# BAB II

# TINJAUAN PUSTAKA

## 2.1 *E-Commerce*

### 2.1.1 Pengertian *e-commerce*

Electronic Commerce (E-Commerce) didefinisikan sebagai proses pembelian dan penjualan produk, jasa dan informasi yang dilakukan secara elektronik dengan memanfaatkan jaringan komputer. Salah satu jaringan yang digunakan adalah internet

*E-commerce* dapat didefinisikan dalam beberapa perspektif lain, antara lain,

**1.** Perspektif Komunikasi

Pada perspektif ini *E-commerce* merupakan kemampuan untuk meyampaikan produk, jasa, informasi atau pembayaran via Network seperti internet atau www.

**2.** Persfektif Interface

E-commerce melibatkan aneka informasi dan pertukaran transaksi.

**3.** Persfektif Proses Bisnis

*E-commerce* melibatkan aktifitas yang secara langsung mendukung perdagangan secara elektronik dengan menggunakan koneksi jaringan.

**4.** Persfektif Online

E-commerce adalah lingkungan elektronik yang memungkinkannya untuk membeli dan menjual produk, jasa atau informasi pada internet.

**5.** Persfektif Struktural

*E-commerce* melibatkan banyak media seperti data, teks, video, gambar, halaman web, dan internet.

### 2.1.2 Elemen – Elemen *E-commerce*

1. *Networking* atau jaringan merupakan beberapa komputer yang saling berhubungan dan saling tukar menukar informasi dan terkoneksi melalui sebuah kartu jaringan dan jalur komunikasi, yang terdiri dari jaringan local atau lebih dikenal dengan LAN dan jaringan internet yang banyak digunakan untuk jaringan LAN atau lokal terdapat berbagai jenis jaringan seperti jaringan bus, token, star dan lain-lain yang cocok untuk digunakan,
2. *Security* (keamanan) *Security* atau keamanan merupakan bagian penting, karena menyangkut masalah keamanan data Member dan juga keamanan server kita, masalah seperti enkripsi data pada saat transaksi memakai kartu kredit, aplikasi yang sudah banyak dipakai yaitu SSL *(Secure Socket Layer).*
3. *Web programming* dan *web design* Kedua masalah ini dalam pelaksanaannya dapat dikerjakan oleh satu orang atau lebih, untuk *web programming* dikhususkan dalam pembuatan bahasa pemrograman. Untuk *web design*, khusus untuk mendesain halaman web atau peraturan gambar, warna maupun tata letak suatu web dari segi keindahan dan

dinamis. *Web programming* biasanya menggunakan *script server* seperti PHP, ASP, CGI dan yang lainnya. Untuk designnya banyak software-software yang mendukung seperti Macromedia, photosop, Frontpage, Office publisher dan masih banyak lagi yang lainnya.

4. *Business Online* Maksud dari *bussines online* disini yaitu menyangkut bagaimana cara pengolahan perusahaan mulai dari manajemen, administrasi keuangan dan lainnya, yang semua itu merupakan suatu strategi dalam menarik *customer* atau pelanggan untuk datang ke toko kita.
5. *Online Payment* merupakan metode pembayaran secara *online*. Pembayaran *online* yang sering digunakan oleh beberapa sistem *e-commerce* yaitu menggunakan jasa pembayaran *Paypal*. *Paypal* adalah salah satu alat pembayaran (Payment procesors) yang terbanyak digunakan dan teraman. Pengguna internet dapat membeli barang di ebay, lisensi software original, keanggotaan situs, urusan bisnis, mengirim dan menerima donasi/sumbangan, mengirim uang ke pengguna PayPal lain di seluruh dunia dan banyak fungsi lainnya dengan mudah dan otomatis menggunakan internet atau mobile. PayPal mengatasi kekurangan dalam pengiriman uang tradisional seperti Cek atau Money order yang prosesnya dapat memakan waktu lama.

6. *Cyberlaw* adalah hukum yang digunakan di dunia *cyber* (dunia maya) yang umumnya diasosiasikan dengan internet. *Cyberlaw* merupakan aspek hukum yang ruang lingkupnya meliputi setiap aspek yang berhubungan dengan orang perorangan atau subyek hokum yang menggunakan dan memanfaatkan teknologi internet yang dimulai pada saat mulai online dan memasuki dunia *cyber* atau maya. *Cyberlaw* sendiri merupakan istilah yang berasal dari *Cyberspace Law*. *Cyberlaw* akan memainkan peranannya dalam dunia masa depan, karena nyaris tidak ada lagi segi kehidupan yang tidak tersentuh oleh keajaiban teknologi dewasa ini dimana kita perlu sebuah perangkat aturan main didalamnya (*virtual world*).

## 2.2 Perpustakaan Digital

### 2.2.1 Pengertian Perpustakan Digital

Banyak definisi dari perpustakaan digital yang dikemukakan oleh para ahli. *The digital library initiatives* menggambarkan perpustakaan digital sebagai lingkungan yang bersama-sama memberi koleksi, pelayanan dan manusia untuk menunjang kreasi, diseminasi, penggunaan dan pelestarian data, informasi dan pengetahuan. Perpustakaan digital mempunyai sistem yang berbeda dengan pelayanan perpustakaan konvensional, baik itu dalam hal penambahan buku, pencarian buku, dan pendaftaran anggota.

## 2.3 Internet

### 2.3.1 Pengertian Internet

Internet merupakan jaringan global komputer dunia, besar dan sangat luas sekali dimana setiap komputer saling terhubung satu sama lainnya dari negara ke negara lainnya di seluruh dunia dan berisi berbagai macam informasi, mulai dari text, gambar, audio, video, dan lainnya. Internet berasal dari kata *interconnection networking* yang berati kumpulan dari jaringan komputer yang saling terhubung dan dapat berkomunikasi. Jaringan komputer tersebut berkomunikasi menggunakan dua protokol yaitu IP *(internet protokol)* yang bertugas mentransmisikan data dari komputer satu dengan komputer lainnya dan TCP (*transfer control protokol*) yang bertugas untuk membuat koneksi dari satu komputer ke komputer yang lain agar dapat saling berkomunikasi.

### 2.3.2 Sejarah Internet

Sejarah internet dimulai dari proyek departemen pertahanan Amerika Serikat melalui proyek ARPA yang disebut ARPANET (A*dvanced Research Project Agency Network*) yang membuat sebuah jaringan komputer yang dapat terhubung dengan jarak yang tidak terbatas dengan menggunakan saluran telepon. Proyek ini merancang bentuk jaringan, kehandalan, dan seberapa banyak data yang dapat dipindahkan, pada akhirnya proyek ini menghasilkan sebuah standar yang menjadi cikal bakal protokol yang dikenal TCP/IP.

### 2.3.3 Layanan Aplikasi Di Internet

Terdapat banyak sekali layanan aplikasi di internet dan masih terus akan berkembang sejalan dengan perkembangan teknologi informasi, dalam laporan ini hanya akan dibahas beberapa contoh aplikasi yang banyak digunakan saja, diantaranya adalah:

1. *Electronic mail (E-mail)*

   *Email* adalah pengiriman surat elektronik baik berupa teks maupun gabungan dengan gambar yang dikirimkan dari satu alamat *email* ke alamat *email* lainnya dijaringan internet. Seperti layaknya surat biasa pada umumnya, email berfungsi untuk mengirimkan surat atau pesan kepada orang lain. Perbedaan *email* dengan surat biasa adalah *email* sudah tidak lagi membutuhan kertas sebagai media untuk menuliskan pesan. Media yang digunakan adalah data digital melalui jaringan internet.

2. *News-USENET*

   Digunakan sebagai sarana untuk berdiskusi antar pemakai jaringan Internet. Aplikasi ini hampir serupa dengan suatu papan pengumuman, dimana setiap orang dapat mengirim, malihat dan menanggapi suatu berita atau suatu topik diskusi dengan fasilitas yang hampir sama dengan e-mail. Topik diskusi dipisahkan oleh group, dan pemakai yang berminat dapat melihat isi diskusi pada newsgourp tersebut.

3. *File Transfer Protocol*

*File Transfer Protocol* (FTP) adalah suatu protokol yang berfungsi untuk tukar-menukar file dalam suatu *network* yang menggunakan TCP koneksi bukan UDP. Dua hal yang penting dalam FTP adalah FTP *Server* dan FTP *Client*. FTP server adalah suatu *server* yang menjalankan *software* yang berfungsi untuk memberikan layanan tukar menukar *file* dimana server tersebut selalu siap memberikan layanan FTP apabila mendapat permintaan *(request)* dari FTP *client*.

FTP *client* adalah komputer yang merequest koneksi ke FTP *server* untuk tujuan tukar menukar *file*. Setelah terhubung dengan FTP *server,* maka *client* dapat men-download, meng-upload, merename, men-delete, dan lain-lain sesuai dengan permission yang diberikan oleh FTP *server*.

Tujuan dari FTP *server* adalah sebagai berikut :

a. Untuk tujuan *sharing* data
b. Untuk menyediakan *indirect* atau *implicit remote* computer
c. Untuk menyediakan tempat penyimpanan bagi *Member*
d. Untuk menyediakan transfer data yang *reliable* dan efisien

4. *Remote Login-Telnet*

   Telnet adalah suatu aplikasi remote login internet yang memungkinkan anda untuk login atau menggunakan komputer yang berbeda pada jaringan secara interaktif. Untuk login dibutuhkan login *account* pada komputer tujuan, jika anda bukan Member terdaftar maka tidak bisa login ke komputer tersebut. Aplikasi ini biasanya digunakan untuk mengakses komputer berbasis sistem operasi UNIX dari tempat yang berbeda dari servernya.

5. *World Wide Web (WWW)*

   Awalanya informasi dapat dicari pada internet dengan menggunakan fasilitas *information service* berbasis *archie, gopher* dan *WAIS (Wide Area Information System).* Pencarian informasi berdasarkan menu-menu pada sistem tersebut dan output yang dihasilkan berbasis teks. Saat ini dengan teknologi *World Wide Web,* dimungkinkan untuk mengakses informasi secara interaktif, dan bentuk informasinya berupa tampilan grafis maupun teks. Hal ini dimungkinkan dengan adanya *Hypertext Transfer Protocol* (HTTP) yang digunakan untuk mengakses suatu informasi yang disimpan pada suatu situs web (*website).* Untuk dapat menggunakan sarana ini, dibutuhkan aplikasi *Web Browser.*

6. *Universal Resource Locater (URL)*

   Pada penggunaan *World Wide Web*, penunjukan suatu sumber informasi

menggunakan metode *Universal Resource Locater* (URL), yang merupakan konsep penamaan lokais standar suatu file, direktori, komputer, lokasi komputernya dan metode yang digunakan. URL tidak hanya dapat menunjuk ke suatu file tapi dapat juga menunjuk suatu query, dokumen dalam suatu *database,* atau hasil dari perintah atau yang lainnya. Dengan URL ini didefinisikan lokasi dan metode pengaksesan file tersebut.

7. *Hyper Text Transfer Protocol (HTTP)*

   Web merupakan terobosan baru bagi teknologi sistem informasi yang menghubungkan data dari banyak sumber dan layanan yang beragam macamnya di internet. Pengguna tinggal mengklik tombol mousenya pada link-link *hypertext* yang ada untuk melompat ke dokumen diberbagai lokasi di internet. Link-link sendiri bisa mengacu pada dokumen web, server FTP *(File Transfer Protocol)*, e-mail ataupun layanan lainnya.

   Server dan browser web berkomunikasi satu sama lain dengan protokol yang memang dibuat khusus untuk ini, yaitu HTTP. HTTP bertugas menangani permintaan *(request)* dari browser untuk mengambil dokumen-dokumen web.

## 2.4 Website

### 2.4.1 Pengertian Website

Website adalah kumpulan halaman-halaman situs yang saling berhubungan dalam suatu domain atau subdomain. Website biasa ditulis dalam bahasa HTML *(Hypertext Markup Language)* halaman web ini biasanya diakses melalui protokol HTTP yang menyampaikan informasi dari web server kepada user dengan menggunakan *web browser.*

### 2.4.2 Sejarah Website

Sir Timothy John Berners-Lee adalah orang yang menciptakan sebuah website pertama kali. Dia menciptakan sebuah website dengan maksud mempermudah tukar menukar atau meperbarui informasi kepada sesama peneliti di tempat dia bekerja. Pada tahun 1991 website terhubung dengan jaringan komputer untuk pertama kalinya, sedangkan pada tanggal 30 April 1993 CERN mengumumkan bahwa WWW dapat digunakan siapa saja secara gratis oleh semua orang.

## 2.5 HTML

### 2.5.1 Pengertian HTML

HTML (*Hyper Text Markup Language*) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah *browser* Internet. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan

yang disebut dengan SGML *(Standard Generalized Markup Language)*, HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World *Wide Web Consortium* (W3C).

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan browser web seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan browser.

HTML dokumen tersebut mirip dengan dokumen teks biasa, hanya dalam dokumen ini sebuah teks bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Sebagai contoh jika ingin membuat teks ditampilkan menjadi tebal seperti: TAMPIL TEBAL, maka penulisannya dilakukan dengan cara: **TAMPIL TEBAL**. Tanda digunakan untuk mengaktifkan instruksi cetak tebal, diikuti oleh teks yang ingin ditebalkan, dan diakhiri dengan tanda  untuk menonaktifkan cetak tebal tersebut.

Secara garis besar, terdapat beberapa jenis elemen dari HTML:

a. Structural. tanda yang menentukan level atau tingkatan dari sebuah teks (contoh,<h1>Golf</h1> akan memerintahkan browser untuk

menampilkan “Golf” sebagai teks tebal besar yang menunjukkan sebagai Heading 1

b. Presentational. tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut (contoh, boldface akan menampilkan bold. Tanda presentational saat ini sudah mulai digantikan oleh CSS dan tidak direkomendasikan untuk mengatur tampilan teks,

*c.* *Hypertext*. tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain (contoh, <a href="http://www.ilmukita.com/">IlmuKita</a>akan menampilkan ilmuKita sebagai sebuah hyperlink ke URL tertentu), Elemen widget yang membuat objek-objek lain seperti tombol (<button>), list (<li>), dan garis horizontal (<hr>). Selain markup presentational, markup yang lin tidak menentukan bagaimana tampilan dari sebuah teks. Namun untuk saat ini, penggunaan tag HTML untuk menentukan tampilan telah dianjurkan untuk mulai ditinggalkan dan sebagai gantinya digunakan *Cascading Style Sheets.*

Contoh HTML Sederhana

```
<!DOCTYPE html>
<html>
  <head>
```

```
  <title>Ilmu Kita : Belajar HTML</title>
</head>
<body>
  <p>Belajar HTML</p>
</body></html>
```

## 2.6 PHP

### 2.6.1 Pengertian PHP

PHP dibuat oleh Rasmus Lerdorf pada tahun 1994, pada awalnya tidak untuk didistribusikan dan hanya digunakan pada *homepage* pribadinya. Pada tahun 1995 dikeluarkan versi pertama yang dapat digunakan leh umum dengan nama personal home page tools. Ditulis kembali pada pertengahan 1995 dan diberi nama sebagai PHP/F1 version 2. F1 berasal dari paket Rasmus yang mana merupakan html interpreter untuk data *form*. Pada hasil kombinasi tersebut juga ditambah dukungan SQL. PHP/F1 terus berkembang dan banyak orang mulai memberikan kontribusi dalam pengembangannya.

Pada tahun 1996 PHP/F1 diperkirakan telah digunakan 15.000 situs web dunia, dan pada pertengahan 1997 jumlah ini berkembang melebihi 50.000. pada pertengahan 1997 juga terjadi perubahan pada PHP dimana berubah menjadi proyek yang didukung oleh team yang lebih terorganisasi. Parsernya ditulis ulang oleh Zeev Suraski dan Andi Gutmans dan parser baru inilah yang membentuk basis untuk PHP versi 3. Banyak kode utility telah selesai ditulis kembali.

PHP seperti kita ketahui ialah bahasa pemrograman berbasis *web*. Bahasa ini mempunyai kelebihan yaitu kompabilitasnya dengan berbagai macam jenis *database* ,dukungan dengan berbagai macam jenis sistem operasi.

### 2.6.2 Sejarah PHP

Pada awalnya PHP kependekan dari *Personal Home Page* yang dibuat oleh Rasmus Lerdoft pada tahun 1995 dengan nama FI *(Form Interpreted)* yang merupakan sekumpulan *script* yang digunakan untuk mengolah data form dari web. Rasmus merilis kode sumber tersebut dan menamakan PHP/FI dan diliris secara *open source* sehingga banyak programmer yang berminat untuk mengembangkan bahasa pemograman ini.

PHP 2.0 dirilis pada bukan November 1997. Pada rilis ini interpreter PHP sudah diimplementasikan dalam bahasa C dan juga disertakan modul-modul ekstensi yang meningkatkan kemampuan PHP secara signifikan. Pada tahun 1997 perusahaan bernama ZEND merilis ulang interpreter yang ada menjadi lebih baik, lebih bersih, dan lebih cepat. Pada bulan Juni 1998 ZEND merilis interpreter baru sebagai PHP 3.0 dan singkatan PHP pun berubah menjadi *Hypertext preprocessor.* Pada tahun 1999 ZEND merilis lagi PHP 4.0 yang banyak digunakan karena kemampuannya membangun aplikasi yang komplek tetapi tetap mempunyai kecepatan yang tinggi. Pada Juni 2004 PHP ZEND merilis versi terbaru yaitu versi 5.0. dalam versi ini inti dari interpreter mengalami perubahan besar karena sudah mendukung pemograman berorientasi objek.

### 2.6.3 Kelebihan PHP

Ketika *e-commerce* semakin berkembang, situs-situs yang statispun semakin ditinggalkan, karena dianggap sudah tidak memenuhi keinginan pasar, padahal situs tersebut harus tetap dinamis. Pada saat ini bahasa PERL dan CGI sudah jauh ketinggalan jaman sehingga sebagian besar *designer web* banyak beralih ke bahasa *server-side scripting* yang lebih dinamis seperti PHP.

Seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatanyang paling utama PHP adalah pada konektivitasnya dengan system *database* di dalam *web*. Sistem *database* yang dapat didukung oleh PHP adalah Oracle, MySQL , Sybase, PostgreSQL, dan lainnya. PHP dapat berjalan di berbagai system operasi seperti windows 98/NT, UNIX/LINUX, solaris maupun macintosh.

Software ini juga dapat berjalan pada web server seperti PWS (Personal Web Server), Apache, 2S, AOLServer, fhttpd, phttpd dan sebagainya. PHP juga merupakan bahasa pemograman yang dapat kita kembangkan sendiri seperti untuk menambah fungsi-fungsi baru. Keunggulan lainnya dari PHP adalah bahwa PHP juga mendukung komunikasi dengan layanan seperti *protocol* IMAP, SNMP, NNTP, POP3 dan bahkan HTTP. PHP dapat diinstal sebagai bagian atau modul dari *apache web server* atau sebagai CGI *script* yang mandiri.

Banyak keuntungan yang dapat diperoleh jika menggunakan PHP sebagai modul dari *apache,* di antaranya adalah :

1. Tingkat keamanan yang cukup tinggi

2. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemograman web lainnya yang berorientasi pada *server-side scripting*.
3. Akses ke sistem *database* yang lebih fleksibel. seperti MySQL

## 2.7 Web Server

### 2.7.1 Pengertian Web Server

*Web server* merupakan *server* internet yang mampu melayani koneksi transfer data dalam protocol HTTP. *Web server* merupakan hal yang terpenting dari server di internet dibandingkan server lainnya seperti *e-mail server*, ftp *server* ataupun *news server*. Hal ini disebabkan *web server* telah dirancang untuk dapat melayani beragam jenis data, dari text sampai grafis 3 dimensi. Kemampuan ini telah menyebabkan berbagai institusi seperti universitas maupun perusahaan dapat menerima kehadirannya dan juga sekaligus menggunakannya sebagai sarana di internet.

*Web server* juga dapat menggabungkan dengan dunia *mobile wireless* internet atau yang sering disebut sebagai WAP (*Wireless Access Protocol*), yang banyak digunakan sebagai sarana handphone yang memiliki fitur WAP. Dalam kondisi ini, *web server* tidak lagi melayani data file HTML tetapi telah melayani WML (*Wireless Markup Language*).

Salah satu *software* yang biasa digunakan oleh banyak web master di dunia adalah apache. *Software* tersebut dapat kita download secara gratis dari

web resmi apache, yaitu http://www.apache.org. Dalam Penggunaannya Apache merupakan software open source yang sekarang ini sudah merebut pasar dunia lebih dari 50%. *Web server* ini fleksibel terhadap berbagai system operasi seperti windows9x/NT ataupun unix/linux. *Apache* merupakan turunan dari webserver yang dikeluarkan oleh NCSA yaitu NCSA HTTPd pada sekitar tahun 1995.

Kelebihan web server Apache :

1. Freeware
2. Mudah diinstall.
3. Mampu beroperasi pada berbagai platform sistem operasi .
4. Mudah mengkonfigurasinya.
5. Apache Web server mudah dalam menambahkan periferal lainnya ke dalam platform web servernya, misalnya : untuk menambahkan modul, cukup hanya menset file konfigurasinya agar mengikutsertakan modul itu ke dalam kumpulan modul lain yang sudah dioperasikan.

### *2.8. Secure Socket Layer (SSL)*

#### *2.8.1.* **Pengertian** ***Secure Socket Layer (SSL)***

SSL (*Secure Socket Layer*) dikembangkan oleh *Netscape* untuk mengamankan HTTP dan sampai sekarang masih inilah pemanfaatan utama SSL. SSL menjadi penting karena beberapa produk umum seperti *Netscape Communicator, Internet Explorer,* dan WS_FTP Pro yang merupakan produk yang lazim digunakan

menggunakan SSL. *Secure Sockets Layer* adalah metode enkripsi yang dikembangkan oleh *Netscape* untuk memberikan keamanan di Internet. Ia mendukung beberapa protokol enkripsi dan memberikan autentikasi *client* dan *server*. SSL beroperasi pada *layer* transpor, menciptakan saluran enkripsi yang aman untuk data, dan dapat mengenkripsi banyak tipe data. Hal ini dapat dilihat ketika mengunjungi *site* yang aman untuk melihat dokumen *online* aman dengan *Communicator*, dan berfungsi sebagai dasar komunikasi yang aman, dengan *Communicator* juga dengan enkripsi data *Netscape Communication* lainnya. Atau dapat dikatakan bahwa SSL merupakan Protokol berlapis. Dalam tiap lapisannya, sebuah data terdiri dari panjang, deskripsi dan isi. SSL mengambil data untuk dikirimkan, dipecahkan kedalam blok-blok yang teratur, kemudian dikompres jika perlu, menerapkan MAC, dienkripsi dan hasilnya dikirimkan. Ditempat tujuan, data didekripsi, verifikasi, dekompres, dan disusun kembali. Hasilnya dikirimkan ke klien di atasnya.

### 2.8.2. Cara Kerja SSL

Dalam berkomunikasi *Secure Socket Layer* (SSL) didefinisikan secara berbeda, pada*client* atau pada *server*. *Client* adalah sistem yang menginisiasi komunikasi, sedangkan server adalah sistem yang merespon *request* dari client. Pada aplikasinya di *internet*, browser web adalah *client* dan website adalah *server*. Client dan server mempunyai perbedaan utama pada aksi yang dilakukan ketika 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. Pada implementasinya SSL berjalan pada *transport layer* dengan aplikasi SSLeay dan OpenSSL.

Secara umum cara kerja SSL dapat dijelaskan dalam beberapa tahapan yaitu :

#### 2.8.2.1. Tahapan Pembangunan Chanel

1. *Client* membentuk koneksi awal ke *server* dan meminta koneksi SSL
2. Jika *server* yang dihubungi telah dikonfigurasi dengan benar, maka server ini mengirimkan *public key* miliknya kepada client.
3. *Client* membandingkan sertifikat dari server ke basis data *trusted authorities*. Jika sertifikat terdaftar di dalamnya, artinya client mempercayai*(trust) server* itu dan akan maju ke tahap 4. Sehingga pemakai harus menambahkan sertifikat tersebut ke *trusted database* sebelum maju ke langkah 4.
4. Client menggunakan *Public Key* yang didapatnya untuk mengenkripsi dan mengirimkan *session key* ke server. Jika server meminta sertifikat *client* di tahap 2, maka client harus mengirimnya sekarang.
5. Jika server di setup untuk menerima sertifikat, maka server akan membandingkan setifikat yang diterimanya dengan basis data *trusted authorities* dan akan menerima atau menolak koneksi yang diminta. Jika kondisi ditolak, suatu pesan kegagalan akan dikirimkan ke client. Apabila koneksi

diterima, atau bila server tidak di setup untuk menerima sertifikat, maka server akan mendekripsi *session key* yang didapat dari client dengan private key milik server dan mengirimkan pesan berhasil ke client yang dengan demikian membuka suatu *secure data chanel.*

#### 2.8.2.2. Tahapan Otentikasi Server

1. *Client* mengirimkan pesan *Client Hello* untuk mengajukan opsi Ssl
2. *Server* memberi respon dengan memilih opsi SSL melalui ServerHello
3. *Server* mengirimkan sertifikat kunci publik pada pesan *Certificate*
4. *Server* mengakhiri bagian negoisasi dengan pesan *ServerHelloDone*
5. *Client* mengirimkan informasi *session key* yang dienkripsi dengan kunci publik *server* melalui pesan *ClientKeyExchange*
6. *Client* mengirimkan pesan *ChangeCipherSpec* untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan
7. *Client* mengirimkan pesan Finished sehingga memungkinkan *server* mengecek opsi baru yang diaktifkan
8. *Server* mengirimkan pesan *ChangeCipherSpec* untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan

9. *Server* mengirimkan pesan *Finished* sehingga memungkinkan *client* mengecek opsi baru yang diaktifkan.

#### 2.8.2.3. Tahapan Pemisahan Otentikasi *Server* dan Enkripsi

1. *Client* mengirimkan pesan *ClientHello* untuk mengajukan opsi SSL
2. *Server* memberi respon dengan memilih opsi SSL melalui *ServerHello*
3. Server mengirimkan sertifikat kunci publik pada pesan *Certifikat*
4. Server mengirimkan kunci publik yang harus digunakan oleh client untuk mengenkripsi kunci simetrik pada *ServerKeyExchange*, kunci ini terdapat pada sertifikat server
5. Server mengakhiri bagian negoisasi dengan pesan *ServerHelloDone*
6. Client mengirimkan informasi *session key* pada pesan ClientKeyExchange (dienkripsi dengan kuncin publik yang disediakan oleh server)
7. Client mengirimkan pesan *ChangeCipherSpec* untuk mengaktifkan opsi yang dinegoisasi untuk semua pesan yang akan dikirimkan
8. Client mengirimkan pesan *Finished* sehingga memungkinkan server mengecek opsi baru yang diaktifkan

9. Server mengirimkan pesan *ChangeCipherSpec* untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan

10. Server mengirimkan pesan Finished yang memungkinkan client mengecek opsi beru yang diaktifkan.

#### 2.8.2.4. Tahapan Otentikasi Client

1. Client mengirimkan pesan *ClientHello* untuk mengajukan opsi SSL

2. Server memberi respon dengan memilih opsi SSL melalui *ServerHello*

3. Server mengirimkan sertifikat kunci publik pada pesan *Certificate*

4. Server mengirimkan pesan *Certificate Request* untuk menunjukan bahwa server ingin mengotentikasi client

5. Server mengakhiri bagian negosiasi dengan pesan *ServerHelloDone*

6. Client mengirimkan sertifikat kunci publik pada pesan *Certificate*

7. Client mengirimkan informasi *session key* pada pesan ClientKeyExchange (dienkripsi dengan kunci publik server)

8. Client mengirimkan pesan *CertificateVerify* yang menandai informasi penting tentang sesi menggunakan kunci privat client, server menggunakan kunci publik dari sertifikat client untuk memverifikasi identitas client

9. Client mengirimkan pesan *ChangeCipherSpec* untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan

10. Client mengirimkan pesan *Finished* sehingga memungkinkan server mengecek opsi baru yang diaktifkan

11. Server mengirimkan pesan *ChangeCipherSpec* untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan

12. Server mengirimkan pesan *Finished* yang memungkinkan client mengecek opsi beru yang diaktifkan

Tahapan untuk melanjutkan sesi

1. Client mengirimkan pesan *ClientHello* yang menetapkan ID sesi sebelumnya

2. Server memberi respon dengan *ServerHello* untuk menyetujui ID sesi

3. Server mengirimkan pesan *ChangeCipherSpec* untuk mengaktifkan kembali opsi pengamanan sesi untuk pesan yang akan dikirim

4. Server mengirimkan pesan *Finished* yang memungkinkan client mengecek opsi baru yang diaktifkan kembali

5. Client mengirimkan pesan *ChangeCipherSpec* untuk mengaktifkan kembali opsi yang dinegoisasi untuk semua pesan yang akan dikirimkan

6. Client mengirimkan pesan *Finished* yang memungkinkan server mengecek opsi baru yang diaktifkan kembali

## 2.9 Basis Data

### 2.9.1. Pengertian Basis Data

Kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer dan dengan *software* untuk melakukan manipulasi, untuk kegunaan tertentu ada juga yang mendefinisikan basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut

1. Alasan Perlunya *Database*
   a. Basis data merupakan salah satu komponen penting dalam system informasi, karena merupakan dasar dalam menyediakan informasi.
   b. Basis data menentukan kualitas informasi : akurat, tepat pada waktunya dan relevan. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
   c. Basis data mengurangi duplikasi data (data *redudancy*).
   d. Dengan mengaplikasikan basis data hubungan data dapat ditingkatkan.
   e. Basis data dapat mengurangi pemborosan tempat simpanan luar.

2. Bahasa basis data

   Bahasa basis data terdiri atas :

   Data Definition Language (DDL), merujuk pada kumpulan perintah yang dapat digunakan untuk mendefinisikan objek – objek basis data, seperti membuat sebuah tabel basis data atau indeks primer atau sekunder.

   Data Manipulation Language (DML), mengacu pada kumpulan perintah yang dapat digunakan untuk melakukan manipulasi data, seperti penyimpanan data ke suatu tabel, kemudian mengubahnya dan menghapusnya atau hanya sekedar menampilkannya kembali.

3. Objektif Basis Data

   Secara lebih lengkap, pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan (objektif) seperti berikut :

   a. Kecepatan dan kemudahan (Speed)

   b. Efisiensi ruang penyimpanan (Space)

   c. Keakuratan (Accuracy)

   d. Ketersediaan (Avaibility)

   e. Kelengkapan (Completeness)

   f. Keamanan (Security)

   g. Kebersamaan pemakaian (Sharability)

### 2.9.3 SQL (Structured Query Language)

SQL adalah bahasa yang digunakan untuk berkomunikasi dengan database.

Menurut ANSI (American National Standards Institute), bahasa ini merupakan standar untuk relational database management systems (RDBMS):

Pernyataan-pernyataan SQL digunakan untuk melakukan beberapa tugas seperti : update data pada database atau menampilkan data dari database. Hampir semua software database mengimplementasikan bahasa SQL sebagai komponen utama dari produknya, salah satunya MySQL.

### 2.9.4 *MYSQL*

Untuk melakukan administrasi dalam basis data MySQL, dapat menggunakan modul yang sudah termasuk yaitu command-line (perintah: mysql dan mysqladmin). Juga dapat di-download dari situs MySQL yaitu sebuah modul berbasis grafik (GUI): MySQL Administrator dan MySQL Query Browser. Selain itu terdapat juga sebuah perangkat lunak gratis untuk administrasi basis data MySQL berbasis web yang sangat populer yaitu phpMyAdmin. Untuk perangkat lunak untuk administrasi basis data MySQL yang dijual secara komersial antara lain: MySQL front, Navicat dan EMS SQL Manager for MySQL

Bahasa SQL memiliki struktur yang mudah dipahami karena perintah – perintahnya pada dasarnya dibuat dari bahasa Inggris. Sehingga kita dapat melakukan perintah – perintah SQL ke dalam database MySQL, yaitu

a. Memasukkan atau menambah record baru ke dalam database.

b. Mengeksekusi query database

c. Mengambil data dari database

d. Mengubah record pada database

e. Menghapus record pada database

Perintah SQL dapat diketik dengan huruf besar atau kecil (non case sensitive). Setelah selesai mengetik perintah di MySQL harus diakhiri dengan tanda titik koma sebagai penanda akhir dari perintah MySQL. Perintah SQL dapat diketik dengan huruf besar atau kecil (non case sensitive). Setelah selesai mengetik perintah di MySQL harus diakhiri dengan tanda titik koma sebagai penanda akhir dari perintah MySQL.

### 2.9.5 ERD

Entity Relationship Diagram (ERD) adalah ilustrasi dari entitas-entitas dalam bisnis dan relationship antar entitas. ERD memisahkan antara informasi yang dibutuhkan dalam bisnis dari aktivitas-aktivitas yang dilakukan dalam bisnis. Jadi, meskipun terjadi perubahan proses bisnis, jenis informasi hampir tetap konstan. Oleh karena itu, struktur data juga hampir tidak berubah. Tujuan utama dari penggambaran ERD adalah untuk menunjukkan struktur objek data (entity) dan hubungan (relationship) yang ada pada objek tersebut. ERD berguna bagi profesional sistem, karena ERD memperlihatkan hubungan antara data store pada Data Flow Diagram

### 2.9.6 *Flowmap*

Merupakan diagram alir yang menunjukan arus bagi dokumen, aliran data fisik entitas sistem informasi dan kegiatan operasi yang berhubungan dengan sistem informasi. Penggambaran niasanya diawali dengan mengamati dokumen apa yang menjadi media data atau informasi dan selanjutnya ditelusuri bagaimana dokumen termasuk ke bagian entitas mana dokumen tersebut, proses apa yang terjadi terhadap dokumen tersebut dan seterusnya.

### 2.9.7 DFD *(Data Flow Diagram)*

*Data flow Diagram* (DFD) adalah diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari sistem. DFD sering digunakan untuk menggambarkan sustu 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, harddisk, tape, diskette, dan lain sebagianya).

Simbol-sombol yang digunakan di DFD mewakili maksud tertentu, yaitu:

1. External entity (kesatuan Luar) atau boundary (batas sistem)

   Setiap sistem pasti memiliki batas sistem (boundary) yang memisahkan suatu sistem dengan lingkungan luarnya. Kesatuan luar (external entity)

merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainya yang berada di lingkungan luarnya yang memberikan input atau menerima output dari sistem.

2. Data flow (arus data)

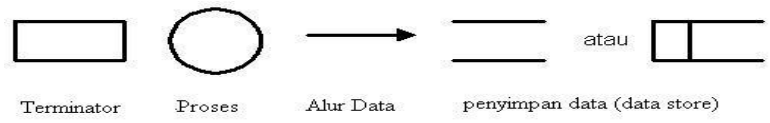
   Arus data di DFD diberi simbol panah. Arus data ini mengalir diantara proses, simpanan, dan kesatuan luar.

3. Process (proses)

   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.

4. Data store (simpanan data)

   Simpanan data (data store) merupakan simpanan dari data yang dapat berupa suatu file atau database di komputer, suatu arsip atau catatan manual dan lain sebagainya.




*Gambar 2.1 Komponen DFD*

### 2.9.8 *Data Dictionary* (Kamus Data)

Adalah daftar organisasi semua elemen yang ada dalam sistem secara lengkap dengan definisi yang baku sehingga Member dan analisis sistem akan memiliki pengertian yang sama untuk input, output, komponen penyimpanan dan perhitungannya. Kamus data dapat digunakan pada saat analisis sitem atau perancangan sistem, kamus data digunakan untuk mencatat terminologi bisnis, aturan standar (batasan panjang karakter, nilai, *system field*).

Untuk membuat spesifikasi elemen data, digunakan notasi struktur data yaitu:

Table 2.1 kamus data

| Notasi | Keterangan |
|---|---|
| = | Terdiri dari, sama dengan, diuraikan |
| + | Dan |
| () | Pilihan, boleh atau tidak |
| N{}M | Iterasi atau pengukuran mulai N kali sampai M kali |
| [] | Pilih salah satu pilihan |
| \| | Pemisalan dalam notasi [] |
| * | Keterangan, komentar atau saran |
| @ | *Key field* |

## 2.10 *AJAX*

Menyerupai J2EE, AJAX adalah 2 hal yang digabungkan merupakan sebuah teknologi, sekaligus arsitektur pemrograman.

### 2.10.1. *AJAX* sebagai sebuah teknologi

AJAX adalah singkatan dari "*Asynchronous JavaScript* and XML", yang dibuat dari serangkaian teknologi dengan berbagai kemampuan : *JavaScript*, XML dan sebuah method komunikasi asinkron antara *client* dan *server*.

3 teknologi yang saling berinteraksi *JavaScript* menangkap isyarat, gerak serta aksi. Sebagaimana pada situasi yang mungkin terjadi, *JavaScript* menggunakan jalur komunikasi pada *server* (*object JavaScript* dengan nama *XMLHttpRequest*) untuk memanggil method yang tersimpan pada *server* dan menggunakan XML sebagai mekanisme pengiriman data. Jika *JavaScript* pada *client* telah menerima respon dari server, maka *JavaScript* akan menggunakan kemampuan bawaannya untuk memanipulasi struktur DOM halaman untuk menambahkan *content* yang didapat dari *server*. Perubahan yang terjadi pada struktur DOM kemudian diterjemahkan oleh *browser* pada *client*, sehingga meningkatkan efek interaktivitas pada *Member*.

### 2.10.2. *AJAX* sebagai sebuah Arsitektur

Pada awalnya AJAX dibuat sebagai sebuah rangkaian dari teknologi, namun kemudian AJAX mengalami perkembangan. Sebagai contoh, adanya aplikasi *web* yang tidak menggunakan XML dalam mentransfer data dari client ke *server*. Cara

tersebut dilakukan dengan menggunakan object *XMLHttpRequest*.

Berdasarkan realita di atas, beberapa pihak mendefinisikan sebagai paradigma baru dalam pemrograman, disamping teknologi yang menyediakan fungsionalitas. Mari kita bahas lebih mendalam tentang arsitektur AJAX.

Pengembang telah mengembangkan *web programming* sebelum hadirnya AJAX *action* dari *Member* yang membutuhkan data dari server ditampilkan dalam halaman yang digunakan, dimana data *request* dari *Member* dikirimkan menuju *server*. Setelah mengolah halaman tersebut, server menampilkan halaman baru bagi *Member* yang mengandung hasil dari proses sebelumnya.

Permasalah dari macam arsitektur tersebut adalah lambat dan cukup memakan waktu, terutama bila dibandingkan dengan aplikasi *desktop*. Aplikasi desktop mampu merespon cepat atas *request* dari *Member*, aplikasi ini tidak memproses ulang masing – masing komponen *interface* yang akan ditampilkan sebagai respon.

AJAX menggunakan arsitektur pemrograman tersebut pada aplikasi *Web*. Daripada memberikan sebuah halaman penuh pada *server* dan mendapatkan pula sebuah halaman penuh sebagai hasil operasi, AJAX mengijinkan kita untuk mengirimkan *request* dalam ukuran yang lebih kecil pada *server*. Halaman yang terpakai hanya termodifikasi untuk menampilkan hasil, bukan tergantikan dengan sebuah halaman baru.

Faktor penting yang lain dari arsitektur AJAX adalah *request* dan *response*

dijalankan secara asinkron AJAX tidak melarang Member untuk melakukan proses lain pada halaman yang dipakai. *Member* dapat mengisi dan menggunakan area lain pada halaman, sedangkan AJAX bekerja pada *background*.

Yang terakhir, AJAX mengijinkan *Member* untuk berinteraksi dengan *server* sebagai respon terhadap seluruh hal yang dilakukan oleh *Member*. Arsitektur yang ada sebelumnya hanya mengijinkan kita untuk berkomunikasi dengan *server* pada saat *Member* menekan tombol atau link yang akan mengirim data pada halaman. AJAX memperbolehkan untuk *me-request* data baru dari *server* dalam bentuk *mouseovers*, *keypress* dan even lain yang dikenali oleh *JavaScript*.

## 2.11 *Paypal*

*PayPal* adalah salah satu alat pembayaran (*Payment procesors*) menggunakan internet yang terbanyak digunakan didunia dan teraman. Pengguna internet dapat membeli barang di *ebay*, lisensi *software* original, keanggotaan situs, urusan bisnis, mengirim dan menerima donasi/sumbangan, mengirim uang ke pengguna *PayPal* lain diseluruh dunia dan banyak fungsi lainnya dengan mudah dan otomatis menggunakan internet, *PayPal* mengatasi kekurangan dalam pengiriman uang tradisional seperti Cek atau *Money order* yang prosesnya dapat memakan waktu PayPal seperti rekening bank, pertama anda membuat *account*, lalu mengisi *account* tersebut dengan dana dari kartu kredit atau transferan dana dari *account paypal* orang lain ke *balance paypal* anda, dan anda sudah dapat menggunakan *account PayPal* untuk bertransaksi.

### 2.11.1 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.

Pada Februari 2008, *PayPal* sudah menerima 190 negara dan 16 mata uang, pengguna *PayPal* Indonesia masih harus menggunakan hitungan *US dollar* karena rupiah belum ada di *PayPal.*

Kebijaksanaan perlindungan tertulis untuk pembeli yang menggunakan PayPal menyatakan pembeli yang menggunakan PayPal dapat melakukan komplain dalam waktu 45 hari jika pembeli belum mendapatkan barang yang dipesan atau jika barang yang dipesan tidak sesuai deskripsi yang di beritahukan penjual. Jika pembeli menggunakan kartu kredit akan mendapatkan pengembalian uang chargeback dari perusahaan kartu kreditnya PayPal juga melindungi penjual dari pengembalian uang atau komplain yang bohong dari pembeli tergantung situasi dan pembuktian. Kebijakan perlindungan tertulis untuk penjual dirancang untuk melindungi penjual dari klaim pembeli yang mengaku telah mengirim uang yang tidak ada catatan bukti pembayaran dan catatan transaksi, setiap pembelian menggunakan PayPal selalu ada catatan bukti pembayarannya di *account* PayPal pengirim dan penerima uang, sebagai bukti jika benar telah terjadi pengiriman uang antara keduanya.

Dari sini bisa diambil kesimpulan menggunakan PayPal lebih aman dari alat pembayaran *online* lain bagi pengirim dan penerima uang karena ada bukti pembayaran dan dapat melakukan komplain yang benar jika terjadi sesuatu atau terhindar dari komplain yang salah.

### 2.11.2. IPN (*Instant Payment Notification*)

*Instant Payment Notification* (IPN) adalah layanan pesan yang memberitahu Anda tentang peristiwa yang berkaitan dengan transaksi PayPal. Anda dapat menggunakannya untuk mengotomatisasi back-office dan administrasi fungsi, seperti memenuhi pesanan, pelacakan pelanggan, dan memberikan status dan

informasi lain yang berkaitan dengan transaksi. IPN memberitahu Anda ketika terjadi proses yang mempengaruhi transaksi. Biasanya, proses ini mewakili berbagai jenis pembayaran, namun proses juga dapat mewakili otorisasi, Manajemen Penipuan Filter tindakan dan tindakan lainnya, seperti pengembalian uang, perselisihan, dan *chargeback*s. IPN adalah layanan pesan yang menggunakan PayPal yang melakukan notifikasi kepada anda tentang proses yang sedang berlangsung. proses-proses ini adalah:

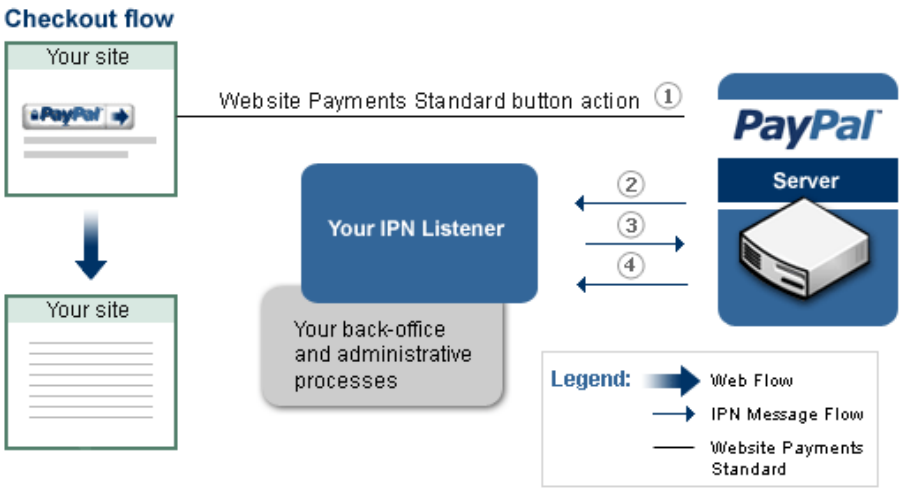
1. *Instant payment*, termasuk Express Checkout dan pembayaran kartu kredit langsung.
2. pembayaran *eCheck* dan status terkait, seperti yang tertunda, selesai, atau ditolak.
3. Pembayaran yang mungkin tertunda karena alasan lain, seperti proses pengkajian untuk potensi penipuan pembayaran.
4. Proses yang terkait dengan pembayaran berkala dan langganan Otorisasi, yang mengindikasikan pembayaran penjualan yang belum dibayarkan.
5. *Chargebacks*, yang dilakukan oleh prosesor kartu kredit, misalnya, ketika seorang pelanggan mempunyai perselisihan dengan tagihannya
6. Sengketa, yang dilakukan oleh pembeli menggunakan proses penyelesaian PayPal
7. Pembalikan, yang terjadi ketika Anda menang sengketa atau

pengembalian dana dibatalkan

8. Pengembalian uang yang Anda sudah bayarkan.

### 2.11.2.1. Cara Kerja IPN Paypal

Permintaan dan tanggapan dari IPN ini dibuat ketika pengguna menekan tombol API operasi di paypal, .PayPal mengirim pesan IPN ketika mengirimkan tanggapan terhadap permintaan. Pesan IPN sebenarnya bukan bagian dari respons yang dikirim ke situs Web. Sebaliknya, IPN ini pesan dikirim *IPN listener* yang ada di situs web ,yang memungkinkan untuk mengambil tindakan yang tidak secara langsung terikat untuk pengoperasian situs web. Cara kerja dari IPN ini digambarkan oleh gambar 2.2.




Gambar 2.2 Cara Kerja IPN

Langkah-langkah yang ditunjukan oleh gambar tersebut adalah:

1. Tombol pembayaran diklik sebagai tanda pembayaran pada PayPal

2. PayPal IPN *listener* mengirimkan pesan yang memberitahu dimulainya proses pembayaran

3. IPN *listener* mengirimkan pesan lengkap tidak berubah kembali ke PayPal ; pesan harus berisi bidang yang sama dalam urutan yang sama dan dikodekan dengan cara yang sama seperti aslinya

4. PayPal mengirim kembali kata tunggal, yaitu VERIFIED jika pesan tersebut sama dengan pesan dari paypal atau INVALID PayPal atau jika ada ketidaksesuaian dengan apa yang awalnya dikirim

IPN listener harus menerapkan protokol otentikasi IPN (langkah 2, 3, dan 4 dalam diagram). Setelah berhasil menyelesaikan protokol, back-office atau administrasi dalam sistem, memroses isi pesan dan merespon dengan tepat. Sebagai contoh, jika status pembayaran transaksi tersebut "Complete," sistem dapat mencetak daftar kemasan atau mengirim email untuk pelanggan untuk men-download media digital.

### 2.11.3. Paypal sandbox

Paypal sandbox adalah lingkungan pengujian yang merupakan duplikat dari situs PayPal. *Paypal Sandbox* memungkinkan Anda untuk menguji integrasi sebelum mengirimkan seluruh transaksi terhadap lingkungan hidup PayPal. Membuat dan

mengelola pengujian akun, melihat email *dan API credentials* untuk pengujian akun tersebut.

#### 2.11.2.1. Pendaftaran Paypal Sandbox

Untuk dapat mengakses paypal sandbox diperlukan sebuah akun yang dapat mengakses sandbox tersebut. Adapun cara pendaftarannya adalah sebagai berikut:

1. Akses https://developer.paypal.com/.
2. Apa bila telah memiliki akun paypal, masukan email dan password dan klik "*login*".
3. Apabila tidak mempunyai akun paypal klik "*sign up now*" dan isi data-data diform pendaftaran.
4. Setelah mendaftar paypal mengirim informasi login melalui email yang digunakan untuk mendaftar.
5. Pada halaman utama setelah login terdapat menu-menu yang digunakan untuk melakukan konfigurasi *paypal sandbox*, menu-menu itu diantaranya:
   a. *Test Accounts* digunakan untuk membuat akun yang akan digunakan untuk melakukan pengujian.
   b. *Test Email* digunakan untuk melakukan pengujian terhadap email akun uji.
   c. API credentials digunakan untuk melihat API credentials dan tanda tangan digital dari akun uji.
   d. *Test tool.* Digunakan untuk simulasi IPN.

#### 2.11.2.2. Membuat Akun Uji

Untuk melakukan pengujian diperlukan sebuah akun uji. Adapun cara mendapatkannya adalah.

a. Setelah login pilih link *Test Accounts*
*b.* Pilih *buyer* untuk *account type* dan isi pilihan lain atau membiarkan secara default.
*c.* Klik Create Account.

#### 2.11.3.3. Membuat Bisnis Akun Uji

Untuk melakukan pengujian sebagai penjual diperlukan sebuah akun uji penjual. Adapun cara mendapatkannya adalah.

d. Setelah login pilih link *Test Accounts*
*e.* Pilih *seller* untuk *account type* dan isi pilihan lain atau membiarkan secara default.
*f.* Klik Create Account.

### *2.12. Google Book Search*

Google Buku atau *Google Books* adalah sebuah layanan mesin pencari buku oleh Google.dengan mesin pencari buku ini, pengguna dapat mengetahui pengarang, jumlah halaman, tahun terbitan, dan membeli buku tersebut.Pengguna juga dapat melihat cuplikan isi buku yang dibatasi.

### *2.12.1. API Google Book Search*

API Pencarian Buku memungkinkan fitur *Google Book Search* dipakai untuk situs lain atau sebuah aplikasi. Data API memungkinkan user untuk mengakses layanan pencarian buku yang ada di google book.

Menggunakan API Data, program dapat pengakses layanan yang tersedia di website Pencarian Buku Google (*Google book search*), seperti mencari buku, mengumpulkan rating, dan banyak lagi.

Data API yang digunakan oleh pengembang menggunakan bahasa pemrograman server-side. Ini memberi Anda akses ke program buku dan pengguna informasi yang tersimpan pada Penelusuran Buku. Dengan ini, Anda dapat menyediakan integrasi lebih dalam dengan Buku Pencarian dalam berbagai cara, seperti:

1. Personalisasi situs atau aplikasi dengan informasi yang ada pada pengguna.
2. Membuat sebuah aplikasi web yang menunjukkan hasil pencarian teks lengkap dari Book Search.
3. Membuat aplikasi desktop memuat fasilitas pencarian buku untuk platform baru.

Untuk menggunakan data API dengan bahasa pemograman PHP, dapat menggunakan php *client* yang telah di sediakan oleh zend frameworks yang telah dirilis khusus untuk menangani google api. Untuk mendowload php client ini dapat di akses

http://framework.zend.com/download/gdata.

#### 2.12.1.1. Penggunaan *Google Books Search API*

Penggunaan PHP *client library* memerlukan PHP versi 5.14 keatas dan Zend_Gdata versi 1.7 keatas yang didistribusikan sebagai bagian dari Zend Framework. Lihat Panduan Persiapan untuk informasi lebih lanjut tentang konfigurasi lingkunganAnda.Potongan kode contoh di bawah ini dapat disalin / disisipkan ke dalam kode Anda dan dimodifikasi sesuai dengan kebutuhan anda. Ada juga contoh aplikasi dua didistribusikan dengan perpustakaan klien yang meliputi banyak operasi API. Kode ini terletak di /demo/MyLibrary dan /demo/BooksBrowser. Sebelum Anda dapat melakukan operasi apapun dengan Buku Pencarian Data API, Anda harus menginisialisasi Zend_Gdata_Books objek seperti dibawah ini.

```
<?php
require_once 'Zend/Loader.php';
Zend_Loader::loadClass('Zend_Gdata_Books');
$books = new Zend_Gdata_Books();
```

Untuk pencarian *book search data API* menyediakan sejumlah *feed dari* daftar koleksi buku.

Tindakan yang paling umum adalah untuk mengambil daftar buku-buku yang cocok dengan permintaan pencarian. Untuk melakukannya Anda membuat objek VolumeQuery dan melewatkannya ke methode Books::getVolumeFeed.

Misalnya, untuk melakukan *Query* kata kunci, dengan filter pada *viewability* untuk membatasi hasil untuk melihat buku penuh atau parsial, gunakan setMinViewability dan metode setQuery dari objek VolumeQuery. Potongan kode berikut cetak judul dan viewability dari semua volume metadata atau teks yang cocok dengan istilah permintaan "PHP", sebagai contoh penggunaan Data API seperti dibawah ini.

```
$books = new Zend_Gdata_Books();
  $query = $books->newVolumeQuery();
  $query->setQuery('PHP');
  $query->setMinViewability('partial_view');
  $feed = $books->getVolumeFeed($query);

  foreach ($feed as $entry) {
    echo $entry->getVolumeId();
    echo $entry->getTitle();
    echo $entry->getViewability();
  }
```

Kelas Query, dan subclass seperti VolumeQuery, bertanggung jawab untuk membangun URL feed. VolumeQuery yang ditunjukkan sama dengan URL sebagai berikut.

```
http://www.google.com/books/feeds/volumes?q=keyword&min-viewability=partial
```

Berikut ini adalah beberapa metode VolumeQuery paling umum untuk pengaturan parameter pencarian:

1. Method setQuery

Digunakan untuk menentukan istilah permintaan pencarian. Pencarian Buku pencarian semua metadata buku dan teks penuh untuk buku-buku yang cocok istilah. Buku metadata termasuk judul, kata kunci, deskripsi, nama penulis, dan subyek. Perhatikan bahwa spasi, tanda kutip atau tanda baca lain di nilai parameter harus URL-escape. (Gunakan plus (+) untuk spasi.). Untuk mencari sebuah frase eksak, menyertakan frasa dalam tanda kutip. Sebagai contoh, untuk mencari buku pencocokan frase "pesawat mata-mata", set parameter q untuk% 22spy + pesawat% 22.Anda juga dapat menggunakan salah satu operator pencarian lanjutan didukung oleh Pencarian Buku. Sebagai contoh, Jane Austen + +-inauthor: Austen hasil pertandingan yang menyebutkan (tetapi tidak ditulis oleh) Jane Austen.

2. Methode setMinViewability

Memungkinkan Anda untuk menyaring hasil menurut status viewability buku '. Parameter ini menerima salah satu dari tiga nilai: 'none' (default, kembali semua buku yang sesuai tanpa viewability), 'partial_view' (kembali buku hanya bahwa pengguna dapat melihat atau melihat secara keseluruhan), atau 'full_view' (kembali hanya buku bahwa pengguna dapat melihat secara keseluruhan).