BAB II

LANDASAN TEORI

2.1. Jaringan Komputer.

2.1.1. Definisi.

Jaringan komputer merupakan sebuah kumpulan komputer, *printer* dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada *printer* yang sama dan bersama-sama menggunakan jaringan *hardware* atau *software* yang terhubung dengan jaringan. Secara umum jaringan komputer dibagi atas 5 (lima) jenis, yaitu :

2.1.1.1. *Local Area Network* (LAN).

Local area network merupakan jaringan milik pribadi didalam sebuah gedung atau beberapa gedung yang jaraknya hingga beberapa kilometer. LAN sering kali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik-pabrik untuk memakai sumber daya bersama dan saling bertukar informasi.

2.1.1.2. *Metropolitan Area Network* (MAN).

Pada dasarnya merupakan versi LAN yang berukuran lebih besar, MAN dapat mencakup kantor-kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi atau umum, MAN

mampu menunjang data dan suara bahkan dapat berhubungan dengan jaringan televisi kabel.

2.1.1.3 *Wide Area Network* (WAN).

Jangkauannya mencakup daerah geografis yang luas seringkali mencakup sebuah negara bahkan benua, WAN terdiri dari kumpulan mesin-mesin yang bertujuan untuk menjalankan program-program aplikasi.

2.1.1.4. *Internet*.

Banyaknya jaringan yang terhubung yang menggunakan perangkat keras dan perangkat lunak yang berbeda-beda, untuk menghubungkan satu jaringan dengan jaringan yang lainnya diperlukan sebuah mesin yang disebut *gateway* untuk melakukan hubungan dan melaksanakan terjemahan yang diperlukan baik perangkat keras maupun perangkat lunaknya. Dan kumpulan jaringan yang terkoneksi inilah yang disebut dengan *internet*.

2.1.1.5. Jaringan tanpa kabel (*Wireless*).

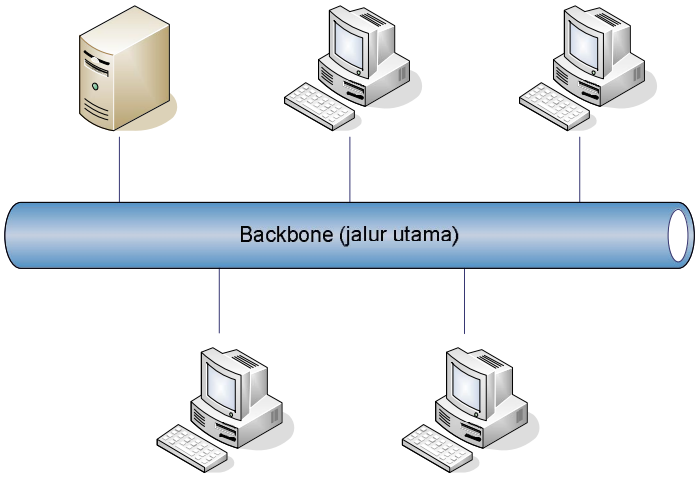
Jaringan tanpa kabel merupakan suatu solusi terhadap komunikasi yang tidak bisa dilakukan dengan jaringan yang menggunakan kabel seperti berkomunikasi dalam kendaraan ataupun perjalanan lainnya. Saat ini jaringan tanpa kabel sudah marak digunakan dengan memanfaatkan jasa satelit dan mampu memberikan kecepatan akses yang lebih cepat dibandingkan dengan jaringan yang menggunakan kabel.

2.1.2. Topologi Jaringan Komputer.

Topologi adalah suatu cara untuk menghubungkan komputer yang satu dengan komputer yang lainnya sehingga membentuk suatu jaringan. Cara yang saat ini banyak digunakan adalah topologi *bus*, *token ring*, *star*, dan *peer-to-peer*.

2.1.2.1. Topologi *Bus*.

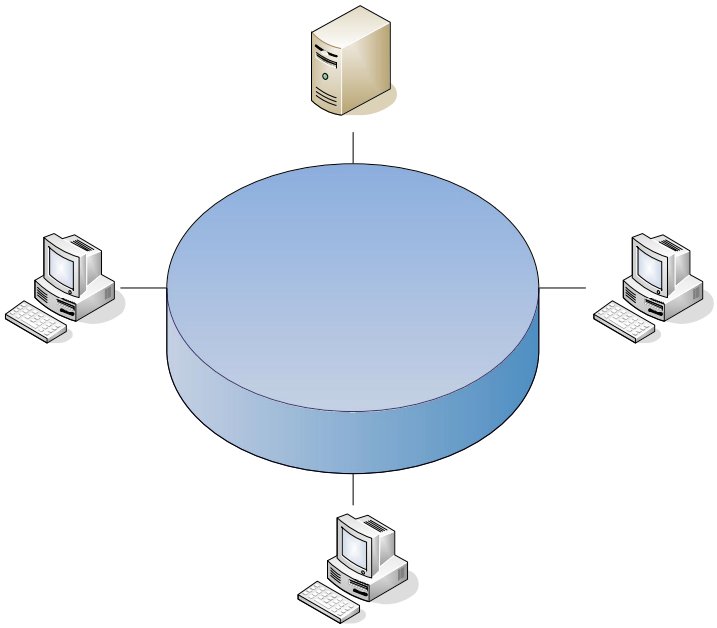
Topologi *bus* merupakan topologi yang paling sederhana dan paling mudah dalam instalasi secara fisik, ciri topologi *bus* adalah hanya memiliki satu jalur utama yang disebut dengan *backbone*. Setiap *host* (peralatan yang terkoneksi ke jaringan) dapat dikoneksikan secara langsung ke *backbone*.



Gambar 2.1 Topologi Jaringan *Bus*

2.1.2.2. 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 setiap informasi yang diterima simpul diperiksa alamatnya apakah data itu untuknya atau bukan.

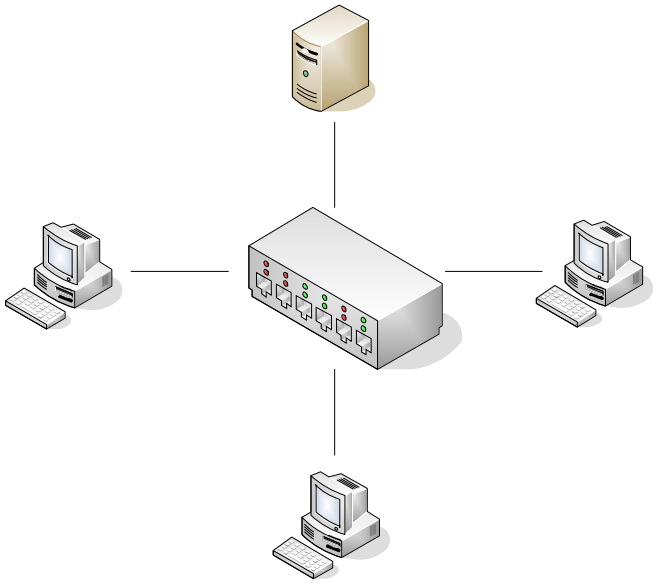


Gambar 2.2 Topologi *Token Ring*.

2.1.2.3. Topologi *Star*.

Merupakan metode terpusat, dimana semua *link* harus melewati pusat yang menyalurkan data tersebut kesemua simpul atau *client* yang dipilihnya, simpul pusat dinamakan stasiun pusat atau primer ataupun *server* dan lainnya dinamakan stasiun sekunder atau *client*, setelah hubungan jaringan dimulai oleh *server* maka

setiap *client* sewaktu-waktu dapat menggunakan hubungan jaringan tersebut tanpa menuggu perintah dari *server*.



Gambar 2.3 Topologi Star

2.1.2.4. Topologi *Peer –to-Peer*.

Merupakan jaringan komputer yang terdiri dari beberapa komputer, dalam sistem jaringan ini yang diutamakan adalah penggunaan data, program dan *printer* secara bersama-sama, ciri utamanya adalah adanya pemasangan *network card* di kedua komputer yang kemudian, dihubungkan dengan kabel yang khusus digunakan untuk sistem jaringan.



Gambar 2.4 Topologi *Peer-to-Peer*.

2.1.3. Protokol TCP/IP.

Sekitar tahun 1970-an *Departement of Deffence* (DOD) di Amerika Serikat mempelopori perkembangan protokol jaringan komputer yang sama sekali tidak terikat pada jenis komputer maupun media komunikasi yang digunakan, protokol yang dikembangkan diberi nama *internet protocol* (pada *transport Layer 1*), dan *Transmission Control Protocol* (pada *transport Layer 2*) atau disingkat TCP/IP. Jaringan komputer yang menggunakan TCP/IP kini lebih dikenal dengan jaringan *internet.*

TCP (*Transmission Control Protocol*) berperan dalam memperbaiki pengiriman data yang benar dari suatu *client* ke *server*. Data dapat hilang di tengah-tengah jaringan, TCP dapat mendeteksi *error* atau data yang hilang dan kemudian melakukan transmisi ulang sampai data diterima dengan benar dan lengkap.

IP (*Internet Protocol*) berperan dalam pentransmisian paket data dari node ke node, IP mendahului setiap paket data berdasarkan 4 *byte* (IP versi 4) alamat tujuan (nomor IP). *Internet authorities* menciptakan range angka untuk organisasi yang berbeda, organisasi menciptakan grup dengan nomornya untuk departemen. IP bekerja pada mesin *gateway* yang memindahkan data dari departemen ke organisasi kemudian ke region dan kemudian keseluruh dunia.

Socket yang merupakan nama diberikan pada subrutin paket yang menyediakan akses TCP/IP.

2.2. Keamanan Jaringan.

2.2.1. Aspek Keamanan Jaringan.

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

2.2.1.1. *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 diperboleh untuk keperluan 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.

Sedangkan 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.

2.2.1.2. *Integrity*.

Integrity lebih menekankan bahwa informasi tidak boleh diubah tanpa seijin pemilik informasi, adanya virus, worm atau pemakai lain yang mengubah informasi tanpa ijin merupakan contoh masalah yang sering dihadapi. Sebagai contoh misalnya sebuah informasi data dapat saja di tangkap di tengah jalan yang kemudian dimodifikasi dan diteruskan ke alamat yang dituju, dengan kata lain integritas dari informasi tersebut sudah tidak terjaga lagi, penggunaan kriptografi dan tanda tangan digital misalnya dapat mengurangi atau bahkan mengatasi masalah tersebut.

2.2.1.3. *Authentification*.

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 dimaksud. Masalah pertama yang muncul membuktikan keaslian dokumen, dapat dilakukan dengan teknologi *watermarking* dan tanda tangan dgital, 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 cirri-ciri khusus yang

berhubungan dengan pemilik atau pengguna yang sah seperti ATM, PIN, sidik jari, *biometric* dan lain-lain.

2.2.1.4. *Availability*.

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

2.2.1.5. *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* sering kali dilakukan dengan menggunakan kombinasi *user-id* atau *password* atau dengan menggunakan mekanisme lain (kartu atau *biometric*).

2.2.1.6. *Non-Repudiation*.

Aspek ini menjaga agar seseorang tidak dapat menyangkal telah melakukan transaksi, sebagai contoh seseorang yang mengirimkan *e-mail* untuk memesan barang tidak dapat menyangkal bahwa dia telah mengirimkan *e-mail* 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.

2.2.2. Klasifikasi Keamanan Jaringan.

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

2.2.2.1. 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 ke tempat sampah untuk mencari berkas-berkas yang mungkin memiliki informasi tentang keamanan atau hal-hal yang berhubungan dengan akses kekabel atau komputer yang digunakan.

2.2.2.2. Keamanan yang berhubungan dengan perorangan (*personal*).

Keamanan yang berhubungan dengan perorangan termasuk identifikasi dan profil dari seorang yang mempunyai akses. Seringakali kelemahan sistem

informasi begantung kepada manusia (pemakai dan pengelola). Teknik *social enggineering* merupakan teknik yang digunakan kriminal untuk berpura-pura sebagai orang yang berhak untuk mengakses informasi.

2.2.2.3. Keamanan dari data dan media serta teknik komunikasi (*Communication*).

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

2.2.2.4. Keamanan dalam operasi.

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

2.2.3 *Firewall*.

Definisi dasar dari *firewall* sendiri adalah komponen atau satu kelompok komponen yang mengatur akses antara *internal network* (*network* yang akan dilindungi) dengan *eksternal network* (*intenet*) atau antara beberapa kumpulan *network*. *Firewall* dapat berupa solusi *hardware* dan *software* yang membatasi akses dari jaringan *internal* ke *internet* atau sebaliknya. *Firewall* dapat juga

digunakan untuk memisahkan dua atau lebih bagian jaringan lokal. *Firewall* merupakan suatu cara yang efektif untuk melindungi jaringan dari ancaman gangguan lewat *internet*. Selain itu *firewall* dalam jaringan komputer membatasi dan menjaga kerusakan pada satu bagian jaringan tidak menyebar ke bagian lain di jaringan.

2.2.4 Autentifikasi *Password*.

Autentifikasi lewat *password* dilakukan untuk membatasi dan mengatur hak akses dari pengguna sistem sehingga tidak sembarangan orang dapat masuk ke dalam sistem tersebut, dengan menyamakan *user-id* dan *password* yang dimiliki program atau 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 ke dalam sistem.

2.2.5 Jenis-Jenis Ancaman.

2.2.5.1. DOS/DDOS

Denial of Services dan *Distributed Denial of Services* adalah sebuah metode serangan yang bertujuan untuk menghabiskan sumber-daya sebuah peralatan jaringan komputer, sehingga layanan jaringan komputer menjadi terganggu.

2.2.5.2. Packet Sniffing

Packet Sniffing adalah sebuah metode serangan dengan cara mendengarkan seluruh paket yang lewat pada sebuah media komunikasi, baik itu media kabel maupun radio. Setelah paket-paket yang lewat itu didapatkan, paket-paket tersebut kemudian disusun ulang sehingga data yang dikirimkan oleh sebuah pihak dapat dicuri oleh pihak yang tidak berwenang.

2.2.5.3. IP Spoofing

IP Spoofing dilakukan dengan cara merubah alamat asal sebuah paket, sehingga dapat melewati perlindungan firewall.

2.2.5.4. Forgery

Salah satu cara yang dapat dilakukan oleh seseorang untuk mencuri data-data penting orang lain adalah dengan cara melakukan penipuan. Salah satu bentuk penipuan yang bisa dilakukan adalah dengan cara membuat sebuah website tiruan (misalkan meniru klikbca.com), lalu memancing pihak yang ingin ditipu untuk meng-akses website palsu tersebut. Setelah kita memiliki data-data yang diperlukan, kita dapat melakukan akses ke website yang asli sebagai pihak yang kita tipu.

2.3. Kritpografi.

2.3.1. Definisi

Kriptografi (*Crytography*) berasal dari bahasa Yunani yaitu dari kata *Crypto* dan *Graphia* yang berarti penulisan rahasia. Kriptografi adalah suatu ilmu

yang mempelajari penulisan secara rahasia. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut *Cryptology*. Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah atau berhak.

Dalam menjaga kerahasian data, kritografi mentransformasikan data jelas (*plaintext*) ke dalam bentuk sandi (*chiphertext*) yang tidak dapat dikenali. *Chiphertext* inilah yang kemudian dikirimkan oleh pengirim (*sender*) kepada penerima (*receiver*). Setelah sampai di penerima, *chiphertext* tersebut ditranformasikan kembali ke dalam bentuk *plaintext* agar dapat dikenali.

Proses tranformasi dari *plaintext* menjadi *chiphertext* disebut proses *Enciphement* atau enkripsi (*encryption*), sedangkan proses mentranformasikan kembali *chiphertext* menjadi *plaintext* disebut proses deskripsi (*decryption*).

Untuk mengenkripsi dan mendepkripsi data. Kriptografi menggunakan suatu algoritma (*Chipher*) dan kunci (*key*). *Chipher* adalah fungsi matematika yang digunakan untuk mengenkripsi dan mendekripsi data. Sedangkan kunci merupakan sederetan bit yang diperlukan untuk mengenkripsi dan mendekripsi data.

Algoritma kritografi modern tidak lagi mengandalkan keamanannya pada kerahasian algoritma tetapi kerahasian kunci. *Plaintext* yang sama bila disandikan dengan kunci yang berbeda akan menghasilkan *chiphertext* yang berbeda pula. Dengan demikian algoritma kriptografi dapat bersifat umum dan boleh diketahui oleh siapa saja, akan tetapi tanpa pengetahuan tentang kunci, data tersandi tetap saja tidak dapat terpecahkan.

2.3.2. Terminologi dasar Kriptografi

Kriptografi merupakan kumpulan teknik untuk mengenkode data dan pesan sedemikian sehingga data dan pesan tersebut dapat disimpan dan ditransmisikan dengan aman.

Berikut ini beberapa terminologi dasar dari kriptografi serta hal-hal yang berkaitan dengan terminologi tersebut,

1. Kriptografi dapat digunakan untuk meningkatkan keamanan komunikasi meskipun komunikasi tersebut dilakukan dengan media komunikasi yang sangat tidak aman (misalnya *Internet*). Kita juga dapat menggunakan kriptografi untuk melakukan enkripsi file-file sensitif kita, sehingga orang lain tidak dapat mengartikan data-data yang ada.
2. Kriptografi dapat digunakan untuk memberikan jaminan integritas data serta menjaga kerahasiaan.
3. Dengan menggunakan kriptografi, maka sangat mungkin untuk meverifikasi asal data dan pesan yang ada menggunakan digital signature.
4. Pada saat menggunakan metoda kriptografi, hanya kunci sesi yang harus tetap dijaga kerahasiannya. Algoritma, ukuran kunci dan format file dapat dibaca oleh siapapun tanpa mempengaruhi keamanan.

2.3.3. Klasifikasi Algoritma Kriptografi

Tingkat keamanan suatu algoritma diperoleh dengan menyembunyikan secara rahasia bagaimana algoritma itu bekerja, algoritma ini disebut dengan algoritma rahasia (*restricted algorithm*). Pada awalnya algoritma jenis ini yang berkembang, namun algoritma jenis ini mempunyai banyak kelemahan seperti

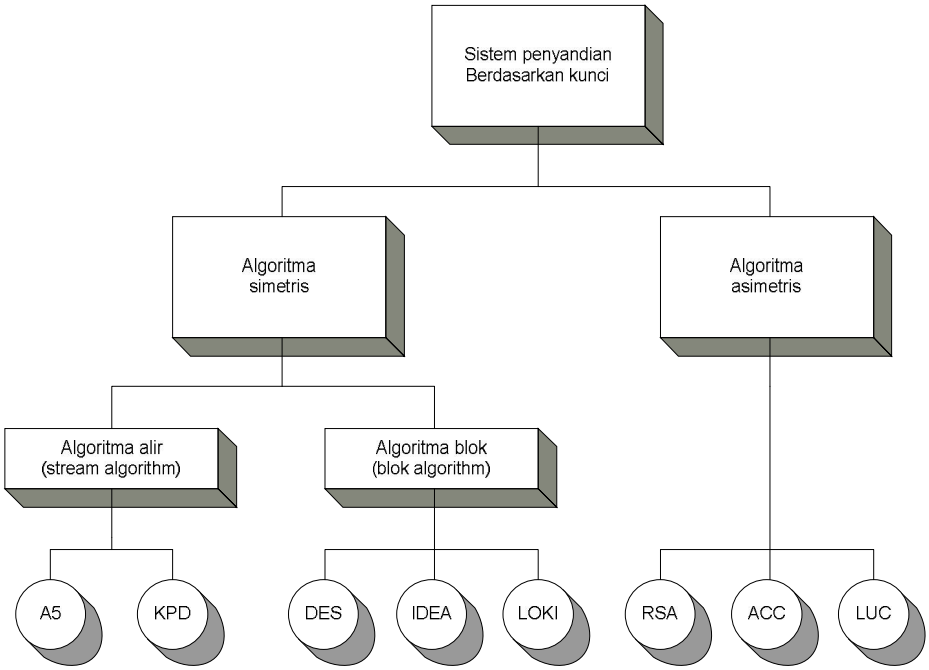
setiap pengguna harus menggunakan algoritmanya sendiri dan jika algoritma ini telah diketahui orang, algoritma ini harus diganti dengan yang baru.

Kelemahan lain dari algoritma jenis ini adalah tidak memungkinkan dilakukan standardisasi sebagia kendali mutu, dimana setiap kelompok pengguna harus mempunyai algotima sendiri-sendiri dan unik.

Kriptografi modern dapat memecahkan masalah algoritma tersebut diatas yaitu dengan algoritma kunci. Kunci ini dapat berupa sembarang dari suatu nilai dari sejumlah angka. Dengan demikian tingkat keamanan dari algoritma yang menggunakan kunci adalah berdasarkan kerahasiaan kuncinya, tidak berdasarkan detail dari algoritma itu sendiri. Oleh karena itu algoritma ini dapat dapat dipublikasikan dan dianalisa, dan algoritma ini dapat diproduksi secara masal.

Dewasa ini algoritma rahasia hanya digunakan terbatas pada aplikasi yang relatif kurang membutuhkan tingkat keamanan yang tinggi misalkan pada teknik pengacakan (*scrembing*) video, sedangkan algoritma algoritma kunci digunakan pada berbagai aplikasi yang membutuhkan tingkat keamanan yang sangat tinggi, misalkan pada kartu ATM dan *secure mail*.

Secara umum ada dua tipe algoritma yang berdasarkan kunci yaitu algoritma simetris (*symetric algorithm*) dan algoritma kunci publik (*public-key algorithm*). Perbedaan utama antara *Symetric algorithm* dengan *public-key algoritm* adalah pada kunci enkripsi. Selain itu juga terdapat perbedaan dalam kecepatan proses dan keamananya. K1 dan kunci dekripsi K2. Berikut ini bagan suatu sistem penyandian berdasarkan kunci beserta contoh-contoh algoritma yang digunakan.



Gambar 2.5 Blok Diagram Pembagian sistem Kriptografi berdasarkan Kunci

2.3.4. Konsep Penggunaan Kriptografi.

Adapun konsep penggunaan dalam ilmu kriptografi yaitu :

1. Kerahasiaan (*Confidentiality*).

   Sederhananya, kerahasiaan adalah proses penyembunyian data dari orang-orang yang tidak punya otoritas.

2. Integritas (*Integrity*).

   Proses untuk menjaga agar sebuah data tidak dirubah-rubah sewaktu ditransfer atau disimpan.

3. Penghindaran Penolakan (*Non-repuditation)*

   Proses untuk menjaga bukti-bukti bahwa suatu data berasal dari seseorang. Seseorang yang ingin menyangkal bahwa data tersebut bukan berasal

darinya, dapat saja melenyapkan bukti-bukti yang ada. Karenanya diperlukan teknik untuk melindungi data-data tersebut.

4. Autentikasi (*Authentication*)

   Proses untuk menjamin keaslian suatu data.

5. Tanda Tangan Data (*Data Signature*)

   Dapat disebut juga sebagai tanda tangan digital. Berguna untuk menandatangi data digital. Contohnya adalah *Digital Signature Algorithm* (DSA)

6. Kontrol Akses (*Access Control*)

   Untuk mengontrol akses terhadap suatu entity.

   Contoh penggunaan kriptografi di dunia *internet* antara lain*: Secure Shell* (SSH), SSL (*Secure Socket Layer*), *Secure Hypertext Transfer Protocol* (HTTP), dan lain lain.

2.3.5. Konsep Dasar Kriptografi.

Ada tiga bagian penting dalam kriptografi, yaitu :

1. *Plaintext*.

   Suatu data yang masih dalam bentuk asli (belum diacak), dapat berupa sebuah urutan *bit, file* teks, *bitmap* ataupun *digital video image.*

2. *Chiphertext*.

   Data dalam bentuk yang telah diacak, bisa berbentuk sandi atau kode tertentu yang berbeda dalam bentuk asli.

3. *Key* (kunci).

   Nilai yang dipakai pada algoritma kriptografi untuk mengacak *plaintext* sehingga menghasilkan *chiphertext*.

Dalam algoritma juga diperlukan suatu metode, yang dimaksud metode ini adalah suatu algoritma yang digunakan untuk pengacakan data. Suatu contoh sederhana yang dapat menjelaskan apa itu algoritma dan kunci adalah sebagai berikut.

Metode (algoritma) ***substract N letter***, sedangkan kuncinya adalah suatu nilai spesifik, contohnya N=1. Jika ada suatu pesan misalnya ”cinta” jika dienkripsikan akan menjadi ”djoub”. Penjelasannya adalah sebagai berikut. Tiap huruf diganti dengan huruf yang ada di depannya dalam susunan alphabet (mirip dengan yang dilakukan oleh Jullius Caesar). Maka huruf c menjadi d, i menjadi j, n menjadi o, t menjadi u dan a menjadi b.

Dalam beberapa kasus, suatu algoritma kriptografi diketahui oleh semua perusahaan atau organisasi yang bersangkutan. Sehingga algoritma tersebut tidak lagi berguna tanpa adanya suatu kunci. Jadi dalam suatu proses kriptografi tidak hanya algoritma yang diperlukan, tetapi juga suatu kunci.

Algoritma suatu proses kriptografi tidak dapat disimpan secara rahasia karena dapat diketahui oleh semua orang atau perusahaan yang menggunakan algoritma tersebut. Yang dapat dirahasiakan hanya kunci saja. Untuk menjaga keamanan data yang disimpan maka sudah menjadi keharusan untuk menjaga kerahasiaan kunci. Seperti pada contoh yang telah dijelaskan diatas, metode

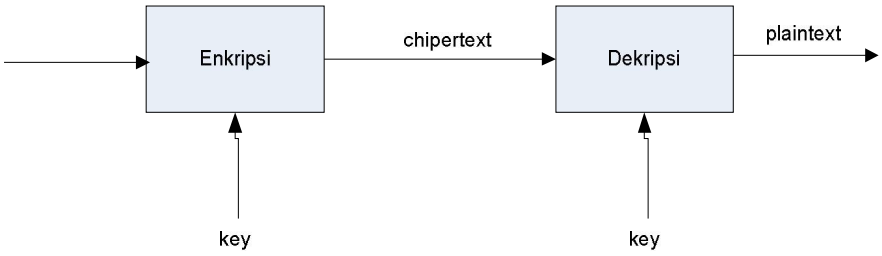
substract N letters dapat diketahui oleh siapa saja. Tapi yang menjadi kuncinya yaitu nilai N berapa yang digunakan tidak boleh diketahui oleh orang lain.

Kemudahan memecahkan suatu kunci tergantung pada panjang bit kunci tersebut. Semakin panjang (dalam satu bit) suatu kunci maka akan semakin sulit dipecahkan, jika panjang suatu kunci hanya 4 bit (sangat pendek) maka akan sangat mudah badi seseorang untuk memecahkannya, misalnya mencoba-coba semua kemungkinan yang ada.

Tetapi di lain pihak, dianjurkan agar sejumlah bit suatu kunci tidak terlalu panjang. Jumlah bit yang panjang sangat lambat, sukar dan kompleks untuk dijalankan. Biasanya jumlah bit yang dipergunakan adalah 64 bit, dengan alasan bahwa dengan jumlah bit data, suatu kunci tidak terlalu mudah dan tidak terlalu sukar pula untuk dimanipulasi.

2.3.6. Enkripsi dan Dekripsi.

Suatu pesan yang tidak disandikan disebut sebagai *plaintext* ataupun dapat disebut juga sebagai *cleartext*. Proses yang dilakukan untuk mengubah plaintext ke dalam ciphertext disebut *encryption* (enkripsi) atau *encipherment*. Sedangkan proses untuk mengubah ciphertext kembali ke plaintext disebut *decryption* (dekripsi) atau *decipherment*. Secara sederhana istilah-istilah di atas dapat dilihat pada gambar 2.6 berikut.



Gambar 2.6 Proses Enkripsi/Dekripsi Sederhana

Cryptography adalah suatu ilmu ataupun seni mengamankan pesan, dan dilakukan oleh *cryptographer*. Sedang, *cryptanalysis* adalah suatu ilmu dan seni membuka (*breaking*) *ciphertext* dan orang yang melakukannya disebut *cryptanalyst.*

Cryptographic system atau *cryptosystem* adalah suatu fasilitas untuk mengkonversikan *plaintext* ke *ciphertext* dan sebaliknya. Dalam sistem ini, seperangkat parameter yang menentukan transformasi pencipheran tertentu disebut suatu set kunci. Proses enkripsi dan dekripsi diatur oleh satu atau beberapa kunci kriptografi. Secara umum, kunci-kunci yang digunakan untuk proses pengenkripsian dan pendekripsian tidak perlu identik, tergantung pada sistem yang digunakan.

Secara umum operasi enkripsi dan dekripsi dapat diterangkan secara matematis sebagai berikut :

EK (M) = C (Proses Enkripsi)

DK (C) = M (Proses Dekripsi)

Pada saat proses enkripsi kita menyandikan pesan M dengan suatu kunci K lalu dihasilkan pesan C. Sedangkan pada proses dekripsi, pesan C tersebut

diuraikan dengan menggunakan kunci K sehingga dihasilkan pesan M yang sama seperti pesan sebelumnya.

Dengan demikian keamanan suatu pesan tergantung pada kunci ataupun kunci-kunci yang digunakan, dan tidak tergantung pada algoritma yang digunakan. Sehingga algoritma-algoritma yang digunakan tersebut dapat dipublikasikan dan dianalisis, serta produk-produk yang menggunakan algoritma tersebut dapat diproduksi massal. Tidaklah menjadi masalah apabila seseorang mengetahui algoritma yang kita gunakan. Selama ia tidak mengetahui kunci yang dipakai, ia tetap tidak dapat membaca data.

2.3.6.1. Enkripsi

Enkripsi merupakan proses awal kriptografi dengan memasukan input berupa *plaintext* kedalam algoritma tertentu beserta suatu kunci untuk memperoleh *output* berupa *chipertext* yang tidak dapat langsung dimengerti oleh orang lain, proses enkripsi dapat diformulakan sebagai berikut :

$C = E_k (P)$ ... P(2.1)

Dimana :

P = *Plaintext*

E_k = Proses enkripsi dengan menggunakan algoritama tertentu dan kunci k

C = *Chipertext*

Proses enkripsi biasanya melibatkan perubahan bentuk data menjadi suatu bentuk yang sulit untuk dibaca dan diartikan tanpa sebuah kunci. Enkripsi digunakan untuk memastikan bahwa data atau pesan yang disimpan atau

dikirimkan melalui suatu jaringan terlindungi dari orang-orang yang tidak berhak untuk membacanya.

2.3.6.2 Dekripsi.

Deskripsi merupakan kebalikan dari proses enkripsi, yaitu proses mengubah bentuk yang sudah diacak pada proses enkripsi kembali ke bentuk semula. Pada deskripsi, *chipertext* menjadi *input* dan akan diolah dengan suatu algoritma dan suatu kunci untuk memperoleh *output* berupa *plaintext*.

Proses deskripsi dapat diformulakan sebagai berikut :

$P = D_k (C)$ ...P(2.2)

Dimana :

P = *Plaintext*

D_k = Proses deskripsi dengan menggunakan algoritma tertentu dan kunci K

C = *Chipertext*

Enkripsi dan deskripsi biasanya membutuhkan penggunaan beberapa informasai rahasia yang berhubungan dengan sebuah kunci. Untuk beberapa mekanisme enkripsi kunci yang sama digunakan untuk enkripsi dan deskripsi berbeda.

2.3.7. Jenis Kriptografi.

Berdasarkan kunci yang dipakai, algoritma kriptografi dapat dibedakan atas dua golongan, yaitu :

2.3.7.1. *Symetric Chrytography.*

Dalam *symmetric chrytography* ini, kunci yang digunakan untuk proses enkripsi dan dekripsi pada prinsipnya identik $K_1 = K_2 = K$, tetapi satu buah kunci dapat pula diturunkan dari kunci yang lainnya. Kunci-kunci ini harus dirahasiakan. Oleh karena itulah sistem ini sering disebut sebagai *secret-key chrytography*. Jumlah kunci yang dibutuhkan umumnya adalah :

$$ {}_nC_2 = \frac{n \,.\, (n-1)}{2} \qquad \text{...P(2.3)} $$

Dengan,

n = menyatakan banyaknya pengguna (user)

C = menyatakan banyaknya kunci.

Contoh dari sistem ini adalah :

- **IDEA** (*International Data Encryption Algoritm*)
- *Data Encryption Standard* (DES)
- *Blowfish*

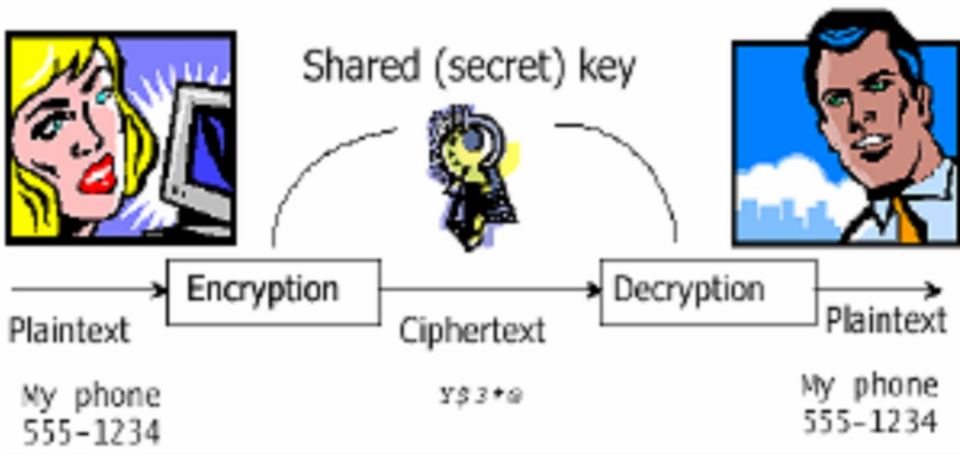
Tingkat keamanan kriptosistem yang menggunakan algoritma ini sangat diitentukan oleh kerahasiaan kunci K yang digunakan . Jika seseorang hendak mengirimkan suatu pesan atau transfer data kepada orang lain, atau melakukan *secure communication*, orang tersebut harus terlebih dahulu memberikan kepada pihak yang dituju kunci K yang hendak digunakanya. Hal ini jelas membutuhkan saluran komunikasi yang benar-benr aan dan tidak dapat disadap (*secure channel*). Secara matematis Algoritma ini dapat ditulis :

$E_k(M) = C \leftrightarrow d_k(C) = M$ ...P(2.4)

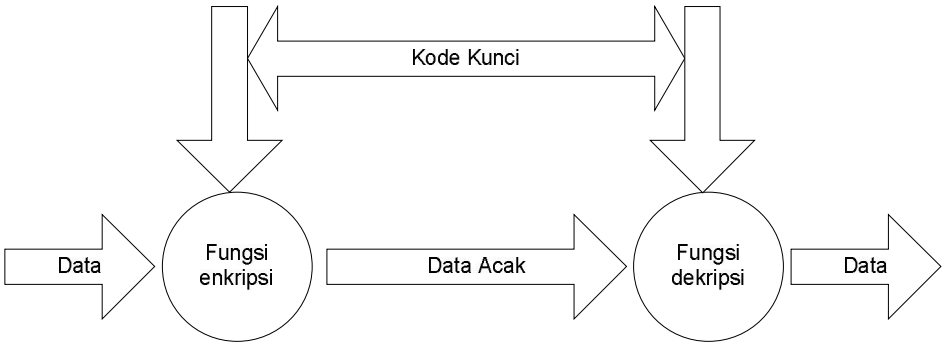
E_k adalah proses enkripsi dengan menggunakan kunci K

M adalah pesan asli (*Plainetext*)

C adalah pesan yang disandikan (*Chipertext*)



a).



b).

Gambar 2.7. a) dan b) Teknik Symmetric Chryptography.

Prinsp kerja dari kriptografi kunci simetrik adalah sebagai berikut :

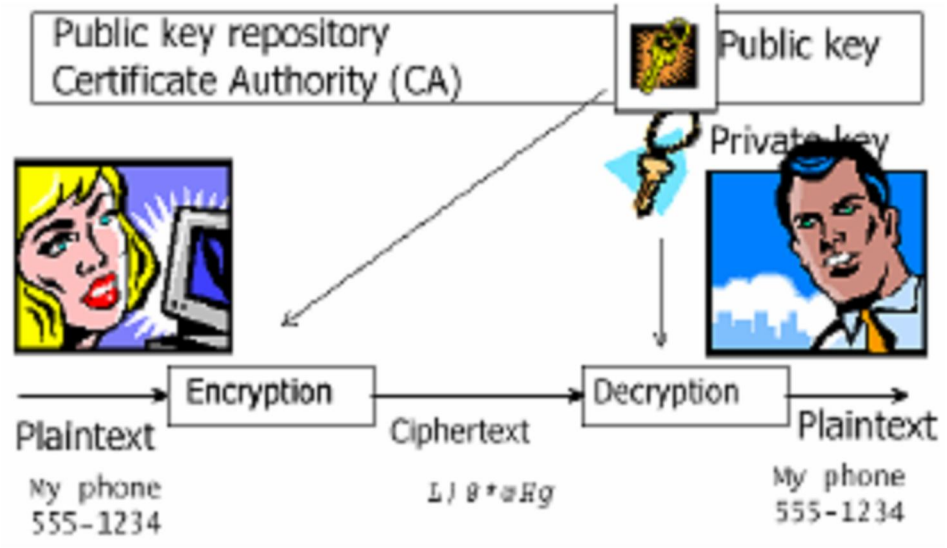
a. Pengirim dan penerima data atau Informasi sepakat menggunakan system kriptografi tertentu.
b. Pengirim dan penerima sepakat menggunakan satu kunci tertentu.
c. Dilakukan enkripsi sebelum data dikirim dan dekripsi setelah data dikirim.

Tingkat keamanan dari kriptograpfi yang menggunakan algoritma ini sangat ditentukan oleh kerahasiaan kunci K yang digunakan. Jika seserang hendak mengirimkan suatu pesan atau data kepada orang lain atau melakukan *secure communication*, orang tersebut harus terlebih dahulu memberitahu kepada pihak yang dituju kunci K yang digunakannya. Harus jelas membutuhkan saluran komunikasi yang benar-benar aman dan tidak ada penyadapan data (*secure channel*). Faktor inilah yang menjadi kelemahan cara ini yaitu masalah keamanan kunci dan bagaimana mendistribusikan kunci K tersebut.

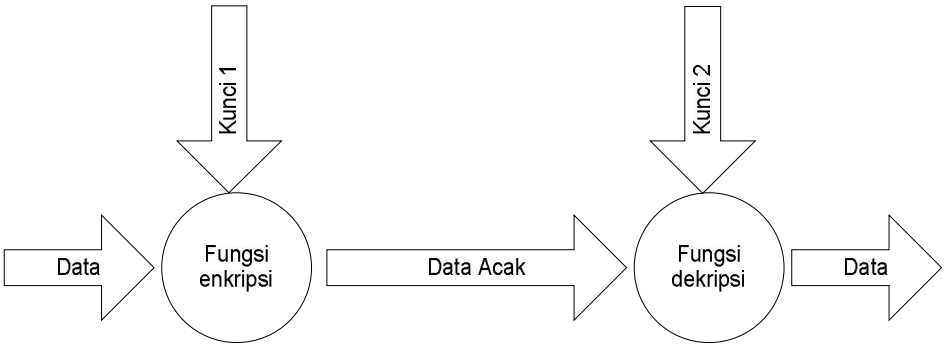
2.3.7.2. *Assymmetric Chryptography*

Dalam *assymmetric chryptography* ini digunakan dua buah kunci. Satu kunci yang disebut kunci publik (*public key*) dapat dipublikasikan, sedang kunci yang lain yang disebut kunci privat (*private key*) harus dirahasiakan. Proses menggunakan sistem ini dapat diterangkan secara sederhana sebagai berikut : bila A ingin mengirimkan pesan kepada B, A dapat menyandikan pesannya dengan menggunakan kunci publik B, dan bila B ingin membaca pesan tersebut, maka ia perlu mendekripsikan surat itu dengan kunci *private*-nya. Dengan demikian kedua belah pihak dapat menjamin asal surat serta keaslian surat tersebut, karena adanya

mekanisme ini. Contoh sistem ini antara lain RSA *Scheme* dan *Merkle-Hellman Scheme*.



a).



b).

Gambar 2.8. a) dan b) Teknik Assymmetric Cryptography

Setiap kriptografi yang baik harus memiliki karakteristik sebagai berikut :

a. Keamanan sistem terletak pada kerahasiaan kunci dan bukan pada kerahasiaan algoritma yang digunakan.
b. *Cryptography* yang baik memiliki ruang kunci (*keyspace*) yang besar.
c. *Cryptography* yang baik akan menghasilkan *chiphertext* yang terlihat acak dalam seluruh tes statistik yang dilakukan terhadapnya.
d. *Cryptography* yang baik mampu menahan seluruh serangan yang telah dikenal sebelumnya

Namun demikian perlu diperhatikan bahwa bila suatu kriptografi berhasil memenuhi seluruh karateristik di atas belum tentu ia merupakan sistem yang baik. Banyak kriptografi lemah yang terlihat baik pada awalnya. Kadang kala untuk menunjukkan bahwa suatu kriptografi kuat atau baik dapat dilakukan dengan menggunakan pembuktian matematika.

Hingga saat ini masih banyak orang yang menggunakan kriptografi yang relatif mudah dibuka, alasannya adalah mereka tidak mengetahui sistem lain yang lebih baik serta kadang kala terdapat motivasi yang kurang untuk menginvestasikan seluruh usaha yang diperlukan untuk membuka suatu sistem.

Berikut ini akan diperlihatkan perbedaan beerbagai aspek penting dari *Symmetric Cryptography* dengan *Assymmetric Cryptography.*

Tabel 2.1 Tabel perbedaan antara asimetris dan simetris kriptografi.

Symmetric Cryptography	*Symmetric Cryptography*
Yang dibutuhkan untuk bekerja : a. Algoritma yang sama dengan kunci yang sama dapat digunakan untuk proes enkripsi maupun dekripsi. b. Pengirim dan penerima harus membagialgoritma dan kunci yang sama	Yang dibutuhkan untuk bekerja : a. Algoritma yang digunakan untuk enkripsidan dekripsi dengan sepasang kunci, satu untuk enkripsi satu untuk dekripsi. b. Pengirim dan penerima harus mempunyai sepasan kunci yang cocok.
Yang dibutuhkan untuk keamanan : a. Kunci harus dirahasiakan. b. Adalah tidakmungkin atau sangat tidak praktis untuk menterjemahkan Informasi yang telah dienkripsi. c. Pengetahuan tentang algoritma dan sampel dari kata yang terenkripsi tidak menukpi untuk menetukan kunci	Yang dibutuhkan untuk keamanan : a. Salah satu kunci harus dirahasiakan . b. Adalah tidak mungkin atau sangat tidak praktis untuk menterjemahkan informasi yang telah di enkripsi. c. Pengetahuan tentang algoritma dan sampel dari kata yang terenkripsi tidak mencukupi untuk menentukan kunci

2.4. Algoritma IDEA.

2.4.1. Deskripsi Umum.

Algoritma penyandian IDEA (*International Data Encryption Algorithm*) muncul pertama kali pada tahun 1990 yang dikembangkan oleh ilmuwan Xueijia Lai dan James L Massey. Algoritma utama dari sistem kriptografi IDEA adalah sebagai berikut :

1. Proses enkripsi : $e_k(M) = C$ …P(2.5)

2. Proses dekripsi : $d_k(C) = M$

Dimana :

E = adalah fungsi enkripsi

D = adalh fungsi dekripsi

M = adalah pesan terbuka

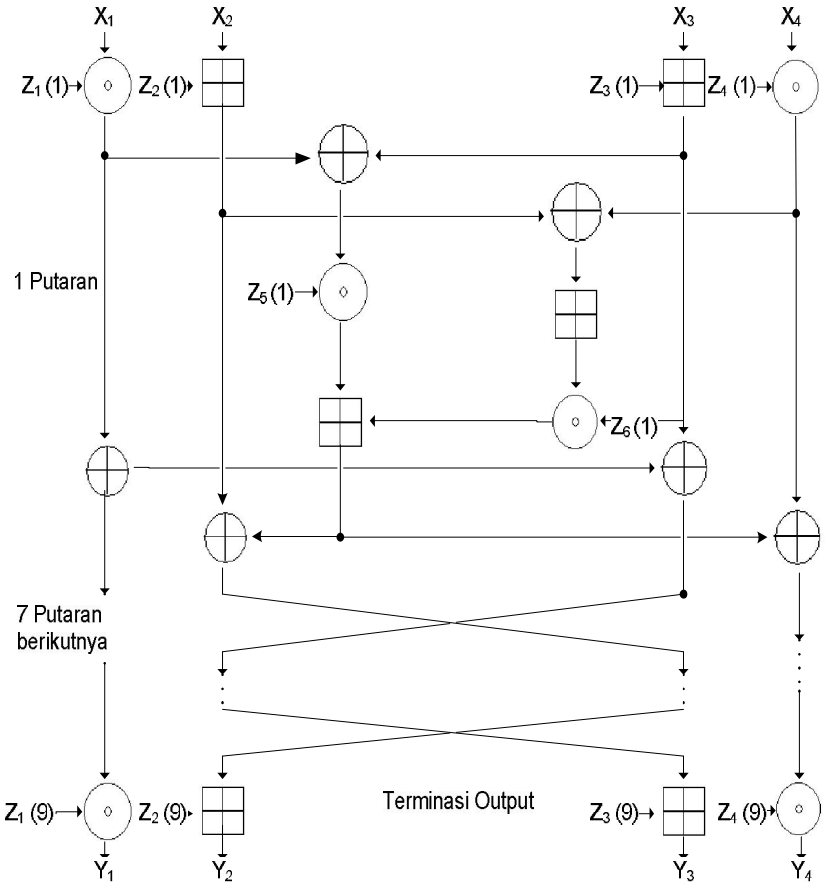
C = adalah pesan rahasia

K = adalah kunci enkripsi atau dekripsi

IDEA (*International Data Encryption Algorithm*) merupakan algoritma simetris yang beroperasi pada sebuah blok pesan terbuka dengan lebar 64-bit. Dan menggunakan kunci yang sama , berukuran 128-bit, untuk proses enkripsi dan dekripsi. Pesan rahasia yang dihasilan oleh algoritma ini berupa blok pesan rahasia dengan lebar atu ukuran 64-bit. Pesan dekripsi menggunakan blok penyandi yang sama dengan blok proses enkripsi dimana kunci dekripsinya diturunkan dari dari kunci enkripsi.

Algoritma ini menggunakan operasi campuran dari tiga operasi aljabar yang berbeda, yaitu XOR, operasi penjumlahan modulo 2^{16} dan operasi perkalian modulo ($2^{16} + 1$) . Semua operasi ini digunakan dalam pengoperasian sub-blok 16-bit.

Algoritma ini melakukan iterasi yang terdiri dari atas 8 putaran dan I transformasi keluaran pada putaran ke 9, dimana gambaran komputasi dan transformasi keluaran ditunjukkan oleh gambar sebagai berikut :



Gambar 2.9 Algoritma IDEA

2.4.2. Proses Enkripsi IDEA.

Pada proses enkripsi, algoritma IDEA ini ditunjukkan oleh gambar di atas, terdapat tiga operasi yang berbeda untuk pasangan sub-blok 16-bit yang digunakan, sebagai berikut :

1. XOR dua sub-blok 16-bit bir per bit, yang disimbolkan dengan tanda $\oplus$.
2. Penjumlahan integer modulo $(2^{16} + 1)$ dua sub-blok 16-bit , dimana kedua sub-blok itu dianggap sebagai representasi biner dari integer biasa, yang disimbolkan dengan tanda $\boxplus$.

3. Perkalian modulo ($2^{16} + 1$) dua sub-blok 16-bit, dimana kedua sub-blok 16-bit itu dianggap sebagai representasi biner dari integer biasa kecuali sub-blok nol dianggap mewakili integer 2^{16}, yang disimbolkan dengan tanda ⊙.

Blok pesan terbuka dengan lebar 64-bit , X, dibagi menjadi 4 sub-blok 16-bit, X_1, X_2, X_3, X_4, sehingga $X = (X_1, X_2, X_3, X_4)$. Keempat sub-blok 16-bit itu ditransformasikan menjadi sub-blok 16-bit, Y_2, Y_2, Y_3, Y_4, sebagai pesan rahasia 64-bit $Y = (Y_1, Y_2, Y_3, Y_4)$ yang berada dibawah kendali 52 sub_blok kunci 16-bit yang dibentuk dari dari blok kunci 128 bit.

Keempat sub-blok 16-bit, X_1, X_2, X_3, X_4, digunakan sebagai masukn untuk putaran pertama dari algoritma IDEA. Dalam setiap putaran dilakukan operasi XOR, penjumlahan, perkalian antara dua sub-blok 16-bit dan diikuti pertukaran antara sub-blok 16-bit putaran kedua dan ketiga. Keluaran putaran sebelumnya menjadi masukan putaran berikutnya. Setelah putaran kedelapan dilakukan transformasi keluara yang dikendalikan oleh 4 sub-blok unci 16-bit.

Pada setiap putaran dilakukan operasi-operasi sebagai berikut :

1) Perkalian X_1 dengan sub-kunci pertama
2) Penjumlahan X_2 dengan sub-kunci kedua
3) Pejumlahan X_3 dengan sub kunci ketiga
4) Perkalian X_4 dengan sub kunci keempat
5) Operasi XOR hasil langkah 1) dan 3)

6) Operasi XOR hasil angkah 2) dan 4)

7) Perkalian hasil langkah 5) dengan sub-kunci kelima

8) Penjumlahan hasil langkah 6) dengan langkah 7)

9) Perkalian hasil langkah 8) dengan sub-kunci keenam

10) Penjumlahan hasil langah 7) dengan 9)

11) Operasi XOR hasil langkah 1) dan 9)

12) Operasi XOR hasil langkah 3) dan 9)

13) Operasi XOR hasil langkah 2) dan 10)

14) Operasi XOR hasil langkah 4) dan 10)

Keluaran setiap putaran adalah 4 sub-blok yang dihasilkan pada langkah 11), 12), 13), dan 14) dan menjadi masukan putaran berikutnya.

Setelah putaran kedelapan terdapat transformasi keluaran, yaitu :

1) Perkalian X_1 dengan sub-kunci pertama.

2) Penjumlahan X_2 dengan sub-kuci ketiga.

3) Penjumlahan X_3 dengan sub-kunci kedua.

4) Perkalian X_4 dengan sub-kunci keempat.

Terahir, keempat sub-blok 16-bit 16-bit yang merupakan hasil operasi 1), 2), 3), dan 4) ii digabung kembali menjadi blok pesan rahasia 64-bit.

2.4.3. Proses Deskripsi IDEA.

Proses dekripsi menggunakan algoritma yang sama dengan proses enkripsi tatapi, 52 buah sub-blok kunci yang digunakan masing-masing merupakan hasil turunan 52 buah sub-blok kunci enkripsi. Tabel sub-blok kunci dekripsi yang diturunkan dari sub-blok kunci enkripsi dapat dilihat pada tabel berikut :

Tabel 2.2 sub-blok kunci Enkripsi

Putaran Ke-1	Z_{11} Z_{21} Z_{31} Z_{41} Z_{51} Z_{61}
Putaran Ke-2	Z_{12} Z_{22} Z_{32} Z_{42} Z_{52} Z_{62}
Putaran Ke-3	Z_{13} Z_{23} Z_{33} Z_{43} Z_{53} Z_{63}
Putaran Ke-4	Z_{14} Z_{24} Z_{34} Z_{44} Z_{54} Z_{64}
Putaran Ke-5	Z_{15} Z_{25} Z_{35} Z_{45} Z_{55} Z_{65}
Putaran Ke-6	Z_{16} Z_{26} Z_{36} Z_{46} Z_{56} Z_{66}
Putaran Ke-7	Z_{17} Z_{27} Z_{37} Z_{47} Z_{57} Z_{67}
Putaran Ke-8	Z_{18} Z_{28} Z_{38} Z_{48} Z_{58} Z_{68}
Tranformasi output	Z_{19} Z_{29} Z_{39} Z_{49}

Tabel 2.3 sub-blok kunci Deskripsi

Putaran Ke-1	$\{Z_{19}\}^{-1}$ $-Z_{29}$ $-Z_{39}$ $\{Z_{49}\}^{-1}$ Z_{58} Z_{68}
Putaran Ke-2	$\{Z_{18}\}^{-1}$ $-Z_{38}$ $-Z_{28}$ $\{Z_{48}\}^{-1}$ Z_{57} Z_{67}
Putaran Ke-3	$\{Z_{17}\}^{-1}$ $-Z_{37}$ $-Z_{27}$ $\{Z_{47}\}^{-1}$ Z_{56} Z_{66}
Putaran Ke-4	$\{Z_{16}\}^{-1}$ $-Z_{36}$ $-Z_{26}$ $\{Z_{46}\}^{-1}$ Z_{55} Z_{65}
Putaran Ke-5	$\{Z_{15}\}^{-1}$ $-Z_{35}$ $-Z_{25}$ $\{Z_{45}\}^{-1}$ Z_{54} Z_{64}
Putaran Ke-6	$\{Z_{14}\}^{-1}$ $-Z_{34}$ $-Z_{24}$ $\{Z_{44}\}^{-1}$ Z_{53} Z_{63}
Putaran Ke-7	$\{Z_{13}\}^{-1}$ $-Z_{33}$ $-Z_{23}$ $\{Z_{43}\}^{-1}$ Z_{52} Z_{62}
Putaran Ke-8	$\{Z_{12}\}^{-1}$ $-Z_{32}$ $-Z_{22}$ $\{Z_{42}\}^{-1}$ Z_{51} Z_{61}
Tranformasi output	$\{Z_{11}\}^{-1}$ $-Z_{31}$ $-Z_{21}$ $\{Z_{41}\}^{-1}$

2.4.4. Pembentukan Sub-Kunci.

Sebanyak 52 sub-blok kunci 16-bit untuk proses enkripsi diperolah dari sebuah kunci 128-bit pilihan pemakai. Blok kunci 128-bit dipartisi menjadi 8 sub-blok kunci 16-bit yang lansung dipakai sebagai 8 sub-blok kunci pertama. Kemudian blok kunci 128-bit dirotasi dari kiri 25 poisi untk dipartisi lagi menjad 8 sub-blok kunci 16-bit berikutnya. Proses rotasi dan pertisi itu diulangi lagi smpai diperoleh 52 sub-blok kunci 16-bit, dengan urutan sebagai berikut :

$$Z_{11}\ Z_{21}\ Z_{31}\ Z_{41}\ Z_{51}\ Z_{61}$$

$$Z_{12}\ Z_{22}\ Z_{32}\ Z_{42}\ Z_{52}\ Z_{62}$$

$$Z_{13}\ Z_{23}\ Z_{33}\ Z_{43}\ Z_{53}\ Z_{63}$$

$$Z_{14}\ Z_{24}\ Z_{34}\ Z_{44}\ Z_{54}\ Z_{64}$$

$$Z_{15}\ Z_{25}\ Z_{35}\ Z_{45}\ Z_{55}\ Z_{65}$$

$$Z_{16}\ Z_{26}\ Z_{36}\ Z_{46}\ Z_{56}\ Z_{66}$$

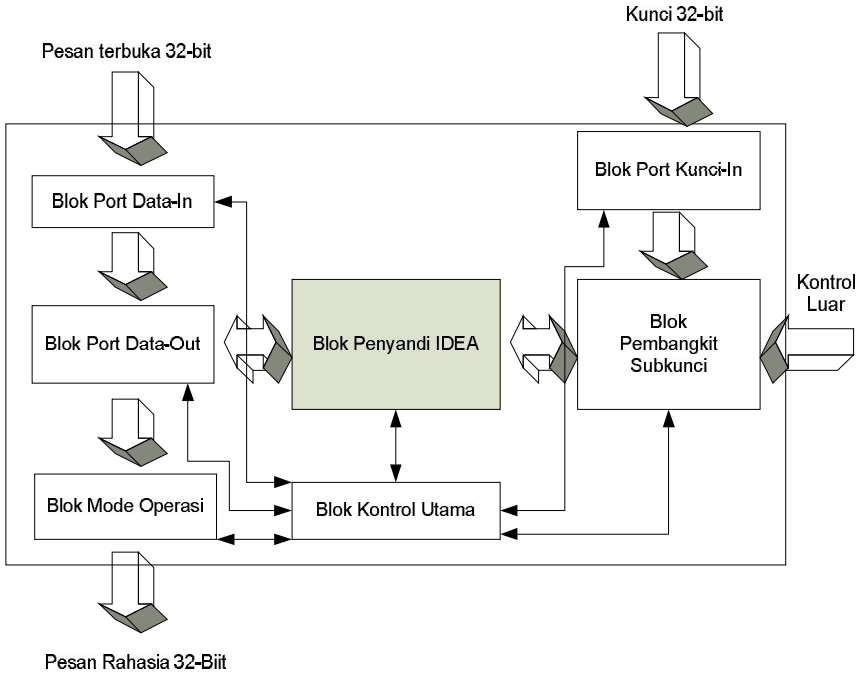
$$Z_{17}\ Z_{27}\ Z_{37}\ Z_{47}\ Z_{57}\ Z_{67}$$

$$Z_{18}\ Z_{28}\ Z_{38}\ Z_{48}\ Z_{58}\ Z_{68}$$

$$Z_{19}\ Z_{29}\ Z_{39}\ Z_{49}$$

2.4.5. Arsitektur Umum Processor Kriptografi IDEA.

Pada gambar berikut diperlihatkan arsitektur atau penggambaran umum sebuah processor yang mengolah sistem keamanan data dengan menggunakan algoritma IDEA;



Gambar 2.10 Arsitektur umum Processor IDEA

Keterangan :

1. Blok penyandi IDEA

   Blok ini berfungsi untuk melakukan proses penyandian data. Jika sub-kunci yang diproses oleh blok ini berupa sub-kunci enkripsi maka pesan yang dihasilkan adalah pesan rahasia (*Chipertext*) dan jika yang diproses berupa sub-kunci dekripsi maka pesan yang dihasilkan adalah pesan sebenarnya (*Plaintext*).

2. Blok pembangkit sub-kunci

   Blok ini berfungsi untuk membentuk 52 buah sub-kunci enkripsi 16 bit dari kunci enkripsi 128 bit membentuk 52 buah sub-kunci dekripsi 16 bit dari kunci dekripsi 128 bit.

3. Blok *port data-in*

   Blok ini berfungsi untuk membaca 2 buah blok data masukan 32 bit dan penyimpananya sebagai blok data masukan 64 bit yang akan dienkripsi atau didekripsi.

4. Blok *port data-out*

   Blok ini berfungsi untuk mengeluarkan blok data keluaran 64 bit yang merupakan hasil enkripsi atau dekripsi engan cara membagi menjadi 2 buah ok data keluaran 32 bit.

5. Blok *port key-in*

   Blok ini berfungsi untuk membaca 4 buah blok kunci 32 bit dan menyimpanya sebagai blok kunci 128 bit.

6. Blok mode operasi

   Blok ini berfungsi untuk menentukan mode operasi yang digunakan paa prses ekripsi dan dekripsi.

7. Blok kontrol

   Blok ini berfungsi untuk mengontrol operasi antara blok fungsional yang menyusun sebuah blok besar seperti sinkronisasi transfer data antar blok.

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*

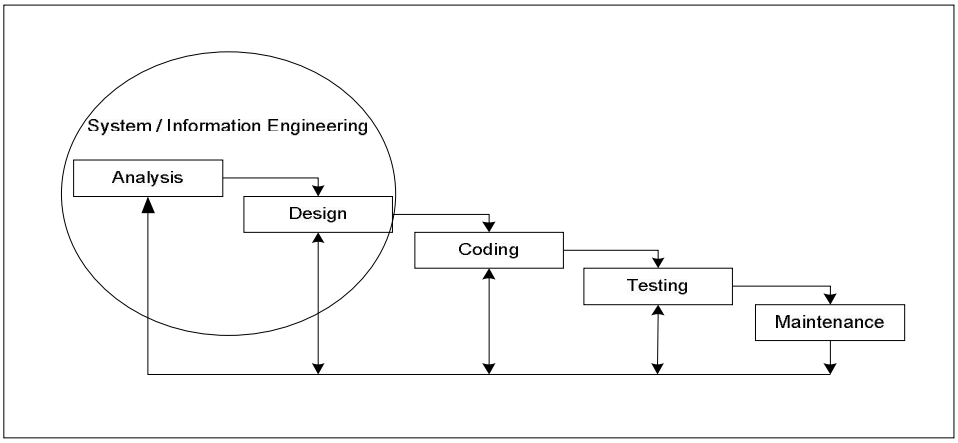
   Tahap penerjemahan data atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman komputer yang ditentukan.

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 2.11 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 dibagun.

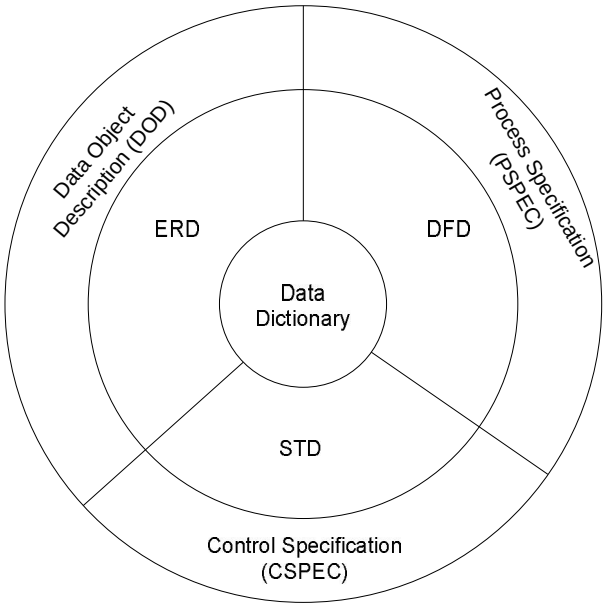
Untuk mencapai sasaran tersebut, model analisis yang ditarik selama analisis terstruktur berlangsung di tujukan dengan Gambar 2.17.

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 2.12 Analisis Terstruktur

2.6. Perangkat Lunak Pendukung

2.6.1. *Microsoft Visual Basic 6.0*.

Microsoft Visual Basic 6.0 merupakan program aplikasi buatan *Microsoft* sangat populer didunia pemrograman. *Visual Basic 6.0* adalah teknik pemrograman yang sangat beda dengan cara pemrograman prosedural.

Selain itu keunggulan lain yang dimiliki oleh *Visual Basic 6.0* adalah teknik pemrograman grafisnya (*Graphical User Interface*) yang sangat handal. Dengan teknik ini tidak perlu lagi kesulitan dalam merancang antarmuka atau

interface untuk program yang dibuat. Karena *Visual Basic 6.0* telah menyediakan bermacam-macam *tool* untuk merancang tampilan program.

Sejarah *Microsoft Visual Basic* diawali dari pengembangan bahasa *Basic* di *Dartmouth College*, Amerika Serikat, pada awal tahun 1960-an. Tahun 1982 *Microsoft* membuat sistem operasi *Ms-Dos* untuk komputer IBM/PC. Di dalamnya disertakan pula bahasa *Basic* yang dikenal sebagai *Quick Basic* (*Qbasic*). Pada tahun 1990-an era *Dos* berlalu diganti era *Windows*. *Microsoft* pun akhirnya membuat *Basic* versi *Windows* yang dikenal dengan *Microsoft Visual Basic*.

2.6.2. Keistimewaan *Microsoft Visual Basic 6.0.*

Sejak dikembangkan pada tahun 80-an, *Visual Basic* kini telah mencapai versinya yang ke-6. Beberapa keistimewaan utama dari *Microsoft Visual Basic 6.0* ini diantaranya adalah :

1. Menggunakan platform pembuatan program yang diberi nama *Developer Studio* yang memilki tampilan dan sarana yang sama dengan *Visual C++* dan *Visual J++*. Dengan begitu dapat bermigrasi atau belajar bahasa pemrograman lainnya dengan mudah dan cepat tanpa harus belajar dari nol lagi.
2. Memiliki *compiler* andal yang dapat menghasilkan file *executable* yang lebih cepat dan lebih efisien dari sebelumnya.
3. Memilki beberapa tambahan sarana *Wizard* yang baru. *Wizard* adalah sarana yang mempermudah di dalam pembuatan aplikasi dengan mengotomatisasi tugas-tugas tertentu.

4. Tambahan kontrol-kontrol baru yang lebih canggih serta peningkatan kaidah struktur bahasa *Microsoft Visual Basic*.
5. Kemampuan membuat *ActiveX* dan fasilitas *Internet* yang lebih banyak.
6. Sarana akses data yang lebih cepat dan andal untuk membuat aplikasi database yang berkemampuan tinggi.
7. *Microsoft Visual Basic 6.0* memiliki beberapa versi atau edisi yang disesuaikan dengan kebutuhan pemakainya.

2.6.3. Spesifikasi *Microsoft Visual Basic 6.0.*

Adapun spesifikasi dari *Microsoft Visual Basic* versi 6.0 adalah :

1. Sistem operasi 32 bit seperti *Windows 9x* atau *Windows NT*.
2. IBM PC atau yang kompatibel dengan prosesor 486 ke atas.
3. RAM 16 MB ke atas.
4. VGA atau resolusi yang lebih tinggi.

2.6.4. Versi Pada *Microsoft Visual Basic 6.0*.

Beberapa versi atau edisi dari *Microsoft Visual Basic 6.0* yang ada diantaranya adalah:

i. *Standard Edition/Learning Edition*. Ini adalah versi standard yang sudah mencakup berbagai sarana dasar dari *Visual Basic 6.0* untuk mengembangkan aplikasi.

ii. *Professional Edition*. Versi ini memberikan berbagai sarana ekstra yang dibutuhkan oleh para programmer professional. Misalnya seperti kontrol-

kontrol tambahan, dukungan untuk pemrograman *Internet*, *compiler* untuk membuat file *help*, serta sarana pengembangan *database* yang lebih baik

iii. *Enterprise Edition*. Versi ini dikhususkan untuk para programmer yang ingin mengembangkan aplikasi *remote computing* atau *client/server*. Biasanya versi ini digunakan untuk membuat aplikasi pada jaringan.

2.7. *Microsoft SQL server 2000*.

Microsoft SQL serevr 2000 ialah perangkat lunak *Relational Database Mangement System* (RDBMS) yang handal. Didesain untuk mendukung proses transaksi data yang besar (seperti order entri yang *online*, inventori, akutansi dan manufaktur). *SQL server 2000* dapat dijalankan pada NT 4.0 *server* atau *microsft 2000 server* dan dapat diinstal juga pada *personal dekstop* di *windows 2000 profesional*, *windows 98* dan *windows milenium*.

2.8. *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.

2.9 Algoritma.

Istilah algoritma berasal dari kata *Algoris* dan *Ritmis*, yang pertama kali diungkapkan oleh Abu Ja'far Mohammed Ibn Musa Al Khowarizmi (tahun 825 sesudah masehi) dalam buku aljabarnya. Dalam buku tersebut, algoritma berarti suatu metode khusus untuk menyelesaikan suatu persoalan. Sedangkan menurut Goodman Hedetniemi, algoritma adalah urutan-urutan terbatas dari operasi-operasi yang terdefinisi dengan baik, yang masing-masin membutuhkan *memory* dan waktu yang terbatas untuk menyelesaikan suatu masalah.

Dalam bidang pemrograman algoritma didefinisikan sebagai suatu metode yang terdiri dari serangkaian langkah-langkah yang terstruktur dan dituliskan secara sistematis yang akan dikerjakan untuk menyelesaikan masalah dengan bantuan komputer.

Beberapa langkah dasar membangun suatu algoritma, yaitu :

- Pernyataan masalah.
- Membangun model.
- Merancang algoritma.
- Menguji kebenaran algoritma.
- Implementasi dengan suatu bahasa pemrograman yang dimengerti *user*.
- Dokumentasi.
- Analisis kompleksitas algoritma.