

BAB II

TINJAUAN PUSTAKA

2.1 Aplikasi

Menurut Jogiyanto (2004:4), aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Jogiyanto menambahkan aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal. Menurut kamus komputer eksekutif, pengertian aplikasi merupakan pemecahan masalah yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Aplikasi biasanya berupa perangkat lunak yang berbentuk *software* yang berisi kesatuan perintah atau program yang dibuat untuk melaksanakan sebuah pekerjaan yang diinginkan.

Selain itu aplikasi juga mempunyai fungsi sebagai pelayan kebutuhan beberapa aktivitas yang dilakukan oleh manusia seperti sistem untuk *software* jual beli, permainan atau game online, pelayanan masyarakat dan hampir semua proses yang dilakukan oleh manusia dapat dibantu dengan menggunakan suatu aplikasi. Lebih dari satu aplikasi jika digabungkan akan menjadi satu paket atau sering juga disebut dengan *application suite*, dimana aplikasi tersebut memiliki posisi antar muka yang mempunyai kesamaan sehingga dapat dengan mudah digunakan atau dipelajari penggunaan tiap aplikasi tersebut.

2.2 Barang

Menurut Bambang (2003:12), barang merupakan suatu produk fisik (berwujud, *tangible*) yang dapat diberikan pada seorang pembeli dan melibatkan perpindahan kepemilikan dari penjual ke pelanggan. Barang didefinisikan sebagai suatu produk fisik (berwujud, *tangible*) yang dapat diberikan pada seorang pembeli dan melibatkan perpindahan kepemilikan dari penjual ke pelanggan, kebalikan dengan suatu jasa (tak berwujud, *intangible*). Istilah "komoditas" sering digunakan dalam mikroekonomi untuk membedakan barang dan jasa.

2.3 Praktikum

Menurut *Hegarty-Hazel* seperti dikutip Lazarowitz & Tamir (1994) praktikum adalah suatu bentuk kerja praktek yang bertempat dalam lingkungan yang disesuaikan dengan tujuan agar siswa terlibat dalam pengalaman belajar yang terencana dan berinteraksi dengan peralatan untuk memahami teori yang telah dipelajari. Metode praktikum adalah cara penyampaian bahan pelajaran dengan memberikan kesempatan berlatih untuk meningkatkan keterampilan sebagai penerapan bahan/pengetahuan yang telah mereka pelajari sebelumnya mencapai tujuan pengajaran.

2.4 Java

Menurut Shalahuddin dan Rosa (2010:1), Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri ataupun pada lingkungan jaringan. Java merupakan bahasa pemrograman yang berorientasi objek dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*. Sebagai sebuah bahasa pemrograman, java dapat membuat seluruh bentuk aplikasi, *desktop*, *website* dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Java dibuat pada tahun 1991 oleh sekelompok insinyur Sun dipimpin Patrick Naughton dan James Gosling. Proyek pembuatan bahasa pemrograman ini diberi nama proyek *green*. Karena orang-orang diproyek *Green* berbasis C++ dan bukan Pascal maka kebanyakan sintaks diambil dari C++, serta mengadopsi orientasi objek dan bukan prosedural. Mulanya bahasa yang diciptakan diberi nama "Oak", namun dikarenakan nama Oak sendiri merupakan nama bahasa pemrograman yang sudah ada sebelumnya, kemudian digantikan dengan Java. Nama Java sendiri terinspirasi pada saat mereka menikmati secangkir kopi di sebuah kedai kopi yang kemudian tidak sengaja salah satu dari mereka menyebutkan kata Java yang mengandung arti biji kopi.

Untuk membuat aplikasi berbasis java, diperlukan *Java Development Kit* (JDK) dan *Java Runtime Environment* (JRE). JDK ini berguna saat anda menulis

kode program, sedangkan JRE ini yang memungkinkan sebuah program java dapat berjalan di mesin. JRE mengeksekusi *binary-binary* dari *class-class* dan mengirimnya ke *Java Virtual Machine (JVM)* untuk diproses lagi ke prosesor.

2.5 *Client Server*

Definisi *Client Server* menurut Budhi Irawan (2005:30), *Client-Server* merupakan jaringan untuk memusatkan fungsi dari aplikasi atau yang biasa disebut dengan *dedicated file server*. Sebuah *file server* ini menjadi jantung dari keseluruhan sistem dan *file server* yang terpusat ini digunakan *client* untuk mengakses dokumen-dokumen yang diperlukan *client*.

Dalam model *Client/Server*, sebuah aplikasi dibagi menjadi dua bagian yang terpisah, tapi masih merupakan sebuah kesatuan yakni komponen *Client* dan komponen *Server*. Komponen *client* juga sering disebut sebagai *front-end*, sementara komponen *server* disebut sebagai *back-end*. Komponen *client* dari aplikasi tersebut dijalankan dalam sebuah *workstation* dan menerima masukan data dari pengguna. Komponen *client* tersebut akan menyiapkan data yang dimasukkan oleh pengguna dengan menggunakan teknologi pemrosesan tertentu dan mengirimkannya kepada komponen *server* yang dijalankan di atas mesin *server*, umumnya dalam bentuk *request* terhadap beberapa layanan yang dimiliki oleh *server*. Komponen *server* akan menerima *request* dari *client* dan langsung memprosesnya lalu mengembalikan hasil pemrosesan tersebut kepada *client*. *Client* pun menerima informasi hasil pemrosesan data yang dilakukan *server* dan menampilkannya kepada pengguna, dengan menggunakan aplikasi yang berinteraksi dengan pengguna.

2.5.1 *Client*

Client adalah sembarang sistem atau proses yang melakukan suatu permintaan data atau layanan ke *server*. *Client* memiliki beberapa fungsi yaitu :

1. Mengatur *user interface*
2. Menerima dan memeriksa sintaks input dari pemakai
3. Memproses aplikasi
4. *Generate* permintaan basis data dan memindahkannya ke *server*

5. Memberikan respon balik kepada pemakai
6. Menyediakan akses basis data secara bersamaan
7. Menyediakan kontrol *recovery*

2.5.2 *Server*

Server adalah sistem atau proses yang menyediakan data atau layanan yang diminta oleh *client*. Secara fisik, sebuah *server* dapat berupa komputer *mainframe*, mini-komputer, *workstation*, ataupun PC atau peranti lain seperti printer, *server* tidak harus berupa sistem fisik, tetapi juga suatu proses. *Server* memiliki beberapa fungsi yaitu :

1. Menerima dan memproses basis data yang diminta dari *client*.
2. Memeriksa otorisasi.
3. Menjamin tidak terjadi pelanggaran terhadap *integrity constraint*.
4. Melakukan *query*/pemrosesan *update* dan memindahkan respon ke *client*.
5. Memelihara data *dictionary*.

2.6 *Netbeans IDE*

Netbeans merupakan sebuah aplikasi *Integrated Development Environment* (IDE) yang berbasiskan Java dari *Sun Microsystems* yang berjalan di atas *swing*. *Swing* merupakan sebuah teknologi Java untuk pengembangan aplikasi *desktop* yang dapat berjalan pada berbagai macam *platform* seperti windows, linux, Mac OS X dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan *Graphic User Interface* (GUI), suatu kode editor atau *text*, suatu *compiler* dan suatu *debugger*.

Netbeans juga digunakan oleh sang programmer untuk menulis, meng-compile, mencari kesalahan dan menyebarkan program *netbeans* yang ditulis dalam bahasa pemrograman java namun selain itu dapat juga mendukung bahasa pemrograman lainnya dan program ini pun bebas untuk digunakan dan untuk membuat professional *desktop*, *enterprise*, *web*, dan *mobile applications* dengan *Java language*, C/C++, dan bahkan *dynamic languages* seperti PHP, JavaScript, Groovy, dan Ruby. *Netbeans* merupakan sebuah proyek kode terbuka yang sukses

dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah). *Sun Microsystems* mendirikan proyek kode terbuka *Netbeans* pada bulan Juni 2000 dan terus menjadi sponsor utama. Dan saat ini pun *netbeans* memiliki dua produk yaitu *Platform Netbeans* dan *Netbeans IDE*. *Platform Netbeans* merupakan *framework* yang dapat digunakan kembali (*reusable*) untuk menyederhanakan pengembangan aplikasi *desktop* dan *Platform Netbeans* juga menawarkan layanan-layanan yang umum bagi aplikasi *desktop*, mengizinkan pengembang untuk fokus ke logika yang spesifik terhadap aplikasi.

2.7 Basis Data (*Database*)

Menurut Pakereng & Wahyono (2004), basis data merupakan kumpulan data yang dipakai ada dalam suatu lingkup tertentu, misalkan instansi, perusahaan, dan lain-lain atau kasus tertentu. Menurut Pakereng & Wahyono (2004), sebuah konsep *database* memiliki beberapa hal sebagai berikut :

1. Entitas

Entitas merupakan tempat informasi direkam, dapat berupa orang, tempat, kejadian dan lain-lain. Sebagai contoh dalam kasus Administrasi Siswa misalnya, maka terdapat entitas siswa, matakuliah, guru, pembayaran.

2. Atribut

Atribut dapat juga disebut sebagai data elemen, data *field*, atau data item yang digunakan untuk menerangkan suatu entitas dan mempunyai harga tertentu, misalnya atribut dari entitas siswa diterangkan oleh nama, tanggal lahir, alamat.

3. *Data value*

Data value merupakan suatu informasi atau data aktual yang disimpan pada tiap data, elemen, atau atribut. Atribut nama pegawai menunjukkan tempat dimana informasi nama karyawan disimpan, nilai datanya misalnya adalah Anja, Arif, Suryo, dan lain-lain yang merupakan isi data nama pegawai tersebut.

3. *File/Table*

Merupakan kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.

4. *Record/Tuple*

Merupakan kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu *record* mewakili satu data atau informasi.

2.8 XAMPP

Menurut Arief Ramadhan dan Hendra Saputra (2005), XAMPP merupakan sebuah tool yang menyediakan beberapa paket perangkat lunak ke dalam satu buah paket. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. XAMPP merupakan *software* yang mudah digunakan, dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah menginstall satu kali sudah tersedia Apache, Web Server, MySQL, Database Server, PHP Support dan beberapa *module* lainnya. Fungsinya sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Bagian-bagian penting XAMPP terdiri dari :

1. Htdoc, adalah folder tempat meletakkan berkas-berkas yang akan dijalankan seperti berkas PHP, HTML, dan script lainnya.
2. phpMyAdmin adalah bagian yang mengelola basis data MySQL yang ada di komputer.
3. *Control Panel* berfungsi untuk mengelola layanan XAMPP. *Control Panel* terdiri dari tombol perintah *stop* dan *start* yang digunakan untuk menjalankan dan menghentikan layanan pada XAMPP.

2.9 MySQL

Menurut Sulhan (2007:118), “MySQL merupakan perangkat lunak yang digunakan untuk membangun *database* yang sering digunakan di lingkungan linux. MySQL merupakan *software open source* yang berarti *free* untuk digunakan. Selain di lingkungan linux, MySQL juga tersedia di lingkungan windows”. Dari kedua pengertian tersebut, dapat disimpulkan bahwa MySQL adalah perangkat lunak yang digunakan untuk membuat aplikasi dengan

menggunakan *database* sebagai sumber pengelolaan datanya. MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas.

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X *Server*, Solaris, Amiga, dan masih banyak lagi.

2. Perangkat lunak sumber terbuka.

MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

3. *Multi-user*.

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tipe data.

MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

6. Perintah dan Fungsi.

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).

7. Keamanan.

MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan Pembatasan.

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas.

MySQL dapat melakukan koneksi dengan *client* menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

10. Lokalisasi.

MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antar Muka.

MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. *Client* dan Peralatan.

MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

13. Struktur tabel.

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.10 Flowchart

Menurut Adhi (2012:26), *Flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur program yang biasanya mempermudah penyelesaian masalah. *Flowchart* atau diagram alir merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing-masing langkah tersebut menggunakan tanda panah. *Flowchart* dibagi menjadi dua bagian yaitu :

1. *Flowchart* Sistem

Bagan yang menggambarkan suatu prosedur dan proses suatu *file* dalam suatu media menjadi *file* dalam media yang lain dalam suatu sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem.

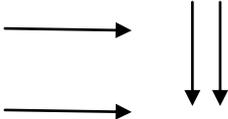
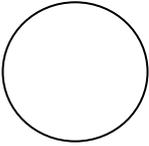
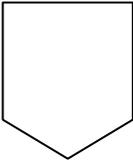
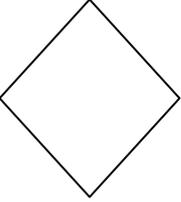
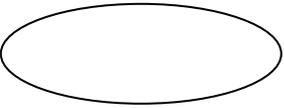
2. *Flowchart* Program

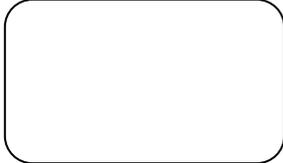
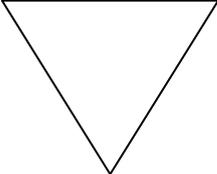
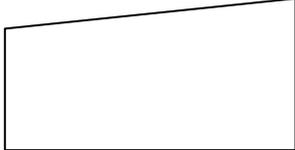
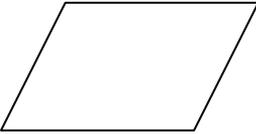
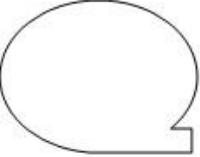
Bagan yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Bagan ini menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagian aliran sistem.

2.10.1 Simbol-simbol *Flowchart*

Simbol - simbol *flowchart* beserta fungsinya dapat ditunjukkan pada tabel berikut:

Tabel 2.1 Simbol-simbol *Flowchart*

NO	SIMBOL	KETERANGAN
1		Simbol arus / <i>flow</i> , yaitu menyatakan jalannya arus suatu proses
2		Simbol <i>connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
3		Simbol <i>offline connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
4		Simbol proses, yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
5		Simbol <i>manual</i> , menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
6		Simbol <i>decision</i> , yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak
7		Simbol <i>terminal</i> , yaitu menyatakan permulaan atau akhir suatu program

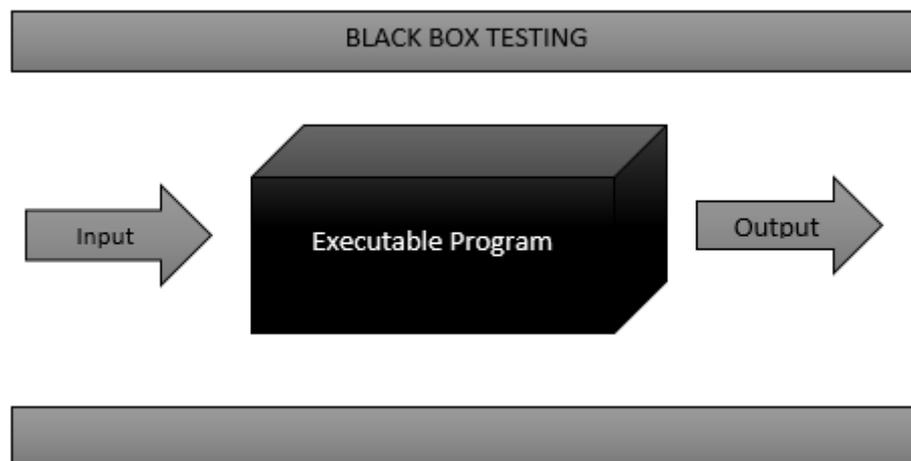
8		Simbol <i>predefined process</i> , menyatakan persediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
9		Simbol <i>keying operation</i> , menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>
10		Simbol <i>offline-storage</i> , menunjukkan bahwa data dalam symbol ini akan disimpan ke dalam suatu media tertentu
11		Simbol <i>manual input</i> , menyatakan data secara manual dengan menggunakan <i>online keyboard</i> .
12		Simbol <i>input / output</i> , menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya
13		Simbol <i>magnetic tape</i> , menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> tersimpan ke dalam pita magnetis
14		Simbol <i>disk storage</i> , menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> tersimpan kedalam <i>disk</i>

16		Simbol <i>punched card</i> , menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu
----	---	---

2.11 Black Box Testing

Menurut Simarmata(2009) menyatakan pengujian adalah sebuah proses terhadap aplikasi atau program untuk menemukan segala kesalahan dan segala kemungkinan yang akan menimbulkan kesalahan sesuai dengan spesifikasi perangkat lunak yang telah ditentukan sebelum aplikasi tersebut diserahkan kepada pengguna. Salah satu jenis module untuk melakukan pengujian pada perangkat lunak yaitu *black box testing*.

Black box testing terfokus pada apakah unit program memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi. Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses yang diinginkan.



Gambar 2.1 *Black Box Testing*

Teknik yang digunakan dalam *black box testing* antara lain :

- a. Digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak.

- b. Kebenaran perangkat lunak yang diuji dilihat berdasarkan keluaran(*output*) yang dihasilkan.
- c. Kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya.

2.12 White Box Testing

Menurut Myers G. J (1979: p50) *White-box testing* adalah struktur dan aliran perangkat lunak yang diuji terlihat ke tester. Pengujian rencana yang dibuat sesuai dengan rincian pelaksanaan perangkat lunak, seperti bahasa pemrograman, logika, dan gaya. Uji kasus berasal dari struktur program. Pengujian kotak putih-disebut juga pengujian kaca-kotak, pengujian logika-*driven* atau desain berbasis pengujian. Kelebihan *White Box Testing* antara lain :

1. Kesalahan Logika menggunakan syntax 'if' dan syntax pengulangan. Langkah selanjutnya metode *white box testing* ini akan mencari dan mendeteksi segala kondisi yang di percaya tidak sesuai dan mencari kapan suatu proses perulangan di akhiri.
2. Ketidaksesuaian Asumsi Menampilkan dan memonitor beberapa asumsi yang diyakini tidak sesuai dengan yang diharapkan atau yang akan diwujudkan, untuk selanjutnya akan dianalisa kembali dan kemudian diperbaiki.
3. Kesalahan Pengetikan Mendeteksi dan mencaribahasa-bahasa pemograman yang di anggap bersifat case sensitif.

Kelemahan *White Box Testing* adalah pada perangkat lunak yang jenisnya besar, metode *white box testing* ini dianggap boros karena melibatkan banyak sumberdaya untuk melakukannya.