

## BAB II

### LANDASAN TEORI

#### 2.1 Pengertian Sistem

Sistem adalah suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang saling berinteraksi untuk mencapai tujuan. (Jogiyanto, 2000)

Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu yaitu: (Hanif Al Fatta, 2007)

a. Batasan (*boundary*)

Pengembangan dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan unsur mana yang termasuk di luar sistem.

b. Lingkungan (*environment*)

Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan *input* terhadap suatu sistem.

c. Masukan (*input*)

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

d. Keluaran (*output*)

Sumber daya atau produk (informasi, laporan, dokumen, tampilan layer komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.

e. *Komponen (component)*

Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.

f. *Penghubung (interface)*

Tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.

g. *Penyimpanan (storage)*

Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

h. *Sasaran sistem*

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objektive*). Jika sistem tidak mempunyai tujuan, maka operasi sistem tidak ada gunanya. Sasaran sistem sangat menentukan masukan yang dibutuhkan oleh sistem. Sistem dikatakan berhasil jika mengenai sasaran (tujuan sistem).

## 2.2 Pengertian Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. (Jogiyanto, 1989) Informasi

yang berkualitas memiliki 3 (tiga) kriteria, yaitu : (Kusrini dan Andri koniyo, 2007)

a. Akurat (*accurate*)

Informasi harus bebas dari kesalahan, tidak bias ataupun menyesatkan. Akurat juga berarti bahwa informasi itu harus dapat dengan jelas mencerminkan maksudnya.

b. Tepat pada waktunya (*timeliness*)

Informasi yang datang pada penerima tidak boleh terlambat. Di dalam pengambilan keputusan, informasi yang sudah usang tidak lagi bernilai. Bila informasi datang terlambat sehingga pengambilan keputusan terlambat dilakukan, hal itu dapat berakibat fatal bagi perusahaan.

c. Relevan (*relevance*)

Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas dengan informasi tersebut. Informasi harus bermanfaat bagi pemakainya.

### 2.3 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (Jogiyanto. 1989)

Komponen-komponen sistem informasi berbasis komputer dalam suatu organisasi terdiri dari komponen-komponen berikut : (Hanif Al Fatta, 2007)

a. Perangkat keras

Adalah perangkat keras komponen untuk melengkapi kegiatan masukkan data, memproses data, dan keluaran data.

b. Perangkat lunak

Adalah program dan instruksi yang diberikan ke komputer.

c. *Database*

yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna informasi.

d. Telekomunikasi

yaitu komunikasi yang menghubungkan antara pengguna sistem dengan sistem komputer secara bersama-sama ke dalam suatu jaringan kerja yang efektif.

e. Manusia

*Personal* dari sistem informasi, meliputi manajer, analisis, programmer, dan operator, serta bertanggung jawab terhadap perawatan sistem.

Sistem informasi yang terdiri dari komponen-komponen diatas disebut dengan istilah blok bangunan (*building block*) yang terdiri dari: (Hanif Al Fatta, 2007)

a. Blok masukan (*input block*)

*input* mewakili data yang masuk ke dalam sistem informasi termasuk metode- metode dan media untuk menangkap data yang akan dimasukkan yang dapat berupa dokumen-dokumen dasar.

b. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan

c. Blok keluaran(*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen serta semua pemakai sistem.

d. Blok teknologi (*technology block*)

Teknologi merupakan kotak alat (*tool-box*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan sekaligus mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

e. Blok *database* (*database block*)

*Database* merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

f. Blok kendali (*control block*)

Pengendalian perlu dirancang dan ditetapkan untuk menyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah, ataupun bila terlanjur terjadi kesalahan dapat langsung diatasi dengan cepat.

## 2.4 Pengertian Ujian

Dalam konteks akademis atau profesional, ujian adalah tes yang bertujuan untuk menentukan kemampuan seseorang. Biasanya ujian tes tertulis, walaupun beberapa mungkin praktis atau komponen praktis, dan sangat bervariasi dalam struktur, isi dan kesulitan tergantung pada subjek, kelompok usia orang yang diuji dan profesi. Pemeriksaan *kompetitif* adalah ujian di mana pelamar bersaing untuk sejumlah posisi, sebagai lawan hanya harus mencapai tingkat tertentu untuk lulus.

”UJIAN adalah kegiatan untuk mengetahui seberapa tinggi tingkat keberhasilan siswa.” ( Arikunto, 2007)

## 2.5 Pengertian Online

*Online* adalah terhubung, terkoneksi, aktif dan siap untuk operasi, dapat berkomunikasi dengan atau dikontrol oleh komputer. *Online* ini juga bisa diartikan sebagai suatu keadaan dimana sebuah *device* (komputer) terhubung dengan *device* lain, biasanya melalui modem.

*Online* adalah sedang menggunakan jaringan, terhubung dalam jaringan, satu perangkat dengan perangkat lainnya yang terhubung sehingga bisa saling berkomunikasi.

Berdasarkan teori *online* diatas dapat disimpulkan bahwa *online* adalah suatu keadaan dimana sebuah *device* (komputer) terhubung dengan *device* lain dengan menggunakan perangkat modem, sehingga biasa saling berkomunikasi. (Sudiana, 2000)

## 2.6 Ujian Online

Ujian *online* sudah tidak lagi menggunakan media kertas atau alat tulis sebagai ujian. Sistem ujian ini dibangun secara komputerisasi, dimana peserta uji langsung mendapat dan menjawab soal ujian melalui komputer. Pemeriksaan ujian dilakukan langsung oleh sistem, dan peserta akan mendapatkan laporan hasil ujian secara langsung. Ujian ini dipakai seperti pada sertifikasi *MICROSOFT, TryOut*, dan lain-lain (Darmawan, 2007).

## 2.7 Website

*Website* atau situs juga dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

Bersifat statis apabila isi informasi *Website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *Website*. Bersifat dinamis apabila isi informasi *Website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *Website*. Contoh *Website* statis adalah berisi profil perusahaan, sedangkan *Website* dinamis adalah seperti Friendster, Multiply, dll. Dalam sisi pengembangannya, *Website* statis hanya bisa diupdate oleh pemiliknya saja, sedangkan *Website* dinamis bisa diupdate oleh pengguna maupun pemilik.

## 2.8 Perancangan Sistem

Analisis perancangan sistem adalah suatu pendekatan yang sistematis untuk mengidentifikasi masalah, peluang, dan tujuan-tujuan, menganalisis arus informasi dalam organisasi, serta untuk merancang sistem informasi terkomputerisasi untuk menyelesaikan suatu masalah. Saat informasi berkembang, suatu pendekatan yang sistematis dan terencana untuk memperkenalkan, memodifikasi, dan pemeliharaan sistem informasi menjadi sangat penting. Analisis dan perancangan sistem menampilkan pendekatan semacam itu. (Kusworo, 2010)

### 2.8.1 *Unified Modelling Language (UML)*

(Sukamto dan Shalahuddin 2013), “*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”

*UML* menyediakan serangkaian gambar dan *Diagram* yang sangat baik. Beberapa *Diagram* memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna



Gambar 2.1 Tampilan Logo UML

Sumber: (Sukamto dan Shalahuddin, 2013)



Menurut (Widodo dan Herlawati, 2011), *UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

- a. Merancang perangkat lunak.
- b. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
- c. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
- d. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Tabel 2.1. Tipe *Diagram UML*

No.	Diagram	Tujuan
1	<i>Class</i>	Memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi
2	<i>Package</i>	Memperlihatkan kumpulan kelas-kelas, merupakan dari <i>Diagram</i> komponen
3	<i>Use case</i>	<i>Diagram</i> ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas)
4	<i>Sequence</i>	<i>Diagram</i> interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu
5	<i>Communication</i>	Sebagai pengganti <i>Diagram</i> kolaborasi <i>UML</i> 1.4 yang menekankan organisasi struktural dari obyek-obyek yang menerima serta mengirim pesan
6	<i>Statechart</i>	<i>Diagram</i> status memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktivitas
7	<i>Activity</i>	Tipe khusus dari <i>Diagram</i> status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem
8	<i>Component</i>	Memperlihatkan organisasi serta kebergantungan sistem / perangkat lunak pada komponen-komponen yang telah ada sebelumnya
9	<i>Deployment</i>	Memperlihatkan konfigurasi saat aplikasi dijalankan ( <i>run-time</i> )






Sumber : ( Widodo dan Herlawati, 2011)






### 2.8.2 Use Case Diagram

Menurut (Sukanto dan Shalahuddin, 2013), “*use case* 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”.

Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.2. Simbol-simbol Use Case Diagram

NO	Simbol	Nama	Deskripsi
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		Generalization	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
5		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.

NO	Simbol	Nama	Deskripsi
6		Association	Apa yang menghubungkan antara objek satu dengan objeklainnya.
7		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		UseCase	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : (Sukamto dan Shalahuddin, 2013)

Komponen pembentuk *Diagram use case* adalah :

1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2. *Use case*, aktivitas / sarana yang disiapkan oleh bisnis / sistem.
3. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.

Setiap *use case* dilengkapi dengan skenario. Skenario *use case* adalah alur jalannya proses *use case* dari sisi aktor dan sistem. Berikut adalah format table skenario *use case*:

Tabel 2.3. Format Tabel Skenario *Use Case*

AKSI AKTOR	REAKSI SISTEM
Skenario Normal	
Skenario Alternatif	

Sumber : (Sukamto dan Shalahuddin , 2013)


Skenario *use case* dibuat per *use case* terkecil, misalkan generalisasi maka skenario yang dibuat adalah *use case* yang lebih khusus. Skenario normal adalah skenario bila sistem berjalan normal tanpa terjadi kesalahan atau *error*. Sedangkan skenario alternatif adalah skenario bila sistem tidak berjalan normal atau mengalami *error*.


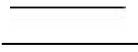
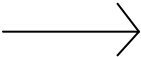


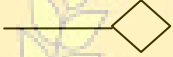
### 2.8.3 *Class Diagram*

Menurut (Sukamto dan Shalahuddin 2013), “*class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

*Diagram* kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam *Diagram* kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Adapun simbol-simbol yang digunakan dalam *class Diagram* adalah sebagai berikut:

Tabel 2.4. Simbol-simbol *Class Diagram*

No	Gambar	Nama	Deskripsi
1		<i>Class</i>	Kelas pada stuktur sistem

No	Gambar	Nama	Deskripsi
2		<i>interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek
3		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
4		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
7		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (whole-part)

Sumber : ( Sukamto dan Shalahuddin, 2013 )







#### 2.8.4 *Activity Diagram*

(Sukamto dan Shalahuddin, 2013) "*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.

Adapun simbol-simbol yang digunakan dalam *activity Diagram* adalah sebagai berikut:

Tabel 2.5. Simbol-simbol *Activity Diagram*

No	Simbol	Nama	Deskripsi
1		Status awal	Status awal aktivitas sistem, sebuah <i>Diagram</i> aktivitas memiliki sebuah status awal
2		Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5		Status akhir	Status akhir yang dilakukan sebuah sistem, sebuah <i>Diagram</i> aktivitas memiliki sebuah status akhir
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi







Sumber : ( Sukamto dan Shalahuddin, 2013 )

### 2.8.5 *Sequence Diagram*

(Sukamto dan Shalahuddin, 2013), “*Diagram* sekuen menggambarkan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

*Sequence Diagram* menunjukkan urutan *event* kejadian dalam suatu waktu. Komponen *Sequence Diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama *message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal. Simbol-simbol yang digunakan dalam *Sequence Diagram* adalah:

Tabel 2.6. Simbol-simbol *Sequence Diagram*

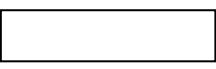
No	Simbol	Nama	Keterangan
1		<i>An Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem
2		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah penggambaran dari <i>form</i>
4		<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel
5		<i>A focus of control</i>	Menggambarkan tempat mulai dan berakhirnya sebuah <i>message</i> (pesan)
6		<i>A line of life</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

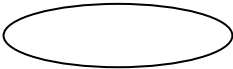
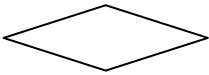

Sumber : ( Sukamto dan Shalahuddin, 2013 )

### 2.8.6 *Entity Relationship Diagram*

*Entity Relationship Diagram* adalah *Diagram* yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta yang ditinjau. (Kusworo, 2010)

Tabel 2.7 *Entity Relationship Diagram*

No	Simbol	Keterangan
1		Persegi panjang mewakili entitas (individu yang mewakili sesuatu yang nyata dan dapat dibedakan

No	Simbol	Keterangan
		dari sesuatu yang lain).
2		Elips mewakili atribut ( property dari entitas)
3		Belah ketupat mewakili relasi (hubungan diantara sejUMLah entitas yang berasal dari himpunan entitas yang berbeda)
4		Garis sebagai penghubung antara relasi dengan entitas dan entitas dengan atributnya

Sumber: (Kusworo, 2010)

### 2.8.7 Tabel *Relationship*

Kerelasian menyatakan hubungan antar relasi dalam basis data. Kerelasian antar relasi dituliskan oleh *foreign key* atau relasi-relasi bertipe transaksi yang digunakan dalam basis data. Jenis-jenis antar relasi meliputi : (Kusworo, 2010) Kerelasian satu ke satu (*one to one relationship*)

- a. Kerelasian satu ke satu dapat terjadi jika setiap nilai pada suatu relasi hanyamengimplementasikan sebuah nilai pada relasi lain yang direalisasikan secara logik.
- b. Kerelasian satu ke banyak (*one to many relationship*)

Kerelasian satu ke banyak dapat terjadi jika setiap nilai pada suaturelasi mengimplementasikan banyak nilai pada relasi lain yang direalisasikan secara logik.



c. Kerelasiaan banyak ke satu (*many to one relationship*)

Kerelasiaan banyak ke satu dapat terjadi jika banyak nilai pada suatu relasi mengimplementasikan satu nilai pada relasi lain yang direalisasikan secara logik.

### 2.8.8 Database

*Database* adalah kumpulan dari data yang saling berhubungan satu sama lain yang diorganisasikan berdasarkan sebuah skema tertentu, tersimpan di *hardware* dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu. (Irmansyah, 2003).

Hubungan antar data dapat ditunjukkan dengan adanya *field/column* kunci dari tiap *file/table* yang ada. Dalam satu *file* atau *table* terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* (umumnya digambarkan sebagai baris data) terdiri dari *field* yang saling berhubungan menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan disimpan dalam satu *record*. *Database* sangat diperlukan dalam sebuah sistem karena data-data yang terdapat dalam *database* inilah yang nantinya akan diberikan *security* sehingga data-data penting dalam sebuah sistem tidak dapat dicuri dengan mudah. (Kusworo, 2010).

### 2.9 Software Pendukung

*Software* pendukung digunakan sebagai alat untuk membantu penulis dalam proses pembuatan sistem ujian *online*. *Software* yang digunakan penulis diantaranya yaitu :

### 2.9.1 *PHP*

*PHP* singkatan dari *page hypertext preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen *HTML*. *PHP* diciptakan pertama kali oleh Rasmus Lerdorf pada tahun 1994. Awalnya, *PHP* digunakan untuk mencatat jumlah serta untuk

mengetahui siapa saja pengunjung pada homepage. Rasmus Lerdorf adalah salah seorang pendukung *open source*. Oleh karena itu, mengeluarkan *personal home page tool* versi 1.0 secara gratis, kemudian menambah kemampuan *PHP* 1.0 dan meluncurkan *PHP* 2.0. pada tahun 1996, *PHP* telah banyak digunakan dalam *Website* di dunia. Sebuah kelompok pengembangan *software* yang berdiri dari Rasmus, Zeev Suraski, Andi Gutman, Stig Bakken, Shane Caraveo, dan Jim Winstead bekerja sama untuk menyempurnakan *PHP* 2.0. Akhirnya, pada tahun 1998, *PHP* 3.0 diluncurkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 dikeluarkan *PHP* 4.0. tidak berhenti sampai di situ, kemampuan terus ditambah, dan dikeluarkan versi terbaru adalah *PHP* 5.0.x. (Peranginangin,2006)

*PHP* memiliki banyak kelebihan yang tidak dimiliki oleh bahasa *script* sejenis. *PHP* difokuskan pada pembuatan *script server-side*, yang bisa melakukan apa saja yang dapat dilakukan oleh CGI, seperti mengumpulkan data dari *form* menghasilkan isi halaman *web* dinamis, dan kemampuan mengirim serta menerima *cookies*, bahkan lebih daripada kemampuan CGI.

*PHP* dapat digunakan pada semua sistem operasi, antara lain Linux, Unix (termasuk variasinya HP-UX, Solaris, dan OpenBSD), Microsoft Windows, Mac OS X, RISC OS. *PHP* juga mendukung banyak *web server*, seperti *Apache*, *Microsoft Internet Information Server* (MMIS), *Personal Web Server* (PWS), *Netscape*, dan masih banyak lagi lainnya, bahkan *PHP* dapat bekerja sebagai suatu CGI processor. *PHP* tidak terbatas pada hasil keluaran *HTML* (*HyperText Markup Languages*). *PHP* juga memiliki kemampuan untuk mengolah keluaran gambar, file *PDF*, dan movies *Flash*. *PHP* juga dapat menghasilkan teks seperti *XHTML* dan file *XML* lainnya. Salah satu fitur yang dapat dikendalikan oleh *PHP* adalah dukungannya terhadap banyak *database*. *Database* yang dapat didukung oleh *PHP* antara lain *FrontBase*, *MSQL*, *MYSQL* dan *PostgreSQL*. (Peranginangin, 2006)

Sintak Program/Script *PHP* ditulis dalam apitan tanda khusus *PHP*. Ada empat macam pasangan tag *PHP* yang dapat digunakan untuk menandai blok script *PHP*: (Peranginangin, 2006)

- a. `<?PHP.....?>`
- b. `<script language = "PHP">.....</script>`
- c. `<?.....?>`
- d. `<%.....%>`

## 2.9.2 CSS

*CSS* mendefinisikan karakteristik tampilan (warna, *style*, dan posisi) suatu elemen pada dokumen *HTML* dalam bentuk property elemen tersebut. Pemisahan isi dengan tampilan yang dilakukan dengan penerapan *CSS* ini, memberikan kemungkinan penyusunan struktur suatu halaman *HTML* dengan lebih fleksibel. *CSS* merupakan bagian/*subset* dari *DOM* yang sangat berperan dalam *HTML*.

Perbedaan Implementasi antara *Netscape* dan *Microsoft* sehingga akhirnya *W3C* melakukan suatu standardisasi CSS. Pada awalnya, upaya standardisasi CSS dibagi dua, yaitu CSS1 untuk pengaturan warna dan *style*, serta CSS-P untuk pengaturan posisi suatu elemen. CSS2 dari *W3C* selanjutnya menggabungkan kedua standar ini menjadi suatu standar yang meliputi semua *setting* tampilan suatu elemen halaman *HTML* (Agus, 2011)

### 2.9.3 *HTML*

menurut (Suyanto , 2007) ”*HTML* itu adalah bahasa yang digunakan untuk menulis halaman web, biasanya menggunakan ekstensi *.htm*, *.HTML* atau *.sHTML*”.

Dalam penamaan sebuah dokumen yang akan ditampilkan pada *web browser* maka nama yang digunakan harus diakhiri dengan ekstensi (*.HTML.*) atau (*.HTML.*). Ekstensi dokumen *HTML* awalnya 3 karakter, adalah untuk mengakomodasi sistem penamaan dalam *DOS*.

Pengaturan warna di halaman *HTML* menggunakan mode kombinasi RGB (*Red, Green, Blue*) dimana setiap warna ditampilkan dalam dua digit nilai heksadesimal (0, 1, 2,...F). Setiap bagian dua digit kode menunjukkan banyaknya intensitas dari kombinasi warna merah , Hijau, dan Biru.

*HTML* saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (*W3C*). *HTML* dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa).

#### 2.9.4 *Java Script*

Menurut (Sunyoto, 2007) adalah bahasa scripting yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode *JavaScript* dapat disisipkan dalam halaman web menggunakan tag `SCRIPT`. *JavaScript* pertama kali dikembangkan oleh Brendan Eich dari Netscape dibawah nama Mocha, yang nantinya namanya diganti menjadi LiveScript, dan akhirnya menjadi *JavaScript*. Navigator sebelumnya telah mendukung Java untuk lebih bisa dimanfaatkan para programmer yang non-Java. Maka dikembangkanlah bahasa pemrograman bernama LiveScript untuk mengakomodasi hal tersebut. Bahasa pemrograman inilah yang akhirnya berkembang dan diberi nama *JavaScript*, walaupun tidak ada hubungan bahasa antara Java dengan *JavaScript*. *JavaScript* bisa digunakan untuk banyak tujuan, misalnya untuk membuat efek rollover baik di gambar maupun teks, dan yang penting juga adalah untuk membuat *AJAX*. *JavaScript* adalah bahasa yang digunakan untuk *AJAX*.

#### 2.9.5 *MYSQL*

Menurut (Junaedi, 2005) "*MYSQL* adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya. *MYSQL* didistribusikan secara *open source* dan gratis mulai tahun 1996, tetapi mempunyai sejarah pengembangan sejak tahun 1979. Database *MYSQL* adalah database yang sangat *powerfull*, stabil, mudah. *MYSQL* sangat banyak dipakai dalam sistem database web dengan menggunakan *PHP*. *PHP*Triad juga

memberikan fasilitas *database* yang lengkap. Karena *PHPTriad* dilengkapi dengan *database MYSQL* maka terdapat tempat untuk menyimpan data (*store*), dan untuk mengambil kembali data anda (*retrieve*). Seperti sistem *database SQL (Structured Query Language)* yang lain, *MYSQL* juga dilengkapi dengan perintah-perintah dan sintaks-sintaks *SQL*, dengan keunggulan sebagai berikut :

- a. Konsep *database MYSQL* berkecepatan tinggi tentang sistem penyajiandata.
- b. Harga yang relatif murah, karena ada yang dapat diperoleh secara gratis.
- c. Sintaks bahasanya menggunakan perintah yang sederhana.
- d. Dapat bekerja dalam beberapa *system* operasi seperti *Windows, Linux, MacOS, Unix (Solaris, AIX, dan DEC Unix), FreeBSD, OS/2, Irix.* (Swastika,2006).

*MYSQL* menerima berbagai macam tipe data, tipe-tipe data ini dibagi menjadi 3, yaitu tipe data untuk bilangan, tipe data untuk tanggal dan jam, dan tipe data untuk karakter.

Tabel 2.8 Tipe Data untuk Tanggal dan Jam

No	Tipe	Keterangan
1	<i>DATE TIME</i>	Ukuran 8 byte.kombinasi tanggal dan jam, dengan jangkauan dari '1000-01-0100:00:00' sampai dengan '9999-12-3123:59:55'
2	<i>DATE</i>	Ukuran 3 byte .Tanggal dengan jangkauan dari '1000-01-01 sampai dengan '9999-12-

No	Tipe	Keterangan
		31'
3	<i>TIMESTAMP</i>	Ukuran 4 byte. Kombinasi tanggal dan jam, dengan jangkauan dari '1970-01-0100:00:00' sampai dengan tahun 2037
4	<i>TIME</i>	Ukuran 3 byte. Waktu dengan jangkauan dari '-838:59:59'
5	<i>YEAR</i>	Ukuran 1 byte. Data tahun antara 1901 sampai dengan 155

Sumber: (Hidayat, 2003)

Tabel 2.9 Tipe data untuk bilangan

NO	Tipe	Keterangan
1	<i>TINYINT</i>	Ukuran 1 byte. Bilangan bulat terkecil, dengan jangkauan untuk bilangan bertanda: 128 sampai dengan 127 dan untuk yang tidak : 0 sampai dengan 255
2	<i>SMALLINT</i>	Ukuran 2 byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -327 sampai dengan 32767 untuk yang tidak bertanda : 0 sampai dengan 65535 Ukuran 3 byte. Tanggal dengan jangkauan dari '1000-01-01 sampai dengan '9999-12-31'
3	<i>MEDIUMINT</i>	Ukuran 3 byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda 8388608 sampai dengan 838 8607 dan untuk yang tidak bertanda : 0 sampai dengan 16777215
4	<i>INT</i>	Ukuran 4 byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -2147483648 sampai dengan 2147483647 dan untuk yang bertanda : 0 sampai dengan 4294967295
5	<i>INTEGER</i>	Ukuran 4 byte. Sinonim dari INT
6	<i>BIGINT</i>	Ukuran 8 byte. Bilangan bulat

NO	Tipe	Keterangan
		terbesar dengan jangkauan untuk bilangan bertanda : -9223372036854775808 sampai dengan 9223372036854775807 dan untuk yang tidak bertanda : 0 sampai dengan 18446744073709551615
7	<i>FLOAT (4)</i>	Ukuran 4 byte. Bilangan pecahan
8	<i>FLOAT (8)</i>	Ukuran 8 byte. Bilangan pecahan
9	<i>FLOAT</i>	Ukuran 4 byte. Bilangan pecahan
10	<i>DOUBLE</i>	Ukuran 8 byte. Bilangan pecahan
11	<i>REAL</i>	Ukuran 8 byte. Sinonim dari <i>DOUBLE</i>

Sumber: (Hidayat, 2003)

Table 3.0 Tipe data untuk karakter

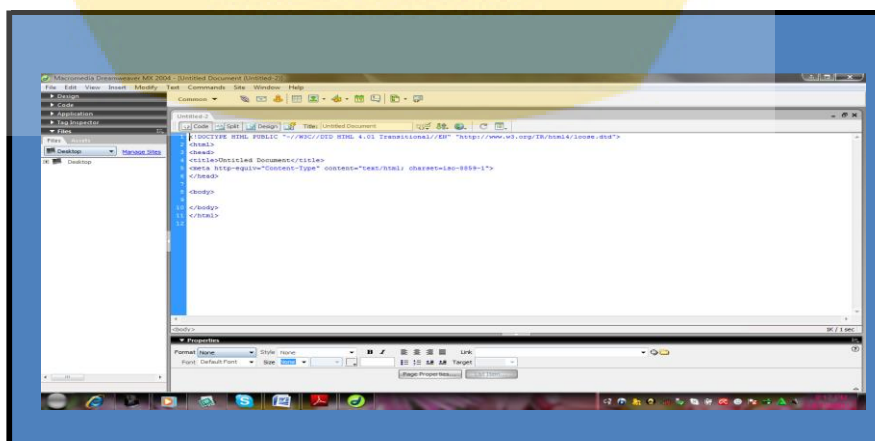
No	Tipe	Keterangan
1	<i>CHAR(M)</i>	Ukuran M byte , $1 \leq M \leq 255$ Data string dengan panjang yang tetap
2	<i>VARCHAR (M)</i>	Ukuran L+1 byte dengan $L \leq M$ Dan $1 \leq M \leq 255$ Data String dengan panjang bervariasi dengan panjang datanya.
3	<i>TINYBLOB, TINYTEXT</i>	L+1 byte, dengan $L < 28$ . Tipe TEXT atau BLOB dengan panjang maksimum 255 karakter.
4	<i>BLOB, TEXT</i>	L+2 byte, $L < 2016$ . Tipe TEXT atau BLOB dengan panjang maksimum 65535 karakter.
5	<i>MEDIUMBLOB, MEDIUMTEXT</i>	L+3 byte, dengan $L < 224$ . Tipe TEXT atau BLOB dengan panjang maksimum 16777215 karakter.
6	<i>ENUM ('nilai 1', 'nilai 2', ...)</i>	Ukuran 1 atau 2 byte tergantung jumlah nilai enumerasinya (Maksimum 65535 nilai)
7	<i>SET (('nilai 1', 'nilai 2', ...))</i>	1, 2, 3, 4, atau 8 byte, tergantung jumlah anggota himpunan (Maksimum 64 anggota).



## 2.9.6 Macromedia Dreamweaver

*Dreamweaver* merupakan program profesional editor *HTML* visual yang digunakan untuk mengelola situs dan menata layout halaman *web*. Saat ini versi terbaru dari *Dreamweaver* yang dikeluarkan oleh Macromedia adalah *Dreamweaver MX 2004*. pada versi ini, tampilannya mengalami perubahan yang kaya akan warna dan area kerjanya menjadi lebih ringkas dan efisien. Hal ini dapat dilihat dengan peletakan tombol-tombol dan panel-panel yang dapat *minimize* (disembunyikan) untuk menghemat area kerja.

Disamping itu, masih banyak terdapat penambahan fasilitas-fasilitas lainnya yang membuat *Dreamweaver MX* lebih *powerfull*, seperti kemampuan menangani penyuntingan kode dengan lebih baik, menangani dokumen-dokumen baru seperti *XHTML*, kemampuan *validasi* dan *debug* di *browser*, panel *snippets* yang berfungsi untuk menyunting dan menyimpan blok kode yang sering digunakan, serta fasilitas penanganan berbagai *database* .( Mutmainah, 2006)



Gambar 2.2 jendela dan panel pada Dreamweaver MX

(Sumber: Herlambang, 2006)

a. *Insert bar* berisi tombol untuk memasukkan beberapa jenis objek, seperti gambar, tabel, dan layer ke dokumen.

b. *Document Toolbar*

*Document toolbar* berisi tombol dan *pop-up* menu yang menyediakan beberapa cara untuk menampilkan dokumen (seperti *Design view* dan *Code view*), beberapa option dan beberapa operasi umum seperti melakukan *preview* dokumen ke *brouser*

c. *Panel Groups*

*Panel Groups* adalah kumpulan beberapa panel sejenis yang diletakkan pada satu tempat

d. *Site Panel*

Pada *Site Panel* bisa untuk mengatur file dan folder yang akan digunakan untuk *site*. Windows Explorer (Windows) akan menggunakan program ini untuk menampilkan file dan folder pada disk lokal.

