

## BAB IV

### PERANCANGAN PERANGKAT LUNAK

#### 4.1 Metode Perancangan

Metode perancangan yang dilakukan pada tahap perancangan sistem optimasi tata letak barang digudang dengan Algoritma Genetika diimplementasikan menggunakan Diagram Alir (*Flow Chart*).

#### 4.2 Hasil Perancangan

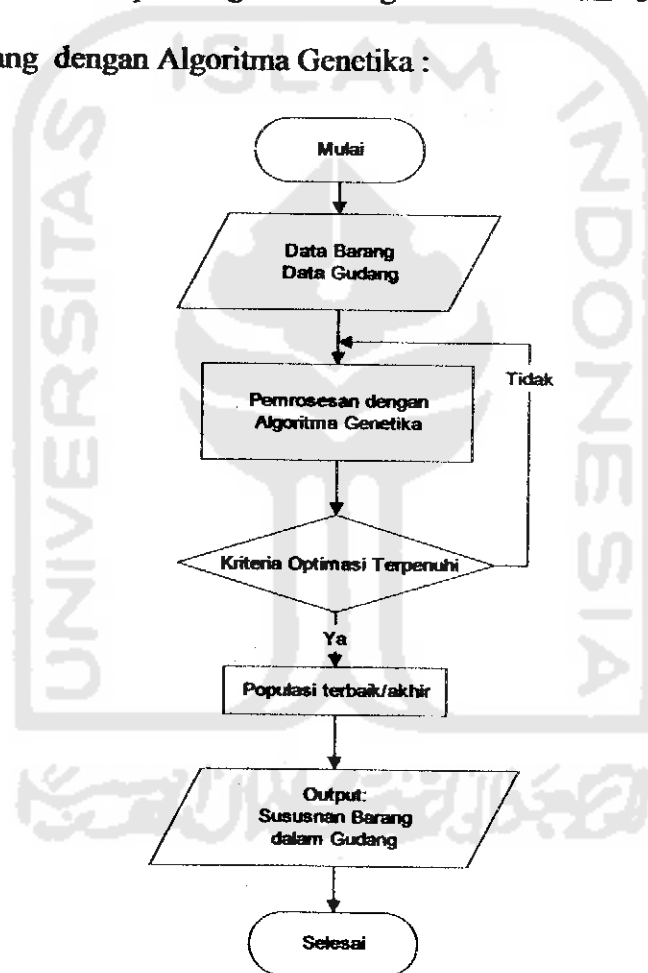
Hasil pada tahap perancangan berkaitan erat dengan hasil tahap analisis, karena pada tahap analisis telah ditentukan fungsi-fungsi dan metode-metode yang digunakan, sistem perangkat keras dan perangkat lunak yang dipakai, serta antarmuka yang diharapkan. Dari hasil tahap analisis tersebut didapat suatu gambaran tentang sistem Optimasi tata letak barang digudang dengan Algoritma Genetika. Perancangan sistem Optimasi tata letak barang digudang dengan algoritma genetika terdiri atas 2 bagian yaitu:

1. Basis data, bagian yang digunakan untuk menyimpan masukkan data dari sistem yang berupa dimensi gudang dan dimensi barang, sehingga dapat digunakan kembali untuk diproses dan sistem hasil proses.
2. Perangkat lunak Optimasi tata letak barang digudang yang memproses data untuk mencari tata letak yang optimum.

### 4.3 Perancangan Diagram Alir Sistem

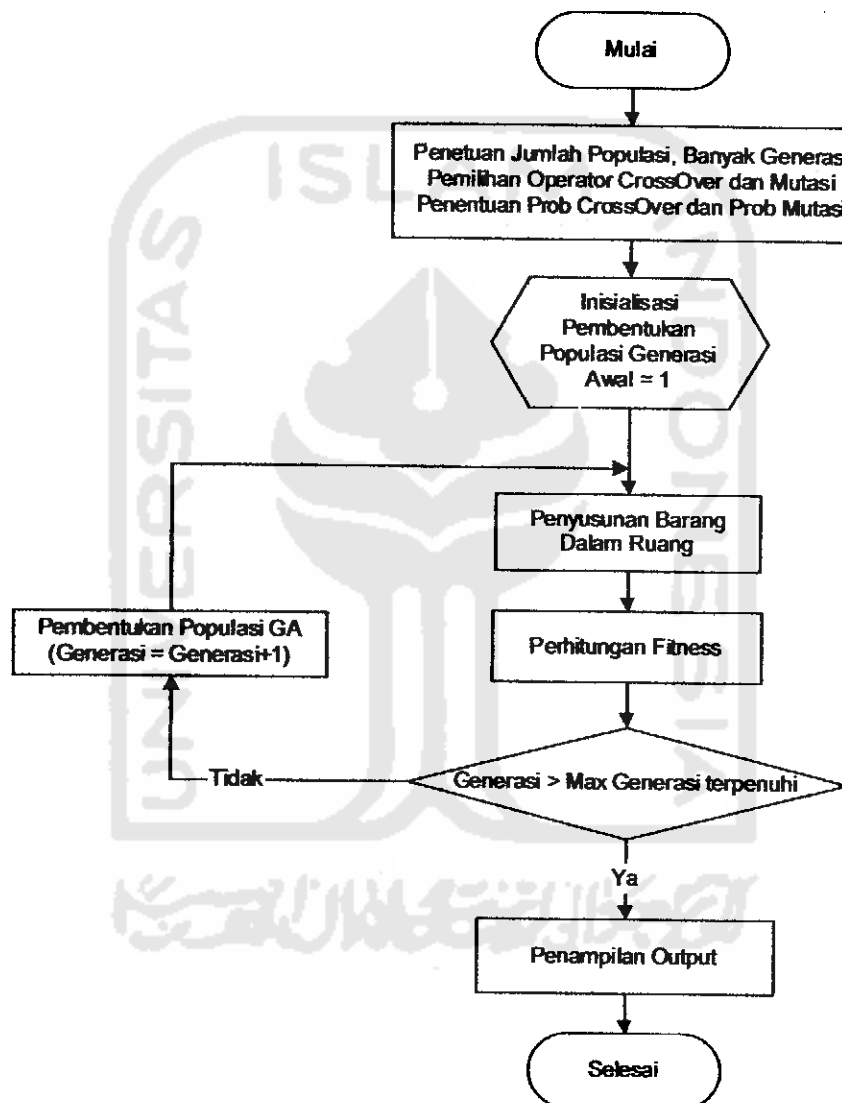
Pada sub bab ini, sistem akan digambarkan sebagai sebuah diagram alir (*Flow Chart*) secara keseluruhan. Dan kemudian akan dibuat diagram alir untuk tiap metode pelatihan yang terdapat didalamnya.

Berikut ini merupakan gambar diagram alir Sistem Optimasi tata letak barang digudang dengan Algoritma Genetika :



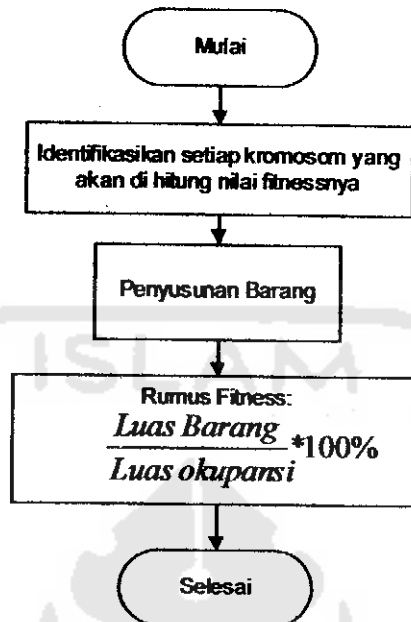
**Gambar 4.1** Diagram Alir Optimasi Tata Letak Barang di Gudang dengan Algoritma Genetika

Dari diagram alir Optimasi tata letak barang di gudang dengan Algoritma Genetika pada bagian pemrosesan Algoritma Genetika dapat dijabarkan ke dalam diagram alir berikut ini.



**Gambar 4.2** Diagram Alir Proses Algoritma Genetika

Dari Diagram alir proses Algoritma Genetika pada bagian proses operasi genetika, terdapat prosedur seleksi, *CrossOver* dan mutasi. Penjabaran dari prosedur-prosedur tersebut dapat dilihat dari diagram alir berikut :

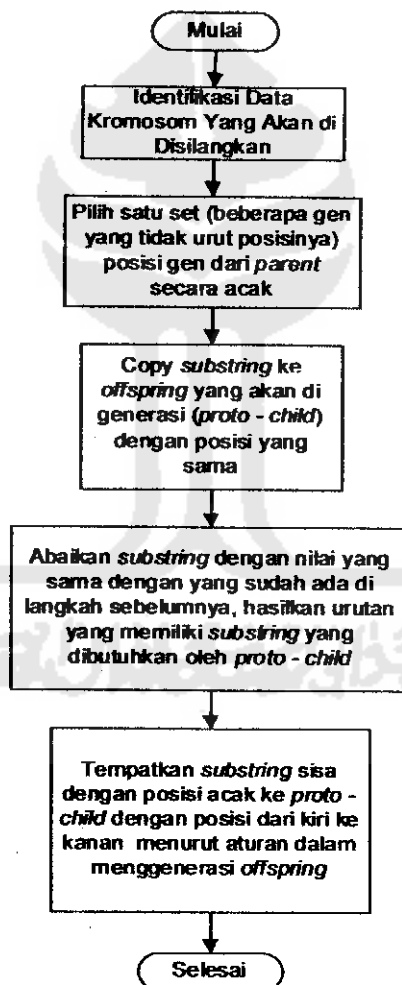


**Gambar 4.3** Diagram Alir Hitung *Fitness*

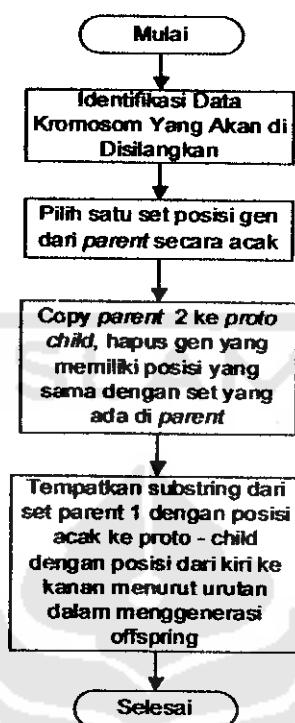
Pada diagram alir hitung *fitness* diatas proses yang pertama dilakukan mengidentifikasi setiap kromosom yang akan dihitung nilai *fitnessnya*, proses selanjutnya setiap kromosom dari gen-gennya yang merupakan urutan barang yang masuk maka dilakukan penyusunan barang, dari penyusunan setiap kromosom dihitung nilai *fitnessnya*. Nilai *fitnessnya* dihitung dari jumlah luas barang yang masuk dibagi dengan luas okupansi. Luas okupansi merupakan ruang gudang yang dianggap terpakai oleh barang.

Pada operasi *CrossOver* dibutuhkan dua induk untuk menghasilkan keturunan. Untuk itu setiap *CrossOver* diperlukan dua induk dari hasil seleksi, jika hasil seleksi merupakan bilangan ganjil maka dibuang satu kromosomnya sehingga menjadi bilangan genap. Berikut adalah proses dari masing-masing metode yang digunakan:

terpilih dan ditempatkan pada *offspring* pada posisi yang sama proses selanjutnya memilih substring pada parent kedua dengan mengabaikan pada langkah sebelumnya yaitu mengabaikan substring yang sama pada parent pertama maka. Setelah sisa substring pada parent kedua terpilih proses selanjutnya menempatkan substring sisa dengan posisi acak *proto-child* dengan posisi dari kiri kekanan menurut aturan dalam menggenerasi *offspring*.



Gambar 4.5 Diagram Alir *CrossOver* dengan *Position Based CrossOver*

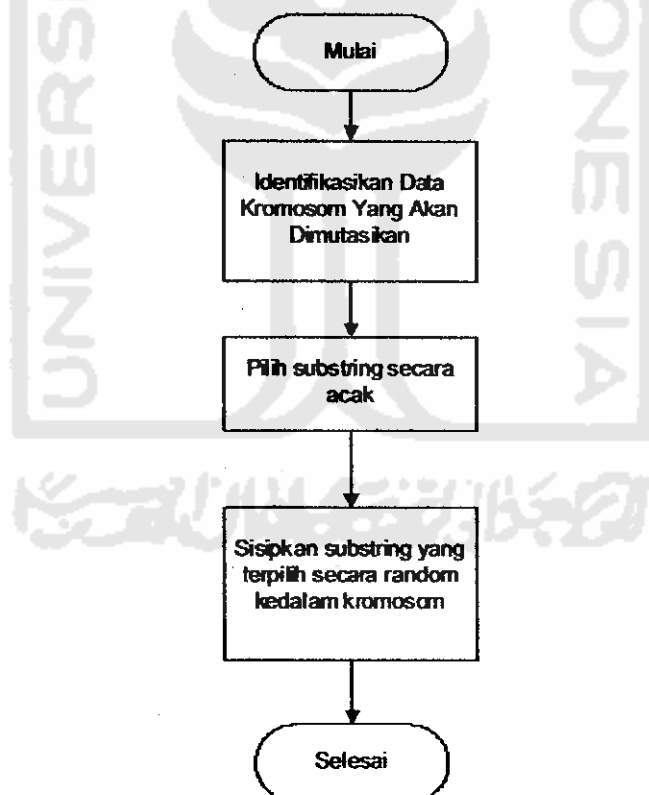


**Gambar 4.6** Diagram Alir *CrossOver* dengan *Order Based CrossOver*

Pada metode *Order Based CrossOver* dengan ketentuan kromosom yang disilangkan terdiri dari dua *parent* dan menghasilkan satu *offspring*. Pada gambar 4.6 merupakan gambar diagram alir proses demi proses pada *Position Based CrossOver*. Langkah pertama yang dilakukan adalah mengidentifikasi data kromosom yang akan disilangkan, data kromosom ini dihasilkan dari proses seleksi. Kemudian setelah langkah pertama telah ditentukan data kromosomnya memilih dua induk yang akan disilangkan sebanyak  $2 \times n$  kromosom, proses kedua pilih satu set posisi gen dari *parent* secara acak, proses selanjutnya Copy *parent* kedua ke *proto-child*, hapus gen yang memiliki posisi yang sama dengan set yang ada di *parent*, proses yang terakhir adalah tempatkan substring dari *parent* pertama dengan posisi acak ke *proto-child* dengan posisi dari kiri kekanan menurut urutan dalam menggenerasi *offspring*.

buat kedua *offspring* dari hasil penyilangan kedua *parent* menjadi legal yaitu mencari nilai gen-gen pada kromosom yang belum ada pada masing-masing *offspring*.

Operasi mutasi melakukan perubahan nilai gen – gen pada kromosom dengan cara menyisipkan kode informasi ke dalam suatu individu. Mutasi bertujuan untuk menjamin bahwa algoritma genetika melakukan eksplorasi pada ruang pencarian sehingga tidak terperangkap ke dalam suatu bagian tertentu. Berikut *flowchart* untuk metode mutasi yang digunakan:



**Gambar 4.8** Diagram Alir *Displacement Mutation*

Pada *Displacement Mutation*, Mutasi yang mungkin terjadi adalah perubahan urutan barang. Hal ini dilakukan secara acak, diambil 2 angka (nomor

rotasi posisi. Proses yang pertama dilakukan mengidentifikasi data kromosom yang terpilih dari hasil seleksi mutasi. Setelah data kromosom terpilih proses selanjutnya dari setiap kromosom yang terpilih melakukan proses pengacakan pada satu posisi penempatan, setelah proses pengacakan dilakukan proses yang terjadi adalah melakukan penggantian posisi penempatan yaitu dari Vertikal ke horizontal atau sebaliknya dan dimasukkan ke dalam kromosom.

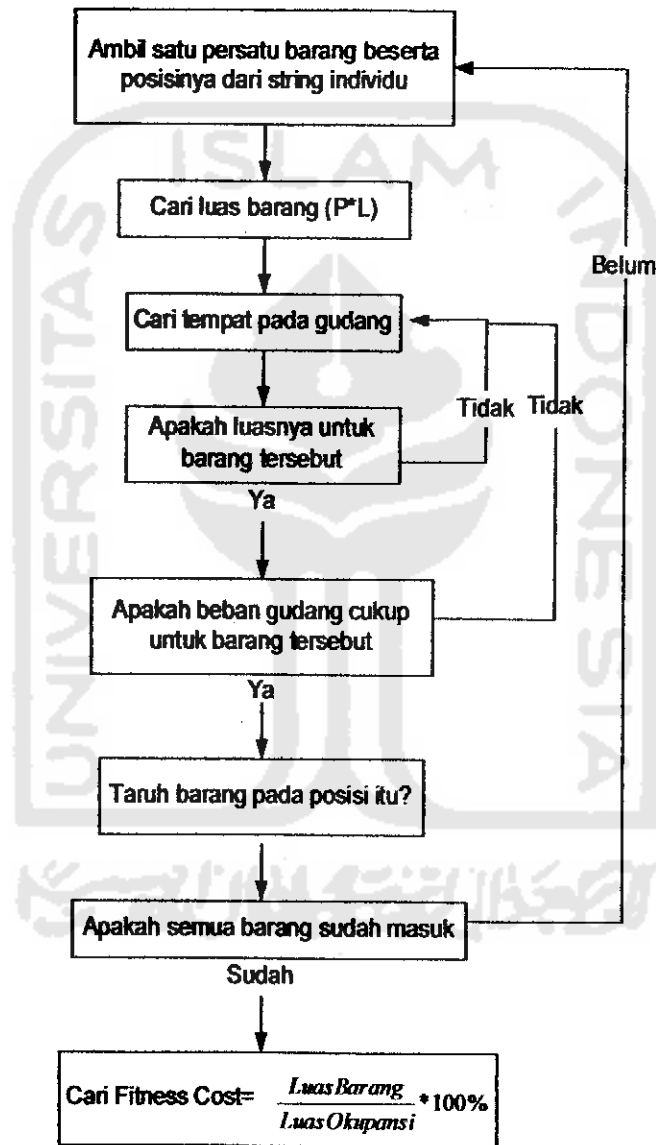
#### 4.4 Perancangan kromosom

Kromosom merupakan kumpulan dari gen-gen yang ada, yang berarti panjang kromosom akan sesuai dengan jumlah barang untuk suatu kategori barang. Pada kasus pengisian barang ke gudang yang menjadi individu adalah urutan barang yang akan dimasukkan ke dalam gudang. Selain urutan barang didalam setiap individu juga tersimpan informasi bagaimana posisi barang ketika masuk dalam gudang yaitu vertikal atau horizontal.

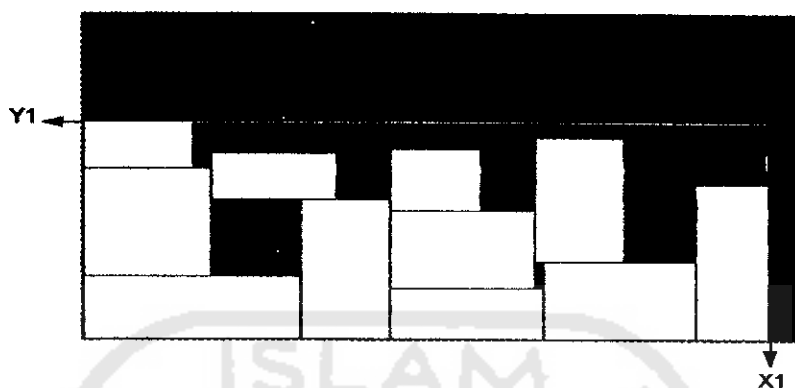
Dalam hal ini populasi adalah sebuah string yang berisi gen yang berjumlah sesuai dengan barang pada tiap kategori. Jumlah panjang gen 3 karakter, maka setiap kode barang dengan nomer barang kurang dari 10 akan ditambah dengan angka 0, jadi panjang string adalah  $3 \times n$ , dengan  $n$  merupakan jumlah barang yang akan dimasukkan ke dalam gudang. Sebagai contoh dapat dilihat Pada gambar 4.10 misalnya 04H, berarti urutan barang pertama yang dimasukkan di gudang dan posisi penempatannya horizontal. Nomer barang 04 merupakan barang yang nilainya kurang dari 10 maka akan ditambahkan angka 0 didepannya.



luas okupansi didapat dari  $X_1 \times Y_1$ .  $X_1$  adalah ordinat x dari posisi barang paling akhir pada ordinat x, sedangkan  $Y_1$  adalah ordinat y dari posisi barang paling akhir pada ordinat y.



**Gambar 4.11** Algoritma Pencarian Nilai *Fitness*



Gambar 4.12 Kondisi Gudang untuk menghitung nilai *fitness*

#### 4.6 Struktur Data

Struktur data yang digunakan di dalam perangkat lunak ini untuk pembangkitan generasi awal ialah *Array* berdimensi dua yang telah ditentukan indeksinya. Contohnya adalah sebagai berikut:

```
Individu:array [1..100] of array [1..100]of string;
```

Indeks pertama menunjukkan baris dan indeks kedua menunjukkan kolom. Baris dan kolom tersebut hanya dikenal dalam representasi logika, tidak dalam representasi fisik pada memori komputer.

Pada penggunaan metode *crossover* menggunakan Struktur data *Dynamic Arrays* (Array Dinamis), sehingga penggunaan memori dapat dialokasikan dengan tepat karena bergantung dari jumlah data masukan serta parameter genetika untuk pemrosesan solusi akhir. Hal ini dikarenakan *Dynamic arrays* tidak mempunyai ukuran atau panjang yang tetap. Tetapi besarnya memori yang digunakan untuk *Dynamic Arrays* dialokasikan begitu suatu nilai di *assign* dengan prosedur *SetLength*. Contohnya suatu kromosom didefinisikan sebagai array dinamis dengan berdimensi dua dengan perintah

kromosom : array of array of integer;

selanjutnya kromosom diinisialisasi ukuran memori yang digunakan dengan perintah

```
setlength(kromosom, 10, 20);
```

sehingga sekarang kromosom mempunyai 10 baris yang berisi elemen-elemen dimana tiap baris mempunyai 20 elemen yang bertipe *integer*.

#### 4.7 Basis Data

Basis data merupakan storage yang akan menyimpan data secara permanen. Walaupun aplikasi optimasi tata letak barang dimatikan. Sehingga data proses kerja yang telah dibuat akan dapat digunakan kembali tanpa melakukan proses penginputan kembali.

Pada sistem ini digunakan basis data sejumlah 2 tabel yaitu tabel barang dan tabel gudang. Berikut ini adalah struktur tabel yang digunakan pada aplikasi optimasi tata letak barang digudang dengan algoritma genetika:

**Tabel 4.2 Tabel Gudang**

No	Nama Field	Type	Ukuran	Keterangan
1	Kode_Gudang	Alpha	10	Nama gudang dengan numerik
2	Panjang	Number		Untuk menentukan panjang gudang
3	Lebar	Number		Untuk menentukan lebar gudang.

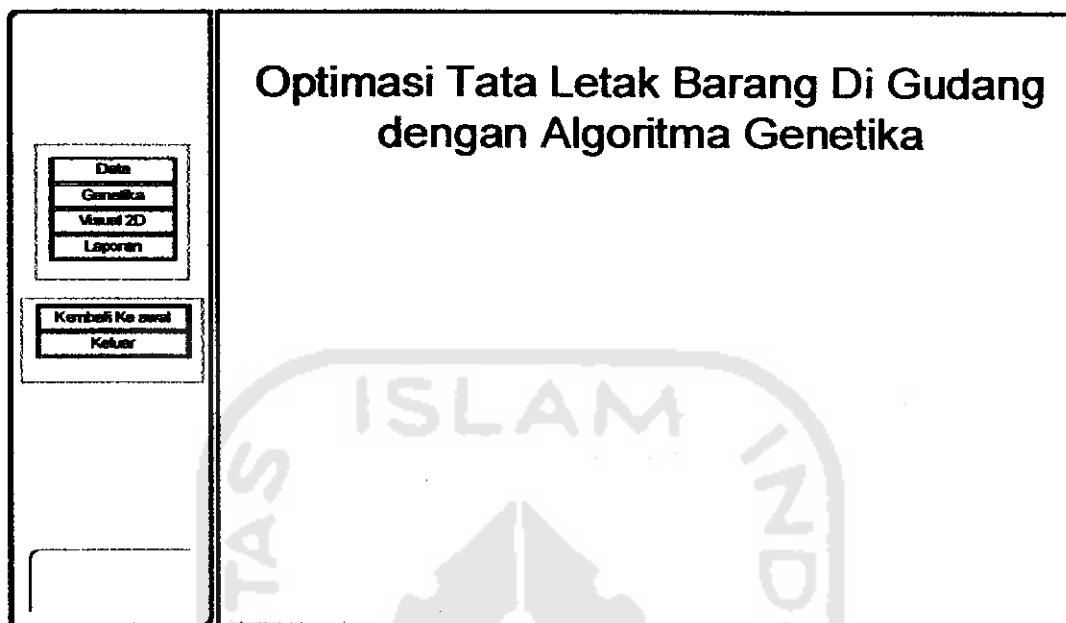
Tabel gudang berfungsi untuk menyimpan dimensi gudang yang akan diproses untuk penyimpanan barang ke dalam gudang setelah dilakukan proses algoritma genetika.

#### **4.8 Perancangan Antarmuka**

Rancangan antarmuka dari sistem Optimasi Tata Letak Barang di Gudang dengan Algoritma Genetika ini menggunakan perancangan model grafis. Terdapat 5 bagian yaitu bagian antarmuka Menu, bagian antarmuka Data, bagian antarmuka Proses Algoritma Genetika, bagian antarmuka Visual, bagian antarmuka Laporan.

##### **4.8.1 Rancangan Antarmuka Menu**

Antarmuka ini digunakan untuk memilih menu-menu yang terdapat di dalam Sistem Optimasi Tata Letak Barang di Gudang dengan Algoritma Genetika.



Gambar 4.13 Rancangan antarmuka menu

#### 4.8.2 Rancangan Antarmuka Data

Antarmuka ini untuk mendefinisikan data masukan untuk proses sistem. Pada bagian ini *input* yang dimasukkan adalah Dimensi Gudang dan Dimensi barang.

The image shows a software interface with the following components:

- A text input field labeled "Nama File" at the top left.
- Three buttons labeled "Simpan", "Baru", and "Buka" located below the "Nama File" field.
- A table with 3 columns and 3 rows, located below the buttons.
- A text input field labeled "Jumlah Barang" with an "OK" button to its right, located to the right of the first table.
- A second table with 3 columns and 6 rows, located below the first table.

**Gambar 4.14** Rancangan antarmuka masukan

#### 4.8.3 Rancangan Antarmuka Proses

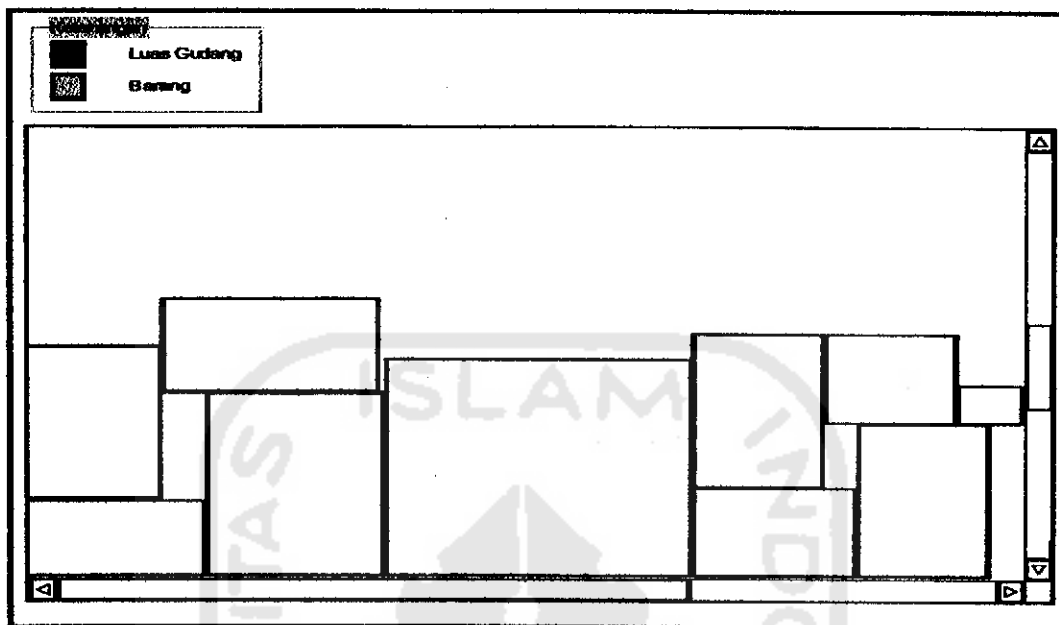
Antarmuka ini digunakan untuk melakukan proses pencarian dengan algoritma genetika, dimana terdapat beberapa parameter yang harus dimasukkan atau dipilih. Diantaranya *Ukuran Populasi*, *Jumlah Generasi*, *Probabilitas CrossOver*, *Probabilitas Mutasi*, dan pemilihan metode *CrossOver* dan *Mutasi*.

The image shows a software interface for a genetic algorithm process. On the left side, there is a control panel with several input fields and buttons. The fields include 'Ukuran Populasi' and 'Jumlah Generasi' with spinners, 'Metode Perilangan' and 'Metode Mutasi' with dropdown menus, and 'Probabilitas (%)' with a slider. Below these are 'Fitness Terbaik' and 'Generasi Ke' with text boxes, and a 'Proses' button. The right side of the interface is a large empty area labeled 'Grafik', intended for displaying the results of the algorithm.

**Gambar 4.15** Rancangan antarmuka proses

#### 4.8.4 Rancangan Antarmuka Visual

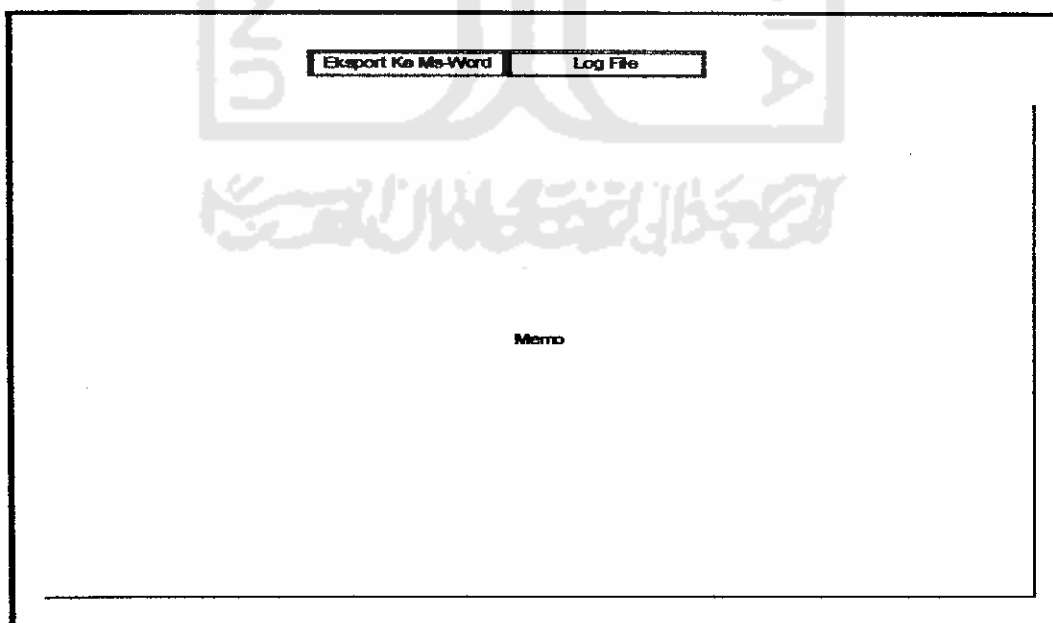
Antarmuka ini digunakan untuk melihat pola susunan barang digudang yang terbaik dengan dua dimensi setelah proses akhir dari algoritma genetika terpenuhi.



**Gambar 4.16** Rancangan Antarmuka Visualisasi Barang

#### 4.8.5 Rancangan Antarmuka Laporan

Antarmuka ini digunakan untuk mencetak laporan solusi yang dihasilkan.



**Gambar 4.17** Rancangan antarmuka laporan



## 4.9 Perancangan File

### 4.9.1 Rancangan File untuk Input Data

Pada bagian input data, file yang digunakan sebagai masukan adalah file yang berekstensi \*.Brg. Baik file yang dibuat oleh perangkat lunak, maupun yang diambil dari file yang tersimpan. Rancangan file dapat dilihat pada gambar 4.15 sebagai berikut:

X	X	X							Kode Gudang, Panjang,Lebar
X									Jumlah Barang
XXXXXXXXXX									Keterangan "Terproses" atau "Belum Terproses"
X	X	X	X	X	X	X	X	}	Dimensi Masing-masing barang yang akan diproses sesuai dengan jumlah barangnya. Urutannya adalah sebagai berikut: Kode Barang, Panjang, Lebar, OrdinatX, OrdinatY, Posisi Penempatan Barang V/H (Vertikal atau Horizontal), Keterangan
X	X	X	X	X	X	X	X		
X	X	X	X	X	X	X	X		
X	X	X	X	X	X	X	X		
X	X	X	X	X	X	X	X		
X	X	X	X	X	X	X	X		
X	X	X	X	X	X	X	X		
X	X	X	X	X	X	X	X		
X	X	X	X	X	X	X	X		
X	X	X	X	X	X	X	X		

**Gambar 4.18** Ilustrasi File untuk input Data

Rancangan file berbentuk matrik dengan baris pertama berisi kode gudang, panjang gudang dan lebar gudang, baris kedua berisi jumlah barang yang akan diproses, kemudian pada baris ketiga keterangan proses atau belum terproses, keterangan proses apabila data pernah diproses menggunakan algoritma genetika sedangkan keterangan belum terproses apabila data belum pernah diproses dengan algoritma genetika dan baris selanjutnya dimensi masing masing barang yang akan diproses beserta ordinatx dan ordinaty beserta posisi barang yang akan diletakan. Pada ordinatx dan ordinaty beserta posisi yang telah ada berarti data telah diproses dengan algoritma genetika yang kemudian dapat diperoleh hasilnya dengan melihat visual tata letak barang.

#### 4.9.2 Rancangan File untuk Proses Data

Pada bagian proses data, file yang digunakan untuk menyimpan proses-proses genetika disimpan dengan nama file '\*.log'. Untuk mengisikan hasil dari proses-proses genetika digunakan suatu algoritma penulisan file teks. Rancangan dari hasil-hasil proses tersebut berisi data kromosom untuk setiap generasi, hasil perhitungan nilai *fitness* untuk setiap generasi, proses seleksi roda rolet (*Roulette Whells*) dengan probabilitas dari masing-masing kromosom untuk setiap generasi, penentuan kromosom-kromosom yang akan disilangkan dan dimutasikan untuk setiap generasi, serta hasil akhir nilai *fitness* terbaik dari masing-masing generasi.

#### 4.9.3 Rancangan File untuk Output Sistem

Pada bagian output sistem, file yang digunakan untuk menyimpan laporan tata letak barang berekstensi \*.doc dan untuk menyimpan urutan proses algoritma genetika digunakan file dengan ekstensi \*.log.