



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS091336

Optimasi Penjadwalan Proyek Menggunakan Metode Algoritma Genetika

Khairil Juhdi Siregar
NRP 5209 100 710

Dosen Pembimbing I
Wiwik Anggraeni, S.Si, M.Kom

Dosen Pembimbing II
Amalia Utamima, S.Kom, MBA

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014

FINAL PROJECT - KS091336

Optimization of Project Scheduling Using Genetic Algorithm Method

Khairil Juhdi Siregar
NRP 5209 100 710

Academic Promotor I
Wiwik Anggraeni, S.Si, M.Kom

Academic Promotor II
Amalia Utamima, S.Kom, MBA

DEPARTMENT OF INFORMATION SYSTEM
Faculty of Information Technology
Institute of Technology Sepuluh Nopember
Surabaya 2014

Optimasi Penjadwalan Proyek Menggunakan Metode Algoritma Genetika

Nama Mahasiswa : KHAIRIL JUHDI SIREGAR
NRP : 5209 100 710
Jurusan : SISTEM INFORMASI FTIF-ITS
Dosen Pembimbing I : Wiwik Anggraeni, S.Si, M.Kom
Dosen Pembimbing II : Amalia Utamima, S.Kom, MBA

Abstrak

Masalah penjadwalan termasuk ke dalam kelas permasalahan NP-hard dengan masalah optimasi kombinatorial yang sangat kompleks. Penjadwalan proyek terdiri dari beberapa aktivitas yang antara aktivitas satu dengan yang lainnya saling berhubungan. Jumlah dan hubungan antar aktivitas ini membuat ruang pencarian solusi untuk permasalahan penjadwalan proyek sangat luas. Dibutuhkan sebuah teknik atau metode optimasi untuk menemukan fungsi tujuan dari permasalahan. Fungsi tujuan yang dimaksud dalam tugas akhir ini adalah meminimalkan durasi proyek. Alba dan Chicano telah menunjukkan bahwa Algoritma Genetika cukup fleksibel dan akurat untuk penjadwalan proyek, dan dianggap sebagai alat penting untuk manajemen proyek otomatis.

Dengan memanfaatkan metode Algoritma Genetika, tugas akhir ini berusaha mengoptimalkan permasalahan penjadwalan proyek dengan fungsi tujuan meminimalkan durasi proyek. Nilai evaluasi per kromosom adalah fungsi tujuan dari permasalahan. Proses seleksi dilakukan pada setengah dari jumlah individu di dalam populasi. Kemudian individu-individu hasil penyeleksian mengalami proses genetika seperti perkawinan silang dan mutasi.

Berdasarkan hasil perhitungan tersebut didapatkan hasil yang optimal atau durasi penyelesaian proyek paling minimum adalah 38 minggu.

Kata Kunci: *Penjadwalan , Algoritma Genetika, Menejemen Proyek, Optimasi.*

Optimisation of Project Scheduling Using Genetic Algorithm Methode

Name : KHAIRIL JUHDI SIREGAR
NRP : 5209 100 710
Departement : INFORMATION SYSTEM FTIF-ITS
Supervisor I : WIWIK ANGGRAENI, S.Si, M.Kom
Supervisor II : AMALIA UTAMIMA, S.Kom, MBA

Abstract

Scheduling problem is a NP-Hard class optimization with a complex optimization combinatory. Project Scheduling instead of some activities wich related each other. Sum of activity and the arcs that related activity with another make the space of searching solution for project scheduling problem is enormous. Optimization technic or method is needed to find the value of objective function of problem. The objective function that aim of this final project is to minimize duration of project. Alba and Chicano have showed that Genetic Algorithm was flexible and accurate for scheduling project, and considered as an important technic in otomatic project management.

Using Genetic Algorithm method, this final project attempts to optimize the project scheduling problem with minimize duration of project as objective function. Evaluation value each chromosome is the objective function of problem. Selection process was held for half of individu in population. Therefor, the individues were relusted selection process going to pass genetic process as crossover process or mutation process.

Based on the counting result found an optimal solution or the minimum project completion time was 38 weeks.

Keyword : Scheduling, Genetic Algorithm, Project Management, Optimization.

Halaman ini sengaja dikosongkan

**OPTIMASI PENJADWALAN PROYEK
MENGUNAKAN METODE
ALGORITMA GENETIKA**

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

KHAIRIL JUHDI SIREGAR
NRP.5209 100 710

Surabaya, Juli 2014

Ketua Jurusan Sistem Informasi

Dr. Eng. FEBRILIYAN SAMOPA, S.Kom., M.Kom.
NIP.19730219 199802 1 001

**OPTIMASI PENJADWALAN PROYEK
MENGUNAKAN METODE
ALGORITMA GENETIKA**

TUGAS AKHIR

**Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer**

pada

**Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember**

Oleh :

KHAIRIL JUHDI SIREGAR

5209 100 710

**Disetujui Tim Penguji : Tanggal Ujian
Periode Wisuda**

: 10 Juli 2014

: September 2014


Wiwik Anggraeni, S.Si, M.Kom


(Pembimbing I)

Amalia Utamima, S.Kom, MBA


(Pembimbing II)

Mahendrawathi ER, S.T, M.T, Ph.D


(Penguji 1)

Irmasari Hafidz, S.Kom, M.Sc


(Penguji 2)

KATA PENGANTAR

Puji syukur kepada Allah SWT, karena dengan rahmat dan ridho-Nya penulis dapat menyelesaikan laporan tugas akhir yang berjudul “OPTIMASI PENJADWALAN PROYEK MENGGUNAKAN METODE ALGORITMA GENETIKA” sebagai salah satu syarat kelulusan di Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya. Dengan terselesaikannya tugas akhir ini, maka selesai pula masa studi penulis yang telah ditempuh selama lima tahun.

Penulis sadar bahwa dalam proses pengerjaan sampai terselesaikannya tugas akhir ini dibutuhkan bantuan dari berbagai pihak dan penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Kedua orang tua penulis, ayah H. Timbul Siregar, S.Ag dan ibu Hj. Derliana Pulungan, abang H. Tunggul Mulia Siregar, kedua adik Santri Syahri Melina Siregar, Am.Keb, dan Irhamiyah Kemil Kamelina Siregar, yang dengan segala ketulusan hati menyebut nama penulis di setiap doa, serta telah memberikan semangat dan motivasi sehingga penulis mampu menyelesaikan pendidikan S1 dengan baik.
2. Seluruh Bapak dan Ibu dosen yang telah membagi banyak ilmu kepada penulis sejak awal masuk sebagai mahasiswa serta seluruh staf di Jurusan Sistem Informasi yang telah memberikan banyak sekali bantuan selama penulis berkuliah.
3. Rully Hendra, ST, M.Si, yang selama sepuluh semester memberikan perhatian dan bimbingan sebagai Dosen Wali penulis.

4. Wiwik Anggraeni, S.Si, M.Kom dan Amalia Utamima, S.Kom, MBA yang selama tiga semester membimbing, mengarahkan serta memotivasi penulis agar dapat menyelesaikan tugas akhir ini.
5. Teman-teman terbaik penulis, Rahmat, Hamzah, Hasry, Faisal, Imam, dan juga abang-abang, kakak, dan adik-adik anggota Parsadaan Naposo Nauli Bulung Tapanuli Bagian Selatan, POSNIROHA.
6. Terima kasih juga kepada Iin Citra Liana Hasibuan yang rela menyisihkan waktunya untuk saling berbagi dengan penulis baik itu semangat dan motivasi. Begitu juga teman-teman D'Juvenile yang sama-sama berusaha menggapai cita-cita di kampus masing-masing.
7. Rekan – rekan mahasiswa Jurusan Sistem Informasi khususnya AEGIS 2009 yang telah memberikan motivasi, spirit dan bantuan selama pengerjaan Tugas Akhir.
8. Serta semua pihak yang terlibat dalam pengerjaan Tugas Akhir ini yang belum mampu penulis sebutkan diatas.

Terima kasih atas segala bantuan, dukungan, serta doanya. Semoga Keberkahan dan Rahmat dari Allah SWT dilimpahkan kepada orang tua, dosen, guru, teman-teman, dan segenap yang telah menjadi motivator, dan inspirator penulis selama menjalankan tugas sebagai mahasiswa di Surabaya. Amin Ya Robbana.

Surabaya, Juli 2014

Penulis

Daftar Isi

Abstrak	v
KATA PENGANTAR.....	viii
Daftar Gambar	xvi
Daftar Tabel.....	xix
BAB I.....	xx
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	2
1.3 Batasan Tugas Akhir.....	3
1.4 Tujuan Tugas Akhir	3
1.5 Relevansi atau Manfaat Kegiatan Tugas Akhir	3
BAB II.....	5
DASAR TEORI	5
2.1 Optimasi	5
2.1.1 <i>Local</i> dan <i>Global Optimum</i>	6
2.2 Penjadwalan Proyek	6
2.2.1 <i>Earliest Start</i> dan <i>Earliest Finish</i>	7
2.2.2 <i>Latest Start</i> dan <i>Latest Finish</i>	9
2.3 Algoritma Genetika.....	10
2.3.1 Kromosom	12
2.3.2 <i>Fitness</i>	12
2.3.3 Elitisme.....	12
2.3.4 <i>Crossover</i> atau Kawin Silang.....	13
2.3.5 Mutasi.....	15

BAB III.....	17
METODOLOGI Pengerjaan Tugas Akhir	17
3.1 Identifikasi Permasalahan.....	19
3.2 Studi Literatur.....	19
3.3 Pengolahan Data.....	20
3.4 Desain dan Implementasi	20
3.4.1 Formulasi Model Penjadwalan Proyek.....	20
3.4.2 Desain Algoritma Genetika	20
3.4.2.1 Representasi Kromosom	21
3.4.2.2 Kontruksi Penjadwalan Proyek	22
3.4.2.3 Proses Evolusi Generasi Selanjutnya	23
3.4.2.4 Pengecekan Kualitas Kromosom.....	25
3.4.3 Pembuatan Kode Program Algoritma Genetika pada Perangkat Lunak Matlab	25
3.4.4 Pembuatan Kode Program PERT pada Perangkat Lunak Matlab... 25	
3.4.4 Penerjemahan Model ke Perangkat Lunak Matlab	26
3.5 Running Program.....	26
3.6 Data Uji Coba	26
3.7 Verifikasi.....	26
3.8 Validasi	27
3.9 Uji Coba dan Analisis Hasil	27
3.9.1 Skenario Uji Coba	27
3.10 Kesimpulan dan Saran.....	28
3.11 Penyusunan Buku Tugas Akhir.....	28
BAB IV.....	29
DESAIN DAN IMPLEMENTASI MODEL.....	29

4.1	Data pengerjaan proyek	29
4.1.1	Aktivitas dan Urutan Pengerjaan Proyek	29
4.1.2	Durasi dan Kebutuhan Sumber Daya.....	31
4.1.3	Kapasitas Sumber Daya	35
4.2	Formulasi Model.....	35
4.2.1	Penentuan Fungsi Tujuan.....	35
4.2.2	Penentuan Batasan 1	37
4.2.3	Penentuan Batasan 2	37
4.2.4	Penentuan Batasan 3	37
4.3	Metodologi Penyelesaian Model dengan Perangkat Lunak Matlab	38
4.3.1	Inisialisasi Data	38
4.3.2	Mencari Nilai Durasi Proyek pada Algoritma Genetika	40
4.3.3	Menentukan <i>cost</i> (biaya) setiap Individu	41
4.3.4	Mengatur Algoritma Genetika.....	42
4.3.5	Inisialisasi Populasi	43
4.3.6	Proses Kawin Silang.....	44
4.3.7	Proses Mutasi	46
4.3.8	PERT	47
4.3.9	Memasukkan Batasan Model	48
4.4	Implementasi Penjadwalan Proyek dengan Algoritma Genetika ke dalam Perangkat Lunak Matlab.....	52
4.4.1	Inisialisasi Data	52
4.4.2	Mencari Waktu Mulai dan Waktu Selesai.....	53
4.4.3	Mencari <i>cost</i> (biaya) per Kromosom	54
4.4.4	Inisialisasi Algoritma Genetika	54
4.4.5	Parameter Algoritma Genetika.....	55

4.4.6	Inisialisasi Populasi	56
4.4.7	Proses Genetika	57
4.4.7.1	Perkawinan Silang	57
4.4.8	Menampilkan Keluaran Proses Algoritma Genetika	58
4.5	Implementasi Penjadwalan Proyek dengan PERT ke dalam Perangkat Lunak Matlab	59
4.5.1	Inisialisasi Data	60
4.5.2	Mencari Nilai ES dan EF	61
4.5.3	Menampilkan Keluaran PERT	62
4.5.4	Batasan	62
BAB V	69
UJI COBA DAN ANALISIS HASIL	69
5.1	Lingkungan Uji Coba	69
5.2	Verifikasi	70
5.2.1	Menjalankan Program Algoritma Genetika	70
5.2.2	Menjalankan Program PERT	74
5.3	Validasi	75
5.3.1	Validasi terhadap Durasi Pelaksanaan Proyek	76
5.3.2	Validasi Terhadap Penggunaan Metode	78
5.4	Skenario Uji Coba	79
5.4.1	Mengurangi Jumlah Individu di dalam Populasi	79
5.4.2	Menurunkan probabilitas mutasi	80
5.5	Analisis Hasil	81
5.5.1	Hasil Optimasi	81
5.5.2	Batasan Pertama	82
5.5.3	Batasan Kedua	83

5.5.4 Batasan Ketiga	84
BAB VI.....	95
PENUTUP	95
DAFTAR PUSTAKA.....	97
RIWAYAT PENULIS	99
LAMPIRAN A.....	A-1
DATA INPUTAN	A-1
LAMPIRAN B	B-1
HASIL OPTIMASI	B-1

Halaman ini sengaja dikosongkan

Daftar Tabel

Tabel 4.1Daftar Aktivitas dan Urutan Aktivitas.....	30
Tabel 4.2 Daftar Durasi dan Kebutuhan	31
Tabel 4.3Daftar Kapasitas Sumber Daya	35
Tabel 5.1.1 Perangkat Keras Lingkungan Uji Coba	69
Tabel 5.1.2 Perangkat Keras Lingkungan Uji Coba	69
Tabel 5.2 1 Waktu Mulai dan Waktu Selesai	70
Tabel 5.2 2 Earliest Start dan Earliest Finish	74
Tabel 5.3.1 Perbandingan Hasil	78
Tabel 5.3.2 Perbandingan Running Time	78
Tabel 5.4 1 Perbandingan hasil uji coba	81
Tabel 5.5.2. 1 Batasan 1.....	82
Tabel 5.5.2. 2 Batasan 2.....	83
Tabel 5.5.2. 3 sumber daya 1	84
Tabel 5.5.2. 4 sumber daya 2	85
Tabel 5.5.2. 5 sumber daya 3	86
Tabel 5.5.2. 6 sumber daya 4	88
A-1 1. Data Pengerjaan Proyek	A-1
A-1 2 Sumber daya yang dibutuhkan	A-2
Tabel B- 1 <i>Output</i> Algoritma Genetika	B-1
Tabel B- 2 <i>Output</i> program PERT.....	B-2
Tabel B- 3 Batasan pertama dan kedua.....	B-4
Tabel B- 4 Batasan Ketiga.....	B-5

Halaman ini sengaja dikosongkan

Daftar Gambar

Gambar 3.4.2 1 Proses Algoritma Genetika.....	21
Gambar 3.4.2 2 Kromosom.....	22
Gambar 3.4.2 3 Proses Evolusi	24
Gambar 4.3 1 Diagram Alur Inisialisasi Data	40
Gambar 4.3 2 Diagram Alur mencari waktu mulai dan waktu selesai.....	41
Gambar 4.3 3 Diagram alur <i>cost</i> kromosom	42
Gambar 4.3 4 Inisialisasi parameter GA	43
Gambar 4.3 5 Diagram Alur Inisialisasi Populasi	44
Gambar 4.3 6 Diagram alur perkawinan silang	45
Gambar 4.3 7 Diagram Alur proses Mutasi	46
Gambar 4.3 8 Mencari nilai ES dan EF	48
Gambar 4.3.9 1 Diagram Alur batasan pertama	49
Gambar 4.3.9 2 Diagram Alur batasan kedua	50
Gambar 4.3.9 3 Diagram Alur batasan ketiga	51
Gambar 4.4. 1 Inisialisasi Data	52
Gambar 4.4. 2 Waktu mulai dan Waktu selesai	53
Gambar 4.4. 3 <i>cost</i> per kromosom.....	54
Gambar 4.4. 4 Inisialisasi Algoritma Genetika	55
Gambar 4.4. 5 Parameter Algoritma Genetika	55
Gambar 4.4. 6 Inisialisasi Populasi	56
Gambar 4.4. 7 Seleksi dan Perkawinan Silang.....	57
Gambar 4.4. 8 Mutasi	58
Gambar 4.4. 9 Menampilkan <i>Output</i>	59
Gambar 4.5 1 Inisialisasi Data	60
Gambar 4.5 2 Mencari nilai ES dan EF	61
Gambar 4.5 3 Menampilkan output PERT.....	62
Gambar 4.5 4 Batasan 1	63
Gambar 4.5 5 Batasan 2.....	65
Gambar 4.5 6 Batasan 3.....	66
Gambar 4.5 7 Batasan 3.....	67
Gambar 4.5 8 Batasan 3.....	68
Gambar 4.5 9 Batasan 3.....	68
Gambar 5.2. 1 Waktu mulai dan waktu selesai setiap aktivitas.....	72
Gambar 5.2.2 Individu terbaik dan lama pengerjaan proyek.....	73
Gambar 5.2.3 Grafik konvergensi generasi.....	73

Gambar 5.2.4 Durasi proyek	75
Gambar 5.3.1 keluaran metode Algoritma Genetika	77
Gambar 5.3.2 keluaran metode PERT	77
Gambar 5.3.3 kode program perhitungan running time	78
Gambar 5.4.1 Skenario pertama	80
Gambar 5.4.2 Skenario kedua	80
Gambar 5.5. 1 Batasan Pertama	89
Gambar 5.5. 2 Batasan Kedua.....	90
Gambar 5.5. 3 Batasan Ketiga sumber daya 1.....	91
Gambar 5.5. 4 Batasan Ketiga sumber daya 2.....	92
Gambar 5.5. 5 Batasan Ketiga sumber daya 3.....	93
Gambar 5.5. 6 Batasan Ketiga sumber daya 4.....	94

BAB I

PENDAHULUAN

Pada pendahuluan ada lima topik utama yang tercakup di dalamnya yaitu latar belakang, rumusan masalah, batasan masalah, tujuan tugas akhir, dan manfaat tugas akhir. Latarbelakang mencakup sebagian besar inti pokok dari isi tugas akhir ini. Rumusan masalah yaitu berupa beberapa pertanyaan-pernyataan yang akan dibahas atau dijawab pada pembahasan. Batasan masalah merupakan batasan-batasan yang diberikan dalam menyelesaikan tugas akhir ini. Tujuan sangat berkaitan dengan rumusan masalah, yaitu menjawab pertanyaan yang ada pada rumusan masalah. Sedangkan manfaat berguna untuk melihat harapan yang bias didapat dalam melaksanakan tugas akhir ini.

5.5. Latar Belakang

Penjadwalan proyek memerlukan waktu untuk penugasan kegiatan dalam perencanaan proyek dengan tujuan meminimalkan durasi proyek. Masalah tersebut muncul dalam berbagai bidang termasuk konstruksi, manufaktur, pengembangan perangkat lunak dan sejumlah aplikasi. Karena pentingnya bidang-bidang tersebut, masalah penjadwalan proyek telah banyak diteliti dengan baik oleh kalangan OR (*Operational Research*) dan AI (*Artificial Intelligence*) (M.Pyle, 2004).

Selanjutnya, penjadwalan proyek termasuk ke dalam kelas optimasi bertipe *NP-hard*, yang berarti bahwa tidak mungkin menemukan sebuah algoritma yang paling efisien untuk menyelesaikan pengoptimalan dengan data yang berukuran besar saat melakukan komputasi. Sehingga dibutuhkan penggunaan *heuristic* ketika menyelesaikan permasalahan dengan ukuran data yang besar (Omara, 2009). Penelitian-penelitian yang bersangkutan dengan permasalahan ini membahas tentang bagaimana membuat jadwal yang bagus dengan mengutamakan durasi proyek seminimal mungkin.

Halaman ini sengaja dikosongkan

Generalisasi ini banyak dikenal dan dipelajari dengan baik dalam kelas masalah penjadwalan (termasuk masalah *job-shop* dengan beberapa variasinya). Metode Algoritma Genetika juga telah banyak dilakukan oleh peneliti-peneliti terdahulu. Beberapa kelebihan dari algoritma genetika dalam penyelesaian masalah antara lain dapat mengoptimasi 2able22e diskrit maupun kontinyu, tidak bermasalah dengan jumlah data yang besar, tidak hanya menyajikan satu solusi namun daftar solusi permasalahan. Untuk itu, Algoritma Genetika banyak digunakan untuk menyelesaikan permasalahan penjadwalan proyek, baik algoritma genetika murni, maupun digabungkan dengan metode-metode yang lain.

(E Alba, 2004) telah menunjukkan bahwa Algoritma Genetika cukup fleksibel dan akurat untuk penjadwalan proyek, dan dianggap sebagai alat penting untuk manajemen proyek otomatis . Mereka menyediakan ide dasar pada penerapan GA untuk tugas tugas otomatis (Carl). Latar belakang permasalahan penjadwalan proyek yang membutuhkan sebuah algoritma untuk mengoptimalkan durasi dalam pengerjaan proyek, dan dengan kelebihan-kelebihan yang dimiliki Algoritma Genetika dalam penyelesaian masalah, maka tugas akhir ini akan mengimplementasikan metode Algoritma Genetika.

Pada tugas akhir ini, kromosom direpresentasikan dengan sebuah matriks yang berisi urutan aktivitas, waktu mulai, dan waktu selesai dari aktivitas. Pembentukan populasi dilakukan secara acak sepanjang aktivitas yang ada. Kemudian dilakukan perhitungan waktu mulai dan waktu selesai dari setiap aktivitas. Pada tahap pembentukan generasi selanjutnya dilakukan proses perkawinan silang sederhana dan dilanjutkan dengan proses mutasi.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan sebelumnya, masalah yang akan diangkat pada Tugas Akhir ini adalah:

1. Bagaimana memodelkan permasalahan penjadwalan proyek

2. Bagaimana menentukan hubungan antar aktivitas
3. Bagaimana menemukan solusi optimal dari permasalahan

1.3 Batasan Tugas Akhir

Batasan permasalahan pada tugas akhir ini adalah :

1. Data yang digunakan pada tahap implementasi diambil dari *Project Scheduling Problem Library* (PSPLIB), yaitu, www.om-db.wi.tum.de/psplib/data.html (Kolisch, 1996).
2. Jenis sumber daya yang digunakan disesuaikan dengan sumber daya pada data yang diambil dari PSPLIB

1.4 Tujuan Tugas Akhir

Berdasarkan rumusan masalah yang telah dijelaskan sebelumnya, tujuan tugas akhir ini adalah :

1. Memodelkan permasalahan penjadwalan proyek
2. Menentukan hubungan antar aktivitas sehingga didapatkan durasi yang minimal
3. Mengimplementasikan proses-proses algoritma genetika untuk menemukan solusi optimal dari permasalahan

1.5 Relevansi atau Manfaat Kegiatan Tugas Akhir

Manfaat yang ingin dicapai dari Tugas Akhir ini adalah diharapkan dapat meningkatkan pemahaman tentang penggunaan atau pengimplementasian metode Algoritma Genetika dalam permasalahan penjadwalan proyek.

Halaman ini sengaja dikosongkan

BAB II

DASAR TEORI

Pada bagian ini dipaparkan teori-teori serta pustaka yang dipakai pada waktu penelitian. Teori-teori ini diambil dari buku 5able55e5g dan dari internet. Teori yang dibahas meliputi teori tentang Penjadwalan Proyek, Optimasi, dan Algoritma Genetika.

5.5. Optimasi

Optimasi digunakan untuk menerjemahkan kata *optimization*. Sebenarnya lebih tepat menggunakan kata optimisasi, tetapi karena istilah optimasi ini sudah sering digunakan, maka dipilihlah kata ini. Optimasi bukan ilmu baru. Ia sudah berkembang bahkan sejak Newton pada abad 17 menemukan cara menghitung akar. Saat inipun ilmu optimasi masih berkembang dalam hal teknik maupun aplikasi. Banyak kasus atau masalah dalam kehidupan sehari-hari yang melibatkan optimasi untuk memecahkannya(Santosa, 2011).

Belakangan ini banyak berkembang terutama setelah munculnya teknik-teknik baru untuk menyelesaikan masalah optimasi. Optimasi memegang peranan penting dalam proses mendesain suatu sistem. Dengan optimasi, desain suatu sistem bisa menghasilkan ongkos yang lebih murah atau profit yang lebih tinggi, menurunkan waktu proses dan sebagainya.

Keberhasilan penerapan teknik optimasi, paling tidak memerlukan tiga syarat. Syarat-syarat tersebut adalah kemampuan membuat model matematika dari permasalahan yang dihadapi, pengetahuan teknik optimasi dan pengetahuan akan program tertentu. Pengertian dari optimasi bisa dijelaskan sebagai suatu kumpulan formula matematis dan metoda numerik untuk menemukan dan mengidentifikasi kandidat terbaik dari sekumpulan alternatif tanpa harus secara eksplisit menghitung dan mngevaluasi semua alternatif yang memungkinkan(Santosa, 2011).

5.5.4 Local dan Global Optimum

Jika suatu permasalahan optimasi atau fungsi matematik hanya mempunyai satu titik optimum dengan nilai paling baik dibanding dengan titik yang lain, maka persoalan ini dapat dinamakan optimasi global (*global optimization*). Masalahnya adalah bagaimana mencari titik terbaik tersebut. Fungsi yang lain juga bisa memiliki nilai local optimal yang nilai fungsi objektifnya bisa berbeda atau sama. Jika ini yang terjadi maka ini masuk kategori optimasi lokal (*local optimization*). Titik yang ditemukan hanya bersifat lokal karena keoptimalannya hanya berlaku untuk area disekitarnya tidak untuk keseluruhan permasalahan.

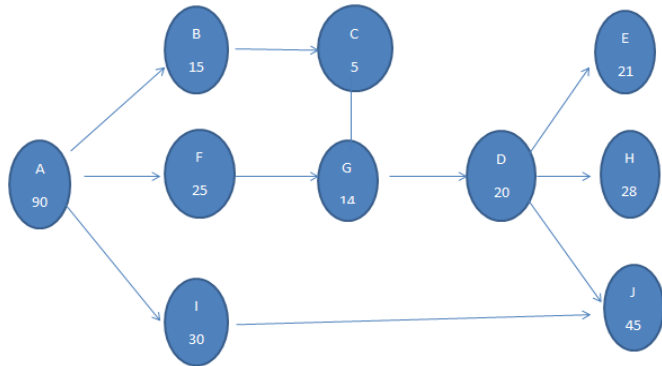
2.2 Penjadwalan Proyek

Proyek dapat diartikan sebagai kumpulan dari beberapa tugas, orang, atau sumber daya dengan tujuan untuk meminimalkan waktu atau untuk meminimalkan biaya (A.Lawrence, 2002). Contohnya untuk mengecat sebuah rumah, seorang kontraktor harus mengerjakan tugas-tugas seperti: 1) memilih warna, 2) membeli cat, 3) membersihkan area yang ingin di cat, 4) mengecat dinding, dan 5) membersihkannya.

Tugas-tugas dari sebuah proyek disebut aktivitas. Untuk mengestimasi penyelesaian proyek selalu dihubungkan dengan setiap aktivitas – aktivitas yang ada dalam proyek tersebut. Aktivitas dalam sebuah proyek gabl saja sedikit maupun banyak, tergantung dari situasi dan jenis proyeknya.

Dalam sebuah proyek, ada beberapa aktivitas yang harus terlebih dahulu dikerjakan sebelum aktivitas sebelumnya dimulai. Ada juga aktivitas-aktivitas yang dilakukan secara bersamaan. Ketepatan relasi atau hubungan aktivitas yang satu dengan aktivitas yang lainnya sangatlah penting guna mengoptimalkan jadwal untuk setiap aktivitas itu sendiri.

Selain berhubungan dengan durasi aktivitas-aktivitas dalam sebuah proyek juga direalisasikan dengan sumber daya yang merujuk ke aktivitas tersebut. Contohnya untuk mengecat dinding diperlukan sumber daya seperti : cat, kuas, dan lain-lain.



Gambar 2.1 1 Diagram Jaringan Aktivitas

Gambar 2.1.1 menunjukkan aktivitas-aktivitas yang telah berelasi antara satu dengan yang lainnya, mulai dari aktivitas A hingga J dengan durasi masing-masing aktivitas. Durasi pada masing-masing aktivitas merupakan seberapa lama waktu yang diperlukan untuk menyelesaikan aktivitas tersebut. Kemudian akan dihitung waktu mulai tercepat (*earliest start time*), waktu selesai tercepat (*earliest finish time*), waktu terlama mulai (*latest start time*), dan waktu selesai terlama (*latest finish time*).

2.2.1 Earliest Start dan Earliest Finish

Untuk menghitung waktu mulai tercepat (ES) dan waktu selesai tercepat (EF) dari aktivitas yang tidak memiliki penghubung yang menuju kepada aktivitas tersebut. Dalam hal ini dimulai dengan aktivitas A. Aktivitas A memiliki nilai ES (A) = 0, karena merupakan aktivitas pemula. Sedangkan untuk nilai EF (A) = 90, yaitu durasi waktu yang diperlukan untuk menyelesaikan aktivitas tersebut.

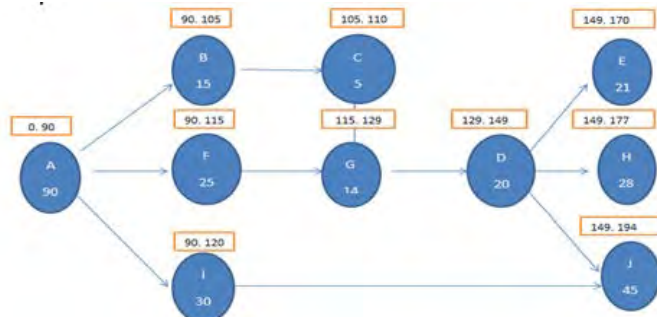
Aktivitas B, F, dan I berhubungan langsung dengan aktivitas A, sehingga nilai ES dari aktivitas-aktivitas tersebut adalah EF dari aktivitas A. Karena aktivitas B, F, dan I tidak akan 7abl dimulai

sebelum aktivitas A selesai. Sehingga $ES(B), (F), (I) = 90$. Sedangkan untuk nilai EF dari masing-masing aktivitas B, F, dan I adalah nilai ES dari masing-masing aktivitas dijumlahkan dengan durasi waktu yang diperlukan untuk menyelesaikan aktivitas tersebut. Jadi $EF(B) = 90 + 15 = 105$, $EF(F) = 90 + 25 = 115$, dan $EF(I) = 90 + 30 = 120$.

Adapun untuk aktivitas yang terhubung dengan lebih dari satu aktivitas maka dalam penentuan ES dari aktivitas tersebut memilih EF yang paling besar/maksimum dari aktivitas-aktivitas yang terhubung dengannya. Seperti aktivitas G yang terhubung dengan aktivitas C dan F. Maka untuk penentuan $ES = \text{MAX}(EF(C), EF(F)) = \text{MAX}(110, 115) = 115$. Sedangkan untuk EF dari aktivitas G $= 115 + 14 = 129$.

ES = Maksimum EF dari masing-masing aktivitas yang menuju kepada aktivitas tersebut
 EF = ES + (durasi waktu penyelesaian aktivitas)

Dengan menggunakan cara yang sama untuk aktivitas-aktivitas selanjutnya, maka akan didapatkan :



Gambar 2.1 2 Earliest Start dan Earliest Finish

2.2.2 Latest Start dan Latest Finish

Waktu terlama mulai dan waktu terlama selesai untuk aktivitas yang berada pada akhir dari proyek adalah LF = waktu minimum pengerjaan proyek. Jadi, LF (E), (H), (J) = 194. Sedangkan untuk LS dari masing-masing aktivitas tersebut adalah LF dikurangi dengan durasi waktu masing-masing proyek. $LS(E) = LF(E) - 21 = 194 - 21 = 173$. $LS(H) = 194 - 28 = 166$. Dan $LS(J) = 194 - 45 = 149$.

Aktivitas yang hanya terhubung dengan satu aktivitas saja, maka LF untuk aktivitas terbut adalah LS dari aktivitas yang terhubung aktivitas tersebut. Aktivitas I yang terhubung dengan aktivitas J, $LF(I) = LS(J) = 149$. Sedangkan untuk $LS(I) = 149 - 30$.

Aktivitas yang terhubung dengan lebih dari satu aktivitas, maka LF aktivitas tersebut adalah nilai minimal dari LS dari aktivitas-aktivitas tersebut. Aktivitas D terhubung dengan aktivitas E, H, dan J, maka $LF(D) = \text{MIN}(LS(E), LS(H), LS(J)) = \text{MIN}(173, 166, 149) = 149$. Sedangkan untuk $LS(D) = 149 - 20 = 129$.

LF = nilai minimum dari LS aktivitas yang terhubung dengan aktivitas tersebut
 LS = LF – durasi waktu aktivitas tersebut

Dengan menggunakan cara yang sama untuk aktivitas-aktivitas yang selanjutnya, maka akan didapatkan :



Gambar 2.1 3 Latest Start dan Latest Finish

Kemudian aktivitas-aktivitas yang memiliki nilai $ST = 0$, maka aktivitas tersebut berada pada jalur kritis atau *critical path*. Karena nilai ST dari aktivitas-aktivitas ini = 0, maka jika terjadi penundaan pada aktivitas tersebut maka akan mengakibatkan terlambatnya waktu penyelesaian proyek.

Jalur kritis (*critical path*) adalah jalur aktivitas-aktivitas yang memiliki $ST = 0$

2.3 Algoritma Genetika

Algoritma Genetika pertama kali diperkenalkan oleh John Holland dari Universitas Michigan (1975), Setiap masalah yang berbentuk adaptasi dapat diformulasikan dalam biologi genetika (Holland, 1975). (Beasley, 1993) Algoritma Genetika adalah metode yang berbentuk adaptasi, sehingga dapat digunakan untuk memecahkan permasalahan pencarian dan optimasi. Ide dasar dari algoritma genetika adalah proses evolusi yang diperkenalkan oleh Charles Darwin dalam karyanya *The Origin of Species*. Algoritma genetika menyerupai proses atau tahap-tahap pada teori evolusi, yaitu adanya proses seleksi, kawin silang, dan mutasi.

Sebuah solusi yang dibangkitkan dalam algoritma genetika disebut sebagai **kromosom**, sedangkan kumpulan kromosom-kromosom tersebut disebut sebagai **populasi**. Sebuah kromosom dibentuk dari komponen-komponen penyusun yang disebut sebagai **gen** dan nilainya dapat berupa bilangan bulat, biner, atau real ataupun karakter tergantung dari permasalahan yang ingin diselesaikan. Kromosom-kromosom tersebut akan berevolusi secara berkelanjutan yang disebut dengan **generasi**. Dalam tiap generasi kromosom-kromosom tersebut dievaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang ingin diselesaikan (**fungsi_objektif**) menggunakan ukuran yang disebut dengan **fitness**.

Untuk memilih kromosom yang tetap dipertahankan untuk generasi selanjutnya dilakukan proses yang disebut dengan *seleksi*. Proses seleksi kromosom menggunakan konsep aturan evolusi Darwin yang telah disebutkan sebelumnya yaitu kromosom yang mempunyai nilai fitness tinggi akan memiliki peluang lebih besar untuk terpilih lagi pada generasi selanjutnya.

Kromosom-kromosom baru yang disebut dengan *offspring*, dibentuk dengan cara melakukan perkawinan antar kromosom dalam satu generasi yang disebut sebagai proses *crossover*. Mekanisme perubahan susunan *11able11* penyusun makhluk hidup akibat adanya *11able11* alam yang disebut dengan *mutasi* direpresentasikan sebagai proses berubahnya satu atau lebih nilai gen dalam kromosom dengan suatu nilai acak.

Langkah-langkah dari Algoritma Genetika :

Inisialisasi :

1. Set $t = 0$.
2. Inisialisasi Populasi P_t

Algoritma :

3. Evaluasi Populasi P_t
4. WHILE *stopping_criterion* belum ditemukan DO
 - a. {Pilih beberapa elemen dari P_t untuk dipindahkan ke $P_t + 1$
 - b. Jalankan proses perkawinan silang pada P_t kemudian pindahkan ke $P_t + 1$, ukuran individu disesuaikan dengan probabilitas kawin silang
 - c. Jalankan proses mutasi pada P_t kemudian pindahkan ke $P_t + 1$, ukuran individu disesuaikan dengan probabilitas mutasi
 - d. Evaluasi populasi baru, $P_t + 1$

2.3.1 Kromosom

Dalam Algoritma Genetika, kromosom merupakan bagian penting dari algoritma. Satu kromosom atau individu mewakili satu solusi. Dalam permasalahan optimasi fungsi, sering kali nilai kontinyus akan diwakili dengan bilangan biner. Dengan bilangan biner langkah-langkah kawin silang atau mutasi akan lebih banyak variasinya.

Pada akhir algoritma genetika, jika dilakukan pengkodean terhadap bilangan-bilangan kontinyus, maka akan dilakukan proses pengembalian nilai-nilai biner ke nilai kontinyus. Dalam algoritma genetika akan dibangkitkan populasi sebagai kumpulan dari kromosom, dimana masing-masing kromosom mewakili satu solusi. Setiap anggota kromosom disusun gen-gen, dimana masing-masing gen mewakili elemen solusi. Dengan dibangkitkannya populasi, maka akan tersedia banyak pilihan solusi.

2.3.2 *Fitness*

Fungsi *fitness* digunakan untuk mengukur tingkat kebaikan atau kesesuaian suatu solusi dengan solusi yang dicari. Fungsi *fitness* berhubungan langsung dengan fungsi tujuan, atau juga sedikit modifikasi terhadap fungsi tujuan. Sejumlah solusi yang dibangkitkan dalam populasi akan dievaluasi menggunakan fungsi *fitness*.

Setelah setiap solusi dievaluasi dengan fungsi *fitness*, perlu dilakukan proses seleksi terhadap kromosom. Proses seleksi dilakukan untuk memilih diantara kromosom anggota populasi ini, mana yang menjadi induk atau melakukan identifikasi diantara populasi ini, kromosom yang akan menjadi anggota populasi berikutnya. Ada beberapa cara melakukan seleksi ini. Sebagian anggota populasi dipilih untuk proses reproduksi.

2.3.3 *Elitisme*

Konsep elitism dalam algoritma genetika berarti usaha mempertahankan individu-individu terbaik yang telah diperoleh di suatu

generasi ke dalam generasi selanjutnya. Sehingga individu-individu terbaik ini akan tetap muncul di populasi berikutnya. Langkah ini dilakukan dalam berbagai cara. Misalnya, melalui penyalinan individu terbaik, atau dapat juga melalui kombinasi antara solusi-solusi turunan atau anak dengan induk. Penggunaan operator crossover ini telah terbukti memiliki pengaruh yang sangat penting saat menggunakan algoritma genetika untuk menyelesaikan persoalan optimasi dengan tujuan tunggal (Santosa, 2011).

2.3.4 *Crossover* atau Kawin Silang

Istilah *crossover* juga sering disebut kawin silang. Ada bermacam-macam teknik kawin silang yang digunakan dalam algoritma genetika. Contohnya yaitu kawin silang sederhana dan kawin silang aritmatik. Dalam kawin silang sederhana, jika ada dua induk P1 dan P2 dan menghasilkan dua keturunan C1 dan C2. Misalkan anggota induk adalah $X = [x_1, x_2, \dots, x_n]$ dan $Y = [y_1, y_2, \dots, y_n]$ dan r adalah bilangan acak diskrit yang bernilai antara 1 dan panjang x , maka U dan V yang mewakili keturunan C1 dan C2 didefinisikan sebagai :

$$\mu_i = \begin{cases} x_i & i < r \\ y_i & \text{sebaliknya} \end{cases}$$

$$v_i = \begin{cases} x_i & i < r \\ y_i & \text{sebaliknya} \end{cases}$$

Nilai bilangan acak r sering disebut dengan titik potong. Jadi mulai gen ke-berapa kawin silang dilakukan. Misalkan dari 8 gen didapat $r = 3$, maka mulai gen ke-4, gen dari kromosom X akan ditukarkan dengan gen dari kromosom Y.

Sedangkan dalam kawin silang aritmatik, keturunan dihasilkan dengan cara melakukan kombinasi linier dari induk. Secara matematis dapat dituliskan

$$\begin{aligned} C_1 &= \lambda_1 X + \lambda_2 Y \\ C_2 &= \lambda_2 X + \lambda_1 Y \end{aligned} \tag{2.1}$$

Dengan syarat $\lambda_1 + \lambda_2 = 1$. Kawin silang aritmatik lebih cocok dengan kasus yang memiliki keputusan bernilai kontinyus.

Kawin silang dilakukan untuk mendapatkan kombinasi yang lebih baik antara satu individu dengan individu yang lain dalam satu populasi.

Parameter penting dalam kawin silang adalah probabilitas kawin silang. Jika parameter ini bernilai kecil, maka hanya akan sedikit kromosom yang akan mengalami kawin silang. Jika nilai parameter kawin silangnya besar, maka akan semakin banyak kromosom yang akan mengalami kawin silang. Misalkan parameter kawin silang adalah 0.5, maka sekitar separuh dari populasi yang akan mengalami kawin silang.

Fungsi *Crossover*

1. Pilih dua kromosom *chrom1* dan *chrom2*
2. Biarkan RN bilangan real acak antara 0 dan 1
3. Jika probabilitas $R < 0.5$ /* operator *crossover*

Crossover_Map(chrom1, chrom2)

else

Crossover_Order(chrom1, chrom2)

2.3.5 Mutasi

Mutasi memungkinkan munculnya individu-individu baru yang bukan berasal dari hasil kawin silang. Mutasi mengacu pada perubahan urutan atau penggantian elemen dari 15able15 solusi, pemunculan nilai baru (optimasi fungsi). Elemen tersebut juga dipilih secara acak. Misalkan untuk 15able15 solusi X, untuk elemen terpilih k

$$C = \begin{cases} xi & i \neq k \\ xk & i = k \end{cases}$$

Parameter penting dalam mutasi adalah probabilitas mutasi. Probabilitas ini akan menentukan kromosom mana yang akan mengalami perubahan gen. Semakin besar nilai probabilitas mutasi, semakin banyak kromosom dalam populasi yang akan mengalami mutasi. Misalkan nilai probabilitas mutasi adalah 0.01, maka 15able1515e sekitar 1% dari seluruh kromosom dalam populasi yang akan mengalami mutasi.

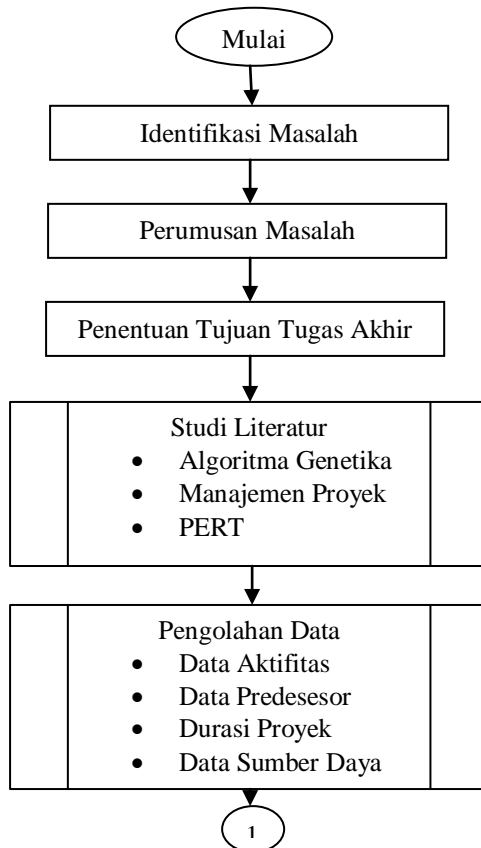
Mutasi dimaksudkan untuk memunculkan individu baru yang berbeda sama sekali dengan individu yang sudah ada. Dalam konteks optimasi memungkinkan munculnya solusi baru untuk bisa keluar dari *local optimum*.

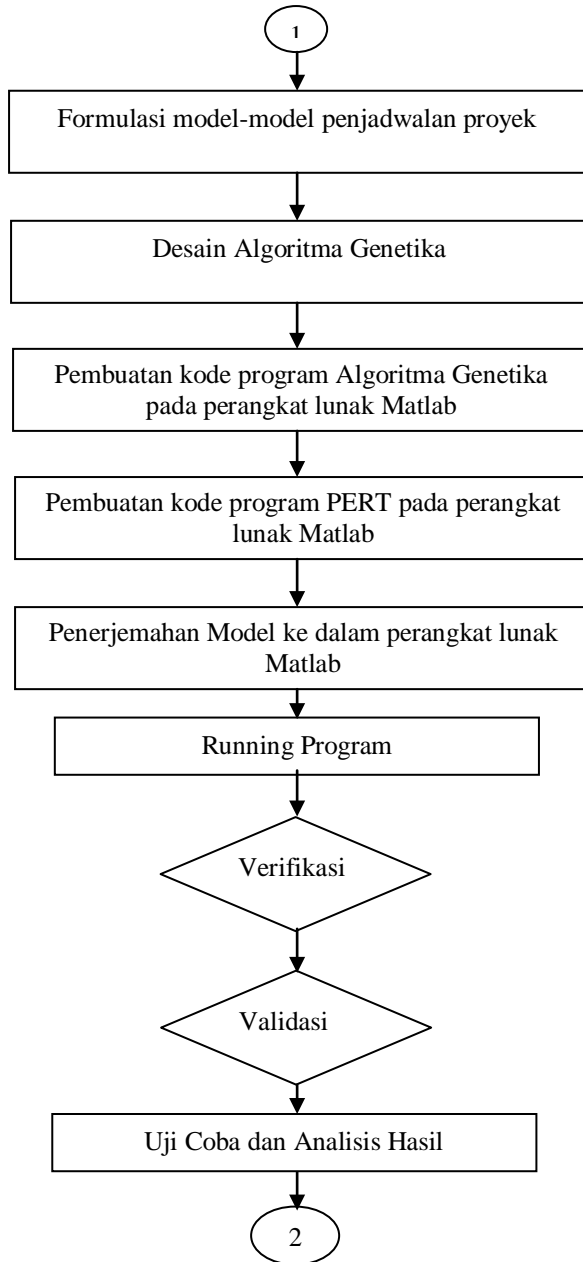
Halaman ini sengaja dikosongkan

BAB III

METODOLOGI Pengerjaan Tugas Akhir

Pada bagian ini diuraikan mengenai metodologi yang digunakan dalam pengerjaan tugas akhir. Metodologi ini digunakan sebagai panduan agar pengerjaan tugas akhir berjalan terarah dan sistematis. Tahapan pengerjaan dalam tugas akhir secara diagramatik diilustrasikan dalam gambar dan dijelaskan lebih rinci dalam uraian di bawah ini.





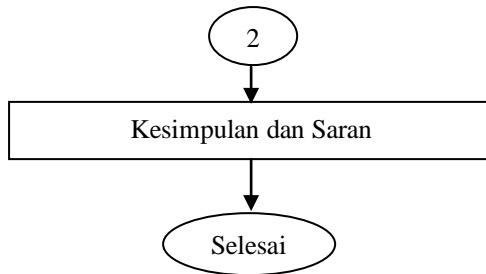


Diagram di atas dapat dijelaskan sebagai berikut :

5.5. Identifikasi Permasalahan

Pada tahapan ini dilakukan pengkajian terhadap masalah pengoptimalan durasi pada penjadwalan proyek. Tahapan identifikasi masalah meliputi pembuatan rumusan masalah dan penentuan tujuan tugas akhir.

5.5. Studi Literatur

Pada tahap ini akan dilakukan penggalian teori-teori maupun informasi lain yang menunjang pengerjaan tugas akhir sesuai dengan permasalahan yang telah dirumuskan. Penggalian teori maupun informasi dilakukan melalui banyak sumber seperti internet, artikel ilmiah, buku, dan jurnal. Hal ini juga bertujuan untuk mendapatkan kerangka kerja dan berpikir dalam menyelesaikan permasalahan yang telah dirumuskan sebelumnya. Tinjauan pustaka dilakukan sesuai dengan fokus tugas akhir. Mempelajari berbagai literatur yang berhubungan dengan penjadwalan proyek, Algoritma Genetika, dan penjadwalan proyek menggunakan metode PERT. Dengan melakukan studi pustaka ini diharapkan bisa teridentifikasi konsep dasar yang harus dipahami sebelum melanjutkan ke metode berikutnya.

Literatur pendukung dalam pengerjaan tugas akhir ini diantaranya mengenai review terhadap penelitian serupa yang telah dilakukan sebelumnya, atau penelitian yang bersangkutan dengan masalah penjadwalan proyek.

5.5. Pengolahan Data

Data yang dipakai dalam pengerjaan tugas akhir ini berupa kumpulan data proyek yang disediakan oleh salah satu situs web yaitu *Project Scheduling Problem Library* (PSPLIB), www.om-db.wi.tum.de/psplib/data.html. Pengolahan data yang dimaksud disini merupakan penginputan data ke dalam perangkat lunak Microsoft Office Excel dengan menggabungkan dan mengurutkan aktivitas proyek, *predesor*, durasi, dan kebutuhan sumber daya aktivitas.

5.5. Desain dan Implementasi

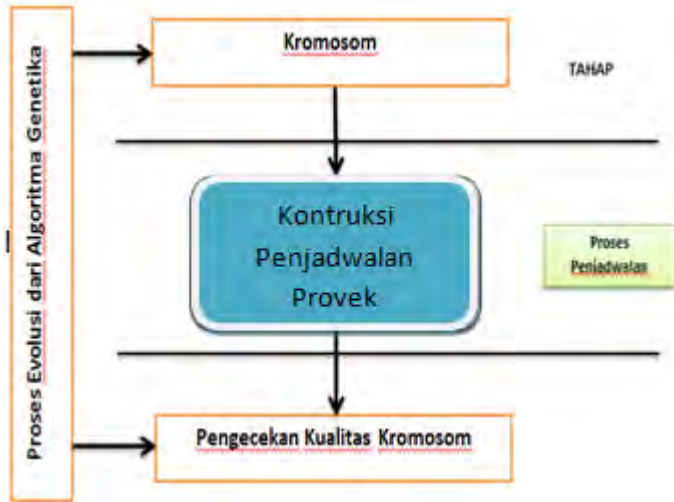
Pada tahap ini akan digambarkan tahap-tahap yang akan dilakukan untuk menyelesaikan permasalahan penjadwalan proyek dengan menggunakan metode Algoritma Genetika. Sebelum membuat langkah-langkah pengoptimalan penjadwalan proyek dengan menggunakan Algoritma Genetika, ada beberapa tahapan yang perlu dilakukan, tahapan-tahapan tersebut adalah sebagai berikut :

3.4.1 Formulasi Model Penjadwalan Proyek

Formulasi model penjadwalan proyek dilakukan dengan menentukan fungsi tujuan dan beberapa batasan. Fungsi tujuan pada tugas akhir ini yaitu meminimalkan waktu pengerjaan proyek. Sedangkan untuk batasan model terdapat 3 batasan yang masing-masing memiliki tujuan tertentu.

5.5.4 Desain Algoritma Genetika

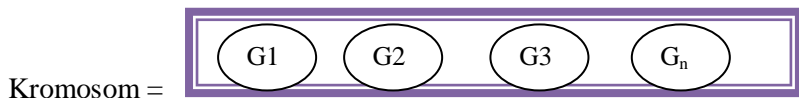
Pada tahapan ini akan digambarkan bagaimana penerapan Algoritma Genetika untuk mengoptimalkan durasi penjadwalan proyek. **Gambar 3.4.2.1** mengilustrasikan proses yang akan dijalani saat penerapan algoritma genetika, kemudian dijelaskan lebih rinci pada uraian berikutnya.



Gambar 3.4.2 1 Proses Algoritma Genetika

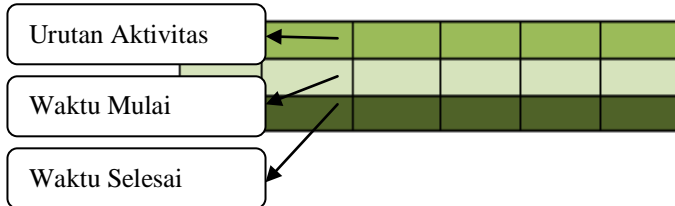
Representasi Kromosom

Untuk merepresentasikan permasalahan ke dalam betuk kromosom, maka tugas akhir ini membentuk sebuah kromosom dengan jumlah baris adalah tiga, dan jumlah kolomnya adalah n . Dimana n , adalah jumlah dari aktivitas-aktivitas dari proyek. Baris pertama adalah urutan dari aktivitas proyek, baris kedua adalah waktu mulai, dan baris ketiga adalah waktu selesai dari aktivitas tersebut. **Gambar 3.4.2.2** merupakan representasi dari kromosom.



n adalah jumlah dari aktivitas yang ada, dan menjadi parameter waktu selesai dari proyek.

Kemudian kromosom dari algoritma ini akan dibentuk seperti pada **Gambar 3.4.2.2** di bawah ini.



Gambar 3.4.2 2 Kromosom

Kontruksi Penjadwalan Proyek

Kontruksi penjadwalan proyek diperlukan untuk membuat batasan-batasan pada pengoptimalan yang dilakukan oleh algoritma Genetika. Kontruksi penjadwalan yang perlu diperhatikan yaitu aturan dalam penghitungan durasi proyek. Ketika melakukan penjadwalan program mengecek jumlah predesesor dari aktivitas-aktivitas yang ada. Jumlah predesesor aktivitas menentukan lama penyelesaian aktivitas yang dijadwalkan.

Algoritma Genetika akan mengecek urutan aktivitas yang harus dikerjakan. Namun, tidak menutup kemungkinan algoritma akan melakukan proses kawin silang maupun mutasi terhadap urutan aktifitas dengan memperhatikan fungsi tujuan dari proyek.

Langkah 1. Inisialisasi

- Inisialisasi untuk jumlah aktivitas

Langkah 2. Dilakukan perulangan untuk sepanjang predesesor aktivitas.

- 2.1 Pengecekan terhadap jumlah predesesor aktivitas
- 2.2 Jika aktivitas j memiliki 1 predesesor, maka ES aktivitas j adalah EF dari aktivitas sebelumnya.
- 2.3 Jika aktivitas j memiliki predesesor lebih dari 1, maka ES dari aktivitas j adalah $\max(Ef_{ij})$.
- 2.4 Jika aktivitas j memiliki tidak memiliki predesesor, maka ES aktivitas j adalah 0, dan aktivitas ini merupakan aktivitas pertama.
- 2.5 EF dari aktivitas adalah $ES + \text{durasi aktivitas}$.

Langkah 3. Menghitung Completion time/ CT dari seluruh aktivitas.

- $f_n = \max (EF)$

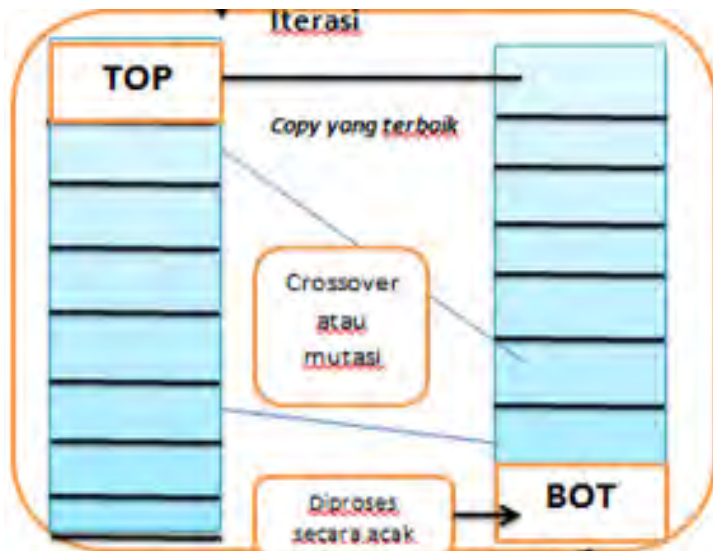
*Keterangan : ES = Earliest Start
EF = Earliest Finish*

Proses Evolusi Generasi Selanjutnya

Untuk mendapatkan generasi atau populasi selanjutnya, langkah pertama yang dilakukan adalah dengan memindahkan individu yang berada pada urutan pertama pada populasi pertama. Urutan populasi pertama terdiri dari yang terbaik hingga yang terburuk, dari urutan atas ke bawah. Jadi urutan individu yang berada pada tingkat teratas akan dipindahkan pada populasi selanjutnya sebanyak yang ditentukan. Jika kemungkinan kawin silang adalah 0,7, maka yang akan dipindahkan adalah sekitar 10% hingga 15%.

Proses kawin silang atau *crossover* akan menggunakan metode kawin silang sederhana. Misalkan anggota induk adalah $pa = [x_1, x_2, \dots, x_n]$ dan $ma = [y_1, y_2, \dots, y_n]$ dan r adalah bilangan acak diskrit yang bernilai antara 1 dan panjang 24able24 n (panjang aktivitas).

Kemudian untuk individu-individu yang tersisa kan diproses secara acak, kemudian dimasukkan ke dalam populasi selanjutnya. Jika terjadi konvergensi dini di dalam proses evolusi, maka akan dilakukan proses mutasi. Konvergensi dini ini mengakibatkan pencarian terhadap solusi atau generasi yang lebih baik melemah. Mutasi bekerja secara spontan dengan cara mengubah kromosom tanpa melalui persilangan dengan kromosom lain. Proses mutasi memiliki peranan penting saat populasi terjebak pada situasi konvergensi dini.



Gambar 3.4.2 3 Proses Evolusi

Pengecekan Kualitas Kromosom

Untuk mengecek seberapa tinggi tingkat kromosom dalam beradaptasi dengan perubahan maka dilakukan pengecekan terhadap kualitas kromosom yang telah mengalami perulangan. Fungsi Fitness (*Fitness Function*) adalah fungsi yang bekerja untuk menjaga kualitas kromosom agar tetap berada pada fungsi tujuan permasalahan. Fungsi *Fitness* dari algoritma genetika ini didefinisikan sebagai *makespan-dependent*. Karena fungsi objektif dari penyelesaian permasalahan ini adalah untuk meminimalkan *makespan* dari proyek maka fungsi *fitness* dapat dihitung sebagai berikut :

$$\text{Fitness} = f(x) \quad (3.1)$$

5.5.4 Pembuatan Kode Program Algoritma Genetika pada Perangkat Lunak Matlab

Algoritma Genetika digunakan untuk mengoptimalkan durasi proyek. Variabel-variabel yang dapat diketahui dari metode ini adalah :

- Pop_terbaik : populasi terbaik
- Cost : fungsi objektif dari permasalahan
- Fitness : fungsi evaluasi tiap-tiap individu di setiap generasi
- ES : *Earliest Start*
- EF : *Earliest Finish*

Agar nilai-nilai dari variabel diatas dapat diketahui, maka perlu dibuat kode program yang akan memproses secara otomatis

5.5.4 Pembuatan Kode Program PERT pada Perangkat Lunak Matlab

Selain Algoritma Genetika, PERT merupakan suatu metode yang biasa digunakan dalam manajemen proyek. PERT digunakan untuk mengetahui waktu mulai dan berakhir dari masing-masing aktivitas. Variabel-variabel yang dapat diketahui dari metode PERT ini adalah :

- ES : *Earliest Start*
- EF : *Earliest Finish*

Agar nilai-nilai dari variabel diatas dapat diketahui, maka perlu dibuat kode program yang akan memproses secara otomatis.

5.5.4 Penerjemahan Model ke Perangkat Lunak Matlab

Penerjemahan model ke dalam perangkat lunak Matlab bertujuan untuk dapat memproses model otomatis. Maksud dari penerjemahan model yaitu dengan mengubah model menjadi bahasa pemrograman sehingga dapat memproses data sesuai dengan batasan-batasan yang telah dibuat pada formulasi model.

5.5. Running Program

Tahap ini dilakukan dengan menjalankan program-program yang telah dibuat sebelumnya sehingga menghasilkan keluaran yang dapat dievaluasi nantinya. Lingkungan yang digunakan untuk melakukan *running program* optimasi adalah sistem operasi Windows 7. Perangkat lunak yang digunakan untuk melakukan komputasi adalah Matlab.

5.5. Data Uji Coba

Data yang digunakan pada tahap implementasi diambil dari PSPLIB, yang terdiri dari **Jumlah Aktivitas, Jumlah Predesor setiap Aktivitas, Durasi setiap Aktivitas, Jenis dan jumlah sumber daya yang dibutuhkan setiap Aktivitas.**

5.5. Verifikasi

Setelah diperoleh hasil, maka akan dilakukan verifikasi, yaitu memeriksa keseluruhan model dan program, apakah model dan program yang dibuat telah bebas dari *error* dan telah sesuai dengan yang diharapkan. Kemudian dari hasil yang dikeluarkan oleh program akan dilihat apakah semua aktivitas telah terjadwal, hal ini dapat dilihat dari setiap aktivitas telah memiliki waktu mulai dan waktu selesai.

Jika syarat-syarat tersebut tidak terpenuhi, maka akan dilakukan perhitungan kembali, dan juga jika masih ada yang belum sesuai atau masih terdapat *error* maka harus dilakukan peninjauan ulang.

5.5. Validasi

Selanjutnya akan dilakukan validasi untuk membentuk kesimpulan yang benar yaitu dengan memastikan apakah program yang telah dibuat sudah valid. Untuk mengetahui valid atau tidaknya program yang dibuat, maka dilakukan perbandingan antara hasil keluaran Algoritma Genetika dengan hasil keluaran PERT. Jika hasil dari kedua proses tersebut sama maka program yang telah dibuat dapat dikatakan valid. Selain daripada itu, data yang diambil dari PSPLIB memiliki *duedate* (tenggat waktu) yang harus terpenuhi. Jika kedua proses ini memenuhi batas waktu yang disediakan, maka program dapat dikatakan telah valid.

5.5. Uji Coba dan Analisis Hasil

Pada tahap ini dilakukan uji coba dan analisis terhadap hasil desain dan implementasi optimasi yang telah dilakukan pada tahap sebelumnya untuk melihat tingkat akurasi terhadap data yang sudah ada. Apabila dirasa kurang akurat dan kurang sesuai, maka akan dilakukan perubahan yang akan meningkatkan tingkat akurasi pengoptimalan penjadwalan proyek ini.

5.5.4 Skenario Uji Coba

Ada beberapa uji coba yang akan dilakukan pada tahap uji coba antara lain:

1. Menambah atau mengurangi jumlah individu pada populasi
2. Meningkatkan atau menurunkan probabilitas mutasi

Skenario ini dilakukan untuk melihat bagaimana hasil yang akan didapatkan dengan perubahan-perubahan yang dilakukan pada parameter-parameter tertentu, apakah hasilnya sama, lebih baik, atau lebih buruk. Skenario ini dilakukan bertujuan melihat tingkat sensitifitas parameter tertentu, dalam hal ini yaitu fungsi objektif permasalahan.

5.5. Kesimpulan dan Saran

Setelah melakukan uji coba dan analisi hasil, maka dapat ditarik kesimpulan tugas akhir untuk menjawab tujuan yang telah ditetapkan di awal. Dari hasil tersebut dapat dianalisa dan diberi saran-saran untuk lebih mengembangkan model maupun program dalam mengoptimalkan durasi penjadwalan proyek. Sehingga dapat lebih mengembangkan permasalahan penjadwalan proyek yang dapat diselesaikan menggunakan Algoritma Genetika.

5.5. Penyusunan Buku Tugas Akhir

Tahapan penyusunan buku merupakan bagian terakhir dari proses pengerjaan tugas akhir ini. Dalam tahapan ini, semua hasil yang telah diperoleh dari tahapan-tahapan sebelumnya didokumentasikan secara sistematis sesuai dengan standar yang berlaku di Jurusan Sistem Informasi ITS.

BAB IV

DESAIN DAN IMPLEMENTASI MODEL

Pada bab ini akan dijelaskan mengenai implementasi model penjadwalan proyek. Pada awalnya akan dijelaskan mengenai data yang dipakai pada tugas akhir ini. Data diambil dari *Project Scheduling Problem Library(PSPLIB)* yang meliputi data aktivitas, data durasi, data urutan pengerjaan proyek, data mode, dan data sumber daya yang dibutuhkan. Setelah semua data diolah sedemikian rupa maka langkah selanjutnya yaitu dengan mengimplementasikannya pada model. Langkah terakhir yaitu dengan melakukan optimasi penjadwalan dengan menggunakan bantuan *toolbox* algoritma genetika pada Matlab.

4.1 Data pengerjaan proyek

Data yang dipakai pada tugas akhir ini diambil dari *Project Scheduling Problem Library(PSPLIB)*. Kemudian data disusun ke dalam *Microsoft Excel*. Berikut ini merupakan data-data yang ada dalam pengerjaan proyek.

4.1.1 Aktivitas dan Urutan Pengerjaan Proyek

Seperti pada umumnya, sebuah proyek terdiri dari beberapa aktivitas dan urutan pengerjaan aktivitas. Setelah mengetahui aktivitas-aktivitas yang akan dijadwalkan, urutan aktivitas proyek merupakan komponen yang harus terdaftar. Dengan mengetahui urutan dari aktivitas maka dapat dibuat jadwal pengerjaan proyek. Urutan dari aktivitas-aktivitas yang ada dapat diketahui dari *predesor* aktivitas. *Predesor* merupakan aktivitas yang terlebih dahulu dijalankan. Setiap aktivitas memiliki satu atau lebih *predesor* yang akan menandakan waktu mulai dari aktivitas tersebut. Daftar aktivitas proyek dan urutan pengerjaan proyek dapat dilihat pada 29able 4.1 berikut:

Tabel 4.1Daftar Aktivitas dan Urutan Aktivitas

Aktivitas	Urutan Aktivitas		
1	0	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	4	0	0
6	2	0	0
7	3	0	0
8	3	0	0
9	4	0	0
10	4	0	0
11	2	0	0
12	8	0	0
13	3	0	0
14	9	12	0
15	2	0	0
16	10	0	0
17	13	14	0
18	13	0	0
19	8	0	0
20	5	11	18
21	16	0	0
22	16	17	18
23	20	22	0
24	19	23	0

Aktivitas	Urutan Aktivitas		
25	10	20	0
26	11	0	0
27	8	0	0
28	21	27	0
29	19	0	0
30	6	24	25
31	26	28	0
32	29	30	31

4.1.2 Durasi dan Kebutuhan Sumber Daya

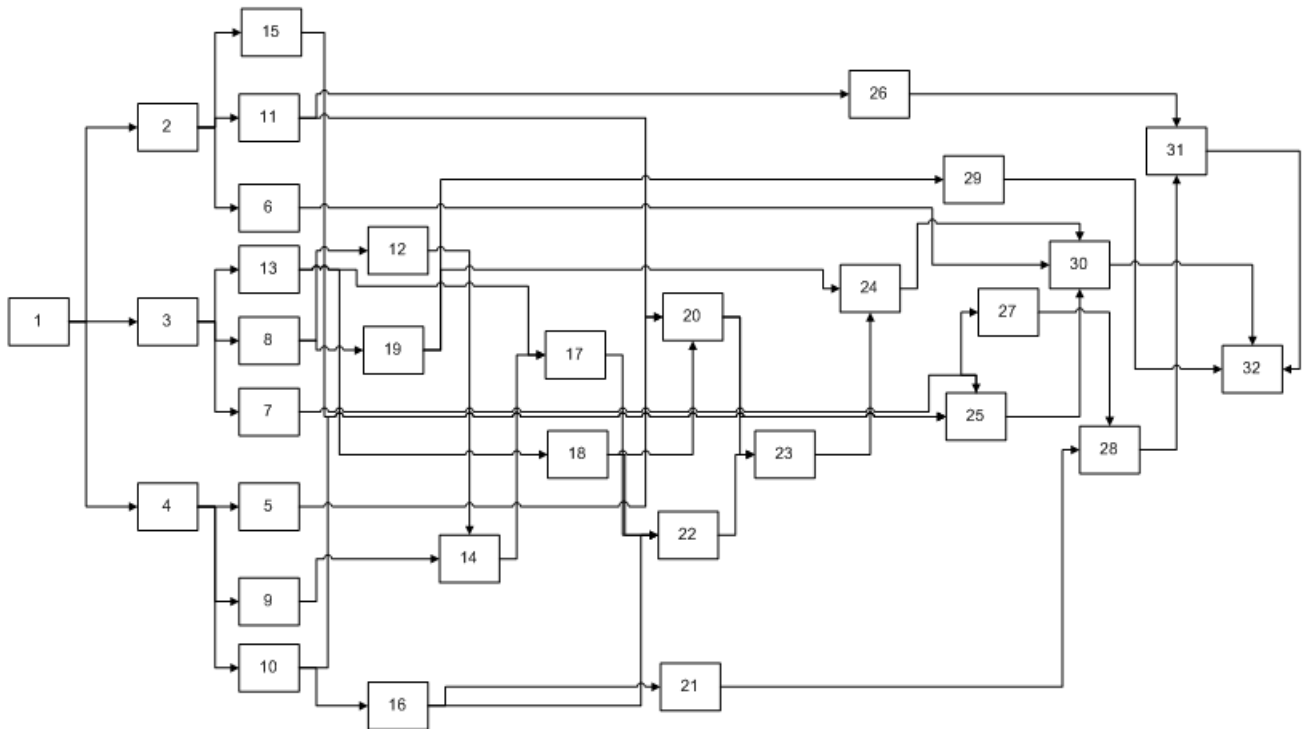
Komponen selanjutnya adalah durasi dan kebutuhan sumber daya. Durasi merupakan salah satu tujuan optimasi dalam pengerjaan proyek. Setiap aktivitas memiliki durasi yang berbeda. Selain durasi kebutuhan sumber daya juga komponen yang diperhitungkan dalam pengerjaan proyek. Ketersediaan sumber daya mempengaruhi durasi aktivitas yang kemudian berdampak pada durasi proyek. Setiap aktivitas membutuhkan sumber daya yang tidak sama. Daftar durasi aktivitas dan kebutuhan sumber daya setiap aktivitas ditunjukkan pada 31table 4.2 berikut:

Tabel 4.2 Daftar Durasi dan Kebutuhan Sumber Daya

Aktivitas	Durasi	Kebutuhan sumber daya1	Kebutuhan sumber daya2	Kebutuhan sumber daya3	Kebutuhan sumber daya1
1	0	0	0	0	0
2	8	4	0	0	0
3	4	10	0	0	0
4	6	0	0	0	3

Aktivitas	Durasi	Kebutuhan sumber daya1	Kebutuhan sumber daya2	Kebutuhan sumber daya3	Kebutuhan sumber daya1
5	3	3	0	0	0
6	8	0	0	0	8
7	5	4	0	0	0
8	9	0	1	0	0
9	2	6	0	0	0
10	7	0	0	0	1
11	9	0	5	0	0
12	2	0	7	0	0
13	6	4	0	0	0
14	3	0	8	0	0
15	9	3	0	0	0
16	1	0	0	0	5
17	6	0	0	0	8
18	5	0	0	0	7
19	3	0	1	0	0
20	7	0	0	0	0
21	2	0	0	0	6
22	7	2	0	0	0
23	2	3	0	0	0
24	3	0	9	0	0
25	3	4	0	0	0
26	7	0	0	4	0
27	8	0	0	0	7
28	3	0	8	0	0
29	7	0	7	0	0

Aktivitas	Durasi	Kebutuhan sumber daya1	Kebutuhan sumber daya2	Kebutuhan sumber daya3	Kebutuhan sumber daya1
30	2	0	7	0	0
31	2	0	0	2	0
32	0	0	0	0	0



Gambar 4.1 1 Urutan Pengerjaan Proyek

4.1.3 Kapasitas Sumber Daya

Kapasitas sumber daya menjadi batasan dalam permasalahan penjadwalan proyek. Kebutuhan aktivitas-aktivitas terhadap sumber daya tidak bernilai *infinity* atau tidak terhingga. Keterbatasan sumber daya ini menjadi batasan kedua setelah urutan aktivitas. Kebutuhan sumber daya setiap aktivitas tidak boleh melebihi kapasitas sumber daya. Daftar kapasitas sumber daya dapat dilihat pada 35able 4.3 berikut :

Tabel 4.3Daftar Kapasitas Sumber Daya

SD1	SD2	SD3	SD4
12	13	4	12

Keterangan : SD : Sumber daya

4.2 Formulasi Model

Formulasi model penjadwalan proyek dengan keterbatasan sumber daya terdiri dari fungsi tujuan dan beberapa batasan. Disini, penulis akan memaparkan penentuan fungsi tujuan dan beberapa batasan tersebut.

4.2.1 Penentuan Fungsi Tujuan

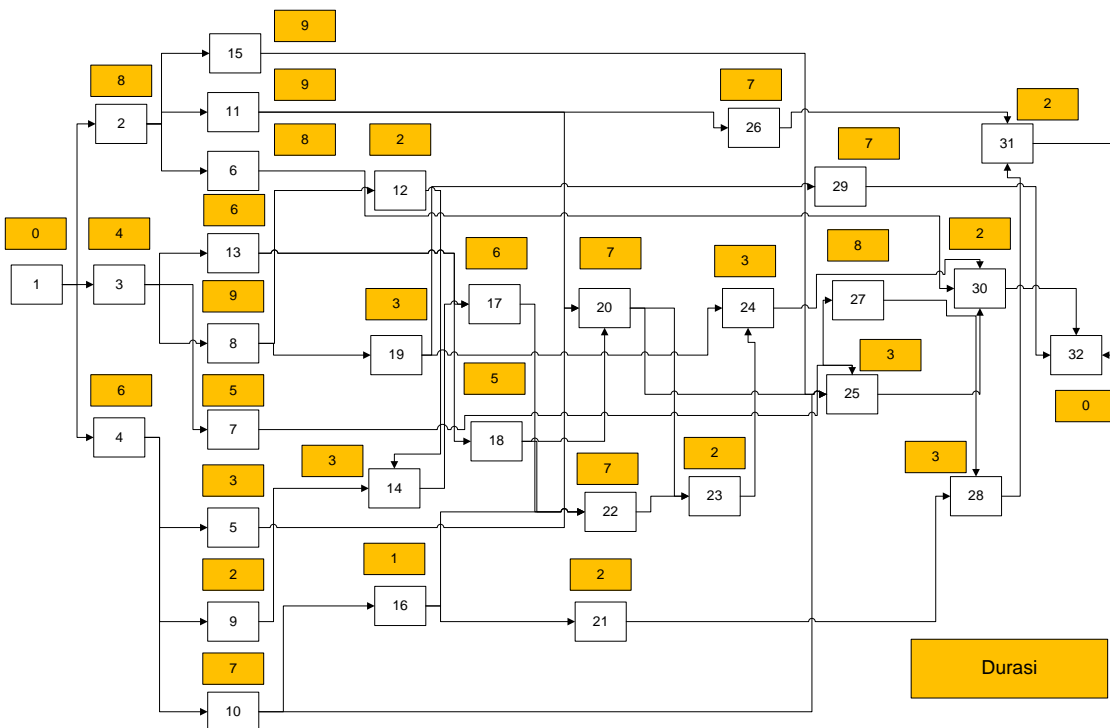
Seperti halnya metode penjadwalan lainnya, penjadwalan proyek dengan keterbatasan sumber daya atau resource-constrained project scheduling juga memiliki fungsi tujuan. Fungsi tujuan yang diangkat pada tugas akhir ini yaitu untuk meminimalkan durasi waktu keseluruhan penyelesaian pengerjaan proyek. Fungsi tujuan pada tugas akhir ini dapat dituliskan sebagai berikut :

$$\text{Meminimalkan } f_j = \max[EF_n] \quad (4.1)$$

Keterangan :

f_j = Waktu selesai keseluruhan proyek.

EF_n = *Earliest Finish* dari aktivitas terakhir dari proyek.



Gambar 4.1 2 Durasi setiap Aktivitas

4.2.2 Penentuan Batasan 1

Batasan pertama pada tugas akhir ini adalah untuk memastikan bahwa setiap aktivitas hanya dijalankan pada 1 mode. Batasan pertama pada tugas akhir ini dapat ditulis sebagai berikut :

$$\sum_{m=1}^M y_{jm} = 1 \quad \forall j \in \{1, \dots, J\} \quad (4.2)$$

Keterangan :

m = Mode aktivitas.

M_j = Jumlah mode yang dimiliki oleh aktivitas j.

Y_{jm} = Nilai yang diberikan pada mode m. Bernilai 1 jika mode tersebut dijalankan pada aktivitas j dan bernilai 0 jika mode tersebut tidak dijalankan.

4.2.3 Penentuan Batasan 2

Batasan kedua untuk tugas akhir ini dimaksudkan untuk memastikan bahwa waktu selesai suatu aktivitas lebih kecil dari atau sama dengan waktu mulai aktivitas berikutnya. Hal ini dimaksudkan agar semua aktivitas dapat berjalan dengan baik dan tidak ada tumpang tindih antar aktivitas. Batasan ini juga merupakan implementasi dari *predesesor* atau urutan aktivitas. Berikut ini merupakan formulasi model untuk batasan kedua :

$$f_i \leq s_j \quad \forall j \in \{1, \dots, J\}, \quad \forall i \in P_j \quad (4.3)$$

Keterangan :

s_j = Waktu mulai dari sebuah aktivitas.

f_i = Waktu selesai dari aktivitas sebelumnya.

P_j = *Predesesor* dari sebuah aktivitas.

4.2.4 Penentuan Batasan 3

Batasan ketiga pada tugas akhir ini yaitu untuk membatasi jumlah sumber daya pada setiap aktivitas agar kurang dari atau sama

dengan kapasitas sumber daya pada suatu periode. Hal ini untuk menghindari adanya kemungkinan kekurangan sumber daya selama proyek berjalan. Berikut ini merupakan formulasi model dari batasan ketiga :

$$\sum_{j=2}^{j-1} \sum_{m=1}^M (k_{jmt} \times y_{jm}) \leq K_{rt} \quad \forall r \in \{1 \dots, R\}, \forall t \in \{1 \dots, T\} \quad (4.4)$$

Keterangan :

y_{jm} = Nilai yang diberikan pada mode m. Bernilai 1 jika mode tersebut dijalankan pada aktivitas j dan bernilai 0 jika mode tersebut tidak dijalankan.

K_{jmt} = Kebutuhan sumber daya r pada aktivitas j jika aktivitas tersebut dijalankan pada mode m.

K_{rt} = Kapasitas sumber daya yang tersedia pada periode t.

4.3 Metodologi Penyelesaian Model dengan Perangkat Lunak Matlab

Penjadwalan Proyek dengan tujuan untuk meminimalkan durasi proyek menggunakan metode Algoritma Genetika dapat diselesaikan dengan bantuan perangkat lunak Matlab. Perangkat lunak Matlab yang digunakan dalam pengerjaan tugas akhir ini yaitu Matlab versi R2009a. Hal pertama yang akan dibuat sebelum membuat program adalah membuat diagram alur. Dengan membuat diagram alur, langkah-langkah pembuatan program telah terdefinisikan lebih dahulu sehingga diharapkan akan menjadi lebih mudah pada tahap pembuatan kode program.

4.3.1 Inisialisasi Data

Tahap awal pada metodologi penyelesaian model ialah inisialisasi data. Data yang telah diinput ke dalam program Microsoft Excel harus memasuki tahapan inisialisasi sebelum masuk ke proses selanjutnya. Berikut ini merupakan data-data yang akan diinisialisasi.

- **Algoritma Genetika**

1. Aktivitas

Aktivitas merupakan kegiatan yang akan dilakukan selama masa hidup proyek. Aktivitas yang terdapat pada tugas akhir ini adalah sejumlah 32 aktivitas.

2. Durasi

Durasi adalah lama waktu yang dibutuhkan untuk menjalankan sebuah aktivitas.

3. Predesesor

Predesesor adalah aktivitas-aktivitas yang harus terlebih dahulu dijalankan sebelum aktivitas setelahnya. Predesesor dibutuhkan untuk mengetahui kapan sebuah aktivitas akan dimulai. Dengan adanya predesesor maka diharapkan tidak 39able3939e aktivitas yang saling mendahului.

- **PERT**

1. Aktivitas

Aktivitas merupakan kegiatan yang akan dilakukan selama masa hidup proyek. Aktivitas yang terdapat pada tugas akhir ini adalah sejumlah 32 aktivitas.

2. Durasi

Durasi adalah lama waktu yang dibutuhkan untuk menjalankan sebuah aktivitas.

3. Predesesor

Predesesor adalah aktivitas-aktivitas yang harus terlebih dahulu dijalankan sebelum aktivitas setelahnya. Predesesor dibutuhkan untuk mengetahui kapan sebuah aktivitas akan dimulai. Dengan adanya predesesor maka diharapkan tidak 39able3939e aktivitas yang saling mendahului.

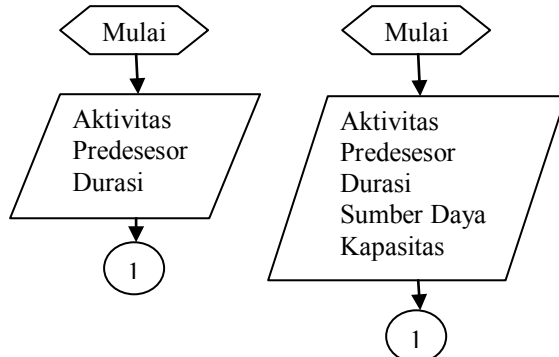
4. Sumber Daya

Sumber daya ialah orang, unit, maupun sejumlah yang dibutuhkan sebuah aktivitas untuk menyelesaikan

proyek. Sumber daya yang terdapat pada tugas akhir ini adalah sejumlah empat sumber daya.

5. Kapasitas Sumber Daya

Kapasitas sumber daya adalah batasaan atau jumlah keseluruhan sumber daya yang tersedia pada waktu pengerjaan proyek.

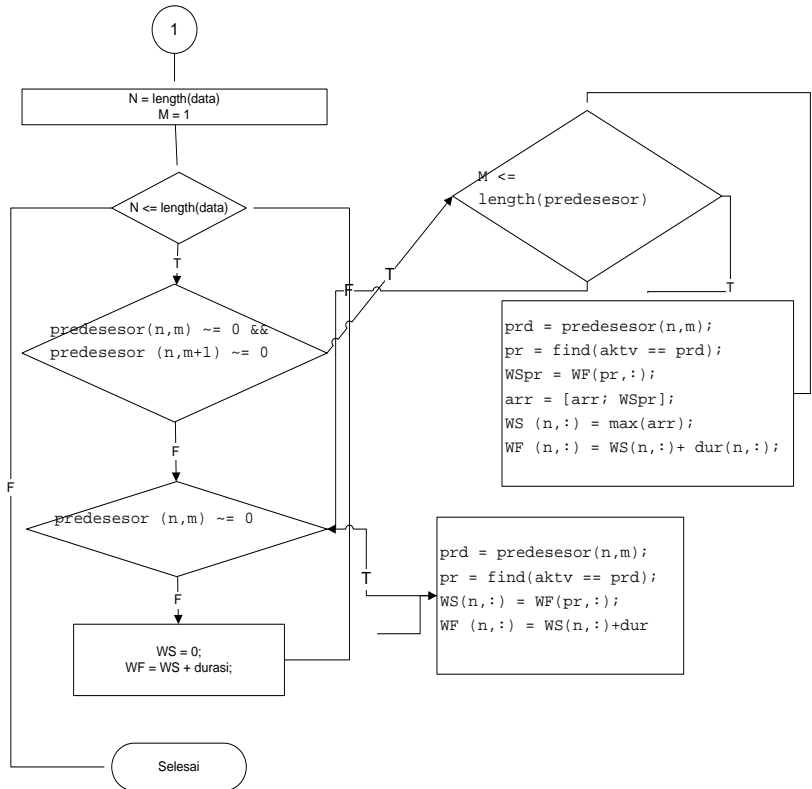


Gambar 4.3 1 Diagram Alur Inisialisasi Data

4.3.2 Mencari Nilai Durasi Proyek pada Algoritma Genetika

Metode algoritma genetika lebih sering menggunakan istilah *cost*(biaya) sebagai parameter fungsi objektif permasalahan. Durasi merupakan salah satu parameter yang menjadi variable utama yang akan dioptimalkan pada permasalahan penjadwalan proyek. Ada beberapa istilah yang dipakai dalam perhitungan durasi proyek, salah satunya ES yang lebih dikenal *Earliest Start* dan EF yang lebih dikenal *Earliest Finish*. ES adalah waktu mulai paling cepat dari sebuah aktivitas. EF adalah waktu selesai paling cepat dari sebuah aktivitas. Mencari nilai durasi yang dimaksud pada bagian ini adalah mencari nilai ES dan EF. Program pencarian ES dan EF akan dimasukkan ke dalam metode algoritma genetika sebagai panduan penghitungan durasi proyek. Seperti yang telah dijelaskan pada bab sebelumnya bahwa kromosom atau individu yang akan diproses pada algoritma genetika terdiri dari

aktivitas, waktu mulai, dan waktu selesai. Langkah-langkah pencarian nilai mulai dan nilai selesai digambarkan berikut ini :

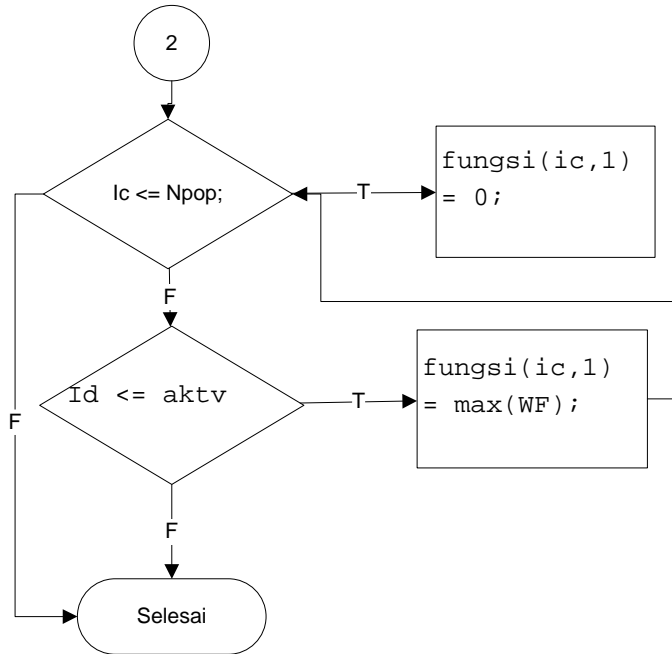


Gambar 4.3.2 Diagram Alur mencari waktu mulai dan waktu selesai

4.3.3 Menentukan *cost*(biaya) setiap Individu

Kromosom atau individu memiliki ukuran yang menjadi objek perhitungan program. Pada kasus penjadwalan proyek, ukuran setiap individu dihitung melalui waktu mulai dan waktu selesai. Pada langkah

sebelumnya telah dibuat diagram alur pencarian waktu mulai dan waktu selesai. Tahap selanjutnya adalah bagaimana menentukan ukuran per kromosom dari waktu selesai dan waktu mulai. Nilai maksimal dari waktu selesai keseluruhan aktivitas merupakan *cost*(biaya) per kromosom. Tahapan ini digambarkan diagram alur berikut :

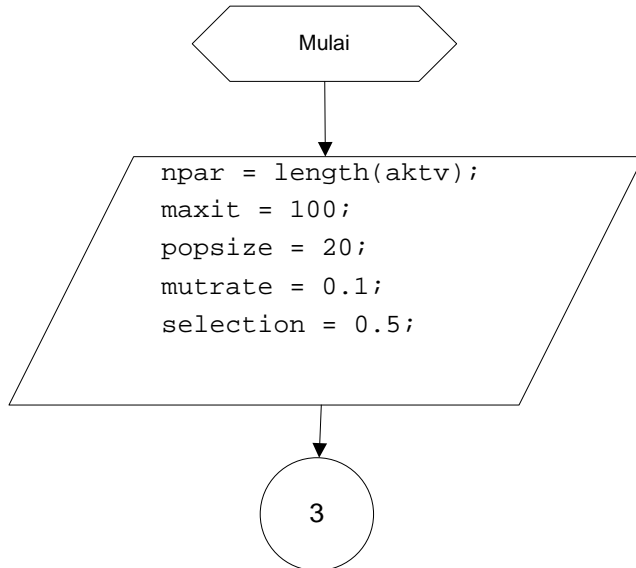


Gambar 4.3 Diagram alur *cost* kromosom

4.3.4 Mengatur Algoritma Genetika

Sebelum mengoptimalkan permasalahan, dibutuhkan penentuan parameter-parameter algoritma genetika yang akan digunakan pada saat proses pencarian solusi. Panjang kromosom, maksimal iterasi, nilai

mutasi, nilai seleksi, dan ukuran populasi. Diagram berikut ini menggambarkan inialisasi parameter-parameter algoritma genetika.

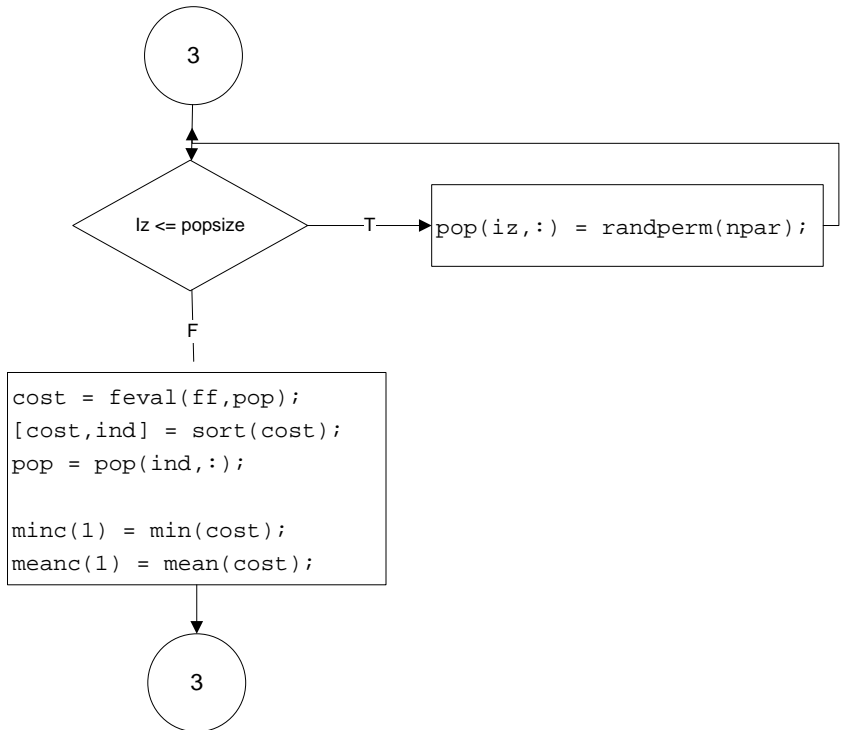


Gambar 4.3 4 Inialisasi parameter GA

4.3.5 Inialisasi Populasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada. Inialisasi populasi yang dimaksud pada tugas akhir ini yaitu menginialisasi panjang populasi sepanjang besar populasi, kemudian menginialisasi kromosom, dengan membangkitkan bilangan acak sepanjang aktivitas. Fungsi *Fitness* juga dideklarasikan pada tahap ini

yaitu $cost$ (biaya) proyek yang paling minimal. Diagram berikut menggambarkan alur dari inisialisasi populasi pada tugas akhir ini.

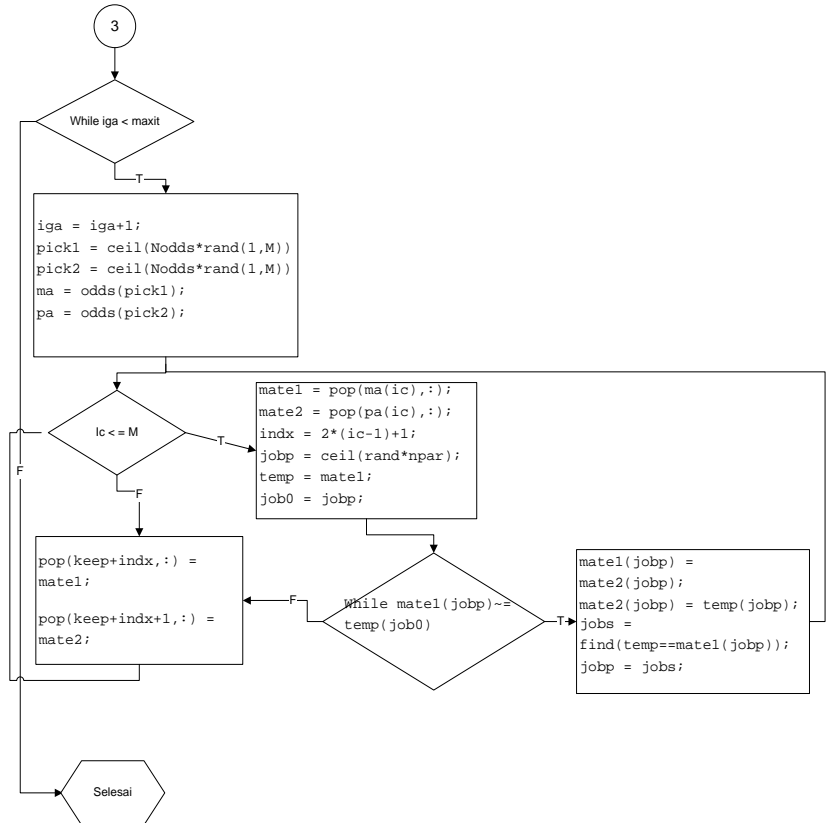


Gambar 4.3 5 Diagram Alur Inisialisasi Populasi

4.3.6 Proses Kawin Silang

Crossover adalah operator algoritma genetika yang membutuhkan parameter dua kromosom. Dua buah kromosom tersebut disebut kromosom induk. Operator ini akan menghasilkan dua buah kromosom baru. Perkawinan silang dibutuhkan untuk mendapatkan urutan aktivitas yang paling optimal. Oleh karena itu pada langkah

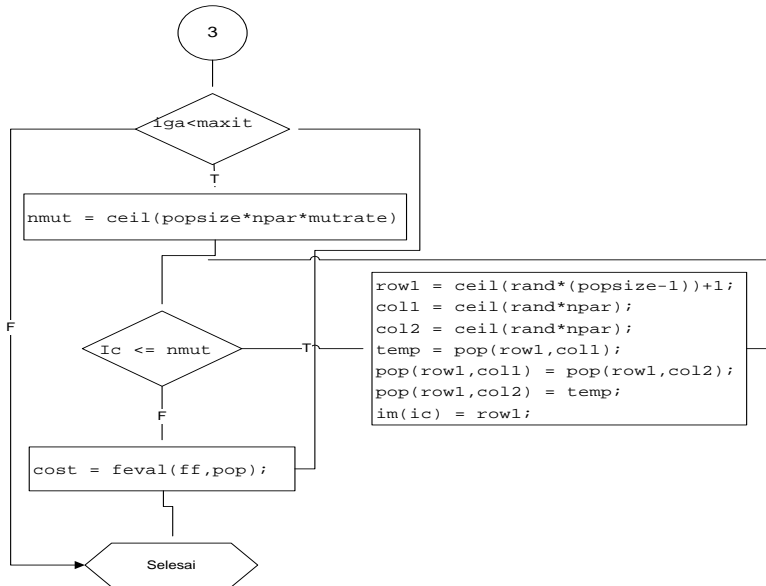
sebelumnya diinisialisasi kromosom dari bilangan acak sepanjang aktivitas yang ada dengan harus memperhatikan predesesor dan durasi dari aktivitas yang diproses. Proses perkawinan silang pada permasalahan penjadwalan proyek digambarkan diagram berikut :



Gambar 4.3 6 Diagram alur perkawinan silang

4.3.7 Proses Mutasi

Mutasi adalah operator yang membutuhkan satu parameter. Kromosom operator ini merupakan nilai suatu gen dari sebuah kromosom sehingga kromosom yang baru ini berbeda dengan kromosom yang lama. Baik perkawinan silang, begitu juga proses mutasi harus memperhatikan fungsi tujuan dari permasalahan yaitu meminimalkan durasi. Proses mutasi akan memindahkan beberapa sehingga urutan pengerjaan aktivitas akan berubah dengan tujuan mendapatkan individu terbaik. Dengan perubahan urutan aktivitas tersebut, akan menghasilkan durasi proyek yang paling optimal atau mencapai titik konvergen. Diagram dibawah ini menunjukkan langkah-langkah proses mutasi yang dilakukan.



Gambar 4.3 7 Diagram Alur proses Mutasi

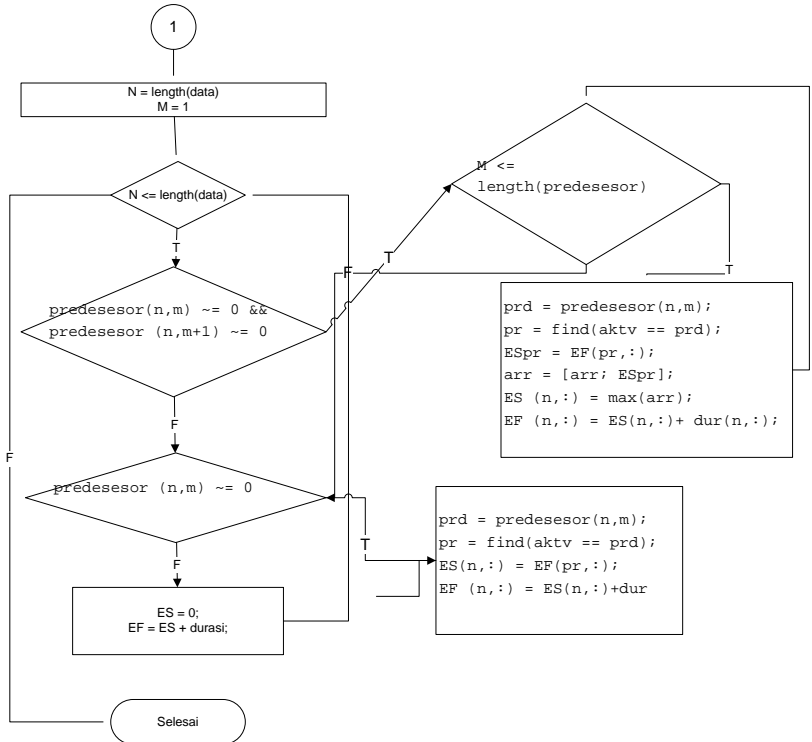
4.3.8 PERT

Valid tidaknya keluaran(*output*) dari program algoritma genetika yang dibuat, diukur dengan membandingkan hasil algoritma genetika dengan hasil yang dikeluarkan PERT. Jika hasil dari keduanya sama, maka dapat disimpulkan bahwa program yang dibuat telah valid. Oleh karena itu, dibutuhkan pembuatan program PERT untuk mengoptimalkan permasalahan yang dikerjakan dengan fungsi tujuan sama yaitu meminimalkan durasi proyek. Perhitungan nilai ES yang lebih dikenal dengan *Earliest Start* dan juga EF yang lebih dikenal dengan *Earliest Finish* menjadi objek optimisasi pada program PERT yang dibuat.

4.1.1.1 Mencari Nilai ES dan EF

Untuk menghitung waktu mulai tercepat(ES) dan waktu selesai tercepat dimulai(EF) dari aktifitas yang tidak memiliki penghubung yang menuju kepada aktifitas tersebut. Setiap aktifitas perlu diketahui kapan waktu mulai paling awal dan waktu selesai paling awal agar dapat diestimasi waktu penyelesaian aktifitas tersebut. Untuk mencari nilai ES diketahui dari aktifitas predesesornya. Nilai ES = nilai EF aktifitas predesesornya, sedangkan nilai EF didapat dari nilai ES dijumlahkan dengan durasi aktifitas tersebut. Untuk aktifitas yang pertama maka nilai ES = 0. Gambar berikut merupakan diagram alur proses penentuan nilai ES dan EF.

ES = Maksimum EF dari masing-masing aktifitas yang menuju kepada aktifitas tersebut
EF = ES + (durasi waktu penyelesaian aktifitas)



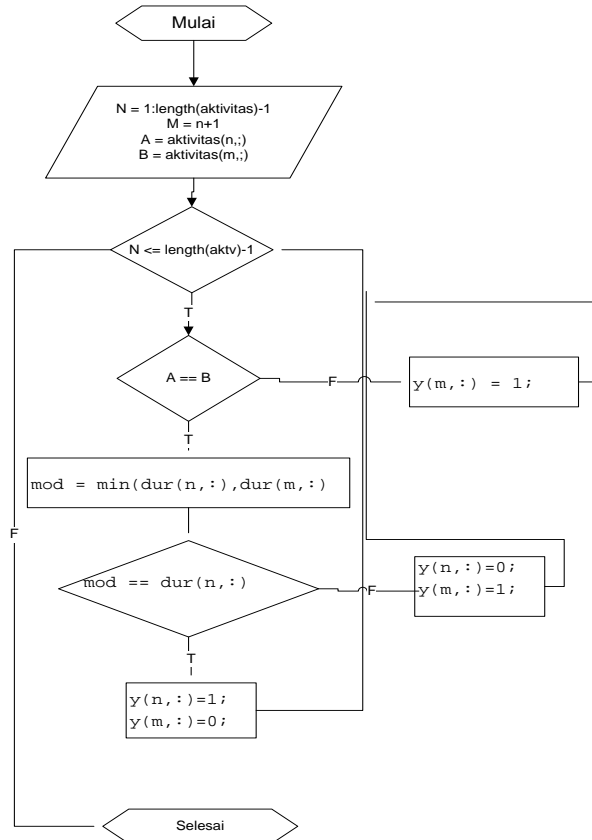
Gambar 4.3 8 Mencari nilai ES dan EF

4.3.9 Memasukkan Batasan Model

Pada tugas akhir ini terdapat tiga batasan seperti yang ada pada pembahasan sebelumnya, yang masing-masing memiliki fungsi berbeda. Sebelum membuat programnya, maka dibuat diagram alur terlebih dahulu agar memudahkan pada saat pembuatan program. Selanjutnya akan dijelaskan diagram alur dari masing-masing batasan yang digunakan pada tugas akhir ini.

4.3.9.1 Memasukkan Batasan 1

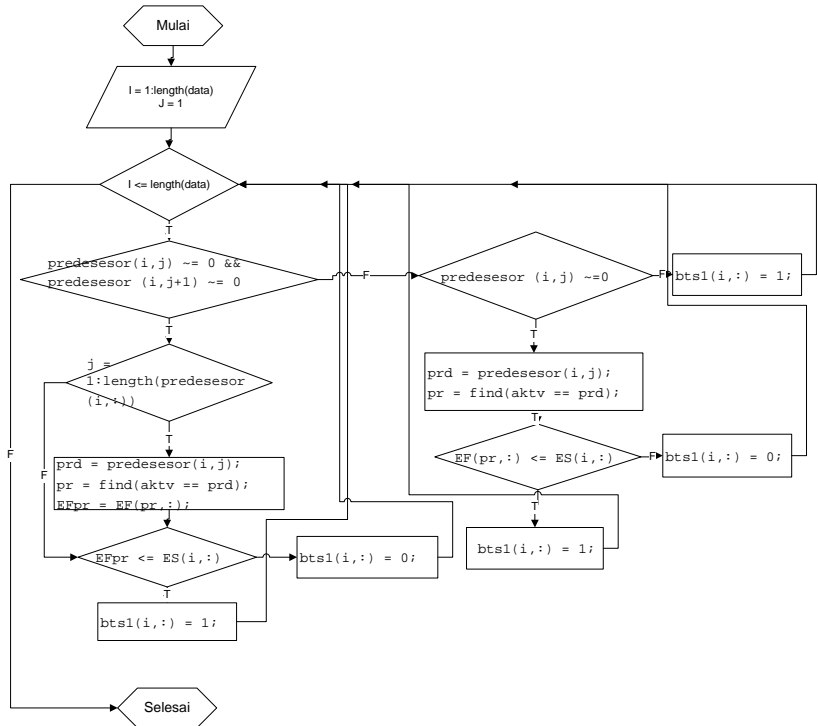
Seperti yang telah dijelaskan sebelumnya batasan yang pertama pada tugas akhir ini bertujuan untuk memastikan bahwa setiap aktivitas memiliki mode = 1. Berikut merupakan diagram alur dari batasan yang pertama.



Gambar 4.3.9 1 Diagram Alur batasan pertama

4.3.9.2 Memasukkan Batasan 2

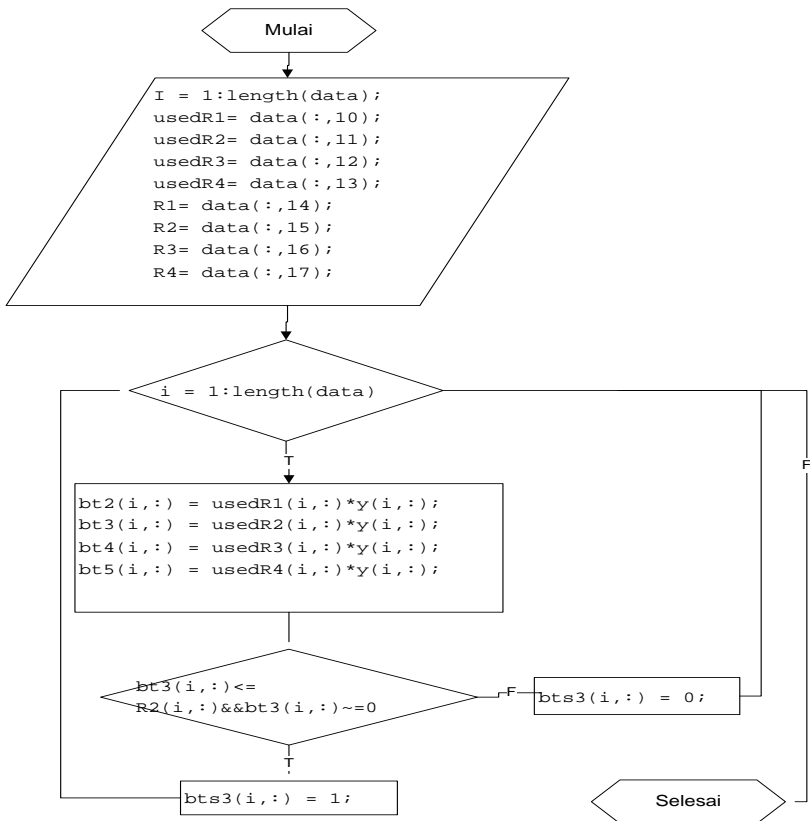
Batasan kedua untuk tugas akhir ini dimaksudkan untuk memastikan bahwa waktu selesai suatu aktivitas lebih kecil dari atau sama dengan waktu mulai aktivitas berikutnya atau aktivitas predesesor. Hal ini dimaksudkan agar semua aktivitas dapat berjalan dengan baik dan tidak ada tumpang tindih antar aktivitas. Gambar dibawah ini merupakan diagram alur dari batasan kedua.



Gambar 4.3.9 2 Diagram Alur batasan kedua

4.3.9.3 Memasukkan Batasan 3

Batasan ketiga pada tugas akhir ini yaitu untuk membatasi jumlah sumber daya pada setiap aktivitas kurang dari atau sama dengan kapasitas jumlah sumber daya yang tersedia pada waktu aktivitas dijalankan. Hal ini untuk menghindari adanya kemungkinan kekurangan sumber daya selama proyek berjalan. Diagram dibawah ini merupakan alur dari batasan ketiga.



Gambar 4.3.9 3 Diagram Alur batasan ketiga

5.5. Implementasi Penjadwalan Proyek dengan Algoritma Genetika ke dalam Perangkat Lunak Matlab

Dalam program ini akan ada dua file utama yaitu file *waktu.m* dan *PenjGA.m*. File *waktu.m* merupakan file yang berisi cara untuk mendapatkan waktu mulai, waktu selesai, dan *cost*(biaya) kromosom. *Cost*(biaya) kromosom merupakan fungsi tujuan daripada permasalahan yaitu durasi waktu selesai paling besar dari keseluruhan aktifitas. Kemudian fungsi *fitness* akan mendapatkan nilai durasi yang paling minimal dengan mencari disetiap generasi nilai *cost* yang paling kecil. File *PenjGA.m* merupakan file yang berisi langkah-langkah algoritma genetika.

4.4.1 Inisialisasi Data

Pada tahap pertama ini dilakukan inisialisasi data. Baris 2 merupakan cara untuk mengambil data dari excel dan menyimpannya dengan variabel data yang nantinya akan berupa tabel array. Baris 4,5, dan 6 adalah inisialisasi predesesor, aktivitas, dan durasi yang diambil dari file excel yang dibaca pada baris sebelumnya.

```

1. %% Pengambilan Data
2. data = xlsread('D:\\Khairil Juhdi
   Siregar\\All about
   TA\\dataPen\\data.xls','dataPenjadwala
   n','A1:Q34');
3. %melakukan inisialisasi data
4. predesesor = data (:,2:4);
5. aktv       = data (:,1);
6. dur       = data (:,9);

```

Gambar 4.4.1 Inisialisasi Data

4.4.2 Mencari Waktu Mulai dan Waktu Selesai

Tahap pertama dilakukan perulangan sebanyak Naktivitas di dalam populasi. Kemudian dilakukan pengecekan terhadap jumlah predesesor yang dimiliki setiap aktivitas. Ada tiga kondisi yang dipertimbangkan untuk menghitung waktu mulai dan waktu selesai dari suatu aktivitas. Kondisi pertama adalah aktivitas yang memiliki lebih dari 1 predesesor. Nilai waktu selesai predesesor akan disimpan pada array *arr*, kemudian akan dicari nilai yang paling maksimal diantara waktu selesai predesesor. Waktu selesai yang paling maksimal merupakan waktu mulai aktivitas. Kondisi kedua ialah aktivitas yang memiliki 1 predesesor atau hanya memiliki satu predesesor. Waktu mulai aktivitas merupakan waktu selesai predesesor. Dan kondisi ketiga adalah aktivitas yang tidak memiliki predesesor, aktivitas ini adalah aktivitas pertama. Waktu mulai dari aktivitas pertama adalah 0. Waktu selesai dari seluruh aktivitas pada setiap kondisi adalah nilai waktu mulai ditambah durasi aktivitas. Segmen berikut merupakan kode program pencarian waktu mulai dan waktu selesai.

```

1. [Npop,Naktv] = size(pop);
2. jadwal = [pop pop(:,1)];
%Menghitung waktu mulai dan waktu selesai
3. for ic = 1:Naktv % Inisiasi
4. for id = 1:Naktv
%Sebuah aktivitas mempunyai predesesor lebih dari 1
5. k = 1;
6. if predesesor(ic,k) ~= 0 &&
    predesesor(ic,k+1) ~= 0
        for k = 1:length(predesesor(ic,:))
            prd = predesesor(ic,k);
            pr = find(aktv == prd);
            WSpr = WF(pr,:);
            arr = [arr; WSpr];
        end
        WS(ic,:) = max(arr);
        WF(ic,:) = WS(ic,)+ dur(ic,:);
%Sebuah aktivitas mempunyai predesesor 1
7. elseif predesesor(ic,k) == 0
        prd = predesesor(ic,k);
        pr = find(aktv == prd);
        WS(ic,:) = WF(pr,:);
        WF(ic,:) = WS(ic,)+dur(ic,:);
%Sebuah aktivitas merupakan aktivitas awal dan
tidak memiliki %predesesor
8. else
        WS(ic,:) = 0;
        WF(ic,:) = WS(ic,)+ dur(ic,:);
9. end
10. end
11. end

```

Gambar 4.4.2 Waktu mulai dan Waktu selesai

4.4.3 Mencari *cost*(biaya) per Kromosom

Biaya di setiap kromosom merupakan variabel utama yang menjadi acuan untuk menentukan fungsi *fitness* dari algoritma genetika. Perulangan *ic* sepanjang *Npop* akan menyimpan variabel fungsi. Jika aktivitas sama dengan aktivitas *id* maka fungsi adalah $\max(WF)$, nilai maksimal dari waktu selesai dari keseluruhan aktivitas. Jadi setiap kromosom yang dihasilkan dalam populasi akan membawa variabel fungsi. Segmen berikut merupakan kode program mencari biaya(*cost*) kromosom.

```

%% ----- cost of chromosome -----
for ic = 1:Npop
    fungsi(ic,1) = 0;
    for id = 1:aktv
        fungsi(ic,1) = max(WF);
    end
end
end

```

Gambar 4.4.3 *cost* per kromosom

4.4.4 Inisialisasi Algoritma Genetika

Inisialisasi algoritma genetika dimulai dengan membuat sebuah fungsi yang dapat mengambil data-data penjadwalan pada file *waktu.m*. Fungsi yang dimaksudkan dalam hal ini adalah fungsi *ff*. Variabel *npar* adalah nilai sepanjang aktivitas yang ada. Variabel *job* dibuat untuk menghasilkan bilangan acak sepanjang aktivitas. Iterasi maksimal atau yang lebih dikenal dengan kriteria pemberhentian algoritma sepanjang 100 iterasi. Kode program berikut merupakan inisialisasi algoritma genetika.

```

%% setup the GA
ff = 'waktu';
npar = length(aktv);
Nt = npar;
rand('state',3)
job = rand(1,npar);
%% stopping criteria
maxit = 100;

```

Gambar 4.4.4 Inisialisasi Algoritma Genetika

4.4.5 Parameter Algoritma Genetika

Parameter-parameter algoritma genetika dibutuhkan sebagai variabel yang dipakai ketika menjalankan proses inisialisasi populasi dan proses genetika. Besar populasi yaitu sebesar 20, kadar untuk proses mutasi adalah sebesar 0.1, dan kadar generasi yang mengalami proses genetika sebesar 0.5. Generasi yang tetap dijaga di dalam populasi adalah hasil perkalian dari kadar generasi yang akan mengalami seleksi dengan besar populasi. Generasi yang akan mengalami proses perkawinan silang adalah besar populasi dikurangi generasi yang dijaga dibagi dengan 2. Kode program dibawah ini merupakan kode untuk parameter algoritma genetika.

```

%% GA parameter
popsize = 20;
mutrate = 0.1;
selection = 0.5;

keep = floor(selection*popsize);

M = ceil((popsize-keep)/2);
odds = 1;
for ii = 2:keep
    odds = [odds ii*ones(1,ii)];
end
Nodds = length(odds);

```

Gambar 4.4.5 Parameter Algoritma Genetika

4.4.6 Inisialisasi Populasi

Pada saat waktu sama dengan 0, maka dilakukan perulangan untuk menghasilkan nilai acak sepanjang populasi yang telah ditentukan. Bilangan acak yang dimaksud adalah permasalahan yang ingin diselesaikan dalam hal ini adalah aktivitas. Jika bilangan acak aktivitas telah ditemukan disetiap aktivitas maka fungsi evaluasi generasi di setiap populasi adalah nilai biaya yang dibawa setiap kromosom yang telah dihitung pada program *waktu.m*. Selanjutnya biaya disetiap kromosom akan diurutkan sesuai dengan nilai terbaik yang dimiliki. Populasi yang dihasilkan adalah populasi yang telah memiliki urutan kromosom sesuai dengan biaya yang dimiliki. Kemudian dinilai *fitness* disetiap kromosom dengan melihat biaya yang paling minimal. Kode program inisialisasi populasi ditunjukkan pada gambar dibawah ini.

```
%% create Initial Population
iga = 0;
for iz = 1:popsiz
    pop(iz,:) = randperm(npar);
end

cost = feval(ff,pop);
[cost,ind] = sort(cost);
pop = pop(ind,:);

minc(1) = min(cost); % Fungsi Fitness
meanc(1) = mean(cost);
```

Gambar 4.4.6 Inisialisasi Populasi

4.4.7 Proses Genetika

Proses genetika dibutuhkan untuk mendapatkan individu-individu terbaik pada generasi selanjutnya. Proses genetika dilakukan selama kriteria untuk berhenti belum ditemukan yaitu maxit. Proses genetika akan berjalan kepada generasi yang berada pada area seleksi. Setelah kromosom-kromosom diurutkan maka akan diambil setengah dari populasi untuk menjalani proses seleksi, dan setengah dari yang telah terpilih akan mengalami proses genetika untuk mendapatkan individu-individu terbaik yang akan dikirim pada generasi selanjutnya.

4.4.7.1 Perkawinan Silang

Perkawinan silang yang digunakan adalah perkawinan silang sederhana. Langkah pertama adalah dengan menyimpan *57able5757e ma* yaitu *57able57* yang berisi kumpulan kromosom ibu, dan *57able5757e pa* yaitu *57able57* yang berisi kumpulan kromosom ayah. Langkah selanjutnya membuat *57able57* yang berisi bilangan acak antara bilangan 1 dan panjang aktivitas. Jika pada *57able57* induk *ma* yang telah diacak tidak sama dengan *57able57 temp* maka dilakukan pertukaran nilai antara *57able57 ma* dan *57able57 pa*. Setelah ditemukan sekumpulan kromosom yang telah disilangkan maka akan dipindahkan ke populasi generasi selanjutnya.

```

%% iterate generation
while lga<maxit
    lga = lga+1;
    %% pair and mate
    pick1 = ceil(Nodds*rand(1,M));
    pick2 = ceil(Nodds*rand(1,M));
    ma = odds(pick1);
    pa = odds(pick2);
    %% mating or crossover
    for ic = 1:M
        mate1 = pop(ma(ic),:);
        mate2 = pop(pa(ic),:);
        indx = 2*(ic-1)+1;
        jobp = ceil(rand*npar);
        temp = mate1;
        job0 = jobp;
        while mate1(jobp)~= temp(job0)
            mate1(jobp) = mate2(jobp);
            mate2(jobp) = temp(jobp);
            jobs = find(temp==mate1(jobp));
            jobp = jobs;
        end
        pop(keep+indx,:) = mate1;
        pop(keep+indx+1,:) = mate2;
    end
end

```

Gambar 4.4.7 Seleksi dan Perkawinan Silang

4.4.7.2 Mutasi

Vektor yang akan dilakukan proses mutasi adalah sebanyak kromosom di dalam populasi dikalikan probabilitas mutasi. Kemudian dilakukan iterasi sepanjang 58able58 yang akan mengalami mutasi. Variabel *row* dan *col* merupakan 58able58 yang akan dimutasikan satu sama lain.

```

%% mutate the population

nmut = ceil(popsize*npar*mutrate);

for ic = 1:nmut
    row1 = ceil(rand*(popsize-1))+1;
    col1 = ceil(rand*npar);
    col2 = ceil(rand*npar);
    temp = pop(row1,col1);
    pop(row1,col1) = pop(row1,col2);
    pop(row1,col2) = temp;
    im(ic) = row1;
end
cost = feval(ff,pop);
%% sort the cost
part = pop;
[cost,ind] = sort(cost);
pop = pop(ind,:);
%% statistic
minc(iga) = min(cost);
meanc(iga) = mean(cost);

```

Gambar 4.4.8 Mutasi

5.5.4 Menampilkan Keluaran Proses Algoritma Genetika

Untuk melihat solusi terbaik dari proses algoritma genetika yang telah dibuat, maka dibutuhkan kode program untuk menampilkan keluaran dari proses-proses yang telah berjalan. Beberapa variabel yang

ditampilkan adalah besar populasi, kadar proses mutasi, biaya kromosom terbaik dan individu terbaik. Kemudian dilakukan juga penampilan dalam bentuk grafik untuk melihat titik konvergensi dari permasalahan. Kode program berikut merupakan program menampilkan hasil dari proses algoritma genetika.

```
%% display the output

day = clock;
disp(datestr(datenum(day(1),day(2),day(3),day(4),day(5),day(6)),0))
disp(['optimized function is ' ff])
format short g
disp(['popsize = ' num2str(popsize) 'mutrate = '
num2str(mutrate) '# par = ' num2str(npar)])
disp([' best cost = ' num2str(cost(1))])

disp([' best solution = '])
disp([num2str(pop(1,:))])
figure(2)
iters = 1:maxit;
plot(iters, minc,iters,meanc,'--');
xlabel('generation');ylabel('cost');
```

Gambar 4.4.9 Menampilkan *Output*

5.5. Implementasi Penjadwalan Proyek dengan PERT ke dalam Perangkat Lunak Matlab

Dalam program ini akan ada satu file yaitu file *PERT.m*. File *PERT.m* merupakan file yang berisi cara untuk mencari nilai ES dan EF, fungsi tujuan dari penjadwalan proyek, lama pengerjaan proyek, dan batasan-batasan tugas akhir yang berjumlah 3 dan masing-masing memiliki fungsi berbeda seperti yang telah dijelaskan pada subbab sebelumnya.

4.5.1 Inisialisasi Data

Pada tahap ini dilakukan inisialisasi data. Pertama dilakukan pengambilan data dari microsoft excel dan disimpan sebagai variabel data yang nantinya akan berupa tabel array. Baris selanjutnya adalah inisialisasi predesesor, aktivitas, duras, penggunaan sumber daya, dan kapasitas sumber daya dengan cara mengambil nilai pada baris dan kolom variabel data. Array selanjutnya adalah variabel tempat menyimpan EF, yang nantinya akan diambil yang paling maksimal pada aktivitas yang memiliki lebih dari 1 predesesor. Dan array yang terakhir adalah tempat menyimpan hasil mode yang dilaksanakan pada suatu aktivitas yang sedang berjalan. Gambar berikut merupakan kode program yang menunjukkan inisialisasi data.

```

% file excel
data = xlsread('D:\\Khairil Juhdi Siregar\\All
about
TA\\dataPen\\data.xls', 'dataPenjadwalan', 'A1:Q34')
;
%inisialisasi data
predesesor = data (:,2:4);
succesor   = data (:,5:7);
aktv       = data (:,1);
dur        = data (:,9);
usedR1     = data (:,10);
usedR2     = data (:,11);
usedR3     = data (:,12);
usedR4     = data (:,13);
R1         = data (:,14);
R2         = data (:,15);
R3         = data (:,16);
R4         = data (:,17);
arr        = [];
y          = [];

```

Gambar 4.5 1 Inisialisasi Data

4.5.2 Mencari Nilai ES dan EF

Tahap pertama dilakukan perulangan sebanyak data yang dimiliki. Kemudian dilakukan pengecekan terhadap jumlah predesesor yang dimiliki setiap aktivitas. Ada tiga kondisi yang dipertimbangkan untuk menghitung ES dan EF dari suatu aktivitas. Kondisi pertama adalah aktivitas yang memiliki lebih dari 1 predesesor. Nilai EF predesesor akan disimpan pada array arr, kemudian akan dicari nilai yang paling maksimal diantara nilai EF predesesor. EF yang paling maksimal merupakan ES aktivitas. Kondisi kedua ialah aktivitas yang memiliki 1 predesesor atau hanya memiliki satu predesesor. Nilai ES aktivitas merupakan nilai EF predesesor. Dan kondisi ketiga adalah aktivitas yang tidak memiliki predesesor, aktivitas ini adalah aktivitas pertama. ES dari aktivitas pertama adalah 0. Nilai EF dari seluruh aktivitas pada setiap kondisi adalah nilai ES ditambah durasi aktivitas. Segmen berikut merupakan kode program pencarian nilai ES dan EF.

```

%% ----- Menghitung ES dan EF (PERT)---
for n = 1:length(data) % Inisiasi
    m = 1;%Sebuah aktivitas mempunyai predesesor
    lebih dari 1
    if predesesor(n,m) ~= 0 && predesesor(n,m+1) ~= 0
        for m = 1:length(predesesor(n,:))
            prd = predesesor(n,m);
            pr = find(aktv == prd);
            ESpr = EF(pr,:);
            arr = [arr; ESpr];
            ES(n,:) = max(arr);
            EF(n,:) = ES(n,)+ dur(n,:);
        end
    %Sebuah aktivitas mempunyai predesesor 1
    elseif predesesor(n,m) ~= 0
        prd = predesesor(n,m);
        pr = find(aktv == prd);
        ES(n,:) = EF(pr,:);
        EF(n,:) = ES(n,)+dur(n,:);
    %Sebuah aktivitas merupakan aktivitas awal dan tidak
    memiliki
    %predesesor
    else
        ES(n,:) = 0;
        EF(n,:) = ES(n,)+ dur(n,:);
    end
end
end

```

Gambar 4.5 2 Mencari nilai ES dan EF

4.5.3 Menampilkan Keluaran PERT

Pada tahap ini dibuat kode program yang bertujuan untuk menunjukkan nilai ES, EF, aktivitas, fungsi_tujuan, dan lama pengerjaan proyek. Segmen ini dibuat untuk menampilkan nilai optimal yang dikeluarkan dengan menggunakan metode PERT. Variabel Jadwal merupakan array yang berisikan urutan aktivitas, nilai ES, dan nilai EF. Fungsi tujuan penjadwalan proyek pada tugas akhir ini yaitu meminimalkan durasi proyek. Nilai EF yang paling maksimal merupakan durasi proyek yang paling optimal dari keseluruhan aktivitas yang dikerjakan. Lama pengerjaan proyek adalah fungsi tujuan yang telah ditentukan sebelumnya. Kode program untuk menampilkan keluaran PERT dapat dilihat pada segmen berikut :

```
Jadwal = [aktv ES EF];  
disp(Jadwal);  
fungsi_tujuan = max(EF);  
disp(['Lama Pengerjaan Proyek = '  
num2str(fungsi_tujuan) ' week'])
```

Gambar 4.5 3 Menampilkan output PERT

4.5.4 Batasan

Setelah nilai ES, EF dan menampilkan output telah didapatkan maka selanjutnya yaitu memasukkan batasan model ke program matlab. Pada tugas akhir ini terdapat tiga batasan yang masing-masing batasan memiliki fungsi yang berbeda seperti yang telah dibahas pada bab sebelumnya. Kode program dari batasan model yang akan dibahas kemudian, merupakan pengembangan dari diagram alur yang telah dibuat sebelumnya.

4.5.4.1 Batasan 1

Batasan pertama pada tugas akhir ini untuk memastikan bahwa setiap aktivitas hanya dijalankan pada 1 mode. Hal ini bertujuan jika pada kenyataannya ada beberapa mode yang dikerjakan pada satu aktivitas. Sehingga dibutuhkan kode program yang mengantisipasi hal tersebut.

Program ini berjalan dengan membandingkan dua baris yaitu baris pertama dan baris setelahnya. Jika aktivitas pada baris pertama sama dengan baris kedua maka dicari nilai minimal dari kedua baris tersebut dan dicek lagi apakah hasil minimal dari kedua baris tersebut sama dengan durasi pada baris ke-n. Jika sama maka baris n diberi nilai 1 dan baris m diberi nilai 0. Jika aktivitas n tidak sama dengan aktivitas m diberi nilai 1. Berikut merupakan kode program yang menunjukkan kondisi-kondisi tersebut.

```
%% batasan 1
for n = 1:length(aktv)-1
    m = n+1;
    A = aktv (n, :);
    B = aktv (m, :);
    if A == B
        mod = min(dur(n, :), dur(m, :));
        if mod == dur(n, :)
            y(n, :) = 1;
            y(m, :) = 0;
        else
            y(n, :) = 0;
            y(m, :) = 1;
        end
    else
        y(m, :) = 1;
    end
end
```

Gambar 4.5 4 Batasan 1

4.5.4.2 Batasan 2

Batasan kedua untuk tugas akhir ini dimaksudkan untuk memastikan bahwa waktu selesai suatu aktivitas lebih kecil dari atau sama dengan waktu mulai aktivitas berikutnya yang berhubungan secara predesesor dengan aktivitas tersebut. Hal ini dimaksudkan agar semua aktivitas dapat berjalan dengan baik dan tidak ada tumpang tindih antar aktivitas.

Pada batasan ini dilakukan perulangan baris sebanyak jumlah data. Ada tiga kondisi yang menjadi acuan dalam perhitungan batasan ini. Kondisi pertama adalah aktivitas memiliki predesesor lebih dari 1. Kondisi kedua ialah aktivitas memiliki predesesor sama dengan 1 atau hanya memiliki satu predesesor. Kondisi ketiga yaitu aktivitas tidak memiliki predesesor. Jika suatu aktivitas berada pada kondisi pertama, maka dari keseluruhan EF predesesornya dicari yang paling besar dan selanjutnya dibandingkan dengan ES aktivitas tersebut. Jika suatu aktivitas berada pada kondisi kedua, maka EF aktivitas predesesornya langsung dibandingkan dengan ES aktivitas tersebut. Jika EF predesesor \geq ES aktivitas maka diberi nilai 1 dan jika tidak maka diberi nilai 0. Jika suatu aktivitas tidak memiliki predesesor maka langsung diberi nilai 0. Kode program selengkapnya dapat dilihat pada segmen berikut :

```

%% Batasan 2
arrpr = [];
for i = 1:length(data)
    j = 1;
    if predesesor(i,j) ~= 0 && predesesor (i,j+1) ~= 0
        for j = 1:length(predesesor(i,:))
            prd = predesesor(i,j);
            pr = find(aktv == prd);
            EFpr = EF(pr,:);
            arrpr = [arrpr;EFpr];
        end
        if arrpr <= ES(i,:)
            bts1(i,:) = 1;
        else
            bts1(i,:) = 0;
        end
    elseif predesesor (i,j) ~=0
        prd = predesesor(i,j);
        pr = find(aktv == prd);
        if EF(pr,:) <= ES(i,:)
            bts1(i,:) = 1;
        else
            bts1(i,:) = 0;
        end
    else
        bts1(i,:) = 1;
    end
end
end

```

Gambar 4.5 5 Batasan 2

4.5.4.3 Batasan 3

Batasan ketiga pada tugas akhir ini yaitu untuk membatasi penggunaan sumber daya pada setiap pengerjaan aktivitas agar kurang dari atau sama dengan kapasitas sumber daya yang tersedia pada periode tertentu. Hal ini akan menghindari adanya kemungkinan kekurangan sumber daya selama proyek berjalan.

Ada empat jenis sumber daya yang digunakan pada tugas akhir ini, sehingga ada empat perulangan yang akan menghitung jumlah sumber daya yang dipakai pada waktu tertentu.

4.5.4.3.1 Sumber Daya 1

Perulangan pada baris dilakukan sepanjang data yang dimiliki. Kemudian pada baris selanjutnya merupakan proses perhitungan variabel *bt2*. Variabel *bt2* merupakan hasil perkalian dari *usedR1* yang merupakan kebutuhan sumber daya pada suatu aktivitas dengan *y* yang merupakan mode aktivitas yang sedang berjalan. Jika *bt2* kurang dari atau sama dengan kapasitas sumber daya atau *R1* dan *bt2* tidak sama dengan 0, maka *bt2* diberi nilai 1. Jika tidak memenuhi kondisi tersebut maka *bt2* diberi nilai 0. Kode program selengkapnya dapat dilihat pada segmen program berikut :

```
%% Batasan 3.1
for i = 1:length(data)
    bt2(i,:) = usedR1(i,:)*y(i,:);
    if bt2(i,:) <= R1(i,:) && bt2(i,:) ~= 0
        bt2(i,:) = 1;
    else
        bt2(i,:) = 0;
    end
end
```

Gambar 4.5 6 Batasan 3

4.5.4.3.2 Sumber Daya 2

Perulangan pada baris dilakukan sepanjang data yang dimiliki. Kemudian pada baris selanjutnya merupakan proses perhitungan variabel *bt3*. Variabel *bt3* merupakan hasil perkalian dari *usedR2* yang merupakan kebutuhan sumber daya pada suatu aktivitas dengan *y* yang merupakan mode aktivitas yang sedang berjalan. Jika

$bt3$ kurang dari atau sama dengan kapasitas sumber daya atau $R2$ dan $bt3$ tidak sama dengan 0, maka $bt3$ diberi nilai 1. Jika tidak memenuhi kondisi tersebut maka $bt3$ diberi nilai 0. Kode program selengkapnya dapat dilihat pada segmen program berikut :

```
%% batasan 3.2
for i = 1:length(data)
    bt3(i,:) = usedR2(i,:)*y(i,:);
    if bt3(i,:) <= R2(i,:) && bt3(i,:) ~= 0
        bt3(i,:) = 1;
    else
        bt3(i,:) = 0;
    end
end
```

Gambar 4.5 7 Batasan 3

4.5.4.3.3 Sumber Daya 3

Perulangan pada baris dilakukan sepanjang data yang dimiliki. Kemudian pada baris selanjutnya merupakan proses perhitungan variabel $bt4$. Variabel $bt4$ merupakan hasil perkalian dari $usedR3$ yang merupakan kebutuhan sumber daya pada suatu aktivitas dengan y yang merupakan mode aktivitas yang sedang berjalan. Jika $bt4$ kurang dari atau sama dengan kapasitas sumber daya atau $R3$ dan $bt4$ tidak sama dengan 0, maka $bt2$ diberi nilai 1. Jika tidak memenuhi kondisi tersebut maka $bt4$ diberi nilai 0. Kode program selengkapnya dapat dilihat pada segmen program berikut :


```

%% batasan 3.3
for i = 1:length(data)
    bt4(i,:) = usedR3(i,:)*y(i,:);
    if bt4(i,:) <= R3(i,:) && bt4(i,:) ~= 0
        bts4(i,:) = 1;
    else
        bts4(i,:) = 0;
    end
end
end

```

Gambar 4.5 8 Batasan 3

4.5.4.3.4 Sumber Daya 4

Perulangan pada baris dilakukan sepanjang data yang dimiliki. Kemudian pada baris selanjutnya merupakan proses perhitungan variabel *bt5*. Variabel *bt5* merupakan hasil perkalian dari *usedR4* yang merupakan kebutuhan sumber daya pada suatu aktivitas dengan *y* yang merupakan mode aktivitas yang sedang berjalan. Jika *bt5* kurang dari atau sama dengan kapasitas sumber daya atau *R4* dan *bt5* tidak sama dengan 0, maka *bt5* diberi nilai 1. Jika tidak memenuhi kondisi tersebut maka *bt5* diberi nilai 0. Kode program selengkapnya dapat dilihat pada segmen program berikut :

```

%% batasan 3.4
for i = 1:length(data)
    bt5(i,:) = usedR4(i,:)*y(i,:);
    if bt5(i,:) <= R4(i,:) && bt5(i,:) ~= 0
        bts5(i,:) = 1;
    else
        bts5(i,:) = 0;
    end
end
end

```

Gambar 4.5 9 Batasan 3

BAB V

UJI COBA DAN ANALISIS HASIL

Pada bab ini dijelaskan mengenai proses uji coba dan analisis terhadap hasil yang diperoleh dari proses implementasi yang telah dibahas pada bab 4 sehingga diharapkan nantinya hasil dari tugas akhir ini benar-benar akurat.

5.5. Lingkungan Uji Coba

Lingkungan uji coba ini menjelaskan mengenai lingkungan pengujian yang digunakan untuk melakukan implementasi tugas akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang digunakan. Spesifikasi perangkat lunak dan perangkat keras yang digunakan dalam pembuatan aplikasi ini ditunjukkan pada tabel 5.1.1 berikut :

Tabel 5.1.1 Perangkat Keras Lingkungan Uji Coba

Perangkat Keras	Spesifikasi
Jenis	Personal Computer
Processor	Intel Core 2Duo
RAM	2 GB
Hard Disk Drive	320GB

Selain itu juga terdapat lingkungan perangkat lunak yang digunakan dalam uji coba model. Tabel 5.1.2 berikut adalah daftar perangkat lunak yang digunakan dalam uji coba.

Tabel 5.1.2 Perangkat Lunak Lingkungan Uji Coba

Perangkat Lunak	Fungsi
Windows 7	Sistem Operasi
Matlab R2009a	Membuat kode program
Microsoft Excel 2007	Tempat penyimpanan data

5.2 Verifikasi

Verifikasi adalah proses untuk memastikan bahwa program telah bebas dari error. Cara untuk melakukan verifikasi yaitu melihat kode program yang dibuat. Apakah kode program yang telah dibuat bebas dari kesalahan, dalam hal ini kesalahan pada kode program adalah munculnya pesan *error*. Setelah kode program telah dipastikan tidak ada *error* maka hal selanjutnya yang akan dilakukan yaitu dengan menjalankan program untuk mengetahui hasil program. Pada gambar-gambar dibawah ini merupakan hasil keluaran program. Dengan adanya hasil keluaran program maka dapat dipastikan bahwa program telah terbebas dari *error*. Langkah verifikasi selanjutnya adalah memastikan setiap aktivitas telah terjadwal, hal ini dapat dilihat dari setiap aktivitas telah memiliki waktu mulai dan waktu selesai.

5.2.1 Menjalankan Program Algoritma Genetika

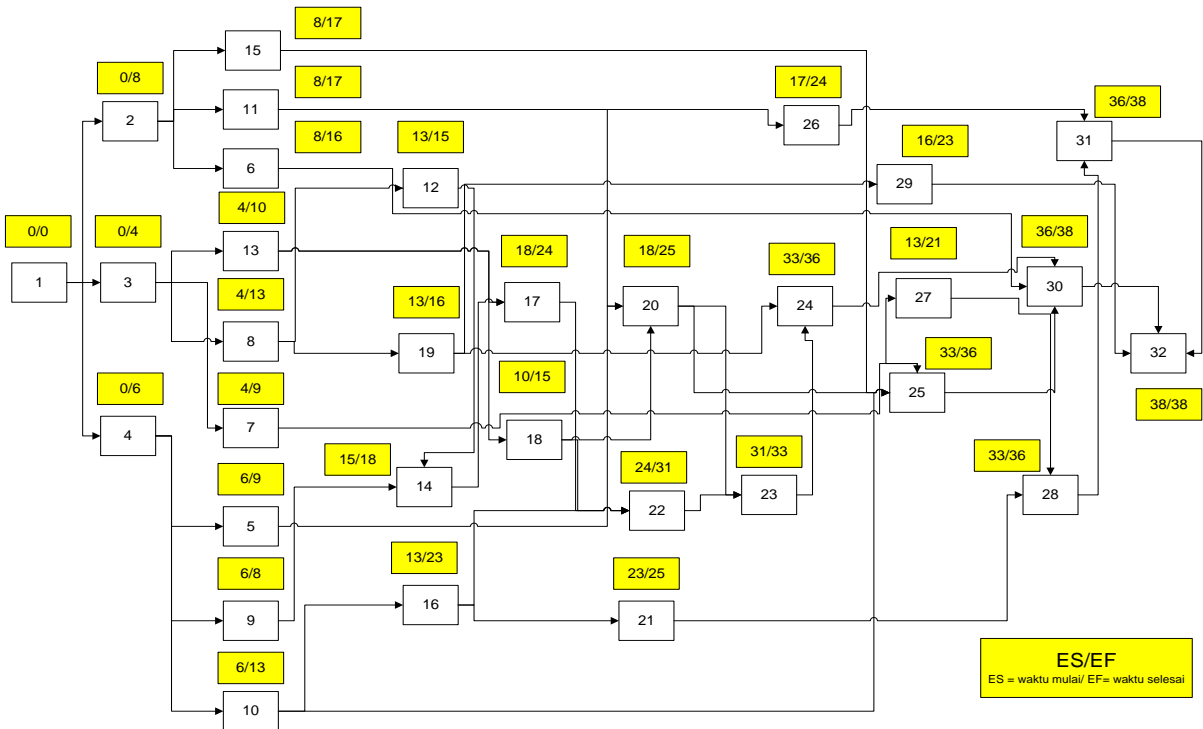
Tabel dibawah ini merupakan hasil dari program algoritma genetika yang telah dijalankan. Tabel 5.2.1 menunjukkan waktu mulai setiap aktivitas dan waktu selesai setiap aktivitas.

**Tabel 5.2 1 Waktu Mulai dan Waktu Selesai
Algoritma Genetika**

Aktivitas	Waktu Mulai	Waktu Selesai
1	0	0
2	0	8
3	0	4
4	0	6
5	6	9
6	8	16
7	4	9
8	4	13
9	6	8

Aktivitas	Waktu Mulai	Waktu Selesai
10	6	13
11	8	17
12	13	15
13	4	10
14	15	18
15	8	17
16	13	23
17	18	24
18	10	15
19	13	16
20	18	25
21	23	25
22	24	31
23	31	33
24	33	36
25	33	36
26	17	24
27	13	21
28	33	36
29	16	23
30	36	38
31	36	38
32	38	38

Gambar 5.2.2 dibawah ini merupakan hasil dari output biaya per kromosom, dengan nilai yaitu 38. Oleh karena itu lama pengerjaan



Gambar 5.2. 1 Waktu mulai dan waktu selesai setiap aktivitas

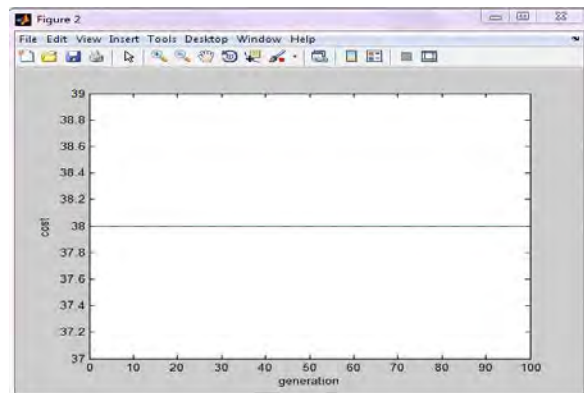
proyek adalah 38 minggu. Bagian selanjutnya adalah individu terbaik yang diambil dari peringkat pertama sepanjang 20 populasi.

Lama Pengerjaan Proyek: 38 week																			
Individu Terbaik:																			
1	22	9	2	27	32	18	8	19	10	29	6	14	23	25	4	12			
15	17	5	30	24	3	16	7	31	26	11	20	28	13	21					

Gambar 5.2.2 Individu terbaik dan lama pengerjaan proyek

Individu terbaik yang ditunjukkan pada gambar diatas bukan urutan pengerjaan proyek. Individu terbaik ini adalah hasil acak dan proses genetika yang dialami oleh kromosom.

Grafik di bawah ini menunjukkan tingkat konvergensi permasalahan pada titik 38. Konvergensi suatu permasalahan diartikan dengan telah tercapainya global optimum dari permasalahan. Tingkat konvergensi merupakan nilai terbaik dari permasalahan yang diselesaikan.



Gambar 5.2.3 Grafik konvergensi generasi

5.2.2 Menjalankan Program PERT

Tabel dan gambar di bawah ini adalah hasil dari perhitungan nilai ES dan EF setiap aktivitas. Fungsi tujuan adalah nilai EF yang paling besar pada aktivitas terakhir, sehingga lama pengerjaan proyek adalah 38 minggu.

**Tabel 5.2.2 Earliest Start dan Earliest Finish
program PERT**

Aktivitas	<i>Earliest Start</i>	<i>Earliest Finish</i>
1	0	0
2	0	8
3	0	4
4	0	6
5	6	9
6	8	16
7	4	9
8	4	13
9	6	8
10	6	13
11	8	17
12	13	15
13	4	10
14	15	18
15	8	17
16	13	23
17	18	24
18	10	15
19	13	16

Aktivitas	<i>Earliest Start</i>	<i>Earliest Finish</i>
20	18	25
21	23	25
22	24	31
23	31	33
24	33	36
25	33	36
26	17	24
27	13	21
28	33	36
29	16	23
30	36	38
31	36	38
32	38	38

Tabel di atas telah menunjukkan bahwa setiap aktivitas telah memiliki *earliest start* dan *earliest finish*. Syarat-syarat yang ditentukan dalam proses verifikasi telah terpenuhi, sehingga proses selanjutnya dapat dilakukan.

Lama Pengerjaan Proyek = 38 week

Gambar 5.2.4 Durasi proyek

5.3 Validasi

Validasi adalah suatu proses untuk memastikan bahwa sebuah program telah sesuai dengan tujuan yang diharapkan. Pada tugas akhir ini terdapat dua proses validasi yaitu validasi terhadap durasi pelaksanaan proyek dan validasi terhadap penggunaan metode optimasi.

Cara yang dilakukan untuk melakukan validasi terhadap durasi pelaksanaan proyek yaitu dengan membandingkan hasil keluaran

program algoritma genetika yang telah dibuat dengan menggunakan perangkat lunak Matlab dengan hasil yang dikeluarkan oleh metode lain, dalam hal ini yaitu PERT. Jika hasil keluaran program algoritma genetika sama dengan hasil keluaran metode PERT maka dapat dikatakan bahwa program tersebut telah valid.

Sedangkan cara yang digunakan untuk melakukan validasi terhadap penggunaan metode optimasi yaitu dengan membandingkan *running time* dari kedua metode. Metode yang *running time* nya lebih kecil maka dapat dikatakan metode yang dipakai lebih efisien daripada metode pembanding. Jadi jika metode algoritma genetika yang dipakai pada tugas akhir ini, *running time*-nya lebih kecil dari metode PERT maka dapat dikatakan penggunaan metode algoritma genetika telah valid karena lebih efisien.

5.3.1 Validasi terhadap Durasi Pelaksanaan Proyek

Seperti yang telah dikatakan sebelumnya, pada tugas akhir ini proses validasi dilakukan dengan membandingkan keluaran antara metode algoritma genetika dan PERT yang telah dibuat programnya pada perangkat lunak Matlab. Berikut adalah langkah-langkah jalannya program.

1. Langkah yang dijalankan yaitu dengan menjalankan program yang telah dibuat pada Matlab. File program yang pertama dibuat yaitu file waktu.m yang bertujuan untuk menjadi sumber data yang dibutuhkan oleh program algoritma genetika dan juga sebagai acuan perhitungan waktu mulai dan waktu selesai dari aktivitas. File yang kedua yaitu file PenjGA.m yang bertujuan sebagai kumpulan proses-proses algoritma genetika.
2. Selanjutnya menjalankan file PenjGA.m yang bertujuan untuk mendapatkan durasi proyek yang paling optimal, dan urutan aktivitas atau individu yang terbaik. Nilai optimal permasalahan mengacu pada fungsi tujuan dari tugas akhir yaitu meminimalkan durasi proyek. Sehingga fungsi fitness

dari kromosom adalah nilai biaya yang paling minimal diantara kromosom yang ada dalam populasi.

Hasil keluaran program algoritma genetika yang telah dibuat pada Matlab dapat dilihat pada gambar 5.3.1 berikut ini.

```
popsiZe = 20 mutrate = 0.1# par = 32
best cost = 38
best solution =
1 22 9 2 27 32 18 8 19 10 29 6 14 23 25 4 12 15 17
```

Gambar 5.3.1 keluaran metode Algoritma Genetika

Setelah mendapatkan keluaran waktu lama pengerjaan proyek dari program algoritma genetika yang dibuat di Matlab yakni 38 minggu, kemudian dilakukan perbandingan hasil dengan menggunakan metode PERT yang dibuat di Matlab.

1. Langkah yang dilakukan yaitu dengan menjalankan file program yang telah dibuat di Matlab. Ada satu file yang dibuat untuk perhitungan durasi proyek dengan metode PERT yaitu PERT.m.
2. Selanjutnya menjalankan file PERT.m untuk mengetahui durasi proyek paling optimal. Durasi proyek paling optimal adalah nilai EF yang paling maksimal.

Hasil keluaran program PERT yang telah dibuat pada Matlab dapat dilihat pada gambar 5.3.1.2 berikut.

```
Lama Pengerjaan Proyek = 38 week
```

Gambar 5.3.2 keluaran metode PERT

Untuk memudahkan proses validasi maka dibuat tabel perbandingan antara keluaran program algoritma genetika dan program PERT. Perbandingan antara hasil program algoritma genetika dan hasil program PERT dapat dilihat pada tabel dibawah ini.

Tabel 5.3.1 Perbandingan Hasil

NO	Metode yang digunakan	Hasil
1	Algoritma Genetika	38 minggu
2	PERT	38 minggu

Berdasarkan tabel 5.3.1 yang merupakan hasil keluaran program Algoritma Genetika dan program PERT yang telah dibuat menggunakan perangkat lunak Matlab maka dapat disimpulkan bahwa model dan program telah valid karena nilai yang dikeluarkan sama.

5.3.2 Validasi Terhadap Penggunaan Metode

Untuk validasi yang dilakukan terhadap penggunaan metode optimasi yaitu dengan membandingkan *running time* yang dibutuhkan metode dalam mengeluarkan hasil yang diharapkan.

- Langkah pertama yang dilakukan dengan menambahkan *source code* penghitungan *running time* pada masing-masing file yang telah dibuat.

```
t = cputime; surf(peaks(40)); e = cputime-t
```

Gambar 5.3.3 kode program perhitungan running time

- Langkah kedua adalah menjalankan file-file yang bersangkutan untuk melihat perbandingan nilai-nilai *running time* yang dihasilkan. Pada tugas akhir ini dilakukan 10 kali percobaan untuk melihat *running time* kedua metode.

Tabel 5.3.2 Perbandingan Running Time

No	Algoritma Genetika	PERT
1	0.0156	0.5148
2	0.0468	0.4680
3	0.0156	0.078001
4	0.0156	0.093601

No	Algoritma Genetika	PERT
5	0.0156	0.078001
6	0.0156	0.0624
7	0.0156	0.078001
8	0.0156	0.078001
9	0.0156	0.078001
10	0.0156	0.078001

Berdasarkan tabel 5.3.2 yang merupakan perbandingan running time antara metode algoritma Genetika dan metode PERT yang dijalankan di perangkat lunak Matlab. Hasil perbandingan dapat dilihat bahwa running time Algoritma Genetika lebih kecil dari metode PERT. Sehingga dapat disimpulkan bahwa dengan ruang lingkup permasalahan yang kompleks, Algoritma Genetika lebih efisien daripada metode PERT.

5.4 Skenario Uji Coba

Skenario ini dilakukan untuk melihat bagaimana hasil yang akan didapatkan dengan perubahan-perubahan yang dilakukan pada parameter-parameter tertentu, apakah hasilnya sama, lebih baik, atau lebih buruk. Skenario ini dilakukan bertujuan melihat tingkat sensitifitas parameter tertentu, dalam hal ini yaitu fungsi objektif permasalahan. Ada dua skenario yang akan dilakukan pada tugas akhir ini.

5.4.1 Mengurangi Jumlah Individu di dalam Populasi

Pada bab 3 sebelumnya telah dikatakan bahwa akan ada dua jenis skenario yang akan dilakukan pada permasalahan penjadwalan proyek menggunakan metode algoritma genetika. Salah satunya adalah dengan menambah atau mengurangi jumlah individu di dalam populasi. Dalam hal ini dilakukan skenario uji coba dengan mengurangi jumlah individu di dalam populasi. Hasil dari uji coba ini adalah seberapa sensitif program yang dibuat terhadap perubahan yang dilakukan.

Gambar dibawah ini merupakan hasil dari skenario uji coba yang telah dilakukan.

```
popsize = 16 mutrate = 0.1 # par = 32
Lama Pengerjaan Proyek = 38
```

Gambar 5.4.1 Skenario pertama

Ukuran individu di dalam populasi yang awalnya berjumlah 20, telah diubah menjadi 16 individu. Dengan perubahan ini ternyata fungsi tujuan yang meminimalkan durasi waktu proyek tetap menghasilkan durasi proyek selama 38 minggu.

5.4.2 Menurunkan probabilitas mutasi

Pada bab 3 sebelumnya telah dikatakan bahwa akan ada dua jenis skenario yang akan dilakukan pada permasalahan penjadwalan proyek menggunakan metode algoritma genetika. Salah satunya adalah dengan meningkatkan atau menurunkan probabilitas mutasi. Dalam hal ini dilakukan skenario uji coba dengan mengurangi kemungkinan proses mutasi. Hasil dari skenario ini juga untuk melihat tingkat sensitivitas dari program yang dibuat untuk menyelesaikan permasalahan penjadwalan proyek. Gambar di bawah ini merupakan hasil dari skenario uji coba yang telah dilakukan.

```
popsize = 16 mutrate = 0.01 # par = 32
Lama Pengerjaan Proyek = 38
```

Gambar 5.4.2 Skenario kedua

Ukuran individu di dalam populasi tetap berjumlah 16, sesuai dengan skenario pertama. Variabel *mutrate* yang pada awalnya bernilai 0.1 telah diubah menjadi 0.01. Perubahan variabel *mutrate* tidak mempengaruhi lama pengerjaan proyek yaitu selama 38 minggu.

Table dibawah ini menunjukkan hasil dari skenario uji coba yang telah dilakukan.

Tabel 5.4 1 Perbandingan hasil uji coba

Uji Coba	Durasi
Tidak ada Perubahan	38 minggu
Pengurangan Jumlah Populasi	38 minggu
Pengurangan probabilitas Mutasi	38 minggu

Dari hasil 81able di atas dapat disimpulkan bahwa permasalahan yang diselesaikan pada tugas akhir ini tidak begitu sensitive dikarenakan tidak adanya perubahan pada durasi proyek.

5.5 Analisis Hasil

Analisis hasil pada tugas akhir ini meliputi beberapa hal yang telah dimodelkan pada bab sebelumnya. Hasil yang akan dianalisa meliputi, hasil optimasi algoritma genetika, dan batasan-batasan yang ada.

5.5.1 Hasil Optimasi

Nilai optimal yang didapatkan dengan meminimalkan durasi proyek menggunakan algoritma genetika adalah 38 minggu. Hasil optimal telah divalidasi dengan dibandingkannya dengan metode lain yaitu PERT. Individu terbaik yang dihasilkan algoritma genetika bukan urutan pengerjaan proyek. Urutan pengerjaan proyek dihasilkan dari membandingkan antara waktu mulai dan waktu selesai suatu aktivitas dengan aktivitas yang lainnya. Urutan pengerjaan proyek harus memenuhi syarat waktu selesai suatu aktivitas lebih kecil dari atau sama dengan waktu mulai aktivitas berikutnya yang berhubungan secara predesesor dengan aktivitas tersebut.

Gambar 5.5.1 adalah urutan pengerjaan proyek dari hasil optimasi yang telah dilakukan. Urutan pengerjaan proyek telah divalidasi dengan batasan kedua yang akan dijelaskan pada subbab berikutnya.

5.5.2 Batasan Pertama

Batasan pertama pada tugas akhir ini untuk memastikan bahwa setiap aktivitas hanya dijalankan pada 1 mode. Hal ini bertujuan jika pada kenyataannya ada beberapa mode yang dikerjakan pada satu aktivitas.

Tabel dibawah ini menunjukkan hasil yang didapatkan dari program yang telah dijalankan. Dari ke-32 aktivitas yang ada, semuanya bernilai 1. Dapat disimpulkan bahwa seluruh aktivitas dijalankan pada satu mode.

Tabel 5.5.2. 1 Batasan 1

Batasan Pertama			
Aktivitas	Nilai	Aktivitas	Nilai
1	0	17	1
2	1	18	1
3	1	19	1
4	1	20	1
5	1	21	1
6	1	22	1
7	1	23	1
8	1	24	1
9	1	25	1
10	1	26	1
11	1	27	1
12	1	28	1
13	1	29	1
14	1	30	1
15	1	31	1
16	1	32	1

5.5.3 Batasan Kedua

Batasan kedua untuk tugas akhir ini dimaksudkan untuk memastikan bahwa waktu selesai suatu aktivitas lebih kecil dari atau sama dengan waktu mulai aktivitas berikutnya yang berhubungan secara predesesor dengan aktivitas tersebut. Hal ini dimaksudkan agar semua aktivitas dapat berjalan dengan baik dan tidak ada tumpang tindih antar aktivitas.

Tabel dibawah ini menunjukkan hasil yang didapatkan dari program yang telah dijalankan. Nilai 1 jika batasan memenuhi, dan nilai 0 jika batasan tidak memenuhi. Dari gambar dapat dilihat bahwa semua aktivitas bernilai 1. Dapat disimpulkan bahwa seluruh aktivitas yang dijalankan tidak mengalami tumpang tindih dalam pengerjaan.

Tabel 5.5.2. 2 Batasan 2

Batasan kedua			
Aktivitas	Nilai	Aktivitas	Nilai
1	1	17	1
2	1	18	1
3	1	19	1
4	1	20	1
5	1	21	1
6	1	22	1
7	1	23	1
8	1	24	1
9	1	25	1
10	1	26	1
11	1	27	1
12	1	28	1
13	1	29	1
14	1	30	1
15	1	31	1
16	1	32	1

5.5.4 Batasan Ketiga

Batasan ketiga pada tugas akhir ini yaitu untuk membatasi penggunaan sumber daya pada setiap pengerjaan aktivitas agar kurang dari atau sama dengan kapasitas sumber daya yang tersedia pada periode tertentu. Hal ini akan menghindari adanya kemungkinan kekurangan sumber daya selama proyek berjalan. Ada empat jenis sumber daya yang digunakan pada tugas akhir ini.

5.5.4.1 Sumber Daya 1

Variabel $bt2$ merupakan hasil perkalian dari $usedR1$ yang merupakan kebutuhan sumber daya pada suatu aktivitas dengan y yang merupakan mode aktivitas yang sedang berjalan. Jika $bt2$ kurang dari atau sama dengan kapasitas sumber daya atau $R1$ dan $bt2$ tidak sama dengan 0, maka $bt2$ diberi nilai 1. Jika tidak memenuhi kondisi tersebut maka $bt2$ diberi nilai 0. Namun pada tugas akhir ini, tidak semua aktivitas yang bernilai 0 diartikan tidak memenuhi kondisi, aktivitas diberi nilai 0 juga jika tidak membutuhkan sumber daya tertentu.

Gambar di bawah ini menunjukkan hasil yang diperoleh dari program yang telah dijalankan. Dari pengecekan yang telah dilakukan didapatkan hasil bahwa setiap aktivitas yang membutuhkan sumber daya 1 pada waktu tertentu tidak melebihi kapasitas sumber daya 1. Hal itu dapat dilihat dari hasil yang menunjukkan nilai 1.

Tabel 5.5.2. 3 sumber daya 1

Sumber daya 1			
Aktivitas	Nilai	Aktivitas	Nilai
1	0	17	0
2	1	18	0
3	1	19	0
4	0	20	0
5	1	21	0
6	0	22	1
7	1	23	1

Aktivitas	Nilai	Aktivitas	Nilai
8	0	24	0
9	1	25	1
10	0	26	0
11	0	27	0
12	0	28	0
13	1	29	0
14	0	30	0
15	1	31	0
16	0	32	0

5.5.4.2 Sumber Daya 2

Variabel $bt3$ merupakan hasil perkalian dari $usedR2$ yang merupakan kebutuhan sumber daya pada suatu aktivitas dengan y yang merupakan mode aktivitas yang sedang berjalan. Jika $bt3$ kurang dari atau sama dengan kapasitas sumber daya atau $R2$ dan $bt3$ tidak sama dengan 0, maka $bt3$ diberi nilai 1. Jika tidak memenuhi kondisi tersebut maka $bt3$ diberi nilai 0. Namun pada tugas akhir ini, tidak semua aktivitas yang bernilai 0 diartikan tidak memenuhi kondisi, aktivitas diberi nilai 0 juga jika tidak membutuhkan sumber daya tertentu.

Gambar dibawah ini menunjukkan hasil yang didapatkan dari program yang telah dijalankan. Dari pengecekan yang telah dilakukan didapatkan hasil bahwa setiap aktivitas yang membutuhkan sumber daya 2 pada waktu tertentu tidak melebihi kapasitas sumber daya 2. Hal itu dapat dilihat dari hasil yang menunjukkan nilai 1.

Tabel 5.5.2. 4 sumber daya 2

Sumber daya 2			
Aktivitas	Nilai	Aktivitas	Nilai
1	0	17	0
2	0	18	0
3	0	19	1
4	0	20	1

Aktivitas	Nilai	Aktivitas	Nilai
5	0	21	0
6	0	22	0
7	0	23	0
8	1	24	1
9	0	25	0
10	0	26	0
11	1	27	0
12	1	28	1
13	0	29	1
14	1	30	1
15	0	31	0
16	0	32	0

5.5.4.3 Sumber Daya 3

Variabel $bt4$ merupakan hasil perkalian dari $usedR3$ yang merupakan kebutuhan sumber daya pada suatu aktivitas dengan y yang merupakan mode aktivitas yang sedang berjalan. Jika $bt4$ kurang dari atau sama dengan kapasitas sumber daya atau $R3$ dan $bt4$ tidak sama dengan 0, maka $bt2$ diberi nilai 1. Jika tidak memenuhi kondisi tersebut maka $bt4$ diberi nilai 0. Namun pada tugas akhir ini, tidak semua aktivitas yang bernilai 0 diartikan tidak memenuhi kondisi, aktivitas diberi nilai 0 juga jika tidak membutuhkan sumber daya tertentu.

Gambar dibawah ini menunjukkan hasil dari program yang telah dijalankan. Dari pengecekan yang telah dilakukan didapatkan hasil bahwa setiap aktivitas yang membutuhkan sumber daya 3 pada waktu tertentu tidak melebihi kapasitas sumber daya 3. Hal itu dapat dilihat dari hasil yang menunjukkan nilai 1.

Tabel 5.5.2. 5 sumber daya 3

Sumber daya 3			
Aktivitas	Nilai	Aktivitas	Nilai
1	0	17	0
2	0	18	0

Aktivitas	Nilai	Aktivitas	Nilai
3	0	19	0
4	0	20	0
5	0	21	0
6	0	22	0
7	0	23	0
8	0	24	0
9	0	25	0
10	0	26	1
11	0	27	0
12	0	28	0
13	0	29	0
14	0	30	0
15	0	31	1
16	0	32	0

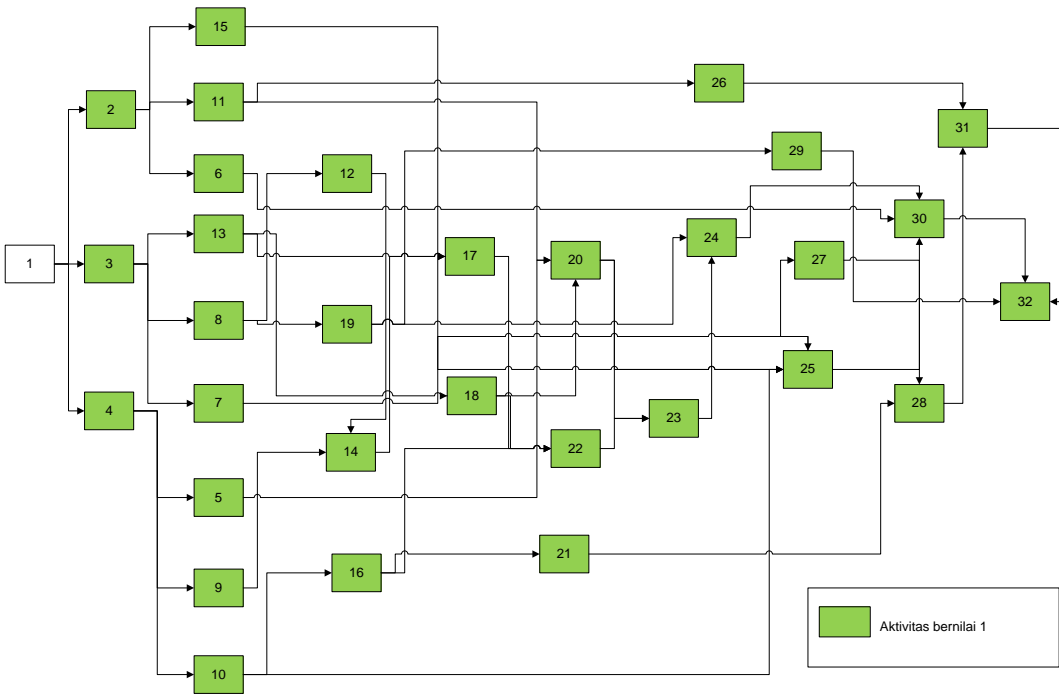
5.5.4.4 Sumber Daya 4

Variabel $bt5$ merupakan hasil perkalian dari $usedR4$ yang merupakan kebutuhan sumber daya pada suatu aktivitas dengan y yang merupakan mode aktivitas yang sedang berjalan. Jika $bt5$ kurang dari atau sama dengan kapasitas sumber daya atau $R4$ dan $bt5$ tidak sama dengan 0, maka $bt5$ diberi nilai 1. Jika tidak memenuhi kondisi tersebut maka $bt5$ diberi nilai 0. Namun pada tugas akhir ini, tidak semua aktivitas yang bernilai 0 diartikan tidak memenuhi kondisi, aktivitas diberi nilai 0 juga jika tidak membutuhkan sumber daya tertentu.

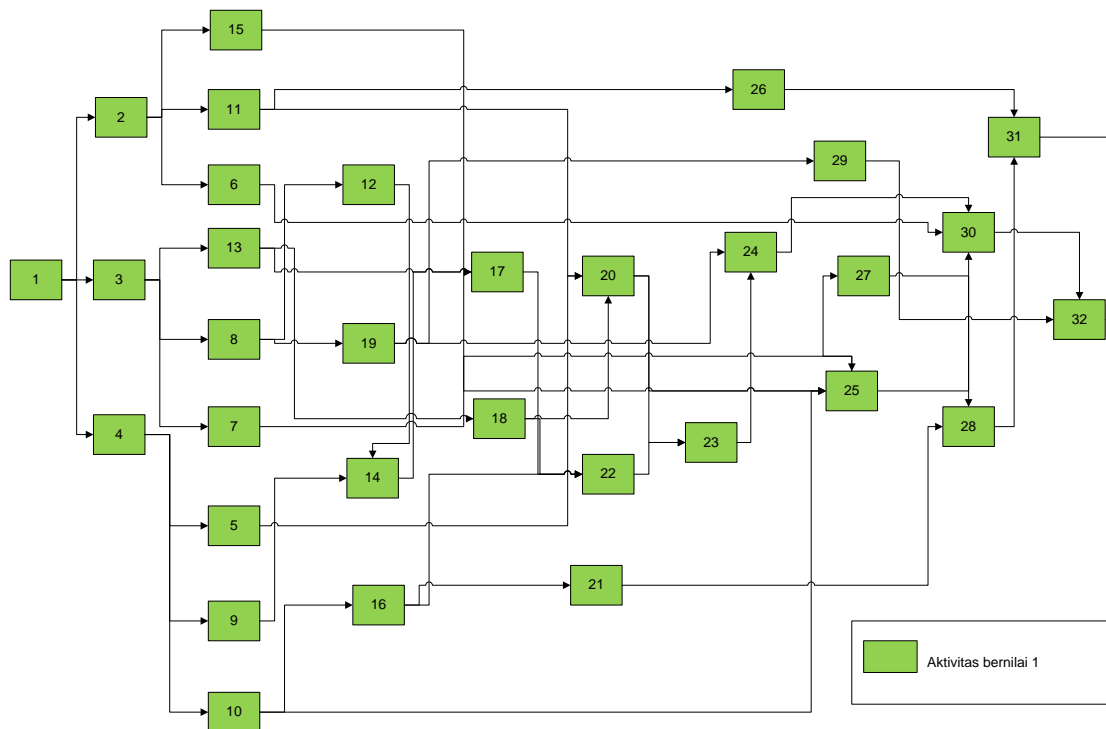
Gambar dibawah ini menunjukkan hasil yang didapatkan dari program yang telah dijalankan. Dari pengecekan yang telah dilakukan didapatkan hasil bahwa setiap aktivitas yang membutuhkan sumber daya 4 pada waktu tertentu tidak melebihi kapasitas sumber daya 4. Hal itu dapat dilihat dari hasil yang menunjukkan nilai 1.

Tabel 5.5.2. 6 sumber daya 4

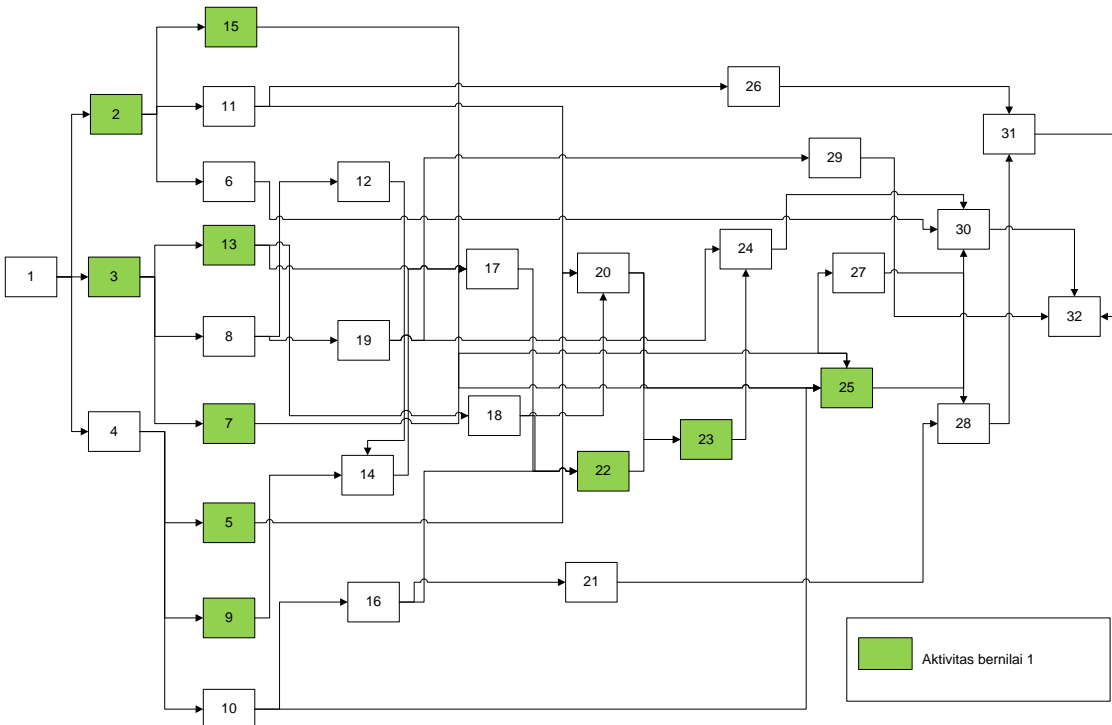
Sumber daya 4			
Aktivitas	Nilai	Aktivitas	Nilai
1	0	17	1
2	0	18	1
3	0	19	0
4	1	20	0
5	0	21	1
6	1	22	0
7	0	23	0
8	0	24	0
9	0	25	0
10	1	26	0
11	0	27	1
12	0	28	0
13	0	29	0
14	0	30	0
15	0	31	0
16	1	32	0



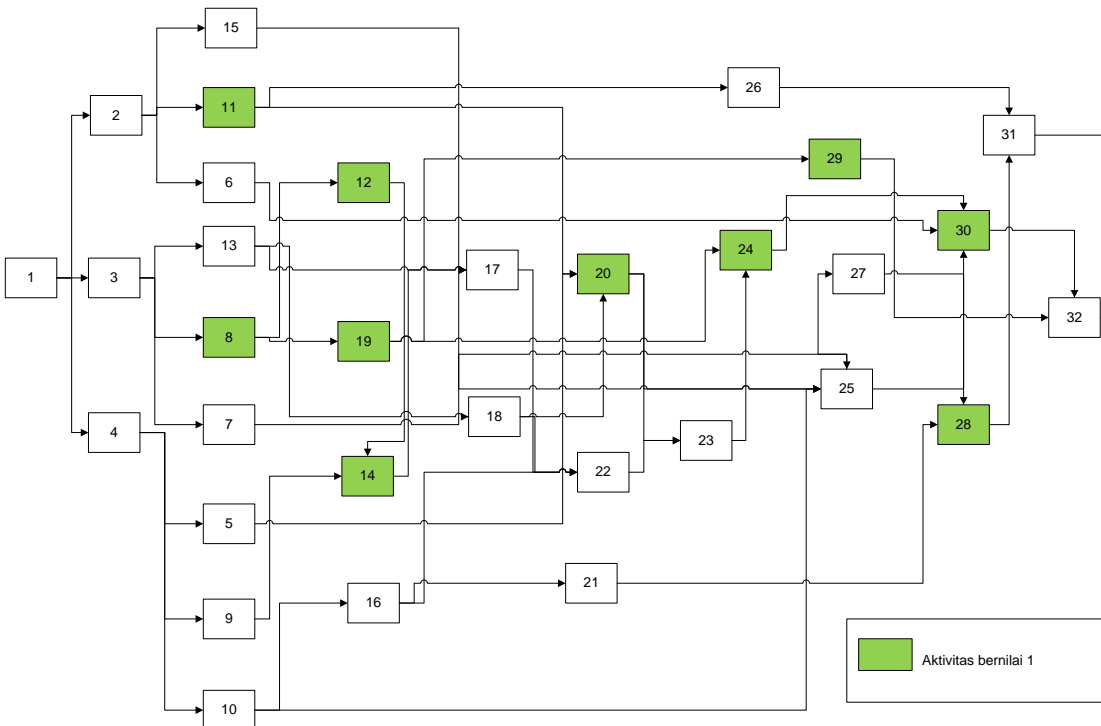
Gambar 5.5. 1 Batasan Pertama



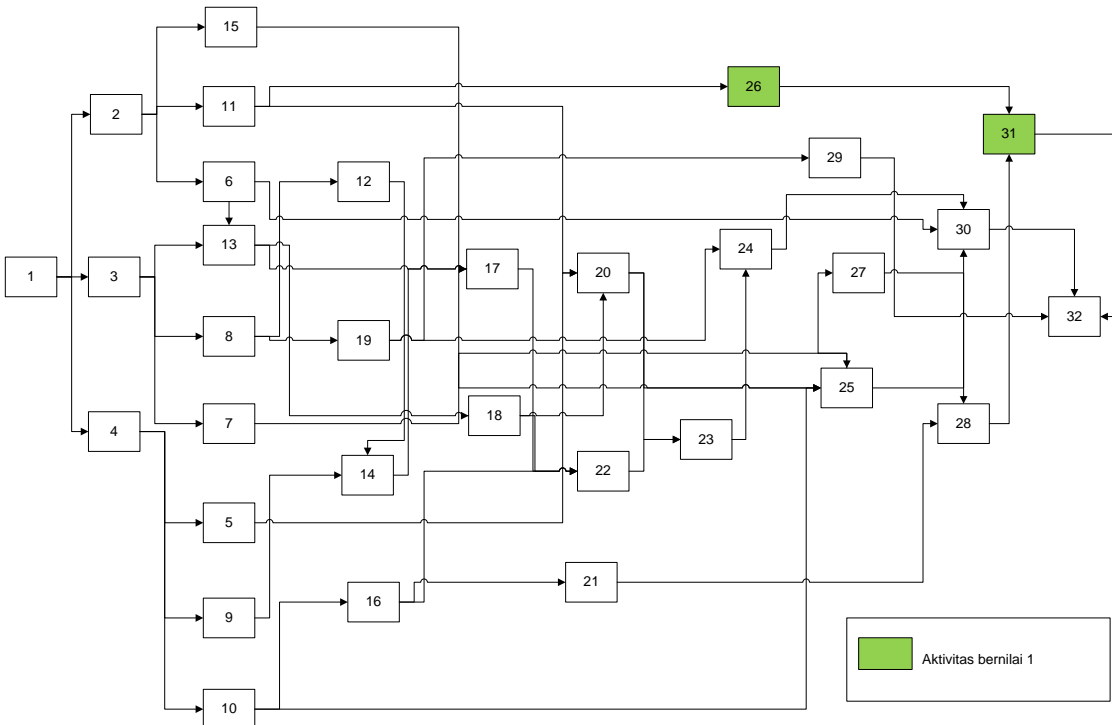
Gambar 5.5. 2 Batasan Kedua



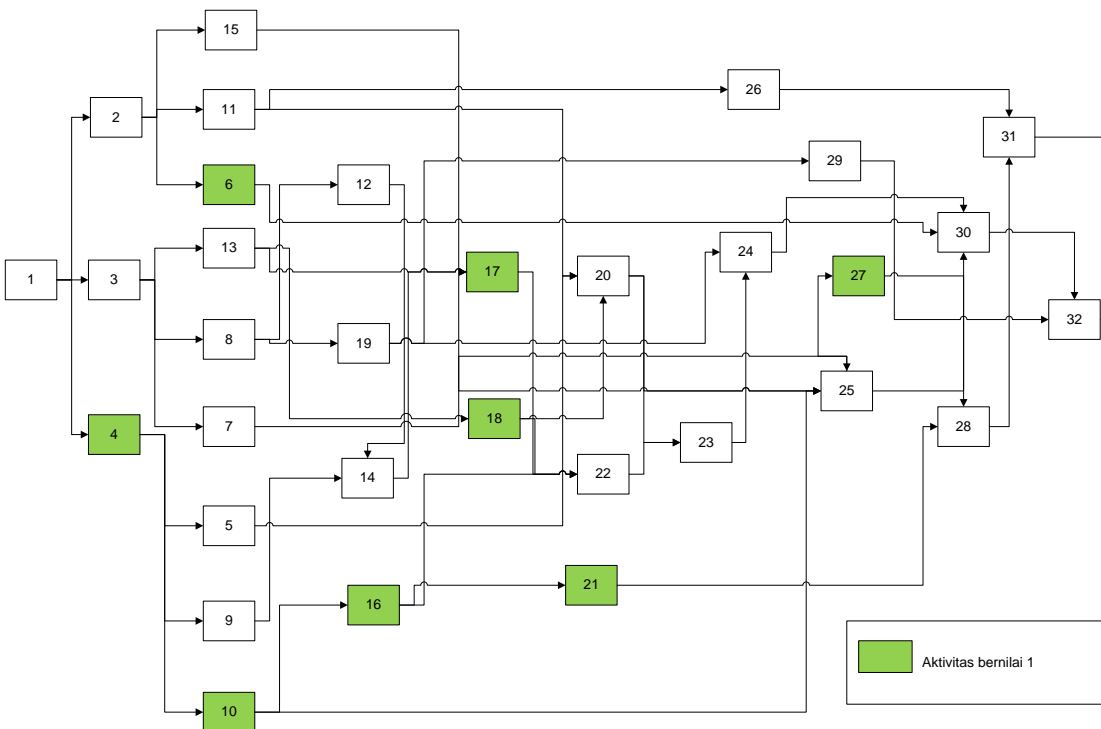
Gambar 5.5. 3 Batasan Ketiga sumber daya 1



Gambar 5.5. 4 Batasan Ketiga sumber daya 2



Gambar 5.5. 5 Batasan Ketiga sumber daya 3



Gambar 5.5. 6 Batasan Ketiga sumber daya 4

BAB VI PENUTUP

Pada bab ini dibahas mengenai kesimpulan dari semua proses yang telah dilakukan.

6.1.Kesimpulan

Berdasarkan proses-proses yang telah dilakukan dalam pengerjaan tugas akhir ini maka ada beberapa kesimpulan yang dapat diambil, diantaranya adalah :

1. Metode Algoritma Genetika mampu menghasilkan durasi optimal dari permasalahan penjadwalan proyek yang diselesaikan pada tugas akhir ini yaitu 38 minggu.
2. Hasil optimal dari algoritma genetika disimpulkan valid karena telah dibandingkan dengan hasil optimal dari metode PERT.
3. Dari uji coba yang telah dilakukan, disimpulkan bahwa algoritma genetika lebih efisien daripada metode PERT
4. Untuk menghindari adanya ketidak bersediaan kapasitas sumber daya tertentu, maka dibutuhkan aturan yang membatasi pemakaian sumber daya yang ada.
5. Tingkat sensitifitas fungsi tujuan terhadap perubahan yang dilakukan pada program tidak begitu sensitif.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- A.Lawrence, J. &. (2002). " *Applied Management Science: Modelling, Spreadsheet Analysis, and Communication for Decision Making*". California: California State University Fullerton: Jhon Wiley & Son.Inc.
- Beasley, D. B. (1993). "An overview of genetic algorithms: Part 1, fundamentals". *University Computing* , 58-69.
- Carl, K. &. (2004). "Time-line based model for software project scheduling with genetic algorithms". *European Journal of Operational Research.Elsevier B.V.*
- Kolisch, R. a. (1996). " PSPLIB - A project scheduling library". *European Journal of Operational Research* , 205-216.
- M.Pyle, T. B. (2004). "An Effective Algorithm For Project Scheduling With Arbitrary TemporalConstraints". *American Association for Artificial Intelligence*.
- Omara, F. A. (2009). " Genetic algorithms for task scheduling problem ". *European Journal of Operational Research.Elsevier B.V.*
- Santosa, B. &. (2011). " *Metoda Metaheuristik: Konsep dan Implementasi.* ". Guna Widya.
- Holland, J. H. (1975). "Adaptation in Natural and Artificial Systems". *Republished by the MIT press* .

E Alba, J. C. (2004). "Genetic and Evolutionary Computation".
International Journal Foundation , 852-863.



RIWAYAT PENULIS

Penulis dilahirkan di Sialogo, 26 Agustus 1990, anak kedua dari empat bersaudara dan merupakan anak dari Bapak H. Timbul Siregar, S.Ag dan Ibu Hj. Derliana Pulungan. Penulis telah menempuh pendidikan formal yaitu di TK Aisyiah Padang Sidempuan, SDN.200117/26 Padang Sidempuan, dan menempuh pendidikan Tsanawiyah dan Aliyah di Pondok Pesantren

Darul Mursyid, Tapanuli Selatan.

Penulis masuk di jurusan Sistem Informasi FTIF - Institut Teknologi Sepuluh Nopember (ITS) Surabaya pada tahun 2009 melalui jalur Program Beasiswa Santri Berprestasi(PBSB) Kementerian Agama Indonesia. Di jurusan tersebut penulis terdaftar dengan NRP 5209100710. Di Jurusan Sistem Informasi ini penulis mengambil bidang minat *Decision Support System* dan *Business Intelligence* untuk tugas akhir.

Selama menempuh pendidikan S1 ini penulis juga menjadi asisten praktikum untuk mata kuliah Statistika, Kalkulus dan Aljabar Linear, Rekayasa Kebutuhan Perangkat Lunak, dan Manajemen Sains Lanjut. Selain itu, penulis juga mengikuti organisasi mahasiswa penerima beasiswa santri berprestasi, yaitu Community of Santri Scholar of Manestry of Religion Affair atau CSSMoRA ITS dan dipercaya menjadi Kepala Departemen Kesejahteraan Mahasantri 2011-2012. Penulis juga pernah menjadi bagian redaksi Majalah Santri Nasional.

Penulis dapat dihubungi melalui email khairilsiregar354@gmail.com untuk keperluan penelitian.

Halaman ini sengaja dikosongkan

LAMPIRAN A DATA INPUTAN

Pada lampiran A akan ditampilkan data yang diambil dari *Project Scheduling Problem Library*(PSPLIB). Data berikut merupakan data yang telah diolah ke dalam *Microsoft Excel*.

A-1 1. Data Pengerjaan Proyek

aktivitas	predececors			mode	durasi
1	0	0	0	1	0
2	1	0	0	1	8
3	1	0	0	1	4
4	1	0	0	1	6
5	4	0	0	1	3
6	2	0	0	1	8
7	3	0	0	1	5
8	3	0	0	1	9
9	4	0	0	1	2
10	4	0	0	1	7
11	2	0	0	1	9
12	8	0	0	1	2
13	3	0	0	1	6
14	9	12	0	1	3
15	2	0	0	1	9
16	10	0	0	1	10
17	13	14	0	1	6
18	13	0	0	1	5
19	8	0	0	1	3

A-2

aktivitas	predececors			mode	durasi
20	5	11	18	1	7
21	16	0	0	1	2
22	16	17	18	1	7
23	20	22	0	1	2
24	19	23	0	1	3
25	10	20	0	1	3
26	11	0	0	1	7
27	8	0	0	1	8
28	21	27	0	1	3
29	19	0	0	1	7
30	6	24	25	1	2
31	26	28	0	1	2
32	29	30	31	1	0

A-1 2 Sumber daya yang dibutuhkan

aktivitas	usedR1	usedR2	usedR3	usedR4
1	0	0	0	0
2	4	0	0	0
3	10	0	0	0
4	0	0	0	3
5	3	0	0	0
6	0	0	0	8
7	4	0	0	0
8	0	1	0	0
9	6	0	0	0

aktivitas	usedR1	usedR2	usedR3	usedR4
10	0	0	0	1
11	0	5	0	0
12	0	7	0	0
13	4	0	0	0
14	0	8	0	0
15	3	0	0	0
16	0	0	0	5
17	0	0	0	8
18	0	0	0	7
19	0	1	0	0
20	0	10	0	0
21	0	0	0	6
22	2	0	0	0
23	3	0	0	0
24	0	9	0	0
25	4	0	0	0
26	0	0	4	0
27	0	0	0	7
28	0	8	0	0
29	0	7	0	0
30	0	7	0	0
31	0	0	2	0
32	0	0	0	0

Halaman ini sengaja dikosongkan

LAMPIRAN B HASIL OPTIMASI

Lampiran berikut ini adalah hasil dari program yang dijalankan. Tabel pertama adalah hasil atau keluaran program algoritma genetika. Kemudian dilanjutkan dengan tabel hasil atau keluaran program PERT. Tabel yang terakhir merupakan nilai yang diperoleh setiap aktivitas dengan batasan-batasan yang telah ditentukan.

Tabel B- 1Output Algoritma Genetika

Algoritma Genetika		
Aktivitas	Waktu Mulai	Waktu Selesai
1	0	0
2	0	8
3	0	4
4	0	6
5	6	9
6	8	16
7	4	9
8	4	13
9	6	8
10	6	13
11	8	17
12	13	15
13	4	10
14	15	18
15	8	17
16	13	23
17	18	24

Aktivitas	Waktu Mulai	Waktu Selesai
18	10	15
19	13	16
20	18	25
21	23	25
22	24	31
23	31	33
24	33	36
25	33	36
26	17	24
27	13	21
28	33	36
29	16	23
30	36	38
31	36	38
32	38	38

Tabel B- 2Output program PERT

PERT		
Aktivitas	<i>Earliest Start</i>	<i>Earliest Finish</i>
1	0	0
2	0	8
3	0	4
4	0	6
5	6	9
6	8	16
7	4	9

Aktivitas	<i>Earliest Start</i>	<i>Earliest Finish</i>
8	4	13
9	6	8
10	6	13
11	8	17
12	13	15
13	4	10
14	15	18
15	8	17
16	13	23
17	18	24
18	10	15
19	13	16
20	18	25
21	23	25
22	24	31
23	31	33
24	33	36
25	33	36
26	17	24
27	13	21
28	33	36
29	16	23
30	36	38
31	36	38
32	38	38

Tabel B- 3 Batasan pertama dan kedua

	Batasan Pertama	Batasan Kedua
Aktivitas	Nilai	Nilai
1	0	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1
13	1	1
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	1	1
20	1	1
21	1	1
22	1	1
23	1	1
24	1	1

Aktivitas	Nilai	Nilai
25	1	1
26	1	1
27	1	1
28	1	1
29	1	1
30	1	1
31	1	1
32	1	1

Tabel B- 4 Batasan Ketiga

	Batasan Ketiga			
	Sumber daya 1	Sumber daya 2	Sumber daya 3	Sumber daya 4
Aktivitas	Nilai	Nilai	Nilai	Nilai
1	0	0	0	0
2	1	0	0	0
3	1	0	0	0
4	0	0	0	1
5	1	0	0	0
6	0	0	0	1
7	1	0	0	0
8	0	1	0	0
9	1	0	0	0
10	0	0	0	1
11	0	1	0	0
12	0	1	0	0

	Sumber daya 1	Sumber daya 2	Sumber daya 3	Sumber daya 4
Aktivitas	Nilai	Nilai	Nilai	Nilai
13	1	0	0	0
14	0	1	0	0
15	1	0	0	0
16	0	0	0	1
17	0	0	0	1
18	0	0	0	1
19	0	1	0	0
20	0	1	0	0
21	0	0	0	1
22	1	0	0	0
23	1	0	0	0
24	0	1	0	0
25	1	0	0	0
26	0	0	1	0
27	0	0	0	1
28	0	1	0	0
29	0	1	0	0
30	0	1	0	0
31	0	0	1	0
32	0	0	0	0