

BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Hartono (2013:27), “komputer adalah sebuah mesin yang dapat dikendalikan melalui perintah (*programmable machine*) yang dirancang untuk secara otomatis melakukan serangkaian urutan perhitungan (*arithmetic*) atau proses-proses yang diurutkan secara logis”.

Dhanta (2009:10), “komputer adalah alat yang dipakai untuk mengolah data menurut prosedur yang telah dirumuskan.”

Dari pengertian komputer diatas dapat disimpulkan bahwa komputer adalah komputer adalah alat yang dikendalikan oleh *user* melalui sejumlah perintah yang dipakai untuk mengolah data agar menghasilkan informasi.

2.1.2. Pengertian Aplikasi

Asropudin (2013:6), “aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Ms-Word*, *Ms-Excel*.”

Sujatmiko (2012:23), “aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu, misalnya *Ms-Word*, *Ms-Excel*.”

Dari pengertian aplikasi diatas dapat disimpulkan bahwa aplikasi adalah program yang dibuat untuk membantu manusia dalam menyelesaikan tugasnya.

2.1.3. Pengertian Data

Supardi (2013:12), “data adalah bentuk jamak dari *datum*. Data merupakan keterangan-keterangan tentang suatu hal, dapat berupa sesuatu yang diketahui atau dianggap.”

Arikunto dalam Supardi (2013:12), “data merupakan segala fakta dan angka yang dapat dijadikan bahan untuk menyusun suatu informasi.”

Ladjamudin (2013:20), “data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.”

Mulyanto (2009:16), “data merupakan material atau bahan baku yang belum mempunyai makna atau belum berpengaruh langsung kepada pengguna sehingga perlu diolah untuk dihasilkan sesuatu yang lebih bermakna.”

Dhanta (2009:212), “dalam ilmu komputer data dapat berupa angka, huruf, gambar, atau simbol yang di *input* (dimasukkan) ke komputer dan setelah diolah data itu dapat ditampilkan dalam bentuk *output*.”

Dari beberapa pengertian data diatas dapat disimpulkan bahwa data adalah komponen terkecil dari informasi, data merupakan bahan baku dari informasi.

2.1.4. Pengertian Pengolahan Data

Ladjamudin (2013:9), “pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan.”

Kristanto (2008:8), “pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan.”

Dari pengertian pengolahan data diatas dapat disimpulkan bahwa pengolahan data adalah waktu yang dihabiskan untuk menggambarkan perubahan bentuk data menjadi informasi yang lebih berarti bagi pengguna informasi.

2.1.5. Pengertian Database

Wayteg (2013:58), “*database* adalah suatu jenis perangkat lunak yang dirancang untuk menyimpan dan mengambil data secara tertata.”

Kadir (2013:15), “*database* adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data.”

Dari pengertian *database* diatas dapat disimpulkan bahwa *database* adalah tempat penyimpanan data yang terorganisir dengan baik sehingga mempermudah *user* dalam mengakses data.

2.1.6. Pengertian Fasilitas

Nimpuno (2014:236), “fasilitas adalah sarana yang menunjang atau melancarkan suatu usaha atau kegiatan; kemudahan.”

2.1.7. Pengertian Aset

Nimpuno (2014:76), “aset adalah modal; kekayaan.”

2.1.8. Pengertian Basis

Suharso dan Retnoningsih (2011:78), “basis adalah dasar; pokok; pangkalan; garis dasar; bidang dasar pada kerucut dan sebagainya.”

2.1.9. Pengertian Web

Wayteg (2013:209), “*web* adalah versi singkat dari istilah *World Wide Web*, yaitu kumpulan *file*, atau dokumen yang besar dan saling terkait, yang dibuat tersedia untuk semua orang melalui internet.”

Yosefmurya (2012:3), “*world wide web* atau lebih sering dikenal sebagai *web* adalah suatu layanan informasi yang menggunakan konsep *hyperlink* (tautan), yang memudahkan *surfer* (sebutan para pemakai komputer yang melakukan *browsing* atau penelusuran informasi melalui internet).”

Sidik dan Pohan (2010:1), “*world wide web* (*www*), lebih dikenal dengan *web* merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet.”

Dari pengertian *web* diatas dapat disimpulkan bahwa *web* adalah layanan penyedia informasi melalui internet yang dapat digunakan oleh *user* untuk memperoleh berbagai informasi.

2.1.10. Pengertian Politeknik

Wahya, Suzana dan Waridah (2013:481), “politeknik adalah hal-hal yang berkaitan dengan pengajaran keterampilan dan ilmu-ilmu terapan.”

Suharso dan Retnoningsih (2011:386), “politeknik adalah hal-hal yang bersangkutan dengan pengajaran keterampilan dan ilmu-ilmu terapan.”

Dari pengertian politeknik diatas dapat disimpulkan bahwa politeknik adalah hal yang berhubungan dengan suatu keterampilan dan ilmu-ilmu terapan yang dikukan dengan pengajaran.

2.1.11. Pengertian Negeri

Wahya, Suzana dan Waridah (2013:420), ”negeri adalah wilayah tempat tinggal suatu bangsa; negara; pemerintah, bukan swasta.”

2.1.12. Pengertian Sriwijaya

Cribb dan Kahin (2012:484), ”sriwijaya adalah kerajaan Buddha yang berpusat di kota Palembang modern di Sumatera Selatan dan mendominasi selat Malaka sejak abad ke-7 hingga awal abad ke-11.”

2.1.13. Pengertian Aplikasi Pengolahan Data Fasilitas dan Aset Berbasis Web pada Politeknik Negeri Sriwijaya

Aplikasi Pengolahan Data Fasilitas dan Aset Berbasis *Web* pada Politeknik Negeri Sriwijaya adalah aplikasi yang dibangun untuk pengolahan data fasilitas dan aset dengan tujuan untuk memberikan kemudahan bagi Bagian Inventaris Politeknik Negeri Sriwijaya untuk menerima informasi terbaru mengenai fasilitas dan aset dari masing-masing Staf Pelaksana Inventaris (PI) di semua jurusan dan program studi di Politeknik Negeri Sriwijaya serta untuk mengontrol keberadaan fasilitas dan aset tersebut.

2.2. Teori Khusus

2.2.1. Model SDLC

Sukamto dan Shalahuddin (2013:26), “*SDLC* atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya.”

2.2.2. Model Waterfall

Model *SDLC* air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*).”

2.2.2.1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2.2.2.2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap yang selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

2.2.2.3. 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 desain.

2.2.2.4. Pengujian

Pengujian fokus pada perangkat lunak atau aplikasi secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

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

Tidak menutup kemungkinan sebuah perangkat lunak atau aplikasi mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak atau aplikasi yang sudah ada, tetapi tidak membuat perangkat lunak atau aplikasi baru.

2.2.3. Pengertian *Unified Modeling Language (UML)*

Sukamto dan Shalahuddin (2013:133), “*Unified Modeling Language (UML)* 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.”




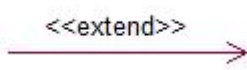

2.2.4. Diagram dalam *Unified Modeling Language (UML)*

2.2.4.1. *Use Case Diagram*

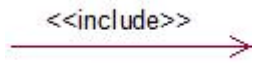
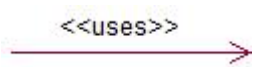
Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case*

mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel 2.1. Simbol-simbol pada *Use Case Diagram*

No.	Simbol	Deskripsi
1.	Use case  nama use case	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
2.	Aktor / <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang, biasanya menggunakan kata benda di nama aktor.
3.	Asosiasi / <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.	Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5.	Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

Lanjutan Tabel 2.1. Simbol-simbol pada *Use Case Diagram*

6.	Menggunakan / <i>include</i> / <i>uses</i>  atau 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , <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 : Sukamto dan Shalahuddin (2013:156)

2.2.4.2. Skenario *Use Case*

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 2.2. Skenario *Use Case*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
Skenario Alternatif	

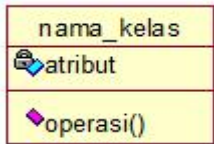






Sumber : Sukamto dan Shalahuddin (2013:161)

Skenario *use case* dibuat per *use case* terkecil, misalkan untuk 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*. Skenario normal dan skenario alternatif dapat lenih dari satu. Alur dari skenario inilah yang nantinya menjadi dasar pembuatan diagram sekuen.

2.2.4.3. Class Diagram

Class diagram atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi. *Class diagram* dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Tabel 2.3. Simbol-simbol pada *Class Diagram*


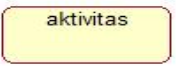



No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber : Sukamto dan Shalahuddin (2013:146)

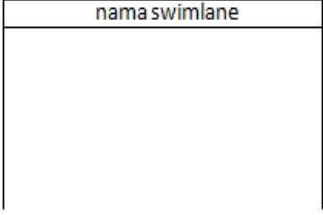

2.2.4.4. Activity Diagram

Activity diagram atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah diagram aktivitas ini menggunakan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.4. Simbol-simbol pada *Activity Diagram*

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Lanjutan Tabel 2.4. Simbol-simbol pada *Activity Diagram*

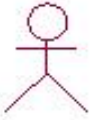
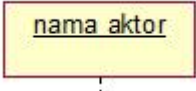

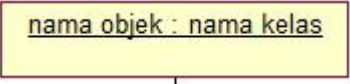

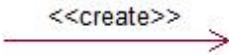
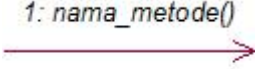
6.	<p>Swimlane</p>  <p>atau</p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>
----	---	---

Sumber : Sukamto dan Shalahuddin (2013:162)

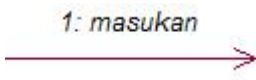
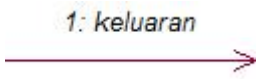
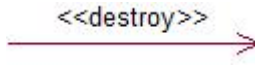
2.2.4.5. *Sequence Diagram*

Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Untuk menggambarkan diagram ini maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Tabel 2.5. Simbol-simbol pada *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang, biasanya menggunakan kata benda di awal frase nama aktor.
2.	<p>Garis hidup / <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek.
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	<p>Pesan tipe <i>create</i></p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	<p>Pesan tipe <i>call</i></p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.

Lanjutan Tabel 2.5. Simbol-simbol pada *Sequence Diagram*

7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : Sukamto dan Shalahuddin (2013:165)

2.2.4.6. *Collaboration Diagram*

Collaboration diagram melihat pada interaksi dan hubungan terstruktur antar objek. Tipe diagram ini menekankan pada hubungan (*relationship*) antar objek, sedangkan *sequence diagram* menekankan pada urutan kejadian. Dalam satu *collaboration diagram* digunakan sebagai alat untuk menggambarkan interaksi yang mengungkapkan keputusan mengenai perilaku sistem.

2.2.5. Analisis *PIECES*

Al Fatta (2007:51), “untuk mengidentifikasi masalah, harus dilakukan analisis terhadap kinerja informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan. Panduan ini dikenal dengan analisis *PIECES* (*Performance, Information, Economy, Control, Eficiency, dan Services*).”

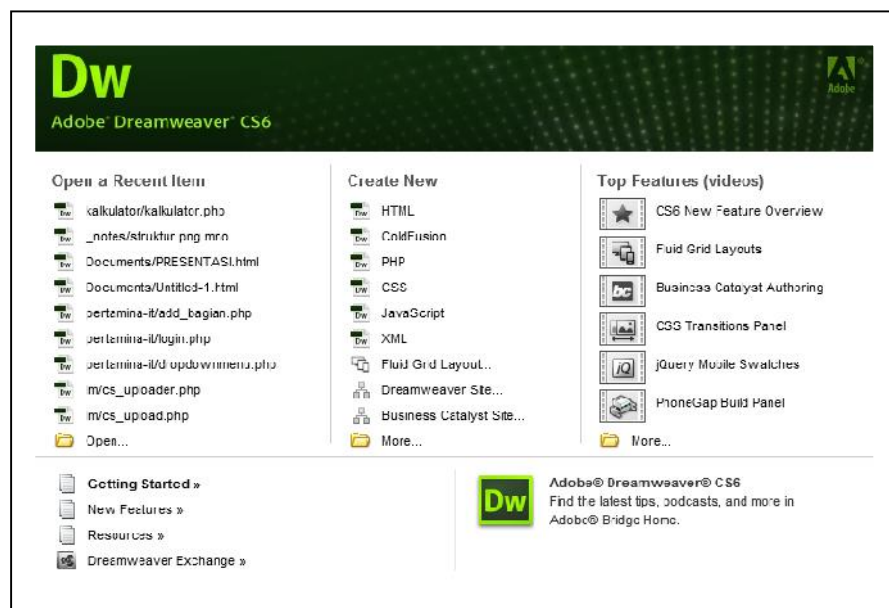
2.3. Teori Program

2.3.1. Pengertian Adobe Dreamweaver CS6

Sulistya (2013:1), “*dreamweaver* merupakan *software* utama yang digunakan oleh desainer *web* dan *programmer web* untuk mengembangkan suatu situs *web*”. *Dreamweaver* memiliki ruang kerja, fasilitas, dan kemampuan yang mampu meningkatkan produktivitas dan efektifitas, baik dalam desain maupun pembuatan situs *web*.

Adobe Dreamweaver CS6 merupakan *software* yang digunakan untuk mendesain suatu *web*. Versi terbaru dari *Adobe Dreamweaver* ialah *Adobe Dreamweaver CS6* memiliki beberapa kemampuan antara lain untuk desain *web*, menyunting kode serta pembuatan aplikasi *web* antara lain *JSP*, *PHP*, *ASP*, *XML*, dan *ColdFusion*.

Dreamweaver CS6 memiliki peningkatan dalam kemampuan *toolbar*, yaitu dapat digunakan untuk memodifikasi tampilan *toolbar* atau menambahkan fungsi baru. Selain itu, *Dreamweaver CS6* memiliki kemampuan untuk menyunting kode dengan lebih baik. *Dreamweaver CS6* juga dapat melakukan *print* kode pada jendela *Code View*, serta memiliki fasilitas *Code Hints* yang membantu dalam urutan *tag-tag*, serta *Tag Inspector* yang sangat berguna dalam menangani *tag-tag HTML*.



Gambar 2.1. Tampilan Awal *Adobe Dreamweaver CS6*

2.3.2. Ruang Kerja *Adobe Dreamweaver CS6*

Di dalam ruang kerja *Dreamweaver CS6*, terdapat beberapa komponen utama yang membentuk suatu ruang kerja *Dreamweaver CS6* ini sehingga memberikan kesatuan tampilan antara menu utama, *panel*, *property inspector* serta *toolbar*.

1. Menu Utama

Sistem menu yang terdapat pada *Dreamweaver CS6* sangat sederhana dan mudah untuk dipahami karena perintah-perintah yang terdapat pada hampir sebagian besar fungsi menu terdapat juga dalam *panel*. Dengan demikian, tidak akan banyak menghabiskan waktu dengan melakukan perintah yang terdapat pada menu, tetapi lebih banyak bekerja dengan menggunakan perintah yang terdapat pada menu dan lebih banyak bekerja dengan mengundang dan menggunakan perintah-perintah yang terdapat pada *panel*.

Di dalam sistem menu utama *Dreamweaver CS6*, terdapat beberapa komponen yang dapat digunakan, yaitu:

a. Menu *File*

Berisikan *item menu* standar untuk *menu File* dan *menu Edit*, seperti *New*, *Open*, *Save*, *Save All*, *Cut*, *Copy*, *Paste*, *Undo*, dan *Redo*. *Menu File* terdiri dari beberapa macam perintah untuk menampilkan atau memperlakukan dokumen baru, seperti *Preview in Browser* dan *Print Code*.

b. Menu *Edit*

Berisikan perintah untuk pemilihan seperti *Select Parent Tag*, serta *Find* dan *Replace*.

c. Menu *View*

Berisikan perintah untuk menampilkan beberapa macam tampilan dari dokumen (seperti tampilan *Design view* atau tampilan *code*), menampilkan dan menyembunyikan beberapa elemen halaman, tombol-tombol *Dreamweaver*, *toolbar* dan lain-lain.

d. Menu *Insert*

Berisikan alternatif baris *insert* untuk menyisipkan objek di dalam dokumen.

e. Menu *Modify*

Berfungsi melakukan perubahan properti pada *item* atau halaman terpilih. Dengan menggunakan menu ini, *user* dapat melakukan pengeditan atribut-atribut *tag*, mengganti tabel dan elemen tabel, dan bermacam bentuk aksi untuk *item library* dan *template*.

f. Menu *Format*

Berfungsi mempermudah dalam melakukan format *text*.

g. Menu *Commands*

Berisikan akses ke beberapa perintah, format kode yang sesuai dengan format pilihan pengguna, membuat *photo album*, dan dapat melakukan pengeditan gambar.

h. Menu *Site*

Berisikan *item* menu untuk mengelola situs dan melakukan *upload* dan *download file*.

i. Menu *Window*

Menyediakan akses ke seluruh *panel*, *inspector*, dan jendela di dalam *Dreamweaver*.

j. Menu *Help*

Menyediakan akses ke dokumen *Dreamweaver*, membuka lembar kerja *Help* untuk membantu pengguna dalam menggunakan *Dreamweaver*, dan menyediakan bahan referensi untuk beberapa bahasa.




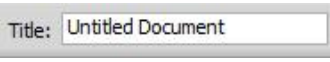







2. *Document Toolbar*

Document Toolbar berisikan tombol-tombol dan menu *pop-up* yang menyediakan tampilan berbeda dari jendela dokumen.

Tampilan jendela dokumen dalam terdapat dalam bentuk *Design View* atau *View* atau kedua-duanya, dan juga terdapat bagian *Title* yang berfungsi untuk memberikan judul dokumen.



Gambar 2.2. Tampilan *Document Toolbar*
Tabel 2.6. Keterangan *Document Toolbar* pada *Adobe Dreamweaver CS6*

No.	Tombol	Keterangan
1.		Untuk menampilkan <i>mode</i> pengetikan kode <i>HTML</i> dalam <i>Code View</i> .
2.		Untuk menampilkan jendela <i>Code View</i> dan jendela <i>Design View</i> .
3.		Untuk menampilkan jendela <i>Design View</i> . Pada tampilan ini yang terlihat akan tampak sama seperti pada jendela <i>browser</i> .
4.		Untuk memberikan judul pada dokumen <i>web</i> .
5.		Untuk proses <i>upload</i> dan <i>download file</i> .
6.		Melihat hasil dalam <i>browser</i> , mencari kesalahan <i>script</i> dalam dokumen, dan digunakan untuk memasukkan <i>browser</i> dalam daftar <i>browser</i> yang dikenali oleh <i>Dreamweaver</i> .
7.		Melakukan pembaruan terhadap isi situs yang dibuat.
8.		Digunakan untuk mengecek <i>CSS</i> yang mirip dari <i>browser</i> berbeda.
9.		Untuk menampilkan hasil sebenarnya (<i>Live</i>), perubahan dari <i>Design View</i> ke hasil sebenarnya di <i>browser</i> .
10.		Untuk menampilkan <i>preview</i> halaman <i>editing</i> saat ini karena akan muncul dalam beberapa perangkat yang mendukung resolusi layar berbeda antara lain <i>phone</i> (320 x 300), <i>tablet</i> (768 x 300), dan <i>desktop</i> (1126 x 214).
11.		Digunakan untuk menciptakan halaman standar <i>HTML</i> dan <i>XHTML</i> yang sesuai.

Sumber : Sulistya (2013:5)

3. *Insert Bar*

Insert Bar terdiri dari tombol-tombol untuk membuat dan menyisipkan objek seperti tabel, *layer* dan gambar. Ketika pengguna menggulung *pointer* melewati sebuah tombol, sebuah kotak keterangan akan tampak dan menampilkan nama tombol.

Tombol-tombol tersebut mengatur beberapa kategori. Jika kategori tidak ingin ditampilkan, klik kategori *Insert*.

Insert Bar dikategorikan dan digunakan untuk mengatur beberapa pilihan sebagai berikut:

a. Kategori *Common*

Digunakan untuk membuat dan menyisipkan objek seperti gambar dan *label*.

b. Kategori *Layout*

Digunakan untuk menyisipkan tabel, *div tag*, *layer*, dan *frame*. Pengguna juga dapat memilih salah satu dari 3 (tiga) tampilan tabel, yaitu *Standard (default)*, *Expanded Tables*, dan *Layout*. Ketika *mode* tampilan terpilih, pengguna dapat menggunakan tombol *Layout Dreaweaver*, yaitu *Draw Layout Cell* dan *Draw Layout Table*.

c. Kategori *Form*

Berisi tombol-tombol untuk membuat *form* dan menyisipkan elemen *form*.

d. Kategori *Text*

Digunakan untuk menyisipkan beberapa teks dan daftar format *tag* seperti *b*, *em*, *p*, *h1*, dan *u1*.

e. Kategori *Favorites*

Berguna untuk mengelompokkan tombol-tombol baris *insert* yang digunakan dalam suatu wadah.

4. Jendela Dokumen

Jendela Dokumen merupakan bagian yang digunakan untuk mendesain halaman situs *web*. Dalam jendela dokumen, pengguna dapat menyisipkan

teks, *image* serta objek lain yang mendukung pembuatan situs *web* atau tempat untuk membuat desain *web*.

Dreamweaver memuat dokumen-dokumen yang sedang dibuka dalam jendela yang sama. Jika diperhatikan, sebuah dokumen yang terbuka dalam jendela yang sama, dokumen akan disimpan dalam *tab-tab* seperti lembar kerja pada *Microsoft Excel*. Dengan demikian, pengguna dapat dengan mudah berpindah antar dokumen tanpa harus menutup tampilan jendela yang sudah terbuka dengan jalan mengklik *tab* dokumen yang pengguna inginkan.

5. *Panel Group*

Panel Group adalah kumpulan *panel* yang dikelompokkan bersama di bawah satu judul. Kumpulan *panel* tersebut antara lain: *Design*, *Code*, *Application*, dan *Files*. Untuk menguraikan *panel group* pengguna bisa mengklik panah *expander* di sebelah kiri nama *group*.

Pengguna juga dapat memisahkan *pane group* yang tersusun tersebut dengan cara menggeser *mouse* pada bagian pojok kiri dari judul *group* sampai *panel* tersebut terlepas dari *panel group*.

6. *Panel CSS*

CSS merupakan pendukung *HTML* dalam mendesain *web*. Perintah-perintah yang sering digunakan dapat dikumpulkan dalam *CSS* sehingga dapat digunakan kembali dalam mendesain *web* dan pengguna tidak perlu mengulang perintah dari awal. Dalam *panel CSS*, pengguna dapat membuat, mengedit, atau menampung seluruh perintah *CSS*.

7. *Panel Files*

Dalam *panel files* terdapat *toolbar* yang membantu pengguna dalam membangun *web*. Dalam panel ini, pengguna dapat melihat deretan *file-file* yang ada dalam *web* pengguna dan menentukan lokasi dari situs yang pengguna buat. Dalam *panel files* juga disediakan tombol untuk koneksi atau diskoneksi dari *Remote Host*.

8. *Panel Snippets*

Panel Snippets merupakan kode yang sering digunakan dan disisipkan pada dokumen. Pengguna dapat mengambil kode *snappet built-in* yang ada di

dalam bagian *snippet* dengan cara menentukan terlebih dahulu kode apa yang akan disisipkan ke dalam dokumen. Misalnya, apabila ingin membuat sebuah judul beserta *menu*, maka pengguna harus memilih *Header*. Selanjutnya, pengguna bisa menentukan bentuk *Header* tersebut. Selain itu, pengguna juga bisa menentukan kode-kode yang lain, seperti *footer*, *layout* navigasi, fungsi *JavaScript* dan masih banyak lagi yang dapat dilihat pada hierarki.

9. *Tag Inspector*

Dalam *Tag inspector*, pengguna dapat melihat struktur hierarki *tag* dan atribut-atribut yang dapat ditambahkan atau *diedit*.

10. *Tag Selector*

Tag Selector terdapat pada bagian bawah jendela dokumen atau pada bagian *status bar*. Fungsi *tag selector* adalah menampilkan hierarki *tag* pada *design view* yang aktif. Secara otomatis, pada saat pengguna mengaktifkan jendela jendela dokumen, maka pada bagian *status bar* akan muncul indikator *<body>*.

11. *Property Inspector*

Property Inspector digunakan untuk melihat dan mengubah berbagai *property* objek dan teks yang terpilih. Setiap objek maupun teks mempunyai properti yang berbeda-beda.

2.3.3. Sejarah Singkat *PHP*

Pada awalnya *PHP* merupakan kependekan dari *Personal Home Page* (Situs Personal) dan *PHP* itu sendiri pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995, dan pada saat *PHP* masih bernama *FI* (*Form Interpreter*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari *web*. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum.



Gambar 2.3. Logo *PHP* (*Hypertext Preprocessor*)

2.3.3.1. Pengertian *PHP (Hypertext Preprocessor)*

Sukanto dan Shalahuddin (2013:102), “*PHP* merupakan bahasa pemrograman yang digunakan untuk pemrograman *web*.”

Saputra (2013:5), “*PHP* (dibaca : *PHP Hypertext Preprocessor*) merupakan suatu bahasa pemrograman yang hanya dapat berjalan pada sisi *server* (*Server Side Scripting*). Artinya proses yang dibuat dengan *PHP* tidak akan berjalan tanpa menggunakan *web server*. *PHP* digunakan untuk membangun aplikasi berbasis *web* agar *web* tersebut dapat digunakan secara dinamis, seperti menambah, mengubah, membaca, serta menghapus suatu konten.

Wahana Komputer (2011:14), “*PHP* merupakan bahasa pemrograman yang digunakan untuk membuat program *website* dimana kode program yang telah dibuat dikompilasi dan dijalankan pada sisi *server* untuk menghasilkan halaman *website* yang dinamis.”

Madcoms (2011:228), “*PHP* adalah sebuah bahasa *scripting* yang terpasang pada *HTML*. Sebagian besar sintaksnya mirip dengan bahasa pemrograman *C*, *Java*, *asp*, dan *Perl*, ditambah beberapa fungsi *PHP* yang spesifik dan mudah dimengerti.”

Dari beberapa pengertian *PHP* diatas dapat disimpulkan bahwa *PHP* adalah bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis *web* atau *website* yang hanya dapat berjalan pada sisi *server*.

Adapun kelebihan-kelebihan dari *PHP* antara lain sebagai berikut :

- a. *PHP* mudah dibuat dengan kecepatan akses tinggi.
- b. *PHP* dapat berjalan dalam *web server* yang berbeda dan dalam sistem operasi yang berbeda pula.
- c. *PHP* diterbitkan secara gratis.
- d. *PHP* juga dapat berjalan pada *web server Microsoft Personal, Web Server, XAMPP, Apache, IIS, Xitami*, dan sebagainya.
- e. *PHP* adalah termasuk bahasa yang *embedded* (bisa ditempel atau diletakkan dalam *tag HTML*).
- f. *PHP* termasuk *server-side programming*.

2.3.4. Pengertian MySQL

Kadir (2013:15), “MySQL adalah nama *database server*. *Database server* adalah *server* yang berfungsi untuk menangani *database*.”

Wahana Komputer (2011:15), “MySQL merupakan salah satu perangkat lunak untuk sistem manajemen *database SQL*.”

MySQL termasuk jenis *RDMS (Relational Database Management System)*. Sehingga istilah seperti tabel, baris, dan kolom tetap digunakan dalam *MySQL*. Pada *MySQL* sebuah *database* mengandung satu atau beberapa tabel, tabel terdiri dari sejumlah baris dan kolom. Dalam konteks bahasa *SQL*, pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi yang terdiri atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom (*coloumn*). Baris pada tabel sering disebut sebagai *instance* dari data sedangkan kolom sering disebut sebagai *attributes* atau *field*.



Gambar 2.4. Logo MySQL

2.3.4. Pengertian XAMPP

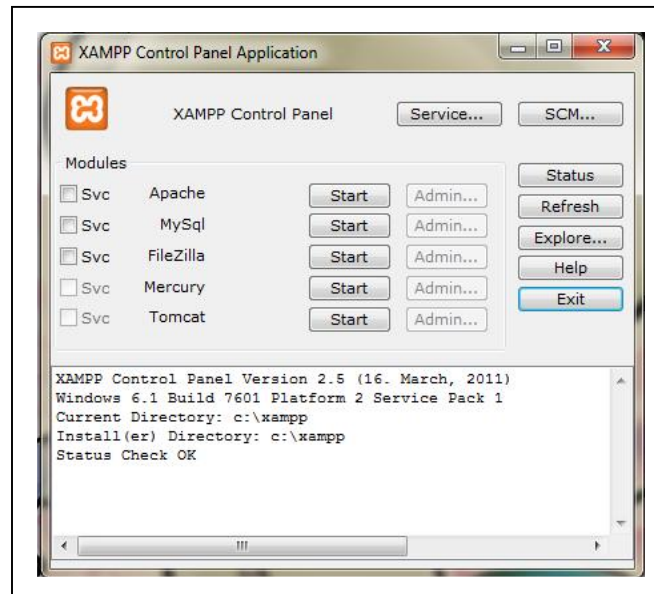
Riyanto (2014:1), “XAMPP merupakan paket *PHP* dan *MySQL* berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis *PHP*.”

Mulhim (2013:4), “XAMPP adalah paket instalasi program yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*.”

Paket XAMPP, sesuai dengan kepanjangannya, X yang berarti *Windows* atau *Linux*, pengguna bisa memilih paket yang diinginkan untuk *Windows* atau *Linux*. XAMPP termasuk paket *server* yang paling mudah untuk digunakan sebagai paket untuk pengembangan aplikasi *web*. XAMPP termasuk paket yang

paling bagus *updatenya*, sehingga paling baik dipilih untuk digunakan untuk *development* atau pun untuk produksi.

Dari beberapa pengertian *XAMPP* diatas dapat disimpulkan bahwa *XAMPP* adalah paket instalasi program yang digunakan untuk *server* dalam membuat aplikasi berbasis *web*.



Gambar 2.5. Tampilan *XAMPP Control Panel Application*

2.3.5. Pengertian *PHPMysqlAdmin*

Supardi (2013:25), "*PHPMysqlAdmin* merupakan sebuah aplikasi *open source* yang berfungsi untuk memudahkan pengelolaan *database MySQL*."

Sutarman (2007:193), "*PHPMysqlAdmin* merupakan salah satu tool manajemen *database MySQL* berbasis *web*, artinya interaksi pemeliharaan dilakukan dengan oleh *client* dengan menggunakan antarmuka *browser*."

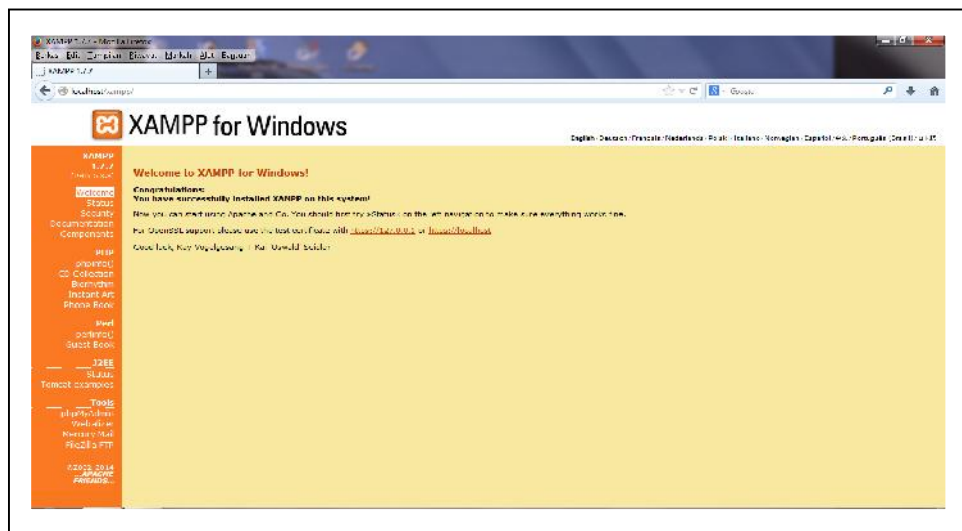
Dari beberapa pengertian *PHPMysqlAdmin* diatas dapat disimpulkan bahwa *PHPMysqlAdmin* adalah aplikasi yang berfungsi untuk menyimpan data dan mengelola *database MySQL*.

Dengan menggunakan *PHPMysqlAdmin*, pengguna dapat membuat *database*, membuat tabel, meng*insert*, meng*hapus*, dan meng*update* data dengan menggunakan tampilan (*GUI*) yang diakses oleh *browser* sehingga pengelolaan

database *MySQL* dapat menjadi lebih mudah karena bisa dilakukan tanpa mengetikkan perintah *SQL* secara manual.

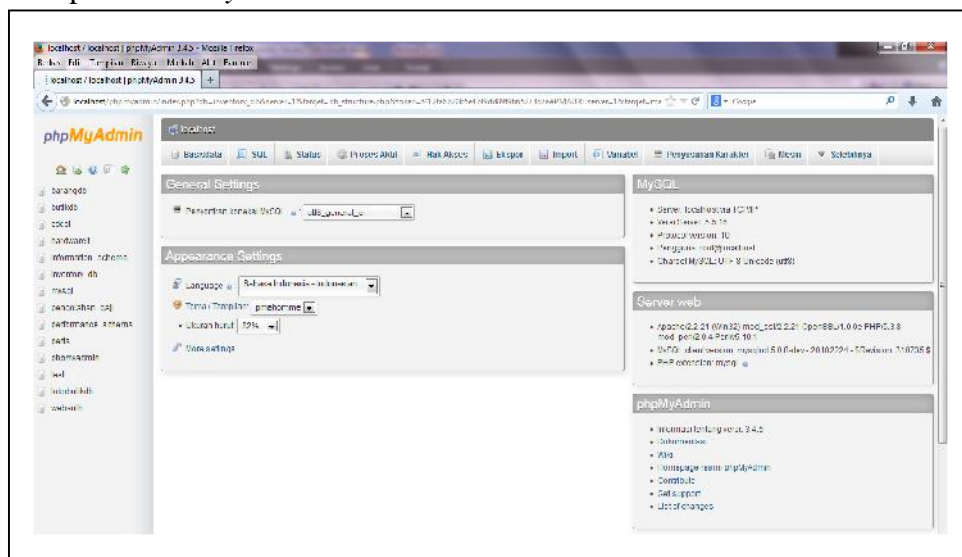
Langkah-langkah untuk menjalankan *PHPMyAdmin* adalah sebagai berikut:

1. Buka *XAMPP Control Panel Application*, pastikan *Apache* dan *MySQL* dalam keadaan *running*, kemudian *exit*.
2. Jalankan *web browser* (misalnya *mozilla firefox* atau *internet explorer*).
3. Ketikkan pada *URL: HTTP//localhost* dan tekan tombol *Enter*.



Gambar 2.6. Tampilan halaman *XAMPP*

4. Geser halaman sehingga muncul bagian.
5. Klik pada *PHPMyAdmin*.



Gambar 2.7. Tampilan Awal *PHPMyAdmin*