

BAB 2

LANDASAN TEORI

2.1. *Internet*

Internet adalah jaringan internasional yang terdiri dari jutaan jaringan kecil yang menghubungkan pendidikan, komersial, nonprofit, dan organisasi kemiliteran, serta individual. (Williams and Sawyer,2007,p.60).

Internet merupakan jalur komunikasi terbesar yang menghubungkan semua orang yang tidak terpengaruh oleh jarak dan waktu. Semua orang dapat bertukar informasi dengan jaringan *internet* melalui *E-Mail*, *newsgroup*, *chatting*, dan juga *E-Commerce*.

2.1.1 *Sejarah Internet*

“ARPAnet adalah sebuah jaringan antar komputer. *Internet* merupakan jaringan dari jejaring. Jaringan antarplanet akan menjadi jaringan *internet*.” – Vinton G. Cerf – Pencipta *internet*.

(Haas,2005,p2-13)

Sebelum ARPAnet

- 1957

USSR menerbitkan *Sputnik*

Terjadi “*Sputnik Shock*” di USA

- 1958

Kongres US memberikan dana untuk *Advanced Research Projects Agency* (ARPA)

Desain ARPAnet dimulai

Kelahiran ARPAnet

- 1969

Bell Labs menciptakan ESS-1, *switch* pertama

- 1970

ARPAnet menggunakan *Network Control Protocol* (NCP)

- 1971

ARPAnet menghubungkan 15 situs termasuk organisasi dan universitas

- 1972

Ray Tomlinson menciptakan aplikasi *email* pertama

Kesuksesan TCP/IP

- 1978

TCP terpecah menjadi TCP dan IP

- 1983

ARPAnet meng-convert TCP/IP

UNIX dirilis, sebuah kernel untuk TCP/IP

Internet Activities Board (IAB) berdiri

Lanjutan Milestone

- 1984

Terciptanya *Domain Name System* (DNS)

- 1985

Symbolics.com domain pertama yang tercatat

Pendiri ARPAnet

- J.C.R Licklinder
Distributor *network* dan peng-alamatan
- Robert Taylor
Designer ruang kerja ARPA
- Larry Roberts
Arsitektur ARPA
- Barry Wessler
Administrator ARPA

2.1.2 World Wide Web (WWW)

World Wide Web (WWW) merupakan sistem informasi *hypertext global*, interaktif, dinamis, *cross-platform*, terdistribusi, dan grafis yang bekerja di *internet*. (Lemay,2000,p.10).

World Wide Web sebagai bagian dari *internet*, mencakup teknologi berbasis multimedia yang memungkinkan untuk mengakses tidak hanya text. Melainkan juga dapat mengunduh seni, *audio*, *video*, dan animasi serta interaktif *game*.

2.1.3 Web Server

Web server adalah aplikasi yang berjalan pada *web site* dan bertanggung jawab untuk membalas request data oleh *web browser*. Untuk mempublikasikan data pada web dibutuhkan *web server* (Lemay,2000,p.23).

Web server tidak hanya menyimpan data tetapi juga bertanggung jawab dalam mengatur inputan data dan menghubungkan data dengan

browser menggunakan program seperti *database* yang berjalan pada *server*.

2.1.4 Web Browser

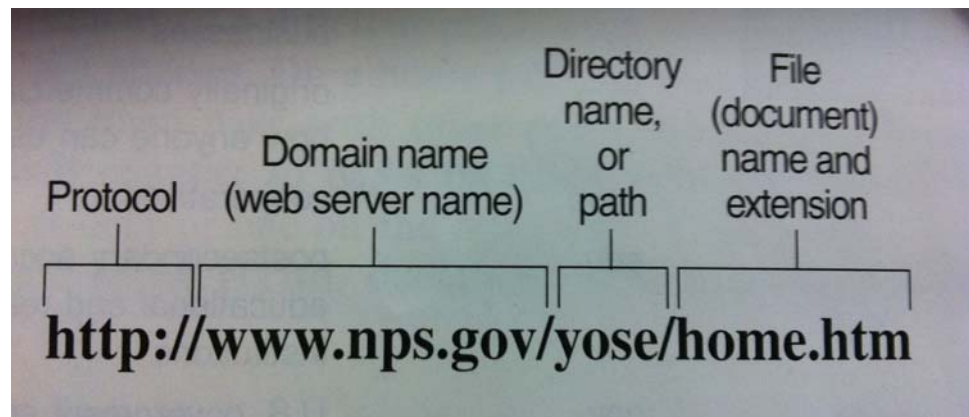
Web browser adalah aplikasi yang digunakan untuk menampilkan halaman dan arahan pada *World Wide Web* (Lemay,2000,p.19).

Browser memungkinkan pengguna untuk berselancar mengikuti arus pada web. Berselancar mempunyai arti menjelajahi web menggunakan *mouse* untuk bergerak melalui sarangkaian jalur yang terhubung, atau link, dari satu lokasi, atau *website*, ke lokasi, atau *website* lainnya.

2.1.5 Uniform Resource Locator (URL)

URL adalah sebuah *string* karakter-karakter yang menunjuk ke lokasi spesifik suatu informasi yang berada di mana saja pada web. Dengan kata lain, URL adalah alamat unik suatu *website* (Williams and Sawyer,2007,p.66).

Sebelum *web browser* dapat terhubung dengan *website*, *browser* harus mengetahui alamat *website* tersebut, yaitu URL nya. URL terdiri dari *web protocol*, nama *domain* atau *server*, direktori, dan berkas yang berada di dalam direktori tersebut.



Gambar 2.1. Contoh penulisan URL

2.1.6 *HyperText Transfer Protocol (HTTP)*

HTTP adalah aturan komunikasi yang memungkinkan *browser* untuk terhubung dengan *web server*. Tanpa HTTP, file-file tidak dapat ditransfer melalui web (Williams and Sawyer,2007,p.67).

HTTP merupakan *protocol* yang dapat melayani akses informasi dalam bentuk text, gambar, suara, *video*, dan bentuk *multimedia* lainnya secara cepat. Ketika pemakai menggunakan HTTP, *browser* akan terhubung kepada *server*, mengambil dokumen yang dibutuhkan, kemudian memutuskan kembali hubungan dengan *server*. *Server* HTTP akan memberikan informasi kepada *browser* mengenai tipe dokumen yang akan ditampilkan sehingga *browser* akan menampilkan dengan benar.

2.2 *Software Engineering*

Software adalah sebuah instruksi (aplikasi komputer) yang ketika di eksekusi akan memberikan fungsi dan performa yang diinginkan, merupakan struktur data yang memungkinkan aplikasi untuk memanipulasi informasi, dan

dokumen yang menjelaskan operasi dan guna aplikasi tersebut. (Pressman,2001,p.6)

Software Engineering bertujuan untuk merancang dan mengaplikasikan aplikasi agar dapat digunakan dalam berbagai macam cara dan dengan berbagai macam program. Perubahan komponen aplikasi modern memungkinkan *software engineer* untuk membuat aplikasi yang tidak hanya mengandung algoritma, tetapi juga struktur data.

2.3 Database

Database adalah kumpulan koleksi data-data yang berhubungan secara logis, dan deskripsi dari data, dirancang untuk memenuhi kebutuhan akan informasi suatu organisasi (Connolly,2005,p15).

Database merupakan tempat penyimpanan kumpulan data yang besar dan tunggal yang dapat digunakan secara bersamaan oleh banyak departemen dan pengguna. Semua data telah terintegrasi dengan jumlah duplikasi yang minimum. *Database* tidak hanya dimiliki oleh satu departemen tetapi merupakan sumber daya bersama satu perusahaan. *Database* tidak hanya memegang data operasional organisasi, tetapi juga menyimpan deskripsi data.

2.3.1 Database Management System (DBMS)

DBMS adalah sebuah sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengontrol akses ke *database* (Connolly,2005,p16).

DBMS merupakan komponen paling utama dan yang harus ada dalam komputasi modern, dan hasil dari penelitian dan pengembangan

selama bertahun-tahun baik di sisi akademis dan industri (Hellerstein, etc; 2007, p1).

DBMS adalah perangkat lunak yang berinteraksi antara aplikasi pengguna dan *database*. DBMS menyediakan fasilitas sebagai berikut:

- Mengizinkan pengguna dalam mendefinisikan database, umumnya menggunakan *Data Definition Language* (DDL). DDL mengizinkan pengguna untuk menentukan tipe data dan struktur dan batasan pada data yang akan di simpan.
- Mengizinkan pengguna untuk menyisipkan, memperbarui, menghapus, dan mendapatkan kembali data dari database, umumnya melalui *Data Manipulation Language* (DML). Memiliki penyimpanan sentral untuk semua data dan deskripsi data memungkinkan DML untuk memberikan fasilitas permintaan data, disebut bahasa *query*. Bahasa *query* yang paling umum adalah *Structured Query Language* (SQL), dimana sekarang menjadi bahasa standard untuk relasi DBMS.
- Memberikan akses yang terkontrol ke *database*. Sebagai contoh:
 - Sistem Keamanan
 - Sistem Integritas
 - Sistem Konkurensi
 - Sistem Kontrol Pemulihan
 - Katalog Akses Pengguna

2.3.2 Program Aplikasi *Database*

Program Aplikasi Database merupakan program komputer yang berinteraksi dengan *database* dengan cara memberikan permintaan yang sesuai (umumnya pernyataan *SQL*) dengan DBMS (Connolly,2005,p17).

Pengguna berinteraksi dengan *database* melalui sejumlah program program aplikasi yang digunakan untuk menciptakan dan memelihara *database* serta menghasilkan informasi. Program-program ini dapat membuat aplikasi *batch* yang konvensional atau aplikasi online.

2.4 *Unified Modeling Language (UML)*

UML adalah penerus dari metode *object-oriented analysis and design* (OOA&D) yang muncul pada tahun 80an dan 90an (Fowler dan Scott,2000,p1). UML lanjut melalui proses standarisasi dengan *Object Management Group* (OMG) dan sekarang menjadi standar OMG.

UML disebut sebagai Bahasa *Modeling*, bukan sebuah metode. Sebagian besar metode terdiri dari bahasa pemodelan dan sebuah proses. Bahasa pemodelan adalah notasi bahwa metode digunakan untuk mengekspresikan desain. Proses ini merupakan saran tentang langkah apa yang harus diambil dalam melakukan desain.

2.4.1 *Use Case Diagram*

Use case diperkenalkan sebagai elemen pokok dalam pengembangan perangkat lunak, Jacobson (1994) juga memperkenalkan diagram yang menampilkan kasus penggunaan. Use Case Diagram kini menjadi bagian dari UML (Fowler dan Scott,2000,p41).

2.4.2 *Class Diagram*

Class Diagram menggambarkan tipe objek di dalam sistem dan bermacam jenis hubungan statis yang ada di antara objek-objek tersebut. Ada 2 jenis pokok akan hubungan statis:

- Asosiasi (sebagai contoh, Pelanggan meminjam beberapa *video*)
- Subtipe (perawat adalah orang yang baik)

Class diagram juga menampilkan atribut dan operasi kelas dan batasan yang berlaku terhadap objek yang berhubungan (Fowler dan Scott,2000,p49).

2.4.3 *Activity Diagram*

Activity Diagram menggabungkan ide dari beberapa teknik: *event diagram*, teknik *SDL modeling*, *work flow modeling*, dan *Petri nets*. Diagram-diagram ini khususnya berguna dalam hubungan dengan *workflow* dan dalam menggambarkan perilaku yang memiliki banyak pengolahan paralel (Fowler dan Scott,2000,p129).

Activity Diagram menggambarkan urutan aktivitas, dimana dukungan kondisi dan perilaku paralel. *Activity diagram* merupakan variasi dari diagram kondisi, dimana kondisi ini merupakan sebuah aktivitas.

2.4.4 *Interaction Diagram*

Diagram interaksi adalah model yang menjelaskan bagaimana kelompok-kelompok objek berkolaborasi dalam beberapa perilaku.

Biasanya, diagram interaksi menangkap perilaku kasus penggunaan tunggal. Ada 2 jenis diagram interaksi : *Sequence diagram* dan *Collaboration diagram*.

2.5 *PHP Hypertext Preprocessor (PHP)*

PHP adalah bahasa pemrograman berbasis *server* yang dirancang khusus untuk web (Welling dan Thomson,2003,p2-3). Dalam halaman HTML, dapat dimasukkan kode PHP yang akan dieksekusi setiap kali halaman web tersebut diakses. Kode PHP ini akan diterjemahkan oleh *web server* dan akan dijalankan bersamaan dengan HTML atau output lainnya, yang akan dilihat oleh pengunjung situs web.

Beberapa kelebihan PHP dibandingkan dengan bahasa pemrograman sejenis seperti *Microsoft Active Server Pages (ASP)* dan *Java Server Pages (JSP)* adalah:

- Kemampuan yang tinggi.
- Kemampuan untuk dapat terhubung dengan banyak sistem basis data, seperti : *MySQL, Oracle*.
- Tidak dibutuhkan biaya untuk mendapatkan PHP.
- Mudah dipelajari dan digunakan, karena PHP dibuat berdasarkan bahasa pemrograman dasar, yaitu bahasa *C dan Perl*.
- Dapat berjalan pada berbagai sistem operasi, seperti *Linux, Solaris*, dan berbagai versi *Microsoft Windows*.

2.6 *MySQL*

MySQL merupakan sistem manajemen hubungan antar basis data yang sangat cepat dan sempurna (Welling dan Thomson,2003,p3). *MySQL* merupakan alat bantu untuk memanipulasi basis data, sehingga basis data dapat dengan mudah diisi, diambil, disusun dan diubah datanya. Server *MySQL* pun dapat mengatur kontrol akses dari data, sehingga beberapa pengguna dapat sekaligus bekerja pada waktu yang bersamaan.

Beberapa kelebihan *MySQL*, dibandingkan dengan sistem basis data sejenis seperti *Microsoft SQL Server*, *Oracle*:

- Kemampuan yang tinggi.
- Tidak dibutuhkan biaya untuk mendapatkan *MySQL*.
- Mudah untuk konfigurasi dan dipelajari.
- Dapat dijalankan pada beberapa sistem operasi seperti sistem *Unix* dan *Microsoft Windows*.

2.7 **Interaksi Manusia dan Komputer**

Interaksi manusia dan komputer (*human-computer interaction*, HCI) adalah disiplin ilmu yang mempelajari hubungan antara manusia dan komputer yang meliputi perancangan, evaluasi, dan implementasi antarmuka pengguna komputer agar mudah digunakan oleh manusia. Sedangkan interaksi manusia dan komputer sendiri adalah serangkaian proses, dialog dan kegiatan yang dilakukan oleh manusia untuk berinteraksi dengan komputer yang keduanya saling memberikan masukan dan umpan balik melalui sebuah antarmuka untuk memperoleh hasil akhir yang diharapkan.

Sebuah sistem berbasis komputer selalu mempunyai elemen manusia yang terkait di dalamnya (Shneiderman,2010,p88-89). Elemen manusia ini berinteraksi langsung dengan perangkat keras dan perangkat lunak melalui sebuah antarmuka (*interface*).

Penggunaan delapan aturan emas berasal dari prinsip – prinsip yang didapatkan dari pengalaman – pengalaman, dan direvisi selama lebih dari 2 dekade. Delapan aturan emas tersebut adalah:

- *Strive for Consistent*

Urutan aksis yang konsisten diperlukan pada situasi yang sama. Konsistensi juga harus digunakan pada *prompt*, menu, layar bantu. Warna tampilan, kapitalisasi, huruf dan sebagainya juga harus konsisten.

- *Cater to Univeral Usability*

Mengenal dan mengkategorikan kebutuhan pengguna yang beragam, memfasilitasi transformasi konten. Menambahkan fitur untuk pemula seperti penjelasan dan untuk pemakai yang sudah ahli seperti *shortcuts* dan *faster pacing* sehingga bisa memperkaya *design interface* dan meningkatkan kualitas sistem.

- *Offer Informative Feedback*

Untuk setiap tindakan yang dilakukan pemakai, diharapkan adanya respon dari sistem. Untuk tindakan yang sering terjadi dan tidak membutuhkan banyak aksi, respon yang ada bisa dibuat secara

sederhana, sedangkan tindakan yang jarang dilakukan dan membutuhkan banyak aksi harus lebih ditonjolkan.

- *Design Dialogs to Yield Closure*

Urutan aksi harus disusun ke dalam kelompok awal, tengah dan akhir suatu umpan balik yang informatif pada akhir pekerjaan sebaiknya dibuat untuk mengindikasikan bahwa pekerjaan telah selesai dan siap untuk melanjutkan ke aksi berikutnya.

- *Prevent Errors*

Sebisa mungkin dibuat sistem yang tidak akan menghasilkan kesalahan yang serius jika digunakan oleh pemakai. Jika pemakai membuat kesalahan, sistem tersebut harus dapat mendeteksi kesalahan dan menawarkan instruksi yang sederhana, konstruktif, dan spesifik untuk perbaikan.

- *Permit Easy Reversal of Actions*

Aksi yang telah dilakukan harus dapat dikembalikan ke keadaan awal.

- *Support Internal Locus of Control*

Menjadikan pemakai sebagai pemberi aksi, bukan sebagai orang yang merespon aksi.

- *Reduce Short-term Memory Load*

Karena keterbatasan manusia dalam pemrosesan informasi pada ingatan jangka pendek, sebaiknya tampilan dibuat sederhana, beberapa halaman dijadikan satu, frekuensi pergerakan *window*

dikurangi, dan harus ada waktu yang cukup untuk mempelajari kode-kode, singkatan, serta urutan aksi. Informasi seperti singkatan atau kode sebaiknya juga tersedia.

2.8 *E-Commerce (EC)*

E-Commerce adalah pengembangan konsep yang menggambarkan proses pembelian, penjualan, atau pertukaran produk, jasa, dan informasi melalui jaringan komputer, termasuk *Internet* (Turban,2002,p.4).

EC dijabarkan dalam berbagai sudut pandang antara lain:

- **Dari Sudut pandang Komunikasi**, EC adalah pengiriman barang, jasa, informasi, atau pembayaran melalui jaringan komputer atau sarana elektronik lainnya.
- **Dari Sudut pandang Bisnis Proses**, EC adalah aplikasi sebuah teknologi yang menuju pada otomatisasi terhadap bisnis transaksi dan alur kerja.
- **Dari Sudut pandang Pelayanan**, EC adalah alat yang menunjukkan keinginan perusahaan, konsumen, dan manajemen untuk mengurangi biaya jasa sekaligus meningkatkan kualitas barang dan mempercepat proses pengiriman.
- **Dari Sudut pandang Online**, EC menyediakan kemampuan untuk membeli dan menjual produk dan informasi pada internet dan pelayanan online lainnya.

2.9 Klasifikasi *E-Commerce* berdasarkan jenis Transaksi

E-Commerce diklasifikasikan berdasarkan jenis transaksinya. Berikut adalah tipe transaksi:

- *Business-to-Business* (B2B)

Jenis transaksi dari perusahaan atau organisasi yang satu ke yang lain.

- *Business-to-Consumer* (B2C)

Transaksi dari perusahaan yg menjual produk atau jasa ke konsumen.

- *Consumer-to-Consumer* (C2C)

Konsumen langsung menjual produk atau jasa ke konsumen lain.

- *People-to-People* (P2P)

Tipe transaksi spesial dari C2C dimana individual saling melakukan pertukaran produk atau jasa yang lebih detail.

- *Consumer-to-Business* (C2B)

Meliputi individual yang menggunakan internet untuk menjual produk atau jasa ke organisasi/perusahaan.

- *Intrabusiness (Organizational) EC*

Tipe EC yang mencakup semua aktifitas organisasi, biasanya dilakukan secara *intranet* atau *portal* perusahaan, yang meliputi pertukaran, pelayanan, atau informasi antara berbagai unit perusahaan.

- *Business-to-Employees (B2E)*
Merupakan subset kategori dari *intrabusiness* yang memberikan pelayanan, informasi, atau produk ke karyawan.
- *Government-to-Citizens (G2C)*
Tipe EC ini menghubungkan pemerintah dengan bisnis penduduknya.
- *Exchange-to-Exchange (E2E)*
E2E merupakan sistem formal yang menghubungkan pertukaran.
- *Collaborative Commerce*
C-Commerce merupakan aplikasi untuk kolaborasi elektronik antara rekan bisnis dan karyawan organisasi.
- *Mobile Commerce*
Ketika EC mengambil tempat di lingkungan wireless, maka dinamakan *mobile commerce* (m-commerce).