

# Implementasi Algoritma Genetika dan Regresi Linier Berganda Untuk Prediksi Persediaan Bahan Makanan Pada Restoran Cepat Saji

## Implementation Of Genetic Algorithm and Multi-Linear Regression For Predicting Food Supplies At Fast Food Restaurants

Chairul Rozikin<sup>1</sup>, Achmad Solichin<sup>2</sup>

Fakultas Teknologi Informasi  
Universitas Budi Luhur, Jakarta, 12260  
Telp : (021) 5853753, Fax : (021) 5866369  
E-mail : [irulchazi@gmail.com](mailto:irulchazi@gmail.com), [achmad.solichin@budiluhur.ac.id](mailto:achmad.solichin@budiluhur.ac.id)

### Abstract

*Food supplies is an important thing in fast food restaurants. Many foods can not be stored too long to maintain the freshness and quality of the food ingredients. Currently, if an excess of food then the restaurant will move these foods into other branches that need. If no, the food ingredients will be thrown away. The transfer of foodstuffs to another branch would be an added expense. Therefore, we need a system that can predict exactly food supplies needed at a particular time. The primary objective of this research is to apply genetic algorithms to predict how much the amount of food that will be sold in the coming months. Thus, the fast-food restaurants can estimate how many foods that should be provided. The test results showed that the level of accuracy of prediction system generated 89.9% and an error rate of 10.1%. The results of this study are expected to be useful, especially for the restaurants in optimizing production costs.*

**Keywords** : *Genetic Algorithm, Multi-Linear Regression, Prediction, Food Supplies*

### Abstrak

*Persediaan bahan makanan merupakan suatu hal yang penting di restoran cepat saji. Banyak bahan makanan tidak bisa disimpan terlalu lama untuk menjaga kesejukan dan kualitas dari bahan makanan tersebut. Saat ini, jika terjadi kelebihan bahan makanan maka pihak restoran akan memindahkan bahan makanan tersebut ke cabang lain yang membutuhkan. Jika tidak ada, bahan makanan akan dibuang. Pemindahan bahan makanan ke cabang lain tentu akan memakan biaya tambahan. Oleh karena itu, diperlukan sistem yang dapat memprediksi dengan tepat persediaan bahan makanan yang diperlukan pada kurun waktu tertentu. Tujuan utama dari penelitian ini adalah menerapkan algoritma genetika untuk memprediksi seberapa banyak jumlah makanan yang akan terjual pada bulan selanjutnya. Dengan demikian, pihak restoran cepat saji dapat memperkirakan berapa jumlah bahan makanan yang harus disediakan. Hasil pengujian menunjukkan bahwa tingkat akurasi sistem prediksi yang dihasilkan sebesar 89,9% dan tingkat kesalahan sebesar 10,1%. Hasil penelitian ini diharapkan dapat bermanfaat khususnya bagi pihak restoran dalam mengoptimalkan biaya produksi.*

**Kata kunci** : *Algoritma Genetika, Regresi Linier Berganda, Prediksi, Persediaan Bahan Makanan*

### 1. PENDAHULUAN

Persediaan bahan makanan merupakan suatu hal yang penting khususnya di perusahaan yang bergerak di bidang penjualan makanan seperti restoran cepat saji. Namun pada praktiknya, merencanakan bahan makanan (belum diolah) hingga selanjutnya akan diolah bukanlah persoalan mudah, jika persediaan terlalu banyak akan merugikan perusahaan, karena sifat bahan makanan yang tidak tahan lama mengharuskan bahan makanan tersebut untuk dibuang atau dipindahkan ke cabang perusahaan lain secepatnya, yang berarti hal tersebut perlu memakan *cost* lebih untuk biaya transportasi. Di lain sisi jika persediaan terlalu sedikit juga akan menimbulkan biaya akibat kekurangan persediaan yang biasa disebut *stock out cost* seperti, mahalnya harga karena membeli dalam partai kecil, terganggunya proses produksi, dan tidak tersedianya produk jadi untuk pelanggan. 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 dengan *elitism*.

## 2. METODOLOGI PENELITIAN

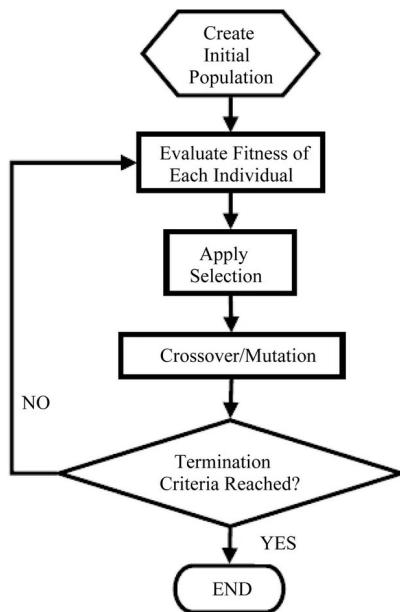
Pada penelitian ini, metode yang digunakan untuk menentukan nilai prediksi adalah dengan menggunakan proses algoritma genetika, dimana untuk pencarian nilai prediksi tersebut didapat dari data historis penjualan 12 bulan sebelumnya yang kemudian data tersebut akan dimodelkan sesuai dengan rumus regresi linier berganda, dan selanjutnya akan dikenai proses umum algoritma secara umum seperti *crossover*, mutasi, dan *update* generasi.

### 2.1. 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. Dengan meniru teori evolusi ini, algoritma genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata [2]. Pada awal perkembangannya, metode algoritma genetika ini pertama kali diperkenalkan oleh John Holland dari Universitas Michigan pada tahun 1975 dalam bukunya yang berjudul “*Adaption in Natural and Artificial Systems*”, dan pada akhirnya dipopulerkan oleh salah seorang muridnya, David Goldberg, yang mampu memecahkan masalah sulit yang melibatkan kontrol transmisi gas-pipa untuk disertasinya yang berjudul “*Computer-aided gas pipeline operation using genetic algorithms and rule learning*” [3]. Secara umum algoritma genetika memiliki lima komponen dasar menurut Michalewicz [4], yaitu :

- Representasi genetik dari solusi-solusi masalah.
- Cara membentuk populasi awal dari solusi-solusi.
- Fungsi evaluasi yang me-rate (rating) solusi-solusi berdasarkan *fitness* mereka.
- Operator-operator genetik yang merubah komposisi genetik dari *offspring* selama rekombinasi.
- 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 [5].

Algoritma genetika berangkat dari himpunan solusi yang dihasilkan secara acak yang disebut populasi. Sedangkan setiap individu dalam populasi disebut kromosom yang merupakan representasi dari suatu solusi dan masing-masing akan dievaluasi tingkat ketangguhannya (*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* [6].

### 2.1.1. Pengkodean

Hal pertama yang perlu diperhatikan dalam proses algoritma genetika adalah menentukan teknik pengkodean yang akan digunakan, dimana pengkodean itu sendiri merupakan suatu teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom. Ada beberapa jenis pengkodean yang dapat digunakan dalam algoritma genetika, seperti pengkodean biner, integer, bilangan real, maupun permutasi, dimana pada penelitian kali ini pengkodean bilangan real akan digunakan untuk menyelesaikan permasalahan yang ada.

0.391	0.221	0.501	0.701	0.232	0.594	0.695	0.926	0.875	0.708
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Gambar 2: Representasi pengkodean algoritma genetika menggunakan bilangan real

### 2.1.2. Regresi linier berganda

Regresi linier berganda adalah analisis regresi yang menjelaskan hubungan antara satu variabel tergantung (variabel dependen) dengan faktor-faktor yang mempengaruhi lebih dari satu prediktor (variabel independen) [7]. Tujuan analisis regresi linier berganda adalah untuk mengukur intensitas hubungan antara dua variabel atau lebih dan membuat prediksi perkiraan nilai. Secara umum model regresi linier berganda untuk populasi dapat dilihat pada persamaan (1) :

$$P(n) = \theta_0 + \theta_1 y_1 + \theta_2 y_2 + \theta_3 y_3 + \dots + \theta_n y_n \quad (1)$$

Dimana :  $P(n)$  = Hasil prediksi penjualan pada periode  $n$ .  
 $y_n$  = Hasil penjualan aktual pada periode  $n$ .  
 $\theta_0, \theta_1, \theta_2, \dots \theta_n$  = bilangan yang dibangkitkan secara *random* dari 0-1.

Setelah didapat nilai dari suatu prediksi, metode yang digunakan untuk mengevaluasi nilai *fitness* dari suatu individu adalah dengan menggunakan *Mean Squared Error* (MSE). MSE dihitung dengan menjumlahkan kuadrat semua kesalahan peramalan pada setiap periode dan membaginya dengan jumlah periode peramalan [8]. Proses dari MSE itu sendiri dapat dihitung menggunakan persamaan (2) :

$$MSE = \frac{\sum (X_n - Y_n)^2}{n} \quad (2)$$

Dimana :  $X_n$  = Hasil penjualan aktual pada periode  $n$ .  
 $Y_n$  = Hasil prediksi penjualan pada periode  $n$ .  
 $n$  = banyaknya periode waktu, pada kasus ini  $n = 12$ .

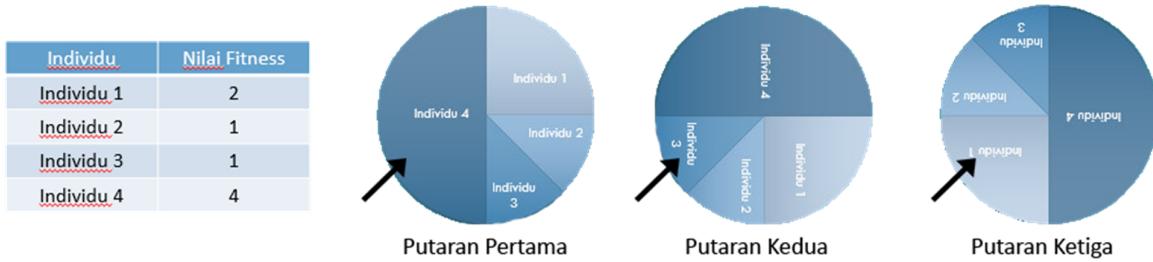
Langkah selanjutnya adalah menentukan nilai *fitness* dari tiap individu untuk mengetahui seberapa baik individu tersebut dibanding dengan individu lainnya, dimana nilai *fitness* tersebut didapat dari persamaan (3) :

$$\text{Nilai Fitness} = \frac{1}{MSE} \quad (3)$$

### 2.1.3. Roulette wheel selection

Tujuan dilakukannya proses seleksi adalah untuk melakukan pemilihan terhadap individu yang hendak diikutkan dalam proses rekombinasi. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Seleksi mempunyai tujuan untuk memberikan kesempatan rekombinasi yang lebih besar bagi anggota populasi yang mempunyai nilai *fitness* terbaik. Metode *roulette wheel selection* merupakan metode yang paling sederhana serta paling banyak digunakan [9], 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.



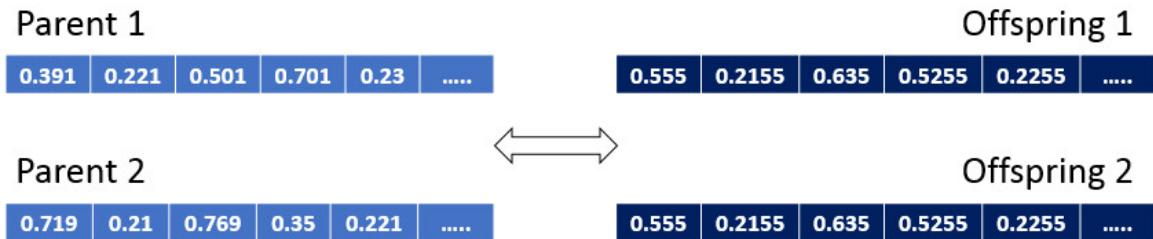
*Gambar 3: Ilustrasi proses roulette wheel selection*

#### 2.1.4. Whole arithmetic crossover

Pada whole arithmetic *crossover*, gen pada kromosom *offspring (child)* diperoleh dari hasil operasi aritmatika gen pada kromosom *parent*, dimana proses aritmatika yang dilakukan menurut Eiben & Smith [10] adalah seperti pada persamaan (4) :

$$child = \alpha \cdot \bar{x} + (1 - \alpha) \cdot \bar{y} \quad (4)$$

Dimana  $\bar{x}$  dan  $\bar{y}$  adalah nilai gen dari dua *parent* yang berbeda, dan juga  $\alpha$  adalah variabel pengali yang nilainya berkisar antara 0 – 1. Berikut merupakan ilustrasi proses *whole arithmetic crossover* dengan nilai  $\alpha$  sebesar 0.5, yang berarti hasil kedua *offspring* akan identik.



*Gambar 4: Ilustrasi proses whole arithmetic crossover*

### 2.1.5. *Random mutation*

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.

### 2.1.6. *Update generasi dengan elitism*

Setelah proses seleksi, *crossover* (rekombinasi) 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. Suatu 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 *crossover* 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 [6].

### 2.1.7. Parameter kontrol algoritma genetika

Parameter kontrol merupakan sebuah parameter yang mengontrol keseluruhan proses algoritma genetika mulai dari ukuran populasi, persentase probabilitas *crossover*, dan probabilitas mutasi. Nilai parameter ini ditentukan berdasarkan permasalahan yang ada, dimana nilai dari parameter 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.

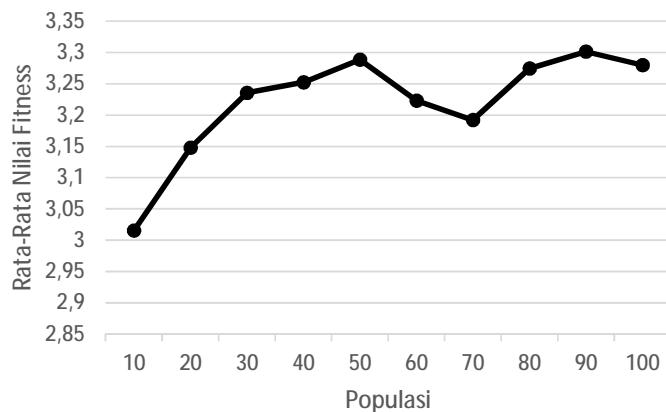
Bila *fitness* dari individu terbaik dipantau pada setiap generasi, maka usulannya adalah : ukuran populasi 80; probabilitas crossover 0.45; dan probabilitas mutasi 0.01. Ukuran populasi sebaiknya tidak lebih kecil dari 30, untuk sembarang jenis permasalahan [11].

## 3. HASIL DAN PENGUJIAN

Untuk mendapatkan nilai prediksi yang maksimal atau mendekati dengan data aktual, maka diperlukan pengujian sebelumnya untuk parameter kontrol algoritma genetika, karena pada dasarnya parameter kontrol algoritma genetika itu sendiri akan berbeda-beda tergantung dengan permasalahan yang dihadapi. Pada penelitian ini, pengujian akan dilakukan untuk setiap elemen dari parameter kontrol algoritma genetika yang meliputi : ukuran jumlah individu awal, ukuran generasi, nilai probabilitas *crossover*, nilai alfa, dan nilai probabilitas mutasi.

### 3.1. Pengujian terhadap ukuran jumlah individu awal (populasi)

Pada pengujian ini diberikan nilai probabilitas *crossover* sebesar 0.6, nilai alfa sebesar 0.5, nilai probabilitas mutasi sebesar 0.1, dengan jumlah generasi sebanyak 100. Jumlah individu yang diuji mulai dari 10 sampai dengan 100, dengan 5 kali pengujian pada setiap kelipatannya. Hasil yang didapat ialah:

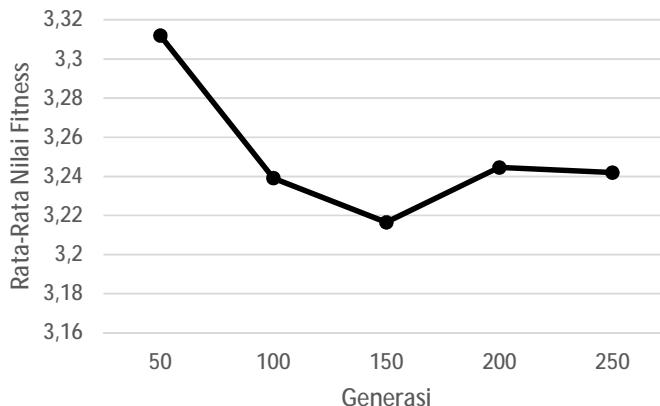


Gambar 5: Grafik pengujian ukuran jumlah individu awal (populasi)

Pada percobaan kali ini dapat ditarik kesimpulan bahwa jika dilihat dari nilai *fitness*-nya maka ukuran populasi 90 merupakan nilai terbaik, sedangkan jika dilihat dari peningkatan nilai *fitness* yang signifikan, maka ukuran populasi 20 merupakan nilai yang terbaik, diikuti oleh 30, dan kemudian 80.

### 3.2. Pengujian terhadap ukuran generasi

Pada pengujian ini diberikan jumlah individu sebanyak 90, nilai probabilitas *crossover* sebesar 0.6, nilai alfa sebesar 0.5, dan nilai probabilitas mutasi sebesar 0.1. Jumlah generasi yang diuji mulai dari 50 sampai dengan 250, dengan 5 kali pengujian pada setiap kelipatannya. Hasil yang didapat ialah:

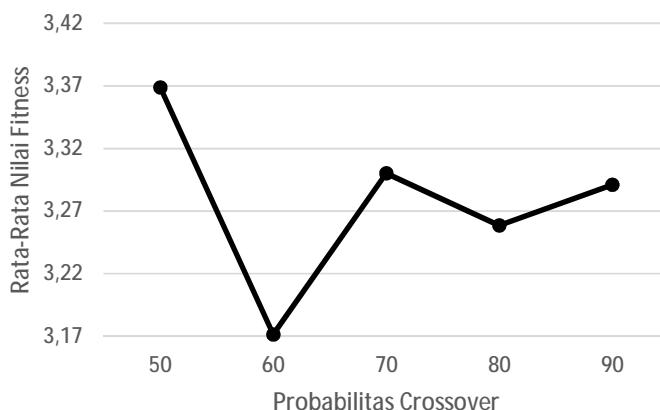


Gambar 6: Grafik pengujian ukuran generasi

Untuk pengujian selanjutnya ukuran generasi 50 dan 100 akan digunakan dan dianalisis, dimana ukuran generasi yang menghasilkan solusi dengan nilai konvergen akan digunakan, namun jika keduanya sama-sama menghasilkan nilai konvergen maka ukuran generasi 50 yang akan digunakan demi mempercepat proses running dari aplikasi ini, penelitian lainnya juga menunjukkan bahwa ukuran generasi yang besar belum tentu menghasilkan solusi yang terbaik [12].

### 3.3. Pengujian terhadap nilai probabilitas crossover

Pada pengujian ini diberikan jumlah individu sebanyak 90, nilai alfa sebesar 0.5, nilai probabilitas mutasi sebesar 0.1, dengan jumlah generasi sebanyak 100. Nilai probabilitas crossover yang diuji mulai dari 0.5 sampai dengan 0.9, dengan 5 kali pengujian pada setiap kelipatannya. Hasil yang didapat ialah:

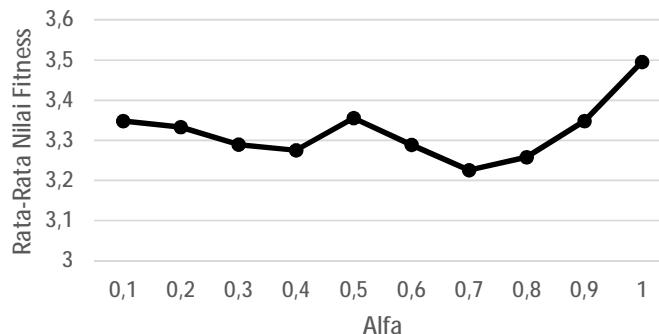


Gambar 6: Grafik pengujian nilai probabilitas crossover

Dapat diketahui dari Gambar 6 bahwa nilai probabilitas crossover 50, 60, dan 70 mengalami peningkatan dan penurunan yang signifikan, sedangkan 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.

### 3.4. Pengujian terhadap nilai alfa

Pada pengujian ini diberikan jumlah individu sebanyak 90, nilai probabilitas crossover sebesar 0.7, nilai probabilitas mutasi sebesar 0.1, dengan jumlah generasi sebanyak 100. Nilai alfa yang diuji mulai dari 0.1 sampai dengan 1, dengan 5 kali pengujian pada setiap kelipatannya. Hasil yang didapat ialah:

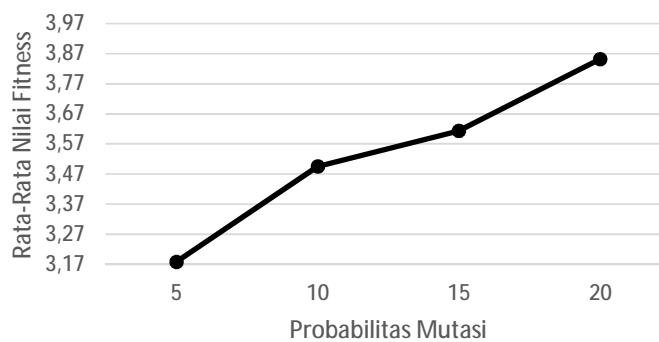


Gambar 7: Grafik pengujian nilai alfa

Pada Gambar 7 menunjukkan bahwa rata-rata nilai *fitness* terendah ada pada nilai alfa 0.7, dan rata-rata *fitness* tertinggi ada pada nilai alfa 1, diikuti oleh 0.9, lalu 0.5.

### 3.4. Pengujian terhadap nilai probabilitas mutasi

Pada pengujian ini diberikan jumlah individu sebanyak 90, nilai probabilitas *crossover* sebesar 0.7, nilai alfa sebesar 1, dengan jumlah generasi sebanyak 100. Nilai probabilitas mutasi yang diuji mulai dari 0.05 sampai dengan 0.2, dengan 5 kali pengujian pada setiap kelipatannya. Hasil yang didapat ialah:

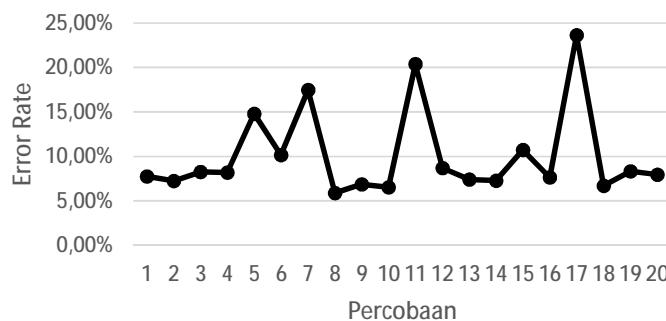


Gambar 8: Grafik pengujian nilai probabilitas mutasi

Pada Gambar 8 menunjukkan bahwa nilai *fitness* terendah ada pada nilai probabilitas mutasi sebesar 5%, dan nilai *fitness* tertinggi ada pada nilai probabilitas mutasi sebesar 20%, dengan begitu nilai mutasi 20% atau 15% akan digunakan.

### 3.4. Rangkuman Hasil Pengujian

Melihat hasil dari beberapa pengujian sebelumnya, maka digunakan parameter kontrol terbaik sesuai dengan pengujian sebelumnya dan dilakukan *running* sebanyak 20 kali demi melihat *stability*, *consistency* dan *reliability* dari hasil prediksi yang dihasilkan.



Gambar 9: Grafik hasil pengujian

Dimana didapat nilai *error* terendah sebesar 5.89% dan nilai *error* tertinggi sebesar 23.65% dengan rata-rata nilai *error* sebanyak 10.10% dan rata-rata *running time* selama 36.69 detik. Nilai *error* ini didapat dengan cara membandingkan data penjualan aktual dengan data hasil prediksi, sehingga diharapkan, dengan nilai *error* yang kecil dapat membantu pihak restoran cepat saji untuk menentukan banyaknya bahan makanan yang perlu disediakan untuk bulan selanjutnya.

#### 4. KESIMPULAN

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

- a. Algoritma genetika dapat digunakan untuk mencari nilai prediksi pada kasus persediaan bahan makanan dengan memanfaatkan data historis penjualan dari beberapa bulan sebelumnya.
- b. Waktu yang diperlukan untuk mendapatkan nilai prediksi yang akurat berbanding lurus dengan banyaknya iterasi yang ada pada proses algoritma genetika.
- c. Hasil solusi yang dihasilkan akan berbeda-beda setiap kali dijalankan, namun dengan proses iterasi yang besar dapat menghasilkan satu solusi terbaik dimana hasil yang didapat akan konvergen, data pengujian menunjukkan bahwa algoritma gentika dapat menghasilkan nilai prediksi dengan rata-rata *error rate* sebesar 10.10% atau dengan kata lain algoritma genetika mempunyai akurasi sebesar 89.90%.

#### 5. DAFTAR PUSTAKA

- [1] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, Second Edi. New Jersey: John Wiley & Sons, Inc., 2004.
- [2] R. M. Sukaton, "Penggunaan Algoritma Genetika Dalam Masalah Jalur Terpendek Pada Jaringan Data," Universitas Indonesia, 2011.
- [3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, vol. 35, no. 2. 1989.
- [4] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs," *Computational Statistics & Data Analysis*, vol. 24, no. 3. Springer, pp. 372–373, 1996.
- [5] 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.
- [6] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*. John Wiley & Sons, Inc., 2000.
- [7] 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.
- [8] Farizal, A. Rachman, and H. Al Rasyid, "Model Peramalan Konsumsi Bahan Bakar Jenis Premium Di Indonesia Dengan Regresi Linier Berganda," *J. Ilm. Tek. Ind.*, vol. 13, no. 2, pp. 166–176, 2014.
- [9] D. Y. Liliana, T. A. Dewi, D. E. Ratnawati, S. Y. Prakoso, and D. Herawatie, "Komputasi Cerdas," in *Seminar Nasional Teknologi Informasi & Komputasi 2012 (SENASTIK 2012) Bangkalan, 13-14 Nopember ISSN*, 2012.
- [10] A. E. Eiben and J. E. Smith, "Introduction to Evolutionary Computing Genetic Algorithms." 2012.
- [11] S. Kusumadewi, *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu, 2003.
- [12] A. I. Permatasari and W. F. Mahmudy, "Pemodelan Regresi Linear dalam Konsumsi Kwh Listrik di Kota Batu Menggunakan Algoritma Genetika," *DORO Repos. J. Mhs. PTIHK Univ. Brawijaya*, vol. 5, no. 14, pp. 1–9, 2015.