

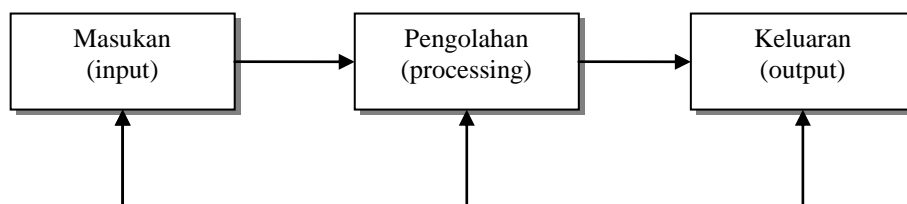
BAB II

TINJAUAN PUSTAKA

II.1. 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. Murdick dan Ross (1993) mendefinisikan sistem sebagai perangkat elemen yang digabungkan satu sama lainnya untuk suatu tujuan bersama.

Menurut *Scott (1996)*, sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*processing*), serta keluaran (*output*). Ciri pokok sistem menurut Gaspert ada empat, yaitu sistem itu beroperasi dalam suatu lingkungan, terdiri atas unsur-unsur, ditandai dengan saling berhubungan, dan mempunyai satu fungsi atau tujuan utama.



Gambar II.1. Model Sistem

(Sumber : Hanif Al Fatta2007:4)

Gambar di atas menunjukkan bahwa sistem atau pendekatan sistem minimal harus mempunyai empat komponen, yakni masukan, pengolahan, keluaran, dan balikan atau *control*.

Sementara *Mc. Leod (1995)* mendefinisikan sistem sebagai sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Sumber daya mengalir dari elemen output dan untuk menjamin prosesnya berjalan dengan baik maka dihubungkan dengan mekanisme *control*.

Banyak ahli mengajukan konsep sistem dengan deskripsi yang berbeda, tetapi pada prinsipnya hampir sama dengan konsep dasar sistem umumnya. *Schronderberg (1971)* dalam *Suradinata (1996)* secara ringkas menjelaskan bahwa sistem adalah :

1. Komponen-komponen yang saling berhubungan satu sama lain.
2. Suatu keseluruhan tanpa memisahkan komponen pembentuknya.
3. Bersama-sama dalam mencapai tujuan.
4. Memiliki input dan output yang dibutuhkan oleh sistem lainnya.
5. Terdapat proses yang mengubah input menjadi output.
6. Menunjukkan adanya entropi.
7. Memiliki aturan.
8. Memiliki subsistem yang lebih kecil.
9. Memiliki deferensi antar subsistem.
10. Memiliki tujuan yang sama meskipun mulainya berbeda. (Hanif Al Fattah, 2007 : 5)

II.1.1. 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 : 5-6)

II.2. 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.3. 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.4. Peminjaman

Pemberian kredit, dalam pengertian sebagai *cash loan*, merupakan salah bentuk usaha yang dapat dilakukan oleh sebuah bank. Berdasarkan UU nomor 10 Tahun 1998 tentang perubahan atas UU nomor 7 Tahun 1992 tentang perbankan, yang dimaksud dengan kredit adalah penyediaan uang atau tagihan yang dapat dipersamakan dengan itu, berdasarkan persetujuan atau kesepakatan pinjam-meminjam antara bank dengan pihak lain yang mewajibkan pihak peminjam untuk melunasi utangnya setelah jangka waktu tertentu dengan pemberian bunga.

Menurut Undang-undang tersebut, penyediaan dana untuk nasabahnya tidak hanya bisa dalam bentuk kredit. Penyediaan dana tersebut dapat juga berupa penyediaan pembiayaan berdasarkan prinsip syariah sesuai dengan ketentuan yang ditetapkan oleh Bank Indonesia, seperti tercantum dalam pasal 1 UU nomor 1998. penyaluran dana dalam bentuk kredit ini biasanya mendominasi sebagian besar pengalokasian dana bank.

II.5. Basis Data (*Database*)

Basis data adalah tempat penyimpanan file data. Sebagai file data, suatu basis data tidak menyajikan informasi secara langsung kepada pengguna. Pengguna harus menjalankan aplikasi untuk mengakses dari basis data dan menyajikan dalam bentuk yang bisa dimenegrti. Ketika bekerja dengan file-file data, suatu aplikasi harus dikodekan agar bekerja dengan struktur masing-masing

file data. biasanya, suatu basis data berisi suatu katalog yang menggunakan aplikasi untuk menentukan cara data diorganisir.

Basis data biasanya memiliki dua bagian utama. Pertama, file yang memegang basis data fisik. kedua, perangkat lunak sistem manajemen basis data (DBMS) menggunakan aplikasi untuk mengakses data. DBMS bertanggung jawab menguatkan struktur basis data termasuk :

- a. Memelihara hubungan antar data di dalam basis data.
- b. Memastikan bahwa data tersimpan secara tepat, dan menetapkan aturan hubungan data agar tidak terlanggar.\
- c. Pemulihan (*recovery*) semua data dari kegagalan sistem (Janner Simarmata; 2007 : 2)

II.6. Normalisasi

Redudansi data cenderung melebihi ukuran dari basis data dan itu menjadi sebuah masalah yang sangat serius dalam media basis data yang besar. Selain itu, redudansi data bisa mendorong ke arah yang tidak normal (*anomalies*). Untuk memahami permasalahan yang bisa mengakibatkan pemborosan (*redudansi*), kita harus mengetahui pengertian redudansi lebih dekat lagi.

Sebelumnya, mari kita mulai dengan mengamati bahwa atribut dari skema tabel bisa digolongkan ke dalam tiga kelompok :

1. Atribut yang digunakan untuk tujuan identifikasi.
2. Atribut yang digunakan untuk tujuan informasi
3. Atribut yang digunakan untuk tujuan keduanya, baik identifikasi maupun informasi.

a. Bentuk Tidak Normal.

untuk membuat perancangan basis data, Anda pasti sudah mengenal sejumlah bentuk khusus, sifat-sifat, atau batasan skema tabel yang dimiliki untuk mencapai suatu tujuan yang diinginkan, misalnya memperkecil redundansi. Bentuk itu disebut “Bentuk Normal”

ada enam bentuk normal yang dikenal yaitu :

1. Bentuk Normal Pertama (1NF)
2. Bentuk Normal Kedua (2NF)
3. Bentuk Normal Ketiga (3NF)
4. Bentuk Normal Boyce Codd (BCNF)
5. Bentuk Normal Keempat (4NF)
6. Bentuk Normal Kelima (5NF)

Bentuk normal pertama sampai ketiga (dibuat oleh E.F. Codd) merupakan bentuk normal yang umum dipakai. Maksudnya, pada kebanyakan relasi, persoalan anomaly tidak akan muncul lagi bila ketiga bentuk normal tersebut telah terpenuhi.

b. Bentuk Normal Pertama (1 NF)

Bentuk normal pertama sangat sederhana. skema tabel disebut dalam bentuk normal pertama jika nilai atribut tidak terpisahkan. Untuk mengilustrasikannya, semua penulis buku di dalam atribut tunggal disebut penulis.

Berikut sebuah contoh :

ISBN = 979 -763-120-6

Judul = Basis Data

Penulis = Janner Simarmata Dan Iman Prayudi

Penerbit = Andi Yogyakarta

Oleh karena itu skema dalam kasus ini mengizinkan lebih dari satu nama penulis sebagai atribut penulis, maka skema bukan dalam bentuk normal pertama. Tentu saja, salah satu dari permasalahan yang jelas nyata dengan atribut penerbit adalah ketidakmungkinan mengurutkan data menggunakan nama penulis individu. Akan lebih sulit juga, sebagai contoh, bila menyiapkan label surat untuk setiap penulis dan seterusnya.

c. Bentuk Normal Kedua (2 NF)

Berdasarkan skema tabel T memiliki bentuk normal kedua jika semua atribut informasi (atribut yang tidak memiliki kunci mana pun) adalah atribut dari entitas lain di dalam skema tabel dan bukan dari kelas entitas lainnya. Dengan kata lain, atribut informasi menyediakan informasi secara rinci tentang entitas di dalam kelas entitas itu dan bukan beberapa entitas lain.

d. Bentuk Normal Ketiga (3 NF)

Bentuk normal kedua adalah baik, tetapi kita bisa melakukan yang lebih baik lagi. Kita sudah melihat bahwa jika skema tabel berbentuk normal kedua, maka bukan informasinya yang kuat, atribut tergantung pada subset yang sesuai dengan kunci. Bagaimanapun juga, ada kemungkinan lain yang tidak diinginkan.

Lihat contoh berikut :

Asumsi dan skema tabel berikut adalah untuk tujuan ilustrasi, yakni bahwa tidak ada dua buku dengan judul yang sama dengan penerbit yang sama :

{Judul, KdPenerbit, JmlHalaman, Harga}

Satu-satunya kunci untuk skema tabel itu adalah **{Judul, KdPenerbit, JmlHalaman}**, dan **Harga** hanya lah atribut informasi.

e. Bentuk Normal Boyce Codd (BCNF)

BCNF merupakan bentuk normal sebagai perbaikan terhadap 3 NF. Suatu relasi BCNF selalu memenuhi 3NF, tetapi tidak seaiknya. Suatu relasi yang memenuhi 3 NF belum tentu memenuhi BCNF. Dalam banyak literature disebutkan bentuk normal ketiga pun mungkin masih mengandung anomali sehingga masih perlu dinormalisasikan selanjutnya (Janner Simarmata; 2007: 75-83).

II.7. *Unified Modeling Language (UML)*

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standart. (Chonoles; 2003 : 6) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudji Widodo, Herlawati; 2011 : 6-7).

II.7.1. Diagram-Diagram *UML*

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung

menjadi diagram interaksi. 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, serta relasi-relasi diagram. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas.
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 waktu tertentu.
5. Diagram Komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi *structural* 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 sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member 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 diagram kelas dimana komponen dipetakan kedalam 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 di dalamnya. 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 (Probowo Pudji Widodo, Herlawati; 2011 : 10-12).

II.8. Microsoft Visual Studio 2008

Microsoft Visual Studio 2008 merupakan kelanjutan dari *Microsoft Visual Studio* sebelumnya, yaitu *Visual Studio .Net 2003* yang diproduksi oleh Microsoft. Pada bulan Februari 2002 *Microsoft* memproduksi teknologi. *Net Framework* versi 1.0, teknologi. *Net* ini didasarkan atas susunan berupa *Net Framework*, sehingga setiap produk baru yang terkait dengan teknologi. *Net* akan selalu berkembang mengikuti perkembangan. *Net Frameworknya*. Pada perkembangannya nantinya mungkin untuk membuat program dengan teknologi. *Net* memungkinkan para pengembang perangkat lunak akan dapat menggunakan lintas sistem operasi, yaitu dapat dikembangkan di sistem operasi windows juga dapat dijalankan pada sistem operasi lain, misalkan pada sistem operasi *Linux*, seperti yang telah dilakukan pada pemrograman *Java* oleh *Sun Microsystems*. Pada saat ini perusahaan-perusahaan sudah banyak mengupdate aplikasi lama yang dibuat *Microsoft Visual Basic 6.0* ke teknologi. *Net* karena kelebihan-kelebihan yang ditawarkan, terutama memungkinkan pengembang perangkat lunak secara cepat mampu membuat program *robust*, serta berbasiskan integrasi ke internet yang dikenal dengan *XML Web Service* (Wahana Komputer ; 2008 : 1)

II.9. Microsoft SQL Server

SQL Server Management Studio membantu anda mengatur database dengan mudah. Anda dapat melakukan pengaturan atas beberapa pada sebuah komputer

saja atau melakukan pengaturan *server* secara *remote*. Anda dapat juga membuat *database*, *table*, *index*, dan melakukan manipulasi data terhadap *database* dan tabel-tabelnya.

SQL Server Management Studio memiliki beberapa komponen penting yang mewakili kegunaanya dalam perancangan *database*, dan melakukan pengaturan sistem secara keseluruhan. Komponen-komponen tersebut itu adalah :

- a. *Registered Server*
- b. *Object Explorer*
- c. *Query editor*

Adapun tampilan Microsoft SQL Server 2008 dapat dilihat pada gambar.16. sebagai berikut (Wahana Komputer; 2008 : 40)