



UNIVERSITAS INDONESIA

**OPTIMASI PENJADWALAN KARYAWAN PENGAWAS
PEMBANGUNAN KAPAL DENGAN MENGGUNAKAN
ALGORITMA GENETIKA**

SKRIPSI

**ANDREW LAMPATAR SIANTURI
0806337440**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2012**



UNIVERSITAS INDONESIA

**OPTIMASI PENJADWALAN KARYAWAN PENGAWAS
PEMBANGUNAN KAPAL DENGAN MENGGUNAKAN
ALGORITMA GENETIKA**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**ANDREW LAMPATAR SIANTURI
0806337440**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2012**

ii

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun yang dirujuk
telah saya nyatakan dengan benar.**



Nama : ANDREW LAMPATAR SIANTURI
NPM : 0806337440
Tanda tangan : 
Tanggal : Juni 2012


HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Andrew Lampatar Sianturi
NPM : 0806337440
Program Studi : Teknik Industri
Judul Skripsi : Optimasi Penjadwalan Karyawan Pengawas
Pembangunan Kapal Dengan Menggunakan
Algoritma Genetika


Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Industri, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Ir. Amar Rachman, MEIM ()

Penguji : Dr. Akhmad Hidayatno, S.T.,MBT ()

Penguji : Armand OM., S.T.,M.Sc. ()

Penguji : Romadhani Ardi,S.T.,M.T ()

Ditetapkan di : Depok
Tanggal : 20 Juni 2012

KATA PENGANTAR

Puji syukur penulis ucapkan kepada Tuhan Yesus Kristus atas kasih dan penyertaan-Nya yang tidak pernah berhenti sehingga penulis dapat menyelesaikan skripsi ini tepat waktu. Penulis juga ingin menyampaikan ucapan terima kasih kepada:

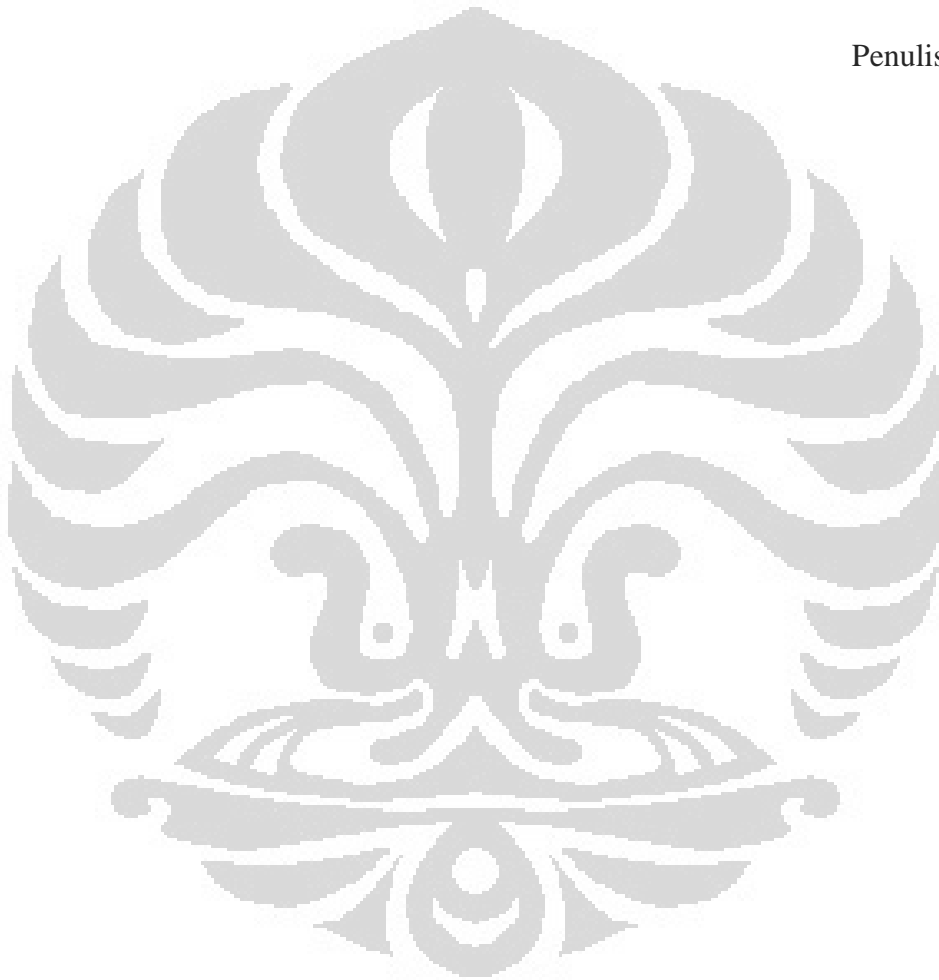
1. Bapak Ir. Amar Rachman, MEIM, selaku dosen pembimbing skripsi yang telah membimbing dan memberikan masukan dan arahan selama penulisan.
2. Bapak Sumarsono, ST, MT, selaku dosen pembimbing skripsi II yang telah memberikan masukan selama penulisan.
3. Departemen Pembangunan Kapal PT. Pertamina dan Bapak Slamet Harianto, yang telah memberikan banyak bantuaan penjelasan dan data yang dibutuhkan kepada penulis.
4. Bapak Akhmad Hidayatno, selaku dosen pembimbing akademis, dan dosen-dosen lainnya, yang telah memberikan masukan dan ilmu selama penulis melakukan pembelajaran.
5. Kedua orang tua terkasih, adik-adikku tercinta, Grace, Advend, Agnes, dan Ardika, dan keluarga besar lainnya atas segala doa dan dukungan kepada penulis selama ini.
6. Paulus, Mariana, Jessica, Friska, Gabriela, Kristina, Roberton, Stefani, Eltina, Rizal, Andalusia, dan Andreas, dan semua teman-teman Teknik Industri UI 2008 lainnya selaku teman seperjuangan penulis selama masa perkuliahan, khususnya dalam pengerjaan tugas akhir ini.
7. Fariz Muriyadi dan Rahardika Arista selaku sahabat-sahabat penulis yang telah banyak berbagi pengalaman, kegembiraan, kesusahan dan banyak hal lainnya.
8. Shanny Dorothy, Samuel Guswindo dan Apraz atas semua bantuan yang telah diberikan, khususnya dalam menyelesaikan tugas akhir dengan program MATLAB.
9. Teman-teman PSPO UI Ribka, Angel Ruth, Maratur, Hariyanti, Saut, Ricky, Johannes, , Teta, Surya, Shelvyana, Ehon, Shinta, Dee, Nia, Deco, Pebri, Jupita, Hutomo, Crist, Agnes, Joshua, Mikhael, Messi, Merry, Henny, Juwita, Monic, Gunawan, Erwin, Erika, Ennike, Eko, Areta, dan Eliz yang telah

memberikan semangat, dukungan doa dan memberikan banyak momen yang indah selama latihan PSPO.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan. Selain itu penulis berharap skripsi ini dapat memberikan manfaat bagi semua pihak yang membacanya.

Depok, Juni 2012

Penulis



**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Andrew Lampatar Sianturi
NPM : 0806337440
Departemen : Teknik Industri
Fakultas : Teknik
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

Optimasi Penjadwalan Karyawan Pengawas Pembangunan Kapal Dengan Menggunakan Algoritma Genetika

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : Juni 2012
Yang Menyatakan


(Andrew Lampatar Sianturi)

ABSTRAK

Nama : Andrew Lampatar Sianturi
Program Studi : Teknik Industri
Judul : Optimasi Penjadwalan Karyawan Pengawas Pembangunan Kapal Dengan Menggunakan Algoritma Genetika

Permasalahan penjadwalan karyawan secara umum adalah masalah untuk menentukan berapa banyak pekerja yang dibutuhkan untuk ditugaskan terhadap setiap periode waktu kerja yang direncanakan oleh sebuah organisasi dengan tujuan mengcover seluruh penugasan dengan biaya minimum. Permasalahan penjadwalan karyawan mempunyai kompleksitas tinggi karena banyak faktor yang harus dipertimbangkan seperti aturan yang berlaku serta biaya yang ditimbulkan. Metode yang digunakan untuk optimasi penjadwalan karyawan dalam penelitian ini adalah Algoritma Genetika. Solusi melalui metode ini diharapkan dapat membantu perusahaan dalam mengoptimal jadwal karyawan pengawas pembangunan kapal.

Kata kunci:

Penjadwalan, Penjadwalan karyawan, Optimasi, Algoritma Genetika

ABSTRACT

Name : Andrew Lampatar Sianturi
Study Program : Industrial Engineering
Title : Crew Scheduling Optimization of Ship Building Surveyor Using Genetic Algorithm.

Crew scheduling problem is in general, the problem of determining how many workers must be assigned to each of the planning periods of work time for an organization in order to cover all assignments at minimum cost. Crew scheduling has a high complexity because of many aspects must be considered, such as work-rule agreements and cost of individual assignment. The method to optimize the crew scheduling problem is Genetic Algorithm. The solution through this method could help the organization to optimize workforce scheduling of ship building surveyor.

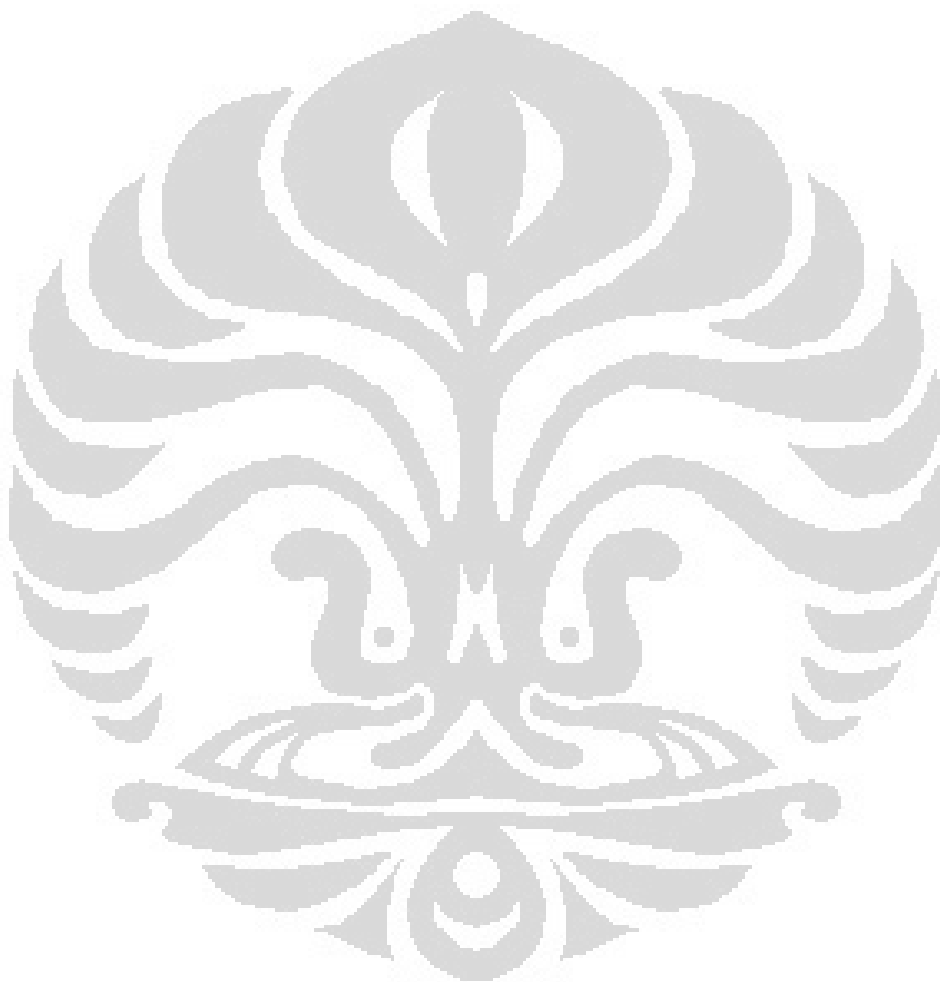
Key words:

Scheduling, Crew Scheduling, Genetic Algorithm

DAFTAR ISI

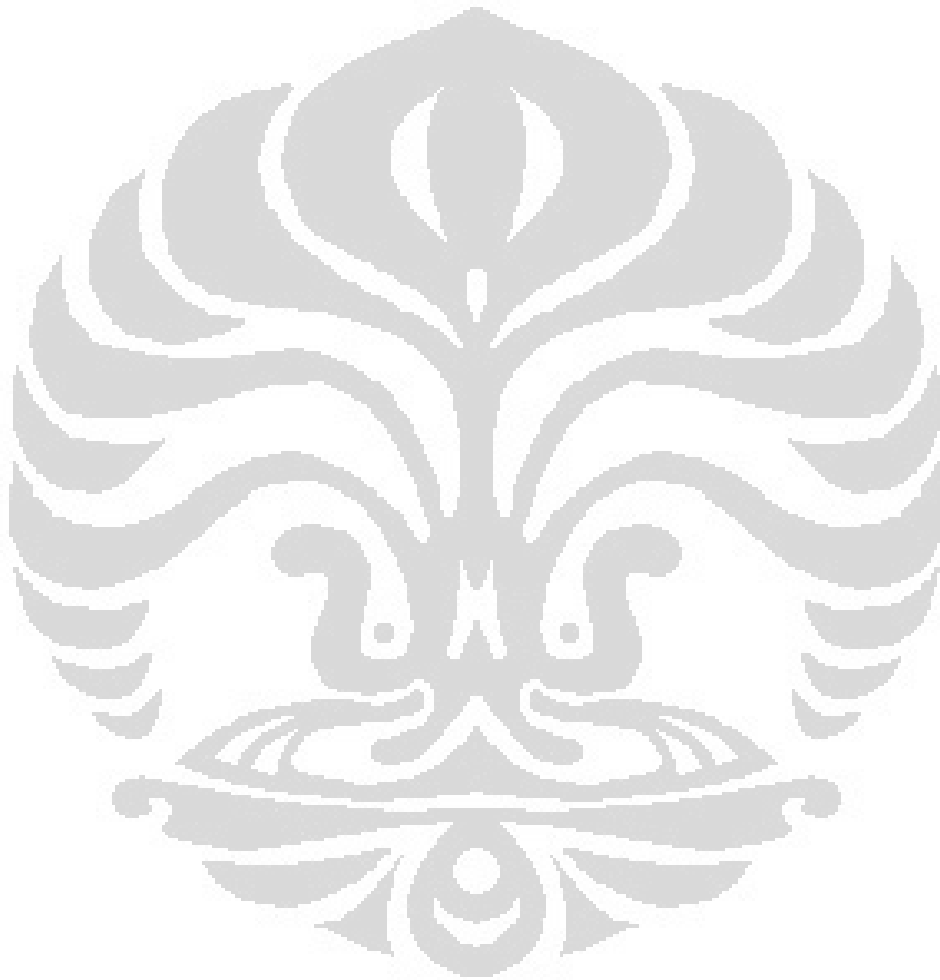
1. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Diagram Keterkaitan Masalah.....	4
1.3 Rumusan Masalah	4
1.4 Tujuan Penelitian.....	4
1.5 Ruang Lingkup Penelitian	6
1.6 Metodologi Penelitian	6
1.7 Sistematika Penulisan.....	7
2. LANDASAN TEORI.....	10
2.1 Penjadwalan.....	10
2.1.1 Model Penjadwalan.....	11
2.1.2 Pendekatan Umum Terhadap Masalah Penjadwalan	12
2.1.3 Kompleksitas dalam Penjadwalan	14
2.1.4 Penjadwalan Karyawan dengan <i>Resources</i> Terbatas	15
2.2 Masalah Optimasi.....	17
2.3 Algoritma Genetika	21
2.3.1 Kromosom.....	21
2.3.2 <i>Fitness</i>	22
2.3.3 Elitisme	23
2.2.4 Seleksi dengan Roda Lotere.....	23
2.2.5 <i>Crossover</i> atau Kawin Silang.....	24
2.2.6 Mutasi.....	26
3. PENGUMPULAN DATA.....	29
3.1 Profil Perusahaan.....	30
3.2.1 Data Proyek.....	30
3.2.1 Data Proyek.....	30
3.2.2 Data Kebutuhan Karyawan Beserta Jenis Pekerjaannya	31
3.2.3 Data Periode.....	32
3.2.4 Biaya-Biaya Penugasan	32
3.3 Model Penjadwalan Karyawan Pengawas Pembangunan Kapal	33
3.3.1 Notasi dan Asumsi	33
3.3.2 Fungsi Tujuan dan Kendala	35
4. PENGOLAHAN DATA DAN ANALISIS	38
4.1 Penyusunan Algoritma	38
4.1.1 Langkah-Langkah Penyusunan Algoritma Genetika	38
4.1.2 Verifikasi dan Validasi Program.....	43
4.2 Input Data	44
4.2.1 Data	44
4.2.2 Parameter	44
4.3 Pengolahan Dara dan Hasil	45
4.4 Analisis.....	46
4.2.1 Analisis Metode	47

4.2.2 Analisis Program.....	48
4.2.1 Analisis Hasil Optimasi	48
5. KESIMPULAN dan SARAN.....	52
DAFTAR REFERENSI	53



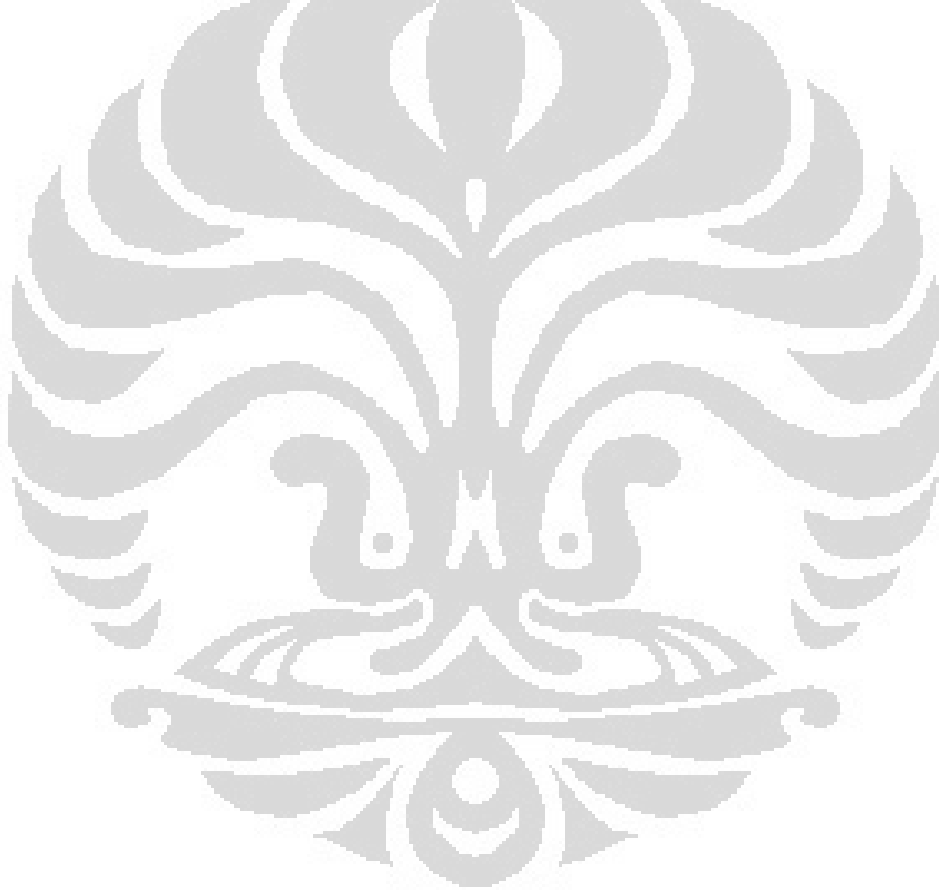
DAFTAR TABEL

Tabel 3.1 Proyek Pembangunan Kapal>Nama Kapal	30
Tabel 3.1 Data Karyawan di Pengawasan 1	31
Tabel 3.2 Data Karyawan di Pengawasan 2	31
Tabel 3.2 Data Karyawan di Perencanaan.....	31
Tabel 4.1 Skenario Parameter	43
Tabel 4.2 Perhitungan Manual Biaya	43
Tabel 4.3 Kombinasi Parameter yang Berbeda	45
Tabel 4.4 Parameter Model	45
Tabel 4.5 Hasil <i>Run</i> Program	46



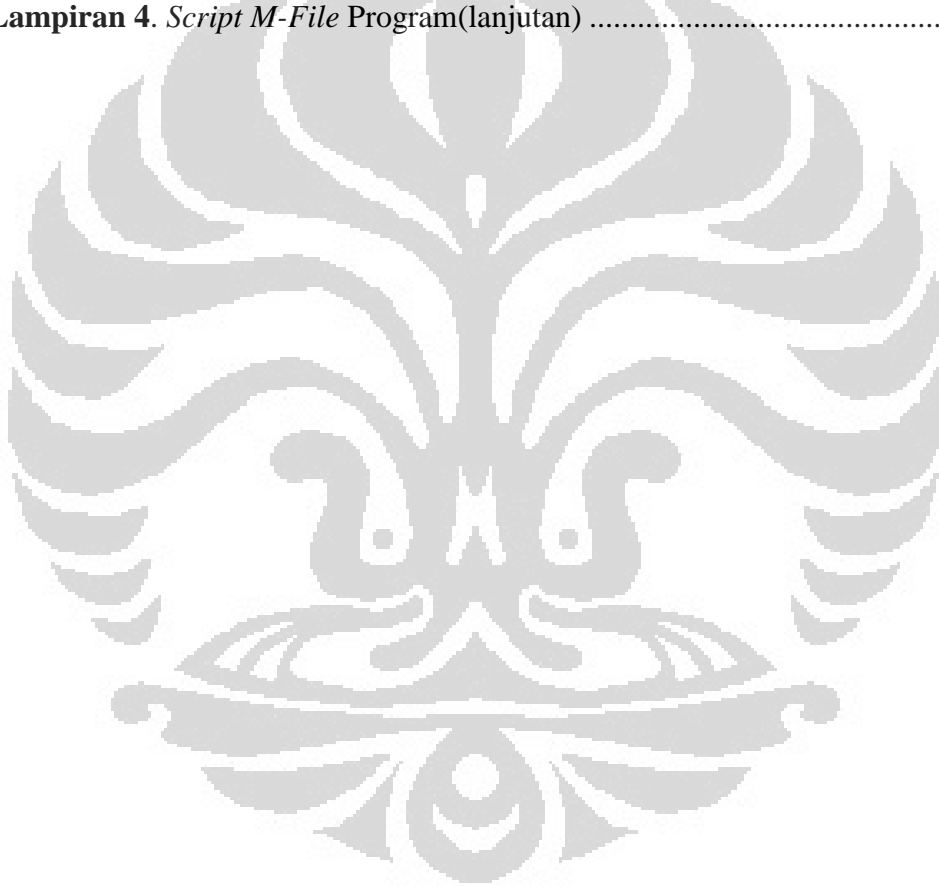
DAFTAR GAMBAR

Gambar 1.1 Diagram Keterkaitan Masalah.....	5
Gambar 1.2 Diagram Alir Metodologi Penelitian.....	8
Gambar 2.1 Diagram Masalah Penjadwalan.....	14
Gambar 2.2 Precedence, Durasi, dan Kebutuhan Sumberdaya.....	16
Gambar 2.3 Jadwal dan Kebutuhan Tenaga Kerja.....	16
Gambar 2.4 Siklus Algoritma Genetika.....	22
Gambar 2.5 Roda Lotere.....	24
Gambar 2.6 Contoh Kawin Silang.....	25
Gambar 2.7 Contoh Mutasi.....	26
Gambar 4.1 Diagram Alir Algoritma Genetika yang Digunakan.....	39
Gambar 4.1 Grafik Penurunan Nilai <i>Fitness</i>	50



DAFTAR LAMPIRAN

Lampiran 1. Penggal Waktu Pembangunan Kapal & Persebaran Karyawan.....	54
Lampiran 2. Matrik Penjadwalan Hasil Optimasi dengan 18 Karyawan tetap	55
Lampiran 3. Matrik Penjadwalan Hasil Optimasi dengan 32 Karyawan tetap	56
Lampiran 4. <i>Script M-File</i> Program(lanjutan)	57
Lampiran 4. <i>Script M-File</i> Program(lanjutan)	58
Lampiran 4. <i>Script M-File</i> Program(lanjutan)	59
Lampiran 4. <i>Script M-File</i> Program(lanjutan)	60
Lampiran 4. <i>Script M-File</i> Program(lanjutan)	61
Lampiran 4. <i>Script M-File</i> Program(lanjutan)	62
Lampiran 4. <i>Script M-File</i> Program(lanjutan)	63
Lampiran 4. <i>Script M-File</i> Program(lanjutan)	64
Lampiran 4. <i>Script M-File</i> Program(lanjutan)	65
Lampiran 4. <i>Script M-File</i> Program(lanjutan)	66



BAB 1

PENDAHULUAN

1.1 Latar Belakang Permasalahan

Secara geografis Indonesia merupakan negara kepulauan dimana dua pertiga wilayahnya merupakan wilayah perairan atau lautan, dan tersusun lebih dari tujuh belas ribu pulau-pulau yang membentang dari Sabang sampai Merauke. Kondisi geografis ini mengakibatkan distribusi harus dilakukan melalui transportasi kapal ataupun pesawat terbang. Pesawat terbang dapat digunakan sebagai alat transportasi maupun mengangkut barang-barang ringan, akan tetapi untuk beberapa jenis muatan dimana jumlah dan ukurannya yang besar akan membutuhkan kapal. Keberadaan armada kapal-kapal merupakan suatu mata rantai dari proses perpindahan muatan dari satu lokasi ke lokasi yang lainnya sebagai akibat dari kegiatan “jual-beli” antara penjual dan pembeli.

Di sektor Pertambangan, keberadaan kapal memegang peranan yang sangat penting. Hal ini terkait dengan aktivitas transportasi mulai dari hasil tambang, peralatan maupun tenaga kerja. Adapun jenis kapal yang dibutuhkan, antara lain : Oil Tankers, Barges, Liquid Carriers, Offshore Support Vessels, Survey Vessels, dsb. Banyak perusahaan – perusahaan yang membutuhkan kapal-kapal jenis tersebut, khususnya perusahaan yang bergerak di bidang pertambangan dan perminyakan seperti Pertamina.

PT Pertamina, setelah UU Migas No 22 tahun 2001 diterbitkan membuat status PT Pertamina (Perseroan) menjadi Perseroan Terbatas dari sebelumnya perusahaan negara. Perubahan mendasar ada pada peran *regulator* menjadi *player* dan bergerak di bidang energi, petrokimia, dan usaha lain yang menunjang bisnis utama PT Pertamina (Persero), baik di dalam maupun di luar negeri yang berorientasi pada mekanisme pasar, namun tender PSO (*Public Service Obligation*) yang dimenangkan oleh PT Pertamina (Persero) dari Pemerintah Indonesia selama beberapa tahun terakhir, mengharuskan PT Pertamina (Persero) harus menjamin ketersediaan bahan bakar di seluruh nusantara (www.pertamina.com).

Untuk melakukan pendistribusian bahan bakar, PT Pertamina membutuhkan kapal sebagai armada pengangkut. Akan tetapi, untuk memenuhi kebutuhan armada kapalnya, PT Pertamina harus bekerja sama dengan perusahaan galangan kapal untuk membangun kapal sebagai kebutuhan distribusi. Perusahaan - perusahaan galangan yang bekerja sama dengan PT Pertamina tersebar di beberapa tempat/negara seperti, Indonesia, China dan Korea. Meskipun PT Pertamina mengontrak perusahaan galangan kapal dalam pembuatan kapalnya, PT Pertamina tetap mengawasi pembuatan kapal ini. Tim pengawas yang mengawasi pembuatan kapal ini terdiri dari *site manager* yang dibantu oleh asisten *manager*, *surveyor hull & outfitting*, *surveyor hull & painting*, *surveyor machinery & outfitting*, *surveyor electric & outfitting*.

Dalam proses pembuatan kapal, ada beberapa tahapan yang dikerjakan; *first cutting*, *keel laying*, *launching*, dan *delivery*. Dan dalam proses pembangunan kapal tersebut, tidak semua pengawas bekerja secara bersamaan. Setiap pengawas datang/bekerja sesuai periode mereka dibutuhkan. Saat ini, proyek pembangunan kapal baru yang sedang diadakan oleh PT Pertamina bagian perkapalan sebagian besar berada di dalam negeri sedangkan sebagian lagi dikerjakan oleh perusahaan korea dan china. Untuk mengawasi pengerjaan pembangunan kapal ini, PT Pertamina harus merotasi tim pengawas ke dalam setiap proyek. Dengan lokasi proyek yang tersebar di berbagai tempat, PT Pertamina bagian kapal harus mengoptimalkan penjadwalan pengawas sehingga pengerjaan proyek bisa terlaksana dengan baik dan biaya yang dikeluarkan juga bisa dimimalisir. Penjadwalan tim pengawas merupakan salah satu aspek penjadwalan yang mempertimbangkan banyak hal, seperti jadwal pembangunan kapal, aturan pekerjaan, dan biaya perjalanan atau biaya pengawas sehingga penjadwalan bisa dibuat secara optimum.

Kegiatan penjadwalan pengawas pembangunan kapal ini dilakukan dengan mengikuti jadwal proses pembangunan kapal yang diadakan oleh perusahaan galangan kapal. Dengan melihat jadwal proses pembangunan kapal, selanjutnya dilakukan rotasi penugasan pengawas. Pengawas pembangunan kapal idealnya merupakan karyawan dari PT Pertamina, akan tetapi untuk kondisi tertentu, kebutuhan tenaga kerja tidak bisa ditutupi dengan keterbatasan karyawan

yang ada di PT Pertamina. Untuk memperlancar kegiatan pembangunan kapal, PT Pertamina melakukan antisipasi dengan mengontrak pengawas tambahan dari perusahaan lain. Hal inipun diharapkan akan memperlancar kegiatan pembangunan.

Meskipun kegiatan pembangunan kapal ini membutuhkan karyawan untuk pengawasan demi memperlancar kegiatan pembangunan, PT Pertamina tetap memperhatikan aturan dalam pekerjaan, karena permasalahan penjadwalan karyawan memperhatikan banyak aspek seperti aspek psikologi, kesehatan dan social. Di dalam Pertamina, Karyawan tetap mempunyai masa dinas maksimum 3 bulan sebelum kembali ke kantor pusat. Akan tetapi, terkadang keterbatasan jumlah karyawan, tim pengawas pembangunan kapal juga melibatkan orang lain dari departemen perencanaan pada kondisi yang dibutuhkan. Dengan mengamodasikan hal-hal tersebut di atas, timbul biaya-biaya lain seperti biaya perjalanan (*travel allowance*), biaya penginapan, dan biaya tenaga tambahan (*extra crew cost*). Dalam optimasi penjadwalan tim pengawas, biaya-biaya inilah yang harus diminimumkan. Sementara itu usaha optimasi yang dilakukan selama ini hanya berdasarkan metode coba-coba dan pengalaman di masa lalu.

Masalah penjadwalan merupakan masalah yang sulit. Bahkan menjadi lebih sulit ketika harus mempertimbangkan tujuan yang banyak (*multiple objectives*). Adanya kombinasi dalam masalah penjadwalan mengakibatkan hal ini menjadi sulit untuk diselesaikan walaupun dengan cara teknik matematika. Hal-hal yang menjadi faktor pertimbangan dalam penyusunan jadwal yaitu adanya batasan waktu pengerjaan (*completion time*), aliran waktu (*flow time*), yang merupakan urutan pengerjaan aktivitas termasuk waktu tunggu (*waiting time*), dan waktu proses (*processing time*). Penjadwalan yang optimal membuat proses operasi berjalan dengan efektif. Untuk itu diusulkan penjadwalan dengan menggunakan pendekatan algoritma genetika yang dapat menyelesaikan masalah multi-kriteria dan multi-objektif.

Optimasi merupakan suatu metode pendekatan matematis yang bertujuan untuk memperoleh tujuan yang maksimal dengan *cost* (biaya) yang minimal. Pengertian *cost* (biaya) dalam hal ini bisa berupa uang, sumber daya, tenaga kerja, atau berbagai bentuk pengorbanan lain yang perlu dikeluarkan untuk mencapai

tujuan. Terdapat berbagai macam metode yang bisa digunakan untuk melakukan optimasi, namun untuk kasus-kasus yang berskala besar dibutuhkan teknik teknik yang lebih modern seperti metode metaheuristik.

Metaheuristik menjadi alat optimasi yang telah mencatatkan sejarah sukses. Para peneliti telah menunjukkan kemampuan metode ini dalam memecahkan permasalahan *hard combinatorial* dengan skala yang cukup besar dengan waktu komputasional yang kompetitif. Banyak sekali pendekatan yang masuk kategori metaheuristik, seperti *Simulated Annealing*, Algoritma Genetika, *Cross Entropy*, *Particle Swarm Optimization*, *Ant Colony Optimization*, dll. Metaheuristik mencari solusi dengan memadukan interaksi antara prosedur pencarian local dan strategi yang lebih tinggi untuk menciptakan proses yang mampu keluar dari titik-titik *local optima* dan melakukan pencarian di ruang solusi untuk menemukan solusi global.

1.2 Diagram Keterkaitan Masalah

Diagram keterkaitan masalah merupakan sebuah ulasan yang memberikan gambaran mengenai permasalahan terkait hingga solusi yang diberikan. Berdasarkan apa yang telah dituliskan pada latar belakang, maka berikut ini adalah diagram keterkaitan masalah tersebut pada gambar 1.1.

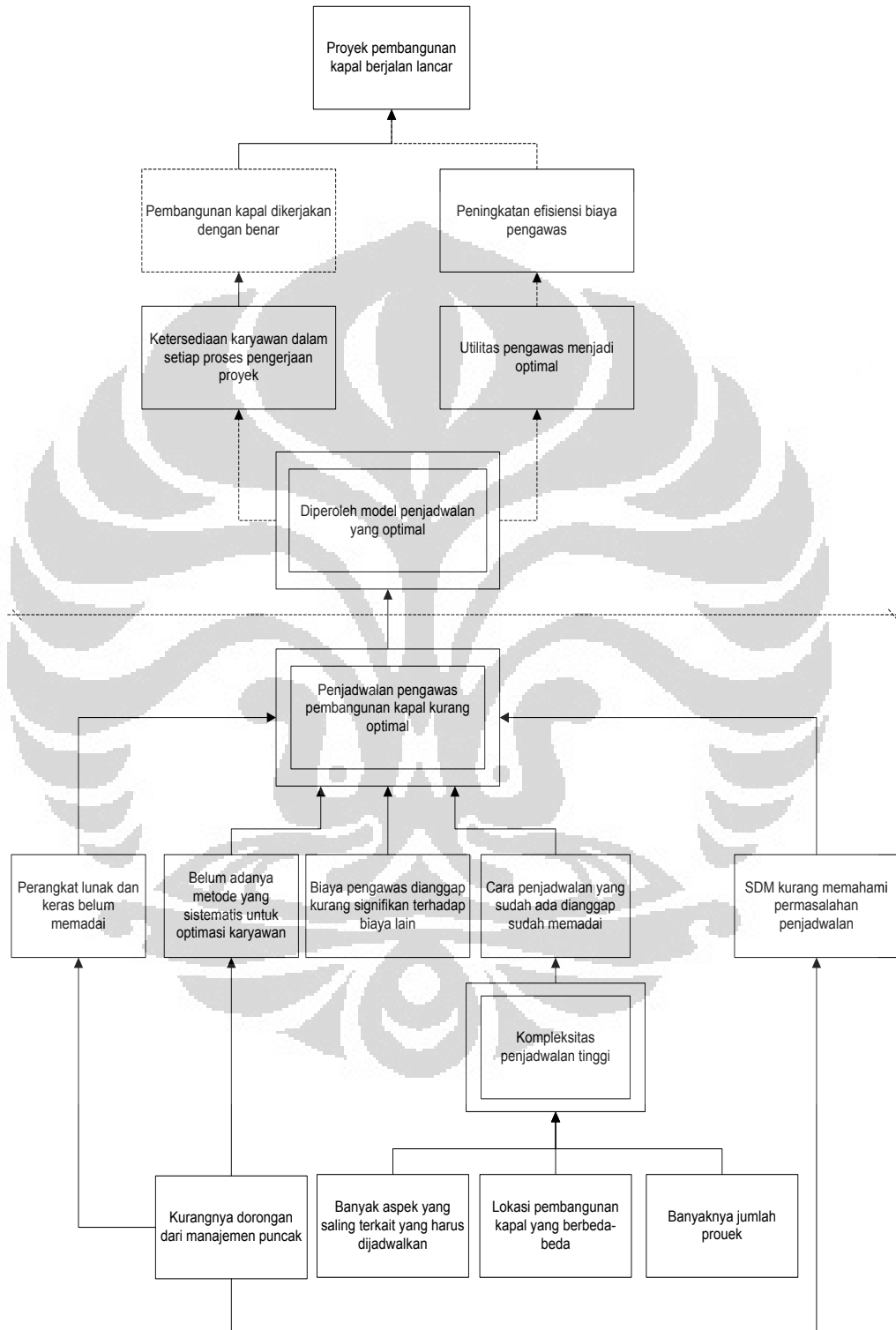
1.3 Perumusan Permasalahan

Permasalahan yang dijadikan fokus penelitian adalah optimasi suatu sistem penjadwalan pengawas pembangunan kapal di PT Pertamina dalam rangka meminimalisasi biaya. Penyelesaian sistem penjadwalan pengawas pembangunan kapal PT Pertamina dilakukan dengan metode algoritma genetika.

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan diagram keterkaitan masalah yang telah dijelaskan sebelumnya, maka tujuan dari penelitian ini adalah memperoleh model penjadwalan pengawas pembangunan kapal PT Pertamina dimana sesuai dengan

batas-batas aturan karyawan dengan meminimumkan biaya yang dikeluarkan dengan menggunakan algoritma genetika.



Gambar 1.1 Diagram Keterkaitan Permasalahan

1.5 Batasan Masalah

Dalam penelitian ini dilakukan pembatasan masalah agar pelaksanaan serta hasil yang akan diperoleh sesuai dengan tujuan pelaksanaannya. Adapun ruang lingkupnya adalah:

1. Penelitian ini dilakukan pada pengawasan proyek pembangunan yang sedang dalam proses pembangunan (16 proyek), meliputi *surveyor hull & outfitting*, *surveyor hull & painting*, *surveyor machinery & outfitting*, *surveyor electric & outfitting*.
2. Waktu proses pembangunan kapal dari awal sampai selesai dianggap sama untuk semua kapal, yaitu 14 bulan.
3. Pergerakan karyawan dibatasi dari lokasi kantor (jakarta) ke lokasi proyek (Indonesia, China, dan Korea Selatan) atau sebaliknya.
4. Biaya yang akan dimasukkan adalah biaya perjalanan dan biaya hotel.
5. Waktu perencanaan selama 36 bulan (waktu proyek pembangunan kapal).

1.6 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini secara sistematis diuraikan sebagai berikut:

1. Perumusan ide-ide topik penelitian dan mengidentifikasi permasalahan
 Pada tahap pertama, dilakukan pencarian tema-tema yang menarik untuk diangkat, baik dari pencarian pada situs-situs internet, jurnal, maupun buku dan diskusi dengan dosen pembimbing.
2. Studi literatur dasar teori penelitian
 Dilakukan studi literatur teori-teori yang menjadi dasar dalam pelaksanaan penelitian. Landasan teori yang terkait dengan penelitian ini adalah proses pembuatan kapal, *crew scheduling/staffing*, dan algoritma genetika.
3. Perumusan masalah
 Berdasarkan identifikasi masalah dan studi literatur teori, dapat dirumuskan masalah dalam penelitian ini, yaitu perlu adanya pengembangan optimasi sistem penjadwalan pengawas pembangunan kapal PT Pertamina.

4. Penentuan topik penelitian

Setelah masalah teridentifikasi, maka topik dapat ditentukan. Adapun topik dari penelitian ini adalah pembuatan model optimasi penjadwalan pengawas pembangunan kapal dengan menggunakan Algoritma Genetika. (Studi kasus: Departemen Pembangunan Kapal PT Pertamina)

5. Pengumpulan data

Pada tahap ini dilakukan identifikasi data yang dibutuhkan untuk menyelesaikan permasalahan dan kemudian dilakukann pengumpulan data-data tersebut. Pengumpulan data terdiri dari data kuantitatif dan kualitatif dari PT Pertamina.

6. Pengembangan model

Setelah pemantapan studi terhadap dasar teori dan pengumpulan data, maka data diolah dengan membuat model matematika dan diselesaikan dengan metode algoritma genetika dengan menggunakan *software* MATLAB.

7. Analisa hasil.

Pada tahap ini, dilakukan analisis terhadap penjadwalan pengawas pembangunan kapal yang telah dibuat dengan membandingkan model penjadwalan yang biasa digunakan.

8. Menarik kesimpulan

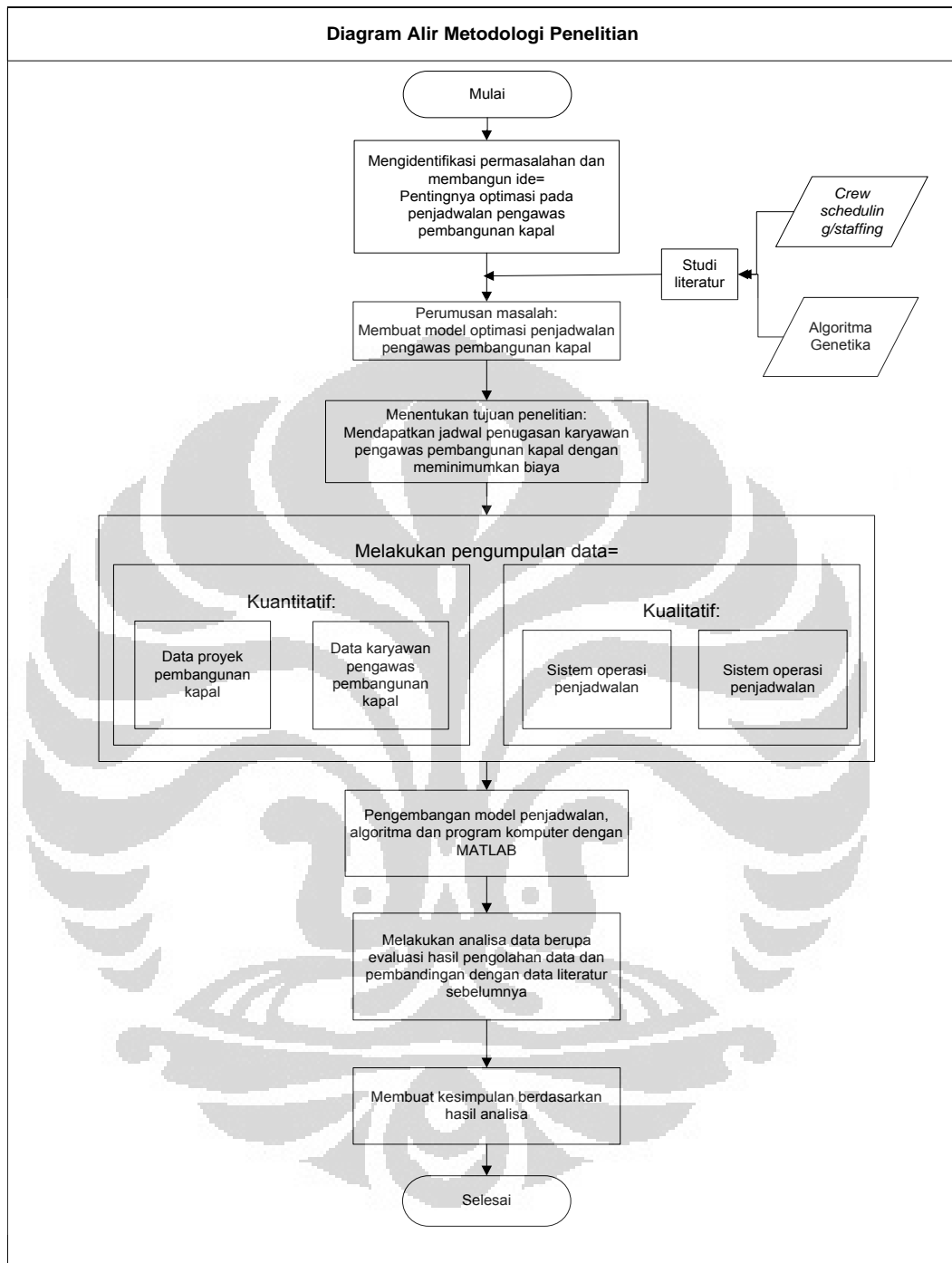
Dalam tahapan ini akan dihasilkan kesimpulan mengenai keseluruhan penelitian. Kesimpulan dari penelitian ini merupakan ringkasan dari hasil pengolahan data dan analisis yang telah dilakukan sebelumnya.

Untuk menggambarkan secara sistematis, maka dibuat diagram alir metodologi penelitian yang dapat dilihat pada gambar 1.2.

1.7 Sistematika Penulisan

Untuk mempermudah pemahaman alur penelitian ini, maka penulisan penelitian mengenai model optimasi penjadwalan pengawas pembangunan kapal ini dibuat mengikuti aturan standar baku penulisan tugas akhir mahasiswa dan disajikan dalam beberapa bab. Bab pertama adalah Pendahuluan. Pada bab

Pendahuluan, penulis menjelaskan mengenai latar belakang permasalahan yang



Gambar 1.2 Diagram Alir Metode Penelitian

menyebabkan dilakukannya penelitian ini. Selain itu, tujuan penelitian dan metodologi penelitian juga dipaparkan dalam bab ini. Penjelasan dalam bab Pendahuluan juga dilengkapi dengan diagram-diagram yang dapat menggambarkan alur permasalahan dan alur penelitian secara sistematis, yaitu diagram keterkaitan masalah dan diagram alir metodologi penelitian.

Universitas Indonesia

Pada bab kedua, penulis memaparkan dasar teori yang digunakan dalam mengerjakan penelitian ini. Landasan teori ini diperoleh dari tinjauan pustaka baik dari buku, jurnal, artikel, maupun informasi dari situs-situs di internet. Teori-teori yang dipakai meliputi teori proses pembuatan kapal, *crew pairing problem*, dan algoritma genetika.

Bab ketiga berisi pengumpulan data yang dibutuhkan penulis dalam melakukan penelitian ini. Pengumpulan data dilakukan dengan cara observasi, wawancara, dan pengumpulan dokumen perusahaan.

Bab keempat merupakan bab yang berisi pengolahan data dan analisis. Penulis menjelaskan secara terperinci langkah-langkah yang digunakan dalam pengolahan data sampai diperoleh hasil yang diharapkan. Pengolahan data pada penelitian ini dilakukan dengan menggunakan algoritma genetika. Setelah melakukan pengolahan data, dilakukan analisa hasil pengolahan data.

Setelah pengolahan data dan analisa dilakukan, maka dapat ditarik suatu kesimpulan dari pengerjaan penelitian secara keseluruhan. Kesimpulan ini merupakan jawaban dari tujuan penelitian. Kesimpulan ini ditulis pada bab yang kelima.

BAB 2 LANDASAN TEORI

2.1 Penjadwalan

Penjadwalan adalah sebuah proses pengambilan keputusan yang digunakan sebagai acuan dasar pada berbagai industri . Hal ini berkaitan dengan alokasi *resources* terhadap pekerjaan dalam suatu periode waktu yang diberikan dan tujuannya adalah untuk mengoptimalkan satu tujuan atau lebih.

Resources dan pekerjaan dalam sebuah organisasi dapat dilihat dari berbagai bentuk. *Resources* dapat berupa mesin dalam ruang kerja, landasan kapal terbang di *airport*, karyawan dalam tempat konstruksi, dan lain sebagainya. Sedangkan pekerjaan dapat berupa jenis-jenis operasi dalam sebuah proses produksi, *take-offs and landings* pada sebuah *airport*, tahap/langkah-langkah dalam sebuah proyek konstruksi. Setiap pekerjaan mempunyai level prioritas tertentu, dan waktu memulai dan berakhirnya sebuah pekerjaan. Tujuan juga dapat dilihat dalam berbagai bentuk yang berbeda. Sebuah tujuan dapat berupa meminimalkan penyelesaian sebuah pekerjaan atau tujuan lain dapat berupa meminimalkan jumlah pekerjaan diselesaikan setelah jatuh tempo (Michael L. Pinedo, 2010).

Masalah penjadwalan tenaga kerja secara umum adalah masalah dalam penentuan berapa banyak pekerja yang harus ditempatkan terhadap setiap periode perencanaan dari waktu kerja pada sebuah organisasi. Karena sumber daya manusia adalah komponen yang sangat penting diantara semua organisasi baik di industri manufaktur ataupun jasa, menjadwalkan tenaga kerja adalah masalah yang umum bagi semua organisasi (Evrin Didem Gunes, 1999). Kriteria untuk penjadwalan tenaga kerja didasarkan pada efisiensi penggunaan sumber daya tenaga kerja/manusia dengan tetap memperhatikan aturan/perjanjian dalam pekerjaan.

Masalah penjadwalan merupakan masalah yang sulit. Bahkan menjadi lebih sulit ketika harus mempertimbangkan tujuan yang banyak (*multiple objectives*). Adanya kombinasi dalam masalah penjadwalan mengakibatkan hal ini menjadi sulit untuk diselesaikan walaupun dengan cara teknik matematika.

Hal-hal yang menjadi faktor pertimbangan dalam penyusunan jadwal yaitu adanya batasan waktu pengerjaan (*completion time*), aliran waktu (*flow time*), yang merupakan urutan pengerjaan aktivitas termasuk waktu tunggu (*waiting time*), dan waktu proses (*processing time*). Penjadwalan yang tepat akan mencapai tujuan yang optimal dengan memenuhi semua kendala (Yanti Nitra Sianipar, 2010).

2.1.1 Model Penjadwalan

Penyelesaian masalah penjadwalan dapat dilakukan dengan memodelkan penjadwalan dalam dunia nyata terlebih dahulu sehingga akan mempermudah penyelesaiannya.

Ada beberapa syarat yang harus dipertimbangkan dalam proses perencanaan tenaga kerja yang juga akan menjadi faktor-faktor dalam menentukan model penjadwalan tenaga kerja yang akan dijabarkan sebagai berikut (Evrin Didem Gunes, 1999) :

1. Sifat kebutuhan seseorang apakah konstan, berubah-ubah seiring waktu,
2. Standar jadwal pekerjaan yang digunakan. Yang paling umum adalah standar 5 hari, 10 jam minggu kerja. Contoh yang lain adalah : “flexitime” atau mengizinkan pekerja untuk memilih waktu mulai dan berhenti bekerja selama dia bekerja dalam “core hours” yang biasanya dari jam 10 pagi sampai jam 3 sore.
3. Jumlah dan bentuk shift kerja sehari. Seperti siang, sore, dan malam dengan alternative waktu mulai dan berhenti bekerja, dan juga beristirahat diberikan.
4. Aturan pekerjaan yang berhubungan dengan bentuk hari kerja/cuti para pekerja. Jadwal cuti harus berurutan atau tidak, dan rotasi shift dibolehkan atau tidak.
5. Kelonggaran bekerja maksimum yang diperbolehkan.
6. Atribut tenaga kerja. Ada banyak perbedaan keahlian pekerja, selain itu pekerja part time/subkontrak juga bisa dipertimbangkan.

Dengan mempertimbangkan semua faktor-faktor ini, yang menjadi tugas adalah menemukan jadwal yang memungkinkan dan berbiaya murah. Disamping

faktor-faktor tersebut, kesulitan dalam masalah penjadwalan pekerja adalah berurusan dengan keberadaan manusia. Jadi “biaya” sebuah penjadwalan berbeda dengan mendapatkan total gaji. Selain itu, semangat/moril tergantung seberapa baik pemilihan penjadwalan sesuai dengan pilihan perseorangan, yang akan mempengaruhi produktivitas. Lebih jauh lagi, biaya *training*, *overtime*, faktor kesehatan dan keselamatan harus diperhitungkan ketika mempersiapkan jadwal pekerjaan. Karakteristik dari jadwal pekerjaan yang harus dipertimbangkan ketika membuat keputusan penjadwalan adalah:

- *Coverage* : Keberadaan jumlah pekerja minimum yang dibutuhkan setiap periode waktu.
- *Quality* : Ukuran dari jadwal yang sesuai atau diinginkan oleh pekerja yang akan bekerja.
- *Stability* : Ukuran sejauh mana pekerja dapat mengetahui waktu bekerja dan berhenti di kemudian hari
- *Flexibility* : Kemampuan menangani perubahan, seperti perubahan dari “full time” ke “part time”, dan munculnya persyaratan baru bagi para pekerja.
- *Fairness* : Ukuran sejauh mana setiap pekerja dipengaruhi oleh waktu yang sama dalam hal perubahan yang tidak diinginkan.

Langkah-langkah umum yang digunakan dalam menganalisa dan menyelesaikan penjadwalan tenaga kerja adalah sebagai berikut:

1. Menentukan jumlah pekerjaan yang harus diselesaikan.
2. Menentukan susunan pegawai yang dibutuhkan untuk melaksanakan pekerjaan (untuk setiap periode waktu).
3. Menentukan ketersediaan personalia.
4. Menyesuaikan personalia untuk kebutuhan staff.

2.1.2 Pendekatan Umum Terhadap Masalah Penjadwalan

Penjadwalan tenaga kerja adalah persoalan riset yang sudah dipelajari secara luas. Ada banyak jenis masalah, seperti *air crew scheduling*, *audit scheduling*, *nurse scheduling*, *construction crew scheduling*, dan lain-lain. Masalah penjadwalan tenaga kerja dapat diklasifikasikan sebagai berikut:

1. Sifat permintaan dan stabilitas siklik tenaga kerja dibandingkan asiklik
2. Selisih waktu kerja yang dijadwalkan shift, hari kerja (hari libur) dan penjadwalan tur.

Penjadwalan siklis mengasumsikan pola permintaan yang konstan dalam periode waktu yang spesifik. Contohnya, jika permintaan dapat bervariasi diantara hari dalam seminggu dan pola mingguan adalah konstan, maka penjadwalan siklis dapat digunakan. Sekali penjadwalan ditentukan setiap karyawan secara berulang kali melalui pola mingguan set yang sama. Penjadwalan siklik mengasumsikan sejumlah tenaga konstan yang semuanya tersedia setiap saat. Untuk kasus asiklik, masalah harus diselesaikan setiap kali pola permintaan dan perubahan ketersediaan karyawan, dalam perencanaan yang ditentukan di lain waktu.

Masalah penjadwalan shift diselesaikan dengan mengembangkan sebuah penjadwalan kerja harian. Mereka akan muncul ketika dibutuhkan untuk menentukan shift karyawan mana yang harus ditempatkan untuk setiap hari kerja, khususnya ketika shift dalam *overlapping*. Dalam perusahaan jasa, ini hanya merupakan masalah yang diselesaikan karena hari kerja sudah tetap dengan kebijakan manajemen.

Dalam Solusi penjadwalan minggu kerja terdapat penjadwalan hari libur, yang menentukan hari-hari setiap karyawan harus diistirahatkan. Masalah ini terjadi ketika waktu minggu kerja (*work week*) seseorang berbeda dengan minggu bisnis (*business week*). Contohnya, 5 hari kerja seminggu dan operasi bisnis 6-7 hari, seperti perawat, retail salespersons. Penjadwalan hari libur mengasumsikan shift tunggal setiap hari yang tidak sesuai untuk organisasi yang beroperasi 24 jam sehari.

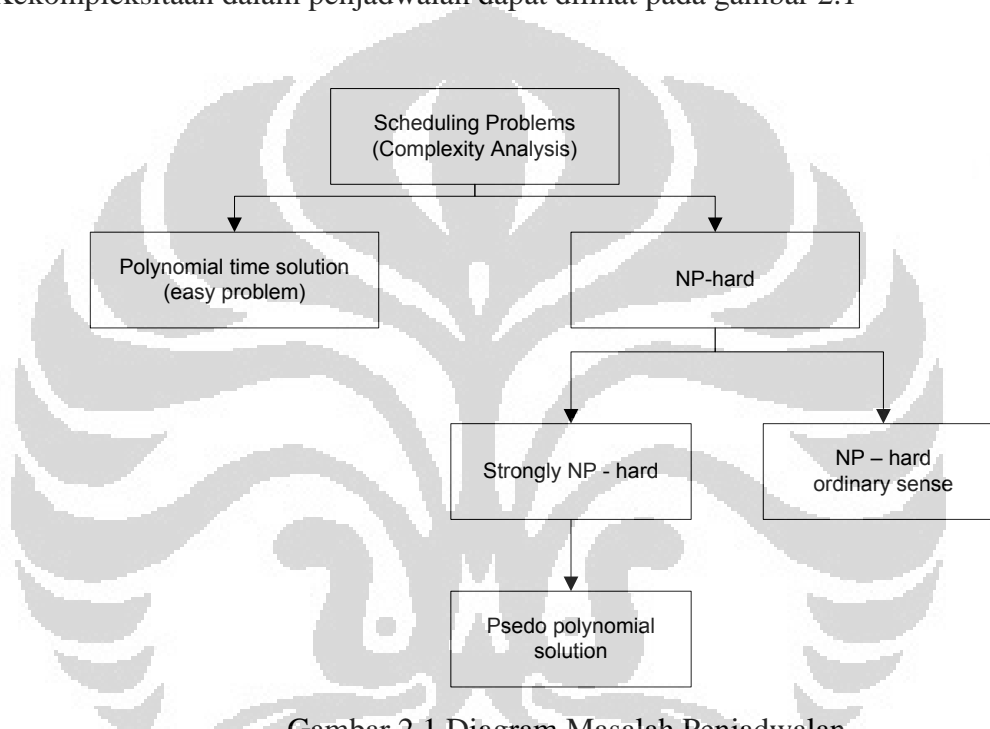
Ketika kedua shift harian untuk setiap hari kerja dan hari libur dibutuhkan, dua masalah yang harus diselesaikan secara bersamaan, membawa hal ini ke dalam masalah *tour scheduling* secara umum. *Tour scheduling* melibatkan penentuan hari kerja dan tidak kerja selama penggal waktu yang disesuaikan dengan waktu mulai dan berhenti bekerja bagi setiap karyawan.

Tour schedules untuk operasi yang kontinyu kebanyakan membutuhkan karyawan yang berotasi berdasarkan shift. Shift kerja mempunyai banyak efek negatif terhadap kesehatan karyawan dan kehidupan social, namun hal ini adalah

masalah waktu tidak bisa dihindari. Dalam kasus ini, rotasi yang cepat antara shift sangat disarankan.

2.1.3 Kompleksitas Dalam Penjadwalan

Penjadwalan menjadi semakin sulit ketika memiliki kendala-kendala yang banyak dan cukup rumit. Disamping itu, apabila jumlah populasi yang harus dijadwalkan semakin banyak akan memicu kekompleksitan masalah penjadwalan. Kekompleksitan dalam penjadwalan dapat dilihat pada gambar 2.1



Gambar 2.1 Diagram Masalah Penjadwalan

(Sumber: Uwe Schwiegelshohn, CEI University Dortmund 2004)

Terdapat dua tingkat kesulitan dalam melakukan penjadwalan, yaitu:

1. Mudah (*polynomial time complexity*)

Penjadwalan dikatakan mudah apabila sebuah algoritma dapat digunakan untuk

mengoptimalkan masalah dengan kompleksitas waktu $O((n \log(\max P_j))^k)$ untuk nilai k tertentu.

2. NP – *hard*, terdiri dari dua jenis:

- a. NP – *hard* dalam tingkat kesulitan yang luar biasa (*pseudo polynomial time complexity*). Masalah penjadwalan yang tidak dapat dioptimalkan

dengan sebuah algoritma dengan kompleksitas *polynomial* waktu namun dengan sebuah algoritma dengan kompleksitas $O((n \max P_j)^k)$.

b. NP – *hard* dalam strong sense

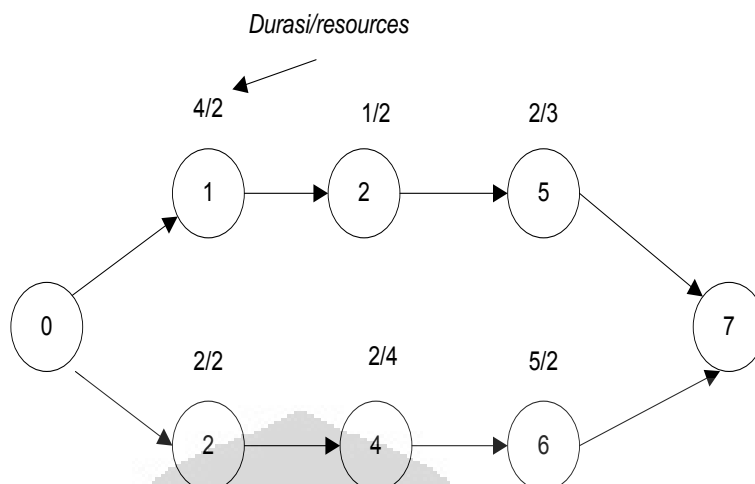
Masalah tidak dapat dioptimalkan dengan sebuah *pseudo polynomial complexity*.

Oleh karena tingkat kesulitannya, maka masalah penjadwalan diselesaikan dengan menggunakan algoritma. Algoritma adalah kumpulan perintah untuk penyelesaian suatu masalah dan dapat diterjemahkan secara bertahap dari awal hingga akhir yang bersifat diskrit dan jelas serta dapat dijalankan secara mekanik.

2.1.4 Penjadwalan Karyawan Dengan *Resource* Terbatas

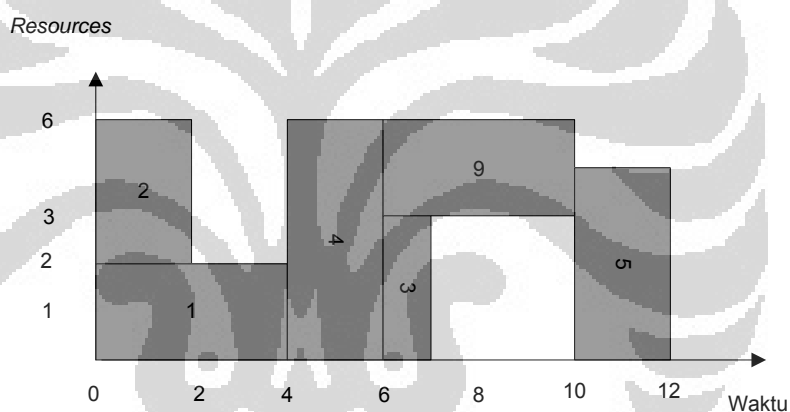
Pada umumnya untuk menyelesaikan masalah penjadwalan karyawan dengan *resource* terbatas digunakan pendekatan *critical path method* (CPM). Untuk menerapkan CPM harus diketahui data aktivitas (pekerjaan), durasi dan hubungan ketergantungan antar pekerjaan. Dengan CPM akan diketahui berapa umur proyek (*makespan*) terpendek. Namun pemakaian CPM tidak lagi layak jika ada konstrain tambahan berupa ketersediaan *resource* (Budi Santosa 2011). Jika terjadi keterbatasan sumber daya, untuk jadwal yang sudah diperoleh dengan CPM, perlu dilakukan lagi pengaturan jadwal menyesuaikan ketersediaan *resource* yang dimiliki atau perlu ditambahkan sumber daya agar jadwal yang didapatkan bisa berlaku. Cara lain yang bisa digunakan untuk melakukan penjadwalan karyawan pembangunan kapal ini adalah melalui pendekatan metaheuristik. Ini bisa dilakukan karena sebenarnya menentukan jadwal urutan pengerjaan aktivitas adalah permasalahan optimasi kombinatorial yang cukup sulit jika jumlah aktivitasnya membesar.

Sebagai gambaran bagaimana permasalahan ini muncul akan diilustrasikan sebagai berikut. Misalkan ada 6 aktivitas yang harus dijadwalkan dengan hubungan ketergantungan, kebutuhan *resource* dan durasi tiap aktivitas seperti ditunjukkan gambar 2.2, akan didapatkan jadwal dan pemakaian *resource* yang layak seperti gambar 2.3. Dari gambar tersebut nampak bahwa kegiatan 3 bisa dimulai setelah kegiatan 1 selesai. Kegiatan 4 bisa dilakukan setelah kegiatan 2 selesai. Sementara itu kegiatan 1 atau 2 tidak saling bergantung.



Gambar 2.2 Precedence, durasi, dan kebutuhan sumberdaya

Dalam penjadwalan, terdapat kendala. Kendala terjadi akibat adanya interaksi antara beberapa entitas dalam perancangan jadwal.



Gambar 2.3 Jadwal dan kebutuhan tenaga kerja

Model penjadwalan ini akan terdiri dari beberapa entitas seperti akan dijelaskan berikut:

1. Penggal Waktu jadwal pembangunan kapal dan persebaran karyawan

Dalam memodelkan rancangan terdapat penggal waktu untuk mengerjakan tahap demi tahap pembangunan kapal seperti *first cutting*, *keel laying*, *launching*, dan *delivery*.

2. Pengawas

Penjadwalan yang dilakukan dalam penelitian ini adalah penjadwalan tenaga pengawas pembangunan kapal. Pengawas terdiri dari site manager, asisten pengawasan, pengawas hull, pengecatan, mesin dan elektrikal.

Setiap pengawas tidak berada di lokasi secara bersama, tergantung dari pengalokasian waktu kapan mereka dibutuhkan.

3. Lokasi Proyek

Pembangunan kapal dilakukan di beberapa tempat yang berbeda sehingga pengalokasian proyek harus disesuaikan dengan jumlah pengawas yang tersedia.

4. Peraturan Kerja

Untuk mengatasi masalah psikologis dan kehidupan sosial, setiap karyawan diberlakukan peraturan bahwa mereka hanya boleh berada maksimal 3 bulan di lokasi proyek dan harus kembali ke kantor dan baru boleh kembali ke lokasi proyek setelah 1 bulan.

Dalam penjadwalan pengawas pembangunan kapal, terdapat beberapa kendala yang timbul akibat interaksi antar entitas model, yaitu sebagai berikut:

1. Tidak ada karyawan yang bertugas di lokasi yang berbeda dalam waktu yang bersamaan.
2. Jumlah karyawan pengawas pembangunan kapal harus cukup untuk memenuhi semua proyek.
3. Seorang Pengawas pembangunan kapal hanya boleh berada di lokasi proyek maksimal 3 bulan, setelah itu harus kembali ke Jakarta.
4. Seorang karyawan paling cepat 1 bulan harus kembali ke lokasi proyek setelah berada di Jakarta.
5. Pengawas harus ada di kantor Pertamina sekurang-kurangnya 2 orang karyawan.

2.2 Masalah Optimasi

Optimasi memegang peran 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 computer.

Pengertian optimasi bisa dijelaskan sebagai suatu kumpulan formula matematis dan metoda numeric untuk menemukan dan mengidentifikasi kandidat terbaik dari sekumpulan alternative tanpa harus secara eksplisit menghitung dan mengevaluasi semua alternatif yang mungkin. Masalah optimasi biasanya dinyatakan dalam bentuk fungsi matematik. Optimasi adalah proses memaksimalkan atau meminimasi suatu fungsi tujuan dengan tetap memperhatikan pembatas yang ada. Suatu fungsi didefinisikan sebagai suatu aturan yang menugaskan setiap pilihan nilai x dengan satu nilai unik $y = f(x)$. Dalam hal ini x adalah variabel independent dan y adalah variabel dependent.

Permasalahan optimasi bisa dibagi menurut beberapa kategori:

1. Optimasi tanpa pembatas (*unconstrained optimization*)

Jika suatu fungsi f berlaku untuk $S = R$, maka fungsi disini adalah fungsi satu variabel tanpa pembatas atau *unconstrained function*. Misalkan terdapat fungsi sebagai berikut

$$f(x) = x^3 - 2x^2 - x + 3, x \in R,$$

adalah fungsi tanpa pembatas (*unconstrained function*) dengan satu variabel. Sedangkan

$$f(x) = x_1^2 - x_2^2 - x_1 + 3, x \in R,$$

adalah fungsi tanpa pembatas (*unconstrained function*) dengan dua variabel. Secara umum formulasi fungsi tanpa pembatas dinyatakan sebagai

$$\text{Minimasi } f(x), x \in R \quad (2.1)$$

2. Optimasi dengan pembatas (*constrained optimization*)

Kalau S adalah subset dari R , maka akan didapat fungsi yang didefinisikan dalam daerah yang terbatas atau *constrained region*. Misalkan kita punya masalah optimasi dengan satu pembatas berupa satu pembatas sebagai berikut

$$\text{Min } f(x) \quad (2.2)$$

Subject to

$$h(x) = 0$$

$$g(x) \leq 0$$

Pembatas $h(x) = 0$ dan $g(x) \leq 0$ menyatakan bahwa S adalah bagian atau subset dari R ($S \subset R$). Daerah dimana x memenuhi pembatas $h(x)$ maupun $g(x)$ disebut dengan daerah feasible atau *feasible region*.

Masalah optimasi juga bisa dilihat dari nilai variabelnya. Pengelompokan masalah optimasi berdasarkan nilai variabel adalah:

1. Optimasi dengan variabel kontinu, masalah optimasi dengan nilai x bisa berapa saja dalam daerah feasibel. Masalah program linier atau program kuadratik adalah contoh optimasi kontinu.
2. Optimasi diskrit, adalah masalah optimasi dengan nilai solusi terbatas pada nilai-nilai tertentu yang biasanya bilangan bulat. Secara umum penyelesaian masalah optimasi diskrit lebih sulit dibanding optimasi kontinu. Masalah integer programming adalah contoh optimasi diskrit. Dalam keadaan dimana tidak ada informasi mengenai algoritma *polynomial* yang eksak (waktu komputasi proporsional terhadap N^n , dengan N adalah jumlah parameter yang dicari, dan n suatu konstanta integer) dikatakan masalah tersebut sebagai *NP – hard*. Dalam kategori *NP – hard* tidak ada nilai n sehingga waktu komputasi dibatasi oleh suatu *polynomial* dengan pangkat n .

Secara spesifik problem optimasi bisa dikategorikan berdasarkan variabel keputusan, fungsi objektif dan konstrain sebagai:

- *Linear Programming* (LP) : baik fungsi objektif dan konstrainnya linier, variabel keputusannya bersifat kontinu. Sebagai contoh

$$\text{Minimasi } f(x) = 2x_1 + 3x_2 \quad (2.3)$$

Subject to

$$x_1 + 3x_2 \geq 10$$

$$2x_1 + 5x_2 \geq 18 \leq 0$$

$$x_1, x_2 \geq 0$$

- *Nonlinear Programming* (NLP) : fungsi objektif dan atau konstrainnya tidak linier. Nilai variabel keputusannya kontinu. Sebagai contoh

$$\text{Minimasi } f(x) = 2x_1^2 + 3x_2 \quad (2.4)$$

Universitas Indonesia

Subject to

$$x_1 x_2 \geq 10$$

$$2x_2^2 + 5x_2 \geq 18$$

- *Integer Programming* (IP): Variabel keputusannya bernilai integer. Sebagai contoh:

$$\text{Maksimasi } f(x) = 2x_1 + 3x_2 \quad (2.4)$$

Subject to

$$4x_1 + 3x_2 \leq 23$$

$$4x_1 + 10x_2 \leq 44$$

$$x_1, x_2 \geq 0 \text{ dan integer}$$

- *Mixed Integer Linier Programming* (MILP) : fungsi objektif dan konstrain linier. Variabel keputusannya bernilai campuran integer dan riil. Sebagai contoh

$$\text{Maksimasi } 6x_1 + 4x_2 + 7x_3 - 200y_1 + 150y_2 - 100y_3 \quad (2.6)$$

Subject to

$$3x_1 + 2x_2 + 6x_3 \leq 150$$

$$4x_1 + 3x_2 + 4x_3 \leq 160$$

$$x_1 - 500y_1 \leq 0$$

$$x_2 - 500y_1 \leq 0$$

$$x_3 - 500y_1 \leq 0$$

$$x_1, x_2, x_3 \geq 0, \text{ integer}$$

$$y_1, y_2, y_3 = 0 \text{ atau } 1$$

- *Mixed Integer Nonlinier Programming* (MILP) : nonlinear programming problem dengan variabel keputusan integer dan kontinyus. Sebagai contoh

$$\text{Maksimasi } 6x_1^2 + 4x_2^2 + 7x_3 - 200y_1 + 150y_2 - 100y_3 \quad (2.7)$$

Subject to

$$3x_1 + 2x_2 + 6x_3 \leq 150$$

$$4x_1 + 3x_2 + 4x_3 \leq 160$$

$$x_1 - 500y_1 \leq 0$$

$$x_2 - 500y_1 \leq 0$$

$$x_3 - 500y_1 \leq 0$$

$$x_1, x_2, x_3 \geq 0, \text{ integer}$$

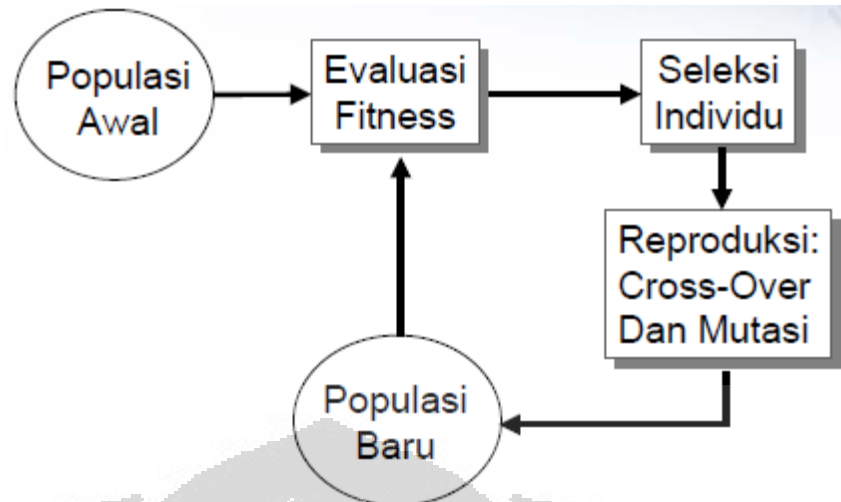
$$y_1, y_2, y_3 = 0 \text{ atau } 1$$

2.3 Algoritma Genetika

Algoritma genetika masuk dalam kelompok *Evolutionary Algorithm*. GA didasarkan pada prinsip-prinsip genetika dan seleksi alam. Elemen-elemen dasar dari genetika alam adalah: reproduksi, *crossover*, dan mutasi. GA termasuk temuan dalam bidang optimasi, dimana suatu algoritma diciptakan dengan meniru mekanisme evolusi dalam perkembangan makhluk hidup. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya (“Hanya individu-individu yang kuat yang mampu bertahan.”) Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan. Dalam algoritma genetika ini, proses perkembang-biakan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar berpikir: “Bagaimana mendapatkan keturunan yang lebih baik”. Dalam GA prosedur pencarian hanya didasarkan pada nilai fungsi tujuan, tidak ada pemakaian gradient atau teknik kalkulus. Gambar 2.4 adalah siklus Algoritma Genetika:

2.3.1 Kromosom

Dalam GA, kromosom merupakan bagian penting dari algoritma. Satu kromosom atau individu mewakili satu vector solusi. Kadang kita bisa langsung menggunakan vektor solusi ini dalam implementasi GA. Atau kadang bisa juga dilakukan *encoding* atau pengkodean. Pengkodean dilakukan untuk mewakili suatu n solusi dengan menggunakan bilangan biner. Ini tergantung pada permasa-



Gambar 2.4 Siklus Algoritma Genetika

han optimasi yang dihadapi. Dalam permasalahan optimasi fungsi, seringkali n kontinyus akan diwakili dengan bilangan biner. Dengan bilangan

Beberapa istilah dipakai dalam GA dijelaskan sebagai berikut:

bilangan biner, langkah – langkah kawin silang atau mutasi akan lebih banyak variasinya, sebaliknya bisa juga dilakukan penghitungan tanpa melalui proses *encoding* . Jadi solusi memang dalam bentuk kontinyus dan proses-proses seleksi, kawin silang dan mutasi dilakukan dengan menggunakan bilangan kontinyus. Pada akhir algoritma GA, jika dilakukukan pengkodean, maka akan dilakukan proses mengembalikan ke nilai kontinyus asalnya, yang sering disebut *decoding*. Dalam GA akan dibangkitkan populasi sebagai sebagai kumpulan dari kromosom, dimana masing-masing kromosom mewakili satu vektor solusi. Dengan dibangkitkannya populasi ini, maka akan tersedia banyak pilihan solusi.

2.3.2 Fitness

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

$$F(x) = \frac{1}{1 + f(x)}$$

dimana $f(x)$ adalah fungsi tujuan dari masalah yang diselesaikan.

Untuk minimasi, jika didapatkan $f(x)$ yang kecil maka nilai *fitness*nya besar.

Sebaliknya untuk kasus maksimasi, fungsi *fitness*nya bisa menggunakan nilai $f(x)$ sendiri, jadi $F(x) = f(x)$.

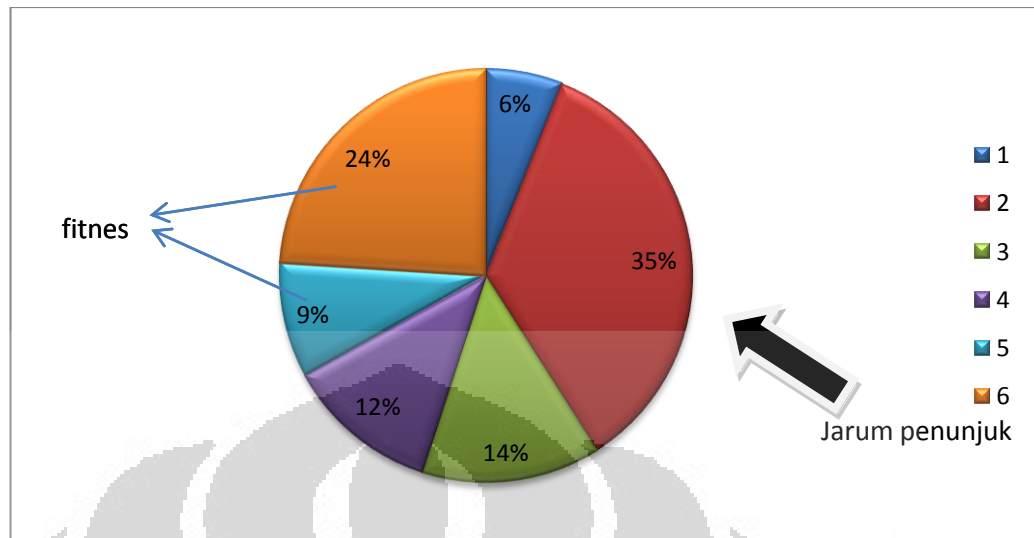
Setelah setiap solusi dievaluasi dengan fungsi *fitness*, perlu dilakukan proses seleksi terhadap kromosom. Proses seleksi dilakukan untuk memilih diantara kromosom anggota populasi ini, mana yang bisa menjadi induk (*parent*) atau melakukan identifikasi diantara populasi ini, kromosom yang akan menjadi anggota populasi berikutnya. Ada beberapa cara melakukan seleksi ini. Sebagian anggota populasi bisa dipilih untuk proses reproduksi. Cara yang umum digunakan adalah melakukan *roulette wheel selection* atau roda lotere.

2.3.3 Elitisme

Konsep elitism (*elitism*) dalam GA 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. Terbukti bahwa penggunaan operator elitisme ini telah terbukti memiliki pengaruh yang sangat penting saat menggunakan GA untuk menyelesaikan persoalan optimasi dengan tujuan tunggal.

2.3.4 Seleksi dengan Roda Lotere

Untuk memahami proses seleksi dengan roda lotere bisa dilihat pada gambar 2.5. Sebuah lingkaran lotere yang daerahnya dibagi – bagi sebanyak jumlah individu atau solusi yang dibangkitkan. Dalam contoh ini ada 6 individu. Dalam roda lotere setiap individu mempunyai kesempatan yang sama untuk terpilih. Setiap area dalam lingkaran ini menunjukkan peluang setiap solusi akan dibangkitkan bilangan random. Misalkan untuk individu 1, dibangkitkan bilangan random (dalam contoh ini bilangan random dikalikan dengan fungsi *fitness* (tf)) berada di area mana dari lingkaran lotere tersebut. Misalkan nilai $r \times tf$ lebih kecil dari akumulasi *fitness* 1 + *fitness* 2, maka individu 2 yang akan dipilih untuk menggantikan individu 1.



Gambar 2.5 Roda Lotere

Setiap kali roda diputar sampai berhenti, jarum penunjuk akan tertuju pada individu tertentu. Individu dengan nilai *fitness* paling besar akan punya kesempatan terpilih paling besar untuk menjadi induk. Dalam contoh ini, jika roda diputar sebanyak jumlah individu (6), peluang individu 1 dengan nilai *fitness* 35 akan terpilih paling besar, disbanding individu dengan *fitness* yang lebih rendah. Dalam optimasi, *fitness* ini mewakili nilai fungsi tujuan. Dalam kasus minimasi fungsi, semakin kecil nilai fungsi tujuan, semakin besar nilai *fitness*. Proses seleksi induk ini akan diulang sejumlah yang diinginkan.

Dari contoh ini, kelihatan bahwa solusi ke-2 punya peluang paling tinggi untuk diseleksi menjadi induk karena kromosom ke-2 ini mempunyai nilai *fitness* besar.

2.3.5 Crossover atau Kawin Silang

Cross Over (Pindah Silang) merupakan salah satu operator dalam algoritma genetika yang melibatkan dua induk untuk menghasilkan keturunan yang baru. Ada bermacam-macam teknik *crossover* yang bisa digunakan dalam GA. Tetapi yang akan dibahas disini adalah *crossover* sederhana dan *crossover aritmetika*. Dalam *crossover* 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 random diskrit yang bernilai

antara 1 dan panjang vector x , maka U dan V yang mewakili keturunan C_1 dan C_2 didefinisikan sebagai:

$$u_i = \begin{cases} x_i & i < r \\ y_i & \text{sebaliknya} \end{cases} \quad (2.7)$$

$$v_i = \begin{cases} x_i & i < r \\ y_i & \text{sebaliknya} \end{cases} \quad (2.8)$$

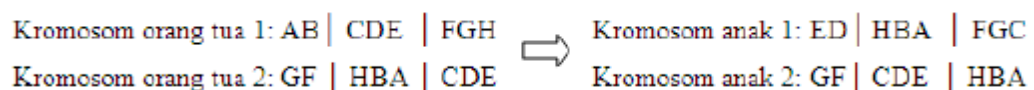
Nilai bilangan random r sering disebut dengan titik potong. Jadi mulai gen ke-berapa kawin silang akan dilakukan. Misalkan dari 8 gen didapat $r = 3$, maka mulai gen ke 4, gen dari kromosom X akan ditukarkan dengan gen dari kromosom Y. Sedangkan dalam *crossover* aritmatik, keturunan dihasilkan dengan melakukan kombinasi linier dari vektor induk. Secara matematis bisa dituliskan

$$C_1 = \lambda_1 X + \lambda_2 Y \quad (2.9)$$

$$C_2 = \lambda_2 X + \lambda_1 Y \quad (2.10)$$

dengan syarat $\lambda_1 + \lambda_2 = 1$. *Crossover* aritmatik terutama cocok untuk kasus dimana variabel keputusan bernilai kontinyus. Kawin silang dilakukan untuk mendapatkan kombinasi yang lebih baik antara satu individu dengan individu yang lain dalam satu populasi.

Kombinasi linier parameter yang 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 ini membesar, maka akan semakin sedikit besar kromosom yang akan mengalami kawin silang. Misalkan nilai parameter ini adalah 0.5, maka akan ada sekitar separuh populasi yang akan mengalami kawin silang.



Gambar 2.6 Contoh Kawin Silang

(Sumber: Genetic Algorithms-A Tutorial, A. Townsend 2003)

2.3.6 Mutasi

Mutasi memungkinkan memunculkan individu-individu baru yang bukan berasal dari kawin silang. Mutasi mengacu pada perubahan urutan atau penggantian elemen dari vector solusi (pada masalah TSP), pemunculan nilai baru (optimasi fungsi). Elemen tersebut juga dipilih secara random. Misalkan untuk vektor solusi X , untuk elemen terpilih k

$$C = \begin{cases} x_i & i \neq k \\ x_k & i = k \end{cases} \quad (2.11)$$

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 0.01, maka akan ada 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*.

Sebelum Mutasi	1	1	0	1	0	0	0	0	0	1
Setelah Mutasi	1	1	1	1	0	0	0	0	0	1

Gambar 2.6 Contoh Mutasi

(Sumber: Genetic Algorithms-A Tutorial, A. Townsend 2003)

Secara garis besar, Algoritma Genetika dasar akan dijelaskan sebagai berikut

1. Bangkitkan populasi awal

Bangkitkan populasi awal atau kromosom-kromosom awal secara random. Evaluasi nilai setiap individu di dalam populasi awal ini dengan menggunakan fungsi *fitness*. Tentukan ukuran populasi, probabilitas kawin silang, dan probabilitas mutasi.

2. Set iterasi $t=1$

3. Pilih individu terbaik untuk disalin sejumlah tertentu untuk mengganti individu lain (elitisme)
4. Lakukan seleksi kompetitif untuk memilih anggota populasi sebagai induk untuk dilakukan kawin silang.
5. Lakukan kawin silang antar induk yang terpilih
6. Tentukan beberapa individu dalam populasi untuk mengalami proses mutasi
7. Jika belum mencapai konvergensi, set iterasi $t = t+1$
8. Kembali ke langkah 2

Dalam implementasi GA, sering digunakan bilangan biner (binary) untuk mewakili nilai variabel yang dicari. Umumnya akan ditentukan dulu berapa interval nilai kontinyus yang dicari. Sesudah itu akan bisa dicari berapa jumlah bilangan biner yang diperlukan untuk mewakili (berapa jumlah bit).

Jika terdapat bilangan biner $b_q b_{q-1}, \dots, b_2 b_1 b_0$ dimana $b_k = 0$ atau 1. Secara umum untuk mengubah bilangan biner menjadi bilangan nilai decimal bisa dilakukan dengan cara

$$y = \sum_{k=0}^q 2^k b_k \quad (2.12)$$

dengan $k = 0, 1, 2, \dots, q$. Misalkan set nilai biner 1 1 0 bisa dihitung desimalnya sebagai $2^0(0) + 2^1(1) + 2^2(2) = 6$, dimana perhitungan dilakukan dari belakang. Setelah dibangkitkan sejumlah individu dengan nilai biner, maka individu ini bisa dikembalikan nilainya ke dalam nilai kontinyus (decoding) dalam interval BB dan BA dengan rumus berikut

$$x = BB + \frac{BA-BB}{2^q-1} \sum_{k=0}^q 2^k b_k \quad (2.13)$$

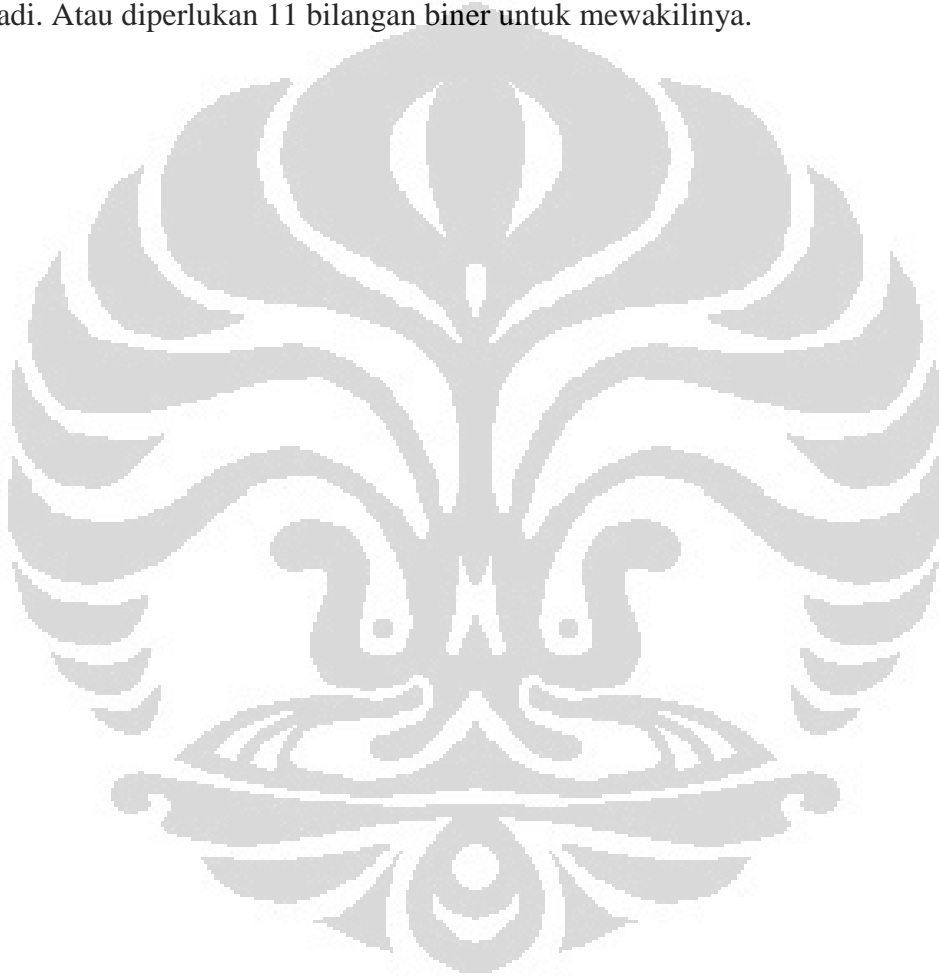
Jika sebuah variabel x yang batas bawah dan batas atasnya diberikan sebagai BA dan BB , akan diwakili dengan menggunakan satu set nilai biner, maka bisa dihitung berapa digit bilangan biner yang diperlukan jika diketahui tingkat akurasinya Δx .

$$2^q \geq \frac{BA-BB}{\Delta x} + 1 \quad (2.14)$$

Misalkan dikehendaki akurasi 0.01, suatu variabel kontinyus x dengan batas bawah -6 dan batas atas 6 akan diwakili dengan bilangan biner, maka kita perlu tahu berapa bilangan biner (bit) yang diperlukan.

$$2^q \geq \frac{6-(-6)}{0.01} + 1 = 1201 \quad (2.15)$$

Maka $q = 11$. Artinya, diperlukan 11 bit nilai biner untuk mewakili nilai x tadi. Atau diperlukan 11 bilangan biner untuk mewakilinya.



BAB 3

PENGUMPULAN DATA

3.1 Profil Perusahaan

Penelitian ini dilakukan di PT Pertamina (Persero). Perusahaan ini adalah perusahaan minyak dan gas bumi yang dimiliki pemerintah Indonesia (*National Oil Company*), yang berdiri sejak tanggal 10 Desember 1957. Sampai dengan perubahan status menjadi perusahaan perseroan pada tanggal 17 September 2003 berdasarkan Undang – Undang Republik Indonesia Nomor 22 tahun 2001 pada tanggal 23 November 2001 tentang Minyak dan Gas Bumi.

Adapun tujuan dari Perusahaan Perseroan adalah untuk:

1. Mengusahakan keuntungan berdasarkan prinsip pengelolaan Perseroan secara efektif dan efisien.
2. Memberikan kontribusi dalam meningkatkan kegiatan ekonomi untuk kesejahteraan dan kemakmuran rakyat.

Untuk mencapai maksud dan tujuan tersebut, Perseroan melaksanakan kegiatan usaha sebagai berikut:

1. Menyelenggarakan usaha di bidang minyak dan gas bumi beserta hasil olahannya dan turunannya.
2. Menyelenggarakan kegiatan usaha di bidang panas bumi yang ada pada saat pendiriannya, termasuk Pembangkit Listrik Tenaga Panas Bumi (PLTP) yang telah mencapai tahap akhir negosiasi dan berhasil menjadi milik Perseroan.
3. Melaksanakan pengusahaan dan pemasaran *Liquified Natural Gas* (LNG) dan produk lain yang dihasilkan dari kilang LNG.
4. Menyelenggarakan kegiatan usaha lain yang terkait atau menunjang kegiatan usaha sebagaimana dimaksud dalam nomor 1,2, dan3.

Sesuai dengan ketentuan dalam Undang – Undang MIGAS baru, perusahaan ini tidak lagi menjadi satu – satunya perusahaan yang memonopoli industri MIGAS dimana kegiatan usaha minyak dan gas bumi diserahkan kepada mekanisme pasar.

3.2 Data Yang Dibutuhkan

Untuk dapat menyelesaikan permasalahan penjadwalan karyawan ini, maka dibutuhkan beberapa data yang akan digunakan untuk mendapatkan solusi penjadwalan karyawan yang optimal dan untuk melaksanakan analisis perbandingan. Data yang dibutuhkan antara lain adalah data mengenai proyek pembangunan kapal (lokasi proyek), data karyawan pengawas pembangunan kapal beserta jenis-jenis pekerjaan yang berhubungan dengan pembangunan kapal, dan biaya-biaya penugasan.

3.2.1 Data Proyek

Data proyek yang dimaksud adalah proyek pembangunan kapal yang sedang dilaksanakan oleh Pertamina dan perusahaan galangan. Dalam penelitian ini, data proyek yang digunakan berjumlah 16 proyek yang semuanya tersebar di beberapa tempat yang berbeda, termasuk di China dan Korea Selatan. Jumlah proyek ini diambil sesuai jadwal pembangunan kapal yang sedang dilaksanakan.

Pada pelaksanaannya, proses pembangunan kapal ini tidak berjalan sesuai dengan waktu ideal pembangunan sebuah kapal yang disebabkan oleh beberapa faktor. Akan tetapi pada model yang akan dibuat, proses pembangunan kapal akan diasumsikan pada kondisi ideal dan waktu pengerjaan untuk semua kapal sama. Untuk data proyek pembangunan kapal, dapat dilihat pada tabel 3.1.

Tabel 3.1 Proyek Pembangunan Kapal>Nama Kapal

No.	Galangan	Negara	Nama Kapal	Hull No.	DWT	First cutting	Keel laying	Launching	Expected Delivery
1	Daya Radar Utama	Indonesia	Musi	148	3500	3/28/2011	11/23/2011	2/4/2012	4/1/2012
2	Taizhou Wuzhou Shipyard	China	Arjuna	WZL 1001	3500	3/14/2011	11/2/2011	2/1/2011	4/19/2012
3	PT. Dok Perkapalan Surabaya	Indonesia	Kakap	8602	6500	4/18/2011	12/28/2011	3/3/2012	5/13/2012
4	Taizhou Wuzhou Shipyard	China	Attaka	WZL 1002	3500	5/17/2011	1/5/2012	4/13/2012	6/23/2012
5	Daya Radar Utama	Indonesia	Meditran	149	3500	7/5/2011	3/15/2012	6/22/2012	8/27/2012
6	Dumas	Indonesia	Matindok	109	3500	9/18/2011	5/26/2012	8/14/2012	10/11/2012
7	PT. Dok Perkapalan Surabaya	Indonesia	Kasim	10604	6500	8/14/2011	4/2/2012	7/11/2012	9/1/2012
8	Zhejiang Chenye Shipyard	China	Senipah	804	30000	8/9/2011	4/15/2012	7/7/2012	9/25/2012
9	Daya Radar Utama	Indonesia	Mauhau	158	3500	10/17/2011	6/5/2012	9/13/2012	11/1/2012
10	Dumas	Indonesia	Merauke	111	3500	10/20/2011	6/25/2012	9/29/2012	11/13/2012
11	Zhejiang Chenye Shipyard	China	Sembakung	805	30000	11/8/2011	7/10/2012	10/4/2012	12/3/2012
12	NCS (New Century Shipyard)	China	Gamkonora	310501	85000	11/19/2011	7/8/2012	10/16/2012	12/17/2012
13	Pt. PAL	Indonesia	Pagar Dewa	271	17500	1/18/2012	9/28/2012	11/3/2012	1/12/2013
14	Pt. PAL	Indonesia	P. Berandan	272	17500	2/17/2012	10/25/2012	1/23/2013	3/18/2013
15	Taizhou Wuzhou Shipyard	China	not yet name	WZL 1106	3500	2/7/2012	10/17/2012	1/19/2013	3/12/2013
16	Hyundai Heavy Industry	Korea Selatan	not yet name		84000	1/9/2013	9/13/2013	12/9/2013	2/1/2014

Tabel di atas memperlihatkan data pembangunan kapal yang tersebar di beberapa tempat berbeda dan pada tabel ini juga diperlihatkan ukuran kapal dan

jadwal proses pembangunan kapal dari proses awal hingga kapal diantarkan. Dari tabel ini yang akan digunakan dalam model adalah periode pembangunan dan lokasi proyek.

3.2.2 Data Karyawan Beserta Jenis Pekerjaannya.

Data karyawan pengawas pembangunan kapal yang dibutuhkan dalam penelitian ini adalah jumlah pengawas pembangunan kapal yang akan dialokasikan terhadap pembangunan kapal yang sedang dilaksanakan. Pengawas pembangunan kapal disini terdiri dari *surveyor hull & outfitting*, *surveyor hull & painting*, *surveyor machinery & outfitting*, *surveyor electric & outfitting*. Selain dari divisi pengawasan, pengawasan pembangunan kapal juga melibatkan divisi perencanaan pembangunan kapal. Tabel 3.2 sampai tabel 3.4 berikut akan menunjukkan jumlah tenaga kerja pengawas pembangunan kapal yang ada di Pertamina.

Tabel 3.2 Data Karyawan di Pengawasan 1

No	Pengawas	Jumlah
1	Asisten Manager (hull & Outfitting)	1
2	Surveyor Hull & Outfitting	1
3	Surveyor Hull & Painting	1
4	Surveyor Machinery & Outfitting	2
5	Surveyor Electric & Outfitting	1

Tabel 3.3 Data Karyawan di Pengawasan 2

No	Pengawas	Jumlah
1	Asisten Manager (Machinery & Outfitting)	1
2	Surveyor Hull & Outfitting	2
3	Surveyor Hull & Painting	1
4	Surveyor Machinery & Outfitting	1
5	Surveyor Electric & Outfitting	1

Tabel 3.4 Data Karyawan di Perencanaan

No	Pengawas	Jumlah
1	Asisten Manager (machinery & Outfitting)	1
2	Surveyor Hull & Outfitting	1
3	Surveyor Hull & Painting	1
4	Surveyor Machinery & Outfitting	2
5	Surveyor Electric & Outfitting	1

Seperti yang sudah dijelaskan pada bab 1, bahwa dalam kondisi tertentu, jumlah tenaga kerja pengawas pembangunan kapal tidak akan mencukupi, sehingga Pertamina akan mengadakan kontrak tenaga kerja tambahan atau disebut pengawas subkontrak. Tenaga pengawas subkontrak ini digunakan selama pembangunan kapal berlangsung.

3.2.3 Data Periode

Proses pembangunan kapal memerlukan waktu ± 14 bulan, dan secara garis besar proses pembangunan kapal ini terdiri dari 4 tahapan, *first cutting* dikerjakan selama 8 bulan, *keel laying* dikerjakan selama 3 bulan, *launching* dikerjakan selama 2 bulan, dan *delivery* dikerjakan selama 1 bulan. Pengawasan tidak dilakukan secara bersama-sama melainkan sesuai dengan kebutuhan kapan pengawas tersebut akan diperlukan. Lampiran 1 menunjukkan periode waktu proses pembangunan kapal dari awal proses hingga selesai dan persebaran karyawan dengan jenis pekerjaan masing-masing pada salah satu jenis kapal unit *crude oil tanker 85,000 LTDW*.

3.2.4 Biaya – Biaya Penugasan

Biaya-biaya penugasan untuk pengawasan pembangunan kapal ini terdiri dari:

1. Biaya dinas karyawan

Biaya dinas karyawan adalah biaya karyawan ketika mereka menjalani tugas pengawasan ke lokasi proyek. Biaya dinas yang diterima karyawan adalah Rp 1.000.000/hari atau Rp 30.000.000/bulan apabila bertugas di dalam negeri sedangkan apabila bertugas di luar negeri maka Biaya dinas yang diterima oleh karyawan adalah \$ 160/hari atau dalam rupiah Rp 1.440.000/hari atau Rp 43.200.000/bulan (asumsi kurs 1 USD = Rp 9000).

2. Biaya *extra crew*

Biaya ini timbul jika tenaga pengawas dari Pertamina tidak mencukupi untuk meng-*cover* pengawasan di lokasi proyek, sehingga Pertamina harus menyewa tenaga subkontrak. Jika Pertamina menggunakan jasa tenaga

kerja subkontrak maka biaya yang harus dikeluarkan adalah Rp 50.000.000/ bulan. Dan karyawan subkontrak hanya berlaku di luar negeri, sedangkan di dalam negeri akan di-*cover* oleh karyawan Pertamina dengan memperpanjang durasi pengawasan (lembur). Namun dalam penelitian ini, karyawan yang lembur diasumsikan sebagai karyawan subkontrak.

3. Biaya Perjalanan Karyawan

Biaya ini merupakan biaya transportasi dari kantor (Jakarta) ke lokasi proyek dan sebaliknya. Untuk wilayah di Indonesia biaya transportasi disamakan, yaitu Rp 3.000.000 dan dari Jakarta ke China/Korea dan sebaliknya adalah Rp 7.000.000.

3.3 Model Penjadwalan Karyawan Pengawas Pembangunan Kapal

Dalam penjadwalan pembangunan kapal ini terdapat beberapa kendala yang harus diselesaikan.

3.3.1 Notasi dan Asumsi

➤ Asumsi:

- Setiap kapal mempunyai durasi pembangunan yang sama (14 bulan).
- Jenis pekerjaan *painting*, *machinery*, dan *electric* bisa mengerjakan pekerjaan *hull*.
- Kendala sumber daya yang dipertimbangkan hanyalah karyawan pengawas, sumber daya lain yang dibutuhkan dalam dalam pembangunan dianggap tak terbatas.
- Karyawan hanya boleh berada di lokasi proyek maksimal 3 bulan secara berurutan.
- Untuk karyawan subkontrak dianggap selalu tersedia untuk setiap jenis pekerjaan.
- Karyawan harus berada di kantor pusat minimal 2 orang.
- Penjadwalan dilaksanakan selama 36 periode atau sampai semua proyek selesai dilaksanakan.

➤ Indeks:

i = pekerjaan 0, 1, 2, 3

j = karyawan 1, 2, ..., J ($\rightarrow J = 18$)

k = proyek 0, 1, ..., K ($\rightarrow K = 16$)

l = periode 1, 2, ..., T ($\rightarrow T = 36$)

p = trip 1, ..., P

➤ Parameter:

p_{ij} = lama waktu karyawan j dengan pekerjaan i bekerja

Q_{ik} = himpunan kebutuhan karyawan dengan pekerjaan i pada periode l

➤ Variabel:

$x_{ijkl} = \begin{cases} 1, & \text{karyawan } j \text{ dengan pekerjaan } i \text{ berada di proyek } k \text{ pada periode } l \\ 0, & \text{jika tidak} \end{cases}$

$Z_{ikl} = \begin{cases} 1, & \text{karyawan subkontrak dengan pekerjaan } i \text{ di proyek } k \text{ pada periode } l \\ 0, & \text{jika tidak} \end{cases}$

tD_{ijkp} = waktu datang karyawan j dengan pekerjaan i pada proyek k pada trip ke- p

tP_{ijkp} = waktu karyawan j dengan pekerjaan i meninggalkan proyek k pada trip ke p

➤ Keterangan:

1. Pada trip ganjil ($2m-1$), karyawan j dengan pekerjaan i berada di kantor ($k=0$)
2. Pada trip genap ($2m$), karyawan j dengan pekerjaan i berada di proyek ($k=1-16$)

Untuk memberikan ilustrasi, berikut akan diberikan contoh pergerakan seorang karyawan dari dari trip 1 ke trip 2:

$$tP_{ij01} - tD_{ij01} = P_{1j1} \geq 1$$

$$tP_{ij01} \cdot X_{ijk01} = \sum_k tD_{ijk2} \cdot X_{ijk(p1j1)}$$

3.3.2 Fungsi Tujuan dan Kendala

➤ Fungsi Tujuan:

$$\begin{aligned}
 \text{Min } Z = & 30.000.000 \sum_{k=1}^9 \sum_{l=1}^{36} (tP_{ijk(2m)} - tD_{ijk(2m)}) \cdot X_{ijkl} \\
 & + 43.200.000 \sum_{k=10}^{16} \sum_{l=1}^{36} (tP_{ijk(2m)} - tD_{ijk(2m)}) \cdot X_{ijkl} \\
 & + 2 \times 3.000.000 \sum_{k=1}^9 tD_{ijk(2m)} \cdot (X_{ijk(P_{ij(2m)})} + Z_{ikl}) \\
 & + 2 \times 7.000.000 \sum_{k=10}^{16} tD_{ijk(2m)} \cdot (X_{ijk(P_{ij(2m)})} + Z_{ikl}) \\
 & + 50.000.000 \sum_{k=1}^K \sum_{l=1}^{36} (tP_{ijk(2m)} + tD_{ijk(2m)}) \cdot Z_{ikl} \\
 & + \sum_{l=tD_{ijk}}^{tP_{ijk-1}} 200.000.000 (Z_{ikl} - Z_{ik(l+1)})
 \end{aligned} \tag{3.1}$$

➤ Fungsi Kendala:

1. Karyawan j akan berada di salah satu proyek 0 s/d K pada setiap periode. Kendala ini digunakan untuk memastikan bahwa semua karyawan akan berada di kantor pusat atau di proyek.

$$\sum_{k=0}^K X_{ijkl} = 1 \quad \forall l, \forall j, \forall k \tag{3.2}$$

2. Karyawan j dengan pekerjaan i berada di kantor dapat lebih dari satu bulan. Kendala ini digunakan untuk menjelaskan bahwa seorang karyawan paling sedikit 1 bulan harus berada di kantor pusat setelah bertugas.

$$tP_{ij0(2m-1)} - tD_{ij0(2m-1)} = P_{ij(2m-1)} \geq 1 \tag{3.3}$$

3. Setelah berada di kantor, karyawan j dengan pekerjaan i boleh pergi ke proyek. Setelah karyawan sudah berada 1 bulan di kantor, maka karyawan tersebut boleh kembali ke proyek.

$$tP_{ij0(2m-1)} \cdot X_{ijk0(tP_{ij0(2m-1)})} = \sum_k tD_{ijk(2m)} \cdot X_{ijkP_{ij(2m)}} \quad (3.4)$$

4. Karyawan j dengan pekerjaan i berada di proyek k paling lama 3 bulan. Kendala ini menjelaskan bahwa jika karyawan sudah bertugas selama 1-3 bulan, maka dia harus kembali ke kantor pusat.

$$\sum_k (tP_{ijk(2m)} - tD_{ijk(2m)}) \cdot X_{ijk(P_{ij(2m)})} = \sum_k 3 \cdot X_{ijk(P_{ij(2m)})} \quad (3.5)$$

5. Setelah berada di proyek k , karyawan j dengan pekerjaan i harus kembali ke kantor.

$$tP_{ijk(2m)} \cdot X_{ijk(2m)} = tD_{ij0(2m+1)} \cdot X_{ij0(2m+1)} \quad (3.6)$$

6. Salah satu karyawan akan berada di proyek k pada saat kegiatan pekerjaan i pada proyek k dimulai.

$$\sum_j tD_{ijk(2m)} \cdot (X_{ijk(P_{ij(2m)})} + Z_{ikl}) = TM_{ik} (\sum_j X_{ijk(P_{ij(2m)})}) + Z_{ikl}, \quad \forall i, \forall k \quad (3.7)$$

7. Karyawan j harus meninggalkan proyek k setelah pekerjaan i selesai

$$\sum_j tP_{ijk(2m)} \cdot (X_{ijk(P_{ij(2m)})} + Z_{ikl}) = TS_{ik} (\sum_j X_{ijk(P_{ij(2m)})}) + Z_{ikl}, \quad \forall i, \forall k \quad (3.8)$$

8. Jika karyawan subkontrak pekerjaan i berada di proyek k maka karyawan tersebut harus berada di lokasi sampai proyek k tersebut selesai.

$$TS_{ik} \cdot Z_{ik(TS_{ik+1})} = tP_{ijk(tD_{ijk})} \cdot Z_{ik(tD_{ijk})}, \quad \forall i, \forall k \quad (3.9)$$

9. Pekerjaan *hull & outfitting* dapat dikerjakan oleh salah satu $i=1,2,3$

$i = 0 = \text{hull \& outfitting}$

$i = 1 = \text{painting \& hull}$

$i = 2 = \text{machinery \& outfitting}$

$i = 3 = \text{electric \& outfitting}$

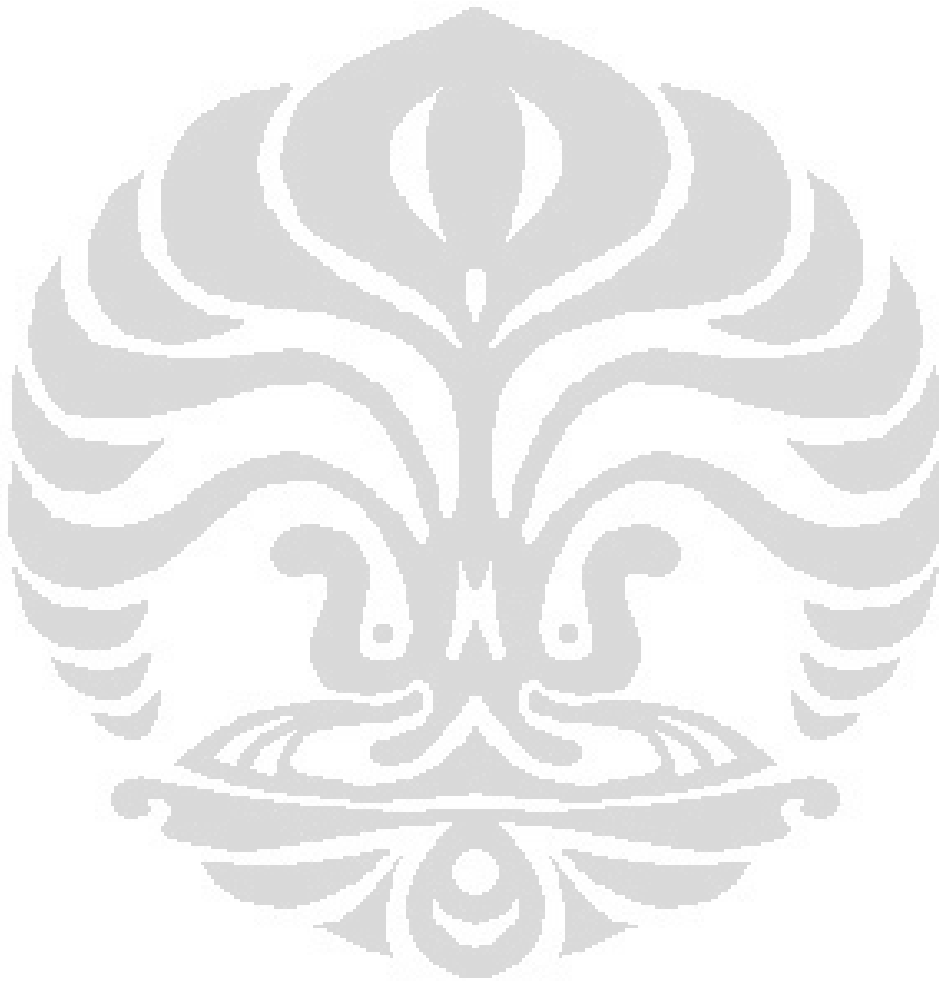
$$\sum_{i=1}^3 X_{ijkl} + \sum_{i=1}^3 Z_{ikl} \geq 1 \quad \forall j, \forall k, \forall l \quad (3.10)$$

10. Pada suatu periode, jumlah pekerja kegiatan i pada periode l adalah Q_{il}

$$\sum_{j=1}^J X_{ijkl} + \sum_k Z_{ikl} = Q_{il}, \forall i, \forall l \quad (3.11)$$

11. Di kantor pusat harus selalu ada karyawan sebanyak 2 orang

$$\sum_i \sum_j X_{ij0l} \geq 2, \forall l \quad (3.12)$$



BAB 4

PENGOLAHAN DATA DAN ANALISIS

4.1 Penyusunan Algoritma

Dalam melakukan optimasi penjadwalan karyawan dalam penelitian ini, digunakan algoritma genetika, yaitu algoritma yang berdasarkan prinsip-prinsip genetika dan seleksi alam, seperti yang sudah dijelaskan pada Bab 2. Untuk dapat melakukan optimasi penjadwalan karyawan dengan menggunakan algoritma genetika tersebut, maka dibuat suatu program dengan bantuan *software* MATLAB 7.9.0(R2009b). Tahapan algoritma genetika digunakan sebagai dasar dari langkah kerja yang dilakukan oleh program tersebut.

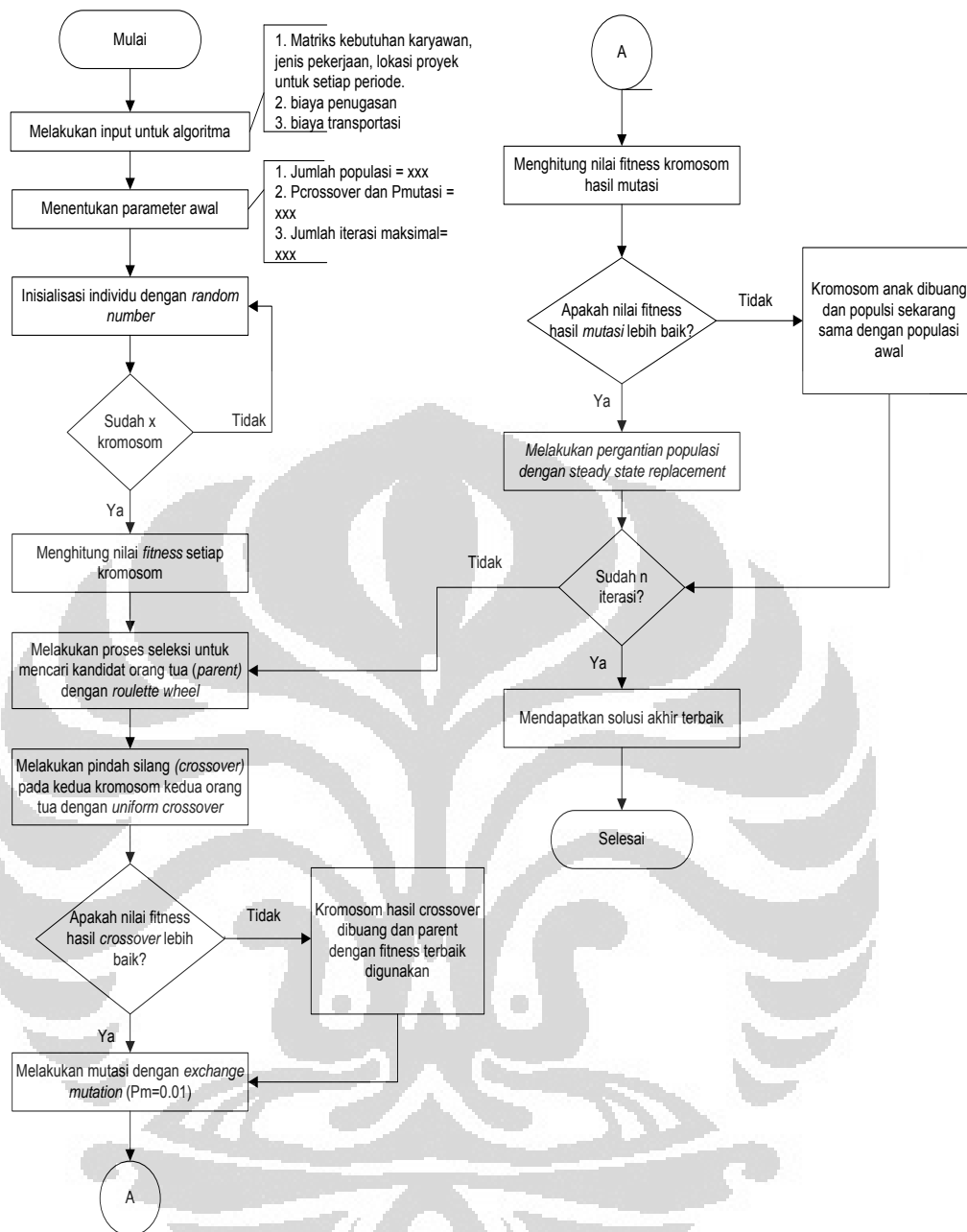
MATLAB adalah bahasa komputasi teknik tinggi dan merupakan lingkungan yang interaktif untuk pengembangan algoritma, visualisasi data, analisis data, dan komputasi numeric. MATLAB menyediakan fungsi-fungsi matematis untuk aljabar linear, statistic, optimasi, dan lain-lain. Selain itu, MATLAB juga menyediakan fitur-fitur dokumentasi dan integrasi algoritma berbasis MATLAB dengan bahasa dan aplikasi lain, seperti C, C++, Fortran, Java, COM, dan Microsoft Excel. Bahasa MATLAB memudahkan operasi-operasi vektor dan matriks yang merupakan dasar bagi permasalahan di bidang teknik dan ilmiah (www.mathworks.com).

4.1.1 Langkah-langkah Penyusunan Algoritma Genetika

Optimasi penjadwalan karyawan akan diselesaikan dengan algoritma genetika. Gambar 4.1 adalah prosedur algoritma genetika yang dibuat berdasarkan fungsi optimasi meminimumkan biaya penugasan karyawan. Prosedur penyelesaiannya akan diuraikan sebagai berikut:

1. Menentukan parameter/variabel yang digunakan, yaitu:

Parameter adalah parameter kontrol algoritma genetika, yaitu: ukuran populasi (*popsize*), peluang *crossover* (*pc*) dan mutasi (*pm*). Dalam menentukan parameter ada beberapa pilihan yang digunakan, tergantung jenis permasalahan maupun langkah-langkah yang digunakan:



Gambar 4.1 Diagram Alir Algoritma Genetika yang Digunakan

- Untuk permasalahan yang memiliki kawasan solusi cukup besar, De Jong merekomendasikan nilai parameter : (popsize; pc; pm) = (50;0,6;0,001).
- Bila rata-rata *fitness* setiap generasi digunakan sebagai indikator, maka Grefenstette merekomendasikan: (popsize; pc; pm) =(30;0,95;0,01)

- Bila *fitness* dari individu terbaik dipantau pada setiap generasi, maka usulannya adalah: (popsize; pc; pm) = (80;0,45;0,01)

Dalam penelitian ini, *fitness* dari individu terbaik dipantai pada setiap generasi dan berdasarkan referensi yang diperoleh dalam menentukan parameter, maka *fitness* dari individu terbaik dipantau pada setiap generasi, sehingga (popsize; pc; pm) = (80;0,45;0,01)

2. Membuat matrik-matrik yang akan menjadi input dalam model penjadwalan karyawan.

Data yang menjadi input awal pembuatan model adalah:

- Matrik konstanta X_{ijkl} , yaitu matrik kebutuhan karyawan j dengan pekerjaan i pada proyek k pada saat periode l . Hubungan antara karyawan, jenis pekerjaan, proyek, dan periode dilambangkan dengan notasi (1) atau (0), berdasarkan keterangan berikut:

$$x_{ijkl} = \begin{cases} 1, & \text{karyawan } j \text{ dengan pekerjaan } i \text{ berada di proyek } k \text{ pada} \\ & \text{periode } l \\ 0, & \text{jika tidak} \end{cases}$$

- Matrik konstanta Z_{ikl} , yaitu matrik kebutuhan karyawan subkontrak dengan pekerjaan i pada proyek k pada saat periode l . Hubungan antara karyawan, jenis pekerjaan, proyek, dan periode dilambangkan dengan notasi (1) atau (0), berdasarkan keterangan berikut:

$$Z_{ikl} = \begin{cases} 1, & \text{karyawan subkontrak dengan pekerjaan } i \text{ di proyek } k \text{ pada} \\ & \text{periode } l \\ 0, & \text{jika tidak} \end{cases}$$

- Biaya-biaya penugasan, yang terdiri dari biaya dinas dan biaya transportasi.

3. Menentukan kromosom/populasi awal secara acak.

Penentuan populasi inilah yang nantinya akan diperbaiki dengan menggunakan operator genetika yang ada. Penentuan populasi awal dilakukan

berdasarkan matrik X_{ijkl} dan Z_{ikl} . Prosedur penentuan populasi awal ini digunakan dengan mengikuti fungsi tujuan dan fungsi kendala seperti yang sudah dijelaskan pada bab 3.

4. Membuat salinan dari hasil elitism.

Karena terdapat pemilihan secara random pada tahap berikutnya, maka tidak ada jaminan bahwa suatu individu bernilai *fitness* tertinggi akan selalu terpilih. Walaupun individu bernilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitness*nya menurun) karena proses pindah silang dan mutasi. Oleh karena itu perlu dibuat kopian dari individu hasil elitism sehingga individu-individu terbaik akan selalu muncul.

5. Melakukan evaluasi nilai *fitness* terhadap kromosom hasil elitisme.

Oleh karena fungsi tujuan dari permasalahan ini adalah fungsi meminimumkan, maka evaluasi nilai *fitness* dilakukan dengan menggunakan rumus $1/(f(h)+1)$, dimana nilai *fitness* dengan nilai terendahlah yang terbaik.

6. Melakukan proses seleksi untuk mencari kandidat orang tua (*parent*).

Proses seleksi dilakukan dengan memilih dua kandidat jadwal (dua kromosom) sebagai orang tua. Yang menjadi input dalam proses seleksi adalah jadwal hasil elitisme. Sistem pemilihan dilakukan dengan *roulette wheel*. Proses seleksi ini melakukan operasi pemilihan kromosom dengan menggunakan bilangan acak, dimana kromosom dengan nilai *fitness* terendah mempunyai peluang terbesar untuk dipilih.

7. Melakukan pindah silang (*crossover*)

Setelah 2 *parent* dipilih dari populasi, diberlakukan operasi *crossover* untuk memproduksi satu *offspring* yang mewariskan karakteristik dari kedua parent tersebut. *Crossover* dilakukan dengan metode *uniform crossover* dimana gen yang akan diturunkan ke individu baru ditentukan sendiri. Dihitung biaya-

biaya penugasan dari kedua kromosom orang tua, dari masing-masing kromosom dicek tiap gennya sehingga yang memiliki biaya terkecil yang akan diturunkan ke kromosom anak (*offspring*). Jika hasil *crossover* memiliki nilai *fitness* yang lebih tinggi dari *parent* maka akan dilanjutkan ke proses mutasi, namun jika tidak maka yang maju ke tahap selanjutnya adalah *parent* dengan *fitness* terbaik.

8. Melakukan mutasi

Setelah operasi *crossover* selesai, *offspring* tersebut diberlakukan operasi mutasi. Mutasi dilakukan dengan dengan menukar-nukarkan gen pada kromosom anak. Metode mutasi yang digunakan adalah *exchange mutation*.

9. Melakukan pergantian populasi

Sebelum dilakukan pergantian populasi, dilakukan evaluasi biaya terhadap *offspring* hasil mutasi. Apabila biaya *offspring* tersebut lebih rendah dari biaya maksimum pada populasi awal, maka *offspring* tersebut akan menggantikan kromosom dengan biaya maksimum tersebut. Sebelum *offspring* dimasukkan ke dalam populasi, susunan *offspring* dimasukkan ke dalam populasi, susunan gen *offspring* diperiksa apakah mempunyai kesamaan identik dengan susunan gen tiap kromosom pada populasi. Hal ini mencegah duplikasi kromosom pada populasi.

Prosedur-prosedur di atas akan berulang terus hingga tercapainya *stopping criteria*, yaitu maksimum iterasi sesuai dengan jumlah yang ditentukan. Pada penelitian ini, maksimum iterasi yang digunakan adalah 2000 iterasi. Penentuan maksimum iterasi 2000 iterasi dilakukan dengan melihat bahwa kromosom terbaik tidak lagi mengalami perubahan setelah iterasi 1000-1400 pada beberapa kali percobaan, sehingga disimpulkan bahwa dengan menggunakan maksimum iterasi 2000 sudah mendapatkan hasil yang mendekati optimal. Lalu akan diambil kromosom terbaik yang akan menjadi solusi penjadwalan karyawan. Kromosom terbaik adalah kromosom dengan biaya paling minimum.

4.1.2 Verifikasi dan Validasi Program

Seperti yang telah dijelaskan sebelumnya bahwa masalah penjadwalan ini akan diselesaikan dengan menggunakan *software* MATLAB. Oleh karena itu, perlu dilakukan terlebih dahulu verifikasi dan validasi terhadap program. Verifikasi merupakan tahap untuk melihat kesesuaian model program dengan konsep model yang kita inginkan. Salah satu parameter model program dikatakan telah terverifikasi apabila program tersebut berjalan sesuai dengan konsep, di mana terdapat perubahan *output*. Apabila program yang dijalankan dengan mengubah-ubah kombinasi parameter dan menghasilkan *output* yang berbeda-beda, berarti program tersebut telah terverifikasi. Tahap verifikasi yang dilakukan dengan mengubah kombinasi parameter dapat dilihat pada tabel 4.1.

Tabel 4.1 Skenario Parameter

No	Kombinasi			Hasil	
	Ukpop	Pmutasi	Iterasi	Lama <i>Running</i> (detik)	Total Biaya
1	80	0.01	50	46.3707	29852600000
2			100	68.5	29842000000
3			1000	478.3097	29745000000
4			2000	1037.9	29574800000
5	50	0.3	50	40.9673	29956000000
6			100	66.3722	29683400000
7			1000	460.3286	29651000000
8			2000	107.6	29634000000

Dari tabel di atas, terlihat bahwa terjadi perubahan output atau total biaya apabila dilakukan kombinasi parameter yang berbeda-beda sehingga dapat disimpulkan bahwa program tersebut telah terverifikasi.

Setelah program terverifikasi, maka selanjutnya perlu dilakukan validasi dengan tujuan untuk memastikan bahwa program tersebut menghasilkan output yang benar. Indikatornya adalah ketika program dihadapkan pada suatu masalah, hasil perhitungannya bernilai sama dengan perhitungan manual. Berikut adalah hasil perhitungan secara manual dengan kombinasi Ukpop = 80, Pm=0.01, Maksimum iterasi = 2000.

Tabel 4.2 Perhitungan Manual Total Biaya

No	Jenis Biaya	Total Biaya (Rupiah)
1	Dinas Karyawan dalam negeri	$135 \times 30.000.000 = 4050000000$
2	Dinas Karyawan luar negeri	$69 \times 43.200.000 = 2980800000$
3	Dinas Subkontrak	$424 \times 50.000.000 = 21200000000$
4	Transportasi dalam negeri	$2 \times 98 \times 3.000.000 = 588000000$
5	Transportasi luar negeri	$2 \times 54 \times 7.000.000 = 756000000$
Total Biaya		29574800000

Dari tabel di atas, dapat dilihat bahwa hasil perhitungan manual di atas sama dengan hasil dari *run* program pada tabel 4.1 kombinasi $Ukpop = 80$, $Pm=0.01$, Maksimum iterasi = 2000 . Dengan demikian program telah tervalidasi.

4.2 Input Data

4.2.1 Data

Untuk melakukan optimasi penjadwalan karyawan pengawas pembangunan kapal dengan menggunakan Algoritma Genetika dibutuhkan data seperti yang telah dijelaskan dalam Bab 3, yaitu data proyek, data karyawan, penggal waktu dan persebaran karyawan beserta jenis pekerjaan, dan komponen biaya. Namun yang menjadi input dalam model penjadwalan ini adalah hasil pengolahan data dari keempat data tersebut yaitu kebutuhan karyawan j dengan pekerjaan i pada proyek k pada periode l dan juga biaya dinas.

4.2.2 Parameter

Seperti yang sudah dijelaskan sebelumnya bahwa ada beberapa pendekatan dalam menentukan parameter ukuran populasi, parameter kontrol *crossover*, parameter kontrol mutasi, dan jumlah maksimum iterasi. Dalam penelitian ini, *fitness* dari individu terbaik dipantau pada setiap generasi dan berdasarkan referensi yang diperoleh dalam menentukan parameter, bila *fitness* dari individu terbaik dipantau pada setiap generasi, maka usulannya adalah: $(popsize; pc; pm) = (80;0,45;0,01)$.

Selanjutnya untuk menentukan nilai parameter jumlah maksimum iterasi yang akan digunakan, akan dilakukan perbandingan hasil *running* program dengan beberapa kombinasi parameter yang berbeda(Tabel 4.3). Sedangkan dari

Tabel 4.4 dapat dilihat bahwa dengan maksimum iterasi 2000 memberikan hasil terbaik dibandingkan dengan nilai maksimum iterasi lainnya. Oleh karena itu, dalam model penjadwalan karyawan pengawas pembangunan kapal digunakan nilai parameter yang ditunjukkan oleh tabel berikut.

Tabel 4.3 Kombinasi Parameter yang Berbeda

No	Kombinasi			Hasil	
	Ukpop	Pmutasi	Iterasi	Lama <i>Running</i> (detik)	Total Biaya
1	80	0.01	50	46.3707	29752600000
				46.6031	29872000000
				47.2569	29872000000
				44.6745	29872000000
	Rata-rata			46.2263	29842150000
2	80	0.01	100	68.5	29842000000
				67.7571	29948000000
				67.9877	29805400000
				67.6982	29915600000
	Rata-rata			67.98575	29877750000
3	80	0.01	1000	478.3097	29745000000
				449.1709	29891000000
				457.9172	29710000000
				472.5568	29695200000
	Rata-rata			464.48865	29760300000
4	80	0.01	2000	935.2334	29642200000
				1037.9	29574800000
				997.7484	29691200000
				1014.7	29672000000
	Rata-rata			996.39545	29594400000

Tabel 4.4 Parameter Model

Parameter	
Ukpop	80
Pmutasi	0.01
Maks.Iterasi	2000

4.3 Pengolahan Data dan Hasil

Setelah ditentukan parameter permasalahan penjadwalan ini, maka program akan di-*run* sebanyak 5 kali. Karena fungsi tujuan permasalahan ini ialah

meminimalkan, maka akan diambil hasil terbaik dari kelima *run* tersebut dengan indikator ialah *fitness* terbaik yang ditandai dengan total biaya terkecil. Tabel 4.5 menunjukkan hasil *run* sebanyak 5 kali.

Dari hasil *running* program sebanyak 5 kali, hasil *running* ke memberikan hasil terbaik dengan total biaya terkecil. Solusi jadwal bisa dilihat pada lampiran 2 yang merupakan hasil *running* program dengan total biaya terkecil sebesar Rp 29.574.800.000 dan waktu *running* selama 1037.9 detik.

Tabel 4.5 Hasil *Run* Program

<i>Running</i> ke-	Hasil	
	Lama <i>running</i> (detik)	Total Biaya (Rupiah)
1	935.2334	29642200000
2	1037.9	29574800000
3	997.7484	29691200000
4	1014.7	29672000000
5	989.4841	29661300000
Rata-rata		29627850000

4.4 Analisis

Kasus penjadwalan karyawan termasuk permasalahan yang cukup kompleks karena terdapat banyak aspek permasalahan yang harus dipertimbangkan dalam menghasilkan jadwal yang baik. Tingkat kerumitan penyelesaian masalah penjadwalan karyawan tergantung pada karakteristik yang terdapat pada masing-masing instansi atau organisasi karena ini berhubungan dengan peraturan yang ada di instansi tersebut.

Usulan penyelesaian terhadap masalah penjadwalan karyawan ini adalah dengan menggunakan algoritma genetika yang dituangkan dalam satu bahasa pemrograman MATLAB, yang selanjutnya digunakan untuk melakukan pengolahan data untuk membentuk solusi jadwal yang optimal. Fungsi tujuannya adalah untuk meminimalkan biaya total.

Analisis terhadap pengolahan data ini dibagi menjadi tiga bagian, yaitu analisis metode, analisis program, dan analisis hasil optimasi.

4.4.1 Analisis Metode

Metode yang digunakan untuk menyelesaikan permasalahan penjadwalan karyawan ini adalah algoritma genetika. Algoritma Genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup.

Cara kerja algoritma ini dimulai dari inisialisasi populasi awal yang akan menjadi kandidat solusi. Solusi awal dibuat dengan pemilihan secara acak dari kemungkinan-kemungkinan yang ada. Lalu setiap individu dalam populasi awal tersebut akan masuk ke dalam proses seleksi yang dilakukan secara acak kemudian dikawinsilangkan, namun kawin silang tidak dilakukan secara acak karena memiliki konsep agak berbeda dengan proses mutasi. Sementara mutasi juga dipilih secara acak dimana apabila bilangan random yang dihasilkan lebih kecil dari probabilitas mutasi maka individu tersebut akan mengalami proses mutasi.

Terdapatnya sistem pemilihan secara acak dalam algoritma ini akan membuat perbedaan urutan jadwal yang dihasilkan pada setiap *run* program yang dilakukan. Kombinasi parameter yang sama sekalipun belum tentu menghasilkan urutan urutan jadwal yang sama jika di-*run* beberapa kali. Oleh karena itu skenario parameter merupakan bagian penting dalam pengujian metode ini, karena tiap kasus membutuhkan kombinasi parameter yang unik untuk menghasilkan jadwal yang sesuai dengan fungsi tujuan.

Kelebihan Algoritma Genetika adalah pada cara kerjanya yang paralel. Algoritma Genetika bekerja dalam ruang pencarian yang menggunakan banyak individu sekaligus, sehingga kemungkinan metode ini untuk terjebak pada kondisi ekstrim lokal lebih kecil dibandingkan metode lain. Algoritma Genetika juga mudah diimplementasikan, yaitu dengan menyelesaikan masalah lain hanya dengan membuat kromosom dan fungsi *fitness* yang baru sehingga Algoritma bisa dijalankan.

Kekurangan Algoritma Genetika adalah dalam hal waktu komputasi karena harus melakukan evaluasi *fitness* pada semua solusi di setiap iterasinya.

Sehingga Algoritma Genetika bisa lebih lambat dibanding metode lain, namun hal ini bisa diakhiri dengan mengakhiri komputasi pada waktu yang diinginkan dengan maksimum iterasi yang diinginkan. Akan tetapi perlu diingat bahwa semakin banyak iterasi, semakin besar peluang untuk mendapatkan hasil yang terbaik. Waktu komputasi yang lama bisa dicegah juga dengan menggunakan perangkat yang lebih canggih. Karena semakin canggih, maka komputasi akan semakin cepat. Kekurangan lain dari metode ini adalah, Algoritma Genetika tidak cukup baik untuk mencari solusi yang sangat dekat dengan solusi optimal. Optimasi penjadwalan dengan menggunakan Algoritma Genetika ini hanya mendapatkan solusi mendekati optimal.

4.4.2 Analisis Program

Bahasa pemrograman yang digunakan untuk menginterpretasikan algoritma genetika pada penelitian ini adalah MATLAB 7.9.0 (R2009b). Secara umum program yang dibuat untuk menyelesaikan permasalahan penjadwalan ini dapat berjalan dengan baik. Solusi akhir yang dihasilkan selalu lebih baik dari solusi awal.

Kelemahan dari program ini adalah waktu *run* program yang relatif lama. Seperti yang telah dijelaskan sebelumnya, waktu *run* program ini dipengaruhi oleh konfigurasi parameter yang digunakan. Nilai iterasi terkecil yang digunakan dalam *trial* adalah 50 dan memerlukan waktu *running* tercepat selama 44.6745 detik. Sementara iterasi terbesar yang digunakan adalah 2000 dan memerlukan waktu *running* terlama yaitu 1037.9 detik. Selain itu, kemampuan performa komputer juga sangat mempengaruhi waktu *running* program.

4.4.3 Analisa Hasil Optimasi

Pada bagian ini akan dianalisis mengenai hasil optimasi penjadwalan karyawan pengawas pembangunan kapal. Dari jadwal yang dihasilkan dapat dilihat bahwa seluruh karyawan telah terjadwal pada periode yang berbeda. Semua kebutuhan karyawan pada proyek, jenis pekerjaan, dan periode telah terpenuhi. Artinya pembangunan kapal pun bisa berlangsung dengan baik. Selain itu, sesuai dengan peraturan yang berlaku di Pertamina sendiri, bahwa karyawan

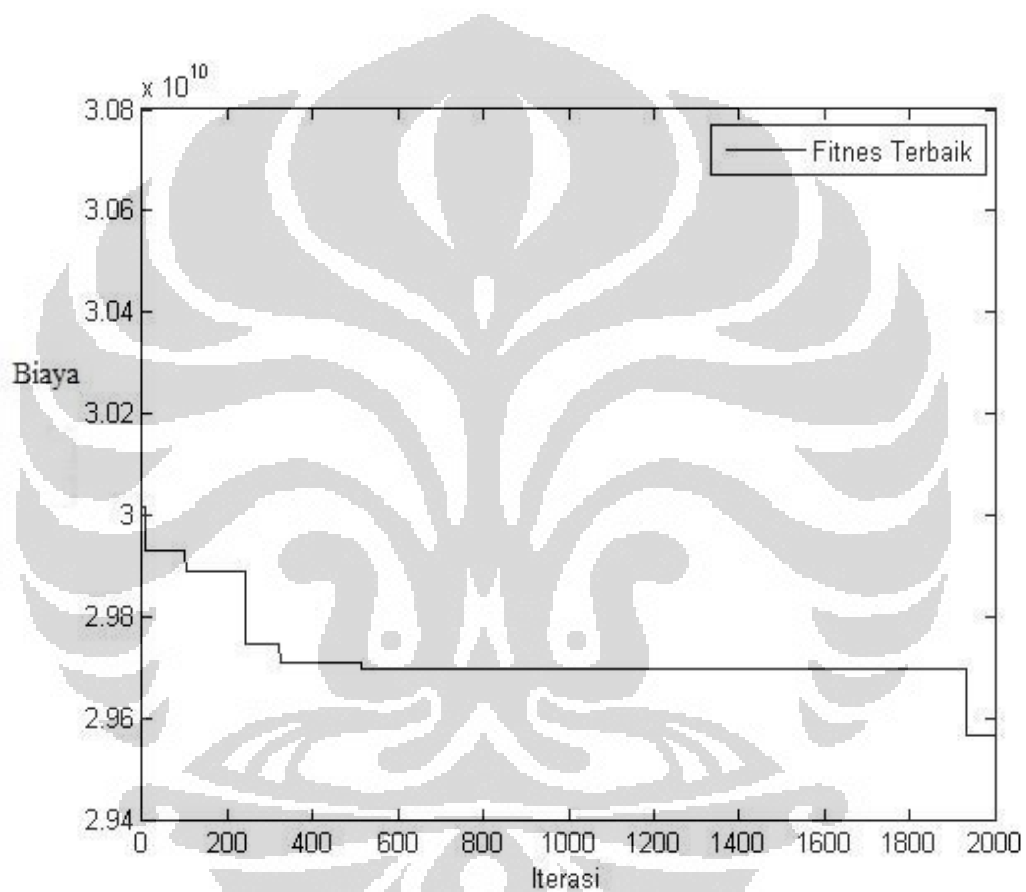
hanya boleh berada di proyek maksimal 3 bulan dan harus kembali ke kantor untuk periode lebih atau sama dengan 1 bulan juga sudah terpenuhi. Selain itu, tidak terdapat karyawan yang bertugas lebih dari satu tempat atau proyek dalam waktu yang bersamaan.

Kendala karyawan subkontrak “ketika sudah berada di lokasi proyek k , maka karyawan subkontrak tidak bisa berpindah ke proyek lain sampai proyek selesai juga sudah terpenuhi dan kendala dimana karyawan subkontrak harus meninggalkan proyek pada saat proyek sudah selesai juga terpenuhi.” Semua kendala dalam penelitian ini sudah terpenuhi secara sempurna sehingga optimasi penjadwalan karyawan pengawas pembangunan kapal dengan metode algoritma genetika ini disimpulkan sudah menghasilkan jadwal yang layak.

Melalui model optimasi penjadwalan karyawan pengawas pembangunan kapal dengan mengambil data dari departemen pembangunan kapal Pertamina bisa dilihat bahwa kebutuhan pengawas pembangunan kapal sangat bergantung terhadap karyawan subkontrak karena keterbatasan jumlah karyawan tetap yang tersedia.

Tujuan dari penjadwalan ini adalah meminimumkan biaya penugasan karyawan yang ada di Pertamina ataupun tenaga subkontrak. Dalam penelitian ini juga digunakan biaya penalti terhadap karyawan tetap apabila menggantikan posisi karyawan subkontrak. Karena Algoritma Genetika mencari biaya terendah, maka dia akan memilih karyawan tetap sebagai alternatif karena biayanya lebih rendah, sehingga diperlukan pemberian biaya penalti sehingga Algoritma Genetika akan tetap memilih karyawan subkontrak untuk memenuhi kendala “ketika sudah berada di lokasi proyek k , maka karyawan subkontrak tidak bisa berpindah ke proyek lain sampai proyek selesai juga sudah terpenuhi dan kendala dimana karyawan subkontrak harus meninggalkan proyek pada saat proyek sudah selesai juga terpenuhi.” Dengan demikian, jadwal dengan total biaya penugasan yang minimum adalah jadwal yang optimal dengan menggunakan algoritma genetika. Berikut merupakan grafik yang menunjukkan penurunan nilai *fitness* dengan beberapa iterasi. Nilai *fitness* disini adalah nilai total biaya atau sesuai dengan fungsi tujuan dan karena ini adalah fungsi meminimumkan biaya, maka semakin rendah semakin baik.

Dengan menggunakan parameter ukuran populasi: 80, probabilitas *crossover*: 0.45, probabilitas mutasi: 0.01, dan maksimum iterasi: 2000. Dari grafik dapat dilihat bahwa dengan iterasi kurang dari 2000, total biaya mengalami penurunan dari iterasi awal hingga iterasi 2000. Hasil optimasi penjadwalan karyawan pengawas pembangunan kapal dengan menggunakan algoritma genetika bisa dilihat di lampiran 2, dengan menggunakan parameter seperti di atas.

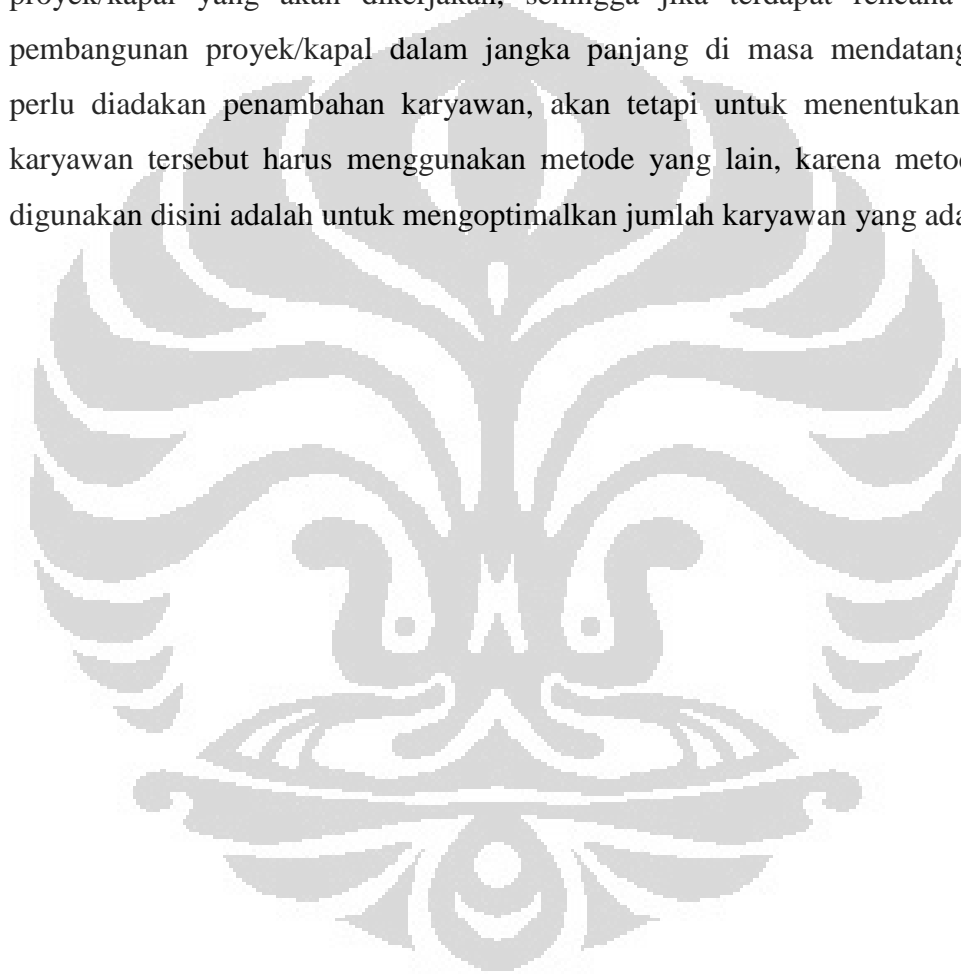


Gambar 4.1 Grafik Penurunan Nilai Total Biaya

Sebagai perbandingan, program ini juga dicoba menggunakan skenario penambahan jumlah karyawan menjadi 32 orang dengan 8 orang masing-masing keahlian untuk mengetes apakah masih layak apabila jumlah karyawan ditambah menjadi 32 (lampiran 3). Skenario ini digunakan untuk melihat seberapa besar pengaruh jumlah karyawan tetap terhadap biaya penugasan. Setelah melakukan beberapa kali *running* dengan menggunakan parameter yang sama, hasilnya karyawan dengan alokasi karyawan subkontrak berkurang dan biaya total

penugasan Rp 27.885.600.000 atau memiliki perbedaan biaya Rp 1.689.200.000 dari penggunaan 18 karyawan

Jumlah karyawan subkontrak yang tersedia juga tidak bisa diketahui secara pasti jumlahnya, sehingga apabila karyawan kebutuhan karyawan subkontrak semakin banyak, akan sangat sulit membuat jadwal yang baik. Akan tetapi masalah penambahan karyawan tetap sebaiknya dilihat dari seberapa perlu kebutuhan karyawan di masa mendatang, tergantung seberapa banyak jumlah proyek/kapal yang akan dikerjakan, sehingga jika terdapat rencana banyak pembangunan proyek/kapal dalam jangka panjang di masa mendatang, maka perlu diadakan penambahan karyawan, akan tetapi untuk menentukan jumlah karyawan tersebut harus menggunakan metode yang lain, karena metode yang digunakan disini adalah untuk mengoptimalkan jumlah karyawan yang ada.



BAB 5

KESIMPULAN DAN SARAN

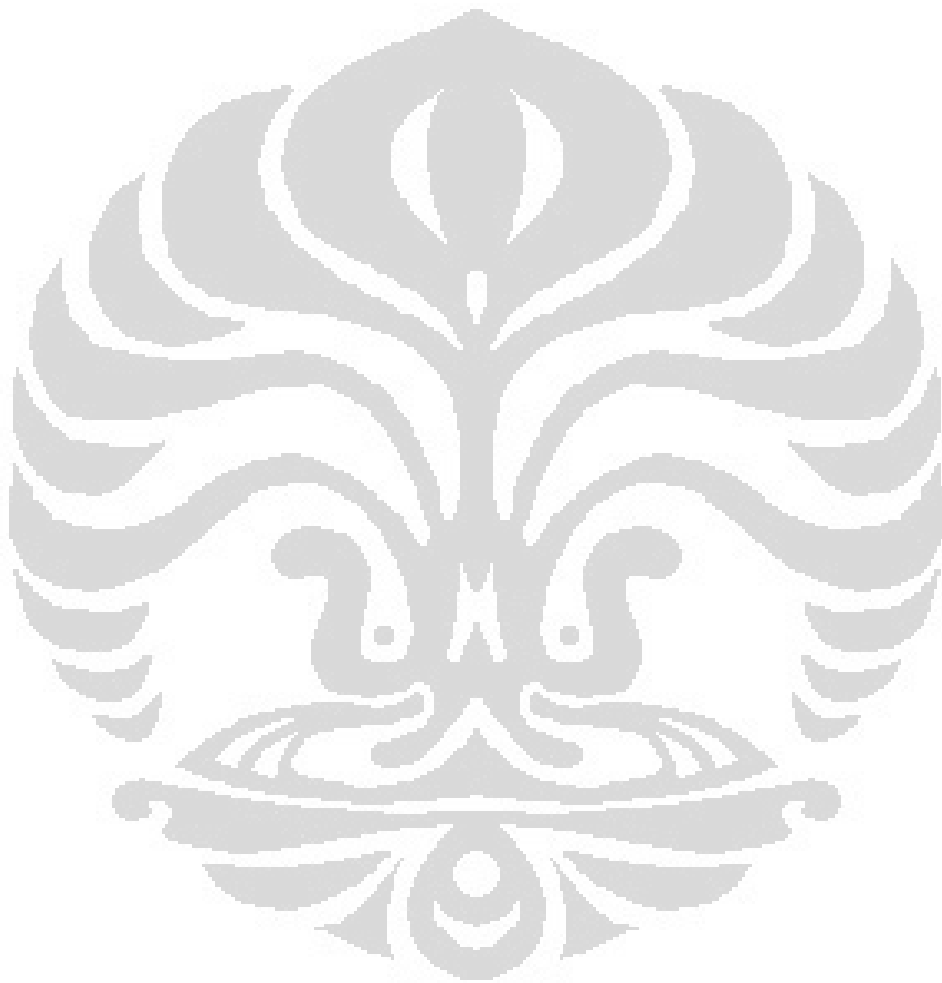
5.1 Kesimpulan

Berdasarkan penelitian optimasi penjadwalan karyawan pengawas pembangunan kapal dengan algoritma genetika dengan bahasa pemrograman MATLAB, diperoleh kesimpulan sebagai berikut:

- Diperoleh model penjadwalan karyawan pengawas pembangunan kapal yang memberikan penjadwalan yang layak dan optimal.
- Dari ujicoba yang telah dilakukan dengan menggunakan beberapa kombinasi parameter, penjadwalan karyawan dengan algoritma genetika telah memenuhi semua kendala dengan ukuran populasi 80, probabilitas mutasi 0.45, dan maksimum iterasi 2000 dengan total biaya terkecil Rp 29.574.800.000 dan hasil ini mengalami perbaikan dari solusi awal sampai maksimum iterasi.
- Jadwal yang dihasilkan juga telah mendekati optimal dengan memenuhi tujuan penelitian, yaitu meminimumkan biaya total penugasan karyawan. Dengan demikian algoritma genetika dapat menyelesaikan masalah penjadwalan karyawan dengan memenuhi kendala.

5.2 Saran

- Apabila jumlah proyek bertambah untuk jangka waktu yang panjang, maka penambahan jumlah karyawan yang sesuai perlu dipertimbangkan apabila ingin memperoleh jadwal lebih baik dan memperoleh biaya yang lebih minimum.



DAFTAR REFERENSI

- Antoni, Akbar. (2011). *Integrasi dan Ship Scheduling Dengan Time Window Constraint Pada Mobile Floating Storage Dengan Model Mixed Integer Programming*. Universitas Indonesia, Departemen Teknik Industri
- Astri, Dhanika. (2006). *Model Optimasi Rotasi Penugasan Awak Pesawat Menggunakan Metode Algoritma Genetika (Studi Kasus: Pesawat Boeing 737 Pada PT X)*. Universitas Indonesia, Indonesia, Departemen Teknik Industri.
- DeJong, K.A. & Spears, W.M. (1990). *An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms*. Proc. First Workshop Parallel Problem Solving from Nature, Springer-Verlag, Berlin, hal. 38-47.
- Grefenstette, J.J. (1986). *Optimization of Control Parameters for Genetic Algorithms*. IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-16, No. 1, hal. 122-128.
- Güneş, Evrim Didem. (1999). *Workforce Scheduling*. Department of Industrial Engineering, Bilkent University.
- Pinedo, Michael, L. (2004). *Planning and Scheduling in Manufacturing and Services*. Department of Operations Management, Stern School of Business, New York University.
- Santosa, Budi. (2011). *Metoda Metaheuristik: Konsep dan Implementasi*. Penerbit: Guna Wisya, Surakarta, Indonesia.
- Sianipar, Yanti Nitra. (2010). *Optimasi Penjadwalan Mata Kuliah di Perguruan Tinggi dengan Menggunakan Algoritma Memetika*. Universitas Indonesia, Departemen Teknik Industri.
- Schwiegelshohn, Uwe. (2004, Summer Term). *Scheduling Problems and Solution*. Computer Engineering Institute, University of Dortmund.
- Taha, Hamdy A. (1997). *Operation Research: An Introduction 6th ed*. Prentice-Hall, Inc. New Jersey.
- Townsend, A. (2003). *Genetic Algorithms – a Tutorial*.

Lampiran 1: Penggal Waktu Pembangunan Kapal & Persebaran Karyawan

Deskripsi/Bulan ke-	1	2	3	4	5	6	7
Schedule Pembangunan	First Cutting						
Asisten Manager	1	1	1	1	1	1	1
Surveyor Hull & Outfitting	1	1	1	1	1	1	1
Surveyor Hull & Painting				1	1	1	1
Surveyor Machinery & Outfitting				1	1	1	1
Surveyor Electric & Outfitting							

Deskripsi/Bulan ke-	8	9	10	11	12	13	14
Schedule Pembangunan		Keel Laying			Launching		Delivery
Asisten Manager	1	1	1	1	1	1	1
Surveyor Hull & Outfitting	1	1	1	1	1	1	1
Surveyor Hull & Painting	1	1	1	1	1	1	1
Surveyor Machinery & Outfitting	1	1	1	1	1	1	1
Surveyor Electric & Outfitting			1	1	1	1	1

Lampiran 2: Matriks Penjadwalan Hasil Optimasi dengan 18 Karyawan Tetap

No	Proyek	Jenis Pekerjaan	Periode																																												
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36									
1	Musri	Hull & Outfitting	10	10	10	2	2	5	5	5	15	15	15	16	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
		Hull & Painting	0	0	0	8	8	8	1	1	1	SK25	SK25	SK25	SK25	SK25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
		Machinery & Outfitting	0	0	0	14	14	14	SK1	SK1	SK1	SK1	SK1	SK1	SK1	SK1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
2	Kalap	Electricity & Outfitting	0	0	0	0	0	0	0	0	0	11	11	11	SK41	SK41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
		Hull & Outfitting	0	1	1	1	12	12	12	SK6	SK6	SK6	SK6	SK6	SK6	SK6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
		Hull & Painting	0	0	0	0	15	15	15	SK7	SK7	SK7	SK7	SK7	SK7	SK7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
3	Meditran	Machinery & Outfitting	0	0	0	0	11	11	11	SK8	SK8	SK8	SK8	SK8	SK8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
		Electricity & Outfitting	0	0	0	0	0	0	0	0	0	6	6	6	18	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
		Hull & Outfitting	0	0	0	0	17	17	17	SK9	SK9	SK9	SK9	SK9	SK9	SK9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	MatIndok	Hull & Painting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
		Machinery & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
		Electricity & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
5	Kasim	Hull & Outfitting	0	0	0	0	4	4	4	4	4	SK17	SK17	SK17	SK17	SK17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		Hull & Painting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
		Machinery & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
6	Mauhau	Electricity & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
		Hull & Outfitting	0	0	0	0	0	0	0	0	0	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12	SK12		
		Hull & Painting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
7	Merauke	Machinery & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
		Electricity & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
		Hull & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
8	Pagar Dewa	Hull & Painting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
		Machinery & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
		Electricity & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	P. Berandan	Hull & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
		Hull & Painting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
		Machinery & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	Arjuna	Electricity & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
		Hull & Outfitting	5	5	5	18	18	18	SK2	SK2	SK2	SK2	SK2	SK2	SK2	SK2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
		Hull & Painting	0	0	0	9	9	9	SK3	SK3	SK3	SK3	SK3	SK3	SK3	SK3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	Ataka	Machinery & Outfitting	0	0	0	13	13	13	SK4	SK4	SK4	SK4	SK4	SK4	SK4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		Electricity & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		Hull & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	Senipah	Hull & Painting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
		Machinery & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		Electricity & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	Sembakung	Hull & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		Hull & Painting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Machinery & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	Gamkonora	Electricity & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		Hull & Outfitting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Hull & Painting	0	0																																											

Lampiran 4: Script M-File Program

1. Bangkitkan Solusi

```

function [min_Z,data_model_karyawan]=bangkit_solusi()
load data_model
for i=1:64
    for j=1:36
        data_model_karyawan{i,j}=[];
    end
end
subkontraktor_tersedia=(1:64)';
tabel_dummy = zeros(64,36);
tempat_subkontraktor=zeros(64,1);
lama_karyawan_di_tempat=zeros(64,1);
alur_tempat_karyawan=zeros(18,36);
karyawan_di_kantor=(1:18)';
lama_karyawan_di_kantor=zeros(18,1);
karyawan_di_tempat=zeros(64,1);
Xijkl=zeros(4,18,17,36);
Zikl=zeros(4,17,36);
load keahlian_karyawan_awal
for l=1:36
    if l==1
        karyawan=(1:18)';
    else
        karyawan=[];
        if length(karyawan_di_kantor)>2
            for kk=1:length(karyawan_di_kantor)
                if lama_karyawan_di_kantor(kk,1)>=1
                    karyawan=[karyawan;karyawan_di_kantor(kk,1)];
                end
            end
        end
    end
end
keahlian_karyawan=keahlian_karyawan_awal(karyawan,:);
for i=1:64
    tempat=ceil(i/4);
    if data_model(i,1)==1
        jenis_pekerjaan=mod(i,4);
        if jenis_pekerjaan==0
            jenis_pekerjaan=4;
        end
        switch jenis_pekerjaan
            case 1
                jenis_dibutuhkan{1,1}='Hull';
                jenis_dibutuhkan{2,1}='Outfitting';
            case 2
                jenis_dibutuhkan{1,1}='Hull';
                jenis_dibutuhkan{2,1}='Painting';
            case 3
                jenis_dibutuhkan{1,1}='Machinery';
                jenis_dibutuhkan{2,1}='Outfitting';
            case 4

```

```

        jenis_dibutuhkan{1,1}='Electricity';
        jenis_dibutuhkan{2,1}='Outfitting';
    end

    karyawan_terpenuhi1=sum(strcmp(keahlian_karyawan,jenis_dibutuhkan{
1,1}),2);

    karyawan_terpenuhi2=sum(strcmp(keahlian_karyawan,jenis_dibutuhkan{
2,1}),2);

    karyawan_terpenuhi=find((karyawan_terpenuhi1|karyawan_terpenuhi2)=
=1);
        if ~isempty(karyawan_terpenuhi)
            if 1 == 1

                if karyawan_di_tempat(i,1)==0
                    random_pilih=randint(1,1,[1
length(karyawan_terpenuhi)]);

                karyawan_terpilih=karyawan(karyawan_terpenuhi(random_pilih));
                alur_tempat_karyawan(karyawan_terpilih,1)=tempat;
                pos_dikantor=find(karyawan_di_kantor==karyawan_terpilih);
                    karyawan_di_kantor(pos_dikantor,:)=[];

                lama_karyawan_di_kantor(pos_dikantor,:)=[];
                karyawan(karyawan_terpenuhi(random_pilih),:)=[];
                keahlian_karyawan(karyawan_terpenuhi(random_pilih),:)=[];
                    karyawan_di_tempat(i,1)=karyawan_terpilih;

                lama_karyawan_di_tempat(i,1)=lama_karyawan_di_tempat(i,1)+1;
                Xijkl(jenis_pekerjaan,karyawan_terpilih,tempat,1)=1;
                    elseif lama_karyawan_di_tempat(i,1)<3
                        random_pilih=randint(1,1,[0
length(karyawan_terpenuhi)]);
                            if random_pilih==0

                                karyawan_terpilih=karyawan_di_tempat(i,1);
                                alur_tempat_karyawan(karyawan_terpilih,1)=tempat;
                                lama_karyawan_di_tempat(i,1)=lama_karyawan_di_tempat(i,1)+1;
                                Xijkl(jenis_pekerjaan,karyawan_terpilih,tempat,1)=1;
                                    else

                                        karyawan_terpilih=karyawan(karyawan_terpenuhi(random_pilih));
                                        alur_tempat_karyawan(karyawan_terpilih,1)=tempat;
                                        pos_dikantor=find(karyawan_di_kantor==karyawan_terpilih);
                                            karyawan_di_kantor(pos_dikantor,:)=[];

                                        lama_karyawan_di_kantor(pos_dikantor,:)=[];

```



```

karyawan(karyawan_terpenuhi(random_pilih),:)=[];
keahlian_karyawan(karyawan_terpenuhi(random_pilih),:)=[];
karyawan_di_kantor=[karyawan_di_kantor;karyawan_di_tempat(i,1)];
lama_karyawan_di_kantor=[lama_karyawan_di_kantor;-1];
karyawan_di_tempat(i,1)=karyawan_terpilih;
                    lama_karyawan_di_tempat(i,1)=1;
Xijkl(jenis_pekerjaan,karyawan_terpilih,tempat,1)=1;
                    end
                    else
                        random_pilih=randint(1,1,[1
length(karyawan_terpenuhi)]);
karyawan_terpilih=karyawan(karyawan_terpenuhi(random_pilih));
alur_tempat_karyawan(karyawan_terpilih,1)=tempat;
pos_dikantor=find(karyawan_di_kantor==karyawan_terpilih);
                    karyawan_di_kantor(pos_dikantor,:)=[];
lama_karyawan_di_kantor(pos_dikantor,:)=[];
karyawan(karyawan_terpenuhi(random_pilih),:)=[];
keahlian_karyawan(karyawan_terpenuhi(random_pilih),:)=[];
karyawan_di_kantor=[karyawan_di_kantor;karyawan_di_tempat(i,1)];
lama_karyawan_di_kantor=[lama_karyawan_di_kantor;-1];
                    karyawan_di_tempat(i,1)=karyawan_terpilih;
                    lama_karyawan_di_tempat(i,1)=1;
Xijkl(jenis_pekerjaan,karyawan_terpilih,tempat,1)=1;
                    end
                    else
                        if numel(data_model_karyawan{i,1-1,1}) > 2
                            karyawan_terpilih =
data_model_karyawan{i,1-1,1};
                            data_model_karyawan{i,1,2} =
data_model_karyawan{i,1-1,2};
                            karyawan_di_tempat(i,1)=0;
                            lama_karyawan_di_tempat(i,1)=0;
                            Zikl(jenis_pekerjaan,tempat,1)=1;
                        else
                            if karyawan_di_tempat(i,1)==0
                                random_pilih=randint(1,1,[1
length(karyawan_terpenuhi)]);
karyawan_terpilih=karyawan(karyawan_terpenuhi(random_pilih));
alur_tempat_karyawan(karyawan_terpilih,1)=tempat;
pos_dikantor=find(karyawan_di_kantor==karyawan_terpilih);
                            karyawan_di_kantor(pos_dikantor,:)=[];

```

```

lama_karyawan_di_kantor(pos_dikantor,:)=[];
karyawan(karyawan_terpenuhi(random_pilih),:)=[];
keahlian_karyawan(karyawan_terpenuhi(random_pilih),:)=[];
karyawan_di_tempat(i,1)=karyawan_terpilih;
lama_karyawan_di_tempat(i,1)=lama_karyawan_di_tempat(i,1)+1;
Xijkl(jenis_pekerjaan,karyawan_terpilih,tempat,1)=1;
    elseif lama_karyawan_di_tempat(i,1)<3
        random_pilih=randint(1,1,[0
length(karyawan_terpenuhi)]);
        if random_pilih==0
karyawan_terpilih=karyawan_di_tempat(i,1);
alur_tempat_karyawan(karyawan_terpilih,1)=tempat;
lama_karyawan_di_tempat(i,1)=lama_karyawan_di_tempat(i,1)+1;
Xijkl(jenis_pekerjaan,karyawan_terpilih,tempat,1)=1;
        else
karyawan_terpilih=karyawan(karyawan_terpenuhi(random_pilih));
alur_tempat_karyawan(karyawan_terpilih,1)=tempat;
pos_dikantor=find(karyawan_di_kantor==karyawan_terpilih);
karyawan_di_kantor(pos_dikantor,:)=[];
lama_karyawan_di_kantor(pos_dikantor,:)=[];
karyawan(karyawan_terpenuhi(random_pilih),:)=[];
keahlian_karyawan(karyawan_terpenuhi(random_pilih),:)=[];
karyawan_di_kantor=[karyawan_di_kantor;karyawan_di_tempat(i,1)];
lama_karyawan_di_kantor=[lama_karyawan_di_kantor;-1];
karyawan_di_tempat(i,1)=karyawan_terpilih;
        lama_karyawan_di_tempat(i,1)=1;
Xijkl(jenis_pekerjaan,karyawan_terpilih,tempat,1)=1;
        end
    else
        random_pilih=randint(1,1,[1
length(karyawan_terpenuhi)]);
karyawan_terpilih=karyawan(karyawan_terpenuhi(random_pilih));
alur_tempat_karyawan(karyawan_terpilih,1)=tempat;
pos_dikantor=find(karyawan_di_kantor==karyawan_terpilih);
        karyawan_di_kantor(pos_dikantor,:)=[];

```

```

lama_karyawan_di_kantor(pos_dikantor,:)=[];

karyawan(karyawan_terpenuhi(random_pilih),:)=[];

keahlian_karyawan(karyawan_terpenuhi(random_pilih),:)=[];

karyawan_di_kantor=[karyawan_di_kantor;karyawan_di_tempat(i,1)];

lama_karyawan_di_kantor=[lama_karyawan_di_kantor;-1];

karyawan_di_tempat(i,1)=karyawan_terpilih;
    lama_karyawan_di_tempat(i,1)=1;

Xijkl(jenis_pekerjaan,karyawan_terpilih,tempat,1)=1;
    end
end
else
    if karyawan_di_tempat(i,1)==0
        if tempat_subkontraktor(i,1)==0
            karyawan_terpilih=['dummy'];
            data_model_karyawan{i,1,2} =
subkontraktor_tersedia(1,1);

tempat_subkontraktor(i,1)=subkontraktor_tersedia(1,1);
            subkontraktor_tersedia(1,:)=[];
        else
            karyawan_terpilih=['dummy'];
            data_model_karyawan{i,1,2} =
tempat_subkontraktor(i,1);
        end

        karyawan_di_tempat(i,1)=0;
        lama_karyawan_di_tempat(i,1)=0;
        Zikl(jenis_pekerjaan,tempat,1)=1;
        elseif lama_karyawan_di_tempat(i,1)<3
            karyawan_terpilih=karyawan_di_tempat(i,1);

alur_tempat_karyawan(karyawan_terpilih,1)=tempat;

lama_karyawan_di_tempat(i,1)=lama_karyawan_di_tempat(i,1)+1;

Xijkl(jenis_pekerjaan,karyawan_terpilih,tempat,1)=1;
    else
        if tempat_subkontraktor(i,1)==0
            karyawan_terpilih=['dummy'];
            data_model_karyawan{i,1,2} =
subkontraktor_tersedia(1,1);

tempat_subkontraktor(i,1)=subkontraktor_tersedia(1,1);
            subkontraktor_tersedia(1,:)=[];
        else
            karyawan_terpilih=['dummy'];
            data_model_karyawan{i,1,2} =
tempat_subkontraktor(i,1);
        end
    end
end

```

```

karyawan_di_kantor=[karyawan_di_kantor;karyawan_di_tempat(i,1)];

lama_karyawan_di_kantor=[lama_karyawan_di_kantor;-1];
                        karyawan_di_tempat(i,1)=0;
                        lama_karyawan_di_tempat(i,1)=0;
                        Zikl(jenis_pekerjaan,tempat,1)=1;
                        end
                    end
                    data_model_karyawan{i,1}=karyawan_terpilih;
                else
                    if lama_karyawan_di_tempat(i,1)>0

karyawan_di_kantor=[karyawan_di_kantor;karyawan_di_tempat(i,1)];
                    lama_karyawan_di_kantor=[lama_karyawan_di_kantor;-
1];
                    karyawan_di_tempat(i,1)=0;
                    lama_karyawan_di_tempat(i,1)=0;
                    end
                end
            end
            lama_karyawan_di_kantor=lama_karyawan_di_kantor+1;
        end
        min_Z=hitung_min_z(Xijkl,Zikl);
        %hitung dummy
        jumlah_dummy=0;
        jumlah_transport_dummy_dlm=0;
        jumlah_transport_dummy_luar=0;
        for i=1:64
            if i<=37
                flag=0;
                for j=1:36
                    temp_sub=data_model_karyawan{i,j};
                    if length(temp_sub)>=5
                        cek=temp_sub(1:5);
                    else
                        cek=temp_sub;
                    end
                    if strcmp(cek,'dummy')
                        if flag==0

jumlah_transport_dummy_dlm=jumlah_transport_dummy_dlm+1;
                            flag=1;
                            end
                        else
                            if flag==1;
                                flag=0;
                                jumlah_dummy=jumlah_dummy+1;

jumlah_transport_dummy_dlm=jumlah_transport_dummy_dlm+1;
                            end
                        end
                    end
                end
            else
                flag=0;
                for j=1:36
                    temp_sub=data_model_karyawan{i,j};
                    if length(temp_sub)>=5
                        cek=temp_sub(1:5);

```

```

        else
            cek=temp_sub;
        end
        if strcmp(cek, 'dummy')
            if flag==0

jumlah_transport_dummy_luar=jumlah_transport_dummy_luar+1;
                flag=1;
            end
        else
            if flag==1;
                flag=0;
                jumlah_dummy=jumlah_dummy+1;

jumlah_transport_dummy_luar=jumlah_transport_dummy_luar+1;
            end
        end
    end
end
end

%hitung trip
jumlah_transport_dlm=0;
jumlah_transport_luar=0;
for i=1:18
    alur_karyawan=alur_tempat_karyawan(i,:);
    flag=0;
    for j=1:36
        if alur_karyawan(1,j)==0
            if flag==1
                if sebelum<=9
                    jumlah_transport_dlm=jumlah_transport_dlm+1;
                else
                    jumlah_transport_luar=jumlah_transport_luar+1;
                end
                flag=1;
            end
        else
            if i==1
                sebelum=alur_karyawan(1,j);
                if alur_karyawan(1,j)<=9
                    jumlah_transport_dlm=jumlah_transport_dlm+1;
                else
                    jumlah_transport_luar=jumlah_transport_luar+1;
                end
            else
                if flag==1
                    if alur_karyawan(1,j)<=9

jumlah_transport_dlm=jumlah_transport_dlm+1;
                        else

jumlah_transport_luar=jumlah_transport_luar+1;
                                end
                            sebelum=alur_karyawan(1,j);
                            flag=0;
                        end
                    end
                end
            end
        end
    end
end
end

```

```

    end
end
min_Z=min_Z+(jumlah_transport_dlm*3000000)+(jumlah_transport_dummy
_dlm*3000000)+(jumlah_transport_luar*7000000)+(jumlah_transport_du
ummy_luar*7000000);

```

2. Evaluasi Fitness

```

function f=evaluasi_fitnes(solusi)
Xijkl=zeros(4,18,17,36);
Zikl=zeros(4,17,36);
alur_tempat_karyawan=zeros(18,36);
for l=1:36
    for i=1:64
        tempat=ceil(i/4);
        if ~isempty(solusi{i,l})
            jenis_pekerjaan=mod(i,4);
            if jenis_pekerjaan==0
                jenis_pekerjaan=4;
            end
            if ~strcmp(solusi{i,l},'dummy')
                Xijkl(jenis_pekerjaan,solusi{i,l},tempat,l)=1;
                alur_tempat_karyawan(solusi{i,l},l)=tempat;
            else
                Zikl(jenis_pekerjaan,tempat,l)=1;
            end
        end
    end
end
end
min_Z=hitung_min_z(Xijkl,Zikl);
%hitung dummy
jumlah_dummy=0;
jumlah_transport_dummy_dlm=0;
jumlah_transport_dummy_luar=0;
for i=1:64
    if i<=37
        flag=0;
        for j=1:36
            if strcmp(solusi{i,j},'dummy')
                if flag==0
                    jumlah_transport_dummy_dlm=jumlah_transport_dummy_dlm+1;
                    flag=1;
                end
            else
                if flag==1;
                    flag=0;
                    jumlah_dummy=jumlah_dummy+1;
                end
            end
        end
        jumlah_transport_dummy_dlm=jumlah_transport_dummy_dlm+1;
    end
end
else
    flag=0;
    for j=1:36
        if strcmp(solusi{i,j},'dummy')
            if flag==0

```

```

jumlah_transport_dummy_luar=jumlah_transport_dummy_luar+1;
    flag=1;
    end
    else
        if flag==1;
            flag=0;
            jumlah_dummy=jumlah_dummy+1;

jumlah_transport_dummy_luar=jumlah_transport_dummy_luar+1;
        end
    end
end
end
end

%hitung trip
jumlah_transport_dlm=0;
jumlah_transport_luar=0;
for i=1:18
    alur_karyawan=alur_tempat_karyawan(i,:);
    flag=0;
    for j=1:36
        if alur_karyawan(1,j)==0
            if flag==1
                if sebelum<=9
                    jumlah_transport_dlm=jumlah_transport_dlm+1;
                else
                    jumlah_transport_luar=jumlah_transport_luar+1;
                end
                flag=1;
            end
        else
            if i==1
                sebelum=alur_karyawan(1,j);
                if alur_karyawan(1,j)<=9
                    jumlah_transport_dlm=jumlah_transport_dlm+1;
                else
                    jumlah_transport_luar=jumlah_transport_luar+1;
                end
            else
                if flag==1
                    if alur_karyawan(1,j)<=9

jumlah_transport_dlm=jumlah_transport_dlm+1;
                    else

jumlah_transport_luar=jumlah_transport_luar+1;
                    end
                sebelum=alur_karyawan(1,j);
                flag=0;
            end
        end
    end
end
end
end

f=min_Z+(jumlah_transport_dlm*3000000)+(jumlah_transport_dummy_dlm
*3000000)+(jumlah_transport_luar*7000000)+(jumlah_transport_
dummy_luar*7000000);

```

3. Seleksi Parent

```
function [parent1,parent2]=seleksi_parent (populasi, fkromosom)
[n urut]=sort (fkromosom);
En=fkromosom (urut);
populasi=populasi (urut, :);
for i=1:size (populasi, 1)
    q (i, 1)=sum (En (1:i, 1));
end
q=q/q (end);
qparent=[];
while size (qparent, 1)~=2
    qbaru=[];
    while isempty (qbaru)
        r=rand;
        for i=2:length (q)
            if q (i-1)<r && q (i)>r
                qbaru=q (i);
            end
        end
        end
        ada=find (qparent==qbaru);
        if isempty (ada)
            qparent=[qparent; qbaru];
        end
    end
    pos1=find (q==qparent (1));
    parent1=populasi {pos1, 1};
    pos2=find (q==qparent (2));
    parent2=populasi {pos2, 1};
end
```

3. Crossover

```
function [hasil1, hasil2]=order_crossover (solusi1, solusi2)
posisi_random=randint (1, 1, [1 length (solusi1)]);
hasil1=[solusi2 (1:posisi_random, :); solusi1 (posisi_random+1:end, :)]
;
hasil2=[solusi1 (1:posisi_random, :); solusi2 (posisi_random+1:end, :)]
;
```

4. Mutasi

```
function out=mutasi (in)
byk=randint (1, 1, [1 64]);
out=in;
for j=1:byk
    pos=randint (1, 1, [1 size (in, 1)]);
    temp=in (pos, :);
    posisi=[];
    for i=1:length (temp)
        if ~isempty (temp {1, i})
            posisi=[posisi; i];
        end
    end
    end
    pos1=randint (1, 1, [1 length (posisi)]);
    pos2=randint (1, 1, [1 length (posisi)]);
    while pos1==pos2
        pos2=randint (1, 1, [1 length (posisi)]);
    end
```



```

end
a=out{1,pos1};
out{1,pos1}=out{1,pos2};
out{pos,pos2}=a;
end

```

5. Hitung Biaya

```

function min_Z=hitung_min_z(Xijkl,Zikl)
min_Z=0;
for i=1:4
    for j=1:18
        for k=1:16
            for l=1:36
                if k<10
                    min_Z=min_Z+30000000*Xijkl(i,j,k,l);
                else
                    min_Z=min_Z+43200000*Xijkl(i,j,k,l);
                end
            end
        end
    end
end
for i=1:4
    for k=1:16
        for l=1:36
            if k<10
                min_Z=min_Z+50000000*Zikl(i,k,l);
            else
                min_Z=min_Z+50000000*Zikl(i,k,l);
            end
        end
    end
end
end

```

6. Pemilihan Solusi

```

function [xbest,fbest,f_rata2,f,totalwaktu]=al_gen(Pc,Pm,maximp)
tic
uk_populasi=80;
[kromosom,fkromosom]=bangkit_populasi(uk_populasi);
for iterasi=1:maximp
    [parent1,parent2]=seleksi_parent(kromosom,fkromosom);
    if rand < Pc
        [anak1,anak2]=order_crossover(parent1,parent2);
    else
        anak1=parent1;
        anak2=parent2;
    end
    if rand<Pm
        anak1=mutasi(anak1);
        anak2=mutasi(anak2);
    end
    f_anak1=evaluasi_fitnes(anak1);
    f_anak2=evaluasi_fitnes(anak2);
    if f_anak1<fkromosom(end)
        kromosom{end,:}=anak1;
    end
end

```

```

        fkromosom(end,:)=f_anak1;
        [fkromosom,index]=sort(fkromosom);
        kromosom=kromosom(index,:);
    end
    if f_anak2<fkromosom(end)
        kromosom{end,:}=anak2;
        fkromosom(end,:)=f_anak2;
        [fkromosom,index]=sort(fkromosom);
        kromosom=kromosom(index,:);
    end
    if f_anak1<f_anak2
        fnew=f_anak1;
        xnew=anak1;
    else
        fnew=f_anak2;
        xnew=anak2;
    end
    if iterasi==1
        xbest=xnew;
        fbest=fnew;
    else
        if fnew<fbest
            xbest=xnew;
            fbest=fnew;
        end
    end
    f(iterasi)=fbest
    f_rata2(iterasi)=mean(fkromosom);
    iterasi
end
totalwaktu=toc;

```

7. Program Utama

```

%Program utama
% clc,clear all,close all;
Pc=0.45;
Pm=0.01; %Pm nya pake rumus 10/jumlah gen
maximp=1;
[xbest,fbest,f_rata2,f,totalwaktu]=al_gen(Pc,Pm,maximp);
totalwaktu
figure,plot(f,'k');xlabel('Iterasi');ylabel('fitnes');legend('Fit
nes Terbaik');

```