

BATCHING & SEQUENCING

**Model Matematis untuk
Sistem Manufaktur Flow Shop**

Tengku Nurainun

Sekapur Sirih

Bismillahirrahmanirrahim,
Assalamu'alaikum, Wr. Wb.

Puji syukur kehadiran Allah SWT atas rahmat dan karunia-Nya penerbitan buku hasil penelitian dosen pada Fakultas Sains dan Teknologi UIN Suska Riau tahun 2012 dapat diwujudkan.

Kami selaku Dekan menyambut baik antusiasme dan semangat penelitian dari para dosen di lingkungan FST UIN Suska hingga akhirnya dapat menghasilkan karya-karya yang bernilai dan bermanfaat bagi perkembangan ilmu pengetahuan. Pembuatan buku ini bertujuan agar setiap hasil penelitian dosen dapat disebarluaskan sehingga karya tersebut tidak hanya bermanfaat bagi penulis tetapi juga bagi masyarakat ilmiah, baik dosen maupun mahasiswa, setidaknya di lingkungan UIN Suska.

Kami mengharapkan semoga kehadiran buku ini dapat memperkaya literatur yang sudah ada dan bermanfaat bagi pembaca.

Wassalam,
Dekan Fakultas Sains dan Teknologi

Dra. Hj. Yenita Morena, M.Si

Kata Pengantar



Assalamu'alaikum, Wr, Wb.

Penjadwalan produksi merupakan aktivitas yang sangat penting pada suatu Perusahaan. Keputusan yang diambil oleh seorang *scheduler* sangat menentukan masa depan perusahaan karena tidak hanya berefek pada keuntungan secara finansial saja namun juga secara tidak langsung menyentuh pada loyalitas konsumen terhadap perusahaan berkaitan dengan komitmen akan ketepatan pengiriman produk ke tangan konsumen.

Kendati demikian, tidak semua Jurusan Teknik Industri menempatkan mata kuliah Penjadwalan Produksi ini di dalam kurikulum sebagai mata kuliah wajib. Selain itu, referensi yang membahas khusus mengenai Teori Penjadwalan terutama yang disajikan dalam bahasa Indonesia pun masih sangat jarang. Oleh karena itu, penulis berkeinginan untuk menuangkan Teori Penjadwalan, khususnya mengenai Penjadwalan *Batch*, ke dalam buku untuk memperkaya referensi.

Buku *Batching & Sequencing* ini berisikan teori dasar tentang Penjadwalan Produksi menggunakan pendekatan matematis. Buku ini tidak menyajikan secara utuh semua teori tentang penjadwalan *batch* namun fokus pada pengembangan model matematis untuk sistem manufaktur *flow shop* dinamis. Kehadiran buku ini diharapkan dapat membantu para mahasiswa dan praktisi yang ingin meningkatkan keterampilannya di dalam memodelkan sistem penjadwalan serta menjadi referensi untuk pengembangan keilmuan lebih lanjut.

Akhir kata, tak ada gading yang tak retak, tak ada manusia yang luput dari kesalahan. Penulis menyadari bahwa karya ini masih terlalu jauh dari sempurna. Oleh karenanya, penulis sangat mengharapkan masukan dari pembaca demi kesempurnaan tulisan ini.

Selamat membaca.

Wassalam,

Penulis

Daftar Isi

Sekapur Sirih.....	i
Kata Pengantar.....	i
Daftar Isi.....	iii
Daftar Tabel.....	v
Daftar Gambar.....	vii
Bab I Pengantar	1
1.1 Fungsi Penjadwalan.....	1
1.2 Terminologi.....	3
1.3 Ukuran Performansi	5
1.4 Tahap-tahap Penjadwalan.....	7
1.5 Gantt Chart.....	8
Bab 2 Model Penjadwalan	11

2.1	Klasifikasi Sistem.....	11
2.2	Sistem Statis dan Dinamis.....	14
2.3	Aproksimasi.....	16

Bab 3 Penjadwalan *Batch* 19

3.1	Penjadwalan <i>Job</i>	21
3.2	Penjadwalan <i>Batch</i>	23
3.3	Biaya Produksi.....	25
3.3.1	Sistem Biaya berdasarkan Pesanan.....	25
3.3.2	Sistem Biaya berdasarkan Proses.....	27

Bab 4 *State of The Art* 29

4.1	<i>Review Literature</i>	30
4.2	Posisi Penelitian.....	32

Bab 5 Membangun Model Matematis..... 37

5.1 Model Statis 39

 5.1.1 Kriteria Kinerja 40

 5.1.2 Parameter..... 41

 5.1.3 Variabel Keputusan..... 42

 5.1.4 Formulasi Model..... 42

5.2 Model Dinamis 45

Bab 6 Algoritma *Batching*..... 51

6.1 Algoritma Statis (Algoritma BFSS) 52

6.2 Algoritma Dinamis (Algoritma BFSD)..... 54

6.3 Sub Algoritma Pemecahan *Batch* 57

Bab 7 Studi Kasus..... 59

7.1 Persoalan 1 59

7.2 Persoalan 2 64

Bab 8 Validitas 71

8.1 Rancangan Pengujian Model..... 71

8.1.1 Skenario 1..... 72

8.1.2 Skenario 2..... 73

8.1.3 Skenario 3..... 74

8.1.4 Skenario 4..... 74

8.2 Hasil Pengujian dan Analisis..... 75

8.2.1 Skenario 1..... 75

8.2.2 Skenario 2..... 78

8.2.3 Skenario 3..... 81

8.2.4 Skenario 4..... 81

Bab 9 Penutup 83

Daftar Pustaka.

Lampiran

Daftar Tabel

Tabel		Halaman
4.1	Posisi Penelitian	34
7.1	Data Waktu Proses dan Biaya Setup Kasus Statis.....	60
7.2	Data Biaya WIP dan Biaya Simpan Kasus Statis	60
7.3	Jadwal untuk $N = 2$ (Kasus Statis)	61
7.4	Total Biaya Produksi Setiap <i>Batch</i> (Kasus Statis)	62
7.5	Solusi Optimal untuk Kasus Statis.....	63
7.6	Data Waktu Proses Produk Baru.....	64
7.7	Data Biaya WIP dan Biaya Simpan Kasus Dinamis.....	64
7.8	Availabilitas Mesin.....	65
7.9	Jadwal untuk $N = 2$ (Kasus Dinamis)	66

7.10	Total Biaya Produksi Setiap <i>Batch</i> (Kasus Dinamis)	67
7.11	Solusi Optimal untuk Kasus Dinamis	68
8.1	Data Hipotetik Waktu Proses dan Biaya <i>Setup</i>	72
8.2	Data Hipotetik Biaya WIP dan Biaya Simpan	72
8.3	Data Hipotetik Skenario 1	73
8.4	Data Hipotetik Skenario 2.....	73
8.5	Data Hipotetik Waktu Proses dan Biaya <i>Setup</i> Skenario 3.....	74
8.6	Data Hipotetik Biaya WIP dan Biaya Simpan Skenario 3	74
8.7	Hasil Pengujian Skenario 1	75
8.8	Solusi Optimal untuk Skenario 1	76
8.9	Hasil Pengujian Skenario 2.....	80
8.10	Perbandingan Waktu Komputasi	81
8.11	Perbandingan TC Akibat Perubahan Nilai Parameter Biaya	82

Daftar Gambar

Gambar		halaman
1.1	Contoh Gantt <i>Chart</i>	9
2.1	Model Penjadwalan.....	13
2.2	Contoh Sistem Dinamis.....	16

2.3	Pendekatan Maju	17
2.4	Pendekatan Mundur	17
6.1	<i>Flow Chart</i> Algoritma BFSS.....	53
6.2	<i>Flow Chart</i> Algoritma BFSD.....	56
7.1	<i>Gantt Chart</i> Hasil Penjadwalan Kasus Statis.....	63
7.2	<i>Gantt Chart</i> Posisi <i>Batch</i> Sisa	65
7.3	<i>Gantt Chart</i> Jadwal Revisi	69
9.1	Aplikasi <i>Software</i> Lingo.....	84



PENGAN TAR

1.1 Fungsi Penjadwalan

Di dalam sistem perencanaan

produksi, penjadwalan merupakan aktivitas pengambilan keputusan pada level terbawah (eksekusi). Pada level tersebut jenis-jenis operasi yang akan dilakukan serta informasi lain yang berkaitan dengan pekerjaan tersebut seperti kuantitas dan *deadline* penyelesaian pekerjaan sudah terdefinisi dengan jelas. Artinya, proses penjadwalan baru akan dimulai setelah bagian perencanaan selesai menelurkan keputusan yang bersifat fundamental seperti produk apa yang akan diproduksi, skala produksi, bagaimana teknis pembuatan produk tersebut, dan ketersediaan fasilitas produksi.

Fungsi penjadwalan menjadi sangat penting karena berkaitan dengan pengambilan keputusan strategis tentang penggunaan secara bersama sumber-sumber daya yang dimiliki dalam memroses berbagai macam pekerjaan. Oleh karena itu, seorang *scheduler* selalu berhadapan dengan pertanyaan tentang “**pekerjaan**

mana yang harus diprioritaskan untuk diproses terlebih dahulu (sequencing)?”.

Hal ini akan berdampak secara langsung kepada efisiensi penggunaan sumber daya yang dimiliki dan berujung kepada penghematan biaya, sehingga untuk mendapatkan hasil keputusan terbaik benar-benar dibutuhkan strategi didalam penyusunan jadwal produksi tersebut.

Penjadwalan didefinisikan oleh Baker (1974) sebagai proses alokasi sumber-sumber dalam jangka waktu tertentu untuk melakukan sejumlah pekerjaan. Sedangkan menurut Morton dan Pentico (1993) penjadwalan merupakan pengambilan keputusan tentang penyesuaian aktivitas dan sumber daya dalam rangka menyelesaikan sekumpulan pekerjaan agar tepat pada waktunya dan mempunyai kualitas seperti yang diinginkan. Keputusan yang dimaksud dalam hal ini menyangkut tentang :

1. Pengurutan pekerjaan (*sequencing*),
2. Waktu mulai dan selesai pekerjaan (*timing/release*),
3. Urutan operasi untuk suatu pekerjaan (*routing*).

Dengan demikian, masalah penjadwalan muncul karena terbatasnya sumber daya yang dimiliki untuk mengerjakan sejumlah tugas pada saat bersamaan dan akibat dari keterbatasan tersebut perlu diambil keputusan mengenai urutan pengerjaan tugas dan alokasi dari sumber daya yang akan digunakan.

Oleh karena itu, pencapaian hasil yang optimal merupakan tujuan yang selalu dikehendaki di dalam pengambilan keputusan penjadwalan, akan tetapi kompleksitas permasalahan terkadang menyebabkan pencapaian tersebut semakin mustahil, dikenal dengan istilah NP-complete, sehingga kemudian berkembang pendekatan-pendekatan heuristik dalam usaha mencapai hasil

penjadwalan yang setidaknya mampu mendekati hasil optimal dengan waktu komputasi yang diharapkan tidak terlalu lama.

1.2 Terminologi

Terdapat banyak sekali istilah yang digunakan didalam penjadwalan dan dilambangkan dalam notasi-notasi unik untuk memudahkan didalam proses memodelkan sistem. Diantara istilah-istilah tersebut beberapa istilah yang umum digunakan antara lain :

1. **Processing time (waktu proses, t_j)** merupakan estimasi waktu penyelesaian pekerjaan (*job, task*).
2. **Setup time (waktu setup, s_j)** adalah waktu yang dibutuhkan untuk kegiatan persiapan sebelum pemrosesan *job* dilaksanakan.
3. **Ready time (saat siap, r_j)** yaitu saat sebuah *job* siap

diproses.

4. **Due date (saat batas, d_j)** merupakan *deadline* untuk *job* dimana pengerjaan yang melewati batas tersebut dinyatakan terlambat.
5. **Arrival time (saat datang, a_j)** adalah saat *job* mulai berada di *shop floor* (*production line*, lantai produksi).
6. **Delivery date (saat kirim, d_j)** ialah saat pengiriman *job* dari *shop floor* ke proses berikutnya atau ke konsumen.
7. **Waiting time (waktu menunggu)** adalah waktu *job* menunggu karena mesin yang seharusnya memroses *job* tersebut pada saat yang bersamaan juga sedang memroses *job* lain.
8. **Idle time (waktu menganggur)** yaitu waktu mesin tidak bekerja (menganggur) karena tidak ada *job* yang harus diproses.

Dalam penelitian bidang

penjadwalan, istilah “*schedule*” dan “*sequence*” biasanya mempunyai pengertian yang saling dipertukarkan (*schedule* = *sequence*) meskipun sebenarnya istilah *schedule* mengandung cakupan yang lebih luas dimana *schedule* selain menggambarkan urutan (*sequence*) dari pemrosesan *job* juga menggambarkan saat dimulai dan saat selesai dari suatu pekerjaan. Demikian juga dengan istilah-istilah lainnya seperti *job*, *task*, aktivitas, dan sebagainya.

1.3 Ukuran Performansi

Ukuran performansi tentu saja berkaitan dengan tujuan yang ingin dicapai dari suatu proses pengambilan keputusan. Didalam proses penjadwalan kita harus membedakan antara atribut yang akan digunakan, dalam hal ini dikategorikan sebagai *input* dari fungsi penjadwalan, dengan variabel

keputusan yang merupakan hasil dari keputusan penjadwalan (*output* dari fungsi penjadwalan).

Terdapat perbedaan dalam penulisan notasi untuk atribut dan variabel keputusan dimana biasanya atribut dilambangkan dengan huruf kecil (*lower case letters*) sedangkan variabel keputusan dilambangkan dengan huruf besar (*capital letters*).

Beberapa *output* (kriteria) dari keputusan penjadwalan yang umum digunakan untuk mengevaluasi jadwal, antara lain :

1. **Completion time** (**saat selesai, C_j**) yaitu saat penyelesaian pengerjaan dari *job j*.
2. **Flow time** (**waktu tinggal, F_j**) menunjukkan lamanya suatu *job j* berada di dalam sistem yang merupakan waktu antara saat datang (*arrival time*) dengan saat kirim (*delivery date*).

3. **Lateness (L_i)** menunjukkan perbedaan antara saat selesai (*completion time*) dengan *due date*. Terdapat dua kondisi berkaitan dengan *lateness*, yaitu:
 - a. **Tardiness (positive *lateness*, T_i)** yaitu menunjukkan waktu keterlambatan (*completion time* melampaui *due date*).
 - b. **Earliness (negative *lateness*, E_i)** untuk menunjukkan kondisi bahwa suatu tugas diselesaikan lebih awal dari *due date*.
4. **Makespan (M)** adalah interval waktu total yang dibutuhkan untuk menyelesaikan seluruh *job*.
5. **Slack** yaitu sisa waktu sampai *due date* dari suatu *job*, merupakan selisih antara *due date*, waktu yang dibutuhkan untuk memroses *job* tersebut, dan saat sekarang.

Ukuran performansi merupakan kuantitas agregat yang

diperoleh untuk mengevaluasi jadwal dimana kriteria yang ingin dicapai dapat berupa minimisasi maupun maksimisasi, misalnya: meminimasi *mean flow time*, meminimasi *makespan*, memaksimalkan *minimum job lateness*, dan sebagainya.

Pemilihan ukuran performansi ini disesuaikan dengan kebutuhan dalam pengambilan keputusan apakah berdasarkan sudut pandang perusahaan atau berdasarkan pertimbangan dalam pemenuhan keinginan konsumen. Misalnya kriteria performansi penjadwalan untuk produk yang diproduksi secara massal akan berbeda dengan kriteria performansi penjadwalan untuk produk yang bersifat *customized (based on order)* dimana terdapat komitmen waktu penyelesaian antara produsen dengan konsumen yang harus dipenuhi untuk menjaga kredibilitas perusahaan.

1.4 Tahap-tahap Penjadwalan

Perencanaan produksi sebagai bagian dari tingkat pengambilan keputusan level atas dan menengah tidak dapat dipisahkan secara independen dengan aktivitas penjadwalan produksi. Keputusan manajerial dari bagian perencanaan produksi yang secara perkiraan kapasitas dinilai mampu dipenuhi oleh sumber daya yang dimiliki belum tentu fisibel setelah dievaluasi dengan hasil penjadwalan produksi.

Dengan demikian, berdasarkan jadwal tentatif yang telah dibuat jika ternyata hasilnya tidak fisibel maka *scheduler* harus memberikan respon balik kepada perencana produksi agar divisi tersebut melakukan revisi terhadap rencana yang telah dibuat, misalnya mengurangi jumlah produksi atau memberikan alternatif keputusan lainnya seperti menambah fasilitas produksi untuk memenuhi kuantitas produk yang diinginkan. Proses ini terus berlangsung hingga diperoleh

keputusan akhir berupa jadwal produksi yang mapan.

Keputusan penjadwalan dicapai melalui serangkaian tahapan menggunakan pendekatan sistem (*system approach*). Sebagaimana diketahui bahwa ada empat tahapan utama dalam pendekatan sistem, yaitu : tahap membangun formulasi, analisis, sintesis, dan evaluasi.

1. **Formulasi** berperan sebagai tahap penentu baik atau tidaknya hasil keputusan yang akan dicapai. Pada tahap ini, permasalahan diidentifikasi dan atribut yang diperlukan dalam proses pengambilan keputusan harus sudah ditentukan secara jelas. Tahap ini terkadang sulit dan kompleks karena kriteria performansi yang ingin dicapai juga harus sudah ditetapkan.
2. **Analisis** merupakan proses identifikasi variabel keputusan dan spesifikasi

keterkaitan hubungan antara satu variabel dengan variabel lainnya serta batasan-batasan (*constraint*) yang harus diikuti.

3. **Sintesis** adalah proses membangun alternatif solusi yang mungkin dari permasalahan. Dalam hal ini seluruh opsi yang fisibel dipertimbangkan untuk mendapatkan hasil keputusan yang terbaik.
4. **Evaluasi** merupakan tahapan akhir yang harus dilalui untuk mencapai keputusan final yaitu membandingkan semua alternatif solusi yang mungkin kemudian memilih alternatif terbaik sebagaimana kriteria performansi yang telah ditetapkan.

1.5 Gantt Chart

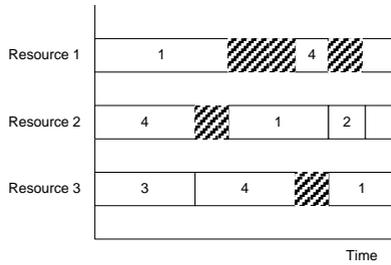
Model grafis biasanya cukup handal dalam memberikan

pemahaman tentang gambaran sistem yang sedang diamati. Model grafis yang sangat sederhana dan dipergunakan secara luas untuk merepresentasikan jadwal produksi adalah *Gantt chart*.

Gantt chart merupakan peta visual yang menggambarkan *loading* (beban mesin) dan *scheduling* dimana *loading* dan *scheduling* dipetakan pada grafik dua dimensi yang terdiri atas sumbu vertikal (sumber daya, mesin) dan sumbu horizontal (skala waktu).

Gantt chart dapat berfungsi sebagai alat untuk menghasilkan jadwal, yaitu dengan membaca perilaku dari sistem berdasarkan jadwal sementara yang telah dibuat dan dipetakan ke dalam grafik kemudian memanipulasi elemen-elemen pada grafik tersebut sebagai usaha untuk mengembangkan alternatif solusi dalam pengambilan keputusan, atau bisa pula *Gantt chart* hanya berfungsi sebagai media

penyampaian informasi tentang suatu hasil penjadwalan tanpa kemudian melakukan perubahan terhadap solusi yang diperoleh.



Gambar 1.1 Contoh Gantt Chart



MODEL PENJADWALAN

2.1 Klasifikasi Sistem

Teori penjadwalan selalu berhubungan dengan model matematis yang disusun dalam bentuk fungsi penjadwalan. Pendekatan kuantitatif dilakukan dengan mengamati struktur dari permasalahan kemudian

diterjemahkan ke dalam bentuk formulasi matematis dimana tujuan yang ingin dicapai dirumuskan sebagai fungsi tujuan (*objective function*) dan aturan-aturan yang berlaku dirumuskan sebagai batasan (*constraint*).

Elemen penting didalam membangun model penjadwalan adalah *resources* (sumber daya, mesin) dan *tasks* (pekerjaan). Konfigurasi mesin dan urutan operasi dari suatu pekerjaan sangat menentukan dalam proses merumuskan batasan (*technological constraints*), disamping adanya batasan kapasitas yang dimiliki oleh masing-masing sumber daya yang tersedia.

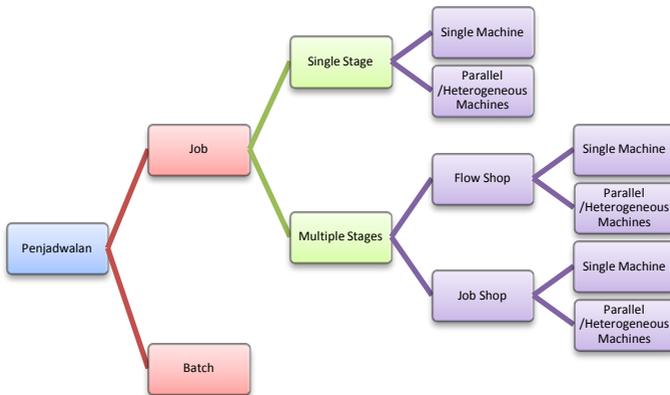
Struktur masalah yang dihadapi dalam penjadwalan dapat berupa :

1. **Penjadwalan *job* (*job scheduling*)** dimana ukuran *job* telah diketahui sehingga permasalahan yang harus dipecahkan hanya *sequencing* saja.
2. **Penjadwalan *batch* (*batch scheduling*)** yaitu mengelompokkan pekerjaan ke dalam *batch* dengan suatu ukuran tertentu sehingga pengerjaan *batch* dapat dilakukan dengan sekali *setup* untuk mencapai kriteria performansi tertentu.

Pada penjadwalan *batch*, permasalahan yang dihadapi bukan hanya pada *sequencing* saja melainkan juga pada penentuan ukuran *batch* dimana kedua hal tersebut diselesaikan secara simultan.

Sedangkan konfigurasi *resources* yang dimaksud dapat berupa :

1. **Satu tahap (*single stage*)** yaitu apabila pemrosesan hanya melalui satu tahapan dimana pada stasiun kerja tersebut kemungkinan bisa terdiri atas mesin tunggal maupun beberapa mesin seragam yang disusun secara paralel.
2. **Multi tahap (*multiple stage*)** yaitu apabila pemrosesan melalui lebih dari satu tahapan dimana susunan (*layout*) dari mesin dapat berupa aliran (*flow shop*) maupun disusun berdasarkan fungsi dari mesin (*job shop*). Dalam setiap tahapan pun terdapat kemungkinan stasiun kerja terdiri atas mesin tunggal atau mesin paralel.



Gambar 2.1 Model Penjadwalan

Dengan demikian permasalahan penjadwalan akan selalu berkaitan dengan pengambilan keputusan mengenai dua

hal penting yaitu keputusan mengenai pengalokasian sumber daya dan keputusan mengenai pengurutan pekerjaan (*sequencing*).

Informasi yang dibutuhkan mengenai sumber daya antara lain jenis-jenis sumber daya yang tersedia dan jumlah dari masing-masing jenis sumber daya. Sedangkan informasi yang dibutuhkan terkait dengan pekerjaan misalnya sumber daya yang dibutuhkan untuk memroses suatu pekerjaan, durasi pekerjaan (waktu proses), saat kedatangan (*arrival time*), *due date*, urutan proses operasi, dan sebagainya.

Secara umum permasalahan penjadwalan dapat diselesaikan menggunakan teknik-teknik optimisasi. Semakin meningkat kompleksitas permasalahan maka penyelesaian model penjadwalannya juga akan semakin rumit.

2.2 Sistem Statis dan Dinamis

Sebagaimana di dalam teori pemodelan sistem disebutkan bahwa kompleksitas sistem yang akan dituangkan ke dalam model mengharuskan pembuat model untuk melakukan simplifikasi karena keterbatasan metode pemecahan solusi. Dengan kata lain, model pasti lebih sederhana daripada sistem nyata.

Demikian pula halnya dalam membangun model penjadwalan. Kompleksitas permasalahan penjadwalan

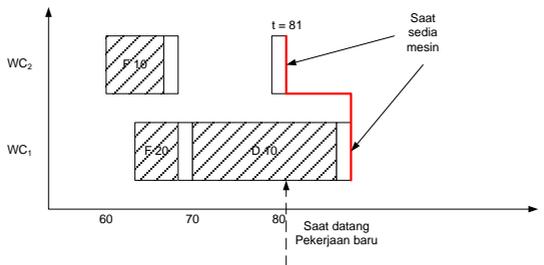
seringkali disederhanakan dengan menetapkan asumsi-asumsi keberlakuan model. Contoh sederhana misalnya, model penjadwalan yang mengasumsikan bahwa bahan tersedia setiap saat diperlukan. Padahal kenyataannya tidak selalu bahan bisa tersedia setiap saat. Ada kalanya terjadi ketidakterediaan material akibat keterlambatan kedatangan pesanan, kerusakan bahan, bahan tidak sesuai pesanan, dan sebagainya.

Salah satu penyederhanaan yang umum digunakan adalah asumsi sistem statis dan sistem dinamis. Pada sistem statis berarti selama proses produksi dianggap tidak terjadi gangguan (*disturbance*) berupa kedatangan pekerjaan baru, ketidakterediaan mesin yang disebabkan karena *preventive maintenance* maupun *downtime* mesin akibat kerusakan, serta kondisi lainnya selama proses produksi yang menyebabkan pekerjaan menjadi terhenti dan perlu dilakukan penjadwalan ulang (*re-scheduling*). Penyederhanaan ini sulit terjadi di dunia nyata karena hampir tidak mungkin sistem bisa berjalan secara sempurna setiap saat.

Oleh karena itu penyederhaan model berupa sistem statis berkembang menjadi sistem dinamis untuk lebih mendekati kondisi nyata. Pemodelan sistem statis tetap dilakukan sebagai titik keberangkatan pengembangan model menuju ke arah sistem dinamis. Kendati demikian pemodelan sistem dinamis tetap harus melakukan simplifikasi dimana tidak semua kemungkinan gangguan sistem dapat dipertimbangkan dalam perumusan sebuah model dan dipecahkan karena peningkatan kompleksitas

permasalahan memunculkan kesulitan dalam pencarian solusi.

Pada Gambar 2.2 diperlihatkan contoh kondisi apabila terjadi kedatangan pekerjaan baru pada saat $t = 81$ dimana pada saat tersebut proses produksi sedang berlangsung sehingga perlu dilakukan revisi terhadap jadwal yang sudah ada untuk memenuhi semua kebutuhan. Model penjadwalan yang mampu menyelesaikan permasalahan seperti ini disebut sebagai model dinamis.



Gambar 2.2 Contoh Sistem Dinamis

2.3 Aproksimasi

Pendekatan dasar yang digunakan dalam menyusun suatu jadwal dapat dibedakan menjadi dua pendekatan, yaitu pendekatan maju (*forward approach*) dan pendekatan mundur (*backward approach*).

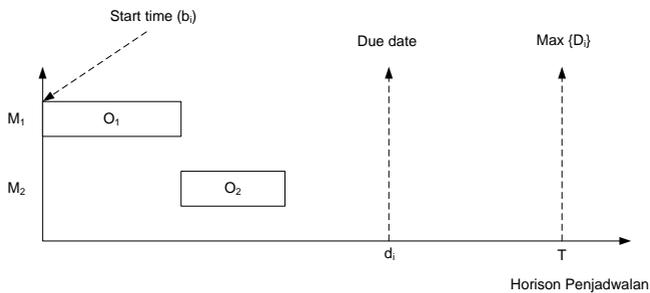
Dalam Morton dan Pentico (1993) definisi dari kedua pendekatan ini adalah:

1. Pendekatan maju

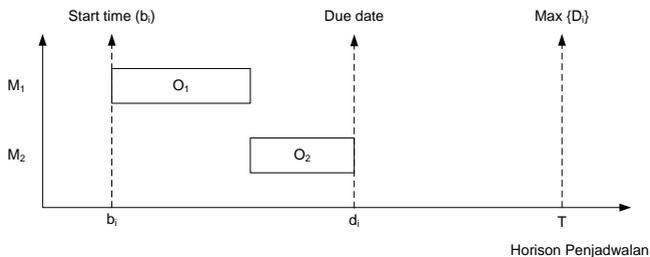
Penjadwalan maju didefinisikan sebagai pengurutan pekerjaan dimulai dari arah saat nol (*time zero*) atau saat sekarang bergerak maju ke arah yang akan datang.

2. Pendekatan mundur

Penjadwalan mundur merupakan penjadwalan yang dimulai dari *due date* dan bergerak mundur ke arah *time zero*.



Gambar 2.3 Pendekatan Maju



Gambar 2.4 Pendekatan Mundur

Pendekatan maju maupun mundur dipilih menurut kriteria yang ingin dicapai dan tuntutan dari sistem itu sendiri. Apabila sistem menghendaki tidak terjadi keterlambatan pengiriman produk (sistem *just in time*) maka pendekatan

mundur lebih tepat untuk dipergunakan karena produk selesai tepat pada saat diperlukan yaitu pada saat *due date*.

Kemungkinan yang terjadi dari penggunaan pendekatan mundur adalah jadwal menjadi tidak fisibel akibat kedatangan pekerjaan mendahului *starting time* ($T = 0$). Salah satu kriteria yang populer digunakan untuk mengukur performansi dengan pendekatan mundur adalah Waktu Tinggal Aktual (*Actual Flow Time*).

Sedangkan pada pendekatan maju akan dihasilkan suatu jadwal layak, tetapi tidak menjamin *due date* akan terpenuhi. Pendekatan ini mengizinkan pekerjaan diselesaikan lebih maju atau lebih mundur dari *due date* yang ditetapkan, namun dalam pencarian solusi diusahakan deviasi antara waktu penyelesaian pekerjaan dengan *due date* minimal. Dengan demikian kriteria yang tepat untuk digunakan sehubungan dengan hal ini adalah *earliness*, *tardiness*, *completion time*, *number of tardy jobs*, dan sebagainya.



PENJADWALAN BATCH

Perkembangan teknologi modern pada saat ini memberikan kemungkinan bagi para inovator untuk memproduksi produk dengan tingkat variasi yang sangat tinggi. Pergeseran ini mengikut pada pola permintaan konsumen yang variatif demi memenangkan persaingan pasar karena tingkat persaingan konsumen yang sangat ketat.

Pada produksi massal produk dibuat dalam jumlah yang sangat besar dengan variasi yang sedikit. Dengan

demikian pembebanan biaya produksi didistribusikan pada unit produk yang sangat banyak sehingga produsen dapat menjual produk dengan harga bersaing serta terus meningkatkan keuntungan.

Namun tipe produksi massal pada mayoritas jenis produk saat ini sudah tidak cocok lagi untuk diterapkan karena peningkatan jumlah produsen yang sedemikian tinggi dan mampu menawarkan jenis produk yang bervariasi. Akibatnya sistem manufaktur mulai beradaptasi dan bergeser dari sistem produksi massal menjadi sistem produksi *batch*. Hal ini tentu saja berdampak pada sistem penjadwalan. Penjadwalan *batch* menjadi semakin rumit karena memuat tidak hanya keputusan mengenai pengurutan pekerjaan (*sequencing*) namun juga keputusan mengenai ukuran *batch* (*batching*).

Penggunaan sumber daya yang sama untuk mengerjakan produk yang bervariasi berpeluang terhadap peningkatan biaya terutama biaya *setup*. Perbedaan spesifikasi dari setiap varian produk menuntut penyesuaian pada rantai produksi dimana proses *setting* membutuhkan waktu tersendiri dan menyebabkan proses produksi terhenti sementara dalam memenuhi kebutuhan tersebut. Semakin lama dan sering aktivitas *setting* mesin ini dilakukan maka akan semakin lama pula investasi yang tertahan pada setiap produk yang semestinya diproduksi. Hal ini perlu disiasati dengan menyusun jadwal yang mampu meminimasi biaya produksi sehingga diharapkan akan memaksimalkan keuntungan.

3.1 Penjadwalan Job

Pada penjadwalan *job* waktu *setup* tidak dispesifikasi secara terpisah dengan waktu proses sehingga ukuran pekerjaan dianggap telah diketahui (*given*) dan keputusan yang diambil hanya berkaitan dengan penentuan jadwal saja. Hal ini karena pengerjaan produk yang seragam dalam jumlah besar menjadikan waktu antar *setup* menjadi sangat panjang dan apabila waktu *setup* ini didistribusikan secara merata pada setiap produk yang sejenis tersebut menjadikan waktu *setup* per produk menjadi sangat kecil sehingga bisa diabaikan.

Penjadwalan *job* bisa dilakukan mengikuti aturan FCFS (*First Come First Serve*), SPT (*Shortest Processing Time*), LPT (*Longest Processing Time*), EDD (*Earliest Due Date*), dan CR (*Critical Ratio*).

Contoh :

Tentukan urutan pengerjaan dari 8 *job* yang akan diproses pada mesin yang sama sehingga rata-rata waktu tinggal dari seluruh pekerjaan minimal. Diketahui waktu pemrosesan masing-masing pekerjaan adalah sebagai berikut,

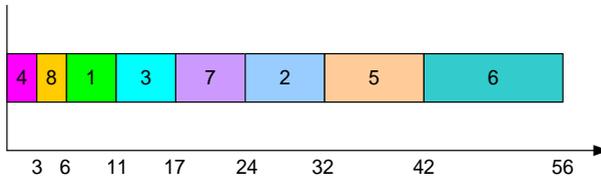
Job	Waktu Proses
1	5
2	8
3	6
4	3
5	10
6	14
7	7
8	3

Solusi :

Berdasarkan data di atas, urutan pekerjaan menurut waktu pemrosesan terpendek adalah job 4-8-1-3-7-2-5-6. Rata-rata waktu tinggal yang diperoleh adalah sebesar 23,875 dengan perhitungan sebagai berikut :

Job	Waktu Proses	Flow Time
4	3	3
8	3	6
1	5	11
3	6	17
7	7	24
2	8	32
5	10	42
6	14	56
F		191
\bar{F}		23,875

Gantt chart dari urutan pekerjaan dapat digambarkan sebagai berikut:



3.2 Penjadwalan *Batch*

Penggunaan istilah “*batch*” dapat terjadi pada sistem manufaktur yang bersifat diskrit maupun pada sistem manufaktur yang bersifat *continuous*. Meskipun berbeda dari sisi penggunaan namun secara umum bisa dikatakan bahwa istilah “*batching*” mengacu kepada aktivitas pengelompokan sejumlah pekerjaan (manufaktur) atau material cair/liquid (*chemical process*) ke dalam suatu ukuran tertentu dan diproses secara bersama-sama.

Pengerjaan sejumlah *job* yang dikelompokkan dalam suatu ukuran *batch* bisa dilakukan pada sistem manufaktur *job shop* maupun *flow shop*. Kriteria performansi yang digunakan juga tidak berbeda dengan penjadwalan *job* tetapi variabel keputusan dari penjadwalan *batch* sedikit berbeda dimana selain harus menentukan jadwal yang mengoptimasi kriteria yang ingin dicapai juga harus memutuskan tentang ukuran *batch*.

Pencarian solusi dapat dilakukan secara matematis maupun heuristik. Pembangunan sebuah model matematis dapat mengarahkan seorang *scheduler* pada pencapaian solusi optimal. Namun hal ini akan membutuhkan waktu

yang relatif lebih lama karena pencarian solusi optimal berarti melakukan enumerasi terhadap seluruh alternatif solusi yang mungkin dimana pemecahan *batch* akan berdampak pada peningkatan alternatif solusi sehingga memperpanjang proses pencarian.

Oleh karena itu berkembang berbagai pendekatan heuristik yang berbasis pada karakteristik dari penjadwalan *batch* sehingga proses pencarian solusi bisa dipersingkat namun solusi yang dihasilkan tidak dijamin optimalitasnya.

Selanjutnya pada buku ini akan diperlihatkan salah satu hasil penelitian dari penulis mengenai model penjadwalan *batch* pada sistem manufaktur *flow shop* dinamis untuk meminimasi biaya produksi. Penelitian tersebut melakukan pencarian solusi menggunakan pendekatan matematis untuk memperoleh hasil yang optimal. Dengan demikian diharapkan pembaca memperoleh gambaran mengenai langkah-langkah pembentukan model matematis dari suatu sistem penjadwalan sehingga membantu pembaca untuk memahami dan mengaplikasikan pada permasalahan penjadwalan yang ada.

Sebelumnya pada bab ini akan ditambahkan terlebih dahulu sub bab mengenai biaya produksi sebagai kriteria yang ingin dicapai pada contoh aplikasi model matematis yang akan dibahas pada bab berikutnya.

3.3 Biaya Produksi

Tujuan penting dari sistem perhitungan biaya adalah untuk menentukan biaya dari barang atau jasa yang dihasilkan oleh perusahaan. Sistem perhitungan biaya sebaiknya ekonomis untuk dioperasikan dan membebankan sejumlah biaya ke sejumlah produk sedemikian rupa sehingga merefleksikan biaya dari sumber daya yang digunakan untuk memproduksi produk tersebut.

Setiap perusahaan memiliki perbedaan dalam hal teknologi manufaktur, pengelolaan sistem produksi dan bauran produk, oleh karena itu dapat diperkirakan bahwa sistem perhitungan biayanya juga akan berbeda. Sistem perhitungan biaya sebaiknya disesuaikan agar dapat memenuhi kebutuhan pelanggan.

3.3.1 Sistem Biaya berdasarkan Pesanan

Pada sistem manufaktur berdasarkan pesanan (*job order cost system*), produk dipertanggungjawabkan dalam *batch*. Setiap *batch* diperlakukan sebagai pesanan yang terpisah, dan pesanan tersebut merupakan objek biayanya. Semua biaya yang terjadi dalam produksi suatu pesanan dibebankan ke kartu biaya pesanan tersebut.

Jika pekerjaan yang dilakukan untuk suatu pesanan terjadi di lebih dari satu *department* atau pusat biaya

(*cost center*) yang lain, biaya yang terjadi di setiap pusat biaya diakumulasikan di kartu biaya pesanan. Ketika pesanan selesai, biaya pesanan per unit dan produk ditentukan dengan membagi total biaya yang dibebankan ke kartu biaya pesanan dengan jumlah unit yang diproduksi untuk pesanan tersebut.

Ketika produk yang diproduksi selama periode akuntansi dalam suatu pusat biaya memerlukan kuantitas dan kombinasi sumber daya yang berbeda-beda, perhitungan biaya berdasarkan pesanan merupakan pilihan yang logis karena banyak biaya untuk memproduksi produk yang berbeda-beda adalah tidak sama. Dalam perhitungan biaya berdasarkan pesanan, produk yang berbeda dapat diproduksi untuk pesanan yang berbeda dan biayanya dapat ditentukan secara terpisah.

Sebaliknya, ketika semua unit dari produk yang dihasilkan dalam suatu pusat biaya adalah serupa (homogen), pencatatan biaya dari setiap *batch* produk secara terpisah tidak lagi diperlukan serta tidak praktis. Lebih baik menggunakan perhitungan berdasarkan proses.

3.3.2 Sistem Biaya berdasarkan Proses

Dalam perhitungan biaya berdasarkan proses (*process cost system*), bahan baku, tenaga kerja, dan *overhead* dibebankan ke pusat biaya. Biaya yang dibebankan ke setiap unit ditentukan dengan membagi total biaya yang dibebankan ke pusat biaya dengan total unit yang diproduksi. Persyaratan utama adalah semua produk yang diproduksi dalam suatu pusat biaya selama suatu periode harus sama dalam hal sumber daya yang dikonsumsi, bila tidak, perhitungan biaya berdasarkan proses dapat mendistorsi biaya produk.



STATE OF THE ART

Dalam membangun sebuah formulasi matematis maupun algoritma heuristik dalam rangka memperkaya referensi di bidang penjadwalan khususnya, perlu dilakukan *literature review* untuk melihat perkembangan penelitian yang sudah dilakukan peneliti-peneliti sebelumnya dan menempatkan penelitian yang akan dilakukan jika

dibandingkan dengan penelitian-penelitian yang telah ada tersebut. Hal ini bertujuan untuk menghindari duplikasi dan membuktikan orisinalitas dari penelitian yang akan dilakukan. *Literature* yang dipilih lebih mengerucut pada penelitian mengenai penjadwalan *batch* pada *flow shop* dinamis sesuai dengan tujuan akhir dari penelitian yang akan dilakukan. Pada sub bab berikut akan diperlihatkan bagaimana *literature review* dan penempatan posisi penelitian tersebut dilakukan.

4.1 Review Literature

Penelitian mengenai penjadwalan *batch* pada sistem manufaktur *flow shop* telah banyak dilakukan antara lain oleh Kurniawan (2003) dan Suryadhini (2006). Namun diantara penelitian-penelitian yang telah dilakukan masih mengabaikan adanya gangguan (*disturbance*) selama horizon perencanaan. Pada kondisi nyata, sistem produksi sering mengalami banyak gangguan antara lain kerusakan mesin, kemunculan pekerjaan (*job*) baru, pembatalan pekerjaan, perubahan *due date*, dan sebagainya sehingga mengakibatkan jadwal yang telah dibuat harus direvisi.

Setiawati dan Halim (2003) mengembangkan sebuah model penjadwalan yang bersifat *nonresumable* pada *flow shop* dua mesin dimana model ini telah mempertimbangkan kemungkinan ketidakterediaan mesin karena mengalami kerusakan maupun *preventif maintenance* untuk meminimasi *mean earliness*.

Ras dan Halim (2002) mengembangkan model penjadwalan *batch* yang mengakomodasi kedatangan pesanan baru pada sistem manufaktur *job shop* dengan menggunakan teknik insersi, yang diusulkan oleh Sotskov *et al.* (1999), untuk meminimasi waktu tinggal aktual. Kriteria performansi ini dikembangkan oleh Halim dan Ohta (1993) untuk mengakomodasi kasus dinamis, yaitu kenyataan bahwa *demand* bisa datang setiap saat (tidak harus datang pada saat nol), dan tuntutan penyelesaian pekerjaan tepat waktu. Dengan demikian berarti saat selesai *job* tidak diperkenankan melanggar *due date* dan *job* yang akan diproses tidak perlu berada di lantai produksi secara bersamaan pada saat nol, tetapi bisa tiba di lantai produksi pada saat diperlukan.

Pada kondisi nyata, sulit sekali untuk menghasilkan jadwal yang tepat waktu karena pekerjaan datang relatif bersamaan dalam jumlah yang banyak. Untuk memenuhi hal tersebut sistem manufaktur terpaksa menjadwalkan pemrosesan sejumlah pekerjaan lebih awal sehingga mendahului *due date*. Hal ini memunculkan konflik antara minimasi ongkos simpan yang akan mengakibatkan peningkatan ongkos *setup*, dan sebaliknya. Kedatangan pesanan baru juga menimbulkan konflik penerimaan atau penolakan atas kedatangan pesanan baru tersebut karena penerimaan berarti penyesuaian terhadap jadwal yang telah dibuat sedangkan penolakan berarti kehilangan kesempatan atas pekerjaan yang ditawarkan dimana kedua keputusan ini memuat konsekuensi biaya. Halim dan Zakaria (2002) menjawab permasalahan ini dengan mengembangkan model penjadwalan yang sudah

mempertimbangkan biaya produksi, yang merupakan derivasi dari definisi waktu tinggal aktual untuk sistem manufaktur tanpa *tardy jobs*, sebagai ukuran performansi. Biaya produksi yang dipertimbangkan yaitu biaya *set up*, biaya simpan, dan biaya penolakan pekerjaan.

Penelitian mengenai penjadwalan *batch* dinamis pada *flow shop* sebenarnya telah dilakukan oleh Halim dan Barnali (1998) namun belum melibatkan biaya produksi sebagai pertimbangan dalam pengambilan keputusan. Oleh karena itu penelitian ini mencoba untuk mengembangkan sebuah model penjadwalan *batch* pada *flow shop* dengan mempertimbangkan kemungkinan kedatangan pekerjaan (*job*) baru untuk meminimasi biaya produksi.

4.2 Posisi Penelitian

Beberapa penelitian yang sudah dilakukan dan dijadikan sebagai dasar pengembangan model pada penelitian ini adalah :

1. Halim and Ohta (1996) yang mengembangkan model penjadwalan *batch* dinamis pada *flow shop* dengan mempertimbangkan konsep tepat waktu dimana pekerjaan tidak harus datang seluruhnya secara serentak di awal produksi serta tidak perlu dilakukan pengiriman produk setiap saat kepada konsumen melainkan dapat dilakukan bersamaan pada saat *due date*.

2. Halim dan Barnali (1998) yang mengembangkan model penjadwalan *batch* dinamis pada *flow shop* untuk kasus *multi-item* dengan kriteria untuk meminimasi waktu tinggal aktual.
3. Halim dan Zakaria (2002) yang mengembangkan model penjadwalan *batch* pada *job shop* dinamis dimana kriteria yang ingin dicapai adalah minimasi biaya produksi, yang merupakan derivasi dari definisi waktu tinggal aktual untuk sistem manufaktur tanpa *tardy jobs*, dan biaya penolakan pekerjaan.
4. Aisyati, et.al (2007) yang mengembangkan model penjadwalan *batch* pada *flow shop* dinamis untuk meminimasi rata-rata keterlambatan penyelesaian *order* (*mean tardiness*) dimana peneliti sudah mempertimbangkan kedatangan *order* baru berupa pengembangan algoritma penjadwalan ulang yang sudah dilengkapi dengan keputusan penolakan atau penerimaan *order* baru berdasarkan fisibilitas pemenuhan *due date*.

Posisi penelitian yang dilakukan dibandingkan penelitian sebelumnya dapat dilihat pada Tabel 4.1. sebagai berikut.

Tabel 4.1 Posisi Penelitian

	Halim dan Ohta (1996)	Halim dan Barnali (1998)	Halim dan Zakaria (2002)	Aisyati, et.al (2007)	Penelitian Usulan
---	-----------------------	--------------------------	--------------------------	-----------------------	-------------------

Sistem Manufaktur	FS	FS	JS	FS	FS
Due date	Common	Common	Common	Multi	Common
Earliness	Infeasible	Infeasible	Feasible	Feasible	Feasible
Tardiness	Infeasible	Infeasible	Infeasible	Feasible	Infeasible
Disturbance	Tidak	Tidak	Ya	Ya	Ya
Biaya Produksi	Tidak	Tidak	Ya	Tidak	Ya
Biaya Penolakan Pekerjaan	Tidak	Tidak	Ya	Tidak	Ya
Kriteria Performansi	TAFT	TAFT	TC	MT	TC

Keterangan : FS: *Flow Shop*, JS: *Job Shop*, TAFT : *Total Actual Flow Time*, TC : *Total Cost*, MT : *Mean Tardiness*

Pada Tabel 4.1 dapat dilihat bahwa penelitian yang dilakukan mengakomodasi beberapa hal yang belum dipertimbangkan dalam penelitian sebelumnya. Pada penelitian Ohta (1996) dan Barnali (1998) penjadwalan dilakukan menggunakan pendekatan mundur sehingga semua *batch* harus memenuhi *due date* secara tepat waktu, namun hal ini akan menyebabkan semua pekerjaan yang melanggar saat mulai ($T = 0$) menjadi tidak fisibel atau ditolak pengerjaannya. Padahal kenyataannya sulit sekali mengatur agar semua pekerjaan selesai tepat waktu terutama apabila semua pekerjaan tersebut berada pada *due date* yang ketat.

Penelitian yang dilakukan oleh Halim dan Zakaria (2002) sudah mempertimbangkan faktor toleransi untuk pekerjaan yang tidak diselesaikan tepat pada saat *due date* dimana model yang dikembangkan hanya mengizinkan pekerjaan diselesaikan lebih awal daripada *due date* dan tidak boleh ada pekerjaan yang terlambat. Penelitian ini sudah melibatkan perhitungan biaya

produksi di dalam model yang dikembangkan dimana keputusan untuk menerima atau menolak suatu pekerjaan dilakukan berdasarkan perbandingan antara biaya pengerjaan produk dengan resiko kerugian apabila pekerjaan tersebut ditolak pengerjaannya.

Pada bab berikutnya akan dibahas langkah-langkah membangun sebuah model penjadwalan *batch* pada *flow shop* dinamis. Model tersebut memberikan kemungkinan bagi suatu pekerjaan untuk selesai mendahului *due date* dengan mempertimbangkan resiko terhadap biaya simpan dan pengerjaan *batch* tidak boleh melampaui *due date*.

Kriteria performansi yang digunakan sama dengan Halim dan Zakaria (2002) namun model yang dikembangkan berlaku untuk sistem manufaktur *flow shop*, berbeda dengan Halim dan Zakaria yang mengembangkan model untuk sistem manufaktur *job shop*.

Pengembangan model juga dilakukan berdasarkan penelitian Aisyati, *et.al* (2007) dengan menggunakan kriteria yang berbeda yaitu total biaya produksi. Ketiga penelitian yaitu Halim dan Zakaria(2002), Aisyati, *et.al* (2007), dan penelitian yang akan dilakukan lebih realistis karena mempertimbangkan faktor gangguan (*disturbance*) berupa kedatangan *order* baru.



MEMBANGUN MODEL MATEMATIS

Pada tahap pengembangan model matematis terlebih dahulu dilakukan identifikasi faktor-faktor yang relevan dengan masalah yang dihadapi sesuai dengan tujuan

yang ingin dicapai, serta hubungan antar faktor kemudian diformulasikan menjadi sebuah model.

Jika dilihat pada penelitian yang penulis lakukan tujuan yang ingin dicapai adalah menghasilkan model penjadwalan *batch* pada *flow shop m* mesin untuk meminimasi total biaya produksi.

Formulasi model yang dikembangkan dinyatakan dalam formulasi *Integer Non-Linear Programming* (INLP) dimana formulasi INLP ini sesuai karena pada permasalahan terdapat beberapa variabel keputusan yang dikehendaki bernilai integer, yaitu jumlah dan ukuran *batch*. Selain itu dalam formulasi model juga diperlukan penggunaan variabel biner, yaitu variabel integer yang memiliki nilai 0 dan 1, untuk menyelesaikan permasalahan berupa penentuan jenis *item* yang dimuat dalam suatu *batch* serta penentuan *item* baru yang akan ditolak atau diterima pengerjaannya disebabkan karena biaya pengerjaan yang akan ditanamkan lebih berkontribusi terhadap peningkatan total biaya produksi dibandingkan dengan biaya penolakan pekerjaan dari *item* tersebut apabila ditolak dan sebaliknya apabila diterima.

Dengan demikian, pada formulasi model digunakan beberapa variabel yang memiliki nilai-nilai integer yaitu jumlah *batch* (N), ukuran *batch* (Q_i), variabel biner yang menyatakan eksistensi suatu jenis *item* pada suatu *batch* ($r_{i,g}$), dan variabel biner penolakan pekerjaan (X_g).

Sub bab berikut akan menjelaskan tahapan pengembangan model matematis berkaitan dengan

permasalahan penjadwalan *batch* yang akan dilakukan. Pengembangan model dilakukan secara bertahap yaitu dimulai dari pengembangan model pada sistem statis dan selanjutnya pengembangan model pada sistem dinamis.

5.1 Model Statis

Model statis pada kasus ini merupakan model yang dikembangkan untuk menyelesaikan permasalahan penjadwalan *batch* yang dikirim pada *common due date* dan diproses pada sistem manufaktur *flow shop* tanpa mempertimbangkan penolakan pekerjaan.

Permasalahan ini dapat digambarkan sebagai berikut. Sejumlah produk dengan g jenis yang berbeda ($g = 1, 2, \dots, G$) dan memiliki kuantitasnya masing-masing (n_g) akan diproses pada sistem manufaktur *flow shop* n tahap dimana setiap tahap terdiri atas mesin tunggal ($m = 1, 2, \dots, M$). pengerjaan *job* dimulai di Tahap 1 dan berakhir di Tahap M (jumlah tahapan sama dengan jumlah mesin). Pada setiap tahapan mesin hanya dapat mengerjakan satu operasi dan setiap operasi hanya diperkenankan untuk dikerjakan pada satu mesin. Proses pengerjaan dilakukan dengan mengelompokkan pekerjaan ke dalam *batch* dengan suatu ukuran tertentu. Waktu *setup* setiap *batch* yang dikerjakan pada mesin yang sama diasumsikan sama dan batas penyerahan seluruh *order* dilakukan pada saat yang bersamaan (*common due date*).

Berdasarkan gambaran permasalahan ini kemudian dibentuk formulasi model matematis dimana model ini selanjutnya disebut sebagai Model BFSS (*Batch Flow Shop Dinamis*).

Bagaimana membentuk sebuah model matematis ?

5.1.1 Kriteria Kinerja

Sebelum menyusun sebuah model matematis perlu didefinisikan terlebih dahulu kriteria kinerja yang ingin dicapai. Kriteria kinerja yang dirumuskan menjadi fungsi tujuan dalam permasalahan ini adalah minimasi total biaya produksi. Sebagai basis pengembangan model dinamis maka model yang dikembangkan adalah model penjadwalan *batch* yang diproduksi pada *flow shop* statis yang terdiri atas *multi item* dan dikirim pada *due date* yang bersamaan (*common due date*).

Nilai variabel keputusan yang akan dicari untuk meminimumkan kriteria kinerja yang telah ditetapkan adalah jumlah *batch* (N), ukuran *batch* (Q_i), urutan pemrosesan *batch* yang ditandai dengan saat mulai ($B_{i,m}$) dan saat selesai ($F_{i,m}$). Pada model ini terdapat variabel biner yang digunakan yaitu variabel biner yang menyatakan eksistensi suatu jenis *item* pada suatu *batch* ($r_{i,g}$). Variabel dependen dalam hal ini adalah total biaya produksi (TC).

5.1.2 Parameter

Setelah menetapkan kriteria kinerja maka langkah selanjutnya adalah mendefinisikan dan simbolisasi dari parameter dan variabel yang akan disusun ke dalam bentuk formulasi model matematis. Parameter berkaitan dengan nilai yang bisa dikontrol atau dapat diubah oleh pengambil keputusan (*scheduler*) sedangkan variabel berkaitan dengan nilai yang akan dicari (solusi).

Sesuai dengan kriteria kinerja dan variabel keputusan yang telah ditentukan, maka parameter yang digunakan di dalam model usulan ini adalah sebagai berikut :

- d : batas waktu penyerahan untuk semua *batch*
- t_{gm} : waktu proses *item* Jenis g di Mesin m
- s_m : waktu *setup* di Mesin m
- M : jumlah mesin (jumlah tahapan proses)
- G : jumlah jenis *item*
- n_g : jumlah permintaan dari *item* Jenis g
- $C_{1,m}$: ongkos *setup* pada Mesin m per satuan waktu
- $C_{2,g}$: ongkos simpan *item* Jenis g yang sedang dikerjakan per satuan waktu
- $C_{3,g}$: ongkos simpan produk jadi dari *item* Jenis g

5.1.3 Variabel Keputusan

Penting bagi seorang pengambil keputusan untuk memahami perbedaan dari parameter dan variabel model. Berikut adalah variabel keputusan dari model yang akan dibentuk formulasinya.

- $B_{i,m}$: saat mulai pengerjaan *batch* urutan ke- i pada Mesin m
- $F_{i,m}$: saat selesai pengerjaan *batch* urutan ke- i pada Mesin m
- Q_i : ukuran *batch* yang diproses pada posisi ke- i
- N : jumlah *batch*
- $r_{i,g}$: variabel biner yang menyatakan jenis *item* yang dimuat dalam *batch* urutan ke- i
- TC : total biaya produksi

5.1.4 Formulasi Model

Setelah kriteria kinerja terdefinisi serta parameter dan variabel model ditetapkan maka kita siap untuk membentuk sebuah formulasi model matematis. Formulasi model dibangun dengan memperhatikan relasi antar parameter dan variabel serta hukum keberlakuan model yang dijabarkan pada asumsi.

Adapun asumsi yang digunakan dalam model yang akan dikembangkan adalah :

1. Bahan baku tersedia setiap saat dibutuhkan sehingga biaya simpan dari bahan baku diabaikan.

2. Waktu dan biaya *setup* untuk setiap *batch* pada mesin yang sama adalah sama.
3. Tidak ada *batch* yang dikerjakan pada lebih dari satu mesin pada saat yang bersamaan dan tidak ada mesin yang mengerjakan lebih dari satu *batch* sekaligus pada saat yang sama.

Formulasi model terdiri atas fungsi tujuan dan batasan (*constraint*) sebagai berikut :

Fungsi tujuan:

Minimasi

$$\begin{aligned}
 TC = & \left(N \cdot \sum_{m=1}^M s_m \cdot C_{1,m} \right) + \sum_{g=1}^G \sum_{i=1}^N r_{i,g} \left[(F_{i,m} - B_{i,1}) Q_i \cdot C_{2,g} \right] \\
 & + \sum_{g=1}^G \sum_{i=1}^N r_{i,g} \left[(d - F_{i,M}) Q_i \cdot C_{3,g} \right] \quad (1)
 \end{aligned}$$

Pembatas:

$$B_{1,1} = s_1 \quad (2)$$

$$F_{1,1} = B_{1,1} + \sum_{g=1}^G r_{1,g} (t_{g,1} \cdot Q_1) \quad (3)$$

$$B_{i,1} = F_{i-1,1} + s_1 \quad ; \quad i = 2, \dots, N \quad (4)$$

$$F_{i,1} = B_{i,1} + \sum_{g=1}^G r_{i,g} (t_{g,1} \cdot Q_i) \quad ; \quad i = 2, \dots, N \quad (5)$$

$$B_{1,m} = F_{1,m-1} \quad ; \quad m = 2, \dots, M \quad (6)$$

$$F_{1,m} = B_{1,m} + \sum_{g=1}^G r_{1,g} (t_{g,m} \cdot Q_1) \quad ; \quad m = 2, \dots, M \quad (7)$$

$$B_{i,m} = \max(F_{i,m-1}, F_{i-1,m} + s_m) \quad ; \quad i = 2, \dots, N \quad ; \quad m = 2, \dots, M \quad (8)$$

$$F_{i,m} = B_{i,m} + \sum_{g=1}^G r_{i,g} (t_{g,m} \cdot Q_i) \quad ; \quad i = 2, \dots, N \quad ; \quad m = 2, \dots, M \quad (9)$$

$$F_{N,M} \leq d \quad (10)$$

$$r_{i,g} \in \{0,1\} \quad ; \quad i = 1, \dots, N \quad ; \quad g = 1, \dots, G \quad (11)$$

$$\sum_{g=1}^G r_{i,g} = 1 \quad ; \quad i = 1, \dots, N \quad (12)$$

$$\sum_{g=1}^G r_{i,g} \cdot Q_i = n_g \quad ; \quad i = 1, \dots, N \quad (13)$$

$$N \geq G \quad (14)$$

$$Q_i \geq 1, \text{ integer} \quad (15)$$

Formulasi (1) menyatakan tujuan model yaitu meminimasi total biaya produksi. Pembatas (2) menyatakan saat mulai *batch* urutan pertama di Tahap 1. Pembatas (3) menyatakan saat selesai *batch* urutan pertama di Tahap 1. Pembatas (4) menyatakan saat mulai *batch* urutan ke-*i* di Tahap 1. Pembatas (5) menyatakan saat selesai *batch* urutan ke-*i* di Tahap 1. Pembatas (6) menyatakan saat mulai *batch* urutan pertama di Mesin *m*.

Pembatas (7) menyatakan saat selesai *batch* urutan pertama di Mesin *m*. Pembatas (8) menyatakan saat mulai *batch* urutan ke-*i* di Mesin *m*. Pembatas (9) menyatakan saat selesai *batch* urutan ke-*i* di Mesin *m*. Pembatas (10) menjamin bahwa saat selesai pengerjaan *batch* terakhir yang diproses di mesin terakhir tidak menabrak *due date*. Pembatas (11) merupakan variabel biner untuk pemilihan jenis *item* yang ditempatkan pada suatu *batch*. Pembatas (12) menjamin bahwa *batch* pada urutan ke-*i* hanya terdiri atas satu jenis *item* tertentu. Pembatas (13) menjamin bahwa jumlah *part* dari suatu jenis *item* yang diproses harus sama dengan jumlah total *part* dari jenis *item* tersebut yang harus diproduksi. Pembatas (14) menyatakan bahwa jumlah *batch* minimal sama dengan jumlah jenis *item* yang akan diproses. Pembatas (15) menjamin bahwa ukuran *batch* harus lebih besar atau sama dengan satu, serta jumlah dan ukuran *batch* bernilai integer.

5.2 Model Dinamis

Setelah model untuk kondisi statis dibangun maka sesuai dengan tujuan yang ingin dicapai yaitu menyelesaikan permasalahan penjadwalan pada sistem dinamis maka selanjutnya dikembangkan formulasi model untuk sistem dinamis tersebut. Pada tahapan pembentukan model dinamis langkah-langkah yang dilakukan sama dengan pembentukan model statis pada sub bab sebelumnya disertai dengan beberapa penyesuaian.

Pada perumusan kriteria kinerja perbedaan mendasar adalah pada perubahan sistem statis menjadi dinamis sehingga kriteria kinerja yang menjadi fungsi tujuan dalam permasalahan ini adalah minimasi total biaya produksi pada model penjadwalan *batch* yang diproduksi pada *flow shop* dinamis yang terdiri atas *multi item* dan dikirim pada *due date* yang berbeda (*multi due date*).

Selanjutnya juga perlu dilakukan penambahan parameter dan variabel berkaitan dengan kondisi dinamis. Adapun parameter dan variabel yang ditambahkan adalah :

1. Parameter

- A_m : availabilitas ketersediaan kapasitas pada Mesin- m
 $C_{4,g}$: ongkos penolakan pekerjaan *item* Jenis g
 d_g : *due date item* Jenis g

2. Variabel

- X_g : variabel biner penolakan pekerjaan *item* Jenis g

Adapun asumsi tambahan yang digunakan pada sistem dinamis adalah kedatangan *job* baru tidak bisa menginterupsi *batch* yang sedang dikerjakan atau dengan kata lain *job* baru tersebut harus menunggu (*waiting*) hingga status mesin menjadi *idle*. Selain itu juga asumsi penyelesaian pekerjaan pada batas yang seragam (*common due date*) digugurkan dengan mengembangkan model menggunakan asumsi baru yaitu masing-masing

item dikirim pada *due date*-nya masing-masing (*multi due date*).

Dengan demikian model penjadwalan *batch* yang mempertimbangkan penolakan pekerjaan terhadap kedatangan order baru, selanjutnya disebut sebagai Model BFSD, dikembangkan sebagai berikut:

Fungsi tujuan:

Minimasi

$$TC = \left(N \cdot \sum_{m=1}^M s_m \cdot C_{1,m} \right) + \sum_{g=1}^G \sum_{i=1}^N r_{i,g} \left[(F_{i,m} - B_{i,1}) Q_i \cdot C_{2,g} \right] \\ + \sum_{g=1}^G \sum_{i=1}^N r_{i,g} \left[(d_g - F_{i,M}) Q_i \cdot C_{3,g} \right] + \sum_{g=1}^G X_g \cdot C_{4,g} \quad (16)$$

Pembatas:

$$B_{1,1} = A_1 + s_1 \quad (17)$$

$$F_{1,1} = B_{1,1} + \sum_{g=1}^G r_{1,g} (t_{g,1} \cdot Q_1) \quad (18)$$

$$B_{i,1} = F_{i-1,1} + s_1 \quad ; i = 2, \dots, N \quad (19)$$

$$F_{i,1} = B_{i,1} + \sum_{g=1}^G r_{i,g} (t_{g,1} \cdot Q_i) \quad ; i = 2, \dots, N \quad (20)$$

$$B_{1,m} = \max(F_{1,m-1}, A_m) \quad ; m = 2, \dots, M \quad (21)$$

$$F_{1,m} = B_{1,m} + \sum_{g=1}^G r_{1,g} (t_{g,m} \cdot Q_1) \quad ; m = 2, \dots, M \quad (22)$$

$$B_{i,m} = \max(F_{i,m-1}, F_{i-1,m} + s_m) \quad ; i = 2, \dots, N \quad ; m = 2, \dots, M \quad (23)$$

$$F_{i,m} = B_{i,m} + \sum_{g=1}^G r_{i,g} (t_{g,m} \cdot Q_i) ; i = 2, \dots, N ; m = 2, \dots, M \quad (24)$$

$$F_{N,M} \leq d_g \quad (25)$$

$$r_{i,g} \in \{0,1\} ; i = 1, \dots, N ; g = 1, \dots, G \quad (26)$$

$$\sum_{g=1}^G r_{i,g} = 1 ; i = 1, \dots, N \quad (27)$$

$$\sum_{g=1}^G r_{i,g} \cdot Q_i = n_g ; i = 1, \dots, N \quad (28)$$

$$X_g \begin{cases} 1, & \text{jika } \sum_{i=1}^N r_{i,g} = 0 ; g = 1, \dots, G \\ 0, & \text{jika lainnya} \end{cases} \quad (29)$$

$$N \geq G \quad (30)$$

$$Q_i \geq 1, \text{ integer} \quad (31)$$

Formulasi (16) menyatakan tujuan model yaitu meminimasi total biaya produksi. Pembatas (17) menyatakan saat mulai *batch* urutan pertama di Tahap 1. Pembatas (18) menyatakan saat selesai *batch* urutan pertama di Tahap 1. Pembatas (19) menyatakan saat mulai *batch* urutan ke-*i* di Tahap 1. Pembatas (20) menyatakan saat selesai *batch* urutan ke-*i* di Tahap 1. Pembatas (21) menyatakan saat mulai *batch* urutan pertama di Mesin *m*. Pembatas (22) menyatakan saat selesai *batch* urutan pertama di Mesin *m*. Pembatas (23) menyatakan saat mulai *batch* urutan ke-*i* di Mesin *m*. Pembatas (24) menyatakan saat selesai *batch* urutan ke-*i* di Mesin *m*. Pembatas (25) menjamin bahwa saat selesai pengerjaan *batch* terakhir yang diproses di mesin terakhir tidak menabrak *due date*. Pembatas (26) merupakan variabel biner untuk pemilihan jenis *item* yang ditempatkan pada suatu *batch*. Pembatas (27) menjamin bahwa *batch* pada

urutan ke- i hanya terdiri atas satu jenis *item* tertentu. Pembatas (28) menjamin bahwa jumlah *part* dari suatu jenis *item* yang diproses harus sama dengan jumlah total *part* dari jenis *item* tersebut yang harus diproduksi. Pembatas (29) merupakan variabel biner untuk menentukan penerimaan atau penolakan pekerjaan baru. Pembatas (30) menyatakan bahwa jumlah *batch* minimal sama dengan jumlah jenis *item* yang akan diproses. Pembatas (31) menjamin bahwa ukuran *batch* harus lebih besar atau sama dengan satu, serta jumlah dan ukuran *batch* bernilai integer.

Apabila model matematis sudah diperoleh, maka langkah selanjutnya adalah mengembangkan algoritma pencarian solusi untuk mengarahkan kepada penggunaan dari formulasi model matematis yang telah dibangun. Hal ini akan dibahas lebih lanjut pada bab berikutnya.

Diharapkan hingga tahap ini pembaca sudah lebih memahami dan mulai mencoba untuk mengembangkan model-model penjadwalan secara matematis berdasarkan pengembangan dari model yang sudah ada maupun kondisi *real* pada suatu sistem manufaktur.



ALGORITMA BATCHING

Untuk menyelesaikan permasalahan menggunakan model matematis yang telah dibangun, perlu disusun sebuah algoritma pencarian solusi untuk mengarahkan pengguna (*user*) dalam mengaplikasikan model matematis tersebut. Kompleksitas permasalahan kemudian disederhanakan dengan mengubah variabel jumlah *batch* (N) menjadi parameter yang nilainya bisa diubah-ubah. Nilai-nilai ini

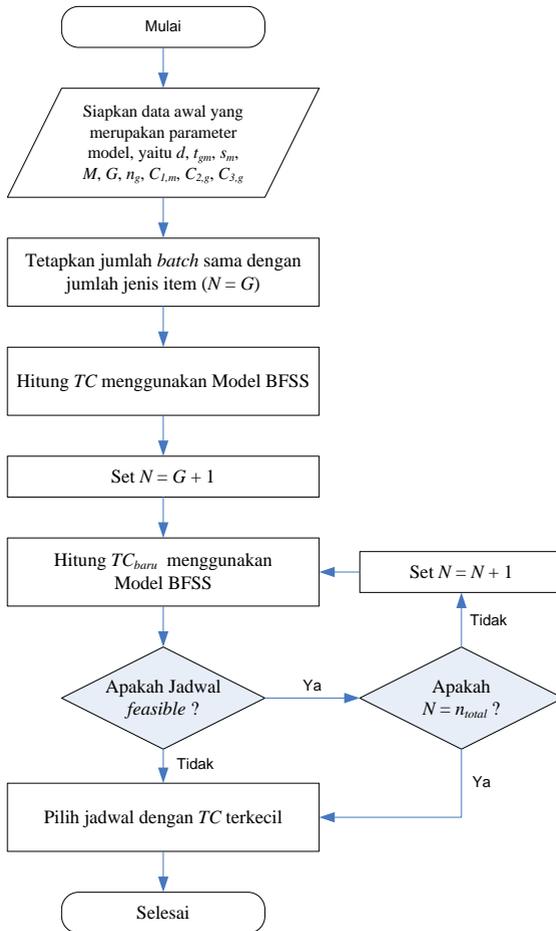
kemudian dicobakan dimulai dari jumlah *batch* sama dengan jumlah jenis *item* ($N = G$) hingga jumlah *batch* dipecah secara maksimal yaitu sejumlah total pesanan ($N = n_{total}$).

6.1 Algoritma Statis (Algoritma BFSS)

Algoritma statis dikembangkan untuk menyelesaikan permasalahan penjadwalan *batch* dalam kondisi tidak ada gangguan berupa kedatangan pekerjaan baru. Algoritma pencarian solusi mengikuti aturan sebagai berikut :

- Langkah 1 : Siapkan data awal yang merupakan parameter model, yaitu $d_g, t_{gm}, s_m, M, G, n_g, C_{1,m}, C_{2,g}, C_{3,g}$.
- Langkah 2 : Tetapkan jumlah *batch* sama dengan jumlah jenis item ($N = G$).
- Langkah 3 : Selesaikan permasalahan menggunakan Model BFSS.
- Langkah 4 : Lakukan pemecahan *batch* menggunakan Sub Algoritma Pemecahan *Batch*.
- Langkah 5 : Tetapkan jadwal yang menghasilkan total biaya produksi terkecil sebagai jadwal terpilih.
- Langkah 6 : Selesai.

Alur Algoritma BFSS adalah sebagai berikut:



Gambar 6.1 Flow Chart Algoritma BFSS

6.2 Algoritma Dinamis (Algoritma BFSD)

Algoritma dinamis digunakan untuk menyelesaikan permasalahan penjadwalan pekerjaan baru yang kedatangannya pada saat proses produksi sedang berlangsung. Secara umum algoritma pencarian solusi ini disusun sebagai berikut :

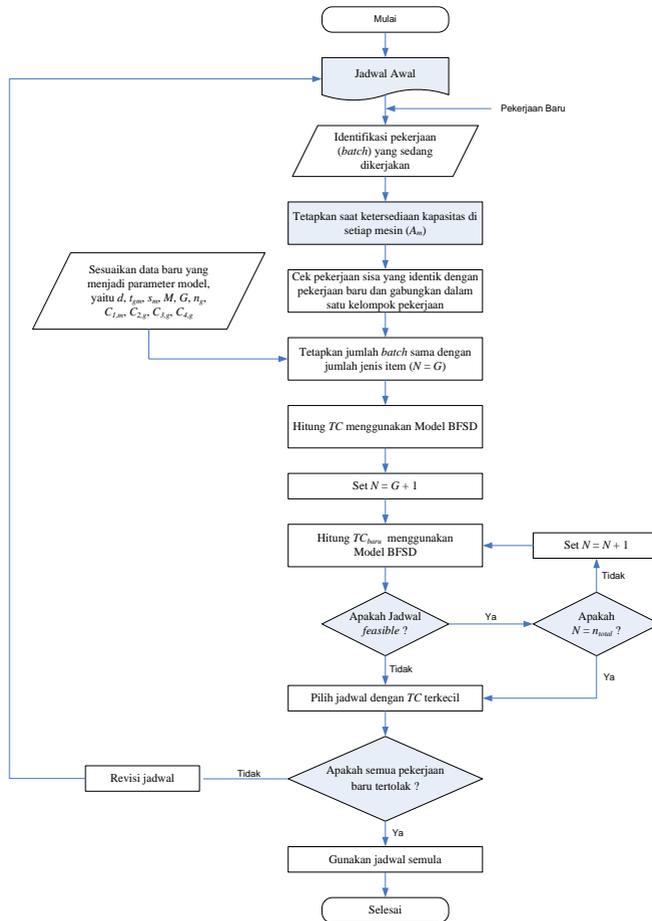
1. Pengecekan ketersediaan kapasitas dengan menghitung pekerjaan sisa yang sedang dikerjakan dan menentukan saat mulai yang paling memungkinkan untuk melakukan penjadwalan ulang.
2. Mengelompokkan pekerjaan baru yang sejenis dengan pekerjaan lama dan menghitung total jumlah produksi dari masing-masing jenis *item* yang akan dikerjakan.
3. Melakukan penjadwalan ulang.

Adapun secara rinci tahap demi tahap algoritma pencarian solusi penjadwalan *batch* pada *flow shop* dinamis adalah sebagai berikut :

- Langkah 1 : Identifikasi pekerjaan (*batch*) yang belum selesai diproses.
- Langkah 2 : Tetapkan saat selesai pengerjaan *batch* yang sedang dikerjakan di setiap mesin sebagai availabilitas kapasitas mesin (A_m).
- Langkah 3 : Cek pekerjaan sisa yang identik dengan pekerjaan baru dan gabungkan dalam satu kelompok pekerjaan.

- Langkah 4 : Siapkan data baru (data gabungan dari sisa pekerjaan lama dan pekerjaan baru) yang merupakan parameter model, yaitu $d_g, t_{gm}, s_m, M, G, n_g, C_{1,m}, C_{2,g}, C_{3,g}, C_{4,g}$.
- Langkah 5 : Tetapkan jumlah *batch* sama dengan jumlah jenis item ($N = G$).
- Langkah 6 : Selesaikan permasalahan menggunakan Model BFSD.
- Langkah 7 : Lakukan pemecahan *batch* menggunakan Sub Algoritma Pemecahan *Batch*.
- Langkah 8 : Tetapkan jadwal yang menghasilkan total biaya produksi terkecil sebagai jadwal terpilih.
- Langkah 9 : Periksa kelayakan pekerjaan. Jika seluruh pekerjaan baru tertolak kembali ke jadwal semula. Jika terdapat pekerjaan baru yang layak, revisi jadwal produksi berdasarkan hasil penjadwalan ulang.
- Langkah 10 : Selesai.

Alur Algoritma BFSD dapat dilihat pada *flow chart* berikut :



Gambar 6.2 Flow Chart Algoritma BFSD

6.3 Sub Algoritma Pemecahan *Batch*

Dalam setiap algoritma yang dikembangkan sebelumnya, baik Algoritma BFSS maupun Algoritma BFSD, terdapat sub algoritma yang disusun untuk mengatur proses pemecahan *batch*. Pada model yang dikembangkan, jumlah dan ukuran *batch* merupakan variabel keputusan yang akan dicari nilainya. Pada pencarian solusi awal ditetapkan jumlah *batch* sama dengan jumlah jenis *item*. Kemudian dilakukan pemecahan *batch* secara bertahap hingga dicapai jumlah *batch* sama dengan total jumlah produk.

Secara lengkap Algoritma Pemecahan *Batch* disusun sebagai berikut :

Langkah 1 : Set $N = G + 1$.

Langkah 2 : Selesaikan permasalahan menggunakan formulasi matematis.

Langkah 3 : Catat total biaya produksi yang dihasilkan.

Langkah 4 : Apakah saat penyelesaian *batch* terakhir lebih besar dari *due date*? Jika ya hentikan pencarian solusi, lanjut ke Langkah 7. Jika tidak lanjut ke Langkah 5.

Langkah 5 : Apakah $N = \sum_{g=1}^G n_g$? Jika ya lanjut ke

Langkah 7. Jika tidak lanjut ke Langkah 6.

Langkah 6 : Set $N = N + 1$. Kembali ke Langkah 2.

Langkah 7 : Pilih jadwal dengan total biaya produksi paling kecil.

Langkah 8 : Selesai.

Algoritma pencarian solusi seperti yang dicontohkan di atas dapat dikembangkan sesuai dengan pendekatan yang dipilih untuk dilakukan. Pada kondisi yang berbeda maka algoritma yang dikembangkan juga akan berbeda pula. Misalnya penjadwalan dinamis untuk kasus ketidaktersediaan mesin. Jika pencarian solusi dilakukan dengan menggunakan teknik sisipan maka algoritma pencarian solusi juga mengarah pada pendekatan yang digunakan tersebut.

Untuk melatih keterampilan pembaca dalam membangun sebuah algoritma, coba kembangkan algoritma pencarian solusi berdasarkan model yang sudah diperoleh pada tahap sebelum ini. Kemudian evaluasi kembali efektivitas dan efisiensi dari algoritma yang dibangun tersebut.



STUDI KASUS

Pada bab ini akan diperlihatkan penggunaan model dan algoritma pencarian solusi yang telah dikembangkan. Agar pembaca lebih mudah untuk memahami penggunaan dari masing-masing model maka persoalan akan dibedakan menurut sistem statis dan sistem dinamis.

7.1 Persoalan 1

Terdapat 2 jenis produk yang akan diproduksi, yaitu produk A dan B. Produk A akan diproduksi sebanyak

6 unit dan produk B sebanyak 7 unit. Kedua produk akan diproses pada sistem manufaktur *flow shop* 2 tahap dimana masing-masing tahap terdiri atas mesin tunggal. *Setup* dilakukan pada masing-masing mesin dengan waktu masing-masing adalah 2 menit dan 4 menit. Kedua jenis produk harus selesai pada *due date* yang sama yaitu 200 menit. Data waktu proses dari kedua jenis produk dan Biaya *setup* pada setiap mesin disajikan pada Tabel 7.1. Sedangkan data biaya *WIP* dan biaya simpan disajikan pada Tabel 7.2.

Tabel 7.1 Data Waktu Proses dan Biaya Setup Kasus Statis

Mesin	Waktu Proses		Waktu Setup	Biaya Setup
	Produk A	Produk B		
1	8	6	2	30
2	10	8	4	35

Tabel 7.2 Data Biaya WIP dan Biaya Simpan Kasus Statis

Jenis Biaya	Jenis Produk	
	Produk A	Produk B
Biaya WIP	60	75
Biaya Simpan	120	150

Penyelesaian:

Proses pencarian solusi adalah sebagai berikut:

Langkah 1 : Siapkan data awal yang merupakan parameter model, yaitu $d = 200$, $t_{1,1} = 8$, $t_{1,2} = 10$, $t_{2,1} = 6$, $t_{2,2} = 8$, $s_1 = 2$, $s_2 = 4$, $M = 2$, $G = 2$, $n_1 = 6$, $n_2 = 7$, $C_{1,1} = 30$, $C_{1,2} = 35$, $C_{2,1} = 60$, $C_{2,2} = 75$, $C_{3,1} = 120$, $C_{3,2} = 150$

Langkah 2 : Tetapkan jumlah *batch* sama dengan jumlah jenis item ($N = G$). Pada kasus ini Terdapat 2 jenis produk, sehingga $N = 2$.

Langkah 3 : Selesaikan permasalahan menggunakan Model BFSS. Total biaya yang dihasilkan adalah Rp. 197.530,- dengan jadwal yang diperoleh adalah sebagai berikut:

Tabel 7.3 Jadwal untuk $N = 2$ (Kasus Statis)

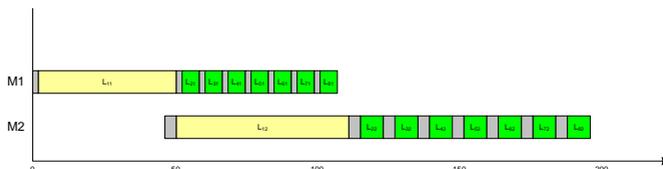
<i>Batch</i> ke-	Ukuran <i>Batch</i>	Jenis Produk	Mesin	Saat Mulai (B_{im})	Saat Selesai (F_{im})
1	6	A	1	2	50
			2	50	110
2	7	B	1	52	94
			2	114	170

Langkah 4 : Lakukan pemecahan *batch* menggunakan Sub Algoritma Pemecahan *Batch*. Solusi yang diperoleh adalah sebagai berikut :

Tabel 7.4 Total Biaya Produksi Setiap Batch (Kasus Statis)

Jumlah Batch (N)	Total Biaya Produksi (TC)	Jumlah Batch (N)	Total Biaya Produksi (TC)
3	195.930	9	196.800
4	194.630	10	200.690
5	193.630	11	204.100
6	192.930	12	207.030
7	192.530	13	209.480
8	192.430		

Langkah 5 : Tetapkan jadwal yang menghasilkan total biaya produksi terkecil sebagai jadwal terpilih. Berdasarkan Tabel 7.4 diperoleh Total Biaya Produksi terendah pada saat batch dipecah menjadi 8 batch dengan total biaya sebesar Rp. 192.430,-. Gantt chart dari hasil penjadwalan akhir dapat dilihat pada Gambar 7.1.



Gambar 7.1 Gantt Chart Hasil Penjadwalan Kasus Statis

Secara rinci hasil yang diperoleh adalah :

Tabel 7.5 Solusi Optimal untuk Kasus Statis

<i>Batch ke-</i>	<i>Ukuran Batch</i>	<i>Jenis Produk</i>	<i>Mesin</i>	<i>Saat Mulai (B_{im})</i>	<i>Saat Selesai (F_{im})</i>
1	6	A	1	2	50
			2	50	110
2	1	B	1	52	58
			2	114	122
3	1	B	1	60	66
			2	126	134
4	1	B	1	68	74
			2	138	146
5	1	B	1	76	82
			2	150	158
6	1	B	1	84	90
			2	162	170
7	1	B	1	92	98
			2	174	182
8	1	B	1	100	106
			2	186	194

Langkah 6 : Selesai.

7.2 Persoalan 2

Berdasarkan pada contoh kasus statis di atas, jika pada menit ke 155 terjadi kedatangan *order* baru berupa pekerjaan *item* jenis C sebanyak 8 unit dengan data *order* sebagai berikut:

Tabel 7.6 Data Waktu Proses Produk Baru

Mesin	Waktu Proses Produk C
1	3
2	8

Tabel 7.7 Data Biaya WIP dan Biaya Simpan Kasus Dinamis

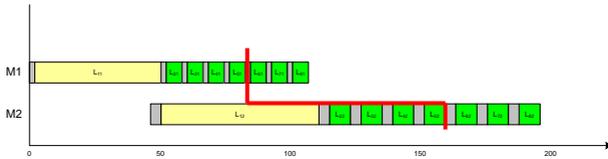
Jenis Biaya	Produk C
Biaya WIP	90
Biaya Simpan	120

Adapun batas pengerjaan dari pekerjaan baru adalah pada menit ke 300 dan biaya penolakan pekerjaan baru sebesar Rp. 1.500.000,-

Penyelesaian:

Proses pencarian solusi adalah sebagai berikut:

Langkah 1 : Identifikasi pekerjaan (*batch*) yang belum selesai diproses. Berdasarkan jadwal terpilih pada Tabel 7.5 dapat dilihat bahwa pada menit ke 155 sedang dilaksanakan pemrosesan *batch* ke 5 di mesin 2 dan harus diselesaikan pengerjaannya. Dengan demikian terdapat 3 *batch* yang belum selesai diproses.



Gambar 7.2 Gantt Chart Posisi Batch Sisa

Langkah 2 : Tetapkan saat selesai pengerjaan *batch* yang sedang dikerjakan di setiap mesin sebagai availabilitas kapasitas mesin (A_m). Jika dilihat pada jadwal, pengerjaan *batch* ke 5 yang belum selesai terdapat pada mesin 2. Pengerjaan tersebut tidak bisa diinterupsi (harus menunggu hingga pengerjaan *batch* ke 5 selesai) sehingga saat paling memungkinkan untuk memulai pengerjaan pekerjaan sisa dan pekerjaan baru pada masing-masing mesin adalah sebagai berikut:

Tabel 7.8 Availabilitas Mesin

Mesin ke-	Saat Mulai (A_m)
1	84
2	162

Langkah 3 : Cek pekerjaan sisa yang identik dengan pekerjaan baru dan gabungkan dalam satu kelompok pekerjaan. Dalam hal ini tidak ada pekerjaan baru yang identik dengan pekerjaan lama.

Langkah 4 : Siapkan data baru (data gabungan dari sisa pekerjaan lama dan pekerjaan baru) yang menjadi parameter model, yaitu

$d_{lama}=200, d_{baru}=300, t_{1,1} = 6, t_{1,2} = 8,$
 $t_{2,1} = 3, t_{2,2} = 8, s_1 = 2, s_2 = 4, M = 2,$
 $G = 2, n_1 = 3, n_2 = 8, C_{1,1} = 30, C_{1,2} = 35,$
 $C_{2,1} = 75, C_{2,2} = 90, C_{3,1} = 150, C_{3,2} = 120$

Langkah 5 : Tetapkan jumlah *batch* sama dengan jumlah jenis item ($N = G$). Pada kasus ini Terdapat 2 jenis produk, sehingga $N = 2$.

Langkah 6 : Selesaikan permasalahan menggunakan Model BFS. Total biaya yang dihasilkan adalah Rp. 181.810,- dengan jadwal yang diperoleh adalah sebagai berikut:

Tabel 7.9 Jadwal untuk $N = 2$ (Kasus Dinamis)

Batch ke-	Ukuran Batch	Jenis Produk	Mesin	Saat Mulai (B_{im})	Saat Selesai (F_{im})
1	3	B	1	84	102
			2	162	186
2	8	C	1	104	128
			2	190	254

Langkah 7 : Lakukan pemecahan *batch* menggunakan Sub Algoritma Pemecahan *Batch*. Solusi yang diperoleh adalah sebagai berikut :

Tabel 7.10 Total Biaya Produksi Setiap *Batch* (Kasus Dinamis)

Jumlah <i>Batch</i> (N)	Total Biaya Produksi (TC)
3	178.515
4	176.510
5	174.400
6	172.620
7	171.170

8	170.050
9	169.260
10	168.800
11	168.670

Langkah 8 : Tetapkan jadwal yang menghasilkan total biaya produksi terkecil sebagai jadwal terpilih. Berdasarkan Tabel 7.10 diperoleh Total Biaya Produksi terendah pada saat *batch* dipecah menjadi 11 *batch* dengan jadwal sebagai berikut:

Tabel 7.11 Solusi Optimal untuk Kasus Dinamis

Batch ke-	Ukuran Batch	Jenis Produk	Mesin	Saat Mulai (B_{im})	Saat Selesai (F_{im})
1	1	B	1	84	90
			2	162	170
2	1	B	1	92	98
			2	174	182
3	1	B	1	100	106
			2	186	194

4	1	C	1	108	111
			2	198	206
5	1	C	1	113	116
			2	210	218
6	1	C	1	118	121
			2	222	230
7	1	C	1	123	126
			2	234	242
8	1	C	1	128	131
			2	246	254
9	1	C	1	133	136
			2	258	266
10	1	C	1	138	141
			2	270	278
11	1	C	1	143	146
			2	282	290

Langkah 9 : Periksa kelayakan pekerjaan. Jika seluruh pekerjaan baru tertolak kembali ke jadwal semula. Jika terdapat pekerjaan baru yang layak, revisi jadwal produksi berdasarkan hasil penjadwalan ulang. Dalam kasus ini tidak ada pekerjaan yang tertolak sehingga jadwal yang digunakan adalah jadwal revisi.



Gambar 7.3 Gantt Chart Jadwal Revisi

Langkah 10 : Selesai.



VALIDITAS

8.1 Rancangan Pengujian Model

Untuk mengetahui performansi model yang dikembangkan, dilakukan pengujian model dengan menggunakan beberapa skenario. Bila ditinjau dari parameter yang ada, terdapat beragam variasi nilai yang dapat dibangkitkan. Parameter yang dimaksud adalah jumlah jenis *item*, kuantitas masing-masing jenis *item*, waktu *setup*, waktu proses, *due date*, dan biaya-biaya yang terlibat.

Pada pengujian model, kuantitas masing-masing jenis *item* ditetapkan 10 dan 15 unit. Nilai ini dipilih untuk mempercepat pencarian solusi karena pemecahan *batch* maksimal, yaitu pada saat setiap *batch* hanya memuat satu unit *item*, adalah pada saat $N = 25$. *Due date* ditetapkan sebesar 850 satuan waktu. Adapun data-data umum lainnya adalah sebagai berikut :

Tabel 8.1 Data Hipotetik Waktu Proses dan Biaya Setup

Mesin	Waktu Proses		Waktu Setup	Biaya Setup
	Produk A	Produk B		
1	8	6	25	125
2	4	9	20	235
3	12	7	35	320

Tabel 8.2 Data Hipotetik Biaya WIP dan Biaya Simpan

Jenis Biaya	Jenis Produk	
	Produk A	Produk B
Biaya WIP	150	180
Biaya Simpan	125	230

8.1.1 Skenario 1

Skenario 1 dilakukan untuk menguji kemampuan model dalam menyelesaikan permasalahan penjadwalan pada jumlah mesin yang lebih banyak. Dalam hal ini dicobakan apabila jumlah mesin (tahapan proses) terdiri dari lima urutan proses. Data waktu proses dibangkitkan secara acak pada interval 5 sampai 15 satuan waktu. Data waktu dan biaya *setup* juga dibangkitkan secara acak.

Due date yang harus dipenuhi adalah 2000 satuan waktu. Data hipotetik yang dibangkitkan adalah sebagai berikut:

Tabel 8.3 Data Hipotetik Skenario 1

Mesin	Waktu Proses		Waktu <i>Setup</i>	Biaya <i>Setup</i>
	Produk A	Produk B		
1	14	10	43	403
2	5	15	40	144
3	7	13	31	497
4	15	10	20	480
5	13	12	26	158

8.1.2 Skenario 2

Skenario 2 dilakukan untuk menguji pengaruh peningkatan waktu *setup* dibandingkan dengan peningkatan waktu proses. Dengan demikian berarti akan dicobakan pengujian apabila diset terjadi peningkatan waktu proses dengan waktu *setup* tetap, dan sebaliknya. *Due date* yang harus dipenuhi adalah 2000 satuan waktu. Data hipotetik peningkatan waktu proses dan waktu *setup* adalah sebagai berikut :

Tabel 8.4 Data Hipotetik Skenario 2

Mesin	Waktu Proses		Waktu <i>Setup</i>
	Produk A	Produk B	
1	16	12	50
2	8	18	40
3	24	14	70

8.1.3 Skenario 3

Pada skenario 3 akan diuji pengaruh peningkatan jumlah jenis *item* terhadap waktu komputasi dibandingkan dengan pengaruh peningkatan jumlah mesin terhadap waktu komputasi (Skenario 1). Pada pengujian ini ditambahkan satu jenis *item* lagi sebanyak 5 unit dengan data-data sebagai berikut :

Tabel 8.5 Data Hipotetik Waktu Proses dan Biaya Setup Skenario 3

Mesin	Waktu Proses			Waktu Setup	Biaya Setup
	Produk A	Produk B	Produk C		
1	14	10	8	43	403
2	5	15	12	40	144
3	7	13	6	31	497

Tabel 8.6 Data Hipotetik Biaya WIP dan Biaya Simpan Skenario 3

Jenis Biaya	Jenis Produk		
	Produk A	Produk B	Produk C
Biaya WIP	150	180	200
Biaya Simpan	125	230	120

8.1.4 Skenario 4

Pada skenario 4 akan diuji pengaruh perubahan nilai-nilai biaya terhadap performansi penjadwalan. Dengan demikian akan diketahui pola pemecahan *batch* dan penentuan kuantitas dari masing-masing *batch*.

8.2 Hasil Pengujian dan Analisis

8.2.1 Skenario 1

Pengujian untuk skenario 1 menghasilkan nilai total biaya produksi (*TC*) untuk setiap *batch* sebagai berikut :

Tabel 8.7 Hasil Pengujian Skenario 1

Jumlah <i>Batch</i> (<i>N</i>)	Total Biaya Produksi (<i>TC</i>)	Jumlah <i>Batch</i> (<i>N</i>)	Total Biaya Produksi (<i>TC</i>)
2	8.130.958	12	6.789.063
3	8.079.752	13	6.760.697
4	7.730.841	14	6.738.421
5	7.793.205	15	6.722.235
6	7.435.674	16	6.712.139
7	7.298.378	17	6.708.133
8	7.167.382	18	6.710.217
9	7.229.746	19	6.718.391
10	6.929.490	20	6.732.655
11	6.823.519	21	<i>infeasible</i>

Berdasarkan Tabel 8.7 di atas dapat dilihat bahwa peningkatan jumlah *batch* menurunkan nilai total biaya produksi dimana hasil optimal diperoleh pada saat pekerjaan dikelompokkan pengerjaannya menjadi 12 *batch*. Pada saat pemecahan *batch* menjadi 13 solusi menjadi *infeasible* disebabkan karena saat selesai *batch* terakhir menabrak *due date*. Dengan demikian terbukti bahwa model yang dihasilkan mampu menyelesaikan permasalahan penjadwalan *flow shop n* mesin. Secara rinci, jadwal yang dihasilkan dari solusi optimal dapat dilihat pada Tabel 8.8.

Tabel 8.8 Solusi Optimal untuk Skenario 1

Batch ke-	Ukuran Batch	Jenis Produk	Mesin	Saat Mulai ($B_{i,m}$)	Saat Selesai ($F_{i,m}$)
1	1	A	1	43	57
			2	57	62
			3	62	69
			4	69	84
			5	84	97
2	1	A	1	100	114
			2	114	119
			3	119	126
			4	126	141
			5	141	154
3	1	A	1	157	171
			2	171	176
			3	176	183
			4	183	198
			5	198	211
4	1	A	1	214	228
			2	228	233
			3	233	240
			4	240	255
			5	255	268
5	1	A	1	271	285
			2	285	290
			3	290	297
			4	297	312
			5	312	325
6	1	A	1	328	342
			2	342	347
			3	347	354
			4	354	369
			5	369	382
7	1	A	1	385	399
			2	399	404
			3	404	411
			4	411	426
			5	426	439

Tabel 8.8 (lanjutan)

Batch ke-	Ukuran Batch	Jenis Produk	Mesin	Saat Mulai (B_{im})	Saat Selesai (F_{im})
8	1	A	1	442	456
			2	456	461
			3	461	468
			4	468	483
			5	483	496
9	1	A	1	499	513
			2	513	518
			3	518	525
			4	525	540
			5	540	553
10	1	A	1	556	570
			2	570	575
			3	575	582
			4	582	597
			5	597	610
11	1	B	1	613	623
			2	623	638
			3	638	651
			4	651	661
			5	661	673
12	1	B	1	666	676
			2	678	693
			3	693	706
			4	706	716
			5	716	728
13	1	B	1	719	729
			2	733	748
			3	748	761
			4	761	771
			5	771	783
14	14	B	1	772	782
			2	788	803
			3	803	816
			4	816	826
			5	826	838

Tabel 8.8 (lanjutan)

<i>Batch ke-</i>	<i>Ukuran Batch</i>	<i>Jenis Produk</i>	<i>Mesin</i>	<i>Saat Mulai ($B_{i,m}$)</i>	<i>Saat Selesai ($F_{i,m}$)</i>
15	1	B	1	825	835
			2	843	858
			3	858	871
			4	871	881
			5	881	893
16	1	B	1	878	888
			2	898	913
			3	913	926
			4	926	936
			5	936	948
17	9	B	1	931	1021
			2	1021	1156
			3	1156	1273
			4	1273	1363
			5	1363	1471

8.2.2 Skenario 2

Tabel 8.9 menunjukkan perbandingan hasil pengujian peningkatan nilai parameter waktu proses dengan peningkatan waktu *setup*. Pada saat waktu proses ditingkatkan menjadi dua kali lipat dan nilai waktu *setup* tidak berubah (tetap) terlihat bahwa di awal pemecahan *batch* nilai total biaya produksi lebih kecil dibandingkan dengan total biaya produksi yang dihasilkan dari pengujian peningkatan nilai waktu *setup*. Hal ini disebabkan karena peningkatan waktu proses tidak terlalu berdampak pada biaya WIP karena waktu *setup* yang kecil tidak mempengaruhi total waktu pengerjaan. Hal ini disebabkan karena *setup* untuk pemrosesan *batch*

pada suatu tahap dapat dilaksanakan secara simultan dengan pemrosesan *batch* pada tahap sebelumnya.

Sebaliknya pada saat nilai waktu *setup* ditingkatkan menjadi dua kali lipat berdampak pada waktu penyelesaian *batch* dikarenakan *batch* harus menunggu untuk diproses hingga *setup* pada mesin tahap berikutnya selesai dilaksanakan. Hal ini mengakibatkan peningkatan biaya WIP secara signifikan sehingga mempengaruhi performansi yang dihasilkan.

Pada saat pemecahan *batch* terus dilakukan sehingga mengurangi kuantitas di dalam *batch*, nilai total biaya produksi pada saat waktu proses ditingkatkan menjadi lebih besar dibandingkan dengan nilai total biaya produksi pada saat terjadi peningkatan waktu *setup*. Hal ini disebabkan karena pemecahan *batch* pada kasus peningkatan waktu proses menyebabkan penyelesaian menjadi semakin cepat sehingga meningkatkan ongkos simpan. Sedangkan akibat dari peningkatan waktu *setup*, meskipun biaya *setup* menjadi meningkat namun tidak memberikan dampak yang signifikan terhadap biaya simpan karena penyelesaian *batch* yang menjadi lebih lambat dan mendekati *due date*.

Tabel 8.9 Hasil Pengujian Skenario 2

Peningkatan Waktu Proses		Peningkatan Waktu Setup	
Jumlah Batch (N)	Total Biaya Produksi (TC)	Jumlah Batch (N)	Total Biaya Produksi (TC)
2	8.404.800	2	8.653.850
3	8.241.950	3	8.442.050
4	8.130.375	4	8.248.000
5	8.028.625	5	8.060.350
6	7.933.050	6	7.880.400
7	7.840.675	7	7.706.850
8	7.751.500	8	7.541.000
9	7.668.500	9	7.383.000
10	7.610.385	10	7.232.850
11	7.515.150	11	7.090.550
12	7.457.035	12	6.969.280
13	7.404.380	13	6.859.390
14	7.357.185	14	6.760.880
15	<i>infeasible</i>	15	6.673.750
16	<i>infeasible</i>	16	6.598.000
17	<i>infeasible</i>	17	6.533.630
18	<i>infeasible</i>	18	6.480.640
19	<i>infeasible</i>	19	6.439.030
20	<i>infeasible</i>	20	6.408.800
21	<i>infeasible</i>	21	6.389.950
22	<i>infeasible</i>	22	<i>Infeasible</i>

Dengan demikian, dapat dikatakan bahwa peningkatan nilai waktu proses memberikan dampak yang lebih nyata dibandingkan peningkatan waktu setup terhadap performansi yang dihasilkan.

8.2.3 Skenario 3

Waktu komputasi untuk memperoleh solusi optimal dapat dilihat pada Tabel 8.10 di bawah ini,

Tabel 8.10 Perbandingan Waktu Komputasi

Parameter	Waktu Komputasi (detik)
Jenis <i>Item</i>	598
Jumlah Mesin	342

Berdasarkan tabel di atas dapat dilihat bahwa peningkatan jenis *item* memberikan dampak yang lebih nyata terhadap waktu komputasi dibandingkan dengan peningkatan jumlah mesin.

8.2.4 Skenario 4

Pengujian dilakukan dengan meningkatkan nilai masing-masing biaya dua kali lipat secara bergantian. Perbandingan hasil pencarian solusi optimal dapat dilihat pada Tabel 8.11.

Tabel 8.11 Perbandingan TC Akibat Perubahan Nilai Parameter Biaya

Peningkatan	Total	Peningkatan	Total	Peningkatan	Total
-------------	-------	-------------	-------	-------------	-------

Biaya Setup	Biaya Produksi	Biaya WIP	Biaya Produksi	Biaya Simpan	Biaya Produksi
N = 2	3.452.600	N = 2	4202050	N = 2	6003550
N = 3	3.347.250	N = 3	3884205	N = 3	5902938
N = 4	3.265.975	N = 4	3671355	N = 4	5892034
N = 5	3.188.050	N = 5	3489271	N = 5	5682394
N = 6	3.114.700	N = 6	3209482	N = 6	4738203
N = 7	3.044.550	N = 7	3109283	N = 7	4603948
N = 8	2.979.050	N = 8	3092812	N = 8	4402040
N = 9	2.916.750	N = 9	3091827	N = 9	4203940
N = 10	2.859.100	N = 10	2983048	N = 10	3904719
N = 11	2.806.175	N = 11	2981029	N = 11	3802912
...

Berdasarkan Tabel 8.11 dapat dilihat bahwa parameter biaya yang paling berpengaruh terhadap total biaya produksi adalah biaya simpan, diikuti dengan biaya WIP, dan yang paling sedikit pengaruhnya terhadap total biaya produksi adalah biaya *setup*.

Contoh pengembangan skenario di atas hanyalah sebagian kecil dari kemungkinan-kemungkinan lain yang bisa dikembangkan. Misalnya untuk melihat pengaruh *due date* yang ketat terhadap solusi yang dihasilkan, atau sebaliknya pengaruh *due date* yang longgar terhadap performansi model dapat dilakukan dengan mencobakan beberapa nilai *due date*. Dengan melakukan sejumlah percobaan melalui skenario-skenario, kita dapat mengetahui parameter mana yang paling berpengaruh terhadap performansi model. Hal ini sangat berguna dilakukan, salah satunya adalah sebagai dasar untuk mengembangkan algoritma heuristik.

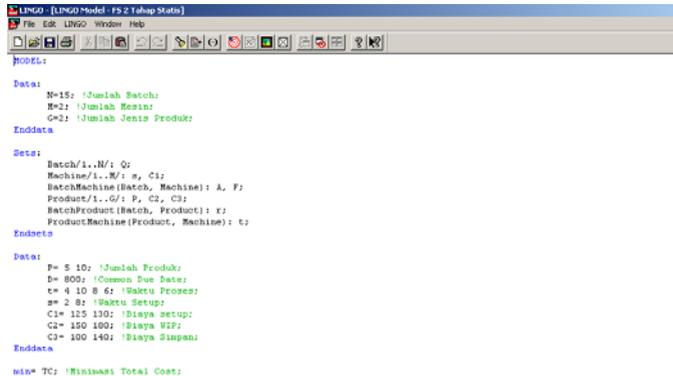


PENUTUP

Demikian teori sederhana tentang penjadwalan *batch* dan contoh penelitian pengembangan model matematis untuk permasalahan tersebut beserta langkah-langkah pencarian solusi. Buku ini memang tidak menjelaskan secara rinci tentang teori optimisasi maupun metode matematis yang sesuai karena dianggap pembaca telah menguasai hal tersebut.

Dalam penyelesaian masalah penjadwalan menggunakan pendekatan matematis ini sebaiknya didukung juga oleh

keterampilan mengaplikasikan *software* matematika maupun optimasi untuk membantu menyelesaikan permasalahan secara lebih cepat. Pada contoh kasus yang disajikan dalam buku ini misalnya, penyelesaian dilakukan dengan menggunakan bantuan *software* lingo.



```

LINGO - [LINGO Model - F5 2 Tahap Statis]
File Edit LINGO Windows Help

MODEL:

Data:
    N=15: 'Jumlah Batch';
    M=2: 'Jumlah Mesin';
    C=2: 'Jumlah Jenis Produk';
Enddata

Sets:
    Batch/1..N/: Q;
    Machine/1..M/: s, C1;
    BatchMachine(Batch, Machine): A, F;
    Product/1..C/: P, C2, C3;
    BatchProduct(Batch, Product): X;
    ProductMachine(Product, Machine): t;
Endsets

Data:
    P= 5 10: 'Jumlah Produk';
    D= 800: 'Common Due Date';
    t= 4 10 8 4: 'Waktu Proses';
    s= 2 8: 'Waktu Setup';
    C1= 125 130: 'Biaya setup';
    C2= 150 160: 'Biaya WIP';
    C3= 100 140: 'Biaya Simpan';
Enddata

min= TC: 'Minimasi Total Cost';

```

Gambar 9.1 Aplikasi Software Lingo

Pada contoh kasus, setelah pengujian dilakukan mengikuti beberapa skenario dapat ditarik kesimpulan sebagai berikut :

1. Model yang dikembangkan mampu menyelesaikan permasalahan penjadwalan *batch* pada sistem manufaktur *flowshop* dinamis yang memproduksi lebih dari satu jenis *item* dan diproses pada lebih dari 2 mesin. Penentuan urutan pemrosesan jenis *item*, *batching*, dan *sequencing* dapat diselesaikan secara simultan.
2. Model yang dikembangkan mampu mengakomodasi kedatangan pekerjaan baru di saat proses produksi

sedang berlangsung dengan keputusan penerimaan atau penolakan pekerjaan dan menghasilkan jadwal revisi yang meminimasi total biaya produksi.

3. Peningkatan waktu *setup* lebih berpengaruh terhadap penentuan jumlah *batch* yang dihasilkan serta total biaya produksi dibandingkan peningkatan waktu proses.
4. Peningkatan jumlah jenis *item* dan jumlah tahapan (mesin) memberikan dampak yang signifikan terhadap waktu komputasi. Dampak dari jumlah jenis *item* terlihat lebih nyata dibandingkan dampak penambahan jumlah mesin.
5. Parameter biaya yang paling berpotensi terhadap peningkatan total biaya produksi adalah biaya simpan. Parameter kedua adalah biaya WIP. Sedangkan biaya *setup* memiliki pengaruh yang paling kecil terhadap peningkatan total biaya produksi.
6. Penjadwalan *batch* terbukti memberikan solusi yang lebih baik dibandingkan dengan penjadwalan *job* namun meningkatkan kompleksitas permasalahan yang berpengaruh terhadap waktu komputasi.

Agar pembaca lebih memahami tentang proses perumusan model matematis untuk memecahkan permasalahan penjadwalan, khususnya penjadwalan *batch*, di bawah ini terdapat beberapa ide pengembangan model :

1. Pengembangan model yang melibatkan biaya *penalty* keterlambatan pengerjaan produk (*tardiness*).

2. Pengembangan model dengan *due date* yang berbeda (*multiple due date*) dan waktu *setup* yang berbeda terhadap masing-masing produk sehingga lebih mendekati kondisi nyata.
3. Pengembangan model dengan memasukkan faktor-faktor gangguan lainnya seperti pembatalan order atau ketidakterediaan sumber daya.
4. Pengembangan algoritma heuristik untuk efisiensi waktu komputasi.

Daftar Pustaka

- Cahya, B.I. Model Overlap Operasi (Operation Overlapping) untuk Kasus Multi-Batch dan Multi Item. Tugas Akhir. Bandung: Fakultas Teknik Industri ITB; 1998.
- Halim, A.H., Ohta, H. Batch Scheduling Problems of Multiple Items through The Flow Shop with Both Receiving and Delivery Just in Times. *International Journal of Production Research*. 1993; 31: 1943-1955.
- Halim, A.,H., Barnali, L., Model Penjadwalan Batch Dinamis pada Flow Shop untuk Kasus Multi Item dengan Kriteria Waktu Tinggal Aktual. *Jurnal TMI*. 1998; 18(1): 14-25.
- Kurniawan, D. Model Penjadwalan Batch untuk Meminimumkan Total Flow Time pada Sistem Produksi Flow Shop. Tesis Magister: Program Studi Teknik dan Manajemen Industri ITB; 2003.
- Setiawati, L., Halim, A.,H., Model Penjadwalan Flow Shop 2 Mesin dengan Interval Ketidaktersediaan untuk Meminimasi Mean Earliness. *Jurnal TMI*. 2003; 23(2): 27-39.
- Suryadhini, P.P. Model Penjadwalan Batch pada Flow Shop dengan Mesin Umum pada Tahap Satu dan Mesin Unik pada Tahap Dua untuk Meminimasi Total Actual Flow Time. Tesis Magister: Program Studi Teknik dan Manajemen Industri ITB; 2006.
- Tejaasih, I. Pengembangan Model Penjadwalan Batch Sumber Serentak untuk Sistem Produksi Flow Shop. Tesis Magister: Program Studi Teknik dan Manajemen Industri ITB; 1998.

- Wahyuni, S. Penjadwalan Flow Shop Kelompok Mesin Heterogen untuk Meminimasi Total Actual Flow Time. Tesis Magister: Program Studi Teknik dan Manajemen Industri ITB; 2001.
- Zaini, E. Model Penjadwalan Batch pada Flow Shop Dinamis Sumber Simultan dengan Kriteria Minimasi Total Actual Flow Time. Tesis Magister: Program Studi Teknik dan Manajemen Industri ITB; 1999.