BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

2.1.1 Pengertian Sistem

Terdapat dua kelompok pendekatan didalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut ini :

> *"Suatu 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."*

Menurut Jerry Fitzgerald, Ardra F. Fitzgerald dan Warren D. Stallings, Jr., mendefinisikan prosedur sebagai berikut :

> *"Suatu prosedur adalah urut-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan Apa (What) yang harus dikerjakan, Siapa (Who) yang mengerjakannya, Kapan (When) dikerjakan dan Bagaimana (How) mengerjakannya."*

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut ini :

"Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu."

Kedua kelompok definisi tersebut adalah benar dan tidak bertentangan, yang berbeda adalah cara pendekatannya. Pendekatan sistem yang merupakan kumpulan elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang lebih luas. Definisi ini lebih banyak diterima, karena kenyataannya suatu sistem dapat terdiri dari beberapa subsistem atau sistem bagian. Sebagai misal, sistem akuntansi dapat terdiri dari beberapa subsistem-subsistem, yaitu subsistem akuntansi penjualan, subsistem akuntansi pembelian, subsistem akuntansi penggajian, subsistem akuntansi biaya dan lain sebagainya.

2.1.2 Basis Data

Terdapat beberapa definisi basis data, yaitu :

"Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah."

"Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundancy) yang tidak perlu, untuk memenuhi berbagai kebutuhan."

"Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan tertentu."

Penggunaan Basis Data mempunyai tujuan-tujuan sebagai berikut :

1. Kecepatan dan Kemudahan (Speed)

Yakni agar pengguna basis data bisa:

1. menyimpan data
2. melakukan perubahan/manipulasi terhadap data
3. menampilkan kembali data

dengan lebih cepat dan mudah dibandingkan dengan cara biasa (baik manual ataupun elektronis).

2. Efisiensi Ruang Penyimpanan (Space)

Dengan basis data kita mampu melakukan penekanan jumlah redundansi (pengulangan) data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi antara kelompok data yang saling berhubungan.

3. Keakuratan (Accuracy)

Agar data sesuai dengan aturan dan batasan tertentu dengan cara memanfaatkan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (*constraint*) tipe data, domain data, keunikan data dsb.

4. Ketersediaan (Availability)

Agar data bisa diakses oleh setiap pengguna yang membutuhkan, dengan penerapan teknologi jaringan serta melakukan pemindahan/penghapusan data

yang sudah tidak digunakan / kadaluwarsa untuk menghemat ruang penyimpanan.

5. Kelengkapan (Completeness)

Agar data yang dikelola senantiasa lengkap baik relatif terhadap kebutuhan pemakai maupun terhadap waktu, dengan melakukan penambahan baris-baris data ataupun melakukan perubahan struktur pada basis data; yakni dengan menambahkan field pada tabel atau menambah tabel baru.

6. Keamanan (Security)

Agar data yang bersifat rahasia atau proses yang vital tidak jatuh ke orang / pengguna yang tidak berhak, yakni dengan penggunaan account (username dan password) serta menerapkan pembedaan hak akses setiap pengguna terhadap data yang bisa dibaca atau proses yang bisa dilakukan.

7. Kebersamaan (Sharability)

Agar data yang dikelola oleh sistem mendukung lingkungan multiuser (banyak pemakai), dengan menjaga / menghindari munculnya problem baru seperti ***inkonsistensi data*** (karena terjadi perubahan data yang dilakukan oleh beberapa user dalam waktu yang bersamaan) atau kondisi ***deadlock*** (karena ada banyak pemakai yang saling menunggu untuk menggunakan data).

2.1.3 Analisis Sistem

Analisis Sistem (system analysis) dapat didefinisikan sebagai berikut :

"*Penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya*"

Sedangkan fungsi dari analisis sistem adalah :

1. Mengidentifikasikan masalah-masalah dari user.
2. Menyatakan secara spesifik sasaran yang harus dicapai untuk memenuhi kebutuhan user.
3. Memilih alternatif-alternatif metode pemecahan masalah.
4. Merencanakan dan menerapkan rancangan sistemnya sesuai dengan permintaan user.

Tahap analisis sistem dilakukan setelah tahap perencanaan (*system planning*), dan sebelum tahap desain sistem (*system design*). Tahap analisis merupakan tahap yang sangat kritis dan sangat penting, karena kesalahan pada tahap ini akan menyebabkan kesalahan pada tahap selanjutnya.

Analisis yang dilakukan menggunakan metode analisis terstruktur, dimana metode analisis terstruktur ini banyak memakai *tool-tool* dalam mempresentasikan

suatu sistem, sehingga penguraian sistem dapat terstruktur dengan baik, jelas dan lebih mudah dipahami.

2.1.4 Diagram Konteks

Diagram konteks merupakan diagram yang mengandung satu proses yang menggambarkan hubungan keterkaitan antara sistem dengan pihak-pihak diluar lingkungan sistem dan posisi sistem didalam lingkungan tersebut. Pihak-pihak tersebut merupakan pihak-pihak yang membutuhkan informasi dan data dari sistem ataupun pihak-pihak yang menjadi sumber informasi dan data bagi sistem. Hubungan keterkaitannya digambarkan sebagai aliran informasi dan data yang masuk ke dalam sistem dan keluar dari sistem.

2.1.5 Data Flow Diagram

DFD adalah sebuah teknik grafik yang menggambarkan aliran data dan transformasi yang digunakan sebagai perjalanan data dari masukan menuju ke keluaran. DFD dapat diartikan juga sebagai model jaringan dari sebuah sistem. DFD dapat menggambarkan proses-proses yang terjadi dan aliran data diantaranya.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut disimpan. Secara umum DFD dapat diartikan sebagai salah satu *tools* untuk

analisis sistem yang dapat bermanfaat untuk menggambarkan proses, aliran data, *entity* yang terlibat serta *data store* yang digunakan dalam sistem yang dipelajari. Dengan menuangkan hasil analisis ke dalam DFD, seorang analis dapat memahami sistem yang sedang dipelajari dengan mudah dan baik.

Dibawah ini terdapat simbol-simbol untuk DFD yang diusulkan oleh Yourdon :

1. *Proses*, suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, prosedur atau alat yang digunakan untuk mentransformasikan data.
2. *Data Flow* (Arus Data), data yang mengalir dengan arah tertentu dari asal ke tujuan. Data yang mengalir dapat berupa dokumen, surat atau bentuk lainnya.
3. *Data Store* (Penyimpanan Data), digunakan untuk menyimpan dan mengambil data oleh proses. Data yang disimpan dapat berupa data yang terkomputerisasi maupun tidak terkomputerisasi.
4. *Source* atau *destination* atau dikenal juga dengan *external entity*, berupa orang, organisasi atau sistem lain yang berada diluar batas sistem yang berinteraksi dengan sistem yang sedang dikembangkan.

Dalam DFD terdapat beberapa aturan dasar penyusunan DFD yang dapat membantu untuk mempermudah penggambaran diagram arus data, diantaranya :

1. Setiap data yang dihasilkan atau keluar dari proses harus didasarkan pada data yang masuk ke proses tersebut.
2. Semua aliran data yang dihasilkan harus diberi nama, nama yang diberikan harus mencerminkan aliran data antara proses, *data store* dan *entity* yang ada.

3. Hanya data yang diperlukan untuk melakukan suatu proses saja yang harus digunakan sebagai input suatu proses.
4. Suatu proses harus tergantung pada input dan output yang masuk ke dalam proses itu saja, tidak perlu memperhatikan apa yang terjadi pada proses lainnya.
5. Setiap proses yang ada harus merupakan proses yang berjalan terus menerus, setiap proses harus diasumsikan siap menangani fungsi atau kerja setiap proses.

Agar representasi sistem dalam DFD mudah dipahami, maka DFD disusun dalam bentuk bertingkat (*leveled*) yang merupakan rincian lanjut dari proses pada level yang sebelumnya. Dimulai dari tingkat yang tertinggi dilakukan identifikasi proses apa saja yang ada dan data apa saja yang mengalir antar proses yang ada. Selanjutnya, dilakukan analisis terhadap setiap proses yang ada pada level tersebut, apakah sudah merupakan proses yang melakukan satu fungsi saja atau masih melakukan beberapa fungsi yang berbeda. Jika proses yang dianalisis masih melakukan lebih dari satu fungsi yang berbeda, maka perlu dilakukan pemecahan terhadap proses tersebut menjadi level lebih tinggi. Demikian seterusnya sampai didapat semua proses yang ada pada level paling tinggi dilakukan satu fungsi saja.

Jika DFD untuk suatu sistem yang dianalisis telah dibuat, sebelum melanjutkan kegiatan lainnya selalu dilakukan evaluasi atau pemeriksaan terhadap DFD tersebut. Pemeriksaan dilakukan terhadap munculnya kesalahan yang dapat diakibatkan oleh salah gambar, tidak konsistennya penggunaan simbol dan analisis terhadap proses yang terjadi.

2.1.6 Pengertian dan istilah dalam web

Portal web

- Menurut kamus webster portal web adalah :

*"A **web portal** is a site that provides a single function via a web page or site. Web portals often function as a point of access to information on the World Wide Web. Portals present information from diverse sources in an unified way. Aside from the search engine standard, web portals offer other services such as e-mail, news, stock prices, infotainment and various other features. Portals provide a way for enterprises to provide a consistent look and feel with access control and procedures for multiple applications, which otherwise would have been different entities altogether."*

- Menurut wikipedia :

"**Portal web** adalah situs web yang menyediakan kemampuan tertentu yang dibuat sedemikian rupa mencoba menuruti selera para pengunjungnya. Kemampuan portal yang lebih spesifik adalah penyediaan kandungan informasi yang dapat diakses menggunakan beragam perangkat, misalnya komputer pribadi, komputer jinjing (*notebook*), PDA (*Personal Digital Assistant*), atau bahkan telepon genggam."

Website

- Definisi website menurut wikipedia (indonesia):

"sebutan bagi sekelompok halaman web (*web page*), yang umumnya merupakan bagian dari suatu nama domain (*domain name*) atau subdomain di *World Wide Web* (WWW) di Internet. WWW terdiri dari seluruh situs web yang tersedia kepada publik. Halaman-halaman sebuah situs web diakses dari sebuah URL yang menjadi "akar" (*root*), yang disebut *homepage* (halaman induk; sering diterjemahkan menjadi "beranda", "halaman muka" atau laman web), dan biasanya disimpan dalam server yang sama."

- Menurut wikipedia (inggris) :

"*is a collection of related web pages, images, videos or other digital assets that are addressed relative to a common Uniform Resource Locator (URL), often consisting of only the domain name, or the IP address, and the root path ('/') in an Internet Protocol-based network. A web site is hosted on at least one web server, accessible via a network such as the Internet or a private local area network.*"

- Menurut kamus webster :

"*The entire collection of web pages and other information (such as images, sound, and video files, etc.) that are made available through what appears to users as a single web server. Typically all the of pages in a web site share the same basic URL*"

2.2 Perancangan Sistem

Perancangan sistem adalah suatu fase kreatif dalam siklus pengembangan sistem, dimana sistem yang dirancang harus memenuhi tujuan sistem secara teknis dan operasional. Pada tahap ini perlu dikembangkan spesifikasi untuk keluaran, database, operasi pemrosesan data, masukkan data, pengendalian serta pengamanan.

Perancangan sistem informasi yang dilakukan, bertujuan untuk menghilangkan beberapa kelemahan sistem dan meningkatkan kemampuan sistem dalam menyelesaikan permasalahan yang ada dengan acuan hasil-hasil analisis yang telah dilakukan.

2.2.1 Konsep Relasi Entitas

Berikut adalah beberapa jenis hubungan yang dapat terjadi dalam suatu basis data :

1. Relasi 1-1 (*One to One*)

   Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu.

2. Relasi 1-N (*One to Many*)

   Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak, tetapi tidak sebaliknya, dimana *file* kedua hanya berhubungan dengan satu entitas pada *file* pertama.

3. Relasi N-1 (*Many to One*)

   Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding satu, tetapi tidak sebaliknya, dimana *file* pertama hanya berhubungan dengan satu entitas pada *file* kedua.

4. Relasi N-N (*Many to Many)*

   Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak.

2.2.2 Kamus Data

Kamus data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dibuat untuk dapat digunakan pada

tahap analisis sistem dan saling berhubungan pada tahap perancangan sistem yang didasarkan pada aliran data di DFD.

Pada tahap analisis dan perancangan sistem, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem untuk dilakukannya proses perancangan sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan informasi yang dihubungkan oleh pemakai sistem.

2.2.3 Model-Model SDLC

Proses pengembangan sistem informasi adalah sekumpulan tahap, tugas dan aktivitas yang dibutuhkan untuk secara efisien mentransformasikan kebutuhan pemakai ke suatu solusi sistem informasi yang efektif. Pemodelan proses sistem informasi bertujuan untuk mempresentasikan aktivitas yang terjadi selama pembuatan sistem informasi dan perubahan-perubahan (evolusi). Latar belakang penggunaan model-model tersebut adalah kebutuhan untuk menghasilkan suatu sistem yang benar sedini mungkin di dalam proses pengembangannya. Alasan utama adalah biaya, semakin dini suatu kesalahan bisa dideteksi dalam pengembangan sistem, dan perbaikannya semakin rendah.

Suatu model proses yang baik dan terstruktur akan: (Boehm,1998)

1. Menentukan urutan tahap dalam suatu proses
2. Menciptakan kriteria transisi untuk maju ke tahap selanjutnya

Model-model *Sistem Development Life Cycle* (SDLC) adalah :

a. Model *Waterfall*

Model *Waterfall* ini juga dikenal dengan nama model tradisional atau model klasik. Model ini paling banyak dikenal dan digunakan. Langkah-langkah yang penting dalam model ini adalah :

1. Penentuan dan analisis spesifikasi
2. Desain sistem dan sistem informasi
3. Implementasi dan uji coba unit
4. Integrasi dan ujicoba sistem
5. Operasi dan pemeliharaan

b. Model *Prototyping*

Model ini adalah suatu proses yang memungkinkan pengembangan aplikasi untuk menciptakan suatu model dari sistem informasi yang harus dikembangkan. Tujuan dari *prototyping* adalah untuk mengurangi resiko dan ketidakpastian dari tahap-tahap awal dari *life cycle* pengembangan sistem informasi.

c. Model *Exploratory*

Model ini dipakai dengan awal seadanya. Pengembangan sistem tidak sepenuhnya mengerti dan menguasai *requirement* (kebutuhan) dari sistem. Sistem dikembangkan sejalan dengan adanya *requirement* baru.

d. Model *Incremental*

Model ini adalah pengembangan lebih lanjut dari model *exploratory* yang didesain untuk menjawab kritik terhadap *prototyping*. *Requirement* didefinisikan dan diserahkan kepada pemakai dalam bentuk bertahap.

e. Model *Transformational*

Model ini adalah model yang berorientasi pada spesifikasi formal. Spesifikasi ini ditransformasikan melalui beberapa tahap yang telah diuji kebenarannya sebelum menjadi sistem seutuhnya.

f. Model *Spiral*

Model ini dikembangkan oleh Boehm. Model ini cocok digunakan untuk organisasi yang besar.

g. Model 4GT

Pada alat bantu *Fourth Generation Technique* (4GT) memungkinkan pengembangan spesifikasi sistem informasi secara otomatis sesuai dengan yang akan diterjemahkan ke program. Dengan menggunakan 4GT, *software engineering* memusatkan perhatian hanya kepada spesifikasi sistem informasi yang diinginkan.

2.2.4 Pengujian Perangkat Lunak

Adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean.

Sejumlah aturan yang berfungsi sebagai sasaran pengujian pada perangkat lunak adalah:

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan
2. Test case yang baik adalah test case yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya

3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya

Sasaran itu berlawanan dengan pandangan yang biasanya dipegang yang menyatakan bahwa pengujian yang berhasil adalah pengujian yang tidak ada kesalahan yang ditemukan. Data yang dikumpulkan pada saat pengujian dilakukan memberikan indikasi yang baik mengenai reliabilitas perangkat lunak dan beberapa menunjukkan kualitas perangkat lunak secara keseluruhan, tetapi ada satu hal yang tidak dapat dilakukan oleh pengujian, yaitu pengujian tidak dapat memperlihatkan tidak adanya cacat, pengujian hanya dapat memperlihatkan bahwa ada kesalahan perangkat lunak.

Sebelum mengaplikasikan metode untuk mendesain test case yang efektif, perekayasa perangkat lunak harus memahami prinsip dasar yang menuntun pengujian perangkat lunak, yaitu:

- semua pengujian harus dapat ditelusuri sampai ke persyaratan pelanggan, maksudnya mengungkap kesalahan dari cacat yang menyebabkan program gagal.
- Pengujian harus direncanakan lama sebelum pengujian itu mulai, maksudnya semua pengujian dapat direncanakan dan dirancang sebelum semua kode dijalankan.

- Prinsip Pareto berlaku untuk pengujian perangkat lunak, maksudnya dari 80% kesalahan yang ditemukan selama pengujian dapat ditelusuri sampai 20% dari semua modul program.
- Pengujian harus mulai “dari yang kecil” dan berkembang ke pengujian “yang besar”, Selagi pengujian berlangsung maju, pengujian mengubah focus dalam usaha menemukan kesalahan pada cluster modul yang terintegrasi dan akhirnya pada sistem.
- Pengujian yang mendalam tidak mungkin karena tidak mungkin mengeksekusi setiap kombinasi jalur skema pengujian dikarenakan jumlah jalur permutasi untuk program menengah pun sangat besar.
- Untuk menjadi paling efektif, pengujian harus dilakukan oleh pihak ketiga yang independent

Dalam lingkungan yang ideal, perekayasa perangkat lunak mendesain suatu program computer, sebuah sistem atau produk dengan testabilitas dalam pikirannya. Hal ini memungkinkan individu yang berurusan dengan pengujian mendesain test case yang efektif secara lebih mudah. Testabilitas adalah seberapa mudah sebuah program computer dapat diuji. Karena sangat sulit, perlu diketahui apa yang dapat dilakukan untuk membuatnya menjadi lebih mudah. Procedural dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur eksekusi.

Sasaran utama desain test case adalah untuk mendapatkan serangkaian pengujian yang memiliki kemungkinan tertinggi di dalam pengungkapan kesalahan

pada perangkat lunak. Untuk mencapai sasaran tersebut, digunakan 4 kategori yang berbeda dari tehnik desain test case: Pengujian *white-box*, pengujian *black-box*, *Integrasi Bottom-Up* dan *Integrasi Top-Down.*

1. **Pengujian white-box**

Berfokus pada struktur control program. Test case dilakukan untuk memastikan bahwa semua statemen pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji. Pengujian basic path, tehnik pengujian white-box, menggunakan grafik (matriks grafiks) untuk melakukan serangkaian pengujian yang independent secara linear yang akan memastikan cakupan.

Pengujian aliran data dan kondisi lebih lanjut menggunakan logika program dan pengujian loop menyempurnakan tehnik white-box yang lain dengan memberikan sebuah prosedur untuk menguji loop dari tingkat kompleksitas yang bervariasi. Pengujian black-box didesain untuk mengungkap kesalahan pada persyaratan fungsional tanpa mengabaikan kerja internal dari suatu program.

2. **Tehnik pengujian black-box**

Berfokus pada domain informasi dari perangkat lunak, dengan melakukan test case dengan menpartisi domain input dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

Metode pengujian graph-based mengeksplorasi hubungan antara dan tingkah laku objek-objek program. Partisi ekivalensi membagi domain input ke dalam kelas data yang mungkin untuk melakukan fungsi perangkat lunak tertentu. Analisis nilai batas memeriksaa kemampuan program untuk menangani data pada batas yang dapat diterima.

Metode pengujian yang terspesialisasi meliputi sejumlah luas kemampuan perangkat lunak dan area aplikasi. GUI, arsitektur client/ server, dokumentasi dan fasilitas help dan sistem real time masing-masing membutuhkan pedoman dan tehnik khusus untuk pengujian perangkat lunak.

3. Integrasi Top-Down

Adalah pendekatan incremental dengan menggerakkan ke bawah melalui hirarki control, dimulai dengan control utama. Strategi intergrasi top-down memeriksa control mayor atau keputusan pada saat awal di dalam proses pengujian. Pada struktur program yang difaktorkan dengan baik, penarikan keputusan terjadi pada tingkat hirarki yang lebih tinggi sehingga terjadi lebih dulu.

Strategi top-down kelihatannya tidak sangat rumit, tetapi di dalam praktenya banyak menimbulkan masalah logistic. Biasanya masalah ini terjadi jika dibutuhkan pemrosesan di dalam hirarki pada tingkat rendah untuk menguji secara memadai tingkat yang lebih tinggi.

4. Pengujian Integrasi Bottom-up

Memulai konstruksi dan pengujian dengan modul atomic (modul pada tingkat paling rendah pada struktur program). Karena modul diintegrasikan dari bawah ke atas, maka pemrosesan yang diperlukan untuk modul subordinate ke suatu tuingkat yang diberikan akan selalu tersedia dan kebutuhan akan stub dapat dieliminasi. Strategi integrasi bottom-up dapat diimplementasi dengan langkah-langkah:

1. modul tingkat rendah digabung ke dalam cluster (build) yang melakukan subfungsi perangkat lunak spesifik.
2. Driver (program control untuk pengujian) ditulis untuk mengkoordinasi input dan output test case
3. cluster diuji
4. driver diganti dan cluster digabungkan dengan menggerakkannya ke atas di dalam struktur program.

2.3 Perangkat Lunak Penunjang

2.3.1 *Personal Home Page* (PHP)

Personal Home Page (PHP) merupakan bahasa pemograman *scripting* yang bersifat *open source*. *PHP* adalah salah satu bahasa *Server-side* yang didesain khusus untuk aplikasi web. *PHP* dapat disisipkan diantara bahasa *HTML* (*Hypertext Markup Language*) dan karena bahasa *Server side,* maka bahasa *PHP* akan dieksekusi di

server, sehingga yang dikirimkan ke *browser* adalah “hasil jadi” dalam bentuk *HTML*, dan kode *PHP* anda tidak akan terlihat.

PHP dahulunya merupakan proyek pribadi dari Rasmus Lerdorf (dengan dikeluarkannya *PHP* versi 1) yang digunakan untuk membuat *home page* pribadinya. Versi pertama ini berupa kumpulan *script PERL*. Untuk versi keduanya, Rasmus menulis ulang *script-script PERL* tersebut menggunakan bahasa *C*, kemudian menambahkan fasilitas untuk *Form HTML* dan koneksi *MySQL*. Setelah mengalami perkembangan oleh suatu kelompok *open source* (termasuk Rasmus) maka mulai versi 3 nya. Sintak yang digunakan berasal dari bahasa *C* , *Java* maupun *PERL*. Untuk *release* terbaru dari *PHP* dapat anda lihat pada *website http://www.php.net/*.

PHP merupakan bahasa *script* yang digunakan untuk membuat halaman *web* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru. Semua *script PHP* dieksekusi pada *server* dimana *script* tersebut dijalankan. Oleh karena itu, spesifikasi *server* lebih berpengaruh pada eksekusi dari *script PHP* daripada spesifikasi *client*. Namun tetap diperhatikan bahwa halaman *web* yang dihasilkan tentunya harus dapat dibuka oleh *browser* pada *client*. Dalam hal ini versi dari *HTML* yang digunakan harus didukung oleh *browser client*.

2.3.2 MySQL

MySQL merupakan sebuah *database* yang berbasiskan *server database*. Kemampuannya dalam menangani *RDBMS* (*Relational Database Management System*) membuat *database* ini sangat popular saat ini. *MySQL* mampu mengelola data yang sangat besar hingga ukuran *Gigabyte*. *MySQL* dapat digunakan untuk kepentingan komersial maupun *personal* (*non profit*). *Database server* itu sendiri merupakan suatu *software* yang bertugas untuk melayani permintaan (*request*) *query* dari *client*.

MySQL sebagai suatu *database server* mempunyai beberapa kemampuan, salah satunya harus menyediakan suatu sistem manajemen *database* yang dapat mengatur bagaimana menyimpan, menambah, mengakses data dan transaksi-transaksi *database* lainnya. *MySQL* cepat sekali berkembang, karena *MySQL* merupakan suatu *software* yang *Open Source*. *Website MySQL* adalah *www.mysql.com*, di *website* inilah ada banyak hal yang bisa ditemukan seperti *MySQL* versi terbaru atau dokumentasi-dokumentasi serta program-program *utility MySQL*.