

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem merupakan sekelompok elemen – elemen yang terintegrasi dengan tujuan yang sama untuk mencapai suatu tujuan (Taufik hidayat, Volume 1, 2012).

Prof. Dr. Mr. S. Prju Atmosudirjo di dalam bukunya, ”Pengambilan Keputusan “, mendefinisikan : Sistem adalah setiap sesuatu yang terdiri dari obyek-obyek, atau komponen – komponen yang berkaitan, tertata dan saling berhubungan satu sama lain sedekimian rupa sehingga unsur-unsur tersebut menjadi satu kesatuan dari pemrosesan atau pengolahan data tertentu (Muh. Nur Luthfi Azis, Sukadi, 2012).

Menurut para ahli diatas, dapat disimpulkan sistem adalah mengandung arti kumpulan, unsur atau komponen yang saling berhubungan satu sama lain secara teratur dan merupakan satu kesatuan yang saling ketergantungan untuk mencapai suatu tujuan.

II.1.1. Karakteristik Sistem

Ada beberapa karakteristik yang membentuk sebuah system yaitu:

1. Komponen Sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen – komponen

tersebut dapat berupa suatu bentuk subsistem atau bagian – bagian dari sistem.

2. Batasan Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan, karena dengan batas system ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu system menunjukkan ruang lingkup (*scope*) dari system tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu diluar batas sistem yang memengaruhi operasi dari suatu sistem. Lingkungan luar system ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan atau dikendalikan agar tidak mengganggu operasi sistem.

4. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber – sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain, *output* dari suatu subsistem akan menjadi *input* dari subsistem yang lainnya.

5. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa perawatan (*maintenance input*) adalah energi yang dimasukkan supaya system tersebut dapat beroperasi. Masukan sinyal (*signal input*) adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh system, meliputi *output* yang berguna.

7. Pengolah Sistem (*Process*)

Merupakan bagian yang memproses masukan menjadi keluaran yang diinginkan.

8. Tujuan Sistem (*Goal*)

Setiap system pasti mempunyai tujuan ataupun sasaran yang memengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

II.1.2. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut pandangannya antara lain :

1. Sistem Abstrak (*Abstract System*)

Sistem yang berupa pemikiran atau ide – ide yang tidak tampak secara fisik.

2. Sistem Fisik (*Physical System*)

Merupakan system yang ada secara fisik sehingga setiap makhluk dapat melihatnya.

3. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam, dalam artian tidak dibuat, seperti system tata surya, system galaxy, system reproduksi, dan lain – lain.

4. Sistem Buatan Manusia (*Human Made System*)

Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *Human Machine System*.

5. Sistem Tertentu (*Deterministic System*)

Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian – bagiannya dapat dideteksi dengan pasti sehingga keluaran dari system dapat diramalkan.

6. Sistem Tak Tentu (*Probabilistic System*)

Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

7. Sistem Tertutup (*Closed System*)

Sistem yang tidak berhubungan dan tidak terpengaruh dengan system luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teori, system tersebut ada, tetapi kenyataannya tidak ada system yang benar – benar tertutup, yang ada hanyalah *relatively closed system* (secara relative tertutup, tidak benar – benar tertutup).

8. Sistem Terbuka (*Open System*)

Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan system terotomasi, yang merupakan bagian dari system buatan manusia dan berinteraksi dengan control oleh satu atau lebih computer sebagai bagian dari system yang digunakan dalam masyarakat modern.

II.1.3. Sistem Informasi Manajemen

Sistem informasi manajemen merupakan penerapan sistem informasi di dalam organisasi untuk mendukung informasi – informasi yang dibutuhkan oleh semua tingkatan manajemen.

Sistem informasi manajemen terbagi dalam sejumlah katagori, yaitu :

1. Permulaan Sistem

Pada fase ini terdapat pendefinisian dari permasalahan yang ada untuk menentukan ruang lingkup, menentukan metodologi yang digunakan, serta membuat jadwal kegiatan dengan menggunakan

beberapa teknik pengumpulan data seperti wawancara, observasi, dan dokumentasi.

2. Analisis Sistem

Pada fase ini, dilakukan analisis pada sistem yang telah ada dengan mengidentifikasi permasalahan, penentuan tujuan dari perbaikan sebuah sistem, dan mengidentifikasi sebuah kebutuhan pengguna sistem.

3. Desain Sistem.

Fase ini menyatakan bagaimana sebuah desain sistem lanjutan yang akan dibuat dengan menggambarkan sebuah model sistem untuk menyelesaikan permasalahan yang dihadapi perusahaan. Pada fase ini dilakukan pembuatan sistem baru (hardware dan software) dengan alat bantu yang digunakan antara lain *Microsoft Visual Studio 2008*, *SQL Server 2008*.

4. Implementasi Sistem

Pada fase ini, dilakukan penginstalan *database* dan program baru, pelatihan bagi pengguna dan konversi ke sistem yang baru.

5. Pemeliharaan Sistem

Setelah sistem dioperasikan maka dibutuhkan dukungan sistem yang berkesinambungan agar sisa siklus hidup sistem tetap berguna, produktif dan optimal (Wenny Widia, Iskandar zulkarnain, 2015).

II.1.4. Metode Sistem

1. *Blackbox Approach*

Suatu sistem dimana input dan output-nya dapat didefinisikan tetapi prosesnya tidak dapat diketahui atau tidak terdefinisi. Metode ini hanya dapat dimengerti oleh pihak dalam (yang menangani) sehingga pihak luar hanya mengetahui masukan dan hasilnya.

Contoh : bagian percetak uang, proses pencernaan.

2. Analytic System

Suatu metode yang mencoba untuk meliha hubungan seluruh masalah yang menyelidiki kesistematian tujuan dari sistem yang tidak efektif dan evaluasi pilihan dalam bentuk ketidak efektifan dan biaya. Dalam metode ini, beberapa langkah diberikan seperti dibawah ini :

1. Menentukan identitas dari sistem.
 - a. Sistem apa yang diterapkan.
 - b. Batasanya.
 - c. Apa yang dilaksanakan sistem tersebut.
2. Menentukan tujuan dari sistem.
 - a. Output yang dihasilkan dari isi sistem.
 - b. Fungsi dan tujuan yang diminta untuk mencoba menanggulangi lingkungan.
3. Bagian apa saja yang terdapat dalam sistem dan apa tujuan dari masing – masing bagian tersebut.

- a. Tujuan masing – masing bagian sistem harus jelas.
 - b. Cara apa yang digunakan subsistem untuk berhubungan dengan subsistem lainnya.
4. Bagaimana bagian – bagian yang ada dalam sistem itu saling berhubungan menjadi satu kesatuan.

II.1.5. Informasi

Definisi menurut Husain dan Wibowo Informasi merupakan seperangkat komponen yang saling berhubungan yang berfungsi mengumpulkan, memproses, menyimpan dan mendistribusikan informasi untuk mendukung pembuatan keputusan dan pengawasan dalam organisasi (Rizal Mahmud, Vol 9 No.3 September 2013).

Definisi menurut *Gordon B. Davis*: Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi penerima dan mempunyai nilai yang nyata atau yang dirasakan dalam keputusan-keputusan sekarang dan yang akan datang (Muh. Nur Luthfi Azis, Sukadi, 2012).

Menurut para ahli diatas, dapat di simpulkan informasi adalah produk dari pengolahan data Informasi. Informasi memiliki arti yang lebih penting dan lebih berarti sebagai bahan baku dari informasi yang berasal dari gambaran kejadian yang berwujud karakter, angka atau simbol tertentu yang memiliki arti. Informasi bisa terdiri dari data, gambar, naskah, dokumen dan suara yang berhubungan.

II.1.6. Sistem Informasi

“Suatu sistem terintegrasi yang mampu menyediakan informasi yang bermanfaat bagi penggunanya.”

“Sebuah system terintegrasi atau system manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dalam suatu organisasi.”

“Sekumpulan prosedur manual atau terkomputerisasi yang mengumpulkan/mengambil, mengolah, menyimpan, dan menyebarkan informasi dalam mendukung pengambilan dan kendali keputusan.”

“Sekelompok orang, prosedur, *input*, *output*, dan pengolahannya secara bersama – sama menghasilkan informasi yang akurat, tepat waktu, dan relevan bagi penggunanya”

Menurut Witarto, sistem informasi adalah sistem yang berisi SPD (sistem pengolah data) yang dilengkapi dengan kanal-kanal komunikasi yang digunakan dalam sistem organisasi data. Elemen proses dari sistem informasi antara lain mengumpulkan data. Mengolah data yang tersimpan, menyebarkan informasi (Taufik Hidayat, Vol 1, 2012)

Menurut para ahli diatas, dapat di simpulkan sistem informasi adalah sistem yang berisi jaringan sistem pengolahan data, yang dilengkapi dengan kanal-kanal komunikasi yang digunakan dalam sistem organisasi data. Elemen proses dari sistem informasi antara lain mengumpulkan data yang tersimpan, menyebarkan informasi.

II.1.7. Data

Data menurut *Drs.Jhon J Longkutoy* (Pengalaman komputer) mengatakan data adalah suatu istilah mejemuk dari fakta yang mengandung arti yang dihubungkan dengan kenyataan, simbol, gambar, angka, huruf yang menunjukkan suatu ide, objek, kondisi atau situasi (Muh.Nur Luthfi Azis, Sukardi Vol 1, 2012).

Data adalah representasi dari suatu fakta, yang dimodifikasi dalam bentuk gambar, kata, dan/atau angka. Manfaat data adalah sebagai satuan representasi yang dapat diingat, direkam, dan dapat diolah menjadi informasi. Karakteristiknya, data bukanlah fakta, namun representasi dari fakta. Kata sederhananya, data adalah catatan tentang fakta, atau data merupakan rekaman catatan tentang fakta. Data yang baik, adalah yang sesuai dengan faktanya (Taufik Hidayat, 2012).

Menurut para ahli diatas, dapat di simpulkan Data adalah bahan baku informasi, didefinisikan sebagai kumpulan simbol-simbol, tindakan, benda dan sebagainya.

II.2. Sistem Pendukung Keputusan

Pada dasarnya sistem pendukung keputusan merupakan pengembangan lebih lanjut dari sistem informasi manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Sifat interaktif dimaksudkan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan seperti prosedur, kebijakan,

teknik analisis, serta pengalaman suatu kerangka keputusan bersifat fleksibel (Desi Leha Kurniasih, 2013).

Sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Asep Hendar Rustiawan, 2012).

Menurut para ahli diatas, dapat di simpulkan Sistem pendukung keputusan merupakan suatu sistem informasi interaktif yang menyediakan informasi atau hasil dari data yang telah diolah, sehingga didapatkannya sebuah hasil yang tepat.

Adapun cirri-ciri sebuah SPK seperti yang dirumuskan oleh Alters Keen adalah sebagai berikut:

1. SPK ditujukan untuk membantu pengambilan keputusan-keputusan yang kurang terstruktur dan umumnya dihadapi oleh para manajer yang berada ditingkat puncak.
2. SPK merupakan gabungan antara kumpulan model kualitatif dan kumpulan data.
3. SPK memiliki fasilitas interaktif yang dapat mempermudah hubungan antara manusia dengan komputer.

4. SPK bersifat luwes dan dapat menyesuaikan dengan perubahan-perubahan yang terjadi (Jurnal Pelita Informatika Budi Darma, Volume : III , Nomor:2 , April 2013, Hal 7).

II.3. *Technique For Order Preference By Similarity to Ideal Solution* (TOPSIS)

Topsis merupakan suatu bentuk metode pendukung keputusan yang didasarkan pada konsep bahwa alternatif yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif tetapi juga memiliki jarak terpanjang dari solusi ideal negatif.

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 positif dan terjauh 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 seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut (Jurnal Pelita Informatika Budi Darma, Volume : III , Nomor:2 , April 2013, Hal 8).

Menurut para ahli diatas, Dapat di simpulkan Topsis adalah konsep dimana banyak digunakan untuk menyelesaikan masalah keputusan secara praktis. Hal ini disebabkan konsepnya sederhana, mudah dipahami, komputasinya

efisien, dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana.

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

$i = 1, 2, 3, \dots, m$; (Persamaan Rumus 1)

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

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

$$V = \begin{bmatrix} w_{11}r_{11} & \cdots & w_{1n}r_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1}r_{m1} & \cdots & w_{mn}r_{nm} \end{bmatrix} \quad (\text{Persamaan Rumus 2})$$

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 (-)

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

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

(Persamaan Rumus 3)

4. Menghitung separasi (*Separation measure*) ini merupakan pengukuran jarak dari suatu alternatif ke solusi ideal positif dan solusi ideal negatif. Perhitungan matematisnya adalah sebagai berikut :

Separation measure untuk solusi ideal positif

$$S_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2}, \text{ dengan } i = 1, 2, 3, \dots, m \quad (\text{Persamaan Rumus 4})$$

5. Menghitung kedekatan relatif terhadap solusi ideal Kedekatan relatif dari alternatif A+ dengan solusi ideal A- direpresentasikan dengan :

$$C_i = \frac{S_i^-}{S_i^- + S_i^+}, \text{ dengan } 0 < C_i^+ < 1 \text{ dan } i = 1, 2, 3, \dots, m \quad (\text{Persamaan Rumus 5})$$

6. Meranking alternatif Alternatif dapat diranking berdasarkan urutan C_i^* . Maka dari itu, alternatif terbaik adalah salah satu yang berjarak terpendek terhadap solusi ideal dan berjarak terjauh dengan solusi ideal negatif (Jurnal Pelita Informatika Budi Darma, Volume : III , Nomor:2 , April 2013, Hal 8-9).

II.4. Basis Data (*Database*)

Database merupakan sekumpulan data berbentuk tabel yang digunakan untuk menyimpan suatu informasi. Penggunaan *database* tidak dapat dipungkiri memang banyak sekali digunakan hampir di semua jenis aplikasi saat ini (Wahana Komputer, 2010).

Database dapat diartikan sebagai sebuah kumpulan data yang terdiri atas satu atau lebih tabel yang terintegrasi satu sama lain. Dalam sebuah

database, seorang pemakai atau *user* dapat mengubah, menambah, menganalisa, bahkan menghapus data dalam tabel-tabel tersebut. Dalam sebuah *database*, tabel berfungsi untuk menyimpan data yang saling berhubungan, misalnya tabel nama hanya berisi daftar nama-nama. Tabel sendiri memiliki dua bagian penting yaitu kolom (*field*) dan baris (*record*). Dalam komputer, *database* dapat diolah menggunakan *software* pengelola database seperti *Microsoft Access*, *Microsoft SQL Server*, *MySQL* (Fajar Rahadian, 2011).

Menurut para ahli diatas, dapat disimpulkan *Database* merupakan kumpulan data yang terdiri dari satu atau lebih tabel yang saling berhubungan satu dengan yang lain dan memiliki kolom dan baris.

Sebelum memperoleh definisi formal basis data, kita akan mencoba memahaminya secara sederhana terlebih dahulu. Istilah basis data tersusun atas dua suku kata, yaitu basis dan data (basis data = basis + data). Dalam sistem bilangan biner, kita dapat menuliskan beberapa contoh bilangan sebagai berikut:

0 → sama dengan 0 dalam sistem bilangan desimal

1 → sama dengan 1 dalam sistem bilangan desimal

10 → sama dengan 2 dalam sistem bilangan desimal

11 → sama dengan 3 dalam sistem bilangan desimal

100 → sama dengan 4 dalam sistem bilangan desimal

(Edhy Sutanta, 2011, Hal 172)

II.5. Normalisasi

Normalisasi diartikan sebagai teknik yang menstrukturkan / mendekomposisi data dalam cara – cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan – penyimpangan (*anomallies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan in-efisiensi pengolahan (Martin, 1975).

Proses normalisasi menghasilkan relasi yang optimal, yaitu (Martin, 1975) :

1. memiliki struktur *record* yang konsisten secara logic.
2. memiliki struktur *record* yang mudah untuk dimengerti.
3. memiliki struktur *record* yang sederhana dalam pemeliharaan.
4. memiliki struktur *record* yang mudah ditampilkan kembali untuk memenuhi kebutuhan pengguna.
5. minimalisasi kerangkapan data guna meningkatkan kinerja system.

II.5.1. Level Normalisasi

Teori normalisasi dibangun menurut konsep level normalisasi. Level normalisasi atau sering disebut sebagai bentuk normal suatu relasi dijelaskan berdasarkan kriteria tertentu pada bentuk normal. Bentuk normal yang dikenal hingga saat ini meliputi bentuk: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF, DKNF, dan RUNF (Edhy Sutanta, 2011, Hal 172-175)

II.5.2. Entity Relationship

Entity Relationship Model/ER_M merupakan suatu model data yang dikembangkan berdasarkan obyek. ER_M digunakan untuk menjelaskan

hubungan antar data dalam basis data kepada pengguna secara logic. ER_M didasarkan pada suatu persepsi bahwa *real world* terdiri atas obyek-obyek dasar yang mempunyai hubungan/kerelasian antar obyek-obyek dasar tersebut (Martin, 1975). ER_M digambarkan dalam bentuk diagram yang disebut diagram ER (*ER_Diagram/ER_D*). untuk menggambarkan ER_D digunakan simbol-simbol grafis tertentu. (Edhy Sutanta, 2011, Hal 173)

II.6. *Unified Modeling Language (UML)*

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, system operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, java, C# atau VB.NET. walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi procedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantic. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna

tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*) (Yuni Sugiarti, ST.M.Kom, 2013, Hal 35).

Unified Modelling Language (UML) adalah menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna (Mulawarman, Vol 6, 2011).

Menurut para ahli diatas, dapat disimpulkan *Unified Modelling Language* (UML) merupakan adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. 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.

Unified Modeling Language (UML) biasa digunakan untuk:

1. Menggambarkan batasan system dan fungsi-fungsi system secara umum, dibuat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.

3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagrams*.
4. Membuat model *behavior* “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *functionality* dengan *stereotypes*.

Berikut ini adalah penjelasan mengenai berbagai diagram UML serta tujuannya:

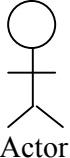
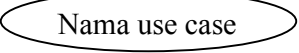

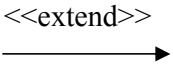
II.6.1. Use Case Diagram

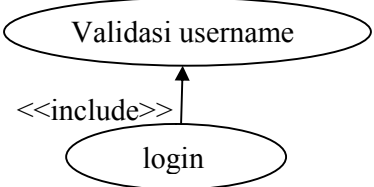
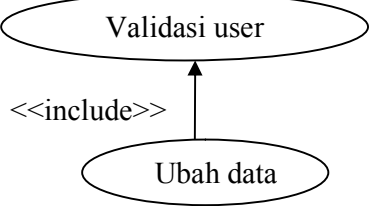
Use case Diagram atau diagram use case merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (M.Salahuddin 2013 ,Hal 155).

Use case Diagram adalah Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case* (Mulawarman, Vol 6, 2011).

Menurut para ahli diatas, dapat disimpulkan deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara deskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna.

Tabel II.1. Simbol – Simbol *Use Case Diagram*

Simbol	Deskripsi
 <p>Actor</p>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem yang akan dibuat sendiri.
<p>Use Case</p> 	Fungsionalitas yang disediakan sistem sebagai unit unit yang saling bertukar pesan antar unit atau aktor.
<p>Asosiasi / <i>Asociation</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> .
<p>Ekstensi / <i>extend</i></p> 	Relasi use case tambahan ke sebuah dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambaha itu.

<p><<include>> →</p> <p><<uses>> →</p>	<p>Ada dua sudut <i>use case</i> pandang yang cukup besar mengenai include di <i>use case</i> :</p> <p>a. <i>Include</i> berarti <i>use case</i> yang ditambahkan dijalankan .misal pada kasus berikut :</p>  <pre> graph BT login((login)) -- <<include>> --> validasi_username((Validasi username)) </pre> <p>b. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan sebelumnya .misal :</p>  <pre> graph BT ubah_data((Ubah data)) -- <<include>> --> validasi_user((Validasi user)) </pre> <p>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	---

(Sumber :M.Shalahuddin, 2013,56-58)

II.6.2.. *Sequence Diagram*

Diagram sekuences menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuences maka harus diketahui objek-objek yang terlibat dalam

sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuence yang harus digambar adalah sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuence sehingga semakin banyak use case yang didefinisikan maka diagram sekuence yang harus dibuat juga semakin banyak (M.shalahuddin 2013, Hal 165).

Sequence Diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram (Mulawarman, Vol 6, 2011).

Menurut para ahli diatas, dapat disimpulkan *Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah sekenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang diletakkan diantara objek-objek ini di dalam *use case*.


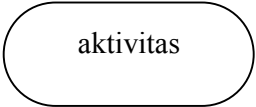
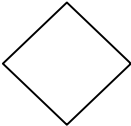

II.6.3. *Activity Diagram*


Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas mendukung perilaku paralel (Yuni Sugiarti ST.M.Kom 2013, Hal 61)

Activity Diagram adalah Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas (Mulawarman, Vol 6, 2011).

Menurut para ahli diatas, dapat disimpulkan *Activity Diagram* merupakan teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus.

Tabel II.2. Simbol – Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status wal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah setatus awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber :M.Shalahuddin, 2013,162)

II.6.4. *Class Diagram*

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Class Diagram adalah menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas (Mulawarman, Vol 6, 2011).

Menurut para ahli diatas, dapat disimpulkan *Class Diagram* merupakan sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- Atribut mendeskripsikan properti dengan sebaris teks didalam kontak kelas tersebut.

- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

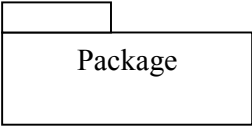
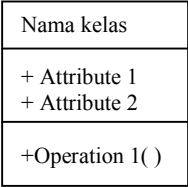
Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

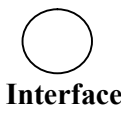
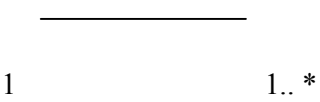
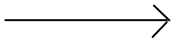
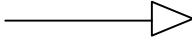
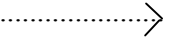

Diagram kelas menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi dan lain-lain.

Kelas memiliki tiga area pokok :

1. Nama
2. Atribut
3. Operasi

Tabel II.3. Simbol – Simbol Class Diagram

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem

Antarmuka / interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek
Asosiasi 	Relasi antara kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah / directed asosiasi 	Relasi antara kelas dengan makna kelas satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	Relasi antara kelas dengan makna generalisasi-spesialisasi (umum Khusus)
Kebergantungan / dependency 	Relasi antara kelas dengan makna keberuntungan antara kelas
Agregasi 	Relasi antara kelas dengan makna semua-bagian (whole-part)

(Sumber :Yuni Sugiarti, S.T.M.Kom, 2013,Hal 58-59)

II.7. Bahasa Pemrograman *Microsoft Visual Basic 2010*

Visual Basic 2010 merupakan salah satu bagian dari pemrograman terbaru yang dikeluarkan oleh *Microsoft Visual Studio 2010*. Sebagai pengembangan terintegrasi atau *IDE* andalan yang dikeluarkan oleh *Microsoft, Visual Studio 2010* menambahkan perbaikan – perbaikan fitur dan fitur baru yang lebih lengkap dibandingkan versi *Visual Studio* pendahulunya, Yaitu

Microsoft Visual Studio 2008. *Visual studio* berisi beberapa jenis *IDE* pemrograman seperti *Visual Basic*, *Visual C++* , *Visual Web Developer*, *Visual C#* dan *Visual F#* (Wahana Komputer, 2010, Hal 2).

Microsoft Visual Studio 2010 merupakan perangkat lunak yang terintegrasi, di dalamnya terdapat beberapa paket *software* yang dapat digunakan oleh programer dalam membangun sebuah program profesional, diantaranya adalah *Visual Basic*, *Visual J#*, *Visual C*, *#Visual C++* dan *Java Runtime* yang sama-sama berada dalam naungan *platform Microsoft* (Fajar Rahadian, 2011).

Menurut para ahli diatas, dapat disimpulkan *Visual Basic* merupakan salah satu bahasa pemograman yang andal dan banyak digunakan oleh pengembang untuk membangun berbagai macam aplikasi *windows*.

II.8. *Sql Server 2008 R2*

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang *database*. *SQL Server* adalah *DBMS* (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. *SQL Server* 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server* 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data (Wahana Komputer, 2010).

SQL Server 2008 adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan *multiuse* (Taufik Hidayat, vol 1, 2012).

Menurut para ahli diatas, dapat disimpulkan *SQL Server* 2008 adalah salah satu *Relational Database Management System* (RDBMS) yang diproduksi oleh *Microsoft*. *SQL Server* adalah hasil kerja sama antara *Microsoft* dengan *Sybase* untuk memproduksi sebuah *software* penyimpanan data (*database*).

Microsoft merilis *SQL Server* 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut. Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan 2 jenis yaitu :

- Versi 32-bit(x86), yang biasanya digunakan untuk komputer dengan single prosesor (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi Windows XP.
- Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya Core 2 Duo) dan system operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7.

Sedangkan secara keseluruhan terdapat versiversi seperti berikut ini:

- Versi Compact, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi desktop pada *SQL Server* 2000. Versi ini juga digunakan pada handled drvice seperti Pocket PC, PDA, SmartPhone, Tablet PC.

Versi Express, ini adalah versi “Ringan” dari semua versi yang ada(tetapi versi ini berbeda dengan versi compact) dan paling cocok untuk latihan para pengembang aplikasi. Versi ini memuat Express Manager standar, integrasi dengan CLR dan XML (Jurnal : Sistem Informasi STMIK GI MDP , Wahana Komputer 2010, hal.2).