BAB II

LANDASAN TEORI

2.1 SISTEM PRODUKSI

Sistem produksi bertujuan untuk merencanakan dan mengendalikan produksi agar lebih efisien, efektif, produktif, atau optimal. Untuk memilih sistem produksi yang tepat, harus dilihat dulu mengenai jenis produksinya. Jenis atau tipe produksi sangat bergantung pada jumlah produksi dan bagaimana cara memproduksinya. Industri manufaktur terdiri atas banyak jenis yang berbeda satu sama lain. Jenis-jenis industri ini dapat dikelompokkan berdasarkan beberapa kriteria.

Berdasarkan cara pembuatan (dan masa pengerjaan), produksi dapat diklasifikasikan menjadi tipe-tipe berikut (Teguh Baroto, hal 18) :

1. *Engineering To Order* (ETO)

   Bila perusahaan melakukan rekayasa mulai penyiapan fasilitas sampai pembuatan untuk memenuhi pesanan (*order*). Produk yang dipesan biasanya berjumlah satu unit dan spesifikasinya sangat berbeda antara pesanan yang satu dengan yang lainnya. Disamping itu, sangat banyak aktivitas yang terlibat dalam pembuatannya.

2. *Made To Order* (MTO)

   Bila perusahaan berproduksi (membuat) dengan fasilitas produksi yang dimiliki untuk memenuhi pasanan (*order*).

3. *Assembly To Order* (ATO)

   Bila perusahaan memproduksi (merakit) dengan fasilitas produksi yang dimiliki untuk memenuhi pesanan (*order*).

4. *Made To Stock* (MTS)

   Bila produksi perusahaan tidak ditujukan untuk melayani pesanan, namun distok untuk mengantisipasi permintaan.

Berdasarkan ukuran jumlah produk yang dihasilkan, produksi dapat dikelompokkan menjadi :

1. Produksi proyek

   Biasanya jumlah unit yang diproduksi satu dengan jumlah operasi banyak dan melibatkan banyak sumber daya.

2. Produksi batch

   Bila jumlah unit yang diproduksi berukuran sedang, biasanya perusahaan memproduksi banyak jenis produk.

3. Produksi massal

   Bila jumlah unit yang diproduksi sangat besar, jenis yang diproduksi perusahaan umumnya lebih sedikit dibanding batch.

Berdasarkan cara memproduksi (berhubungan dengan pengaturan fasilitas produksi), produksi dikelompokkan menjadi :

1. Produksi flow shop
2. Produksi fleksibel (*Flexible manufactiring systems*)
3. Produksi job shop, biasanya untuk volume produksi batch
4. Produksi kontinu, biasanya untuk volume produksi massal.

2.1.1 Ruang Lingkup Perencanaan Dan Pengendalian Produksi

Perencanaan merupakan salah satu fungsi manajemen. Dalam perencanaan ditentukan beberapa usaha atau tindakan yang perlu diambil oleh manager dengan mempertimbangkan masalah yang mungkin timbul di masa yang akan datang. Perencanaan dibedakan menjadi dua hal, yaitu perencanaan usaha (*General Bussines Planning*) dan perencanaan produksi (*Production Planning*).

Perencanaan produksi yang disusun harus fleksibel agar peralatan dan fasilitas produksi dapat dipergunakan secara optimal. Salah satu kegiatan dalam perencanaan produksi adalah penjadwalan, yaitu pengalokasian beberapa sumber untuk melaksanakan sekumpulan pekerjaan selama waktu tertentu.

Perencanaan dan pengendalian produksi (PPC) pada industri manufaktur apa pun akan memiliki fungsi yang sama. Fungsi atau aktivitas-aktivitas yang ditangani oleh departemen PPC atau PPIC secara umum adalah sebagai berikut (Teguh Baroto, hal 15) :

1. Mengelola pesanan (*order*) dari pelanggan

   Para pelanggan memasukkan pesanan-pesanan untuk berbagai produk. Pesanan-pesanan ini dimasukkan dalam jadwal produksi utama, ini bila jenis produksinya *made to order*.

2. Meramalkan permintaan

   Perusahaan biasanya berusaha memproduksi secara lebih independen terhadap fluktuasi permintaan. Permintaan iniperlu diramalkan agar skenario produksi dapat mengantisipasi fluktuasi permintaan tersebut. Permintaan ini harus dilakukan bila tipe produksinya adalah *made to stock*.

3. Mengelola persediaan

   Tindakan pengelolaan persediaan berupa melakukan transaksi persediaan, membuat kebijakan persediaan pengaman, kebijakan kuantitas pesanan/produksi, kebijakan produksi dan periode pemesanan, dan mengukur performansi keuangan dari kebijakan yang dibuat.

4. Menyusun rencana agregat (penyesuaian permintaan dengan kapasitas)

   Pesanan pelanggan dan atau ramalan permintaan harus dikompromikan dengan sumber daya perusahaan (fasilitas, mesin, tenaga kerja, keuangan, dan lain-lain). Rencana agregat bertujuan untuk membuat skenario pembebanan kerja untuk mesin dan tenaga

kerja (reguler, lembur, dan subkontrak) secara optimal untuk keseluruhan produk dan sumber daya secara terpadu.

5. Membuat jadwal induk produksi (JIP)

   JIP adalah suatu rencana terperinci mengenai apa dan berapa unit yang harus diproduksi pada suatu periode tertentu untuk setiap item produksi.

6. Merencanakan kebutuhan.

   Perencanaan kebutuhan material bertujuan untuk menentukan apa, berapa, dan kapan komponen, sub-assembly, dan bahan penunjang yang harus disiapkan.

7. Melakukan penjadwalan pada mesin atau fasilitas produksi

   Penjadwalan ini meliputi urutan pengerjaan, waktu penyelesaian pesanan, kebutuhan waktu penyelesaian, prioritas pengerjaan, dan lain-lainnya.

8. Monitoring dan pelaporan pembebanan kerja dibanding kapasitas produksi

   Kemajuan tahap demi tahap dimonitor dan dibuat laporannya untuk dianalisis. Apakah pelaksanaan sesuai rencana yang telah dibuat ?

9. Evaluasi skenario pembebanan dan kapasitas

   Bila realisasi tidak sesuai rencana, maka rencana agregat, JIP, dan penjadwalan dapat diubah/disesuaikan kebutuhannya. Untuk jangka panjang, evaluasi ini dapat digunakan untuk mengubah (menambah) kapasitas produksi.

dibuat. Order aktual, dalam fase agregat, adalah dasar untuk penjadwalan sumber daya produksi (fasilitas, tenaga kerja, dan peralatan), kemudian pada setiap unit produksi untuk mendapatkan tingkat penggunaan optimal dari kapasitas yang ada atau tujuan lainnya. (Teguh Baroto, hal 167)

Penjadwalan didefinisikan oleh Kenneth R. Barker (K.R Baker, hal 2) yaitu proses pengalokasian sumber untuk memilih sekumpulan tugas dalam jangka waktu tertentu. Dari definisi ini dapat dijabarkan dalam dua (2) artian yang berbeda. Yang pertama yaitu penjadwalan merupakan sebuah fungsi pengambilan keputusan, yaitu dalam menentukan jadwal yang paling tepat. Arti kedua bahwa penjadwalan merupakan sebuah teori yang berisi kumpulan prinsip, model, teknik dan konklusi logis dalam proses pengambilan keputusan. Tujuan penjadwalan secara umum yaitu (K.R Baker, hal 3) :

1. Meningkatkan produktivitas mesin, yaitu dengan mengurangi waktu mesin menganggur.
2. Mengurangi persediaan barang setengah jadi dengan jalan mengurangi jumlah rata-rata pekerjaan yang menunggu dalam antrian suatu mesin karena mesin tersebut sibuk.
3. Mengurangi keterlambatan karena batas waktu telah terlampaui dengan cara mengurangi maksimum keterlambatan maupun dengan mengurangi jumlah pekerjaan yang terlambat.
4. Meminimasi ongkos produksi.

2.2.1 Ukuran Keberhasilan Penjadwalan

Ukuran keberhasilan dari suatu pelaksanaan aktivitas penjadwalan adalah meminimasi kriteria-kriteria keberhasilan sebagai berikut (A.H Nasution, hal 172) :

1. Rata-rata waktu alir (*Mean Flow Time*).
2. Makespan, yaitu total waktu proses yang dibutuhkan untuk menyelesaikan suatu kumpulan job.
3. Rata-rata keterlambatan (*Mean Tardiness*).
4. Jumlah job yang terlambat.
5. Jumlah mesin yang menganggur.
6. Jumlah persediaan.

Meminimasi makespan, misalnya, dimaksudkan untuk meraih utilisasi yang tinggi dari peralatan dan sumberdaya dengan cara menyelesaikan seluruh job secepatnya; meminimasi waktu alir akan mengurangi persediaan barang setengah jadi; sedangkan meminimasi jumlah job yang menganggur berarti akan meminimasi nilai dari maksimum ukuran kelambatan. Kesemua kriteria keberhasilan pelaksanaan penjadwalan tersebut dilandasi keinginan untuk memuaskan konsumen dan efisiensi biaya internal perusahaan.

2.2.2 Output Sistem Penjadwalan

Untuk memastikan bahwa suatu aliran kerja yang lancar akan melalui tahapan produksi, maka sistem penjadwalan harus membentuk aktivitas-aktivitas output sebagai berikut (A.H Nasution, hal 174) :

1. Pembebanan (*Loading*)

   Pembebanan melibatkan penyesuaian kebutuhan kapasitas untuk order-order yang diterima atau diperkirakan dengan kapasitas yang tersedia. Pembebanan dilakukan dengan menugaskan order-order pada fasilitas-fasilitas, operator-operator, dan peralatan tertentu.

2. Penentuan Urutan (*Sequencing*)

   Penentuan urutan ini merupakan penugasan tentang order-order mana yang diprioritaskan untuk diproses dahulu bila suatu fasilitas harus memproses banyak job.

3. Prioritas Job (*Dispatching*)

   Dispatching merupakan prioritas kerja tentang job-job mana yang diseleksi dan diprioritaskan untuk diproses.

4. Pengendalian Kinerja Penjadwalan

   Dilakukan dengan cara :

   a. Memonitor perkembangan pencapaian pemenuhan order dalam semua sektor.

   b. Merancang ulang *sequencing*, bila ada kesalahan atau ada prioritas utama baru.

5. Up-dating Jadwal

   Pelaksanaan jadwal biasanya selalu ada masalah baru yang berbeda dari saat pembuatan jadwal, maka jadwal harus segera diperbaharui bila ada permasalahan baru yang memang perlu diakomodasi.

2.2.3 Input Sistem Penjadwalan

Pekerjaan-pekerjaan yang berupa alokasi kapasitas untuk order-order, penugasan prioritas job, dan pengendalian jadwal produksi membutuhkan informasi terperinci, dimana informasi-informasi tersebut akan menyatakan input dari sistem penjadwalan.

Pada bagian ini, kita harus menentukan kebutuhan-kebutuhan kapasitas dari order-order yang dijadwalkan dalam hal macam dan jumlah sumberdaya yang digunakan. Untuk produk-produk tertentu, informasi ini bisa diperoleh dari lembar kerja operasi (berisi ketrampilan dan peralatan yang dibutuhkan, waktu standar, dan lain-lain) dan BOM (berisi kebutuhan-kebutuhan akan komponen, sub komponen, dan bahan pendukung). Kualitas dari keputusan-keputusan pnjadwalan sangat dipengaruhi oleh ketepatan estimasi input-input diatas. Oleh karena itu, pemeliharaan catatan terbaru tentang status tenaga kerja dan peralatan yang tersedia, dan perubahan kebutuhan kapasitas yang diakibatkan disain produk atau disain proses menjadi sangat penting.

2.2.4 Jenis Penjadwalan Operasi

Ada 2 bagian besar jenis penjadwalan operasi, yaitu penjadwalan (Hendri Tanjung, hal 437) :

1. Proses lini yang berlangsung terus menerus (*Line Process*),
2. Proses yang terputus-putus (*Intermitten Process*)

2.2.4.1 *Line Process*

Line Process atau pemrosesan lini dibutuhkan oleh *assembling line* (lini perakitan) atau *processing industry* (industri pengolahan). Sebagian masalah penjadwalan dapat diselesaikan dengan desain proses, apalagi untuk satu produk pada satu lini. Akan tetapi, lini proses pada umumnya terkait dengan pembuatan banyak barang pada line tunggal. Artinya, banyak produk dibuat pada sebuah fasilitas, sehingga bersaing menggunakan sumber daya yang terbatas. Produk yang dihasilkan adalah tumpukan (*batch*), sehingga diperlukan berapa lot yang ekonomis dengan mempertimbangkan biaya yang terjadi.

Karena banyaknya barang yang dibuat pada lini tunggal, maka dibutuhkan waktu peralihan untuk mengolah barang yang satu ke barang yang lain. Peralihan ini dapat berupa penyetelan dan mempertahankan sediaan yang seimbang atau peralihan yang kompleks seperti penggantian alat, dan modifikasi stasiun kerja. Jika satu produk pada satu lini, maka tidak ada persoalan penjadwalan dan tidak ada peralihan. Faktor penting dan patut diperhatikan dalam waktu peralihan ini adalah

2.2.5 Klasifikasi Persoalan Penjadwalan

Klasifikasi persoalan penjadwalan yang sering digunakan dapat dibedakan sebagai berikut (Sri Lisa Susanty, hal 36) :

1. Berdasarkan jumlah mesin

   Yaitu penjadwalan N job pada satu mesin dan penjadwalan N job pada M mesin yang terbagi lagi menjadi 2 bagian disesuaikan dengan keadaan, yaitu M mesin pararel, dimana setiap job hanya dikerjakan pada satu mesin dan M mesin seri, dimana setiap job harus melewati beberapa mesin.

2. Berdasarkan pola kedatangan pekerjaan

   Yaitu pola kedatangan statistik dan pola kedatangan dinamik. Pola kedatangan statistik adalah semua pekerjaan datang secara bersamaan dan semua fasilitas tersedia pada saat kedatangan pekerjaan. Sedangkan pola kedatangan dinamik adalah pekerjaan datang secara acak selama masa penjadwalan.

3. Ketidakpastian pada pekerjaan dan mesin

   Yaitu deterministik dan stokastik. Deterministik adalah terdapat kepastian tentang pekerjaan dan mesin, misalnya tentang waktu kedatangan, waktu set-up, dan waktu proses. Stokastik adalah terdapat ketidakpastian mengenai pekerjaan dan mesin.

2.2.6 Penjadwalan Flow Shop

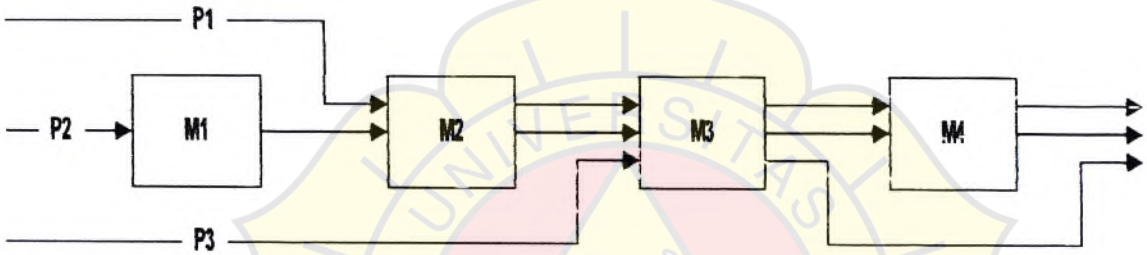
Usaha terbanyak yang dibutuhkan untuk melaksanakan suatu bentuk penjadwalan flow shop dilakukan pada tahapan desain sistem produksinya. Pada flow shop ini, terjadilah suatu pergerakan unit-unit yang benar-benar terus-menerus melalui suatu rangkaian stasiun-stasiun kerja yang disusun berdasarkan produk.

Susunan suatu proses produksi jenis flow shop dapat diterapkan dengan tepat untuk produk-produk dengan desain yang stabil dan diproduksi secara banyak volume, sehingga investasi dengan tujuan khusus (*special purpose*) yang digunakan dapat secepatnya kembali.

Suatu permasalahan kritis dalam flow shop adalah pengelompokan tugas-tugas yang dibutuhkan dalam stasiun kerja, sehingga dicapai suatu kondisi yang memenuhi pembatas-pembatas urutan dan terjadi keseimbangan pada tingkat output produksi. Jika tingkat output bervariasi untuk masing-masing stasiun kerja, maka hal ini berarti bahwa lintasan produksi tersebut tidak seimbang. Ketidakseimbangan lintasan akan menghasilkan aliran yang tidak teratur dan rendahnya utilisasi kapasitas yang disebabkan turunnya kecepatan aliran pada stasiun-stasiun penyebab *bottle neck* (operasi akan berjalan terputus-putus).

Problem lain pada penjadwalan flow shop adalah berhubungan dengan ketegangan yang diakibatkan susunan aliran ini terhadap pekerja. Pekerja biasanya menjadi sangat bosan karena terbatasnya variasi kerja pada tiap-tiap stasiun dan panjangnya rentang pengendalian sepanjang

lintasan produksi. Oleh karena itu manajemen dianjurkan melakukan job rotasi, mengubah lintasan produksi yang panjang menjadi segmen-segmen yang lebih pendek sehingga dapat dikendalikan oleh kelompok kecil pekerja, dan menyediakan penghargaan tingkat output produksinya *tinggi dan* berkualitas. Dengan cara ini, maka kebosanan dan rasa frustasi pekerja dapat dieliminir.



Gambar 2.1 Pola Aliran Produksi Flow Shop

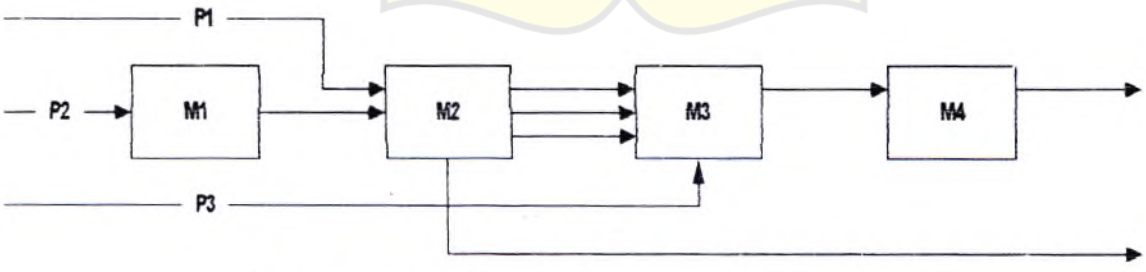
2.2.7 Penjadwalan Job Shop

Penjadwalan pada proses produksi tipe job shop lebih sulit dibandingkan penjadwalan flow shop. Hal ini disebabkan oleh 3 alasan (A.H Nasution, hal 180) :

1. Job shop menangani variasi produk yang sangat banyak, dengan pola aliran yang berbeda-beda melalui pusat-pusat kerja.
2. Peralatan pada Job shop digunakan secara bersama-sama oleh bermacam-macam order dalam prosesnya, sedangkan peralatan pada flow shop digunakan khusus hanya untuk satu jenis produk.

3. Job-job yang berbeda mungkin ditentukan oleh prioritas yang berbeda pula. Hal ini mengakibatkan order tertentu yang dipilih harus diproses seketika pada saat order tersebut ditugaskan pada suatu pusat kerja. Sedangkan pada flow shop tidak terjadi permasalahan seperti diatas karena keseragaman output yang diproduksi untuk persediaan. Prioritas order pada flow shop dipengaruhi terutama pada pengirimannya dibandingkan tanggal pemrosesan.

Faktor-faktor tersebut diatas menghasilkan sangat banyak kemungkinan kombinasi dari pembebanan (*loading*) dan urut-urutan (*sequencing*). Perhitungan dari identifikasi dan evaluasi jadwal-jadwal yang mungkin menjadi sangat sulit sehingga banyak perhatian diarahkan pada riset penjadwalan job shop. Selain itu, persiapan suatu penjadwalan job shop, penyesuaian dan pembaharuannya membutuhkan investasi yang besar untuk fasilitas komputer.

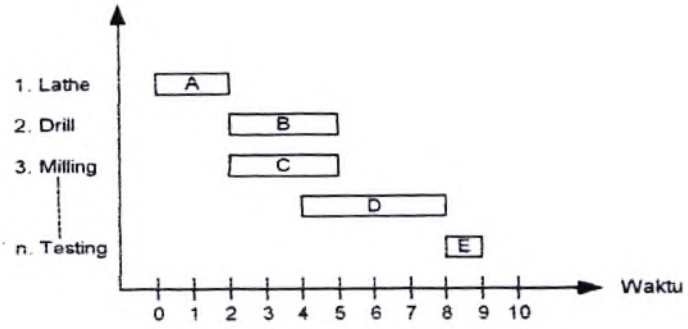


Gambar 2.2 Pola Aliran Produksi Job Shop

2.2.7.1 Job Shop Loading

Ketika order-order tiba pada suatu job shop, kegiatan pertama dari penjadwalan adalah menugaskan order-order tersebut kepada bermacam-macam pusat-pusat kerja untuk diproses. Permasalahan *loading* menjadi lebih sederhana ketika suatu job tidak dapat dipisah. Meskipun hal ini sering terjadi, biasanya suatu industri sering dalam prakteknya melakukan pemisahan job dan menugaskan bagian-bagian terpisah dari job tersebut kepada pusat-pusat kerja yang berbeda untuk tujuan meningkatkan utilisasi sumber daya. Untuk permasalahan yang sederhana dimana kita mengasumsikan tidak ada pemisahan job, maka *shop loading* dapat dibuat dengan mudah menggunakan *Gantt Chart* dan metode penugasan.

Loading dengan *Gantt Chart* merupakan cara yang paling sederhana, paling tua dan paling banyak digunakan untuk bermacam-macam aktivitas penjadwalan. Meskipun sederhana dan tervisualisasikan, *Gantt Chart* sangat lemah dalam mengevaluasi rencana-rencana alternatif untuk *loading*. Pengguna harus memakai *trial error* dalam improvisasi jadwal. Bila jumlah job meningkat, proses ini menjadi cukup sulit dan tidal layak.



Gambar 2.3 Contoh Gantt Chart (A.H Nasution, hal 182)

Loading dengan Metode Penugasan merupakan cara pembebanan pekerja-pekerja untuk job-job yang tersedia dengan tujuan meminimasi total waktu yang bisa dipakai untuk permasalahan ini.

2.2.7.2 Job Shop Sequencing

Sekali beberapa job telah ditugaskan (*loading*) pada pusat kerja tertentu, maka langkah berikutnya adalah menentukan urut-urutan memprosesnya. Pemrosesan order merupakan hal yang penting karena mempengaruhi lamanya suatu job akan diproses dalam sistem tertentu. Lamanya job dalam proses ini akan mempengaruhi batas waktu janji pengiriman kepada konsumen. Yang tidak kalah pentingnya adalah pengaruh urut-urutan pemrosesan job terhadap utilisasi sumberdaya-sumberdaya organisasi, khusunya pada kondisi suplai yang kritis.

Penjadwalan job shop melibatkan aturan-aturan prioritas sequencing. Aturan-aturan prioritas *sequencing* diaplikasikan untuk seluruh job yang sedang menunggu dalam antrian. Bila pusat kerja telah

Fungsi-fungsi tersebut dalam praktek tidak semua perusahaan akan melaksanakannya. Fungsi tersebut berlaku secara umum, kadang kala suatu perusahaan hanya memiliki beberapa fungsi.

2.2 PENJADWALAN

Dalam area shop penjadwalan produksi yang tepat waktu dan efektif sangat diperlukan untuk memenuhi order pesanan dari konsumen. Masalah yang dihadapi sekarang adalah bagaimana merencanakan penjadwalan tugas (*job*) dan mengurutkan pekerjaan (*sequence*) secara efektif, sehingga dapat menyelesaikan order tepat waktu. (Majalah Widya, hal 35).

Penjadwalan yang tidak efektif akan menghasilkan tingkat penggunaan yang rendah dari kapasitas yang ada. Fasilitas, tenaga kerja, dan peralatan akan menunggu (*idle*) untuk waktu tertentu, karena tidak ada jadwal. Sebagai akibatnya, biaya produksi membengkak. Ini dapat menurunkan efektifitas dan daya saing perusahaan. Meskipun kapasitas keseluruhan mungkin didesain agar biaya sumber daya minimal, penjadwalan yang tidak tepat dapat menyebabkan menurunnya tingkat pelayanan dan banyak hal lain secara tidak langsung.

Untuk jangka pendek, dalam rentang periode beberapa hari sampai satu bulan, perusahaan harus melakukan penjadwalan produksi untuk memenuhi order atau permintaan konsumen. Penjadwalan itu untuk melaksanakan rencana agregat dan jadwal induk produksi yang telah

bagaimana caranya mengurangi waktu peralihan agar mendekati nol. Inilah yang ditekankan oleh berbagai perusahaan, yaitu berusaha sekuat mungkin untuk mengurangi waktu peralihan, sehingga proses menjadi luwes.

Kepentingan pertama dari line proses adalah menemukan lot yang ekonomis dengan biaya terkecil. Hanya saja, persoalannya adalah, jika penjadwalan sering, maka sediaan sedikit. Sementara itu, jika penyetelan jarang, maka sediaan semakin banyak. Untuk itu, perlu adanya keseimbangan antara biaya penyetelan dan biaya mempertahankan sediaan.

2.2.4.2 *Intermitten Process*

Kegiatan tergolong dalam operasi *intermitten* adalah pabrik, rumah sakit, kantor dan sekolah. Untuk operasi jasa, "Pekerjaan" digantikan oleh "pelanggan, pasien, klien", atau apa saja yang mengalir melalui suatu proses. Sementara itu, "Pusat Kerja" digantikan oleh "ruangangan kantor, fasilitas, keahlian khusus", atau apa saja yang menjadi pusat pengolahan. Karakteristik *Intermitten Process* adalah (Hendri Tanjung, hal 445) :

a. Setiap titik yang mengalir, mempunyai banyak titik awal dan akhir, tidak bersambungan.
b. Aliran tak teratur disebabkan oleh tata letak proses menurut kelompok mesin atau keahlian dalam pusat kerja.

lowong untuk satu job baru, maka job dengan prioritas terdahulu akan diproses.

Pemilihan prioritas *sequencing* tersebut mempertimbangkan efisiensi penggunaan fasilitas dengan kriteria antara lain biaya set-up, biaya persediaan WIP, waktu menganggur, rata-rata jumlah job yang menunggu, dan sebagainya.

2.2.8 Penjadwalan N Job Pada M Mesin

Penentuan urutan (*sequence*) yang optimal dalam masalah N job M mesin memerlukan suatu prosedur penelitian yang kompleks. Pencapaian tujuan dilakukan dengan menerapkan aturan prioritas penjadwalan sebagai aturan pemilihan prioritas pekerjaan berdasarkan kedatangannya dalam sistem. Adapun aturan prioritas yang digunakan dalam penjadwalan tugas untuk menentukan tugas mana yang akan dapat didahulukan dan dikemudiankan adalah sebagai berikut (Sri Lisa Susanty, hal 37) :

1. SPT (*Shortest Processing Time*)

   Job dengan waktu proses terpendek akan diproses lebih dahulu, demikian berlanjut untuk job yang waktu prosesnya terpendek kedua. Aturan SPT ini tidak memperdulikan due date maupun kedatangan order baru.

2. LPT (*Long Processing Time*)

   Memilih proses dari suatu pekerjaan yang memiliki waktu proses terpanjang.

3. FCFS (*First Come First Serve*)

   Pekerjaan dikerjakan sesuai dengan saat kedatangannya. Pekerjaan yang datang lebih dahulu dikerjakan lebih awal.

4. EDD (*Earliest Due Date*)

   Pekerjaan dilakukan berdasarkan due date yang lebih mendesak. Aturan ini dipakai untuk meminimasikan waktu keterlambatan yang paling maksimum (*maximum lateness*).

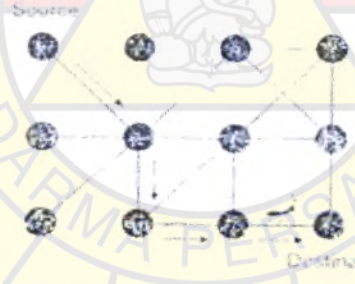
2.3 ANT COLONY OPTIMIZATION

2.3.1 Pengenalan Terhadap Ant Colony Optimization (ACO)

Ide tentang Ant Colony Optimization pertama kali dilontarkan oleh Marco Dorigo ketika sedang menjadi mahasiswa doktoral di Milan, Italia pada tahun 1991 dengan dibantu oleh Alberto Colorni dan juga Vittorio Maniezzo. Pertama kali metode *Ant Colony Optimization* (ACO) diterapkan untuk permasalahan *Travelling Salesman Problem* (TSP). Tetapi pengembangannya sudah demikian luas saat ini, sehingga tidak menjadi terbatas hanya pada masalah TSP saja, tetapi ACO juga dapat menyelesaikan permasalahan-permasalahan yang mempunyai tingkat kesulitan berupa kombinasi dan permutasi yang sangat banyak jumlahnya.

2.3.2 Cara Kerja ACO

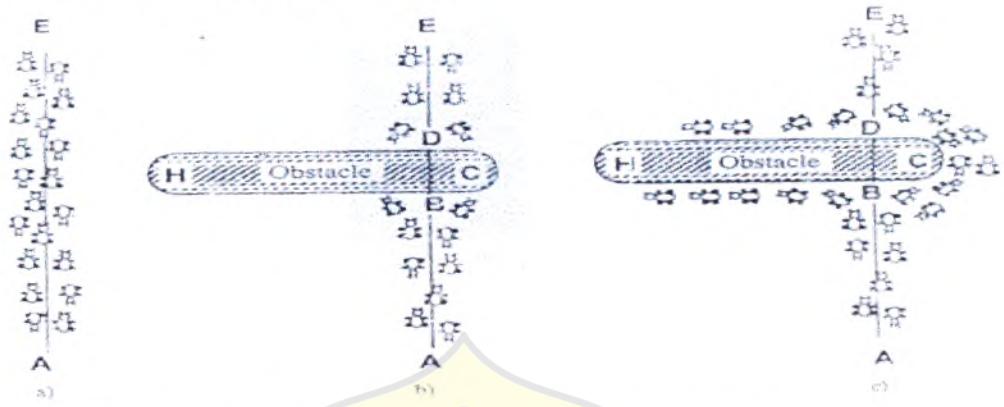
Ide dasar dari penggunaan ACO ini sesuai dengan namanya menggunakan sifat dari semut yang selalu mencari rute terpendek yang bisa digunakannya untuk menghubungkan antar tempat dimana makanannya berada dengan sarangnya. Semut selalu dapat menemukan suatu jalur yang paling pendek yang dapat ditempuh dari sumber makanan ke sarangnya. Dalam perjalannya dari sarangnya ke tempat makanan semut-semut ini meninggalkan suatu zat yang diberi nama *zat pheromone* yang akan digunakan bagi teman-temannya untuk mengikuti semut-semut yang ada di depannya untuk selanjutnya menemukan tempat makanan sesuai dengan jalur dimana terdapat zat yang ditinggalkannya tersebut.



Gambar 2.4 Semut Yang Membangun Solusi
(Dorigo, hal 3)

Gambar 2.4 menunjukkan seekor semut yang membangun solusi, yaitu suatu jalur dari suatu sumber ke suatu tujuan dari semut tersebut. Semut ini dalam perpindahannya meninggalkan suatu zat yang diberi nama zat *pheromone* yang selanjutnya akan digunakan oleh semut-semut

lain untuk mengikuti jejak semut tersebut agar sampai ke tujuannya. Zat yang ditinggalkan ini berjumlah tertentu.



Gambar 2.5 Contoh Dengan Semut Sebenarnya

a. Semut mengikuti jalur antara point A dan E
b. Sebuah benda diletakkan, semut dapat memilih untuk melewati dua buah jalur dengan probabilitas yang sama
c. Di jalur yang lebih pendek, *pheromone* diletakkan lebih banyak

Gambar 2.5 menunjukkan sifat semut yang sebenarnya dalam nyatanya. Ada sebuah jalur yang diikuti oleh semut (sebagai contoh A yaitu sumber makanan, sedangkan E adalah sarang semut). Tiba-tiba ada sebuah benda yang diletakkan sedemikian hingga jalurnya menjadi terpotong antara sarang dengan sumber makanan. Jadi pada saat posisi B, semut harus berjalan dari A ke E (atau pada posisi D yang berjalan di arah sebaliknya) harus menentukan apakah ia harus berbelok ke kiri atau ke kanan. Pilihan ini dipengaruhi oleh jumlah jejak *pheromone* yang ditinggalkan oleh semut yang sebelumnya. Level yang lebih tinggi dari

pheromone dapat menjadi stimulus yag kuat untuk semut mengikuti jalur dan akhirnya semut memilih untuk belok ke arah kanan.

Semut pertama yang mencapai point B (atau D) mempunyai probabilitas yang sama untuk berbelok ke kiri atau ke kanan (karena belum ada *pheromone* di kedua alternatif tersebut). Tetapi karena jalur BCD lebih pendek dibandingkan jalur BHD, semut pertama yang mengambil jalur ini akan sampai ke D lebih cepat dibandingkan semut yang mengikuti jalur BHD. Hasilnya yaitu semut yang kembali dari E ke D akan menemukan jejak yang lebih kuat di jalur DCB, karena sebagian dari koloni semut mempunyai peluang yang lebih besar untuk mengikuti jalur DCBA dan yang telah datang melalui jalur BCD, mereka akan (dalam probabilitas) memilih jalur DCB daripada jalur DHB.

Sebagai hasilnya, jumlah semut yang mengikuti jalur BCD per unit waktu akan menjadi lebih tinggi daripada jumlah semut yang mengikuti jalur BHD. Hal ini mengakibatkan jumlah pheromone di jalur yang terpendek meningkat dengan lebih cepat daripada yang ada dalam jalur panjang. Sebagai hasil akhir semua semut akan lebih memilih jalur yang lebih pendek.

Algoritma yang akan didefinisikan selanjutnya adalah model yang didapat dari pembelajaran terhadap koloni semut. Kita menggunakan semut-semut ini sebagai suatu agen yang dipergunakan dalam alat optimasi, sistem ini akan mempunyai sedikit perbedaan dengan semut yang sesungguhnya :

a. Semut artifisial mempunyai memori.

b. Semut-semut ini tidak sepenuhnya buta.

c. Mereka tinggal di lingkungan dimana waktu menjadi diskret.

2.3.3 Perancangan Model Penjadwalan Dengan Metode *Ant Colony*

Model penjadwalan *Ant Colony* terdiri dari lima (5) tahap, yaitu (Jurnal Sistem Produksi) :

1. Menentukan *initial sequence*
2. Inisialisasi nilai *pheromone*
3. Menentukan *Ant sequence*
4. Melakukan *Local Search*
5. Memperbaharui nilai *pheromone* yang diperoleh

2.3.3.1 Menentukan *Initial Sequence*

Tahap ini merupakan tahap pertama yang ada dalam metode *Ant Colony*. Dalam tahap ini, ditentukan *seed sequence*, yaitu sequence yang didapat setelah melakukan penentuan *ant sequence*, yang akan menjadi *initial sequence* dalam menentukan *ant sequence*. Caranya yaitu dengan menggunakan algoritma *Nawaz, Enscore, and Ham* (NEH) sebagai fungsi objektif untuk meminimasi makespan karene metode ini merupakan metode heuristik yang cukup baik dan cepat untuk diimplementasikan. Algoritma ini menggunakan aturan prioritas *Longest Processing Time*

dalam menentukan job mana yang selanjutnya akan dimasukkan ke dalam sequence. Metode NEH ini dapat dijelaskan sebagai berikut :

Step 1 : Susun job dalam urutan menurun dari jumlah waktu proses.

Step 2 : Ditentukan sebuah variabel yaitu nilai k = 2. Ambil dua (2) job pertama dari job yang sudah disusun dan jadwalkan job-job tersebut seperti hanya ada dua job tersebut.

Step 3 : Naikkan nilai k oleh i, kemudian masukkan job pertama yang ada dalam urutan kesusunan job dalam setiap slot di solusi yang sekarang. Diantara kandidat ini pilih yang terbaik. Update solusi ini dengan solusi yang baru.

Step 4 : Jika k = n penjadwalan yang terbaik ditemukan dan stop. Bila tidak kembali ke step 3. Hasil yang didapat dijadikan *initial sequence* yang nantinya akan dihitung fungsi Objektif (*makespan*) yang ada dan dijadikan nilai Z_{best}.

Inti dari metode ini yaitu pertama-tama dengan mengurutkan job-job yang ada sesuai dengan jumlah waktu dengan susunan menurun (mulai dari jumlah waktu proses paling besar ke yang paling kecil), kemudian urutan job yang dijadwalkan diambil dari urutan job berdasarkan waktu proses tadi. Cara menempatkan job itu sendiri yaitu dengan mencoba memasukkannya ke masing-masing slot yang tersedia dan melihat perubahan fungsi obyektif yang terjadi. Dari sini dipilih penempatan job

pada slot yang mempunyai fungsi obyektif terbaik dan dipilih slot tersebut sebagai tempat penempatan kandidat job.

2.3.3.2 Inisialisasi Nilai *Pheromone*

Pheromone merupakan suatu zat yang dapat mengundang semut untuk mengikuti jalan semut-semut yang sudah mendahuluinya.

Setelah ditemukan nilai Z_{best} untuk initial sequence, selanjutnya dilakukan perhitungan nilai batas atas (τ_{max}) dan batas bawah (τ_{min}) *pheromone* serta inisialisasi nilai *pheromone* untuk masing-masing posisi. Nilai *pheromone* (τ_{ik}) menunjukkan keinginan untuk menempatkan job i pada posisi k dalam suatu *sequence* dan nilai awal $\tau_{ik} = \tau_{max}$. Perhitungan dilakukan dengan rumus :

$$\tau_{max} = \frac{1}{((1-\rho).Z_{best})} \qquad (2\text{-}1)$$

$$\tau_{min} = \frac{\tau_{max}}{5} \qquad (2\text{-}2)$$

Dimana :

ρ = menunjukkan keberadaan dari jejak tersebut yang berarti (1- ρ) merupakan tingkat evaporasi *pheromone*. ρ ditentukan sebesar 0.75 yang artinya tingkat keberadaan jejak *pheromone* sebesar 0.75 setiap kali iterasi untuk perjalanan masing-masing ant *pheromone* yang sudah ada dan akan berkurang sebanyak 0.75 kalinya.

Z_{best} = menunjukkan nilai objektif terbaik yang ada pada metode NEH.

5 = adalah sebuah parameter yang menentukan apakah dalam iterasinya *pheromone* semut berada pada range yang sempit atau luas. Angka 5 ditentukan oleh *Stuetzle*.

2.3.3.3 Menentukan *Ant Sequence*

Setelah tahap inisialisasi nilai pheromone, kemudian dilakukan penentuan sequence berdasarkan ant yang dibantu dengan probabilitas kemungkinan pemilihan. Sebelumnya dicari terlebih dahulu nilai τ_{ik} (i menunjukkan job dan k menunjukkan posisi) yang merupakan penjumlahan dari nilai pheromone yang didapat dari rumus :

$$\tau_{ik} = \sum_{q=1}^{k} \tau_{iq} \quad . \qquad (2\text{-}3)$$

Penentuan ant Sequence dilakukan dengan cara memilih diantara dua metode yang ada, yaitu *MAX-MIN Ant System* (MMAS) dan juga *Summation Value*.

I. *MAX-MIN Ant System*

MAX-MIN Ant System (MMAS) pertama kali dikembangkan oleh *Stutzle & Hoss* pada tahun 1997 yang memperkenalkan empat modifikasi dalam *Ant System*, antara lain :

a) Variasi ini secara kuat mengeksploitasi jalur yang terbaik yang ditemukan. Hanya semut yang mempunyai hasil yang terbaik dalam iterasi untuk mendeposit *pheromone*, tetapi sayangnya strategi seperti ini dapat menuju ke kondisi stagnan dimana semut akan mengikuti

jalur yang sama secara terus-menerus, karena pertumbuhan jejak *pheromone* yang terlalu besar dan terus-menerus.

b) Membatasi batas kemungkinan nilai jejak *pheromone* ke interval [τ_{min}, τ_{max}].

c) Jejak *pheromone* diinisialisasi ke tingkat atas dari batas jejak *pheromone* untuk pertama kali.

d) Dalam MMAS jejak *pheromone* kembali diinisialisasi setiap kali sistem mengalami kondisi stagnansi atau ketika tidak ada jalur baru yang dapat diambil dalam sejumlah iterasi.

II. *Pheromone Summation Rule*

Variasi ini biasanya diterapkan di metode ACO yang khusus membahas masalah penjadwalan. Dalam permasalahan permutasi penjadwalan, diasumsikan, karena sifat stokastik dari algoritma ini walaupun job j mempunyai nilai *pheromone* yang lebih tinggi yaitu τ_{ij} seringkali terjadi bahwa job h, dengan tingkat τ_{ih} yang lebih rendah, yang ditugaskan untuk job i dalam penjadwalan. Kemudian, dalam banyak permasalahan penjadwalan, akan lebih baik jika menugaskan job j ke posisi yang dekat dengan posisi i. Tetapi untuk posisi 1 > i jejak *pheromone* τ_{ij} sangat rendah, ada kemungkinan bahwa job ini dijadwalkan pada bagian akhir penjadwalan, jauh dari posisi i yang menuju ke nilai penjjadwalan yang kurang optimal.

Cara menentukan *ant Sequence* yaitu dengan menggunakan bilangan acak yang dibangkitkan dengan menggunakan rumus *Linier Congruential Generator* (LCG) yang diperkenalkan oleh *Lehmer* pada tahun 1951 dibandingkan dengan suatu bilangan yang didapat dari rumus :

$$Z_i = (a \times Z_{j-1} + c) \bmod (m) \qquad (2\text{-}4)$$

Dimana nilai a, c, dan m merupakan variabel.

$$U_i = Z_i / m \qquad (2\text{-}5)$$

Dimana :

U = Uniform [0,1]

m = 2^b, b didapat berdasarkan jumlah bit per kata dari komputer yang digunakan, dalam hal ini digunakan b = 5.

c = ditentukan sedemikian hingga faktor persekutuan terbesarnya adalah 1.

a = 1 + 4 k, dimana k adalah integer.

Sehingga :

Nilai a = 1 + (4 (1) = 5

Nilai m = 32

Nilai c = 15

Nilai Z_0 = 99

Bila nilai U ≤ (n-4) / n, n menunjukkan jumlah job, maka menggunakan prinsip *Summation Value*, sedangkan bila nilai U ≥ (n-4) / n digunakan prinsip *MAX-MIN Ant System* (MMAS). Hal ini berarti, dengan nilai n yang semakin besar, maka kemungkinan menggunakan *Summation Value* semakin besar dan sebaliknya bila nilai n semakin kecil maka kemungkinan penggunaan MMAS semakin besar.

- Bila terpilih MMAS dari cara diatas, penentuan job dilakukan dengan memilih job diantara 5 job paling atas yang belum dijadwalkan yang ada di sequence terbaik yang diperoleh sampai saat itu berdasarkan nilai P_{ik} yang didapatkan dari rumus :

$$P_{ik} = \left[\frac{\tau_{ik}}{\sum_{l} \tau_{lk}} \right] \qquad (2\text{-}6)$$

  Pemikiran dasar dari penggunaan rumus ini, yaitu pekerjaan dapat diletakkan untuk posisi k dengan mempertimbangkan posisi k. Dengan kata lain jumlah dari nilai *pheromone* untuk sampai ke posisi k mengindikasikan keinginan untuk menjadwalkan pekerjaan i tidak lebih dari posisi k.

- Bila terpilih *Summation Value*, maka dari 5 job teatas yang ada di sequence terbaik yang belum dijadwalkan dipilih job yang mempunyai nilai τ_{ik} yang terbaik. Nilai τ_{ik} sendiri didapatkan dari rumus :

$$\tau_{ik} = \sum_{q=1}^{k} \tau_{iq} \qquad (2\text{-}7)$$

2.3.3.4 Melakukan *Local Search*

Setelah ditemukan *Ant Sequence* selanjutnya dilakukan *Local Search* untuk mencari hasil yang lebih optimal lagi dari *sequence* yang sudah ada sekarang. *Local Search* dilakukan dengan menggunakan metode *Job-Index-Based Local Search*. Caranya yaitu dengan melakukan *insertion* mulai dari job 1 sampai ke job n yang coba di-*insert* ke dalam masing-masing posisi yang ada yaitu dari posisi 1 sampai ke posisi k. Nilai obyektif yang diperoleh selanjutnya dibandingkan dengan nilai obyektif yang terbaik yang diperoleh sampai saat itu, kemudian bila fungsi obyektif lebih baik maka *sequence* yang ada sekarang diupdate dengan menggunakan *sequence* dengan fungsi obyektif yang lebih baik tadi. Iterasi terus dilakukan sampai seluruh job di-*insert* ke seluruh posisi.

2.3.3.5 Memperbaharui Nilai *Pheromone* Yang Diperoleh

- Setelah mendapatkan *sequence* terbaik dan nilai Z_{best} dari *sequence* tersebut, langkah selanjutnya yaitu melakukan perhitungan memperbaharui nlai *pheromone*, karena nilai nilai Z_{best} yang diperoeh sudah berubah dari yang sebelumnya. Proses perhitungan nilai *pheromone* ini dilakukan dengan menggunakan rumus :

$$\tau_{ik}^{new} = \rho \times \tau_{ik}^{old} + \left(\frac{1}{Z_{current}} \right) \qquad (2\text{-}8)$$

Untuk job i ditempatkan di posisi k dalam *sequence* terbaik, rumusnya :

$$\tau_{ik}^{new} = \rho \times \tau_{ik}^{old} \qquad (2\text{-}9)$$

- Setelah sampai di tahap ini berarti perjalanan satu semut sudah menemui jalannya sampai ke tujuan. Selanjutnya semut kedua akan memulai jalannya dengan memulai lagi dari langkah awal sampai menemukan suatu *sequence* yang optimal untuknya.
- Begitu seterusnya sampai ditemukan suatu kondisi dimana tidak dapat lagi ditemukan jalur lain yang dapat menemukan hasil yang lebih optimal, sehingga perjalanan semut tersebut dihentikan.

Nilai-nilai τ_{ik}^{new} , τ_{max}, τ_{min} langsung diaplikasikan untuk ant yang selanjutnya akan melakukan perjalanan yang sama dengan ant yang sekarang. Iterasi ini mulai dari awal sampai update *pheromone* ini terus dilakukan berulang-ulang sesuai dengan jumlah semut yang digunakan dalam membangun solusi ini.

Sampai pada akhirnya nanti ditemukan Z_{best} untuk semut yang terakhir yang akan menjadi output dari metode *Ant Colony*, atau iterasi juga dapat dihentikan bila terjadi kondisi stagnansi dalam proses iterasi yang dijalani yaitu nilai Z_{best} yang tidak pernah berubah yang berarti sudah mencapai kondisi optimal. Kondisi perhentian yang ada, yaitu :

- Algoritma ini sudah menemukan suatu solusi dengan jarak antara *lower bound* yang optimal dan berkualitas.
- Sejumlah maksimum perjalanan semut dan jumlah iterasi algoritma sudah tercapai.
- Algoritma menunjukkan kondisi stagnansi.

2.4 Perancangan Penjadwalan Dengan Metode Lain Menggunakan Bantuan Software Win QSB

WIN QSB adalah paket *software General Purpose* yang memiliki kemampuan dalam melakukan analisis dan pemecahan masalah secara kuantitatif. Fungsi-fungsi dasar dari software WIN QSB adalah (Modul Workshop WIN QSB, hal 12) :

1. WIN QSB memberikan kemudahan dalam menganalisis dan memecahkan masalah-masalah atau kasus-kasus secara kuantitatif dengan paket-paket yang disediakan sesuai dengan kriteria permasalahan yang dianalisis.
2. Untuk mempermudah penggunaan, WIN QSB dirancang menggunakan *Graphical User Interface* (GUI) yang secara otomatis memberikan template-template sehingga mempermudah pemasukan dan pembacaan data maupun hasil pengolahan data.

Metode-metode yang terdapat dalam software WIN QSB untuk masalah penjadwalan antara lain :

1. Metode CDS
2. Metode Dennenbring's
3. Metode Gupta's
4. Metode Ho and chang's
5. Metode Hundal and Rajgopal's
6. Metode Johnson's

7. Metode Palmer's
8. Metode All Hueristics
9. Metode Random Generation
10. Metode Full Enumeration

2.4.1 Penjadwalan Dengan Metode Campbell Dudek and Smith (CDS)

Metode heuristik yang paling penting untuk problem makespan adalah metode *Campbell, Dudek and Smith (CDS).* Metode CDS ini memiiiki kelebihan dalam dua hal, yaitu (Teguh Baroto, hal 184) :

1. Pemakaian aturan *Johnson's* dalam sebuah cara heuristik.
2. Biasanya menghasilkan beberapa jadwal yang dapat dipilih sebagai yang terbaik.

Algoritma CDS ini cocok untuk persoalan yang memiliki banyak tahapan (*multi-stage*) yang memakai aturan *johnson's* dan diterapkan pada masalah baru, yang diperoleh dari yang asli dengan waktu proses $t^{*}_{I,1}$ dan $t^{*}_{I,2}$.

Pada tahap I : $t^{*}_{I,1} = t^{*}_{I,1}$ dan $t^{*}_{I,2} = t^{*}_{I,m}$

Rumus diatas adalah waktu proses pada mesin pertama (M-1) dan mesin terakhir (M-2).

Pada tahap II : $t^{*}_{I,2} = t^{*}_{I,1} + t^{*}_{I,2}$ dan $t^{*}_{I,2} = t^{*}_{I,m} = t^{*}_{I,m-1}$

Oleh karena itu, aturan *johnson's* diaplikasikan pada jumlah dari dua mesin yang pertama (*first – two*) dan dua mesin terakhir (*last – two*) waktu proses operasi ke - i.

$$t^{*}_{i,1} = \sum_{k=1}^{i} t_{i,k} \text{ dan } t^{*}_{i,2} = \sum_{k=1}^{i} t_{i,m-k+1}$$

Dimana :

$t^{*}_{i,1}$ = Waktu proses pada job ke - i dengan menggunakan mesin pertama

$t^{*}_{i,2}$ = Waktu proses pada job ke - i dengan menggunakan mesin terakhir

I = Produk yang diproses (Job)

m = Jumlah mesin

K = Tahapan (*stage*)

Untuk tiap tahap k (k = 1, 2, ..., m-1), job yang diperoleh dipakai untuk menghitung sebuah makespan untuk masalah yang sesungguhnya. Setelah tahap demi tahap (m-1) dilakukan, maka dapat diketahui makespan terbaik diantara tahap (m-1).

Langkah-langkah penjadwalan produksi dengan metode CDS adalah sebagai berikut (Teguh Baroto, hal 185) :

1. Menyusun matrik n X m dari t_{ij} dimana n = jumlah job, m = jumlah mesin, dan t_{ij} = waktu pengerjaan job i pada mesin ke – j.
2. Menentukan jumlah urutan (p) untuk n job 2 mesin, dimana $p \leq m - 1$.
3. Memulai penjadwalan dengan tahap 1 (k = 1).
4. Menghitung $t^{*}_{i,1}$ (M – 1) dan $t^{*}_{i,i}$ (M – 2).

Dimana : $M - 1 = \sum_{j=1}^{k} t_{i,j}$

$M - 2 = \sum_{j=m,k+1}^{m} t_{i,j}$

5. Dengan bantuan algoritma Johnson's, n job 2 mesin, maka dapat ditentukan urutan job.
6. Jika k ≠ p. maka penghitungan kembali pada langkah ketiga dengan (k+1), jika k = p, maka perhitungan selesai.
7. Menghitung makespan (total waktu pengerjaan produk terpanjang yang berada dalam suatu sistem).
8. Memilih urutan penjadwalan yang memiliki makespan terkecil.

Campbell, Dudek and Smith mencoba algoritma mereka dan menguji *performance*-nya pada beberapa masalah. Mereka menemukan bahwa algoritma CDS efektif untuk masalah kecil maupun masalah besar.