

**PENGOPTIMALAN PENGHITUNGAN KEUNTUNGAN PRODUKSI  
MENGUNAKAN ALGORITMA GENETIKA  
(STUDI KASUS : USAHA SABUN MAPACCING)**

**Syarkia Y<sup>1</sup>, Heliawaty Hamrul<sup>2</sup>, Nahya Nur<sup>3</sup>**

<sup>1,2,3</sup> Program Studi Informatika, Fakultas Teknik, Universitas Sulawesi Barat

Email: \*<sup>1</sup>kiayunus671@gmail.com, <sup>2</sup>heliawatyhamrul@unsulbar.ac.id, <sup>3</sup>nahya.nur@unsulbar.ac.id

**Abstrak**

*Perkembangan ilmu pengetahuan dan teknologi yang semakin maju di era sekarang memunculkan berbagai permasalahan baru pula. Hal tersebut tentu saja akan memberi tuntutan tersendiri pada para pelaku bisnis, sehingga dituntut untuk dapat mengambil langkah yang bijaksana dari pilihan-pilihan alternatif yang ada untuk mendukung keputusan agar tujuan perusahaan dapat tercapai. Adapun sampel yang akan menjadi tempat penelitian ini adalah perusahaan sabun mapaccing. Namun perusahaan ini tergolong masih baru, sehingga tentu saja perhitungan laba/kuntungan pada perusahaan ini masih menggunakan cara manual. Perhitungan dan analisis yang tepat sangat dibutuhkan untuk mendukung tercapainya keuntungan yang maksimal dari perusahaan. Maka dari itu, dibutuhkan sebuah yang digunakan untuk mendapatkan hasil optimasi keuntungan maksimum yang akurat dan efisien.. Penelitian ini akan menggunakan algoritma genetika. Dan pada pengujian ukuran populasi diperoleh populasi pada ke - 80 lah yang mempunyai rata-rata nilai fitness terbesar yaitu 57.84. Selanjutnya pengujian nilai CR-MR memperoleh hasil dimana nilai CR-MR yang terbaik yaitu masing-masing untuk CR = 0.3 dan MR = 0.7 dimana rata-rata fitness yang diperoleh yaitu 57.69*

**Kata kunci:** — Optimasi laba, Algoritma Genetika

**Abstract**

The development of science and technology that is increasingly advanced in the current era raises various new problems as well. This of course will put its own demands on business people, so they are required to be able to take wise steps from existing alternative choices to support decisions so that company goals can be achieved. The sample that will be the place of this research is the mapaccing soap company. However, this company is still relatively new, so of course the calculation of profit/profit in this company still uses the manual method. Accurate calculations and analysis are needed to support the achievement of maximum profits from the company. Therefore, we need a tool that is used to get maximum profit optimization results that are accurate and efficient. This research will use a genetic algorithm. And in the population size test, it was found that the 80th population had the largest average fitness value, namely 57.84. Furthermore, testing the CR-MR value obtained results where the best CR-MR value was for CR = 0.3 and MR = 0.7 where the average fitness obtained was 57.69

**Keywords:** — Profit optimization, Genetic Algorithm

## 1. PENDAHULUAN

Era teknologi yang semakin maju saat ini membuat banyak orang terbantu dalam menjalankan aktifitasnya, begitupun untuk perusahaan. Kehadiran teknologi ini diharapkan dapat membantu dalam mengembangkan atau memudahkan bisnisnya. Salah satu hal yang dapat kita lihat dengan adanya optimasi keuntungan untuk mendapatkan hasil utama dalam proses produksi, apalagi. Adanya perhitungan dan analisis yang tepat dibutuhkan untuk mendukung tercapainya keuntungan yang maksimal yang dapat diraih oleh sebuah perusahaan. Dimana kita ketahui bahwa adanya produksi yang berlebihan dapat merugikan sebuah perusahaan, begitupun jika produksi terlalu sedikit maka tidak dapat memenuhi permintaan konsumen. Maka dari itu dibutuhkan perencanaan produksi yang tepat untuk mencegah kerugian produksi pada perusahaan.

Dalam penelitian ini mengambil sampel perusahaan Zawaichi Sabun Mapaccing. perusahaan ini tergolong masih baru, sehingga tentu saja perhitungan laba/kuntungan pada perusahaan ini masih menggunakan cara manual dan berdasarkan asumsi saat memproduksi sabun Mapaccing. Cara seperti ini tentu saja tidak akan mampu memberikan hasil yang akurat dan cepat melihat banyaknya konsumen perusahaan tentu saja harus mencari solusi yang efektif untuk menangani masalah tersebut. Adanya optimasi keuntungan menjadi hasil utama dalam proses produksi pada perusahaan. Adanya perhitungan dan analisis yang tepat sangat menjadi hal yang sangat dibutuhkan untuk mendukung target capaian keuntungan yang maksimal yang dapat diraih oleh sebuah perusahaan atau industri. Semua perusahaan menginginkan adanya keseimbangan antara pengeluaran dan pemasukan, maka dibutuhkan perencanaan produksi yang tepat untuk mencegah kerugian produksi dalam perusahaan. Untuk mengatasi permasalahan tersebut dibutuhkan system yang dapat membantu para perusahaan dalam memperoleh optimasi keuntungan yang maksimum secara akurat untuk menghindari kelebihan produksi.

Salah satu algoritma yang dapat mengatasi masalah optimasi laba/rugi yaitu algoritma genetika, dimana algoritma ini mempunyai kemampuan dalam mengatasi berbagai permasalahan yang kompleks begitupun dalam kasus optimasi ini. Penelitian ini diharapkan membantu para pelaku usaha khususnya di perusahaan Zawaichi Sabun Mapaccing untuk memperoleh capaian keuntungan yang maksimal namun dengan pengeluaran yang seminimal mungkin pada proses produksi sabun Mapaccing.

## 2. TINJAUAN PUSTAKA

### 2.1 Optimasi

Proses optimasi atau dalam optimasi bahasa Inggris disebut *optimization* memiliki arti memaksimalkan atau meminimalkan sesuatu yang diberikan berdasarkan pada beberapa kendala-kendala yang terjadi. Secara lebih sederhana dapat dijelaskan bahwa optimisasi merupakan cabang dari ilmu matematika yang digunakan untuk mencari nilai maksimum atau meminim fungsi dengan mempertimbangkan beberapa kendala yang diberikan pada proses penyelesaiannya.

### 2.2 Algoritma Genetika

*John Holland* pada awal tahun 70-an memperkenalkan algoritma genetika, yang dimana algoritma ini adalah algoritma Stokastik berdasarkan prinsip genetika alami yang saat ini banyak digunakan atau diterapkan dalam pembelajaran mesin, termasuk untuk dalam kasus melakukan optimasi (*Guo, Wang, & Han, 2010*)

Algoritma genetika juga dikategorikan sebagai *search algorithm* yang dimana mengambil prinsip seleksi alam pada bidang ilmu genetika untuk mengembangkan solusi terhadap suatu permasalahan. Dengan mengelola suatu populasi individu yang menjadi prinsip dasar algoritma genetika, kemudia merepresentasikan kandidat solusi yang dijadikan sebagai sebuah solusi, yang

dimana solusi tersebut dievaluasi menggunakan fitness untuk mencari nilai terbaik yang akan dijadikan sebagai rekomendasi.

Adapun langkah-langkah penyelesaian algoritma genetika adalah :

1. Inisialisasi populasi

Proses Inisialisasi dilakukan dengan random untuk melakukan pengenalan terhadap populasi yang ada atau sederhananya pelabelan. Proses Inisialisasi akan menghasilkan kumpulan populasi awal dengan banyak kromosom yang diinginkan.

2. Evaluasi

Pada tahapan evolusi ini dilakukan proses perhitungan fitness dari masing-masing kromosom. Tahap evaluasi ini akan berlangsung secara terus menerus sampai tercapai tujuan atau kriteria.

3. Seleksi

Proses seleksi pada tahapan algoritma genetika dilakukan dengan mencari nilai terbaik yang dimana didapatkan pada generasi baru yang terdapat dalam kromosom sebelumnya.

4. Crossover

Crossover adalah proses persilangan antara dua kromosom yang ada, nantinya akan membentuk kromosom baru yang dimana diharapkan menjadi solusi terbaik.

5. Mutasi

Proses ini dilakukan dengan penambahan nilai acak yang sangat kecil dengan probabilitas rendah pada variabel keturunan. Proses mutasi ini akan melakukan penggantian suatu gen dengan gen yang baru yang dilakukan secara acak.

### 2.3 Python

Python adalah bahasa pemrograman yang populer saat ini, dengan dilengkapi berbagai library yang memudahkan pengguna dalam menuliskan kodenya. Python merupakan Bahasa yang mudah dipahami karena sintaks yang digunakan singkat. Hal ini membuat python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain.

### 2.4 Pengujian Sistem

Pada tahap pengujian ini akan dilakukan pengujian ukuran populasi untuk melihat perbandingan dari data yang di dapatkan dalam populasi yang dibuat menggunakan algoritma genetika. Setelah itu dilakukan pengujian dengan melakukan kombinasi nilai *Crossover Rate* (CR) dan *Mutation Rate* (MR). Dengan mencari kombinasi nilai CR dan MR terbaik yang dapat memberikan solusi optimal. Pada tahapan pengujian ini dilakukan pengambilan nilai rata-rata terhadap data yang di ujikan.

## 3. METODOLOGI PENELITIAN

### 3.1 Jenis Penelitian

Penelitian kuantitatif merupakan jenis penelitian yang dimana menekankan pada kesistematiskan, tersusun, serta terencana dengan jelas dari awal sampai akhir. Adapun defenisi lain mengatakan bahwa penelitian kuantitatif merupakan penelitian yang dimana menggunakan angka-angka dalam proses pembuktiannya.

### 3.2 Tahap Penelitian

Tahapan penelitian yang dilakukan dalam pengoptimalan keuntungan produksi di perusahaan sabun Mapaccing dengan mengimplementasikan algoritma genetika, adalah sebagai berikut:

3.2.1. Wawancara

Pengumpulan data merupakan teknik untuk mendapatkan data melalui tanya jawab kepada pihak terkait yang mendukung penelitian. Adapun data yang didapatkan pada penelitian ini adalah biaya produksi, harga jual, laba, permintaan tertinggi dan terendah pada itemnya. Narasumber dalam penelitian ini adalah bapak zakariah selaku karyawan di zawaichi sabun mapaccing. Zawaichi sabun mapaccing terletak di lingkungan lemosusu, kelurahan Limboro, Kecamatan Limboro, Kabupaten Polewali Mandar.

3.2.2. Analisis Data

Teknik analisis data pada penelitian ini menggunakan analisis data kuantitatif, teknik ini merupakan kegiatan mengolah data yang telah dikumpulkan baik dari responden atau sumber data-data lain yang berkaitan dengan penelitian yang dilakukan.

3.2.3. Pembuatan Program

Bahasa pemrograman yang akan digunakan adalah bahasa pemrograman python. Program ini akan dibuat sesuai dengan alir penyelesaian masalah dengan menggunakan algoritma genetika. Proses penyelesaian masalah ditunjukkan pada gambar 1.



Gambar 1. Alur algoritma genetika

3.2.4. Pengujian (testing)

Pada tahapan pengujian ukuran populasi dilakukan untuk melihat perbandingan dari data yang di dapatkan dalam populasi yang dibuat menggunakan algoritma genetika. Pada pengujian ini dilakukan pengujian dengan menggabungkan nilai CR dan MR, hasil dari pengujian ini diharapkan mendapatkan solusi terbaik dari permasalahan optimasi ini melalui proses kombinasi CR dan MR dalam proses pengujiannya, kemudian diambil nilai rata-ratanya.

#### 4. HASIL DAN PEMBAHASAN

Dari penjelasan diatas penulis telah melakukan implementasi perhitungan genetika kedalam bahasa pemograman Python yang dimana hasilnya sebagai berikut :

##### 4.1 Input dan Output Data

Data inputan merupakan data yang diambil dari usaha sabun Mapaccing yang menjadi tempat penelitian penulis, dimana data tersebut dapat dilihat melalui gambar dibawah :

```
In [2]: #mengambil data pada datauji_genetika.xlsx
dataUji = pd.read_excel('datauji.xlsx')
#menampilkan 5 data teratas
dataUji.head(9)
```

No	Jenis Produk	Ukuran	Biaya Produksi	Harga Jual Per Item	Laba Per Item	Permintaan Terendah'n(Dihitung Perbungkus)	Permintaan Tertinggi'n(Dihitung Perbungkus)
0	1.0	Produk A	5 liter	47000	60000	13000	200
1	NaN	NaN	1.5 liter	14400	25000	10600	100
2	NaN	NaN	600 ml	5760	10000	4240	50
3	2.0	Produk B	5 liter	6350	75000	68650	200
4	NaN	NaN	1.5 liter	19050	30000	10950	50
5	NaN	NaN	600 ml	7620	15000	7380	50
6	3.0	Produk C	5 liter	100000	150000	50000	200
7	NaN	NaN	1.5 liter	40050	55000	14950	50
8	NaN	NaN	600 ml	16020	25000	8980	50

Gambar 2 Data Inputan

##### 4.2 Reproduksi

pada tahapan ini menggunakan proses *crossover* dan *mutation* untuk mendapatkan keturunan baru dari induk yang telah ada sebelumnya. Pembentukan generasi baru atau reproduksi dapat dilihat melalui gambar dibawah.

Gambar 3. Proses Reproduksi

```
In [10]: populasi=[]
jumlahPopulasi = 100
populationClass(populasi, jumlahPopulasi, data0L.biayaProduksi.count(), data0L.minimalOrder, data0L.maksimalOrder)
```

```
In [11]: populasiAwal = pd.DataFrame(populasi, columns=["a1", "a2", "a3", "a4", "a5", "a6", "a7", "a8", "a9"])
populasiAwal.head()
```

	a1	a2	a3	a4	a5	a6	a7	a8	a9
0	246	136	63	339	60	50	383	85	53
1	221	132	77	408	86	70	317	77	60
2	225	127	88	329	75	69	236	89	52
3	285	135	95	295	97	88	407	85	53
4	272	110	50	461	52	87	443	58	65

##### 4.3 Menghitung Fitness

Individu akan melakukan evaluasi nilai fitness yang dimana setiap individu pada populasi harus memiliki nilai pembanding (*fitness cost*) untuk melihat solusi yang berada pada suatu

---

individu. Adanya nilai pembandingan ini didapatkan solusi terbaik dengan cara pengurutan nilai pembandingan dari individu-individu dalam suatu populasi. Solusi terbaik ini akan dipertahankan, sedangkan solusi-solusi yang lainnya akan di ubah-ubah untuk mendapatkan solusi yang lain dengan mengimplementasikan proses *crossover* dan *mutation*. Nilai *fitness* tersebut dapat dilihat pada gambar 4 serta tahapan *cros over* dan *mutation* masing-masing pada gambar 5 dan 6 berikut

```
In [12]: fitness=[]
c1=[]
c2=[]
totalLaba = []
fitnessClass(populasi,len(populasi),fitness,c1,c2)

In [13]: populasiAwal["totalLaba"]=totalLaba
populasiAwal["c1"]=c1
populasiAwal["c2"]=c2
populasiAwal["fitness"]=fitness
populasiAwal.head()

Out[13]:
```

	a1	a2	a3	a4	a5	a6	a7	a8	a9	totalLaba	c1	c2	fitness
0	246	136	63	339	60	50	383	85	53	50101760	0	0	50.10176
1	221	132	77	408	86	70	317	77	60	51606130	0	0	51.60613
2	225	127	88	329	75	69	236	89	52	42158150	0	0	42.15815
3	285	135	95	295	97	88	407	85	53	49598830	0	0	49.59883
4	272	110	50	461	52	87	443	58	65	61373910	0	0	61.37391

Gambar 4 Menghitung Nilai Fitness

```
In [14]: #melakukan crossover
childs = []
CR = 0.5
crossoverClass(childs,populasi,dataOL.biayaProduksi.count(),CR)

In [15]: populasiCrossover = pd.DataFrame(childs, columns=["a1","a2","a3","a4","a5","a6","a7","a8","a9"])
populasiCrossover.head()

Out[15]:
```

	a1	a2	a3	a4	a5	a6	a7	a8	a9
0	263	144	64	266	79	66	422	77	93
1	263	167	64	388	75	67	454	70	71
2	277	109	61	374	63	71	338	59	78
3	250	111	45	319	66	87	402	82	80
4	259	144	60	427	65	66	369	80	78

```
In [16]: fitness=[]
c1=[]
c2=[]
totalLaba = []
fitnessClass(childs,len(childs),fitness,c1,c2)
populasiCrossover["totalLaba"]=totalLaba
populasiCrossover["c1"]=c1
populasiCrossover["c2"]=c2
populasiCrossover["fitness"]=fitness
populasiCrossover.head()

Out[16]:
```

	a1	a2	a3	a4	a5	a6	a7	a8	a9	totalLaba	c1	c2	fitness
0	263	144	64	266	79	66	422	77	93	48264980	0	0	48.26498
1	263	167	64	388	75	67	454	70	71	57656250	0	0	57.65625
2	277	109	61	374	63	71	338	59	78	50388490	0	0	50.38849
3	250	111	45	319	66	87	402	82	80	49746210	0	0	49.74621
4	259	144	60	427	65	66	369	80	78	56335120	0	0	56.33512

Gambar 5. Proses Crossover

```
In [17]: #melakukan mutasi
parents = []
MR = 0.5
mutationClass(parents,populasi,MR)

In [18]: populasiMutasi = pd.DataFrame(parents, columns=["a1","a2","a3","a4","a5","a6","a7","a8","a9"])
populasiMutasi.head()

Out[18]:
```

	a1	a2	a3	a4	a5	a6	a7	a8	a9
0	225	101	62	235	90	90	448	79	54
1	246	136	63	339	60	50	383	85	53

```
In [19]: fitness=[]
c1=[]
c2=[]
totalLaba = []
fitnessClass(parents,len(parents),fitness,c1,c2)
populasiMutasi["totalLaba"]=totalLaba
populasiMutasi["c1"]=c1
populasiMutasi["c2"]=c2
populasiMutasi["fitness"]=fitness
populasiMutasi.head()

Out[19]:
```

	a1	a2	a3	a4	a5	a6	a7	a8	a9	totalLaba	c1	c2	fitness
0	225	101	62	235	90	90	448	79	54	46100900	0	0	46.10090
1	246	136	63	339	60	50	383	85	53	50101760	0	0	50.10176



Gambar 6. Proses Mutation

#### 4.4 Seleksi

Tahap ini dilakukan dengan pencarian nilai atau solusi terbaik dalam pencarian laba produksi ini. Untuk mencari solusi terbaik adalah dengan cara melihat nilai *fitness* terbesar itulah solusi terbaik.

```
In [22]: #melakukan proses seleksi
populasiGabungan = populasiGabungan.sort_values('fitness', ascending=False)
populasiGabungan
```

```
Out[22]:
```

	a1	a2	a3	a4	a5	a6	a7	a8	a9	totalLaba	c1	c2	fitness
79	292	143	81	499	90	98	446	79	88	63832120	0	0	63.83212
28	277	135	74	498	51	68	426	55	78	63416440	0	0	63.41644
16	201	126	60	487	70	52	435	83	77	62468120	0	0	62.46812
4	272	110	50	461	52	87	443	58	65	61373910	0	0	61.37391
15	294	107	70	439	70	85	448	53	80	60694900	0	0	60.69490
...	...	...	...	...	...	...	...	...	...	...	...	...	...
85	227	132	72	245	57	50	249	79	95	36952030	0	0	36.95203
37	267	137	58	246	52	91	220	98	67	36364780	0	0	36.36478
115	272	95	46	218	66	64	229	70	69	34014880	0	0	34.01488
14	252	105	53	232	95	63	205	57	50	33596880	0	0	33.59688
88	222	104	96	201	50	93	243	90	50	33372430	0	0	33.37243

152 rows x 13 columns

```
In [23]: populasiTerbaik = populasiGabungan[:5]
populasiTerbaik
```

```
Out[23]:
```

	a1	a2	a3	a4	a5	a6	a7	a8	a9	totalLaba	c1	c2	fitness
79	292	143	81	499	90	98	446	79	88	63832120	0	0	63.83212
28	277	135	74	498	51	68	426	55	78	63416440	0	0	63.41644
16	201	126	60	487	70	52	435	83	77	62468120	0	0	62.46812
4	272	110	50	461	52	87	443	58	65	61373910	0	0	61.37391
15	294	107	70	439	70	85	448	53	80	60694900	0	0	60.69490

Gambar 7. Proses Seleksi

Dari hasil seleksi tersebut kita melihat bahwa nilai *fitness* terbesar berada di populasi 79 dengan nilai *fitness* yang dihasilkan adalah 65925280 dengan total laba sebanyak 65925810.

#### 4.5 Pengujian Ukuran Populasi

Pada tahap ini dilakukan pengujian ukuran populasi untuk melihat rentang populasi yang mempunyai *fitness* yang terbaik. Adapun hasil pengujian ukuran populasi dapat dilihat pada gambar 8.



Gambar 8. Hasil Pengujian Ukuran Populasi

#### 4.6 Pengujian CR-MR

Untuk pengujian ini digunakan ukuran populasi yang menghasilkan nilai fitness terbaik pada tahapan uji coba ukuran populasi. Adapun nilai CR dan MR yang digunakan pada tahapan pengujian berada dalam rentang nilai antara 0 sampai 1. Setiap generasi akan dilakukan percobaan 10 kali kemudian akan dihitung nilai rata-rata fitness yang didapatkan. Setelah proses selesai maka kita akan mendapatkan nilai CR dan MR yang optimal. Adapun pengujian tersebut dapat dilihat melalui gambar dibawah.

Out[28]:

	iterasi 1	iterasi 2	iterasi 3	iterasi 4	iterasi 5	iterasi 6	iterasi 7	iterasi 8	iterasi 9	iterasi 10	Rata-rata Fitness	MR	CR
0	44.01688	40.73108	48.90995	49.47005	52.41030	60.85211	58.42893	47.18692	45.48123	49.77826	49.77826	0	1
1	49.82300	46.57072	40.78928	45.50336	43.75276	63.04503	54.67728	38.63704	52.62800	58.28064	58.28064	0.1	0.9
2	66.24340	58.59830	52.07687	48.40114	40.42048	53.21528	52.27803	47.73787	39.99166	42.56518	42.56518	0.2	0.8
3	59.84677	50.64589	56.53306	58.36854	58.53420	51.61870	35.57521	53.49276	51.06561	39.48027	39.48027	0.3	0.7
4	55.08497	52.70684	59.69620	47.24781	53.23207	45.38841	55.87009	65.33497	46.33434	56.08078	56.08078	0.4	0.6
5	46.29463	43.59809	50.08167	57.08497	45.79672	48.77673	62.09969	47.91986	36.38143	59.72849	59.72849	0.5	0.5
6	34.66727	50.58825	45.28151	63.17836	51.52041	38.92945	48.97527	53.97538	34.21520	36.16485	36.16485	0.6	0.4
7	59.72218	49.71457	53.23782	56.51655	40.36796	41.46894	42.52821	50.18020	47.22207	53.07811	53.07811	0.7	0.3
8	55.31955	60.02193	54.29955	58.29389	49.44819	45.21473	47.78579	37.95751	55.74570	50.74532	50.74532	0.8	0.2
9	52.06009	42.43741	50.70773	49.08993	46.14219	53.43854	41.19798	57.43207	61.97079	51.12424	51.12424	0.9	0.1
10	49.07913	53.30605	38.25604	49.04452	56.74328	49.70329	56.90592	63.88966	43.81389	57.23247	57.23247	1	0

Gambar 9. Pengujian CR-MR

Berdasarkan data pada gambar 4.1. diatas ditarik sebuah kesimpulan bahwa nilai terbesar terdapat pada CR = 0.2 dan MR = 0.8 dengan rata-rata *fitness* yang dihasilkan adalah 50.74532.

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil pengujian terhadap masalah yang di turunkan pada penelitian ini dengan mengimplementasikan algoritma genetika untuk memecahkan masalah optimasi laba produksi, yang dimana pada penelitian ini menggunakan 9 sampel data yang disimbolkan dengan a1, a2, a3, a4, a5, a6, a7, a8, a9 yang kemudian dijadikan sebuah kromosom yang dibatasi dengan angka maksimum serta minimum jumlah produksi pada masing-masing item.

Adapun hasil yang diperoleh dari pengujian tersebut bahwa penerapan algoritma genetika pada pencarian laba produksi telah berhasil dibuat dengan menggunakan sampel data dari sabun mapaccing. Algoritma genetika berkerja dengan melakukan inialisasi awal yang berisi representasi nilai kromosom. Selanjutnya proses reproduksi dengan melakukan Crossover dan mutasi menggunakan random mutation, dan terakhir tahap seleksi yaitu dengan mencari individu yang dimana memiliki fitness terbaik yang menjadi solusi terbaik. Dari proses genetika tersebut nilai fitness terbesar berada di populasi 79 dengan nilai fitness yang dihasilkan adalah 63.83 dengan total laba sebesar Rp. 63.832.120. Dan pada pengujian ukuran populasi diperoleh populasi pada ke - 80 lah yang mempunyai rata-rata nilai fitness terbesar yaitu 57.84. Selanjutnya pengujian nilai CR-MR memperoleh hasil dimana nilai CR-MR yang terbaik yaitu masing- masing untuk CR = 0.3 dan MR = 0.7 dimana rata-rata fitness yang diperoleh yaitu 57.69.

## 5.2 Saran

Adapun saran yang penulis berikan sekiranya dapat menjadi rekomendasi dan mempunyai manfaat kedepannya adalah sebagai berikut :

- Untuk penelitian selanjutnya diharapkan dapat dibuatkan User Interface untuk memudahkan dalam penggunaannya kepada user.
- Untuk penelitian selanjutnya juga diharapkan dapat menggunakan beberapa algoritma lain seperti algoritma Branch and Bound, algoritma Goal Programming atau yang lainnya untuk melihat Algoritma terbaik untuk kasus optimasi laba produksi.

## REFERENSI

- [1] Aditya, C., & Mahmudi, W. F. 2016. Optimasi Persediaan Baju Menggunakan Algoritma Genetika. *Prosiding Seminar Nasional Teknologi dan Rekayasa Informasi*.
- [2] Agung Mustika Rizki, W. F. 2017. Optimasi Multi Travelling Selesman Problem (M- TSP) Untuk Distribusi Produk Pada Home Industri Tekstil Dengan Algoritma Genetika. *Kumpulan Jurnal Ilmu Komputer (KLIK)*.
- [3] Ankur Kumar, R. S. 2014. A Comparative Study of the Various Genetic Approaches to Solve Multi-Objective Optimization Problems. *IEEE*.
- [4] Fitri Ayu, N. P. 2018. Perancangan Sistem Informasi Pengolahan Data Praktek Kerja Lapangan (PKL) Pada Devisi HUMAS PT. Pegadaian. *Jurnal Intra-Tech ISSN*.
- [5] Guo, P., Wang, X., Han, Y. 2010. The Enhanced Genetic Algorithms for the Optimization Design. *International Conference on Biomedical Engineering and Informatics*.
- [6] Kadarina, T.M., Hajar, M.H.I. 2019. Pengenalan Bahasa Pemrograman Python Menggunakan Aplikasi Games Untuk Siswa/I Di Wilayah Kembangan Utara. *Jurnal Abdi Masyarakat (JAM)*.

- 
- [7] Mustaqbal, M., Fidaus, R. F., & Rahmadi, H. 2015. Pengujian Aplikasi Menggunakan Black Box Testing Bondary Vale Analysis. *Jurnal Ilmiah Teknologi Informasi Terapan*.
- [8] Samaher, & Mahmudy, W. F. 2015. penerapan algoritma genetika untuk memaksimalkan laba produksi jilbab. *Journal of Environmental Engineering & Sustainable Technology*, (Online), (<https://jeest.uib.ac.id/index.php/jeest/article/view/22/47>, Diakses 12 Maret 2020)
- [9] Tominaga, Y., Okamoto, Y., & Wakao, S. 2013. Binary-Based Topology Optimization of Magnetostatic Shielding by a Hybrid Evolutionary Algorithm Combining Genetic Algorithm and Extended Compact Genetic Algorithm. *IEEE Transactions On*.