

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Membangun sebuah *website* yang baik agar sesuai dengan kebutuhan yang diinginkan penulis dan program studi Psikologi Universitas Sahid Surakarta, maka diperlukan referensi *website* yang sama-sama menyajikan informasi seputar Psikologi. Referensi ini juga di ambil dari beberapa situs *website* perguruan tinggi dan lembaga psikologi lainnya yang selama ini menyajikan informasi tentang Psikologi. Berikut referensi *website* yang menjadi acuan, diantaranya:

2.1.1 Website Psikologi Zone

Website Psikologi Zone dapat diakses di url <http://www.psikologizone.com>, *website* ini adalah sebagai media berita psikologi yang fokus pada pemberitaan terkait kesehatan mental dan perilaku, baik dalam maupun luar negeri. Berikut tampilan *website* Psikologi Zone ditunjukkan pada gambar 2.1.



Gambar 2.1 Website Psikologi Zone

Website Psikologi Zone memiliki beberapa menu utama antara lain : menu *home*, menu kirim opini, menu kirim konsultasi, menu *events* psikologi, menu tes kepribadian MBTI, menu tentang kami, menu anak, menu industri, menu keluarga, menu kepribadian, menu klinis, menu kolom, menu konsultasi, menu pendidikan, menu remaja, menu sosial, menu *zeitgeist*.

2.1.2 *Website Program Studi Psikologi Universitas Muhammadiyah Surakarta (UMS)*

Website program studi Psikologi Universitas Muhammadiyah Surakarta (UMS) dapat diakses di url <http://www.psikologi.ums.ac.id>. *Website* ini memiliki beberapa menu utama antara lain : menu beranda, menu profil, menu fasilitas, menu berita, menu publikasi ilmiah, menu kemahasiswaan, menu alumni. Berikut tampilan *website* Program Studi Psikologi Universitas Muhammadiyah Surakarta ditunjukkan pada gambar 2.2.

The image shows a screenshot of the website for the Psychology Program at Muhammadiyah University of Surakarta (UMS). The page features a blue navigation bar with links for Beranda, Profil, Fasilitas, Berita, Publikasi Ilmiah, Kemahasiswaan, and Alumni. Below the navigation bar is a banner image of a group of people in a hall, with text "Selamat Datang" and names "Dr. Jagan Harjoto" and "Dr. Robert Oloan Rajagukguk". To the right of the banner is a green sidebar titled "UNIT KERJA" with links for Laboratorium, BKPP, and CIIP. Below the banner is a "Selamat Datang" section with a welcome message and a "Tanya Penerimaan Mahasiswa Baru" button. The page also includes a logo for "PSIKOLOGI Universitas Muhammadiyah Surakarta" and a "SEMINAR NASIONAL" logo.

Gambar 2.2 *Website* Program Studi Psikologi UMS

2.1.3 Website Program Studi Psikologi Universitas Sebelas Maret (UNS)

Website program studi Psikologi Universitas Sebelas Maret (UNS) dapat diakses di url <http://www.psikologi.uns.ac.id>. Website psikologi Universitas Sebelas Maret memiliki beberapa menu utama antara lain : menu beranda, menu profil, menu layanan, menu syarat penerimaan, menu kontak, menu alumni, menu perpustakaan, menu *download*. Berikut tampilan *website* Program Studi Psikologi Universitas Sebelas Maret ditunjukkan pada gambar 2.3.



Gambar 2.3 Website Program Studi Psikologi UNS

Kesimpulan dari ketiga *website* pada tinjauan pustaka di atas, peneliti melihat ada beberapa kekurangan pada *website* tersebut, yaitu dari sisi tampilan warna *website* UMS dan UNS kurang sesuai dengan Psikologi dan informasi data alumni pada kedua *website* UMS dan UNS tidak ditampilkan secara detail.

Pengamatan peneliti dari kedua referensi *website* tersebut dan referensi dari *website* psikologi *zone* yang desain tampilannya bagus, peneliti akan membangun *website* program studi psikologi Universitas Sahid Surakarta dengan

beberapa perbedaan antara lain data alumni ditampilkan dihalaman pengunjung yang disertai foto, nama, tempat tanggal lahir, tahun masuk, tahun lulus, *email*, judul skripsi, tempat kerja dan dari sisi warna tampilan *website* program studi Psikologi Universitas Sahid Surakarta menggunakan warna ungu karena sesuai dengan dasar karakter psikologi.

Penyusunan Tugas Akhir ini memerlukan sebuah landasan teori yang memiliki relevansi dengan masalah yang akan dibahas. Landasan teori ini diperoleh dari beberapa literatur yang mempublikasikan pendapat beberapa ilmuwan yang dipakai sebagai penunjang pembahasan masalah.

2.2 Pengertian Analisis dan Perancangan Sistem

Menurut Hanif Al Fatta (2007) Analisis sistem didefinisikan bagaimana memahami dan menspesifikasikan dengan detail apa yang harus dilakukan oleh sistem. Sementara sistem desain diartikan sebagai menjelaskan dengan detail bagaimana bagaian-bagaian dari sistem informasi diimplementasikan, sehingga analisis dan desain sistem informasi (ANSI) bisa didefinisikan sebagai proses organisasional kompleks dimana sistem informasi berbasis komputer diimplementasikan, bisa diringkas *Analysis* mendefinisikan masalah (*From requirements to specification*), *Design* memecahkan masalah (*From specification to implementation*).

2.2.1 Tahapan Analisis

Tahapan analisis adalah tahapan di mana sistem yang sedang berjalan di pelajari dan sistem pengganti diusulkan. Pada tahapan ini dideskripsikan sistem yang sedang berjalan, masalah, dan kesempatan didefinisikan, dan rekomendasi umum untuk bagaimana memperbaiki, meningkatkan atau mengganti sistem yang sedang berjalan diusulkan. Tujuan utama dari *fase* analisis adalah untuk memahami dan mendokumentasikan kebutuhan bisnis dan persyaratan proses dari sistem baru. Ada lima aktivitas utama dalam *fase* ini:

1. Pengumpulan Informasi

Langkah awal pada tahapan analisis adalah mengumpulkan informasi tentang bagaimana proses-proses bisnis yang ada pada sistem lama berjalan. Kemudian ditentukan pada titik-titik mana saja proses bisnis yang mengalami masalah yang bisa diselesaikan dengan sistem informasi. Kelemahan-kelemahan dari sistem lama diidentifikasi dan diperbaiki dengan sistem baru.

2. Mengidentifikasi sistem *requirement*

Informasi kelemahan sistem yang didapat, analisis sistem kemudian mendefinisikan apa saja sebenarnya yang dibutuhkan oleh sistem lama untuk mengatasi masalahnya. Inilah yang disebut sebagai *system requirement* (kebutuhan sistem). Seringkali kebutuhan ini akan mengubah total keseluruhan proses bisnis pada sistem lama, tetapi kadang-kadang hanya perubahan penambahan beberapa prosedur baru.

3. Memprioritaskan kebutuhan

Pada beberapa kasus kebutuhan yang diperoleh sangat lengkap dan rumit. Ketersediaan waktu dan sumber daya lain untuk menyelesaikan keseluruhan *requirement* bisa saja tidak mencukupi. Pada kondisi seperti ini maka analisis akan memprioritaskan kebutuhan - kebutuhan yang dianggap kritis untuk diprioritaskan.

4. Menyusun dan mengevaluasi alternatif

Satu hal yang tidak boleh dilupakan analisis adalah rencana kedua. Setelah menyusun dan memprioritaskan kebutuhan, analisis harus menyiapkan alternatif jika seandainya susunan kebutuhan nantinya akan ditolak oleh klien.

5. Mengulas kebutuhan dengan pihak manajemen.

Langkah terakhir adalah mengulas kebutuhan yang sudah ada dengan pihak klien, karena pihak klien yang paling tahu kebutuhan sistem mereka.

2.2.2 Tahapan Desain

Tahapan desain adalah tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang riil. Tahapan desain sistem dapat dibagi

menjadi 2 tahap, yaitu desain logis (*logical design*) dan tahapan desain fisik (*physical design*). Adapun perbedaan dari keduanya dapat dijelaskan sebagai berikut:

1. Desain Logis

Desain logis adalah bagian dari fase desain dalam *Systems Development Life Cycle* (SDLC) di mana semua fitur-fitur fungsional dari sistem dipilih dari tahapan analisis dideskripsikan terpisah dari *platform* komputer yang nanti digunakan.

Hasil dari tahapan ini adalah:

- a. Deskripsi fungsional mengenai data dan proses yang ada dalam sistem baru.
- b. Deskripsi yang detail dari spesifikasi sistem, meliputi: *Input* (data apa saja yang menjadi *input*), *Output* (informasi apa saja yang menjadi *output*), *Process* (prosedur apa saja yang harus dieksekusi untuk mengubah *input* menjadi *output*).

Tahapan desain logis biasanya menghasilkan beberapa dokumen, di antaranya dokumen model data, dokumen model proses, rancangan tabel, hierarki antar modul, sampai desain antar muka dari sistem yang akan dibuat.

2. Desain Fisik

Pada bagian ini, spesifikasi logis diubah kedalam detail teknologi dimana pemrograman dan pengembangan sistem bisa diselesaikan. Pada tahapan inilah aktifitas *coding* dilakukan adapun *output* dari sistem ini adalah:

- a. Deskripsi teknis, mengenai pilihan teknologi perangkat lunak dan perangkat keras yang digunakan.
- b. Deskripsi yang detail dari spesifikasi sistem meliputi: modul-modul program, *file-file*, sistem jaringan, sistem perangkat lunak

Pada tahapan desain, ada beberapa aktivitas utama yang dilakukan, yaitu:

- a. Merancang dan mengintegrasikan jaringan.
- b. Merancang arsitektur.
- c. Mendesain antar muka pengguna.
- d. Mendesain sistem antar muka.

- e. Mendesain dan mengintegrasikan *database*.
- f. Membuat *prototype* untuk detail dari desain.
- g. Mendesain dan mengintegrasikan kendali sistem.

2.2.3 Implementasi

Pada tahapan ini terdapat beberapa hal yang perlu dilakukan, yaitu:

1. Testing, yaitu menguji hasil kode program yang telah dihasilkan dari tahapan desain fisik. Tujuan pengujian ada dua. Dari sisi pengembang sistem, harus dijamin kode program yang dibuat bebas dari kesalahan sintaks maupun logika. Dari sisi pengguna, program yang dihasilkan harus mampu menyelesaikan masalah yang ada pada klien dan sistem baru harus mudah dijalankan dan dipahami oleh pengguna akhir.
2. Instalasi. Setelah program lulus uji coba, maka perangkat lunak dan perangkat keras akan diinstal pada organisasi atau perusahaan klien dan secara resmi mulai digunakan untuk mengganti sistem lama.

Output dari tahapan ini adalah *source code free*, prosedur pelatihan, dan buku panduan.

2.2.4 Pemeliharaan

Dimana pada tahapan ini sistem secara sistematis diperbaiki dan ditingkatkan. Hasil dari tahapan ini adalah versi baru dari perangkat lunak yang telah dibuat. Perbaikan yang dilakukan tingkatannya bisa sangat variatif, mulai dari memperbaiki program yang *crash* hingga berfungsi kembali sampai pada penambahan modul-modul program yang baru sebagai jawaban atas perubahan kebutuhan pengguna.

2.3 Pengertian *Internet*

Internet merupakan contoh jaringan terbesar yang menghubungkan jutaan komputer yang terbesar di seluruh penjuru dunia dan tak terikat pada satu organisasi, siapa saja dapat bergabung pada *internet*. Penggunaan jaringan *internet* ini, sebuah organisasi dapat melakukan pertukaran informasi secara

internal ataupun melakukan pertukaran informasi secara *eksternal* dengan organisasi-organisasi yang lain. Hal ini jaringan tersusun atas berbagai jenis komputer dan sistem operasi.

Pada awalnya *internet* dibangun oleh departemen pertahanan Amerika Serikat dalam rangka untuk melakukan hubungan dengan para ilmuwan dan profesor Universitas di seluruh dunia. Sehingga *internet* dapat digunakan oleh siapa saja untuk melakukan akses informasi apa saja dan bahkan untuk melakukan transaksi bisnis.

Supaya bisa berhubungan dengan *internet*, seorang pemakai dapat mengakses komputer pada perusahaan yang telah terkoneksi ke *internet* atau perlu menjadi pelanggan dari sebuah ISP (*Internet Service Provider*). Alternatif lain, cukup melakukan pengaksesan pada *internet* (*warnet*). ISP adalah organisasi komersial yang bergerak dalam bidang penyediaan jasa akses ke *internet*.

Koneksi jutaan komputer di *internet* ditangani dengan menggunakan protokol TCP/IP (*Transmission Control Protocol/Internet Protocol*) (Abdul Kadir, 2003).

2.4 Website

2.4.1 Pengertian Website

Website (situs *web*) adalah merupakan alamat (URL) yang berfungsi sebagai tempat penyimpanan data dan informasi dengan berdasarkan topik tertentu. *Web* menyediakan jaringan dimana-mana dan murah. *Website* mempunyai pengguna berdasarkan *software web browser* yang distandarisasi yang berjalan pada berbagai komputer biasa (Jhanner Shimarta, 2006).

Web Page (halaman *web*) merupakan halaman khusus dari situs *web* tertentu yang tersimpan dalam bentuk *file*. Pada *web page* tersimpan berbagai informasi dan *link* yang menghubungkan suatu informasi ke informasi lain baik itu dalam *page* yang sama ataupun *web page* lain pada *website* yang berbeda.

Website atau situs dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, *video* dan atau gabungan dari semuanya, baik yang bersifat statis maupun

dinamis yang membentuk satu rangkaian bangunan yang paling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website*. Contoh *website* statis adalah berisi profil perusahaan, sedangkan *website* dinamis adalah seperti *Facebook*, *Multiply*, dan lain-lain. Pada sisi pengembangnya, *website* statis hanya bisa diupdate oleh pemiliknya saja, sedangkan *website* dinamis bisa di update oleh pengguna maupun pemilik.

2.5 Pengertian Web Server

Web server dibutuhkan untuk melakukan pemrograman yang membutuhkan teknologi *server-side*, seperti PHP, *ColdFusion*, ASP dan JSP. Oleh karena itu, *web server* merupakan perangkat yang penting untuk melakukan uji coba pemrograman *server-side* yang anda buat. Sehingga tidak perlu melakukan *upload* data ke sebuah hosting hanya untuk mengetahui hasil pemrograman untuk sisi *server*. Pemrograman *server-side* biasanya digunakan untuk mengakses *database* yang terdapat pada *server hosting*, sehingga pengembang dengan mudah mengelola *content website* dengan mudah tanpa mengubah halaman-halaman *web* (Wahana Komputer, 2010).

Web Server atau HTTP/HTTPD *Server (Hyper Text Transfer Protocol/Daemon)* adalah sebuah sistem yang melayani permintaan (*Request*) HTTP *Client (web browser)* pada komputer *server* sehingga perlu sebuah layanan yang dapat menjalankan kedua sistem.

Web server adalah *server* yang melayani permintaan *client* terhadap halaman *web*. *Middleware* adalah perangkat lunak yang bekerja sama dengan *web server* dan berfungsi menterjemahkan kode-kode tertentu, menjalankan kode-kode tersebut dan memungkinkan berinteraksi dengan basis data. *Web Browser* adalah perangkat lunak di sisi *client* yang digunakan untuk mengakses informasi *web*. Contoh *Internet Explore*, *Mozilla Firefox* dll.

2.6 Pengertian *Web Browser*

Web Browser merupakan program yang berfungsi untuk menampilkan dokumen-dokumen *web* dalam format HTML. Bagaimana halaman *web* yang dibuat ditampilkan sangat tergantung pada *web engine* yang digunakan oleh masing-masing *browser*. Semua jenis *web browser* yang ada saat ini mengikuti standarisasi yang dibuat oleh W3C yang merupakan badan independen yang mengurus semua hal yang berkaitan dengan *web* di dunia.

Web Browser adalah aplikasi yang digunakan untuk menampilkan *file-file website*, baik html, php, mht dan lain sebagainya. Anda dapat menggunakan *browser Internet Explorer (IE)* yang merupakan bawaan *Windows*, *Mozilla FireFox*, *Opera*, *Safari*, *Chrome* (Wahana Komputer, 2010).

Web browser adalah perangkat lunak yang berguna untuk mengakses informasi *web* ataupun untuk melakukan transaksi via *web*. *Microsoft Internet Explorer* dan *Netcape Navigator* merupakan contoh *web browser* yang terkenal di lingkungan *Windows* (Abdul Kadir, 2003).

2.7 Pengertian HTML

Menurut Bernard Renaldy Suteja dkk (2007) untuk membangun sebuah *web page* dibutuhkan sebuah bahasa pemrograman yang lebih dikenal dengan sebutan *web scripting*. Dikatakan *script* karena perintah kode program tersebut akan di *interpreter* dan tidak ada kompilasi untuk menjadikannya *executable*. Berdasarkan letak proses *interpreter* maka *web scripting* dibagi menjadi dua kategori, yaitu yang bersifat *client side* dan *server side*. *Client side* dilakukan oleh *web browser* seperti *Internet Explorer*, *Netscape*, *Opera*, dan *Firefox*. Untuk contoh bahasa *client side* adalah html, CSS, *Javascript*, *VBscript*, dan XML. Sedangkan *server side* dilakukan oleh *web server* seperti PWS (*Personal Web Server* untuk sistem operasi *windows 98*), IIS (untuk sistem operasi *windows 2000/windows XP*), *side* adalah ASP (.Net), PHP, JSP, CFM, dan CGI/PL.

Web scripting yang bersifat *client side* akan menghasilkan *web page* yang statis artinya lebih menekankan pada desain format tampilan informasi dan informasi yang disajikan tidak dapat di *update* seketika, karena tidak dapat

dilakukan *request* interaktif dari pengguna dan proses tidak terjadi di *server* sehingga tidak akan menghasilkan *output* apa pun. Sehingga sangatlah tidak mungkin untuk menciptakan aplikasi *web* yang bersifat dinamis dengan *web scripting* bersifat *client side* ini, tetapi harus dikombinasikan juga dengan *web scripting* yang bersifat *server side*.

HTML (*Hypertext Markup Language*) adalah bahasa dasar untuk *web scripting* bersifat *client side* yang memungkinkan untuk menampilkan informasi dalam bentuk teks, grafik, serta multimedia dan juga untuk menghubungkan antar tampilan *web page* (*hyperlink*).

Tidak diperlukan suatu program *editor* khusus untuk menggunakan kode perintah-perintah html. Anda dapat mempergunakan *Notepad*, *Edit Plus*, ataupun *editor* lainnya yang berbasis GUI (*Graphical User Interface*) seperti *Microsoft Frontpage*, *Macromedia Dreamweaver*, *Adobe GoLive* dan sebagainya (dengan program-program ini anda tidak perlu mengetik kode htmlnya, semua perintahnya diwujudkan secara *icon base*). Tetapi bagi seorang pengembang aplikasi *web* maka kemampuan penguasaan terhadap kode-kode html sangat diperlukan, sehingga sangatlah disarankan untuk menguasai kode perintah html pergunakanlah *editor* teks (misalnya *Notepad*).

Aturan penulisan HTML ini sebagai berikut:

1. Memiliki struktur minimal dari dokumen HTML adalah:

```
<html>
  <head>
    <title> Judul Web Page </title>
  </head>
  <body>
    Isi dokumen ada disini.
  </body>
</html>
```

2. Bentuk penulisan seperti diatas disebut tag. Tag memiliki pembukaan <tag>, tetapi tidak semua tag punya penutup </tag>, tag memberi instruksi *interpreter* kepada *browser* (misalnya *Internet Explorer*). Contoh pada

dokumen di atas, tag yang pertama memberitahukan kepada *browser* bahwa dokumen tersebut adalah dokumen html dengan tag `<html></html>` di awal dan di akhir dokumen. Dalam dokumen html penggunaan tag-tag penutup html disesuaikan dengan urutan penggunaan tag tersebut.

3. Pada sebuah tag html terdapat atribut tag. Atribut yang terkandung dalam tag satu dengan yang lainnya belum tentu sama. Atribut yang dipakai tidak memiliki urutan pendefinisian tertentu. Atribut tag digunakan untuk memodifikasi sifat-sifat dari tag, bergantung dari nilai yang diberikan. Namun, juga ada atribut tag yang tidak memiliki nilai.
4. Penulisan tag bersifat *non case-sensitive*, yang artinya bahwa bahwa penulisan tag huruf besar dan atau huruf kecil akan dianggap sama. Sebagai contoh `<HTML>` akan sama dengan `<html>` atau `<Html>`.
5. Setelah tanda “<” harus langsung diikuti oleh nama tag dan ditutup tanda “>” tidak boleh ada spasi, angka, tanda baca, dan nama tag tidak dapat dibuat sendiri. Nama tag harus mengikuti standar W3C, yaitu sebuah lembaga atau organisasi internasional yang mengatur standarisasi *web* (<http://www.w3c.org>).
6. Dokumen html diperlukan tag `<head> </head>`, dalam tag ini diletakan judul *web page* `</title>`, sedangkan di dalam tag `<body> </body>` berisikan isi dari *web page* anda. Untuk ekstensi dari *file* html perlu ditambahkan `.html` atau `.htm` di belakang nama *file*-nya (misalnya: `coba.htm`).
7. Penulisan tag-tag html tidak akan terpengaruh oleh perintah langsung *keyboard* seperti penekanan tombol *enter* atau spasi, *countinue horizontal* seperti ini:

`<html><head><title> Judul Web Page </title></head>` dan seterusnya.

Menurut Madcoms (2008) pada dasarnya html adalah semacam standar yang digunakan dalam dunia *web*. html bersifat fleksibel, artinya html dapat disisipi kode-kode *script* seperti *JavaScript*, *VBScript*, dan juga bahasa pemrograman berbasis *web* seperti PHP dan ASP. Animasi seperti *flash* serta beberapa jenis *file* grafis dan multimedia juga dapat disisipkan dan ditampilkan

melalui html. Fleksibilitas html dan daya dukungnya ini meningkatkan daya guna dan menjadikan sebagai bahasa kode yang ‘dinamis’.

HTML (*HyperText Markup Language*) dikenal sebagai bahasa kode berbasis teks untuk membuat sebuah halaman *web*. Keberadaannya dikenali dengan adanya ekstensi *.htm atau *.html, misalnya index.html. Sementara *web* sendiri sebenarnya berasal dari kata *World Wide Web*, salah satu layanan *global* yang diperuntukkan bagi semua pengguna *internet*.

World Wide Web dikembangkan oleh W3C (*World Wide Web Consortium*) yang merupakan badan resmi dalam dalam membuat standar *web*. W3C (<http://www.w3.org>) yang didirikan pada tahun 1994 ini juga menjadi standar acuan dalam aturan spesifikasi, *guide line*, *software*, dan *tool* yang dibuat di seluruh dunia.

Membuat *file* html dapat mengguankan *software editor* seperti *Dreamweaver*. Sebenarnya, *notepad* atau *wordpad* sudah cukup apabila ingin membuat sebuah *file* html sederhana atau ingin mendalami bahasa kode html. Salah satu hal terpenting untuk mempelajari untuk mempelajari html adalah mengenal dan menguasai penggunaan tag-tag html dan dukungan *browser* terhadap tag-tag tersebut.

Menurut Bunafit Nugroho (2004) HTML adalah kependekan dari (*HyperText Markup Language*), merupakan sebuah bahasa *scripting* yang berguna untuk menuliskan halaman *web*. Pada halaman *web*, html dijadikan sebagai bahasa *script* dasar yang berjalan bersama berbagai bahasa *scripting* pemrograman lainnya.

Semua tag-tag HTML bersifat dinamis, artinya kode html tidak dapat dijadikan sebagai *file executable* program. Hal ini disebabkan html hanyalah sebuah bahasa *scripting* yang dapat berjalan apabila dijalankan di dalam *browser* (pengaksesan *web*), *browser-browser* yang mendukung html antara lain adalah *Internet Explorer*, *Netscape Navigator*, *Opera*, *Mozilla* dan lain-lain. Jadi pada saat ingin membuka halaman yang berasal dari html anda dapat melihat bentuk pengkodeanya dengan cara mengklik menu *View source*, maka disana akan ditampilkan semua tag beserta isi dari halaman *web* tersebut.

Semua bahasa *scripting* yang berjalan di bawah *web* dapat didukung oleh html, biasanya bahasa-bahasa tersebut melakukan *embedded script* pada tag-tag html.

Karena html hanyalah merupakan sebuah kode *scripting* dan bukan merupakan program *compiler* maka anda dapat menuliskan semua kode-kode program dengan menggunakan *editor* yang anda sukai, adapun *editor* yang anda digunakan adalah *macromedia dreamweaver*, *front page*, *home site* atau anda juga dapat menggunakan *notepad* sebagai *editor* standar bawaan *windows*.

2.8 Pengertian PHP

Pengertian PHP menurut Abdul Kadir (2008) menurut dokumen resmi PHP, PHP merupakan singkatan dari PHP *Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*. Hasilnya yang dikirimkan ke klien, tempat pemakai menggunakan *browser*.

Secara khusus, PHP dirancang untuk membentuk aplikasi *web* dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, Anda bisa menampilkan isi *database* ke halaman *web*. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *cold fusion*, ataupun *perl*. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *command line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip *Perl* yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi *tool* yang disebut "*Personal Home Page*". Paket inilah yang menjadi cikal-bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/F1 versi 2. Pada versi inilah pemrogram dapat menempelkan kode terstruktur di dalam tag HTML. Selanjutnya dalam kode PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks sambil jalan.

Pada saat ini PHP cukup populer sebagai peranti pemrograman *web*, terutama di lingkungan *linux*. Walaupun demikian, PHP sebenarnya juga dapat berfungsi pada *server-server* yang berbasis *unix*, *windows*, dan *macintosh*.

Pada awalnya, PHP dirancang untuk diintegrasikan dengan *web server apache*. Namun, belakangan PHP juga dapat bekerja dengan *web server* seperti PWS (*Personal Web Server*), ISS (*Internet Information Server*), dan *xitami*.

Menurut Ahmad Luthfie (2005) PHP atau “*PHP Hypertext Preprocessor*” tidak termasuk bahasa pemrograman, tetapi sebuah bahasa *scripting* yang menyatu dengan kode-kode (TAG) HTML, menggunakan dasar bahasa C, *Java* atau *Perl*, lalu dijalankan (dieksekusi) oleh *server* agar menghasilkan sebuah *website* dinamis.

Pada prinsipnya, PHP berbeda dengan bahasa-bahasa yang lain seperti CGI/PERL, atau bahasa *scripting* lainnya, seperti *Java Script*. Kita ketahui bahwa CGI/PERL berisi banyak perintah untuk mengerate sebuah *file* HTML. Pada *Java Script*, perintah dieksekusi dan bekerja pada komputer *client*. Sebaliknya, PHP dieksekusi dan bekerja pada posisi komputer *server*, kemudian mengirimkan hasil eksekusi ke komputer *client* melalui jalur HTTP.

Berikut adalah beberapa alasan mengapa PHP menjadi sangat terkenal dan digunakan oleh sebagian besar *web developer*:

1. PHP, seiring dengan kemajuan dan kebutuhan, berkembang cepatnya sehingga selalu berusaha menyempurnakan atau melengkapi dirinya agar dapat digunakan maksimal.
2. PHP dapat menerima berbagai jenis sistem operasi seperti *windows*, *linux*, *Unix*, *FreeBSD*, dan lain-lain sehingga memungkinkan PHP digunakan lintas *platform* sistem operasi.
3. PHP dapat dijalankan pada berbagai jenis *web server* seperti *apache*, *fhhttpd*, *phhttpd*, *AOLServer*, *ISS*, *PWS*, dan lain-lain.
4. PHP dapat berjalan dan mendukung paket-paket *database*, baik komersial maupun nonkomersial, seperti *MySQL*, *mSql*, *PostgreSQL*, *Oracle*, *Informix*, *Microsoft SQL Server*, dan lain-lain.

Menurut Bunafit Nugroho (2004) PHP (PHP: *Hypertext Preprocessor*) adalah sebuah bahasa pemrograman yang berbentuk *Scripting*, sistem kerja dari program ini adalah sebagai *interpreter* bukan sebagai *compiler*.

Seperti pada pemrograman-pemrograman lainnya PHP memiliki beberapa aturan penulisan yang harus anda ketahui sebelumnya, yaitu bagaimana memulai program PHP dan mengakhiri PHP, sehingga dengan memenuhi beberapa aturan yang ada anda dapat mengerjakan program yang dihadapi. Untuk memulai program PHP, anda dapat memulainya dengan mengenal sebuah tag pengenalan PHP yang digunakan untuk menuliskan kode PHP.

Penulisan kode PHP harus memulainya dengan tanda `<?php`, setelah tanda tersebut anda dapat melanjutkan dengan kode program isi didalamnya. Untuk mengakhiri kode program yang dibuat, anda dapat menutupnya dengan tanda `?>`. Selain penggunaan tanda seperti itu, Anda juga dapat menggunakan beberapa bentuk lain seperti yang ditunjukkan pada tabel 2.1 berikut:

Tabel 2.1 Bentuk Pembuka Program PHP

Awal	Akhir
<code><?</code>	<code>?></code>
<code><?php</code>	<code>?></code>
<code><script language="php"></code>	<code></script></code>
<code><%</code>	<code>%></code>

Bentuk sederhana dari program PHP ditunjukkan pada *script* yang melakukan pemanggilan salah satu fungsi php berikut:

```
<?php
    php();
?>
```

2.9 CSS (*Cascading Style Sheets*)

CSS (*Cascade Style Sheet*) adalah sebuah fitur yang diperkenalkan sejak HTML versi 4.0 dan berfungsi untuk menangani masalah tampilan pada HTML seperti jenis, ukuran dan warna *font*, posisi teks, batas tulisan atau *margin*, warna *background*, dan sebagainya.

Dari sisi manajemen dan perawatan, penggunaan CSS dipandang lebih praktis karena para *web developer* tidak perlu membuka setiap *file* dalam sebuah situs untuk melakukan perubahan. Sebagai contoh, katakanlah dalam sebuah situs digunakan *font* berjenis *arial*.

Hal penting yang perlu diperhatikan adalah cara meletakkan CSS dan juga bahasa berbasis *web* lain untuk memudahkan manajemen *file*, *editing*, dan *maintenance*. Banyak di antara *programmer web* yang belum menyadari aspek-aspek penting ini menyisipkan CSS, *JavaScript*, *VBScript*, PHP, maupun ASP langsung ke dalam dokumen HTML (*embedded script*) (Madcoms, 2008).

Menurut Budi Kurniawan (2008) CSS atau *Cascading Style Sheet* adalah sebuah dokumen yang berisi aturan yang digunakan untuk memisahkan isi dengan *layout* dalam halaman-halaman *web* yang dibuat. CSS memperkenalkan “*template*” yang berupa *style* untuk dibuat dan mengizinkan penulisan kode yang lebih mudah dari halaman-halaman *web* yang dirancang. CSS mampu menciptakan halaman yang tampak sama pada revolusi layar yang berbeda dari pengunjung berbeda tanpa memerlukan penggunaan tabel seperti html klasik.

CSS (*Cascading Style Sheet*) akan lebih mudah dalam melakukan *setting* tampilan keseluruhan *web* hanya dengan mengganti atribut-atribut atau perintah dalam *style* CSS dengan atribut yang diinginkan. Anda tidak perlu repot mengubah satu per satu atribut tiap elemen yang ada dalam halaman *web* jika anda menggunakan *style* CSS ini.

Penggunaan CSS dokumen yang dibuat menggunakan HTML murni akan dapat lebih disingkat dan akan lebih mengefisienkan waktu yang dibutuhkan. Bayangkan jika kita membuat sebuah situs yang kompleks dengan banyak atribut untuk setiap halaman dalam situs tersebut, maka kita harus banyak menentukan tiap atribut untuk tiap halaman tersebut satu per satu.

Saat ini CSS merupakan *style* yang banyak digunakan dikarenakan berbagai kemudahan dan lengkapnya atribut yang dimilikinya, disamping berbagai kemudahan lain yang ditawarkannya. Pengguna CSS dalam *web* akan lebih efisien dikarenakan CSS dapat digunakan secara berulang pada tag-tag tertentu,

sehingga kita tidak usah menyetikkan ulang seluruh perintah pemformatan seperti halnya dalam HTML.

2.10 Adobe Dreamweaver

Adobe Dreamweaver merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah *website*, baik secara grafis maupun dengan menulis kode sumber secara langsung.

Adobe Dreamweaver CS4 memudahkan pengembang *website* untuk mengelola halaman-halaman *website* dan aset-asetnya, baik gambar (*image*), animasi *flash*, *video*, suara dan lain sebagainya. Selain itu, *Adobe Dreamweaver CS4* juga menyediakan fasilitas untuk melakukan pemrograman *scripting*, baik ASP (*Active Server Page*), JSP (*Java Server Page*), PHP (*PHP Hypertext Preprocessor*), *JavaScript* (*js*), *Cold Fusion*, CSS (*Cascading Style Sheet*), XML (*Extensible Markup Language*) dan lainnya (Wahana Komputer, 2010).

2.11 Pengertian Basis Data

Basis data (*Database*) menurut Fathansyah (2007) terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis Data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

3. Kumpulan *file/tabel/arsip* yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Menurut Stephens dan Plew (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data harus mudah ditambah ke dalam basis data, dimodifikasi, dan dihapus (Janner Simarmata & Imam Paryudi, 2006).

Menurut Silberschatz et al (2002) mendefinisikan basis data sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan. Sistem manajemen basis data (DBMS) adalah kumpulan data yang saling berhubungan dan kumpulan program untuk mengakses data. Tujuan utama sistem manajemen basis data adalah menyediakan cara menyimpan dan mengambil informasi basis data secara mudah dan efisien (Janner Simarmata & Imam Paryudi, 2006).

2.12 MySQL

Menurut Abdul Kadir (2008) MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *databasenya*. Selain itu, ia bersifat *Open Source* (Anda tidak perlu membayar untuk menggunakannya) pada berbagai *platform* (kecuali untuk jenis *Enterprise*, yang bersifat komersial). Perangkat lunak MySQL sendiri bisa di-*download* dari <http://www.mysql.com>.

MySQL termasuk jenis 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 atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Istilah tabel, baris, dan kolom.

Menurut Bunafit Nugroho (2004) MySQL merupakan *database* yang paling digemari dikalangan *programmer web*, dengan alasan bahwa program ini

merupakan *database* yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data. Sebagai sebuah *database server* yang mampu untuk memajemen *database* dengan baik, MySQL terhitung merupakan *database* yang paling digemari dan paling banyak digunakan dibandingkan *database* lainnya. Selain MySQL masih terdapat beberapa jenis *database server* yang juga memiliki kemampuan yang juga tidak bisa dianggap enteng, *database* itu adalah *Oracle* dan *PostgreSQL*.

Di dalam dunia *internet*, MySQL dijadikan sebagai sebuah *database* yang paling banyak digunakan selain *database* yang bersifat *share ware* seperti *Ms Access*, penggunaan MySQL ini biasanya dipadukan dengan menggunakan program aplikasi PHP, karena dengan menggunakan kedua program tersebut di atas telah terbukti akan kehandalannya dalam menangani permintaan data.

Pada *distro database* ini, MySQL memiliki *query* yang telah distandarkan oleh ANSI/ISO yaitu menggunakan bahasa SQL sebagai bahasa permintaanya, hal tersebut juga telah dimiliki oleh bentuk-bentuk *database server* seperti *Oracle*, *PostgreSQL*, MySQL, SQL *server* maupun bentuk-bentuk *database* yang berjalan pada mode grafis (sifatnya visual) seperti *Interbase* yang diproduksi oleh *Borland*.

Kemampuan lain yang dimiliki MySQL adalah mampu mendukung Relasional *Database* Manajemen Sistem (RDBMS), sehingga dengan kemampuan ini MySQL akan mampu menangani data-data sebuah perusahaan yang berukuran sangat besar hingga berukuran *Giga Byte*.

Distro MySQL dapat mengunjungi *portal* pengembangnya pada alamat <http://www.mysql.com>, pada situs tersebut anda dapat memperoleh segala informasi yang berkaitan dengan versi terbaru dan *tool* yang berkenan dengan MySQL.

Hal ini yang perlu anda ketahui mengenai MySQL adalah bahwa MySQL merupakan sebuah *software database* yang bersifat *free* (gratis) karena MySQL dilisensi dibawah GNU *General Public Licence* (GPL). Dengan adanya keadaan seperti itu maka anda dapat menggunakan *software database* ini dengan bebas tanpa harus takut dengan lisensi yang ada.

Menurut Heni A. Puspitosari (2011) jika ingin membuat *website* yang interaktif dan dinamis, perlu adanya media penyimpanan data yang fliexibel dan mudah untuk diakses. Dalam bahasa pemrograman sering ada istilah *database*. *Database* adalah kumpulan data-data yang saling terkait, tersimpan, dan mudah untuk diakses.

Salah satu program yang dapat digunakan sebagai *database* adalah MySQL. MySQL merupakan salah satu *software* untuk *database server* yang banyak digunakan, MySQL bersifat *open souce* dan menggunakan SQL. MySQL bisa dijalankan di berbagai *platform* misalnya *Windows*, *Linux*, dan lain sebagainya, MySQL memiliki beberapa kelebihan, antara lain:

1. *Multiuser*. MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.
2. *Performance tuning*. MySQL memiliki kecepatan yang bagus dalam menangani *query* sederhana, yaitu dapat memproses lebih banyak SQL per satuan waktu.
3. Jenis kolom. MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer*, *float*, *double*, *char*, *text*, *date*, *timestamp*, dan masih banyak lagi.
4. Perintah dan fungsi. MySQL memiliki operator dan fungsi secara penuh dan mendukung perintah *Select* dan *Where* dalam perintah *query*.
5. Keamanan. MySQL memiliki keamanan yang bagus karena beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang lengkap serta sandi terenskripsi.
6. 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 milyar baris. Selain itu, batas *indeks* yang dapat ditampung mencapai 32 *indeks* pada tiap tabelnya.
7. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan *protocol TCP/IP*, *Unix socket (UNIX)*, atau *Named Pipes (NT)*.
8. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa.

9. *Antar Muka*. MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
10. *Portabilitas*. MySQL dapat berjalan stabil pada berbagai *system* operasi seperti *windows, linux, FreeBSD, Mac Os X Server, solaris, amiga*, dan masih banyak lagi.
11. *Open Souce*. MySQL didistribusikan secara *open souce*, di bawah lisensi GPL sehingga dapat digunakan secara gratis.
12. *Klien dan Peralatan*. MySQL dilengkapi dengan berbagai peralatan yang dapat digunakan untuk adminitrasi 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, misalnya *Oracle*.

2.13 Normalisasi

Menurut Abdul Kadir (2009) Normalisasi adalah suatu proses yang digunakan untuk menentukan pengelompokan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang berstruktur baik. Dalam hal ini yang dimaksud dengan relasi yang berstruktur baik adalah relasi yang memenuhi dua kondisi berikut:

1. Mengandung redundansi sedikit mungkin.
2. Memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi, dan dihapus tanpa menimbulkan kesalahan atau ketidakkonsistenan.

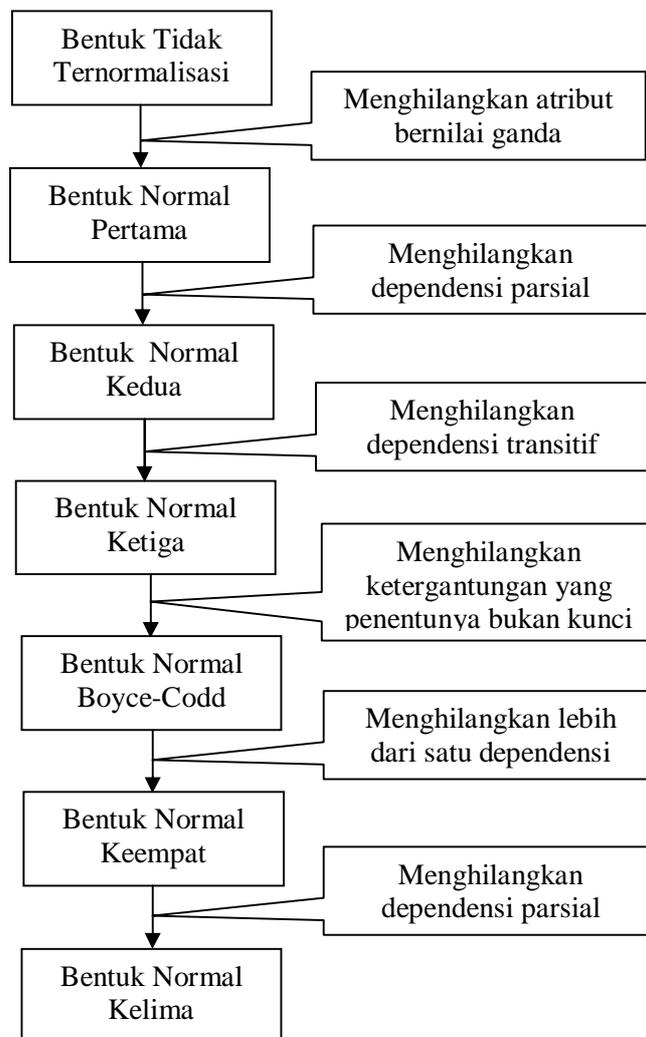
Redudansi data adalah data disimpan berkali-kali, redudansi adalah istilah lain untuk duplikasi. Normalisasi juga dapat dipakai untuk memverifikasi relasi-relasi hasil transformasi model E-R apakah sudah berstruktur baik atau belum.

Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. Hal ini yang disebut bentuk normal adalah “ suatu keadaan relasi yang dihasilkan oleh penerapan aturan-aturan sederhana yang berhubungan dengan dependensi fungsional terhadap relasi tersebut “ (Hoffer, dkk., 2005). Aturan-aturan tersebut dan juga

istilah dependensi fungsional akan dibahas belakangan. Sekedar perlu diketahui terlebih dahulu bahwa bentuk normal dalam normalisasi dapat berupa:

- a. Bentuk normal pertama (1 NF / *First normal form*)
- b. Bentuk normal kedua (2 NF / *Second normal form*)
- c. Bentuk normal ketiga (3 NF / *Thrid normal form*)
- d. Bentuk normal *Boyce-Codd* (BCNF / *Boyce-Codd normal form*)
- e. Bentuk normal keempat (4 NF / *Fourth normal form*)
- f. Bentuk normal kelima (5 NF / *Fifth normal form*)

Berikut masing-masing bentuk normal, adapun hubungan keenam bentuk normal tersebut dapat dilihat pada Gambar 2.4.



Gambar 2.4 Langkah-Langkah Dalam Normalisasi

2.14 Kamus data (KD)

Menurut Jogiyanto (2005) Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, Analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis, KD dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, KD digunakan untuk merancang *input*, merancang laporan-laporan dan *database*. KD dibuat berdasarkan arus data yang ada di DAD. Arus data di DAD sifatnya global, hanya ditunjukkan nama arus datanya saja.

2.14.1 Isi Kamus Data

KD harus dapat mencerminkan keterangan yang jelas tentang data yang dicatat. Untuk masud keperluan ini, maka KD harus membuat hal-hal berikut ini.

1. Nama arus data

Karena KD dibuat berdasarkan arus data yang mengalir di DAD , maka nama dari arus data juga harus dicatat di KD, sehingga mereka yang membaca DAD dan memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu di DAD dapat langsung mencarinya dengan mudah di KD.

2. Alias

Alias atau nama lain dari data dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya. Misalnya bagian pembuat faktur dan langganan menyebut bukti penjualan sebagai faktur, sedangkan bagian gudang menyebutnya sebagai tembusan permintaan persediaan. Baik faktur dan tembusan permintaan persediaan ini mempunyai struktur data yang sama, tetapi mempunyai struktur yang berbeda.

3. Bentuk data

Telah diketahui bahwa arus data dapat mengalir:

- a. Dari kesatuan luar ke suatu proses, data yang mengalir ini biasanya tercatat di suatu dokumen atau formulir.
- b. Hasil dari suatu proses ke satuan luar, data yang mengalir ini biasanya terdapat di media laporan atau *query* tampilan layar atau dokumen hasil cetakan komputer.
- c. Hasil suatu proses ke proses yang lain, data yang mengalir ini biasanya dalam bentuk variabel atau parameter yang dibutuhkan oleh proses penerimanya.
- d. Hasil suatu proses yang direkamkan ke simpanan data, data yang mengalir ini biasanya berbentuk suatu variabel.
- e. Dari simpanan data dibaca oleh suatu proses, data yang mengalir ini biasanya berupa suatu *field* (item data).

Dengan demikian bentuk dari data yang mengalir dapat berupa:

- a. Dokumen dasar atau formulir
- b. Dokumen hasil cetakan komputer
- c. Laporan tercetak
- d. Tampilan dilayar monitor
- e. Variabel
- f. Parameter
- g. *Field*

Bentuk dari data ini perlu dicatat di KD, karena dapat digunakan untuk mengelompokkan KD ke dalam kegunaannya sewaktu perancangan sistem. KD yang mencatat data yang mengalir dalam bentuk dokumen dasar atau formulir akan digunakan untuk merancang bentuk *input* sistem. KD yang mencatat data yang mengalir dalam bentuk laporan tercetak dan dokumen hasil cetakan komputer akan digunakan untuk merancang *output* yang akan dihasilkan oleh sistem. KD yang mencatat data yang mengalir dalam bentuk tampilan di layar monitor akan digunakan juga untuk merancang tampilan layar yang akan dihasilkan oleh sistem. KD yang

mencatat data yang mengalir dalam bentuk parameter dan variabel akan digunakan untuk merancang proses dari program. KD yang mencatat data yang mengalir dalam bentuk dokumen, formulir, laporan, dokumen cetakan komputer, tampilan di layar monitor, variabel dan *field* akan digunakan untuk merancang *database*.

4. Arus data

Arus data menunjukkan dari mana data mengalir dan kemana data akan menuju. Keterangan arus data ini perlu dicatat di KD supaya memudahkan mencari arus data ini di DAD.

5. Penjelasan

Untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat di KD, maka bagian penjelasan dapat di isi dengan keterangan-keterangan tentang arus data tersebut. Sebagai misalnya nama dari arus data adalah tembusan permintaan persediaan, maka dapat lebih dijelaskan sebagai tembusan dari faktur penjualan untuk meminta barang dari gudang.

6. Periode

Periode ini menunjukkan kapan terjadinya arus data ini. Periode perlu dicatat di KD karena dapat digunakan untuk mengidentifikasi kapan *input* data harus dimasukan ke sistem, kapan proses dari program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

7. *Volume*

Volume yang diperlukan dicatat di KD adalah tentang *volume* rata-rata dan *volume* puncak dari arus data. *Volume* rata-rata menunjukkan banyaknya rata-rata arus data yang mengalir dalam satu periode tertentu dan *volume* puncak menunjukkan *volume* yang terbanyak. *Volume* ini digunakan untuk mengidentifikasi besarnya simpanan luar yang akan digunakan, kapasitas dan jumlah dari alat *input*, alat pemroses dan alat *output*.

8. Struktur data

Struktur data menunjukkan arus data yang dicatat di KD terdiri dari item-item data apa saja.

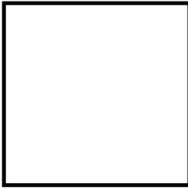
2.15 Diagram Berjenjang (*Hirarchy Chart*)

Bagan berjenjang digunakan untuk mempersiapkan penggambaran DAD ke level-level lebih bawah lagi. Bagan berjenjang dapat digambarkan dengan menggunakan notasi proses yang digunakan di DAD (Jogiyanto, 2005).

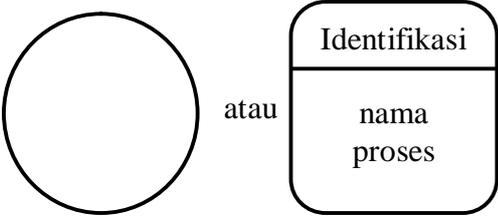
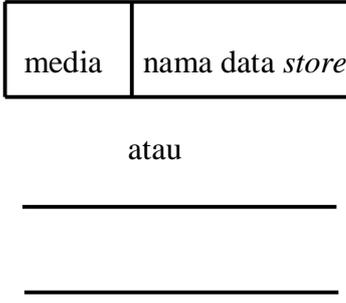
2.16 Pengertian *Data Flow Diagram* (DFD)

Menurut Jogiyanto (2005) 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 (misalnya lewat telpon, surat dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan (misalnya *file* kartu, *microfiche*, *hard disk*, *tape*, *disket* dan lain sebagainya). DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured Analysis and design*). DFD merupakan alat yang cukup populer sekarang ini, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas. DFD juga merupakan dokumentasi dari sistem yang baik. Simbol digunakan di DFD untuk masud mewakili : *external entity* (kesatuan luar) atau *boundary* (batas sistem), *data flow* (arus data), *process* (proses), *data store* (simpanan data).

Tabel 2.2. Simbol-Simbol *Data Flow Diagram* (DFD).

Notasi	Keterangan
	<p>Notasi <i>external entity</i> (kesatuan luar) merupakan kesatuan (<i>entity</i>) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainya yang berada di lingkungan luarnya yang akan memberikan <i>input</i> atau menerima <i>output</i> dari sistem. Kesatuan luar (<i>external entity</i>) dapat diberi identifikasi dengan huruf kecil di ujung kiri atas.</p>

Lanjutan Tabel 2.2. Simbol-Simbol *Data Flow Diagram* (DFD)

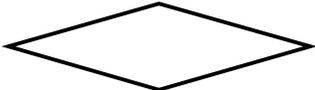
Notasi	Keterangan
	<p>Simbol arus data (<i>data flow</i>) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (<i>process</i>), simpanan data (<i>data store</i>) dan kesatuan luar (<i>external entity</i>). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.</p>
	<p>Notasi proses di DFD</p> <p>Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.</p>
	<p>Simbol dari simpanan data di DFD</p> <p>Simpanan data (<i>data store</i>) merupakan simpanan dari data yang dapat berupa: suatu <i>file</i> atau <i>database</i> di sistem komputer, suatu arsip atau catatan manual, suatu kotak tempat data di meja seseorang, suatu tabel acuan manual, suatu agenda atau buku.</p> <p>Simpanan data di DFD dapat disimbolkan dengan sepasang garis horisontal paralel yang tertutup di salah satu ujungnya.</p>

(sumber : Jogiyanto, 2005)

2.17 Pengertian ERD

Entitas Relationship Diagram adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas (Janner Simarmata & Imam Paryudi, 2006). Komponen-komponen dalam ERD dapat dilihat pada tabel 2.3 berikut ini.

Tabel 2.3. Simbol-Simbol ERD

Notasi	Keterangan
	Himpunan Entitas merupakan sekumpulan objek dasar atau individu sejenis yang berada dalam lingkup yang sama.
	Atribut merupakan sifat-sifat atau properti yang dimiliki oleh suatu entitas.
	Himpunan relasi menggambarkan bagaimana entitas saling berhubungan.
	Garis menghubungkan <i>atribut</i> dengan kumpulan entitas dan hubungn entitas dengan relasi.

(Sumber : Janner Simarmata & Imam Paryudi, 2006)

2.18 Bagan Alir (*Flowchart*)

Menurut Jogiyanto (2005) bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Pada waktu akan menggambar suatu bagan alir, analisis sistem atau pemrogram dapat mengikuti pedoman-pedoman sebagai berikut ini.

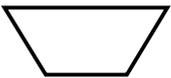
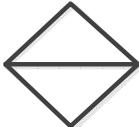
1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhirnya.

4. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan, misalnya “persiapan” dokumen, “Hitung” gaji.
5. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakanlah simbol-simbol bagan alir yang standar.

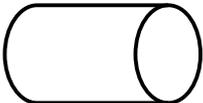
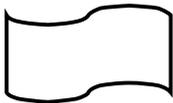
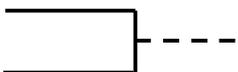
2.18.1 Bagan Alir Sistem (*Systems flowchart*)

Bagan alir sistem (*systems flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di sistem. Bagan alir sistem digambar dengan menggunakan simbol-simbol yang tampak sebagai berikut ini.

Tabel 2.4 Simbol - Simbol *Flowchart* Sistem

Notasi	Keterangan
	Simbol dokumen Menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer.
	Simbol kegiatan manual Menunjukkan pekerjaan manual.
	Simbol kartu plong Menunjukkan <i>input</i> / <i>output</i> yang menggunakan kartu plong (<i>punched card</i>).
	Simbol proses Menunjukkan kegiatan proses dari operasi program komputer.
	Simbol operasi luar Menunjukkan operasi yang dilakukan di luar proses operasi komputer.
	Simbol pengurutan <i>offline</i> Menunjukkan proses pengurutan data di luar proses komputer

Lanjutan Tabel 2.4 Simbol - Simbol *Flowchart* Sistem

Notasi	Keterangan
	Simbol pita magnetik Menunjukkan <i>input</i> / <i>output</i> menggunakan pita magnetik
	Simbol <i>Hard disk</i> Menunjukkan <i>input</i> / <i>output</i> menggunakan <i>hard disk</i>
	Simbol <i>diskette</i> Menunjukkan <i>input</i> / <i>output</i> menggunakan <i>diskette</i> .
	Simbol drum magnetik Menunjukkan <i>input</i> / <i>output</i> menggunakan drum magnetik
	Simbol pita kertas berlubang Menunjukkan <i>input</i> / <i>output</i> menggunakan pita kertas berlubang.
	Simbol <i>keyboard</i> Menunjukkan <i>input</i> yang menggunakan <i>on-line keyboard</i>
	Simbol <i>display</i> Menunjukkan <i>output</i> yang ditampilkan di monitor.
	Simbol penghubung Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain.
	Simbol garis alir Menunjukkan arus dari proses.
	Simbol penjelasan Menunjukkan penjelasan dari suatu proses.

(Sumber : Jogiyanto, 2005)

2.18.2 Bagan Alir Program (*Flowchart Program*)

Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari verifikasi bagan alir sistem. Bagan alir program dibuat dengan menggunakan simbol-simbol sebagai berikut ini.

Tabel 2.5 Simbol - Simbol *Flowchart* program

Notasi	Keterangan
	Simbol <i>Input / Output</i> Digunakan untuk mewakili data <i>input / output</i>
	Simbol Proses Digunakan untuk mewakili suatu proses.
	Simbol penghubung (<i>connector symbol</i>) Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama atau di halaman lainnya.
	Simbol garis alir (<i>flow lines symbol</i>) Digunakan untuk menunjukkan arus dari proses.
	Simbol Keputusan (<i>decision symbol</i>). Digunakan untuk suatu penyelesaian kondisi di dalam program.
	Simbol proses terdefinisi (<i>predefined process symbol</i>) Digunakan untuk menunjukkan suatu operasi yang rinciannya di tunjukkan di tempat lain.
	Simbol Persiapan (<i>preparation symbol</i>) Digunakan untuk memberi nilai awal suatu besaran
	Simbol titik terminal (<i>terminal point symbol</i>) digunakan untuk menunjukkan awal dan akhir dari suatu proses.

(Sumber : Jogiyanto, 2005)

2.19 Metode *Webqual*

Jurnal (Imam Sanjaya, 2012) *WebQual* merupakan salah satu metode atau teknik pengukuran kualitas *website* berdasarkan persepsi pengguna akhir. Metode ini merupakan pengembangan dari *SERVQUAL* yang banyak digunakan sebelumnya pada pengukuran kualitas jasa. *WebQual* sudah mulai dikembangkan sejak tahun 1998 dan telah mengalami beberapa interaksi dalam penyusunan dimensi dan butir pertanyaannya.

Versi pertama dari instrume *webqual* (*webqual* 1.0) dikembangkan sebagai bagian dari hasil lokakarya yang diselenggarakan dengan melibatkan para siswa yang diminta untuk mempertimbangkan kualitas *website* sekolah. Instrumen

webqual disaring melalui proses perbaikan secara iteratif dengan menggunakan kuesioner percobaan sebelum disebarkan untuk populasi yang lebih besar. Dua puluh empat pertanyaan di dalam instrumen *webqual* diuji dengan aplikasi dalam ruang lingkup *website* sekolah bisnis di Inggris. Analisis dari data yang dikumpulkan mendorong penghapusan atas satu item pertanyaan. Berdasarkan analisis reliabilitas, tersisa 23 pertanyaan yang kemudian dikelompokkan menjadi empat dimensi utama, yaitu kemudahan penggunaan, pengalaman, informasi, komunikasi dan integrasi (Barnes dan Vidgen, 2001).

Kualitas yang diidentifikasi dalam *webqual* 1.0 membentuk titik awal untuk menilai kualitas informasi dari suatu *website* di *WebQual* 2.0. Namun demikian, dalam penerapan *webqual*, pada *website* berjenis B2C (*Business to Consumer*) terlihat jelas bahwa perspektif interaksi kualitas tidak terwakili dengan baik dalam *webqual* 1.0. Terkait dengan kualitas pelayanan, terutama *servqual*, digunakan untuk meningkatkan aspek kualitas informasi dari *webqual* dengan kualitas interaksi. Kualitas layanan umumnya didefinisikan dengan seberapa baik layanan yang disampaikan apakah sesuai dengan eskpektasi pelanggan. Pengembangan *webqual* 2.0 memerlukan beberapa perubahan signifikan pada instrumen *webqual* 1.0. Dalam rangka memperluas model untuk kualitas interaksi, Barnes dan Vidgen (2001), melakukan analisis terhadap instrumen *serqual* dan membuat perbandingan rinci antara *serqual* dan *webqual* 1.0. Tinjauan ini berhasil mengidentifikasi pertanyaan yang mubazir dan kemudian wilayah yang tumpang tindih dihapus, hasilnya sebagian besar pertanyaan-pertanyaan kunci dalam *servqual* tidak sesuai dengan *webqual* 2.0, jumlah instrumen dengan 24 pertanyaan tetap dipertahankan (Barnes dan Vidgen, 2001).

Webqual 1.0 mungkin kuat dalam hal kualitas informasi, Sehingga kurang kuat dalam hal interaksi layanan. Demikian juga untuk *webqual* 2.0 yang menekankan kualitas interaksi menghilangkan beberapa kualitas informasi dari *webqual* 1.0. Kedua versi tersebut mengandung berbagai kualitas terkait dengan *website* sebagai artefak perangkat lunak. Dalam tinjauan yang dilakukan oleh Barnes dan Vidgen (2001) menemukan bahwa semua kualitas dapat dikategorikan menjadi tiga wilayah yang berbeda, yaitu kualitas *website*, kualitas informasi, dan

kualitas interaksi pelayanan. Versi baru *webqual* 3.0 telah diuji dalam *domain* lelang *online* (Barnes dan Vidgen, 2001).

Analisis dari hasil *webqual* 3.0 membawa pada identifikasi tiga dimensi dari kualitas *website*, yaitu kegunaan, kualitas informasi, dan kualitas interaksi pelayanan. Kegunaan adalah kualitas yang berkaitan dengan desain *website*, misalnya penampilan, kemudahan penggunaan, navigasi dan tampilan yang disampaikan kepada pengguna. Kualitas informasi adalah kualitas isi *website*, kesesuaian informasi untuk keperluan pengguna seperti akurasi, format, dan relevansi. Kualitas interaksi layanan adalah kualitas interaksi layanan yang dialami oleh pengguna ketika mereka mempelajari lebih dalam suatu *website*, diwujudkan oleh kepercayaan dan empati, misalnya masalah transaksi dan keamanan informasi, pengiriman produk, personalisasi, dan komunikasi dengan pemilik *website* (Barnes dan Vidgen, 2001). Kegunaan telah menggantikan kualitas *website* di *webqual* versi 4.0 karena menjaga penekanan pada pengguna dan persepsi mereka daripada perancang *website*. Istilah kegunaan juga mencerminkan dengan lebih baik tingkat abstraksi dua dimensi lain dari *webqual*, yaitu interaksi layanan dan informasi. Kegunaan berkaitan dengan pragmatik tentang bagaimana pengguna melihat dan berinteraksi dengan *website* : apakah mudah bernavigasi? Apakah desain sesuai dengan jenis *website*?

Jurnal (Henny Medyawati dkk, 2012) yang di kutip dari Barnes dan Vigen, *webqual* pada dasarnya mengukur mutu sebuah *web* berdasarkan persepsi dari pengguna atau pengunjung situs. Berdasarkan hal tersebut maka pengukurannya menggunakan instrument penelitian atau kuesioner dengan struktur instrumennya mengacu ke model *Serqual*.

Barnes dan Vigen (2003) menggunakan *webqual* 4.0 untuk mengukur kualitas *website* yang dikelola oleh OECD (*Organization for Economic Cooperation and Development*). Pada tabel 2.6 uraian dimensi pengukuran kualitas layanan *website* yang digunakan.

Tabel 2.6 Dimensi Pengukuran *WebQual*

No.	Category	Webqual 4.0 Quetions
1.	Usability	<i>I fin the site easy to learn to operate</i>
		<i>My interaction with the site is clear and understandable</i>
		<i>I fin the site easy to navigate</i>
		<i>I fin the site easy to use</i>
		<i>The site has an attractive appearance</i>
		<i>The design is appropriate to the type of site</i>
		<i>The site conveys a sense of competency</i>
		<i>The site creates a positive experience for me</i>
2.	Information Quality	<i>Provides accurate information</i>
		<i>Provides believable information</i>
		<i>Provides timely information</i>
		<i>Provides relevant information</i>
		<i>Provides easy to understand information</i>
		<i>Provides information in appropriate format</i>
3.	Service Interaction	<i>Has a good reputation</i>
		<i>It feels safe to complete transaction</i>
		<i>My personal information feels secure</i>
		<i>Creates a sense of perzonalitation</i>
		<i>Convey a sense of community</i>
		<i>Makes it easy to communicate with the organization</i>
		<i>I feel confident that good / services will be delivered as promised</i>
4.	Overall	<i>Overal view of the website</i>

(Sumber : Barnes dan Vidgen, 2003)

WebQual 4.0 terdapat 23 pertanyaan yang mencakup 4 karakteristik kualitas yang dirasakan pelanggan yaitu : (1) *Information Quality* adalah mutu dari isi yang terdapat pada situs, pantas tidaknya informasi untuk tujuan pengguna seperti akurasi, format dan keterkaitannya. (2) *Service Interaction Quality* adalah mutu dari interaksi pelayanan yang dialami oleh pengguna ketika mereka menyelidiki situs lebih dalam, yang terwujud dengan kepercayaan dan empati, sebagai contoh isu dari keamanan transaksi dan informasi, pengantaran produk, personalisasi dan komunikasi dengan pemilik situs. (3) *Usability* adalah mutu yang berhubungan dengan rancangan situs, sebagai contoh penampilan, kemudahan penggunaan, navigasi dan gambaran yang disampaikan kepada pengguna. (4) *Overall* adalah keseluruhan penilaian situs *web* dari sisi pengguna.

Jurnal (Imam Sanjaya, 2012) menurut teori *WebQual*, terdapat tiga dimensi yang mewakili kualitas suatu *website*, yaitu kegunaan (*usability*), kualitas informasi (*information quality*) dan interaksi layanan (*service interaction*).

Skala pengukuran yang digunakan dalam penelitian ini adalah empat poin skala Likert. Pengguna akan diminta menilai *website* untuk kualitas masing – masing menggunakan skala mulai dari 1 (sangat tidak setuju) hingga 4 (sangat setuju). Skala sengaja dibuat genap untuk menghindari kecenderungan responden yang bersikap netral. Terlepas dari perdebatan apakah skala Likert memiliki jenis data ordinal ataukah interval, dalam penelitian ini diasumsikan data yang diperoleh adalah berjenis interval, sehingga dapat digunakan untuk statistika parametrik seperti analisis regresi linier berganda. Penelitian dilaksanakan menggunakan teknik *survey*, dimana pengumpulan data primer dilakukan melalui penyebaran daftar pertanyaan (kuesioner). Sampel yang diambil sebanyak 30 responden. Ukuran sampel ini sesuai dengan penelitian yang dilakukan oleh Barnes dan Vidgen (2001) dalam mengukur kualitas *website* berita. Sampel dipilih secara *purposive (judgment sampling)* karena responden harus memiliki kriteria pernah menggunakan *website* bersangkutan sebelumnya, sehingga semua responden yang dipilih berasal dari internal dan eksternal.