

BAB 4 PERANCANGAN

Dalam bab ini akan membahas perancangan pembuatan sistem untuk optimasi penyusunan barang dalam mobil *box*.

4.1 Formulasi Permasalahan

Permasalahan yang akan diselesaikan adalah permasalahan penyusunan barang dalam mobil *box* agar mendapatkan hasil yang optimal dalam penyusunan barang, yang artinya agar ruang dalam mobil *box* dapat terisi maksimal dan meminimalisir ruang kosong. Oleh karena itu, untuk mengatasi permasalahan tersebut diperlukan adanya sistem optimasi penyusunan barang dalam mobil *box* dengan algoritme genetika. Data yang digunakan dalam menyelesaikan permasalahan optimasi penyusunan barang adalah dengan menggunakan tabel data barang perusahaan dan tabel data transportasi (mobil *box*) perusahaan.

Data barang dibutuhkan untuk mengetahui berapa banyak barang yang akan disusun ke dalam mobil *box* serta berapa ukuran dari masing-masing barang. Barang yang akan disusun harus memiliki bentuk balok atau kubus. Parameter yang digunakan antara lain panjang barang, lebar barang, tinggi barang, berat barang dan rotasi barang. Parameter berat digunakan untuk membatasi agar berat barang yang dimasukkan tidak melebihi kapasitas berat mobil *box*. Parameter perotasian barang diperlukan sebagai batasan atau *constraint* untuk mengidentifikasi apakah barang tersebut bisa dirotasi atau tidak. Jika bernilai "tidak" maka barang hanya dapat dirotasi sebanyak satu variasi, jika bernilai "ya" maka barang dapat dirotasi sebanyak enam variasi. Data barang beserta parameternya dapat dilihat pada Tabel 4.1 (Safrina, 2015):

Tabel 4.1 Parameter Barang

No	Kode Barang	p_b (cm)	l_b (cm)	t_b (cm)	w_b (kg)	R
1	B1	48	52	105	95	Ya
2	B2	75	35	175	115	Ya
3	B3	40	45	195	45	Ya
4	B4	97	50	180	125	Tidak
5	B5	60	45	215	45	Tidak
6	B6	40	60	80	33	Ya
7	B7	40	50	40	51	Tidak
8	B8	80	80	170	57	Ya
9	B9	80	60	30	26	Ya
10	B10	15	8	23	54	Ya

Keterangan:

p_b = panjang barang (cm)

l_b = lebar barang (cm)

t_b = tinggi barang (cm)

w_b = berat barang (kg)

cm = centimeter

kg = kilogram

R = rotasi

Sedangkan untuk data mobil *box* diperlukan untuk mengetahui kapasitas ruangan dalam proses penyusunan barang dalam mobil *box*. Parameter mobil *box* yang digunakan antara lain panjang, lebar, tinggi dan beban optimal yang mampu ditampung oleh mobil *box*. Data mobil *box* dan parameternya dapat dilihat pada Tabel 4.2 (Safrina, 2015):

Tabel 4.2 Parameter Mobil Box

No.	Kode Kendaraan	p_k (cm)	l_k (cm)	t_k (cm)	w_k (kg)
1	A	459	170	210	450

Keterangan:

p_k = panjang kendaraan (cm)

l_k = lebar kendaraan (cm)

t_k = tinggi kendaraan (cm)

w_k = beban maksimal yang dapat ditampung kendaraan (kg)

cm = centimeter

kg = kilogram

Pada contoh kasus nyata dalam hal penyusunan barang dalam mobil *box*, terdapat dua versi cara penyusunan barang secara manual. Yaitu dengan mendata berat barang keseluruhan yang akan dimasukkan ke dalam mobil *box* dan kemudian menjumlah total semua berat barang. Jika berat total barang diperoleh melebihi beban maksimal mobil *box*, maka barang ada yang dikurangi lagi dan dihitung total lagi sampai berat total keseluruhan barang kurang dari kapasitas beban maksimal mobil *box*. Misalkan, data barang pada Tabel 4.1 dan data kendaraan pada Tabel 4.2:

Barang yang akan dimasukkan ke dalam mobil *box* yaitu barang dengan kode 1 sampai barang dengan kode 10. Sedangkan beban maksimal kendaraan = 450 kg. Kemudian jumlah total beban keseluruhan barang:

$$\begin{aligned}
\sum \text{ beban seluruh kendaraan} &= \text{beban barang ke 1} + \text{beban barang ke 2} + \dots + \\
&\quad \text{beban barang ke 10} \\
&= 95 + 115 + 45 + 125 + 45 + 33 + 51 + 57 + 26 + 54 \\
&= 646 \text{ kg}
\end{aligned}$$

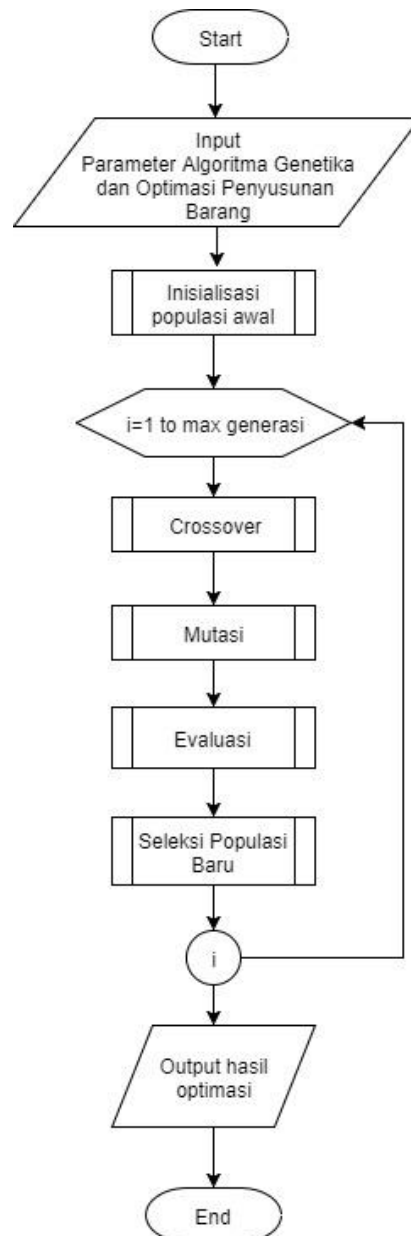
Karena beban maksimal barang hanya bisa menampung kurang dari 450 kg, maka daftar barang yang sudah dihitung totalnya, dihitung kembali dengan mengurangi beberapa barang, sampai akhirnya ketemu barang apa yang dibawa kurang dari beban maksimal kendaraan.

Dan cara manual yang kedua yaitu semua barang langsung dimasukkan ke dalam mobil box tanpa menghitung terlebih dahulu. Dapat ditandai dengan lengkungan di atas ban, jika sudah mulai melengkung, maka barang sudah berhenti dimasukkan. Lengkungan ban dapat berarti bahwa kendaraan tersebut telah mencapai beban maksimal.

Dalam cara manual lainnya, petugas akan mencoba untuk menyusun dan menggeser barang ketika berada langsung di lapangan. Tentu hal tersebut juga akan lebih banyak menguras tenaga dan waktu yang dibutuhkan. Dan barang-barang yang dimasukkan tersebut memiliki ukuran yang berbeda-beda. Jika barang dimasukkan secara sembarangan, ruang dalam kendaraan tidak maksimal, terdapat banyak ruang kosong. Bahkan membongkar kembali penyusunan barang juga pernah dilakukan agar mendapatkan penyusunan barang yang maksimal. Hal tersebut tentu menyita banyak waktu, dan kurang efisien.

4.2 Siklus Algoritme Genetika

Pada siklus algoritme genetika akan dibahas mengenai bagaimana proses penyelesaian masalah dalam optimasi penyusunan barang dalam mobil *box*. Terdapat beberapa tahapan dalam menyelesaikan suatu permasalahan dengan menggunakan Algoritme Genetika. Tahapan awal yaitu melakukan inisialisasi parameter awal yang telah diinputkan oleh user, lalu merandom populasi awal. Setelah populasi awal yang ditemukan sesuai dengan *popSize*, maka selanjutnya melakukan reproduksi berupa *crossover* dan mutasi. Hasil dari reproduksi tersebut yaitu nilai *fitness* dan kemudian diseleksi untuk mendapatkan populasi terpilih sebagai populasi selanjutnya. Kromosom yang terbaik yaitu yang memiliki nilai *fitness* terbesar. Tahapan tersebut dapat dilihat di Gambar 4.1.



Gambar 4.1 Flowchart Proses Algoritme Genetika

Proses optimasi penyusunan barang dengan algoritme genetika sesuai dengan Gambar 4.1 adalah sebagai berikut:

1. Inisialisasi Parameter Awal

Input data awal berupa data kendaraan dan data barang. Parameter yang digunakan yaitu:

- a. *popSize* (jumlah populasi)
- b. Jumlah Generasi
- c. *Crossover Rate* (*cr*)
- d. *Mutation rate* (*mr*)

2. Membangkit generasi awal yang terdiri dari beberapa individu secara acak.
3. Membuat generasi baru sesuai dengan jumlah generasi yang telah ditentukan dengan tahapan berikut:
 - a. Melakukan proses *crossover* dengan metode *partially mapped crossover*, jumlah *offspring* yang dihasilkan sebanyak *crossover rate* (*cr*) dikali dengan jumlah populasi awal.
 - b. Melakukan proses mutasi dengan metode *reciprocal exchange mutation*, yang menghasilkan *offspring* sesuai dengan *mutation rate* dikali dengan jumlah populasi awal.
 - c. Melakukan proses evaluasi perhitungan *fitness* untuk menentukan individu yang terbaik dengan nilai *fitness* terbesar yang kemudian akan digunakan sebagai populasi pada generasi berikutnya.
4. Kondisi berhenti ketika semua hal tersebut telah terpenuhi dan hasil akhir terdapat kromosom terbaik pada generasi akhir.

4.3 Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika

Langkah penyelesaian masalah optimasi penyusunan barang dalam mobil *box* telah dijelaskan dalam Gambar 4.1. Inisialisasi parameter awal yang digunakan meliputi:

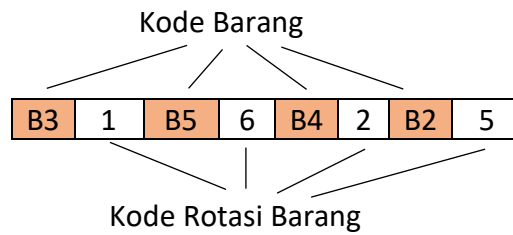
- a. *popSize* (jumlah populasi) = 3
- b. *Crossover rate* (*cr*) = 0,8
- c. *Mutation Rate* (*mr*) = 0,6
- d. Jumlah Generasi = 1

Berdasarkan parameter Algoritme Genetika di atas, *popSize* merupakan sebuah nilai yang menggambarkan banyaknya individu yang akan ditampung dalam populasi, *crossover rate* yaitu nilai yang menggambarkan jumlah *offspring* yang kemudian akan dihasilkan pada proses *crossover*. Banyaknya *offspring* ditentukan dari hasil kali *crossover rate* dengan *popSize* ($cr \times popSize$). Peluang Mutasi (*mutation rate*) yaitu nilai yang menggambarkan jumlah *offspring* yang dihasilkan dalam proses mutasi terhadap populasi. Banyaknya jumlah *offspring* didapatkan dari hasil kali *mutation rate* dengan *popSize* ($mr \times popSize$). Sedangkan untuk jumlah generasi yaitu nilai yang menggambarkan banyaknya generasi atau perulangan dalam proses algoritme genetika.

4.3.1 Representasi Kromosom dan Perhitungan *Fitness*

Representasi kromosom yang digunakan untuk optimasi penyusunan barang dalam mobil *box* yaitu representasi *integer* atau data angka. Dalam hal ini populasi merupakan sebuah *string* yang berisi gen yang berjumlah sesuai dengan jumlah barang pada masing-masing kategori atau segmen. Karena jumlah panjang gen terdapat 2 karakter yaitu kode barang dan rotasi barang, maka panjang *string* adalah $2 \times n$, yang menunjukkan bahwa n merupakan jumlah barang yang akan

dimasukkan ke dalam kendaraan. Berikut visualisasi individu dalam populasi yang terdapat pada Gambar 4.2:



Gambar 4.2 Visualisasi Individu dalam Populasi

Berikut contoh representasi kromosom:

Tabel 4.3 Representasi Kromosom

Individu	Segmen 1		Segmen 2			Segmen 10	
	x1	x2	x3	x4	x19	x20
	k	r	K	r	k	r
P1	B3	1	B5	6	B1	5
P2	B5	6	B2	5	B7	1
P3	B2	5	B10	4	B4	2

Tabel 4.3 berisi contoh representasi kromosom yang digunakan dalam perhitungan ini. Panjang kromosom yang terbentuk sebanyak 20 gen yang terdiri dari 10 segmen berdasarkan jumlah barang yang akan dimasukkan. Nilai pada gen pertama masing-masing segmen sesuai dengan kode barang (k merupakan kode barang) sesuai dengan banyaknya barang yang akan dimasukkan dan nilai pada gen kedua masing-masing segmen menunjukkan batasan rotasi barang (r) yang nilainya dari 1 sampai 6 sesuai dengan variasi barang pada Gambar 2.6. Pada tabel tersebut terdapat 3 individu yaitu P1, P2 dan P3. Banyaknya individu sesuai dengan *popSize* yang dimasukkan.

Nilai *fitness* dari masing-masing kandidat solusi atau individu diperoleh dengan cara menghitung total volume barang yang dapat disusun dalam mobil *box* berdasarkan dengan urutan masuk barang (Safrina, 2015). Pada tabel 4.3, dapat dilihat bahwa kandidat solusi 1 atau individu 1 memiliki urutan penyusunan barang dari barang pertama sampai terakhir memiliki kode barang ke 3-5-4-2-6-8-10-7. Berikut merupakan langkah-langkah perhitungan nilai *fitness* masing-masing individu. Dari tabel representasi kromosom pada Tabel 4.3, kemudian dari tabel representasi kromosom tersebut menghitung *fitness* sesuai dengan Persamaan 2.7. Berikut adalah permisalan hasil perhitungan solusi:

$$\begin{aligned}
 f(x) &= [\text{seluruh volume barang yang masuk/volume kendaran}] * 100\% \\
 &= [(40 * 45 * 195 * 1) + (60 * 45 * 215 * 0) + (97 * 50 * 180 * 1) + (75 * 35 * 175 * 0) + \\
 &\quad (40 * 60 * 80 * 1) + (80 * 80 * 170 * 1) + (15 * 8 * 23 * 1) + (40 * 50 * 40 * 1) + \\
 &\quad (80 * 60 * 30 * 0) + (48 * 52 * 105 * 1) / 16.386.300] * 100\% \\
 &= (2.848.840 / 16.386.300) * 100\%
 \end{aligned}$$

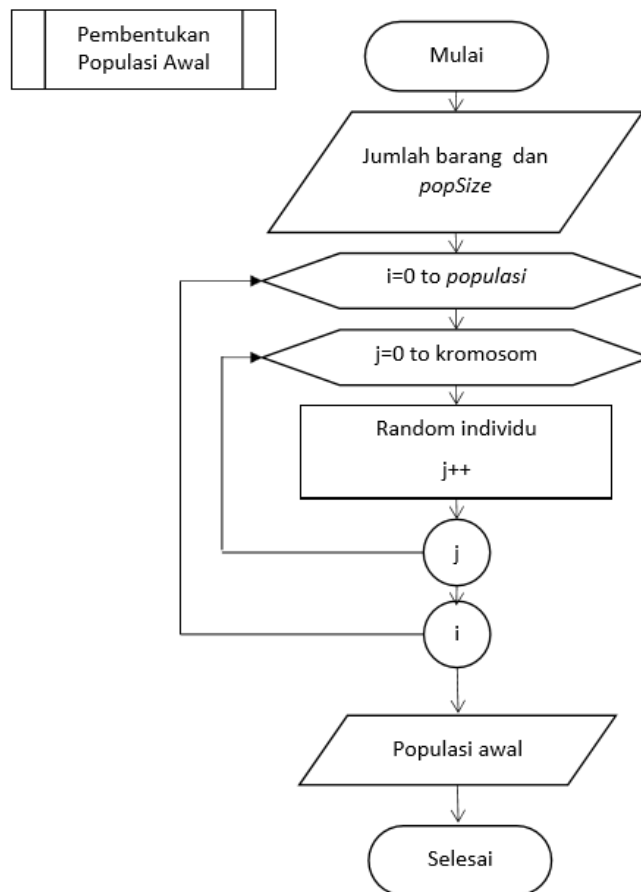
$$= 0.174 * 100\%$$

$$= 0.174$$

Dari perhitungan di atas dapat dilihat bahwa perhitungan *fitness* disesuaikan dengan urutan masuk barang, kemudian dicari dimensi dari masing-masing barang dan dicari volumenya. Kemudian untuk nilai 0 atau 1 dimensi terakhir pada masing – masing barang menunjukkan bahwa angka 0 diibaratkan jika barang tersebut tidak dimasukkan ke dalam mobil box, sedangkan angka 1 menunjukkan bahwa barang tersebut diibaratkan masuk ke dalam mobil box. Kemudian dibagi dengan volume kendaraan. Sehingga, semakin banyak volume barang yang masuk, maka akan semakin besar *fitness* yang diperoleh.

4.3.2 Inisialisasi Populasi Awal

Tahap inisialisasi populasi awal dilakukan dengan cara membangkitkan bilangan acak sebanyak *popSize* yang telah ditentukan. Jumlah kromosom sesuai dengan jumlah barang yang telah dimasukkan. Langkah-langkah pembentukan populasi awal dijelaskan dalam Gambar 4.3 berikut:



Gambar 4.3 Flowchart Pembentukan Populasi Awal

Berdasarkan *flowchart* pada Gambar 4.3, langkah dalam pembentukan populasi awal yaitu:

1. Masukkan jumlah barang dan banyaknya populasi.
2. Setelah itu sistem secara acak akan menentukan nilai gen dengan range yang telah ditentukan.
3. Sistem melakukan perulangan sebanyak jumlah barang yang dimasukkan dan sebanyak populasi yang ditentukan.

Berikut ini terdapat tiga individu yang dibangkitkan sebagai populasi awal yang ditunjukkan dalam Tabel 4.4:

Tabel 4.4 Populasi Awal

Indv.	Kromosom																			
	P1	B3	1	B5	6	B4	2	B2	5	B10	4	B7	1	B9	2	B1
P2	B5	6	B2	5	B8	2	B3	1	B4	2	B9	2	B6	3	B7	1
P3	B2	5	B10	4	B1	5	B7	1	B8	2	B6	3	B3	1	B4	2

P1, P2 dan P3 merupakan individu-individu yang terbentuk dalam tahap inialisasi awal dengan masing-masing terdiri atas 20 gen. Sedangkan nilai dari masing-masing gen seperti yang telah dijelaskan dalam Gambar 4.2.

4.3.3 Reproduksi

Reproduksi dilakukan agar dapat menghasilkan keturunan baru yang ada dalam populasi, yang kemudian akan ditempatkan di penampungan *offspring*. Operator genetika yang digunakan dalam proses reproduksi yaitu tukar silang (*crossover*) dan mutasi (*mutation*) (Mahmudy, 2015). Metode proses *crossover* yang digunakan adalah metode *partially mapped crossover*, sedangkan dalam proses mutasi menggunakan metode *reciprocal exchange mutation*.

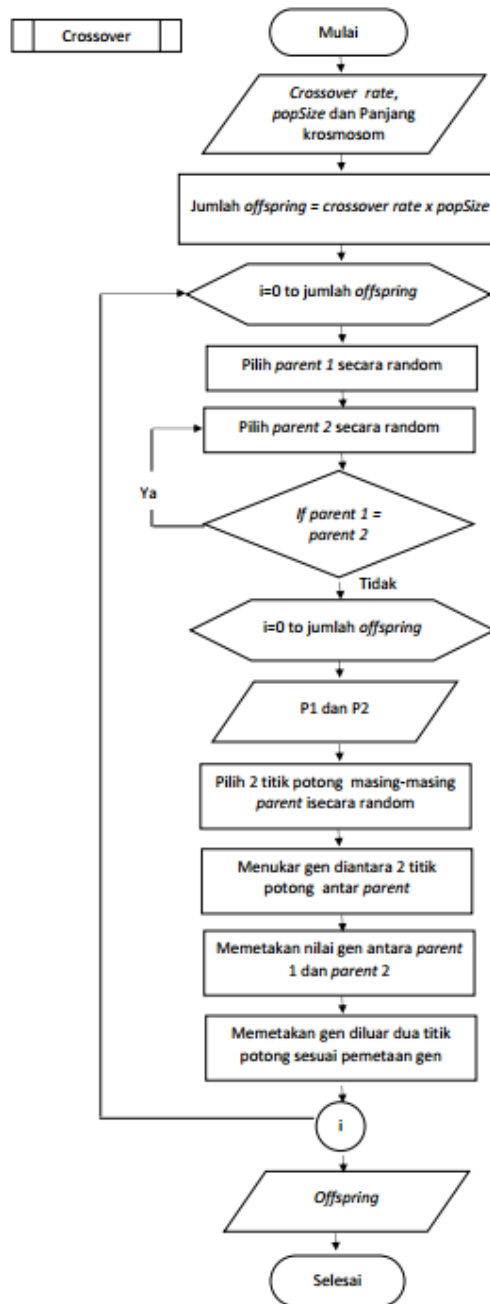
4.3.3.1 Perhitungan Crossover

Metode *partially mapped crossover* menghasilkan *offspring* dari dua induk yang berkombinasi secara *random*. Jumlah *offspring* didapat dari hasil perkalian antara *crossover rate* dengan *popSize* (populasi) yang telah diinputkan user. Diagram alir proses crossover terdapat dalam Gambar 4.4.

Misalkan pada perhitungan manual ini telah ditentukan nilai Cr adalah 0,4. Maka nilai *offspring* yang diperoleh sebagai berikut

$$offspring = 0,8 \times 3 = 2,4$$

Hasil perkalian tersebut dibulatkan menjadi 2, sehingga akan didapatkan sebanyak 2 *offspring*.



Gambar 4.4 Flowchart Proses Crossover

Pada proses *crossover* dalam Gambar 4.4, pertama menentukan dahulu jumlah *offspring* berdasarkan *crossover rate* dengan jumlah populasi. Menentukan *parent* 1 dan *parent* 2 secara random. Kemudian menentukan dua titik potong (*substrings / S*) secara acak. Selanjutnya dilakukan proses tukar silang gen diantara dua titik potong antar parent. Setelah ditukar silang dilakukan proses pemetaan nilai gen antar dua titik potong. Sehingga hasil pemetaan yang didapatkan digunakan untuk memetakan nilai gen diluar dua titik potong, agar dihasilkan nilai gen yang tidak sama. Maka akan dihasilkan dua *offspring* (C_1 dan C_2) seperti pada Tabel 4.5:

Langkah 1: potong 2 substring pada masing-masing induk dengan posisi yang sama

					S ↓									S ↓						
P1	B3	1	B5	6	B4	2	B2	5	B6	3	B8	2	B10	4	B7	1	B9	2	B1	5
P2	B5	6	B2	5	B8	2	B3	1	B10	4	B1	5	B4	2	B9	2	B6	3	B7	1

Induk

Langkah 2: tukar induk antara dua substring yang terpilih

P1	B3	1	B5	6	B8	2	B3	1	B10	4	B1	5	B4	2	B7	1	B9	2	B1	5
P2	B5	6	B2	5	B4	2	B2	5	B6	3	B8	2	B10	4	B9	2	B6	3	B7	1

Langkah 3: tentukan hubungan pemetaan berdasarkan substring yang terpilih

B8	2	B3	1	B10	4	B1	5	B4	2
↕		↕		↕		↕		↕	
B4	2	B2	5	B6	3	B8	2	B10	4

Hasil pemetaan:

1 ↔ 8 ↔ 4 ↔ 10 ↔ 6

3 ↔ 2

Langkah 4: menentukan kromosom turunan sesuai dengan hubungan pemetaan

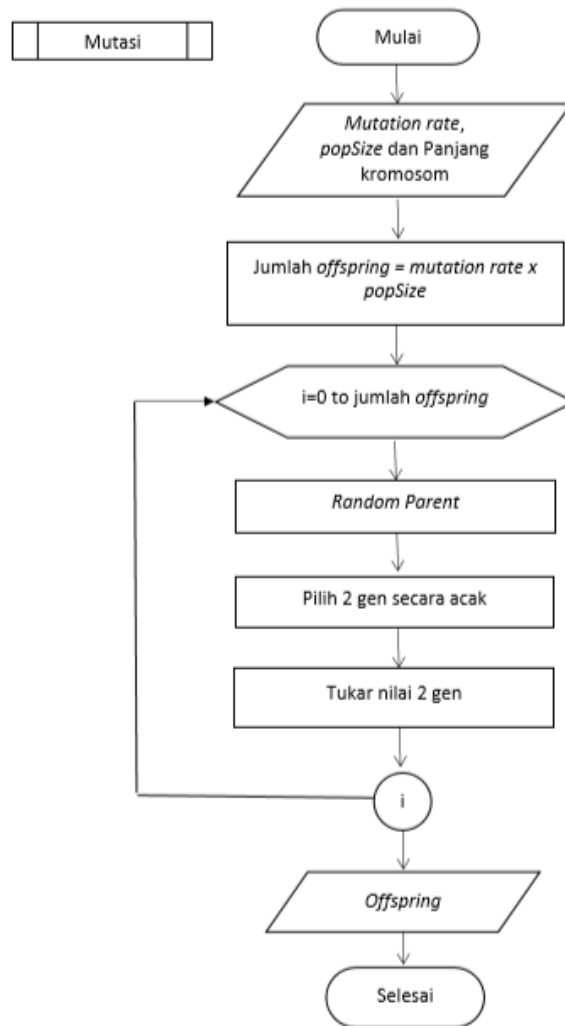
C1	B2	1	B5	6	B8	2	B3	1	B10	4	B1	5	B4	2	B7	1	B9	2	B6	5
C2	B5	6	B3	5	B4	2	B2	5	B6	3	B8	2	B10	4	B9	2	B1	3	B7	1

Tabel 4.5 Hasil Proses Crossover

P1	B3	1	B5	6	B4	2	B2	5	B6	3	B8	2	B10	4	B7	1	B9	2	B1	5
P2	B5	6	B2	5	B8	2	B3	1	B10	4	B1	5	B4	2	B9	2	B6	3	B7	1
C1	B2	1	B5	6	B8	2	B3	1	B10	4	B1	5	B4	2	B7	1	B9	2	B6	5
C2	B5	6	B3	5	B4	2	B2	5	B6	3	B8	2	B10	4	B9	2	B1	3	B7	1

4.3.3.2 Perhitungan Mutasi

Proses mutasi yang digunakan pada penelitian ini adalah metode *reciprocal exchange mutation* yang akan dilakukan pada setiap segmen. Jumlah individu baru atau *offspring* sesuai dengan input *mutation rate* dikali dengan *popSize*. Proses pembentukan mutasi terdapat dalam Gambar 4.5.



Gambar 4.5 Flowchart Proses Mutasi

Berdasarkan Gambar 4.5, proses mutasi dilakukan dengan menentukan jumlah *offspring* dengan cara mengkalikan jumlah populasi dengan *mutation rate*. Kemudian memilih induk secara acak. Selanjutnya memilih dua gen secara acak, dan dilakukan proses tukar gen antar induk. Dilakukan proses perulangan sampai *offspring*. Banyaknya *offspring* yang dihasilkan pada proses mutasi dengan cara sebagai berikut.

$$offspring = mr \times popSize = 0,6 \times 3 = 1,8$$

Nilai tersebut kemudian dibulatkan menjadi 2, yang berarti ada 2 *offspring*. Karena terdapat 2 *offspring*, maka diperlukan empat induk untuk mendapatkan 2 anak. Dipilih dua *parent* (induk) secara *random*, misalkan P1 dan P3. Proses mutasi awal yang dilakukan adalah mengambil 2 angka secara acak, kemudian ditukar yang dapat dilihat pada Tabel 4.6.

Tabel 4.6 Proses Mutasi Ke-1

P1	B3	1	B5	6	B4	2	B2	5	B6	3	B8	2	B10	4	B7	1	B9	2	B1	5
C3	B6	3	B5	6	B4	2	B2	5	B3	1	B8	2	B10	4	B7	1	B9	2	B1	5

Maka proses mutasi yang kedua dapat dilihat pada Tabel 4.7.

Tabel 4.7 Proses Mutasi Ke-2

P3	B2	5	B10	4	B1	5	B7	1	B5	6	B9	2	B8	2	B6	3	B3	1	B4	2
C4	B2	5	B6	3	B1	5	B7	1	B5	6	B9	2	B8	2	B10	4	B3	1	B4	2

4.3.4 Evaluasi dan Seleksi

Setelah proses *crossover* dan mutasi selesai, maka selanjutnya yaitu melakukan proses evaluasi dengan cara menghimpun semua individu yaitu *parent* dan *offspring* dan melakukan perhitungan *fitness* tiap individu. Semakin besar nilai *fitness* suatu individu maka semakin besar peluang untuk terpilih. Perhitungan *fitness* sesuai dengan Persamaan 2.7. *Parent* dan *child* dapat dilihat pada Tabel 4.8.

Tabel 4.8 Parent dan child

P1	B3	1	B5	6	B4	2	B2	5	B6	3	B8	2	B10	4	B7	1	B9	2	B1	5
P2	B5	6	B2	5	B8	2	B3	1	B10	4	B1	5	B4	2	B9	2	B6	3	B7	1
P3	B2	5	B10	4	B1	5	B7	1	B5	6	B9	2	B8	2	B6	3	B3	1	B4	2
C1	B2	1	B5	6	B8	2	B3	1	B10	4	B1	5	B4	2	B7	1	B9	2	B6	5
C2	B5	6	B3	5	B4	2	B2	5	B6	3	B8	2	B10	4	B9	2	B1	3	B7	1
C3	B6	3	B5	6	B4	2	B2	5	B3	1	B8	2	B10	4	B7	1	B9	2	B1	5
C4	B2	5	B6	3	B1	5	B7	1	B5	6	B9	2	B8	2	B10	4	B3	1	B4	2

Pada Tabel 4.8 masing-masing calon solusi atau *child* dapat diketahui urutan-urutan barang yang akan disusun. Secara ringkas, urutan-urutan barang yang akan dimasukkan dari awal sampai akhir dapat dilihat pada Tabel 4.8.

Tabel 4.9 Posisi Urutan Masuk Barang

	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	p ₉	p ₁₀
P1	3	5	4	2	6	8	10	7	9	1
P2	5	2	8	3	10	1	4	9	6	7
P3	2	10	1	7	5	9	8	6	3	4
C1	2	5	8	3	10	1	4	7	9	6
C2	5	3	4	2	6	8	10	9	1	7
C3	6	5	4	2	3	8	10	7	9	1
C4	2	6	1	7	5	9	8	10	3	4

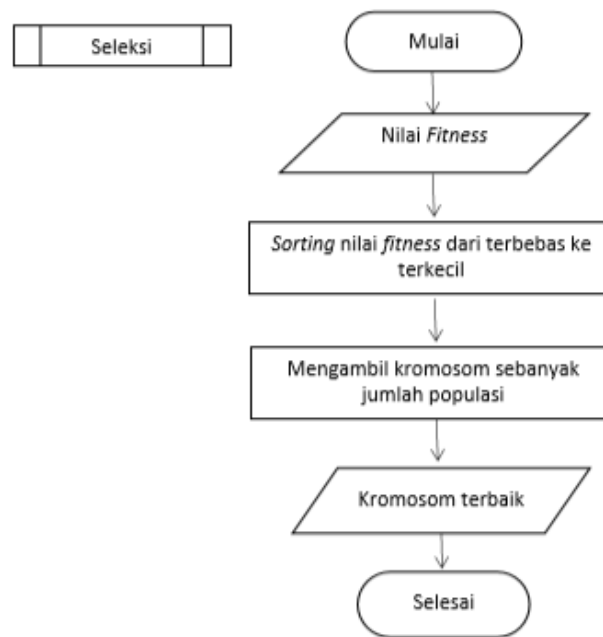
Penjelasan pada Tabel 4.8 yaitu p₁ – p₁₀ adalah urutan posisi barang yang akan masuk pertama sampai barang habis. Misal pada posisi yang pertama masuk yaitu barang dengan kode nomer 3, barang yang masuk kedua yaitu barang dengan kode 5 dan seterusnya.

Setelah diketahui urutan masuk barang, maka langkah selanjutnya yaitu mencari *fitness* sesuai Persamaan 2.7. Permisalan untuk mengetahui *fitness* dari masing-masing solusi dapat dilihat pada tabel 4.10.

Tabel 4.10 Perhitungan *fitness*

Kandidat Solusi	Perhitungan <i>fitness</i>	<i>fitness</i>
P1	$\left[\frac{(40 \cdot 45 \cdot 195 \cdot 1) + (60 \cdot 45 \cdot 215 \cdot 0) + (97 \cdot 50 \cdot 180 \cdot 1) + (75 \cdot 35 \cdot 175 \cdot 0) + (40 \cdot 60 \cdot 80 \cdot 1) + (80 \cdot 80 \cdot 170 \cdot 1) + (15 \cdot 8 \cdot 23 \cdot 1) + (40 \cdot 50 \cdot 40 \cdot 1) + (80 \cdot 60 \cdot 30 \cdot 0) + (48 \cdot 52 \cdot 105 \cdot 1)}{16.386.300} \right] \cdot 100\%$	0.174
P2	$\left[\frac{(60 \cdot 45 \cdot 215 \cdot 0) + (75 \cdot 35 \cdot 175 \cdot 1) + (80 \cdot 80 \cdot 170 \cdot 0) + (40 \cdot 45 \cdot 195 \cdot 1) + (15 \cdot 8 \cdot 23 \cdot 1) + (48 \cdot 52 \cdot 105 \cdot 0) + (97 \cdot 50 \cdot 180 \cdot 1) + (80 \cdot 60 \cdot 30 \cdot 0) + (40 \cdot 60 \cdot 80 \cdot 1) + (40 \cdot 50 \cdot 40 \cdot 0)}{16.386.300} \right] \cdot 100\%$	0.115
P3	$\left[\frac{(75 \cdot 35 \cdot 175 \cdot 0) + (15 \cdot 8 \cdot 23 \cdot 0) + (48 \cdot 52 \cdot 105 \cdot 1) + (40 \cdot 50 \cdot 40 \cdot 1) + (60 \cdot 45 \cdot 215 \cdot 0) + (80 \cdot 60 \cdot 30 \cdot 1) + (80 \cdot 80 \cdot 170 \cdot 1) + (40 \cdot 60 \cdot 80 \cdot 1) + (40 \cdot 45 \cdot 195 \cdot 1) + (97 \cdot 50 \cdot 180 \cdot 0)}{16.386.300} \right] \cdot 100\%$	0.129
C1	$\left[\frac{(75 \cdot 35 \cdot 175 \cdot 0) + (60 \cdot 45 \cdot 215 \cdot 1) + (80 \cdot 80 \cdot 170 \cdot 0) + (40 \cdot 45 \cdot 195 \cdot 1) + (60 \cdot 45 \cdot 215 \cdot 1) + (80 \cdot 60 \cdot 30 \cdot 1) + (80 \cdot 80 \cdot 170 \cdot 1) + (40 \cdot 50 \cdot 40 \cdot 0) + (80 \cdot 60 \cdot 30 \cdot 1) + (40 \cdot 60 \cdot 80 \cdot 1)}{16.386.300} \right] \cdot 100\%$	0.188
C2	$\left[\frac{(60 \cdot 45 \cdot 215 \cdot 1) + (40 \cdot 45 \cdot 195 \cdot 1) + (97 \cdot 50 \cdot 180 \cdot 1) + (75 \cdot 35 \cdot 175 \cdot 0) + (40 \cdot 60 \cdot 80 \cdot 1) + (80 \cdot 80 \cdot 170 \cdot 1) + (15 \cdot 8 \cdot 23 \cdot 1) + (80 \cdot 60 \cdot 30 \cdot 0) + (48 \cdot 52 \cdot 105 \cdot 1) + (40 \cdot 50 \cdot 40 \cdot 1)}{16.386.300} \right] \cdot 100\%$	0.209
C3	$\left[\frac{(40 \cdot 60 \cdot 80 \cdot 1) + (60 \cdot 45 \cdot 215 \cdot 0) + (97 \cdot 50 \cdot 180 \cdot 1) + (75 \cdot 35 \cdot 175 \cdot 0) + (40 \cdot 45 \cdot 195 \cdot 1) + (80 \cdot 80 \cdot 170 \cdot 1) + (15 \cdot 8 \cdot 23 \cdot 1) + (40 \cdot 50 \cdot 40 \cdot 1) + (80 \cdot 60 \cdot 30 \cdot 0) + (48 \cdot 52 \cdot 105 \cdot 1)}{16.386.300} \right] \cdot 100\%$	0.173
C4	$\left[\frac{(75 \cdot 35 \cdot 175 \cdot 1) + (40 \cdot 60 \cdot 80 \cdot 1) + (48 \cdot 52 \cdot 105 \cdot 0) + (40 \cdot 50 \cdot 40 \cdot 1) + (60 \cdot 45 \cdot 215 \cdot 0) + (80 \cdot 60 \cdot 30 \cdot 0) + (80 \cdot 80 \cdot 170 \cdot 1) + (15 \cdot 8 \cdot 23 \cdot 1) + (40 \cdot 45 \cdot 195 \cdot 0) + (97 \cdot 50 \cdot 180 \cdot 1)}{16.386.300} \right] \cdot 100\%$	0.164

Seleksi dilakukan dengan cara memilih individu dari populasi yang akan dipertahankan hidup pada generasi berikutnya. Semakin besar *fitness* kromosom, semakin besar peluang kromosom tersebut untuk terpilih. Metode seleksi yang digunakan yaitu *elitism*. Yaitu dengan cara melakukan pengurutan nilai *fitness* dari terbesar ke terkecil. Proses seleksi dalam dilihat dalam Gambar 4.7. Hasil seleksi elitism terdapat pada Tabel 4.11.



Gambar 4.6 Flowchart Proses Seleksi

Dalam diagram alir Gambar 4.6, setelah mendapatkan nilai *fitness* dari seluruh individu, kemudian dilakukan pengurutan nilai *fitness* dari yang terbesar ke terkecil. Sehingga didapatkan individu terbaik sebanyak jumlah populasi.

Tabel 4.11 Hasil seleksi elitism

Individu	<i>fitness</i>
C2	0.209
C1	0.188
P1	0.174
C3	0.173
C4	0.164
P3	0.129
P2	0.115

Setelah dilakukan proses seleksi, kemudian memilih individu terbaik sebanyak jumlah popsiZe yang dimasukkan di awal, yang akan dijadikan *parent* pada proses generasi berikutnya. Jika generasi telah selesai, maka akan terpilih hasil seleksi yang memiliki nilai *fitness* terbesar. Misalkan, hasil kromosom terbaik dapat dilihat pada Tabel 4.12.

Tabel 4.12 Pemilihan kromosom terbaik

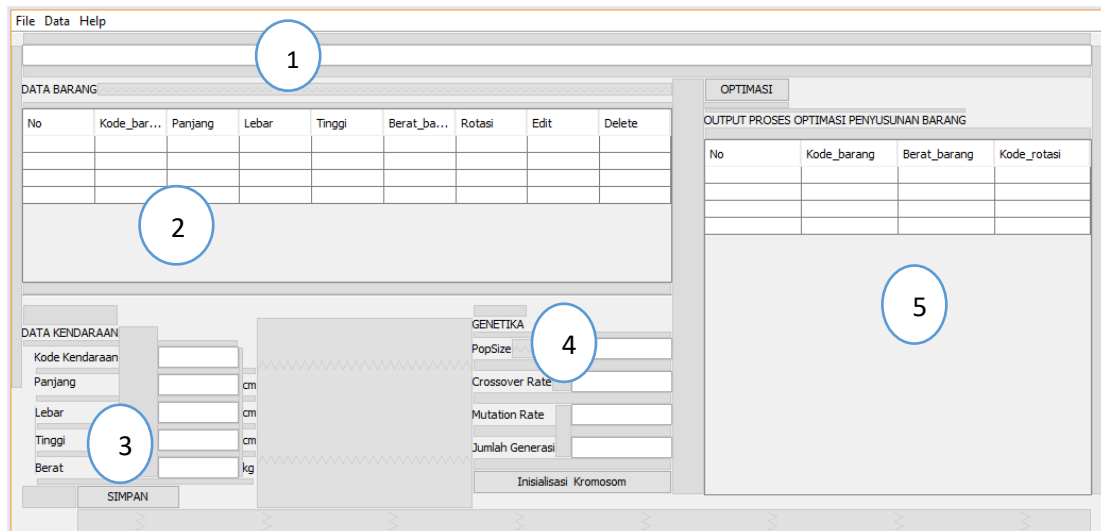
Parent	Kromosom										Fitness
	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	p ₉	p ₁₀	
C2	5	3	4	2	6	8	10	9	1	7	0.209

4.4 Perancangan User Interface

Pada perancangan *user interface* aplikasi optimasi penyusunan barang dalam mobil *box* dengan algoritme genetika terdiri dari dua halaman *interface* yaitu halaman input data barang dan mobil *box* serta halaman algoritme genetika.

4.4.1 Rancangan Tampilan Halaman Utama

Halaman utama terdiri dari import data barang dan input mobil *box*. Pada halaman ini *user* diminta memasukkan data barang berupa import excel dan menginputkan data mobil *box*. Selain itu, pada halaman ini *user* harus menginputkan jumlah populasi, jumlah generasi, nilai *cr* dan *mr*. Parameter tersebut nantinya akan digunakan dalam proses perhitungan algoritme genetika. Halaman utama dapat dilihat pada Gambar 4.7:



Gambar 4.7 Rancangan antarmuka halaman utama

Keterangan Gambar 4.7:

1. Hasil link untuk menginput data
2. Data barang hasil dari import excel
3. Data kendaraan yang diinput manual
4. Operasi algoritme genetika
5. Output dari hasil optimasi