

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Definisi sistem berkembang sesuai dengan konteks dimana sistem itu digunakan. Berikut akan diberikan beberapa definisi sistem secara umum :

1. Kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan tertentu. Contoh :
 - a. Sistem Tata Surya
 - b. Sistem Pencernaan
 - c. Sistem Transportasi Umum
 - d. Sistem Otomotif
 - e. Sistem Komputer
 - f. Sistem Informasi
2. Sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai suatu tujuan.

Dengan demikian, secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain (Hanif Al Fatta ; 2009 : 4).

Sistem memiliki beberapa karakteristik, yaitu 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 Sistem (*Input*)

Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh sebuah sistem.

4. Keluaran Sistem (*Output*)

Sumber daya produk (informasi, laporan, dokumen, tampilan layer 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 sebuah subsistem dari sebuah sistem.

6. Penghubung (*Interface*)

Tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.

7. Penyimpanan (*storage*)

Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, bahan baku, energi dan sebagainya. (Hanif Al Fatta : 2009 : 5 – 6).

II.2. Sistem Pendukung Keputusan

Sejarah evolusi sistem pendukung keputusan dimulai pada tahun 1965, yang dibutuhkan oleh industri untuk menyimpan data dan menggabungkan ide, orang, sistem dan teknologi. Pada masa itu dimulai pembangunan *mainframe* IBM system 360 untuk mendukung terciptanya *Management Information System* (MIS) yang menitikberatkan pada fasilitas kepada manajer dalam bentuk laporan yang terstruktur dan periodik seperti laporan keuangan dan laporan transaksi.

Sebuah penelitian tentang implementasi sistem pendukung keputusan *model-driven decision support system* dipublikasikan dalam jurnal bisnis pada tahun 1970-an. Penelitian ini menjadi pionir bagaimana komputer dan model dalam sistem pendukung keputusan terus dikembangkan pada tahun 1980-an dan diikuti oleh perkembangan *knowledge-oriented Decision Support System* (DSS). Awal 1990-an, sistem pendukung keputusan dibangun menggunakan teknologi basis data relasional. Sejak dikenalkannya teknologi web sistem pendukung keputusan ini berkembang menjadi *Web-based DSS*. (Magdalena Karismariyanti ; 2011 : 55).

II.2.1. Ciri-ciri Sistem Pendukung Keputusan (SPK)

Menurut Kosasi dan Kusri (2007), adapun ciri-ciri sebuah SPK seperti yang dirumuskan oleh Alters Keen adalah berikut :

1. SPK ditunjukkan untuk membantu pengambilan keputusan-keputusan yang terstruktur dan umumnya dihadapi oleh para manajer berada di tingkat puncak.
2. SPK merupakan gabungan antara kumpulan model kualitatif dan kumpulan data.
3. SPK memiliki fasilitas interaktif yang dapat mempermudah hubungan antara manusia dengan computer.
4. SPK bersifat luwes dan dapat menyesuaikan dengan perubahan-perubahan yang terjadi.

II.2.2. Karakteristik, Kemampuan dan Keterbatasan SPK

Sehubungan banyaknya definisi yang dikemukakan mengenai pengertian dan penerapan dari sebuah SPK, sehingga menyebabkan terdapat banyak sekali pandangan mengenai sistem tersebut. Selanjutnya Turban (1996), menjelaskan terdapat sejumlah karakteristik dan kemampuan SPK yaitu :

1. Karakteristik SPK
 - a. Mendukung seluruh kegiatan organisasi.
 - b. Mendukung beberapa keputusan yang saling berinteraksi.
 - c. Dapat digunakan berulang kali dan bersifat konstan.
 - d. Terdapat dua komponen utama, yaitu data dan model.

- e. Menggunakan baik data eksternal dan internal.
- f. Memiliki kemampuan *what-if analysis* dan *goal seeking analysis*.
- g. Menggunakan beberapa model kuantitatif.

2. Kemampuan SPK

- a. Menunjang pembuatan keputusan manajemen dalam menangani masalah semi terstruktur dan tidak terstruktur.
- b. Membantu manajer pada bagian tingkatan manajemen, mulai dari manajemen tingkat atas sampai manajemen tingkat bawah.
- c. Menunjang pembuatan keputusan secara kelompok maupun perorangan.
- d. Menunjang pembuatan keputusan yang saling bergantung dan berurutan.
- e. Menunjang tahap-tahap pembuatan keputusan antara lain *intelligensi*, *desain*, *choise*, dan *implemention*.
- f. Kemampuan untuk melakukan adaptasi setiap saat dan bersifat fleksibel.
- g. Meningkatkan efektivitas dalam pembuatan keputusan dari pada efisiensi.
- h. Kemampuan pemodelan dan analisis pembuatan keputusan.
- i. Kemudahan melakukan pengaksesan berbagai sumber dan format data.

3. Keterbatasan SPK

- a. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan sebenarnya.
- b. Kemampuan suatu SPK terbatas pada pembendaharaan yang dimiliki (pengetahuan dasar serta model dasar).

- c. Proses-proses yang tidak dapat dilakukan oleh SPK biasanya tergantung juga pada kemampuan perangkat lunak yang digunakan.

SPK tidak memiliki kemampuan intuisi seperti yang dimiliki oleh manusia. Karena walau bagaimanapun canggihnya suatu SPK, hanyalah suatu kumpulan perangkat keras, perangkat lunak dan sistem operasi yang tidak dilengkapi dengan kemampuan berpikir. (Desi Leha kurnasih; 2013: 7)

II.3. Metode TOPSIS

TOPSIS adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981). TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlah dari sejumlah nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut.

TOPSIS memperimbangan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunan prioritas alternatif bisa dicapai. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan. Hal ini disebabkan konsepnya sederhana, mudah dipahami, komputasinya efisien, dan memiliki kemampuan mengukur kinerja relatif dan alternatif-alternatif keputusan

II.3.1. Langkah-langkah Metode TOPSIS

1. Membangun *normalized decision matrix*

Elemen r_{ij} hasil dari normalisasi *decision matrix* R dengan metode *Euclidean length of a vector* adalah :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_i^m x_{ij}^2}}$$

Dimana :

r_{ij} = Hasil dari normalisasi matriks keputusan R

$i = 1, 2, 3, \dots, m;$

$j = 1, 2, 3, \dots, n;$

2. Membangun *weighted normalized decision matrix* dengan bobot $W = (w_1, w_2, \dots, w_n)$, maka normalisasi bobot matriks V adalah :

$$V = \begin{bmatrix} w_{11}r'_{11} & \cdots & w_{1n}r'_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1}r'_{m1} & \cdots & w_{mn}r'_{mn} \end{bmatrix}$$

3. Menentukan solusi ideal positif dan solusi ideal negatif

Solusi ideal positif dinotasikan dengan A^+ dan solusi ideal negatif dinotasikan dengan A^- , sebagai berikut :

Menentukan solusi ideal (+) dan (-)

$$A^+ = \left\{ \left(\max_{j \in J} v_{ij} \right) \left(\min_{j \in J} v_{ij} \right) \mid i = 1, 2, 3, \dots, m \right\} = \{v_1^+, v_2^+, \dots, v_m^+\}$$

$$A^- = \left\{ \left(\max_{j \in J} v_{ij} \right) \left(\min_{j \in J} v_{ij} \right) \mid i = 1, 2, 3, \dots, m \right\} = \{v_1^-, v_2^-, \dots, v_m^-\}$$

Dimana :

V_{ij} = elemen mantriiks V baris ke- i dan kolom ke- j

$J = \{1, 2, 3, \dots, n \text{ dan } j \text{ berhubungan dengan } \textit{benefit criteria}\}$

$J = \{1, 2, 3, \dots, n \text{ dan } j \text{ berhubungan dengan } \textit{cost criteria}\}$

4. Menghitung Separasi

Separation measure ini merupakan pengukuran jarak dari suatu alternatif ke solusi ideal positif dan solusi ideal negatif perhitungan matematisnya sebagai berikut :

$$S_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2}, \text{ dengan } i = 1, 2, 3, \dots, m$$

Dimana :

$J = \{j=1, 2, 3, \dots, n \text{ j merupakan } \textit{benefit criteria}\}$

$J' = \{j=1, 2, 3, \dots, n \text{ dan } j \text{ merupakan } \textit{cost criteria}\}$

Separation measure untuk solusi ideal negatif

$$S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}, \text{ dengan } i = 1, 2, 3, \dots, n$$

Dimana :

$J = \{j=1, 2, 3, \dots, n \text{ dan } j \text{ merupakan } \textit{benefit criteria}\}$

$J' = \{j=1, 2, 3, \dots, n \text{ dan } j \text{ merupakan } \textit{cost criteria}\}$

5. Menghitung kedekatan relatif terhadap solusi ideal

Kedekatan relatif dari alternatif A^+ dengan solusi ideal A^- direpresentasikan dengan :

$$C_i = \frac{S_i^-}{S_i^- + S_i^+}, \text{ dengan } 0 < C_i < 1 \text{ dan } i = 1, 2, 3, \dots, m$$

6. Meranking alternatif

Alternatif dapat dirangki berdasarkan urutan C_i^* . Maka dari itu, alternatif terbaik adalah salah satu dan jarak terpendek terhadap solusi ideal dan berjarak terjauh dengan solusi ideal negatif. (Desi Leha Kurniasih; 2013: 8)

II.4. Entity Relationship Diagram (ERD)

Pada dasarnya ERD (*Entity Relationship Diagram*) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD di atas. ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik.

Entity dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen – komponen data (atribut atau field) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia database entity memiliki atribut yang menjelaskan karakteristik dari entity tersebut. Ada dua macam atribut yang dikenal dalam entity yaitu atribut yang berperan sebagai kunci primer dan atribut deskriptif. Hal ini berarti setiap entity

memiliki himpunan yang diperlukan sebuah primary key untuk membedakan anggota – anggota dalam himpunan tersebut.

Atribut dapat memiliki sifat – sifat sebagai berikut :

1. Atomic, atomik adalah sifat dari atribut yang menggambarkan bahwa atribut tersebut berisi nilai yang spesifik dan tidak dapat dipecah lagi. Contoh dari sifat atomik adalah field status dari tabel karyawan yang hanya berisi menikah atau single.
2. multivalued, sifat ini menandakan atribut ini bisa memiliki lebih dari satu nilai untuk tiap entity tertentu. Misalnya adalah field hobi, hobi dari tiap karyawan mungkin dan hampir pasti lebih dari satu. Misalnya karyawan A memiliki hobi : membaca, nonton TV dan bersepeda.
3. Composite, atribut yang bersifat komposit adalah atribut yang nilainya adalah gabungan dari beberapa atribut yang bersifat atomik. Contohnya adalah atribut alamat yang dapat dipecah menjadi atribut atomik berupa alamat, kode pos, no telepon, dan kota, (Wahana Komputer ; 2010 : 30).

II.5. Normalisasi

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional*.

Dengan normalisasi kita ingin mendesain *database relasional* yang terdiri dari tabel-tabel berikut :

1. Berisi data yang diperlukan.
2. Memiliki sesedikit mungkin redundansi.

3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.
4. Mengefisienkan update.
5. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal.

II.5.1. Bentuk-bentuk Normalisasi

1. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

No_Pasien
Nama_pasien
No_Pendaftaran
Tanggal_daftar
Alamat
Kode_Pos
Telpn
Tanggal_Periksa 1...n
Kode_Dokter 1...n
Nama_Dokter 1...n
Kode_Sakit 1...n
Deskripsi_Sakit 1...n
Kode_Obat 1...n
Nama_Obat 1... n
Dosis 1...n

Gambar II.1. Tabel Bentuk Tidak Normal
(Sumber : E. Utami dan S. Raharjo ; 2006 : 34)

2. Bentuk normal tahap pertama (1st Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

No_ Pendaftaran	Tgl_daftar	Tgl_ Periksa	No_Pasien
D/01/0001	15/07/2005	19/07/2005	PS/0001
D/01/0001	15/07/2005	19/07/2005	PS/0001
D/01/0002	18/08/2005	18/08/2005	PS/0002
D/01/0004	29/09/2005	01/10/2005	PS/0004

Nama	Alamat	Telpon
DENY	JL.TANJUNG 11	0274882255
DENY	JL.TANJUNG 11	0274882255
DESY	JL.PURDADI	081512345678
VITA	JL.ANGGREK	-

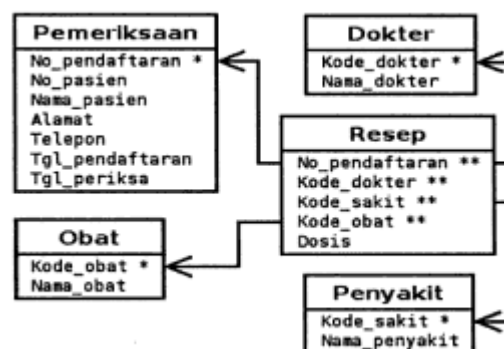
Kode_ Dokter	Nama_ Dokter	Kode_Sakit	Deskripsi_Sakit
DR/0001	Dr.Naufal	PD/KPL/001	Sakit kepala setengah
DR/0001	Dr.Naufal	PD/KPL/001	Sakit kepala setengah
DR/0003	Dr.Najwa	PL/KLT/003	Cacar air
DR/0001	Dr.Naufal	PD/JTN/007	Jantung koroner

Kode_Obat	Nama_Obat	Dosis
OD/KPL/001	Migrain Min	2 x 1

Gambar II.2. Tabel Bentuk Normal Pertama
(Sumber : E. Utami dan S. Raharjo ; 2006 : 35)

3. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

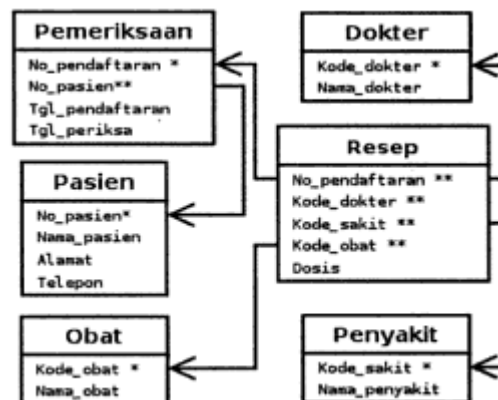


Gambar II.3. Tabel Bentuk Normal Kedua
(Sumber : E. Utami dan S. Raharjo ; 2006 : 38)

4. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

1. X haruslah superkey pada tabel tersebut.
2. Atau A merupakan bagian dari primary key pada tabel tersebut.



Gambar II.4. Tabel Bentuk Normal Ketiga
 (Sumber : E. Utami dan S. Raharjo ; 2006 : 40)

5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

6. Boyce Code Normal Form (BCNF)

1. Memenuhi 1st NF
2. Relasi harus bergantung fungsi pada atribut superkey (Kusrini, M.Kom ; 2007 : 39-43).

II.6. Pengertian Database

Banyak sekali definisi tentang database yang diberikan oleh para pakar dibidang ini. Database terdiri dari dua penggalan kata yaitu data dan base, yang artinya berbasiskan pada data. Tetapi secara konseptual, database diartikan sebuah koleksi atau kumpulan data yang saling berhubungan (relation), disusun menurut aturan tertentu secara logis, sehingga menghasilkan informasi. Sebuah informasi yang berdiri sendiri tidaklah dikatakan database.

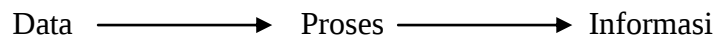
Contoh : Nomor telepon seorang pelanggan, disimpan dalam banyak tempat apakah itu di file pelanggan, di file alamat dan di lokasi yang lain. Antara file yang satu dengan file yang lainnya tidak saling berhubungan, sehingga apabila salah seorang pelanggan berganti nomor telepon dan anda hanya mengganti di file pelanggan saja, akibatnya akan terjadi ketidakcocokan data, karena di lokasi yang lain masih tersimpan data telepon yang lama.

Dalam sistem database hal ini tidak boleh dan tidak bisa terjadi, karena antara file yang satu dengan file yang lain saling berhubungan. Jika suatu data yang sama anda ubah, data tersebut di file yang lain akan otomatis berubah juga. Sehingga mampu menjadi informasi yang diinginkan dan dapat dilakukan proses pengambilan, penghapusan, pengeditan, terhadap data secara mudah dan cepat (Efektif, Efisien dan Akurat).

Data adalah fakta, baik berupa sebuah objek, orang dan lain – lain yang dapat dinyatakan dengan suatu nilai tertentu (angka, simbol, karakter tertentu, dan lain – lain). Sedangkan informasi adalah data yang telah diolah sehingga bernilai

guna dan dapat dijadikan bahan dalam pengambilan keputusan, (Yuhefizard ; 2008 : 2).

Hubungan data dan informasi dapat digambarkan sebagai berikut :



Gambar II.5. Data dan Informasi

Sumber : (Yuhefizard ; 2008 : 2)

II.7. Menenal Visual Basic

Visual Basic dibuat oleh *microsoft*, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain menawarkan kemudahan, Visual Basic juga cukup andal untuk digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi database. Visual basic merupakan bahasa pemrograman event drive, dimana program akan menunggu sampai ada respons dari user/pemakai program aplikasi yang dapat berupa kejadian atau event, misalnya ketika user mengklik tombol atau menekan enter. Jika kita membuat aplikasi dengan visual basic maka kita akan mendapatkan file yang menyusun aplikasi tersebut, yaitu :

1. File Project (*.vbp)

File ini merupakan kumpulan dari aplikasi yang kita buat. File project bisa berupa file *.frm, *.dsr atau file lainnya.

2. File Form (*.frm)

File ini merupakan file yang berfungsi untuk menyimpan informasi tentang bentuk form maupun interface yang kita buat, (Edy Winarto ; 2010 : 1).

II.8. SQL Server

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang di *develop* oleh *Microsoft*, yang digunakan untuk menyimpan dan mengolah data. Pada SQL Server 2008, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada SQL Server 2008, kita bisa membuat *object – object* yang sering digunakan pada aplikasi bisnis, seperti membuat *database, table, fuction, stored procedure, trigger* dan *view*. Selain *object*, kita juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data. (Cybertron Solution ; 2010 : 101).

II.9. Unified Modeling Language (UML)

UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industry perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat

dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut. (Windu Gata ; 2013 : 4).

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. (Windu Gata ; 2013 : 6).

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada usecase dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. (Windu Gata ; 2013 : 7).

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut – atribut dan operasi – operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. (Windu Gata ; 2013 : 8).