

BAB II

LANDASAN TEORI

2.1. PENGERTIAN DAN TUJUAN PENJADUALAN

Richard W. Conway [CON67], penjadualan adalah proses pengurutan pembuatan produk secara simultan dan sinkron pada beberapa mesin. Menurut Kenneth R. Baker [KEN74], penjadualan adalah proses pengalokasian sumber – sumber untuk memilih sekumpulan tugas dalam jangka waktu tertentu. Definisi yang diberikan Baker ini, mengandung dua arti, yaitu :

1. Penjadualan adalah fungsi pengambilan keputusan, yakni untuk menentukan jadwal (nilai praktis).
2. Penjadualan adalah suatu teori, yakni sekumpulan prinsip – prinsip dasar, model – model, teknik – teknik, dan kesimpulan logis yang memberikan pengertian dalam fungsi penjadualan (nilai konseptual).

Sedangkan tujuan penjadualan secara umum, antara lain adalah :

1. Meningkatkan utilitas mesin, dengan kata lain mengurangi waktu menganggur mesin.
2. Mengurangi tingkat persediaan barang setengah jadi, dengan kata lain mengurangi rata – rata antrian pekerjaan. Jika *makespan* konstan, urutan (*sequence*) yang mengurangi rata – rata *flowtime*, juga akan mengurangi rata – rata persediaan barang setengah jadi.
3. Mengurangi keterlambatan (*tardiness*). Dalam banyak hal, beberapa atau semua pekerjaan mempunyai *due date* (batas waktu penyelesaian) dan jika

suatu pekerjaan melewati batas tersebut, maka akan dikenai penalti. Keterlambatan dapat diperkecil dengan mengurangi maksimum *tardiness* atau mengurangi jumlah pekerjaan yang terlambat (*tardy*). Dalam metoda penjadualan baru, pekerjaan yang selesai lebih cepat (*early*) juga terkena penalti karena menimbulkan ongkos simpan.

2.2. ISTILAH – ISTILAH DALAM PENJADUALAN

Dalam pembahasan masalah penjadualan, sering ditemui beberapa istilah yang umum digunakan, antara lain :

a. Waktu proses (p_i)

Waktu proses adalah : Waktu yang diperlukan untuk menyelesaikan operasi atau proses dari pekerjaan ke-i. Waktu proses ini sudah mencakup waktu untuk persiapan dan pengaturan proses.

b. Due date (d_i)

Due date adalah : Batas akhir waktu pekerjaan ke-i boleh diselesaikan. Lewat dari batas ini, suatu pekerjaan dikatakan *tardy*.

$$d_i = d \dots \dots \dots (2.1)$$

dimana, d : *due date*

c. Completion Time / Waktu Penyelesaian (C_i)

Completion Time adalah : Rentang waktu sejak pekerjaan pertama dimulai ($t = 0$) hingga pekerjaan ke-i diselesaikan.

$$C_i = d - S_i \dots \dots \dots (2.2)$$

dimana, C_i : *completion time job ke-i*

S_i : *slack job ke-i*

d. **Lateness (L_i)**

Lateness adalah : Penyimpangan dari waktu penyelesaian hingga saat *due date*.

$$L_i = C_i - d_i < 0, \text{ saat penyelesaian memenuhi batas (early job)..... (2.3)}$$

$$L_i = C_i - d_i > 0, \text{ saat penyelesaian memenuhi batas (tardy job).....(2.4)}$$

dimana, L_i : penyimpangan dari waktu penyelesaian *job ke-i* hingga saat *due date-nya*.

e. **Tardiness (T_i)**

Tardiness adalah : Keterlambatan penyelesaian suatu pekerjaan dari saat *due date*.

$$T_i = \max_{1 \leq i \leq n} \{0, L_i\} \dots \dots \dots (2.5)$$

dimana, T_i : keterlambatan penyelesaian *job ke-i* dari saat *due date*

L_i : *lateness job ke-i*

f. **Slack (S_i)**

Slack adalah : Waktu sisa yang tersedia bagi suatu pekerjaan.

$$S_i = d_i - p_i \dots \dots \dots (2.6)$$

dimana, S_i : waktu sisa yang tersedia bagi *job ke-i*

d_i : *due date job ke-i*

p_i : waktu proses *job* ke- i

g. Makespan (MS)

Makespan adalah : Jangka waktu penyelesaian suatu penjadualan.
Merupakan penjumlahan dari seluruh waktu proses pada suatu mesin.

$$MS = p_1 + p_2 + p_3 + \dots + p_i \dots\dots\dots(2.7)$$

dimana, MS : jangka waktu penyelesaian suatu penjadualan.

$p_1, p_2, p_3, \dots, p_i$: waktu proses *job* 1 hingga *job* ke- i

h. Flow Time (F_i)

Flow Time adalah : Rentang waktu mulai dari pekerjaan ke- i siap untuk dikerjakan hingga pekerjaan diselesaikan.

$$F_i = C_i - R_i \dots\dots\dots(2.8)$$

dimana, C_i : waktu penyelesaian *job* ke- i

R_i : saat *job* ke- i siap untuk dikerjakan

i. Ready Time (R_i)

Ready Time adalah : Saat pekerjaan ke- i dapat dikerjakan (siap dijadualkan).

$$R_i = S^* \dots\dots\dots(2.9)$$

dimana, R_i : saat pekerjaan ke- i dapat dikerjakan.

S^* : *start time* atau saat pekerjaan mulai dikerjakan.

2.3. KLASIFIKASI MASALAH PENJADUALAN

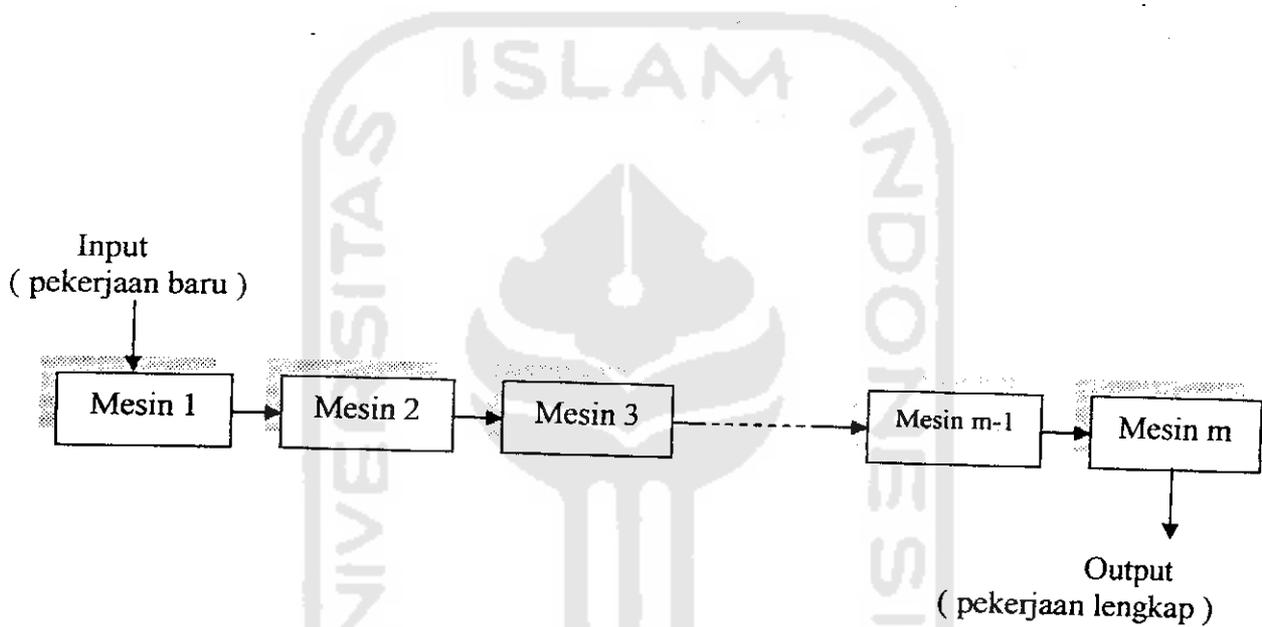
Struktur masalah penjadualan dipengaruhi oleh kondisi sistem produksi itu sendiri. Karena kompleksnya sistem produksi dalam keadaan nyata, maka masalah penjadualan dimodelkan untuk memudahkan penyelesaian masalah – masalah tersebut. Model penjadualan yang sering digunakan selama ini, umumnya dapat dibedakan atas beberapa keadaan :

1. Berdasarkan mesin yang digunakan.
2. Berdasarkan pola aliran proses.
3. Berdasarkan kedatangan pekerjaan.
4. Berdasarkan sifat informasi yang diterima.

Pada model pertama, jumlah mesin dibedakan atas mesin tunggal (penjadualan N job Satu mesin) dan mesin ganda (penjadualan N job M mesin). Masalah mesin tunggal sangat mendasar untuk analisa menyeluruh dan masalah ini dapat diterapkan pada mesin ganda.

Pada model kedua, pola aliran proses dapat dibedakan menjadi *flow shop* dan *job shop*. *Flow shop* ditandai dengan semua pekerjaan mengikuti lintas yang sama dari satu mesin ke mesin yang lain, melalui urutan mesin yang sama. Jika mesin diberi nomor, maka nomor mesin untuk operasi B akan lebih besar dari pada nomor mesin untuk operasi A pada pekerjaan yang sama, jika A mendahului B. Pola *flow shop* seperti ini, dikenal sebagai *pure flow shop*. Ada pula yang dikenal dengan *general flow shop*, yaitu pekerjaan yang mengikuti lintasan yang sama tetapi urutan mesin berbeda. Ini disebabkan karena suatu *shop* dapat menangani pekerjaan yang bervariasi, sedangkan pekerjaan yang datang ke *shop*, tidak harus dikerjakan pada

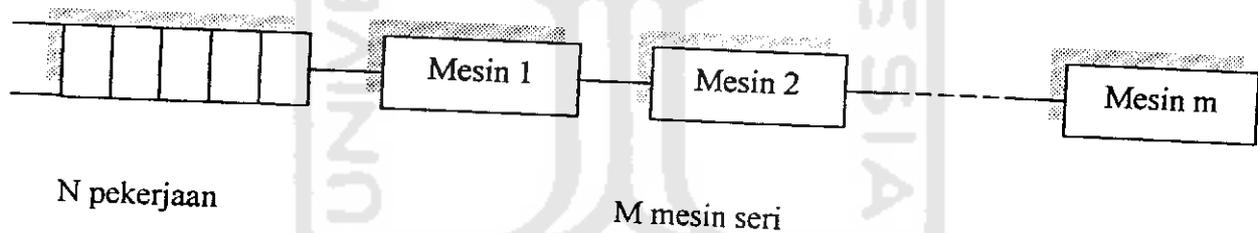
semua mesin. Pada pola *job shop*, aliran proses dalam *shop* tidak mempunyai pola yang umum. Nomor mesin yang sama, mungkin akan muncul dua kali atau lebih pada operasi – operasi suatu pekerjaan. Perbedaan kedua pola aliran tersebut, dapat dilihat pada gambar aliran pekerjaan berikut ini :



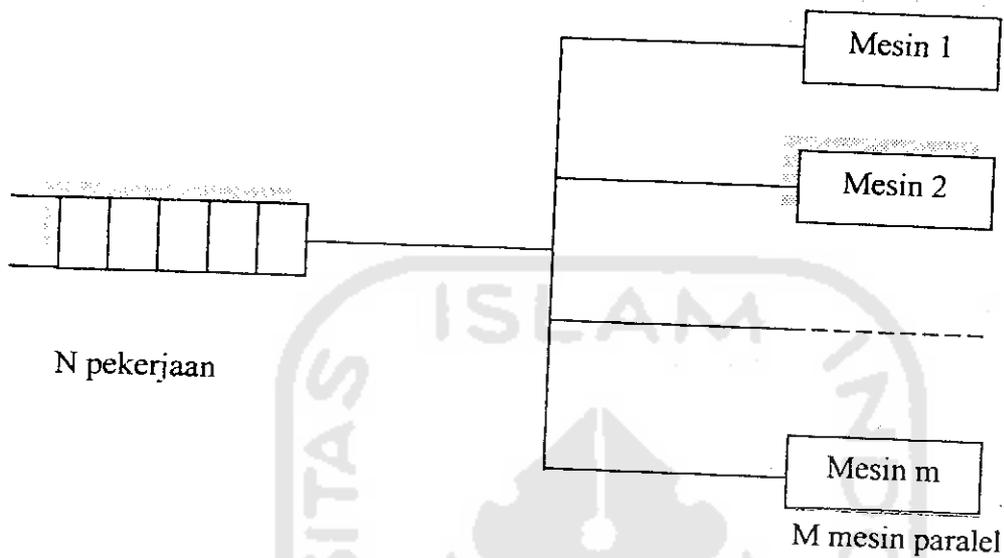
Gambar 2.1 Pola Aliran Kerja *Pure Flow Shop*

2.4. PENJADUALAN N JOB PADA M MESIN *STAGE* TUNGGAL

Pada model pertama klasifikasi masalah penjadualan yang dibahas pada sub – sub sebelumnya, terdapat penjadualan untuk mesin ganda atau penjadualan N job pada M mesin. Model ini terbagi lagi untuk mesin seri dan mesin paralel. Untuk mesin seri, tiap pekerjaan harus dikerjakan pada beberapa mesin secara berurutan (gambar 2.4), sedangkan untuk mesin paralel, tiap pekerjaan hanya dikerjakan pada satu mesin (gambar 2.5). Penjadualan jenis inilah yang akan menjadi topik bahasan dalam penelitian ini, dengan pekerjaan yang terlibat, bersifat *independent* (tiap pekerjaan hanya mempunyai satu operasi).



Gambar 2.4 N Job pada M Mesin Seri



Gambar 2.5 N Job pada M Mesin Paralel

Penjadualan pekerjaan *independent* ini, dapat ditemukan dalam buku “*Introduction to Sequencing and Scheduling*” karya Kenneth R. Baker [BAK74] ataupun “*Integrated Production Control Systems*” karya David D. Bedworth dan James E. Bailey [BED82].

2.4.1. Asumsi – Asumsi Yang Digunakan

Penelitian dalam memecahkan masalah penjadualan, sering mengalami berbagai kesulitan, terutama yang berkaitan dengan penundaan atau perubahan keputusan yang tidak terduga. Penundaan demikian dapat disebabkan oleh :

- a. Kemungkinan kerusakan mesin.
- b. Variasi kondisi kerja.
- c. Ketidak-hadiran pekerja.

- d. Penundaan pengiriman bahan baku.
- e. Perubahan yang mungkin dibuat pada spesifikasi tugas dan batas waktu penyerahan produk.
- f. Penekanan oleh pelanggan untuk mempercepat pekerjaan sehingga mengakibatkan penundaan pada pekerjaan yang lain.
- g. Kerusakan pada beberapa unit produk yang mengakibatkan perlunya pengulangan operasi.
- h. Kemungkinan – kemungkinan yang lain.

Berkenaan dengan kemungkinan – kemungkinan diatas, maka dalam memecahkan masalah penjadualan, dalam kasus ini penjadualan mesin paralel, diperlukan asumsi – asumsi yang menyangkut karakteristik pekerjaan dan mesin.

2.4.2. Asumsi mengenai pekerjaan

- a. Setiap pekerjaan diselesaikan menurut jadual yang telah disusun.
- b. Setiap pekerjaan merupakan satu kesatuan.
- c. Setiap pekerjaan yang telah dimulai, harus diselesaikan tanpa penyisipan pekerjaan yang lain diantaranya.
- d. Setiap pekerjaan tidak boleh diproses pada lebih dari satu mesin pada saat yang sama.
- e. Setiap pekerjaan hanya mempunyai satu operasi dan dapat dikerjakan pada mesin yang mana saja.

2.4.3. Asumsi mengenai mesin

- a. Setiap mesin dapat mengerjakan pekerjaan mana pun yang dibebankan padanya.
- b. Setiap mesin, secara kontinyu, siap untuk dibebankan pekerjaan selama periode penjadualan, tanpa mengalami interupsi oleh kerusakan / pemeliharaan mesin.
- c. Setiap mesin hanya dapat memproses satu pekerjaan pada satu saat.
- d. Operator yang menangani mesin selalu tersedia.

2.5. PENJADUALAN DENGAN ATURAN E / T (*Earliness / Tardiness*)

Seiring dengan berkembangnya prinsip *Just In Time* (JIT), metoda – metoda penjadualan yang dibuat mulai memperhatikan masalah waktu penyelesaian pekerjaan, dimana *earliness*, sama halnya dengan *tardiness*, perlu mendapat perhatian khusus. Dalam penjadualan JIT, pekerjaan yang selesai terlalu dini harus menunggu dalam gudang hingga saat *due date*, sedangkan pekerjaan yang selesai melampaui *due date* akan mengecewakan pelanggan. Karenanya, jadwal yang ideal adalah jadwal dimana semua pekerjaan selesai tepat pada *due date*-nya. Hal ini dapat diterjemahkan dalam fungsi tujuan penjadualan yang beraneka ragam. Salah satunya yang sedang banyak dikaji adalah untuk meminimasi simpangan waktu penyelesaian pekerjaan (*completion time*) disekitar *due date*-nya, baik *job* yang *early* maupun yang *tardy*, secara proporsional.

Sebenarnya, seluruh pembahasan mengenai masalah E/T, berkaitan dengan penjadualan statis. Dengan kata lain, pekerjaan yang akan dijadualkan telah diketahui

dan ada secara simultan. Kebanyakan penelitian mengenai masalah E/T masih dilakukan untuk mesin tunggal, walaupun beberapa diantaranya telah dikembangkan untuk mesin paralel.

Model – model untuk masalah E/T, biasanya menggunakan satu batas *due date* yang disebut *common due date* ($d_i = d$). *Common due date* tersebut dijadikan sebagai patokan pembentukan jadual dan ditentukan secara internal oleh pihak perusahaan, berdasarkan pengalaman pada masa lampau. Dari batas *due date* inilah, pekerjaan mulai disusun sesuai dengan aturan yang digunakan, apakah diletakkan di sebelah kiri atau di sebelah kanannya. *Common due date* juga dapat digunakan apabila order – order yang akan dikerjakan, berasal dari pelanggan tunggal atau bisa juga merupakan komponen – komponen dari suatu produk yang nantinya akan dirakit secara bersamaan. Dengan menggunakan *common due date*, model penyelesaian masalah penjadualan akan menjadi lebih sederhana dibandingkan dengan menggunakan *due date* yang berbeda untuk tiap pekerjaan.

Penentuan *due date* ini, bersifat fleksibel karena banyak kemungkinan yang bisa dipilih, hanya saja harus diperhatikan bahwa tidak boleh ada pekerjaan yang dimulai sebelum $t = 0$ setelah jadual dibuat. Jika hal ini terjadi, maka *due date* harus diubah sehingga semua pekerjaan dapat diselesaikan sesuai jadual. *Due date* tidak boleh terlalu ketat karena kita tidak dapat memuat cukup banyak pekerjaan didepannya, disebabkan oleh adanya batasan bahwa tidak ada pekerjaan yang dapat dimulai sebelum $t = 0$. Perusahaan juga dapat menetapkan *due date* sedemikian rupa, sehingga tersedia tenggang waktu sebelum pekerjaan mulai dilaksanakan, untuk keperluan persiapan awal bagi pekerjaan – pekerjaan baru ataupun pemesanan bahan

baku yang tidak tersedia. Jika pekerjaan tidak memerlukan tenggang waktu dan dapat segera dimulai, maka perusahaan dapat menggunakan *due date* minimum, untuk mempercepat penyelesaian pekerjaan.

Dalam beberapa masalah E/T, *due date* ditentukan, sedangkan pada masalah lain, *due date* optimal yang dicari. Beberapa model menggunakan penalti yang sama untuk *earliness* dan *tardiness*, sementara model yang lain menggunakan penalti yang berbeda. Dalam penelitian ini, akan digunakan penalti dari kebijaksanaan perusahaan, bagi *job – job* yang *early* maupun yang *tardy*.

Dengan banyaknya model untuk masalah E/T dalam literatur, orang mungkin bertanya, mana model yang paling baik. Sebenarnya, setiap model mempunyai kelebihan masing – masing. Tujuan utamanya sama, yaitu untuk mencari solusi dalam memenuhi target penyelesaian pekerjaan, tepat pada saat *due date*. Salah satu model yang penting dalam masalah E/T adalah model yang menggunakan kriteria minimasi penyimpangan waktu penyelesaian pekerjaan (*completion time*) disekitar *due date*-nya, baik *job* yang *early* maupun yang *tardy*, secara proporsional. Penelitian mengenai masalah ini, salah satunya dilakukan oleh Ram Rachamadugu.

2.5.1. Penjadualan *Job – Job* Dengan Penalti *Early / Tardy* Yang Proporsional

Dalam menyelesaikan masalah penjadualan *job – job* dengan penalti – penalti *early / tardy* yang proporsional, akan digunakan teorema – teorema yang dikemukakan oleh Ram Rachamadugu. Salah satu dari teorema tersebut, membahas tentang *job – job* yang mempunyai suatu *common due date*. Dikatakan bahwa jika *job – job* mempunyai *common due date*, banyak terdapat solusi optimal; dan dalam

prinsip tentang penalti yang proporsional untuk *job – job* yang *early* dan yang *tardy*, aturan *Longest Processing Time* (LPT) akan menghasilkan urutan *job* yang optimal pada *single* mesin. Sementara, suatu rumus yang sederhana, disediakan pula untuk menentukan waktu mulai (*start time*) yang optimal bagi *job* pertama dari urutan tersebut. Sebelum melangkah lebih jauh, dibawah akan diuraikan notasi – notasi yang digunakan dalam penelitian ini.

Notasi – notasi yang digunakan :

J_i : indeks *job*, $i = 1, 2, 3, \dots, n$

N : set dari *job – job* yang akan dijadualkan, $N = \{1, 2, 3, \dots, n\}$

p_i : waktu proses untuk *job* i

$$P : \sum_{i \in N} p_i$$

d_i : *due date* untuk *job* i (jika *job* mempunyai *common due date*, subscript $[i]$ dihilangkan)

w_i : bobot penalti *tardiness* untuk *job* i ($w_i = \beta p_i$)

h_i : bobot penalti *earliness* untuk *job* i ($h_i = \alpha p_i$)

α : konstanta ke-proporsional-an untuk bobot penalti *earliness*

β : konstanta ke-proporsional-an untuk bobot penalti *tardiness*

$x^+ : \max(0, x)$

C_i : completion time dari job i

s_i : slack untuk job i

T : set dari job – job yang *tardy* dalam suatu urutan

(termasuk didalamnya, job yang selesai tepat pada *due date*-nya : j^*)

E : set dari job – job yang *early* dalam suatu urutan ($E = N/T$)

S^* : start time untuk job yang pertama dalam suatu urutan optimal dalam masalah *common due date*

j^* : indeks job (jika ada) yang selesai tepat pada *common due date* (d)

LPT : urutan *longest processing time*

SPT : urutan *shortest processing time*

CDD : masalah *common due date*

RS : ruas kanan dari suatu persamaan

LS : ruas kiri dari suatu persamaan

\prec : menunjukkan *job precedence*

($J_i \prec J_j$ berarti J_i mendahului J_j dalam suatu urutan)

Tujuan dari penelitian ini adalah untuk mencari suatu jadual yang meminimasi :

$$Z = \sum_{i=1}^n \alpha (d - C_i)^+ + \beta (C_i - d)^+ \dots \dots \dots (2.19)$$

Dibawah ini adalah teorema – teorema yang dikemukakan oleh Rachamadugu tentang kondisi – kondisi yang dominan ada pada suatu solusi optimal dalam permasalahan *early / tardy* yang proporsional.

TEOREMA I

Terdapat suatu jadual yang optimal (yang mungkin telah disisipi *idle time*), dimana *job – job* yang berurutan ($J_i < J_j$) memenuhi kondisi berikut :

$$p_i \min\{(d_i - t - p_i)^+, p_j\} \leq p_j \min\{(d_j - t - p_j)^+, p_i\} \dots \dots \dots (2.20)$$

dengan t adalah *start time* untuk *job i*.

TEOREMA II

Jika s_i adalah fungsi yang *nondecreasing* dari p_i (disebabkan *due date* yang dipakai, ditentukan menggunakan aturan – aturan, seperti : *total work content & equal slack*), maka aturan SPT (*Shortest Processing time*) menghasilkan urutan yang optimal.

Dalam penyusunan laporan Tugas Akhir ini, teorema ini tidak digunakan.

TEOREMA III

Aturan LPT (*Longest Processing Time*) menghasilkan suatu urutan yang optimal untuk permasalahan *common due date* dalam prinsip penalti yang proporsional pada *single* mesin.

2.5.2. Penentuan *Start Time* yang Optimal (S^*)

Setelah diurutkan dengan aturan LPT (Teorema III), *start time* yang optimal dari *job* pertama dalam urutan tersebut pada suatu mesin, masih harus ditentukan.

Rachamadugu menyediakan suatu rumus yang sederhana untuk menyelesaikannya.

Dibawah ini adalah rumus yang diberikan :

$$\text{Jika, } \sum_{k \in T \setminus J^*} p_k \leq \frac{\alpha}{(\alpha + \beta)} P \leq \sum_{k \in T} p_k \dots \dots \dots (2.21)$$

S^* dapat dicari dengan mengidentifikasi set dari *job - job* yang berurutan pada akhir LPT, yang memenuhi persamaan (2.21). Hal ini dapat dilakukan dengan mudah, yaitu dengan menjumlahkan waktu proses *job* terakhir dalam urutan LPT, sampai persamaan (2.21) terpenuhi. Sedangkan :

$$S^* = d - p_{j^*} - \sum_{k \in E} p_k \dots \dots \dots (2.22)$$

Jika tidak ada solusi dari persamaan tersebut (S^* negatif), maka *start time* yang optimal adalah 0 (nol).

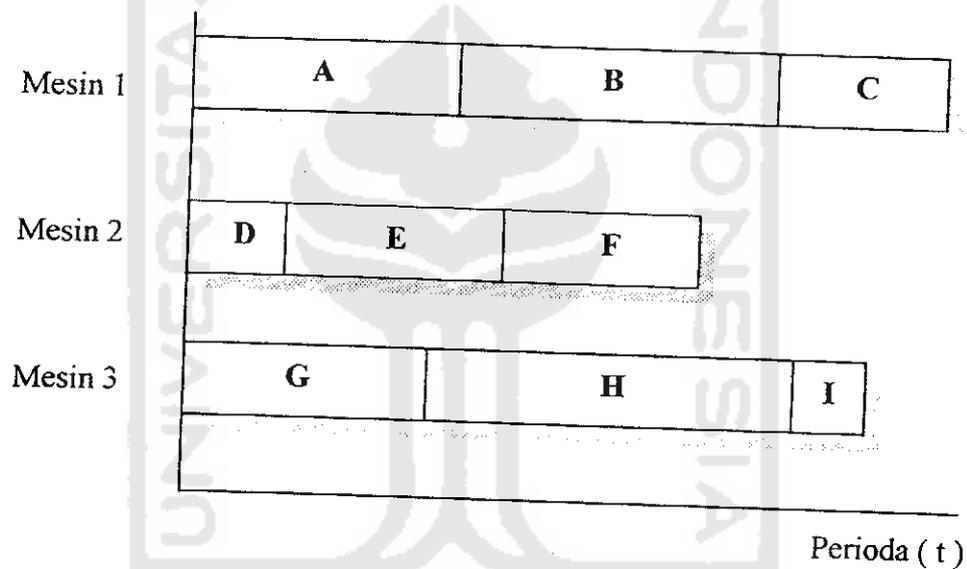
2.6. PETA GANTT (*GANTT CHART*)

Peta Gantt dikembangkan oleh Henri L. Gantt sewaktu Perang Dunia I. Peta Gantt merupakan representasi grafis dari pekerjaan - pekerjaan yang harus diselesaikan, dan digambarkan dalam bentuk batang yang analog dengan waktu penyelesaian pekerjaan - pekerjaan tersebut. Keuntungan dari peta Gantt, antara lain :

- a. Semua pekerjaan diperlihatkan secara grafis dalam satu peta yang mudah dipahami.

- b. Kemajuan pekerjaan, mudah diamati / diperiksa setiap waktu karena sudah tergambar dengan jelas.
- c. Dengan situasi keterbatasan sumber, penggunaan peta Gantt memungkinkan evaluasi yang lebih awal mengenai penggunaan sumber, seperti yang telah direncanakan.

Sedangkan salah satu contoh peta Gantt, diperlihatkan pada gambar 2.6 dibawah ini.



Gambar 2.6 Contoh Peta Gantt

Keterangan :

A, B, C, D, E, F, G, H, I = Job