

## **BAB II**

### **TINJAUAN PUSTAKA**

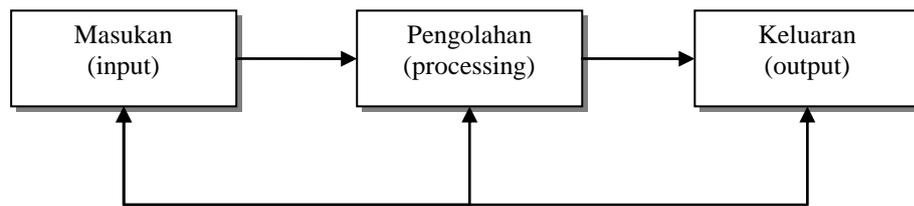
#### **II.1. Analisis dan Perancangan Sistem Secara Umum**

Analisis Sistem didefinisikan untuk memahami dan menspesifikasikan dengan detail apa yang harus dilakukan oleh sistem. Sementara sistem desain diartikan untuk menjelaskan dengan detail bagaimana bagian-bagian dari sistem informasi diimplementasikan. Atau bisa didefinisikan sebagai berikut : Analisis mendefinisikan masalah (*from requirements to specification*) dan Design memecahkan masalah (*from specification to implementation*). (Hanif Al Fatta, 2007 : 24)

#### **II.2. Sistem**

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain. Sistem dapat didefinisikan sebagai sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. (Eko Nugroho, 2008 : 17)

Susunan suatu sistem pada dasarnya terdiri dari unsur-unsur seperti unit masukan (*input*), unit pengolahan (*processing*), serta unit keluaran (*output*). Input atau masukan masuk ke dalam sistem melalui unit input. Selanjutnya, input di proses oleh unit pemroses dan hasilnya ditampilkan ataupun dicetak keluar melalui unit output.



**Gambar II.1. Model Sistem**

(Sumber: Hanif Al Fatta, 2007 : 4)

### II.2.1. Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem yang lainnya :

1. Batasan (*boundary*) : Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*environment*) : Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*input*) : Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*output*) : sumber daya atau produk (informasi, laporan, dokumen, tampilan layer computer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*component*) : Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.

6. Penghubung (*interface*) : Tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*storage*) : Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama. (Hanif Al Fattah, 2007 : 5)

### **II.3. Sistem Informasi**

Untuk memahami pengertian sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi. Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. Dari pengertian di atas kita dapat mendefinisikan sistem informasi sebagai integrasi antara orang, data, alat dan prosedur yang bekerja sama dalam mencapai suatu tujuan. Jadi sistem informasi adalah data yang telah diproses, atau data yang memiliki arti yang didalamnya terdapat elemen orang, alat, data dan prosedur atau cara. (Eko Nugroho, 2008 : 17)

Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi subsistem suatu perusahaan.

#### **II.4. Sistem Informasi Akuntansi**

Sistem Informasi Akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Misalnya : salah satu input dari sistem informasi akuntansi pada sebuah toko baju seperti transaksi penjualan.

Kita memproses transaksi dengan mencatat penjualan tersebut ke dalam jurnal penjualan, mengklasifikasikan transaksi dengan menggunakan kode rekening, dan memposting transaksi ke dalam jurnal. Kemudian secara periodik sistem akan menghasilkan output berupa laporan keuangan yang terdiri dari neraca dan laporan laba rugi. (Anastasia Diana, 2011 : 4)

#### **II.5. Sistem Informasi Produksi dan Harga Pokok Produksi**

Sistem Informasi Produksi ini memberikan dukungan penyediaan informasi untuk kegiatan produksi. Ada empat informasi penting di bidang produksi, yaitu informasi tentang proses produksi, informasi tentang persediaan produksi, informasi tentang kualitas hasil produksi dan informasi tentang biaya produksi. (Eko Nugroho, 2008 : 137)

Harga Pokok Produksi adalah biaya manufaktur yang berkaitan dengan barang-barang yang diselesaikan dalam periode tertentu. (Garrison, 2006 : 78)

Kebanyakan perusahaan membagi biaya produksi kedalam tiga kategori yaitu : bahan langsung (*direct material*), tenaga kerja langsung (*direct labor*) dan biaya overhead pabrik (*manufacturing overhead*) untuk membuat perencanaan, pengendalian, pengarahan dan membuat keputusan.

### II.5.1. Metode Harga Pokok Produksi

#### 1. Harga Pokok Produksi Berdasarkan Pesanan (*job order costing*)

Digunakan untuk perusahaan yang memproduksi berbagai produk selama periode tertentu. Biaya di telusuri dan dialokasikan ke pekerjaan dan biaya untuk menyelesaikan pekerjaan tersebut dibagi dengan jumlah unit yang dihasilkan untuk menghasilkan harga rata-rata per unit. (Garrison, 2006 : 123)

$$\text{Biaya per unit} = \frac{\text{Total Biaya Produksi}}{\text{Total Unit yang Dipesan}}$$

#### 2. Harga Pokok Produksi Berdasarkan Proses (*process costing*)

Digunakan dalam perusahaan yang memproduksi satu jenis produk dalam jumlah besar dalam jangka panjang. Prinsip dasar dari perhitungan biaya ini adalah mengakumulasikan biaya dari operasi atau departemen tertentu selama satu periode penuh dan kemudian membaginya dengan jumlah unit yang di produksi selama periode tersebut. (Garrison, 2006 : 123)

$$\text{Biaya per unit} = \frac{\text{Total Biaya Produksi}}{\text{Total Unit yang Diproduksi}}$$

### II.6. Basis Data (*Database*)

*Database* merupakan sekumpulan data berbentuk tabel yang digunakan untuk menyimpan suatu informasi. Penggunaan database tidak bisa dipungkiri memang banyak sekali digunakan hampir disemua jenis aplikasi saat ini. Mulai dari aplikasi multimedia, browser web, aplikasi penjualan dan kepegawaian. (Wahana Komputer, 2010 : 150)

Pada era kemajuan teknologi seperti sekarang ini, nilai informasi sangatlah penting, terlebih bagi kemajuan perusahaan. Oleh karena itu penggunaan dan penguasaan database sangat penting. Dalam database ada sebutan-sebutan untuk satuan data yaitu:

1. Karakter, ini adalah satuan data terkecil. data terdiri atau susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
2. *Field*, adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama siswa, tanggal lahir, dan lain-lain. Dalam dunia perancangan database, *field* juga disebut atribut. Bila dipandang dari sudut pemrograman berorientasi objek maka name dan properti tipe. Properti name atau nama adalah properti dari *field* yang berisi *field* yang mewakili data sejenis yang disimpannya. Sedangkan properti tipe adalah properti yang mengatur tipe data dari data yang akan ditampungnya. Misalnya nama *field* nya adalah nama siswa maka tipe datanya adalah char, bila nama *field* nya adalah tanggal lahir maka tipe datanya adalah date.
3. *Record*, adalah kumpulan dari *field*. Pada *record* anda dapat menemukan banyak sekali informasi penting dengan cara mengombinasikan *field-field* yang ada.
4. Tabel, adalah sekumpulan dari *record-record* yang memiliki kesamaan entity dalam dunia nyata. Kumpulan dari tabel adalah database, wujud fisik sebuah database dalam komputer adalah sebuah file yang didalamnya terdapat berbagai tingkatan data yang telah disebutkan di atas.

5. File, adalah bentuk fisik dari penyimpanan data. File database berisi semua data yang telah disusun dan diorganisasikan sedemikian rupa sehingga memudahkan pemberian informasi. (Wahana Komputer, 2010 : 24)

## II.7. Normalisasi

Untuk merealisasikan database pada tabel-tabel maka diperlukan sebuah tahapan yang disebut normalisasi. Normalisasi data adalah proses di mana tabel-tabel pada database dites dalam hal kesalingtergantungan di antara field-field pada sebuah tabel. Misalnya jika pada sebuah tabel terdapat ketergantungan terhadap lebih dari satu field dalam tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel.

Pada proses normalisasi data, aturan yang dijadikan acuan adalah metode ketergantungan fungsional. Teorinya adalah bahwa tiap kolom dalam sebuah tabel selalu memiliki hubungan yang unik dengan sebuah kolom kunci. Misalnya pada sebuah tabel data\_mahasiswa ada *field* nomor induk mahasiswa, data *field* nama mahasiswa serta *field* alamat. Maka ketergantungan fungsionalnya dapat dinyatakan sebagai berikut:  $nim \rightarrow nm\_mhs$  dan  $nmr\_induk \rightarrow alamat$ . Artinya  $nm\_mhs$  memiliki ketergantungan fungsional terhadap  $nim$ . *Field*  $nm\_mhs$  isinya juga ditentukan oleh *field*  $nim$ . Maksud dari semua itu adalah  $nim$  adalah *field* kunci yang menentukan karena tidak ada nomor induk yang sama pada satu universitas, jadi *field*  $nmr\_induk$  dapat dijadikan patokan untuk mengisi  $nm\_mhs$  dan *field* lainnya.

Ada beberapa langkah dalam normalisasi tabel, yaitu:

1. *Decomposition*, dekomposisi adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil jika tabel yang dikenal dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di –dekomposisi. Dekomposisi akan sering dilakukan dalam proses normalisasi untuk memenuhi syarat-syaratnya
2. Bentuk tidak normal, pada bentuk ini semua data yang ada pada tiap entity (diambil atributnya) masih ditampung dalam satu tabel besar. Data yang ada pada tabel ini masih ada yang redundansi dan ada juga yang kosong. Semuanya masih tidak tertata rapi.
3. Normal Form pertama (1<sup>st</sup> Normal Form), pada tahapan ini tabel di-dekomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel-tabel kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat atomik.
4. Normal Form kedua (2<sup>nd</sup> Normal Form), pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer tabel tersebut .
5. Normal Form ketiga (3<sup>rd</sup> Normal Form), setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum efisien. Biasanya penggunaan bentuk normal(normalisasi) hanya sampai pada bentuk ketiga, dan tabel yang dihasilkan telah memiliki kualitas untuk membentuk sebuah database yang dapat diandalkan. Semua tabel diatas juga telah memenuhi bentuk normal tahap ketiga. (Wahana Komputer, 2010 : 32)

## **II.8. Pemrograman Visual Basic 2010**

Visual Basic merupakan salah satu aplikasi pemrograman visual yang paling banyak digunakan dan cukup populer karena kemudahan penggunaannya. Selain itu, Visual Basic yang mengadopsi bahasa pemrograman basic juga memiliki segudang fitur dan fasilitas yang memudahkan para programmer mengembangkan sebuah aplikasi.

Visual Basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *microsoft* yaitu Microsoft Visual Studio 2010 sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang dikeluarkan oleh *microsoft*, seperti Visual Basic, Visual C++, Visual Web Developer, Visual C# dan Visual F#. Semua IDE pemrograman tersebut sudah mendukung penuh implementasi .Net Framework terbaru yaitu .Net Framework 4.0 yang merupakan pengembangan dari .Net Framework 3.5. Adapun database standar yang disertakan adalah Microsoft SQL Server 2008 Express. Microsoft Visual Studio 2010 menambahkan perbaikan – perbaikan fitur dan fitur baru yang lebih lengkap dibandingkan versi sebelumnya yaitu Microsoft Visual Studio 2008 (Wahana Komputer, 2010 : 2)

## **II.9. Database SQL Server 2008**

Microsoft SQL Server 2008 Express Edition merupakan edisi lite dari Microsoft SQL Server 2008. Fitur – fitur yang terdapat didalamnya bisa dikatakan hampir sama dengan versi sebelumnya. Hal yang membedakan hanyalah fungsi-fungsi enterprise dan ketiadaan database manager yang bersifat independen atau berupa aplikasi mandiri. Bagi sebagian user dan developer aplikasi memang agak

sedikit menyusahkan apabila hendak melakukan maintenance database secara langsung, tetapi bagi pengguna Visual Basic 2010 hal tersebut tidak menjadi masalah, sebab Visual Basic 2010 sudah ditanamkan fitur untuk mengelola database yang berasal dari Microsoft SQL Server 2008 Express Editio. (Wahana Komputer, 2010 : 152)

## **II.10.UML**

*Unified Modeling Language* (UML) adalah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembangan sistem yang membuat *blue print* atas visinya dalam bentuk yang baku. (Yuni Sugiarti, S.T. M.Kom, 2013 : 34)

### **II.10.1. Diagram-DiagramUML**

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram , yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interkasi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka- antarmuka, kolaborasi- kolaborasi serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula digram kelas memuat kelas-kelas aktif.
2. Diagram Paket (*Package Diagram*). Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.
3. Diagram *Use-Case*. Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram Interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. Diagram Komunikasi (*Communication Diagram*). Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram*). Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. Diagram Aktivitas (*Activity Diagram*). Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu

aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. Diagram Komponen (*Component Diagram*). Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. Diagram *Deployment* (*Deployment Diagram*). Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan. Pada UML dimungkinkan kita menggunakan diagram- diagram lainnya (misalnya *Data Flow Diagram*, *Entity Relationship Diagram* dan sebagainya). (Prabowo Pudjo Widodo, 2007 : 10)