



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

Teori ini membahas tentang pengertian-pengertian yang berkaitan dengan judul Laporan Akhir.

##### 2.1.1. Pengertian Komputer

Dhanta (2009:11), “komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas, yaitu menerima *input*, memproses *output* sesuai dengan instruksi yang diberikan, menyimpan perintah-perintah dan hasil pengolahannya, serta menyediakan *output* dalam bentuk informasi”.

Wahyudi (2012:3), “komputer adalah peralatan (*device*) yang menerima data (*input*) dan menyimpan (*storage*) kemudian di proses (*process*) untuk menghasilkan data dalam bentuk lain (*output*).

Siallagan (2009:1), “komputer adalah sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (*process*), memberikan informasi (*output*) dan terkoordinasi di bawah kontrol program yang tersimpan dalam memorinya”.

##### 2.1.2. Pengertian Data

Asropudin (2013:22), “data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti”.

Sutabri (2005:16), “data adalah suatu istilah majemuk yang berarti fakta atau bagian dari fakta yang mengandung arti yang dihubungkan dengan kenyataan, simbol-simbol, gambar-gambar, angka-angka, huruf-huruf, atau simbol-simbol yang menunjukkan suatu ide, objek, kondisi, dan lain-lain”.

Ladjamudin (2013:9), “data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata”.

##### 2.1.3. Pengertian Sistem

Ladjamudin (2013:3), pengertian sistem menurut para ahli yaitu sebagai berikut:

---



*Davis:*

“Sistem sebagai bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud”.

*Lucas:*

“Sistem sebagai suatu komponen atau variabel yang terorganisir, saling berinteraksi, saling bergantung, satu sama lain dan terpadu”.

*Robert G. Murdick:*

“Sistem sebagai seperangkat elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan bersama”.

*Gerald J.:*

“Sistem yaitu suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu”.

#### **2.1.4. Karakteristik Sistem**

Ladjamudin (2013:3), suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu sebagai berikut:

a. **Komponen Sistem**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

b. **Batasan Sistem**

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan dan menunjukkan ruang lingkup dari sistem tersebut.

c. **Lingkungan Luar Sistem**

Lingkungan luar dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan. Lingkungan luar yang menguntungkan

---



merupakan energi dari sistem dan dengan demikian harus dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

d. Penghubung Sistem

Penghubung merupakan media yang menghubungkan antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini kemungkinan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran dari satu subsistem akan menjadi masukan untuk subsistem lainnya melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

e. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal *maintenance input*. Sinyal *maintenance* adalah energi yang dimasukkan supaya sistem tersebut dapat berjalan. Sinyal *input* adalah energi yang diproses untuk mendapatkan keluaran dari sistem.

f. Keluaran Sistem

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

g. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

h. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.



### 2.1.5. Analisis PIECES

Al-Fatta (2007:58-60), untuk mengidentifikasi masalah harus dilakukan analisis terhadap kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi dan pelayanan pelanggan. Paduan ini dikenal dengan analisis PIECES (*Performance, Information, Economy, Control, Efficiency dan Services*).

a. Analisis Kinerja (*Performance*)

Analisis Kinerja adalah kemampuan dalam menyelesaikan tugas bisnis dengan cepat sehingga sasaran segera tercapai.

b. Analisis Informasi (*Information*)

Laporan – laporan yang sudah selesai diproses digunakan untuk menghasilkan informasi yang dibutuhkan oleh manajemen didalam pengambilan keputusan.

c. Analisis Ekonomi (*Economy*)

Analisis Ekonomi adalah penilaian sistem atas pengurangan dan keuntungan yang akan didapatkan dari sistem yang dikembangkan. Sistem ini akan memberikan penghematan operasional dan meningkatkan keuntungan perusahaan.

d. Analisis Keamanan (*Control*)

Analisis Keamanan adalah analisis yang digunakan untuk membandingkan sistem yang dianalisa berdasarkan pada segi ketepatan waktu, kemudahan akses, dan ketelitian data yang diproses. sistem keamanan yang digunakan harus dapat mengamankan data dari kerusakan, misalnya dengan membuat *backup data*. Selain itu sistem keamanan juga harus dapat mengamankan data dari akses yang tidak diijinkan.

e. Analisis Efisiensi (*Efficiency*)

Analisis Efisiensi adalah analisis yang melakukan penggunaan sumber daya secara minimum guna pencapaian hasil yang optimum.

f. Analisis Layanan (*Services*)

Analisis Layanan adalah peningkatan pelayanan yang memperlihatkan kategori beragam. Perkembangan organisasi dipicu peningkatan pelayanan yang lebih baik.



### 2.1.6. Tahapan Studi Kelayakan Sistem

Al-Fatta (2007:75), “studi kelayakan adalah mekanisme untuk menjustifikasi apakah kebutuhan sistem yang dibuat layak untuk dilanjutkan menjadi sistem atau tidak”.

Dokumen yang dihasilkan dari tahapan-tahapan sebelumnya dikumpulkan menjadi suatu proposal pendahuluan proyek. Untuk memastikan usulan tersebut bisa diteruskan menjadi proyek yang menguntungkan maka proposal proyek harus dievaluasi kelayakannya dari berbagai segi kelayakan, diantaranya:

a. Kelayakan teknis

Kelayakan teknis menyoroti kebutuhan sistem yang telah disusun dari aspek teknologi yang akan digunakan. Jika teknologi yang dikehendaki untuk pengembangan sistem merupakan teknologi yang mudah didapat, murah dan tingkat pemakaiannya mudah, maka secara teknis usulan kebutuhan sistem bisa dinyatakan layak.

b. Kelayakan operasional

Kelayakan operasional menyangkut beberapa aspek, usulan kebutuhan sistem harus benar-benar bisa menyelesaikan masalah yang ada di sisi pemesan sistem informasi. Di samping itu, informasi yang dihasilkan oleh sistem harus merupakan informasi yang benar-benar dibutuhkan oleh pengguna tepat pada saat pengguna menginginkannya.

c. Kelayakan ekonomi

Kelayakan ekonomi berhubungan dengan return on investment atau berapa lama biaya investasi dapat kembali. Analisis kelayakan ekonomi juga akan mempertimbangkan apakah bermanfaat melakukan investasi ke proyek ini atau harus melakukan sesuatu yang lain.

### 2.1.7. Pengenalan *E-Library*

Supriyanto dan Muhsin (2012:31), beberapa istilah yang digunakan untuk menggambarkan konsep perpustakaan digital seperti perpustakaan elektronik (*e-library*), perpustakaan maya, perpustakaan *hyper*, perpustakaan *cyber* dan perpustakaan tanpa dinding.



Jadi perpustakaan digital atau *e-library* adalah sebuah sistem yang memiliki berbagai layanan dan obyek informasi yang mendukung akses objek informasi tersebut melalui perangkat digital.

Pada dasarnya, perpustakaan elektronik itu sama saja dengan perpustakaan biasa, hanya saja memakai prosedur kerja berbasis komputer dan sumber informasinya digital. Jaringan informasi *internet* memberikan kesempatan luas untuk mengakses lembaga yang menyediakan informasi.

#### **2.1.8. Keunggulan dan Manfaat *E-library***

Supriyanto dan Muhsin (2012:31), keunggulan *e-library* adalah sebagai berikut:

- a. Layanan jarak jauh (*long distance service*)
- b. Akses yang mudah
- c. Murah (*cost effective*)
- d. Pemeliharaan koleksi secara digital
- e. Jawaban yang tuntas
- f. Jaringan global

Sedangkan manfaat *e-library* adalah sebagai berikut:

- a. Sebagai sumber pengetahuan
- b. Media penyebaran pengetahuan
- c. Untuk penyimpanan (*repository*)
- d. Untuk perawatan/preservasi
- e. Media promosi/etalase hasil karya civitas akademika
- f. Mencegah duplikasi dan plagiat

#### **2.1.9. Pengertian *Web***

Asropudin (2013:109), "*website* adalah sebuah kumpulan halaman (*webpages*) yang diawali dengan halaman muka (*homepage*) yang berisikan informasi, iklan serta program interaksi".

Sutarman (2012:288), "*web* adalah *software* yang digunakan untuk mengakses informasi di *internet*".



#### **2.1.10. Pengertian *Internet***

Asropudin (2013:48), “*internet* adalah istilah umum yang dipakai untuk menunjuk *network* tingkat dunia yang terdiri dari komputer dan layanan servis atau sekitar 30 sampai 50 juta pemakai komputer dan puluhan sistem informasi termasuk *Email, Gopher, FTP dan World Wide Web*”.

Sutarman (2012:283), “*internet* adalah kumpulan dari berbagai macam jenis komputer yang saling terhubung dengan menggunakan media telekomunikasi (telepon, *wireless*, satelit dan sebagainya) dengan jangkauan seluruh dunia/*global*”.

Supriyanto dan Muhsin (2012:120), “*internet* merupakan hubungan antara berbagai jenis komputer dan jaringan di dunia yang berbeda sistem operasi maupun aplikasinya”.

#### **2.1.11. Pengertian Implementasi *E-library* pada Jurusan Manajemen**

##### **Informatika Politeknik Negeri Sriwijaya**

Dari uraian di atas dapat kami simpulkan bahwa pengertian Implementasi *E-library* pada Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya merupakan suatu penerapan perangkat lunak yang biasa disebut dengan perpustakaan elektronik yang mengolah data-data buku dalam bentuk format digital yang diterapkan di Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya dan bisa diakses secara cepat dan mudah.

### **2.2. Teori Khusus**

Teori ini akan membahas mengenai desain sistem yang merupakan suatu fase yang memerlukan suatu keahlian perencanaan untuk elemen-elemen komputer yang akan menggunakan sistem baru.

#### **2.2.1. Model SDLC (*Software Development Life Cycle*) *Waterfall***

Sukamto dan Shalahuddin (2013:28), model SDLC air terjun (*waterfall*) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari:



a. Analisis kebutuhan perangkat lunak

Adalah proses pengumpulan kebutuhan yang dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

b. Desain

Adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengkodean.

c. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap sistem.

d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logis dan fungsional untuk memastikan bahwa semua bagian sudah diuji.

e. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

### **2.2.2. Pengujian Perangkat Lunak *Black-Box Testing***

Sukamto dan Shalahuddin (2013:275), "*black-box testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran perangkat lunak sesuai dengan spesifikasi yang dibutuhkan".

### **2.2.3. *Unified Modelling Language (UML)***

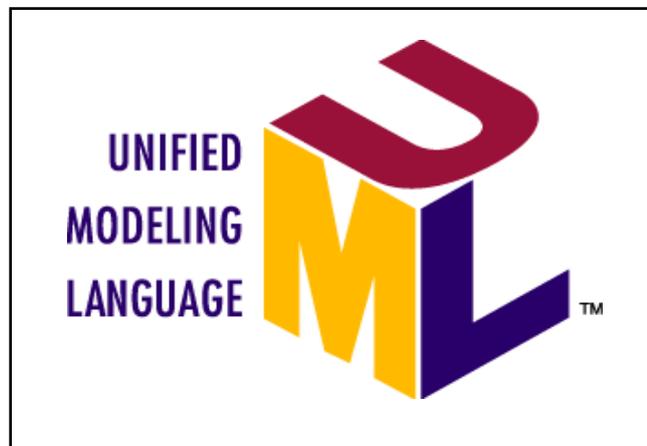
Sukamto dan Shalahuddin (2013:133), "*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek".

---



Widodo dan Herlawati (2011:6), menyatakan bahwa “*UML* singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. *UML* juga dapat diartikan sebagai bahasa yang memiliki sintaks dan semantik”.

*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*

Widodo dan Herlawati (2011:6-7), *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

Lanjutan Tabel 2.1. Tipe Diagram *UML*

2	<i>Package</i>	Memperlihatkan kumpulan kelas-kelas, merupakan dari diagram komponen
3	<i>Use case</i>	Diagram ini memperlihatkan himpunan <i>use case</i> dan aktor-aktor (suatu jenis khusus dari kelas)
4	<i>Sequence</i>	Diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu
5	<i>Communication</i>	Sebagai pengganti diagram kolaborasi <i>UML</i> 1.4 yang menekankan organisasi struktural dari obyek-obyek yang menerima serta mengirim pesan
6	<i>Statechart</i>	Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status ( <i>state</i> ), transisi, kejadian serta aktivitas
7	<i>Activity</i>	Tipe khusus dari diagram 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 (2011:10-12)

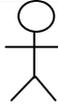
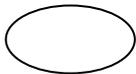
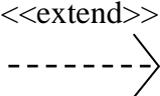
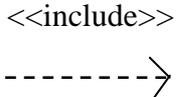
## 2.2.4. Jenis-jenis Diagram pada *UML*

### 2.2.4.1. *Use Case* Diagram

Sukanto dan Shalahuddin (2013:155), “*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		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri
2		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
4		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu
5		<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
6		<i>Include</i>	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 dijalankan <i>use case</i> ini

Sumber : Sukanto dan Shalahuddin (2013:156-158)

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 tabel skenario *use case*:

**Tabel 2.3.** Format Tabel Skenario *Use Case*

AKSI AKTOR	REAKSI SISTEM
Skenario Normal	
Skenario Alternatif	

Sumber : Sukamto dan Shalahuddin (2013:156-158)

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*.

#### 2.2.4.2. Class Diagram

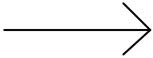
Sukamto dan Shalahuddin (2013:141), “*class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

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

Lanjutan Tabel 2.4. Simbol-simbol *Class Diagram*

2		<i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
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 ( <i>whole-part</i> )

Sumber : Sukanto dan Shalahuddin (2013:146-147)

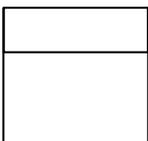
### 2.2.4.3. Activity Diagram

Sukanto dan Shalahuddin (2013:161) “*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 diagram aktivitas memiliki sebuah status awal
2		Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja
3		Decision	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 diagram aktivitas memiliki sebuah status akhir
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : Sukamto dan Shalahuddin (2013:162-163)

#### 2.2.4.4. *Sequence Diagram*

Sukamto dan Shalahuddin (2013:165), “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

### 2.2.5. Kamus Data (*Data Dictionary*)

Sukamto dan Shalahuddin (2013:73), “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)”.

Ladjamudin (2013:70), “kamus data berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses”.

Adapun simbol-simbol yang ada dalam kamus data yaitu sebagai berikut:

**Tabel 2.6.** Simbol-simbol Kamus Data

<b>Simbol</b>	<b>Keterangan</b>
=	disusun atau terdiri dari
+	Dan
[   ]	baik...atau...
{ } <sup>n</sup>	n kali diulang/bernilai banyak
( )	data opsional
*..*	batas komentar

Sumber : Sukamto dan Shalahuddin (2013:73)

## 2.3. Teori Program

### 2.3.1. Pengertian Basis Data (*Database*)

Fathansyah (2004:2), “basis data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan”.

Sukamto dan Shalahuddin (2013:43), “basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

Sutarman (2012:15), “*database* adalah sekumpulan file yang saling berhubungan dan terorganisasi atau kumpulan *record-record* yang menyimpan data dan hubungan diantaranya”.

### 2.3.2. Pengertian Pemrograman *PHP*

Nugroho (2004:140), “*PHP* adalah bahasa program yang berbentuk *script* yang diletakkan di dalam *server web*”.

Sidik (2006:3), “*PHP* merupakan *script* untuk pemrograman *script web server-side*, *script* yang membuat dokumen *HTML* secara *on the fly*, dokumen *HTML* yang dihasilkan dari suatu aplikasi bukan dokumen *HTML* yang dibuat dengan menggunakan editor teks atau editor *HTML*”.

Kadir (2008:2), “*PHP (hypertext Preprocessor)* merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*, hasilnya akan dikirimkan ke klien, tempat pemakai menggunakan *browser*”.



**Gambar 2.2.** Tampilan Logo *PHP*

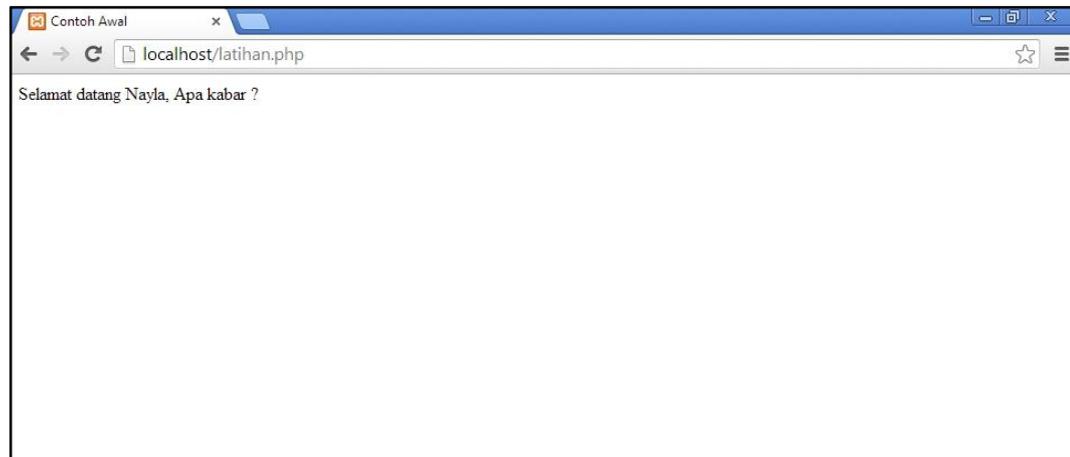
### 2.3.3. *Script PHP*

Nugroho (2004:151), *PHP* adalah program yang fleksibel, artinya *script-script PHP* dapat Anda tuliskan pada sela-sela tag *HTML*. Karena *PHP* memiliki sifat yang dapat berinteraksi dengan program lain, maka kita tidak usah bingung dalam menuliskan skripnya. Jadi, kita dapat membuat sebuah program *PHP* yang di dalamnya berupa tag-tag *HTML* murni yang kemudian disisipi *script PHP*. Akan tetapi, *PHP* juga dapat berdiri sendiri sehingga tidak memerlukan *HTML* murni untuk meletakkan *script-scriptnya*.

#### **Awal *script***

```
<HTML>
<HEAD>
<TITLE> Contoh Awal </TITLE>
</HEAD>
<?php
    Print ("Selamat datang Nayla, Apa kabar ? ");
?>
<BODY>
</BODY>
</HTML>
```

#### **Akhir *script***



**Gambar 2.3.** Contoh *Script PHP*

#### 2.3.4. Teknik Penulisan *Script PHP*

Nugroho (2004:144), ada beberapa cara menuliskan *script PHP* :

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

Jadi, semua kata dan *script* yang diletakkan pada daerah *script* akan dianggap sebagai perintah *PHP* sehingga jika terjadi kesalahan atau kata-katanya tidak sesuai dengan program akan dianggap salah dan akan mengakibatkan program yang kita buat menjadi *error*.

Adapun kriteria yang harus diperhatikan dalam penulisan *script PHP* adalah sebagai berikut :

- a. Setiap halaman yang mengandung *script PHP* harus disimpan dengan EXTENSI *PHP* sesuai dengan program *PHP* yang mendukungnya (nama\_file.php, atau nama\_file.php3, nama\_file.php4).
- b. Setiap *script PHP* harus didahului dengan pembuka *PHP* (`<?php` dan lain-lain) dan kemudian diakhiri dengan penutup (`?>`).
- c. Setiap baris *script* harus didahului pernyataan cetak atau dalam program pascal kita sebut *write* atau *REM*, sedangkan dalam *PHP* pernyataan cetak dibedakan menjadi dua yaitu *Print* dan *Echo*. Adapun kriteria penulisannya sebagai berikut :



**Print ( “ Isi Perintah “ )**

**Printf ( “ Isi Perintah “ )**

**Echo “ Isi Perintah “**

**Echo Perintah**

- d. Setiap akhir baris perintah harus diakhiri dengan titik koma (;) misalnya kita akan menuliskan “Hello, Nama Saya Nayla”, maka *script* yang harus anda tuliskan adalah sebagai berikut :

```
<?php
```

```
Echo “Hello, Nama Saya Nayla”;
```

```
?>
```

- e. Semua variabel harus diberi tanda *string* dolar (\$) pada penulisan awalnya. Sebagai contoh, kita akan menuliskan nama sebagai variabel yang isinya Nayla, maka kita harus menuliskan sebagai berikut :

```
$nama = Nayla;
```

- f. Penulisan *comment* atau keterangan didahului dengan pembuka /\* dan diakhiri dengan \*/. Biasanya kita menggunakan ini untuk memberikan *comment* yang berbentuk kalimat. Sebagai contoh, kita akan memberi keterangan penulisan nama diatas.

Contoh *Script* :

```
<?php
```

```
/* Untuk menuliskan komentar, biasanya komentar yang berbentuk kalimat */
```

```
Echo “Hello, Nama Saya Nayla”;
```

```
?>
```

- g. Selain menggunakan tanda /\*, penulisan *comment* juga dapat menggunakan tanda *slash* ganda (//). Tanda ini hanya digunakan untuk menuliskan pesan yang hanya berisi satu baris saja.

Contoh *Script* :

```
<?php
```

```
// Untuk menuliskan komentar satu baris
```

```
Echo “Hello, Nama Saya Nayla”; ?>
```



- h. Semua *script HTML* yang akan digabungkan dalam *script PHP* harus dihilangkan tanda petiknya (“ ”). Sebagai contoh, jika kita ingin memberi warna *background* pada halaman kita, maka perintah yang kita gunakan dalam *HTML* adalah `<BODY bgcolor=”#000000”>`. Akan tetapi, jika kita letakkan dalam *PHP*, maka *script* tersebut harus diubah menjadi `<BODY bgcolor=#000000>`. Untuk lebih jelasnya, lihatlah contoh berikut.

**Script pada HTML:**

```
.....
<BODY bgcolor=”#000000”>
<?php
Print (“Nama Saya Nayla”);
?>
```

**Script pada PHP:**

```
<?
Echo (“ <BODY bgcolor=#3366FF? “);
Print (“Halo Nayla, apa kabar”);
?>
```

### 2.3.5. Pengertian MySQL

Kadir (2008:2), “*MySQL* (baca: mai-se-kyu-el) merupakan *software* yang tergolong *DBMS (Database Management System)* yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara men-*download* (mengunduh) di *internet* secara gratis”.

Nugroho (2004:29), “*MySQL (My Structure Query Language)* adalah sebuah program pembuat *database* yang bersifat *open source*, artinya siapa saja boleh menggunakannya dan tidak dicekal”.

Kadir (2008:348), “*MySQL* adalah salah satu jenis *database* yang sangat terkenal yang menggunakan *SQL* sebagai bahasa dasar untuk mengakses *datasenya* dan ia bersifat *Open Source* (anda tidak perlu membayar untuk



menggunakannya) pada berbagai *platform* (kecuali untuk jenis *Enterprise* yang bersifat komersial”.

### 2.3.6. Fungsi-fungsi *MySQL*

Kadir (2008:348), berikut adalah sejumlah fungsi yang berawalan `mysql_` yang digunakan untuk mengakses *database server MySQL*, yaitu sebagai berikut:

**Tabel 2.8.** Fungsi-fungsi *MySQL*

No	Nama Fungsi	Kegunaan	Bentuk Pemanggilan
1	<code>MySQL_connect()</code>	Membuat hubungan ke <i>database MySQL</i> yang terdapat pada suatu <i>host</i>	<code>mysql_connect</code> ( <i>host</i> , <i>nama_pemakai</i> , <i>password</i> )
2	<code>MySQL_close()</code>	Menutup hubungan ke <i>database MySQL</i>	<code>mysql_close</code> ( <i>pengenal_hubungan</i> )
3	<code>MySQL_select_db()</code>	Memilih <i>database</i>	<code>mysql_select_db</code> ( <i>database</i> , <i>pengenal_hubungan</i> )
4	<code>MySQL_query()</code>	Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel	<code>mysql_query</code> ( <i>permintaan</i> , <i>pengenal_hubungan</i> )
5	<code>MySQL_db_query()</code>	Menjalankan suatu permintaan terhadap suatu <i>database</i>	<code>mysql_db_query</code> ( <i>database</i> , <i>permintaan</i> , <i>pengenal_hubungan</i> )
6	<code>MySQL_num_rows()</code>	Memperoleh jumlah baris dari suatu hasil permintaan ( <i>query</i> ) yang menggunakan <b>SELECT</b>	<code>mysql_num_rows</code> ( <i>pengenal_hasil</i> )
7	<code>MySQL_affected_rows()</code>	Memperoleh jumlah baris yang dikenai operasi <b>INSERT</b> , <b>DELETE</b> , dan <b>UPDATE</b>	<code>mysql_affected_rows</code> ( <i>pengenal_hubungan</i> )

Lanjutan Tabel 2.8. Fungsi-fungsi *MySQL*

8	<i>MySQL_num_fields()</i>	Memperoleh jumlah kolom pada suatu hasil permintaan	<code>mysql_num_fields</code> ( <code>pengenal_hasil</code> )
9	<i>MySQL_fetch_row()</i>	Menghasilkan suatu <i>array</i> yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil	<code>mysql_fetch_row</code> ( <code>pengenal_hasil</code> )
10	<i>MySQL_fetch_array()</i>	Mempunyai kegunaan serupa dengan <code>mysql_fetch_row()</code> . Hanya saja, setiap kolom akan disimpan dua kali pada <i>array</i> hasil	<code>mysql_fetch_array</code> ( <code>pengenal_hasil</code> )
11	<i>MySQL_fetch_field()</i>	Memperoleh informasi suatu kolom	<code>mysql_fetch_field</code> ( <code>pengenal_hasil</code> [ , <code>nomor_kolom</code> ])
12	<i>MySQL_data_seek()</i>	Memindah <i>pointer</i> pada suatu himpunan hasil supaya menunjuk ke baris tertentu	<code>mysql_data_seek</code> ( <code>pengenal_hasil</code> , <code>nomor_baris</code> )
13	<i>MySQL_field_seek()</i>	Memindah <i>pointer</i> pada suatu himpunan hasil supaya menunjuk ke kolom tertentu	<code>mysql_data_seek</code> ( <code>pengenal_hasil</code> , <code>nomor_kolom</code> )
14	<i>MySQL_create_db()</i>	Menciptakan <i>database MySQL</i>	<code>mysql_create_db</code> ( <code>database</code> [ , <code>pengenal_hubungan</code> ])
15	<i>MySQL_drop_db()</i>	Menghapus <i>database MySQL</i>	<code>mysql_drop_db</code> ( <code>database</code> [ , <code>pengenal_hubungan</code> ])
16	<i>MySQL_list_dbs()</i>	Menghasilkan daftar <i>database MySQL</i>	<code>mysql_list_dbs</code> ( [ <code>pengenal_hubungan</code> ])

**Lanjutan Tabel 2.8.** Fungsi-fungsi *MySQL*

17	<i>MySQL_list_tables()</i>	Memperoleh daftar nama tabel dalam suatu <i>database</i>	<code>mysql_list_tables</code> ( <code>database [ ,</code> <code>pengenal_hubungan</code> <code>]</code> )
18	<i>MySQL_list_fields()</i>	Memperoleh daftar nama kolom dalam suatu <i>database</i>	<code>mysql_list_fields</code> ( <code>database [ ,</code> <code>pengenal_hubungan</code> <code>]</code> )

Sumber : Kadir (2008:360-379)

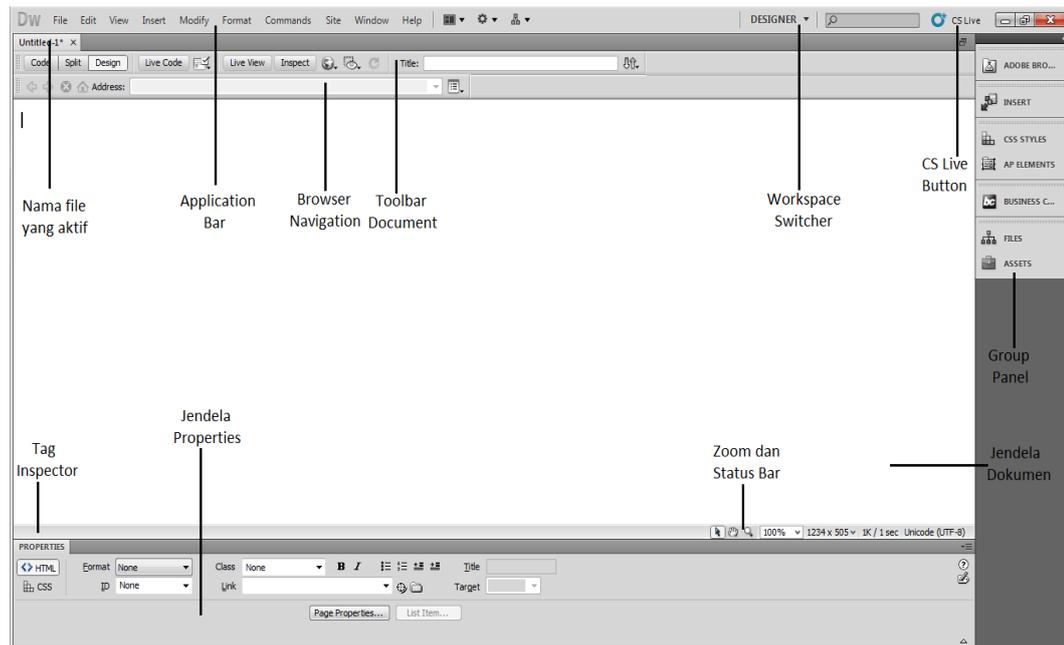
### 2.3.7. Pengertian *Adobe Dreamweaver CS5*

Madcoms (2011:2), “*dreamweaver* adalah sebuah *HTML editor* profesional untuk mendesign *Web* secara visual dan mengelola situs atau halaman *Web*”. Pada *adobe dreamweaver CS5*, terdapat beberapa kemampuan bukan hanya sebagai *software* untuk *design Web* saja tetapi juga untuk menyunting kode serta pembuatan aplikasi *Web* dengan menggunakan berbagai bahasa pemrograman *Web*.

**Gambar 2.4.** Tampilan Logo *Adobe Dreamweaver CS5*

### 2.3.8. Ruang Kerja *Adobe Dreamweaver CS5*

Madcoms (2011:3), tampilan utama ruang kerja *Adobe Dreamweaver CS5* adalah sebagai berikut :



**Gambar 2.5.** Tampilan Ruang Kerja Adobe Dreamweaver CS5

a. *Application Bar*

Berada di bagian paling atas jendela aplikasi *Dreamweaver CS5*. Baris ini berisi tombol *workspace* (*workspace switcher*), *CS Live*, menu dan aplikasi lainnya.



**Gambar 2.6.** Tampilan *Application Bar*

b. *Toolbar Document*

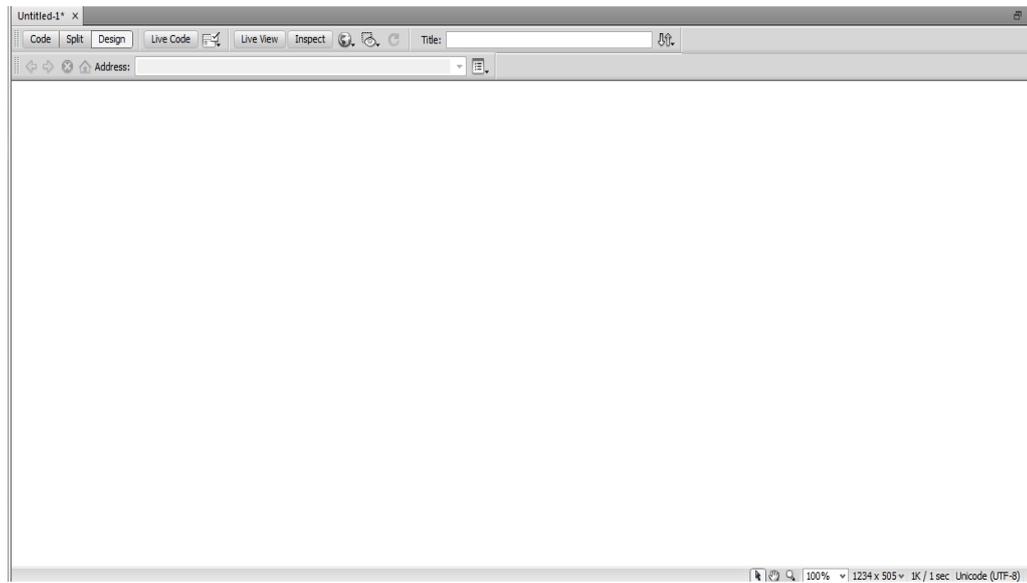
Berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan *Design* atau tampilan *Code*. Juga dapat digunakan untuk operasi-operasi umum, misalnya untuk melihat hasil sementara halaman *web* pada jendela *browser*.



**Gambar 2.7.** Tampilan *Toolbar Document*

c. *Jendela Dokumen*

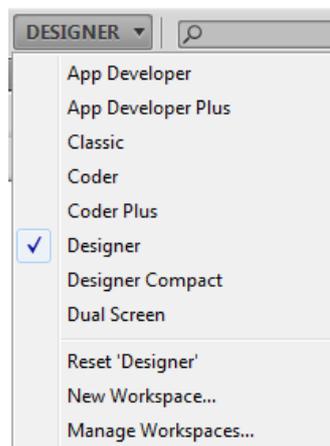
Adalah lembar kerja tempat membuat dan mengedit desain halaman *web*.



**Gambar 2.8.** Tampilan Jendela Dokumen

*d. Workspace Switcher*

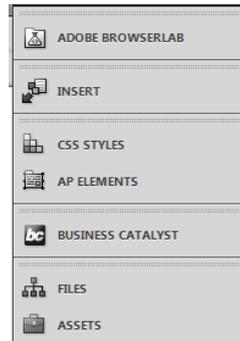
Digunakan untuk mengubah tampilan ruang kerja (*workspace*) Adobe Dreamweaver CS5.



**Gambar 2.9.** Tampilan *Workspace Switcher*

*e. Panel Groups*

Merupakan kumpulan-kumpulan panel yang saling berkaitan, panel-panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan.



**Gambar 2.10.** Tampilan *Panel Groups*

f. *Tag Selector*

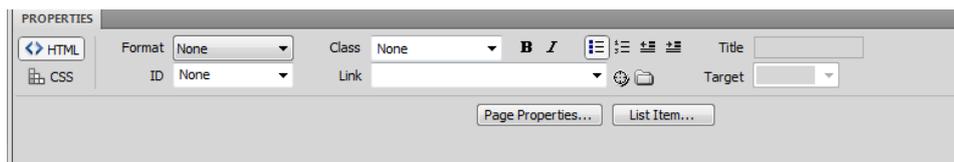
Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut.



**Gambar 2.11.** Tampilan *Tag Selector*

g. *Panel Property*

Digunakan untuk melihat dan mengubah berbagai properti objek atau teks pada jendela *design*.



**Gambar 2.12.** Tampilan Jendela *Properties*

h. *Toolbar Coding*

Berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. Jendela ini hanya akan tampil pada jendela *Code* dan *Split*.



```

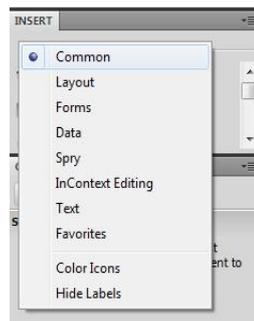
1 <?php
2 $go = (empty($_REQUEST['go'])) ? 'tabel' :
3 $_REQUEST['go'];
4 $go = ($go) ?
5 function tabel() {
6     echo "
7     <a href='\"menu=Deskripsi&go=baru'>Tambah</a>
8     <br>
9     <table>
10        <tr>
11            <td>KodeDeskripsi </td>
12            <td>NamaDeskripsi </td>
13            <td>HargaDeskripsi </td>
14        </tr>
15        <tr>
16            <td>
17            <?mysql_query("select * from Deskripsi");
18            while($row=mysql_fetch_array($res)) {
19                echo "
20                <tr>
21                    <td>{$row[kodeDeskripsi]}</td>
22                    <td>{$row[namaDeskripsi]}</td>
23                    <td>{$row[hargaDeskripsi]}</td>
24                </tr>
25            </td>
26            <a
27            href='\"menu=Deskripsi&go=ubah&kodeDeskripsi={$row[kodeDeskripsi]}\">Ubah</a>

```

**Gambar 2.13.** Tampilan *Toolbar Coding*

*i. Panel Insert*

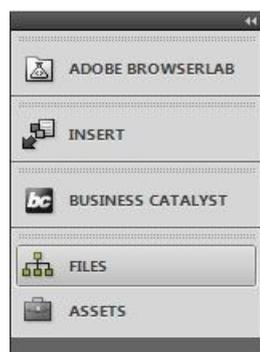
Berisi tombol-tombol untuk menyisipkan berbagai jenis objek, seperti *image*, tabel, atau objek media ke dalam jendela dokumen.



**Gambar 2.14.** Tampilan *Panel Insert*

*j. Panel Groups*

Kumpulan panel yang saling berkaitan satu sama lainnya yang dikelompokkan dibawah satu judul, seperti: *Insert*, *Files*, *CSS Styles*, dan sebagainya.



**Gambar 2.15.** Tampilan *Panel Groups*



k. *Panel Files*

Digunakan untuk mengatur *file-file* dan *folder-folder* yang membentuk situs Web Anda.



**Gambar 2.16.** Tampilan *Panel Files*