

# APLIKASI PREDIKSI PERSEDIAAN TABUNG GAS MENGGUNAKAN ALGORITMA GENETIKA DAN REGRESI LINIER BERGANDA STUDI KASUS : PT TRIMITA UTAMA

Silviana Sulistiani<sup>1</sup>, Wahyu Pramusinto<sup>2</sup>

<sup>1, 2)</sup> Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur  
Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan, Jakarta Selatan 12260  
Telp. (021) 5853753, Fax. (021) 5853752  
<sup>1</sup>silviana536@gmail.com, <sup>2</sup>wahyu.pramusinto@budiluhur.ac.id

## ABSTRAK

Gas merupakan bahan bakar yang saat ini digunakan untuk memasak oleh ibu rumah tangga. Kebutuhan tabung gas di masyarakat tiap bulan berbeda-beda. PT Trimita Utama sebagai pihak perusahaan distributor tabung gas sulit memperkirakan berapa kebutuhan tabung gas yang harus disediakan tiap bulannya. Persediaan yang terlalu banyak menyebabkan terjadi penumpukan di gudang dan bahaya kebocoran. Persediaan yang terlalu sedikit akan menimbulkan kerugian. Oleh karena itu dibutuhkan suatu aplikasi yang dapat memprediksikan seberapa banyak jumlah tabung gas yang akan terjual di bulan selanjutnya, sehingga pihak perusahaan dapat memperkirakan seberapa banyak jumlah tabung gas yang harus disediakan. Sistem prediksi sendiri hingga saat ini telah banyak dikembangkan menggunakan berbagai macam metode yang ada. Pada penelitian kali ini, aplikasi akan dikembangkan dengan metode Regresi Linier Berganda dengan menggunakan proses Algoritma Genetika. Pada kasus ini algoritma genetika dapat digunakan untuk memprediksikan persediaan tabung gas mengacu kepada data penjualan dari 12 bulan sebelumnya. Diharapkan aplikasi dapat memprediksikan secara akurat atau mendekati dengan data aktual, sehingga pihak perusahaan dapat mengetahui berapa banyak persediaan tabung gas yang harus disiapkan untuk bulan berikutnya. Hasil pengujian aplikasi prediksi menggunakan algoritma genetika ini menghasilkan rata-rata persentase error sebesar 4.1% yang berarti aplikasi prediksi persediaan tabung gas ini mempunyai tingkat keakuratan sebesar 95.99%.

**Kata Kunci :** Algoritma Genetika, Roulette Wheel Selection, Whole Arithmetic Crossover, Random Mutation, Regresi Linier Berganda.

## I. PENDAHULUAN

Persediaan Tabung gas merupakan suatu hal yang penting khususnya di perusahaan yang bergerak di bidang penjual tabung gas seperti distributor. Karena, dengan adanya persediaan tabung gas yang optimal dapat membantu untuk mempertahankan aktivitas operasi perusahaan atau menjamin kelancaran arus distributor ke penjual kecil (ecer), serta memberikan jaminan tetap tersedianya tabung gas tersebut.

Namun pada praktiknya, merencanakan penyediaan tabung gas hingga disetorkan ke penjual atau ke agen-agen lainnya bukan perkara yang mudah. Jika persediaan terlalu banyak, tabung gas akan menumpuk dan bisa saja terjadi bocor pada salah tabung tersebut yang tidak diketahui oleh pihak distributor itu sendiri. Di lain sisi, jika persediaan gas terlalu sedikit juga akan menimbulkan kerugian bagi perusahaan distributor maupun pembeli (agen-agen) baik dari sisi kesempatan memperoleh keuntungan maupun sisi pendapatan perusahaan dalam membeli tabung gas selanjutnya. Untuk menyelesaikan masalah tersebut, akan digunakan metode analisis regresi linier berganda dengan proses algoritma genetika yang menggunakan pengkodean bilangan *real*,

*roulette wheel selection, whole arithmetic crossover, random mutation, dan update generasi.*

## II. LANDASAN TEORI

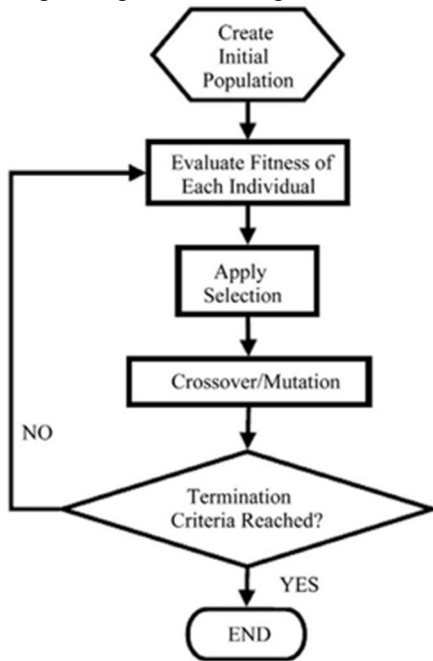
### A. Pengertian Algoritma Genetika

Algoritma Genetika adalah teknik optimasi dan pencarian berdasarkan prinsip-prinsip genetika dan seleksi alam [1]. Algoritma ini didasarkan pada proses genetik yang ada dalam makhluk hidup, yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam.

Secara umum, algoritma genetika memiliki lima komponen dasar menurut [2] yaitu :

- Representasi genetic dan dari solusi-solusi masalah.
- Cara membentuk populasi awal dari solusi-solusi.
- Fungsi evaluasi yang *me-rate (rating)* solusi-solusi berdasarkan *fitness* mereka.
- Operator-operator genetic yang mengubah komposisi genetic dari *offspring* selama reproduksi.
- Nilai-nilai untuk parameter algoritma genetika.

Secara umum struktur dari suatu algoritma genetika dapat dijelaskan dengan diagram alur sebagai berikut :

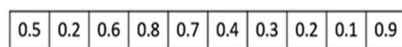


Gambar 1: Diagram alur algoritma genetika [3]

Algoritma Genetik berangkat dari himpunan solusi yang dihasilkan secara acak yang disebut populasi. Sedangkan setiap individu dalam populasi disebut kromosom yang merupakan representasi dari solusi dan masing-masing dievaluasi tingkat ketanggguhannya (*fitness*) oleh fungsi yang telah ditentukan. Melalui proses seleksi alam atas operator genetik, gen-gen dari dua kromosom (disebut *parent*) diharapkan akan menghasilkan kromosom baru dengan tingkat *fitness* yang lebih tinggi sebagai generasi baru atau keturunan (*offspring*) berikutnya. Kromosom-kromosom tersebut akan mengalami iterasi yang disebut generasi (*generation*). Pada setiap generasi, kromosom dievaluasi berdasarkan nilai fungsi *fitness* [4].

1.1 Pengkodean Bilangan Real

Pengkodean adalah suatu teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom sebagai suatu kunci pokok persoalan ketika menggunakan algoritma genetika. Ada beberapa jenis pengkodean yang dapat digunakan dalam algoritma genetika, salah satunya yaitu pengkodean bilangan *real* (*real number encoding*). Masalah optimasi fungsi dan optimasi kendala lebih cepat jika diselesaikan dengan pengkodean bilangan real karena struktur topologi ruang genotif untuk pengkodean bilangan real identik dengan ruang fenotifnya, sehingga mudah membentuk operator genetik yang efektif dengan cara memakai teknik yang dapat digunakan yang berasal dari metode konvensional [5].



Gambar 2 : Pengkodean Bilangan Real

1.2 Regenerasi Linier Berganda

Regenerasi linier berganda adalah analisis regresi yang menjelaskan hubungan antara peubah respon (variabel dependen) dengan faktor-faktor yang mempengaruhi lebih dari satu prediktor (variabel independen). Regresi linier berganda hampir sama dengan regresi linier sederhana, hanya saja pada regresi linier berganda variabel bebasnya lebih dari satu variabel penduga [6]. Tujuan analisis regresi linier berganda adalah untuk mengukur intensitas hubungan antara dua variabel atau lebih. Secara umum model regresi linier berganda untuk populasi dapat dilihat pada persamaan 1.

$$Y(t) = \theta_0 + \theta_1 y(t-1) + \theta_2 y(t-2) + \theta_3 y(t-3) + \dots + \theta_n y(t-n) \quad \text{(persamaan 1)}$$

Keterangan :

Y(t) : Jumlah prediksi penjualan pada periode t.

(t) : Jumlah penjualan actual pada periode t.

θ0, θ1, θ2, θ3, . . . θn : Bilangan yang dibangkitkan secara acak dari 0-1.

Setelah mendapatkan nilai dari suatu prediksi metode yang digunakan untuk mengevaluasi nilai *fitness* dari suatu individu adalah dengan menggunakan MSE. *Mean squared error* (MSE) adalah metode lain untuk mengevaluasi metode peramalan. Masing-masing kesalahan atau sisa dikuadratkan. Kemudian dijumlahkan dan ditambahkan dengan jumlah observasi. Proses dari MSE itu sendiri dapat dilihat pada persamaan 2.

$$MSE = \frac{\sum(X_i - F_i)^2}{n} \dots\dots\dots \text{(persamaan 2)}$$

Keterangan :

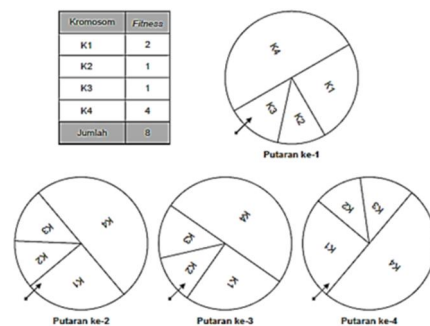
Xi : Hasil penjualan actual pada periode i.

Fi : Hasil prediksi penjualan pada periode i.

n : banyaknya periode waktu (n = 12).

1.3 Seleksi Roda Roulette (*Roulette Wheel Selection*)

Metode seleksi roda *roulette* ini merupakan metode yang paling sederhana serta paling banyak digunakan, Pada metode ini, orangtua dipilih berdasarkan nilai *fitness*nya, semakin baik nilai *fitness*nya maka semakin besar kemungkinannya untuk terpilih. Diandaikan semua kromosom diletakkan pada sebuah roda *roulette*, besarnya kemungkinan bagi setiap kromosom adalah tergantung dari nilai *fitness*nya. Seleksi ini bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang memiliki *fitness* tinggi untuk melakukan reproduksi.



Gambar 3 : Seleksi Roda Roulette

### 1.4 Whole Arithmetic Crossover

Pada *whole arithmetic crossover*, gen pada kromosom *offspring* diperoleh dari hasil operasi aritmatika gen pada kromosom *parent*, di mana proses aritmatika yang dilakukan menurut [7] adalah seperti pada persamaan 3.

$$child = \alpha \cdot \bar{x} + (1 - \alpha) \cdot \bar{y} \dots\dots\dots(\text{persamaan 3})$$

Keterangan :  
 $\alpha$  = Variabel pengali yang nilainya berkisar dari 0-1

Ilustrasi dari proses *whole arithmetic crossover* dengan  $\alpha = 0.5$  adalah :



Gambar 4 : Proses Whole Arithmetic Crossover

### 1.5 Mutasi

Proses mutasi ini dilakukan setelah proses rekombinasi dengan cara memilih kromosom yang akan dimutasi secara acak, dan kemudian menentukan titik mutasi pada kromosom tersebut secara acak pula. Banyaknya kromosom yang akan mengalami mutasi dihitung berdasarkan probabilitas mutasi yang telah ditentukan terlebih dahulu. Apabila probabilitas mutasi adalah 100% maka semua kromosom yang ada pada populasi tersebut akan mengalami mutasi. Sebaliknya, jika probabilitas mutasi yang digunakan adalah 0% maka tidak ada kromosom yang mengalami mutasi pada populasi tersebut.

### 1.6 Update Generasi

Setelah seleksi proses, *crossover* dan mutasi yang dilakukan terhadap suatu populasi, proses berikutnya adalah update generasi untuk menentukan genome yang masih bertahan (*survive*) pada populasi tersebut, dan membentuk generasi selanjutnya. strategi bantuan yang baik untuk digunakan dengan skema *continuous update* adalah *elitism*, yang berarti mempertahankan kromosom kromosom baik dalam populasi lama dengan menyertakannya lagi pada populasi baru. Sejumlah kromosom terbaik pada populasi lama dapat hilang karena proses perkawinan silang dan mutasi. Secara teori, *elitism* dapat meningkatkan kemampuan dari algoritma genetika karena mempertahankan kromosom yang baik dari populasi lama, tetapi kadangkala hal itu justru dapat menyebabkan konvergensi prematur karena nilai *fitness* terjebak pada *optimum* lokal [4].

### 1.7 Parameter Control Dalam Algoritma Genetika

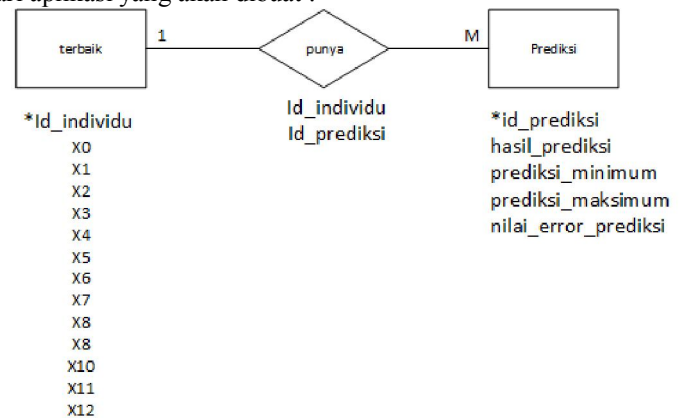
Parameter kontrol merupakan sebuah parameter yang mengontrol keseluruhan proses algoritma genetika mulai dari ukuran populasi, presentase probabilitas *crossover*, dan probabilitas mutasi. Nilai parameter ini ditentukan berdasarkan permasalahan yang ada, dimana nilai dari paramater kontrol algoritma genetika berbeda-beda satu sama lain, sehingga perlu pengujian lebih lanjut untuk mengetahui nilai terbaik dari parameter kontrol sesuai dengan permasalahan yang dihadapi.

Bila *fitness* dari individu terbaik dipantau pada setiap generasi, maka usulannya adalah : ukuran populasi 80; probabilitas *crossover* 0.45; dan probabilitas mutasi 0.01. Ukuran populasi sebaiknya tidak lebih kecil dari 30, untuk sembarang jenis permasalahan.

## B. Rancangan Sistem Dan Aplikasi

### 1. ERD (Entity Relationship Diagram)

*Entity Relationship Diagram* (ERD) ini berisi komponen-komponen himpunan entitas dan himpunan relasi. Masing-masing dilengkapi dengan atribut-atribut yang mewakili seluruh data. Pada gambar 5 adalah gambar rancangan ERD dari aplikasi yang akan dibuat :



Gambar 5 : Rancangan Entity Relationship Diagram

### 2. Spesifikasi Basis Data

Berikut ini adalah struktur tabel yang digunakan dalam pembuatan *database* untuk aplikasi prediksi persediaan tabung gas ini :

#### 2.1 Tabel Data Penjualan

Nama tabel : data\_penjualan  
 Primary key : bulan & tahun

Tabel 1 : Spesifikasi Tabel data\_penjual

Field	Tipe	Length	Ket.
bulan	char	2	bulan dari data penjual
tahun	char	4	tahun dari data penjual
jumlah	int	11	jumlah penjual

#### 2.2 Tabel Hasil Prediksi

Nama tabel : hp  
 Primary key : id\_prediksi

Tabel 2 : Spesifikasi Tabel Hasil Prediksi

Field	Tipe	Length	Keterangan
id_prediksi	int	11	Id dari prediksi
hasil_prediksi	int	11	Hasil dari prediksi
nilai_error	int	11	Nilai error dari prediksi
id_individu	int	11	Id dari individu terbaik

### 2.3 Tabel Hasil Terbaik

Nama tabel : terbaik  
 Primary key : id\_individu  
 Foreign key : -

Tabel 3: Spesifikasi Tabel terbaik

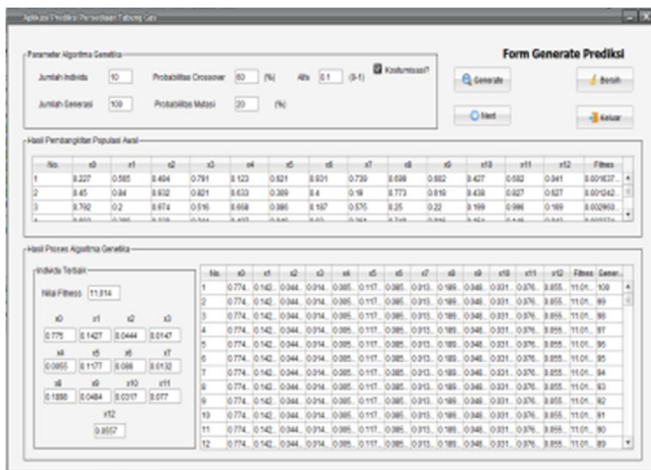
Field	Type	Length	Ket.
id_individu	Int	11	Id dari individu terbaik
X0	double	22,21	Nilai dari theta ke-0
X1	double	22,21	Nilai dari theta ke-1
X2	double	22,21	Nilai dari theta ke-2
X3	double	22,21	Nilai dari theta ke-3
X4	double	22,21	Nilai dari theta ke-4
X5	double	22,21	Nilai dari theta ke-5
X6	double	22,21	Nilai dari theta ke-6
X7	double	22,21	Nilai dari theta ke-7
X8	double	22,21	Nilai dari theta ke-8
X9	double	22,21	Nilai dari theta ke-9
X10	double	22,21	Nilai dari theta ke-10
X11	double	22,21	Nilai dari theta ke-11
X12	double	22,21	Nilai dari theta ke-12

## III. HASIL DAN PEMBAHASAN

### 1. Hasil Pengujian

#### 1.1. Tampilan Layar Form Generate Prediksi

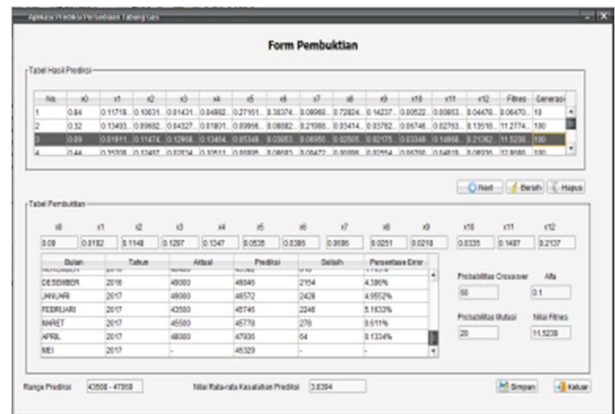
Pada form ini dapat melakukan generate prediksi yang hasilnya akan otomatis ke dalam database.



Gambar 6 : Tampilan Layar Form Menu Generate Prediksi

#### 1.2. Tampilan Layar Form Pembuktian

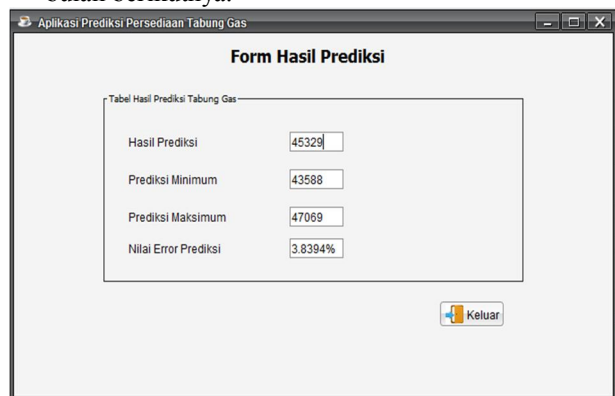
Pada form ini user memilih hasil Prediksi yang akan digunakan yang kemudian user dapat melihat akurasi nilai error rate, serta range dari hasil yang di prediksi. Selain itu user dapat memilih hasil prediksi yang terbaik diproses kedalam form hasil prediksi.



Gambar 7 : Tampilan Layar Form Pembuktian

#### 1.3. Tampilan Layar Form Hasil Prediksi

Pada form ini user dapat melihat hasil prediksi pada bulan berikutnya.

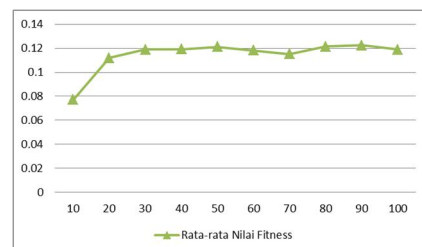


Gambar 8 : Tampilan Layar Form Hasil Prediksi

### 2. Hasil Pengujian

#### 2.1 Pengujian Dan Analisis Ukuran Jumlah Individu

Pada pengujian kali ini jumlah generasi yang digunakan adalah 50 dengan banyak populasi mulai dari 10 populasi sampai dengan 100 populasi. Nilai probabilitas crossover yang digunakan adalah 50%, nilai alfa 0.1, serta probabilitas mutasi sebesar 5%.

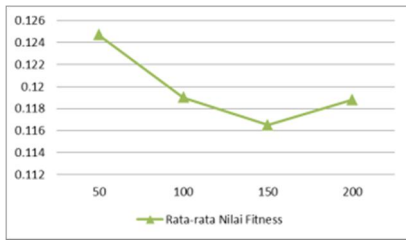


Gambar 9 : Grafik Pengujian Berdasarkan Jumlah Individu

Pada percobaan kali ini dapat disimpulkan jika dilihat dari nilai fitness-nya maka ukuran populasi 90 merupakan nilai terbaik, sementara jika dilihat dari peningkatan nilai fitness yang signifikan, maka ukuran populasi 40 merupakan nilai yang terbaik, diikuti oleh 50, dan 80.

### 2.2 Pengujian Dan Analisi Ukuran Generasi

Pada pengujian kali ini jumlah individu yang digunakan adalah 90, nilai probabilitas *crossover* sebesar 50%, nilai alfa 0.1, serta probabilitas mutasi sebesar 5%.

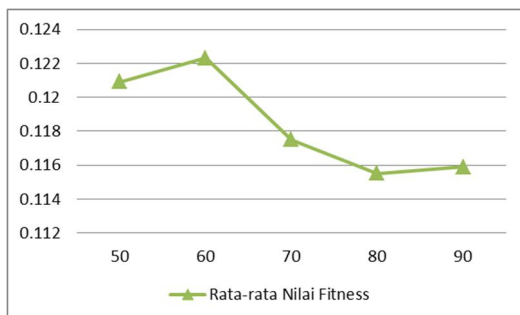


Gambar 10 : Grafik Pengujian Berdasarkan Jumlah Generasi

Untuk Pengujian selanjutnya generasi 50 dan 100 akan digunakan dan dianalisis, dimana ukuran generasi yang dihasilkan akan menghasilkan nilai kovagen makan ukuran generasi 100 yang akan digunakan untuk memperoleh hasil nilai terbaik.

### 2.3 Pengujian Dan Analisis Nilai Probabilitas Crossover

Pada pengujian ini diberikan jumlah individu sebanyak 90, nilai alfa 0.1, serta probabilitas mutasi sebesar 5%.

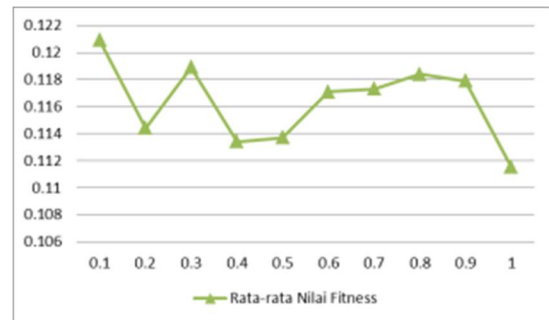


Gambar 11 : Grafik Pengujian Berdasarkan Nilai Probabilitas Crossover

Dapat diketahui bahwa nilai probabilitas 50, 60, dan 70 mengalami peningkatan dan penurunan yang signifikan, sementara nilai probabilitas *crossover* lainnya tidak mengalami peningkatan maupun penurunan yang signifikan, dengan begitu nilai probabilitas *crossover* 50, 60, atau 70 akan digunakan pada pengujian selanjutnya.

### 2.4 Pengujian Dan Analisis Nilai Alfa

Pada pengujian kali ini diberikan jumlah individu 90, nilai probabilitas *crossover* sebesar 60%, nilai alfa dari 0.1 sampai dengan 1 dengan kelipatan 0.1, serta probabilitas mutasi sebesar 5%.

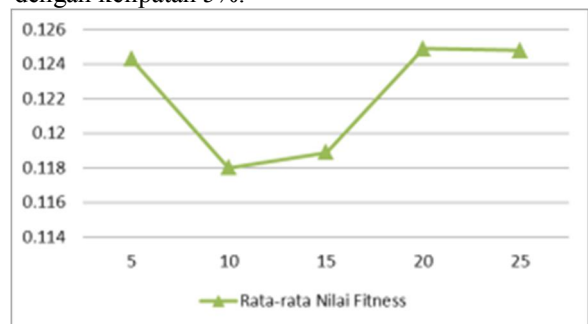


Gambar 12 : grafik pengujian berdasarkan nilai alfa

Bahawa rata-rata nilai *fitness* alfa 1, dan rata-rata *fitness* tertinggi ada pada nilai alfa 0.1, diikuti oleh 0.3, lalu 0.8.

### 2.5 Pengujian Dan Analisis Nilai Probabilitas Mutasi

Pada pengujian kali ini jumlah individu sebanyak 90, nilai probabilitas *crossover* sebesar 60%, nilai alfa 0.1, serta probabilitas mutasi sebesar 5% sampai dengan 20% dengan kelipatan 5%.



Gambar 13 : grafik pengujian berdasarkan nilai mutasi

Pada gambar di atas nilai *fitness* terendah ada pada nilai probabilitas mutasi sebesar 10%, dan rata-rata *fitness* tertinggi ada pada nilai probabilitas mutasi sebesar 5%, dengan begitu nilai mutasi 20% atau 25% akan digunakan.

### 2.6 Hasil Pengujian Ketiga

Melihat hasil dari beberapa pengujian sebelumnya, maka digunakan parameter kontrol sesuai dengan pengujian pertama dimana pada pengujian terakhir ini akan dilakukan sebanyak 5 kali demi melihat *stability*, *consistency* dan *reability* dari hasil prediksi yang dihasilkan. Hasil dari pengujian ketiga dapat dilihat pada tabel 4 di bawah ini :

Tabel 4 : Hasil Pengujian

Pengujian	Range Prediksi	Error Average	Running Time (second)
1	43336 - 47075	4.13%	266 detik
2	43914 - 47695	4.12%	258 detik
3	43467 - 47660	4.60%	254 detik
4	43634 - 47515	4.25%	304 detik
5	43215 - 46652	3.82%	343 detik
6	43588 - 47069	3.83%	217 detik
7	44894 - 48821	4.19%	247 detik
8	43455 - 46824	3.73%	216 detik
9	42736 - 46407	4.11%	312 detik
10	43440 - 47351	4.30%	222 detik
Rata-rata	43568-47307	4.108%	263.9

Dimana nilai *error* terendah sebesar 3.73% dan nilai *error* tertinggi sebesar 4.60% dengan rata-rata nilai *error* sebanyak 4.108% dan rata-rata running time selama 263.9 detik.

#### IV. KESIMPULAN

Berdasarkan analisis yang dilakukan sebelumnya dimulai dari tahap awal hingga tahap akhir serta tahap pengujian yang sudah dilakukan, maka dapat ditarik sebuah kesimpulan diantaranya :

- a. Algoritma genetika dapat digunakan untuk mencari nilai prediksi pada kasus persediaan tabung gas dengan memanfaatkan data penjualan dari beberapa bulan sebelumnya.
- b. Waktu komputasi yang diperlukan berbanding lurus dengan banyaknya iterasi yang dilakukan pada proses algoritma genetika.

- c. Hasil solusi yang dihasilkan akan berbeda-beda setiap kali dijalankan, namun dengan proses iterasi yang banyak dapat menghasilkan satu solusi terbaik dimana hasil yang didapat akan konvergen, data pengujian menunjukkan bahwa algoritma genetika dapat menghasilkan nilai prediksi dengan rata-rata *error rate* sebesar 4.60% atau dengan kata lain algoritma genetika mempunyai akurasi sebesar 95.40%.

#### DAFTAR PUSTAKA

- [1] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, Second Edi. New Jersey: John Wiley & Sons, Inc., 2004.
- [2] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs," *Computational Statistics & Data Analysis*, vol. 24, no. 3. Springer, pp. 372–373, 1996.
- [3] G. C. Pereira, M. M. F. de Oliveira, and N. F. F. Ebecken, "Genetic Optimization of Artificial Neural Networks to Forecast Virioplankton Abundance from Cytometric Data," *J. Intell. Learn. Syst. Appl.*, vol. 5, no. 1, pp. 57–66, 2013.
- [4] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*. John Wiley & Sons, Inc., 2000.
- [5] F. Budi, "Cara Pengkodean dalam Algoritma Genetika," 2013. [Online]. Available: <http://infokitabersama123.blogspot.co.id/2013/12/cara-pengkodean-dalam-algoritma-genetika.html>. [Accessed: 20-Jan-2016].
- [6] N. Samosir, P. Siagian, and P. Bangun, "Analisa Metode Backward Dan Metode Forward Untuk Menentukan Persamaan Regresi Linier Berganda (Studi Kasus : Jumlah Kecelakaan Lalu Lintas Di Kotamadya Medan)," *Saintia Mat.*, vol. 2, no. 4, pp. 345–360, 2014.
- [7] A. E. Eiben and J. E. Smith, "Introduction to Evolutionary Computing Genetic Algorithms." 2012.