

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

*Telephony* adalah teknologi yang berhubungan dengan transmisi elektronik suara yang disampaikan ke tempat yang berjarak jauh melalui telepon, sebuah perangkat yang terdiri atas *speaker*, *transmitter* dan *receiver*. Hadirnya komputer dan perangkat transmisi digital yang berbasis sistem telepon serta penggunaan radio untuk mengirim dan menerima sinyal telepon, maka perbedaan antara *telephony* dan telekomunikasi semakin sulit ditemukan.

*Voice over Internet Protocol* dikenal juga dengan sebutan *IP Telephony*. Secara umum, VoIP didefinisikan sebagai suatu sistem yang menggunakan jaringan internet untuk mengirimkan data paket suara dari suatu tempat ke tempat yang lain menggunakan perantara *protocol* IP. Pada kenyataannya, VoIP lebih terfokus pada penggunaan internet jika dibandingkan dengan telepon tradisional yang infrastrukturnya dibangun lebih awal.

Perusahaan jasa yang mengoperasikan VoIP disebut *Internet Telephony Service Provider (ITSP)*, salah satu sistem penting pada perusahaan jasa ITSP adalah sistem *billing*. Sistem *billing* memiliki peranan penting karena berhubungan dengan kelangsungan proses bisnis perusahaan, terutama dalam hal administrasi yang berinteraksi langsung dengan pelanggan sehingga akan sangat mempengaruhi kualitas pelayanan administrasi.

Pada saat ini, fungsi sistem *billing* yang diterapkan operator telekomunikasi meningkat seiring dengan perkembangan jaringan dan layanan-layanan baru yang diterapkan oleh perusahaan jasa telekomunikasi. Secara umum komponen sistem *billing* terdiri dari empat komponen utama yaitu *mediation*, *rating tariff*, proses *billing / invoicing*, dan *bill*.

PT Ejazz Indonesia Baru yang berdiri sejak tahun 2000, merupakan salah satu perusahaan yang bergerak pada jasa telekomunikasi dengan menggunakan

teknologi *Voice over Internet Provider (VoIP)*, layanan yang disediakan oleh PT Ejazz Indonesia Baru, hanya terminasi dari perusahaan lain.

Pelayanan administrasi kepada pelanggan pada PT Ejazz Indonesia Baru sangat diutamakan karena keakuratan dan ketepatan waktu penagihan serta pembayaran akan berpengaruh kepada kelangsungan proses bisnis perusahaan. Sistem pelayanan administrasi khususnya pada saat penagihan kepada pelanggan yang ada pada saat ini kurang dapat memberikan pelayanan yang maksimal karena masih adanya keterlambatan dalam proses pengumpulan data pelanggan mengenai jumlah waktu pemakaian, tarif pemakaian per-menit, dan total biaya pemakaian keseluruhan yang terangkum dalam lembar tagihan (*invoice*).

Waktu pemakaian (*Call Data Recording*) VoIP dicatat oleh *Radius server* yang nantinya akan di-*generate* oleh program *Radbconv* untuk dimasukkan ke dalam *database server*. Setelah proses tersebut selesai, jumlah waktu pemakaian VoIP dapat dilihat dengan menggunakan *Microsoft SQL 2000 Query Analyzer*, setelah itu dipindahkan satu persatu data tersebut ke dalam laporan per-hari jumlah waktu pemakaian VoIP. Setelah laporan jumlah pemakaian VoIP per-hari diperoleh, maka bagian administrasi PT Ejazz Indonesia Baru akan membuat tagihan kepada pelanggan pada akhir bulan.

Berdasarkan masalah di atas penulis bermaksud untuk membangun Sistem Informasi berbasis web dengan judul “*Billing Voice over Internet Protocol (VoIP) Berbasis Web*”. Diharapkan dengan adanya sistem ini dapat menunjang pelayanan administrasi dalam menjaga dan meningkatkan kepuasan pelanggan.

## 1.2 Perumusan Masalah

Permasalahan yang ada dalam PT Ejazz Indonesia Baru berdasarkan latar belakang masalah di atas adalah sebagai berikut :

1. Sistem *Billing* yang ada belum *online*.
2. Proses pencatatan jumlah waktu pemakaian membutuhkan waktu yang lama karena dilakukan secara manual oleh *Technical Support*.

3. Sering terjadinya perbedaan pencatatan jumlah waktu pemakaian antara pihak PT Ejazz Indonesia Baru dengan pelanggannya, menyebabkan terjadi keterlambatan pembayaran dari pelanggan ke PT Ejazz Indonesia Baru.

### 1.3 Tujuan Penelitian

Tujuan penelitian yang akan dilakukan adalah sebagai berikut :

1. Membangun *billing* berbasis web yang dapat memudahkan pelanggan untuk mengetahui jumlah waktu pemakaian dan total biaya pemakaian VoIP.
2. Meningkatkan kualitas layanan administrasi kepada pelanggan.
3. Meminimalisasi keterlambatan proses pembayaran.

### 1.4 Batasan Masalah

Sistem *billing* serta penyajian informasi layanan dan biaya pemakaian VoIP untuk pelanggan merupakan sistem yang berbasis web. Sistem ini merupakan sistem inisiasi karena belum adanya sistem serupa yang dibangun sebelumnya. Sedangkan sistem sebelumnya menggunakan sistem manual dan untuk pembangunan sistem ini harus menyesuaikan dengan sistem yang telah berjalan. Batasan permasalahan yang ada pada perancangan aplikasi ini adalah sebagai berikut :

1. *Billing* VoIP ini berbasis web, sehingga setiap pengguna dapat mengakses melalui internet.
2. Sistem ini masih menggunakan *database* yang telah ada yaitu *Raddb* yang menyimpan data mengenai jumlah waktu penggunaan VoIP dan atribut-atribut lainnya.
3. Tarif ditentukan berdasarkan dua jenis terminasi yaitu terminasi ke telepon rumah atau terminasi ke seluler (*mobile phone*).
4. *Billing* VoIP berbasis web ini hanya menampilkan informasi yang telah di *generate* oleh program *Radbconv* ke dalam *database raddb*.
5. Data survey yang digunakan adalah data waktu pemakaian bulan Mei 2005.

### 1.5 Tahapan Pengembangan Sistem

Dalam pengembangan sistem ini digunakan metode *waterfall* dengan tahapan-tahapan sebagai berikut :

#### 1. Pengumpulan Data

Pengumpulan data dilakukan dengan studi lapangan yaitu studi pengamatan, tinjauan dokumen dan studi kepustakaan yang berkaitan dengan masalah ini.

#### 2. Analisis dan Perancangan

Menggunakan paradigma pengembangan metode *waterfall*, dengan menggunakan analisis terstruktur, dilanjutkan dengan perancangan yang meliputi perancangan proses, perancangan basis data, dan perancangan prosedur kerja.

#### 3. Pengimplementasian

Implementasi hasil perancangan, meliputi pengkodean (*coding*) dengan menggunakan bahasa pemrograman yang telah ditetapkan sampai ke penerapan sistem.

### 1.6 Sistematika Penulisan

Sistematika penulisan laporan Tugas Akhir ini adalah sebagai berikut :

#### BAB I Pendahuluan

Dalam bab ini dijelaskan tentang latar belakang masalah dan perumusan masalah, tujuan penelitian, batasan masalah, tahapan pengembangan sistem dan sistematika penulisan.

#### BAB II Dasar Teori

Pada bab ini berisi landasan teori yang digunakan penulis untuk kebutuhan penulisan laporan Tugas Akhir ini. Teori-teori yang dituliskan pada bab ini seluruhnya adalah dasar dari pengembangan *billing Voice over Internet Protocol (VoIP)* berbasis Web, yaitu konsep *Voice over Internet Protocol (VoIP)*, sistem *billing*, Radius, konsep *HTML (Hyper Text Markup Language)*, *FTP Server*, *TCP/IP*, *HTTP*, Internet, *WWW (World Wide Web)*, Konsep *Database*, *SQL (Structure Query Language)*, Sistem *Database Client/Server*, Konsep *VBScript*, *ASP (Active Server Page)*, Perancangan Sistem, Perancangan Aliran Data.

### **BAB III Keadaan Umum PT Ejazz Indonesia Baru**

Pada bab ini berisi tentang sejarah singkat perusahaan serta struktur organisasi perusahaan dimana penulis melakukan Tugas Akhir.

### **BAB IV Analisa dan Perancangan**

Bab ini membahas tentang identifikasi permasalahan, analisis dan perancangan *database*, diagram alur informasi, *context diagram*, *data flow diagram*, dan *entity relationship diagram*.

### **BAB V Implementasi**

Dalam bab ini penulis mengimplementasikan perancangan aplikasi program yang telah dibuat dengan menampilkan hasil perancangan aplikasi *billing* VoIP berbasis web serta spesifikasi *hardware* dan *software* yang dipakai untuk mengimplementasikan program yang telah dibuat.

### **BAB VI Kesimpulan**

Dalam bab kesimpulan, penulis menjelaskan tentang pembangunan aplikasi sistem *billing* VoIP berbasis web yang dibuat serta saran-saran yang ada kaitannya dengan pembangunan aplikasi ini.

## BAB II

### DASAR TEORI

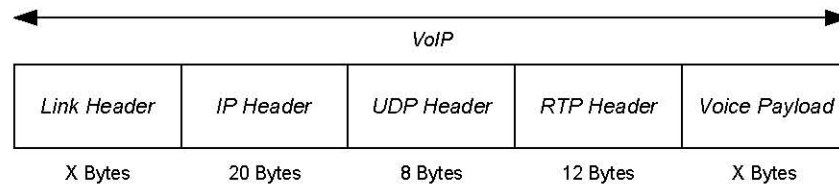
#### 2.1 *Voice over Internet Protocol (VoIP)*

VoIP adalah teknologi yang dapat mengubah percakapan suara ke dalam paket data yang dapat disalurkan lewat koneksi internet *broadband*. Bahkan, kalangan industri mensinyalir teknologi ini akan menggantikan jaringan sirkuit telepon biasa yang sudah berusia 130 tahun.

*Voice over Internet Protocol (VoIP)* adalah teknologi yang dapat mentransfer suara dengan menggunakan *circuit-switched networks* atau *over IP networks*. Jumlah *voice*, *video* dan *data traffic* akan bertambah besar volumenya jika semakin banyak pengguna yang *online*. Komunikasi menggunakan *Voice over Internet Protocol* untuk perusahaan mempunyai banyak keunggulannya dibandingkan dengan telepon rumah biasa atau *Private Branches Exchanges based (PBXs)*. Dari segi biaya relatif murah, karena sambungan internasional tetap dikenai biaya pulsa lokal, selain itu juga semakin beragam aplikasi yang ditawarkan [Tha01].

Sistem VoIP terus berkembang dengan berbagai *feature*, yang juga mempunyai fungsi *call waiting*, *call transfer*, *multiparty conferencing*. Ada kelebihan lainnya yaitu *dial* dari komputer PC dan *voice mail* menggunakan *mail client* dan kemampuan mengintegrasikan suara dan data. Dengan terpisahnya *voice* dan jaringan data, VoIP termasuk sistem yang dapat menekan biaya *maintenance*.

Tiap paket VoIP terdiri atas dua bagian, yaitu *Header* dan *Payload*. *Header* terdiri atas *link header*, *IP header*, *User Datagram Protocol (UDP) header*, *Real-time Transport Protocol (RTP) header*. Bagian-bagian paket VoIP dapat dilihat pada gambar dibawah ini [Tha01]:



**Gambar 2.1** Format Paket VoIP

Teknologi VoIP, dalam perkembangam dan pengembangannya terbentur dengan berbagai masalah yang mendasar, di antaranya :

1. Standarisasi

Permasalahan besar yang dihadapi oleh teknologi ini adalah interoperabilitas antara produk *Internet Telephony* dengan layanan dan sistem yang berbasiskan *Public Switched Telephone Network (PSTN)*, yang sudah terlebih dahulu ada. Pengguna yang ingin melakukan panggilan dengan *Internet Telephony* harus mempunyai suatu perangkat lunak tertentu. Pengembangan standar dan adopsi adalah kunci pemecahan masalah interoperabilitas tersebut. Saat ini, isu utama yang berkembang dan harus dipecahkan serta diantisipasi adalah format *codec*, *protocol transport*, dan layanan direktori.

2. Kualitas

Kinerja suara diukur berdasarkan *delay*. Panggilan pada *PSTN* biasanya mencapai *delay* 50 (lima puluh) sampai 70 (tujuh puluh) milidetik. *Delay* tersebut akan bertambah secara substansial sampai 500 (lima ratus) milidetik pada jaringan internet. Oleh sebab itu, sampai sekarang beberapa pengguna masih mengeluh tentang kualitas tersebut ketika menggunakan *Internet Telephony*. *Delay* sangat mempengaruhi kualitas pembicaraan melalui internet. Manusia dapat mentoleransi sampai dengan dua ratus lima puluh milidetik sebelum terdapat efek lainnya. Permasalahan kualitas suara ini diselesaikan dengan meningkatkan hal-hal berikut ini :

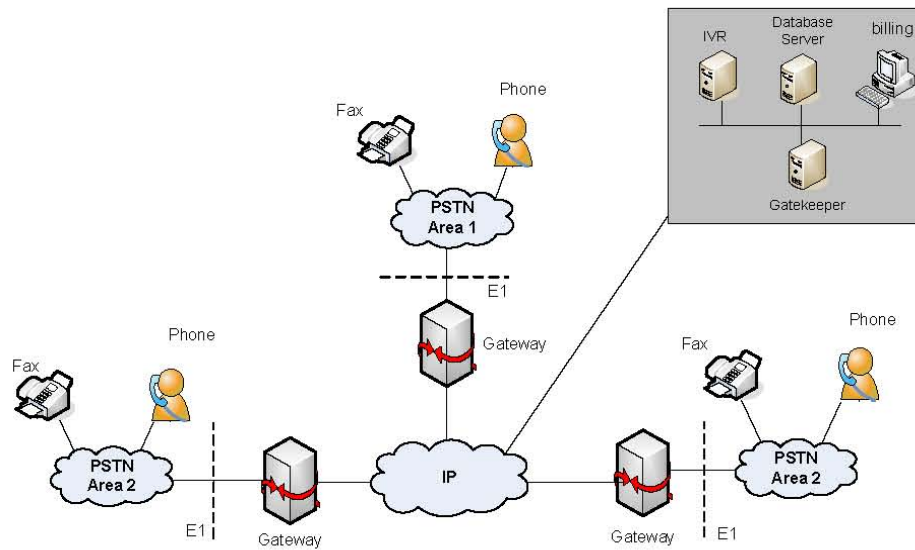
- a. Pengembangan dan peningkatan protokol.
  - b. Menggunakan saluran layanan yang terdedikasi dengan beban trafik yang telah diatur sebelumnya.
  - c. Melakukan re-lokasi akses telepon dengan simpul-simpul *backbone*.
3. Kapasitas

Internet merupakan jaringan terbuka dari banyak jaringan *Internet Service Provider (ISP)* yang berbeda dengan konsekuensi sulit untuk memastikan *bandwidth* jaringan yang baik, sekuen paket data yang teratur, serta *delay* yang kecil. Hilangnya paket adalah masalah yang ada hampir setiap saat, pada umumnya disebabkan oleh bertambahnya populasi pengguna internet sehingga berdampak terhadap kenaikan beban jaringan di internet. Solusi yang diperoleh lebih menekankan kepada penambahan kecepatan *routing* daripada meletakkan banyak *router*.

Konfigurasi *enterprise* atau *corporate* digunakan untuk kebutuhan layanan telekomunikasi antar perusahaan. Model ini memposisikan operator sebagai *Clearing House*, yaitu penyelenggara atau penyedia jaringan (*network provider*) yang dibangun untuk trafik VoIP pada rute yang diinginkan dengan suatu jaminan kualitas tertentu seperti besarnya *delay* maksimum, lebar *bandwidth*, dan batasan bit *error rate* yang terhubung pada beberapa VoIP *service provider*. *Clearing House* dapat dikategorikan sebagai *wholesaler*. Pada konfigurasi ini, *gateway* yang diperlukan adalah kelas *enterprise*, dengan spesifikasi sebagai berikut :

1. Kapasitas *gateway* lebih dari 4 (empat) E1.
2. *Interface* menggunakan *Integrated Services Digital Network (ISDN)*.
3. Diperlukan sistem *billing* yang terintegrasi dengan baik.
4. Diperlukan sistem operasi pendukung.

Pada umumnya, topologi jaringan untuk kelas *enterprise* adalah *point to point* atau *point to multipoint* terbatas. Fungsi *database routing* untuk konfigurasi *multipoint* biasanya ditangani oleh *gatekeeper* yang terintegrasi dengan *gateway* atau bisa berdiri sendiri [Tha02].



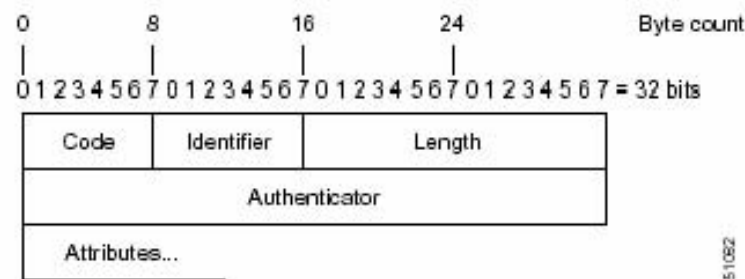
**Gambar 2.2 Konfigurasi VoIP Enterprise**

Keuntungan *Clearing House* bisa dilihat dari sisi VoIP provider, antara lain :

1. *Clearing house* merupakan jembatan bisnis dan teknis, semakin besar dimensi (*coverage*) dan kualitas jaringan *clearing house*, maka semakin besar pula peluang bisnis yang dapat diperoleh VoIP service provider.
2. Model ini memungkinkan penyelenggaraan VoIP *end to end* dengan cakupan yang luas dengan biaya yang lebih efisien.
3. Dengan model ini, pihak *developer* dapat mengurangi resiko ketidakpuasan pelanggan terhadap kualitas layanan *clearing house* dengan memberikan garansi kualitas suara.
4. Meningkatkan pendapatan dari segi *termination call* yang berasal dari VoIP provider lain yang masuk ke dalam aliansi *clearing house* yang bersangkutan.
5. Memudahkan dan mempercepat pembentukan hubungan ke berbagai VoIP provider karena negosiasi hanya perlu dilakukan satu kali yaitu ke *clearing house* yang bersangkutan, tanpa harus bernegosiasi atau mengikat kontrak.

## 2.2 Radius

Radius protokol dibangun oleh *Livingston Enterprise, Inc.*, sebagai akses *server authentication* dan *accounting protocol* yang dapat menerima akses koneksi pengguna, hak pengguna, dan termasuk seluruh informasi yang penting untuk pengguna. Pada umumnya *Radius* diperuntukkan kepada pengguna jaringan yang terkait kepada *server*. Berikut adalah detail mengenai *Radius Packet Diagram* yang tertera pada gambar di bawah ini [Wil197]:



**Gambar 2.3 Radius Packet Diagram**

Keterangan:

a. *Code*

Berisi 1 *octet* = 8 *bytes*, *Code* merupakan indikasi dari *type Radius Packet*.

1. *Access-Request*
2. *Access-Accept*
3. *Access-Reject*
4. *Accounting-Request*
5. *Accounting-Response*

b. *Identifier*

Berisi 1 *octet* = 8 *bytes*, *Identifier* akan membantu *Radius Server* dalam mencocokkan antara permintaan dan respon serta mendeteksi duplikasi permintaan.

c. *Length*

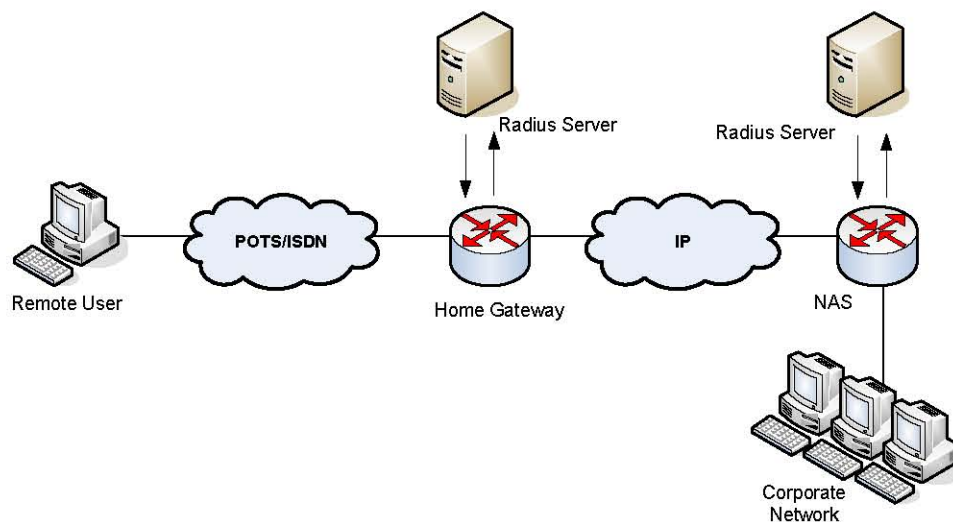
Berisi 2 *octet* = 16 *bytes*, *Length* merupakan panjang dari paket yang dibawa.

d. *Authenticator*

Berisi 16 *octet* = 32 *bytes*, *Authenticator* berguna dalam memberikan balasan izin yang diminta oleh *Radius Server*. Ada 2 (dua) tipe *Authenticator* yang digunakan antara lain :

1. *Request Authenticator* terdapat di *Access-Request* dan *Accounting-Request Packet*.
2. *Response Authenticator* *Access-Accept*, *Access-Reject*, *Access-Challenge*, *Accounting-Response Packet*.

Komunikasi yang dilakukan antara *Network Access Server (NAS)* dan *Radius Server* adalah berdasarkan *User Datagram Protocol (UDP)* yang mendukung *client/server protocol*. *Radius* biasanya digunakan pada *UNIX* dan *Windows NT*. *Radius server* akan merekam seluruh transaksi yang ada pada *server* sekitar 64 atribut berdasarkan standarisasi rancangan *Internet Engineering Task Force (IETF)*, tetapi tidak seluruh atribut yang akan diberikan kepada pengguna tergantung informasi apa saja yang akan digunakan, hal yang paling penting sesuai dengan atribut *billing VoIP* berdasarkan standar *IETF* yang tertera pada Lampiran C. Berikut ini adalah gambar jaringan dengan *radius server* [Tha02] :



Gambar 2.4 Radius Server

### 2.3 Sistem *Billing*

Sistem *Billing* merupakan rangkaian dari berbagai proses yang terbentuk untuk menghasilkan suatu laporan yang dapat digunakan berbagai pihak. Secara umum komponen dari sistem *billing* terdiri dari tiga komponen utama yaitu *mediation*, *rating*, dan *billing application* [Boy03].

1. Sistem Mediasi (*Mediation System*)

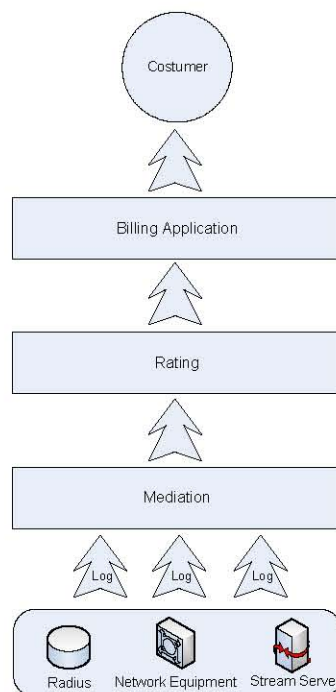
Berfungsi untuk mengolah data mentah yang dihasilkan oleh sumber data sehingga menjadi data yang siap diolah oleh proses *billing*.

2. Sistem *Rating*

Digunakan untuk menghitung dan melakukan konversi data pemakai jaringan menjadi data besaran uang yang akan menjadi komponen dalam perhitungan besar tagihan di proses *billing*.

3. *Billing Application*

Merupakan media yang akan disampaikan kepada pelanggan untuk mengetahui jumlah tagihan yang harus dibayarkan. Aplikasi *billing* ini dapat berupa *website*, laporan tagihan (*invoice*), dan *Interactive Voice Response (IVR)*.



Gambar 2.5 Komponen dan Proses *Billing*

## 2.4 *Hyper Text Markup Language (HTML)*

*Hypertext Transfer Protocol* adalah protokol *TCP/IP* pada *layer* aplikasi yang merupakan dasar *World Wide Web*. Diimplementasikan pertama kali pada akhir tahun 1990. *HTTP* merupakan protokol *request-response*. Sebuah program *client HTTP* membuat koneksi dengan *HTTP Server* dan meminta halaman tertentu dari *server*, program *server HTTP* menerima koneksi tersebut dan menjawab permintaan dari *client*. Secara singkat, *HTTP* adalah protokol yang melakukan hubungan antara *client* dan *server*, yang mana pada sisi *server*, akan menerima permintaan berupa halaman web dari *client*, lalu kemudian akan mengirim permintaan tersebut ke *client*. Permintaan halaman web dari *client* dilakukan dengan program yang sering disebut *browser*. Program ini banyak jenisnya, diantaranya adalah *Microsoft Internet Explorer*, *Netscape*, *Mozilla*, *Firefox*, *Lynx*, dsb. Secara umum, *HTTP* bekerja pada *port* 80, pada jaringan *TCP/IP* [Rus04].

## 2.5 *TCP/IP*

Jaringan komputer yang berada dalam internet memiliki tipe yang berbeda-beda sehingga dibutuhkan suatu aturan tertentu yang mendefinisikan bagaimana sesuatu semestinya dilakukan. Aturan ini dinamakan dengan protokol. yang juga digunakan dalam jaringan internet. *TCP/IP* merupakan serangkaian protokol dimana setiap protokol melakukan sebagian dari keseluruhan tugas komunikasi jaringan [Rus04].

## 2.6 *Hyper Text Transfer Protocol (HTTP)*

Internet beroperasi menggunakan satu set protokol yang mengontrol dan mengarahkan data di dalam jaringan (*Network*), secara keseluruhan protokol ini disebut sebagai *Transfer Control Protocol / Internet Protocol (TCP/IP)*.

Pelayanan *HTTP* ini memungkinkan *user* untuk mengakses suatu halaman web yang dapat berupa teks, gambar, suara, bahkan film. *User* bisa melihat informasi yang ada dengan memasuki sebuah alamat pada suatu URL.

## 2.7 Internet

Pada tahun 1969, lembaga riset Departemen Pertahanan Amerika, *Defense Advanced Research Project Agency (DARPA)*, mendanai riset untuk mengembangkan jaringan komunikasi data antar komputer. Riset ini bertujuan untuk mengembangkan aturan komunikasi data antar komputer. Pengembangan jaringan ini ternyata sukses dan melahirkan *Advanced Research Project Agency Network (ARPANET)*. Tahun 1972, *ARPANET* didemonstrasikan di depan peserta *The First International Conference on Computer Communications* dengan menghubungkan 40 *node*.

Aplikasi internet yang pertama kali ditemukan adalah *File Transfer Protocol (FTP)*. Kemudian diikuti oleh *e-mail* dan *telnet*. *E-mail* menjadi aplikasi yang paling populer di masa *ARPANET*. Protokol komunikasi data yang digunakan pada waktu itu, yaitu *Network Communication Protocol (NCP)*, tidak sanggup menampung *node* komputer yang besar. *DARPA* kemudian mendanai pembuatan protokol komunikasi yang lebih umum. Protokol ini dinamakan *Transmission Control Protocol / Internet Protocol (TCP/IP)*. Departemen Pertahanan Amerika menyatakan *TCP/IP* menjadi standar untuk jaringannya pada tahun 1982. Protokol ini kemudian diadopsi menjadi standar *ARPANET* pada tahun 1983. Perusahaan *Bolt Beranek Newman (BBN)* membuat protokol *TCP/IP* berjalan di atas komputer dengan sistem operasi *UNIX*. Sejak saat itu perkawinan antara *UNIX* dan *TCP/IP* dimulai.

Pada tahun 1984 jumlah *host* di internet melebihi 1000 buah. Pada tahun itu pula diperkenalkan *Domain Name System (DNS)* yang mengganti fungsi tabel nama *host*. Sistem *domain* inilah yang sampai saat ini kita gunakan untuk menuliskan nama *host*. Tahun 1986, lembaga ilmu pengetahuan nasional Amerika Serikat (*US National Science Foundation – NSF*) mendanai pembuatan jaringan *TCP/IP* yang dinamai *NSFNET*. Jaringan ini digunakan untuk menghubungkan lima pusat komputer super dan memungkinkan terhubungnya universitas-universitas di Amerika Serikat dengan kecepatan jaringan tulang punggung sebesar 56 kbps. Jaringan inilah yang kemudian menjadi embrio berkembangnya internet yang kita kenal sekarang ini.

Pada tahun 1987 berdiri *UUNET* yang saat ini merupakan salah satu *provider* utama internet. Tercatat pula pada tahun tersebut jumlah *host* melewati angka 10.000. Setahun kemudian kecepatan jaringan tulang punggung *NSFNET* ditingkatkan menjadi T1 (1,544 Mbps). Di samping itu juga terdapat beberapa negara di Eropa yang masuk ke jaringan *NSFNET*.

Perkembangan internet semakin luas dan sampai menjangkau Australia dan Selandia Baru pada tahun 1989. Pada tahun tersebut jumlah *host* di internet mencapai jumlah 100.000. Dua tahun kemudian aplikasi di internet bertambah dengan diciptakannya *World Wide Web* (WWW) dan *Gopher*. Pada tahun tersebut kecepatan jaringan tulang punggung *NSFNET* ditingkatkan menjadi T3 (45 Mbps).

Pada tahun 1992 jumlah *host* di internet mencapai 1 juta *host*. Salah satu pemicu perkembangan ini adalah semakin meluasnya penggunaan layanan *Gopher* yang terdapat di internet. Pada tahun ini juga untuk pertama kalinya dilaksanakan siaran *audio* dan *video multicast* melalui *Internet Engineering Task Force Multicast Backbone (IETF MBONE)* [Rus04].

## 2.8 *World Wide Web (WWW)*

*World Wide Web (WWW)* dapat disebut juga dengan web, yaitu perangkat yang digunakan untuk mencari informasi di internet yang mempunyai kemampuan untuk menampilkan kata-kata, tipografi, dan gambar di layar komputer. Web dapat dengan mudah digunakan oleh pengguna komputer yang telah akrab dengan antarmuka grafis komputer *desktop*. Aplikasi untuk membaca halaman web disebut *web browser*, antara lain adalah *Mosaic*, *Netscape Navigator*, *Internet Explorer*, dan *Opera* [Bru97].

Halaman (*Page*) milik seseorang atau suatu organisasi dapat ditampilkan pada saat pertama kali *web browser* dijalankan atau dapat juga dikatakan sebagai halaman yang langsung muncul dilayar ketika pertama kali membuka browser. Web memakai bahasa *Hyper-Text Markup Language (HTML)*. Halaman-halaman web disimpan dalam web server dalam bentuk file *HTML*. Semua *server web* menggunakan *Hyper-Text Transfer Protocol (HTTP)* untuk mengirim informasi hiperteks ke seluruh internet. Keunikan web adalah fleksibilitas dan

kemudahannya, maksudnya ialah sistem informasi yang dirancang oleh penulis adalah sistem informasi yang berbasis web karena sistem ini sangat cocok sebagai sarana untuk memberikan informasi [Bru97].

### 2.9 Konsep *Visual Basic Script (VBScript)*

*VBScript* merupakan sebuah *script* yang unik. Ketika berdiri sendiri sebagai *script* pemrograman yang disebut dengan *VBScript* yang berwujud *client-side*. Akan tetapi, jika dimanfaatkan sebagai *ASP programming* maka akan berbentuk *server-side*. Penggunaan dan cara tulis *VBScript* berjenis *client-side* dan *ASP* berbeda secara jelas [Kur00].

### 2.10 *Active Server Pages (ASP)*

Microsoft telah menyediakan suatu bahasa pemrograman *script* yang dinamakan dengan *ASP* yang merupakan pengembangan lebih lanjut dari *Internet Database Connector (IDC)*, sehingga dapat menggabungkan pengelolaan *database*. Dibanding dengan *IDC*, *ASP* memiliki format yang lebih terintegrasi mengenai koneksi *query* ke *database server* dan mengenai penampilan hasil dari *query*-nya, akan diformulasikan dengan suatu format tertentu.

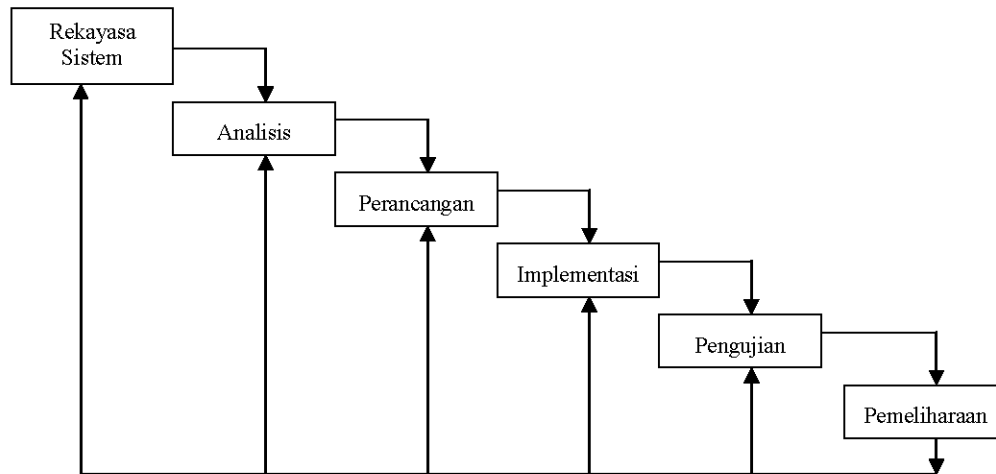
File *script ASP* selalu berekstensi *.ASP* dimana terdiri dari bahasa *HTML* biasa dan pemrograman *script* diawali dengan “<%” dan diakhiri dengan “%>”.

Dalam suatu *script ASP* melalui pemakaian *request object* dapat diperoleh data yang dimasukkan ke dalam *form* dari suatu halaman web. *Script ASP* juga bisa memperoleh masukan data dari suatu halaman web yang dimasukan melalui pemakaian perintah *HREF*. Adapun jumlah data yang dimasukan bisa lebih dari satu dengan menggunakan tanda “&” [Kur00].

### 2.11 Metode Perancangan Sistem

Pengembangan sebuah sistem informasi berbasis komputer dapat menggunakan beberapa metode sebagai acuan. Setiap metode akan dibagi menjadi tahapan-tahapan yang akan memudahkan dalam pembangunan sistem informasi. Metode yang sering juga disebut “*waterfall*” atau “*classic life cycle*” ini menggunakan pendekatan yang sistematis dan sekuensial dalam membangun

perangkat lunak yang dimulai pada tingkatan sistem dan pengembangan melalui tahapan analisis, perancangan, pengkodean, pengujian, dan pemeliharaan seperti terlihat pada gambar di bawah ini [Pre97]:



**Gambar 2.6** Metode Perancangan Sistem *Waterfall*

### 2.11.1 Rekayasa Sistem

Merupakan tahapan yang pertama kali dilakukan yaitu merumuskan sistem yang akan dibuat. Hal ini bertujuan agar pengembang benar-benar memahami sistem yang akan dibuat dan langkah-langkah serta kebijaksanaan apa saja yang berkaitan dengan pengembangan sistem tersebut.

### 2.11.2 Analisis

Dari rumusan sistem yang diperoleh dari tahap pertama, selanjutnya dilakukan analisis yang berkaitan dengan proses dan data yang diperlukan oleh sistem serta keterkaitannya. Tujuan dilakukannya tahapan ini adalah sebagai berikut:

1. Memahami sistem yang ada pada saat ini.
2. Mendefinisikan permasalahan sistem.
3. Menentukan kebutuhan sistem secara garis besar sebagai persiapan ke tahap perancangan.

Analisis dilakukan dengan permodelan menggunakan metode *Data Flow Oriented* dengan *tool Data Flow Diagram (DFD)*.

### 2.11.3 Perancangan

Pada tahap perancangan ini diberikan gambaran umum yang jelas kepada pengguna dan rancang bangun yang lengkap tentang sistem yang akan dikembangkan kepada pihak-pihak yang terlibat dalam pengembangan sistem.

Tahapan perancangan sistem di sini dibagi menjadi dua bagian, yaitu perancangan global dan perancangan rinci. Perancangan global dilakukan untuk memberikan gambaran umum kepada pengguna tentang sistem yang dirancang dan sebagai persiapan untuk tahap perancangan rinci. Perancangan rinci dilakukan untuk memberikan gambaran rancang bangun yang lengkap kepada *programmer* dan pihak-pihak lain yang terlibat dalam pengembangan sistem sebagai persiapan untuk tahap implementasi.

### 2.11.4 Implementasi

Setelah tahap perancangan sistem selesai dilakukan, selanjutnya adalah pengimplementasian rancangan sistem ke dalam kode-kode bahasa pemrograman yang diinginkan. Pada tahap ini dilakukan pembuatan komponen-komponen sistem yang meliputi implementasi modul-modul program, antarmuka dan basis data.

### 2.11.5 Pengujian

Tahap pengujian, digunakan untuk memeriksa apakah *software* yang sudah dibuat tersebut sudah sesuai dengan hasil analisis dan permintaan *user* serta *software* tersebut sudah berjalan dengan baik. Proses pengujian diutamakan pada logika pemrograman yang ada pada *software*.

### 2.11.6 Pemeliharaan

Setelah dilakukan pengujian dan sistem diyakini telah *valid*, selanjutnya sistem tersebut didistribusikan kepada pengguna. Hal ini tidak berarti pekerjaan pengembang telah selesai, karena masih terdapat tahap pemeliharaan sistem yang

harus dilakukan. Pada tahap ini dilakukan evaluasi terhadap sistem yang baru untuk melihat apakah sistem telah memenuhi tujuan yang ingin dicapai. Dari hasil evaluasi ini dimungkinkan untuk melakukan perubahan-perubahan yang diperlukan terhadap sistem.

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

*Data Flow Diagram* merupakan diagram alir yang menggambarkan aliran data, proses data dan aliran kontrol dalam menghasilkan *output*. Komponen di dalam *DFD* :

1. Proses, digambarkan dengan lingkaran yang berfungsi untuk menunjukkan proses data dalam sistem atau transformasi data dari masukan menjadi keluaran.
2. Aliran, digambarkan dengan anak panah yang berfungsi sebagai petunjuk aliran data atau gerakan informasi.
3. Penyimpanan (*storage*), digambarkan dua garis atas bawah yang mengapit nama dari penyimpanan tersebut. Berfungsi untuk memodelkan data.
4. *Terminator*, digambarkan dengan persegi panjang yang berfungsi untuk mewakili komponen luar yang berhubungan dengan sistem.

*DFD* dapat digambarkan dalam diagram *context* dan level n, huruf n melambangkan level dan sub proses dari setiap proses.

#### a. Diagram *context*

Diagram *context* adalah gambar yang menjelaskan secara garis besar proses-proses yang terjadi di dalam sistem. Komponennya berupa *input*, proses dan *output*.

#### b. *DFD* level n

*DFD* level n ini digunakan untuk menggambarkan proses secara lebih rinci dari setiap proses maupun sub proses.

Dalam penterjemahan data secara lebih rinci yang dipakai dalam suatu *DFD* diperlukan sebuah sarana penterjemah, di dalam perancangan maupun basis data sarana tersebut disebut sebagai kamus data dan *PSPEC* (*Process Specification*) [Pre97].

### 2.13 Konsep Basis Data

*Database* adalah sekumpulan data atau informasi yang saling terhubung atau terkait dan disimpan secara bersamaan dalam media penyimpanan. Kumpulan data ini harus mengandung informasi untuk mendukung suatu sistem. Penyimpanan data pada *database* dikelola oleh satu atau banyak file. *Database* merupakan bagian penting dari sistem informasi. Penyimpanan data secara konvensional akan menyebabkan duplikasi data, sehingga diperlukan *database* yang baik dengan kriteria sebagai berikut :

1. *Database* harus bersifat *Object Oriented* bukan *Program Oriented*. Maksudnya adalah dalam aplikasi data apabila diakses secara bersamaan, tidak terjadi perubahan pada struktur *database*.
2. Dapat dikembangkan dalam hal volume dan struktur.
3. Kerangkapan data (*Data Redundancy*) harus seminimal mungkin.

Bahasa yang digunakan dalam sistem *database* adalah *SQL*. Elemen-elemen yang terkait dalam sistem *database* adalah :

1. Basis Data
2. Perangkat Keras (*Hardware*)
3. Perangkat Lunak (*Software*)
4. Pengguna *database* yaitu *Programmer* dan sistem analis / *administrator* data.

### 2.14 *Database Management System (DBMS)*

Sistem manajemen basis data (*database management system*) adalah sekumpulan file yang saling berhubungan dengan sekumpulan program yang memungkinkan pengguna mengakses dan memodifikasi file tersebut. Fungsi utama sistem basis data adalah untuk menyediakan abstraksi data bagi pengguna tanpa perlu ditunjukkan bagaimana data tersebut disimpan dan dijaga [Sil97].

*Microsoft SQL 2000 Server* adalah salah satu program yang dapat digunakan untuk mengorganisasikan dan mendukung pembuatan basis data yang berbasis internet.

## 2.15 *Structured Query Language (SQL)*

### 2.15.1 *Data Definition Language (DDL)*

*Data Definition Language (DDL)* adalah bahasa yang digunakan untuk mendefinisikan tempat penyimpanan data berupa tabel, *index* dan *view*. Dalam pendefinisian sebuah tabel ada beberapa jenis data yang harus ditetapkan sesuai dengan jenis data yang akan disimpan untuk memperoleh informasi yang diperlukan. Jenis data yang tersedia biasanya tergantung dari masing-masing *vendor database*, namun secara umum terdiri dari *varchar*, *char*, *text*, *number*, dan *date* [Tim98].

### 2.15.2 *Data Manipulation Language (DML)*

Setelah tempat penyimpanan data terdefiniskan dan dibuat, langkah selanjutnya adalah melakukan pengolahan data berupa mengisi (*insert*), mengganti (*update*), dan menghapus (*delete*) [Tim98].

### 2.15.3 *Transaction Processing Language (TPL)*

Setiap proses manipulasi data *insert*, *update*, dan *delete* tidak secara otomatis langsung bersifat permanen di *disk*, namun masih dalam kondisi sementara. Permanen atau tidaknya manipulasi data tersebut ditentukan oleh perintah *TPL*. Yang terdiri dari *commit* dan *rollback*. Jika hasil manipulasi data tersebut ingin dipermanenkan maka perlu dilanjutkan dengan perintah *commit*, sebaliknya untuk membatalkannya maka lakukan *rollback*, dan nantinya data akan kembali kepada kondisi semula sebelum terjadinya manipulasi [Tim98].

### 2.15.4 *Data Control Language (DCL)*

Bahasa ini erat kaitannya dengan sekuritas data yang mengontrol kewenangan *user* dalam mengakses *database*. Untuk bisa mengakses *database* minimal seorang *user* harus mempunyai *account* yang terdefiniskan dari *username* dan *password*. Dalam suatu *database* setiap *account* mempunyai tabel masing-masing dan pada dasarnya hanya bisa diakses oleh *user* lainnya maka si pemilik tersebut harus memberikan kewenangan pengaksesan tertentu kepada *user*

lain yang umumnya menggunakan perintah *grant*. Sebaliknya untuk menghapus kewenangan akses tabel digunakan perintah *revoke* [Tim98].

### 2.16 Sistem *Database Client/Server*

Sekarang ini, pemakaian teknologi *client / server* sudah merupakan suatu kebutuhan pokok yang mutlak perlu diimplementasikan dalam setiap pengembangan suatu sistem aplikasi *database*, karena teknologi tersebut memiliki beberapa keuntungan yang dapat diraih oleh sistem yang memakainya. Adanya pembagian tugas yang jelas antara *client* sebagai *user interface* terhadap *user* dan *server* sebagai pengolah *database* untuk memanipulasi dan memelihara data yang diperlukan oleh *user*. Pemakaian teknologi *client / server* ini dapat memanfaatkan jaringan komputer secara lebih efisien dan efektif dengan menekan sekecil mungkin lalu lintas data yang terjadi antara komputer *client* dan komputer *server*.

Sistem *two-tiers* (penghubung dua sistem) telah berkembang menjadi sistem *multi-tiers* (penghubung banyak sistem), sejak diperkenalkannya teknologi *client/server*. Sistem ini memiliki performansi yang lebih baik, dilihat dari sisi keamanan cukup baik karena hubungan antara *user* sebagai *client* dan *database* sebagai *server* tidak secara langsung terkoneksi, akan tetapi harus melalui suatu media penghubung sistem (*middle-tier*) seperti *Internet Information Server (IIS)*, termasuk *Active Server Pages (ASP)* yang berfungsi sebagai aplikasi *server* antara *user* dan *database*. Keuntungan dari sistem aplikasi *database* berbasis *client/server multi-tier* ini adalah bahwa *software* aplikasi *database* yang telah dibuat tidak harus ter-*install* pada setiap *client* tetapi terpasang di komputer yang berfungsi sebagai *middle-tier*.

Sistem basis data yang berkategori *database server* yang berfungsi untuk menjaga keamanan data, umumnya memberlakukan pemakaian *user account* yang disertai dengan *password*, sehingga dalam setiap pengaksesan basis data selalu disertai dengan permintaan pengecekan *user* dan *password*-nya. Struktur penyimpanan data secara hirarkis terdiri dari :

1. *Database*

Struktur yang tertinggi yang menampung keseluruhan *user account* dan *object database* yang dimiliki *user*.

2. **Tabel**  
Struktur data dua dimensi yang terdiri dari *rows (record/tuple)* dan kolom (*field/atribut*).
3. **Index**  
Struktur data penunjang untuk lebih meningkatkan kinerja proses pencarian data.
4. **View**  
*Visual* tabel yang hanya mendefinisikan kumpulan data dari satu tabel atau lebih dengan syarat-syarat tertentu, namun secara fisik tidak memiliki data tersendiri, tetapi berasal dari tabel-tabel yang sudah ada.

## 2.17 Perancangan Aliran Data

### 2.17.1 Basis Data dan *Entity Relationship Diagram (ERD)*

- a. **Basis Data**  
Basis data merupakan susunan dari *file* maupun data operasional yang lengkap dan terintegrasi disimpan dengan cara-cara tertentu untuk kepentingan tertentu [Fat99].
- b. ***Entity Relationship Diagram (ERD)***  
*ERD* menunjukkan hubungan antara basis data secara konsisten yang didasarkan pada atribut kunci pada masing-masing *field* [fat99].  
Komponen-komponen dari *ERD* adalah :
  1. *Entity*, digambarkan dengan kotak persegi panjang yang menunjukkan entitas dari suatu basis data.
  2. Atribut, digambarkan dengan lingkaran yang menerangkan karakter *entity*.
  3. Hubungan, disebut *relationship* yang menunjukkan hubungan bentuk antara *entity* dengan isi dari hubungan itu sendiri.
- c. **Normalisasi**  
Pada normalisasi akan diuji beberapa kondisi, apakah terjadi kesulitan pada penambahan data, menghapus data, merubah data dan membaca data. Bila ditemukan suatu kesulitan dalam pengujian maka relasi tersebut perlu

dipecah dalam beberapa tabel lagi, dengan kata lain perancangan tersebut belum mendapatkan perancangan yang optimal [Fat99].

Hal yang perlu diperhatikan dalam menormalisasi basis data :

1. *Super key*  
*Super key* adalah atribut atau kumpulan atribut yang bernilai unik dan dapat digunakan untuk mengidentifikasi suatu entitas.
2. *Candidate key*  
*Candidate Key* adalah atribut atau kumpulan atribut yang bernilai unik dan dapat digunakan untuk mengidentifikasi suatu entitas dengan jumlah atribut minimal.
3. *Primary key*  
*Primary key* adalah *candidate key* yang dipilih oleh perancangan basis data sebagai kunci utama.
4. *Foreign key*  
*Foreign key* adalah atribut pada suatu tabel yang menjadi kunci primer pada tabel lain untuk mendukung *referential integrity*.

### 2.17.2 Perancangan Antarmuka

Perancangan antarmuka digunakan untuk menunjukkan struktur rancangan tampilan dari *software* kepada pembuat *software* aplikasi. Perancangan antarmuka berfokus pada tiga hal, yaitu perancangan antar modul *software*, rancangan antarmuka antara *software* dengan lingkungannya dan rancangan antar muka *software* dan *user* [Pre97].

### 2.17.3 Program Design Language (PDL)

*Program Design Language (PDL)* kadang juga disebut sebagai *pseudocode*, yang merupakan sebuah bahasa pemrograman yang diterjemahkan ke bahasa sehari-hari yang sekiranya bisa dimengerti secara langsung [Pre97].

Sebuah *PDL* harus mempunyai beberapa karakteristik, yaitu :

- a. Mempunyai sebuah *syntax* tertata dari kata kunci yang menterjemahkan semua struktur konstruksi, deklarasi data dan bersifat modular.

- b. Mempunyai sebuah *syntax* bebas dari bahasa alami yang menterjemahkan proses.
- c. Mempunyai fasilitas pendeklarasian.
- d. Sub program *definition* dan teknik pemanggilan harus mendukung model dan deskripsi antarmuka.

#### **2.17.4 *Control Specification (CSPEC)***

*Control Specification* merupakan penjelasan dalam bentuk tabel dari tiap proses mengenai hubungan antar proses yang ada pada perancangan. Apabila proses satu dengan yang lainnya mempunyai hubungan maka ditandai dengan angka 1 dan apabila tidak mempunyai hubungan maka ditandai dengan angka 0.

#### **2.17.5 *Process Specification (PSPEC)***

*Process Specification* adalah penjelasan mengenai aliran data yang masuk atau keluar dari proses. Bentuk dari *PSPEC* adalah suatu tabel yang berisi mengenai aliran, data, dan proses.

## BAB III

### Keadaan Umum PT Ejazz Indonesia Baru

#### 3.1 Sejarah Singkat PT Ejazz Indonesia Baru

PT Ejazz Indonesia Baru (EIB) berdiri sejak 10 Maret 2000 sesuai dengan akta pendirian perusahaan No.1.899.692.6-027 yang disahkan di Jakarta. PT Ejazz Indonesia Baru merupakan anak perusahaan PT Sugiron yang bergerak di bidang distribusi oli untuk kendaraan. EIB adalah perusahaan yang bergerak di bidang jasa pelayanan telekomunikasi dengan menggunakan teknologi *Voice Over Internet Protocol (VoIP)* sebagai *Incoming Clearing House Termination Call* dari *VoIP provider* lainnya. Layanan telekomunikasi sendiri baru dimulai pada tahun 2001.

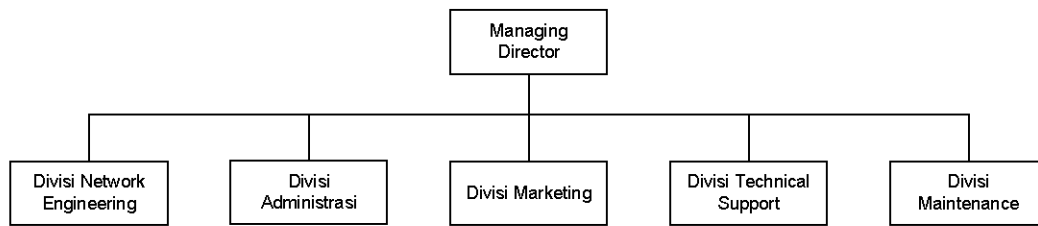
Pada tahun 2002, EIB membuka cabang baru di Bandung untuk meningkatkan penetrasi *termination call* sesuai dengan permintaan pelanggan yang menginginkan koneksi ke kota Bandung. Pada waktu yang akan datang tidak menutup kemungkinan EIB untuk memperluas bidang bisnisnya ke *outgoing termination call* memakai Sambungan Langsung Internasional (SLI) dan Sambungan Langsung Jarak Jauh (SLJJ) yang sampai saat ini sedang dalam tahap pembangunan.

#### 3.2 Struktur Organisasi

PT Ejazz Indonesia Baru, dikepalai oleh seorang *Managing Director* yang langsung membawahi 5 (lima) divisi, diantaranya adalah sebagai berikut :

1. Divisi *Network Engineering*
2. Divisi Administrasi
3. Divisi *Marketing*
4. Divisi *Technical Support*
5. Divisi *Maintenance*

Masing-masing divisi dikepalai oleh seorang *Manager* yang bertanggung jawab langsung kepada *Managing Director*, seperti yang terlihat pada Gambar 3.1.



**Gambar 3.1 Struktur Organisasi PT Ejazz Indonesia Baru**

### **Tugas dan Tanggung Jawab**

Berikut ini adalah tugas dan tanggung jawab yang dimiliki oleh masing-masing divisi di PT Ejazz Indonesia Baru

#### **a. Divisi *Network Engineering***

1. *Instalasi Gateway & Router*
2. *Maintenance Gateway*
3. *Gateway Monitoring Jakarta*
4. *Routing Plan , Gatekeeper setting*
5. *Router monitoring & instalasi*
6. *Bandwidth monitoring*
7. *Network Monitoring*
8. *Instalasi Billing Software (radius) pada Server*
9. *Web Hosting maintenance*
10. *E-mail Server Maintenance*
11. *Network Diagram up-date*
12. *SQL Administrator*

**b. Divisi Administrasi**

1. Monitor *account balance* (sisa *prepaid*) dari semua *customer prepaid*
2. Mengirim Tagihan (*invoice*) kepada *customer*
3. Memproses *customer* baru
4. Membuat laporan *Minute of Meeting* (MoM)
5. Membuat Laporan Keuangan
6. Membayar tagihan

**c. Divisi Marketing**

1. Mencari *customer* terminasi baru
2. Menjaga hubungan dengan *customer*.
3. Negosiasi harga / tarif untuk *customer* baru
4. Mencari *partner* / mitra terminasi
5. Menjaga hubungan dengan *partner* / mitra
6. *Contact Person* dengan Telkom
7. *Contact Person* dengan Indosat
8. *Contact Person* dengan XLnet
9. *Contact Person* dengan IM2
10. Monitor jumlah trafik per hari setiap *customer*
11. Monitor pembayaran dari *customer*
12. *Follow up* ke pelanggan apabila trafik turun
13. Mencari Informasi kondisi pasar *VoIP*

**d. Technical Support**

1. *Back-up* Data Cisco Gateway & Router
2. *Gateway Monitoring*
3. *Maintenance Billing* (Radius)
4. *Generating Billing*
5. *Back-up & Restore, Deleting, Compressing Billing*
6. Transfer data from *pending* to *posted* CDR to CD
7. Instalasi Program Aplikasi dan *updating*
8. Instalasi *Operating System* (OS)

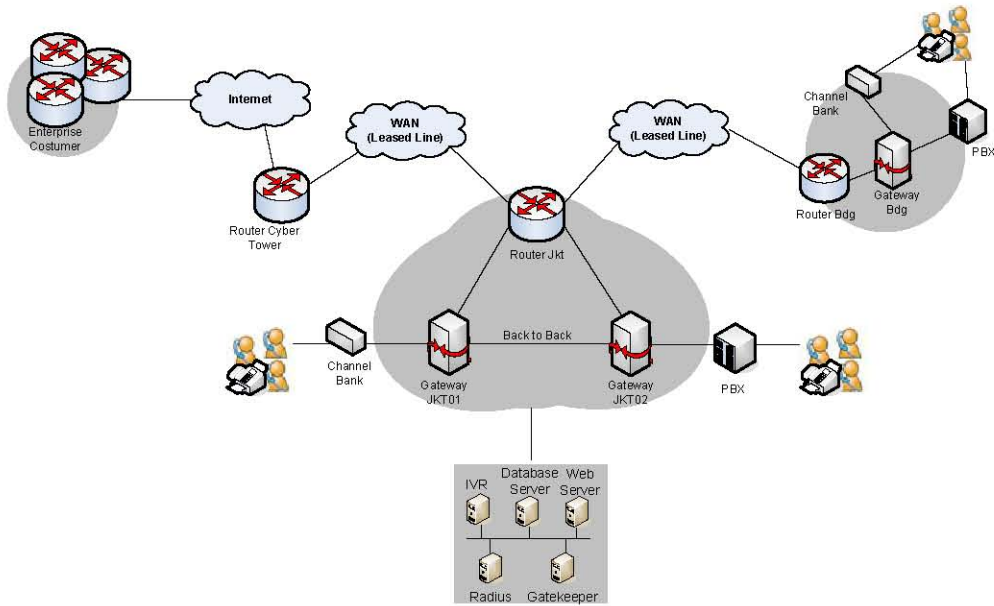
9. Instalasi *Hardware* PC
10. Trafik *monitoring*
11. Dokumentasi prosedur teknis

e. **Maintenance**

1. *Tools Kit* Jakarta
2. *Monitoring* Listrik / UPS / AC
3. *AC rolling*
4. *Maintenance* PaBX
5. *Maintenance* Channel Bank
6. *Telepon Line monitoring*
7. Instalasi Telepon Baru

### 3.3 Konfigurasi Jaringan *VoIP*

PT Ejazz Indonesia Baru menerapkan konfigurasi jaringan *VoIP* dengan model *enterprise* yang memosisikan sebagai *Clearing House* yaitu penyelenggara atau penyedia jaringan yang dibangun untuk trafik *VoIP* pada rute yang diinginkan dengan jaminan kualitas tertentu. Koneksi yang digunakan untuk melakukan hubungan antara *Gateway* di Jakarta dengan *Gateway* yang berada di Bandung adalah menggunakan *leased line* E1 sebesar 2 MB guna menghindari *packet loss*, walaupun kapasitas untuk melakukan koneksi *VoIP* sudah lebih dari cukup. Untuk *Gateway* Jakarta menggunakan *Back to Back* antara *Gateway* JKT01 dengan *Gateway* JKT02 berfungsi sebagai *backup*, apabila salah satu *Gateway* mengalami kelebihan proses. Untuk lebih lengkapnya konfigurasi jaringan EIB dapat dilihat pada Gambar 3.2.



Gambar 3.2 Konfigurasi Jaringan PT Ejazz Indonesia Baru