



**UNIVERSITAS INDONESIA**

**OPTIMASI PENJADWALAN FLEXIBLE FLOW SHOP  
DENGAN ALGORITMA GENETIKA PADA INDUSTRI  
PEMBUATAN SILICON POLISHED WAFER**

**SKRIPSI**

**EKO HERU SARTONO  
0606043502**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
DEPARTEMEN TEKNIK INDUSTRI  
DEPOK  
DESEMBER 2008**



**UNIVERSITAS INDONESIA**

**OPTIMASI PENJADWALAN FLEXIBLE FLOW SHOP  
DENGAN ALGORITMA GENETIKA PADA INDUSTRI  
PEMBUATAN SILICON POLISHED WAFER**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**EKO HERU SARTONO  
0606043502**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
DEPARTEMEN TEKNIK INDUSTRI  
DEPOK  
DESEMBER 2008**

## **PERNYATAAN KEASLIAN SKRIPSI**

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

### **OPTIMASI PENJADWALAN FLEXIBLE FLOW SHOP DENGAN ALGORITMA GENETIKA PADA INDUSTRI PEMBUATAN SILICON POLISHED WAFER**

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Industri Departemen Teknik Industri Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 23 Desember 2008

( Eko Heru Sartono )  
NPM 0606043502

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : EKO HERU SARTONO

NPM : 0606043502

Program Studi : Teknik Industri

Judul Skripsi : **Optimasi Penjadwalan Flexible Flow Shop Dengan Algoritma Genetika pada Industri Pembuatan Silicon Polished Wafer**

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

### DEWAN PENGUJI

Pembimbing : Ir. Amar Rachman, MEIM ( )

Penguji : Ir. Isti Surjandari, MT, MA, Ph.D ( )

Penguji : Ir. Yadrifil, M.Sc ( )

Penguji : Ir. Erlinda Muslim, MEE ( )

Ditetapkan di : Depok

Tanggal : 23 Desember 2008

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Allah SWT karena atas rahmat, dan ridho-Nya akhirnya penyusunan skripsi ini dapat diselesaikan. Penulis menyadari bahwa skripsi ini tidak akan dapat dibuat tanpa bantuan dan bimbingan dari berbagai pihak. Karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

- Ir. Amar Rachman, MEIM, selaku dosen pembimbing yang telah banyak memberi bantuan, masukan dan bimbingan yang berharga bagi penulis.
- Segenap karyawan SPTI yang telah memberikan kesempatan kepada penulis untuk mengumpulkan data untuk penelitian ini dan memberikan jawaban atas pertanyaan-pertanyaan yang sering penulis tanyakan.
- Keluarga, atas curahan kasih sayang, dukungan, dan doa yang diberikan.
- Teman-teman penulis, khususnya rekan-rekan TIUI 2006 yang telah memberikan dukungan, semangat, serta kebersamaan selama empat tahun ini.
- Pihak-pihak lain yang juga telah membantu penyelesaian skripsi ini namun tidak dapat disebutkan satu per satu.

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

Depok, 23 Desember 2008

Penulis

**LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI  
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS**

---

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

Nama : EKO HERU SARTONO  
NPM : 0606043502  
Program Studi : Teknik Industri  
Fakultas : Teknik  
Jenis karya : Skripsi

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

**Optimasi Penjadwalan Flexible Flow Shop Dengan Algoritma Genetika pada  
Industri Pembuatan Silicon Polished Wafer**

Beserta perangkat yang ada (bila diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah ini menjadi tanggung jawab saya pribadi.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada tanggal : 23 Desember 2008  
Yang menyatakan

(Eko Heru)

## RIWAYAT HIDUP PENULIS

Nama : EKO HERU SARTONO  
Tempat / Tanggal Lahir : Boyolali, 02 Mei 1979  
Alamat : Graha Asri Residence E10/10, Cikarang Barat,  
Bekasi, Jawa Barat  
No. Telepon : 021 93537801 / 085216062656  
Email : [heru2100@yahoo.co.id](mailto:heru2100@yahoo.co.id)  
Pendidikan :

a.	SD	MIN Tinawas Boyolali (1985-1991)
b.	SLTP	SMP Al-Islam I Surakarta (1991-1994) MTs Al-Islam I Surakarta (1991-1994)
c.	SLTA	STM Katolik Mikail Surakarta (1994-1997)
d.	Diploma	Teknik Mesin UGM Yogyakarta (1997-2000)
e.	Sarjana	Teknik Industri UI Jakarta (2006- 2008)

## ABSTRAK

Nama : EKO HERU SARTONO  
Program Studi : Teknik Industri  
Judul : Optimasi Penjadwalan Flexible Flow Shop Dengan  
Algoritma Genetika pada Industri Pembuatan Silicon  
Polished Wafer

Penelitian ini mengenai pemodelan untuk penjadwalan proses produksi pada industri pembuatan *silicon polished wafer 5"* dan *6"*. Proses produksi untuk kedua produk ini memiliki tipikal *flexible flow shop* dengan variasi type barang mencapai 12 jenis. Setiap type barang harus melalui empat tahapan proses yang harus dilakukan secara berurut. Adanya limitasi fleksibilitas proses pada tiga buah tahapan menyebabkan masalah penjadwalan menjadi kompleks.

Pendekatan yang dilakukan untuk memecahkan masalah penjadwalan ini adalah dengan menerapkan dua metode *dispatching rule*, yaitu SPT (*shortest processing time*) dan EFT (*earliest finish time*). Metode lain yang digunakan adalah pengelompokan unit pesanan ke dalam sebuah *family* untuk meminimalkan kendala limitasi fleksibilitas proses pada dua buah tahapan. Kombinasi ketiga metode ini dituangkan ke dalam suatu algoritma *multiproduct-multistage* dan diaplikasikan pada inisialisasi solusi. Dari inisialisasi solusi ini dihasilkan penjadwalan awal untuk masing-masing tahapan.

Optimasi dilakukan terhadap inisialisasi solusi dengan menggunakan Algoritma Genetika. Algoritma Genetika akan mencari solusi terbaik yang mendekati optimal melalui serangkaian proses seleksi terhadap sekumpulan alternatif solusi yang ada.

Kata kunci:

*Scheduling, flexible flow shop, multi-product, multistage, dispatching rule, Genetic Algorithm*



## ABSTRACT

Name : EKO HERU SARTONO  
Study Program : Industrial Engineering  
Title : Optimize Flexible Flow Shop Scheduling Using Genetic Algorithm in Silicon Polished Wafer Manufacturing

The study concerns in modeling the production schedule of 5” and 6” Silicon Polished Wafer manufacturing. The process of both products is categorized as flexible flow shop with 12 different types of product. Each product is processed through four stages with fixed sequence. There are some limitation of process flexibility among three of four stages that also causes the scheduling become difficult to arrange.

Some dispatching rule is applied to solve the schedule problem. SPT (shortest processing time) and EFT (earliest finish time) is the two of dispatching rule which used for this kinds of problem. The other method that also carried out with those two dispatching rule is job families approaching. Job families approaching will minimize the effect of limited process flexibility. Those combined methods will be developed to be an algorithm called multiproduct-multistage algorithm and applied to the initial solution. The initial solution consists of the schedule of each stage.

Initial solution is optimized by using Genetic Algorithm. Genetic Algorithm is search process of the best solutions among possible solutions by simulating the natural evolutionary process.

Keywords:

Scheduling, flexible flow shop, multi-product, multistage, dispatching rule, Genetic Algorithm

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>PERNYATAAN KEASLIAN SKRIPSI</b> .....	ii
<b>HALAMAN PENGESAHAN</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH</b> .....	v
<b>RIWAYAT HIDUP PENULIS</b> .....	vi
<b>ABSTRAK</b> .....	vii
<b>ABSTRACT</b> .....	viii
<b>DAFTAR ISI</b> .....	ix
<b>DAFTAR GAMBAR</b> .....	xi
<b>DAFTAR TABEL</b> .....	xii
<b>DAFTAR LAMPIRAN</b> .....	xiii
<b>1. PENDAHULUAN</b>	
1.1. Latar Belakang Masalah.....	2
1.2. Diagram Keterkaitan Masalah.....	3
1.3. Perumusan Permasalahan.....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
1.6. Ruang Lingkup Penelitian.....	4
1.7. Metodologi Penelitian.....	5
1.8. Sistematika Penulisan.....	7
<b>2. DASAR TEORI</b>	
2.1. Penjadwalan .....	8
2.1.1. Faktor Utama Penjadwalan.....	9
2.1.2. Penjadwalan Flexible Flow Shop.....	11
2.2. Genetic Algorithm.....	13
2.2.1. Struktur Umum Genetic Algorithm.....	13
2.2.2. Komponen-komponen Utama dalam Genetic Algorithm .....	15
2.2.2.1. Teknik Penyandian.....	15
2.2.2.2. Penentuan Parameter.....	16
2.2.2.3. Prosedur Inisialisasi.....	16
2.2.2.4. Fungsi Evaluasi.....	16
2.2.2.5. Seleksi.....	16
2.2.2.6. Operator Genetika.....	17
2.2.2.7. Penggantian Populasi.....	18
<b>3. PENGUMPULAN DATA</b>	
3.1. Profil Perusahaan.....	19
3.2. Pengumpulan Data Penelitian.....	19
3.2.1. Data Kapasitas Waktu Produksi.....	19
3.2.2. Rute Proses Produksi <i>Silicon Polished Wafer</i> .....	20
3.2.3. Waktu Baku dan Spesifikasi Produk.....	21
3.2.4. Kegiatan Perencanaan Produksi.....	23
3.2.5. Data Pemesanan.....	24

<b>4. PENGOLAHAN DATA DAN ANALISIS</b>	
4.1. Fungsi Tujuan.....	25
4.2. Inisialisasi Solusi.....	26
4.2.1. Menetapkan Metode Penjadwalan.....	27
4.2.1.1. Metode Penjadwalan Tahapan 1.....	27
4.2.1.2. Metode Penjadwalan Tahapan 2.....	29
4.2.1.3. Metode Penjadwalan Tahapan 3.....	29
4.2.1.4. Metode Penjadwalan Tahapan 4.....	30
4.2.2. Menyusun Penjadwalan Dengan Algoritma <i>Multistage-Multiproduct</i> .....	30
4.3. Verifikasi Program.....	38
4.3.1. Penjadwalan dan Perhitungan Manual.....	38
4.3.1.1. Penjadwalan Tahapan 1.....	38
4.3.1.2. Penjadwalan Tahapan 2.....	39
4.3.1.3. Penjadwalan Tahapan 3.....	40
4.3.1.4. Penjadwalan Tahapan 4.....	41
4.3.1.5. Menghitung Makespan.....	42
4.4. Penjadwalan Dengan Genetic Algorithm.....	43
4.4.1. Penetapan Parameter Genetic Algorithm.....	46
4.4.2. Pengkonversian Masalah.....	46
4.4.3. Penentuan Populasi awal.....	47
4.4.4. Evaluasi Kromosom.....	47
4.4.5. Seleksi.....	48
4.4.6. Kawin Silang (crossover).....	48
4.4.7. Mutasi.....	48
4.4.8. Pengolah algoritma genetika pada penjadwalan.....	48
4.5. Analisis.....	49
4.5.1. Analisis Metode.....	50
4.5.2. Analisis Hasil.....	52
<b>5. KESIMPULAN</b> .....	53
<b>DAFTAR REFERENSI</b> .....	54

## DAFTAR GAMBAR

Gambar1.1. Diagram Keterkaitan masalah .....	3
Gambar1.2. Diagram alir metodologi penelitian .....	6
Gambar2.1. Sistem informasi manufacter .....	9
Gambar2.2. Prosedur algoritma genetika secara umum .....	15
Gambar2.3. Contoh kawin silang .....	17
Gambar2.4. Contoh mutasi .....	17
Gambar3.1. Proses produksi silicon polished wafer .....	21
Gambar4.1. Diagram alir algoritma <i>multiproduct-milti stage</i> .....	31
Gambar4.2. Algoritma genetika untuk penjadwalan .....	44

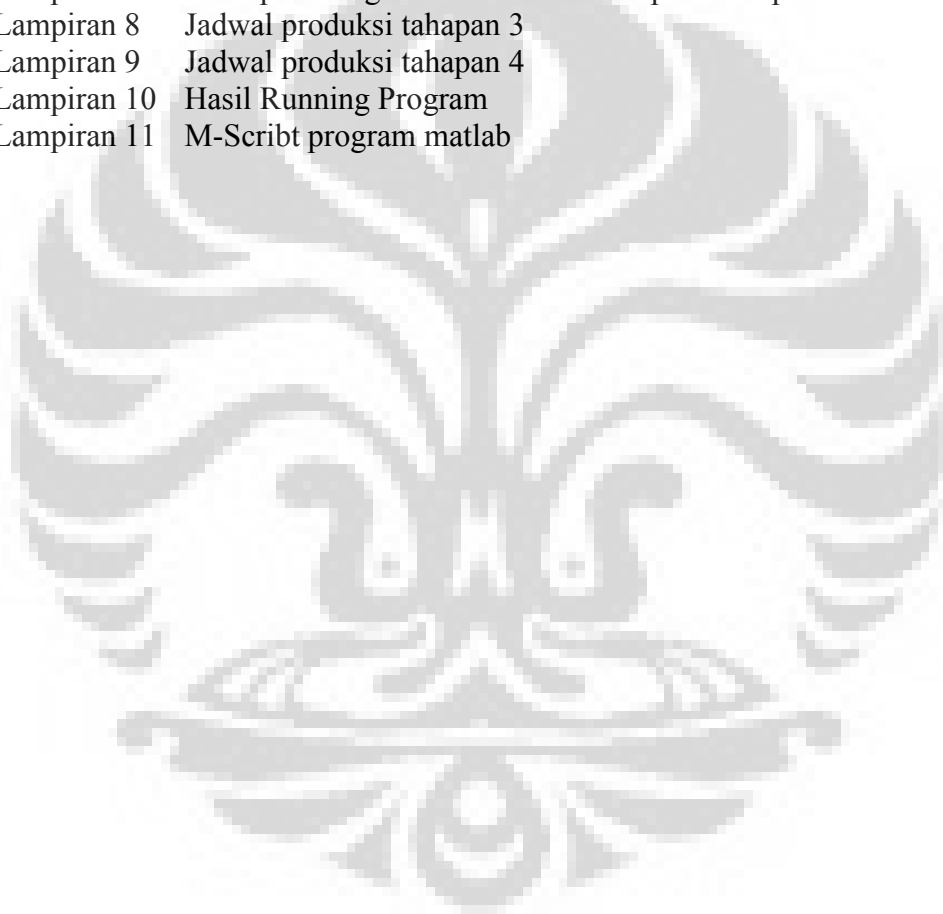


## DAFTAR TABEL

Tabel 3.1.	Tabel jam kerja di SPTI .....	20
Tabel 3.2.	Waktu baku untuk semua type.....	22
Tabel 3.3.	Spesifikasi dan nilai proses per individu .....	22
Tabel 3.4.	Jumlah pesanan <i>silicon polished wafer 5"</i> dan <i>6"</i> .....	24
Tabel 4.1.	Data jumlah pemesanan untuk inialisasi solusi .....	26
Tabel 4.2.	Pengelompokan unit pesanan menjadi job .....	28
Tabel 4.3.	Penetapan metode penjadwalan untuk setiap tahapan .....	30
Tabel 4.4.	Urutan Pengerjaan Job Pada Tahapan 1 .....	39
Tabel 4.5.	Penetapan unit pertama untuk setiap <i>workstation</i> pada tahapan 2 .....	39
Tabel 4.6.	<i>Finish time</i> (FT) unit ke-1 dari setiap <i>workstation</i> pada tahapan 2 .....	40
Tabel 4.7.	Hasil pemeringkatan FT pada tahapan 2 .....	40
Tabel 4.8.	Penetapan unit pertama untuk setiap mesin pada tahapan 3 .....	41
Tabel 4.9.	Rekomendasi parameter Algoritma Genetika.....	45
Tabel 4.10	Hasil running program Matlab .....	49

## DAFTAR LAMPIRAN

- Lampiran 1 Standard time untuk setiap job pada setiap tahapan
- Lampiran 2 Data total waktu proses pada tahapan 1
- Lampiran 3 Pemeringkatan job berdasarkan total waktu proses pada tahapan 1
- Lampiran 4 Jadwal produksi tahapan 1
- Lampiran 5 Hasil pemeringkatan FT semua unit pada tahapan 1
- Lampiran 6 Jadwal produksi tahapan 2 untuk setiap *workstation*
- Lampiran 7 Hasil pemeringkatan FT semua unit pada tahapan 2
- Lampiran 8 Jadwal produksi tahapan 3
- Lampiran 9 Jadwal produksi tahapan 4
- Lampiran 10 Hasil Running Program
- Lampiran 11 M-Script program matlab



## **BAB I**

### **PENDAHULUAN**

Kompetisi di dunia usaha, terutama pada industri manufaktur saat ini tidak hanya memperhatikan mutu dan harga tetapi juga pengiriman. Ketiga faktor ini sering diistilahkan sebagai QCD yaitu kependekan dari *quality*, *cost* dan *delivery*, dan saat ini QCD menjadi salah satu kebijakan perusahaan dalam upaya memenangkan persaingan.

*Delivery* (pengiriman) memiliki pengertian sebagai suatu komitmen untuk memenuhi keinginan konsumen terhadap ketersediaan barang pada waktu dan moda pengiriman yang telah ditentukan. Penetapan waktu dan moda pengiriman ditetapkan di awal penerimaan pesanan dan merupakan kesepakatan antara konsumen dan produsen. Dalam sebuah organisasi modern, penetapan waktu pengiriman yang diinginkan oleh konsumen akan dikomunikasikan terlebih dahulu dengan bagian-bagian yang terkait (bagian pemasaran, produksi, engineering, purchasing, dll). Kegiatan ini disebut *contract review*. Pada kegiatan *contract review*, bagian produksi harus dapat menghitung kemampuan/kapasitas produksi apabila menerima pesanan tersebut. Penghitungan kemampuan berproduksi harus didasarkan pada kondisi paling aktual yaitu jumlah barang yang harus diproduksi, waktu baku produksi untuk setiap jenis barang, jumlah sumber (*resource*) yang dimiliki dan waktu produksi yang tersedia. Hasil perhitungan tersebut menjadi dasar untuk menerima, menunda atau menolak pesanan konsumen.

Ketika perusahaan memutuskan untuk menerima pesanan dari konsumen, maka organisasi berkewajiban untuk memenuhi komitmennya kepada konsumen untuk membuat dan mengirimkannya sesuai dengan kualitas, harga dan waktu pengiriman yang disepakati. Bagian produksi sebagai salah satu bagian dari organisasi berkewajiban untuk memproduksi barang sesuai dengan tenggat waktu yang ditetapkan. Untuk memastikan bahwa kegiatan produksi tidak akan melewati tenggat waktu yang telah ditetapkan, disusun sebuah skema penjadwalan produksi yang menjadi acuan aktifitas berproduksi. Penjadwalan produksi juga merupakan dasar pembagian tugas kerja untuk setiap unit kerja. Penjadwalan produksi harus menggunakan metode yang tepat sehingga dihasilkan suatu nilai maksimal dari

pemakaian sumber (*resource*) yang dimiliki untuk memenuhi tenggat waktu produksi. Pada beberapa perusahaan penjadwalan produksi dilakukan oleh scheduler atau bagian PPIC, tetapi untuk beberapa perusahaan penjadwalan produksi diserahkan sepenuhnya kepada bagian produksi.

### **1.1. Latar Belakang Permasalahan**

Sebagai salah satu perusahaan manufaktur yang bergerak di bidang produksi *silicon polished wafer*, SPTI yang juga memasok material *semiconductor* ke beberapa perusahaan sejenis, memiliki permasalahan dalam penjadwalan produksi dalam memenuhi tenggat waktu pengiriman yang diinginkan konsumen. Saat ini, tidak ada metode baku yang dilakukan untuk penjadwalan produksi. Bagian Produksi menjadwalkan kegiatan produksinya secara trial and error berdasarkan pengalaman. Metode ini menyebabkan kesulitan dalam pembagian tugas dan alokasi sumber karena tidak memperhitungkan kapasitas sumber (*resource*) dengan cermat. Permasalahan ini muncul karena adanya kompleksitas antara beberapa proses yang memiliki karakter proses dan kemampuan berproduksi yang berbeda di dalam satu antrian proses. Proses pertama adalah proses *slicing* yaitu kegiatan memotong batangan silindris dari kristal silika (*silicon ingot x-tal*) menjadi lembaran tipis sekitar 700 mikron. Proses selanjutnya adalah proses *edge grinding*, dimana barang yang sudah Rajang menjadi lembaran bundar dengan diameter 125 mm sampa dengan 150 mm digerinda sudut tepinya hingga menghasilkan sisi yang halus untuk menghindari keetakan saat diproses pada tahap berikutnya. Hasil dari pekerjaan ini diteruskan ke proses berikutnya yaitu proses *lapping*, yaitu proses penghalusan permukaan dan juga pembentukan ketebalan sesuai dengan spesifikasi yang diinginkan konsumen. Setelah tahap ketiga dilalui maka tahap selanjutnya adalah proses penghalusan permukaan dengan zat kimia atau *chemical etching*.

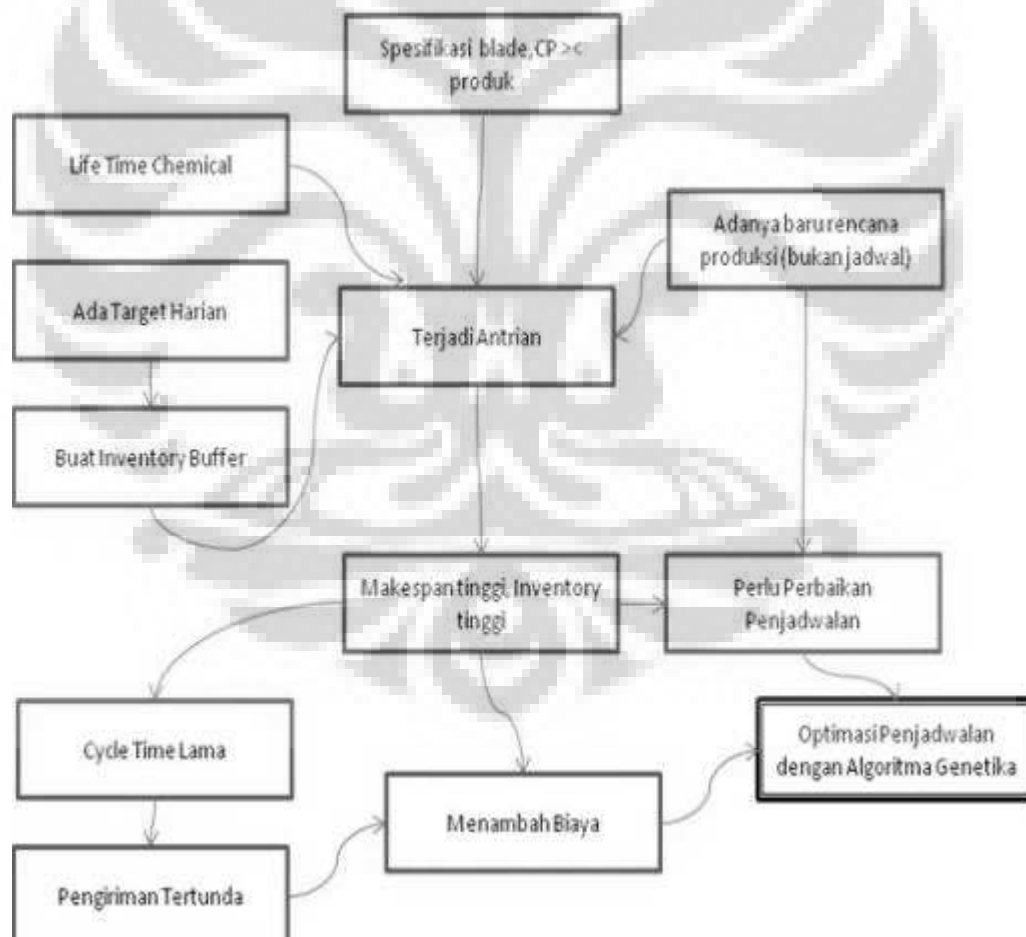
Kondisi yang dihadapi saat ini adalah keterbatasan fleksibilitas proses produksi. Pada tahapan kesatu, yaitu proses *slicing*, proses produksi dilakukan dengan bantuan seperangkat peralatan *x-ray* untuk memantau sudut pemotongan maupun master wafer sebagai referensi selama proses. Setiap type barang memiliki spesifikasi pisau yang berbeda antara satu type dengan type lainnya.



Demikian pula dengan proses di tahapan ketiga, yaitu proses *lapping*, yang memiliki *carrier plate (cp)* yang berbeda untuk setiap type barang. Pergantian pisau maupun *carrier plate* akan mengakibatkan bertambahnya waktu idle sehingga berpotensi menimbulkan keterlambatan apabila pengaturan pergantian type barang tidak dilaksanakan. Jumlah pesanan yang bervariasi dan waktu proses yang berbeda antara satu type dengan type yang lain dan perbedaan waktu proses antara tahapan yang satu dengan tahapan yang lain membuat penjadwalan sulit untuk dilakukan.

## 1.2. Diagram Keterkaitan Masalah

Untuk memperlihatkan keterkaitan masalah tersebut secara sistemik maka dibuat sebuah diagram seperti di bawah ini.



Gambar 1.1. Diagram keterkaitan masalah

### 1.3. Perumusan Permasalahan

Berdasarkan latar belakang masalah yang telah dijelaskan sebelumnya dan melihat keterkaitan masalah seperti tersusun dalam diagram di atas, maka pokok masalah yang akan dibahas dalam penelitian ini adalah melakukan optimasi dan membuat model penjadwalan produksi yang tepat dengan kendala-kendala yang ada.

### 1.4. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah untuk mendapatkan suatu penjadwalan produksi yang paling tepat dalam hal optimasi penggunaan sumber (resource) yang dimiliki sesuai dengan waktu pengiriman yang telah ditetapkan. Perhitungan dan penyusunan jadwal produksi dilakukan dengan menggunakan metode Algoritma Genetika (AG) yang diaplikasikan dengan menggunakan perangkat lunak MATLAB.

### 1.5. Manfaat Penelitian

Dari hasil penelitian ini diharapkan:

1. Dapat dibuat satu model penjadwalan dengan metode yang tepat
2. Dapat dibuat suatu penjadwalan produksi yang terintegrasi untuk semua produk dalam semua proses produksi
3. Penjadwalan produksi dapat dilakukan dengan cepat.

### 1.6. Ruang Lingkup Penelitian

Ruang lingkup penelitian ini hanya meliputi:

1. Proses pembuatan silicon polished wafer dimensi 5" dan 6" di bagian tahap awal produksi yaitu antara bagian *slicing* sampai dengan *chemical etching* di SPTI
2. Data produksi diambil dari pemesanan bulan Agustus 2008
3. Waktu baku produksi diambil dari motion study dan standard time yang dikeluarkan oleh bagian PPIC di SPTI

4. Perhitungan dilakukan dengan asumsi:
  - a. Waktu proses sudah termasuk waktu set up
  - b. Kondisi mesin dianggap stabil
  - c. Setiap workstation/mesin hanya mengerjakan satu unit barang pada satu waktu dan tidak dapat diselingi oleh pekerjaan lain (non preemptive).
  - d. Tidak ada kendala dalam penyediaan material

### 1.7. Metodologi Penelitian

Dalam melakukan penelitian ini, dilakukan langkah-langkah metodologi sebagai berikut:

1. Mengidentifikasi masalah di SPTI
2. Melakukan studi literatur.  
Pengumpulan literatur yang berkaitan dengan sistem aliran produksi, metode-metode penjadwalan yang berhubungan dengan multiproduct dan multistage baik berupa buku-buku teori ataupun jurnal hasil penelitian.
3. Merumuskan masalah.  
Perumusan masalah ini adalah membuat model penjadwalan produksi yang tepat dengan kendala-kendala yang ada dengan menggunakan Genetic Algorithm.
4. Menentukan tujuan penelitian.  
Tujuan dari penelitian ini adalah untuk mendapatkan suatu penjadwalan produksi yang paling tepat dalam hal optimasi penggunaan sumber (resource) yang dimiliki sesuai dengan waktu pengiriman yang telah ditetapkan.
5. Mengumpulkan data.  
Mengumpulkan data pendukung yang diperlukan yaitu; data pesanan konsumen, data kapabilitas mesin dan workstation, waktu baku produksi dan waktu penyetalan saat penggantian job.
6. Membuat model algoritma untuk permasalahan yang diteliti. Pembuatan model ini berdasarkan jurnal internasional – *A genetic algorithm for flow shop scheduling problems* yang disusun O Etiler<sup>1</sup> (2004)
7. Mengolah data dengan menggunakan perangkat lunak MATLAB 7.0.1.

---

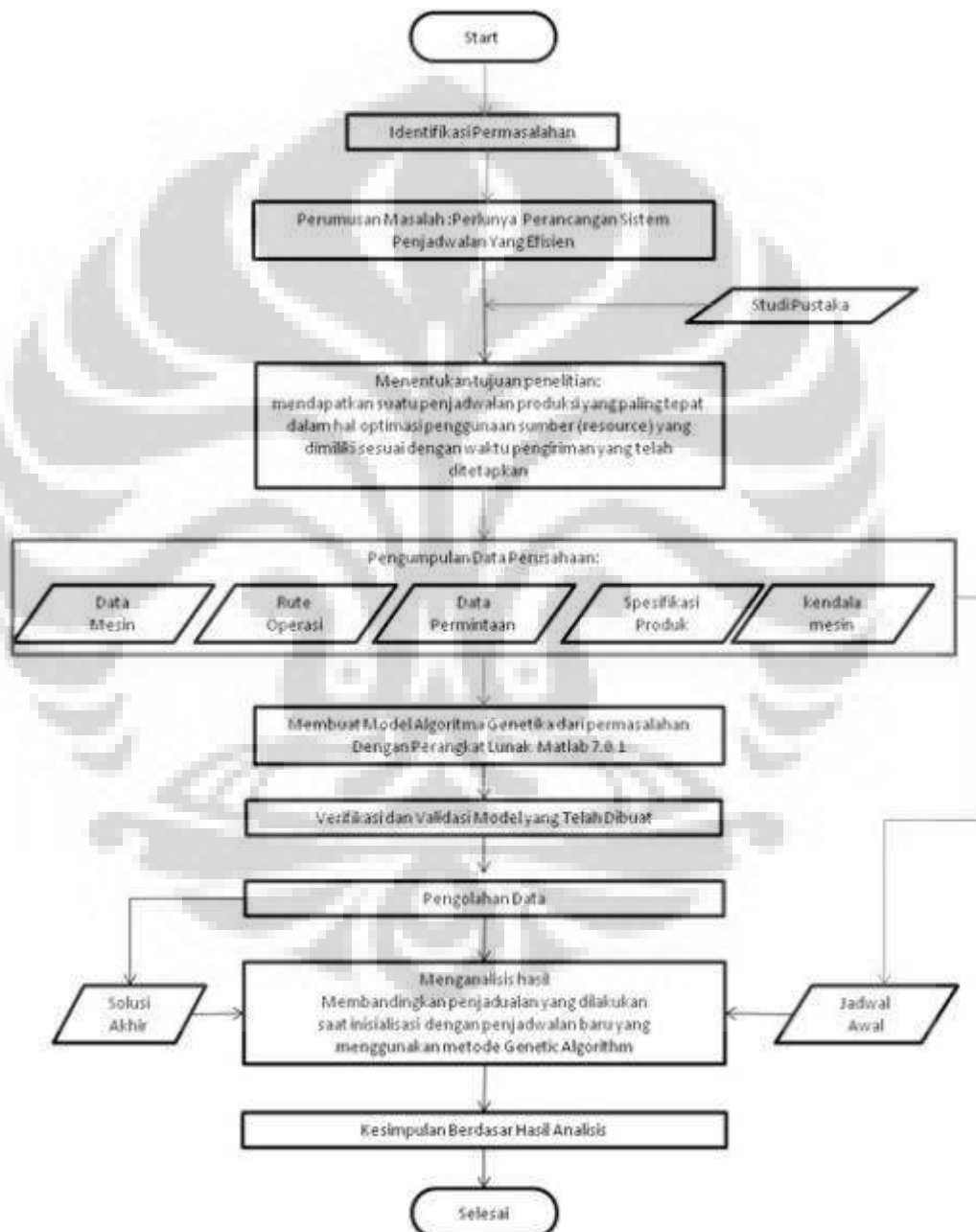
<sup>1</sup> O. Etiler, B. Toklu, M. Atak and J. Wilson, “ *A Genetic Algorithm For Flow Shop Scheduling Problems*”, Journal of The Operational Research Society, Vol 55, No 8,2004, hal 831

8. Melakukan analisis terhadap hasil penelitian.

Membandingkan penjadwalan yang dilakukan saat inialisasi solusi dengan penjadwalan baru yang menggunakan metode Genetic Algorithm.

9. Membuat kesimpulan.

Secara sederhana, langkah-langkah penelitian di atas tampak seperti gambar berikut ini



Gambar 1.2. Diagram alir metodologi penelitian

### **1.8. Sistematika Penulisan**

Penulisan penelitian ini dibagi menjadi lima, yaitu;

Bab 1, berisi penjelasan mengenai latar belakang masalah, diagram keterkaitan masalah untuk memahami interaksi masalah dan perumusan masalah. Pada bab ini juga dijelaskan tentang tujuan penelitian yang ingin dicapai, ruang lingkup penelitian yaitu batasan masalah dalam penelitian ini dan metodologi penelitian yang dilakukan.

Bab 2 adalah bagian yang menjelaskan teori-teori yang digunakan dalam penelitian ini, yaitu teori-teori yang berhubungan dengan sistem aliran produksi, scheduling, algoritma dan metode-metode penjadwalan yang berhubungan dengan multiproduct dan multistage.

Bab 3 menjelaskan data-data yang diambil untuk keperluan penelitian. Datadata tersebut merupakan data sekunder yang paling aktual dan akurat untuk menjamin bahwa data yang diambil merupakan representasi kondisi sebenarnya.

Bab 4 merupakan bagian pengolahan data dimana data-data yang diambil akan diolah dan dianalisis. Pada bagian ini terdapat hasil perhitungan suatu model matematis dan aplikasi pengembangan program komputer untuk penjadwalan produksi sebagai hasil akhir.

Bab 5 berisi kesimpulan tentang hasil penelitian yang telah dilakukan.

## **BAB 2**

### **DASAR TEORI**

Penjadwalan merupakan salah satu bentuk dari pengambilan keputusan yang memegang peranan penting baik di industri manufaktur maupun industri jasa. Dalam kondisi saat ini dimana kompetisi sudah demikian ketatnya, penjadwalan yang efektif menjadi suatu keharusan untuk bisa bertahan di pasaran. Penjadwalan atau scheduling secara mendalam mulai diperhatikan pada awal abad 20. Henry Laurence Gantt merupakan salah seorang perintis yang memperkenalkan sistem penjadwalan. Penjadwalan mulai memasuki bidang riset operasi pada tahun 1950-an. Dynamic programming dan integer programming mulai digunakan untuk memecahkan masalah penjadwalan pada tahun 1960-an. Tahun 1970-an complexity theory diperkenalkan oleh Richard Karp, teori ini menggambarkan relasi antar permasalahan pada penjadwalan. Pada tahun 1980-an, banyak akademisi maupun industri menggunakan stochastic dalam melakukan penjadwalan. Seiring dengan semakin berkembangnya ilmu dan teknologi komputer, penjadwalan dengan menggunakan program computer semakin banyak digunakan. Masalah penjadwalan telah menarik minat para programmer, ahli riset operasi dan industrial engineer terus melakukan studi mendalam tentang masalah penjadwalan yang semakin kompleks.

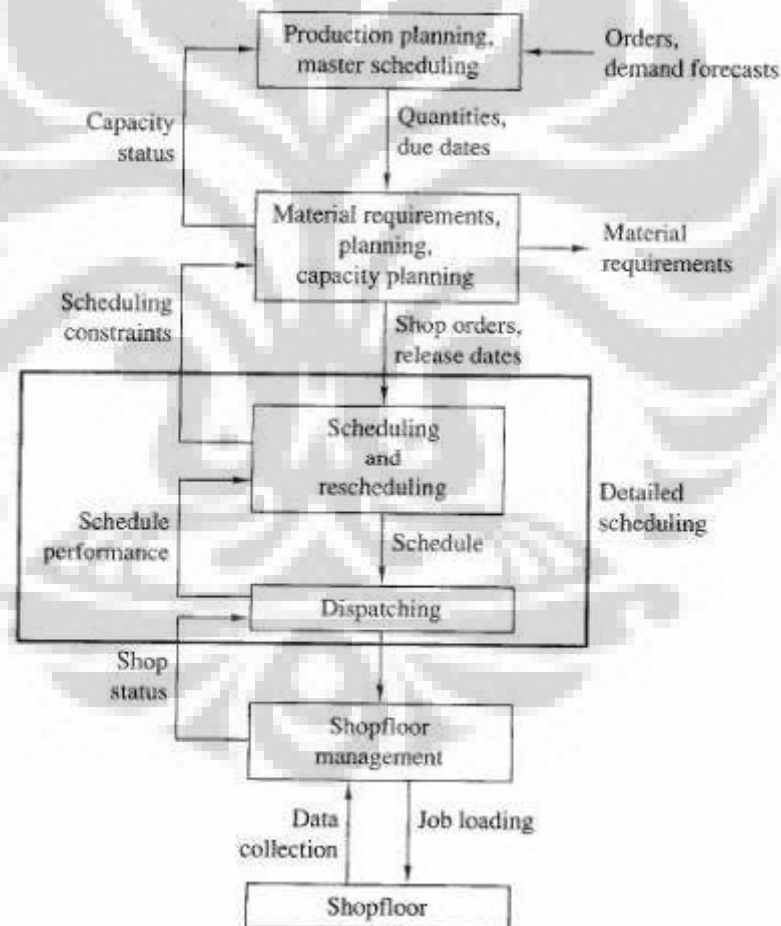
#### **2.1. Penjadwalan**

Perbedaan utama antara perencanaan dengan penjadwalan adalah pada kerangka waktu (time frame). Perencanaan merupakan gambaran tentang kondisi yang dilakukan pada jangka waktu yang relatif panjang (bulan, semester, tahun). Penjadwalan adalah penugasan yang lebih terperinci dalam melakukan suatu aktifitas dalam rentang waktu yang pendek (jam, hari, minggu). Lebih jauh, Baker (1993)<sup>2</sup> mengatakan bahwa dalam penjadwalan perlu disebutkan jenis maupun jumlah sumber daya yang dipergunakan sehingga dapat diperkirakan kapan satu pekerjaan akan selesai dikerjakan.

---

<sup>2</sup>Baker, Kenneth R. (2001). *Elements of Sequencing and Scheduling*. Hanover: Tock School of Business, Dartmouth College

Di dalam industri manufaktur, pesanan akan diterima dan diteruskan kepada bagian-bagian yang terkait dalam sebuah organisasi. Pesanan ini akan menginformasikan jenis barang yang dipesan, jumlahnya dan waktu penyelesaian yang diharapkan. Sebuah penjadwalan yang terperinci tentang mesin yang digunakan, tahapan proses yang dilalui, jumlah barang yang diproduksi dan waktu berproduksi, akan membantu proses pemenuhan pemesanan tersebut. Sebuah penjadwalan akan menginformasikan kebutuhan sumber-sumber lain kepada bagian terkait (misalnya kebutuhan material). Sebaliknya, informasi tentang perubahan jadwal produksi yang dilakukan juga harus diinformasikan kepada bagian lain. Sistem informasi ini digambarkan sebagai berikut.



Gambar 2.1. Sistem informasi manufaktur

### 2.1.1. Faktor Utama Penjadwalan

Faktor utama dalam penjadwalan adalah job, mesin dan fungsi objektif.

Notasi umum yang sering digunakan terhadap faktor job adalah;

- 1) *Processing time* ( $p_{ij}$ ) yaitu waktu pemrosesan job  $j$  pada mesin  $i$ .
- 2) *Release date* ( $r_j$ ) atau ready date yaitu waktu ketika job memasuki system atau waktu dimulainya pemrosesan job  $j$ .
- 3) *Due date* ( $d_j$ ) yaitu tenggat waktu penyelesaian pesanan yang sudah disepakati dengan pemesan.
- 4) *Weight* ( $w_j$ ) yaitu faktor prioritas terhadap sebuah job dibandingkan dengan job yang lain.

Berikutnya adalah notasi terhadap kondisi mesin yang digunakan.

- 5) *Single machine* ( $1$ ). Kondisi ini adalah kondisi yang paling sederhana hanya terdapat satu tahapan proses.
- 6) *Identical machines in parallel* ( $P_m$ ). Terdapat  $m$  mesin yang sejenis yang tersusun secara paralel dengan job yang diproses hanya dengan satu tahapan proses.
- 7) *Machines in parallel with different speed* ( $Q_m$ ). Terdapat beberapa mesin sejenis yang memiliki kecepatan berbeda dalam memproses job yang sama.
- 8) *Flow shop* ( $F_m$ ). Terdapat  $m$  mesin yang disusun secara seri. Setiap job harus melalui proses dari mesin  $m$  mulai dari awal.
- 9) *Flexible flow shop* ( $FFs$ ). Sama seperti pada flow shop, hanya pada Flexible flow shop mesin tersusun secara paralel pada setiap tahapan.
- 10) *Open shop* ( $O_m$ ). Terdapat  $m$  mesin yang digunakan berulang-ulang oleh sebuah job. Job memiliki kebebasan dalam menentukan rutanya sendiri.
- 11) *Job shop* ( $J_m$ ). Job memiliki rute tersendiri dalam memasuki  $m$  mesin yang ada, adakalanya sebuah job akan memasuki mesin  $m$  lebih dari satu kali.

Faktor terakhir yang merupakan faktor utama penjadwalan adalah fungsi objektif.

- 1) *Makespan* ( $C_{max}$ ). Penyelesaian job terakhir dan meninggalkan sistem.
- 2) *Makespan* yang minimal menunjukkan utilitas yang besar dari mesin yang digunakan.



- 2) *Maximum Lateness (Lmax)*. Kondisi terburuk dari simpangan terhadap tenggat waktu yang ditetapkan.
- 3) *Total weighted completion time ( $\sum w_j C_j$ )*. Penjumlahan dari bobot penyelesaian job yang mengindikasikan inventori atau job yang ditahan

### 2.1.2. Penjadwalan *Flexible Flow Shop*

Masalah penjadwalan flow shop dapat diselesaikan dengan menggunakan metode heuristik. *Dispatching rule* atau sering diistilahkan *scheduling in advance* merupakan teknik yang heuristik yang sering digunakan dalam penjadwalan. Beberapa metode *dispatching rules* adalah;

- 1) *First Come First Serve (FCFS)*. Menurut aturan ini, urutan penjadwalan dilakukan berdasarkan waktu kedatangan job atau pesanan pelanggan. Jadi, job yang pertama kali datang, akan dikerjakan terlebih dahulu dan begitu seterusnya untuk job-job berikutnya.
- 2) *Earliest Due Date First (EDD)* Menurut aturan ini, urutan penjadwalan dilakukan berdasarkan pada due date setiap job. Aturan ini mengabaikan waktu kedatangan dan total waktu proses setiap job. Artinya job yang memiliki due date yang paling awal diantara job-job lainnya dipilih sebagai job yang memiliki prioritas paling tinggi untuk diproses pada sebuah mesin. Aturan ini cenderung digunakan untuk meminimalkan maximum lateness pada job-job yang ada dalam antrian.
- 3) *Minimum Slack First (MS)*. Menurut aturan ini, merupakan job diurutkan berdasarkan waktu slack yang paling kecil. Pada saat sebuah mesin selesai memproses suatu job, maka kemudian dihitung waktu slack yang tersisa ( $d_i - p_i - t$ , 0) dari tiap-tiap job yang ada dalam antrian, dimana  $t$  adalah waktu sekarang. Job yang mempunyai waktu slack yang paling kecil kemudian dipilih sebagai job yang memiliki prioritas paling tinggi untuk diproses selanjutnya. Aturan ini digunakan untuk meminimalkan fungsi tujuan yang berkaitan dengan due date, yaitu *lateness* dan *tardiness*.

4) *Shortest Processing Time First (SPT)*. Menurut aturan ini, job diurutkan berdasarkan pada lamanya waktu proses tiap job. Jadi job yang mempunyai waktu proses paling singkat akan diproses terlebih dahulu dan kemudian dilanjutkan job-job lainnya sampai pada job yang paling lama waktu prosesnya. Aturan ini berguna untuk penyeimbangan beban kerja antar mesin yang disusun secara paralel. Disamping itu, terdapat metode heuristik yang lebih baru atau sering disitilahkan meta-heuristik yang digunakan dalam penjadwalan. Algoritma heuristik ini memecahkan masalah penjadwalan dengan melakukan perbaikan mulai dengan satu atau lebih solusi awal. Solusi awal ini bisa dihasilkan secara acak, bisa pula dihasilkan berdasarkan heuristik tertentu. Tiga algoritma metaheuristik yang sering digunakan dalam memecahkan masalah penjadwalan, yaitu2

#### 1. *Simulated Annealing*

Ide dasar Simulated Annealing terbentuk dari pemrosesan logam. Annealing (memanaskan kemudian mendinginkan) dalam pemrosesan logam ini adalah suatu proses bagaimana membuat bentuk cair berangsur-angsur menjadi bentuk yang lebih padat seiring dengan penurunan temperatur. Simulated Annealing biasanya digunakan untuk penyelesaian masalah yang mana perubahan keadaan dari suatu kondisi ke kondisi yang lainnya membutuhkan ruang yang sangat luas.

#### 2. *Tabu Search*

Tabu search merupakan metode optimasi yang menggunakan short-term memory untuk menjaga agar proses pencarian tidak terjebak pada nilai optima local. Metode ini menggunakan Tabu List untuk menyimpan sekumpulan solusi yang baru saja dievaluasi. Selama proses optimasi, pada setiap iterasi, solusi yang akan dievaluasi akan dicocokkan terlebih dahulu dengan isi Tabu list untuk melihat apakah solusi tersebut sudah ada pada Tabu List. Apabila sudah ada, maka solusi tersebut tidak akan dievaluasi lagi. Keadaan ini terus berulang sampai tidak ditemukan lagi solusi yang tidak terdapat dalam tabu list. Pada metode tabu search, solusi baru dipilih jika solusi tersebut yang merupakan anggota bagian himpunan solusi tetangga merupakan solusi dengan fungsi tujuan paling optimal jika dibandingkan dengan solusi-solusi lainnya dalam himpunan solusi tetangga

tersebut. Tetangga (neighbour) dari suatu solusi adalah solusisolusi lain yang dapat diperoleh dari solusi tersebut dengan cara memodifikasinya berdasarkan aturan-aturan tertentu yang dikenal dengan nama *neighbourhood functions*.

### 3. Algoritma Genetika

Algoritma genetika dimodelkan berdasar proses alami, yaitu model seleksi alam oleh Darwin, sedemikian hingga kualitas individu akan sangat kompatibel dengan lingkungannya (dalam hal ini kendala permasalahan) (Holland, 1975, Goldberg, 1989, dalam Jain dan Meeran, 1998b). Algoritma genetika memberikan suatu alternatif untuk proses penentuan nilai parameter dengan meniru cara reproduksi genetika. Teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang disebut dengan populasi. Setiap individu adalah satu buah solusi unik dan populasi adalah satu himpunan solusi pada setiap tahapan iterasi. Algoritma genetika bekerja untuk mencari struktur individu berkualitas tinggi yang terdapat dalam populasi.

#### 2.2. Genetic Algorithm

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

Ada empat kondisi yang sangat mempengaruhi proses evaluasi, yaitu:

1. Kemampuan organisme untuk melakukan reproduksi
2. Keberadaan populasi organisme yang bisa melakukan reproduksi.
3. Keberadaan organisme dalam suatu populasi.
4. Perbedaan kemampuan untuk dapat bertahan hidup.

Individu yang lebih kuat akan memiliki tingkat survival dan reproduksi yang lebih tinggi jika dibandingkan dengan individu yang lemah. Algoritma genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). Ia mengatakan bahwa setiap masalah yang berbentuk adaptasi dapat diformulasikan dalam terminologi genetika.

### 2.2.1. Struktur Umum *Genetic Algorithm*

Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut kromosom. Kromosom ini merupakan suatu solusi yang terdiri dari sejumlah gen.. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut fungsi fitness. Nilai fitness dari satu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (offspring), terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (parent) dengan menggunakan operator penyilangan (crossover). Selain operasi penyilangan, kromosom dapat juga dimodifikasi dengan menggunakan operator genetika yang lain yaitu mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai fitness dari kromosom induk dan nilai fitness dari kromosom anak, serta menolak kromosom yang lainnya (kromosom dengan nilai fitness yang kecil) sehingga ukuran populasi konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik yang akan menjadi solusi penyelesaian. Gambar 2.2 merupakan prosedur algoritma genetika secara umum<sup>3</sup>.

### 2.2.2 . Komponen-komponen utama dalam *Genetic Algorithm*

#### 2.2.2.1 Teknik penyandian

Teknik penyandian di sini meliputi penyandian gen dan kromosom. Gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Terdapat tiga teknik dalam penyandian yang utama, yaitu<sup>4</sup>:

1) Real-number encoding. Nilai gen berada dalam interval  $[0,R]$ , di mana  $R$  adalah bilangan real positif dan bernilai 1.

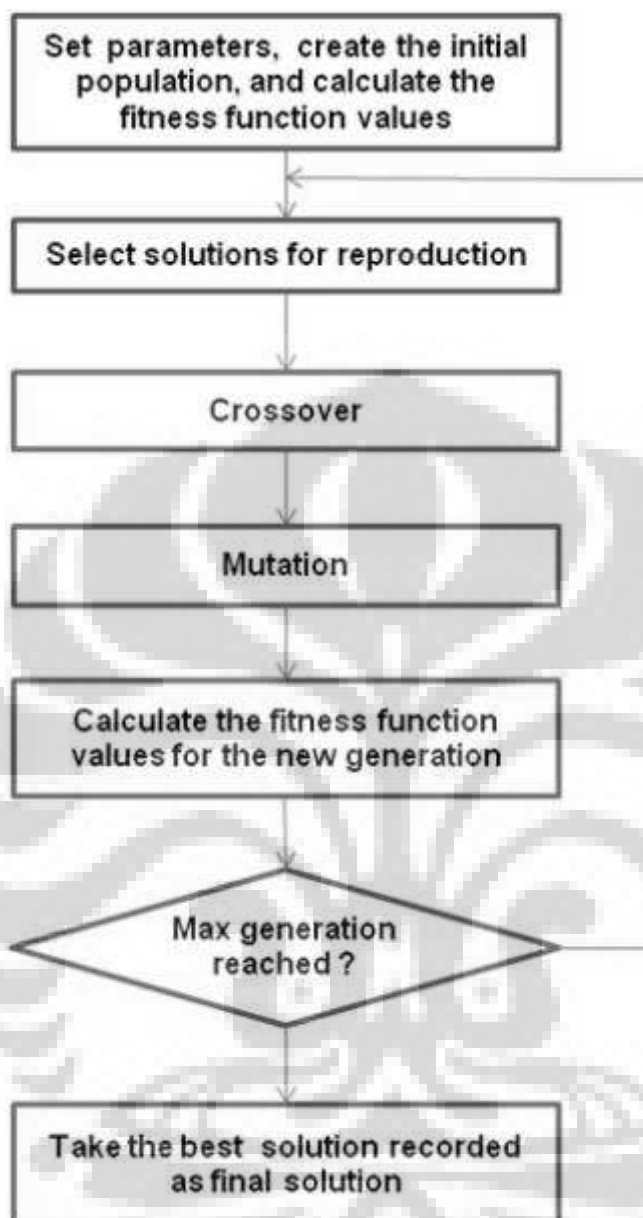
Misalnya: (0.256 0.567 0.654 0.156 0.895)

2) Discrete decimal encoding. Setiap gen bisa bernilai salah satu dari bilangan bulat dalam interval  $[0,9]$ . Misalnya: (2 3 4 5 6 8 9 7 1 0)

3) Binary encoding. Setiap gen hanya bernilai 0 dan 1. Misalnya : (1 1 1 0 1 0)

<sup>3</sup> Lixing, T, Jiyin L,(2002), Modified genetic Algorithm for the flow shop sequencing problem to minimize mean flow time,Journal of intelligent manufacturing, Vol. 13 no 1 pg. 61

<sup>4</sup> Suyanto, Algoritma Genetika Dalam Matlab, Andi Yogyakarta, 2005, hal. 6.



Gambar 2.2. Prosedur algoritma genetika secara umum

#### 2.2.2.2 Penentuan parameter

Parameter dalam algoritma genetika terdiri dari ukuran populasi, peluang kawin silang, peluang mutasi, dan maksimum generasi. Untuk ukuran populasi dan maksimum ditentukan tergantung seberapa besar permasalahan yang ingin

diselesaikan. Sedangkan peluang probabilitas kawin silang dan mutasi ditentukan setelah ukuran populasi ditetapkan.

#### 2.2.2.3 Prosedur inialisasi

Ukuran populasi tergantung pada masalah yang akan diimplementasikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, dilakukan inialisasi kromosom dalam populasi tersebut. Inialisasi kromosom dilakukan secara acak, namun harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada<sup>5</sup>.

#### 2.2.2.4 Fungsi evaluasi

Fungsi evaluasi memiliki peranan penting sebab merupakan alat ukur untuk menentukan kualitas kromosom dalam populasi. Fungsi evaluasi merupakan fungsi objektif dari masalah yang akan dioptimasi. Fungsi evaluasi terdiri dari fungsi meminimalkan dan memaksimalkan. Dua hal yang harus dilakukan dalam melakukan evaluasi kromosom yaitu evaluasi fungsi objektif dan konversi fungsi objektif ke dalam fungsi fitness yang tidak negatif.

#### 2.2.2.5 Seleksi

Seleksi bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi kromosom yang paling fit. Seleksi akan menentukan individu-individu mana saja yang akan dipilih untuk dilakukan kawin silang (crossover) dan bagaimana anak terbentuk dari individu yang dipilih tersebut. Nilai fitness yang nantinya digunakan dalam menyeleksi setiap kromosom. Metode seleksi yang paling sering dipakai ialah metode roulette wheel. 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. Kemudian bilangan acak dibangkitkan untuk menentukan kromosom mana yang akan dipilih. Kromosom dengan nilai fitness lebih besar akan lebih banyak terpilih karena memiliki potongan lingkaran yang besar dalam roda roulette.

---

<sup>5</sup> Sri Kusumadewi, & Hari Purnomo, "Penyelesaian Masalah Optimasi Menggunakan Teknik-teknik Heuristik, Graha Ilmu, Yogyakarta, 2005, hal. 233

### 2.2.2.6 Operator genetika

#### a. Kawin silang (Crossover)

Kawin silang merupakan operator paling utama dalam algoritma genetika. Kawin silang bertujuan untuk menambah keanekaragaman gen dalam populasi dengan penyilangan antar gen yang diperoleh dari peroduksi sebelumnya. Dua kromosom induk akan dikawinsilangkan dan menghasilkan 2 kromosom anak yang baru (seperti pada gambar 2.4).



Gambar 2.3. Contoh Kawin Silang

#### b. Mutasi

Mutasi adalah proses pertukaran sejumlah gen dalam satu individu dengan menukar nilai karakter pada gen-gen tersebut dengan kebalikannya. Contoh mutasi dapat dilihat pada gambar 2.5, yaitu gen 3 akan digantikan dengan gen 4, dan sebaliknya. Mutasi dilakukan untuk menjaga agar tidak terciptanya konvergensi prematur (solusi yang tidak optimal/lokal optima).

	TM1			TM2		
Sebelum Mutasi	2	3	5	1	4	6
Sesudah Mutasi	2	4	5	1	3	6

Gambar 2.4. Contoh Mutasi

### 2.2.2.7 Penggantian populasi

Dalam algoritma genetika dikenal skema penggantian populasi yang disebut *generational replacement*, yang berarti semua individu (misalkan N individu dalam satu populasi) dari suatu generasi digantikan sekaligus oleh N individu baru hasil kawin silang dan mutasi. Pada dasarnya, skema penggantian ini tidak realistis dari sudut pandang biologi. Di dunia nyata individu-individu dari generasi berbeda bisa berada dalam waktu yang bersamaan. Fakta lainnya adalah individu-individu muncul dan hilang secara konstan, tidak pada generasi tertentu. Secara umum, skema penggantian populasi dapat dirumuskan berdasarkan suatu ukuran yang disebut *generational gap*, G. Ukuran ini menunjukkan presentasi populasi yang digantikan dalam setiap generasi. Pada skema *generational*

replacement,  $G = 1$ . Skema penggantian yang paling ekstrim adalah hanya mengganti satu individu dalam setiap generasi, yaitu  $G = 1/N$ , di mana  $N$  adalah jumlah individu dalam populasi. Skema penggantian ini disebut sebagai steady-state reproduction. Pada skema tersebut,  $G$  biasanya sama dengan  $1/N$  atau  $2/N$ .

Dalam setiap generasi, sejumlah  $NG$  individu harus dihapus untuk menjaga ukuran populasi agar selalu tetap  $N$ . Terdapat beberapa prosedur penghapusan individu, yaitu penghapusan individu yang bernilai fitness paling rendah atau penghapusan individu yang paling tua. Penghapusan bisa berlaku hanya pada orang tua atau bisa juga berlaku pada semua individu dalam populasi<sup>6</sup>

---

<sup>6</sup> Suyanto, Op. cit. hal.15.



## BAB 3

### PENGUMPULAN DATA

#### 3.1 Profil Perusahaan

SPTI mulai berdiri pada 7 Juli 1997 terletak dikawasan industri MM2100 sekitar 30 km sebelah timur kota Jakarta merupakan salah satu anak perusahaan kelompok Mitsubishi Material. Sebagai perusahaan pertama di Indonesia yang memproduksi silicon polished wafer 5” dan 6”, SPTI kini telah mampu mengembangkan kapasitas produksinya hingga 280.000 lembar per bulan dimana orientasi pasar dari produk ini adalah 100% eksport baik ke Jepang, Amerika, Singapura maupun beberapa negara di Eropa.

Dengan semakin meningkatnya teknologi pembuatan silicon wafer maka beberapa perusahaan dari group ini telah merubah bisnis dari silicon wafer 5” dan 6” menjadi dimensi yan lebih besar sehingga kedepannya direncanakan SPTI ini akan menjadi pusat pembuatan *silicon polished wafer 5” dan 6”* di dunia

#### 3.2. Pengumpulan Data Penelitian

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari data perusahaan di SPTI. Data yang digunakan dalam penelitian optimasi penjadwalan *flexible flow shop* berupa:

1. Data Kapasitas waktu produksi di SPTI
2. Data rute proses operasi yang harus dilalui oleh tiap *item* yang akan diproduksi dan data jumlah mesin yan digunakan untuk produksi
3. Data waktu proses tiap rute yang dibutuhkan untuk mengerjakan tiap *item*.
4. Data penanganan pesanan dan kegiatan perencanaan di SPTI
5. Data pesanan, yaitu berupa jenis-jenis *silicon polished wafer* yang dipesan dan jumlah pesanan pada periode Agustus 2008.

##### 3.2.1. Data Kapasitas Waktu Produksi

Pembagian jam kerja di SPTI ditentukan berdasar kebutuhan masing-masing bagian akan tetapi secara garis besar ada 2 macam jam kerja di SPTI yaitu day time dan shift. Daytime biasanya hanya untuk bagian non produksi. Sedangkan untuk

bagian produksi biasanya dibagi dalam 4 group 3 shift namun adapula yang dibagi dalam 3 group 2 shift dan 3 group 3 shift. Pada bagian produksi yang menggunakan aturan 4 group 3 shift maka proses produksi akan berjalan terus menerus selama 24 jam dan hanya berhenti pada jadwal *shutdown* yang direncanakan. Dalam waktu yang sama selalu ada 1 group yang libur. Selain itu beberapa stasiun kerja biasanya sudah menggunakan mesin semi otomatis sehingga akan tetap berjalan ketika operator mesin sedang istirahat. Dalam kondisi produksi yang tinggi beberapa mesin yang masih manual juga masih tetap dioperasikan pada jam istirahat dengan cara pengaturan jam istirahat secara bergantian. Secara umum jadwal kerja di SPTI dapat disajikan dalam table 3.1

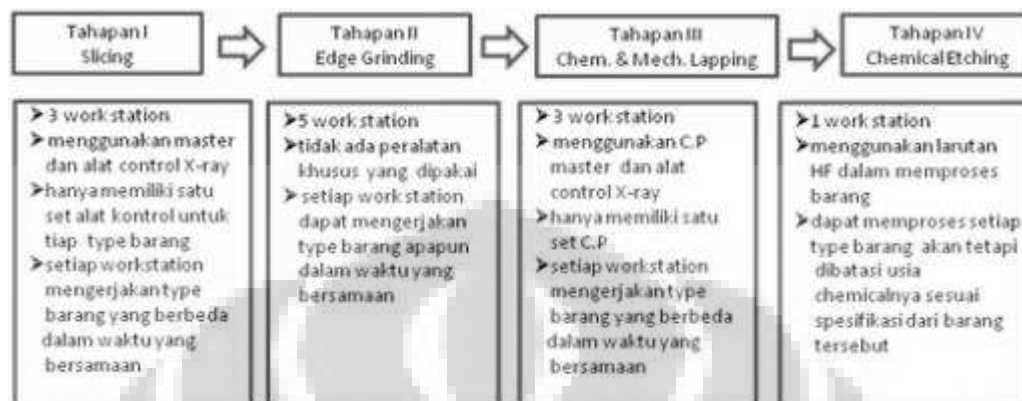
3.1. Tabel jam kerja di SPTI

Jenis Jam Kerja	Waktu Kerja dan Jumlah Jam Kerja		
Daytime	Pk. 07.00-16.00 8 jam		
4 group 2 shift	Pk. 07.00-16.00 8 jam	Pk.15.40-12.00 7 jam	
	Shift	Shift 2	
4 group 3 shift	Pk. 07.00-16.00 8 jam	Pk.15.40-12.00 7 jam	Pk. 23.40-07.20 6.8 jam
	Shift	Shift 2	Shift 3

### 3.2.2. Rute Proses Produksi *Silicon Polished Wafer*

Proses produksi untuk *Silicon Polished Wafer* memiliki beberapa tahapan. Tahapan pertama adalah proses *slicing*. Proses ini merupakan kegiatan memotong batangan silindris dari kristal silika (*silicon ingot x-tal*) menjadi lembaran tipis sekitar 700 mikron. Proses selanjutnya adalah proses *edge grinding*, dimana barang yang sudah Rajang menjadi lembaran bundar dengan diameter 125 mm sampai dengan 150 mm digerinda sudut tepinya hingga menghasilkan sisi yang halus untuk menghindari keretakan saat diproses pada tahap berikutnya. Hasil dari pekerjaan ini diteruskan ke proses berikutnya yaitu proses *lapping*, yaitu proses penghalusan permukaan dan juga pembentukan ketebalan sesuai dengan

spesifikasi yang diinginkan konsumen. Setelah tahap ketiga dilalui maka tahap selanjutnya adalah proses penghalusan permukaan dengan zat kimia atau *chemical etching*. Secara umum proses fabrikasi silicon polished wafer di SPTI adalah sebagai berikut :



Gambar 3.1. Alur proses produksi *silicon polished wafer*

### 3.2.3. Waktu baku dan spesifikasi produk

Lamanya proses pengerjaan *silicon polished wafer* sangat dipengaruhi oleh jumlah atau panjang dari tiap unit job atau lot. Akan tetapi untuk masing-masing type biasanya mempunyai jumlah yang hampir sama dalam tiap lotnya. Hal ini sebabkan karena tiap type telah memiliki spesifikasi tertentu sehingga dalam proses fabrikasi material *x-tal ingot* hanya dihasilkan jumlah yang kurang lebih sama untuk tiap lot-nya. Sedangkan waktu baku untuk masing-masing disajikan dalam table sebagai berikut :

Tabel 3.2. Waktu baku untuk semua type (dalam menit)

Type	Waktu Baku			
	Tahapan I	Tahapan II	Tahapan III	Tahapan IV
B296-010-D	355	552	300	79
C306-034-A	250	376	208	68
C905-015-C	441	689	376	86
C906-058-B	378	592	328	82

Tabel 3.2. Waktu baku untuk semua type (dalam menit) (sambungan)

Type	Waktu Baku			
	Tahapan I	Tahapan II	Tahapan III	Tahapan IV
C906-059-C	404	627	358	81
C906-101-A	236	356	223	66
C906-103-A	483	753	442	89
D105-001-B	135	196	143	57
D806-003-D	377	588	357	81
D866-024-E	382	595	367	79
D866-025-D	412	639	402	82
D866-026-B	390	606	394	79

Proses tahapan ke IV merupakan proses dengan cairan kimia. Proses ini sangat tergantung dari usia larutan kimia yang dipakai yang ditentukan oleh jumlah akumulasi dari job yang telah dikerjakan. Oleh karena itu selain waktu baku tiap tipe pesanan juga memiliki nilai standar dan spesifikasi etchant yang bisa digunakan untuk memproses dalam masing-masing tipe. Spesifikasi dan jumlah standart ini akan menentukan kualitas dari hasil proses pada tahapan ini. Jumlah standart dan spesifikasi tiap job dapat dilihat pada pada table 3.3

Tabel 3.3 Spesifikasi dan nilai per individu masing-masing job

Type	Nilai individu	Spesifikasi <i>etchant</i>
B296-010-D	289	1750
C306-034-A	199	600
C905-015-C	363	1750
C906-058-B	309	3000
C906-059-C	331	3000
C906-101-A	187	3000
C906-103-A	399	1750
D105-001-B	100	1750

Tabel 3.3 Spesifikasi dan nilai per individu masing-masing job  
(sambungan)

Type	Nilai individu	Spesifikasi <i>etchant</i>
D105-001-B	100	1750
D806-003-D	308	2250
D866-024-E	313	1750
D866-025-D	338	1750
D866-026-B	320	1750

#### 3.2.4. Kegiatan Perencanaan Produksi

Kegiatan perencanaan produksi dimulai dari proses penerimaan pesanan (*purchase order / PO*) oleh bagian Marketing. Bagian Marketing meneruskan informasi ini kepada bagian *Production Administration* (PA). Kemudian bagian PA menginformasikan ke bagian *Production and Planning Arrangement* (PPA), Produksi, *Quality Control*, *Engineering* dan *Purchasing*. Bagian PPA akan memasukkan data tersebut ke dalam system yang disebut MIPS dan SAP. Data di SAP akan dipergunakan sebagai acuan untuk pemesanan material sedangkan data di MIPS akan dipakai sebagai acuan jadwal produksi dan rencana pengiriman. Berdasarkan data ini bagian Produksi akan membuat perhitungan kapasitas produksi dan perencanaan produksi.

Apabila berdasarkan perhitungan kapasitas produksi, bagian Produksi tidak dapat memenuhi tenggat waktu yang diminta, maka bagian Produksi akan menginformasikannya kepada bagian PPA dan selanjutnya akan diteruskan ke bagian *marketing* untuk dinegosiasikan kepada pemesan. Bersamaan dengan informasi tersebut, bagian Produksi akan mengajukan tenggat waktu baru sesuai dengan kapasitas produksi yang tersedia.

Penjadwalan produksi yang selama ini dilakukan oleh bagian Produksi dilakukan dengan perhitungan yang sederhana. Tidak ada acuan yang baku untuk menetapkan type barang yang mana yang akan diproses terlebih dahulu.

### 3.2.5. Data Pemesanan

Untuk penelitian ini data pemesanan silicon polished wafer diambil dari data pemesanan untuk pengiriman bulan agustus 2008. Proses produksi terhadap pesanan ini dilakukan pada bulan akhir bulan Juli dan awal bulan Agustus 2008 dimana total hari kerja normal yang tersedia berjumlah 31 hari. Data pemesanan selengkapnya dapat dilihat pada table 3.4

Tabel 3.4 Jumlah pesanan *silicon polished wafer 5” dan 6”*

No	Nama Barang	Jumlah Pesanan
1	B296-010-D	81
2	C306-034-A	1
3	C905-015-C	108
4	C906-058-B	91
5	C906-059-C	98
6	C906-101-A	38
7	C906-103-A	4
8	D105-001-B	5
9	D806-003-D	6
10	D866-024-E	8
11	D866-025-D	91
12	D866-026-B	35

## BAB 4

### PENGOLAHAN DATA DAN ANALISIS

#### 4.1. Fungsi Tujuan

Seperti dijelaskan pada bab sebelumnya bahwa objek pada penelitian ini memiliki type pekerjaan *flexible flow shop* dimana barang yang diproses harus melalui beberapa tahapan (stage) dengan mesin-mesin yang disusun secara paralel. Fungsi tujuan dari penelitian ini adalah untuk mendapatkan minimum makespan ( $C_{max}$ ). Makespan adalah rentang waktu yang dibutuhkan untuk memproses pekerjaan yang dimulai pada waktu  $RT$  dan diselesaikan pada waktu  $FT$ . Untuk kondisi dimana *flexible flow shop* memiliki objektif meminimalkan makespan, Pinedo menotasikannya sebagai  $FFs|C_{max}$  (Pinedo, 1995)<sup>7</sup>

$$\begin{aligned} FFs|C_{max} = & (\text{Max}\{FT_{ijk4m}\} - \text{Min}\{RT_{ijk4m}\}) + \\ & (\text{Min}\{RT_{ijk4m}\} - \text{Min}\{RT_{ijk3m}\}) + \\ & (\text{Min}\{RT_{ijk3m}\} - \text{Min}\{RT_{ijk2m}\}) + \\ & (\text{Min}\{RT_{ijk2m}\} - \text{Min}\{RT_{ijk1m}\}) \end{aligned} \quad (4-1)$$

dimana  $\forall i \in I, \forall j \in J, \forall m \in M$

Notasi:

FFs : Flexible Flow shop dengan 4 stage

FT : Finish Time, waktu selesai pekerjaan

RT : Release Time, waktu mulai pekerjaan

$i$  : job

$j$  : unit  $j$  dari job  $i$

$k(n)$  : tahapan ke- $n$  (1,2,3,4)

$m$  : mesin/workstation ke- $m$  pada  $k(n)$

$I$  : jumlah keseluruhan job

$J$  : jumlah keseluruhan unit

$M$  : jumlah keseluruhan mesin/workstation

---

<sup>7</sup> Pinedo, Michael, *Scheduling Theory, Algorithm and Systems*, Prentice Hall, 1995

## 4.2. Inisialisasi Solusi

Dari data pemesanan yang ada, dilakukan inisialisasi solusi untuk mendapatkan penjadwalan awal yang akan dioptimasi pada langkah selanjutnya. Pada tahap inisialisasi solusi ini, jumlah pemesanan setiap type barang dibagi diambil secara sampling yang merepresentasikan jumlah barang yang akan diproses selama 5 hari kerja. Angka pada data tersebut dibulatkan ke angka terdekat.

Kondisi yang digunakan pada penjadwalan adalah sebagai berikut:

- 1) Setiap mesin/workstation hanya mengerjakan satu unit barang pada satu waktu dan tidak dapat diselingi oleh pekerjaan lain (*non preemptive*).
- 2) Tidak ada kendala dalam ketersediaan material.
- 3) Kondisi mesin dianggap stabil.
- 4) Waktu proses (*processing time*) sudah termasuk waktu set up.
- 5) Pada proses di tahapan 1 penggantian pisau antara tipe barang yang satu dengan tipe barang yang lain adalah tetap. Waktu untuk penggantian pisau adalah 60 menit pada tahapan 1. Kemudian penggantian carrier plate dan setting ulang pada tahap ke 3 adalah 45 menit.

Tahapan yang dilakukan dalam melakukan inisialisasi solusi adalah sebagai berikut:

- 1) Menetapkan metode penjadwalan.
- 2) Menyusun penjadwalan.
- 3) Menghitung makespan.

Tabel 4.1. Data jumlah pemesanan *silicon polished wafer* untuk inisialisasi solusi

No	Nama Barang	Jumlah Pesanan
1	B296-010-D	10
2	C306-034-A	1
3	C905-015-C	12
4	C906-058-B	7
5	C906-059-C	10
6	C906-101-A	7
7	C906-103-A	2
8	D105-001-B	5



Tabel 4.1. Data jumlah pemesanan *silicon polished wafer* untuk inisialisasi solusi  
(sambungan)

No	Nama Barang	Jumlah Pesanan
9	D806-003-D	6
10	D866-024-E	8
11	D866-025-D	11
12	D866-026-B	7

#### 4.2.1. Menetapkan Metode Penjadwalan

Penetapan metode penjadwalan dilakukan terhadap ke empat tahapan (stage) dari proses pembuatan *silicon polished wafer*.

##### 4.2.1.1. Metode Penjadwalan Tahapan 1

Dalam menetapkan metode yang akan digunakan untuk menyusun penjadwalan tahapan 1, perlu diperhatikan kondisi yang menyertainya. Berdasarkan data yang ada, kondisi tersebut adalah:

- 1) Tahapan 1 memiliki tiga buah *workstation* yang tersusun secara parallel dimana setiap *workstation* memiliki kemampuan yang sama dalam memproses semua unit pesanan.
- 2) Setiap type barang menggunakan pisau yang berbeda antara type yang satu dengan type yang lain.
- 3) Perubahan pengerjaan dari type yang satu ke type yang lain pada mesin yang sama akan menyebabkan bertambahnya total waktu pemrosesan sebagai akibat dari pergantian *slicing blade* dimana waktu yang dibutuhkan untuk pergantian *blade* ( $COT = \text{changeover time}$ ) adalah 60 menit yang berlaku tetap untuk semua tipe dan semua *workstation*.

Dari kondisi di atas, penetapan metode penjadwalan tahapan 1 adalah:

- 1) Penjadwalan dengan *workstation* secara paralel, seluruh unit pesanan harus dibagi ke seluruh mesin secara maksimal sehingga waktu pemrosesan menjadi singkat.
- 2) Setiap tipe memiliki karakter yang unik dalam melakukan prosesnya. yaitu penggunaan *slicing blade*, peralatan pengendali x-ray dan *wafer master* yang berbeda untuk masing-masing type. Pergantian type akan mengakibatkan bertambahnya waktu proses keseluruhan, sehingga pergantian type harus

dilakukan seminimal mungkin. Terhadap kondisi seperti ini, pengelompokan unit pesanan dapat dilakukan seperti diungkapkan oleh Kenneth R Baker (2001)<sup>8</sup>. Berdasarkan hal tersebut, maka unit pesanan dikelompokkan berdasarkan tipe barang, untuk selanjutnya kelompok unit pesanan ini disebut job.

- 3) Waktu penyelesaian sebuah pekerjaan adalah waktu yang dibutuhkan untuk menyelesaikan satu unit pesanan. Waktu penyelesaian sebuah job merupakan akumulasi dari waktu penyelesaian seluruh unit dari job tersebut.
- 4) Karena tahapan 1 merupakan awal proses dari tiga proses yang harus dijalani oleh setiap unit, maka barang yang telah dikerjakan harus segera dikerjakan pada tahapan selanjutnya. Metode yang dapat digunakan untuk hal tersebut adalah SPT (*shortest processing time*). SPT merupakan dispatching rule yang digunakan untuk mendahulukan pekerjaan dengan waktu proses terpendek. Dengan SPT, diharapkan idle time pada tahapan 2 akan minimal.

Tabel 4.2. Pengelompokan unit pesanan menjadi job

Nama Barang	Job (i)	Jumlah Pesanan
B296-010-D	(1)	10
C306-034-A	(2)	1
C905-015-C	(3)	12
C906-058-B	(4)	7
C906-059-C	(5)	10
C906-101-A	(6)	7
C906-103-A	(7)	2
D105-001-B	(8)	5
D806-003-D	(9)	6
D866-024-E	(10)	8
D866-025-D	(11)	11
D866-026-B	(12)	7

<sup>8</sup> Kenneth R. Baker., Elements of Sequencing and Scheduling, Tuck School of Business, Hanover, 2001.h.10.1

#### 4.2.1.2. Metode Penjadwalan Tahapan 2

Pada tahapan 2, kondisi yang menjadi dasar penetapan metode penjadwalan adalah sebagai berikut;

- 1) Tahapan ini memiliki 5 buah *workstation* dimana workstation ini memiliki kemampuan untuk memproses semua unit pesanan.
- 2) Tidak ada kondisi khusus pada tahapan ini, sehingga workstation memiliki keleluasaan untuk diberikan pekerjaan apapun.

Dari kondisi di atas, penetapan metode penjadwalan tahapan 2 adalah:

- 1) Pekerjaan yang dilakukan oleh workstation adalah unit base, dimana setiap workstation dapat mengerjakan unit pesanan dari job yang manapun. Tidak ada penambahan waktu terhadap pergantian job.
- 2) Tahapan 2 merupakan input untuk proses tahapan 3, maka barang yang telah dikerjakan harus segera dikirimkan pada tahapan selanjutnya. Untuk itu, perlu dipergunakan EFT (*earliest finish time*) sebagai *dispatching rule* yang digunakan agar barang yang telah terselesaikan dengan waktu tercepat bisa segera dikirimkan ke tahapan selanjutnya.

#### 4.2.1.3. Metode Penjadwalan Tahapan 3

Kondisi pada tahapan 3 sama seperti pada kondisi tahapan 1 dimana:

- 1) Tahapan 3 memiliki tiga buah workstation yang tersusun secara parallel dimana setiap workstation memiliki kemampuan yang sama dalam memproses semua unit pesanan.
- 2) Setiap type barang menggunakan carrier plate yang berbeda antara type yang satu dengan type yang lain. Selain itu pada tahapan ini setiap tipe juga mempunyai peralatan sensor x-ray untuk mencari sudut orientasi *x-tal* dan carrier plate Masing-masing type hanya memiliki satu buah alat.
- 3) Perubahan pengerjaan dari type yang satu ke type yang lain pada workstation yang sama akan menyebabkan bertambahnya total waktu pemrosesan sebagai akibat dari pergantian carrier plate dan penyetelan ulang mesin dimana waktu yang dibutuhkan untuk pergantian ( $COT = \text{changeover time}$ ) adalah 45 menit yang berlaku tetap untuk semua type dan semua workstation.

Dengan demikian, pada tahapan 3 penjadwalan akan disusun berdasarkan pengelompokkan unit pesanan (job). Namun demikian, terdapat perbedaan

antara keduanya, dimana pada tahapan 3 penggunaan EFT sebagai *dispatching rule* yang digunakan untuk menetapkan pemilihan mesin saat pergantian job.

#### 4.2.1.4. Metode Penjadwalan Tahapan 4

Kondisi pada tahapan 4 berbeda dengan semua tahapan yang ada sama seperti pada kondisi tahapan 1 dimana :

- 1) Tahapan 4 hanya memiliki satu buah workstation akan tetapi job yang bisa diproses tergantung dari nilai akumulasi dari cairan kimia yang telah digunakan. Sebagai contoh umur kimia yang baru dapat digunakan untuk memproses semua pesanan akan tetapi umur kimia yang telah dipakai memproses 1500 pcs hanya bisa dipakai untuk memproses job dengan spesifikasi tertentu saja.
- 3) Penggantian cairan kimia setelah pemakaian 3000 pcs akan menyebabkan bertambahnya waktu total waktu pemrosesan sebagai akibat dari pergantian cairan kimia dimana waktu yang dibutuhkan untuk pergantian kimia (COT=changeover time) adalah 30 menit yang berlaku tetap.

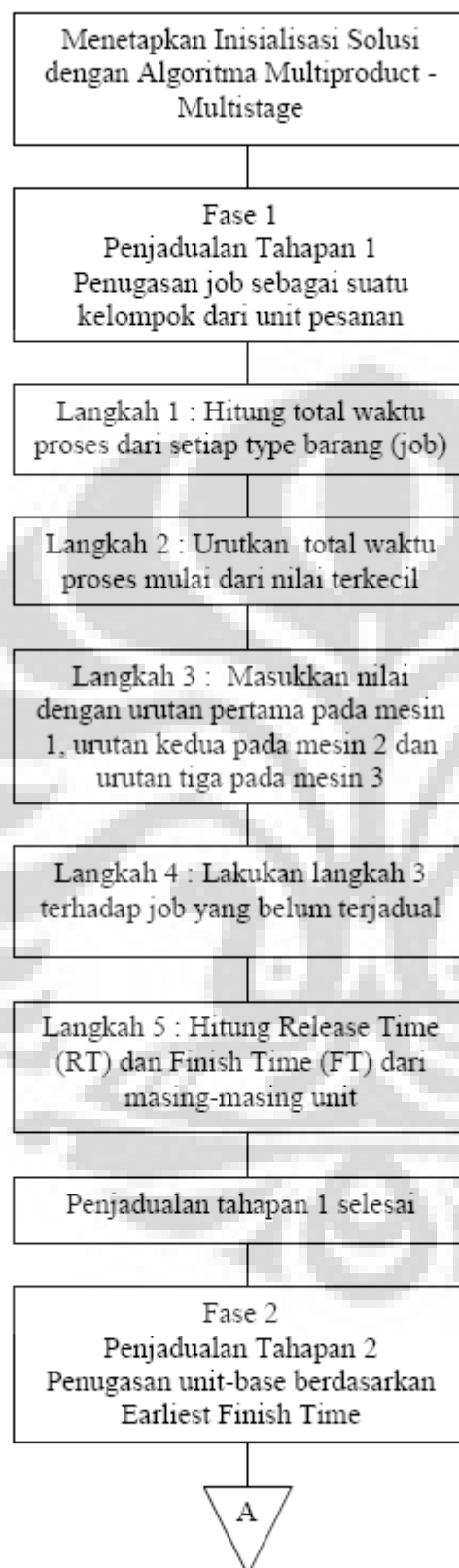
Dengan demikian, pada tahapan 4 penjadwalan akan disusun bukan berdasarkan pengelompokan unit pesanan (job) akan tetapi berdasar individu. Dari pemilihan metode di atas, maka metode penjadwalan pada setiap tahapan adalah seperti pada tabel di bawah ini

Tabel 4.3. Penetapan metode penjadwalan untuk setiap tahapan

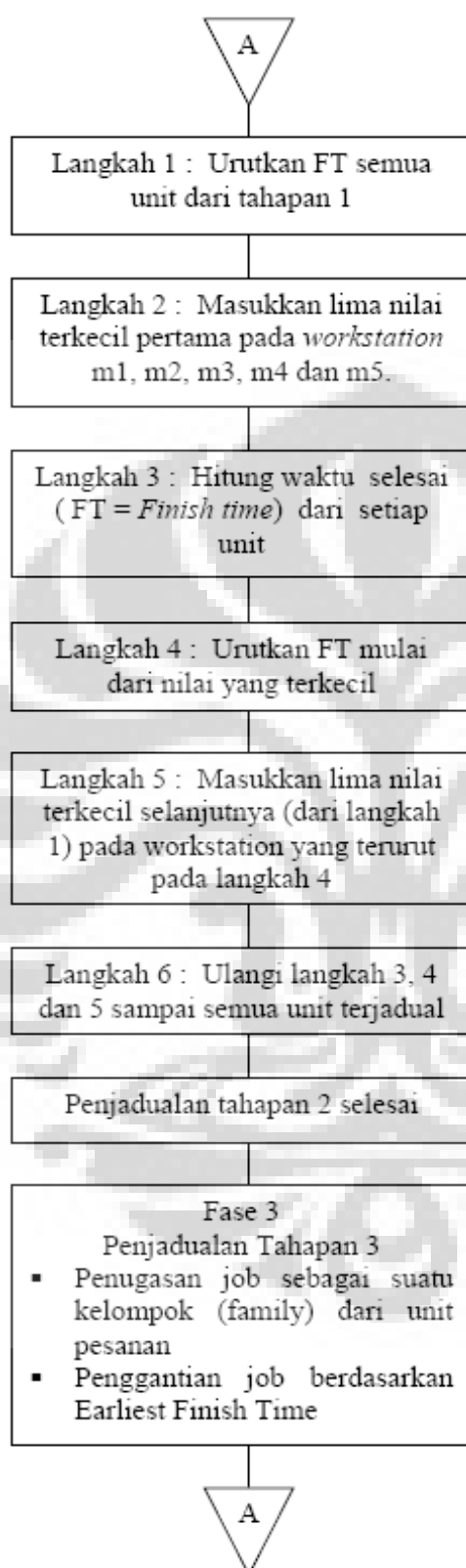
Tahapan Proses	Susunan Mesin	Dispatching Rule
Tahapan I	Paralel 3 mesin	SPT, kelompok (job)
Tahapan II	Paralel 5 mesin	EFT, <i>unit base</i>
Tahapan III	Paralel 3 mesin	EFT, kelompok (job)
Tahapan IV	Single 1 mesin	EFT, <i>unit base</i>

#### 4.2.2. Menyusun Penjadwalan dengan algoritma *multi stage multi product*

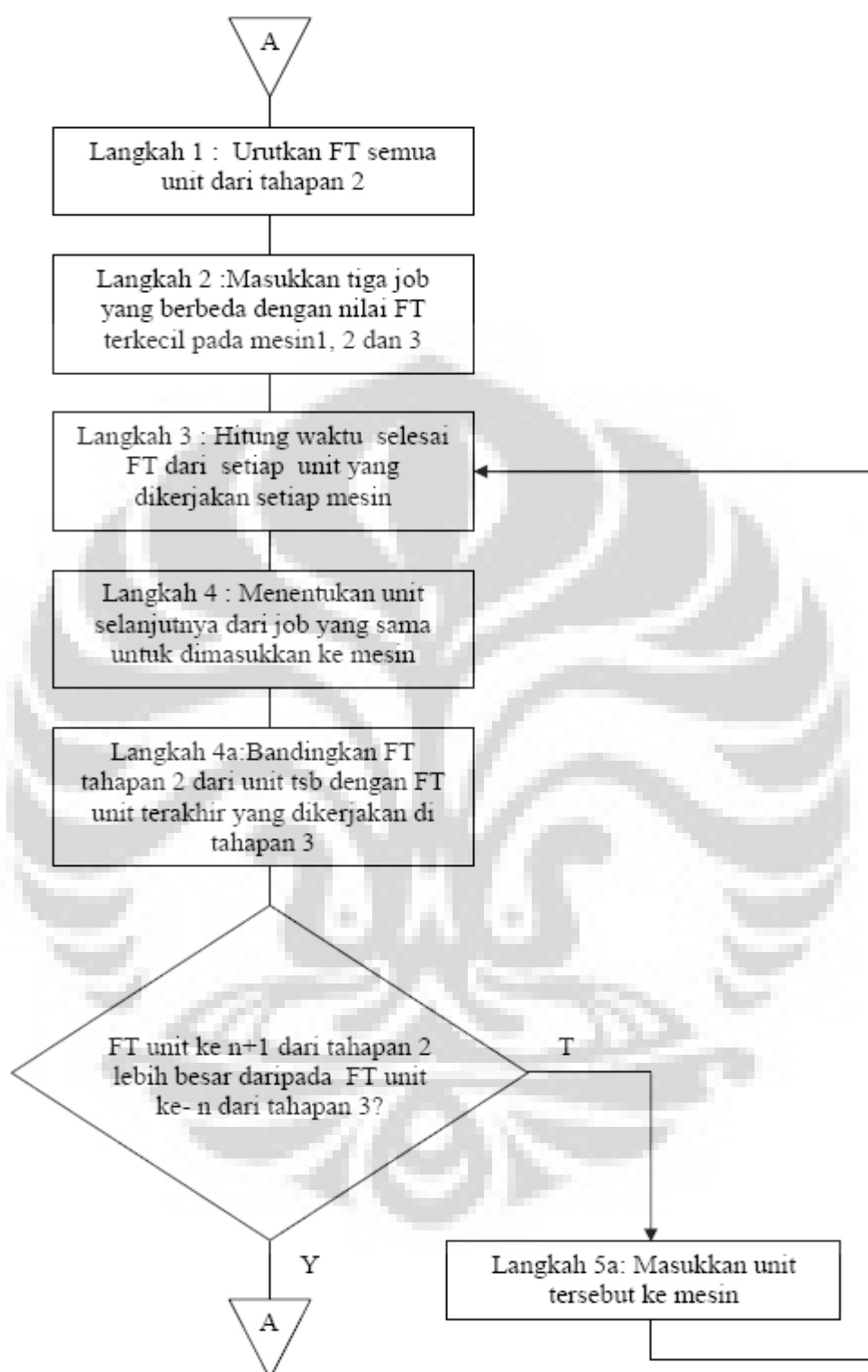
Berdasarkan metode yang telah ditetapkan, dibuatkan algoritma sebagai acuan dalam melakukan penjadwalan seperti pada bagan alir berikut ini.



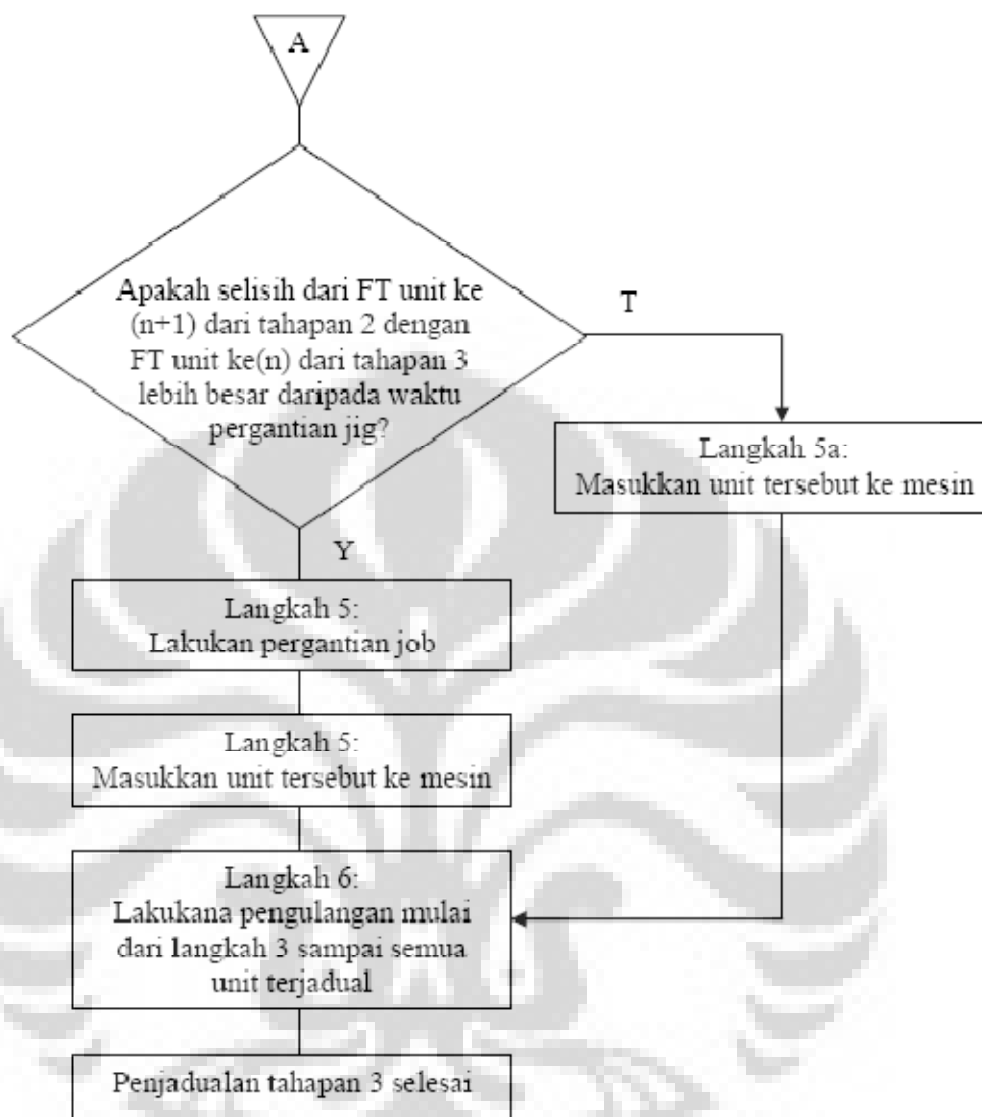
Gambar 4.1. Diagram Alir Algoritma multi product multi stage tahap awal



Gambar 4.1. Diagram Alir Algoritma multi product multi stage tahap awal (lanjt.)



Gambar 4.1. Aliran Algoritma Multiproduct-Multistage tahap awal (sambungan)



Gambar 4.1. Aliran algoritma multiproduct-multistage

#### A. Algoritma untuk penjadwalan Tahapan 1

Langkah 1 Hitung waktu proses yang dibutuhkan untuk setiap job

$$PT_i = ST_i \cdot J_m \quad (4.2)$$

dimana ;  $PT_i$  = process time job  $i$

$ST_i$  = Standard time untuk satu unit job  $i$

$J_m$  = Jumlah unit job  $i$

Nilai perhitungan dibulatkan dalam satuan terdekat



Langkah 2 Urutkan hasil perhitungan mulai dari nilai yang terkecil

Langkah 3 Masukkan nilai urutan pertama pada workstation m1, urutan kedua pada workstation m2 dan urutan ketiga pada workstation m3.

Langkah 4 Lakukan langkah 3 untuk sisa job yang belum terjadwal.

Langkah 5 Hitung waktu selesai ( FT = Finish time) dari setiap unit yang dikerjakan setiap workstation.

- 1) *Finish time* pada unit (j) ke-n merupakan akumulasi *standard time* (ST<sub>i</sub>) dari unit yang telah dikerjakan.

$$FT = \sum_{j=1}^n ST_{ijk1m} \quad (4.3)$$

- 2) Untuk setiap pergantian job pada workstation yang sama, nilai FT pada unit ke-j dari job ke-(i+1) ditambah dengan nilai konstanta  $COT_{k1} = 60$  atau  $FT_{(i+1)jk1m} + COT_{k1}$ .

#### B. Algoritma untuk penjadwalan Tahapan 2

Langkah 1 Berdasarkan perhitungan FT dari semua workstation pada tahapan 1, urutkan nilai FT mulai dari nilai yang terkecil.

Langkah 2 Masukkan lima nilai terkecil pertama (dari langkah 1) pada workstation m1, m2, m3, m4 dan m5.

Langkah 3 Hitung waktu selesai ( FT = Finish time) dari setiap unit yang dikerjakan setiap workstation.

- 1) *Finish time* unit ke-n pada tahapan 2 merupakan nilai penjumlahan dari *release time* (RT) unit tersebut pada tahapan 2 dengan *standard time* unit tersebut pada tahapan 2.
- 2) *Release time* dari unit ke-1 pada tahapan 2 adalah *finish time* unit ke-1 dari Tahapan 1

$$FT_{ilk2} = RT_{ilk2} + ST_{ilk2} \quad (4.4)$$

$$RT_{ilk2} = FT_{ilk1} \quad (4.5)$$

- 3) *Finish time* pada unit ke - n pada setiap mesin merupakan akumulasi *standard time* (ST<sub>i</sub>) dari unit yang telah dikerjakan ditambah *release time* unit ke-1

$$FT = \sum_{j=1}^n ST_{ijk2} + RT_{ilk2} \quad (4-6)$$

Langkah 4 Urutkan FT dari langkah 3 mulai dari nilai yang terkecil.

\*\* Pada saat ini, urutan workstation akan berubah sesuai dengan nilai FT

Langkah 5 Masukkan lima nilai terkecil selanjutnya (dari langkah 1) pada workstation sesuai dengan urutan pada langkah 4.

Langkah 6 Lakukan langkah 3, 4 dan 5 sampai unit yang terakhir dari langkah 1.

### C. Algoritma untuk penjadwalan Tahapan 3

Langkah 1 Berdasarkan perhitungan FT dari semua workstation pada tahapan 2, urutkan nilai FT mulai dari nilai yang terkecil.

Langkah 2 Masukkan tiga job yang berbeda dengan nilai FT terkecil pada mesin 1, 2 dan 3

Langkah 3 Hitung waktu selesai ( FT = Finish time) dari setiap unit yang dikerjakan setiap mesin.

1) Finish time unit ke-1 pada tahapan 3 merupakan penjumlahan dari release time (RT) unit tersebut pada tahapan 3 dengan standard time unit tersebut pada tahapan 3.

2) Release time dari unit ke-1 pada tahapan 3 adalah finish time unit ke-1 dari tahapan 2

$$FT_{ilk3} = RT_{ilk3} + ST_{ilk3} \quad (4.7)$$

$$RT_{ilk3} = FT_{ilk2} \quad (4.8)$$

Langkah 4 Periksa FT unit dengan urutan selanjutnya atau  $FT_{i(j+1)k2}$  dari job yang sama yang telah ditugaskan. Bandingkan dengan FT tersebut dengan unit ke-n pada tahapan 3 yang telah dikerjakan oleh mesin m ( $FT_{ijk3m}$ ).

1) Apabila nilai  $FT_{i(j+1)k2}$  lebih kecil dari  $FT_{ijk3m}$ , maka masukkan unit ke-j pada mesin tersebut.

2) Apabila nilai  $FT_{i(j+1)k2}$  lebih besar dari  $FT_{ijk3m}$  tetapi selisih keduanya lebih kecil dari waktu pergantian jig ( $COT_{k3} = 45$ ), masukkan unit ke-n pada mesin tersebut dan selisih antara keduanya dijumlahkan dengan  $FT_{ijk3m}$

Jika  $FT_{i(j+1)k2} > FT_{ijk3m}$  dan

$$FT_{i(j+1)k2} - FT_{ijk3m} = \alpha < 45 \text{ maka}$$

$$FT_{ijk3m} = FT_{ijk3m} + \alpha.$$

\*\*Pada kondisi ini, mesin dalam keadaan idle sebesar  $\alpha$

- 3) Apabila nilai  $FT_{i(j+1)k2}$  lebih besar dari  $FT_{ijk3m}$  dan selisih keduanya lebih besar atau sama dengan waktu pergantian jig ( $COT_{k3} = 45$ ), maka lakukan pergantian job.

Jika  $FT_{i(j+1)k2} > FT_{ijk3m}$  dan

$$FT_{i(j+1)k2} - FT_{ijk3m} \geq 45, \text{ maka ganti job}$$

- 4) Untuk setiap pergantian job pada workstation yang sama, nilai FT pada unit ke-n dari job(i) ke-n+1 ditambah dengan nilai konstanta  $COT_{k3} = 45$  atau  $FT_{ijk3m} + COT_{k3}$

Langkah 5 Lakukan langkah 3 dan 4 sampai semua unit terjadwal.

#### D. Algoritma untuk penjadwalan Tahapan 4

Langkah 1 Berdasarkan perhitungan FT dari semua workstation pada tahapan 3, urutkan nilai FT mulai dari nilai yang terkecil.

Langkah 2 Masukkan satu nilai terkecil pertama (dari langkah 1)

Langkah 3 Hitung waktu selesai ( FT = Finish time) dari setiap unit yang dikerjakan setiap workstation.

- 1) Finish time unit ke-n pada tahapan 4 merupakan nilai penjumlahan dari release time (RT) unit tersebut pada tahapan 4 dengan *standard time* unit tersebut pada tahapan 4.
- 2) Release time dari unit ke-1 pada tahapan 4 adalah finish time unit ke-1 dari Tahapan 3

$$FT_{i1k4} = RT_{i1k4} + ST_{i1k4} \quad (4.9)$$

$$RT_{i1k4} = FT_{i1k3} \quad (4.10)$$

- 3) Finish time pada unit ke - n pada setiap mesin merupakan akumulasi standard time (ST<sub>i</sub>) dari unit yang telah dikerjakan ditambah release time unit ke-1

$$FT = \sum_{j=1}^n ST_{ijk4} + RT_{i1k4} \quad (4-11)$$

Dari algoritma inialisasi solusi ini akan dibuatkan program penjadwalan dan perhitungan sebagai dasar penetapan nilai *fitness* fungsi tujuan pada program yang menggunakan *Genetic Algorithm*.

### 4.3. Verifikasi Program

Tujuan dari verifikasi program adalah untuk memastikan bahwa program telah dibuat dengan benar sesuai dengan algoritma yang ditetapkan.

- 1) Verifikasi program ini hanya dilakukan terhadap algoritma yang digunakan pada inialisasi solusi.
- 2) Verifikasi program dilakukan dengan membandingkan penjadwalan dan perhitungan yang dilakukan secara manual dengan penjadwalan dan perhitungan yang dihasilkan oleh program
- 3) Penjadwalan dan perhitungan manual dilakukan dengan menggunakan piranti Microsoft Excel.

#### 4.3.1. Penjadwalan dan Perhitungan Manual

##### 4.3.1.1. Penjadwalan Tahapan 1

Penjadwalan tahapan 1 dimulai dengan menghitung total waktu proses setiap job. Total waktu proses adalah jumlah perkalian antara jumlah unit pesanan pada job tertentu dengan *standard time* job tersebut. Data job dan *standard time* setiap tahapan terdapat pada lampiran 1. Hasil perhitungan total waktu proses setiap job terdapat pada lampiran 2.

Dari data pada lampiran 2, dilakukan pemeringkatan job berdasarkan total waktu proses dimulai dari total waktu proses terkecil (lihat lampiran 3). Berdasarkan data pada lampiran 3 tersebut, dilakukan penugasan job pada setiap workstation dimulai dari job dengan urutan pertama untuk workstation m1, urutan kedua untuk workstation m2, urutan ketiga untuk workstation m3, urutan keempat untuk workstation m1 dan seterusnya. Hasil penugasan job pada setiap mesin seperti pada tabel di bawah ini

Tabel 4.4. Urutan Pengerjaan Job Pada Tahapan 1

mesin	urutan pengerjaan job (i)			
m1	2	6	12	5
m2	8	9	10	11
m3	7	4	1	3

Urutan selengkapnya dari pengerjaan dalam penjadwalan produksi setiap mesin pada tahapan 1 adalah seperti pada lampiran 4.

#### 4.3.1.2. Penjadwalan Tahapan 2

Untuk menyusun penjadwalan tahapan 2, terlebih dahulu dilakukan pemeringkatan terhadap FT semua unit yang dikerjakan oleh *workstation* m1, m2 dan m3 pada tahapan 1. Pemeringkatan dimulai dari unit dengan nilai FT terkecil seperti pada lampiran 5. Langkah selanjutnya adalah memasukkan lima unit yang memiliki FT terkecil ke *workstation* m1, m2, m3, m4 dan m5 seperti pada tabel di bawah ini.

Tabel 4.5. Penetapan unit pertama untuk setiap workstation pada tahapan 2

No	Job,unit	FT Tahap I	Work Station
1	(8,1)	135	m1
2	(2,1)	250	m2
3	(8,2)	270	m2
4	(8,3)	405	m3
5	(7,1)	483	m4

Langkah selanjutnya adalah menghitung FT tahapan 2 sesuai dengan persamaan 4.2 untuk setiap job yang sudah terjadwal. Hasil perhitungan FT pada tahapan 2 adalah sebagai berikut Tabel 4.13. Finish time (FT) unit ke-1 dari setiap workstation pada tahapan 2.

Tabel 4.6. Finish time (FT) unit ke-1 dari setiap workstation pada tahapan 2

Job,unit	FT Tahapan I	Standart time (ijk2)	FT(ijk2)	Work station
(8,1)	135	196	331	m1
(2,1)	250	376	626	m2
(8,2)	270	196	466	m2
(8,3)	405	196	601	m3
(7,1)	483	753	1236	m4

Dari perhitungan di atas, dilakukan pemeringkatan FT<sub>ijk2</sub> mulai dari FT terkecil. Hasil pemeringkatan adalah sebagai berikut:

Tabel 4.7. Hasil pemeringkatan FT pada tahapan 2

Job,unit	FT(ijk2)	Work station
(8,1)	331	m1
(8,2)	466	m3
(8,3)	601	m4
(2,1)	626	m2
(7,1)	1236	m5

Berdasarkan hasil pemeringkatan di atas, dapat dilihat bahwa workstation m1 lebih dahulu menyelesaikan pekerjaannya diikuti oleh workstation m3, m4, m2 dan m5. Untuk unit kedua yang akan dikerjakan di workstation m1, diambil unit kedelapan dari lampiran 5, unit keenam untuk untuk m2, unit kedelapan untuk m3, unit keenam untuk m4, dan unit ketujuh untuk m5. Dilakukan kembali perhitungan FT untuk unit kedua sesuai dengan persamaan (4.7). Setelah itu, hasil perhitungan FT tersebut diurutkan kembali seperti pada tabel 4.14 untuk mendapatkan unit ketiga yang akan dikerjakan. Pengulangan kegiatan ini dilakukan sampai semua unit terjadwal. Jadwal produksi untuk tahapan 2 secara lengkap terdapat pada lampiran 6.

#### 4.3.1.3. Penjadwalan Tahapan 3

Untuk penjadwalan tahapan 3, dilakukan pemeringkatan terhadap FT semua unit yang dikerjakan oleh workstation m1, m2, m3, m4 dan m5 pada tahapan 2. Pemeringkatan dimulai dari unit dengan nilai FT terkecil.(lihat

lampiran 7). Dari data tersebut, dipilih tiga unit dari tiga job berbeda yang memiliki nilai FT terkecil. Ketiga unit tersebut dimasukkan ke mesin m1, m2 dan m3 seperti pada tabel di bawah ini

Tabel 4.8. Penetapan unit pertama untuk setiap mesin pada tahapan 3

Job,unit	FT(ijk2)	Work station
(8,1)	331	m1
(8,2)	466	m2
(8,3)	601	m3

Langkah selanjutnya adalah menghitung FT tahapan 3 sesuai dengan persamaan 4.2 untuk setiap job yang sudah terjadwal. Nilai FT<sub>ijk3</sub> tersebut dibandingkan dengan FT unit selanjutnya dengan job yang sama dari tahapan 2 atau FT<sub>i(j+1)k2</sub>.

- 1) Apabila nilai FT<sub>i(j+1)k2</sub> lebih kecil dari FT<sub>ijk3m</sub>, maka masukkan unit ke-j+1 dari tahapan 2 pada mesin tersebut.
- 2) Apabila nilai FT<sub>i(j+1)k2</sub> lebih besar dari FT<sub>ijk3m</sub> tetapi selisih keduanya ( $\alpha$ ) lebih kecil dari waktu pergantian jig ( $COTk3 = 45$ ), masukkan unit ke-j+1 pada mesin tersebut dan selisih antara keduanya dijumlahkan dengan FT<sub>ijk3m</sub>.
- 3) Apabila nilai FT<sub>i(j+1)k2</sub> lebih besar dari FT<sub>ijk3m</sub> dan selisih keduanya lebih besar atau sama dengan waktu pergantian jig ( $COTk3 = 0.67$ ), maka lakukan pergantian job.

Tujuan dari metode di atas adalah untuk menempatkan unit dari job yang sama pada mesin-mesin di tahapan 3 secara berurut. Pengulangan langkah ini Job(i), dilakukan sampai semua unit terjadwal. Hasil penyusunan jadwal produksi untuk tahapan 3.dapat dilihat pada lampiran 8.

#### 4.3.1.4. Penjadwalan Tahapan 4

Untuk menyusun penjadwalan tahapan 4, terlebih dahulu dilakukan pemeringkatan terhadap FT semua unit yang dikerjakan oleh *workstation* m1, m2 dan m3 pada tahapan 3. Pemeringkatan dimulai dari unit dengan nilai FT terkecil. Langkah selanjutnya adalah memasukkan memasukkan nilai dari spesifikasi

kimianya dan nilai jumlah dari masing-masing job. Setelah semua ketemu lalu dibuat iterasi dengan keketentuan jika nilai akumulasi dari job yang masuk lebih kecil dari spesifikasi job, maka proses diteruskan tetapi jika telah melebihi maka proses harus ditunda hingga nilainya sesuai. Mesin akan di reset menjadi nol jika nilai akumulasi job yang telah dip roses telah mencapai 3000

#### 4.3.1.5. Menghitung Makespan

Sesuai dengan persamaan (4.1), dilakukan perhitungan makespan terhadap inisialisasi solusi. Perhitungan makespan adalah sebagai berikut

$$FFs|(Cmax) = (\text{Max}\{FT_{ijk4m}\} - \text{Min}\{RT_{ijk4m}\}) + \\ (\text{Min}\{RT_{ijk4m}\} - \text{Min}\{RT_{ijk3m}\}) + \\ (\text{Min}\{RT_{ijk3m}\} - \text{Min}\{RT_{ijk2m}\}) + \\ (\text{Min}\{RT_{ijk2m}\} - \text{Min}\{RT_{ijk1m}\})$$

dimana  $\forall i \in I, \forall j \in J, \forall m \in M$

notasi

$\text{Max}\{FT_{ijk4m}\}$  : nilai FT tertinggi dari job(i) unit(j) yang dikerjakan oleh seluruh mesin di tahapan 4

$\text{Min}\{RT_{ijk4m}\}$  : nilai RT terendah dari job(i) unit(j) yang dikerjakan oleh seluruh mesin di tahapan 4

$\text{Min}\{RT_{ijk3m}\}$  : nilai RT terendah dari job(i) unit(j) yang dikerjakan oleh seluruh mesin di tahapan 3

$\text{Min}\{RT_{ijk2m}\}$  : nilai RT terendah dari job(i) unit(j) yang dikerjakan oleh seluruh mesin di tahapan 2

$\text{Min}\{RT_{ijk1m}\}$  : nilai RT terendah dari job(i) unit(j) yang dikerjakan oleh seluruh mesin di tahapan 1



Diketahui :

$\text{Max}\{FT_{ijk4m}\} = 13179$ ---- (pada unit ke-12 dari job ke-3 pada tahapan 4)

$\text{Min}\{RT_{ijk4m}\} = 474$  (pada unit ke-1 dari job ke-8 pada mesin m1 di tahapan 3)

$\text{Min}\{RT_{ijk3m}\} = 331$  (pada unit ke-1 dari job ke-8 pada mesin m1 di tahapan 3)

$\text{Min}\{RT_{ijk2m}\} = 135$  (pada -1 dari job ke-8 pada mesin m1 di tahapan 2)

$\text{Min}\{RT_{ijmk1}\} = 0$

maka

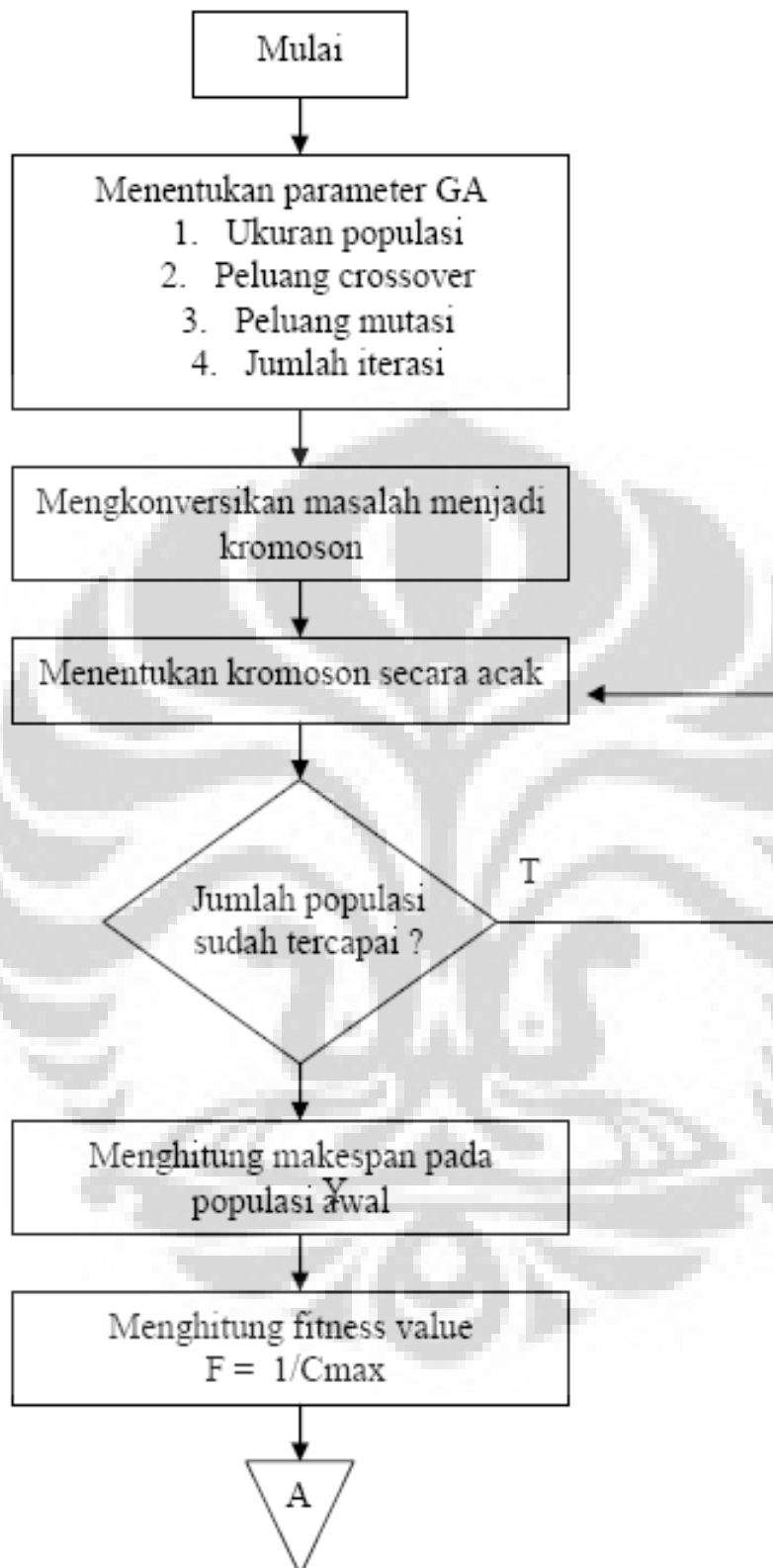
$$\begin{aligned} C_{\max} &= (13179-474) + (474-331) + (331-135) + (135-0) \\ &= 13179 \text{ menit} \end{aligned}$$

Sedangkan jika penghitungan hanya sampai tahapan 3 saja maka  $C_{\max}$  adalah 13093 menit.

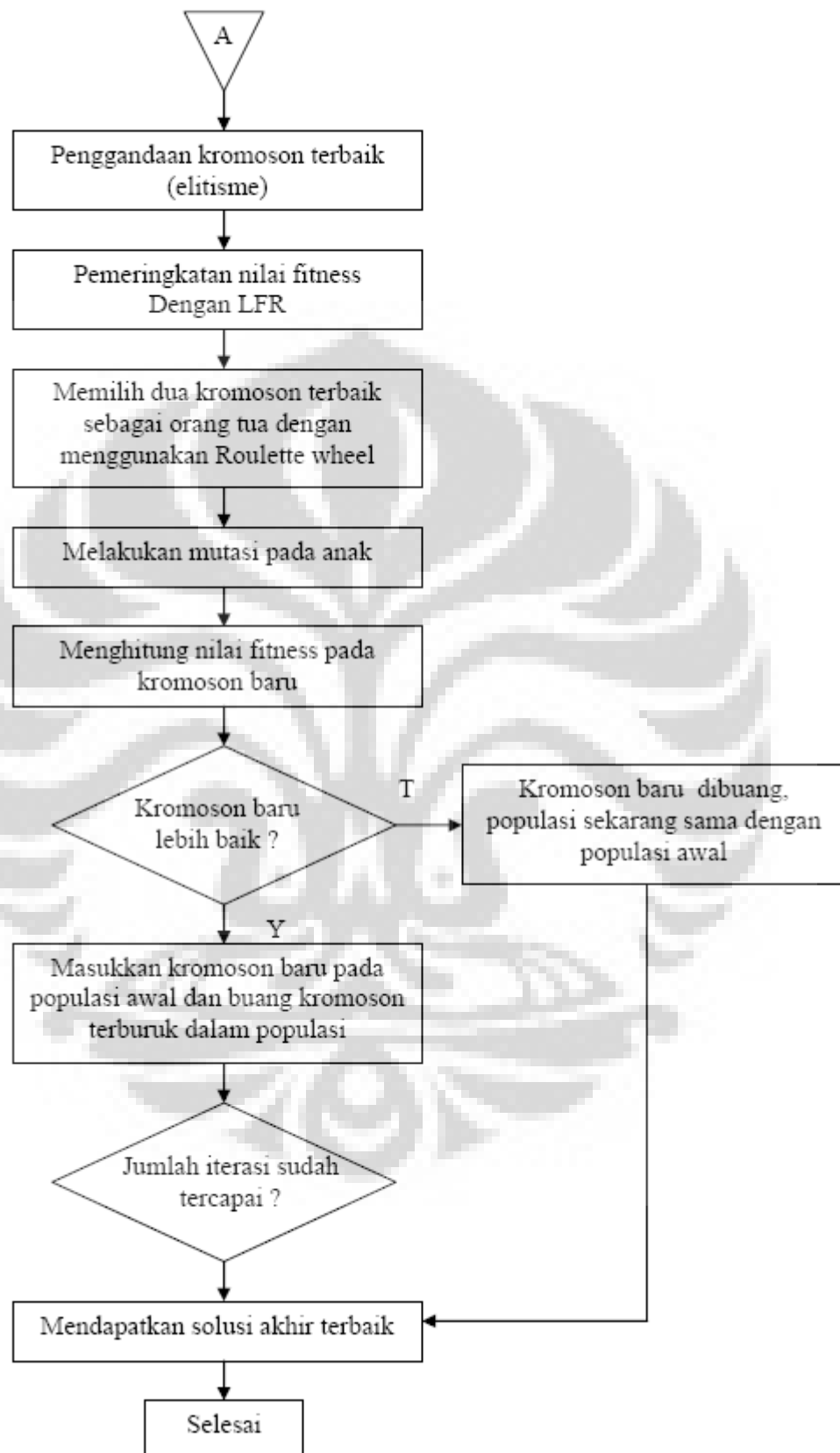
Hasil penjadwalan dan perhitungan secara manual ini dibandingkan dengan hasil penjadwalan dan perhitungan dengan menggunakan program. Data keluaran dari program dapat dilihat pada lampiran 9. Hasil perbandingan memperlihatkan hasil yang sama, dengan demikian program sudah terverifikasi dan dapat digunakan. Untuk selanjutnya, program ini akan menjadi algoritma awal pada program penjadwalan dengan Genetic Algorithm.

#### 4.4. Penjadwalan dengan *Genetic Algorithm*

Berdasarkan penjadwalan yang diperoleh dari inisialisasi solusi, dilakukan optimasi dengan menggunakan metode *Genetic Algorithm* (GA). Pada penelitian ini, penerapan Genetic Algorithm hanya dilakukan pada penjadwalan tahapan 1, untuk penjadwalan tahapan 2, tahapan 3 dan tahapan 4 akan mengikuti penjadwalan hasil GA sesuai dengan algoritma pada butir 4.2.2. Optimasi GA ini akan menggunakan piranti lunak MATLAB 7.0.1. Langkah-langkah yang dilakukan dalam penjadwalan dengan *Genetic Algorithm* adalah sebagai berikut.



Gambar 4.2. Alur algoritma genetika untuk penjadwalan



Gambar 4.2. Alur algoritma genetika untuk penjadwalan (sambungan)

#### 4.4.1. Penetapan parameter *Genetic Algorithm*

Parameter yang dimaksud dalam GA adalah parameter kontrol GA, yaitu ukuran populasi (popsize), peluang crossover (pc) dan peluang mutasi (pm). Sri Kusumadewi & Hari Purnomo, (2005)<sup>9</sup> menyebutkan bahwa ada beberapa rekomendasi yang bisa digunakan dalam penetapan parameter GA.. Rekomendasi parameter tersebut seperti pada tabel di bawah.

Tabel 4.9. Rekomendasi parameter *Genetic Algorithm*

Kondisi	Popsize	pc	pm	Keterangan
Kawasan solusi cukup besar	50	0.6	0.001	Rekomendasi De Jong
Rata-rata fitness setiap generasi digunakan sebagai indikator	30	0.95	0.01	Rekomendasi Grefenstette
Fitness individu terbaik dipantau setiap generasi	80	0.45	0.01	

Disamping itu, baik Sri Kusumadewi dan Hari Purnomo (2005) maupun Suyanto (2005)<sup>10</sup> menyebutkan bahwa ukuran populasi sebaiknya tidak kurang dari 30 dengan argumentasi bahwa sampel mengikuti distribusi normal. Pada penelitian ini, ukuran populasi yang dipakai merupakan matriks dari jumlah mesin pada tahapan 1 yaitu 3 buah mesin dan jumlah job sebanyak 12, sehingga ukuran populasi adalah 36. Peluang mutasi dan peluang crossover masing-masing adalah 0.95 dan 0.01. Dengan penetapan parameter tersebut, maka penelitian ini menggunakan parameter yang direkomendasikan oleh Grefenstette

#### 4.4.2. Pengkorvesian Masalah

Masalah penjadwalan pada tahapan 1 akan dikonversikan sebagai kromosom dengan menggunakan pengkodean (permutation encoding) yang

<sup>9</sup> Sri Kusumadewi & Hari Purnomo, Penyelesaian Masalah Optimasi Dengan Teknik-teknik Heuristik, 2005.

<sup>10</sup> Suyanto, Algoritma Genetika Dalam Matlab, 2005

dimasukkan ke dalam satu matriks baris-kolom dimana baris menunjukkan mesin  $m$  dan kolom menunjukkan job  $i$ .

#### 4.4.3. Penentuan Populasi Awal

Populasi awal ditetapkan secara acak. Dari populasi awal tersebut akan didapat penjadwalan untuk tahapan 1 yang akan digunakan untuk menyusun penjadwalan tahapan 2 dan tahapan 3. Pada tahap ini juga akan didapatkan nilai makespan. Tahap ini merupakan tahap inisialisasi dari GA terhadap penjadwalan tahapan 1, 2 dan 3. Kondisi ini akan diperbaiki dengan menggunakan operasi genetika.

#### 4.4.4. Evaluasi Kromosom

Evaluasi terhadap kromosom dilakukan dengan mencari individu dengan nilai nilai fitness terbaik. Fitness merupakan objektif atau tujuan yang telah ditetapkan. Pada penelitian ini fitness adalah fungsi tujuan untuk meminimalkan makespan, dengan notasi nilai fitness =  $1/C_{max}$ . Untuk mencegah local-optima, dilakukan pemeringkatan terhadap nilai fitness. Metode yang digunakan untuk pemeringkatan adalah rank-based fitness atau disebut juga linear rank fitness dimana populasi diurutkan menurut nilai objektifnya. Nilai fitness suatu individu dapat dihitung sebagai

$$\text{Fitness}(\text{Pos}) = 2 - SP + 2(SP - 1)(\text{Pos} - 1) / (N - 1) \quad (4.11)$$

dimana :

$N$  = jumlah individu dalam populasi

$\text{Pos}$  = posisi individu dalam populasi ( $\text{Pos}=1$  adalah terendah dan

$\text{Pos}=N$  adalah tertinggi)

$SP$  = selection pressure  $SP \in [1,2]$

#### 4.4.5. Seleksi

Berdasarkan hasil pemeringkatan nilai fitness, akan diambil dua kromosom untuk dijadikan orang tua (parent). Penyeleksian menggunakan metode roulette

wheel selection, dimana kromosom akan dipetakan dalam suatu potongan lingkaran dari sebuah roda rolet. Pembagian potongan lingkaran akan disesuaikan nilai fitnessnya, dimana kromosom dengan nilai fitness terbaik akan memiliki potongan lingkaran yang lebih besar sehingga kemungkinan untuk terpilih menjadi orang tua juga lebih besar.

#### 4.4.6. Kawin Silang (crossover)

Metode kawin silang yang digunakan yaitu Linear Order Crossover dimana proses kawin silang akan dimulai pada kromosom ke-2 bila ukuran populasi ganjil, dan pada kromosom ke-3 bila ukuran populasi genap.

- a Membuat dua buah titik potong TP1 dan TP2 secara acak untuk memotong dua buah kromosom orangtua K1 dan K2. Hasil pemotongan tersebut disebut kromosom anak A1 dan A2
- b Kedua kromosom anak A1 dan A2 akan disilangkan terhadap potongan kromosom K1 dan K2
- c Kromosom baru telah terbentuk.

#### 4.4.7. Mutasi

Metode mutasi yang digunakan adalah *exchange mutation*, dimana dua buah kromosom akan ditukarkan secara acak. Mutasi hanya dilakukan terhadap kromosom anak. setelah operasi genetika dilakukan, langkah selanjutnya adalah generational replacement dimana kromosom pada populasi yang lama akan digantikan oleh kromosom baru, kecuali kromosom yang pertama akan tetap dipertahankan. Langkah ini akan dilakukan berulang sampai iterasi yang ditetapkan telah tercapai. Pada saat pengulangan selesai, optimasi penjadwalan telah dilakukan.

#### 4.4.8. Pengolahan *Genetic Algorithm* Pada Penjadwalan

Setelah penetapan parameter dan operasi GA, dilakukan pengolahan data untuk mendapatkan penjadwalan terbaik dengan makespan minimal. Tabel di bawah hasil running program.

Tabel 4.11. Hasil running program Matlab

Runing kel	makespan	Elapsed time
1	15893	14.609000
5	13096	12.703000
10	12079	86.391000
20	12973	12.141000
30	16715	58.203000
40	13444	16.422000
50	14552	14.000000
60	16458	43.390000
70	12927	13.000000
80	11110	17.609000
90	15198	14.718000
100	15104	15.922000

#### 4.5. Analisis

Tidak adanya metode penjadwalan yang baku pada proses produksi *silicon polished wafer* di SPTI dapat mengakibatkan bertambahnya waktu proses produksi dan potensi keterlambatan pengiriman. Untuk itu diperlukan suatu metode penjadwalan sebagai solusi masalah tersebut. Beberapa faktor yang dipertimbangkan dalam menetapkan metode penjadwalan proses produksi *silicon polished wafer* adalah:

- 1) Kompleksitas masalah, dimana terdapat;
  - (a) 12 type barang yang dikerjakan,
  - (b) 4 tahapan proses produksi yang harus dilalui oleh setiap type barang,
  - (c) waktu proses yang berbeda untuk setiap type barang dan tahapan proses yang dilaluinya dan
  - (d) jumlah pesanan yang berbeda untuk setiap type barang
- 2) Metode penjadwalan dapat dilakukan dengan cepat sehingga koreksi penjadwalan dapat dilakukan sesegara mungkin oleh scheduler.

Berdasarkan hal di atas, penelitian ini mengusulkan untuk menggunakan metode heuristik *Genetic Algorithm*. Metode heuristik akan mendapatkan solusi yang

mendekati optimal dengan waktu pemrosesan yang lebih cepat dibandingkan dengan metode perhitungan konvensional.

#### 4.5.1. Analisis Metode

Penelitian ini diawali dengan membuat sebuah inisialisasi solusi terhadap kasus penjadwalan produksi silicon polished wafer di SPTI. Inisialisasi ini penting dilakukan karena tidak adanya metode penjadwalan baku di SPTI. Inisialisasi solusi disusun untuk membuat suatu model penjadwalan baru yang memiliki tujuan meminimalkan makespan. Inisialisasi solusi yang dihasilkan harus dapat merepresentasikan kondisi sebenarnya dari kondisi penjadwalan yang ada. Penjadwalan dengan kondisi multiproduct-multistage yang ada pada proses produksi silicon polished wafer diatasi dengan pendekatan dispatching rule (schedule given in advance) yang berbeda untuk setiap stage (tahap), sementara kendala untuk meminimalkan waktu idle akibat pergantian jig diatasi dengan metode batch (family) job.

Penerapan metode SPT (*shortest processing time*) untuk menetapkan type barang yang dikerjakan oleh sekumpulan mesin yang tersusun secara paralel dapat meminimalkan mengurangi waktu tunggu (*idle*) pada mesin di *stage* (tahap selanjutnya). Dengan metode SPT, mesin-mesin pada *stage* (tahap) pertama akan mengerjakan type barang yang memiliki waktu proses terpendek sehingga barang-barang tersebut dapat dengan segera dikirimkan ke tahapan berikutnya. Pemilihan metode SPT ini digabungkan dengan metode pengelompokan unit pesanan menjadi sebuah family. Pertimbangan pengelompokan ini didasarkan pada limitasi fleksibilitas proses pada tahapan kesatu dimana pengerjaan dilakukan dengan menggunakan sebuah seperangkat alat kontrol dan setiap type barang hanya memiliki sebuah alat kontrol. Dengan metode family, penambahan waktu akibat pergantian job pada mesin yang sama menjadi minimal. Penggabungan kedua metode ini menghasilkan SPT untuk setiap type barang (job/family)

Untuk stage (tahap) kedua, digunakan metode EFT (*earliest finish time*) dimana mesin-mesin pada tahapan kedua mengirimkan barang yang telah selesai lebih dahulu ke *stage* (tahap) berikutnya. Hal ini berbeda dengan aliran manufaktur yang pada umumnya menggunakan metode FIFO (first in first out)



walaupun sama-sama memiliki tahapan yang banyak (multistage). Pada kasus ini, susunan mesin yang paralel dan variasi waktu pengerjaan yang besar antara type barang yang satu dengan yang lain dapat menimbulkan waktu penyelesaian yang berbeda. Dengan kondisi tersebut, sulit untuk dipastikan bahwa barang yang pertama masuk pada salah satu mesin akan menjadi barang pertama yang dikerjakan oleh mesin lain di tahapan selanjutnya. Untuk itu, penerapan EFT adalah hal tepat dilakukan pada kasus ini.

Pada tahapan ketiga, digunakan metode EFT untuk menentukan mesin mana yang akan berganti job pada saat suatu job telah selesai dilakukan. Scheduler membandingkan dua mesin atau lebih yang pada saat hampir bersamaan telah menyelesaikan salah satu jobnya Metode EFT digunakan untuk melihat mesin mana yang telah menyelesaikan jobnya lebih dahulu. Tahapan ketiga juga memiliki limitasi fleksibilitas proses yang sama dengan tahapan satu, dimana pengerjaan dilakukan dengan bantuan jig. Atas kondisi yang serupa seperti ini, maka pengelompokan unit pesanan menjadi sebuah family dilakukan juga pada proses di tahapan ketiga. Metode pada inisialisasi solusi diterjemahkan pada suatu algoritma *multiproduct-multistage*. Algoritma ini akan menjadi salah satu acuan pada pembuatan *Genetic Algorithm* disamping fungsi tujuan meminimalkan makespan.

Optimasi terhadap inisialisasi solusi dilakukan dengan menggunakan *Genetic Algorithm*. Penerapan GA hanya dilakukan untuk stage (tahapan) kesatu, sementara tahapan kedua, ketiga dan keempat mengikuti algoritma *multiproduct-multistage* sebelumnya. Hal tersebut dilakukan untuk meminimalkan waktu running program pada saat program GA dijalankan. Operasi genetika dilakukan dengan parameter yang direkomendasikan oleh Grefenstette yaitu ukuran populasi sebesar 36, peluang crossover 0.95 dan peluang mutasi 0.01. Pada saat evaluasi kromosom dilakukan pemeringkatan nilai fitness dengan menggunakan *linear fitness rank* dimana *fitness* masing-masing kromosom akan diurut berdasarkan nilai objektifnya. Sesuai dengan aturan pada penyusunan GA, *fitness* pada kasus penjadwalan ini adalah fungsi meminimalkan, sehingga nilai fitness adalah  $1/C_{max}$ . Seleksi merupakan salah satu operasi genetika yang harus dilakukan. Pada penelitian ini, metode seleksi yang digunakan adalah metode roulette wheel.

Penerapan metode roulette wheel dilakukan karena metode ini mengambil data dari nilai fitness yang sudah didapat. Kromosom yang memiliki nilai fitness yang tinggi akan ditempatkan pada potongan lingkaran yang lebih besar pada sebuah roda rolet sehingga peluang untuk terpilih menjadi kromosom orang tua juga lebih besar. Crossover atau kawin silang dilakukan untuk mendapatkan kromosom baru sebagai hasil dari kombinasi antara dua kromosom.

Kromosom baru tersebut juga akan mengalami evaluasi dan seleksi untuk bersaing dengan kromosom lama. Mutasi dilakukan untuk membuang kromosom yang memiliki nilai fitness yang buruk, sehingga hanya kromosom-kromosom yang memiliki nilai fitness yang baik yang akan bertahan. Operasi genetika terakhir dari penyusunan GA adalah generational replacement dimana kromosom pada populasi yang lama akan digantikan oleh kromosom baru, kecuali kromosom yang pertama akan tetap dipertahankan. Kromosom pertama pada *generational replacement* diasumsikan sebagai kromosom terbaik pada satu generasi, sehingga langkah selanjutnya adalah mencari kromosom yang terbaik untuk yang kedua dan seterusnya. Dengan melakukan evolusi terhadap kromosom-kromosom tersebut, GA akan menghasilkan solusi yang mendekati optimal.

#### **4.5.2. Analisis Hasil**

Perhitungan makespan pada inisialisasi solusi adalah sebesar 13093 menit, sementara itu dengan penggunaan *Genetic Algorithm* dihasilkan 11110 pada iterasi ke 80. Pada iterasi ke-10 nilai makespan sudah turun menjadi 12079 atau sudah lebih bagus dari inisialisasi awal akan tetapi saat running ke 30,70 dan sebagian besar lainnya nilai makespan cenderung lebih tinggi dari inisialisasi awal. Hal ini dimungkinkan karena pada inisialisasi solusi telah terjadi proses optimasi sehingga peluang untuk berevolusi menjadi kecil. Untuk hal tersebut, perlu dipertimbangkan untuk merubah parameter operasi genetika dengan parameter lain selain yang direkomendasikan oleh Grefenstette.

## BAB 5

### KESIMPULAN

Dari hasil penelitian tentang pemodelan penjadwalan untuk proses produksi tipe flexible flow shop dengan algoritma genetika pada industri pengolahan *silicon polished wafer* untuk fungsi tujuan menurunkan makespan dapat disimpulkan bahwa :

1. Inisialisasi solusi dilakukan untuk menetapkan metode awal model penjadwalan sebagai dasar optimasi penjadwalan baru yang akan dikembangkan.
2. Dengan kondisi proses yang memiliki banyak tahapan dan banyaknya variasi type barang, penjadwalan produksi dapat dilakukan dengan menggunakan metode *dispatching rule* yaitu memberikan penjadwalan di muka (*schedule in advance*) untuk memaksimalkan penjadwalan pada tahapan selanjutnya.
3. Jenis *dispatching rule* dapat dikombinasikan antara satu tahapan dengan tahapan berikutnya atas pertimbangan kondisi dan kendala yang dimiliki. Dalam penelitian ini, penggabungan metode SPT (*shortest processing time*) dan EFT (*earliest finish time*) digunakan secara bersamaan sesuai dengan tujuan penjadwalan yaitu meminimalkan makespan.
4. Pengelompokan unit pesanan menjadi sebuah *family* dilakukan untuk meminimalkan *idle time* sebagai akibat dari keterbatasan fleksibilitas proses pada kondisi tertentu.
5. Optimasi penjadwalan terhadap inisialisasi solusi dilakukan dengan menggunakan *Genetic Algorithm* untuk mendapatkan penjadwalan yang mendekati optimal.
6. Hasil optimasi dengan menggunakan *Genetic Algorithm* menunjukkan pengurangan makespan sebesar 1983 menit atau 8.45% dibandingkan dengan hasil penjadwalan pada inisialisasi solusi.

## DAFTAR REFERENSI

O. Etiler, B. Toklu, M. Atak and J. Wilson, “ *A Genetic Algorithm For Flow Shop Scheduling Problems*”, Journal of The Operational Research Society, Vol 55, No 8,2004, hal 830-835

Lixing, T, Jiyin L,(2002), Modified genetic Algorithm for the flow shop sequencing problem to minimize mean flow time,Journal of intelligent manufacturing,Vol. 13 no 1 pg. 61

Baker, Kenneth R. (2001). *Elements of Sequencing and Scheduling*. Hanover: Tock School of Business, Dartmouth College

Kusumadewi, Sri & Hari Purnomo.(2005). *Penyelesaian Masalah Optimasi Menggunakan Teknik-teknik Heuristik*. Yogyakarta: Graha Ilmu

Pinedo, Michael. (1993). *Scheduling Theory, Algorithms and Systems*. New Jersey: Prentice Hall

Suyanto (2006). *Algoritma Genetika dalam MATLAB*. Yogyakarta: Penerbit ANDI

Lu, Yufeng, (2001) Thesis : *Scheduling of wafer test process in semiconductor manufacturing*, Virginia university

Lampiran 1. Waktu baku untuk setiap job pada setiap tahapan

Type	Job (i)	Waktu Baku			
		Tahapan I	Tahapan II	Tahapan III	Tahapan IV
B296-010-D	(1)	355	552	300	79
C306-034-A	(2)	250	376	208	68
C905-015-C	(3)	441	689	376	86
C906-058-B	(4)	378	592	328	82
C906-059-C	(5)	404	627	358	81
C906-101-A	(6)	236	356	223	66
C906-103-A	(7)	483	753	442	89
D105-001-B	(8)	135	196	143	57
D806-003-D	(9)	377	588	357	81
D866-024-E	(10)	382	595	367	79
D866-025-D	(11)	412	639	402	82
D866-026-B	(12)	390	606	394	79

Lampiran 2. Data total waktu proses pada tahapan I

Job (i)	Jumlah (Jm)	Waktu Baku (STi)	Total Waktu Proses (PTi=STi * Jm)
(1)	10	355	3550
(2)	1	250	250
(3)	12	441	5292
(4)	7	378	2646
(5)	10	404	4040
(6)	7	236	1652
(7)	2	483	966
(8)	5	135	675
(9)	6	377	2262
(10)	8	382	3056
(11)	11	412	4532
(12)	7	390	2730

Lampiran 3. Pemeringkatan job berdasar total waktu proses pada tahapan 1

Job (i)	Jumlah (Jm)	Waktu Baku (STi)	Total Waktu Proses (PTi=STi * Jm)
(2)	1	250	250
(8)	5	135	675
(7)	2	483	966
(6)	7	236	1652
(9)	6	377	2262
(4)	7	378	2646
(12)	7	390	2730
(10)	8	382	3056
(1)	10	355	3550
(5)	10	404	4040
(11)	11	412	4532
(3)	12	441	5292

Lampiran 4. Jadwal produksi pada tahapan I

Penjadwalan *workstation* m1

Urutan	1	2	3	4	5	6	7	8	9	10	11	12	13
Job, unit (i,j)	(2,1)	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(12,1)	(12,2)	(12,3)	(12,4)	(12,5)
Start (RT)	0	310*	546	782	1018	1254	1490	1726	2022*	2412	2802	3192	3582
Finish (FT)	250	546	782	1018	1254	1490	1726	1962	2412	2802	3192	3582	3972

Urutan	14	15	16	17	18	19	20	21	22	23	24	25	26
Job, unit (i,j)	(12,6)	(12,7)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(5,9)	(5,10)	
Start (RT)	3582	3972	4422*	4826	5230	5634	6038	6442	6846	7250	7654	8058	
Finish (FT)	3972	4362	4826	5230	5634	6038	6442	6846	7250	7654	8058	8462	

\*) : RT dengan penambahan waktu  $COTk1 = 60$  sebagai waktu pergantian blade dan setup mesin



Lampiran 4. Jadwal produksi pada tahapan I (sambungan)

Penjadwalan *workstation* m2

Urutan	1	2	3	4	5	6	7	8	9	10	11	12	13
Job, unit (i,j)	(8,1)	(8,2)	(8,3)	(8,4)	(8,5)	(9,1)	(9,2)	(9,3)	(9,4)	(9,5)	(9,6)	(10,1)	(10,2)
Start (RT)	0	135	270	405	540	735*	1112	1489	1866	2243	2620	3057*	3439
Finish (FT)	135	270	405	540	675	1112	1489	1866	2243	2620	2997	3439	3821

Urutan	14	15	16	17	18	19	20	21	22	23	24	25	26
Job, unit (i,j)	(10,3)	(10,4)	(10,5)	(10,6)	(10,7)	(10,8)	(11,1)	(11,2)	(11,3)	(11,4)	(11,5)	(11,6)	(11,7)
Start (RT)	3821	3956	4091	4226	4361	4496	4691*	5103	5515	5927	6339	6751	7163
Finish (FT)	3956	4091	4226	4361	4496	4631	5103	5515	5927	6339	6751	7163	7575

Urutan	27	28	29	30
Job, unit (i,j)	(11,8)	(11,9)	(11,10)	(11,11)
Start (RT)	7575	7987	8399	8811
Finish (FT)	7987	8399	8811	9223

\*) : RT dengan penambahan waktu  $COTk1 = 60$  sebagai waktu pergantian blade dan setup mesin

Lampiran 4. Jadwal produksi pada tahapan I (sambungan)

Penjadwalan *workstation* m3

Urutan	1	2	3	4	5	6	7	8	9	10	11	12	13
Job, unit (i,j)	(7,1)	(7,2)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)	(4,7)	(1,1)	(1,2)	(1,3)	(1,4)
Start (RT)	0	483	1026*	1404	1782	2160	2538	2916	3294	3732*	4087	4442	4797
Finish (FT)	483	966	1404	1782	2160	2538	2916	3294	3672	4087	4442	4797	5152

Urutan	14	15	16	17	18	19	20	21	22	23	24	25	26
Job, unit (i,j)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)	(1,10)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)
Start (RT)	4797	5152	5507	5862	6217	6572	6987*	7428	7869	8310	8751	9192	9633
Finish (FT)	5152	5507	5862	6217	6572	6927	7428	7869	8310	8751	9192	9633	10074

Urutan	26	27	28	29	30
Job, unit (i,j)	(3,8)	(3,9)	(3,10)	(3,11)	(3,12)
Start (RT)	9633	10074	10515	10956	11397
Finish (FT)	10074	10515	10956	11397	11838

\*) : RT dengan penambahan waktu  $COTk1 = 60$  sebagai waktu pergantian blade dan setup mesin

Lampiran 5. Hasil pemeringkatan semua unti pada tahapan I

No.	Job, Unit (I <sub>j</sub> )	FT	No.	Job, Unit (I <sub>j</sub> )	FT
1	(8,1)	135	24	(4,4)	2538
2	(2,1)	250	25	(9,5)	2620
3	(8,2)	270	26	(12,2)	2802
4	(8,3)	405	27	(4,5)	2916
5	(7,1)	483	28	(9,6)	2997
6	(8,4)	540	29	(12,3)	3192
7	(6,1)	546	30	(4,6)	3294
8	(8,5)	675	31	(10,1)	3439
9	(6,2)	782	32	(12,4)	3582
10	(7,2)	966	33	(4,7)	3672
11	(6,3)	1018	34	(10,2)	3821
12	(9,1)	1112	35	(10,3)	3956
13	(6,4)	1254	36	(12,5)	3972
14	(4,1)	1404	37	(12,6)	3972
15	(9,2)	1489	38	(1,1)	4087
16	(6,5)	1490	39	(10,4)	4091
17	(6,6)	1726	40	(10,5)	4226
18	(4,2)	1782	41	(10,6)	4361
19	(9,3)	1866	42	(12,7)	4362
20	(6,7)	1962	43	(1,2)	4442
21	(4,3)	2160	44	(10,7)	4496
22	(9,4)	2243	45	(10,8)	4631
23	(12,1)	2412	46	(1,3)	4797

Lampiran 5. Hasil pemeringkatan semua unii pada tahapan I (sambungan)

No.	Job, Unit (I,j)	FT	No.	Job, Unit (I,j)	FT
47	(5,1)	4826	67	(3,1)	7428
48	(11,1)	5103	68	(11,7)	7575
49	(1,4)	5152	69	(5,8)	7654
50	(1,5)	5152	70	(3,2)	7869
51	(5,2)	5230	71	(11,8)	7987
52	(1,6)	5507	72	(5,9)	8058
53	(11,2)	5515	73	(3,3)	8310
54	(5,3)	5634	74	(11,9)	8399
55	(1,7)	5862	75	(5,10)	8462
56	(11,3)	5927	76	(3,4)	8751
57	(5,4)	6038	77	(11,10)	8811
58	(1,8)	6217	78	(3,5)	9192
59	(11,4)	6339	79	(11,11)	9223
60	(5,5)	6442	80	(3,6)	9633
61	(1,9)	6572	81	(3,7)	10074
62	(11,5)	6751	82	(3,8)	10074
63	(5,6)	6846	83	(3,9)	10515
64	(1,10)	6927	84	(3,10)	10956
65	(11,6)	7163	85	(3,11)	11397
66	(5,7)	7250	86	(3,12)	11838

Lampiran 6. Jadwal produksi tahapan II untuk setiap *workstation*

Work station	Urutan	1	2	3	4	5	6	7	8	9	10	11	12
m1	Job,unit	(8,1)	(8,4)	(6,3)	(6,5)	(4,3)	(12,2)	(10,1)	(12,5)	(10,6)	(1,3)	(5,2)	(11,3)
	FT	331	736	1374	1846	2752	3408	4034	4640	5235	5787	6414	7053
m2	Job,unit	(2,1)	(6,1)	(9,1)	(6,6)	(9,4)	(4,5)	(12,4)	(12,6)	(12,7)	(5,1)	(1,6)	(5,4)
	FT	626	982	1700	2082	2831	3508	4188	4794	5400	6027	6579	7206
m3	Job,unit	(8,2)	(8,5)	(6,4)	(4,2)	(12,1)	(9,6)	(4,7)	(1,1)	(1,2)	(11,1)	(11,2)	(1,8)
	FT	466	871	1610	2374	3018	3606	4264	4816	5368	6007	6646	7198
m4	Job,unit	(8,3)	(6,2)	(4,1)	(9,3)	(4,4)	(12,3)	(10,2)	(10,4)	(10,7)	(1,4)	(5,3)	(11,4)
	FT	601	1138	1996	2584	3176	3798	4416	5011	5606	6158	6785	7424
m5	Job,unit	(7,1)	(7,2)	(9,2)	(6,7)	(9,5)	(4,6)	(10,3)	(10,5)	(10,8)	(1,5)	(1,7)	(5,5)
	FT	1236	1989	2577	2933	3521	4113	4708	5303	5898	6450	7002	7629

Lampiran 6. Jadwal produksi tahapan II untuk setiap *workstation*(sambungan)

Work station	Urutan	13	14	15	16	17	18
m1	Job,unit	(1,9)	(5,7)	(11,8)	(3,4)	(3,7)	(3,12)
	FT	7605	8232	8871	9560	10763	12527
m2	Job,unit	(11,5)	(3,1)	(5,9)	(11,10)	(3,8)	
	FT	7845	8534	9161	9800	10763	
m3	Job,unit	(5,6)	(11,7)	(3,3)	(3,5)	(3,9)	
	FT	7825	8464	9153	9881	11204	
m4	Job,unit	(1,10)	(5,8)	(11,9)	(11,11)	(3,10)	
	FT	7976	8603	9242	9881	11645	
m5	Job,unit	(11,6)	(3,2)	(5,10)	(3,6)	(3,11)	
	FT	8268	8957	9584	10322	12086	

Lampiran 7. Hasil pemeringkatan semua unit pada tahapan II

No	Job,unit	FT <sub>ijk2</sub>	No	Job,unit	FT <sub>ijk2</sub>
1	(8,1)	331	24	(4,4)	3176
2	(8,2)	466	25	(12,2)	3408
3	(8,3)	601	26	(4,5)	3508
4	(2,1)	626	27	(9,5)	3521
5	(8,4)	736	28	(9,6)	3606
6	(8,5)	871	29	(12,3)	3798
7	(6,1)	982	30	(10,1)	4034
8	(6,2)	1138	31	(4,6)	4113
9	(7,1)	1236	32	(12,4)	4188
10	(6,3)	1374	33	(4,7)	4264
11	(6,4)	1610	34	(10,2)	4416
12	(9,1)	1700	35	(12,5)	4640
13	(6,5)	1846	36	(10,3)	4708
14	(7,2)	1989	37	(12,6)	4794
15	(4,1)	1996	38	(1,1)	4816
16	(6,6)	2082	39	(10,4)	5011
17	(4,2)	2374	40	(10,6)	5235
18	(9,2)	2577	41	(10,5)	5303
19	(9,3)	2584	42	(1,2)	5368
20	(4,3)	2752	43	(12,7)	5400
21	(9,4)	2831	44	(10,7)	5606
22	(6,7)	2933	45	(1,3)	5787
23	(12,1)	3018	46	(10,8)	5898

Lampiran 7. Hasil pemeringkatan semua unit pada tahapan II

No	Job,unit	FT <sub>ijk2</sub>	No	Job,unit	FT <sub>ijk2</sub>
47	(11,1)	6007	67	(11,7)	8464
48	(5,1)	6027	68	(3,1)	8534
49	(1,4)	6158	69	(5,8)	8603
50	(5,2)	6414	70	(11,8)	8871
51	(1,5)	6450	71	(3,2)	8957
52	(1,6)	6579	72	(3,3)	9153
53	(11,2)	6646	73	(5,9)	9161
54	(5,3)	6785	74	(11,9)	9242
55	(1,7)	7002	75	(3,4)	9560
56	(11,3)	7053	76	(5,10)	9584
57	(1,8)	7198	77	(11,10)	9800
58	(5,4)	7206	78	(3,5)	9881
59	(11,4)	7424	79	(11,11)	9881
60	(1,9)	7605	80	(3,6)	10322
61	(5,5)	7629	81	(3,7)	10763
62	(5,6)	7825	82	(3,8)	10763
63	(11,5)	7845	83	(3,9)	11204
64	(1,10)	7976	84	(3,10)	11645
65	(5,7)	8232	85	(3,11)	12086
66	(11,6)	8268	86	(3,12)	12527



Lampiran 8. Jadwal produksi tahapan 3

Penjadwalan mesin m1

Urutan	1	2	3	4	5	6	7	8	9	10	11	12
job,unit	(8,1)	(8,2)	(8,3)	(8,4)	(8,5)	(9,2)	(9,3)	(9,4)	(9,5)	(9,6)	(10,1)	(10,2)
RT	331	466	601	736	871	2622	2584	2831	3521	3606	4124	4416
FT	474	617	760	903	1046	2979	3336	3693	4050	4407	4774	5141

Urutan	14	15	16	17	18	19	20	21	22	23	24	25
job,unit	(10,4)	(10,6)	(10,5)	(10,7)	(10,8)	(11,2)	(11,3)	(11,4)	(11,5)	(11,6)	(11,7)	(11,8)
RT	5011	5235	5303	5606	5898	6646	7053	7424	7845	8268	8464	8871
FT	5875	6242	6609	6976	7343	7745	8147	8549	8951	9353	9755	10157

Urutan	26	27	28
job,unit	(11,9)	(11,10)	(11,11)
RT	9242	9800	9881
FT	10559	10961	11363

Lampiran 8. Jadwal produksi tahapan 3 (sambungan)

Penjadwalan mesin m2

Urutan	1	2	3	4	5	6	7	8	9	10	11	12
job,unit	(2,1)	(7,1)	(9,1)	(7,2)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)	(4,7)	(1,1)
RT	626	1281	1745	1989	1996	2374	2752	3176	3508	4113	4264	4885
FT	834	1723	2102	2544	2872	3200	3528	3856	4184	4512	4840	5185

Urutan	13	14	15	16	17	18	19	20	21	22	23	24
job,unit	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)	(1,10)	(3,1)	(3,2)	(3,3)
RT	5368	5787	6158	6450	6579	7002	7198	7605	7976	8534	8957	9153
FT	5668	6087	6458	6758	7058	7358	7658	7958	8276	8910	9333	9709

Urutan	25	26	27	28	29	30	31	32	33
job,unit	(3,4)	(3,5)	(3,6)	(3,7)	(3,8)	(3,9)	(3,10)	(3,11)	(3,12)
RT	9560	9881	10322	10763	10763	11204	11645	12086	12527
FT	10085	10461	10837	11213	11589	11965	12341	12717	13093

Lampiran 8. Jadwal produksi tahapan 3 (sambungan)

Penjadwalan mesin m3

Urutan	1	2	3	4	5	6	7	8	9	10	11	12
job,unit	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(12,1)	(12,2)	(12,3)	(12,4)	(12,5)
RT	982	1138	1374	1610	1846	2082	2933	3201	3408	3798	4188	4640
FT	1205	1428	1651	1874	2097	2320	3156	3595	3989	4383	4777	5171

Urutan	13	14	15	16	17	18	19	20	21	22	23	24	25
job,unit	(12,6)	(12,7)	(11,1)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(5,9)	(5,10)
RT	4794	5400	6007	6027	6414	6785	7206	7629	7825	8232	8603	9161	9584
FT	5565	5959	6409	6767	7125	7483	7841	8199	8557	8915	9273	9631	9989

Lampiran 9. Jadwal produksi tahapan IV

FTijk3	job,unit	Qty	akumulasi	spec	Urutan Proses	t	FT
474	(8,1)	100	100	1750	1	57	531
617	(8,2)	100	200	1750	2	57	674
760	(8,3)	100	300	1750	3	57	817
834	(2,1)	199	499	600	4	68	902
903	(8,4)	100	599	1750	5	57	960
1046	(8,5)	100	699	1750	6	57	1103
1205	(6,1)	187	886	3000	7	66	1271
1428	(6,2)	187	1073	3000	8	66	1494
1651	(6,3)	187	1260	3000	9	66	1717
1723	(7,1)	399	1659	1750	10	89	1812
1874	(6,4)	187	1846	3000	11	66	1940
2097	(6,5)	187	2033	3000	12	66	2163
2320	(6,6)	187	2220	3000	13	66	2386
2872	(4,1)	309	2529	3000	14	82	2954
3156	(6,7)	187	2716	3000	15	66	3222
2102	(9,1)	308	308	2250	16	81	3303
2544	(7,2)	399	707	1750	17	89	3392
2979	(9,2)	308	1015	2250	18	81	3473
3200	(4,2)	309	1324	3000	19	82	3555
3336	(9,3)	308	1632	2250	20	81	3636
3528	(4,3)	309	1941	3000	21	82	3718
3693	(9,4)	308	2249	2250	22	81	3799
3856	(4,4)	309	2558	3000	23	82	3938
4184	(4,5)	309	2867	3000	24	82	4266
3595	(12,1)	320	320	1750	25	79	4345
3989	(12,2)	320	640	1750	26	79	4424
4050	(9,5)	308	948	2250	27	81	4505
4383	(12,3)	320	1268	1750	28	79	4584
4407	(9,6)	308	1576	2250	29	81	4665
4512	(4,6)	309	1885	3000	30	82	4747
4840	(4,7)	309	2194	3000	31	82	4922
6767	(5,1)	331	2525	3000	32	81	6848
7125	(5,2)	331	2856	3000	33	81	7206
4774	(10,1)	313	313	1750	34	79	7285

Lampiran 9. Jadwal produksi tahapan IV (sambungan)

FTijk3	job,unit	Qty	akumulasi	spec	Urutan Proses	t	FT
4777	(12,4)	320	633	1750	35	79	4856
5141	(10,2)	313	946	1750	36	79	5220
5171	(12,5)	320	1266	1750	37	79	5299
5185	(1,1)	289	1555	1750	38	79	5378
7483	(5,3)	331	1886	3000	39	81	7564
7841	(5,4)	331	2217	3000	40	81	7922
8199	(5,5)	331	2548	3000	41	81	8280
8557	(5,6)	331	2879	3000	42	81	8638
5508	(10,3)	313	313	1750	43	79	8717
5565	(12,6)	320	633	1750	44	79	8796
5668	(1,2)	289	922	1750	45	79	8875
5875	(10,4)	313	1235	1750	46	79	8954
5959	(12,7)	320	1555	1750	47	79	9033
8915	(5,7)	331	1886	3000	48	81	9114
9273	(5,8)	331	2217	3000	49	81	9354
9631	(5,9)	331	2548	3000	50	81	9712
9989	(5,10)	331	2879	3000	51	81	10070
6087	(1,3)	289	289	1750	52	79	10149
6242	(10,6)	313	602	1750	53	79	10228
6409	(11,1)	338	940	1750	54	82	10310
6458	(1,4)	289	1229	1750	55	79	10389
6609	(10,5)	313	1542	1750	56	79	10468
6758	(1,5)	289	289	1750	57	79	10547
6976	(10,7)	313	602	1750	58	79	10626
7058	(1,6)	289	891	1750	59	79	10705
7343	(10,8)	313	1204	1750	60	79	10784
7358	(1,7)	289	1493	1750	61	79	10863
7658	(1,8)	289	289	1750	62	79	10942
7745	(11,2)	338	627	1750	63	82	11024
7958	(1,9)	289	916	1750	64	79	11103
8147	(11,3)	338	1254	1750	65	82	11185
8276	(1,10)	289	1543	1750	66	79	11264
8549	(11,4)	338	338	1750	67	82	11346
8910	(3,1)	363	701	1750	68	86	11432

Lampiran 9. Jadwal produksi tahapan IV (sambungan)

FTijk3	job,unit	Qty	akumulasi	spec	Urutan Proses	t	FT
8951	(11,5)	338	1039	1750	69	82	11514
9333	(3,2)	363	1402	1750	70	86	11600
9353	(11,6)	338	1740	1750	71	82	11682
9709	(3,3)	363	363	1750	72	86	11768
9755	(11,7)	338	701	1750	73	82	11850
10085	(3,4)	363	1064	1750	74	86	11936
10157	(11,8)	338	1402	1750	75	82	12018
10559	(11,9)	338	1740	1750	76	82	12100
10461	(3,5)	363	363	1750	77	86	12186
10837	(3,6)	363	726	1750	78	86	12272
10961	(11,10)	338	1064	1750	79	82	12354
11213	(3,7)	363	1427	1750	80	86	12440
11363	(11,11)	338	338	1750	81	82	12522
11589	(3,8)	363	701	1750	82	86	12608
11965	(3,9)	363	1064	1750	83	86	12694
12341	(3,10)	363	1427	1750	84	86	12780
12717	(3,11)	363	363	1750	85	86	12866
13093	(3,12)	363	726	1750	86	86	13179

Lampiran 10. Hasil Running Program

```
Generasi ke
masukkan generasi ke 1
n =
  1
M1S1 =
  8  4  1  2
M2S1 =
  3  6 12  5
M3S1 =
  7  9 10 11
DeretMesin1_2 =
Columns 1 through 15
  8  8  3  4  4  9  3  1 10  1 11 11 11  6 12
Columns 16 through 17
  5  5
DeretMesin2_2 =
Columns 1 through 15
  8  8  9  9  3  4 10  3  1 10  2 11  6  6 12
Columns 16 through 18
 12  5  5
DeretMesin3_2 =
Columns 1 through 15
  8  3  4  3  9  3  1  1 10  1 11 11  6  6 12
Columns 16 through 17
  5  5
DeretMesin4_2 =
Columns 1 through 15
  3  7  9  4  4 10  3 10 10  1 11  3 11 12 12
Columns 16 through 17
  5  5
DeretMesin5_2 =
Columns 1 through 15
  7  4  3  9  3  1 10  1  1 11 11 11  6  6 12
Columns 16 through 17
  5  5
DeretMesin1_3 =
  8
  9
DeretMesin2_3 =
  2
  3
  5
 10
 11
 12
DeretMesin3_3 =
  1
  4
  6
```

7  
C\_Max =

15893

Elapsed time is 14.609000 seconds.

\*\*\*\*\*

\*\*\*

Generasi ke  
masukkan generasi ke 5

n = 5

M1S1 =

7 9 10 8

M2S1 =

3 4 1 2

M3S1 =

11 6 12 5

DeretMesin1\_2 =

Columns 1 through 15

11 7 9 11 3 3 11 6 6 12 12 5 4 5 4

Columns 16 through 17

1 1

DeretMesin2\_2 =

Columns 1 through 15

3 11 3 9 9 10 10 6 6 8 12 12 5 4 5

Columns 16 through 18

1 1 2

DeretMesin3\_2 =

Columns 1 through 15

7 3 11 3 11 10 6 10 10 8 12 5 5 4 5

Columns 16 through 17

1 1

DeretMesin4\_2 =

Columns 1 through 15

11 9 9 11 3 11 3 6 6 8 8 12 4 5 4

Columns 16 through 17

1 1

DeretMesin5\_2 =

Columns 1 through 15

3 11 3 9 11 3 10 10 10 8 12 3 5 4 5

Columns 16 through 17

1 1

DeretMesin1\_3 =

2

4

5

10

11



DeretMesin2\_3 =

1  
3  
6  
8  
12

DeretMesin3\_3 =

7  
9

C\_Max =

13096

Elapsed time is 12.703000 seconds.

\*\*\*\*\*

\*\*\*

Generasi ke  
masukkan generasi ke 10

n =

10

M1S1 =

3 4 1 5

M2S1 =

11 9 10 8

M3S1 =

2 6 12 7

DeretMesin1\_2 =

Columns 1 through 15

2 11 3 3 11 12 12 3 9 10 10 8 4 1 1

Columns 16 through 17

5 5

DeretMesin2\_2 =

Columns 1 through 15

11 6 6 6 3 11 12 12 9 10 10 8 4 4 1

Columns 16 through 18

1 5 5

DeretMesin3\_2 =

Columns 1 through 15

3 11 3 11 12 3 11 9 9 10 8 8 4 1 1

Columns 16 through 17

5 5

DeretMesin4\_2 =

Columns 1 through 15

6 3 6 11 12 3 11 9 7 10 10 4 4 1 1

Columns 16 through 17

5 5

DeretMesin5\_2 =

Columns 1 through 15

6 6 11 12 3 11 3 7 9 10 3 8 4 1 1

Columns 16 through 17

5 5

DeretMesin1\_3 =

2

3

7

DeretMesin2\_3 =

6

12

DeretMesin3\_3 =

1

4

5

8

9

10

11

C\_Max =

12079

Elapsed time is 86.391000 seconds.

\*\*\*\*\*

\*\*\*

Generasi ke

masukkan generasi ke 20

n =

20

M1S1 =

11 9 10 7

M2S1 =

2 4 1 8

M3S1 =

5 6 12 3

DeretMesin1\_2 =

Columns 1 through 15

2 11 5 4 11 5 11 1 9 1 1 8 12 10 3

Columns 16 through 17

7 3

DeretMesin2\_2 =

Columns 1 through 15

5 4 11 5 4 11 1 9 6 9 12 8 10 12 10

Columns 16 through 17

3 3

DeretMesin3\_2 =

Columns 1 through 15

11 5 4 11 5 1 6 6 6 6 9 8 8 12 10

Columns 16 through 18

3 3 3

DeretMesin4\_2 =

```

Columns 1 through 15
 4 11 5 4 11 5 6 6 1 12 8 12 12 10 3
Columns 16 through 17
 7 3
DeretMesin5_2 =
Columns 1 through 15
 5 4 11 5 1 1 11 1 9 1 9 10 10 3 10
Columns 16 through 17
 3 3
DeretMesin1_3 =
 2
 4
DeretMesin2_3 =
 1
 3
 5
 8
DeretMesin3_3 =
 6
 7
 9
10
11
12
C_Max =
 12973
Elapsed time is 12.141000 seconds.

*****
***
Generasi ke
masukkan generasi ke 30
n =
 30
M1S1 =
 7 6 12 11
M2S1 =
 3 4 1 5
M3S1 =
 2 9 10 8
DeretMesin1_2 =
Columns 1 through 15
 2 7 6 9 3 12 10 12 12 8 11 11 4 4 1
Columns 16 through 17
 5 5
DeretMesin2_2 =
Columns 1 through 15
 3 9 3 6 10 3 12 10 8 8 11 11 4 1 1
Columns 16 through 18
 1 5 5

```

DeretMesin3\_2 =  
 Columns 1 through 15  
 7 6 6 3 6 10 3 3 10 11 11 3 4 1 1  
 Columns 16 through 17  
 5 5  
 DeretMesin4\_2 =  
 Columns 1 through 15  
 9 3 9 9 12 12 10 10 8 8 11 11 4 1 1  
 Columns 16 through 17  
 5 5

DeretMesin5\_2 =  
 Columns 1 through 15  
 3 9 6 6 3 10 3 12 11 11 11 4 4 1 1  
 Columns 16 through 17  
 5 5

DeretMesin1\_3 =  
 2  
 9

DeretMesin2\_3 =  
 1  
 3  
 5  
 8  
 10  
 11

DeretMesin3\_3 =  
 4  
 6  
 7  
 12

C\_Max =  
 16715

Elapsed time is 58.203000 seconds.

\*\*\*\*\*  
 \*\*\*

Generasi ke  
 masukkan generasi ke 40

n =  
 40  
 M1S1 =  
 3 9 10 8  
 M2S1 =  
 7 6 12 2  
 M3S1 =  
 11 4 1 5

DeretMesin1\_2 =  
 Columns 1 through 15

```

11 7 3 11 12 12 11 4 4 1 1 9 9 5 10
Columns 16 through 17
5 8
DeretMesin2_2 =
Columns 1 through 15
3 11 6 3 3 11 3 12 4 1 1 1 5 10 5
Columns 16 through 17
10 8
DeretMesin3_2 =
Columns 1 through 15
7 6 11 6 6 12 11 3 4 4 1 1 9 9 5
Columns 16 through 17
10 10
DeretMesin4_2 =
Columns 1 through 15
11 3 11 3 11 3 12 12 4 1 3 5 5 10 5
Columns 16 through 17
10 8
DeretMesin5_2 =
Columns 1 through 15
3 6 6 6 11 3 12 4 2 1 1 9 9 5 10
Columns 16 through 18
5 8 0
DeretMesin1_3 =
5
9
10
11
12
DeretMesin2_3 =
1
2
3
4
8
DeretMesin3_3 =
6
7
C_Max =
13444
Elapsed time is 16.422000 seconds.

```

.....

```

Generasi ke
masukkan generasi ke 50
n =
50
M1S1 =
3 6 12 11

```

```

M2S1 =
  8 4 1 5
M3S1 =
  2 9 10 7
DeretMesin1_2 =
  Columns 1 through 15
    8 8 4 9 9 3 1 1 1 7 5 5 6 6 12
  Columns 16 through 18
    11 11 11
DeretMesin2_2 =
  Columns 1 through 15
    2 3 4 3 4 3 10 1 10 7 5 5 6 12 12
  Columns 16 through 17
    11 11
DeretMesin3_2 =
  Columns 1 through 15
    8 8 3 4 4 4 3 10 10 1 5 5 5 6 12
  Columns 16 through 17
    11 11
DeretMesin4_2 =
  Columns 1 through 15
    8 9 9 9 3 10 1 3 1 1 5 3 6 6 12
  Columns 16 through 17
    11 11
DeretMesin5_2 =
  Columns 1 through 15
    3 9 3 4 10 10 3 10 1 1 5 6 5 12 12
  Columns 16 through 17
    11 11
DeretMesin1_3 =
  8
  9
DeretMesin2_3 =
  2
  3
  7
DeretMesin3_3 =
  1
  4
  5
  6
  10
  11
  12
C_Max =
  14552

```

Elapsed time is 14.000000 seconds.

```

*****
***

```

Generasi ke  
masukkan generasi ke 60

n =

60

M1S1 =

8 6 12 5

M2S1 =

11 9 10 3

M3S1 =

7 4 1 2

DeretMesin1\_2 =

Columns 1 through 15

8 8 6 6 4 11 12 11 1 1 9 5 10 5 10

Columns 16 through 18

3 3 3

DeretMesin2\_2 =

Columns 1 through 15

8 8 11 4 6 4 4 12 9 5 1 9 2 10 5

Columns 16 through 17

3 3

DeretMesin3\_2 =

Columns 1 through 15

8 11 4 6 11 12 11 1 1 9 5 1 5 5 10

Columns 16 through 17

3 3

DeretMesin4\_2 =

Columns 1 through 15

11 7 11 6 12 11 1 12 9 5 1 1 10 5 3

Columns 16 through 17

3 3

DeretMesin5\_2 =

Columns 1 through 15

7 6 6 11 4 4 12 11 12 1 9 5 10 10 10

Columns 16 through 17

3 3

DeretMesin1\_3 =

6

8

DeretMesin2\_3 =

2

3

5

9

10

11

12

DeretMesin3\_3 =

1

4

7

C\_Max =  
16458  
Elapsed time is 43.390000 seconds.

\*\*\*\*\*

\*\*\*

Generasi ke  
masukkan generasi ke 70

n =

70

M1S1 =

3 6 12 5

M2S1 =

11 9 10 7

M3S1 =

8 4 1 2

DeretMesin1\_2 =

Columns 1 through 15

8 8 11 4 4 11 3 11 9 1 10 10 6 6 12

Columns 16 through 18

12 5 5

DeretMesin2\_2 =

Columns 1 through 15

8 8 3 11 3 4 11 3 1 9 2 10 7 6 12

Columns 16 through 17

5 5

DeretMesin3\_2 =

Columns 1 through 15

8 11 4 3 11 3 1 1 9 1 10 10 6 6 12

Columns 16 through 17

5 5

DeretMesin4\_2 =

Columns 1 through 15

11 3 3 11 3 1 1 9 9 1 10 3 7 12 12

Columns 16 through 17

5 5

DeretMesin5\_2 =

Columns 1 through 15

3 4 11 4 4 11 3 1 1 9 10 10 6 6 12

Columns 16 through 17

5 5

DeretMesin1\_3 =

4

8

DeretMesin2\_3 =

1

6

7

11

12



DeretMesin3\_3 =

2  
3  
5  
9  
10

C\_Max =

12927

Elapsed time is 13.000000 seconds.

\*\*\*\*\*

\*\*\*

Generasi ke  
masukkan generasi ke 80

n =

80

M1S1 =

2 4 1 3

M2S1 =

8 9 10 11

M3S1 =

7 6 12 5

DeretMesin1\_2 =

Columns 1 through 15

8 8 9 4 9 4 12 10 1 12 10 5 3 11 3

Columns 16 through 17

5 3

DeretMesin2\_2 =

Columns 1 through 15

2 7 9 6 6 9 1 12 10 1 1 11 5 5 11

Columns 16 through 17

3 11

DeretMesin3\_2 =

Columns 1 through 15

8 8 6 6 6 6 10 1 12 10 5 11 11 5 5

Columns 16 through 18

11 11 3

DeretMesin4\_2 =

Columns 1 through 15

8 4 4 9 4 12 10 1 12 10 5 3 11 3 5

Columns 16 through 17

11 3

DeretMesin5\_2 =

Columns 1 through 15

7 4 6 4 9 1 12 10 1 1 1 5 3 11 3

Columns 16 through 17

3 3

DeretMesin1\_3 =

4  
8

```

11
12
DeretMesin2_3 =
  2
  5
  9
 10
DeretMesin3_3 =
  1
  3
  6
  7
C_Max =
 11110
Elapsed time is 17.609000 seconds.

```

.....

Generasi ke  
masukkan generasi ke 90

```

n =
 90
M1S1 =
  7  6 12  8
M2S1 =
 11  4  1  3
M3S1 =
  5  9 10  2
DeretMesin1_2 =
Columns 1 through 15
  5  7 11  6  5 12 11  9 12  8  4 10  1  1  1
Columns 16 through 17
  3  3
DeretMesin2_2 =
Columns 1 through 15
 11  5  5  5 12 11  9 12  8  8  9  4 10 10  1
Columns 16 through 17
  3  3
DeretMesin3_2 =
Columns 1 through 15
  7 11  5 11  5 12 11  9  4  4  4 10  1  1  3
Columns 16 through 17
  3  3
DeretMesin4_2 =
Columns 1 through 15
  5  6  6 11 11  5 12  4  9  9 10  1 10 10  1
Columns 16 through 17
  3  3
DeretMesin5_2 =
Columns 1 through 15

```

```

11 6 6 6 6 11 5 12 4 8 8 10 1 1 2
Columns 16 through 18
3 3 3
DeretMesin1_3 =
2
5
8
12
DeretMesin2_3 =
1
3
4
9
10
11
DeretMesin3_3 =
6
7
C_Max =
15198
Elapsed time is 14.718000 seconds.

```

```

*****
***

```

```

Generasi ke
masukkan generasi ke 100
n =
100
M1S1 =
5 9 10 2
M2S1 =
3 6 12 11
M3S1 =
8 4 1 7
DeretMesin1_2 =
Columns 1 through 15
8 8 5 4 4 5 3 1 9 1 10 10 3 6 12
Columns 16 through 18
11 11 11
DeretMesin2_2 =
Columns 1 through 15
8 8 3 5 3 4 5 3 1 9 7 10 6 6 12
Columns 16 through 17
11 11
DeretMesin3_2 =
Columns 1 through 15
5 3 5 4 4 5 3 1 1 10 10 10 6 12 12
Columns 16 through 17

```

```
11 11
DeretMesin4_2 =
Columns 1 through 15
8 5 4 3 5 3 1 9 9 1 10 10 6 6 12
Columns 16 through 17
11 11
DeretMesin5_2 =
Columns 1 through 15
3 4 3 5 3 1 9 1 9 1 7 2 6 12 12
Columns 16 through 17
11 11
DeretMesin1_3 =
4
8
DeretMesin2_3 =
1
5
10
11
DeretMesin3_3 =
2
3
6
7
9
12
C_Max =
15104
Elapsed time is 15.922000 seconds.
```

## Lampiran 11. M-Sribt Program

```
% Genetic Algorithm
clear
clc
pc = 0.25;
pm = 0.01;
t1 = 355;
t2 = 250;
t3 = 441;
t4 = 378;
t5 = 404;
t6 = 236;
t7 = 483;
t8 = 135;
t9 = 377;
t10 = 382;
t11 = 412;
t12 = 390;
disp(' Banyaknya Generasi ?' )
n = input('banyaknya n Generasi ')

% n mencerminkan banyaknya generasi
T = [t1;t2;t3;t4;t5;t6;t7;t8;t9;t10;t11;t12];

Job = [ 1 2 3 4 5 6 7 8 9 10 11 12];
% inisialisasi populasi
populasi = [2 6 12 5;8 9 10 11;7 4 1 3];

Waktu =[t2 t6 t12 t5;t8 t9 t10 t11;t7 t4 t1 t3];
Fitness = zeros(3,1);
Fitness (1,1)=sum(Waktu(1,:));
Fitness (2,1)=sum(Waktu(2,:));
Fitness (3,1)=sum(Waktu(3,:));
Total_Fitness = sum(Fitness(:));

%%% Roulette Wheel
for t =1:3
Fitness_Relatif (t,1) = Fitness(t,1)/Total_Fitness;
end

for t =1:3
a = t-1;
if a==0
Fitness_Kumulatif (t,1) = Fitness_Relatif(t,1);
else
Fitness_Kumulatif(t,1) = Fitness_Relatif(t,1)
+Fitness_Kumulatif(a,1);
end
end

% bangkitkan bilangan acak
for u =1:3

acak(u,1) = rand(1);
```

```

end

% Melakukan Seleksi
for u =1:3
    if u<3
        if Fitness_Kumulatif(u,1)< acak(u,1)
            if Fitness_Kumulatif(u,1)<Fitness_Kumulatif(u+1,1)
                a = populasi(u,:);
                populasi(u,:)= populasi(u+1,:);
                populasi(u+1,:) =a;
            end
        end
    end
    if u == 3
        if Fitness_Kumulatif(u,1)< acak(u,1)
            if Fitness_Kumulatif(u,1)<Fitness_Kumulatif(u-1,1)
                a = populasi(u,:);
                populasi(u,:)= populasi(u-1,:);
                populasi(u-2,:)= populasi(u-1,:);
                populasi(u-1,:) =a;
            end
        end
    end
end

% melakukan X -OVER
% Karena hanya ada size populasi sebesar 3
% maka bila hanya ada 1 Fitness Kumulatif > bilangan acak ,maka
cross over
% ditiadakan, kalau ada 3 maka buang yang ketiga (berlaku untuk
genap)
x = populasi(1,:);
y = populasi(3,:);
populasi(1,2:end)=y(1,2:end);
populasi(3,2:end)=x(1,2:end);
% MUTASI
%pm = 0.01 berarti kita merubah 1 persen dari 3x16 bit atau 4
bit atau
%1 job saja untuk saat ini kita ubah dari job 12 ke job 1
xx = populasi(1,4);
yy = populasi(2,1);
populasi(2,1) = xx;
populasi(1,4) = yy;

% Dirubah menjadi Ranking_St

    savefile = 'GAny.mat';
    N = n;
    Rank_St = zeros(12,N);
    Rank_St(1:12,1)= reshape(populasi,12,1);
    save(savefile, 'Rank_St')

% Memasuki selanjutnya

```

```

for i = 2:n
biner1 = populasi(1,1);
biner2 = populasi(1,2);
biner3 = populasi(1,3);
biner4 = populasi(1,4);
biner5 = populasi(2,1);
biner6 = populasi(2,2);
biner7 = populasi(2,3);
biner8 = populasi(2,4);
biner9 = populasi(3,1);
biner10 = populasi(3,2);
biner11 = populasi(3,3);
biner12 = populasi(3,4);

% inisialisasi populasi
populasi = [biner1 biner2 biner3 biner4;biner5 biner6 biner7
biner8;biner9 biner10 biner11 biner12];

Waktu =[T(populasi(1,1)) T(populasi(1,2)) T(populasi(1,3))
T(populasi(1,4)); T(populasi(2,1)) T(populasi(2,2))
T(populasi(2,3)) T(populasi(2,4)); T(populasi(3,1))
T(populasi(3,2)) T(populasi(3,3)) T(populasi(3,4))];
Fitness = zeros(3,1);
Fitness (1,1)=sum(Waktu(1,:));
Fitness (2,1)=sum(Waktu(2,:));
Fitness (3,1)=sum(Waktu(3,:));
Total_Fitness = sum(Fitness(:));

%% Roulette Wheel
for t =1:3
Fitness_Relatif (t,1) = Fitness(t,1)/Total_Fitness;
end

for t =1:3
a = t-1;
if a==0
Fitness_Kumulatif (t,1) = Fitness_Relatif(t,1);
else
Fitness_Kumulatif(t,1) = Fitness_Relatif(t,1)
+Fitness_Kumulatif(a,1);
end
end

% bangkitkan bilangan acak
for u =1:3

acak(u,1) = rand(1);

end

% Melakukan Seleksi
for u =1:3
if u<3
if Fitness_Kumulatif(u,1)< acak(u,1)
if Fitness_Kumulatif(u,1)<Fitness_Kumulatif(u+1,1)
a = populasi(u,:);
populasi(u,:) = populasi(u+1,:);

```

```

        populasi(u+1,:) =a;
        end
    end
end
if u == 3
    if Fitness_Kumulatif(u,1)< acak(u,1)
        if Fitness_Kumulatif(u,1)<Fitness_Kumulatif(u-1,1)
            a = populasi(u,:);
            populasi(u,:)= populasi(u-1,:);
            populasi(u-2,:)= populasi(u-1,:);
            populasi(u-1,:) =a;
        end
    end
end
end
% melakukan cross over
% Karena hanya ada size populasi sebesar 3
% maka bila hanya ada 1 Fitness Kumulatif > bilangan acak ,maka
cross over
% ditiadakan, kalau ada 3 maka buang yang ketiga (berlaku untuk
genap)
% Kita ambil posisi cross over di titik 4
xx = populasi(1,4);
yy = populasi(2,1);
populasi(2,1) = xx;
populasi(1,4) = yy;

% Dirubah menjadi Ranking_St
savefile = 'GAny.mat';

Rank_St(1:12,i)= reshape(populasi,12,1);
save(savefile, 'Rank_St')

end

clc
clear
tic
I = zeros(12);
% isi data job i1
I (1,1:10) =1;
i1 = I(1,:);
i1(1,12) = sum(i1(:));

% isi data job i2
I (2,1:1) =1;
i2 = I(2,:);
i2(1,12) = sum(i2(:));

% isi data job i3

```



```

I (3,1:12) =1;
i3 = I(3,:);
i3(1,12) = sum(i3(:));

% isi data job i4
I (4,1:7) =1;
i4 = I(4,:);
i4(1,12) = sum(i4(:));

% isi data job i5
I (5,1:10) =1;
i5 = I(5,:);
i5(1,12) = sum(i5(:));

% isi data job i6
I (6,1:7) =1;
i6 = I(6,:);
i6(1,12) = sum(i6(:));

% isi data job i7
I (7,1:2) =1;
i7 = I(7,:);
i7(1,12) = sum(i7(:));

% isi data job i8
I (8,1:5) =1;
i8 = I(8,:);
i8(1,12) = sum(i8(:));

% isi data job i9
I (9,1:6) =1;
i9 = I(9,:);
i9(1,12) = sum(i9(:));

% isi data job i10
I (10,1:8) =1;
i10 = I(10,:);
i10(1,12) = sum(i10(:));

% isi data job i11
I (11,1:11) =1;
i11 = I(11,:);
i11(1,12) = sum(i11(:));

% isi data job i12
I (12,1:7) =1;
i12 = I(12,:);
i12(1,12) = sum(i12(:));

I= [i1;i2;i3;i4;i5;i6;i7;i8;i9;i10;i11;i12];
% memberikan nilai pada waktu
t1 = 355;
t2 = 250;
t3 = 441;
t4 = 378;
t5 = 404;

```

```

t6 = 236;
t7 = 483;
t8 = 135;
t9 = 377;
t10 = 382;
t11 = 412;
t12 = 390;

T = [t1;t2;t3;t4;t5;t6;t7;t8;t9;t10;t11;t12];

% membuat penjadwalan
load GAnyA
disp(' Generasi ke ')
n = input(' masukkan generasi ke ')
% Pemilihan untuk stage
M1S1 = [Rank_St(1,n) Rank_St(4,n) Rank_St(7,n) Rank_St(10,n)]
M2S1 = [Rank_St(2,n) Rank_St(5,n) Rank_St(8,n) Rank_St(11,n)]
M3S1 = [Rank_St(3,n) Rank_St(6,n) Rank_St(9,n) Rank_St(12,n)]

pause

% T comparison Tc
TcM1S1 = zeros(12,44);
TcM2S1 = zeros(12,44);
TcM3S1 = zeros(12,44);
% Mesin 1 Stage 1
for k =1:44
    if k <=I(M1S1(1,1),12)
        TcM1S1(M1S1(1,1),k) = T(M1S1(1,1),1) * I(M1S1(1,1),k);
    end
    if k <=I(M1S1(1,2),12)
        TcM1S1(M1S1(1,2),k+I(M1S1(1,1),12)) = T(M1S1(1,2),1) *
I(M1S1(1,2),k);
    end
    if k <=I(M1S1(1,3),12)
        TcM1S1(M1S1(1,3),k+I(M1S1(1,1),12)+ I(M1S1(1,2),12)) =
T(M1S1(1,3),1) * I(M1S1(1,3),k);
    end
    if k <=I(M1S1(1,4),12)
        TcM1S1(M1S1(1,4),k+I(M1S1(1,1),12)+I(M1S1(1,2),12)+I(M1S1(1,3),12)
)= T(M1S1(1,4),1) * I(M1S1(1,4),k);
    end
end
% Mesin 2 Stage 1
for k =1:44
    if k <=I(M2S1(1,1),12)
        TcM2S1(M2S1(1,1),k) = T(M2S1(1,1),1) * I(M2S1(1,1),k);
    end
    if k <=I(M2S1(1,2),12)
        TcM2S1(M2S1(1,2),k+I(M2S1(1,1),12)) = T(M2S1(1,2),1) *
I(M2S1(1,2),k);
    end
    if k <=I(M2S1(1,3),12)
        TcM2S1(M2S1(1,3),k+I(M2S1(1,1),12)+ I(M2S1(1,2),12)) =
T(M2S1(1,3),1) * I(M2S1(1,3),k);
    end
end

```

```

        if k <=I(M2S1(1,4),12)

TcM2S1(M2S1(1,4),k+I(M2S1(1,1),12)+I(M2S1(1,2),12)+I(M2S1(1,3),12)
)= T(M2S1(1,4),1) * I(M2S1(1,4),k);
        end
    end
% Mesin 3 Stage 1
for k =1:44
    if k <=I(M3S1(1,1),12)
        TcM3S1(M3S1(1,1),k) = T(M3S1(1,1),1) * I(M3S1(1,1),k);
    end
    if k <=I(M3S1(1,2),12)
        TcM3S1(M3S1(1,2),k+I(M3S1(1,1),12)) = T(M3S1(1,2),1) *
I(M3S1(1,2),k);
    end
    if k <=I(M3S1(1,3),12)
        TcM3S1(M3S1(1,3),k+I(M3S1(1,1),12)+ I(M3S1(1,2),12)) =
T(M3S1(1,3),1) * I(M3S1(1,3),k);
    end
    if k <=I(M3S1(1,4),12)

TcM3S1(M3S1(1,4),k+I(M3S1(1,1),12)+I(M3S1(1,2),12)+I(M3S1(1,3),12)
)= T(M3S1(1,4),1) * I(M3S1(1,4),k);
    end
end

% penambahan waktu untuk pergantian jig
for t = 1:12
    for tt = 2:44
        if TcM1S1(t,tt)~=0 && TcM1S1(t,tt-1)==0
            TcM1S1(t,tt) = 60 + TcM1S1(t,tt);
        end
    end
end

for t = 1:12
    for tt = 2:44
        if TcM2S1(t,tt)~=0 && TcM2S1(t,tt-1)==0
            TcM2S1(t,tt) = 60 + TcM2S1(t,tt);
        end
    end
end

for t = 1:12
    for tt = 2:44
        if TcM3S1(t,tt)~=0 && TcM3S1(t,tt-1)==0
            TcM3S1(t,tt) = 60 + TcM3S1(t,tt);
        end
    end
end

% pengakumulasian masing2 mesin pada stage 1
% Mesin 1
    for tt = 2:44

```

```

        for t = 1:12
            if TcM1S1(t,tt)~=0
                TcM1S1(t,tt) = TcM1S1(t,tt) + sum(TcM1S1(:,tt-1));
            end
        end
    end
end
% Mesin 2
for tt = 2:44
    for t = 1:12
        if TcM2S1(t,tt)~=0
            TcM2S1(t,tt) = TcM2S1(t,tt) + sum(TcM2S1(:,tt-1));
        end
    end
end
% Mesin 3
for tt = 2:44
    for t = 1:12
        if TcM3S1(t,tt)~=0
            TcM3S1(t,tt) = TcM3S1(t,tt) + sum(TcM3S1(:,tt-1));
        end
    end
end

% penyusunan semua pekerjaan
B = zeros(12,44);
B = B + TcM1S1 + TcM2S1 + TcM3S1;
P = B;
B = reshape(B,528,1);
B = sort(B);

xx = find(B~=0);
for tt = 1:numel(xx)
    BB(tt,1) = B(xx(tt,1),1);
end
is = zeros(12,5);
for t = 1:numel(xx)
    for r = 1:12
        IS(t,1) = ismember(BB(t,1),P(r,:));
        if IS(t,1) ==1
            is(t,1)=r;
        end
    end
end
end

% Standard waktu stage 2
t1_2 = 552;
t2_2 = 376;
t3_2 = 689;
t4_2 = 592;
t5_2 = 627;
t6_2 = 356;
t7_2 = 753;
t8_2 = 196;

```

```

t9_2 = 588;
t10_2 = 595;
t11_2 = 639;
t12_2 = 606;

% mengisi waktu
waktu_2 =
[t1_2;t2_2;t3_2;t4_2;t5_2;t6_2;t7_2;t8_2;t9_2;t10_2;t11_2;t12_2];

% mencari nilai waktu
t1a = find(is(:,1) ==1);
t2a = find(is(:,1)==2);
t3a = find(is(:,1)==3);
t4a = find(is(:,1)==4);
t5a = find(is(:,1)==5);
t6a = find(is(:,1)==6);
t7a = find(is(:,1)==7);
t8a = find(is(:,1) ==8);
t9a = find(is(:,1)==9);
t10a = find(is(:,1)==10);
t11a = find(is(:,1)==11);
t12a = find(is(:,1)==12);

% mencari waktu pada job i 1
T1a = numel(t1a);
t1P = find(P(1,:)~=0);
k = 0;
for x = 1:T1a
    k = k+1;
    is(t1a(x,1),2) = P(1,t1P(1,k));
end

% mencari waktu pada job i 2
T2a = numel(t2a);
t2P = find(P(2,:)~=0);
k = 0;
for x = 1:T2a
    k = k+1;
    is(t2a(x,1),2) = P(2,t2P(1,k));
end

% mencari waktu pada job i 3
T3a = numel(t3a);
t3P = find(P(3,:)~=0);
k = 0;
for x = 1:T3a
    k = k+1;
    is(t3a(x,1),2) = P(3,t3P(1,k));
end

% mencari waktu pada job i 4
T4a = numel(t4a);
t4P = find(P(4,:)~=0);
k = 0;
for x = 1:T4a
    k = k+1;

```

```

        is(t4a(x,1),2) = P(4,t4P(1,k));
    end

    % mencari waktu pada job i 5
    T5a = numel(t5a);
    t5P = find(P(5,:)~=0);
    k = 0;
    for x = 1:T5a
        k = k+1;
        is(t5a(x,1),2) = P(5,t5P(1,k));
    end

    % mencari waktu pada job i 6
    T6a = numel(t6a);
    t6P = find(P(6,:)~=0);
    k = 0;
    for x = 1:T6a
        k = k+1;
        is(t6a(x,1),2) = P(6,t6P(1,k));
    end

    % mencari waktu pada job i 7
    T7a = numel(t7a);
    t7P = find(P(7,:)~=0);
    k = 0;
    for x = 1:T7a
        k = k+1;
        is(t7a(x,1),2) = P(7,t7P(1,k));
    end

    % mencari waktu pada job i 8
    T8a = numel(t8a);
    t8P = find(P(8,:)~=0);
    k = 0;
    for x = 1:T8a
        k = k+1;
        is(t8a(x,1),2) = P(8,t8P(1,k));
    end

    % mencari waktu pada job i 9
    T9a = numel(t9a);
    t9P = find(P(9,:)~=0);
    k = 0;
    for x = 1:T9a
        k = k+1;
        is(t9a(x,1),2) = P(9,t9P(1,k));
    end

    % mencari waktu pada job i 10
    T10a = numel(t10a);
    t10P = find(P(10,:)~=0);
    k = 0;
    for x = 1:T10a
        k = k+1;
        is(t10a(x,1),2) = P(10,t10P(1,k));
    end
end

```

```

% mencari waktu pada job i 11
T11a = numel(t11a);
t11P = find(P(11,:)~=0);
k = 0;
for x = 1:T11a
    k = k+1;
    is(t11a(x,1),2) = P(11,t11P(1,k));
end

% mencari waktu pada job i 12
T12a = numel(t12a);
t12P = find(P(12,:)~=0);
k = 0;
for x = 1:T12a
    k = k+1;
    is(t12a(x,1),2) = P(12,t12P(1,k));
end

pause

% memulai stage 2
MS2 = zeros(5,18);
MS2_ranking = zeros(5,18);
MS2(1:5,1) = is(1:5,1);

% perankingan step pertama
for c = 1:5
    if is(c,1) ==1
        is(c,3) = is(c,2) + t1_2;
    end
    if is(c,1) ==2
        is(c,3) = is(c,2) + t2_2;
    end
    if is(c,1) ==3
        is(c,3) = is(c,2) + t3_2;
    end
    if is(c,1) ==4
        is(c,3) = is(c,2) + t4_2;
    end
    if is(c,1) ==5
        is(c,3) = is(c,2) + t5_2;
    end
    if is(c,1) ==6
        is(c,3) = is(c,2) + t6_2;
    end
    if is(c,1) ==7
        is(c,3) = is(c,2) + t7_2;
    end
    if is(c,1) ==8
        is(c,3) = is(c,2) + t8_2;
    end
    if is(c,1) ==9
        is(c,3) = is(c,2) + t9_2;
    end
    if is(c,1) ==10
        is(c,3) = is(c,2) + t10_2;
    end
end

```

```

    if is(c,1) ==11
        is(c,3) = is(c,2) + t11_2;
    end
    if is(c,1) ==12
        is(c,3) = is(c,2) + t12_2;
    end
end

% inisialisasi urutan
is(1,4) =1;
is(2,4) =2;
is(3,4) =3;
is(4,4) =4;
is(5,4) = 5;

for v = 6: 86
    if is(v,1) ==1
        is(v,2) = waktu_2(1,1);
    end
    if is(v,1) ==2
        is(v,2) = waktu_2(2,1);
    end
    if is(v,1) ==3
        is(v,2) = waktu_2(3,1);
    end
    if is(v,1) ==4
        is(v,2) = waktu_2(4,1);
    end
    if is(v,1) ==5
        is(v,2) = waktu_2(5,1);
    end
    if is(v,1) ==6
        is(v,2) = waktu_2(6,1);
    end
    if is(v,1) ==7
        is(v,2) = waktu_2(7,1);
    end
    if is(v,1) ==8
        is(v,2) = waktu_2(8,1);
    end
    if is(v,1) ==9
        is(v,2) = waktu_2(9,1);
    end
    if is(v,1) ==10
        is(v,2) = waktu_2(10,1);
    end
    if is(v,1) ==11
        is(v,2) = waktu_2(11,1);
    end
    if is(v,1) ==12
        is(v,2) = waktu_2(12,1);
    end
end

% ranking urutan
% membuat matrix ranking
ranking = zeros(86,1);
ranking(1:5,1) = is(1:5,3);

```



```

ranking(1:5,1) = sort(ranking(1:5,1));

for t =1:5
    if is(t,3) ==ranking(1,1)
        is(t,5) =1;
    end
    if is(t,3) ==ranking(2,1)
        is(t,5) =2;
    end
    if is(t,3) ==ranking(3,1)
        is(t,5) =3;
    end
    if is(t,3) ==ranking(4,1)
        is(t,5) =4;
    end
    if is(t,3) ==ranking(5,1)
        is(t,5) =5;
    end
end

end

% urut 2
% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t,5)==1
        A = find(is(1:5,5)==1);
        is(is(t,5)+5,4) = is(A,4);
        is(is(t,5)+5,3) = is(A,3);
    end
    if is(t,5)==2
        A = find(is(1:5,5)==2);
        is(is(t,5)+5,4) = is(A,4);
        is(is(t,5)+5,3) = is(A,3);
    end
    if is(t,5)==3
        A = find(is(1:5,5)==3);
        is(is(t,5)+5,4) = is(A,4);
        is(is(t,5)+5,3) = is(A,3);
    end
    if is(t,5)==4
        A = find(is(1:5,5)==4);
        is(is(t,5)+5,4) = is(A,4);
        is(is(t,5)+5,3) = is(A,3);
    end
    if is(t,5)==5
        A = find(is(1:5,5)==5);
        is(is(t,5)+5,4) = is(A,4);
        is(is(t,5)+5,3) = is(A,3);
    end
end

end

% mengisi matrix waktu urut 2
for t =1:5
    is(t+5,3) = is(t+5,3)+is(t+5,2) ;
end
end

```

```

    % ranking utkurut 3
    % membuat matrix ranking
    ranking(6:10,1) = is(6:10,3);
    ranking(6:10,1) = sort(ranking(6:10,1));

    for t =1:5
        if is(t+5,3) ==ranking(6,1)
            is(t+5,5) =1;
        end
        if is(t+5,3) ==ranking(7,1)
            is(t+5,5) =2;
        end
        if is(t+5,3) ==ranking(8,1)
            is(t+5,5) =3;
        end
        if is(t+5,3) ==ranking(9,1)
            is(t+5,5) =4;
        end
        if is(t+5,3) ==ranking(10,1)
            is(t+5,5) =5;
        end
    end

    end

    % urut 3 mulai dari sini

    % mengisi urutan mesin plus waktunya
    for t =1:5
        if is(t+5,5)==1
            A = find(is(6:10,5)==1);
            is(is(t+5,5)+10,4) = is(A+5,4);
            is(is(t+5,5)+10,3) = is(A+5,3);
        end
        if is(t+5,5)==2
            A = find(is(6:10,5)==2);
            is(is(t+5,5)+10,4) = is(A+5,4);
            is(is(t+5,5)+10,3) = is(A+5,3);
        end
        if is(t+5,5)==3
            A = find(is(6:10,5)==3);
            is(is(t+5,5)+10,4) = is(A+5,4);
            is(is(t+5,5)+10,3) = is(A+5,3);
        end
        if is(t+5,5)==4
            A = find(is(6:10,5)==4);
            is(is(t+5,5)+10,4) = is(A+5,4);
            is(is(t+5,5)+10,3) = is(A+5,3);
        end
        if is(t+5,5)==5
            A = find(is(6:10,5)==5);
            is(is(t+5,5)+10,4) = is(A+5,4);
            is(is(t+5,5)+10,3) = is(A+5,3);
        end
    end

    end

    % mengisi matrix waktu urut 3
    for t =1:5

```

```

        is(t+10,3) = is(t+10,3)+is(t+10,2) ;
    end

    % ranking utk urut 4
    % membuat matrix ranking
    ranking(11:15,1) = is(11:15,3);
    ranking(11:15,1) = sort(ranking(11:15,1));

for t =1:5
    if is(t+10,3) ==ranking(11,1)
        is(t+10,5) =1;
    end
    if is(t+10,3) ==ranking(12,1)
        is(t+10,5) =2;
    end
    if is(t+10,3) ==ranking(13,1)
        is(t+10,5) =3;
    end
    if is(t+10,3) ==ranking(14,1)
        is(t+10,5) =4;
    end
    if is(t+10,3) ==ranking(15,1)
        is(t+10,5) =5;
    end
end

% urut 4 mulai dari sini
% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t+10,5)==1
        A = find(is(11:15,5)==1);
        is(is(t+10,5)+15,4) = is(A+10,4);
        is(is(t+10,5)+15,3 )= is(A+10,3);
    end
    if is(t+10,5)==2
        A = find(is(11:15,5)==2);
        is(is(t+10,5)+15,4) = is(A+10,4);
        is(is(t+10,5)+15,3 )= is(A+10,3);
    end
    if is(t+10,5)==3
        A = find(is(11:15,5)==3);
        is(is(t+10,5)+15,4) = is(A+10,4);
        is(is(t+10,5)+15,3 )= is(A+10,3);
    end
    if is(t+10,5)==4
        A = find(is(11:15,5)==4);
        is(is(t+10,5)+15,4) = is(A+10,4);
        is(is(t+10,5)+15,3 )= is(A+10,3);
    end
    if is(t+10,5)==5
        A = find(is(11:15,5)==5);
        is(is(t+10,5)+15,4) = is(A+10,4);
        is(is(t+10,5)+15,3 )= is(A+10,3);
    end
end
end

```

```

% mengisi matrix waktu urut 4
for t =1:5
    is(t+15,3) = is(t+15,3)+is(t+15,2) ;
end

% ranking utk urut 5
% membuat matrix ranking
ranking(16:20,1) = is(16:20,3);
ranking(16:20,1) = sort(ranking(16:20,1));

for t =1:5
    if is(t+15,3) ==ranking(16,1)
        is(t+15,5) =1;
    end
    if is(t+15,3) ==ranking(17,1)
        is(t+15,5) =2;
    end
    if is(t+15,3) ==ranking(18,1)
        is(t+15,5) =3;
    end
    if is(t+15,3) ==ranking(19,1)
        is(t+15,5) =4;
    end
    if is(t+15,3) ==ranking(20,1)
        is(t+15,5) =5;
    end
end

% urut 5 mulai

% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t+15,5)==1
        A = find(is(16:20,5)==1);
        is(is(t+15,5)+20,4) = is(A+15,4);
        is(is(t+15,5)+20,3) = is(A+15,3);
    end
    if is(t+15,5)==2
        A = find(is(16:20,5)==2);
        is(is(t+15,5)+20,4) = is(A+15,4);
        is(is(t+15,5)+20,3) = is(A+15,3);
    end
    if is(t+15,5)==3
        A = find(is(16:20,5)==3);
        is(is(t+15,5)+20,4) = is(A+15,4);
        is(is(t+15,5)+20,3) = is(A+15,3);
    end
    if is(t+15,5)==4
        A = find(is(16:20,5)==4);
        is(is(t+15,5)+20,4) = is(A+15,4);
        is(is(t+15,5)+20,3) = is(A+15,3);
    end
    if is(t+15,5)==5
        A = find(is(16:20,5)==5);
        is(is(t+15,5)+20,4) = is(A+15,4);
        is(is(t+15,5)+20,3) = is(A+15,3);
    end
end

```

```

end

% mengisi matrix waktu urut 5
for t =1:5
    is(t+20,3) = is(t+20,3)+is(t+20,2) ;
end

% ranking utk urut 6
% membuat matrix ranking
ranking(21:25,1) = is(21:25,3);
ranking(21:25,1) = sort(ranking(21:25,1));

for t =1:5
    if is(t+20,3) ==ranking(21,1)
        is(t+20,5) =1;
    end
    if is(t+20,3) ==ranking(22,1)
        is(t+20,5) =2;
    end
    if is(t+20,3) ==ranking(23,1)
        is(t+20,5) =3;
    end
    if is(t+20,3) ==ranking(24,1)
        is(t+20,5) =4;
    end
    if is(t+20,3) ==ranking(25,1)
        is(t+20,5) =5;
    end
end

% urut 6 mulai dari sini

% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t+20,5)==1
        A = find(is(21:25,5)==1);
        is(is(t+20,5)+25,4) = is(A+20,4);
        is(is(t+20,5)+25,3) = is(A+20,3);
    end
    if is(t+20,5)==2
        A = find(is(21:25,5)==2);
        is(is(t+20,5)+25,4) = is(A+20,4);
        is(is(t+20,5)+25,3) = is(A+20,3);
    end
    if is(t+20,5)==3
        A = find(is(21:25,5)==3);
        is(is(t+20,5)+25,4) = is(A+20,4);
        is(is(t+20,5)+25,3) = is(A+20,3);
    end
    if is(t+20,5)==4
        A = find(is(21:25,5)==4);
        is(is(t+20,5)+25,4) = is(A+20,4);
        is(is(t+20,5)+25,3) = is(A+20,3);
    end
    if is(t+20,5)==5
        A = find(is(21:25,5)==5);
        is(is(t+20,5)+25,4) = is(A+20,4);
    end
end

```

```

    is(is(t+20,5)+25,3 )= is(A+20,3);
end
end

% mengisi matrix waktu urut 6
for t =1:5
    is(t+25,3) = is(t+25,3)+is(t+25,2);
end

% ranking utk urut 7
% membuat matrix ranking
ranking(26:30,1) = is(26:30,3);
ranking(26:30,1) = sort(ranking(26:30,1));

for t =1:5
    if is(t+25,3) ==ranking(26,1)
        is(t+25,5) =1;
    end
    if is(t+25,3) ==ranking(27,1)
        is(t+25,5) =2;
    end
    if is(t+25,3) ==ranking(28,1)
        is(t+25,5) =3;
    end
    if is(t+25,3) ==ranking(29,1)
        is(t+25,5) =4;
    end
    if is(t+25,3) ==ranking(30,1)
        is(t+25,5) =5;
    end
end

end

% urut 7 mulai dari sini

% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t+25,5)==1
        A = find(is(26:30,5)==1);
        is(is(t+25,5)+30,4) = is(A+25,4);
        is(is(t+25,5)+30,3 )= is(A+25,3);
    end
    if is(t+25,5)==2
        A = find(is(26:30,5)==2);
        is(is(t+25,5)+30,4) = is(A+25,4);
        is(is(t+25,5)+30,3 )= is(A+25,3);
    end
    if is(t+25,5)==3
        A = find(is(26:30,5)==3);
        is(is(t+25,5)+30,4) = is(A+25,4);
        is(is(t+25,5)+30,3 )= is(A+25,3);
    end
    if is(t+25,5)==4
        A = find(is(26:30,5)==4);
        is(is(t+25,5)+30,4) = is(A+25,4);
        is(is(t+25,5)+30,3 )= is(A+25,3);
    end
end

```

```

end
if is(t+25,5)==5
A = find(is(26:30,5)==5);
is(is(t+25,5)+30,4) = is(A+25,4);
is(is(t+25,5)+30,3) = is(A+25,3);
end
end

% mengisi matrix waktu urut 7
for t =1:5
    is(t+30,3) = is(t+30,3)+is(t+30,2) ;
end

% ranking utk urut 8
% membuat matrix ranking
ranking(31:35,1) = is(31:35,3);
ranking(31:35,1) = sort(ranking(31:35,1));

for t =1:5
    if is(t+30,3) ==ranking(31,1)
        is(t+30,5) =1;
    end
    if is(t+30,3) ==ranking(32,1)
        is(t+30,5) =2;
    end
    if is(t+30,3) ==ranking(33,1)
        is(t+30,5) =3;
    end
    if is(t+30,3) ==ranking(34,1)
        is(t+30,5) =4;
    end
    if is(t+30,3) ==ranking(35,1)
        is(t+30,5) =5;
    end
end

end

% urut 8 mulai dari sini

% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t+30,5)==1
        A = find(is(31:35,5)==1);
        is(is(t+30,5)+35,4) = is(A+30,4);
        is(is(t+30,5)+35,3) = is(A+30,3);
    end
    if is(t+30,5)==2
        A = find(is(31:35,5)==2);
        is(is(t+30,5)+35,4) = is(A+30,4);
        is(is(t+30,5)+35,3) = is(A+30,3);
    end
    if is(t+30,5)==3
        A = find(is(31:35,5)==3);
        is(is(t+30,5)+35,4) = is(A+30,4);
        is(is(t+30,5)+35,3) = is(A+30,3);
    end
    if is(t+30,5)==4
        A = find(is(31:35,5)==4);

```

```

is(is(t+30,5)+35,4) = is(A+30,4);
is(is(t+30,5)+35,3) = is(A+30,3);
end
if is(t+30,5)==5
A = find(is(31:35,5)==5);
is(is(t+30,5)+35,4) = is(A+30,4);
is(is(t+30,5)+35,3) = is(A+30,3);
end
end

% mengisi matrix waktu urut 8
for t =1:5
is(t+35,3) = is(t+35,3)+is(t+35,2) ;
end

% ranking utk urut 9
% membuat matrix ranking
ranking(36:40,1) = is(36:40,3);
ranking(36:40,1) = sort(ranking(36:40,1));

for t =1:5
if is(t+35,3) ==ranking(36,1)
is(t+35,5) =1;
end
if is(t+35,3) ==ranking(37,1)
is(t+35,5) =2;
end
if is(t+35,3) ==ranking(38,1)
is(t+35,5) =3;
end
if is(t+35,3) ==ranking(39,1)
is(t+35,5) =4;
end
if is(t+35,3) ==ranking(40,1)
is(t+35,5) =5;
end
end

% urut 9 mulai dari sini

% mengisi urutan mesin plus waktunya
for t =1:5
if is(t+35,5)==1
A = find(is(36:40,5)==1);
is(is(t+35,5)+40,4) = is(A+35,4);
is(is(t+35,5)+40,3) = is(A+35,3);
end
if is(t+35,5)==2
A = find(is(36:40,5)==2);
is(is(t+35,5)+40,4) = is(A+35,4);
is(is(t+35,5)+40,3) = is(A+35,3);
end
if is(t+35,5)==3
A = find(is(36:40,5)==3);
is(is(t+35,5)+40,4) = is(A+35,4);
is(is(t+35,5)+40,3) = is(A+35,3);
end
end

```



```

    if is(t+35,5)==4
    A = find(is(36:40,5)==4);
    is(is(t+35,5)+40,4) = is(A+35,4);
    is(is(t+35,5)+40,3) = is(A+35,3);
    end
    if is(t+35,5)==5
    A = find(is(36:40,5)==5);
    is(is(t+35,5)+40,4) = is(A+35,4);
    is(is(t+35,5)+40,3) = is(A+35,3);
    end
end

% mengisi matrix waktu urut 9
for t =1:5
    is(t+40,3) = is(t+40,3)+is(t+40,2) ;
end

% ranking utk urut 10
% membuat matrix ranking
ranking(41:45,1) = is(41:45,3);
ranking(41:45,1) = sort(ranking(41:45,1));

for t =1:5
    if is(t+40,3) ==ranking(41,1)
        is(t+40,5) =1;
    end
    if is(t+40,3) ==ranking(42,1)
        is(t+40,5) =2;
    end
    if is(t+40,3) ==ranking(43,1)
        is(t+40,5) =3;
    end
    if is(t+40,3) ==ranking(44,1)
        is(t+40,5) =4;
    end
    if is(t+40,3) ==ranking(45,1)
        is(t+40,5) =5;
    end
end

end

% urut 10

% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t+40,5)==1
    A = find(is(41:45,5)==1);
    is(is(t+40,5)+45,4) = is(A+40,4);
    is(is(t+40,5)+45,3) = is(A+40,3);
    end
    if is(t+40,5)==2
    A = find(is(41:45,5)==2);
    is(is(t+40,5)+45,4) = is(A+40,4);
    is(is(t+40,5)+45,3) = is(A+40,3);
    end
    if is(t+40,5)==3
    A = find(is(41:45,5)==3);
    is(is(t+40,5)+45,4) = is(A+40,4);

```

```

is(is(t+40,5)+45,3) = is(A+40,3);
end
if is(t+40,5)==4
A = find(is(41:45,5)==4);
is(is(t+40,5)+45,4) = is(A+40,4);
is(is(t+40,5)+45,3) = is(A+40,3);
end
if is(t+40,5)==5
A = find(is(41:45,5)==5);
is(is(t+40,5)+45,4) = is(A+40,4);
is(is(t+40,5)+45,3) = is(A+40,3);
end
end

% mengisi matrix waktu urut 10
for t =1:5
is(t+45,3) = is(t+45,3)+is(t+45,2);
end

% ranking utk urut 11
% membuat matrix ranking
ranking(46:50,1) = is(46:50,3);
ranking(46:50,1) = sort(ranking(46:50,1));

for t =1:5
if is(t+45,3) ==ranking(46,1)
is(t+45,5) =1;
end
if is(t+45,3) ==ranking(47,1)
is(t+45,5) =2;
end
if is(t+45,3) ==ranking(48,1)
is(t+45,5) =3;
end
if is(t+45,3) ==ranking(49,1)
is(t+45,5) =4;
end
if is(t+45,3) ==ranking(50,1)
is(t+45,5) =5;
end

end

% urut 11

% mengisi urutan mesin plus waktunya
for t =1:5
if is(t+45,5)==1
A = find(is(46:50,5)==1);
is(is(t+45,5)+50,4) = is(A+45,4);
is(is(t+45,5)+50,3) = is(A+45,3);
end
if is(t+45,5)==2
A = find(is(46:50,5)==2);
is(is(t+45,5)+50,4) = is(A+45,4);
is(is(t+45,5)+50,3) = is(A+45,3);
end

```

```

end
if is(t+45,5)==3
A = find(is(46:50,5)==3);
is(is(t+45,5)+50,4) = is(A+45,4);
is(is(t+45,5)+50,3) = is(A+45,3);
end
if is(t+45,5)==4
A = find(is(46:50,5)==4);
is(is(t+45,5)+50,4) = is(A+45,4);
is(is(t+45,5)+50,3) = is(A+45,3);
end
if is(t+45,5)==5
A = find(is(46:50,5)==5);
is(is(t+45,5)+50,4) = is(A+45,4);
is(is(t+45,5)+50,3) = is(A+45,3);
end
end

% mengisi matrix waktu urut 11
for t =1:5
    is(t+50,3) = is(t+50,3)+is(t+50,2) ;
end

% ranking utk urut 12
% membuat matrix ranking
ranking(51:55,1) = is(51:55,3);
ranking(51:55,1) = sort(ranking(51:55,1));

for t =1:5
    if is(t+50,3) ==ranking(51,1)
        is(t+50,5) =1;
    end
    if is(t+50,3) ==ranking(52,1)
        is(t+50,5) =2;
    end
    if is(t+50,3) ==ranking(53,1)
        is(t+50,5) =3;
    end
    if is(t+50,3) ==ranking(54,1)
        is(t+50,5) =4;
    end
    if is(t+50,3) ==ranking(55,1)
        is(t+50,5) =5;
    end
end

end

% urut 12

% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t+50,5)==1
        A = find(is(51:55,5)==1);
        is(is(t+50,5)+55,4) = is(A+50,4);
        is(is(t+50,5)+55,3) = is(A+50,3);
    end
    if is(t+50,5)==2
        A = find(is(51:55,5)==2);

```

```

is(is(t+50,5)+55,4) = is(A+50,4);
is(is(t+50,5)+55,3) = is(A+50,3);
end
if is(t+50,5)==3
A = find(is(51:55,5)==3);
is(is(t+50,5)+55,4) = is(A+50,4);
is(is(t+50,5)+55,3) = is(A+50,3);
end
if is(t+50,5)==4
A = find(is(51:55,5)==4);
is(is(t+50,5)+55,4) = is(A+50,4);
is(is(t+50,5)+55,3) = is(A+50,3);
end
if is(t+50,5)==5
A = find(is(51:55,5)==5);
is(is(t+50,5)+55,4) = is(A+50,4);
is(is(t+50,5)+55,3) = is(A+50,3);
end
end

% mengisi matrix waktu urut 12
for t =1:5
is(t+55,3) = is(t+55,3)+is(t+55,2);
end

% ranking utk urut 12
% membuat matrix ranking
ranking(56:60,1) = is(56:60,3);
ranking(56:60,1) = sort(ranking(56:60,1));

for t =1:5
if is(t+55,3) ==ranking(56,1)
is(t+55,5) =1;
end
if is(t+55,3) ==ranking(57,1)
is(t+55,5) =2;
end
if is(t+55,3) ==ranking(58,1)
is(t+55,5) =3;
end
if is(t+55,3) ==ranking(59,1)
is(t+55,5) =4;
end
if is(t+55,3) ==ranking(60,1)
is(t+55,5) =5;
end

end

% urut 13

% mengisi urutan mesin plus waktunya
for t =1:5
if is(t+55,5)==1
A = find(is(56:60,5)==1);
is(is(t+55,5)+60,4) = is(A+55,4);
is(is(t+55,5)+60,3) = is(A+55,3);
end

```

```

    if is(t+55,5)==2
    A = find(is(56:60,5)==2);
    is(is(t+55,5)+60,4) = is(A+55,4);
    is(is(t+55,5)+60,3) = is(A+55,3);
    end
    if is(t+55,5)==3
    A = find(is(56:60,5)==3);
    is(is(t+55,5)+60,4) = is(A+55,4);
    is(is(t+55,5)+60,3) = is(A+55,3);
    end
    if is(t+55,5)==4
    A = find(is(56:60,5)==4);
    is(is(t+55,5)+60,4) = is(A+55,4);
    is(is(t+55,5)+60,3) = is(A+55,3);
    end
    if is(t+55,5)==5
    A = find(is(56:60,5)==5);
    is(is(t+55,5)+60,4) = is(A+55,4);
    is(is(t+55,5)+60,3) = is(A+55,3);
    end
end

% mengisi matrix waktu urut 13
for t =1:5
    is(t+60,3) = is(t+60,3)+is(t+60,2) ;
end

% ranking utk urut 14
% membuat matrix ranking
ranking(61:65,1) = is(61:65,3);
ranking(61:65,1) = sort(ranking(61:65,1));

for t =1:5
    if is(t+60,3) ==ranking(61,1)
        is(t+60,5) =1;
    end
    if is(t+60,3) ==ranking(62,1)
        is(t+60,5) =2;
    end
    if is(t+60,3) ==ranking(63,1)
        is(t+60,5) =3;
    end
    if is(t+60,3) ==ranking(64,1)
        is(t+60,5) =4;
    end
    if is(t+60,3) ==ranking(65,1)
        is(t+60,5) =5;
    end
end

end

% urut 14

% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t+60,5)==1
    A = find(is(61:65,5)==1);
    is(is(t+60,5)+65,4) = is(A+60,4);

```

```

is(is(t+60,5)+65,3 )= is(A+60,3);
end
if is(t+60,5)==2
A = find(is(61:65,5)==2);
is(is(t+60,5)+65,4) = is(A+60,4);
is(is(t+60,5)+65,3 )= is(A+60,3);
end
if is(t+60,5)==3
A = find(is(61:65,5)==3);
is(is(t+60,5)+65,4) = is(A+60,4);
is(is(t+60,5)+65,3 )= is(A+60,3);
end
if is(t+60,5)==4
A = find(is(61:65,5)==4);
is(is(t+60,5)+65,4) = is(A+60,4);
is(is(t+60,5)+65,3 )= is(A+60,3);
end
if is(t+60,5)==5
A = find(is(61:65,5)==5);
is(is(t+60,5)+65,4) = is(A+60,4);
is(is(t+60,5)+65,3 )= is(A+60,3);
end
end

% mengisi matrix waktu urut 14
for t =1:5
    is(t+65,3) = is(t+65,3)+is(t+65,2) ;
end

% ranking utk urut 15
% membuat matrix ranking
ranking(66:70,1) = is(66:70,3);
ranking(66:70,1) = sort(ranking(66:70,1));

for t =1:5
    if is(t+65,3) ==ranking(66,1)
        is(t+65,5) =1;
    end
    if is(t+65,3) ==ranking(67,1)
        is(t+65,5) =2;
    end
    if is(t+65,3) ==ranking(68,1)
        is(t+65,5) =3;
    end
    if is(t+65,3) ==ranking(69,1)
        is(t+65,5) =4;
    end
    if is(t+65,3) ==ranking(70,1)
        is(t+65,5) =5;
    end
end

end

% urut 15

% mengisi urutan mesin plus waktunya

```

```

for t =1:5
    if is(t+65,5)==1
        A = find(is(66:70,5)==1);
        is(is(t+65,5)+70,4) = is(A+65,4);
        is(is(t+65,5)+70,3) = is(A+65,3);
    end
    if is(t+65,5)==2
        A = find(is(66:70,5)==2);
        is(is(t+65,5)+70,4) = is(A+65,4);
        is(is(t+65,5)+70,3) = is(A+65,3);
    end
    if is(t+65,5)==3
        A = find(is(66:70,5)==3);
        is(is(t+65,5)+70,4) = is(A+65,4);
        is(is(t+65,5)+70,3) = is(A+65,3);
    end
    if is(t+65,5)==4
        A = find(is(66:70,5)==4);
        is(is(t+65,5)+70,4) = is(A+65,4);
        is(is(t+65,5)+70,3) = is(A+65,3);
    end
    if is(t+65,5)==5
        A = find(is(66:70,5)==5);
        is(is(t+65,5)+70,4) = is(A+65,4);
        is(is(t+65,5)+70,3) = is(A+65,3);
    end
end

% mengisi matrix waktuurut 15
for t =1:5
    is(t+70,3) = is(t+70,3)+is(t+70,2);
end

% ranking utk urut 16
% membuat matrix ranking
ranking(71:75,1) = is(71:75,3);
ranking(71:75,1) = sort(ranking(71:75,1));

for t =1:5
    if is(t+70,3) ==ranking(71,1)
        is(t+70,5) =1;
    end
    if is(t+70,3) ==ranking(72,1)
        is(t+70,5) =2;
    end
    if is(t+70,3) ==ranking(73,1)
        is(t+70,5) =3;
    end
    if is(t+70,3) ==ranking(74,1)
        is(t+70,5) =4;
    end
    if is(t+70,3) ==ranking(75,1)
        is(t+70,5) =5;
    end
end

end

```

```

%urut 16

% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t+70,5)==1
        A = find(is(71:75,5)==1);
        is(is(t+70,5)+75,4) = is(A+70,4);
        is(is(t+70,5)+75,3) = is(A+70,3);
    end
    if is(t+70,5)==2
        A = find(is(71:75,5)==2);
        is(is(t+70,5)+75,4) = is(A+70,4);
        is(is(t+70,5)+75,3) = is(A+70,3);
    end
    if is(t+70,5)==3
        A = find(is(71:75,5)==3);
        is(is(t+70,5)+75,4) = is(A+70,4);
        is(is(t+70,5)+75,3) = is(A+70,3);
    end
    if is(t+70,5)==4
        A = find(is(71:75,5)==4);
        is(is(t+70,5)+75,4) = is(A+70,4);
        is(is(t+70,5)+75,3) = is(A+70,3);
    end
    if is(t+70,5)==5
        A = find(is(71:75,5)==5);
        is(is(t+70,5)+75,4) = is(A+70,4);
        is(is(t+70,5)+75,3) = is(A+70,3);
    end
end

% mengisi matrix waktuurut 16
for t =1:5
    is(t+75,3) = is(t+75,3)+is(t+75,2) ;
end

% ranking utk urut 17
% membuat matrix ranking
ranking(76:80,1) = is(76:80,3);
ranking(76:80,1) = sort(ranking(76:80,1));

for t =1:5
    if is(t+75,3) ==ranking(76,1)
        is(t+75,5) =1;
    end
    if is(t+75,3) ==ranking(77,1)
        is(t+75,5) =2;
    end
    if is(t+75,3) ==ranking(78,1)
        is(t+75,5) =3;
    end
    if is(t+75,3) ==ranking(79,1)
        is(t+75,5) =4;
    end
    if is(t+75,3) ==ranking(80,1)
        is(t+75,5) =5;
    end
end

```



```

end

%urut 17

% mengisi urutan mesin plus waktunya
for t =1:5
    if is(t+75,5)==1
        A = find(is(76:80,5)==1);
        is(is(t+75,5)+80,4) = is(A+75,4);
        is(is(t+75,5)+80,3) = is(A+75,3);
    end
    if is(t+75,5)==2
        A = find(is(76:80,5)==2);
        is(is(t+75,5)+80,4) = is(A+75,4);
        is(is(t+75,5)+80,3) = is(A+75,3);
    end
    if is(t+75,5)==3
        A = find(is(76:80,5)==3);
        is(is(t+75,5)+80,4) = is(A+75,4);
        is(is(t+75,5)+80,3) = is(A+75,3);
    end
    if is(t+75,5)==4
        A = find(is(76:80,5)==4);
        is(is(t+75,5)+80,4) = is(A+75,4);
        is(is(t+75,5)+80,3) = is(A+75,3);
    end
    if is(t+75,5)==5
        A = find(is(76:80,5)==5);
        is(is(t+75,5)+80,4) = is(A+75,4);
        is(is(t+75,5)+80,3) = is(A+75,3);
    end
end

% mengisi matrix waktuurut 17
for t =1:5
    is(t+80,3) = is(t+80,3)+is(t+80,2);
end

% ranking utkurut 18
% membuat matrix ranking
ranking(81:85,1) = is(81:85,3);
ranking(81:85,1) = sort(ranking(81:85,1));

for t =1:5
    if is(t+80,3) ==ranking(81,1)
        is(t+80,5) =1;
    end
    if is(t+80,3) ==ranking(82,1)
        is(t+80,5) =2;
    end
    if is(t+80,3) ==ranking(83,1)
        is(t+80,5) =3;
    end
    if is(t+80,3) ==ranking(84,1)
        is(t+80,5) =4;
    end
    if is(t+80,3) ==ranking(85,1)

```

```

        is(t+80,5) =5;
    end

end

%urut 18

% mengisi urutan mesin plus waktunya

    A = find(is(81:85,5)==1);
    is(is(81,5)+85,4) = is(A+80,4);
    is(is(81,5)+85,3) = is(A+80,3);
    is(86,3) = is(86,3)+is(86,2) ;

% urutan stage 2

M1S2 = find(is(:,4)==1);
M2S2 = find(is(:,4)==2);
M3S2 = find(is(:,4)==3);
M4S2 = find(is(:,4)==4);
M5S2 = find(is(:,4)==5);

for y = 1: numel(M1S2)
    DeretMesin1_2(y) = is(M1S2(y),1);
    WaktuDeretMesin1_2(y) = is(M1S2(y),3);
end

for y = 1: numel(M2S2)
    DeretMesin2_2(y) = is(M2S2(y),1);
    WaktuDeretMesin2_2(y) = is(M2S2(y),3);
end

for y = 1: numel(M3S2)
    DeretMesin3_2(y) = is(M3S2(y),1);
    WaktuDeretMesin3_2(y) = is(M3S2(y),3);
end

for y = 1: numel(M4S2)
    DeretMesin4_2(y) = is(M4S2(y),1);
    WaktuDeretMesin4_2(y) = is(M4S2(y),3);
end

for y = 1: numel(M5S2)
    DeretMesin5_2(y) = is(M5S2(y),1);
    WaktuDeretMesin5_2(y) = is(M5S2(y),3);
end

% tinggal di plot dech stage 2 nya

DeretMesin1_2
DeretMesin2_2

```

```

DeretMesin3_2
DeretMesin4_2
DeretMesin5_2
pause

% memasuki stage 3
% family job i
i1_3 = find(is (:,1)==1);
for a =1:numel(i1_3)
    t1_3(a,1) =is(i1_3(a,1),3);
end

i2_3 = find(is (:,1)==2);
for a =1:numel(i2_3)
    t2_3(a,1) =is(i2_3(a,1),3);
end

i3_3 = find(is (:,1)==3);
for a =1:numel(i3_3)
    t3_3(a,1) =is(i3_3(a,1),3);
end

i4_3 = find(is (:,1)==4);
for a =1:numel(i4_3)
    t4_3(a,1) =is(i4_3(a,1),3);
end

i5_3 = find(is (:,1)==5);
for a =1:numel(i5_3)
    t5_3(a,1) =is(i5_3(a,1),3);
end

i6_3 = find(is (:,1)==6);
for a =1:numel(i6_3)
    t6_3(a,1) =is(i6_3(a,1),3);
end

i7_3 = find(is (:,1)==7);
for a =1:numel(i7_3)
    t7_3(a,1) =is(i7_3(a,1),3);
end

i8_3 = find(is (:,1)==8);
for a =1:numel(i8_3)
    t8_3(a,1) =is(i8_3(a,1),3);
end

i9_3 = find(is (:,1)==9);
for a =1:numel(i9_3)
    t9_3(a,1) =is(i9_3(a,1),3);
end

```

```

end

i10_3 = find(is(:,1)==10);
for a = 1:numel(i10_3)
    t10_3(a,1) = is(i10_3(a,1),3);
end

i11_3 = find(is(:,1)==11);
for a = 1:numel(i11_3)
    t11_3(a,1) = is(i11_3(a,1),3);
end

i12_3 = find(is(:,1)==12);
for a = 1:numel(i12_3)
    t12_3(a,1) = is(i12_3(a,1),3);
end

% versi terakhir
X1 = find(is(:,1)==1);
X2 = find(is(:,1)==2);
X3 = find(is(:,1)==3);
X4 = find(is(:,1)==4);
X5 = find(is(:,1)==5);
X6 = find(is(:,1)==6);
X7 = find(is(:,1)==7);
X8 = find(is(:,1)==8);
X9 = find(is(:,1)==9);
X10 = find(is(:,1)==10);
X11 = find(is(:,1)==11);
X12 = find(is(:,1)==12);

% Standard waktu stage 3
t1_3 = 300;
t2_3 = 208;
t3_3 = 376;
t4_3 = 328;
t5_3 = 358;
t6_3 = 223;
t7_3 = 442;
t8_3 = 143;
t9_3 = 357;
t10_3 = 367;
t11_3 = 402;
t12_3 = 394;

% mengisi waktu
waktu_3 =
[t1_3;t2_3;t3_3;t4_3;t5_3;t6_3;t7_3;t8_3;t9_3;t10_3;t11_3;t12_3];
pause

% matrix is stage 3
is3 = zeros(12,4);
is3(1,1) = is(X1(1),3);
is3(2,1) = is(X2(1),3);

```

```

is3(3,1) =is(X3(1),3);
is3(4,1) =is(X4(1),3);
is3(5,1) =is(X5(1),3);
is3(6,1) =is(X6(1),3);
is3(7,1) =is(X7(1),3);
is3(8,1) =is(X8(1),3);
is3(9,1) =is(X9(1),3);
is3(10,1) =is(X10(1),3);
is3(11,1) =is(X11(1),3);
is3(12,1) =is(X12(1),3);

pause

% membuat ranking di stage 3
ranking3 = zeros(12,1);
ranking3(1:12,1) = is3(1:12,1);
ranking3(1:12,1) = sort(ranking3(1:12,1));

for t =1:12
    if is3(t,1) == ranking3(1,1)
        is3(t,2) =1;
    end
    if is3(t,1) == ranking3(2,1)
        is3(t,2) =2;
    end
    if is3(t,1) == ranking3(3,1)
        is3(t,2) =3;
    end
    if is3(t,1) == ranking3(4,1)
        is3(t,2) =4;
    end
    if is3(t,1) == ranking3(5,1)
        is3(t,2) =5;
    end
    if is3(t,1) == ranking3(6,1)
        is3(t,2) =6;
    end
    if is3(t,1) == ranking3(7,1)
        is3(t,2) =7;
    end
    if is3(t,1) == ranking3(8,1)
        is3(t,2) =8;
    end
    if is3(t,1) == ranking3(9,1)
        is3(t,2) =9;
    end
    if is3(t,1) == ranking3(10,1)
        is3(t,2) =10;
    end
    if is3(t,1) == ranking3(11,1)
        is3(t,2) =11;
    end
    if is3(t,1) == ranking3(12,1)
        is3(t,2) =12;
    end
end
pause
% mengisi kolom ke 3 mesin

```

```

% inisialisasi
x1 = find(is3(:,2)==1);
is3(x1,3)=1;
x2 = find(is3(:,2)==2);
is3(x2,3)=2;
x3 = find(is3(:,2)==3);
is3(x3,3)=3;
pause
% mengisi kolom 4
% inisialisasi
for t = 1:12
    if is3(t,3)==1
        is3(t,4)=I(t,12)*waktu_3(t,1)+ is3(t,1);
    end
    if is3(t,3)==2
        is3(t,4)=I(t,12)*waktu_3(t,1)+is3(t,1);
    end
    if is3(t,3)==3
        is3(t,4)=I(t,12)*waktu_3(t,1)+is3(t,1);
    end
end
pause
% selanjutnya utkurut ke 2
a = sort(is3(:,4));
p = find(is3(:,4)==a(10));
q = find(is3(:,4)==a(11));
r = find(is3(:,4)==a(12));
pp = find(is3(:,2)==4);
qq = find(is3(:,2)==5);
rr = find(is3(:,2)==6);

is3(pp,4) = is3(p,4)+45 +I(pp,12)*waktu_3(pp,1);
is3(pp,3)=is3(p,3);
is3(qq,4) = is3(q,4)+45 +I(qq,12)*waktu_3(qq,1);
is3(qq,3)=is3(q,3);
is3(rr,4) = is3(r,4)+45 +I(rr,12)*waktu_3(rr,1);
is3(rr,3)=is3(r,3);
pause

% selanjutnya utkurut ke 3
aa = sort(is3(:,4));
pa = find(is3(:,4)==aa(10));
qa = find(is3(:,4)==aa(11));
ra = find(is3(:,4)==aa(12));
pb = find(is3(:,2)==7);
qb = find(is3(:,2)==8);
rb = find(is3(:,2)==9);

is3(pb,4) = is3(pa,4)+45 +I(pb,12)*waktu_3(pb,1);
is3(pb,3)=is3(pa,3);
is3(qb,4) = is3(qa,4)+45 +I(qb,12)*waktu_3(qb,1);
is3(qb,3) = is3(qa,3);
is3(rb,4) = is3(ra,4)+45 +I(rb,12)*waktu_3(rb,1);
is3(rb,3)=is3(ra,3);

pause

```

```

% selanjutnya utk urut ke 4
ab = sort(is3(:,4));
pc = find(is3(:,4)==ab(10));
qc = find(is3(:,4)==ab(11));
rc = find(is3(:,4)==ab(12));
pd = find(is3(:,2)==10);
qd = find(is3(:,2)==11);
rd = find(is3(:,2)==12);

is3(pd,4) = is3(pc,4)+45 +I(pd,12)*waktu_3(pd,1);
is3(pd,3)=is3(pc,3);
is3(qd,4) = is3(qc,4)+45 +I(qd,12)*waktu_3(qd,1);
is3(qd,3)=is3(qc,3);
is3(rd,4) = is3(rc,4)+45 +I(rd,12)*waktu_3(rd,1);
is3(rd,3)=is3(rc,3);
pause
% mencari mesin

M1S3 = find(is3(:,3)==1);
M2S3 = find(is3(:,3)==2);
M3S3 = find(is3(:,3)==3);

for y = 1: numel(M1S3)
    DeretMesin1_3(y) = is3(M1S3(y),2);
    WaktuDeretMesin1_3(y) = is3(M1S3(y),4);
end
for y = 1: numel(M2S3)
    DeretMesin2_3(y) = is3(M2S3(y),2);
    WaktuDeretMesin2_3(y) = is3(M2S3(y),4);
end
for y = 1: numel(M3S3)
    DeretMesin3_3(y) = is3(M3S3(y),2);
    WaktuDeretMesin3_3(y) = is3(M3S3(y),4);
end

end

DeretMesin1_3 =find(is3(:,3)==1)
DeretMesin2_3 =find(is3(:,3)==2)
DeretMesin3_3 =find(is3(:,3)==3)
pause
% menghitung waktu
C_Max = max(is3(:,4))
toc

```