

BAB II

TINJAUAN PUSTAKA

Pada Bab ini menjelaskan landasan teori yang mendukung pembahasan dan berguna dalam menganalisis dan mengolah data. Tinjauan pustaka bersumber dari buku, jurnal ilmiah, internet, penelitian, dan sumber-sumber yang lainnya.

2.1 Penelitian Terdahulu

Hasil penelitian terdahulu mengemukakan beberapa konsep yang relevan dan dijadikan acuan yang dijadikan peneliti dalam penelitian ini. Berikut merupakan penelitian terdahulu yang dijadikan acuan peneliti.

1. Wang et al (2013) melakukan penelitian yang membahas *travelling salesman problem* (TSP) sebagai salah satu masalah NP (*non deterministic polynomial time*)-complete. Algoritma *simulated annealing* dasar (BSA) ditingkatkan menjadi *improved simulated annealing* (ISA) dengan empat kota (*vertices*) dan tiga lini yang tidak setara untuk mencari rangkaian Hamiltonian yang optimal (OHC). Empat kota dan tiga lini yang tidak setara diambil sebagai informasi untuk mengubah jalur lokal Hamiltonian di rangkaian Hamiltonian (HC) ke jalur Hamiltonian lokal yang optimal dan HC menjadi lebih pendek.
2. Mendes et al (2005) melakukan penelitian yang menerapkan metode heuristik untuk jenis lintasan *mixed model assembly line*, yang berfungsi untuk meminimalkan jumlah stasiun kerja sesuai dengan scenario level tingkat permintaan konsumen (*demand*) dan untuk menyeimbangkan beban kerja pada setiap stasiun kerja dan antar stasiun kerja. Dalam penelitian ini digunakan metode heuristik dengan menggunakan *ranked positional weight, simulated annealing* untuk mengevaluasi hasil dari metode heuristik dan kemudian hasil metode heuristik dan kondisi aktual disimulasikan dengan *software* simulasi ARENA untuk melihat perbandingan antara kondisi aktual dan hasil dari metode heuristik.
3. Susanto (2012) melakukan penelitian yang membahas mengenai model lini perakitan yang dapat sekaligus merakit beberapa model produk yang memiliki kesamaan *part* yang biasa disebut dengan *mixed-model assembly line*. Maka pendekatan analitikal dilakukan dengan cara menggunakan pendekatan secara heuristik untuk menyelesaikan permasalahan *mixed-model assembly line balancing* dan menghasilkan

konfigurasi lini dengan jumlah stasiun kerja yang minimum dan beban kerja yang seimbang antar stasiun kerja. Selanjutnya hasil konfigurasi lini akan digunakan sebagai masukan pada model *dicrete event simulation* untuk mengukur total waktu dan utilitas tiap stasiun kerja dari solusi yang ada sehingga didapatkan hasil skenario memiliki hasil total waktu dan utilitas tiap stasiun kerja yang lebih baik daripada kondisi eksisting.

Tabel 2.1
Perbandingan Penelitian Terdahulu dengan Penelitian yang Digunakan

Peneliti	Tahun	Metode	Hasil Penelitian
Wang et al	2013	<i>Travelling salesman problem, annealing simulated</i>	Metode ISA menunjukkan performansi lebih baik daripada BSA untuk TSP. Penulis menuliska bahwa performansi ISA dipengaruhi posisi, waktu eksekusi, dan pemilihan parameter <i>simulated annealing</i> .
Mendes et al	2005	<i>Ranked positional weight, heuristic procedure, ARENA simulation</i>	Hasil menunjukan keuntungan bagi manajer lini untuk mengetahui konfigurasi lini pada setiap skenario tingkatan level <i>demand</i> yang digambarkan melalui simulasi jalannya produksi melalui ARENA.
Susanto	2014	<i>Ranked positional weight, simulated annealing, ARENA simulation</i>	Hasil waktu dari simulasi menunjukan jumlah lebih sedikit dari kondisi eksisting dengan tingkat utilisasi yang lebih besar dan jumlah stasiun kerja lebih sedikit.
Penelitian saat ini	2017	<i>Ranked positional weight, simulated annealing, MATLAB</i>	Hasil optimal dalam jumlah stasiun kerja dengan nilai <i>cycle time</i> tertentu serta meminimalkan beban kerja antara dan di dalam stasiun kerja.

Dari keterangan yang disajikan pada Tabel 2.1, penelitian ini melakukan analisa keseimbangan lintasan dengan jenis *mixed model line balancing* menggunakan metode heuristik yaitu *ranked positional weight* (RPW) dan metode metaheuristik yaitu *simulated annealing* dengan menggunakan *software* MATLAB. *Mixed model line balancing* digunakan karena penelitian ini memiliki lebih dari satu ragam produk dengan kebutuhan aktivitas berbeda-beda. Pada *ranked positional weight* (RPW), data yang diperlukan adalah data aktivitas pada lapangan serta waktu aktivitas yang didapatkan dengan metode *stopwatch time study* dan hasilnya merupakan pengurutan dari bobot posisi yang menggunakan waktu aktivitas yang telah disesuaikan dengan proporsi produk-produk yang di proses dalam lintasan produksi. Hasil dari *ranked positional weight* (RPW) dijadikan sebagai solusi awal *simulated annealing* yang akan dilakukan dengan *software* MATLAB, karena berdasarkan penelitian terdahulu metode ini dapat memberikan hasil optimal dalam jumlah stasiun kerja dengan nilai *cycle time* tertentu serta meminimalkan beban kerja antara dan di dalam stasiun kerja.

2.2 *Line Balancing*

Menurut Gaspersz (2004:142) *line balancing* atau keseimbangan lintasan merupakan penyeimbangan penugasan elemen-elemen aktivitas dari suatu lintasan ke stasiun kerja untuk meminimumkan banyaknya stasiun kerja dan meminimumkan total harga *idle time* pada semua stasiun untuk tingkat *output* tertentu. Dalam penyeimbangan aktivitas ini, kebutuhan waktu per unit produk yang dispesifikasikan untuk setiap aktivitas dan hubungan sekuensial harus dipertimbangkan. Prinsip dasar dari suatu lintasan produksi adalah pergerakan atau aliran dari suatu benda kerja dari seorang pekerja kepada pekerja lainnya. Atau dengan kata lain, merupakan rangkaian dari urutan proses pengerjaan yang diperlukan untuk memproduksi suatu produk. Dengan demikian, beberapa pekerjaan yang harus dilakukan untuk menyelesaikan satu unit produk yang dibagi menjadi beberapa stasiun kerja di sepanjang lintasan produksi. Artinya, seorang pekerja melakukan pekerjaan yang sama pada setiap benda kerja yang melewatinya.

Fungsi dari keseimbangan lintasan adalah membuat suatu lintasan yang seimbang. Menurut Baroto (2002:192) tujuan pokok dari penyeimbangan lintasan adalah meminimumkan waktu menganggur (*idle time*) pada lintasan yang ditentukan oleh operasi yang paling lambat. Selain itu, tujuan yang ingin dicapai adalah tercapainya target produksi yang diharapkan. Tujuan tersebut dapat dicapai dengan membuat waktu siklus setiap stasiun kerja yang ada tidak melebihi waktu siklus yang sudah ditetapkan. Untuk dapat menyelesaikan masalah penyeimbangan lintasan (*line balancing*) maka harus diketahui terlebih dahulu metode kerja, mesin atau peralatan yang digunakan, serta informasi waktu yang dibutuhkan untuk setiap lintasan kerja.

Manajemen industri dalam menyelesaikan masalah harus mengetahui tentang metode kerja, peralatan-peralatan, mesin-mesin, dan personil yang digunakan dalam proses kerja. Data yang diperlukan adalah informasi tentang waktu yang dibutuhkan untuk setiap lintasan dan *precedence relationship*. Aktivitas-aktivitas yang merupakan susunan dan urutan dari berbagai aktivitas yang perlu dilakukan, manajemen industri perlu menetapkan tingkat produksi per hari yang disesuaikan dengan tingkat permintaan total, kemudian membaginya ke dalam waktu produktif yang tersedia per hari. Hasil ini adalah *cycle time* yang merupakan waktu dari produk yang tersedia pada setiap stasiun kerja (Baroto, 2002:192).

Hubungan atau saling keterkaitan antara satu pekerjaan dengan pekerjaan lainnya digambarkan dalam suatu diagram yang disebut *precedence diagram* atau diagram pendahuluan. Dalam suatu perusahaan yang memiliki tipe produksi massal, yang

melibatkan sejumlah besar komponen yang harus dirakit, perencanaan produksi memegang peranan yang penting dalam membuat penjadwalan produksi (*production schedule*) terutama dalam masalah pengaturan operasi-operasi atau penugasan kerja yang harus dilakukan. Keseimbangan lintasan sangat penting karena menentukan aspek-aspek lain dalam sistem produksi dalam jangka waktu yang cukup lama. Beberapa aspek yang terpengaruh antara lain biaya, keuntungan, tenaga kerja, peralatan, dan sebagainya.

Keseimbangan lintasan ini digunakan untuk mendapatkan lintasan perakitan yang memenuhi tingkat produksi tertentu. Demikian penyeimbangan lini harus dilakukan dengan metode yang tepat sehingga menghasilkan keluaran berupa keseimbangan lintasan yang terbaik. Tujuan akhir pada keseimbangan lintasan adalah memaksimalkan kecepatan di tiap stasiun kerja sehingga dicapai efisiensi kerja yang tinggi di tiap stasiun (Kusuma, 1999:38).

Terdapat dua tipe permasalahan yang terjadi dalam lintasan perakitan, yaitu: (Bedworth & Bailey, 1987:363)

1. Diketahui waktu siklus yang diharapkan dalam suatu lintasan untuk menentukan jumlah stasiun kerja atau pekerja minimal dalam melakukan pekerjaan perakitan berdasarkan urutan pengerjaan dan waktu operasi.
2. Diketahui jumlah pekerja atau stasiun kerja yang tersedia untuk mencari waktu siklus minimum yang bisa didapatkan berdasarkan urutan pengerjaan dan waktu operasi.

Dalam menentukan suatu lintasan perakitan yang hanya menggunakan *single station*, terdapat batasan-batasan yang harus terpenuhi, seperti halnya yang. Batasan-batasan tersebut adalah sebagai berikut (Bedworth & Bailey, 1987:364).

1. $1 \leq M \leq N$

Jumlah stasiun kerja tidak boleh lebih besar dari pada jumlah operasi yang ada. Selain itu, jumlah minimal stasiun kerja adalah satu (1).

2. $t_i \leq C$

Tidak boleh ada waktu operasi maupun waktu siklus suatu stasiun kerja yang lebih besar dari waktu siklus yang telah ditetapkan sebelumnya.

Ada tiga jenis lintasan menurut variasi produk, yaitu: (Groover, 1987:142)

1. *Single Model Lines*

Dalam model ini semua komponen atau produk dibuat sama oleh sistem manufaktur.

2. *Mix model Lines*

Perbedaan komponen atau produk yang dibuat oleh sistem. Tetapi perbedaan ini dapat ditangani oleh sistem tanpa dibutuhkan pergantian persiapan secara fisik secara total atau pergantian peralatan dibutuhkan untuk setiap model.

3. *Multi Model Lines*

Perbedaan komponen atau produk yang dibuat oleh sistem. Tetapi komponen dibuat dalam *batch* karena pergantian persiapan secara fisik atau pergantian peralatan dibutuhkan untuk setiap model.

Hasil yang diharapkan dari penerapan keseimbangan lintasan adalah sebagai berikut.

1. Menghindari penumpukan barang dalam proses pada suatu bagian produksi.
2. Menghindari pengangguran pada bagian produksi lainnya.
3. Mendapatkan efisiensi sistem yang cukup tinggi.
4. Memenuhi rencana produksi yang telah ditetapkan.

2.2.1 *Mixed-Model Line Balancing*

Menurut Bukchin (2001:540) *mixed model line balancing* merupakan lintasan produksi secara manual yang mampu memproduksi jenis produk (model) yang berbeda-beda secara bersamaan dan berkelanjutan. Masing-masing stasiun kerja terdapat elemen pekerjaan yang mengatasi aktivitas produk yang homogen tersebut, dengan masing-masing memiliki spesifikasi aktivitas proses dan variasi waktu yang berbeda, sehingga stasiun kerja cukup fleksibel dan dapat melakukan pekerjaan secara kontinyu terhadap model yang berbeda. Metode ini termasuk ke dalam pendekatan heuristik yang digunakan untuk memecahkan masalah ketidakseimbangan lintasan produksi yang dalam proses produksinya membuat produk yang lebih dari satu jenis pada waktu yang bersamaan.

Tujuan dari *mixed model line balancing* adalah sebagai berikut.

1. Memaksimalkan hasil produksi.
2. Meminimalkan waktu siklus.
3. Meminimalkan jumlah stasiun kerja.
4. Meminimalkan waktu *delay*.
5. Meningkatkan efisiensi lintasan.

Pada permasalahan *mixed-model line balancing* ini digunakan permodelan dengan stasiun kerja paralel untuk mengatasi kemungkinan adanya waktu operasi maupun waktu siklus suatu stasiun kerja yang lebih besar dari waktu siklus yang telah ditetapkan

sebelumnya. Pada pembahasan *mixed-model line balancing* dengan stasiun kerja paralel ini membutuhkan karakteristik sebagai berikut (Vilarinho et al., 2002:1407).

1. Memiliki *planning horizon* (waktu sepanjang proses) yang tetap (P).
2. Sekelompok model sejenis ($m= 1, \dots, M$) dapat dirakit secara bersamaan.
3. *Forecast demand*, sepanjang *planning horizon*, untuk model m adalah D_m , yang digunakan untuk mencari *cycle time*.

$$C = P / \sum_{m=1}^M D_m \quad (2-1)$$

4. Proporsi keseluruhan dari jumlah model m yang dirakit, dimana D_m adalah jumlah produksi produk M dan D_p adalah jumlah produksi semua produk.

$$q_m = D_m / \sum_{p=1}^M D_p \quad (2-2)$$

5. *Precedence diagram* untuk semua model dapat digabungkan dan memiliki aktivitas sebesar N .
6. Aktivitas ($i=1, \dots, N$) dilakukan pada sekumpulan stasiun kerja ($k=1, \dots, S$).
7. Waktu yang dibutuhkan untuk menyelesaikan aktivitas i pada model m , t_{im} , dapat berbeda-beda sesuai modelnya ($t_{im} = 0$, menunjukkan bahwa model m tidak memerlukan aktivitas i).
8. Sebuah aktivitas hanya dapat dikerjakan pada satu stasiun kerja dan beberapa aktivitas yang digunakan oleh beberapa model dikerjakan pada satu stasiun kerja.
9. Beberapa kumpulan aktivitas yang tidak bisa dilakukan sebelum aktivitas i selesai, F_i (penerus aktivitas i), dibuat berdasarkan *precedence diagram* gabungan.
10. Konstrain zoning (ZP) adalah sekumpulan aktivitas yang saling berkaitan dan harus berada pada satu stasiun kerja (*compatible tasks*) dan ZN adalah sekumpulan aktivitas yang tidak dapat dikerjakan pada satu stasiun kerja (*incompatible tasks*).
11. Sebuah stasiun kerja yang dapat diduplikasi sampai nilai maksimum $MAXP$ (jumlah maksimum replika yang dibolehkan pada stasiun kerja), namun hanya jika waktu aktivitas dari salah satu aktivitas melebihi nilai standar ($\alpha\%$ dari *cycle time*).

2.2.2 Istilah dalam Keseimbangan Lintasan

Ada beberapa istilah yang lazim digunakan dalam keseimbangan lintasan. Berikut ini adalah istilah-istilah yang dimaksud (Baroto, 2002:192).

1. *Precedence diagram*

Precedence diagram digunakan sebelum melangkah pada penyelesaian menggunakan metode keseimbangan lintasan. *Precedence diagram* sebenarnya merupakan gambaran secara grafis dari urutan operasi kerja, serta ketergantungan pada operasi kerja lainnya

yang tujuannya untuk memudahkan pengontrolan dan perencanaan kegiatan yang terkait di dalamnya, adapun tanda yang dipakai dalam *precedence diagram* adalah sebagai berikut.

- a. Simbol lingkaran dengan huruf atau nomor di dalamnya untuk mempermudah identifikasi asli dari suatu proses operasi.
- b. Tanda panah menunjukkan ketergantungan dan urutan proses operasi. Hal ini operasi yang ada di pangkal panah berarti mendahului operasi kerja yang ada pada ujung anak panah.
- c. Angka di atas simbol lingkaran adalah waktu standar yang diperlukan untuk menyelesaikan setiap proses operasi.

2. Waktu Siklus (*cycle time*)

Waktu siklus adalah yang dibutuhkan seorang operator untuk menyelesaikan 1 siklus pekerjaannya termasuk untuk melakukan kerja manual dan berjalan dalam satu stasiun kerja. Terkadang diartikan sebagai waktu yang dibutuhkan untuk menghasilkan 1 unit produk, dalam hal ini ditentukan dari proses yang paling lama (*bottleneck*), apakah itu pekerjaan manusia atau mesin.

3. Waktu Menunggu (*Idle Time*)

Dimana operator atau pekerja menunggu untuk melakukan proses kerja ataupun kegiatan operasi yang selanjutnya dikerjakan, dimana merupakan hasil dari selisih atau perbedaan antara *Cycle time* (CT) dan *Station Time* (ST).

4. Stasiun Kerja

Stasiun kerja adalah tempat pada lintasan dimana proses dilakukan.

5. Efisiensi lintasan (*Line efficiency*)

Line efficiency merupakan rasio dari total waktu stasiun kerja dibagi dengan siklus dikalikan jumlah stasiun kerja atau jumlah efisiensi stasiun kerja dibagi jumlah stasiun kerja. Perhitungan efisiensi lintasan dirumuskan sebagai berikut.

$$EL = \frac{\sum_{i=1}^k ST_i}{K.C} \times 100\% \quad (2-3)$$

Sumber: Elsayed (1994)

Dimana:

EL = Efisiensi keseimbangan lintasan

ST_i = Waktu stasiun dari stasiun ke-i

K = Jumlah banyak stasiun kerja

C = Waktu siklus

6. Keseimbangan Waktu Senggang (*Balance Delay Time*)

Balance delay time merupakan jumlah total *idle time* dari semua stasiun kerja. Istilah ini juga dapat disebut sebagai *smoothness index* (SI). Nilai dari BDT adalah satuan waktu. Pada perhitungan ini rumus *balance delay time* yang digunakan adalah sebagai berikut.

$$BDT = \sum_{m=1}^M \sum_{k=1}^K q_m (SP_k \cdot C - \sum_{i \in O_k} t_{im}) \quad (2-4)$$

Sumber: Vilarinho et al (2002)

Dimana:

q_m = *Production share*

SP_k = Jumlah replikasi dari stasiun k

C = *Cycle time*

t_{im} = Waktu aktivitas produk

O_k = Aktivitas-aktivitas yang dikerjakan stasiun k

K = Stasiun kerja

M = Produk

7. *Balance between Station/* keseimbangan antar stasiun kerja

Balance between station adalah penyeimbangan waktu *idle*, dimana untuk setiap model waktu *idle* didistribusikan dari satu stasiun kerja ke stasiun kerja yang lain secara merata sehingga dipastikan bahwa semua pekerja dalam lintasan melakukan kira-kira jumlah pekerjaan yang sama pada setiap model yang dirakit. Nilai dari BWS antara 0 atau 0% hingga 1 atau 100%, dimana 0 atau 0% adalah nilai terbaik dan 1 atau 100% adalah nilai terburuk.

$$BWS = \left(\frac{S'}{S'-1} \sum_{m=1}^M q_m \sum_{k=1}^{S'} \left(\frac{S_{km}}{\sum_{i=1}^{S'} S_{im}} - \frac{1}{S'} \right)^2 \right) \times 100\% \quad (2-5)$$

Sumber: Mendes et al (2005)

Dimana:

S' = Stasiun kerja yang merupakan hasil dari tahap 1

S_{km} = Waktu menunggu (*idle time*) dari stasiun kerja k dari model m

$S_{im} = \sum S_{km}$

8. *Balance within Station/* keseimbangan dalam stasiun kerja

Balance within Station adalah penyeimbangan waktu *idle*, dimana untuk setiap stasiun kerja waktu *idle* didistribusikan dari satu model produk ke model produk yang lain secara merata sehingga dipastikan bahwa jumlah pekerjaan konstan dilakukan di setiap stasiun kerja terlepas dari model yang dirakit. Nilai dari BS antara 0 atau 0%

hingga 1 atau 100%, dimana 0 atau 0% adalah nilai terbaik dan 1 atau 100% adalah nilai terburuk.

$$BS = \left(\frac{M}{S'(M-1)} \sum_{k=1}^{S'} \sum_{m=1}^M \left(\frac{q_m s_{km}}{S_k} - \frac{1}{M} \right)^2 \right) \times 100\% \quad (2-6)$$

Sumber: Mendes et al (2005)

Dimana:

$$S_k = \sum_{m=1}^M q_m \cdot s_{km} \quad (2-7)$$

Sumber: Mendes et al (2005)

2.2.3 Pengukuran Waktu

Pengukuran waktu kerja adalah pekerjaan mengamati pekerja dan mencatat waktu kerjanya baik setiap elemen ataupun siklus dengan menggunakan alat-alat yang telah disiapkan (Sutalaksana et al., 1979:131). Faktor yang menyebabkan menurunnya produktivitas perusahaan adalah sifat dan keadaan barang, proses yang berjalan tidak semestinya, waktu tidak efektif yang bertumpuk selama produksi berlangsung, kekurangan pihak manajemen atau kelalaian para buruh. Selain itu bisa menjadi teknik utama untuk mengurangi kerja, terutama dengan meniadakan gerak yang tidak perlu dan dengan menggantikan metode yang tidak memenuhi syarat.

Tujuan pengukuran waktu kerja adalah untuk mendapatkan waktu baku yang harus dicapai oleh pekerja dalam menyelesaikan suatu pekerjaan. Pengukuran waktu yang dilakukan terhadap beberapa alternatif sistem kerja, maka yang terbaik dilihat dari waktu penyelesaian tersingkat. Pengukuran waktu juga ditujukan untuk mendapatkan waktu baku penyelesaian pekerjaan, yaitu waktu yang dibutuhkan secara wajar, normal, dan terbaik.

Pada garis besarnya teknik-teknik pengukuran waktu dibagi dalam dua bagian, yaitu: (Sutalaksana et al., 1979:117)

1. Pengukuran waktu kerja secara langsung, yaitu:
 - a. Jam waktu (*Stop watch time study*)
 - b. Sampling kerja (*Work sampling*)
2. Pengukuran waktu kerja secara tidak langsung, yaitu:
 - a. PMTS (*Predetermined Motion Time System*)
 - b. Standar Data

Waktu standar dapat digunakan untuk hal-hal berikut ini, yaitu: (Purnomo, 2004:327)

1. Penentuan jadwal dalam perencanaan kerja.
2. Penentuan biaya standar dan sebagai alat bantu dalam mempersiapkan anggaran.
3. Estimasi biaya produk sebelum memproses produk.

4. Penentuan efektivitas mesin.
5. Penentuan waktu standar yang digunakan sebagai dasar untuk upah insentif tenaga kerja langsung.
6. Penentuan waktu standar yang digunakan sebagai dasar untuk upah insentif tenaga kerja tidak langsung.
7. Penentuan waktu standar yang digunakan sebagai dasar untuk pengawasan biaya tenaga kerja.

2.2.3.1 Pengukuran Waktu Secara Langsung

Metode pengukuran langsung yaitu mengamati secara langsung pekerjaan yang dilakukan oleh operator dan mencatat waktu yang diperlukan oleh operator dalam melakukan pekerjaannya dengan terlebih dahulu membagi operasi kerja dalam elemen-elemen kerja yang sedetail mungkin dengan syarat masih bisa diamati dan diukur. Kemudian dari hasil pengamatan dan pengukuran tersebut akan didapatkan waktu baku ataupun distribusi waktu operator untuk mengerjakan pekerjaan tersebut. Ada dua metode yang digunakan pada pengukuran langsung yaitu metode jam henti (*Stopwatch Time Study*) dan metode *work sampling*.

2.2.3.2 Stopwatch Time Study (Pengukuran Waktu dengan Jam Henti/ *Stopwatch*)

Pengukuran waktu kerja menggunakan jam henti diperkenalkan Frederick W. Taylor pada abad ke-19. Metode ini baik untuk diaplikasikan pada pekerjaan yang singkat dan berulang (*repetitive*). Dari hasil pengukuran akan diperoleh waktu baku untuk menyelesaikan suatu siklus pekerjaan yang akan dipergunakan sebagai waktu standar penyelesaian suatu pekerjaan bagi semua pekerja yang akan melaksanakan pekerjaan yang sama. Dalam pengukuran kerja, hal-hal penting yang harus diketahui dan ditetapkan adalah untuk apa hasil pengukuran (dalam hal ini tentu saja waktu baku) tersebut digunakan dalam kaitannya dengan proses produksi. Biasanya, penetapan waktu baku akan dikaitkan dengan maksud-maksud pemberian insentif/bonus pekerja langsung (*direct labour*). Pengukuran kerja ini dapat diaplikasikan pada industri manufaktur dengan jumlah *output* yang konstan untuk selang waktu yang lama.

Pada aktivitas pengukuran kerja, operasi yang akan diukur dibagi menjadi elemen-elemen yang lebih kecil dan dibuat sedetail dan sependek mungkin akan tetapi masih mudah untuk diukur waktunya dengan teliti. Pengukuran elemen kerja ini sendiri dapat dilakukan dengan tiga metode, yaitu:

1. Metode kontinyu, dimana pengukuran dilakukan dengan memulai gerakan jarum jam henti pada permulaan pengerjaan elemen kerja yang pertama dan jarum jam tetap bergerak selama pengamatan berjalan.
2. Metode berulang, dimana pengukuran dilakukan dengan menggerakkan jarum jam henti pada saat elemen kerja pertama mulai berjalan dan dihentikan pada saat elemen kerja tersebut berhenti. Waktu dicatat dan jarum jam henti dikembalikan lagi ke posisi nol untuk melakukan pengukuran selanjutnya.
3. Cara akumulatif, dimana pengukuran dilakukan dengan menggunakan dua buah jam henti yang dipasang bersama didekat papan pengamatan dan dihubungkan sedemikian rupa sehingga ketika jarum jam henti pertama bergerak, jarum jam henti kedua akan berhenti, begitu terus sebaliknya.

2.3 Metode Dalam Keseimbangan Lintasan

Untuk menyeimbangkan lintasan, secara garis besar metode yang sering digunakan adalah sebagai berikut.

1. Metode Heuristik

Metode ini menggunakan aturan-aturan yang logis dalam memecahkan masalah. Metode ini tidak menjamin hasil yang optimum, tetapi dirancang untuk menghasilkan strategi yang relatif lebih baik dan mendekati hasil optimum yang sesungguhnya, sehingga metode ini digunakan dalam penelitian ini. Untuk beberapa metode umum heuristik adalah sebagai berikut.

- a. Metode bobot posisi (*ranked positional weight*)
- b. Metode Pembebanan Berurut (*Largest-Candidate Rule*)
- c. Metode Pembagian Wilayah (*Region Approach*)

2. Metode Analitis

Metode analitis atau matematis adalah model pemecahan masalah dengan menggunakan persamaan-persamaan atau model-model matematis agar diperoleh hasil yang optimal. Pada umumnya, metode matematis dapat memecahkan masalah keseimbangan lintasan dengan menggunakan *operation research*. Program linier adalah salah satu metode dari *operation research* yang dapat digunakan untuk memecahkan keseimbangan lintasan. Namun penggunaan model ini dinilai kurang ekonomis karena menggunakan perhitungan matematis yang cukup rumit dan membutuhkan waktu yang cukup lama untuk memperoleh hasilnya. Metode analitis atau matematis dapat dibagi menjadi sebagai berikut.

- a. Metode *Linier Programming*
 - b. Metode *Dynamic Programming*
 - c. Metode *Integer Programming*
3. Metode Komputerisasi

Metode ini menggunakan bantuan komputer dalam menyeimbangkan lintasan, yaitu dengan COMSOAL (*Computer Method of Sequencing Operation for Assembly Line*), ALPACA (*Assembly Line Balancing and Control Activity*), dan COMSAL (*Computer Method or Saumming Operation for Assemble*).

2.3.1 Metode Heuristik untuk Keseimbangan Lintasan Aktivitas

Yang termasuk ke dalam metode heuristik ini adalah sebagai berikut.

1. Metode Bobot Posisi (*Ranked Positional Weight*)

Metode ini dikemukakan oleh W.B. Hegelson dan D.P. Birnie pada tahun 1961, sehingga metode pengurutan bobot posisi ini biasa juga disebut dengan metode Hegelson-Birnie. Dalam metode ini dijelaskan bahwa proses perakitan terdiri dari beberapa elemen pekerjaan dengan urutan ketergantungan terhadap elemen pekerjaan sebelumnya. Untuk setiap elemen pekerjaan tersebut diberi bobot. Berdasarkan urutan bobotnya, pekerjaan-pekerjaan dikelompokkan ke dalam sejumlah stasiun kerja dengan memperhatikan waktu siklus yang ditetapkan. Bobot dihitung dengan *positional weight*. Bobot posisi (*positional weight*) adalah jumlah dari waktu pekerjaan tersebut dengan waktu pekerjaan yang mengikutinya. Yang dimaksud dengan posisi dari suatu pekerjaan adalah jumlah waktu pelaksanaan pekerjaan tersebut dengan semua waktu pelaksanaan semua pekerjaan yang mengikutinya.

Metode bobot posisi mempunyai kelebihan dalam kecepatan pemecahan masalah keseimbangan lintasan perakitan karena metode ini mudah dan sederhana. Metode ini cocok untuk pengambilan keputusan dalam pemecahan masalah keseimbangan lintasan perakitan secara cepat. Metode bobot posisi ini dapat dijelaskan dengan langkah-langkah sebagai berikut (Kusuma, 2002:97).

- a. Menghitung kecepatan lintasan yang diinginkan.
- b. Kecepatan lintasan adalah kecepatan produksi yang diinginkan atau kecepatan operasi paling lambat jika waktu operasi paling lambat itu lebih kecil dari kecepatan lintasan yang diinginkan.
- c. Membuat matriks keterdahuluan (*precedence matrix*) berdasarkan *precedence diagram*.

- d. Menghitung bobot posisi setiap operasi yang yang dihitung berdasarkan jumlah waktu operasi dan operasi- operasi yang mengikutinya.
 - e. Mengurutkan operasi yang dimulai dari bobot posisi terbesar sampai dengan bobot posisi paling kecil.
 - f. Lakukan pembebanan operasi pada stasiun kerja yang dimulai dari operasi dengan bobot posisi terbesar sampai dengan bobot posisi terkecil, dengan kriteria total waktu operasi lebih kecil dari pada kecepatan lintasan yang ditentukan.
 - g. Menghitung efisiensi rata-rata dari stasiun kerja yang terbentuk.
 - h. Gunakan prosedur *trial and error* untuk mencari pembebanan yang akan menghasilkan efisiensi rata-rata lebih besar dari efisiensi rata-rata pada langkah (f) di atas.
 - i. Ulangi langkah (f) dan (g) sampai tidak ditemukan lagi stasiun kerja yang memiliki efisiensi rata-rata yang lebih tinggi.
2. Metode Pembebanan Berurut (*Largest-Candidate Rule*)
- Pada metode ini memiliki cara pengerjaan yang lebih sederhana, namun hasil yang diperoleh masih harus saling dipertukarkan dengan cara *trial and error* untuk mendapatkan penyusunan stasiun kerja yang lebih akurat sehingga dapat memakan waktu lama. Metode ini dapat dijelaskan dengan langkah-langkah sebagai berikut.
- a. Menghitung kecepatan lintasan yang diinginkan.
 - b. Kecepatan lintasan adalah kecepatan produksi yang diinginkan atau kecepatan operasi paling lambat jika waktu operasi paling lambat itu lebih kecil dari kecepatan lintasan yang diinginkan.
 - c. Membuat matriks operasi pendahulu (P) dan operasi yang mengikuti (F) untuk setiap operasi berdasarkan *precedence diagram*.
 - d. Perhatikan baris pada matriks kegiatan pendahulu (P) yang semuanya terdiri dari angka nol (0) dan dibebankan elemen pekerjaan yang paling besar yang mungkin terjadi jika ada lebih dari satu baris yang memiliki elemen nol (0).
 - e. Perhatikan nomor elemen pada baris matriks kegiatan yang mengikuti (F) yang sesuai dengan elemen yang ditugaskan. Setelah itu, kembali perhatikan baris pada matriks (P) yang ditunjukkan. Kemudian, nomor identifikasi elemen yang telah dibebankan pada stasiun kerja diganti dengan nol (0).
 - f. Lanjutkan penugasan elemen-elemen pekerjaan itu pada setiap stasiun kerja dengan ketentuan bahwa waktu total operasi tidak melebihi kecepatan lintasan

yang ditetapkan. Proses ini dikerjakan hingga semua baris pada matriks (P) bernilai nol (0).

- g. Menghitung efisiensi rata-rata dari stasiun kerja yang terbentuk.
- h. Gunakan prosedur *trial and error* untuk mencari pembebanan yang akan menghasilkan efisiensi rata-rata lebih besar dari efisiensi rata-rata pada langkah (f) di atas.
- i. Ulangi langkah (f) dan (g) sampai tidak ditemukan lagi stasiun kerja yang memiliki efisiensi rata-rata yang lebih tinggi.

Dalam metode *Largest-Candidate Rule*, elemen-elemen pekerjaan diatur dalam waktu yang semakin menurun menurut data yang ada dimulai dari waktu terbesar sampai dengan waktu paling kecil, seperti yang terlihat pada Tabel 2.2 berikut ini.

Tabel 2.2

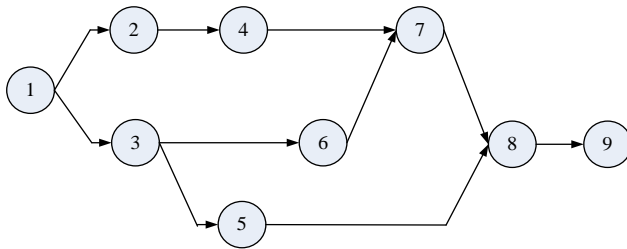
Daftar elemen pekerjaan untuk *Largest-Candidate Rule*

Operasi	Waktu Pengerjaan	Aktivitas Pendahulu
3	0,70	1
8	0,60	3, 4
11	0,50	9, 10
2	0,40	-
10	0,38	5, 8
7	0,32	3
5	0,30	2
9	0,27	6, 7, 8
1	0,20	-
12	0,12	11
6	0,11	3
4	0,10	1, 2

Sumber: Bukchin et al (2001:535)

3. Metode Pembagian Wilayah (*Region Approach*)

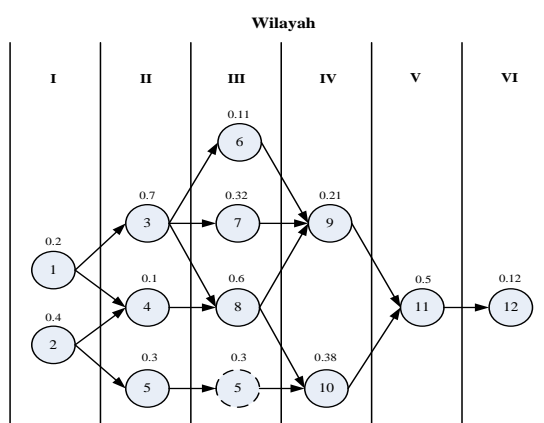
Metode ini biasa juga disebut metode Kilbridge-Wester dan diperkenalkan pada tahun 1961. Metode ini merupakan suatu prosedur heuristik yang memasukkan elemen-elemen pekerjaan dengan memilih elemen-elemen pekerjaan yang memiliki keterkaitan untuk mencapai keseimbangan. Operasi-operasi yang mempunyai tanggung jawab berat dan berada pada awal harus dijadwalkan terlebih dahulu. Sehubungan dengan hal tersebut, suatu operasi dengan waktu yang lebih besar dapat melewati operasi lain yang mempunyai tanggung jawab ketergantungan jika beberapa operasi *dependent* lainnya mempunyai waktu yang lebih kecil. Tanggung jawab ketergantungan ini maksudnya adalah jika suatu operasi A diikuti oleh rangkaian operasi lain maka operasi A tersebut mempunyai tanggung jawab terhadap operasi-operasi lainnya. Jika operasi A terhambat maka operasi-operasi lainnya akan terhambat.



Gambar 2.1 Precedence diagram yang menggambarkan urutan perakitan
Sumber: Kusuma (2002:107)

Pada metode ini, elemen-elemen pekerjaan dalam *precedence diagram* diatur ke dalam kolom, seperti yang ditunjukkan pada Gambar 2.1. Elemen-elemen pekerjaan tersebut kemudian bisa diorganisir ke dalam suatu daftar menurut kolom dengan elemen-elemen pekerjaan dalam kolom yang pertama dibuat terlebih dahulu. Langkah dasar metode *Region Approach* adalah sebagai berikut.

- a. Menghitung kecepatan lintasan yang diinginkan.
- b. Kecepatan lintasan adalah kecepatan produksi yang diinginkan atau kecepatan operasi paling lambat jika waktu operasi paling lambat itu lebih kecil dari kecepatan lintasan yang diinginkan.
- c. Membagi *precedence diagram* ke dalam wilayah dari kiri ke kanan.
- d. Urutkan pekerjaan mulai dari waktu operasi paling besar sampai dengan waktu operasi paling kecil pada setiap wilayah.
- e. Perhatikan pula untuk menyesuaikan diri terhadap batas wilayah. Bebaskan pekerjaan dengan urutan sebagai berikut.
 - 1) Daerah paling kiri terlebih dahulu.
 - 2) Kemudian pada setiap wilayah dibebaskan pekerjaan dengan waktu operasi yang terbesar pertama kali.



Gambar 2.2 Precedence diagram untuk metode *region approach*
Sumber: Kusuma (2002:107)

Proses dari pembebanan pekerjaan pada setiap wilayah tersebut dapat dilihat pada Tabel 2.3 berikut ini.

Tabel 2.3

Daftar Elemen Pekerjaan Menurut Wilayah Untuk Metode *Region Approach*

Operasi	Wilayah	Waktu Pengerjaan	Aktivitas Pendahulu
2	I	0,40	-
1	I	0,20	-
3	II	0,70	1
5	II, III	0,30	2
4	II	0,10	1, 2
8	III	0,60	3, 4
7	III	0,32	3
6	III	0,11	3
10	IV	0,38	5, 8
9	IV	0,27	6, 7, 8
11	V	0,50	9, 10
12	VI	0,12	11

Sumber: Bukchin et al (2001:537)

2.4 Simulated Annealing

Simulated annealing termasuk algoritma yang meniru perilaku fisik proses pendinginan baja. Teknik ini meniru perilaku baja yang mengalami pemanasan sampai suhu tertentu kemudian didinginkan secara perlahan. Ketika baja dipanaskan sampai suhu mendidih, atom-atom dalam baja tersebut bergerak bebas dan semakin terbatas gerakannya ketika suhunya turun. Ketika suhunya turun, susunan atomnya menjadi lebih teratur dan akhirnya membentuk kristal dan mempunyai energi internal yang minimum. *Simulated annealing* telah banyak diterapkan diberbagai masalah optimasi seperti TSP, VRP, penjadwalan pekerjaan dan beberapa masalah lain.

Proses pembentukan kristal ini pada dasarnya sangat bergantung pada laju penurunan suhu. Jika laju penurunan ini terlalu besar mungkin tidak mencapai status kristal tetapi statusnya berupa *polycrystalline* dimana nilai energi internalnya lebih besar. Dalam aplikasi teknik, pendinginan yang cepat menyebabkan kerusakan di dalam material. Sehingga, suhu baja yang mendidih perlu diturunkan secara perlahan dan teratur sehingga memungkinkan pemadatan yang bagus dan menghasilkan Kristal dengan susunan yang bagus dan kandungan energy internal yang kecil. Proses pendinginan secara perlahan ini disebut *annealing*.

Metode ini bertujuan mencapai nilai minimum fungsi dalam permasalahan minimasi. Proses pendinginan ini ditiru dengan cara menentukan parameter yang serupa dengan suhu lalu mengontrolnya dengan menggunakan konsep distribusi probabilitas Boltzmann yang menyatakan bahwa energi (E) dari suatu sistem dalam keseimbangan panas pada suhu T terdistribusikan secara probabilistik mengikuti rumus.

$$P(E) = e^{-E/kT} \quad (2-8)$$

Sumber: Santosa et al (2011)

Dimana:

$P(E)$ = Peluang mencapai tingkat energi E

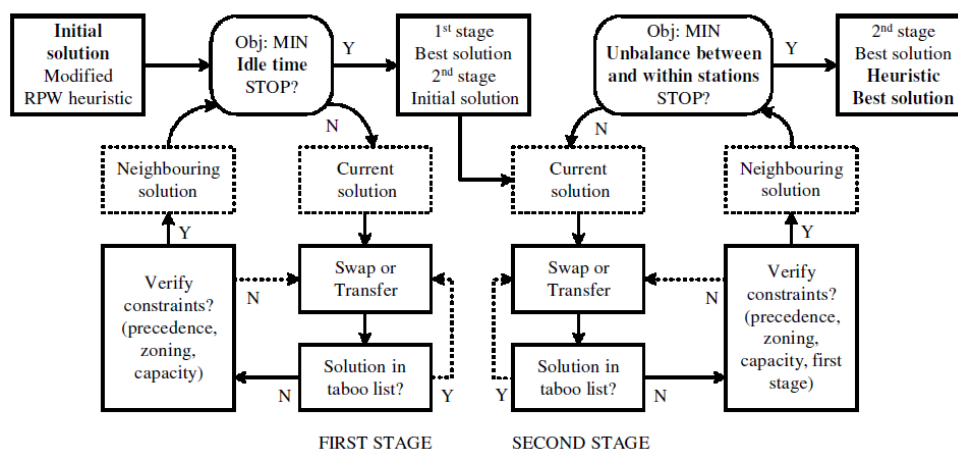
T = Suhu

k = Konstanta Boltzmann

Persamaan (2-9) memperlihatkan bahwa pada suhu tinggi sistem tersebut mempunyai probabilitas yang hampir *uniform* berada pada sembarang status, sedangkan pada suhu rendah sistem mempunyai probabilitas yang rendah berada pada status energi yang tinggi. Ini menunjukkan bahwa jika proses pencarian solusi mengikuti probabilitas Boltzmann konvergensi algoritma *simulated annealing* dapat diatur dengan mengatur temperatur T (Santosa et al., 2011:225).

Penggunaan *simulated annealing* pada kasus *mixed-model line balancing* ini dipilih karena metode ini mampu untuk merespon modifikasi dari fungsi tujuan ataupun konstrain masalah (Vilarinho et al., 2002:1406) selain dalam masalah ini dan struktur *neighbourhood* dapat dengan mudah ditemukan.

2.4.1 Prosedur Dua Tingkat untuk *Mixed Model Line Balancing*



Gambar 2.3 Prosedur dua tahap untuk *mixed model line balancing*

Sumber: Vilarinho et al (2002:1411)

Prosedur dua tingkat untuk *mixed model line balancing* dengan pendekatan *simulated annealing* ditunjukkan pada Gambar 2.3. Untuk masukan, setelah mendapatkan nilai *initial solution* dari metode heuristik, dilanjutkan dengan perhitungan heuristik yang diadaptasi dari algoritma milik Simaria&Vilarinho (2002:1411) dimana terbagi menjadi dua tahap perhitungan. Pada tahap pertama, tujuan perhitungan heuristik adalah untuk meminimasi jumlah stasiun kerja, dimana hal itu direpresentasikan dengan perhitungan *balance delay*

time (BDT) seperti pada rumus (2-3). Kemudian untuk menemukan *neighbouring solution* untuk tahap pertama didapatkan dengan dua cara, yaitu dengan penukaran aktivitas atau menukar aktivitas dari stasiun yang berbeda dan yang kedua pemindahan aktivitas atau mentransfer aktivitas ke stasiun lainnya. Penentuan pekerjaan yang ditukar atau ditransfer dipilih secara random, kemudian dicek apakah telah memenuhi syarat konstrain yang ada, yaitu kesesuaian dengan *precedence diagram* yang telah dibuat sebelumnya, lalu jumlah *cycle time*. Pada tahap pertama karena tujuannya meminimalkan jumlah stasiun kerja, maka pemindahan aktivitas atau transfer aktivitas memiliki peran besar daripada penukaran aktivitas. Setelah menemukan titik solusi baru (BDT baru) jika solusi lebih kecil dari nilai awal (BDT), maka hasil diterima. Namun jika sebaliknya, solusi dapat diterima dengan menggunakan rumus sebagai berikut.

$$P = \exp(-\Delta / kT) > r \quad (2-9)$$

Sumber: Santosa et al (2011)

Dimana:

$$\Delta = \frac{BDT_{baru} - BDT}{BDT_{baru}} \times 100 \quad (2-10)$$

Sumber: Vilarinho et al (2002)

k = Konstanta Boltzman dengan nilai 1

T = Temperatur pada saat iterasi

r = Angka random bernilai antara 0 hingga 1

Setelah ditemukan solusi untuk tahap pertama kemudian dilanjutkan dengan perhitungan pada tahap kedua dimana tujuan utamanya adalah untuk meminimasi nilai beban kerja pada stasiun kerja dan antar stasiun kerja yang direpresentasikan sebagai *balance within station* (BS) dan *balance between station* (BWS) seperti pada rumus (2-5) dan (2-6). Pada tahap kedua karena tujuannya untuk meminimalkan beban kerja antar dan dalam stasiun kerja, maka penukaran aktivitas atau menukar aktivitas memiliki peran yang lebih besar dalam pencarian solusi yang baru. Setelah menemukan titik solusi baru (BWS dan BS baru) jika solusi lebih kecil dari nilai awal (BWS dan BS), maka hasil diterima. Namun jika sebaliknya, solusi dapat diterima dengan menggunakan rumus (2-9), dengan nilai Δ sebagai berikut.

$$\Delta = \frac{(BS_{baru} - BS) + (BWS_{baru} - BWS)}{BS_{baru} + BWS_{baru}} \times 100 \quad (2-11)$$

Sumber: Vilarinho et al (2002)

Annealing schedule secara umum digunakan untuk kedua tahap prosedur, dimana parameter kontrol didefinisikan sebagai berikut (Vilarinho et al., 2002:1410).

1. Temperatur awal (T_0), jika nilai awal temperatur ini terlalu tinggi maka diperlukan lebih banyak pengurangan temperatur untuk konvergen (Santosa et al, 2011:228).
2. Faktor pereduksi temperatur (Cr), faktor pengurangan temperatur dimana jika terlalu besar (0,8 atau 0,9) akan memerlukan terlalu banyak langkah komputasi untuk konvergen. Sebaliknya, jika terlalu kecil (0,1 atau 0,2) bisa berakibat terlalu cepatnya penurunan temperatur sehingga akan banyak titik potensial yang dapat menjadi solusi global terlewat (Santosa et al., 2011:228).
3. Jumlah iterasi setiap temperatur (L), merupakan jumlah solusi yang dihasilkan setiap temperatur, dimana faktor yang mendominasi pada usaha komputasi pemecahan masalah adalah banyaknya aktivitas (N). Jadi, untuk membatasi usaha komputasi jumlah solusi yang dicari pada masing-masing tingkat suhu dirumuskan dengan KN , di mana K adalah konstanta yang ditentukan pengguna, dengan $K = 1$ adalah nilai yang disarankan (Vilarinho et al., 2002:1410-1411).
4. Kriteria perhentian, dua kriteria alternatif yang ditetapkan, menentukan kapan sistem telah mencapai tingkat energi yang diinginkan. Beberapa kriteria yang paling umum didasarkan pada total solusi yang dihasilkan dan titik temperatur berhenti (*freezing temperature*). Walaupun begitu, tidak ada nilai temperatur awal, faktor pereduksi suhu, jumlah iterasi setiap temperatur dan temperatur berhenti yang unik yang sesuai untuk semua permasalahan (Santosa et al., 2011:228).

Adapun langkah-langkah penukaran aktivitas dan pemindahan aktivitas adalah sebagai berikut (Vilarinho et al., 2002:1413).

1. Penukaran Aktivitas

- a. Z adalah stasiun kerja yang dipilih secara acak dan X adalah model yang untuk stasiun itu memiliki penyimpangan waktu *idle* tertinggi ($S_{zm.qm}$) dari rata-rata waktu *idle* (SZ/M) stasiun kerja.

$$X : \Delta S_{ZX} = \max_m \left\{ \left| S_{zm} \cdot q_m - \frac{SZ}{M} \right| \right\} \quad (2-12)$$

- b. Jika $SZX > SZ / M$, lanjutkan ke langkah c, jika tidak ke langkah e.
- c. Pilih aktivitas pada stasiun kerja Z dengan waktu pemrosesan paling rendah untuk model X .

$$T_1 : t_{T_1X} = \min_1 \{t_{iX}\} \wedge i \in O_Z \quad (2-13)$$

- d. Dari serangkaian aktivitas yang dilakukan pada model X yang tidak ada pada stasiun kerja Z dan memiliki waktu aktivitas lebih tinggi dari waktu aktivitas T_1 , pilih salah satu secara acak. Lanjutkan ke langkah 7.

$$T_2 : t_{T_2X} > t_{T_1X} \wedge T_2 \notin O_Z \quad (2-14)$$

- e. Pilih aktivitas yang diberikan ke stasiun kerja Z dengan waktu pemrosesan tertinggi untuk model X.

$$T_1 : t_{T_1X} = \max_i \{t_{iX}\} \wedge i \in O_Z \quad (2-15)$$

- f. Dari set aktivitas yang dilakukan pada model X yang tidak ditugaskan untuk stasiun kerja Z dan memiliki waktu aktivitas lebih kecil dari waktu aktivitas T1, pilih salah satu secara acak.

$$T_2 : t_{T_2X} < t_{T_1X} \wedge T_2 \notin O_Z \quad (2-16)$$

- g. Jika *precedence*, konstrain zoning, kapasitas, dan jumlah stasiun kerja terpenuhi, tukar posisi aktivitas T1 dan T2.

2. Pemindahan Aktivitas

- a. Z adalah stasiun kerja yang dipilih secara acak dan X adalah model yang untuk stasiun itu memiliki penyimpangan waktu *idle* tertinggi ($S_{zm.qm}$) dari rata-rata waktu *idle* (SZ/M) stasiun kerja.

$$X : \Delta S_{ZX} = \max_m \left\{ \left| S_{Zm} \cdot q_m - \frac{SZ}{M} \right| \right\} \quad (2-17)$$

- b. Jika $SZX > SZ / M$, lanjutkan ke langkah c, jika tidak ke langkah e.
c. Pilih aktivitas yang tidak ada pada stasiun Z dengan waktu pemrosesan untuk model X lebih tinggi dari model lain.

$$T_1 : t_{T_1X} = \max_m \{t_{T_1m}\} \wedge i \notin O_Z \quad (2-18)$$

- d. Jika *precedence*, konstrain zoning, kapasitas, dan jumlah stasiun kerja terpenuhi, transfer aktivitas T1 ke stasiun kerja Z.
e. Pilih aktivitas yang ada di stasiun Z dengan waktu pemrosesan tertinggi untuk model X.

$$T_1 : t_{T_1X} = \max_i \{t_{iX}\} \wedge i \in O_Z \quad (2-19)$$

- f. Secara acak memilih stasiun kerja (W) dimana beban kerja untuk model X lebih rendah dari rata-rata waktu *idle* stasiun kerja.

$$W = S_{WX} < S_W/M \quad (2-20)$$

- g. Jika *precedence*, konstrain zoning, kapasitas, dan jumlah stasiun kerja terpenuhi, transfer aktivitas T1 ke stasiun kerja W.

Tujuan dari stasiun paralel ini yaitu: (Vilarinho et al, 2002:1410)

1. Untuk meminimasi jumlah stasiun kerja dengan diberikan waktu siklus (*cycle time*).
2. Untuk menyeimbangkan beban kerja pada stasiun kerja dan antar stasiun kerja.

3. Batas atas dari maksimal jumlah replikasi juga dapat ditentukan oleh pembuat keputusan.
4. Konstrain Zoning (ZP) juga dapat dihitung.
5. Pembuat keputusan dapat menentukan waktu aktivitas minimal yang memicu adanya replikasi.

2.5 MATLAB

MATLAB merupakan suatu bahasa pemrograman tingkat tinggi yang diperuntukkan untuk komputasi teknis. MATLAB mengintegrasikan aspek komputasi, visualisasi, dan pemrograman dalam suatu lingkungan yang mudah dilakukan. MATLAB dapat digunakan untuk aplikasi sebagai berikut.

1. Pemodelan, simulasi, *prototype*
2. Komputasi dan matematika
3. *Data analysis*, explorasi, visualisasi
4. *Scientific and engineering graphics*
5. *Application development*, termasuk pembuatan *graphical user interface* (GUI) yang memudahkan penggunaan bagi kalangan yang awam dengan komputasi.

Dalam hal penulisan program, MATLAB bisa menghemat banyak waktu jika dibandingkan penulisan dalam C atau Fortran. Nama MATLAB merupakan singkatan dari *matrix laboratory*. MATLAB memiliki banyak *toolbox* yang memungkinkan pengguna mempelajari dan mengaplikasikan teknologi tertentu. *Toolbox* adalah kumpulan fungsi dalam MATLAB (*M-file*) yang komprehensif yang digunakan untuk menyelesaikan kelas masalah di bidang tertentu. MATLAB juga merupakan bahasa pemrograman komputer berbasis *window* dengan yang walaupun orientasi dasarnya adalah matrik, namun tidak menutup kemungkinan untuk pengerjaan permasalahan non-matrik. Pada MATLAB terdapat tiga *windows* yang digunakan dalam operasinya, yaitu:

1. *Command windows* (layar perintah)
2. *Figure windows* (layar gambar)
3. *Note Pad* (sebagai editor program)

Berbagai *toolbox* dalam MATLAB antara lain *signal processing*, *control systems*, *neural networks*, *fuzzy logic*, *wavelets*, *simulation*, *statistics*, *optimization*, *bioinformatics*, *genetic algorithm*, dan lain-lain (Santosa, 2008:1). Sistem MATLAB terdiri atas lima bagian utama, yaitu:

1. *Development Environment*

Ini adalah kumpulan semua alat-alat dan fasilitas untuk membantu kita dalam menggunakan fungsi dan *file* MATLAB. Bagian ini memuat *desktop*, *command window*, *command history*, *editor and debugger*, dan *browser* untuk melihat *help*, *workspace*, *files*.

2. *The MATLAB Mathematical Function Library*

Bagian ini adalah koleksi semua algoritma komputasi, mulai dari fungsi sederhana seperti *sum*, *sine*, *cosine* sampai fungsi lebih rumit seperti, invers matriks, nilai eigen, fungsi Bessel, dan *fast Fourier transform*.

3. *The MATLAB language*

Ini adalah bahasa matriks/array level tinggi dengan *control flow*, fungsi, struktur data, *input/output*, dan fitur objek *programming* lainnya.

4. *Graphics*

MATLAB mempunyai fasilitas untuk menampilkan vektor dan matriks sebagai grafik. Fasilitas ini mencakup visualisasi data dua/tiga dimensi, pemrosesan citra (*image*), animasi, dan grafik animasi.

5. *The MATLAB Application Program Interface (API)*

Paket ini memungkinkan kita menulis bahasa C dan Fortran yang berinteraksi dengan MATLAB dan memuat fasilitas untuk pemanggilan kode-kode dari MATLAB (*dynamic linking*), yang disebut MATLAB sebagai mesin penghitung serta untuk membaca dan menulis *MAT-files*.

2.5.1 Pseudocode

Dalam penelitian ini, tahap *simulated annealing* dilakukan dengan *software* MATLAB dan langkah-langkahnya disusun terlebih dahulu melalui sebuah *pseudocode*. *Pseudocode* adalah deskripsi dari algoritma pemrograman komputer yang menggunakan struktur sederhana dari beberapa bahasa pemrograman tetapi bahasa tersebut hanya ditujukan agar dapat dibaca manusia. Biasanya yang ditulis dari *pseudocode* adalah variabel dan fungsi. Tujuan penggunaan utama dari *pseudocode* adalah untuk memudahkan manusia dalam memahami prinsip-prinsip dari suatu algoritma. Dalam *pseudocode*, tidak ada *syntax* standar yang resmi. Karena itu, *pseudocode* ini dapat kita terapkan dalam berbagai bahasa pemrograman. Tentu saja harus disesuaikan setiap tahapnya dengan bahasa pemrograman yang digunakan. Fungsi dari *pseudocode* mungkin sama dengan *flowchart*. Perbedaannya terletak pada cara penyampaianya. *Pseudocode* menggunakan kata-kata untuk

menjelaskan suatu algoritma, sedangkan *flowchart* menggunakan gambar. Berikut adalah contoh *pseudocode* fungsi untuk menentukan nilai minimal dari dua bilangan.

```
function min(x,y)
if x > y then
return y
else
return x
end if
```

Halaman ini sengaja dikosongkan