

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Tinjauan pustaka dalam melaksanakan penelitian dan untuk menyusun tugas akhir ini, mahasiswa melakukan penelitian untuk mencari beberapa *website* PAUD yang sudah diterapkan di beberapa sekolah yang sudah mempunyai *website*. Hal ini berguna bagi mahasiswa untuk menjadikan bahan referensi dan pembandingan bagi penulis. Sebagai referensi dalam membangun *website* PAUD Pambudi Luhur Karanganyar mahasiswa melakukan pengamatan dari berbagai contoh *website* misalnya seperti *webiste* www.paudaisyiahnuraini.sch.id dan www.paud-inarah.com. Dari *website* tadi penulis banyak mengetahui fasilitas dan fitur-fitur yang dimiliki *website* tersebut.

2.1.1. Website Paud Aisyiyah Nur'aini



Gambar 2.1 Website Paud Aisyiyah Nur'aini
(Sumber : www.paudaisyiahnuraini.sch.id)

Website PAUD Aisyiyah Nur'aini yang dapat diakses melalui alamat www.paudaisyiahnuraini.sch.id, *website* ini dari sisi penampilannya sangat *simple* bahkan tidak terlalu banyak *content-content* seperti Gambar. Jika dilihat

dari segi penggunaannya mudah digunakan dan ada fitur yang berguna untuk media chat online ke admin, jika dari sisi informasi website ini sangat lengkap.

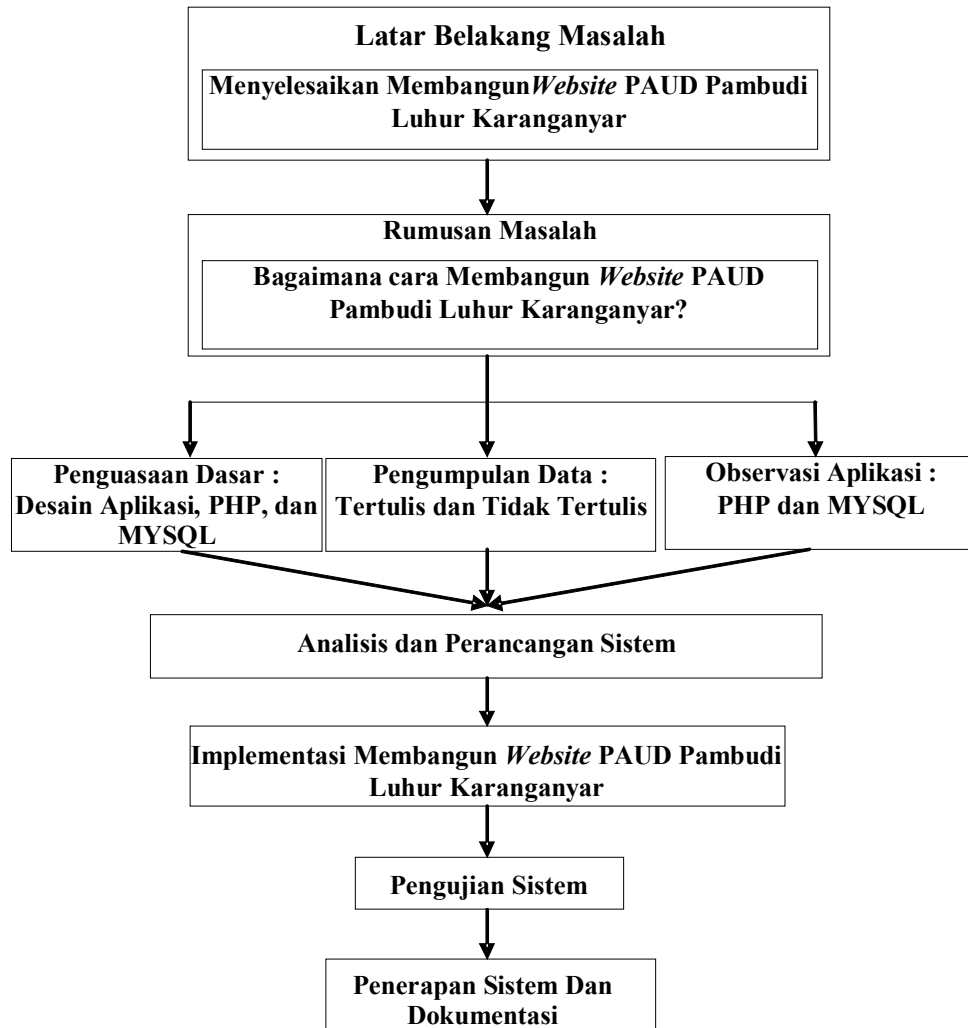
2.1.1. Website Paud Inarah



Gambar 2.2 Website Paud Inarah
(Sumber : www.paud-inarah.com)

Website PAUD Inarah yang dapat diakses melalui alamat www.paud-inarah.com, *website* ini sangat bagus dan mempunyai fitur dan fasilitas banyak di halaman *website* seperti informasi profil sekolah yang sangat lengkap, informasi sekolah, informasi pembelajaran dan berita terbaru. Dari sisi tampilan sangat *simple* mudah digunakan oleh user atau orang tua yang ingin mengakses dan membaca *website* tersebut.

2.2. Kerangka Pemikiran



Gambar 2.3. Kerangka Pemikiran

Penjelasan Kerangka Pemikiran :

a. Latar Belakang Masalah

Pokok permasalahan yang mendasari perlunya dibangun sebuah *website* PAUD Pambudi Luhur Karanganyar .

b. Perumusan Masalah

Perumusan masalah merupakan rumusan masalah atau inti permasalahan dan jalan keluar untuk menyelesaikan permasalahan yang didapatkan berdasarkan pokok permasalahan pada latar belakang masalah.

c. Pengumpulan Data Tertulis dan Tidak Tertulis

Mengumpulkan semua data yang dibutuhkan, baik melalui *interview* dengan Guru, Karyawan dan observasi di PAUD Pambudi Luhur Karanganyar.

d. Penguasaan Dasar (Desain Aplikasi, PHP dan MySQL)

Melakukan beberapa percobaan membuat aplikasi sederhana dengan tujuan agar dapat lebih menguasai bahasa pemrograman PHP dan *database* MySQL sehingga hasilnya menjadi lebih maksimal.

e. Observasi Aplikasi

Mengamati beberapa aplikasi yang sudah ada, baik dari karya ilmiah, buku, atau *internet* yang dapat dijadikan referensi untuk membangun aplikasi.

f. Analisis dan Perancangan Sistem

Menganalisa dan merancang sistem yang akan dibangun seperti apa, bagaimana desainnya, dan apa saja isinya, sehingga sistem ini dapat membantu memecahkan permasalahan.

g. Implementasi Membangun *Website* PAUD Pambudi Luhur Karanganyar

a) Perancangan *Database* MySQL

Membuat *database* dari data-data yang telah didapatkan sesuai dengan kebutuhan sistem.

b) Perancangan Aplikasi Pengolahan Data

Membuat aplikasi dengan dasar *database* yang telah selesai dibuat.

h. Pengujian Sistem

Pengujian dilakukan bertujuan untuk mengetahui jika ternyata masih ada kesalahan atau kekurangan pada aplikasi.

i. Penerapan Sistem Dan Dokumentasi

Pada tahap akhir, dimana sistem telah siap digunakan pada website PAUD Pambudi Luhur Karanganyar setelah melewati tahap pengujian dan membuat dokumentasi dari keseluruhan kegiatan penyusunan Tugas Akhir.

2.3. Teori-teori Pendukung

2.3.1. Pengertian Sistem

Sistem (*system*) dapat di definisikan dengan pendekatan prosedur dan dengan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dan prosedur-prosedur yang mempunyai tujuan tertentu. Contoh: sistem yang di definisikan dengan pendekatan prosedur ini adalah sistem akuntansi. Sistem ini didefinisikan sebagai kumpulan dan prosedur-prosedur penerimaan kas, pengeluaran kas, penjualan, pembelian dan buku besar. Dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu. Contoh sistem yang didefinisikan dengan pendekatan ini misalnya adalah sistem yang didefinisikan sebagai kumpulan dari program keras dan perangkat lunak (Jogiyanto,2003:34).

Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan. Informasi adalah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi (Yakub,2012).

2.3.2. Pengertian Informasi

Informasi merupakan salah satu sumber daya yang sangat diperlukan dalam suatu organisasi. Suatu sistem apabila tidak mendapatkan informasi yang cukup tentu saja tidak akan bertahan lama. Informasi yang berguna bagi suatu sistem akan menghindari *entropy*, yaitu suatu keadaan di mana suatu sistem sudah tidak berjalan sesuai dengan tujuannya atau keadaan dimana suatu sistem sudah

hampir mati. informasi adalah data yang diolah menjadi bentuk-bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya sedangkan data merupakan sumber informasi yang menggambarkan suatu kejadian yang nyata (Agus Mulyanto,2000:2).

Menurut Jogianto (2005:10), informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau data item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

2.3.3. Pengertian Sistem Informasi

Menurut Andri Kristanto (2008:11), sebuah Sistem Informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengelola data menggunakan perangkat keras dan perangkat lunak tersebut. Selain itu data juga memegang peranan yang penting dalam sistem informasi. Data yang akan dimasukkan dalam sebuah sistem informasi dapat berupa formulir-formulir, prosedur-prosedur dan bentuk data lainnya.

Selain itu sistem informasi dapat didefinisikan sebagai berikut :

1. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
2. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi.
3. Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial.

Sistem Informasi adalah suatu sistem yang tujuannya menghasilkan informasi. Sebagai suatu sistem, untuk dapat memahami sistem informasi, akan lebih baik jika konsep dari sistem itu dipahami terlebih dahulu. Demikian juga sebagai sistem penghasil informasi, maka konsep informasi perlu dipahami terlebih dahulu (Jogiyanto,2003:33).

Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada pemakai (Abdul Kadir, 2003).

Sistem Informasi adalah aplikasi komputer untuk mendukung operasi dari suatu organisasi: operasi, instalasi, perawatan komputer, *software*, dan data. Sistem Informasi dimulai sebagai dasar dari analisis komunikasi dan basis data (Akhmad Fauzi,2008:9)

2.3.4. Pengertian *Website*

Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*) (Hendra W Saputro,2007).

Website disebut juga *site*, situs, situs *web* atau portal. Merupakan kumpulan halaman *web* yang berhubungan antara satu dengan lainnya, halaman pertama sebuah *website* adalah *home page*, sedangkan halaman demi halamannya secara mandiri disebut *web page*, dengan kata lain *website* adalah situs yang dapat diakses dan dilihat oleh para pengguna internet diseluruh dunia. *Website* adalah situs yang dapat diakses dan dilihat oleh para pengguna Internet. Pengguna Internet semakin hari semakin bertambah banyak, sehingga hal ini adalah potensi pasar yang berkembang terus (Wahidin Abbas,2013).

WWW atau *World Wide Web* adalah suatu program yang ditemukan oleh Tim Berners-Lee pada tahun 1991. Awalnya Berners-Lee hanya ingin menemukan cara untuk menyusun arsip-arsip risetnya. Untuk itu, dia mengembangkan suatu sistem untuk keperluan pribadi. Sistem itu adalah program peranti lunak yang diberi nama *Equire*. Dengan program itu, Berners-Lee berhasil menciptakan jaringan terkait antara berbagai arsip sehingga memudahkan informasi yang dibutuhkan. Inilah yang kemudian menjadi dasar dari sebuah revolusi yang dikenal sebagai *web*.

WWW dikembangkan pertama kali di Pusat Penelitian Fisika Partikel Eropa (CERN), Jenewa, Swiss. Pada tahun 1989 Berners-lee membuat proposal untuk proyek pembuatan hypertext secara global, kemudian pada bulan Oktober 1990, 'World Wide Web' sudah bisa dijalankan dalam lingkungan CERN. Pada musim panas tahun 1991, WWW resmi digunakan secara luas pada jaringan Internet.

2.3.5. Pengertian *Hyper Text Markup Language* (HTML)

HTML Merupakan suatu script dimana kita bisa menampilkan informasi dan daya kreasi kita lewat Internet. HTML memiliki perbedaan dengan dokumen teks biasa, misalnya : perbedaan yang paling men-colok adalah pada dokumen word, banyaknya karakter akan terbatas oleh besarnya kertas. Sedangkan HTML tak memiliki batasan teks (Rika Wulandari,2012).

Hypertext Markup Language (HTML) adalah bahasa standar dalam menulis halaman *web*, HTML merupakan pengembangan dari standar pemformatan dokumen teks. HTML sebenarnya adalah dokumen dalam bentuk ASCII atau teks biasa yang dapat diterjemahkan oleh browser web menjadi suatu halaman yang menarik (Tonni Limbong,2007).

HTML adalah Bahasa pemrograman yang fleksibel di mana kita bisa meletakkan skrip dari bahasa pemrograman lain seperti JAVA, *Visual Basic*, C++ dan lain-lain.jika HTML tersebut tidak dapat mendukung suatu perintah pemrograman tertentu. *Browser* tidak akan menampilkan kotak dialog "Syntax Error" jika terdapat penulisan kode yang keliru pada skrip HTML sepanjang kode-kode yang kita tuliskan merupakan kode-kode HTML tanpa penambahan kode-kode dari luar seperti java. Oleh karena itu,jika terjadi *syntax error* pada skrip HTML, efek yang paling jelas adalah HTML tersebut tak akan ditampilkan pada halaman jendela *Browser* (Rika Wulandari,2012).

Elemen pada HTML dapat didefinisikan sebagai suatu kode tertentu yang akan menyediakan tempat untuk meletakkan beberapa kode di dalamnya. Berbeda dengan Tag yang menangani satu kode saja.Untuk lebih jelasnya perhatikan skema elemen di bawah ini (Rika Wulandari,2012).

Struktur umum dokumen HTML adalah sebagai berikut:


```

<html>
  <head>
    <title>Judul Website</title>
  </head>
  <body>
    Isi Website
  </body>
</html>

```

Berikut ini adalah *tag-tag* HTML yang umum digunakan:

Tag Utama		
Tag	Atribut	Deskripsi
<html> </html>		Baris paling atas dari setiap file HTML
<head> </head>		Informasi umum dari sebuah halaman web
<title> </title>		Judul halaman. Terdapat pada head
<body> </body>	background bgcolor bgsound font link alink vlink topmargin leftmargin marginheight marginwidth	Settingan atribut untuk seluruh dokumen.

Gambar 2.4.Tag Utama

Modifikasi Teks		
Tag	Deskripsi	Contoh
 	Teks tebal	Teks tebal
<i> </i>	Teks miring	<i>Teks miring</i>
<u> </u>	Teks garis bawah	<u>Teks garis bawah</u>
<pre> </pre>	Preformatted teks	Contoh teks
<h1> </h1>	Header 1	Header 1
<h2> </h2>	Header 2	Header 2
<h3> </h3>	Header 3	Header 3
<h4> </h4>	Header 4	Header 4
<h5> </h5>	Header 5	Header 5
<h6> </h6>	Header 6	Header 6
	Subscript	e.g. script

Gambar 2.5. Modifikasi Teks

Gambar		
Tag	Atribut	Deskripsi
	src alt name border height width	Menampilkan sebuah gambar

Gambar 2.6. Gambar dan Formating

Formatting		
Tag	Deskripsi	Contoh
<blockquote> </blockquote>	Digunakan untuk mengatur text dan gambar dalam suatu tag	Sample of text in a block quote format
 	Ordered List (digunakan dengan)	1. Item 1
 	Unordered List (digunakan dengan)	• Item 1
	Elemen List	
<dd> </dd>	Definition List	Sample text
<dt>	Definition Term	Sample text
<dd>	Definition Description	Sample text
<p> </p>	Paragraf	Sample
 	Ganti baris	text Sample text
<hr>	Garis horizontal	
<center> </center>	Menengahkan elemen	Sample

Gambar 2.7. Formatting

Font		
Tag	Atribut	Deskripsi
 	color size name	Mengubah gaya suatu huruf.

Links		
Tag	Atribut	Deskripsi
<a> 	href target style class name id	Membuat link ke dokumen atau situs lainnya.

Gambar 2.8. Font dan Links

Tabel		
Tag	Atribut	Deskripsi
<code><table> </table></code>	border cellpadding cellspacing width height name id title bgcolor background align valign	Mengatur semua elemen table
<code><tr> </tr></code>	height bgcolor background align valign title	Membuat baris baru
<code><td> </td></code>	height width bgcolor background align valign title colspan rowspan	Membuat kolom
<code><th> </th></code>	height width bgcolor background align valign title colspan rowspan	Header(kepala tabel). Otomatis ke tengah dan tebal.
<code><tbody> </tbody></code>	height width align valign bgcolor background	Format yang berlaku bagi cell yang diapit tag.
<code><colgroup> </colgroup></code>	height width align valign bgcolor background colspan	Format yang berlaku bagi kolom

Gambar 2.9. Tabel

Form		
Tag	Atribut	Deskripsi
<form> </form>	method action name	Mengatur elemen dari form
<input type=>	text password hidden radio checkbox submit image reset	Variasi dari tipe elemen form
text	name value width maxlength	<input type="text" value="Sample text"/>
password	name value width maxlength	<input type="password" value="*****"/>
hidden	name value	Digunakan untuk membuat variable hidden.
radio	name value	<input type="radio"/> Radio 1 <input type="radio"/> Radio 2
checkbox	name value checked	<input type="checkbox"/> Check 1 <input type="checkbox"/> Check 2
submit	value	<input type="submit" value="Tombol"/>
image	src height width alt name border	
reset	value	<input type="reset" value="Reset"/>
<select> </select>	name size	Membuat combo-box. Digunakan bersama dengan option
<option>	selected name value	<input type="text" value="Option 1"/>
<textarea> </textarea>	name rows cols wrap	<pre>Text area box. Text is typed between the tags</pre>
wrap	off virtual physical	no wrap word wrap, sent as one line word wrap, sent with breaks

Gambar 2.10. Form

2.3.6. Pengertian PHP

PHP Menurut Antonius Nugraha (2010:19), PHP adalah kependekan dari *HyperText Preprocessor* yang dibangun oleh Rasmus Lerdorf pada awal pengembangan PHP disebut sebagai kependekan dari *Personal Home Page*. PHP merupakan produk *open source* sehingga kita dapat mengakses *source code*, menggunakan, dan mengubahnya tanpa harus membayar sepeserpun.

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya hasilnya saja. PHP mampu berjalan diatas beberapa *platform* seperti *Windows*, *Unix* serta varian *Linux*. Beberapa kelebihan PHP antara lain dapat membuat situs yang interaktif dengan forum diskusi, *guestbook* dan sebagainya, koneksitas yang baik dengan bermacam-macam *database* seperti *Oracle*, *PostgreSQL*, dan lain-lain.

2.3.7. Pengertian MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak seperti PHP atau *Apache* yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius (Achmad Solichin, 2010:8).

Fitur-Fitur MySQL antara lain

1. *Relational Database System*. Seperti halnya software database lain yang ada di pasaran, MySQL termasuk RDBMS.

2. Arsitektur *Client-Server*. MySQL memiliki arsitektur client-server dimana server database MySQL terinstal di server. Client MySQL dapat berada di komputer yang sama dengan server, dan dapat juga di komputer lain yang berkomunikasi dengan server melalui jaringan bahkan internet.
3. Mengenal perintah SQL standar. SQL (*Structured Query Language*) merupakan suatu bahasa standar yang berlaku di hampir semua software database. MySQL mendukung SQL versi SQL:2003.
4. Mendukung *Sub Select*. Mulai versi 4.1 MySQL telah mendukung *select* dalam *select (sub select)*.
5. Mendukung Views. MySQL mendukung views sejak versi 5.0
6. Mendukung Stored Prosedured (SP). MySQL mendukung SP sejak versi 5.0
7. Mendukung Triggers. MySQL mendukung trigger pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan trigger pada versi 5.1. Mendukung *replication*.
8. Mendukung transaksi.
9. Mendukung *foreign key*.
10. Tersedia fungsi GIS.
11. Stabil dan tangguh.
12. Fleksibel dengan berbagai pemrograman.
13. Security yang baik.
14. Dukungan dari banyak komunitas.
15. Perkembangan software yang cukup cepat.

Menurut Bunafit Nugroho (2005:29), SQL merupakan singkatan dari *Structured Query Language*, SQL merupakan bahasa komputer standard *ANSI*, Selanjutnya dengan menggunakan SQL kita dapat mengakses *database*, menjalankan *query* untuk mengambil data dari *database*, menambahkan data ke *database*, menghapus data di dalam *database*, dan meng-*update* data di dalam *database*.

Pernyataan SQL dapat dikelompokkan menjadi beberapa kelompok, yakni :

1. DDL

DDL adalah bahasa DBMS yang digunakan untuk membuat atau mendefinisikan obyek-obyek di dalam *database*. Statement DDL adalah perintah-perintah yang digunakan untuk menjelaskan objek dari *database*, Dengan kata lain DDL digunakan untuk mendefinisikan kerangka *database*. Secara umum membuat Obyek *table*.

2. DML

DML merupakan kelompok perintah yang berfungsi untuk memanipulasi data dalam basis data, misalnya untuk pengambilan, penyisipan, pengubahan, dan penghapusan data. Perintah yang termasuk dalam kelompok DML adalah *Select, Insert, Delete, dan Update*.

2.3.8. Pengertian Macromedia Dreamweaver 8

Versi terbaru *Macromedia Dreamweaver* di tahun 2005 adalah *Macromedia Dreamweaver 8* yang merupakan penyempurnaan dari versi sebelumnya dan tentu saja semakin mudah penggunaannya. Fasilitas terbaru dari *Macromedia Dreamweaver 8* adalah *Zoom Tool and Guides*, Panel CSS yang baru, *Code collapse*, *Coding Toolbar* dan *Insert Flash Video*. Tidak jauh beda dengan kemampuan versi sebelumnya, *Macromedia Dreamweaver 8* mendukung pemrograman *Client Side* yang terkenal, yaitu *JavaScript* dengan penggunaan yang sangat mudah. *Macromedia dreamweaver 8* juga mendukung pemrograman *ScriptServer Side*, seperti *Personal Home Page (PHP)*, *Active Server Pages (ASP)*, *ASP.NET*, *ColdFusion* dan *Java Server Pages (JSP)* (Nurwanyah, 2010). *IDE (Integrated Development Environment)* atau area kerja *Adobe Dreamweaver*.

2.3.9. Pengertian CorelDraw12

CorelDraw 12 merupakan aplikasi grafis yang fenomenal. Kemampuannya yang sangat fleksibel dalam mengolah objek membuat aplikasi ini begitu dikenal sehingga pengguna komputer terbiasa mendengar keunggulan aplikasi ini dan mungkin juga telah menjadikannya default pekerjaan grafis (Edi S Mulyanta, 2004:1).

2.3.10. Pengertian *Adobe Photoshop CS 5*

Adobe Photoshop CS5 merupakan *software* yang dapat digunakan untuk editing dan manipulasi foto. Kelengkapan fasilitas dan kemampuannya yang luar biasa dalam editing foto atau gambar, menjadikan software ini paling banyak dipakai fotografer dan desainer grafis (Madcoms,2011:2).

2.3.11. Analisis Perancangan Berorientasi Objek

Pemrograman berorientasi objek adalah suatu cara baru dalam berpikir serta berlogika dalam menghadapi masalah-masalah yang akan dicoba diatasi dengan bantuan komputer. Tidak seperti pendahulunya (pemrograman terstruktur), pemrograman berorientasi objek mencoba melihat permasalahan lewat pengamatan dunia nyata dimana setiap objek adalah entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu (Adi Nugroho,2002:1). Orietasi objek adalah sebuah pendekatan untuk mengembangkan software sistem yang berdasarkan pada item data dan atribut dan operasi operasi yang menjelaskanya.

Dalam analisis dan perancangan metodologi berorientasi objek menggunakan diagram UML. UML adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (Martin Flower, 2005).

Kelebihan dari perancangan berorientasi objek adalah sebagai berikut :

- a) *Maintainable* : pemeliharaan
- b) *Testable* : dicoba
- c) *Reuuseable* : dapat digunakan kembali


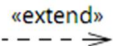
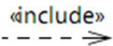



Albe to cope with large and complex system : dapat berkerjasama dengan system yang luas dan kompleks.

2.3.11.1 *Use Case Diagram*

Use Case Diagram adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan (Martin Fowler, 2005:141). *Use Case*

Diagram dibuat untuk memvisualisasikan atau menggambarkan hubungan antara *Actor* dan *Use Case*. Simbol *use case* dapat dilihat pada Tabel 2.1:

Tabel 2.1. Simbol-Simbol *Use Case*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Association</i>	Jalur komunikasi antara aktor dan use case yang berpartisipasi
2		<i>Extent</i>	<i>Insertion</i> tambahan ke <i>Usecase</i> yang tidak diketahui
3		<i>Class</i>	<i>Insertion</i> tambahan ke <i>Usecase</i> yang secara explicit menggambarkan <i>insertion</i> .
4		<i>Use case generalization</i>	Hubungan antara <i>use case</i> satu dengan <i>use case</i> yang lainnya.
5		<i>Actor</i>	Pengguna di luar sistem
6		<i>Use case</i>	Sebuah spesifikasi dari perilaku sebuah entitas dalam interaksinya dengan agen luar

2.3.11.2 *Class Diagram*

Class Diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class Diagram* juga menunjukkan properti dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut (Martin Fowler, 2005:53).

Diagram kelas mempunyai 3 macam *relationships* (hubungan), sebagai berikut :

a. *Association*

Suatu hubungan antara bagian dari dua kelas, terjadi *association* antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Di dalam diagram, sebuah *association* adalah penghubung yang menghubungkan dua kelas.






b. *Aggregation*

Suatu *association* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan, *aggregation* memiliki titik pusat yang mencakup keseluruhan bagian sebagai contoh: *Order detail* merupakan kumpulan dari order

c. *Generalization*

Suatu hubungan turunan dengan mengansumsikan satu kelas merupakan suatu *super class* (kelas super) dari kelas yang lain. *Generallization* memiliki tingkatan yang berpusat pada *super class*, contoh: *Payment* adalah *superclass* dari *Cash, Check, Credit*. Simbol *Class Diagram* dapat dilihat pada Tabel 2.2.





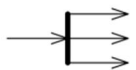
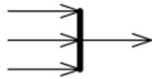
Tabel 2.2 Simbol-Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
2		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
4		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri

2.3.11.3 Activity Diagram

Activity diagram adalah teknik untuk menggambar logika *prosedural*, proses bisnis, dan jalur kerja, dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung *behavior* paralel (Martin Fowler, 2005:163). Simbol *Activity Diagram* dapat dilihat pada Tabel 2.3.

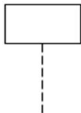
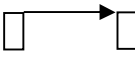


Tabel 2.3. Simbol-Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Decision node</i>	suatu titik/point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6		<i>Join node</i>	Beberapa aliran masukan tertentu berubah menjadi satu aliran

2.3.11.4 Sequence Diagram

Sequence Diagram adalah penjabaran behavior sebuah skenario tunggal, *Sequence Diagram* menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini didalam use case (Martin Fowler, 2005:81). Simbol *Sequence Diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol-Simbol *Sequence Diagram*





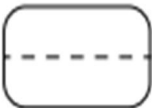



NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Actor</i>	Pengguna di luar system
4		<i>Fragment</i>	Menggambarkan batas grafis suatu <i>diagram</i>

2.3.11.5 *Statechart Diagram*

Statechart Diagram adalah teknik yang umum digunakan untuk menggambar behavior sebuah sistem (Martin Fowler, 2005:151). *Statechart Diagram* menggambarkan semua state atau kondisi yang dimiliki oleh suatu object dari class dan kejadian yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan.

Statechart Diagram menurut Rosa Ariani Sukamto (2010) menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek, Status awal (kondisi awal alur hidup objek dan setiap state diagram hanya memiliki satu status awal), Status akhir (kondisi akhir alur hidup objek), Transisi (garis transisi antar status pada alur hidup objek). Simbol *Statechart Diagram* dapat dilihat pada Tabel 2.5.





Tabel 2.5. Simbol-Simbol *Statechart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Transition</i>	Sebuah kejadian yang memicu sebuah <i>state</i> objek dengan cara memperbaharui satu atau lebih nilai atributnya
4		<i>Final State</i>	Kondisi akhir alur hidup objek
5		<i>Concurrent composite state</i>	sebuah state yang dibagi menjadi dua atau lebih concurrent sub state, semua dari concurrently aktif ketika <i>composite state</i> aktif.
6		<i>Junction state</i>	Sebuah <i>pseudostates</i> egmen rantai transisi dalam <i>single run</i> untuk menyelesaikan transisi
7		<i>Suquential composite state</i>	<i>State</i> yang berisi satu atau lebih <i>substate</i> , tepatnya salah satunya aktif pada satu waktu ketika <i>composite state</i> aktif
8		<i>History state</i>	Sebuah <i>pseudostate</i> mengembalikan keadaan <i>state</i> sebelumnya di dalam <i>compositestate</i>

2.3.11.6 Component Diagram

Component Diagram merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada dianalisis. *Component* terhubung melalui antar muka yang digunakan dan dibutuhkan (Martin Fowler, 2005:189). *Component* merupakan implementasi *software* dari sebuah atau lebih class. *Component* dapat berupa *sourcecode*, komponen biner, atau *executable component*. Sebuah komponen berisi informasi tentang logic class atau class yang diimplementasikan sehingga membuat pemetaan dari *logicalview* ke *component view*. Sehingga *component diagram* mempresentasikan dunia riil yaitu *component software* yang mengandung *component*, *interface* dan *relationship*. Simbol *Component Diagram* dapat dilihat pada Tabel 2.6.

Tabel 2.6 Simbol-Simbol *Component Diagram*

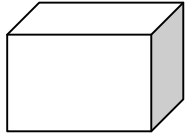
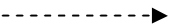

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Component</i>	<i>Physical</i> dari sebuah sistem
2		<i>Interface</i>	A <i>name state</i> informasi yang menjadi ciri dari perilaku
3		<i>Usage</i>	situasi di mana satu elemen membutuhkan lain untuk fungsi yang benar
4		<i>Realization</i>	hubungan antara spesifikasi dan implementasinya

2.3.11.7 Deployment Diagram

Deployment Diagram menunjukkan susunan fisik sebuah sistem, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras mana (Martin Fowler, 2005:137). *Deployment Diagram* juga menggambarkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan

perangkat (*nodes*) satu sama lain dan jenis hubungannya. Simbol *Deployment Diagram* dapat dilihat pada Tabel 2.7.

Tabel 2.7 Simbol-Simbol *Deployment Diagram*

NO	GAMBAR	NAMA	PENJELASAN
1		<i>node</i>	Biasanya mengacu pada perangkat keras (<i>hardware</i>) dan perangkat lunak (<i>software</i>), jika didalam node disertakan komponen untuk mengkonsistensikan rancangan maka komponen yang di ikutsertakan harus sesuai dengan komponen yang sudah didefinisikan sebelumnya pada diagram komponen
2		<i>Dependency</i>	Sebuah <i>dependency</i> digunakan untuk menotasikan relasi antara dua <i>component</i>
3		<i>link</i>	Relasi antar <i>node</i>

2.3.12. Pengertian Internet

Internet merupakan media komunikasi yang menggunakan komputer dan saluran telekomunikasi sebagai tulang punggungnya. Seperti halnya sebuah terminal telepon, lewat sebuah terminal komputer yang tersambung ke internet kita dapat menghubungi rekan kita dimana saja yang juga tersambung ke internet. Akan tetapi internet memiliki beberapa kelebihan dibandingkan telepon atau media komunikasi lainnya, dari segi biaya internet tidak memperhitungkan jarak seperti percakapan di telpon (Akhmad Fauzi,2008:334).