

BAB II

TINJAUAN PUSTAKA

II.1. Sistem.

Menurut Asbon Hendra (2012 : 157), sistem dapat diartikan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

II.1.1. Karakteristik Sistem.

a. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk suatu kesatuan. Komponen – komponen sistem dapat berupa suatu bentuk subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut dengan *supra sistem*.

b. Batasan Sistem (*Boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan yang lainnya berbeda tetapi tetap saling berinteraksi.

c. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu di luar batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem 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.

d. 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.

e. Masukan Sistem (*Input*).

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan Sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran.

f. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem, meliputi *output* yang berguna, dan *output* yang tidak berguna dikenal sebagai sisa pembuangan.

g. Pengolahan Sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

h. Tujuan Sistem (*Goal*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran dan tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. (Asbon Hendra; 2012; 158-160).

II.1.2. Klasifikasi Sistem.

a. Sistem Abstrak (*Abstract System*)

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

b. Sistem Fisik (*Physical System*)

Merupakan Sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya. Contohnya, sistem komputer, sistem akuntansi, sistem produksi, dan lain-lain.

c. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam, dalam artian tidak dibuat oleh manusia, seperti tata surya, sistem galaksi, sistem reproduksi, dan lain-lain.

d. Sistem Buatan Manusia (*Human Mode System*)

Sistem yang dirancang manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human machine system*, contohnya sistem informasi.

e. Sistem Tertentu (*Deterministic System*)

Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Contohnya sistem komputer.

f. Sistem Tak Tentu (*Probabilistic System*)

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

g. Sistem Tertutup (*Closed System*)

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

h. Sistem Terbuka (*Open System*)

Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomatis, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagian bagian dari sistem yang digunakan dalam masyarakat modern.

Sistem terotomatis mempunyai sejumlah komponen, yaitu :

- Perangkat keras (CPU, *disk*, *printer*, *tape*).
- Perangkat lunak (sistem operasi, sistem *database*, program pengontrol komunikasi, program aplikasi).
- Personel (yang mengoperasikan sistem, menyediakan masukan, mengonsumsi keluaran dan melakukan aktivitas manual yang mendukung sistem)
- Data (yang harus tersimpan dalam sistem selama jangka waktu tertentu)

- Prosedur (instruksi dan kebijakan untuk mengoperasikan sistem).

Sistem terotomatis terbagi dalam sejumlah kategori, yaitu :

a. Online Systems.

Sistem online adalah sistem yang menerima langsung *input* pada area dimana *input* tersebut direkam dan menghasilkan *output* yang dapat berupa hasil komputasi pada area dimana mereka butuhkan.

b. Real-time Systems.

Sistem *real-time* adalah mekanisme pengontrolan, perekaman data, dan pemrosesan yang sangat cepat sehingga *output* yang dihasilkan dapat diterima dalam waktu yang relatif sama.

c. Decision Support System and Strategic Planning System.

Sistem ini memproses transaksi organisasi secara harian dan membantu para manajer mengambil keputusan, mengevaluasi, dan menganalisa tujuan organisasi.

d. Knowledge-based System.

Program komputer ini dibuat mendekati kemampuan dan pengetahuan seorang pakar. (Asbon Hendra; 2012; 160-163).

II.1.3. Pelaku Sistem.

- a. Pemakai :** Pada umumnya ada tiga jenis pemakai, yaitu operasional, pengawas, dan eksekutif.

b. Manajemen : Umumnya terdiri dari tiga jenis, yaitu manajemen pemakai yang bertugas menangani pemakaian dimana sistem baru diterapkan, manajemen sistem yang terlibat dalam pengembangan sistem itu sendiri, dan manajemen umum yang terlibat dalam strategi perencanaan sistem dan sistem pendukung pengambilan keputusan.

c. Pemeriksa : Pemeriksa biasanya menentukan segala sesuatunya berdasarkan ukuran-ukuran standar yang dikembangkan pada banyak perusahaan sejenis.

d. Penganalisa Sistem : Fungsi – fungsinya antara lain sebagai :

- Arkeolog, yaitu yang menelusuri bagaimana sebenarnya sistem lama berjalan, bagaimana sistem tersebut dijalankan, dan segala hal yang menyangkut sistem lama.
- Inovator, yaitu yang membantu mengembangkan dan membuka wawasan pemakai bagi kemungkinan-kemungkinan lain.
- Mediator, yaitu yang menjalankan fungsi komunikasi dari semua level, antara lain pemakai, manajer, *programmer*, pemeriksa, dan pelaku sistem yang lainnya yang mungkin belum punya sikap dan cara pandang yang sama.
- Pimpinan proyek. Penganalisa sistem haruslah personel yang lebih berpengalaman dari *programmer* atau desainer.

e. Pendesign Sistem : Pendesain sistem menerima hasil penganalisa sistem berupa kebutuhan pemakai yang tidak berorientasi pada teknologi tertentu,

yang kemudian ditransformasikan ke desain arsitektur tingkat tinggi dan dapat diformulasikan oleh *programmer*.

f. Programmer : Mengerjakan dalam bentuk program dari hasil desain yang telah diterima dari pendesain.

g. Personel Pengoperasian : Bertugas dan bertanggung jawab dipusat komputer, misalnya jaringan, keamanan perangkat keras, keamanan perangkat lunak, pencetakan, dan *backup*. (Asbon Hendra; 2012; 163-165).

II.1.4. Konsep Dasar Informasi.

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi, ada suatu proses transformasi data menjadi suatu informasi = *input-proses-output*. (Asbon Hendra; 2012; 167).

II.1.5. Defenisi Sistem Informasi.

”Suatu sistem terintegritasi yang mampu menyediakan informasi yang bermanfaat bagi penggunanya”.

”Sebuah sistem terintegrasi atau sistem 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 Robert A. Leitch, sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Ada empat operasi dasar dari sistem informasi, yaitu mengumpulkan, mengolah, menyimpan dan menyebarkan informasi. Informasi mungkin dikumpulkan dari lingkungan dalam atau luar dan memungkinkan didistribusikan ke dalam atau ke luar organisasi. (Asbon Hendra; 2012; 168-169).

II.1.6. Komponen Fisik Sistem Informasi.

- a. Perangkat keras komputer: CPU, *storage*, perangkat *input/output*, terminal untuk interaksi, serta media komunikasi data.
- b. Perangkat lunak komputer: perangkat lunak sistem (sistem operasi dan *utility*-nya), perangkat lunak umum aplikasi (bahasa pemograman), dan perangkat lunak aplikasi (aplikasi akuntansi).

- c. Basis Data: penyimpanan data pada media penyimpan komputer.
- d. Prosedur: langkah-langkah penggunaan sistem.
- e. *Personnel* untuk pengelolaan operasi (SDM), meliputi:
 - a. *Clerical personnel* (untuk menangani transaksi dan pemrosesan data dan melakukan *inquiry* = operator).
 - b. *First level manager*, untuk mengelola pemrosesan data didukung dengan perencanaan, penjadwalan, identifikasi situasi *out-of-control*, dan pengambilan keputusan level menengah ke bawah.
 - c. *Staff specialist*, digunakan untuk analisis perencanaan dan pelaporan.
 - d. *Management*, untuk pembuatan laporan berkala, permintaan khusus, analisis khusus, laporan khusus, serta pendukung identifikasi masalah dan peluang. Aplikasi = program + prosedur pengoperasian. (Asbon Hendra; 2012; 170-171).

II.2. Sistem Informasi geografis.

Sistem Informasi Georafis (SIG) atau *Georaphic Information Sistem (GIS)* merupakan suatu sistem informasi yang berbasis komputer, dirancang untuk bekerja dengan menggunakan data yang memiliki informasi spasial (bereferensi keruangan). Sistem ini menangkap, mengecek, mengintegrasikan, memanipulasi, menganalisa, dan menampilkan data yang secara spasial mereferensikan kepada kondisi bumi. Teknologi SIG mengintegrasikan operasi-operasi umum *database*, seperti *query* dan analisa statistik, dengan kemampuan visualisasi dan analisa yang unik yang dimiliki oleh pemetaan. Kemampuan inilah yang membedakan

SIG dengan Sistem Informasi lainnya yang membuatnya menjadi berguna berbagai kalangan untuk menjelaskan kejadian, merencanakan strategi, dan memprediksi apa yang terjadi.

Menurut Dra. Romenah (2010:5) berikut ini beberapa definisi SIG menurut para ahli :

1. Menurut Aronaff, 1989.

SIG adalah sistem informasi yang didasarkan pada kerja komputer yang memasukkan, mengelola, memanipulasi dan menganalisa data serta memberi uraian.

2. Menurut Barrough, 1986.

SIG merupakan alat yang bermanfaat untuk pengumpulan, penimbunan, pengambilan kembali data yang diinginkan dan penayangan data keruangan yang berasal dari kenyataan dunia.

3. Menurut Marble et al, 1983.

SIG merupakan sistem penanganan data keruangan.

4. Menurut Berry, 1988.

SIG merupakan sistem informasi, referensi internal, serta otomatisasi data keruangan.

5. Menurut Calkin dan Tomlison, 1984.

SIG merupakan sistem komputerisasi data yang penting. (Lusi Melian dan Hilman Agus).

Sistem Informasi Geografis dibagi menjadi dua kelompok yaitu sistem manual (*analog*), dan sistem otomatis (yang berbasis digital komputer).

Perbedaan yang paling mendasar terletak pada cara pengelolaannya. Sistem Informasi manual biasanya menggabungkan beberapa data seperti peta, lembar transparansi untuk tumpang susun (*overlay*), foto udara, laporan statistik dan laporan survey lapangan. Kesemua data tersebut dikompilasi dan dianalisis secara manual dengan alat tanpa komputer. Sedangkan Sistem Informasi Geografis otomatis telah menggunakan komputer sebagai sistem pengolah data melalui proses digitasi. Sumber data digital dapat berupa citra satelit atau foto udara digital serta foto udara yang terdigitasi. Data lain dapat berupa peta dasar terdigitasi.

Menurut Anon (2003: 73) ada beberapa alasan mengapa perlu menggunakan SIG, diantaranya adalah:

- a. Menggunakan data spasial maupun atribut secara terintegrasi.
- b. Dapat digunakan sebagai alat bantu interaktif yang menarik dalam usaha meningkatkan pemahaman mengenai konsep lokasi, ruang, kependudukan, dan unsur-unsur.
- c. SIG dapat memisahkan antara bentuk presentasi dan basis data.
- d. Memiliki kemampuan menguraikan unsur-unsur yang ada dipermukaan bumi kedalam beberapa *layer* atau *coverage* data spasial.
- e. SIG memiliki kemampuan yang sangat baik dalam memvisualisasikan data spasial berikut atributnya.
- f. Semua operasi SIG dapat dilakukan secara interaktif.
- g. SIG dengan mudah menghasilkan peta-peta tematik.

- h. Semua operasi SIG dapat dicustomize dengan menggunakan perintah-perintah dalam bahasa *script*.
- i. Perangkat lunak SIG menyediakan fasilitas untuk berkomunikasi dengan perangkat lunak lain.
- j. SIG sangat membantu pekerjaan yang erat kaitannya dengan bidang spasial dan *geoinformatika*.

II.2.1. Kemampuan Sistem Informasi geografis.

1. Memasukan data mengumpulkan data geografi (spasial dan atribut).
2. Mengintegrasikan data geografi (spasial dan atribut).
3. Memeriksa, meng-update (meng-edit) data geografis (spasial dan atribut).
4. Menyimpan dan memanggil kembali data geografi (spasial dan atribut).
5. Mempresentasikan atau menampilkan data geografi (spasial dan atribut).
6. Mengolah data geografi (spasial dan atribut).
7. Manipulasi data geografi (spasial dan atribut).
8. Menganalisa data geografi (spasial dan atribut).
9. Menghasilkan keluaran (*output*) data geografi.

II.2.2. Web-Geographic Information System (Web-GIS).

Web-GIS atau *Web-Geographic Information System* adalah sistem informasi geografis atau sistem informasi keruangan pada, diatas, dan/atau di bawah permukaan bumi berbasis web. Web-GIS menggabungkan desain grafis pemetaan, peta digital dengan analisa geografis, pemrograman computer dan

basis data yang terhubung menjadi satu bagian desain web dan web-pemetaan. Aplikasi *Web-Gis* dapat digunakan secara global tidak bergantung pada platform maupun sistem operasi. (Hidra Wira Buana; 2011).

II.2.3. Website.

Pada saat membuat sebuah website, kita tidak bisa langsung mengharapkan website yang baru dibuat menjadi sebuah website yang sempurna, maka besar kemungkinan website yang kita buat tidak akan pernah diluncurkan.

Untuk tahapan pengembangan website kita bisa menggunakan pendekatan System Development Life Cycle (SDLC) sebagaimana disampaikan oleh Laudon (2009 : 4-6). Pendekatan tersebut memiliki 5 tahapan utama yang meliputi :

- a. Analisis/perencanaan.
- b. Desain.
- c. Pembuatan website.
- d. Pengujian, dan
- e. Implementasi.

Mari kita lihat satu demi satu tahapan tersebut dan apa saja item yang perlu dipertimbangkan dalam setiap tahap :

a. Analisis/Perencanaan.

Dalam tahap ini, kita harus mulai dengan menentukan tujuan dan kegunaan pembuatan web. Apakah kita membuat website untuk menjual produk atau jasa, berbagi ide dan informasi, promosi, atau social. Mengetahui tujuan

pembuatan website ini sangat penting karena akan membantu kita menentukan arah bagi tahapan selanjutnya.

b. Desain.

Pada tahapan desain ini, kita harus menentukan hal-hal berikut ini :

- Siapa saja calon pengunjung.
- Layout.
- Warna tema.
- Pengorganisasian halaman web, mulai dari halaman utama hingga sub-sub halaman.
- Konten apa saja yang perlu disediakan.

Dalam tahap ini output yang dikeluarkan antara lain adalah mock up design. Perhatikan juga jenis browser apa saja yang digunakan mayoritas pengunjung, apakah PC (termasuk apakah banyak yang menggunakan smartphone), resolusi monitor, dan lain sebagainya.

c. Pembuatan Website.

Proses pembuatan website ini adalah proses utama, mulai dengan membuat sitemap untuk memberikan arah dan gambaran global tentang halaman apa saja yang diperlukan.

d. Pengetesan.

Tidak ada proses pembuatan website yang langsung sempurna. Lakukan pengecekan pengecekan dengan membuka halaman website kita satu persatu di berbagai browser, PC, dan resolusi monitor. Lakukan validasi kode,

pengecekan, dan perbaikan error yang terjadi. Pastikan juga navigasi mudah digunakan.

Dalam tahap ini, Anda juga bisa melakukan optimasi, misalnya gambar yang terlalu besar ukurannya perlu diperkecil, atau pembersihan kode website dari *chunk code* (kode yang tidak diperlukan).

e. Implementasi.

Pada tahap ini, Anda juga perlu melakukan pemeliharaan untuk memastikan bahwa konten selalu update serta melakukan perbaikan pada website tersebut.

(Adi Prasetyo; 2012; 394-397)

II.3. Mapinfo.

MapInfo diminati oleh pengguna SIG karena mempunyai karakteristik yang menarik, seperti mudah digunakan harga relatif murah, tampilan yang interaktif dan menarik, *user friendly* dan dapat di-*customised* menggunakan bahasa script yang dimiliki.

MapInfo Profesional 7.5 hadir bersama dengan *user interface* yang diimplementasikan dalam berbagai bentuk menu, *tools* dan lain sebagainya. Misalnya toolbar standar, toolbar drawing untuk menggambar peta dan toolbar main yang merupakan toolbar utama pada MapInfo. (Agus Supriyanto dan Eny Wulandari ; 2010).

II.4. Bangunan Tua Di Kota Medan.

Pertumbuhan kota Medan sejak masa lalu hingga sekarang, meninggalkan banyak bangunan tua bersejarah di beberapa kawasan. Sejarah Kota Medan yang sekarang dimulai dari kawasan kota tua dengan bangunan tua sebagai ciri identitasnya yang mengandung nilai dan makna sejarah dan arsitektural, sehingga secara total mencerminkan citra Kota Medan itu sendiri. Tanpa adanya kawasan kota tua tersebut, dipastikan bahwa pemerintah dan bahkan masyarakat akan kehilangan tentang asal-usul lingkungannya yang tidak mempunyai orientasi pada masa lampau. (Damardjati Kun Marjanto, Ernayanti, Robby Ardiwijaya ; 2013 ; 17).

Sama juga halnya kawasan kota lama Kesawan. Menurut sejarah, daerah Kesawan dahulunya adalah cikal bakal berdirinya Kota Medan pada tahun 1870, yang wilayahnya terhubung dari daerah Kesawan hingga Labuhan Deli. Pada awal abad ke- 19 pembangunan Kota Medan menjadi sedemikian pesat, ditandai dengan banyaknya infrastruktur yang dibangun pada masa itu. Pembangunan Kota Medan dimulai pada tahun 1880 dengan berdirinya bangunan-bangunan baru. Tahun 1885 rel kereta api sudah dibangun dan stasiun kereta api yang berdekatan dengan lapangan Esplanade juga sudah berdiri pada tahun 1887. Kemudian diikuti dengan pembangunan Hotel De Boer dan Bank Java. Hotel De Boer dibangun pada tahun 1889 dan menjadi terkenal di Eropa karena kemewahannya dan keindahannya pada masa itu. Selain perusahaan kereta api, hotel, dan lapangan Esplanade (sekarang Lapangan Merdeka/*Merdeka Walk*), berbagai macam gedung perkantoran dan perusahaan dibangun di sekeliling lapangan Esplanade, di

antaranya adalah bangunan kantor perusahaan perkebunan Inggris Harrison & Crosfield yang berdiri tahun 1909 dan terletak di sudut kawasan Kesawan dan lapangan Esplanade.

Kawasan Kesawan sampai saat ini masih menyisakan bangunan tua yang sebagian besar masih difungsikan untuk kegiatan ekonomi, seperti kantor, warung, restoran, butik, dan toko olahraga. Menurut sejarahnya, dahulu Kesawan masuk ke dalam wilayah perkebunan, kemudian berkembang menjadi pertokoan yang dibangun oleh orang-orang keturunan Tionghoa. Misalnya yang terkenal adalah Tjong A Fie, seorang saudagar besar Tionghoa pada masa penjajahan Belanda. Rumah Tjong A Fie merupakan bangunan dengan perpaduan gaya Cina dan Eropa yang selesai dibangun tahun 1900. Disain rumah Tjong A Fie mirip dengan rumah kerabatnya di Pulau Penang, Malaysia, bernama Cheong Fatt Tze. Tjong A Fie datang dari Kanton pada tahun 1875 bersama saudaranya, Tjong Yong Hian, dan kemudian menjadi *Majoer der Chineezen*. Tjong A Fie adalah salah seorang pendiri *Colonial Institute* di Amsterdam yang saat ini bernama *Koninklijk Instituut voor de Tropen*. (Damardjati Kun Marjanto, Ernayanti, Robby Ardiwijaya ; 2013 ; 17).

II.5. PHP Hypertext Preprocessor.

PHP atau *Hyperxt Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website yang dinamis. PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa *server side scripting*. Artinya bahwa dalam setiap/untuk menjalankan PHP, wajib adanya web server.

PHP ini bersifat *open source* sehingga dapat dipakai secara cuma-cuma dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi *Windows* maupun *Linux*.

II.5.1. Cara Kerja PHP.

Secara prinsip, server akan bekerja apabila ada permintaan dari *client*, yaitu kode-kode PHP. Client tersebut akan dikirimkan kepada server, kemudian server akan mengembalikan pada halaman sesuai yang diminta.

II.5.2. Sintak Dasar PHP.

Sintaks dasar PHP meliputi bagaimana cara memulai suatu struktur pemrograman PHP. Ada empat cara memulai pemrograman PHP, diantaranya :

- `<?php .....?>`
- `<? .....?>`
- `<script Language="php"> ..... </script>`
- `<% ..... %>`

Dari beberapa sintaks tersebut, yang paling banyak digunakan adalah cara pertama dan kedua. (Agus Saputra; 2011; 1-6).

II.6. Macromedia Dreamweaver CS5.

Macromedia Dreamweaver adalah software aplikasi yang digunakan sebagai HTML editor profesional untuk mendesain web secara visual. Aplikasi ini juga yang biasa dikenal dengan istilah WYSIWYG (*What You See Is What You Get*). Yang artinya bahwa anda tidak harus berurusan dengan tag-tag HTML untuk membuat situs. Selain itu, Dreamweaver juga memberikan keleluasaan kepada anda untuk menggunakannya sebagai media penulisan bahasa pemrograman web. (Madcoms; 2010; 1).

II.6.1. Mengenal Alur Kerja Dreamweaver.

Sebuah proyek web yang kompleks biasanya memiliki alur kerja yang sistematis agar hasil yang tercapai dapat optimal. Berikut ini adalah alur kerja pembuatan sebuah website dengan menggunakan dreamweaver.

a. Membuat rancangan dan mencari lokasi situs

Tentukanlah kemana file akan diletakkan serta lakukan pengujian kebutuhan lokasi situs. Pertimbangkan juga kebutuhan teknis dari situs tersebut, seperti akses pemakai, browser yang digunakan, kebutuhan *plug-in*, serta pengaturan download. Setelah proses organisir dan pembuatan struktur, berikutnya Anda dapat menentukan lokasi situs.

b. Melakukan organisasi dan manajemen file situs

Didalam panel **files**, anda dengan mudah menambahkan, menghapus, dan mengganti nama file atau folder yang diperlukan. Panel **files** juga memiliki banyak *tool* untuk memanajemen situs Anda.

c. Membuat layout halaman web

Tambahkan asset dan unsur-unsur desain halaman web Anda, seperti teks, image (gambar), rollover image, warna, movie, suara, link HTML, jump menu, dan background halaman. Dreamweaver juga menyediakan fasilitas untuk memaksimalkan hasil website, dan melakukan pengujian halaman untuk memastikan kecocokan dengan browser web yang berbeda.

d. Membuat halaman dengan kode program

Fasilitas kode merupakan langkah alternatif untuk mendesain sebuah halaman web. Dreamweaver tidak hanya memberikan fasilitas untuk membuat dan mengedit desain visual, namun juga menyediakan fasilitas kode-kode program yang canggih.

e. Mengatur web menjadi aplikasi dengan isi yang dinamis

Banyak sekali halaman web yang mempersilahkan pengunjung untuk dapat melihat dan mengisi data dalam *database*.

f. Menciptakan halaman dinamis

Dalam dreamweaver, Anda dapat menampilkan beberapa tipe isi yang dinamis, diantaranya : menampilkan data record dari sebuah database, membuat form batasan, dan menggunakan komponen JavaBeans. Untuk

menambahkan isi yang dinamis, Anda cukup klik tahan geser kedalam halaman web.

g. Menguji situs dan melakukan publish

Pengujian halaman yang telah Anda buat, merupakan suatu keharusan dalam sebuah siklus pembuatan sebuah situs. Dan pada akhir siklus, maka Anda harus melakukan publis untuk situs tersebut. Sebaiknya Anda memiliki jadwal secara periodik untuk menguji apakah web Anda tersebut masih dapat berfungsi dan berjalan dengan baik. (Madcoms; 2010; 8-9).

II.6.1. Mengenal Tipe File Dreamweaver.

Anda dapat bekerja dengan berbagai tipe file Dreamweaver. Tipe file utama adalah XHTML. File HTML atau *Hypertext Markup Language files* berisi bahasa tag dasar yang memberikan respon untuk menampilkan suatu halaman web dalam browser. File HTML dapat disimpan dengan ekstensi .html atau .xhtml. Secara default Dreamweaver menyimpan file sebagai .html. Berikut ini adalah beberapa tipe file yang dapat Anda gunakan dalam Dreamweaver :

- a. **CSS**, file *Cascading Style Sheet* memiliki ekstensi **.css**. CSS digunakan untuk format isi HTML dan mengendalikan posisi berbagai unsur-unsur halaman.
- b. **GIF**, file *Graphics interchange Format* memiliki ekstensi **.gif**. Format GIF paling populer untuk desain grafis halaman web, diantaranya untuk kartun, logo, grafis, untuk area transparan, dan animasi. GIF memiliki maksimum 256 jenis warna.

- c. **JPEG**, *Joint Photographic Experts Group* memiliki ekstensi **.jpg**, dan biasanya untuk foto dan gambar dengan kualitas warna yang tinggi. Format JPEG sangat baik untuk hasil foto digital atau hasil scan, gambar dengan tekstur, gambar dengan transisi warna gradasi, dan gambar-gambar yang memerlukan lebih dari 256 jenis warna.
- d. **XML**, *Extensible Markup Language* memiliki ekstensi **.xml**. File ini berisi data didalam suatu form mentah yang dapat diformat dengan menggunakan XSL (*Extensible Stylesheet Language*).
- e. **XSL**, *Extensible Stylesheet Language* memiliki ekstensi **.xsl** atau **.xslt**. File ini digunakan untuk tipe data XML yang ingin Anda tampilkan dalam suatu halaman web.
- f. **CFML**, *ColdFusion Markup Language* memiliki ekstensi **.cfm**. File ini digunakan untuk memproses sebuah halaman dinamis.
- g. **PHP**, *Hypertext Preprocessor* memiliki ekstensi **.php**, dan juga digunakan untuk memproses sebuah halaman dinamis.

II.7. MySQL.

SQL merupakan kependekan dari *Structured Query Language*. SQL digunakan untuk berkomunikasi dengan sebuah *database*. SQL merupakan bahasa standard untuk sistem manajemen *database* relasional.

Statement SQL digunakan untuk melakukan tugas-tugas seperti melakukan *update* terhadap *database*, atau mengambil data dari sebuah *database*. Perintah standar SQL seperti “*Select*”, “*Insert*”, “*Update*”, “*Delete*”, “*Create*”, dan

“*Crop*” dapat digunakan untuk melakukan hampir semua hal yang perlu dilakukan terhadap sebuah *database*. (Adi Prasetyo; 2012; 182).

II.8. Basis Data.

II.8.1. Pengertian Basis Data.

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah *file* teks maupun *Database Management System* (DBMS).

Kebutuhan basis data dalam sistem informasi meliputi :

- Memasukkan, menyimpan, dan mengambil data.
- Membuat laporan berdasarkan data yang telah disimpan.

Tujuan dari dibuatnya tabel – tabel adalah untuk menyimpan data kedalam tabel – tabel agar mudah diakses. Oleh karena itu, untuk merancang tabel – tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris – baris data (*record*) dimana setiap baris terdiri dari beberapa kolom.

II.8.2. DBMS.

DBMS (*Database Management System*) atau dalam bahas Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. Suatu sistem aplikasi yang disebut DBMS jika terpenuhi persyaratan minimal sebagai berikut :

1. Menyediakan fasilitas untuk mengelola akses data.
2. Mampu menangani integritas data.
3. Mampu menangani akses data yang dilakukan secara
4. Mampu menangani *backup* data.

DBMS sudah mulai berkembang sejak tahun 1960-an, kemudian sekitar tahun 1970-an mulai berkembang teknologi *Relational* DBMS yaitu DBMS berbasis *relasional* model. *Relasional* model pertama kali dikembangkan oleh Edgar J. Codd pada tahun 1970-an.

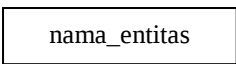
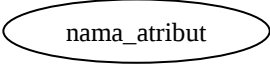
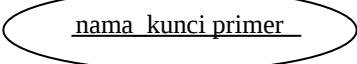
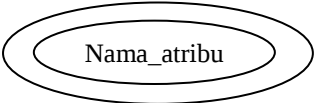
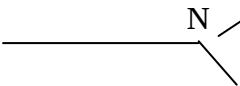
Pada tahun 1980-an mulai berkembang *Object Oriented* DBMS (OODBMS). OODBMS berkembang seiring dengan perkembangan pemrograman berorientasi objek. Secara umum dapat diartikan bahwa OODBMS merupakan DBMS yang memandang data sebagai suatu objek. Saat ini OODBMS juga cukup berkembang namun belum dapat menggeser kepopuleran RDBMS. (Rosa A.S, M.Shalahuddin; 2013; 43-45).

II.8.3. ERD.

ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa

aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi Chen.

Berikut adalah tabel II.1 simbol – simbol yang digunakan pada ERD dengan notasi Chen :

Simbol	Deskripsi
Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
Atribut multivalai / <i>multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas dimana kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan yang lain disebut kardinalitas.

Tabel II.1 Simbol- Simbol ERD.

Sumber : (Rosa A.S, M.Shalahuddin; 2013; 50-51).

II.8.3. Kamus Data.

Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi :

- Nama – nama dari data.
- Digunakan pada merupakan proses – proses yang berkaitan data.
- Deskripsi merupakan deskripsi data.
- Informasi tambahan seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol. Berikut adalah tabel II.2 simbol – simbol yang digunakan pada kamus data:

Simbol	Deskripsi
=	Disusun atau terdiri dari
+	Dan
[1]	Baik...atau...
{ }n	N kali diulang/ bernilai banyak
()	Data opsional
* ... *	Batas komentar

Tabel II.2 Simbol-Simbol Kamus Data

Sumber : (Rosa A.S, M.Shalahuddin; 2013; 73-74).

II.9. *Unified Modelling Language (UML).*

UML (*Unified Modelling Language*) adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefenisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek. (Rosa A.S, M.Shalahuddin; 2013; 133).

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. (Rosa A.S, M.Shalahuddin; 2013; 137-138).

Ada beberapa pendapat para ahli tentang pengertian *UML* diantaranya :

- Menurut (Braun, *et. al.* 2001), *Unified Modelling Language (UML)* adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual.
- Menurut (Whitten, *et, al.* 2004), *Unified Modelling Language (UML)* merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. (Haviluddin; 2011; 1).
- Menurut (Eriksson dan Penker,1998), *Unified Modelling Language (UML)* merupakan alat bantu, bahasa pemodelan yang dapat digunakan untuk rancang bangun berorientasi- objek. *UML* dapat digunakan untuk spesifikasi, visualisasi dan dokumentasi sistem pada *fase* pengembangan. (Haviluddin; 2011; 8-9).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan *UML* adalah sebagai berikut :

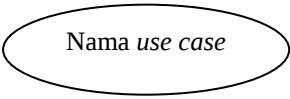
a. *Usecase* Diagram

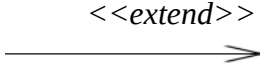
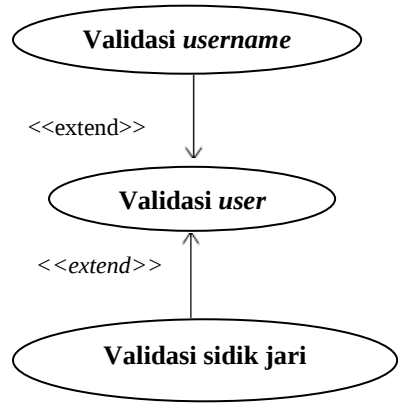
Usecase diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

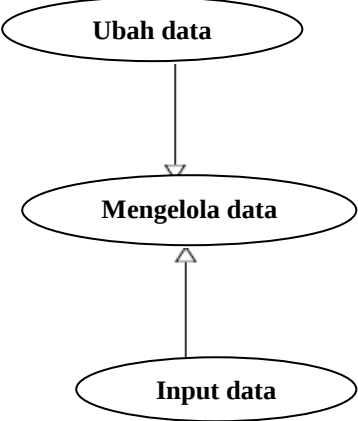
Ada dua hal utama pada *use case* yaitu pendefenisian apa yang disebut aktor dan *use case*.

- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah tabel II.3. simbol-simbol yang ada pada diagram *use case*:

Simbol	Deskripsi
<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal diawal frase nama <i>use case</i>

Aktor / <i>actor</i>  nama aktor	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal nama <i>actor</i>.
Asosiasi/ <i>association</i> 	Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu: mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, missal  Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.
Generalisasi/<i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:

	 Arah panah mengarah pada use case yang menjadi generalisasinya (umum).
Menggunakan / <i>include</i> / <i>uses</i> <code><<include>></code>  <code><<uses>></code> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : • <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan. • <i>include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.

Tabel II.3. Simbol-Simbol Diagram Use Case.

Sumber : (Rosa A.S, M.Shalahuddin; 2013; 155-158).

b. Diagram Aktivitas (*Activity Diagram*)

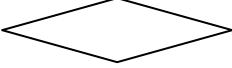
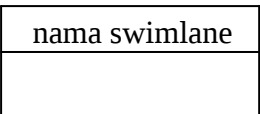
Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas

menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefenisikan hal-hal berikut:

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefenisikan.
- Urutan atau pengelompokkan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.

Berikut adalah tabel II.4. simbol – simbol yang ada pada *activity diagram* :

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status 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 bertanggungjawab terhadap aktivitas yang terjadi.

Tabel II.4. Simbol-Simbol Activity Diagram.

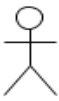
Sumber : (Rosa A.S, M.Shalahuddin; 2013; 161-163).

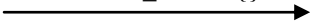
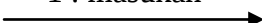
c. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat digram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Berikut adalah tabel II.5. simbol - simbol yang ada pada *sequence diagram*

:

Simbol	Deskripsi
Aktor  nama aktor	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal nama <i>actor</i>.
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
Objek <u>nama objek :nama kelas</u>	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah tahapan yang dilakukan didalamnya.

Pesan tipe create <code><<create>></code> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe call <code>1 : nama_metode{ }</code> 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe send <code>1 : masukan</code> 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return <code>1 : keluaran</code> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe destroy <code><<destroy>></code> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i> .

Tabel II.5. Simbol-Simbol Sequence Diagram.

Sumber : (Rosa A.S, M.Shalahuddin; 2013; 165-167).

d. *Class Diagram* (Diagram Kelas)

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang yang disebut atribut metode atau operasi.

- Atribut merupakan variabel –variabel yang dimiliki oleh suatu kelas.
- Operasi atau metode adalah fungsi – fungsi yang dimiliki oleh suatu kelas.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

a. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

b. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

c. Kelas yang di ambil dari pendefenisian use case (*controller*).

Kelas yang menangani fungsi – fungsi yang harus ada diambil dari pendefenisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

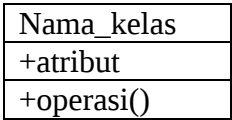
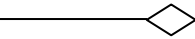
d. Kelas yang di ambil dari pendefenisian data (*model*).

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun yang akan disimpan ke basis data. Semua tabel yang dibuat di basis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut multivalued pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan pengaksesannya dapat dipertanggungjawabkan atau tetap ada didalam perancangan kelas.

Dalam mendefenisikan metode yang ada didalam kelas perlu memperhatikan apa yang di sebut dengan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi didalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan

metode yang dibuat harus memiliki kadar *cohesion* yang kuat dan kadar *coupling* yang lemah.

Berikut adalah Tabel II.6. simbol – simbol yang ada pada diagram kelas :

Simbol	Deskripsi
Kelas 	Kelas pada struktur 46system.
Antarmuka / <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna <i>generalisasi</i> – spesialisasi (umum – khusus).
Ketergantungan / <i>dependency</i> 	Ketergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua bagian (<i>whole-part</i>).

Tabel II.6. Simbol-Simbol *Class Diagram*

Sumber : (Rosa A.S, M.Shalahuddin; 2013; 146-147).

II.6.1. Keuntungan dan Kelemahan *UML*.

Keuntungan

Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.

Taylor (1992) menyatakan bahwa membangun *software* menggunakan pendekatan teknologi objek memberikan beberapa keuntungan, antara lain: memungkinkan penggunaan kembali objek yang ada (*reusable*), memungkinkan *software* yang baru dengan konstruksi yang lebih besar, *software* berorientasi objek secara umum lebih mudah dimodifikasi dan dirawat karena sebuah objek dapat dimodifikasi tanpa banyak berpengaruh pada objek yang lain.

Kelemahan

UML dipandang masih mempunyai kekurangan terutama dalam *generate* kode program secara komplit. Hal ini karena kurangnya cara memodelkan aspek kelakuan *internal* perangkat lunak untuk dipetakan ke dalam kode program. (Haviluddin;2011;8-9).