BAB II

LANDASAN TEORI

2.1 *Electronic Mail*

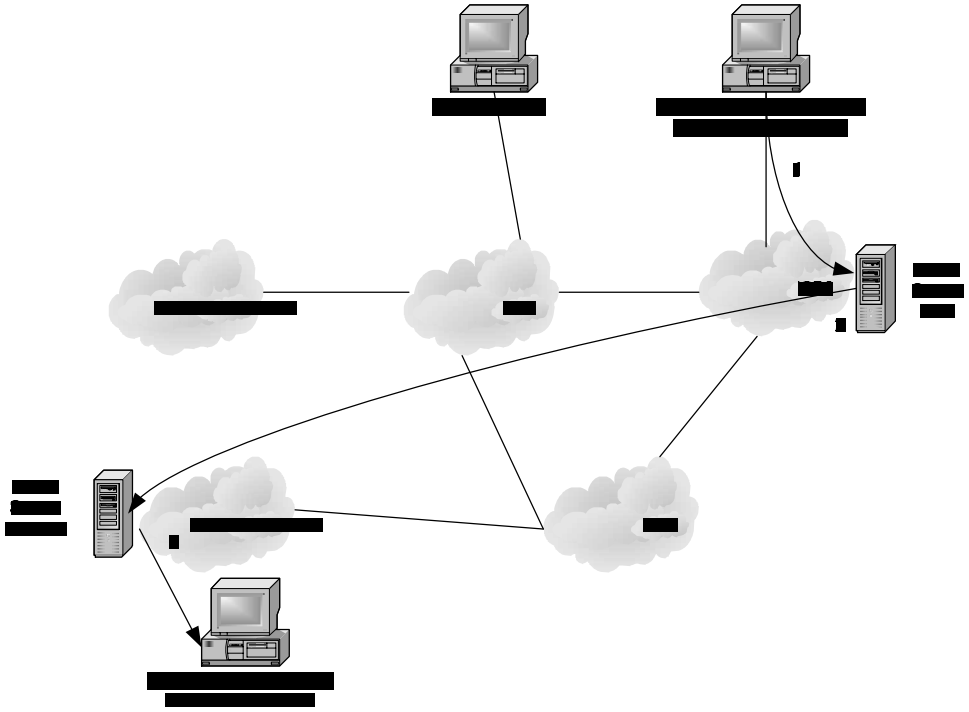
2.1.1 Pengertian *Electronic Mail*

Electronic Mail atau dapat disebut dengan E-Mail, merupakan salah satu layanan Internet yang sangat popular dan paling banyak digunakan oleh orang banyak, baik di lingkungan organisasi maupun perusahaan. E-Mail digunakan untuk saling bertukar informasi atau mengirim pesan antara seseorang dengan orang lainnya yang terpisahkan oleh jarak dan kondisi cuaca apapun dengan melewati perangkat telekomunikasi.

E-Mail beroperasi seperti halnya surat kertas dengan layanan pos (dikenal dengan *snail mail*). Seseorang dapat menulis pada kertas dan menempatkannya pada amplop. Jika orang tersebut membubuhkan nama dan alamat yang benar di depan amplop dan menempatkannya dalam kotak surat, maka orang tersebut dapat mengharapkan layanan pos mengirimkan surat ke tempat yang benar. Jika orang tersebut menempatkan alamat pengirim di amplop, penerima dapat membalas dengan menempatkan alamat orang tersebut di depan amplop sebagai tujuan surat tersebut dikirim.

Sama halnya dengan sistem pengiriman E-Mail, namun tidak menggunakan kertas melainkan menggunakan suatu aplikasi berbentuk program

komputer, dengan media komunikasi jaringan sebagai contohnya seperti Gambar 2.1 dibawah ini.



Gambar 2.1 E-Mail Dalam Internet Mengunakan Mail Server

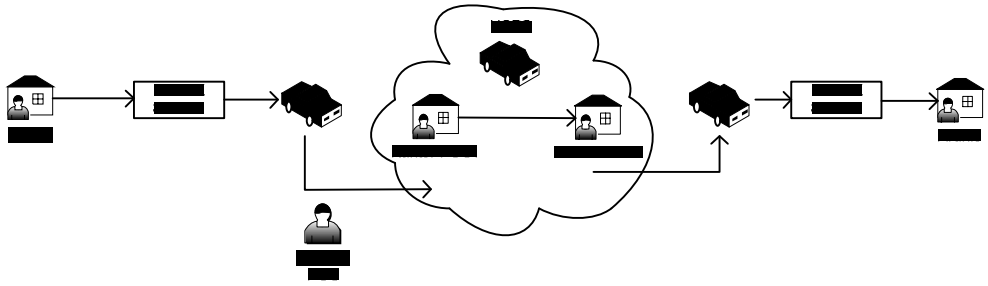
E-Mail memungkinkan seseorang menuliskan beberapa teks, mengidentifikasi siapa saja yang ingin orang tersebut kirimi dengan menuliskan alamat E-Mail seseorang di bagian atas, dan kirimkan ke alamat tersebut. Dengan mengirimkan E-Mail, seseorang dapat mengerjakan hal yang sama seperti mengirimkan E-Mail kepada layanan pos. Layanan ini mengirim E-Mail dan kemudian orang yang dikirim pesan mencek E-Mailnya, kemudian dia menerima pesan yang telah orang lain kirimkan.

Tiap E-mail mencakup teks yang seseorang tuliskan, alamat E-Mail seseorang. Karena kini penerima mengetahui alamat E-Mail seseorang yang

mengiriminya pesan, dia dapat dengan mudah membalas E-Mail tersebut. E-Mail juga memampukan seseorang untuk mengirim pesan ke banyak penerima sekaligus, mengirim *file* menggunakan E-mail dari pada menggunakan program transfer *file.* Rata-rata pesan dalam *mail* tidak mencapai sepuluh *kilobyte* dan beberapa pesan mengandung beberapa *megabyte* data, karena digunakan untuk mengirim *file*.

2.1.2 Sistem Pengiriman Pesan E-Mail

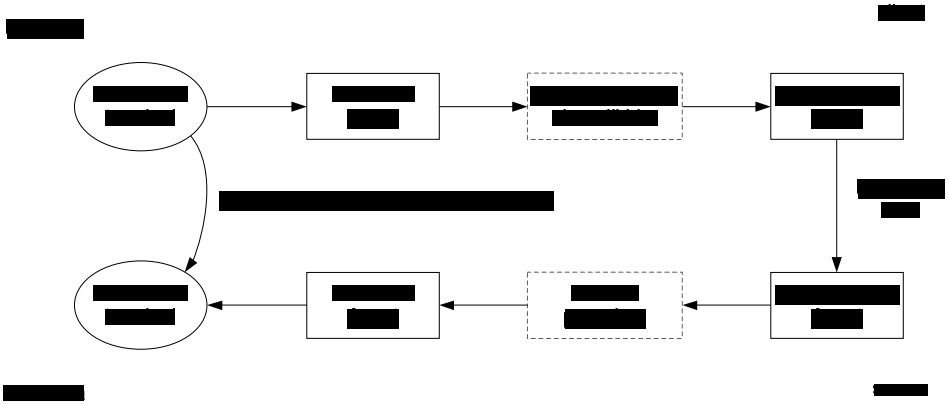
Pada umumnya sistem pengiriman pesan pada E-Mail sama seperti halnya surat kertas dengan layanan pos (dikenal dengan *snail mail*).



Gambar 2.2 Sistem Pengiriman Surat

Namun pada E-Mail tidak menggunakan kertas melainkan menggunakan suatu aplikasi berbentuk program komputer sebagai medianya. E-Mail selalu memanfaatkan standar TCP/IP yaitu menggunakan IMF (*Internet Message Format*) untuk menentukan header yang digunakan untuk mengenkapsulasi teks E-Mail, termasuk pengiriman E-mail SMTP (*Simple Mail Transport Protocol*) dan pembacaannya menggunakan protokol POP (*Post Office Protocol*) / IMAP

(*Internet Mail Access Protocol*) karena untuk mendapatkan pesan, maka sebelumnya *account* E-mail kita harus terdafarkan dulu di server mail yang kita kontak. Kemudian menentukan dulu protokol bagi *client* untuk mendapatkan E-Mail dari *server* yang kita kontak. Sebagai media komunikasi datanya proses pengiriman data, dari satu komputer ke komputer yang lain dengan bantuan protokol diatas dan aplikasi *mail server*. karena *mail server* adalah aplikasi yang mengatur lalu lintas dari pesan E-Mail yang dikirim dan Pesan E-Mail yang diterima, serta *mail server* memiliki *mailbox* tujuan dari pesan tersebut. Seperti proses pertukaran E-Mail yang terjadi dibawah ini.



Gambar 2.3 Komponen Konseptual Sistem E-Mail

Pemakai di terminalnya berhubungan dengan *Mail User Agent* (MUA). Beberapa *agent* E-Mail yang popular antara lain: Pine, Pegasus, dan Eudora. Pertukaran *mail* menggunakan TCP dilakukan oleh *Message Transport Agent* (MTA). MTA yang paling umum untuk *Windows* adalah MDaemon dan untuk Unix menggunakan *qmail* dan *sendmai*. Pemakai awam biasanya tidak berhubungan dengan MTA ini. Ini adalah tanggung jawab administrator *server mail* untuk

mengatur MTA. Sebagaimana dengan pengiriman pesan E-Mail kedua aplikasi ini memiliki fungsi, diantaranya yaitu fungsi dari MUA adalah sebagai berikut:

1. Menulis dan membaca pesan E-mail yang masuk baik secara *online* dan *offline* karena MUA adalah aplikasi *client* E-Mail contoh seperti: *Microsoft Outlook, Thunderbird, Eudora, Pine* dan *Pegasus*.
2. MUA dapat beroperasi tanpa perlu berkomunikasi.
3. Kemudian *user* pengguna E-mail *client* dapat melakukan penyesuaian konfigurasi E-mail sesuai dengan konfigurasi MTA yang ada pada *server mail* .

Sedangkan fungsi dari MTA adalah sebagai berikut:

1. Menentukan *header* yang digunakan untuk untuk mengenkapsulasi *teks* E-Mail, termasuk pengiriman dan penerimaan E-mail *Address.*
2. Menentukan protokol untuk mentransmisikan dan menerima E-mail
3. Menentukan protokol bagi *client* untuk memperoleh E-mail dari *server.*

2.1.3 Protokol dan Standar E-Mail

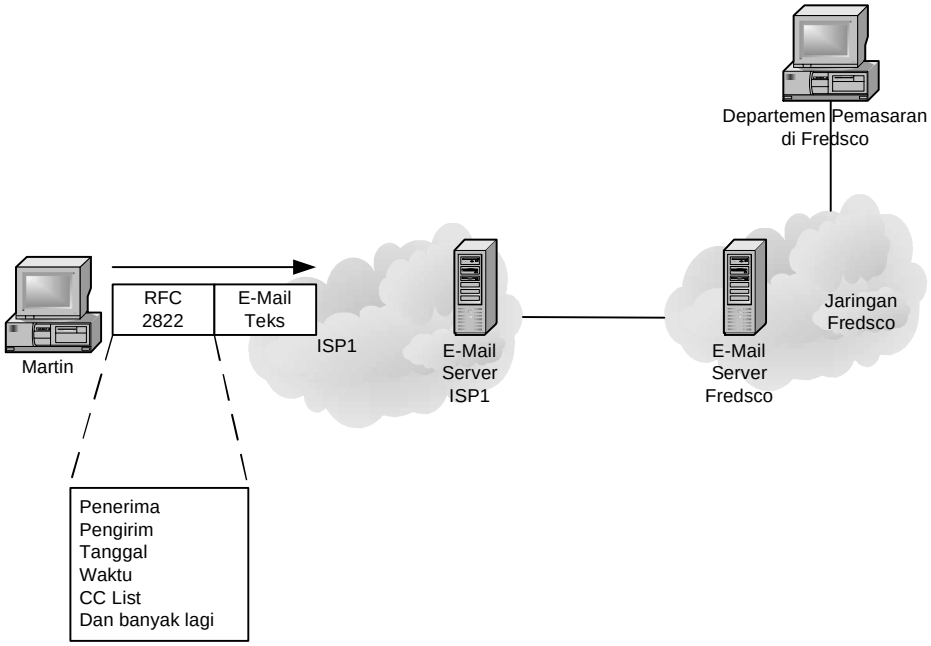
E-Mail memiliki protokol dan standar yang sering dipakai dalam pengiriman dan pembacaan pesan, agar pesan tersebut sampai tujuan. Standar dan Protokol utama yang digunakan dalam layanan E-mail adalah:

1. RFC 2822 atau *Internet Message Format* (IMF).
2. RFC 2821 atau *Simple Mail Transport Protocol* (SMTP).
3. RFC 1939 atau *Post Office Protocol* versi tiga (POP3).

Selain protokol POP3 yang digunakan dalam medapatkan pesan dari E-mail *server* adapun protokol yang memiliki fungsi sama yaitu *Internet Mail Access Protocol* (IMAP).

2.1.3.1 Internet Message Format (IMF)

Saat *client* mengirim E-Mail dan E-mail *server* meneruskannya, mereka tidak hanya mengirimkan teks E-Mail *message* (teks pesan E-Mail), tetapi juga mengirimkan *header*. *Header* terdiri dari beberapa *field*, termasuk E-Mail *address* dari penerima dan pengirim, seperti di gambarkan pada Gambar 2.4.



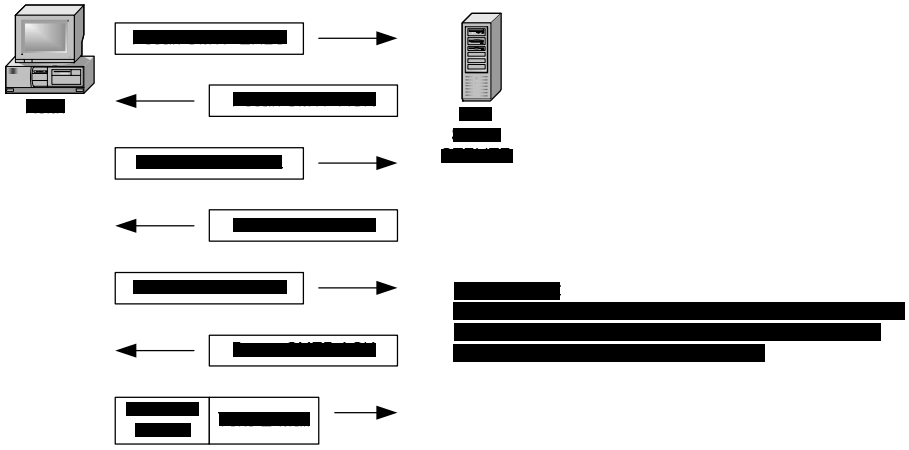
Gambar 2.4. Sampling E-Mail *Header* yang ditetapkan oleh RFC 2822

RFC 2822, "*Internet Message Format*" menetapkan format Internet E-Mail *Message* (format pesan Internet E-Mail) namanya adalah *bit self-defining*. RFC 2822 menetapkan *header field* yang ditunjukan pada Gambar 2.4, dan lebih

banyak lagi. Sehingga *client* dan *server* mengetahui dimana mencari semua informasi yang dibutuhkan untuk mengirim E-Mail dengan Benar.

2.1.3.2 Simple Mail Transport Protocol (SMTP)

Salah satu protokol E-mail yang populer juga berusaha membuat segala sesuatunya sederhana adalah *Simple Mail Transport Protocol* (SMTP). Karena itu SMTP dapat dikatakan sebagai prinsip KISS yang merujuk pada pepatah bijak yang berbunyi *Keep it Simple, Stupid* (buat sederhana saja). Jadi *client* E-mail dan *server* menggunakan protokol SMTP dengan menggunakan port 25 untuk mengatur proses pengiriman dan penerimaan E-Mail. Sebagaimana yang telah dibahas, daripada hanya mengirim E-Mail dengan RFC2822 *header* yang benar disekitarnya, *client* E-Mail dan *server* harus berkomunikasi terlebih dahulu. Misalnya, *client* harus mengidentifikasi dirinya sendiri ke *server* sebelum mengirimkan sebuah E-Mail. Selain itu pengirim E-mail harus mengidentifikasi penerima sebelum mengirim E-Mail. dengan cara tersebut, jika *client/server* berikutnya tidak dapat mengirim E-Mail, Seseorang tidak membuang waktu mengirim E-Mail lagi. Untuk penekanan maka dapat dilihat pada Gambar 2.5



Gambar 2.5 SMTP Mengidentifikasi *client* dan Penerima

Protokol SMTP menetapkan pesan sehingga *client* dan *server* E-mail dapat mengatur proses pengiriman E-Mail. *Extended Hello* (EHLO) *command* mengidentifikasi *client.* Mail *command* memberitahukan *server* bahwa *client* ingin mengirim atau menerima E-Mail. RCPT (*short for recipient*) mengidentifikasikan penerima E-Mail yang akan dikirimi. Sesudah server membalas dengan pesan *acknowledgement* (ACK) pada tiap-tiap tiga *command* pertama ini, E-Mail dapat ditransmisikan. Perlu dicatat bahwa sebenarnya saat semua perkakas mengirim E-Mail, pesan tersebut diselaraskan dengan spesifikasi *Internet Message Format* (RFC 2822) untuk isi *header* dan teks. SMTP RFC (RFC 2821) menetapkan pesan-pesan yang mengontrol proses dengan SMTP seperto EHLO dan MAIL.

2.1.3.3 Post Office Protocol (POP3)

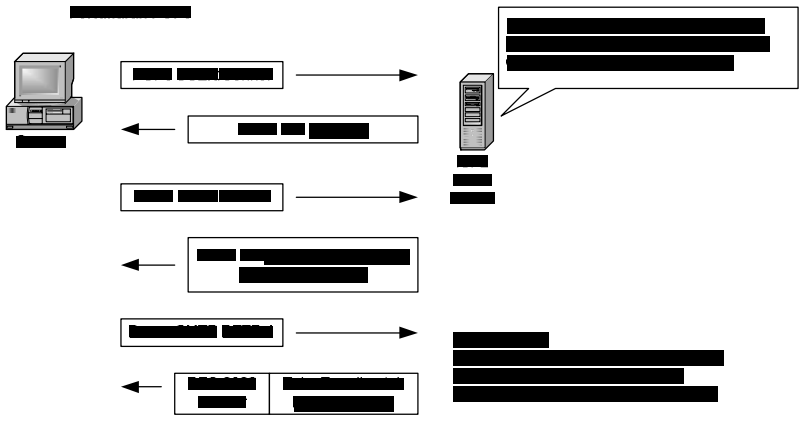
Protokol E-Mail terakhir yang berikutnya yaitu *Post Office Protocol* (POP3) yang berjalan di port 110, versi saat ini, POP3 yang memungkinkan

seseorang untuk mentransfer E-Mail, seperti SMTP. Akan tetapi POP3 didesain secara khusus untuk pengunnaan antara *client* E-mail dan *mail server* tetapi dia tidak dapat mentransfer E-Mail antar *server*. POP3 pada dasarnya memungkinkan seseorang untuk mengerjakan subset fungsi dari SMTP. Jadi, mengapa bersusah-susah menggunakan protokol yang lain saat SMTP memungkinkan *client* untuk mendapatkan E-Mail mereka? Memang, POP3 memungkinkan beberapa opsi untuk otentifikasi, dan beberapa fitur berguna lainnya untuk *client* E-Mail fitur-fitur yang tidak dimasukan dalam SMTP.

Salah satu fitur yang lebih baik dari POP3 adalah otentifikasi. Otentifikasi mengacu pada proses sebuah perkakas untuk mengidentifikasi dirinya sendiri dengan perkakas yang lain melalui nama dan password perkakas lain untuk selanjutnya memutuskan apakah perkakas pertama bisa mengakses. Misalnya, jika seseorang memiliki E-Mail *account* pada *Internet Service Provider* (ISP) yang dia punya, dalam beberapa hal seseorang di beri E-Mail *address* dan *password*. Saat orang itu memeriksa E-Mail, jika orang tersebut menggunakan POP3 (dan jika menggunakannya), *client* E-Mail tersebut mengidentifikasi dirinya sendiri dan menyediakan password. E-Mail *server* memutuskan apakah mugkin orang tersebut mendapatkan E-Mail. Gambar 2.6 menekankan pada proses POP3.

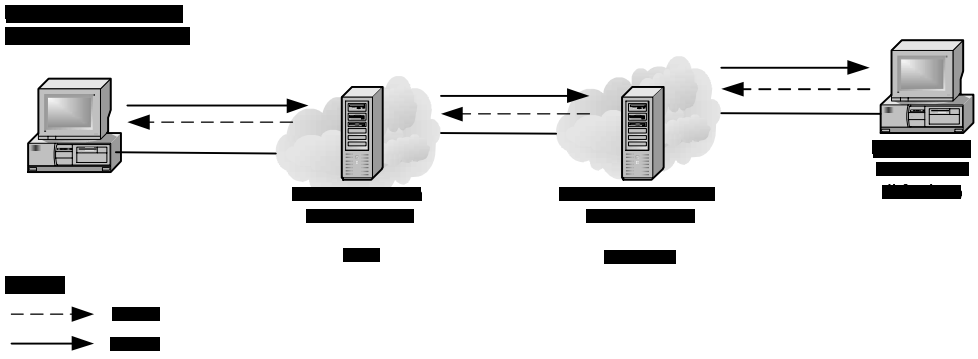
Seperti yang dapat Anda lihat pada Gambar 2.6. Conner mengirim *username* dan *password* ke *server*. *Server* memeriksa daftar *username* dan *password* dan mencari kecocokan. Jadi, *server* memberitahu *client* "OK" berarti *server* telah siap memulai pengiriman E-Mail ke *client*. POP3 juga memberitahu

user berapa banyak E-Mail yang menunggu dia. Untuk mendapatkan E-Mail, *client* meminta E-Mail sekali waktu seperti ditunjukan pada gambar.



Gambar 2.6 Otentifikasi Dasar dengan POP3

Dalam suatu jaringan E-mail, kedua protokol SMTP dan POP3 selalu bekerja sama dalam meyampaikan pesan yang akan dikirim melalui *server. Client* menggunakan POP3 untuk mendapatkan E-Mail dari server mereka, tetapi mereka menggunakan SMTP untuk mengirim E-Mail ke server lain. Gambar 2.7. menggambarkan aliran ini.



Gambar 2.7 Penggunaan SMTP dan POP3

Dari kejadian diatas dapat diketahui bahwa dalam gambaran ini. POP3 hanya digunakan untuk mengambil E-Mail dari *server*. Selain itu catat bahwa *server* E-Mail dilabelkan sebagai SMTP *server* dan POP3 *server*. Merujuk pada *server* E-Mail pada umumnya dalam kenyataannya, untuk mengerjakan SMTP, seseorang membutuhkan *server software* yang memahami SMTP, jadi supaya semua ini bekerja, *server* perlu memiliki SMTP *server software* untuk menjalankannya. Serupa dengan itu, untuk memungkinkan agar *client* menerima *mail, server hardware* harus mengoperasikan POP3 *server software* untuk SMTP *server software.* Efeknya, komputer harus menjalankan kedua *server*. SMTP server dan POP3 server pada komputer yang sama menyimpan dan memperoleh E-Mail dari lokasi penyimpanan pesan yang sama. Misalnya pada Gambar 2.7, E-Mail *server* Fredsco menerima E-Mail menggunakan SMTP *server*, menyimpannya pada *disk drive*. Selanjutnya, saat orang yang menggunakan *account* sales@fredsco.com memeriksa E-Mailnya dengan POP3, POP3 *server* pada komputer yang sama, menerima E-Mail, mengirimkanya kepada komputer *client.*

2.1.4 Komponen E-Mail

Secara umum Elektonik Mail terdiri atas tiga komponen yang selalu digunakan dalam pengiriman dan penerimaan pesan E-Mail, yaitu:

2.1.4.1 Envelope

Envelope, atau amplop, ini digunakan oleh MTA untuk melakukan pengiriman. *Envelope* ditandai dengan dua buah perintah SMTP:

Mail From: <angga@mail.net>

RCPT To: <senno@box.net>

Isi dan interpretasi dari envelope SMTP ditentukan di RFC 821. RFC ini digunakan untuk mengirim E-Mail melalui koneksi TCP.

2.1.4.2 Header

Header, digunakan oleh *user* agent sebagai identitas yang menyimpan pengalamatan dari E-Mail tersebut. *Header*, berisi *field-field* seperti: *received, Message-Id, From, Date, Reply To, X-Phone, X-Mailer, TO, Subject*. Setiap *field* header berisi sebuah nama yang diikuti oleh sebuah titik dua (:), dan nilai dari *field header* tersebut. Format dan interpretasi atas field header ini ditentukan dalam RFC 822. Field header yang panjang, seperti *Recevied,* akann dilipat ke dalam beberapa baris, dengan tambahan sebuah spasi kosong didepannya.

2.1.4.3 Body

Body merupakan isi pesan dari pengirim ke penerima dalam RFC 822 disebutkan bahwa body ini merupakan baris-baris dalam bentuk teks NVT ASCII. Setiap baris yang dikirim menggunakan perintah DATA, tidak boleh melebihi 1024 byte.

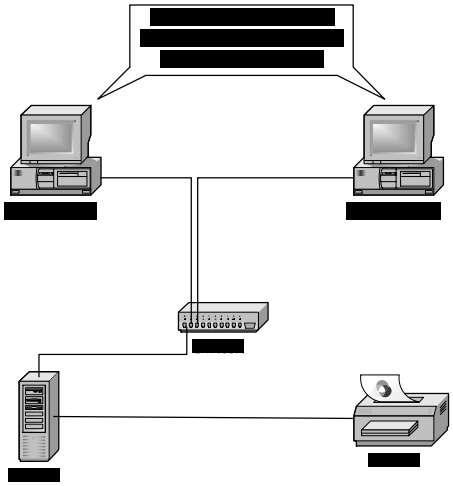
2.2 Jaringan Komputer

Jaringan komputer yaitu sekumpulan atau kombinasi yang terdiri dari:

1. Perangkat keras, seperti komputer, printer, switch, router dan hub.

2. Perangkat lunak, seperti sistem operasi, novel dan aplikasi lainnya.
3. Pengkabel (*cabling*), seperti kabel UTP dan konektor RJ45 sebagai kabel media komunikasinya.
4. Tanpa pengkabelan, seperti *wireless router* sebagai kabel media komunikasinya.

Dengan adanya kombinasi perangkat keras, perangkat lunak, kabel dan tanpa kabel maka dapat memungkinkan berbagai alat komputasi berkomunikasi satu sama lain. Singkatnya, jaringan memampukan komputer untuk berkomunikasi akan tetapi, definisi tersebut dapat diaplikasikan pada banyak arsitektur dan tipe jaringan yang berbeda-beda. Jadi, beberapa contoh dapat membantu. Gambar 2.8 menunjukan ide dasar mengenai jaringan sederhana yang digunakan oleh perusahaan kecil pada satu situs tunggal.



Gambar 2.8 Jaringan Sederhana

2.2.1 Dasar Arsitektur Jaringan Komputer

Pada dasarnya, Arsitektur jaringan di bagi menjadi beberapa macam menurut fungsi dan jangkauan yang diperlukan untuk melakukan komunikasi antar perangkat, diantaranya yaitu:

2.2.1.1 Local Area Network (LAN)

Jaringan yang meliputi suatu area geografis yang relatif kecil (didalam satu lantai atau gedung kecil). Dibandingkan dengan WAN, LAN biasanya dicirikan dengan kecepatan data relatif tinggi dan kecepatan *error* yang relatif rendah. LAN mengubungkan *workstation*, perangkat jaringan, terminal, dan perangkat lain dalam area yang terbatas. Standar LAN menentukan pengkabelan dan pensinyalan pada layer fisik dan layer *data link* dalam model OSI. *Ethernet*, FDDI, dan Token Ring menggunakan teknologi LAN. Kebanyakan LAN dibangun dengan perangkat keras yang relatif murah seperti ethernet, adaptor jaringan, dan hub. Juga ada LAN nirkabel (*wireless*) dan pilihan perangkat keras LAN lain yang lebih maju. Perangkat lunak sistem operasi khusus bisa digunakan untuk menyusun LAN. Misalnya sebagian besar karakter khas Microsoft Windows menyediakan paket perangkat lunak yang dinamakan *Internet Connection Sharing* (ICS) yang mendukung akses terkendali ke sumber-sumber LAN.

2.2.1.2 Metropolitan Area Network (MAN)

Pada dasarnya jaringan MAN sama halnya dengan LAN. Tetapi MAN memiliki jaringan yang berukuran lebih besar, MAN dapat mencakup kantor kantor perusahaan yang letaknya berdekatan atau komunikasi jaringan antara kota satu dengan kota lainnya. MAN dapat dimanfaatkan untuk keperluan pribadi atau umum.

2.2.1.3 Wide Area Network (WAN)

Sebuah jaringan yang biasanya dibangun menggunakan jalur serial, yang mencakup area geografis yang luas seperti komunikasi jaringan antar negara satu dengan negara yang lainnya atau antar benua.

2.2.1.4 Internet

Jaringan komputer global yang memungkinkan dua komputer atau lebih berkoneksi dengannya untuk mentransfer file dan tukar-menukar E-Mail dan pesan-pesan real-time. Internet merupakan landasan untuk World Wide Web. Internet juga merupakan kumpulan jaringan komputer yang berbeda-beda dan saling berhubungan di seluruh dunia. Semua komputer di hubungkan dengan Internet agar dapat berkomunikasi satu sama lain dengan menggunakan *Transmission Control Protocol/Internet Protocol*.

2.2.1.5 Jaringan Tanpa Kabel (Wireless Network)

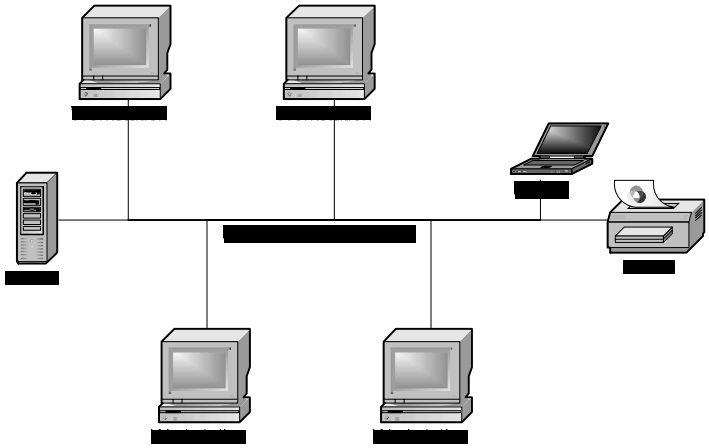
Jaringan tanpa kabel dengan menggunakan gelombang radio dan / atau gelombang mikro untuk membentuk kanal komunikasi antar komputer. *Wireless network* adalah alternatif yang lebih modern terhadap jaringan berkabel yang bergantung pada kabel tembaga dan atau serat optik antar perangkat jaringan. Dengan memanfaatkan frekwensi gratis 2,4 GHz yang disebut kanal ISM (*Industrial Scientific and Medical*), yaitu band frekwensi yang biasanya dipakai untuk perangkat-perangkat di industrial, penelitian dan kedokteran. Selain itu dalam melakukan komunikasi pun teknologi ini menjadi salah satu solusi yang dapat dipertimbangkan dalam hal kecepatan akses.

2.2.2 Topologi Jaringan Komputer

Topologi jaringan merupakan cara bagimana memperlihatkan komputer-komputer yang tersambungan di antara komputer satu dengan komputer lainnya dalam suatu jaringan. Layer jaringan harus mengikuti topologi jaringan saat ini diantaranya yaitu topologi bus, token ring, star, *peer to peer* dan *hybrid*.

2.2.2.1 Topologi Bus

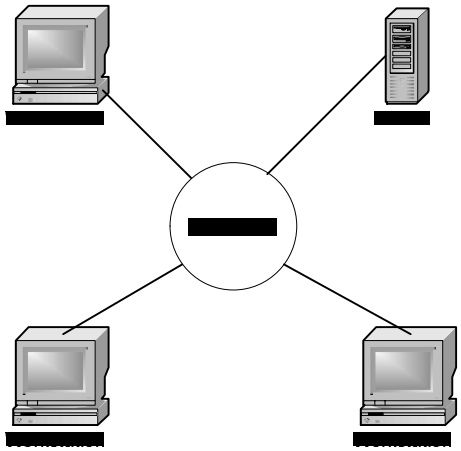
Topologi bus merupakan topologi yang paling sederhana dan paling mudah dalam instalasi secara fisik, ciri topologi bus adalah hanya memiliki satu jalur utama yang disebut dengan *backbone*. Setiap *host* (peralatan yang terkoneksi kejaringan) dapat dikoneksikan secara langsung ke *backbone*.



Gambar 2.9 Topologi Bus

2.2.2.2 Topologi Token Ring

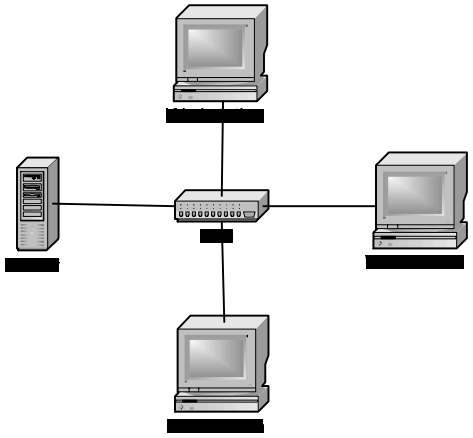
Metode token ring merupakan cara menghubungkan komputer sehingga berbentuk ring (lingkaran), yang mana setiap simpulnya memiliki tingkatan yang sama, jaringan akan disebut sebagai loop, data dikirimkan kesetiap simpul dan setip informasi yang diterima simpul diperiksa alamatnya apakah data itu untuknya atau bukan. Sebagai penekan lihat Gambar 2.10



Gambar 2.10 Topologi Token Ring

2.2.2.3 Topologi Star

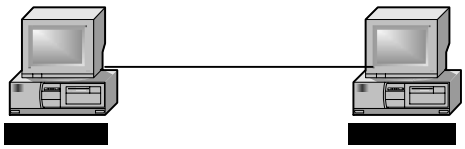
Topologi LAN dimana titik akhir pada suatu jaringan dikoneksikan ke suatu *switch* sentral bersama melalui sambungan *point-to-point*. Suatu topologi ring yang diatur sebagai topologi *star* mengimplementasikan *unidirectional closed-loop star* dan bukannya sambungan *point-to-point.*



Gambar 2.11 Topologi Star

2.2.2.4 Topologi Peer to Peer

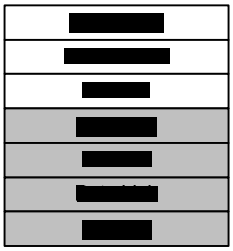
Konfigurasi jaringan di mana komputer-komputer terhubung secara langsung satu dengan yang lainnya. Setiap komputer memiliki sendiri hardisk, dan dapat berbagai pakai drive CD-ROM, sistem untuk backup, dan printer. Tidak ada komputer pusat yang berperan sebagai tempat penyimpanan utama.



Gambar 2.12 Topologi Peer-to-Peer

2.2.3 Protokol Jaringan Komputer

Didalam suatu jaringan komputer terdapat suatu protokol yang berkerja pada bagian dan fungsinya masing-masing. Seperti pada layer-layer OSI (*Open Systems Interconnection*) yang telah menjadi standar internasional yang dikembangkan oleh ISO (*International Standarts Oraganization*) untuk keperluan interkoneksi sistem komputer yang kooperatif. Jadi OSI adalah sekumpulan protocol yang dijadikan satu, dirancang oleh komite ISO agar menjadi standar internasional arsitektur jaringan komputer. OSI terdiri atas struktur tujuh layer. Ke tujuh layer tersebut dapat dijabarkan pada Gambar 2.13.

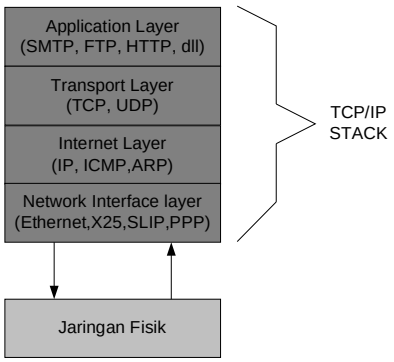


Gambar 2.13 Layer OSI

Berkaitan dengan penulisan tugas akhir tentang pengamanan pesan pada E-Mail maka penulis hanya membahas protokol yang berkenaan dengan pengiriman E-Mail yaitu protokol TCP/IP.

Transport Control Protocol (TCP) berperan dalam memperbaiki pengiriman data dari *client* ke *server.* TCP dapat mendeteksi *error* atau data yang hilang dan kemudian melakukan transmisi ulang sampai data tersebut benar-benar di terima dengan lengkap. Sedangkan *Internet Protocol* (IP) berfungsi untuk meyampaikan paket data ke alamat yang tepat dengan mendahului setiap paket data berdasarkan alamat tujuan. *Internet Authorities* menciptakan range angka

untuk organisasi yang berbeda. Jadi TCP/IP adalah sekumpulan protokol yang didesain untuk melakukan fungsi–fungsi komunikasi data pada LAN, MAN, WAN. TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Sekumpulan protokol TCP/IP ini dimodelkan dengan empat layer TCP/IP, sebagaimana terlihat pada Gambar 2.14.



Gambar 2.14 Layer TCP/IP

TCP/IP terdiri atas empat lapis kumpulan protokol yang betingkat. Keempat lapis/layer tersebut adalah:

1. *Network Interface Layer*
2. *Internet Layer*
3. *Transport Layer*
4. *Application Layer*

2.3 Keamanan Jaringan Komputer

Saat ini pemakaian jaringan komputer dari *Local Area Network* sampai jaringan *Internet* bukan merupakan hal yang aneh lagi hampir semua bidang usaha

dan semua informasi telah menggunakan dan dapat diakses lewat jaringan komputer, dalam kondisi yang demikian keamanan atau *security* terhadap jaringan merupakan hal yang sangat penting dan saat ini telah menjadi masalah yang menarik dan aktual sejalan dengan pesatnya perkembangan internet dan aplikasi serta kegiatan yang ada di dalamnya, baik itu transaksi keuangan maupun transaksi informasi yang bersifat penting .

Beberapa alternatif muncul untuk mengamankan data dan informasi yang melewati jaringan internet tersebut seperti pemasangan *Firewall* autentifikasi lewat password, maupun proses enkripsi dari data tersebut.

2.3.1 Aspek Keamanan Jaringan Komputer

Keamanan komputer jaringan melingkupi empat aspek utama yaitu privacy, integrity, autentication dan availabilty serta dua aspek lain yang erat kaitannya dengan keamanan komputer yang berbasis jaringan yaitu access control dan *non-repudiation*.

2.3.1.1 Privacy / Confidentiality

Aspek *Privacy* merupakan usaha untuk menjaga informasi dari orang yang tidak berhak mengakses suatu sistem, dan lebih kearah data – data yang sifatnya privat sedangkan *Confidentiality* berhubungan dengan data yang diberikan ke pihak lain untuk keperluan tertentu misalnya sebagai bagian dari pendaftaran sebuah servis dan hanya diperbolehkan untuk keperluan tertentu tersebut. Contoh hal yang berhubungan dengan *privacy* adalah E-mail seorang pemakai tidak boleh

dibaca oleh seorang *administrator* sedangkan contoh *confidential information* adalah data data yang sifatnya pribadi seperti nama, tempat tanggal lahir, agama, status perkawinan, nomor kartu kredit dan sebagainya yang merupakan data – data yang ingin diproteksi penggunaanya dan penyebarannya. Serangan terhadap aspek *privacy* misalnya adalah usaha untuk melakukan penyadapan (dengan program *snifer*) dan usaha-usaha yang dapat dilakukan untuk meningkatkan *privacy* dan *confidentiality* misalnya dengan menggunakan teknologi kriptografi.

2.3.1.2 Integrity

Integrity lebih menekankan bahwa informasi tidak boleh diubah tanpa seijin pemilik informasi, adanya virus, trojan, atau pemakai lain yang mengubah informasi tanpa ijin merupakan contoh masalah yang sering dihadapi, sebagai contoh misalnya sebuah E-Mail dapat saja ditangkap di tengah jalan yang kemudian dimodifikasi dan diteruskan ke alamat yang dituju dengan kata lain integritas dari informasi tersebut sudah tidak terjaga lagi, penggunaan enkripsi dan tanda tangan digital misalnya dapat mengurangi atau bahkan bisa mejadi salah satu solusi untuk mengatasi masalah tersebut.

2.3.1.3 Authentication

Aspek ini berhubungan dengan metode untuk menyatakan bahwa informasi benar benar asli, orang yang mengakses atau memberikan informasi adalah benar benar orang yang dimaksud atau *server* yang kita hubungi adalah benar benar *server* yang asli. Masalah pertama yang muncul membuktikan

keaslian dokumen, dapat dilakukan dengan teknologi watermarking dan tanda tangan digital, masalah selanjutnya biasanya berhubungan dengan access control yaitu berkaitan dengan pembatasan orang yang dapat mengakses informasi dalam hal ini pengguna harus menunjukan bukti bahwa memang dia adalah pengguna yang sah, misalnya dengan menggunakan password atau ciri – ciri khusus yang berhubungan dengan pemilik/pengguna yang sah seperti ATM, PIN, Sidik jari, Biometric dan lain – lain.

2.3.1.4 Availability

Availability berhubungan dengan ketersediaan informasi ketika dibutuhkan, sistem informasi yang diserang atau dijebol dapat menghambat atau atau meniadakan akses ke informasi salah satu contohnya adalah serangan DOS (Denial of Service) dimana *server* biasanya dikirimi *request* yang berkelanjutan atau permintaan di luar perkiraan sehingga tidak dapat melayani permintaan lain atau bahkan sampai *down*, *hang*, *crash.*

2.3.1.5 Access Control

Access Control berhubungan dengan cara pengaturan akses kepada informasi dan biasanya berhubungan dengan klasifikasi data (*public*, *private*, *confidential*, *top secret*) dan *user* (*guest*, *admin*, *top manager* dan sebagainya), *access control* seringkali dilakukan dengan menggunakan kombinasi *user-id/password* atau dengan menggunakan mekanisme lain (kartu, biometrik).

2.3.1.6 Non-Repudiation

Aspek ini menjaga agar seseorang tidak dapat menyangkal telah melakukan transaksi, sebagai contoh seseorang yang mengirimkan email untuk memesan barang tidak dapat menyangkal bahwa dia telah mengirimkan email tersebut. Penggunaan tanda tangan digital, sertifikat, dan teknologi kriptografi secara umum dapat menjaga aspek ini dan tentu saja harus didukung dengan aspek hukum sehingga status dari *digital signature* itu jelas dan legal.

2.3.2 Klasifikasi Keamanan Jaringan Komputer

Suatu jaringan komputer yang terdiri dari beberapa komputer yang saling terhubung satu dengan yang lainnya serta melewati beberapa titik/hub yang menghubungkan satu dengan lainnya dan dengan berbagai karakter alat serta *tool* yang digunakan menyebabkan rentannya terhadap keamanan jaringan tersebut berbagai ancaman muncul untuk menggerogoti data dan informasi yang ada, metode dan konsep pengamananpun muncul dan digunakan untuk memproteksi data dari adanya pencurian ataupun manipulasi oleh pihak - pihak yang tidak berhak dan dapat merugikan pemilik informasi tersebut, dimana konsep dan metode pengamanan tersebut dapat diklasifikasikan menjadi empat, yaitu:

2.3.2.1 Keamanan yang Bersifat Fisik *(Physical Security)*

Keamanan yang bersifat fisik termasuk akses orang ke gudang, peralatan dan media yang digunakan, beberapa *crackers* mengatakan bahwa mereka sering pergi ketempat sampah untuk mencari berkas - berkas yang mungkin memiliki

informasi tentang keamanan atau hal hal yang berhubungan dengan akses ke kabel atau komputer yang digunakan.

2.3.2.2 Keamanan yang Berhubungan dengan Orang *(Personel)*

Keamanan yang berhubungan dengan orang termasuk identifikasi dan profil dari orang yang mempunyai akses. Seringkali kelemahan keamanan sistem informasi bergantung kepada manusia (pemakai dan pengelola),. Teknik *social engineering* merupakan teknik yang digunakan oleh kriminal untuk berpura – pura sebagai orang yang berhak untuk mengakses informasi.

2.3.2.3 Keamanan Data dan Media Komunikasi *(Communication)*

Keamanan data dan media komunikasi pengamanan dari software yang digunakan untuk mengelola informasi / data, kehandalan software yang digunakan merupakan aspek penting dalam pengelolaan informasi, seorang kriminal dapat memasang virus atau trojan horse sehingga dapat mengumpulkan informasi seperti password yang semestinya tidak berhak untu diakses.

2.3.2.4 Keamanan dalam Operasi

Keamanan dalam operasi termasuk prosedur yang digunakan untuk mengatur dan megelola sistem keamanan dan juga termasuk prosedur setelah serangan.

2.3.3 *Firewall*s

Secara sepintas, penerapan konsep *firewall* itu terlihat cukup sederhana, yaitu bilamana terjadi suatu aliran trafik yang datang menuju suatu jaringan, perangkat *firewall* kemudian akan melakukan pemeriksaan serta pengontrolan terhadap trafik tersebut untuk selanjutnya dikirimkan menuju tujuannya. Jadi definisi dari *firewall* adalah suatu perangkat keamanan jaringan yang memperkenankan berbagai bagian ruas jaringan untuk melaksanakan komunikasi antara satu sama lainnya sesuai dengan definisi kebijakan keamanan (*Security policy*) yang telah ditetapkan sebelumnya.

Firewall bisa dikatakan sama fungsinya dengan *router* jika ditempatkan dalam konteks sebagai perangkat yang melaksanakan interkoneksi sebagai ragam ruas jaringan secara bersamaan. Perangkat lunak *firewall* berjalan pada sebuah host yang akan mengkoneksikan beragam jaringan yang memiliki berbagai tingkat keamanan (seperti jaringan internal dengan Internet). *Firewall* berbeda dengan *router* dalam konteks kemampuan mereka menyediakan mekanisme keamanan dalam rangka memperkenankan atau menolak beragam trafik, seperti aktifitas autentifikasi, enkripsi, keamanan isi atau muatan (*Content Security*) serta translasi alamat. Meskipun banyak router yang mampu menyediakan kemampuan sejenis (contohnya perangkat high-end keluaran Cisco). Fungsi utama dari *firewall* adalah menyeleanggarakan kebijakan keamanan terhadap suatu jaringan tertentu dan memang perangkat ini sengaja didesain untuk mencegah akses yang tidak diinginkan dari atau ke dalam suatu jaringan internal. *Firewall* dapat dibagi atas tiga bagian diantaranya yaitu:

2.3.3.1 Teknologi Penyaringan Paket (*Filtering Packets*)

Teknologi penyaringan paket dapat memeriksa setiap paket data yang masuk maupun yang keluar dari suatu jaringan dan menerima ataupun menolak paket-paket data tersebut berdasarkan pada aturan yang telah ditentukan sebelumnya. Teknologi ini dapat bekerja hingga layer *transport*. Kemudian dengan adanya teknologi ini tidak membutuhkan memori dan proses tambahan yang besar dalam menjalankan aktivitasnya, sehingga *router* dengan kemampuan rendah pun diharapkan dapat mengatasi beban trafik skala menengah dengan cukup baik.

2.3.3.2 Teknologi Gateway Layer Aplikasi atau (*Proxy*)

Teknologi *gateway layer* aplikasi (*Application Layer Gateways*) atau proxy akan mengimplementasikan *firewall* pada *application layer*. Pada dasarnya jaringan eksternal dapat mengakibatkan terjadinya dinamika perubahan lingkungan jaringan yang secara terus-menerus menawarkan protokol *service* dan aplikasi baru, proxy sudah tidak mampu menangani beragam tipe komunikasi pada jaringan eksternal. Jika dibandingkan dengan teknologi sebelumnya, teknologi proxy cukup meningkatkan tinggal keamanan karena ia dapat memeriksa seluruh layer aplikasi, membawa konteks informasi sampai pada suatu proses keputusan. Disamping itu, setiap *proxy* membutuhkan suatu proses aplikasi, suatu daemon dan skalabilitas tertentu agar dapat mendukung suatu aplikasi baru tertentu sehingga hal ini memunculkan masalah baru tentunya. Seperti telah diterangkan, teknologi ini nyata-nyata telah melanggar konfigurasi

model *client/server* karena teknologi *proxy* ini dalam berkomunikasi membutuhkan dua buah jenis koneksi, yaitu:

1. Satu koneksi dari *client* ke *firewall.*
2. Satu koneksi lagi dari *firewall* tersebut ke *server.*

2.3.3.3 Teknologi Inspeksi Keseluruhan Kondisi (*Stateful Inspection*)

Teknologi Inspeksi keseluruhan kondisi paket (*Stateful Inspection)* menggabungkan beragam fitur terbaik dari teknologi Penyaringan Paket dan teknologi *Gateway Layer* Aplikasi. *Engine* dari teknologi inspeksi keseluruh kondisi paket diletakan di antara layer *Datalink* dan layer *Network* yang dalam kasus ini diletakan di antara kartu interface jaringan dengan driver TCP/IP. Paket-paket yang berasal dari layer network akan diperiksa sesuai dengan kebijakan keamanan (*Security Policy*). Anda sendiri yang masing-masing akan memperoleh izin untuk lewat atau ditolak untuk lewat. *Stack* TCP/IP tidak akan melihat paket-paket yang telah dijatuhkan (*drop*) atau ditolak (*reject*), akan tetapi menyediakan layer proteksi tambahan. Inspeksi seluruh kondisi dapat melihat keseluruhan paket dan membuat suatu keputusan terhadap kebijakan keamanan (*Security Policy*) berdasarkan muatan dan isi paket tersebut.

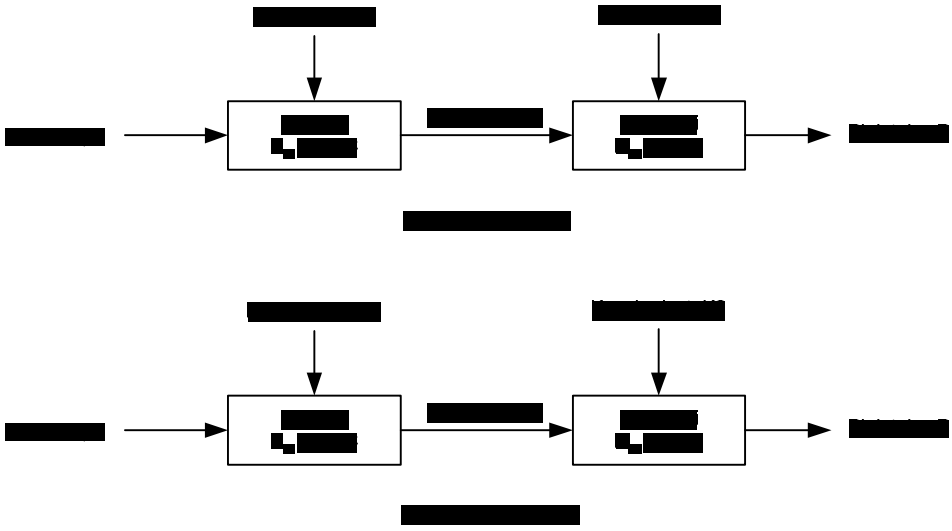
2.3.4 Autentifikasi Password

Autentifikasi lewat password dilakukan untuk membatasi dan mengatur hak akses dari pengguna sistem sehingga tidak sembarangan orang dapat masuk kedalam sistem tersebut, dengan menyamakan *user* id dan password yang

dimiliki program/ sistem dapat mengenali apakah *user* boleh memasuki sistem atau tidak, bila *user* id dan password yang dimiliki *user* sama dengan *user* id dan password yang ada di dalam database maka sistem memperbolehkan *user* untuk masuk kedalam sistem.

2.4 Kriptografi Menggunakan Algoritma CAST-128

Kriptografi adalah imu dan seni untuk menjaga keamanan pesan. Kata "seni" didalam definisi di atas berasal dari fakta sejarah bahwa pada masa-masa awal sejarah kriptografi, setiap orang mungkin mempunyai cara unik untuk merahasiakan pesan.cara-cara unik tersebut mungkin berbeda-beda pada setiap pelaku kriptografi sehingga setiap cara menulis pesan rahasia pesan mempunyai nilai estetika tersendiri sehingga kriptografi berkembang menjadi seni merahasiakan pesan. Dimana ilmu kriptografi sering menemukan berbagai istilah atau terminologi diantaranya yaitu proses penyandian pesan seperti halnya proses enkripsi dan deskripsi. Enkripsi yaitu proses penyandian *plainteks* (pesan biasa) menjadi *cipherteks* (bentuk pesan yang tersandikan) agar pesan tersebut tidak bisa dibaca orang sembarang, sebaliknya jika orang tersebut ingin membaca isi dari *chipertext*, maka ia harus mendeskripsikan pesan tersebut dengan proses pengembaliaan *cipherteks* menjadi *plainteks.* sedangkan berdasarkan kuncinya kriptografi dibagi menjadi dua yaitu kriptografi kunci simetris dan asimetris, sebagai penekanan dapat dilihat pada gambar 2.15.



Gambar 2.15 Kriptografi Simetri dan Asimetri

Secara umum, *cipher* (kunci) yang termasuk ke dalam kriptografi simetris beroperasi dalam mode (*block cipher*), yaitu setiap kali melakukan proses enkripsi dan deskripsi terhadap satu blok data (yang berukuran tertentu) menggunakan kunci yang sama yaitu menggunakan kunci privat. Kunci privat ini hanya boleh dimiliki oleh pihak orang ke satu dan orang kedua. Sedangkan pada cipher (kunci) yang termasuk kriptografi asimetris dalam melakukan enkripsi dan deskripsinya menggunakan kunci yang berbeda, yaitu kunci privat dan kunci publik. Jadi dalam pendistribusiannya, kunci publik dapat berikan kepada siapa saja.

Dalam penulisan tugas akhir ini selanjutnya penulis akan membahas sedikit tentang salah ilmu kriptografi kunci simetri yang digunakan dalam mengenkripsi pesan yaitu menggunakan Algoritma CAST-128. Algoritma CAST-

128 adalah algoritma yang dipopulerkan oleh C.M Adams dan Saffords Tavares yang dipublikasikan tahun 1997.

2.4.1 Deskripsi Algoritma

Algoritma CAST-128 termasuk diklasifikasikan sebagai algoritma enkripsi atau penyandian data *feistel cipher*. dari semua operasinya serupa dengan algoritma DES (Data Encryption Standard). Didalam algoritma CAST-128 terdapat empat langkah yang harus dilakukan dalam melakukan enkripsi yaitu sebagai berikut;

INPUT : plaintext m1…m64; key K = k1… k128;

OUTPUT : ciphertext c1…c64;

1. (penjadwalan kunci) hitung 16 pasang dari sub kunci {Kmi, Kri} dari K
2. (LO,RO) ß (m1…m64). (potong plaintext dengan membagi dua yaitu kiri dan kanan masing masing 32 bit seperti LO = m1…m32 dan RO = m32…m64)
3. (16 putaran) for i dari 1 sampai 16, hitung Li dan Ri sebagai contoh:

   Li = Ri-1;

   Ri = Li-1 ^ f(Ri-1,Kmi,Kri), dimana f dinyatakan sebagai fungsi. (fungsi f dapat dibedakan menjadi 3 tipe dan berdasarkan pada nilai i.
4. c1…c64 ß (R16,L16). Perubahan *block* terakhir L16 dan R16 digabungkan kedalam *ciphertext.*

Pendeskripsian identik dengan penggunaan algoritma enkripsi yang diberikan, kecuali putaran tersebut dapat mengembalikan sepasang sub kunci yang

digunakan untuk mengembalikan urutan untuk menghitung (LO,RO) from (R16, L16).

2.4.2 Pasangan Kunci Putaran (*Pair of Round Keys*)

Algoritma enkripsi CAST-128 digunakan dari perputaran sepasang sub kunci: dengan menggunakan 32 bit jumlah Km (*masking key*) dan 5 bit jumlah dari Kr (*rotation key*).

2.4.3 Putaran Tidak Identik (*Non-Identical Rounds*)

Seperti yang dijelaskan diatas bahwa di dalam algoritma Cast-128 terdiri dari tiga fungsi yang berbeda dalam melakukan perputaran (*round*). Misalkan "D" adalah data yang di inputkan kedalam suatu fungsi dan "Ia" – "Id' adalah nilai bobot (*byte*) terbesar dan nilai bobot (*byte*) yang terkecil dari I, diputar secara berturut-turut. Dengan catatan adanya operasi penambahan "+", pengurangan "-", Modulo 2^{**32}, "^" atau XOR, dan operasi pergeseran bit ke sebelah kiri "<<<".

Fungsi Tipe 1 : I = ((Kmi + D) <<< Kri)

f = ((S1[Ia] ^ S2[Ib]) – S3[Ic]) + S4 [Id]

Fungsi Tipe 2 : I = ((Kmi ^ D) <<< Kri)

f = ((S1[Ia] – S2[Ib]) + S3[Ic]) ^ S4 [Id]

Fungsi Tipe 3 : I = ((Kmi – D) <<< Kri)

f = ((S1[Ia] + S2[Ib]) ^ S3[Ic]) – S4 [Id]

Putaran 1, 4, 7, 10, 13, dan 16 menggunakan f fungsi tipe 1

Putaran 2, 5, 8, 11, dan 14 menggunakan f fungsi tipe 2

Putaran 3, 6, 9, 12, dan 15 menggunakan f fungsi tipe 3

2.4.4 Kotak Operasi Subsitusi (*Substitution Boxes*)

Dalam algoritma CAST-128 terdapat delapan kotak / operasi subsitusi atau disebut *s-boxes*. *S-boxes* S1, S2, S3, S4 adalah fungsi rotasi *s-boxes*; S5, S6, S7, dan S8 adalah penjadwalan kunci *s-boxes*, meskipun 8 *s-boxes* mengharuskan total dari penyimpanan adalah 8 KByte, dengan catatan hanya 4 KByte adalah mengharuskan hanya pada masa enkripsi / deskripsi sejak generasi sub kunci, setelah terlebih dahulu dilakukan untuk segala input data.

2.4.5 Penjadwalan Kunci (*Key Schedule*)

Ukuran kunci yang digunakan dalam algoritma CAST-128 adalah 128 bit kunci seperti x0x1x2x3x4x5x6x7x8x9xAxBxCxDxExF, dimana x0 di representasikan sebagai nilai bobot (byte) terbesar dan xF direpresentasikan sebagai nilai bobot (byte) paling kecil. Proses yang dilakukan dalam penjadwalan adalah melakukan operasi penambahan "+", pengurangan "-", Modulo 2^{**32}, "^"atau XOR, dan operasi pergeseran bit ke sebelah kiri "<<<". Kemudian untuk z0..zF diberikan sebagai byte (*temporary*) dan Si[ ] representasi dari *S-boxes* i dan "^" representasi penambahan XOR. Sub kunci dibentuk dari kunci seperti x0x1x2x3x4x5x6x7x8x9xAxBxCxDxExF. Seperti Operasi yang terjadi dibawah ini:

z0z1z2z3 = x0x1x2x3 ^ S5[xD] ^ S6[xF] ^ S7[xC] ^ S8[xE] ^ S7[x8]

z4z5z6z7 = x8x9xAxB ^ S5[z0] ^ S6[z2] ^ S7[z1] ^ S8[z3] ^ S8[xA]

z8z9zAzB = xCxDxExF ^ S5[z7] ^ S6[z6] ^ S7[z5] ^ S8[z4] ^ S5[x9]

zCzDzEzF = x4x5x6x7 ^ S5[zA] ^ S6[z9] ^ S7[zB] ^ S8[z8] ^ S6[xB]

K1 = S5[z8] ^ S6[z9] ^ S7[z7] ^ S8[z6] ^ S5[z2]

K2 = S5[zA] ^ S6[zB] ^ S7[z5] ^ S8[z4] ^ S6[z6]

K3 = S5[zC] ^ S6[zD] ^ S7[z3] ^ S8[z2] ^ S7[z9]

K4 = S5[zE] ^ S6[zF] ^ S7[z1] ^ S8[z0] ^ S8[zC]

x0x1x2x3 = z8z9zAzB ^ S5[z5] ^ S6[z7] ^ S7[z4] ^ S8[z6] ^ S7[z0]

x4x5x6x7 = z0z1z2z3 ^ S5[x0] ^ S6[x2] ^ S7[x1] ^ S8[x3] ^ S8[z2]

x8x9xAxB = z4z5z6z7 ^ S5[x7] ^ S6[x6] ^ S7[x5] ^ S8[x4] ^ S5[z1]

xCxDxExF = zCzDzEzF ^ S5[xA] ^ S6[x9] ^ S7[xB] ^ S8[x8] ^ S6[z3]

K5 = S5[x3] ^ S6[x2] ^ S7[xC] ^ S8[xD] ^ S5[x8]

K6 = S5[x1] ^ S6[x0] ^ S7[xE] ^ S8[xF] ^ S6[xD]

K7 = S5[x7] ^ S6[x6] ^ S7[x8] ^ S8[x9] ^ S7[x3]

K8 = S5[x5] ^ S6[x4] ^ S7[xA] ^ S8[xB] ^ S8[x7]

z0z1z2z3 = x0x1x2x3 ^ S5[xD] ^ S6[xF] ^ S7[xC] ^ S8[xE] ^ S7[x8]

z4z5z6z7 = x8x9xAxB ^ S5[z0] ^ S6[z2] ^ S7[z1] ^ S8[z3] ^ S8[xA]

z8z9zAzB = xCxDxExF ^ S5[z7] ^ S6[z6] ^ S7[z5] ^ S8[z4] ^ S5[x9]

zCzDzEzF = x4x5x6x7 ^ S5[zA] ^ S6[z9] ^ S7[zB] ^ S8[z8] ^ S6[xB]

K9 = S5[z3] ^ S6[z2] ^ S7[zC] ^ S8[zD] ^ S5[z9]

K10 = S5[z1] ^ S6[z0] ^ S7[zE] ^ S8[zF] ^ S6[zC]

K11 = S5[z7] ^ S6[z6] ^ S7[z8] ^ S8[z9] ^ S7[z2]

K12 = S5[z5] ^ S6[z4] ^ S7[zA] ^ S8[zB] ^ S8[z6]

x0x1x2x3 = z8z9zAzB ^ S5[z5] ^ S6[z7] ^ S7[z4] ^ S8[z6] ^ S7[z0]

x4x5x6x7 = z0z1z2z3 ^ S5[x0] ^ S6[x2] ^ S7[x1] ^ S8[x3] ^ S8[z2]

x8x9xAxB = z4z5z6z7 ^ S5[x7] ^ S6[x6] ^ S7[x5] ^ S8[x4] ^ S5[z1]

xCxDxExF = zCzDzEzF ^ S5[xA] ^ S6[x9] ^ S7[xB] ^ S8[x8] ^ S6[z3]

K13 = S5[x8] ^ S6[x9] ^ S7[x7] ^ S8[x6] ^ S5[x3]

K14 = S5[xA] ^ S6[xB] ^ S7[x5] ^ S8[x4] ^ S6[x7]

K15 = S5[xC] ^ S6[xD] ^ S7[x3] ^ S8[x2] ^ S7[x8]

K16 = S5[xE] ^ S6[xF] ^ S7[x1] ^ S8[x0] ^ S8[xD]

2.4.6 Penyembunyian Kunci dan Rotasi Kunci

Km1, . . ., Km16 menjadi 32 bit kunci tersembunyi (satu/putaran).Kr1, . . ., Kr16 menjadi 32 bit rotasi kunci (satu/putaran). Paling sedikit 5 bit yang digunakan dalam setiap putaran. For (i=1; i<=16; i++) { Kmi = ki; Kri = K16+I;}

2.4.7 Ukuran Kunci Penentu (*Variable Keysize*)

Pada algoritma enkripsi CAST-128 telah didesain bahwa penggunaan ukuran kunci telah disetujui dapat diubah-ubah dari 40 bit sampai 128 bit ukuran kunci, kenaikan nya setiap 8 bit kunci. Contohnya ukuran kunci 40 menjadi 48, 56, 64,. . .,128 bit.. Untuk ukuran operasi variabel, di spesifikasikan sebagai berikut:

1. Untuk kenaikan ukuran kunci sampai 80 bit yaitu dimulai dari 40, 48, 56, 64, 72, 80 bit. Dalam putarannya algoritma ini minimal melakukan putaran 12 sampai 16 kali. Jika menggunakan kunci di bawah 80 maka algoritma ini menggunakan putaran 12 kali.

2. Untuk ukuran kunci lebih dari 80 bit, algoritma ini menggunakan satu putaran penuh.
3. Untuk ukuran kunci yang lebih sedikit dari 128 bit (16 *byte*), kunci ditambahkan dengan 0 *byte* (di sebelah kanan akhir) sedangkan lebih dari 128 bit (16 *byte*), algoritma ini dibatasi dengan 128 bit kunci sehingga telah disetujui bahwa inputan kunci hanya sampai 128 bit.

2.5 Rekayasa Perangkat Lunak

Perangkat lunak kini sudah menjadi kekuatan yang menentukan. Perangkat lunak menjadi mesin pengendali pengambilan keputusan didalam dunia bisnis; berfungsi sebagai dasar dari semua bentuk pelayanan serta penelitian keilmuan modern. Perangkat lunak dilekatkan dengan segala bentuk sistem; transportasi, medis, komunikasi, militer, proses industri, hiburan, produk-produk kantor, dan seterusnya. Perangkat lunak dapat diaplikasikan ke berbagai situasi dimana serangkaian langkah procedural (seperti algoritma) telah didefinisikan (pengecualian-pengecualian yang dapat dicatat pada aturan ini adalah sistem pakar dan perangkat lunak jaringan syaraf kecerdasan buatan). Kandungan (*content*) informasi dan determinasi merupakan faktor penting dalam menentukan sifat aplikasi perangkat lunak, analisis dan perancangan sistem dari perangkat lunak tersebut.

2.5.1 Model Proses Pembuatan Perangkat Lunak

Berkaitan dengan model proses perangkat lunak yang digunakan dalam pembuatan aplikasi pengamanan pesan pada *mail client.* Penulis menggunakan model analisa proses *Prototipe* dimana model prototipe ini dipakai pelanggan untuk mendefinisikan sasaran umum bagi perangkat lunak, tetapi tidak melakukan mengidentifikasi kebutuhan output, pemprosesan, ataupun input detail. Pada kasus lain, pengembangan mungkin tidak memiliki kepastian terhadap efisiensi algoritme, kemampuan penyesuaian dari suatu sistem operasi, atau bentuk bentuk yang harus dilakukan oleh interaksi manusia dengan mesin. Biasanya pada model prototipe berfungsi sebagai ”sistem pertama” dimana pengembang dan pelanggan bertemu dan mendefinisikan obyektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan ”perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek–aspek perangkat lunak tersebut nampak bagi pelanggan/pemakai (contohnya pendekatan input dan format output). Serta model analisa *Rapid Application Development* (RAD) pendekatan secara *Linear Sequential Model* atau disebut model *waterfall,* dalam pembuatan perangkat lunaknya.

Model ini sering disebut model klasik. Model ini menyarankan pendekatan pengembangan secara sekuen dan sistematik untuk pengembangan perangkat lunak dimulai di level sistem, berlanjut ke analisis, lalu perancangan, pemrograman, pengujian dan pemeliharaan. Model ini merupakan model yang tertua.

Kelemahan model ini adalah:

1. Proyek-proyek nyata jarang mengikuti alur sekuen yang diusulkan model. Meskipun linear model dapat mengakomodasikan iterasi, namun model melakukan secara tidak langsung. Sebagai hasilnya, perubahan-perubahan dapat menyebabkan kebingungan saat tim pengembangan melakukannya.
2. Sering *customer* sulit menyatakan semua kebutuhannya secara eksplisit. Model ini memerlukan pernyataan eksplisit itu dan sulit mengakomodasi ketidakpastian yang terdapat di awal dari kebanyakan proyek.
3. *Customer* harus memiliki kesabarann. *Versi* yang dapat bekerja dari program tidak akan tersedia sampai akhir dari proyek. Kesalahan besar utama, jika tidak terdeteksi sampai pada program kerja dikaji ulang, maka kesalahan itu dapat mengakibatkan program sama sekali tidak dapat digunakan. Menyiapkan banyak sumber daya.

Alasan kelemahan model ini adalah:

1. Kebutuhan harus telah ditetapkan di awal di siklus hidup.
2. Kebutuhan divalidasi terlalu lambat.

Meskipun terdapat kelemahan, namun model ini memberikan daftar lengkap jumlah aktivitas minimal yang terdapat di model pengembangan perangkat lunak. Umumnya, model-model pengembangan yang lain merupakan superset dari model pengembangan model *waterfall*, memuat aktivitas-aktivitas yang lebih banyak daripada model *waterfall* ini. Tahapan yang digunakan untuk membangun perangkat lunak dalam model ini yaitu sebagai berikut:

1. *System Requirement Specification (SRS)*

   Merupakan bagian dari sistem yang terbesar dalam pengerjaan suatu proyek, dimulai dengan menetapkan kebutuhan – kebutuhan dari semua elemen yang diperlukan sistem dan mengalokasikannya kedalam pembentukan perangkat lunak.

2. *Analysis*

   Merupakan tahap dimana sistem *engineering* menganalisis hal – hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak.

3. *Design*

   Tahap penerjemahan dari keperluan atau data yang dianalisis kedalam bentuk yang mudah dimengerti oleh *user* atau pemakai.

4. *Coding*

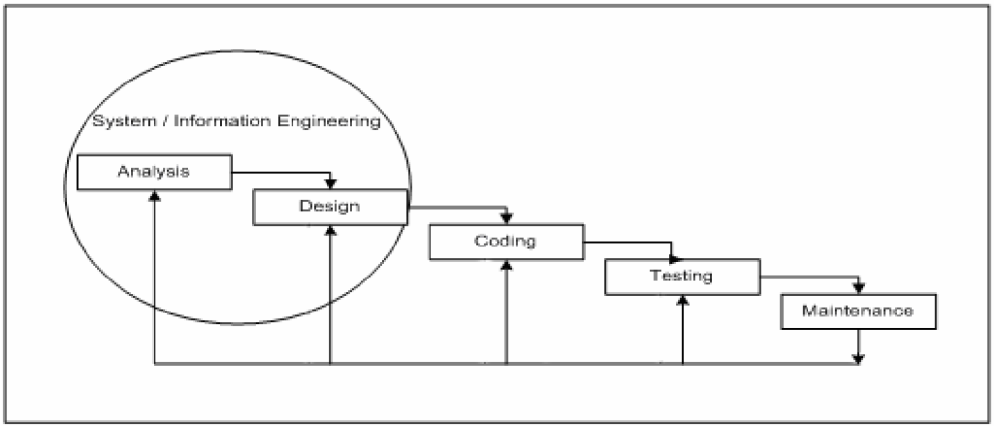
   Tahap penerjemahan data atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman komputer yang ditentukan.

5. *Testing*

   Tahap pengujian dari perangkat lunak yang dibangun.

6. *Maintenance*

   Tahap terakhir dimana suatu perangkat lunak yang sudah selesai dapat mengalami perubahan - perubahan atau penambahan sesuai dengan permintaan *user*.



Gambar 2.16 Diagram *Waterfall*

2.5.2 Metode Proses Pembuatan Perangkat Lunak

Pada tingkat teknik, rekayasa perangkat lunak dimulai dengan serangkaian tugas dari pemodelan yang membawa kepada spesifikasi lengkap dari persyaratan representasi desain dan representasi desain yang komprehensif bagi perangkat lunak yang akan dibangun. Model analisis, yang sebenarnya merupakan serangkaian model, merupakan representasi teknis yang pertama dari sistem. Berkaitan dengan metode pemodelan analisis terstruktur dimana metode pemodelan analisis terstruktur adalah serangkain aktivitas pembangunan model. Dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional, kita dapat menciptakan model yang menggambarkan muatan dan aliran informasi (data dan kontrol). Model analisis harus dapat mencapai tiga sasaran utama yaitu:

1. Untuk menggambarkan apa yang dibutuhkan oleh pelanggan.
2. Untuk membangun dasar bagi pembuatan desain perangkat lunak.
3. Untuk membatasi serangkaian persyaratan yang dapat divalid begitu perangkat lunak dibangun.

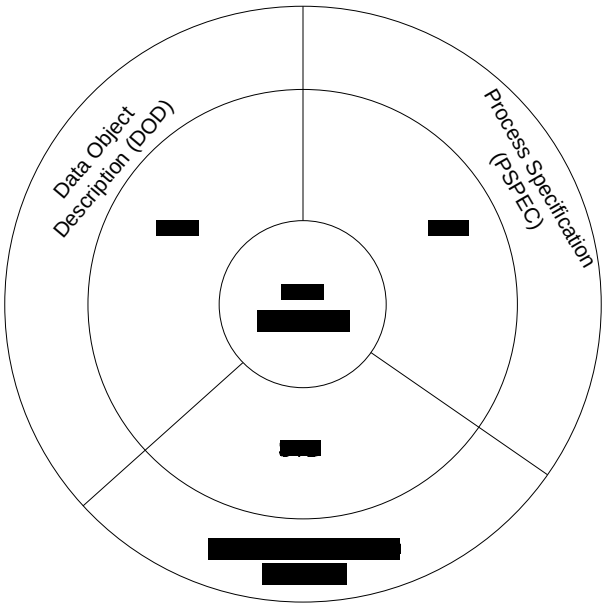
Untuk mencapai sasaran tersebut, model analisis yang ditarik selama analisis terstruktur berlangsung di tujukan dengan Gambar 2.17.

Pada inti model data ada kamus data (*Data Dictionary*) – penyimpanan yang berisi deskripsi dari semua objek data yang di konsumsi atau diproduksi oleh perangkat lunak. Disitu ada tiga diagram yang mengelilingi inti. *Entitiy-relationship diagram* (ERD) menggambarkan hubungan antara objek data. ERD adalah notasi yang digunakan untuk melakukan aktivitas pemodelan data. Atribut dari masing-masingobjek data yang ditulis pada ERD dapat digambarkan dengan menggunakan deskripsi objek data.

Data flow diagram (DFD) melayani dua tujuan yaitu untuk memberikan indikasi mengenai bagaimana data ditransformasikan pada saat data bergerak melalaui sistem dan untuk menggambarkan fungsi-fungsi (dan sub-fungsi) yang mentransformasikan aliran data. DFD memberikan informasi tambahan yang digunakan selama analisis domain informasi dan berfungsi sebagai dasar pemodelan fungsi. Deskripsi setiap fungsi yang disajikan pada DFD diisikan dalam bentuk spesifikasi proses atau *process specification* (PSPEC).

State-transition diagram (STD) menunjukan bagaimana sistem bertingkah laku sebagai akibat dari kejadian eksternal. Untuk melakukannya, STD menunjukan berbagai model tingkah laku (disebut *state*) sistem dan cara di mana transisi dibuat dari *state* satu ke *state* lainnya. STD berfungsi sebagai dasar bagi pemodelan tingkah laku. Informasi tambahan mengenai aspek kontrol dari perangkat lunak diisikan dalam spesifikasi kontrol atau *Control Specification*

(CSPEC). Model analisis meliputi masing-masing diagram, spesifikasi, dan deskripsi, serta kamus yang ditulis pada Gambar 2.17



Gambar 2.17 Analisis Terstruktur

2.6 Perangkat Lunak Pendukung

2.6.1 Borland Delphi

Borland Delphi adalah paket paket bahasa pemograman yang bekerja dalam sistem operasi windows. Delphi merupakan bahasa pemograman yang mempunyai cakupan kemampuan yang luas dan canggih. Berbagai jenis aplikasi dapat anda buat dengan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web. Secara umum, kemampuan Delphi adalah menyediakan komponen-komponen dan bahasa pemograman yang andal, sehingga memungkinkan Anda untuk membuat program aplikasi sesuai dengan keinginan, dengan tampilan dan kemampuan yang canggih.

Untuk mempermudah pemograman dalam membuat program aplikasi, dengan menyediakan fasilitas pemograman yang sangat lengkap. Fasilitas pemograman tersebut dibagi dalam dua kelompok, yaitu *object* dan bahasa pemograman. Secara ringkas, *objec*t adalah komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat (visual). Object biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu. Delphi menggunakan struktur bahasa pemograman *Object Pascal* yang sudah sangat dikenal dikalangan pemograman profesioanl. Seperti komponen visual (VCL) yang dikelompokan kedalam beberapa palet sesuai yang disesuaikan dengan klasifikasi dan fungsinya. Contohnya dengan adanya properti di dalam suatu objek form, yang dapat diubah-ubah seperti perubahan nilai border dan visible dari suatu form yang dapat diatur-atur. Dengan adanya pemograman visual maka para *programmer* dapat leluasa membuat komponen dan program sendiri.

2.6.2 MDaemon

MDaemon adalah program yang dipakai sebagai E-mail server, dimana MDemon merupakan aplikasi yang mengatur lalu lintas dari pesan E-Mail yang dikirim dan Pesan E-Mail yang diterima, serta *server mail* MDaemon dapat dikatakan sebagai MTA yang menjembatani antara *client* dan server, adapun tugas yang dapaat dilakukannya yaitu Menentukan header yang digunakan untuk

untuk mengenkapsulasi teks E-Mail, termasuk pengiriman dan penerimaan E-mail *Address*, Menentukan protokol untuk mentransmisikan dan menerima E-mail dan Menentukan protokol bagi *client* untuk memperoleh E-mail dari *server*.

2.6.3 Ethereal

Ethereal adalah perangkat lunak yang digunakan untuk melihat lalu-lintas data yang bertebaran di jaringan, mencari celah keamanan sampai pengumpulan data yang diakses *user*. Ethereal biasanya menjadi *tool* andalan yang dipakai seseorang untuk kegiatan *Wardriving* (*Wireless LAN Driving*) atau kegiatan mecari akses jaringan nirkabel, karena dengan menggunakan perangkat lunak tersebut di area WLAN, pengguna ethereal mendapatkan nama dari suatu *accces point*, jenis autentifikasi *access point* (*Open* atau *Encrypted*), jenis enkripsi yang digunakan (WEP, WPA, WPA2, WPA2 dengan PSK, dan data yang lewat pada jaringan nirkabel (termasuk *user* dan Password). Sama halnya dengan jaringan WLAN. *Tool* ini dapat dilakukan di lingkungan LAN untuk menguji keamanan lalu lintas data.