

**PEMBENTUKAN MODEL REGRESI ARIMA UNTUK
MELAKUKAN PREDIKSI CURAH HUJAN MENGGUNAKAN
ALGORITMA GENETIKA**

SKRIPSI

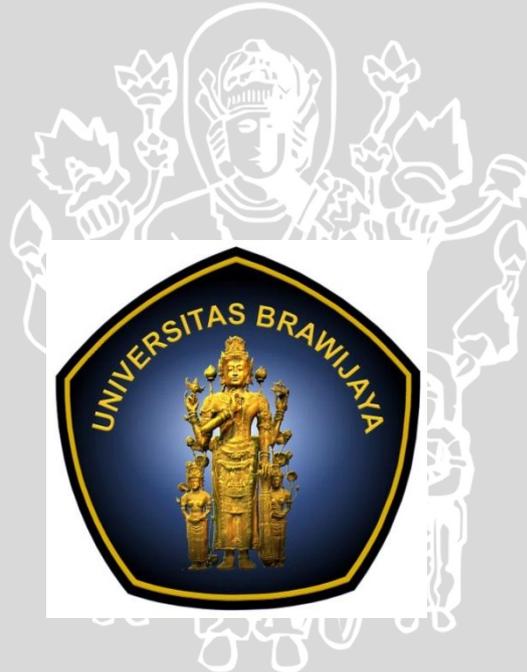
Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Sukma Wardana Hadi Putra

NIM: 125150207111007

UNIVERSITAS BRAWIJAYA



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

PENGESAHAN

PEMBENTUKAN MODEL REGRESI ARIMA UNTUK MELAKUKAN PREDIKSI CURAH
HUJAN MENGGUNAKAN ALGORITMA GENETIKA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Sukma Wardana Hadi Putra
NIM: 125150207111007

Skripsi ini telah diuji dan dinyatakan lulus pada
10 Maret 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Wayan Firdaus Mahmudy, S.Si., M.T., Ph.d.
NIP: 197209191997021001

Mengetahui
Ketua Program Studi Informatika

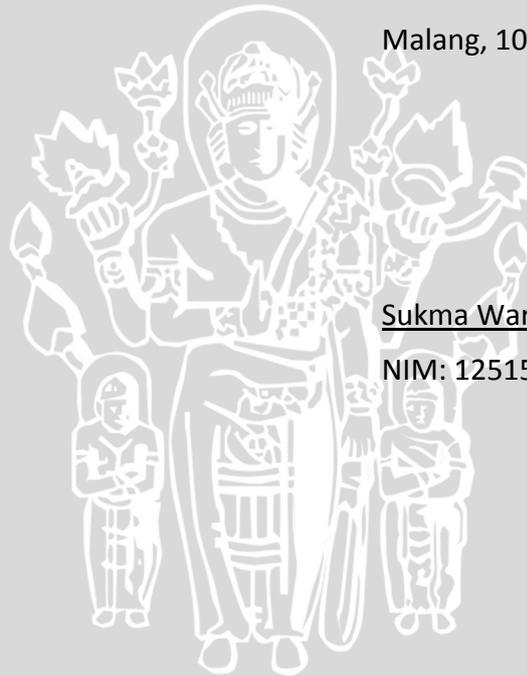
Drs. Marji, M.T.
NIP: 196708011992031001

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, 10 Maret 2016



Sukma Wardana Hadi Putra

NIM: 125150207111007

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat, taufik, serta hidayahnya sehingga penulis mampu menyelesaikan penelitian mengenai Membentuk Model Regresi ARIMA untuk Memprediksi Curah Hujan dengan Algoritma Genetika sebagai salah satu kegiatan akhir sebelum mendapatkan gelar Sarjana Komputer di Program Teknologi Informasi dan Ilmu Komputer (PTIIK), Universitas Brawijaya Malang. Atas keberhasilan dalam penelitian ini, penulis mengucapkan terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., selaku dosen pembimbing yang telah banyak membantu penulis selama penelitian ini dari awal hingga akhir.
2. Segenap bapak dan ibu dosen PTIIK yang selama penulis menempuh pendidikan telah banyak memberikan ilmu serta kemampuan yang menjadikan penulis lebih baik dari sebelumnya.
3. Staff dan karyawan PTIIK yang telah membantu penulis selama menempuh pendidikan serta membantu dalam urusan *administrative* sehingga penelitian ini berjalan dengan baik dari awal hingga akhir.

Penulis menyadari hasil penelitian yang telah penulis lakukan jauh dari sempurna dan masih terdapat banyak kekurangan. Sehingga, penulis mengharapkan kritik dan saran dari berbagai pihak yang dapat menjadikan penelitian ini lebih baik lagi. Penulis berharap hasil penelitian ini memberi manfaat bagi sesama, terutama kepada pihak yang terkait dalam disiplin ilmu dengan penelitian ini.

Malang, 10 Maret 2016

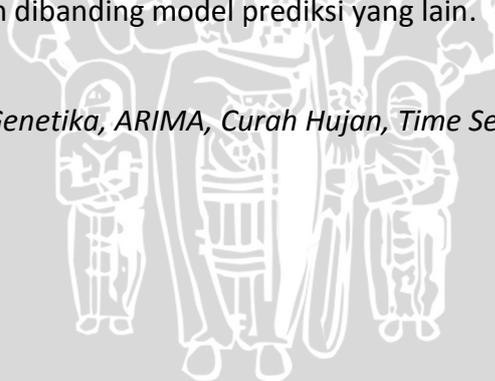
Penulis

sukma.wardana@mail.com

ABSTRAK

Curah hujan merupakan salah satu faktor utama dalam terjadinya perubahan iklim. Indonesia dengan karakteristiknya memiliki fenomena iklim yang cukup kompleks, hal ini diperparah dengan adanya pemanasan global. Sektor pertanian sangat bergantung terhadap pola distribusi curah hujan yang baik, sehingga kebutuhan akan prediksi curah hujan sangatlah diperlukan. Data curah hujan berupa data *time series* yang tidak stasioner, oleh karena itu digunakan metode *Autoregressive Integrated Moving Average* (ARIMA) yang terkenal mampu digunakan dalam data *time series* yang tidak stasioner. Dengan algoritma genetika yang membentuk orde model pada ARIMA, dapat dihasilkan sebuah model dengan tingkat prediksi yang sangat baik. Dalam penggunaan algoritma genetika, representasi kromosom menggunakan *binary*, proses *crossover* yang digunakan adalah *two-cut-point*, *random mutation* untuk proses mutasi, serta proses seleksi menggunakan metode *elitism selection*. Hasil penelitian memberikan nilai parameter terbaik yakni ukuran populasi sebanyak 120, jumlah generasi sebesar 500, kombinasi $cr : mr$ berturut-turut adalah 0.5 : 0.5, dengan periode data curah hujan terbaik sebanyak 12. Model ARIMA terbaik dibuktikan dengan nilai MAPE terkecil 1.462213 dengan banyak periode data yang digunakan adalah 6 dari 12 periode data. Sehingga dapat disimpulkan model ARIMA yang dibangun dengan algoritma genetika mampu untuk memprediksi data curah hujan dengan tingkat akurasi yang baik serta dengan usaha yang lebih mudah dibanding model prediksi yang lain.

Kata kunci: *Algoritma Genetika, ARIMA, Curah Hujan, Time Series*



ABSTRACT

Rainfall is one of the main factors in the climate change. Indonesia has a climate phenomenon with characteristics that are quite complex, it is exacerbated by global warming. The agricultural sector is highly dependent on the distribution pattern of rainfall, so the need for rainfall prediction is required. Rainfall data mostly in time series and not stationary, Autoregressive Integrated Moving Average (ARIMA) is famous and capable of being used in time series data are not stationary. With genetic algorithms that form the ARIMA model order, can produce a model with a very good level of prediction. In the use of genetic algorithm, the chromosomes using binary representation, the crossover process is two-cut-point, random mutation in the process of mutation, the selection process is elitism selection. The results of the research provide the best parameter for the size of population is 120, the best generation was 500, with the best combination of $cr : mr$ 0.5 : 0.5 as well as the best rainfall data is 12 periods. The best ARIMA models that given by genetic algorithm have MAPE value is 1.462212 with using 6 periods data of 12 periods data. Therefore, we can conclude that ARIMA model was built with genetic algorithm was able to predict the rainfall with a good degree of accuracy as well as the effort much easier rather than other predictive models.

Keywords: Genetic Algorithm, ARIMA, Rainfall, Time Series



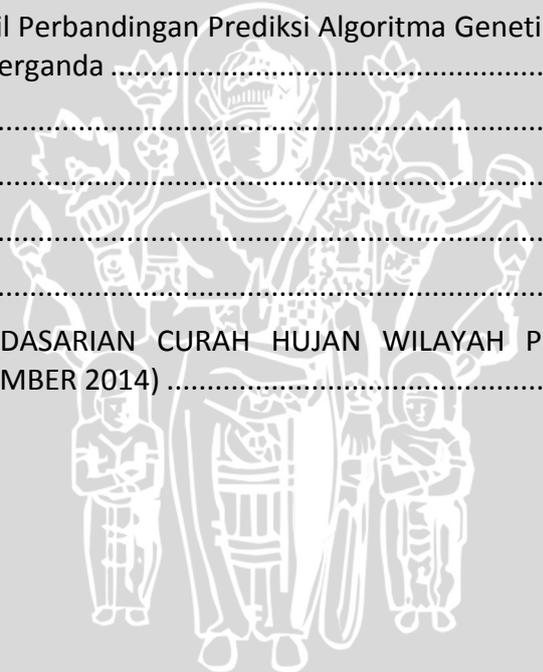
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR SOURCE CODE	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Curah Hujan	7
2.3 Data <i>Time Series</i>	7
2.3.1 Klasifikasi Model ARIMA	8
2.3.2 <i>Box and Jenkins Methodology</i>	9
2.4 Algoritma Genetika	12
2.4.1 Tahap Inisialisasi.....	13
2.4.2 Tahap Reproduksi.....	13
2.4.3 Nilai <i>Fitness</i>	14
2.4.4 Tahap Seleksi.....	14
2.5 Parameter Algoritma Genetika.....	15
BAB 3 METODOLOGI PENELITIAN	16

3.1 Tahapan Penelitian	16
3.2 Teknik Pengumpulan Data	16
3.3 Algoritma yang Digunakan.....	16
3.4 Kebutuhan Sistem.....	17
3.4.1 Kebutuhan Perangkat Keras.....	17
3.4.2 Kebutuhan Perangkat Lunak	17
3.5 Pengujian Algoritma.....	18
3.5.1 Uji Coba Ukuran Populasi.....	18
3.5.2 Uji Coba Banyaknya Generasi	19
3.5.3 Uji Coba Kombinasi cr dan mr.....	19
3.5.4 Uji Coba Banyaknya Periode Curah Hujan	20
BAB 4 ANALISIS DAN PERANCANGAN SISTEM	22
4.1 Formulasi Permasalahan.....	22
4.2 Siklus Pembentukan Model ARIMA dengan Algoritma Genetika.....	27
4.3 Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika	28
4.3.1 Representasi Kromosom dan Perhitungan Fitness	29
4.3.2 Inialisasi Populasi Awal	31
4.3.3 Reproduksi	31
4.3.4 Evaluasi dan Seleksi.....	32
4.4 Perancangan <i>User Interface</i>	34
4.4.1 Tampilan Halaman Utama.....	34
4.4.2 Tampilan Halaman Algoritma Genetika	35
4.4.3 Tampilan Halaman Prediksi.....	36
BAB 5 IMPLEMENTASI	37
5.1 Struktur Class	37
5.1.1 Class Algoritma Genetika	37
5.1.2 Class Regresi ARIMA.....	38
5.2 Source Code	39
5.2.1 Mengambil Data <i>time series</i> Curah Hujan	39
5.2.2 Pembangkitan Populasi Awal	40
5.2.3 Perhitungan <i>Fitness</i>	40
5.2.4 Proses <i>Crossover</i>	41



5.2.5 Proses Mutasi	42
5.2.6 Proses Seleksi dengan Metode Elitism	43
5.2.7 Proses Pemilihan Kromosom Terbaik	43
5.3 Implementasi	44
5.3.1 Halaman Utama	44
5.3.2 Halaman Hasil Algoritma Genetika	45
BAB 6 PEMBAHASAN	46
6.1 Uji Coba & Analisis Parameter Populasi	46
6.2 Uji Coba & Analisis Parameter Generasi.....	48
6.3 Uji Coba & Analisis Parameter Kombinasi cr & mr	50
6.4 Uji Coba & Analisis Periode Data Curah Hujan	52
6.5 Analisa Hasil Perbandingan Prediksi Algoritma Genetika dengan Model Regresi Linier Berganda	54
BAB 7 PENUTUP	56
7.1 Kesimpulan.....	56
7.2 Saran	57
DAFTAR PUSTAKA	58
LAMPIRAN A DATA DASARIAN CURAH HUJAN WILAYAH PUSPO, TENGER (JANUARI 2014 – DESEMBER 2014)	60



DAFTAR TABEL

Tabel 2.1 Perbandingan dengan Penelitian Sebelumnya	5
Tabel 3.1 Perancangan uji coba ukuran populasi	18
Tabel 3.2 Perancangan uji coba banyaknya generasi	19
Tabel 3.3 Perancangan uji coba kombinasi cr dan mr	20
Tabel 3.4 Perancangan uji coba banyak periode data curah hujan.....	21
Tabel 4.1 Data curah hujan dasarian tahun 2014 wilayah Puspo, Tengger.....	22
Tabel 4.2 Hasil <i>differencing</i> data.....	23
Tabel 4.3 Data <i>time series</i> curah hujan 5 periode	24
Tabel 4.4 Paramater AR dan MA model ARIMA	25
Tabel 4.5 Perbandingan prediksi dengan model arima terhadap data asli.....	26
Tabel 4.6 Pembentukan Kromosom <i>Parent 1</i>	29
Tabel 4.7 Persamaan Regresi ARIMA Tiap Data.....	29
Tabel 4.8 Hasil Perhitungan Prediksi Curah Hujan dan Nilai <i>Error</i>	30
Tabel 4.9 Hasil Perhitungan <i>Fitness Parrent 1</i>	31
Tabel 4.10 Data populasi awal	31
Tabel 4.11 Hasil <i>two cut point cross over</i>	32
Tabel 4.12 Hasil <i>random mutation</i>	32
Tabel 4.13 Transformasi kromosom biner menjadi model ARIMA	33
Tabel 4.14 Hasil nilai <i>fitness</i> dari seluruh kromosom	33
Tabel 4.15 Hasil seleksi <i>elitism selection</i>	34
Tabel 6.1 Hasil Uji Coba Parameter Populasi	46
Tabel 6.2 Hasil Uji Coba Parameter Generasi	48
Tabel 6.3 Hasil Uji Coba Kombinasi Parameter cr dan mr	50
Tabel 6.4 Hasil Uji Coba Periode Data Curah Hujan.....	53
Tabel 6.5 Hasil Perbandingan dengan Regresi Linier Berganda.....	54

DAFTAR SOURCE CODE

Source Code 5.1 Proses Membaca Data	40
Source Code 5.2 Membangkitkan Kromosom Biner	40
Source Code 5.3 Melakukan Inisialisasi Populasi Awal Sebesar popSize.....	40
Source Code 5.4 Proses Mencari Nilai MAPE.....	41
Source Code 5.5 Proses Menghitung <i>fitness</i>	41
Source Code 5.6 <i>Crossover</i> dengan Metode <i>Two-Cut-Point</i>	42
Source Code 5.7 Proses Pertukaran Gen pada Kromosom.....	42
Source Code 5.8 Mutasi dengan Metode <i>RandomMutation</i>	43
Source Code 5.9 Proses Mutasi Gen pada Kromosom	43
Source Code 5.10 Proses Seleksi dengan Metode <i>ElitismSelection</i>	43
Source Code 5.11 Proses Pemilihan Krosomom Terbaik.....	43



DAFTAR GAMBAR

Gambar 2.1 Flowchart ARIMA	9
Gambar 2.2 Pengecekan data model ARIMA.....	11
Gambar 4.1 Tahapan alur sistem	28
Gambar 4.2 Rancangan halaman utama.....	35
Gambar 4.3 Rancangan halaman proses algoritma genetika-ARIMA.....	35
Gambar 4.4 Rancangan halaman prediksi dengan model ARIMA terbaik.....	36
Gambar 5.1 <i>Class Diagram</i> Algoritma Genetika.....	38
Gambar 5.2 <i>Class Diagram</i> Model Regresi ARIMA.....	39
Gambar 5.3 Halaman Awal dari <i>Prototype</i>	44
Gambar 5.4 Halaman Hasil Proses Algoritma Genetika.....	45
Gambar 6.1 Grafik Pengujian Parameter Populasi	47
Gambar 6.2 Grafik Pengujian Parameter Generasi.....	49
Gambar 6.3 Grafik Pengujian Kombinasi Parameter c_r & m_r	52
Gambar 6.4 Grafik Pengujian Periode Data Curah Hujan	53
Gambar 6.5 Grafik Hasil Perbandingan Prediksi Curah Hujan	55

DAFTAR LAMPIRAN

LAMPIRAN A DATA DASARIAN CURAH HUJAN WILAYAH PUSPO, TENGGER
(JANUARI 2014 – DESEMBER 2014) 60



BAB 1 PENDAHULUAN

Bab ini membahas mengenai latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan skripsi.

1.1 Latar belakang

Sebagai negara yang terdiri dari pulau-pulau besar dan kecil, Indonesia memiliki kerentanan dalam perubahan iklim, dimana curah hujan merupakan salah satu faktor utama dalam terjadinya perubahan iklim (Mirawati, et al., 2013). Selain sebagai faktor perubahan iklim, curah hujan juga memiliki pengaruh terhadap beberapa sektor yang cukup penting bagi kehidupan manusia. Sebagai contoh, dalam sektor komunikasi curah hujan dapat mengakibatkan redaman hujan terhadap gelombang radio dengan frekuensi diatas 10 GHz (Mauludiyanto, et al., 2009). Pertanian juga merupakan salah satu sektor yang sangat bergantung terhadap curah hujan (Estiningtyas, et al., 2007) hal ini diperjelas oleh penelitian Iriany, et al., (2015) yang menyebutkan produktifitas pertanian kentang di wilayah Tengger, Bromo mengalami kesulitan dikarenakan perubahan iklim yang tidak menentu.

Dengan berbagai karakteristik yang ada, negara Indonesia memiliki fenomena iklim yang cukup kompleks. Sulitnya mendapatkan prediksi mengenai curah hujan dengan baik, diperparah dengan adanya pemanasan global yang menyebabkan perubahan musim di Indonesia semakin tidak teratur (Indrabayu, et al., 2011). Hal tersebut memicu banyaknya penelitian mengenai data curah hujan (Mirawati, et al., 2013) sehingga semakin banyak penelitian mengenai bagaimana memprediksi data curah hujan. Seperti penelitian yang dilakukan oleh Iriany, et al., (2015) yang menerapkan *Generalized Model Space Time Autoregressive* (GSTAR) untuk memprediksi curah hujan di wilayah Tengger, Bromo. Selain itu tercatat juga penelitian dalam melakukan prediksi curah hujan dengan menggunakan metode *Kalman Filter* pada Kota Semarang (Mirawati, et al., 2013), kemudian penelitian oleh Indrabayu, et al., (2011) mengenai penggunaan metode *Wavelet-Neural Network* untuk memprediksi curah hujan di wilayah Makasar.

Teknik peramalan *time series* merupakan salah satu metode yang aktif dikembangkan hingga saat ini (Wiyanti & Pulungan, 2012). Hal ini juga selaras bahwa data curah hujan bersifat *time series* hal ini dibuktikan oleh Iriany, et al., (2015) dan Mirawati, et al., (2013) dimana data yang mereka gunakan berupa data *time series*. Selain bersifat *time series* data curah hujan juga memiliki kecenderungan bersifat tidak stasioner (Mauludiyanto, et al., 2009). Salah satu metode yang populer digunakan dalam prediksi dengan data *time series* adalah *Autoregressive Integrated Moving Average* (ARIMA).

ARIMA sendiri terkenal mampu memprediksi dengan data *time series* yang tidak stasioner, beberapa penelitian diantaranya mengenai penggunaan ARIMA dalam memprediksi rasio kematian ibu melahirkan menyatakan model ARIMA menghasilkan prediksi yang akurat (Sarpong, 2013). Mauludiyanto, et al., (2009) dalam penelitiannya juga menyebutkan keunggulan ARIMA dalam memprediksi data *time series* yang tidak stasioner adalah sangat baik. Namun terdapat pernyataan bahwa ARIMA merupakan model konvensional dan *inferior* terhadap metode kecerdasan buatan dalam melakukan prediksi (Crone, et al., 2006).

Crone, et al., (2006) dalam penelitiannya mengenai penggunaan kecerdasan buatan dalam melakukan prediksi terhadap data *time series* membandingkan metode kecerdasan buatan dengan statistika seperti ARIMA dan menyatakan meski memiliki keunggulan dalam poin masing-masing, ARIMA masih kalah terhadap metode kecerdasan buatan. Bahkan terdapat penelitian mengenai perbandingan penggunaan *support vector regression* (SVR) dengan algoritma genetika (GA-SVR) terhadap algoritma *Back Propagation* (BPN) dan ARIMA dalam melakukan prediksi jumlah kedatangan turis memberikan hasil dimana disebutkan GA-SVR memberikan hasil yang lebih baik (Chen & Wang, 2007).

Algoritma genetika merupakan salah satu cabang ilmu dari *evolutionary algorithm* dengan konsep menirukan rangkaian evolusi yang terjadi pada makhluk hidup sehingga mampu menyelesaikan permasalahan dengan tingkat kesulitan yang cukup tinggi. Algoritma genetika populer digunakan dalam melakukan prediksi, hal ini dibuktikan dalam penelitian mengenai prediksi harga saham (Rahmi, et al., 2015) dan prediksi penggunaan konsumsi listrik (Permatasari & Mahmudy, 2015). Selain itu algoritma genetika juga mampu digunakan untuk memprediksi data *time series*, hal ini dibuktikan dengan penelitian mengenai prediksi temperatur menggunakan kombinasi BPN dan algoritma genetika (Singh, et al., 2011).

Terdapat penelitian yang cukup menarik terkait algoritma genetika dan ARIMA, dimana metode ARIMA dapat dibentuk dengan algoritma genetika sehingga lebih *powerfull* dalam melakukan prediksi data *time series*. Hal ini dibuktikan oleh penelitian yang menyebutkan penggunaan ARIMA dengan algoritma genetika jauh lebih baik dibandingkan hanya menggunakan ARIMA saja (Maarof, et al., 2014). Selain itu terdapat pula penelitian mengenai membentuk model ARIMA dengan algoritma genetika dengan hasil yang menyatakan bahwa algoritma genetika mampu membentuk model ARIMA yang akurat digunakan dalam melakukan prediksi data secara aktual (Ong, et al., 2005).

Berdasar penelitian terdahulu, dimana model ARIMA merupakan metode prediksi data *time series* yang tidak stasioner terbaik ditambah dengan pernyataan oleh Ong, et al., (2005) bahwa ARIMA yang dibentuk dengan algoritma genetika mampu menghasilkan model ARIMA yang dapat digunakan untuk memprediksi data secara aktual maka penelitian mengenai membentuk model ARIMA untuk prediksi curah hujan dengan algoritma genetika sangat penting untuk dilakukan.

1.2 Rumusan masalah

Berdasarkan penjelasan mengenai latar belakang di atas, maka muncul permasalahan terhadap penelitian ini seperti berikut:

1. Bagaimana membangun model ARIMA dengan algoritma genetika?
2. Bagaimana menentukan model terbaik dari ARIMA yang dibentuk dengan algoritma genetika?
3. Bagaimana menentukan nilai parameter algoritma genetika yang tepat?
4. Bagaimana menguji model ARIMA yang dibangun dengan algoritma genetika?

1.3 Tujuan

Tujuan dari penelitian ini dilakukan agar dapat menghasilkan luaran seperti berikut:

1. Agar dapat membangun model regresi ARIMA dengan menggunakan algoritma genetika.
2. Agar dapat menentukan parameter algoritma genetika terbaik untuk membentuk model ARIMA.
3. Agar dapat mengetahui tingkat akurasi model ARIMA yang dibentuk dengan algoritma genetika.

1.4 Manfaat

Manfaat dari penelitian ini adalah agar didapatkannya model ARIMA terbaik yang dapat digunakan untuk melakukan prediksi khususnya prediksi data curah hujan berbasis *time series*. Sehingga metode prediksi akan mengalami peningkatan akurasi atau kemudahan dalam sisi komputasi.

1.5 Batasan masalah

Penelitian ini memiliki batasan agar dapat memfokuskan terhadap permasalahan serta tidak meluas, maka batasan terhadap penelitian ini sebagai berikut:

1. Pemodelan regresi yang digunakan adalah model ARIMA
2. Penentuan parameter model ARIMA menggunakan metode *ordinary least square* dan dalam *system prototype* menggunakan *library* pihak ketiga.
3. Metode yang dipakai adalah algoritma genetika yang dimulai dengan proses inisialisasi, reproduksi, evaluasi, dan seleksi.
4. Data yang digunakan merupakan data curah hujan dasarian yang tidak stasioner, merupakan data hasil penelitian yang telah dilakukan sebelumnya oleh Iriany, et al., (2015) dalam melakukan prediksi curah hujan dengan model GSTAR-SUR di wilayah Tengger, Bromo.

1.6 Sistematika pembahasan

Sistematika penulisan dalam skripsi sebagai berikut:

BAB 1 PENDAHULUAN

Memuat latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

Menguraikan tentang dasar teori dan referensi yang mendasari penerapan algoritma genetika dalam membentuk model ARIMA dalam melakukan prediksi terhadap curah hujan.

BAB 3 METODOLOGI PENELITIAN

Membahas metode yang digunakan dalam penelitian yang terdiri dari tahapan penelitian, teknik pengumpulan data, algoritma yang digunakan, kebutuhan sistem, dan pengujian algoritma.

BAB 4 ANALISIS DAN PERANCANGAN SISTEM

Membahas analisis kebutuhan dan perancangan sistem *prototype* guna melakukan penerapan algoritma genetika dalam melakukan prediksi terhadap curah hujan.

BAB 5 IMPLEMENTASI

Membahas implementasi perangkat lunak sesuai dengan perancangan sistem yang telah dibuat.

BAB 6 PEMBAHASAN

Membuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan serta melakukan analisis terhadap hasil yang didapat.

BAB 7 PENUTUP

Membuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran untuk pengembangan sistem yang lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dibahas mengenai tinjauan pustaka diantaranya kajian pustaka dan dasar teori yang digunakan dalam penelitian. Kajian pustaka merupakan penelitian yang telah dilakukan sedang dasar teori merupakan pembahasan teori yang diperlukan dalam menyusun penelitian.

2.1 Kajian Pustaka

Berikut adalah data mengenai penelitian sejenis yang memiliki kemiripan dengan penelitian yang sedang dilakukan:

Tabel 2.1 Perbandingan dengan Penelitian Sebelumnya

No.	Judul	Penulis	Perbandingan	
			Kajian Pustaka	Skripsi Penulis
1.	Prediksi Harga Saham Berdasarkan Data Historis Menggunakan Model Regresi yang dibangun dengan Algoritma Genetika	Asyrofa Rahmi, Wayan Firdaus Mahmudy, Budi Darma Setiawan	Penggunaan model regresi linier yang dioptimasi dengan algoritma genetika	Penggunaan model regresi ARIMA yang dibentuk dengan algoritma genetika
2.	<i>Modeling and Forecasting Maternal Mortality; an Application of ARIMA Models</i>	Smart A. Sarpong	Melakukan prediksi dengan menggunakan model ARIMA	Melakukan prediksi dengan menggunakan model ARIMA yang dibentuk dengan algoritma genetika
3.	<i>Optimization of SARIMA Model using Genetic Algorithm Method in Forecasting Singapore Tourist Arrivals to Malaysia</i>	Mohd Zulariffin Md Maarof, Zuhaimy Ismail, Mohammad Fadzli	Algoritma genetika membentuk parameter dari model ARIMA	Algoritma membentuk model dari ARIMA

Tabel 2.1 (Lanjutan)

4.	<i>Model identification of ARIMA family using genetic algorithms</i>	Chorng-Shyong Ong, Jih-Jeng Huang, Gwo-Hshiung Tzeng	Bagaimana membentuk model ARIMA dengan algoritma genetika	Penerapan model ARIMA yang dibentuk dengan algoritma genetika untuk prediksi curah hujan
----	--	--	---	--

Sebagai kajian pustaka dalam penelitian ini, seperti terlihat pada Tabel 2.1 terdapat 4 penelitian. Penelitian pertama menjelaskan bagaimana algoritma genetika mampu membentuk model regresi linier berganda yang dapat digunakan untuk memprediksi harga saham. Hasil penelitian tersebut menyatakan representasi kromosom *real coded* mampu mengatasi permasalahan dalam menentukan nilai parameter dari regresi linier berganda. Dengan kombinasi parameter terbaik adalah *popSize* 1200, generasi 1500, *crossover rate* (*cr*) 0.5 dan *mutation rate* (*mr*) 0.5 dapat dihasilkan model terbaik dengan nilai *error* sebesar 47.5023 (Rahmi, et al., 2015).

Penelitian kedua menjelaskan bagaimana model ARIMA mampu digunakan dalam memprediksi rasio kematian ibu melahirkan dengan baik. Disebutkan dengan data *time series* yang telah stasioner, model ARIMA (1, 0, 2) merupakan model terbaik yang dapat digunakan dalam kondisi nyata terhadap studi kasus Okomfo Anokye Teaching Hospital Kumasi, Ghana (Sarpong, 2013).

Penelitian ketiga memberikan gambaran bagaimana algoritma genetika mampu meningkatkan kinerja dari model SARIMA yang masih memiliki kemiripan dengan ARIMA, dimana perbedaan terdapat pada SARIMA yang menekankan data musiman yakni data dengan suatu pola yang berulang-ulang. Penelitian tersebut berfokus bagaimana menentukan nilai parameter terbaik untuk model SARIMA dengan algoritma genetika, dengan hasil yang menyatakan bahwa penggunaan SARIMA dengan algoritma genetika lebih baik dibandingkan SARIMA biasa (Maarof, et al., 2014).

Penelitian keempat menjelaskan bagaimana membentuk model ARIMA dengan algoritma genetika, disini yang dibentuk adalah orde model dari ARIMA dengan penggunaan algoritma genetika. Dijelaskan ARIMA memiliki beberapa tahap diantaranya penentuan model, penentuan parameter, dan mendiagnosa kelayakan model. Tahapan paling krusial dari ARIMA adalah pada saat penentuan model, dan algoritma genetika disebut mampu mengatasi permasalahan tersebut dengan sangat baik (Ong, et al., 2005).

2.2 Curah Hujan

Curah hujan dapat dijelaskan dengan air hujan yang terkumpul pada suatu wadah datar dimana air hujan tersebut tidak meresap, tidak menguap, dan tidak mengalir sehingga dapat diukur ketinggiannya. Sebagai contoh curah hujan 1 (satu) milimeter dapat dijelaskan dalam suatu wadah datar dengan luas satu meter persegi tertampung air dengan ketinggian berkisar satu milimeter atau tertampung air sebanyak satu liter. metode dalam mendapatkan data curah hujan adalah dengan penggunaan penakar hujan yang kemudian diukur dengan satuan milimeter (mm) (User, 2015). Intensitas curah hujan sendiri dapat dikategorikan jika dibawah 20mm dalam 24 jam maka termasuk ringan, diatas 50mm dikategorikan lebat, sedang selang 20mm hingga 50mm adalah kategori normal (User, 2015).

Curah hujan merupakan salah satu faktor penting yang mempengaruhi perubahan iklim suatu wilayah (Mirawati, et al., 2013) salah satu kasus nyata adalah terjadinya perubahan iklim yang sangat tajam, atau biasa disebut dengan cuaca ekstrim. Kemudian beberapa sektor yang juga dipengaruhi oleh curah hujan diantaranya sektor komunikasi juga ikut terpengaruh terhadap curah hujan dimana penggunaan frekuensi diatas 10 GHz dapat terkena redaman hujan (Mauludiyanto, et al., 2009). Namun curah hujan paling berpengaruh terhadap sektor pertanian (Estiningtyas, et al., 2007). Hal tersebut diperkuat oleh penelitian Iriany, et al., (2015) yang menyatakan produktifitas pertanian kentang di wilayah Tengger, Bromo tidak stabil dikarenakan perubahan iklim yang tidak bersahabat.

2.3 Data *Time Series*

Data *time series* merupakan sebuah data statistik yang telah dihimpun dari masa lampau hingga sekarang dan dapat digunakan sebagai acuan dalam melakukan prediksi terhadap data yang akan datang (Wiyanti & Pulungan, 2012). Pendapat lain menyatakan bahwa data *time series* merupakan sebuah kegiatan mengamati data statistik dimana terdapat korelasi (depedensi) antar datanya, sehingga data yang sekarang sedang diamati memiliki ketergantungan dengan satu atau lebih data pengamatan sebelumnya (Mirawati, et al., 2013).

Prediksi merupakan sebuah perilaku mengamati data lampau hingga sekarang sebagai acuan untuk melakukan prakiraan sesuatu yang mungkin akan terjadi dimasa mendatang. Penggunaan data *time series* merupakan salah satu proses prediksi kuantitatif dimana berdasar data kuantitatif masa lampau yang direpresentasikan dalam bentuk angka atau nilai.

Wiyanti & Pulungan, (2012) lebih lanjut menjelaskan bahwa terdapat dua teknik prediksi yang menggunakan basis data *time series* yaitu teknik model matematika statistik dan berdasarkan kecerdasan buatan. Analisa *time series* sendiri memiliki beberapa peruntukan salah satunya digunakan dalam melakukan prediksi untuk masa yang akan datang (Mirawati, et al., 2013).

Namun perlu diingat, penggunaan data *time series* yang digunakan untuk melakukan prediksi akan memiliki hasil yang *powerfull* apabila datanya dalam keadaan stasioner (Brockwell & Davis, 2002). Data stasioner sendiri merupakan keadaan dimana data tidak mengalami pertumbuhan atau penurunan yang signifikan, sehingga nilai rata-rata yang dihasilkan hampir selalu konstan.

2.3.1 Klasifikasi Model ARIMA

Model ARIMA terdiri dari 3 unsur, yaitu: model *autoregressive* (AR) p , model *moving average* (MA) q dan *integrated* (I) d . Ketiga unsur tersebut nantinya akan membentuk model baru yaitu ARIMA. Dari model yang terbentuk tersebut maka model ARIMA dibagi menjadi 3 kelompok yaitu AR, MA, dan campuran (ARMA dan ARIMA). Model ARIMA merupakan metode prediksi yang cukup ampuh mengatasi data *time series*. Apabila data stasioner, dapat digunakan model AR, MA, atau model ARMA karena telah terbukti ampuh dalam mengatasi data stasioner (Brockwell & Davis, 2002). Sedangkan apabila ditemukan data yang tidak stasioner maka perlu dilakukan *differencing* yang berarti memenuhi *integrated* dan terbentuklah model ARIMA yang juga dikenal ampuh mengatasi data yang tidak stasioner (Sarpong, 2013).

2.3.1.1 Autoregressive Model (AR)

Model AR merupakan model peramalan dimana sebuah nilai x akan dipengaruhi oleh nilai x sebelumnya hingga periode ke- p . Sehingga variabel tersebut berpengaruh terhadap dirinya sendiri. Bentuk umum model AR dengan ordo p (AR (p)) atau model ARIMA ($p, 0, 0$) dinyatakan Brockwell & Davis (2002):

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + e_t \quad (2.1)$$

Dimana: c = konstanta

ϕ_p = parameter *autoregressive* ke- p

e_t = nilai kesalahan pada saat t

2.3.1.2 Moving Average Model (MA)

Model MA merupakan model peramalan dimana sebuah nilai x akan dipengaruhi oleh nilai eror dari variabel x tersebut. Bentuk umum model MA dengan ordo q (MA (q)) atau model ARIMA ($0, 0, q$) dinyatakan Brockwell & Davis (2002) :

$$X_t = \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} \quad (2.2)$$

Dimana: θ_1 sampai θ_q adalah parameter *moving average*

e_{t-q} = nilai kesalahan pada saat $t - q$

2.3.1.3 Model Campuran

Model campuran sendiri terbagi menjadi dua, yaitu model ARMA dimana terdapat proses AR (1) dan MA (1) dengan model ARIMA (1, 0, 1) sehingga didapat persamaan seperti berikut (Brockwell & Davis, 2002):

$$X_t = c + \phi_1 X_{t-1} + e_t - \theta_1 e_{t-1} \quad (2.3)$$

Dimana:

$c + \phi_1 X_{t-1} + e_t$ = persamaan model AR

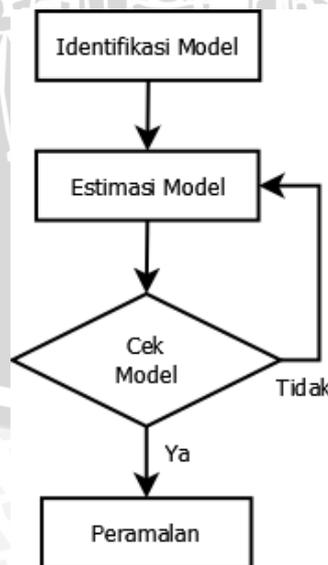
$\theta_1 e_{t-1}$ = persamaan model MA

Namun model ARMA hanya mampu menyelesaikan permasalahan data stasioner, sedangkan data *time series* mayoritas bukanlah data stasioner melainkan *integrated*. Sehingga data yang tidak stasioner terlebih dahulu harus distasionerkan dengan cara *differencing* (Sarpong, 2013) dimana differencing merupakan ordo I yang berarti memenuhi model ARIMA (1, 1, 1).

2.3.2 Box and Jenkins Methodology

Box dan Jenkins (1976) telah menjelaskan terdapat empat tahap dalam melakukan prediksi dengan menggunakan model ARIMA, yaitu: identifikasi model, estimasi parameter, memilih model terbaik, *forecasting* (Sarpong, 2013). Tahapan Box Jenkins secara umum dapat dilihat pada Gambar 2.1.

Perlu digaris bawahi bahwa dalam pemodelan ARIMA hal tersulit adalah menentukan model yang tepat, padahal penentuan model adalah hal yang paling penting dalam tahap prediksi menggunakan model ARIMA (Ong, et al., 2005). Apabila dalam pemilihan model tidak tepat maka hasil akurasi peramalan tidak akan baik, begitu sebaliknya apabila didapat model yang baik hasil peramalan akan mencapai akurasi yang baik.



Gambar 2.1 Flowchart ARIMA

2.3.2.1 Cek Stasioner Data

Kelebihan model ARIMA adalah kemampuan untuk dapat mengatasi data yang tidak stasioner, dimana proses menangani data tidak stasioner tersebut adalah dengan melakukan *differencing*. Data stasioner merupakan kondisi tidak adanya pertumbuhan atau penurunan data, sehingga dapat dinyatakan fluktuasi data berkisar pada nilai rata-rata yang konstan, tidak tergantung pada waktu dan varians dari fluktuasi.

Tahapan cek stasioner data diawali dengan melakukan pengujian terhadap data yang digunakan apakah stasioner atau tidak, terdapat banyak metode pengujian stasioner data diantaranya adalah melihat apakah dalam periode tertentu rata-rata tidak mengalami perbedaan yang signifikan, atau dapat dengan cara menggunakan metode *Augmented Dickey-Fuller (ADF) test* (Sarpong, 2013).

$$Y_t = \rho Y_{t-1} + u_t \quad (2.4)$$

Sehingga dengan pengujian ADF akan diketahui apakah data stasioner atau tidak dengan mengikuti kaidah hipotesis seperti berikut:

$H_1: \rho = 0$ (Terdapat *unit roots*, variabel Y tidak stasioner)

$H_2: \rho = 1$ (Tidak terdapat *unit roots*, variabel Y stasioner)

Proses *differencing* sendiri merupakan transformasi nilai data dengan melakukan Persamaan 2.5 berikut:

$$Y'_t = Y_t - Y_{t-1} \quad (2.5)$$

Dimana:

Y'_t = data *differencing* pada periode ke-t.

Y_t = data pada periode ke-t

Jika data mengalami *differencing* maka untuk mendapatkan data prediksi yang sesungguhnya data hasil prediksi perlu dilakukan *undifferencing* yakni mengembalikan data agar sesuai dengan data awal yang digunakan, sehingga didapatkan data prediksi yang sesungguhnya. Proses *undifferencing* menggunakan Persamaan 2.6

$$Y''_t = Y'_t + Y_{t-1} \quad (2.6)$$

Dimana:

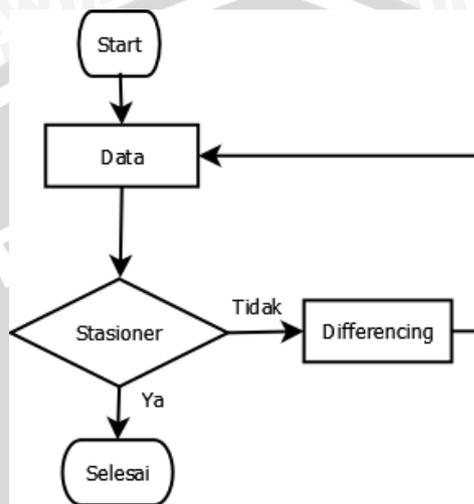
Y''_t = data *undifferencing* pada periode ke-t.

Y'_t = data prediksi pada periode ke-t.

Y_{t-1} = data asli pada periode t-1.

2.3.2.2 Identifikasi Model

Tahapan melakukan identifikasi model diawali dengan memastikan data yang digunakan apakah termasuk stasioner atau tidak seperti terlihat pada Gambar 2.2. Jika data yang dipakai telah stasioner maka tinggal menentukan model peramalan menggunakan AR, MA, atau ARMA namun jika didapatkan data yang tidak stasioner maka perlu dilakukan *differencing* sehingga didapatkan model ARIMA.



Gambar 2.2 Pengecekan data model ARIMA

Menentukan model ARIMA dapat dilakukan dengan melihat *autocorrelation Function* (ACF) dan *partial autocorrelation Function* (PACF) (Mauludiyanto, et al., 2009), namun sebuah penelitian menyebutkan penentuan model ARIMA dengan menggunakan algoritma genetika dapat menghasilkan model yang lebih akurat dan hasil sesuai dengan harapan (Ong, et al., 2005).

2.3.2.3 Estimasi Parameter

Selain memiliki kesulitan dalam menentukan model ARIMA yang baik kesulitan berikutnya adalah menentukan nilai parameter yang digunakan dalam pemodelan ARIMA. Cara yang sering digunakan adalah dengan mencoba berbagai nilai kemudian di cek (*trial and error*) atau dengan melakukan perbaikan iteratif dengan membiarkan program komputer yang melakukan perbaikan.

Penentuan parameter dari model ARIMA disebutkan dapat menggunakan pendekatan *ordinary least-square* (OLS). OLS mampu untuk menentukan nilai parameter dari model regresi linier dan termasuk dalam menentukan nilai parameter dari model ARIMA.

2.3.2.4 Diagnosa Model

Salah satu langkah yang cukup penting adalah setelah didapatkan model ARIMA maka perlu dilakukan pengecekan apakah model yang didapatkan dalam mengekstrak semua informasi dari data yang digunakan (Sarpong, 2013). Sehingga dengan diagnosa model diharapkan model yang digunakan dalam prediksi akan menghasilkan nilai yang lebih baik lagi.

Beberapa metode dalam melakukan diagnosa model ARIMA adalah dengan melakukan pengecekan *mean absolute percentage error* (MAPE) yang dapat dilihat pada Persamaan 2.7

$$MAPE = \frac{1}{N} \sum_{t=1}^N \left| \frac{z_t - \hat{z}_t}{z_t} \right| \quad (2.7)$$

Dimana: z_t = nilai data aktual

\hat{z}_t = nilai data peramalan

N = banyaknya periode peramalan

Model terbaik adalah yang memiliki nilai MAPE terkecil, setelah didapatkan model terbaik berdasarkan diagnosa model maka dapat dilakukan tahap selanjutnya yaitu *forecasting*.

2.4 Algoritma Genetika

Pada awal tahun 1970 John Holland salah satu mahasiswa Universitas Michigan New York Amerika Serikat berhasil menemukan algoritma genetika untuk pertama kalinya, kemudian pada tahun 1975 dengan murid serta rekan kerjanya menghasilkan sebuah buku yang diberi judul "*Adaption in Natural and Artificial Systems*" (Sivanandam & Deepa, 2008).

Algoritma genetika bekerja sesuai ilmu alam dimana pada proses reproduksi akan terjadi proses seleksi gen secara natural, dan yang mampu melewati seleksi tersebut yang dapat bertahan hidup. Dengan mengambil gen terbaik maka diharapkan akan menghasilkan keturunan yang baik pula.

```
1 procedure Algen
2 begin
3   i = 0
4   inisialisasi P(i)
5   while (bukan kondisi berhenti) do
6     reproduksi O(i) dari P(i)
7     evaluasi P(i) dan O(i)
8     seleksi P(i+1) dari P(i) dan O(i)
9     i = i + 1
10  end while
11 end
```

Sumber: Diadaptasi dari Mahmudy, 2013

Algoritma genetika dianggap mampu mengatasi masalah kompleks seperti misalnya perencanaan dan penjadwalan produksi, kompresi citra, penjadwalan dan alokasi ruang, dan masih banyak lagi (Mahmudy, 2013). Tahapan dalam algoritma genetika diantaranya adalah:

1. Tahap inisialisasi
2. Tahap reproduksi
3. Tahap seleksi

2.4.1 Tahap Inisialisasi

Tahap inisialisasi adalah tahapan memetakan permasalahan kedalam bentuk kromosom. Kromosom sendiri terdiri dari gen yang direpresentasikan sebagai variabel keputusan yang akan digunakan dalam solusi (Mahmudy, 2013). Kromosom ini nantinya akan ditampung didalam populasi, populasi / *popSize* sendiri menyatakan banyaknya individu / kromosom dengan sebuah ukuran yang telah ditetapkan sebelumnya. Lebih lanjut dalam melakukan pemetaan solusi kedalam kromosom terdapat tiga skema umum yaitu *real number encoding*, *discrete decimal encoding*, dan *binary encoding*.

Dalam melakukan penentuan model ARIMA yang baik, dapat digunakan pengkodean biner dimana terbagi menjadi dua bagian yang akan merepresentasikan AR dan MA dengan tiap bagian akan terdiri dari beberapa bit. Kromosom akan menyatakan lag pada orde model, nilai 1 menyatakan lag signifikan sedang 0 tidak signifikan (Ong, et al., 2005). Sebagai contoh didapat bentuk kromosom [10011;01010] maka model ARIMA yang memungkinkan adalah ARIMA ([1,4,5], d, [2,4]).

2.4.2 Tahap Reproduksi

Tahap reproduksi berfungsi dalam menghasilkan keturunan (*offspring*) dari individu yang ada pada populasi. Pada algoritma genetika sendiri terdapat dua proses dalam reproduksi sendiri yaitu tukar silang (*crossover*) dan mutasi (*mutation*).

2.4.2.1 Crossover

Tahapan melakukan *crossover* adalah dengan memilih dua induk (*parent*) secara acak pada populasi. Terlebih dahulu harus ditentukan parameter *crossover rate* (*cr*). Nilai *cr* nantinya akan menjadi rasio dari *offspring* yang dihasilkan ketika proses *crossover* terhadap ukuran populasi, maka akan dapat diketahui jumlah *offspring* yang dihasilkan sebesar $cr \times popSize$.

Teknik *crossover* yang digunakan pada tahap pemilihan model ARIMA dimana dilakukan pengkodean biner maka dapat dilakukan metode *two-point-crossover* (Ong, et al., 2005). Cara kerja metode ini adalah memilih sejumlah *parent* secara acak sesuai dengan kemungkinan *offspring* yang akan dihasilkan berdasarkan $cr \times popSize$.

Setelah ditemukan *parent* kemudian menentukan dua titik potong dengan rentang $[1 - (N-1)]$ dimana diantara titik potong dilakukan tukar silang terhadap gen, sehingga akan dihasilkan *offspring* baru hasil *crossover*. Misalkan didapatkan *parent* 1 dan *parent* 2 dengan titik potong $[3,7]$ maka metode *crossover* adalah seperti berikut

$$P_1 [010|0100|0111]$$

$$P_2 [101|0110|0100]$$

Maka akan menghasilkan *offspring*

$$C_1 [01001100111]$$

$$C_2 [10101000100]$$

2.4.2.2 Mutation

Tahap mutasi berfungsi menjaga keberagaman populasi dengan cara menciptakan individu baru dengan jalan melakukan modifikasi satu atau lebih gen pada individu yang sama. Metode mutasi yang dapat digunakan adalah dengan sebuah proses acak dimana memilih salah satu gen didalam kromosom yang ditentukan untuk kemudian dirubah nilainya dari 0 menjadi 1 dan sebaliknya (Ong, et al., 2005). Misal terpilih *parent* 1 dengan titik random adalah 4 maka mutasinya akan menjadi seperti berikut

$$P_1 [01001000111]$$

$$C_1 [01011000111]$$

2.4.3 Nilai Fitness

Fitness merupakan sebuah prosedur untuk mengukur tingkat kebaikan suatu solusi yang diberikan oleh individu. Sebuah individu dengan nilai *fitness* yang tinggi memiliki kemungkinan besar untuk menghasilkan solusi terbaik. Perhitungan nilai *fitness* untuk penelitian pembentukan model regresi ARIMA dengan algoritma genetika dapat dilihat pada Persamaan 2.8 berikut:

$$f = \frac{1}{(\varepsilon + \alpha)} \quad (2.8)$$

Dimana nilai α merupakan nilai yang sangat kecil sehingga menghindarkan nilai f tak terhingga apabila didapatkan nilai *error* = 0.

2.4.4 Tahap Seleksi

Tahapan seleksi berfungsi untuk memastikan individu terbaik yang dipertahankan untuk tetap hidup sehingga diharapkan generasi yang akan dihasilkan juga akan mewarisi kebaikan dari *parentnya*. Metode seleksi yang digunakan adalah *elitism selection* dimana dengan mengumpulkan semua individu baik *parent* maupun *offspring* dalam satu penampungan (Mahmudy, 2013). Metode ini menjamin bahwa individu terbaik saja yang dapat lolos kedalam generasi selanjutnya.

1	PROCEDURE SeleksiElitisme
2	Input:
3	<i>Pop</i> : berisi individu yang terhimpun dalam populasi
4	<i>popSize</i> : ukuran populasi yang digunakan
5	<i>Os</i> : berupa himpunan <i>offspring</i> yang dihasilkan dari proses seleksi
6	(<i>crossover</i> dan mutasi)
7	
8	Output:
9	<i>POP</i> : hasil individu baru yang terbentuk setelah dilakukan proses
10	Seleksi dan terhimpun kedalam populasi baru
11	
12	<i>/* menempatkan individu Pop dan Os kedalam variabel sementara */</i>
13	<i>Temp</i> ← Merge (<i>Pop</i> , <i>Os</i>)
14	<i>/* merubah urutan individu berdasarkan fitness dengan cara ascending</i>
15	<i>*/</i>
16	OrderAscending (<i>Temp</i>)
17	<i>/* simpan individu terbaik kedalam POP */</i>
18	<i>POP</i> ← CopyBest (<i>Temp</i> , <i>popSize</i>)
	END PROCEDURE

Sumber: Diadaptasi dari Mahmudy, 2013

2.5 Parameter Algoritma Genetika

Penentuan parameter dalam algoritma genetika terbilang cukup sulit, dikarenakan jika nilai parameter untuk populasi (*popSize*), *crossover* (*cr*), dan *mutation* (*mr*) dibuat kecil maka kemungkinan untuk menemukan solusi terbaik tidak akan maksimal. Tetapi penggunaan nilai parameter yang semakin besar memang dapat meningkatkan kemampuan eksplorasi dalam menemukan solusi terbaik namun hal ini harus dibayar dengan waktu komputasi yang cukup lama. Selain waktu komputasi yang cukup lama, nilai parameter yang besar juga memungkinkan tidak didapatkannya solusi terbaik dikarenakan eksplorasi yang terlalu luas. Penentuan nilai parameter dalam algoritma genetika tidak ada metode yang pasti, padahal kombinasi nilai yang tepat akan mempengaruhi permasalahan yang dapat diselesaikan.

Oleh karena itu sebuah penelitian dilakukan guna mencari kombinasi nilai parameter yang dianggap memadai dalam pengujian awal (Mahmudy, 2013) adalah seperti berikut:

1. Nilai populasi (*popSize*) = 30 – 50
2. Nilai *crossover* (*cr*) = 0.3 – 0.8
3. Nilai *mutation* (*mr*) = 0.1 – 0.3

BAB 3 METODOLOGI PENELITIAN

Skripsi dengan judul Pembentukan Model Regresi ARIMA untuk Melakukan Prediksi Curah Hujan Menggunakan Algoritma Genetika termasuk dalam penelitian implementatif dengan pendekatan perancangan (*design*). Penelitian ini akan menghasilkan sebuah *prototype* berupa perangkat lunak yang dapat digunakan untuk memprediksi curah hujan.

3.1 Tahapan Penelitian

Sebagai penelitian implementatif dengan pendekatan perancangan (*design*) maka memiliki proses yang dimulai dari:

1. Analisis mengenai formulasi permasalahan yang diteliti yaitu bagaimana membangun model ARIMA dengan algoritma genetika dalam melakukan prediksi curah hujan.
2. Merancang tahap penyelesaian terhadap analisis masalah menggunakan algoritma genetika dengan merancang operator genetika yang digunakan (*crossover*, mutasi, seleksi).
3. Pengujian terhadap parameter algoritma *heuristic* sehingga didapatkan solusi terbaik.

3.2 Teknik Pengumpulan Data

Data yang digunakan dalam penelitian merupakan data sekunder, dimana data yang telah diperoleh melalui penelitian yang telah dilakukan mengenai peramalan cuaca menggunakan GSTAR-SUR di Tengger yang terdiri dari beberapa wilayah diantaranya Puspo, Sumber, Tosari, dan Tutur (Iriany, et al., 2015). Data yang diperoleh berupa data dasarian, data dasarian merupakan data yang diperoleh dengan rentang waktu sepuluh (10) hari selama pengamatan berlangsung (User, 2015). Periode data yang didapatkan dari penelitian Iriany, et al., (2015) mulai dari tahun 2005 hingga 2013. Selain data curah hujan, juga dibutuhkan data input untuk parameter algoritma genetika yang hendak digunakan, meliputi jumlah populasi (*popSize*), jumlah generasi, *crossover rate* (*cr*), dan *mutation rate* (*mr*).

3.3 Algoritma yang Digunakan

Penelitian ini adalah bagaimana membangun model ARIMA dengan algoritma genetika dimana telah diuraikan pada BAB 2 bahwa model ARIMA yang dibangun dengan algoritma genetika terbukti mampu dan menghasilkan solusi terbaik untuk digunakan dalam melakukan prediksi curah hujan. Hal ini juga telah dibuktikan mengenai penelitian yang menyebutkan membangun model ARIMA dengan algoritma genetika terbukti mampu menghasilkan peramalan yang lebih baik (Maarof, et al., 2014).

Dalam membangun *prototype* aplikasi peramalan curah hujan menggunakan model ARIMA dengan algoritma genetika digunakan bahasa pemrograman JAVA. Bahasa pemrograman JAVA merupakan bahasa pemrograman berbasis *object oriented* yang telah banyak digunakan oleh *programmer* diseluruh dunia (Deitel & Deitel, 2015). Selain itu java juga dikenal memiliki performa yang sangat baik apabila digunakan dalam komputasi tingkat tinggi. Sebuah penelitian mengenai JAVA dalam komputasi tingkat tinggi telah dilakukan baik dengan *shared memory* ataupun tidak, memberikan hasil bahwa JAVA dapat mendekati performa bahasa pemrograman yang dikompilasi secara native baik secara *sequential* maupun *parallel* (Taboada, et al., 2013).

3.4 Kebutuhan Sistem

Analisa kebutuhan sistem digunakan dalam menentukan kebutuhan dalam menjalankan sistem sehingga dapat berjalan dengan optimal serta dapat memenuhi kebutuhan pengguna.

3.4.1 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam tahap pengembangan perangkat lunak *prototype* untuk membentuk model regresi ARIMA dengan algoritma genetika sebagai prediksi curah hujan sebagai berikut:

1. *Processor* Intel® Core™ i7-3612QM CPU @ 2.10GHz
2. Memori 4096MB
3. *Harddisk* dengan kapasitas 500GB
4. VGA AMD Radeon HD 7550M 2GB

3.4.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan sebagai bagian dalam penelitian ini sebagai berikut:

1. Sistem operasi Windows 7 Ultimate 64-bit
2. *Java Development Kit* (JDK) 1.8.0
3. Eclipse IDE 4.5.0
4. Maven berfungsi dalam *dependency management*.
5. Dia sebagai pembuatan diagram alir

3.5 Pengujian Algoritma

Pengujian algoritma bertujuan untuk menentukan uji coba serta evaluasi terhadap penentuan parameter algoritma. Hal ini dikarenakan tidak adanya metode yang pasti dalam menentukan nilai parameter algoritma genetika untuk menghasilkan nilai *fitness* yang selalu baik. Pengujian yang akan dilakukan antara lain sebagai berikut:

1. Uji coba penentuan ukuran populasi yang optimal
2. Uji coba penentuan banyaknya generasi yang optimal
3. Uji coba mencari kombinasi *cr* dan *mr* yang optimal
4. Uji coba banyaknya periode data curah hujan yang akan diprediksi.

3.5.1 Uji Coba Ukuran Populasi

Pengujian pada tahap ini dilakukan guna menentukan ukuran populasi yang dapat menghasilkan model yang terbaik, sehingga dapat dilakukan prediksi curah hujan mendekati dengan data aslinya. Ukuran populasi yang digunakan mulai dari 15. Percobaan diulang sepuluh kali kemudian dihitung rata-rata nilai fitnessnya. Percobaan dilanjutkan dengan menaikkan nilai ukuran populasi sebesar 15 dan terus berlanjut sampai rata-rata fitness tidak mengalami kenaikan signifikan. Rancangan pengujian ukuran populasi dapat dilihat pada Tabel 3.1, dengan penggunaan parameter algoritma genetika yang lain sebagai berikut:

- a. Ukuran populasi = 15 – 120
- b. Ukuran generasi = 250
- c. *Cr* = 0.8
- d. *Mr* = 0.2
- e. Periode data = 6

Tabel 3.1 Perancangan uji coba ukuran populasi

Ukuran Populasi	Nilai <i>Fitness</i> dari Percobaan ke-										Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
15											
30											
45											
60											
75											
90											

Tabel 3.1 (Lanjutan)

105										
120										

3.5.2 Uji Coba Banyaknya Generasi

Pengujian banyaknya generasi adalah untuk dapat memastikan jumlah terbaik untuk generasi dalam menentukan model ARIMA terbaik. Dengan harapan model yang dihasilkan dapat melakukan prediksi curah hujan mendekati data aslinya. Banyak generasi yang digunakan adalah kelipatan 250 dengan rancangan pengujian seperti pada Tabel 3.2. Parameter yang dipergunakan adalah sebagai berikut:

- Ukuran populasi = Hasil populasi terbaik dari pengujian populasi.
- Ukuran generasi = 250 – 1500
- Cr = 0.8
- Mr = 0.2
- Periode data = 6

Tabel 3.2 Perancangan uji coba banyaknya generasi

Jumlah Generasi	Nilai <i>Fitness</i> dari Percobaan ke-										Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
250											
500											
750											
1000											
1250											
1500											

3.5.3 Uji Coba Kombinasi cr dan mr

Pengujian kombinasi cr dan mr adalah bagaimana menentukan komposisi cr dan mr terbaik sehingga harapannya dapat menemukan model ARIMA terbaik yang dapat digunakan dalam prediksi curah hujan. Penggunaan nilai cr dan mr adalah selang antara 0 hingga 1 (Mahmudy, 2013). Rencana pengujian kombinasi cr dan mr dapat dilihat pada Tabel 3.3 dengan parameter yang digunakan sebagai berikut:



- a. Jumlah populasi = hasil populasi terbaik dari pengujian populasi.
- b. Jumlah generasi = hasil generasi terbaik dari pengujian generasi.
- c. Periode data = 6

Tabel 3.3 Perancangan uji coba kombinasi cr dan mr

Kombinasi		Nilai <i>Fitness</i> dari Percobaan ke-										Rata-rata <i>fitness</i>
		1	2	3	4	5	6	7	8	9	10	
cr	Mr											
1	0											
0.9	0.1											
0.8	0.2											
0.7	0.3											
0.6	0.4											
0.5	0.5											
0.4	0.6											
0.3	0.7											
0.2	0.8											
0.1	0.9											
0	1											

3.5.4 Uji Coba Banyaknya Periode Curah Hujan

Pengujian banyaknya periode curah hujan bertujuan untuk menentukan banyaknya periode data curah hujan yang dapat menghasilkan prediksi terbaik dengan model ARIMA. Data yang digunakan merupakan data dasarian dimana dalam satu bulan terdapat 3 dasarian, sehingga penggunaan periode data adalah kelipatan 3. Rancangan pengujian banyaknya periode data curah hujan seperti pada Tabel 3.4 dengan parameter algoritma genetika seperti berikut:

- a. Ukuran populasi = populasi terbaik berdasarkan pengujian populasi.
- b. Ukuran generasi = generasi terbaik berdasarkan pengujian generasi.
- c. Cr dan mr = cr dan mr terbaik berdasarkan pengujian kombinasi cr dan mr.

Tabel 3.4 Perancangan uji coba banyak periode data curah hujan

Banyak Periode	Nilai <i>Fitness</i> dari Percobaan ke-										Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
3 Dasarian											
6 Dasarian											
9 Dasarian											
12 Dasarian											



BAB 4 ANALISIS DAN PERANCANGAN SISTEM

Pada Bab ini akan menjelaskan mengenai bagaimana melakukan pembentukan model ARIMA dengan algoritma genetika sehingga dapat dibangun kedalam sistem *prototype* untuk melakukan prediksi curah hujan yang akan dihasilkan.

4.1 Formulasi Permasalahan

Dalam melakukan prediksi curah hujan terdapat beberapa metode yang dapat digunakan, salah satunya menggunakan pendekatan statistika dengan mempelajari data yang telah ada kemudian diolah dengan model ARIMA yang dikenal mampu memprediksi dengan input data *time series* yang tidak stasioner (Wiyanti & Pulungan, 2012). Kemudian model ARIMA tersebut akan dibentuk menggunakan algoritma genetika sehingga akan didapatkan model ARIMA terbaik untuk melakukan prediksi terhadap curah hujan.

Tabel 4.1 Data curah hujan dasarian tahun 2014 wilayah Puspo, Tengger.

No.	Bulan	Dasarian	Data Curah Hujan
1.	Januari	1	23.800
2.		2	3.300
3.		3	10.273
4.	Februari	1	14.000
5.		2	6.000
6.		3	10.375
7.	Maret	1	6.500
8.		2	15.600
9.		3	3.273
10.	April	1	6.300
11.		2	9.300
12.		3	14.700
13.	Mei	1	2.600
14.		2	13.600
15.		3	12.909

Proses prediksi menggunakan model ARIMA sendiri terdiri dari beberapa tahapan diantaranya:

1. Melakukan pengecekan stasioner data dan melakukan *differencing* apabila data tidak stasioner.
2. Melakukan pendugaan model.
3. Memilih model terbaik dengan menghitung nilai MAPE terkecil.

Sehingga dengan menggunakan data pada Tabel 4.1 proses prediksi curah hujan untuk data dasarian pada periode selanjutnya akan diawali dengan mengecek stasioner data pada Tabel 4.1 dengan menggunakan *ADF test* dengan Persamaan 2.4 dimana didapatkan hipotesa: $H_1: \rho = 0$, yang menandakan data tidak stasioner. Untuk hal itu perlu dilakukan *differencing* terhadap data pada Tabel 4.1 dimana hasil *differencing* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil *differencing* data

No.	Hasil <i>Differencing</i>
1.	-20.500
2.	6.973
3.	3.727
4.	-8.000
5.	4.375
6.	-3.875
7.	9.100
8.	-12.327
9.	3.027
10.	3.000
11.	5.400
12.	-12.100
13.	11.000
14.	-0.691

Setelah melalui hasil *differencing*, data Tabel 4.2 dilakukan *ADF tes* untuk memastikan apakah data tersebut sudah stasioner dengan Persamaan 2.4, hasil yang didapatkan adalah $H_2: \rho = 1$ yang berarti data sudah stasioner. Proses *differencing* juga memberi indikasi nilai $d = 1$ pada variabel p, d, q untuk model ARIMA ($p, 1, q$).



Setelah diketahui bahwa model yang akan digunakan adalah model ARIMA maka langkah selanjutnya adalah menentukan parameter dari AR (p) dan MA (q) dalam model ARIMA. Data yang digunakan dalam parameter AR (p) adalah data curah hujan, sedangkan data yang digunakan dalam parameter MA (q) merupakan data *error term*. Penentuan parameter AR (p) dan MA (q) didasarkan data *time series* curah hujan yang ditentukan berdasarkan periodenya (X). Dengan periode tertentu (X) data satu dasarian membutuhkan data curah hujan sejumlah periode dengan lebih satu data yang digunakan sebagai parameter Y, yang berfungsi dalam mencari nilai koefisien. Sehingga dengan menggunakan data curah hujan 5 periode (5X), data satu dasarian memerlukan data curah hujan sebanyak 6 serta data *error term* sebanyak 5.

Tabel 4.3 Data *time series* curah hujan 5 periode

Data ke-	Y	X1	X2	X3	X4	X5	e ₁	e ₂	e ₃	e ₄	e ₅
1.	- 20. 5	6.9 72	3.7 27	-8	4.3 75	- 3.8 75	- 19. 72 2	7.7 50 72	4.5 05 27	- 7.2 22	5.15 3
2.	6.9 72	3.7 27	-8	4.3 75	- 3.8 75	9.1	7.7 50 72	4.5 05 27	- 7.2 22	5.1 53	- 3.09 7
3.	3.7 27	-8	4.3 75	- 3.8 75	9.1	- 12. 32 72	4.5 05 27	- 7.2 22	5.1 53	- 3.0 97	9.87 8
4.	-8	4.3 75	- 3.8 75	9.1	- 12. 32 72	3.0 27	- 7.2 22	5.1 53	- 3.0 97	9.8 78	- 11.5 4927 273
5.	4.3 75	- 3.8 75	9.1	- 12. 32 72	3.0 27	3	5.1 53	- 3.0 97	9.8 78	- 11. 54 92 72	3.80 5272 727
6.	- 3.8 75	9.1	- 12. 32 72	3.0 27	3	5.4	- 3.0 97	9.8 78	- 11. 54 92 72	3.8 05 27	3.77 8



Tabel 4.3 (lanjutan)

7.	9.1 - 12. 32 72	- 12. 32 72	3.0 27	3	5.4	- 12. 1	9.8 78	- 11. 54 92 72	3.8 05 27	3.7 78	6.17 8
8.	- 12. 32 72	3.0 27	3	5.4	- 12. 1	11	- 11. 54 92 72	3.8 05 27	3.7 78	6.1 78	- 11.3 22
9.	3.0 27	3	5.4	- 12. 1	11	- 0.6 90	3.8 05 27	3.7 78	6.1 78	- 11. 32 2	11.7 78

Setelah membentuk model ARIMA dengan 5 periode data curah hujan kebelakang, maka dapat dilakukan proses prediksi dengan menggunakan model ARIMA (5, 1, 5) dengan parameter C: -1.587, serta untuk parameter AR dan MA tertera pada Tabel 4.4.

Tabel 4.4 Paramater AR dan MA model ARIMA

No.	AR (p)	MA (q)
1.	0.00	1.00
2.	0.00	0.00
3.	0.00	0.00
4.	0.00	0.00
5.	0.00	0.00

Sehingga dengan menggunakan data Tabel 4.2 diuji tingkat keberhasilan peramalan model ARIMA (5, 1, 5) terhadap data pada Tabel 4.1 dengan Persamaan 2.3 yang berarti persamaan model ARIMA maka didapatkan persamaan berikut:

$$Y_t = -1.587 + (0.0 Y_{t-1}) + (0.0 Y_{t-2}) + (0.0 Y_{t-3}) + (0.0 Y_{t-4}) + (0.0 Y_{t-5}) + (1 * e_{t-1}) - (0.00 e_{t-2}) - (0.0 e_{t-3}) - (0.0 e_{t-4}) - (0.00 e_{t-5}) \tag{4.1}$$



Menggunakan Persamaan 4.1 nantinya didapatkan data prediksi, disini dikarenakan data prediksi menggunakan data hasil *differencing* maka perlu melakukan tahap pengembalian data semula agar didapatkan data prediksi yang sesungguhnya. Tahapan tersebut dikarenakan agar data prediksi dapat dibandingkan terhadap data asli yang tercantum pada Tabel 4.1. Proses membalikan *differencing* menggunakan Persamaan 2.6.

Tabel 4.5 Perbandingan prediksi dengan model arima terhadap data asli

No.	Bulan	Dasarian	Data Asli (Y)	Data Prediksi Differencing (Y')	Data Prediksi Undifferencing (Y'')
1.	Januari	1	23.800	-1.587	22.213
2.		2	3.300	-1.587	1.713
3.		3	10.273	-1.587	8.686
4.	Februari	1	14.000	-1.587	12.413
5.		2	6.000	-1.587	4.413
6.		3	10.375	-1.587	8.788
7.	Maret	1	6.500	-19.84027273	26.340
8.		2	15.600	-16.94154545	-1.342
9.		3	3.273	-1.559727273	1.713
MAPE					0.649758553

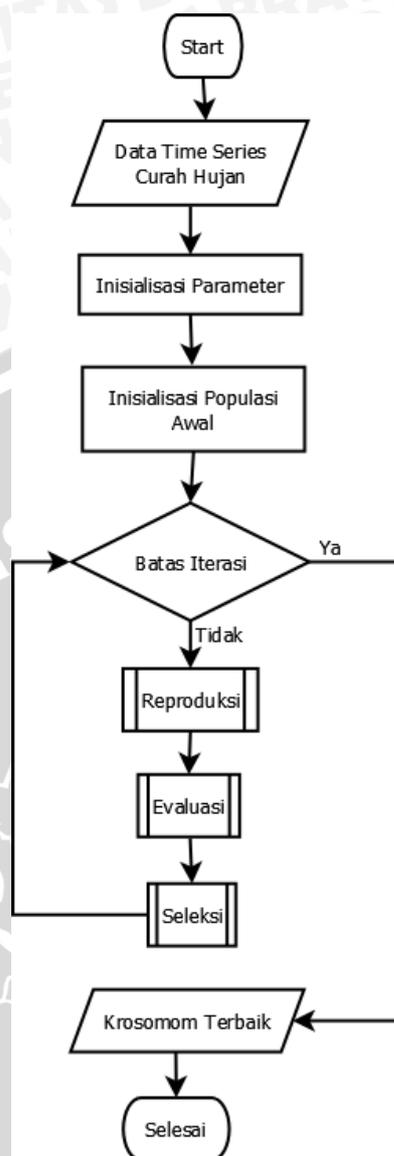
Tabel 4.5 merupakan penjelasan dari hasil prediksi dengan menggunakan model ARIMA, dapat dilihat bagaimana perbandingan antara data asli dengan data prediksi yang telah dilakukan transformasi untuk mengembalikan efek *differencing*. Setelah itu dihitung nilai *Mean Percentage Error* (MAPE) dari hasil pengujian Persamaan 4.1 sebagai model peramalan dengan menggunakan Persamaan 2.7 maka didapatkan nilai MAPE sebesar 0.649758553. Nilai MAPE yang dihasilkan termasuk masih besar, karena semakin besar nilai MAPE maka tingkat akurasi dari peramalan semakin menurun. Oleh karena itu diperlukan perbaikan dalam penentuan model ARIMA dengan menggunakan algoritma genetika guna memperbaiki pemilihan nilai model dengan nilai MAPE yang lebih kecil (Ong, et al., 2005).



4.2 Siklus Pembentukan Model ARIMA dengan Algoritma Genetika

Berikut adalah penjelasan mengenai bagaimana alur kerja prediksi curah hujan menggunakan model ARIMA yang dibentuk dengan algoritma genetika, sebagai tahapan dapat dilihat pada diagram alur seperti pada Gambar 4.1 dimana proses yang terjadi dapat dijelaskan seperti berikut:

1. Memberi input data *time series* curah hujan yang akan digunakan sebagai pembelajaran guna untuk melakukan prediksi curah hujan.
2. Melakukan pengecekan apakah data yang di masukan berupa data stasioner atau tidak, apabila data tidak stasioner maka akan dilakukan *differencing* hingga didapat data stasioner dimana jumlah melakukan *differencing* akan menjadi nilai dari d pada model ARIMA (0, d , 0).
3. Inialisasi parameter awal untuk algoritma genetika diantaranya yaitu jumlah populasi (*popSize*), jumlah generasi, *crossover rate* (cr), dan *mutation rate* (mr).
4. Membuat populasi awal sesuai jumlah populasi yang telah ditentukan.
5. Melakukan tahap reproduksi terhadap populasi awal untuk menghasilkan generasi baru:
 - a) Melakukan proses *two-cut-point crossover* terhadap individu dari populasi yang terpilih sejumlah cr yang telah ditentukan.
 - b) Melakukan proses *random mutation* terhadap individu dari populasi yang terpilih sejumlah mr yang telah ditentukan.
 - c) Menghitung nilai *fitness* dari keseluruhan populasi yang telah dihasilkan.
 - d) Melakukan proses seleksi terhadap individu dalam populasi menggunakan metode *elitism selection*.
6. Setelah proses 5 selesai maka akan diulangi kembali hingga batas generasi yang ditentukan atau ditemukannya solusi optimal dari permasalahan yang sedang diteliti.



Gambar 4.1 Tahapan alur sistem

Sumber: Diadaptasi dari (Rahmi, et al., 2015)

4.3 Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika

Permasalahan dalam memprediksi curah hujan menggunakan model ARIMA dengan algoritma genetika dapat dijabarkan dengan perhitungan manual yang seperti telah dijelaskan pada Gambar 4.1. Sebagai contoh dalam perhitungan ini akan menggunakan contoh data pada Tabel 4.1 kemudian untuk inisialisasi algoritma genetika akan menggunakan parameter seperti berikut:

1. Jumlah Generasi = 1
2. Ukuran Populasi (*popSize*) = 5
3. *Crossover Rate* (*cr*) = 0,4
4. *Mutation Rate* (*mr*) = 0,1

4.3.1 Representasi Kromosom dan Perhitungan Fitness

Dalam algoritma genetika pembentukan kromosom sebagai bentuk solusi permasalahan adalah cukup penting. Sebagai solusi dalam menentukan model ARIMA terbaik maka representasi kromosom yang dapat digunakan adalah dengan membangkitkan bilangan biner acak. Hasil nilai acak tersebut menyatakan orde periode data yang berpengaruh dalam prediksi dengan model ARIMA. Panjang kromosom adalah 2 (dua) kali jumlah periode data dimana dibagi menjadi dua bagian sebagai representasi AR dan MA, yang nantinya bertindak sebagai orde untuk AR dan MA (Ong, et al., 2005). Sehingga dalam menentukan model ARIMA (p, d, q) terlihat pada Tabel 4.6 index ke-1 hingga ke-10 menyatakan orde periode data yang akan digunakan apabila nilai biner bernilai 1 dan periode data dengan nilai biner 0 tidak digunakan. Berdasarkan Tabel 4.6 maka didapatkan model ARIMA ([1,2,4,5],0,[1,2]).

Tabel 4.6 Pembentukan Kromosom Parent 1

1	2	3	4	5	1	2	3	4	5
Y1	Y2	Y3	Y4	Y5	e1	e2	e3	e4	E5
1	1	0	1	1	1	1	0	0	0

Fitness merupakan sebuah nilai yang digunakan untuk menentukan calon solusi, semakin besar nilai *fitness*nya maka kromosom tersebut semakin baik digunakan sebagai calon solusi permasalahan. Untuk menghitung nilai *fitness* digunakan Persamaan 2.8. Terlebih dahulu dihitung dahulu prediksi harga curah hujan setiap data menggunakan regresi pada Tabel 4.7 dengan menggunakan 5 periode untuk keseluruhan data curah hujan.

Tabel 4.7 Persamaan Regresi ARIMA Tiap Data

No.	Persamaan
1.	$Y'_1 = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-4} + \beta_4 Y_{t-5} + e_t - \beta_5 e_{t-1} - \beta_6 e_{t-2}$
2.	$Y'_2 = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-4} + \beta_4 Y_{t-5} + e_t - \beta_5 e_{t-1} - \beta_6 e_{t-2}$
3.	$Y'_3 = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-4} + \beta_4 Y_{t-5} + e_t - \beta_5 e_{t-1} - \beta_6 e_{t-2}$
4.	$Y'_4 = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-4} + \beta_4 Y_{t-5} + e_t - \beta_5 e_{t-1} - \beta_6 e_{t-2}$
5.	$Y'_5 = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-4} + \beta_4 Y_{t-5} + e_t - \beta_5 e_{t-1} - \beta_6 e_{t-2}$
6.	$Y'_6 = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-4} + \beta_4 Y_{t-5} + e_t - \beta_5 e_{t-1} - \beta_6 e_{t-2}$
7.	$Y'_7 = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-4} + \beta_4 Y_{t-5} + e_t - \beta_5 e_{t-1} - \beta_6 e_{t-2}$
8.	$Y'_8 = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-4} + \beta_4 Y_{t-5} + e_t - \beta_5 e_{t-1} - \beta_6 e_{t-2}$
9.	$Y'_9 = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-4} + \beta_4 Y_{t-5} + e_t - \beta_5 e_{t-1} - \beta_6 e_{t-2}$

Kemudian dihitung tingkat eror dengan Persamaan 2.7 diolah dari Tabel 4.7. Hasil perhitungan *error* ditampilkan pada Tabel 4.8 serta nilai *fitness* untuk *parent* 1 dapat dilihat pada Tabel 4.9.

1. Menghitung prediksi curah hujan dengan persamaan regresi ARIMA.

$$Y'_1 = C + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-4} + \beta_4 Y_{t-5} + e_t - \beta_5 e_{t-1} - \beta_6 e_{t-2}$$

$$Y'_1 = -1.587 + (0.00 * 6.9727) + (0.00 * 3.7272) + (0.00 * -8) + (0.00 * 4.375) + (0.00 * -3.875) + (-19.722) - (1 * -19.722) - (0.00 * 7.7507) - (0.00 * 4.505) - (0 * -7.22) - (0 * 5.153)$$

$$Y'_1 = -1.587$$

$$Y''_1 = 22.213$$

2. Menghitung nilai *error*.

$$MAPE = \frac{1}{6} \sum_{t=1}^6 \left| \frac{z_t - \hat{z}_t}{z_t} \right|$$

$$MAPE = \frac{1}{6} \left| \left(\frac{23.8 - (22.213)}{23.8} \right) \right| + \left| \left(\frac{3.3 - 1.713}{3.3} \right) \right| + \dots + \left| \left(\frac{3.2727 - (1.73)}{3.2727} \right) \right|$$

$$MAPE = 0.2293$$

Tabel 4.8 Hasil Perhitungan Prediksi Curah Hujan dan Nilai Error

No.	Y1	Y2	Y4	Y5	e_1	e_2	Y''
1.	6.97	22.213	22.213	22.213	22.213	22.213	22.213
2.	3.72	1.713	1.713	1.713	1.713	1.713	1.713
3.	-8	8.685727	8.685727	8.685727	8.685727	8.685727	8.6857
4.	4.375	12.413	12.413	12.413	12.413	12.413	12.413
5.	-3.875	4.413	4.413	4.413	4.413	4.413	4.413
6.	9.1	8.788	8.788	8.788	8.788	8.788	8.788
7.	-12.33	4.913	4.913	4.913	4.913	4.913	4.913
8.	3.027	14.013	14.013	14.013	14.013	14.013	14.013
9.	3	1.685727	1.685727	1.685727	1.685727	1.685727	1.6857
ϵ							0.2293

3. Menghitung nilai *fitness*.

$$f = \frac{1}{(\epsilon + \alpha)}$$

$$f = \frac{1}{(0.2292 + 0.0001)}$$

$$f = 4.3592$$

Tabel 4.9 Hasil Perhitungan *Fitness Parent* 1

1	2	3	4	5	6	7	8	9	10	<i>Fitness</i>
Y1	Y2	Y3	Y4	Y5	e1	e2	e3	e4	e5	$\frac{1}{(\epsilon + \infty)}$
1	1	0	1	1	1	1	0	0	0	4.3592

4.3.2 Inisialisasi Populasi Awal

Tahap pertama adalah menentukan populasi awal, dimana jumlah kromosom dibentuk sejumlah ukuran populasi (*popSize*) yang telah ditentukan sebelumnya yakni sebesar 5. Sehingga dalam satu populasi berisi kromosom (*parent*) sebanyak 5, representasi tersebut dapat dilihat pada Tabel 4.10 berikut:

Tabel 4.10 Data populasi awal

<i>Parent</i>	Kromosom									
	AR					MA				
P1	1	1	0	1	0	1	0	0	1	1
P2	1	0	0	1	1	1	0	1	1	0
P3	0	0	1	0	1	0	0	1	0	1
P4	0	1	1	0	1	1	0	0	1	0
P5	0	1	1	0	0	1	0	0	0	0

4.3.3 Reproduksi

Tahap berikutnya setelah didapatkan populasi awal adalah tahapan reproduksi. Algoritma genetika memiliki 2 operator genetika, yaitu *crossover* dan *mutation*. Tahap ini berfungsi menghasilkan keturunan (*offspring*) terhadap *parent* pada populasi awal.

4.3.3.1 Crossover

Proses *crossover* menggunakan metode *two-cut-point crossover*. Jumlah kromosom anak (*offspring*) yang dihasilkan didapatkan dari perkalian *crossover rate* (*cr*) dengan jumlah populasi (*popSize*). Populasi awal berjumlah 5 dengan *cr* sebesar 0.4 maka akan menghasilkan *offspring* $5 \times 0.4 = 2$ ($C_1 - C_2$). Tahapan *two-cut-point crossover* adalah sebagai berikut:

1. Memilih secara acak 2 kromosom *parent* berbeda.
2. Menentukan titik awal dan titik akhir sebagai *cut point*.
3. Gen yang berada diantara titik potong akan di pindah silang.

4. Hasil pindah silang tersebut adalah hasil *offspring*.

Tabel 4.11 Hasil *two cut point cross over*

P3	0	0	1	0	1	0	0	1	0	1
P5	0	1	1	0	0	1	0	0	0	0
C1	0	0	1	0	0	1	0	1	0	1
C2	0	1	1	0	1	0	0	0	0	0

4.3.3.2 Mutation

Proses mutasi menggunakan metode *random mutation*. Kromosom anak (*offspring*) dihasilkan dengan perkalian *mutation rate* (*mr*) terhadap jumlah populasi (*popSize*) sehingga didapatkan jumlah *offspring* $5 \times 0.1 = 0.5$ dibulatkan menjadi 1. Tahap *random mutation* dapat dijelaskan dengan langkah-langkah seperti berikut:

1. Memilih secara acak 1 kromosom *parent*.
2. Menentukan titik mutasi.
3. Melakukan mutasi terhadap gen apabila nilai awal bernilai 1 maka berubah menjadi 0, begitu juga sebaliknya.
4. Hasil mutasi tersebut merupakan kromosom *offspring*.

Tabel 4.12 Hasil *random mutation*

P1	1	1	0	1	0	1	0	0	1	1
C3	1	0	0	1	0	1	0	0	1	1

4.3.4 Evaluasi dan Seleksi

Seluruh kromosom yang ada baik itu *parent* maupun *offspring* harus dilakukan evaluasi yakni dengan menghitung nilai *fitness* dari kromosom tersebut. Proses menghitung nilai *fitness* adalah dengan menghitung nilai MAPE terkecil dari kemungkinan model yang terbentuk dari kromosom, untuk itu kromosom biner perlu ditransformasikan kedalam kemungkinan model ARIMA seperti pada Tabel 4.13 berikut:



Tabel 4.13 Transformasi kromosom biner menjadi model ARIMA

Kromosom	AR					MA					MODEL
P1	1	1	0	1	0	1	0	0	1	1	(([1,2,4], d, [1,4,5])
P2	1	0	0	1	1	1	0	1	1	0	(([1,4,5], d, [1,3,4])
P3	0	0	1	0	1	0	0	1	0	1	(([3,5], d, [3,5])
P4	0	1	1	0	1	1	0	0	1	0	(([2,3,5], d, [1,3])
P5	0	1	1	0	0	1	0	0	0	0	(([2,3], d, [1])
C1	0	0	1	0	0	1	0	1	0	1	(([3], d, [1,3,5])
C2	0	1	1	0	1	0	0	0	0	0	(([2,3,5], d, 0)
C3	1	0	0	1	0	1	0	0	1	1	(([1,4], d, [1,4,5])

Setelah didapatkan representasi model ARIMA masing-masing model dihitung nilai MAPE, kemudian untuk menghitung nilai *fitness* dengan Persamaan 2.7. Nilai *fitness* hasil Tabel 4.13 dapat dilihat pada Tabel 4.14 berikut:

Tabel 4.14 Hasil nilai *fitness* dari seluruh kromosom

No.	Kromosom	Model	MAPE	$Fitness \left(\frac{1}{(\epsilon + \alpha)} \right)$
1	P1	(([1,2,4], d, [1,4,5])	0.229	4.3592
2	P2	(([1,4,5], d, [1,3,4])	0.229	4.3592
3	P3	(([3,5], d, [3,5])	5.78759	0.17278
4	P4	(([2,3,5], d, [1,3])	0.229	4.3592
5	P5	(([2,3], d, [1])	0.229	4.3592
6	C1	(([3], d, [1,3,5])	0.229	4.3592
7	C2	(([2,3,5], d, 0)	0.8693	1.15
8	C3	(([1,4], d, [1,4,5])	0.229	4.3592

Tahap selanjutnya adalah melakukan proses seleksi terhadap kromosom *parent* dan *offspring*. Proses seleksi memungkinkan terbentuknya kromosom baru dengan solusi yang lebih baik, oleh karena itu proses seleksi adalah mengeliminasi kromosom dengan nilai *fitness* yang rendah dan mempertahankan kromosom dengan nilai *fitness* yang baik. Proses seleksi menggunakan metode *elitism selection* dimana semua kromosom *parent* dan *offspring* dikumpulkan didalam satu penampung, individu dengan nilai *fitness* yang lebih besar dibanding individu lain akan tetap dipertahankan.

Jumlah individu terbaik yang dipertahankan sesuai dengan *popSize* yang ditentukan sebelumnya. Menurut Mahmudy (2013) proses *elitism selection* akan selalu menghasilkan individu terbaik yang dapat lolos dalam seleksi. Hasil akhir dari proses seleksi individu terbaik dengan jumlah sesuai populasi awal yang dapat berlanjut kedalam generasi selanjutnya hal ini ditampilkan pada Tabel 4.15 berikut.

Tabel 4.15 Hasil seleksi *elitism selection*

No	Kromosom	Model	<i>Fitness</i>
1	P1	([1,2,4], d, [1,4,5])	4.3592
2	P2	([1,4,5], d, [1,3,4])	4.3592
3	P4	([3,5], d, [3,5])	4.3592
4	P5	([2,3,5], d, [1,3])	4.3592
5	C1	([2,3], d, [1])	4.3592
6	C3	([3], d, [1,3,5])	4.3592
7	C2	([2,3,5], d, 0)	1.15
8	P3	([1,4], d, [1,4,5])	0.17278

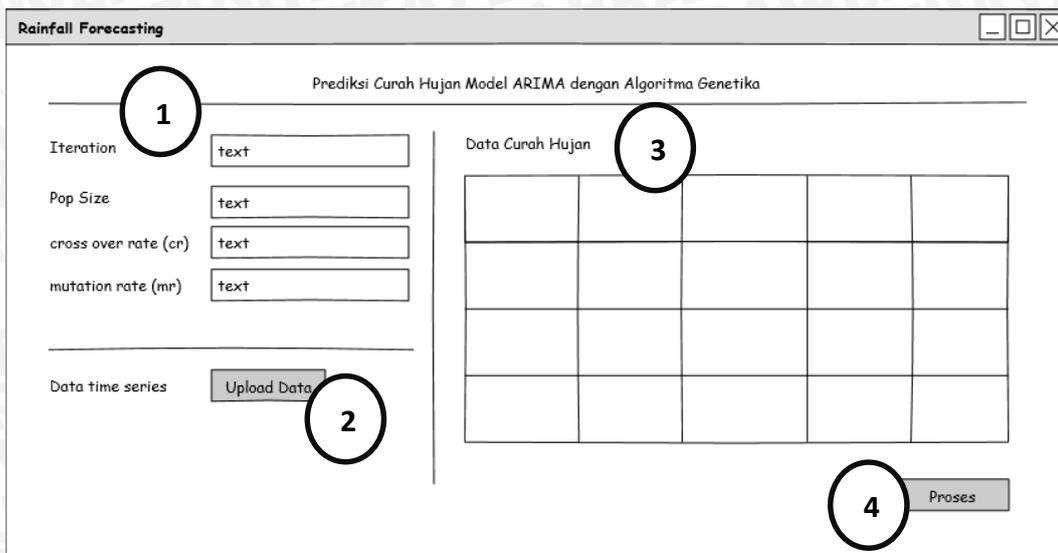
Melalui proses seleksi seperti pada Tabel 4.15 dapat dilanjutkan dengan memilih kromosom berdasarkan nilai *fitness* terbesar, dimana nilai *fitness* pada Tabel 4.15 kromosom **P1**, **P2**, **P4**, **P5**, **C1**, dan **C3** memiliki nilai *fitness* yang sama. Oleh karena itu dalam memilih kromosom terbaik adalah dengan cara mengambil kromosom yang berada pada peringkat pertama. Sehingga individu yang memiliki nilai *fitness* terbesar serta berada pada peringkat pertama adalah kromosom **P1**. Hasil dari proses algoritma genetika menghasilkan kromosom **P1** yang menjadi kandidat model solusi terbaik, maka didapatkan model ARIMA ([1,2,4], d, [1,4,5]) sebagai model terbaik dalam memprediksi curah hujan.

4.4 Perancangan *User Interface*

Pada perancangan *user interface* aplikasi terdiri dari tiga halaman diantaranya halaman awal untuk mengatur nilai parameter algoritma genetika dan data curah hujan, halaman berikutnya untuk menampilkan hasil proses algoritma genetika, dan yang terakhir halaman untuk melakukan prediksi curah hujan berdasarkan model ARIMA terbaik hasil pembentukan dengan algoritma genetika.

4.4.1 Tampilan Halaman Utama

Menampilkan halaman utama disini nilai parameter dari algoritma genetika ditentukan, serta merupakan halaman untuk input data curah hujan, desain *interface* seperti tampak pada Gambar 4.2 berikut



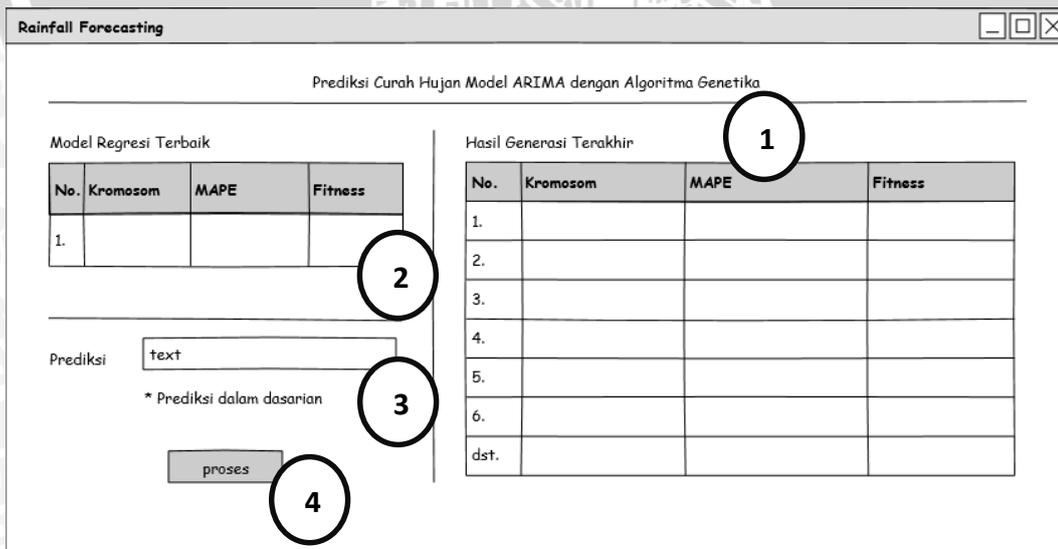
Gambar 4.2 Rancangan halaman utama

Keterangan:

1. *TextBox* untuk memasukan nilai parameter algoritma genetika.
2. *Button* memasukan Data Curah Hujan *time series*.
3. *DataGridView* untuk menampilkan data curah hujan.
4. *Button* untuk proses data dengan algoritma genetika.

4.4.2 Tampilan Halaman Algoritma Genetika

Berupa halaman yang menampilkan hasil proses dari algoritma genetika untuk menentukan model ARIMA terbaik yang dapat digunakan dalam prediksi curah hujan. Rancang halaman dapat dilihat pada Gambar 4.3 berikut:



Gambar 4.3 Rancangan halaman proses algoritma genetika-ARIMA

Keterangan:

1. Tabel untuk menampilkan hasil generasi algoritma genetika terakhir.
2. Tabel untuk menampilkan hasil generasi algoritma dengan nilai *fitness* terbaik.
3. *TextBox* untuk memasukan banyaknya prediksi yang diinginkan.
4. *Button* untuk melakukan proses prediksi dengan model ARIMA terbaik.

4.4.3 Tampilan Halaman Prediksi

Berupa halaman yang berfungsi menampilkan nilai prediksi berdasarkan model ARIMA yang dibentuk dengan algoritma genetika dengan rentang waktu dasarian yang telah ditetapkan. Perancangan untuk halaman prediksi seperti pada Gambar 4.4.

Hasil Prediksi

Prediksi Curah Hujan Data Dasarian

No	Prediksi
1.	
2.	
3.	
dst	

Close

Gambar 4.4 Rancangan halaman prediksi dengan model ARIMA terbaik

Keterangan:

1. Tabel untuk menampilkan hasil prediksi sejumlah nilai yang ditentukan sebelumnya.
2. *Button* untuk menutup halaman hasil prediksi.

BAB 5 IMPLEMENTASI

Implementasi berfungsi untuk menjelaskan bagaimana proses perancangan serta pembuatan *prototype* dari program prediksi curah hujan menggunakan model ARIMA yang dibentuk dengan algoritma genetika.

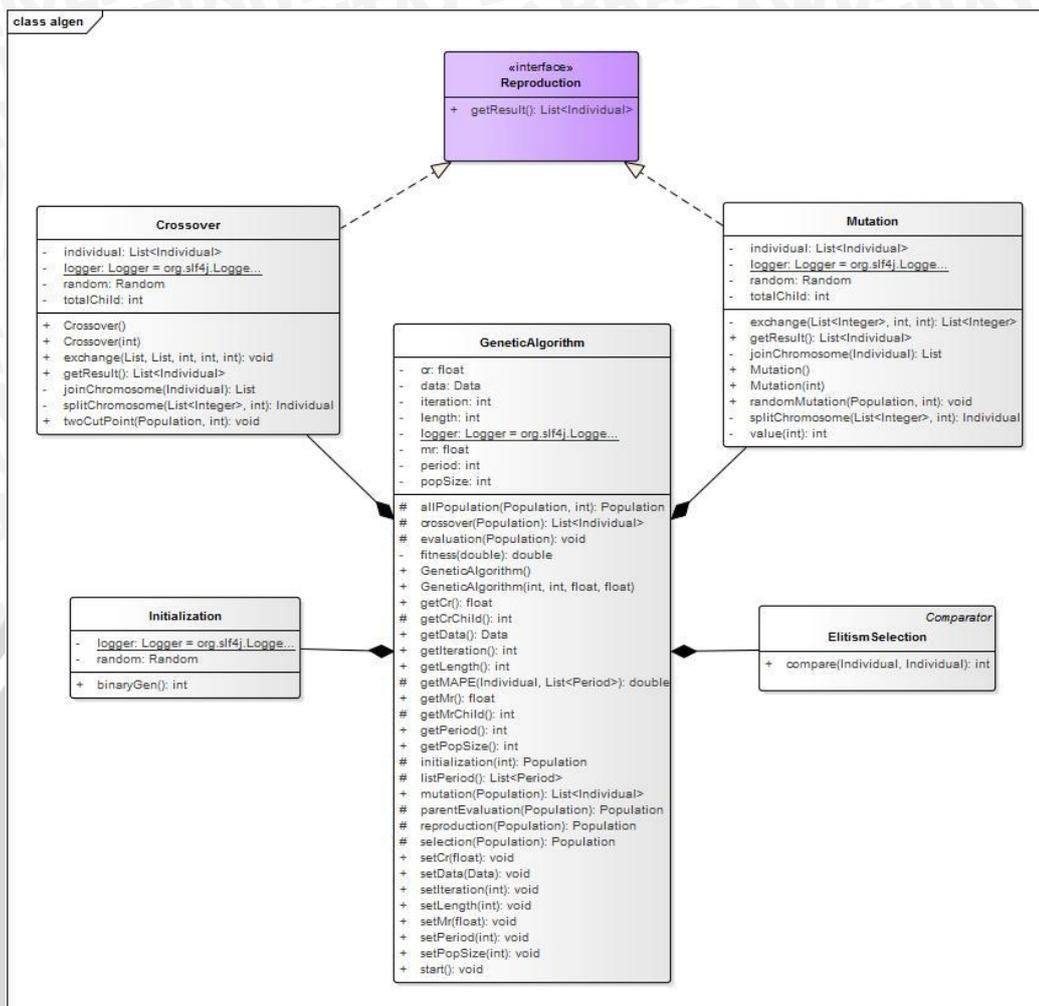
5.1 Struktur Class

Prototype program dikembangkan dengan *Object-Oriented Analysis & Design* (OOAD) menggunakan bahasa pemrograman Java. Sehingga dalam melakukan implementasi diperlukan perancangan sebagai dasar dalam proses *development*. OOAD memiliki diagram perancangan yang disebut *Unified Modeling Language* (UML) yang mengatur mengenai perancangan, salah satunya adalah dengan membuat *Class Diagram*.

5.1.1 Class Algoritma Genetika

Rancang *Class Diagram* untuk proses algoritma genetika dalam membangun model regresi ARIMA dengan algoritma genetika dapat dilihat pada Gambar 5.1. Disini proses utama algoritma genetika terdapat pada *class* GeneticAlgorithms dimana proses inialisasi populasi, reproduksi, evaluasi dan seleksi dilakukan. Disini proses reproduksi dibuatkan *class* tersendiri dengan mengimplementasi *interface* Reproduction.

Pada rancangan *class* GeneticAlgorithm memiliki asosiasi dengan *class* Crossover guna melakukan proses *two-cut-point crossover* terhadap individu yang terdapat pada populasi, *class* Mutation guna menerapkan metode *random mutation* terhadap individu dalam populasi. *Class* ElitismSelection berfungsi untuk melakukan komparasi terhadap nilai *fitness* tiap individu dalam populasi untuk mendapatkan kromosom dengan nilai *fitness* terbaik.

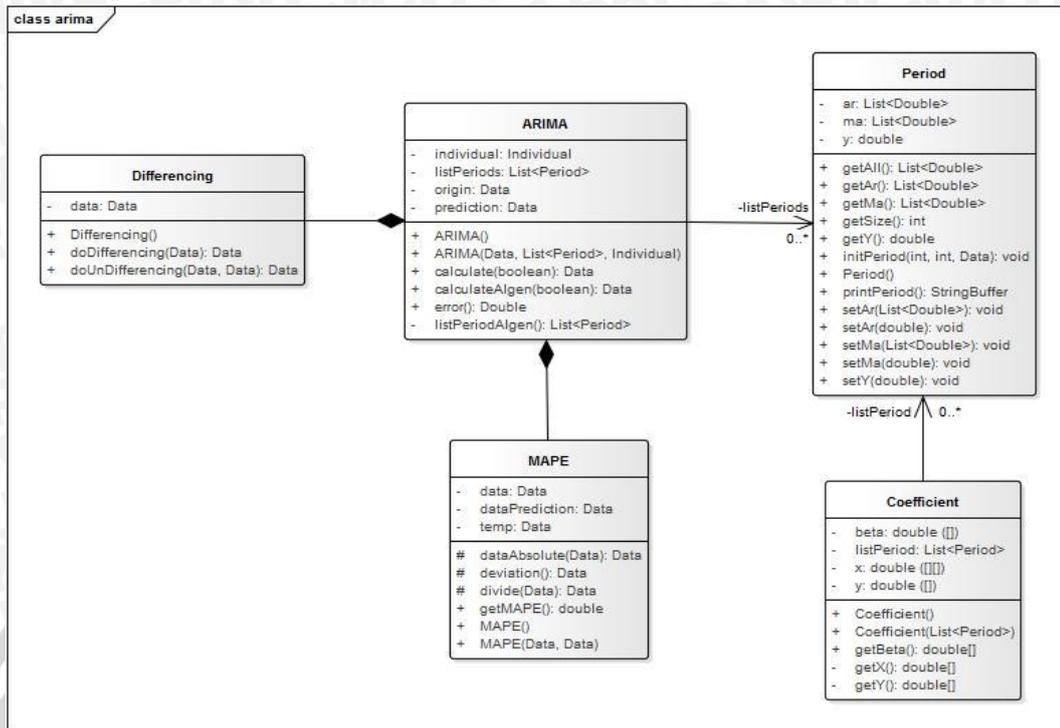


Gambar 5.1 Class Diagram Algoritma Genetika

5.1.2 Class Regresi ARIMA

Rancang *Class* diagram untuk membuat model regresi ARIMA dapat dilihat pada Gambar 5.2. *Class* ARIMA berfungsi untuk melakukan segala tahap yang terdapat pada proses regresi model ARIMA. Mulai dari melakukan *differencing* terhadap data yang tidak stasioner serta *undifferencing* untuk mendapatkan data prediksi yang sesungguhnya, menghitung prediksi menggunakan model AR, MA, ARMA / ARIMA.

Dalam menghitung nilai *fitness* diperlukan nilai *error* untuk tiap individu, nilai *error* didapatkan dari hasil perhitungan MAPE yang didapatkan dalam perbandingan nilai data aktual dengan nilai hasil prediksi yang dihasilkan oleh model regresi ARIMA yang telah dibentuk dengan algoritma genetika.



Gambar 5.2 Class Diagram Model Regresi ARIMA

5.2 Source Code

5.2.1 Mengambil Data *time series* Curah Hujan

Proses membaca data dalam format .xls yang digunakan sebagai membaca data *time series* dapat dilihat pada Source Code 5.1. Proses membaca file berekstensi .xls menggunakan *library* apache-poi. Hasil pembacaan data terhadap file .xls akan disimpan dalam *Object* Data, dimana akan berisi fungsi untuk menghitung nilai *mean* dan nilai *error term*.

```

1 public void readData() throws IOException{
2     data = new ArrayList<Double>();
3     ExcelFile excelFile = new ExcelFile();
4     double value = 0.0;
5     Workbook workbook = excelFile.getWorkbook(inputStream, path);
6     //mendapatkan sheet excel 1
7     Sheet sheet = workbook.getSheetAt(0);
8     //mendapatkan jumlah row yang terdapat data
9     Iterator<Row> iteratorRow = sheet.iterator();
10    while (iteratorRow.hasNext()) {
11        Row nextRow = iteratorRow.next();
12        //mendapatkan jumlah cell yang terdapat data
13        Iterator<Cell> cellIterator = nextRow.cellIterator();
14        while (cellIterator.hasNext()){
15            Cell nextCell = cellIterator.next();
16            //mendapatkan posisi index column
17            int columnIndex = nextCell.getColumnIndex();
18            switch (columnIndex){//column dimulai dari 0
19                case 0:
20                    value = (Double)
21                    excelFile.getCellValue(nextCell);
22                    //rainfall.setData((Double)
23                    excelFile.getCellValue(nextCell));

```

```

24         break;
25     }
26 }
27     data.add(value);
28 }
29     workbook.close();
30     inputStream.close();
31 }

```

Source Code 5.1 Proses Membaca Data

5.2.2 Pembangkitan Populasi Awal

Proses membangkitkan populasi awal terdiri dari proses pembentukan individu yang memiliki kromosom solusi dalam bentuk biner. Pada Source Code 5.2 merupakan tahap membentuk *chromosome* yang berfungsi mewakili solusi untuk model ARIMA AR (p) dan MA (q).

```

1 public void createChromosome(){
2     Initialization init = new Initialization();
3     //kromosom p
4     for(int i=0; i<getLength(); i++){
5         int pGen = init.binaryGen();
6         pChromosome.add(pGen);
7     }
8     //kromosom q
9     for(int j=0; j<getLength(); j++){
10        int qGen = init.binaryGen();
11        qChromosome.add(qGen);
12    }
13 }

```

Source Code 5.2 Membangkitkan Kromosom Biner

Pada Source Code 5.3 merupakan proses pembentukan populasi, dimana membangkitkan individu sebanyak *popSize* yang juga secara tidak langsung melakukan pembentukan kromosom dengan panjang sama seperti jumlah periode data yang akan digunakan dalam perhitungan nantinya.

```

1 protected Population initialization(){
2     Population population = new Population(this.popSize);
3     population.initPopulation(this.length);
4     return population;
5 }

```

Source Code 5.3 Melakukan Inisialisasi Populasi Awal Sebesar popSize

5.2.3 Perhitungan *Fitness*

Perhitungan nilai *fitness* didapatkan dari satu per nilai *error* ditambah koefisien dengan nilai yang sangat kecil untuk satu individu. Nilai *error* didapatkan dari perhitungan MAPE terhadap model ARIMA. Source Code 5.4 menjelaskan tahapan proses dalam mendapatkan nilai MAPE bagi individu yang akan dicari nilai *fitness*.

```

1 protected Double error(List<Period> listPeriods, Individual
2 individual, boolean undiff){
3     double error = 100.0;
4     ARIMA arima = new ARIMA(this.origin, listPeriods,
5 individual);
6     try {
7         arima.calculateAlgen(undiff);
8         error = arima.error();

```

```

9      } catch (Exception e) {
10         // TODO: handle exception
11     }
12     return error;
13 }

```

Source Code 5.4 Proses Mencari Nilai MAPE

Source Code 5.5 merupakan langkah dalam menghitung nilai *fitness* yang didapatkan dari 1 per nilai MAPE ditambahkan variabel *a* dengan nilai yang sangat kecil sehingga menghindari apabila terdapat nilai MAPE berupa 0.

```

1 private double fitness(double mape){
2     //membuat format angka dibelakang koma
3     DecimalFormat df = new DecimalFormat("#.#####");
4     df.setRoundingMode(RoundingMode.CEILING);
5     double a = 0.00001;
6     double result = (double)1/(mape+a);
7     String txtResult = df.format(result);
8     return Double.parseDouble(txtResult);
9 }

```

Source Code 5.5 Proses Menghitung *fitness*

5.2.4 Proses Crossover

Proses *crossover* menggunakan metode *two-cut-point* dimana membangkitkan titik potong awal dan titik potong akhir secara acak. Kemudian memilih dua *parent* secara acak untuk dilakukan proses *crossover*. Tahapan proses metode *two-cut-point* dapat dilihat pada Source Code 5.6 sedangkan tahapan pertukaran gen pada kromosom yang terpilih terdapat pada Source Code 5.7.

```

1 public void twoCutPoint(Population population, int genLength){
2     int i=0;
3     while(i<totalChild){
4         Individual P1 = new Individual();
5         Individual P2 = new Individual();
6         boolean parent = false;
7         boolean point = false;
8         int limit = population.getSize();
9         int parent1 = random.nextInt(limit);
10        int parent2 = random.nextInt(limit);
11        int point1 = random.nextInt(genLength);
12        int point2 = random.nextInt(genLength);
13        //chek tidak boleh parent sama
14        while(parent != true){
15            if(parent1 == parent2){
16                parent2 = random.nextInt(limit);
17            }else{
18                parent = true;
19            }
20        }
21        //check p1 tidak boleh diatas p2
22        while(point != true){
23            if(point1 >= point2){
24                point1 = random.nextInt(genLength);
25                point2 = random.nextInt(genLength);
26            }else{
27                point = true;
28            }
29        }
30        P1 = population.getIndividu(parent1);
31        List temp1 = joinChromosome(P1);

```

```

32     P2 = population.getIndividu(parent2);
33     List temp2 = joinChromosome(P2);
34     exchange(temp1, temp2, point1, point2, genLength);
35     //karena akan terbentuk 2 child dari 2 parent
36     i += 2;
37 }
38 }

```

Source Code 5.6 Crossover dengan Metode Two-Cut-Point

```

1  public void exchange(List parent1, List parent2, int point1, int
2  point2, int genLength){
3      ArrayList<Integer> temp1 = new ArrayList<Integer>();
4      ArrayList<Integer> temp2 = new ArrayList<Integer>();
5      int i = point1;
6      int j = point1;
7      int p=0;//sebagai index mengambil array temp
8      while(i<point2){
9          temp1.add((Integer) parent2.get(i));
10         temp2.add((Integer) parent1.get(i));
11         i++;
12     }
13     while(j<point2){
14         parent1.set(j, temp1.get(p));
15         parent2.set(j, temp2.get(p));
16         j++;p++;
17     }
18     this.individual.add(splitChromosome(parent1, genLength));
19     this.individual.add(splitChromosome(parent2, genLength));
20 }

```

Source Code 5.7 Proses Pertukaran Gen pada Kromosom

5.2.5 Proses Mutasi

Proses mutasi menggunakan metode *random mutation* diimplementasikan kedalam Source Code 5.8 sedangkan proses pertukaran nilai gen pada Source Code 5.9. *Random Mutation* berfungsi menukar nilai gen yang terpilih, jika awalnya bernilai 1 maka akan dirubah menjadi 0 begitu juga sebaliknya.

```

1  public void randomMutation(Population population, int genLength){
2      int i=0;
3      while(i < this.totalChild){
4          Individual individu = new Individual();
5          Individual mutationIndividu = new Individual();
6          int limit = population.getSize();
7          //memilih parent secara acak
8          int parent = random.nextInt(limit);
9          //memilih gen secara acak
10         int gen = random.nextInt(genLength);
11         //assign parent yang terpilih
12         individu = population.getIndividu(parent);
13         List<Integer> temp = joinChromosome(individu);
14         int value = value(temp.get(gen));
15         List<Integer> mutationTemp = exchange(temp, gen,
16 value);
17         mutationIndividu = splitChromosome(mutationTemp,
18 genLength);
19         this.individual.add(mutationIndividu);
20         i++;
21     }
22 }

```

Source Code 5.8 Mutasi dengan Metode *RandomMutation*

```

1 private List<Integer> exchange(List<Integer> temp, int position, int
2 value){
3     temp.set(position, value);
4     return temp;
5 }

```

Source Code 5.9 Proses Mutasi Gen pada Kromosom**5.2.6 Proses Seleksi dengan Metode Elitism**

Proses seleksi kromosom menggunakan metode *Elitism Selection* yang terlihat pada Source Code 5.10 dimana hanya kromosom dengan nilai *fitness* terbaik yang akan bertahan (Ong, et al., 2005). Proses seleksi sendiri adalah mengurutkan individu pada populasi dengan nilai *fitness* terbaik berada paling atas sedangkan selanjutnya adalah dengan nilai yang mendekati individu teratas.

```

1 protected Population selection(Population population){
2     Population newPop = new Population();
3     List<Individual> individu = population.getIndividual();
4     //Collections.sort(individu, new ElitismSelection());
5     Collections.sort(individu, new Comparator<Individual>() {
6         public int compare(Individual o1, Individual o2) {
7             // TODO Auto-generated method stub
8             //urutan besar ke kecil
9             return Double.compare(o2.getFitness(),
10 o1.getFitness());
11         }
12     });
13     for(Individual newInd : individu){
14         newPop.insertIndividu(newInd);
15     }
16     return newPop;
17 }

```

Source Code 5.10 Proses Seleksi dengan Metode *ElitismSelection***5.2.7 Proses Pemilihan Kromosom Terbaik**

Proses pemilihan kromosom terbaik dilakukan untuk membuat populasi baru yang akan digunakan dalam iterasi selanjutnya dengan memastikan populasi yang baru dibentuk terdiri dari individu dengan nilai *fitness* terbaik.

```

1 protected Population parentEvaluation(Population newPopulation){
2     Population population = new Population(popSize);
3     for(int i=0; i<this.popSize; i++){
4         Individual individual = newPopulation.getIndividu(i);
5         population.insertIndividu(individual);
6     }
7     return population;
8 }

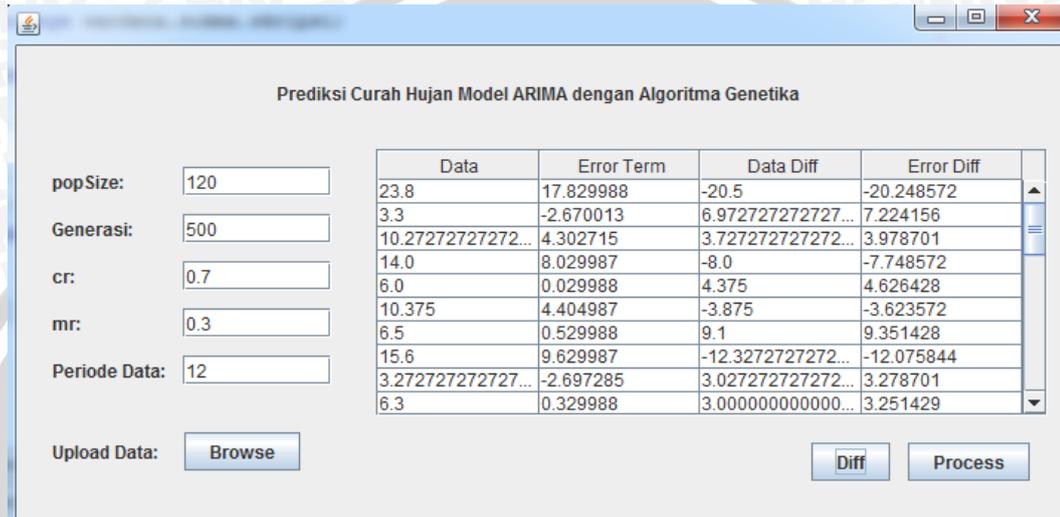
```

Source Code 5.11 Proses Pemilihan Kromosom Terbaik

5.3 Implementasi

Berikut adalah hasil implementasi *prototype*, terdiri dari halaman utama yang berfungsi memberikan input berupa *popSize*, generasi, *cr*, *mr*, dan juga periode data. Kemudian terdapat halaman yang menampilkan urutan individu terbaik dalam populasi yang telah dihasilkan selama proses algoritma genetika berlangsung.

5.3.1 Halaman Utama



Gambar 5.3 Halaman Awal dari *Prototype*

Gambar 5.3 merupakan tampilan antarmuka awal dari sistem *prototype* pembentukan model regresi ARIMA dengan algoritma genetika. Disini terdapat beberapa kolom untuk memberikan input parameter yang terdiri dari *popSize*, generasi, *cr*, *mr*, dan juga jumlah periode data. Data yang digunakan diupload terlebih dahulu menggunakan *button browse*, jika data dirasa perlu dilakukan *differencing* maka *button diff* berfungsi untuk melakukan *differencing*. Setelah data dan parameter selesai diinputkan tahap selanjutnya adalah menekan tombol *process* guna melakukan proses algoritma genetika.

5.3.2 Halaman Hasil Algoritma Genetika

No.	Kromosom	MAPE	Fitness
1	AR: [1][1][0][0][1][0][1][1][0][1][0][0]MA: [1][0][1][1][1][1][1]...	3.228594	0.309732
2	AR: [1][1][0][0][1][0][1][1][0][1][0][0]MA: [1][0][1][1][1][1][1]...	3.228594	0.309732
3	AR: [1][1][0][0][1][0][1][1][0][1][0][0]MA: [1][0][1][1][1][1][1]...	3.228594	0.309732
4	AR: [1][1][0][0][1][0][1][1][0][1][0][0]MA: [1][0][1][1][1][1][1]...	3.228594	0.309732
5	AR: [1][1][0][0][1][0][1][1][0][1][0][0]MA: [1][0][1][1][1][1][1]...	3.228594	0.309732
6	AR: [1][1][0][0][1][0][1][1][0][1][0][0]MA: [1][0][1][1][1][1][1]...	3.228594	0.309732
7	AR: [1][1][0][0][1][0][1][1][0][1][0][0]MA: [1][0][1][1][1][1][1]...	3.228594	0.309732
8	AR: [1][1][0][0][1][0][1][1][0][1][0][0]MA: [1][0][1][1][1][1][1]...	3.228594	0.309732
9	AR: [1][1][0][0][1][0][1][1][0][1][0][0]MA: [1][0][1][1][1][1][1]...	3.228594	0.309732

Gambar 5.4 Halaman Hasil Proses Algoritma Genetika

Gambar 5.4 merupakan tampilan antarmuka hasil populasi dari proses algoritma genetika pada sistem *prototype*. Pada tampilan ini diberikan sebuah tabel yang terdiri dari kromosom, nilai MAPE, dan juga nilai *fitness*. Pada tabel ini berisi populasi terbaik yang telah dihasilkan selama proses algoritma genetika dan juga telah melalui tahap seleksi sehingga individu paling teratas merupakan individu terbaik yang dihasilkan oleh algoritma genetika dalam membentuk model regresi ARIMA.



BAB 6 PEMBAHASAN

Pembahasan berisi mengenai hasil penelitian guna menjawab pertanyaan dari masalah penelitian. Berisi mengenai hasil pengujian beserta analisisnya serta pembahasan mengenai hasil penelitian. Dalam pengujian menggunakan data wilayah Puspo, Tenger selama satu tahun pada tahun 2014 yang dapat dilihat pada Lampiran A.

6.1 Uji Coba & Analisis Parameter Populasi

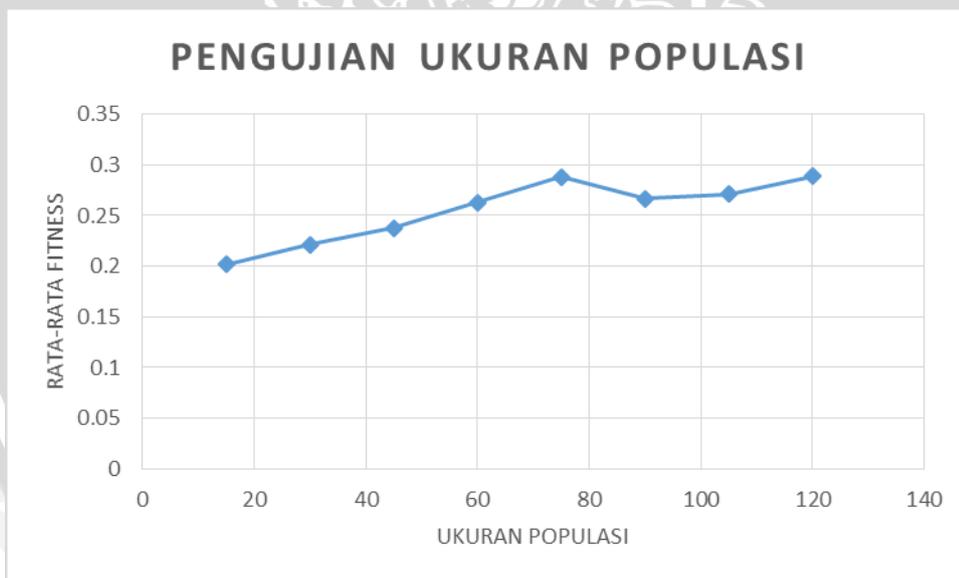
Dalam uji coba parameter populasi digunakan rentang pengujian ukuran parameter populasi antara 15 – 120 dengan kelipatan 15. Sedangkan untuk parameter algoritma yang lain diantaranya untuk kombinasi cr dan mr adalah 0.8 : 0.2, jumlah generasi sebanyak 250, dan penggunaan periode data sebanyak 6 periode. Masing-masing penggunaan jumlah populasi dilakukan pengujian sebanyak 10 kali yang kemudian dihitung nilai *fitness* rata-ratanya. Pengujian ini dimaksudkan untuk mendapatkan parameter populasi terbaik yang dapat menghasilkan solusi masalah yang optimal. Hasil dari pengujian parameter populasi dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Uji Coba Parameter Populasi

Ukuran Populasi	Nilai <i>Fitness</i> dari Percobaan ke-										Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	9	10	
15	0.1 53 49 3	0.2 07 80 7	0.1 81 28 4	0.1 85 23 5	0.2 45 75 8	0.2 06 91 1	0.2 39 72 9	0.2 45 75 8	0.2 20 55	0.1 32 50 7	0.2019032
30	0.2 45 75 8	0.2 52 63 2	0.1 63 72 6	0.2 52 63 2	0.1 69 03 4	0.2 06 91 1	0.2 52 63 2	0.2 45 75 8	0.2 52 63 2	0.1 73 56 4	0.2215279
45	0.2 93 26 6	0.2 45 75 8	0.1 63 72 6	0.1 74 86 1	0.1 85 23 5	0.2 93 26 6	0.2 52 63 2	0.2 93 26 6	0.2 91 30 2	0.1 85 23 5	0.2378547
60	0.2 93 26 6	0.2 45 75 8	0.2 91 30 2	0.2 45 75 8	0.2 93 26 6	0.2 93 26 6	0.2 52 63 2	0.2 45 75 8	0.2 45 75 8	0.2 20 55	0.2627314

Tabel 6.1 (lanjutan)

75	0.2 91 30 2	0.2 93 26 6	0.2 91 30 2	0.2 91 30 2	0.2 91 30 2	0.2 93 26 6	0.2 52 63 2	0.2 93 26 6	0.2 91 30 2	0.2 91 30 2	0.2880242
90	0.2 45 75 8	0.2 52 63 2	0.2 45 75 8	0.2 52 63 2	0.2 93 26 6	0.2 93 26 6	0.2 45 75 8	0.2 93 26 6	0.2 93 26 6	0.2 52 63 2	0.2668234
105	0.2 93 26 6	0.2 45 75 8	0.2 93 26 6	0.2 45 75 8	0.2 52 63 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 52 63 2	0.2 45 75 8	0.2708868
120	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 91 30 2	0.2 52 63 2	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2888098



Gambar 6.1 Grafik Pengujian Parameter Populasi

Hasil pengujian parameter populasi pada Tabel 6.1 direpresentasikan kedalam bentuk grafik yang dapat dilihat pada Gambar 6.1. Mahmudy, (2013) menyatakan bahwa penggunaan jumlah populasi yang semakin banyak akan menghasilkan nilai *fitness* yang semakin baik, hal ini juga dibuktikan oleh penelitian oleh Rahmi, et al., (2015) dan Permatasari & Mahmudy, (2015) yang mendapatkan hasil dimana semakin banyak jumlah populasi yang digunakan maka akan didapatkan grafik yang menanjak.

Hal tersebut juga terjadi pada hasil pengujian seperti yang dapat dilihat pada Gambar 6.1 semakin banyak nilai populasi yang digunakan maka rata-rata *fitness* yang dihasilkan semakin baik. Namun terdapat sesuatu yang cukup unik dimana terjadi penurunan nilai rata-rata *fitness* dengan jumlah populasi sebanyak 95 dibanding dengan sebelumnya. Meski demikian, grafik menunjukkan terjadinya kembali peningkatan nilai rata-rata *fitness*. Jika melihat hasil nilai rata-rata *fitness* terbaik adalah penggunaan populasi sebanyak 75 dan 120 hal ini wajar karena penggunaan populasi yang sedikit akan menghambat area eksplorasi dalam mencari solusi, namun semakin banyak jumlah populasi juga dapat memperlambat proses komputasi dan belum tentu menghasilkan nilai *fitness* yang lebih baik lagi (Rahmi, et al., 2015).

Melihat kemungkinan adanya penurunan nilai rata-rata *fitness* seperti yang terjadi pada penggunaan populasi sebanyak 95, maka pengujian jumlah populasi tidak dilanjutkan melebihi 120 dikarenakan nilai rata-rata *fitness* hampir sama dengan penggunaan jumlah populasi sebanyak 75. Oleh karena itu dalam pengujian nilai parameter populasi terbaik adalah penggunaan jumlah populasi didapatkan sebanyak 120 dengan nilai rata-rata *fitness* sebesar 0.2888098.

6.2 Uji Coba & Analisis Parameter Generasi

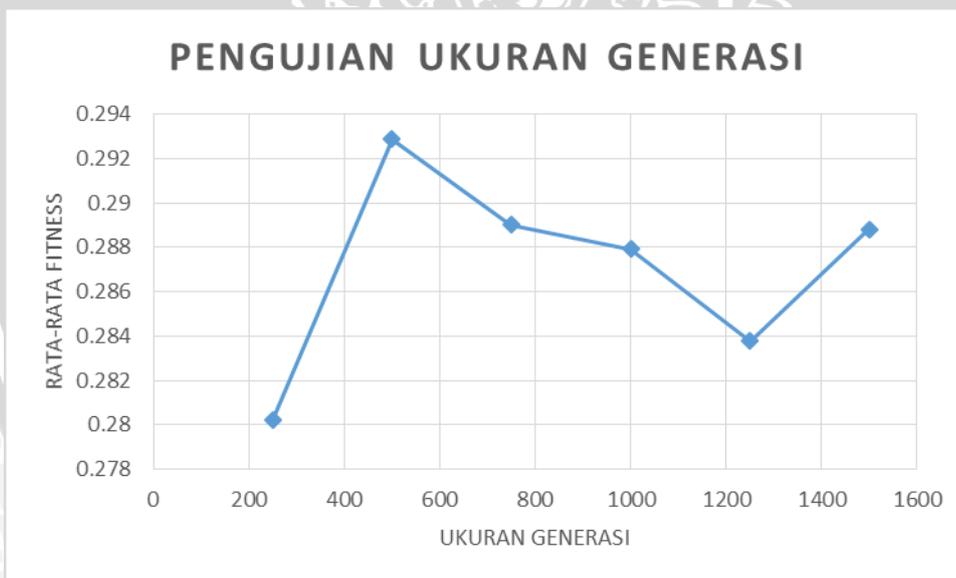
Pengujian parameter generasi menggunakan nilai parameter algoritma genetika sebagai berikut, untuk jumlah populasi digunakan hasil terbaik yang dihasilkan dalam pengujian sebelumnya yakni sebanyak 120. Untuk kombinasi *cr* dan *mr* adalah 0.8 : 0.2 dengan jumlah penggunaan periode data sebanyak 6 periode. Ukuran generasi sendiri dimulai dari rentang 250 hingga 1500 dengan kelipatan 250. Setiap ukuran generasi akan dilakukan pengujian sebanyak 10 kali guna mencari nilai parameter generasi paling baik dalam menentukan solusi optimal dalam menemukan solusi masalah. Hasil pengujian parameter generasi dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Uji Coba Parameter Generasi

Jumlah Generasi	Nilai <i>Fitness</i> dari Percobaan ke-										Rata-rata <i>fitness</i>	
	1	2	3	4	5	6	7	8	9	10		
250	0.2 93 26 6	0.2 93 26 6	0.2 45 75 8	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 52 63 2	0.2 52 63 2	0.2 91 30 2	0.280192
500	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2928732

Tabel 6.2 (lanjutan)

750	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 52 63 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2890062
1000	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 91 30 2	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 91 30 2	0.2 45 75 8	0.2 93 26 6	0.287926
1250	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 45 75 8	0.2 45 75 8	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2837644
1500	0.2 93 26 6	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 52 63 2	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2888098



Gambar 6.2 Grafik Pengujian Parameter Generasi

Penggunaan parameter generasi semakin banyak akan mengalami peningkatan perbaikan nilai rata-rata fitness dan pada titik tertentu akan mengalami kestabilan rata-rata fitness hingga kelipatan jumlah generasi selanjutnya, hal ini dialami oleh Rahmi, et al., (2015). Sehingga dalam penggunaan parameter generasi yang baik adalah tidak terlalu sedikit namun juga tidak terlalu banyak.

Melihat Gambar 6.2 menunjukkan grafik pengujian nilai parameter generasi mengalami perbedaan dengan hasil penelitian Rahmi, et al., (2015) dimana tidak terdapat titik kestabilan, justru mengalami grafik naik turun yang cukup signifikan. Dapat dilihat nilai rata-rata *fitness* terbaik didapat ketika jumlah generasi sebanyak 500, setelah itu nilai rata-rata *fitness* mengalami penurunan hingga titik dimana penggunaan jumlah generasi sebanyak 1250 setelah itu mengalami kenaikan yang cukup signifikan ketika penggunaan jumlah generasi sebanyak 1500.

Mengingat penggunaan jumlah generasi yang semakin banyak akan berakibat terhadap waktu komputasi yang cenderung membuang waktu serta grafik pada Gambar 6.2 menunjukkan penggunaan jumlah generasi paling baik adalah sebanyak 500 maka dapat dikatakan nilai parameter untuk jumlah generasi terbaik adalah sebanyak 500.

6.3 Uji Coba & Analisis Parameter Kombinasi cr & mr

Pengujian kombinasi cr dan mr bertujuan mencari kombinasi terbaik sehingga dapat menghasilkan solusi optimal. Penggunaan parameter algoritma yang lain adalah untuk parameter populasi merupakan hasil parameter terbaik pada pengujian sebelumnya yakni sebanyak 120, untuk parameter generasi juga menggunakan parameter terbaik pada pengujian sebelumnya yakni sebanyak 500, dengan penggunaan periode data sebanyak 6 periode. Hasil pengujian dapat dilihat pada Tabel 6.3.

Tabel 6.3 Hasil Uji Coba Kombinasi Parameter cr dan mr

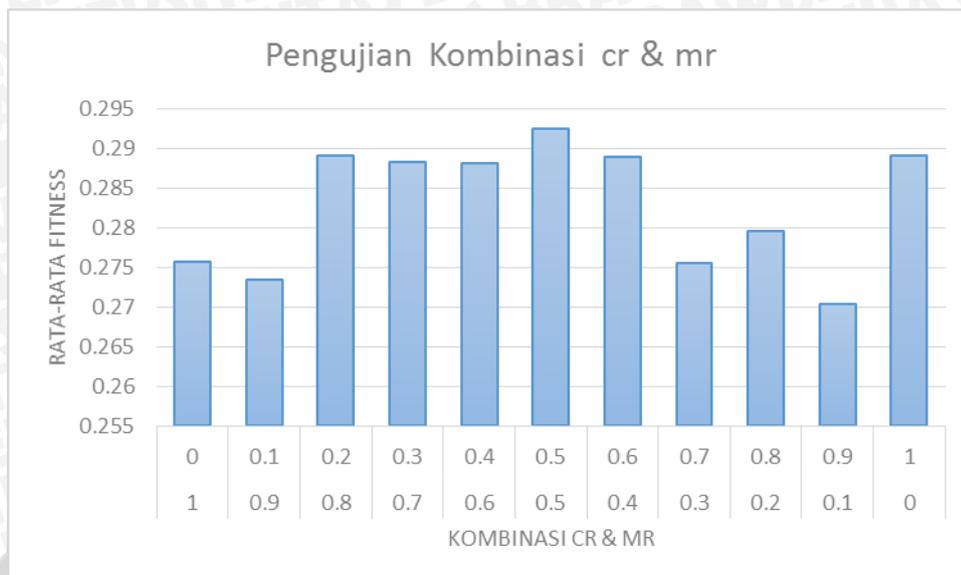
Kombinasi		Nilai <i>Fitness</i> dari Percobaan ke-										Rata-rata <i>fitness</i>	
		1	2	3	4	5	6	7	8	9	10		
cr	mr												
1	0	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.27566 22
		93	52	86	93	45	91	93	20	86	93		
		26	63	65	26	75	30	26	55	65	26		
		6	2	8	6	8	2	6	8	6			
0.9	0.1	0.2	0.2	0.1	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.27345 23
		93	93	85	91	93	93	45	93	52	93		
		26	26	23	30	26	26	75	26	63	26		
		6	6	5	2	6	6	8	6	2	6		
0.8	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.28900 62
		93	93	93	52	93	93	93	93	93	91		
		26	26	26	63	26	26	26	26	26	30		
		6	6	6	2	6	6	6	6	6	2		



Tabel 6.3 (lanjutan)

0.7	0.3	0.2 93 26 6	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 45 75 8	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.28831 88
0.6	0.4	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 45 75 8	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.28812 24
0.5	0.5	0.2 91 30 2	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 91 30 2	0.29248 04
0.4	0.6	0.2 93 26 6	0.2 93 26 6	0.2 91 30 2	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 52 63 2	0.2 93 26 6	0.28880 98
0.3	0.7	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 52 63 2	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 45 75 8	0.2 93 26 6	0.2 45 75 8	0.2 52 63 2	0.27544 12
0.2	0.8	0.2 93 26 6	0.2 93 26 6	0.2 45 75 8	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 45 75 8	0.2 52 63 2	0.2 93 26 6	0.2 93 26 6	0.27950 46
0.1	0.9	0.2 93 26 6	0.2 93 26 6	0.2 45 75 8	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 39 72 9	0.2 52 63 2	0.2 52 63 2	0.2 45 75 8	0.2 93 26 6	0.27028 39
0	1	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 52 63 2	0.2 93 26 6	0.2 93 26 6	0.28900 62

Penggunaan parameter kombinasi cr dan mr yang terbaik adalah untuk cr berkisar 0.3 – 0.8 sedangkan untuk mr 0.1 – 0.3 (Mahmudy, 2013). Sehingga kombinasi parameter cr dan mr yang ideal adalah kombinasi diantara nilai yang telah disebutkan.



Gambar 6.3 Grafik Pengujian Kombinasi Parameter cr & mr

Dalam grafik pada Gambar 6.3 menunjukkan bahwa kombinasi cr dan mr dalam rentang yang telah disebutkan diatas memang benar menunjukkan nilai rata-rata *fitness* terbaik dan dapat dibilang konsisten. Namun disini terjadi perbedaan asumsi bahwa penggunaan cr dengan rentang 0.4 – 0.8 sedangkan mr dengan rentang 0.2 – 0.6 yang dapat menghasilkan nilai rata-rata *fitness* terbaik dibanding kombinasi yang lain meski terdapat juga kombinasi cr dan mr yang cukup unik yakni 0 : 1 dimana menghasilkan rata-rata fitness sebesar 0.289.

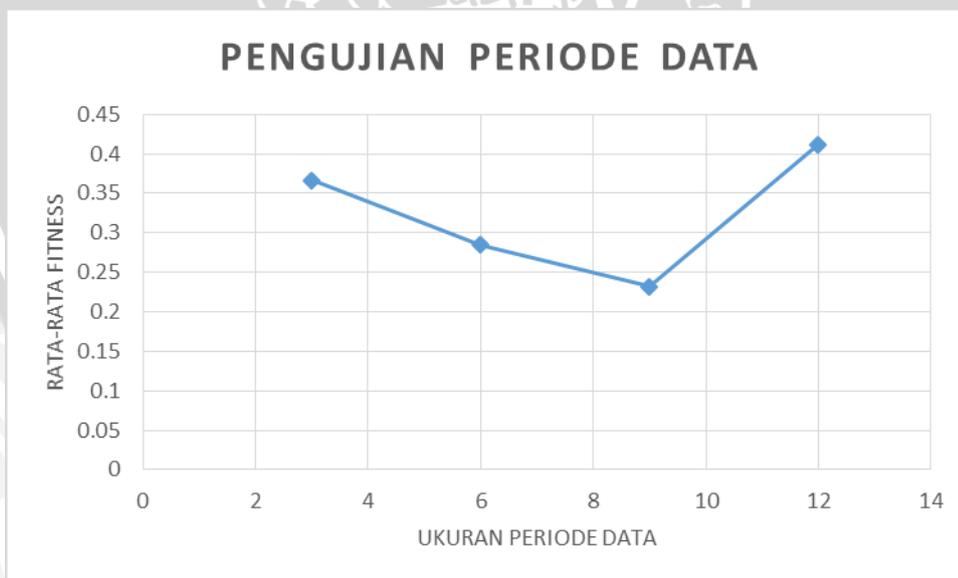
Meski demikian kombinasi parameter cr dan mr adalah untuk menghasilkan *offspring* yang berbeda dari *parent* untuk mendapatkan kromosom yang lebih baik. Sehingga Hasil pengujian menunjukkan bahwa kombinasi cr dan mr terbaik adalah 0.5 : 0.5 dengan nilai rata-rata *fitness* sebesar 0.2924804. Sehingga nilai parameter kombinasi cr dan mr terbaik untuk menghasilkan solusi yang optimal adalah cr : mr = 0.5 : 0.5.

6.4 Uji Coba & Analisis Periode Data Curah Hujan

Pengujian penggunaan periode data curah hujan dengan parameter algoritma genetika terbaik hasil pengujian sebelumnya diantara lain penggunaan parameter populasi sebanyak 120, ukuran generasi sebesar 500, dan kombinasi cr dan mr 0.5 : 0.5. Dengan rentang penggunaan periode data curah hujan antara 3 – 12 periode dengan kelipatan 3 serta pengujian sebanyak 10 kali untuk masing-masing kelipatan data hal ini mengingat jumlah data yang digunakan sebanyak 36 data. Hasil pengujian dapat dilihat pada Tabel 6.4.

Tabel 6.4 Hasil Uji Coba Periode Data Curah Hujan

Banyak Periode	Nilai <i>Fitness</i> dari Percobaan ke-										Rata-rata <i>fitness</i>	
	1	2	3	4	5	6	7	8	9	10		
3 Dasarian	0.3 66 73 2	0.3 66 73 2	0.3 66 73 2	0.3 66 73 2	0.3 66 73 2	0.3 66 73 2	0.3 66 73 2	0.3 66 73 2	0.3 66 73 2	0.3 66 73 2	0.3 66 73 2	0.366732
6 Dasarian	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 07 80 7	0.2 93 26 6	0.2 91 30 2	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2 93 26 6	0.2845237
9 Dasarian	0.2 43 82 5	0.2 12 72 5	0.2 53 76 7	0.2 10 13 3	0.2 53 76 7	0.2 14 76 2	0.2 09 02 9	0.2 53 76 7	0.2 53 76 7	0.2 13 67 2	0.2 13 67 2	0.2319214
12 Dasarian	0.4 12 05 8	0.4 85 68 5	0.4 31 93 9	0.3 81 52 1	0.3 54 01 9	0.3 56 38 2	0.3 96 43 8	0.4 02 91 5	0.3 55 77 5	0.5 42 08 8	0.2 13 67 2	0.4118815



Gambar 6.4 Grafik Pengujian Periode Data Curah Hujan

Pada Gambar 6.4 terlihat grafik pengujian menunjukkan penurunan nilai rata-rata *fitness* dimulai dengan penggunaan periode data sebanyak 6 dan 9 namun mengalami peningkatan yang cukup signifikan pada penggunaan periode data sebanyak 12 dengan nilai rata-rata *fitness* sebesar 0.4118815.

Sehingga pengujian periode data menunjukkan bahwa penggunaan periode data curah hujan paling optimal untuk mengatasi permasalahan membentuk model regresi ARIMA adalah sebanyak 12 periode data.

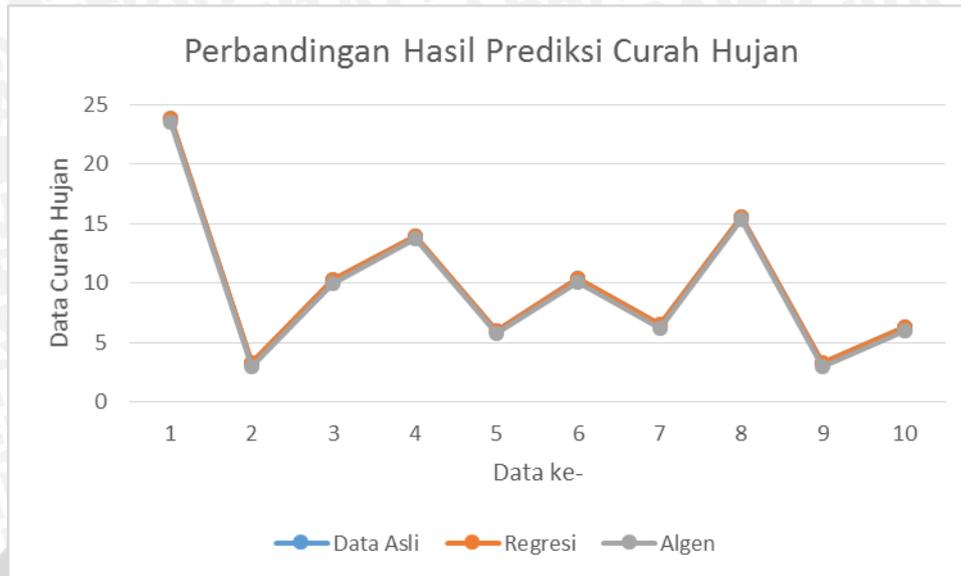
6.5 Analisa Hasil Perbandingan Prediksi Algoritma Genetika dengan Model Regresi Linier Berganda

Setelah melakukan pengujian untuk menentukan parameter algoritma genetika terbaik maka didapatkan model ARIMA yang dibentuk dengan algoritma genetika terbaik adalah ARIMA ([1,2,7,9,10,11], d, [1,3,7,11,12]). Untuk itu perlu dilakukan evaluasi apakah penelitian membangun model ARIMA dengan algoritma genetika nantinya mampu untuk digunakan sebagai model prediksi khususnya untuk data curah hujan. Proses evaluasi dilakukan dengan membandingkan hasil prediksi antara model ARIMA yang dibentuk dengan algoritma genetika terhadap model regresi linier berganda yang cukup populer digunakan sebagai model prediksi.

Dalam pengujian dilakukan perlakuan yang sama yakni penggunaan periode data sebanyak 12 periode, dengan data uji adalah data wilayah Puspo, Tengger selama periode tahun 2014 (Lampiran A). Hasil perbandingan pengujian dapat dilihat pada Tabel 6.5.

Tabel 6.5 Hasil Perbandingan dengan Regresi Linier Berganda

No.	Bulan	Dasarian	Data Aktual	Data Hasil Algen	Data Hasil Regresi Linier
1.	Januari	1	23.8	23.549	23.80017049
2.		2	3.3	3.049	3.300095305
3.		3	10.27272727	10.02172727	10.27287593
4.	Februari	1	14	13.749	14.0000806
5.		2	6	5.749	6.000185109
6.		3	10.375	10.124	10.37507656
7.	Maret	1	6.5	6.249	6.500127945
8.		2	15.6	15.349	15.60013538
9.		3	3.272727273	3.021727273	3.272741545
10.	April	1	6.3	6.049	6.3000298
MAPE				1.462213	0.000013
fitness				0.683848	986.976233



Gambar 6.5 Grafik Hasil Perbandingan Prediksi Curah Hujan

Melihat hasil perbandingan MAPE dan *fitness* pada Tabel 6.5 terlihat bahwa hasil terbaik didapatkan dengan penggunaan model regresi linier berganda. Dimana untuk nilai MAPE yang semakin besar berakibat nilai *fitness* yang semakin kecil yang berarti semakin buruk model tersebut untuk digunakan sebagai model prediksi.

Namun perlu diperhatikan grafik pada Gambar 6.5 menunjukkan bahwa hasil prediksi antara model ARIMA yang dibentuk algoritma genetika dengan model regresi linier berganda memiliki hasil prediksi yang sama akuratnya terhadap data asli. Perbedaan paling mencolok adalah terhadap penggunaan jumlah data, untuk regresi linier berganda menggunakan semua 12 periode data, sedangkan untuk model ARIMA yang dibentuk dengan algoritma genetika hanya menggunakan 6 dari 12 periode data.

Dalam menentukan model prediksi selain melihat dari tingkat *error* atau dari nilai *fitness* perlu juga dipertimbangkan jumlah penggunaan periode data, semakin sedikit penggunaan periode data maka akan semakin sedikit usaha yang diperlukan untuk melakukan prediksi. Oleh karena itu model ARIMA yang dibentuk dengan algoritma genetika dapat dikatakan lebih baik daripada model regresi linier berganda dalam melakukan prediksi khususnya memprediksi curah hujan dengan data *time series*.

BAB 7 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap skripsi. Kesimpulan dan saran disajikan secara terpisah, dengan penjelasan sebagai berikut:

7.1 Kesimpulan

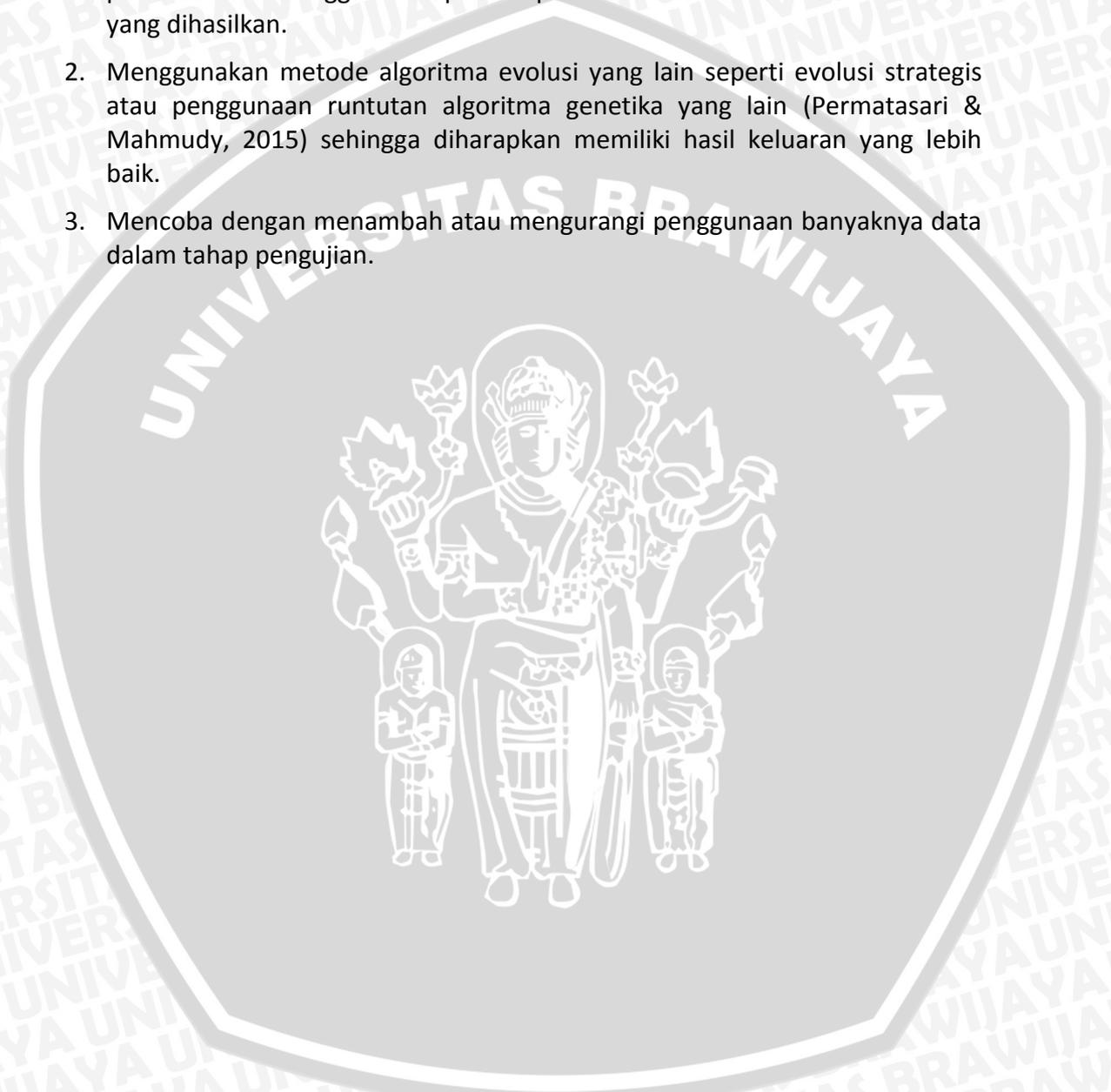
Berdasarkan hasil penelitian dalam membangun model ARIMA dengan algoritma genetika untuk memprediksi data *time series* curah hujan dapat disimpulkan seperti berikut:

1. Dalam membangun model ARIMA dengan algoritma genetika dapat dilakukan dengan penggunaan *binary* kromosom yang merepresentasikan orde model AR (p) dan MA (q), dimana nilai biner 1 menandakan orde model signifikan dan nilai biner 0 menandakan orde model tidak signifikan. Penggunaan algoritma genetika dalam menentukan orde AR (p) dan MA (q) mampu meningkatkan hasil akurasi serta komputasi dalam melakukan prediksi data curah hujan, hal ini juga dialami oleh penelitian yang dilakukan oleh Ong, et al., (2005).
2. Dalam menentukan model ARIMA yang dibentuk dengan algoritma genetika terbaik adalah dengan melihat nilai *fitness* yang dihasilkan. Semakin besar nilai *fitness* maka model tersebut akan menghasilkan tingkat akurasi yang semakin baik pula. Sedangkan begitu juga sebaliknya apabila nilai *fitness* yang dihasilkan rendah, maka tingkat akurasi prediksi model tersebut juga akan rendah.
3. Penentuan nilai parameter algoritma sangatlah penting dalam membangun model ARIMA sehingga didapatkan model terbaik. Sehingga melalui hasil pengujian didapatkan nilai parameter algoritma terbaik guna membangun model ARIMA adalah ukuran *popSize* sebesar 120, ukuran generasi 500, serta kombinasi *cr* dan *mr* 0.5 : 0.5.
4. Berdasarkan hasil perbandingan model ARIMA yang dibentuk dengan algoritma genetika terhadap model regresi linier berganda memang menunjukkan nilai MAPE serta *fitness* paling baik adalah model regresi linier berganda dengan nilai MAPE 0.000013 sedangkan nilai MAPE hasil algoritma genetika adalah 1.462213. Namun grafik pada Gambar 6.5 menunjukkan hasil akurasi prediksi keduanya sama baik, dengan catatan model ARIMA hanya menggunakan 6 periode data dari 12 periode data sedangkan model regresi linier berganda menggunakan 12 periode data. Sehingga dalam hal ini model ARIMA yang dibentuk dengan algoritma genetika lebih baik dibanding model regresi linier berganda.

7.2 Saran

Berdasarkan kesimpulan diatas, maka untuk penelitian selanjutnya perlu diperhatikan beberapa saran berikut agar mendapatkan hasil yang lebih baik, diantaranya adalah:

1. Penggunaan metode *crossover*, mutasi, serta seleksi yang berbeda daripada penelitian ini sehingga diharapkan dapat menambah variasi hasil kromosom yang dihasilkan.
2. Menggunakan metode algoritma evolusi yang lain seperti evolusi strategis atau penggunaan runtutan algoritma genetika yang lain (Permatasari & Mahmudy, 2015) sehingga diharapkan memiliki hasil keluaran yang lebih baik.
3. Mencoba dengan menambah atau mengurangi penggunaan banyaknya data dalam tahap pengujian.



DAFTAR PUSTAKA

- Brockwell, P. J. & Davis, R. A., 2002. *Introduction to Time Series and Forecasting*. 2nd ed. New York: Springer-Verlag.
- Chen, K. Y. & Wang, C. H., 2007. Support Vector Regression with Genetic Algorithms in Forecasting Tourism Demand. *Tourism Management*, Volume 28, pp. 215-226.
- Crone, S. F., Lessmann, S. & Pietsch, S., 2006. Forecasting with Computational Intelligence - An Evaluation of Support Vector Regression and Artificial Neural Networks for Time Series Prediction. *International Joint Conference on Neural Networks*, pp. 16-21.
- Deitel, P. & Deitel, H., 2015. *Java How to Program*. 10th ed. United State of America: Pearson Education.
- Estiningtyas, W., Ramadhani, W. & Aldrian, E., 2007. Analisis Korelasi Curah Hujan dan Suhu Permukaan Laut Wilayah Indonesia, serta Implikasinya untuk Prakiraan Curah Hujan (Studi Kasus Kabupaten Cilacap). *J. Agromet Indonesia*, 21(2), pp. 46-60.
- Indrabayu, Harun, N., Pallu, M. S. & Achmad, A., 2011. Prediksi Curah Hujan di Wilayah Makassar Menggunakan Metode Wavelet-Neural Network. *Jurnal Ilmiah "Elektrikal Enjiniring" UNHAS*, 09(2), pp. 50-59.
- Iriany, A., Mahmudy, W. F. & Handoyo, S., 2015. *GSTAR-SUR Model for Rainfall Forecasting in Tengger Region, East Java*. Malang, UMM International Conference on Pure and Applied Research.
- Maarof, M. Z., Ismail, Z. & Fadzli, M., 2014. Optimization of SARIMA model using Genetic Algorithm Method in Forecasting Singapore Tourist Arrivals to Malaysia. *Applied Mathematical Sciences*, 8(170), pp. 8481-8491.
- Mahmudy, W. F., 2013. *Algoritma Evolusi. Modul Kuliah Semester Ganjil 2013-2014*. Malang: Universitas Brawijaya.
- Mauludiyanto, A., Hendrantoro, G., P., M. H. & Suhartono, 2009. Pemodelan ARIMA dan Deteksi Outlier Data Curah Hujan Sebagai Evaluasi Sistem Radio Gelombang Milimeter. *JUTI*, 7(3), pp. 109-114.
- Mirawati, T. D., Yasin, H. & Rusgiyono, A., 2013. Prediksi Curah Hujan dengan Metode Kalman Filter (Studi Kasus di Kota Semarang Tahun 2012). *JURNAL GAUSSIAN*, 2(3), pp. 239-248.
- Ong, C. S., Huang, J. J. & Tzeng, G. H., 2005. Model Identification of ARIMA Family using Genetic Algorithms. *Applied Mathematics and Computation*, Volume 164, pp. 885-912.
- 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(14).

- Rahmi, A., Mahmudy, W. F. & Setiawan, B. D., 2015. Prediksi Harga Saham Berdasarkan Data Historis Menggunakan Model Regresi yang dibangun dengan Algoritma Genetika. *DORO: Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*, 5(12).
- Sarpong, S. A., 2013. Modeling and Forecasting Maternal Mortality; an Application of ARIMA Models. *International Journal of Applied Science and Technology*, 3(1), pp. 19-28.
- Singh, S., Bhambri, P. & Gill, J., 2011. Time Series based Temperature Prediction using Back Propagation with Genetic Algorithm Technique. *International Journal of Computer Science (IJCSI)*, 8(5), pp. 28-32.
- Sivanandam, S. N. & Deepa, S. N., 2008. *Introduction to Genetic Algorithms*. 1 ed. New York: Springer-Verlag.
- Taboada, G. L. et al., 2013. Java in High Performance Computing Arena: Research, Practie, and Experience. *Science of Computer Programming*, 78(5), pp. 425-444.
- User, S., 2015. <http://www.staklimnegara.net>. [Online] Available at: <http://www.staklimnegara.net/index.php/buletin-hujan/pengertian-istilah> [Accessed 13 September 2015].
- Wiyanti, D. T. & Pulungan, R., 2012. Peramalan Deret Waktu Menggunakan Model Fungsi Basis Radial (RBF) dan Auto Regressive Integrated Moving Average (ARIMA). *Jurnal MIPA*, 35(2), pp. 175-182.

LAMPIRAN A DATA DASARIAN CURAH HUJAN WILAYAH PUSPO, TENGGER (JANUARI 2014 – DESEMBER 2014)

No.	Bulan	Dasarian	Data Curah Hujan
1.	Januari	1	23.800
2.		2	3.300
3.		3	10.273
4.	Februari	1	14.000
5.		2	6.000
6.		3	10.375
7.	Maret	1	6.500
8.		2	15.600
9.		3	3.273
10.	April	1	6.300
11.		2	9.300
12.		3	14.700
13.	Mei	1	2.600
14.		2	13.600
15.		3	12.909
16.	Juni	1	0.000
17.		2	3.500
18.		3	0.900
19.	Juli	1	0.200
20.		2	0.000
21.		3	2.091
22.	Agustus	1	0.000
23.		2	0.000
24.		3	0.000
25.	September	1	0.000
26.		2	0.000
27.		3	0.000

28.		1	0.000
29.	Oktober	2	0.000
30.		3	0.000
31.		1	0.400
32.	November	2	3.400
33.		3	9.500
34.		1	4.500
35.	Desember	2	22.900
36.		3	15.000

