BAB III

METODE PENELITIAN

Algortima pencarian dengan menggunakan model reproduksi sexual dinamakan *Genetic Algorithm (GA)*. Proses reproduksi dilakukan dengan jalan merekombinasi 2 (dua) individu menjadi sebuah individu lain atau *offspring*. Proses GA dalam mencari suatu nilai optimum adalah sebagai berikut:

1. Penentuan model representasi genetika dari problem yang dihadapi yang berupa model dari sistem buatan, representasi individu, pembatas, fungsi evaluasi.
2. Pembangkitan generasi awal dengan memproses database.
3. Penentuan nilai fitness (nilai ketepatan) dari setiap individu berdasarkan struktur gennya. Nilai fitness ini dijadikan sebagai ukuran apakah individu tersebut sudah optimal atau belum.
4. Pemilihan individu dengan nilai fitness terbaik untuk dijadikan induk dalam menghasilkan individu-individu baru.
5. Proses reproduksi yang terdiri dari *Crossover*, *Mutation,* dan *Replacement.* Dari proses reproduksi ini dihasilkan individu baru.

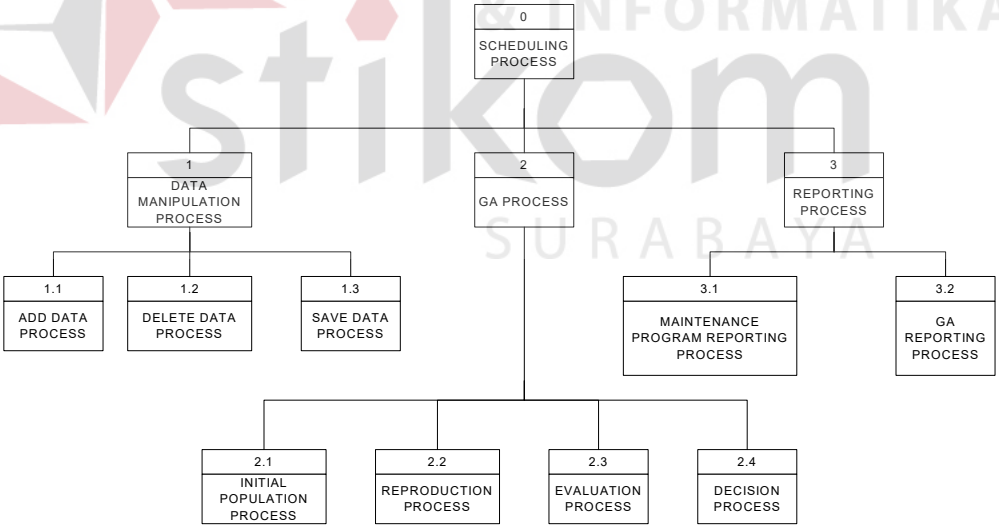
Dengan melakukan proses di atas secara berulang-berulang, diharapkan induk yang baik akan menghasilkan suatu generasi dengan individu-individu yang lebih baik. Untuk menerapkan hal-hal di atas maka diperlukan suatu *Data Flow Diagram (DFD)* dan *Entity Relational Diagram (E–R Diagram)*, sebagai pendekatan terhadap pokok permasalahan.

3.1 Model Penelitian

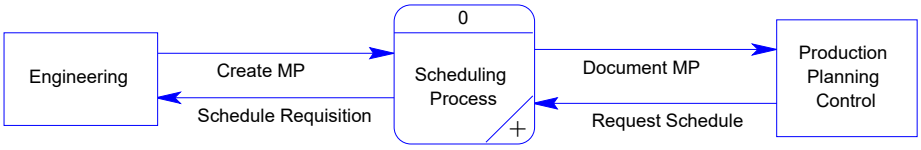
Dalam melakukan analisis terhadap permasalahan, dibuat beberapa tahapan model penelitian. Tahapan tersebut adalah: (1) membuat *data flow diagram* sistem perawatan dan membuat struktur database-nya, (2) menetapkan representasi individu dan fungsi evaluasi (3) menetapkan kesimpulan.

3.1.1 Data Flow Diagram (DFD)

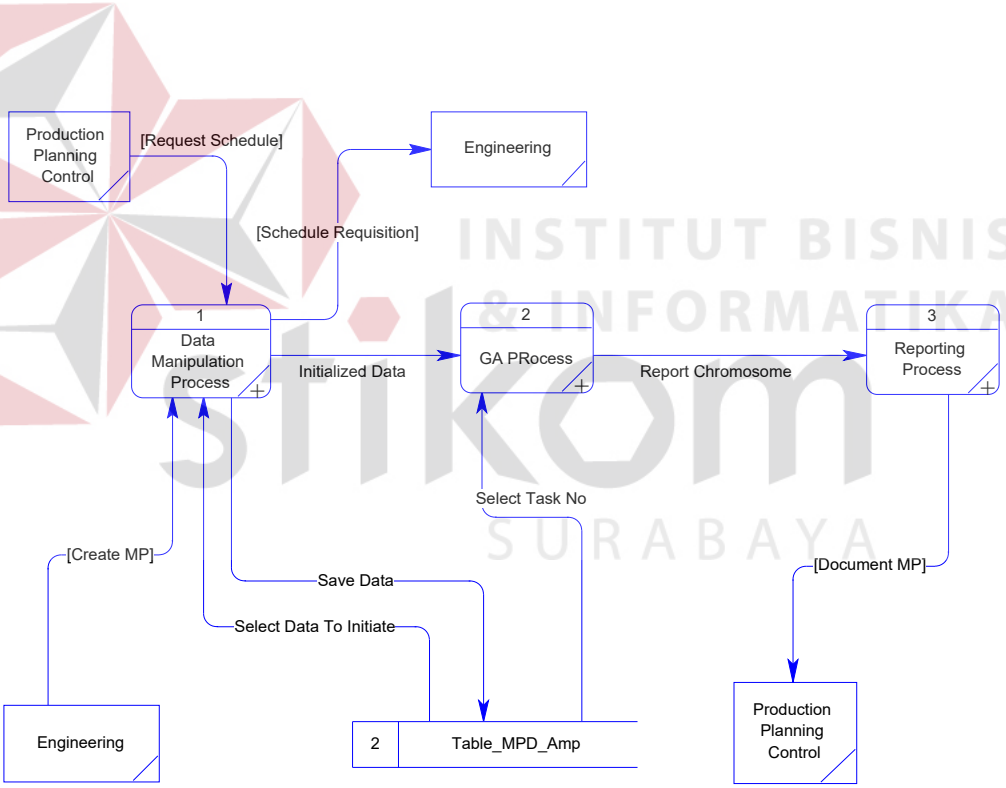
DFD di bawah ini adalah merupakan DFD dari sistem perawatan dimana dalam salah satu proses penetapan master scheduling, fungsi GA ini dimanfaatkan untuk mencari penjadwalan yang optimum dari seluruh pekerjaan dalam sistem perawatan. Context Diagram dan DFD dari sistem perawatan tersebut adalah sebagai berikut:



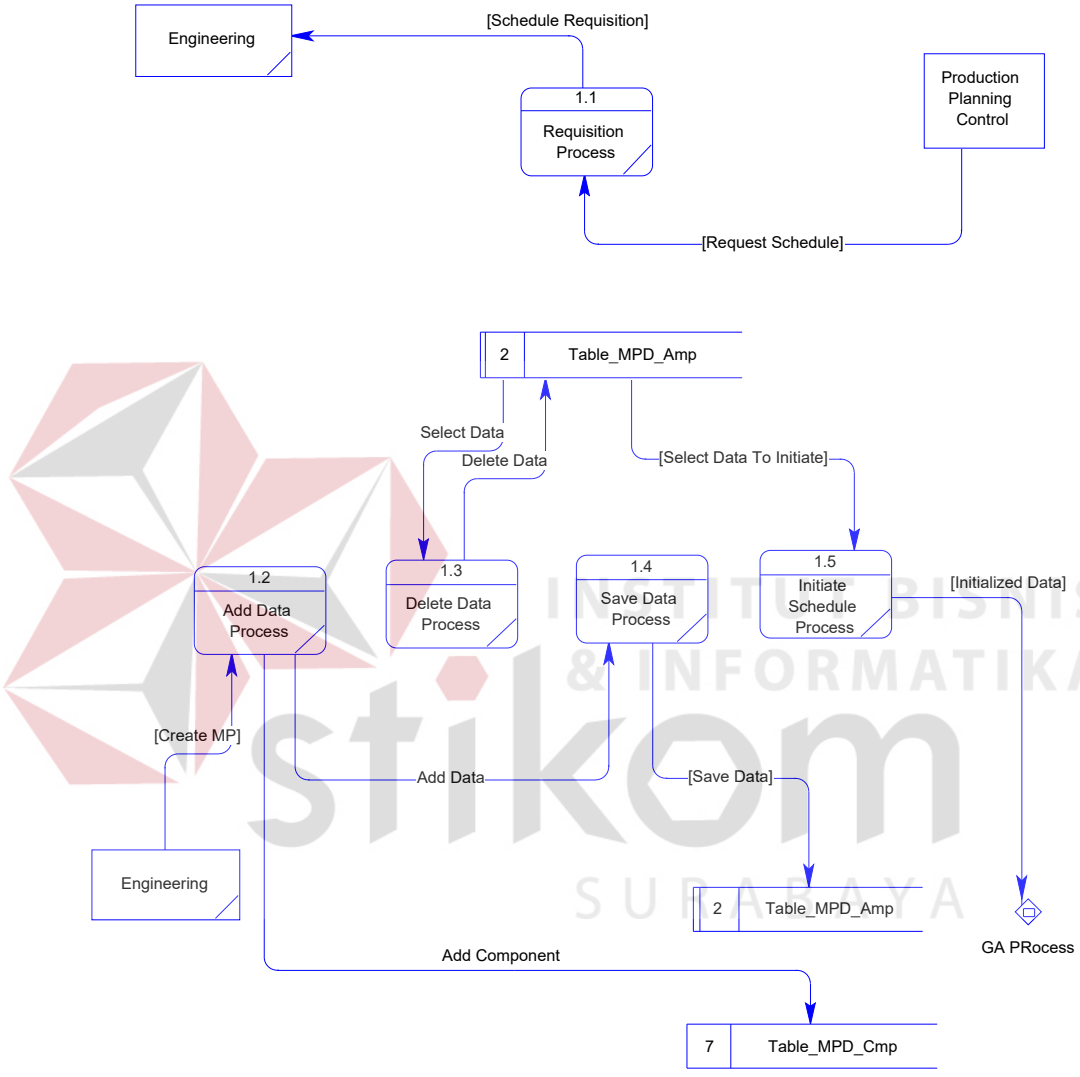
Gambar 3.1 Scheduling Context Diagram



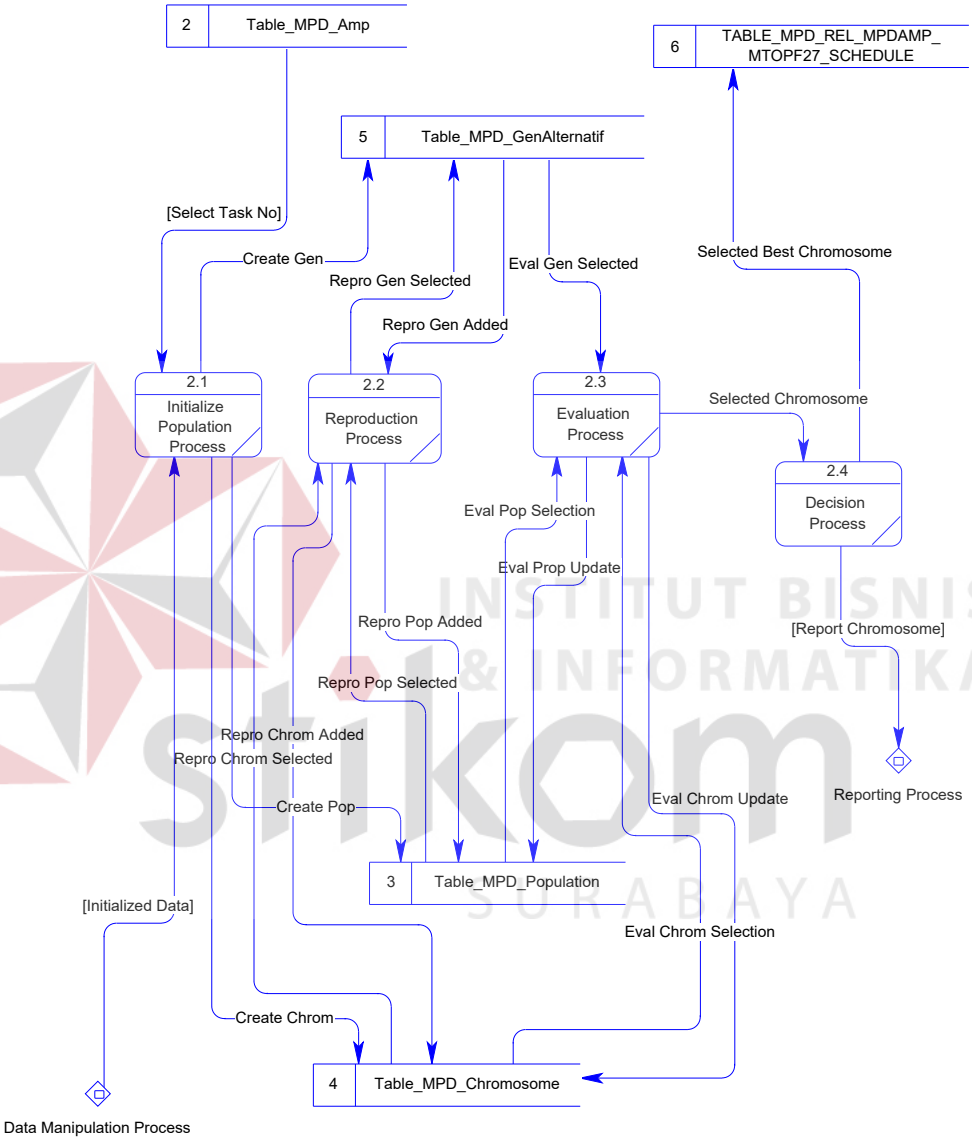
Gambar 3.2 DFD Level 0, Scheduling Management



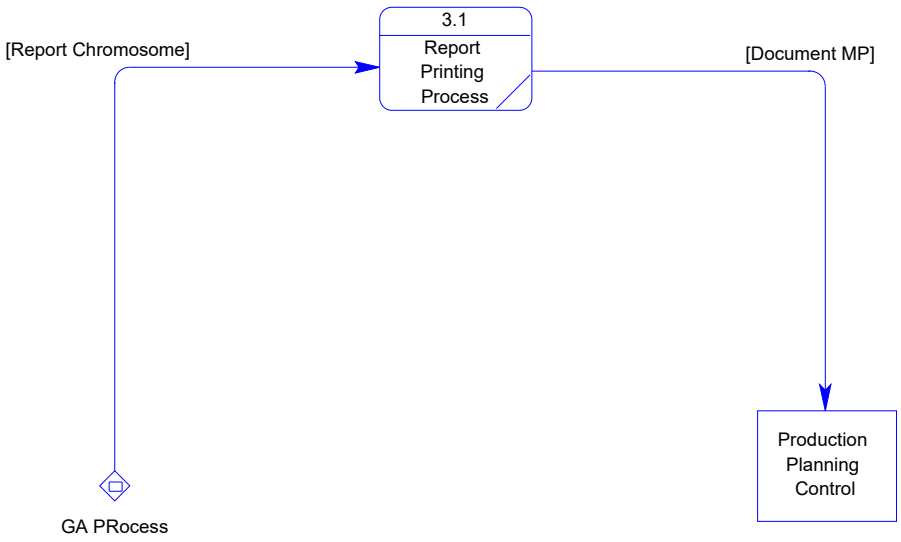
Gambar 3.3 DFD Level 1, Scheduling Management



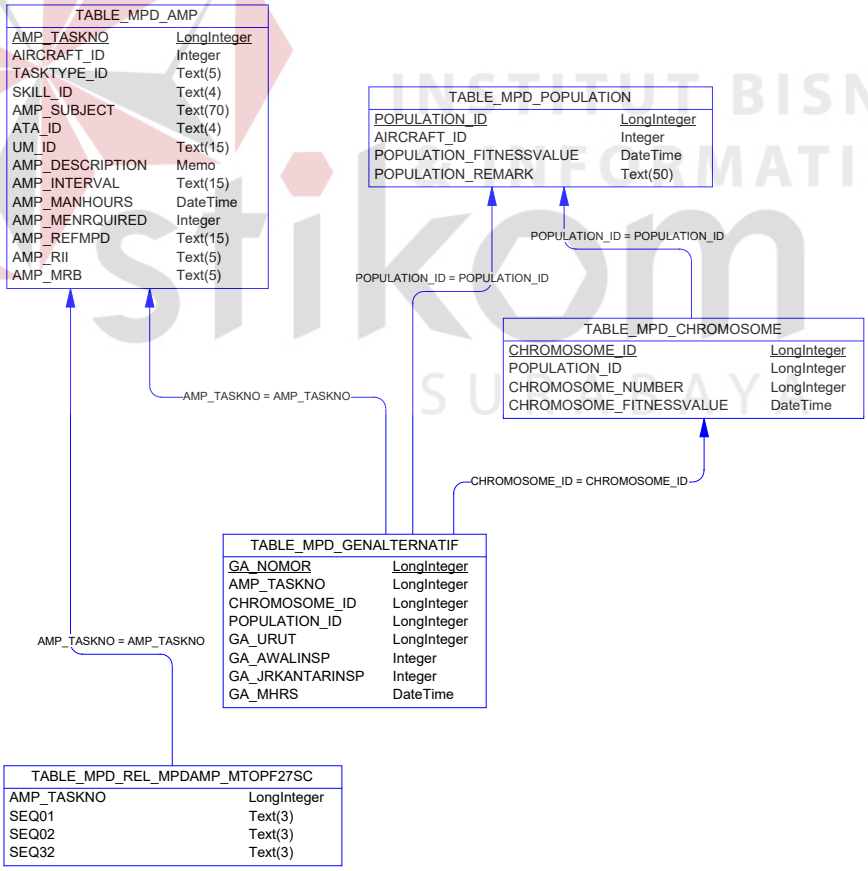
Gambar 3.4 DFD Level 2, Proses Manipulasi Data



Gambar 3.5 DFD Level 2, Proses GA



Gambar 3.6 DFD Level 2, Proses Pelaporan



Gambar 3.7 E-R Diagram Scheduling Management

Berdasarkan E-R Diagram, kemudian dibuat struktur file database dengan perincian sebagai berikut:

a. Nama File : Table_MPD_Amp
Primary Key : AMP_TASKNO
Foreign Key : -
File Relasi : TABLE_MPD_REL_MPDAMP_MTOPF27SCHED, Table_MPD_GenAlternatif
Keterangan : Tabel Utama untuk menyimpan data penjadwalan perawatan

No.	Nama Field	Tipe Data	Panjang	Keterangan
1.	Amp_Taskno	Long Integer		Kode Nomor Pekerjaan
2.	Aircraft_ID	Integer		Kode Jenis Pesawat
3.	TaskType_ID	Text	5	Kode Jenis Pekerjaan
4.	Skill_ID	Text	4	Kode Jenis Keahlian
5.	Amp_Subject	Text	70	Judul Pekerjaan
6.	ATA_ID	Text	4	Kode Kategori Sistem Pada Pesawat
7.	UM_ID	Text	15	Kode Unit Pengukuran
8.	Amp_Description	Memo		Penjelasan Pekerjaan
9.	Amp_Interval	Text	15	Interval Pelaksanaan Pekerjaan
10.	Amp_Manhours	Double		Kebutuhan JamOrang untuk pelaksanaan pekerjaan
11.	Amp_Menrquired	Integer		Jumlah kebutuhan Orang untuk melaksanakan pekerjaan
12.	Amp_RefMPD	Text	15	Nomor Referensi Pekerjaan dari pabrik pesawat
13.	Amp_RII	Text	5	Pilihan untuk pemeriksaan ganda
14.	Amp_MRB	Text	5	Pilihan untuk kode pemeriksaan spesifik dari pabrik pesawat

b. Nama File : Table_MPD_Population
Primary Key : Population_ID
Foreign Key : -
File Relasi : Table_MPD_Chromosome, Table_MPD_GenAlternatif
Keterangan : Tabel untuk menyimpan data populasi

No.	Nama Field	Tipe Data	Panjang	Keterangan
1.	Population_ID	Long Integer		Kode Populasi
2.	Aircraft_ID	Integer		Kode Jenis Pesawat
3.	Population_Fitness_Value	Integer		Nilai fitness populasi dari hasil evaluasi
4.	Population_Remark	Varchar2	50	Keterangan Populasi

c. Nama File : Table_MPD_Chromosome
Primary Key : Chromosome_ID
Foreign Key : Population_ID
File Relasi : Table_MPD_Population, Table_MPD_GenAlternatif
Keterangan : Tabel untuk menyimpan data kromosom

No.	Nama Field	Tipe Data	Panjang	Keterangan
1.	Chromosome_ID	Long Integer		Kode Kromosom
2.	Population_ID	Long Integer		Kode Populasi
3.	Chromosome_Number	Integer		Jumlah Kromosom
4.	Chromosome_Fitness_Value	Integer		Nilai Fitness Kromosom dari hasil evaluasi

d. Nama File : Table_MPD_GenAlternatif
Primary Key : GA_Nomor
Foreign Key : Population_ID, Chromosome_ID
File Relasi : Table_MPD_Population, Table_MPD_Chromosome
Keterangan : Tabel untuk menyimpan data gen

No.	Nama Field	Tipe Data	Panjang	Keterangan
1.	Population_ID	Long Integer		Kode Populasi
2.	Chromosome_ID	Long Integer		Kode Kromosom
3.	GA_Nomor	Long Integer		Kode Gen
4.	GA_Urut	Long Integer		Nomor Urut Sesuai Nomor Pekerjaan
4.	AMP_Taskno	Long Integer		Kode Nomor

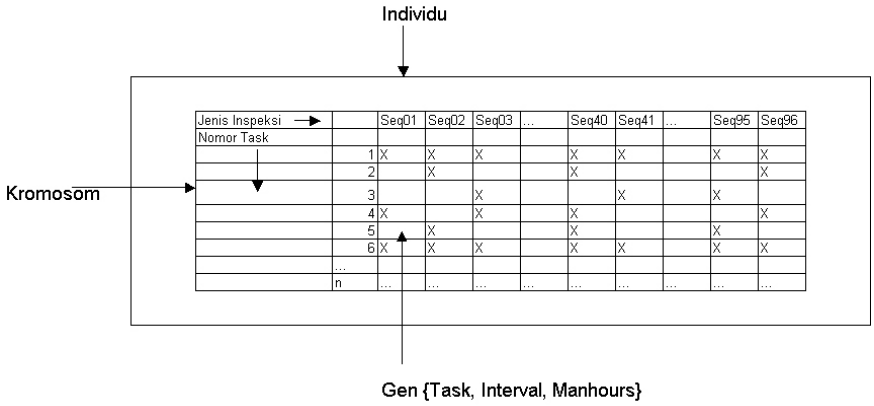
No.	Nama Field	Tipe Data	Panjang	Keterangan
				Pekerjaan
5.	GA_AwalInsp	Integer		Posisi Awal Pekerjaan Untuk Ditempatkan Dalam Jadwal
6	GA_JrkAntarInsp	Integer		Besarnya Interval Pekerjaan
7.	AMP_Manhours	Double		Kebutuhan JamOrang untuk pelaksanaan pekerjaan

e. Nama File : Table_MPD_Rel_MPDAMP_MtopF27Sched
Primary Key : No
Foreign Key : AMP_Taskno
File Relasi : Table_MPD_AMP
Keterangan : Tabel untuk menyimpan data Jadwal terbaik

No.	Nama Field	Tipe Data	Panjang	Keterangan
1.	No	Long Integer		Nomor Urut
2.	AMP_Taskno	Long Integer		Kode Nomor Pekerjaan
3.	Seq_01	Text		Sekuen 01
4.	Seq_02	Text		Sekuen 02
5.	…			
6.	Seq_32	Text		Sekuen 32

3.1.2 Desain Representasi Individu

Model yang digunakan dalam pembuatan jadwal adalah sebuah Maintenance Program, dimana dinyatakan bahwa satu jadwal adalah satu Individu. Kromosom dari individu adalah susunan jadwal berdasarkan waktu pelaksanaan perawatan. Gen adalah satu pekerjaan (*task card*). Lebih jelas dapat dilihat pada Gambar 3.8.



Gambar 3.8 Individu, Kromosom dan Gen dari Sistem Perawatan

Maintenance Program adalah merupakan daftar pekerjaan yang harus dilaksanakan (*task card*) pada interval waktu tertentu dan membutuhkan sumber daya manusia tertentu. Daftar tersebut berisi tentang pekerjaan yang harus dikerjakan (*task*) *v* atau $V=\{v_1,v_2,v_3,...,v_n\}$ yang dilaksanakan pada interval waktu *i* atau $I=\{i_1,i_2,i_3,...,i_n\}$ dan membutuhkan sumber daya manusia (*manhours*) *m* atau $M=\{m_1,m_2,m_3,...,m_n\}$.

Jika terdapat sebuah pekerjaan a pada interval waktu b dan membutuhkan sumber daya manusia c, maka dapat digambarkan J = { (a,b,c) | a ∈ *V*, b ∈ *I*, c ∈ *M*} atau terdapat jadwal (a,b,c) sedemikian rupa sehingga a ∈ *V* , b ∈ *I* dan c∈ *M*. Representasi notasi tersebut dalam sebuah record atau tuple adalah sebagai berikut:

Tabel 3.1 Representasi Maintenance Program Menggunakan Record atau Tuple

No Gen	1	2	3	…	N
AMP_Taskno	70001	70001	70002		70856
Insp_ID	Seq01	Seq03	Seq01	...	Seq32
ManHours	0.5	0.5	0.1		0.1

Nomor record dari tuple ini diidentifikasi dengan nilai dari field GA_Nomor pada tabel Table_MPD_GenAlternatif.

Representasi kromosom dalam bentuk tuple adalah seperti yang tergambar pada tabel di bawah ini.

Tabel 3.2 Representasi Kromosom Menggunakan Record atau Tuple

Chromosome_ID	1	2	3	…	M
Population_ID	1	1	2		N
Fitness_Value	234	55	42	...	P

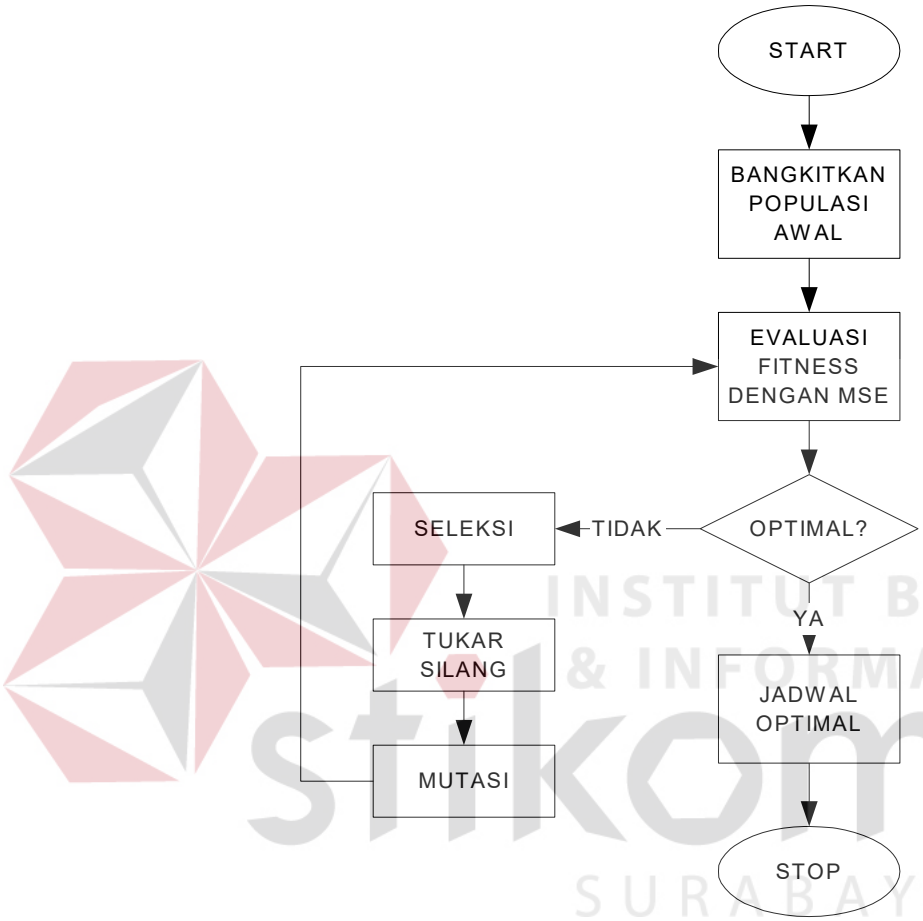
3.1.3 Desain Fungsi Evaluasi

Fungsi evaluasi digunakan untuk mencari sebuah jadwal yang optimum artinya kebutuhan sumber daya manusia akan relatif sama dalam setiap interval, dengan demikian diharapkan terdapat kemudahan dalam pengaturan penggunaan sumber daya manusia.

Penelitian ini akan membuktikan bahwa implementasi GA dapat diterapkan dalam melakukan pendekatan pemenuhan keseimbangan manhours yang diperlukan setiap sekuen dalam perawatan pesawat Fokker F27 dengan menggunakan metode MSE. Dalam penentuan desain fungsi evaluasi, standar dari pabrik dianggap sebagai Target Pengukuran sedangkan hasil rekayasa GA adalah Obyek yang diukur. Secara deterministik fungsi evaluasi ini digambarkan dengan Minimumkan $z = \frac{1}{m}\sum_{i=1}^{m}\left(x_i - T\right)^2$. Individu terbaik adalah z dengan nilai 0. Untuk memberikan gambaran bahwa distribusi dari obyek yang diukur adalah lebih baik dari target pengukuran, maka koefisien variasi dari distribusi obyek yang diukur harus lebih kecil dari 1 atau CV ≤ 1 dimana $CV = \frac{\sqrt{MSE}}{\overline{X}} \times 100$.

3.2 Prosedur GA

Proses pembuatan jadwal dengan menggunakan GA adalah seperti yang terlihat dalam gambar di bawah ini.



Gambar 3.9 Flow Chart Proses Pembuatan Jadwal dengan GA

Proses di atas dapat digambarkan secara pseudocode sebagai berikut:

```
t:=0;
Initialize(P(0));
Evaluate(P(0));
While not good_fitness do
    Select(P(t));
    Crossover(P(t));
    Mutation(P(t));
    Evaluate(P(t));
```

Rancangan Struktur Data di memory untuk proses di atas adalah sebagai berikut:

```
Private
    v_Array_Induk1,  v_Array_Induk2,  v_Array_NoUrutChrDariTrunc: Array[1..2000] of Integer;
    v_Array_MHrs: Array[1..2000] of Double;
```

Keterangan:

1. v_Array_Induk1 dan v_Array_Induk2 adalah array berdimensi satu yang akan digunakan untuk menampung nomor gen dari tabel Table_MPD_GenAlternatif.
2. v_Array_Amp_Taskno adalah array berdimensi satu yang akan digunakan untuk menampung nomor pekerjaan yang didefinisikan sebagai AMP_Taskno.
3. v_Array_MHrs adalah adalah array berdimensi satu yang akan digunakan untuk menampung manhours di setiap nomor pekerjaan dimana nilai ini nanti akan digunakan sebagai penghitung fitness-nya.

Struktur Data array tersebut di atas terutama akan sangat diperlukan pada saat melakukan proses Tukar Silang dan Mutasi.

3.2.1 Pembangkitan Populasi Awal

Populasi dibangkitkan secara urut berdasarkan penomoran pada sistem, demikian pula kromosomnya. Isi dari kromosom adalah merupakan rangkaian gen yang membentuk suatu individu (jadwal). Pembangkitan gen pada jadwal dilakukan secara acak. Langkah pertama adalah membaca data dari database dan melihat interval pelaksanaan pekerjaan. Kemudian, dibangkitkan sebuah angka sebagai awal dari sekuen pekerjaan tersebut diletakkan. Interval dari pekerjaan tersebut dibagi dengan jarak setiap sekuen akan menentukan jarak dari awal pekerjaan menuju ke sekuen paling akhir. Contoh, dipilih satu pekerjaan dengan

nomor X1 dengan interval 1000 jam. Secara acak sistem akan membangkitkan sebuah angka untuk menempatkan pekerjaan tersebut pada posisi awalnya yang besarnya harus lebih kecil dari pembagian antara interval pekerjaan dengan standar sekuennya, dengan kata lain posisi awal pekerjaan tersebut harus lebih kecil dari 1000 : 250 = 4 atau posisi awal pekerjaan harus lebih kecil dari atau sama dengan 4. Penempatan posisi awal pekerjaan ini penting karena apabila posisi awal penempatan pekerjaan pada susunan struktur jadwal tersebut lebih besar dari jarak setiap sekuennya, maka terdapat kemugkinan susunan jadwal menjadi tidak lengkap. Misal pada pekerjaan X1, posisi awal yang dibangkitkan oleh sistem adalah 5, maka dengan interval 1000 jam susunan jadwalnya adalah Seq05, Seq09, Seq13, Seq17, …, Seq29 (berulang setiap 4 kali sekuen). Dalam rentang 8000 jam (lihat tabel 2.1) maka seharusnya pekerjaan X1 dijadwalkan sebanyak 8 kali, tetapi karena penempatan posisi awal berada pada posisi Seq05, maka pekerjaan X1 hanya dijadwalkan sebanyak 7 kali atau terdapat kekurangan 1 kali sekuen. Hal ini harus dihindari untuk mencegah kekacauan sistem penjadwalan.

Setelah penentuan posisi awal pekerjaan dan intervalnya diketahui, maka proses selanjutnya adalah penyimpanan pada table_mpd_genalternatif. Algoritma untuk proses pembangkitan populasi awal adalah sebagai berikut:

```
Cari_Nomor_Terakhir_Chrom(Cari_NoChromAkhir);
  While not  TABLE_MPD_AMP end of file
    V_Pos_Awal_Seq:=0;
    Ambil_Interval_Dari_Tabel(V_AMP_Interval);
    Ambil_Nomor_Pekerjaan_Dari_Tabel(V_Amp_Taskno);
    Ambil_Manhours_Dari_Tabel(V_Amp_MHrs);
    V_JarakAntarSeq:=V_AMP_Interval div 250
    Cari_AwalSeq(V_Pos_Awal_Seq,V_AMP_Interval);
    Mencari_Nomor_Gen_id_Terakhir_di_Table_MPD_GenAlternatif;
    while V_Pos_Awal_Seq<=Std_Jml_Seq do
```

```
            Masukkan_ke_ Table_MPD_GenAlternatif;
            V_Pos_Awal_Seq:= V_Pos_Awal_Seq + V_JarakAntarSeq
```

Keterangan:

1. Cari_Nomor_Terakhir_Chrom, adalah mencari nomor terakhir dari tabel kromosom untuk menentukan nomor kromosom selanjutnya
2. V_Pos_Awal_Seq, adalah variabel yang menyimpan nomor awal sekuen nantinya, V_AMP_Interval adalah variabel yang menyimpan nilai interval dari Maintenance Program yang akan diolah, V_Amp_Taskno adalah nomor pekerjaan, V_Amp_MHrs adalah kebutuhan manhours yang diperlukan untuk mjelakukan pekerjaan dengan nomor V_Amp_Taskno, V_JarakAntarSeq adalah jarak antar sekuen sesuai dengan interval yang ada (misal untuk interval 500 berarti jarak antar sekuen adalah 500 **:** 250 = 2)
3. Cari_AwalSeq, adalah sebuah proses untuk mencari awal sekuen dari pekerjaan V_Amp_Taskno dengan interval V_AMP_Interval.
4. Cari_Nomor_Terakhir_Gen, adalah mencari nomor terakhir dari tabel Table_MPD_GenAlternatif untuk menentukan nomor gen selanjutnya
5. Masukkan Ke Table_MPD_Gen, adalah proses penyimpanan untuk Gen tersebut ke dalam database.

3.2.2 Perhitungan Nilai Fitness

Perhitungan nilai Fitness adalah menggunakan metode MSE dengan proses sebagai berikut:

```
Baca_Table_MPD_Chromosome;
  while not eof do
    Ambil_Nomor_Kromosom_Dari_Tabel(V_Chromosome_ID);
    V_Fitness_Value:=0.00;
    V_Fitness_Seq:=0.00;
    for i:=1 to 32 do
      Hitung_FitnessSeq(V_Chromosome_ID,i,V_Fitness_Seq);
```

```
Cari_Std_MHrs(i,V_MHrs_Std)
V_Fitness_Value:=V_Fitness_Value +
Power((V_Fitness_Seq-V_MHrs_Std),2)
V_Fitness_Chrom:=V_Fitness_Value/Std_Jml_Seq;
Update_Table_MPD_Chromosome(V_Chromosome_ID,
V_Fitness_Chrom);
```

Keterangan:

1. Baca_Table_MPD_Chromosome, adalah proses membaca tabel yang nantinya akan diiterasi.
2. Hitung_FitnessSeq adalah proses penghitungan nilai Fitness dengan menggunakan metode MSE.
3. Update_Table_MPD_Chromosome adalah proses mengubah tabel untuk mengisi nilai fitnessnya.

3.2.3 Seleksi Truncation

Untuk menghindari tingginya kecepatan konvergen dalam proses GA, maka digunakan seleksi Truncation. Proses seleksi tersebut adalah sebagai berikut:

```
Baca_Parameter_Truncation(Parameter_Truncation);
Baca_Jumlah_Individu(Jumlah_Individu);
s_Jml_Individu:=Bulatkan(Parameter_Truncation * Jumlah_Individu);
s_Counter:=1;
while s_Counter<=(s_Jml_Individu) do
    Baca_Table_Chromosome(s_No_ChromAwal, s_Chrom_Fitness);
    Cari_Nomor_Terakhir_Chrom(s_No_ChromAkhir);
    v_Array_NoUrutChrDariTrunc[s_Counter]:=s_No_ChromAkhir;
    Isi_Table_MPD_Chromosome(s_No_ChromAkhir,s_No_ChromAwal,
    s_No_PopAkhir,s_Chrom_Fitness);
    Cari_DataGenDariChrom(s_No_ChromAwal,s_No_ChromAkhir,
    s_No_PopAkhir);
    Next;
```

Keterangan:

1. Baca_Parameter_Truncation(Parameter_Truncation) adalah membaca konstanta parameter truncation berdasarkan input dari user dengan rentang dari 0.2 s/d 0.8.
2. Baca_Jumlah_Individu(Jumlah_Individu) adalah membaca konstata parameter individu sesuai input dari user.
3. Baca_Table_Chromosome(s_No_ChromAwal, s_Chrom_Fitness), adalah proses membaca tabel dan mengurutkan berdasarkan nilai Fitness-nya.
4. Cari_Nomor_Terakhir_Chrom(s_No_ChromAkhir), adalah mencari nomor kromosom baru untuk membentuk individu baru.
5. Isi_Table_MPD_Chromosome(s_No_ChromAkhir, s_No_ChromAwal, s_No_PopAkhir, s_Chrom_Fitness) adalah mengisi nomor kormosom baru di tabel Table_MPD_Chromosome.
6. Cari_DataGenDariChrom(s_No_ChromAwal, s_No_ChromAkhir, s_No_PopAkhir), adalah mencari data Gen yang telah ada di tabel Table_MPD_GenAlternatif untuk disalin pada individu baru sekaligus menyalinnya ke gen baru.

3.2.4 Proses Crossover

Proses tukar silang dilakukan dengan menggunakan metode Uniform Crossover karena panjang Gen yang cukup besar. Proses Tukar Silang dilakukan melalui array v_Array_Induk1, v_Array_Induk2. Proses tukar silang tersebut adalah sebagai berikut:

```
Baca_Jumlah_Gen_Dalam_Kromosom(Jml_Gen);
Hitung_Jumlah_Kekurangan_Individu(Jml_KekuranganIndividu);
For i:=1 to Jml_KekuranganIndividu
    Masukkan_Data_Gen_Pertama_Ke_Dalam_Array_Induk1
    Masukkan_Data_Gen_Pertama_Ke_Dalam_Array_Induk2
    For j:=1 to Jml_Gen
        Bangkitkan_Nilai_Acak_Antara_0_Dan_1(Nilai_Acak);
        If  Nilai_Acak=1 then Array_Induk1[j]:= Array_Induk2[j];
```

Keterangan:

1. Baca_Jumlah_Gen_Dalam_Kromosom(Jml_Gen), adalah membaca jumlah gen yang akan dilakukan tukar silang.
2. Hitung_Jumlah_Kekurangan_Individu(Jml_KekuranganIndividu), adalah menghitung banyaknya individu baru yang akan dibangkitkan dari tukar silang.
3. Masukkan_Data_Gen_Pertama_Ke_Dalam_Array_Induk1 dan Masukkan_Data_Gen_Pertama_Ke_Dalam_Array_Induk2, adalah proses untuk menyalin data dari tabel Table_MPD_GenAlternatif ke Array_Induk1 dan Array_Induk2 untuk memudahkan proses tukar silang.
4. Bangkitkan_Nilai_Acak_Antara_0_Dan_1(Nilai_Acak), adalah proses membangkitkan bilangan random antara 0 dan 1 dimana apabila Nilai_Acak bernilai 1, maka akan dilakukan Tukar Silang. Dalam hal ini untuk efisiensi memory Array_Induk1 langsung dijadikan sebagai off spring atau individu baru dengan cara menyalin nilai dari Array_Induk2.

3.2.5 Mutasi

Untuk efisiensi pemrograman, proses mutasi dilakukan bersamaan dengan proses tukar silang. Proses mutasi ini dilakukan pada setiap gen dalam kromosom terpilih. Proses mutasi dilakukan dalam Array dengan tingkat

probabilitasnya adalah sebesar 0.01. Proses mutasi tersebut adalah sebagai berikut:

```
Baca_Jumlah_Gen_Dalam_Kromosom(Jml_Gen);
For i:=1 to Jml_Gen
    Bangkitkan_Bilangan_Pecah_Secara_Acak_Antara_0_Dan_1(Nilai_Aca
    k)
    If Nilai_Acak < 0.01 then Lakukan_Mutasi(Array_Induk1[i]);
    Pindahkan_Data_Dari_Array_Induk1_ke_Table_MPD_GenAlternatif;
```

Keterangan:

1. Baca_Jumlah_Gen_Dalam_Kromosom(Jml_Gen), adalah membaca panjang gen yang akan dilakukan tukar silang.
2. Bangkitkan_Bilangan_Pecah_Secara_Acak_Antara_0_Dan_1(Nilai_Acak), adalah sebuah proses membangkitkan bilangan acak dari 0.00 s/d 1.00.
3. Jika bilangan acak yang dibangkitkan kurang dari 0.01 maka akan dilakukan mutasi dalam array.
4. Pindahkan_Data_Dari_Array_Induk1_ke_Table_MPD_GenAlternatif, adalah proses memindahkan data dari Array 1 ke dalam tabel Table_MPD_GenAlternatif.

3.2.6 Pemilihan Jadwal Optimal

Proses ini adalah merupakan pemilihan individu terbaik untuk dijadikan sebagai Jadwal. Proses ini sebenarnya adalah mencari nilai terkecil dari kumpulan individu yang telah dihasilkan dari proses GA. Prosesnya dalah sebagai berikut

```
Baca_Tabel_Kromosom
While not eof
    If  Nilai_Fitness1 < Nilai_Fitness2 then
    Nilai_Fitness1= Nilai_Fitness2;
    Next;
Pilih_Individu_Terbaik;
```

Keterangan:

1. Baca_Tabel_Kromosom, adalah proses menyalin nilai Fitness ke dalam variabel memori.
2. Pilin_Individu_Terbaik adalah proses memindahkan data individu terbaik ke dalam tabel jadwal.

3.3 Rancangan Evaluasi

Berdasarkan data dari database Access, maka tabel-tabel di bawah ini adalah tabel-tabel pendukung dalam proses GA.

1. Table_MPD_Amp
2. Table_MPD_Population
3. Table_MPD_Chromosome
4. Table_MPD_GenAlternatif
5. Table_MPD_Rel_MPDAMP_MtopF27Sched

3.3.1 Menentukan Kapan Algoritma Berhenti

Dalam proses GA, proses akan berhenti apabila telah dicapai solusi yang optimal atau dihentikan dengan metode tertentu (dengan menentukan jumlah individu yang diinginkan).

Dalam metode MSE, nilai optimal didapatkan apabila tidak terdapat perbedaan antara jadwal yang dihasilkan oleh GA dengan standar yang telah ada (target pengukuran) seperti dalam Tabel 2.2, atau dengan kata lain tidak terdapat perbedaan yang signifikan dengan nilai MSE = 0 dan nilai CV ≤ 1. Namun demikian, untuk efisiensi waktu dalam percobaan dan memudahkan perbandingan nilai fitness dari truncation parameter yang berubah-ubah, maka dalam percobaan

nanti algoritma akan dihentikan pada 250 generasi untuk setiap pemilihan nilai truncation selection-nya.

3.3.2 Rancangan Percobaan

Percobaan dilakukan dengan menggunakan aplikasi yang telah dibuat untuk melakukan proses pembangkitan jadwal dengan menggunakan metode GA. Aplikasi tersebut sekaligus sebagai sistem informasi maintenance program yang akan digunakan untuk melakukan manajemen atas maintenance program tersebut.

Evaluasi terhadap hasil percobaan adalah berdasarkan pengujian beberapa kali dengan mengubah nilai Truncation-nya. Hasil dari pengujian tersebut akan dibandingkan satu dengan yang lain untuk mengetahui kinerja aplikasi dihubungkan besaran truncation-nya.