



**UNIVERSITAS INDONESIA**

**OPTIMASI PENJADWALAN *JOB SHOP* DENGAN METODE  
ALGORITMA TABU SEARCH UNTUK MEMINIMUMKAN  
TOTAL WAKTU Pengerjaan SELURUH JOB**

(STUDI KASUS DI POLMAN BANDUNG)

**SKRIPSI**

diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana teknik

**HASANUDIN  
0906603934**


**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK INDUSTRI  
DEPOK  
DESEMBER 2011**

## HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Hasanudin

NPM : 0906603934

TandaTangan : 

Tanggal : 23 Desember 2011

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Hasanudin

NPM : 0906603934

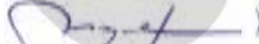
Program Studi : Teknik Industri

Judul Skripsi : Optimasi Penjadwalan *Job Shop* Dengan Metode Algoritma *Tabu Search* Untuk Meminimumkan Total Waktu Pengerjaan Seluruh *Job* (Studi Kasus Di Polman Bandung)

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

### DEWAN PENGUJI

Pembimbing 1 : Ir. Amar Rachman, MEIM.

(  )

Pembimbing 2 : Sumarsono, ST, MT.

(  )

Penguji : Armand Omar Moeis, ST, MSc

(  )

Penguji : Farizal, Ph.D

(  )

Penguji : Romadhani Ardi, ST, MT

(  )

Ditetapkan di : Depok

Tanggal : 29 Desember 2011

## KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT, atas segalanya yang telah memberikan rahmat dan hidayah-NYA dan junjungan Nabi Muhammad SAW, sehingga penulis dapat menyelesaikan Skripsi berjudul **“Optimasi Penjadwalan *Job Shop* dengan Metode Algoritma *Tabu Search* untuk Meminimumkan Total Biaya Keterlambatan (Studi Kasus di POLMAN Bandung)”**.

Penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam penyusunan skripsi ini. Adapun pihak-pihak itu adalah:

1. Bapak dan Ibu yang telah melahirkan dan membesarkan penulis, semoga bahagia selalu dalam sisi Allah SWT.
2. Bapak Ir. Amar Rachman, MEIM, selaku dosen pembimbing 1 skripsi yang telah memberikan waktu, tenaga, pikiran, serta dorongan dalam membimbing penulis.
3. Bapak Sumarsono, ST. MT selaku dosen pembimbing 2 skripsi yang telah memberikan waktu, tenaga, pikiran, serta dorongan dalam membimbing penulis.
4. Bapak Gamawan Ananto Subekti, S.S.T. selaku dosen Polman Bandung atas waktu, tenaga, pikiran serta dorongan kepada penulis dalam pengambilan data.
5. Seluruh staf pengajar dan karyawan dan karyawan di Departemen Teknik Industri, Fakultas Teknik, Universitas Indonesia.
6. Ervan, Wegha, Fuad dan teman-teman serta sahabat Ekstensi TI UI 2009 yang selalu memberikan dukungan, dorongan dan semangat.

Penulis menyadari bahwa Skripsi ini tidak lepas dari kekurangan, maka kritik dan saran sangat penulis harapkan, semoga sebuah karya ini dapat bermanfaat bagi yang membacanya.

Depok, Desember 2011

Penulis

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

---

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

Nama : Hasanudin  
NPM : 0906603934  
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 *Job Shop* Dengan Metode Algoritma *Tabu Search*  
Untuk Meminimumkan Total Biaya Keterlambatan  
(Studi Kasus di Polman Bandung)”**

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 23 Desember 2011

Yang menyatakan

( Hasanudin )

## ABSTRAK

Nama : Hasanudin  
Program Studi : Teknik Industri  
Judul : Optimasi Penjadwalan *Job Shop* Dengan Metode Algoritma *Tabu Search* Untuk Meminimumkan Total Waktu Pengerjaan Seluruh *Job*.  
(Studi Kasus di Polman Bandung)

Penelitian ini akan membahas masalah penjadwalan *Job shop* (*Job shop scheduling problem*). Kerumitan pada masalah penjadwalan *job shop* disebabkan karena pada proses setiap komponen memiliki aliran yang berbeda sehingga dibutuhkan penjadwalan untuk menentukan urutan pengerjaan setiap komponen.

Karena kompleksnya masalah penjadwalan produksi, maka solusi penyelesaian terhadap masalah ini dilakukan dengan menggunakan pendekatan heuristik yaitu metode algoritma *tabu search*. Algoritma *tabu search*, yaitu suatu pendekatan heuristik dalam pencarian solusi berdasarkan pada metode optimasi, dimana algoritma ini menggunakan daftar tabu dan iterasi lokal untuk mencegah terjebak pada *local optimal* hingga tercapainya solusi mendekati terbaik. Pada model *jobshop* penelitian ini terdapat 5 *job* dengan 98 komponen yang di kerjakan di 8 mesin. Fungsi tujuan dari permasalahan ini ialah meminimalkan total waktu pengerjaan seluruh *job*.

Hasil penjadwalan produksi yang diperoleh melalui algoritma *tabu search* setelah 20 iterasi menghasilkan *minimal makespan* seluruh *job* sebesar 197.50 jam. Jadi, jika dibandingkan dengan jadwal produksi yang lama, maka terjadi penurunan *makespan* yaitu sebesar 53,87 %.

Kata kunci:  
Penjadwalan *job shop*; total waktu pengerjaan seluruh *job*; Algoritma *Tabu search*

## ABSTRACT

Name : Hasanudin  
Study Program : Industrial Engineering  
Title : Optimization Job Shop Scheduling Using Tabu Search Algorithm for Minimizing Makespan (Case Studies in Polman Bandung).

This research will present Job shop scheduling problem. The complexity of the job shop scheduling problem is caused the process of each component having different flow process. that it takes to determine sequencing of processing for each component in the scheduling.

Due to the complex problem of production scheduling, then the solution to the problem of settlement is done by using a heuristic approach to taboo search algorithm method. Taboo search algorithm, which is a heuristic search approach based on the solution methods of optimization, where this algorithm uses a local list of taboo and iterations to prevent getting stuck on a local optimum to the achievement of a solution approach the best. In this model there are 5 jobs with 98 components that are in working on the 8 machines. The purpose of this function is to minimize the problems of the total cost of makespan.

Production scheduling results obtained through taboo search algorithm after 200 iterations produces minimal makespan whole job of 197.5 hours. So, when compared to the long production schedule, then decline the makespan of 53.87%.

Key words:

Job shop scheduling; Makespan; Tabu Search Algorithm

## DAFTAR ISI

	<b>Halaman</b>
<b>HALAMAN JUDUL</b> .....	i
<b>HALAMAN PERNYATAAN ORISINALITAS</b> .....	ii
<b>HALAMAN PERSETUJUAN</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH</b> .....	v
<b>ABSTRAK</b> .....	vi
<b>DAFTAR ISI</b> .....	viii
<b>DAFTAR TABEL</b> .....	x
<b>DAFTAR GAMBAR</b> .....	xi
<b>DAFTAR LAMPIRAN</b> .....	xii
<b>1. PENDAHULUAN</b> .....	1
1.1. Latar Belakang .....	1
1.2. Diagram Keterkaitan Masalah .....	3
1.3. Rumusan Masalah .....	3
1.4. Tujuan Penelitian.....	3
1.5. Batasan Masalah.....	4
1.6. Metodologi Penelitian .....	4
1.7. Sistematika Penulisan.....	5
<b>2. LANDASAN TEORI</b> .....	7
2.1. Penjadwalan Produksi.....	7
2.1.1 Pengertian Penjadwalan Produksi .....	7
2.1.2 Tujuan Penjadwalan Produksi.....	7
2.1.3 Klasifikasi penjadwalan Produksi .....	8
2.1.4 Istilah dalam Penjadwalan Produksi.....	9
2.1.5 Karakteristik dan Kendala Produksi.....	9
2.1.6 Fungsi Tujuan dan Pengukuran Performa Penjadwalan Produksi10	
2.2. Penjadwalan <i>Job Shop</i> .....	10
2.3. Metode Penyelesaian Masalah Penjadwalan Produksi .....	11
2.3.1 Tipe Heuristik Klasik.....	12
2.3.2 Tipe Heuristik Modern.....	13
2.4. Algoritma <i>Tabu Search</i> .....	15
2.4.1 Latar Belakang <i>Tabu Search</i> .....	15
2.4.2 Algoritma <i>Tabu Search</i> .....	16
<b>3. PENGUMPULAN DATA</b> .....	19
3.1. Profil Polman Bandung .....	19
3.2. Pengumpulan Data Penelitian.....	19
3.2.1 Data Pesanan dan Jam Kerja.....	19
3.2.2 Rute dan Waktu Operasi.....	21
3.2.3 Formulasi Model Permasalahan.....	22
<b>4. PENGOLAHAN DATA DAN ANALISIS</b> .....	27
4.1. Solusi Awal Penjadwalan Polman Bandung.....	27
4.2. Penyusunan Algoritma .....	30



4.2.1 Langkah-langkah Penyusunan Algoritma <i>Tabu Search</i> .....	31
4.2.2 Verifikasi dan Validasi Program .....	35
4.2.2.1 Hasil Perhitungan Manual.....	36
4.2.2.2 Hasil Perhitungan dengan Piranti Lunak .....	39
4.3. Pengolahan Data dan Analisis.....	40
4.3.1 Input Data.....	40
4.3.2 Parameter.....	41
4.3.3 Hasil Penjadwalan dengan Algoritma <i>Tabu Search</i> .....	41
4.3.4 Metode dan <i>Running</i> Program .....	43
<b>5. KESIMPULAN DAN SARAN</b> .....	<b>45</b>
5.1. Kesimpulan.....	45
5.2. Saran.....	45
<b>DAFTAR PUSTAKA</b> .....	<b>46</b>
<b>LAMPIRAN</b> .....	<b>47</b>



## DAFTAR TABEL

<b>Tabel</b>		<b>Halaman</b>
Tabel 3.1.	Waktu Kerja Polman Bandung .....	20
Tabel 3.2.	Jumlah Mesin yang Digunakan.....	21
Tabel 3.3.	Waktu Operasi Setiap Komponen Dalam Jam.....	22
Tabel 3.4.	Data Waktu Proses Operasi.....	23
Tabel 4.1.	Jadwal Awal Polman Bandung .....	29
Tabel 4.2.	Data <i>Dummy</i> .....	38
Tabel 4.3.	Perhitungan Waktu Penyelesaian Total dari Data <i>Dummy</i> .....	38
Tabel 4.4.	Daftar Tabu .....	38
Tabel 4.5.	Iterasi Global Pertama.....	39
Tabel 4.6.	Rekapitulasi 10 Iterasi Hingga Mendapatkan Solusi Global Akhir	40
Tabel 4.7.	Daftar Tabu Hingga Iterasi Ke-10 .....	41
Tabel 4.8.	Hasil Pengujian Algoritma Dengan Panjang Solusi Tetangga 50... 43	
Tabel 4.9.	Hasil Pengujian Algoritma Dengan Panjang Solusi Tetangga 100..43	

## DAFTAR GAMBAR

<b>Gambar</b>	<b>Halaman</b>
Gambar 1.1. Diagram Keterkaitan Masalah .....	3
Gambar 1.2. Diagram Alir Metodologi Penelitian .....	6
Gambar 2.1 Contoh Rute Penjadwalan <i>Job Shop</i> .....	11
Gambar 3.1 Rute Operasi .....	21
Gambar 4.1 <i>Gantt Chart</i> Penjadwalan <i>Job Shop</i> Polman Bnadung .....	28
Gambar 4.2 Algoritma Tabu Search penjadwalan <i>Job Shop</i> Polman Bandung	33
Gambar 4.2 Grafik Hasil Run Algoritma <i>Tabu Search</i> .....	43



## DAFTAR LAMPIRAN

<b>Lampiran</b>		<b>Halaman</b>
Lampiran 1.	<i>Gantt Chart</i> Penjadwalan Polman Bandung.....	46
Lampiran 2.	Hasil Run Progrsm Solusi Akhir.....	49
Lampiran 3	Bahasa Pemrograman VBA .....	53



# BAB I PENDAHULUAN

## 1.1. Latar Belakang

Dalam sebuah industri, baik itu industri manufaktur maupun jasa, adanya suatu proses penyusunan penjadwalan yang baik merupakan suatu hal yang sangat penting. Hal ini dikarenakan dengan adanya penjadwalan yang baik akan dapat meningkatkan efektivitas serta efisiensi sistem produksi industri tersebut yang pada akhirnya akan mengurangi biaya produksi. Penjadwalan dapat diartikan sebagai pengalokasian sebagian pekerjaan dalam serangkaian mesin, setiap mesin hanya mengerjakan tidak lebih dari satu pekerjaan dalam satu waktu (Applegate dan Cook, 1991). Secara umum penjadwalan merupakan suatu permasalahan dalam hal melakukan pengurutan terhadap sejumlah operasi dan mengalokasikannya ke dalam slot waktu tertentu tanpa melanggar batasan teknis dan keterbatasan kapasitas yang dimiliki. Baik secara teori maupun prakteknya di lapangan, untuk dapat melakukan suatu proses penjadwalan yang baik sangat sulit untuk dibuat. Hal ini berdasar pada kenyataan bahwa begitu banyak parameter yang harus diperhatikan.

Secara umum permasalahan yang di hadapi perusahaan dalam proses penyusunan penjadwalan adalah terbatasnya waktu untuk menyelesaikan proses produksi yang harus sesuai dengan waktu penyerahan yang diminta konsumen (*due date*), di mana setiap konsumen mempunyai waktu penyerahan yang berbeda-beda sementara perusahaan memiliki keterbatasan dalam hal tenaga kerja, mesin, dan kapasitas produksi. Adakalanya perusahaan harus membayar biaya penalti karena pesanan konsumen diselesaikan melebihi waktu yang telah ditentukan. Oleh karena itu penjadwalan produksi diperlukan untuk mengatasi keterbatasan yang dimiliki perusahaan sehingga bisa meminimumkan waktu penyelesaian produksi secara keseluruhan.

Karena penjadwalan, khususnya penjadwalan *job shop*, merupakan suatu permasalahan *combinatorial optimization* yang kompleks maka permasalahan penjadwalan dapat dikategorikan sebagai permasalahan *nondeterministic polynomial hard* (np-hard), yaitu suatu permasalahan yang pencarian solusinya

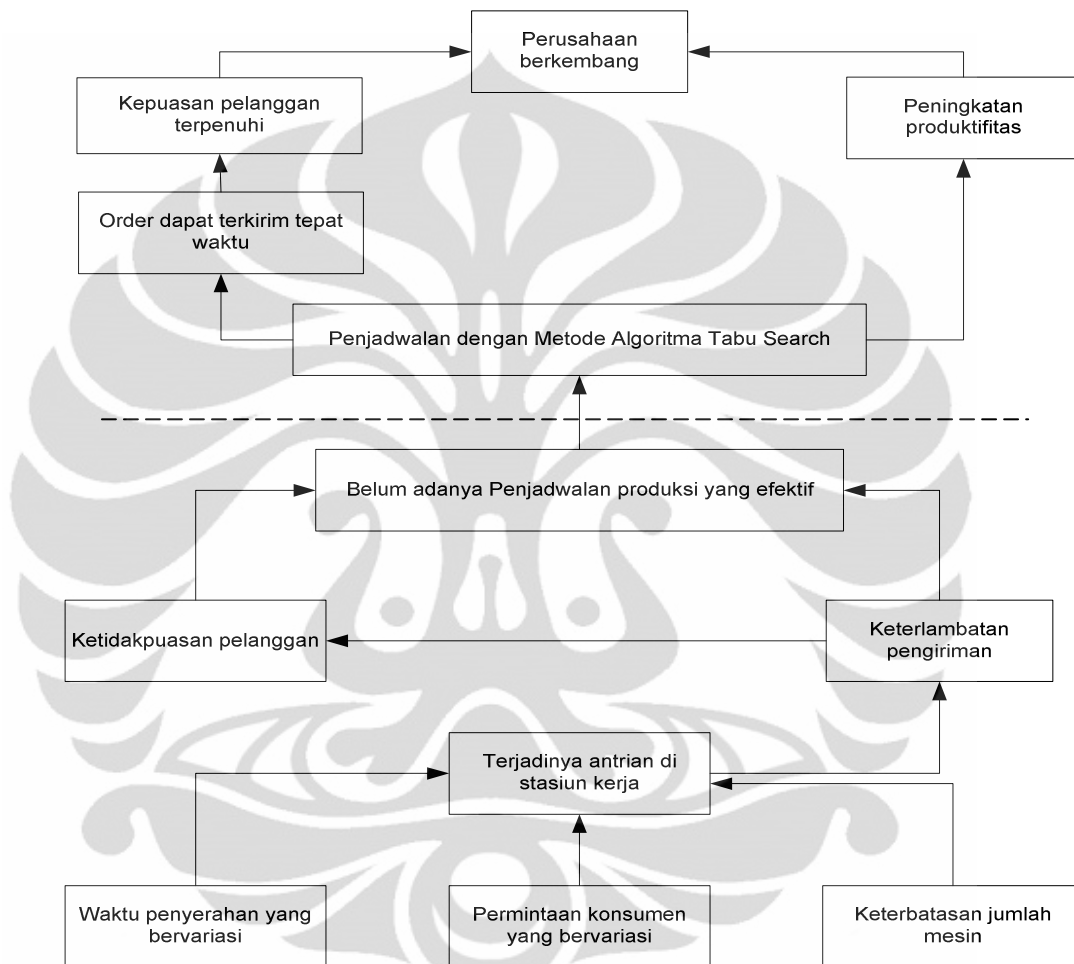
(waktu komputasinya) akan naik secara eksponensial seiring dengan naiknya ukuran permasalahan secara linier (Betrianis dan Putu T. A., 2003). Untuk itu diperlukan suatu metode yang lebih baik dalam memecahkan permasalahan ini.

Persoalan penjadwalan *jobshop* merupakan persoalan yang NP-hard untuk semua kriteria optimalitas. Sehingga dalam menyelesaikan masalah tersebut algoritma eksak hanya dapat digunakan untuk persoalan yang berukuran kecil saja (Zulkarnain, 2011). Untuk persoalan yang berdimensi besar algoritma heuristik dan metaheuristik menjadi pilihan yang menjanjikan.

Berdasarkan literatur yang ada, algoritma *tabu search* (TS) terbukti lebih cepat dari pada algoritma yang lain dalam menyelesaikan masalah penjadwalan *job shop* dan *flow shop* (Albert Jones dan Luis Rabelo, 1998). TS merupakan suatu jenis algoritma yang termasuk ke dalam jenis algoritma sub-optimal, jenis algoritma yang memberikan suatu solusi yang mendekati optimal. Ide dasar dari algoritma *tabu search* adalah mencegah proses pencarian dari *local search* agar tidak melakukan pencarian ulang pada ruang solusi yang sudah pernah ditelusuri, dengan memanfaatkan suatu struktur memori yang mencatat sebagian jejak proses pencarian yang telah dilakukan (Henry P. P., 2005). Algoritma *tabu search* digunakan untuk meminimalisasi makespan dalam permasalahan penjadwalan *job shop* (Eugeniusz Nowicki dan Czeslaw Smutnicki, 1996). Dilihat dari struktur spesial solusi berikutnya, metode *tabu search* bekerja lebih cepat dan lebih efisien dari pada algoritma yang lain. (Eugeniusz Nowicki dan Czeslaw Smutnicki, 1996). Pada beberapa tahap pelacakan dapat dikategorikan sebagai langkah *tabu* (dilarang) karena akan menghasilkan lokal optima dan juga karena akan mengakibatkan langkah pengulangan kembali ke solusi yang pernah ditemukan sebelumnya (*entrapment*). Langkah-langkah ini kemudian dimasukkan ke dalam daftar yang disebut dengan *tabu list*. Proses pencariannya itu sendiri dilakukan dengan cara menentukan solusi awal dan kemudian dilakukan gerakan (*move*) ke solusi-solusi berikutnya (*neighborhood*) dan baru berhenti sampai kriteria penghentian (*stopping conditions*) tercapai.

## 1.2. Diagram Keterkaitan Masalah

Untuk melihat gambaran sistematis yang lebih menyeluruh, termasuk hubungan antar masalah yang berinteraksi satu sama lain, maka disusun suatu diagram keterkaitan masalah seperti pada gambar 1.1.



**Gambar 1.1.** Diagram Keterkaitan Masalah

## 1.3. Perumusan Masalah

Tidak adanya penjadwalan yang optimal sehingga total waktu pengerjaan meningkat dari waktu seharusnya.

## 1.4. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah memperoleh suatu usulan sistem penjadwalan *job shop* yang lebih baik (lebih mendekati optimal)

dengan meminimumkan total waktu produksi untuk penyelesaian seluruh pekerjaan melalui penerapan Algoritma *Tabu Search*.

### 1.5. Batasan Masalah

Untuk mendapatkan hasil penelitian yang spesifik dan terarah, maka ruang lingkup permasalahan dari penelitian ini adalah sebagai berikut:

1. Rute proses produksi yang akan dibahas adalah rute proses produksi *jig and fixture, gauge, dan pitch gauge*, dan merupakan sistem penjadwalan yang akan diteliti.
2. Data besarnya waktu *set-up* dan perpindahan semua material sudah termasuk ke dalam waktu proses produksi yang bersangkutan.
3. Satu mesin hanya dapat memproses satu pekerjaan .
4. Penjadwalan bersifat statis dan *non-preemptive*, artinya semua order diterima di awal periode penjadwalan dan penjadwalan dilakukan di awal periode
5. Penelitian hanya memperhitungkan ketersediaan mesin.
6. Kondisi mesin produksi diasumsikan berjalan dengan normal, mengabaikan terjadinya *breakdown, dan rework*.
7. Fungsi tujuan yang ingin diperoleh yaitu meminimumkan total waktu pengerjaan seluruh job (*makespan*).

### 1.6. Metodologi Penelitian

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

- 1 Melakukan identifikasi permasalahan di perusahaan.
- 2 Mengumpulkan dan menyusun studi literatur yang berkaitan dengan masalah yang telah diidentifikasi.
- 3 Merumuskan masalah, yaitu perlunya perancangan suatu sistem penjadwalan yang efisien.
- 4 Menentukan tujuan, yaitu memperoleh suatu sistem penjadwalan yang lebih baik untuk meminimumkan total waktu pengerjaan untuk penyelesaian seluruh pekerjaan.



- 5 Mengidentifikasi data yang dibutuhkan dan selanjutnya mengumpulkan data sekunder perusahaan.
- 6 Membuat model matematis dari permasalahan lalu dilakukan pembuatan program dengan perangkat lunak *Visual Basic for Application*.
- 7 Melakukan validasi dan verifikasi terhadap program yang telah dibuat.
- 8 Melakukan *Experiment* untuk menentukan parameter-parameter yang akan digunakan untuk pengolahan data.
- 9 Membandingkan dan menganalisis solusi jadwal yang baru dengan jadwal yang lama, dimana faktor pembanding yaitu total waktu pengerjaan seluruh *job*.
- 10 Menarik kesimpulan berdasarkan hasil analisis tersebut.

### **1.7.Sistematika Penulisan**

Penulisan laporan penelitian ini dibagi menjadi lima bab.

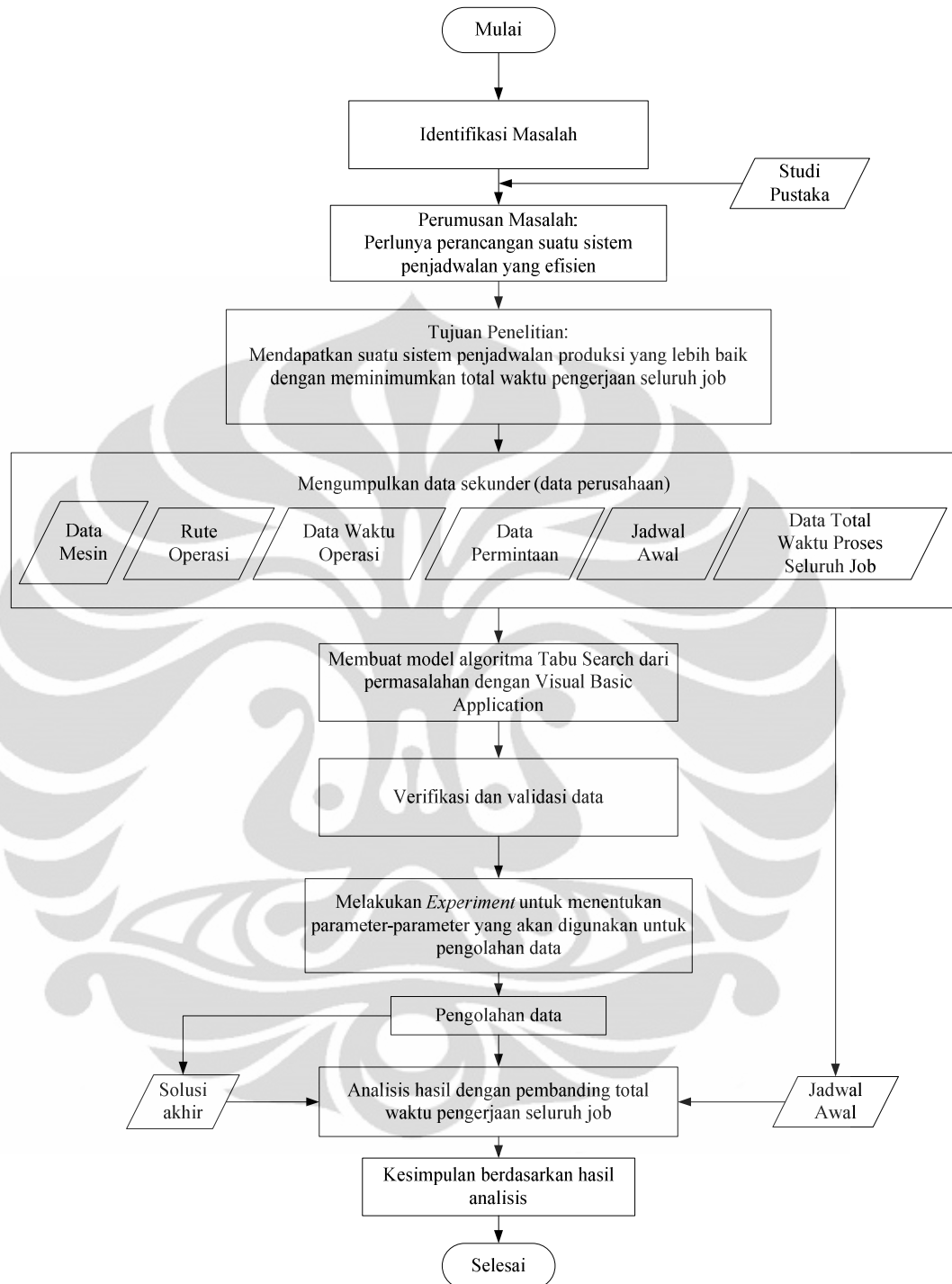
Bab 1 merupakan bab pendahuluan, menjelaskan mengenai latar belakang permasalahan, diagram yang menggambarkan keterkaitan masalah, perumusan masalah, tujuan penelitian yang ingin dicapai, batasan masalah yang dilakukan, metodologi penelitian yang dilakukan oleh penulis, serta sistematika penulisan.

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

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

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

Bab 5 merupakan kesimpulan yang diambil berdasarkan hasil penelitian dan analisisnya.



**Gambar 1.2.** Diagram Alir Metodologi Penelitian

## BAB II

### LANDASAN TEORI

#### 2.1. Penjadwalan Produksi

Penjadwalan produksi merupakan hal yang penting dalam organisasi untuk memperoleh pemanfaatan atau utilisasi yang optimal dari sumber daya produksi dan aset lain yang dimiliki. Penjadwalan diperlukan agar alokasi tenaga operator, mesin dan peralatan produksi, urutan proses, jenis produk, pembelian material dan sebagainya menjadi efisien. Di samping keputusan perencanaan jangka menengah, ada masalah lain yang disebut penjadwalan yang mana alokasi sumber daya dan urutan pengerjaan menjadi sangat penting. Dalam hierarki pengambilan keputusan, penjadwalan merupakan langkah terakhir sebelum dimulainya operasi.

##### 2.1.1. Pengertian Penjadwalan Produksi

Penjadwalan produksi secara umum didefinisikan sebagai penetapan waktu dari penggunaan peralatan, fasilitas, dan aktivitas manusia dalam sebuah organisasi (Everett dan Ronald, 1992). Penjadwalan produksi mencakup tahapan *loading*, *sequencing*, dan *detailed scheduling*. Pada tahap *loading*, setiap *job* ditentukan prosesnya, kemudian beban (*load*) setiap mesin ditentukan melalui pekerjaan yang harus diproses, dan ditentukan urutan pengerjaan *job* yang dikenal dengan sebutan *sequencing*. Dari urutan tersebut diatur waktu mulai dan selesainya pekerjaan melalui penjadwalan secara mendetail.

##### 2.1.2. Tujuan Penjadwalan Produksi

Penjadwalan memiliki beberapa tujuan. Namun tujuan tersebut dapat saling berkontradiksi. Oleh karena itu, upaya pengoptimalan penjadwalan sangat diperlukan. Adapun tujuan penjadwalan produksi (Steven Nahmias, 1997) antara lain:

- Memenuhi waktu pesanan.
- Meminimumkan waktu *set-up*, waktu *work in process*, dan *idle time*.
- Menghasilkan tingkat kegunaan mesin atau pekerja yang tinggi.
- Menetapkan informasi pekerjaan yang cepat.

- meminimumkan biaya produksi dan tenaga kerja.

### 2.1.3. Klasifikasi Penjadwalan Produksi

Penjadwalan produksi menurut Pinedo dan Chao (1999) dibagi menjadi beberapa kriteria yaitu:

Berdasarkan mesin yang dipergunakan dalam proses:

- penjadwalan pada mesin tunggal (*single machine shop*).
- penjadwalan pada mesin jamak.

Berdasarkan pola kedatangan *job*:

- penjadwalan statis, dimana *job* datang bersamaan dan siap dikerjakan pada mesin yang tidak bekerja.
- penjadwalan dinamis, dimana kedatangan *job* tidak menentu.

Berdasarkan lingkungan penjadwalan:

- *Flow Shop*

Tiap *job* atau pesanan memiliki rute pengerjaan (*routing*) yang sama. Aliran bisa bersifat diskrit, kontinu, maupun semikontinu.

- *Job Shop*

Setiap *job* atau pesanan memiliki rute pengerjaan yang berbeda-beda, sesuai permintaan konsumen (*complex routing*). Karena kompleksnya aliran, maka penjadwalan pun sangat kompleks. Aliran bersifat diskrit, dan *part* tidak bersifat multiguna (*part* yang mungkin menjadi WIP pada *job* yang satu tidak bisa digunakan pada *job* yang lain).

- *Assembly Line*

Hampir serupa dengan *flow shop*, akan tetapi proses hanya meliputi bagian perakitan dengan volume yang tinggi dan karakteristik produk yang sedikit. Tidak ditemui *buffer inventory*, kecuali pada bagian awal lini perakitan.

#### 2.1.4. Istilah dalam Penjadwalan Produksi

Berikut istilah-istilah beserta notasinya yang digunakan dalam penjadwalan (Pinedo dan Chao, 1999):

- Setiap *Job*  $i \in \{1, 2, \dots, n\}$  yang akan dijadwalkan pada  $j$  mesin  $\{j=1, 2, \dots, m\}$ . Proses pengerjaan *job*  $i$  pada mesin  $j$  disebut dengan operasi  $O_{ij}$ .
- Waktu proses (*processing time*),  $p_{ij}$ , yaitu lamanya waktu yang harus dihabiskan *job*  $i$  di mesin  $j$  untuk memproses operasi  $O_{ij}$ .
- Waktu tenggat (*due date*),  $d_i$ , adalah batas waktu penyelesaian *job*  $i$  yang telah ditentukan. Apabila penyelesaian *job* diluar waktu ini, maka akan dikenakan penalti pada *job* tersebut.
- Waktu siap (*release date*),  $r_i$ , adalah waktu ketika *job*  $i$  masuk ke sistem, yaitu waktu paling awal *job*  $i$  bisa mulai diproses. Biasanya  $r_i = 0$ .
- Waktu mulai (*start time*),  $s_{ij}$ , adalah waktu mulai diprosesnya *job*  $i$  di mesin  $j$ .
- Waktu penyelesaian (*completion time*),  $C_{ij}$ , adalah waktu penyelesaian pemrosesan *job*  $i$  pada mesin  $j$ .
- *Makespan* biasanya dilambangkan dengan  $C_{max}$ , yaitu waktu pengerjaan seluruh *job*.
- Keterlambatan (*lateness*),  $L_i = C_i - d_i$ , adalah selisih antara waktu penyelesaian *job*  $i$  dengan waktu tenggatnya. *Lateness* baru dapat dihitung setelah *job*  $i$  selesai menjalani semua proses, dan dapat bernilai negatif, nol, atau positif.
- Keterlambatan positif (*tardiness*),  $T_i = \max(L_i, 0)$ , adalah besarnya keterlambatan penyelesaian *job*  $i$ .
- Keterlambatan negatif (*earliness*),  $T_i = \min(L_i, 0)$ , adalah besarnya keterlambatan penyelesaian *job*  $i$ .

#### 2.1.5. Karakteristik dan Kendala Proses

Kendala penjadwalan produksi menurut Pinedo dan Chao (1999), yaitu:

- Kendala *precedence*

Kendala ini terjadi ketika suatu *job* baru dapat mulai diproses setelah satu atau sekumpulan *job* lainnya telah selesai diproses

- Kendala biaya dan waktu *setup* yang bergantung pada urutan *job* (*sequence-dependent*)

- *Preemption*

*Preemption* berarti jika proses produksi sedang berlangsung, maka dapat dihentikan dan digantikan dengan mengerjakan *job* yang baru datang. Keadaan ini biasanya dikarenakan *job* yang berprioritas rendah dapat disela prosesnya oleh *job* yang berprioritas tinggi.

- Kendala mesin dan pekerja

Dalam lingkungan mesin paralel, karakteristik mesin yang digunakan harus sama. Jika tidak sama, maka akan mengganggu proses produksi. Selain itu, umur mesin juga mempengaruhi kapasitas produksi yang dihasilkan. Sedangkan kendala pekerja berkaitan dengan penjadwalan jam kerja operator.

#### 2.1.6. Fungsi Tujuan dan Pengukuran Performa Penjadwalan Produksi

Tujuan yang biasa digunakan untuk menilai performa penjadwalan yang dibuat adalah sebagai berikut

- Meminimumkan *flow time* dan *makespan*.
- Memaksimumkan utilisasi (minimasi waktu mesin dan pekerja yang menganggur).
- Meminimumkan *inventory* dan WIP (*work in process*).
- Meminimumkan keterlambatan baik *earliness* maupun *tardiness*.
- Meminimumkan jumlah *job* yang terlambat (*number of tardy job*).
- Meminimumkan total waktu pengerjaan seluruh *job*.

## 2.2. Penjadwalan *Job Shop*

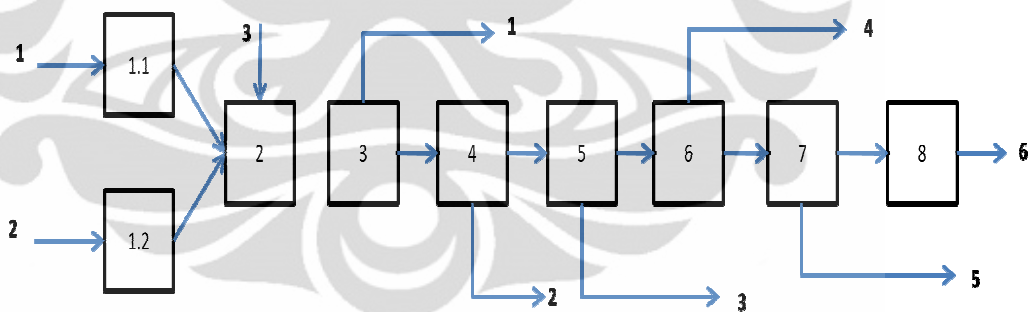
*Job shop* adalah suatu lingkungan manufaktur dimana *job-job* yang datang memiliki rute pengerjaan atau operasi yang seringkali tidak sama. Bentuk sederhana dari model ini mengasumsikan bahwa setiap *job* hanya melewati satu jenis mesin sebanyak satu kali dalam rutenya pada proses tersebut. Namun ada

juga model lainnya dimana setiap *job* diperbolehkan untuk melewati mesin sejenis lebih dari satu kali pada rutenya. Model ini disebut juga *job shop* dengan *recirculation* (pengulangan).

Karakteristik penjadwalan *job shop* dapat dijabarkan sebagai berikut:

- Ada sejumlah  $m$  mesin dan sejumlah  $n$  job.
- Setiap job terdiri dari satu rantai urutan yang dapat berbeda satu sama lain.
- Setiap operasi dalam job diproses oleh salah satu mesin yang ada dengan waktu proses yang diasumsikan tetap.
- Setiap proses operasi dapat melewati satu jenis mesin lebih dari satu kali.
- Tidak ada *preemption* (penundaan satu *job* oleh *job* lain).
- Permasalahan penjadwalan untuk model *job shop* merupakan salah satu permasalahan optimasi kombinatorial yang kompleks sehingga disebut *NP-hard* (*NP* merupakan singkatan dari *nondeterministic polynomial*).

Bentuk permasalahan penjadwalan model *job shop* dapat digambarkan dalam bentuk seperti di bawah ini:



**Gambar 2.1.** Contoh Rute Penjadwalan *Job Shop*

Sumber: Polman Bandung

### 2.3. Metode Penyelesaian Masalah Penjadwalan Produksi

Masalah penjadwalan produksi dapat diselesaikan dengan menggunakan metode heuristik yang terdiri dari 2 jenis, yaitu:

### 2.3.1 Tipe Heuristik Klasik

Algoritma ini menyusun satu per satu solusi dari masalah penjadwalan. Mulai dari nol, algoritma-algoritma ini memilih mesin-mesin atau *job-job* atau operasi-operasi mana yang harus dijadwalkan terlebih dahulu. Algoritma heuristik klasik yang sering digunakan untuk menyelesaikan penjadwalan *job shop* yaitu *priority dispatch rule*.

*Priority dispatch rule* adalah suatu aturan penjadwalan yang mengatur *job* mana pada suatu antrian *job* pada suatu mesin yang harus diproses terlebih dahulu berdasarkan prioritas-prioritas tertentu. Jadi, pada saat suatu mesin telah selesai memroses satu *job*, maka berdasarkan *priority dispatch rule* dipilih satu *job* yang memiliki prioritas tertinggi untuk selanjutnya diproses pada mesin tersebut. Berikut ini adalah beberapa aturan yang merupakan *basic priority dispatch rules* (Everett dan Ronald, 1992), yaitu:

- *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.
- *Earliest Due Date (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 di antara *job-job* lainnya dipilih sebagai *job* yang memiliki prioritas paling tinggi untuk diproses pada sebuah mesin. Aturan ini cenderung digunakan untuk meminimumkan maksimum *lateness* pada *job-job* yang ada dalam antrian.
- *Minimum Slack First (MS)*  
Menurut aturan ini, *job* diurutkan berdasarkan waktu *slack* yang paling kecil. Pada saat sebuah mesin selesai memroses 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 memiliki waktu *slack* yang paling kecil kemudian dipilih sebagai *job* yang memiliki prioritas paling tinggi



untuk diproses selanjutnya. Aturan ini digunakan untuk meminimumkan fungsi tujuan yang berkaitan dengan *due date*, yaitu *lateness* dan *tardiness*.

- *Shortest Processing Time First (SPT)*

Menurut aturan ini, *job* diurutkan berdasarkan pada lamanya waktu proses tiap *job*. Jadi, *job* yang memiliki waktu proses paling singkat akan diproses terlebih dahulu dan kemudian dilanjutkan oleh *job-job* lainnya sampai pada *job* yang paling lama waktu prosesnya. Aturan ini berguna untuk penyeimbangan beban kerja antar mesin yang disusun secara paralel.

### 2.3.2 Tipe Heuristik Modern (Meta-Heuristik)

Algoritma heuristik modern atau yang lebih dikenal dengan meta-heuristik memecahkan masalah penjadwalan produksi dengan melakukan perbaikan mulai dengan satu atau lebih solusi awal. Solusi awal ini dapat dihasilkan secara acak, dapat pula dihasilkan berdasarkan heuristik tertentu. Empat algoritma meta-heuristik yang dapat digunakan dalam memecahkan masalah penjadwalan *job shop* yaitu:

- *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.

- *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 baik jika dibandingkan dengan solusi-solusi lainnya dalam himpunan solusi tetangga tersebut. Tetangga (*neighbour*) dari suatu solusi adalah solusi-solusi lain yang dapat diperoleh dari solusi tersebut dengan cara memodifikasinya berdasarkan aturan-aturan tertentu yang dikenal dengan nama *neighborhood functions*.

- 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). 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.

- Algoritma *Differential Evolution*

*Differential Evolution Algorithm* (Algoritma Evolusi Diferensial) merupakan metode metaheuristik akhir. Metode ini terbilang cukup baru, merupakan versi pengembangan dari Algoritma Genetika. Prinsipnya adalah berdasarkan analogi evolusi biologi, yang terdiri dari proses penginisiasian populasi, proses mutasi, proses penyilangan, dan proses penyeleksian. Keunggulan algoritma ini adalah berstruktur sederhana, mudah dalam pengimplementasian, cepat dalam mencapai solusi, dan bersifat tangguh (memiliki standar deviasi yang kecil).

## 2.4. Algoritma *Tabu Search* (TS)

### 2.4.1. Latar Belakang *Tabu Search*

Menghadapi tantangan dalam menyelesaikan permasalahan optimalisasi yang sulit pada dunia nyata, metode klasik terkadang menjumpai kesulitan yang besar, yang terkadang tidak dapat mengatasi beberapa aplikasi penting dalam bisnis, teknik, ekonomi, dan sains dengan kemungkinan keberhasilan yang tinggi dalam jangka waktu yang singkat. Pendekatan heuristik *tabu search* secara dramatis merubah kemampuan kita untuk menyelesaikan permasalahan ketidakpraktisan tersebut. *Tabu search* menghasilkan solusi yang memiliki kualitas melebihi hasil yang didapatkan dari metode yang terdahulu.

Kata *tabu* (atau *taboo*) berasal dari Tongan, bahasa Polynesia, untuk mengindikasikan sesuatu yang tidak dapat disentuh karena hal itu adalah keramat. Menurut *Webster's Dictionary*, kata itu berarti sebuah larangan yang dibentuk oleh kebiasaan sosial sebagai ukuran pencegahan atau dilarang karena mengandung risiko. "*Tabu*" dari *tabu search* dapat dikesampingkan bila terdapat bukti yang "memaksa" atau mendorong dari alternatif yang lebih baik. Elemen-elemen terlarang dalam daftar "*tabu*" dari *tabu search* menerima statusnya dengan bergantung kepada perkembangan memori, yang dapat berganti sesuai dengan pergantian waktu dan lingkungan.

*Tabu search* berdasarkan pada alasan bahwa pemecahan masalah harus menggabungkan memori yang mampu beradaptasi dan penggalan responsif. Memori yang mampu beradaptasi (*adaptive memory*) dari *tabu search* memperbolehkan implementasi prosedur yang mampu mencari ruang solusi secara ekonomis dan efektif, yang kontras dengan metode tanpa memori (seperti pendekatan genetik dan *annealing*) dan mengandalkan pada proses semi-acak. *Tabu search* juga kontras dengan metode pencarian yang kaku seperti strategi *branch and bound*. Sedangkan penekanan pada penggalan responsif adalah diturunkan dari perkiraan bahwa pilihan strategi yang buruk dapat mengeruk lebih banyak informasi dan menyediakan berbagai petunjuk yang berguna tentang bagaimana strategi tersebut dapat diubah secara menguntungkan. Penggalan responsif mengintegrasikan prinsip-prinsip dasar dari pencarian cerdas, yaitu

memanfaatkan ciri-ciri solusi yang baik selama menggali menggali daerah solusi baru yang menjanjikan.

#### 2.4.2. Algoritma *Tabu Search*

Tidak seperti pencarian menurun, algoritma *tabu search* mampu meninggalkan optimasi lokal dan melanjutkan pencarian. Pendekatan *tabu search* mulai dari solusi awal, dan langkah berpindah ke solusi tetangga dipilih dengan harapan dapat mengembangkan nilai kriteria tujuan yang ingin dicapai. Algoritma ini mencoba untuk mengambil langkah yang memastikan bahwa setiap metode tidak masuk kembali ke solusi yang telah dipakai sebelumnya agar tidak terperangkap pada penyelesaian lokal. Untuk mencegah penyelesaian lokal tersebut, digunakan *tabu list* (struktur data berdasarkan kekinian waktu) yang berisi pergerakan-pergerakan terlarang pada iterasi terkini. Pergerakan-pergerakan terlarang ini hanya terbatas pada beberapa iterasi, dan algoritma ini tidak menjamin penyelesaian yang paling optimal, tetapi mendekati optimal. Algoritma ini secara spesifik digunakan pada permasalahan gabungan (*combinatorial*), dan diaplikasikan pada, penjadwalan *job-shop*, penjadwalan *flow-shop* (Ben-Daya & Al-Fawzan, 1998), lokasi fasilitas tidak berkapasitas (Al-Sultan & Al-Fawzan, 1999), disain arsitektur (Bland & Dawson, 1991), permasalahan *time-tabling* (Hertz, 1991), dan lain-lain. Secara singkat, elemen-elemen dasar pada teknik *tabu search* terdiri dari:

- Konfigurasi adalah suatu solusi atau penempatan nilai kepada suatu variabel;
- Suatu pergerakan menggambarkan proses/prosedur penurunan sebuah solusi yang mungkin pada permasalahan kombinasi yang berhubungan dengan solusi saat ini.
- Suatu set kandidat pergerakan-pergerakan adalah suatu set dari semua pergerakan yang mungkin pada konfigurasi terkini.
- Pembatasan tabu digambarkan sebagai suatu kondisi tertentu yang menentukan pergerakan-pergerakan menjadi terlarang (tabu). Hal ini dilakukan dengan menyusun suatu daftar dengan ukuran tertentu yang menyimpan pergerakan-pergerakan terlarang ini (*tabu list*).

**Universitas Indonesia**

- Kriteria aspirasi. Kriteria ini merupakan peraturan yang mengesampingkan pembatasan *tabu*, atau bahkan mengizinkan pergerakan yang sebenarnya terlarang.

Kerangka *tabu search* dapat digambarkan secara ringkas sebagai berikut:

- Mulai dengan konfigurasi terkini/tertentu (solusi awal), dan evaluasi fungsi tujuan dari konfigurasi tersebut.
- Lalu, turunkan suatu set kandidat pergerakan-pergerakan dari konfigurasi terkini.
- Jika yang terbaik dari semua pergerakan ini tidak *tabu* atau jika yang terbaik tersebut adalah *tabu* tetapi memenuhi kriteria aspirasi, pergerakan itu diambil dan ditentukan sebagai konfigurasi yang baru. Jika tidak, tentukan pergerakan terbaik lainnya yang tidak *tabu* dan pergerakan tersebut diambil menjadi konfigurasi terbaru.
- Ulangi prosedur di atas hingga jumlah iterasi tertentu.
- Pada perhentian, solusi terbaik yang dihasilkan sedemikian jauh adalah solusi yang didapat dengan algoritma ini.
- Pergerakan yang diambil pada iterasi tertentu diletakkan pada daftar *tabu* (*tabu list*) sehingga pergerakan tersebut tidak diperbolehkan diambil lagi pada beberapa iterasi berikutnya.
- Daftar *tabu* memiliki ukuran tertentu, dan ketika panjang daftar *tabu* mencapai ukurannya dan pergerakan baru memasuki daftar tersebut, maka pergerakan pertama dari daftar *tabu* tersebut dibebaskan dari *tabu* dan proses berlanjut (daftar *tabu* adalah sirkular). Ukuran daftar *tabu* ini digunakan untuk mengontrol intensitas pencarian.
- Kriteria aspirasi dapat diambil berdasarkan nilai dari fungsi tujuan, yaitu jika pergerakan *tabu* menghasilkan nilai fungsi tujuan yang lebih baik dibandingkan dengan hasil terbaik hingga sejauh ini, sehingga kriteria aspirasi terpenuhi dan pembatasan *tabu* dengan ini terlampaui.

Secara garis besar, algoritma *Tabu Search* dapat dilihat sebagai berikut:

**Step 1**

$k=1$

Select an initial schedule  $S_1$  using some heuristic and set  $S_{best}=S_1$

**Step 2**

Select  $S_c \in N(S_k)$

If the move  $S_k \rightarrow S_c$  is prohibited by a move on the tabu-list

Then go to Step 2

If the move  $S_k \rightarrow S_c$  is not prohibited by a move on the tabu-list

Then  $S_{k-1}=S_c$

Enter reverse move at the top of the tabu-list

Push all other entries in the tabu-list one position down

Delete the entry at the bottom of the tabu-list

If  $F(S_c) < F(S_{best})$  then  $S_{best}=S_c$

Go to Step 3

**Step 3**

$k=k+1$

If stopping condition=true then STOP

Else go to Step 2

## **BAB III**

### **PENGUMPULAN DATA**

#### **3.1. Profil Polman Bandung**

Politeknik Mekanik Swiss – ITB (PMS-ITB) berdiri sejak tahun 1976 merupakan hasil kerjasama bilateral antara pemerintah RI dan pemerintah Konfederasi Swiss, yang dalam pelaksanaannya Pemerintah Indonesia diwakili oleh ITB dan Swisscontact mewakili Pemerintah Swiss yang berakhir pada tahun 1995. Setelah kerjasama berakhir dan pemerintah Indonesia menggulirkan kebijakan politeknik mandiri maka PMS-ITB berubah nama menjadi Politeknik Manufaktur Negeri Bandung (Polman Bandung). Polman Bandung dapat membuat produk-produk Industri yang berkualitas sesuai dengan kepresisian yang didisain, diproses dan dikontrol dengan prosedur peralatan yang terkini dan didukung oleh tenaga-tenaga profesional.

#### **3.2. Pengumpulan Data Penelitian**

Data yang digunakan dalam penyusunan tugas akhir ini merupakan data sekunder. Data sekunder tersebut berasal dari bagian PPC Polman Bandung. Data-data yang digunakan untuk penyusunan skripsi ini berupa:

- Data jam kerja Polman Bandung;
- Data rute proses operasi produksi yang harus dilalui oleh setiap *komponen* yang akan diproduksi;
- Data jumlah mesin yang harus dilalui untuk rute-rute tiap operasi;
- Data waktu proses setiap mesin.
- Data pesanan konsumen;

##### **3.2.1. Data Pesanan dan Jam Kerja**

Pesanan diterima kurang lebih 3 bulan sebelum *due date*-nya. Keadaan ini diharuskan sehingga PPC dapat terlebih dahulu menyusun MPS, MRP dan rencana pemesanan barang. Apabila PPC merasa bahwa permintaan dapat dipenuhi sesuai *due date*-nya, maka PPC akan membuat kartu proses yang di dalamnya terdapat jadwal kapan material harus diantar, kapan waktu di mulai

pekerjaan A dikerjakan di mesin 1, dan setelah selesai di proses aliran selanjutnya diserahkan sesuai dengan yang tertulis di kartu proses tersebut. Dalam kartu proses juga disertai dengan *lead time* setiap operasi.

Untuk penelitian ini, data pesanan yang akan dibahas yaitu Pitch gauge PP YL 8, Pitch gauge PP YL 8/DO1N, Wire ring gauge YL 8, Assy jig fixture Spline hub yl8, PM 110407. Komponen ini diambil hanya sebagai contoh kasus penjadwalan. Jadi, ketika PPC selesai membuat estimasi untuk waktu pengerjaan dan harga, maka akan disusun *Production Order* (PO) yang berisi *due date* setiap *job* pada *stage* produksi sampai ke *stage assembly*. Setiap paket pesanan berbeda jenis, jumlah pesannya, dan *due datenya*. *Due date* dihitung mulai dari PPC memberikan PO ke bagian produksi sampai jadwal barang dikirimkan ke bagian *assembly*. Dalam 1 minggu terdapat 5 hari kerja dimana 1 hari hanya memiliki 1 shift dengan rincian waktu pada tabel 3.1.

**Tabel 3.1.** Waktu Kerja Polman Bandung

	waktu kerja	Durasi (mnt)	total jam kerja (mnt)	
Senin - Kamis	07.00-11.30	270	450	
	12.30-15.30	180		
jumat	07.00-11.30	270	420	
	13.00-15.30	150		
Total jam kerja dalam seminggu			2,220	37 (jam)

Sumber : Jam kerja Polman Bandung

Berikut data pesanan yang diambil sebagai contoh kasus penjadwalan. Total masing-masing item yang harus di produksi adalah 1 buah.

Data pesanan konsumen:

1. *Pitch gauge PP YL 8*: semacam *checking fixture* untuk part dari produk *clutch* yang merupakan pesanana PT.Exedy
2. *Pitch gauge PP YL 8/DO1N*: semacam *checking fixture* untuk part dari produk *clutch* yang merupakan pesanana PT.Exedy
3. *Wire ring gauge YL 8*: semacam *checking fixture* untuk part dari produk *clutch* yang merupakan pesanana PT.Exedy
4. *Assy jig fixture Spline hub yl8*. semacam *checking fixture* untuk part dari produk *clutch* yang merupakan pesanana PT.Exedy
5. PM 110407: komponen penggiling sutau material dari batu pada sub proses di PT. KHI (anak perusahaan krakatau steel).

Universitas Indonesia



### 3.2.2. Rute dan Waktu Operasi

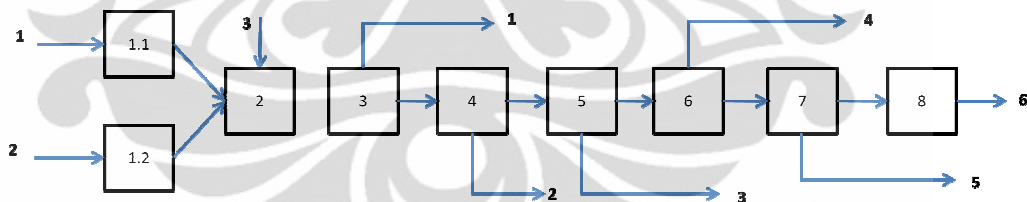
Rute proses operasi seluruh *job* merupakan rute *job shop* karena tidak semua barang melewati rute yang sama. Tabel 3.3. merupakan data jumlah mesin yang digunakan untuk proses penjadwalan.

**Tabel 3.2.** Jumlah Mesin yang Digunakan

No.	Nama Mesin	Spesifikasi	Tipe	Jumlah	No. Mesin
1	Mesin Bubut	245 x 2000	Dania 245	2	BU
2	Mesin Frais CNC	600x300x300	M - V5CN Mitsubishi	1	F09 CNC
3	Mesin Bor Coordinat	550 x 320	23 ST 42 Aciera	1	BOK 01P
4	Mesin Gerinda Datar	600 x 350	SJ 24 Jacobsen	1	GD 01P
5	Mesin Gerinda Silinder	100 x 400	HTG 400	1	GS 01P
6	Mesin EDM	500 X 400 X 260	FORM 4-LC Charmilles	1	EDM 04 P
7	Mesin WEDM (wire cut)	0	Robofil 390 Charmilles	1	EDM P
8	Hardening	0	HTG 234	1	Hard

Sumber: Data Mesin Polman Bandung

Untuk lebih jelas mengenai urutan rute operasi seluruh *job*, dapat dilihat pada gambar 3.1. berikut ini



**Gambar 3.1** Rute Operasi

Sumber: Data Polman Bandung

Satu pekerja hanya mengerjakan satu operasi pada satu alat atau mesin. Jadi, ketika *job* 1 proses A pada mesin A selesai, selanjutnya pekerja langsung akan mengerjakan *job* 2 proses A pada mesin A juga. Tabel 3.4. merupakan waktu operasi komponen untuk setiap proses. Waktu operasi pada suatu rute yang bernilai nol menunjukkan bahwa jenis komponen yang bersangkutan tidak melewati rute tersebut.

Tabel 3.3. Waktu Operasi Setiap Komponen Dalam Jam (sampel)

Jenis Komponen	1	2	3	4	5	6	7	8
	BU	fr cnc	BOK	HARD-1	GS-1	GD	WCU	edm
Assy jig fixture spline hub yl 8-1	1.25	0	1.10	2.00	0	2.25	2.50	0
Assy jig fixture spline hub yl 8-2	1.50	0	0.25	2.00	0	2.00	0	0
Assy jig fixture spline hub yl 8-3	1.50	0	0.25	2.00	0	2.00	0	0
Assy jig fixture spline hub yl 8-4	1.50	0	0.25	2.00	0	2.00	0	0
Assy jig fixture spline hub yl 8-5	1.50	0	0.25	2.00	0	2.00	0	0

Sumber: Data Polman Bandung

### 3.2.3. Formulasi Model Permasalahan

Dari data dan permasalahan yang telah diidentifikasi di awal, maka dapat diformulasikan sebuah model permasalahan dalam bentuk model matematis. Adapun model matematis tersebut bisa dijelaskan sebagai berikut.

Variabel :

$X_{ijk}$  = Waktu dimulainya job i part j pada mesin k

$Y_{ijkl} = \begin{cases} 1 & \text{job i part j pada mesin k ke l} \\ 0 & \text{Tidak} \end{cases}$

Untuk  $k=1$   $l = 1,2$

Untuk  $k \neq 1$   $l = 1$

Untuk  $k=1$   $Y_{ij11} + Y_{ij12} = 1$

Untuk  $k \neq 1$   $Y_{ij12} = 0$

Parameter :

$W_{ijk}$  = Lama proses job I part j pada mesin k

- Kendala part yang berurutan

$$X_{ijk+1} \geq X_{ijk} + Y_{ijkl} * W_{ijk}$$

- Pada sebuah mesin , 2 buah kegiatan tidak dapat dilakukan secara bersamaan.

Untuk  $k > 1$

Pekerjaan a b (i = a ; j = b) dan pekerjaan c d (i = c, j = d)

$$\text{ab setelah cd} \quad X_{abk} \geq X_{cdk} + W_{cdk} \rightarrow X_{abk} - X_{cdk} \geq W_{cdk}$$

Atau

$$\text{cd setelah ab} \quad X_{cdk} \geq X_{abk} + W_{abk} \rightarrow X_{cdk} - X_{abk} \geq W_{abk}$$

Dilinerkan menjadi :

$$MY_{abcdkl} + X_{abk} - X_{cdk} \geq W_{cdk} \quad \forall i \forall j \forall k \forall l$$

$$M(1-Y_{abcdkl}) + X_{cdk} - X_{abk} \geq W_{abk} \quad \forall i \forall j \forall k \forall l$$

$$Y_{abcdkl} = \begin{cases} 0 & \text{job a part b dikerjakan setelah job c part d pada mesin k ke l} \\ 1 & \text{Tidak} \end{cases}$$

Untuk mesin  $k = 1$  jumlah mesinnya ada 2 yaitu  $l = 1$  dan 2

$$Y_{ij11} + Y_{ij12} = 1$$

$$MY_{abcd11} + X_{ab1} - X_{cd1} \geq Y_{cd11} * W_{cd1} \quad \forall i \forall j \forall k \forall l$$

$$M(1-Y_{abcd11}) + X_{cd1} - X_{ab1} \geq Y_{ab11} * W_{ab1} \quad \forall i \forall j \forall k \forall l$$

$$Y_{abcd11} = \begin{cases} 0 & \text{Jika job a part b dikerjakan setelah job c part d pada mesin k ke l} \\ 1 & \text{Jika tidak} \end{cases}$$

- Fungsi tujuan :

$$\text{Job } i \text{ selesai pada waktu } X_{ij9} + W_{ijk} = y_i$$

Meminimumkan makespan

$$\text{Min } z = p$$

$$p \geq y_i \quad \rightarrow \quad p - y_i \geq 0 \quad \forall i$$

Tabel 3.4. Data waktu proses operasi

Job	No	Kode part	(dalam jam)							
			1	2	3	4	5	6	7	8
			BU	fr cnc	BOK	HARD-1	GS-1	GD	WCU	edm
	1	PITCH GAUGE PP YL 8-1	0.60	-	-	2.00	1.10	-	5	-
	2	PITCH GAUGE PP YL 8-2	0.35	-	0.60	1.00	2.35	-	2.76	1.12
	3	PITCH GAUGE PP YL 8-3	3.00	-	-	1.20	1.10	-	-	-
	4	PITCH GAUGE PP YL 8-4	0.35	-	-	0.97	1.10	-	-	-
	5	PITCH GAUGE CNP -YL8/DO1N-1	4.25	-	4.50	0.75	-	2.50	16.00	-
	6	PITCH GAUGE CNP -YL8/DO1N-2	2.50	-	-	2.00	1.10	-	-	-
	7	PITCH GAUGE CNP -YL8/DO1N-3	2.60	-	-	3.00	1.10	-	-	-
	8	PITCH GAUGE CNP -YL8/DO1N-4	1.00	-	2.1	-	-	0.35	-	2.50
	9	PITCH GAUGE CNP -YL8/DO1N-5	0.75	-	1.00	-	-	-	1.00	1.25
	10	PITCH GAUGE CNP -YL8/DO1N-6	0.75	-	1.00	-	2.35	-	1.00	-
	11	PITCH GAUGE CNP -YL8/DO1N-7	1.25	-	-	3.21	-	-	-	2.00
	12	PITCH GAUGE CNP -YL8/DO1N-8	1.00	-	-	2.50	1.10	-	2.50	1.10
	13	PITCH GAUGE CNP -YL8/DO1N-9	1.25	-	-	2.00	1.25	-	-	-
	14	PITCH GAUGE CNP -YL8/DO1N-10	0.75	1.75	-	-	1.20	3.25	-	-
	15	PITCH GAUGE CNP -YL8/DO1N-11	0.35	-	2.00	1.30	0.97	-	-	-
	16	PITCH GAUGE CNP -YL8/DO1N-12	1.00	-	3.00	1.10	0.75	-	-	-
	17	PITCH GAUGE CNP -YL8/DO1N-13	1.00	-	-	0.50	2.30	-	-	-
	18	PITCH GAUGE CNP -YL8/DO1N-14	1.25	-	-	2.00	-	-	1.50	-
	19	PITCH GAUGE CNP -YL8/DO1N-15	2.35	-	-	2.50	1.25	-	-	-
	20	PITCH GAUGE CNP -YL8/DO1N-16	1.25	2.50	-	-	3.00	-	-	-
2	21	PITCH GAUGE CNP -YL8/DO1N-17	1.25	-	2.34	2.50	1.20	-	-	-
	22	PITCH GAUGE CNP -YL8/DO1N-18	0.75	-	-	2.00	0.97	-	-	-
	23	PITCH GAUGE CNP -YL8/DO1N-19	0.35	-	-	-	0.75	-	2.00	-
	24	PITCH GAUGE CNP -YL8/DO1N-20	1.00	-	-	2.00	-	-	2.00	-
	25	PITCH GAUGE CNP -YL8/DO1N-21	2.00	-	-	1.50	2.00	-	-	-
	26	PITCH GAUGE CNP -YL8/DO1N-22	3.35	-	-	-	2.00	-	-	-
	27	PITCH GAUGE CNP -YL8/DO1N-23	2.70	-	-	1.50	-	-	-	-
	28	PITCH GAUGE CNP -YL8/DO1N-24	2.70	2.00	1.10			-	1.10	-
	29	PITCH GAUGE CNP -YL8/DO1N-25	1.20	2.00	1.10			-	2.50	-
	30	PITCH GAUGE CNP -YL8/DO1N-26	0.97	-	-	3.00	1.10	-	-	-
	31	PITCH GAUGE CNP -YL8/DO1N-27	0.75	-	-	3.00	1.10	-	-	-
	32	PITCH GAUGE CNP -YL8/DO1N-28	2.70	-	-	3.00	2.50	-	-	-
	33	PITCH GAUGE CNP -YL8/DO1N-29	2.70	-	2.50	-	2.00	-	1.2	-
	34	PITCH GAUGE CNP -YL8/DO1N-30	3.35	2.00	-	-	2.50	-	2.4	-
	35	PITCH GAUGE CNP -YL8/DO1N-31	2.00	-	-	2.00	2.50	-	-	1.3
	36	PITCH GAUGE CNP -YL8/DO1N-32	2.00	-	-	2.00	-	-	2.00	2.50
	37	PITCH GAUGE CNP -YL8/DO1N-33	2.00	-	1.4	-	3.00	-	2.00	2.00

Sumber: Data Polman Bandung

Tabel 3.4 Data waktu proses operasi (lanjutan)

Job	No	Kode part	(dalam jam)							
			1	2	3	4	5	6	7	8
			BU	fr cnc	BOK	HARD-1	GS-1	GD	WCU	edm
2	38	PITCH GAUGE CNP -YL8/DO1N-34	2.00	-	-	3.00	1.10	-	-	-
	39	PITCH GAUGE CNP -YL8/DO1N-35	0.35	-	-	2.00	1.10	-	-	-
	40	PITCH GAUGE CNP -YL8/DO1N-36	1.25	-	-	2.00	1.10	-	-	-
	41	PITCH GAUGE CNP -YL8/DO1N-37	1.00	-	-	1.20	2.50	-	-	-
3	42	WIRE RING GAUGE YL 8-1	1.25	1.50	-	0.97	2.00	-	-	3.00
	43	WIRE RING GAUGE YL 8-2	0.75	-	-	0.75	1.10	-	-	-
	44	WIRE RING GAUGE YL 8-3	0.35	-	-	1.00	1.10	-	-	-
	45	WIRE RING GAUGE YL 8-4	1.00	-	-	1.00	2.25	-	-	-
	46	WIRE RING GAUGE YL 8-5	1.00	-	-	2.00	2.10	-	-	-
4	47	Assy jig fixture spline hub yl 8-1	1.25	-	1.10	2.50		2.25	2.50	-
	48	Assy jig fixture spline hub yl 8-2	2.35	-	0.25	2.00		2.00	-	-
	49	Assy jig fixture spline hub yl 8-3	1.25	-	0.25	2.00		1.20	-	-
	50	Assy jig fixture spline hub yl 8-4	1.25	-	0.25	2.00		0.97	-	-
	51	Assy jig fixture spline hub yl 8-5	3.00	-	0.25	2.00		0.75	-	-
	52	Assy jig fixture spline hub yl 8-6	2.20	-	-	3.00	1.10	-	-	-
	53	Assy jig fixture spline hub yl 8-7	1.50	-	-	3.00	1.25	-	-	2.45
	54	Assy jig fixture spline hub yl 8-8	0.75	-	-	2.50	1.10	-	-	-
	55	Assy jig fixture spline hub yl 8-9	1.25	-	-	2.00	1.25	-	-	-
5	56	PM 110407-1	1.25	7.00	-	-	-	-	-	-
	57	PM 110407-2	1.20	7.00	-	-	-	-	2.35	-
	58	PM 110407-3	0.97	7.00	-	-	-	-	-	4.3
	59	PM 110407-4	0.75	7.00	-	-	-	-	-	-
	60	PM 110407-5	3.00	7.00	3.00	-	-	-	-	-
	61	PM 110407-6	3.00	7.00	3.00	-	-	-	-	-
	62	PM 110407-7	1.00	-	-	-	2.35	-	-	3.2
	63	PM 110407-8	1.25	-	2.1	-	-	1.25	-	-
	64	PM 110407-9	2.35	-	-	-	1.00	-	2.00	-
	65	PM 110407-10	1.25	-	1.3	-	-	-	-	1.2
	66	PM 110407-11	1.25	2.2	-	0.75	-	-	-	3.2
	67	PM 110407-12	2.25	-	-	-	-	1.25	-	-
	68	PM 110407-13	2.25	-	2.3	-	-	-	2.00	1
	69	PM 110407-14	3.00	3.25	1.00	2.35	-	-	-	-
	70	PM 110407-15	3.00	-	1.00	-	1.25	-	-	-
	71	PM 110407-16	4.00	-	-	1.00	-	2.00	-	-
	72	PM 110407-17	4.23	-	3.25	-	1.20	-	-	-
	73	PM 110407-18	1.21	-	3.00	-	0.97	-	-	-
	74	PM 110407-19	2.60	-	-	1.00	0.75	-	-	-

Sumber: Data Polman Bandung

Tabel 3.4 Data waktu proses operasi (lanjutan)

Job	No	Kode part	(dalam jam)							
			1	2	3	4	5	6	7	8
			BU	fr cnc	BOK	HARD-1	GS-1	GD	WCU	edm
5	75	PM 110407-20	2.35	-	-	3.25	1.00	2.35	-	-
	76	PM 110407-21	1.25	3.25	-	-	1.00	-	1.25	-
	77	PM 110407-22	1.25	-	2.00	-	-	1.00	-	2.00
	78	PM 110407-23	2.25	-	1.20	-	3.25	-	1.20	-
	79	PM 110407-24	2.25	-	0.97	-	3.00	-	0.97	-
	80	PM 110407-25	3.00	-	0.75	-	-	1.00	0.75	-
	81	PM 110407-26	1.00	-	1.00	-	-	-	-	-
	82	PM 110407-27	1.25	-	1.25	-	-	-	2.00	-
	83	PM 110407-28	2.35	-	2.35	-	-	-	-	-
	84	PM 110407-29	2.10	-	-	1.25	-	3.5	3.2	-
	85	PM 110407-30	2.35	-	-	1.25	-	-	-	-
	86	PM 110407-31	1.25	-	-	2.25	-	1.00	-	-
	87	PM 110407-32	1.25	-	-	-	2.00	1.25	-	1.50
	88	PM 110407-33	2.25	-	-	-	2.00	-	-	1.50
	89	PM 110407-34	2.25	2.1	-	1	-	-	1.25	-
	90	PM 110407-35	3.00	-	2.5	-	-	2.2	-	-
	91	PM 110407-36	1.20	-	-	-	2.00	-	2.00	-
	92	PM 110407-37	0.97	2.1	-	1	-	-	2.00	-
	93	PM 110407-38	0.75	-	-	-	1.25	-	2.00	1.25
	94	PM 110407-39	1.00	7.00	0.50	-	-	-	-	-
	95	PM 110407-40	-	7.00	1.50	-	0.50	-	-	-
	96	PM 110407-41	-	7.00	1.50	-	1.50	-	-	-
	97	PM 110407-42	-	7.00	1.50	1.25	-	-	2.00	0.50
	98	PM 110407-43	-	7.00	1.50	-	-	1.25	-	1.50

Sumber: Data Polman Bandung

## BAB IV PENGOLAHAN DATA DAN ANALISIS

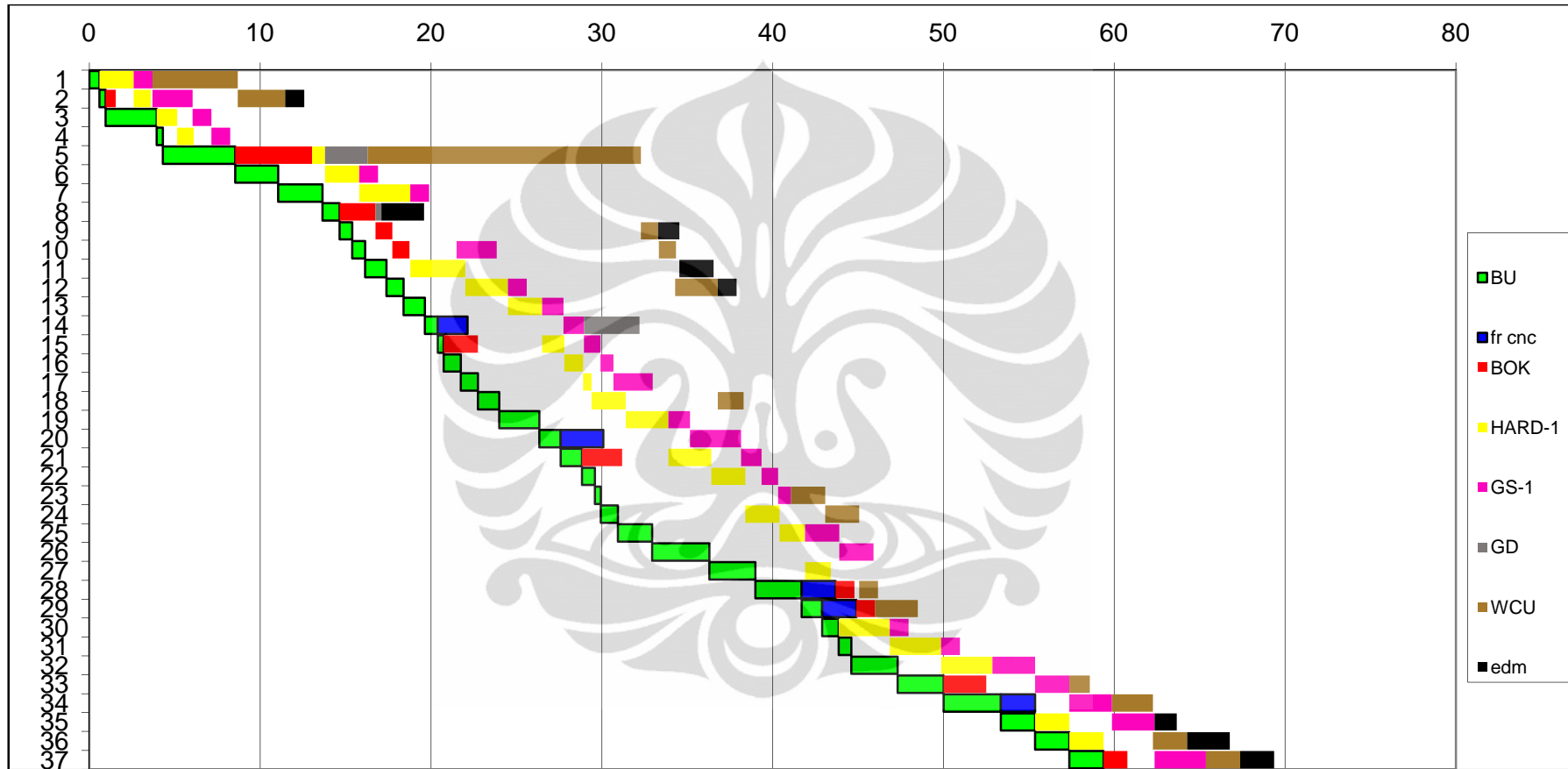
### 4.1 Solusi Awal Penjadwalan Polman Bandung

Penjadwalan *job* pada solusi awal ini juga dilakukan dengan metode yang paling sederhana dan mudah, yaitu mengurutkan semua *job* berdasarkan komponen yang akan diproduksi dari komponen ke-1 hingga komponen ke-98 atau *first come first serve* (FCFS).

Perhitungan waktu penyelesaian total semua komponen dilakukan dengan formula sederhana dan cara yang manual, yang selanjutnya juga dihitung dengan menggunakan piranti lunak yang dibuat. Hasil perhitungan dari solusi awal penjadwalan setiap komponen dari komponen 1 hingga ke-98 dapat dilihat pada tabel 4.1.

Ketika komponen 1 keluar dari mesin BU setelah diproses selama 0.60 jam, setelah selesai dikerjakan kemudian komponen 1 masuk ke mesin Hard-1 untuk diproses selama 2.0 jam hingga selesai pada jam ke-2.60. Ketika komponen 2 yang mempunyai waktu operasi 0.35 jam ingin memasuki mesin BU maka komponen 2 harus menunggu komponen 1 yang telah lebih dulu dikerjakan di mesin BU selesai dikerjakan dan akhirnya komponen 2 selesai dikerjakan pada jam ke-0.95. Sehingga dapat disimpulkan bahwa waktu masuknya suatu komponen ke dalam suatu mesin tertentu untuk diproses adalah sama dengan waktu maksimal antara waktu penyelesaian komponen tersebut pada mesin sebelumnya dengan waktu penyelesaian komponen pendahulunya pada mesin yang sama.

Formula di atas terus berlaku hingga *job* ke-98. Terlihat pada tabel 4.1 solusi penjadwalan awal, terdapat banyak waktu komponen yang menunggu untuk diproses (*idle time*). Dengan metode pencarian hasil optimal menggunakan algoritma *tabu search*, akan dicari penjadwalan yang mendekati optimal dengan *idle time* yang minimal, sehingga secara otomatis, waktu penyelesaian minimal untuk semua *job* pun akan tercapai ( $\leq 197.50$  jam).



Gambar 4.1 Gantt Chart Penjadwalan Job Shop Polman Bandung (Sampel komponen 1 s/d 37)

Sumber: Data Polman Bandung



Tabel 4.1 Jadwal awal POLMAN Bandung

Job	No	Kode Komponen		(dalam jam)							
				1	2	3	4	5	6	7	8
				BU	fr cnc	BOK	HARD-1	GS-1	GD	WCU	edm
1	1	PITCH GAUGE PP YL 8-1	0	0.60	-	-	2.60	3.70	-	8.70	-
	2	PITCH GAUGE PP YL 8-2	0.60	0.95	-	1.55	3.60	6.05	-	11.46	12.58
	3	PITCH GAUGE PP YL 8-3	0.95	3.95	-	-	5.15	7.15	-	-	-
	4	PITCH GAUGE PP YL 8-4	3.95	4.30	-	-	6.12	7.22	-	-	-
2	5	PITCH GAUGE CNP -YL8/DO1N-1	4.30	8.55	-	13.05	13.80	-	16.30	32.30	-
	6	PITCH GAUGE CNP -YL8/DO1N-2	8.55	11.05	-	-	15.80	16.90	-	-	-
	7	PITCH GAUGE CNP -YL8/DO1N-3	11.05	13.65	-	-	18.80	19.90	-	-	-
	8	PITCH GAUGE CNP -YL8/DO1N-4	13.65	14.65	-	16.75			17.10	-	19.60
	9	PITCH GAUGE CNP -YL8/DO1N-5	14.65	15.40	-	17.75			-	33.30	20.85
	10	PITCH GAUGE CNP -YL8/DO1N-6	15.40	16.15	-	18.75		22.25	-	34.30	-
	11	PITCH GAUGE CNP -YL8/DO1N-7	16.15	17.40	-	-	22.01		-	-	24.01
	12	PITCH GAUGE CNP -YL8/DO1N-8	17.40	18.40	-	-	24.51	25.61	-	36.80	37.90
	13	PITCH GAUGE CNP -YL8/DO1N-9	18.40	19.65	-	-	26.51	27.76	-	-	-
	14	PITCH GAUGE CNP -YL8/DO1N-10	19.65	20.40	22.15	-		28.96	32.21	-	-
	15	PITCH GAUGE CNP -YL8/DO1N-11	20.40	20.75	-	22.75	27.81	29.93	-	-	-
	16	PITCH GAUGE CNP -YL8/DO1N-12	20.75	21.75	-	-	28.91	30.68	-	-	-
	17	PITCH GAUGE CNP -YL8/DO1N-13	21.75	22.75	-	-	29.41	32.98	-	-	-
	18	PITCH GAUGE CNP -YL8/DO1N-14	22.75	24.00	-	-	31.41		-	38.30	-
	19	PITCH GAUGE CNP -YL8/DO1N-15	24.00	26.35	-	-	33.91	35.16	-	-	-
	20	PITCH GAUGE CNP -YL8/DO1N-16	26.35	27.60	30.10	-		38.16	-	-	-
	21	PITCH GAUGE CNP -YL8/DO1N-17	27.60	28.85	-	31.19	36.41	39.36	-	-	-
	22	PITCH GAUGE CNP -YL8/DO1N-18	28.85	29.60	-	-	38.41	40.33	-	-	-
	23	PITCH GAUGE CNP -YL8/DO1N-19	29.60	29.95	-	-		41.08	-	43.08	-
	24	PITCH GAUGE CNP -YL8/DO1N-20	29.95	30.95	-	-	40.41		-	45.08	-
	25	PITCH GAUGE CNP -YL8/DO1N-21	30.95	32.95	-	-	41.91	43.91	-	-	-
	26	PITCH GAUGE CNP -YL8/DO1N-22	32.95	36.30	-	-		45.91	-	-	-
	27	PITCH GAUGE CNP -YL8/DO1N-23	36.30	39.00	-	-	43.41		-	-	-
	28	PITCH GAUGE CNP -YL8/DO1N-24	39.00	41.70	43.70	44.80			-	46.18	-
	29	PITCH GAUGE CNP -YL8/DO1N-25	41.70	42.90	45.70	46.80			-	49.30	-
	30	PITCH GAUGE CNP -YL8/DO1N-26	42.90	43.87	-	-	46.87	47.97	-	-	-
	31	PITCH GAUGE CNP -YL8/DO1N-27	43.87	44.62	-	-	49.87	50.97	-	-	-
	32	PITCH GAUGE CNP -YL8/DO1N-28	44.62	47.32	-	-	52.87	55.37	-	-	-
	33	PITCH GAUGE CNP -YL8/DO1N-29	47.32	50.02	-	52.52		57.37	-	58.57	-
	34	PITCH GAUGE CNP -YL8/DO1N-30	50.02	53.37	55.37	-	-	59.87	-	62.27	-
	35	PITCH GAUGE CNP -YL8/DO1N-31	53.37	55.37	-	-	57.37	62.37	-	-	63.67
	36	PITCH GAUGE CNP -YL8/DO1N-32	55.37	57.37	-	-	59.37		-	64.27	66.77
	37	PITCH GAUGE CNP -YL8/DO1N-33	57.37	59.37	-	60.77		65.37	-	67.37	69.37

Sumber: Data Polman Bandung

**Tabel 4.1 Jadwal awal POLMAN Bandung (lanjutan)**

Job	No	Kode Komponen	(dalam jam)								
			1	2	3	4	5	6	7	8	
			BU	fr cnc	BOK	HARD-1	GS-1	GD	WCU	edm	
2	38	PITCH GAUGE CNP -YL8/DO1N-34	59.37	61.37	-	-	64.37	65.47	-	-	-
	39	PITCH GAUGE CNP -YL8/DO1N-35	61.37	61.72	-	-	66.37	67.47	-	-	-
	40	PITCH GAUGE CNP -YL8/DO1N-36	61.72	62.97	-	-	68.37	69.47	-	-	-
	41	PITCH GAUGE CNP -YL8/DO1N-37	62.97	63.97	-	-	69.57	72.07	-	-	-
3	42	WIRE RING GAUGE YL 8-1	63.97	65.22	66.72	-	70.54	74.07	-	-	77.07
	43	WIRE RING GAUGE YL 8-2	65.22	65.97	-	-	71.29	75.17	-	-	-
	44	WIRE RING GAUGE YL 8-3	65.97	66.32	-	-	72.29	76.27	-	-	-
	45	WIRE RING GAUGE YL 8-4	66.32	67.32	-	-	73.29	78.52	-	-	-
	46	WIRE RING GAUGE YL 8-5	67.32	68.32	-	-	75.29	80.62	-	-	-
4	47	Assy jig fixture spline hub yl 8-1	68.32	69.57	-	70.67	77.79	-	80.04	82.54	-
	48	Assy jig fixture spline hub yl 8-2	69.57	71.92	-	72.17	79.79	-	82.04	-	-
	49	Assy jig fixture spline hub yl 8-3	71.92	73.17	-	73.42	81.79	-	83.24	-	-
	50	Assy jig fixture spline hub yl 8-4	73.17	74.42	-	74.67	83.79	-	84.76	-	-
	51	Assy jig fixture spline hub yl 8-5	74.42	77.42	-	77.67	85.79	-	86.54	-	-
	52	Assy jig fixture spline hub yl 8-6	77.42	79.62	-	-	88.79	89.89	-	-	-
	53	Assy jig fixture spline hub yl 8-7	79.62	81.12	-	-	91.79	93.04	-	-	95.49
	54	Assy jig fixture spline hub yl 8-8	81.12	81.87	-	-	94.29	95.39	-	-	-
	55	Assy jig fixture spline hub yl 8-9	81.87	83.12	-	-	96.29	97.54	-	-	-
5	56	PM 110407-1	83.12	84.37	91.37	-	-	-	-	-	-
	57	PM 110407-2	84.37	85.57	92.57	-	-	-	-	94.92	-
	58	PM 110407-3	85.57	86.54	99.57	-	-	-	-	-	99.79
	59	PM 110407-4	86.54	87.29	106.57	-	-	-	-	-	-
	60	PM 110407-5	87.29	90.29	113.57	116.57	-	-	-	-	-
	61	PM 110407-6	90.29	93.29	120.57	123.57	-	-	-	-	-
	62	PM 110407-7	93.29	94.29	-	-	-	99.89	-	-	103.09
	63	PM 110407-8	94.29	95.54	-	125.67	-	-	126.92	-	-
	64	PM 110407-9	95.54	97.89	-	-	-	100.89	-	102.89	-
	65	PM 110407-10	97.89	99.14	-	99.19	-	-	-	-	104.29
	66	PM 110407-11	99.14	100.39	122.77	-	123.52	-	-	-	126.72
	67	PM 110407-12	100.39	102.64	-	-	-	-	-	-	-
	68	PM 110407-13	102.64	104.89	-	107.19	-	-	-	109.19	127.72
	69	PM 110407-14	104.89	107.89	126.02	127.02	125.87	-	-	-	-
	70	PM 110407-15	107.89	110.89	-	128.02	-	129.27	-	-	-
	71	PM 110407-16	110.89	114.89	-	-	126.87	-	128.92	-	-
	72	PM 110407-17	114.89	119.12	-	131.27	-	132.47	-	-	-
	73	PM 110407-18	119.12	120.33	-	134.27	-	135.24	-	-	-
	74	PM 110407-19	120.33	122.93	-	-	127.87	135.99	-	-	-

Sumber: Data Polman Bandung

**Tabel 4.1 Jadwal awal POLMAN Bandung (lanjutan)**

Job	No	Kode Komponen	(dalam jam)								
			1	2	3	4	5	6	7	8	
			BU	fr enc	BOK	HARD-1	GS-1	GD	WCU	edm	
5	75	PM 110407-20	122.93	125.28	-	-	131.12	136.99	139.34	-	-
	76	PM 110407-21	125.28	126.53	129.78	-	-	137.99	-	139.24	-
	77	PM 110407-22	126.53	127.78	-	136.27	-	-	140.34	-	142.34
	78	PM 110407-23	127.78	130.03	-	137.47	-	141.24	-	142.44	-
	79	PM 110407-24	130.03	132.28	-	138.44	-	144.24	-	145.21	-
	80	PM 110407-25	132.28	135.28	-	139.19	-	-	141.34	145.96	-
	81	PM 110407-26	135.28	136.28	-	140.19	-	-	-	-	-
	82	PM 110407-27	136.28	137.53	-	141.44	-	-	-	147.96	-
	83	PM 110407-28	137.53	139.88	-	143.79	-	-	-	-	-
	84	PM 110407-29	139.88	141.98	-	-	143.23	-	146.73	149.93	-
	85	PM 110407-30	141.98	144.33	-	-	145.58	-	-	-	-
	86	PM 110407-31	144.33	145.58	-	-	147.83	-	148.83	-	-
	87	PM 110407-32	145.58	146.83	-	-	-	148.83	150.08	-	151.58
	88	PM 110407-33	146.83	149.08	-	-	-	151.08	-	-	153.08
	89	PM 110407-34	149.08	151.33	153.43	-	154.43	-	-	155.68	-
	90	PM 110407-35	151.33	154.33	-	156.83	-	-	159.03	-	-
	91	PM 110407-36	154.33	155.53	-	-	-	157.53	-	159.53	-
	92	PM 110407-37	155.53	156.50	158.60	-	159.60	-	-	161.60	-
	93	PM 110407-38	156.50	157.25	-	-	-	158.78	-	163.60	164.85
94	PM 110407-39	157.25	158.25	165.25	165.75	-	-	-	-	-	
95	PM 110407-40		-	172.25	173.75	-	174.25	-	-	-	
96	PM 110407-41		-	179.25	180.75	-	182.25	-	-	-	
97	PM 110407-42		-	186.25	187.75	189.00	-	-	191.00	191.50	
98	PM 110407-43		-	193.25	194.75	-	-	196.00	-	197.50	

Sumber: Data Polman Bandung

## 4.2 Penyusunan Algoritma

Model penyelesaian penjadwalan *job shop* pada Polman Bandung menggunakan algoritma heuristik yaitu algoritma *tabu search*, seperti yang telah dijelaskan pada bab 2. Berdasarkan tahapan algoritma penyelesaian tersebut, dilakukan penerjemahan algoritma tersebut ke dalam bahasa program untuk mengolah data yang ada dengan bantuan VBA macro di Ms. Excel.

Data penelitian ini diolah dengan menggunakan bahasa pemrograman VBA pada Microsoft Excel. Bahasa pemrograman ini dipilih karena penggunaannya yang tidak memerlukan pemasangan software baru melainkan sudah ada di microsoft. Di samping itu ada kemudahan-kemudahan lainnya bila

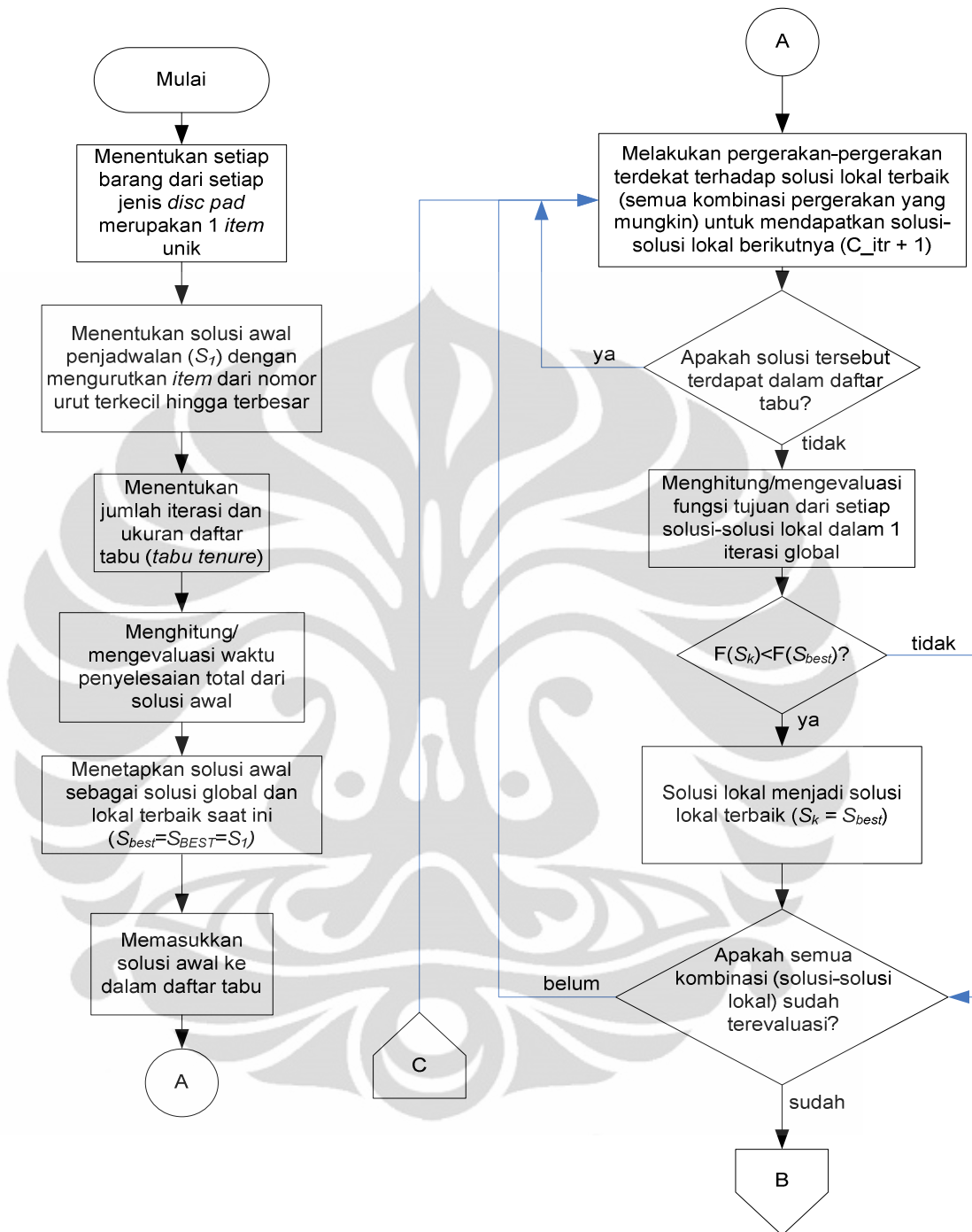
memerlukan pengolahan lebih lanjut. Microsoft Visual Basic for Applications (VBA) adalah sebuah turunan bahasa pemrograman Visual Basic yang dikembangkan oleh Microsoft dan dirilis pada tahun 1993. VBA adalah hasil kombinasi yang terintegrasi antara lingkungan pemrograman (Visual Basic Editor) dengan bahasa pemrograman (Visual Basic) yang memudahkan user untuk mendesain dan membangun program Visual Basic dalam aplikasi utama Microsoft Office, serta aplikasi-aplikasi tertentu. VBA didesain untuk melakukan beberapa tugas, seperti halnya membuat alternatif spesifikasi sebuah aplikasi seperti Microsoft Office atau Microsoft Visual Studio. Kegunaan VBA adalah mengotomatisasi pekerjaan-pekerjaan kompleks yang dilakukan secara berulang-ulang.

#### 4.2.1 Langkah-langkah Penyusunan Algoritma *Tabu Search*

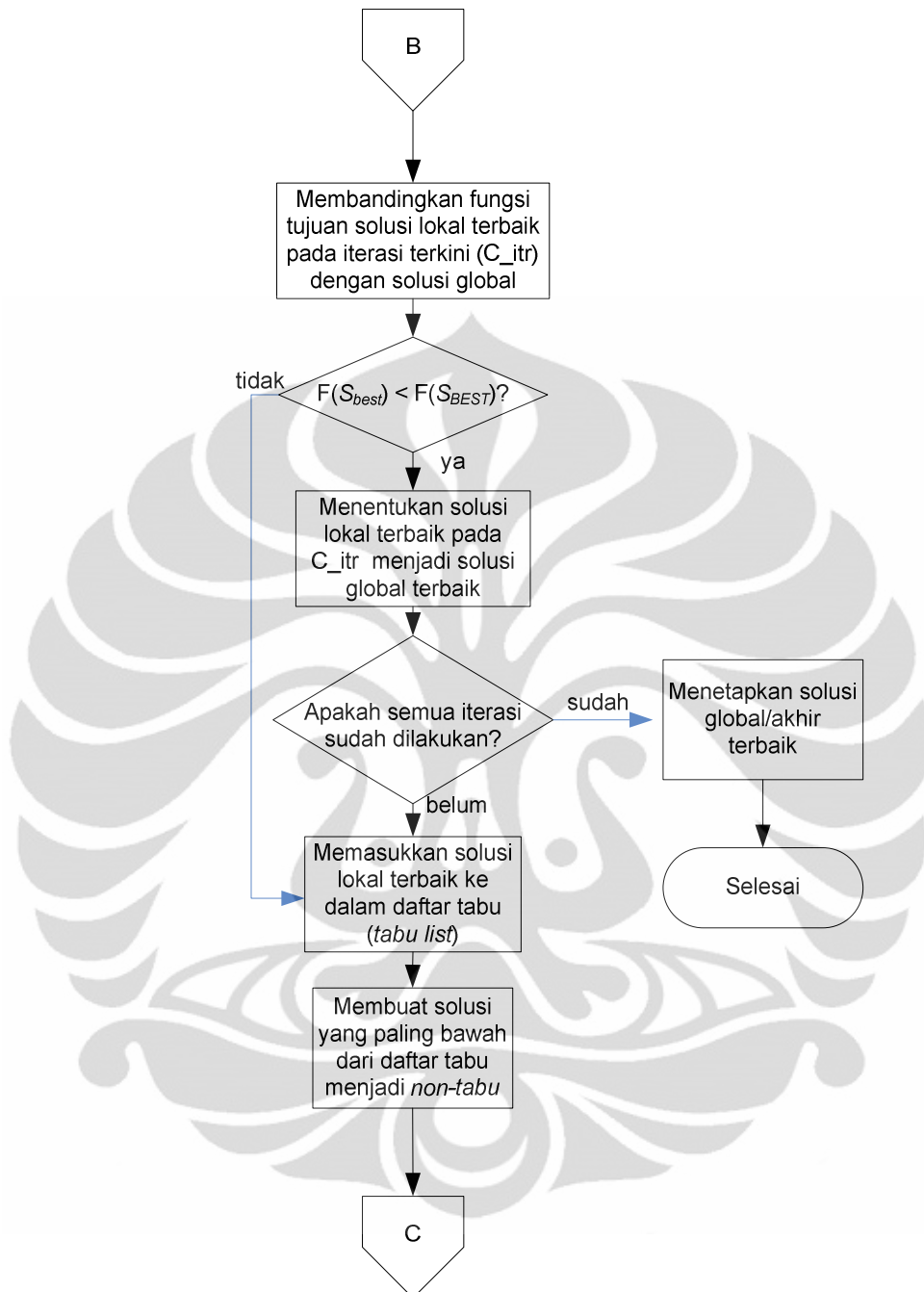
Algoritma *tabu search* yang digunakan untuk menyelesaikan permasalahan penjadwalan pada Polman Bandung ini dapat dilihat dengan lebih jelas pada gambar 4.1.

Diagram alir berikut menunjukkan algoritma *tabu search* secara detail yang dapat diuraikan sebagai berikut:

- Solusi awal penjadwalan. Hal ini telah dijabarkan pada sub bab Solusi Awal Penjadwalan Polman Bandung, yang secara singkat dilakukan dengan cara:
  - Menentukan setiap komponen yang akan di *assembly* dari setiap jenis menjadi 1 *item* unik;
  - Menentukan solusi awal penjadwalan ( $S_1$ ) dengan mengurutkan *item* dari nomor urut terkecil hingga terbesar atau *first come first serve* (job ke-1 hingga job ke-98).



**Gambar 4.2** Algoritma *Tabu Search* Penjadwalan *Job Shop* Polman Bandung



**Gambar 4.2 Algoritma Tabu Search Penjadwalan Job Shop Polman Bandung (Lanjutan)**

- Terdapat 2 parameter/variable yang digunakan dalam algoritma ini, yaitu:
  - Jumlah iterasi, dimana program komputer akan memproses dan mengulang algoritma hingga mendapatkan solusi terbaik dari sejumlah iterasi yang diinginkan tersebut;
  - Ukuran daftar tabu. Parameter ini sangat penting karena harus disesuaikan dengan jumlah data yang ada. Jika ukurannya terlalu kecil, maka akan mempercepat terjadinya *local optimal*, dan jika terlalu besar, algoritma ini tidak bekerja dengan efektif dan semakin lambat dalam mencari solusi terbaik.
- Mengevaluasi/menghitung fungsi tujuan, yaitu waktu penyelesaian total ( $C_{max}$ ) dari solusi awal. Gambaran perhitungan waktu penyelesaian total solusi awal ini telah dijelaskan pula pada sub bab Solusi Awal Penjadwalan Polman Bandung, sehingga didapatkan  $C_{max} = 197.50$  jam
- Menentukan solusi awal dengan  $C_{max}$ -nya menjadi solusi global dan lokal terbaik saat ini ( $S_{BEST} = S_{best} = S_1$ ), dan memasukkannya ke dalam daftar tabu, sehingga solusi awal ini tidak bisa dipakai selama ia masih berada dalam daftar tabu.
- Dari solusi lokal terbaik saat ini, mulai dilakukan pergerakan-pergerakan lokal, yaitu menentukan semua konfigurasi yang mungkin (iterasi-iterasi lokal). Setiap fungsi tujuan dari konfigurasi lokal ini dievaluasi dan didapatlah solusi-solusi lokal yang baru ( $F(S_k)$ ). Hal ini dilakukan di dalam setiap iterasi global. Jika terdapat konfigurasi yang terdapat dalam daftar tabu, maka konfigurasi itu diabaikan dan tidak dievaluasi.
- Kemudian, fungsi tujuan yang didapat dari setiap konfigurasi/solusi lokal dibandingkan dengan fungsi tujuan dari solusi lokal terbaik saat ini. Jika ia lebih kecil ( $F(S_k) < F(S_{best})$ ), maka solusi lokal tersebut menjadi solusi lokal terbaik saat ini ( $S_k = S_{best}$ ).

- Membandingkan fungsi tujuan dari solusi lokal terbaik yang baru dengan solusi global terbaik saat ini. Jika ia lebih kecil ( $F(S_{best}) < F(S_{BEST})$ ), maka solusi lokal terbaik tersebut menjadi solusi global terbaik saat ini  $F(S_{best}) = F(S_{BEST})$ .
- Memasukkan kembali solusi lokal terbaik saat ini ke dalam daftar tabu, dan mengeluarkan solusi yang terdapat pada daftar tabu paling bawah, sehingga statusnya menjadi *non tabu*.
- Setiap solusi lokal terbaik saat ini mengulangi prosedur pergerakan-pergerakan/iterasi-iterasi lokal meskipun solusi tersebut bukanlah solusi global terbaik. Hal ini terus dilakukan hingga mencapai iterasi global yang diinginkan.
- Setelah semua iterasi global tercapai, maka didapatkan solusi global terbaik, yaitu susunan penjadwalan terbaik dengan waktu penyelesaian total ( $C_{max}$ ) terkecil.

#### 4.2.2 Verifikasi dan Validasi Program

Setelah program dijalankan sesuai dengan prosedur yang ada dan telah mendapatkan hasil, maka belum cukup sampai di situ, program tersebut harus diverifikasi dan divalidasi. Tujuan dari tahap verifikasi adalah untuk membandingkan model program beserta parameternya dengan konsep yang diinginkan. Jika model tersebut menghasilkan keluaran yang berbeda-beda ketika parameternya diubah, maka program telah terverifikasi.

Setelah program menjalankan fungsinya pada data yang telah ada pada tahap verifikasi, dilakukan uji coba terhadap data yang lain (data *dummy*) dengan tujuan untuk mengetahui/memvalidasi program tersebut apakah memang berjalan sesuai dengan fungsinya. Keluaran program dari pengolahan data *dummy* ini kemudian dibandingkan dengan keluaran yang dihasilkan dari pengolahan secara manual.

Data *dummy* terdiri dari 5 *job*, yaitu dimisalkan produksi 5 *item* yang berbeda. Pada tabel 4.2 merupakan data *dummy* yang dimaksudkan.



#### 4.2.2.1 Hasil Perhitungan Manual

Perhitungan manual dimulai dengan inisialisasi solusi yang juga dilakukan dengan cara mengurutkan nomor *job* dari yang terkecil hingga terbesar, sehingga susunan solusi awal adalah sama dengan data *dummy* pada tabel 4.2. Selanjutnya perhitungan dilakukan dengan mengikuti langkah-langkah sebagai berikut:

1. Jumlah iterasi global ditentukan sebanyak 10 iterasi dan ukuran daftar tabu sebesar 2 solusi ( $\sqrt{5} \cong 2$ ). Perhitungan waktu penyelesaian total 5 *job* ini dilakukan secara manual dan dengan prosedur proses seperti yang dijelaskan pada sub bab solusi awal penjadwalan Polman Bandung. Hal ini dapat lebih jelas terlihat pada tabel 4.3.

Notasi yang digunakan:

- $X_{ijk}$  : waktu dimulainya job *i* part *j* pada mesin *k*
- $W_{ijk}$  : waktu yang diperlukan untuk mengerjakan job ke *i* komponen ke *j* pada mesin *k*, perhitungan waktu penyelesaian total 5 *part* ini menggunakan beberapa prosedur sebagai berikut:

Terlihat bahwa waktu penyelesaian total dari 5 *job* yang ada adalah 14.6 jam.

2. Menetapkan solusi awal tersebut menjadi solusi lokal dan global terbaik saat ini, sehingga  $F(S_{BEST}) = 14.6$  jam, serta memasukkan solusi awal ke dalam daftar tabu seperti terlihat pada tabel 4.4.
3. Melakukan pergerakan-pergerakan lokal terhadap solusi lokal terbaik saat ini, sehingga didapat solusi-solusi yang merupakan semua kombinasi penjadwalan yang mungkin dari solusi lokal terbaik tersebut. Fungsi tujuan dari setiap konfigurasi dievaluasi untuk mencari solusi lokal terbaik yang baru. Semua ini dilakukan dalam 1 iterasi global, dan digambarkan dengan lebih jelas pada tabel 4.5.

**Tabel 4.2 Data Dummy**

Job	No	Kode part	Rute proses (jam)					
				1	3	4	6	8
				BU	BOK	HARD-1	GD	WCU
4	1	Assy jig fixture spline hub yl 8-1		1.25	1.10	2.00	2.25	2.50
	2	Assy jig fixture spline hub yl 8-2		1.50	0.25	2.00	2.00	-
	3	Assy jig fixture spline hub yl 8-3		1.50	0.25	2.00	2.00	-
	4	Assy jig fixture spline hub yl 8-4		1.50	0.25	2.00	2.00	-
	5	Assy jig fixture spline hub yl 8-5		1.50	0.25	2.00	2.00	-

Sumber: Data Polman Bandung

**Tabel 4.3 Perhitungan Waktu Penyelesaian Total dari Data Dummy**

Job	No	Kode part	Rute proses (jam)					
				1	3	4	6	8
				BU	BOK	HARD-1	GD	WCU
4	1	Assy jig fixture spline hub yl 8-1	0.00	1.25	2.35	4.35	6.60	9.10
	2	Assy jig fixture spline hub yl 8-2	1.25	2.75	3.00	6.35	8.60	-
	3	Assy jig fixture spline hub yl 8-3	2.75	4.25	4.50	8.35	10.60	-
	4	Assy jig fixture spline hub yl 8-4	4.25	5.75	6.00	10.35	12.60	-
	5	Assy jig fixture spline hub yl 8-5	5.75	7.25	7.50	12.35	14.60	-

Sumber: Data Polman Bandung

**Tabel 4.4 Daftar Tabu**

	Daftar Tabu	
	1	2
	0	1
Susunan Komponen	1	-
	2	-
	3	-
	4	-
	5	-

Tabel 4.5 Iterasi Global Pertama

Part	Rute proses (jam)				
	1	3	4	6	8
1	1.25	2.35	4.35	6.60	9.10
2	2.75	3.00	6.35	8.60	-
3	4.25	4.50	8.35	10.60	-
4	5.75	6.00	10.35	12.60	-
5	7.25	7.50	12.35	14.60	-
<b>Kombinasi 1</b>					
4	1.5	1.75	3.75	5.75	-
1	2.75	3.85	5.85	8.1	10.6
3	4.25	4.5	7.85	10.1	-
5	5.75	6	9.85	12.1	-
2	7.25	7.5	11.85	14.1	-
<b>Kombinasi 2</b>					
5	1.5	1.75	3.75	5.75	-
1	2.75	3.85	5.85	8.1	10.6
2	4.25	4.5	7.85	10.1	-
3	5.75	6	9.85	12.1	-
4	7.25	7.5	11.85	14.1	-
<b>Kombinasi 3</b>					
1	1.25	2.35	4.35	6.6	9.1
3	2.75	3	6.35	8.6	-
2	4.25	4.5	8.35	10.6	-
4	5.75	6	10.35	12.6	-
5	7.25	7.5	12.35	14.6	-
<b>Kombinasi 4</b>					
5	1.5	1.75	3.75	5.75	-
3	3	3.25	5.75	7.75	-
2	4.5	4.75	7.75	9.75	-
4	6	6.25	9.75	11.75	-
1	7.25	8.35	11.75	14	16.5
<b>Kombinasi 5</b>					
1	1.25	2.35	4.35	6.6	9.1
2	2.75	3	6.35	8.6	-
5	4.25	4.5	8.35	10.6	-
3	5.75	6	10.35	12.6	-
4	7.25	7.5	12.35	14.6	-
<b>Kombinasi 6</b>					
1	1.25	2.35	4.35	6.6	9.1
4	2.75	3	6.35	8.6	-
3	4.25	4.5	8.35	10.6	-
5	5.75	6	10.35	12.6	-
2	7.25	7.5	12.35	14.6	-
<b>Kombinasi 7</b>					
3	1.5	1.75	3.75	5.75	-
4	3	3.25	5.75	7.75	-
5	4.5	4.75	7.75	9.75	-
2	6	6.25	9.75	11.75	-
1	7.25	8.35	11.75	14	16.5
<b>Kombinasi 8</b>					
1	1.25	2.35	4.35	6.6	9.1
3	2.75	3	6.35	8.6	-
2	4.25	4.5	8.35	10.6	-
5	5.75	6	10.35	12.6	-
4	7.25	7.5	12.35	14.6	-
<b>Kombinasi 9</b>					
4	1.5	1.75	3.75	5.75	-
3	3	3.25	5.75	7.75	-
2	4.5	4.75	7.75	9.75	-
1	5.75	6.85	9.75	12	14.5
5	7.25	7.5	11.75	14	-
<b>Kombinasi 10</b>					
2	1.5	1.75	3.75	5.75	-
3	3	3.25	5.75	7.75	-
5	4.5	4.75	7.75	9.75	-
4	6	6.25	9.75	11.75	-
1	7.25	8.35	11.75	14	16.5

Terlihat bahwa terdapat 10 kombinasi yang mungkin dari solusi awal, dan semua kombinasi ini tidak terdapat di dalam daftar tabu. Dalam iterasi global pertama ini, solusi lokal terbaik atau *makespan* ( $C_{max}$  terkecil) adalah kombinasi ke-1, dengan urutan pengerjaan *part* 4-1-3-5-2, dan dengan  $C_{max}(F(S_{best})) = 14.1$  jam

4. Solusi lokal terbaik tersebut dimasukkan ke dalam daftar tabu.
5. Fungsi tujuan dari solusi lokal terbaik saat ini dibandingkan dengan fungsi tujuan dari solusi global terbaik saat ini. Karena  $F(S_{best}) < F(S_{BEST})$ , maka solusi lokal terbaik dari iterasi pertama ini menjadi solusi global terbaik terbaru ( $F(S_{BEST}) = F(S_{best})$ ).
6. Langkah 3-5 terus diulangi hingga mencapai iterasi ke-10. Sehingga rekapitulasi dari 10 iterasi global digambarkan pada tabel 4.6 Begitu juga dengan daftar tabu yang terus diperbaharui hingga iterasi ke-10 dapat dilihat pada tabel 4.7.

Terlihat pada tabel rekapitulasi bahwa setelah mencapai 10 iterasi, solusi global terbaik atau solusi akhir terbaik adalah penjadwalan produksi dengan susunan komponen 5-4-3-1-2 atau 4-2-1-3-5, dengan waktu penyelesaian total ( $C_{max}$ ) = 14 jam.

**Tabel 4.6 Rekapitulasi 10 Iterasi Hingga Mendapatkan Solusi Global Akhir**

	iterasi 0	iterasi 1	iterasi 2	iterasi 3	iterasi 4	iterasi 5	iterasi 6	iterasi 7	iterasi 8	iterasi 9	iterasi 10
Solusi lokal terbaik	1	4	4	1	3	5	4	3	4	1	5
	2	1	3	4	5	4	5	5	2	2	1
	3	3	5	3	2	1	1	2	1	4	2
	4	5	1	2	4	3	2	1	3	3	3
	5	2	2	5	1	2	3	4	5	5	4
$F_{S_{best}}$ (jam)	14.6	14.1	14.5	14.6	16.5	14	14.1	14.5	14	14.6	14.1
$F_{S_{BEST}}$ (jam)	14.6	14.1	14.1	14.1	14.1	14	14	14	14	14	14

#### 4.2.2.2 Hasil Perhitungan dengan Piranti Lunak

Hasil *run* program menunjukkan urutan jadwal terbaik ialah komponen 4-2-1-3-5, dengan waktu penyelesaian total ( $C_{max}$ ) = 14 jam. Jika membandingkan

hasil tersebut dengan hasil perhitungan secara manual, maka didapat nilai yang sama. Dengan demikian program telah tervalidasi.

**Tabel 4.7 Daftar Tabu Hingga Iterasi ke-10**

		Daftar Tabu	
		1	2
Iterasi		9	10
Susunan Komponen		5	4
		4	2
		1	1
		3	3
		2	5

### 4.3 Pengolahan Data Dan Analisis

Sebelum dilakukan pengolahan data, ada beberapa asumsi yang ditetapkan yaitu:

- Release date* semua *job* adalah 0, artinya semua *job* bisa dikerjakan kapan saja dan memiliki prioritas yang sama untuk menjadi *job* yang pertama kali dikerjakan.
- Semua material yang dibutuhkan dalam proses produksi dianggap tersedia tepat waktu dan waktu perpindahan material diabaikan dalam perhitungan.
- Data besarnya waktu proses dianggap tetap.
- Besarnya waktu *setup* mesin sudah termasuk dalam waktu proses.

Indikator penyelesaian masalah ialah urutan yang menghasilkan waktu penyelesaian total terkecil untuk semua *job*.

#### 4.3.1 Input Data

Data utama yang diperlukan untuk mendapatkan hasil urutan *job* dengan waktu penyelesaian produksi terkecil dengan menggunakan algoritma *tabu search* ini merupakan data yang telah dijelaskan pada Bab 3, yaitu aliran proses produksi *pitch gauge PP YL 8*, *pitch gauge PP YL 8/DO1N*, *wire ring gauge YL 8*, *assy jig fixture Spline hub yl8*, PM 110407, data jumlah mesin, dan data waktu proses setiap mesin.

#### 4.3.2 Parameter

Seperti telah diketahui bahwa terdapat 2 parameter yang digunakan dalam algoritma ini, dimana 2 parameter ini tidak bergantung satu dengan yang lainnya.

Dua parameter tersebut adalah:

1. Jumlah iterasi, dimana program komputer akan memproses dan mengulang algoritma hingga mendapatkan solusi terbaik dari sejumlah iterasi yang diinginkan tersebut. Jumlah iterasi ini disebut juga iterasi global. Hal ini disebabkan karena di dalam 1 iterasi global terdapat sejumlah iterasi lokal yang terdiri dari kombinasi-kombinasi atau pergerakan-pergerakan lokal dari solusi lokal terbaik dari iterasi global sebelumnya. Sedangkan, untuk jumlah iterasi lokal, hal itu tidak perlu ditentukan, karena program komputer yang dibuat akan melakukan semua kombinasi atau pergerakan dari suatu urutan *job* terbaik yang berasal dari iterasi global sebelumnya.
2. Ukuran daftar tabu. Parameter ini sangat penting karena harus disesuaikan dengan jumlah data yang ada. Jika ukurannya terlalu kecil, maka akan mempercepat terjadinya *local optima*, dan jika terlalu besar, algoritma ini tidak bekerja dengan efektif dan semakin lambat dalam mencari solusi terbaik.

#### 4.3.3 Hasil Penjadwalan dengan Algoritma *Tabu Search*

Pada pengujian algoritma *Tabu Search* digunakan level kombinasi parameter sebagai berikut:

- Panjang solusi tetangga : 50 dan 100
- Jumlah iterasi : 50, 200, 500 dan 1000
- Ukuran daftar tabu : 10

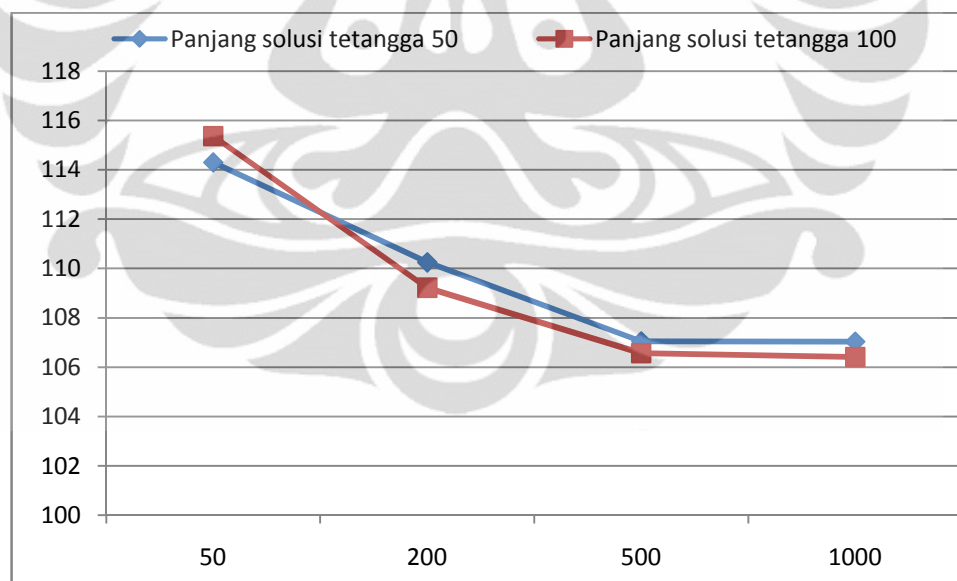
Hasil pengujian dengan algoritma *Tabu Search* pada dengan panjang solusi tetangga 50 dan 100 disajikan berturut-turut pada tabel 4.8 dan tabel 4.9 berikut ini.

**Tabel 4.8 Hasil pengujian algoritma dengan panjang solusi tetangga 50**

Iterasi maksimum	Panjang solusi tetangga					
	50					
	Run 1	Run 2	Run 3	Run 4	Run 5	AVG
50	112.67	111.05	118.18	114.1	115.52	114.30
200	106.27	117.3	106.65	104.95	116.1	110.25
500	103.86	102.55	112.34	109.65	106.85	107.05
1000	108.25	106.34	104.8	108	107.8	107.04

**Tabel 4.9 Hasil pengujian algoritma dengan panjang solusi tetangga 100**

Iterasi maksimum	Panjang solusi tetangga					
	100					
	Run 1	Run 2	Run 3	Run 4	Run 5	AVG
50	116.7	116.79	111.64	117.98	113.76	115.37
200	110.27	110.15	110.15	105.15	110.39	109.22
500	103.3	104.65	108.06	109.9	106.9	106.56
1000	104.75	106.05	107.05	107.03	107.15	106.41

**Gambar 4.3 Grafik hasil run algoritma *tabu search***

Terlihat pada hasil *running* program, *minimal makespan* yang didapat adalah sebesar 106.41 jam, dimana terjadi perubahan sebesar 53,87 % terhadap *makespan* solusi penjadwalan awal (*initial solution*). *Running* program akan dilakukan sebanyak 1000 iterasi dan panjang solusi tetangga 100, karena dari hasil

pengujian menunjukkan tidak adanya perubahan yang terlalu signifikan dari iterasi 500 menuju iterasi 1000. Jika dilakukan penambahan iterasi lagi maka akan menambah waktu *running* program secara signifikan.

#### 4.3.4 Metode dan *Running* Program

Karena algoritma ini terkenal dengan logika yang sederhana tetapi dengan kerumitan yang tinggi, sehingga algoritma ini tidak dikembangkan ke dalam bahasa program oleh para ahli, seperti layaknya algoritma Genetika, dimana VBA telah memiliki petunjuk *source code* dan contoh-contoh aplikasi didalamnya. Hal ini menyebabkan penyusunan algoritma *tabu search* ke dalam bahasa macro VBA dapat dikatakan merupakan pembuatan program murni. Metode ini terkenal dengan metode yang juga menampilkan hasil mendekati terbaik.

Berdasarkan metode yang terdapat pada algoritma *tabu search* ini, algoritma tersebut sangat layak untuk dikembangkan lebih lanjut dan diterapkan untuk berbagai aplikasi, khususnya pada bidang riset operasi. Hal ini disebabkan karena penyelesaian permasalahan penjadwalan menggunakan algoritma ini memiliki kelebihan seperti yang telah dijelaskan pada Bab II, antara lain:

- Adanya iterasi-iterasi lokal dari setiap iterasi global. Adapun tujuan dari iterasi-iterasi lokal ini adalah untuk mengevaluasi setiap fungsi tujuan dari semua kombinasi yang mungkin dari solusi lokal terbaik pada iterasi global sebelumnya.
- Selanjutnya, terdapat daftar tabu dengan ukuran tertentu untuk menyimpan setiap solusi lokal terbaik sepanjang waktu tertentu. Hal inilah yang akan mencegah terjadinya *local optima* ketika solusi tersebut mungkin saja terevaluasi kembali pada beberapa iterasi global berikutnya, maka algoritma ini menjamin hasil yang mendekati optimal.

Dapat terlihat bahwa *minimal makespan* yang didapat adalah sebesar 106.41 jam, sehingga terjadi penurunan *makespan* sebesar 53,87 %.



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil analisis dari permasalahan penjadwalan *job-shop* pada POLMAN Bandung menggunakan algoritma *tabu search* dengan bantuan bahasa pemrograman VBA diperoleh model penjadwalan produksi *job shop*.

Penjadwalan dengan metode algoritma TS memiliki waktu penyelesaian total seluruh *job* sebesar 106.41 jam, jika dibandingkan dengan jadwal awal sebesar 197.50 jam, mengalami penurunan sebesar 53,87 %.

#### **5.2 Saran**

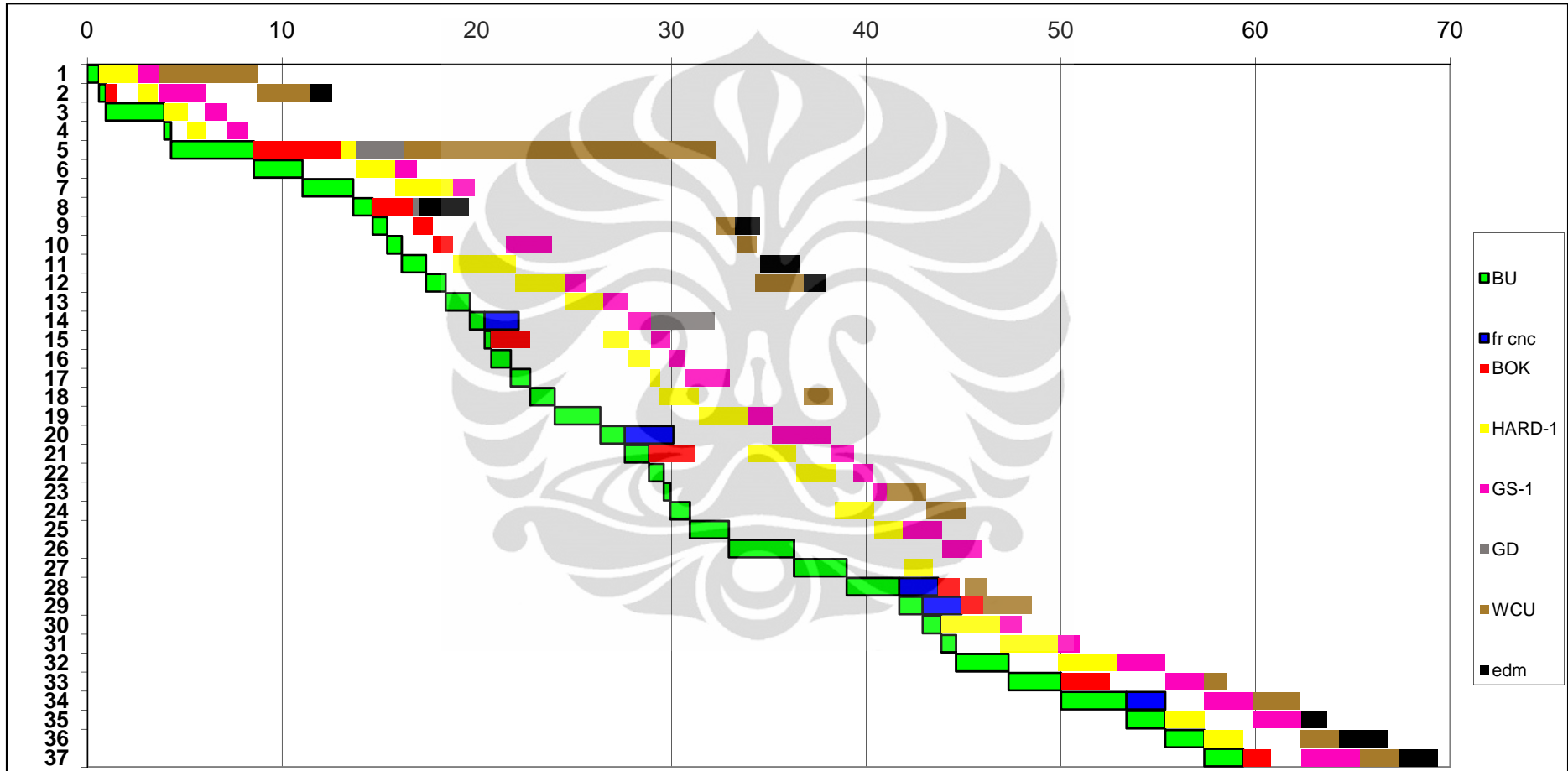
Banyak pengembangan yang dapat dilakukan terhadap penelitian ini di masa yang akan datang. Pengembangan penelitian perlu dilakukan untuk keberlanjutan keilmuan agar manfaatnya dapat dirasakan oleh seluruh civitas akademika dalam menghasilkan riset-riset yang berkualitas, serta pihak industri dalam memecahkan persoalan operasional yang ditemui di lapangan. Adapun beberapa area pengembangan penelitian yang dapat dilakukan di masa mendatang, antara lain sebagai berikut.

- Penggunaan algoritma metaheuristik lainnya yang mungkin akan menghasilkan performa lebih baik, seperti Memetic Algorithm, Simulated Annealing, Ant Colony Optimization, atau Particle Swarm Optimization.
- Penentuan parameter algoritma yang nondeterministik, sehingga performa algoritma dapat lebih dioptimalkan
- Pengembangan masalah penjadwalan agar semakin mencerminkan kondisi aktual di lapangan, seperti kendala ketersediaan mesin, keterbatasan kapasitas buffer, waktu transportasi material, waktu proses yang nondeterministik (stokastik), dan lain sebagainya.
- Perluasan permasalahan dengan mempertimbangkan multi fungsi tujuan.

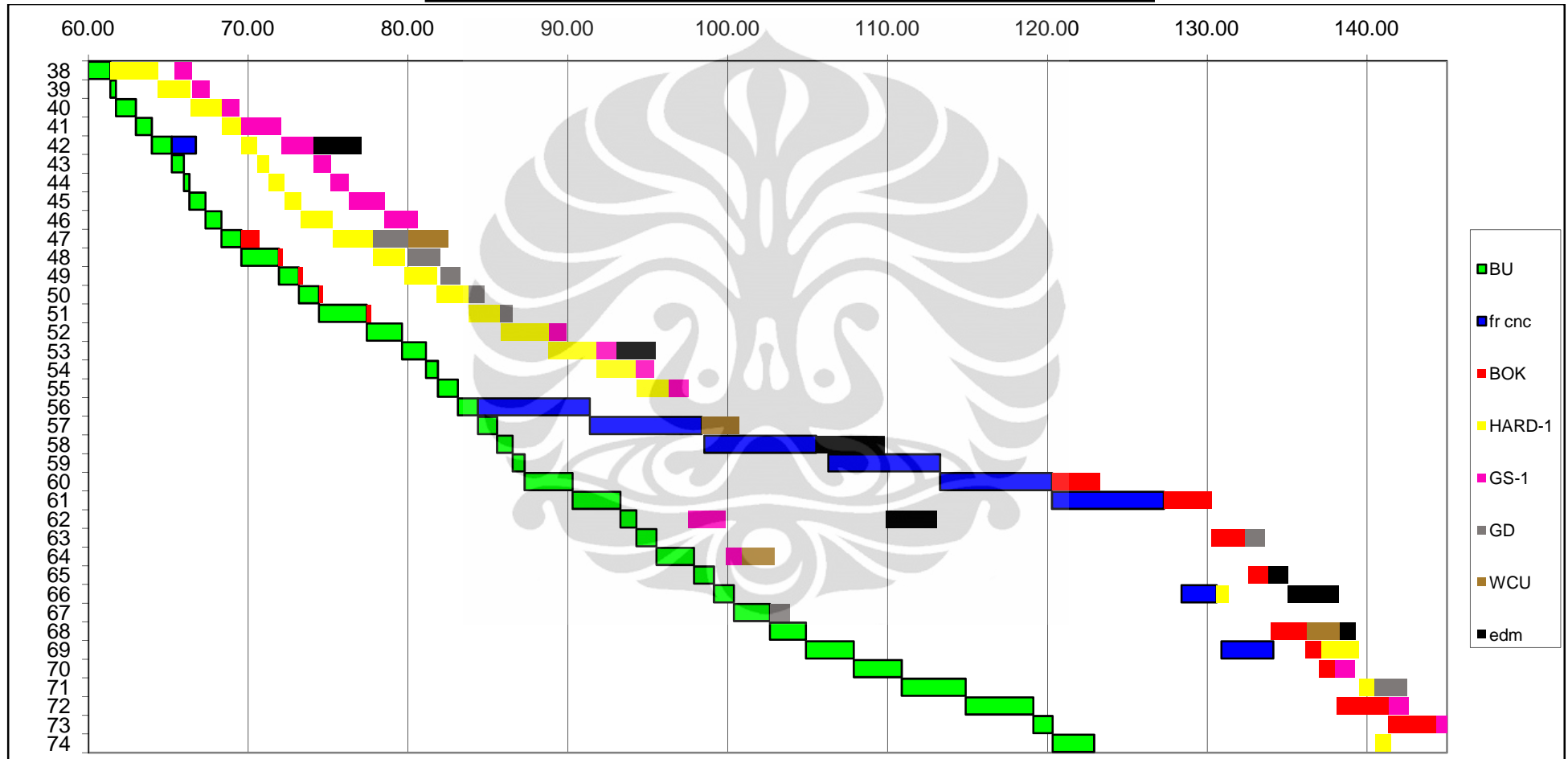
## DAFTAR PUSTAKA

- Adam, J. E. E. and Ebert, R. J.,1992. Production and Operation Management. New Jersey, Prentice Hall
- Albert Jones dan Luis Rabelo, 1998 – Jones, A. and Rabelo, L. C. (1998). “Survey of Job Shop Scheduling Techniques.”
- Applegate, D. and Cook, W (1991). “A Computational Study of the Job-Shop Sceduling Problem.” ORSA Journal on Computing.
- Betrianis dan Teguh, A. T. (2003). “Penerapan Algoritma Tabu Search dalam Penjadwalan Job Shop.” Makara, Teknologi, Vol. 7 No. 3.
- Fryman, M. A., 2002- Mark A. Fryman, Quality and Process Improvement, USA, 2002, p.320
- Montgomery, D. C. , 1996. Design and Analysis of Experiments. New York:John Wiley & Sons.
- Nahmias, S. ,1997. Production and Operation Analysis. New York, McGraw-Hill.
- Nowicki, E. and Smutnicki, C. , 1996. "A Fast Taboo Search Algorithm for the Job Shop Problem." Technical University of Wroctaw, Institute of Engineering Cybernetics, ul. Janiszewskiego.
- Panggabean, H. P. , 2005. “Penjadwalan Job Shop Statik dengan Algoritma Tabu Search.” Integral, Vol 10 No 1.
- Pinedo, M & Chao, 1999, “Operation Scheduling with Applications in Manufacturing and Services”, McGraw-Hill, New York.

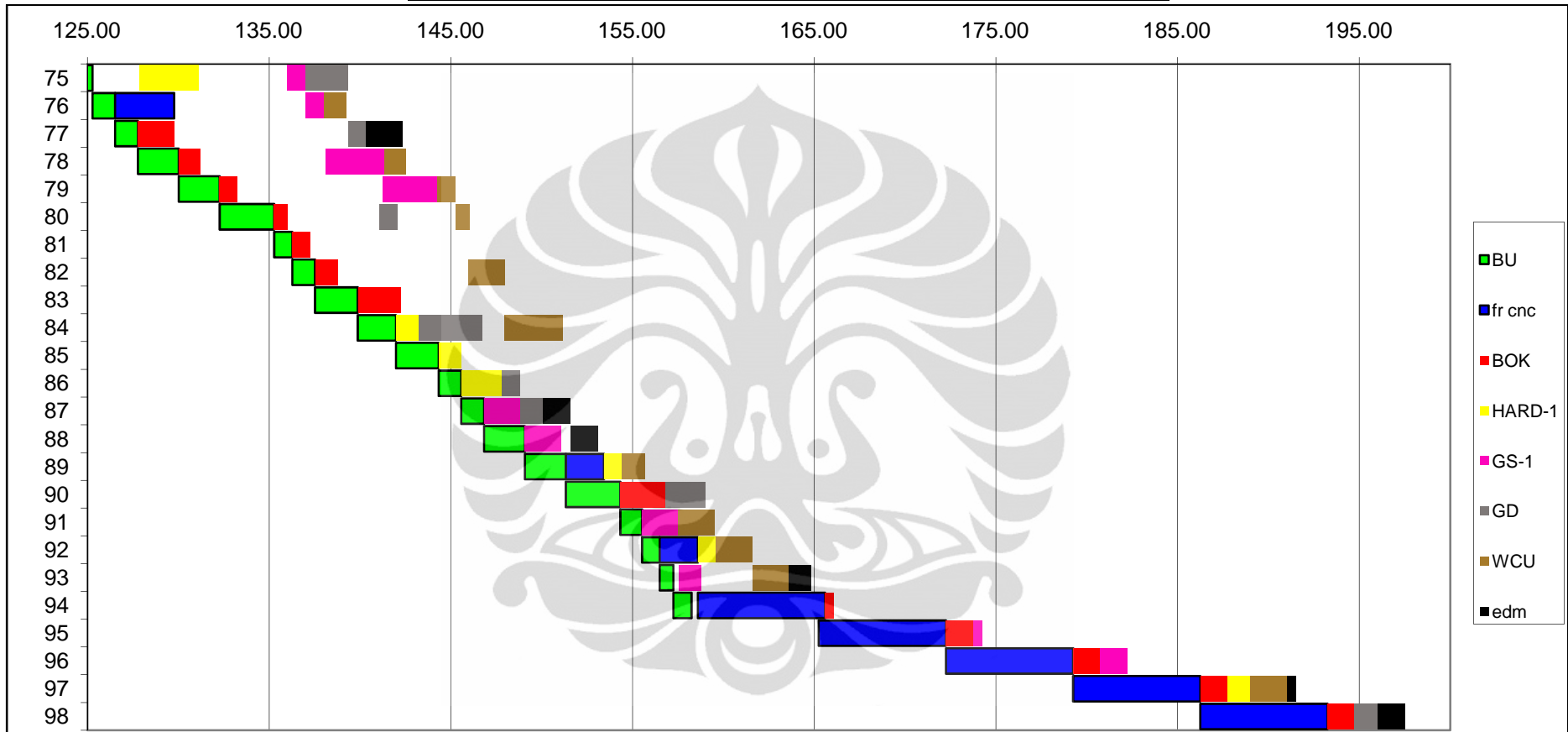
Lampiran 1: Gantt Chart Penjadwalan Polman



Lampiran 1: Gantt Chart Penjadwalan Polman (lanjutan)



Lampiran 1: Gantt Chart Penjadwalan Polman (lanjutan)



**Lampiran 2: Hasil Run Program Solusi Akhir**

Nomor Mesin	Mesin Ke	Item	Urutan pengerjaan															
1	1	Job number	82	93	77	10	76	67	1	8	33	91	41	80	65	85	50	53
		Start Time	0	1.25	2	3.25	4	5.25	7.5	8.1	9.1	11.8	13	14	17	18.25	20.6	21.85
		End Time	1.25	2	3.25	4	5.25	7.5	8.1	9.1	11.8	13	14	17	18.25	20.6	21.85	23.35
		Job number	2	79	78	88	42	29	15	38	62	18	84	27	17	28	24	73
		Start Time	23.35	23.7	25.95	28.2	30.45	31.7	32.9	33.25	35.25	36.25	37.5	39.6	42.3	43.3	46	47
		End Time	23.7	25.95	28.2	30.45	31.7	32.9	33.25	35.25	36.25	37.5	39.6	42.3	43.3	46	47	48.21
		Job number	14	40	21	58	16	35	26	32	69	48	61	20	3	46	89	86
		Start Time	48.21	48.96	50.21	51.46	52.43	53.43	55.43	58.78	61.48	64.48	66.83	69.83	71.08	74.08	75.08	77.33
		End Time	48.96	50.21	51.46	52.43	53.43	55.43	58.78	61.48	64.48	66.83	69.83	71.08	74.08	75.08	77.33	78.58
		Job number	39															
		Start Time	78.58															
		End Time	78.93															
1	2	Job number	12	9	22	43	7	68	81	55	11	34	30	52	83	87	63	54
		Start Time	0	1	1.75	2.5	3.25	5.85	8.1	9.1	10.35	11.6	14.95	15.92	18.12	20.47	21.72	22.97
		End Time	1	1.75	2.5	3.25	5.85	8.1	9.1	10.35	11.6	14.95	15.92	18.12	20.47	21.72	22.97	23.72
		Job number	45	57	56	4	51	71	49	37	36	59	70	25	13	66	74	94
		Start Time	23.72	24.72	25.92	27.17	27.52	30.52	34.52	35.77	37.77	39.77	40.52	43.52	45.52	46.77	48.02	50.62
		End Time	24.72	25.92	27.17	27.52	30.52	34.52	35.77	37.77	39.77	40.52	43.52	45.52	46.77	48.02	50.62	51.62
		Job number	44	75	47	23	72	64	5	90	31	6	92	60	19			
		Start Time	51.62	51.97	54.32	55.57	55.92	60.15	62.5	66.75	69.75	70.5	73	73.97	76.97			
		End Time	51.97	54.32	55.57	55.92	60.15	62.5	66.75	69.75	70.5	73	73.97	76.97	79.32			
2	1	Job number	97	76	96	34	98	57	29	56	42	59	66	14	28	58	95	61
		Start Time	0	7	10.25	17.25	19.25	26.25	33.25	35.25	42.25	43.75	50.75	52.95	54.7	56.7	63.7	70.7
		End Time	7	10.25	17.25	19.25	26.25	33.25	35.25	42.25	43.75	50.75	52.95	54.7	56.7	63.7	70.7	77.7

**Lampiran 2: Hasil Run Program Solusi Akhir (lanjutan)**

Nomor Mesin	Mesin Ke	Item	Urutan pengerjaan															
2	1	Job number	20	69	60	89	92	94										
		Start Time	77.7	80.2	83.45	90.45	92.55	94.65										
		End Time	80.2	83.45	90.45	92.55	94.65	101.65										
3	1	Job number	82	9	77	10	68	81	8	33	80	65	83	50	63	98	78	97
		Start Time	1.25	2.5	3.5	5.5	8.1	10.4	11.4	13.5	17	18.25	20.47	22.82	23.07	26.25	28.2	29.4
		End Time	2.5	3.5	5.5	6.5	10.4	11.4	13.5	16	17.75	19.55	22.82	23.07	25.17	27.75	29.4	30.9
		Job number	79	51	29	15	37	2	70	49	73	16	28	47	21	72	5	90
		Start Time	30.9	31.87	35.25	36.35	38.35	39.75	43.52	44.52	48.21	53.43	56.7	57.8	58.9	61.24	66.75	71.25
		End Time	31.87	32.12	36.35	38.35	39.75	40.35	44.52	44.77	51.21	56.43	57.8	58.9	61.24	64.49	71.25	73.75
		Job number	95	96	48	61	69	60	94									
		Start Time	73.75	75.25	76.75	77.7	83.45	90.45	101.65									
		End Time	75.25	76.75	77	80.7	84.45	93.45	102.15									
4	1	Job number	12	43	22	7	1	55	11	41	30	85	52	53	45	50	4	54
		Start Time	1	3.5	4.25	6.25	9.25	11.25	13.25	16.46	17.66	20.66	21.91	24.91	27.91	28.91	30.91	31.88
		End Time	3.5	4.25	6.25	9.25	11.25	13.25	16.46	17.66	20.66	21.91	24.91	27.91	28.91	30.91	31.88	34.38
		Job number	97	38	51	71	2	84	42	18	36	15	27	74	49	66	44	13
		Start Time	34.38	35.63	38.63	40.63	41.63	42.63	43.88	44.85	46.85	48.85	50.15	51.65	52.65	54.65	55.4	56.4
		End Time	35.63	38.63	40.63	41.63	42.63	43.88	44.85	46.85	48.85	50.15	51.65	52.65	54.65	55.4	56.4	58.4
		Job number	16	17	24	40	75	25	21	35	32	6	47	5	3	39	46	48
		Start Time	58.4	59.5	60	62	64	67.25	68.75	71.25	73.25	76.25	78.25	80.75	81.5	82.7	84.7	86.7
		End Time	59.5	60	62	64	67.25	68.75	71.25	73.25	76.25	78.25	80.75	81.5	82.7	84.7	86.7	88.7
		Job number	31	89	69	19	86	92										
		Start Time	88.7	92.55	93.55	95.9	98.4	100.65										
		End Time	91.7	93.55	95.9	98.4	100.65	101.65										

**Lampiran 2: Hasil Run Program Solusi Akhir (lanjutan)**

Nomor Mesin	Mesin Ke	Item	Urutan pengerjaan															
5	1	Job number	93	12	43	22	10	7	76	1	55	91	33	41	30	34	87	52
		Start Time	2	3.5	4.6	6.25	7.22	9.57	10.67	11.67	13.25	14.5	16.5	18.5	21	22.1	24.6	26.6
		End Time	3.25	4.6	5.7	7.22	9.57	10.67	11.67	12.77	14.5	16.5	18.5	21	22.1	24.6	26.6	27.7
		Job number	53	45	78	4	54	62	38	79	88	42	70	37	15	73	74	2
		Start Time	27.91	29.16	31.41	34.66	35.76	36.86	39.21	40.31	43.31	45.31	47.31	48.56	51.56	52.53	53.5	54.25
		End Time	29.16	31.41	34.66	35.76	36.86	39.21	40.31	43.31	45.31	47.31	48.56	51.56	52.53	53.5	54.25	56.6
		Job number	44	14	23	16	13	17	64	40	72	75	26	25	21	35	32	6
		Start Time	56.6	57.7	58.9	59.65	60.4	61.65	63.95	64.95	66.05	67.25	68.25	70.25	72.25	73.45	76.25	78.75
		End Time	57.7	58.9	59.65	60.4	61.65	63.95	64.95	66.05	67.25	68.25	70.25	72.25	73.45	75.95	78.75	79.85
		Job number	20	3	39	46	96	31	95	19								
		Start Time	80.2	83.2	84.7	86.7	88.8	91.7	92.8	98.4								
		End Time	83.2	84.3	85.8	88.8	90.3	92.8	93.3	99.65								
6	1	Job number	77	67	8	80	63	87	50	98	51	71	84	49	14	75	47	5
		Start Time	5.5	7.5	13.5	17.75	25.17	26.6	30.91	31.88	40.63	41.63	43.88	54.65	58.9	68.25	80.75	83
		End Time	6.5	8.75	13.85	18.75	26.42	27.85	31.88	33.13	41.38	43.63	47.38	55.85	62.15	70.6	83	85.5
6	1	Job number	90	48	86													
		Start Time	85.5	88.7	100.65													
		End Time	87.7	90.7	101.65													
7	1	Job number	82	9	12	10	68	76	1	33	80	34	91	57	78	29	97	93
		Start Time	2.5	4.5	5.5	9.57	10.57	12.57	13.82	18.82	20.02	24.6	27	33.25	35.6	36.8	39.3	41.3
		End Time	4.5	5.5	8	10.57	12.57	13.82	18.82	20.02	20.77	27	29	35.6	36.8	39.3	41.3	43.3
		Job number	79	18	36	37	84	2	28	23	24	64	47	5	89	92		
		Start Time	43.31	46.85	48.85	51.56	53.56	56.76	59.52	60.62	62.62	64.95	83	85.5	101.5	102.75		
		End Time	44.28	48.35	50.85	53.56	56.76	59.52	60.62	62.62	64.62	66.95	85.5	101.5	102.75	106.41		



**Lampiran 2: Hasil Run Program Solusi Akhir (lanjutan)**

Nomor Mesin	Mesin Ke	Item	Urutan pengerjaan															
			9	77	12	68	8	11	65	53	62	97	87	93	88	42	36	37
8	1	Job number																
		Start Time	5.5	6.75	8.75	12.57	13.85	16.46	19.55	29.16	39.21	42.41	42.91	44.41	45.66	47.31	50.85	53.56
		End Time	6.75	8.75	9.85	13.57	16.35	18.46	20.75	31.61	42.41	42.91	44.41	45.66	47.16	50.31	53.35	55.56
		Job number	66	2	58	98												
		Start Time	55.56	59.52	63.7	68												
		End Time	58.76	60.64	68	69.5												



### Lampiran 3: Bahasa Pemrograman VBA

#### Modules Job Shop Scheduling

```

Option Explicit 'every variables must be declared first
'Option Private Module
Option Base 1 'start arrays with one
Dim myData As JSSData
Dim mySolution As JSSSolution
Dim DESolution1, DESolution2, DESolution3, DESolution4 As JSSSolution
Dim SolutionAwal, SolutionBarus As Collection
Dim UrutanSolusiKontinyu() As Double
Dim UrutanSolusiInteger() As Integer
Dim TotalPanjangSolusi As Integer
Dim i, j, k, l As Integer
Dim indexTerbaik As Integer
Dim TotalBiayaTerbaik As Double
Dim jumlahIterasiTanpaPerbaikan As Integer
Dim Vector1, Vector2, Vector3 As Integer
Dim randTemp As Double
Dim TimeBefore, TimeAfter As Date
Public TotalBiayaTabu() As Double
Public MakeSpanTabu() As Double
Public NomorUpdateTabu As Integer
Dim UpdateTabu As Boolean

Sub RunAlgorithm()
TimeBefore = Now
'----- Randomize random number generator -----
Randomize

'----- Ambil data data data -----
Dim myData As New JSSData
myData.GetDataFromWorksheet

'----- Buat initial random Solution -----
Dim SolusiAwal As New JSSSolution
SolusiAwal.SetJSSData myData
SolusiAwal.GenerateRandomSolution
For i = 1 To myData.JumlahSolusiTetangga
    Dim mySolution As New JSSSolution
    mySolution.SetJSSData myData
    mySolution.GenerateRandomSolution
    If IsBetter(mySolution, SolusiAwal) Then
        Set SolusiAwal = mySolution
    End If
    Set mySolution = Nothing
Next i

```

```

'-----Tampilkan solusi awal terbaik -----
SolusiAwal.PrintDataToWorksheet "SolusiAwal"

'-----Inisialisasi iterasi -----
Dim SolusiTerbaik As New JSSSolution
Set SolusiTerbaik = SolusiAwal
Dim SolusiIterasi As New JSSSolution
Set SolusiIterasi = SolusiAwal
ReDim TotalBiayaTabu(myData.PanjangTabuList)
ReDim SolutionFeasibilityTabu(myData.PanjangTabuList)
ReDim MakeSpanTabu(myData.PanjangTabuList)
NomorUpdateTabu = 0
TotalPanjangSolusi = SolusiTerbaik.TotalPanjangSolusi

For i = 1 To myData.JumlahIterasiMaksimum

    Dim SolusiIterasi2 As New JSSSolution
    SolusiIterasi2.SetJSSData myData
    Set SolusiIterasi2.SetUrutanInteger = SolusiIterasi.getUrutanInteger
    SolusiIterasi2.GenerateSolusiTetangga

    'cari solusi tetangga
    For j = 1 To myData.JumlahSolusiTetangga
        Dim SolusiIterasi3 As New JSSSolution
        SolusiIterasi3.SetJSSData myData
        Set SolusiIterasi3.SetUrutanInteger = SolusiIterasi.getUrutanInteger
        SolusiIterasi3.GenerateSolusiTetangga
        If IsBetter(SolusiIterasi3, SolusiIterasi2) Then
            Set SolusiIterasi2 = SolusiIterasi3
        End If
        Set SolusiIterasi3 = Nothing
    Next j

    'kalau lebih baik dari solusi global, langsung diterima
    If IsBetter(SolusiIterasi2, SolusiTerbaik) Then
        Set SolusiTerbaik = SolusiIterasi2
        Set SolusiIterasi = SolusiIterasi2
        UpdateTabu = UpdateTabuList(SolusiIterasi2, myData)
    Else 'cek apakah tabu atau tidak
        If IsTabu(SolusiIterasi2, myData) = False Then
            Set SolusiIterasi = SolusiIterasi2
            UpdateTabu = UpdateTabuList(SolusiIterasi2, myData)
        End If
    End If

    Set SolusiIterasi2 = Nothing

Next i

```

'----- Tampilkan solusi terbaik -----'  
 SolusiTerbaik.PrintDataToWorksheet "SolusiAkhir"

```
TimeAfter = Now
Sheets("SolusiAkhir").Cells(4, 1).Value = "StartTime"
Sheets("SolusiAkhir").Cells(4, 2).Value = TimeBefore
Sheets("SolusiAkhir").Cells(5, 1).Value = "EndTime"
Sheets("SolusiAkhir").Cells(5, 2).Value = TimeAfter
```

End Sub

```
Public Function IsBetter(Solusi1 As JSSSolution, Solusi2 As JSSSolution) As Boolean
  If Solusi1.getTotalBiaya < Solusi2.getTotalBiaya Then
    IsBetter = True
  End If
End Function
```

```
Public Function UpdateTabuList(Solusi As JSSSolution, myData As JSSData) As Boolean
  NomorUpdateTabu = NomorUpdateTabu + 1
  If NomorUpdateTabu > myData.PanjangTabuList Then
    NomorUpdateTabu = 1
  End If
  TotalBiayaTabu(NomorUpdateTabu) = Solusi.getTotalBiaya
  MakeSpanTabu(NomorUpdateTabu) = Solusi.getMakespan
  UpdateTabuList = True
End Function
```

```
Public Function IsTabu(Solusi As JSSSolution, myData As JSSData) As Boolean
  IsTabu = False
  For l = 1 To myData.PanjangTabuList
    If TotalBiayaTabu(l) = Solusi.getTotalBiaya Then
      If MakeSpanTabu(l) = Solusi.getMakespan Then
        IsTabu = True
        Exit Function
      End If
    End If
  Next l
End Function
```

### Class Modules JSSData

```

Option Explicit 'every variables must be declared first
Option Base 1 'start arrays with one
Public JumlahJob, JumlahMesin, JumlahSolusiTetangga, PanjangTabuList,
JumlahIterasiMaksimum, jumlahIterasiTanpaPerbaikan As Integer
Public FaktorMutasi, TingkatCrossover As Double
Dim JumlahTiapMesin() As Integer
Dim JumlahProsesTiapJob() As Integer
Dim UrutanProses() As Integer
Dim WaktuProses() As Double
Dim JumlahWaktuProsesTiapJob() As Double
Dim BiayaProduksi() As Double
Dim BiayaPenalty() As Double
Dim DueDate() As Double
Public MaksimumBiayaPenalty, MaksimumJumlahWaktuProses, MaksimumDueDate As
Double
Public JumlahMesinMaksimum As Integer
Dim i, j As Integer
Dim BarisKe, KolomKe As Integer

Public Sub GetDataFromWorksheet()

'Ambil data umum

JumlahJob = Sheets("DataUmum").Cells(7, 4).Value
JumlahMesin = Sheets("DataUmum").Cells(8, 4).Value
JumlahSolusiTetangga = Sheets("DataUmum").Cells(10, 4).Value
PanjangTabuList = Sheets("DataUmum").Cells(11, 4).Value
JumlahIterasiMaksimum = Sheets("DataUmum").Cells(13, 4).Value

'Inisialisasi ulang

ReDim JumlahTiapMesin(JumlahMesin)
ReDim JumlahProsesTiapJob(JumlahJob)
ReDim UrutanProses(JumlahJob, 2 * JumlahMesin)
ReDim WaktuProses(JumlahJob, 2 * JumlahMesin)
ReDim BiayaProduksi(JumlahJob)
ReDim BiayaPenalty(JumlahJob)

ReDim DueDate(JumlahJob)
ReDim JumlahWaktuProsesTiapJob(JumlahJob)

```

'Jumlah mesin

JumlahMesinMaksimum = 0

For i = 1 To JumlahMesin

    JumlahTiapMesin(i) = Sheets("DataMesin").Cells(i + 6, 3).Value

    If JumlahMesinMaksimum < JumlahTiapMesin(i) Then

        JumlahMesinMaksimum = JumlahTiapMesin(i)

    End If

Next i

'Ambil data Urutan Proses dan update jumlahProsesTiapJob

MaksimumJumlahWaktuProses = 0

For i = 1 To JumlahJob

    JumlahProsesTiapJob(i) = 0

    JumlahWaktuProsesTiapJob(i) = 0

    Do While Sheets("DataUrutanProses").Cells(i + 4, JumlahProsesTiapJob(i) + 3).Value <> ""

        JumlahProsesTiapJob(i) = JumlahProsesTiapJob(i) + 1

        UrutanProses(i, JumlahProsesTiapJob(i)) = Sheets("DataUrutanProses").Cells(i + 4, JumlahProsesTiapJob(i) + 2).Value

        WaktuProses(i, JumlahProsesTiapJob(i)) = Sheets("DataWaktuProses").Cells(i + 4, JumlahProsesTiapJob(i) + 2).Value

        JumlahWaktuProsesTiapJob(i) = JumlahWaktuProsesTiapJob(i) + WaktuProses(i, JumlahProsesTiapJob(i))

    Loop

    If JumlahWaktuProsesTiapJob(i) > MaksimumJumlahWaktuProses Then

        MaksimumJumlahWaktuProses = JumlahWaktuProsesTiapJob(i)

    End If

Next i

'Data Biaya

MaksimumBiayaPenalty = 0

MaksimumDueDate = 0

For i = 1 To JumlahJob

    BiayaProduksi(i) = Sheets("DataBiaya").Cells(i + 4, 2).Value

    BiayaPenalty(i) = Sheets("DataBiaya").Cells(i + 4, 3).Value

    DueDate(i) = Sheets("DataBiaya").Cells(i + 4, 4).Value

```

If BiayaPenalty(i) > MaksimumBiayaPenalty Then
    MaksimumBiayaPenalty = BiayaPenalty(i)
End If
If DueDate(i) > MaksimumDueDate Then
    MaksimumDueDate = DueDate(i)
End If
Next i

End Sub

Public Sub PrintDataToWorksheet()

'Delete Summary sheet if it exists
On Error Resume Next
Application.DisplayAlerts = False
Sheets("JSSDataDebug").Delete
On Error GoTo 0
Application.DisplayAlerts = True

'Add new worksheet
Worksheets.Add
ActiveSheet.Name = "JSSDataDebug"

'data umum

Sheets("JSSDataDebug").Cells(7, 3).Value = "JumlahJob"
Sheets("JSSDataDebug").Cells(7, 4).Value = JumlahJob
Sheets("JSSDataDebug").Cells(8, 3).Value = "JumlahMesin"
Sheets("JSSDataDebug").Cells(8, 4).Value = JumlahMesin
Sheets("JSSDataDebug").Cells(9, 3).Value = "JumlahSolusiTetangga"
Sheets("JSSDataDebug").Cells(9, 4).Value = JumlahSolusiTetangga

Sheets("JSSDataDebug").Cells(10, 3).Value = "PanjangTabuList"
Sheets("JSSDataDebug").Cells(10, 4).Value = PanjangTabuList
Sheets("JSSDataDebug").Cells(11, 3).Value = "JumlahIterasiMaksimum"
Sheets("JSSDataDebug").Cells(11, 4).Value = JumlahIterasiMaksimum
Sheets("JSSDataDebug").Cells(12, 3).Value = "JumlahIterasiTanpaPerbaikan"
Sheets("JSSDataDebug").Cells(12, 4).Value = jumlahIterasiTanpaPerbaikan

```

'Jumlah mesin

```
Sheets("JSSDataDebug").Cells(16, 3).Value = "Nomor Mesin"
```

```
Sheets("JSSDataDebug").Cells(16, 4).Value = "Jumlah Mesin"
```

```
BarisKe = 17
```

```
For i = 1 To JumlahMesin
```

```
    Sheets("JSSDataDebug").Cells(BarisKe, 3).Value = i
```

```
    Sheets("JSSDataDebug").Cells(BarisKe, 4).Value = JumlahTiapMesin(i)
```

```
    BarisKe = BarisKe + 1
```

```
Next i
```

'data Urutan Proses dan jumlahProsesTiapJob

```
BarisKe = BarisKe + 2
```

```
Sheets("JSSDataDebug").Cells(BarisKe, 3).Value = "Nomor Job"
```

```
Sheets("JSSDataDebug").Cells(BarisKe, 4).Value = "Total Count"
```

```
Sheets("JSSDataDebug").Cells(BarisKe, 5).Value = "Urutan"
```

```
BarisKe = BarisKe + 1
```

```
For i = 1 To JumlahJob
```

```
    Sheets("JSSDataDebug").Cells(BarisKe, 3).Value = i
```

```
    Sheets("JSSDataDebug").Cells(BarisKe, 4).Value = JumlahProsesTiapJob(i)
```

```
    KolomKe = 5
```

```
    For j = 1 To JumlahProsesTiapJob(i)
```

```
        Sheets("JSSDataDebug").Cells(BarisKe, KolomKe).Value = UrutanProses(i, j)
```

```
        KolomKe = KolomKe + 1
```

```
    Next j
```

```
    BarisKe = BarisKe + 1
```

```
Next i
```

```
BarisKe = BarisKe + 2
```

```
Sheets("JSSDataDebug").Cells(BarisKe, 3).Value = "Nomor Job"
```

```
Sheets("JSSDataDebug").Cells(BarisKe, 4).Value = "Total Time"
```

```
Sheets("JSSDataDebug").Cells(BarisKe, 5).Value = "Waktu Proses"
```

```
BarisKe = BarisKe + 1
```

```
For i = 1 To JumlahJob
```

```
    Sheets("JSSDataDebug").Cells(BarisKe, 3).Value = i
```

```
    KolomKe = 5
```

```
    For j = 1 To JumlahProsesTiapJob(i)
```

```
        Sheets("JSSDataDebug").Cells(BarisKe, KolomKe).Value = WaktuProses(i, j)
```

```
        KolomKe = KolomKe + 1
```

```
    Next j
```



```
BarisKe = BarisKe + 1
```

```
Next i
```

```
BarisKe = BarisKe + 2
```

```
Sheets("JSSDataDebug").Cells(BarisKe, 3).Value = "Nomor Job"
```

```
Sheets("JSSDataDebug").Cells(BarisKe, 4).Value = "BiayaProduksi"
```

```
Sheets("JSSDataDebug").Cells(BarisKe, 5).Value = "BiayaPenalty"
```

```
Sheets("JSSDataDebug").Cells(BarisKe, 6).Value = "DueDate"
```

```
BarisKe = BarisKe + 1
```

```
'Data Biaya
```

```
For i = 1 To JumlahJob
```

```
    Sheets("JSSDataDebug").Cells(BarisKe, 3).Value = i
```

```
    Sheets("JSSDataDebug").Cells(BarisKe, 4).Value = BiayaProduksi(i)
```

```
    Sheets("JSSDataDebug").Cells(BarisKe, 5).Value = BiayaPenalty(i)
```

```
    Sheets("JSSDataDebug").Cells(BarisKe, 6).Value = DueDate(i)
```

```
    BarisKe = BarisKe + 1
```

```
Next i
```

```
End Sub
```

```
Public Property Get GetJumlahTiapMesin() As Variant
```

```
    GetJumlahTiapMesin = JumlahTiapMesin
```

```
End Property
```

```
Public Property Get GetJumlahProsesTiapJob() As Variant
```

```
    GetJumlahProsesTiapJob = JumlahProsesTiapJob
```

```
End Property
```

```
Public Property Get GetUrutanProses() As Variant
```

```
    GetUrutanProses = UrutanProses
```

```
End Property
```

```
Public Property Get GetWaktuProses() As Variant
```

```
    GetWaktuProses = WaktuProses
```

```
End Property
```

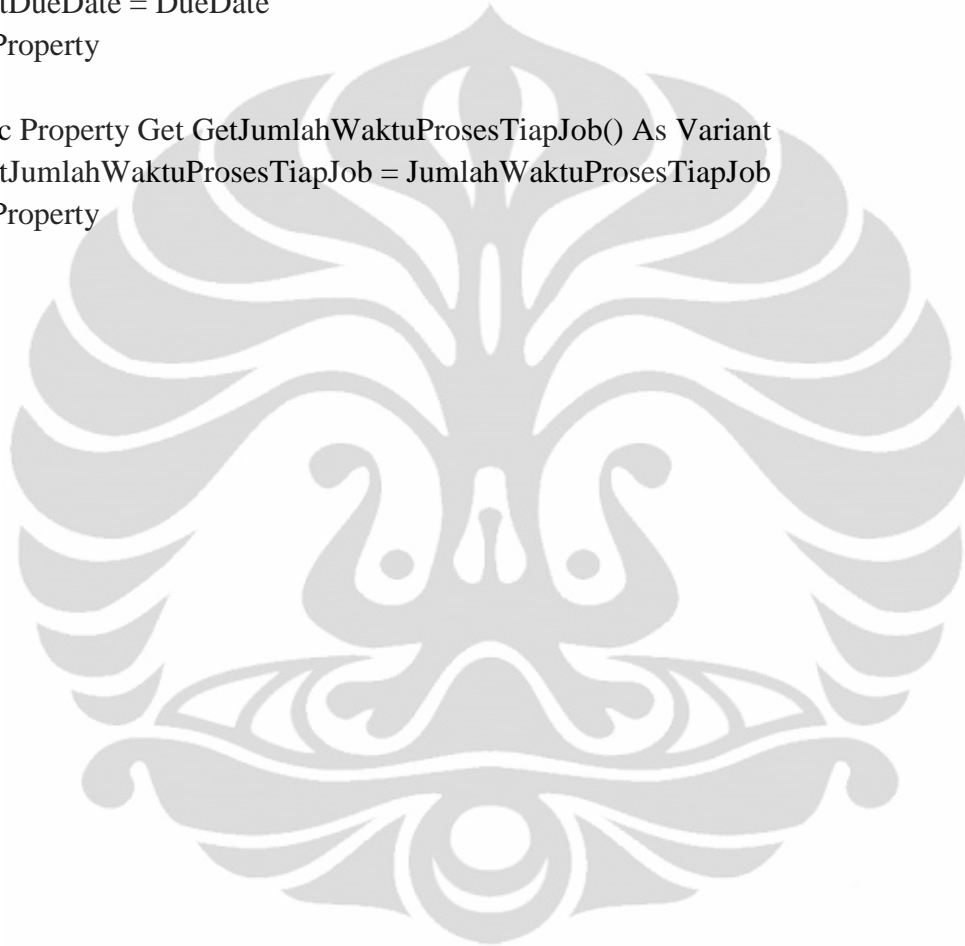
```
Public Property Get GetBiayaProduksi() As Variant
```

```
    GetBiayaProduksi = BiayaProduksi  
End Property
```

```
Public Property Get GetBiayaPenalty() As Variant  
    GetBiayaPenalty = BiayaPenalty  
End Property
```

```
Public Property Get GetDueDate() As Variant  
    GetDueDate = DueDate  
End Property
```

```
Public Property Get GetJumlahWaktuProsesTiapJob() As Variant  
    GetJumlahWaktuProsesTiapJob = JumlahWaktuProsesTiapJob  
End Property
```



### **Class Modules JSSSolution**

Option Explicit 'every variables must be declared first

Option Base 1 'start arrays with one

```

Dim myJSSData As JSSData
Dim UrutanSolusiKontinyu() As Double
Dim UrutanSolusiKontinyuTemp() As Double
Dim UrutanSolusiInteger() As Integer
Dim StartTime(), EndTime() As Double
Dim EndTimeForMachine() As Double
Dim CurrentProcessForJob() As Integer
Dim MachineNumberForJob() As Integer
Dim MachineNumberForUrutanSolusi() As Integer
Dim UrutanProsesForUrutanSolusi() As Integer
Dim Makespan As Double
Dim TotalBiaya As Double
Dim NomorJob, UrutanProsesKe, NomorMesin As Integer
Dim WaktuProses As Double
Dim MesinTerpilih As Integer
Dim WaktuMesinTerkecil As Double
Public TotalPanjangSolusi
Dim Weight() As Double
Dim ResidualTime As Double
Dim HasBeenAssigned() As Boolean
Dim MinimumWeight As Double
Dim IndeksWeightTerpilih, IndeksUrutanSolusiInteger As Integer
Dim i, j, k, l As Integer
Dim BarisKe, KolomKe As Integer
Dim Sorted As Boolean
Dim DoubleTemp As Double
Dim IntegerTemp As Integer
Dim DoubleTempArray() As Double
Dim IntegerTempArray() As Integer
Dim First, Second, urutanKe As Integer
Dim randTemp As Double

```

```

Public Sub Initialize()
    Set myJSSData = New JSSData
    myJSSData.GetDataFromWorksheet
    Call DataInitialize
End Sub

```

```

Public Sub SetJSSData(iJSSData As JSSData)

```

```

Set myJSSData = iJSSData
Call DataInitialize
End Sub

Private Sub DataInitialize()
'calculate panjangSolusi
TotalPanjangSolusi = 0
For i = 1 To myJSSData.JumlahJob
    TotalPanjangSolusi = TotalPanjangSolusi + myJSSData.GetJumlahProsesTiapJob(i)
Next i

ReDim UrutanSolusiKontinyu(TotalPanjangSolusi)
ReDim UrutanSolusiKontinyuTemp(TotalPanjangSolusi)
ReDim UrutanSolusiInteger(TotalPanjangSolusi)
ReDim StartTime(myJSSData.JumlahJob, 2 * myJSSData.JumlahMesin)
ReDim EndTime(myJSSData.JumlahJob, 2 * myJSSData.JumlahMesin)
ReDim EndTimeForMachine(myJSSData.JumlahMesin)
ReDim CurrentProcessForJob(myJSSData.JumlahJob)
ReDim MachineNumberForJob(myJSSData.JumlahJob, 2 * myJSSData.JumlahMesin)
Makespan = 0

End Sub

Public Sub GenerateRandomSolution()
For i = 1 To TotalPanjangSolusi
    UrutanSolusiKontinyu(i) = Rnd
Next i
Call ConvertToUrutanInteger
Call CalculateObjectiveFunction
End Sub

Public Sub GenerateHeuristicSolution()
ReDim Weight(myJSSData.JumlahJob)
For i = 1 To myJSSData.JumlahJob
    ResidualTime = myJSSData.GetDueDate(i) -
myJSSData.GetJumlahWaktuProsesTiapJob(i)
    If ResidualTime < 0 Then
        Weight(i) = ResidualTime * myJSSData.GetBiayaPenalty(i)
    Else
        Weight(i) = ResidualTime / myJSSData.GetBiayaPenalty(i)
    End If
Next i
ReDim HasBeenAssigned(myJSSData.JumlahJob)
IndeksWeightTerpilih = 1

```

```

IndeksUrutanSolusiInteger = 1
For i = 1 To myJSSData.JumlahJob
  MinimumWeight = myJSSData.MaksimumJumlahWaktuProses *
myJSSData.MaksimumBiayaPenalty
  For j = 1 To myJSSData.JumlahJob
    If Weight(j) < MinimumWeight Then
      If HasBeenAssigned(j) = False Then
        MinimumWeight = Weight(j)
        IndeksWeightTerpilih = j
      End If
    End If
  Next j
  HasBeenAssigned(IndeksWeightTerpilih) = True
  For j = 1 To myJSSData.GetJumlahProsesTiapJob(IndeksWeightTerpilih)
    UrutanSolusiInteger(IndeksUrutanSolusiInteger) = IndeksWeightTerpilih
    IndeksUrutanSolusiInteger = IndeksUrutanSolusiInteger + 1
  Next j
Next i
For i = 1 To TotalPanjangSolusi
  UrutanSolusiKontinyu(i) = i / TotalPanjangSolusi
Next i
Call CalculateObjectiveFunction
End Sub

Public Sub GenerateDueDateSortedSolution()
  ReDim HasBeenAssigned(myJSSData.JumlahJob)
  IndeksWeightTerpilih = 1
  IndeksUrutanSolusiInteger = 1
  For i = 1 To myJSSData.JumlahJob
    MinimumWeight = myJSSData.MaksimumDueDate
    For j = 1 To myJSSData.JumlahJob
      If myJSSData.GetDueDate(j) <= MinimumWeight Then
        If HasBeenAssigned(j) = False Then
          MinimumWeight = myJSSData.GetDueDate(j)
          IndeksWeightTerpilih = j
        End If
      End If
    Next j
    HasBeenAssigned(IndeksWeightTerpilih) = True
    For j = 1 To myJSSData.GetJumlahProsesTiapJob(IndeksWeightTerpilih)
      UrutanSolusiInteger(IndeksUrutanSolusiInteger) = IndeksWeightTerpilih
      IndeksUrutanSolusiInteger = IndeksUrutanSolusiInteger + 1
    Next j
  Next i

```

```

For i = 1 To TotalPanjangSolusi
    UrutanSolusiKontinyu(i) = i / TotalPanjangSolusi
Next i
Call CalculateObjectiveFunction
End Sub

```

```

Public Sub GenerateResidualDueDateSortedSolution()
    ReDim HasBeenAssigned(myJSSData.JumlahJob)
    IndeksWeightTerpilih = 1
    IndeksUrutanSolusiInteger = 1
    For i = 1 To myJSSData.JumlahJob
        MinimumWeight = myJSSData.MaksimumDueDate
        For j = 1 To myJSSData.JumlahJob
            If (myJSSData.GetDueDate(j) - myJSSData.GetJumlahWaktuProsesTiapJob(j)) <=
MinimumWeight Then
                If HasBeenAssigned(j) = False Then
                    MinimumWeight = (myJSSData.GetDueDate(j) -
myJSSData.GetJumlahWaktuProsesTiapJob(j))
                    IndeksWeightTerpilih = j
                End If
            End If
        Next j
        HasBeenAssigned(IndeksWeightTerpilih) = True
        For j = 1 To myJSSData.GetJumlahProsesTiapJob(IndeksWeightTerpilih)
            UrutanSolusiInteger(IndeksUrutanSolusiInteger) = IndeksWeightTerpilih
            IndeksUrutanSolusiInteger = IndeksUrutanSolusiInteger + 1
        Next j
    Next i
    For i = 1 To TotalPanjangSolusi
        UrutanSolusiKontinyu(i) = i / TotalPanjangSolusi
    Next i
    Call CalculateObjectiveFunction
End Sub

```

```

Public Sub GeneratePenaltySortedSolution()
    ReDim HasBeenAssigned(myJSSData.JumlahJob)
    IndeksWeightTerpilih = 1
    IndeksUrutanSolusiInteger = 1
    For i = 1 To myJSSData.JumlahJob
        MinimumWeight = 0
        For j = 1 To myJSSData.JumlahJob
            If myJSSData.GetBiayaPenalty(j) >= MinimumWeight Then
                If HasBeenAssigned(j) = False Then
                    MinimumWeight = myJSSData.GetBiayaPenalty(j)
                End If
            End If
        Next j
    Next i

```

```

        IndeksWeightTerpilih = j
    End If
End If
Next j
HasBeenAssigned(IndeksWeightTerpilih) = True
For j = 1 To myJSSData.GetJumlahProsesTiapJob(IndeksWeightTerpilih)
    UrutanSolusiInteger(IndeksUrutanSolusiInteger) = IndeksWeightTerpilih
    IndeksUrutanSolusiInteger = IndeksUrutanSolusiInteger + 1
Next j
Next i
For i = 1 To TotalPanjangSolusi
    UrutanSolusiKontinyu(i) = i / TotalPanjangSolusi
Next i
Call CalculateObjectiveFunction
End Sub

Public Sub GenerateJobSortedSolution()
    IndeksUrutanSolusiInteger = 1
    For i = 1 To myJSSData.JumlahJob
        For j = 1 To myJSSData.GetJumlahProsesTiapJob(i)
            UrutanSolusiInteger(IndeksUrutanSolusiInteger) = i
            IndeksUrutanSolusiInteger = IndeksUrutanSolusiInteger + 1
        Next j
    Next i
    For i = 1 To TotalPanjangSolusi
        UrutanSolusiKontinyu(i) = i / TotalPanjangSolusi
    Next i
    Call CalculateObjectiveFunction
End Sub

Private Sub ConvertToUrutanInteger()
    'Temporary for sorting
    UrutanSolusiKontinyuTemp = UrutanSolusiKontinyu
    'Initial UrutanSolusiInteger
    k = 0
    For i = 1 To myJSSData.JumlahJob
        For j = 1 To myJSSData.GetJumlahProsesTiapJob(i)
            k = k + 1
            UrutanSolusiInteger(k) = i
        Next j
    Next i
    'perform bubble sort
    Sorted = False
    Do While Sorted = False

```

```

Sorted = True
For i = 1 To TotalPanjangSolusi - 1
  If UrutanSolusiKontinyuTemp(i) > UrutanSolusiKontinyuTemp(i + 1) Then
    Sorted = False
    DoubleTemp = UrutanSolusiKontinyuTemp(i)
    UrutanSolusiKontinyuTemp(i) = UrutanSolusiKontinyuTemp(i + 1)
    UrutanSolusiKontinyuTemp(i + 1) = DoubleTemp
    IntegerTemp = UrutanSolusiInteger(i)
    UrutanSolusiInteger(i) = UrutanSolusiInteger(i + 1)
    UrutanSolusiInteger(i + 1) = IntegerTemp
  End If
Next i
Loop
End Sub

Private Sub CalculateObjectiveFunction()
  ReDim StartTime(myJSSData.JumlahJob, 2 * myJSSData.JumlahMesin)
  ReDim EndTime(myJSSData.JumlahJob, 2 * myJSSData.JumlahMesin)
  ReDim EndTimeForMachine(myJSSData.JumlahMesin,
myJSSData.JumlahMesinMaksimum)
  ReDim CurrentProcessForJob(myJSSData.JumlahJob)
  ReDim MachineNumberForJob(myJSSData.JumlahJob, 2 * myJSSData.JumlahMesin)
  ReDim MachineNumberForUrutanSolusi(TotalPanjangSolusi)
  ReDim UrutanProsesForUrutanSolusi(TotalPanjangSolusi)
  Makespan = 0

  For i = 1 To myJSSData.JumlahMesin
    For j = 1 To myJSSData.JumlahMesinMaksimum
      EndTimeForMachine(i, j) = 0
    Next j
  Next i
  For i = 1 To myJSSData.JumlahJob
    CurrentProcessForJob(i) = 1
  Next i
  For i = 1 To TotalPanjangSolusi
    NomorJob = UrutanSolusiInteger(i)
    UrutanProsesKe = CurrentProcessForJob(NomorJob)
    NomorMesin = myJSSData.GetUrutanProses(NomorJob, UrutanProsesKe)
    WaktuProses = myJSSData.GetWaktuProses(NomorJob, UrutanProsesKe)

    MesinTerpilih = 1
    WaktuMesinTerkecil = EndTimeForMachine(NomorMesin, 1)

```



```

For j = 2 To myJSSData.GetJumlahTiapMesin(NomorMesin)
  If EndTimeForMachine(NomorMesin, j) < WaktuMesinTerkecil Then
    WaktuMesinTerkecil = EndTimeForMachine(NomorMesin, j)
    MesinTerpilih = j
  End If
Next j
MachineNumberForUrutanSolusi(i) = MesinTerpilih
UrutanProsesForUrutanSolusi(i) = UrutanProsesKe

If UrutanProsesKe = 1 Then
  StartTime(NomorJob, UrutanProsesKe) = WaktuMesinTerkecil
  EndTime(NomorJob, UrutanProsesKe) = StartTime(NomorJob, UrutanProsesKe) +
WaktuProses
  EndTimeForMachine(NomorMesin, MesinTerpilih) = EndTime(NomorJob,
UrutanProsesKe)
  MachineNumberForJob(NomorJob, UrutanProsesKe) = MesinTerpilih
  CurrentProcessForJob(NomorJob) = CurrentProcessForJob(NomorJob) + 1
Else
  If WaktuMesinTerkecil >= EndTime(NomorJob, UrutanProsesKe - 1) Then
    StartTime(NomorJob, UrutanProsesKe) = WaktuMesinTerkecil
  Else
    StartTime(NomorJob, UrutanProsesKe) = EndTime(NomorJob, UrutanProsesKe -
1)
  End If
  EndTime(NomorJob, UrutanProsesKe) = StartTime(NomorJob, UrutanProsesKe) +
WaktuProses
  EndTimeForMachine(NomorMesin, MesinTerpilih) = EndTime(NomorJob,
UrutanProsesKe)
  MachineNumberForJob(NomorJob, UrutanProsesKe) = MesinTerpilih
  CurrentProcessForJob(NomorJob) = CurrentProcessForJob(NomorJob) + 1
End If
Next i

For i = 1 To myJSSData.JumlahJob
  If EndTime(i, myJSSData.GetJumlahProsesTiapJob(i)) > Makespan Then
    Makespan = EndTime(i, myJSSData.GetJumlahProsesTiapJob(i))
  End If
Next i

TotalBiaya = 0
For i = 1 To myJSSData.JumlahJob
  TotalBiaya = TotalBiaya + myJSSData.GetBiayaProduksi(i)
  If EndTime(i, myJSSData.GetJumlahProsesTiapJob(i)) > myJSSData.GetDueDate(i)
Then

```

```

    TotalBiaya = TotalBiaya + (EndTime(i, myJSSData.GetJumlahProsesTiapJob(i)) -
myJSSData.GetDueDate(i)) * myJSSData.GetBiayaPenalty(i)
    End If
Next i

```

```
End Sub
```

```

Public Sub SetUrutanKontinyu(newUrutanKontinyu() As Double)
    If UBound(newUrutanKontinyu) = TotalPanjangSolusi Then
        UrutanSolusiKontinyu = newUrutanKontinyu
    End If
    Call ConvertToUrutanInteger
    Call CalculateObjectiveFunction
End Sub

```

```

Public Sub SetUrutanInteger2(newUrutanInteger() As Integer)
    If UBound(newUrutanInteger) = TotalPanjangSolusi Then
        UrutanSolusiInteger = newUrutanInteger
    End If
    Call CalculateObjectiveFunction
End Sub

```

```

Public Property Get getUrutanInteger() As Variant
    getUrutanInteger = UrutanSolusiInteger
End Property

```

```

Public Property Set SetUrutanInteger(iUrutanSolusiInteger As Variant)
    UrutanSolusiInteger = iUrutanSolusiInteger
End Property

```

```

Public Sub PerformLocalSearchInsert()
    First = Round(Rnd * (TotalPanjangSolusi - 1)) + 1
    Second = Round(Rnd * (TotalPanjangSolusi - 1)) + 1

```

```

    IntegerTemp = UrutanSolusiInteger(First)

```

```

    If First < Second Then

```

```

        For i = First To Second - 1

```

```

            UrutanSolusiInteger(i) = UrutanSolusiInteger(i + 1)

```

```

        Next i

```

```

        UrutanSolusiInteger(Second) = IntegerTemp

```

```

    ElseIf First > Second Then

```

```

        For i = First To Second + 1 Step -1

```

```

            UrutanSolusiInteger(i) = UrutanSolusiInteger(i - 1)

```

```

    Next i
    UrutanSolusiInteger(Second) = IntegerTemp
End If
Call CalculateObjectiveFunction
End Sub

Public Sub PerformLocalSearch2Opt()
    First = Round(Rnd * (TotalPanjangSolusi - 1)) + 1
    Second = Round(Rnd * (TotalPanjangSolusi - 1)) + 1

    Do While First >= Second
        First = Round(Rnd * (TotalPanjangSolusi - 1)) + 1
        Second = Round(Rnd * (TotalPanjangSolusi - 1)) + 1
    Loop

    ReDim IntegerTempArray(Second - First + 1)
    urutanKe = 0
    For i = Second To First Step -1
        urutanKe = urutanKe + 1
        IntegerTempArray(urutanKe) = UrutanSolusiInteger(i)
    Next i
    urutanKe = 0
    For i = First To Second
        urutanKe = urutanKe + 1
        UrutanSolusiInteger(i) = IntegerTempArray(urutanKe)
    Next i
    Call CalculateObjectiveFunction
End Sub

Public Sub PerformLocalSearchChange()
    First = Round(Rnd * (TotalPanjangSolusi - 1)) + 1
    Second = Round(Rnd * (TotalPanjangSolusi - 1)) + 1

    IntegerTemp = UrutanSolusiInteger(First)
    UrutanSolusiInteger(First) = UrutanSolusiInteger(Second)
    UrutanSolusiInteger(Second) = IntegerTemp

    Call CalculateObjectiveFunction
End Sub

Public Function GenerateSolusiTetangga()
    randTemp = Rnd
    If randTemp < 0.33 Then

```

```

    Call PerformLocalSearch2Opt
ElseIf randTemp < 0.66 Then
    Call PerformLocalSearchChange
Else
    Call PerformLocalSearchInsert
End If
End Function

Public Property Get getMakespan()
    getMakespan = Makespan
End Property

Public Property Get getTotalBiaya()
    getTotalBiaya = TotalBiaya
End Property

Public Sub PrintDataToWorksheet(ByVal NamaWorksheet As String)

'Delete JSSSolutionDebug sheet if it exists
On Error Resume Next
Application.DisplayAlerts = False
Sheets(NamaWorksheet).Delete
On Error GoTo 0
Application.DisplayAlerts = True

'Add new worksheet
Worksheets.Add
ActiveSheet.Name = NamaWorksheet

BarisKe = 1
Sheets(NamaWorksheet).Cells(BarisKe, 1).Value = "Total Biaya"
Sheets(NamaWorksheet).Cells(BarisKe, 2).Value = TotalBiaya

BarisKe = BarisKe + 1
Sheets(NamaWorksheet).Cells(BarisKe, 1).Value = "Makespan"
Sheets(NamaWorksheet).Cells(BarisKe, 2).Value = Makespan

'Solusi Kontinyu
BarisKe = BarisKe + 4
Sheets(NamaWorksheet).Cells(BarisKe, 1).Value = "Solusi Kontinyu"
KolomKe = 1
For i = 1 To TotalPanjangSolusi
    KolomKe = KolomKe + 1
    Sheets(NamaWorksheet).Cells(BarisKe, KolomKe).Value = UrutanSolusiKontinyu(i)

```

```

Next i
'Solusi Kontinyu Sorted
BarisKe = BarisKe + 1
Sheets>NamaWorksheet).Cells(BarisKe, 1).Value = "Solusi Kontinyu Sorted"
KolomKe = 1
For i = 1 To TotalPanjangSolusi
    KolomKe = KolomKe + 1
    Sheets>NamaWorksheet).Cells(BarisKe, KolomKe).Value =
UrutanSolusiKontinyuTemp(i)
Next i
'Solusi Integer
BarisKe = BarisKe + 1
Sheets>NamaWorksheet).Cells(BarisKe, 1).Value = "Solusi Job"
KolomKe = 1
For i = 1 To TotalPanjangSolusi
    KolomKe = KolomKe + 1
    Sheets>NamaWorksheet).Cells(BarisKe, KolomKe).Value = UrutanSolusiInteger(i)
Next i

BarisKe = BarisKe + 2
Sheets>NamaWorksheet).Cells(BarisKe, 1).Value = "Nomor Job"
Sheets>NamaWorksheet).Cells(BarisKe, 2).Value = "Mesin Ke"
BarisKe = BarisKe + 1
For i = 1 To myJSSData.JumlahJob
    Sheets>NamaWorksheet).Cells(BarisKe, 1).Value = i
    KolomKe = 2
    For j = 1 To myJSSData.GetJumlahProsesTiapJob(i)
        Sheets>NamaWorksheet).Cells(BarisKe, KolomKe).Value = MachineNumberForJob(i,
j)
        KolomKe = KolomKe + 1
    Next j
    BarisKe = BarisKe + 1
Next i

BarisKe = BarisKe + 2
Sheets>NamaWorksheet).Cells(BarisKe, 1).Value = "Nomor Job"
Sheets>NamaWorksheet).Cells(BarisKe, 2).Value = "Item"
Sheets>NamaWorksheet).Cells(BarisKe, 3).Value = "Time"
BarisKe = BarisKe + 1
For i = 1 To myJSSData.JumlahJob
    Sheets>NamaWorksheet).Cells(BarisKe, 1).Value = i
    Sheets>NamaWorksheet).Cells(BarisKe, 2).Value = "Start Time"
    Sheets>NamaWorksheet).Cells(BarisKe + 1, 2).Value = "End Time"
    KolomKe = 3

```

```

For j = 1 To myJSSData.GetJumlahProsesTiapJob(i)
  Sheets>NamaWorksheet).Cells(BarisKe, KolomKe).Value = StartTime(i, j)
  Sheets>NamaWorksheet).Cells(BarisKe + 1, KolomKe).Value = EndTime(i, j)
  KolomKe = KolomKe + 1
Next j
BarisKe = BarisKe + 2
Next i

BarisKe = BarisKe + 2
Sheets>NamaWorksheet).Cells(BarisKe, 1).Value = "Nomor Mesin"
Sheets>NamaWorksheet).Cells(BarisKe, 2).Value = "Mesin Ke"
Sheets>NamaWorksheet).Cells(BarisKe, 3).Value = "Item"
BarisKe = BarisKe + 1
For i = 1 To myJSSData.JumlahMesin
  For j = 1 To myJSSData.GetJumlahTiapMesin(i)
    Sheets>NamaWorksheet).Cells(BarisKe, 1).Value = i
    Sheets>NamaWorksheet).Cells(BarisKe, 2).Value = j
    Sheets>NamaWorksheet).Cells(BarisKe, 3).Value = "Job number"
    Sheets>NamaWorksheet).Cells(BarisKe + 1, 3).Value = "Start Time"
    Sheets>NamaWorksheet).Cells(BarisKe + 2, 3).Value = "End Time"
    KolomKe = 4
    UrutanProsesKe = 1
    For k = 1 To TotalPanjangSolusi
      If myJSSData.GetUrutanProses(UrutanSolusiInteger(k),
UrutanProsesForUrutanSolusi(k)) = i Then
        If MachineNumberForUrutanSolusi(k) = j Then
          Sheets>NamaWorksheet).Cells(BarisKe, KolomKe).Value =
UrutanSolusiInteger(k)
          Sheets>NamaWorksheet).Cells(BarisKe + 1, KolomKe).Value =
StartTime(UrutanSolusiInteger(k), UrutanProsesForUrutanSolusi(k))
          Sheets>NamaWorksheet).Cells(BarisKe + 2, KolomKe).Value =
EndTime(UrutanSolusiInteger(k), UrutanProsesForUrutanSolusi(k))
          UrutanProsesKe = UrutanProsesKe + 1
          KolomKe = KolomKe + 1
        End If
      End If
    Next k
    BarisKe = BarisKe + 3
  Next j
Next i

BarisKe = BarisKe + 2
Sheets>NamaWorksheet).Cells(BarisKe, 1).Value = "Nomor Job"
Sheets>NamaWorksheet).Cells(BarisKe, 2).Value = "Biaya Produksi"

```

```

Sheets>NamaWorksheet).Cells(BarisKe, 3).Value = "DueDate"
Sheets>NamaWorksheet).Cells(BarisKe, 4).Value = "EndTime"
Sheets>NamaWorksheet).Cells(BarisKe, 5).Value = "Jumlah Waktu Keterlambatan"
Sheets>NamaWorksheet).Cells(BarisKe, 6).Value = "Biaya Penalty persatuan waktu"
Sheets>NamaWorksheet).Cells(BarisKe, 7).Value = "Biaya Penalty"
Sheets>NamaWorksheet).Cells(BarisKe, 8).Value = "Total Biaya"
BarisKe = BarisKe + 1
For i = 1 To myJSSData.JumlahJob
    Sheets>NamaWorksheet).Cells(BarisKe, 1).Value = i
    Sheets>NamaWorksheet).Cells(BarisKe, 2).Value = myJSSData.GetBiayaProduksi(i)
    Sheets>NamaWorksheet).Cells(BarisKe, 3).Value = myJSSData.GetDueDate(i)
    Sheets>NamaWorksheet).Cells(BarisKe, 4).Value = EndTime(i,
myJSSData.GetJumlahProsesTiapJob(i))
    If myJSSData.GetDueDate(i) < EndTime(i, myJSSData.GetJumlahProsesTiapJob(i)) Then
        Sheets>NamaWorksheet).Cells(BarisKe, 5).Value = (EndTime(i,
myJSSData.GetJumlahProsesTiapJob(i)) - myJSSData.GetDueDate(i))
    Else
        Sheets>NamaWorksheet).Cells(BarisKe, 5).Value = 0
    End If
    Sheets>NamaWorksheet).Cells(BarisKe, 6).Value = myJSSData.GetBiayaPenalty(i)
    Sheets>NamaWorksheet).Cells(BarisKe, 7).Value = myJSSData.GetBiayaPenalty(i) *
Sheets>NamaWorksheet).Cells(BarisKe, 5).Value
    Sheets>NamaWorksheet).Cells(BarisKe, 8).Value = myJSSData.GetBiayaProduksi(i) +
Sheets>NamaWorksheet).Cells(BarisKe, 7).Value
    BarisKe = BarisKe + 1
Next i

End Sub

Public Function AmbilUrutanSolusiKontinyu(ByVal urutanKe As Integer) As Double
    AmbilUrutanSolusiKontinyu = UrutanSolusiKontinyu(urutanKe)
End Function

Public Function GetUrutanSolusiKontinyu() As Variant
    GetUrutanSolusiKontinyu = UrutanSolusiKontinyu
End Function

Public Function AmbilUrutanSolusiInteger(ByVal urutanKe As Integer) As Integer
    AmbilUrutanSolusiInteger = UrutanSolusiInteger(urutanKe)
End Function

Public Function GetUrutanSolusiInteger() As Variant
    GetUrutanSolusiInteger = UrutanSolusiInteger
End Function

```

