

BAB II

TINJAUAN PUSTAKA

II.1. Konsep Sistem Informasi

Konsep dasar sistem informasi akan menguraikan konsep dasar sistem, informasi, dan sistem informasi serta pengembangan sistem informasi, yang mulai dari tahap investigasi, analisis, perancangan dan implementasi sistem.

II.1.1. Sistem

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu (Tata Sutabri; 2012: 10). Menurut (Hanif Al Fatta; 2007: 3), dalam bukunya yang berjudul “Analisa dan Perancangan Sistem Informasi” Sistem adalah kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama atau sekumpulan objek-objek yang saling berelasi dan berinteraksi serta berhubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan. Sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu (Anastasia Diana,dkk ; 2011: 3).

II.1.2. Karakteristik Sistem

Sebuah sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut :

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 layar komputer, 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 menyimpan 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 Fatta; 2007: 5-6).

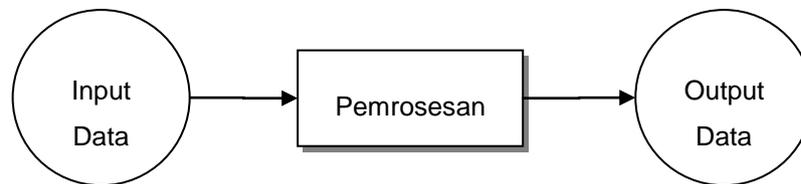
II.1.3. Pengertian Informasi

Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tidak berguna menjadi berguna bagi penerimanya (Tata Sutabri; 2012 :29).

Davis (1995) dalam buku (Hanif Al Fatta ; Analisa dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi *Modern* ; 2007 : 9) mengungkapkan bahwa informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat sekarang.

Untuk memahami pengertian sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi.

Dengan demikian, sistem informasi berdasarkan konsep (*input, processing, output* – *IPO*) dapat dilihat pada gambar berikut:



Gambar II.1. Konsep sistem informasi

(Sumber : Hanif Al Fattah, 2007 : 5)

II.1.4. Komponen Sistem Informasi

Stair (1992) dalam buku (Hanif Al Fatta ; Analisa dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi *Modern* ; 2007: 9-10), menjelaskan bahwa sistem informasi berbasis komputer (CBIS) dalam suatu organisasi terdiri dari komponen-komponen berikut :

1. Perangkat keras, yaitu perangkat keras komponen untuk melengkapi kegiatan memasukkan data, memproses data, dan keluaran data.
2. Perangkat lunak, yaitu program dan instruksi yang diberikan ke komputer.
3. *Database*, yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.
4. Telekomunikasi, yaitu komunikasi yang menghubungkan antara pengguna sistem dengan sistem komputer secara bersama-sama ke dalam suatu jaringan kerja yang efektif.

5. Manusia, yaitu personel dari sistem informasi, meliputi manajer, analis, programmer, dan operator, serta yang bertanggung jawab terhadap perawatan sistem.
6. Prosedur, yakni tata cara yang meliputi strategi, kebijakan, metode, dan peraturan-peraturan dalam menggunakan sistem informasi berbasis komputer.

Sementara *Burch dan Grudnitski (1986)* dalam buku (Hanif Al Fatta ; Analisa dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi *Modern* ; 2007: 10-11) berpendapat, sistem informasi yang terdiri dari komponen-komponen di atas disebut dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), dan blok kendali (*control block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

1. Blok Masukan. Input mewakili data yang masuk ke dalam sistem informasi. Input di sini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan yang dapat berupa dokumen-dokumen dasar.
2. Blok Model. Blok ini terdiri dari kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.
3. Blok Keluaran. Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen serta semua pemakai sistem.

4. Blok Teknologi. Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan sekaligus mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.
5. Blok Database. Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.
6. Blok Kendali. Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

II.1.5. Sistem Informasi Akuntansi

Akuntansi merupakan proses mengidentifikasi, mengukur, mencatat dan mengkomunikasikan peristiwa-peristiwa ekonomi dari suatu organisasi (bisnis maupun non bisnis) kepada pihak-pihak yang berkepentingan dengan informasi bisnis tersebut (pengguna informasi). Pada dasarnya fokus utama dari akuntansi adalah transaksi bisnis (Anastasia Diana,dkk ; 2011: 14).

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Adapun tujuan sistem informasi akuntansi adalah :

1. Mengamankan harta/kekayaan perusahaan. Harta/kekayaan di sini meliputi kas perusahaan, persediaan barang dagangan, termasuk aset tetap perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan.

3. Menghasilkan informasi untuk pihak eksternal.
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan).
6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan.
7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian (Anastasia Diana,dkk ; 2011: 5-7).

II.1.6. Tunggakan

Kredit yang terutang oleh nasabah harus dibayar tepat pada waktunya. Untuk memperingan nasabah maka pembayaran dapat diangsur tiap bulan berjalan. Oleh karena itu apabila setelah tanggal jatuh tempo kredit belum dilunasi maka timbul tunggakan kredit.

Dalam (Kamus Besar Bahasa Indonesia yang diterbitkan oleh Widya Karya – Semarang, Indonesia ; 2005 : 598) menyatakan, tunggakan adalah angsuran yang belum dibayar atau utang yang masih belum dilunasi setelah tanggal pengenaan denda. Maka dapat disimpulkan bahwa tunggakan kredit timbul apabila *customer* tidak membayar kreditnya saat tanggal jatuh tempo, telah ditegur dan ditagih.

II.1.7. Pengertian Over Kredit

Istilah kredit berasal dari bahasa Romawi yaitu *Credere* yang berarti kepercayaan. Sehingga dasar dari kredit adalah kepercayaan atau keyakinan dari seseorang, bahwa pihak lain pada masa yang akan datang sanggup memenuhi

segala sesuatu yang telah diperjanjikan. Sedangkan tujuan kredit pada umumnya adalah didasarkan kepada usaha untuk memperoleh keuntungan sesuai dengan prinsip ekonomi yaitu dengan pengorbanan yang sekecil-kecilnya untuk memperoleh keuntungan yang sebesar-besarnya. Keuntungan itu terjelma dalam bentuk bunga yang diterima oleh perusahaan selaku kreditur baik bunga kredit ataupun tunggakan sementara bagi konsumen.

Sementara menurut (Kamus Besar Bahasa Indonesia yang diterbitkan oleh Widya Karya – Semarang, Indonesia ; 2005 : 268) kredit adalah cara menjual barang dengan pembayaran secara tunai atau pinjaman uang dengan pembayaran pengembalian secara mengangsur.

Oleh karena itu dapat disimpulkan bahwa pengalihan hak/over kredit merupakan suatu kesepakatan. Kata sepakat dalam suatu perjanjian merupakan suatu keadaan yang menunjukkan kehendak kedua belah pihak, saling menerima satu dengan lainnya. Dengan adanya kata sepakat, maka perjanjian itu telah ada dan telah lahir dan sejak saat itu perjanjian mengikat kedua belah pihak dan dapat dilaksanakan.

II.2. Basisdata (*Database*)

Stephens dan Plew (2000) dalam buku “Basis Data” oleh (Janner Simarmata,dkk ; 2006 : 1), menyatakan basisdata adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basisdata, pengguna dapat menyimpan data secara terorganisasi. Cara data disimpan dalam basisdata

menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basisdata, dimodifikasi dan dihapus.

Sistem manajemen basisdata (DBMS) adalah kumpulan data yang saling berhubungan dan kumpulan program untuk mengakses data. Tujuan utama sistem manajemen basisdata adalah menyediakan cara menyimpan dan mengambil informasi basisdata secara mudah dan efisien.

II.2.1. Komponen Penyusun Basisdata

Komponen penyusun basisdata adalah :

1. Skema basisdata

Pada dasarnya, skema adalah sekelompok objek dalam basisdata yang saling berhubungan. Dalam skema, objek yang dihubungkan memiliki relasi satu sama lain. Ada satu pemilik skema yaitu yang memiliki akses memanipulasi struktur semua objek dalam skema.

2. Tabel

Tabel adalah unit penyimpanan fisik utama untuk data dalam basisdata. Ketika mengakses basisdata, pengguna mengacu tabel untuk data yang diinginkan. Basisdata bisa terdiri atas banyak tabel sehingga harus terdapat relasi antar tabel. Ada 4 macam tabel yaitu :

- a. Tabel data berfungsi menyimpan sebagian besar data yang ditemukan pada basisdata.
- b. Join tabel adalah tabel yang digunakan untuk membentuk relasi antara dua tabel.

- c. Subset tabel mengandung bagian data suatu tabel data.
- d. Tabel validasi, kadang disebut juga tabel kode, digunakan untuk membuat valid data yang dimasukkan pada tabel lainnya.

3. Kolom

Kolom atau field adalah kategori informasi yang terdapat dalam tabel. Kolom bagi tabel seperti atribut bagi entitas. Kolom adalah struktur penyimpanan logis dalam sebuah basisdata. Masing-masing kolom memiliki tipe data.

4. Baris

Satu baris data adalah kumpulan semua kolom yang berhubungan dengan kejadian tunggal. Dengan kata lain, satu baris data adalah record dalam tabel.

5. Tipe Data

Tipe data menentukan tipe data yang dapat disimpan dalam kolom tabel (Janner Simarmata,dkk ; 2010 : 36).

II.2.2. Tujuan Basisdata

Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, kecepatan dalam pengambilan kembali. Untuk mencapai tujuannya, syarat sebuah basis data adalah sebagai berikut :

- a. Tidak adanya redudansi dan inkosistensi data

Redudansi terjadi jika suatu informasi disimpan dibeberapa tempat. Akibat dari redudansi adalah inkosistensi data atau data yang tidak konsisten.

b. Kesulitan Pengaksesan Data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan *Query* ataupun dari tool untuk melihat tabelnya.

c. *Multiple User*

Basis data memungkinkan penggunaan data bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda (Kusrini; 2007: 2-5).

II.3. Kamus Data

Kamus data (KD) atau *data dictionary (DD)* atau disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, analisis sistem dapat mendefinisikan data yang mengalir disistem dengan lengkap. KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis, KD dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir disistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. KD digunakan untuk merancang input, laporan-laporan dan database. KD dibuat berdasarkan arus data yang ada di DAD (Jogiyanto; 2005: 725).

II.3.1. Isi Kamus Data

KD harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk maksud keperluan ini, maka KD harus memuat hal-hal berikut ini:

a. Nama arus data

Karena KD dibuat berdasarkan arus data yang mengalir di DAD, maka nama dari arus data juga harus dicatat di KD.

b. Alias

Alias atau nama lain dari data dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya.

c. Bentuk data

Bentuk dari data ini perlu dicatat di KD, karena dapat digunakan untuk mengelompokkan KD ke dalam kegunaannya sewaktu perancangan sistem. KD yang mencatat data yang mengalir dalam bentuk dokumen dasar atau formulir akan digunakan untuk merancang bentuk input sistem.

d. Arus data

Arus data menunjukkan dari mana data mengalir kemana data akan menuju. Keterangan arus data ini perlu dicatat di KD supaya memudahkan mencari arus data ini di DAD.

e. Penjelasan

Untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat di KD, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

f. Periode

Periode ini menunjukkan kapan terjadinya arus data ini. Periode perlu dicatat di KD karena dapat digunakan untuk mengidentifikasi kapan input data harus dimasukkan ke sistem, kapan proses dari program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

g. Volume

Volume yang perlu dicatat di KD adalah tentang volume rata-rata dan volume puncak dari arus data. Volume rata-rata menunjukkan banyaknya rata-rata arus data yang mengalir dalam satu periode tertentu dan volume puncak menunjukkan volume yang terbanyak.

h. Struktur data

Struktur data menunjukkan arus data yang dicatat di KD terdiri dari item-item data apa saja (jogiyanto; 2005: 726-728).

II.4. SQL Server 2008

SQL server 2008 adalah sebuah terobosan baru dari microsoft dalam bidang database. SQL server adalah sebuah BDMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusun pendahuluanya seperti IBM dan Oracle. SQL server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data (Wahana Komputer ; 2010 : 2).

II.5. Entity Relationship Diagram (ERD)

Entity Relationship Diagram adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu protek kedalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien (Janner Simarmata,dkk ; 2010 : 67).

Relasi adalah hubungan antara beberapa entitas. Sebagai contoh, relasi menghubungkan mahasiswa dengan mata kuliah yang diambilnya. Kumpulan semua entitas bertipe sama disebut kumpulan entitas (*entity set*), sedangkan kumpulan semua relasi bertipe sama disebut kumpulan relasi (*relationship*).

Struktur logis (skema database) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen berikut (Janner Simarmata,dkk ; 2010 : 60) :

Tabel II.1. Komponen-Komponen Diagram ERD

| | |
|---|---|
|  | Persegi panjang mewakili kumpulan entitas |
|  | Elips mewakili atribut |
|  | Belah ketupat mewakili relasi |
|  | Garis menghubungkan atribut dengan kumpulan entitas dengan relasi |

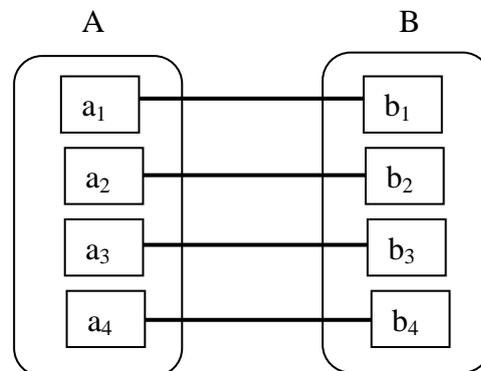
(Sumber : Janner Simarmata, dkk : 2006 : 60)

II.5.1. Pemetaan Kardinalitas

Pemetaan kardinalitas menyatakan jumlah entitas dimana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi. Pemetaan kardinalitas sangat berguna dalam menentukan himpunan relasi biner meskipun pemetaan dapat berperan dalam deskripsi himpunan relasi yang melibatkan lebih dari dua himpunan entitas.

Untuk suatu himpunan relasi biner R antara himpunan entitas A dan B , pemetaan kardinalitas harus salah satu dari berikut:

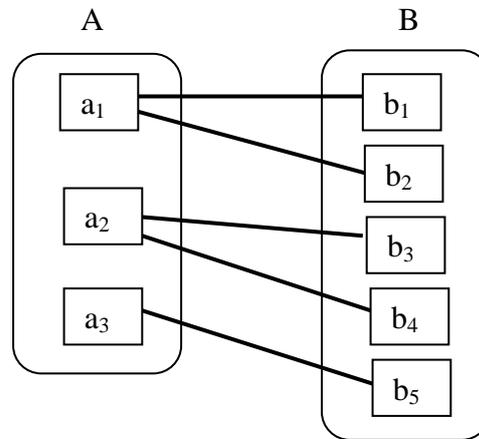
- a. One-to-One, sebuah entitas A berhubungan dengan paling banyak satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A .



Gambar II.2 Hubungan one-to-one

(Sumber : Janner Simarmata,dkk; 2006: 64)

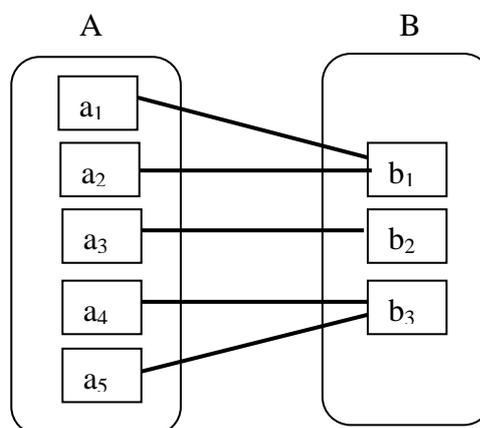
- b. One-to Many, sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B . Sebuah entitas pada B dapat dihubungkan dengan paling banyak satu entitas pada A .



Gambar II.3 Hubungan one-to-many

(Sumber : Janner Simarmata,dkk; 2006: 64)

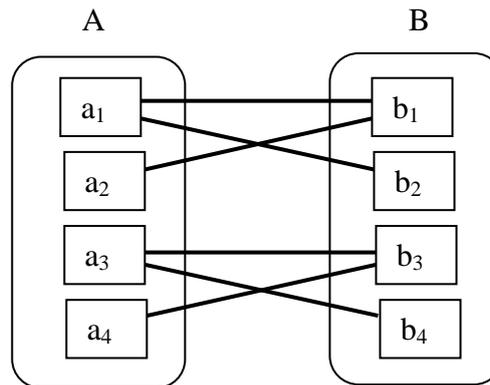
- c. Many-to-One, sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A.



Gambar II.4 Hubungan many-to-one

(Sumber : Janner Simarmata,dkk; 2006: 64)

- d. Many-to-Many, sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B dan sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A.



Gambar II.5 Hubungan many-to-many

(Sumber : Janner Simarmata,dkk; 2006: 64)

II.5.2. Normalisasi

Normalisasi adalah bagian perancangan basis data. Tanpa normalisasi, sistem basisdata menjadi tidak akurat, lambat, tidak efisien, serta tidak membreikan data yang diharapkan.

Pada waktu menormalisasi basisdata, ada empat tujuan yang harus dicapai yaitu:

1. Mengatur data dalam kelompok-kelompok sehingga masing-masing kelompok hanya menangani bagian kecil sistem.
2. Meminimalkan jumlah data berulang dalam basisdata.

3. Membuat basisdata yang datanya diakses dan dimanipulasi secara cepat dan efisien tanpa melupakan integritas data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda akan hanya mengubah pada satu tempat (Janner Simarmata,dkk; 2006: 77).

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(1st 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(2ndNormal 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(3rdNormal 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.6. UML (*Unified Modeling Language*)

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Chonoles (2003) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik (Prabowo Pudjo Widodo, 2011 : 6)

UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar; 2005: 17). UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Karena ini merupakan sebuah bahasa, UML mempunyai sejumlah aturan untuk menggabungkan/mengkombinasikan elemen-elemen tersebut (Munawar; 2005: 22).

Meskipun UML sudah banyak menyediakan diagram yang bisa membantu mendefinisikan sebuah aplikasi, tidak berarti bahwa semua diagram tersebut akan bisa menjawab persoalan yang ada. Dalam banyak kasus, diagram lain selain UML sangat banyak membantu (Munawar; 2005: 23).

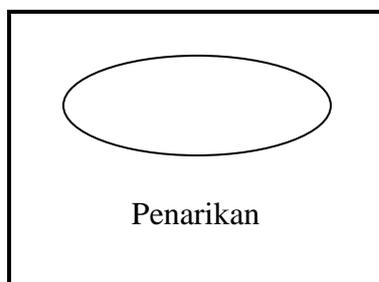
Tabel II.2. Tipe Diagram UML

| Diagram | Tujuan |
|----------------|---|
| Activity | Perilaku prosedural & paralel |
| Class | Class, fitur, dan relasinya |
| Communication | Interaksi diantara obyek. Lebih menekankan ke Link |
| Component | Struktur dan koneksi dari komponen |
| Deployment | Penyebaran / instalasi ke klien |
| Package | Struktur hierarki saat kompilasi |
| Sequence | Interaksi antar obyek. Lebih menekankan pada urutan |
| State Machine | Bagaimana event mengubah sebuah obyek |
| Use Case | Bagaimana user berinteraksi dengan sebuah sistem |

(Sumber : Munawar; 2005 : 23)

II.6.1. Use Case

Menurut (Pilone, 2005: bab7.1) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan (Whitten, 2004: 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomasisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval* (Prabowo Pudjo Widodo, 2011: 21-22).

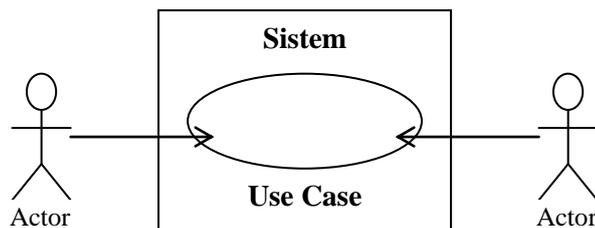


Gambar II.6 Simbol Use case

(Sumber : Prabowo Pudjo Widodo; 2011: 22)

Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan *actor*. *Actor* adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem. *Actor* adalah abstraction dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*.

Diagram use case menunjukkan 3 aspek dari sistem yaitu : *actor*, *use case* dan sistem/sub sistem *boundary*. *Actor* mewakili peran orang, sistem yang lain atau alat berkomunikasi dengan *use case* (Munawar; 2005: 63-65).

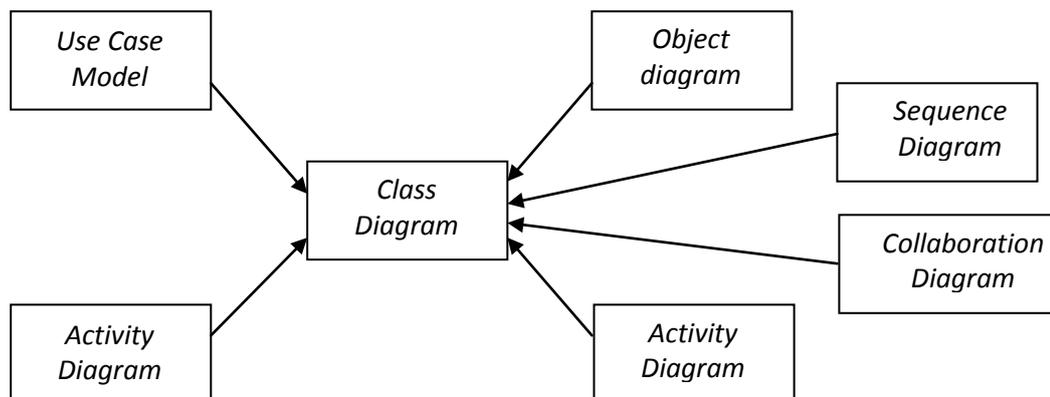


Gambar II.7 Use Case Model

Sumber : Pemodelan Visual Dengan UML (Munawar; 2005: 64)

II.6.2. Class Diagram

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini. Diagram kelas merupakan sekumpulan kelas-kelas objek. (Whitten, 2004: 410) mengartikan kelas sebagai satu set objek yang memiliki atribut dan perilaku yang sama. Secara teknis, (Pender, 2003: bab 5) mengartikan sebuah kelas sebagai suatu definisi sumber daya yang termasuk di dalamnya informasi-informasi yang menggambarkan fitur suatu entitas dan bagaimana penggunaannya. Sedangkan objek adalah entitas yang bersifat unik yang mengikuti aturan-aturan yang sudah didefinisikan dalam kelasnya (Prabowo Pudjo Widodo, dkk, 2011: 39).



Gambar II.8. Hubungan diagram kelas dengan diagram UML lainnya

(Sumber : (Prabowo Pudjo Widodo,dkk, 2011: 38)

Atribut adalah properti dari sebuah *class*. Atribut ini melukiskan batas nilai yang mungkin ada pada *obyek* dari *class*. Sebuah *class* mungkin mempunyai nol atau lebih atribut. Secara konvensi, jika nama atribut terdiri dari atas satu suku kata, maka ditulis dengan huruf kecil. Akan tetapi jika nama atribut mengandung lebih dari satu suku kata maka semua kata digabungkan dengan suku kata pertama menggunakan huruf kecil dan awal suku kata berikutnya menggunakan huruf besar (Munawar; 2005: 35).

II.6.3. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh *obyek* dan *message* (pesan) yang diletakkan diantara *obyek-obyek* ini di dalam *use case*.

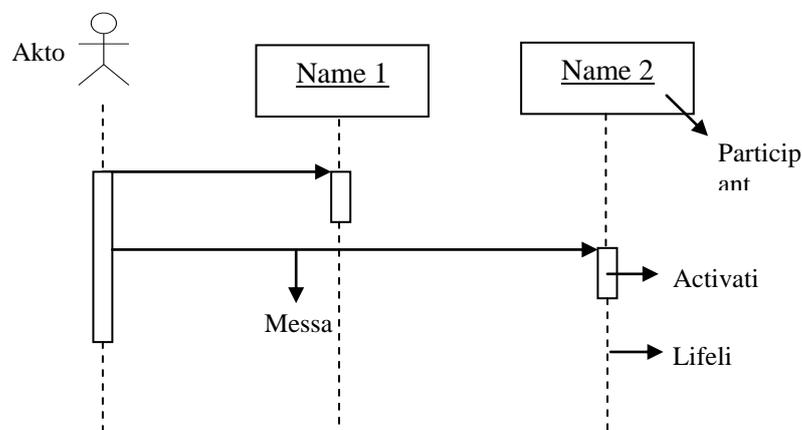
Komponen utama *sequence* diagram terdiri atas *obyek* yang dituliskan dengan kotak segiempat bernama, *message* diwakili oleh garis dengan tanda panah (Munawar; 2005: 87).



Gambar II.9. Participant pada sebuah *Sequence* diagram

(Sumber : Munawar; 2005: 88)

Gambar II.9 menunjukkan esensi simbol dari *sequence* diagram dan simbol kerjanya secara bersama-sama. *Participant* terletak disebelah atas. Setiap *lifeline* menggunakan garis putus-putus yang menurun dari *participant*. Garis yang solid dengan tanda panah menghubungkan antara satu *lifeline* dengan *lifeline* yang lain dan mewakili sebuah *message* dari satu *participant* ke *participant* yang lain. Dari gambar tersebut terlihat seorang *actor* menginisialisasi *sequence* diagram meskipun *actor* bukan bagian dari *sequence* diagram.



Gambar II.10. Simbol-simbol yang ada pada *Sequence* diagram

(Sumber : Munawar; 2005: 89)

II.6.4. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity* diagram mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity* diagram bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa (Munawar; 2005: 109).

Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan *activity* diagram.

Tabel II.3. Simbol-simbol yang sering dipakai pada *activity* diagram

| Simbol | Keterangan |
|---|---|
|  | Titik Awal |
|  | Titik Akhir |
|  | Activity |
|  | Pilihan untuk pengambilan keputusan |
|  | Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu |
|  | Rake; menunjukkan adanya dekomposisi |
|  | Tanda waktu |
|  | Tanda pengiriman |
|  | Tanda penerimaan |
|  | Aliran akhir (Flow Final) |

(Sumber : Munawar; 2005: 110)

II.7. Pemrograman Visual Basic 2008

Visual basic 2008 merupakan lingkungan pengembangan terintegrasi atau biasa disebut IDE yang dikembangkan berbasis bahasa dibuat dan dikembangkan oleh Microsoft Corporation dengan nama Microsoft Quick Basic. Kesederhanaan sintaks dan fleksibilitas bahasa basic yang menyebabkan bahasa pemrograman ini begitu fenomenal sehingga banyak disukai dan dipakai oleh programmer di seluruh dunia.

Berbekal kepopuleran tersebut, Microsoft mengembangkan bahasa Basic ini menjadi produk yang sangat terkenal di kalangan programmer, yaitu mulai Microsoft Visual Basic 6.0 sampai sekarang, yaitu Microsoft Visual Basic 2008. Perkembangan teknologi dan penambahan banyak sekali fitur pada Visual Basic 2008 tidak mengakibatkan adanya perubahan sintaks-sintaks dasar yang terdapat di dalamnya. Sehingga dapat dikatakan, untuk dapat menjadi programmer Visual Basic 2008 sebenarnya harus menguasai dan mengerti bagaimana menggunakan dan mengimplementasikan sintaks dasar dalam bahasa pemrograman Basic (Wahana Komputer ; 2010 : 36).