

BAB II

LANDASAN TEORI

2.1 Penjadwalan

Menurut (Pinedo, 2002) penjadwalan selalu berhubungan dengan pengalokasian sumber daya yang ada pada jangka waktu tertentu. Hal tersebut adalah proses pengambilan keputusan yang bertujuan optimalitas. Menurut (Schroeder, 2000) penjadwalan diartikan sebagai suatu petunjuk atau indikasi apa saja yang harus dilakukan, dengan siapa, dan dengan peralatan apa yang digunakan untuk menyelesaikan suatu pekerjaan pada waktu tertentu. Keputusan dalam suatu penjadwalan yang diartikan penugasan adalah berupa mengurutkan pekerjaan (*sequencing*) dan waktu (*timing*) untuk memulai pekerjaan, di mana untuk menentukan semuanya itu harus diketahui urutan operasinya terlebih dahulu.

Penjadwalan berperan penting dalam industri manufaktur dan industri pelayanan (jasa). Penjadwalan tidak dapat lepas dari *sequencing* yaitu pekerjaan mana yang harus dikerjakan terlebih dahulu dalam suatu pesanan. Penjadwalan dapat menjadi suatu masalah apabila terdapat sekumpulan tugas yang datang secara bersamaan pada waktu tertentu, seperti per bulan, per minggu, per hari atau skala waktu lainnya, sedangkan fasilitas yang dimiliki perusahaan terbatas. Biasanya jika hal ini terjadi, maka akan diberlakukan aturan prioritas.

Untuk membuat suatu penjadwalan maka masukan yang dibutuhkan untuk membuatnya adalah mencakup jenis dan banyaknya pekerjaan yang akan diproses, urutan ketergantungan antar operasi/proses produksinya, waktu proses untuk masing-masing operasi, serta fasilitas yang dibutuhkan oleh setiap operasi. Dari masukan tersebut penjadwalan yang dihasilkan adalah berupa urutan pekerjaan yang akan dijadwalkan. Dalam membuat penjadwalan yang baik, perusahaan membutuhkan suatu perencanaan produksi dan pengendalian produksi agar fasilitas yang digunakan untuk memproduksi dapat digunakan secara efisien.

2.2 Tujuan Penjadwalan Produksi

Dalam penjadwalan terdapat tujuan-tujuan yang ingin dicapai oleh suatu perusahaan yang pastinya akan lebih menguntungkan bagi perusahaan. Tujuan adanya penjadwalan adalah untuk mengurangi waktu keterlambatan dari batas waktu yang ditentukan agar dapat memenuhi batas waktu yang disetujui dengan konsumen. Penjadwalan juga dapat meningkatkan kegunaan sumber daya yang terdapat dalam perusahaan sehingga dapat meningkatkan produktivitas mesin dan mengurangi waktu menganggur.

Dengan produktivitas mesin meningkat dan waktu menganggur berkurang, maka secara tidak langsung perusahaan dapat mengurangi ongkos produksi, dan mengurangi waktu keterlambatan. Jika tepat waktu dalam pemenuhan produk perusahaan maka hal ini dapat menjadi nilai tambah bagi perusahaan dalam hal pelayanan. Jika tujuan penjadwalan ini dapat tercapai

maka hal ini dapat juga dijadikan suatu keuntungan dan strategi bagi perusahaan dalam pemuasan pelanggan.

Secara umum, penjadwalan produksi dibuat dengan tujuan, yaitu:

1. Efisiensi pemakaian sumber daya.
2. Meminimaliasi rata-rata waktu menunggu (*Tardiness*) atau rata-rata *flowtime*.
3. Meningkatkan produktifitas mesin dengan meminimalisasi waktu menganggur mesin.
4. Memenuhi batas waktu (*due dates*).
5. Meminimalisasi rata-rata keterlambatan pekerjaan (*job lateness*).

2.3 Aturan Prioritas

Aturan prioritas memberikan panduan urutan pekerjaan yang harus dilaksanakan. Aturan prioritas mencoba untuk mengurangi waktu penyelesaian, jumlah pekerjaan dalam sistem, dan keterlambatan kerja sementara penggunaan mesin bisa maksimum. Ada beberapa cara penentuan prioritas yang dapat digunakan sebagai simulasi untuk menetapkan pedoman dispatching prioritas yang terbaik. Beberapa pedoman atau metode yang dapat digunakan adalah sebagai berikut:

1. FCFS (*First come first serve*)

Metode ini mempunyai aturan yaitu memprioritaskan pekerjaan yang datang lebih dulu untuk diproses lebih dahulu (Schroeder, 1992). metode ini mengacu kepada konsep keadilan sebagai kelebihannya, karena pekerjaan yang lebih datang lebih dahulu akan diprioritaskan untuk dikerjakan.

2. EDD (*Earliest due date*)

Metode yang memiliki aturan memprioritaskan pekerjaan yang tanggal jatuh tempo paling cepat untuk diproses terlebih dahulu (Schroeder, 1992). Pesanan yang tanggal jatuh temponya paling cepat mendapat prioritas pertama, dan pesanan yang tanggal jatuh temponya paling lama akan mendapat prioritas paling akhir.

3. SPT (*Shortest processing time*)

Metode yang memprioritaskan penyelesaian proses produksi berdasarkan waktu proses terpendek (Schroeder, 1992). Aturan ini didasarkan atas pemikiran bahwa apabila suatu pekerjaan dapat diselesaikan dengan cepat, maka mesin lain di bagian berikut akan menerima pekerjaan lebih cepat sehingga pekerjaan mengalir dengan cepat dan pemanfaatan yang tinggi.

4. LPT (*Longest processing time*)

Adalah sebuah metode yang mendahulukan penyelesaian proses produksi berdasarkan waktu proses yang paling lama.

Untuk menghitung ukuran efektivitas, rumus yang digunakan oleh metode *FCFS*, *SPT*, *EDD*, dan *LPT* adalah sebagai berikut :

a. Waktu penyelesaian rata-rata

Rumus = jumlah aliran waktu total / jumlah pekerjaan

b. Utilisasi

Rumus = jumlah waktu proses total / jumlah aliran waktu total

c. Jumlah pekerjaan rata-rata

Rumus = jumlah aliran waktu total / waktu proses pekerjaan total

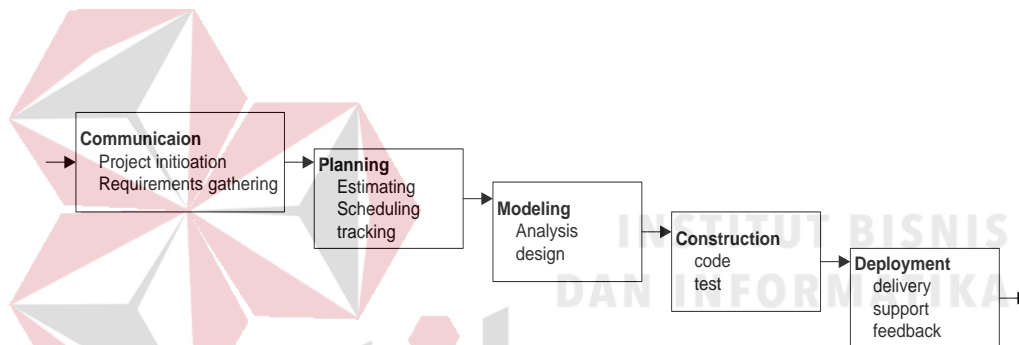
d. Keterlambatan rata-rata

Rumus = jumlah hari keterlambatan / jumlah pekerjaan

2.4 Waterfall

Menurut (Pressman, 2010) model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Berikut ada dua gambaran dari model *waterfall*.

Fase-fase dalam model *waterfall* menurut referensi *pressman* seperti terlihat pada Gambar 2.1



Gambar 2.1 *System Development Life Cycle*

A. *Communication*

Langkah ini merupakan analisis terhadap kebutuhan *software*, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan *customer*, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet.

B. *Planning*

Proses *planning* merupakan lanjutan dari proses *communication* (analysis requirement). Tahapan ini akan menghasilkan dokumen *user requirement*

atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software*, termasuk rencana yang akan dilakukan.

C. *Modeling*

Proses *modeling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan *detail* (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

D. *Construction*

Construction merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian diperbaiki.

2.5 Optimasi

Optimasi adalah tindakan untuk memperoleh hasil yang terbaik dengan keadaan yang diberikan. Dalam desain, konstruksi, dan pemeliharaan dari sistem teknik, harus diambil beberapa teknologi dan keputusan managerial dalam beberapa tahap. Tujuan akhir dari semua keputusan seperti itu adalah

meminimalkan upaya yang diperlukan atau untuk memaksimalkan manfaat yang diinginkan. Mengacu pada pendapat (Singiresu S Rao,2009) optimasi juga dapat didefinisikan sebagai proses untuk mendapatkan keadaan yang memberikan nilai maksimum atau minimum dari suatu fungsi.

2.6 Interaksi Manusia dan Komputer

Interaksi manusia dan komputer (*Human Computer Interaction-HCI*) merupakan satu disiplin ilmu yang mengkaji tentang komunikasi atau interaksi diantara pengguna dengan sistem (Alan dix, 2003). Sistem yang dimaksud adalah disini tidak terhadap kepada sistem-sistemn berkomputer saja, tetapi apa saja produk-produk yang digunakan oleh pengguna seperti kendaraan, peralatan rumah tangga, dan lain-lain. Peranan HCI adalah untuk menghasilkan sebuah sistem yang berguna, selamat,berkesan dan efektif.



Gambar 2.2 Gambar Model Interaksi Manusia dengan Komputer

Model interaksi diantara pengguna dengan sistem melibatkan tiga komponen yaitu pengguna, interaksi dan sistem itu sendiri seperti yang ditunjukkan oleh gambar 2.2. kunci utama dalam HCI adalah usability, yaitu suatu sistem harus mudah digunakan, memberi keleluasaan pada pengguna, serta mudah untuk dipelajari.

2.7 Testing Dan Implementasi

Testing adalah proses yang dibuat sedemikian rupa untuk mengidentifikasi ketidaksesuaian hasil sebuah sistem informasi dengan hasil yang diharapkan (lukman, 2009).

- A. Testing dilakukan untuk memastikan kualitas (quality assurance) yaitu menguji apakah sistem informasi yang dihasilkan sesuai dengan Testing dilakukan untuk memastikan kualitas (quality assurance).
- B. Proses menganalisa suatu entitas software untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan(defect) dan mengevaluasi fitur-fitur dari entitas software.(**Standar ANSI/IEEE 1059**).

2.8 Analisis dan Perancangan Sistem

Analisis sistem dilakukan dengan tujuan untuk dapat mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun aplikasi.

Menurut (Kendall dan Kendall, 2003) Analisis dan Perancangan Sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan

peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi.

2.9 Konsep Dasar Basis Data

Kumpulan data yang saling berhubungan (relasi). Relasi biasanya ditunjukkan dengan kunci dari tiap *file* yang ada. Dalam satu *file* terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* terdiri dari *field* yang saling berhubungan menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*.

2.10 Sistem Basis Data

Menurut (Marlinda, 2004) sistem basis data adalah suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan atau merekam serta memelihara dan operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Pada sebuah sistem basis data terdapat komponen-komponen utama yaitu Perangkat Keras (*Hardware*), Sistem Operasi (*Operating System*), Basis Data (*Database*), Sistem (Aplikasi atau Perangkat Lunak) Pengelola Basis Data (DBMS), Pemakai (*User*), dan Aplikasi (Perangkat Lunak) lain (bersifat opsional).

Tabel 2.4 Keuntungan dan Kerugian Sistem Basis Data

Keuntungan Sistem Basis Data	Kerugian Sistem Basis Data
Mengurangi Kerangkapan data, yaitu data yang sama disimpan dalam berkas data yang berbeda-beda sehingga update dilakukan berulang-ulang	Diperlukan tempat penyimpanan yang besar
Keuntungan Sistem Basis Data	Kerugian Sistem Basis Data
Mencegah ketidakkonsistenan	Diperlukan tenaga yang terampil dalam mengolah data
Keamanan data dapat terjaga, yaitu data dapat dilindungi dari pemakai yang tidak berwenang	Perangkat lunaknya mahal

2.11 Database

Menurut (Yuswanto, 2008) *database* merupakan sekumpulan data yang berisi informasi yang saling berhubungan. Pengertian ini sangat berbeda antara *database Relasional* dan *Non Relasional*. Pada *database Non Relasional*, sebuah *database* hanya merupakan sebuah *file*.

Menurut (Marlinda, 2004) *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Penyusunan satu *database* digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu *redundan* dan inkonsistensi data, kesulitan pengaksesan data, isolasi data untuk standarisasi, *multiple user* (banyak pemakai), masalah keamanan (*security*), masalah integrasi (kesatuan), dan masalah data *independence* (kebebasan data).

A. *Database Management System*

Menurut (Marlinda, 2004) *Database Management System* (DBMS) merupakan kumpulan *file* yang saling berkaitan dan program untuk mengelolanya. Basis Data adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam suatu paket program yang komersial untuk membaca data, menghapus data, dan melaporkan data dalam basis data.

Berikut ini merupakan Bahasa-bahasa yang terdapat dalam *DBMS* , yakni:

1. *Data Definition Language* (DDL)

Pola skema basis data dispesifikasikan dengan satu set definisi yang diekspresikan dengan satu bahasa khusus yang disebut DDL. Hasil kompilasi perintah DDL adalah satu set tabel yang disimpan di dalam file khusus yang disebut *data dictionary/directory*.

2. *Data Manipulation Language* (DML)

Bahasa yang memperbolehkan pemakai mengakses atau memanipulasi data sebagai yang diorganisasikan sebelumnya model data yang tepat.

3. *Query*

Pernyataan yang diajukan untuk mengambil informasi. Merupakan bagian DML yang digunakan untuk pengambilan informasi.

Fungsi-fungsi dari DBMS terdiri dari 3 yaitu *Data Definition*, *Data Manipulation*, *Data Security* dan *Integrity*. Berikut adalah penjelasan dari ketiga fungsi tersebut

1. *Data Definition*

DBMS harus dapat mengolah pendefinisian data.

2. *Data Manipulation*

DBMS harus dapat menangani permintaan-permintaan dari pemakai untuk mengakses data.

3. *Data Security* dan *Integrity*

DBMS dapat memeriksa *security* dan *integrity* data yang didefinisikan oleh *DBA*.

4. *Data Recovery* dan *Concurrency*

1. *DBMS* harus dapat menangani kegagalan-kegagalan pengaksesan basis data yang dapat disebabkan oleh kesalahan sistem, kerusakan *disk*, dan sebagainya.

2. *DBMS* harus dapat mengontrol pengaksesan data yang konkuren yaitu bila satu data diakses secara bersama-sama oleh lebih dari satu pemakai pada saat yang bersamaan.