



**UNIVERSITAS INDONESIA**

**PERANCANGAN SISTEM KESEIMBANGAN LINI  
PERAKITAN HYDRAULIC EXCAVATOR TIPE PC300  
DENGAN METODE ALGORITMA GENETIKA**

**SKRIPSI**

**ANDRI AMIR  
0606043944**

**FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INDUSTRI  
JAKARTA  
DESEMBER 2008**



**UNIVERSITAS INDONESIA**

**PERANCANGAN SISTEM KESEIMBANGAN LINI  
PERAKITAN HYDRAULIC EXCAVATOR TIPE PC300  
DENGAN METODE ALGORITMA GENETIKA**

**SKRIPSI**

**diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana teknik**

**ANDRI AMIR  
0606043944**

**FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INDUSTRI  
JAKARTA  
DESEMBER 2008**

## HALAMAN PERNYATAAN ORISINALITAS

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

**Nama : Andri Amir**

**NPM : 0606043944**

**Tanda Tangan :**

**Tanggal : 23 Desember 2008**

## PERSETUJUAN

Skripsi dengan Judul :

**“Perancangan Sistem Keseimbangan Lini Perakitan *Hydraulic Excavator*  
Tipe PC300-8 Dengan Menggunakan Metode Algoritma Genetika”**

Dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Industri Jurusan Teknik Industri Fakultas Teknik Universitas Indonesia dan disetujui untuk diajukan dalam sidang ujian skripsi.

Depok, 23 Desember 2008

Ir. Amar Rachman, MEIM

NIP. 130 702 238

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Andri Amir  
NPM : 0606043944  
Program Studi : Teknik Industri  
Judul Skripsi : Perancangan Sistem Keseimbangan Lini Perakitan  
*Hydraulic Excavator Tipe PC300-8* Dengan  
Menggunakan Metode Algoritma Genetika.

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

### DEWAN PENGUJI

Pembimbing : Ir. Amar Rachman, MEIM (.....)  
Penguji : Ir. Yadrifil, MSc (.....)  
Penguji : Ir. Akhmad Hidayatno, MBT (.....)  
Penguji : Armand Omar Moeis, ST, MSc (.....)

Ditetapkan di : .....

Tanggal : .....

## UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat mencapai gelar Sarjana Teknik Jurusan Teknik Industri pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Bpk Ir. Amar Rachman, MEIM selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran di dalam mengarahkan penulis dalam menyusun skripsi ini.
2. Orangtua dan keluarga saya yang telah memberikan bantuan dukungan material dan moril.
3. Assembly Team PT Komatsu Indonesia yang telah membantu dalam proses pengambilan data.
4. Rekan-rekan Ekstensi Teknik Industri 2006 yang telah banyak membantu penulis dalam menyelesaikan skripsi ini.
5. Mbak Fatimah dan Mas Dodi, yang selalu direpotkan dengan urusan perkuliahan.
6. Pihak-pihak lain yang tidak mungkin disebutkan satu persatu.

Akhir kata, penulis berharap semoga Allah SWT berkenan membalas segala kebaikan saudara-saudara semua. Dan semoga skripsi ini membawa manfaat bagi pengembangan ilmu pengetahuan.

Jakarta, 23 Desember 2008

Penulis

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

---

---

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

Nama : Andri Amir  
NPM : 0606043944  
Program Studi : Teknik Industri  
Fakultas : Teknik  
Jenis karya : Skripsi

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

Perancangan Sistem Keseimbangan Lini Perakitan *Hydraulic Excavator Tipe PC300-8* Dengan Menggunakan Metode Algoritma Genetika.

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/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.

Demikianlah pernyataan ini saya buat dengan sebenarnya,

Dibuat di : Jakarta  
Pada tanggal : 23 Desember 2008

Yang menyatakan

(Andri Amir)

## ABSTRAK

Nama : Andri Amir  
Program Studi : Teknik Industri  
Judul : Perancangan Sistem Keseimbangan Lini Perakitan *Hydraulic Excavator Tipe PC300-8* Dengan Menggunakan Metode Algoritma Genetika.

Keseimbangan lini lintasan perakitan merupakan salah satu masalah penting di area produksi atau area manajemen operasi. Karena keseimbangan dalam penempatan elemen-elemen kerja dan beban kerja pada tiap stasiun kerja akan memberikan pengaruh yang besar pada performa sistem, maka sangat penting untuk mengembangkan suatu solusi yang praktis dari permasalahan keseimbangan lini dan juga kebutuhan akan waktu perhitungan yang minimal untuk memecahkan masalah keseimbangan lini tersebut. Metode heuristik adalah salah satu cara yang umum digunakan untuk memecahkan masalah keseimbangan lini perakitan ini. Dalam penelitian ini, metode heuristik yang efisien digunakan untuk memecahkan masalah keseimbangan lini perakitan dengan pendekatan deterministik dan model tunggal. Metode heuristik tersebut adalah metode heuristik Algoritma Genetika. Algoritma Genetika yang digunakan untuk memecahkan masalah keseimbangan lini lintasan perakitan ini dibentuk dengan struktur kromosom yang khusus. Pada penelitian ini keseimbangan lini perakitan yang dihasilkan dengan menggunakan metode Algoritma Genetika akan dibandingkan dengan keseimbangan lini perakitan lama yang telah diterapkan, dan terbukti menghasilkan keseimbangan lini perakitan yang lebih baik. Dengan penurunan jumlah stasiun kerja, efisiensi lintasan serta nilai *smoothness* yang meningkat, penggunaan Algoritma Genetika mampu memberikan solusi yang optimal untuk memecahkan permasalahan keseimbangan lini perakitan. Oleh karena itu, pada penelitian ini metode Algoritma Genetika juga digunakan dalam menghitung keseimbangan lini dengan waktu siklus yang akan dicapai yaitu menghasilkan sistem keseimbangan lini perakitan *Hydraulic Excavator tipe PC300-8* untuk waktu siklus (*cycle time*) 38.33 menit dengan jumlah stasiun kerja sebanyak 13 stasiun kerja, efisiensi lintasan sebesar 0,8028, dan nilai *smoothness* sebesar 37,8320 detik.

Kata kunci: lini perakitan, keseimbangan lini perakitan, algoritma genetika

## ABSTRACT

Name : Andri Amir

Major : Industrial Engineering

Title : Assembly Line Balancing Design on Hydraulic Excavator Tipe PC300-8 using Genetic Algorithm.

Assembly Line Balancing is one of the most important issue in operation management area. Balancing in assigning task and work load in every workstation will give big influence in assembling system performance and can affect positive to finance point of view, hence of vital importance to develop the practical solution to solve assembly line balancing problems as well as requirement of minimum calculation time, meta-heuristic method is one of most often used to solve assembly line balancing problem. In this research, meta-heuristic method will be used to solve an assembly line balancing problem with approach of single model and deterministic which called by Genetic Algorithm. In this research, result that achieved in balancing of assembly line by using Genetic Algorithm method will be compared with the former balancing of line which have been applied, and its proven better line characteristic. With decreasing sum of workstation among the assembly line, as well as smoothness index and also increasing line efficiency, using Genetic Algorithm can give optimal solution to solve assembly line balancing problems. Therefore, continuing in this research, Genetic Algorithm method also applied in balancing the new assembly line with cycle time to be reached. And its proven balancing of Hydraulic Excavator type of PC300-8 Assembly line for cycle time of 38.33 minutes with 13 Workstations amount, and 0.8028 line efficiency.

Keywords: assembly system, assembly line balancing, genetic algorithm

## DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS .....	ii
LEMBAR PENGESAHAN .....	iii
RIWAYAT HIDUP PENULIS .....	iv
UCAPAN TERIMA KASIH .....	v
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH .....	vi
ABSTRAK .....	vii
ABSTRAC .....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xii
<b>1. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Diagram Keterkaitan .....	2
1.3 Perumusan Masalah .....	4
1.4 Tujuan Penelitian .....	4
1.5 Batasan Masalah.....	4
1.6 Metodologi Penelitian .....	5
1.7 Sistematika Penulisan.....	6
<b>2. DASAR TEORI .....</b>	<b>8</b>
2.1 Lini Perakitan .....	8
2.1.1 Klasifikasi Lini Perakitan .....	8
2.2 Keseimbangan Lini Perakitan .....	10
2.2.1 Kategori Permasalahan Keseimbangan Lini Perakitan.....	11
2.2.2 Langkah-langkah dalam Penyeimbangan Lini Perakitan .....	12
2.2.3 Metode Penyelesaian Keseimbangan Lini Perakitan .....	14
2.3 Algoritma Genetika .....	18
2.3.1 Definisi Algoritma Genetika .....	18
2.3.2 Aplikasi Algoritma Genetika.....	19
2.3.3 Struktur Dasar Algoritma Genetika Masalah Keseimbangan Lini	20
2.3.4 Komponen-Komponen Algoritma Genetika.....	21
<b>3. PENGUMPULAN DATA .....</b>	<b>25</b>
3.1 Profil Perusahaan .....	25
3.2 Data Penelitian .....	25
3.2.1 Data Waktu Kerja Bagian Perakitan.....	26
3.2.2 Elemen Kerja dan Waktu Pengerjaan Tiap Elemen Kerja.....	27
3.2.3 Diagram Keterkaitan ( <i>Precedence Diagram</i> ). .....	30
3.2.4 Keseimbangan Lini Perakitan saat ini .....	32
<b>4. PENGOLAHAN DATA DAN ANALISA .....</b>	<b>36</b>
4.1 Penyusunan Algoritma.....	36

4.1.1 Algoritma Genetika Pada Keseimbangan Lini Perakitan .....	36
4.1.2 Penerapan Algoritma Genetik Pada Keseimbangan Lini Perakitan <i>Hydraulic Excavator Tipe PC300-8</i> .....	39
4.2 Validasi Program .....	41
4.3 Hasil Dan Analisa .....	43
4.3.1 Hasil Running Program .....	43
4.3.2 Analisa .....	45
<b>5. KESIMPULAN.....</b>	<b>53</b>
<b>DAFTAR REFERENSI</b>	



## DAFTAR GAMBAR

Gambar 1.1 Diagram Keterkaitan Masalah.....	3
Gambar 1.2 Diagram Alir Metodologi Penelitian.....	6
Gambar 2.1 Contoh lini perakitan.....	8
Gambar 2.2 Penyeimbangan lini perakitan.....	10
Gambar 2.3 Klasifikasi permasalahan keseimbangan lini perakitan.....	12
Gambar 2.5 <i>Precedence Diagram</i> .....	13
Gambar 2.5 Prosedur dasar pada Algoritma Genetika.....	21
Gambar 3.1. Lini perakitan <i>Hydraulic Excavator Tipe PC300-8</i> .....	26
Gambar 3.2 Grafik waktu siklus tiap stasiun kerja pada lini perakitan <i>Hydraulic Excavator tipe PC300-8</i> .....	34
Gambar 4.1 Standard Procedure Algoritma Genetika.....	37
Gambar 4.2 Flowchart Algoritma Genetika.....	37
Gambar 4.3 Populasi awal yang layak.....	40
Gambar 4.4 Parameter <i>TaskTime</i> .....	40



## DAFTAR TABEL

Tabel 3.1 Waktu perakitan yang tersedia dalam satu hari kerja .....	27
Tabel 3.2 Elemen kerj lini perakitan <i>Hydraulic Excavator tipe PC300-8</i> .....	28
Tabel 3.3 waktu penyelesaian elemen kerja lini perakitan <i>Hydraulic Excavator tipe PC300-8</i> .....	30
Tabel 3.4 Keterkaitan antar elemen kerja lini perakitan <i>Hydraulic Excavator tipe PC300-8</i> .....	31
Tabel 3.5 Penempatan elemen-elemen kerja pada stasiun kerja di lini perakitan <i>Hydraulic Excavator tipe PC300-8</i> .....	32
Tabel 3.6 Waktu siklus pada tiap stasiun kerja lini perakitan <i>Hydraulic Excavator tipe PC300-8</i> .....	34
Tabel 4.1 <i>Validasi</i> perhitungan manual .....	42
Tabel 4.2 Perbandingan keseimbangan lini perakitan <i>Hydraulic Excavator tipe PC300-8</i> lama Vs baru .....	46
Tabel 4.3 Keseimbangan lini perakitan lama.....	47
Tabel 4.4 Keseimbangan lini perakitan baru (GA).....	48
Tabel 4.5 Perbandingan keseimbangan lini perakitan dengan berbagai varisi waktu siklus.....	50
Tabel 4.6 <i>Gap</i> antara waktu siklus lini perkitan dengan waktu siklus max yang ada pada stasiun kerja.....	51
Tabel 4.7 Effisiensi lini perakitan tiap waktu siklus.....	52

## DAFTAR RUMUS

Persamaan 2.1 <i>Cycle Time</i> .....	13
Persamaan 2.2 Jumlah Minimum Stasiun Kerja .....	13
Persamaan 2.3 Efisiensi Lini Perakitan .....	14
Persamaan 2.4 <i>Smoothnessindex</i> .....	14
Persamaan 2.5 <i>Fitness Function</i> .....	22



## DAFTAR LAMPIRAN

Lampiran 1 : *M-Skript* Program Algoritma Genetika Keseimbangan Lini Perakitan *Hydraulic Excavator PC300-8*.

Lampiran 2 : Hasil perhitungan Keseimbangan Lini Perakitan Dengan Menggunakan Program Algoritma Genetika.

Lampiran 3 : Diagram Keterkaitan (*Precedence Diagram*) Elemen Kerja Perakitan *Hydraulic Excavator PC300-8*.



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Pertumbuhan perekonomian sebuah negara merupakan salah satu tolak ukur kemajuan suatu bangsa dari sudut pandang ekonomi, setiap negara berusaha untuk selalu meningkatkan pertumbuhan ekonomi bangsanya. Seiring dengan pertumbuhan perekonomian tersebut, sektor-sektor industri yang menjadi penopang pertumbuhan tersebut bergerak cepat dengan berbagai kebijakan-kebijakan yang dikeluarkan oleh pemerintah untuk menjamin pertumbuhan perekonomian berada pada jalur yang telah ditetapkan.

Indonesia sebagai negara berkembang dengan pertumbuhan perekonomian yang terus meningkat dari tahun ketahun menjadikan berbagai sektor industri yang ada mengalami kenaikan pertumbuhannya. Seiring dengan pertumbuhan industri tersebut, permintaan akan alat-alat berat yang merupakan peralatan utama dalam peningkatan produksi pada berbagai sektor industri mengalami peningkatan permintaan yang cukup besar. Produsen alat-alat berat, menerapkan lini perakitan (*assembly line*) dalam proses produksinya. Seiring dengan bertambahnya permintaan alat-alat berat yang memberikan dampak positif pada perusahaan yaitu bertambahnya pelanggan yang mempercayakan pemenuhan atas kebutuhan mereka akan alat-alat berat. Namun demikian, muncul beberapa permasalahan terkait dengan lini perakitan yang ada saat ini. Sulitnya melakukan keseimbangan lini perakitan yang telah ada untuk dapat memenuhi target produksi yang dibebankan.

Penerapan sebuah lini perakitan yang seimbang dan efisien sangat dibutuhkan agar dapat meminimalisasikan ketidakseimbangan antara ketersediaan *resource* dengan *output* yang diharapkan pada lini tersebut<sup>1</sup>. Selain itu penyeimbangan lini perakitan dilakukan dengan beberapa tujuan yang akan dicapai yaitu<sup>2</sup>

---

<sup>1</sup> Heizer J, Render B, "Principles of Operations Management", Pearson Education International, Sixth Edition.

<sup>2</sup> Esmaeilian G.R., MAhmad M.M.H, Sulaiman S, Ismail N, "Assembly Line and Balancing Assembly Line"

- meminimalkan waktu siklus pada lini perakitan.
- meminimalkan jumlah stasiun kerja.
- Memaksimalkan beban kerja setiap stasiun kerja.
- Memaksimalkan relasi antar elemen kerja.

Dan masih banyak lagi tujuan yang hendak dicapai pada penyeimbangan lini perakitan.

Masalah keseimbangan pada lini perakitan adalah permasalahan *combinatorial optimization* yang kompleks<sup>3</sup> dikategorikan sebagai permasalahan *nondeterministic polynomial hard (NP-hard)*, yaitu permasalahan dimana waktu penyelesaian pencarian solusinya akan meningkat secara eksponensial seiring dengan semakin besarnya ukuran permasalahan.

Permasalahan keseimbangan pada lini perakitan dapat diselesaikan dengan metode *exact, non exact/heuristic* maupun dengan metode *metaheuristic*. Metode *metaheuristic* merupakan metode pencarian yang dirancang untuk menghasilkan solusi yang lebih baik, cepat, dan murah. Salah satu metode *metaheuristic* ini adalah algoritma genetika (*Genetic Algorithm*).

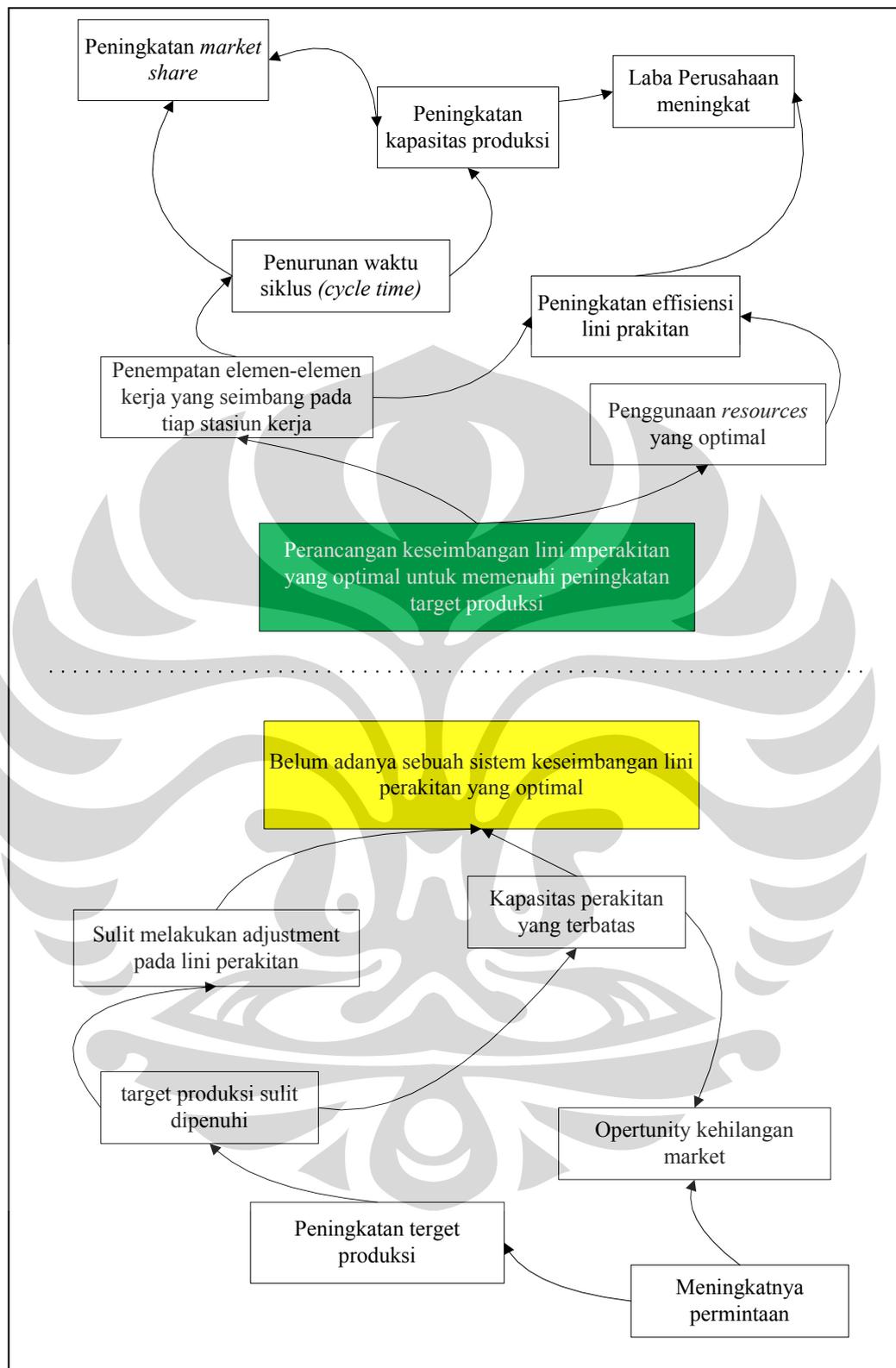
Algoritma Genetika menggunakan proses genetika pada organisme biologi dalam pemecahan permasalahan optimasi<sup>4</sup>. Individu dengan kemampuan bertahan yang lebih baik akan mengalahkan individu yang lemah. Individu tersebut akan mewariskan gen-genya kepada keturunannya sehingga generasi yang terbentuk setelahnya akan mewarisi kekuatan serta ketangguhannya. Prinsip inilah yang digunakan pada pemecahan permasalahan optimasi salah satunya yaitu permasalahan keseimbangan lini perakitan.

## 1.2. Diagram Keterkaitan Masalah

Untuk memberikan gambaran yang lebih menyeluruh atas gejala-gejala permasalahan yang ada serta guna mendapatkan akar permasalahan serta solusi yang mungkin timbul, maka disusun sebuah diagram keterkaitan masalah seperti terlihat pada gambar 1.1

<sup>3</sup> Nearchou Andreas C, "Balancing large assembly lines by a new heuristic based on differential evolution method", Springer-Verlag London Limited, 2006

<sup>4</sup> Sabuncuoglu, Erel E, Tanyer M, "Assembly line balancing using genetic algorithms" Journal of Intelligent Manufacturing (2000) 11, 295-310



**Gambar 1.1.** Diagram Keterkaitan Masalah

### 1.3. Perumusan Masalah

Seiring dengan peningkatan pertumbuhan perekonomian yang menjadi salah satu indikator pertumbuhan sektor-sektor industri yang ada, maka tingkat kebutuhan akan alat-alat berat pun meningkat. Peningkatan kebutuhan akan alat-alat berat yang menjadi investasi utama pada berbagai sektor industri terjadi secara signifikan dan menjadikan sebuah permasalahan bagi industri perakitan alat-alat berat untuk memenuhi kebutuhan tersebut.

Untuk memenuhi permintaan yang meningkat, sebuah lini perakitan dituntut untuk dapat beradaptasi dengan cepat. Permasalahan yang timbul dari peningkatan kapasitas sebuah lini perakitan adalah permasalahan keseimbangan pada lini perakitan tersebut. Peningkatan kapasitas berarti penurunan waktu siklus (*cycle time*) yang ada pada lini perakitan tersebut. Penurunan waktu siklus dapat dilakukan dengan peningkatan jumlah stasiun kerja serta pembagian beban kerja yang seimbang pada setiap stasiun kerja (*workstation*) agar waktu siklus yang dikehendaki dapat dicapai.

Oleh karena itu perlu dibuatnya sebuah system optimasi yang dapat melakukan penyeimbangan lini perakitan yang diterapkan saat ini serta melakukan penyeimbangan lini perakitan yang dapat memenuhi target produksi yang meningkat.

### 1.4. Tujuan Penelitian

Atas permasalahan diatas, maka penelitian ini ditujukan untuk mendapatkan suatu sistem keseimbangan lini perakitan yang optimal dengan menerapkan Algoritma Genetika (*Genetic Algorithm*) dengan fungsi tujuan untuk meminimalkan jumlah stasiun kerja.

### 1.5. Batasan Masalah

Untuk mendapatkan hasil penelitian yang lebih spesifik dan terarah, maka lingkup penelitian didasarkan atas beberapa batasan. Yaitu :

- a. Penelitian dilakukan pada lini perakitan *Hydraulic Excavator tipe PC300-8*.

- b. Keseimbangan lini perakitan yang dilakukan memperhitungkan *precedences* dan *predecessor* tiap elemen kerja.
- c. Perhitungan keseimbangan lini perakitan pada metode algoritma genetika dilakukan dengan menggunakan *software* MATLAB.
- d. Setiap elemen kerja diselesaikan pengerjaannya dalam satu stasiun kerja.
- e. Beberapa elemen kerja yang dilakukan pengerjaannya pada sebuah stasiun kerja merupakan beban kerja stasiun kerja tersebut yang direpresentasikan dengan waktu siklus stasiun kerja tersebut.
- f. Beban kerja sebuah stasiun kerja tidak boleh melebihi waktu siklus yang telah ditentukan.

### 1.6. Metodologi Penelitian

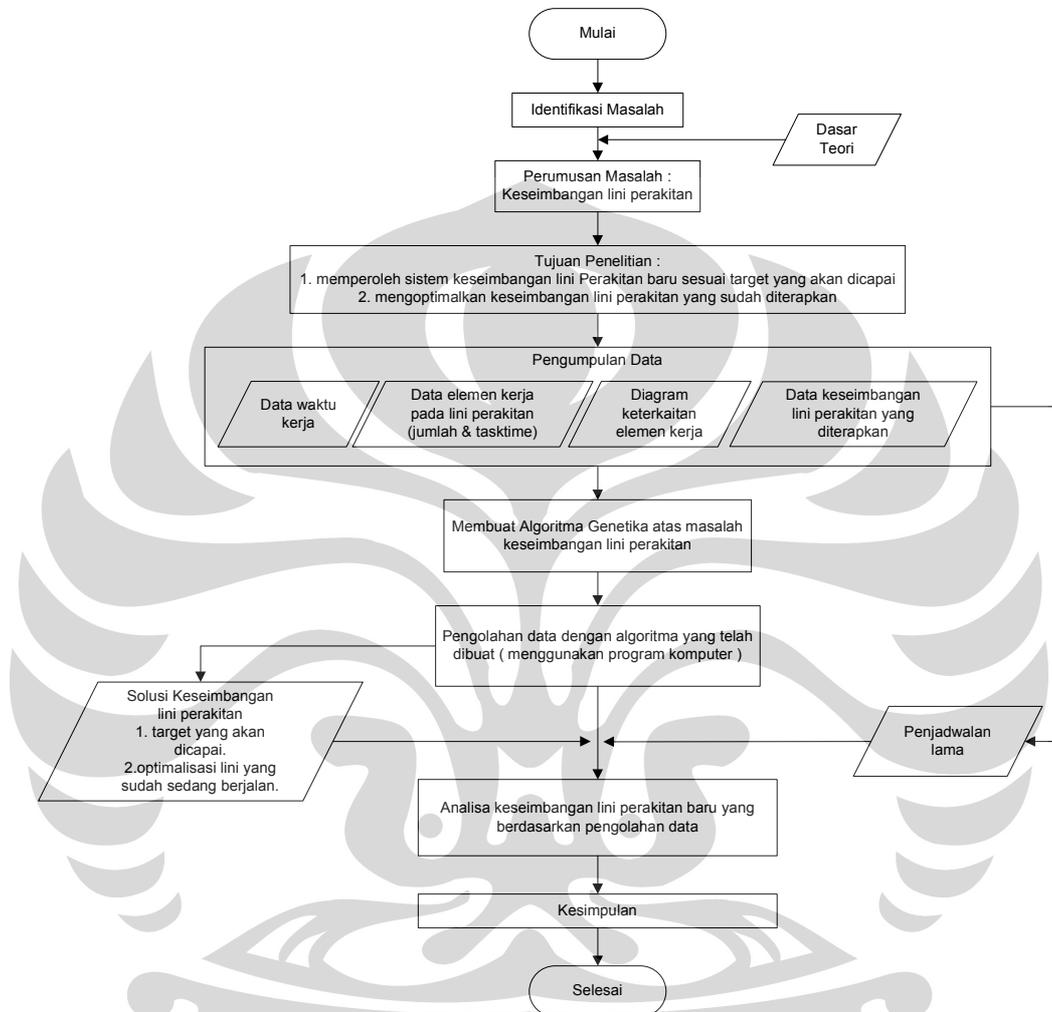
Berikut adalah langkah-langkah metodologi yang digunakan dalam penelitian ini, sebagaimana tergambar pada diagram alir dari metodologi penelitian (gambar 1.2):

- a. Melakukan identifikasi permasalahan.
- b. Mengumpulkan dan menyusun studi literatur yang berkaitan dengan masalah yang telah diidentifikasi.
- c. Merumuskan masalah, yaitu perlunya perancangan keseimbangan lini perakitan.
- d. Menentukan tujuan, yaitu memperoleh keseimbangan lini perakitan yang optimal untuk mencapai target produksi yang meningkat.
- e. Mengidentifikasi data yang dibutuhkan dan selanjutnya mengumpulkan data sekunder perusahaan.
- f. Membuat algoritma untuk menyelesaikan permasalahan keseimbangan lini produksi.
- g. Melakukan validasi dan verifikasi terhadap program yang telah dibuat.
- h. Menganalisis solusi keseimbangan lini perakitan baru untuk memenuhi peningkatan kapasitas produksi yang dihasilkan oleh program dan membandingkan keseimbangan lini perakitan yang

**Universitas Indonesia**

dihasilkan oleh program dengan keseimbangan lini perakitan saat ini.

- i. Menarik kesimpulan berdasarkan hasil analisis tersebut.



**Gambar 1.2.** Diagram Alir Metodologi Penelitian

### 1.7. Sistematika Penulisan

Penulisan laporan penelitian ini dibagi menjadi lima bab.

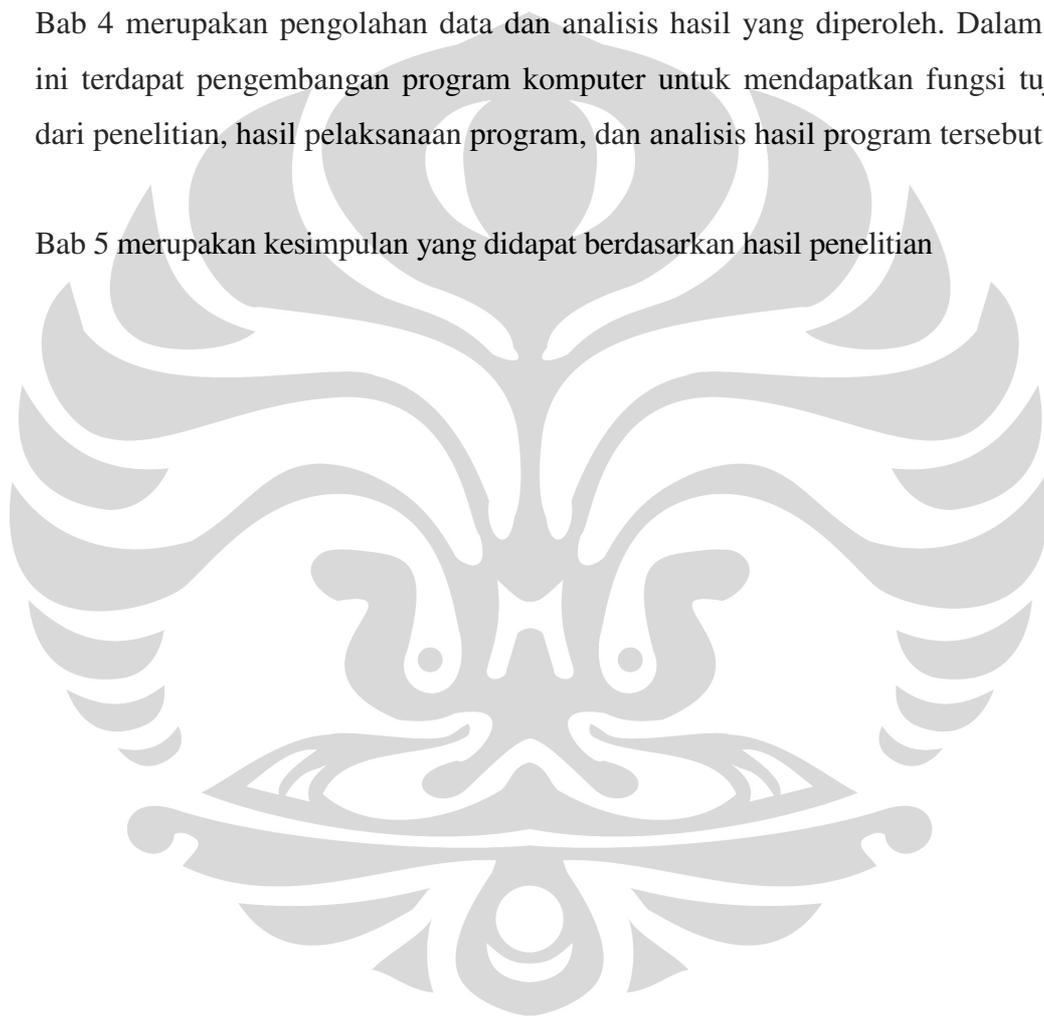
Bab 1 merupakan bab pendahuluan, menjelaskan mengenai latar belakang permasalahan, diagram yang menggambarkan keterkaitan masalah, perumusan masalah, tujuan penelitian yang ingin dicapai, batasan masalah yang dilakukan, metodologi penelitian yang dilakukan oleh penulis, serta sistematika penulisan.

Bab 2 yang merupakan bab landasan teori berisikan mengenai teori-teori yang berkaitan dengan penjadwalan produksi dan algoritma genetika.

Bab 3 merupakan bab pengumpulan data, menjelaskan mengenai data yang diambil oleh penulis selama penelitian yang akan dijadikan input dalam pengolahan data yang dilakukan pada tahap selanjutnya.

Bab 4 merupakan pengolahan data dan analisis hasil yang diperoleh. Dalam bab ini terdapat pengembangan program komputer untuk mendapatkan fungsi tujuan dari penelitian, hasil pelaksanaan program, dan analisis hasil program tersebut.

Bab 5 merupakan kesimpulan yang didapat berdasarkan hasil penelitian

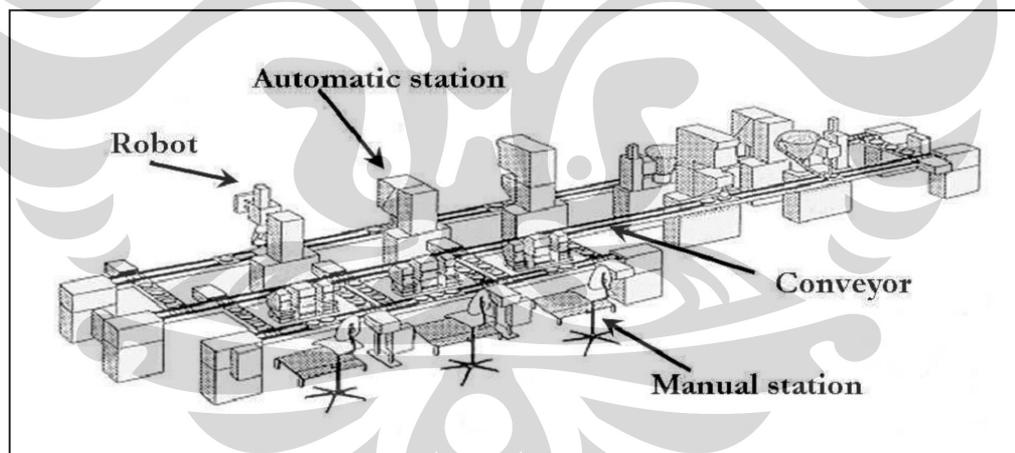


## BAB II DASAR TEORI

### 2.1. Lini Perakitan (*Assembly Line*)

Lini Perakitan adalah sebuah teknik dalam sistem manufaktur dimana sebuah produk dihasilkan dengan mengabungkan beberapa komponen dan subkomponen dalam suatu system ban berjalan yang melalui beberapa stasiun kerja.

Lini perakitan dapat diartikan juga sebagai rangkaian beberapa stasiun kerja (*workstation*) yang terhubung oleh sebuah *material handling system* seperti *conveyor*. Tiap stasiun kerja berisikan satu atau lebih pekerjaan perakitan yang dikerjakan oleh operator, *mechine* maupun robot. Setelah melewati waktu yang telah ditentukan pada suatu stasiun kerja, *conveyor* akan bergerak sehingga posisi produk yang akan dirakit berada pada stasiun kerja berikutnya dan siap untuk dilanjutkan dengan pekerjaan perakitan pada stasiun tersebut. Gambar 2.1 mengilustrasikan sebuah lini perakitan.



Gambar 2.1. Contoh lini perakitan

#### 2.1.1. Klasifikasi Lini Perakitan

Lini perakitan diklasifikasikan menjadi :

1. Berdasarkan jumlah stasiun kerja yaitu :

- *Single-station assembly*.

Yaitu sebuah lini perakitan yang hanya terdiri satu stasiun kerja. Proses perakitan dilakukan pada satu stasiun kerja, komponen

maupun subperakitan (*subassembly*) bergerak menuju satu tempat. Model perakitan ini dilakukan pada perakitan produk yang cukup kompleks dan dengan jumlah yang sedikit.

- *Multi-station assembly line*

Yaitu lini perakitan yang terdiri atas beberapa stasiun kerja. Satu stasiun pada lini perakitan kerja melakukan pekerjaan-pekerjaan yang sama dan berulang-ulang sehingga operator menjadi terlatih dan pekerjaan dapat dilakukan dengan waktu pengerjaan yang lebih baik.

Model lini perakitan ini sangat tepat untuk diterapkan pada perakitan satu jenis produk dengan jumlah produksi yang cukup banyak.

2. Berdasarkan proses perakitanannya :

- *Manual assembly system*

Yaitu lini perakitan yang dengan terdiri atas beberapa stasiun kerja yang berisikan satu atau lebih operator yang bertugas melakukan pekerjaan-pekerjaan perakitan

- *Automated assembly system*

Yaitu lini perakitan yang menggunakan mesin-mesin yang bekerja secara otomatis dalam melakukan proses-proses perakitan.

3. Berdasarkan cara perpindahan komponen :

- *Nonmechanical Lines*

Pada lini perakitan ini, komponen maupun subperakitan bergerak secara manual yaitu dengan menggunakan tangan.

- *Moving Conveyor Line*

Komponen dan subperakitan (*subassembly*) bergerak dengan menggunakan sebuah material handling system yaitu dengan menggunakan *conveyor*.

4. Berdasarkan variasi produk

- *Single Model Line*

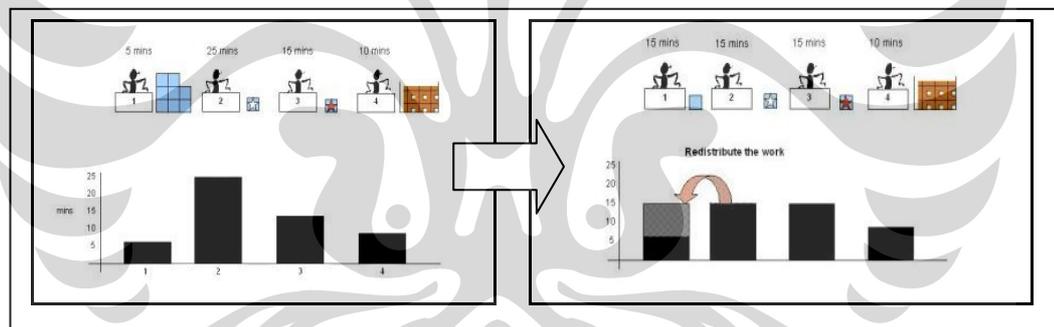
Yaitu lini perakitan yang dirancang untuk merakit satu jenis produk.

**Universitas Indonesia**

- *Batch-model Line*  
Lini perakitan yang digunakan untuk merakit dua atau lebih model produk dengan urutan proses yang sama satu dengan lainnya.
- *Mixed-model Line*  
Lini perakitan yang digunakan untuk proses perakitan beberapa model produk dengan urutan proses yang berbeda digabungkan secara simultan pada satu lini perakitan.

## 2.2. Keseimbangan Lini Perakitan

Menurut Stevenson (2002), keseimbangan lini perakitan adalah suatu proses penempatan elemen kerja (*task*) pada suatu lini perakitan kedalam stasiun-stasiun kerja (*workstation*) sehingga pada setiap stasiun kerja dihasilkan waktu penyelesaian elemen kerja yang sama satu dengan lainnya<sup>5</sup>. Gambar 2.2 menggambarkan contoh sederhana dari proses penyeimbangan lini perakitan.



**Gambar 2.2** Penyeimbangan lini perakitan

Keseimbangan lini perakitan juga digunakan untuk membuat aliran komponen maupun produk setengah jadi pada lini perakitan berada dalam kondisi lancar. Selain itu, juga bertujuan untuk mendapatkan utilisasi yang maksimal dari pekerja maupun kapasitas perakitan yang diinginkan.

<sup>5</sup>Yahya N. M, Mohamad W.M.W, “An Overview Of Assembly Line Balancing”

### 2.2.1. Kategori permasalahan Keseimbangan Lini perakitan

Menurut Scholl and Becker (2006) permasalahan pada keseimbangan lini perakitan terbagi kepada beberapa kriteria terkait dengan kendala serta tujuan yang hendak dicapai pada lini perakitan tersebut. Pengklasifikasian tersebut antara lain :

#### 1. SALBP (*Simple assembly line balancing problem*)

Adalah permasalahan penyeimbangan untuk lini perakitan yang lurus, dimana kendala urutan pengerjaan antar element kerja (*precedence constraint*) yang menjadi objek utama dalam langkah penyeimbangan lini perakitan tersebut.

Ditinjau dari tujuan penyeimbangan lini perakitan, SALBP terbagi dalam beberapa klasifikasi yaitu :

- SALBP-1

Penyeimbangan lini dengan pengaturan element kerja untuk meminimalkan jumlah stasiun kerja (*workstation*) pada sebuah waktu siklus kerja (*cycle time*) yang dikehendaki.

- SALBP-2

Pada SALBP-2, tujuan yang hendak dicapai dari penyeimbangan lini adalah untuk menurunkan waktu siklus kerja (*cycle time*) sebuah lini dengan jumlah stasiun kerja (*workstation*) yang tetap yaitu dengan meminimalkan jumlah waktu kosong (*idle time*).

- SALBP-E

Tujuan yang hendak dicapai dalam SALBP-E adalah untuk memaksimalkan efisiensi lini perakitan yaitu meminimalkan waktu siklus kerja (*cycle time*) dan stasiun kerja (*workstation*) dengan mempertimbangkan hubungan diantaranya.

- SALBP-F

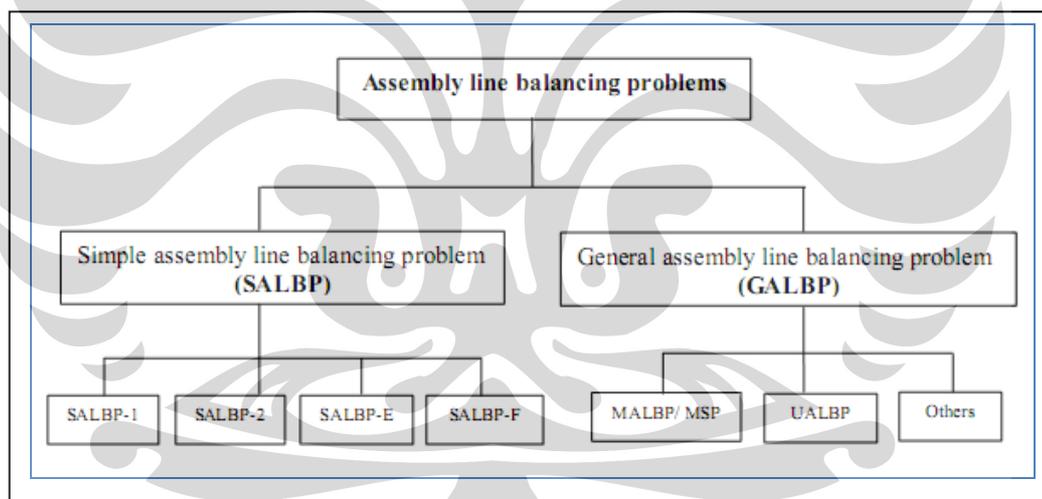
Tujuannya yang hendak dicapai adalah menentukan kelayakan sebuah lini perakitan.

## 2. GALBP (*General Assembly Line Balancing Problem*).

Adalah permasalahan penyeimbangan lini perakitan yang lebih luas cakupan permasalahannya dengan kendala keseimbangan yang kompleks.

GALBP terbagi dalam beberapa klasifikasi, yaitu :

- MALBP (*Mixed model Assembly Line Balancing Problem*)  
Yaitu penyeimbangan pada lini perakitan yang memproduksi beberapa variasi model produk dengan pengaturan waktu element kerja yang berbeda terkait dengan variasi model tersebut untuk mendapatkan kapasitas produksi serta biaya yang optimal.
- UALBP  
Yaitu penyeimbangan lini perakitan yang berbentuk U (*U-shape*) dimana operator memungkinkan untuk mengerjakan beberapa element kerja pada stasiun kerja yang berbeda.



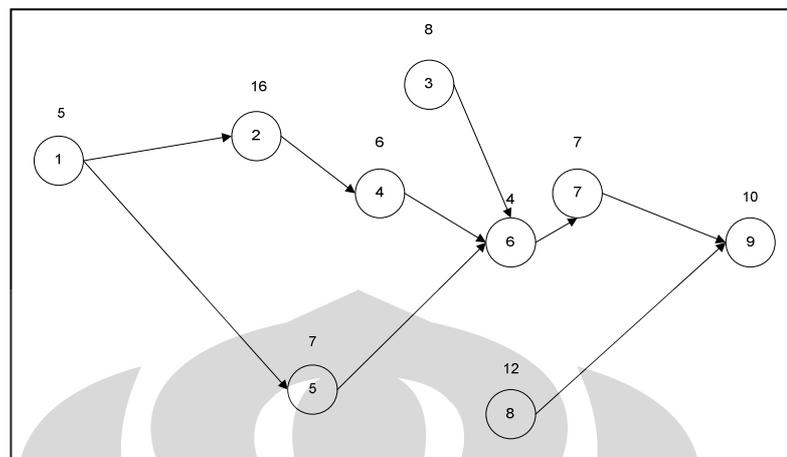
**Gambar 2.3** Klasifikasi permasalahan keseimbangan lini perakitan

### 2.2.2. Langkah-langkah dalam penyeimbangan Lini Perakitan

Beberapa langkah yang dilakukan dalam penyeimbangan sebuah lini perakitan yaitu :

1. Menetapkan urutan pekerjaan perakitan dalam sebuah diagram perakitan (*precedence diagram*) yang menginformasikan hubungan

serta keterkaitan antar elemen-elemen kerja pada lini perakitan tersebut.



**Gambar 2.5** *Precedence Diagram*

2. Waktu siklus (*Cycle Time*).

Waktu yang dibutuhkan oleh sebuah lini perakitan untuk dapat menyelesaikan perakitan sebuah produk. Waktu siklus ini merupakan representasi waktu terlama yang dihasilkan oleh satu stasiun kerja pada lini perakitan tersebut.

Waktu siklus dapat dihitung dengan rumus :

$$CT = \frac{\text{Production Time per hari}}{\text{Output per hari}} \quad (2.1)$$

3. Jumlah minimum stasiun kerja.

Untuk memenuhi pembatasan waktu siklus, jumlah minimum stasiun kerja dapat dihitung dengan menggunakan rumus :

$$N = \frac{\text{Jumlah Total waktu penyelesaian elemen kerja}}{\text{Waktu siklus}} \quad (2.2)$$

4. Metode penyeimbangan lini perakitan.

Untuk penyeimbangan sebuah lini perakitan digunakan pendekatan baik secara *exact* maupun *heuristic* dengan beberapa metode. Pemilihan metode penyeimbangan merupakan hal penting guna didapatkan hasil keseimbangan lini perakitan yang optimal.

5. Efisiensi Lini Perakitan (*Line Efficiency LE*).

Yaitu rasio dari jumlah total penyelesaian tiap elemen-elemen kerja dengan jumlah perkalian antara stasiun kerja dengan waktu siklus pada

lini perakitan. Rasio ini merepresentasikan keseimbangan dalam penempatan elemen-elemen kerja pada stasiun kerja.

Effisiensi lini perakitan ini diperoleh dengan menggunakan rumus :

$$LE = \frac{\sum_{i=1}^k ST_i}{(k)(CT)} \quad (2.3)$$

Dimana :

$ST_i$  = *Station Time* atau waktu stasiun kerja ke  $i$

$k$  = Jumlah Stasiun Kerja

$CT$  = *Cycle Time* atau waktu siklus terpanjang

#### 6. *Smoothnessindex (SI)*

Merupakan indeks yang menunjukkan pancaran relative dari suatu keseimbangan lini. *Smoothnessindex* sempurna jika nilainya 0 .

*Smoothnessindex* dapat dihitung dengan menggunakan rumus :

$$SI = \sqrt{\sum_{i=1}^k (ST_{max} - ST_i)^2} \quad (2.4)$$

Dimana :

$ST_i$  = *Station Time* atau waktu stasiun kerja ke  $i$

$k$  = Jumlah Stasiun Kerja

$ST_{max}$  = waktu stasiun terlama

### 2.2.3. Metode Penyelesaian Masalah Keseimbangan Lini Perakitan

Untuk mendapatkan solusi yang optimal atas permasalahan keseimbangan lini perakitan dapat diselesaikan dengan beberapa metodologi, antara lain<sup>6</sup> :

1. Metode *Exact*
  - *Linear Programming (LP)*
  - *Integer Programming (IP)*
  - *Dynamic Programming (DP)*
  - *Goal Programming (GP)*
  - *Shortest-path tech. (SP)*

<sup>6</sup> K Nuchara, P Nalin "The Assembly Line Balancing Problem". KKU Engineering Journal Vol. 34 No .2 (133 - 140) March – April 2007

- *Maximal-path tech. (MP)*
  - *Branch and bound (BB)*
2. Metode *Nonexact, Heuristic*
- *Priority ranking and assignment*
  - *Tree search (heuristic BB)*
  - *Trade and transfer*
  - *Random sampling*
  - *Others ; task grouping, approximation tech*
3. *Meta-Heuristic*

Beberapa tahun belakang, metode *metaheuristic* merupakan metode yang sering digunakan dalam penyelesaian permasalahan optimasi. Beberapa keunggulannya antara lain<sup>7</sup> :

- Tidak memerlukan kondisi special atas fungsi objektif (*Objective Function*) maupun batasannya (*constraint*) pada system yang sedang dioptimasi.
- Dapat diaplikasikan dalam permasalahan yang bersifat *continuous* maupun permasalahan *combinatorial*.
- Dapat digunakan untuk *multimodal* and *multiobjective optimization*.

Namun terdapat juga beberapa kekurangan pada penggunaan metode ini dalam menyelesaikan permasalahan optimasi yaitu :

- Sangat sensitive terhadap penentuan parameter control
- Pada beberapa permasalahan, memerlukan waktu komputasi yang cukup tinggi

Metode *metaheuristic* diklasifikasikan kedalam 2 group yaitu :

- a. *Neighborhood metaheuristics*, solusi yang dihasilkan pada metode ini hanya satu. Beberapa metoda yang menggunakan pendekatan ini antara lain :

---

<sup>7</sup> F Vitaliy, "Differential Evolution, In Search of Solutions", Springer Science+Business Media, LLC, 2006

- *Simulated Annealing*

*Simulated Annealing* adalah salah satu algoritma untuk optimisasi yang bersifat generik. Berbasiskan probabilitas dan mekanika statistik, algoritma ini dapat digunakan untuk mencari pendekatan terhadap solusi optimum global dari suatu permasalahan. Permasalahan-permasalahan yang membutuhkan pendekatan *Simulated Annealing* adalah masalah-masalah optimisasi serta kombinatorial, di mana ruang pencarian solusi yang ada terlalu besar, sehingga hampir tidak mungkin ditemukan solusi eksak terhadap permasalahan itu.

*Simulated Annealing* didasari oleh sebuah analogi *annealing* proses pada keilmuan *metalurgi*, yaitu proses pembentukan kristal dalam suatu materi dengan didahului oleh sebuah proses pemanasan hingga proses tertentu untuk memberikan kesempatan pada atom-atom dalam materi itu untuk bergerak secara bebas untuk kemudian di lanjut oleh proses pendinginan perlahan dan terkendali agar atom-atom tersebut menemukan tempat yang optimum.

- *Tabu Search*

Adalah algoritma metaheuristik yang dapat digunakan untuk memecahkan masalah *combinatorial optimization*. Dalam *Tabu Search* digunakan prosedur pencarian solusi pada populasi lokal maupun *neighbourhood* untuk dapat berpindah secara perlahan dari solusi  $x$  ke solusi  $x'$  hingga sebuah kriteria terpenuhi.

Untuk mengeksplorasi ruang pencarian yang mungkin terlewatkan oleh prosedur pencarian pada solusi lokal, maka dibentuklah *neighbourhood structure*  $N^*(x)$  dari solusi tersebut untuk kemudian dibuat sebuah prosedur pencarian hingga solusi yang perlahan berpindah dari solusi  $x$  ke solusi  $x'$  pada struktur  $N^*(x)$ .

b. *Population-based metaheuristics*, dimana solusi yang dihasilkan secara simultan bergerak menuju kearah optimal seiring dengan jumlah iterasi yang digunakan. Beberapa metoda yang menggunakan pendekatan ini (*Population-based*) antara lain :

- *Genetic Algorithm*

*Genetik Algorithm* adalah metode yang diadaptasi dari proses genetik dari organisme biologi yang digunakan untuk memecahkan permasalahan optimasi. Solusi terbentuk dari proses generasi sebuah individu, dengan menggunakan prinsip alam bahwa melalui proses generasi akan terbentuk sebuah individu yang lebih tangguh. Proses alami tersebut meliputi proses mutasi, kawin silang (*crossover*) serta proses seleksi atas individu yang lebih baik dan tangguh yang akan dimasukan pada populasi untuk dijadikan solusi pada proses optimasi<sup>8</sup>.

- *Particle Swarm optimization*

*Particle swarm optimization* Adalah algoritma evolusi berbasis populasi yang digunakan untuk memecahkan permasalahan. *Particle swarm optimization* merupakan bagian dari *swarm intelligence* yang berbasis prinsip-prinsip sosial psikologi yang memberikan solusi pada rung lingkup sosial maupun aplikasi engineering.

- *Differential Evolution Algorithm (DEA)*

*Differential Evolution Algorithm* adalah algoritma metaheuristik terbaru yang merupakan pengembangan dari *Genetic Algorithm*. Perbedaan mendasar antara kedua algoritma tersebut terletak pada mekanisme mutasi (*mutation*) dan kawinsilang (*crossover*). DEA telah diterapkan dalam berbagai

---

<sup>8</sup> Sabuncuoglu, Erel E., Tanyer M., "Assembly line balancing using genetic algorithms" Bilkent University, Bilkent, Ankara 06533, Turkey, 1998

aplikasi pada masalah optimasi dengan tingkat kesuksesan yang cukup baik dibandingkan dengan metode lainnya<sup>9</sup>

Penggunaan *population-based metaheuristics* memiliki beberapa keunggulan dalam penyelesaian permasalahan optimasi yaitu :

- Dapat memberikan informasi terkait dengan konvergensi fungsi objektifnya (*objective function*).
- Tidak terlalu sensitive terhadap ketidakpantasan sebuah individu untuk menjadi sebuah solusi.
- Meningkatkan probabilitas untuk pencapaian kondisi optimal.

## 2.3. Algoritma Genetika (*Genetic algorithm*)

### 2.3.1. Definisi Algoritma Genetika.

Algoritma Genetika merupakan salah satu jenis *Evolutionary Algorithm* yang paling populer dan paling banyak digunakan. Algoritma Genetika adalah algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah. Algoritma genetika merupakan suatu metode yang dapat digunakan untuk memecahkan masalah optimasi dengan menggunakan program komputer<sup>10</sup>. Salah satu masalah optimasi yang ada di dunia ini adalah masalah keseimbangan lini lintasan perakitan. Kemunculan Algoritma Genetika dibuat berdasarkan pada proses genetik dari organisme makhluk hidup dalam ilmu biologi. Dalam ilmu biologi, sekumpulan individu yang sama, yang disebut *spesies*, hidup, bereproduksi, dan mati dalam suatu area yang disebut populasi. Jika anggota-anggota populasi terpisah, maka individu-individu akan membentuk beberapa populasi yang terpisah. Dalam waktu yang cukup lama, mungkin saja akan terjadi proses pembentukan *spesies* baru atau dikenal dengan istilah *speciation*. Dalam hal ini terjadi perubahan hereditas (*heredity*) secara bertahap yang membentuk ciri-ciri baru pada spesies tersebut. Perubahan secara bertahap tersebut dikenal sebagai *co-evolution*.

<sup>9</sup> Nearchou A.C., "Balancing large assembly line by a new heuristic based on differential evolution method", Springer-Verlag London Limited 2006

<sup>10</sup> Sabuncuoglu, I., Erel E, Tayer, M., P.296

Holland (1975) menunjukkan bahwa simulasi komputer dari proses makhluk hidup dapat diterapkan untuk memecahkan masalah optimasi. Goldberg (1989) mengemukakan sejumlah aplikasi dari metode algoritma genetika yang dapat digunakan untuk memecahkan masalah optimasi. Pada umumnya, kekuatan dari metode Algoritma Genetika ini adalah adanya fakta bahwa teknik ini sangat kuat dan dapat digunakan untuk lingkup masalah yang luas. Walaupun metode Algoritma Genetika tidak menjamin untuk mendapatkan solusi yang optimal, namun metode ini dapat menghasilkan solusi yang baik (relatif optimal) dengan kebutuhan komputasi yang memungkinkan.

Konsep penting dalam teori evolusi adalah *fitness* dan *selection* untuk proses reproduksi. Pada proses evolusi di dunia nyata, terdapat dua cara reproduksi, yaitu *sexual reproduction* dan *asexual reproduction*. Pada *sexual reproduction*, kromosom-kromosom dari dua individu (sebagai orang tua) dikombinasikan untuk menghasilkan individu baru. Artinya kromosom pada individu baru berisi beberapa gen yang diambil dari orang tua pertama dan beberapa gen lainnya diambil dari orang tua kedua. Hal ini disebut sebagai *crossover* (pindah silang). Namun demikian proses pengkopian gen orang tua ini tidak luput dari kesalahan. Kesalahan pengkopian gen ini dikenal dengan istilah *mutation* (mutasi). Sedangkan pada *asexual reproduction*, hanya satu individu orang tua yang diperhatikan, sehingga tidak terjadi proses *crossover*. Tetapi proses mutasi juga mungkin terjadi pada *asexual reproduction*.

### 2.3.2. Aplikasi Algoritma Genetika

Sejak pertama kali dirintis oleh John Holland pada tahun 1960-an, Algoritma Genetika telah dipelajari, diteliti, dan diaplikasikan secara luas pada berbagai bidang. Algoritma Genetika banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter optimal. Hal ini membuat banyak orang mengira bahwa Algoritma Genetika hanya bisa digunakan untuk memecahkan masalah optimasi. Pada kenyataannya, Algoritma Genetika juga memiliki performansi yang bagus untuk masalah-masalah selain optimasi.

Keuntungan penggunaan Algoritma Genetika sangat jelas terlihat dari

**Universitas Indonesia**

kemudahan implementasi dan kemampuannya untuk menemukan solusi yang baik (bisa diterima) secara cepat untuk masalah-masalah berdimensi tinggi. Algoritma Genetika sangat berguna dan efisien untuk masalah dengan karakteristik sebagai berikut:

- Ruang masalah sangat besar, kompleks, dan sulit dipahami,
- Kurang atau bahkan tidak ada pengetahuan yang memadai untuk merepresentasikan masalah ke dalam ruang pencarian yang lebih sempit,
- Tidak tersedianya analisis matematika yang memadai,
- Ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi,
- Solusi yang diharapkan tidak harus optimal, tetapi cukup baik atau bisa diterima,
- Terdapat batasan waktu, misalnya dalam *real time systems* (sistem waktu nyata)

Algoritma Genetika telah banyak diaplikasikan untuk penyelesaian masalah dan permodelan antara lain:

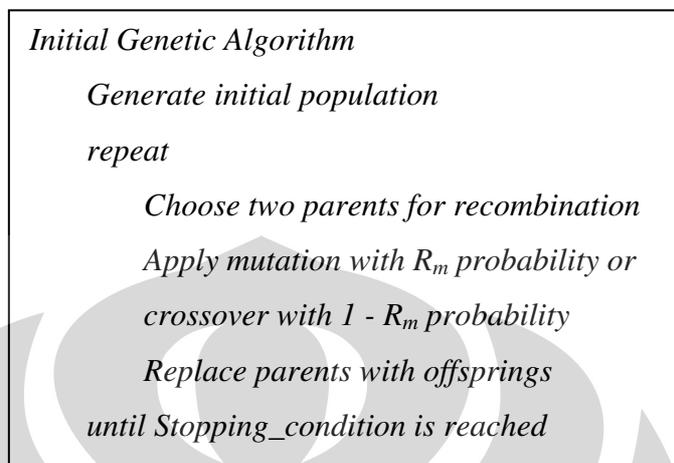
- Optimasi
- Pemrograman otomatis
- *Machine learning*
- Model ekonomi
- Model sistem imunisasi
- Model ekologis
- Dll.

### 2.3.3. Struktur Dasar Algoritma Genetika Masalah Keseimbangan Lini

Struktur metode heuristik Algoritma Genetika yang digunakan untuk memecahkan masalah keseimbangan lini lintasan perakitan adalah sama dengan Whitley dan Kauth's (1988) GENITOR, dimana hanya satu operasi kawin silang (*crossover*) pada setiap iterasi<sup>11</sup>.

<sup>11</sup> Sabuncuoglu, I., Erel E, Tayer, M., P.297-299

Struktur dasar metode heuristik Algoritma Genetika dapat dilihat pada gambar 2.5 di bawah.



**Gambar 2.5** Prosedur dasar pada Algoritma Genetika.

Setelah skema pengkodean ditentukan, Algoritma Genetika diinisialisasi untuk sebuah populasi dengan  $N$  kromosom. Gen-gen yang mengisi masing-masing kromosom dibangkitkan secara random, biasanya menggunakan distribusi seragam (*uniform distribution*). Masing-masing kromosom akan didekodekan menjadi individu dengan nilai *fitness* tertentu. Sebuah populasi baru dihasilkan dengan menggunakan mekanisme seleksi alamiah, yaitu memilih individu-individu secara proporsional terhadap nilai *fitness*-nya, dan genetika alamiah (pindah silang dan mutasi).

#### 2.3.4. Komponen-Komponen Algoritma Genetika

Beberapa karakteristik yang ada pada struktur Algoritma Genetika untuk permasalahan keseimbangan lini lintasan perakitan adalah sebagai berikut:<sup>12</sup>

- *Coding* (pengkodean)

Setiap elemen pekerjaan diwakili oleh suatu angka yang diletakkan benang (kromosom) dengan ukuran benang sama dengan jumlah elemen pekerjaan. Elemen pekerjaan pada kromosom yang terkait dengan pengaturan pada proses kerja, kemudian elemen pekerjaan

<sup>12</sup> Sabuncuoglu, I., Erel E, Tayer, M., P.299

dialokasikan pada stasiun-stasiun kerja dimana jumlah waktu untuk melakukan elemen-elemen pekerjaan pada tiap-tiap stasiun kerja tidak melebihi waktu siklus.

- *Fitness Function*

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Di dalam evolusi alam, individu yang bernilai fitness tinggi yang akan bertahan hidup. Sedangkan individu yang bernilai fitness rendah akan mati.

Tujuan dari permasalahan keseimbangan lini lintasan perakitan adalah untuk meminimumkan jumlah stasiun kerja, walaupun apabila terdapat dua solusi yang berbeda dengan jumlah stasiun kerja yang sama, salah satunya memiliki kemungkinan yang lebih seimbang dibandingkan dengan yang lainnya. Untuk mencapai tujuan tersebut dapat digunakan *fitness function* yang memiliki dua tujuan yaitu untuk meminimumkan jumlah stasiun kerja dan mendapatkan stasiun kerja yang seimbang.

$$Fitness\ Function = \sqrt{\frac{\sum_{k=1}^n (S_{max} - S_k)^2}{n} + \frac{\sum_{k=1}^n (S_{max} - S_k)}{n}} \quad (2.3)$$

Dimana:

- n = jumlah stasiun kerja
- S<sub>max</sub> = waktu maksimum stasiun kerja
- S<sub>k</sub> = waktu stasiun kerja k

Bagian pertama pada *fitness function* bertujuan untuk mendapatkan keseimbangan terbaik diantara solusi yang dihasilkan yang memiliki jumlah stasiun kerja yang sama, sedangkan bagian kedua bertujuan untuk meminimumkan jumlah stasiun kerja.

- *Initial Population* (Populasi Awal)

Populasi awal dihasilkan secara acak (*randomly*) dengan menjamin kelayakan populasi awal tersebut dengan hubungan pendahuluan (*precedence relations*).

- *Crossover* dan *Mutation* (Kawin silang dan Mutasi)

Kawin silang atau mutasi dapat dilakukan berdasarkan probabilitas tertentu, sebagai contoh apabila probabilitas kawin silang adalah 98% maka probabilitas mutasi adalah sebesar 2%. Operator kawin silang merupakan jenis operator kawin silang Davis (1985). Dua *parent* yang dipilih untuk kawin silang dipotong pada dua titik potong acak. *Offspring* (keturunan/anak) mengambil gen yang sama di luar titik potong pada lokasi yang sama dengan *parent* dan gen diantara titik potong yang diacak berdasarkan perintah.

Alasan utama yang membuat operator kawin silang tersebut sangat cocok untuk masalah keseimbangan lini lintasan perakitan adalah operator tersebut menjamin kelayakan dari *offspring* (keturunan/anak). Karena kedua *parent* adalah layak, maka kedua keturunan/anak harus layak. Kunci utama pada permasalahan keseimbangan lini lintasan perakitan adalah menjaga suatu populasi yang layak agar usaha komputasi yang dilakukan menjadi berkurang.

Operator mutasi dari Leu (1994) adalah *scramble mutation* (mutasi aduk) di mana suatu titik potong acak dipilih dan gen setelah titik potong diaduk secara acak sesuai kelayakan. Elitisme yaitu mengganti satu *parent* dengan satu *offspring* (keturunan/anak) hanya jika *offspring* (keturunan/anak) lebih baik dari *parent* dapat dilakukan pada prosedur *crossover* (kawin silang) dan prosedur *mutation* (mutasi). Kedua operator tersebut sama dengan operator kawin silang dan mutasi yang dikemukakan oleh Leu (1994).

- *Scalling* (Penskalaan)

Nilai *fitness* harus di-skala-kan sehingga total nilai skala *fitness* sama dengan 1. yang digunakan untuk mengaktifkan prosedur seleksi.

- *Selection Procedure* (Prosedur Seleksi)

Pemilihan dua buah kromosom sebagai orang tua (*parents*), yang akan dipindahkan, biasanya dilakukan secara proporsional sesuai dengan nilai *fitness*nya. Metode seleksi yang umum digunakan adalah

**Universitas Indonesia**

*roulette-wheel selection* (seleksi roda *roulette*). Sesuai dengan namanya, metode ini menirukan permainan *roulette-wheel* di mana masing-masing kromosom menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai *fitness*-nya. Kromosom yang memiliki nilai *fitness* lebih besar menempati potongan lingkaran yang lebih besar dibandingkan dengan kromosom bernilai *fitness* rendah. Metode *roulette-wheel selection* sangat mudah diimplementasikan dalam pemrograman MATLAB. Pertama, dibuat interval nilai kumulatif (dalam interval  $[0,1]$ ) dari nilai *fitness* masing-masing kromosom dibagi total nilai *fitness* dari semua kromosom. Sebuah kromosom akan terpilih jika bilangan *random* yang dibangkitkan berada dalam interval akumulatifnya.

- *Stopping Condition* (Keadaan Berhenti)

Algoritma akan berhenti setelah beberapa kali iterasi. Parameter jumlah iterasi yang digunakan adalah 100.

## BAB III PENGUMPULAN DATA

### 3.1. Profil Perusahaan

penelitian ini dilakukan pada perusahaan yang menghasilkan alat-alat berat yang berdiri pada tahun 1982 sebagai perusahaan yang memproduksi dan merakit serangkaian alat-alat berat dan komponen-komponen terkait yang sebagian besar digunakan di bidang konstruksi, kehutanan, dan sektor-sektor infrastruktur lainnya. Dengan misinya agar menjadi asset negara yang memproduksi alat-alat berat dengan kualitas dunia, perusahaan yang pada awalnya kegiatan hanya merakit alat-alat berat kemudian berkembang menjadi sebuah industri alat berat terpadu pertama di Indonesia dengan fasilitas produksi yang mencakup pabrik rangka dan komponen serta pabrik cor. Dengan itu maka visi sebagai market leader dapat diwujudkan hingga saat ini.

Dengan kapasitas produksi yang telah ditingkatkan, maka saat ini telah mampu memproduksi *Hydraulic Excavator, Bulldozer, Motor Grader, dan Off-highway Dump Truck* yang banyak digunakan dalam kegiatan penambangan besar. Selain itu juga telah mampu memproduksi *fabricated component* dan *steel cast components* untuk konsumsi dalam negeri maupun ekspor.

Dengan total area perakitan sebesar 62,841 M<sup>2</sup> yang menampung perakitan alat-alat berat dengan kategori *big, medium* dan *small*, kapasitas perakitanpun kian bertambah seiring dengan kebutuhan yang semakin meningkat.

### 3.2. Data Penelitian.

Perakitan alat-alat berat pada perusahaan ini dibagi dalam dua kategori perakitan yaitu perakitan dengan *single station assembly model (bay)* yang terbagi menjadi *big machine assembly bay* dan *medium machine assembly bay* dan *multy station assembly model (assembly line)*. Pemilihan antara dua model perakitan tersebut terkait dengan kapasitas perakitan yang diinginkan serta ketersediaan man power. Model perakitan *single station assembly* di gunakan untuk perakitan produk dengan kuantitas yang sedikit sedangkan model *multy station assembly* diterapkan untuk perakitan yang menuntut kapasitas produksi yang besar.

Untuk meningkatkan kapasitas pada *multy station assembly* ditentukan oleh keseimbangan lini perakitan. Pada penelitian ini, akan dilakukan keseimbangan lini perakitan sebuah produk tunggal dengan kuantiti yang cukup banyak yaitu lini perakitan *Hydraulic Excavator Type PC300-8* yang terdiri dari beberapa stasiun kerja.



**Gambar 3.1.** Lini perakitan *Hydraulic Excavator Tipe PC300-8*

Untuk melakukan penyeimbangan lini perakitan tersebut, digunakan beberapa data sekunder berupa :

- Data waktu kerja di bagian perakitan.
- Data elemen-elemen kerja pada perakitan *Hydraulic Excavator Tipe PC300-8* serta waktu untuk menyelesaikan tiap elemen kerja tersebut.
- Diagram keterkaitan (*precedence diagram*) antar elemen-elemen kerja.
- Keseimbangan lini perakitan yang saat ini telah dijalankan saat ini.

### 3.2.1. Data Waktu Kerja Bagian Perakitan

Waktu kerja normal dibagian perakitan terdiri atas 5 (lima) hari kerja pada 1 (satu) minggu. Pada table 3.1 dapat dilihat total waktu perakitan dalam 1 (satu) hari kerja.

**Tabel 3.1** Waktu perakitan yang tersedia dalam satu hari kerja.

Jam Kerja	Durasi	Total Waktu Kerja
07.00 - 09.50	170	460
10.00 - 12.00	120	
13.00 - 14.50	110	
15.00 - 16.00	60	

Pada perancangan keseimbangan lini perakitan, data waktu kerja ini diperlukan untuk penentuan waktu siklus (*cycle time*) yang hendak dicapai untuk memenuhi target produksi.

### 3.2.2. Elemen Kerja Perakitan dan Waktu Pengerjaan Tiap Elemen Kerja

Pada perakitan *Hydraulic Excavator Type PC300-8* ini terdapat 38 elemen kerja (*task*). Proses perakitan di mulai sejak lini perakitan dinyatakan *On* yang menandakan waktu perhitungan pengerjaan tiap elemen dihitung hingga lini perakitan dinyatakan *Off*. Table 3.2 dibawah memperlihatkan data elemen-elemen kerja pada lini perakitan.

Masing-masing elemen kerja tersebut diselesaikan pengerjaannya atas sebuah satuan waktu yang disebut *tasktime*. Penyeimbangan penempatan elemen kerja pada sebuah lini perakitan didasari oleh penyeimbangan waktu pada tiap stasiun kerja yang terdiri atas beberapa elemen kerja didalamnya.

**Tabel 3.2.** Elemen-Elemen kerja pda lini perakitan  
Hydraulic Excavator tipe PC300-8.

No	Kode Proses	Deskripsi Proses
1	A1	Pasang <i>Bracket Engine</i> (4pcs)
2	A2	Pasang <i>Clamp Hose Breater Engine</i>
3	A3	<i>Mounting Engine</i> pada Revol F
4	A4	Pasang <i>cabl</i> , hubungkan <i>harness</i> & pasang <i>clip</i>
5	B1	Pasang <i>Hose Suction Line</i>
6	B2	Hub. <i>Hose Engine</i> ke <i>Hyd. Tank</i> & pasang <i>clip</i>
7	B3	Hub. <i>Hose Delivery</i>
8	B4	Hub. <i>Hose Fuel</i> & pasang <i>clip</i>
9	B5	Hub. <i>Hose AC &amp; Heater</i>
10	C1	<i>Mounting Radiator</i> pada <i>Revo</i>
11	C2	Hub. <i>Hose Radiator</i> ke <i>Engine</i> (5 hose)
12	C3	Pasang <i>clip</i>
13	C4	Pasang <i>Tank Receiver</i>
14	D1	Hub. <i>Hose Cooler</i>
15	D2	Hub. <i>Hose A/C</i>
16	D3	Pasang <i>Air Cleaner</i>
17	D4	Hub. <i>Hose Air Cleaner</i>
18	D5	Pasang <i>Cover Air Cleaner</i>
19	D6	Pasang <i>Refill Pump</i>
20	E1	Pasang <i>Frame (L.H)</i> + <i>Rear Frame</i>
21	E2	Pasang <i>Filter</i> pada <i>Frame</i>
22	E3	Pasang <i>Frame (R.H)</i> + <i>Adjusment</i>
23	E4	Pasang <i>Switch Refill Pump</i>
24	F1	Pasang <i>Frame Tengah</i>
25	F2	Pasang <i>Cover Tengah / Partition</i>
26	F3	Pasang <i>Corroton</i> pada <i>Frame</i>

**Tabel 3.2.** Elemen-Elemen kerja pada lini perakitan *Hydraulic Excavator tipe PC300-8* (lanjutan).

No	Kode Proses	Deskripsi Proses
27	F4	Pasang <i>Cover Atas</i>
28	G1	Pasang <i>Under Cover</i>
29	G2	Pasang <i>Bracket Hose Refill Pump</i>
30	H1	Pengisian <i>Grease</i>
31	H2	Mounting <i>Revol Frame</i> pada <i>Track Frame</i>
32	H3	Pasang <i>Swing Machinery</i>
33	I1	Hub. <i>Hose machinery</i> dan <i>Hose Swivel Joint</i>
34	I2	Pengisian <i>Fuel</i>
35	I3	Pengisian Air <i>Radiator</i>
36	I4	Pengisian <i>Oil Hydraulic</i>
37	I5	Pasang <i>Silinder Boom</i>
38	J1	Pasang <i>Track Shoe</i>

Seperti yang telah disebut diatas, Selain data elemen-elemen kerja pada proses perakitan, data penting lain berkaitan dengan elemen kerja ini yaitu data waktu pengerjaan/penyelesaian (*tasktime*) tiap elemen kerja. Table 3.3 berikut ini adalah data waktu penyelesaian pengerjaan tiap elemen kerja pada perakitan *Hydraulic Excavator tipe PC300-8*.

**Tabel 3.3.** waktu penyelesaian elemen-elemen kerja pada lini perakitan *Hydraulic Excavator tipe PC300-8*.

No	Kode Proses	Waktu Pengerjaan (Menit)	No	Kode Proses	Waktu Pengerjaan (Menit)
1	A1	8.09	20	E1	24.08
2	A2	1.04	21	E2	4.02
3	A3	5.07	22	E3	17.07
4	A4	20.07	23	E4	3.04
5	B1	15.04	24	F1	5.08
6	B2	2.08	25	F2	19.02
7	B3	7.07	26	F3	4.07
8	B4	5.09	27	F4	10.07
9	B5	3.04	28	G1	37.04
10	C1	7.01	29	G2	2.02
11	C2	21.01	30	H1	3.08
12	C3	9.09	31	H2	34.04
13	C4	1.09	32	H3	18.07
14	D1	10.07	33	I1	6.07
15	D2	4.04	34	I2	12.06
16	D3	3.07	35	I3	3.02
17	D4	6.02	36	I4	12.04
18	D5	4.01	37	I5	19.04
19	D6	5.04	38	J1	30.07

### 3.2.3. Diagram Keterkaitan (*Precedence Diagram*).

Sebelum melakukan penyeimbangan sebuah lini perakitan, perlu untuk menentukan keterkaitan antar elemen kerja, sehingga penyeimbangan yang dilakukan layak untuk diterapkan. Berikut tabel keterkaitan antar elemen kerja pada perakitan *Hydraulic Excavator tipe PC300-8*.

**Tabel 3.4.** Keterkaitan antar elemen-elemen kerja pada lini perakitan *Hydraulic Excavator tipe PC300-8.*

No	Kode Proses	Elemen Kerja yang Mendahului	No	Kode Proses	Elemen Kerja yang Mendahului
1	A1	-	20	E1	19
2	A2	-	21	E2	20
3	A3	1	22	E3	19
4	A4	3	23	E4	22
5	B1	4	24	F1	22,23
6	B2	5	25	F2	24
7	B3	5	26	F3	24
8	B4	5	27	F4	25, 26
9	B5	5	28	G1	27
10	C1	3	29	G2	28
11	C2	10	30	H1	29
12	C3	11	31	H2	30
13	C4	12	32	H3	31
14	D1	5,13	33	I1	32
15	D2	5	34	I2	33
16	D3	15	35	I3	33
17	D4	16	36	I4	33
18	D5	17	37	I5	31
19	D6	14, 6, 7, 8, 9, 18	38	J1	31

Table diatas dapat digantikan oleh sebuah diagram yang menunjukkan keterkaitan antar elemen kerja yang disebut Diagram keterkaitan (*precedence diagram*) yang terlampir dalam penelitian ini.

### 3.2.4. Keseimbangan Lini Perakitan saat ini.

Lini perakitan yang telah diterapkan pada perakitan saat ini mengacu pada rencana produksi harian yang telah ditargetkan. Berikut beberapa data berkaitan dengan penerapan yang telah dilakukan pada lini perakitan saat ini, antara lain :

- Jumlah stasiun kerja (*Workstation*).  
Pada *main line* di bagian perakitan, stasiun kerja yang diterapkan berjumlah 10 stasiun kerja yang terhubung satu dengan lainnya oleh sebuah system ban berjalan (*conveyor*).
- Penempatan elemen kerja dan waktu siklus tiap stasiun kerja.  
Berikut ini penempatan elemen-elemen kerja pada masing-masing stasiun kerja.

**Tabel 3.5.** Penempatan elemen-elemen kerja pada stasiun kerja di lini perakitan *Hydraulic Excavator tipe PC300-8*.

Urutan Kerja		Stasiun Kerja	
No	Kode	Nama	Cycle Time
1	A1	STATION 1	34.27
2	A2		
3	A3		
4	A4		
5	B1	STATION 2	32.32
6	B2		
7	B3		
8	B4		
9	B5		
10	C1	STATION 3	38.2
11	C2		
12	C3		
13	C4		

**Tabel 3.5.** Penempatan elemen-elemen kerja pada stasiun kerja di lini perakitan *Hydraulic Excavator tipe PC300-8* (lanjutan).

Urutan Kerja		Stasiun Kerja	
No	Kode	Nama	Cycle Time
14	D1	STATION 4	32.25
15	D2		
16	D3		
17	D4		
18	D5		
19	D6		
20	E1	STATION 5	48.21
21	E2		
22	E3		
23	E4		
24	F1	STATION 6	38.24
25	F2		
26	F3		
27	F4		
28	G1	STATION 7	39.06
29	G2		
30	H1	STATION 8	55.19
31	H2		
32	H3		
33	I1	STATION 9	52.23
34	I2		
35	I3		
36	I4		
37	I5		
38	J1	STATION 10	30.07

- Waktu siklus lini perakitan (*cycle time/ Pitch time*).

Waktu siklus lini perakitan merupakan waktu siklus terlama diantara stasiun-stasiun kerja yang ada dan merupakan *bottleneck* pada lintasan perakitan. Table 3.6 dibawah ini menunjukkan bahwa stasiun kerja 8 merupakan *bottleneck* karena memiliki waktu siklus terlama yaitu sebesar 55.23 menit dan juga merupakan waktu siklus (*cycle time/pitch time*) pada lini perakitan yang saat ini diterapkan.

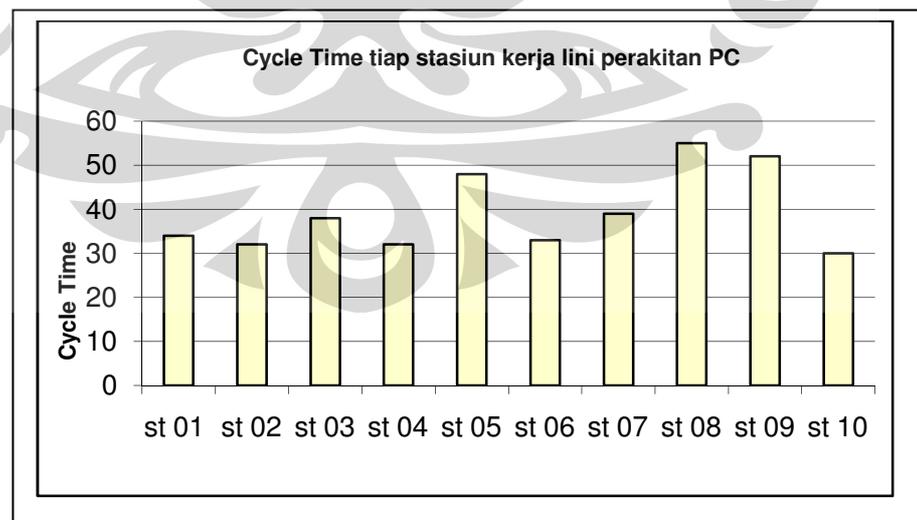
**Tabel 3.6.** Waktu siklus pada tiap-tiap stasiun kerja pada lini perakitan *Hydraulic Excavator tipe PC300-8*.

Stasiun Kerja	st 01	st 02	st 03	st 04	st 05
<i>Cycle time</i>	34.27	32.32	38.2	32.25	48.21

st 06	st 07	st 08	st 09	st 10
38.24	39.06	55.19	52.23	30.07

Grafik dibawah ini menegaskan bahwa stasiun kerja 8 merupakan stasiun dengan waktu siklus yang paling tinggi.



**Gambar 3.1.** Grafik waktu siklus tiap stasiun kerja pada lini perakitan *Hydraulic Excavator tipe PC300-8*

- Kapasitas Produksi

Kapasitas produksi maksimal yang dihasilkan oleh ditentukan oleh waktu siklus pada lini perakitan. Dengan membagi waktu kerja dengan waktu siklus tersebut maka dapat ditentukan kapasitas lini perakitan tersebut.

$$\text{Kapasitas perakitan} = \frac{\text{waktu perakitan yang tersedia}}{\text{Waktu siklus lini perakitan}}$$

$$\text{Kapasitas perakitan} = \frac{460}{55} = 8,3 \approx 8 \text{ unit/day}$$

Untuk meningkatkan kapasitas maka waktu siklus lini perakitan harus diturunkan dengan melakukan penyeimbangan lini perakitan.

- Efisiensi lini perakitan.

Efisiensi lini perakitan yang ada dapat dihitung dengan menggunakan rumus 2.3, dimana dengan membagi jumlah waktu pengerjaan elemen-elemen kerja dengan perkalian jumlah stasiun kerja dengan waktu siklus lini perakitan yang telah diketahui maka didapatkan efisiensi lini perakitan saat ini yaitu :

$$E = \frac{400.04 \text{ (menit)}}{10 \times 55.19} = 0.725$$

## BAB IV

### PENGOLAHAN DATA DAN ANALISA

#### 4.1. Penyusunan Algoritma.

Perancangan keseimbangan lini perakitan dengan menggunakan metoda *Genetic Algorithm* yang dilakukan pada penelitian ini menggunakan bahasa pemrograman pada *Matlab*. Penggunaan *Matlab* untuk aplikasi optimasi metaheuristik dipilih karena program ini memberikan kemudahan untuk mengaplikasikan algoritma-algoritma pada permasalahan optimasi.

*Matlab* adalah suatu bahasa pemrograman tingkat tinggi yang diperuntukkan untuk komputasi teknis. *Matlab* mengintegrasikan aspek komputasi, visualisasi, dan pemrograman dalam suatu lingkungan yang mudah dilakukan

Dalam memvisualisasikan sebuah objek, *Matlab* memiliki kemampuan merotasi obyek tanpa mengubah programnya. Konstruksi penyelesaian komputasi teknis dengan *Matlab* dapat dilakukan lebih cepat dibandingkan dengan bahasa pemrograman tradisional, seperti *C*, *C++*, dan *Fortran*. *Matlab* menyediakan fungsi-fungsi matematis untuk aljabar linear, statistik, optimasi, dan lainnya. 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<sup>13</sup>.

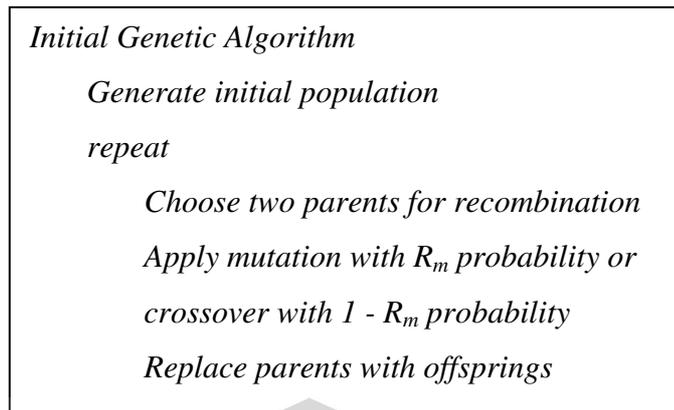
#### 4.1.1. Algoritma Genetika Pada Keseimbangan Lini Perakitan

Struktur dasar algoritma genetik pada permasalahan keseimbangan lini perakitan dapat dilihat pada gambar 4.1 dibawah ini<sup>14</sup>.

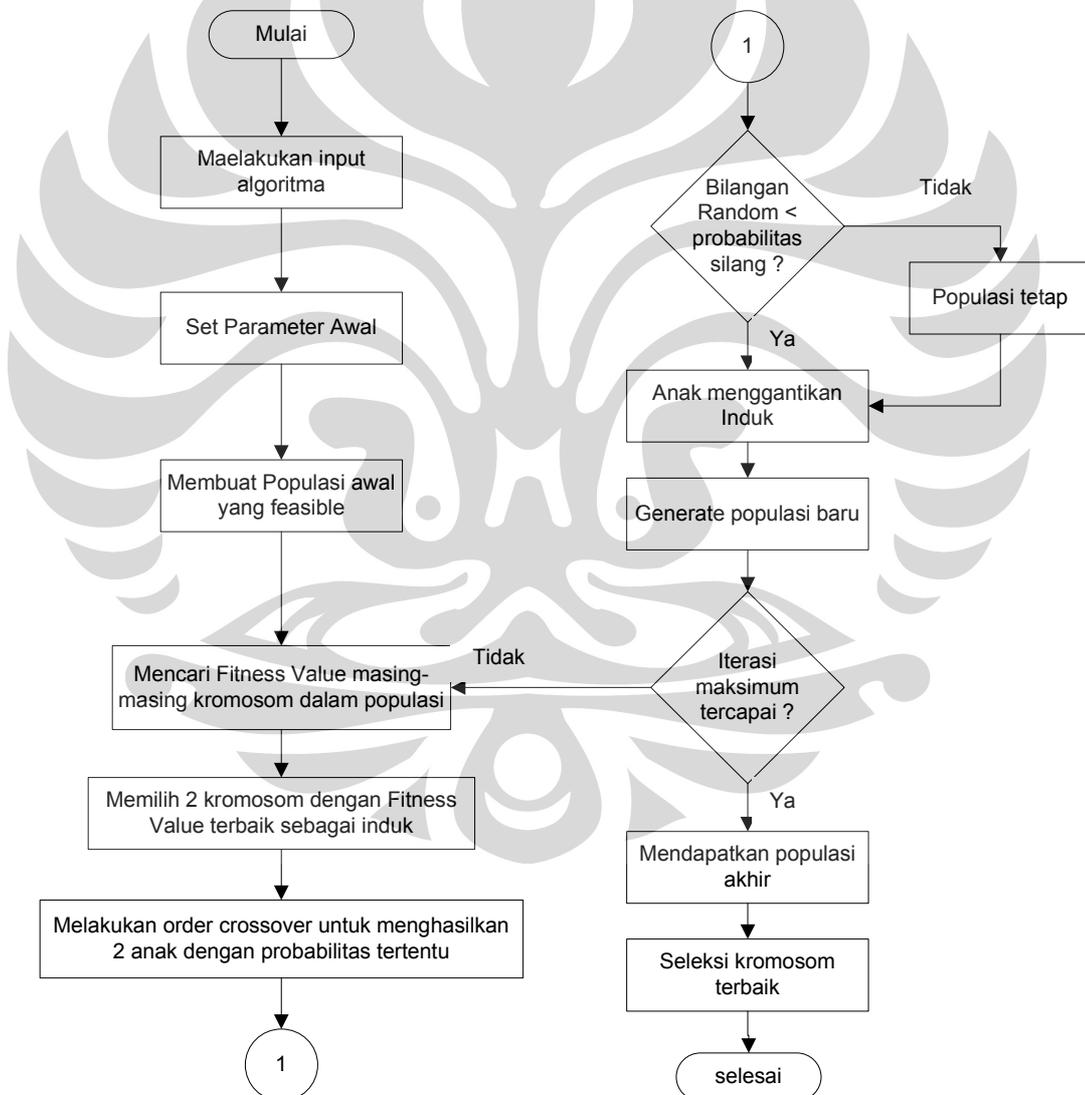
---

<sup>13</sup> <http://www.mathworks.com/products/matlab/description1.html>, (accessed 16 desember 2008)

<sup>14</sup> Erel S. E, Tanyer M., "Assembly line balancing using genetic algorithms", *Journal of Intelligent Manufacturing* (2000) 11, 295-310, Turkey 1998.



**Gambar 4.1** Standard Procedure Algoritma Genetika



**Gambar 4.2** Flowchart Algoritma Genetika untuk masalah keseimbangan lini perakitan

Penyelesaian permasalahan keseimbangan lini perakitan dengan metode algoritma genetik terdiri atas beberapa komponen yaitu :

- Inisialisasi populasi

Populasi awal berisikan sejumlah kromosom yang berisikan gen-gen yang merepresentasikan urutan elemen kerja. Pembentukan populasi awal dibentuk dengan memperhatikan kelayakan (*feasibilitasnya*) kromosom yang berada didalamnya. kelayakan yang dimaksud adalah gen-gen yang membentuk sebuah kromosom haruslah memperhatikan *precedence diagram* sehingga kromosom tersebut menjadi layak agar hasil yang didapatkan nantinya menjadi layak juga.

- *Fitness function*

Suatu individu dievaluasi berdasarkan sebuah nilai tertentu yang menjadi sebagai batasan yang hendak dicapai. *fitness function* pada keseimbangan lini perakitan ditujukan untuk mengevaluasi sebuah individu yang berisikan solusi atas permasalahan keseimbangan lini perakitan berupa urutan elemen kerja serta penempatannya pada sebuah stasiun kerja. Evaluasi tersebut sangat terkait dengan fungsi tujuan yang hendak dicapai.

Fungsi tujuan yang hendak dicapai pada keseimbangan lini perakitan kali ini adalah untuk meminimalkan jumlah stasiun kerja serta untuk mendapatkan stasiun kerja yang lebih seimbang.

- *LinearFitness Ranking*

Untuk menghindari kecenderungan konvergen yang prematur, dilakukan sebuah penskalaan nilai *fitness* yang berada pada interval maximum dan minimum dari nilai *fitness* tersebut.

- *Roulette Wheel*

Yaitu metode pemilihan dua individu sebagai individu induk (*parent*). Sesuai dengan namanya, metode ini menirukan permainan *Roulette Wheel*, dimana Individu diurutkan berdasarkan *fitness value*nya. Individu dengan *fitness value* yang lebih besar mendapatkan

porsi terbesar pada roda dan besar kemungkinan untuk terpilih sebagai Induk (*parent*).

- Pindah Silang (*Crossover*)

Proses pindah silang antara dua individu yang terpilih sebagai induk yang di potong pada dua titik potong acak. Anak akan mengambil gen yang sama di luar titik potong pada lokasi yang sama dengan induk (*parent*) dan gen diantara titik potong yang diacak dengan sebuah perintah.

Penerapan proses pindah silang pada permasalahan keseimbangan lini perakitan sangat sesuai karena hasil dari operasi ini akan menjamin kelayakan pada individu anak yang dihasilkan karena individu anak dihasilkan dari induk (*parent*) yang layak.

#### **4.1.2. Penerapan Algoritma Genetik Pada Keseimbangan Lini Perakitan**

##### ***Hydraulic Excavator Tipe PC300-8***

Penerapan algoritma genetika pada keseimbangan lini perakitan untuk penelitian pada lini perakitan *Hydraulic Excavator Tipe PC300-8* dilakukan beberapa penentuan parameter-parameter. Antara lain :

- Ukuran populasi

Populasi merupakan kromosom yang berisikan gen-gen yang merepresentasikan urutan elemen kerja haruslah memperhatikan kelayakannya. Pada penelitian ini ukuran populasi yang digunakan yaitu sebanyak 3 dengan memperhatikan kelayakan populasi tersebut.

- Populasi awal

Sesuai dengan yang telah dijelaskan diatas, bahwa populasi awal haruslah dibentuk dengan tingkat kelayakn yang terjamin agar solusi yang dihasilkan akan terjamin juga kelayakannya.

Berikut populasi awal yang dibentuk pada penelitian ini.

```

Populasi = [
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  ...
37 38 ;
1  2  3  4  5  8  ...
7  6  9  15 10 11  ...
12 13 14 16 17 18  ...
19 22 20 21 23 24  ...
26 25 27 28 29 30  ...
31 37 38 32 33 35  ...
34 36 ;
2  1  3  10 11 12  ...
13 4  5  15 16 17  ...
18 9  8  7  6  14  ...
19 20 21 22 23 24  ...
25 26 27 28 29 30  ...
31 32 37 38 33 36  ...
35 34];

```

**Gambar 4.3** Populasi awal yang layak

- *TaskTime*

Yaitu sebuah *matrik* yang berisi waktu penyelesaian tiap elemen kerja pada lini perakitan *Hydraulic Excavator Type PC300-8* yang akan dilakukan keseimbangan pada lini perakitannya.

```

TaskTime = [
8.09; 1.04; 5.07; 20.07; 15.04;
2.08; 7.07; 5.09; 3.04; 7.01;
21.01; 9.09; 1.09; 10.07; 4.04;
3.07; 6.02; 4.01; 5.04; 24.08;
4.02; 17.07; 3.04; 5.08; 19.02;
4.07; 10.07; 37.04; 2.02; 3.08;
34.04; 18.07; 6.07; 12.06; 3.02;
12.04; 19.04; 30.07 ];

```

**Gambar 4.4** Parameter *TaskTime*

- Probabilitas pindah silang

Probabilitas silang yang digunakan pada penelitian ini yaitu 0.98 berdasarkan penelitian yang sebelumnya.

- Jumlah iterasi

Jumlah iterasi digunakan untuk memberikan kemungkinan pencapaian *fitness value* yang lebih baik. Jumlah iterasi yang digunakan pada penelitian ini yaitu sebesar 100.

- Waktu siklus yang diinginkan

Penelitian yang dilakukan kali ini ditujukan untuk memperoleh keseimbangan lini perakitan untuk kapasitas perakita sebanyak 12 unit perhari sehingga untuk memperoleh kapasitas tersebut maka waktu siklus yang harus dipenuhi yaitu sebesar 38.33 menit.

Selain itu, dilakukan keseimbangan untuk beberapa rentang waktu siklus yaitu : 57.5, 51.11, 46, 41.82, 38.33 menit.

#### 4.2. Validasi.

Sebelum melakukan pengolahan data untuk mendapat kan solusi atas permasalahan keseimbangan lini perakitan pada program yang telah dibuat, program tersebut haruslah divalidasi apakah keluaran (outpun) yang dihasilkan sesuai dengan yang diharapkan yaitu dengan membandingkan hasil keluaran program dengan dengan perhitungan secara manual.

Perhitungan manual tsb dilakukan dengan menggunakan Microsoft *excel*. Berikut hasil perhitungan secara manual.

**Tabel 4.1.** *Validasi* perhitungan manual.

Urutan Kerja		Waktu Pengerjaan (Menit)	Stasiun Kerja			
No	Kode		Nama	Cycle Time	Eff	CTMax-Cti
1	A1	8.09	ST-01	21.21	0.57	15.83
2	A2	1.04				
3	A3	5.07				
10	C1	7.01				
11	C2	21.01	ST-02	30.1	0.81	6.94
12	C3	9.09				
4	A4	20.07	ST-03	35.11	0.95	1.93
5	B1	15.04				

Tabel 4.7. Validasi perhitungan manual (lanjutan).

Urutan Kerja		Waktu Pengerjaan (Menit)	Stasiun Kerja			
No	Kode		Nama	Cycle Time	Eff	CTMax-Cti
15	D2	4.04	ST-04	35.51	0.96	1.53
13	C4	1.09				
16	D3	3.07				
17	D4	6.02				
18	D5	4.01				
8	B4	5.09				
9	B5	3.04				
7	B3	7.07				
6	B2	2.08				
14	D1	10.07	ST-05	32.18	0.87	4.86
19	D6	5.04				
22	E3	17.07				
20	E1	24.08	ST-06	36.22	0.98	0.82
21	E2	4.02				
23	E4	3.04				
24	F1	5.08				
25	F2	19.02	ST-07	33.16	0.90	3.88
26	F3	4.07				
27	F4	10.07				
28	G1	37.04	ST-08	37.04	1	
29	G2	2.02	ST-09	5.1	0.14	31.94
30	H1	3.08				
31	H2	34.04	ST-10	34.04	0.92	3
32	H3	18.07	ST-11	36.2	0.98	0.84
33	I1	6.07				
34	I2	12.06				
35	I3	3.02	ST-12	34.1	0.92	2.94
36	I4	12.04				
37	I5	19.04				
38	J1	30.07	ST-13	30.07	0.81	6.97
TOTAL				400.04	10.80	
			Effisiensi Lini		0.83	

hasil yang didapat dari perhitungan manual sesuai dengan hasil yang didapat dari perhitungan yang dilakukan oleh program. Oleh karena itu maka program yang dibuat telah tervalidasi hasilnya sehingga telah dapat digunakan untuk penyelesaian keseimbangan lini perakitan.

### 4.3. Hasil dan Analisa.

Setelah program algoritma genetika untuk keseimbangan lini perakitan siap untuk dijalankan (*running*), berikut hasil dan analisa keseimbangan lini perakitan *Hydraulic Excavator Type PC300-8* yang didapatkan berdasarkan parameter-parameter yang telah di tentukan diatas.

#### 4.3.1. Hasil Running Program

Berikut hasil yang diperoleh dari program algoritma genetika yang telah dibuat.

1. Waktu siklus yang sama dengan waktu siklus yang diterapkan saat ini.

Untuk menguji keefektifan program yang program yang telah di buat, maka program dijalankan dengan input waktu siklus lini perakitan sebesar 55.19 menit yang merupakan waktu siklus pada lini perakitan yang saat ini dijalankan.

Berikut hasil dari *running program* dengan menggunakan inputan yang telah dijelaskan sebelumnya.

- Waktu siklus maksimal ( $CT_{max}$ )

$$CT_{max} = 52.4000$$

- Jumlah stasiun kerja

9

- Jumlah Produksi

$$\text{JumProduksi} = 8$$

- Waktu siklus pada tiap-tiap stasiun kerja

T =

Columns 1 through 6

52.4000 52.2500 49.4600 36.2200 33.1600 2.1400

Columns 7 through 9

52.1100 49.1100 33.1900

- Jalur perakitan terbaik

JalurTerbaik =									
Columns 1 through 10									
2	1	3	10	11	12	13	4	5	15
Columns 11 through 20									
16	17	18	9	8	7	6	14	19	22
Columns 21 through 30									
20	21	23	24	25	26	27	28	29	30
Columns 31 through 38									
31	32	37	38	33	36	35	34		

- Effisiensi lini perakitan

$$LineEfficiency = 0.8082$$

- *Smoothness Index*

$$SmothnessIndex = 33.5535$$

## 2. Waktu siklus yang hendak dicapai.

Setelah memasukkan parameter-parameter awal maka hasil perhitungan dengan menggunakan metode *Genetic Algorithm* pada program *Matlab* untuk input *Cycle Time* yang hendak dicapai pada penelitian ini yaitu sebesar 38.33 menit didapatkan hasil sebagai berikut :

- Waktu siklus maksimal (CTmax)

$$CTmax = 37.0400$$

- Jumlah stasiun kerja

13

- Jumlah Produksi

$$JumProduksi = 12$$

- Waktu siklus pada tiap-tiap stasiun kerja

T =
Columns 1 through 6
21.2100 30.1000 35.1100 35.5100 32.180 6.2200
Columns 7 through 12
33.1600 37.0400 5.1000 34.0400 36.2000 4.1000
Column 13
30.0700

- Jalur perakitan terbaik

JalurTerbaik =
Columns 1 through 10
1 2 3 10 11 12 4 5 15 13
Columns 11 through 20
16 17 18 8 9 7 6 14 19 22
Columns 21 through 30
20 21 23 24 25 26 27 28 29 30
Columns 31 through 38
31 32 33 34 35 36 37 38

- Effisiensi lini perakitan

$$\text{LineEfficiency} = 0.8028$$

- Smoothness Index

$$\text{SmoothnessIndex} = 37.8320$$

#### 4.3.2. Analisa.

Hasil yang telah didapat akan dianalisa antara lain dengan membandingkan keseimbangan lini yang diperoleh pada program dengan keseimbangan lini yang lama (manual), membandingkan antara berbagai variasi parameter waktu siklus.

1. Analisa lini perakitan hasil running program (GA) dengan lini perakitan lama.

**Tabel 4.2.** Perbandingan keseimbangan lini perakitan *Hydraulic Excavator tipe PC300-8* baru Vs lama.

<i>Cycle Time / Pitch Time</i>	55.19	Menit
Kapasitas Produksi	8	Unit/hari
	lama	baru
Jumlah Stasiun Kerja	10	9
Urutan Kerja tiap Stasiun Kerja	Table 4.3	Tabel 4.4
Waktu Siklus Maksimal (CTMax)	55.19	53.29
Effisiensi Lini Perakitan	0.725	0.8082
SmothnessIndex	151.86	35.585

Dari table diatas, dapat dilihat bahwa hasil perhitungan dengan menggunakan metode GA dapat memberikan keseimbangan lini perakitan yang lebih baik dari keseimbangan lini perakitan lama, hal ini dapat dilihat dengan membandingkan effisiensi lini perakitan. Effisiensi lini perakitan yang dihasilkan dengan menggunakan program GA memiliki nilai yang lebih baik yaitu 0.8082, sedangkan effisiensi lini perakitan lama hanya sebesar 0.725.

Selain itu, sesuai dengan tujuan awal di buatnya keseimbangan lini perakitan dengan metode GA yaitu meminimalkan jumlah stasiun kerja, hal ini dapat terpenuhi dengan terbentuknya lini perakitan baru dengan jumlah stasiun kerja yang lebih sedikit yaitu 9 stasiun kerja dengan kapasitas produksi/perakitan yang sama.

Tabel dibawah ini merupakan keseimbangan lini perakitan lama yang diterapkan saat ini.

Tabel 4.3 Keseimbangan lini perakitan lama.

Urutan Kerja		Waktu Pengerjaan (Menit)	Stasiun Kerja	
No	Kode		Nama	Cycle Time
1	A1	8.09	STATION 1	34.27
2	A2	1.04		
3	A3	5.07		
4	A4	20.07		
5	B1	15.04	STATION 2	32.32
6	B2	2.08		
7	B3	7.07		
8	B4	5.09		
9	B5	3.04		
10	C1	7.01	STATION 3	38.2
11	C2	21.01		
12	C3	9.09		
13	C4	1.09		
14	D1	10.07	STATION 4	32.25
15	D2	4.04		
16	D3	3.07		
17	D4	6.02		
18	D5	4.01		
19	D6	5.04		
20	E1	24.08	STATION 5	48.21
21	E2	4.02		
22	E3	17.07		
23	E4	3.04		
24	F1	5.08	STATION 6	38.24
25	F2	19.02		
26	F3	4.07		
27	F4	10.07		
28	G1	37.04	STATION 7	39.06
29	G2	2.02		
30	H1	3.08	STATION 8	55.19
31	H2	34.04		
32	H3	18.07		
33	I1	6.07	STATION 9	52.23
34	I2	12.06		
35	I3	3.02		
36	I4	12.04		
37	I5	19.04		
38	J1	30.07	STATION 10	30.07

Sebagai pembandingan, tabel berikut merupakan keseimbangan lini perakitan baru dengan parameter waktu siklus yang sama dengan yang telah diterapkan saat ini.

**Tabel 4.4** Keseimbangan lini perakitan baru (GA).

Urutan Kerja		Waktu Pengerjaan (Menit)	Stasiun Kerja	
No	Kode		Nama	Cycle Time
2	A2	1.04	STATION 1	52.4
1	A1	8.09		
3	A3	5.07		
10	C1	7.01		
11	C2	21.01		
12	C3	9.09		
13	C4	1.09		
4	A4	20.07	STATION 2	52.25
5	B1	15.04		
15	D2	4.04		
16	D3	3.07		
17	D4	6.02		
18	D5	4.01		
9	B5	3.04	STATION 3	32.39
8	B4	5.09		
7	B3	7.07		
6	B2	2.08		
14	D1	10.07		
19	D6	5.04		
20	E1	24.08	STATION 4	53.29
21	E2	4.02		
22	E3	17.07		
23	E4	3.04		
24	F1	5.08		
25	F2	19.02	STATION 5	33.16
26	F3	4.07		
27	F4	10.07		

**Tabel 4.4** Keseimbangan lini perakitan baru (GA) (lanjutan).

Urutan Kerja		Waktu Pengerjaan (Menit)	Stasiun Kerja	
No	Kode		Nama	Cycle Time
28	G1	37.04	STATION 6	42.14
29	G2	2.02		
30	H1	3.08		
31	H2	34.04	STATION 7	52.11
32	H3	18.07		
33	I1	6.07	STATION 8	40.17
35	I3	3.02		
36	I4	12.04		
37	I5	19.04		
38	J1	30.07	STATION 9	42.13
34	I2	12.06		

2. Analisa beberapa waktu siklus.

Selain untuk mendapatkan keseimbangan lini perakitan dengan waktu siklus yang hendak dicapai yaitu waktu siklus sebesar 38.33, dalam penelitian ini, dilakukan percobaan dengan variasi pada waktu siklus. Adapun parameter awal untuk tiap percobaan adalah sama antara lain :

- Ukuran populasi = 3
- Probabilitas silang = 0.98
- Iterasi = 100

Hasil dari percobaan tersebut dapat dilihat pada Tabel dibawah ini.

**Tabel 4.5** Perbandingan keseimbangan lini perakitan dengan berbagai variasi waktu siklus

	Waktu Siklus Perakitan		
	57.5	51.11	46
Kapasitas Perakitan	8	9	10
Jumlah Stasiun Kerja	8	10	11
Waktu Siklus Maks (CTMax)	56.43	49.34	45.3
Effisiensi Lini Perakitan	0.8697	0.818	0.7906
SmoothnessIndex	24.8617	31.0402	33.9835
Elapsed time	9.111389	9.014913	9.046424

	Waktu Siklus Perakitan	
	41.82	38.33
Kapasitas Perakitan	11	12
Jumlah Stasiun Kerja	12	13
Waktu Siklus Maks (CTMax)	39.22	38.2
Effisiensi Lini Perakitan	0.7971	0.8028
SmoothnessIndex	27.3564	39.7892
Elapsed time	8.930505	9.087256

a. Kapasitas Perakitan.

Kapasitas perakitan yang pada tabel diatas bervariasi pada rentang 8 hingga 12 unit perhari. Seiring dengan menurunnya waktu siklus sebuah lini perakitan maka kapasitas perakitan yang dihasilkan akan bertambah karena waktu siklus sebuah lini perakitan merepresentasikan waktu perakitan yang dibutuhkan untuk merakit sebuah produk.

Waktu siklus yang diharapkan untuk dapat diterapkan adalah sebesar 38.33 yang dapat meningkatkan kapasitas perakitan menjadi 12 unit perhari.

b. Jumlah Stasiun Kerja

Stasiun kerja yang dihasilkan pada percobaan berada pada rentang 8 hingga 13 stasiun kerja. Sama halnya dengan kapasitas perakitan, jumlah stasiun kerja akan bertambah seiring dengan

penurunan waktu siklus sebuah perakitan. Hal ini terjadi karena dengan total waktu penyelesaian seluruh kerja yang sama pada setiap variasi waktu siklus, jumlah stasiun kerja akan bertambah seiring dengan penurunan waktu siklus.

c. Waktu Siklus Maks

Pada Tabel 4.6 Dapat dilihat waktu siklus maskimal yang ada pada lini perakitan yang merupakan waktu siklus lini perakitan tersebut. Namu terdapat perbedaan antara waktu siklus yang diinginkan dengan waktu siklus maksimal yang ada pada lini perakitan tersebut.

Hal ini terjadi karena waktu siklus yang harapkan adalah waktu siklus maksimal yang menjadi acuan dalam penempatan elemen-elemen kerja, dengan kata lain bahwa waktu penyelesaian pengerjaan elemen-elemen kerja pada masing-masing stasiun kerja tidak boleh melewati waktu siklus yang telah ditentukan. Selain itu, adanya batasan berupa *precedence diagram* yang harus dipenuhi.

Tabel dibawah menunjukkan perbedaan terkecil yaitu pada waktu siklus 38.33 menit yang hanya memiliki perbedaan sebesar 0.13 menit, sedangkan perbedaan terbesar terjadi pada waktu siklus 41.82 menit yaitu sebesar 2.6 menit.

**Tabel 4.6.** *Gap* antara waktu siklus lini perkitan dengan waktu siklus max yang ada pada stasiun kerja.

<i>CT</i>	<i>CTMaks</i>	<i>Gap</i>
38.33	38.2	0.13
46	45.3	0.7
57.5	56.43	1.07
51.11	49.34	1.77
41.82	39.22	2.6

d. Effisiensi Lini Perakitan.

Percobaan dengan variasi waktu siklus menghasilkan variasi effisiensi pada lini perakitan. Pada tabel dibawah ini dapat

**Universitas Indonesia**

dilihat bahwa efisiensi lini perakitan terendah sebesar 0.7906 terjadi pada waktu siklus 46 menit sedangkan efisiensi terbesar berada pada lini perakitan dengan waktu siklus 57.5 menit yaitu sebesar 0.8697.

**Tabel 4.7.** Efisiensi lini perakitan tiap waktu siklus.

<i>CT</i>	<i>Effisiensi</i>
57.5	0.8697
51.11	0.818
38.33	0.8028
41.82	0.7971
46	0.7906

e. *Elapsed time.*

Waktu komputasi yang dibutuhkan untuk menyelesaikan permasalahan keseimbangan lini perakitan pada percobaan ini bervariasi antara 8.930505 detik yang terendah hingga 9.111389 detik.

## BAB V

### KESIMPULAN

Dari penelitian pada keseimbangan lini perakitan *Hydraulic Excavator Type PC300-8* yang telah dilakukan maka dapat diambil beberapa kesimpulan antara lain :

- Keseimbangan lini perakitan dengan menggunakan algoritma genetika memberikan performa lini perakitan yang lebih baik dibandingkan dengan keseimbangan lini perakitan yang telah diterapkan. Dengan penurunan jumlah stasiun kerja dari 10 stasiun kerja pada keseimbangan lini perakitan yang lama menjadi 9 stasiun kerja pada keseimbangan lini perakitan dengan penggunaan algoritma genetika. Selain itu, efisiensi lini perakitan meningkat dari 0.725 menjadi 0.802 serta penurunan *smoothnessindex* dari 151.86 menjadi 35.585
- Dengan menggunakan program algoritma genetika, dilakukan perancangan keseimbangan lini perakitan baru yang hendak dicapai yaitu lini perakitan dengan waktu siklus (*Cycle Time*) sebesar 38.33 menit terkait dengan output yang diinginkan yaitu kapasitas perakitan sebanyak 12 unit per hari memerlukan 13 stasiun kerja dengan waktu stasiun kerja terlama sebesar 37.04 menit yaitu pada stasiun kerja 8. Selain itu dari hasil penelitian juga didapatkan efisiensi lini perakitan sebesar 0.8028 serta *smoothnessindex* sebesar 37.8320.

## DAFTAR REFERENSI

Heizer J, Render B, "*Principles of Operations Management*", Pearson Education International, Sixth Edition.

Esmailian G.R, M Ahmad M. M. H, Sulaiman S, Ismail N, "*Assembly Line and Balancing Assembly Line*"

Nearchou Andreas C, "*Balancing large assembly lines by a new heuristic based on differential evolution method*", Springer-Verlag London Limited, 2006

Sabuncuoglu, Erel E, Tanyer M, "*Assembly line balancing using genetic algorithms*", Journal of Intelligent Manufacturing (2000) 11, 295-310

Vitaliy F, "*Differential Evolution, In Search of Solutions*", Springer Science + Business Media, LLC, 2006

Sabuncuoglu, Erel E., Tanyer M, "*Assembly line balancing using genetic algorithms*", Bilkent University, Bilkent, Ankara 06533, Turkey, 1998

Nearchou A.C, "*Balancing large assembly line by a new heuristic based on differential evolution method*", Springer-Verlag London Limited 2006

Erel S. E, Tanyer M., "*Assembly line balancing using genetic algorithms*", *Journal of Intelligent Manufacturing* (2000) 11, 295-310, Turkey 1998.

## Lampiran 1

### M-Skript Program Algoritma Genetika Keseimbangan Lini Perakitan

```
%=====LINEAR FITNESS RANKING=====
function LFR = LinearFitnessRanking(UkPop,Fitness,MaxF,MinF)
for ii=1:UkPop,
[SF,IndF] = sort(Fitness(:,:));
    LFR(IndF(UkPop-ii+1)) = MaxF-(MaxF-MinF)*((ii-1)/(UkPop-1));
end
%=====ROULETTE WHEEL=====
function Pindex = RouletteWheel(UkPop,LinearFitness)
JumFitness =sum(LinearFitness);
KumulatifFitness = 0;
RN =rand;
ii =1;
while ii<=UkPop,
    KumulatifFitness = KumulatifFitness + LinearFitness (ii);
    if (KumulatifFitness/JumFitness)>RN,
        Pindex =ii;
        break
    end
    ii = ii+1;
end
%=====PROGRAM UTAMA=====
clc          % Me-refresh command windows
clear       % Menghapus Semua Variable yang sedang aktif
TaskTime = [
8.09; 1.04; 5.07; 20.07; 15.04;
2.08; 7.07; 5.09; 3.04; 7.01;
21.01; 9.09; 1.09; 10.07; 4.04;
3.07; 6.02; 4.01; 5.04; 24.08;
4.02; 17.07; 3.04; 5.08; 19.02;
```

```

4.07; 10.07; 37.04; 2.02; 3.08;
34.04; 18.07; 6.07; 12.06; 3.02;
12.04; 19.04; 30.07 ]; % Waktu elemen pekerjaan

```

```

UkPop = 3;

```

```

% input('masukkan jumlah populasi awal yang diinginkan = ')

```

```

% Jumlah kromosom dalam populasi = 3

```

```

JumProdMin = 460; % waktu kerja

```

```

JumGen = length(TaskTime(:,1));

```

```

% Jumlah gen(jumlah elemen pekerjaan)

```

```

Psilang = 0.98; % Probabilitas pindah silang

```

```

MaxG = 100; % Jumlah generasi maksimum (iterasi)

```

```

CT = input('masukkan parameter waktu siklus = ')

```

```

% Waktu siklus yang diinginkan

```

```

tic % Awal waktu komputasional

```

```

%=====POPULASI=====

```

```

Populasi = [

```

```

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 ...

```

```

37 38 ;

```

```

1 2 3 4 5 8 ...

```

```

7 6 9 15 10 11 ...

```

```

12 13 14 16 17 18 ...

```

```

19 22 20 21 23 24 ...

```

```

26 25 27 28 29 30 ...

```

```

31 37 38 32 33 35 ...

```

```

34 36 ;

```

```

2 1 3 10 11 12 ...
13 4 5 15 16 17 ...
18 9 8 7 6 14 ...
19 20 21 22 23 24 ...
25 26 27 28 29 30 ...
31 32 37 38 33 36 ...
35 34];

```

```

% Loop evolusi
for generasi = 1:MaxG,
    for u = 1:UkPop, % Loop kromosom
        C = 0;
        A = 0;
        x = 0;
        T = [0];
        for ii = 1:JumGen, % Loop gen
            for j = (1 + A) : JumGen,
                x = x + TaskTime (Populasi(u,j),1);
                A = j - 1;
                if x > CT, % Waktu siklus yang diinginkan
                    x = x - TaskTime(Populasi(u,j),1);
                    C = 1;
                    break
                end
            end
        end
        T = [T x];
        if A == JumGen-1 && C == 1,
            T = [T TaskTime(A+1)];
            break
        end
        if A == JumGen-1 && C == 0,
            break
        end
    end
end

```

```

    end
    x = 0;
    C = 0;
end
% Waktu masing-masing stasiun kerja
T = T(1,2:(length(T)));
% Waktu maksimum stasiun kerja yang dihasilkan
Smax = max(T,[],2);
% Jumlah stasiun kerja
JumlahWorkStation = length(T);
k = 1;
bilang = 0;
while k <= JumlahWorkStation;
    bilang = bilang + (Smax - T(1,k))^2;
    k = k + 1;
end
k = 1;
bilang1 = 0;
while k <= JumlahWorkStation;
    bilang1 = bilang1 + (Smax - T(1,k));
    k = k + 1;
end
fitness=2*((bilang/JumlahWorkStation)^0.5)+(bilang1/JumlahWorkStation);
% Nilai fitness masing-masing kromosom
Fitness(u,1) = 1 / fitness;
MaxF=Fitness(1);
MinF=MaxF;
IndeksIndividuTerbaik=1;
if (Fitness(u)>MaxF),
    MaxF=Fitness(u);
    IndeksIndividuTerbaik=u;
% Jalur terbaik yang dihasilkan

```

```

        JalurTerbaik = Populasi(u,:);
    end
    if (Fitness(u)<=MinF),
        MinF=Fitness(u);
    end
    MaxF;
end
FitnessRata2=mean(Fitness(:,:));
TemPopulasi = Populasi;
% Elitisme:
% Buat satu kromosom terbaik jika ukuran populasi ganjil
% Buat dua kromosom terbaik jika ukuran populasi genap
% TemPopulasi = Populasi sementara
if mod(UkPop,2)==0,
    IterasiMulai = 3; %Jika Populasi Genap%
    TemPopulasi(1,:) = Populasi(IndeksIndividuTerbaik,:);
    TemPopulasi(2,:) = Populasi(IndeksIndividuTerbaik,:);
else
    IterasiMulai = 2; %Jika Populasi Ganji%
    TemPopulasi(1,:) = Populasi(IndeksIndividuTerbaik,:);
end

%=====NORMALISASI FITNESS VALUE=====
LinearFitness = LinearFitnessRanking(UkPop,Fitness,MaxF,MinF);

%=====SELEKSI ORANG TUA / ROULETTE-WHEEL=====
for jj = IterasiMulai:2:UkPop,
    Bapak = RouletteWheel(UkPop,LinearFitness);
    Ibu = RouletteWheel(UkPop,LinearFitness);
    if Bapak==Ibu,
        Ibu = RouletteWheel(UkPop,LinearFitness);
    end
end

```

`%=====PINDAH SILANG=====%`

```
if (rand < Psilang),
    TP1 = 1+fix(rand*(JumGen-1)); %TP = Titik Potong%
    TP2 = 1+fix(rand*(JumGen-1));
while TP2==TP1,
    TP2 = 1+fix(rand*(JumGen-1));
end
if TP1 < TP2,
    CP1=TP1; %CP=Cross Point%
    CP2=TP2;
else
    CP1=TP2;
    CP2=TP1;
end
PanjangPotong = CP2 - CP1 + 1;
Bapak1=Populasi(Bapak,:);
Ibu1=Populasi(Ibu,:);
Anak1=[Bapak1(1,1:CP1-1) zeros(1, PanjangPotong) Bapak1(1,
CP2+1:JumGen)];
Anak2=[Ibu1(1,1:CP1-1) zeros(1, PanjangPotong) Ibu1(1, CP2+1:JumGen)];
m = CP1;
k = CP1;
for ii=1:JumGen
    if ~ismember(Ibu1(ii), Anak1(1,1:JumGen));
        Anak1(1,m)=Ibu1(ii);
        m = m + 1;
    end
    if ~ismember(Bapak1(ii), Anak2(1,1:JumGen));
        Anak2(1,k) = Bapak1(ii);
        k = k + 1;
    end
end
end
```

```

Anak1;
Anak2;
Anak=[Anak1; Anak2];
TemPopulasi(jj,:) = Anak(1,:);
TemPopulasi(jj+1,:) = Anak(2,:);
else
    TemPopulasi(jj,:) = Populasi(Bapak,:);
    TemPopulasi(jj+1,:) = Populasi(Ibu,:);
end
end

%=====PENGGANTIAN POPULASI=====
% Generational Replacement : mengganti semua kromosom sekaligus
Populasi = TemPopulasi; % Populasi baru
end
JalurTerbaik % Jalur terbaik yang dihasilkan
C = 0;
A = 0;
x = 0;
T = [0];
for ii = 1:JumGen,
    for j = (1 + A):JumGen,
        x = x + TaskTime(JalurTerbaik(j),1);
        A = j - 1;
        if x > CT,
            x = x - TaskTime(JalurTerbaik(j),1);
            C = 1;
            break
        end
    end
end
T = [T x];
if A == JumGen-1 && C == 1,

```

```

    T = [T TaskTime(A+1)];
    break
end
if A == JumGen-1 && C == 0,
    break
end
    x = 0;
    C = 0;
end
T = T(1,2:(length(T)))
Smax = max(T,[],2)
JumlahWorkStation = length(T)
k = 1;
bilang = 0;
while k <= JumlahWorkStation;
    bilang = bilang + (Smax - T(1,k))^2;
    k = k + 1;
end
k = 1;
bilang3 = 0;
while k <= JumlahWorkStation;
    bilang3 = bilang3 + T(1,k);
    k = k + 1;
end
k = 1;
bilang1 = 0;
while k <= JumlahWorkStation;
    bilang1 = bilang1 + (Smax - T(1,k));
    k = k + 1;
end

```

```
fitness = 2*((bilang/JumlahWorkStation)^0.5) + (bilang1/JumlahWorkStation);
```

```
Fitness = 1 / fitness;
```

```
SmoothnessIndex = bilang^0.5 % Nilai smoothness
```

```
LineEfficiency = bilang3/(JumlahWorkStation*CT)% Efisiensi lintasan
```

```
JumProduksi = round (JumProdMin/CT) % Jumlah Produksi perhari
```

```
toc % Akhir waktu komputasional
```



**Lampiran 2**  
**Hasil perhitungan Keseimbangan Lini perakitan Dengan Menggunakan**  
**Program Algoritma Genetika**

CT = 38.3300

JalurTerbaik =

Columns 1 through 10

1 2 3 10 11 12 4 5 15 13

Columns 11 through 20

16 17 18 8 9 7 6 14 19 22

Columns 21 through 30

20 21 23 24 25 26 27 28 29 30

Columns 31 through 38

31 32 33 34 35 36 37 38

T =

Columns 1 through 6

21.2100 30.1000 35.1100 35.5100 32.1800 36.2200

Columns 7 through 12

33.1600 37.0400 5.1000 34.0400 36.2000 34.1000

Column 13

30.0700

Smax =

37.0400

JumlahWorkStation =

13

SmoothnessIndex =

37.8320

LineEfficiency =

0.8028

JumProduksi =

12

Elapsed time is 8.672024 seconds.

CT = 40

JalurTerbaik =

Columns 1 through 10

1 2 3 4 5 6 7 8 10 11

Columns 11 through 20

12 13 15 16 17 18 9 14 19 20

Columns 21 through 30

21 22 23 24 25 26 27 28 29 30

Columns 31 through 38

31 32 33 34 35 36 37 38

T =

Columns 1 through 6

34.2700 36.2900 38.3000 28.1800 28.1000 25.1900

Columns 7 through 12

33.1600 39.0600 37.1200 39.2200 31.0800 30.0700

Smax =

39.2200

JumlahWorkStation =

12

SmothnessIndex =

25.8351

LineEfficiency =

0.8334

JumProduksi =

12

Elapsed time is 8.814909 seconds.

CT = 41.8200

JalurTerbaik =

Columns 1 through 10

1 2 3 4 5 10 11 12 9 8

Columns 11 through 20

7 6 13 14 15 16 17 18 19 20

Columns 21 through 30

21 22 23 24 25 26 27 28 29 30

Columns 31 through 38

31 32 33 35 34 36 37 38

T =

Columns 1 through 6

34.2700 22.0500 38.2300 37.4500 33.1400 25.1900

Columns 7 through 12

33.1600 39.0600 37.1200 39.2200 31.0800 30.0700

Smax =

39.2200

JumlahWorkStation =

12

SmothnessIndex =

27.3564

LineEfficiency =

0.7971

JumProduksi =

11

Elapsed time is 8.930505 seconds.

CT = 46

JalurTerbaik =

Columns 1 through 10

1 2 3 4 5 6 7 8 9 10

Columns 11 through 20

11 12 13 14 15 16 17 18 19 22

Columns 21 through 30

20 21 23 24 26 25 27 28 29 30

Columns 31 through 38

31 32 33 34 35 36 37 38

T =

Columns 1 through 6

34.2700 39.3300 45.3000 35.2100 40.2900 29.0900

Columns 7 through 11

42.1400 34.0400 39.2200 31.0800 30.0700

Smax =

45.3000

JumlahWorkStation =

11

SmothnessIndex =

33.9835

LineEfficiency =

0.7906

JumProduksi =

10

Elapsed time is 9.046424 seconds.

CT = 51.1100

JalurTerbaik =

Columns 1 through 10

1 2 3 4 5 8 7 6 9 15

Columns 11 through 20

10 11 12 13 14 16 17 18 19 22

Columns 21 through 30

20 21 23 24 25 26 27 28 29 30

Columns 31 through 38

31 32 33 35 37 38 34 36

T =

Columns 1 through 6

49.3100 49.3400 38.3900 48.2100 38.2400 42.1400

Columns 7 through 10

34.0400 46.2000 42.1300 30.0700

Smax =

49.3400

JumlahWorkStation =

10

SmothnessIndex =

31.0402

LineEfficiency =

0.8180

JumProduksi =

9

Elapsed time is 9.014913 seconds.

CT = 55.19

JalurTerbaik =

Columns 1 through 10

2 1 3 10 11 12 13 4 5 15

Columns 11 through 20

16 17 18 9 8 7 6 14 19 22

Columns 21 through 30

20 21 23 24 25 26 27 28 29 30

Columns 31 through 38

31 32 37 38 33 36 35 34

T =

Columns 1 through 6

52.4000 52.2500 49.4600 36.2200 33.1600 42.1400

Columns 7 through 9

52.1100 49.1100 33.1900

Smax =

52.4000

JumlahWorkStation =

9

SmoothnessIndex =

33.5535

LineEfficiency =

0.8082

JumProduksi =

8

Elapsed time is 8.950627 seconds.

CT = 57.5000

JalurTerbaik =

Columns 1 through 10

1 2 3 10 11 12 13 4 5 6

Columns 11 through 20

7 8 9 15 16 17 18 14 19 20

Columns 21 through 30

21 22 23 24 25 26 27 28 29 30

Columns 31 through 38

31 32 33 36 37 38 35 34

T =

Columns 1 through 6

52.4000 56.4300 56.3100 48.2800 52.2100 52.1100

Columns 7 through 8

37.1500 45.1500

Smax =

56.4300

JumlahWorkStation =

8

SmothnessIndex =

24.8617

LineEfficiency =

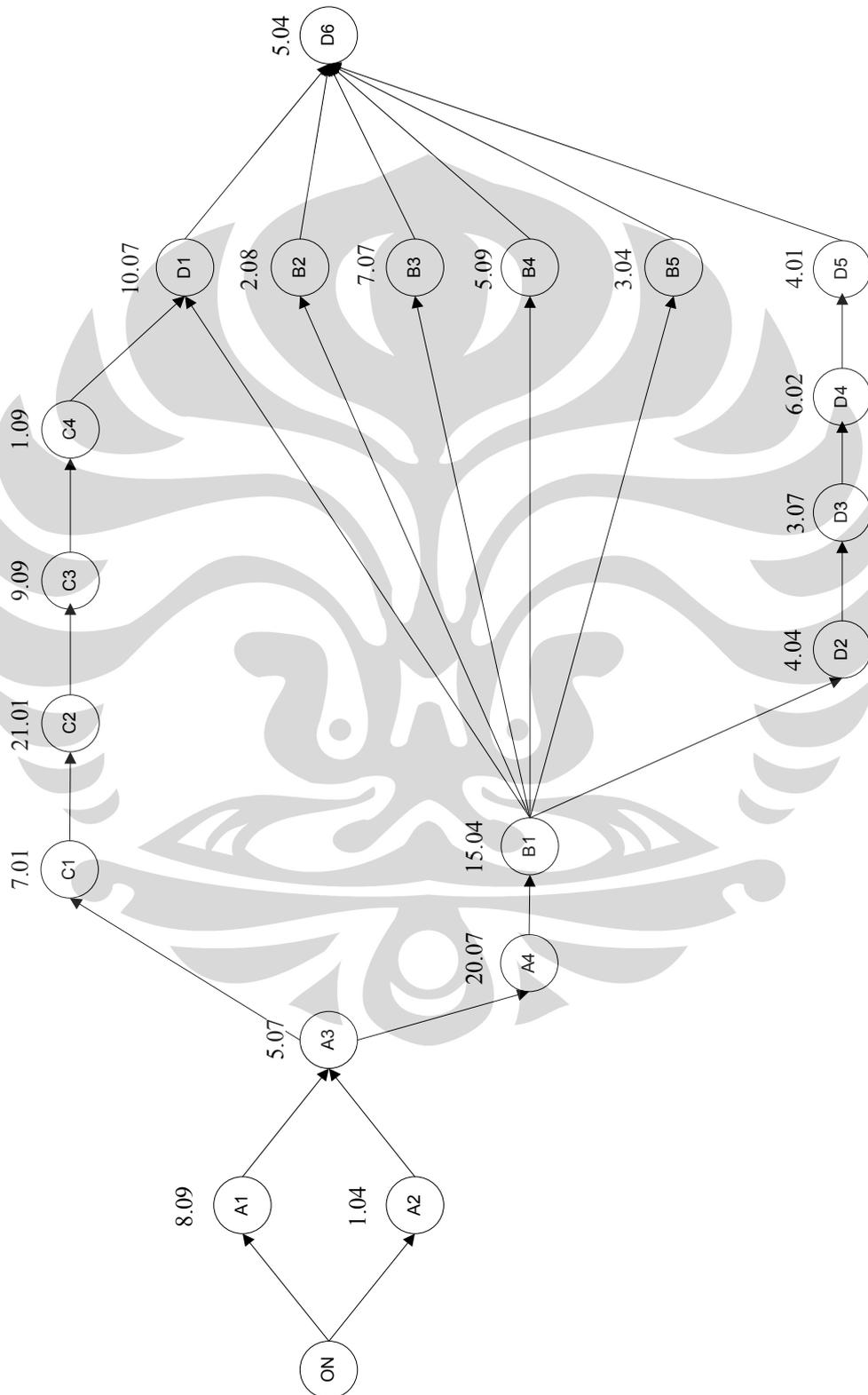
0.8697

JumProduksi =

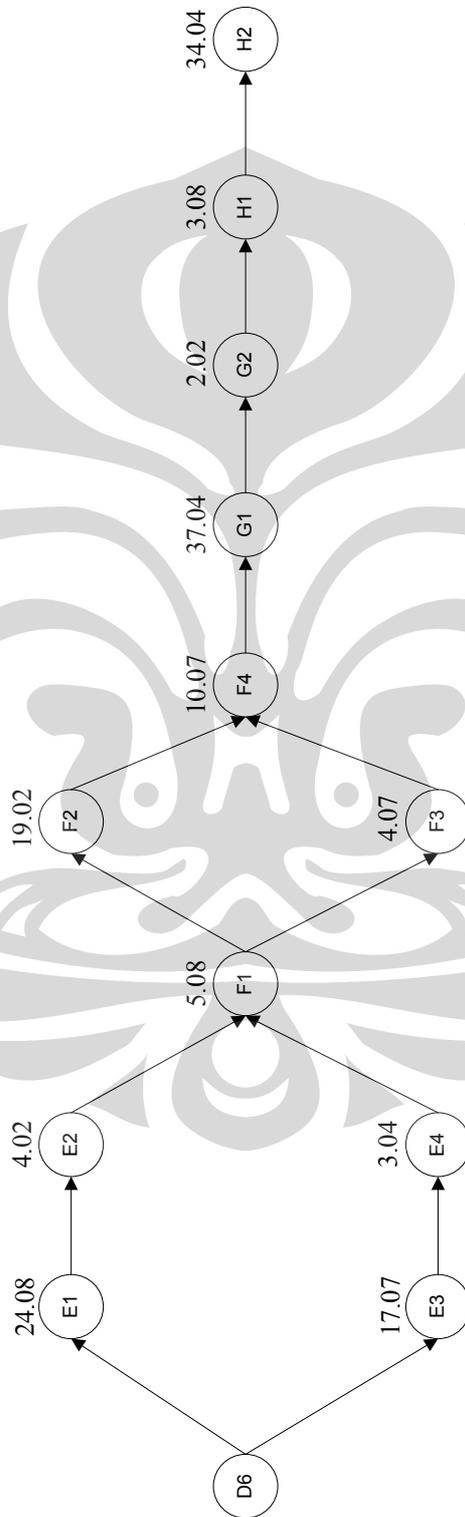
8

Elapsed time is 9.111389 seconds.

**Lampiran 3**  
**Precedence Diagram Perakitan Hydraulic Excavator Tipe PC300-8**



**Precedence Diagram Perakitan Hydraulic Excavator Tipe PC300-8  
(lanjutan)**



**Precedence Diagram Perakitan Hydraulic Excavator Tipe PC300-8  
(lanjutan)**

