BAB II

LANDASAN TEORI

2.1 Jaringan Komputer

Secara umum jaringan komputer adalah beberapa komputer yang saling berhubungan dan melakukan komunikasi satu dengan yang lain menggunakan perangkat keras jaringan (*Ethernet card*, *token ring*, *bridge*, *modem* dan sebagainya).

Komputer yang berada dalam suatu jaringan dapat melakukan tukar-menukar informasi atau data dengan komputer lain yang berada dalam jaringan tersebut. Pengguna suatu komputer dapat melihat dan mengakses data pada komputer lain dalam jaringan apabila dilakukan *file sharing*.

Pada era 70-an, jaringan komputer biasanya hanya terdapat pada perusahaan-perusahaan besar. Jaringan tersebut menghubungkan kantor-kantor dan departemen-departemen dalam lingkungan perusahaan tersebut (intranet)

Pada tahun 1974, internet muncul dan membawa perubahan besar. Proyek *ARPA* semula hanya disiapkan untuk membangun jalur komunikasi antara beberapa universitas, pangakalan-pangkalan militer dan perusahaan yang terlibat dalam proyek tersebut.

Saat ini, jaringan internet sudah digunakan orang di berbagai belahan bumi. Sayangnya, selain membawa dampak positif, internet juga membawa dampak negatif. Oleh karena itu, pengguna harus selektif memanfaatkan informasi dan fasilitas yang ada. Perkembangan jaringan komputer di Indonesia sendiri telah mengalami kemajuan pesat meskipun menemui beberapa kendala di dalamnya.

2.1.1 Tipe-tipe Jaringan Komputer

Jaringan komputer dapat dibedakan berdasarkan cakupan geografisnya, yaitu sebagai berikut:

1. *LAN* (*Local Area Nerwork*)

Jaringan *LAN* digunakan untuk menghubungkan komputer yang berada di dalam suatu area yang kecil, misalnya di dalam suatu gedung perkantoran atau kampus. Jarak antar komputer yang dihubungkannya bisa mencapai 5 sampai 10 km. Suatu *LAN* biasanya bekerja pada kecepatan mulai 10 *Mbps* sampai 100 *Mbps*. *LAN* menjadi populer karena memungkinkan banyak pengguna untuk memakai sumber daya secara bersama-sama. Contoh dari sumber daya yang dapat digunakan itu misalnya suatu *mainframe, file server*, printer, dan sebagainya.

2. *MAN* (*Metropolitan Area Network*)

MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. *MAN* menghubungkan *LAN-LAN* yang lokasinya berjauhan. Jangkauan *MAN* bisa mencapai 10 km sampai beberapa ratus kilometer. suatu *MAN* biasanya bekerja pada kecepatan 1,5 sampai 150 *Mbps*.

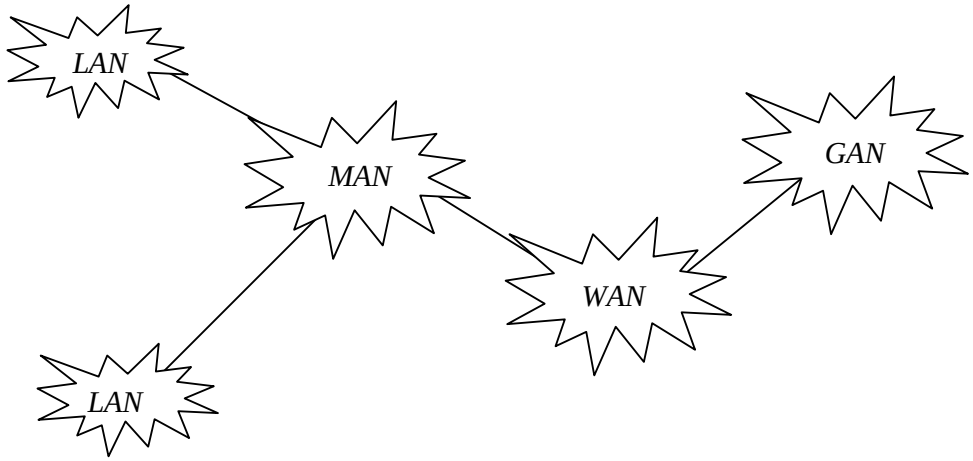
3. *WAN* (*Wide Area Network*)

WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari suatu kota ke kota lain di dalam suatu negara. Cakupan *WAN* bisa meliputi 100 km sampai 1.000 km, dan kecepatan antar kota bisa bervariasi antara 1,5 *Mbps* sampai 2,4 *Gbps*. Dalam *WAN*, biaya untuk peralatan transmisi sangat tinggi, dan biasanya jaringan *WAN* dimiliki dan dioperasikan sebagai suatu jaringan publik.

4. *GAN* (*Global Area Network*)

GAN merupakan suatu jaringan yang menghubungkan negara-negara di seluruh dunia. Kecepatan *GAN* bervariasi mulai dari 1,5 *Mbps* sampai dengan 100 *Gbps* dan cakupannya mencapai ribuan kilometer. Contoh yang sangat baik dari *GAN* adalah Internet.

LAN, *MAN*, *WAN* dan *GAN* dapat berinteraksi satu sama lain. Gambar 2.1 memperlihatkan interaksi antara jaringan-jaringan tersebut.



Gambar 2. 1 Interaksi antara *LAN*, *MAN*, *WAN*, dan *GAN*

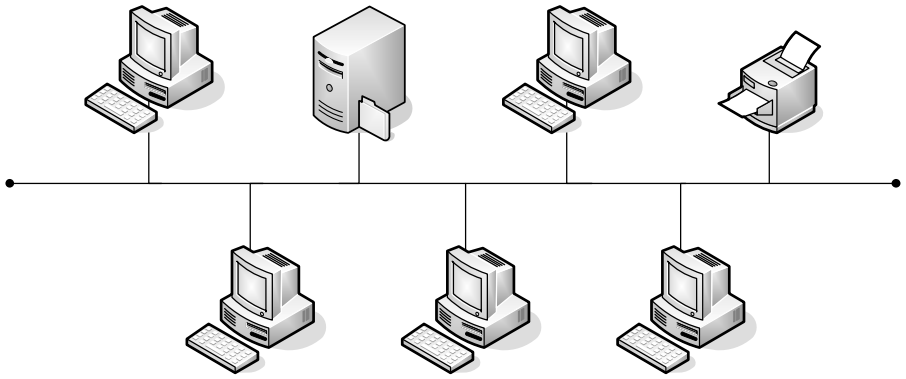
2.1.2 Topologi Jaringan Komputer

Berikut ini akan dijelaskan mengenai topologi fisik yang digunakan di dalam jaringan lokal diantaranya:

1. *Linear Bus* (Garis Lurus)

Topologi *linear Bus* (Garis lurus) terdiri dari satu jalur kabel utama dimana masing-masing ujungnya diberikan sebuah *terminator*. Semua *nodes* pada jaringan (*file server*, *workstation*, dan perangkat lainnya) terkoneksi sebuah kabel

utama *(backbone)*. Jaringan-jaringan *Ethernet* dan *Local talk* menggunakan topologi *linear bus* dan akan diperlihatkan pada gambar 2.2 dibawah ini.



Gambar 2. 2 Topologi *Linear Bus* (Garis Lurus)

Kelebihan dari topologi *Linear Bus* (Garis Lurus) adalah

1. Mudah dalam konfigurasi komputer atau perangakat lain ke dalam sebuah kabel utama.
2. Tidak terlalu banyak menggunakakn kabel dibandingkan dengan topologi *star* atau bintang.

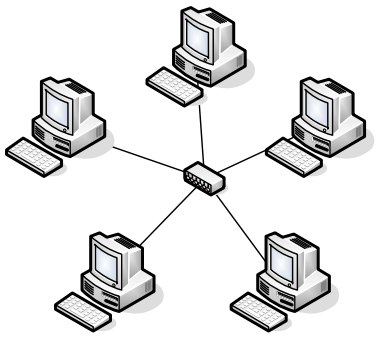
Kekurangan dari topologi *Linear Bus* (Garis Lurus) adalah:

1. Seluruh jaringan akan mati jika ada kerusakan pada kabel utama *(backbone).*
2. Membutuhkan terminator pada kedua sisi dari kabel utamanya.
3. Sangat sulit dalam mengidentifikasi permasalahan jika jaringan sedang jatuh atau rusak.
4. Sangat tidak disarankan dipakai sebagai salah satu solusi pada penggunaan jaringan di gedung besar.

2. *Star* (Bintang)

Topologi model ini dirancang yang mana setiap *nodes* (*file server, workstation*, dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator*.

Data yang dikirim ke jaringan lokal akan melewati *concentrator* sebelum melanjutkan ke tempat tujuannya. *Concentrator* akan mengatur dan mengendalikan keseluruhan fungsi jaringan, dan juga bertindak sebagian *repeater* (penguat aliran data). Konfigurasi pada jaringan model ini menggunakan kabel *Twiated Pair*, dan dapat digunakan pula kabel *coaxial* atau kabel *fibre optic*. Gambar 2.3 di bawah ini adalah skema topologi *star*.



Gambar 2. 3 Topologi *Star* (Bintang)

Kelebihan dari topologi *star* (Bintang)

1. Mudah dalam pemasangan dan pengkabelan.
2. Tidak akan mengakibatkan gangguan pada jaringan pada jaringan ketika akan memasang atau memindahkan perangkat jaringan lainnya.
3. Mudah untuk mendeteksi kesalahan dan memindahkan perangkat-perangkat lainnya.

Kekurangan dari topologi *star* (Bintang)

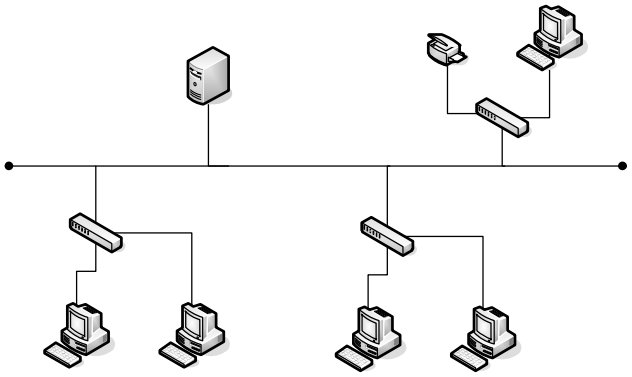
1. Membutuhkan labih banyak kabel dari pada *linear bus*.
2. Membutuhkan *concentrator*, dan bilamana *consentrator* tersebut rusak maka semua *node* yang terkoneksi tidak dapat terdeteksi.
3. Lebih mahal daripada topologi *linear bus*, karena biaya untuk pengadaan *concentrator*.

3. *Ring* (Cincin)

Topologi *Ring* (Cincin) menggunakan teknik konfigurasi yang sama denga topologi *star* tetapi pada pada topologi ini terlihat bahwa jalur media transmisi menyerupai suatu lingkaran tertutup menyerupai cincin (Lingkaran), sehingga diberi nama topologi bintang dalam lingkaran atau *star-wired ring*.

4. *Tree (Pohon)*

Topologi model ini merupakan perpaduan antara topologi *Linear Bus* dan *Star*, yang mana terdiri dari kelompok-kelompok dari *workstation* dengan konfigurasi *star* yang terkoneksi ke kabel utama yang menggunakan topologi *Linear Bus*. Topologi ini memungkinkan untuk pengembangan jaringan yang telah ada, dan memungkinkan untuk mengkonfigurasi jaringan sesuai dengan kebutuhan. Gambar 2.4 akan memperlihatkan interaksi antar komputer.



Gambar 2. 4 Topologi *Tree* (Pohon)

Kelebihan dari Topologi *Tree* (pohon)

1. Proses konfigurasi jaringan dilakukan dari titik ke titik pada masing-masing segmen.
2. Didukung oleh banyak perangkat keras dan perangkat lunak

Kekurangan dari topologi *Tree* (Pohon)

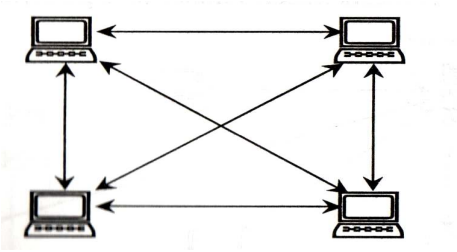
1. Keseluruhan panjang kabel pada tiap-tiap segmen dibatasi oleh tipe kabel yang digunakan.
2. Jika jaringan utama *(backbone)* rusak, maka keseluruhan segmen ikut rusak juga.
3. Sangat relatif sulit untuk di konfigurasi dan proses pengkabelannya dibandingkan topologi jaringan lain.

2.1.3 Model Hubungan pada LAN

1. *Peer-to-Peer*

Model hubungan *peer to peer* memungkinkan user membagi sumber daya yang ada di komputer baik itu berupa *file*, layanan printer dan lain-lain serta mengakses sumber daya yang terdapat pada komputer lain. Namun model ini

tidak mempunyai sebuah *file server* atau sumber daya yang terpusat. Didalam model hubungan *peer to peer* ini, seluruh komputer adalah sama, yang mana mempuyai kemampuan yang sama untuk memakai sumber daya yang tersedia di dalam jaringan model ini di desain untuk jaringan berskala kecil dan menengah. Gambar 2.5 adalah hubungan komputer secara *peer to peer*.



Gambar 2. 5 Model Hubungan *Peer to Peer*

Kelebihan Model Hubugan *Peer to peer*:

1. Tidak terlalu mahal, karena tidak membutuhkan *dedicated file server*.
2. Mudah dalam konfigurasi programnya, hanya tinggal mengatur untuk operasi model hubungan *peer to peer*.

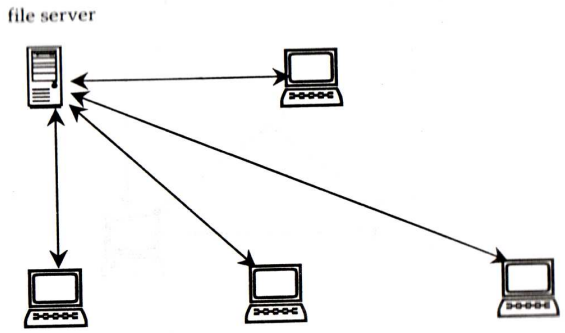
Kekurangan dari Hubungan *Peer to peer*

1. Tidak terpusat, terutama untuk penyimpanan data dan aplikasi
2. Tidak aman, karena tidak menyediakan fasilitas untuk keperluan itu.

2. Client/Server

Model hubungan *Client/Server* memungkinkan jaringan untuk mensentralisasi fungsi dan aplikasi kepada satu atau dua *dedicated file server*. Sebuah *file server* menjadi jantung dari keseluruhan sistem, memungkinkan untuk mengakses sumber daya dan menyediakan keamanan. *Workstation* yang berdiri sendiri dapat mengambil sumber daya yang ada pada *file server*. Model hubungan ini, menyediakan mekanisme untuk mengintegrasikan seluruh komponen yang ada

di jaringan dan memungkinkan banyak penggunaan secara bersama-sama memakai sumber daya pada *file server*. Gambar 2.6 memperlihatkan kinerja hubungan *client/server*.



Gambar 2. 6 Model Hubungan *Client/server*

Kelebihan model hubungan *Client/server*:

1. Terpusat (sumber daya dan keamanan data dikontrol melalui *server*)
2. Skalabilitas
3. Fleksibel
4. Teknologi baru dengan mudah terintegrasi ke dalam system
5. Keseluruhan komponen (*Client, network, server*) dapat bekerja bersama.

Kekurangan model hubungan *client/server*:

1. Mahal
2. Membutuhkan investasi untuk dedicated *file server*
3. Perbaikan (Jaringan besar membutuhkan seorang staff untuk mengatur agar sistem berjalan secara efisien)
4. Berketergantugan.
5. Ketika server jatuh mengakibatkan keseluruhan operasi pada *network* akan jatuh pula.

2.2 Jaringan Multimedia

Multimedia dapat diartikan sebagai penggunaan beberapa media yang berbeda untuk menggabungkan dan menyampaikan informasi dalam bentuk *text*, audio, grafik, animasi, dan video. Istilah multimedia mengacu pada kelas media yang berbeda untuk memakai informasi lalu lintas multimedia yang mengacu pada tranmisi data pada jaringan komunikasi. Sedangkan multimedia berbasis jaringan adalah sistem ini harus terhubung melalui jaringan yang mempunyai bandwidth yang besar. Perbedaannya adalah adanya *sharing* sistem dan pengaksesan terhadap sumber daya yang sama. Contoh: *video converence* dan *video broadcast* tapi permasalahannya adalah bila bandwidth kecil, maka akan terjadi kemacetan jaringan, delay dan masalah infrastruktur yang belum siap

2.2.1 Data *Stream*.

Dalam sistem multimedia terdistribusi, data ditransmisikan (*time dependent*) dan terjadi pertukaran informasi sebagai berikut :

⇨ Sisitem Digital ⇨ Informasi dibagi menjadi beberapa unit (*packets*) ⇨ *Packet-packet* dikirimkan

⇨ *Packet-packet* diterima ⇨ *Packet-packet* disusun ulang ⇨ Informasi disajikan

Transmisi informasi dapat dikategorikan :

1. Berdasarkan mode transmisi

a. *Ansynchronous Trasmission Mode*

1. Komunikasi tanpa batas waktu. Paket mencapai penerima secepat mungkin
2. Paket yang dikirm cepat karena tidak perlu adanya sinkronisasi

3. Informasi untuk *discrete* media dapat ditransmisikan sebagai *ansynchronous data stream*
4. Contoh : transmisi *e-mail*

b. *Synchronous Transmission Mode*

1. Terdapat batas waktu tunda maksimal untuk setiap packet dari suatu data *stream*
2. Butuh sinkronisasi
3. Penerima butuh *buffer* untuk menyimpan data sementara sambil menunggu paket lengkap

c. *Isochronous Transmission Mode*

1. Terdapat batas waktu tunda maksimal dan minimal
2. Melakukan garansi paket diterima dengan baik
3. *Client* memberikan informasi kepada *server* tentang statusnya
4. Membutuhkan *buffer* yang sangat besar

2. Berdasar periode *streaming*

a. *Strongly Periodic Stream*

Interval waktu antara dua *packet* yang berurutan tetap

Contoh : *PCM coded*

b. *Weakly Periodic Stream*

Interval waktu antara dua packet yang berurutan dapat dideskripsikan dengan fungsi periodik

c. *Aperiodic Stream*

Interval waktu tidak beraturan

3. Berdasar ukuran *packet*

a. *Strongly Regular Stream*

Ukuran *packet* konstan

Contoh: *uncompressed audio/video stream*

b. *Weakly Regular Stream*

Ukuran *packet* data berubah secara periodik

Contoh: *MPEG*

c. *Irregular Data Stream*

Ukuran *packet* data tidak tentu

2.3 Kriptografi

Kriptografi atau yang sering dikenal dengan Sebutan ilmu penyandian data, adalah suatu bidang ilmu dan seni (*art and science*) yang bertujuan untuk menjaga kerahasiaan suatu pesan yang berupa data data dari akses oleh orang-orang atau pihak-pihak lain yang tidak berhak sehingga tidak menimbulkan kerugian.

2.3.1 Sejarah Kriptografi

Bidang ilmu Kriptografi ini semula hanya populer dibidang militer dan bidang intelijen untuk menyandikan pesan-pesan panglima perang kepada pasukan yang berada digaris depan, akan tetapi seiring dengan semakin berkembangnya teknologi utamanya teknologi informasi dan semakin padatnya lalu lintas informasi yang terjadi tentu saja semakin menuntut adanya suatu komunikasi data yang aman, bidang ilmu ini menjadi semakin penting. Sekarang

bidang ilmu ini menjadi salah satu isu suatu topik riset yang tidak habis-habisnya diteliti dengan melibatkan banyak peneliti.

Ilmu Kriptografi sebenarnya sudah mulai dipelajari manusia sejak tahun 400 SM, yaitu pada zaman Yunani kuno. Dari catatan bahwa "Penyandian Transposisi" merupakan sistem kriptografi pertama yang digunakan atau dimanfaatkan. Bidang ilmu ini terus berkembang seiring dengan kemajuan peradaban manusia, dan memegang peranan penting dalam strategi peperangan yang terjadi dalam sejarah manusia, mulai dari sistem kriptografi "*Caesar Chiper*" yang terkenal pada zaman Romawi kuno, "*Playfair Cipher*" yang digunakan Inggris dan "*ADFVGX Cipher*" yang digunakan Jerman pada Perang Dunia I, hingga algoritma-algoritma kriptografi rotor yang populer pada Perang Dunia II , seperti *Sigaba / M-134* (Amerika Serikat), *Typex* (Inggris), Purple (Jepang), dan mesin kriptografi legendaris *Enigma* (Jerman).

Induk dari keilmuan dari kriptografi sebenarnya adalah matematika, khususnya teori aljabar yang mendasar ilmu bilangan. Oleh karena itu kriptografi semakin berkembang ketika komputer ditemukan. Sebab dengan penemuan komputer memungkinkan dilakukannya perhitungan yang rumit dan kompleks dalam waktu yang relatif sangat singkat, suatu hal yang sebelumnya tidak dapat dilakukan. Dari hal tersebut lahirlah banyak teori dan algotitma penyandian data yang semakin kompleks dan sulit dipecahkan. Dewasa ini bidang ilmu kriptografi memiliki kemungkinan aplikasi yang sangat luas, mulai dari bidang militer, telekomunikasi, jaringan komputer, keuangan dan perbankan, pendidikan dan singkatnya dimana suatu kerahasiaan data amat diperlukan disitulah kriptografi memegang peranan penting. Produk-produk yamg menggunakan kriptografi

sebagai dasarnya pun cukup beragam, mulai dari kartu ATM, *E-Commerce*, *secure email* dan lain-lain.

2.3.2 Tujuan Kriptografi

Dalam teknologi informasi, telah dan sedang sedang dikembangkan cara-cara untuk menangkal berbagai bentuk serangan semacam penyadapan dan pengubahan data yang dikirimkan. Salah satu cara yang ditempuh mengatasi masalah ini ialah dengan menggunakan kriptografi yang menggunakan transformasi data sehingga data yang dihasilkan tidak dapat dimengerti oleh pihak yang tidak berhak mengakses. transformasi ini memberikan solusi pada dua macam masalah keamanan data, yaitu masalah privasi (*privacy*) dan keotentikan (*authenticatian*).

Privasi mengandung arti bahwa data yang dikirimkan hanya dapat dimengerti informasinya oleh penerima yang sah atau berhak. Sedangkan keotentikan mencegah pihak ketiga untuk mengirimkan data yang salah atau mengubah data yang dikirimkan.

Adapun tujuan sistem kriptografi adalah sebagai berikut :

1. *Confidentiality*

   Yaitu memberikan kerahasiaan pesan dan menyimpan data dengan menyembunyikan informasi lewat teknik-teknik enripsi.

2. *Massage Integrity*

   Yaitu memberikan jaminan untuk tiap bagian bahwa pesan tidak akan mengalami perubahan dari saat data dibuat/dikirim sampai dengan saat data tersebut dibuka.

3. *Non-repudiation*

Yaitu memberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang apbila ia mencoba menyangkal memiliki dokumen tersebut.

4. *Authentication*

Yaitu memberikan dua layanan. Pertama mengidentifikasikan keaslian suatu pesan dan memberikan jaminan keotentikannya. Kedua untuk menguji identitas seseorang apabila ia akan memasuki sebuah sistem.

2.3.3 Klasifikasi Algoritma Kriptografi

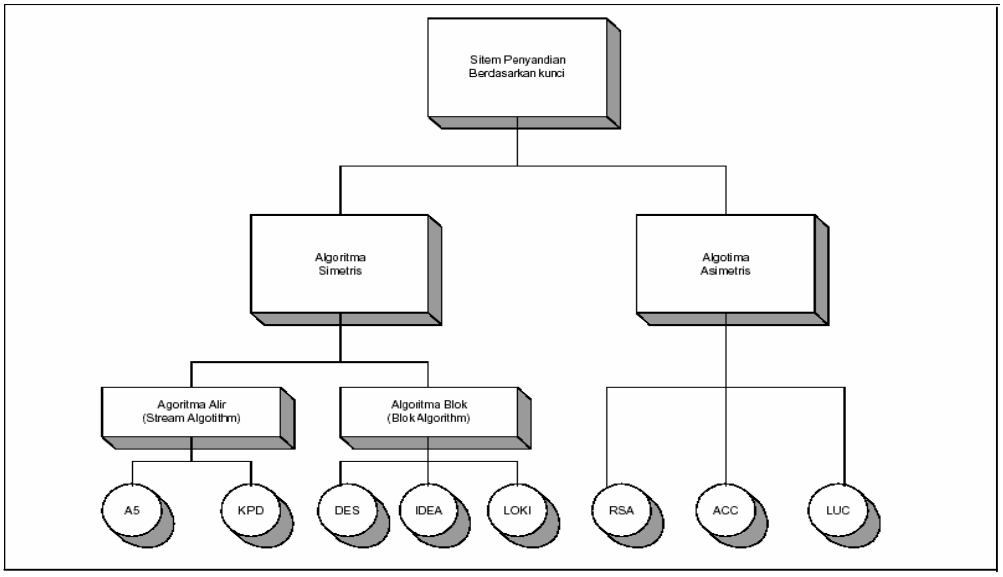
Tingkat keamanan suatu algoritma diperoleh dengan menyembunyikan secara rahasia bagaimana algoritma itu bekerja, algoritma ini disebut dengan algoritma rahasia (*restricted algorithm*). Pada awalnya algoritma jenis ini yang berkembang, namun algoritma jenis ini mempunyai banyak kelemahan seperti setiap 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 sebagian 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 algoritmasimetris (*simetric algorithm*) dan algoritma kunci publik (*public-key algorithm*). Perbedaan utama antara *Symetric algorithm* dengan *public-key algoritm* adalah pada kunci enkripsi *K1* dan kunci dekripsi *K2* . Selain itu juga terdapat perbedaandalam kecepatan proses dan keamanannya. Berikut ini bagan suatu sistem penyandian berdasarkan kunci beserta contoh-contoh algoritma yang digunakan. Gambar 2.7 memperlihatkan system penyandian berdasarkan kunci.

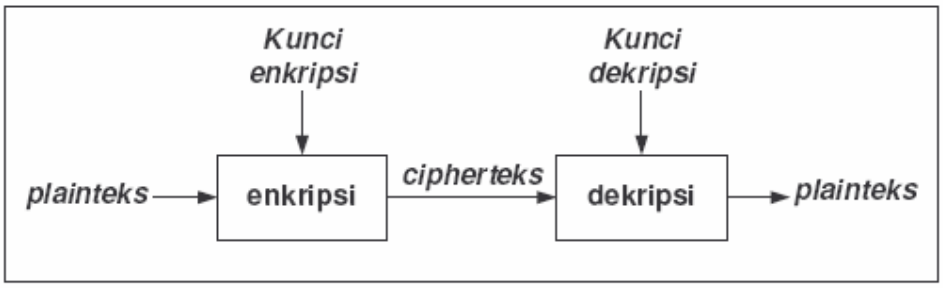


Gambar 2.7 Blok Diagram Pembagian Sistem Kriptografi Berdasarkan Kunci

2.4 Pengertian Enkripsi dan Dekripsi

2.4.1 Pengertian Dasar

Suatu pesan yang tidak disandikan disebut sebagai *plaintext* ataupun dapat disebut juga sebagai *cleartext*. Proses yang dilakukan untuk mengubah *plaintext* ke dalam *ciphertext* disebut enkripsi (*encryption*) atau *encipherment*. Sedangkan proses untuk mengubah *ciphertext* kembali ke *plaintext* disebut dekripsi (*decryption*) atau *decipherment*. Secara sederhana istilah-istilah di atas terdapat pada gambar 2.8 di bawah ini.



Gambar 2.8 Proses Enkripsi dan Dekripsi Secara Umum

Cryptography adalah suatu ilmu ataupun seni mengamankan pesan, dan dilakukan oleh *cryptographer*. Sedangkan, *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) ... [2.1]

$DK(C) = M$ (Proses Dekripsi) ... [2.2]

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 pesan.

2.4.2 *Cryptographic system (cryptosystem)*

Suatu *cryptosystem* terdiri dari sebuah algoritma, seluruh kemungkinan *plaintext, ciphertext* dan kunci-kunci. Secara umum *cryptosystem* dapat digolongkan menjadi dua buah, yaitu :

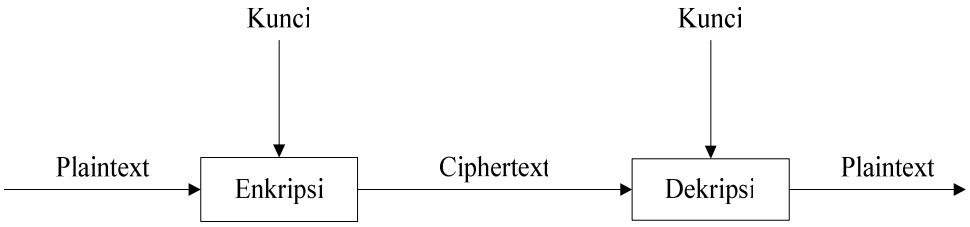
1. *Symmetric Cryptosystem*

Dalam *symmetric cryptosystem* ini, kunci yang digunakan untuk proses enkripsi dan dekripsi pada prinsipnya identik $K1 = K2 = K$, 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 ciphersystem*.

Tingkat keamanan kriptosistem yang menggunakan algoritma ini sangat ditentukan oleh kerahasiaan kunci *K* yang digunakan. Jika seseorang hendak mengirimkan suatu pesan 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-benaran dan tidak dapat disadap (*secure channel*).

Gambar 2.9 berikut ini menggambarkan teknik *symmetric cryptosystem*.



Gambar 2.9 Teknik *symmetric cryptosystem*

Prinsip kerja dari kriptografi kunci simetrik adalah sebagai berikut :

1. Pengirim dan penerima data/Informasi sepakat menggunakan sistem kriptografi tertentu.
2. Pengirim dan penerima sepakat menggunakan satu kunci tertentu.
3. Dilakukan enkripsi sebelum data dikirim dan dekripsi setelah data dikirim

Tingkat keamanan dari kriptosistem yang menggunakan algoritma ini sangat ditentukan oleh kerahasiaan kunci *K* yang digunakan. Jika seseorang hendak mengirimkan suatu pesan kepada orang lain atau melakukan *secure communication*, orang tersebut harus terlebih dahulu memberitahu kepda pihak

yang dituju kunci *K* yang digunakannya. Hal ini jelas membutuhkan saluran komunikasi yang benar-benar aman (*secure channel*). Faktor inilah yang menjadi kelemahan cara ini yaitu masalah keamanan kunci dan bagaimana mendistribusikan kunci *K* tersebut.

Pada *symmetric cryptosystem* ini dibagi menjadi dua teknik algoritma, yaitu algoritma alir (*stream algorithm*) dan algoritma blok (*block algorithm*).

1) Algoritma Alir (*stream algorithm*)

   Pada algoritma alir, enkripsi aliran data melakukan transformasi bit demi bit pada teks asal dibawah kendali sebuah aliran yang biasanya menggunakan operasi yang bersifat dapat balik seperti halnya operasi *XOR*.

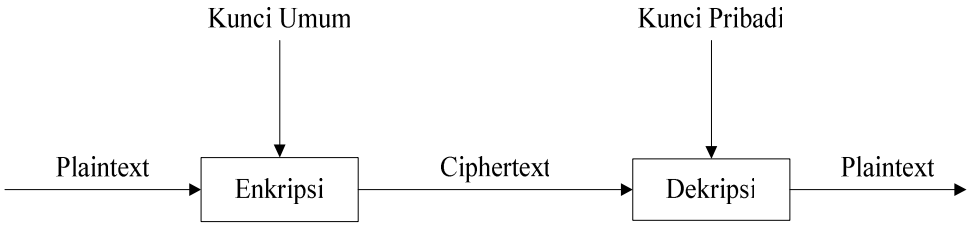
2) Algoritma blok (*blockalgorithm*)

   Pada algoritma blok, enkripsi blok mentrasformasikan bit-bit dalam blok berukuran tertentu, dibawah kontrol sebuah kunci yang biasanya mempunyai ukuran bit yang sama dengan ukuran blok bit yang dienkripsi.

2. *Assymmetric Cryptosystem*

Dalam *assymmetric cryptosystem* 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 surat tersebut, ia perlu mendekripsikan surat itu dengan kunci privatnya.

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*. Gambar 2.10 berikut ini menggambarkan teknik as*symmetric cryptosystem*.



Gambar 2.10 Teknik *Assymmetric Cryptosystem*

Setiap *cryptosytem* yang baik harus memiliki karakteristik sebagai berikut

1. Keamanan sistem terletak pada kerahasiaan kunci dan bukan pada kerahasiaan algoritma yang digunakan.
2. *Cryptosystem* yang baik memiliki ruang kunci (*keyspace*) yang besar.
3. *Cryptosystem* yang baik akan menghasilkan *ciphertext* yang terlihat acak dalam seluruh tes statistik yang dilakukan terhadapnya.
4. *Cryptosystem* yang baik mampu menahan seluruh serangan yang telah dikenal sebelumnya

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

Hingga saat ini masih banyak orang yang menggunakan *cryptosystem* 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 berbagai aspek penting dari *Symmetric Cryptosystem* dengan *Assymmetric Cryptosystem*. Tabel 2.1 menerangakan perbedaan *Symmetric Cryptosystem* dengan *Assymmetric Cryptosystem*

Tabel 2.1 Perbedaan *Symmetric Cryptosystem* dengan *Assymmetric Cryptosystem*

Symmetric Cryptosystem	***Assymmetric Cryptosystem***
Yang dibutuhkan untuk bekerja : 1) Algoritma yang sama dengan kunci yang sama dapat digunakan untuk proes enkripsi maupun dekripsi. 2) Pengirim dan penerima harus membagi algoritma dan kunci yang sama.	Yang dibutuhkan untuk bekerja : 1) Algoritma yang digunakan untuk enkripsi dan dekripsi dengan sepasang kunci, satu untuk ekripsi satu untuk dekipsi. 2) Pengirim dan penerima harus mempunyai sepasang kunci yang cocok.
Yang dibutuhkan untuk keamanan : 1) Kunci harus dirahasiakan. 2) Adalah tidak mungkin atau sangat tidak praktis untuk menterjemahkan Informasi yang telah dienkripsi. 3) Pengetahuan tentang algoritma dan sampel dari kata yang terenkripsi tidak mencukupi untuk menentukan kunci	Yang dibutuhkan untuk keamanan : 1) Salah satu kunci harus dirahasiakan. 2) Adalah tidak mungkin atau sangat tidak praktis untuk menterjemahkan informasi yag telah dienkripsi. 3) Pengetahuan tentang algoritma dan sampel dari kata yang terenkripsi tidak mencukupi untuk menentukan kunci.

2.5 Tanda Tangan Digital

Tanda tangan digital adalah istilah yang sering bermakna ambigu. Kadang, istilah ini diartikan sebagai bagian dari tanda tangan elektronik. Tetapi sebagian orang menggunakan istilah ini sebagai sesuatau yang sejajar dengan tanda tangan elektronik. *U.S. Electronic Signatures in Global and National Commerce Act* yang diselenggarakan pada tahun 2000 menggunakan tanda tangan elektronik saat tanda tangan digital didiskusikan, mengilustrasikan kebingungan secara legal. Tanda tangan elektronik berarti sebuah suara elektronik, simbol, atau proses, secara logika dihubungkan dengan kontrak atau lainnya dan dieksekusi atau diadopsi oleh seseorang dengan maksud menandai rekaman tersebut.

Tanda tangan digital berarti sebuah tanda tangan yang berbasiskan kepada skema kriptografi. Banyak yang sudah diajukan, beberapa sudah ditemukan dan sudah tidak digunakan lagi. Beberapa sudah dipatenkan, beberapa paten sudah tidak berlaku lagi, dan ada beberapa perbedaan pendapat dalam hal ini dengan tujuan komersil.

Tanda tangan digital ini menggunakan algoritma kunci nirsimetri dan biasanya menggunakan skema *public key infrastructur* (*PKI*) dimana kunci publik yang digunakan dalam skema tanda tangan terikat dengan pengguna oleh sebuah setifikat identitas digital yang dikeluarkan oleh sebuah organisasi yang memiliki otoritas untuk mengeluarkan sertifikat, biasanya dikelola oleh perusahaan komersil pihak ketiga. Sistem *PKI* memiliki tujuan untuk menyamarkan informasi pengguna (nama, alamat, no telepon, dan lain-lain) menjadi sebuah kunci publik, ide dasarnya cukup dekat dengan tugas notaris. Ada beberapa skema tanda tangan, yang umumnya menggunakan dua algoritma, satu untuk memberi tanda tangan

dan yang lainnya untuk melakukan pengecekan keaslian tanda tangan tersebut. Keluaran dari proses tanda tangan disebut juga dengan tanda tangan digital.

Fungsi tanda tangan pada dokumen kertas juga diterapkan untuk otentikasi pada data digital seperti pesan yang dikirim melalui saluran komunikasi dari dokumen elektronis yang disimpan didalam memori komputer. Tanda tangan pada data digital ini dinamakan tanda tangan digital (*digital signature*). Yang dimaksud tanda tangan digital bukanlah tanda tangan yang di digitasi dengan alat *scanner*, tetapi suatu nilai kriptrografis yang bergantung pada pesan dan pengirim pesan (hal ini kontras dengan tanda tangan pada dokumen kertas yang bergantung hanya pada pengirim yang selalu sama untuk semua dokumen). Dengan tanda tangan digital, maka integritas data dapat dijamin, disamping itu ia juga digunakan untuk membuktikan asal pesan (keabsahan pengirim), dan nirpenyangkalan.

Cara membuat tanda tangan digital adalah sebagai berikut:

1. Suatu pesan dibuat nilai *hash*-nya menggunakan fungsi *hash* satu arah.
2. Kemudian nilai *hash* tadi dienkrip menggunakan kunci privat milik pihak yang menandatangani, hasilnya adalah tanda tangan digital.

2.5.1 Keuntungan Tanda Tangan Digital

Ada tiga alasan utama untuk memasukan tanda tangan digital dalam komunikasi.

2.5.1.1 *Authentication*

Sistem kriptografi kunci publik memperbolehkan pengenkripsian pesan dengan kunci privat pengguna. Pesan itu sendiri butuh untuk dikirimkan dalam bentuk *cipherteks*. Bila *hash* dari sebuah dokumen di bangun dan dienkripsi, dokumen tersebut tidak dapat di deteksi kecuali dengan mengubah nilai *hash*-nya,

yang mana, bila algoritma yang berkualitas digunakan, secara komputasi tidak dapat dilakukan. Dengan mendekripsi fungsi *hash* menggunakan kunci publik yang dimiliki pengirim, dan mengecek hasilnya dengan *hash* yang baru dibangun, pihak penerima dapat menkonfirmasi bahwa enkripsi tersebut dibuat dengan kunci privat pengirim, dan pesan tersebut tidak diubah atau ditambahkan sejak pesan tersebut diberikan tanda tangan. Penerima tidak akan bisa yakin bahwa yang memberikan tanda tangan adalah benar-benar orang yang seharusnya mengirim pesan tersebut.

Hal yang penting dari keyakinan akan athentifikasi adalah tentunya mengenai hal-hal yang berhubungan dengan finansial. Sebagai contoh, andaikan kantor cabang dari sebuah bank mengirim perintah kepada bank pusat dalam bentuk (a,b) dimana a adalah nomor *account* dan b adalah jumlah yang akan diberikan pada *account* tersebut. Pelanggan yang penuh tipu daya mungkin akan mendepositkan Rp100.000, mengobservasi hasil transmisi dan pengulangan transmit (a,b), mendapatkan deposit setiap saat dan menjadi kaya dengan cepat. Ini adalah contoh dari *replay attack*.

2.5.1.2 *Integrity*

Kedua belah pihak akan selalu menginginkan untuk yakin bahwa pesan yang dikirim tidak diubah atau ditambah-tambahkan dalam pengiriman. Enkripsi dari pesan membuatnya menjadi sulit untuk membacanya, namun pihak ketiga masih dapat menambahkan pesan tersebut tanpa membacanya. Sebagai contoh adalah serangan *homomorphism*

2.5.1.3 *Non-repudiation*

Dalam konteks kriptografi, kata repudisi bermaksud pada pernyataan tidak memiliki tanggung jawab terhadap pesan tersebut. Penerima pesan harus memaksa sang pengirim untuk membeikan tanda tangannya dalam pesan yang akan dikirimkan untuk mempersulit *repudition*, selama sang penerima dapat menunjukkan tanda tangan tersebut pada pihak ketiga. Bagaimanapun, kehilangan kunci privat berarti semua tanda tangan digital yang menggunakan kunci tersebut adalah mencurigakan. Menyadari bahwa kemungkinan kehilangan kunci tersebut dapat terjadi bukanlah bagian dari permasalahan kriptografi, melainkan *human error*, dan tidak ada solusinya.

2.5.2 Implementasi dari Tanda Tangan Digital kunci publik

Skema tanda tangan digital kunci publik bergantung pada kriptografi kunci publik. Dalam kriptografi kunci publik, setiap pengguna memiliki sepasang kunci: satu bersifat publik dan yang satunya bersifat privat. Kunci yang bersifat publik didistribusikan, akan tetapi kunci yang bersifat privat dirahasiakan oleh pengguna, kebutuhan lainnya adalah kunci ini tidak boleh dapat ditemukan secara komputasi apabila diketahui kunci publiknya. Umumnya, tanda tangan digital mencakupi tiga algoritma :

1. Algoritma pembangkit kunci
2. Algoritma pemberi tanda tangan
3. Algoritma verifikasi tanda tangan

Pembangkitan kunci pada algoritma simetri jauh lebih mudah dari pada pembangkitan kunci pada algoritma kunci publik. Karena kunci simetri umumnya

rangkaian bit atau rangkaian karakter, maka setiap pengguna dapat membangkitkan kuncinya sendiri.

Masalah utama dalam pembangkitan kunciadalah bagaimana membuat kunci yang tidak dapat diprediksi. Metode yang dapat digunakan untuk menjawab hal ini adalah dengan teknik manual (misalnya pelemparan dadu atau koin), pembangkitan dari data pribadi (misalnya PIN), atau menggunakan pembangkti kunci acak. Pada algoritma kunci publuk, pembangkitan merupakan masalah tersendiri karena pembangkitan kunci membutuhkan perhitungan matematis yang rumit. Selain itu, karena pembangkitan bilangan prima yang besar juga untuk membentuk kunci. Oleh karena itu, pada algoritma kunci publik juga dibutuhkan program khusus untuk membangkitkan kunci. Masalah yang timbul di sini adalah kepercayaan kepercayaan pengguna terhadap program tersebut.

Algoritma untuk membangkitan pasangan kunci adalah sebagai berikut :

1. Pilih dua buah bilangan prima sembarang, p dan q.
2. Hitung $n = p.q$ (sebaiknya $p \neq q$ maka $n = p^2$ sehingga p dapat diperoleh dengan menarik akar pangkat dua dari n).
3. Hitung $\phi(n) = (p - 1)(q - 1)$.
4. Pilih kunci publik, e, yang relatif prima terhadap $\phi(n)$.
5. Bangkitkan kunci privat dengan menggunakan persamaan $e.d \equiv 1 \pmod{\phi(n)}$. Perhatikan bahwa $e.d \equiv 1 \pmod{\phi(n)}$ ekivaken dengan $e.d \equiv 1 + k\phi(n)$, sehingga secara sederhana d dapat dihitung dengan

$$d = \frac{1 + k\phi(n)}{e}$$

Hasil dari algoritma di atas, yaitu kunci publik adalah pasangan (e, n) dan kunci privat adalah pasangan (d, n).

Sebagai contoh, bayangkan sebuah situasi dimana Bob mengirimkan pesan kepada Alice dan Alice ingin yakin bahwa pesan tersebut benar-benar dari Bob. Bob mengirimkan pesan ke Alice, memasukkan tanda tangan digital. Tanda tangan digital tersebut dibangun dari kunci privat Bob. Pada pihak penerima, Alice dapat memeriksa apakah benar pesan tersebut datang dari Bob atau bukan dengan menjalankan algoritma verifikasi pada pesan tersebut sekaligus dengan tanda tangan digitalnya, dengan menggunakan kunci publik Bob. Apabila cocok, maka Alice dapat yakin bahwa pesan tersebut benar-benar dari Bob.

2.5.3 Hubungan Tanda Tangan Digital dengan Enkripsi

Tanda tangan digital menggunakan tahnik enkripsi namun algoritma yang digunakan belum tentu langsung cocok dengan keperluan untuk membuat tanda tangan digital, banyak metode-metode yang lebih efesien yang dapat digunakan. Tentu saja sebuah dokumen dapat saja dikirimkan dalam bentuk terenkripsi melalui jalur komunikasi publik seperti pesan-pesan lainnya. Yang lebih umum terjadi, Bob pertama-tama menyodorkan fungsi *hash* kriptografi yang berkualitas pada pesan tersebut, lalu secara digital memberikan tanda pada hasil fungsi *hash*. Sebuah *hash* yang tidak aman dapat menurunkan keamanan tanda tangan digital. Sebagai contoh, apabila dimungkinkan untuk membangun kekacauan *hash*, dapat dimungkinkan untuk memalsukan tanda tangan digital. Ada beberapa alasan untuk memberikan tanda pada *hash* yang baik (atau *message digest*) daripada ke seluruh dokumen.

1) Untuk efisiensi : Tanda tangan akan menjadi jauh lebih pendek dan menyimpan banyak waktu karena *hash* dibangun dengan jauh lebih cepat daripada memberikan tanda tangan.

2) Dokumen ditujukan untuk dibaca oleh orang lain : sebagai contoh, diploma, sertifikat identitas, sertifikat kelahiran, SIM, dll. Dokumen ini harus berupa *plainteks,* namun mengandung tanda tangan digital. Tanda tangan digital tersebut dapat digunakan untuk memeriksa bahwa dokumen tersebut asli atau tidak diubah-ubah.
3) Untuk integritas

2.5.4 Tanda Tangan dengan menggunakan Fungsi *Hash*

Penandatanganan pesan dengan cara mengenkripsinya selalu memberikan dua fungsi berbeda, yaitu kerahasiaan dan otentikasi. Pada beberapa kasus, sering kali otentikasi yang diperlukan, tetapi kerahasiaan tidak. Maksudnya, pesan tidak perlu dienkripsikan, sebab yang dibutuhkan hanya otentikasi saja.

Hanya kriptografi kunci publik yang cocok dan alami untuk pemberian tanda tangan digital dengan menggunakan fungsi *hash*. Hal ini disebabkan karena skema tanda tangan digital berbasis kunci publik dapat menyelesaikan masalah *non-repudiation* (baik penerima dan pengirim pesan mempunyai pasangan kunci masing-masing).

2.5.4.1 Proses pemberian Tanda Tangan Digital (*Signing*)

Pengirim pesan mula-mula menghitung *message digest* dari pesan *message digest* (*MD*) diperoleh dengan mentransformasikan pesan *M* dengan mengunakan fungsi *hash* satu arah *H*,

$$MD = H(M)$$

Selanjutnya, *message digest MD* dienkripsi dengan algoritma kunci publik dan menggunakan kunci privat (*SK*) si pengirim. Hasil enkripsi inilah yang dinamakan tanda tangan digital *S*,

$S = E_{SK}(MD)$

Kemudian, tanda tangan digital *S* diletakan ke pesan *M* (dengan cara menyambung atau *append*) *S*, lalu keduanya dikirim melalui saluran komunikasi. Dalam hal ini, kita katakan bahwa pesan *M* sudah ditandatangani dengan tanda tangan digital *S*.

Di tempat penerima, tanda tangan diverifikasi untuk membuktikan keontetikannya dengan cara berikut :

1. Tanda tangan digital *S* dienkripsi dengan menggunakan kunci publik (*PK*) pengirim pesan, menghasilkan *message digest* semula, *MD*, sebagai berikut : $MD = D_{PK}(S)$
2. Penerima kemudian mengubah pesan *M* menjadi *message digest MD'* menggunakan fungsi *hash* satu arah yang sama dengan fungsi *hash* yang digunakan oleh pengirim.
3. Jika *MD' = MD*, berarti tanda tangan yang diterima otentik dan berasal dari pengirim yang benar.

Mekanisme kerja untuk menghasilkan *Digital Signature* tersebut adalah sebagai berikut :

a. Proses *hashing algorithm* akan mengambil intisari dari isi pesan yang akan dikirim, memprosesnya secara *one-way hash* dan menghasilkan *hashed* data. Kemudian *hashed* data tersebut di enkripsi menggunakan *private key* dan menghasilkan apa yang disebut dengan *Digital Signature*. Di dalam *Digital Signature* tersebut juga disertakan metode *hashing algorithm* yang digunakan.

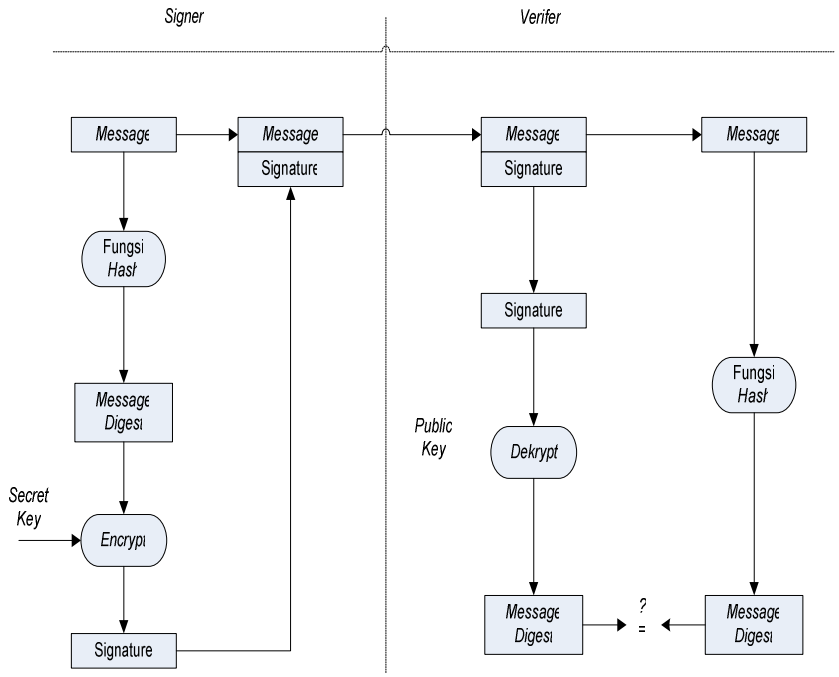
b. Kemudian *Digital Signature* tersebut dikirimkan bersama isi pesan tersebut.

c. Sesampainya di penerima, akan dilakukan proses *hashing algorithm* terhadap isi pesan tersebut. Proses one-way *hash* persis seperti yang dilakukan saat pengiriman, karena algoritmanya turut dibawa dalam *Digital Signature*. Dari proses tersebut menghasilkan *hashed*-data sekunder.

d. Secara pararel, *Digital Signature* yang diterima tadi langsung didekripsi oleh *public key*. Hasil dekripsi tersebut tentunya akan memunculkan *hashed* data yang serupa seperti *hashed* data sebelum dienkripsi oleh pengirim pesan. *Hashed* data disebut *hashed* data primer.

e. Proses selanjutnya adalah memperbandingkan *hashed* data primer dengan *hashed* data sekunder. Jika saja saat diperjalanan ada *hacker* yang mengubah atau menyadap isi pesan, maka *hashed* data sekunder akan berbeda dengan *hashed* data primer. Sehingga mekanisme *Digital Signature* tersebut akan menyampaikan peringatan bahwa telah terjadi 'sesuatu' di tengah jalan.

Skema tanda tangan digital yang merupakan fungsi *hash* ditunjukan pada Gambar 2.11 di bawah ini.



Gambar 2.11 Otentikasi dengan Tanda Tangan Digital
yang menggunakan fungsi *Hash* satu arah

Otentikasi pesan dijelaskan sebagai berikut :

1. Apabila pesan *M* yang diterima sudah berubah, maka *MD'* yang dihasilkan dari fungsi *hash* berbeda dengan *MD* semula. Ini berarti pesan tidak asli
2. Apabila pesan *M* tidak berasal dari orang sebenarnya, maka *message digest MD* yang dihasilkan berbeda dengan *message digest MD'* yang dihasilkan pada proses verifikasi (hal ini karena kunci publik yang digunakan oleh penerima pesan tidak berkoresponden dengan kunci privat pengirim).

3. Bila *MD = MD'*, ini berarti pesan yang diterima adalah pesan yang asli (*message digest authentication*) dan orang yang mengirim adalah orang yang sebenarnya (*user authentication*).

Pengirim pesan tidak dapat menyangkal pesan yang ia kirim, sebab tanda tangan digital daptat digunakan untuk melakukan nirpenyangkalan. Andaikan pengirim berbohong telah mengirim pesan. Sangkalan dari pengirim dapat dibantah dengan cara, yaitu sebagai berikut : jika ia tidak mengirim pesan, berarti ia tidak mengenkripsi *message digest* dari pesan dengan kunci privatnya. Faktanya, kunci publik yang berkoresponden dengan kunci privat pengirim menghasilkan *MD = MD'* ini berarti *message digest* memang benar dienkripsi oleh pengirim sebab hanya pengirim yang mengetahui kunci privatnya sendiri

2.5.4.2 Tanda Tangan Digital dengan menggunakan *RSA*

Langkah-langkah penandatanganan :

1. Pengirim menghitung nilai *hash* dari pesan *M* yang akan dikirim, misalkan nilai *hash* dari *M* adalah *h*.
2. Pengirim mengenkripsi *h* dengan kunci privatnyua menggunakan persamaan enkripsi RSA : $\boldsymbol{S = h^{SK} mod\ n}$. yang dalam hal ini *SK* adalah kunci privat pengirim dan *n* adalah *modulus* (*n = pq*, *p* dan *q* adalah dua buah bilangan prima)
3. pengirim mentransmisikan *M* + *S* ke penerima.

Langkah-langkah pemverifikasian :

1. Penerima menghitung nilai *hash* dari pesan *M* yang akan dikirm, misalkan nilai *hash* dari *M* adalah *h'*.

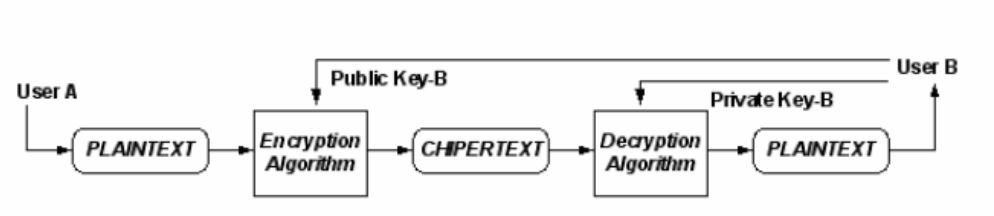
2. Penerima melakukan dekripsi terhadap tanda tangan *S* dengan kunci publik si pengirim menggunakan perssamaan dekripsi *RSA* : $\boldsymbol{h = S^{PK} mod\ n}$. yang dalam hal ini *PK* adalah kunci privat pengirim dan *n* adalah modulus (*n = pq*, *p* dan *q* adalah dua buah bilangan prima).
3. Penerima membandingkan *h* dengan *h'*. Jika *h = h'* maka tanda tangan digital adalah otentik. Jika tidak sama, maka tanda tangan tidak otentik sehingga pesan dianggap tidak asli lagi atau pengirimnya bukan orang yang sebenarnya.

2.6 Algoritma Kunci Publik

Konsep dasar dari algoritma ini adalah menggunakan dua buah kunci yang berbeda untuk proses enkripsi dan dekripsinya. sehingga disebut sistem *asitmetris.* Walaupun demikian, kedua kunci tersebut masih memiliki hubungan matematis. Masing-masing pemakai dalam jaringan komunikasi mengumumkan kunci (kunci publik) untuk proses enkripsi. sedangkan satu kunci lainnya dijaga kerahasiaannva (kunci privat) dan hanva pemakai tersebut yang boleh mengetahuinva. Apabila ada pemakai lain yang ingin mengirim pesan ke pemakai X. maka pemakai lain tersebut akan mencari kunci publik pemakai X dan daftar kunci publik yang diumumkan, untuk kemudian melakukan proses enkripsi dan mengirimkannva pada pemakai X. Pesan yang datang kemudian akan dibuka dengan menggunakan kunci privat yang hana diketahui oleh pemakai X.

Konsep pemakaian dua kunci sistem kriptografi ini pertama kali diperkenalkan oleh Diffie dan Hellman pada tahun 1976. Mereka memperkenalkan metode *bam* dan teknik yang disebut *public key encryption,*

yang dalam hal ini setiap pemakai memiliki *private key* dan *public key.* Munculnva konsep ini sebenarnya merupakan upaya untuk mengatasi masalah kesulitan pada algontma konvensional, yaitu pada pendistnbusian kunci secara aman. Untuk itu dibuatlah mekanisme sistem kriptografi tanpa adanya pendistribusian kunci yang dirasakan kemanannya lebih terjamin. Gambar 2.12 adalah model sederhana proses enkripsi dengan kunci publik.



Gambar 2.12 Model Sederhana dan Enkripsi *Public Key*

Kunci enkripsi *(public key)* diberitahukan kepada umum tanpa membahayakan kerahasiaan kunci dekripsi. Sedangkan kunci dekripsi *(private key)* dijaga kerahasiannva dan hanya diketahui oleh pemiliknya. Hanya mengetahui kunci publik, tidak akan mungkin untuk mengetahui isi pesan maupun mendapatkan atau menurunkan kunci enknipsi. Dengan sifat yang demikian maka sistem kniptografi ini tidak memerlukan saluran khusus yang aman untuk pendistribusian kunci, seperti halnya pada sistem konvensional.

Sistem kniptografi kunci publik perhitungannva berdasarkan pada konsep *trap-door-one-way-function.* Yang dimaksud dengan *one-way-function* adalah suatu fungsi yang mudah dihitung untuk satu arah, namun untuk arah sebaliknva sangat sulit untuk dihitung. Sedangkan *trap-door-one-way-function* adalah suatu *one-way function* yang *inversnya* mudah dihitung apabila diberikan suatu informasi tertentu yang dipergunakan untuk mendesain fungsi tersebut. Fungsi

tersebut hanya mudah dihitung dalam satu arah, dan tanpa infomiasi khusus proses kebalikannva akan memakan waktu yang sangat lama.

Algoritma kriptografi kunci publik harus memenuhi kondisi-kondisi dibawah ini :

1. Metode perhitungan yang mudah di sisi penenima *B*, untuk menghasilkan pasangan *public key* (*PK*) dan *private key* (*SK*)

2. Metode perhitungan yang mudah di sisi penginim *A.* untuk mengetahui *public key B* dan menghasilkan *chipertext C* dari pesan yang dikirimkan *M: C + E [PK, M]*

3. Metode perhitungan yang mudah bagi penenima *B* untuk mendeknipsi *chipertext C* dengan menggunakan *private key*, untuk mendapatkan pesan sebenarnya *M : M = D [SK C] = D [SK E [PK, M]]*.

4. Akan sangat sulit untuk mendapatkan *private key SK*, jika yang diketahui hanva *public key PK*.

5. Sangat sulit untuk mendapatkan pesan M dengan menggunakan *public key* PK dan *chipertext C'*

6. Proses enkripsi membalikkan proses dekripsi: *M = E [PK, D [SK, M]*

Berikut ini langkah-langkah yang harus dilakukan dalam proses pertukaran informasi dengan menggunakan algoritma kunci publik:

1. Masing-masing pemakai di dalam jaringan menghasilkan sepasang kunci, yaitu kunci publik dan kunci privat. Masing-masing kunci privat dijaga kerahasiaannya oleh masing-masing pemakai sehingga tidak ada seorang pun yang mengetahuinya.

2. Masing-masing pemakai mempublikasikan kunci publik miliknya pada suatu media publik atau *file* yang dapat dilihat oleh semua pemakai yang ada di dalam jaringan.

3. Pemakai *A* mengenkripsi pesan *M* dengan menggunakan prosedur enkripsi *E* dan kunci publik *B*, dan mengirim hasilnya berupa *chipertext C = E* [*PKb, M*] kepada pemakai *B*

4. Pemakai *B* kemudian dengan menggunakan kunci privat *B* dan prosedur dekripsi *D*, akan mendekripsi *C* untuk mendapatkan pesan: *M = D [SKb, C] = D [SKb, E [PKb, Mu]]*.

Jika dibandingkan dengan algoritma sistem konvensional, algoritma kunci publik relatif lebih kuat dan aman karena menggunakan dua buah kunci. Beberapa sistem kriptografi yang menerapkan algoritma kriptografi kunci publik antara lain : *RSA (Rivest, Sharnn & Adlernan). LUCAS Public Key Enciyption,* DSS *(Digest Signature Standard)* dan lain-lain.

Berikut ini akan dijelaskan pada tabel 2.2 tentang perbandingan antara sistem kriptografi konvensional dan kunci publik :

Tabel 2.2. Perbandinngan Metode Konvensional dengan Metode Kunci Publik

Metode Konvensional	Metode Kunci Publik
1. Algoritma yang sama dengan kunci yang sama untuk kunci enkripsi dan dekripsi 2. Pengirim dan penerima harus saling membagi algoritma dan kunci 3. Kunci harus dijaga kerahasiannya 4. Harus tidak mungkin atau minimal sangat sulit untuk mendapatkan pesan jika tidak ada informasi yang tersedia. 5. Pengetahuan tentang algonitma dan contoh chipertext harus selalu tidak cukup untuk mendapatkan kunci	1. Satu algonitma untuk enknipsi dan deknipsi dengan sepasang kunci. satu untuk enknipsi dan lainnva untuk deknipsi 2. Penginim dan penenima masing-masing harus memiliki pasangan kunci yang cocok 3. Satu dari dua kunci harus dijaga kerahasiannya 4. Harus tidak mungkin atau minimal sangat sulit untuk mendapatkan pesan jika tidak ada infonmasi yang tersedia. 5. Pengetahuan tentang algonitma dan contoh *chipertext* serta sebuah kunci harus selalu tidak cukup untuk mendapatkan kunci privat

2.7 Manajemen Kunci

Aspek penting lain yang perlu diperhatikan dalam sistem kriptografi adalah manajemen kunci. Tujuan dari manajemen kunci adalah menyediakan prosedur yang aman untuk menangani kunci kniprografi baik yang digunakan secara *symetris* maupun *asyrnetris.* Walaupun algoritma enkripsi sangat sulit secara perhitungan untuk dipecahkan. namun sistem secara keselumhan akan cepat diketahui jika kunci tidak cukup terlindungi.

2.7.1 *Secret Key* (*Private Key*)

Untuk manajemen kunci yang digunakan oleh *single user* (atau proses) dalam melindungi data, pendekatan yang paling mudah adalah mencegah penyimpanan kunci kriptografi di dalam sistem. Para pemakai bertanggung jawab untuk mengelola kuncinya dan menggunakannya pada saat proses enkripsi maupun dekripsi.

Pada sistem konvensional. para pemakai mendaftarkan kunci privat pada sistem untuk membangun saluran yang aman. Kunci-kunci ini harus dilindungi. dan strategi yang paling mudah adalah menyimpannya dalam suatu file yang telah dienkripsi di bawah kontrol *master key* sistem. Tidak seperti *password.* kunci enkripsi tidak dapat dilindungi dengan sistem *one-way-functions* karena akan menjadi sangat sulit untuk mendapatkan kembali kunci-kunci tersebut.

Pada sistem kunci publik. seorang pemakai tidak perlu mendaftarkan kunci privat pada sistem untuk membangun saluran yang aman. Hal ini dikarenakan sifat kunci privat yang tidak boleh diketahui oleh orang lain selain pemakai yang bersangkutan. Kunci privat ini harus selalu dijaga kerahasiannva dan pihak yang tidak berhak. untuk mencegah terjadinva pemalsuan.

2.7.2 Kunci Publik (*Public Key)*

Kunci publik dapat didistribusikan baik ke luar dari sistem maupun di dalam sistem yaitu pada suatu *public key directoiy. Public key directoiy* dikelola oleh *system directory manager S.* Kunci publik tidak perlu dijaga keamanannva, karena kerahasiaan kunci publik tidak diperlukan untuk *secure communication*

atau *digital signatures.* Tetapi sangat penting untuk menjaga integritas dari kunci publik tersebut sehingga dapat digunakan untuk pengiriman pesan yang memenuhi kondisi kerahasiaan *(secrecy).* dan kevalidan *digital signatures.* Jika *S* menyediakan kunci publik yang *valid* tetapi tidak benar. seorang pemakai *A* dapat tanpa diketahui menngenkripsi data rahasia dan mengirimkannya pada pemakai *B.* Tetapi ternvata bukan *B* yang menerima pesan tersebut melainkan pemakai yang sehamsnya tidak boleh mengetahuinva. Hal mi bisa saja terjadi jika kunci publik pemakai *B* diganti oleh pihak yang tidak bertanggung jawab menjadi kunci publik pemakai yang lain. Untuk menangani masalah ini, Kohnfekder mengusulkan penggunaan tanda pengesahan *(certificates).*

Pada saat pendaftaran kunci publik oleh seorang pemakai pada suatu sistem, pemakai tersebut akan menerima *certifIcates* yang telah ditandai oleh *S.* sehingga seorang pemakai yang akan mengirimkan pesan dapat mengetahui dan yakin bahwa kunci publik yang digunakan adalah benar dan ditandai dengan *certificate* dari *S.*