BAB 2

LANDASAN TEORI

2.1 Internet

2.1.1 Pengertian Internet

Istilah Internet berasal dari bahasa latin, *inter*, yang berarti ”*antara*”. Secara kata per kata Internet berarti jaringan antara atau penghubung. Memang itulah fungsinya, Internet menghubungkan berbagai jaringan yang tidak saling bergantung pada satu sama lain dengan sedemikian rupa, sehingga mereka dapat berkomunikasi.

Internet adalah jaringan komputer yang saling terhubung ke seluruh dunia tanpa mengenal batas teritorial, hukum dan budaya. Secara fisik dianalogikan sebagai jaring laba-laba (*The Web*) yang menyelimuti bola dunia dan terdiri dari titik-titik (*node*) yang saling berhubungan [2].

Node bisa berupa komputer, jaringan lokal atau peralatan komunikasi, sedangkan garis penghubung antar simpul disebut tulang punggung (*back-bone*) yaitu media *terestrial* (kabel, serat optic, microwave, radio link) maupun satelit. *Node* terdiri dari pusat informasi dan database, peralatan komputer dan perangkat interkoneksi jaringan serta peralatan yang dipakai pengguna untuk mencari, menempatkan dan bertukar informasi di Internet.

Menurut Lani Sidharta (1996) walaupun secara fisik Internet adalah interkoneksi antar jaringan komputer namun secara umum Internet harus dipandang sebagai suber informasi. Isi Internet adalah informasi, dapat dibayangkan sebagai suatu database atau perpustakaan multimedia yang sangat

besar dan lengkap. Bahkan Internet dipandang sebagai dunia dalam bentuk lain (maya) karena hamper seluruh aspek kehidupan di dunia nyata ada di Internet seperti bisnis, hiburan, olah raga, politik dan lain sebagainya.

Salah satu isu terpenting salah satu isu terpenting di Internet adalah penerapan *standar* komputerisasi terbuka (*open computing standart*). Karena *internetworking* dan Internet mengintegrasikan semua sistem, jenis dan tipe komputer yang ada di dunia, maka harus ada *standar* yang menjamin komputer dapat saling berbicara satu sama lain dalam bahasa yang sama. Menurut Drew Heywood (1966): *standar* bahasa komputer universal telah dikembangkan sejak 1969, terdiri dari serangkaian protocol komunikasi disebut *Transfer Control Protocol (TCP)* yang bertugas mengendalikan transmisi paket data, koreksi kesalahan dan kompresi data dan Internet Protocol (IP) yang bertugas sebagai pengenal (*identifier*) dan pengantar paket data ke alamat yang dituju.

Protokol TCP / IP menyatukan bahasa dan kode berbagai komputer di dunia, sehingga menjadi *standar* utama jaringan komputer. TCP / IP berkembang cepat dan kaya fasilitas karena bersifat terbuka dan bebas digunakan, oleh siapa pun. Oleh karena itu, hingga saat ini jaringan-jaringan tersebut saling terhubung di seluruh belahan dunia.

Seluruh jaringa yang tersebar di setiap penjuru dunia sudah terkondisikan dengan internet. Lalu lintas data Internet ke seluruh Amerika dan penjuru dunia saat ini disebarkan oleh jaringan pusat yang disebut VBNS (*Very High Speed Backbone Network Service*). Adapun seluruh biaya yang dikeluarkan di biayai oleh badal Ilmu Pengetahuan Nasional Amerika Serikat. Dari jaringa itu, komputer-komputer yang terhubung dengan internet akan berkomunikasi dan

saling mengirimkan data sehingga internet dapat dinikmati oleh seluruh umat manusia untuk berkomunikasi melalui komuputer baik di rumah maupun di kantor.

Dengan demikian, internet dapat didefinisikan dengan hubungan berbagai jenis komputer dan jaringan di dunia, yang berbeda sistem operasi maupun aplikasinya di mana hubungan tersebut memanfaatkan kemajuan media komunikasi (telepon dan satelit) yang menggunakan protocol *standar* dalam berkomunikasi yaitu protocol TCP / IP. Internet berfungsi sebagai media komunikasi dan informasi modern.

2.1.2 Pengertian World Wide Web

World Wide Web adalah fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen *web* disebut *web page* dan *link* dalam *web* membuat *user* bisa pindah dari satu *page* ke *page* lain (*hyper text*), baik antar *page* yang disimpan dalam *server* yang sama atau *server* di seluruh dunia. *Pages* diakses dan dibaca melalui *web browser* seperti *netscape navigator* atau *internet explorer.*

World Wide Web sering disingkat WWW atau *web* saja, yaitu sebuah sistem dimana informasi dapat kita nikmati dalam bentuk teks, gambar, suara, dan lain-lain serta ikut dapat diakses oleh perangkat lunak yang disebut *browser*. Informasi di *web* pada umumnya ditulis dalam bentuk HTML. Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam bentuk format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World). WWW dijalankan dalam *server* yang disebut HTTPD.

2.1.3 *Universal Resource Locator* (URL)

URL adalah konsep nama file standar yang diperluas dengan jaringannya. Nama file ini tidak hanya menunjukan direktori dan nama filenya saja, tetapi juga nama mesinnya dalam jaringan. URL dapat disediakan (ada atau diakses) dengan berbagai metode dan bisa jadi bukan sekedar file. Karena URL dapat menunjukan queri-queri dokumen dapat disimpan dalam database, hasil dari suatu perintah finger atau achie atau apapun yang berkaitan dengan data hasil proses.

2.1.4 *Domain Name Service* (DNS)

DNS digunakan untuk mempermudah penggunaan internet, dengan memetakan IP *address* ke nama host. Agar data nama host dapat didistribusikan di banyak server, format data yang digunakan harus mencerminkan terdistribusinya data tersebut. Untuk itu, digunakan format tree dengan masing-masing nodenya disebut domain. Penulisan nama host secara lengkap disebut sebagai *Full Qualifield Domain Name (FQDN).*

DNS bekerja secara hirarki dan berbentuk seperti pohon (tree). Bagian atas adalah *Top Level Domain* (TLD) seperti COM, ORG, EDU, MIL dsb. Seperti pohon DNS mempunyai cabang-cabang yang dicari dari pangkal sampai ke ujung. Pada waktu kita mencari alamat misalnya linux.or.id pertama-tama DNS bertanya pada TLD server tentang DNS Server yang melayani domain .id misalnya dijawab ns1.id, setelah itu dia bertanya pada ns1.id tentang DNS Server yang bertanggung jawab atas .or.id misalnya ns.or.id kemudian dia bertanya pada ns.or.id tentang linux.or.id dan dijawab 64.29.24.175

Sedangkan untuk mengubah IP menjadi nama *host* melibatkan domain in-addr.arpa. Seperti domain lainnya domain in-addr.arpa pun bercabang-cabang.

Yang penting diingat adalah alamat IP-nya ditulis dalam urutan terbalik di bawah in-addr.arpa. Misalnya untuk alamat IP 64.29.24.275 prosesnya seperti contoh linux.or.id: cari server untuk arpa, cari server untuk in-addr.arpa, cari server untuk 64.in-addr.arpa, cari server 29.64.in-addr.arpa, cari server untuk 24.29.64.in-addr.arpa. Dan cari informasi untuk 275.24.29.64.in-addr.arpa. Pembalikan urutan angkanya memang bisa membingungkan.

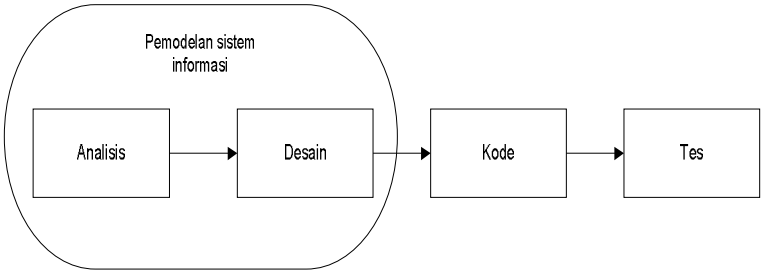
2.1.5 HTTP (*Hypertext Transfer Protocol*)

Server dan *browser web* berkomunikasi satu sama lain dengan protokol yang memang dibuat khusus untuk ini yaitu HTTP. HTTP bertugas menangani permintaan-permintaan *(request)* dan *browser* untuk mengambil dokumen-dokumen *web*. HTTP bisa dianggap sebagai sistem yang bermodel *client-server*.

2.2 SDLC *Model Sequential Linear*

System Development Life Cycle (SDLC) adalah suatu metodologi yang digunakan untuk mengembangkan, memelihara dan menggunakan perangkat Lunak (O'briend, 2001), metodologi penelitian ini mencakup sejumlah fase atau tahapan.

Linear Sequential Model (model sekuensial linear) dapat dilihat pada gambar 2.1, ditemukan pertama kali oleh Royce. Model ini sering disebut model klasik atau *waterfall.* Model ini menyarankan pendekatan pengembangan secara sekuen dan sistematik untuk pengembangan perangkat lunak [3]. Model ini merupakan model tertua. Model ini terdiri atas beberapa tahapan, yaitu rekayasa, dan pemodelan sistem informasi, analisis kebutuhan perangkat lunak, desain, generasi kode, pengujian dan pemeliharaan.

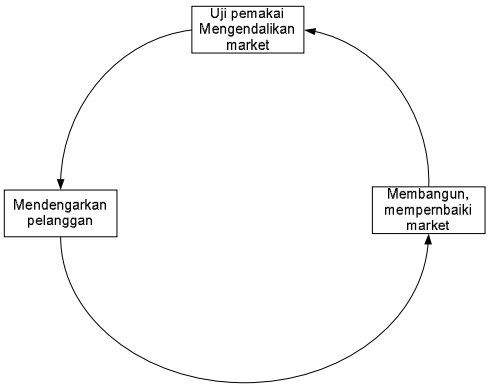


Gambar 2.1 Model Sequensial Linear [3]

Berikut metode pengembangan perangkat lunak lainnya :

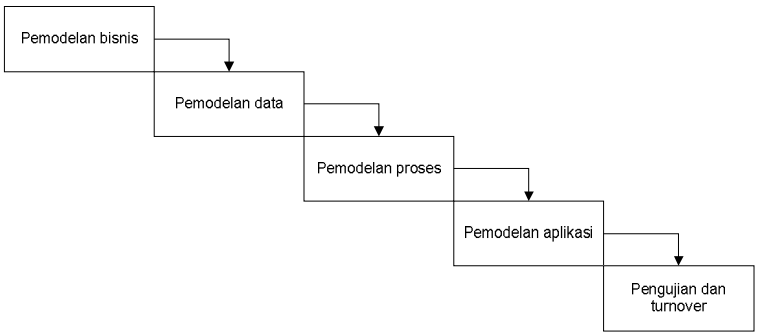
1. *Prototyping Model* (model paradigma) dimulai dengan pengumpulan kebutuhan, pengembangan dan pelanggan bertemu dan mendefinisikan objek keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui dan area garis besar dimana definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat [3]. Perancangan kilat membawa kepada konstruksi sebuah prototipe yang kemudian dievaluasi oleh pemakai. Dipakai untuk menyaring kebutuhan pengembangan perangkat lunak.

Siklus dari model prototipe dapat dilihat pada gambar 2.2 dibawah ini :



Gambar 2.2 Model Prototipe [3]

2. *Rapid Application Development* (RAD), adalah sebuah model proses pengembangan perangkat lunak sekuensial linear yang menekankan siklus pengembangan yang sangat pendek. Model RAD merupakan adaptasi "berkecepatan tinggi" dari linear sequential model pengembangan yang cepat dapat diperoleh dengan menggunakan pendekatan konstruksi berbasis komponen [3]. Lihat gambar 2.3 berikut :



Gambar 2.3 Model *Rapid Application Development* (RAD) [3]

3. *Evolutionary Software Process Model* adalah perulangan. Model ini dicirikan denga pengembang versi-versi sistem yang lebih lengkap sedikit demi sedikit [3]. Model telah mempertimbangkan untuk mengkoordinasikan evaluasi produk, secara lengkap model ini terdiri dari :
   a. *Incremental model*, model ini mengkombinasikan antara linear sequential model dengan filosofi iteratif pada *prototyping*. Pada masing-masing sekuen linear, yang menghasilkan perangkat lunak yang semakin meningkat kompleksitasnya.
   b. *Sipral model*, model ini digunakan oleh Boehm. Model ini menggabungkan antar sifat alami iterasi dati *prototyping* dengan aspek sistematik dan terkendali dari linear sequential model. Model ini memberi peluang untuk pengembangan cepat.

2.3 E-commerce

2.3.1 Definisi *e-commerce*

Definisi *e-commerce* menurut kamus Wikipedia adalah Perdagangan elektonik atau e-dagang (bahasa Inggris: *electronic commerce*) yaitu penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti internet atau televisi, www, dan jaringan komputer lainnya [2]. E-dagang dapat melibatkan transfer dan elektronik, pertukaran data elektronik, sistem manajemen inventaris otomatis, dan sistem pengumpulan data otomatis.

Menurut Mariza Arfina dan Robert Marpaung, *e-commerce* atau yang lebih dikenal *e-com* dapat diartikan sebagai cara berbelanja atau berdagang secara *online* atau *direct selling* yang memanfaatkan fasilitas Internet di mana terdapat *website* yang dapat menyediakan layanan "*get and deliver*" [2].

Definisi *e-commerce* yang sudah distandarkan dan disepakati bersama adalah "*E-commerce is a dynamic set of technologies, applications, and business process that link enterprise, consumers, and communities through electronic transactions and the electronic exchange of goods, services, and information*". *E-Commerce* merupakan satu set dinamis teknologi, aplikasi dan proses bisnis yang menghubungkan perusahaan, konsumen, dan komunitas tertentu melalui transaksi elektronik dan perdagangan barang, pelayanan, dan informasi yang dilakukan secara elektronik [2].

Dengan demikian, dapat disimpulkan bahwa *e-commerce* merupakan proses pembelian dan penjualan jasa atau produk antara dua belah pihak melalui Internet (*Commerce net*) dan sejenis mekanisme bisnis elektronik dengan fokus pada transaksi bisnis berbasis individu dengan menggunkan internet sebagai

media pertukaran barang atau jasa baik antar instansi atau individu dengan instansi (*Net-Ready*).

2.3.2 Jenis *e-commerce*

Dengan melihat keuntungan-keuntungan diatas, maka kita merasakan bahwa *e-commerce* merupakan suatu keharusan. Akan tetapi kita tidak dapat langsung terjun ke dalam tanpa mengerti pola bisnis atau model bisnis yang menjadi berubah dengan adanya *e-commerce*. *E-commerce* dapat dibagi menjadi enam jenis yaitu *Business to Business* (B2B), *Business to Consumer* (B2C), *Consumer to Consumer* (C2C), *Consumer to Business* (C2B), *Non-Business Electronic Commerce*, dan *Intrabusiness (Organizational) Electronic Commerce*. Keenam *e-commerce* tersebut mempunyai karakteristik yang berbeda [10].

1. *Business to Business* (B2B)
    a. *Trading Partner* yang sudah diketahui dan umumnya memiliki hubungan yang cukup lama. Informasi hanya dipertukarkan dengan partner tersebut sehingga jenis informasi yang dikirimkan dapat disusun sesuai kebutuhan dan kepercayaan.
    b. Pertukaran data (*data exchange*) berlangsung berulang-ulang dan secara berkala, dengan format data yang sudah disepakati bersama sehingga memudahkan pertukaran data untuk dua entitas yang menggunakan standar yang sama.
    c. Salah satu pelaku dapat melakukan inisiatif untuk mengirimkan data, tidak harus menunggu *partner*.
    d. Model yang umum digunakan adalah *per-to-per* dimana *processing intelligence* dapat didistribusikan di kedua pelaku bisnis.

2. *Business to Consumer* (B2C)

   Business to Consumer merupakan transaksi ritel dengan pembeli individual. Selain itu *Business to Consumer* juga dapat berarti mekanisme toko online (*electronic shoping mall*) yaitu transaksi antara *e-merchant* dengan *e-customer*.

   Karakteristik *Business to Consumer* adalah sebagai berikut:

   a. Terbuka untuk umum dimana informasi disebarkan ke umum.
   b. Layanan yang diberikan bersifat umum dengan mekanisme yang dapat digunakan oleh khalayak ramai. Contohnya layanan yang diberikan menggunakan *website*.
   c. Layanan diberikan berdasarkan permohonan (*on demand*). Konsumen melakukan inisiatif dan produsen harus siap memberikan respon sesuai permohonan.
   d. Pendekatan *client-server* sering digunakan di mana diambil asumsi klien (c*onsumer*) menggunakan sistem yang minimal (berbasis *web*) dan *processing* diletakan di sisi *server*.

3. *Consumer to Consumer* (C2C)

   Consumer to Consumer merupakan transaksi di mana konsumen menjual produk secara langsung kepada konsumen lainnya. Selain itu, seorang individu juga bisa mengiklankan produknya berupa barang atau jasa, pengetahuan, atau keahliannya di salah satu situs lelang.

4. *Consumer to Business* (C2B)

   Consumer to Business merupakan individu yang menjual produk atau jasa kepada organisasi dan individu yang mencari penjual dan melakukan transaksi.

5. *Non-Business Electronic Commerce*

   Non-Business Electronic Commerce meliputi kegiatan non bisnis seperti kegiatan lembaga pendidikan, organisasi nirlaba, keagamaan dan lain-lain.

6. *Intrabusiness (Organizational) Electronic Commerce*

   Kegiatan ini meliputi semua aktivitas internal organisasi melalui internet untuk melakukan pertukaran barang, jasa, dan informasi serta menjual produk perusahaan kepada karyawan, dan lain-lain.

2.3.3 Sistem transaksi *e-commerce*

Untuk saat ini sistem transaksi di *e-commerce* bisa dibilang sudah lengkap dan terbilang aman untuk melakukan transaksi. Karena dengan adanya kemanan ini pelanggan dapat tertarik untuk melakukan transaksi secara *online* di internet. Beberapa model pembayaran *online* di internet untuk saat ini yang masih sering digunakan sebagai berikut :

1. Paypal

   Paypal merupakan salah satu alat pembayaran (*payment process*) teraman dan terbanyak digunakan di dunia Internet. Saat ini, orang Indonesia sudah dapat memindahkan uang yang berada di *account Paypal* langsung ke rekeningnya. *Paypal* mengatasi kekurangan dalam pengiriman uang tradisional seperti cek atau *money order* yang prosesnya memakan waktu yang lama. *Paypal* tidak berbeda dengan rekening bank. Hal ini pertama yang perlu Anda lakukan

adalah membuat *account* tersebut dengan dana dari kartu kredit atau mentransfer dana dari *account Paypal* orang lain ke *balance Paypal* Anda. Jika proses ini sudah diselsaikan, maka *account Paypal* sudah dapat digunakan untuk bertransaksi.

2. Kartu Kredit

   Kartu kredit dapat digunakan sebagai alat pembayaran dengan terlebih dahulu mendaftarkan diri Anda sebagai konsumen pada *web* tersebut. Selain kartu kredit sebagai alat pembayaran alternatif, Anda bisa menggunaka *e-cash,* yaitu sebuah *account* khusus untuk pembayaran melalui Internet. *Account* tersebut dibuka dengan kartu kredit yang dimiliki sebelumnya. Konsumen hanya perlu mengisi *account e-cash*-nya untuk digunakan.

2.3.4 Keamanan *e-commerce*

Untuk menjalankan *e-commerce*, dibutuhkan tingkat keamanan yang dapat diterima. Salah satu cara untuk meningkatkan keamanan adalah dengan menggunakan teknologi kriptografi, yaitu antara lain dengan menggunakan enkripsi untuk mengacak data. Salah satu metoda yang mulai umum digunakan adalah pengamanan informasi dengan menggunakan *public key system*. Sistem lain yang bisa digunakan adalah *private key system*. Infrastruktur yang dibentuk oleh sistem *public key* ini disebut *Public Key Infrastructure* (PKI), atau diterjemahkan dalam Bahasa Indonesia menjadi Infrastruktur Kunci Publik (IKP), dimana kunci publik dapat dikelola untuk pengguna yang tersebar (di seluruh dunia). Diantara komponen-komponen dari infrastruktur kunci publik ini adalah :

1. *Certification Authority* (CA)

   Merupakan sebuah *entity* yang memberikan dan mengelola sertifikat digital yang dibutuhkan dalam transaksi elektronik. CA berhubungan erat dengan pengelolaan *public key system*. Contoh sebuah CA di Amerika adalah Verisign (www.verisign.com).

2. IP *Secure*

   Keamanan media komunikasi merupakan hal yang penting. Mekanisme untuk mengamankan media komunikasi yang aman (*secure*) selain menggunakan SSL adalah dengan menggunakan IP Secure.

3. *Secure Socket Layer* (SSL)

   Merupakan teknologi yang telah lama dipakai dalam jaringan internet. SSL merupakan salah satu teknologi yang paling banyak dipakai. SSL adalah teknologi yang matang dan stabil tapi tidak berarti SSL menjadi jawaban bagi semua permasalahan keamanan dalam *e-commerce*. *Secure Socket Layer* (SSL) adalah protokol yang digunakan untuk *browsing web* secara aman. Dalam hal ini, SSL bertindak sebagai protokol yang mengamankan komunikasi antara *client* dan *server*. Protokol ini memfasilitasi penggunaan enkripsi untuk data yang rahasia dan membantu menjamin integritas informasi yang dipertukarkan antara *website* dan *web browser*. SSL dikembangkan oleh *Netscape Communications* pada tahun 1994, dan menjadi protokol yang umum digunakan untuk komunikasi aman antara dua komputer pada internet. SSL dibangun ke dalam beberapa *web browser*. Cara kerja SSL sebagai berikut :

a. Pada saat koneksi mulai berjalan, *client* dan *server* membuat dan mempertukarkan kunci rahasia, yang dipergunakan untuk mengenkripsi data yang akan dikomunikasikan. Meskipun sesi antara *client* dan *server* diintip pihak lain, namun data yang terlihat sulit untuk dibaca karena sudah dienkripsi.
b. SSL mendukung kriptografi *public key*, sehingga *server* dapat melakukan autentikasi dengan metode yang sudah dikenal umum seperti *RSA* dan *Digital Signature Standard (DSS)*.
c. SSL dapat melakukan verifikasi integritas sesi yang sedang berjalan dengan menggunakan algoritma *digest* seperti MD5 dan SHA. Hal ini menghindarkan pembajakan suatu sesi.

2.4 Basis Data

2.4.1 Definisi basis data

Basis data dapat diartikan sebagai kumpulan dari beberapa file yang sejenis. Menurut *Hanafin dan Pock* (1998:76) basis data mengandung arti "*Sekumpulan data yang saling berhubungan dan berkaitan satu dengan yang lainnya digunakan oleh suatu organisasi*".

Dalam hal ini data dan informasi berupa perangkat keras (*hardware* yang digunakan) dan dipakai untuk dialokasikan dalam pembuatan sistem informasi pengolahan *database*.

2.4.2 Sistem manajemen basis data

Sistem manajemen basis data (*database management sistem/DBMS)* adalah perangkat lunak yang digunakan untuk mengendalikan data, termasuk penyimpanan data, pegambilan data, keamanan data, dan integritas data. Dimana

dapat digunakan oleh pengguna baik secara *single user* maupun *multiple user* sesuai dengan kepentingan dan kebutuhan pengguna.

Fungsi utama DBMS adalah menyediakan lingkungan yang nyaman dan efisien untuk digunakan dalam pengambilan dan penyimpanan informasi di basis data.

2.4.3 Operasi dasar basis data

Operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat meliputi :

1. Pembuatan basis data baru (*create database*)
2. Penghapusan basis data (*drop database*)
3. Pembuatan file/table baru ke suatu basis data (*create table*)
4. Penghapusan file/table dari suatu basis data (*drop table*)
5. Penambahan atau pengisian data baru kesebuah file/table di sebuah basis data (*insert*)
6. Pengambilan data dari sebuah file/table (*retrieve* atau *search*)
7. Pengubahan data dari sebuah file/table (*update*)
8. Penghapusan data dari sebuah file/table (*delete*)

2.4.4 Arsitektur sistem basis data

Terdapat dua bentuk arsitektur sistem basis data, yaitu sistem terpusat dan *Client-Server*.

Sistem basis data terpusat adalah sistem basis data yang dijalankan pada sistem komputer tunggal dan tidak berinteraksi dengan sistem pada komputer lain. Pengguna terkoneksi ke komputer pusat melalui terminal.

Sistem basis data *Client-Server* adalah sistem basis data yang memisahkan program pengguna dengan program basis data disistem yang berbeda. Pengguna terkoneksi ke pusat data yang disebut *Server* sistem melalui suatu program pengguna (*user interface*) yang terdapat pada *personal computer*. Sistem tempat program pengguna berada disebut *Client System*.

2.5 Konsep dasar analisis sistem

Analisis Sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mendefinisikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan.

Didalam tahap analisis sitem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem sebagai berikut:

1. *Identify*, yaitu mendefinisikan masalah
    a. Mengidentifikasi penyebab masalah
    b. Mengidentifikasi titik keputusan
    c. Mengidentifikasi personil-personil kunci
2. *Understand*, yaitu memahami kerja dari sistem yang ada
    a. Menentukan jenis penelitian
    b. Merencanakan jadwal penelitian
    c. Mengatur jadwal wawancara
    d. Mengatur jadwal observasi
    e. Mengatur jadwal pengambilan *sample*
    f. Membuat penugasan penelitian

g. Membuat agenda wawancara

h. Mengumpulkan hasil penelitian

3. *Analyze*, yaitu menganalisis sistem

a. Menganalisis kelemahan sistem

b. Menganalisis kebutuhan informasi pemakai/manajemen

4. *Report*, yaitu membuat laporan hasil analisis

Tujuan :

a. Pelaporan bahwa analisis telah selesai dilakukan

b. Meluruskan kesalah-pengertian mengenai apa yang telah ditemukan dan dianalisis oleh analis sitem tetapi tidak sesuai menurut manajemen

c. Meminta pendapat-pendapat dan saran-saran dari pihak manajemen

d. Meminta persetujuan kepada pihak manajemen untuk melakukan tindakan selanjutnya.

Model analisis, merupakan representasi teknis yang pertama dari sistem. Selama ini ada dua pemodelan analisis yang telah banyak diusulkan, yaitu:

1. Analisis terstruktur, pembangunan model dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional, kita menciptakan model yang menggambarkan muatan dan aliran informasi (data dan kontrol), kita membagi sistem secara fungsional dan secara behavioral, dan menggambarkan esensi dari apa yang harus dibangun.
2. Analisis berorientasi objek, mendefinisikan semua kelas yang relevan terhadap masalah beserta operasi-operasi dan atribut-atribut yang diasosiasikan dengan kelas itu, keterhubungan dikelas-kelas dan perilaku yang dimilikinya. Sasaran analisis berorientasi objek adalah untuk

mengembangkan model yang mendeskripsikan perangkat lunak yang memenuhi sekelompok kebutuhan yang didefinisikan pemesan. Analisis berorientasi objek menggunakan sejumlah pemodelan untuk memenuhi sasaran. Model analisis akan mengekspresikan informasi, perilaku dan fungsi didalam konteks model objek.

Alat-alat yang digunakan dalam analisi terstruktur adalah :

1. Diagram E-R (*Entity Relationship Diagram*)
2. Diagram konteks (*Context Diagram*)
3. Diagram arus data (*Data Flow Diagram*)
4. Kamus data (*Data Dictionary*)

2.5.1 Diagram E-R (*Entity Relationship Diagram*)

Diagram E-R adalah diagram grafikal yang menggambarkan keseluruhan struktur *logic* dari sebuah basis data. Pada model ini semua data yang ada pada dunia nyata diterjemahkan atau ditransformasikan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data [11].

Untuk melambangkan fungsi diatas maka digunakan simbol-simbol yang memiliki arti sebagai berikut :

a. Data Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Data entitas adalah segala sesuatu baik yang nyata maupun abstrak yang datanya akan direkam.

b. Atribut

Atribut mendefinisikan karakteristik (*property*) dari entitas tersebut.

c. Relasi Data

Relasi menunjukan adanya hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

d. Kardinalitas/Derajat Relasi

Kardinalitas menunjukan jumah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Terdapat tiga kemungkinan kombinasi yang ada pada kardinalitas :

1. Satu ke Satu (*One to One*)

   Setiap entitas pada himpunan entitas A berhubungan palingan banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

2. Satu ke Banyak (*One to Many*)

   Setiap entitas pada himpunan entitas A berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya dimana setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

3. Banyak ke Satu (*Many to One*)

   Setiap entitas pada himpunan A berhubungan paling banyak satu entitas pada himpunan entitas B, tetapi tidak sebaliknya dimana setiap entitas pada himpunan entitas B berhubungan dengan banyak entitas pada entitas himpunan A.

4. Banyak ke Banyak (*Many to Many*)

Setiap entitas pada himpunan entitas A berhubungan dengan banyak entitas pada himpunan entitas B, dan begitu juga sebaliknya dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

2.5.2 Diagram Konteks (*Context Diagram*)

Diagram konteks merupakan alat-alat untuk struktur analisis. Pendekatan struktur ini mencoba untuk menggambarkan sistem secara garis besar atau secara keseluruhan. Diagram konteks ialah kasus khusus dari DFD atau bagian dari DFD yang berfungsi memetakan modul lingkungan yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

Diagram konteks bisa disebut dengan *"Model sistem pokok (Fundamental System Model) mewakili keseluruhan elemen software dengan masukan (input) dan keluaran (output) yang diidentifikasikan dengan anak panah masuk dan keluar memperlihatkan sumber data"* [3].

2.5.3 Diagram Arus Data (*Data Flow Diagram*)

DFD atau singkatan dari *Data Flow Diagram* merupakan representasi grafik dari suatu sistem yang menunjukan proses atau fungsi, aliran data, tempat penyimpan data dan *entitas eksternal*. DFD juga digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan. Dengan menggunakan DFD, rancangan yang akan kita buat akan lebih terarah dan lebih rinci. Sehingga kita tidak akan mengalami kesulitan dalam melakukan perancangan. *Data flow diagram* memiliki empat komponen, antara lain akan dijelaskan dibawah ini [6] :

1. *Terminator (external exitity)*

   Terminator mewakili *entity external* yang berkomunikasi dengan sistem yang sedang dikembangkan. *Terminator* merupakan kesatuan dilingkungan sistem. Biasanya terminator ini dikenal dengan nama entitas (*external*) sumber atau tujuan (*source* dan *sink*). Terminator dapat juga berupa departemen, divisi atau sistem diluar sistem yang berkomunikasi dengan sistem yang dikembangkan.

2. Proses

   Proses sering dikenal dengan nama *Bubble*, fungsi atau informasi. Komponen proses menggambarkan bagian dari sistem yang mentransformasikan *input* ke *output*, atau dapat dikatakan bahwa komponen proses menggambarkan transformasi satu *input* atau lebih menjadi *output*. Setiap proses harus diberikan penjelasan yang lengkap sebagai berikut :

   a. Identifikasi Proses

      Umumnya berupa angka yang menunjukan nomor dari proses atau ditulis pada bagian atas simbol proses.

   b. Nama Proses

      Menunjukan apa yang dikerjakan oleh proses tersebut. Nama proses harus jelas dan lengkap menggambarkan bagian prosesnya nama proses diletakan dibawah identifikasi proses.

3. Penyimpanan data *(data store)*

   Data store digunakan sebagai sarana untuk mengumpulkan data. *Data store* disimbolkan dengan dua garis horizontal yang paralel dimana tertutup pada salah satu ujungnya atau dua garis horizontal saja. *Data store* ini

biasanya berkaitan dengan penyimpanan-pemyimpanan seperti *file* atau *database* yang berkaitan dengan penyimpanan secara komputerisasi, contohnya *file* pita magnetic, *file* disket atau *file harddisk*. *Data store* juga berkaitan dengan pemyimpanan data.

4. Alur Data *(Data Flow)*

Suatu data *flow* atau alur data dapat dipresentasikan dengan anak panah yang menunjukan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau satu paket data atau informasi dari suatu bagian sistem ke bagian lainnya. Selain menunjukan arah, alur data pada model yang dibuat dapat merepresentasikan bit, karakter, pesan, formulir, bilangan *real* dan macam-macam informasi yang berkaitan dengan komputer.

2.5.4 Kamus Data (*Data Dictionary*)

Kamus data adalah suatu daftar atau kamus dari seluruh elemen-elemen data yang diperlukan oleh suatu sistem. Kamus data dibuat berdasarkan arus data yang terdapat pada DFD. Arus data dalam DFD bersifat global sehingga hanya dapat ditunjukkan nama arus datanya saja dan keterangan lebih lanjut mengenai struktur arus data dapat dilihat dari kamus data [5].

2.6 Teknik Pengujian Perangkat Lunak *Black Box*

Pengujian *black box* berfokus pada pertanyaan fungsional perangkat lunak. Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*

3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi.

2.7 Pengenalan HTML

Dokumen HTML adalah file teks regular (disebut juga ASCII) yang diciptakan dengan menggunakan editor teks (misalnya, Emacs dalam Unix, Gnotepad pada Linux, atau Notepad pada Windows) atau dengan menggunakan word processor, tetapi harus disimpan dengan format *"text only with line breaks"*.

Sesuai dengan namanya, bahasa ini menggunakan tanda (MarkUp) untuk menandai perintah-perintahnya [12]. Saat ini banyak sekali aplikasi-aplikasi yang dapat digunakan untuk membuat web page, seperti Microsof Frontpage, Adobe GoLive, Macromedia Dreamweaver, Ultradev dan lain-lainnya. Namun untuk seorang web develover kemampuan dasar perintah HTML tetap dibutuhkan.

Homepage dibangun oleh sebuah file yang didalamnya terdapat kode-kode dan informasi dan dibaca oleh sebuah aplikasi yang bernama browser. Oleh browser, kode dan informasi akan disajikan sebagai sebuah halaman web yang kita kenal sebagai homepage. Kode tersebut haruslah ditulis menurut suatu aturan tertentu yang dapat dimengerti browser. Aturan tersebut dikenal dengan "bahasa pengkodean".

Untuk membuat homepage website, bahasa pengkodean yang digunakan adalah HTML. Sekalipun banyak orang menyebutnya sebagai salah satau bahasa pemrograman, HTML sebenarnya sama sekali bukan bahasa pemrograman karena HTML merupakan bahasa Mark Up (pengkodean) yang digunakan untuk menentukan format atau style dan teks yang ditandai. Simbol Mark Up yang

digunakan oleh HTML ditandai dengan tanda lebih kecil (<) dan lebih besar (>), keduanya dinamai tag. Sebagai contoh akan ditampilkan teks yang tercetak tebal, maka mark up yang digunakan adalah sebagai berikut:

<B> Teks ini bercetak tebal </B>

Untuk menandai bahwa sebuah file teks merupakan file HTML, maka ciri yang terlihat jelas adalah ekstensi filenya yaitu .htm atau .html. Namun labih jauh dari pada itu didalam file tersebut harus mengandung struktur sebagai berikut:

<HTML>

…….

…….

</HTML>

Tag <HTML> harus diletakan pada bagian awal dan tag </HTML> harus diletakan pada bagian terakhir dari suatu file HTML. Tag-tag HTML tidak bersifat case sensive artinya penggunaan huruf besar maupun huruf kecil tidak menjadi masalah. Jadi tag <HTML> akan sama dengan tag <html>.

2.8 *Structure Query Language* (SQL)

SQL adalah bahasa standar yang digunakan untuk *Database Management System* untuk memanipulasi dan memperoleh data dari sebuah *database* relasional [13]. Dengan SQL kita dapat melakukan hal-hal berikut terhadap basisdata :

a. Pengubahan struktur sebuah *database*
b. Pengubahan pengaturan keamanan sistem
c. Memberikan hak akses kepada pengguna
d. Memperoleh informasi data *database*
e. Mengupdate *database*

Perintah-perintah SQL dikelompokan menjadi lima macam yaitu :

1. *Data Definition Language* (DDL)

   Adalah perintah SQL yang digunakan untuk mendefinisikan kerangka basisdata, perintah-perintahnya adalah :

   a. *create* : untuk membuat atau menciptakan objek basisdata
   b. *alter* : untuk memodifikasi atau mengubah objek basisdata
   c. *drop* : untuk menghapus objek basisdata
   d. Objek *databse* yang dimaksud adalah basisdata, *table*, *index*

2. *Data Manupulation Language* (DML)

   Adalah perintah yang digunakan untuk mengoprasikan atau memanipulasi isi basisdata, SQL menyediakan 4 perintah DML yaitu :

   a. *select* : digunakan untuk mengambil data dari basisdata
   b. *delete* : digunkan untuk menghapus data pada basisdata
   c. *insert* : digunkan untuk menambahkan data kedalam table
   d. *update* : digunakan untuk memodifikasi data pada basisdata

3. *Security*

   Adalah perintah-perintah yang digunakan untuk menjamin keamanan data. Perintahnya antara lain :

   a. *grant* : digunakan untuk memberikan akses kepada user tertentu ke basisdata
   b. *revoke* : digunakan untuk mencabut hak akses dari user

4. *Integrity*

   Adalah perintah-perintah yang digunakan untuk menjaga kesatuan data. Contoh : *recover table* : digunakan untuk memperbaiki *table* basisdata.

5. *Auxilliary*

   Adalah perintah-perintah pelengkap atau tambahan seperti : *unload* dan *rename*.

2.9 MYSQL

MySQL adalah multi *user database* yang menggunakan bahasa *Strured Query Language* (SQL). MySQL merupakan database server multi user dan multi *threaded* yang tangguh. Dengan memiliki banyak feature MySQL bisa bersaing dengan datgabase komsersial sekalipun. Tidak mengejutkan MySQL menjadi database pilihan untuk banyak pengguna PHP. MySQL termasuk RDBMS (*Relational Database Management System*) itulah sebabnya istilah seperti tabel, baris dan kolom digunakan pada MySQL. Pada MySQL sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri dari sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

MySQL mampu menangani data yagn cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data. SQL adalah sebuah standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industry.

Dengan menggunakan MySQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan menggunakan *dBASE* atau *Clipper* yang masih menggunakan perintah-perintah pemrograman. MySQL merupakan *software database* yang masih yang masih popular di lingkungan Linux, kepopuleran ini sangat ditunjang oleh performa *query* dari databasenya yang saat ini bisa

dikatakan paling cepat dan jarang bermaslah. MySQL ini juga sudah berjalan pada lingkungan windows.

2.9.1 Keistimewaan MySQL

Sebagai *database server* yang memiliki konsep *database modern*, MySQL memiiki banyak keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL.

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBDS, Mac OS dan lain-lain.

2. *Open Source*

   MySQL didistribusikan secara *Open Source*, sehingga dapat digunakan secara bebas.

3. *Multi-user*

   MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah. Hal ini memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan.

4. *Performance Tuning*

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

5. *Column Types*

   MySQL memiliki kolom yang sangat komplek, seperti *signed/unsigned integer*, *float, double*, *char*, *varchar, text, blob, date*, *time, timestamp, year, set* dan *enum*.

6. *Command* dan *Function*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam *query*.

7. *Security*

   MySQL memiliki lapisan-lapisan sekuritas seperti level subnetmask, nama *host*, dan izin akses *user* dengan *system* perizinan yang mendetail serta *password* terenkripsi.

8. *Scability* dan *Limits*

   MySQL mampu menangani *database* dengan sekala besar, dengan jumlah *records* lebih dari 50 juta dan 60 juta *table* serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.

9. *Connectivity*

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

10. *Localization*

    Deteksi pesan kesalahan pada *client* dengan menggunakan lebih dari 20 bahasa.

11. *Interface*

    Interface terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunkan fungsi API (*Application Programming Interface*).

12. *Client* dan *Tools*

    Dilengkapi dengan *tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

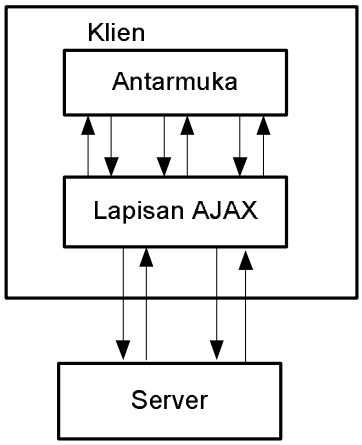
13. Struktur Tabel

Struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibanding *database* lainnya semacam PostgreSQL ataupun Oracle.

2.10 AJAX (*Asynchronus JavaScript and XML*)

AJAX (*Asynchronous JavaScript and XML*) sesungguhnya bukanlah suatu bahasa pemrogarman, melainkan hanya suatu teknik yang memungkinkan untuk membuat aplikasi *web* yang interaktif. Dengan menggunakan AJAX, aplikasi *web* dapat berinteraksi dengan server di latar belakang sehingga tidak mempengaruhi halaman *web* secara keseluruhan. Sebagai gambaran, Anda mungkin sering mendaftarkan diri pada suatu situs *web* sebagai anggota baru. Setelah Anda selesai mengetikan *user ID* saat registrasi, sistem bisa memberitahukan bahwa nama yang Anda pakai sudah digunakan oleh orang lain atau belum.

Bagaimana aplikasi *web* ini bisa tahu hal itu padahal Anda belum mengklik tombol apapun dalam halaman *web*. Rahasianya terletak pada pemrosesan di latar belakang yang dilakukan oleh AJAX. Gambar 2.5 Menunjukan aplikasi *web* yang berbasis AJAX.



Gambar 2.4 Model Aplikasi Web dengan AJAX [9]

Nampak bahwa pada aplikasi *web* yang berbasis AJAX terdapat bagian yang disebut lapisan AJAX (atau terkadang disebut mesin AJAX). Lapisan inilah yang mengatur komunikasi antara *client* dan *server*.

Butir penting yang perlu diperhatikan dalam lapisan AJAX :

1. Lapisan AJAX meminta layanan ke *server* atau tidak, tergantung pada jenis layanan antarmuka. Sebagai contoh, untuk validasi terhadap objek formulir tidak perlu melakukan permintaan ke *server*. Cukup ditangani di sisi *client*. Namun untuk pemeriksaan keabsahan *password*, mau tidak mau harus dilayangkan ke *server* karena harus berhubungan dengan database.
2. Sekalipun penambahan lapisan AJAX berkesan memperlambat proses (karena komunikasi ke server tidak lagi dilakukan secara langsung), pada kenyataanya yang terjadi adalah sebaliknya. Hal ini karena permintaan yang dilakukan oleh lapisan AJAX menghasilkan informasi yang jauh lebih sedikit dibanding kalau yang diminta adalah keseluruhan halaman.
3. Lapisan AJAX bekerja dengan pendekatan asinkron. Asinkron berarti bahwa *client* bisa meminta layanan dari *server* dan tidak perlu menunggu sampai *server* melayaninya. Dengan demikian, saat informasi dari *server* belum didapat, layanan kepada pemakai tetap bisa dilakukan.
4. Komunikasi antara *client* dan *server* tidak perlu mengubah keseluruhan antarmuka dalam halaman *web*, tetapi cukup hanya bagian tertentu yang terpengaruh. Hal seperti ini tidak bisa dilakukan pada aplikasi *web* yang tidak menggunakan AJAX.

AJAX pertama kali ditemukakan pada tahun 2005 oleh Jesse James Garret, presiden dan pendiri perusahaan Adaptive Path. Menurut beliau, nama AJAX

diberikan untuk memudahkan dalam berkomunikasi dengan klien daripada kalau menyebutnya "*Asynchronous JavaScript + CSS + DOM + XMLHttp Request*". Teknik ini sebenarnya sudah ada sejak 1999. Pada masa itu Microsoft membuat objek bernama *XMLHttpRequest* (Objek yang kelak akan dilibatkan dalam membuat AJAX) sebagai control *ActiveX* pada Internet Explorer 5. AJAX menjadi sangat popular baru setelah digunakan intensif oleh Yahoo! Dan Google.

Menurut Jesse James Garret, di dalam artikelnya yang berjudul "*AJAX A New Approach to Web Applications*", AJAX yang lengkap mencakup [9]:

1. XHTML (*Extensible HyperText Markup Language*), sebagai peluasan dari HTML.
2. CSS (*Cascading Style Sheets*) sebagai pemformat bagian dari dokumen.
3. DOM (*Document Object Model*) untuk mengatur interaksi dan penampilan isi HTML dan XHTML secara dinamis.
4. XML (*Extensible Markup Language*) memungkinkan untuk melakukan pertukaran data.
5. XSLT (*Extensible Stylesheet Language Transformation*) adalah bahasa berbasis XML yang ditujukan untuk melakukan transformasi dokumen XML ke bentuk yang lain.
6. *XMLHttpRequest* untuk melakukan pertukaran data secara asinkron.
7. *JavaScript* sebagai bahasa untuk membentuk lapisan AJAX.

Memang nampaknya sangat kompleks karena melibatkan banyak hal. Namun tidak pelu khawatir bahwa hal itu akan menyusahkan dalam mewujudkannya. Dalam praktik, AJAX bisa diimplementasikan tanpa menggunakan XML, XSLT, maupun CSS.

Software yang digunakan untuk membangun aplikasi *web* dengan AJAX :

1. *Browser* (misalnya *Internet Explorer* dan *Mozilla Firefox*) pada sisi klien.
2. *Web Server* (misalnya *Apache*) pada sisi server.

Tentu saja yang dicantumkan di atas adalah kebutuhan minimal untuk membuat aplikasi *web* yang melibatkan AJAX. Jika aplikasi juga melibatkan database, di sisi server juga perlu dipasang *database server.* Selain itu, agar *web server* dapat berkomunikasi dengan *database server*, perantara seperti PHP diperlukan.

2.11 PHP (*Page Hypertext Preprosessor*)

PHP (Personal Home Page Tools) adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang berjalan disisi *server* [14]. Sebagian besar perintahnya berasal dari bahasa *C, Java* dan *Perl* dengan beberapa tambahan fungsi khusus PHP.

Bahasa ini memungkinkan para pembuat aplikasi web yang menyajikan HTML yang dinamis dan interaktif dengan cepat dan mudah, yang di hasilkan server. PHP biasa berinteraksi dengan hampir semua teknologi *web* yang sudah ada. *Developer* biasa menulis sebuah program PHP yang mengeksekusi suatu program CGI di *server* lain. Fleksibilitas ini amat bermanfaat bagi pemilik situs-situs web yang besar dan sibuk, karena pemilik masih menggunakan aplikasi-aplikasi yang sudah terlanjur dibuat dimasa lalu dengan menggunakan CGI, ISAP atau dengan *script* seperti *Perl, awk* atau *Phyton* selama proses migrasi ke aplikasi baru yang di buat dengan menggunakan PHP.

Kode program PHP menyatu dengan tag-tag HTML dalam satu file. Kode PHP diawali dengan tag <? Dan ditutup dengan tag ?>. File yang berisi tag HTML

dan kode PHP ini diberi ekstensi .php atau ekstensi lainnya yang ditetapkan pada *Apache/web server*. Berdasarkan ektensi ini, pada saat file diakses, server akan tahu bahwa file ini mengandung kode PHP. Server akan menerjemahkan kode ini dan menghasilkan output dalam bentuk tag HTML yang akan dikirim ke *browser client* yang mengakses file tersebut. Contohnya dapat dituliskan seperti ini :

```
<?
echo (“<html>
<head>
<title>Pemrograman PHP</title>
</head>
<body>
<center>Selamat menggunakan PHP</center>
</body>
</html>”)
?>
```

Dengan PHP, *developer* tidak perlu lagi berurusan dengan dua buah file yang terpisah seperti pada CGI. *Browser web* mengacu secara langsung ke file yang dituju, yang lalu dibaca oleh *server* sebagaimana file HTML statis biasa. Bedanya, sebelum dikirim balik ke *browser web*, *server web* memeriksa isi file dan menentukan apakah ada kode didalam file tersebut yang harus dieksekusi. Bila ada, kode-kode tersebut akan dieksekusi. Hasilnya akan dimasukkan kedalam dokumen yang sama. *Server web* bekerja secara langsung terhadap file yang bersangkutan tidak memanggil *script* terpisah seperti pada metode CGI. Seluruh kode di eksekusi di *server* (oleh karena itu disebut *server-side script*).

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya, yaitu :

1. *Script* (kode program) terintegrasi dengan file HTML, sehingga *developer* biasa berkonsentrasi langsung pada penampilan webnya.
2. Tidak ada proses *compiling* dan *linking.*
3. Berorientasi objek *(object Oriented).*
4. Sintaksis pemrogramannya udah dipelajari, sangat menyerupai *C* dan *Perl.*
5. Integrasi yang sangat luas ke berbagai *server database*. Menulis *web* yang terhubung ke *database* menjadi sangat sederhana. Database yang didukung oleh PHP : *Oralce, Sybase, mSQL, MySQL, Solid, ODBC, PostgreSQL, Adabas D, FilePRo, Velocis, Informic, dBase, UNIX dbm.*

2.12 *Cascading Style Sheet* (CSS)

Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman *web* yang ditulis dengan HTML dan XHTML [12]. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh World Wide Web Consortium (W3C).

CSS digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa markup lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi,

memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada stuktur isi.

CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh browser basis-suara atau pembaca layar), dan juga alat pembaca *braille*. Halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

2.13 Pengenalan XAMPP

XAMPP merupakan sebuah *tool* yang menyediakan beberapa paket perangkat lunak ke dalam satu buah paket. Dengan menginstal XAMPP tidak pertlu melakukan instalasi dan konfigurasi *web server* Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan konfigurasikannya secara otomatis.

XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan *project* non-profit yang di kembangkan oleh Apache Friends yang didirikan Kai ‘Oswalad’ Seider dan Kay Vogelgesang pada tahun 2002, *project* mereka ini bertujuan untuk mempromosikan penggunaan Apache *web server*.

2.14 Pengenalan *Macromedia Dreamweaver*

Macromedia Dreamweaver adalah sebuah HTML editor professional untuk mendesain secara visual dan mengelola situs *web* maupun halaman *web*. *Macromedia Dreamweaver* merupakan *software* utama yang digunakan oleh *Web Designer* maupun *Web Progremer* guna mengembangkan situs *web*. Ruang kerja,

fasilitas dan kemampuan *Macromedia Dreamweaver* mampu meningkatkan produktivitas dan efektivitas dalam desain maupun membangun situs *web*.

Salah satu kelebihan *Dreamweaver* adalah ruang kerja *Dreamweaver* serta *tool* yang tersedian dapat digunakan dengan sangat mudah dan cepat, sehingga dapat memudahkan membangun suatu *website* dengan cepat dan tanpa harus melakukan *coding*. *Dreamweaver* juga memiliki dua bentuk layer, yaitu bentuk halaman *Design* dan halaman *Code*. Dua bentuk layer ini akan mempermudah dalam menambahkan *script* yang berbasis PHP maupun javascript. *Dreamweaver*, selain mendukung pembuatan *web* yang berbasis HTML, juga dapat mendukung program–program *web* yang lain diantaranya PHP, ASP, Perl, Javascript, dan lain-lain.