

BAB II

LANDASAN TEORI

2.1 Penelitian Sebelumnya

Gilang (2015) pada judul tugas akhir “Rancang Bangun Sistem Informasi Penjadwalan Produksi Menggunakan Aturan Prioritas pada PT. IGLAS” (Persero). Dimana aplikasi pencatatan ini bertujuan untuk menjadwalkan produksi yang lebih sesuai dengan kriteria perusahaan sehingga dapat mengoptimalkan kapasitas mesin yang ada, sehingga semua pesanan *customer* akan terpenuhi, tepat waktu dan dapat mengurangi pembatalan serta keterlambatan terhadap pesanan.

Rudyanto (2011) melakukan penelitian tentang “Rancang Bangun Sistem Informasi Penjadwalan Produksi *Paving Block* pada CV. Eko Joyo”. Dimana sistem informasi penjadwalan produksi ini bertujuan meminimalkan keterlambatan dalam memenuhi target jatuh tempo yang sering terjadi. Metode yang digunakan adalah metode *Earliest Due Date* (EDD). Metode tersebut menghasilkan *maximum tardiness* yang paling *minimum*. Metode ini mengurutkan pekerjaan – pekerjaan berdasarkan tanggal jatuh tempo yang terdekat. Dengan metode EDD, sistem informasi penjadwalan produksi dapat meminimalkan keterlambatan *maximum*.

Penelitian pertama menggunakan lima metode untuk melakukan penjadwalan pada bagian produksi. Metode tersebut antara lain adalah *Earliest Due Date* (EDD), *Longest Processing Time* (LPT), *Shortest Processing Time* (SPT), *Critical Ratio* (CR), dan *Longest Processing Time* (LPT). Kemudian kelima metode tersebut dibandingkan, sehingga mendapatkan metode yang paling cocok

digunakan untuk menjadwalkan pekerjaan pada PT. IGLAS (Persero). Sedangkan pada penelitian kedua penjadwalan produksi dengan menggunakan metode EDD. Namun pada penelitian yang ada pada PT Bina Megah Indowood bukan hanya menggunakan EDD yang digunakan, namun *Shortest Processing Time* (SPT), *Largest Processing Time* (LPT), *First Come First Served* (FCFS), dan *Least Slack First* (LSF). Kemudian kelima metode tersebut dibandingkan, sehingga mendapatkan metode yang paling cocok digunakan untuk menjadwalkan suatu pekerjaan pada PT Bina Megah Indowood.

2.2 Penjadwalan Produksi

Menurut Nasution (2003) penjadwalan produksi dapat didefinisikan sebagai proses pengalokasian sumber daya dan mesin yang ada untuk menyelesaikan semua pekerjaan dengan mempertimbangkan batasan-batasan yang ada. Pada saat merencanakan suatu jadwal produksi, ketersediaan sumber daya yang dimiliki harus dipertimbangkan dengan baik.

Tujuan dari penjadwalan produksi adalah:

1. Meningkatkan penggunaan sumber daya atau mengurangi waktu tunggu, sehingga total waktu proses dapat berkurang dan produktivitasnya dapat meningkat.
2. Mengurangi persediaan barang setengah jadi atau mengurangi sejumlah pekerjaan yang menunggu dalam antrian ketika sumber daya yang ada masih mengerjakan tugas yang lain.
3. Mengurangi beberapa keterlambatan pada pekerjaan yang mempunyai batas waktu penyelesaian sehingga akan meminimaliasi biaya keterlambatan.

4. Membantu pengambilan keputusan mengenai perencanaan penjadwalan pabrik dan jenis penjadwalan yang dibutuhkan sehingga penambahan biaya dapat dihindarkan.

2.3 Metode-Metode Penjadwalan Produksi

Dalam membuat penjadwalan produksi, khususnya penjadwalan produksi di pabrik biasanya dilakukannya dengan bantuan *sequencing* yaitu menentukan urutan pekerjaan yang harus dikerjakan terlebih dahulu agar setiap aktivitas produksi dapat berjalan lebih efektif dan efisien. Terdapat beberapa macam aturan dalam *sequencing* tergantung pada tujuannya.

Beberapa metode sederhana dapat diterapkan untuk memilih satu pekerjaan dari sekumpulan pekerjaan yang menunggu untuk diproses di mesin yang sama pada suatu waktu tertentu. Beberapa metode yang biasa diterapkan menurut Buffa & Sarin (1996) adalah:

2.3.1 *First Come First Served* (FCFS)

Prioritas diberikan kepada pekerjaan yang pertama tiba di mesin. Artinya pekerjaan yang pertama datang dimesin, maka akan diproses terlebih dahulu. Aturan ini cocok untuk tipe organisasi dimana konsumennya lebih mementingkan waktu pelayanan.

2.3.2 *Least Slack First* (LSF)

Prioritas diberikan kepada pekerjaan yang waktu senggangnya terkecil. Waktu senggang (*slack*) adalah selisih antara waktu jatuh tempo (*due date*) dan lama pengerjaan pekerjaan.

2.3.3 Shortest Processing Time (SPT)

Prioritas diberikan kepada pekerjaan dengan waktu pemrosesan paling singkat di mesin yang bersangkutan. Aturan ini biasanya meminimasi *work in process*, rata-rata keterlambatan (*mean latenes*), dan waktu penyelesaian rata-rata (*mean flow time*) produk.

2.3.4 Earliest Due Date (EDD)

Prioritas diberikan kepada pekerjaan dengan tempo paling dini. Metode tersebut menghasilkan *maximum tardiness* yang paling *minimum*. Metode ini mengurutkan pekerjaan-pekerjaan berdasarkan tanggal jatuh tempo yang terdekat. Dengan metode EDD, sistem informasi penjadwalan produksi dapat meminimalkan keterlambatan *maximum*.

2.3.5 Largest Processing Time (LPT)

Pekerjaan yang memiliki waktu pemrosesan paling lama akan dikerjakan terlebih dahulu, semakin lama, semakin besar pekerjaan, seringkali sangat penting dan kemudian dipilih lebih dahulu.

2.4 Sistem

Sistem adalah salah satu bagian dari istilah sistem informasi di mana sistem berperan penting dalam sebuah perusahaan. Menurut Murdick (1991) mengatakan bahwa sistem adalah seperangkat elemen yang membentuk kumpulan atau *procedure* bagian-bagian pengolahan yang mencari suatu tujuan bagian atau tujuan bersama dengan mengoperasikan data dan barang pada waktu rujukan tertentu untuk menghasilkan informasi.

Sistem itu sendiri memiliki beberapa karakteristik atau sifat-sifat tertentu. Menurut Jogiyanto (2005) sistem mempunyai karakteristik atau sifat-sifat tertentu, yakni:

1. Komponen

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sub sistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu mempengaruhi proses sistem secara keseluruhan.

2. Batasan sistem

Batasan sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan luar sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas sistem yang mempengaruhi operasi. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat tidak menguntungkan sistem tersebut. Lingkungan luar yang menguntungkan berupa energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara, sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung sistem

Penghubung (*interface*) merupakan media penghubung antara satu sub sistem dengan sub sistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu sub sistem ke sub sistem yang lainnya. Dengan penghubung satu sub sistem dapat berintegrasi dengan sub sistem yang lainnya membentuk satu kesatuan.

5. Masukan sistem

Masukan sistem adalah energi yang masuk kedalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*), dan masukan sinyal (*signal input*). *Maintenance input* adalah pengendalian data yang dimasukkan agar dapat beroperasi.

6. Keluaran sistem

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolahan sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

8. Sasaran sistem

Sebuah sistem sudah tentu mempunyai sasaran ataupun tujuan. Dengan adanya sasaran sistem, maka kita dapat menentukan masukan yang dibutuhkan sistem dan keluaran apa yang akan dihasilkan sistem tersebut dapat dikatakan berhasil apabila mencapai atau mengenai sasaran ataupun tujuan.

2.5 Pengertian Aplikasi

Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi suatu pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu menurut Noviansyah (2008). Aplikasi *software* yang dirancang untuk suatu tugas khusus dapat dibedakan menjadi dua jenis, yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi *software* paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

2.6 Bahasa Pemrograman

Bahasa pemrograman merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang *programmer* dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan atau diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi. Berikut macam-macam bahasa pemrograman:

1. Bahasa pemrograman JAVA

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Java tidak boleh disalah pahami sebagai JavaScript. JavaScript adalah bahasa *scripting* yang digunakan oleh *web browser*.

2. Bahasa pemrograman PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari *web*.

3. Bahasa pemrograman HTML

HyperText Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman *web* dan menampilkan berbagai informasi di dalam sebuah *browser* internet.

Pada penelitian ini menggunakan bahasa pemrograman Visual Basic .NET dikarenakan spesifikasi komputer pada PT. Bina Megah Indowood mendukung untuk aplikasi berbasis *desktop* dan perusahaan tidak berkeinginan untuk menambah nilai investasi baru pada penjadwalan produksi.

Menurut Hidayatullah(2014) Visual Basic .NET adalah Visual Basic yang direkayasa kembali untuk digunakan pada *platform* .NET sehingga aplikasi yang dibuat menggunakan Visual Basic .NET dapat berjalan pada sistem komputer apa pun, dan dapat mengambil data dari *server* dengan tipe apa pun asalkan terpasang .NET Framework.

Berikut ini perkembangan Visual Basic .NET:

- a. Visual Basic .NET 2002 (VB 7.0)
- b. Visual Basic .NET 2003 (VB 7.1)
- c. Visual Basic 2005 (VB 8.0)
- d. Visual Basic 2008 (VB 9.0)
- e. Visual Basic 2010 (VB 10.0)
- f. Visual Basic 2012 (VB 11.0)
- g. Visual Basic 2013

Pada umumnya Visual Basic .NET terpaket dalam Visual Studio .NET.

Pada distribusinya, terdapat berbagai versi Visual Studio .NET yaitu versi *professional*, *premium* dan yang paling lengkap adalah versi *ultimate*.

Kelebihan Visual Basic .NET:

1. Sederhana dan mudah dipahami
2. Mendukung GUI
3. Menyederhanakan *deployment*
4. Menyederhanakan pengembangan perangkat lunak
5. Mempermudah pengembangan aplikasi berbasis web
6. Migrasi ke VB .NET dapat dilakukan dengan mudah
7. Banyak digunakan oleh *programmer-programmer* di seluruh Indonesia

2.7 Konsep Basis Data

2.7.1 Sistem Basis Data

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

Pada sebuah sistem basis data terdapat komponen-komponen utama yaitu: perangkat keras (*hardware*), sistem operasi (*operating system*), basis data (*database*), sistem (perangkat lunak) pengelola basis data (DBMS), pemakai (*user*), aplikasi lain (bersifat operasional).

Keuntungan sistem basis data adalah:

- a. Mengurangi redundansi data, yaitu data yang sama disimpan dalam berkas data yang berbeda-beda sehingga pembaruan dilakukan berulang-ulang.
- b. Menjaga konsistensi data.
- c. Keamanan data dapat terjaga.
- d. Integritas dapat dipertahankan.
- e. Data dapat digunakan bersama-sama.
- f. Menyediakan *recovery*.
- g. Memudahkan penerapan standarisasi.
- h. Data bersifat mandiri (*data independence*).

- i. Keterpaduan data terjaga, memelihara data berarti data harus akurat. Hal ini sangat erat hubungannya dengan pengontrolan kerangkapan data dan pendidikan keselarasan data.

Kerugian sistem basis data adalah:

- a. Diperlukan tempat penyimpanan yang besar.
- b. Diperlukan tenaga yang terampil dalam mengolah data.
- c. Perangkat lunaknya relatif mahal.

Kerusakan sistem basis data yang dapat mempengaruhi departemen atau bagian yang terkait.

2.7.2 Database Management System (DBMS)

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

Penyusunan data yaitu redundansi untuk mengatasi masalah-masalah pada penyusunan data yaitu: redundansi dan inkosistensi data, kesulitan pengaksesan data, isoalasi data untuk standarisasi, *multiple user* (banyak pemakai), dan masalah keamanan, masalah integrasi, dan masalah data *independence* (kebesaran data). Berikut adalah *database* yang pakai untuk membuat sebuah aplikasi tersebut: Microsoft SQL Server 2008 adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi basis data. Contoh DBMS lainnya

adalah: MySQL, PostgreSQL, MS Access dari Microsoft, DB2 dari IBM, Oracle, Dbase, dsb.

Pada penelitian ini menggunakan basis data SQL Server 2008 selain mudah terkoneksi dengan bahasa pemrograman Visual Basic .NET, SQL Server 2008 memiliki keunggulan-keunggulan lain menurut Hidayatullah (2014) yaitu:

1. Kompatibilitasnya dengan Power Designer 15 lebih terjamin.
2. Tidak perlu menginstal Windows 7 Service Pack 1 jika pengguna Windows
3. Kemampuan MS SQL Server 2008 sudah lebih dari cukup untuk memenuhi kebutuhan pembuatan aplikasi *database* sehari-hari semacam aplikasi penjualan.
4. MS SQL Server 2008 lebih ringan dibanding MS SQL Server 2012.

2.7.3 Desain Sistem

Setelah tahap analisa sistem selesai dilakukan, maka analisis sistem telah mendapatkan gambaran yang jelas apa yang harus dikerjakan. Kemudian memikirkan bagaimana membentuk sistem tersebut. Menurut Jogiyanto (2005) desain sistem dapat diartikan sebagai berikut:

- a. Tahap setelah analisis dari siklus pengembangan sistem.
- b. Pendefinisian dari kebutuhan-kebutuhan fungsional.
- c. Persiapan untuk rancang bangun implementasi.
- d. Menggambarkan bagaimana suatu sistem dibentuk.
- e. Berupa gambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

- f. Menyangkut konfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem.

2.8 Tahapan Software Development Life Cycle (SDLC)

2.8.1 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dapat diartikan sebagai properti yang harus dipamerkan dalam rangka memecahkan beberapa masalah di dunia nyata (IEEE Computer Society, 2014). Dalam menentukan kebutuhan perangkat lunak, yang pertama perlu harus diperhatikan setelah definisi dari kebutuhan perangkat lunak adalah jenis dari kebutuhan tersebut seperti apakah produk atau proses, fungsional atau non-fungsional, dan properti yang akan muncul. Keseluruhan proses tersebut dapat menjelaskan perbedaan antara kebutuhan sistem dan perangkat lunak.

Kedua yaitu, proses dari kebutuhan itu sendiri. Didalamnya digambarkan model, aktor, dukungan manajemen kualitas dan pengembangan dari proses itu sendiri. Ketiga yaitu, elisitasi kebutuhan yang menjelaskan darimana kebutuhan perangkat lunak berasal dan bagaimana caranya mendapatkannya. Keempat yaitu, analisis kebutuhan yang membahas konflik antar kebutuhan, interaksi perangkat lunak dengan lingkungan sekitar, dan mengkolaborasikan antara kebutuhan sistem dengan perangkat lunak. Selain itu, termasuk di dalamnya klasifikasi kebutuhan, pemodelan konseptual, desain arsitektur dan alokasi kebutuhan, dan negosiasi kebutuhan.

Kelima yaitu, spesifikasi kebutuhan yang menghasilkan dokumen spesifikasi kebutuhan perangkat lunak. Keenam yaitu, validasi kebutuhan yang memastikan kebutuhan perangkat lunak benar-benar telah sesuai sebelum

digunakan. Yang terakhir, ketujuh yaitu, pertimbangan praktis, yang menggambarkan beberapa topik yang perlu dipahami dalam pelaksanaannya. Topik itu seperti sifat berulangnya sebuah proses, manajemen dan pemeliharaan, dan pengukuran kebutuhan

2.8.2 Analisis dan Desain Perangkat Lunak

a. *System Flow*

System flow atau bagan alir sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *System flow* menunjukkan urutan-urutan dari prosedur yang ada di dalam sistem dan menunjukkan apa yang dikerjakan sistem. Simbol-simbol yang digunakan dalam *system flow* ditunjukkan pada Gambar 2.1.

Mengenai penjelasan dari simbol-simbol yang digunakan dalam *system flow* adalah sebagai berikut:

1. Simbol Dokumen

Menunjukkan dokumen *input* dan *output* baik untuk proses manual atau komputer.

2. Simbol Kegiatan Manual

Menunjukkan pekerjaan manual.

3. Simbol Simpanan *Offline*

Menunjukkan *file* non-komputer yang diarsip.

4. Simbol Proses

Menunjukkan kegiatan proses dari operasi program komputer.

5. Simbol *Database*

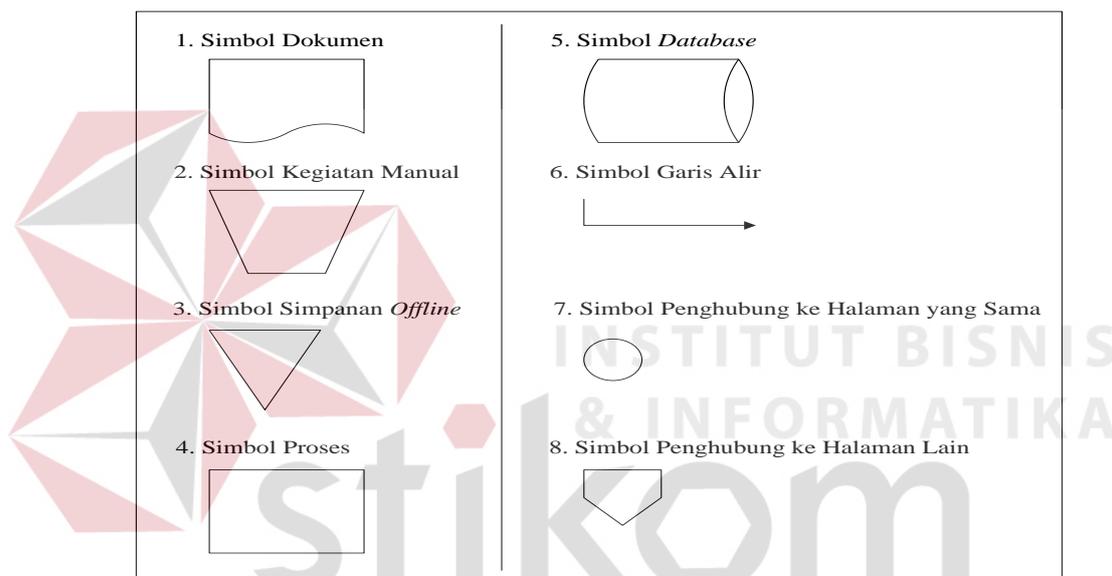
Menunjukkan tempat untuk menyimpan data hasil operasi komputer.

6. Simbol Garis Alir

Menunjukkan arus dari proses.

7. Simbol Penghubung

Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain.



Gambar 2.1 Simbol-Simbol Pada *System Flow*

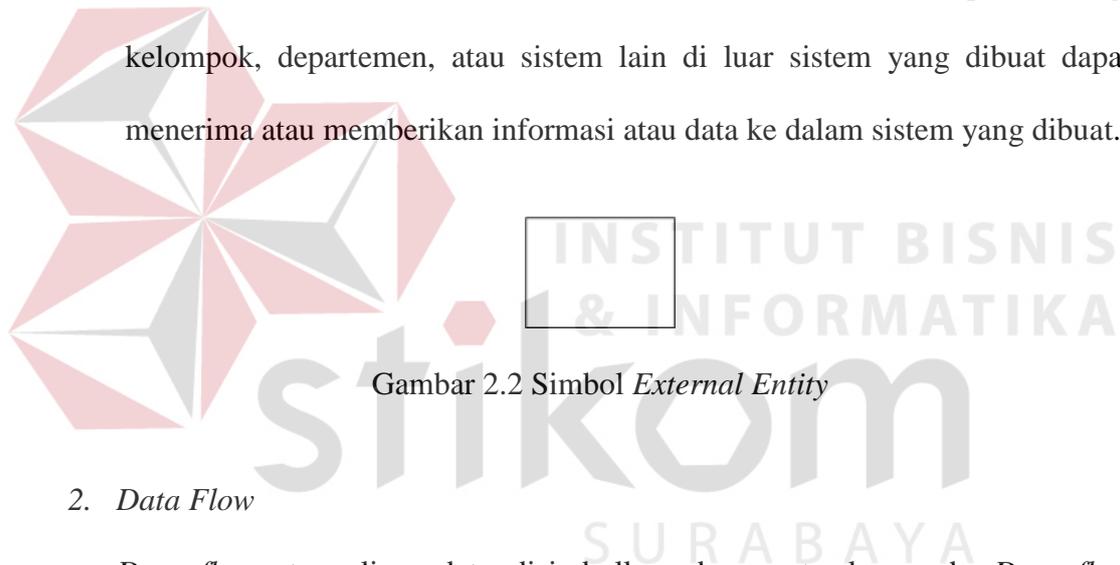
b. *Data Flow Diagram* (DFD)

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik di tempat data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur dan dapat mengembangkan arus data di dalam sistem dengan terstruktur dan jelas.

DFD fokus pada aliran data dari dan ke dalam sistem serta memproses data yang mengalir tersebut (Kendall dan Kendall, 2003). Simbol-simbol dasar dalam DFD yaitu:

1. *External Entity*

Suatu *External entity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat. Gambar 2.2 merupakan simbol entitas dalam DFD. Suatu *external entity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat.



Gambar 2.2 Simbol *External Entity*

2. *Data Flow*

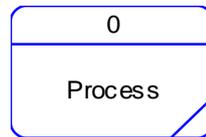
Data flow atau aliran data disimbolkan dengan tanda panah. *Data flow* menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses. Gambar 2.3 merupakan simbol *data flow*.



Gambar 2.3 Simbol *Data Flow*

3. *Process*

Suatu *process* meliputi beberapa tindakan atau sekelompok tindakan dari arus data yang masuk untuk dijalankan atau diproses agar menghasilkan arus data yang akan keluar dari proses. Gambar 2.4 merupakan simbol *process*.



Gambar 2.4 Simbol *Process*

4. *Data Store*

Data store adalah simbol yang digunakan untuk melambangkan proses penyimpanan data. Suatu nama perlu diberikan pada *data store* untuk menunjukkan nama dari *file*-nya. Gambar 2.5 merupakan simbol *file* penyimpanan atau *data store* yang dapat berupa hal-hal sebagai berikut, sebagai gambaran:

1. Suatu *file* atau *database* di sistem komputer.
2. Suatu arsip atau catatan manual.
3. Suatu tabel acuan manual.



Gambar 2.5 Simbol *Data Store*

Berikut ini adalah urutan langkah bagaimana menggambarkan suatu sistem pada DFD:

1. *Context Diagram*

Context diagram merupakan langkah pertama dalam pembuatan *data flow diagram*. Pada *context diagram* dijelaskan sistem apa yang dibuat dan *entity* apa saja yang digunakan. Dalam *context diagram* harus ada arus data yang masuk dan arus data yang keluar.

2. *Data Flow Diagram Level 0*

DFD level 0 adalah langkah selanjutnya setelah *context diagram*. Hal yang digambarkan dalam diagram level 0 ini adalah proses utama dari sistem serta hubungan *entity*, *process*, *data flow* dan *data store*.

3. *Data Flow Diagram Level 1*

DFD level 1 merupakan penjelasan dari DFD level 0. Pada proses ini dijelaskan proses apa saja yang dilakukan pada setiap proses yang terdapat di DFD level 0.

c. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) adalah gambaran pada sistem yang di dalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan terdefiniskan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai *attribute* yang merupakan ciri *entity* tersebut. Menurut Marlinda (2004), *attribute* memiliki pengertian kolom di sebuah relasi. Macam-macam *attribute* yaitu:

1. *Simple Attribute*

Attribute ini merupakan *attribute* yang unik dan tidak dimiliki oleh *attribute* lainnya, misalnya *entity* mahasiswa yang *attribute*-nya NIM.

2. *Composite Attribute*

Composite attribute adalah *attribute* yang memiliki dua nilai harga, misalnya nama besar (nama keluarga) dan nama kecil (nama asli).

3. *Single Value Attribute*

Attribute yang hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya umur (tanggal lahir).

4. *Multi Value Attribute*

Multi value attribute adalah *attribute* yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya pendidikan (SD, SMP, dan SMA).

5. *Null Value Attribute*

Null value attribute adalah *attribute* yang tidak memiliki nilai harga, misalnya *entity* tukang becak dengan *attribute*-nya pendidikan (tanpa memiliki ijazah).

Relasi adalah hubungan antar *entity* yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar *entity*. Macam-macam relasi itu sendiri antara lain:

1. *One To One* (1:1)

Relasi dari *entity* satu dengan *entity* dua adalah satu berbanding satu. Contoh: Pada pelajaran privat, satu guru mengajar satu siswa dan satu siswa hanya diajar oleh satu guru.

2. *One To Many* (1:m)

Relasi antara *entity* yang pertama dengan *entity* yang kedua adalah satu berbanding banyak atau dapat pula dibalik, banyak berbanding satu. Contoh:

Pada sekolah, satu guru mengajar banyak siswa dan banyak siswa diajar oleh satu guru.

Entity relationship diagram ini diperlukan agar dapat menggambarkan hubungan antar *entity* dengan jelas, dapat menggambarkan batasan jumlah *entity* dan partisipasi antar *entity*, mudah dimengerti pemakai dan mudah disajikan oleh perancang *database*. Untuk itu *entity relationship diagram* dibagi menjadi dua jenis model, yaitu:

1. *Conceptual Data Model (CDM)*

CDM adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

2. *Physical Data Model (PDM)*

PDM adalah jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

2.9 Teknik Observasi

Teknik observasi merupakan teknik pengumpulan data dengan menggunakan panca indra, jadi tidak hanya pengamatan menggunakan mata. Mendengarkan, mencium, mengecap, dan meraba termasuk salah satu bentuk observasi. Instrumen yang digunakan dalam observasi adalah panduan pengamatan dan lembar pengamatan, serta bisa juga berupa catatan singkat mengenai hal-hal apa saja yang diobservasi (Suliyanto 2006).

Observasi sering digunakan sebagai teknik pengumpulan data tambahan selain wawancara, namun ada juga yang menggunakan observasi tanpa

menggunakan wawancara. Di dalam melakukan observasi, panca indra yang paling berperan adalah pengamatan dengan mata atau melihat.

2.10 Teknik Wawancara

Teknik wawancara merupakan teknik pengambilan data oleh peneliti dengan langsung berdialog dengan responden untuk menggali informasi dari responden. Dalam wawancara, peneliti tidak harus bertatap muka secara langsung, tetapi dapat melalui media tertentu misalnya telepon, *teleconference*, *chatting* melalui internet, bahkan melalui *short message service* (SMS) dan *e-mail*.

(Suliyanto 2006).

Teknik ini merupakan salah satu teknik yang paling sering digunakan oleh peneliti untuk mendapatkan informasi-informasi yang dibutuhkan terkait penelitian yang dilakukan. Di dalam dunia TI, para pengembang sebuah sistem sering menggunakan teknik ini untuk menggali informasi yang dibutuhkan *stakeholder* atau pemilik kepentingan.

2.11 Analisis dan Perancangan Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya (Jogiyanto 2005). Tahap analisis dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem.

Dalam melakukan analisa dan perancangan sistem diperlukan ketelitian yang sangat detail, karena dapat mempengaruhi hasil yang akan diciptakan. Jika analisa yang di lakukan sudah sesuai dengan prosedur maka sistem yang akan dirancang akan sesuai dengan apa yang akan dirancang sebelumnya.

Adapun langkah-langkah dasar dalam analisis sistem yang harus dilakukan adalah sebagai berikut:

- a. *Identify*, yaitu mengidentifikasi masalah.
- b. *Understand*, yaitu memahami kerja dari sistem yang ada.
- c. *Analyze*, yaitu menganalisis sistem.
- d. *Report*, yaitu membuat laporan hasil analisis.

Analisa dan perancangan sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi (Kendall & Kendall, 2003).

Desain sistem dapat didefinisikan sebagai penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah dari satu kesatuan yang utuh dan berfungsi. Menurut Jhon Burch dan Gary Grudnitski yang telah diterjemahkan oleh Jogiyanto (2005) dalam bukunya yang berjudul “Analisis dan Desain Sistem Informasi”.

2.12 Black Box Testing

Menurut Rizky (2011), pengertian dari *black box testing* adalah suatu tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Berdasarkan hal tersebut, para *tester* memandang perangkat lunak

seperti layaknya “kotak hitam” yang tidak terlihat isinya, tetapi mendapat proses *testing* bagian luarnya saja. *Black box testing* hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah ditentukan pada awal perancangan.

Keuntungan dari jenis *testing* ini antara lain:

1. Anggota tim *tester* tidak harus dari seseorang yang memiliki kemampuan teknis program.
2. Kesalahan dari perangkat lunak ataupun *bug* sering ditemukan oleh komponen *tester* yang berasal dari pengguna.
3. Hasil dari *black box testing* dapat memperjelas kontradiksi ataupun kerancuan yang mungkin timbul dari eksekusi sebuah perangkat lunak.
4. Proses *testing* dapat dilakukan lebih cepat dibandingkan *white box testing*.

Alasan penulis menggunakan *black box testing* dikarenakan dapat dengan mudah mengetahui fungsi aplikasi yang salah, dapat dengan mudah mengetahui kesalahan pada tampilan, dapat dengan mudah mengetahui kesalahan pada *database*, dan dapat dengan mudah mengetahui kesalahan inisialisasi dan tujuan akhir.