BAB II

LANDASAN TEORI

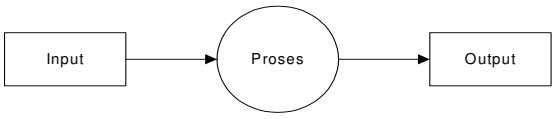
2.1 Definisi Perancangan Sistem

Perancangan sistem informasi Akademik di Akper RSHS. Perancangan itu sendiri dapat diartikan dengan membuat sesuatu hal atau komponen, yang dimulai dari titik nol sampai menjadi sesuatu yang sangat berguna. Perancangan sistem informasi adalah merancang suatu sistem informasi di dalam perusahaan atau institusi guna untuk melakukan peningkatan pemasaran di Instansi atau organisasi tersebut.

2.2 Pengertian Sistem

Pengertian sistem pertama kali dapat diperoleh dari definisi sistem itu sendiri, pendekatan sistem memberikan banyak manfaat dalam memahami lingkungan. Pendekatan sistem berusaha menjelaskan sesuatu yang dipandang dari sudut pandang sistem serta berusaha menemukan struktur unsur sistem dan proses sistem.

Maka dapat diambil kesimpulan bahwa suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.Model umum sebuah sistem terdiri dari masukan, pengolah dan keluaran. Adapun model umum sistem :



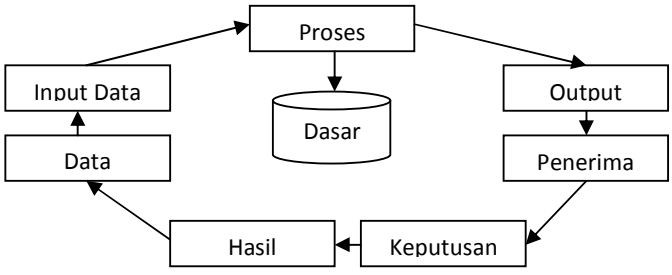
Gambar 2.1 Model Umum Sistem

2.3 Pengertian Data

Data merupakan sumber dan bahan informasi, oleh karena itu data dan informasi saling berkaitan. Data adalah kenyataan yang menggambarkan suatu kejadian serta merupakan suatu kesatuan yang nyata, merupakan bentuk yang masih mentah sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi.

2.4 Pengertian Informasi

Pengertian sistem menurut Jogiyanto (2005 : 15) "Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu". Fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Data diolah melalui suatu model informasi. Si penerima akan menerima informasi tersebut untuk membuat suatu keputusan dan melakukan tindakan yang akan mengakibatkan munculnya sejumlah data lagi. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model, dan seterusnya sehingga membentuk suatu siklus. Siklus inilah yang disebut sebagai *siklus informasi* (*Information Cycle*). Agar lebih jelas, dapat dilihat pada Gambar 2.2 berikut :



Gambar 2.2 Siklus Informasi

2.5 Pengertian Sistem Informasi

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertahankan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.6 Sistem Informasi Akademik

Menurut Salim (2005 : 12) sistem informasi akademik merupakan "tiang utama dalam mengatur segala hal yang berkaitan dengan penyelenggaraan perkuliahan maupun hal-hal yang lainnya". Dan akademik yaitu "suatu rangkaian kegiatan yang disusun secara sistematis untuk mahasiswa sebagai input agar dapat menghasilkan lulusan yang berkualitas". Sebuah perguruan tinggi tentunya memiliki sistem informasi pada instansi pendidikan, serta komponen-komponen yang terkandung didalamnya, kemudian kaitannya masing-masing komponen tersebut pada alur sistem informasi lembaga pendidikan tertinggi.

Sistem informasi ini akan mendukung kinerja perguruan tinggi yang bersangkutan baik dalam pelayanan terhadap mahasiswa sampai karyawan. Namun sesudah perkembangan teknologi informasi yang sedemikian pesatnya, perguruan tinggi harus setiap saat mengupdate sistem informasinya sehingga dalam kinerjanya akan menuju ke titik yang lebih baik.

2.7 Pengenalan UML (Unified Modeling Language)

UML merupakan "pengganti dari metode analisa berorientasi *object* dan desain berorientasi *object* (OOA&D) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an.

UML merupakan gabungan dari metode Booch,Rumbaugh (OMT) dan Jacobson. Tetapi UML ini akan mencakup lebih luas dari pada OOA&D. Pengembangan UML dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan sebagian besar merupakan notasi dari metode yang digunakan untuk mendesain secara cepat.

Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi, jika anda ingin berdiskusi tentang desain dengan seseorang, maka anda hanya membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendesain secara cepat.

UML merupakan bahasa standar untuk penulisan *blue print software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak.

2.7.1 Pengertian UML

Menurut Yahya (2009 : 10) UML adalah bahasa untuk menspesifikasi, menvisualiasasi, membangun dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti dalam pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep berorientasi *object*. UML dibuat oleh *Grady Booch*, *James Rumbaugh* dan *Ivar*

Jacobson dibawah bendera *rational Software Corp*. UML menyediakan notasi-notasi yang membantu memodelkan perangkat lunak, namun hampir dalam sebuah bidang yang membutuhkan pemodelan.

2.7.2 Bagian – bagian UML

Bagian-bagian utama dari UML adalah *view, diagram, model element* dan *general mechanism.*

1. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *view* dalam UML antara lain :

a. *Use Case* View

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakuan sesuai dengan yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa user atau sistem yang lainnya. *View* ini digambarkan dalam *Use Case diagram* dan kadang-kadang dengan *activity diagram*. *View* ini digunakan terutam untuk pelanggan, perancang (*designer*), pengembang (*developer*) dan penguji sistem (*tester*).

b. Logical View

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class, object* dan *relationship*) kolaborasi dinamis yang terjadi ketika *object* lain dalam suatu fungsi tertentu. *View* ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state, sequence*

collaboration dan activity diagram untuk model dinamisnya. *View* ini digunakan untuk perancangan (*designer*) dan pengembangan(*developer*).

c. Component View

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi semberdaya komponen dan informasi administratif lainnya. *View* ini digambarkan dalam *component view* dan digunakan untuk pengembangan (*developer*)

d. Concurrency View

Membagi sistem kedalam proses dan prosesor. *View* ini digambarkan dalam diagram dinamis (*state, sequence, collaboration* dan *activity diagrams*) dan diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

e. Deployment View

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan yang lainnya. *View* ini digambarkan dalam *deployment diagrams* dan digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

2. Diagram

Diagram berbentuk grafik yang menunjukan elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem.

Sebuah diagram merupakan bagian dari view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu.

Adapun jenis diagram antara lain :

a. *Use Case* Diagram

Menggambarkan sejumlah *ekseternal actors* dan hubungannya ke *Use Case* yang diberikan oleh sistem. *Use Case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dari dokumentasi dari *Use Case symbol* namun juga dapat dilakukan dalam *activity diagrams*. *Use Case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

b. *Class* Diagram

Menggambarkan struktur statis *class* di dalam sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara : *associated* (terhubung satu sama lain), *depended* (satu *class* tergantung atau menggunakan *class* yang lain), *specialed*(satu *class* merupakan spesialisasi dari *class* yang lainnya), atau *packed* (group bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram.*

c. State Diagram

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan, *state class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state*

yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

d. Sequence Diagram

Menggambarkan kolaborasi dinamis atara sejumlah *object*. Kegunaanya untuk menunjukan sejumlah pesan yang dikirim antara *object* juga interaksi antar *object* , sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

e. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukan pertukaran pesan, *collaboration diagram* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekanannya pada waktu atau urutan, gunakan *sequence diagram* tapi jika penekanannya pada konteks gunakan *collaboration diagrams*.

f. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *Use Case* atau interaksi.

g. Component Diagram

Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *source code*, komponen biner atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.

h. Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Didalam *nodes, executable* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

2.7.3 Indikator Multicitiply

Walaupun *multiplicity* ditentukan untuk *class*, *multiplicity* menentukan banyaknya *object* yang terlibat dalam relasi. *Multiplicity* menentukan banyaknya *object* yang terhubung satu denagn yang lainnya. *Indikator multiplicity* terdapat pada masing-masing akhir garis relasi baik pada asosiasi maupun agrerasi. Beberapa contoh *multiplicity* adalah :

a. 1 tepat satu
b. 0..* nol atau lebih
c. 1..* satu atau lebih
d. 0..1 nol atau satu
e. 5..8 range 5 s.d 8
f. 4..6,9 rang 4 s.d 6 dan 9

2.8 Pengertian Internet

Internet adalah sumber daya informasi yang menjangkau informasi ke seluruh dunia. Internet berasal dari kata *Interconnection Networking* yang mempunyai arti hubungan berbagai komputer dengan berbagai tipe yang membentuk sistem jaringan yang mencakup seluruh dunia (jaringan komputer global) dengan melalui jalur telekomunikasi seperti telepon. Internet adalah kumpulan yang luas dari jaringan komputer besar dan kecil yang saling bersambungan menggunakan jaringan telekomunikasi yang ada di seluruh dunia.

Setiap komputer pada sebuah jaringan harus dapat berkomunikasi satu sama lain. Ini diwujudkan melalui sebuah protokol, yaitu seperangkat aturan atau kesepakatan mengenai cara berkomunikasi lewat Internet. Bahasa standar Internet adalah TCP/IP (*Transmission Control Protocol/ Internet Protocol*). Banyak protokol yang tersedia, seperti *DECnet, SNA, Novell dan Appletalk*. Namun, agar dua buah komputer dapat berkomunikasi keduanya haruslah menggunakan protokol yang sama pada saat yang bersamaan.

2.9 World Wide Web (WWW)

World Wide Web (WWW) merupakan suatu kumpulan informasi pada beberapa *server* komputer yang terhubung satu sama lain dalam jaringan internet. Informasi-informasi dalam web mempunyai *link-link* yang menghubungkan informasi tersebut ke informasi lain didalam jaringan internet untuk digunakan bersama.

Web pada awalnya adalah ruang informasi dalam internet, dengan menggunakan teknologi *hypertext*, pemakai dituntun untuk menemukan informasi

dengan mengikuti link yang disediakan dalam dokumen web yang ditampilkan dalam *browser* web.

Web memudahkan pengguna komputer berinteraksi dengan pelaku internet lainnya dan menelusuri informasi di internet.

2.10 HTML

HTML singkatan dari *Hyper Text Markup Language* merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web menurut Bimo Sunarfrihanto, ST, (2002 :1). Dokumen HTML adalah *file* teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai *web page.*

Dokumen HTML disusun oleh elemen-elemen. "Elemen" merupakan istilah bagi komponen-komponen dasar pembentuk dokumen HTML. Beberapa contoh elemen adalah: *head, body, table, paragraph,* dan *list.* Elemen dapat berupa teks murni, atau bukan teks, atau keduanya. Penandaan berbagai elemen dalam suatu dokumen HTML menggunakan tag (markup).

Format penulisan tag terdiri dari tanda "<", ">", "/" yaitu:

<namatag> … </namatag>

Penulisan tag umumnya berpasangan

Contoh:

 → untuk mengawali suatu paragraph

 → untuk mengakhiri suatu paragraph

Setiap tag dapat dilengkapi dengan attribut, attribut setiap tag berbeda-beda.

Format penulisan:

<nametag atr1=”nilai-atr1” atr2=”nilai-atr2”> … </namatag>

Struktur file HTML

```
<html>
    <head>
        <title> … </title>
    </head>
    <body>
        …….
        …….
        …….
    </body>
</html>
```

Setiap tag HTML harus diawali dengan menuliskan tag <html> dan tag </html> di akhir dokumen. Tag ini menandakan elemen HTML, yang berarti dokumen ini adalah dokumen HTML. Dalam satu dokumen hanya ada satu elemen HTML.

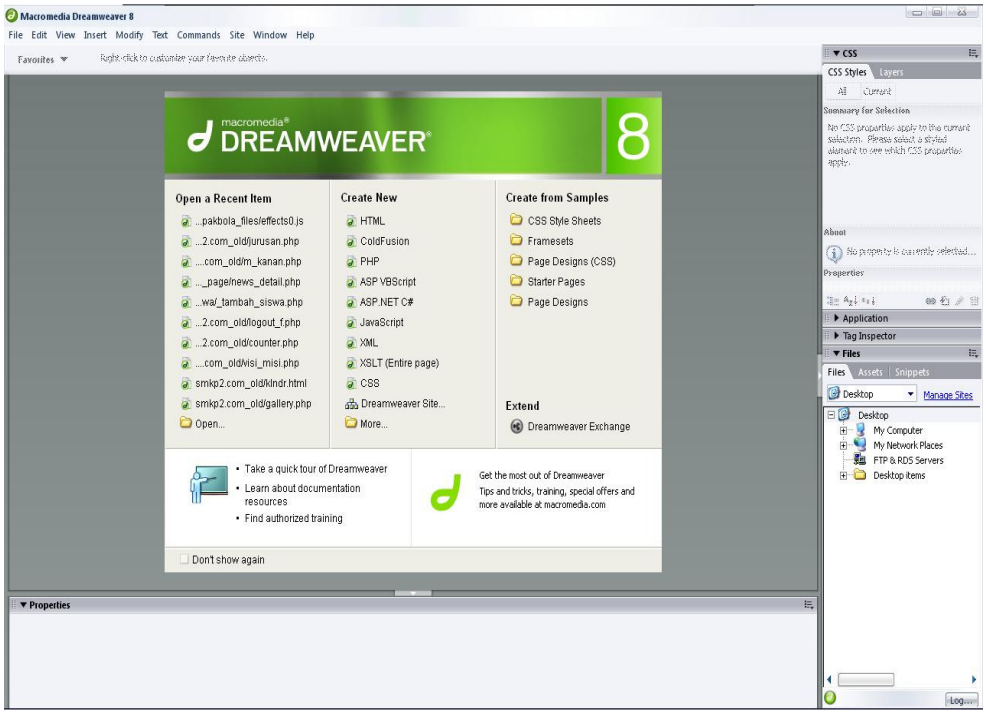
Section atau elemen head ditandai dengan tag <head> di awal, dan tag </head> di akhir. *Section* ini berisi informasi tentang dokumen htmlnya. Minimal informasi yang dituliskan dalam elemen ini adalah judul dari dokumen, judul ini akan ditampilkan pada *caption bar* dari *window browser*, ditandai dengan menggunakan tag <title> dan diakhiri dengan </title>.

Section atau elemen body ditandai dengan tag <body> di awal, dan tag </body> di akhir. *Section body* merupakan elemen terbesar di dalam dokumen html. Elemen ini berisi isi dokumen yang akan ditampilkan pada browser, meliputi paragraph, grafik, link, table, dan sebagainya.

2.11 Program Editor HTML

Ada dua macam editor HTML, yaitu yang berbasiskan teks dan yang berbasiskan WYSIWYG *(What You See Is What You Get).* Untuk editor teks, diperlukan browser untuk melihat hasil dari dokumen HTML yang telah dibuat. Sedangkan untuk editor tersebut sudah menyediakan fasilitas untuk melihat langsung dari dokumen HTML yang telah dibuat.

Program editor HTML berbasiskan teks yang dapat digunakan diantaranya : Notepad, WebEdit, sedangkan program editor HTML berbasiskan WYSIWYG diantaranya adalah Frontpage, Homesite, Macromedia Dreamweaver 8, berikut ini adalah tampilan dari editor Macromedia Dreamweaver 8 :



Gambar 2.3 Program Editor HTML Macromedia Dreamweaver 8

2.12 Hypertext Transfer Protocol (HTTP)

Server HTTP umumnya digunakan untuk melayani dokumen *hypertext*, karena HTTP adalah protokol dengan *overhead* yang sangat rendah, sehingga pada kenyataan navigasi informasi dapat ditambahkan langsung ke dalam dokumen dan dengan demikian protokolnya sendiri tidak harus mendukung navigasi secara penuh seperti halnya protokol FTP dan Gopher.

2.12.1 Uniform Resource Locator (URL)

Konsep nama file standard yang diperluas dengan jaringannya. Nama file ini tidak hanya menunjukkan direktori dan nama filenya, tetapi juga nama mesinnya dalam jaringan. URL dapat disediakan (ada atau diakses) dengan berbagai metode, dan bisa jadi bukan sekedar file, karena URL dapat menunjukkan query-query, dokumen yang disimpan dalam database, hasil dari

suatu perintah *finger* atau *archie*, atau apapun yang berkaitan dengan data hasil proses.

2.12.2 Domain Name System

Komputer-komputer di Internet menggunakan suatu format penamaan standar untuk mempermudah pengelolaan server komputer di Internet yang berkembang dengan cepat. Sistem penamaan server komputer ini adalah *Domain Name System* (DNS). DNS merupakan suatu tingkat-tingkat domain, yang merupakan kelompok komputer-komputer yang terhubung ke Internet.

Sebagai contoh, dalam penulisan URL terdapat akhiran .com, itu berarti menandakan sebuah organisasi komersial. Contoh lain misalnya .gov yang menandakan lembaga pemerintahan dan masih banyak lagi yang lainnya.

2.12.3 Homepage

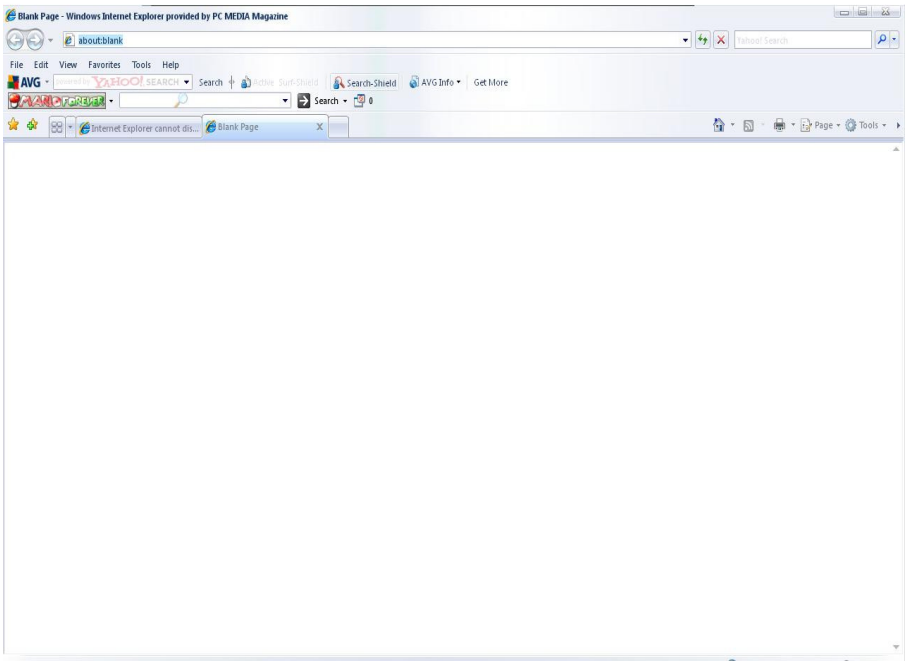
Homepage adalah page pembuka yang akan pertama ditemui sebelum mengakses informasi lainnya pada suatu website. *Homepage* ini merupakan halaman pertama dari suatu website yang biasanya berisi tentang apa dan siapa dari perusahaan atau organisasi pemilik website tersebut.

Dari *homepage* ini, informasi lainnya dapat ditemui pada page-page berikutnya yang tersimpan, yang telah dilink untuk menghubungkan suatu informasi lainnya, baik didalam suatu web page yang sama, ataupun dalam web page lain pada website yang berbeda.

2.12.4 Web Browser

Web browser adalah software yang digunakan untuk menampilkan informasi dari web server. Software tersebut kini telah dikembangkan dengan menggunakan user *interface* grafis, sehingga pemakai dapat dengan melakukan

"*point dan click*" untuk pindah antar dokumen, seperti Internet Explorer 7, Firefox, Opera.

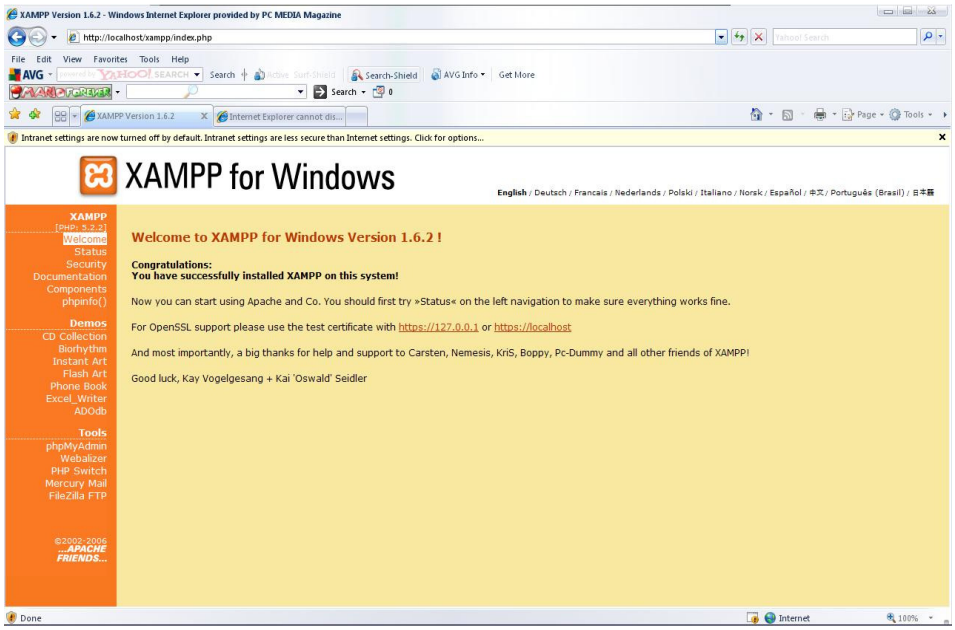


Gambar 2.4 Web Browser Internet Explorer 7

2.12.5 Web Server

Web server adalah komputer yang digunakan untuk menyimpan dokumen-dokumen web, komputer ini akan melayani permintaan dokumen web dari kliennya.

Web browser seperti Explorer atau Navigator berkomunikasi melalui jaringan (termasuk jaringan Internet) dengan web server, menggunakan HTTP. Browser akan mengirimkan request kepada server untuk meminta dokumen tertentu atau layanan lain yang disediakan oleh server. Server memberikan dokumen atau layanannya jika tersedia juga dengan menggunakan protokol HTTP.



Gambar 2.5 Web Server XAMPP 1.6.2

2.13 Tinjauan Perangkat Lunak

Pada bagian ini penulis akan menguraikan penjelasan singkat mengenai perangkat lunak (*software*) utama yang digunakan yaitu Pemograman PHP dan MySQL sebagai database server.

2.13.1 Pemograman PHP

PHP adalah kependekan dari *PHP Hypertext Prepossesor*, bahasa interpreter yang mempunyai kemiripan dengan bahasa C dan Perl yang mempunyai kesederhanaan dalam perintah. PHP merupakan bahasa scripting yang menyatu dengan HTML dan berada di server *(server-side HTML-embedded scripting)*, artinya sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di server tetapi disertakan pada halaman HTML biasa. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi-aplikasi yang dijalankan diatas teknologi web.

"PHP adalah bahasa server-side scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Maksud dari server-side scripting adalah sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di server tetapi disertakan pada dokumen HTML." Menurut Bimo Sunarfrihanto, ST, (2002 :18)

2.13.1.1 Kelebihan PHP

PHP memiliki kelebihan yang tidak dimiliki oleh bahasa-bahasa pemrograman sejenisnya, diantaranya adalah :

1. PHP mudah dibuat dan dijalankan, maksudnya PHP dapat berjalan dalam web server apache, *Internet Informatin Service* (IIS), *Personal Web Server* (PWS), XAMPP dan lainnya, serta dalam Sistem Operasi yang berbeda pula seperti Windows dan Unix.
2. PHP bersifat efisien, karena hanya memerlukan resource sistem yang sangat sedikit dibandingkan dengan bahasa pemrograman lainnya.
3. PHP dapat menggunakan beberapa database seperti MySQL, Sybase, Interbase, Oracle, Ms. SQL Server, Microsoft Access, Dbase, dll.

2.13.1.2 Sintak Dasar PHP

PHP adalah bahasa yang HTML-*embedded*, artinya perintah-perintah dalam PHP dapat menyatu dengan tag-tag HTML dalam sebuah file. Sebuah program PHP tidak lebih adalah file HTML yang didalamnya kita masukkan perintah-perintah tertentu. Karena PHP adalah bahasa HTML yang embedded, maka harus ada sebuah cara untuk memisahkan bagian mana dari file tersebut yang merupakan bahasa PHP dengan HTML. Ada tiga cara untuk memisahkan PHP dengan HTML :

1. <? Echo("Script PHP"); ?>
2. <? PHP ("Script PHP"); ?>

3. <script language = “PHP”> ... </script>

Untuk menuliskan komentar digunakan “/*” dan ditutup dengan “*/” atau meletakkan tanda “#” pada awal baris. Hal yang paling penting dalam memberikan perintah PHP adalah setiap perintah PHP harus selalu diakhiri dengan tanda “;”. Dokumen yang mengandung perintah PHP harus disimpan dalam ekstensi .php atau .phtml ataupun .php3.

2.13.2 Database Server MySQL

“MySQL adalah multiuser database yang menggunakan bahasa SQL. MySQL dalam operasi client-server melibatkan server daemon MySQL disisi server dan berbagai macam program serta library yang berjalan disisi client. SQL (Structured Query Language) adalah bahasa standar yang digunakan untuk mengakses database server. Bahasa ini pada awalnya dikembangkan oleh IBM, namun diadopsi dan digunakan sebagai standar industry.” Menurut Bimo Sunarfrihanto, ST, (2001 :10).

SQL bisa digunakan dalam dua cara. Yang pertama secara interpretasi, yakni dengan memasukkan sebuah pernyataan SQL melaluiterminal atau mikrokomputer dan langsung diproses dan diinterpretasikan. Hasilnya bisa dilihat secara langsung. Cara ini dikenal juga dengan SQL interaktif (*Interactive SQL*). Cara kedua dengan menyisipkan pernyataan SQL ke dalam sebuah program yang ditulis ini tidak dapat dilihat secara langsung oleh pemakai, tetapi diproses oleh program yang memakainya. Cara ini dinamakan dengan SQL sisip *(Embedded SQL).* Alasan digunakannya MySQL sebagai database server :

1. Mendukung standar yang telah ada.
2. Dukungan terhadap berbagai bahasa pemrograman.
3. Mampu membuat tabel berukuran sangat besar.

4. Kecepatan, kemudahan dan kehandalan dalam penggunaannya.

2.13.2.1 Sintak Dasar MySQL

Sintaks dalam bahasa MySQL hampir sama dengan *Structured Query Language* (SQL). SQL adalah bahasa pemrograman standar untuk mengakses dan memanipulasi informasi dari sebuah basis data relasional.

Statemen-statemen yang ada dalam MySQL diantaranya :

1. Perintah untuk membuat database

```
Create database dbname;
```

Perintah untuk membuat tabel didalam database yang telah dibuat

```
Use dbname;

Create table table_name(field1 type_var,field2 type_var);
```

2. Perintah untuk menambah data pada tabel dalam suatu database aktif

```
Insert into table_name(field1,field2) values ('nilai1','nilai2');
```

3. Perintah untuk membuang atau menghapus database

```
Drop database dbname;
```

4. Perintah untuk menghapus satu atau lebih record dalam suatu tabel yang memenuhi kondisi yang ditentukan

```
Delete from table_name;
```

5. Perintah untuk menampilkan data pada suatu tabel

```
Select * from table_name;
```

6. Perintah untuk mengubah data yang telah ada pada tabel yang telah dibuat

```
Update dbname set field1='$field1',field2='$field2',
field3='field3', WHERE field1='$field1';
```

2.13.2.2 Integrasi PHP dan MySQL

Integrasi PHP dengan database MySQL, yang terjadi ketika *client* mengakses web server yang mendukung PHP dan database MySQL melalui browser, antara lain ketika server menerima dan membaca permintaan dari browser, server akan mengeksekusi perintah-perintah dalam PHP. Kemudian PHP akan melakukan *query* (permintaan) pada database MySQL melalui Application Programming Interface (API) dan mengkompilasi hasilnya.

PHP mengaksesdatabase MySQL dengan beberapa langkah yaitu :

1. Melakukan koneksi ke server database MySQL menggunakan fungsi

```
Mysql_connect("localhost","nama_user","password")
```

2. Setelah koneksi berhasil, maka kita dapat membuat database dengan fungsi

```
Mysql_create_db("dbname")
```

3. Untuk menyisipkan perintah internal yang terkait dengan tabel, seperti mendeteksi jumlah kolom suatu tabel, gunakan fungsi

```
$proses_query = "[pernyataan SQL]";

Mysqk_query("dbname",$proses_query);
```

4. Memilih database yang ada di server MySQL menggunakan fungsi

```
Mysql_select_db("dbname");
```

5. Untuk menampilkan data sebagai elemen array

```
Mysql_fetch_array();
```

6. Menutup koneksi database dengan menggunakan fungsi

```
Mysql_close(“dbname”);
```