lebar 10. Setelah kromosom dibangkitkan, kromosom yang bernilai 1 dihitung dengan counter. Berikut ini merupakan proses inialisasi kromosom yang ditunjukkan pada Tabel 5.2.

Tabel 5.2 Source Code Inisialisasi Kromosom

```

1 Object[] generateRandChromosome(int width) {
2     Object chromosome[] = new Object[width];
3     panjangChromosom = new int[popSize];
4     for (int i = 0; i < width; i++) {
5         chromosome[i] = randomRange(0, 1);
6         if ("1".equals(chromosome[i].toString())) {
7             panjangChromosom[counter]++;
8         }
9     }
10    counter++;
11    return chromosome;

```

Pada proses inisialisasi kromosom, panjang kromosom sudah ditentukan sebesar 10. Baris ke-5 adalah pernyataan untuk melakukan generate kromosom secara random kromosom biner yang bernilai 0 dan 1. Representasi biner dibangkitkan sepanjang panjang kromosom dan sebanyak jumlah ukuran populasi.

5.1.3 Proses Generate Random *Parent* Awal

Individu-individu pada populasi awal merupakan *parent* yang diacak untuk dapat melakukan reproduksi. Proses random *parent* populasi awal akan dijelaskan pada Tabel 5.3 berikut ini.

Tabel 5.3 Source Code Generate Random *Parent* Awal

```

1     Object[][] generateRandParent(int width) {
2         parent = new Object[popSize][width + 2];
3         Object chromosome[];
4         for (int i = 0; i < popSize; i++) {
5             chromosome = generateRandChromosome(width);
6             dataFitness = new
7             Double[jumlahData][panjangChromosom[i] + 1];
8             for (int j = 0; j < jumlahData; j++) {
9                 dataFitness[j][0] =
10                Double.valueOf(datafromDB[j][0].toString());
11                for (int k = 0, l = 1; k < width; k++) {
12                    if ("1".equals(chromosome[k].toString())) {
13                        dataFitness[j][l] =
14                        Double.valueOf(datafromDB[j][k + 1].toString());
15                        l++;
16                    }
17                }
18            }
19            parent[i][0] = "P" + (i + 1);
20            System.arraycopy(chromosome, 0, parent[i], 1, width);
21            parent[i][width + 1] = fitness(dataFitness);
22        }
23    }
24    return parent;

```

Untuk melakukan perhitungan *fitness* maka diperlukan jumlah kromosom yang terlibat dalam proses prediksi. Pengecekan jumlah kromosom yang bernilai 1 pada

satu individu dilakukan mulai dari individu pertama sepanjang jumlah kromosom dan sebanyak jumlah individu yang dibuat. Setelah jumlah kromosom bernilai 1 didapatkan maka data pada periode kromosom tersebut akan diambil dari database.

5.1.4 Proses Perhitungan *Error* dan *Fitness*

Pada proses perhitungan *fitness*, perhitungan nilai *error* dilakukan terlebih dahulu. Perhitungan *fitness* dilakukan pada setiap populasi dalam sebuah generasi.

Tabel 5.4 Source Code *Error* dan *Fitness*

```

1  double fitness(Double dataFitness[][]) {
2      double y[] = new double[jumlahData], Y[] = y, error[] = y,
3      sumError = 0, akar, MSE,
4      fitness, x[][] = new
5  double[jumlahData][dataFitness[0].length - 1];
6      int panjang = x[0].length;
7      for (int i = 0; i < jumlahData; i++) {
8          y[i] = dataFitness[i][0];
9          for (int j = 0; j < panjang; j++) {
10             x[i][j] = dataFitness[i][j + 1];
11         }
12     }
13     OLSMultipleLinearRegression regression = new
14     OLSMultipleLinearRegression();
15     regression.newSampleData(y, x);
16     double[] beta = regression.estimateRegressionParameters();
17     //Y'
18     for (int i = 0; i < jumlahData; i++) {
19         Y[i] = beta[0];
20         for (int j = 0; j < panjang; j++) {
21             Y[i] = Y[i] + (x[i][j] * beta[j + 1]);
22         }
23     }
24     //error
25     for (int i = 0; i < jumlahData; i++) {
26         double min = (Y[i] - dataFitness[i][0]);
27         error[i] = Math.pow(min, 2);
28         sumError = sumError + error[i];
29     }
30     //akar
31     akar = Math.sqrt(sumError);
32     //MSE
33     MSE = 1 / Double.valueOf(jumlahData + "");
34     MSE = MSE * akar;
35     //fitness
36     fitness = 10 / MSE;
37     fitness = fitness + (1 / (0.1 * (x[0].length + 1)));
38     return fitness;
39 }

```

Proses perhitungan *fitness* dilakukan sebanyak jumlah data yang disimpan pada fungsi jumlahData sepanjang data panjangChromosom. Data tersebut kemudian dihitung nilai prediksinya (Y') pada baris ke-21. Setelah nilai Y' didapatkan maka nilai *error* tiap baris data dihitung pada baris ke-27. Perulangan ini dilakukan sebanyak jumlah data.

5.1.5 Proses Perhitungan Crossover

Perhitungan crossover dilakukan dengan menggunakan metode *one cut point*. Prosesnya adalah memilih satu titik pada kromosom *parent* ke-1 dan *parent* ke-2 secara random. Setelah cut point didapatkan maka kromosom sebelum cut point dicopy dan kromosom setelah cut point akan ditukarkan dengan pasangannya. Proses crossover dijelaskan pada source code 5.5 berikut ini.

Tabel 5.5 Source Code Crossover

```

1  Object[][] crossover(Object[][] parent) {
2      int offspring = (int) Math.round(cr * popSize), widthP =
3      parent[0].length, rand = 0;
4      randParent = shuffledArray(0, popSize - 1);
5      for (int i = 0; i < offspring; i = i + 2) {
6          int cutPoint = randomRange(1, widthP - 3);
7          child[i][0] = parent[randParent[i]][0] + " C" + (i + 1);
8          System.arraycopy(parent[randParent[i]], 1, child[i], 1,
9          cutPoint);
10         System.arraycopy(parent[randParent[i + 1]], cutPoint + 1,
11         child[i], cutPoint + 1, widthP - cutPoint - 1);
12
13
14         dataFitness = new
15         Double[jumlahData][getpanjangChromosomChild(child[i]) + 1];
16         for (int j = 0; j < jumlahData; j++) {
17             dataFitness[j][0] =
18             Double.valueOf(datafromDB[j][0].toString()); //P11 C20 C29 C12 C29
19             C25 C24
20             for (int k = 0, l = 1; k < width; k++) {
21                 if ("1".equals(child[i][k + 1].toString())) {
22                     dataFitness[j][l] =
23                     Double.valueOf(datafromDB[j][k + 1].toString());
24                     l++;
25                 }
26             }
27         }
28     }
29     child[i][width + 1] = fitness(dataFitness);
30
31
32     child[i + 1][0] = parent[randParent[i + 1]][0] + " C" + (i +
33     2);
34     System.arraycopy(parent[randParent[i + 1]], 1, child[i + 1], 1,
35     cutPoint);
36     System.arraycopy(parent[randParent[i]], cutPoint + 1, child[i +
37     1], cutPoint + 1, widthP - cutPoint - 1);
38
39
40     dataFitness = new
41     Double[jumlahData][getpanjangChromosomChild(child[i + 1]) + 1];
42
43     for (int j = 0; j < jumlahData; j++) {
44         dataFitness[j][0] =
45         Double.valueOf(datafromDB[j][0].toString());
46         for (int k = 0, l = 1; k < width; k++) {
47             if ("1".equals(child[i + 1][k + 1].toString())) {
48                 dataFitness[j][l] = Double.valueOf(datafromDB[j][k +
49                 1].toString());
50                 l++;
51             }
52         }
53     }
54 }

```

```

52     }
53     }
54     }
55     child[i + 1][width + 1] = fitness(dataFitness);
56     }
57     }
58     return child;
59 }

```

Jumlah *offSpring* yang akan dihasilkan dari reproduksi crossover dihitung pada baris ke -2. Titik potong dibuat secara random pada baris ke-7. Setelah individu *offSpring* dihasilkan maka nilai *fitness* dihitung sesuai panjangChromosomChild.

5.1.6 Proses Perhitungan Mutasi

Pada proses mutasi kromosom biner, dilakukan pemilihan titik mutasi (mutation point) secara random. Kromosom yang mengalami mutasi akan menginverskan nilai kromosomnya, apabila nilai kromosom 1 termutasi maka nilai kromosom tersebut diubah menjadi 0. Proses mutasi dijelaskan pada Tabel 5.6 berikut.

Tabel 5.6 Source Code Mutasi

```

1  Object[][] mutasi(Object[][] parent) {
2      int offSpring = (int) Math.round(mr * popSize), offSpringCR =
3      (int) Math.round(cr * popSize), widthP = parent[0].length;
4      for (int i = offSpringCR; i < popSize; i++) {
5          int point = randomRange(1, widthP - 3);
6          child[i][0] = parent[randParent[i]][0] + " C" + (i + 1);
7          System.arraycopy(parent[randParent[i]], 1, child[i], 1,
8          widthP - 1);
9          if ("0".equals(child[i][point].toString())) {
10             child[i][point] = "1";
11         } else if ("1".equals(child[i][point].toString())) {
12             child[i][point] = "0";
13         }
14         dataFitness = new
15         Double[jumlahData][getpanjangChromosomChild(child[i]) + 1];
16         for (int j = 0; j < jumlahData; j++) {
17             dataFitness[j][0] =
18             Double.valueOf(datafromDB[j][0].toString());
19             for (int k = 0, l = 1; k < width; k++) {
20                 if ("1".equals(child[i][k + 1].toString())) {
21                     dataFitness[j][l] =
22                     Double.valueOf(datafromDB[j][k + 1].toString());
23                     l++;
24                 }
25             }
26         }
27         child[i][width + 1] = fitness(dataFitness);
28     }
29     return child;
30 }

```

Pada proses mutasi, jumlah *offSpring* yang dihasilkan dihitung pada baris ke-2. Titik mutasi didapatkan secara random pada baris ke-5. Proses mutasi dilakukan pada baris ke-9. Kromosom yang terpilih pada titik mutasi apabila bernilai 0 maka nilainya kan diganti dengan nilai 1 dan sebaliknya.

5.1.7 Proses Evaluasi Seleksi

Tabel 5.7 Source Code Evaluasi dan Seleksi

```
1 Object[][] evaluasiSeleksi(Object[][] parent, Object[][] child) {
2     Object newParent[][] = parent, gabungan[][] = new Object[popSize
3     * 2][parent[0].length];
4     //gabung parent+child
5     System.arraycopy(parent, 0, gabungan, 0, popSize);
6     System.arraycopy(child, 0, gabungan, popSize, popSize);
7     //seleksi
8     newParent = insertionSort(gabungan);
9     System.out.println("");
10    System.arraycopy(cekFitnessSama(insertionSort(gabungan)), 0,
11    newParent, 0, popSize);
12    return newParent;
13 }
14
15 Object[][] insertionSort(Object array[][]){
16     int n = array.length, m = array[0].length;
17     Object key[] = new Object[m];
18     for (int j = 1; j < n; j++) {
19         System.arraycopy(array[j], 0, key, 0, m);
20         int i = j - 1;
21         while ((i > -1) && (Double.valueOf(array[i][width +
22         1].toString()) < Double.valueOf(key[width + 1].toString()))) {
23             System.arraycopy(array[i], 0, array[i + 1], 0, m);
24             i--;
25         }
26         System.arraycopy(key, 0, array[i + 1], 0, m);
27     }
28     return array;
29 }
30
31 Object[][] cekFitnessSama(Object sort[][]){
32     Object hasil[][] = sort;
33     int j = 0;
34     for (int i = 0; i < sort.length - 1; i++) {
35         if (sort[i][width + 1].equals(sort[i + 1][width +
36         1])) {
37             if ((getpanjangChromosomChild(sort[i])>3) &&
38             getpanjangChromosomChild(sort[i])<getpanjangChromosomChild(h
39             asil[j])) {
40                 System.arraycopy(sort[i], 0, hasil[j], 0, width + 2);
41             }
42             continue;
43         }
44         System.arraycopy(sort[i], 0, hasil[j], 0, width + 2);
45         j++;
46     }
47     return hasil;
48 }
```

Pada proses evaluasi individu *parent* dan *offSpring* akan digabung menjadi satu kedalam array gabungan untuk kemudian dilakukan proses seleksi. Seleksi yang dilakukan yaitu menggunakan metode *Elitism selection*, dimana semua individu *parent* dan *offSpring* yang digabung akan diurutkan nilai *fitness*nya dari yang terbesar hingga terkecil. Individu yang terpilih adalah individu teratas sesuai jumlah populasi awal. Apabila terdapat individu dengan nilai *fitness* yang sama maka akan dilakukan pengecekan jumlah kromosom prediksi, individu yang

terpilih nantinya adalah individu dengan jumlah kromosom prediksi yang lebih sedikit.

5.1.8 Proses Regresi

Berikut ini merupakan source code regresi linear untuk mendapatkan intercept dan koefisien yang diperlukan pada proses regresi.

Tabel 5.8 Source Code Regresi

```

1 OLSMultipleLinearRegression regression = new
2 OLSMultipleLinearRegression();
3     regression.newSampleData(y, x);
4     double[] beta = regression.estimateRegressionParameters();

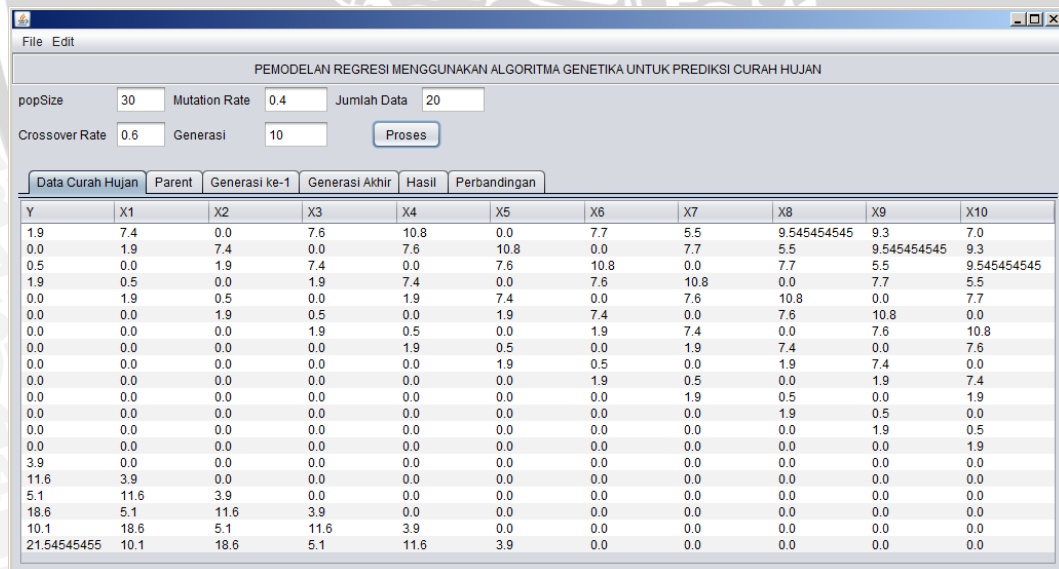
```

5.2 Implementasi User interface

User interface pada sistem ini terdiri dari 6 halaman yaitu halaman Data Curah Hujan, halaman *Parent*, halaman Generasi ke-1, halaman Generasi Akhir, halaman Hasil, dan halaman Perbandingan.

5.2.1 Halaman Data Curah Hujan

Halaman ini merupakan halaman default ketika aplikasi dijalankan pertama kali. Terdapat *textField* untuk memasukkan nilai parameter-parameter yang dibutuhkan sistem oleh *user* dan *button* Proses untuk menjalankan proses algoritma. Parameter tersebut adalah *popSize* (jumlah populasi), nilai *cr*, nilai *mr*, jumlah generasi dan jumlah data yang akan digunakan.



Gambar 5.1 Implementasi Halaman Data Curah Hujan

Setelah *user* memasukkan parameter dan menekan *button* proses maka halaman Data Curah Hujan akan menampilkan data 20 dasarian kebelakang. Terdapat kolom Y, X1, X2 hingga X10. Y merupakan data asli curah hujan dalam

satuan millimeter. X1, X2, hingga X10 merupakan data dasarian periode sebelumnya.

5.2.2 Halaman Populasi *Parent*

Halaman *Parent* menampilkan populasi awal generasi pertama yang terdiri dari kolom nama individu, kolom X1 hingga X10, dan nilai *fitness* yang dihasilkan oleh generasi *parent* tersebut. Populasi yang ditampilkan pada halaman ini sesuai dengan jumlah masukan *user*.

P(i)	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Fitness
P1	1	1	1	0	1	1	0	0	0	1	18.2109634...
P2	0	1	0	1	1	1	0	0	0	1	18.8615648...
P3	0	0	0	1	1	1	0	1	1	1	12.1313833...
P4	1	0	0	1	1	1	1	0	1	1	12.0048490...
P5	0	1	0	0	0	1	0	0	0	1	16.4961163...
P6	1	0	0	0	0	1	0	0	1	1	11.7278857...
P7	1	1	1	0	1	0	1	0	0	1	18.1283643...
P8	1	1	1	0	1	0	1	0	1	0	18.9574168...
P9	1	0	1	1	0	1	1	1	0	1	12.0746381...
P10	0	0	1	0	0	1	0	0	0	0	12.1258771...
P11	0	1	0	1	1	1	1	0	1	1	19.0894346...
P12	0	1	0	0	1	0	1	1	1	0	18.3411740...
P13	0	0	0	0	1	0	1	0	0	0	10.6933842...
P14	0	1	0	0	1	0	0	1	1	1	18.4879203...
P15	1	0	0	0	0	1	0	1	0	0	11.9863659...
P16	1	1	1	0	1	0	0	1	0	1	18.1535419...
P17	0	1	0	0	0	1	0	0	1	0	16.7401112...
P18	1	0	1	0	0	1	0	1	0	0	11.6464733...
P19	0	1	0	1	0	0	1	0	1	0	16.1343598...
P20	1	0	1	1	0	0	1	1	0	1	12.0677548...
P21	0	0	0	1	0	0	0	0	0	1	12.3250440...
P22	1	1	1	1	1	0	0	0	0	0	18.2308202...
P23	1	1	1	1	1	1	0	1	1	0	18.9692542...
P24	1	1	1	1	1	0	0	1	0	1	18.0982246...
P25	0	1	0	1	1	0	1	0	1	0	18.3385524...
P26	1	0	1	0	1	1	1	0	1	0	10.9651466...
P27	1	0	1	1	1	1	1	0	1	1	12.0769222...
P28	1	0	0	0	1	0	1	1	0	1	11.7458025...
P29	0	1	1	1	0	1	1	1	1	1	18.6039480...
P30	0	0	0	0	0	1	0	0	0	0	12.2473687...

Gambar 5.2 Implementasi Halaman Populasi *Parent*

5.2.3 Halaman Generasi Awal

Gambar 5.3 menampilkan halaman Generasi ke-1 dari proses algoritma yang dilakukan.

PEMODELAN REGRESI MENGGUNAKAN ALGORITMA GENETIKA UNTUK PREDIKSI CURAH HUJAN

popSize: 30 Mutation Rate: 0.4 Jumlah Data: 20
 Crossover Rate: 0.6 Generasi: 10 Proses

Data Curah Hujan		Parent	Generasi ke-1	Generasi Akhir	Hasil	Perbandingan							
P(t-1)	P(t) asal	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Fitness	
P1	C22	1	1	1	0	1	0	1	0	1	1	19.523087...	
P2	C26	1	1	1	0	1	1	1	0	1	1	19.413676...	
P3	C9	0	1	0	1	1	1	1	0	1	1	19.089434...	
P4	C11	1	1	1	0	1	1	1	0	1	0	18.968966...	
P5	P8	1	1	1	0	1	0	1	0	1	0	18.957416...	
P6	C20	0	1	1	0	1	0	1	1	1	0	18.947652...	
P7	P23	1	1	1	1	1	1	0	1	1	0	18.869254...	
P8	P2	0	1	0	1	1	1	0	0	1	0	18.861564...	
P9	C27	1	1	1	1	1	1	1	1	1	0	18.834100...	
P10	C3	0	1	0	0	1	0	0	1	1	0	18.665775...	
P11	C13	0	1	1	1	0	1	1	1	1	1	18.603948...	
P12	P14	0	1	0	0	1	0	0	1	1	1	18.487920...	
P13	P12	0	1	0	0	1	0	1	1	1	0	18.341174...	
P14	P25	0	1	0	0	1	0	1	0	1	0	18.338552...	
P15	C15	1	1	1	1	1	0	0	0	0	0	18.230820...	
P16	P1	1	1	1	0	1	1	0	0	0	1	18.210963...	
P17	C18	1	1	1	0	1	0	0	1	0	1	18.153541...	
P18	P7	1	1	1	0	1	0	1	0	0	1	18.128364...	
P19	C17	1	1	1	1	1	0	0	1	0	1	18.098224...	
P20	C21	1	1	0	0	0	1	0	0	1	0	17.056428...	
P21	C19	1	1	1	0	0	0	1	0	1	0	16.672937...	
P22	P19	0	1	0	1	0	0	1	0	1	0	16.134359...	
P23	C23	1	0	1	1	1	1	0	0	1	1	12.197556...	
P24	P3	0	0	0	1	1	1	0	1	1	1	12.131383...	
P25	P27	1	0	1	1	1	1	1	0	1	1	12.076922...	
P26	C5	1	0	1	1	0	1	1	1	0	1	12.074638...	
P27	C6	1	0	1	1	0	0	1	1	0	1	12.067754...	
P28	P4	1	0	0	1	1	1	1	0	1	1	12.004849...	
P29	C1	0	0	0	1	1	0	1	0	1	0	11.751596...	
P30	C16	1	0	0	0	1	0	1	1	0	1	11.745802...	

Gambar 5.3 Implementasi Halaman Generasi Awal

Individu yang ditampilkan pada halaman ini adalah individu yang memiliki *fitness* tinggi sejumlah *popSize* pada generasi sebelumnya. Kolom P(t) asal menampilkan nama individu dari proses sebelumnya beserta dengan representasi kromosom serta *fitness* yang dihasilkan.

5.2.4 Halaman Generasi Akhir

Hasil implementasi perancangan halaman yang menampilkan kromosom generasi akhir dari jumlah generasi yang diinputkan *user* ditampilkan pada Gambar 5.4. Sama dengan halaman generasi ke-1, pada halaman ini juga terdapat nama individu, asal individu, kromosom X1 hingga X10, dan nilai *fitness* dari generasi terakhir tersebut.

P(t-1)	P(t) asal	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Fitness
P1	C15	0	1	1	1	1	1	1	0	1	1	19.748582...
P2	C11	0	1	1	0	1	1	1	0	1	1	19.739404...
P3	C16	0	1	1	0	1	0	1	0	1	1	19.657401...
P4	C29	1	1	1	1	1	1	1	0	1	1	19.639919...
P5	C29	0	1	1	1	1	1	1	1	1	1	19.638139...
P6	C22	0	1	1	0	1	1	1	1	1	1	19.617007...
P7	C20	0	1	1	1	1	0	1	0	1	1	19.611086...
P8	C23	1	1	1	0	1	1	1	1	1	1	19.600617...
P9	C18	1	1	1	1	1	1	1	1	1	1	19.549996...
P10	C25	0	1	1	0	1	0	0	0	1	1	19.538674...
P11	C15	1	1	1	0	1	0	1	0	1	1	19.523087...
P12	C17	1	1	1	0	1	1	1	1	1	1	19.505968...
P13	C26	1	1	1	1	1	0	1	0	1	1	19.499845...
P14	C21	0	1	1	0	1	0	0	0	1	0	19.491760...
P15	C1	0	1	1	1	1	0	1	1	1	1	19.481180...
P16	C7	1	1	0	0	1	1	1	0	1	1	19.456143...
P17	C19	0	1	1	0	1	1	0	0	1	1	19.439705...
P18	C9	1	1	0	1	1	1	1	0	1	1	19.413676...
P19	C29	1	1	1	1	1	0	1	1	1	1	19.397858...
P20	C20	1	1	1	0	1	0	1	1	1	1	19.384198...
P21	C17	1	1	0	0	1	1	0	0	1	1	19.335575...
P22	C3	1	1	0	0	1	1	0	0	1	0	19.321718...
P23	C8	1	1	0	1	1	1	1	1	1	1	19.308055...
P24	C21	0	1	1	0	1	0	0	1	1	1	19.303160...
P25	C14	0	1	1	1	1	0	0	0	1	1	19.302925...
P26	C27	1	1	1	0	1	0	0	0	1	1	19.300599...
P27	C9	1	1	1	0	1	1	0	0	1	1	19.291415...
P28	C3	0	1	1	0	1	1	0	1	1	1	19.282036...
P29	C6	0	1	1	1	1	1	0	0	1	1	19.267881...
P30	C12	0	1	0	0	1	1	1	0	1	1	19.264060...

Gambar 5.4 Implementasi Halaman Generasi Akhir

5.2.5 Halaman Hasil

Gambar 5.5 adalah implementasi halaman Hasil yang menampilkan individu dengan *fitness* terbaik yang menghasilkan persamaan regresi baru untuk melakukan prediksi. Kromosom dari individu tersebut juga ditampilkan untuk mengetahui periode mana saja yang optimal untuk digunakan melakukan prediksi.

P(t-1)	P(t) asal	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Fitness
P1	C15	0	1	1	1	1	1	1	0	1	1	19.748582...

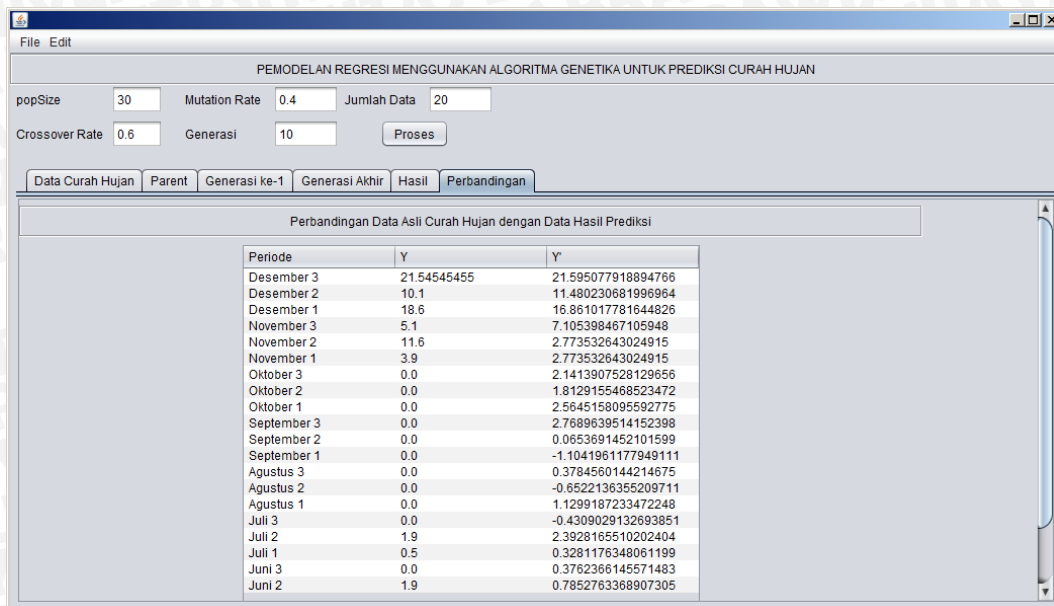
Persamaan Regresi Baru

$$Y = 2.7735 + (1.1107)X_2 + (0.3085)X_3 + (-0.1375)X_4 + (-0.4658)X_5 + (0.2016)X_6 + (0.3303)X_7 + (-0.4180)X_9 + (-0.3327)X_{10}$$

Gambar 5.5 Implementasi Halaman Hasil

5.2.6 Halaman Perbandingan

Gambar 5.6 menampilkan halaman Perbandingan yaitu perbandingan data asli curah hujan dengan data hasil prediksi menggunakan system.



Gambar 5.6 Implementasi Halaman Perbandingan

Halaman ini menampilkan perbandingan nilai data asli (Y) curah hujan dan nilai hasil prediksi (Y'). Nilai prediksi yang ditampilkan adalah hasil dari proses algoritma genetika yang dilakukan sejumlah data yang diinputkan oleh user.



BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan membahas tentang pengujian dan analisis hasil sistem pemodelan regresi menggunakan algoritma genetika untuk prediksi curah hujan. Pengujian dilakukan sesuai dengan perancangan pengujian yang telah dibuat.

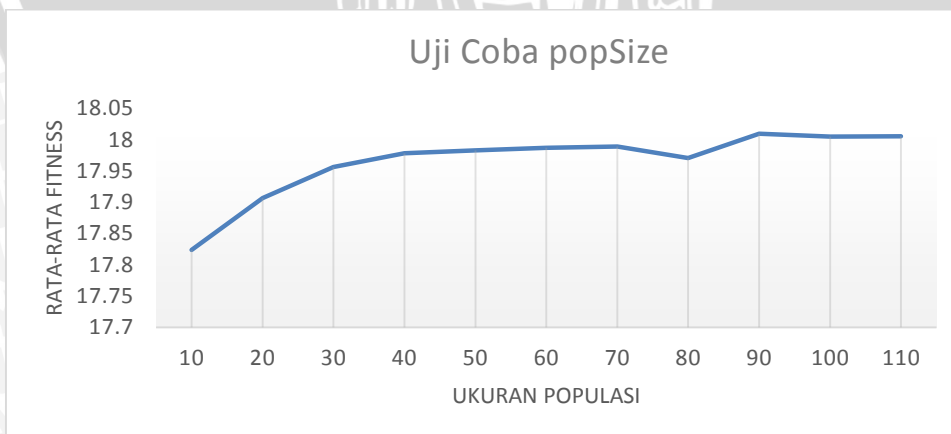
6.1 Hasil dan Analisa Uji Coba Ukuran Populasi

Uji coba yang pertama dilakukan adalah uji coba ukuran populasi terhadap perubahan nilai *fitness*. Pada pengujian ini jumlah generasi yang digunakan adalah 10 dengan *crossover rate* sebesar 0,6 dan *mutation rate* sebesar 0,4. Jumlah data yang digunakan adalah 30 dasarian. Setiap ukuran populasi dilakukan pengujian sebanyak 10 kali percobaan sehingga diperoleh nilai rata-rata *fitness*. Ukuran populasi diuji pada kelipatan 10 mulai dari 10 hingga 110, sehingga terdapat 11 ukuran *popSize* yang berbeda. Tabel 6.1 berikut merupakan hasil uji coba ukuran populasi.

PopSize	Nilai Fitness Percobaan ke-										Rata-rata Fitness
	1	2	3	4	5	6	7	8	9	10	
10	17.8045	17.8903	17.7813	17.7506	17.7111	17.8640	17.8386	17.8829	17.8565	17.8565	17.8236
20	18.0307	17.8640	17.8903	17.7504	17.8261	17.9880	17.8565	18.0307	17.9880	17.8410	17.9066
30	18.0307	17.8641	17.8641	17.9880	17.9880	17.9880	17.8904	18.0307	17.8904	18.0307	17.9565
40	17.8903	18.0307	18.0307	17.8565	17.9880	18.0307	17.8640	18.0307	18.0307	18.0307	17.9783
50	18.0307	17.8641	18.0307	17.9880	17.9880	17.9880	17.8904	17.9880	18.0307	18.0307	17.9829
60	18.0307	17.8904	17.9880	17.9880	18.0307	17.9880	18.0307	18.0307	17.8641	18.0307	17.9872
70	17.8641	18.0307	17.9880	18.0307	18.0307	18.0307	18.0307	17.8641	18.0307	17.9880	17.9888
80	17.8904	18.0307	17.9880	18.0307	18.0307	17.8904	17.8641	17.8830	17.9880	18.0307	17.9707
90	17.8641	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	17.9880	18.0307	18.0307	18.0098
100	18.0307	17.9880	18.0307	18.0307	17.8830	18.0307	18.0307	18.0307	18.0307	17.9880	18.0048
110	17.9880	18.0307	18.0307	17.9880	17.8904	18.0307	18.0307	18.0307	18.0307	17.9880	18.0056

Tabel 6.1 Hasil Pengujian Ukuran Populasi

Hasil uji coba ukuran populasi pada Tabel 6.1 diformulasikan kedalam bentuk grafik untuk mengetahui titik jumlah populasi optimal.



Gambar 6.1 Grafik Hasil Uji Coba Ukuran Populasi

Pada grafik dapat terlihat perbedaan hasil pengujian ukuran populasi terhadap nilai *fitness* yang dihasilkan. Dari hasil uji coba didapatkan bahwa ukuran populasi

30 adalah ukuran populasi yang dianggap optimal dengan rata-rata nilai *fitness* 17.9565 karena uji coba ukuran populasi diatas 30 tidak memberikan nilai *fitness* yang lebih baik karena nilai *fitness* yang dihasilkan cenderung stabil. Sehingga ukuran populasi 30 menjadi ukuran populasi optimal.

Besarnya nilai *fitness* sangat dipengaruhi oleh besarnya ukuran populasi. Semakin besar ukuran populasi maka nilai *fitness* yang dihasilkan juga semakin besar. Namun ukuran populasi yang besar akan mengakibatkan proses algoritma genetika berjalan lebih lama dan tidak memberikan solusi yang lebih baik. Oleh karena itu ukuran populasi maksimal pada uji coba ini adalah 110. Karena setelah batas ini nilai *fitness* yang dihasilkan tidak mengalami perubahan signifikan.

6.2 Hasil dan Analisa Uji Coba Jumlah Generasi

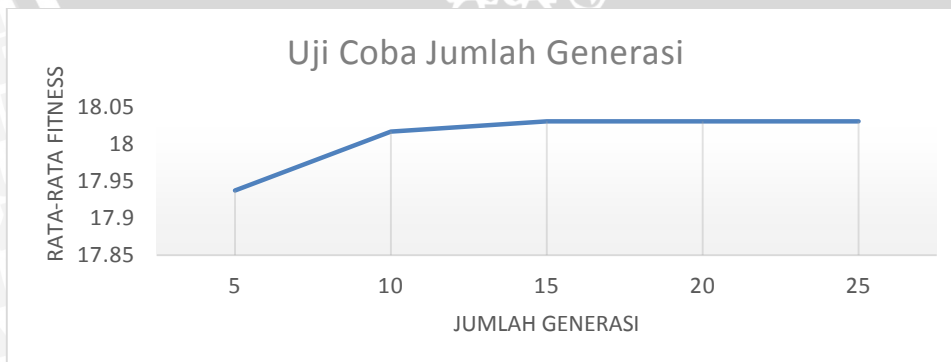
Setelah pengujian ukuran populasi dilakukan dan mendapatkan hasil maka pengujian selanjutnya yang dilakukan adalah pengujian jumlah generasi terhadap perubahan nilai *fitness*. Dari hasil pengujian jumlah generasi yang dilakukan sesuai perancangan didapatkan hasil pada Tabel 6.2 berikut. Generasi yang dilakukan uji coba adalah 5, 10, 15, 20, dan 25 generasi.

Pada setiap generasi dilakukan pengujian nilai *fitness* sebanyak 10 kali dan didapatkan rata-rata nilai *fitness*nya. Kemudian rata-rata *fitness* tersebut akan ditampilkan dalam bentuk grafik untuk mengetahui jumlah generasi yang optimal dengan ukuran populasi optimal sebelumnya yaitu 30, nilai *crossover rate* 0.6, nilai *mutation rate* 0.4, dan jumlah data yang digunakan adalah 30.

Tabel 6.2 Hasil Pengujian Jumlah Generasi

Generasi	Nilai Fitness Percobaan ke-										Rata-rata Fitness
	1	2	3	4	5	6	7	8	9	10	
5	17.9880	17.8566	17.8565	17.8641	18.0307	18.0307	18.0307	17.8641	17.9880	17.8641	17.9374
10	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	17.8904	18.0307	18.0307	18.0307	18.0167
15	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307
20	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307
25	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307

Rata-rata *fitness* hasil uji coba jumlah generasi akan diformulasikan dalam bentuk grafik yang ditunjukkan pada Gambar 6.2.



Gambar 6.2 Grafik Hasil Uji Coba Jumlah Generasi

Berdasarkan Gambar 6.2 jumlah generasi yang optimal didapatkan pada jumlah generasi 10 dengan nilai rata-rata *fitness* 18.0167. Dari pengujian ini dapat

diketahui bahwa jumlah generasi dapat mempengaruhi nilai *fitness* yang dihasilkan dari proses algoritma genetika. Jumlah generasi yang semakin banyak belum tentu akan memberikan solusi yang optimal karena semakin banyak jumlah generasi maka peluang terjadinya konvergensi semakin besar dan waktu yang dibutuhkan semakin lama. Jumlah generasi 10 menjadi solusi yang optimal karena tidak ditemukan solusi yang lebih baik setelah jumlah generasi ini karena rata-rata nilai *fitness* yang dihasilkan cenderung stabil.

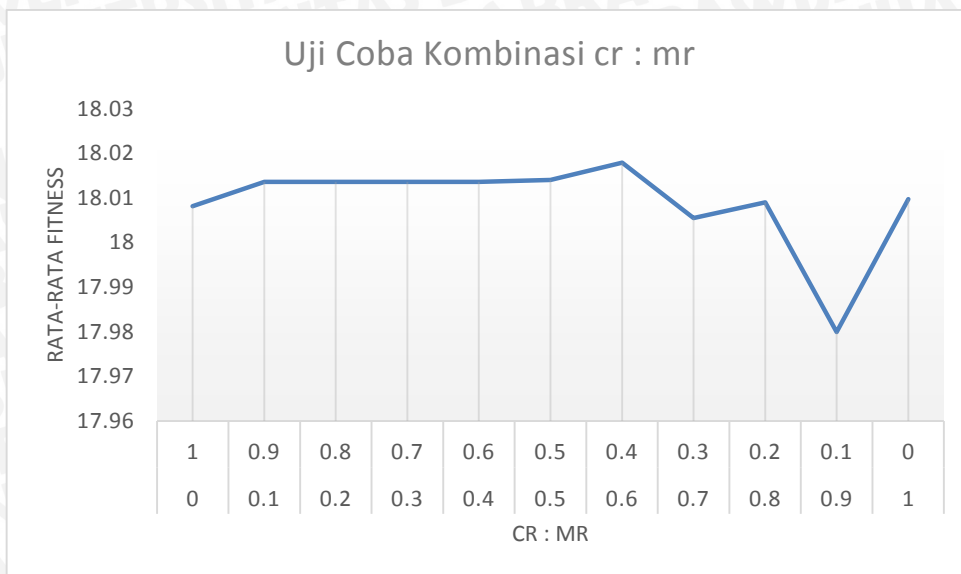
6.3 Hasil dan Analisa Uji Coba Kombinasi cr dan mr

Setelah melakukan pengujian ukuran populasi dan jumlah generasi dan didapatkan hasil optimalnya, maka pengujian selanjutnya adalah pengujian untuk mencari kombinasi yang optimal antara nilai cr dan mr.

Kombinasi		Nilai Fitness Percobaan ke-										Rata-rata
cr	mr	1	2	3	4	5	6	7	8	9	10	Fitness
0	1	18.0307	17.9880	18.0307	18.0307	18.0307	17.8904	18.0307	18.0307	18.0307	17.9880	18.0081
0.1	0.9	18.0307	17.9880	18.0307	17.9880	18.0307	18.0307	17.9880	18.0307	18.0307	17.9880	18.0136
0.2	0.8	18.0307	17.9880	18.0307	18.0307	18.0307	17.9880	17.9880	18.0307	18.0307	17.9880	18.0136
0.3	0.7	18.0307	18.0307	18.0307	17.9880	17.9880	17.9880	17.9880	18.0307	18.0307	18.0307	18.0136
0.4	0.6	18.0307	18.0307	17.9880	17.9880	18.0307	17.9880	18.0307	18.0307	17.9880	18.0307	18.0136
0.5	0.5	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	17.8641	18.0307	18.0307	18.0307	18.0140
0.6	0.4	17.9880	18.0307	18.0307	18.0307	17.9880	18.0307	17.9880	18.0307	18.0307	18.0307	18.0179
0.7	0.3	18.0307	18.0307	18.0307	17.9880	18.0307	17.9880	18.0307	18.0307	17.8641	18.0307	18.0055
0.8	0.2	18.0307	18.0307	18.0307	17.9880	18.0307	18.0307	18.0307	18.0307	17.8565	18.0307	18.0090
0.9	0.1	18.0307	18.0307	18.0307	17.8565	17.8641	18.0307	18.0307	17.8641	18.0307	18.0307	17.9800
1	0	17.9880	17.8641	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0307	18.0098

Tabel 6.3 Hasil Pengujian Kombinasi Nilai cr dan mr

Pada pengujian kombinasi cr dan mr, ukuran populasi yang digunakan adalah hasil pengujian sebelumnya yaitu 30 dan jumlah generasi 10. Sedangkan untuk jumlah data yang digunakan pada pengujian ini adalah 30 dasarian. Kombinasi *crossover rate* dan *mutation rate* yang digunakan dalam pengujian ini adalah kombinasi yang akan menghasilkan jumlah 1. Pada setiap kombinasi nilai cr dan mr dilakukan sepuluh kali percobaan untuk mencari nilai *fitness* yang dihasilkan dan kemudian dicari nilai rata-rata *fitness* tiap kombinasi.



Gambar 6.3 Grafik Hasil Uji Coba Kombinasi Nilai cr dan mr

Rata-rata nilai *fitness* yang dihasilkan oleh 10 kombinasi nilai cr dan mr ini menghasilkan nilai *fitness* yang cenderung stabil dan tidak memiliki perbedaan signifikan. Rata-rata *fitness* terendah didapatkan dari kombinasi *crossover rate* sama dengan 0.9 dan *mutation rate* sama dengan 0.1. Solusi optimal kombinasi nilai cr dan mr didapatkan dari kombinasi yang memiliki nilai *fitness* tertinggi. Dalam pengujian ini kombinasi yang menghasilkan *fitness* tertinggi sebesar 18.0179 adalah kombinasi *crossover rate* sebesar 0.6 dan *mutation rate* sebesar 0.4.

6.4 Hasil dan Analisa Uji Coba Jumlah Data yang Digunakan

Setelah pengujian ukuran populasi, pengujian jumlah generasi, pengujian kombinasi cr dan mr dilakukan maka didapatkan nilai parameter optimal yaitu :

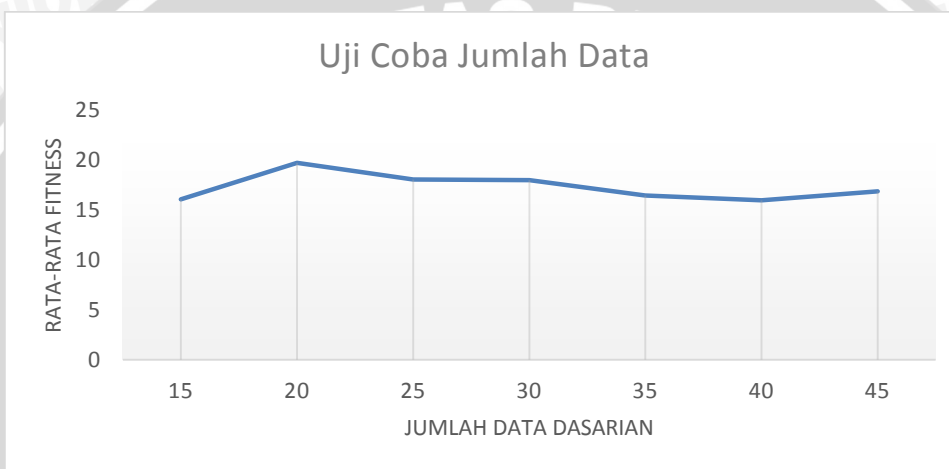
- *popSize* = 30
- Jumlah generasi = 10
- cr = 0.6
- mr = 0.4

Pengujian yang terakhir dilakukan adalah pengujian jumlah data yang digunakan dalam prediksi curah hujan terhadap perubahan nilai *fitness*. Nilai input parameter kemudian digunakan untuk melakukan pengujian jumlah data yang optimal untuk prediksi. Pengujian jumlah data dilakukan mulai dari 15 hingga 45 data. Semakin banyak data yang digunakan untuk memprediksi maka nilai *fitness* yang dihasilkan akan semakin kecil. Hal ini disebabkan karena ruang pencarian terlalu besar.

Tabel 6.4 Hasil Pengujian Jumlah Data

Jumlah Data	Nilai Fitness Percobaan ke-										Rata-rata	
	1	2	3	4	5	6	7	8	9	10	Fitness	
15	16.1014	16.1014	16.1014	16.1014	16.1014	16.1014	16.1014	16.1014	16.0205	16.1014	16.0435	16.0875
20	19.7485	19.7485	19.7485	19.7485	19.7394	19.7485	19.7485	19.7485	19.7485	19.7485	19.7485	19.7475
25	18.1425	17.9922	18.1425	17.9922	18.1425	17.9922	18.1425	18.1425	18.1425	18.1425	18.1425	18.0924
30	17.9880	17.9880	18.0307	17.9880	18.0307	18.0307	18.0307	17.9880	18.0307	18.0307	18.0307	18.0136
35	16.5007	16.3838	16.5007	16.5007	16.5007	16.5007	16.4143	16.5007	16.5007	16.5007	16.5007	16.4781
40	15.9926	16.0323	15.9926	15.9400	16.0323	15.9497	16.0323	16.0323	16.0323	16.0323	16.0323	16.0085
45	16.9037	16.9037	16.9037	16.9037	16.9037	16.9037	16.8674	16.9037	16.9037	16.9037	16.9037	16.9001

Uji coba dilakukan masing-masing 10 kali pada setiap jumlah data kemudian dihitung nilai rata-rata *fitness*nya yang kemudian digambarkan dalam bentuk grafik Gambar 6.4.



Gambar 6.4 Grafik Hasil Uji Coba Jumlah Data

Berdasarkan hasil pengujian yang digambarkan pada Gambar 6.4, didapatkan jumlah data dasar yang optimal digunakan untuk prediksi yaitu 20 data dasar dengan rata-rata fitness 19.7475.

6.5 Hasil dan Analisa Perbandingan Hasil Regresi Linear dengan Regresi Menggunakan Algoritma Genetika

Setelah pengujian dilakukan maka didapatkan parameter-parameter terbaik algoritma genetika yang mampu menghasilkan solusi untuk prediksi curah hujan. Untuk mengetahui apakah parameter tersebut dapat memberikan solusi optimal maka dilakukan analisa dan perbandingan terhadap perhitungan manual menggunakan Microsoft Excell. Dari serangkaian pengujian didapatkan parameter-parameter solusi terbaik yaitu :

- Ukuran populasi = 30
- Jumlah Generasi = 10
- *Crossover rate* = 0.6
- *Mutation rate* = 0.4
- Jumlah Data = 20.

Solusi yang dihasilkan oleh sistem menggunakan nilai input parameter akan dibandingkan dengan data curah hujan aktual. Dengan nilai input parameter hasil pengujian dan panjang kromosom 10, sistem algoritma genetika menampilkan individu dengan kromosom yang baik untuk prediksi yaitu menggunakan 8 periode dari 10 periode diantaranya dasarian ke-2,3,4,5,6,7,9 dan ke-10 kebelakang, selain itu sistem juga menghasilkan persamaan regresi linear.

$$Y' = 2.7735 + 1.1107 X_2 + 0.3085 X_3 + (-0.1375) X_4 + (-0.4658) X_5 + 0.2016 X_6 + 0.3303 X_7 + (-0.4180) X_9 + (-0.0244) X_{10}$$

Pada pengujian prediksi curah hujan dasarian dengan regresi linear berganda tanpa algoritma genetika jumlah data yang digunakan sama yaitu 20 serta panjang kromosom 10. Karena tidak menggunakan algoritma genetika maka semua kromosom yang ada digunakan. Hasil prediksi menggunakan regresi linear ini akan dijelaskan pada tabel berikut.

Tabel 6.5 Prediksi Regresi Linear

No	Bulan	Dasarian	Y	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Y'	Error
1	Juni	2	1.90	7.40	0.00	7.60	10.80	0.00	7.70	5.50	9.30	7.00	0.76	1.30	
2	Juni	3	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70	5.50	9.30	0.41	0.17	
3	Juli	1	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70	5.50	9.30	0.33	0.03
4	Juli	2	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70	5.50	2.41	0.27
5	Juli	3	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00	7.70	-0.46	0.21
6	Agustus	1	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80	0.00	1.12	1.24
7	Agustus	2	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60	10.80	-0.65	0.42
8	Agustus	3	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00	7.60	0.33	0.11
9	September	1	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.00	-1.12	1.26
10	September	2	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90	7.40	0.12	0.01
11	September	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00	1.90	2.74	7.53
12	Oktober	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50	0.00	2.52	6.36
13	Oktober	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90	0.50	1.79	3.22
14	Oktober	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.90	2.13	4.54
15	November	1	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.75	1.32
16	November	2	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.81	77.19
17	November	3	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.25	4.64
18	Desember	1	18.60	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16.79	3.27
19	Desember	2	10.10	18.60	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00	0.00	11.52	2.01
20	Desember	3	21.55	10.10	18.60	5.10	11.60	3.90	0.00	0.00	0.00	0.00	0.00	21.59	0.00
														jumlah Error	115.1134
														akar	10.7291
														MSE	0.5365
														fitness	19.5500

Koefisien yang digunakan untuk menghitung hasil prediksi curah hujan data Tabel 6.5 didapatkan dari perhitungan analisis data regresi yang ada pada Microsoft Excell.



Tabel 6.6 Koefisien Regresi Linear

Coefficients	
Intercept	2.749797
X Variable 1	0.016478
X Variable 2	1.106071
X Variable 3	0.289304
X Variable 4	-0.1372
X Variable 5	-0.45826
X Variable 6	0.213261
X Variable 7	0.324734
X Variable 8	-0.01011
X Variable 9	-0.41773
X Variable 10	-0.32524

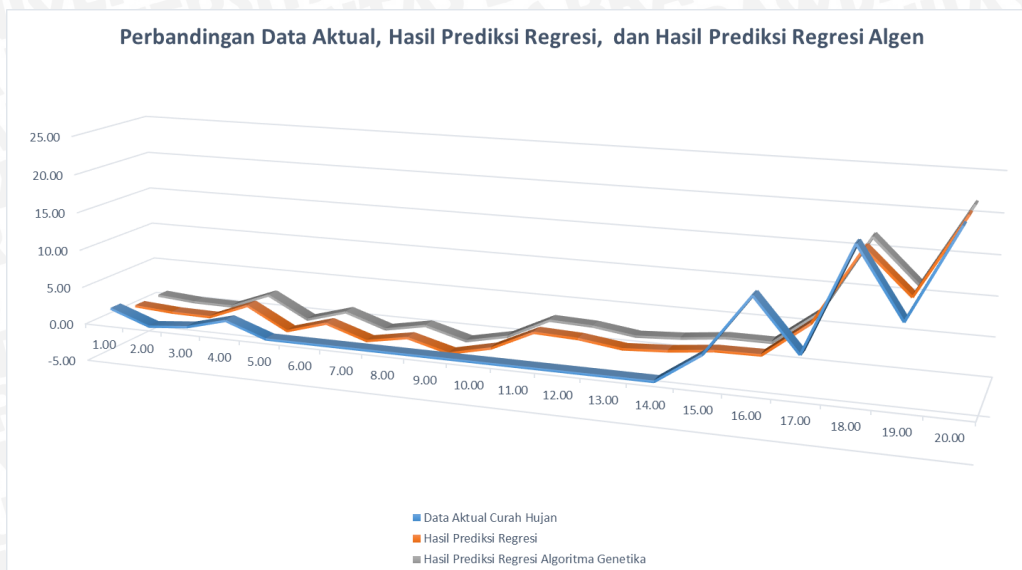
Perbandingan antara data aktual, hasil prediksi regresi, dan hasil prediksi regresi algoritma genetika akan dijelaskan pada Tabel 6.6.

Tabel 6.7 Perbandingan Prediksi Regresi dan Regresi Algoritma Genetika

No	Bulan	Dasarian	Y	Regresi Linear (10 periode)		Regresi Algoritma Genetika (8 periode)	
				Y'	Error	Y'	Error
1	Juni	2	1.90	0.76	1.30	0.79	1.24
2	Juni	3	0.00	0.41	0.17	0.38	0.14
3	Juli	1	0.50	0.33	0.03	0.33	0.03
4	Juli	2	1.90	2.41	0.27	2.39	0.24
5	Juli	3	0.00	-0.46	0.21	-0.43	0.19
6	Agustus	1	0.00	1.12	1.24	1.13	1.28
7	Agustus	2	0.00	-0.65	0.42	-0.65	0.43
8	Agustus	3	0.00	0.33	0.11	0.38	0.14
9	September	1	0.00	-1.12	1.26	-1.10	1.22
10	September	2	0.00	0.12	0.01	0.07	0.00
11	September	3	0.00	2.74	7.53	2.77	7.67
12	Oktober	1	0.00	2.52	6.36	2.56	6.58
13	Oktober	2	0.00	1.79	3.22	1.81	3.29
14	Oktober	3	0.00	2.13	4.54	2.14	4.59
15	November	1	3.90	2.75	1.32	2.77	1.27
16	November	2	11.60	2.81	77.19	2.77	77.91
17	November	3	5.10	7.25	4.64	7.11	4.02
18	Desember	1	18.60	16.79	3.27	16.86	3.02
19	Desember	2	10.10	11.52	2.01	11.48	1.91
20	Desember	3	21.55	21.59	0.00	21.60	0.00
				Jumlah Error	115.11	Jumlah Error	115.16
				akar	10.73	akar	10.73
				MSE	0.54	MSE	0.54
				fitness	19.55	Fitness	19.748582

Pada Tabel 6.7 ditampilkan perbandingan antara prediksi curah hujan dasarian menggunakan regresi linear dengan panjang kromosom 10 dan prediksi menggunakan algoritma genetika dengan panjang kromosom 8. Nilai *error* didapatkan dari hasil pengkuadratan selisih data asli dan data hasil prediksi. Prediksi yang dilakukan dengan menggunakan 8 periode memiliki kecenderungan untuk digunakan daripada prediksi menggunakan 10 periode karena dengan variabel yang lebih sedikit, prediksi yang dilakukan dapat menghasilkan *fitness* yang optimal.





Gambar 6.5 Grafik Perbandingan Hasil Prediksi

Gambar 6.5 menunjukkan bahwa hasil prediksi yang dihasilkan oleh proses perhitungan manual regresi, hasil prediksi sistem algoritma genetika dan data asli memiliki selisih yang tidak terlalu jauh. Penentuan periode yang digunakan untuk memprediksi memiliki pengaruh terhadap nilai *fitness* yang dihasilkan. Dengan penentuan periode yang diwakili oleh kromosom/gen sebelum melakukan prediksi regresi, nilai *fitness* yang dihasilkan lebih tinggi. Pada prediksi regresi linear menggunakan algoritma genetika dengan menggunakan 8 periode, mampu memberikan solusi yang lebih baik daripada prediksi regresi linear dengan 10 periode karena semakin sedikit jumlah variabel yang digunakan maka waktu komputasi yang diperlukan semakin cepat.



BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan yang didapatkan dari pengujian yang telah dilakukan pada pemodelan regresi menggunakan algoritma genetika untuk prediksi curah hujan adalah :

1. Pemodelan regresi linear dapat diterapkan menggunakan algoritma genetika untuk mendapatkan solusi periode yang digunakan dalam memprediksi. Dengan menggunakan representasi kromosom biner sepanjang 10, penentuan digunakan atau tidaknya suatu periode dalam prediksi diwakili oleh nilai 1 dan 0. Apabila kromosom/gen bernilai 0, maka data pada gen tersebut tidak digunakan untuk prediksi, sedangkan apabila kromosom bernilai 1 maka data digunakan untuk melakukan prediksi. Jumlah periode yang digunakan untuk prediksi pada suatu individu akan berpengaruh pada nilai *fitness* yang dihasilkan.
2. Dari pengujian yang telah dilakukan maka didapatkan nilai parameter-parameter dalam algoritma genetika yang mampu memberikan hasil atau solusi yang dianggap optimal. Pada sistem pemodelan regresi menggunakan algoritma genetika untuk prediksi curah hujan, solusi ukuran populasi terbaik adalah 30 *popSize*, solusi jumlah generasi terbaik adalah 10, solusi kombinasi nilai *crossover rate* (*cr*) 0.6 dan *mutation rate* (*mr*) 0.4, serta solusi jumlah data yang digunakan adalah 20 data dasarian. Semakin besar jumlah data yang digunakan untuk melakukan prediksi maka solusi prediksi yang dihasilkan memiliki selisih jauh dengan data aktual curah hujan dan nilai *fitness*nya semakin kecil.
3. Berdasarkan nilai *fitness* yang dihasilkan pemodelan regresi menggunakan algoritma genetika memberikan solusi yang lebih baik daripada pemodelan regresi tanpa algoritma genetika. Nilai *fitness* yang dihasilkan dari pemodelan regresi tanpa algoritma genetika adalah 19.55 dengan nilai MSE (*Mean Square Error*) 0.54. Sedangkan nilai *fitness* yang dihasilkan dari pemodelan regresi menggunakan algoritma genetika adalah 19.74 dengan nilai MSE 0.54. Nilai MSE keduanya sama namun nilai *fitness* yang dihasilkan berbeda karena panjang kromosomnya berbeda.

7.2 Saran

Berdasarkan kesimpulan maka saran untuk penelitian selanjutnya adalah :

1. Melakukan penelitian prediksi curah hujan dengan pemodelan regresi dan algoritma genetika menggunakan panjang kromosom yang lain.
2. Menggunakan metode-metode lain yang digunakan dalam proses reproduksi algoritma genetika dan menggunakan representasi kromosom selain representasi kromosom biner.

DAFTAR PUSTAKA

- Aji, B. A. S., Adiwijaya., Nhita, F. 2014. *Prediksi Curah Hujan Menggunakan Evolving Fuzzy*. Bandung : Universitas Telkom Bandung.
- Apriyanti, N. 2005. *Optimasi Jaringan Syaraf Tiruan Dengan Algoritma Genetika Untuk Peramalan Curah Hujan*. S1. Institut Pertanian Bogor.
- Azkiya, F. 2015. *Pemodelan Persamaan Regresi Linier Berganda Dalam Memprediksi Berat Badan Ideal Menggunakan Algoritma Genetika*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, 6.
- Fadholi, Akhmad. 2013. *Persamaan Regresi Prediksi Hujan Bulanan di Pontianak dengan Prediktor Suhu dan Kelembapan Udara*. Pangkalpinang : Stasiun Meteorologi Depati Amir.
- Hazaki, Hafid. 2011. *Aplikasi untuk Prediksi Jumlah Mahasiswa Pengambil Mata Kuliah dengan Menggunakan Algoritma Genetika (Studi Kasus : Jurusan Teknik Informatika ITS)*. Surabaya : Institut Teknologi Surabaya.
- Indrabayu, Harum, N., Pallu, M. S., Achmad, A., Febriyati, F. 2012. *Prediksi Curah Hujan dengan Fuzzy Logic*. Makassar: Universitas Hasanuddin.
- Iriany, Atiek., Mahmudy, Wayan Firdaus., Handoyo, Samingun. 2015. *GSTAR-SUR Model for Rainfall Forecasting in Tengger Region, East Java*. Malang. In: The 1st UMM International Pure and Applied Research (UMM IcoPar).
- Kurniawan, Deny.2008. *Regresi Linear (Linear Regression)*.Vienna, Austria: R. Development Core Team.
- Mahmudy, Wayan Firdaus. 2013. *Algoritma Evolusi*. Program Teknologi Informasi dan Ilmu Komputer. Malang : Universitas Brawijaya.
- Mukid, Moch. Abdul., Sugito. 2013. *Model Prediksi Curah Hujan dengan Pendekatan Regresi Proses Gaussian*. Semarang : Universitas Diponegoro.
- Permatasari, A. I. & Mahmudy, W. F. 2015. *Pemodelan Regresi Linear dalam Konsumsi Kwh Listrik di Kota Batu Menggunakan Algoritma Genetika*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, 5.
- Purbaya, Rifal. 2014. *Perbandingan Analisis Regresi Linier Berganda dengan Sistem Inferensi Fuzzy Mamdani dalam Memprediksi Berat Badan Ideal*. Malang : Universitas Brawijaya.
- Rahmi, A.&Mahmudy, W. F. 2015. *Prediksi Harga Saham Berdasarkan Data Historis Menggunakan Model Regresi yang Dibangun Dengan Algoritma Genetika*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, 5.
- Saputro, Dewi Retno Sari. 2009. *Memprediksi Curah Hujan (Data Spatio-Temporal) dengan Metode Bayesian Networks*. Prosiding Seminar Nasional Penelitian, Pendidikan dan Penerapan MIPA Universitas Negeri Yogyakarta.

Tukidi, 2010. *Karakter Curah Hujan di Indonesia*. Jurusan Geografi FIS UNNES.

