BAB 2

TINJAUAN PUSTAKA

2.1 TINJAUAN PERUSAHAAN

2.1.1 Sejarah Perusahaan

Toko sepatu jimmi adalah merupakan suatu perusahaan yang bergerak di bidang penjualan sepatu, memiliki berbagai macam model sepatu yang sudah cukup terkenal seperti: Sepatu *DC, Vans, Converse, Quicksilver, Fredperry, Allstar, Ripcurl, Macbeth, Nike, Adidas* dan lain – lain, khususnya di kalangan remaja. Toko ini baru saja berdiri pada bulan Januari 2012, yang terletak di Jl. Surapati No.16 Bandung. Tempatnya sangat strategis dekat dengan jalan raya, sehingga memudahkan kepada para pengunjung yang sedang melewati jalan tersebut untuk langsung datang ke toko. Selain itu harga sepatu yang ditawarkan bervariasi dengan harga yang kompetitif.

2.1.2 Visi dan Misi

2.1.2.1 Visi

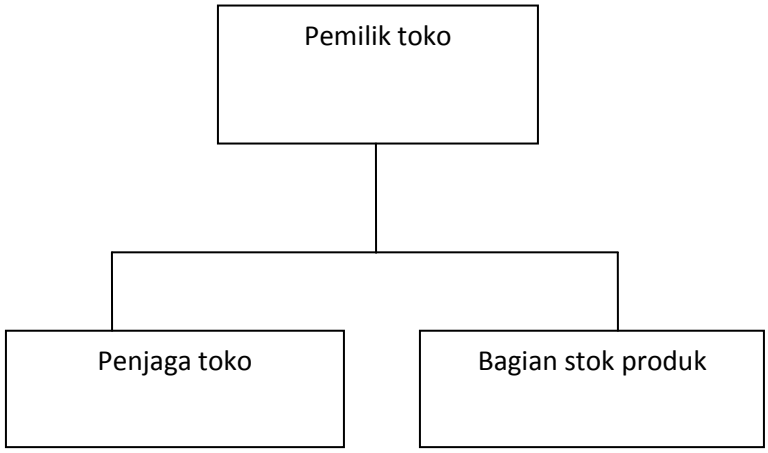
Menjadi toko yang unggul dalam bidang penjualan sepatu yang menawarkan kualitas terbaik dengan harga yang kompetitif dan memberikan kepercayaan kepada pelanggan.

2.1.2.2 Misi

1. Menyediakan berbagai model sepatu yang berkualitas dan bermutu.
2. Mengembangkan inovasi terhadap kualitas produk.

2.1.3 Struktur Organisasi

Struktur organisasi pada toko sepatu jimmi dapat dilihat pada gambar 2.1.



Gambar 2.1 Sruktur organisasi toko sepatu jimmi

2.1.4 Deskripsi Kerja

Berikut ini merupakan deskripsi kerja pada toko sepatu jimmi adalah sebagai berikut:

1. Pemilik toko
    a. Bertanggung jawab atas kelangsungan toko.
    b. Menerima laporan penjualan dan pembelian produk.

2. Penjaga toko
    a. Mencatat di buku setiap pelanggan melakukan pembelian.
    b. Melayani pelanggan yang ingin melakukan pembelian.
    c. Membuat faktur penjualan produk.
    d. Menata sepatu agar telihat rapih.

3. Bagian stok produk
    a. Melakukan pengecekan stok produk.

2.2 Landasan Teori

2.2.1 *E-commerce*

2.2.1.1 Definisi *Electronic Commerce*

Menurut Donna Perry, Pengertian *e-commerce* sangat sederhana yaitu kemampuan untuk melakukan bisnis secara elektronik melalui komputer, fax, telepon, dsb. Menurutnya untuk menjual produk dan / atau jasa di *internet*, sebuah perusahaan membutuhkan:

a) Komputer – bukan hanya beberapa komputer, tetapi dibutuhkan sebuah *server* dengan kapasitas besar dan kecepatan tinggi yang memungkinkan *Secure Socket Layer* (*SSL*) mempunyai enkripsi yang aman. *Server* ini harus benar – benar stabil.

b) *Merchant account* – yang diperoleh melalui sebuah bank atau institusi keuangan dan mengizinkan perusahaan menerima kartu kredit sebagai bentuk pembayarannya. Rekening ini sebaiknya menggunakan sebuah institusi yang mengetahui tentang perdagangan di *internet* dan menawarkan pemrosesan transaksi *online* secara *real – time*.

c) *Website* – sebuah *website e-commerce*.

2.2.1.2 Keuntungan *E-commerce*

1) Keuntungan *e-commerce* bagi bisnis:

   Dengan melakukan kegiatan bisnis secara *online*, perusahaan – perusahaan dapat menjangkau pelanggan di seluruh dunia. Oleh karena itu dengan memperluas bisnis, sama saja dengan meningkatkan keuntungan. Pelaku bisnis dapat mengumpulkan informasi mengenai para pelanggannya melalui penggunaan *cookies*. *Cookies* merupakan *file* kecil yang terdapat di dalam *hard disk* pemakai pada saat pemakai tersebut memasuki sebuah *website*. *Cookies* membantu operator *website* tadi untuk mengumpulkan informasi mengenai kebiasaan membeli yang dilakukan oleh sekelompok orang. Informasi ini tidak terhingga nilainya bagi bisnis karena informasi tadi menjadikan pelaku bisnis membuat target periklanannya lebih baik dengan informasi yang lebih baik mengenai demografis.

Keuntungan lainnya bahwa *e-commerce* menawarkan pengurangan sejumlah biaya tambahan. Sebuah perusahaan yang melakukan bisnis di *internet* akan mengurangi biaya tambahan karena biaya tersebut tidak digunakan untuk gedung dan pelayanan pelanggan (*customer service*), jika dibandingkan dengan jenis bisnis tradisional. Hal ini membantu perusahaan dalam meningkatkan keuntungannya. Salah satu jenis bisnis yang mengambil keuntungan dari *e-commerce* adalah perbankan.

2) Keuntungan *e-commerce* bagi konsumen:

Seperti halnya bisnis yang berkeinginan merangkul *e-commerce* sebagai suatu cara yang sah untuk melakukan kegiatan bisnis, konsumen juga berkeinginan mengambil keuntungan dari seluruh kemungkinan yang ditawarkan oleh *e-commerce*. Keuntungan yang terbesar bagi konsumen adalah melakukan bisnis secara *online* dengan mudah. Seorang pembeli di *internet* dapat menggunakan komputer pribadinya pagi atau malam selama 7 hari per minggu untuk membeli hampir semua barang. Seorang konsumen tidak perlu mengantri di toko atau bahkan meninggalkan rumahnya, yang dilakukan hanya mengklik sebuah produk yang ingin dibelinya, memasukkan informasi kartu kreditnya, kemudian menunggu produk itu tiba melalui pos. Beberapa perusahaan *e-commerce* telah membuat proses ini lebih mudah. Beberapa toko *online* menyimpan informasi kartu kredit

pembelinya di *server*, sehingga informasi yang dibutuhkan hanya dimasukkan sekali saja. Beberapa bisnis *online* bahkan tidak mengirimkan produk – produknya ke pelanggan melalui pos, khususnya yang menjual *software* komputer. Sebagai contoh: *beyond.com* mengizinkan para pelanggannya untuk men – *download software* yang dibelinya langsung ke komputer. Produk – produk lain seperti video dan musik akan tersedia dengan cara seperti ini pada saat mendatang, sejalan dengan meningkatnya *bandwidth* dari waktu ke waktu dan waktu *download* yang meningkat.

Keuntungan lainnya yang ditawarkan oleh *e-commerce* ke konsumen adalah pengurangan biaya. Perusahaan yang menjual saham secara *online*, seperti *e-trade.com* membebankan biaya hanya sekitar $ 10 per perdagangan, yang jauh lebih murah jika dibandingkan dengan membeli saham tersebut melalui perantara saham tradisional.

Secara ringkas keuntungan *e-commerce* tersebut adalah sebagai berikut:

a. Bagi konsumen : harga lebih murah, belanja cukup pada satu tempat.
b. Bagi Pengelola bisnis : efisiensi, tanpa kesalahan, tepat waktu.
c. Bagi Manajemen : peningkatan pendapatan, loyalitas pelanggan.

2.2.1.3 Jenis – jenis *E-commerce*

Secara umum, *e-commerce* meliputi aktifitas – aktifitas bisnis secara *online* untuk produk dan jasa yang bisa dibagi ke dalam 2 jenis *e-commerce*, yaitu *business to business* (*B2B*) dan *business to customer* (*B2C*) melalui *internet*. Perkembangan yang terjadi dewasa ini 2 jenis *e-commerce* tersebut dikembangkan lagi menjadi beberapa jenis antara lain *customer to customer* (*C2C*), *govermant to nation* (*G2N*) dan lain sebagainya. Tetapi dalam literatur – literatur yang membahas tentang *e-commerce* pada umumnya, hanya membagi menjadi 2 jenis saja seperti tersebut di atas.

1) *Business To Customer (B2C)*

   Kelompok ini disebut juga transaksi pasar. Pada transaksi pasar, konsumen mempelajari produk yang ditawarkan melalui publikasi elektronik, membelinya dengan *electronic cash* dan *system secure payment*, kemudian minta agar barang dikirimkan. Secara ringkas jenis *e-commerce* ini merupakan *e-commerce* yang melibatkan konsumen dengan *merchant*-nya secara langsung.

2) *Business To Business* (*B2B*)

   Kelompok ini disebut sebagai transaksi antar perusahaan. Perusahaan, pemerintah dan organisasi lainnya bergantung pada komunikasi antar komputer sebagai sarana bisnis yang cepat, ekonomis dan dapat diandalkan.

Perusahaan kecil saat ini sudah mulai tertarik dengan keuntungan yang diperoleh menggunakan *B2B* ini. Transaksi pada *B2B* menggunakan *EDI* dan *E-MAIL* untuk pembelian barang dan jasa, informasi dan konsultasi. Selain itu juga digunakan untuk pengiriman dan permintaan bisnis.

2.2.1.4 Komponen Utama *E-commerce*

Berikut ini adalah beberapa komponen utama pada *e-commerce*:

1) *Electronic Data Interchange*

   Electronic Data Interchange (EDI) didefinisikan sebagai pertukaran data antar komputer antar berbagai organisasi atas suatu informasi terstruktur dalam format yang *standard* dan bisa diolah oleh komputer. *EDI* merupakan bentuk *e-commerce* sesuai definisinya, dan telah ada bentuk yang sama selama lebih dari 20 tahun. Saat ini teknologi dan implementasi *EDI* sudah sangat berkembang. Tujuan *EDI* adalah untuk memfasilitasi perdagangan dengan cara mengikat aplikasi bisnis antar *partner* dagang. *EDI* meningkatkan proses manual untuk mempertukarkan informasi dengan bidang bisnis lainnya dalam berbagai cara, misalnya data hanya perlu untuk dimasukkan satu kali saja, kemudian data tersebut bisa digunakan oleh pihak pengirim barang, manager kantor dan lain – lainnya. Hal ini akan mengurangi tingkat kesalahan pengisian data, selain itu juga mengurangi

biaya untuk menggaji tenaga *entry* data. Pada dasarnya, data bisa dikirimkan dengan efisien dengan menggunakan *EDI*.

Komponen utama dari *EDI* standar adalah sebagai berikut:

a) Data *Element*

Merupakan potongan data seperti tanggal, harga atau nama organisasi. Setiap data *element* diidentifikasikan dengan nomor referensi tertentu yang berisi judul, keterangan, jenis, nomor dan panjang minimum/maximum.

b) Data *Segment*

Dalam suatu baris data disebut dengan *segment* dan setiap *item* di dalam *segment* mewakili satu elemen. Misalnya, *segment* baris pesanan pembelian terdiri atas nomor barang, keterangan, jumlah, unit pengukuran dan harga barang. Setiap *segment* memiliki satu *identifier*, satu data *element delimiter*, *element diagrams*, data *segment* terminator dan *notes*.

c) *Transaction Set*

Suatu *transaction set* merupakan dokumen khusus seperti dokumen pesanan pembelian. Di dalam *transaction set*, ada 3 area utama: area *header*, area *detail* dan area *summary.*

d) *Funtional Group*

Merupakan sekelompok *transaction set* yang sejenis. *Transaction set* di dalam *functional group* dikelompokkan berdasarkan *functional identifier* yang sama. Untuk mengirimkan transaksi *EDI* pada konsumen, diperlukan 4 fungsi dasar: *Mapping* elemen dalam suatu *database*, *extraction* atas data yang belum diidentifikasi dari *database*, *translation* atas data yang sudah di *extract* ke format *EDI* dan transmisi pesan dalam format *EDI* melalui media komunikasi.

Berikut ini adalah penjelasan singkat dari masing – masing fungsi tersebut :

1. *Mapping*

   Merupakan proses identifikasi elemen di dalam *database* yang diperlukan untuk membuat pesan dalam format *EDI*. *Mapping* adalah pekerjaan yang hanya satu kali dilakukan pada saat diperlukan transaksi *EDI* baru. *Software EDI* tidak bisa melaksanakan pekerjaan ini.

2. *Extraction*

   Merupakan proses pengumpulan data yang belum diidentifikasi dan menempatkannya ke dalam format tertentu. Secara umum, data di *extract* dari *database* dan dijadikan dalam bentuk flat *file*. Struktur dari flat *file* biasanya ditentukan oleh pembuat *translation software*.

3. *Translation*

Untuk mengirimkan pesan ke luar, ketika data yang diperlukan masih dalam bentuk flat *file*, pembentukan pesan *EDI* biasa dilakukan menggunakan *software* translasi atau *formatting*. *Software* translasi akan mengatur data menjadi struktur tertentu yang sesuai dengan kebutuhan transaksi *EDI*.

4. *Communication*

   Pengiriman/transmisi atas pesan *EDI* dikendalikan oleh *software* komunikasi, yang akan mengatur dan memelihara: nomor telepon *partner* dagang, menjalankan *automatic dialing dan up*/*downloading*, juga membuat *activity log*. Setiap pesan *EDI* dibungkus dengan amplop khusus yang bertuliskan alamat tujuan, serta jenis transaksi *EDI* sebagai *header* dan *eror checking codes* sebagai tambahan dibawahnya. Untuk keperluan penerimaan pesan *EDI*, proses tersebut tinggal dibalik.

2) Digital *Currency*

   Digital *currency* dimaksudkan untuk memungkinkan *user* untuk memindahkan dananya secara elektronik dalam lingkungan kerja tertentu. Saat ini, digital *currency* dirancang untuk versi elektronik dari uang kertas dimana memiliki atribut yang sama dengan media fisik sebenarnya baik secara anatomis maupun dari segi likuiditasnya.

Karakteristik digital *currency* adalah sebagai berikut:

a) Mewakili suatu nilai moneter tertentu.
b) Bisa ditukarkan sebagai alat pembayaran untuk barang dan jasa, mata uang dan koin serta token lainnya.
c) Bisa disimpan dan diambil lagi.
d) Sulit diduplikasi atau dipalsukan.

Jenis – jenis digital *currency* antara lain:

a. Electronic Cash

Sistem *electronic cash* telah terintegrasi sepenuhnya dengan *software web browser* untuk memudahkan pembelian barang melalui *internet*. Sistem *electronic cash* bisa menunjukkan saldo terakhir pada *user* tertentu sesuai permintaan. *Electronic cash* pada umumnya memerlukan infrastruktur *public key* dan mekanisme enkripsi tertentu. Saat ini *electronic cash* belum sepopuler pengunaan *smart card* atau model pembayaran lainnya.

b. Micropayments

Micropayments adalah pembayaran untuk *item* dengan nilai relatif rendah, misalnya informasi atau hiburan *on – line* yang biayanya bervariasi antara 1 *cent* sampai 10 *cent*. Sedangkan *Minipayment* adalah

pembayaran untuk *item* dengan nilai antara $ 0,25 sampai $ 10. Ada beberapa skema yang mampu menangani *micropayments* yaitu: *Milicent*, *eCash*, *CyberCoin*, *Mondex*, *Visa Cash* dan *NetBill*.

3) *Electronic Catalogs*

Electronic Catalogs (e-catalogs) telah berada pada aplikasi komersil yang dirancang untuk internet dan merupakan komponen utama dari sistem *e-commerce. E-catalogs* merupakan antar muka grafis (*Graphical User Interface*) yang umumnya berbentuk halaman *WWW* dimana menyediakan informasi tentang penawaran produk dan jasa. *E-catalog* umumnya mendukung *online shopping* dan kemampuan pemesanan dan pembayaran barang. Aplikasi *e-catalog* sebaiknya memiliki karakteristik seperti bersifat interaktif, mampu di *update* secara dinamis, *hypertextuality* dan *global presence*.

4) *Intranets* dan *Extranets*

Umumnya *intranet* digambarkan hanya sebagai *web server* di dalam perusahaan (internal), padahal sebenarnya *intranet* hanyalah kumpulan *website* yang dimiliki oleh suatu kelompok (biasanya perusahaan) yang bisa diakses hanya oleh anggota kelompok tersebut. Sedangkan *extranet* merupakan area tertentu dari *intranet* yang bisa diakses oleh kelompok di

luar anggota kelompok *intranet*, tapi dengan otorisasi tertentu. Fitur *intranet* standar dalam suatu organisasi memiliki 4 kemampuan dasar: *e-mail, on line publishing, on line searches* dan *application distribution*. Sedangkan *extranet* memperluas fitur ini ke *partner* bisnis jika dimungkinkan.

Keuntungan menggunakan *intranet* di dalam suatu organisasi :

1. Mempercepat proses bisnis.
2. Memfasilitasi pertukaran informasi.
3. Meningkatkan komunikasi dan kolaborasi.

2.2.2 *Internet*

Internet berasal dari kata *Interconnection Networking* yang mempunyai arti hubungan komputer dengan berbagi tipe yang membentuk sistem jaringan yang mencakup seluruh dunia (jaringan komputer global) dengan melalui jalur telekomunikasi seperti telepon, radio link, *satellite* dan lainnya.

2.2.3 *Website*

2.2.3.1 Pengertian *Website*

Web adalah salah satu aplikasi yang berisikan dokumen – dokumen multi-media (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol *HTTP* (*HyperText Transfer Protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*.

2.2.3.2 Jenis – jenis *Website*

1) *Web* statis

Web statis adalah *web* yang isinya/*content* tidak berubah – ubah. Maksudnya adalah isi dari dokumen *web* tersebut tidak dapat diubah secara cepat dan mudah. Ini karena teknologi yang digunakan untuk membuat dokumen *web* ini tidak memungkinkan dilakukan perubahan isi/data. Teknologi yang digunakan untuk *web* statis adalah jenis *client side scripting* seperti *HTML*, *Cascading Style Sheet* (*CSS*). Perubahan isi/data pada halaman *web* statis hanya dapat dilakukan dengan cara mengubah langsung isinya pada *file* mentah *web* tersebut. Misalkan jika sebuah dokumen *web* dibuat menggunakan *script HTML* maka perubahan isi dilakukan dengan cara membuka dokumen *file web* tersebut (yang berekstensi *HTML*) lalu isinya diganti langsung di dokumen *web* tersebut. Untuk melakukannya diperlukan pengetahuan yang cukup tentang *client side scripting* atau dengan kata lain hanya dapat dilakukan oleh seorang programmer *web* yang menguasai perintah – perintah *client side scripting*. Jika dokumen *web* tersebut sudah diunggah di *internet*, maka perubahan dilakukan dengan cara membuka *file* dokumen *web* tersebut di komputer local secara *offline* untuk selanjutnya diganti isinya. Hasil perubahannya kemudian disimpan dan diunggah kembali ke *internet*. Jelaslah bahwa perubahan *content* pada *web* statis tidak mudah karena harus melalui beberapa proses yang hanya dapat dilakukan

oleh seorang *user* yang memiliki pengetahuan teknis tentang bahasa pemograman *web* seperti *client side scripting* dan memiliki pengetahuan tambahan mengenai jaringan komputer dan *internet*. Contoh *web* statis di antaranya adalah *web* profil perusahaan yang lebih dominan menggunakan animasi *Flash* atau *HTML*, *web* kumpulan produk animasi.

2) *Web* dinamis

Web dinamis adalah jenis *web* yang *content*/isinya dapat berubah – ubah setiap saat. *Web* yang banyak menampilkan animasi *flash* belum tentu termasuk *web* dinamis karena dinamis/berubah – ubah isinya tidak sama dengan animasi. Untuk melakukan perubahan data, *user* cukup mengubahnya langsung secara *online* di *internet* melalui halaman *control panel* / administrasi yang biasanya telah disediakan untuk *user administrator* sepanjang *user* tersebut memiliki hak akses yang sesuai. Fitur yang disediakan untuk melakukan pengelolaan terhadap *content*/isi halaman *web* dinamis biasanya dibuat semudah mungkin, karena *user* yang akan melakukan perubahan data di halaman *web* tersebut kemungkinan bukanlah *user* yang menguasai *detail* teknis bahasa pemograman dan *database* atau biasa diistilahkan *user* biasa. Untuk membuat *web* dinamis diperlukan beberapa komponen yaitu *client side scripting* (*HTML, JavaScript*, *Cascading style sheet*), *server side scripting* seperti *PHP*, situs *web* dinamis

banyak sekali di *internet*, di antaranya: situs *web* berita, situs *web e-commerce*, situs *web e-banking*.

2.2.4 Situs *Web*

Situs *web* merupakan kumpulan dari halaman *web* yang sudah dipublikasikan di jaringan *internet* dan memilki *domain/URL* (*Uniform Resource Locator*) yang dapat diakses semua pengguna *internet* dengan cara mengetikkan alamatnya.

2.2.5 *WWW*

World wide web atau yang biasa disingkat *WWW* merupakan kumpulan situs *web* yang dapat diakses di *internet* yang berisikan semua informasi yang dibutuhkan semua pengguna *internet*. Jadi antara *web*, situs *web* dan WWW sebenarnya sama pengertiannya, hanya ruang lingkupnya yang berbeda.

2.2.6 *Browser*

Browser dibutuhkan untuk menjalankan dan menampilkan contoh – contoh kode program. Berikut adalah beberapa program aplikasi browser yang cukup popular seperti *Mozilla Firefox*, *Internet Explorer*, *Google Chrome*, *Opera*, *Safari*.

2.2.7 *Web Editor*

Web Editor merupakan aplikasi yang digunakan untuk mengetikkan kode – kode program *PHP*, *HTML*, *Javascript*. Saat ini banyak sekali program aplikasi *web editor*, di antaranya: *Notepad*, *Notepad++*, *Adobe Dreamweaver*.

2.2.8 Pemograman *Web*

2.2.8.1 *Client Side Scripting* (*CSS*)

Client side scripting adalah salah satu jenis bahasa pemograman *web* yang proses pengolahannya (baca: diterjemahkan) dilakukan di sisi *client*. Proses pengolahan *client side scripting* dilakukan oleh *web browser* sebagai *client* – nya. Di dalam *web browser* terdapat *library* yang mampu menerjemahkan semua perintah di halaman *web* yang menggunakan *client side scripting*. *Library* ini secara teknis disebut sebagai *web engine*. Masing – masing *web browser* memiliki *web engine* yang berbeda – beda. Itulah mengapa *script* yang sama dapat ditampilkan dengan *layout* berbeda – beda di *web browser* yang berbeda, karena masing – masing *web engine* menggunakan metode penerjemahan yang sedikit berbeda. Untuk menghindari hal tersebut maka gunakanlah *web browser* yang telah mengikuti standard dan telah disertifikasi oleh *world wide web consortium* (*W3C*). Selain itu, gunakanlah *style* penulisan perintah *client side scripting* yang standar sesuai dengan *W3C*. Misal untuk penulisan *HTML* gunakanlah *style* penulisan *Extensible HyperText Markup*

Language (*XHTML*) yang sudah menjadi standar internasional. Berikut adalah contoh – contoh *client side scripting*: *HyperText Markup Language* (*HTML*), *Extensible HyperText Markup Language (XHTML), Cascading Style Sheet (CSS), JavaScript, Extensible Markup Language (XML).*

1. Kelebihan *Client Side Scripting*

   Berikut adalah beberapa alasan kelebihan jika menggunakan *client side scripting*:

   a. Mudah untuk dipelajari dan digunakan, artinya untuk mempelajari *client side scripting* cukup mudah.
   b. Tidak membutuhkan pengetahuan pemrograman yang tinggi atau pengalaman pemograman yang cukup ahli.
   c. Perubahan dan pemrosesan kode programnya lebih cepat karena dilakukan langsung di sisi *client*/komputer *host* tanpa perlu melakukan proses di sisi *server* melalui jaringan *internet*. Artinya kode sumber tersebut tidak perlu dikirim ke *server* melalui jaringan *internet* untuk diproses cukup dilakukan di *browser* yang *terinstall* di komputer *client*. Tentunya hal ini dapat terjadi jika spesifikasi komputer *host* cukup tinggi untuk melakukan pemrosesannya karena semua kode sumber daya yang ada dikomputer *host*.

2. Kelemahan *Client Side Scripting*

Berikut adalah beberapa alasan kelemahan jika menggunakan *client side scripting:*

a. Kode programnya dapat dilihat melalui *browser*, sehingga dapat dikatakan tidak aman jika konteksnya adalah ingin melindungi kode sumbernya dari pihak lain. Melalui menu *view page source* yang ada di *browser*, maka *user* dapat melihat kode sumber dari dokumen tersebut di *internet*.

b. Karena pemrosesannya dilakukan di sisi *client*/komputer *host* tersebut (*memory, CPU usage*) akan digunakan secara maksimal. Hal ini mengakibatkan *client side scripting* sangat tergantung pada spesifikasi komputer *host* (*machine dependent*). Maksudnya, jika spesifikasi komputer *host* rendah (*memory, CPU, storage media*) maka dookumen *web* yang menggunakan *client side scripting* akan lambat diproses. Sebaiknya, jika spesifikasi komputer *host* tinggi maka pemrosesan dokumen *web* – nya akan cepat.

c. Masalah kompatibilatas *web browser* menjadi isu yang perlu diperhatikan. Karena beberapa *web browser* menggunakan *web engine* yang berbeda oleh masing – masing *web browser* tersebut. Jadi, sangat disarankan agar dokumen *web* yang dibuat menggunakan *client side scripting* diuji coba lebih dulu di *web browser* yang popular.

d. Minim fitur untuk pengaksesan ke sumber daya komputer. Misalnya untuk menulis ke sebuah *file* dikomputer, membaca isi *file*, membuat

file/direktori di *harddisk* komputer, dan mengakses *port – port* di komputer tidak dapat dilakukan oleh *client side scripting*.

2.2.8.2 *Server Side Scripting (SSS)*

Server side scripting adalah bahasa pemograman *web* yang pengolahannya (baca: diterjemahkan) dilakukan di sisi *server*. Maksud *server* di sini adalah *web server* yang di dalamnya telah megintegrasikan komponen *web engine*. Tugas *web engine* adalah memproses semua *script* yang termasuk kategori *client side scripting* di dalam dokumen *web*. *Web engine* biasanya harus diinstall di dalam komputer terlebih dahulu sebagai bagian terpisah dari *web server*. *Server side scripting* adalah *HTML embedded*. Maksudnya adalah semua *server*. *Server side scripting* dapat disipkan ke dalam dokumen *web* yang menggunakan HTML atau sebaliknya. Berikut adalah contoh *server side scripting*, yaitu: *Active Server Pages (ASP*), PHP, *Java Server Pages (JSP).*

1. Kelebihan *Server Side Scripting*

   Berikut adalah kelebihan *server side scripting*:

   a. Aman. Hal ini karena kode sumber *server side scripting* disimpan di *web server* yang ada di sisi *server*, sehingga *user*/pengunjung tidak dapat melihat kode sumber *server side scripting* dari sisi *client/web browser*. Hal ini berbeda dengan *client side scripting* yang dapat dilihat kode sumbernya dari sisi *web browser*.

b. Meminimalkan *traffic* di jaringan. Pada saat *user* melakukan *reguest* ke *server* maka yang dikirimkan ke *user* adalah hasil pemrosesannya saja. Karena pemrosesan dokumennya sudah dilakukan di sisi *server*, maka data yang mengalir dari *client* ke *server* atau sebaliknya adalah relatif kecil dan tidak membebani *bandwidth* di jaringan.
c. Pemrosesannya lebih cepat karena spesifikasi *hardware* untuk mesin *server* biasanya lebih tinggi (bisa menjadi kelemahan juga).
d. Mampu mendukung banyak program basis data/*database management system (DBMS*).
e. Mampu mengelola sumber daya yang ada dikomputer baik perangkat keras maupun perangkat lunak lainnya.
f. Mampu menjalankan di semua sistem operasi (*cross Platform*).
g. Tidak tergantung pada jenis *web browser* yang akan digunakan semua *script* dikelola di sisi *server/web server*.

2. Kelemahan *Server Side Scripting*

   Berikut kelemahan *server side scripting*:

   a. Karena semua pemrosesan dilakukan di sisi *server* maka dibutuhkan spesifikasi komputer *server* yang cukup tinggi agar dapat memproses *server side scripting* secara cepat. Karena itu dibutuhkan investasi yang tidak sedikit untuk pengadaannya.

b. Dibutuhkan kemampuan pemograman yang baik untuk mempelajari *server side scripting*.
c. Tidak memiliki kemampuan untuk membuat *layout*/desain halaman *web* yang menarik.

2.2.9 *HTML*

2.2.9.1 Pengantar *Html*

HTML atau *HyperText Markup Language* merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman *web*. Dokumen ini dikenal sebagai *web page*. Dokumen *HTML* merupakan dokumen yang disajikan pada *web browser*.

Pengenalan Kode *HTML*

```
<HTML>

</HTML>
```

Masing – masing baris diatas disebut *tag*. *Tag* adalah kode yang digunakan untuk me – *mark* – *up* (memoles) teks *ASCII* menjadi *file HTML*. Setiap *tag* diapit dengan tanda kurung runcing. Ada *tag* pembuka yaitu <HTML> dan ada *tag* penutup yaitu </HTML> yang ditandai dengan tanda *slash* (garing miring) di depan awal tulisannya. *Tag* di atas memberikan kaidah bahwa yang akan ditulis di antara kedua *tag* tersebut adalah isi dari dokumen *HTML*.

Dalam penamaan sebuah dokumen yang akan ditampilkan pada *web browser* maka nama yang digunakan harus diakhiri dengan ektensi *(.html*) atau *(.htm*). Ekstensi dokumen *HTML* awalnya 3 karakter, adalah untuk mengakomodasi sistem penamaan dalam *DOS*.

Pengaturan warna di halaman *HTML* menggunakan mode kombinasi *RGB* (*Red, Green, Blue*) dimana setiap warna ditampilkan dalam dua digit nilai heksadesimal (0, 1, 2, …F). Setiap bagian dua digit kode menunjukkan banyaknya intensitas dari kombinasi warna merah, hijau dan biru. Dilihat pada tabel 2.1.

Tabel 2-1 Kombinasi warna

Heksadesimal	Warna
#000000	*Black*
#FFFFFF	*White*
#FF0000	*Red*
#00FF00	*Green*
#0000FF	*Blue*
#FF00FF	*Magenta*

2.2.9.2 Definisi elemen dan *tag html*

Sebuah dokumen *HTML* disusun oleh beberapa elemen atau lebih dikenal dengan sebuah komponen – komponen dasar. Elemen dapat berupa teks murni,

atau bukan teks, atau keduanya. Elemen atau komponen tersebut misal *head*, *body*, paragraf, *list*, dan lain – lain.

Untuk menandai sebuah elemen dalam suatu dokumen *HTML* digunakan *tag*. *Tag HTML* terdiri dari sebuah [(<) (nama *tag* (>)] contoh <h1>, *tag* pada umumnya berpasangan (misalnya <h1> dengan </h1>, tanda / pada *tag* pasangan memberikan tanda bahwa *tag* tersebut merupakan pambatas akhir elemen yang dibuka oleh *tag* awal. Jadi secara umum penulisan sebuah *tag* adalah <nama *tag*>…</nama *tag*> selain itu dalam penamaan *tag* tidak menganut *case sensitive.*

Elemen dasar yang harus dimiliki dalam pembuatan dokumen *HTML* adalah *tag html, tag body* dan *tag head*. Elemen *head* berisi informasi tentang dokumen, sedangkan elemen *body* berisi teks atau elemen lainnya. Lebih jelasnya adalah sebagai berikut:

```
<html>
<head>
...informasi dokumen...
</head>
<body>
...informasi yang di tampilkan pada halaman
browser...
</body>
</html>
```

2.2.9.3 *Tag – tag* dasar *html*

1) *HTML*

Merupakan *tag* dasar yang mendifinisikan bahwa dokumen adalah dokumen *html*. *Tag* ini sebagai *tag* pertama dalam dokumen *html*. Penulisan tag seperti berikut ini:

```
<html>
. . . .
</html>
```

2) *Head*

Merupakan *tag* berikutnya setelah *tag html*, digunakan untuk menuliskan keterangan tentang dokumen *web* yang akan ditampilkan. Penulisan *tag* seperti berikut ini:

```
<html>
<head>
. . . .
</head>
. . . .
</html>
```

3) *Title*

Merupakan *tag* di dalam *head* yang digunakan untuk menuliskan judul dari dokumen *HTML*, yang akan muncul pada *caption* halaman *browser* jika halaman tersebut diakses. Penulisan *tag* seperti berikut ini:

```
<html>
<head>
<title> Judul Dokumen </title>
</head>
. . . .
</html>
```

4) *Body*

Merupakan *section* utama dalam dokumen *web*. Pada s*ection* ini semua isi dokumen yang akan ditampilkan di dalam browser harus dituliskan. Penulisan *tag* seperti berikut ini:

```
<html>
<head>
<title> Judul Dokumen </title>
</head>
<body>
. . .isi dokumen. . .
</body>
</html>
```

2.2.10 *PHP*

2.2.10.1 Pengantar *PHP*

PHP (*PHP: Hypertext Preprocessor*) adalah bahasa *server – side scripting* yang menyatu dengan *HTML* untuk membuat halaman *web* dinamis. Karena *PHP* merupakan *server – side scripting* maka sintaks dan perintah – perintah *PHP* akan dieksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dalam format *HTML*. Dengan demikian kode program yang ditulis dalam *PHP* tidak akan terlihat oleh *user* sehingga keamanan halaman *web* lebih

terjamin. *PHP* dirancang untuk membentuk halaman *web* yang dinamis, yaitu halaman *web* akan dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web*.

PHP termasuk dalam *Open Source Product*, sehingga *source code PHP* dapat diubah dan didistribusikan secara bebas. *PHP* juga dapat berjalan pada berbagai *web server* seperti *IIS* (*Internet Information Server*), *Apache*, *Xitami*. *PHP* juga mampu lintas *platform.* Artinya *PHP* dapat berjalan di banyak sistem operasi yang beredar saat ini, di antaranya: Sistem Operasi *Microsoft Windows* (semua versi), *Linux*, *Mac OS*, *Solaris*. *PHP* dapat dibangun sebagai modul pada *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai *CGI* (*Common Gateway Interface*). *PHP* dapat mengirim *HTTP header*, dapat mengatur *cookies*, mengatur *authentication* dan *redirect users*.

Salah satu keunggulan yang dimiliki oleh *PHP* adalah kemampuannya untuk melakukan koneksi berbagai macam *software* sistem manajemen basis data/*Database Management system* (*DBMS*), sehingga dapat menciptakan suatu halaman *web* yang dinamis. *PHP* mempunyai koneksitas yang baik dengan beberapa *DBMS* antara lain *Oracle*, *Sybase*, *mSQL*, *MySQL*, *Microsoft SQL Server*, *Solid*, *PostgreSQL*, *Adabas*, *FilePro*, *Velocis*, *dBase*, *Unix dbm*, dan tak terkecuali semua *database* ber – *interface ODBC*.

2.2.10.2 Mengenal perintah dasar *PHP*

Script PHP termasuk dalam *HTML – embedded*, artinya kode *PHP* dapat disisipkan pada sebuah halaman *HTML*. Parser *PHP* bekerja membaca *file HTML* sampai ditemukan penanda khusus yang memberitahukan untuk menerjemahkan skrip berikutnya sebagai kode *PHP*. Parser *PHP* akan mengesksekusi semua perintah dalam blok kode *PHP* tersebut. Dengan cara inilah maka kode skrip *PHP* dapat ditempelkan pada dokumen *HTML*. Teks lainnya yang berada di luar blok *PHP* akan dianggap sebagai skrip *HTML* biasa.

Berikut ini contoh sederhana pemakaian bahasa *PHP* yang disisipkan dalam halaman *HTML*:

```
<html>
      <head>
            <title> Skrip PHP pertama</title>
      </head>
      <body>
       <?php
        echo "Ini Skrip PHP pertamaku";
       ?>
      </body>
</html>
```

2.2.11 *Cascading Style Sheet* (CSS)

CSS (*Cascading Style Sheet*) adalah mekanisme memisahkan antara isi dan presentasi dokumen *HTML*. *CSS* menambahkan *style* pada halaman, meningkatkan dan memperbaiki penyajian isi halaman. *CSS* Bertindak sebagai lapisan yang mempengaruhi presentasi dari elemen *HTML* yang diberikan. Warna, *font, text, size, background,* susunan elemen pada halaman dan seluruh aspek presentasi isi, diatur oleh *CSS*.

2.2.12 *Web Server*

Web server adalah *software* yang menjadi tulang belakang dari *world wide web (www). Web server* menunggu permintaan dari *client* yang menggunakan *browser* seperti *Netscape Navigator*, *Internet Explorer, Modzilla*, dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*. Data ini mempunyai format yang standar, disebut dengan format *SGML* (*standar general markup language*). Data yang berupa format ini kemudian akan ditampilkan oleh *browser* sesuai dengan kemampuan *browser* tersebut. Contohnya, bila data yang dikirim berupa gambar, *browser* yang hanya mampu menampilkan teks (misalnya *lynx*) tidak akan mampu menampilkan gambar tersebut dan jika ada akan menampilkan alternatifnya saja.

Web server, untuk berkomunikasi dengan *client*-nya (*web browser*) mempunyai protokol sendiri, yaitu *HTTP* (*hypertext tarnsfer protocol*). Dengan protokol ini, komunikasi antar *web server* dengan *client*-nya dapat saling dimengerti dan lebih mudah.

2.2.13 Media Komunikasi

1) Pengertian media komunikasi

   Suatu alat yang digunakan untuk mempermudah penyampaian informasi dari seseorang kepada orang lain untuk mencapai tujuan tertentu.

2) Fungsi media komunikasi

   Media komunikasi sangat mempengaruhi efektifitas suatu komunikasi dan penggunaannya sangat erat sekali dengan penerima informasi, oleh sebab itu fungsi media komunikasi adalah sebagai berikut:

   a) Mempermudah penyampaian pesan/informasi.
   b) Memberi daya tarik terhadap pesan yang disampaikan.
   c) Mengkonkritkan isi pesan yang masih abstrak.
   d) Merupakan isi dan maksud yang disampaikan.

3) Macam – macam media komunikasi

   1 Media komunikasi berdasarkan penggunaannya, meliputi:

a. Media komunikasi berupa audio yaitu alat yang dapat didengar / ditangkap melalui alat pendengaran. Contoh: radio, *tape rocorder*, telepon.
b. Media komunikasi berupa visual yaitu alat yang dapat ditangkap / dilihat melalui alat penglihatan. Contoh: majalah, surat, telegram, faximil, media cetak seperti surat kabar, majalah, *house* jurnal (penerbitan untuk kalangan sendiri seperti buletin dan majalah dinding), media luar ruang seperti media dengan menggunakan spanduk, papan reklame/billboard, baliho.
c. Media komunikasi audio visual yaitu alat komunikasi yang dapat ditangkap, dilihat dan didengar. Contoh: televisi, film, VCD, DVD, LAPTOP.

2 Media komunikasi berdasarkan sasarannya meliputi:

a. Media komunikasi umum adalah alat penyampaian pesan yang ditujukan kepada sasaran tunggal, kelompok atau massa, contoh telepon, surat, telegram dan lain-lain.
b. Media komunikasi massa adalah alat-alat penyampaian pesan yang hanya ditujukan atau digunakan dalam komunikasi massa, contoh telepon, radio, TV dan lain-ain.

3 Media komunikasi yang sering digunakan sehari-hari:

a. Surat *prive* adalah media komunikasi visual dalam bentuk tulisan yang digunakan dalam komunikasi antar personal.

b. Telepon adalah media komunikasi audio yang digunakan dalam komunikasi antar personal yang banyak digunakan dalam lingkup kantor.

c. Pertemuan yaitu media komunikasi audio visual dalam komunikasi kelompok yang bersifat langsung, tujuannya adalah untuk membentuk saling pengertian antar anggota organisasi, baik dalam hubungan vertikal maupun horizontal dalam organisasi.

2.2.13.1 *Short Message Service (SMS)*

Short Message Service (*SMS*) adalah suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat nirkabel, yaitu perangkat komunikasi telepon selular, dalam hal ini perangkat nirkabel yang digunakan adalah telepon selular. Salah satu kelebihan dari *SMS* adalah biaya yang murah. Selain itu *SMS* merupakan metode *store* dan *forward* sehingga keuntungan yang didapat adalah pada saat telepon selular penerima tidak dapat dijangkau, dalam arti tidak aktif atau diluar *service area*, penerima tetap dapat menerima *SMS*-nya apabila telepon selular tersebut sudah aktif kembali. *SMS* menyediakan mekanisme untuk mengirimkan pesan singkat dari dan menuju

media-media *wireless* dengan menggunakan sebuah *Short Messaging Service Center* (*SMSC*), yang bertindak sebagai sistem yang berfungsi menyimpan dan mengirimkan kembali pesan – pesan singkat. Jaringan *wireless* menyediakan mekanisme untuk menemukan *station* yang dituju dan mengirimkan pesan singkat antara *SMSC* dengan *wireless station*. *SMS* mendukung banyak mekanisme *input* sehingga memungkinkan adanya interkoneksi dengan berbagai sumber dan tujuan pengiriman pesan yang berbeda.

2.2.13.2 *Facebook*

Facebook adalah sebuah layanan *jejaring social* dan situs *web* yang diluncurkan pada Februari 2004 yang dioperasikan dan dimiliki oleh *Facebook, Inc*. *Facebook* didirikan oleh Mark Zuckerberg bersama teman sekamarnya dan sesama mahasiswa ilmu komputer Eduardo Saverin, Dustin Moskovitz dan Chris Hughes. Keanggotaan situs *web* ini awalnya terbatas untuk mahasiswa Harvard saja, kemudian diperluas ke perguruan lain di Boston, Ivy League, dan Universitas Stanford. Situs ini secara perlahan membuka diri kepada mahasiswa di universitas lain sebelum dibuka untuk siswa sekolah menengah atas, dan akhirnya untuk setiap orang yang berusia minimal 13 tahun.

2.2.13.3 *Yahoo Messanger (YM)*

Yahoo! Messenger (sering disingkat "*Y!M*" atau "*YM*") merupakan program pengirim pesan instan populer yang disediakan oleh *Yahoo!*. *Yahoo! Messenger* tersedia secara gratis dan dapat diunduh serta diakses menggunakan *Yahoo! ID* yang biasa digunakan untuk mengakses layanan *Yahoo!* yang lainnya, seperti *Yahoo! Mail*. Penggunaan *ID* ini juga mengakibatkan pengguna dapat langsung diberitahu bila mendapat sebuah *e-mail*. *Yahoo* pertama kali didirikan secara tidak sengaja oleh dua orang mahasiswa *Ph.D Teknik Elektro di Stanford University*, bernama David Filo dan Jerry Yang.

2.2.13.4 *Twitter*

Twitter adalah sebuah situs *web* yang dimiliki dan dioperasikan oleh *Twitter Inc*., yang menawarkan jejaring sosial berupa mikroblog sehingga memungkinkan penggunanya untuk mengirim dan membaca pesan yang disebut kicauan (*tweets*). Kicauan adalah teks tulisan hingga 140 karakter yang ditampilkan pada halaman profil pengguna.

Twitter berawal dari sebuah diskusi yang diselenggarakan oleh anggota dewan dari Podcasting perusahaan Odeo. Dalam pertemuan tersebut, Jack Dorsey memperkenalkan ide *twitter* dimana individu bisa menggunakan *SMS* layanan untuk berkomunikasi dengan sebuah kelompok kecil. Proyek ini dimulai pada tanggal 21 secara terbuka pada tanggal 15 Juli 2006. *Twitter* menjadi

perusahaan sendiri pada bulan April 2007. Popularitas *Twitter* mulai meningkat pada tahun 2007 ketika terdapat festival *South by Southwest* (*SXSW)*. Selama acara tersebut berlangsung, penggunaan *twitter* meningkat dari 20.000 kicauan per hari menjadi 60.000. Reaksi di festival itu sangat positif. Pada tanggal 14 September 2010, *Twitter* mengganti logo dan meluncurkan desain baru.

2.2.13.5 *E-mail*

Email merupakan singkatan dari *electronic mail*, yang berarti surat elektronik. *Email* merupakan sistem yang memungkinkan pesan berbasis teks untuk dikirim dan diterima secara elektronik melalui beberapa komputer atau telepon seluler. *Email* sudah mulai digunakan pada tahun 1960an. Saat itu *internet* belum terbentuk, yang ada hanyalah sekumpulan *mainframe* yang terbentuk sebagai sebuah jaringan. Namun mulai tahun 1980-an, *email* sudah bisa dinikmati oleh masyarakat umum. Akhir – akhir ini banyak perusahaan pos dari berbagai negara di dunia menurun pendapatannya dikarenakan masyarakat yang tidak tertarik lagi memakai jasa pos.

2.2.14 *Search Engine Optimization (SEO)*

Istilah *search engine optimization* (*SEO*) pertama kali digunakan pada 26 Juli tahun 1997 oleh sebuah pesan *spam* yang diposting di *Usenet*. Pada masa itu algoritma mesin pencari belum terlalu kompleks sehingga mudah dimanipulasi.

Versi awal algoritma pencarian didasarkan sepenuhnya pada informasi yang disediakan oleh *webmaster* melalui *meta tag* pada kode *html* situs *web*. *Meta tag* menyediakan informasi tentang konten yang terkandung pada suatu halaman *web* dengan serangkaian katakunci (*keyword*).

Ada beberapa hal yang akan mendorong *website* berada di *rangking* atas pada *search engine*. Salah satunya dengan memilih nama *domain* yang menggunakan kata kunci yang banyak disukai oleh pengguna. Trik-trik yang lain adalah sebagai berikut (Shera, 2010):

1) On Page: *Title*

Title adalah judul *website.* Usahakan judul *website* juga sesuai dengan *keyword* yang paling banyak dicari. Judul yang baik adalah judul yang memenuhi beberapa syarat sebagai berikut:

a) Panjangnya 65 karakter saja. Singkat, padat, jelas, hanya berisi *keyword* yang diinginkan.
b) Apabila terdiri dari beberapa kata kunci, berikan tanda " | " untuk memisahkan.
c) Sebaiknya jangan gunakan kata hubung seperti "dan, yang, untuk, dan lain-lain". Kata hubung ini akan diskip oleh *google* sehingga hanya akan menghabiskan tempat.

d) Jangan mengulang kata yang sama lebih dari 3 kali. Penggunaan yang berlebihan dapat menyebabkan *google* melakukan *penalti* karena dianggap melakukan kecurangan atau *SPAM.*

2) On Page: *Meta description & meta keyword*

Meta description adalah uraian singkat mengenai *website* dan akan ditampilkan di hasil pencarian *search engine*. *Meta keyword* bisa diganti dalam *file index.php* atau *index.html*. Kodenya terletak di bagian atas. Beberapa jenis *CMS* memungkinkan anda menggantinya langsung dari halaman manajemen *web* tanpa harus secara manual mengubahnya di *index.html* atau *index.php.*

3) On Page: *H1 dan H2*

H1 dan H2 adalah kode yang merupakan singkatan dari *heading* 1 dan *heading 2*. *Text* apa pun yang diapit kode ini akan berukuran besar. H1 lebih besar dari H2. Contoh dari *text* yang diapit oleh kode tersebut adalah sebagai berikut:

ini adalah contoh kalimat yang diapit kode H1

ini adalah contoh kalimat yang diapit kode H2

Search engine akan menganggap kode yang diapit oleh H1 dan H2 ini sebagai kata kunci yang bobotnya lebih penting dari pada kata kunci yang didapat

dalam *content*. Dengan demikian, apabila tidak memberikan judul dalam format H1 dan H2, maka kehilangan kesempatan untuk memberitahu *google* tentang kalimat penting dari sebuah *web*.

4) *On Page*: *Keyword* di dalam halaman *website*

a) Pakai huruf besar, huruf miring, tanda kutip pada teks *keyword* di halaman *website*.

b) Ketik *keyword* di awal paragraf atau di akhir paragraf terakhir.

c) Pakai *anchor text* pada *link*. Dari pada membuat *link* *www.arwanahmad.com*, lebih baik membuat *link* arwan ahmad, yang ketika diklik mengarah ke *www.arwanahmad.com*.

5) *Off page*: *One way link* dan *two way link*

One way link adalah menaruh sebanyak-banyaknya *link website* di *website* milik orang lain. Hal ini bisa dilakukan dengan memberikan komentar-komentar di *blog* orang lain dengan mencantumkan halaman *web*. Di samping itu, juga bisa melakukan ini dengan bergabung di milis – milis yang bisa diakses oleh umum (bisa masuk ke *mail-archive.com*) dan mencantumkan alamat *web* sebagai *signature email*.

Two way link adalah menaruh sebanyak-banyaknya *link website* di *website* milik orang lain dengan imbalan orang lain juga taruh linknya di *web*.

2.2.15 *Secure Socket Layer* (*SSL*)

2.2.15.1 Perkembangan *SSL*

Pada awal perkembangan *web*, masalah keamanan telah menjadi perhatian khusus. *National Center for Supercomputing Application* (*NCSA*) merilis *browser web Mosaic* 1.0 yang cukup terkenal pada November 1993. Delapan bulan kemudian, *Netscape Communications* melengkapi rancangan *SSL* versi 1.0 dan diikuti versi – versi selanjutnya dari *SSL*. *SSL* dibangun sedemikian sehingga dapat digunakan untuk hampir semua *browser* dan *server*. *SSL* bertugas memberikan kerahasiaan, otentikasi dan integritas pesan secara aman ke pengguna *web*. Adanya *SSL* dapat diamati dari *URL "https"* yang merupakan *SSL-secured URL* atau dapat diamati dari ikon gembok kecil pada bagian kanan bawah tampilan *windows browser*. Meskipun demikian, untuk mendukung *browsing web* secara aman, *server web* tidak cukup hanya dengan menggunakan *SSL*. *Browser web* juga harus memperoleh sertifikat kunci publik dari organisasi yang dipercaya. Untuk pengguna pada internet publik, organisasi itu biasanya berkuasa atas sertifikat publik. Beberapa pemilik sertifikat yang terkemuka adalah *AT&T Certificate Services*, *GTE CyberTrust*, *KeyWitness International*, *Microsoft*, *Thawte Consulting* dan *VeriSign*.

2.2.15.2 Peran pada *SSL*

SSL mendefinisikan dua peran berbeda untuk komunikasi, yaitu *client* dan *server*. *Client* adalah sistem yang menginisiasi komunikasi. Sedangkan *server*

adalah sistem yang merespon *request* dari *client*. Dalam *SSL*, *browser web* adalah *client* dan *website* adalah *server*. *Client* dan *server* mempunyai perbedaan utama pada aksi yang dilakukan ketika melakukan negosiasi tentang parameter keamanan. *Client* bertugas untuk mengajukan opsi *SSL* yang akan digunakan pada saat pertukaran pesan, dan *server* menentukan opsi mana yang akan digunakan.

2.2.15.3 Pesan *SSL*

Client dan *server* berkomunikasi dengan bertukar pesan. Secara teknis, *SSL* mendefinisikan beberapa level pesan yang dapat dilihat pada Tabel 2.2.

Tabel 2-2 Beberapa Level Pesan pada *SSL*

Message	*Description*
Alert	Memberitahu pihak lain dalam komunikasi tentang kemungkinan adanya penerobosan pada keamanan atau kegagalan komunikasi
ApplicationData	Informasi aktual bahwa kedua pihak saling bertukar pesan yang dienkripsi, diotentikasi dan/atau diverifikasi oleh *SSL*
Certificate	Pesan yang membawa sertifikat kunci publik pengirim
CertificateRequest	*Request* dari *server* kepada *client* untuk memberikan sertifikat kunci publik
CertificateVerify	Pesan dari *client* berisi verifikasi bahwa *client* mengetahui

	kunci privat yang bersesuaian dengan sertifikat kunci publik *client*
ChangeCipherSpec	Tanda untuk mulai munggunakan layanan keamanan yang telah disetujui sebelumnya, seperti enkripsi
ClientHello	Pesan dari *client* yang memberi informasi layanan keamanan yang diinginkan dan dapat didukung oleh *client*
ClientKeyExchange	Pesan dari *client* yang membawa kunci kriptografi untuk komunikasi
Finished	Indikasi bahwa semua negosiasi awal sudah selesai dan komunikasi yang aman telah dibangun
HelloRequest	Pesan dari *server* kepada *client* untuk memulai atau *restart* proses negosiasi *SSL*
ServerHello	Pesan dari *server* yang menunjukkan layanan keamanan yang akan digunakan untuk komunikasi
ServerHelloDone	Indikasi dari *server* bahwa *server* telah melengkapi semua *request* ke *client* yang diperlukan untuk membangun komunikasi
ServerKeyExchage	Pesan dari *server* yang membawa kunci kriptografi untuk komunikasi

2.2.16 *MySQL*

MySQL dikembang oleh sebuah perusahaan Swedia bernama *MySQL AB* yang pada saat itu bernama *TcX DataKonsult AB* sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak 1979. Awalnya *TcX* membuat *MySQL* dengan tujuan mengembangkan aplikasi *web* untuk klien. *TcX* merupakan perusahaan pengembang *software* dan konsultan *database*. Saat ini *MySQL* sudah diakuisisi oleh *Oracle Corp*.

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengelolaan datanya. Kepopuleran *MySQL* antara lain karena *MySQL* menggunkan *SQL* sebagai bahasa dasar untuk mengakses *database*-nya sehingga mudah untuk digunakan, kinerja *query* cepat dan mencukupi untuk kebutuhan *database* perusahaan – perusahaan skala menengah kecil. *MySQL* juga bersifat *open source* dan *free* pada berbagai *platform* (kecuali pada *Windows*, yang bersifat *shareware*). *MySQL* didistribusikan dengan lisensi *open source GPL* (*General Public License*) mulai versi 3.23, pada bulan juni 2000.

MySQL merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (*PHP dan Perl*). *MySQL* dan *PHP* dianggap sebagai pasangan *software* pengembangan aplikasi *web* yang ideal. *MySQL* lebih sering digunakan untuk membangun aplikasi berbasis *web*, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script PHP*.

2.2.17 *Adobe Dreamweaver*

Sebuah program aplikasi yang menyediakan segala kebutuhan untuk membuat sebuah *web* yang utuh dan mudah dengan adanya *tool – tool* yang sudah terintegrasi dalam program Dreamweaver.

2.2.18 *XAMPP*

XAMPP merupakan aplikasi yang mengintegrasikan beberapa aplikasi utama *web* di dalamnya. Dalam *XAMPP* terdapat instalasi modul *PHP*, *MYSQL*, *Web Server Apache*.

2.2.19 Metode Pembayaran

2.2.19.1 Pembayaran *offline*

1. Pembayaran langsung di toko

   Metode pembayaran langsung di toko merupakan pembayaran secara umum yang telah dilakukan oleh banyak orang, biasanya pelanggan datang ke toko untuk melihat barang, apabila ada yang cocok sesuai keinginannya maka pelanggan memesan barang tersebut ke kasir atau penjaga toko. Lalu, pembayaran dilakukan secara langsung di toko tersebut kepada kasir dan pelanggan dapat membawa langsung barang yang telah dibelinya dengan membawa nota penjualan sebagai tanda bukti pembayaran.

2. Transfer Bank

Transfer adalah pemindahan dana antar rekening disuatu tempat ke tempat yang lain, baik untuk kepentingan nasabah (debitur/ non debitur) dan atau untuk kepentingan bank itu sendiri.

2.2.19.2 Pembayaran *online*

1. *Paypal*

a. Sejarah *paypal*

Perusahaan *Paypal* (*Paypal corp*.) seperti yang dikenal sekarang adalah penggabungan antara perusahaan *Confinity* dan *X.com* pada tahun 2000. *Confinity* didirikan pada desember 1998 oleh Peter Thiel dan Max Levchin, awalnya sebagai alat pembayaran *Palm Pilot* dan sebagai perusahaan kriptograpi (ilmu pembacaan sandi, tulisan – tulisan atau angka – angka rahasia). Sedangkan *X.com* didirikan oleh Elon Musk pada maret 1999, yaitu perusahaan penyedia jasa perencanaan keuangan. Kedua perusahaan lokasi pertamanya adalah kantor di 165 University Avenue di Palo Alto, California, rumah dari beberapa pemula di Silicon Valley. *Ebay* melihat perkembangan penggunaan *Paypal* dan menyimpulkan *Paypal* cocok sebagai salah satu alat pembayaran di *ebay*. Pada mei 1999 *ebay* membeli *Billpoint* dan menjadikan *Billpoint* sebagai alat pembayaran utama *ebay* saat itu dan membuat *Billpoint* khusus hanya untuk alat pembayaran di *ebay* saja, karenanya *Paypal*

hanya tampil beberapa kali sebagai pilihan alat pembayaran di *ebay* dibandingkan *Billpoint*. Tetapi karena masyarakat lebih menyukai *Paypal* karena mudah dan lebih aman, pada Februari 2000 rata- rata ada 200.000 penawaran barang perhari yang menggunakan *Paypal* sedangkan *Billpoint* hanya 4.000 penawaran. Pada april 2000 lebih dari 1 juta penawaran menggunakan *Paypal*.

b. Pengertian *paypal*

PayPal adalah salah satu alat pembayaran (*Payment procesors*) menggunakan *internet* yang terbanyak digunakan di dunia dan teraman saat ini. Pengguna *internet* dapat membeli barang di situs *e-commerce*, lisensi *software* original, keanggotaan situs, urusan bisnis, mengirim dan menerima donasi atau sumbangan, mengirim uang ke pengguna *PayPal* lain di seluruh dunia dan banyak fungsi lainnya dengan mudah dan otomatis menggunakan *internet*.

Terdapat tiga buah jenis *account* yang disediakan oleh *PayPal* untuk para konsumennya, diantaranya adalah sebagai berikut:

1) *Personal*

Dengan *account* tipe ini, pengguna sudah dapat mengirim dan menerima uang dan melakukan penjualan dengan *ebay*. Pada tipe ini, pengguna bisa menerima pembayaran dari *account Paypal* lain, namun tidak bisa menerima pembayaran dari *credit* atau *debit card*.

Tidak ada biaya untuk setiap transaksi yang dilakukan pada tipe ini. Terdapat limit berapa banyak uang yang dapat diterima per bulannya. Tipe personal ini tidak cocok untuk pengguna *PayPal* yang merencanakan untuk berjualan produk dalam jumlah yang besar.

2) *Premier*

Tipe ini hampir sama dengan *account personal*, bedanya pada tipe ini pengguna bisa menerima pembayaran dari *credit card*, *debit card* dan rekening bank. Pengguna juga bisa menggunakan fasilitas *shopping cart* dan fasilitas laporan pembayaran (*payment reporting tool*). *Account premier* cocok digunakan untuk penjual yang ingin menjual produknya secara regular. Untuk pengguna yang memiliki *account personal*, pengguna tersebut dapat melakukan *upgrade* ke *account premier*.

3) *Business*

Tipe ini cocok digunakan untuk bisnis yang berskala besar atau *online store*. Pada tipe ini pengguna diperbolehkan menggunakan laporan dan *tool ebay* tanpa adanya persyaratan mengenai jenis transaksi. Pengguna akan dikenakan biaya dalam menggunakan

account tipe ini. Jika saat ini pengguna memiliki *account personal* atau *premier*, pengguna dapat melakukan *upgrade* ke *account business*. *PayPal* tidak mengenakan biaya pada pembeli dalam mengirimkan uang ke penjual. *PayPal* akan mengenakan biaya kepada pengguna sebagai penjual (penerima uang) sebesar 1.9% hingga 2.9% dari jumlah uang yang diterima.

c. Keamanan transaksi pada *paypal*

PayPal lebih aman bila dibandingkan dengan alat pembayaran *online* yang lain. Hal ini karena pembeli yang merasa dirugikan atau tidak sesuai deskripsi yang diberitahukan, bisa meminta uang kembali kepada penjual dalam waktu maksimal 45 hari setelah terjadi pembayaran. Jika pembeli menggunakan kartu kredit akan mendapatkan pengembalian uang *chargeback* dari perusahaan kartu kreditnya. *PayPal* juga melindungi penjual dari pengembalian uang atau komplain yang tidak benar dari pembeli sesuai pembuktian.

Perlindungan untuk penjual dirancang untuk melindungi penjual dari klaim pembeli yang mengaku telah mengirim uang yang tidak mempunyai catatan bukti pembayaran dan deskripsi transaksi, setiap pembelian dan pembayaran menggunakan *PayPal* selalu ada catatan bukti pembayarannya di *account PayPal* pengirim dan penerima uang, sebagai bukti jika benar telah terjadi pengiriman uang antara keduanya.

Bisa diambil kesimpulan bahwa menggunakan *PayPal* lebih aman dari alat pembayaran *online* lain bagi pengirim dan penerima uang karena ada bukti pembayaran dan catatan deskripsi, serta dapat melakukan komplain yang benar jika terjadi sesuatu atau terhindar dari komplain yang salah.

PayPal sebagai alat pembayaran yang teraman didunia telah mengumumkan kebijakan perlindungan pembeli yang baru dan telah disempurnakan bagi para konsumen di seluruh kawasan Asia Pasifik yang ingin berbelanja *online*.

Kebijakan perlindungan pembeli *PayPal* saat ini berlaku untuk semua pengguna yang terdaftar di Asia Pasifik, termasuk Australia, China, Hongkong, Indonesia, Jepang, Malaysia, Filipina, Singapura dan Thailand. Dengan perlindungan pembeli *Paypal* ini, kini telah menjamin konsumen bahwa informasi keuangan dan transaksi *e-commerce* konsumen terlindungi dengan lebih baik saat membeli barang dari Internet. Dengan menggunakan *Paypal* maka pembelian pada transaksi *online* yang baru atau belum dikenal tidak perlu khawatir akan resiko kecurangan, keamanan dan barang tidak dikirim tidak akan terjadi.

Paypal sangat fokus pada keamanan karena salah satu hasil riset *Paypal* dengan ACNielsen menunjukkan bahwa keamanan adalah kekhawatiran utama bagi para pembelanja *online* di Australia,

sedangkan privasi informasi dan perlindungan konsumen merupakan dua kekhawatiran utama bagi para pembelanja di Hongkong.

2. *E-banking*

E-Banking bisa juga disebut *Electronic Banking* merupakan layanan perbankan yang menggunakan media elektronik sebagai perantaranya. Tujuan dari *Electronic Banking* adalah sebagai sarana penyediaan multi *channel* dan juga dapat menghemat biaya transaksi bank, nasabah lebih bebas, mudah, dan memberikan keamanan bertransaksi 24 jam sehari dimanapun nasabah berada.

Fasilitas *electronic banking* yang ditawarkan dewasa ini dibagi menjadi 3 (tiga) bagian dan masing-masing bagian memiliki sistem kerja yang menggunakan media yang berbeda. Masing-masing bagian memiliki kelebihan tersendiri. Mengenai fasilitas yang ditawarkan dari masing-masing media *electronic banking*, seperti dari *Media Internet Banking*, *Mobile Banking*, ATM dan media lainnya yang menggunakan fungsi elektronik.

Manfaat yang diberikan dari fasilitas *electronic banking* diterima oleh pihak bank selaku penyedia dan juga pihak nasabah selaku pengguna. Diantaranya bagi pihak bank *Business expansion*, *Customer loyality*, *Revenue and cost improvement, Competitive advantage, New business*

model serta Fee base Income. Selain itu manfaat bagi para nasabah diantaranya Memberikan kemudahan dan kecepatan, transaksi dimana saja dan kapan saja dapat dilakukan dan hemat biaya dan waktu.

2.2.20 *IP-Dedicated*

IP-Dedicated (*Internet Protocol*) adalah *IP address* yang digunakan secara eksklusif pada satu *account hosting*. *IP Dedicated* biasa digunakan untuk jalur transaksi yang menggunakan *SSL-enncrypted* untuk pengamanan transaksi di *internet*, biasanya digunakan oleh *website* yang memiliki *e-commerce* atau penjualan *online*.

Adapun fungsi dan keuntungan *ip dedicated*:

1. Sebuah *website* dapat diakses langsung dari *IP address* tanpa harus mengetikkan nama *domain*.
2. Menggunakan *IP Dedicated* dapat memaksimalkan *SERP* (*Search Engine Result Page*).
3. Penggunaan *IP Dedicated* dapat dipasangkan dengan *SSL- enncrypted* yang berfungsi menjaga keamanan transaksi melalui *internet* saat menggunakan Kartu Kredit.
4. *Website* lebih cepat diakses.

2.2.21 *Entity Relationship Diagram (ERD)*

ERD adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang tersimpan (*storage* data) dalam sistem secara abstrak. Diagram hubungan entitas tidak menyatakan bagaimana memanfaatkan data, membuat data, mengubah data dan menghapus.

2.2.21.1 Elemen – elemen Diagram Hubungan Entitas

a. Entity

Pada E-R diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

b. Relationship

Pada E-R diagram, *Relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship)* diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau kalaimat pasif). Penggambaran

hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

c. Relationship Degree

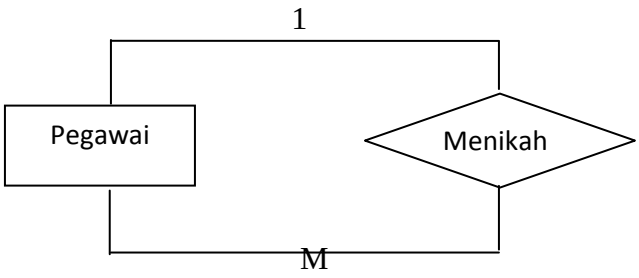
Relationship degree atau derajat *relationship* adalah jumlah entitas yang berpartipasi dalam satu *relationship*. Derajat *Relationship* yang sering dipakai di dalam *ERD*:

1) Unary Relationship

Unary Relationship adalah model *Relationship* yang terjadi diantara *entity* yang berasal dari *entity set* yang sama.

Sering juga disebut sebagai *Recursive Relationship* atau *Reflective Relationship*.

Contoh:

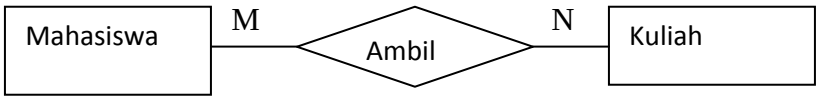


Gambar 2.2 Diagram *Relationship Unary*

2) *Binary Relationship*

Binary Relationship adalah model *Relationship* antara *instance* – *instance* dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data. Gambar dibawah menunjukkan bahwa *relationship* bekerja untuk merupakan *Relationship* banyak-ke-satu, artinya seorang pegawai hanya dapat bekerja untuk departemen dan satu departemen yang memiliki banyak pegawai.

Contoh:

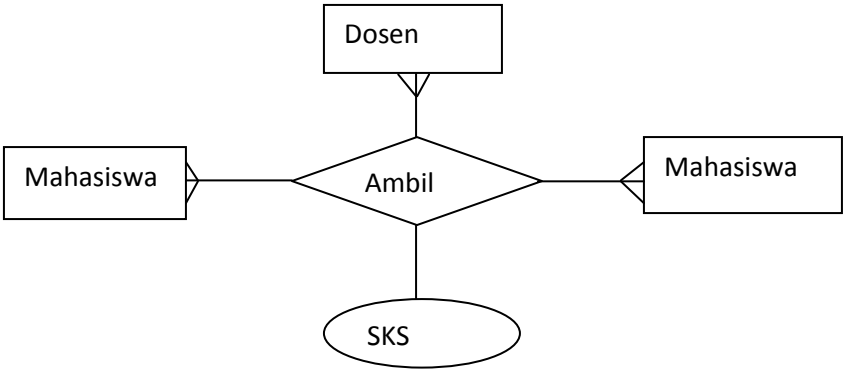


Gambar 2.3 Diagram *Relationship Binary*

3) *Ternary*

Ternary Relationship merupakan *relationship* antara *instance* – *instance* dari tiga tipe entitas secara sepihak. Pada gambar di bawah ini, *relationship* mengirimkan mencatat jumlah suatu alat tertentu yang dikirimkan oleh suatu pabrik menuju ke suatu gudang yang telah ditentukan. Masing – masing entitas mungkin berpartisipasi satu atau

banyak dalam suatu *relationship ternary*. *Relationship ternary* tidak sama dengan tiga *relationship binary*.



Gambar 2.4 Diagram *Relationship Ternary*

d. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *Relationship*, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan *Relationship*.

Atribut Value

Atribut value atau nilai *attribute* adalah suatu *occurrence* tertentu dari sebuah *attribute* di dalam suatu *entity* atau *Relationship*. Ada dua jenis atribut:

a. *Identifier (key)* digunakan untuk menentukan suatu *entity* secara unik *(primary key)*
b. *Decriptor (nonkey attribute)* digunakan untuk menspesifikasikan karakteristik dari suatu *entity* yang unik.

e. Kardinalitas (*cardinality*)

Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Pada contoh sebelumnya, dapat dlihat bahwa tupel – tupel pada entitas mahasiswa dapat berelasi dengan satu tupel, banyak tupel atau bahkan tidak satupun tupel dari entitas kuliah. Begitu juga sebaliknya, entitas – entitas pada entitas kuliah ada yang berelasi dengan beberapa tupel pada entitas mahasiswa dan ada pula yang berelasi dengan tupel pada entitas mahasiswa.

Dari sejumlah kemungkinan banyaknya hubungan antar entitas tersebut, kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat 3 macam kardinalitas relasi, yaitu:

1. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

2. *One to Many* atau *Many to One*

   Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

3. *Many to Many*

   Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi yang kedua.

2.2.22 *Data Flow Diagram (DFD)*

Diagram aliran data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Salah satu keuntungan

menggunakan diagram aliran data adalah memudahkan pemakai atau *user* yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan.

2.2.22.1 Larangan dalam DFD

Dalam menggambar/mendesain DFD ada beberapa hal yang harus dihindari, sehingga DFD tersebut menggambarkan secara keseluruhan sistem yang akan dirancang, hal – hal tersebut adalah:

a. Arus data tidak boleh dari entitas luar langsung menuju entitas luar lainnya, tanpa melalui suatu proses.
b. Arus data tidak boleh dari simpanan data langsung menuju ke entitas luar, tanpa melalui suatu proses.
c. Arus data tidak boleh dari simpanan data langsung menuju ke simpanan data lainnya, tanpa melalui suatu proses.
d. Arus data dari suatu proses langsung menuju proses lainnya, tanpa melalui suatu simpanan data, sebaiknya/sebisa mungkin dihindari.

2.2.23 Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau *output* dari sistem. Ia akan memberi tentang keseluruhan sistem. Sistem dibatasi oleh

boundary (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada *store* dalam diagram konteks.

2.2.24 Kamus Data

Kamus data berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara *detail* dan mengorganisasi semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran penyimpanan dan proses.

Kamus data sering disebut juga dengan sistem data *dictionary* adalah katalog fakta tentang data dan kebutuhan – kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap analisis, kamus data 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.

2.2.25 Normalisasi

Normalisasi adalah suatu proses untuk mengidentifikasi "*table*" kelompok atribut yang memiliki ketergantungan yang sangat tinggi antara satu atribut dengan atribut lainnya.

2.2.26 Metode Untuk *Security*

1. Pengertian Fungsi *Hash*

Hash adalah suatu teknik "klasik" dalam Ilmu Komputer yang banyak digunakan dalam praktek secara mendalam. *Hash* merupakan suatu metode yang secara langsung mengakses *record-record* dalam suatu tabel dengan melakukan transformasi aritmatik pada *key* yang menjadi alamat dalam tabel tersebut. *Key* merupakan suatu input dari pemakai di mana pada umumnya berupa nilai atau string karakter.

Pelacakan dengan menggunakan *Hash* terdiri dari dua langkah utama, yaitu:

a. Menghitung Fungsi *Hash*. Fungsi *Hash* adalah suatu fungsi yang mengubah *key* menjadi alamat dalam tabel. Fungsi *Hash* memetakan sebuah *key* ke suatu alamat dalam tabel. Idealnya, *key-key* yang berbeda seharusnya dipetakan ke alamat-alamat yang berbeda juga. Pada kenyataannya, tidak ada fungsi *Hash* yang sempurna. Kemungkinan besar yang terjadi adalah dua atau lebih *key* yang berbeda dipetakan ke alamat yang sama dalam tabel. Peristiwa ini disebut dengan *collision* (tabrakan). Karena itulah diperlukan langkah berikutnya, yaitu *collision resolution* (pemecahan tabrakan).
b. *Collision Resolution. Collision resolution* merupakan proses untuk menangani kejadian dua atau lebih *key* di-*hash* ke alamat yang sama. Cara yang dilakukan jika terjadi *collision* adalah mencari lokasi yang kosong dalam tabel *Hash* secara terurut. Cara lainnya adalah dengan menggunakan fungsi *Hash* yang lain untuk mencari lokasi kosong tersebut.

2. Sejarah MD5

MD5 adalah salah satu dari serangkaian algortima(merupakan salah satu fungsi *Hash*) *message digest* yang didesain oleh Profesor Ronald Rivest dari MIT (Rivest, 1994). Saat kerja analitik menunjukkan bahwa pendahulu MD5, yaitu MD4 mulai tidak aman, MD5 kemudian didesain pada tahun 1991 sebagai pengganti dari MD4 (kelemahan MD4 ditemukan oleh Hans Dobbertin).

Pada tahun 1993, den Boer dan Bosselaers memberikan awal, bahkan terbatas, hasil dari penemuan *pseudo-collision* dari fungsi kompresi MD5. Dua vektor inisialisasi berbeda *I* dan *J* dengan beda 4-bit diantara keduanya.

MD5compress(I,X) = MD5compress(J,X)

Pada tahun 1996 Dobbertin mengumumkan sebuah kerusakan pada fungsi kompresi MD5. Dikarenakan hal ini bukanlah serangan terhadap fungsi *hash* MD5 sepenuhnya, hal ini menyebabkan para pengguna kriptografi menganjurkan pengganti seperti WHIRLPOOL, SHA-1 atau RIPEMD-160.

Ukuran dari *hash* — 128-bit — cukup kecil untuk terjadinya serangan *brute force*.. MD5CRK adalah proyek distribusi mulai Maret 2004 dengan tujuan untuk menunjukkan kelemahan dari MD5 dengan menemukan kerusakan kompresi menggunakan *brute force attack*. Bagaimanapun juga, MD5CRK berhenti pada tanggal 17 Agustus 2004, saat [[kerusakan *hash*]] pada MD5 diumumkan oleh Xiaoyun Wang, Dengguo Feng, Xuejia Lai dan Hongbo Yu. Serangan analitik mereka dikabarkan hanya memerlukan satu jam dengan menggunakan IBM P690 cluster.

Pada tanggal 1 Maret 2005, Arjen Lenstra, Xiaoyun Wang, and Benne de Weger mendemontrasikan kunstruksi dari dua buah sertifikat X.509 dengan *public key* yang berbeda dan *hash* MD5 yang sama, hasil dari demontrasi menunjukkan adanya kerusakan. Konstruksi tersebut melibatkan *private key* untuk kedua *public key* tersebut. Dan beberapa hari setelahnya, Vlastimil Klima menjabarkan dan mengembangkan algortima, mampu membuat kerusakan Md5 dalam beberapa jam dengan menggunakan sebuah komputer notebook. Hal ini menyebabkan MD5 tidak bebas dari kerusakan.

Dikarenakan MD5 hanya menggunakan satu langkah pada data, jika dua buah awalan dengan *hash* yang sama dapat dibangun, sebuah akhiran yang umum dapat ditambahkan pada keduanya untuk membuat kerusakan lebih masuk akal. Dan dikarenakan teknik penemuan kerusakan mengijinkan pendahuluan kondisi *hash* menjadi arbitari tertentu, sebuah kerusakan dapat ditemukan dengan awalan apapun. Proses tersebut memerlukan pembangkitan dua buah *file* perusak sebagai *file template*, dengan menggunakan blok 128-*byte* dari tatanan data pada 64-*byte* batasan, *file-file* tersebut dapat mengubah dengan bebas dengan menggunakan algoritma penemuan kerusakan.

Saat ini dapat diketahui, dengan beberapa jam kerja, bagaimana proses pembangkitan kerusakan MD5. Yaitu dengan membangkitkan dua *byte string* dengan *hash* yang sama. Dikarenakan terdapat bilangan yang terbatas pada keluaran MD5 (2^{128}), tetapi terdapat bilangan yang tak terbatas sebagai

masukannya, hal ini harus dipahami sebelum kerusakan dapat ditimbulkan, tapi hal ini telah diyakini benar bahwa menemukannya adalah hal yang sulit.

Sebagai hasilnya bahwa *hash* MD5 dari informasi tertentu tidak dapat lagi mengenalinya secara berbeda. Jika ditunjukkan informasi dari sebuah *public key*, *hash* MD5 tidak mengenalinya secara berbeda jika terdapat *public key* selanjutnya yang mempunyai *hash* MD5 yang sama.

Bagaimanapun juga, penyerangan tersebut memerlukan kemampuan untuk memilih kedua pesan kerusakan. Kedua pesan tersebut tidak dengan mudah untuk memberikan serangan *preimage*, menemukan pesan dengan *hash* MD5 yang sudah ditentukan, ataupun *serangan preimage kedua*, menemukan pesan dengan *hash* MD5 yang sama sebagai pesan yang diinginkan.

Hash MD5 lama, yang dibuat sebelum serangan-serangan tersebut diungkap, masih dinilai aman untuk saat ini. Khususnya pada *digital signature* lama masih dianggap layak pakai. Seorang *user* boleh saja tidak ingin membangkitkan atau mempercayai *signature* baru menggunakan MD5 jika masih ada kemungkinan kecil pada teks (kerusakan dilakukan dengan melibatkan pelompatan beberapa bit pada bagian 128-*byte* pada masukan *hash*) akan memberikan perubahan yang berarti. Penjaminan ini berdasar pada posisi saat ini dari kriptoanalisis. Situasi bisa saja berubah secara tiba-tiba, tetapi menemukan kerusakan dengan beberapa data yang belum-ada adalah permasalahan yang lebih susah lagi, dan akan selalu butuh waktu untuk terjadinya sebuah transisi.

2.2.27 Sistem Rekomendasi

Sistem rekomendasi merupakan model aplikasi dari hasil observasi terhadap keadaan dan keinginan pelanggan. Sistem rekomendasi memanfaatkan opini seseorang terhadap suatu barang dalam *domain* atau kategori tertentu, untuk membantu seseorang dalam memilih produk. Untuk mempermudah pelanggan mengambil keputusan yang tepat dalam menentukan produk yang akan dibelinya (McGinty dan Smyth, 2006). Metode yang akan digunakan adalah *Collaborative Filtering.*

Collaborative filtering merupakan proses penyaringan atau pengevaluasian *item* menggunakan opini orang lain. *Collaborative filtering* melakukan penyaringan data berdasarkan kemiripan karakteristik konsumen sehingga mampu memberikan informasi yang baru kepada konsumen karena sistem memberikan informasi berdasarkan pola satu kelompok konsumen yang hampir sama. Perbedaan minat pada beberapa anggota kelompok menjadikan sumber informasi baru yang mungkin bermanfaat bagi anggota kelompok lainnya.

Secara umum proses pemberian rekomendasi terdiri atas tiga langkah, yaitu:

1. Penemuan similar *user*.
2. Pembuatan ketetenggaan (*neighborhood*).
3. Perhitungan prediksi berdasarkan tetangga yang dipilih.

Collaborative filtering menghasilkan prediksi atau rekomendasi bagi pengguna atau pelanggan yang dituju terhadap satu *item* atau lebih. *Rating* dalam *Collaborative filtering* dapat berbentuk, seperti:

1. Model *rating* skalar yang terdiri atas *rating* numerik seperti 1 sampai 5.
2. Model *rating* biner dengan memilih antara setuju atau tidak setuju, atau dapat pula baik atau buruk.
3. *Rating unary* dapat mengindikasikan bahwa pengguna telah mengobservasi atau membeli *item* atau merating *item* dengan positif.

Tidak tersedianya *rating* mengindikasikan tidak terdapat informasi yang menghubungkan pengguna dengan *item*. *Rating* dapat dikumpulkan secara eksplisit, implisit, ataupun gabungan antara eksplisit dan implisit. *Rating* eksplisit yaitu *rating* yang didapatkan pada saat pelanggan atau pengguna diminta menyediakan opini terhadap *item* tertentu. *Rating* implisit yaitu *rating* yang didapatkan melalui aksi yang dilakukan pelanggan.

Algoritma *Collaborative filtering* dibagi ke dalam dua kelas yang berbeda menurut teori dan kepraktisannya, yaitu algoritma non-probabilistik dan algoritma probabilistik. Suatu algoritma dianggap probabilistik bila algoritma probabilistik. Algoritma tersebut mewakili distribusi probabilitas saat menghitung prediksi *rating* atau daftar rangking rekomendasi. Algoritma non-probabalistik yang

terkenal yaitu *nearest neighbours algorithm*. Algoritma ini dibagi menjadi dua kelas yaitu *user-based* dan *item-based.*

1. *User-Based Collaborative filtering*

   User-based nearest neighbour algorithm menggunakan teknik statistika untuk menemukan sekumpulan pengguna, dikenal sebagai tetangga (*neighbour*), yang memiliki sejarah setuju dengan pengguna yang menjadi sasaran. Setelah sekumpulan tetangga terbentuk, sistem menggunakan algoritma yang berbeda untuk menggabungkan kesukaan *neighbours* untuk menghasilkan prediksi atau rekomendasi N-teratas untuk *active user*.

2. *Item-Based Collaborative filtering*

   Item-based Collaborative filtering merupakan metode rekomendasi yang didasari atas adanya kesamaan antara pemberian *rating* terhadap suatu produk dengan produk yang dibeli. Dari tingkat kesamaan produk, kemudian dibagi dengan parameter kebutuhan pelanggan untuk memperoleh nilai kegunaan produk. Produk yang memiliki nilai kegunaan tertinggilah yang kemudian dijadikan rekomendasi.

   Dalam algoritma *item-based Collaborative filtering* untuk membangkitkan rekomendasi akan dihitung dengan persamaan:

$$R = \frac{\frac{\sum B * \text{rating} * 100}{5 * 1}}{C}$$

Keterangan:

B = Nilai pembelian buku (1 jika membeli dan 0 jika tidak membeli)

Rating = Jumlah *rating* yang diberikan oleh pelanggan pada *item* yang disuka.

100 = Nilai maksimum 100, nilai rekomendasi tertinggi adalah 100.

5 = Nilai maksimum *rating* yang diberikan oleh pelanggan.

1 = Nilai maksimum pembelian yang dilakukan oleh pelanggan, jika pelanggan membeli barang lebih dari satu barang, dalam persamaan ini tetap akan dihitung membeli satu buku.

C = jumlah pengguna yang mempunyai pola pembelian yang sama dengan pelanggan yang melakukan *login*.