



UNIVERSITAS INDONESIA

**MODEL PENJADWALAN JOB SHOP MESIN PARALEL
DENGAN MENGGUNAKAN METODE TABU SEARCH**

TESIS

KRISHTINA KUMALA DEWI

1006735201

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



UNIVERSITAS INDONESIA

**MODEL PENJADWALAN JOB SHOP MESIN PARALEL
DENGAN MENGGUNAKAN METODE TABU SEARCH**

TESIS

Diajukan sebagai syarat untuk memperoleh gelar Magister Teknik

KRISHTINA KUMALA DEWI

1006735201

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

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Krishtina Kumala Dewi

NPM : 1006735201

Tanda Tangan :



Tanggal : 23 Juni 2012

HALAMAN PENGESAHAN

Tesis ini diajukan oleh :

Nama : Krishtina Kumaladewi
NPM : 1006735201
Program Studi : Teknik Industri
Judul Tesis : Model Penjadwalan *Job Shop* pada Mesin Paralel dengan Menggunakan Metode *Tabu Search*

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

DEWAN PENGUJI

Pembimbing : Ir. Isti Surjandari, PhD (.....)

Pembimbing : Ir. Amar Rachman, MEIM (.....)

Penguji : Arian Dhini, ST, MT (.....)

Penguji : Ir. Erlinda Muslim, MEE (.....)

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

Penguji : Ir. Boy Nurtjahyo, MSIE (.....)

Ditetapkan di : Depok

Tanggal : 23 Juni 2012

KATA PENGANTAR

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

1. Ibu Ir. Isti Surjandari, Ph.D dan Ir. Amar Rachman, MEIM selaku dosen pembimbing yang telah banyak menyediakan waktu, pikiran, dan kesabarannya yang luar biasa untuk memberikan motivasi, arahan, semangat, dan doa dalam menyelesaikan penelitian ini.
2. Kedua orang tua dan semua anggota keluarga tercinta atas segala doa, motivasi, dukungan serta bantuan dalam penyusunan tesis ini.
3. Segenap dosen dan karyawan Departemen Teknik Industri Universitas Indonesia yang telah membantu selama masa perkuliahan.
4. Rekan-rekan S2 TI UI Depok 2010 untuk segala waktu, canda tawa, dan bantuan yang telah diberikan selama masa penyusunan tesis.
5. Semua pihak yang telah membantu dalam proses pembuatan tesis ini.

Penulis menyadari dalam penyusunan tesis ini masih terdapat kekurangan, karena itu penulis tidak menutup diri terhadap saran dan kritik yang membangun demi kesempurnaan tesis ini. Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 23 Juni 2012

Penulis

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

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

Nama : Krishtina Kumala Dewi
NPM : 1006735201
Program Studi : Teknik Industri
Departemen : Teknik Industri
Fakultas : Teknik
Jenis karya : Tesis

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 :

“Model Penjadwalan Job Shop Mesin Paralel Dengan Menggunakan Metode Tabu Search”

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-eksklusif ini, Universitas Indonesia berhak menyimpan, mengalihmedia / 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 Juni 2012

Yang menyatakan



(Krishtina Kumala Dewi)

ABSTRAK

Nama : Krishtina Kumala Dewi
Program Studi : Teknik Industri
Judul : Model Penjadwalan Job Shop Mesin Paralel Dengan Menggunakan Metode Tabu Search

Agar dapat merespon permintaan pelanggan secara cepat perlu disiapkan sejumlah persediaan yang cukup besar, supaya tersedia pada saat dibutuhkan. Penelitian ini mengembangkan model penjadwalan yang terdiri dari dua model. Model satu bertujuan melakukan penjadwalan untuk pemenuhan *due date* pada lingkungan Job Shop Paralel. Model dua bertujuan melakukan penjadwalan untuk penentuan *due date*. Pada sistem *Job Shop* Paralel dinamis yang memproses multi-item berstruktur multi-level, berdasarkan kinerja kualitas pada lingkungan produksi *Just In Time* (JIT). Pendekatan yang digunakan dalam penelitian ini adalah pendekatan mundur (*backward approach*) untuk pemenuhan *due date* dan pendekatan random untuk penentuan *due date* serta menggunakan metode *Tabu Search* dengan kriteria minimasi *total actual flow time* dengan memperhatikan tingkat cacat (*defect rate*) yang terjadi pada setiap tahapan proses yang dilalui. Peningkatan jumlah item dan jumlah level akan memberikan peningkatan jumlah alternatif jadwal yang berdampak pada *CPU time* yang diperlukan untuk mendapatkan solusi yang dapat memberikan *total actual flow time* yang minimum, namun dalam penelitian ini hanya terjadi untuk satu level multi item saja. Untuk dua level dan seterusnya tidak memberikan alternatif jadwal yang bervariasi untuk tingkatan item maupun level. Hal ini disebabkan karena inisialisasi solusi tetangga, panjang tabu list dan jumlah iterasi yang ditetapkan diawal membuat pencarian solusi berhenti di titik lokal optimum.

Kata kunci :
Job Shop, Tabu Search, Mesin Paralel, JIT, due date.

ABSTRACT

Name : Krishtina Kumala Dewi
Study Program : Industrial Engineering
Title : Model Penjadwalan Job Shop Mesin Paralel Dengan Menggunakan Metode Tabu Search

In order to response the demand from customer quickly, it needs to have a large size of inventory. This research developes two scheduling model. The first models is to fulfill the due date in the paralel job shop manufacturing and the second is to make the due date itself. In the dynamic paralel job shop system which is produce the multi item structured by multi level, according to the quality performance in the just in time manufacturing. The research uses the backward approach to fulfill the due date and the random approach to make the due date using Tabu Search Method which objective is to minimize total actual flow time concern on defect rate. The increase of item and level affect the increase of scheduling alternatif and cpu time, but in this research it only happens for one level and multi item. For two level and so on it doesn't give any varians scheduling alternative for level or item invrease. It might becuase of the inisialisation of neighbor, the lenght of tabu list and iteration which made in the beginning and it makes the solution stop at the optimum local search.

Keywords:

Job Shop, Tabu Search, Paralelel Machine , JIT, due date.

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PERSETUJUAN.....	iii
KATA PENGANTAR	iv
HALAMAN PERSETUJUAN PUBLIKASI TUGAS AKHIR.....	v
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN.....	xii
BAB	1
PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Diagram Keterkaitan Masalah	4
1.3 Perumusan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Ruang Lingkup Penelitian.....	5
1.6 Batasan Masalah	7
1.7 Metodologi Penelitian.....	7
1.8 Sistematika Penulisan	9
BAB 2 LANDASAN TEORI	
2.1 Lingkungan Produksi <i>Just In Time</i>	12
2.2 Konsep Penjadwalan.....	12
2.2.1 Pendekatan Penjadwalan.....	13
2.2.2 Penjadwalan <i>Job Shop</i>	14
2.2.3 Mesin Paralel	16
2.3 Konsep dan Definisi <i>Due Date</i>	16
2.4 <i>Total Actual Flow Time</i>	18
2.5 Algoritma <i>Tabu Search</i>	19
2.6 Penjadwalan Ulang (<i>Rescheduling</i>).....	25

BAB 3 PENGEMBANGAN MODEL

3.1 Skenario Pengembangan Model	27
3.2 Notasi dan Definisi	27
3.3 Model Penjadwalan Pemenuhan <i>Due Date</i>	29
3.4 Model Penjadwalan Penentuan <i>Due Date</i>	36
3.5 Pelaksanaan Pengumpulan Data	31
3.6 Hasil Pengumpulan dan Pengolahan Data	32

BAB 4 PENGUJIAN DAN ANALISA MODEL

4.1 Verifikasi dan Validasi Model	40
4.1.1 Set Data Pengujian Kondisi Statis	40
4.1.2 Set Data Pengujian Kondisi Dinamis.....	42
4.1.3 Hasil Pengujian Kondisi Statis Pemenuhan <i>Due-Date</i>	44
4.1.4 Hasil Pengujian Kondisi Dinamis Pemenuhan <i>Due-Date</i>	45
4.1.5 Hasil Pengujian Kondisi Statis Penentuan <i>Due-Date</i>	47
4.1.5 Hasil Pengujian Kondisi Dinamis Penentuan <i>Due-Date</i>	51
4.2 Pengujian Model	52
4.2.1 Hasil Pengujian Pemenuhan <i>Due-Date</i>	55
4.2.2 Hasil Pengujian Penentuan <i>Due-Date</i>	57
4.3 Analisis Model	59
4.4 Analisis Perbandingan Performa Hasil Penelitian Sebelumnya	60

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan	62
5.2 Saran	62

DAFTAR REFERENSI	63
------------------------	----

LAMPIRAN

DAFTAR TABEL

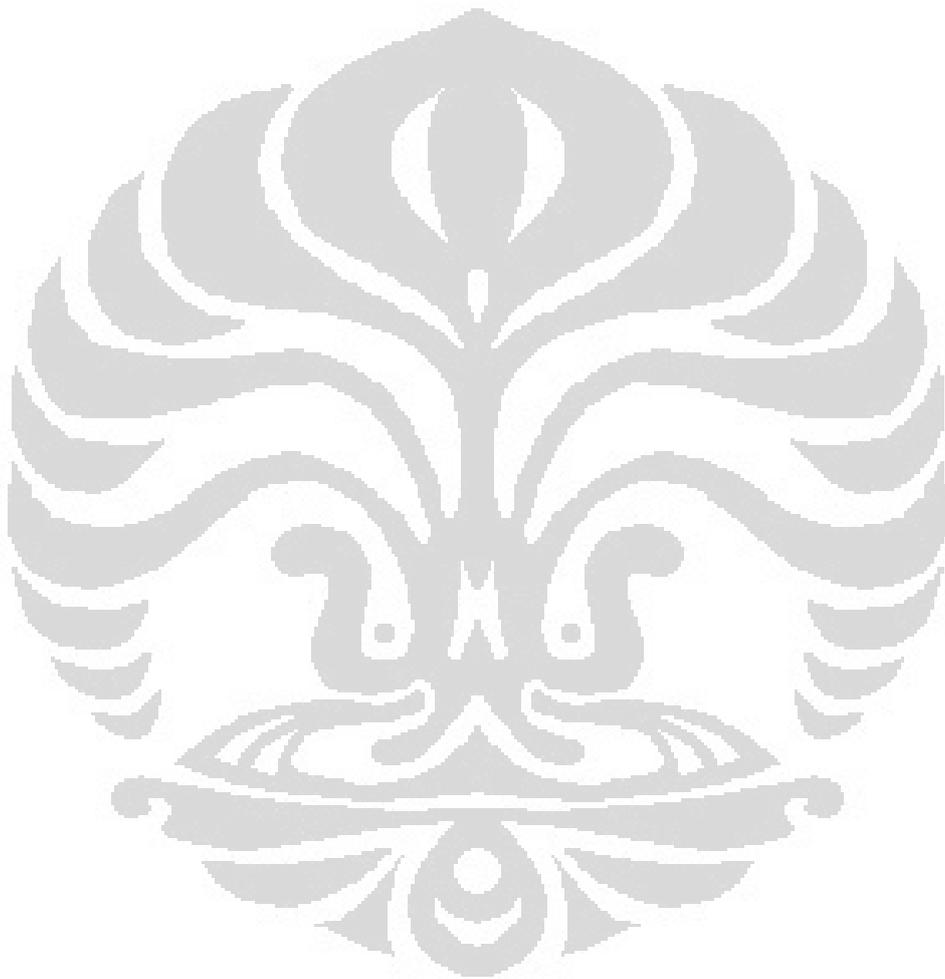
Nomor Tabel	Halaman
Tabel 4.1 Data Mesin Tersedia.....	41
Tabel 4.2 Data Item, Kuantitas dan <i>Due Date</i>	41
Tabel 4.3 Waktu Proses, Set-Up dan Tingkat Cacat	42
Tabel 4.4 Data Item, Kuantitas dan <i>Due Date</i> Pesanan Baru.....	43
Tabel 4.5 Waktu Proses, Set-Up dan Tingkat Cacat	42
Tabel 4.6 Saat Mulai Pengerjaan Masing-masing Item.....	44
Tabel 4.7 Waktu Mulai dan Selesai Setiap Operasi (menit ke-)	44
Tabel 4.8 Saat Mulai Pengerjaan Masing-masing Pesanan Baru	42
Tabel 4.9 Waktu Mulai dan Selesai Setiap Operasi (menit ke-)	42
Tabel 4.10 Waktu Mulai dan Selesai Operasi Penentuan <i>Due Date</i>	47
Tabel 4.11 <i>Due Date</i> dari Setiap Item	48
Tabel 4.12 Waktu Mulai dan Selesai Penentuan <i>Due Date</i> Dinamis	51
Tabel 4.13 <i>Due Date</i> dari Setiap Item	52
Tabel 4.14 Alternatif Jadwal Pemenuhan <i>Due Date</i>	55
Tabel 4.15 CPU <i>Time</i> Pemenuhan <i>Due Date</i> (dalam detik).....	56
Tabel 4.16 Alternatif Jadwal Penentuan <i>Due Date</i>	57
Tabel 4.17 CPU <i>Time</i> Pemenuhan <i>Due Date</i> (dalam detik).....	58
Tabel 4.18 Perbandingan <i>Due Date</i> Kondisi Statis	61
Tabel 4.19 Perbandingan <i>Due Date</i> Kondisi Dinamis	61

DAFTAR GAMBAR

Nomor Gambar	Halaman
Gambar	1.1
Diagram Keterkaitan Masalah	6
Gambar	1.2
Diagram Alir Metodologi Penelitian	10
Gambar	2.1
Model Rute <i>Job Shop</i>	15
Gambar	2.2
Kondisi <i>Job Shop</i> Mesin Paralel.....	17
Gambar	2.3
Waktu Tinggal Aktual (<i>Actual flow time</i>) Pekerjaan	18
Gambar	3.1
Item Berstruktur Multi-Level	31
Gambar	3.2
Diagraph dari Produk Akhir dan Komponennya.....	31
Gambar	3.3
Algoritma <i>Tabu Search</i>	35
Gambar	4.1
Struktur Setiap Produk	41
Gambar	4.2
Struktur Pesanan Baru	10
Gambar	4.3
<i>Gantt Chart</i> Akhir Pemenuhan <i>Due Date</i> Kondisi Statis.....	49
Gambar	4.4
<i>Gantt Chart</i> Akhir Pemenuhan <i>Due Date</i> Kondisi Dinamis	50
Gambar	4.5
<i>Gantt Chart</i> Akhir Penentuan <i>Due Date</i> Kondisi Statis.....	53
Gambar	4.6
<i>Gantt Chart</i> Akhir Penentuan <i>Due Date</i> Kondisi Dinamis	54
Gambar	4.7
Grafik Alternatif Jadwal Pemenuhan <i>Due Date</i>	56
Gambar 4.8	57
Grafik <i>CPU Time</i> Pemenuhan <i>Due Date</i>	57
Gambar	4.9
Grafik Alternatif Jadwal Penentuan <i>Due Date</i>	58
Gambar 4.10	59
Grafik <i>CPU Time</i> Penentuan <i>Due Date</i>	59

DAFTAR LAMPIRAN

Lampiran A	Set Data
Lampiran B	<i>Source Code</i>



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pertumbuhan dunia industri saat ini menyebabkan semakin ketatnya persaingan yang harus dihadapi dalam merespon kebutuhan konsumen atau pelanggan. Agar dapat merespon permintaan pelanggan secara cepat perlu disiapkan sejumlah persediaan yang cukup besar, supaya tersedia pada saat dibutuhkan. Namun, disisi lain untuk meminimasi lamanya suatu pekerjaan berada di lantai pabrik (*shop time*) justru perlu dilakukan dengan meminimasi jumlah persediaan seperti yang dikemukakan oleh Baker (1974), Karmarkar (1987) dan Halim (1993). Dalam penelitian ini juga menyatakan bahwa ketepatan pemenuhan *due-date* cenderung dianggap lebih penting dari pada meminimasi lamanya pesanan berada di lantai pabrik. Hal ini terjadi karena pemenuhan *due-date* secara tepat lebih terkait dengan kepuasan pelanggan (*customer satisfaction*).

Sistem produksi tepat waktu (*Just In Time*) merupakan sistem produksi yang dapat mengakomodir kondisi ini. Di mana pada satu sisi bertujuan melakukan pemenuhan ketepatan *due-date* dan disisi lain juga berusaha untuk meminimasi jumlah persediaan. Karena konsep sistem produksi tepat waktu bertujuan untuk menghasilkan barang atau produk pada waktu dan kuantitas yang tepat serta kualitasnya, Vollmann (2005).

Zhang et al (2006) meneliti tentang penjadwalan *job shop* dengan metode *Tabu Search* dengan struktur tetangga yang baru dengan tujuan meminimasi *makespan*. Dalam penelitian ini dikatakan bahwa *Tabu Search* sebagai salah satu metode yang memberikan hasil paling efektif untuk mengatasi masalah *job shop*. Bagaimana pun juga struktur tetangga dan strategi evaluasi 'move' memainkan peranan penting dalam efektifitas dan efisiensi tersebut. Glover (1995) sebagai pencetus awal penggunaan metode *Tabu Search* dalam penelitiannya mengatakan bahwa di dalam kelas metode metaheuristik, *Tabu Search* menjadi salah satu metode yang paling menjanjikan untuk masalah penjadwalan *job shop* dengan

kriteria meminimasi *makespan*. Zhang et al (2011) meneliti tentang Algoritma Genetika dengan prosedur *Tabu Search* untuk penjadwalan fleksibel *job shop* dengan memperhatikan batasan waktu proses dan juga waktu transportasi. Tujuan dari penelitian ini adalah untuk meminimasi *makespan* dan solusi penyimpanan. Dalam penelitian ini *Tabu Search* digunakan untuk mencari, mengevaluasi dan memperbaiki penjadwalan dengan sejumlah iterasi yang telah ditentukan. Liang et al (2010) meneliti tentang Algoritma Hibrida *Tabu Search* untuk multi objektif masalah penjadwalan *job shop* fleksibel. Tujuannya untuk meminimasi *makespan*, total beban mesin dan juga beban mesin kritis. Dengan penetapan solusi tetangga yang efektif dan dikombinasikan dengan aturan adaptif lain dapat dihasilkan performa pencarian lokal optima yang baik.

Mosheiov et al (2006) telah mengembangkan model penentuan *due-date* dan masalah penjadwalan aktivitas perawatan. Model yang dikembangkan bertujuan untuk menentukan urutan pekerjaan, *due-date* dan aktivitas perawatan. Adapun ukuran performansi yang digunakan meminimumkan total biaya pekerjaan yang diselesaikan lebih awal (*earliness*), total biaya keterlambatan (*tardiness*) dan total biaya penyelesaian pekerjaan tepat pada *due-datenya*.

Huang (2010), menggunakan pendekatan heuristik untuk menyelesaikan masalah *multi-objective job shop* dengan melakukan *lot-splitting* pada lingkungan JIT. Fungsi tujuan yang akan diminimasi total stok, mesin menunggu (*machine idle*), dan biaya pengangkutan. Di mana biaya pengangkutan akan meningkat seiring dengan peningkatan jumlah *batch*, dari hasil penelitian ini dapat diketahui jumlah *batch* yang dapat memberikan total biaya minimum dari ketiga komponen biaya yang sudah ditetapkan. Namun, dalam penelitian ini belum memasukan unsur *due date* baik dalam hal pemenuhan maupun penentuannya.

Penelitian penerapan genetik heuristik untuk meneliti masalah penjadwalan mesin pemrosesan paralel *batch* dengan ukuran pekerjaan yang berubah, di mana setiap mesin dapat memproses sekelompok pekerjaan secara bersamaan sebagai sebuah *batch* dengan tujuan meminimasi *makespan* (Kashan et al.,2008). Penelitian ini belum memasukan *due date* sebagai ukuran kinerja penjadwalan dan

kedatangan pekerjaan yang mungkin terjadi sepanjang horizon waktu perencanaan.

Penentuan *due date* dan penjadwalan mesin paralel dengan penurunan pekerjaan, dengan tujuan untuk meminimasi total *due date*, *earliness* dan *tardiness* penalti (Cheng et al.,2007). Dalam penelitian ini diasumsikan, di mana pekerjaan tergantung tingkat penurunan yang sama untuk semua pekerjaan.

Xia et al., (2008) mengembangkan suatu prosedur heuristik untuk menyelesaikan permasalahan pengurutan pekerjaan dan penentuan *due date* pada lingkungan JIT. Untuk meminimumkan kombinasi linier dari tiga jenis pinalti, yaitu penyelesaian lebih awal pekerjaan (*penalty on job earliness*), pinalti keterlambatan pekerjaan (*penalty on job tardiness*) dan pinalti penentuan *due date* yang panjang (*penalty associated with long due date assignment*). Dalam penelitian ini diasumsikan waktu proses pekerjaan tidak pasti, tetapi permasalahan yang dibahas pada kondisi mesin tunggal.

Penyelesaian permasalahan penjadwalan dan penentuan *due date* yang bertujuan untuk meminimasi pekerjaan yang selesai lebih awal, terlambat, tertunda, penentuan *due date* dan biaya pengiriman atau transfer batch pada mesin tunggal, di mana *due date* dapat dikendalikan telah dilakukan oleh Shabtay, (2010). Dalam penelitian ini, strategi penentuan *due date* optimal sebagai fungsi dari pengurutan pekerjaan dan pembagian urutan pekerjaan kedalam batch-batch. Selain itu pendekatan penentuan *due date* yang digunakan berdasarkan DIF (*diferent due date*) method, sehingga setiap pekerjaan akan memiliki *due date* yang berbeda.

Penentuan *due date* dan penjadwalan pada lingkungan produk JIT dengan ukuran pekerjaan yang sama, di mana penelitian yang dilakukan untuk kondisi mesin tunggal dan mesin paralel dengan tujuan untuk meminimasi total bobot keterlambatan dan penyelesaian lebih awal dari suatu pekerjaan dan biaya *due date* telah dilakukan oleh Tuong dan Soukhal (2010). Pendekatan yang digunakan berdasarkan pada CON (*Common Due date*) method, sehingga semua pekerjaan dianggap memiliki *due date* yang sama.

Pengembangan penentuan *due date* dan aturan keputusan penjadwalan pada lingkungan *job shop* dinamis, melalui pendekatan pemodelan dan analisa simulasi telah dilakukan oleh Vinod dan Sridharan (2011). Metode penentuan *due date* dikembangkan adalah pemrosesan dinamis ditambah waktu tunggu (*waiting time*), total pengerjaan (*Total Work Content-TWK*), total waktu pengerjaan dinamis (*Dynamic Total Work Content-DTWK*) dan metode pengerjaan acak (*Random Work Content-RWK method*). Namun dalam penelitian ini, semua pekerjaan diasumsikan tersedia pada waktu sama dengan nol dan belum memperhatikan keterkaitan waktu penyelesaian antar komponen penyusun produk pada masing-masing tingkat.

Dari uraian tersebut diatas terlihat bahwa adanya kebutuhan untuk pengembangan model penjadwalan pada sistem produksi *job shop* paralel dinamis yang memproduksi multi item, di mana setiap item berstruktur multi level dengan memperhatikan *defect rate* yang dapat dihasilkan dari suatu proses produksi. Dengan tujuan meminimasi total waktu tinggal aktual pada lingkungan produksi *Just In Time (JIT)*. Penggunaan metode *Tabu Search* dikarena metode ini dapat memberikan hasil yang efektif untuk optimasi dalam pencarian lokal disepanjang horizon perencanaan yang mungkin dilakukan. Unsur *defect rate* perlu dipertimbangkan dalam proses penjadwalan karena memiliki pengaruh terhadap produk akhir yang akan dihasilkan, pada setiap tahapan proses produksi dan waktu penyelesaian pekerjaan.

1.2 Diagram Keterkaitan Masalah

Gambaran sistematis yang lebih menyeluruh mengenai keterkaitan permasalahan dapat dilihat pada gambar 1.1, yang merupakan hubungan antar gejala permasalahan yang bentuk suatu permasalahan yang harus diselesaikan.

1.3 Perumusan Masalah

Penelitian ini mengembangkan model penjadwalan yang terdiri dari dua model. Model satu bertujuan melakukan penjadwalan untuk pemenuhan *due date*

pada lingkungan Job Shop Paralel. Model dua bertujuan melakukan penjadwalan untuk penentuan *due date*. Pada sistem *Job Shop* Pararel dinamis yang memproses multi-item berstruktur multi-level, berdasarkan kinerja kualitas pada lingkungan produksi *Just In Time* (JIT). Pendekatan yang digunakan dalam penelitian ini adalah pendekatan mundur (*backward approach*) untuk pemenuhan *due date* dan pendekatan maju (*forward*) untuk penentuan *due date* serta menggunakan metode Tabu Search dengan kriteria minimasi *total actual flow time*. Dengan memperhatikan tingkat cacat (*defect rate*) yang terjadi pada setiap tahapan proses yang dilalui.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah menghasilkan model penjadwalan untuk melakukan pemenuhan (Model satu) dan penentuan (Model dua) *due date* pada sistem produksi job shop mesin paralel yang memproses multi-item berstruktur multi level dan multi *due date*, dengan memperhatikan *defect rate*. Untuk mendapatkan *total actual flow time* yang minimum dengan menggunakan metode *tabu search*.

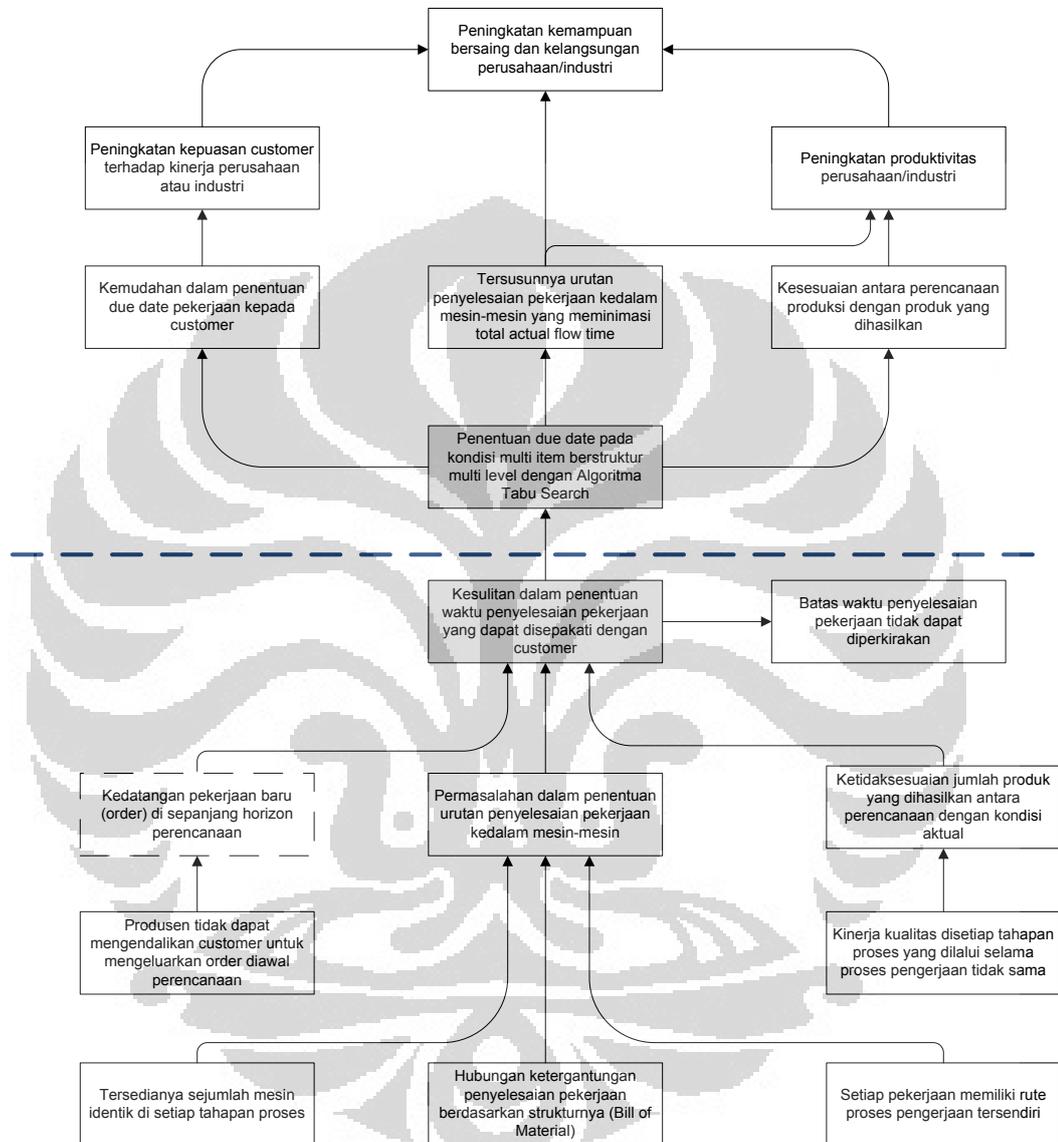
1.5 Ruang Lingkup Penelitian

Dalam penelitian ini, terdapat beberapa hal yang menunjukkan karakteristik masalah dalam penelitian ini adalah :

- *Defect rate* yang terjadi disetiap tahapan proses produksi bersifat deterministik (diketahui secara pasti).
- Waktu pemrosesan setiap operasi bersifat deterministik.
- Sistem produksi yang ditinjau adalah sistem produksi *job shop*. Di mana masing-masing item memiliki urutan (*routing*) produksi dan struktur produksi sendiri yang berbeda dengan waktu *set-up* dan waktu proses diketahui secara pasti.

Kedatangan pekerjaan (*order*) diasumsikan dinamis dalam arti setiap pekerjaan tidak harus datang pada saat waktu nol, tetapi dapat datang sepanjang horizon waktu perencanaan. Untuk setiap pekerjaan yang telah diterima tidak terjadi

perubahan *due-date* pada masalah penjadwalan, perubahan prioritas, atau pun pembatalan pekerjaan.



Gambar 1.1 Diagram Keterkaitan Masalah

- Setiap pekerjaan memiliki *due-date* tertentu yang harus dipenuhi dan tidak diperbolehkan terjadi keterlambatan pekerjaan. Pada permasalahan penjadwalan untuk pemenuhan *due date*.

- Setiap pekerjaan memiliki *due date* yang tidak diketahui dan dapat dinegosiasikan dengan konsumen, pada masalah penentuan *due date*.

Untuk lebih spesifiknya, sistem produksi yang menjadi objek penelitian adalah jenis job shop yang memiliki karakteristik sebagai berikut;

- Terdapat beberapa tahapan proses produksi (*multi-stage*).
- Tiap *stage* terdiri dari lebih dari satu unit mesin (*parallel*), dan
- Batas waktu penyelesaian pekerjaan (*due-date*) tidak sama (*multi due-date*) pada masalah pemenuhan *due date*.
- *Due date* dapat dinegosiasikan dengan konsumen, pada masalah penentuan *due date*.

1.6 Batasan Masalah

Untuk lebih memfokuskan pembahasan, ada beberapa asumsi yang digunakan dalam penelitian ini sebagai berikut.

- Data sekunder dari penelitian sebelumnya digunakan untuk mengevaluasi performa algoritma yang dikembangkan.
- Fungsi tujuan yang digunakan dalam penelitian ini adalah meminimumkan *total flow time*.
- Suatu mesin pada waktu tertentu hanya memproses satu operasi saja.
- Selama proses operasi tidak terjadi gangguan pada fasilitas produksi.
- Setiap mesin memiliki fungsi khusus (*dedicated function*) yang tidak dipertukarkan dengan mesin lainnya.
- Waktu transportasi yang terjadi satu stasiun kerja ke stasiun kerja yang lainnya diabaikan.
- Penyusunan algoritma menggunakan MatLab.

1.7 Metodologi Penelitian

Penelitian ini merupakan pengembangan dari beberapa penelitian dibidang penjadwalan dan penentuan *due date* dilingkungan produksi JIT. Tujuan utama dari penelitian ini adalah menghasilkan suatu model penentuan *due date* melalui penjadwalan untuk job shop dinamis yang memproses multi-item berstruktur multi-level dengan tujuan meminimasi *total actual flow time*. Di mana sistem

produksi yang ditinjau berlingkungan *just in time* serta memperhatikan terjadinya tingkat cacat dari setiap tahap proses produksi, yang dinyatakan dengan persentase terjadinya cacat pada setiap proses yang dilalui. Adapun kerangka penelitian, sebagai berikut.

1. Studi Pendahuluan

Penelitian ini dilakukan melalui studi literatur dan pengamatan terjadi di beberapa industri berbasis *job shop* dengan kedatangan pekerjaan (*job*) bersifat dinamis.

2. Perumusan Masalah

Dari berbagai permasalahan yang diperoleh dari studi literatur maupun permasalahan ditemukan di dunia industri, maka dirumuskan masalah yang akan diselesaikan dalam penelitian ini.

3. Tujuan Penelitian

Dengan inti permasalahan yang ada di industri berbasis *job order* khususnya dan dilakukan studi literatur baik melalui jurnal internasional, laporan penelitian maupun buku teks, maka dirumuskan tujuan dilakukannya penelitian ini.

4. Pengembangan Model Kondisi Statis

Proses pengembangan model penentuan *due date* untuk kondisi dinamis didalam penelitian ini, diawali dari pengembangan model untuk kondisi statis terlebih dahulu. Di mana pada model kondisi statis tidak kedatangan pekerjaan baru yang datang selama horizon perencanaan.

5. Pengembangan Model Kondisi Dinamis

Model penentuan *due date* pada kondisi dinamis merupakan model yang menjadi tujuan penelitian ini. Di mana dalam kondisi dinamis izinkan terjadinya kedatangan pekerjaan baru di sepanjang horizon waktu perencanaan.

6. Pengujian dan Analisa Model

Untuk memastikan model bekerja sesuai dengan tujuan yang diharapkan, maka dilakukan pengujian algoritma penyelesaian model baik pada kondisi statis maupun dinamis. Pengujian algoritma ini juga sebagai bagian dari proses verifikasi dan validasi model. Apabila model yang

dikembangkan belum dapat memberikan hasil yang diharapkan, maka akan dilakukan perbaikan terhadap algoritma penyelesaian.

7. Kesimpulan dan Saran

Langkah terakhir dalam penelitian ini adalah menyimpulkan hasil-hasil yang dicapai dalam penelitian dan memberikan saran

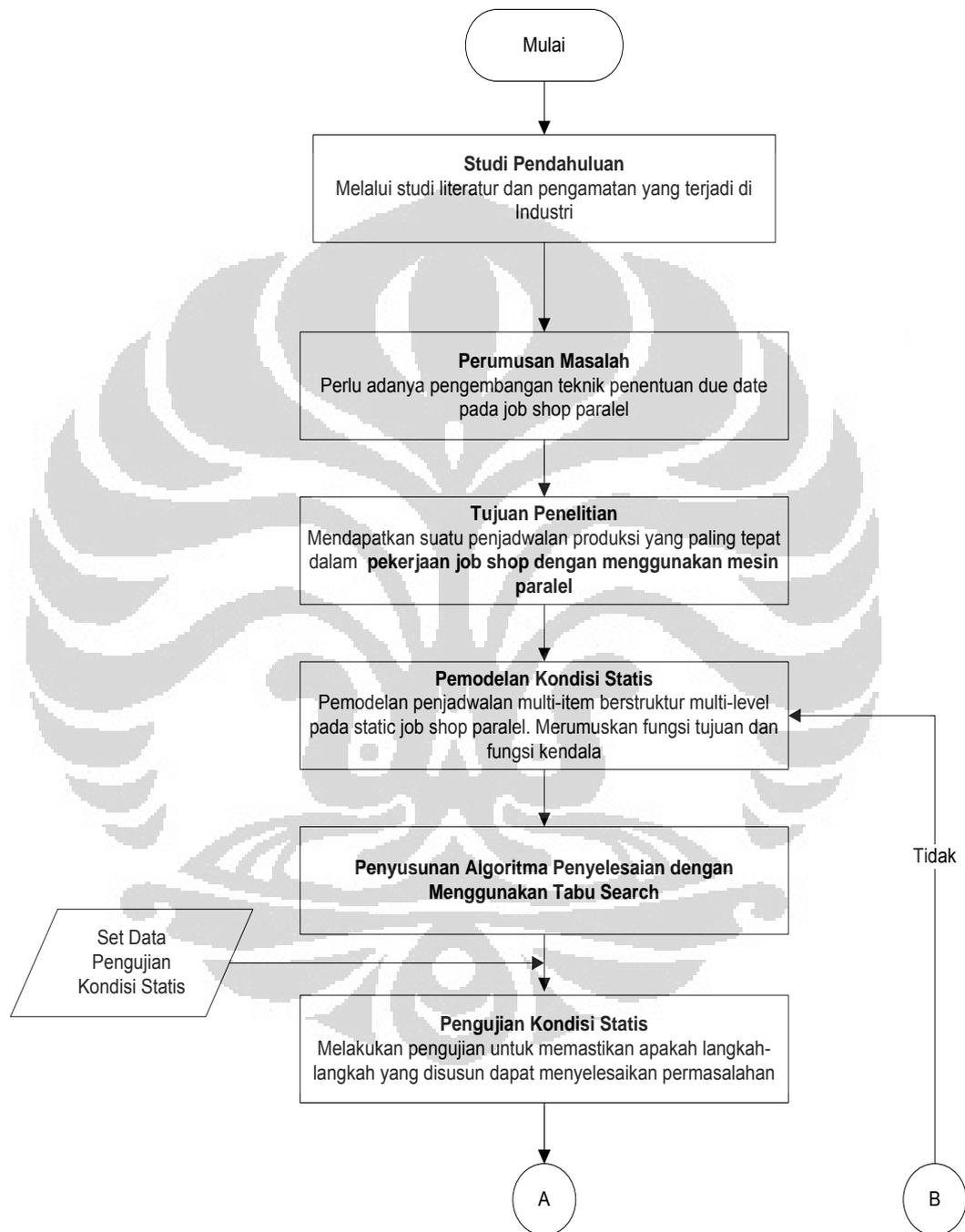
Alur tahapan penelitian yang digunakan, seperti ditunjukkan pada Gambar 1.2.

1.8 Sistematika Penulisan

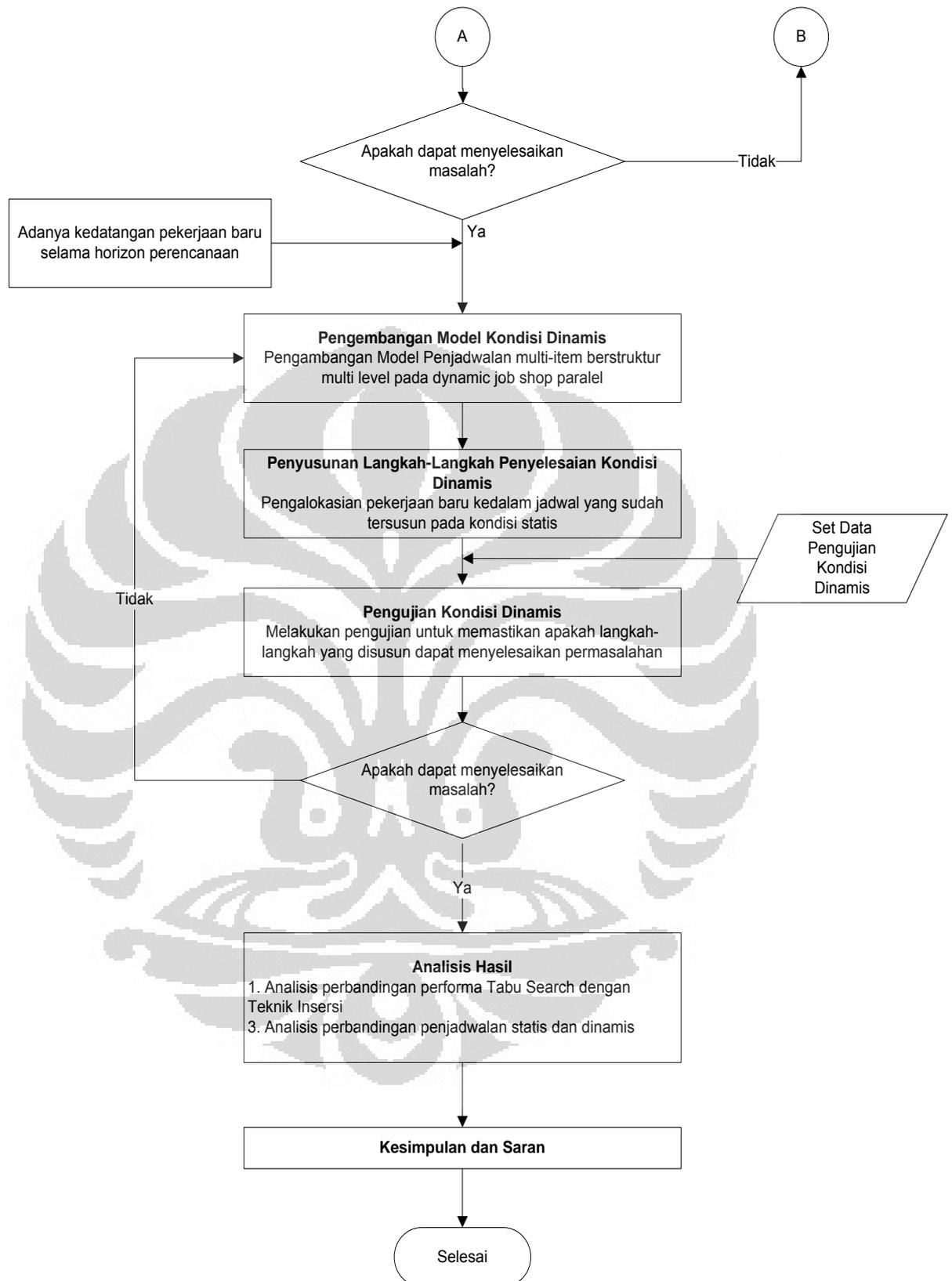
Penelitian tugas akhir ini disusun dalam beberapa bab dengan tujuan memudahkan alur proses berpikir, dengan sistematika sebagai berikut:

- Bab I Pendahuluan.
Bab ini menjelaskan latar belakang masalah, perumusan masalah, tujuan penelitian, ruang lingkup pembahasan, batasan masalah, metodologi penelitian dan sistematika penulisan.
- Bab II Tinjauan Teori.
Bab ini menjelaskan teori-teori terkait dan penelitian-penelitian sebelumnya yang berhubungan dengan penelitian ini. Sebagai dasar teori yang akan digunakan didalam proses pencarian solusi dari permasalahan yang menjadi objek pada penelitian ini.
- Bab III Pengembangan Model.
Bab ini menjelaskan pengembangan model satu dan model dua. Dimana model satu pengembangan model untuk kondisi pemenuhan *due date*, sedangkan pada model dua pengembangan model untuk penentuan *due date*. Algoritma pencarian solusi pada kedua model, menggunakan metode *tabu search* dengan ukuran performansi *total actual flow time*.
- Bab IV Pengujian dan Analisa Model.
Bab ini menjelaskan proses verifikasi dan validasi dari model yang dikembangkan, dengan menggunakan hipotesis data. Proses ini, dilakukan untuk memastikan algoritma yang dikembangkan dapat bekerja dengan baik, untuk menyelesaikan permasalahan.
- Bab V Kesimpulan dan Saran.

Bab ini berisi kesimpulan dari hasil penelitian yang dilakukan dan saran-saran untuk penelitian selanjutnya yang dapat dikembangkan dari penelitian ini.



Gambar 1.2 Diagram Alur Penelitian



Gambar 1.2 Diagram Alur Penelitian

BAB 2

TINJAUAN TEORI

2.1. Lingkungan Produksi Just In Time (JIT)

Sistem *Just In Time* (JIT) dikembangkan di Toyota Motor Company di Jepang. Meskipun Schonberger (1982) mengisyaratkan bahwa JIT mungkin telah ada sejak 20 tahun lalu atau lebih dalam industri galangan kapal Jepang. Namun, aplikasi modern dari JIT dipopulerkan pada pertengahan tahun 1970-an di Toyota oleh Mr Taiichi Ohno, wakil presiden Toyota dan beberapa kolognya. Selanjutnya konsep ini mulai diadopsi oleh industri-industri di Amerika Serikat.

Sistem JIT memiliki tujuan dasar untuk meningkatkan laba dan pengembalian investasi melalui pengurangan persediaan dan peningkatan kualitas. Tujuan ini dapat dicapai dengan menghilangkan berbagai pemborosan dan perencanaan produksi yang sesuai dengan kebutuhan dalam arti tepat dari segi kuantitas dan kualitas. Sistem ini mengintegrasikan permasalahan persediaan dan pejadwalan dalam satu pertimbangan dengan tujuan meminimumkan tingkat persediaan yang dipandang sebagai pemborosan, sekaligus memenuhi kebutuhan konsumen atau pelanggan yang dinyatakan dalam batas waktu pemenuhan order (due-date).

Dalam penelitian ini, model penjadwalan yang diusulkan diarahkan pada lingkungan JIT, sehingga tidak diperkenankan terjadinya keterlambatan penyelesaian pekerjaan, serta pekerjaan yang sudah selesai akan segera langsung dikirimkan ke pelanggan sesuai dengan batas waktu yang diinginkan oleh pelanggan, pada sistem produksi JIT. Dengan mempertimbangkan kinerja kualitas pada setiap tahapan proses yang dilalui, karena faktor ini memiliki pengaruh yang cukup signifikan dalam pemenuhan due-date.

2.2. Konsep Penjadwalan

Penjadwalan adalah suatu proses untuk memutuskan bagaimana menjalankan berbagai sumber daya pada berbagai tugas yang mungkin. Waktu dapat ditentukan atau dibuat mengambang sebagai bagian dari urutan

kejadian. Teori penjadwalan fokus pada pemodelan matematis yang berhubungan dengan proses penjadwalan. Pengembangan model mengarah pada teknik pencarian solusi dan permasalahan secara praktikal, yang berlanjut dengan hubungan antara teori dan praktek, Baker dan Trietsch (2009). Pemodelan yang dilakukan bertujuan untuk mencakup semua struktur masalah kedalam bentuk matematis.

Terdapat tiga jenis pengambilan keputusan yang lazim dilakukan yaitu *turnaround*, *timeliness* dan *throughput*. *Turnaround* mengukur waktu yang dibutuhkan untuk menyelesaikan pekerjaan yang tersedia. *Timeliness* mengukur kesesuaian penyelesaian pekerjaan tertentu selama tenggang waktu yang diberikan. Sedangkan *throughput* mengukur jumlah pekerjaan yang telah diselesaikan selama periode waktu yang telah ditetapkan (*due date*). 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 yang tanpa memerhatikan urutan kegiatan produksi, ada masalah lain yang disebut penjadwalan yang mana alokasi sumberdaya dan urutan pengerjaan menjadi sangat penting.

2.2.1 Pendekatan Penjadwalan

Terdapat dua pendekatan dasar dalam penjadwalan yang digunakan untuk menyusun suatu jadwal yaitu pendekatan maju (*forward approach*) dan pendekatan mundur (*backward approach*). Pendekatan maju adalah pengurutan pekerjaan yang bertolak dari arah sekarang (waktu sama dengan nol) dan bergerak menuju ke waktu yang akan datang, sedangkan penjadwalan mundur adalah penjadwalan yang dimulai dari *due-date* ke arah waktu nol. Pada pendekatan maju akan dihasilkan suatu jadwal yang layak, tetapi tidak menjamin *due-date* akan terpenuhi, sementara dengan pendekatan mundur akan diperoleh jadwal yang memenuhi *due-date*, tetapi tidak ada jaminan jadwal yang diperoleh tersebut layak. Pada kondisi keterlambatan penyelesaian pekerjaan tidak diijinkan, pendekatan yang paling tepat digunakan adalah penjadwalan dengan menggunakan pendekatan mundur (*backward approach*). Sedangkan pada

permasalahan penentuan *due date* maka pendekatan yang paling tepat digunakan adalah pendekatan maju (*forward*). Agar diperoleh waktu penyelesaian pekerjaan yang dimulai dari waktu nol sebagai *due date* yang akan diajukan kepada konsumen.

Penjadwalan *batch* dengan pendekatan mundur berarti urutan *batch* atau pekerjaan dijadwalkan secara mundur mulai dari *due-date*. *Batch* pertama terjadwal merupakan *batch* terakhir yang akan diproses sesuai dengan urutan waktu dan *batch* terakhir yang terjadwal adalah *batch* pertama yang akan diproses. Pendekatan mundur ini sesuai dengan konsep *just-in-time* karena item akan selesai tepat pada saat diperlukan yaitu pada saat *due-date*.

Untuk menyesuaikan dengan kondisi *just-in-time* serta model penjadwalan mundur, Halim dan Ohta (1993), telah mengembangkan kriteria performansi yang disebut dengan *total actual flow time*.

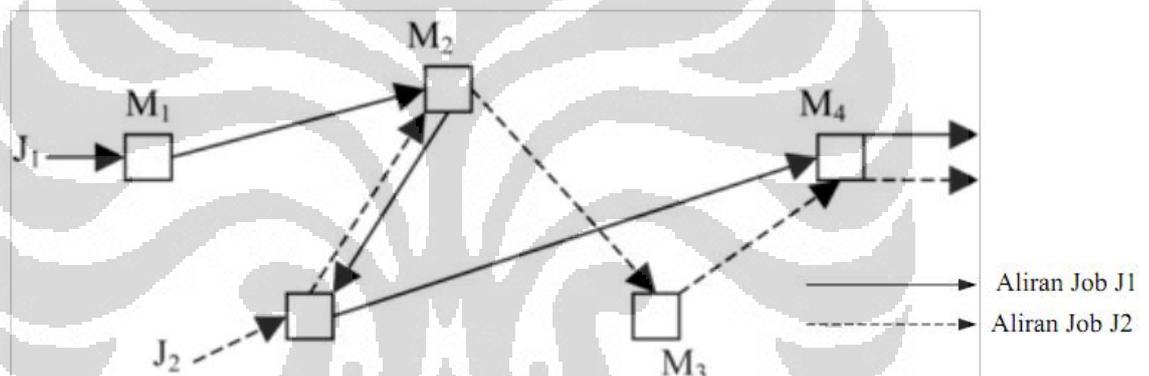
2.2.2 Penjadwalan *Job Shop*

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).

Elemen dari permasalahan *job shop* adanya sejumlah n pekerjaan (J), $j=1, \dots, n$, yang harus diproses pada sekumpulan M mesin. Pekerjaan J_j terdiri dari n_j operasi o_{ij} , $i=1, \dots, n_j$, yang harus diproses dalam suatu rute tertentu, o_{ij} operasi ke- i dari pekerjaan ke J_j . Di mana masing-masing operasi diproses pada mesin tertentu (*dedicated machine*). Sehingga lingkungan produk *job shop* memiliki karakteristik sebagai berikut:

- Terdapat sejumlah m mesin dan n pekerjaan.
- Setiap pekerjaan memiliki rute penyelesaian pemrosesan yang berbeda satu sama lain.
- Setiap operasi dalam pekerjaan 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.
- 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 job *shop* dapat digambarkan seperti pada gambar berikut ini.



Gambar 2.1 Model Rute Job Shop

Didalam penyusunan jadwal job *shop* diperlukan adanya aturan pengurutan pekerjaan ke dalam mesin-mesin sebagai jadwal awal yang akan dilakukan perbaikan pengurutan pekerjaannya. Penelitian eksperimen telah menunjukkan bahwa penyusunan penjadwalan berdasarkan prioritas aturan pengurutan sebagai metode praktis untuk mencari solusi yang baik dalam permasalahan job *shop* meskipun tentunya kondisi optimal tidak dapat dijamin. Terdapat beberapa aturan prioritas umum yang dapat digunakan secara efektif di dalam proses penjadwalan dalam lingkungan job *shop*, yaitu:

- *Shortest Processing Time* (SPT) yaitu memilih waktu operasi dengan waktu proses yang paling minimum.

- *First Come First Served* (FCFS) yaitu memilih operasi yang datang pada mesin lebih awal.
- *Most Work Remaining* (MWKR) yaitu memilih operasi terkait dengan pekerjaan yang paling memiliki pekerjaan tersisa.
- *Least Work Remaining* (LWKR) yaitu memilih operasi terkait dengan pekerjaan yang memiliki pekerjaan paling sedikit tersisa untuk diproses.

Pada kondisi penjadwalan dengan kriteria meminimasi *total flow time*, SPT dan LWKR biasanya lebih efektif dari aturan yang lainnya. Sedangkan untuk kriteria meminimasi *makespan*, maka MWKR dan FCFS memberikan pendekatan yang lebih baik.

2.2.3 Mesin Paralel

Secara umum, penjadwalan membutuhkan keputusan terkait dengan pengurutan pekerjaan dan alokasi sumber daya. Ketika pada kondisi terdapat sejumlah m mesin identik yang dapat digunakan memproses n pekerjaan secara bersamaan, dan diasumsikan bahwa pekerjaan dapat diproses melalui paling tidak pada satu mesin pada suatu waktu tertentu.

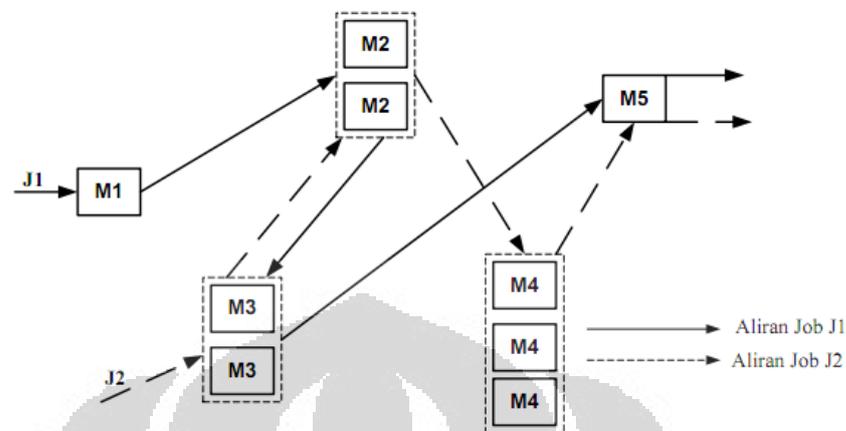
Pada kondisi lingkungan *job shop* mesin paralel berarti terdapat sejumlah m mesin identik untuk menyelesaikan n pekerjaan pada setiap rute proses yang dilalui oleh suatu pekerjaan. Kondisi *job shop* dengan paralel mesin dapat digambarkan seperti pada gambar dibawah ini.

Dalam permasalahan mesin paralel dengan kriteria *total completion time* aturan prioritas SPT memberikan pendekatan yang sangat baik. Berdasarkan aturan SPT, pekerjaan dengan waktu pemroses terkecil akan dialokasikan pada mesin 1 kesatu, pekerjaan terkecil berikutnya dialokasi pada mesin 2 kedua, dan seterusnya

2.3. Konsep dan Definisi *Due Date*

Terdapat banyak definisi mengenai *due-date* yang diberikan oleh para pakar *scheduling* diantaranya: didefinisikan sebagai titik waktu (*point of time*) saat pemrosesan pekerjaan harus dapat diselesaikan, Baker (1974). Bedworth dan Bailey (1987) memperluas definisi *due-date* ini sebagai batas akhir yang

ditetapkan terhadap suatu pekerjaan dan bila melebihi batas akhir itu dikatakan terjadi keterlambatan yang dikenakan *penalty* atau denda.



Gambar 2.2 Kondisi *Job Shop* Mesin Paralel

Halim et al.(1998) dalam kaitannya dengan konsumen mendefinisikan *due-date* sebagai waktu penyerahan produk yang dipesan sesuai dengan kesepakatan antara pihak produsen dan konsumen pada awal proses perencanaan. Lebih spesifik lagi, Halim et al.(1999) mendefinisikan *due-date* sebagai batas waktu dari operasi terakhir item *ke-i* harus sudah selesai dikerjakan.

Penelitian dalam area penentuan *due date* dalam masalah penjadwalan, mulai dirintis oleh Seidman et al.(1981) dan Panwalkar et al.(1982) dalam Shabtay (2010). Pengembangan penentuan *due date* pada awalnya dilakukan melalui pendekatan *due date* yang berlaku umum untuk semua pekerjaan (*common due date*) yang kenal dengan *CON method*.

Penyelesaian pekerjaan yang lebih awal dari *due-date* akan menyebabkan penumpukan persediaan, sedangkan penyelesaian melebihi *due-date* biasanya akan dikenakan *penalty* oleh pemesan. Dalam penelitian ini tidak diperkenankan terjadinya keterlambatan. Pernyataan yang menarik mengenai *due-date* ini adalah kapan saat terakhir suatu pekerjaan harus dimulai agar dapat selesai tepat pada *due-date*.

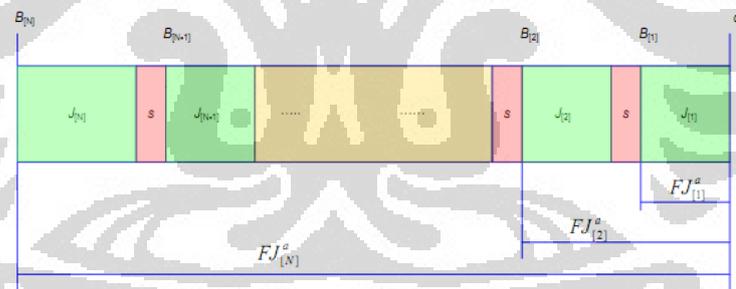
2.4. Total Actual flow time

Halim (1993) menyatakan bahwa konsep *actual flow time* sebagai ukuran performansi untuk *shop-time* pertama kali dikemukakan Miyazaki dan Ohta (1987), yaitu waktu yang diperlukan suatu pekerjaan tersebut sampai due-date, atau

$$FJ_{[i]}^a = d - B_{[i]} \text{ dengan } i = 1, 2, \dots, b \quad (2.1)$$

Dengan d adalah *common due-date* dan $B[i]$ adalah saat mulai pemrosesan pekerjaan $J[i]$. Selanjutnya, Halim (1993) menurunkan formulasi diatas menjadi sebagai berikut: jika diasumsikan waktu *set-up*, s , konstan dan tidak termasuk ke dalam waktu proses pekerjaan, p_i , maka berdasarkan Gambar 2.1 dapat dituliskan :

$$FJ_{[i]}^a = \left\{ \sum_{j=1}^i (p_j + s) \right\} - s ; \text{ dengan } i = 1, 2, \dots, b \quad (2.2)$$



Gambar 2.3 Waktu Tinggal Aktual (*Actual flow time*) Pekerjaan

Bila diketahui ukuran masing-masing *batch*, maka persamaan (2.2) dapat diformulasikan sebagai berikut. Misalkan terdapat N *batch* dari satu jenis item dengan kuantitas masing-masing *batch* $Q_{[i]}$, $i = 1, 2, \dots, N$, waktu proses satuan tiap *part* adalah t , dan waktu *set-up* antar *batch* adalah s . Jika diasumsikan bahwa *batch* terjadwal datang tepat pada saat proses akan dimulai ($B_{[i]}$) dan semua *part* (dari seluruh *batch*) yang selesai diserahkan sekaligus pada suatu *common due-date*, maka dengan pendekatan mundur *actual flow time* untuk *batch* dapat

ditentukan melalui persamaan (2.2) dengan mengganti waktu proses pekerjaan dengan waktu proses *batch* yang didapat dari perkalian antara ukuran *batch* dengan waktu proses sebuah *part*, atau

$$FL_{[i]}^a = \left\{ \sum_{j=1}^i (tQ_{[j]} + s) \right\} - s \quad (2.3)$$

Bila semua *part* dalam suatu *batch* datang pada saat yang bersamaan, maka setiap *part* tersebut akan tinggal di *shop* sepanjang $FL_{[i]}^a$. Dengan kata lain *actual flow time* masing-masing *part* sama dengan waktu tinggal aktual *batch*. Jadi, *actual flow time* $FL_{[i]}^a$ untuk seluruh *part* dalam *batch* $L_{[i]}$ dapat dihitung dengan mengalikan waktu tinggal aktual *batch* dengan jumlah *part* dalam *batch* tersebut, sehingga :

$$F_{[i]}^a = (\{\sum_{j=1}^i (tQ_{[j]} + s)\} - s_i) Q_{[i]} \text{ dengan } i = 1, 2, \dots, N \quad (2.4)$$

Dengan demikian, maka total *actual flow time* untuk seluruh *part* di dalam *shop*:

$$F^a = \sum_{i=1}^N (\{\sum_{j=1}^i (tQ_{[j]} + s)\} - s_i) Q_{[i]} \text{ dengan } i = 1, 2, \dots, N \quad (2.4)$$

2.5. Algoritma *Tabu search*

Tabu search berasal dari Tongan, suatu bahasa Polinesia yang digunakan oleh suku Aborigin Pulau tonga untuk mengindikasikan suatu hal yang tidak boleh “disentuh” karena sakralnya. Menurut kasus Webster, *Tabu* berarti larangan yang dipaksakan oleh kebudayaan sosial sebagai suatu tindakan pencegahan atau sesuatu yang dilarang karena berbahaya. Bahaya yang harus dihindari dalam *Tabu search* adalah penjadwalan yang tidak layak, dan terjebak tanpa ada jalan keluar. Dalam konteks lebih luas, larangan perlindungan dapat diganti jika terjadi tuntutan yang mendadak.

Tabu search adalah sebuah metode optimasi yang berbasis pada *lokal search*. Proses pencarian bergerak dari satu solusi ke solusi berikutnya, dengan cara memilih solusi terbaik *neighbourhood* solusi sekarang (*current*) yang tidak

tergolong solusi terlarang (tabu). 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.

Struktur memori fundamental dalam *tabu search* dinamakan *tabu list*. *Tabu list* menyimpan atribut dari sebagian *move* (transisi solusi) yang telah diterapkan pada iterasi-iterasi sebelumnya. *Tabu search* menggunakan *tabu list* untuk menolak solusi-solusi yang memenuhi atribut tertentu guna mencegah proses pencarian mengalami *cycling* pada daerah solusi yang sama, dan menuntun proses pencarian menelusuri daerah solusi yang belum dikunjungi. Tanpa menggunakan strategi ini, *local search* yang sudah menemukan solusi optimum lokal dapat terjebak pada daerah solusi optimum lokal tersebut pada iterasi-iterasi berikutnya. List ini mengikuti aturan LIFO dan biasanya sangat pendek (panjangnya biasanya sebesar $O(\sqrt{N})$ dimana N adalah jumlah total dari operasi). Setiap saat ada langkah itu akan ditempatkan dalam *tabu list*.

Perekaman solusi secara lengkap dalam sebuah *forbidden list* dan pengecekan apakah sebuah kandidat solusi tercatat dalam *list* tersebut merupakan cara yang mahal, baik dari sisi kebutuhan memori maupun kebutuhan waktu komputasi. Jadi, *tabu list* hanya menyimpan langkah transisi (*move*) yang merupakan lawan atau kebalikan dari langkah yang telah digunakan dalam iterasi sebelumnya untuk bergerak dari satu solusi ke solusi berikutnya. Dengan kata lain *tabu list* berisi langkah-langkah yang membalikan solusi yang baru ke solusi yang lama.

Pada tiap iterasi, dipilih solusi baru yang merupakan solusi terbaik dalam *neighbourhood* dan tidak tergolong sebagai tabu. Kualitas solusi baru ini tidak harus lebih baik dari kualitas solusi sekarang. Apabila solusi baru ini memiliki nilai fungsi objektif lebih baik dibandingkan solusi terbaik yang telah dicapai sebelumnya, maka solusi baru ini dicatat sebagai solusi terbaik yang baru.

Sebagai tambahan dari *tabu list*, dikenal adanya kriteria aspirasi, yaitu suatu penanganan khusus terhadap *move* yang dinilai dapat menghasilkan solusi yang dinilai dapat menghasilkan solusi yang baik namun *move* tersebut berstatus tabu. Dalam hal ini, jika *move* tersebut memenuhi kriteria aspirasi yang telah ditetapkan sebelumnya, maka *move* tersebut dapat digunakan untuk membentuk solusi berikutnya (status tabunya dibatalkan). Berikut ini diberikan kerangka umum algoritma *tabu search* dalam notasi bahasa pascal.

Sebuah informasi akan digunakan sebagai petunjuk untuk bergerak dari i ke solusi selanjutnya dalam $N_{(i)}$. Penggunaan memori sebagai pembatas dalam pemilihan beberapa subset dari $N_{(i)}$ dengan mencegah pergerakan ke beberapa solusi tetangga. Sebuah informasi akan digunakan sebagai petunjuk untuk bergerak dari i ke solusi selanjutnya dalam $N_{(i)}$. Penggunaan memori sebagai pembatas dalam pemilihan beberapa subset dari $N_{(i)}$ dengan mencegah pergerakan ke beberapa solusi tetangga.

Secara formal, kita dapat menganggap masalah optimalisasi dalam cara berikut: Diberikan sebuah himpunan solusi S dan sebuah fungsi $f : S$, temukan solusi i^* dalam S sehingga $f_{(i^*)}$ dapat diterima dengan beberapa kriteria. Secara umum kriteria untuk dapat diterima sebagai solusi i^* harus memenuhi $f(i^*)=f(i)$ untuk setiap i dalam S . Dalam situasi metode pencarian tabu akan menjadi sebuah algoritma minimisasi secara pasti yang menyediakan proses eksplorasi yang menjamin setelah sejumlah langkah-langkah terhingga i^* dapat dicapai.

Untuk mendalami lagi prinsip kerja metode pencarian tabu, kita dapat memformulakan metoda menurun klasik (*classical descent method*) dalam beberapa langkah, yaitu:

1. Memilih sebuah solusi awal i dalam S
2. Membangkitkan subset V^* sebagai solusi dalam $N_{(i)}$
3. Mencari j 'terbaik' dalam V^* dan menetapkan $i=j$
4. Jika $f(j)=f(i)$ maka berhenti, namun jika tidak kembali ke langkah ke-2

```

begin
  {Buat solusi awal s yang feasibel dengan
  menggunakan
  suatu metode heuristik tertentu atau secara
  acak}
  best := cost(s);
  s* := s; {s* adalah solusi terbaik yang
  diperoleh}
  tabu_list := null;
  repeat
  Candidate(s) := (s' ∈ N(s): merupakan move
  dari s
  ke s' yang tidak tergolong elemen dari tabulist
  atau
  memenuhi kriteria aspirasi);
  {pilih s ∈ Candidate(s): s adalah solusi yang
  memiliki nilai cost minimum };
  {simpan move yang berlawanan ke dalam tabulist,
  yaitu
  yang mengubah s ke s};
  s := s;
  if (cost(s) < best) then
  s* := s;
  best := cost(s);
  until (stopping-criteria = TRUE);
  return(s*);
end;

```

Dalam metode menurun secara umum akan dapat langsung ditetapkan bahwa $V^* = N_{(i)}$. Tetapi hal ini seringkali membutuhkan waktu yang lama. Untuk itulah cara pemilihan V^* yang tepat seringkali dijadikan sebagai peningkatan yang penting dalam metode pencarian. Dalam metode menurun secara umum akan dapat langsung ditetapkan bahwa $V^* = N_{(i)}$. Tetapi hal ini seringkali membutuhkan waktu yang lama. Untuk itulah cara pemilihan V^* yang tepat seringkali dijadikan sebagai peningkatan yang penting dalam metode pencarian. Pada kasus yang berlawanan, akan ditetapkan $V^* \neq I$. Hal ini akan menurunkan fase pemilihan j 'terbaik'. Solusi j akan diterima jika $f(j) = f(i)$, sebaliknya hal ini akan diterima dengan kemungkinan tertentu yang bergantung pada nilai-nilai f pada i dan j serta pada sebuah parameter yang disebut temperatur. Tidak ada temperatur dalam metode pencarian tabu, namun pemilihan V^* akan menjadi hal yang penting guna mendefinisikannya dalam setiap langkah di mana akan terjadi penggunaan memori secara sistematis untuk memanfaatkan informasi yang ada di luar fungsi f dan lingkungan $N_{(i)}$.

Sebagai pengecualian pada kasus-kasus istimewa, penggunaan prosedur menurun (*descent procedure*) secara umum lebih rumit karena kita akan terperangkap pada sebuah minimum lokal yang mungkin masih jauh dari minimum global. Maka untuk proses iterasi dalam eksplorasi apapun sebaiknya dalam beberapa hal juga menerima langkah-langkah yang tidak akan memberikan

perkembangan dari i ke j dalam V^* (misal $f(j) > f(i)$). Metode pencarian tabu secara berbeda memilih j 'terbaik' dalam V^* .

Selama pergerakan yang tidak memberi perkembangan itu mungkin, resiko pengunjungan kembali sebuah solusi atau lebih umumnya disebut sebagai siklus mungkin untuk terjadi. Dalam hal inilah penggunaan memori sangat diperlukan untuk mencegah terjadinya pergerakan ke solusi yang telah dikunjungi. Jika memori seperti itu diperkenalkan, maka kita dapat menganggap struktur $N_{(i)}$ tergantung pada pengelilingan yang merupakan pengulangan k . Jadi kita dapat merujuk ke $N_{(i,k)}$ daripada $N_{(i)}$.

Dengan perujukan ini kita dapat mencoba untuk melakukan peningkatan algoritma menurun dalam beberapa hal untuk lebih mendekati metode ini ke prosedur dalam metode pencarian tabu secara umum. Hal ini dapat didefinisikan dalam beberapa langkah (catatan i^* adalah solusi 'terbaik' yang ditemukan dan k adalah penghitung dalam pengulangan):

1. Memilih solusi awal I dalam S . Tetapkan $i^*=I$ dan $k=0$.
2. Tetapkan nilai $k=k+1$ dan membagkitkan subset V^* sebagai solusi dalam $N_{(i,k)}$
3. Pilih j 'terbaik' dalam V^* dan tetapkan $i=j$.
4. Jika $f(i) < f(i^*)$ maka tetapkan $i^*=i$.
5. Jika kondisi berhenti ditemukan, maka proses dihentikan, sedangkan jika belum kembali ke langkah 2.

Amati bahwa langkah-langkah pada metode penurunan klasik termasuk dalam formula ini (kondisi berhenti secara sederhana jika $f(i)=f(i^*)$ dan i^* akan selalu menjadi solusi akhir). Selain itu kita juga dapat mempertimbangkan penggunaan f yang telah dimodifikasi daripada f yang dalam beberapa keadaan di deskripsikan kemudian.

Dalam metode pencarian tabu, kondisi berhenti dapat berupa:

1. $N_{(i,k+1)}$ = tidak terdefinisi.
2. k lebih besar daripada bilangan maksimum pada pengulangan.

3. Banyaknya pengulangan selama peningkatan terakhir i^* lebih besar dari bilangan tertentu.
4. Pembuktian dapat diberikan daripada solusi optimum yang telah didapatkan.

Selama aturan-aturan berhenti ini memungkinkan untuk memiliki beberapa pengaruh dalam prosedur pencarian dan pada hasil-hasilnya, penting untuk menyadari bahwa pendefinisian $N_{(i,k)}$ dalam tiap pengulangan k dan pemilihan V^* adalah hal yang krusial. Definisi dari $N_{(i,k)}$ menyatakan secara tidak langsung bahwa beberapa solusi yang telah dikunjungi dihapus dari $N_{(i)}$, mereka dianggap sebagai solusi-solusi tabu yang harus dihindari dalam pengulangan selanjutnya. Sebagai contoh, pemeliharaan pada pengulangan k sebuah list T (*list tabu*) pada solusi yang telah dikunjungi terakhir $|T|$ akan mencegah terjadinya siklus pada ukuran paling banyak sebesar $|T|$. Pada kasus ini, kita bisa mengambil $N_{(i,k)} = N_{(i)} - T$. Namun list T kemungkinan tidak dapat digunakan secara praktis. Oleh karena itu, kita akan mendeskripsikan proses eksplorasi pada S dalam masa pergerakan dari satu solusi ke solusi lainnya. Untuk setiap solusi I dalam S , kita dapat mendefinisikan $M_{(i)}$ sebagai himpunan gerak yang dapat digunakan oleh i untuk memperoleh solusi baru j .

Jadi daripada mempertahankan list T dari solusi-solusi yang telah dikunjungi, kita dapat secara sederhana memelihara jalur gerak terakhir $|T|$ atau gerak balik terakhir $|T|$ yang diasosiasikan dengan gerakan-gerakan yang sebenarnya telah dilakukan. Maka dengan jelas bahwa pembatasan yang ada adalah kehilangan informasi, dan hal itu tidak menjamin tidak terjadinya siklus dengan panjang paling banyak $|T|$. Untuk kepentingan efisiensi, maka diperlukan penggunaan beberapa list T_j dalam satu waktu maka beberapa unsur pokok t_j (dari i atau dari m) akan diberikan tabu status untuk mengindikasikan bahwa unsur pokok ini sedang tidak diperbolehkan untuk terlibat dalam pergerakan. Secara umum pergerakan untuk tabu status adalah fungsi tabu status pada unsur-unsur pokoknya yang dapat diubah pada setiap pengulangan. Gerak tabu m digunakan pada solusi i yang mungkin tampak menarik karena itu diberikan sebagai contoh

sebuah solusi yang lebih baik dari pada yang telah ditemukan. Kita akan dapat menerima m tanpa memperhatikan statusnya. Kita akan melakukan hal tersebut jika m memiliki tingkat aspirasi (*aspiration level*) yang lebih baik daripada permulaan.

Sekarang kita dapat mendefinisikan karakteristik dari prosedur pencarian tabu dalam langkah-langkah berikut, antara lain:

1. Memilih solusi awal i dalam S . Tetapkan $i^*=I$ dan $k=0$.
2. Tetapkan $k=k+1$ dan bangkitkan sebuah subset V^* dari solusi dalam $N_{(i,k)}$ sehingga salah satu dari kondisi tabu t_γ yang melanggar ($\gamma=1,2,\dots,t$) atau setidaknya satu kondisi aspirasi a_γ yang memiliki ($\gamma=1,2,\dots,a$).
3. Pilih j terbaik melalui $j=i \nabla m$ dalam V^* dan tetapkan $i=j$.
4. Jika $f(i) < f(i^*)$ maka tetapkan $i^*=i$.
5. Perbaharui kondisi Tabu dan aspirasi.
6. Jika kondisi berhenti ditemukan, maka proses dihentikan. Jika tidak, kembali ke langkah 2.

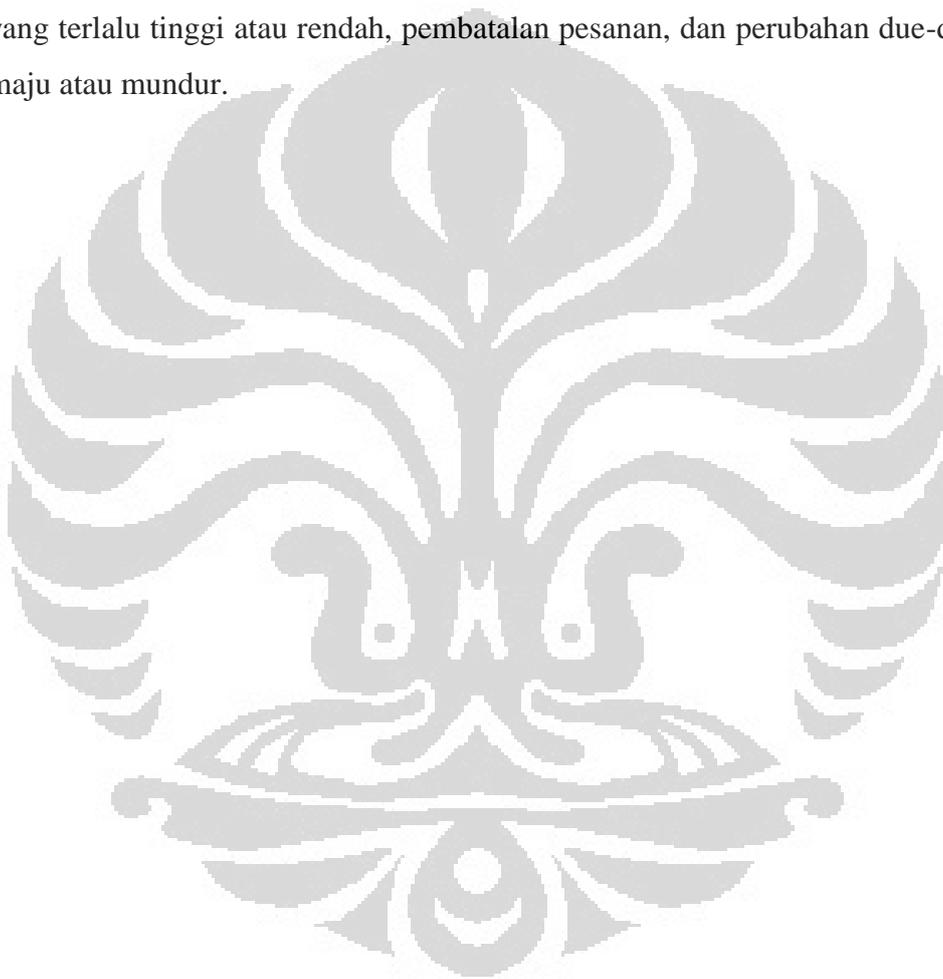
2.6. Penjadwalan Ulang (*Rescheduling*)

Penjadwalan ulang (*rescheduling*) merupakan suatu proses interatif atau berulang dari tiga langkah berikut Wu dan Li (1995):

- 1) Langkah evaluasi, yaitu mengevaluasi dampak atau pengaruh terjadinya faktor-faktor *rescheduling*. Jika dampak bisa diterima, misalnya faktor-faktor *rescheduling* tidak mempengaruhi jadwal yang ada, maka tidak dibutuhkan tindak lanjut apapun.
- 2) Langkah solusi, yaitu menentukan solusi *rescheduling* yang dapat meningkatkan performansi jadwal yang ada. Akan tetapi, cara menentukan solusi *rescheduling* masih terbuka sebagai masalah penelitian dan konsekuensinya adalah langkah ini merupakan bagian yang paling sulit dalam *rescheduling*.
- 3) Langkah perbaikan atau revisi, yaitu mirip dengan langkah evaluasi kecuali inputnya yang berbeda yaitu inputnya adalah solusi *rescheduling*.

Jadwal yang ada direvisi dan diperbaharui. Jika hasil jadwal yang direvisi dapat diterima, maka selesai. Jika tidak, maka ulangi langkah 2 dan 3 sampai diperoleh jadwal revisi yang dapat diterima.

Faktor-faktor yang menimbulkan gangguan pada jadwal semula disebut dengan faktor-faktor penjadwalan ulang (*rescheduling factors*), diantaranya adalah Li et al.(1993) kerusakan mesin, kedatangan pesanan baru yang harus segera dikerjakan, kekurangan material, masalah kualitas, perkiraan waktu proses yang terlalu tinggi atau rendah, pembatalan pesanan, dan perubahan due-date bisa maju atau mundur.



BAB 3

PENGEMBANGAN MODEL

3.1 Skenario Pengembangan Model

Penelitian ini mengembangkan dua buah model algoritma yang terdiri dari model satu pemenuhan *due date* dan model dua penentuan *due date*, masing-masing model dikembangkan untuk kondisi dinamis. Pengembangan model yang dilakukan terlebih dahulu diawali dengan pengembangan model pada kondisi statis, pada kondisi ini diasumsikan semua item (job) yang harus dikerjakan tersedia pada awal periode perencanaan ($t=0$). Setelah permasalahan pada kondisi statis dapat terselesaikan, maka dilakukan pengembangan pada kondisi dinamis, didalam permasalahan terjadi kedatangan item baru yang harus dikerjakan disepanjang horizon perencanaan.

Pada kondisi dinamis, pada prinsipnya akan melakukan evaluasi dan melakukan pencarian solusi atas kedatangan item-item baru yang terjadi disepanjang horizon perencanaan. Dari solusi kondisi dinamis ini, akan menunjukkan item baru mana saja yang realistis untuk dikerjakan, dengan memperhatikan item-item yang sudah terjadwal sebelumnya. Kedua model yang dikembangkan bertujuan untuk meminimasi *total actual flow time* dengan menggunakan *Tabu Search*, didalam proses pencarian solusi.

3.2 Notasi dan Definisi

Dalam pemodelan permasalahan penjadwalan pada sistem produksi job shop mesin paralel yang memproduksi multi item berstruktur multi level, didalam pemenuhan dan penentuan *due date*. Kondisi permasalahan yang dibahas seperti dijelaskan didalam Sub Bab 3.1 menggunakan notasi-notasi sebagai berikut.

a. Subscript

i = menyatakan jenis item i yang akan diproduksi, sehingga $i=1,2,\dots,r$, dimana r menyatakan banyaknya item yang akan diproduksi.

- j = menyatakan komponen ke- j dari c_{i0} jenis komponen untuk membuat item i , sehingga $j=1,2,\dots,c$.
- k = menyatakan urutan ke- k dari h_{ij} proses yang harus dilalui didalam pembuatan komponen ke- j dari item i , sehingga $k=1,2,\dots,h_{ij}$.
- l = menyatakan level ke- l dari e_{i0} dari item i , sehingga $l=0,1,2,\dots,e_{i0}$.
- m = menyatakan mesin ke- m dari v mesin yang tersedia, sehingga $m=1,2,3,\dots,v$.
- n = menyatakan mesin identik ke- n dari jumlah mesin w pada setiap tahapan proses atau kelompok mesin, sehingga $n=1,2,3,\dots,w$.

b. Parameter

- c_{i0} = menyatakan jumlah jenis komponen yang diperlukan untuk membuat item i , dimana c_{i0} tidak menyatakan jumlah level.
- h_{ij} = menyatakan jumlah proses yang harus dilalui dalam pembuatan komponen ke- j dari item ke- i .
- e_{i0} = menyatakan jumlah level yang dimiliki oleh item i didalam struktur produknya.
- v = menyatakan jumlah mesin yang tersedia didalam sistem produksi.
- w = menyatakan jumlah jenis mesin identik yang tersedia untuk setiap mesin ke- m .
- p_{i0} = jumlah item i yang harus diproduksi dimana $i=1,2,\dots,r$.
- p_{ij} = menyatakan jumlah komponen ke- j dari item i .
- n_{ij} = menyatakan jumlah komponen ke- j dari item i .
- $Z(p_{ij})$ = set dari induk-induk komponen j dari item i berdasarkan struktur produknya.
- H_{ij} = banyaknya komponen j yang dibutuhkan untuk membuat satu unit induk langsung didalam struktur produknya (*bill of*

material). Dapat juga diartikan sebagai jumlah komponen j yang diperlukan untuk membuat komponen pada level di atasnya.

d_i = *due date* item i .

O_{i0k0mn} = operasi ke- k untuk item i pada level ke-0 yaitu p_{i0} yang dilakukan di mesin m untuk semua mesin n dimana $m=1,2,\dots,v$ dan $n=1,2,\dots,w$.

O_{ijklmn} = operasi ke- k untuk komponen p_{ij} yang dikerjakan pada level ke- l yang dilakukan di mesin m ke n dimana $m=1,2,\dots,v$ dan $n=1,2,\dots,w$.

t_{i0klmn} = waktu operasi O_{i0k0mn} untuk setiap p_{i0} .

t_{ijklmn} = waktu operasi O_{ijklmn} untuk setiap p_{ij} .

S_{ijklmn} = waktu set-up semua mesin m identik untuk operasi O_{ijklmn} .

dr_{ijklmn} = tingkat cacat (*defect rate*) operasi ke- k level ke- l dari komponen p_{ij} pada semua mesin m .

c. Variabel

TF = *total actual flow time*.

Q_{i0} = variabel yang menyatakan unit item p_{i0} yang dibuat.

Q_{ij} = variabel yang menyatakan unit komponen p_{ij} yang akan dibuat.

S_{i0klmn} = variabel yang menyatakan saat dimulainya operasi O_{i0k0mn} .

S_{ijklmn} = variabel yang menyatakan saat dimulainya operasi O_{ijklmn} .

C_{i0klmn} = variabel yang menyatakan saat selesainya operasi O_{i0k0mn} .

C_{ijklmn} = variabel yang menyatakan saat selesainya operasi O_{ijklmn} .

3.3 Model Penjadwalan Pemenuhan *Due date*

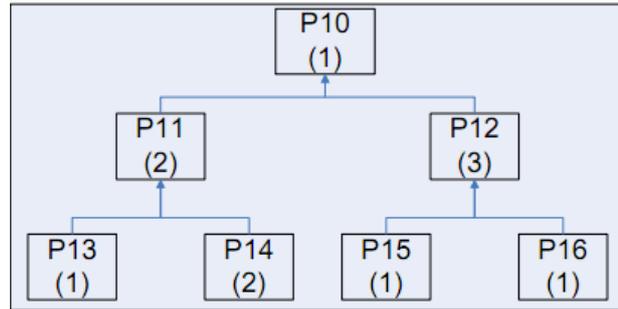
Sistem yang dibahas pada model penjadwalan ini adalah sistem yang memproduksi beberapa jenis item dan berstruktur multi level. Dalam sistem ini setiap item memiliki struktur produknya (*Bill of Material-BoM*) masing-masing, dimana produk akhir dan setiap komponen pada tiap levelnya dimungkinkan

untuk diproses di beberapa mesin berbeda yang juga dengan *routing* (urutan) yang berbeda-beda.

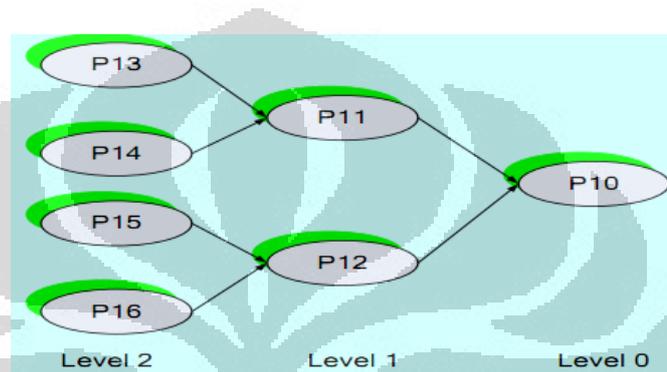
Sistem produksi yang ditinjau berlingkungan *Just in Time* (JIT), dengan memperhatikan tingkat cacat pada setiap operasi yang dilalui selama proses produksi. Kondisi ini selaras dengan lingkungan JIT, dalam JIT kualitas dari produk yang dihasilkan harus sesuai dengan spesifikasi konsumen atau dengan kata lain kualitas harus tetap dijaga. Masalah tersebut dapat diformulasikan sebagai berikut.

Terdapat r jenis item yang dinyatakan dalam p_{i0} , dengan indeks $i=1,2,\dots,r$. Dimana setiap p_{i0} akan diproduksi sebanyak n_{i0} yang harus diselesaikan pada *due date*, d_{i0} . Tiap item yang diproduksi memiliki level sebanyak e_{i0} level pada struktur produknya, sehingga $l=0,1,2,\dots,e_{i0}$ dan diperlukan sebanyak c_{i0} jenis komponen, dengan setiap jenis komponen dinyatakan dalam p_{ij} , sehingga $j=0,1,2,3,\dots,c_{i0}$. Dengan jumlah item p_{ij} yang akan diproduksi sebanyak n_{ij} . Setiap item dan komponennya diproses dalam h_{ij} operasi, dengan urutan operasi dinyatakan dalam k sehingga $k=1,2,3,\dots,h_{ij}$, pada suatu sistem produksi job shop mempunyai v jenis mesin dengan indeks $m=1,2,3,\dots,v$ dan w mesin identik sehingga mesin identik ke-1, ke-2 dan seterusnya dinyatakan dalam n , sehingga $n=1,2,3,\dots,w$.

Setiap mesin memiliki waktu set-up s_{ijklmn} dan waktu proses t_{ijklmn} , dimana waktu operasi sama untuk semua n mesin identik. Dengan masing-masing proses operasi memiliki tingkat cacat untuk setiap item dan komponen dinyatakan dalam dr_{ijklm} . Dalam hal ini, besarnya tingkat cacat sama untuk semua mesin paralel yang identik pada setiap tahapan pemrosesan. Kedatangan pekerjaan bersifat dinamis yang artinya tidak harus semua pekerjaan tersedia pada awal horizon perencanaan, tetapi terjadi kedatangan pekerjaan-pekerjaan disepanjang horizon perencanaan. Kondisi multi level beserta *directed graph* (*digraph*) dapat digambarkan seperti pada gambar 3.1, berikut ini.



Gambar 3.1 Item Berstruktur Multi-Level



Gambar 3.2 Diagraph dari Produk Akhir dan Komponennya

Gambar 3.1 dan 3.2 menjelaskan bahwa untuk membuat item p_{10} diperlukan sebanyak enam buah komponen yang terdiri yang terdiri dari tiga level, dimulai dari level nol sampai dengan level dua ($e_{10}=3$). Didalam membuat satu item p_{10} diperlukan komponen p_{11} sebanyak dua unit ($H_{11}=2$ unit) dan komponen p_{12} sebanyak tiga unit ($H_{12}=3$ unit). Sehingga item p_{10} baru dapat diproses setelah komonen p_{11} dan p_{12} selesai diproses, karena p_{10} adalah induk langsung dari komponen p_{11} dan p_{12} .

Model yang dikembangkan ini harus dapat menjawab permasalahan sebagai berikut;

- Bagaimana menentukan penjadwalan untuk pemenuhan *due date* dalam lingkungan JIT untuk setiap item dan komponen-komponennya berdasarkan tingkat cacat untuk setiap operasi dengan memperhatikan ketersediaan mesin.
- Bagaimana melakukan penjadwalan ulang bila terjadi kedatangan pesananan-pesanan baru.

Permasalahan penjadwalan tersebut diformulasikan ke dalam fungsi tujuan dan kendala yang harus dipenuhi dalam proses pencarian solusi. Maka diperoleh fungsi tujuan,

$$\text{Minimasi } TF = \sum_{i=1}^r (d_i - B_i) \quad (3.1)$$

Persamaan (3.1) menjelaskan fungsi tujuan minimasi total waktu tinggal aktual untuk semua item, dengan B_i menyatakan saat mulai untuk operasi pertama dari komponen yang berada pada level e struktur produk item i yang terjadwal paling akhir.

Adapun fungsi-fungsi kendala dari permasalahan ini terdiri dari beberapa kendala yang berhubungan dengan ketersediaan mesin dan hubungan ketergantungan pekerjaan pada setiap levelnya.

- Fungsi kendala hubungan kuantitas item dengan kuantitas komponen.

$$n_{ij} = n_{i0} * H_{ij} \left[\prod_{P_{iq} \in Z(P_{ij})} H_{iq} \right], \quad H_{iq} \geq 1; \forall i, j \quad (3.2)$$

Persamaan (3.2) menyatakan hubungan kuantitas item yang dibuat dengan kuantitas komponen-komponen penyusunnya. H_{ij} menyatakan jumlah komponen ke- j yang diperlukan untuk membuat item i , untuk membuat satu unit induk langsung. Dengan P_{iq} menyatakan komponen ke- q sebagai elemen himpunan $Z(P_{ij})$ dari induk-induk komponen ke- j hingga item i . Persamaan ini untuk menjamin bahwa didalam penentuan jumlah komponen ke- j harus memperhatikan struktur produk penyusunannya.

- Fungsi kendala hubungan tingkat cacat (defect rate) terhadap jumlah komponen yang diproduksi.

$$n_{ijk(\text{diproduksi})} \geq n_{ij(k+1)(\text{diproduksi})} * (1 + dr_{ijkmn}) \quad (3.3)$$

Dimana, $n_{ijk(\text{diproduksi})} \geq n_{ij(k+1)(\text{diproduksi})} * (1 + dr_{ijkmn})$, sehingga persamaan (3.3) menyatakan bahwa jumlah komponen ke- j dari item i pada

operasi ke- k yang harus diproduksi, pada kondisi suatu item harus diproses lebih dari satu kali pada mesin yang berbeda, ditentukan oleh jumlah komponen pada operasi ke- $k+1$ dan *defect rate* (dr) untuk semua mesin m yang identik yang tersedia dalam tahapan produksi pada operasi ke- $k+1$, dengan pembulatan dilakukan ke atas.

- Fungsi kendala kelayakan waktu memulai penjadwalan

$$B_{ij1lmn}[N] \geq 0; \forall i, j, l, m, n \quad (3.4)$$

$$B_{ij1lmn}[N] \geq B_i; \forall i, j, l, m, n \quad (3.5)$$

Persamaan (3.4) menjamin bahwa waktu dimulai pemrosesan dari komponen p_{ij} harus lebih besar atau sama dengan nol, yang berarti pekerjaan tersebut layak dikerjakan didalam horizon waktu perencanaan. Persamaan (3.5) waktu dimulai pemrosesan pertama dari komponen p_{ij} harus lebih besar atau sama dengan waktu dimulai pemrosesan item- i .

- Fungsi kendala hubungan waktu penyelesaian komponen terhadap induk langsungnya.

$$C_{ijhl+1mn} - B_{ij1lmn} \leq 0; \forall i, j, k, l, m, n \quad (3.6)$$

Persamaan (3.6) menjamin bahwa pemroses pertama operasi pertama $k=1$ untuk komponen p_{ij} pada level ke- l sebagai induk langsung dari komponen p_{ij} pada level ke- $l+1$, dapat dimulai apabila operasi terakhir $k=h$ komponen p_{ij} telah selesai dikerjakan, disemua mesin.

- Fungsi kendala ketepatan pemenuhan *due date*.

$$B_{i0hemn}[1] = d_{i0} - (Q_{i0h0}[1] * t_{i0hemn}); \forall i, m, n \quad (3.7)$$

Persamaan (3.7) menyatakan bahwa operasi terakhir $k=h$ dari tiap item i , pada level terakhir $l=0$ harus selesai tepat pada saat *due date*-nya.

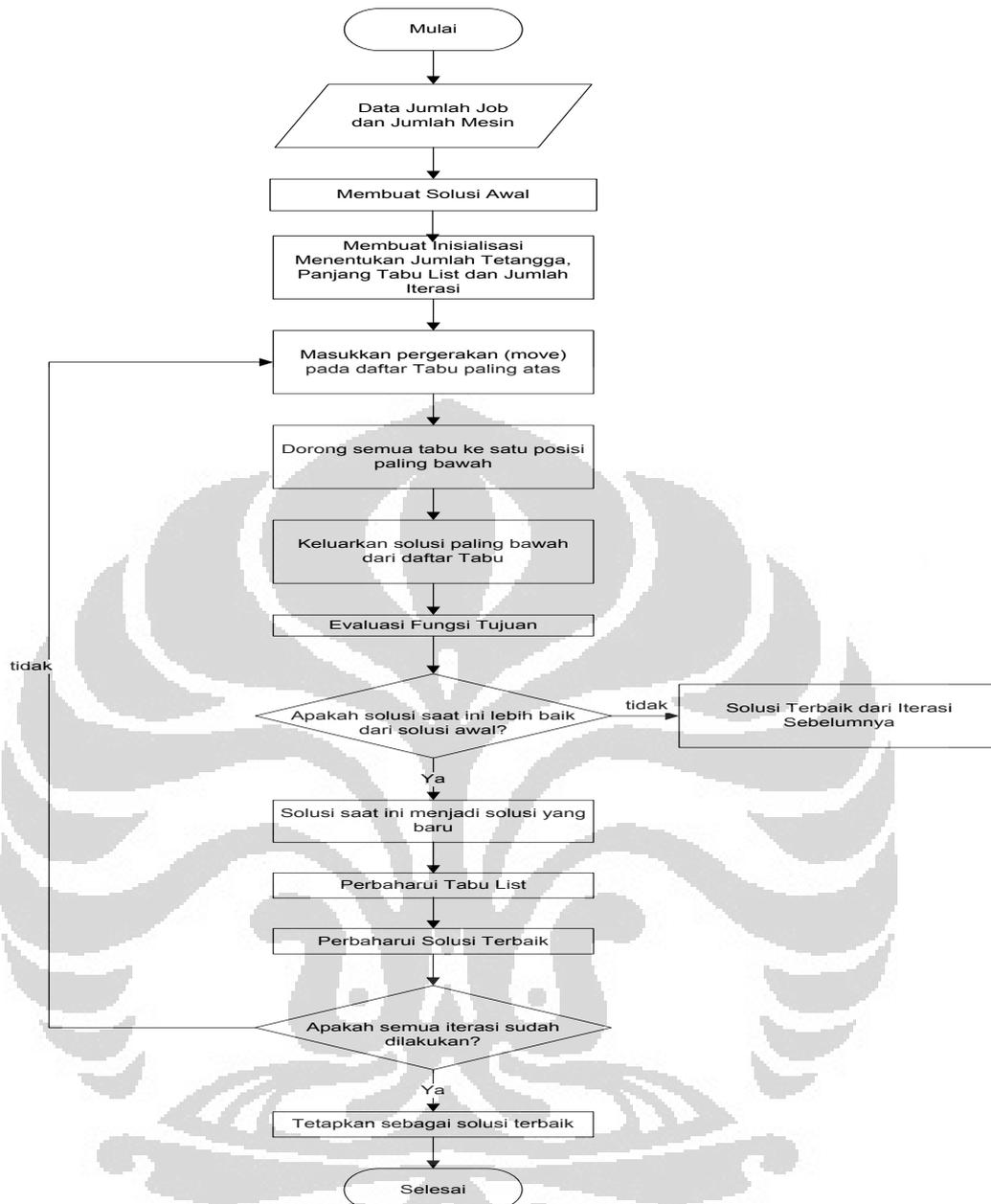
Formulasi dari permasalahan tersebut merupakan model yang kompleks karena banyak variabel terkait. Oleh karena itu, diusulkan menggunakan pendekatan

heuristik untuk mendapatkan solusi, dengan mengubah variabel-menjadi parameter. Dengan demikian, sebelum menyelesaikan formulasi matematis di atas, diperlukan sejumlah iterasi yang jumlahnya tergantung jumlah, jenis, dan level pada struktur produk item serta jumlah stasiun kerja yang dilalui (*multi-stage*).

Algoritma yang diusulkan untuk menyelesaikan masalah penjadwalan yaitu Algoritma *Tabu Search*. Salah satu metode meta-heuristik, yang memiliki struktur memori yang disebut dengan *Tabu List* untuk menyimpan atribut dari sebagian pergerakan (*move*) yaitu langkah transisi dari solusi ke solusi yang lain yang digunakan untuk menolak solusi-solusi yang memenuhi atribut tertentu agar proses pencarian tidak berulang-ulang pada daerah solusi yang sama. Secara lebih jelasnya bisa dilihat di Gambar 3.3 Alur dari Algoritma *Tabu Search*.

Untuk permasalahan pemenuhan *due date* kondisi statis, langkah-langkah *Tabu Search* sebagai berikut:

- Mulai dengan solusi awal. Pembangkitan solusi awal untuk permasalahan pemenuhan *due date* kondisi statis ditetapkan dengan cara mengurutkan *due date* dari yang terkecil, agar penyelesaian tidak mengalami keterlambatan.
- Menetapkan solusi tetangga, panjang *tabu list* dan jumlah iterasi. Dalam penelitian ini, digunakan jumlah solusi tetangga = 5 ; panjang tabu list = 20 ; dan jumlah iterasi sebanyak 100 kali iterasi.
- Evaluasi fungsi tujuan dari solusi awal.
- Membentuk daftar kandidat pergerakan solusi yang mungkin.
- Jika solusi tidak tabu maka masukan solusi tersebut ke daftar solusi baru.
- Ulangi langkah-langkah diatas hingga iterasi maksimal yang sudah ditentukan diawal.
- Solusi-solusi terbaik yang dapat dimasukkan ke dalam daftar tabu, untuk mencegah pengulangan dalam pencarian solusi yang sama pada tiap iterasi.
- Jika semua iterasi telah dilakukan maka dapat diperoleh solusi terbaiknya.



Gambar 3.3 Algoritma *Tabu Search*

Untuk permasalahan pemenuhan *due date* kondisi dinamis secara garis besar bekerja dengan tahapan sebagai berikut:

- Mulai dengan solusi awal. Pembangkitan solusi awal untuk permasalahan pemenuhan *due date* kondisi dinamis ditetapkan dengan cara mengurutkan *due date* dari yang terkecil, tetapi tidak mengganggu penjadwalan sebelumnya

agar penyelesaian tidak mengalami keterlambatan. Karena pada kondisi dinamis yang artinya kedatangan pekerjaan baru dalam horizon perencanaan.

- Menetapkan solusi tetangga, panjang *tabu list* dan jumlah iterasi. Dalam penelitian ini, digunakan jumlah solusi tetangga = 5 ; panjang tabu list = 20 ; dan jumlah iterasi sebanyak 100 kali iterasi.
- Evaluasi fungsi tujuan dari solusi awal.
- Membentuk daftar kandidat pergerakan solusi yang mungkin.
- Jika solusi tidak tabu maka masukan solusi tersebut ke daftar solusi baru.
- Ulangi langkah-langkah diatas hingga iterasi maksimal yang sudah ditentukan diawal.
- Solusi-solusi terbaik yang dapat dimasukkan ke dalam daftar tabu, untuk mencegah pengulangan dalam pencarian solusi yang sama pada tiap iterasi.
- Jika semua iterasi telah dilakukan maka dapat diperoleh *due date* untuk kondisi dinamis.

3.4 Model Penjadwalan Penentuan *Due date*

Kondisi permasalahan pada model ini, pada dasarnya berlawanan dengan kondisi permasalahan pemenuhan *due date* (model satu), di mana dalam model ini, *due date* pada masing-masing item tidak diketahui. Karena *due date* diasumsikan dapat dinegosiasikan dengan konsumen. Sehingga fungsi kendala pada persamaan (3.7) pada model ini, dapat diabaikan. Fungsi tujuan untuk kondisi penentuan diperoleh dengan memodifikasi persamaan (3.1), sehingga fungsi tujuan untuk penentuan *due date* sebagai berikut:

$$\text{Minimasi } TF = \sum_{i=1}^r (C_i - B_i) \quad (3.13)$$

Persamaan (3.13) menyatakan minimasi total waktu tinggal aktual untuk semua item, dengan B_i menyatakan saat operasi pertama dari komponen yang berada pada level e struktur produk item i yang terjadwal paling pertama, dimesin yang manapun. Sedangkan C_i menyatakan waktu selesai dari semua operasi item i pada semua mesin. Dimana C_i dapat diartikan item i dapat terselesaikan, sebagai dasar penentuan *due date* dengan customer.

Adapun fungsi kendala dari permasalahan ini terdiri dari beberapa kendala yang berhubungan dengan ketersediaan mesin dan hubungan ketergantungan pekerjaan pada setiap levelnya.

- Fungsi kendala hubungan kuantitas item dengan kuantitas komponen.

$$n_{ij} = n_{i0} * H_{ij} \left[\prod_{P_{iq} \in Z(P_{ij})} H_{iq} \right], \quad H_{iq} \geq 1; \forall i, j \quad (3.2)$$

Persamaan (3.2) menyatakan hubungan kuantitas item yang dibuat dengan kuantitas komponen-komponen penyusunnya. H_{ij} menyatakan jumlah komponen ke- j yang diperlukan untuk membuat item i , untuk membuat satu unit induk langsung. Dengan P_{iq} menyatakan komponen ke- q sebagai elemen himpunan $Z(P_{ij})$ dari induk-induk komponen ke- j hingga item i . Persamaan ini untuk menjamin bahwa didalam penentuan jumlah komponen ke- j harus memperhatikan struktur produk penyusunannya.

- Fungsi kendala hubungan tingkat cacat (defect rate) terhadap jumlah komponen yang diproduksi.

$$n_{ijk(diproduksi)} \geq n_{ij(k+1)(diproduksi)} * (1 + dr_{ijkmn}) \quad (3.3)$$

Dimana, $n_{ijk(diproduksi)} \geq n_{ij(k+1)(diproduksi)} * (1 + dr_{ijkmn})$, sehingga persamaan (3.3) menyatakan bahwa jumlah komponen ke- j dari item i pada operasi ke- k yang harus diproduksi, pada kondisi suatu item harus diproses lebih dari satu kali pada mesin yang berbeda, ditentukan oleh jumlah komponen pada operasi ke- $k+1$ dan *defect rate* (dr) untuk semua mesin m yang identik yang tersedia dalam tahapan produksi pada operasi ke- $k+1$, dengan pembulatan dilakukan ke atas.

- Fungsi kendala kelayakan waktu memulai penjadwalan

$$B_{ij1lmn}[N] \geq 0; \forall i, j, l, m, n \quad (3.4)$$

$$B_{ij1lmn}[N] \geq B_i; \forall i, j, l, m, n \quad (3.5)$$

Persamaan (3.4) menjamin bahwa waktu dimulai pemrosesan pertama dari komponen p_{ij} harus lebih besar atau sama dengan nol, yang berarti pekerjaan

tersebut layak dikerjakan didalam horizon waktu perencanaan. Persamaan (3.5) waktu dimulai pemrosesan pertama dari komponen p_{ij} harus lebih besar atau sama dengan waktu dimulai pemrosesan item- i .

- Fungsi kendala hubungan waktu penyelesaian komponen terhadap induk langsungnya.

$$C_{ijhl+1mn} - B_{ijl1mn} \leq 0 ; \forall i, j, k, l, m, n \quad (3.6)$$

Persamaan (3.6) menjamin bahwa pemroses pertama operasi pertama $k=l$ untuk komponen p_{ij} pada level ke- l sebagai induk langsung dari komponen p_{ij} pada level ke- $l+1$, dapat dimulai apabila operasi terakhir $k=h$ komponen p_{ij} telah selesai dikerjakan, disemua mesin.

Untuk permasalahan penentuan *due date* kondisi statis, secara garis besar algoritma yang dikembangkan sebagai berikut:

- Mulai dengan solusi awal. Pembangkitan solusi awal untuk permasalahan penentuan *due date* kondisi statis ditetapkan dengan cara pembangkitan random.
- Menetapkan solusi tetangga, panjang *tabu list* dan jumlah iterasi. Dalam penelitian ini, digunakan jumlah solusi tetangga = 5 ; panjang tabu list = 20 ; dan jumlah iterasi sebanyak 100 kali iterasi.
- Evaluasi fungsi tujuan dari solusi awal.
- Membentuk daftar kandidat pergerakan solusi yang mungkin.
- Jika solusi tidak tabu maka masukan solusi tersebut ke daftar solusi baru.
- Ulangi langkah-langkah diatas hingga iterasi maksimal yang sudah ditentukan diawal.
- Solusi-solusi terbaik yang dapat dimasukkan ke dalam daftar tabu, untuk mencegah pengulangan dalam pencarian solusi yang sama pada tiap iterasi.
- Jika semua iterasi telah dilakukan maka dapat diperoleh *due date* untuk kondisi statis.

Untuk permasalahan penentuan *due date* kondisi dinamis, secara garis besar bekerja dengan tahapan sebagai berikut:

- Mulai dengan solusi awal. Pembangkitan solusi awal untuk permasalahan penentuan *due date* kondisi dinamis juga ditetapkan dengan cara pembangkitan random, tetapi tidak mengganggu penjadwalan yang sudah diperoleh dari kondisi statis.
- Menetapkan solusi tetangga, panjang *tabu list* dan jumlah iterasi. Dalam penelitian ini, digunakan jumlah solusi tetangga = 5 ; panjang tabu list = 20 ; dan jumlah iterasi sebanyak 100 kali iterasi.
- Evaluasi fungsi tujuan dari solusi awal.
- Membentuk daftar kandidat pergerakan solusi yang mungkin.
- Jika solusi tidak tabu maka masukan solusi tersebut ke daftar solusi baru.
- Ulangi langkah-langkah diatas hingga iterasi maksimal yang sudah ditentukan diawal.
- Solusi-solusi terbaik yang dapat dimasukkan ke dalam daftar tabu, untuk mencegah pengulangan dalam pencarian solusi yang sama pada tiap iterasi.
- Jika semua iterasi telah dilakukan maka dapat diperoleh *due date* untuk kondisi dinamis.

BAB 4

PENGUJIAN DAN ANALISA MODEL

4.1 Verifikasi dan Validasi Model

Untuk memastikan algoritma yang dikembangkan dapat bekerja untuk menyelesaikan permasalahan sesuai model yang telah diformulasikan. Didalam proses verifikasi dan validasi ini dilakukan dengan menggunakan program MATLAB. Pemeriksaan apakah kode yang dibuat sudah sesuai dengan prosedur masing-masing algoritma semestinya, serta sudah mengakomodir kendala persoalan yang ada. Selain pemeriksaan pada saat kodifikasi program, verifikasi dilakukan saat akan mengkompilasi kode dan menjalankan program tersebut pada pengujian persoalan kecil. Jika program sudah dapat dijalankan dan keluaran program yang dihasilkan sudah sesuai dengan yang diinginkan maka algoritma yang dikembangkan tersebut dapat dikatakan sudah terverifikasi. Sedangkan untuk memvalidasi program diperlukan langkah-langkah pengecekan terhadap keluaran yang dihasilkan oleh program tersebut setelah berhasil dijalankan. Pengecekan dapat dilakukan misalnya dengan mengecek nilai *total actual flow time*, nilai *due date* dan waktu selesai setiap operasi.

Data hipotesis yang akan digunakan dalam verifikasi dan validasi model secara manual terdiri dari tiga item produk akhir yang diproduksi, dengan masing-masing item memiliki dua level (yaitu level 0 dan level 1 serta satu item memiliki level 2) dan diproses lebih dari satu mesin. Di mana setiap jenis mesin dapat memiliki lebih dari satu mesin identik. Kondisi ini, untuk menjamin bahwa algoritma yang dikembangkan dapat menyelesaikan permasalahan pada lingkungan *job shop* yang memproduksi multi item dan berstruktur multi level, dengan mesin paralel.

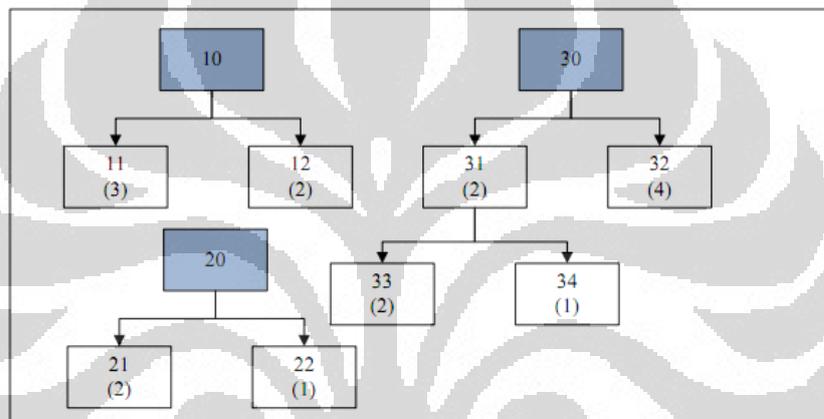
4.1.1 Set Data Pengujian Kondisi Statis

Set data diperoleh dari Data Sekunder Purdianta (2012), dapat dilihat di Lampiran 3. Terdapat 3 (tiga) buah produk atau item yang diproduksi, setiap produk tersebut terdiri dari (2) dua buah komponen dan lebih dengan kebutuhan setiap komponen untuk membentuk produk akhir berbeda-beda seperti pada

Gambar 4.1. Dalam proses produksi digunakan 4 (empat) buah mesin. Adapun data detailnya sebagai berikut.

Tabel 4.1 Data Mesin Tersedia

No	Mesin	Jumlah (Unit)
1	1	3
2	2	1
3	3	2
4	4	2



Gambar 4.1 Struktur Setiap Produk

Adapun kuantitas yang harus diproduksi dan due date dari masing-masing pekerjaan seperti ditunjukkan pada Tabel 4.2, berikut ini.

Tabel 4.2 Data Item, Kuantitas dan *Due Date*

Pio	Kuantitas (unit)	Due-date (menit)*
10	10	3400
20	15	3250
30	20	3300

*untuk permasalahan pemenuhan *due date*

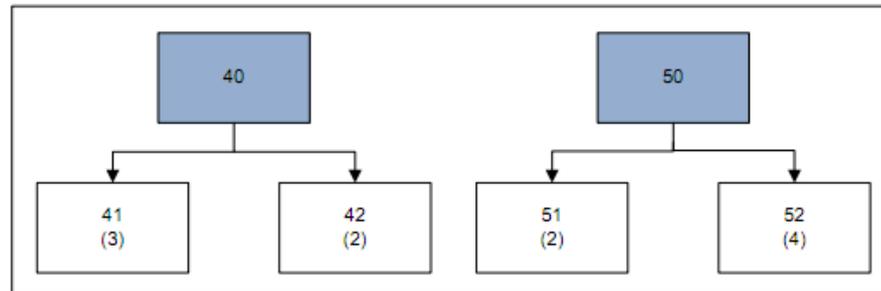
Setiap item dan komponen dilakukan pemrosesan pada setiap mesin sesuai dengan routing atau urutan prosesnya, data waktu set up, waktu proses dan tingkat cacat seperti ditunjukkan pada Tabel 4.3, seperti ditunjukkan berikut ini.

4.1.2 Set Data Pengujian Kondisi Dinamis

Pada saat kegiatan produksi sudah berlangsung terjadi kedatangan pesanan baru pada $T= 2100$ (menit) untuk permasalahan pemenuhan due date dan pada $T= 1100$ untuk permasalahan pemenuhan due date, dimana terjadi dua buah kedatangan pesanan baru dengan struktur produk seperti pada Gambar 4.2, sebagai berikut.

Tabel 4.3 Waktu Proses, Set-Up dan Tingkat Cacat

p_{ij}	Level	$Z(p_{ij})$	h_{ij}	Operation	Mesin	Set-up time (menit)	Waktu proses (menit)	Tingkat Cacat (%)
10	0			1	3	30	10	4
11	1	10	3	1	4	10	5	6
11	1	10	3	2	3	20	5	5
12	1	10	2	1	2	25	10	10
12	1	10	2	2	1	45	15	4
20	0			1	1	40	15	5
21	1	20	2	1	3	15	5	8
21	1	20	2	2	2	45	15	6
22	1	20	1	1	1	100	20	5
22	1	20	1	2	4	50	15	7
30	0			1	2	15	5	8
31	1	30	2	1	3	25	5	8
32	1	30	4	1	1	30	10	6
33	2	31	2	1	3	35	5	5
33	2	31	2	2	4	50	10	10
34	2	31	1	1	4	45	5	5
34	2	31	1	2	2	60	15	8



Gambar 4.2 Struktur Pesanan Baru

Adapun data mengenai kuantitas yang harus diproduksi, due date, urutan pemrosesan, waktu proses, waktu set up dan tingkat cacat seperti ditunjukkan pada Tabel 4.4 dan Tabel 4.5, sebagai berikut.

Tabel 4.4 Data Item, Kuantitas dan *Due Date* Pesanan Baru

Pio	Kuantitas (unit)	<i>Due-date</i> (menit)*
40	15	3875
50	25	3950

*digunakan untuk permasalahan pemenuhan *due date*

Tabel 4.5 Waktu Proses, Set Up dan Tingkat Cacat

p_{ij}	Level	$Z(p_{ij})$	hij	Operation	Mesin	<i>Set-up time</i> (menit)	Waktu proses (menit)	Tingkat Cacat (%)
40	0			1	3	25	10	6
41	1	40	3	1	4	10	1	6
41	1	40	3	2	3	20	5	5
42	1	40	2	1	3	25	10	10
42	1	40	2	2	1	15	15	4
50	0			1	2	15	5	4
51	1	50	4	1	4	20	5	8
52	1	50	1	1	1	15	1	6

4.1.3 Hasil Pengujian Kondisi Statis Pemenuhan *Due-Date*

Dengan menggunakan algoritma model satu yaitu algoritma pemenuhan *due date* untuk penyelesaian model statis maka diperoleh jadwal akhir untuk data kondisi statis sebagaimana ditunjukkan pada Gambar 4.3.

Berdasarkan hasil perhitungan yang telah dilakukan, maka diketahui kapan waktu pemrosesan operasi pertama setiap pekerjaan dimulai, agar dapat diselesaikan tepat pada *due date*-nya, seperti tujuan dari penelitian ini. Dimana dalam penelitian ini tidak diperkenankan terjadinya keterlambatan yang disebabkan oleh terlewatnya waktu penyelesaian pekerjaan yang dinyatakan dalam *due date*.

Dari jadwal akhir yang terbentuk dan perhitungan yang telah dilakukan, maka diketahui masing-masing pekerjaan dimulai pada waktu sebagai berikut.

Tabel 4.6 Saat Mulai Pengerjaan Masing-masing Item

Pio	10	20	30
Saat Mulai menit ke-	2870	2090	1965

Sehingga diperoleh *total actual flow time* (TF) sebesar **3025** menit. Dari Tabel 4.6 terlihat bahwa semua pekerjaan layak untuk dikerjakan karena waktu mulai lebih besar dari nol untuk semua item, dimana untuk item satu (p_{10}) harus dimulai pada $t = 287$, supaya item tersebut dapat diselesaikan tepat pada *due date*-nya. Sedangkan untuk p_{20} dan p_{30} masing-masing pada $t = 2090$ dan $t = 1965$. Dengan waktu mulai dan selesai, dari setiap operasi dan semua item, seperti ditunjukkan pada Tabel 4.7 sebagai berikut.

Tabel 4.7 Waktu Mulai dan Selesai Setiap Operasi (menit ke-)

Oijklmn	Sijklmn	Cijklmn	Mesin
O100131	3290	3400	m31
O111141	2870	3045	m41
O111231	2885	3060	m31
O121121	2875	3125	m21
O121213	2915	3260	m13

Tabel 4.7 Waktu Mulai dan Selesai Setiap Operasi (sambungan)

Oijklmn	Sijklmn	Cijklmn	Mesin
O200111	3010	3250	<i>m11</i>
O211131	2305	2480	<i>m31</i>
O211221	2460	2970	<i>m21</i>
O221111	2090	2450	<i>m11</i>
O221233	2480	2750	<i>m33</i>
O300121	3190	3300	<i>m21</i>
O311131	2435	2905	<i>m31</i>
O321112	1985	2925	<i>m12</i>
O332131	1975	2220	<i>m31</i>
O332241	1995	2485	<i>m41</i>
O342133	1965	2085	<i>m33</i>
O342221	1985	2345	<i>m21</i>

Berdasarkan dari hasil perhitungan, mesin m_{32} tidak digunakan didalam pemrosesan penyelesaian pekerjaan. Hal ini dikarenakan semua pekerjaan dapat terselesaikan dengan menggunakan mesin m_{31} , dengan memberikan hasil TF yang sama besarnya.

4.1.4 Hasil Pengujian Kondisi Dinamis Pemenuhan *Due-Date*

Dengan menggunakan algoritma model dua yaitu algoritma untuk penyelesaian pemenuhan *due date* kondisi dinamis. Maka diperoleh *ganttt chart* akhir seperti pada Gambar 4.4.

Berdasarkan hasil perhitungan yang telah dilakukan, untuk kondisi dinamis terjadi kedatangan pesanan baru pada $T=2100$ yaitu sebanyak dua buah pesanan, yaitu p_{40} dan p_{50} , berdasarkan dari hasil perhitungan kedua pesanan baru tersebut dapat diterima. Perhitungan kondisi dinamis terlebih dahulu melakukan evaluasi terhadap semua operasi dari masing-masing item yang akan diproses ketika terjadi kedatangan pesanan baru. Dari jadwal akhir untuk kondisi

dinamis maka diketahui bahwa pemrosesan dari masing-masing pesanan baru, sebagai berikut:

Tabel 4.8 Saat Mulai Pengerjaan Masing-masing Pesanan Baru

Pio	40	50
Saat Mulai Menit ke-	3050	3234

Sehingga diperoleh *total actual flow time* (TF) sebesar **4566** menit. Dari Tabel 4.9 terlihat waktu mulai dan selesai, dari setiap operasi dan semua item setelah terjadi kedatangan pesanan item baru. Semua operasi dari setiap item yang dalam pemrosesan tidak mengalami perubahan kecuali operasi yang belum dilakukan pemrosesan pada waktu terjadi kedatangan pesanan baru.

Tabel 4.9 Waktu Mulai dan Selesai Setiap Operasi (menit ke-)

Oijklmn	Sijklmn	Cijklmn	Mesin
O100131	3290	3400	m31
O111133	2895	3070	m33
O111231	2910	3085	m31
O121121	2885	3135	m21
O121213	2915	3260	m13
O200111	3010	3250	m11
O211131	2270	2445	m31
O211221	2460	2970	m21
O221111	2530	2890	m11
O221241	2500	2770	m41
O300121	3190	3300	m21
O311131	2400	2870	m31
O321112	1970	2910	m12
O332131	1940	2185	m31
O332233	1960	2450	m33

Tabel 4.9 Waktu Mulai dan Selesai Setiap Operasi (sambungan)

Oijklmn	Sijklmn	Cijklmn	Mesin
O342141	1965	2085	<i>m41</i>
O342221	1985	2345	<i>m21</i>
O400131	3715	3875	<i>m31</i>
O411133	3050	3101	<i>m33</i>
O411231	3435	3690	<i>m31</i>
O421131	3055	3415	<i>m31</i>
O421211	3056	3566	<i>m11</i>
O500121	3820	3950	<i>m21</i>
O511141	3240	3805	<i>m41</i>
O521112	3234	3262	<i>m12</i>

4.1.5 Hasil Pengujian Kondisi Statis Penentuan *Due-Date*

Dengan menggunakan algoritma penentuan *due date* pada model dua, dapat ditentukan *due date* dari masing-masing item seperti ditunjukkan pada Tabel 4.10. Adapun *ganttt chart* dari setiap operasi seperti ditunjukkan pada Gambar 4.5.

Tabel 4.10 Waktu Mulai dan Selesai Operasi Penentuan *Due Date* (menit ke-)

Oijklmn	Sijklmn	Cijklmn	Mesin
O100131	1190	1300	<i>m31</i>
O111133	465	640	<i>m33</i>
O111231	985	1160	<i>m31</i>
O121121	455	705	<i>m21</i>
O121212	485	830	<i>m12</i>
O200113	1300	1540	<i>m13</i>
O211131	295	470	<i>m31</i>
O211221	750	1260	<i>m21</i>
O221113	385	745	<i>m13</i>
O221241	770	1040	<i>m41</i>

Tabel 4.10 Waktu Mulai dan Selesai Operasi Penentuan *Due Date* (sambungan)

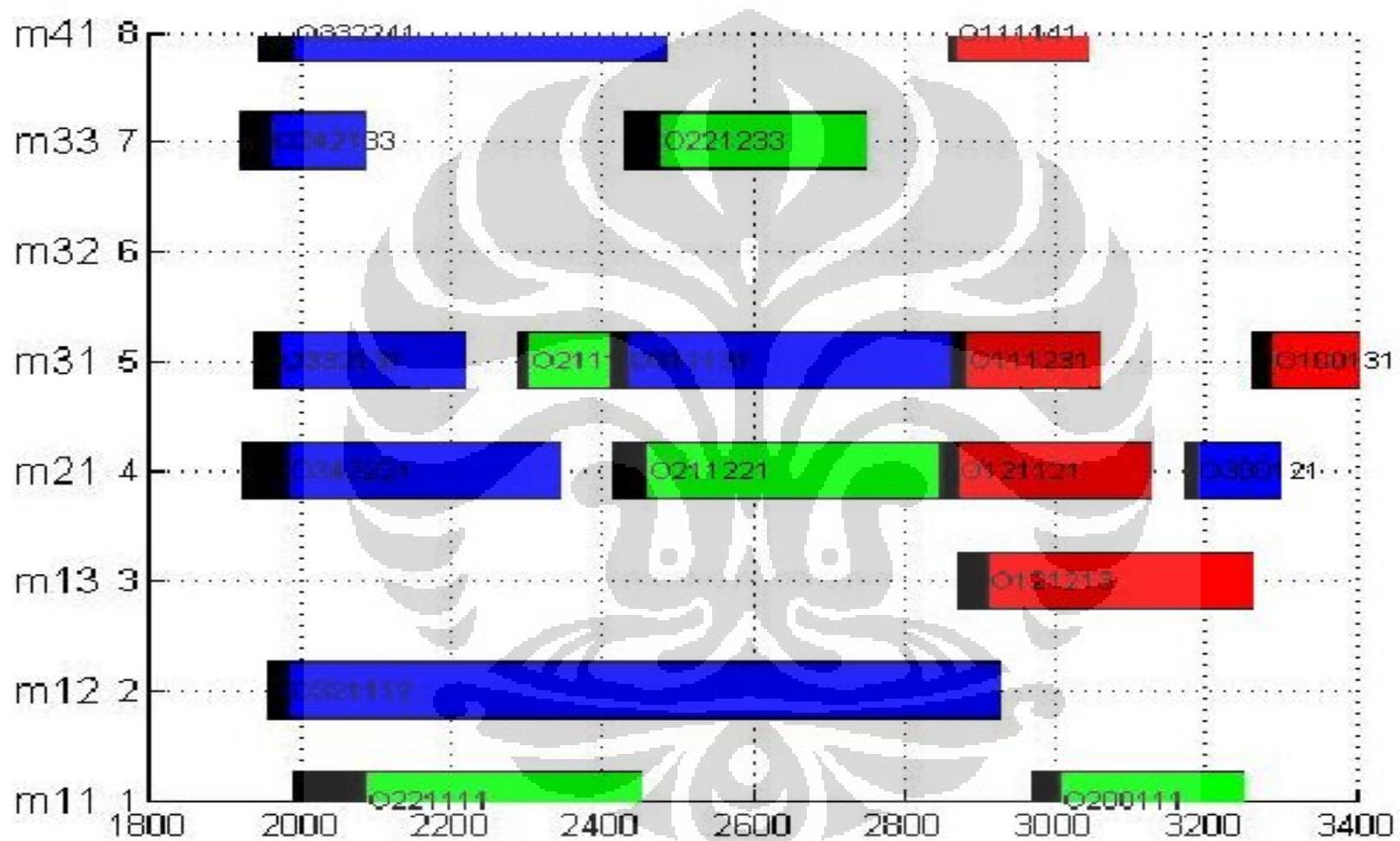
Oijklmn	Sijklmn	Cijklmn	Mesin
O300121	1275	1385	<i>m21</i>
O311131	495	965	<i>m31</i>
O321111	65	1005	<i>m11</i>
O332131	35	280	<i>m31</i>
O332241	75	565	<i>m41</i>
O342133	50	170	<i>m33</i>
O342221	70	430	<i>m21</i>

Sehingga dari waktu mulai dan selesai semua operasi dapat ditentukan *due date* dari setiap item, yang ditunjukkan dari nilai waktu penyelesaian maksimum semua operasi pada suatu item. Dimana waktu penyelesaian maksimum tersebut, menunjukkan waktu pekerjaan dapat diselesaikan dan siap untuk diserahkan kepada pelanggan. *Due date* dari masing-masing item, sebagai berikut.

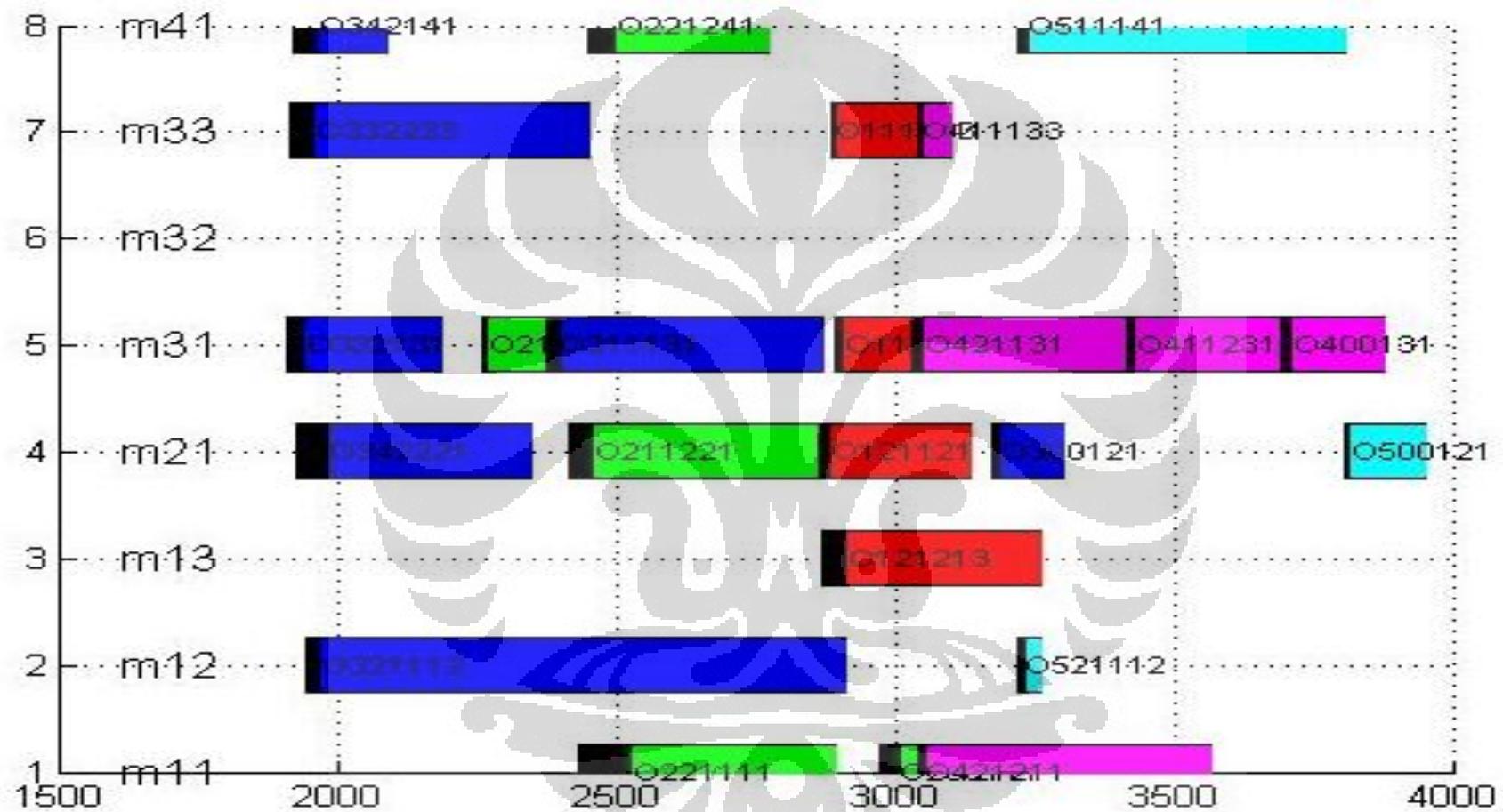
Tabel 4.11 *Due Date* dari Setiap Item

P_{i0}	10	20	30
Due Date (d_{i0})	1300	1540	1385

Total actual flow time yang dihasilkan sebesar 3440 menit. Dimana item p_{10} baru dapat diselesaikan dan diserahkan kepada pelanggan pada $t = 1300$, sedangkan untuk item p_{20} dan p_{30} masing-masing pada $t = 1540$ dan $t = 1385$. Waktu-waktu penyelesaian ini sebagai acuan waktu penyelesaian pekerjaan yang dapat dinegosiasikan kepada pelanggan.



Gambar 4.3 Gantt Chart Akhir Pemenuhan *Due Date* Kondisi Statis



Gambar 4.4 Gantt Chart Akhir Pemenuhan Due Date Kondisi Dinamis

4.1.6 Hasil Pengujian Kondisi Dinamis Penentuan *Due-Date*

Dengan menggunakan algoritma penentuan *due date* pada model dua untuk kondisi dinamis, diperoleh waktu mulai dan selesai dari masing-masing operasi setelah terjadi kedatangan pesanan baru pada $T=1100$, seperti ditunjukkan pada Tabel 4.12. Adapun *ganttt chart* akhir untuk kondisi dinamis seperti ditunjukkan pada Gambar 4.6.

Tabel 4.12 Waktu Mulai dan Selesai Penentuan *Due Date* Dinamis (menit ke-)

Oijklmn	Sijklmn	Cijklmn	Mesin
O100131	1430	1540	m31
O111141	1015	1190	m41
O111231	1045	1220	m31
O121121	1020	1270	m21
O121213	1055	1400	m13
O200112	1035	1275	m12
O211131	300	475	m31
O211221	485	995	m21
O221112	555	915	m12
O221233	525	795	m33
O300121	1285	1395	m21
O311131	500	970	m31
O321111	65	1005	m11
O332131	40	285	m31
O332241	60	550	m41
O342133	45	165	m33
O342221	80	440	m21

Tabel 4.12 Waktu Mulai dan Selesai Penentuan *Due Date* Dinamis (sambungan)

Oijklmn	Sijklmn	Cijklmn	Mesin
O400131	1786	1946	m31
O411133	1110	1161	m33
O411231	1506	1761	m31
O421131	1126	1486	m31
O421212	1126	1636	m12
O500121	1701	1831	m21
O511141	1121	1686	m41
O521111	1115	1143	m11

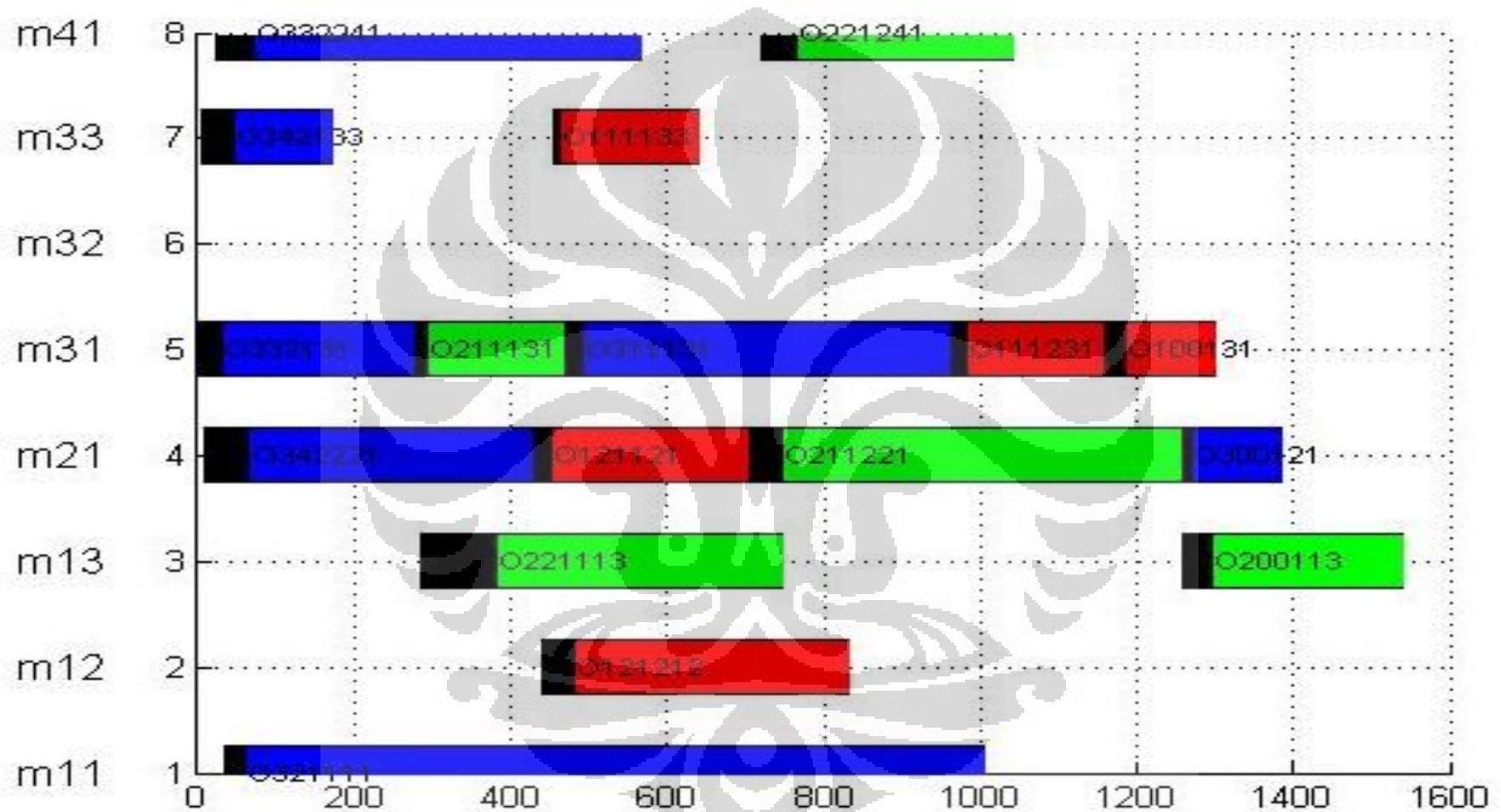
Setelah terjadi kedatangan pesanan baru, maka *total actual flow time* baru yang dihasilkan sebesar **4992** menit, dengan *due date* dari pesanan baru seperti ditunjukkan pada Tabel 4.13, sebagai berikut

Tabel 4.13 *Due Date* dari Setiap Item

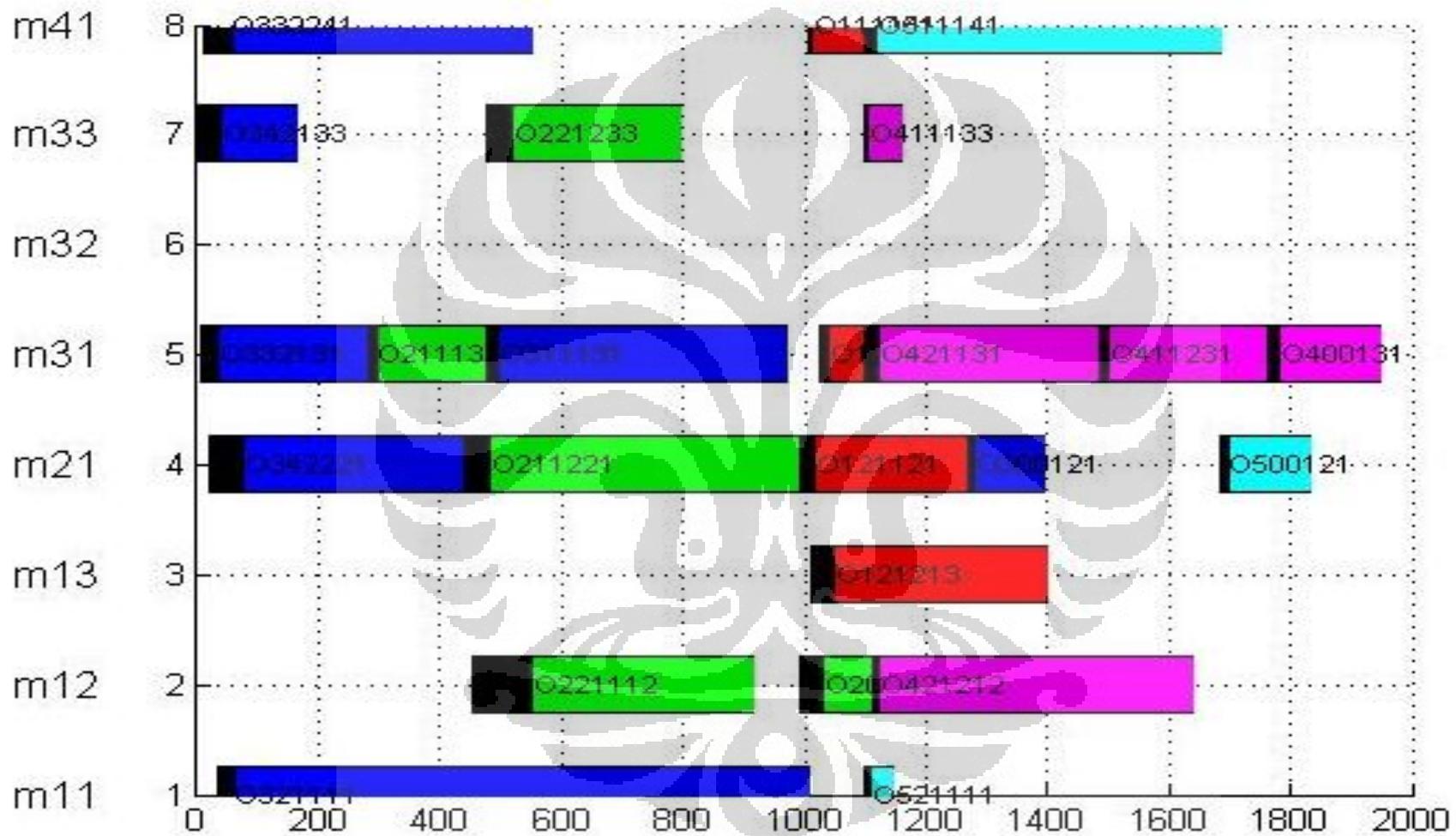
P_{i0}	40	50
Due Date (d_{i0})	1946	1831

4.2 Pengujian Model

Set data yang digunakan dalam pengujian model adalah data sistem manufaktur yang memproses sebanyak 10 jenis item dengan jumlah dan due-date yang berbeda-beda dan dikerjakan pada 5 buah mesin, dimana setiap mesin terdiri dari 3 buah mesin identik. Struktur produk seluruh item terdiri dari 4 level dan komponen pada suatu level tertentu memiliki sebanyak 2 buah komponen pembentuk pada level dibawahnya. Penetapan 2 buah komponen pembentuk ini (dengan jumlah unit masing masing komponen berbeda) dimaksudkan untuk dapat mewakili kondisi bahwa suatu komponen memiliki multi-komponen pembentuknya.



Gambar 4.5 Gantt Chart Akhir Penentuan Due Date Kondisi Statis



Gambar 4.6 Gantt Chart Akhir Penentuan Due Date Kondisi Dinamis

Tiap komponen memiliki sebanyak 2 buah operasi yang dikerjakan pada 2 buah mesin yang berbeda. Penetapan 2 buah operasi dimaksudkan untuk dapat mewakili kondisi bahwa suatu komponen memiliki multi-operasi, sementara pengerjaan pada 2 buah mesin yang berbeda adalah untuk menjaga tingkat variabilitas permasalahan penjadwalan. Tiap item dan komponen-komponennya juga memiliki waktu set-up dan waktu operasi (keduanya dalam satuan menit) pada tiap mesin dalam *routing* masing-masing yang besarnya berbeda beda. Serta *defect rate* yang berbeda untuk setiap tahapan pemrosesan atau mesin yang dilakui. Secara rinci data pengujian yang digunakan diperlihatkan pada Lampiran1.

Pengujian dilakukan untuk mendapatkan jumlah alternatif jadwal untuk suatu jumlah jenis item tertentu dan jumlah level produk yang tertentu pula, serta rata-rata *CPU processing time*. Proses pengujian dilakukan dengan program komputer yang dikembangkan dengan menggunakan MATLAB.

4.2.1 Hasil Pengujian Pemenuhan *Due Date*

Dari hasil pengujian yang dilakukan diperoleh alternatif jadwal dan waktu rata-rata *CPU time* (dalam satuan detik), seperti ditunjukkan pada Tabel 4.14 dan Tabel 4.15 sebagai berikut.

Tabel 4.14 Alternatif Jadwal Pemenuhan *Due Date*

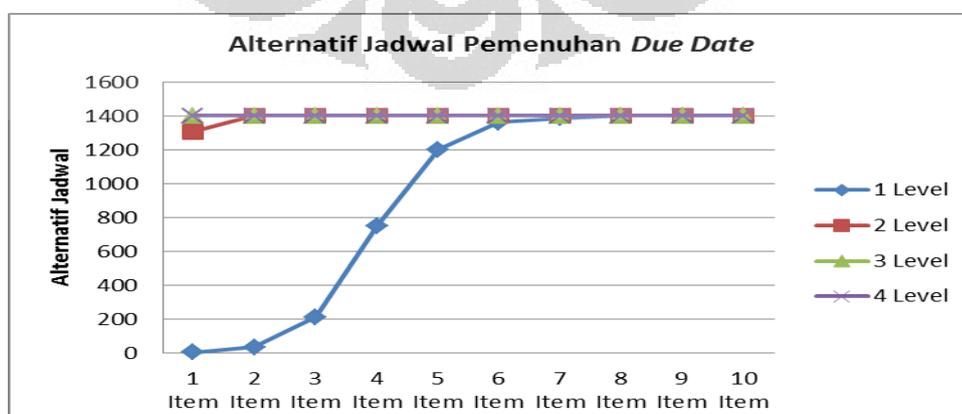
	1 Level	2 Level	3 Level	4 Level
1 Item	6	1307	1401	1401
2 Item	36	1401	1401	1401
3 Item	212	1401	1401	1401
4 Item	748	1401	1401	1401
5 Item	1200	1401	1401	1401
6 Item	1362	1401	1401	1401
7 Item	1388	1401	1401	1401
8 Item	1400	1401	1401	1401
9 Item	1401	1401	1401	1401
10 Item	1401	1401	1401	1401

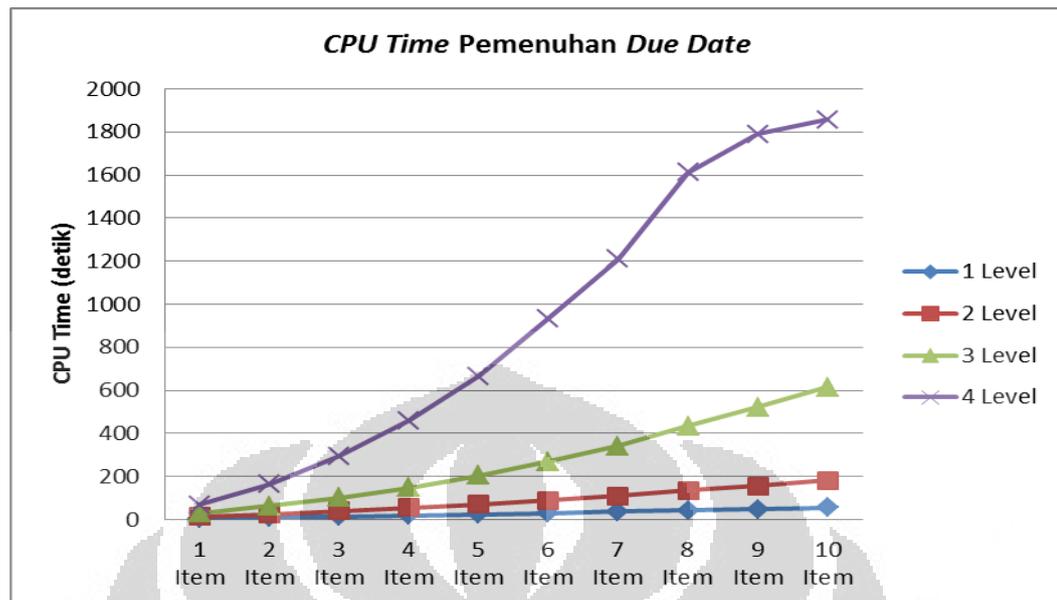
Tabel 4.15 CPU Time Pemenuhan *Due Date* (dalam detik)

	1 Level	2 Level	3 Level	4 Level
1 Item	6,948	13,872	29,383	68,694
2 Item	11,02	26,038	62,428	164,949
3 Item	15,71	39,893	102,786	295,442
4 Item	20,58	55,081	149,708	460,721
5 Item	25,856	71,832	206,053	665,915
6 Item	31,252	89,899	270,01	933,58
7 Item	37,013	110,472	340,852	1209,15
8 Item	42,831	134,857	434,192	1611,15
9 Item	49,395	155,584	522,234	1789,15
10 Item	56,157	181,639	614,333	1856,11

Tabel 4.14 menunjukkan jumlah alternatif jadwal yang dapat terbentuk dari setiap kondisi perlakuan pemenuhan *due date* untuk item 1 sampai dengan item 10 dengan level yang digunakan mulai dari 1 level sampai dengan 4 level. Sedangkan Tabel 4.15 menunjukkan waktu CPU Time yang diperlukan untuk pencarian solusi dari setiap permasalahan sesuai kondisi perlakuan yang diberikan.

Dari Tabel diatas terlihat bahwa algoritma yang dikembangkan dalam penelitian ini tidak memberikan alternatif jadwal yang bervariasi untuk tingkatan item maupun level. Hal ini bisa jadi disebabkan karena inisialisasi solusi tetangga, panjang tabu list dan jumlah iterasi yang ditetapkan diawal membuat pencarian solusi berhenti di titik lokal optimum. Adapun grafik hubungan antara jumlah item dengan level terhadap peningkatan jumlah alternatif jadwal dan CPU Time, seperti ditunjukkan pada Gambar 4.7.

Gambar 4.7 Grafik Alternatif Jadwal Pemenuhan *Due Date*



Gambar 4.8 Grafik *CPU Time Pemenuhan Due Date*

4.2.2 Hasil Pengujian Penentuan *Due Date*

Dari hasil pengujian yang dilakukan diperoleh alternatif jadwal dan waktu rata-rata *CPU time* (dalam satuan detik), seperti ditunjukkan pada Tabel 4.16 dan Tabel 4.17 sebagai berikut.

Tabel 4.16 Alternatif Jadwal Penentuan *Due Date*

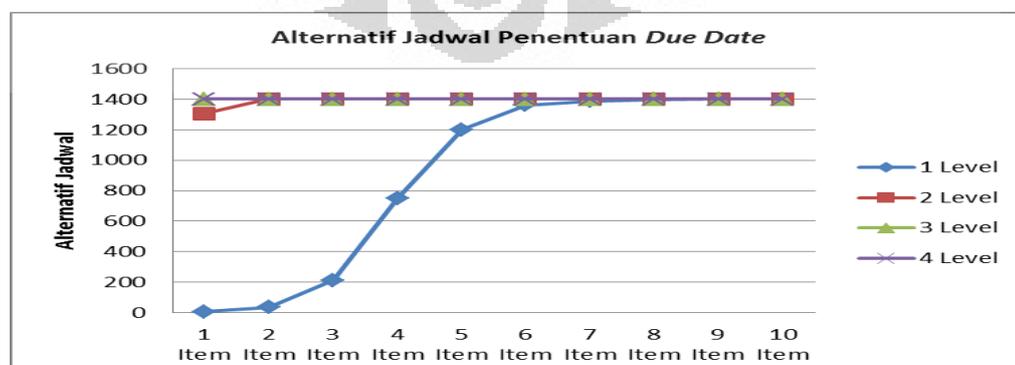
	1 Level	2 Level	3 Level	4 Level
1 Item	6	1307	1401	1401
2 Item	36	1401	1401	1401
3 Item	212	1401	1401	1401
4 Item	748	1401	1401	1401
5 Item	1200	1401	1401	1401
6 Item	1362	1401	1401	1401
7 Item	1388	1401	1401	1401
8 Item	1400	1401	1401	1401
9 Item	1401	1401	1401	1401
10 Item	1401	1401	1401	1401

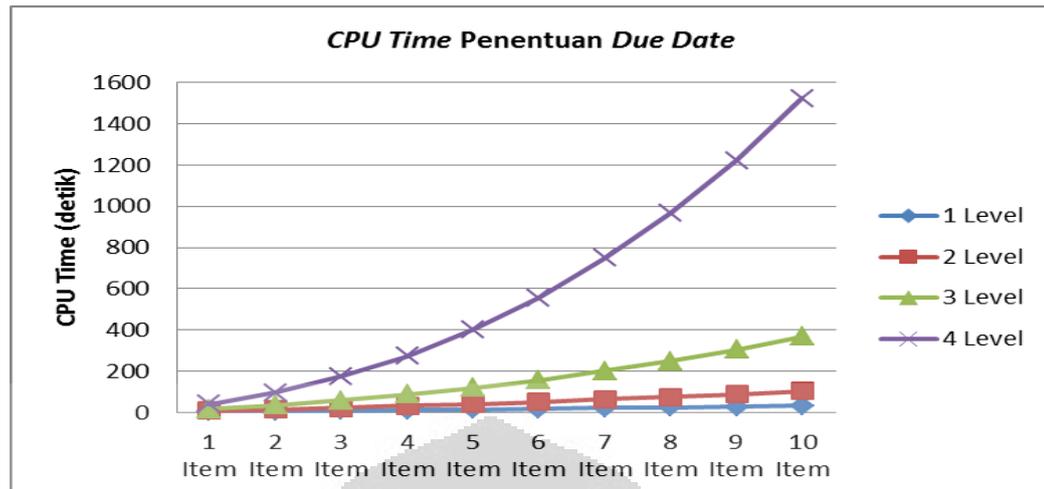
Tabel 4.17 CPU *Time* Penentuan *Due Date* (dalam detik)

	1 Level	2 Level	3 Level	4 Level
1 Item	3,802	8,009	17,874	40,093
2 Item	6,473	15,019	36,011	95,855
3 Item	8,889	22,964	59,50	175,68
4 Item	12,105	31,382	87,51	274,36
5 Item	14,446	40,686	120,752	401,763
6 Item	17,312	50,993	157,47	556,46
7 Item	20,619	63,041	200,512	748,199
8 Item	24,094	75,502	248,291	969,488
9 Item	27,6848	89,094	305,119	1222,84
10 Item	31,379	103,939	367,393	1523,29

Tabel 4.16 menunjukkan jumlah alternatif jadwal yang dapat terbentuk dari setiap kondisi perlakuan penentuan *due date*, untuk item 1 sampai dengan item 10 dengan level yang digunakan mulai dari 1 level sampai dengan 4 level. Sedangkan Tabel 4.17 menunjukkan waktu CPU *Time* yang diperlukan untuk pencarian solusi dari setiap permasalahan sesuai kondisi perlakuan yang diberikan.

Dari Tabel diatas terlihat bahwa algoritma yang dikembangkan dalam penelitian ini tidak memberikan alternatif jadwal yang bervariasi untuk tingkatan item maupun level. Hal ini bisa jadi disebabkan karena inisialisasi solusi tetangga, panjang tabu list dan jumlah iterasi yang ditetapkan diawal membuat pencarian solusi berhenti di titik lokal optimum. Adapun grafik hubungan antara jumlah item dengan level terhadap peningkatan jumlah alternatif jadwal dan CPU *Time*, seperti ditunjukkan pada Gambar 4.9.

Gambar 4.9 Grafik Alternatif Jadwal Penentuan *Due Date*



Gambar 4.10 Grafik *CPU Time* Penentuan *Due Date*

4.3 Analisis Model

Berdasarkan pada data-data yang diperoleh pada sub bab 4.2.1 dan sub bab 4.2.2, dapat dilakukan analisa untuk hal-hal sebagai berikut:

Peningkatan jumlah item pada tingkatan level tertentu memberikan pengaruh terhadap peningkatan alternatif jadwal yang dapat terbentuk secara signifikan untuk kondisi satu level. Misalnya pada kondisi 1 level dengan jumlah item sebanyak 3 buah terbentuk sebanyak 212 alternatif jadwal sedangkan dengan jumlah item sebanyak 4 buah terbentuk sebanyak 748 alternatif pada permasalahan pemenuhan *due date* artinya terjadi 3,52 kali peningkatan alternatif jadwal. Tetapi ini hanya terjadi untuk satu level multi item saja. Untuk dua level dan seterusnya tidak memberikan alternatif jadwal yang bervariasi untuk tingkatan item maupun level. Hal ini bisa jadi disebabkan karena inisialisasi solusi tetangga, panjang tabu list dan jumlah iterasi yang ditetapkan diawal membuat pencarian solusi berhenti di titik lokal optimum.

Peningkatan alternatif jadwal yang terbentuk berpengaruh terhadap waktu *CPU Time* yang diperlukan untuk melakukan penyusunan operasi-operasi sesuai dengan urutan proses pemrosesannya. Pada kondisi 3 item dengan 1 level diperlukan sebesar 15,71 detik sedangkan 3 item dengan 2 level diperlukan sebesar 39,893 detik. Artinya ada peningkatan waktu 2,47 kali.

Kondisi ini menunjukkan bahwa peningkatan jumlah item akan memberikan peningkatan jumlah alternatif jadwal yang berdampak pada CPU

time yang diperlukan untuk mendapatkan solusi yang dapat memberikan *total actual flow time* yang minimum.

4.4 Analisis Perbandingan Performa Hasil Penelitian Sebelumnya

Pada penelitian sebelumnya (Purdianta 2012), algoritma yang digunakan adalah Algoritma Inseri. Ada perbedaan mendasar yang dilakukan pada penelitian sebelumnya yaitu penetapan inisialisasi awal pada algoritma. Algoritma inseri menggunakan *Latest Due Date* pada permasalahan pemenuhan *due date* dan *Shortest Processing Time* pada permasalahan penentuan *due date*. Pada Algoritma *Tabu Search* digunakan inisialisasi *Latest Due Date* pada permasalahan pemenuhan *due date* guna mencapai kondisi *Just In Time* sedangkan pada permasalahan penentuan *due date* digunakan pembangkitan secara random.

- Pemenuhan *Due Date* Statis

Total Actual Flow Time pada algoritma inseri sebesar 3108,15 menit sedangkan pada algoritma *tabu search* 3025 menit. Dapat disimpulkan bahwa untuk permasalahan ini algoritma *tabu search* memiliki performansi yang lebih baik karena memberikan *Total Actual Flow Time* yang lebih kecil.

- Pemenuhan *Due Date* Dinamis

Total Actual Flow Time pada algoritma inseri sebesar 4194,05 menit sedangkan pada algoritma *tabu search* 4566 menit. Dapat disimpulkan bahwa untuk permasalahan ini algoritma inseri memiliki performansi yang lebih baik karena memberikan *Total Actual Flow Time* yang lebih kecil. Hal ini bisa jadi disebabkan karena perbedaan inisialisai solusi di awal.

- Penentuan *Due Date* Statis

Perbandingan *Due Date* pada algoritma inseri dan algoritma *tabu search* untuk permasalahan status dapat dilihat pada Table 4.18. Dari Tabel tersebut dapat disimpulkan bahwa algoritma *tabu search* memberikan *due date* yang lebih lama dibandingkan algoritma inseri untuk item 10. Sedangkan untuk item 30 algoritma *tabu search* memberikan *due date* yang lebih cepat dibandingkan algoritma inseri. Perbedaan penentuan *due*

date tersebut bisa disebabkan karena adanya perbedaan metode inisialisasi yang digunakan diawal, dimana algoritma insersi menggunakan metode SPT sedangkan algoritma *tabu search* melakukan pembangkitan solusi awal dengan cara random. Hal ini bisa juga disebabkan karena perbedaan prioritas penjadwalan, dalam algoritma insersi penjadwalan dilakukan per item sedangkan pada algoritma TS penjadwalan dilakukan per level artinya menyelesaikan semua yang di level rendah terlebih dahulu.

Tabel 4.18 Perbandingan *Due Date* Kondisi Statis

	Item 10	Item 20	Item 30
Inseri	500	1080	2030
TS	1300	1540	1385

- Penentuan *Due Date* Dinamis

Perbandingan *Due Date* pada algoritma insersi dan algoritma *tabu search* untuk permasalahan dinamis dapat dilihat pada Table 4.19. Dari Tabel tersebut dapat disimpulkan bahwa algoritma *tabu search* memberikan *due date* yang lebih lama dibandingkan algoritma insersi untuk item 40 dan item 50. Perbedaan penentuan *due date* tersebut bisa disebabkan karena adanya perbedaan metode inisialisasi yang digunakan diawal, dimana algoritma insersi menggunakan metode SPT sedangkan algoritma *tabu search* melakukan pembangkitan solusi awal dengan cara random.

Tabel 4.19 Perbandingan *Due Date* Kondisi Dinamis

	Item 40	Item 50
Inseri	1865	1825
TS	1946	1831

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Algoritma yang dikembangkan dapat menyelesaikan permasalahan pemenuhan dan penentuan *due date* baik pada kondisi statis dan dinamis, dengan tujuan meminimalkan *total actual flow time* dan mempertimbangkan *defect rate*. Peningkatan jumlah item dan jumlah level akan memberikan peningkatan jumlah alternatif jadwal yang berdampak pada *CPU time* yang diperlukan untuk mendapatkan solusi yang dapat memberikan *total actual flow time* yang minimum, namun dalam penelitian ini hanya terjadi untuk satu level multi item saja. Untuk dua level dan seterusnya tidak memberikan alternatif jadwal yang bervariasi untuk tingkatan item maupun level. Hal ini disebabkan karena inisialisasi solusi tetangga, panjang tabu list dan jumlah iterasi yang ditetapkan diawal membuat pencarian solusi berhenti di titik lokal optimum.

5.2 Saran

Dari hasil penelitian tugas akhir ini, dapat dilakukan pengembangan seperti berikut ini.

1. Pengembangan model yang memungkinkan dilakukannya proses penentuan jumlah dan ukuran *batch (batching)*, pengurutan operasi item, dan penjadwalan *batch* yang dihasilkan secara simultan.
2. Penelitian untuk menghasilkan algoritma penjadwalan pada kondisi mesin parallel dengan mempertimbangkan aktivitas perawatan atau *breakdown machine*.
3. Pengembangan model yang memperhatikan aspek finansial dalam mengambil keputusan terhadap kedatangan pesanan baru berdasarkan keuntungan dan penalti dari konsumen atas keterlambatan penyelesaian pekerjaan.

DAFTAR REFERENSI

Cheng, TCE et al., (2007). Due-Date Assignment And Paralle Machine Scheduling With Deteriorating Jobs. *Journal of Operational Research Society*, Vol.58, No.8, pp.1103-1108.

Huang, R.H.,Yang,C.L. (2010) Multi-objective Job-Shop Scheduling With Lot-Splitting Production. *International Journal of Production Economics*.,pp.206-213

Khasnan, Ali Husseinzadeh et al.,(2008). A Hybrid Genetic herutistic for Scheduling Paralel Batch Processing Machines With Arbitrary Job Sizes. *Journal Computer and Operation Research*. Vol 35, pp.1084-1098.

Liang et al (2010). An Effective Hybrid Tabu Search Algorithm For Multi-Objective Flexible Job-Shop Scheduling Problems. *Journal Computer and Industrial Engineering*. Vol 59, pp.647-662

Mosheiov, Gur., Oron, Daniel. (2006). Due-date Assignment and Maintenance Activity Scheduling Problem, *Mathematical and Computer Modeling Journal*,Vol.44, pp 1053-1057.

Purdianta. (2012). Model Dinamis Penjadwalan Job Shop pada Mesin Paralel dengan menggunakan Algoritma Inseri.

Shabtay,Dvir. (2010).Scheduling and Due Date Assignment to Minimize Earliness, Tardiness, Holding, Due Date Assignment and Batch Delibery Costs. *International Journal of Production Economics*. Vol 123, pp.235-242.

Tuong, Nguyen Huynh and Soukhal Ameer. (2010). Due Date Assignment and JIT Scheduling with Equal-Size Jobs. *European Journal of Operational Research*.Vol 205, pp.280-289.

Vinod, V., R. Sriharan. (2011). Simulation Modeling and Analysis of due-date Assignment Methods and Scheduling Decision Rules in Dynamic Job Shop Production System. *International Journal Production Economics*, Vol.129,pp.127-146.

Xia,Yu et al.(2008), Job Squencing and Due Date Assignment in a Single Machine Shop with Uncertain Processing Times. *European Journal of Operational Research*. Vol.184. pp.63-75.

Zhang et al (2006). A Tabu Search Algorithm With A New Neighborhood Structure For The Job Shop Scheduling Problem. *Journal Computer and Operation Research*. Vol 34, pp.3299-3242

Zhang et al (2012). A Genetic Algorithm With Tabu Search Procedure For Flexible Job Shop Scheduling With Transportation Constraints And Bounded Processing Time. *Journal Computer and Operation Research*. Vol 39, pp.1713-1723