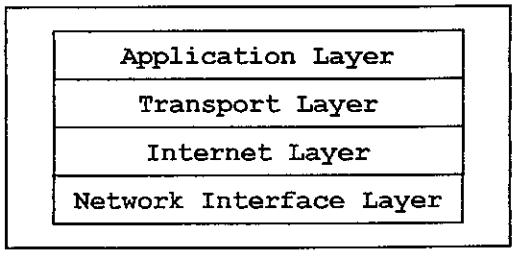
BAB II

TEORI PENUNJANG

2.1. INTERNET

Internet adalah jaringan global dari jaringan-jaringan komputer di seluruh dunia yang dihubungkan dengan satu protokol yaitu TCP/IP (*Transmission Control Protocol/Internet Protocol*). Hal ini karena suatu komputer dikatakan berada di Internet apabila komputer tersebut mengoperasikan TCP/IP, mempunyai alamat IP dan memiliki kemampuan untuk mengirim paket IP ke komputer lainnya yang ada di Internet (Tanenbaum, 1996).

TCP/IP terdiri dari empat lapisan yaitu *Network Interface Layer*, *Transport Layer*, *Internet Layer* dan *Application Layer*. Setiap lapisan dapat terdiri dari berbagai macam protokol. Dengan memecah TCP/IP menjadi beberapa lapisan maka setiap lapisan dapat dikembangkan secara terpisah dari lapisan lainnya. Namun demikian komunikasi antar lapisan harus tetap dijaga meskipun protokol pada suatu lapisan berubah. Berikut gambaran umum lapisan-lapisan TCP/IP (Purbo, et. al., 1998) :



Gambar 1 : Lapisan-Lapisan TCP/IP

Pada lapisan paling atas yaitu *Application Layer* terjadi interaksi dengan pengguna karena lapisan inilah yang dihadapi pengguna. Lapisan ini memiliki beberapa aplikasi di Internet yang banyak digunakan yaitu :

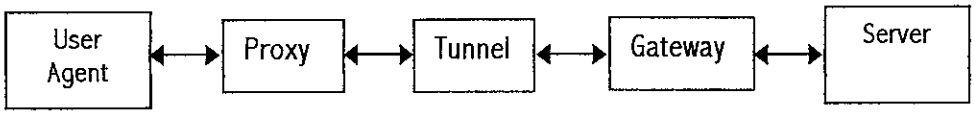
- *World Wide Web* (WWW) atau biasa disebut Web dengan protokol transfer adalah HTTP (*Hypertext Transfer Protocol*).
- *Electronic Mail* (E-Mail) dengan protokol transfer SMTP (*Simple Mail Transfer Protocol*).
- Transfer file dengan protokol FTP (*File Transfer Protocol*).

2.1.1. HTTP (Hypertext Transfer Protocol)

HTTP adalah protokol pada lapisan aplikasi untuk sistem informasi bagi data-data berupa gambar, suara, video dan data multimedia yang seringkali dalam jumlah besar yang terdistribusi dan saling berhubungan (Fielding, et.al.:1999).

HTTP dimulai dengan HTTP/0.9 kemudian berkembang hingga HTTP/1.1 yang digunakan sekarang. Pada HTTP/0.9, protokol ini hanya berfungsi sebagai alat transfer data melalui Internet. Pada HTTP/1.0 *message* telah dapat dikirim dalam format MIME-*like* (*Multipurpose Internet Mail Extensions)-like*, yang berisi metainformasi tentang data yang ditransfer dan telah memiliki kemampuan modifikasi untuk semantik *request/respons*. Terakhir HTTP/1.1 telah memungkinkan suatu *persistent connections* dan telah mendukung absolut URL (*Universal Resource Locator*) *request* (Purbo, et. al., 1998).

HTTP merupakan protokol *request/response*. *Client* mengirim *request* (permintaan) kepada *server* dan *server* memberikan *response* (tanggapan) sesuai yang diminta *client*. Setiap kali *client* melakukan *request* atau *server* melakukan *response* maka proses tersebut terdiri dari tiga bagian, yaitu baris *request/response*, bagian *header* dan *entity body*. Berikut contoh model hubungan HTTP (Purbo, et. al., 1998) :



Gambar 2 : Contoh Hubungan HTTP

A. Client – Request

Mekanisme *client* untuk melakukan transaksi dengan *server* HTTP adalah (Purbo, et. al., 1998):

- *Client* terhubung dengan *server* HTTP secara default pada port 80 dari TCP/IP. *Client* melakukan *request* dengan cara mengirim suatu perintah HTTP yang disebut *method* yang diikuti suatu alamat dokumen dan nomor versi protokol HTTP yang digunakan.
- *Client* mengirim informasi *header* yang opsional kepada *server* mengenai konfigurasi dan format dokumen yang akan diterima. Informasi *header* tersebut dikirim secara baris per baris.
- *Client* mengirim data tambahan yang disebut *request entity*. Data tambahan ini biasanya dipakai dalam pemrograman CGI yang memakai *method* POST.
- *Client* mengirim baris kosong untuk mengakhiri suatu *request*.

Sebuah contoh *client-request*:

```
GET /index.html HTTP/1.0
Accept: */*
Connection: Keep Alive
Host: www.w3c.org
User-Agent: Generic
```

B. Server – Response

Mekanisme *server* untuk mengirim *response* kepada *client* adalah (Purbo, et. al., 1998) :

- *Server* memberi *response* dengan cara mengirim baris status yang berisi versi HTTP, kode status dan deskripsi kode status.
- *Server* mengirim informasi *header* tentang konfigurasi *server* dan dokumen yang diminta *client*. Setiap *header* dikirim secara per baris dan masing-masing dengan nama *header* beserta nilainya.
- Mengirim data yang diminta *client* bila *request* tersebut berhasil. Data yang dikirimkan tersebut berbentuk dokumen atau hasil program CGI. Hasil ini biasa disebut *response entity*. Bila *request* gagal maka *server* mengirim pesan berisi sebab-sebab *server* tidak memenuhi *client-request*.
- Mengirim baris kosong untuk mengakhiri suatu *response*.

Sebuah contoh *server-response* :

```
HTTP/1.1 200 OK
Date: Wed, 05 May 2000 19:20:32 GMT
Server: Apache/1.3.6 (Unix) PHP/3.0.7
Last-Modified: Mon, 17 February 2000 13:33:33 GMT
Etag: "2da0dc-2870-374039cd"
Accept-Ranges: bytes
Content-Length: 10352
Connection: close
Content Type: text/html; charset=iso-8859-1
<! DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
http://www.w3c.org/TR/Rec-html40/loose.dtd>
<html><body><p>Belajar XML</p></body></html>
```

2.1.2. World Wide Web

World Wide Web atau yang biasa disebut Web merupakan suatu kerangka arsitektur untuk mengakses dokumen-dokumen yang tersebar di ribuan komputer yang ada di Internet (Tanenbaum, 1996). Dalam menyajikan suatu tampilan di Web melibatkan data-data berupa gambar, suara, video dan data multimedia yang seringkali dalam jumlah besar sehingga disebut *hypermedia.* Kemampuan ini ditunjang adanya sarana *hypertext* dan *hyperlink. Hypertext* adalah mekanisme yang digunakan pada halaman Web untuk merujuk ke halaman Web lainnya sedangkan *Hyperlink* adalah teks yang diinginkan pengguna yang digunakan untuk merujuk ke dokumen lain. Media yang digunakan untuk menyajikan suatu dokumen ke dalam Web adalah HTML (*Hypertext Markup language*) dan protokolnya adalah HTTP (*Hypertext Transfer Protocol*).

Awalnya tampilan Web berupa halaman-halaman statis berisi teks, gambar atau data multimedia. Halaman-halaman tersebut disebut statis karena pengguna tidak bisa berinteraksi dengan sistem yang dihadapinya. Kemudian muncul tampilan Web yang dinamis karena adanya pemrograman dengan CGI (*Common Gateway Interface*) pada sisi *server* dan adanya pemrograman *scripting* pada sisi pengguna. Dengan adanya Web dinamis maka dapat terjadi interaksi antara pengguna dengan sistem. Hal ini kemudian dimanfaatkan oleh dunia bisnis untuk membuka toko on-line di Web sebagai sarana transaksi jual beli on-line.

Untuk melihat suatu halaman Web, pengguna harus menggunakan *browser. Browser* mengambil halaman yang diminta pengguna dengan cara *request* ke *server* kemudian menerjemahkan teks serta tampilan data dan menampilkannya

ke pengguna. *Browser* tidak bisa berdiri sendiri karena sering kali dibutuhkan perangkat lunak lain seperti *Acrobat Reader* untuk file-file PDF (*Portable Document Format*) atau *Windows Media Player* untuk data-data video dan suara.

Untuk dapat melihat suatu tampilan di Web pengguna harus memasukkan alamat URL yang diinginkan ke dalam *browser*. URL adalah format penamaan di Internet untuk menunjuk suatu *resource* pada suatu lokasi tertentu. Suatu URL terdiri dari protokol yang dipakai, nama *server* dan lokasi *resource* di *server*. Di HTTP, aturan URL yang digunakan adalah :

```
http_URL = "http:" "//" host [ ":" port ] [abs_path]
```

Keterangan :

- *host* menunjukkan nama *host* yang legal atau nomor IP dari *host* tersebut.
- *port* adalah bilangan yang menunjukkan port HTTP di *host*. Jika kosong maka yang digunakan adalah *port* nomor 80.
- *Abs_path* menunjukkan lokasi *resource* dalam *host* tersebut.

Contoh URL : `http://www.w3c.org/TR/REC-xml`

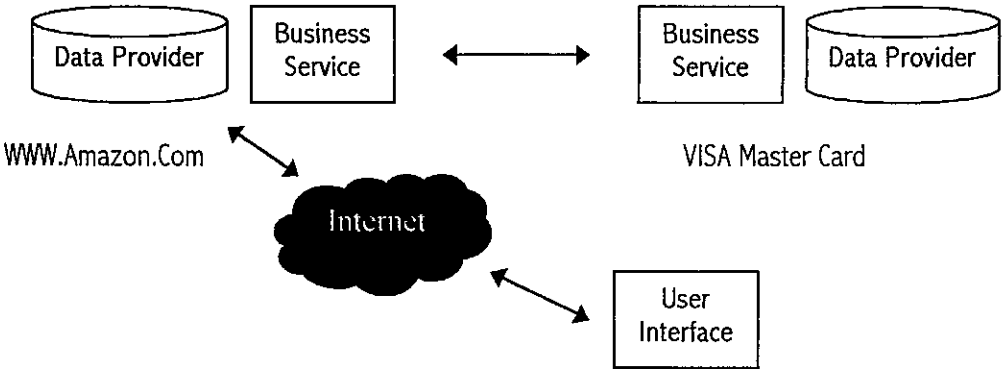
2.1.3. Aplikasi Berbasis Web

Proses untuk aplikasi berbasis Web biasanya lebih banyak dilakukan di *server*. Hasil dari proses itulah yang kemudian dikirim ke *client* melalui HTTP dalam bentuk halaman Web. Selain *Webserver* dan *browser*, aplikasi berbasis Web juga sering melibatkan basis data. Aplikasi berbasis Web semacam ini dapat disebut sebagai aplikasi *3-tier* karena memiliki tiga komponen yaitu *user interface*, *business service* dan *data provider* (Hadiwinata, 2003).

Ketiga komponen tersebut memiliki fungsi yang saling mendukung dimana *user interface* berfungsi sebagai bagian yang berinteraksi dengan pemakai, *business service* berfungsi mengontrol semua data yang diakses dari bagian penyimpanan data dan *data provider* sebagai tempat untuk proses manipulasi data serta bertanggung jawab terhadap penyimpanan dan penyediaan data. Aplikasi *3-tier* digambarkan sebagai berikut (Hadiwinata, 2003) :

Gambar 3 : Model Aplikasi 3-Tier

Aplikasi *3-tier* ini bisa berkembang menjadi *N-tier* bila melibatkan banyak *tier* dalam satu sistem seperti yang dapat dijumpai di sistem pemesanan buku Amazon.com. Di sana pelanggan bisa melihat daftar katalog buku pada *database* Amazon.com dan melakukan proses pembayaran melalui kartu kredit dimana masing-masing bagian dapat terdiri dari aplikasi *3-tier* yang saling terpisah dan melakukan prosesnya sendiri sebagaimana yang dapat dilihat pada contoh model aplikasi *N-tier* berikut (Hadiwinata, 2003) :



Gambar 4 : Contoh Model Aplikasi N-Tier

2.1.3.1. SGML (Standard Generalized Markup Language)

SGML dikembangkan dari GML (*Generalized Markup Language*) oleh Charles Goldfarb. SGML adalah suatu standar internasional untuk mendefinisikan metode yang bersifat *device-independent* dan *system-independent* untuk memformat teks ke dalam dokumen elektronik. SGML menyediakan perangkat yang memungkinkan membuat aturan dokumen yang spesifik sesuai *domain*-nya.

Akhir 1960-an peneliti IBM, Charles Goldfarb, Ed Mosher dan Ray Lorie mulai bekerja dengan problem utama yaitu kurangnya portabilitas antar dokumen karena adanya ribuan dokumen yang terdapat pada sistem-sistem yang terpisah yang saling berbeda antara satu dengan lainnya. Dan berdasarkan kesulitan yang mereka hadapi kemudian mereka mengembangkan GML untuk mengatasi permasalahan dalam pengolahan dokumen berjumlah banyak dengan format yang berbeda-beda. Usaha mereka tersebut memberikan pemecahan yang dibutuhkan untuk menjaga interoperabilitas antar sistem, yaitu (Navaro, et. al., 2000) :

- Pengolah dokumen harus mendukung format dokumen yang umum.
- Format dokumen tersebut harus bersifat spesifik pada *domain* dokumen.
- Agar reliabilitasnya terjaga, format dokumen harus mengikuti aturan-aturan yang spesifik pula.

SGML menyediakan sarana untuk mendeskripsikan dokumen dan untuk membuat ketentuan spesifik yang sesuai untuk dokumen baru tersebut atau dapat dikatakan SGML menyediakan aturan untuk membuat suatu aturan dokumen yang spesifik sesuai karakteristik dari dokumen tersebut. SGML digambarkan sebagai berikut (Navaro, et. al. 2000) :

- SGML tidak menyediakan struktur dokumen yang spesifik.
- SGML tidak membatasi himpunan *tag* yang harus digunakan.
- SGML tidak mengharuskan adanya pembuatan standar dokumen yang baru.

Markup language (bahasa penanda) adalah suatu himpunan aturan yang digunakan secara bersama untuk melakukan penandaan (*encode*) suatu teks. Suatu bahasa penanda harus mendefinisikan tanda (*markup*) yang diperbolehkan, tanda yang diperlukan, mekanisme perbedaan antara tanda dengan teksnya dan arti dari tanda tersebut dan SGML telah menyediakan semua persyaratan tersebut. SGML juga memberikan kebebasan dalam memberi spesifikasi suatu aturan dokumen. Setiap dokumen SGML harus memiliki DTD (*document type definition*) yang berfungsi memberi definisi elemen dan atribut yang digunakan dokumen tersebut.

SGML mendefinisikan data dalam strukturnya dan bukan dalam mekanisme tampilannya. Fokus utama SGML pada struktur informasi dan tidak menyediakan satu aturan yang spesifik, sebaliknya SGML memberi kebebasan dalam mengatur data. Dengan mengikat data hanya pada strukturnya dan bukan pada tampilannya, data tersebut dapat lebih cepat untuk diproses lebih lanjut atau dibawa ke berbagai bentuk yang diinginkan untuk dikirim ke hampir semua media seperti dalam bentuk cetak, tampilan, suara atau yang lainnya. Aturan-aturan SGML terdokumentasi dalam *International Standard ISO 8879 Information Processing – Text and Office Systems – Standard Generalized Markup Language (SGML).*

2.1.3.2. HTML (Hypertext Markup Language)

HTML adalah format dokumen yang digunakan di Web. Dokumen HTML yang diletakkan di *Webserver* akan ditansfer ke *client* melalui protokol HTTP. HTML ditampilkan oleh *browser* ke dalam halaman Web yang berisi teks, gambar atau data multimedia. HTML memberikan definisi bagaimana data-data tersebut ditandai dengan memakai *tag* yang merupakan tanda dari tampilan data tersebut.

HTML dikembangkan berdasarkan standar aturan SGML dan dirancang dalam bentuk yang sederhana agar pengguna yang tidak memiliki pemahaman tentang SGML dapat memakainya. Karena perkembangan Web sangat pesat maka penggunaan HTML juga secara langsung ataupun tidak juga tinggi. Hal ini dikarenakan kemampuan Web untuk mendistribusikan dokumen elektronik secara mudah dan relatif murah kepada masyarakat (Bosak, 1997).

Ketika penggunaan dokumen Web meningkat pada pengolahan data pada pihak *client* maka timbul berbagai kesulitan karena data pada dokumen Web tersebut tidak bisa diolah secara elektronik. Terdapat beberapa keterbatasan HTML untuk mengolah data dari dokumen Web *client* dalam jumlah besar terutama untuk data-data perdagangan dan bisnis berskala besar. Menurut Jon Bosak hal-hal yang menjadi keterbatasan HTML adalah (Bosak, 1997) :

- HTML tidak memungkinkan pengguna membuat *tag* atau atributnya sendiri untuk memberi paramater atau memberi kualifikasi data secara semantik.
- HTML tidak mendukung spesifikasi dari struktur yang dibutuhkan untuk menggambarkan skema basis data atau hirarki berorientasi obyek.

- Spesifikasi HTML tidak memungkinkan aplikasi yang menggunakannya untuk memeriksa data berdasarkan validitas struktur jika dibutuhkan.

Kendala yang sering dihadapi saat pengguna ingin mengolah dokumen Web yang ditemuinya adalah data yang ditandai dengan *tag* HTML. Sebagai contoh, bagi manusia mungkin bisa dengan mudah membedakan angka "10" sebagai jumlah dan sebagai kode barang. Namun hal tersebut tidak berlaku bagi program komputer, karena komputer membutuhkan suatu penanda untuk membedakan antara angka "10" sebagai jumlah atau kode barang. Jika program komputer melihat *tag* yang menandai data tersebut dalam bentuk *tag* HTML maka program komputer membaca angka "10" tersebut sebagai angka biasa. Hal ini karena tidak ada *tag* yang memberi tahu komputer bahwa angka "10" yang dibacanya menyatakan jumlah barang dan bukan kode barang. Untuk mengatasinya dibutuhkan suatu penanda yang bisa menjelaskan data tersebut secara jelas.

Untuk itu W3C (*World Wide* Web *Consortium*) mengembangkan XML (*Extensible Markup Language*) sebagai salah satu subset SGML. Tujuannya agar generik SGML dapat dilayani, diterima dan diproses di Web seperti yang dapat dilakukan oleh HTML. XML dirancang untuk penerapan yang mudah dan interoperabilitas baik dengan SGML atau dengan HTML (Navaro, et. al. 2000).

2.2. PERTUKARAN DATA

Sekarang sudah memungkinkan terjadinya transfer data sebagai mekanisme pertukaran data secara elektronik tanpa kertas. Dalam dunia bisnis dikenal adanya E-Commerce yang secara prinsip berfungsi menggantikan aliran data yang semula berbasis kertas menjadi tanpa kertas sehingga aliran data menjadi jauh lebih efisien. Proses-proses dalam E-Commerce merupakan suatu sistem yang meliputi aliran data masukan, pengolahan dan keluaran dari suatu transaksi bisnis yang bisa melibatkan berbagai komunitas mulai dari dalam negeri sampai internasional. E-Commerce memiliki cakupan luas yang meliputi proses, teknologi dan praktek dalam melakukan transaksi bisnis tanpa menggunakan kertas sebagai mekanismenya. Konsep E-Commerce tidak hanya dipakai di bidang bisnis namun juga digunakan pada bidang lain seperti pada E-Government dan E-Academic. Hal ini karena konsep E-Commerce menyediakan dua hal pokok yang telah teruji handal yaitu adanya faktor kemudahan dan kecepatan (Purbo, et. al., 2001).

2.2.1. Electronic Data Interchange (EDI)

EDI (*Electronic Data Interchange*) adalah pertukaran informasi bisnis antara satu komputer dengan komputer lainnya berdasarkan aturan-aturan transaksi yang telah dijadikan standar dan dikembangkan pada bidang-bidang khusus. Sejak awal EDI ditujukan untuk proses B2B (*Business to Business*), namun banyak proses B2C (*Business to Consumer*) yang bisa menggunakannya untuk memenuhi kebutuhan konsumen. Format EDI sudah distandarkan dan memiliki dua standarisasi utama, yaitu :

- Standar X12, dikembangkan oleh (*Accredited Standards Committee*) ASC X12 yang dibantu oleh ANSI (*American National Standards Institute*). Standar ini umumnya digunakan negara-negara di kawasan Amerika Utara.
- Standar EDIFACT (*EDI For Administration, Commerce And Transport*) yang dikembangkan oleh *United Nations Economic Commission* di Eropa.

Selain standar-standar yang digunakan tadi, masih banyak lagi standar-standar lain yang bersifat pribadi (*proprietary*) (Purbo, et. al., 2001). Walaupun terdapat beragam variasi dalam EDI tetapi dua hal yang pokok adalah komunikasi antar komputer dan aturan-aturan transaksi.

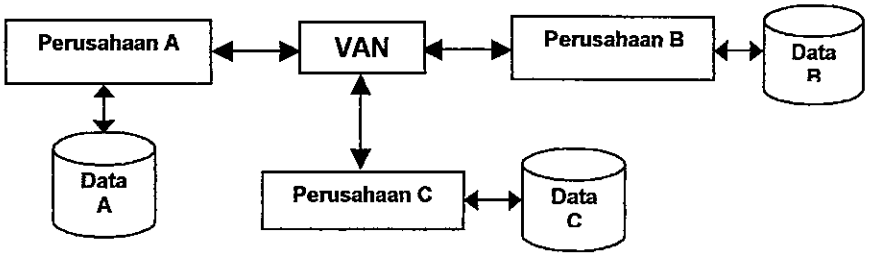
2.2.1.1. Komunikasi Antar Komputer

Komunikasi antar komputer dilakukan dalam bentuk jaringan komputer dan pertumbuhan jaringan komputer yang begitu cepat dan luas mengakibatkan protokol komunikasi data juga ikut berkembang secara pesat. Protokol adalah kumpulan aturan yang mengatur komunikasi antar komputer.

EDI bekerja pada tingkatan aplikasi sehingga berbagai protokol yang ada dapat digunakan dalam berbagai proses penerapan EDI. EDI tidak menangani proses penyampaian dan transfer data antar komputer dari perusahaan yang menjadi pemakai layanannya. EDI hanya mengatur proses pembentukan data yang akan dikirim dan yang diterima sesuai format data yang diinginkan oleh masing-masing pihak.

Setiap pelaku bisnis memiliki format data yang berbeda sehingga saat antar pelaku bisnis melakukan pertukaran informasi timbul masalah yang disebabkan

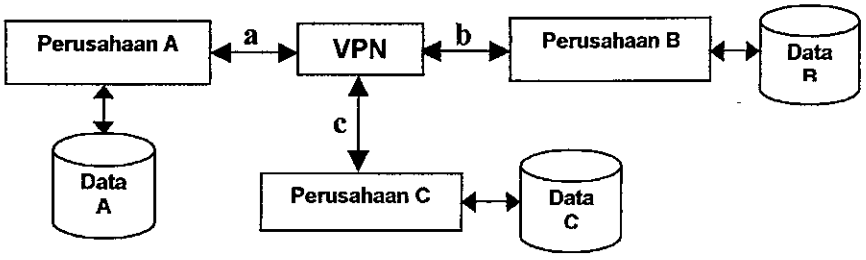
format data yang berbeda tadi. Pada proses ini EDI berfungsi sebagai penengah, EDI berfungsi merubah data yang dikirim pihak pertama sehingga sesuai dengan format data pihak kedua. Dengan demikian pihak pertama dan kedua tidak perlu saling mengetahui format data masing-masing sehingga keamanan data dari kedua belah pihak bisa lebih terjaga. Berikut gambaran mekanisme EDI :



Gambar 5 : Mekanisme EDI Menggunakan VAN

Dalam EDI proses pengubahan data diserahkan pada *Value Added Network* (VAN) sebagai perusahaan penyedia jasa layanan telekomunikasi yang memiliki jaringan telekomunikasi cukup luas dan layanan dari VAN disebut *Value Added Services* (VASs). Pada gambar 4, VAN memberi layanan pada perusahaan A, B dan C dan layanan dari VAN disebut sebagai VASs.

Selain melalui VAN, EDI dapat dilakukan melalui Internet dengan teknologi *tunneling*. Dengan teknologi ini dapat dibangun suatu koneksi tanpa terganggu oleh pemakai Internet lainnya. Koneksi semacam ini disebut *Virtual Private Network* (VPN). Berikut gambarannya :



Gambar 6 : Mekanisme EDI Melalui Internet

Pada gambar 6 ditunjukkan proses EDI melalui Internet. VPN adalah jaringan dimana koneksi antar perusahaan A, B dan C yang dilakukan dengan bantuan teknologi *tunneling* melalui media Internet, sedangkan A, B dan C adalah perusahaan yang melakukan koneksi melalui koneksi a, b dan c.

2.2.1.2. Aturan-Aturan Transaksi

Aturan-aturan transaksi merupakan persetujuan bersama antar pelaku bisnis. Persetujuan bersama ini memiliki peranan yang sangat penting karena akan menentukan bentuk transaksi yang akan dilakukan antar pelaku bisnis. Pada sistem EDI persetujuan tersebut diterapkan antara penyedia layanan VAN dengan penyedia komunikasi data lainnya dan dengan para pelaku bisnis sesuai dengan yang diinginkan (Purbo, et. al., 2001).

Persetujuan bersama tersebut secara umum menggambarkan layanan data yang disediakan, kewajiban pelaku bisnis dan jaminan klaim serta pertanggungjawaban. Persetujuan juga berisi informasi teknis tentang pemakaian, pengakuan dan pemrosesan format pesan yang dikirim.

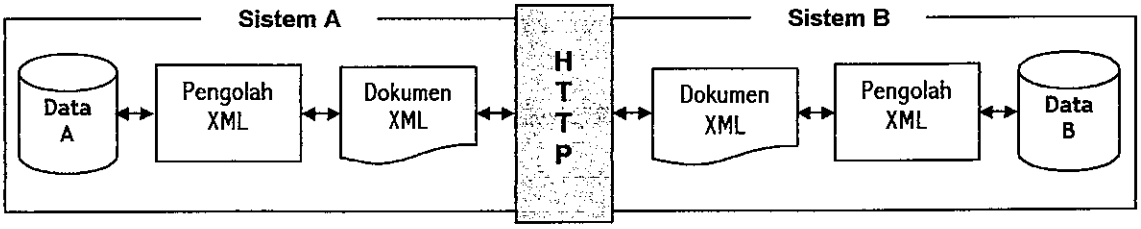
2.2.2. Pertukaran Data Di Internet

Pada umumnya proses B2B (*Business to Business Commerce*) menggunakan mekanisme EDI (*Electronic Data Interchange.* Namun orang tetap saja menginginkan suatu sistem yang lebih sederhana dan lebih murah namun tetap mengadopsi kelebihan-kelebihan EDI. Hal ini karena sistem yang menggunakan

EDI relatif masih sangat mahal dan standar yang digunakan seringkali menyulitkan interkomunikasi antar pelaku bisnis (Purbo, et. al., 2001).

Berbagai pihak mencoba pendekatan lain melalui pengembangan teknologi Web dan Internet. Hal ini karena kedua teknologi tersebut telah terbukti handal dan memberikan efisiensi yang sangat tinggi. Alternatif yang dikembangkan dan sekarang cukup populer dalam bidang standarisasi pengiriman data dengan menggunakan XML (*Extensible Markup Language*) yang dikembangkan oleh W3C (*World Wide Web Consortium*). Perkembangan Internet yang semakin luas mengharuskan adanya format standar baru yang bersifat global dalam pengiriman data menjadi penyebab dari berkembangnya sistem pertukaran data dengan menggunakan XML (Purbo, et. al., 2001).

Untuk melakukan transfer data di Internet digunakan protokol HTTP (*Hypertext Transfer Protocol*). HTTP merupakan salah satu subprotokol dari TCP/IP. Hal ini karena dalam pertukaran data ada dua hal yang menjadi kendala utama yaitu format data yang berbeda-beda dan platform komputer yang berbeda-beda. Dengan demikian pertukaran data yang dilakukan harus mampu mengatasi kedua kendala tersebut. Di sini XML berperan sebagai penyusun format data yang akan dipertukarkan antar sistem sedang HTTP berperan mengatasi platform komputer yang berbeda-beda tersebut. Berikut gambaran sederhananya :



Gambar 7 : Gambaran Umum Pertukaran Data Di Internet

Di dalam gambar 6 di atas ditunjukkan pertukaran data melalui HTTP. Data A dan data B adalah data asal yang akan saling dipertukarkan. Data A dan data B oleh pengolah XML diolah menjadi dokumen XML yang siap untuk ditransfer melalui HTTP. Proses transfer melalui HTTP dilakukan dengan menggunakan fasilitas *World Wide Web* yang telah tersedia di Internet.

2.3. TRANSAKSI SECARA ON-LINE

Saat ini tidak ada definisi pasti dari E-Commerce yang sudah distandarkan dan disepakati bersama. Namun secara umum E-Commerce mempunyai cakupan luas yang berkaitan dengan berbagai bidang sehingga E-Commerce bisa diartikan sebagai suatu kumpulan dinamis yang berisi teknologi, aplikasi dan proses bisnis yang menghubungkan perusahaan, konsumen dan komunitas tertentu melalui transaksi secara on-line dan perdagangan barang, pelayanan dan informasi yang dilakukan secara on-line pula (Purbo, et. al., 2001).

Salah satu cakupan E-Commerce adalah transaksi secara on-line. Hal ini karena transaksi secara on-line adalah bagian dari E-Commerce yang melakukan transaksi jual beli secara on-line dengan memanfaatkan fasilitas-fasilitas yang tersedia di Internet.

2.3.1. Kegiatan Yang Berhubungan Dengan E-Commerce

Sebelum adanya E-Commerce, transaksi dilakukan dari tangan ke tangan secara langsung, antara pembeli dan penjual bertatap muka, melakukan kesepakatan dan akhirnya terjadi kesepakatan. Namun setelah ada E-Commerce

kini semua bisa dilakukan dengan mudah tanpa harus memperhitungkan adanya keterbatasan sarana, jarak dan waktu transaksi. Bidang-bidang yang dilakukan dalam E-Commerce, antara lain (Purbo, et. al., 2001) :

- Perdagangan on-line melalui *World Wide Web*.
- Transaksi bisnis on-line.
- Internet banking.
- TV interaktif.
- WAP (*Wireless Application Protocol*).

2.3.2. Komponen-Komponen E-Commerce

2.3.2.1. E-Commerce Trust

Konsep *trust* (kepercayaan) ini penting sekali karena berpengaruh besar terhadap transaksi on-line termasuk di dalamnya adalah faktor *security* (keamanan) dan *privacy* (kerahasiaan). Jika tidak ada faktor *trust* yang benar-benar disetujui oleh para pelaku bisnis maka E-Commerce tidak akan berkembang untuk mencapai hasil yang diinginkan. Komponen-komponen yang menjadi pembentuk E-Commerce *trust* adalah (Purbo, et. al., 2001) :

- *Seal of approval* (simbol pengesahan).
- *Brand* (legalitas badan hukum terhadap produk atau jasa).
- *Navigation* (pencarian yang mudah).
- *Fulfillment* (kemudahan proses dan transaksi bisnis).
- *Presentation* (desain Web yang menarik dan informatif).
- *Technology* (pemanfaatan teknologi yang handal).

2.3.2.2. Kelas-Kelas Yang Meliputi Sektor B2B

E-Commerce memiliki dua bentuk yaitu B2B dan B2C. Proses pada B2C biasanya dilakukan melalui portal atau toko on-line yang berfungsi seperti halnya sebuah toko. Proses di dalam B2B antara lain (Purbo, et. al., 2001):

- *Sell-Side*. Fokusnya terletak pada penjualan produk atau jasa layanan ke pembeli atau konsumen. Pada sistem ini biasanya pembeli atau konsumen berada di luar lingkup perusahaan itu.
- *Buy-Side*. Pada sistem ini perusahaan membangun aplikasi dimana anggota internal bisa menggunakan sistem untuk melakukan sebuah usaha. Aplikasi menyediakan pusat kontrol usaha, membantu perusahaan untuk mengatur sistem pembelanjaan dan penerimaan, negosiasi harga yang kompetitif serta penghematan uang dan waktu. Aplikasi juga dapat menyediakan daftar produk secara umum dan memberikan layanan dimana pemakai dapat memilih sendiri apa yang diinginkan.
- *Marketplace* (*Buyers and Sellers*). Sistem ini relatif baru dan masih sedikit yang menerapkannya. *Marketplace* menciptakan sebuah komunitas virtual dimana pembeli dan penjual dalam jumlah besar berkumpul menjadi satu. Perangkat yang tersedia pada sistem ini memungkinkan semua yang terlibat dapat melakukan transaksi antara satu dengan lainnya.