**BAB V**

**IMPLEMENTASI DAN HASIL**

* 1. **Implementasi**

1. **Analisis Permasalahan**

Analisis permasalahan dilakukan berdasarkan penelitian di PT Telkom Indonesia. Dimana terdapat beberapa permasalahan, yaitu diantaranya mengenai strategi pemasaran, peramalan nilai pendapatan per bulannya, dan optimasi nilai pendapatan tersebut. Hal ini dirasa belum efektif dikarenakan masih ada beberapa produk yang kurang diminati oleh pelanggan. Maka dari itu perlunya mengetahui jumlah produk yang terjual tiap bulannya, dan jumlah pelanggan yang berlangganan tiap bulannya sebagai langkah awal untuk menganalisa, memprediksi serta mengoptimasi nilai pendapatan tersebut. Untuk mengetahui nilai hasil peramalan dan nilai optimasi tersebut digunakanlah teknik *data mining* dengan menggunakan Regresi Linier Berganda untuk meramalkan nilai, dan Algoritma Genetika untuk mengoptimasi nilai. Data yang dianalisis merupakan data pelanggan yang didapat dari arsip perusahaan. Kemudian dari data tersebut diambil 3 atribut, yaitu jumlah pelanggan yang berlangganan, jumlah produk, dan harga dari setiap produk. Data tersebut mewakili nilai untuk peramalan dan optimasi pendapatan.

1. **Pengumpulan Data**

Pengumpulan data dilakukan dengan mengambil data dari arsip perusahaan yang berupa data pelanggan yang berada dalam wilayah operasional witel Bandung. Data tersebut berisikan informasi yang diperoleh perusahaan mengenai pelanggan selama berlangganan layanan dari mulai bulan Januari hingga Oktober 2019. Informasi tersebut beberapa diantaranya nomor pelanggan, alamat, tanggal registrasi, tanggal layanan dapat digunakan, tanggal berlangganan, layanan *addOn* yang digunakan, serta harga dari masing-masing layanan *addOn*.

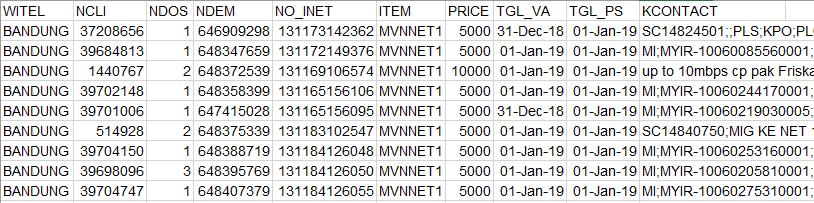
1. **Pengolahan Data Menggunakan Regresi Linier Berganda**
2. **Data Pelanggan**

Pada tahap ini dilakukannya pengumpulan dan penyiapan data pelanggan yang akan dilakukan peramalan dan optimasi. Berikut merupakan data pelanggan yang akan digunakan.

1. Data Pelanggan Movin:

Data pelanggan movin merupakan data pelanggan yang berlangganan *addOn* movin.

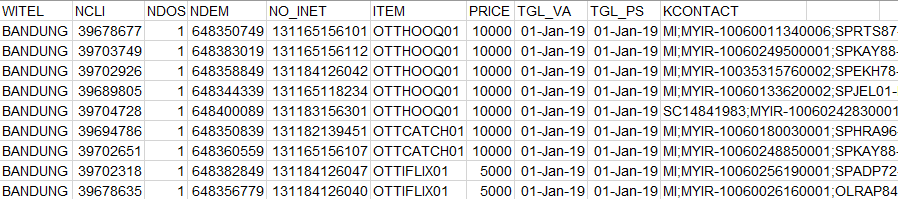
Tabel 5. 1 Data Pelanggan Movin



1. Data Pelanggan OTT

Data pelanggan ott merupakan data pelanggan yang berlangganan *addOn* ott.

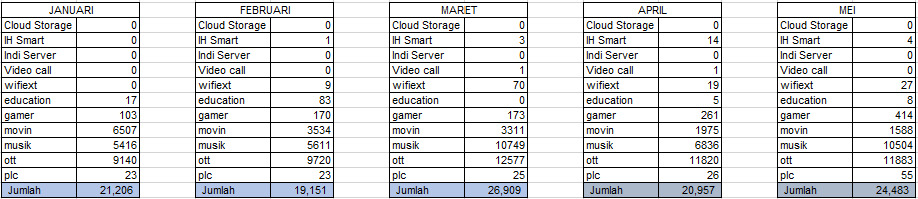
Tabel 5. 2 Data Pelanggan OTT



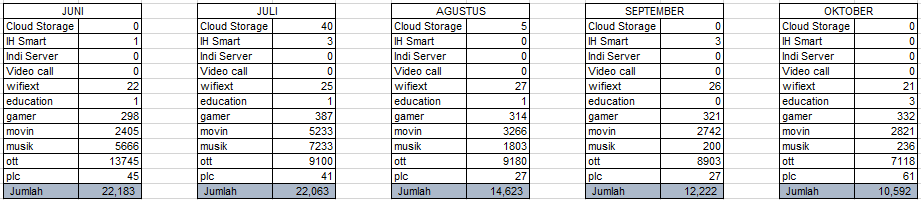
1. **Menjumlahkan Total Pelanggan**

Untuk menjumlahkan total pelanggan, kita harus mengetahui jumlah pelanggan yang berlangganan dari setiap produknya. Berikut merupakan cara yang digunakan untuk menjumlahkan total pelanggan dari setiap produknya.

Tabel 5. 3 Penjumlahan Total Pelanggan

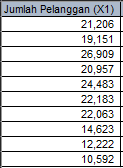


Tabel 5. 4 Penjumlahan Total Pelanggan



Setelah dilakukan penjumlahan seperti tabel tersebut, maka langkah selanjutnya tinggal menyatukan masing-masing jumlah tersebut ke dalam kolom tabel jumlah pelanggan (X1). Berikut hasilnya.

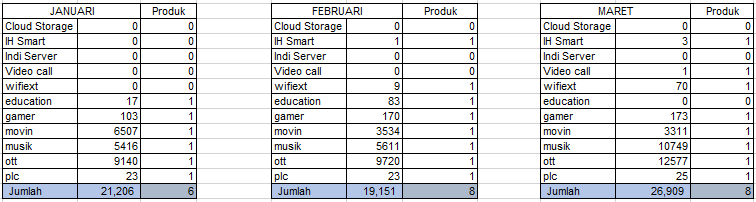
Tabel 5. 5 Jumlah Pelanggan



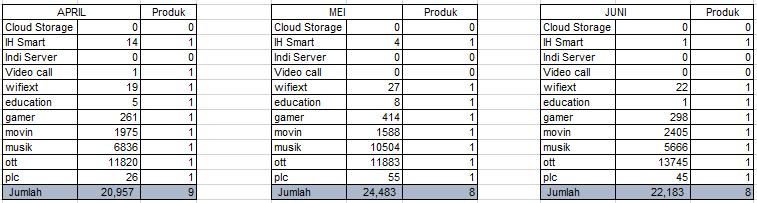
1. **Menjumlahkan Total Produk**

Dalam menentukan jumlah produk pada setiap bulannya, saya melakukan penjumlahan sesuai dengan pelanggan yang berlangganan di setiap bulannya. Berikut hasil dari total produk dari bulan Januari – Oktober.

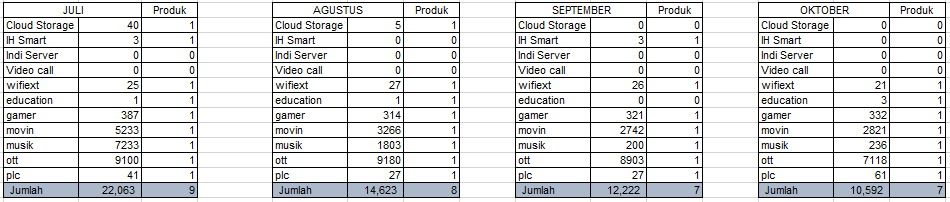
Tabel 5. 6 Penjumlahan Total Produk



Tabel 5. 7 Penjumlahan Total Produk

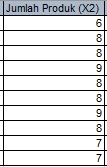


Tabel 5. 8 Penjumlahan Total Produk



Setelah menjumlahkan total produk pada setiap bulannya, selanjutnya kita jadikan menjadi satu kolom tabel total produk (X2). Berikut hasilnya.

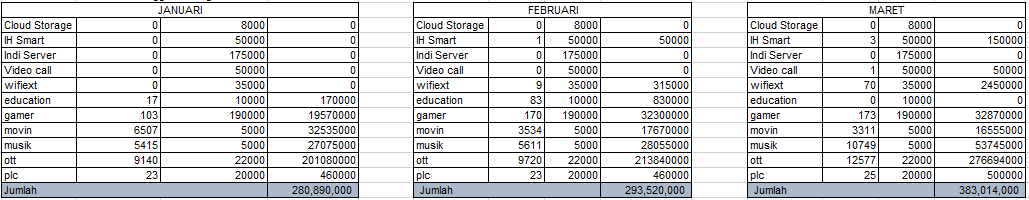
Tabel 5. 9 Jumlah Produk



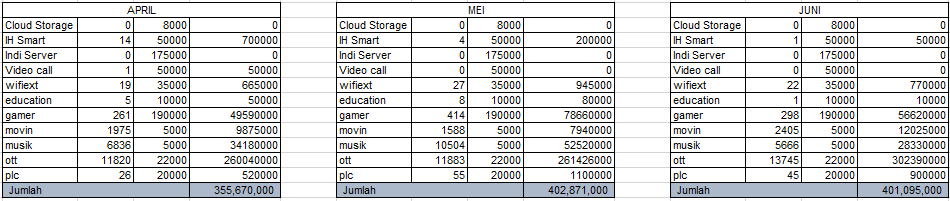
1. **Menjumlahkan Total Pendapatan**

Dalam menentukan jumlah pendapatan ini, hal yang harus diperhatikan ialah jumlah pelanggan dan harga dari masing-masing produk yang dipasarkan. Jika kedua atribut tersebut telah didapatkan, maka selanjutnya kita mengalikan nilai jumlah pelanggan dengan harga dari produk tersebut. Berikut hasilnya.

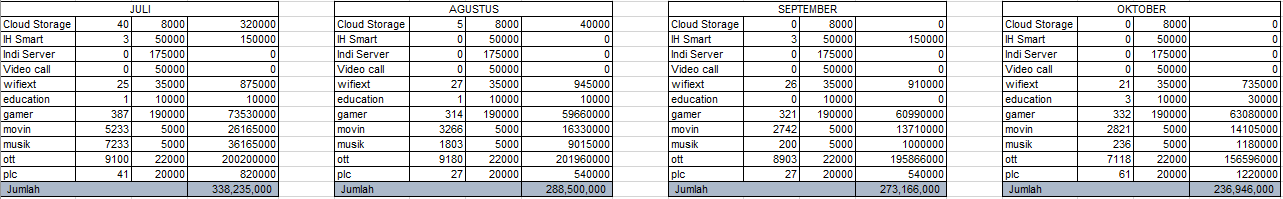
Tabel 5. 10 Penjumlahan Nilai Pendapatan



Tabel 5. 11 Penjumlahan Nilai Pendapatan

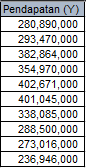


Tabel 5. 12 Penjumlahan Nilai Pendapatan



Setelah menjumlahkan nilai pendapatan dari setiap bulannya, langkah selanjutnya ialah menggabungkan nilai tersebut ke dalam kolom pendapatan (Y).

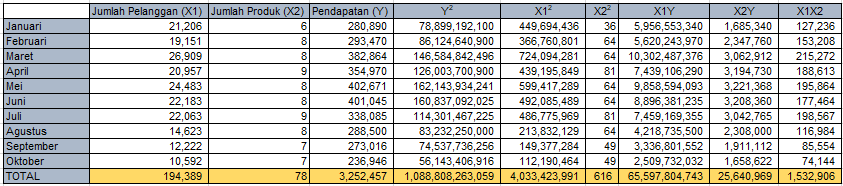
Tabel 5. 13 Hasil Penjumlahan Nilai Pendapatan



1. **Proses Pemangkatan Nilai**

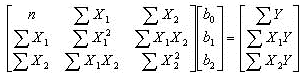
Proses ini dilakukan untuk mencari nilai matrix dan determinan. Proses tersebut diantaranya dengan memangkatkan masing-masing variabel bebas dan variabel terikat. Kemudian, nilai variabel bebas dikalikan dengan variabel terikat. Dan, yang terakhir mengalikan antara variabel bebas.

Tabel 5. 14 Hasil Nilai Setelah Dilakukan Proses Pemangkatan.



1. **Menentukan Nilai Determinan A, B1, dan B2**

Untuk mencari nilai determinan A, B1, dan B2 diperlukan nilai matrix yang didapatkan dari tabel hasil nilai pemangkatan. Proses mencari nilai matrix ini menggunakan rumus sebagai berikut.



Gambar 5. 1 Rumus Matrix

Masukkan nilai hasil pemangkatan pada rumus tersebut untuk mencari nilai determinan A. Sehingga hasilnya akan seperti berikut.

Tabel 5. 15 Nilai Matrix Untuk Mencari Nilai Determinan A



Hasil dari nilai matrix A untuk mencari nilai determinan A ialah sebagai berikut.



Kemudian untuk mencari nilai A, diperlukan nilai matrix A1 dan nilai determinan A1. Dalam menentukan nilai matrix A1 ada langkah yang perlu diperhatikan, diantaranya. Pada kolom pertama (n, ∑X1, dan ∑X2) dirubah menjadi (∑Y, ∑X1Y, dan ∑X2Y). Sehingga hasilnya seperti berikut.

Tabel 5. 16 Nilai Matrix Untuk Mencari Nilai Determinan A1

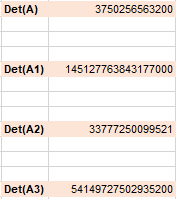


Hasil dari nilai matrix A1 untuk mencari nilai determinan A1 ialah sebagai berikut.



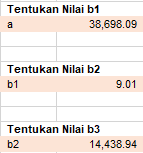
Hal yang sama dilakukan untuk mencari nilai determinan A2, dan A3. Hasil dari nilai determinan tersebut digunakan untuk mencari nilai a, b1, dan b2 yang akan digunakan pada proses perhitungan menggunakan rumus Regresi Linier Berganda. Berikut merupakan hasil nilai determinan yang telah dilakukan.

Tabel 5. 17 Hasil Perhitungan Untuk Mencari Nilai Determinan



Untuk mencari nilai a rumus yang digunakan ialah determinan A1 / determinan A. Begitupun hal yang sama dilakukan untuk mencari nilai b1, dan b2, yakni dengan cara membagi determinan A2 maupun A3 dengan determinan A. Sehingga hasil yang didapatkan ialah sebagai berikut.

Tabel 5. 18 Hasil Untuk Nilai A, B1, dan B2



1. **Menghitung Menggunakan Rumus Regresi Linier Berganda**

Setelah proses-proses tersebut dilewati dan mendapatkan nilai untuk a, b1, dan b2. Selanjutnya tinggal menghitung variabel terikat yang akan diprediksi. Berikut contoh perhitungan dan hasilnya.

Tabel 5. 19 Perhitungan Menggunakan Rumus Regresi Linier Berganda



Disini peneliti mencoba untuk memprediksi nilai pendapatan pada bulan Oktober 2019 dengan nilai X1 = 10,592, dan X2 = 7. Hasil yang didapatkan dari perhitungan tersebut ialah Y = 238,851.

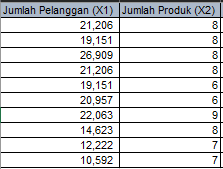
1. **Optimasi Data Menggunakan Algoritma Genetika**

Optimasi ini dilakukan karena nilai *error* yang dihasilkan dari metode Regresi Berganda ini cukup tinggi, sehingga diperlukan optimasi dengan harapan dapat mengurangi nilai *error.* Berikut merupakan tahapan-tahapan yang akan dilakukan dalam proses optimasi menggunakan Algoritma Genetika.

1. **Dataset**

Untuk nilai yang akan dioptimasi diantaranya, jumlah pelanggan dan jumlah produk yang telah diolah pada tahap sebelumnya.

Tabel 5. 20 Dataset Untuk Optimasi



1. **Inisialisasi**

Inisialisasi digunakan untuk membuat kromosom yang dibuat secara *random*, yaitu dengan mengacak nilai gen inisialisasi berupa angka dari jumlah data yang akan digunakan, sehingga didapatkan sebuah kromsom yang urutan gennya acak.

ID untuk nilai X1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 6 | 8 | 2 | 3 | 5 | 7 | 9 | 1 | 4 |

ID untuk nilai X2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 6 | 10 | 8 | 1 | 3 | 9 | 7 | 5 | 4 |

Setiap angka tersebut adalah sebuah gen dari kromosom yang telah diberikan identitas sebagai pengganti dari nilai jumlah pelanggan dan jumlah produk. Untuk jumlah gen telah ditentukan sebanyak jumlah nilai yang akan diproses. Kromosom digunakan untuk merepresentasikan suatu solusi yang mungkin dari permasalahan yang akan diselesaikan menggunakan algoritma genetika. Berikut merupakan kromosom yang dibangkitkan atau yang akan digunakan dalam proses optimasi.

Inisialisasi

* Kromosom[1] = [1 4]
* Kromosom[2] = [2 5]
* Kromosom[3] = [3 6]
* Kromosom[4] = [6 3]
* Kromosom[5] = [3 2]
* Kromosom[6] = [4 1]

1. **Evaluasi Nilai Kromosom**

Permasalahan yang ingin diselesaikan adalah nilai variabel X1 dan X2, maka fungsi objektif dapat digunakan untuk mendapatkan solusi. Kita hitung fungsi objektif dari kromsom yang telah dibangkitkan.

* Kromosom[1] = (27 + 8.15\*1 + 17\*4)

= (27 + 8.15 + 68)

= 103.15

* Kromosom[2] = (27 + 8.15\*2 + 17\*5)

= (27 + 17 + 85)

= 129

* Kromosom[3] = (27 + 8.15\*3 + 17\*6)

= (27 + 24.45 + 102)

= 153.45

* Kromosom[4] = (27 + 8.15\*6 + 17\*3)

= (27 + 48.9 + 51)

= 126.9

* Kromosom[5] = (27 + 8.15\*3 + 17\*2)

= (27 + 24.45 + 34)

= 85.45

* Kromosom[6] = (27 + 8.15\*4 + 17\*1)

= (27 + 32.6 + 17)

= 76.6

* Rata-rata nilai dari evaluasi

= (103.15 + 129 + 153.45 + 126.9 + 85.45 + 76.6) / 6

= 112.425

1. **Menghitung Nilai *Fitness***

Pada proses seleksi ini dilakukan dengan cara membuat kromsom yang mempunyai nilai fungsi objektif kecil, mendapatkan kemungkinan terpilih lebih besar dengan nilai probabilitasnya yang tinggi. Untuk itu dapat digunakan fungsi *fitness* = 1 / fungsi\_objektif.

* Fitness[1] = 1 / 104.15

= 0.0096

* Fitness[2] = 1 / 130

= 0.0076

* Fitness[3] = 1 / 154.45

= 0.0064

* Fitness[4] = 1 / 127.9

= 0.0078

* Fitness[5] = 1 / 86.45

= 0.0115

* Fitness[6] = 1 / 77.6

= 0.0128

* Total Nilai *Fitness*

= 0.0096 + 0.0076 + 0.0064 + 0.0078 + 0.0115 + 0.0128

= 0.0557

1. **Mencari Nilai Probabilitas**

Setelah nilai *fitness* ditemukan, selanjutnya mencari nilai probabilitas dengan rumus P[i] = *fitness*[i] / total\_fitness.

* P[1] = 0.0096 / 0.0557

= 0.1723

* P[2] = 0.0076 / 0.0557

= 0.1364

* P[3] = 0.0064 / 0.0557

= 0.1149

* P[4] = 0.0078 / 0.0557

= 0.1400

* P[5] = 0.0115 / 0.0557

= 0.2064

* P[6] = 0.0128 / 0.0557

= 0.2298

Dari probabilitas tersebut terlihat bahwa kromosom ke-6 memiliki *fitness* paling besar, maka kromosom tersebut mempunyai probabilitas untuk terpilih pada generasi selanjutnya lebih besar dari kromosom lainnya.

1. **Mencari Nilai Komutatif**

Untuk proses seleksi digunakan *roulete-wheel*, untuk itu terlebih dahulu mencari nilai komutatif dari probabilitasnya.

C[1] = 0.1723

C[2] = 0.1723 + 0.1364 = 0.3087

C[3] = 0.3087 + 0.1149 = 0.4236

C[4] = 0.4236 + 0.1400 = 0.5636

C[5] = 0.5636 + 0.2064 = 0.77

C[6] = 0.77 + 0.2298 = 1

Proses *roulete-wheel* adalah membangkitkan nilai acak R antara 0-1. Jika R[k] < C[k], maka kromosom ke-k tersebut sebagai induk. Setelah itu putar *roulete-wheel* sebanyak jumlah kromsom.

1. **Membangkitkan Nilai Acak R**

Nilai acak ini dibangkitkan untuk membantu proses terbentuknya populasi baru.

R[1] = 0.201

R[2] = 0.284

R[3] = 0.009

R[4] = 0.822

R[5] = 0.398

R[6] = 0.501

1. **Populasi Baru Terbentuk**

Angka acak R[1] nilainya lebih besar dari C[1] dan lebih kecil dari C[2], maka pilih kromsom [2] sebagai kromosom pada populasi baru. Kromosom baru dari hasil proses seleksi, sebagai berikut.

Kromosom[1] = [2] = [2 5]

Kromosom[2] = [2] = [2 5]

Kromosom[3] = [1] = [1 4]

Kromosom[4] = [5] = [3 2]

Kromosom[5] = [2] = [2 5]

Kromosom[6] = [3] = [3 6]

1. ***Crossover***

Kromsom yang dijadikan induk dipilih secara acak dari jumlah kromsom yang di *crossover* dipengaruhi oleh parameter *crossover* probabilitas, misalnya = 25% = 0,25.

1. Membangkitkan nilai acak R

R[1] = 0.191

R[2] = 0.259

R[3] = 0.760

R[4] = 0.006

R[5] = 0.159

R[6] = 0.340

\*kromosom ke-k yang dipilih sebagai induk. Jika R[k] < nc, maka yang akan dijadikan induk adalah kromosom [1], [4], dan [5].

1. Bilangan acak untuk 3 kromosom induk yang akan di *crossover*

C[1] = 1

C[4] = 1

C[5] = 2

1. Proses *Crossover*

Kromosom[1] = kromosom[1] >< kromosom[4]

= [1 4] >< [6 3]

= [1 6]

Kromosom[4] = kromosom[4] >< kromosom[5]

= [6 3] >< [3 2]

= [3 2]

Kromosom[5] = kromosom[5] >< kromosom[1]

= [3 2] >< [1 4]

= [2 1]

Populasi setelah dilakukan *crossover*

* Kromosom[1] = [1 6]
* Kromosom[2] = [2 5]
* Kromosom[3] = [3 6]
* Kromosom[4] = [3 2]
* Kromosom[5] = [2 1]
* Kromosom[6] = [4 1]

1. **Mutasi**

Menghitung total panjang gen.

Panjang total gen = jumlah gen dalam 1 kromsom \* jumlah kromosom = 2 \* 6 = 12.

Untuk memilih posisi gen yang mengalami mutasi dilakukan dengan memberikan bilangan acak 1 – 12. Misal kita tentukan nilai nm = 20%, maka jumlah gen yang dimutasi adalah 0,2 \* 12 = 2,4 = 2. Dua buah posisi gen yang akan dimutasi, setelah diacak adalah posisi gen 4 dan 7. Lalu gen pada posisi tersebut nilainya diganti dengan ID yang telah ditentukan sebelumnya.

1. Proses mutasi

Kromosom[1] = [1 6]

Kromosom[2] = [2 6]

Kromosom[3] = [3 6]

Kromosom[4] = [1 2]

Kromosom[5] = [2 1]

Kromosom[6] = [4 1]

1. **Menghitung Nilai Fungsi Objektif**

Kromosom[1] = (27 + 8.15\*1 + 17\*6)

= (27 + 8.15 + 102)

= 137.15

Kromosom[2] = (27 + 8.15\*2 + 17\*6)

= (27 + 16.3 + 102)

= 145.3

Kromosom[3] = (27 + 8.15\*3 + 17\*6)

= (27 + 24.45 + 102)

= 153.45

Kromosom[4] = (27 + 8.15\*1 + 17\*2)

= (27 + 8.15 + 34)

= 69.15

Kromosom[5] = (27 + 8.15\*2 + 17\*1)

= (27 + 16.3 + 17)

= 60.3

Kromosom[6] = (27 + 8.15\*4 + 17\*1)

= (27 + 32.6 + 17)

= 76.6

Rata-rata nilai dari evaluasi

= (137.15 + 145.3 + 153.45 + 69.15 + 60.3 + 76.6) / 6

= 106.991

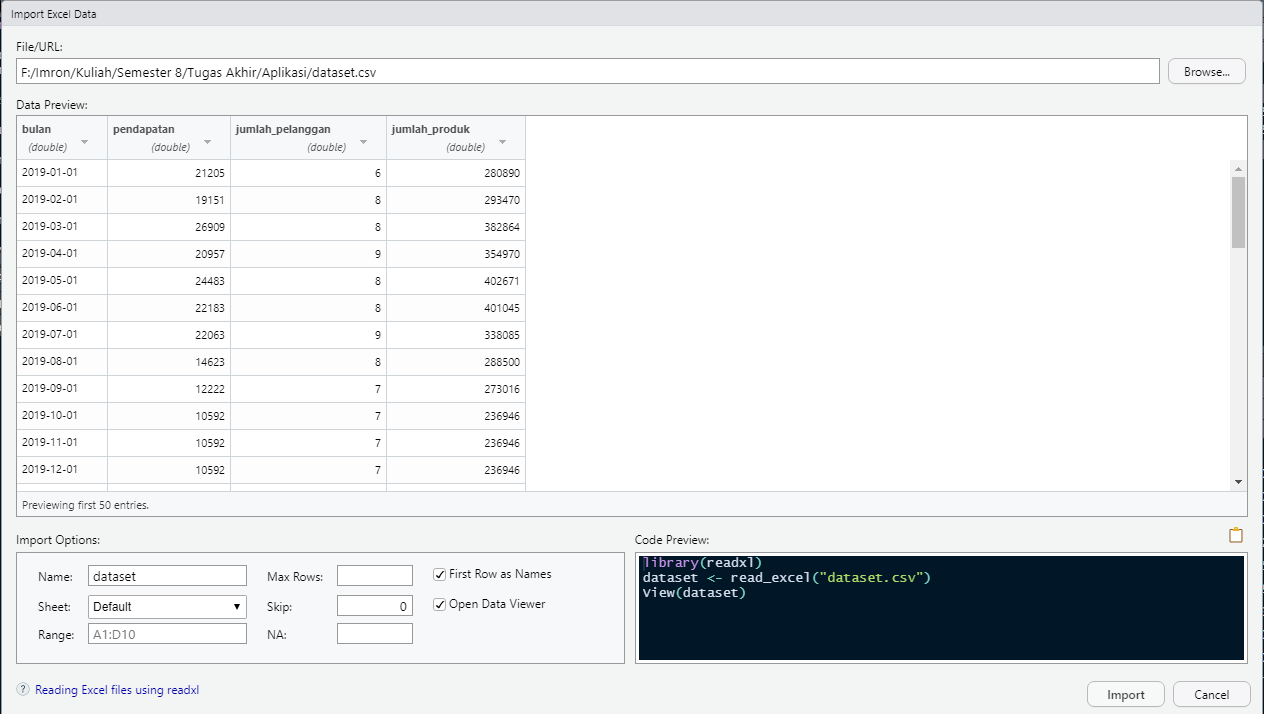
* 1. **Pemodelan Regresi Berganda**

Pemodelan adalah proses untuk membuah model dari sistem. Model simulasi merupakan salah satu bentuk model matematis yang bersifat deskriprif atau prediktif. Simulasi didefinisikan sebagai sekumpulan metode dan aplikasi untuk menirukan atau merepresentasikan perilaku dari suatu sistem nyata, yang biasanya dilakukan pada komputer dengan menggunakan perangkat lunak tertentu. Tujuan suatu pemodelan adalah untuk menganalisa dan memberi prediksi yang dapat mendekati kenyataan sebelum sistem diterapkan di lapangan.

R studio adalah salah satu *Graphic User Interface* (GUI) untuk bahasa pemrograman R, salah satu keunggulan pada R studio ini adalah dapat dijalankan pada browser, maka pengguna tidak memerlukan lagi installasi R, kecuali *package* pemrograman sesuai dengan kebutuhan[1]. Selain itu R studio dapat digunakan untuk merancang, mengembangkan, melaksanakan, dan berkomunikasi dengan kode khusus, R studio mengandung kode komputer dan mempunyai fitur tampilan.

1. ***Import Dataset***

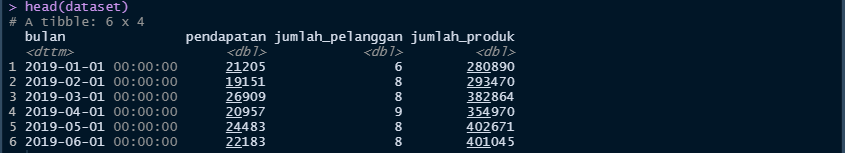
Pada tahap ini penulis melakukan *import dataset* yang akan digunakan dalam proses pemodelan menggunakan metode regresi bergandadan optimasi menggunakan algoritma genetika.



Gambar 5. 2 Import Dataset

1. **Baca Data**

Pada tahap ini melakukan pembacaan data yang telah di *import* pada tahap sebelumnya. Perintah yang digunakan untuk melihat data teratas dengan head(dataset), perintah tersebut dapat menampilkan data dalam bentuk excel dengan format xlsx maupun csv.



Gambar 5. 3 Baca Data Teratas

1. ***Import Module***

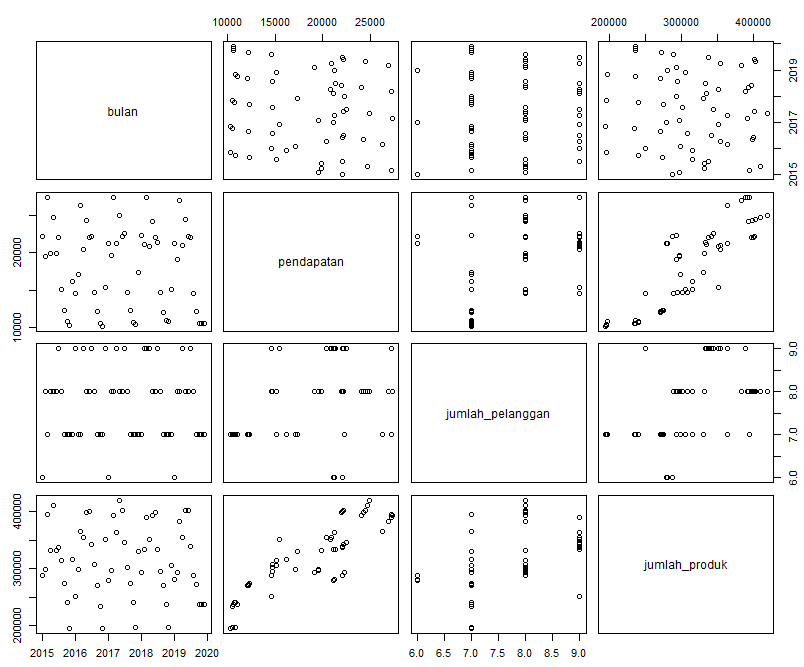
Pada tahap ini penulis melakukan *import module* atau *package* yang digunakan untuk memanggil fungsi yang digunakan selama proses pemodelan, dikarenakan R studio merupakan salah satu aplikasi yang menggunakan bahasa pemrograman R, sehingga *package* yang digunakan merupakan *package* yang berjalan pada bahasa pemrograman R seperti shiny, shinydashboard, lm, ggplot2, DT. *Library* ggplot2 ini berguna untuk menampilkan chart dari proses pengolahan data menggunakan metode regresi berganda.



Gambar 5. 4 Import Module

1. ***Pairs Dataset***

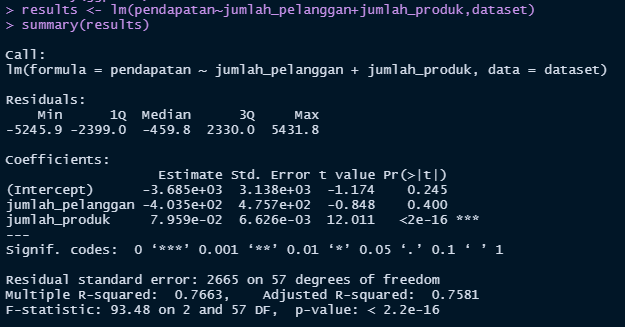
Fungsi *pairs* ini berguna untuk menampilkan korelasi dataset sebelum dilakukan pengolahan. Dengan begitu kita dapat mengetahui variabel mana saja yang saling berkorelasi satu sama lain. Hasil yang ditampilkan berupa *plot chart* sebagai berikut.



Gambar 5. 5 Hasil Pairs Dataset

1. **Perhitungan Menggunakan Metode Regresi Berganda**

Pada tahap ini dataset yang telah di *import* akan dihitung terkait korelasi yang terjadi diantara variabel-variabel tersebut.

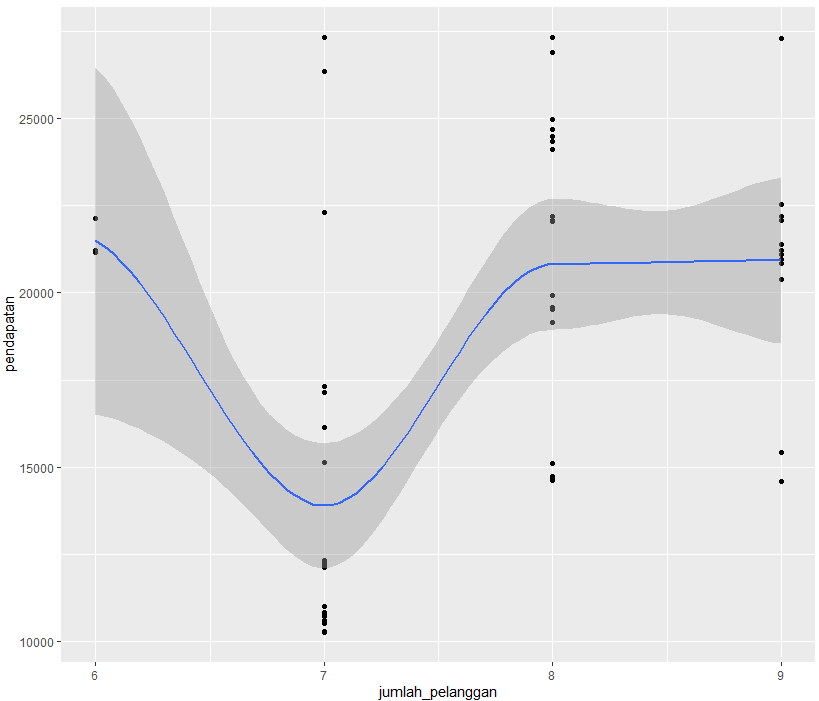


Gambar 5. 6 Perhitungan Metode Regresi Berganda

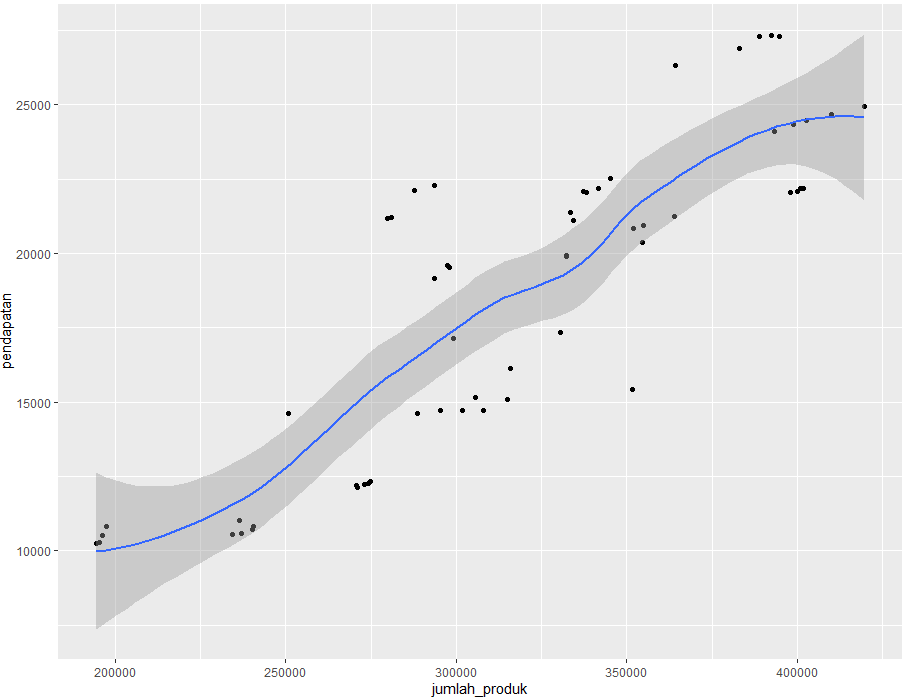
Hasil tersebut cukup baik untuk korelasi antara variabel bebas (jumlah pelanggan dan jumlah produk) terhadap variabel terikat (pendapatan). Karena untuk *Multiple R-squared* yang dihasilkan sebesar 76% dan *Adjusted R-squared* sebesar 75% artinya korelasi tersebut ada, namun belum cukup kuat dikarenakan masih menghasilkan *standard error* yang cukup tinggi.

1. **Plot Hasil Perhitungan**

Tahap ini akan menampilkan plot hasil dari perhitungan yang telah dilakukan pada tahap sebelumnya. Berikut hasil plot untuk jumlah pelanggan dengan pendapatan dan jumlah produk dengan pendapatan.



Gambar 5. 7 Hasil Plot Jumlah Pelanggan dengan Pendapatan



Gambar 5. 8 Hasil Plot Jumlah Produk dengan Pendapatan

* 1. **Pemodelan Algoritma Genetika**

Setelah pemodelan regresi berganda, selanjutnya akan dilakukan pemodelan dengan menggunakan algoritma genetika. Hal ini bertujuan untuk optimasi nilai variabel bebas dari dataset tersebut. Berikut langkah-langkahnya untuk pemodelan menggunakan algoritma genetika.

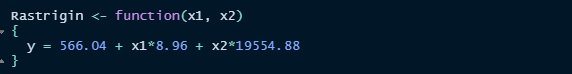
1. ***Import Module***



Gambar 5. 9 Import Module Algoritma Genetika

Pada tahap ini penulis melakukan *import module* atau *library* yang digunakan dalam proses pemodelan menggunakan algoritma genetika. *Library* yang digunakan ini dapat berjalan menggunakan bahasa pemrograman R. *Library* GA ini berfungsi untuk optimasi nilai, namun tidak hanya untuk optimasi, *library* GA ini dapat digunakan sesuai dengan kebutuhan *user* yang menggunakannya.

1. **Mendifinisikan Rumus**

******

Gambar 5. 10 Mendefinisikan Rumus

Tahapan ini digunakan untuk mendefinisikan rumus dan nilai yang akan dilakukan optimasi. Untuk rumusnya menggunakan rumus regresi linier berganda dengan nilai x1 dan x2 yang akan dilakukan optimasi.

1. **Mengurutkan Angka**

****

Gambar 5. 11 Mengurutkan Angka Acak

Fungsi seq() ini akan menghasilkan urutan angka secara acak dengan penjumlahan yang telah ditentukan, pada penelitian ini peneliti membangkitkan urutan angka acak dari -2 sampai 2 dengan penambahan 0.5 untuk setiap angkanya. Penambahan angka ini berfungsi untuk menentukan *values* yang akan dilakukan optimasi pada tahap selanjutnya. Berikut merupakan hasil dari urutan angka yang telah dilakukan.



Gambar 5. 12 Hasil Urutan Angka

1. **Menerapkan Fungsi Array Pada Nilai X1 dan X2**

****

Gambar 5. 13 Function Outer

Fungsi outer() ini berfungsi untuk menerapkan array pada variabel yang akan dilakukan optimasi. Fungsi outer() ini diterapkan pada *values* yang sudah dihasilkan pada tahapan sebelumnya, sehingga hasil dari *function* outer() ini sebagai berikut.



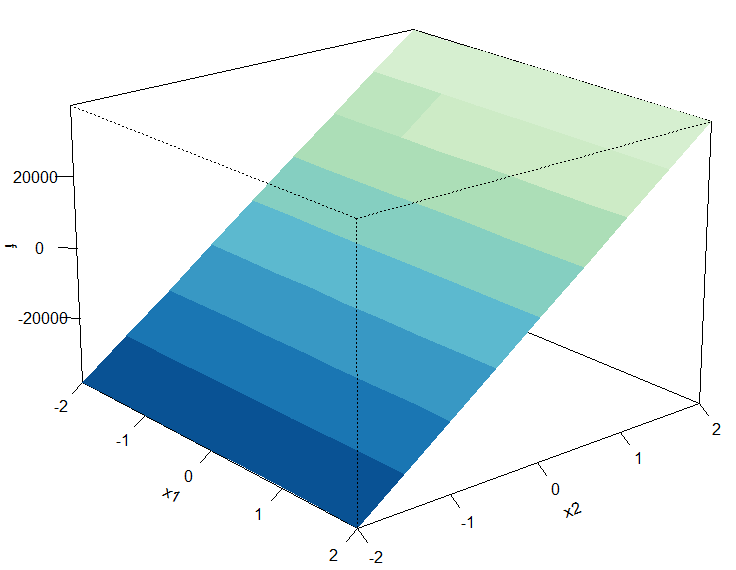
Gambar 5. 14 Hasil Outer

1. **Menampilkan Gambaran Perspektif Nilai X**



Gambar 5. 15 Function Persp

Fungsi persp3D ini berguna untuk menggambarkan plot perspektif nilai X sebelum dilakukannya proses optimasi. Pada tahap ini gambar yang dihasilkan berbentuk 3D. Berikut merupakan gambar yang dihasilkan.



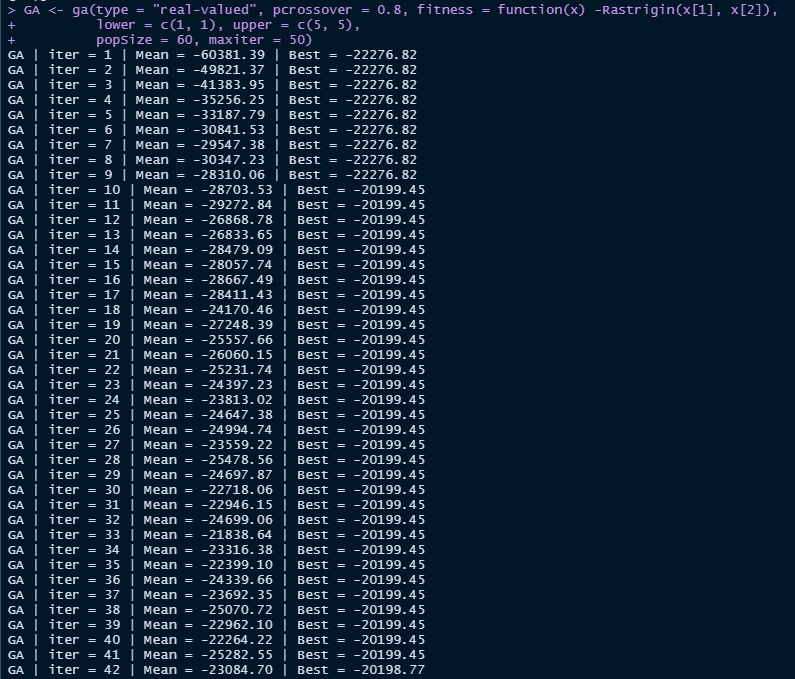
Gambar 5. 16 Hasil Dari Function Persp

1. **Optimasi dan Iterasi**

****

Gambar 5. 17 Proses Optimasi dan Iterasi

Pada penelitian ini jenis pengkodean yang akan digunakan ialah bilangan real karena cocok dengan penelitian ini yakni untuk optimasi nilai. *Pcrossover* berfungsi untuk memberika nilai probabilitas antara kromosom dengan nilai yang diberikan 0.8. *Fitness* ini berguna untuk menentukan solusi terbaik dari sekian solusi yang didapatkan dari proses optimasi. Untuk *lower* dan *upper* ini merupakan nilai batas bawah dan atas dalam proses optimasi. *Popsize* ini merupakan populasi yang dibangkitkan, penelitian ini membangkitkan populasi sebesar 60 dengan maksimal iterasi atau pengulangan sebanyak 50 kali.



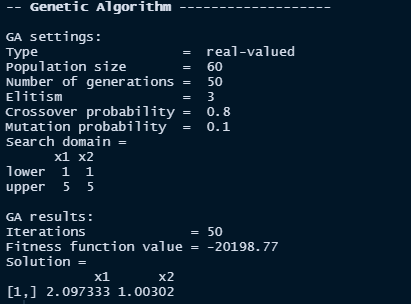
Gambar 5. 18 Proses Iterasi

1. **Summary**

****

Gambar 5. 19 Function Summary

Setelah proses optimasi dan iterasi, maka hasil yang didaptkan ialah sebagai berikut.



Gambar 5. 20 Hasil Dari Optimasi dan Iterasi

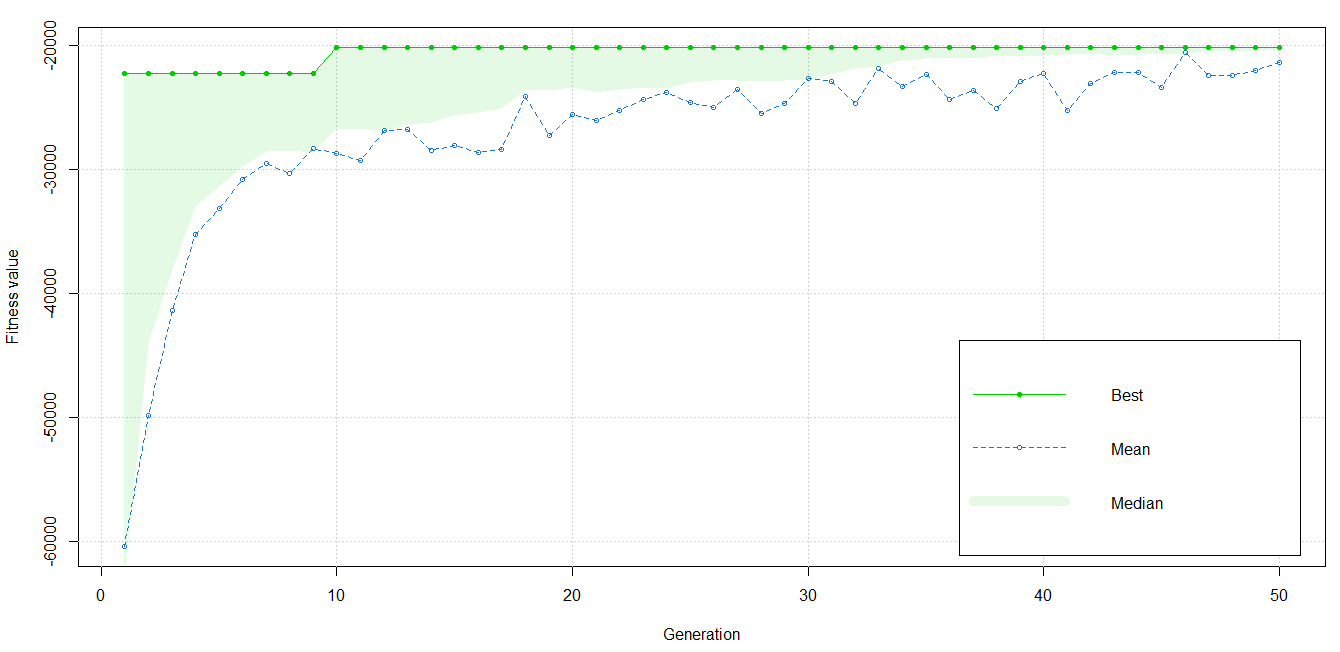
Dari proses iterasi sebanyak 50 kali, mendapatkan nilai solusi x1 = 2.097333 dan x2 = 1.003002. Untuk *elitsm* itu merupakan jumlah individu atau kromosom terbaik yang dapat bertahan pada setiap generasinya atau iterasi yang dilakukan. Penelitian ini kromosom yang dapat berahan dari 50 kali iterasi sebanyak 3 kromosom.

1. **Plot Hasil Optimasi dan Iterasi**

****

Gambar 5. 21 Function Plot

Setelah proses optimasi dan iterasi selesai, selanjutnya kita plot atau kita lihat dalam bentuk chart. Berikut merupakan hasil yang didapatkan.



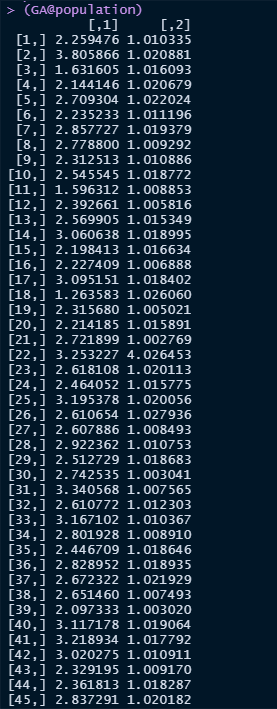
Gambar 5. 22 Hasil Plot

1. **Populasi Baru Terbentuk**

****

Gambar 5. 23 Function Untuk Menampilkan Keseluruhan Populasi

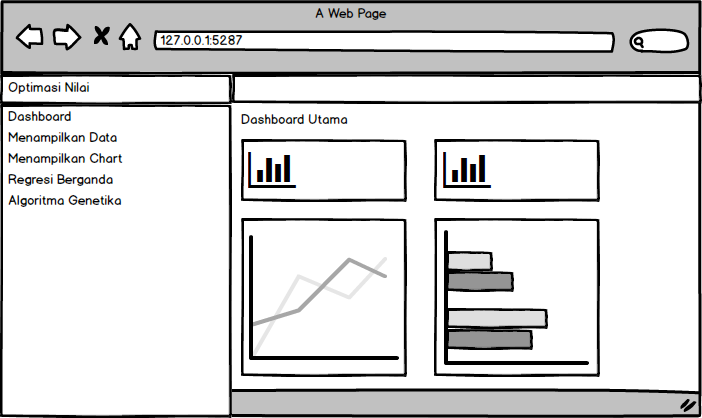
Setelah semua proses dilakukan, berikut merupakan hasil dari keseluruhan solusi yang dihasilkan pada proses optimasi dan iterasi.



Gambar 5. 24 Keseluruhan Nilai Populasi Yang Terbentuk

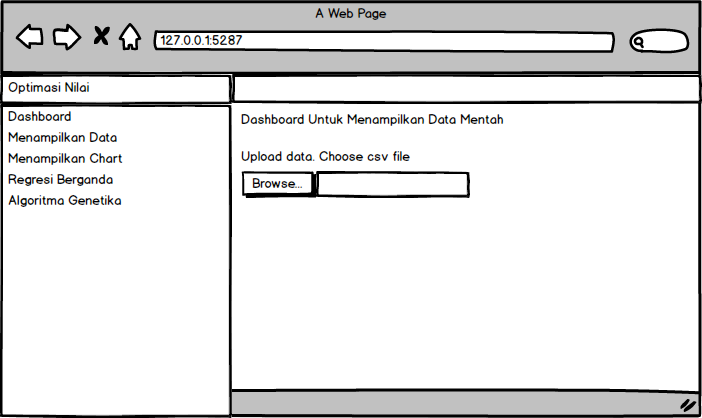
* 1. **Perancangan *User Interface***

1. *Dashboard* Utama



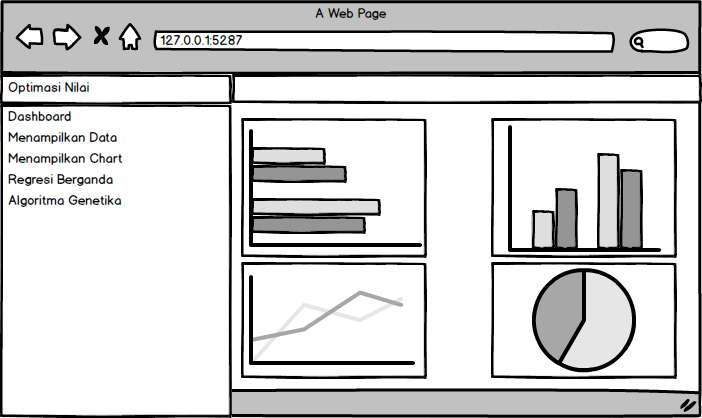
Gambar 5. 25 Dashboard Utama

1. Menampilkan Data



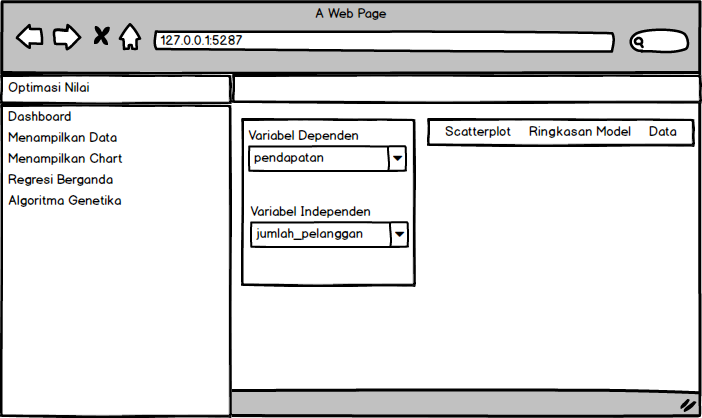
Gambar 5. 26 Halaman Menampilkan Data

1. Menampilkan Chart



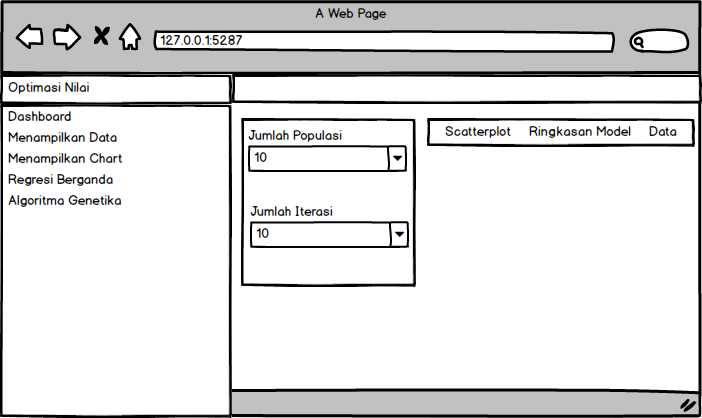
Gambar 5. 27 Halaman Menampilkan Chart

1. Regresi Berganda



Gambar 5. 28 Halaman Regresi Berganda

1. Algoritma Genetika

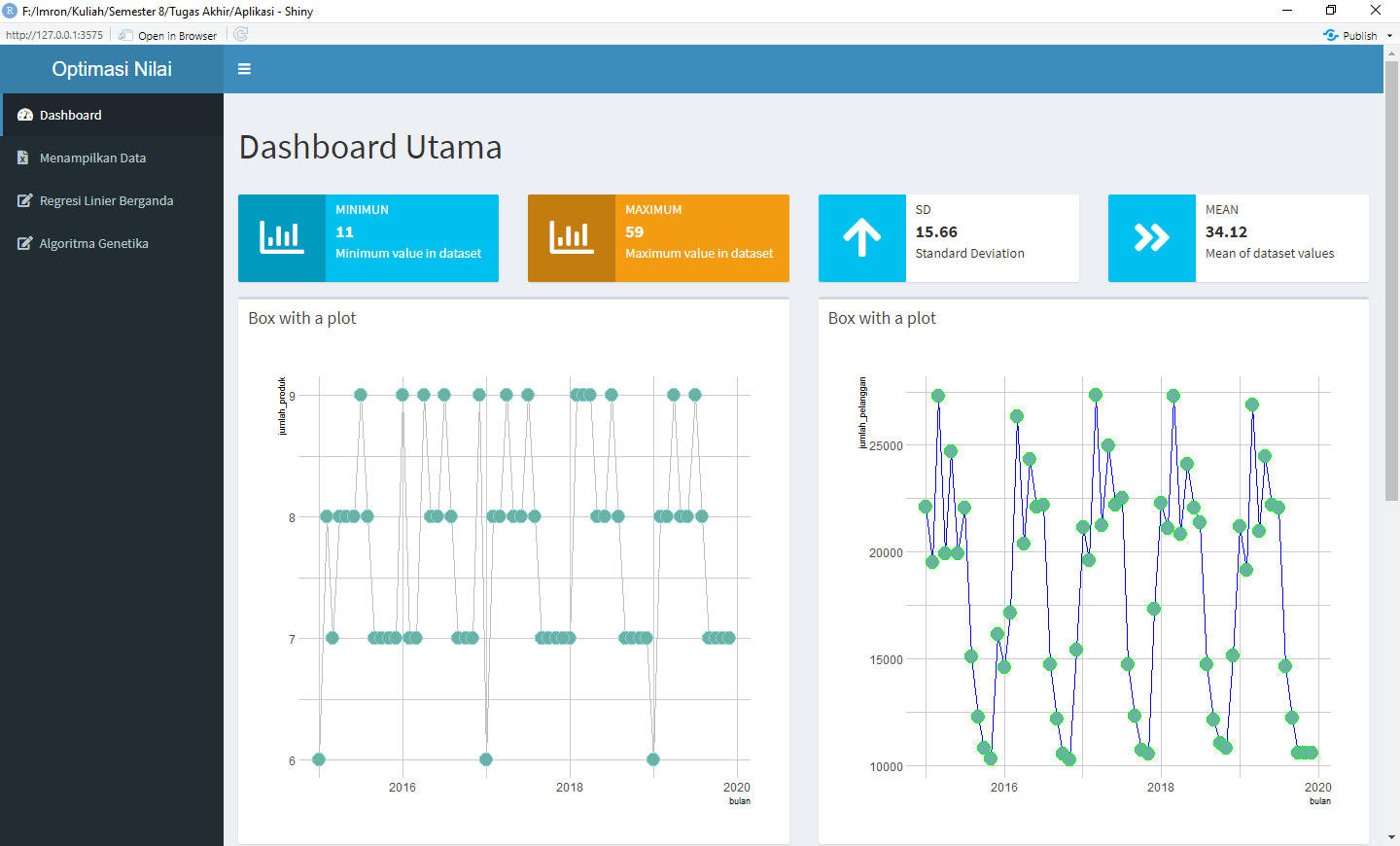


Gambar 5. 29 Halaman Algoritma Genetika

1. **Demo Aplikasi**

Ini merupakan tahap akhir dari penelitian. Setelah semua tahap sudah dilakukan, maka tahap selanjutnya adalah menjalankan aplikasi dan melihat apakah aplikasi dapat melakukan prediksi dan optimasi dengan benar. Untuk proses ini peneliti melakukan beberapa percobaan yaitu, apa dan apa.

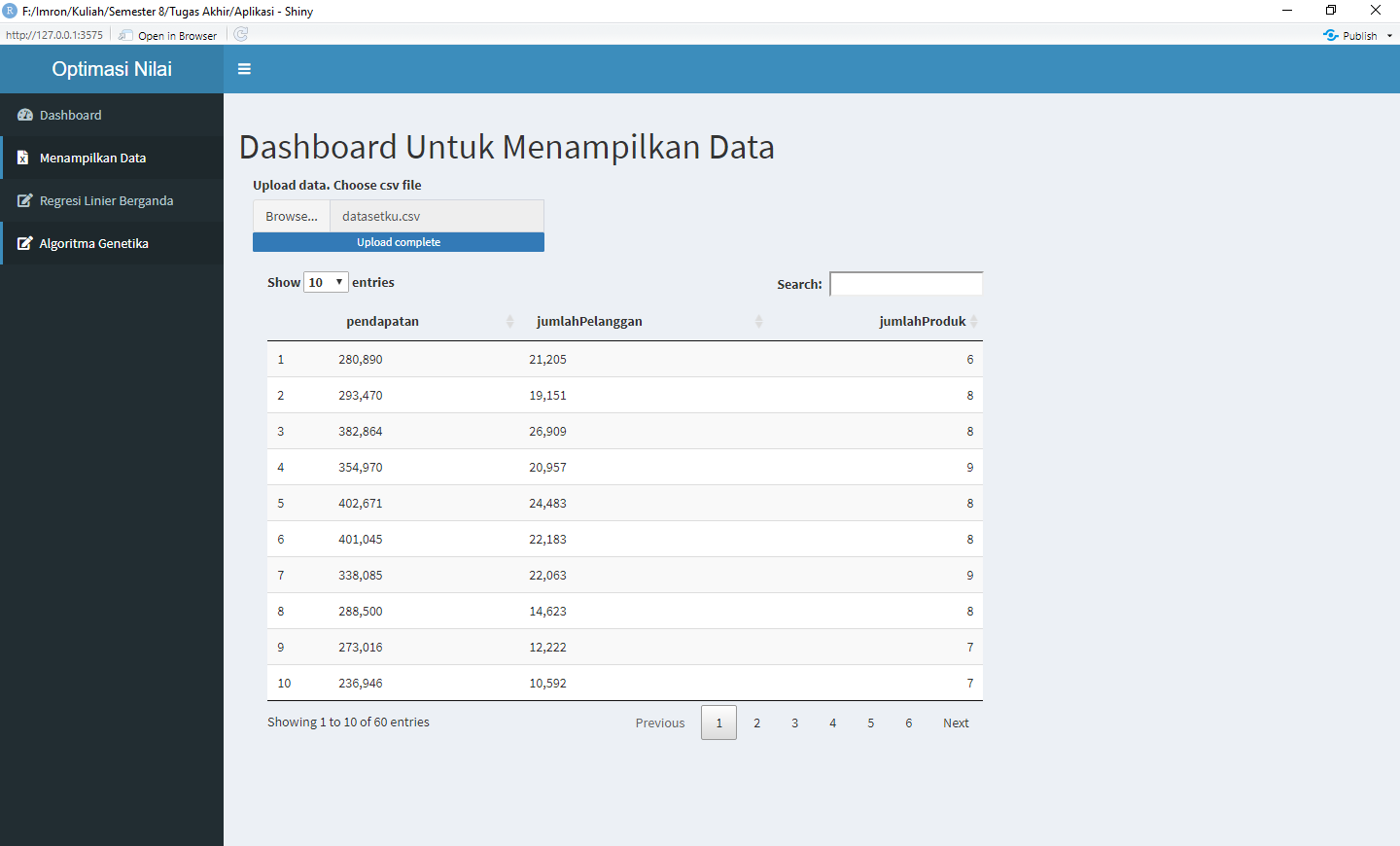
1. *Dashboard*



Gambar 5. 30 Dashboard Utama

Pada halaman utama ini terdiri dari *infoBox* yang berfungsi untuk menampilkan jumlah nilai minimum data, maksimum data untuk diunggah dalam aplikasi tersebut. Selanjutnya ada *boxplot* yang berfungsi untuk menampilkan grafik perkembangan data yang akan diolah. Untuk datanya sendiri terdiri dari Januari 2015 sampai dengan Desember 2019. Masing-masing *boxplot* tersebut menampilkan grafik dari variabel yang telah ditentukan.

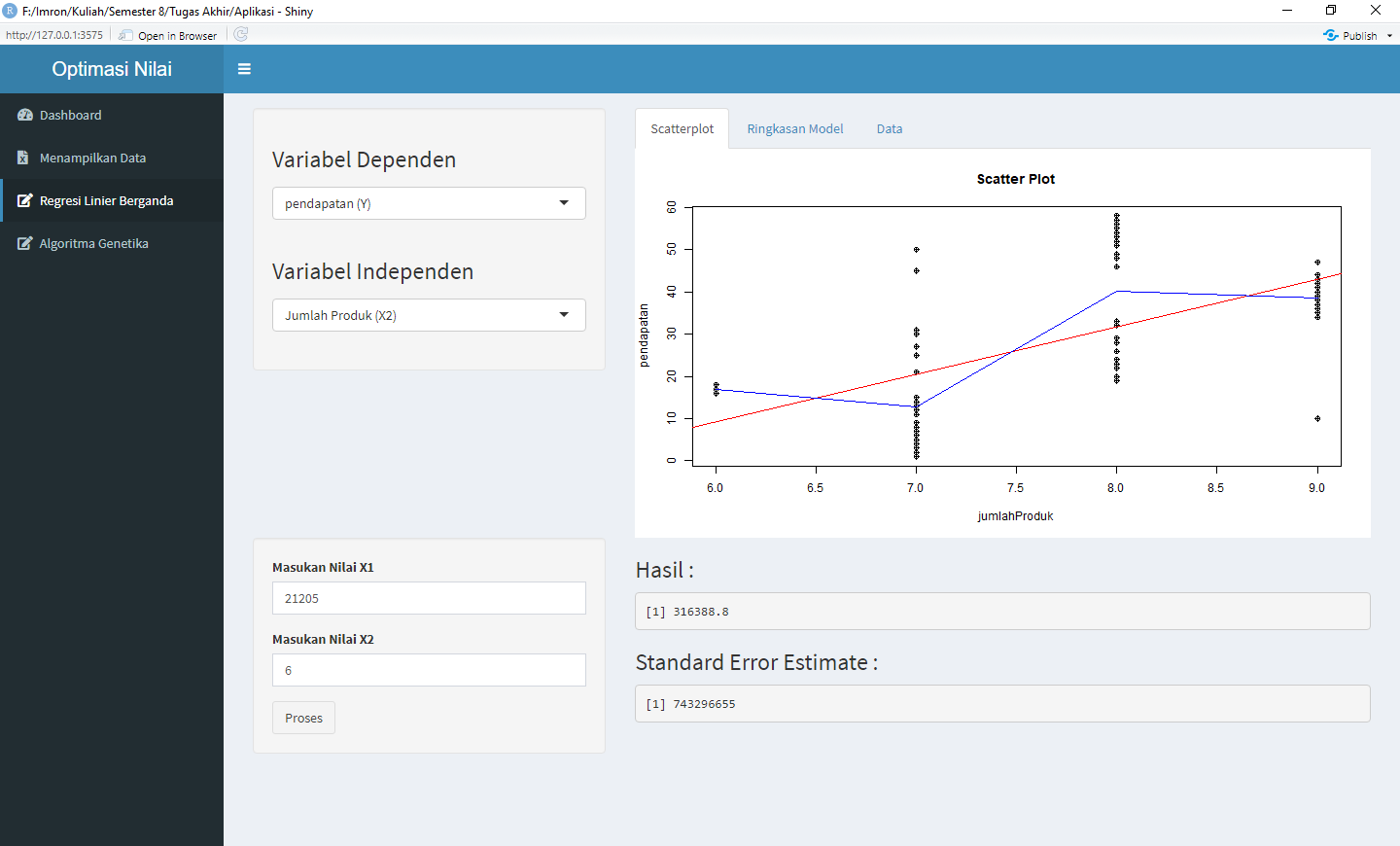
1. Menampilkan Data



Gambar 5. 31 Menampilkan Data

Pada halaman untuk menampilkan data ini berfungsi untuk menampilkan data yang akan digunakan dalam proses pengolahan data menggunakan Regresi Linier Berganda dan Algoritma Genetika. Pada halaman ini file yang dapat diunggah ialah file dengan ekstensi (csv). Selain itu pada halaman ini terdapat juga box untuk mencari data tertentu sehingga dapat memudahkan *user* dalam mencari data.

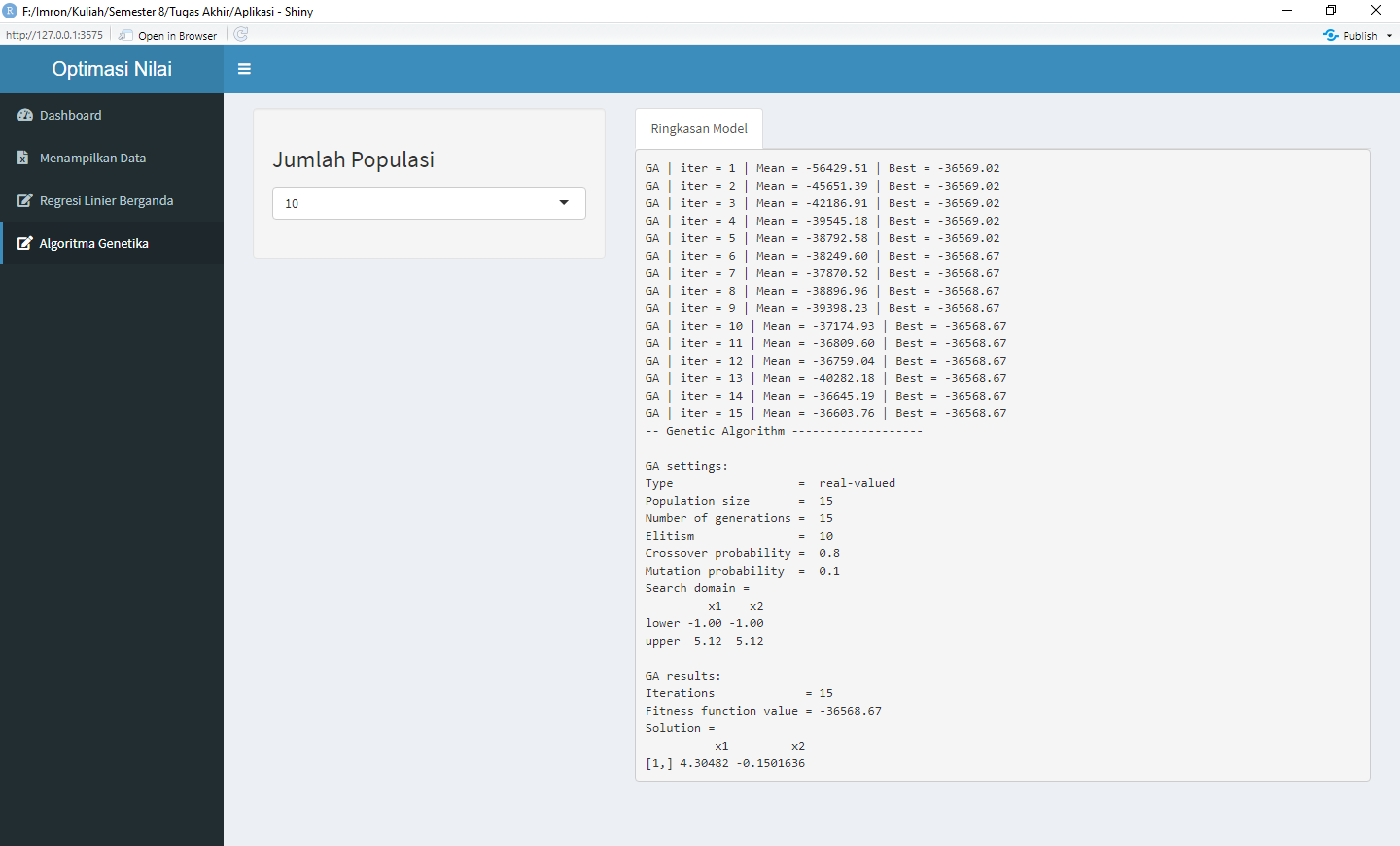
1. Regresi Linier Berganda



Gambar 5. 32 Halaman Regresi Linier Berganda

Pada halaman ini menampilkan proses pengolahan data menggunakan Regresi Linier Berganda, dapat dilihat bahwa *chart* yang ditampilkan dalam halaman tersebut terdiri dari variabel dependen (Y) dengan variabel independen (X1 / X2). Selain dapat menampilkan *chart* pada halaman ini juga dapat menampilkan hasil dari proses pengolahan data, yakni pada tab Ringkasan model. Tab tersebut memberikan informasi mengenai koefisien dari data, nilai *standard error* dari keseluruhan data, dan lain sebagainya. Untuk lebih meyakinkan mengenai prediksi yang dilakukan, peneliti menambahkan *textBox* untuk melakukan prediksi terkait data yang akan diprediksi. Pada contoh tersebut data bulan Januari 2015 akan diprediksi nilainya dan akan dibandingkan dengan nilai sebelum diprediksi.

1. Algoritma Genetika



Gambar 5. 33 Halaman Algoritma Genetika

Pada halaman ini menampilkan proses optimasi menggunakan Algoritma Genetika. Dapat dilihat bahwa Ringkasan Model yang tertera merupakan hasil dari pengolahan data dengan Algoritma Genetika. Selain itu kita dapat menentukan jumlah populasi yang akan dilakukan optimasi. Sehingga hasil akhir dari pengolahan ini dapat lebih baik dan dapat diimplementasikan kembali pada Regresi Linier Berganda dengan harapan nilai *standard error estimate*-nya lebih kecil dari data yang tidak dilakukan optimasi.

# **DAFTAR PUSTAKA**

|  |  |
| --- | --- |
| [1] | L. A. Muharom, A. F. Hadi and D. Anggraeni, "Rancang Bangun Data Warehouse dan R Studio Serta Pemanfaatannya dalam Peramalan Pola Konsumsi Masyarakat di Kabupaten Jember," *JUSTINDO,* vol. 1, no. 1, pp. 17-25, 2016. |