BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian mengenai steganografi telah banyak dilakukan, terutama steganografi pada data citra. Bandyopadhyay dan Chakraborty melakukan penelitiannya pada tahun 2011, penelitian ini memaparkan tentang metode steganografi pada citra menggunakan empat algoritma yang disesuaikan dalam urutan DNA. Pada tahun sebelumnya Bandyopadhyay et al (2010) juga telah memperkenalkan sebuah metode baru steganografi berdasarkan algoritma genetik dalam penelitiannya. Penggunaan steganografi dan kriptografi secara bersamaan juga pernah dilakukan, diantaranya pada tahun 2010, Narayana dan Prasad memperkenalkan sebuah pendekatan baru untuk keamanan steganografi pada citra menggunakan teknik kriptografi dan konversi, metode ini menunjukkan bagaimana mengamankan citra dengan mengubahnya menjadi *cipher image* menggunakan algoritma kunci rahasia S-DES dan menyembunyikan citra tersebut dalam citra lain menggunakan metode steganografi. Metode yang diusulkan ini juga mencegah kemungkinan steganalysis.

Steganografi pada media lain seperti suara digital juga pernah dilakukan, Geetha dan Muthu dalam penelitiannya (2010) tentang audio steganografi mengatakan bahwa menyisipkan pesan rahasia ke dalam suara digital adalah sangat sulit dilakukan dibandingkan citra digital (Bandyopadhyay et al, 2008), hal ini dikarenakan tidak adanya *byte* tambahan yang dihasilkan untuk penyisipan

(Geetha and Muthu, 2010). Pada audio steganography, pesan rahasia disisipkan dengan sedikit mengubah urutan biner dari berkas suara. Pada penelitian yang dilakukan oleh Atoum et al (2011), mengusulkan sebuah metode baru penyisipan informasi ke dalam berkas audio (MP3) di antara frame (BF) dalam berkas MP3.

Sebagian besar penelitian steganografi menggunakan media penutup seperti gambar, klip video dan suara. Namun, steganografi pada teks biasanya tidak disukai karena kesulitan dalam menemukan *redundant bits* pada dokumen teks (Singh et al, 2009). Untuk menyisipkan informasi ke dalam dokumen, karakteristiknya harus diubah terlebih dahulu. Karakteristik ini dapat berupa format teks atau karakteristik dari karakter. Tapi masalahnya adalah bahwa jika perubahan kecil telah dilakukan pada dokumen, maka akan menjadi mudah terlihat oleh pihak ketiga atau penyerang. Beberapa metode diusulkan untuk memecahkan masalah tersebut, diantaranya dengan pergeseran garis, pergeseran kata, sampai dengan manipulasi *whitespace* pada teks penutup (*cover text*) (Carro, 2007).

Banerjee et al (2011) melakukan penelitiannya tentang steganografi teks, pada makalahnya, Banerjee memperkenalkan suatu metode penyembunyian pesan pada teks dengan mengubah awalan "a" atau "an" pada *cover text* berbahasa Inggris. Sebuah pendekatan lainnya diusulkan dalam penyembunyian informasi menggunakan spasi antar-kata dan spasi antar-paragraf sebagai metode hibrida, Por et al (2008) menyebutnya "*Whitesteg*". Skema yang digunakan pada *Whitesteg* adalah dengan mengkonversi pesan rahasia ke dalam bilangan biner lalu kemudian tiap bit pesan disisipkan ke setiap *whitespace* dalam *cover text*

dengan mengubahnya terlebih dahulu ke dalam karakter *whitespace* yang lain, satu spasi untuk nilai "0" dan dua spasi untuk nilai "1". Kekurangan dari metode ini adalah terbatasnya jumlah bit pesan yang akan disisipkan tergantung jumlah bit dari teks yang digunakan sebagai penutup, serta belum di terapkannya sistem enkripsi untuk mengamankan pesan yang disisipkan, sehingga pesan akan mudah dipecahkan jika *stego text* telah dicurigai oleh pihak ketiga mengandung pesan rahasia.

Sebuah penelitian memberikan cakrawala baru untuk komunikasi yang aman melalui XML di Internet (Memon et al, 2008). XML memungkinkan pengiriman pesan tidak dapat diubah jika dicegat oleh pihak ketiga. Skema yang digunakan adalah dengan menyisipkan pesan diantara tag-tag XML. Namun kekurangan dari skema ini adalah belum dimanfaatkannya teknik kriptografi untuk meng-enkripsi pesan sehingga jika penyusup mengetahui teknik-teknik steganografi tersebut maka pesan yang disembunyikan dapat terpecahkan. Berbeda dengan makalah Mir dan Hussain (2010), yang membahas steganografi teks melalui XML dengan pesan rahasia yang dienkripsi terlebih dahulu menggunakan algoritma kriptografi AES (*Advanced Encryption Standard*).

Tabel 2.1 Tabel Pembanding Penelitian

No.	Pembanding	Penelitian sebelumnya	Usulan Metode
1.	Metode Penyisipan	Penyisipan informasi rahasia dilakukan pada *whitespace* antar kata, hal ini mengakibatkan susunan paragraf menjadi kacau dan terlihat mencurigakan, metode ini baik	Penyisipan hanya pada *End of Line*, tepatnya setelah karakter *semicolon* sehingga susunan

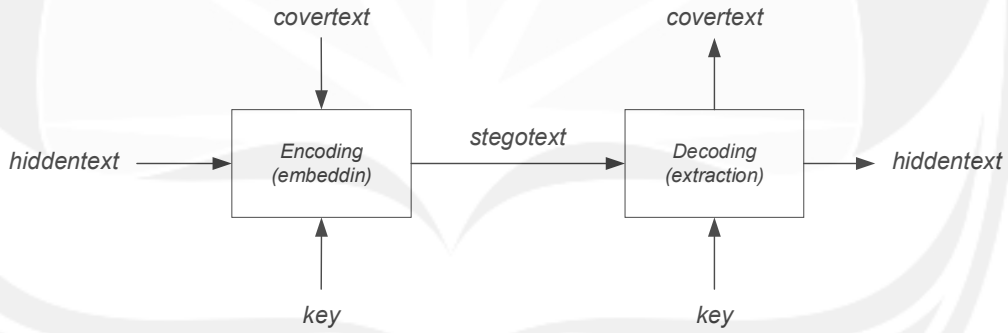
		digunakan pada teks yang lebih berorientasi pada rata tengah (Por et al, 2008).	teks tidak mengalami perubahan.
2.	Media Penutup	Media yang digunakan pada makalah yang diusulkan oleh Mir dan Hussain (2010) adalah XML.	Media yang diusulkan menggunakan CSS.
3.	Sekuritas	Informasi rahasia disisipkan ke dalam *whitespace* tanpa di enkripsi terlebih dahulu, sehingga jika stego-teks telah dicurigai akan mudah untuk dipecahkan (Por et al, 2008).	Penyisipan informasi dilakukan dengan terlebih dahulu melalui proses enkripsi.
4.	Algoritma Enkripsi	Algoritma yang digunakan pada makalah yang diusulkan oleh Mir dan Hussain (2010) adalah AES, yaitu algoritma kriptografi simetris. Masalah distribusi kunci adalah salah satu kelemahan dari algoritma kriptografi simetris.	Menggunakan algoritma kriptografi kunci publik (asimetris) yaitu RSA.

Tabel 2.1 menunjukkan beberapa penelitian mengenai steganografi teks yang telah dilakukan sebelumnya, beberapa memanfaatkan *whitespace* dalam penyisipannya dan menggunakan kriptografi untuk pengamanannya. Penelitian ini memperkenalkan sebuah pengembangan dari skema penyembunyian informasi pada teks yang disisipkan melalui berkas *Cascading Style Sheet* (CSS) dengan memanfaatkan *End Of Line (EOL)* pada tiap *property style* CSS, tepatnya setelah tanda *semicolon*. Sebelum disisipkan ke dalam *covertext*, pesan terlebih dahulu di enkripsi menggunakan sistem kriptografi kunci publik RSA.

2.2 Landasan Teori

2.2.1 Steganografi

Steganografi berasal dari bahasa Yunani *Steganós* (*Covered*) dan *Graptos* (*Writing*) (Jalab et al, 2009; Walia et al, 2010; Sultahan and kanmani, 2011). Steganografi secara teknis berarti pesan yang ditutupi atau pesan yang tersembunyi. Tujuan utama dari steganografi adalah untuk menyembunyikan data ke dalam data data lainnya sehingga tidak memungkinkan pihak ketiga untuk mendeteksi keberadaan pesan yang dimaksud. Gambar 2.1 menunjukkan skema proses steganografi yang disisipkan ke dalam media penutup (*cover*).



Gambar 2.1 Proses Steganografi

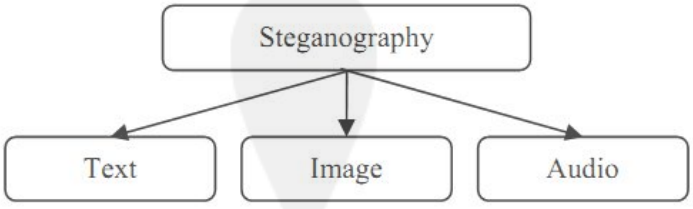
Sejarah penggunaan steganografi tercatat pada jaman Yunani kuno ketika pesan dituliskan pada kepala sang pengirim pesan yang kemudian mengirimnya setelah rambut sang pengirim pesan tumbuh kembali. Metode lain yang digunakan pada jaman yunani kuno adalah dengan mengukir pesan di dinding yang kemudian ditutupi dengan lilin untuk menyembunyikan ukiran asli tersebut. Proses ini secara efektif akan menyembunyikan pesan yang ada didalamnya,

sehingga melindungi data rahasia dari pengamat biasa. Bahkan selama Perang Dunia I dan II tinta tak terlihat digunakan untuk menulis informasi pada lembaran kertas sehingga hanya menjadi potongan-potongan kertas kosong yang tak mencurigakan (Thampi, 2004). Hingga jaman modern seperti sekarang ini, bahkan teroris pun memanfaatkan steganografi untuk berkomunikasi, beberapa sumber pemerintah bahkan menduga bahwa rekaman video Osama bin Laden yang diputar di stasiun-stasiun televisi di seluruh dunia ber-isi-kan pesan tersembunyi (Kumar and Muttoo, 2008).

Kelebihan steganografi jika dibandingkan dengan kriptografi adalah pesan-pesannya tidak menarik perhatian orang lain. Pesan-pesan berkode dalam kriptografi yang tidak disembunyikan, walaupun tidak dapat dipecahkan, akan menimbulkan kecurigaan. Seringkali, steganografi dan kriptografi digunakan secara bersamaan untuk menjamin keamanan pesan rahasianya (Por et al,2008).

2.2.2 Media Steganografi

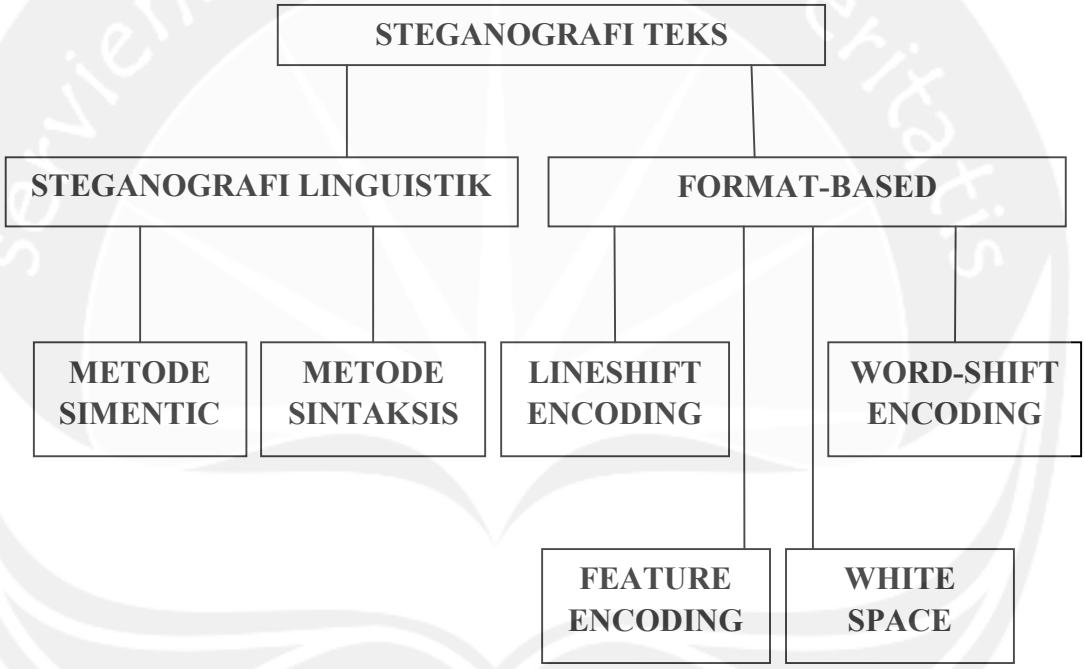
Menurut Nosrati et al (2011), tipe media yang digunakan untuk steganografi terbagi menjadi tiga kategori, teks, citra dan suara (gambar 2.2).



Gambar 2.2 Diagram tipe media steganografi (Nosrati et al, 2011)

a. Steganografi Teks

Steganografi teks dapat di bagi menjadi dua kategori, seperti terlihat pada gambar 2.3, steganografi linguistik yang kemudian dibagi kembali ke dalam metode semantik dan sintaksis. Kategori lain adalah *Format-Based Steganography* yang dibagi menjadi beberapa kategori, *line-shift encoding*, *word-shift encoding*, *open-space encoding* dan *feature encoding*.



Gambar 2.3 Tipe steganografi teks (Singh et al, 2009).

b. Steganografi Citra

Menyembunyikan informasi dalam gambar merupakan teknik yang populer saat ini. Sebuah gambar dengan pesan rahasia di dalamnya dengan mudah dapat menyebar melalui *World Wide Web* atau di *newsgroup*. Untuk menyembunyikan pesan dalam gambar tanpa mengubah sifat yang terlihat, media

penutup dapat diubah di dalam wilayah "*noisy*" dengan variasi warna yang lebih banyak, sehingga lebih sedikit perhatian pada daerah modifikasi tersebut. Metode yang paling umum digunakan pada media gambar adalah dengan *Least Significant Bits* atau LSB, *masking*, *filtering* serta transformasi pada gambar cover (Aboalsamh et al, 2008). Teknik ini dapat digunakan dengan berbagai tingkat keberhasilan yang berbeda pada berbagai jenis berkas gambar.

Metode penyisipan LSB (*least significant bit*) ini adalah menyisipi pesan dengan cara mengganti bit ke 8, 16 dan 24 pada representasi biner berkas gambar dengan representasi biner pesan rahasia yang akan disembunyikan. Dengan demikian pada setiap piksel berkas gambar 24 bit dapat disisipkan 3 bit pesan (Por et al, 2008). Misal terdapat data raster original berkas gambar adalah sebagai berikut (Das et al, 2008) :

00100111 11101001 11001000

00100111 11001000 11101001

11001000 00100111

Sedangkan representasi biner huruf A adalah 01100101, dengan menyisipkannya kedalam piksel di atas maka akan dihasilkan,

00100110 11101001 11001001

00100110 11001000 11101000

11001000 00100111

Terlihat pada bit ke-8, 16 dan 24 diganti dengan representasi biner huruf A, dan hanya tiga bit rendah yang berubah (cetak tebal), untuk penglihatan mata

manusia sangatlah mustahil untuk dapat membedakan warna pada berkas gambar yang sudah diisi pesan rahasia jika dibandingkan dengan berkas gambar asli sebelum disisipi dengan pesan rahasia.

c. Steganografi Audio

Dalam steganografi audio, pesan rahasia disisipkan ke dalam sinyal audio digital dengan cara sedikit mengubah urutan biner yang sesuai dari berkas audio (Atoum et al, 2011). Ada beberapa metode yang tersedia untuk steganografi audio, secara singkat akan dijelaskan sebagai berikut :

a. *LSB Encoding* (Begum and Venkataramani, 2011)

Teknik sampling diikuti dengan proses kuantisasi untuk mengkonversi sinyal audio analog ke dalam biner digital. Dalam teknik LSB ini, urutan biner dari masing-masing sampel berkas audio digital diganti dengan setiap biner dari pesan rahasia.

b. *Phase Encoding* (Dutta et al, 2009)

Sistem *Auditory* Manusia tidak dapat dengan mudah mengenali perubahan fasa dalam sinyal audio, metode *Phase Encoding* mengeksploitasi fakta ini. Teknik ini mengkodekan bit pesan rahasia sebagai pergeseran fase dalam spektrum fase dari sinyal digital.

c. *Spread Spectrum*

Ada dua pendekatan yang digunakan dalam teknik ini: *Direct Sequence Spread Spectrum* (DSSS) dan *Frequency Hopping Spread Spectrum* (FHSS) (Geetha and Muthu, 2010). DSSS adalah teknik

modulasi yang digunakan di bidang telekomunikasi. Seperti dengan teknologi spread spectrum lain, sinyal yang ditransmisikan membutuhkan bandwidth yang lebih dari sinyal informasi yang sedang dimodulasi. Transmisi DSSS mengalikan data yang dikirim oleh sinyal "*noise*". Sinyal *noise* ini adalah sebuah urutan *pseudorandom* yang memiliki frekuensi lebih tinggi dibandingkan dengan sinyal asli, sehingga menyebarkan energi dari sinyal asli menjadi *band* yang lebih luas (Dutta et al, 2009). Sinyal yang dihasilkan menyerupai *white noise*. Namun, sinyal yang mirip seperti *noise* ini dapat digunakan dengan tepat untuk merekonstruksi data asli di sisi penerima, dengan mengalikan urutan *pseudorandom* yang sama.

d. *Echo Hiding*

Dalam metode ini, pesan rahasia disisipkan ke dalam sinyal audio sebagai sebuah gema (*echo*). Tiga parameter sinyal gema yaitu amplitudo, *decay rate* dan *offset* dari sinyal asli yang bervariasi untuk mewakili pesan biner dikodekan secara rahasia. Tiga parameter tersebut diatur sedemikian sehingga berada di bawah ambang Sistem Pendengaran Manusia, *Human Auditory System* (HAS) sehingga gema tidak dapat dengan mudah dipecahkan (Das et al, 2008).

2.2.3 Teori Bilangan

Sebagian besar algoritma kriptografi kunci publik menggunakan teori bilangan sebagai dasarnya. Masing-masing algoritma memerlukan teori bilangan yang berbeda, namun ada beberapa pula yang sama. Pada bagian ini akan dijelaskan teori bilangan apa saja yang diperlukan dalam algoritma RSA.

2.2.3.1 Modulo

Operasi aritmatika modulo dapat ditulis dengan notasi a mod n. Modulo akan menghasilkan bilangan bulat yang merupakan sisa dari pembagian suatu bilangan. Pada a mod n akan menghasilkan kemungkinan bilangan antara 0 sampai dengan n-1 (Stallings, 1999). Contoh :

a. 23 mod 12 = 11 → 23 mod 12 = 1.12 + 11
b. 11 mod 12 =11 → 11 mod 12 = 0.12 + 11
c. 36 mod 6 = 0 → 36 mod 6 = 6.6 + 0
d. 20 mod 6 = 2 → 20 mod 6 = 6.3 + 2
e. 23 ≡ 11 mod 12 = 11
f. -18 mod 7 = 3 → -3.7 + 3
g. -43 mod 9 = 2 → -5.9 + 2

Pada contoh a dan b disebut sebagai modulo kongruen (*congruent modulo*). Modulo kongruen n jika (a mod n) = (b mod n), dapat ditulis dengan notasi a ≡ b mod n, seperti pada contoh e. untuk contoh a mod b dengan nilai negatif dapat dilihat pada contoh f dan g. Kriptografi banyak menggunakan perhitungan mod a^x mod n dengan nilai a^x yang cukup besar. Contoh a^8 mod n dapat dikerjakan dengan cara :

a. Langsung, a^8 dikerjakan terlebih dahulu (a*a*a*a*a*a*a*a) mod n atau
b. Secara bertahap, $((a^2 \bmod n)^2 \bmod n)^2$

2.2.3.2 Bilangan Prima

Bilangan prima p adalah bilangan bulat yang lebih besar dari 1 dan hanya dapat difaktorkan oleh bilangan 1 dan bilangan itu sendiri (Benedetto, 2009).

Contoh beberapa bilangan prima adalah 2, 3, 5, 7, 11, 13, 17, 19, 23, dan seterusnya. Jumlah bilangan prima yang berhasil ditemukan terus bertambah dan semakin besar nilainya. Kriptografi kunci publik khususnya RSA, menggunakan bilangan prima yang cukup besar hingga 100 digit atau lebih (Raskind and Blum, 2011).

Informasi yang dicatat meliputi peringkat bilangan prima berdasarkan besarnya bilangan, bilangan prima itu sendiri, besarnya digit, nama penemu, tahun ditemukan dan keterangan singkat algoritma yang digunakan.

2.2.3.3 *Greatest Common Divisor* dan Bilangan Prima Relatif

Greatest Common Divisor atau pembagi (faktor) persekutuan terbesar dapat ditulis dengan notasi gcd(a,b) = n. pembagi persekutuan terbesar dari 60 dan 24 adalah gcd(60,24) = 12. Karena faktor dari 60 adalah 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60 dan faktor dari 24 adalah 1, 2, 3, 4, 6, 8, 12, 24. Pada gcd(a,b) = n, a dan b disebut bilangan relatif prima jika gcd (a,n) = 1. Contoh gcd(7,19) = 1, maka bilangan 7 dan 19 adalah bilangan prima relatif, karena faktor dari 7 adalah 1, 7 dan faktor dari 19 adalah 1, 19. Untuk mencari gcd (a,b) dengan nilai a dan b yang cukup besar dapat menggunakan algoritma *Euclid*, sehingga tidak perlu mencari faktor yang sama dengan nilai terbesar dari seluruh faktor yang dimiliki oleh bilangan a dan b. Cara kerja algoritma *Euclid* menggunakan perhitungan modulo, yaitu :

$R_1 = a \bmod b$

$R_2 = b \bmod R_1$

$R_3 = R_1 \bmod R_2$

$R_i = R_{i-2} \bmod R_{i-1}$

Sampai dengan $R_i = 0$, maka gcd (a,b) adalah R_{i-1}.

Contoh : gcd(1812,1572) adalah

```
240 = 1812 mod 1572
132 = 1572 mod 240
108 = 240 mod 132
24   = 132 mod 108
12   = 108 mod 24
0  = 24 mod 12 → maka gcd(1812,1572) = 12
```

2.2.3.4 Fungsi Totient Euler (*Euler Totient Function*)

Fungsi totient euler dari n ditulis dengan φ (n), adalah bilangan bulat positif yang kurang dari n dan juga bilangan prima relatif terhadap n (Stallings, 1999). Pada pemakaian algoritma RSA, n berupa bilangan prima. Untuk sebuah bilangan prima p maka :

$$\varphi (p) = (p - 1)$$

Jika diketahui dua buah bilangan prima p dan q, dimana n =pq maka :

$$\varphi (n) = \varphi (pq) = \varphi (p) . \varphi (q) = (p-1) . (q-1)$$

2.2.4 Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani, terdiri dari dua suku kata yaitu *kripto* dan *graphia*. *Kripto* artinya menyembunyikan, sedangkan

graphia artinya tulisan. Kriptografi, secara umum adalah ilmu dan seni untuk menjaga keamanan pesan (Radha et al, 2011). Selain pengertian tersebut terdapat pula pengertian ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data (Menezes et al, 1996).

2.2.4.1 Terminologi dalam Kriptografi

Beberapa terminologi dalam kriptografi antara lain (Zou, 2010) :

a. Plainteks dan cipherteks

Pesan adalah suatu informasi atau data yang dapat dibaca dan dimengerti maknanya. Dalam kriptografi, nama lain untuk pesan adalah plainteks (*plaintext*).

Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan disandikan ke bentuk lain. Bentuk pesan yang tersandi disebut cipherteks (*ciphertext*). Cipherteks harus dapat ditransformasi kembali menjadi plainteks.

b. Pengirim dan penerima

Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan. Entitas dapat berupa orang, mesin, kartu ATM dan sebagainya.

c. Enkripsi dan dekripsi

Enkripsi (*encryption*) atau *enchipering* adalah sebuah proses penyandian yang melakukan perubahan sebuah kode (pesan) dari yang dapat dimengerti (*plainteks*) menjadi sebuah kode yang tidak bisa dimengerti (*cipherteks*).

Sedangkan proses kebalikannya yaitu mengembalikan cipherteks menjadi plainteks disebut dekripsi (*decryption*) atau *deciphering*.

d. Kunci

Kriptografi modern mengatasi masalah keamanan algoritma kriptografi dengan penggunaan kunci. Kunci (*key*) adalah parameter yang digunakan untuk proses *enchipering* ataupun sebaliknya proses *dechipering*. Kunci biasanya berupa *string* atau deretan bilangan.

e. Kriptanalisis

Kriptanalisis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan cipherteks menjadi plainteks tanpa mengetahui kunci yang diberikan. Pelakunya disebut kriptanalis *(cryptanalyst)*.

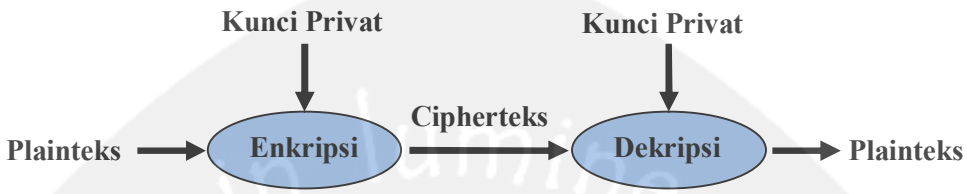
2.2.4.2 Algoritma Kriptografi

Terdapat dua macam algoritma kriptografi berdasarkan kunci yang digunakan untuk enkripsi dan dekripsi (Menezes et al, 1996), yaitu algoritma simetris (*symmetric algorithms*) dan algoritma asimetris (*asymmetric algorithms*).

a. Algoritma Simetris

Pada sistem kriptografi kunci-simetris, kunci untuk enkripsi sama dengan kunci untuk dekripsi, oleh karena itulah dinamakan kriptografi simetris (Gambar 4). Istilah lain untuk kriptografi kunci-simetris adalah kriptografi kunci privat (*private-key cryptography*), kriptografi kunci rahasia (*secret-key cryptography*), atau kriptografi konvensional (*conventional cryptography*). Sistem kriptografi kunci-simetris (atau disingkat menjadi "kriptografi simetris" saja), mengasumsikan pengirim dan penerima pesan sudah berbagi kunci yang

sama sebelum bertukar pesan. Skema sistem kriptografi kunci-simetris ditunjukkan pada gambar 2.4.



Gambar 2.4 Skema kriptografi simetris

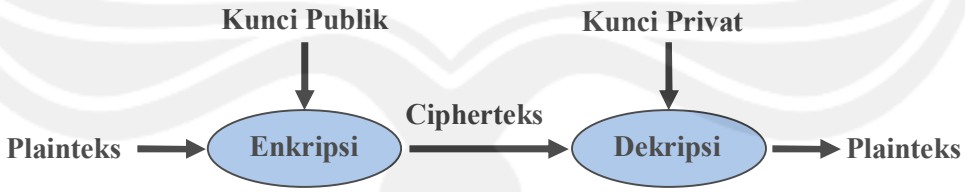
Keamanan sistem kriptografi simetris terletak pada kerahasiaan kuncinya. Kriptografi simetris merupakan satu-satunya jenis kriptografi yang dikenal dalam catatan sejarah hingga tahun 1976. Semua algoritma kriptografi klasik termasuk ke dalam sistem kriptografi simetris. Di sisi lain, ada puluhan algoritma kriptografi modern yang termasuk ke dalam sistem kriptografi simetris, diantaranya adalah *DES* (*Data Encryption Standard*), *Blowfish*, *Twofish*, *Triple-DES*, *IDEA*, *Serpent*, dan yang terbaru adalah *AES* (*Advanced Encryption Standard*).

Secara umum, *cipher* yang termasuk ke dalam kriptografi simetris beroperasi dalam mode blok (*block cipher*), yaitu setiap kali enkripsi/dekripsi dilakukan terhadap satu blok data (yang berukuran tertentu), atau beroperasi dalam mode aliran (*stream cipher*), yaitu setiap kali enkripsi/dekripsi dilakukan terhadap satu-bit atau satu *byte* data. Aplikasi kriptografi simetris yang utama adalah adalah melindungi kerahasiaan data yang dikirim melalui saluran tidak aman dan melindungi kerahasiaan data yang disimpan pada media yang tidak aman. Kelemahan dari sistem ini adalah baik pengirim maupun penerima pesan

harus memiliki kunci yang sama, sehingga pengirim pesan harus mencari cara yang aman untuk memberitahukan kunci kepada penerima pesan.

b. Algoritma Asimetris

Algoritma asimetris menggunakan dua jenis kunci, yaitu kunci publik (*public key*) dan kunci rahasia (*secret key*). Nama lainnya adalah kriptografi kunci publik (*public-key cryptography*), sebab kunci untuk enkripsi tidak rahasia dan dapat diketahui oleh siapapun (diumumkan ke publik), sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan (karena itu rahasia). Pada kriptografi jenis ini, setiap orang yang berkomunikasi mempunyai sepasang kunci, yaitu kunci privat dan kunci publik. Pengirim mengenkripsi pesan dengan menggunakan kunci publik si penerima pesan (*receiver*). Karena kunci privat hanya diketahui oleh penerima pesan, maka hanya penerima pesan yang dapat mendekripsi pesan tersebut (Gambar 2.5). Contoh algoritma kriptografi kunci publik diantaranya *RSA*, *ElGamal*, *DSA*.



Gambar 2.5 Skema kriptografi asimetris.

Keuntungan sistem ini ada dua. Pertama, tidak ada kebutuhan untuk mendistribusikan kunci privat sebagaimana pada sistem kriptografi simetris. Kunci publik dapat dikirim ke penerima melalui saluran yang sama dengan

saluran yang digunakan untuk mengirim pesan. Saluran untuk mengirim pesan umumnya tidak aman.

Kedua, jumlah kunci dapat ditekan. Untuk berkomunikasi secara rahasia dengan banyak orang tidak perlu kunci rahasia sebanyak jumlah orang tersebut, cukup membuat dua buah kunci, yaitu kunci publik bagi para koresponden untuk mengenkripsi pesan, dan kunci privat untuk mendekripsi pesan. Berbeda dengan kriptografi kunci-simetris dimana jumlah kunci yang dibuat adalah sebanyak jumlah pihak yang diajak berkorespondensi.

2.2.4.3 Algoritma RSA (Rivest, Shamir, Adleman)

RSA adalah singkatan dari huruf depan 3 orang yang menemukan algoritmanya, pada tahun 1977 di MIT yaitu Ron Rivest, Adi Shamir dan Leonard Adleman. Dari berbagai algoritma kunci asimetri yang paling populer adalah algoritma kriptografi RSA (Hamdi, 2010). Keamanan dari algoritma RSA ini terletak pada sulitnya memfaktorkan bilangan bulat komposit yang besar (Thome, 2009). Selama pemfaktoran bilangan komposit yang besar menjadi faktor-faktor prima belum ditemukan algoritma yang efektif, maka selama itu pula keamanan algoritma kriptografi RSA ini tetap terjamin keamanannya.

Penemu pertama algoritma kriptografi kunci asimetri adalah Clifford Cocks, James H. Ellis dan Malcolm Williamson (sekelompok ahli matematika yang bekerja untuk *United Kindom's Government Communication Head Quarters*, agen rahasia Inggris) pada awal tahun 1970 (Al-Vahed, 2011). Pada waktu itu temuan itu dipublikasikan dan fakta mengenai temuan tersebut tetap menjadi rahasia hingga tahun 1997.

Algoritma kriptografi kunci asimetri untuk pertama kalinya dipublikasikan pada tahun 1976 oleh Whitfield Diffie dan Martin Hellman. Dua orang tersebut merupakan ilmuwan dari Stanford University, yang membahas metode pendistribusian kunci rahasia melalui saluran komunikasi umum (publik), yang kemudian metode tersebut dikenal dengan metode pertukaran kunci Diffie-Hellman (Diffie-Hellman Key Exchange) (Al-Vahed, 2011).

Ide awal Clifford Cocks ditemukan kembali oleh sekelompok ilmuwan dari Massachussets Institute of Technology pada tahun 1977. Sekelompok orang ini adalah Ron Rivest, Adi Shamir, dan Leonard Adleman. Mereka kemudian mempublikasikan temuan mereka pada tahun 1978 dan algoritma kriptografi kunci asimetri yang mereka temukan dikenal dengan nama algoritma kriptografi RSA. RSA itu sendiri merupakan akronim dari nama keluarga mereka, Rivest, Shamir, dan Adleman. Prosedur dalam algoritma RSA antara lain (Jaseena and John, 2011) :

a. Pembangkitan Kunci

Proses pembangkitan kunci dilakukan oleh pihak penerima data atau pesan, berikut proses yang berlaku pada pembangkitan kunci algoritma RSA.

1. Pilih dua buah bilangan prima acak yang sangat besar, p dan q. Untuk mendapatkan keamanan yang maksimal, bisa dipilih dua bilangan p dan q yang hampir sama besarnya.
2. Hitung n=p*q, dimana nilai n sebagai modulus.
3. Pilih e secara acak, yaitu bilangan bulat random dengan $1<e<\varphi(n)$, gcd $(e, \varphi(n)) = 1$. $\varphi(n)$ merupakan bilangan bulat positif kurang dari n dan

relatif prima terhadap n dengan φ (n) = (p-1)(q-1). Sehingga diperoleh pasangan kunci publik (e, n).

4. Hitung nilai d, dengan ed mod φ(n) = 1 , pasangan kunci privat nya adalah (d, n).

Perhatikan bahwa d dan n juga relatif prima. Bilangan e dan n merupakan kunci publik, sedangkan d kunci privat. Dua bilangan prima p dan q tidak diperlukan lagi, namun p dan q kadang diperlukan untuk mempercepat perhitungan dekripsi.

b. Prosedur Enkripsi

Prosedur enkripsi pada Algoritma RSA adalah dengan mengubah plainteks menjadi cipherteks dengan mengikuti aturan berikut :

1. Bagi pesan menjadi beberapa kelompok m_i, dengan i=1,2,…,| m_i |=|n|-1.
2. Enkrip setiap kelompok dengan $c_i = m_i^e \bmod n$ (ingat bahwa proses enkripsi dilakukan dengan menggunakan kunci publik).
3. Gabungkan setiap c_i sehingga diperoleh cipherteks c.

c. Prosedur Dekripsi

Prosedur dekripsi merupakan kebalikan dari enkripsi, proses ini mengubah cipherteks menjadi plainteks, atau pesan asli. Prosedur dari proses dekripsi algoritma RSA adalah sebagai berikut :

1. Bagi cipherteks c ke dalam c_i, dengan i=1,2,…, | c_i |= |n|-1.
2. Dekrip setiap c_i dengan $m_i = c_i^d \bmod n$ (proses dekripsi menggunakan kunci privat).
3. gabungkan setiap m_i sehingga diperolah plainteks m.

2.2.5 Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam atau dengan kata lain CSS merupakan bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML (Keller and Nussbaumer, 2010).

Sintaks CSS terdiri dari tiga bagian, *selector, property* dan *value* (Stamey et al, 2005; Quint and Vatton, 2007). Contoh penulisan sintaks CSS adalah sebagai berikut:

```
body {
  color: #0789de;
  }
```

Bagian pertama sebelum tanda '{}' disebut *selector*, sedangkan yang diapit oleh '{}' disebut *declaration* yang terdiri dari dua unsur, yaitu *property* dan *value*. *Selector* dalam pernyataan di atas adalah h1, sedangkan color adalah *property*, dan #0789de adalah *value*. Satu *selector* dapat terdiri dari lebih dari satu *property* yang dipisahkan oleh *semicolon* (;).

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

2.2.6 Hypertext Preprocessor (PHP)

Hypertext Preprocessor merupakan bahasa pemrograman *server-side* yang didesain khusus untuk aplikasi berbasis web (Supaartagorn, 2011). PHP seringkali ditanamkan atau disisipkan ke dalam HTML. Penggunaan PHP biasanya difokuskan pada pengembangan aplikasi yang *serverside scripting.* Namun sebenarnya terdapat beberapa area utama penggunaan PHP, diantaranya *Serverside Scripting, Commandlinescripting, Desktop application* (Menggunakan PHPGTK). Penggunaan pada *serverside scripting* merupakan yang paling sering digunakan, terutama bila membutuhkan *website* atau aplikasi berbasis *web* yang dinamis. Dalam penulisan *script* nya, PHP dapat menggunakan *procedural programming* atau *object oriented programmming,* dapat juga menggunakan gabungan keduanya. Beberapa kelebihan PHP dari bahasa pemrograman web lainnya, antara lain (Sunyoto, 2007) :

1. Mudah dibuat dan dijalankan
2. Mampu berjalan pada *Web Server* dengan sistem operasi yang berbeda-beda : PHP mampu berjalan pada sistem operasi UNIX, keluarga Windows dan Macintosh.
3. PHP bisa didapatkan secara gratis.
4. Dapat berjalan pada *Web Server* yang berbeda : PHP mampu berjalan pada *Web Server* yang berbeda-beda, seperti Microsoft personal *Web Server*, Apache, IIS, Xitami, dll.
5. Dapat di-*embedded* atau dengan kata lain, PHP dapat diletakkan dalam tag HTML.

Script PHP diawali dan diakhiri dengan tag khusus. Contoh *script* yang ditulis menggunakan bahasa PHP adalah sebagai berikut :

```
<?php
        echo "Hello World";
?>
```

variabel pada PHP bersifat *case sensitive* namun tidak pada fungsi-fungsinya. Keamanan yang diberikan pada PHP dapat berupa penyaringan data yang masuk dalam sistem, atau pemilahan hak akses yang diberikan melalui tag-tag PHP.