

BAB II

TINJAUAN PUSTAKA

II.I. Sistem

Sistem merupakan sekumpulan komponen yang saling berhubungan, bekerja sama untuk mencapai suatu tujuan dengan menerima input serta menghasilkan output dalam proses transformasi yang teratur.

Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Sebagai contoh, sistem kekebalan tubuh untuk menghadapi penyakit cacar dan *diphtheri*, namun masih banyak pula berbagai sistem yang belum dapat diungkapkan dengan teknologi yang sekarang telah dimiliki oleh manusia, misalnya sistem kekebalan tubuh untuk menghadapi penyakit AIDS. Contoh sistem lain dari diri manusia adalah sistem pernafasan, yang berfungsi untuk menyediakan oksigen bagi tubuh dan untuk mengeluarkan zat asam arang yang merupakan sampah hasil pembakaran didalam tubuh.. sistem pernafasan terdiri dari unsur unsure yang membentuk struktur sistem pernafasan dan setiap unsur struktur sistem tersebut bekerja dengan suatu pola tertentu untuk memenuhi tujuan sistem. (Tata Sutabri : 2004 ; 7)

II.2. Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya. Nilai informasi berhubungan dengan keputusan. Bila tidak pilihan atau keputusan maka informasi tidak diperlukan. Keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks pengambilan keputusan. Nilai informasi ini didasarkan atas 10 (sepuluh) sifat, yaitu :

a. Mudah Diperoleh

Sifat ini menunjukkan kemudahan dan kecepatan untuk memperoleh informasi. Kecepatannya dapat diukur, misalnya 1 menit versus 24 jam. Akan tetapi berapa nilainya bagi pemakai informasi sulit untuk mengukurnya.

b. Luas dan Lengkap

Sifat ini menunjukkan kelengkapan isi informasi. Hal ini tidak hanya mengenai volumenya, akan tetapi juga mengenai keluaran informasinya. Sifat ini sangat kabur dan karena itu sulit untuk mengukurnya.

c. Ketelitian

Sifat ini berhubungan dengan tingkat kebebasan dari kesalahan keluaran informasi. Pada volume data yang besar biasanya terdapat dua jenis kesalahan pencatatan dan kesalahan perhitungan.

d. Kecocokan

Sifat ini menunjukkan seberapa baik keluaran informasi dalam hubungannya dengan permintaan para pemakai. Isi informasi harus ada hubungannya dengan masalah yang sedang dihadapi sedangkan semua keluaran yang lainnya tidak berguna. Sifat ini sulit mengukurnya.

e. Ketepatan Waktu

Sifat ini berhubungan dengan waktu yang dilalui, yang lebih pendek dari siklus untuk mendapatkan informasi. Masukan, pengolahan, dan pelaporan keluaran kepada para pemakai, biasanya tepat waktu. Dalam beberapa hal, ketepatan waktu dapat diukur. Misalnya berapa banyak penjualan ditingkatkan dengan menanggapi permintaan pelanggan mengenai ketersediaan barang-barang inventaris.

f. Kejelasan

Sifat ini menunjukkan tingkat kejelasan informasi. Informasi hendaknya terbebas dari istilah-istilah yang tidak jelas.

g. Keluwesan

Sifat ini berhubungan dengan apakah informasi tersebut dapat digunakan untuk membuat lebih dari satu keputusan, tetapi juga apakah dapat digunakan untuk lebih dari seorang pengambil keputusan. Sifat ini sulit mengukurnya, akan tetapi dalam beberapa hal dapat diukur dengan suatu nilai tertentu.

h. Dapat Dibuktikan

Sifat ini menunjukkan sejauh mana informasi itu dapat diuji oleh beberapa pemakai hingga sampai didapatkan kesimpulan yang sama.

i. Tidak Ada Prasangka

Sifat ini berhubungan dengan ada tidaknya keinginan untuk mengubah informasi tersebut guna mendapatkan kesimpulan yang telah diarahkan sebelumnya.

j. Dapat Diukur

Sifat ini menunjukkan hakikat informasi yang dihasilkan oleh sistem informasi formal. Meskipun kabar angin, desas-desus, dugaan-dugaan, dan lainnya sering dianggap sebagai informasi.

II.3. Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi perusahaan yang bersifat manajerial dengan kegiatan strategis dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. (Tata Sutabri : 2004 ; 36)

Dari penjelasan mengenai sistem informasi tersebut dapat disimpulkan bahwa sistem informasi merupakan suatu elemen yang saling berhubungan untuk menyediakan suatu informasi yang digunakan untuk pengambilan keputusan dalam manajemen operasional dan perusahaan dalam pengambilan keputusan serta memberikan data yang akurat, tepat waktu, dan dapat dipahami oleh orang

yang menerimanya. Adapun tujuan dari sistem informasi ini adalah sebagai berikut :

1. Meneyediakan informasi untuk membantu pengambilan keputusan manajemen.
2. Membantu petugas dalam melaksanakan tugas operasional perusahaan dari hari ke hari.
3. Menyediakan informasi yang layak bagi pemakai pihak luar perusahaan.

II.4. Perhitungan

Perhitungan merupakan kegiatan mengakumulasikan segala jenis biaya berdasarkan berbagai proses atau departemen dan membebankannya kesejumlah besar produk yang hamper sama.

Perhitungan biaya *storage & demorate* yang dilakukan di PT.Alamui Logistik – Medan masih menggunakan cara yang manual. Oleh karena itu dengan dirancangnya suatu media berbasis website dapat memudahkan pihak perusahaan dalam melakukan perhitungan biaya *storage & demorate* tersebut.

Dengan demikian penulis mencoba untuk membuat **Sistem Informasi Perhitungan Biaya Storage dan Demorate pada PT.Alamui Logistik – Medan** yang nantinya akan membantu pihak perusahaan.

II.5. Biaya

Biaya (*Cost*) merupakan kas atau nilai yang telah dikeluarkan oleh setiap perusahaan khususnya untuk memperoleh barang atau jasa yang diharapkan akan

membawa manfaat sekarang ataupun dimasa yang akan datang bagi setiap organisasi atau perusahaan.

II.6. PHP

II.6.1. Pengenalan PHP

PHP (*Hypertext Preprocessor*) adalah jenis *middleware open source*, sehingga dapat digunakan oleh siapa saja dengan cuma-cuma. Selain gratis, PHP memiliki berbagai kelebihan, di antaranya untuk membuat website yang dinamis. PHP akan dijalankan pada sisi server (*server side*) sehingga klien tidak bias melihat kode aslinya. (Dadan Sutisna : 2007 ; 40).

PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, PHP dapat membentuk suatu tampilan berdasarkan permintaan. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Cold Fussion*, ataupun *Perl*. Namun perlu diketahui bahwa PHP sebenarnya bias dipakai secara *Command Line*, artinya skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

II.7. Kelebihan PHP

Ada beberapa kelebihan yang dimiliki oleh PHP yaitu :

- a. Bahasa pemrograman PHP adalah sebuah bahasa skrip yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. *Web server* mendukung PHP dapat ditemukan dimana-mana dari mulai *apache*, *IIS*, hingga *web server* yang lain dengan konfigurasi yang relative mudah.

- c. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux, Unix, Windows*) dan dapat dijalankan secara *runtime* melalui *cencole*, serta juga dapat menjalankan perintah-perintah sistem.
- d. PHP mampu berkomunikasi dengan berbagai database yang terkenal sekalipun, meskipun dengan kelengkapan yang berbeda-beda, sehingga PHP sangat cocok untuk membangun halaman-halaman *web dinamis*.

II.7.1. Konsep Dasar PHP

PHP (*PHP Hypertext Preprocessor*) adalah bahasa scripting server-side bagi pemrograman web. Secara sederhana, PHP merupakan tool bagi pengembangan web dinamis. PHP sangat populer karena memiliki fungsi *built-in* lengkap, cepat, mudah dipelajari, dan bersifat gratis. Skrip PHP cukup disisipkan pada kode HTML agar dapat bekerja. PHP dapat berjalan di berbagai web server dan sistem operasi yang berbeda. (Angga Wibowo : 2007 ; 2).

Berikut ini adalah beberapa bentuk dan contoh penulisan untuk memasukan skrip PHP yaitu :

- Penulisan PHP dimulai dengan `<?` dan diakhiri dengan `?>` .

Contoh :

```
<?
```

```
Disini skrip PHP.
```

```
?>
```

- Setiap perintah pada PHP diakhiri dengan titik koma (;).

Contoh :

```
<?
Perintah 1;
Perintah 2;
Dst;
?>
```

- Variabel PHP ditulis dengan tanda dolar (\$) dan kemudian diikuti nama variabel tanpa spasi.

Contoh :

```
<?
$namavariabel = " Ini adalah variabel ";
$hitung = 2 + 5;
?>
```

II.7.2. Konsep Kerja PHP

Model kerja HTML diawali dengan permintaan satu web oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau lebih dikenal dengan sebutan alamat internet, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang diinginkan, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*.

Selanjutnya *web server* akan mencari *file* yang diminta dan diproses. Setelah diproses maka *web server* akan memberikan isinya ke *web browser* dan *browser*

yang mendapatkan isinya segera melakukan proses penerjemah kode HTML dan menampilkannya ke layar pemakai. (Sumber : Abdul Kadir : 2008 ; 4).

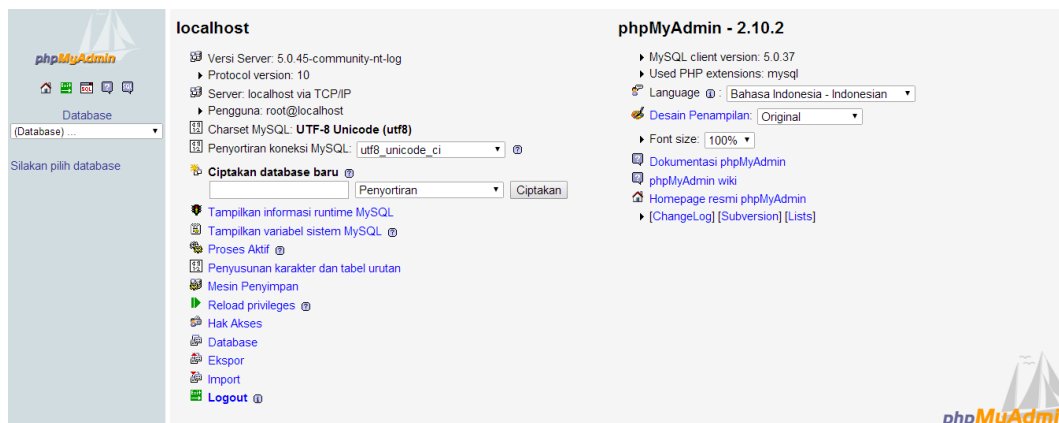
II.8. MySQL

II.8.1. Pengenalan MySQL

MySQL (My Structure Query Language) adalah sebuah program pembuatan database yang bersifat *open source*, artinya siapa saja boleh menggunakannya. *MySQL* sebenarnya produk yang berjalan pada *platform Linux*. Karena sifatnya yang *open source*, *MySQL* dapat dijalankan pada semua *platform* baik *Windows* maupun *Linux*. Selain itu *MySQL* juga merupakan program pengakses database yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *Multi User* (banyak pengguna).

Database digunakan untuk penyimpanan data. Demikian pula dengan *MySQL*. Kita akan memanggil data pada *MySQL* melalui *PHP*, kemudian hasilnya dikirim ke computer klien untuk ditampilkan pada browser. Data pada *MySQL* dapat dipanggil, dihapus, atau ditambah melalui query. (Dadan Sutisna : 2007 ; 45).

Dengan demikian dalam memanipulasi data penulis menggunakan *MySQL* sebagai database dalam pengolahan data :



Gambar II.2. MySQL diakses dengan phpMyAdmin

(Sumber : Dadan Sutisna : 2007 ; 58)

II.9. Entity Relationship Diagram (ERD)

Merupakan suatu model untuk menjelaskan hubungan antar dua dalam basis data berdasarkan suatu persepsi bahwa *real word* terdiri dari *object-object* dasar yang mempunyai hubungan atau antar *object-object* tersebut. Relasi antar *object* dengan menggunakan simbol-simbol grafis tertentu. Model *entity relationship* adalah suatu penyajian dengan menggunakan *entitiy* dan *relationship*. Diperkenalkan pada tahun 1976 oleh P.P. Chen.

II.9.1. Komponen-komponen yang terdapat didalam *Entity Relationship*

Model.

1. *Entity*

- a. Adalah sesuatu yang dapat dibedakan dalam dunia nyata dimana informasi yang berkaitan dengannya dikumpulkan.

- b. *Entity* set adalah kumpulan *entity* yang sejenis.
- c. Symbol yang digunakan untuk *entity* adalah persegi panjang.
- d. *Entity* set dapat berupa :
 - 1) *Entity* yang bersifat fisik, yaitu *entity* yang dapat dilihat.
 Contohnya : rumah, kendaraan, mahasiswa, dosen, dan lain-lain.
 - 2) *Entity* yang bersifat konsep atau logic, yaitu *entity* yang tidak dapat dilihat.
 Contohnya : pekerjaan, perusahaan, rencana, mata kuliah, dan lain-lain.
- e. Simbol yang digunakan *entity* adalah persegi panjang.

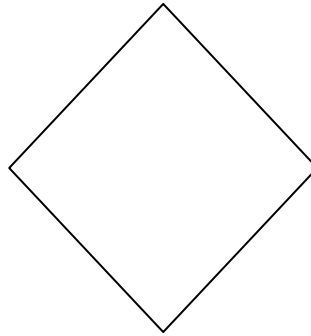


Gambar : II.2. *Entity*

(Sumber : Linda Marlinda, S.Kom : 2004 ; 17)

2. *Relationship*

- a. Adalah hubungan yang terjadi antara satu atau lebih *entity*.
- b. *Relationship* tidak mempunyai keberadaan fisik, kecuali yang mewarisi hubungan antara *entity* tersebut.
- c. *Relationship* set adalah kumpulan *relationship* yang sejenis.
- d. Simbol yang digunakan adalah bentuk belah ketupat, *diamond* atau *rectangle*.

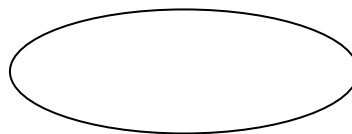


Gambar : II.3. Relationship

(Sumber : Linda Marlinda, S.Kom : 2004 ; 17)

3. Attribute

- a. Adalah karakteristik dari *entity* atau *relationship* yang menyediakan penjelasan detail tentang *entity* atau *relationship* tersebut.
- b. Attribute value (nilai attribute) adalah suatu data actual atau informasi yang disimpan di suatu attribute didalam suatu *entity* atau *relationship*
- c. Terdapat dua jenis attribute, yaitu :
 1. *Inditifier (key)*, untuk menentukan suatu *entity secara* unik..
 2. *Descriptor (nonkey attribute)*, untuk menentukan karakteristik dari suatu *entity* yang tidak unik.
 3. Simbol yang digunakan adalah bentuk oval.



Gambar : II.4. Attribute

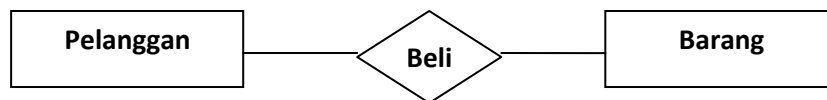
(Sumber : Linda Marlinda, S.Kom : 2004 ; 18)

4. Indicator Tipe

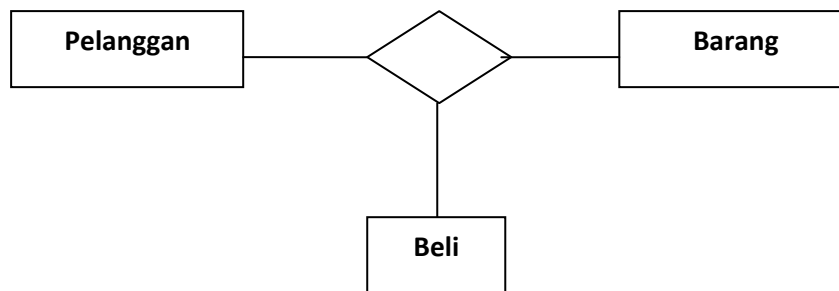
a. *Indicator tipe associative object*

Berfungsi sebagai suatu objek dan suatu *relationship*.

Contoh :



Menjadi :



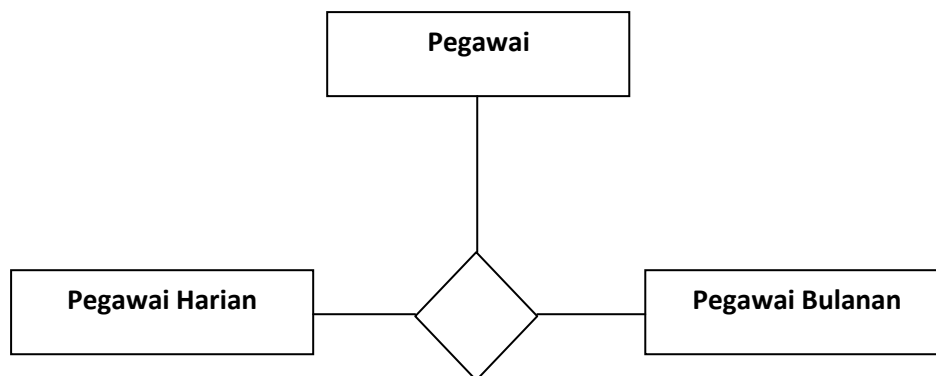
Gambar : II.5. Indicator Tipe

(Sumber : Linda Marlinda, S.Kom : 2004 ; 19)

b. *Indicator tipe supertipe.*

Terdiri dari suatu object dan sub kategori atau lebih yang dihubungkan dengan *relationship* yang tidak bernama. (Sumber : Linda Marlinda, S.Kom : 2004 ; 19)

Contoh :



Gambar : II.6. *Indicator Tipe Super Tipe*
(Sumber : Linda Marlinda, S.Kom : 2004 ; 19)

II.10. Normalisasi

Normalisasi adalah proses pengelompokkan elemen data menjadi table-tabel yang menunjukkan entitas dan relasinya.

Ada beberapa macam kunci (*key function*) yang digunakan untuk proses pencarian, penyaringan, penghapusan, dan lainnya yaitu sebagai berikut :

1. *Candidate Key* (Kunci Calon)

Kunci kandidat adalah satu atribut atau satu set minimal atribut yang mengidentifikasi secara unik suatu kejadian yang spesifik dari suatu entitas.

2. *Primary Key* (Kunci Primer)

Kunci primer adalah satu atribut atau satu set minimal atribut yang tidak hanya mengidentifikasi secara unik suatu kejadian yang spesifik, akan tetapi juga dapat mewakili setiap kejadian dari suatu entitas. Setiap kunci

kandidat punya peluang menjadi *primary key*, akan tetapi sebaiknya dipilih satu saja yang dapat mewakili entitas yang ada secara menyeluruh.

3. *Alternate Key* (Kunci Alternatif)

Kunci alternative adalah kunci kandidat yang tidak dipakai sebagai *primary key*. Kerap kali kunci alternative ini dipakai sebagai kunci pengurutan dalam pembuatan laporan.

4. *Foreign Key* (Kunci Tamu)

Kunci tamu adalah satu atribut atau set atribut yang melengkapi satu relationship (hubungan) yang menunjukkan ke induknya.

Pada proses normalisasi ini perlu dikenal dahulu definisi dari tahap normalisasi, yaitu sebagai berikut :

1. Bentuk Tidak Normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan untuk mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

2. Bentuk Normal Kesatu (*1NF. First Normal Form*)

Bentuk normal kesatu mempunyai cirri yaitu bahwa setiap data dibentuk dalam *flat file* (file datar/rata), data dibentuk dalam satu *record* demi *record* dan nilai dari *field* berupa *atomic value*. Tidak ada set atribut yang berulang atau atribut bernilai ganda (*multivalue*).

3. Bentuk Normal Kedua (2NF/Second Normal Form)

Bentuk normal kedua mempunyai syarat, yaitu bentuk data telah memenuhi criteria bentuk normal kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama/ *primary key*. Dengan demikian untuk membentuk normal kedua haruslah sudah ditentukan kunci *field*, kunci *field* haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

4. Bentuk Normal Ketiga (3NF/Third Normal Form)

Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* secara menyeluruh.

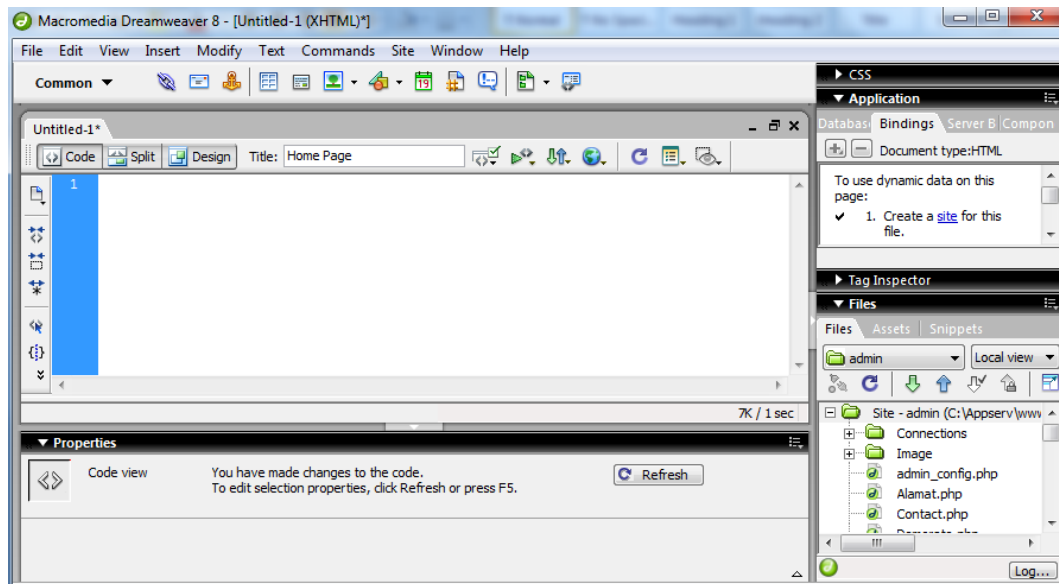
5. Boyce-Codd Normal Form (BCNF)

Boyce-Codd Normal Form mempunyai paksaan yang lebih kuat dari bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribut *superkey* (Tata Sutabri : 2004 : 202 – 208).

II.10.1. Macromedia Dreamweaver

Macromedia Dreamweaver adalah sebuah software *web design* yang menawarkan cara mendesain web dengan dua langkah sekaligus dalam satu waktu, yaitu mendesain dan memprogram. Dreamweaver memiliki satu jendela mini yang disebut *HTML source* tempat kode-kode HTML tertulis. Setiap kali kita mendesain website seperti menulis kata-kata, meletakkan gambar, membuat

table, dan proses lainnya, tag-tag HTML yang akan membangun website itu. (Gregorius Agung : 2001 ; 9)



Gambar : II.7. Tampilan Macromedia Dreamweaver 8

(Sumber : Dadan Sutisna : 2007 ; 52)

II.10.2. Aplikasi Web

Pada perkembangan teknologi web saat ini, penggunaan teknik pemrograman semakin kompleks. Sebuah situs tidak cukup hanya menghadirkan interaktifitas. Situs web secara lebih dinamis juga harus mampu menangani database dan memperhatikan faktor keamanan. Teknologi *server side* seperti ASP dan PHP hadir hadir untuk menangani hal-hal tersebut. Namun, pada pengimplementasiannya, pemrograman berbasis *web* masih dirasa cukup merepotkan bagi sebagian pengembang dan pengelola situs *web*. Hal ini mengingat banyak faktor yang harus dipersiapkan agar sebuah situs *web* dapat terpublikasikan dan berdaya guna.

Para pengelola situs kini dapat berlega, mengingat saat ini banyak komunitas aplikasi web instan diinternet. Aplikasi-aplikasi *web* tersedia secara lengkap untuk berbagai kebutuhan. Sebagian penyedia menawarkan aplikasi-aplikasi ini dalam bentuk aplikasi shareware, yang dapat digunakan dengan beberapa pembatasan. Namun, beberapa penyedia menawarkan hasil pengembangannya secara gratis. Umumnya untuk kategori kedua, penyedia hanya mensyaratkan penyertaan *link* kesitus mereka sebagai imbal-balik pemanfaatan aplikasi gratis tersebut. (Angga Wibowo : 2006 ; 2 – 3).

II.10.3. Web Browser

Web Browser merupakan alat terpenting untuk melihat hasil dari situs *web* yang anda bangun. *Web browser* juga dapat menampilkan kekurangan atau kesalahan yang mungkin terjadi dalam pengembangan situs *web* dengan aplikasi-aplikasi PHP. Dengan menggunakan beberapa *web browser* yang berbeda sebagai penguji tampilan *web*, konsistensi tampilan web dapat lebih terjamin. (Angga Wibowo : 2006 ; 5).

II.10.4. Web Server

Web server adalah komputer yang digunakan untuk menyimpan dokumen-dokumen *web*, komputer ini akan melayani permintaan dokumen *web* dari kliennya. Contohnya adalah sebagai berikut :

1. Apache, merupakan *web server* yang digunakan diberbagai *platform Operating System* (OS), seperti *linux*, *windows*, dan lain-lain.
2. IIS (*Internet Information Service*), digunakan di sistem operasi *Windows NI* dan *Windows 2002*.

3. PWS (*Personal Web Server*), digunakan di sistem operasi *Windows 9x*.

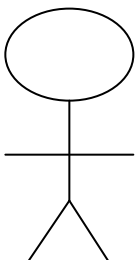
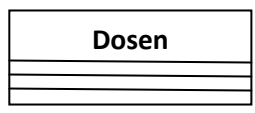
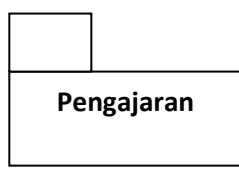
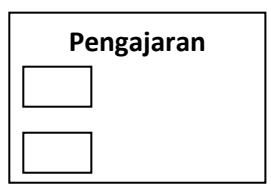
(Sumber : Abdul Kadir : 2008 ; 2)

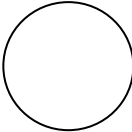
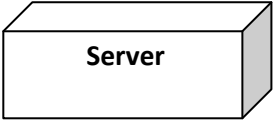
II.11. UML (*Unified ModelingLanguage*)

UML (*Unified Modeling Language*) merupakan bahasa pemodelan untuk sistem atau komponen-komponen perangkat lunak yang berparadigma berorientasi objek. Pemodelan (*Modeling*) digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga mudah dipelajari dan dipahami.

(Sumber : Djon Irwanto, S.Kom, MM : 2006 ; 2)

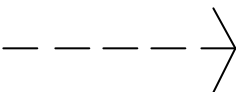
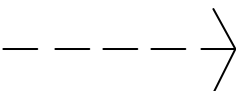
Tabel II.I. Beberapa Pengklasifikasian (*Classifier*)

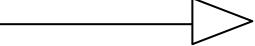
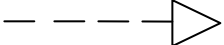
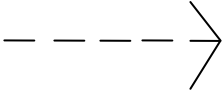
Pengklasifikasian	Kegunaan	Notasi
Actor	Menggunakan semua objek diluar sistem (bukan hanya pengguna sistem/perangkat lunak) yang berinteraksi dengan sistem yang digunakan.	
Use Case	Menggambarkan fungsionalitas yang dimiliki sistem.	
Kelas (<i>Class</i>)	Menggambarkan konsep dasar pemodelan sistem.	
Subsistem (<i>Subsystem</i>)	Menggambarkan paket spesifikasi serta implementasi.	
Komponen (<i>Component</i>)	Menggambarkan bagian-bagian fisik sistem/perangkat lunak yang dikembangkan.	

Antar Muka (<i>Interface</i>)	Menggambarkan antar muka pengiriman pesan (<i>Message</i>)	 Antar muka pengajaran mahasiswa
Simpul (<i>Node</i>)	Menggambarkan sumber daya komputasional yang digunakan oleh sistem.	

(Sumber : Djon Irwanto : 2006 ; 23 – 45)

Tabel II.2. Relasi - Relasi dalam UML

Relasi	Fungsi	Notasi
Assosiaisi (<i>Assosiation</i>)	Mendeskripsikan hubungan antar instance suatu kelas.	
Ketergantungan (<i>Dependency</i>)	Relasi antar dua elemen model	
Aliran (<i>Flow</i>)	Relasi antar dua versi suatu objek.	

Generalisasi (<i>Generalization</i>)	Relasi antara pengklasifikasian memiliki deskripsi yang bersifat lebih umum dengan berbagai pengklasifikasian yang lebih spesifik	
Realisasi (<i>Realization</i>)	Relasi antara spesifikasi dan implementasinya.	
Penggunaan (<i>Usage</i>)	Situasi dimana salah satu elemen yang lainnya agar dapat berfungsi dengan baik.	

(Sumber : Djon Irwanto : 2006 ; 126)

II.11.1. Diagram-Diagram UML

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, diantaranya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis dan dinamis. Jenis diagram itu antara lain :

1. Diagram Kelas (*Class Diagram*) bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umumnya dijumpain pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, seringpula diagram kelas memuat kelas-kelas.
2. Diagram Paket (*Package Diagram*) bersifat statis. Diagram yang memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.
3. Diagram *Use Case* bersifat statis. Diagram ini memperlihatkan komponen *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram Interaksi Dan *Sequence* (urutan) bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.
5. Diagram Komunikasi (*Communication Diagram*) bersifat dinamis. Diagram ini sebagai ganti diagram kolaborasi *UML* yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram*) bersifat dinamis. Diagram status ini memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi kejadian serta aktifitas. Diagram ini sangat penting

terutama untuk memperlihatkan sifat dinamis antarmuka (*interface*), kelas, kolaborasi, dan penting untuk pemodelan sistem-sistem yang reaktif.

7. Aktivitas Diagram (*Aktivity Diagram*) bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini penting untuk pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali suatu objek.
8. Diagram Komponen (*Component Diagram*) bersifat statis. Diagram ini memperlihatkan organisasi serta ketergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka, serta kolaborasi-kolaborasi.
9. Diagram *Deployment* (*Deployment Diagram*) bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (run time). Memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*). (Sumber : Prabowo Pudji Widodo, Herlawati : 2011 ; 10-12).