



INSTITUT TEKNOLOGI DEL

**Perbandingan Prediksi Harga Saham dengan
Penerapan Algoritma Genetika dan Algoritma
Genetika Adaptif**

TUGAS AKHIR

11317050	Rizky Martin Sianturi
11317051	Mutiara Magdalena Simamora
11317062	Bernika Arni Siahaan

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
PROGRAM STUDI DIPLOMA 3 TEKNOLOGI
INFORMASI
LAGUBOTI
AGUSTUS 2020**



INSTITUT TEKNOLOGI DEL

Perbandingan Prediksi Harga Saham dengan Penerapan Algoritma Genetika dan Algoritma Genetika Adaptif

TUGAS AKHIR

**Diajukan sebagai salah Satu Syarat untuk Memperoleh
Gelara Diploma 3 Program Studi Teknologi Informasi**

11317050	Rizky Martin Sianturi
11317051	Mutiara Magdalena Simamora
11317062	Bernika Arni Siahaan

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
PROGRAM STUDI DIPLOMA 3 TEKNOLOGI
INFORMASI
LAGUBOTI
AGUSTUS 2020**

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Rizky Martin Sianturi

NIM : 11317050

Tanda Tangan :



Tanggal : 03 Agustus 2020

Nama : Mutiara Magdalena Simamora

NIM : 11317051

Tanda Tangan :



Tanggal : 03 Agustus 2020

Nama : Bernika Arni Siahaan

NIM : 11317062

Tanda Tangan :



Tanggal : 03 Agustus 2020

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

1. Nama : Rizky Martin Sianturi
NIM : 11317050
Program studi : D3 Teknologi Informasi
2. Nama : Mutiara Magdalena Simamora
NIM : 11317051
Program studi : D3 Teknologi Informasi
3. Nama : Bernika Arni Siahaan
NIM : 11317062
Program studi : D3 Teknologi Informasi
Judul Tugas Akhir : Perbandingan Prediksi Harga Saham dengan Penerapan Algoritma Genetika Adaptif.

Telah berhasil dipertahankan dihadapannya dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Diploma 3, pada program studi Diploma 3 Teknologi Informasi Fakultas Informatika dan Teknik Elektro Institut Teknologi Del.

DEWAN PENGUJI

- Pembimbing : Ike Fitriyaningsih, S.Si., M.Si ()
- Pembimbing : Tiurma Lumban Gaol, S.P., M.P. ()
- Penguji : Iustisia Natalia Simbolon, S.Kom., M.T. ()
- Penguji : Teamsar Muliadi Panggabean, S.Kom., PGCert. ()
- Ditetapkan : Laguboti
- Tanggal : 03 Agustus 2020

KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yesus Yang Maha Esa atas berkat dan rahmat-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini sesuai dengan waktu yang direncanakan.

Tugas Akhir ini ditulis sebagai bagian dari syarat kelulusan Program Studi Diploma 3 Teknologi Informasi di Institut Teknologi Del. Judul Tugas Akhir ini adalah ***“Perbandingan Prediksi Harga Saham dengan Penerapan Algoritma Genetika dan Algoritma Genetika Adaptif”***.

Dalam penyusunan dan penulisan Tugas Akhir ini tidak terlepas dari bantuan, bimbingan serta dukungan dari berbagai pihak sehingga dalam kesempatan ini penulis menyampaikan terima kasih kepada:

1. Kedua orangtua dan keluarga dari ketiga penulis dalam Tugas Akhir ini, atas dukungan dalam doa, moril dan materil kepada penulis dalam proses pengerjaan Tugas Akhir ini.
2. Ibu Ike Fitriyaningsih, S.Si., M.Si selaku dosen pembimbing pertama yang selalu bersedia memberikan bimbingan dan mengarahkan Penulis selama pengerjaan Tugas Akhir ini sehingga pengerjaan dan penulisan Tugas Akhir ini berjalan dengan baik.
3. Ibu Tiurma Lumban Gaol, S.P., M.P. selaku dosen pembimbing kedua yang selalu bersedia memberikan bimbingan dan mengarahkan Penulis selama pengerjaan Tugas Akhir ini sehingga pengerjaan dan penulisan Tugas Akhir ini berjalan dengan baik.
4. Ibu Iustisia Natalia Simbolon, S.Kom., M.T. dan Bapak Teamsar Muliadi Panggabean, S.Kom., PGCert selaku dosen penguji Tugas Akhir yang telah memberikan banyak saran dan kritikan yang membangun kepada penulis demi keberhasilan dari pengerjaan Tugas Akhir.
5. Kepada teman-teman seperjuangan pada Prodi D 3 Teknologi Informasi angkatan 2017 yang selalu mendukung penulis dalam pengerjaan Tugas Akhir ini.
6. Kepada sahabat dan seluruh pihak yang tidak dapat disebutkan satu persatu yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

Dalam penyusunan Tugas Akhir ini, penulis menyadari banyak kekurangan yang terdapat dalam penulisan Tugas Akhir ini. Oleh karena itu, penulis memohon maaf atas segala

kekurangan tersebut dan menerima saran dan kritik yang membangun untuk membantu pengembangan Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi pembaca.

Sitoluama, 03 Agustus 2020

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Institut Teknologi Del, saya yang bertanda tangan di bawah ini:

1. Nama : Rizky Martin Sianturi
NIM : 11317050
Program Studi : D3 Teknologi Informasi
2. Nama : Mutiara Magdalena Simamora
NIM : 11317051
Program studi : D3 Teknologi Informasi
3. Nama : Bernika Arni Siahaan
NIM : 11317062
Program studi : D3 Teknologi Informasi
Judul Tugas Akhir : Perbandingan Prediksi Harga Saham dengan Penerapan Algoritma Genetika dan Algoritma Genetika Adaptif
Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Del Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*) atas Tugas Akhir yang berjudul: ***Perbandingan Prediksi Harga Saham dengan Penerapan Algoritma Genetika dan Algoritma Genetika Adaptif***.

Dengan Hak Bebas Royalty Noneksklusif ini Institut Teknologi Del berhak menyimpan, mengalih/media-format dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama kami sebagai penulis/pencipta dan sebagai pemilik hak cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Laguboti
Pada tanggal : 03 Agustus 2020

Yang menyatakan



(Rizky Martin Sianturi)



(Mutiara Magdalena Simamora)



(Bernika Arni Siahaan)

ABSTRAK

Pasar saham selalu menjadi salah satu investasi yang menguntungkan karena keuntungannya yang tinggi. Setiap investor berusaha bagaimana agar keuntungan yang diperoleh dari saham semakin meningkat sambil meminimalkan risiko yang ada. Faktanya harga saham selalu berfluktuasi atau selalu berubah. Oleh karena itu, prediksi harga saham merupakan hal yang sangat penting bagi setiap orang yang berinvestasi maupun perusahaan yang berinvestasi dalam memutuskan membeli atau menjual saham. Dalam menyikapi hal tersebut, Penulis melakukan prediksi harga saham untuk lima hari kedepan disesuaikan dengan penelitian sebelumnya yang dilakukan oleh Asyrofa-Rahmi bahwa hasil uji coba banyaknya periode saham harian memiliki fitness rata – rata terbaik pada periode 5 hari, dengan mengimplementasinya menggunakan algoritma genetika dan algoritma genetika adaptif. Penggunaan kedua algoritma digunakan untuk melihat prediksi harga saham yang lebih baik diantara kedua algoritma berdasarkan nilai RMSE nya. Algoritma yang mempunyai RMSE lebih kecil dikatakan lebih akurat daripada algoritma yang mempunyai RMSE lebih besar. Penggunaan algoritma genetika adaptif untuk prediksi harga saham dimodifikasi dengan menggunakan bantuan algoritma *Particle Swarm Optimization* (PSO) yang ditambahkan pada proses *crossover* yang bertujuan untuk menghasilkan *parent* yang lebih baik. Proses prediksi data saham menggunakan data harga saham *close* perusahaan Gudang Garam Tbk untuk kurun waktu tiga tahun, terhitung dari 1 Januari 2016 hingga 1 Januari 2019, dengan jumlah dataset yang digunakan sebanyak 762 baris data. Hasil dari Tugas Akhir ini mendapatkan nilai RMSE algoritma genetika cenderung lebih besar dibandingkan algoritma genetika adaptif. Oleh karena nya untuk lebih meningkatkan efektivitas kedua algoritma disarankan untuk menggunakan lebih dari satu atribut selain atribut *close* pada prediksi data saham untuk penelitian selanjutnya dan juga dapat menggunakan algoritma lain untuk membantu pemodifikasian untuk algoritma genetika adaptif.

Kata Kunci : Algoritma Genetika, Algoritma Genetika Adaptif, PSO, RMSE, Saham

ABSTRACT

The stock market has always been a profitable investment because of its high profits. Every investor is trying to how to increase profits from shares while minimizing existing risks. The fact is stock prices always fluctuate or always change. Therefore, stock price prediction is very important for everyone who invests and the company that invests in deciding to buy or sell shares. In response to this, the authors make stock price predictions for the next five days adjusted to previous research conducted by Asyrofa-Rahmi that the results of trials of many daily stock periods have the best average fitness for a 5-day period, by implementing them using genetic algorithms and algorithms adaptive genetics. The use of both algorithms is used to see a better stock price prediction between the two algorithms based on its RMSE value. Algorithms that have smaller RMSE are said to be more accurate than algorithms that have larger RMSE. The use of adaptive genetic algorithms for stock price predictions is modified using the help of the Particle Swarm Optimization (PSO) algorithm which is added to the crossover process which aims to produce a better parent. The stock data prediction process uses the company's close stock price data of Gudang Garam Tbk for a period of three years, starting from January 1, 2016 to January 1, 2019, with a total of 762 data lines used. The results of this Final Project get the value of RMSE genetic algorithm tends to be greater than the adaptive genetic algorithm. Therefore to further improve the effectiveness of the two algorithms it is recommended to use more than one attribute besides the close attribute in the prediction of stock data for further research and can also use other algorithms to help modify the adaptive genetic algorithm.

Keywords: *Adaptive Genetic Algorithm, Genetic Algorithm, PSO, RMSE, Stock*

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	v
ABSTRAK	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
DAFTAR CODE	xiv
Bab 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Lingkup	2
1.5 Pendekatan	3
1.6 Sistematika Penyajian	3
Bab 2 TINJAUAN PUSTAKA	5
2.1 Saham	5
2.2 Yahoo Finance	6
2.3 Prediksi.....	8
2.4 Regresi	8
2.5 RMSE	9
2.6 Algoritma Genetika	10
2.6.1 Membangkitkan Populasi Awal	11
2.6.2 Menghitung Nilai <i>Fitness</i>	12
2.6.3 Seleksi	12
2.6.4 <i>Crossover</i>	13
2.6.5 Mutasi.....	15
2.7 Algoritma Genetika Adaptif	16
2.8 Particle Swarm Optimization (PSO)	16
2.9 Penelitian Sejenis.....	17
Bab 3 ANALISIS DAN DESAIN	22
3.1 Analisis Data	22
3.1.1 Dataset yang Digunakan	22
3.2 Analisis Algoritma	22
3.2.1 Membangkitkan Populasi Awal	24
3.2.2 Evaluasi Nilai <i>Fitness</i>	28
3.2.3 Selection	30
3.2.4 <i>Crossover</i>	31
3.2.5 <i>Mutation</i>	35
3.3 Product Main Function	35
3.4 Use case Pada Simulator	36
3.5 Use case Scenario	37
3.5.1 Use case Scenario FR-01	37
3.5.2 Use case Scenario FR-02	38
3.5.3 Use case Scenario FR-03	39
3.6 Desain Antarmuka Simulator	40
Bab 4 IMPLEMENTASI DAN TESTING.....	42
4.1 Implementasi	42
4.1.1 Lingkungan Implementasi	42
4.1.2 Implementasi Antarmuka.....	43
4.1.3 Tahapan Implementasi Menampilkan Hasil	55
4.1.4 Tahapan Implementasi Menyimpan Hasil	57
4.2 Pengujian	58
4.2.1 Pengujian AG.....	58
4.2.2 Pengujian AGA	60
Bab 5 HASIL DAN PEMBAHASAN	64
5.1 Hasil Implementasi Memuat Data	64
5.2 Hasil Melakukan Forecasting Data.....	66

5.2.1	Implementasi Prediksi Harga Saham dengan Algoritma Genetika	66
5.2.2	Implementasi Prediksi Harga Saham dengan Algoritma Genetika Adaptif	67
5.3	Menyimpan Hasil <i>Forecasting</i>	68
Bab 6	KESIMPULAN DAN SARAN	69
6.1	Kesimpulan.....	69
6.2	Saran.....	69
	Daftar Pustaka dan Rujukan	70

DAFTAR TABEL

Tabel 2.1 Penelitian Sejenis	18
Tabel 3.1 Data Saham dengan 5 Periode	25
Tabel 3.2 Contoh Representasi Kromosom / Data Saham	26
Tabel 3.3 Data Saham Menurut Kromosom Tabel 3.4	27
Tabel 3.4 Pembentukan Populasi Awal	27
Tabel 3.5 Nilai Koefisien Regresi	28
Tabel 3.6 Proses Perhitungan Nilai RMSE	29
Tabel 3.7 Contoh 100 Populasi dengan Nilai <i>Fitness</i>	30
Tabel 3.8 Contoh <i>Crossover</i> Populasi	31
Tabel 3.9 Contoh Hasil <i>Crossover</i> Populasi.....	32
Tabel 3.10 Contoh Individu yang Terpilih untuk Dimodifikasi	33
Tabel 3.11 Inisialisasi Posisi, Kecepatan dan Nilai <i>Fitness</i> Individu.....	34
Tabel 3.12 Contoh Mutasi Populasi.....	35
Tabel 3.13 Fungsi Utama Simulator dan Deskripsi Fungsi.....	36
Tabel 3.14 <i>Use case Scenario</i> FR-01	37
Tabel 3.15 <i>Use case Scenario</i> FR-02	38
Tabel 3.16 <i>Use case Scenario</i> FR-03	39
Tabel 4.1 Spesifikasi Perangkat Keras	42
Tabel 4.2 Spesifikasi Perangkat Lunak	43
Tabel 4.3 Prediksi Harga Saham Percobaan I AG.....	59
Tabel 4.4 Prediksi Harga Saham Percobaan II AG	59
Tabel 4.5 Prediksi Harga Saham Percobaan III AG	60
Tabel 4.6 Prediksi Harga Saham Percobaan I AGA	61
Tabel 4.7 Prediksi Harga Saham Percobaan II AGA	61
Tabel 4.8 Prediksi Harga Saham Percobaan III AGA	62
Tabel 4.9 Nilai RMSE AG dan AGA	62
Tabel 5.1 Prediksi Harga Saham dengan AG	67
Tabel 5.2 Prediksi Harga Saham dengan AGA	68

DAFTAR GAMBAR

Gambar 2.1 Tampilan Saham PT Gudang Garam Tbk	7
Gambar 2.2 <i>Flowchart</i> Algoritma Genetika	11
Gambar 2.3 Visualisasi Teknik 1-Point <i>Crossover</i>	13
Gambar 2.4 Visualisasi Teknik K-Point <i>Crossover</i>	14
Gambar 2.5 Visualisasi Teknik Uniform <i>Crossover</i>	15
Gambar 3.1 Flowchart Desain Simulator Algoritma Genetika	23
Gambar 3.2 Flowchart Desain Simulator Algoritma Genetika Adaptif	24
Gambar 3.3 Plot Data Harga Close Saham	25
Gambar 3.4 <i>Use case</i> Diagram Simulator	36
Gambar 3.5 Desain Antarmuka Simulator	40
Gambar 4.1 Implementasi Tampilan Utama Antarmuka Simulator	43
Gambar 4.2 <i>Window</i> Select File Data Aktual Saham	45
Gambar 4.3 Implementasi Plot Data Aktual Saham	46
Gambar 4.4 Tampilan Hasil Prediksi Harga Saham dengan Algoritma Genetika	57
Gambar 4.5 Percobaan I AG	59
Gambar 4.6 Percobaan II AG	59
Gambar 4.7 Percobaan III AG	60
Gambar 4.8 Percobaan I AGA	61
Gambar 4.9 Percobaan II AGA	61
Gambar 4.10 Percobaan III AGA	62
Gambar 5.1 Tampilan Awal simulator	64
Gambar 5.2 <i>Window</i> Select File Data Aktual Saham	65
Gambar 5.3 Plot Data Aktual Saham	66
Gambar 5.4 Grafik Hasil Prediksi Harga Saham dengan AG	66
Gambar 5.5 Nilai RMSE AG	67
Gambar 5.6 Grafik Hasil Prediksi Harga Saham dengan AGA	68
Gambar 5.7 Nilai RMSE AGA	68

DAFTAR CODE

Code 4.1 Kode Program Muat Data.....	44
Code 4.2 Kode Program Plot Data Aktual Saham	46
Code 4.3 Kode Program Membangkitkan Populasi Awal	47
Code 4.4 Kode Program Pembentukan Kromosom dan Koesfisien	47
Code 4.5 Kode Program Evaluasi Nilai <i>Fitness</i> Setiap Populasi	48
Code 4.6 Kode Program Proses Seleksi	50
Code 4.7 Kode Program Proses <i>Crossover</i>	51
Code 4.8 Kode Program Proses Mutasi	51
Code 4.9 Kode Program Proses Regenerasi	52
Code 4.10 Kode Program Proses Inisialisasi Posisi dan Velocity Particle	53
Code 4.11 Kode Program Evaluasi Nilai <i>Fitness</i> Partikel	54
Code 4.12 Kode Program Proses <i>Update</i> Global dan Local Best Position	54
Code 4.13 Kode Program Proses <i>Update</i> Position dan Velocity Partikel	55
Code 4.14 Kode Program Menampilkan Grafik Prediksi Harga Saham	56
Code 4.15 Kode Program Menampilkan Hasil Prediksi (Angka)	57
Code 4.16 Kode Program Menyimpan Hasil Prediksi Harga Saham	58

Bab 1

PENDAHULUAN

Bab pendahuluan berisi deskripsi umum dari kajian yang dikerjakan. Deskripsi ini mencakup latar belakang mengapa topik ini dipilih, tujuan dari kajian yang dilakukan, dan sistematika penyajian laporan hasil kajian yang dilakukan.

1.1 Latar Belakang

Pasar saham selalu menjadi salah satu investasi yang menguntungkan karena keuntungannya yang tinggi. Saham merupakan satuan nilai atau pembukuan dalam berbagai instrumen finansial yang mengacu pada bagian kepemilikan sebuah perusahaan. Dalam pengertian lain saham merupakan tanda penyertaan modal seseorang atau pihak (badan usaha) dalam suatu perusahaan atau perseroan terbatas. Setiap investor berusaha bagaimana agar keuntungan yang diperoleh dari saham semakin meningkat sambil meminimalkan risiko yang ada [1].

Faktanya harga saham selalu berfluktuasi atau selalu berubah, sehingga dalam dunia pasar saham, para *investor* harus cermat dalam pengambilan suatu keputusan kapan suatu saham sebaiknya dijual dan kapan suatu saham sebaiknya dipertahankan. Harga suatu saham yang terjadi di pasar bursa pada saat tertentu ditentukan oleh pelaku pasar, permintaan dan penawaran saham yang bersangkutan di pasar modal [2]. Sehingga, prediksi harga saham merupakan hal yang sangat penting bagi setiap orang yang berinvestasi maupun perusahaan yang berinvestasi dalam memutuskan membeli atau menjual saham.

Penelitian mengenai prediksi harga saham telah dilakukan sebelumnya oleh Sonal, Ankita dan Upendra [3]. Dalam penelitian tersebut mereka menggunakan Algoritma Genetika dan strategi evolusioner untuk membantu dalam memprediksi harga saham dari delapan perusahaan yang ada di India dengan akurasi prediksi menghasilkan bahwa Algoritma Genetika memiliki akurasi yang lebih tinggi yaitu 73,87% dibandingkan dengan strategi evolusioner 71,77%. Penelitian oleh Sonal, Ankita dan Upendra menganjurkan Algoritma genetika ini juga dapat dibandingkan dengan algoritma populer lainnya untuk melihat apakah algoritma genetika memiliki akurasi yang lebih tinggi dibandingkan dengan algoritma lainnya dalam memprediksi harga saham [3].

Pada penelitian ini, penulis akan membandingkan algoritma genetika dengan algoritma genetika adaptif untuk memprediksi harga saham. Algoritma Genetika Adaptif ialah algoritma genetika yang telah mengalami modifikasi pada operator *crossover* dengan menggunakan algoritma PSO, pemilihan algoritma ini diharapkan mendapatkan hasil

prediksi harga saham yang lebih baik dari algoritma genetika. Modifikasi menggunakan algoritma PSO dikarenakan PSO dan algoritma genetika memiliki beberapa kesamaan, salah satunya ialah kedua algoritma ini memiliki inisial populasi, menggunakan iterasi untuk mencari solusi yang optimal dan menggunakan *fitness value* untuk mengevaluasi kinerja calon solusi.

Penulis memilih metode *crossover* untuk dimodifikasi dengan PSO, karena dapat menghasilkan keturunan yang cenderung menjadi kromosom dengan posisi terbaik dan meningkatkan *convergence speed* [4]. Berdasarkan hal tersebut maka modifikasi operator *crossover* pada algoritma genetika dengan menggunakan PSO dapat dilakukan. Penelitian ini juga akan menggunakan data saham perusahaan Gudang Garam Tbk. Data saham perusahaan Gudang Garam Tbk dipilih karena perusahaan tersebut merupakan perusahaan yang memiliki saham termahal di Indonesia ditahun 2018 dengan harga Rp.54.000/unit [5]. Sehingga dari tugas akhir ini dapat diketahui hasil prediksi yang lebih baik antara algoritma genetika dengan algoritma genetika adaptif yang dilihat berdasarkan tingkat keakurasiannya.

1.2 Rumusan Masalah

Rumusan masalah yang menjadi acuan pengerjaan tugas akhir ini adalah, apakah hasil prediksi harga saham dengan menggunakan Algoritma Genetika Adaptif lebih baik dibandingkan Algoritma Genetika dilihat dari nilai RMSE nya?

1.3 Tujuan

Adapun tujuan pelaksanaan tugas akhir ini adalah mendapatkan hasil prediksi harga saham yang terbaik antara Algoritma Genetika dengan Algoritma Genetika Adaptif.

1.4 Lingkup

Lingkup pembahasan dalam kajian ini bertujuan agar masalah yang akan diteliti dapat lebih terarah dan terfokus sesuai dengan apa yang direncanakan yaitu:

1. Data yang digunakan dalam penelitian ini adalah data indeks saham gabungan Indonesia yang diperoleh dari *Yahoo! Finance* dengan kurun waktu tiga tahun terakhir (01 Januari 2016 – 01 Januari 2019) dengan jumlah dataset 762.
2. Melihat keakuratan prediksi harga saham dengan acuan fungsi *Root Mean Square Error* (RMSE)
3. Prediksi harga saham dilakukan untuk lima hari, dikarenakan merupakan jumlah hari aktif data saham per minggu di *Yahoo! Finance Web*

4. *Tools* yang digunakan adalah Matlab.

1.5 Pendekatan

Beberapa pendekatan yang akan digunakan dalam pelaksanaan Tugas Akhir ini meliputi:

1. Studi Literatur

Studi literatur adalah mencari informasi yang berkaitan dengan tugas akhir dan mengumpulkannya. Adapun informasi tersebut dapat diambil dari sumber tertulis seperti sumber buku [6], majalah ilmiah, jurnal karya ilmiah, maupun sumber bacaan dalam bentuk digital seperti e-jurnal yang dapat dijadikan sebagai acuan untuk melakukan prediksi harga saham menggunakan Algoritma Genetika Adaptif.

2. Sumber Data (Pengumpulan Data)

Adapun data yang digunakan untuk pengerjaan Tugas Akhir ini adalah data harga saham dari tahun 2016 sampai dengan tahun 2018. Semua data saham tersebut diperoleh dari website *Yahoo! Finance* dalam kurun waktu tiga tahun.

3. Analisis

Pada tahap ini dilakukan analisis data saham dan analisis algoritma yang akan didesain yaitu algoritma genetika dan algoritma genetika adaptif dengan modifikasi *crossover* yang akan diimplementasikan dalam pengerjaan prediksi harga saham.

4. Desain

Pada tahap ini dilakukan desain pada *user interface* simulator yang akan diimplementasikan.

5. Implementasi

Pada tahap ini, hasil rancangan yang telah dibuat diimplementasikan untuk membuat simulator prediksi harga saham menggunakan Algoritma Genetika dan Algoritma Genetika Adaptif.

6. Testing

Pada tahap ini hasil implementasi diuji dengan menggunakan testing jenis *Black Box Testing*. Dimana *Black Box Testing* berfokus pada pengujian fungsionalitas, tampilan simulator, kesesuaian alur fungsi dengan bisnis proses.

1.6 Sistematika Penyajian

Untuk mempermudah pemahaman dalam pembahasan tugas akhir ini, maka penulisan dokumen ini dibagi menjadi enam bab dan masing-masing diuraikan dengan sistematika sebagai berikut:

1. Bab I Pendahuluan

Pada bab ini, pendahuluan menjelaskan tentang latar belakang, tujuan, ruang lingkup, pendekatan yang digunakan dan sistematika penulisan dokumen Tugas Akhir.

2. Bab II Tinjauan Pustaka

Pada bab ini, tinjauan pustaka menjelaskan informasi baik referensi maupun dasar teori yang digunakan dalam pengerjaan penelitian Tugas Akhir ini.

3. Bab III Analisis dan Desain

Pada bab ini, data dan desain menjelaskan mengenai analisis data dan analisis algoritma yang akan diolah dalam penelitian Tugas Akhir serta rancangan tahapan penelitian yang akan dilakukan serta rancangan simulator yang akan dibangun.

4. Bab IV Implementasi dan Pengujian

Pada bab ini, implementasi menjelaskan bagaimana proses implementasi yang dilakukan dan proses pengujian serta rancangan berdasarkan data dan desain yang telah dibahas pada bab tiga.

5. Bab V Hasil dan Pembahasan

Pada bab ini, hasil dan pembahasan menjelaskan hasil dari implementasi dan eksperimen yang telah dilakukan beserta dengan pembahasan setiap hasil tersebut.

6. Bab VI Kesimpulan dan Saran

Pada bab ini, kesimpulan dan saran menjelaskan kesimpulan dari hasil dan pembahasan yang didapatkan dan saran yang diberikan untuk penelitian selanjutnya.

Bab 2

TINJAUAN PUSTAKA

Tinjauan Pustaka menjelaskan berbagai teori, metode, teknik, proses dan prosedur yang terkait dengan tema penelitian, dan kajian mengenai hasil-hasil penelitian sebelumnya.

2.1 Saham

Saham merupakan satuan nilai atau pembukuan dalam berbagai instrumen finansial yang mengacu pada bagian kepemilikan sebuah perusahaan [1]. Menerbitkan saham merupakan salah satu pilihan perusahaan ketika memutuskan untuk pendanaan perusahaan. Pada sisi yang lain, saham merupakan instrumen investasi yang banyak dipilih para *investor* karena saham mampu memberikan tingkat keuntungan yang menarik.

Saham merupakan investasi yang sangat menguntungkan yang diperoleh dari beberapa hal berikut:

1. *Dividen*

Dividen yaitu keuntungan yang dibagi secara merata oleh perusahaan kepada para pemegang saham, yang dibagi menjadi dua jenis yaitu, *dividen tunai* berupa uang tunai dalam jumlah rupiah tertentu untuk setiap saham dan *dividen saham* berupa pemberian sejumlah saham kepada pemilik modal sehingga jumlah saham yang dimilikinya bertambah.

2. *Capital Gain*

Capital Gain merupakan keuntungan atau laba yang diperoleh investor ketika nilai jual saham lebih besar dari harga belinya.

Investasi saham juga memiliki risiko, antara lain:

1. *Capital Loss*

Capital Loss merupakan kerugian yang diperoleh investor ketika nilai beli saham lebih besar daripada nilai jual.

2. Risiko Likuidasi

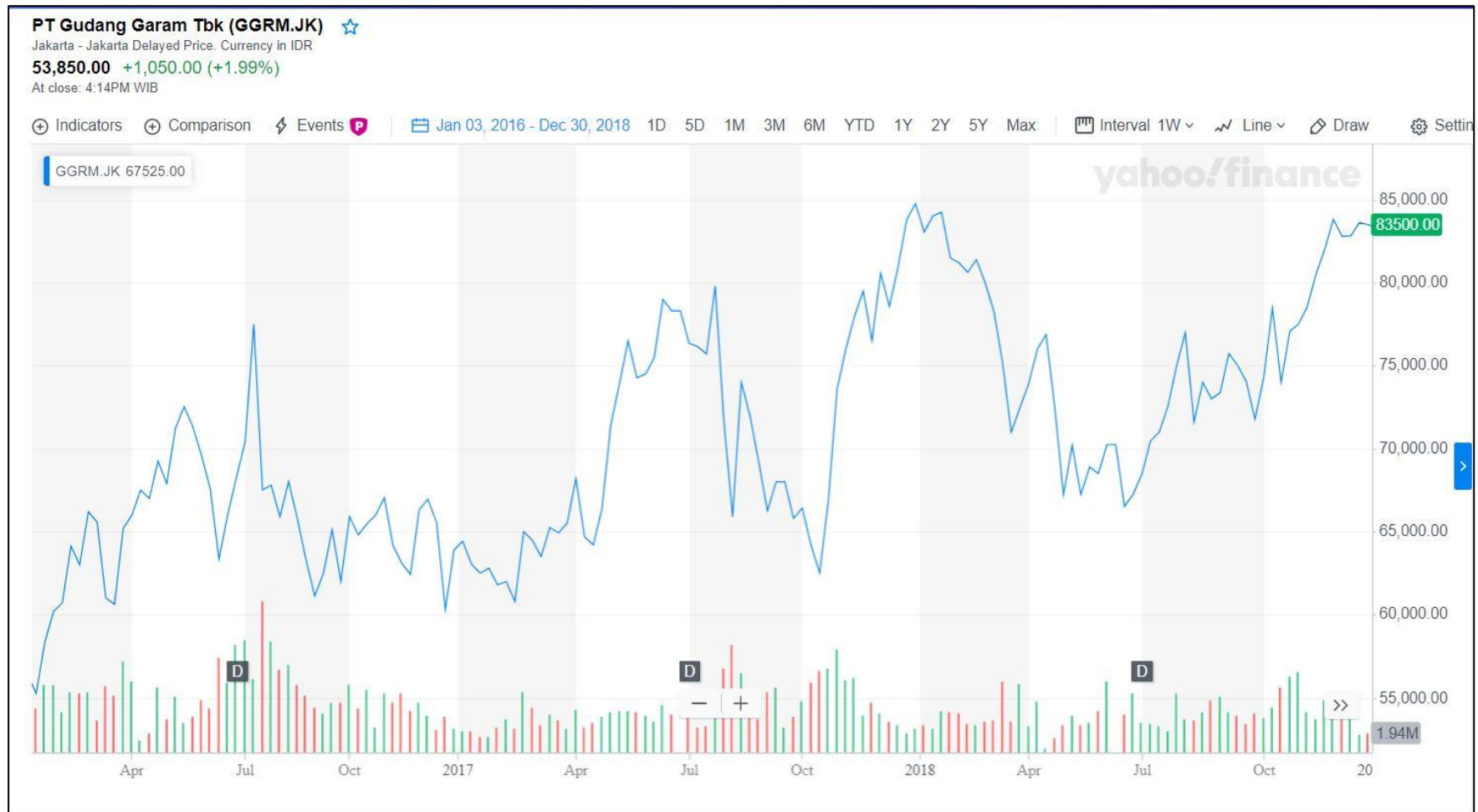
Risiko Likuidasi merupakan kerugian yang diperoleh *investor* karena saham dari suatu perusahaan yang dibeli dinyatakan bangkrut oleh pengadilan, atau perusahaan tersebut dibubarkan.

Menghindari risiko yang dapat terjadi, para *investor* harus cermat dalam pengambilan suatu keputusan kapan suatu saham sebaiknya dijual dan kapan suatu saham sebaiknya

dipertahankan. Bagi *investor* untuk mengetahui prediksi harga saham dapat membantu *investor* untuk mengambil sebuah keputusan. Memprediksi harga saham yaitu dengan menggunakan data lampau dari suatu saham sesuai dengan rentang waktu yang diinginkan, sehingga mendapatkan prediksi harga saham yang akan datang untuk mengambil suatu keputusan. Faktor yang mampu mempengaruhi harga saham yaitu berita ekonomi, laporan keuangan perusahaan, tingkat suku bunga, kondisi perekonomian suatu negara, kebijaksanaan pemerintah, laju inflasi, kepercayaan *investor*, dan lain sebagainya [2].

2.2 Yahoo Finance

Yahoo finance merupakan situs *web* yang menyajikan data historis harga saham dalam kurun waktu yang cukup panjang. Situs ini menyediakan informasi tentang data keuangan, informasi pasar saham, dan aplikasi lainnya yang membantu pengguna memperoleh informasi pasar saham dengan terperinci dan data terkini. *Yahoo Finance* memiliki misi untuk membantu pengguna membuat keputusan berdasarkan informasi mengenai investasi dan pasar *finance*. Data pada *yahoo finance* ditampilkan dalam bentuk tabel yang memuat tanggal, harga pembukuan, harga tertinggi, harga terendah, harga penutupan, dan *volume* [7]. Tampilan saham PT Gudang Garam Tbk dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tampilan Saham PT Gudang Garam Tbk

Sumber : *Yahoo Finance*

2.3 Prediksi

Prediksi adalah suatu proses memperkirakan atau menduga secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti terhadap kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi. Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengambilan keputusan. Prediksi harga saham berarti memperkirakan harga saham pada masa mendatang. Teknik yang dipakai untuk memprediksi harga saham yaitu dengan prediksi kuantitatif. Prediksi kuantitatif adalah prediksi yang didasarkan atas data kuantitatif pada masa lalu.

2.4 Regresi

Regresi merupakan metode analisis statistika untuk melihat pergerakan suatu data dari waktu ke waktu. Klasifikasi model regresi terbagi atas dua yaitu, regresi linear dan non-linear. Berikut ini adalah penjelasan klasifikasi regresi.

a. Regresi *Linear*

Model regresi *linear* digunakan untuk mengamati pengaruh variabel bebas atau variabel *independent* atau variabel *predictor* dengan terhadap variabel tergantung atau variabel *dependent* atau variabel terikat [8]. Persamaan model regresi *linear* dapat pada persamaan (2.1).

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_i X_i + \varepsilon \quad (2.1)$$

keterangan:

- β_i : parameter regresi (koefisien)
- Y : variabel *dependent* atau variabel terikat
- X_i : variabel *independent* atau variabel bebas
- ε : *random error*

b. Regresi Nonlinear

Model regresi *nonlinear* merupakan model regresi yang melihat hubungan antara variabel bebas atau variabel *predictor* dengan variabel terikat yang tidak bersifat linear. Persamaan model regresi *nonlinear* ditulis pada persamaan (2.2):

$$Y' = I + \alpha_1 X_1 + \alpha_2 X_2 + \dots + \alpha_n X_n + \alpha_{n+1} X_1^2 + \alpha_{n+2} X_2^2 + \dots + \alpha_{2n} X_n^2 \quad (2.2)$$

keterangan:

- Y' : data hasil regresi saham
 X_i : data harga saham pada i hari sebelumnya
 α_i : koefisien acak ke- i
 i : hari ke-1 sampai dengan hari ke- n

2.5 RMSE

Pada penelitian ini hasil prediksi saham akan dilihat dari besar kecilnya *Root Mean Square Error* (RMSE) yang dihasilkan dari prediksi. RMSE merupakan prinsip skor kuadrat (*quadratic score principle*) yang digunakan untuk menentukan besarnya rata-rata kesalahan estimasi pada tren pasar saham (*stock market*), yang dapat direpresentasikan secara matematis pada persamaan (2.3).

$$RMSE = \frac{1}{n} \sqrt{\sum_{i=1}^n (Y_i - Y'_i)^2} \quad (2.3)$$

keterangan:

- Y_i : data aktual harga saham
 Y'_i : data hasil regresi harga saham (dari persamaan (2.2))

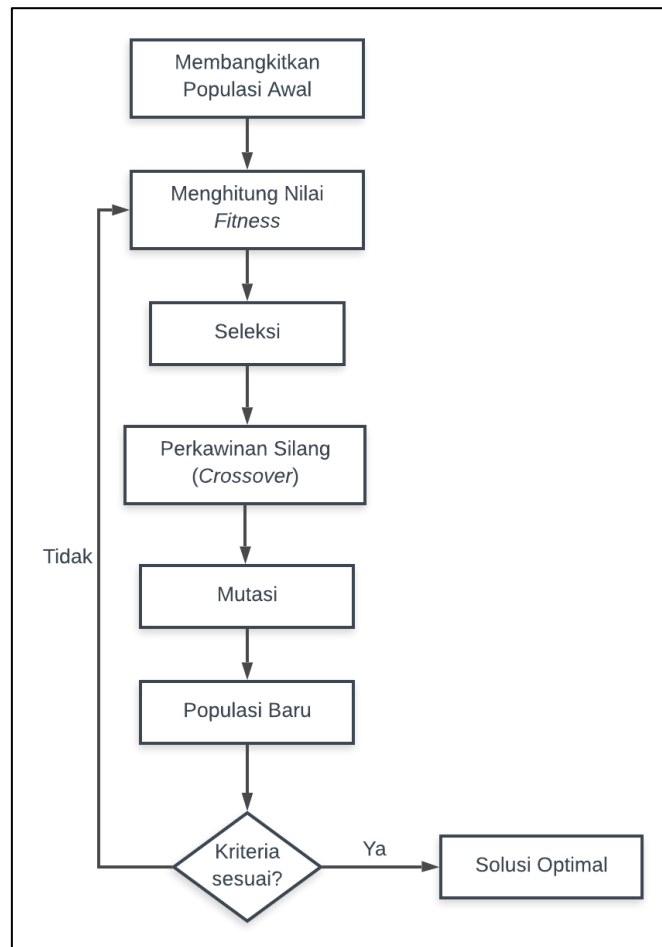
Menurut (Murtaza Roondiwala, 2015) kesalahan atau perbedaan antara target dan nilai *output* yang diperoleh dapat diminimalkan dengan menggunakan nilai RMSE. Penggunaan RMSE sangat baik untuk prediksi numerik. Jika dibandingkan dengan *Mean Absolute Error* (MAE), RMSE lebih baik dalam mengukur suatu kesalahan (*error*) dalam prediksi.

Keakuratan metode estimasi kesalahan pengukuran diindikasikan dengan adanya RMSE yang kecil. Metode estimasi yang mempunyai RMSE lebih kecil dikatakan lebih akurat daripada metode estimasi yang mempunyai RMSE lebih besar [9].

2.6 Algoritma Genetika

Algoritma Genetika (AG) merupakan metode *metaheuristic* yang terinspirasi dari proses seleksi natural evolusi biologi berdasarkan teori evolusi Charles Darwin [5]. Algoritma genetika adalah suatu tipe algoritma optimasi digunakan untuk mencari solusi optimal dari suatu permasalahan dengan memaksimalkan atau meminimumkan suatu fungsi terkait [10]. Dalam algoritma genetika, solusi kandidat dari permasalahan akan dikodekan menjadi array variabel dengan panjang tertentu. Setiap variabel tersebut disebut sebagai kromosom. Setiap solusi memiliki nilai kualitas yang disebut dengan nilai *fitness*. Nilai *fitness* dijadikan sebagai tolak ukur baik atau tidaknya solusi.

Setiap solusi akan diseleksi untuk mendapatkan solusi yang terbaik dengan membandingkan nilai *fitness*-nya. Setelah lolos dari tahapan seleksi, solusi tersebut akan dikawinkan dengan solusi lainnya untuk menghasilkan solusi baru, proses inilah yang disebut dengan *crossover*. Setelah solusi baru dihasilkan, solusi tersebut kemudian dimutasi untuk memodifikasi sebagian kecil bagian dari solusi tersebut. Proses ini akan berlangsung sampai banyak n iterasi yang telah ditentukan [11]. Adapun implementasi Algoritma Genetika menggunakan komponen dasar yang sama yaitu populasi, seleksi, *crossover*, dan operator mutasi. *Flowchart* yang menunjukkan tahapan dari proses algoritma genetika ditunjukkan pada Gambar 2.2. Penjelasan tahapan algoritma genetika akan dijelaskan pada subbab berikutnya.



Gambar 2.2 Flowchart Algoritma Genetika

Sumber : Columbia University, (2016)

2.6.1 Membangkitkan Populasi Awal

Membangkitkan populasi awal merupakan proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu [6]. Populasi yang dimaksud terdiri dari sekumpulan individu yang merupakan solusi terhadap permasalahan yang akan diselesaikan. Sebelumnya ukuran populasi harus ditentukan untuk membangkitkan populasi awal dengan cara inisialisasi solusi yang mungkin kedalam sejumlah kromosom dan dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran kinerjanya. Terdapat berbagai teknik dalam membangkitkan populasi awal yaitu:

1. Random Generator

Teknik random generator melibatkan pembangkitan bilangan random untuk setiap nilai gen sesuai dengan representasi kromosom yang digunakan.

2. Pendekatan Tertentu

Teknik pendekatan tertentu dilakukan dengan memasukkan nilai tertentu ke dalam gen dari populasi awal yang terbentuk.

3. Permutasi Gen

Cara permutasi gen dalam pembangkitan populasi awal adalah menggunakan permutasi Josephus dalam permasalahan kombinatorial seperti pada TSP.

2.6.2 Menghitung Nilai *Fitness*

Setelah proses pembangkitan populasi dilakukan, akan dihitung nilai *fitness* dari setiap individu. Nilai *fitness* adalah nilai yang menyatakan baik tidaknya suatu solusi (individu). Nilai *fitness* akan menunjukkan kualitas dari suatu kromosom dalam populasi tersebut. Rumus perhitungan nilai *fitness* ditentukan berdasarkan masalah yang ada.

2.6.3 Seleksi

Tahapan seleksi akan menerima sejumlah n individu yang mempunyai nilai *fitness* masing-masing. Pada proses seleksi, semua kandidat individu yang mempunyai nilai *fitness* akan dievaluasi dan dibandingkan dengan kandidat individu lainnya untuk memilih kandidat individu yang terbaik. Kandidat individu yang terbaik akan masuk ke proses selanjutnya berdasarkan *fitness value* individu tersebut. Terdapat beberapa metode seleksi pada algoritma genetika yaitu sebagai berikut:

1. *Rank Selection*

Metode *rank selection* diusulkan oleh Baker pada tahun 1985. Pemilihan kromosom terbaik dengan melakukan pengurutan individu berdasarkan nilai *fitness*-nya kemudian memberi nilai *fitness* baru dengan nilai 1 sampai dengan N . Dimana N merupakan jumlah kromosom di dalam populasi.

2. *Roulette Wheel*

Roulette wheel adalah metode yang paling sederhana yang dikenal dengan nama *Stochastic Sampling with Replacement*. Adapun langkah-langkah metode ini adalah sebagai berikut:

1. Hitung nilai *fitness* dari setiap individu (f_i), dimana i adalah individu ke-1 sampai dengan individu ke- n .
2. Hitung total nilai *fitness* semua individu.
3. Hitung probabilitas masing-masing individu.
4. Dari probabilitas tersebut, hitung jatah setiap individu pada angka 1 sampai 100.
5. Bangkitkan bilangan *random* antara 1 sampai 100.

6. Dari bilangan *random* yang diperoleh, tentukan individu mana yang terpilih dalam proses seleksi.

3. *Tournament Section*

Metode seleksi *tournament* adalah metode seleksi dengan menetapkan suatu nilai *tour* untuk individu-individu yang dipilih secara random dari suatu populasi. Parameter yang digunakan pada metode ini adalah ukuran *tour* yang bernilai antara 2 sampai N (jumlah individu dalam suatu populasi). Setiap kromosom akan dibandingkan nilai *fitness*-nya dengan nilai *fitness* kromosom lainnya untuk mendapatkan satu kromosom terkuat yang akan dijadikan sebagai solusi baru.

2.6.4 *Crossover*

Crossover adalah tahapan mempersilangkan dua kromosom untuk menghasilkan kromosom-kromosom baru yang memiliki karakter (gen) yang berasal dari induknya. Kromosom yang dihasilkan akan digunakan untuk menghasilkan individu baru untuk proses berikutnya. Tingkat penyilangan atau peluang penyilangan (dinotasikan sebagai P_c) adalah rasio antara jumlah kromosom yang diharapkan mengalami penyilangan dalam setiap generasi dengan kromosom total dalam populasi [12]. *Crossover* memiliki beberapa jenis teknik yang dapat diterapkan pada proses *crossover* yaitu:

a. *1-POINT Crossover*

1-Point crossover adalah salah satu teknik *crossover* sederhana yang digunakan untuk aplikasi AG secara acak. Teknik ini memilih dua *parent* yang digunakan untuk *crossover*, kemudian memilih fragmentasi/titik tunggal dari dua *parent* tersebut secara acak. Nantinya, hasil dari *crossover* di titik yang telah dipilih akan menghasilkan dua keturunan.

Parent 1:	1 0 1 0		1 0 0 1 0
Parent 2:	1 0 1 1		1 0 1 1 0
Offspring 1:	1 0 1 0		1 0 1 1 0
Offspring 2:	1 0 1 1		1 0 0 1 0

Gambar 2.3 Visualisasi Teknik *1-Point Crossover*

Pada Gambar 2.3 Visualisasi Teknik *1-Point Crossover*, titik antara posisi keempat dan kelima dipilih sebagai titik *crossover*.

b. *K-POINT CROSSOVER*

K-point crossover merupakan teknik *crossover* ini mirip dengan teknik *One-Point Crossover*. Namun, yang menjadikan kedua *crossover* ini berbeda ialah teknik *crossover* ini memilih lebih dari satu *point crossover* untuk menghasilkan keturunan.

Parent 1:	1 0	1 0	1 0 0	1 0
Parent 2:	1 1	0 0	1 0 1	1 0
Offspring 1:	1 0	0 0	1 0 0	1 0
Offspring 2:	1 1	1 0	1 0 1	1 0

Gambar 2.4 Visualisasi Teknik K-Point Crossover

Pada Gambar 2.4 Visualisasi Teknik *K-Point Crossover* terpilih *crossover point* yang berada diantara posisi kedua dan ketiga, keempat dan kelima, serta keenam dan ketujuh.

c. *REDUCED SURROGATE CROSSOVER*

Reduced Surrogate Crossover merupakan teknik *crossover* yang tidak diterapkan pada *parent* yang memiliki gen yang sama. Dalam kasus ini, *Reduced Surrogate Crossover* pertama-tama memeriksa gen individu pada *parent*. Teknik ini membuat daftar semua *crossover points* yang mungkin di mana gen yang dimiliki kedua *parent* berbeda.

Setelah melakukan pemeriksaan ini, jika tidak ada *crossover points*, maka tidak ada tindakan yang diambil. Tetapi jika ada kasus, dimana *parent* memiliki lebih dari 1 gen berbeda, maka ia menyimpan daftar semua *crossover points*. Kemudian secara acak memilih satu *crossover points* dari daftar dan melakukan *1-Point crossover* untuk membuat keturunan.

d. *UNIFORM CROSSOVER*

Pada Teknik *Uniform Crossover* dipilih dua *parent* untuk melakukan *crossover*. Hasil dari *crossover* ini ialah 2 *offspring* dari n gen yang dipilih secara seragam dari kedua *parent*. Bilangan *real* acak menentukan apakah *offspring* pertama akan memilih gen ke-i dari *parent* pertama atau *parent* kedua.

Parent 1:	1 1 1 0 1 0 0 1 0
Parent 2:	1 0 0 0 1 0 1 1 0
Offspring 1:	1 1 0 0 1 0 1 1 0
Offspring 2:	1 0 1 0 1 0 0 1 0

Gambar 2.5 Visualisasi Teknik Uniform Crossover

Pada Gambar 2.5 diketahui bahwa nilai alel pada gen berindeks ganjil di offspring 1 merupakan alel yang dipilih dari *parent* 2 dan nilai alel pada gen berindeks genap di offspring 1 merupakan alel yang dipilih dari *parent* 1. Hal ini berlaku juga untuk offspring 2.

2.6.5 Mutasi

Proses mutasi adalah suatu proses memodifikasi informasi gen secara acak pada suatu kromosom untuk menghasilkan individu yang berbeda dalam suatu populasi. Perubahan ini dapat membuat solusi duplikasi yang memiliki nilai *fitness* yang lebih rendah atau lebih tinggi dari solusi induknya [13]. Probabilitas mutasi atau peluang mutasi (P_m) merupakan rasio antara jumlah gen yang diharapkan mengalami mutasi disetiap generasi dengan jumlah gen total dalam populasi. Nilai P_m yang digunakan untuk mutasi dalam menjalankan program biasanya cukup rendah yaitu 0,001 sampai dengan 0,2 . Mutasi memiliki beberapa jenis teknik yang dapat diterapkan pada proses mutasi yaitu:

a. Mutasi *random*

Mutasi *random* disebut juga mutasi gen. Pada mutasi *random*, gen yang mengalami mutasi akan digantikan dengan gen lain dimana gen Pengganti tersebut dipilih secara acak.

Kromosom awal : 0010110010

Kromosom hasil mutasi : 0010010010

b. Mutasi *Exchange*

Mutasi *exchange* disebut juga mutasi *swap*. Pada mutasi *exchange*, secara acak dipilih 2 gen yang akan mengalami mutasi, kemudian posisi kedua Gen tersebut akan ditukar. Mutasi *exchange* digambarkan sebagai berikut

Kromosom awal : 0010110010

Kromosom hasil mutasi : 0010010110

c. *Smart Mutation*

Smart Mutation dapat mempercepat evolusi, karena mutasi terjadi pada gen yang mempunyai *fitness* value terkecil atau *penalty* terbesar. Gen *penalty* akan diganti dengan gen lain secara acak.

2.7 Algoritma Genetika Adaptif

Algoritma Genetika Adaptif (AGA) adalah modifikasi dari standar operator yang ada di AG. Ada dua karakteristik yang dianggap penting dalam AG. Yang pertama adalah kemampuan untuk mencapai nilai optimal, baik lokal maupun global, setelah menemukan wilayah yang mengandung nilai tersebut. Karakteristik yang kedua adalah kapasitas untuk menjelajahi wilayah baru dari ruang solusi untuk mencari global optimal. Keseimbangan antara karakteristik ini dapat dicapai dengan memengaruhi cara operator AG dilakukan. Operator AG yang biasanya dimodifikasi yaitu, pada operator *crossover* dan mutasi. Pemodelan operator ini dilakukan untuk dengan bantuan algoritma lainnya, seperti Particle Swarm Optimization (PSO). Pemodelan dengan algoritma lain diharapkan hasil yang diperoleh dapat terhindar dari terjebak dalam solusi optimal lokal dan menghasilkan akurasi yang tinggi untuk mendapatkan solusi optimal global dari suatu masalah [4]. Proses dari Algoritma Genetika Adaptif memiliki kesamaan dengan Algoritma genetika, hanya memiliki sedikit perbedaan di bagian operator yang akan dimodifikasi. Tetapi untuk keseluruhan alur proses dari kedua algoritma tetap sama.

2.8 Particle Swarm Optimization (PSO)

Particle Swarm Optimization merupakan teknik yang digunakan untuk mengeksplorasi ruang pencarian dari suatu masalah yang diberikan untuk menemukan parameter yang dibutuhkan untuk memaksimalkan suatu tujuan tertentu. Algoritma ini pertama kali ditemukan oleh James Kennedy dan Russell C. Eberhart pada tahun 1995 [14]. Algoritma ini dikembangkan berdasarkan simulasi interaksi sosial di antara anggota spesies tertentu yang mencari makanan atau sumber daya secara umum. Pada algoritma ini sebuah *particle* dianggap sebagai sebuah calon solusi, setiap *particle* memiliki posisi dan kecepatan dimana *particle* akan menyesuaikan dengan solusi yang paling optimal kemudian mengubah statusnya. Kumpulan dari beberapa *particle* akan membentuk suatu *swarm*, ukuran *swarm* bergantung pada seberapa kompleks masalah yang dihadapi. Algoritma PSO memiliki 3 tahapan yang dilakukan sampai tercapainya *stopping condition* yaitu,

1. Evaluasi nilai *fitness* dari setiap *particle*
2. *Personal best* and *global best fitness* and *position*
3. Memperbaharui *velocity* dan *position* dari masing-masing *particle*.

Personal best position dan *global best position* akan diperbaharui dengan cara membandingkan nilai *fitness* yang didapat pada iterasi $t + 1$ dengan iterasi ke t , dan mengganti posisi jika nilai *fitness* yang didapat pada iterasi $t + 1$ lebih besar dibandingkan dengan nilai *fitness* yang didapat pada iterasi ke t . *Stopping condition* yang diterapkan pada algoritma PSO adalah *maximum number of iterations* dan *minimum error* yang didapatkan [15]. Posisi setiap particle akan di-update melalui persamaan (2.4).

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2.4)$$

keterangan:

X_i^{t+1} : posisi dari *particle* i pada waktu ke $t + 1$

X_i^t : posisi dari *particle* i pada waktu ke t

V_i^{t+1} : *velocity*/Kecepatan dari *particle* i pada waktu ke $t+1$

Nilai V_i^{t+1} didapatkan melalui persamaan

$$V_i^{t+1} = WV_i^t + C_1 r_1^t [P_{best,i}^t - X_i^t] + C_2 r_2^t [G_{best} - X_i^t] \quad (2.5)$$

keterangan:

V_i^{t+1} : *velocity*/kecepatan dari *particle* i pada waktu ke $t + 1$

W : *inertia weight*

V_i^t : *velocity*/kecepatan dari *particle* i pada waktu ke t

C_1 : *cognitive learning factor*

C_2 : *social learning factor*

r_1 dan $r_2 \in [0,1]^d$: vektor acak

D : dimensi

$P_{best,i}^t$: (*personal best position*) posisi terbaik dari *particle* i pada iterasi ke t

G_{best} : (*global best position*) posisi dari *particle* terbaik dari *swarm*

2.9 Penelitian Sejenis

Penelitian tentang prediksi saham sudah dilakukan sebelumnya dengan penggunaan algoritma yang berbeda-beda. Pada Tabel 2.1 disajikan penelitian yang terkait dengan prediksi harga saham serta penelitian yang berkaitan dengan algoritma genetika yang

dipublikasikan pada jurnal seperti ICECA (*International Conference on Electronics, Communication and Aerospace Technology*).

Tabel 2.1 Penelitian Sejenis

No.	Judul Penelitian	Nama Penulis	Tahun	Kesimpulan
1	Prediksi Harga Saham Berdasarkan Data Historis Menggunakan Model Regresi	Asyrofa Rahmi, Wayan Firdaus Mahmudy, Budi Darma Setiawan	2015	Dari hasil penelitian ini disimpulkan bahwa parameter algoritma genetika yang optimal dalam memprediksi harga saham adalah ukuran populasi 1200, <i>crossover rate (cr)</i> 0.5 <i>mutation rate (mr)</i> 0.5 dan ukuran generasi 1500. Nilai MSE terkecil yang didapatkan sebesar 47.5023 [16].
2	<i>A Novel Adaptive Genetic Algorithm for Global Optimization of Mathematical Test Functions and Real-World Problems</i>	M.J Mahmoodabadi, A.R. Nemati	2016	Pada penelitian ini, digunakan Algoritma Genetika Adaptif (AGA) yang terdiri dari operator <i>crossover</i> dan mutasi yang dimodifikasi untuk memperkirakan permintaan minyak di iran berdasarkan indikator sosial ekonomi. Hasil penelitian ini menunjukkan AGA dapat menghasilkan solusi global optimal. Penelitian ini menggunakan probabilitas global <i>crossover</i> (PGBC= 0,9), dan probabilitas <i>quasi</i>

No.	Judul Penelitian	Nama Penulis	Tahun	Kesimpulan
				<i>sliding surface mutation</i> (PQSSm = 0,1) dengan iterasi sebanyak 100 [4].
3	<i>Stock Prediction using Genetic Algorithms and Evoluution Strategies</i>	Sonal Sabel, Anita Porwal, Upendra Singh	2017	<p>Penelitian ini bertujuan untuk membandingkan keakuratan prediksi harga saham menggunakan algortima genetika dan evoluution strategies. Kesimpulan yang diperoleh algoritma genetika mampu melakukan prediksi lebih baik dibandingkan <i>evolution strategies</i>. Eksperimen algoritma genetika yang dilakukan pada penelitian ini adalah:</p> <ul style="list-style-type: none"> c. Populasi terdiri dari 100 individu d. Probabilitas <i>crossover</i> = 0.5 e. Probabilitas <i>mutation</i> = 0.013 f. Menggunakan 1000 iterasi g. Metode seleksi yang digunakan <i>Roulette Wheel</i> [3].
4	Penjadwalan Kuliah Menggunakan Penggabungan	Rajasa Nagara E.R. Silitonga, Yan Ramadaniel Christoper Pane	2018	Berdasarkan hasil pengujian algoritma, dengan penambahan fungsi mutasi pada algoritma PSO, dapat

No.	Judul Penelitian	Nama Penulis	Tahun	Kesimpulan
	Algoritma Genetika dan <i>Particle Swarm Optimization</i> (PSO)			membantu algoritma untuk terhindar dari kondisi stagnansi, algoritma PSOM yang dihasilkan dapat mencapai akurasi 79,8% yang membutuhkan durasi \pm 21 jam dengan batasan yang masih dilanggar adalah batasan B2 dan B4 [17].
5	Peramalan Optimal Uang Tunai ATM Menggunakan <i>Gradient Boosting Decision Tree</i> dan Algoritma Genetika Adaptif	Riahta Mei Ulina Ketaren, Dwi Advenia Sinaga, Winner Immanuel Siringoringo	2019	Pada penelitian ini, dilakukan peramalan optimal uang tunai ATM menggunakan Gradient Boosting Decision Tree dan Algoritma Genetika Adaptif. Algoritma Genetika Adaptif (AGA) yang terdiri dari operator <i>crossover</i> dan mutasi dimodifikasi untuk mengatasi kelemahan AG. Hasil penelitian ini menunjukkan kombinasi GBDT dan AGA dapat model regresi yang optimal dari model regresi tanpa optimasi untuk meramal uang tunai pada suatu ATM. Parameter AGA yang digunakan adalah <i>population size</i> $1 \times$ jumlah <i>rows</i> data untuk suatu ATM, probabilitas <i>crossover</i>

No.	Judul Penelitian	Nama Penulis	Tahun	Kesimpulan
				adalah 0.85, probabilitas mutasi adalah 0.2 dan iterasi sebanyak 100 [18].

Bab 3

ANALISIS DAN DESAIN

Pada bab ini, akan menguraikan analisis dan desain yang dilakukan untuk memperoleh prediksi harga saham yang kemudian akan digunakan sebagai acuan pada proses implementasi.

3.1 Analisis Data

Pada bab ini akan dijelaskan seluruhnya mengenai analisis yang akan dilakukan pada penelitian ini, yang dimulai dari analisis dataset yang digunakan pada penelitian ini. Dataset ini akan dianalisis mulai dari tahap pengumpulan data sampai data tersebut dapat digunakan sebagai data untuk prediksi.

3.1.1 Dataset yang Digunakan

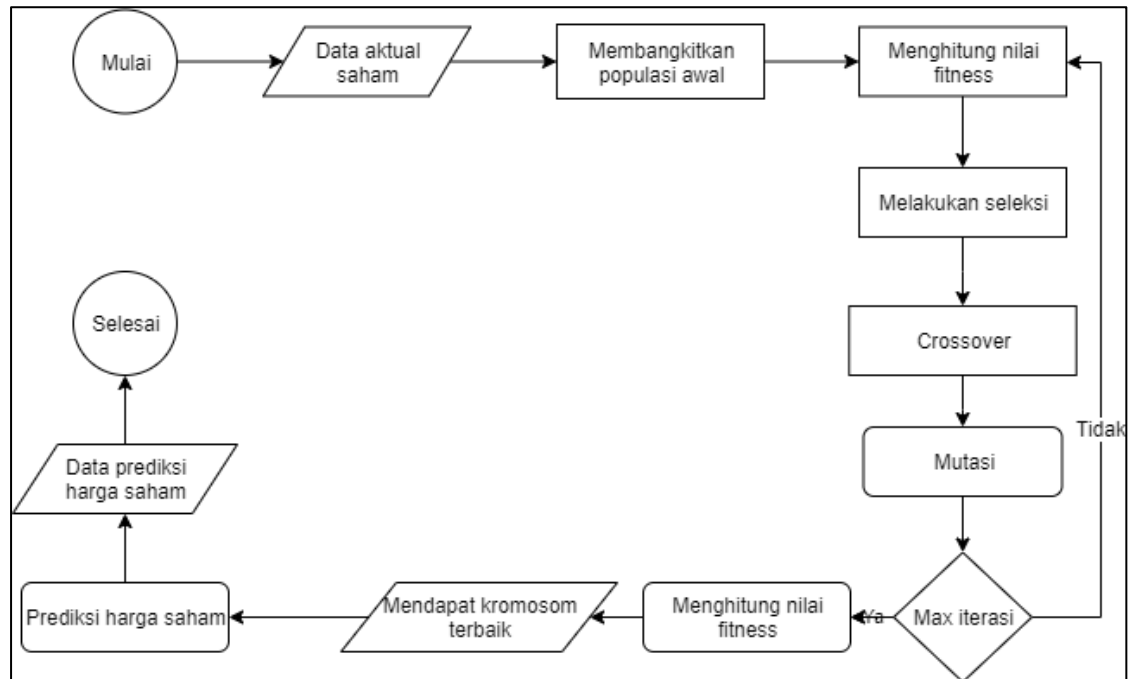
Dalam penelitian ini dataset yang digunakan berupa data harian selama tiga tahun yaitu data saham perusahaan Gudang Garam Tbk yang diperoleh dari website *Yahoo Finance*. Dataset yang digunakan merupakan *record* data perusahaan Gudang Garam Tbk terhitung dari 1 Januari 2016 hingga 1 Januari 2019. Jumlah dataset yang dimiliki pada penelitian ini adalah 762 *rows*. Atribut yang digunakan dalam memprediksi harga saham yaitu harga penutupan (*close*), karena harga penutupan adalah harga akhir saham setiap harinya. Jadi penulis akan memprediksi nilai akhir dari harga saham. Pada tahapan analisis data dilakukan dilakukan penghapusan atribut *open*, *high*, *low*, *volume* dan *adj close*. Atribut *open*, *high*, *low*, *volume* dan *adj close* dihapus karena penulis hanya membutuhkan nilai akhir dari harga saham tiap hari nya untuk diprediksi pada penelitian ini.

Jumlah data saham yang digunakan sebanyak 762 data dengan satu atribut yaitu *close*, dimana dari 762 data, 662 data digunakan sebagai data *training* dan 100 data digunakan untuk data *testing*. Pembagian dataset diperoleh berdasarkan penelitian yang telah dilakukan sebelumnya [19].

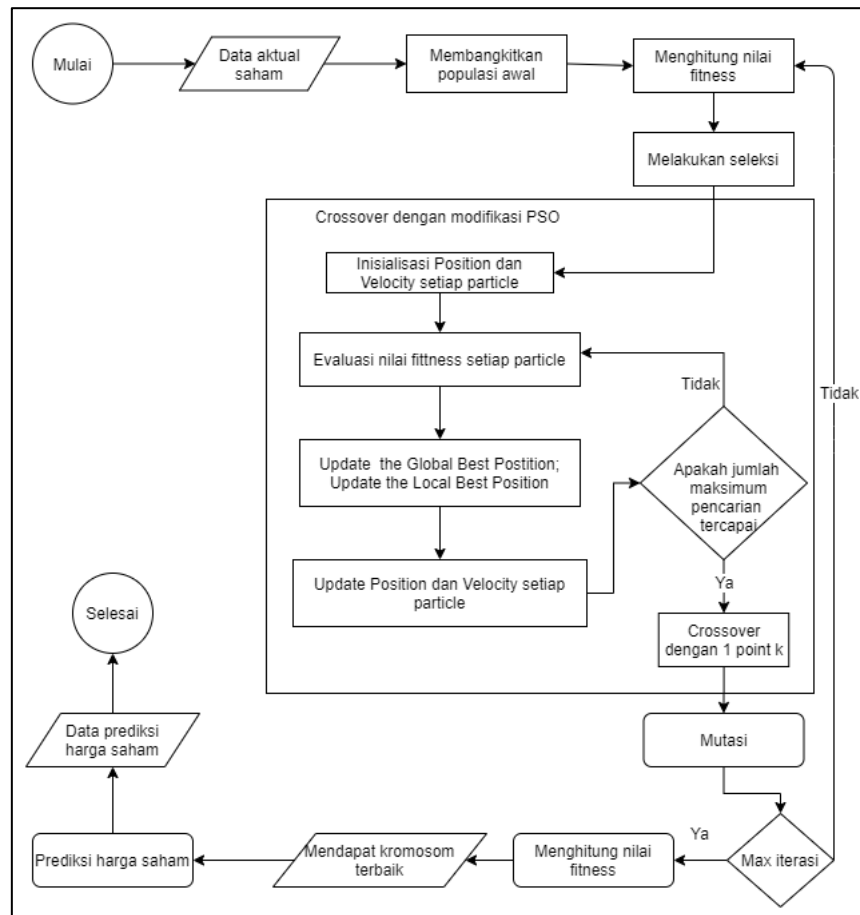
3.2 Analisis Algoritma

Pada subbab ini akan dijelaskan analisis mengenai penggunaan algoritma genetika dan algoritma genetika adaptif untuk melakukan prediksi harga saham. Langkah-langkah proses prediksi harga saham menggunakan algoritma genetika dan algoritma genetika adaptif hampir sama pada setiap tahapan, perbedaannya adalah pada algoritma genetika adaptif yaitu tahapan *crossover* dilakukan modifikasi dengan menggunakan algoritma PSO.

Rancangan rangkaian proses atau prosedur pengimplementasian algoritma genetika untuk memprediksi harga saham dengan algoritma genetika digambarkan pada *flowchart* pada Gambar 3.1. Rangkaian prosedur pengimplementasian algoritma genetika adaptif digambarkan pada Gambar 3.2.



Gambar 3.1 Flowchart Desain Simulator Algoritma Genetika



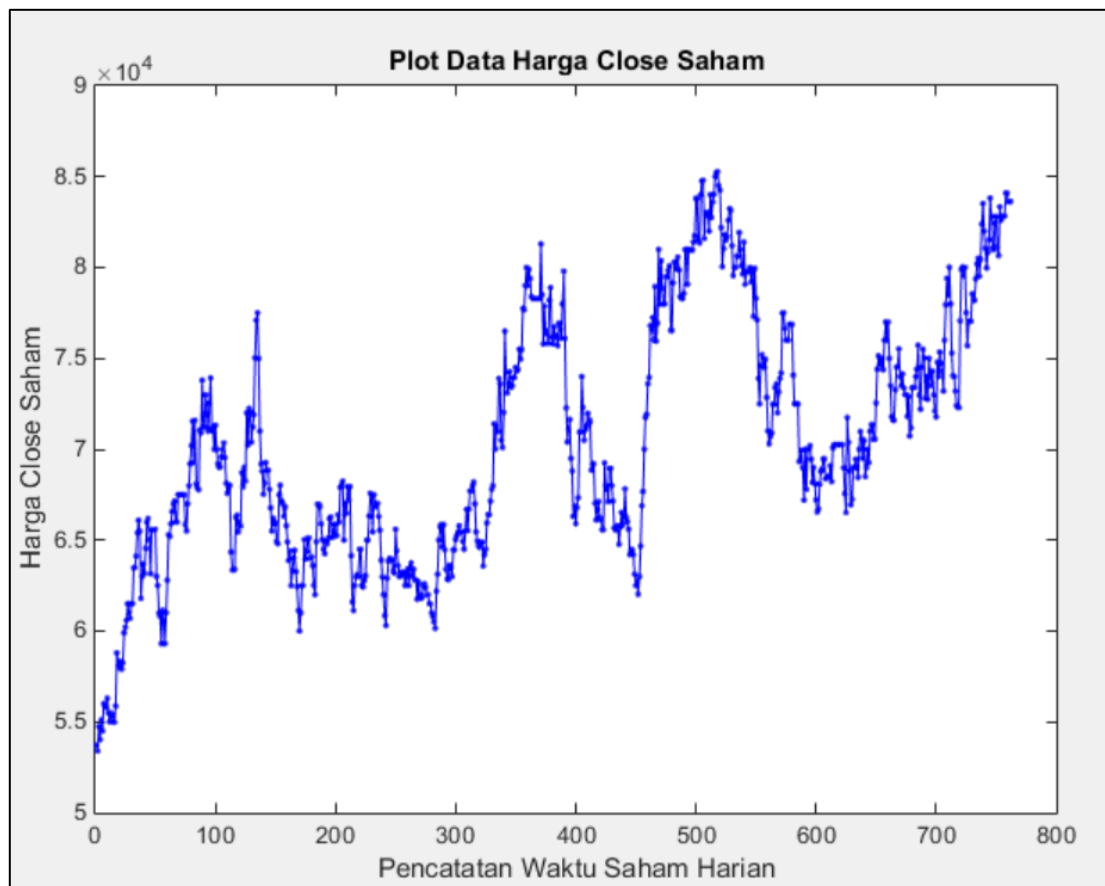
Gambar 3.2 Flowchart Desain Simulator Algoritma Genetika Adaptif

Proses prediksi harga saham menggunakan algoritma genetika sebagai berikut:

3.2.1 Membangkitkan Populasi Awal

Proses ini membangun populasi dari data saham secara *random* sebanyak 100 populasi. Jumlah populasi ini ditentukan berdasarkan jumlah populasi terbaik yang diperoleh pada penelitian sebelumnya [3]. Populasi merupakan kumpulan dari individu ataupun kromosom ataupun data saham yang akan digunakan pada Tugas Akhir ini. Dimana untuk membangkitkan populasi, perlu menentukan ukuran atau panjang dari kromosom atau data saham agar dapat mengetahui representasi kromosom atau data saham pada tiap populasi. Representasi kromosom yang digunakan adalah biner. Pada penelitian ini dilakukan prediksi harga saham untuk 5 hari selanjutnya yang disesuaikan dengan penelitian yang dilakukan oleh Asyrofa-Rahmi bahwa hasil uji coba banyaknya periode saham harian memiliki fitness rata – rata terbaik pada periode 5 hari. Hal ini menunjukkan bahwa periode yang optimal adalah periode awal 5 hari kerja. Penentuan panjang kromosom atau data saham berhubungan dengan jumlah hari prediksi yang diperoleh dari model regresi. Pada

penelitian ini, model regresi yang digunakan adalah regresi non-linear. Pada regresi non-linear, data yang dianalisis adalah data yang tidak membentuk garis lurus atau linear yang ditunjukkan Gambar 3.3.



Gambar 3.3 Plot Data Harga Close Saham

Jenis model regresi yang diterapkan adalah regresi kuadratik yang terdapat pada persamaan (2.2). Pada persamaan (2.2), jumlah n yang digunakan adalah 5 karena data harga saham yang digunakan adalah data hari ke-1 sampai hari ke-5.

Panjang kromosom dalam satu individu pada penelitian ini adalah 11 bit. Sebelas bit ini berasal dari dua kali jumlah periode data saham ditambah satu *intercept* (data aktual saham pada hari tersebut), dimana X_1, X_2, X_3, X_4, X_5 adalah data aktual harga *close* saham mulai dari hari ke-1 sampai dengan hari ke-5 dan $X_1^2, X_2^2, X_3^2, X_4^2, X_5^2$ adalah hasil perhitungan pangkat dari data aktual saham. Data saham dalam 5 periode dapat dilihat pada Tabel 3.1.

Tabel 3.1 Data Saham dengan 5 Periode

No.	Y	X_1	X_2	X_3	X_4	X_5	(X_1^2)	(X_2^2)	(X_3^2)	(X_4^2)	(X_5^2)
1	55500	56300	55800	55900	56000	54500	$(56300)^2$	$(55800)^2$	$(55900)^2$	$(56000)^2$	$(54500)^2$
2	55000	55500	56300	55800	55900	56000	$(55500)^2$	$(56300)^2$	$(55800)^2$	$(55900)^2$	$(56000)^2$

No.	Y	X_1	X_2	X_3	X_4	X_5	(X_1^2)	(X_2^2)	(X_3^2)	(X_4^2)	(X_5^2)
3	55500	55000	55500	56300	55800	55900	$(55000)^2$	$(55500)^2$	$(56300)^2$	$(55800)^2$	$(55900)^2$
4	55400	55500	55000	55500	56300	55800	$(55400)^2$	$(55000)^2$	$(55500)^2$	$(56300)^2$	$(55800)^2$
...											
672	83625	83625	83625	84100	84100	82825	$(83625)^2$	$(83625)^2$	$(84100)^2$	$(84100)^2$	$(82825)^2$

Pada Tabel 3.2 dapat dilihat contoh representasi susunan kromosom awal atau representasi data saham yang didapatkan setelah proses pembangkitan data saham terjadi. Setiap data saham dalam populasi diberikan angka 0 atau 1, maksud dari angka 0, yaitu nilai data saham yang dibangkitkan tidak digunakan dan angka 1, yaitu nilai data saham yang dibangkitkan digunakan. Contohnya, pada Gen ke-1 data saham bernilai 1, artinya pada solusi data saham tersebut *intercept* digunakan. Kemudian pada gen ke-2 yang bernilai 0, artinya pada kromosom tersebut harga saham pada sehari sebelumnya (X_1) tidak akan digunakan, kemudian nilai gen ke-3 bernilai 1, artinya pada kromosom tersebut data harga saham 2 hari sebelumnya (X_2) akan digunakan dan seterusnya sampai pada kuadrat harga saham. Angka 0 atau 1 ini dibangkitkan secara acak.

Tabel 3.2 Contoh Representasi Kromosom / Data Saham

	1	2	3	4	5	6	7	8	9	10	11
	I	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
Kromosom	1	0	1	0	0	1	1	1	0	0	1

Calon solusi yang optimal akan dipetakan ke dalam bentuk populasi yang berisi data saham dengan panjang 11. Populasi didapatkan dari proses pembentukan representasi kromosom atau data saham sejumlah ukuran populasi yang telah ditentukan pada Tabel 3.2. Berdasarkan representasi kromosom atau data saham seperti pada Tabel 3.2 maka bentuk data saham yang ada pada Tabel 3.1 akan berubah menjadi data saham yang ditunjukkan pada Tabel 3.3. Pada tulisan Tugas Akhir ini dicontohkan 10 data yang digunakan dalam proses pembentukan kromosom-kromosom atau bisa disebut data saham pada Tugas Akhir ini.

Tabel 3.3 Data Saham Menurut Kromosom Tabel 3.4

Data	Y	X_1	X_2	X_3	X_4	X_5	(X_1^2)	(X_2^2)	(X_3^2)	(X_4^2)	(X_5^2)
1	55500	0	55800	55900	0	0	$(56300)^2$	$(55800)^2$	0	0	$(54500)^2$
2	55000	0	56300	55800	0	0	$(55500)^2$	$(56300)^2$	0	0	$(56000)^2$
3	55500	0	55500	56300	0	0	$(55000)^2$	$(55500)^2$	0	0	$(55900)^2$
4	55400	0	55000	55500	0	0	$(55500)^2$	$(55000)^2$	0	0	$(55800)^2$
5	55250	0	55500	55000	0	0	$(55400)^2$	$(55500)^2$	0	0	$(56300)^2$
6	54975	0	55400	55500	0	0	$(55250)^2$	$(55400)^2$	0	0	$(55500)^2$
7	55875	0	55250	55400	0	0	$(54975)^2$	$(55250)^2$	0	0	$(55000)^2$
8	58800	0	54975	55250	0	0	$(55875)^2$	$(54975)^2$	0	0	$(55500)^2$
9	58000	0	55875	54975	0	0	$(58000)^2$	$(55875)^2$	0	0	$(55400)^2$
10	58350	0	58800	55875	0	0	$(58350)^2$	$(58800)^2$	0	0	$(55250)^2$

Populasi awal pada Tabel 3.4 akan dihitung nilai regresinya menggunakan persamaan (2.1) untuk mendapatkan satu model regresi dengan 10 nilai error pada masing-masing baris data. Dari model regresi tersebut, akan didapatkan koefisien yang akan digunakan untuk menghitung nilai RMSE dari tiap baris data. Populasi didapatkan dari proses pembentukan representasi kromosom atau data saham sejumlah ukuran populasi yang sudah ditentukan di awal. Sehingga populasi awal jika dimasukkan sejumlah 100 kromosom (*parent*) maka pembentukan populasi dapat dilihat pada Tabel 3.4.

Tabel 3.4 Pembentukan Populasi Awal

Populasi	1	2	3	4	5	6	7	8	9	10	11
	I	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
1	1	0	1	0	0	1	1	1	0	0	1
2	0	1	1	1	0	0	0	1	1	0	1
3	1	1	0	1	1	0	0	0	1	1	1
4	1	1	1	1	0	0	0	0	0	1	1
...											

100	1	0	0	0	1	1	1	1	1	0	0
-----	---	---	---	---	---	---	---	---	---	---	---

3.2.2 Evaluasi Nilai *Fitness*

Evaluasi nilai *fitness* digunakan untuk memberi nilai pada *fitness* pada setiap baris data saham yang dibangkitkan, untuk mengetahui seberapa baik data saham yang dihasilkan. Setelah pembentukan populasi awal, perhitungan nilai *fitness* dilakukan dengan menggunakan rumus pada persamaan (3.1).

$$fitness = \frac{1}{RMSE} \quad (3.1)$$

keterangan:

RMSE : Root Mean Square (dari persamaan (2.3))

Pada proses perhitungan *fitness*, dimisalkan menggunakan representasi kromosom atau data saham *parent* 1 pada Tabel 3.2 sehingga membentuk data pada Tabel 3.3, maka dilakukan pencarian koefisien regresi. Koefisien regresi dibentuk dengan membangkitkan nilai *random* dengan *interval* [0,100] dan didapatkan model koefisien seperti yang ditunjukkan pada persamaan

$$Y' = a + b_1 0 + b_2 55800 + b_3 55900 + b_4 0 + b_5 0 + b_6 (56300)^2 + b_7 (55800)^2 + b_8 0 + b_9 0 + b_{10} (54500)^2 \quad (3.2)$$

Nilai koefisien regresi yang dibentuk secara random dimisalkan dengan nilai seperti pada Tabel 3.5:

Tabel 3.5 Nilai Koefisien Regresi

a	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
56,999	65,943	23,435	54,234	43,435	23,234	76,454	64,456	51,445	86,535	78,246

Uraian proses perhitungan nilai RMSE adalah

$$Y' = 56,999 + 0 + 23,435(55800) + 54,234(55900) + 0 + 0 + 76,454(56300)^2 + 64,456(55800)^2 + 0 + 0 + 78,246(54500)^2$$

$$Y' = 6.75E + 11$$

$$E1 = (675442780010.599 - 55500)^2$$

$$E1 = 4.56E + 23$$

Proses perhitungan nilai RMSE dilakukan kepada 10 data yang terdapat pada Tabel 3.3 untuk mendapatkan data hasil regresi saham. Tabel 3.6 menunjukkan proses perhitungan hasil regresi dan RMSE setiap data.

Tabel 3.6 Proses Perhitungan Nilai RMSE

Data	Y	X ₁	X ₂	X ₃	...	(X ₅ ²)	Y'	E'
1	55500	0	55800	5900	...	(54500) ²	6.754E+11	4.562E+23
2	55000	0	56300	55800	...	(56000) ²	6.852E+11	4.695E+23
3	55500	0	55500	56300	...	(55900) ²	6.743E+11	4.547E+23
4	55400	0	55000	55500	...	(55800) ²	6.741E+11	4.544E+23
5	55250	0	55500	55000	...	(56300) ²	6.812E+11	4.64E+23
6	54975	0	55400	55500	...	(55500) ²	6.722E+11	4.519E+23
7	55875	0	55250	55400	...	(55000) ²	6.645E+11	4.416E+23
8	58800	0	54975	55250	...	(55500) ²	6.745E+11	4.55E+23
9	58000	0	55875	54975	...	(55400) ²	6.986E+11	4.88E+23
10	58350	0	58800	55875	...	(55250) ²	7.22E+11	5.213E+23

Berdasarkan data hasil regresi saham, diperoleh nilai RMSE dengan menggunakan persamaan (2.3).

$$RMSE = \frac{1}{10} \sqrt{\frac{(4.562E + 23) + (4.695E + 23) + \dots + (5.213E + 23)}{10}}$$

$$RMSE = 4.634E - 13$$

Nilai RMSE akan digunakan untuk menghitung nilai *fitness* dengan menggunakan persamaan (3.2), sehingga diperoleh nilai *fitness* sebagai berikut untuk satu populasi.

$$\begin{aligned} fitness &= \frac{1}{4.634E - 13} \\ &= 2.158E + 12 \end{aligned}$$

Pada Tabel 3.7 dihitung nilai *fitness* dari 100 populasi atau baris data saham yang telah dibangkitkan sebelumnya, seperti yang telah dimisalkan untuk populasi ke - 1

Tabel 3.7 Contoh 100 Populasi dengan Nilai *Fitness*

Populasi	1	2	3	4	5	...	8	9	10	11	<i>Fitness</i>
	<i>I</i>	<i>X</i> ₁	<i>X</i> ₂	<i>X</i> ₃	<i>X</i> ₄	...	<i>X</i> ₇	<i>X</i> ₈	<i>X</i> ₉	<i>X</i> ₁₀	
1	1	0	1	0	0	...	1	0	0	1	2.158E+12
2	0	1	1	1	0	...	1	1	0	1	1.419E+12
3	1	1	0	1	1	...	0	1	1	1	1.519E+12
...					
100	1	0	0	0	1	...	1	1	0	0	2.018E+12

3.2.3 Selection

Seleksi merupakan proses untuk mendapatkan *parent* terbaik yang digunakan untuk menghasilkan calon individu baru. Pemilihan *parent* dilakukan berdasarkan nilai *fitness* tiap baris data yang telah dibangkitkan. Metode seleksi dalam penelitian ini menggunakan *Tournament selection*, seleksi ini menerapkan prinsip turnamen, dimana akan dipilih sebanyak dua populasi secara acak untuk ditandingkan nilai *fitness*-nya. Populasi yang memiliki nilai *fitness* yang terbaik akan menjadi indukan untuk proses selanjutnya, yaitu cross over dan mutation. *Fitness value* dihitung berdasarkan nilai RMSE untuk ukuran nilai 1 sampai dengan 0. Proses *selection* menghasilkan 2 individu terbaik yaitu individu yang memiliki nilai *fitness* tertinggi.

3.2.4 Crossover

a. Crossover pada algoritma genetika

Setelah populasi diseleksi, 2 populasi hasil dari proses seleksi akan dikawinsilangkan yang disebut juga dengan *crossover*. Dengan kata lain *crossover* merupakan proses penyilangan *parent* yang terpilih untuk membentuk kromosom yang baru dengan probabilitas *crossover* (P_c) 0.5. Probabilitas *crossover* berpengaruh pada jumlah populasi yang akan mengalami persilangan. Teknik yang digunakan untuk proses *crossover* adalah *one point crossover*. *Crossover* ini melakukan penukaran gen-gen atau tiap data saham dari satu populasi dengan populasi lainnya untuk menghasilkan populasi baru melalui satu titik potong secara random. Setelah titik potong terpilih maka dilakukan proses penukaran informasi dari kedua populasi berdasarkan titik potong yang telah ditentukan. Jika proses pindah silang tidak dilakukan, maka nilai dari *parent* akan diturunkan kepada keturunannya [20]. Perbedaan nilai dalam p_c sangat berpengaruh dalam penyelesaian pada algoritma genetika, jika nilai p_c lebih dari 1, maka dapat menyebabkan banyak individu awal hilang, yang berakibat tidak adanya solusi yang ditemukan [21]. Jika *crossover* rate yang digunakan sebesar 0,5, maka nantinya akan didapatkan $0,5 \times 2 = 1$ anak dari proses *crossover*. Selanjutnya adalah menentukan posisi *crossover* atau letak individu yang akan di silangkan dengan cara membangkitkan bilangan acak dengan batasan 1 sampai dengan (panjang kromosom – 1) yaitu $11 - 1 = 10$. Sebagai contoh proses *crossover* dengan *one point crossover* dapat dilihat pada Tabel 3.8

Tabel 3.8 Contoh Crossover Populasi

Crossover	1	2	3	4	5	6	7	8	9	10	11	Fitness
	I	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	
1	1	0	1	0	0	1	1	1	0	0	1	2.158E+12
33	1	0	0	0	1	1	1	1	1	0	0	3.618E+12

Bilangan acak yang terbentuk adalah 4, sehingga titik pemotongan kromosom berada di antara gen 4 dan gen 5. Dua populasi tersebut akan dikawinsilangkan berdasarkan titik potong yang telah dibuat. Hasil dari kawin silang ini adalah dua buah individu baru dengan nilai kromosom yang telah ditukarkan satu sama lain seperti pada Tabel 3.9.

Tabel 3.9 Contoh Hasil *Crossover* Populasi

Hasil <i>Crossover</i>	1	2	3	4	5	6	7	8	9	10	11
	I	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
Populasi 2	1	0	0	0	0	1	1	1	0	0	1
Populasi 33	1	0	1	0	1	1	1	1	1	0	0

Nilai pc yang lebih kecil dapat menyebabkan jumlah kromosom baru lebih sedikit untuk generasi berikutnya sehingga menghasilkan nilai *fitness* yang lebih sedikit. Sedangkan nilai pc yang terlalu besar dapat menyebabkan jumlah kromosom baru yang lebih banyak, namun banyaknya kromosom yang mengalami *crossover* dapat merusak kromosom yang telah memiliki nilai *fitness* yang tinggi [22].

b. *Crossover* pada algoritma genetika adaptif menggunakan PSO

Pada operator *crossover* dilakukan modifikasi menggunakan PSO. Pada penelitian ini PSO digunakan untuk memilih salah satu individu yang akan menjadi *parent* kemudian akan dipasangkan pada masing-masing individu yang akan masuk ke operator *crossover* secara acak untuk dikawinkan.

Istilah-istilah yang digunakan dalam penerapan algoritma PSO adalah sebagai berikut:

- Particle* : individu atau data saham pada suatu swarm. Setiap *particle* merepresentasikan suatu solusi yang potensial yaitu baris data saham pada permasalahan yang diselesaikan.
- Swarm* : populasi dari sekawanan individu ataupun data saham
- Pbest (Personal best)* : suatu individu yang menunjukkan posisi terbaik pada suatu iterasi
- Gbest (Global best)*: posisi terbaik individu pada *swarm* atau posisi terbaik diantara *Pbest* yang ada.
- Velocity (v)* : vektor yang menggerakkan proses optimisasi yang menentukan arah dimana suatu individu diperlukan untuk berpindah (*move*) untuk memperbaiki posisinya semula atau kecepatan yang menggerakkan proses optimasi yang

menentukan arah di mana individu diperlukan untuk berpindah dan memperbaiki posisinya semula.

- f. *Inertia weight*: parameter yang digunakan untuk mengontrol dampak dari adanya *velocity* yang diberikan oleh suatu individu.

Inisialisasi algoritma PSO dimulai dengan menetapkan posisi awal individu secara acak dan kemudian mencari nilai optimal dengan memperbarui posisinya. Pada tahapan *crossover* dengan modifikasi PSO akan menerima populasi terbaik yang telah dipilih pada tahap *selection*. Selanjutnya dilakukan inisialisasi nilai parameter yang ada pada algoritma PSO, yaitu *velocity* dan *position* pada setiap individu. Semua individu bergerak menuju titik optimal dengan suatu kecepatan. Awalnya semua kecepatan dari individu diasumsikan sama dengan nol pada iterasi $i = 1$.

Dimisalkan pada tahap *selection* sebelumnya, dari 8 individu terpilih 4 individu yang nilai *fitness* nya memenuhi aturan 50% RMSE untuk masuk ke dalam operator *crossover* yang telah dimodifikasi dengan PSO.

Tabel 3.10 Contoh Individu yang Terpilih untuk Dimodifikasi

Individu	<i>Fitness Value</i>
1	0.56E-02
17	0.54E-30
24	0.55E-07
33	0.55E-80

Dengan algoritma PSO ini, akan dihasilkan satu individu yang akan menjadi *parent*. Berikut merupakan proses pencarian satu individu tersebut dengan menggunakan PSO. Berdasarkan Tabel 3.10 terdapat 4 individu. Individu tersebut yang disebut dengan *particle* dalam algoritma PSO. Sehingga individu yang terbentuk adalah x_1 , x_2 , x_3 dan x_4 . Misalkan inisial posisi dari masing-masing individu yaitu $x_1 = -2$, $x_2 = -1$, $x_3 = 3$ dan $x_4 = 5$. Kemudian setiap individu akan dihitung nilai *fitness* nya. Penghitungan nilai *fitness* dilakukan untuk mendapatkan individu dengan nilai *fitness* yang maksimum.

Posisi awal partikel, kecepatan dan nilai *fitness* untuk setiap partikel dapat dilihat pada Tabel 3.11, Kecepatan setiap partikel bernilai 0 pada iterasi ke $t = 0$. Persamaan (2.4)

digunakan untuk melakukan *update position* dan persamaan (2.5) digunakan untuk melakukan *update velocity*. Untuk mendapatkan nilai posisi dari suatu individu, dibutuhkan nya nilai *velocity* individu terlebih dahulu untuk iterasi berikutnya. Nilai dari *velocity* didapatkan dari beberapa parameter yang ditentukan dari awal, seperti *Inertia weight* (W), *Cognitive learning factor* (C_1) dan *Social learning factor* (C_2). *Inertia weight* berfungsi mengontrol *velocity* dari waktu ke waktu sehingga *velocity* mengalami penurunan yang membuat PSO mencari ruang yang lebih besar di awal untuk mendapatkan posisi dengan cepat yaitu solusi yang paling optimis. *Inertia weight* yang besar dapat memudahkan dalam penelusuran global sedangkan *inertia weight* yang kecil memudahkan untuk penelusuran lokal. *Inertia weight* diset dengan nilai antara 0.4 sampai 0.9. Untuk penelitian ini nilai *inertia* yang dipakai yaitu 0.5. Nilai ini dipilih berdasarkan hasil percobaan yang dilakukan untuk bobot *inertia weight* yaitu, apabila bobot *inertia* yang digunakan semakin kecil maka kecepatan partikel akan semakin cepat sehingga kesempatan eksploitasi partikel semakin kecil dan partikel akan lebih cepat melakukan eksplorasi. Untuk nilai faktor asselarasi c_1 dan c_2 memiliki rentang nilai antara $0 < c_1, c_2 \leq 2$. Konstanta tersebut disebut sebagai koefisien akselerasi yang mempengaruhi jarak maksimum yang dapat diambil oleh sebuah partikel dalam sebuah iterasi. Nilai faktor asselarasi c_1 dan c_2 yang digunakan sama yaitu 2.

Tabel 3.11 Inisialisasi Posisi, Kecepatan dan Nilai *Fitness* Individu

$F_{[Individu, Iterasi]}$	Posisi	Nilai <i>Fitness</i>	Kecepatan ($V_{[partikel, iterasi]}$)
$F_{[1,0]}$	-2	0.67E-12	$V_{[1,0]} = 0$
$F_{[33,0]}$	-1	0.36E-02	$V_{[2,0]} = 0$
$F_{[24,0]}$	3	0.45E-08	$V_{[3,0]} = 0$
$F_{[17,0]}$	5	0.22E-02	$V_{[4,0]} = 0$

Pada Tabel 3.11, nilai Pbest dari individu merupakan posisi awal dari masing-masing individu untuk iterasi $t = 0$, dan nilai Gbest yang didapat adalah nilai maksimum *fitness* dari seluruh individu, yaitu individu pada posisi ke -2. Setelah tahap ini akan dilakukan iterasi selanjutnya untuk memperbarui posisi dan *velocity* individu menggunakan persamaan (2.4) dan persamaan (2.5), setelah mendapatkan posisi dan *velocity* serta nilai *fitness* individu yang baru. Tahapan selanjutnya yaitu membandingkan *local best*

dan *global best* saat ini dengan iterasi sebelumnya, lalu memperbaharui nilai *local best* dan *global best* yang lebih baik. Hal ini akan terus dilakukan sampai iterasi yang penulis tentukan selesai dilakukan. Hasil individu yang didapat melalui modifikasi menggunakan PSO, akan dicrossover dengan individu lainnya yang terdapat pada tabel Tabel 3.11 menggunakan teknik *one point crossover* untuk menghasilkan individu baru.

3.2.5 Mutation

Mutation merupakan proses untuk meningkatkan variansi populasi. Pada proses ini, mutasi memiliki peran untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Probabilitas mutasi (P_m) yang digunakan $0.013 \times pop_size$. Probabilitas mutasi ini digunakan untuk mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika nilai P_m yang digunakan terlalu kecil, tingkat kemiripan anak dengan parent sangat tinggi, yang dimana proses ini bertujuan untuk meningkatkan variansi. Tetapi bila nilai p_m terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari parent nya. Untuk tugas akhir ini, jumlah populasi yang dimutasi $= 0.13 * pop_size$. Jadi akan ada 0.13 peluang gen yang akan terjadi mutasi.

Tabel 3.12 Contoh Mutasi Populasi

Mutasi	1	2	3	4	5	6	7	8	9	10	11
	I	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
2	1	0	0	0	0	1	1	1	1	0	1
33	1	0	1	0	1	1	1	1	0	0	0

Tabel 3.12 merupakan hasil dari *crossover* populasi yang ada pada Tabel 3.9 yang telah dimutasi. Modifikasi mutasi dilakukan secara acak, dimisalkan mutasi dilakukan pada data saham ke – 9. Populasi 2 yang semula nya bernilai 0 pada data saham ke – 9 diubah menjadi bernilai 1 dan populasi 33 yang semula nya bernilai 1 pada data saham ke – 9 diubah menjadi bernilai 0.

3.3 Product Main Function

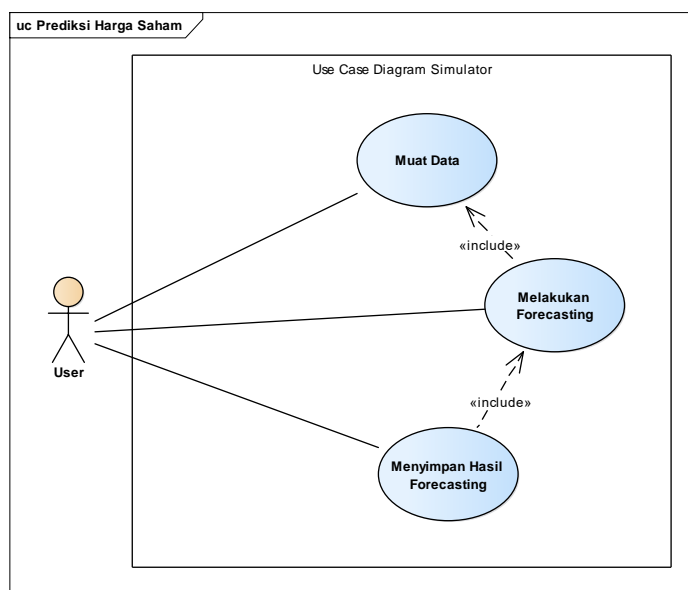
Pada subbab *product main function* dijelaskan list fungsi yang akan digunakan pada simulator prediksi harga saham yang dibangun.

Tabel 3.13 Fungsi Utama Simulator dan Deskripsi Fungsi

Nomor Fungsi	Nama Fungsi	Deskripsi
FR-01	Muat Data	Fungsi ini digunakan untuk memuat data aktual saham dalam format .xlsx yang akan dilanjutkan ke proses pembersihan data.
FR-02	Melakukan <i>forecasting</i>	Fungsi ini digunakan untuk melakukan prediksi harga saham.
FR-03	Menyimpan hasil <i>forecasting</i>	Fungsi ini berguna untuk menyimpan data hasil prediksi harga saham.

3.4 Use case Pada Simulator

Pada bagian *use case* simulator dijelaskan fungsi-fungsi yang ada pada simulator prediksi harga saham, yang dapat dilihat pada Gambar 3.4.



Gambar 3.4 Use case Diagram Simulator

Pada *use case* diagram menjelaskan terdapat satu aktor yaitu *user* atau pengguna yang dapat melakukan interaksi terhadap simulator. Terdapat 3 kebutuhan fungsional yang dapat dilakukan *user* pada simulator yaitu memuat data, melakukan *forecasting* data saham, dan menyimpan hasil *forecasting*. Langkah yang pertama yang dilakukan *user* adalah memuat data. Jika berhasil memuat data, *user* dapat melakukan *forecasting* yaitu melakukan

prediksi harga saham. Kemudian data prediksi saham akan ditampilkan pada simulator. Fungsi selanjutnya adalah menyimpan hasil prediksi harga saham.

3.5 Use case Scenario

Use case scenario akan dijelaskan lebih detail pada subbab ini.

3.5.1 Use case Scenario FR-01

Deskripsi *use case scenario* fungsi memuat data dapat dilihat pada Tabel 3.14.

Tabel 3.14 Use case Scenario FR-01

<i>Use case Name</i>	UC-01 Memuat Data	
<i>Stakeholders and Interest</i>	User memuat data ke dalam simulator	
<i>Actor</i>	User	
<i>Preconditions</i>	Menjalankan simulator	
<i>Basic Flow Event</i>	<i>User Action</i>	<i>System Response</i>
	1. Meng-klik tombol Muat Data	
		2. Menampilkan tampilan untuk jendela baru penyimpanan folder dan file pada PC dengan ekstensi .xlsx
	3. Memilih data yang akan digunakan dalam prediksi.	
		4. Simulator menghasilkan plot/grafik data
<i>Alternate Flows</i>	-	-

<i>Error Flow</i>	1. Memasukkan file dengan format yang salah.	
		2. Menampilkan pesan <i>error</i> bahwa format data file tidak valid.
<i>Postconditions</i>	Data berhasil dimuat ke dalam simulator	

Fungsi memuat data dilakukan untuk meng-*import* data pada simulator. *User* memilih tombol Muat Data, kemudian simulator akan menampilkan jendela baru penyimpanan folder dan file pada PC dengan ekstensi .xlsx. *User* memilih data harga saham yang ingin di-*import*. Jika berhasil maka simulator akan menampilkan plot/grafik data harga saham. Jika format yang dimasukkan salah maka akan muncul pesan *error* yang memberitahukan bahwa format data file tidak valid.

3.5.2 Use case Scenario FR-02

Deskripsi *use case scenario* fungsi melakukan *forecasting* dapat dilihat pada Tabel 3.15.

Tabel 3.15 Use case Scenario FR-02

<i>Use case Name</i>	UC-02 Melakukan <i>Forecasting</i>	
<i>Stakeholders and Interest</i>	<i>User</i> melakukan <i>forecasting</i>	
<i>Actor</i>	<i>User</i>	
<i>Preconditions</i>	Data berhasil dimuat	
<i>Basic Flow Event</i>	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> melakukan <i>import</i> data saham dengan format Excel.	
		2. Simulator menampilkan data aktual harga saham

	3. <i>User</i> meng-klik tombol prediksi untuk GA ataupun AGA	
		4. Simulator memproses <i>forecasting</i>
		5. Simulator menampilkan hasil prediksi harga saham
<i>Alternate Flows</i>	-	-
<i>Error Flow</i>	1. Data belum dimuat	
		2. Menampilkan pesan <i>error</i> bahwa data belum dimasukkan
<i>Postconditions</i>	Data prediksi harga saham ditampilkan	

Fungsi melakukan *forecasting* dapat dilakukan setelah berhasil memuat data. Data saham yang telah di-import akan diproses menggunakan algoritma genetika dan algoritma genetika adaptif. Langkah-langkah yang dilakukan *user* adalah memilih tombol Prediksi algoritma genetika ataupun algoritma genetika adaptif. Jika berhasil melakukan *forecasting* maka simulator akan menampilkan hasil prediksi saham. Jika data belum dimuat maka akan ditampilkan pesan *error* yang menampilkan bahwa data belum dimasukkan.

3.5.3 Use case Scenario FR-03

Deskripsi *use case scenario* menyimpan hasil prediksi harga saham dapat dilihat pada Tabel 3.16.

Tabel 3.16 Use case Scenario FR-03

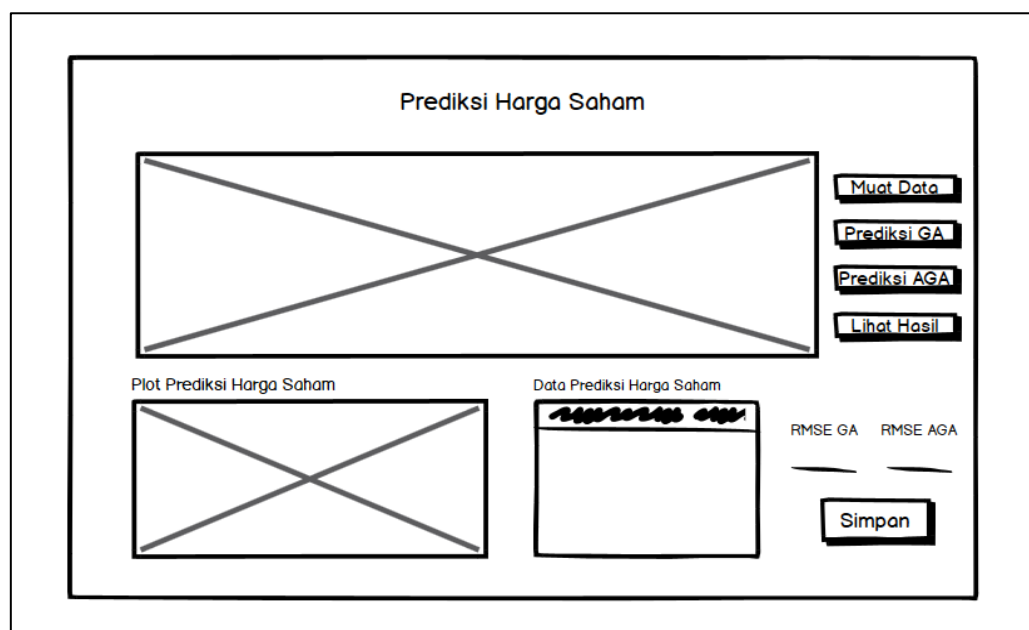
<i>Use case Name</i>	UC-03 Menyimpan hasil <i>forecasting</i>
<i>Stakeholders and Interest</i>	<i>User</i> mengunduh hasil <i>forecasting</i>
<i>Actor</i>	<i>User</i>

<i>Preconditions</i>	Melakukan <i>forecasting</i>	
<i>Basic Flow Event</i>	<i>User Action</i>	<i>System Response</i>
	1. Meng-klik tombol Simpan Hasil Prediksi	
		2. Mengunduh Data yang telah di- <i>forecasting</i>
<i>Alternate Flows</i>	-	-
<i>Error Flow</i>		2. Menampilkan pesan <i>error</i> bahwa <i>file</i> tidak diketahui.
<i>Postconditions</i>	Data berhasil diunduh ke dalam PC	

Fungsi mengunduh file data prediksi saham dapat dilakukan jika berhasil melakukan forecasting data. *User* dapat memilih tombol Unduh Data. Jika berhasil file akan terdownload ke PC *user*.

3.6 Desain Antarmuka Simulator

Pada subbab ini akan dijelaskan mengenai desain simulator yang akan dibangun.



Gambar 3.5 Desain Antarmuka Simulator

Desain antarmuka di atas terdiri dari 5 buah tombol yaitu:

1. Tombol Muat Data berfungsi untuk memuat data harga saham, setelah data dimuat maka akan ditampilkan grafik dari harga saham yang akan diprediksi.
2. Tombol Prediksi AG berfungsi untuk melakukan prediksi harga saham yang telah dimuat dengan algoritma genetika.
3. Tombol Prediksi AGA berfungsi untuk melakukan prediksi harga saham yang telah dimuat dengan algoritma genetika adaptif.
4. Tombol Lihat Hasil berfungsi untuk melihat hasil prediksi Algoritma Genetika dan Algoritma Genetika Adaptif
5. Tombol Simpan Hasil Prediksi berfungsi untuk mengunduh hasil prediksi yang didapatkan dari simulator.

Bab 4

IMPLEMENTASI DAN TESTING

Pada bab ini akan dibahas mengenai implementasi simulator yang telah dirancang dan pengujian algoritma genetika dan algoritma genetika adaptif.

4.1 Implementasi

Pada subbab implementasi dijelaskan tentang implementasi yang telah dilakukan sesuai dengan analisis dan desain yang telah dijelaskan pada bab 3. Pada bab ini akan dibahas mengenai implementasi simulator prediksi harga saham dengan algoritma genetika dan algoritma genetika adaptif

4.1.1 Lingkungan Implementasi

Lingkungan implementasi pada subbab ini terbagi atas perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan dalam proses implementasi. Detail dari perangkat keras dan perangkat lunak akan dijelaskan secara detail pada bagian ini.

1. Perangkat Keras

Perangkat keras yang digunakan untuk implementasi penelitian Tugas Akhir ini dijelaskan pada Tabel 4.1.

Tabel 4.1 Spesikasi Perangkat Keras

Hardware	Spesifikasi
Laptop	Lenovo ideaped 310
<i>Processor</i>	Intel® Core™ i5-7200U
CPU	@ 2.50GHz (4 CPUs), ~2.7GHz
<i>Memory</i>	8192 MB RAM

2. Perangkat Lunak

Perangkat lunak yang digunakan untuk implementasi Tugas Akhir ini dijelaskan pada Tabel 4.2.

Tabel 4.2 Spesifikasi Perangkat Lunak

Software	Spesifikasi
<i>Operating System</i>	Windows 10
<i>Tools</i>	Matlab
Bahasa Pemrograman	Matlab

4.1.2 Implementasi Antarmuka

Tampilan antarmuka simulator prediksi harga saham memiliki 1 tampilan yaitu tampilan utama. Tampilan utama simulator dapat dilihat pada gambar berikut ini.

**Gambar 4.1 Implementasi Tampilan Utama Antarmuka Simulator**

Keterangan Gambar 4.1 :

Tampilan utama terbagi atas 4 panel yaitu,

- Panel 1 : Panel Muat Data Saham

Panel ini berguna untuk menampilkan grafik data harga saham. Grafik yang digunakan adalah jenis grafik garis yang menggambarkan perubahan harga saham dari waktu ke waktu. Grafik data harga saham terdiri dari sumbu X dan sumbu Y. Dimana sumbu X

merupakan data waktu saham per hari. Sedangkan sumbu Y merupakan data harga saham dalam puluhan ribu.

- Panel 2 : Menampilkan Grafik prediksi harga saham

Panel ini berguna untuk menampilkan hasil prediksi harga saham dengan algoritma genetika ketika ditekan tombol prediksi AG dan prediksi harga saham dengan algoritma genetika adaptif jika menekan tombol prediksi AGA. Hasil prediksi harga saham dibuat dalam bentuk grafik.

- Panel 3 : Menampilkan data prediksi harga saham

Panel ini terdiri dari satu baris dan lima kolom. Kolom berisi data prediksi harga saham menggunakan AG, ketika ditekan tombol prediksi AG, dan hasil data prediksi AGA akan muncul jika menekan tombol prediksi AGA.

- Panel 4 : Menampilkan nilai RMSE AG dan AGA

Panel ini berguna untuk menampilkan nilai RMSE dari kedua algoritma yang digunakan.

4.1.2.1 Tahapan Persiapan Data

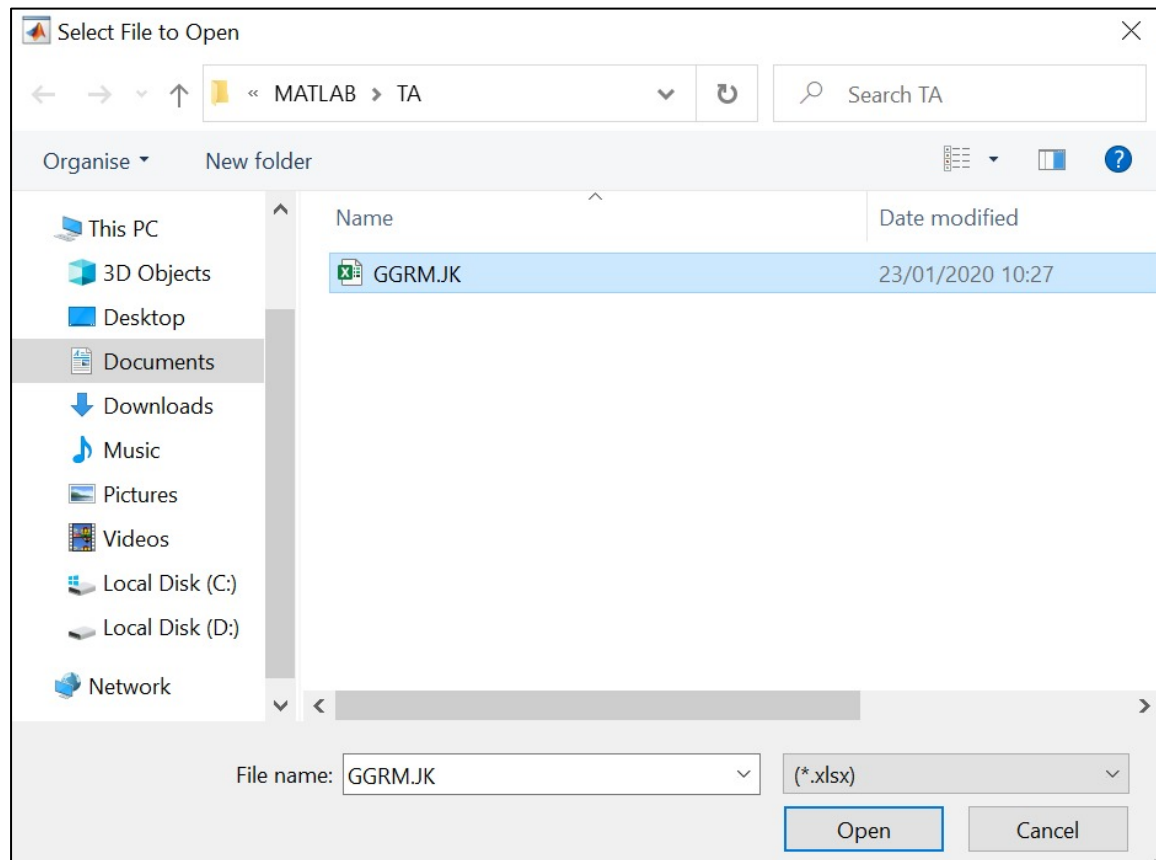
Langkah pertama yang dilakukan untuk prediksi harga saham menggunakan AG dan AGA adalah persiapan data. Sebelum melakukan tahapan ini, data aktual saham harus dipersiapkan terlebih dahulu. Fungsi yang digunakan dalam simulator pada tahapan persiapan data adalah FR-01 yaitu muat data. Kode program yang digunakan untuk memuat data dapat dilihat pada

```
% persiapan data train dan data test
% baca file excel dan txt
[hargaSahamXls,path] = uigetfile('.xlsx');
% buka data excel
[dataExcel] = xlsread(fullfile(path,hargaSahamXls));
data_size = length(dataExcel);
lenTrain = 662;
lenTest = data_size-lenTrain;
```

Code 4.1 Kode Program Muat Data

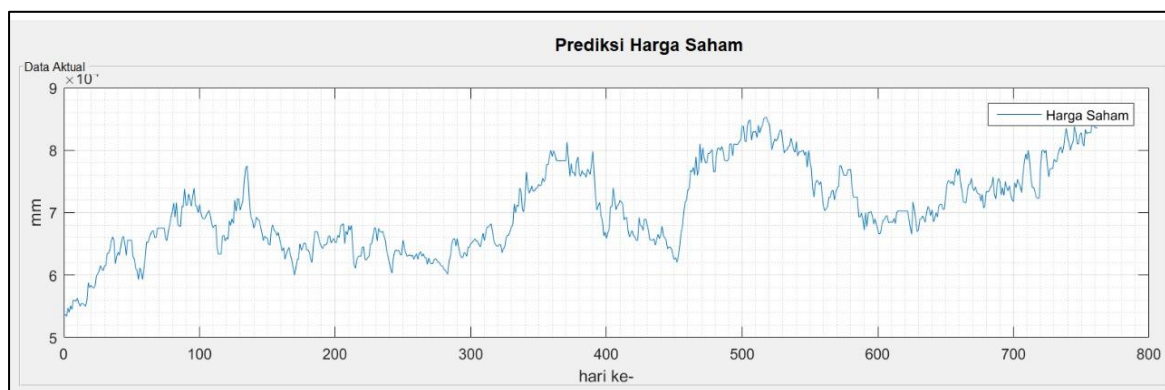
Proses persiapan data dapat dilakukan dengan menekan *button* Muat Data. Simulator akan memunculkan *window* untuk pemilihan file data aktual saham yang memiliki ekstensi .xlsx. Pada saat *button* muat data diklik, maka simulator secara otomatis akan membagi data menjadi data training dan data test. Pembagian data training dan dataset tersebut dilakukan

sesuai dengan analisis dataset yang digunakan pada bab 3.1.1. Tahapan persiapan data dapat dilihat pada Gambar 4.2.



Gambar 4.2 Window Select File Data Aktual Saham

Setelah memilih file data aktual saham yang akan digunakan, file tersebut akan diproses kemudian data akan ditampilkan pada panel 1 dalam bentuk plot. Plot yang ditampilkan merupakan data harga *close* (harga penutupan) saham setiap harinya. Implementasi plot tersebut dapat dilihat pada Gambar 4.3.



Gambar 4.3 Implementasi Plot Data Aktual Saham

Kode program untuk gambar Implementasi Plot Data Aktual Saham dapat dilihat pada Code 4.2

```
axes(handles.plotDataSaham )
plot(dataExcel);
grid on
grid minor
legend('Harga Saham');
xlabel('Hari ke-');
ylabel('Rp');
```

Code 4.2 Kode Program Plot Data Aktual Saham

Berdasarkan kode program plot data aktual saham, sumbu X menunjukkan hari yang merupakan waktu pencatatan harian sedangkan sumbu Y menunjukkan jumlah harga close saham.

4.1.2.2 Implementasi Algoritma Genetika

Hasil implementasi Algoritma Genetika akan ditampilkan pada panel 2, yang disajikan dalam bentuk grafik. Proses implementasi algoritma genetika dilakukan sesuai tahapan algoritma genetika seperti yang ada pada bab 3.2 yaitu mulai dari membangkitkan populasi awal sampai ke tahap mutasi.

a. Pembentukan Populasi Awal

Tahapan implementasi pertama yang dilakukan adalah membangkitkan populasi awal. Populasi yang dibentuk ada sebanyak 100 populasi sesuai dengan analisis algoritma pada bab sebelumnya. Parameter algoritma yang digunakan adalah sebagai berikut:

Ukuran populasi : 100
Crossover Rate : 0.5
Mutation Rate : 0.013

Iteration : 1000

Jumlah populasi ini ditulis pada *code* yaitu variabel *popSize*. Kode program untuk membangkitkan populasi dapat dilihat pada Code 4.3.

```
m = 1;
while m <= popSize
% Inisialisasi populasi awal
[population.kromosom(m).gen, population.kromosom(m).koefisien] = initialize_population(
popSize, colTarget );

% Menghitung nilai fitness
[population.kromosom(m).fitness, yAksen(m)] = calculate_fitness( target,
population.kromosom(m).gen, population.kromosom(m).koefisien );

% Megecek apakah anda nilai regresi yang negatif
neg = any(yAksen(m)<0);
if (neg) == 1
    m = m-1;
end
    m = m+1
end
```

Code 4.3 Kode Program Membangkitkan Populasi Awal

Kode program membangkitkan populasi awal memiliki *code* yang berkelanjutan yaitu fungsi *initialize_population* yang ditunjukkan pada Code 4.4.

```
function [ gen, koefisien ] = initialize_population( ukuran_populasi, panjang_kromosom )
try
    for j = 1 : panjang_kromosom
        gen(j) = randi([0 1]);
        koefisien(j) = -1 + (2) .* rand(1,1);
    end
catch
    msgbox('Proses inisialisasi populasi gagal');
end
```

Code 4.4 Kode Program Pembentukan Kromosom dan Koesfisien

Fungsi *initialize_population* akan menghasilkan 11 kromosom dan 11 koefisien pada setiap populasi.

b. Evaluasi Nilai *Fitness*

Tahapan selanjutnya setelah populasi telah terbentuk adalah evaluasi nilai *fitness*. Pada tahap ini setiap populasi akan dihitung nilai *fitness*nya. Nilai *fitness* diperoleh dengan cara

menghitung nilai RMSE terlebih dahulu seperti pada persamaan (3.1). Kode program menghitung nilai *fitness* dituliskan pada Code 4.5.

```
% PROSES PERHITUNGAN NILAI FITNESS
function [fitness,regresi] = calculate_fitness( dataset, populasi, koefisien )
try
    global rowDataset;
    global colDataset;
    [rowDataset, colDataset] = size(dataset);

    % nilai awal sumx di-state sebagai array pertama chromosome
    sumx = zeros;
    for i=1:rowDataset
        sumx(i) = populasi(1) * koefisien(1);
        for j=2:colDataset
            sumx(i) = sumx(i) + (populasi(j) * koefisien(j) * dataset(i,j));
        end
    end
    regresi = sum(sumx);
    square = 0;
    jumlah = 0;
    for i=1:rowDataset
        square = power((dataset(i,1)-sumx(i)),2);
        jumlah = jumlah + square;
    end
    error = sqrt(jumlah)/rowDataset;
    fitness = 1/error;
catch
    msgbox('Proses penghitungan nilai fitness gagal');
end
```

Code 4.5 Kode Program Evaluasi Nilai *Fitness* Setiap Populasi

Kode program menghitung nilai *fitness* mengandung proses menghitung nilai regresi dan nilai RMSE.

c. Proses Seleksi

Setelah setiap populasi memiliki nilai *fitness* masing-masing maka akan dilakukan proses seleksi. Pada tahapan ini proses seleksi yang digunakan adalah jenis *Tournamen Selection* seperti yang telah ditetapkan pada bab sebelumnya. Dimana pada proses ini masing-masing nilai *fitness* pada setiap populasi akan diurutkan dan dipilih sesuai dengan ketentuan pada bab 3.2.3 Selection. Populasi yang terpilih akan masuk ke proses selanjutnya, yaitu *crossover* dan proses mutasi. Kode program untuk melakukan proses seleksi dapat dilihat pada Code 4.6.

```
function [parent1, parent2] = selection(population)
```

```

M = length(population.Chromosomes(:));

if any([population.Chromosomes(:).fitness] < 0 )
    % Fitness scaling in case of negative values scaled(f) = a * f + b
    a = 1;
    b = abs( min( [population.Chromosomes(:).fitness] ) );
    Scaled_fitness = a * [population.Chromosomes(:).fitness] + b;

    normalized_fitness = [Scaled_fitness] ./ sum([Scaled_fitness]);
else
    normalized_fitness = [population.Chromosomes(:).fitness] ./
sum([population.Chromosomes(:).fitness]);
end

%normalized_fitness = [population.Chromosomes(:).fitness] ./
sum([population.Chromosomes(:).fitness]);

[sorted_fitness_values , sorted_idx] = sort(normalized_fitness , 'descend');

for i = 1 : length(population.Chromosomes)
    temp_population.Chromosomes(i).Gene = population.Chromosomes(sorted_idx(i)).Gene;
    temp_population.Chromosomes(i).fitness = population.Chromosomes(sorted_idx(i)).fitness;
    temp_population.Chromosomes(i).normalized_fitness = normalized_fitness(sorted_idx(i));
end

cumsum = zeros(1 , M);

for i = 1 : M
    for j = i : M
        cumsum(i) = cumsum(i) + temp_population.Chromosomes(j).normalized_fitness;
    end
end

R = rand(); % in [0,1]
parent1_idx = M;
for i = 1: length(cumsum)
    if R > cumsum(i)
        parent1_idx = i - 1;
        break;
    end
end
parent2_idx = parent1_idx;
while_loop_stop = 0; % to break the while loop in rare cases where we keep getting the same
index
while parent2_idx == parent1_idx
    while_loop_stop = while_loop_stop + 1;
    R = rand(); % in [0,1]
    if while_loop_stop > 20
        break;
    end
    for i = 1: length(cumsum)
        if R > cumsum(i)
            parent2_idx = i - 1;

```

```

        break;
    end
end
end
parent1 = temp_population.Chromosomes(parent1_idx);
parent2 = temp_population.Chromosomes(parent2_idx);
end

```

Code 4.6 Kode Program Proses Seleksi

d. Proses *Crossover*

Setelah melewati proses *selection* proses selanjutnya adalah proses *crossover*. Pada tahapan ini individu yang terpilih akan dikawinsilangkan dengan menggunakan *one point cut crossover* sesuai dengan analisis algoritma pada bab sebelumnya. Kode program *crossover* dapat dilihat pada Code 4.7.

```

% proses crossover
function [child1,child2] = crossover(parent1,parent2, cr_rate)
try
    global colTarget;
    child1 = parent1;
    child2 = parent2;
    ub = colTarget - 1;
    lb = 1;
    Cross_P1 = round ( (ub - lb) *rand() + lb );
    Cross_P2 = Cross_P1;
    while Cross_P2 == Cross_P1
        Cross_P2 = round ( (ub - lb) *rand() + lb );
    end
    if Cross_P1 > Cross_P2
        temp = Cross_P1;
        Cross_P1 = Cross_P2;
        Cross_P2 = temp;
    end
    Part1 = parent1.gen(1:Cross_P1);
    Part2 = parent2.gen(Cross_P1 + 1 :Cross_P2);
    Part3 = parent1.gen(Cross_P2+1:end);
    child1.gen = [Part1 , Part2 , Part3];
    Part1 = parent2.gen(1:Cross_P1);
    Part2 = parent1.gen(Cross_P1 + 1 :Cross_P2);
    Part3 = parent2.gen(Cross_P2+1:end);
    child2.gen = [Part1 , Part2 , Part3];
    R1 = rand();
    if R1 <= cr_rate
        child1 = child1;
    else
        child1 = parent1;
    end
    R2 = rand();
    if R2 <= cr_rate
        child2 = child2;
    end
end

```



```

else
    child2 = parent2;
end
catch
    msgbox('Proses cross over gagal');
end

```

Code 4.7 Kode Program Proses Crossover

Hasil dari proses *crossover* adalah populasi yang memiliki kromosom yang telah berubah karena proses kawin silang.

e. Proses Mutasi

Proses selanjutnya adalah proses mutasi dimana, terdapat probabilitas mutasi yang digunakan yaitu $0.013 \times pop_size$ seperti yang telah dijelaskan pada bab sebelumnya. Kode program proses mutasi dapat dilihat pada Code 4.8.

```

% proses mutasi
function [child] = mutation(child,mutation_rate)
try
    global colTarget;
    for k = 1: colTarget
        R = rand();
        if R < mutation_rate
            child.gen(k) = ~ child.gen(k);
        end
    end
catch
    msgbox('Proses mutasi gagal');
end

```

Code 4.8 Kode Program Proses Mutasi

f. Proses Regenerasi

Populasi yang melewati proses mutasi selanjutnya akan melewati proses regenerasi, dimana pada tahap ini populasi yang memiliki nilai *fitness* yang terbaik akan diteruskan ke proses iterasi selanjutnya, yaitu seleksi, *crossover* dan mutasi. Kode program proses regenerasi dapat dilihat pada Code 4.9 [23].

```

% proses regenerasi
function [ newPopulation2 ] = regeneration(population , newPopulation, Er)
try
    global popSize;
    Elite_no = round(popSize * Er);
    [max_val , indx] = sort([ population.kromosom(:),fitness ] , 'descend');

    % The elites from the previous population

```

```

for k = 1 : Elite_no
newPopulation2.kromosom(k).gen = population.kromosom(indx(k)).gen;
newPopulation2.kromosom(k).koefisien = population.kromosom(indx(k)).koefisien;
newPopulation2.kromosom(k).fitness = population.kromosom(indx(k)).fitness;
end

% The rest from the new population
for k = Elite_no + 1 : popSize
newPopulation2.kromosom(k).gen = newPopulation.kromosom(k).gen;
newPopulation2.kromosom(k).koefisien = newPopulation.kromosom(k).koefisien;
newPopulation2.kromosom(k).fitness = newPopulation.kromosom(k).fitness;
end
catch
msgbox('Proses regenerasi gagal');
end

```

Code 4.9 Kode Program Proses Regenerasi

4.1.2.3 Implementasi Algoritma Genetika Adaptif

Tahapan implementasi yang dilakukan pada algoritma genetika adaptif yaitu menjalankan algoritma PSO sebelum proses *crossover* dimulai. Algoritma PSO bertugas sebagai penghasil *parent* yang nantinya akan di *crossover* untuk menghasilkan individu baru.

a. Proses Inisialisasi *Position* dan *Velocity Particle*

Proses algoritma PSO diawali dengan proses awal yaitu inisialisasi *position* dan *velocity* setiap data saham, *velocity* dan *position* merupakan parameter yang ada pada algoritma PSO yang digunakan oleh partikel atau data saham untuk bergerak ke ruang yang lain mencari solusi yang optimal. Sebelum dilakukan proses inisialisasi, telah dilakukan proses seleksi untuk lima puluh populasi yang memiliki nilai *fitness* tertinggi yang akan digunakan pada tahap pertama proses algoritma PSO. Kode program untuk inisialisasi *position* dan *velocity* dapat dilihat pada Code 4.10.

```

function [parent1, parent2] = selectionPSO(population)
try
% Jumlah populasi
global size;
size = 50;

% Nilai batasan posisi
Boundary=[1,50];

% Initialize Constants
weight=0.5;
constant1=2;
constant2=2;

```

```

% Jumlah seleksi yang diinginkan
nOfSelection=2;

% Jumlah iterasi
iterationMax=5;

% Fungsi nilai rata-rata
fun=@meandata;

% Initialize Particles
Positions=zeros(size,nOfSelection);
for i=1:size
    tempVar = randperm(Boundary(2));
    Positions(i,:) = tempVar(1:nOfSelection);
End

% velocity initialization
Velocity = ones(size,nOfSelection);

% Initialize Global best
iteration = 1;
globalBest.value(iteration) = inf;

```

Code 4.10 Kode Program Proses Inisialisasi Posisi dan Velocity Particle

b. Proses evaluasi nilai *fitness* setiap partikel

Tahapan setelah proses inisialisasi *position* dan *velocity* yaitu evaluasi nilai *fitness*. Setiap partikel atau data saham dihitung nilai *fitness* nya masing-masing. Perhitungan nilai *fitness* akan ditujukan untuk menghasilkan 2 *parent*, sehingga dari 50 populasi akan dihitung nilai rata-rata *fitness* setiap populasi. Perhitungan nilai *fitness* dapat dilihat pada Code 4.11

```

% Optimizing
Particles = struct;

while(1)
    % Calculating Fitness Values
    for i=1:size
        % Inisialisasi posisi partikel
        Particles(iteration).position(i,:) = Positions(i,:);
        % Membuat array berdasarkan nilai fitness pada posisi tertentu
        % Cth: nilai posisi 3 sehingga nilai fitness yang dipakai adalah kromosom nomer 3
        fit = [population.kromosom(Positions(i,1)).fitness
population.kromosom(Positions(i,2)).fitness];
        % Merata-rata nilai fitness untuk dijadikan nilai partikel
        Particles(iteration).value(i) = fun(fit);
    End
End

```

Code 4.11 Kode Program Evaluasi Nilai *Fitness* Partikel

c. Proses *update global best position, update local best position*

Setelah proses perhitungan nilai *fitness*, setiap partikel yang memiliki nilai *fitness*, dievaluasi untuk setiap generasi berdasarkan *local best (pbest)* dan *global best (gbest)* untuk menghasilkan solusi terbaik. Proses *update local best position* digunakan untuk menentukan populasi yang terbaik dalam satu putaran iterasi. Sedangkan proses *update global best* digunakan untuk menentukan nilai *fitness* terbaik dari keseluruhan putaran iterasi. *Update* untuk *global best position* dan *local best position* dapat dilihat pada Code 4.12.

```
% Update Local Best
[Particlesbest.value(iteration),ind] = min(Particles(iteration).value);
Particlesbest.position(iteration,:) = Particles(iteration).position(ind,:);

% Update Global Best
if globalBest.value(iteration) > Particlesbest.value(iteration)
    [globalBest.value(iteration),globalBest.value(iteration+1)] =
deal(Particlesbest.value(iteration));
    globalBest.position(iteration,:) = Positions(ind(1),:);
else
    globalBest.value(iteration+1) = globalBest.value(iteration);
    globalBest.position(iteration,:) = globalBest.position(iteration-1,:);
end

disp('Iteration---->')
disp(iteration)
disp('gbest value of fitness=---->')
disp(globalBest.value(iteration))
```

Code 4.12 Kode Program Proses *Update Global dan Local Best Position*

d. Proses *update position* dan *velocity* setiap partikel

Ketika proses sebelumnya telah didapatkan *local best* dan *global best* dari populasi, setiap bertambahnya iterasi akan terus dilakukan proses memperbarui posisi dan kecepatan, serta nilai *fitness* individu yang baru. *Update* untuk *position* dan *velocity* dapat dilihat pada Code 4.13.

```
% Velocity Update\
Velocity = (weight*Velocity) +
(constant1*rand(1)*(repmat(Particlesbest.position(iteration,:),size,1)-Positions)) + ...
(constant2*rand(1)*(repmat(globalBest.position(iteration,:),size,1)-Positions));
```

```

% Postion Update
Positions = Positions+round(Velocity);

% Boundary Checking for Position
Positions(Positions>Boundary(2)) = round(rand(1)*(Boundary(2)-1))+1;
Positions(Positions<Boundary(1)) = round(rand(1)*(Boundary(2)-1))+1;
for i=1:size
    if length(unique( Positions(i,:) )) ~= nOfSelection
        tempVar = randperm(Boundary(2));
        Positions(i,:) = tempVar(1:nOfSelection);
    end
end

% Loop Breaking is Positions all are same or iteration acheived maximum
% iteration
count=0;
for i=1:size-1;
    if Positions(i,:) == Positions(i+1,:)
        count = count+1;
    end
end
if (count == size-1) || (iteration >= iterationMax)
    fprintf('\n*****Iteration Completed*****\n')
    break
end
iteration = iteration+1;
end

Selection=globalBest.position(iteration,:);

%SelectionValue=globalBest.value(iteration);
disp('Selection---->')
disp(Selection)
parent1 = population.kromosom(Selection(1));
parent2 = population.kromosom(Selection(2));
catch
    msgbox('Proses seleksi PSO gagal');
end

```

Code 4.13 Kode Program Proses *Update Position* dan *Velocity Partikel*

Proses dari Algoritma Genetika Adaptif merupakan proses algoritma genetika yang pada proses *crossover* nya dimodifikasi dengan menambahkan algoritma PSO. Ketika pada proses algoritma PSO berhasil menghasilkan dua *parent*. Kedua *parent* tersebut akan di *crossover* dan dimutasi seperti pada algoritma genetika.

4.1.3 Tahapan Implementasi Menampilkan Hasil

Setelah selesai mengimplementasikan algoritma genetika dan algoritma genetika adaptif, langkah selanjutnya adalah membuat tampilan hasil prediksi saham. Hasil prediksi harga

saham akan ditampilkan pada 2 panel sekaligus, dimana hasilnya dibuat berupa grafik yang ditampilkan pada panel 2 dan berupa angka yang ditampilkan pada panel 3. Kode program menampilkan prediksi harga saham menggunakan algoritma genetika dapat dilihat pada Code 4.14.

```
function buttonLihatHasil_Callback(hObject, eventdata, handles)
global data_saham_aktual;
global uji_prediksi;
global data_prediksi_saham;

data_prediksi_saham = getappdata(0, 'prediksiHasil')
data_saham_aktual = getappdata(0, 'data_saham_aktual');
uji_prediksi = getappdata(0, 'uji_prediksi');
axes(handles.plotPerbandingan);
plot(1:size(data_prediksi_saham,2), data_prediksi_saham, 'b');
grid on
grid minor
legend('Data Prediksi')
xlabel('hari ke-');
ylabel('mm');
handles.textGantiPerbandinganPrediksi.String = {'Perbandingan Data Aktual dan Data Prediksi  
Harga Saham'; 'pada tahun 2016-2019'};
```

Code 4.14 Kode Program Menampilkan Grafik Prediksi Harga Saham

Kode program untuk menampilkan hasil prediksi dalam bentuk angka dapat dilihat pada Code 4.15.

```
%DATA PREDIKSI
function [dataPrediksi] = prediksi_data(size, target, dataUjiPrediksi, BestChrom)
try
global counter;
global errorPredict;
errorPredict=100;
counter = length(target);
fid2 = fopen(dataUjiPrediksi, 'w');
for prepare=1:5
    dataPreparation(prepare) = target(counter);
    counter = counter - 1;
end
for predict=1:size
    for kolom1=1:size
        preX(predict,kolom1) = dataPreparation(kolom1);
        fprintf(fid2, '%d\t', preX(predict,kolom1));
    end
    for kolom2=(size+1):(2*size)
        preX(predict,kolom2) = dataPreparation(kolom2-5);
        fprintf(fid2, '%d\t', preX(predict,kolom2).^2);
    end
    fprintf(fid2, '\n');
    temp_dataPreparation = dataPreparation;
    % regresiData = randi([0,100],1,1);
```

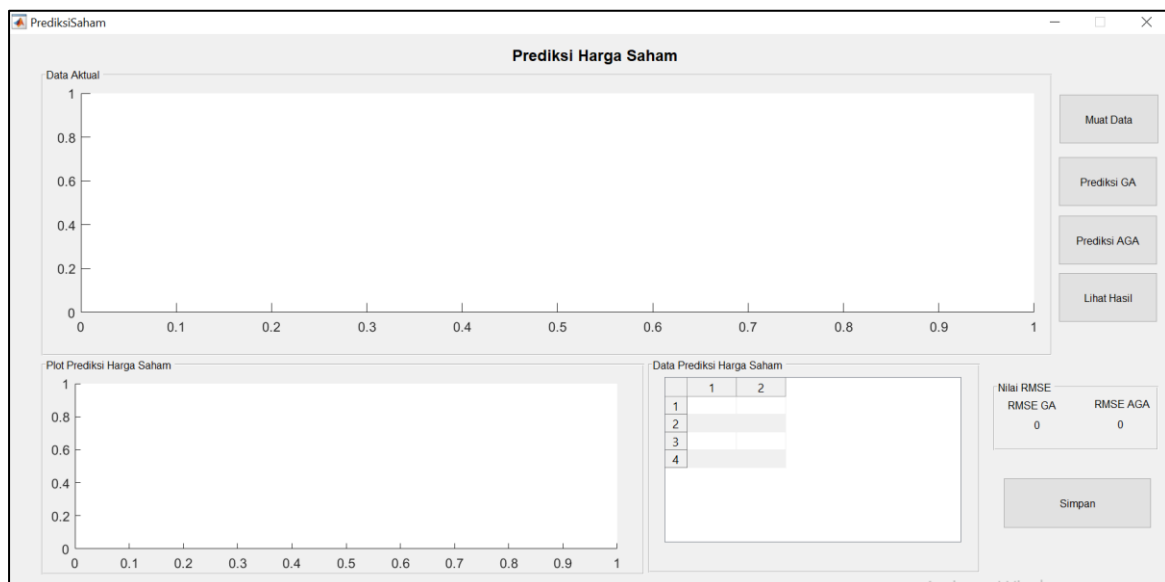
```

[~,regresiData] = calculate_fitness(temp_dataPreparation, BestChrom.kromosom.gen,
BestChrom.kromosom.koefisien);
dataPreparation(1) = regresiData+errorPredict;
for swap=2:5
    dataPreparation(swap) = temp_dataPreparation(swap-1);
end
dataPrediksi = dataPreparation(end:-1:1);
end
fclose(fid2);
catch
    msgbox('Proses prediksi data gagal');
end

```

Code 4.15 Kode Program Menampilkan Hasil Prediksi (Angka)

Hasil penerapan antarmuka menampilkan hasil prediksi harga saham menggunakan algoritma genetika dan algoritma genetika adaptif dapat dilihat pada Gambar 4.4.



Gambar 4.4 Tampilan Hasil Prediksi Harga Saham dengan Algoritma Genetika

4.1.4 Tahapan Implementasi Menyimpan Hasil

Hasil dari prediksi harga saham menggunakan algoritma genetika dan algoritma genetika adaptif dapat disimpan dengan menekan tombol simpan, hasil penyimpanan akan menghasilkan file prediksi harga saham dengan ekstensi .xlsx. Jika ingin menyimpan hasil prediksi harga saham menggunakan algoritma genetika, ketika simulator selesai menampilkan hasil, kita dapat langsung menekan tombol simpan, begitu juga untuk algoritma genetika adaptif. Kode program menampilkan prediksi harga saham menggunakan algoritma genetika dapat dilihat pada Code 4.16

```

function simpanButton_Callback(hObject, eventdata, handles)
try
% hObject   handle to savebutton (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
global data_prediksi_saham;
data_prediksi_saham = getappdata(0, 'prediksiHasil')
[excelName,excelPath] = uiputfile('* .xlsx')
headers = {'Hari 1','Hari 2','Hari 3','Hari 4','Hari 5'};
data = get(handles.uitable1, 'data');
datacell = num2cell(data);
a = [headers;datacell];
x1Range = 'Prediksi Saham'
xlswrite([excelPath excelName],a,x1Range)
catch exception
    msgText(exception)
    %msgbox('Proses prediksi data gagal');
end

```

Code 4.16 Kode Program Menyimpan Hasil Prediksi Harga Saham

4.2 Pengujian

Pengujian yang dilakukan pada simulator yaitu melakukan tiga kali percobaan prediksi harga saham pada masing-masing algoritma. Pengujian dilakukan untuk memastikan hasil prediksi harga saham menggunakan AG dan AGA tetap konsisten. Pada setiap percobaan nilai-nilai parameter AG dan AGA yang digunakan tetap sama. Simulator dijalankan dan setiap hasilnya dicatat pada Excel dan kemudian akan dihitung nilai RMSE dari setiap percobaan.

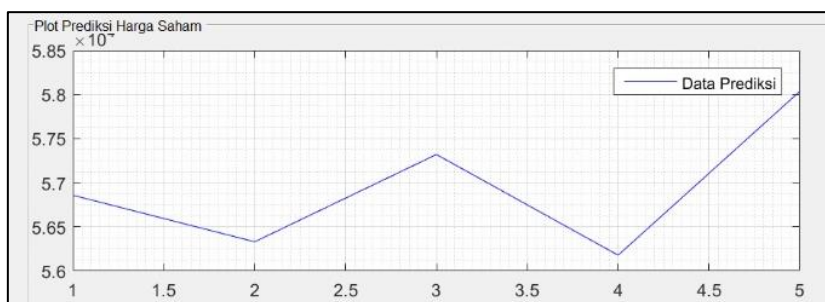
4.2.1 Pengujian AG

Nilai-nilai parameter AG yang digunakan pada setiap percobaan adalah sebagai berikut:

- a. Jumlah Populasi : 100
- b. Jumlah Iterasi : 1000
- c. *Mutation Rate* : 0.013
- d. *Crossover Rate* : 0.5

4.2.1.1 Percobaan I AG

Pada percobaan I, hasil prediksi harga saham dapat dilihat pada Gambar 4.5



Gambar 4.5 Percobaan I AG

Untuk data angka dari percobaan I dapat dilihat pada Tabel 4.3

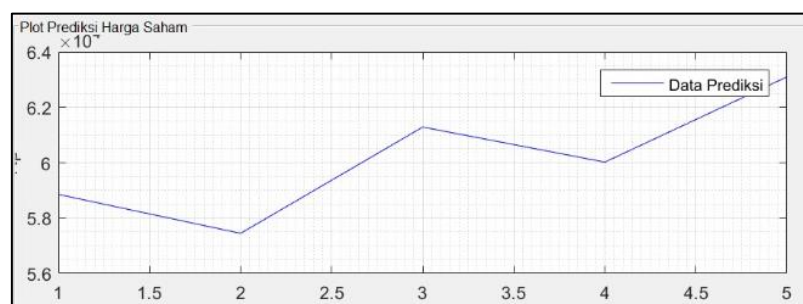
Tabel 4.3 Prediksi Harga Saham Percobaan I AG

Hari ke-	Prediksi Harga Saham
1	50858
2	58331
3	57320
4	56179
5	58036

Untuk percobaan I, harga saham tertinggi yaitu pada hari ke-2 dan harga saham terendah yaitu pada hari ke-1.

4.2.1.2 Percobaan II AG

Pada percobaan II, hasil prediksi harga saham dapat dilihat pada Gambar 4.6



Gambar 4.6 Percobaan II AG

Untuk data angka dari percobaan II dapat dilihat pada Tabel 4.4

Tabel 4.4 Prediksi Harga Saham Percobaan II AG

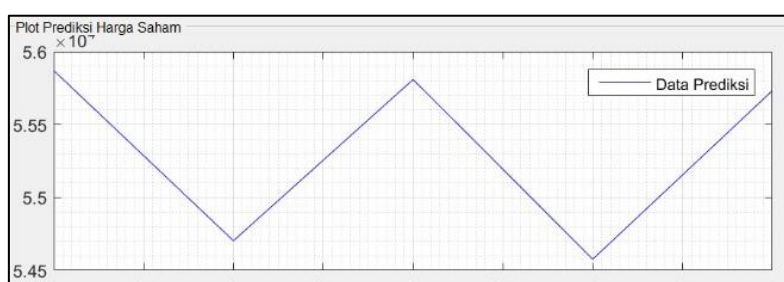
Hari ke-	Prediksi Harga Saham
1	58854
2	57450

3	61285
4	60020
5	63098

Untuk percobaan II, harga saham tertinggi yaitu pada hari ke-5 dan harga saham terendah yaitu pada hari ke-2.

4.2.1.3 Percobaan III AG

Pada percobaan III, hasil prediksi harga saham dapat dilihat pada Gambar 4.7



Gambar 4.7 Percobaan III AG

Untuk data angka dari percobaan III dapat dilihat pada Tabel 4.5

Tabel 4.5 Prediksi Harga Saham Percobaan III AG

Hari ke-	Prediksi Harga Saham
1	55870
2	54703
3	55808
4	54577
5	55730

Untuk percobaan III, harga saham tertinggi yaitu pada hari ke-1 dan harga saham terendah yaitu pada hari ke-4.

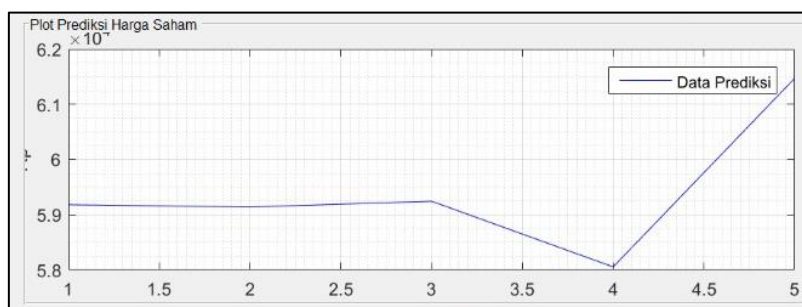
4.2.2 Pengujian AGA

Nilai –nilai parameter AGA yang digunakan pada setiap percobaan adalah sebagai berikut:

- Weight : 0.5
- Iterasi : 50

4.2.2.1 Percobaan I AGA

Pada percobaan I, hasil prediksi harga saham dapat dilihat pada Gambar 4.8



Gambar 4.8 Percobaan I AGA

Untuk data angka dari percobaan I dapat dilihat pada Tabel 4.6

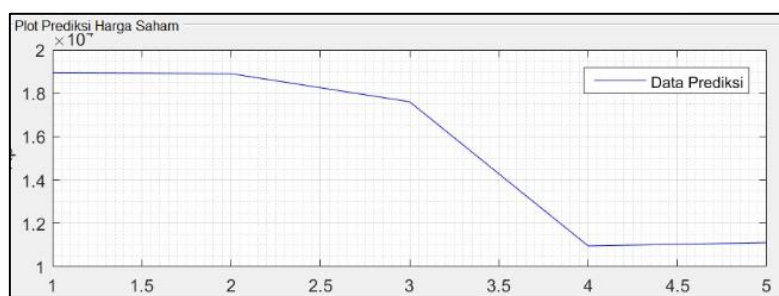
Tabel 4.6 Prediksi Harga Saham Percobaan I AGA

Hari ke-	Prediksi Harga Saham
1	59181
2	59138
3	59244
4	58055
5	61473

Untuk percobaan I, harga saham tertinggi yaitu pada hari ke-5 dan harga saham terendah yaitu pada hari ke-4.

4.2.2.2 Percobaan II AGA

Pada percobaan II, hasil prediksi harga saham dapat dilihat pada Gambar 4.9



Gambar 4.9 Percobaan II AGA

Untuk data angka dari percobaan II dapat dilihat pada Tabel 4.7

Tabel 4.7 Prediksi Harga Saham Percobaan II AGA

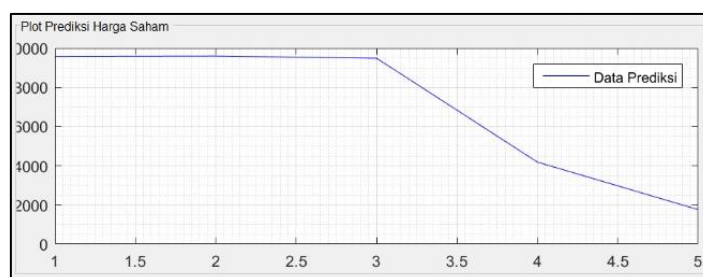
Hari ke-	Prediksi Harga Saham
----------	----------------------

1	18948
2	18902
3	17607
4	10958
5	11112

Untuk percobaan II, harga saham tertinggi yaitu pada hari ke-1 dan harga saham terendah yaitu pada hari ke-4.

4.2.2.3 Percobaan III AGA

Pada percobaan III, hasil prediksi harga saham dapat dilihat pada Gambar 4.10



Gambar 4.10 Percobaan III AGA

Untuk data angka dari percobaan III dapat dilihat pada Tabel 4.8

Tabel 4.8 Prediksi Harga Saham Percobaan III AGA

Hari ke-	Prediksi Harga Saham
1	9578
2	9595
3	9483
4	4187
5	1761

Untuk percobaan III, harga saham tertinggi yaitu pada hari ke-2 dan harga saham terendah yaitu pada hari ke-5.

Nilai RMSE yang dihasilkan oleh algoritma genatika dan algoritma genetika adaptif untuk tiga kali percobaan dapat dilihat pada Tabel 4.9

Tabel 4.9 Nilai RMSE AG dan AGA

Percobaan	RMSE Algoritma Genetika	RMSE Algoritma Genetika Adaptif
I	0.013818	0.00612889
II	0.00761469	0.000583838
III	0.0198804	0.00055964

Berdasarkan nilai RMSE yang diperoleh dari setiap percobaan dapat disimpulkan bahwa pada ketiga percobaan nilai RMSE terbesar diperoleh dari penggunaan algoritma genetika dan nilai RMSE terkecil diperoleh dari penggunaan algoritma genetika adaptif.

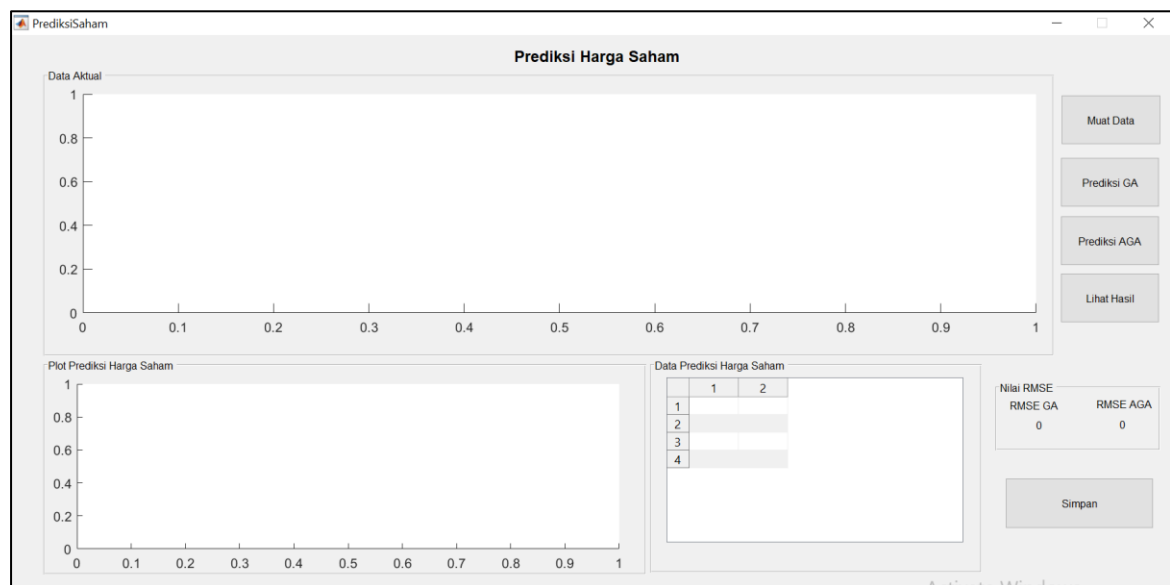
Bab 5

HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan mengenai hasil dan pembahasan dari implementasi yang dilakukan pada penelitian ini. Hasil yang telah diperoleh dari tahap implementasi akan dilampirkan mulai dari memuat data, *forecasting* data, dan menyimpan hasil *forecasting* prediksi harga saham menggunakan algoritma genetika dan algoritma genetika adaptif.

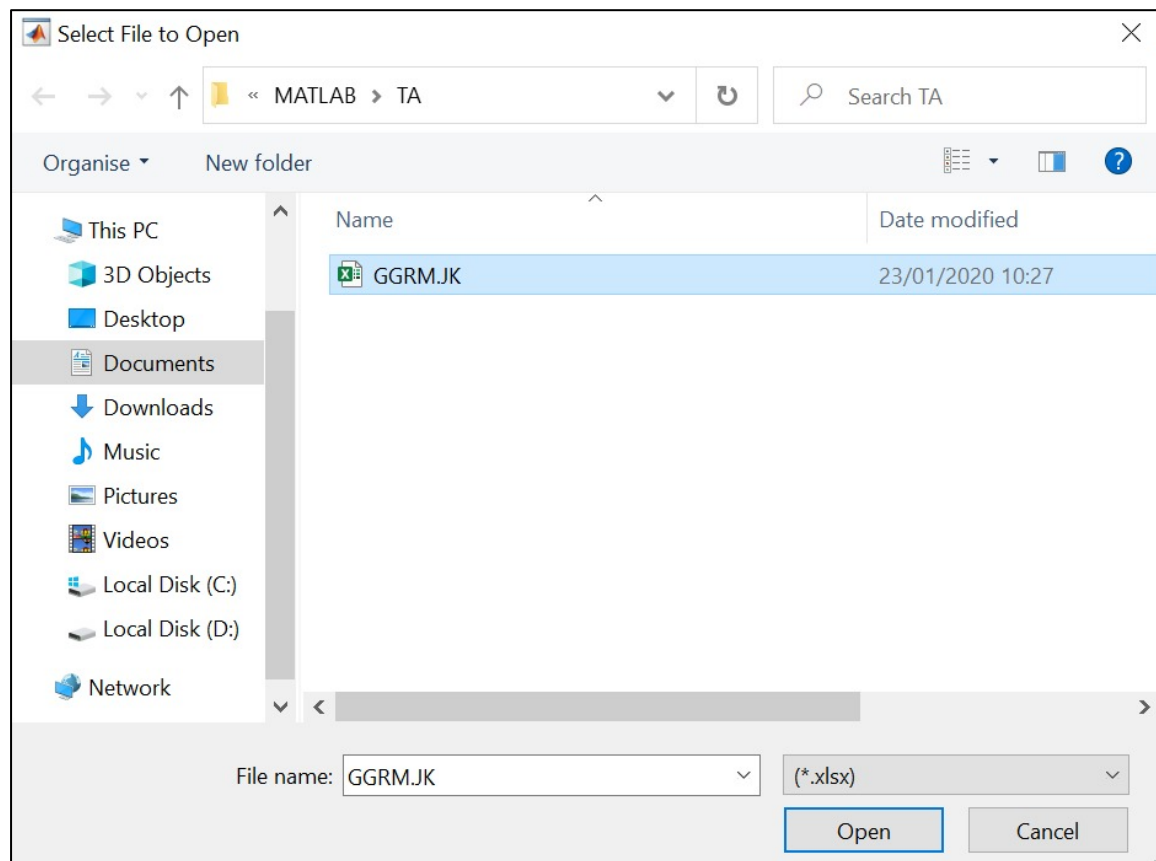
5.1 Hasil Implementasi Memuat Data

Proses memuat data merupakan tahap pertama dalam penggunaan simulator prediksi harga saham yang telah dibangun. Pada tahapan ini diperlukan data harga saham yaitu variabel harga *close* saham di masa lampau. Data yang digunakan sebagai percobaan pada penelitian ini adalah data harga saham dalam kurun waktu 3 tahun terakhir yaitu tahun 2016-2019. Proses memuat data dapat dilakukan dengan menekan *button* Muat Data seperti pada Gambar 5.1.



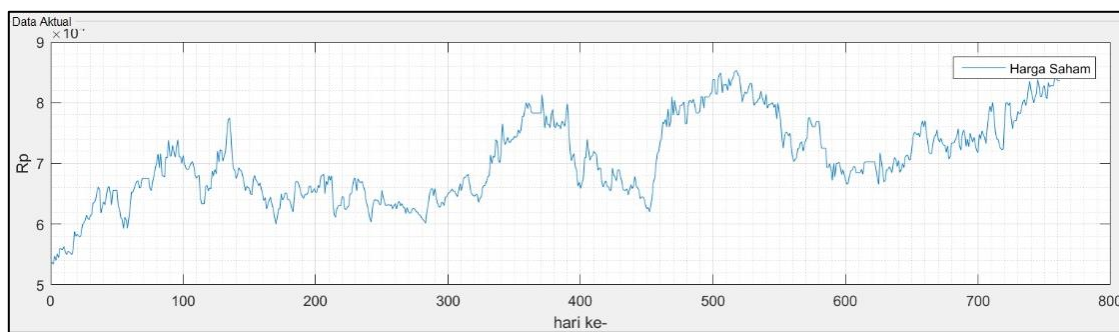
Gambar 5.1 Tampilan Awal simulator

Simulator akan memunculkan *window* untuk pemilihan file data aktual saham yang memiliki ekstensi *.xlsx*. Setelah memilih file data aktual saham pilih *button* open seperti pada Gambar 5.2.



Gambar 5.2 Window Select File Data Aktual Saham

Dalam proses ini, maka simulator secara otomatis akan membagi data menjadi data training dan data test. Pembagian data training dan dataset tersebut dilakukan sesuai dengan analisis dataset yang digunakan pada bab 3.1. File yang telah dipilih diproses dan ditampilkan dalam bentuk plot seperti pada Gambar 5.3.



Gambar 5.3 Plot Data Aktual Saham

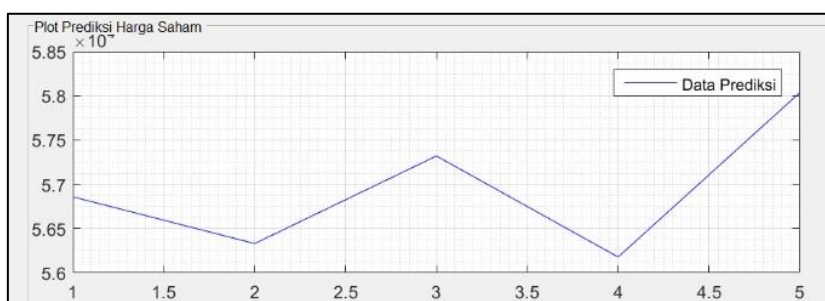
Nilai x atau garis horizontal pada plot data aktual saham menjelaskan waktu pada hari keberapa harga saham tersebut, sedangkan nilai y atau garis vertikal merupakan data harga *close* saham.

5.2 Hasil Melakukan Forecasting Data

Proses melakukan *forecasting* data adalah proses yang menjalankan prediksi harga saham menggunakan AG dan AGA. *Forecasting* data dapat dilakukan jika proses memuat data selesai dilakukan. Untuk menjalankan fungsi *forecasting* data, langkah yang dilakukan yaitu dengan menekan *button* prediksi. Detail hasil implementasi fungsi *forecasting* data dibagi mejadi 2 bagian yaitu hasil prediksi menggunakan algoritma genetika dan menggunakan algoritma genetika adaptif.

5.2.1 Implementasi Prediksi Harga Saham dengan Algoritma Genetika

Hasil yang diperoleh dari prediksi harga saham dengan menggunakan Algoritma Genetika menghasilkan 5 (lima) hari data harga saham. Hasilnya dapat dilihat pada Gambar 5.4 dan Tabel 5.1.



Gambar 5.4 Grafik Hasil Prediksi Harga Saham dengan AG

Data angka hasil prediksi harga saham menggunakan AG dapat dilihat dalam Tabel 5.1.

Tabel 5.1 Prediksi Harga Saham dengan AG

Hari ke-	Prediksi Harga Saham
1	50858
2	58331
3	57320
4	56179
5	58036

Berdasarkan hasil yang diperoleh dari prediksi harga saham nilai prediksi tertinggi adalah pada hari ke-2 yaitu senilai 58331 dan nilai prediksi terendah yaitu pada hari ke-1 yaitu senilai 50858. Selain itu nilai RMSE dari prediksi menggunakan AG juga diperoleh yang ditampilkan pada gambar berikut.



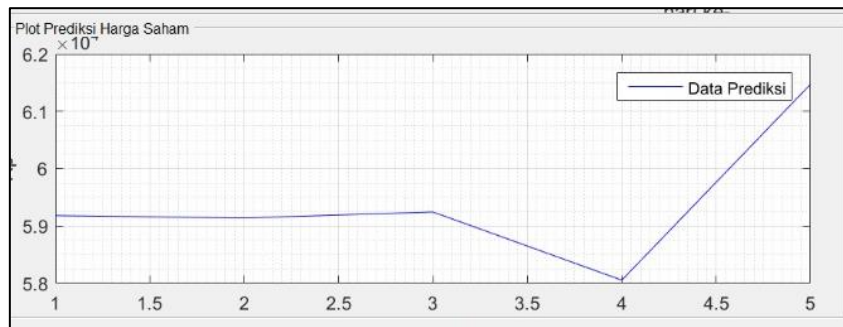
Nilai RMSE	
RMSE GA	0.013818

Gambar 5.5 Nilai RMSE AG

Nilai RMSE menggunakan algoritma genetika pada hasil implementasi algoritma genetika adalah 0.013818 ditampilkan seperti pada Gambar 5.5. Dari perolehan nilai RMSE menggunakan AG ini akan dapat dibandingkan dengan nilai RMSE AGA. Melalui perbandingan nilai RMSE AG dan AGA akan dapat ditentukan algoritma mana yang lebih efektif memecahkan masalah prediksi harga saham.

5.2.2 Implementasi Prediksi Harga Saham dengan Algoritma Genetika Adaptif

Jumlah data prediksi yang dihasilkan menggunakan Algoritma Genetika Adaptif sama dengan jumlah hasil prediksi Algoritma Genetika. Hasilnya dapat dilihat pada Gambar 5.6 dan Tabel 5.2.



Gambar 5.6 Grafik Hasil Prediksi Harga Saham dengan AGA

Data angka hasil prediksi harga saham menggunakan AGA dapat dilihat dalam Tabel 5.2.

Tabel 5.2 Prediksi Harga Saham dengan AGA

Hari ke-	Prediksi Harga Saham
1	59181
2	59138
3	59244
4	58055
5	61473

Berdasarkan hasil yang diperoleh dari prediksi harga saham nilai prediksi tertinggi adalah pada hari ke-5 yaitu senilai 61473 dan nilai prediksi terendah yaitu pada hari ke-4 yaitu senilai 58055. Selain itu nilai RMSE dari prediksi menggunakan AGA juga diperoleh yang ditampilkan pada gambar berikut.

RMSE AGA	0.00612889
----------	------------

Gambar 5.7 Nilai RMSE AGA

Nilai RMSE menggunakan algoritma genetika pada hasil implementasi algoritma genetika adalah 0.00612889 ditampilkan seperti pada Gambar 5.7.

5.3 Menyimpan Hasil *Forecasting*

Hasil dari prediksi harga saham menggunakan kedua algoritma akan disimpan sebagai file prediksi harga saham dengan ekstensi .xlsx. Untuk menyimpan hasil *Forecasting* dapat dilakukan dengan meng-klik *button* Simpan.

Bab 6

KESIMPULAN DAN SARAN

Pada bab ini dijelaskan kesimpulan dari hasil penelitian tugas akhir yang dilakukan beserta saran penulis pada penelitian di masa yang akan datang.

6.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa Algoritma yang paling baik dalam memprediksi harga Saham PT.Gudang Garam menggunakan Algoritma Genetika dan Algoritma Genetika Adaptif dilihat dari nilai RMSE nya yaitu Algoritma Genetika Adaptif. Nilai RMSE dari Algoritma Genetika Adaptif menunjukkan nilai yang lebih kecil dibandingkan nilai RMSE Algoritma Genetika dalam tiap pengujian yang dilakukan. Hasil prediksi harga saham menggunakan Algoritma Genetika menghasilkan RMSE dengan nilai 0.013818, sedangkan untuk prediksi harga saham menggunakan Algoritma Genetika Adaptif menghasilkan nilai RMSE dengan nilai 0.00612889.

6.2 Saran

Berdasarkan hasil yang telah diperoleh saran yang dapat dikemukakan adalah:

1. Untuk implementasi menggunakan AGA, disarankan melakukan modifikasi dengan algoritma lainnya selain algoritma PSO. Operator yang dimodifikasi juga bisa dilakukan pada proses algoritma genetika yang lain, seperti pada proses mutasi ataupun juga modifikasi di kedua proses *crossover* dan mutasi.
2. Penelitian berikutnya disarankan untuk menggunakan lebih dari satu atribut data saham lainnya seperti atribut *open*, *high*, *low*, *volume* dan *adj close* agar hasil penelitian lebih luas lagi.

Daftar Pustaka dan Rujukan

- [1] A. Novita, “Prediksi Pergerakan Harga Saham Pada Bank Terbesar Di Indonesia Dengan Metode Backpropagation Neural Network,” *Jutisi*, vol. 05, no. 01, pp. 965–972, 2016.
- [2] S. Sari, “Faktor-Faktor Yang Mempengaruhi Harga Saham,” *Univ. Muhammadiyah Surakarta*, pp. 28–43, 2015.
- [3] S. Sable, A. Porwal, and U. Singh, “Stock price prediction using genetic algorithms and evolution strategies,” *Proc. Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2017*, vol. 2017-Janua, pp. 549–553, 2017.
- [4] M. J. Mahmoodabadi and A. R. Nemati, “A novel adaptive genetic algorithm for global optimization of mathematical test functions and real-world problems,” *Eng. Sci. Technol. an Int. J.*, vol. 19, no. 4, pp. 2002–2021, 2016.
- [5] Cnnbcindonesia, “Ini Deretan Harga Saham ‘Tertinggi’ di BEI, Berat Buat Ritel?,” 2019.
- [6] M. Melanie, *An Introduction to Genetic Algorithms*. London, England: A Bradford Book The MIT Press, 1999.
- [7] “Yahoo Finance,” 2019. [Online]. Available: <https://finance.yahoo.com/>.
- [8] C. S and A. S. Hadi, *Regression Analysis By Example*, Fourth Edi., vol. 53, no. 9. Cairo, Egypt: SAMPRIT CHATTEFUEE, 2013.
- [9] C. Sri and W. Widayati, “Komparasi Beberapa Metode Estimasi Kesalahan Pengukuran,” *J. Penelit. dan Eval. Pendidik.*, vol. 13, no. 2, pp. 182–197, 2013.
- [10] J. Carr, “An Introduction to Genetic Algorithms,” pp. 1–40, 2014.
- [11] U. Aickelin and D. Dasgupta, “Search Methodologies : Introductory Tutorials in Optimization and Decision Support Techniques Edmund K . Burke (Editor), Graham Kendall (Editor) ARTIFICIAL IMMUNE SYSTEMS,” pp. 1–29.
- [12] M. Learning and K. A. Publishers, “GUEST EDITORIAL Genetic Algorithms and Machine Learning,” pp. 95–99, 1988.
- [13] D. R. Anamisa and A. Djunaidy, “Dan Algoritma Koloni Semut,” *JUTI*, vol. 12, no. 1, pp. 15–20, 2014.
- [14] J. Blondin, “Particle swarm optimization: A tutorial,” ... *Site Http//Cs. Armstrong. Edu/Saad/Csci8100/Pso Tutor*. ..., pp. 1–5, 2009.
- [15] K. Zielinski and R. Laur, “Stopping criteria for differential evolution in constrained single-objective optimization,” *Stud. Comput. Intell.*, vol. 143, no. 1, pp. 111–138,

- 2008.
- [16] A. Rahmi, W. F. Mahmudy, B. D. Setiawan, U. Brawijaya, and P. H. Saham, “PREDIKSI HARGA SAHAM BERDASARKAN DATA HISTORIS MENGGUNAKAN MODEL REGRESI,” vol. 5, no. 12, pp. 1–9, 2015.
 - [17] I. Christinawati, Y. Ramadaniel, C. Pane, and R. N. E. R. Silitonga, “Penjadwalan Kuliah Menggunakan Penggabungan Algoritma Genetika dan Particle Swarm Optimization (Studi Kasus Institut Teknologi Del).”
 - [18] R. Ketaren, D. Sinaga, W. Siringoringo, “PERAMALAN OPTIMAL UANG TUNAI ATM MENGGUNAKAN GRADIENT BOOSTING DECISION TREE DAN ALGORITMA GENETIKA ADAPTIF,” 2019.
 - [19] Q. Ye and L. Wei, “The Prediction of Stock Price Based on Improved Wavelet Neural Network,” *Open J. Appl. Sci.*, vol. 05, no. 04, pp. 115–120, 2015.
 - [20] Z. Michalewicz, *Genetic Algorithms + Data Structure*. Verlag: Springer, 1996.
 - [21] A. Andhyka, “Penerapan Algoritma Genetika pada Permasalahan Matematika,” vol. 04, no. 01, pp. 23–29, 2018.
 - [22] I. Permadi, F. Mipa, and U. Gadjah, “Penerapan Algoritma Genetika untuk Optimasi Penjadwalan Tebangan Hutan (Applying of Genetic Algorithm for Scheduling Optimization Cuts Away Forest),” vol. I, pp. 19–27, 2010.
 - [23] P. S. Mirjalili, *Introduction to Genetic Algorithms: Theory and Applications*. 2017, p. udemy.