

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem adalah sekumpulan objek, ide, berikut saling keterkaitannya (inter-relasi) di dalam (usaha) mencapai suatu tujuan atau sasaran bersama tertentu atau dengan kata lain, sistem dapat disebutkan sebagai kumpulan komponen (sub-sistem fisik maupun non-fisik/logika) yang saling berhubungan satu sama lainnya dan bekerja sama secara harmonis untuk mencapai suatu tujuan. (Eddy Prahasta ; 2009 : 89).

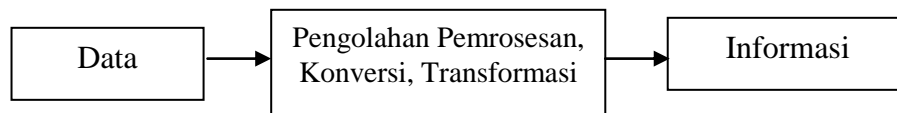
II.2. Data

Data merupakan bahasa *mathematical* atau simbol-simbol pengganti lain yang (telah) disepakati secara umum di dalam (usaha) menggambarkan suatu objek, manusia, peristiwa, aktivitas, konsep, atau objek-objek penting lainnya. (Eddy Prahasta ; 2009 : 78).

II.3. Informasi

Informasi adalah data yang telah ditempatkan pada konteks yang penuh arti oleh penerimanya. Informasi masih menurut sumber yang sama adalah makna atau pengertian yang dapat diambil dari suatu data dengan menggunakan konvensi-konvensi yang telah umum digunakan di dalam representasinya. Hubungan data dengan informasi adalah Sangat erat sekali. Data diibaratkan

sebagai bahan baku yang mengalami proses transformasi, sehingga keluarannya berubah menjadi barang jadi. (Eddy Prahasta ; 2009 : 78).



Gambar II.1. Relasi Antara Data dan Informasi
Sumber : Eddy Prahasta (2009 ; 79)

II.4. Sistem Informasi

Sistem informasi merupakan sebuah entitas (kesatuan) formal yang terdiri dari berbagai sumber daya fisik maupun logika. Dari organisasi, sumber daya ini disusun atau distrukturkan dengan beberapa cara (yang bisa jadi berlainan satu sama lainnya), karena suatu organisasi dan sistem informasi terkait merupakan sumber daya yang bersifat dinamis. (Eddy Prahasta ; 2009 : 93).

Sistem informasi dapat terdiri dari komponen-komponen yang disebut dengan istilah Blok rancangan yang terdiri dari :

1. Input

Blok ini memberikan ilustrasi mengenai berbagai data yang menjadi masukan sistem informasi. Pada umumnya, data yang diperlukan sebagai masukan sistem diturunkan dari kebutuhan informasi tertentu (khusus).

2. Pemrosesan

Blok ini sering dirujuk sebagai operasi data, bersama dengan blok *input*, nilai-nilai khusus blok pemrosesan juga diturunkan dari kebutuhan

informasi tertentu.

3. Basis Data

Defenisi yang paling umum untuk basis data adalah tempat penyimpanan (repository) data yang diperlukan. Basis data sangat memiliki arti bagi pengguna sistem informasi.

4. Pengendalian

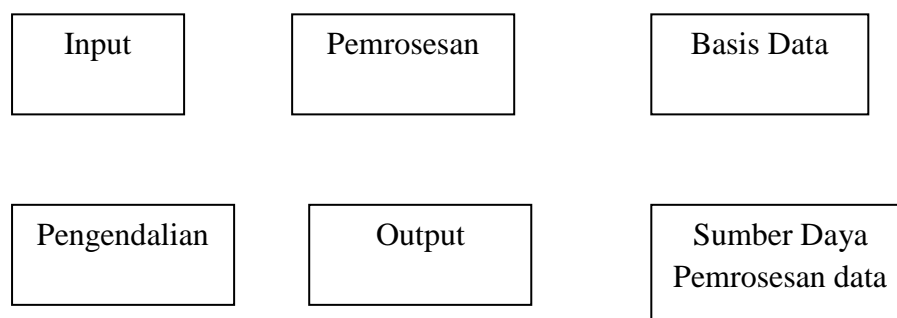
Sistem informasi merupakan suatu sumber daya yang bersifat kompleks dan dinamis. Seperti hasilnya pada sumber daya yang lain, perlu suatu cara untuk memastikan bahwa sistem informasi tersebut dapat beroperasi sebagaimana rancangannya.

5. Output

Blok ini mengacu pada bentuk dan isi informasi aktual yang diberikan kepada pengguna system informasi.

6. Sumber Daya Pemrosesan Data

Implementasi fisik dari blok rancangan ini dinyatakan oleh sekumpulan data, perangkat keras, perangkat lunak dan manusia. (Eddy Prahasta; 2009 : 95-96).



Gambar II.2. Komponen Sistem Informasi
Sumber : Eddy Prahasta (2009 : 95)

II.5. SIA (Sistem Informasi Akuntansi)

II.5.1. SIA

Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai tujuan. Sistem selalu terdiri dari beberapa subsistem kecil, yang masing-masing melakukan fungsi khusus yang penting untuk dan mendukung bagi sistem yang lebih besar, tempat mereka berada.

SIA terdiri dari beberapa komponen :

1. *Orang-orang* yang mengoperasikan system tersebut dan melaksanakan sebagai fungsi.
2. *Prosedur-Prosedur*, baik manual maupun terotomatisasi, yang dilibatkan dalam mengumpulkan memproses, dan menyimpan data dalam aktifitas-aktifitas organisasi.
3. *Data* tentang proses-proses bisnis organisasi.
4. *Software* yang dipakai untuk memproses data organisasi.
5. *Infrastruktur teknologi informasi*, termasuk computer, peralatan pendukung (*peripheral device*), dan peralatan untuk komunikasi jaringan.

Kelima komponen ini secara bersama-sama memungkinkan suatu SIA memenuhi tiga fungsinya dalam organisasi, yaitu :

1. Mengumpulkan dan menyimpan data tentang aktifitas-aktifitas yang dilaksanakan oleh organisasi, sumberdaya yang dipengaruhi oleh aktifitas-aktifitas tersebut, dan para pelaku yang terlibat dalam

aktifitas tersebut, agar pihak manajemen, para pegawai, dan pihak-pihak luar yang berkepentingan dapat meninjau ulang (*review*) hal-hal yang telah terjadi.

2. Mengubah data menjadi informasi yang berguna bagi pihak manajemen untuk membuat keputusan dalam aktifitas perencanaan, pelaksanaan, dan pengawasan.
3. Menyediakan pengendalian yang memadai untuk menjaga aset-aset organisasi, termasuk data organisasi, untuk memastikan bahwa data tersebut tersedia saat dibutuhkan, akurat, dan andal.

(Marshal B. Romney Paul John Steinbart; 2006:2-3)

James A.Hall (2009 : 10) menjelaskan, subsistem SIA memproses berbagai transaksi keuangan dan transaksi non keuangan yang secara langsung mempengaruhi pemrosesan transaksi keuangan.

II.5.2. Akuntansi

menjelaskan ada beberapa kutipan pendapat dari ahli mengenai definisi akuntansi, yaitu sebagai berikut : Menurut **ABP Statement** No.4 (Tahun 1970) yang berjudul “*Basic Concepts and Accounting principles Underlying Financial Statement of Business Enterprises*” : “Akuntansi adalah sebuah aktivitas jasa, dimana fungsinya adalah memberikan informasi kuantitatif, terutama informasi mengenai keuangan dan entitas ekonomi yang dimaksudkan akan menjadi berguna dalam pengambilan keputusan ekonomi (dalam membuat pilihan di antara berbagai alternative yang ada)”. Sedangkan menurut *American Institute of*

Certified Public Accountants (AICPA), akuntansi adalah : “seni pencatatan, pengklasifikasian, dan pengikhtisaran transaksi dan peristiwa keuangan dengan cara tertentu dan dalam ukuran moneter, termasuk penafsiran atas hasil-hasilnya”.

(Sindi Bulandari; Sistem informasi akuntansi penjualan dan pembelian (studi kasus : dealer yamaha CV. Lapang jaya motor; 2011: 2) Siklus akuntansi diawali dengan menganalisa dan menjurnal transaksi kemudian diakhiri dengan neraca saldo setelah penutupan yang terpenting dari siklus akuntansi adalah dihasilkannya laporan keuangan. Langkah-langkah yang harus dilakukan dalam siklus akuntansi antara lain:

1. Menganalisa transaksi bisnis
2. Menjurnal transaksi-transaksi bisnis .
3. Memposting transaksi-transaksi tersebut.
4. Menyiapkan neraca saldo
5. Menjurnal dan memposting data-data penyesuaian
6. Menyiapkan neraca saldo disesuaikan
7. Menyiapkan laporan keuangan
8. Menjurnal dan memposting jurnal penutup.
9. Menyiapkan neraca saldo setelah penutupan

II.5.2.1 Jurnal

Jurnal umum (*General Journal*) digunakan untuk mencatat semua transaksi yang tidak dicatat pada jurnal khusus, sedangkan jurnal khusus (*Special Journal*) digunakan untuk mencatat transaksi tertentu yang sering terjadi. Jurnal

khusus diterapkan pada perusahaan dagang terdiri dari jurnal pembelian, jurnal penjualan, jurnal pengeluaran kas, dan jurnal penerimaan kas. Jurnal penerimaan kas digunakan untuk mencatat semua penerimaan uang, termasuk penjualan tunai dan penerimaan tagihan.

II.5.2.2 Buku Besar (*General Ledger*)

Buku besar merupakan suatu penggolongan dari transaksi-transaksi menurut kelompok rekening akun (*posting*) atau dapat diartikan setiap jurnal yang sudah dibuat dipindahkan ke dalam buku besar (*general ledger*) sesuai dengan kelompok rekeningnya (Suharli, 2006: 134). Contoh : Kas akan masuk ke buku besar khusus rekening kas yang berada di sisi debet maupun kredit, sedangkan penjualan akan masuk ke buku besar penjualan yang berada di sisi kredit. (Sistem Informasi Akuntansi Penjualan Pada Pt.Afixkogyo Indonesia ;Arfeny Oktantia Mariena Heri, Fitri Sukmawati,S.E.M.M, Hanura Ian Pratowo, ST ; 2011: 4-5)

II.6. CV. Sumber Tani

II.6.1. Sejarah Tentang CV. Sumber Tani

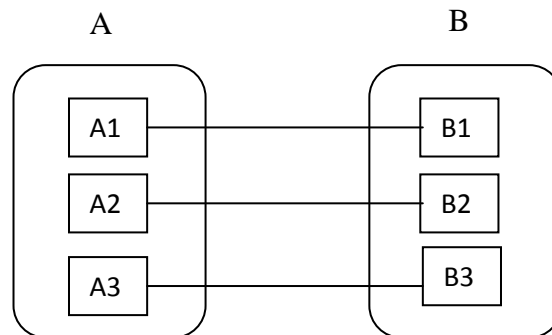
CV. Sumber Tani berdiri sejak tahun 1993 di Desa Pematang Sentang kecamatan Tanjung Pura kabupaten Langkat, yang sekarang usianya sudah mencapai 20 tahun. CV. Sumber Tani didirikan oleh satu orang, yaitu Bapak Musdi yang menjabat sebagai Pemilik CV. Sekarang. CV. Sumber Tani telah memiliki banyak rekan untuk bekerja sama contohnya seperti agen-agen kecil yang ada di dalam maupun luar desa.

CV. Sumber Tani dapat menghasilkan 70 ton perharinya, dan hasil ini di dapat dari penghasilan kebun masyarakat dan perkebunan yang ada di Desa pematang sentang maupun di luar desa karena CV. Sumber Tani juga bekerja sama dengan masyarakat yang ada di luar Desa Pematang sentang guna memperoleh Tandan Buah Segar yang lebih baik untuk di pasarkan. (Sumber : CV. Sumber Tani)

II.7. Basis Data

Basis data menurut Stephens dan Plew (2000), 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. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basisdata menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basisdata, dimodifikasi, dan dihapus.

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



Gambar II.3. Hubungan one-to-one

Sumber : Janner Simarmata & Iman Prayudi (2006,2010 ; 64)

Contoh:

Aturan bisnis :

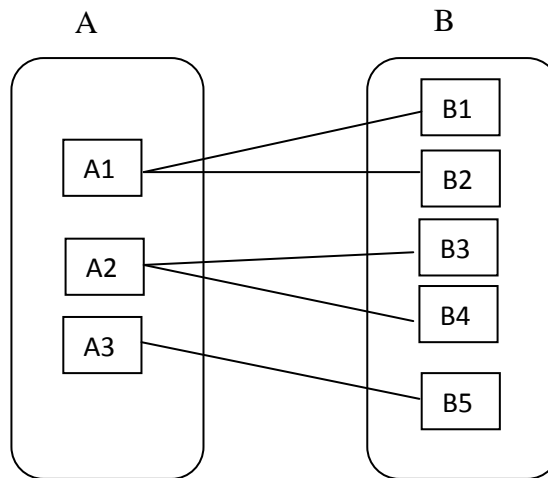
- a. Pada pengajar privat, satu guru satu siswa.
- b. Seorang guru mengajar seorang siswa, seorang siswa diajar oleh seorang guru.



Gambar II.4. Contoh Hubungan one-to-one

Sumber : Janner Simarmata & Iman Prayudi (2006,2010 ; 64)

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



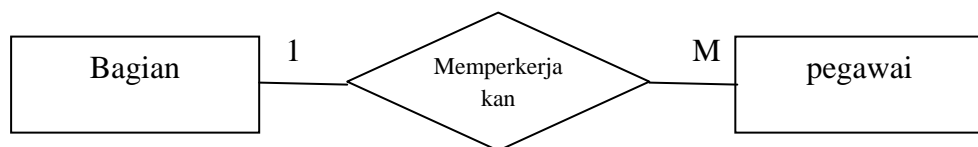
Gambar II.5. Hubungan one-to-Many

Sumber : Janner Simarmata & Iman Prayudi (2006,2010 ; 65)

Contoh:

Aturan bisnis :

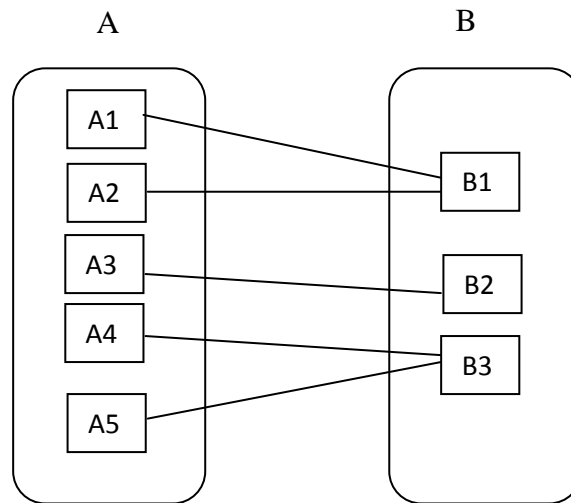
- a. Dalam satu perusahaan, satu bagian mempekerjakan banyak pegawai.
- b. Satu bagian mempekerjakan banyak pegawai, satu pegawai kerja dalam satu bagian.



Gambar II.6. Contoh Hubungan one-to-Many

Sumber : Janner Simarmata & Iman Prayudi (2006,2010 ; 65)

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



Gambar II.7. Hubungan Many-to-one

Sumber : Janner Simarmata & Iman Prayudi (2006,2010 ; 65)

Contoh:

Aturan bisnis :

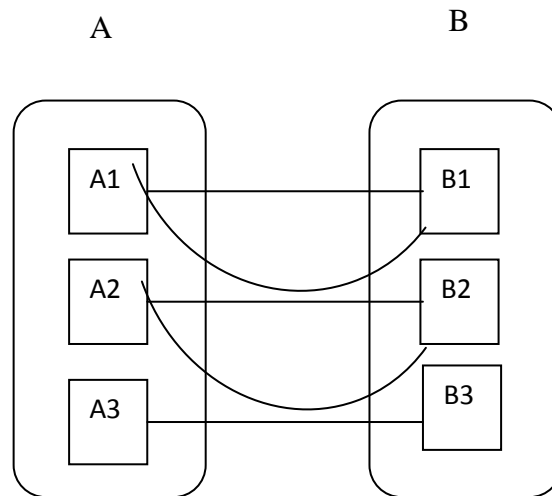
- a. Dalam suatu perusahaan, banyak pegawai dipekerjakan pada satu bagian.
- b. Banyak pegawai dipekerjakan pada satu bagian, satu pegawai bekerja dalam satu bagian.



Gambar II.8. Contoh Hubungan Many-to-one

Sumber : Janner Simarmata & Iman Prayudi (2006,2010 ; 65)

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



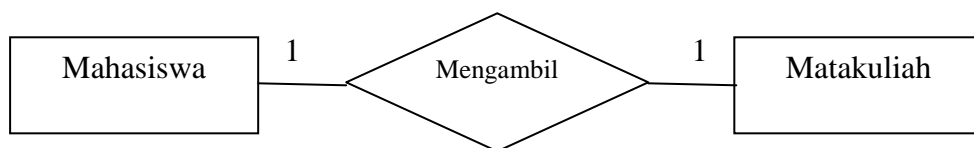
Gambar II.9. Hubungan Many-to-Many

Sumber : Janner Simarmata & Iman Prayudi (2006,2010 ; 66)

Contoh:

Aturan bisnis:

- Dalam universitas, seorang mahasiswa dapat banyak mengambil mata kuliah.
- Satu mahasiswa banyak mengambil mata kuliah dan satu mata kuliah diambil banyak mahasiswa. (Janner Simarmata & Iman Prayudi (2006,2010 ; 64-66)



Gambar II.10. Contoh Hubungan Many-to-Many

Sumber : Janner Simarmata & Iman Prayudi (2006,2010 ; 66)

II.7.1. Komponen Database

Sebuah database dapat terdiri dari berbagai macam komponen, yaitu:

- Tabel-tabel
- Index
- Stored Prosedur
- Function
- Informasi tentang Scurity

Sebenarnya masih ada komponen lainnya akan tetapi semua dapatdi ambil dari praktik, seperti membuat sebuah form aplikasi windows. Komponen yang paling dasar dalam database adalah tabel. Tabel terdiri dari baris, kolom, kolom mewakili data yang dalam kolom yang sama memiliki kesamaan (bukan berarti isinya sama). Misalnya, kolom nama pada database siswa, mulai dari nama, kelas, alamat, Nama Orangtua, dan lain sebagainya. baris bisa disebut ricord dan kolom bisa disebut Field. (Wahana komputer ; 2009 : 244)

II.8. *Diagram Entity Relationship (ERD)*

Entity Relationship Diagram adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek kedalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien (www.infocom.cqu.edu.au)

1. Entitas (Entity)

Entitas adalah sesuatu yang nyata atau abstrak dimana kita akan menyimpan data. Ada 4 kelas entitas, yaitu misalnya pegawai, pembayaran, kampus dan buku.

2. Relasi (Relationship)

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai.

3. Atribut (Attribute)

Atribut adalah ciri umum semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah property, elemen data, dan field. Misalnya nama, alamat, nomor pegawai, dan gaji adalah atribut entitas pegawai. (Janner Simarmata & Iman Prayudi ; 2006,2010 : 67)

II.9. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata relational. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional (www.uteksas.edu).

Teori normalisasi didasarkan pada kosep bentuk normal. Sebuah tabel relasional dikatakan berada pada benetuk normal tertentu jika tabel memenuhi himpunan batas tertentu. Ada lima bentuk normal yang telah ditemukan.

a. Bentuk normal pertama (1NF)

Contoh yang kita gunakan disini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri. Masing-masing pemasok bisa menyediakan banyak barang. Tabel relasionalnya dapat dituliskan sebagai berikut :

PEMASOK(p#, setatus, kota, b#, qyt) dimana

P# : kode pemasik (kunci utama)

Status : kode status kota

B# : barang yang dipasok

Qyt : jumlah barang yang dipasok

Tabel II.6 contoh pemasokdalam 1NF

PEMASOK

P#	Status	kota	B#	Qyt
P1	20	yogyakarta	B1	300
P2	20	yogyakarta	B2	200
P3	20	yogyakarta	B3	400
P4	30	yogyakarta	B4	200
P5	20	medan	B5	100
P6	10	medan	B6	100
P7	20	medan	B7	300
P8	20	yogyakarta	B8	400
P9	30	medan	B9	200

Sebagai contoh, anomaly berikut dapat terjadi pada tabel pemasok :

1. **INSERT** : Fakta bahwa pemasok tertentu (p5) berlokasi pada kota tertentu (Bandung) tidak dapat ditambahkan hingga mereka memasok barang.
2. **DELETE** : Jika sebuah baris dihapus maka yang hilang tidak hanya informasi tentang barang dan jumlahnya, tetapi juga informasi tentang pemasok.

3. **UPDATE** : Jika pemasok p1 pindah dari Yogyakarta ke Jakarta, maka enam baris harus di-update karena adanya perubahan.

b. Bentuk normal kedua (2NF)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal ke dua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama. Contoh PEMASOK2, yaitu :

1. **INSERT** : Fakta bahawa kota tertentu mempunyai status tertentu (Semarang mempunyai status 40) tidak dapat memasukkan hingga ada pemasok di kota tertentu.
2. **DELETE** : Menghapus sembarang baris pada PEMASOK2 akan menghilangkan informasi status tentang kota tersebut serta hubungan antara pemasok dan kota.

c. Bentuk normal ketiga (3NF)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika table sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

d. Bentuk normal keempat (4NF)

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) di dasarkan pada konsep ketergantungan multivalued (MVD). Sebuah ketergantungan multivalued terjadi ketika dalam sebuah tabel relasional yang mengandung setidaknya tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda.

e. Bentuk Normal kelima (5NF)

Sebuah tabel berada pada bentuk normal kelima (5NF) jika dia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*Join dependence*). Ketergantungan gabungan berarti bahwa sebuah tabel, setelah di dekomposisi menjadi tiga atau lebih kecil, harus dapat digabungkan kembali untuk membentuk tabel asal. Dengan kata lain, 5NF menunjukkan ketika sebuah tabel tidak dapat didekomposisi lagi.

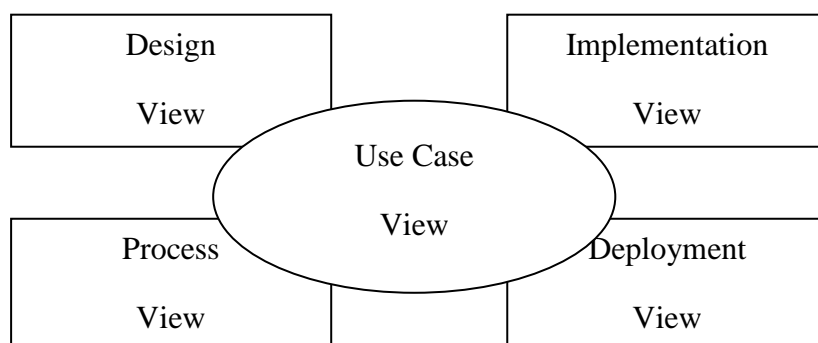
(Janner Simarmata & Iman Prayudi ; 2006,2010 : 77-86)

II.10. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah alat bantu yang sangat handal di dunia pengembangan system yang berorientasi objek. Hal ini disebabkan

karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang system 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.

UML dibangun atas model 4+1 view, model ini didasarkan pada fakta bahwa struktur sebuah system dideskripsikan dalam 5 view dimana salah satu di antaranya use case view, Use case view ini memegang peran khusus untuk mengintegrasikan content ke view yang lain. (Munawar ; 2005 : 17,20)



Gambar II.11. Model 4 + 1 view
Sumber : Munawar (2005 ; 20)




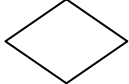



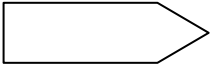


1. Activity Diagram

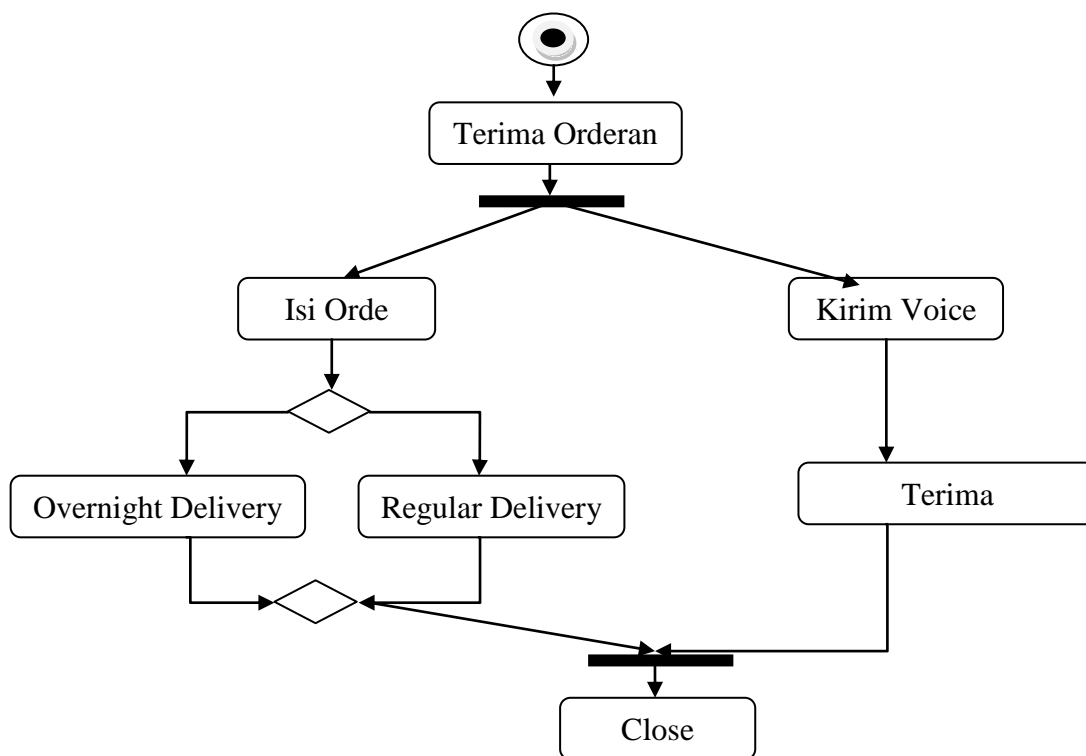
Activity diagram adalah teknik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity Diagram mempunyai peran seperti halnya Flowchart, akan tetapi perbedaannya dengan flowchart

adalah activity diagram bisa mendukung perilaku parallel sedangkan flowchart tidak bisa. (Munawar; 2005: 109)

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

Tabel II.2 Simbol-simbol yang sering dipakai pada activity diagram

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork; digunakan untuk menunjukan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Rake; menunjukan adanya dekomposisi
	Tanda Waktu
	Tanda Pengiriman
	Tanda Penerimaan
	Aliran akhir (Flow Final)



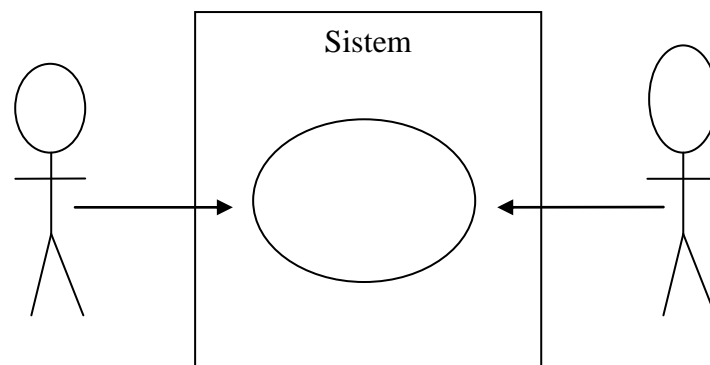
Gambar II.12. Contoh activity diagram sederhana
 Sumber : Munawar (2005 ; 111)

2. Class Diagram

Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu system. Hal ini disebabkan karena class adalah deskripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Disamping itu class diagram bisa memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class-class yang ada dan relasinya satu dengan lainnya. Itu sebabnya class diagram menjadi diagram paling populer di UML. (Munawar; 2005: 219)

3. Use Case Diagram

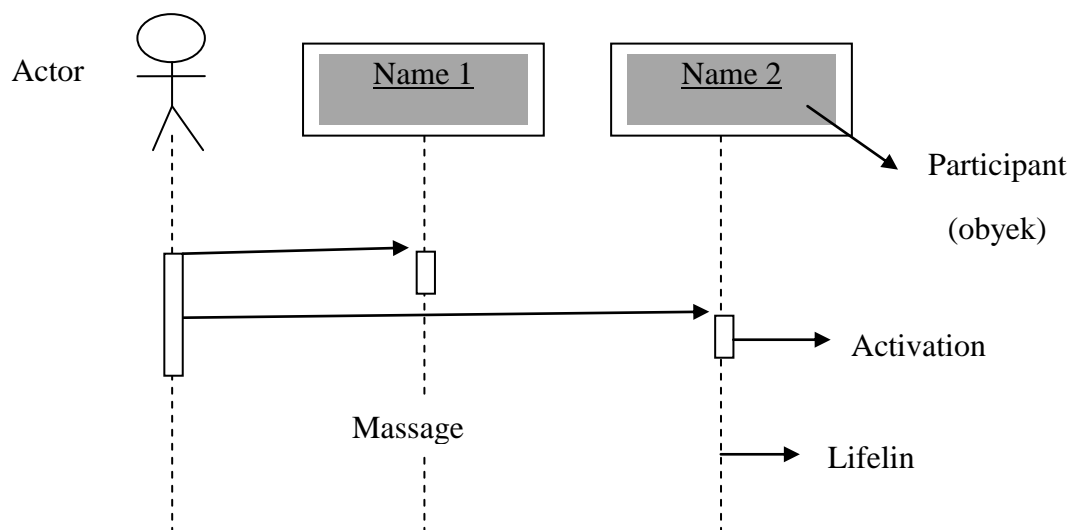
Use case adalah deskripsi fungsi dari sebuah system dari perspektif pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan system di sebut **scenario**. Setiap scenario mendeskripsikan urutan kejadian. Setiap urutan di inisialisasi oleh orang, system yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan use case adalah serangkaian scenario yang digabungkan bersama-sama oleh tujuan umum pengguna. (Munawar; 2005: 63)



Gambar II.13. Use Case Model
Sumber : Munawar (2005 ; 64)

4. Sequence Diagram

Sequence diagram digunakan untuk perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam use case. (Munawar; 2005: 87)



Gambar II.14. Simbol-simbol yang ada pada swquence diagram
Sumber : Munawar (2005 ; 89)

II.11. Visual Basic

Visual Basic adalah bahasa pemrograman generasi ketiga dari Microsoft dengan IDE (Integrated Development Environment) atau pemrograman pengembangan terpadu, Visual Basic di buat untuk dirancang untuk mudah digunakan baik oleh programmer pemula sekalipun.

Selain digunakan untuk program sederhana Visual Basic digunakan juga untuk membangun program kompleks, dengan menggunakan bahasa Basic, serta memungkinkan pengembangan program dengan cepat dengan antar muka grafis (GUI). (Eko Hari Atmoko;2013;1)

Visual basic adalah bahasa pemrograman visual yang sudah melegenda karena beberapa kemudahannya dalam pengembangan aplikasi Windows. Hal ini di tunjang dari kepopuleran bahasa BASIC yang merupakan pendahulu Visual

Basic. Ketika bahasa BASIC merambah dunia pemrograman visual mengikuti langkah sukses Windows yang menggantikan Dos- maka dari itu namanya berubah menjadi Visual Basic hal ini terjadi karena para pengembang dapat mengembangkan aplikasi secara visual, yaitu dengan menata dan memberi kode pada beberapa control yang digunakan.

Visual basic adalah bahasa pemrograman tingkat tinggi yang sudah sangat dikenal, dimulai dengan BASIC yang terdapat pada computer "angkatan tua", seperti AT286. Pada saat itu, bahasa basic merupakan sebuah bahasa yang sangat diandalkan dalam pembuatan beberapa aplikasi penting. BASIC digemari karena susunan programnya yang membebaskan kita untuk "melompat" dari satu baris program ke baris lainnya, versi BASIC lainnya adalah BASICA, Qbasic, Turbo Basic, dan lain-lain. Bahasa Basic banyak terdapat di masa penggunaan system operasi DOS. (Wahana Komputer ; 2009 : iii-2)