

BAB III

LANDASAN TEORI

3.1 Tinjauan Umum

Proyek konstruksi merupakan suatu usaha untuk mencapai tujuan atau hasil tertentu dalam bentuk bangunan dengan dibatasi oleh tiga kendala, yaitu waktu, biaya dan kualitas. Proyek konstruksi memiliki sifat-sifat, yaitu dimulai dari awal proyek (awal rangkaian kegiatan) dan diakhiri dengan akhir proyek (akhir rangkaian kegiatan) serta mempunyai waktu yang terbatas dan rangkaian kegiatan proyek tersebut hanya satu kali, sehingga menghasilkan produk yang bersifat unik. Oleh karena itu dalam pelaksanaan proyek konstruksi tidak dikehendaki adanya keterlambatan. Sebab dengan adanya keterlambatan tersebut akan mengakibatkan terjadinya penambahan biaya.

Perlu adanya pendekatan pengelolaan proyek yang sesuai dengan tuntutan dan sifat dari kegiatan proyek konstruksi tersebut tersebut, yaitu dengan manajemen proyek. Manajemen proyek merupakan sebuah proses perencanaan, pengorganisasian, memimpin dan mengendalikan sumber daya perusahaan untuk mencapai sasaran jangka pendek yang telah ditentukan (Iman Suharto, 1995). Manajemen konstruksi dapat diartikan dengan penerapan fungsi-fungsi manajemen secara sistematis pada suatu proyek dengan menggunakan sumber daya secara efektif dan efisien agar tujuan proyek tercapai secara optimal.

3.2 Penjadwalan (“*Scheduling*”)

Dalam industri konstruksi sering terdapat kerancuaan antara istilah “*planning*” dengan “*scheduling*”. Kedua istilah tersebut sering kali disinonimkan dengan istilah “*planning and scheduling engineer*” (teknik perencanaan dan penjadwalan). Proses penjadwalan dalam proyek konstruksi merupakan salah satu bagian dari tiga aspek perencanaan, yaitu estimasi biaya, penjadwalan dan pengendalian (Callahan, 1992). Perencanaan sebagai bagian dari fungsi manajemen konstruksi merupakan sebuah proses pemilihan tujuan yang kemudian disusun dalam bentuk rencana dan metode untuk mencapai tujuan tersebut (Ahuja, 1994). Secara umum terdapat beberapa elemen dari proses perencanaan yang berhubungan dengan penjadwalan (Taylor, 1996), yaitu :

- 1) menetapkan tujuan proyek,
- 2) mengidentifikasi kegiatan-kegiatan,
- 3) menyusun hubungan *precedence*,
- 4) membuat estimasi waktu,
- 5) menentukan waktu penyelesaian proyek,
- 6) membandingkan sasaran-sasaran dari penjadwalan proyek,
- 7) menentukan kebutuhan sumber daya yang memenuhi tujuan.

Penjadwalan konstruksi (“*construction schedule*”) dapat berarti sebuah alat untuk menentukan kegiatan-kegiatan yang diperlukan untuk menyelesaikan suatu proyek dan urutan serta kerangka waktu dari kegiatan-kegiatan yang harus diselesaikan, dengan tujuan untuk mendapatkan penyelesaian proyek yang tepat pada waktunya dan ekonomis (Callahan, 1992). Terdapat beberapa metode penjadwalan

yang dapat digunakan untuk mengelola proyek konstruksi. Pemilihan metode penjadwalan tersebut tergantung dari karakteristik dari tiap-tiap proyek. Pada proyek dengan jumlah kegiatan yang sedikit atau tidak banyak interaksi antar kegiatan-kegiatan, dapat menggunakan metode penjadwalan diagram balok ("*Gantt Chart*") atau penjadwalan *linear*. Proyek dengan jumlah kegiatan yang banyak dan kegiatan tersebut saling mempengaruhi, maka dapat menggunakan penjadwalan dengan diagram jaringan CPM/PERT atau dengan diagram jaringan PDM.

3.3 *Gantt Chart*

Diagram balok yang ditemukan oleh H.L. Gantt pada tahun 1917 ini merupakan diagram yang paling banyak digunakan dalam penjadwalan proyek konstruksi karena kemudahannya. Dengan berpedoman bahwa diagram balok disusun dengan maksud mengidentifikasi unsur-unsur waktu dari urutan dalam merencanakan suatu kegiatan, yang terdiri dari saat dimulai sampai saat selesai. Diagram balok masih digunakan secara luas, disebabkan oleh karena bagan balok mudah dibuat dan dipahami oleh setiap level manajemen, sehingga sangat berguna sebagai alat komunikasi dalam pelaksanaan proyek. *Gantt Chart* juga diartikan sebagai suatu diagram yang terdiri dari sekumpulan garis yang menunjukkan saat mulai dan saat selesai yang direncanakan untuk item-item pekerjaan didalam proyek.

3.4 *Network Planning*

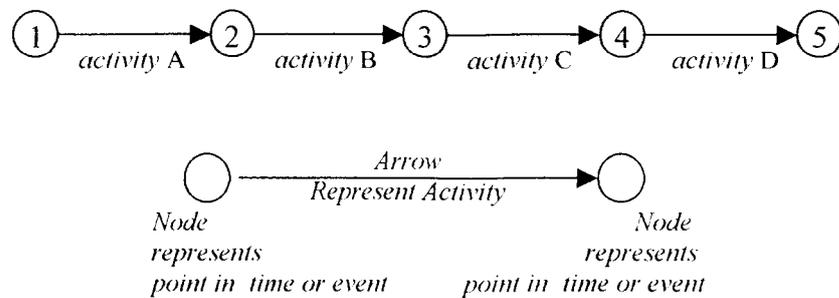
Network planning adalah alat untuk mengkoordinasikan berbagai macam kegiatan yang ada yang satu sama lainnya bebas dan atau saling bergantung berdasarkan pertimbangan sumber daya yang digunakan, logika proses yang berlangsung, dan hasil proses itu sendiri. Dalam pemakainya, yaitu pada

penyelenggaraan proyek, *network planning* menggunakan model yang berupa diagram yang disebut *network diagram*. *Network diagram* adalah visualisasi proyek berdasarkan *network planning* berupa diagram yang berisi lintasan-lintasan yang terdiri dari kegiatan-kegiatan yang harus dikerjakan dan terdiri dari peristiwa-peristiwa yang harus terjadi selama penyelenggaraan proyek (Ali, 1989).

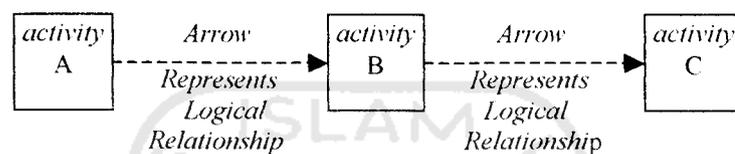
Pada *network diagram* dapat dilihat kaitan suatu kegiatan dengan kegiatan-kegiatan yang lainnya, sehingga bila sebuah kegiatan terlambat maka dengan segera dapat dilihat kegiatan apa saja yang dipengaruhi oleh keterlambatan tersebut dan seberapa besar pengaruhnya. Juga dengan *network diagram* dapat diketahui kegiatan-kegiatan mana saja atau lintasan-lintasan mana saja yang kritis, sehingga dengan mengetahui tingkat kekritisannya dapat ditetapkan skala prioritas dalam menangani masalah-masalah yang timbul selama penyelenggaraan proyek.

Menurut F.H. (Bud) Grifis dan John V. Farr (2000), terdapat dua macam penjadwalan dengan *network diagram*, yaitu *activity on arrow* (AOA) dan *activity on node* (AON). Pada jaringan dengan metode AOA, anak panah menunjukkan kegiatan atau aktivitas sedangkan *node* (titik) menunjukkan *event* (peristiwa). Pada jaringan dengan metode AON atau *precedence*, *node* menunjukkan aktivitas sedangkan anak panah menunjukkan hubungan antar *node*. Jaringan dengan metode AOA tersebut digunakan dalam CPM (“*Critical Path Method*”) dan PERT (“*Program Evaluation and Review Technique*”). Sedangkan jaringan yang menggunakan metode AON adalah PDM (“*Precedence Diagram Method*”).

Activity-on-arrow diagram



Activity-on-node diagram



Gambar 3.1 Diagram AOA Dan Diagram AON

(Sumber : Callahan, 1992)

Pada *precedence* diagram tidak dikenal adanya peristiwa sedangkan pada CPM/PERT dikenal adanya peristiwa pada setiap awal dan akhir kegiatan. Pada *precedence* diagram dimungkinkan suatu kegiatan dimulai sebelum kegiatan pendahulunya selesai seluruhnya. Hal ini tidak mungkin terjadi pada *network diagram* CPM/PERT. Ini merupakan kelemahan dari *network diagram* CPM/PERT, tetapi kelemahan tersebut dapat diatasi dengan menggunakan modul operasi, yaitu pekerjaan yang jenisnya sama dibagi-bagi berdasarkan lokasi dan kemampuan gerak sumber daya yang tersedia.

3.4.1 PERT

CPM dan PERT dikembangkan secara terpisah pada akhir tahun 1950-an. PERT dikembangkan oleh sekelompok peneliti yang terdiri dari U.S. Navy Special Project Office dan konsultan manajemen Booz, Allen, and Hamilton untuk

merencanakan dan mengendalikan produksi dari kapal selam Polaris. PERT dikembangkan untuk menghadapi situasi dengan kadar ketidakpastian yang tinggi pada aspek kurun waktu kegiatan. PERT juga memakai pendekatan yang menganggap kurun waktu kegiatan tergantung pada banyak faktor dan variasi.

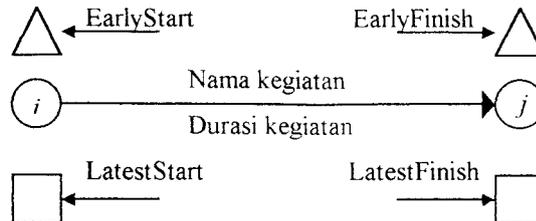
3.4.2 CPM

CPM dikembangkan antara tahun 1956 sampai dengan 1958 oleh Morgan Walker dari E. I. Du Pont de Nemours Company dan James E. Kelly, kemudian oleh UNIVAC Application Research Center of Remington Rand dalam usahanya untuk mengembangkan sistem kontrol manajemen. CPM merupakan suatu teknik perencanaan dengan analisis jaringan ("*network*") berdasarkan logika ketergantungan antar aktivitas yang ada dalam proyek. Metode CPM bersifat determinan, dimana metode ini mengasumsikan bahwa setidaknya terdapat satu jalur yang melalui jaringan merupakan jalur yang menentukan durasi proyek dan jalur tersebut disebut jalur kritis ("*critical path*"). Dengan penggunaan CPM ini akan terjawab hal-hal yang belum bisa digunakan dalam bagan balok, seperti berapa lama perkiraan kurun waktu penyelesaian proyek, kegiatan mana yang bersifat kritis dan jika terjadi keterlambatan maka bagaimana pengaruhnya terhadap kegiatan yang lain.

Dasar dari mendirikan bangunan dalam jaringan proyek adalah kegiatan (Grifis dan Farr, 2000). Diagram jaringan AOA seperti terlihat pada Gambar 3.2, mewakili kegiatan tersebut. Titik permulaan dan titik akhir dari kegiatan disebut dengan peristiwa-peristiwa ("*events*"). Peristiwa-peristiwa ini sama dengan *node* dalam jaringan dan diwakili dengan lingkaran seperti terlihat dalam Gambar 3.2.

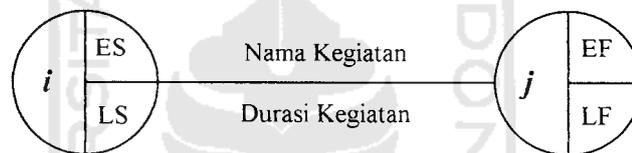
Pada diagram jaringan CPM terdapat aturan-aturan dasar sebagai berikut :

- 1) Jaringan harus mempunyai satu awalan *node* (tanpa ada kegiatan yang mendahului/"*prodecessor*") dan satu akhiran *node* (tanpa ada kegiatan yang didahului/"*successor*") masing-masing mewakili permulaan dan penyelesaian proyek.



Gambar 3.2 Elemen Dasar Dari Jaringan AOA
(Sumber : Grifis dan Farr, 2000)

Diagram jaringan AOA tersebut diatas dapat disederhanakan menjadi :



Gambar 3.3 *Node* Jaringan AOA Yang Disederhanakan
(Sumber : Diktat Manajemen Konstruksi, Faisol)

- 2) Sebelum sebuah kegiatan dapat dimulai, seluruh kegiatan yang mendahului kegiatan tersebut harus sudah selesai terlebih dahulu.
- 3) Hanya dua peristiwa yang dapat dihubungkan dengan hanya satu kegiatan.

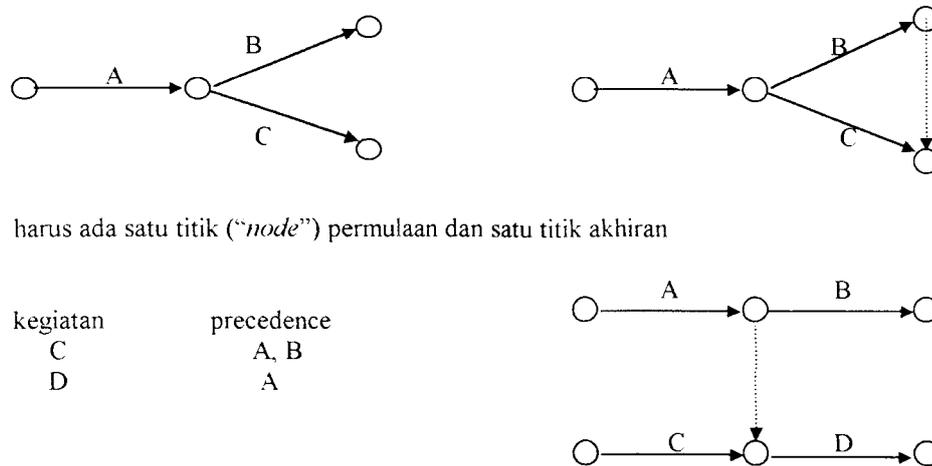
Pada Gambar 3.2 dapat kita lihat bahwa *node* pada permulaan anak panah (bagian ekor) disebut dengan *node* "i". Sedangkan *node* yang berada pada bagian akhir anak panah (bagian kepala) disebut dengan *node* "j". Karena diagram AOA menghubungkan keseluruhan *node* kegiatan, maka *node j* dari kegiatan sebelumnya merupakan *node i* dari kegiatan sesudahnya. Diagram AOA juga sering disebut dengan diagram *i-j*, karena *i* atau *j* merupakan penandaan dari *node* (Callahan, 1992).

Setiap kegiatan (“*arrow*”) memiliki diskripsi yang singkat. Diskripsi dari kegiatan biasanya dapat dituliskan dibawah atau diatas anak panah. Sebagai tambahan dari diskripsi kegiatan adalah setiap kegiatan telah ditentukan durasinya. Dalam pembahasan kali ini diskripsi kegiatan diletakkan diatas anak panah dan durasi kegiatan diletakkan dibawah anak panah. Setiap *node* mewakili satuan waktu. *Node i* merupakan satuan waktu dimana kegiatan dimulai dan *node j* merupakan satuan waktu dimana kegiatan berakhir. Jadi satuan waktu mewakili “*event*” baik pada permulaan atau akhiran kegiatan. Penyelesaian dari kegiatan sebelumnya mewakili satuan waktu permulaan kegiatan sesudahnya. Setiap *node* mempunyai nomer identitas. Setiap anak panah atau kegiatan dapat diidentifikasi dengan nomer yang terletak pada *node i* dan *node j*.

Seringkali sebuah kegiatan *dummy* harus digunakan dalam jaringan untuk mengoreksi permasalahan ketergantungan dan agar sesuai dengan syarat-syarat dasar CPM. Perulangan dicegah dengan adanya anak panah “*dummy*” (digambarkan dengan garis putus-putus). Kegiatan *dummy* juga merupakan alat yang mengijinkan beberapa batasan yang logis. Anak panah *dummy* hanya menunjukkan hubungan ketergantungan saja, tidak memiliki durasi. Jumlah anak panah *dummy* dalam jaringan harus sekecil mungkin. Pada Gambar 3.4 ditunjukkan kegunaan dari kegiatan *dummy*.



hanya satu kegiatan yang dapat menghubungkan dua peristiwa



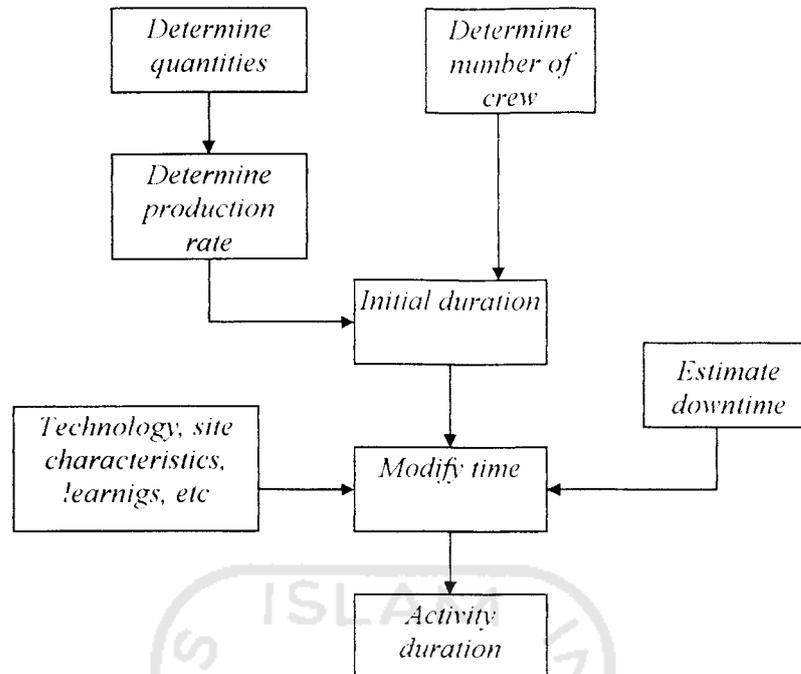
Gambar 3.4 Penggunaan Kegiatan *Dummy*
(Sumber : Grifis dan Farr, 2000)

Terdapat beberapa tahap dalam perhitungan dengan metode CPM sebagai berikut :

- 1) estimasi durasi kegiatan
- 2) melakukan perhitungan maju
- 3) melakukan perhitungan mundur
- 4) menghitung *total float*
- 5) menghitung *free float*
- 6) menentukan jalur kritis

1) Estimasi durasi kegiatan

Salah satu aspek tersulit dalam penjadwalan adalah estimasi durasi dari kegiatan-kegiatan. Seperti halnya estimasi biaya dari kegiatan. Kita harus mengetahui metode konstruksi yang akan digunakan, sumber daya yang tersedia, volume pekerjaan, dan yang paling penting adalah produktivitas. Secara umum algoritma untuk menentukan durasi kegiatan dapat dilihat pada skema dibawah ini.



Gambar 3.5 Skema Estimasi Durasi Kegiatan
(Sumber : Grifis dan Farr, 2000)

Durasi kegiatan jaringan AOA ini dihitung dengan rumus : $T_{ij} = (Q_{ij} / k_{ij}N_{ij}) + D_{ti}$,

dimana : T_{ij} = durasi kegiatan

Q_{ij} = kuantitas/volume pekerjaan

k_{ij} = jumlah tenaga kerja

N_{ij} = produktivitas

DT_{ij} = *downtime* yang berhubungan dengan pengendalian sumber daya

2) Melakukan perhitungan maju

Dasar dari semua perhitungan penjadwalan pertama kali melibatkan perhitungan maju baru kemudian melakukan perhitungan mundur dari awal sampai akhir jaringan. Untuk menjelaskan perhitungan tersebut, istilah-istilah dibawah ini harus ditetapkan terlebih dahulu :

$d(i,j)$ = estimasi durasi dari kegiatan (i,j)

$ES(i,j)$ = waktu paling cepat yang diharapkan dari permulaan kegiatan (i,j)

$EF(i,j)$ = waktu paling cepat yang diharapkan dari akhir kegiatan (i,j)

$LS(i,j)$ = waktu paling lambat yang diharapkan dari permulaan kegiatan (i,j)

$LF(i,j)$ = waktu paling lambat yang diharapkan dari akhir kegiatan (i,j)

$TF(i,j)$ = total *float (slack)* dari kegiatan (i,j)

$FF(i,j)$ = free *float* dari kegiatan (i,j)

Perhitungan maju digunakan untuk menghitung waktu paling cepat dari awal dan akhir kegiatan. Algoritma berikut ini digunakan dalam perhitungan maju:

- a. Atur waktu permulaan proyek sama dengan 0
- b. $ES(i,j) = \max[EF(k,i)$ untuk semua $k]$
- c. $EF(i,j) = ES(i,j) + d(i,j)$ untuk setiap titik (“*node*”)

dimana k adalah akhir dari peristiwa untuk semua kegiatan yang mendahului (“*precedence*”) (i,j) .

3) Melakukan perhitungan mundur

Perhitungan ini menentukan waktu paling lambat yang diharapkan dari permulaan dan akhir dari setiap kegiatan. Pada intinya, persamaan ini adalah kebalikan dari perhitungan maju. Algoritma berikut ini digunakan dalam perhitungan mundur :

- a. Waktu penyelesaian proyek T merupakan waktu maksimum dari semua waktu paling cepat dari akhir kegiatan dengan tidak ada kegiatan yang didahului (“*successor*”) lagi.
- b. $LF(i,j) = \min[LS(j,k)$ untuk semua $k]$
- c. $LS(i,j) = LF(i,j) - d(i,j)$

4) Menghitung *total float*

Total float (TF) seringkali diartikan dengan *slack* (waktu luang), terdapat perbedaan untuk setiap kegiatan antara waktu paling lambat dan waktu paling cepat dari akhir kegiatan atau antara waktu paling lambat dan waktu paling cepat dari permulaan kegiatan. *Total float* adalah waktu maksimum dari kegiatan yang dapat ditunda tanpa memperpanjang waktu penyelesaian proyek.

$$TF(i,j) = LS(i,j) - ES(i,j) = LF(i,j) - EF(i,j)$$

5) Menghitung *free float*

Free float (FF) berbeda dari *total float*, FF adalah waktu maksimum dari kegiatan yang dapat ditunda tanpa mempengaruhi waktu paling cepat dari awal urutan kegiatan manapun.

$$FF(i,j) = \text{Min}[ES(j,k) \text{ untuk semua } k] - EF(i,j)$$

6) Menentukan jalur kritis

Jalur kritis adalah urutan dari kegiatan yang mana ketika durasi dari kegiatan tersebut dijumlahkan akan menentukan durasi proyek yang minimum. Jalur kritis adalah urutan kegiatan yang memiliki *total float* sama dengan nol. Setiap keterlambatan dari kegiatan yang terdapat pada jalur kritis akan mengakibatkan terlambatnya waktu penyelesaian proyek (Grifis dan Farr, 2000). Dalam suatu jaringan mungkin saja terdapat lebih dari satu jalur kritis. Sifat atau syarat umum jalur kritis menurut Iman Suharto (1995), adalah :

- a. Pada kegiatan pertama $ES = LS = 0$ atau $E(1) = L(1) = 0$
- b. Pada kegiatan terakhir $LF = EF$
- c. *Float Total* : $TF = 0$

3.5 Pemadatan Durasi

Menurut Callahan (1992), cara termudah untuk mengurangi durasi proyek atau bagian dari proyek adalah dengan menghilangkan batasan yang tidak perlu diantara bermacam-macam kegiatan. Cara yang lain adalah dengan mengurutkan kembali kegiatan-kegiatan, kemudian membaginya dalam kegiatan-kegiatan yang lebih kecil sehingga dapat dilakukan bersamaan dan mengurangi keseluruhan durasi dengan *overlapping* berbagai kegiatan. Sedangkan menurut Willis (1986), durasi proyek dapat dikurangi dengan cara sebagai berikut :

1) Mengerjakan proyek dengan giliran kerja ("*multiple shifts*").

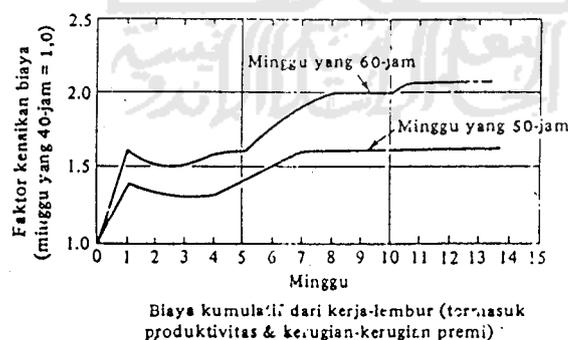
Penggunaan giliran kerja akan menyebabkan performa dari kegiatan-kegiatan tertentu akan menjadi lebih cepat. Pekerja pada *shift* malam hari akan mendapatkan upah lebih dari upah normal dan biasanya produktivitas pekerja pada malam hari lebih rendah dari produktivitas pekerja pada siang hari. Jika giliran kerja tersebut dirotasikan secara teratur (tiap minggu atau tiap dua minggu), maka sifat alamiah dari tenaga kerja (bekerja pada siang hari dan pada malam hari istirahat) akan terganggu dan akan menyebabkan produktivitas tenaga kerja akan menurun.

Selain itu terdapat faktor lain, yaitu bahwa beberapa orang memang benar-benar orang "siang hari" dan orang-orang lainnya memang orang "malam hari". Mereka akan dapat berprestasi lebih baik bilamana pekerjaan mereka cocok dengan fisiologis mereka sendiri. Kesimpulan dari penggunaan giliran kerja akan lebih produktif bila giliran kerja tersebut tidak dirotasikan secara tetap dan dilakukan penyesuaian tenaga kerja dengan regu giliran dimana dia akan berprestasi lebih baik (Barrie dan Poulson, 1990). Tetapi penggunaan giliran kerja secara tepat harus

mempertimbangkan hubungan antara setiap kegiatan dengan penerangan pada waktu malam hari dan batasan lingkungan kerja yang lain, *support services*, dan *safety*.

2) Dengan memperpanjang waktu atau hari kerja (lembur).

Pekerja yang bekerja lebih dari waktu normal 8 jam kerja tiap hari atau lebih dari waktu normal 40 jam kerja tiap minggu akan mendapatkan upah lebih dari upah normal untuk waktu lembur mereka. Akibat dari kerja lembur ini adalah akan menurunnya produktivitas keseluruhan pada waktu kerja 50 jam atau 60 jam per minggu dibawah produktivitas pada waktu kerja 40 jam per minggu. Meskipun produktivitas mereka tiap hari atau tiap minggu lebih tinggi dari pada produktivitas mereka pada waktu normal, tetapi produktivitas mereka tiap jamnya akan menurun. Penurunan produktivitas ini dikarenakan keletihan, meningkatnya ketidakhadiran, gangguan operasi dan angka-angka kecelakaan yang semakin meningkat. Penggunaan kerja lembur ini akan menaikkan biaya tenaga kerja sebanyak 50 hingga 100 persen, hal tersebut dapat dilihat pada grafik pada Gambar 3.6 (Barrie dan Poulson, 1990).



Gambar 3.6 Kerja Lembur vs Produktivitas
(Sumber : Barrie dan Poulson, 1990)

3) Menggunakan peralatan yang lebih besar atau tambahan peralatan di tempat kerja.

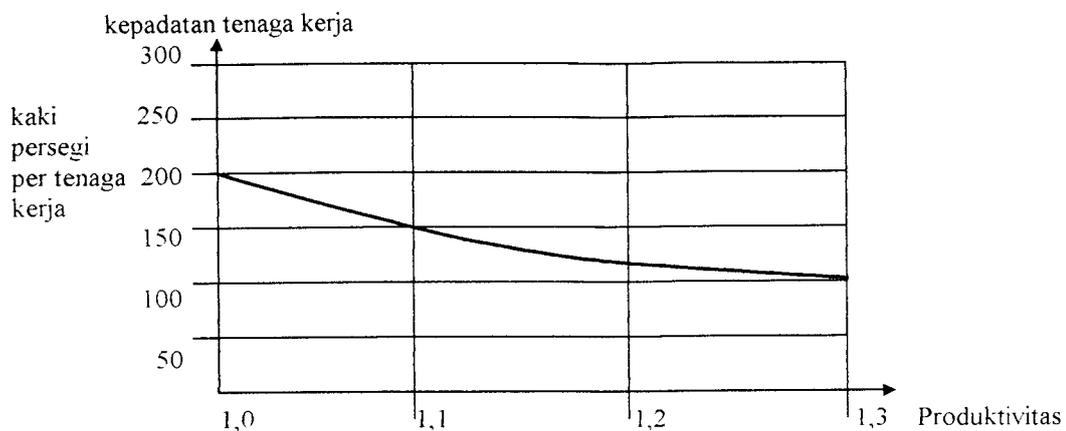
Biaya dari penyediaan peralatan di tempat kerja adalah fungsi dari biaya per jam dan biaya untuk mobilisasi peralatan tersebut. Jika peralatan akan digunakan

hanya satu hari saja, maka biaya untuk mobilisasi peralatan merupakan sebagian besar dari biaya total. Terdapat beberapa hal yang perlu dipertimbangkan dalam menghitung produksi alat berat dan pemilihan alat berat yang akan digunakan (Suryadharma dan Wigroho, 1998) antara lain sebagai berikut :

- a. waktu yang dibutuhkan untuk menyelesaikan pekerjaan,
- b. material yang dikerjakan, berat volume dan jenis tanah,
- c. kemampuan operator,
- d. keadaan medan kerja,
- e. kondisi alat yang digunakan,
- f. efisiensi kerja, disini dipertimbangkan karena efisiensi kerja untuk siang atau malam akan berbeda.

4) Menempatkan orang lebih banyak pada tiap pekerjaan.

Terdapat batasan praktis untuk menentukan banyaknya orang yang dapat bekerja secara serentak pada tiap pekerjaan dengan tidak saling bertabrakan satu dengan yang lain. Menurut Iman Suharto (1995), terdapat hubungan antara jumlah tenaga kerja konstruksi, luas tempat kerja dan produktivitas. Hubungan ini dinyatakan sebagai kepadatan tenaga kerja, yaitu jumlah luas tempat kerja bagi setiap tenaga kerja. Jika kepadatan ini melewati tingkat jenuh, maka produktivitas tenaga kerja akan menurun. Hal ini disebabkan di lokasi proyek terdapat sejumlah buruh bekerja, kesibukan manusia, pergerakan peralatan serta kebisingan yang ditimbulkannya. Angka kepadatan tenaga kerja juga dipengaruhi oleh faktor-faktor kompleksitas teknis dan jenis kontrak.



Gambar 3.7 Kepadatan Tenaga Kerja vs Produktivitas
(Sumber : Iman Suharto, 1995)

5) Menggunakan material dengan biaya yang lebih mahal tetapi dengan pemasangan yang lebih cepat.

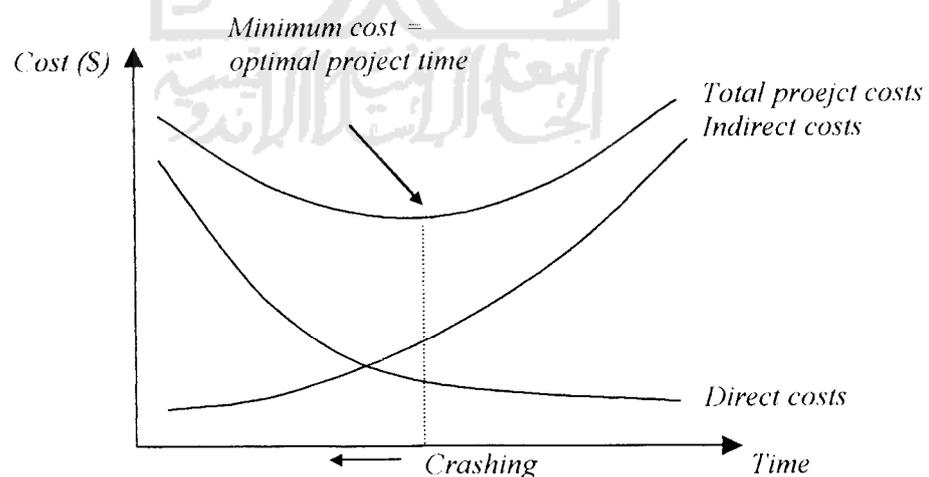
3.6 Hubungan Biaya dan Waktu

Menurut Callahan (1992) biaya total dari sebuah proyek terdiri dari biaya langsung (*“direct costs”*) dan biaya tidak langsung (*“indirect cost”*). *Direct cost* adalah biaya yang dibutuhkan oleh kegiatan-kegiatan untuk menyelesaikan pekerjaan fisik di lapangan. *Direct cost* ini meliputi biaya untuk material, tenaga kerja dan peralatan, biaya-biaya langsung tersebut dapat saling berhubungan dalam penyelesaian sebuah kegiatan. Jadi *direct cost* dari sebuah proyek adalah jumlah dari seluruh biaya langsung dari kegiatan-kegiatan.

Indirect cost adalah biaya-biaya yang dibutuhkan untuk jalannya perusahaan (kontraktor) dan biaya-biaya ini tidak memiliki hubungan langsung dengan kegiatan-kegiatan tertentu dalam proyek atau proyek-proyek tertentu. *Indirect cost* dapat dibagi menjadi dua, yaitu pengeluaran umum (*“general overhead”*) dan pengeluaran proyek (*“project overhead”*). *General overhead* adalah biaya-biaya yang

dibutuhkan dalam menjalankan perusahaan dan tidak memiliki hubungan dengan proyek tertentu, misalnya biaya operasi kantor pusat seperti : biaya utilitas kantor, biaya sewa gedung, pembelanjaan kantor dan gaji pegawai. *Project overhead* adalah biaya-biaya yang berhubungan dengan proyek tertentu tetapi tidak dapat menentukan biaya dari kegiatan, seperti biaya asuransi proyek, biaya pengawasan dan utilitas di lokasi proyek.

Biaya total proyek tersebut mempunyai hubungan dengan durasi. Dimana *direct cost* mempunyai hubungan yang terbalik dengan durasi, sedangkan *indirect cost* akan meningkat seiring dengan bertambahnya durasi proyek. Seperti halnya hubungan antara biaya total proyek dengan durasi, biaya *crash* proyek dengan *indirect cost* juga mempunyai hubungan yang terbalik, yaitu biaya *crash* akan meningkat ketika durasi proyek dipadatkan dan *indirect cost* akan meningkat ketika durasi proyek bertambah. Hubungan-hubungan antara *direct cost*, *indirect cost*, *crash cost* dan durasi proyek ditunjukkan pada gambar berikut ini :

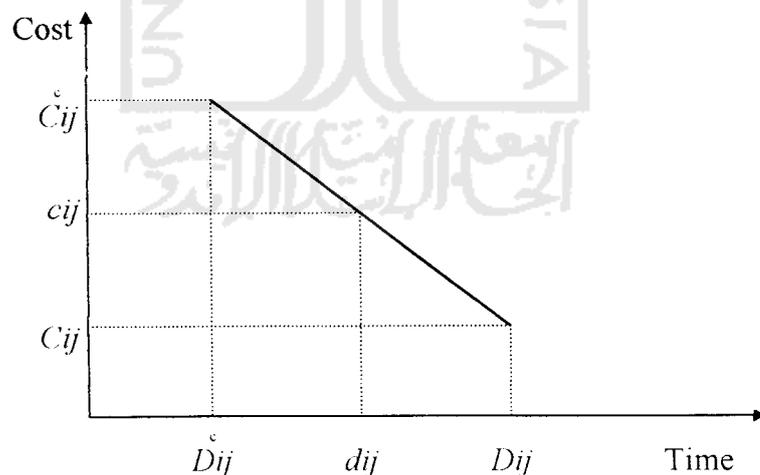


Gambar 3.8 Hubungan Antara Biaya Dan Waktu
(Sumber : Bernard W Taylor III, 1996)

3.7 Crashing dan *Time-cost trade-off*

Durasi-durasi dari kegiatan sering kali berbeda-beda tergantung pada tipe dan jumlah sumber daya yang akan digunakan. Dengan menempatkan lebih banyak tenaga kerja pada kegiatan tertentu biasanya akan menghasilkan durasi yang lebih pendek. Bagaimanapun juga durasi yang lebih cepat akan menghasilkan biaya yang lebih tinggi dan kualitas yang rendah (Chris Hendrickson dan Tung Au, 1989).

Pada Gambar 3.9 dapat dilihat hubungan antara durasi kegiatan dengan *direct costs*. Mengingat kegiatan pada gambar tersebut terpisah dan bukan merupakan referensi untuk batas waktu penyelesaian proyek, niscaya perencana akan memilih durasi yang mempunyai *direct cost* yang minimum, ditunjukkan dalam gambar oleh D_{ij} dan C_{ij} . Jika setiap kegiatan dijadwalkan untuk durasi yang menghasilkan *direct cost* yang minimal, maka waktu untuk menyelesaikan keseluruhan proyek akan menjadi sangat lama dan proyek akan terkena pinalti akibat adanya keterlambatan.



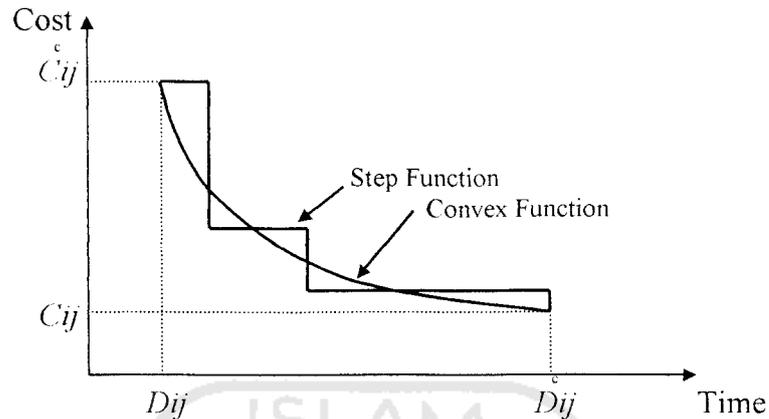
Gambar 3.9 Hubungan Linier *Time-Cost Trade-Off*

(Sumber : Chris Hendrickson dan Tung Au, 1989)

Pada kemungkinan yang lain, perencana akan memilih untuk menyelesaikan kegiatan dengan waktu yang minimum, D_{ij} , tetapi dengan biaya yang lebih tinggi C_{ij} . Waktu penyelesaian minimum ini biasa disebut dengan waktu *crash* kegiatan. Waktu minimum ini dapat ditemukan dengan mengaplikasikan CPM dengan keseluruhan durasi kegiatan diatur pada nilai yang minimum D_{ij} . Waktu penyelesaian proyek yang minimum ini digunakan dalam permasalahan waktu dan biaya untuk menentukan biaya *crash* proyek yang minimum. Perlu dicatat bahwa *crash cost* tidak akan dapat ditemukan dengan mengatur setiap kegiatan dengan durasi *crash* dan kemudian menjumlahkan biaya yang dihasilkannya, solusi semacam ini disebut *all-crash cost*.

Hubungan linier yang ditunjukkan pada Gambar 3.9 menyatakan bahwa sembarang nilai durasi diantara dua titik dapat juga dipilih. Terdapat kemungkinan bahwa beberapa nilai diantara dua titik tersebut dapat memberikan kondisi *trade-off* yang optimal antara waktu dan biaya untuk kegiatan ini. Salah satu penyebab dari kenaikan *direct cost* akibat dari pengurangan durasi adalah penggunaan tambahan waktu kerja. Penjadwalan waktu kerja pada hari libur atau malam hari, maka waktu penyelesaian dari kegiatan akan berkurang. Upah lembur harus dibayarkan untuk penambahan waktu kerja tersebut, sehingga akan mengakibatkan peningkatan biaya. Penambahan waktu kerja cenderung akan mengakibatkan terjadinya kecelakaan dan permasalahan kualitas, sehingga *indirect cost* akan meningkat. Lebih umum lagi kita mungkin tidak mengharapkan hubungan linier antara durasi dengan *direct cost*, tetapi

sebuah fungsi cembung (“*convex*”) seperti kurva non-linier dan fungsi bertahap yang terlihat dalam Gambar 3.10.



Gambar 3.10 Hubungan Non Linier *Time-Cost Trade-Off*
(Sumber : Chris Hendrickson dan Tung Au, 1989)

3.8 Metode Optimalisasi

Dalam permasalahan optimalisasi tujuannya adalah untuk memaksimalkan atau meminimumkan sebuah besaran tertentu yang disebut dengan tujuan obyektif yang bergantung pada sejumlah berhingga variabel masukan (“*input variable*”). Variabel-variabel ini dapat tidak saling bergantung atau saling bergantung melalui satu atau lebih kendala (“*constraints*”) (Bronson, 1982). Terdapat dua jenis perhitungan yang berbeda dalam permasalahan optimalisasi, yaitu perhitungan yang melibatkan simulasi dan yang berkaitan dengan model matematis (Taha, 1993).

Simulasi adalah duplikasi atau abstraksi dari persoalan-persoalan dalam kehidupan nyata ke dalam model-model matematis (Subagyo, 1990). Dalam hal ini biasanya dilakukan penyederhanaan, sehingga pemecahan dengan model-model matematis bisa dilakukan. Seringkali didalam model simulasi sudah dimasukkan unsur ketidakpastian. Dalam simulasi kita dapat selalu yakin bahwa hasil yang

diinginkan akan diperoleh dengan pasti. Masalahnya semata-mata adalah menyediakan waktu yang cukup bagi komputer. Terdapat beberapa model simulasi yang dapat dikelompokkan ke dalam beberapa penggolongan, antara lain adalah : 1) model stokastik atau probabilistik, 2) model deterministik, 3) model statik, 4) model dinamik dan 5) model heuristik.

Simulasi stokastik adalah suatu teknik yang menggunakan komputer untuk mensimulasikan tindakan-tindakan dari sebuah sistem. Sistem tersebut dapat dalam bentuk matematis atau model-model logika tentang bagaimana sebuah asumsi tertentu terbentuk (Grifis dan Farr, 2000). Didalam proses stokastik sifat-sifat keluaran ("*output*") dari proses ditentukan berdasarkan dan merupakan hasil dari konsep random (acak). Salah satu model simulasi stokastik yang telah dikenal luas adalah model simulasi Monte Carlo.

Menurut Bernard W Taylor III (1996), bilangan acak ("*random number*") memegang peranan penting dalam model simulasi probabilistik (stokastik). Jika bilangan acak tersebut tidak benar-benar acak, maka keabsahan dari hasil simulasi perlu dipertanyakan. Bilangan acak tersebut dibangkitkan dengan menggunakan teknik numerik ("*numerical technique*"). Jadi bilangan acak tersebut tidak benar-benar acak tetapi *pseudorandom number*. Bilangan acak yang sebenarnya hanya dapat diperoleh dengan proses fisik, seperti perputaran roda roulette secara berulang-ulang. Tetapi proses tersebut tidak dapat diterapkan dalam model simulasi dengan menggunakan komputer. Terdapat beberapa karakteristik dari pembangkitan bilangan secara acak, yaitu : 1) bilangan acak harus didistribusikan secara seragam, artinya setiap bilangan acak dalam interval tertentu dari bilangan acak (misalnya interval 0 sampai 1 atau 0

sampai 100) memiliki kesempatan yang sama untuk terpilih, 2) teknik numerik yang digunakan untuk membangkitkan bilangan acak haruslah efisien dan 3) urutan dari bilangan acak haruslah tidak mencerminkan sebuah pola tertentu. Saat ini sudah terdapat program-program komputer yang dapat membangkitkan bilangan acak dan dapat digunakan pada semua sistem komputer.

3.9 *Simulated Annealing* (Simulasi *Annealing*)

Simulated annealing menggunakan ide dari mekanika statistik sebagai dasar untuk pendekatan kepada bilangan bulat yang besar (*large integer*) atau permasalahan optimalisasi diskrit (Ignizio dan Cavalieer, 1994). Seperti pada algoritma genetik, penerapan dari *simulated annealing* tidak dapat memberikan garansi tercapainya solusi yang optimal, walaupun mereka mungkin menghasilkan solusi yang sangat bagus dalam waktu yang masuk akal. Seperti algoritma genetik, *simulated annealing* merupakan sebuah prosedur umum yang dapat digunakan untuk menguraikan bermacam-macam variasi kombinasi permasalahan optimalisasi. Pada akhirnya, *simulated annealing* relatif mudah untuk diprogram dan digunakan. Kombinasi dari faktor-faktor ini, yaitu solusi yang bagus untuk permasalahan yang sulit, aplikasi umum, dan mudah dipergunakan-membuat *simulated annealing* menjadi populer dikalangan praktisi diberbagai bidang.

Dalam proses pepadatan logam, *annealing* merupakan proses fisik dimana logam dalam *heat bath* (tungku) dipanaskan dengan menaikkan temperatur dari *heat bath* sampai pada temperatur maksimum, dimana semua partikel-partikel penyusun logam secara acak menyusun diri mereka pada tahap cair, kemudian diikuti dengan pendinginan dengan penurunan temperatur secara lambat dari *heat bath*. Dalam hal

ini, dengan menyediakan temperatur maksimum cukup tinggi dan pendinginan yang cukup lambat sehingga semua partikel-partikel logam dapat menyusun diri mereka sendiri pada keadaan energi terendah (“*ground state*”) dari atom-atom. Bagaimanapun, telah diketahui (Kirkpatrick, 1982) bahwa jika pendinginan terlalu cepat, maka kerusakan atau cacat akan “membeku” didalam logam dan struktur logam yang tidak berberntuk serta memiliki energi yang tinggi (“*metastable amorphous structure*”) akan tercapai daripada struktur logam dengan pola-pola geometris berbentuk kristal (“*crystalline lattice structure*”) dan memiliki energi yang rendah. Dalam proses pepadatan logam yang dikenal sebagai *quenching*, temperatur dari *heat bath* diturunkan seketika itu juga, sehingga menghasilkan pembekuan partikel-partikel logam menjadi *metastable amorphous structures* (Laarhoven dan Aarts, 1987).

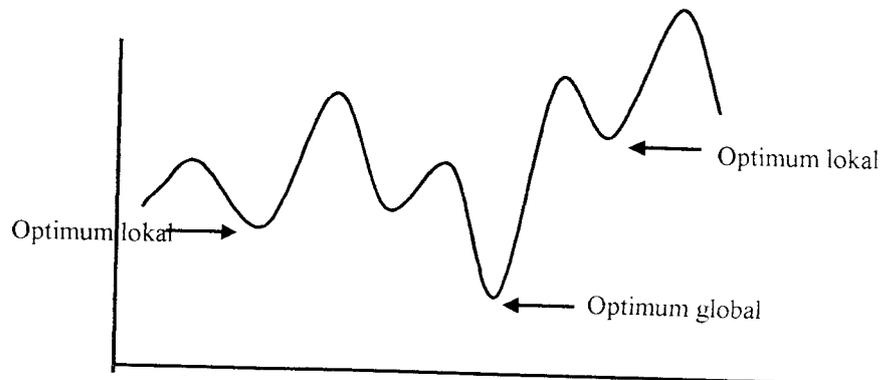
Algoritma proses *annealing* pertamakali dirumuskan oleh Metropolis dkk. pada tahun 1953, sehingga disebut dengan Algoritma Metropolis (Laarhoven dan Aarts, 1987). Hampir 30 tahun kemudian, kelompok lain dari fisikawan bekerja pada permasalahan optimalisasi diskrit untuk rancangan VLSI, mulai menyelidiki persamaan antara permasalahan optimalisasi diskrit ini dan simulasi *annealing* pada logam. Mereka mengamati hubungan-hubungan sebagai berikut :

Simulated annealing => *Discrete optimization procedure*

Ground state => *Global optimum*

Metastable states => *Local optima*

Energy => *Cost*



Gambar 3.11 Optimum Global Dan Optimum Lokal

Mereka mengeksploitasi hubungan-hubungan ini untuk mengembangkan pendekatan *simulated annealing* pada optimalisasi diskrit (Kirkpatrick, Gelatt, and Vecchi, 1983). Sejak pengenalan ini, jumlah aplikasi *simulated annealing* bertambah banyak dan termasuk *image processing* (Geman dan Geman, 1984), *computer-aided design* (Fleisher, Tavel, dan Martin, 1985), dan *clustering* (pengelompokan) (Brown dan huntley, 1992). Dalam bidang teknik sipil, metode optimalisasi dengan menggunakan *simulated annealing* dipergunakan pertama kali dalam permasalahan optimalisasi diskrit rangka baja. Seperti yang dilakukan oleh Reinschmidt (1971), Liebman et al. (1981), Hua (1983), Grieson and Lee (1986) dan Balling and Fonesca (1989). Mereka melakukan optimalisasi diskrit pada rangka baja dengan tujuan untuk memilih profil yang optimum untuk bermacam-macam girder dan kolom dalam struktur rangka baja.

3.9.1 Konsep Dasar Program “*Anneal5.EXE*”

Program *Anneal5.EXE* adalah sebuah program komputer yang berfungsi untuk mengoptimalkan durasi dan biaya proyek akibat dilakukannya percepatan durasi kegiatan dalam proyek menggunakan metode optimalisasi *simulated annealing*. Program “*Anneal5.EXE*” ini disusun dan dibuat oleh Hadiwidodo

berdasarkan algoritma *simulated annealing* yang dikembangkan oleh Balling (1991) dan P. J. M. van Laarhoven dan E. H. L. Aarts (1987). Program “*Anneal5.EXE*” ini disusun dengan menggunakan bahasa pemrograman komputer, yaitu bahasa C dan dikompilasi dengan menggunakan *compiler* Borland C/C++ versi 3.0.

Konsep dasar probabilitas penerimaan pada metode *simulated annealing* dalam program “*Anneal5.EXE*” ini didasarkan pada algoritma proses *annealing* yang dirumuskan oleh Metropolis dkk. pada tahun 1953 (Hadiwidodo, 2001). Algoritma ini mengasumsikan pada nilai temperatur T tertentu, posisi satu atau beberapa atom terpilih acak dirubah dan perubahan energi ΔE pada sistem akibat perubahan tersebut dihitung. Apabila tingkat energi baru (setelah atom-atom penyusun berada dalam keadaan terganggu “*perturbed*”) lebih rendah dari tingkat energi lama, konfigurasi atom yang baru diterima. Tetapi bila tingkat energi lama ($\Delta E \geq 0$), dilakukan analisis probabilitas untuk memutuskan apakah konfigurasi atom yang baru diterima atau ditolak. Persamaan probabilitas untuk menentukan kriteria penerimaan tersebut dirumuskan sebagai :

$$P = \exp(-\Delta E / K_B \cdot T) \quad (3.1)$$

dimana K_B adalah konstanta Boltzman. P memiliki interval antara 0 sampai 1. Tampak bahwa nilai probabilitas dipengaruhi oleh temperatur sistem T . Pada keadaan dimana temperatur sistem tinggi atau pada tahap awal proses *annealing*, probabilitas untuk penerimaan akan lebih besar dibanding pada temperatur rendah atau pada tahap akhir proses *annealing*. Untuk memutuskan apakah keadaan baru diterima atau ditolak, dipilih suatu nilai secara acak dengan interval 0 sampai 1 yang kemudian dibandingkan dengan nilai P . Bila bilangan acak yang dibangkitkan lebih kecil dari P

maka konfigurasi baru diterima. Bila lebih besar atau sama maka konfigurasi baru ditolak.

Proses perubahan konfigurasi atom dan analisis probabilitas tersebut diulangi untuk seluruh atom sistem, sampai tingkat energi yang tetap pada nilai temperatur T tercapai atau disebut sebagai keadaan *thermal equilibrium*. Kemudian nilai temperatur T diturunkan dan keseluruhan proses diulangi. Proses iterasi akan dihentikan bila keadaan *thermal equilibrium* satu nilai temperatur dan nilai temperatur yang lebih rendah mencapai konvergensi, yaitu keadaan dimana tidak dijumpai adanya perbedaan yang signifikan antara nilai fungsi objektif keadaan baru dengan nilai fungsi objektif keadaan lama. Pada keadaan tersebut sistem diasumsikan telah berada pada kondisi optimum global dengan tingkat energi minimal dan stabil.

Penerapan algoritma Metropolis pada masalah optimalisasi dengan banyak variabel disain dikembangkan oleh Kirkpatrick dkk. Pada tahun 1982 (Balling, 1991). Penerapannya dilakukan dengan mengasumsikan tingkat energi atom sebagai fungsi objektif dan konfigurasi atom sebagai variabel disain. Temperatur menjadi parameter kontrol yang mengatur tercapainya konvergensi. Proses iterasi diawali dengan perubahan konfigurasi variabel disain pada nilai temperatur tertentu dengan mengacak nilai salah satu variabel disain sehingga nilai fungsi objektif berubah. Kriteria penerimaan konfigurasi baru tersebut tergantung pada selisih antara nilai fungsi objektif awal dengan nilai fungsi objektif baru. Konfigurasi yang menghasilkan nilai fungsi objektif baru lebih kecil atau lebih baik dari nilai fungsi objektif awal diterima menjadi konfigurasi awal bagi proses iterasi berikutnya. Tetapi bila konfigurasi baru memiliki nilai fungsi objektif lebih besar atau lebih buruk dari

nilai fungsi objektif konfigurasi awal, dilakukan analisis probabilitas dengan perumusan sebagai berikut :

$$P = \exp(-\Delta E / K_B.T) \quad (3.2)$$

dimana ΔE adalah selisih fungsi objektif antara konfigurasi awal dengan konfigurasi baru, K_B adalah konstanta Boltzman dan T adalah temperatur sistem. Nilai P dibandingkan dengan sebuah bilangan acak yang dibangkitkan pada interval nol sampai satu. Bila bilangan acak yang dibangkitkan lebih kecil dari P maka konfigurasi diterima sebagai konfigurasi sistem, sedangkan bila lebih besar atau sama maka konfigurasi baru ditolak dan konfigurasi awal tetap digunakan. Selanjutnya dipilih variabel disain lain secara acak dan proses tersebut diulangi untuk seluruh variabel disain.

Setelah konfigurasi baru pada satu nilai temperatur T dirubah, temperatur diturunkan dan dilakukan perubahan konfigurasi. Proses iterasi tersebut dilakukan sampai tercapai konvergensi, yaitu pada keadaan dimana setelah sejumlah tertentu iterasi, nilai fungsi objektif pada temperatur $T+1$ tidak lebih baik dibanding nilai fungsi objektif pada temperatur T . Menurut Laarhoven dan Aarts (1987) ada empat parameter yang disebut *cooling schedule* akan menentukan kinerja algoritma.

Keempat parameter *cooling schedule* adalah sebagai berikut :

- 1) Penentuan temperatur awal dan metoda untuk memperbarui nilai temperatur

Temperatur awal harus ditentukan sedemikian rupa supaya probabilitas penerimaan pada tahap awal optimalisasi relatif besar (mendekati 1). Kirkpatrick dkk. (1982) merumuskan metoda untuk memperbarui nilai temperatur.

$$T_{k+1} = \alpha.T_k, \quad k = 0,1,2,\dots; \quad 0,5 \leq \alpha \leq 0,99 \quad (3.3)$$

2) Kriteria untuk mengakhiri proses optimalisasi

Umumnya proses optimalisasi dianggap selesai bila perubahan pada nilai fungsi objektif terlalu kecil atau bila suatu temperatur yang ditetapkan sebagai temperatur akhir telah tercapai.

3) Jumlah iterasi yang diperlukan untuk mencapai thermal equilibrium pada suatu nilai temperatur. Pada beberapa kasus terkadang tidak dimungkinkan tercapainya konvergensi pada suatu nilai temperatur tertentu (jumlah iterasi = ∞) sehingga perlu diterapkan suatu nilai maksimum jumlah iterasi yang bisa dilakukan oleh sistem. Nilai ini harus cukup besar untuk menghindarkan terjadinya konvergensi prematur yang akan mengarah pada pencapaian optimum lokal.

4) Metoda untuk merubah disain awal ke disain terganggu

Metode ini akan mempengaruhi kemampuan sistem mencapai *thermal equilibrium* pada suatu nilai temperatur. Balling (1991) merubah disain asal dengan mengganti nilai seluruh variabel disain satu persatu secara acak dengan nilai baru yang juga dipilih secara acak.

Pada penyusunan *cooling schedule* dikenal adanya kondisi *quasi-equilibrium*, yaitu kondisi dimana sistem mencapai *thermal equilibrium* pada nilai temperatur tertentu. Pencapaian *quasi-equilibrium* dilakukan dengan melakukan iterasi pada temperatur tertentu sebanyak maksimal M kali sampai nilai fungsi objektif relatif tidak berubah. Adanya nilai M bagi pembatasan jumlah iterasi adalah untuk menghindarkan *osilasi*, yaitu terjebaknya algoritma disekitar suatu nilai fungsi objektif tertentu. Pada Gambar 3.12 ditunjukkan diagram alir algoritma *simulated annealing*. Parameter-parameter yang dipakai pada algoritma tersebut adalah : i

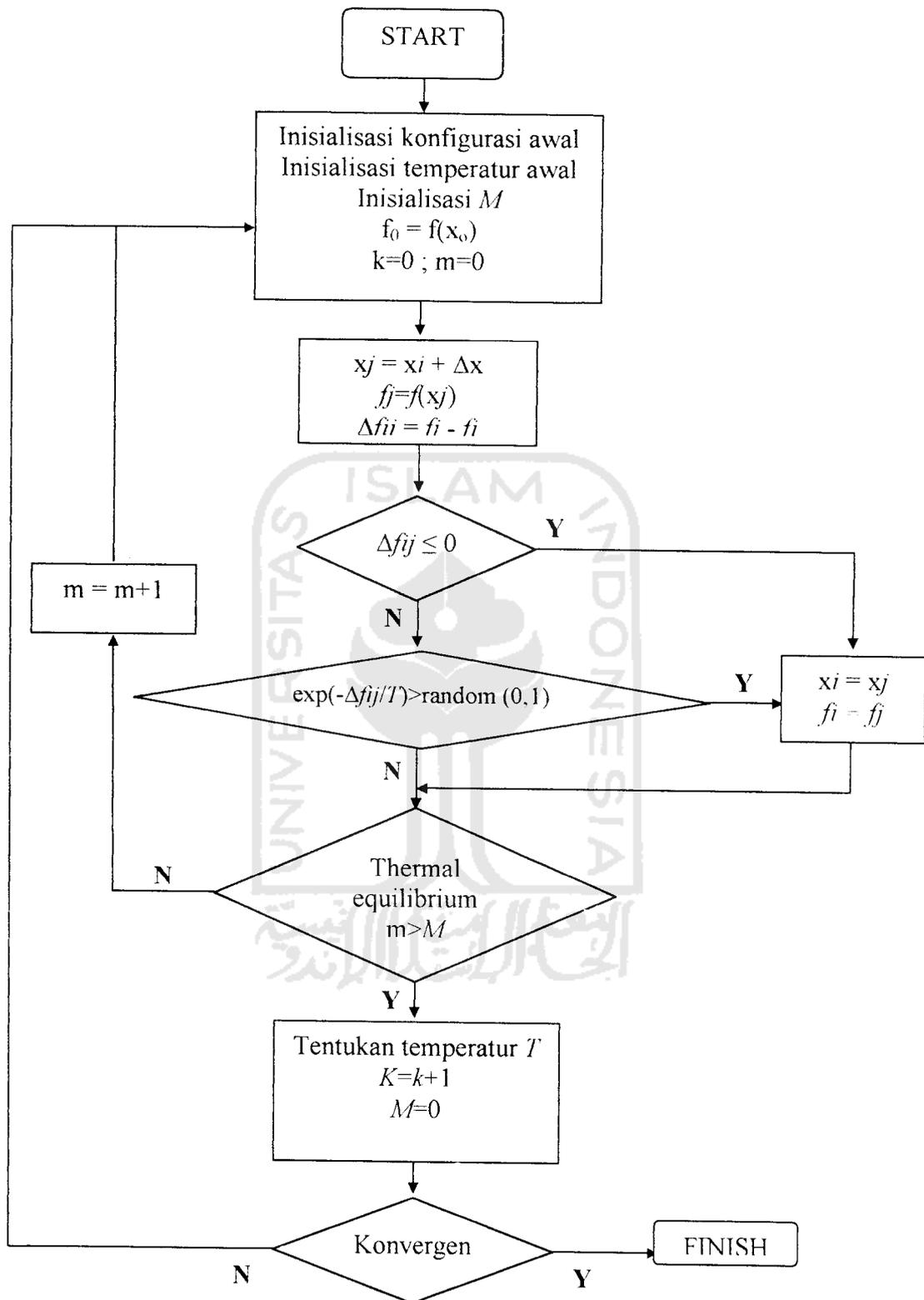
sebagai konfigurasi awal, j sebagai konfigurasi akhir, T sebagai temperatur sistem, M sebagai kriteria jumlah iterasi untuk mencari konfigurasi j dari konfigurasi i , atau untuk mencapai kondisi *quasi-equilibrium*, f sebagai nilai fungsi objektif yang diperoleh berdasarkan konfigurasi tertentu dan X sebagai nilai variabel disain pada konfigurasi tertentu.

Pada penerapan masalah optimalisasi untuk masalah tertentu dibutuhkan perumusan probabilitas penerimaan yang spesifik. Pada studi ini akan digunakan perumusan probabilitas penerimaan dari Balling (1991) sebagai berikut :

$$P = \exp(-D/CT) \quad (3.4)$$

dimana D adalah nilai fungsi objektif disain kandidat fungsi objektif disain sebelumnya, C adalah konstanta normalisasi dan T adalah temperatur sistem. Kemudian sebuah bilangan acak $X(0,1)$ dibangkitkan dan dibandingkan dengan nilai P . Apabila $X < P$, disain kandidat diterima sebagai disain awal untuk iterasi berikutnya.

Persamaan 3.4 tersebut diturunkan dari persamaan pada algoritma Metropolis dimana D identik dengan peningkatan energi sistem dari keadaan awal ke keadaan "terganggu", C identik dengan konstanta Boltzman dan T identik dengan temperatur sistem. Pada persamaan 3.4 konstanta normalisasi C diperoleh rerata berjalan ("*running average*") D , dengan variabel M digunakan untuk memperbarui nilai C .



Gambar 3.12 Diagram Alir Algoritma *Simulated Annealing*
(Sumber : Hadiwidodo, 2001)

Pada awal algoritma, nilai C dan M diatas diinisialisasi dengan nilai masing-masing 1 dan 0. Setiap kali dihasilkan disain kandidat yang lebih buruk dari disain sebelumnya, nilai C dan M diperbarui dengan persamaan berikut :

$$C = M \times C + D/M+1 \quad (3.5)$$

$$M = M+1 \quad (3.6)$$

Selanjutnya nilai baru tersebut digunakan untuk melaksanakan analisa probabilitas dengan persamaan (3.4). Parameter-parameter lain yang harus didefinisikan pada awal algoritma adalah :

- a. P_s , yaitu probabilitas penerimaan awal untuk rata-rata $D=C$
- b. P_f , yaitu probabilitas penerimaan akhir untuk rata-rata $D=C$
- c. N , yaitu jumlah iterasi keseluruhan (seandainya terjadi kasus dimana sistem tidak bisa mencapai konvergensi dan/atau ber-*osilasi* disekitar nilai fungsi objektif tertentu, kriteria ini akan menentukan penghentian iterasi)

Apabila diasumsikan nilai probabilitas penerimaan pada awal optimalisasi sebesar P_s , maka nilai temperatur awal T ditentukan dengan persamaan berikut :

$$T = -1 / \log(P_s) \quad (3.7)$$

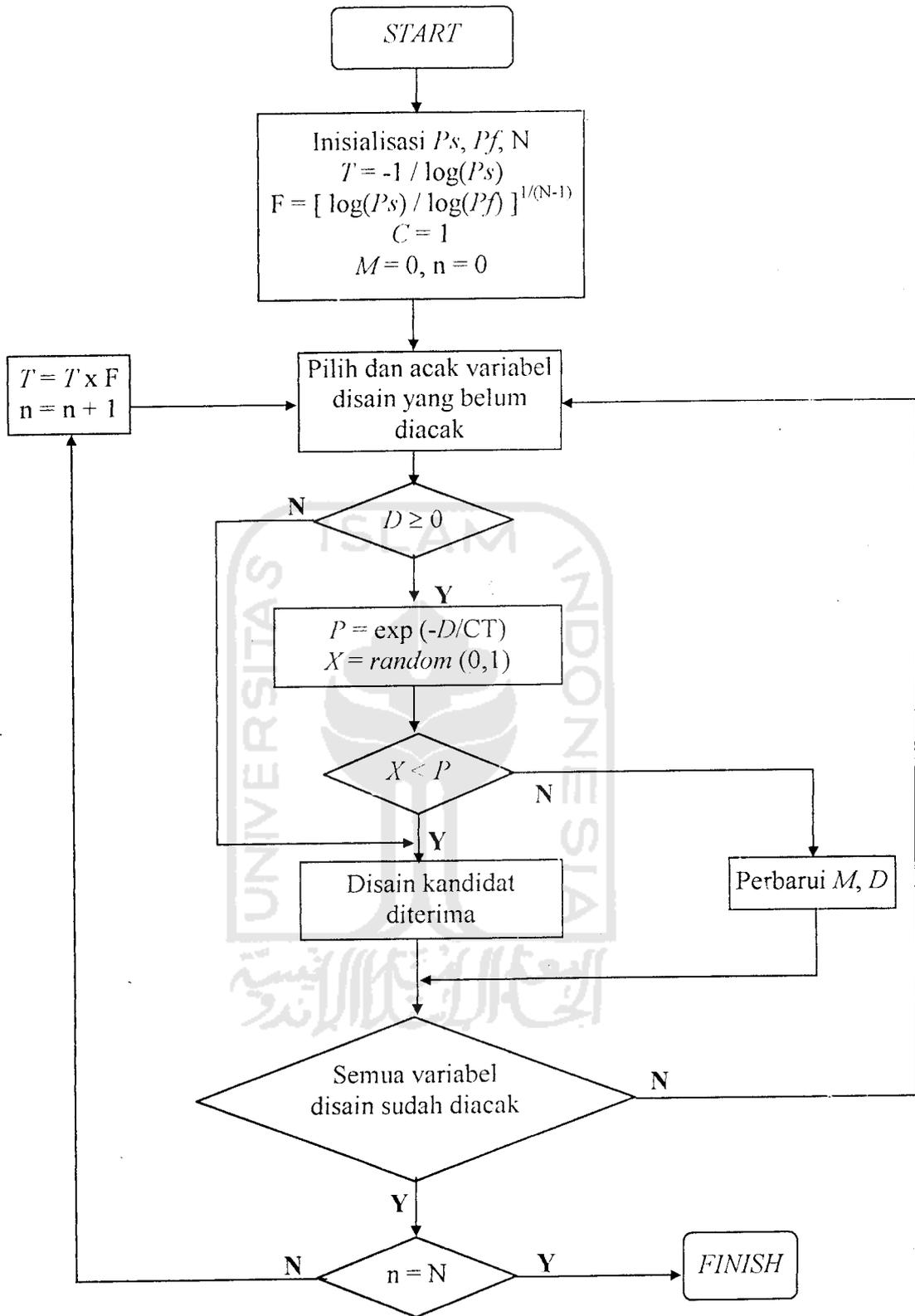
Nilai T diperbarui dengan menggunakan faktor pendingin (*cooling factor*) F yang dirumuskan sebagai berikut :

$$T = T \times F \quad (3.8)$$

Dengan nilai F dirumuskan sebagai berikut :

$$F = [\log(P_s) / \log(P_f)]^{1/(N-1)} \quad (3.9)$$

Algoritma *simulated annealing* dari Balling (1991) tampak pada Gambar 3.13.



Gambar 3.13 Diagram Alir Algoritma *Simulated Annealing* Menurut Balling (1991)
(Sumber : Hadiwidodo, 2001)

Menurut Hadiwidodo (2001) pada algoritma pemrograman menggunakan metode optimalisasi yang disusun oleh Balling (1991) yang telah diuraikan diatas terdapat beberapa perubahan atau penambahan yang dilakukan antara lain sebagai berikut :

- a. Kriteria untuk mencegah terjadinya *osilasi*, yaitu proses iterasi berulang-ulang yang menghasilkan nilai fungsi objektif yang secara signifikan tidak berubah. Hal ini akan terjadi pada tahap akhir optimalisasi dimana kriteria probabilitas penerimaan sudah sedemikian kecil (mendekati probabilitas akhir P_f) sehingga penerimaan terhadap nilai fungsi objektif yang lebih buruk dari fungsi objektif sebelumnya hampir tidak diperkenankan. Pada permasalahan optimalisasi biaya-waktu ini, iterasi akan dihentikan bila disain yang sama telah diacak sebanyak kuadrat jumlah iterasi, dan selanjutnya iterasi diteruskan dengan menggunakan disain lama serta nilai temperatur baru.
- b. Perhitungan fungsi kendala pada algoritma dari Balling yang diterapkan untuk optimalisasi struktur rangka batang, perhitungan fungsi kendala dilakukan segera setelah satu elemen diacak. Sedangkan pada algoritma yang disusun untuk optimalisasi biaya-waktu ini, fungsi kendala dihitung setelah seluruh elemen diacak. Sehingga kombinasi variabel disain yang memenuhi diharapkan dapat lebih bervariasi.

3.9.2 Analogi Metode *Simulated Annealing* Pada Permasalahan Optimalisasi Waktu dan Biaya Proyek

Pencarian solusi yang paling optimal pada proses pembentukan logam dengan metode *simulated annealing* bertujuan untuk mendapatkan energi yang paling kecil

dari logam. Optimalisasi dilakukan dengan memberikan rentang nilai temperatur yang lebar dan penurunan temperatur secara bertahap (pelan-pelan) serta pada setiap temperatur tertentu harus didapatkan sebuah solusi energi yang minimum. Dibawah ini adalah asumsi-asumsi hubungan antara *simulated annealing* dengan permasalahan optimalisasi durasi dan biaya proyek :

Simulated annealing => Proses optimalisasi durasi dan biaya proyek

Logam => Proyek

Konfigurasi => Diagram jaringan CPM

Partikel => Durasi dan Biaya Kegiatan

Pendinginan => Pengurangan durasi proyek secara bertahap

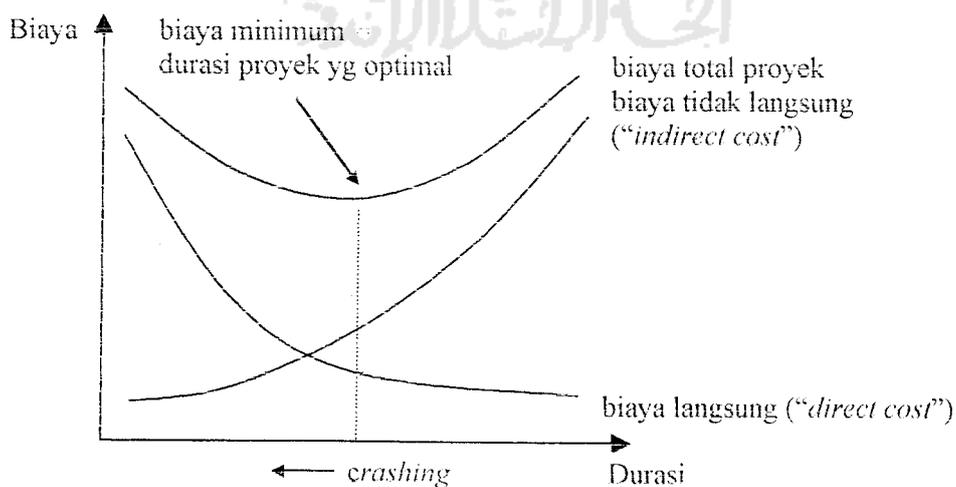
Temperatur => Durasi proyek

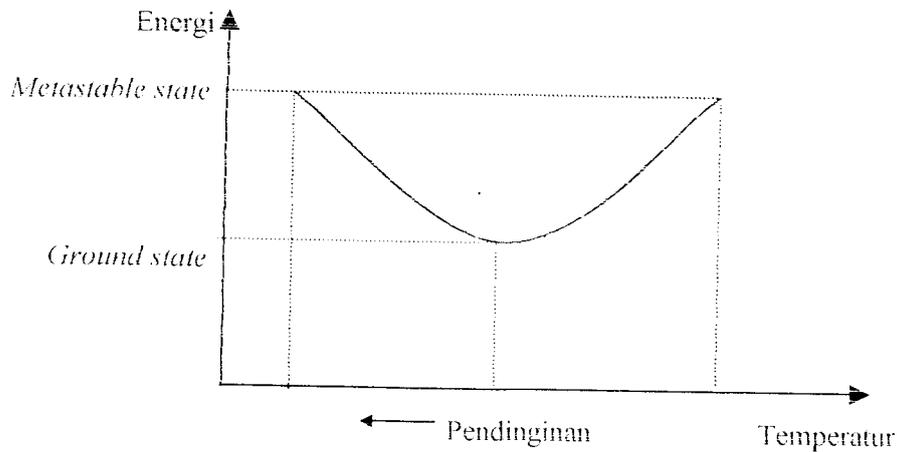
Energi => Biaya total proyek

Ground state => Biaya total proyek terendah

Metastable state => Biaya total proyek tertinggi

Hubungan antara temperatur dengan durasi dan energi dengan biaya proyek, dapat dilihat pada gambar berikut ini :





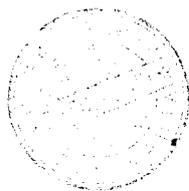
Gambar 3.14 Hubungan Proyek Konstruksi Dengan *Simulated Annealing*

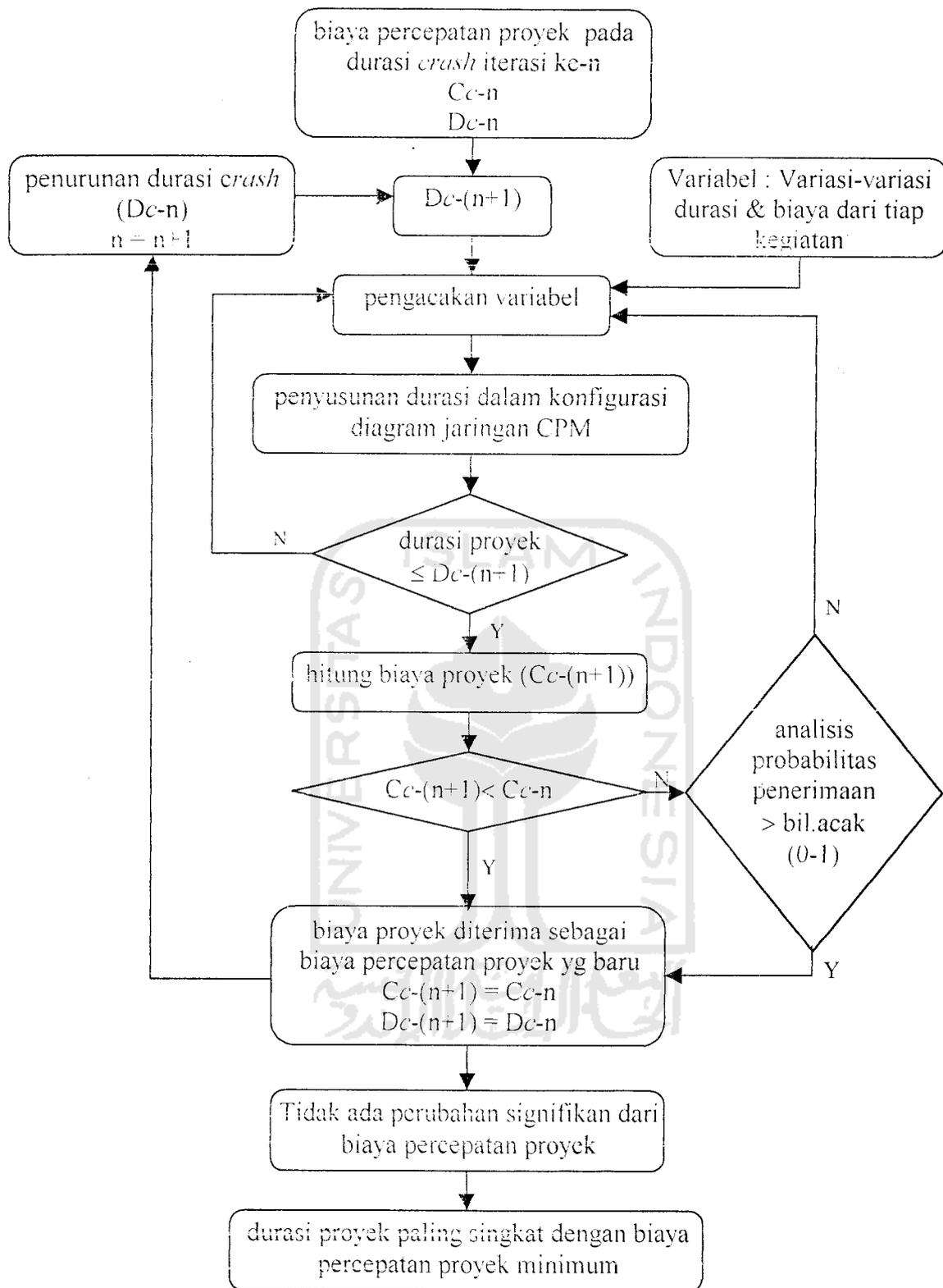
Proses optimalisasi durasi dan biaya proyek menggunakan metode *simulated annealing* bertujuan untuk mendapatkan peningkatan biaya proyek paling kecil akibat adanya percepatan durasi proyek. Optimalisasi dilakukan dengan memberikan rentang yang lebar antara durasi proyek yang cukup besar (durasi normal = temperatur maksimum) dengan durasi proyek minimum (durasi *crash* = temperatur minimum). Dengan pemberian rentang durasi tersebut, maka terdapat kemungkinan bahwa beberapa nilai diantara dua titik durasi tersebut dapat memberikan kondisi *trade-off* yang optimal antara durasi proyek paling singkat dengan peningkatan biaya proyek yang minimum. Agar kondisi optimal tersebut dapat tercapai maka dibutuhkan variabel-variabel masukan (variabel disain) yang berupa variasi-variasi durasi dan biaya percepatan dari tiap-tiap kegiatan dalam proyek. Dalam penelitian ini, variasi-variasi tersebut didapatkan dengan mempercepat durasi kegiatan dengan 5 macam metode pemadatan.

Pada analogi ini, proses pencarian peningkatan biaya proyek yang minimum akibat percepatan durasi proyek dilakukan dengan menurunkan durasi proyek secara bertahap. Proses ini dilakukan dengan cara iterasi, dimana pada setiap nilai penurunan

durasi (durasi *crash* ($Dc-n$)) pada iterasi ke- n proyek harus didapatkan sebuah solusi peningkatan biaya proyek yang minimal ($Cc-n$). Untuk mendapatkan durasi *crash* dan biaya proyek tersebut dilakukan dengan melakukan pengacakan variabel disain, kemudian variabel-variabel yang berupa durasi kegiatan disusun dalam konfigurasi diagram jaringan CPM. Jika menghasilkan durasi proyek yang lebih singkat atau sama dengan nilai durasi *crash* proyek, kemudian biaya proyek dihitung dan jika menghasilkan durasi proyek yang lebih besar dilakukan pengacakan ulang variabel disain. Kemudian jika biaya proyek yang dihasilkan pada $Dc-n$ lebih baik dari biaya proyek pada Dc sebelumnya, maka biaya proyek ($Cc-n$) diterima sebagai solusi biaya percepatan proyek yang baru. Jika sebaliknya lebih buruk dari durasi sebelumnya, maka untuk menentukan biaya proyek tersebut diterima atau tidak dilakukan dengan analisis probabilitas.

Jika pada $Dc-n$ tertentu didapatkan sebuah solusi biaya proyek yang bagus, kemudian proses optimalisasi dilanjutkan dengan menurunkan durasi secara bertahap ($Dc-(n+1)$). Proses pencarian dilakukan sampai tidak ada perubahan yang signifikan antara biaya percepatan proyek satu ($Cc-n$) dengan biaya percepatan proyek yang baru ($Cc-(n+1)$). Gambaran analogi metode *simulated annealing* pada permasalahan optimalisasi durasi dan biaya proyek dapat dilihat pada diagram berikut ini :





Gambar 3.15 Diagram Alir Analogi Proses *Simulated Annealing* Pada Proses Optimalisasi Durasi dan Biaya Proyek Konstruksi