BAB 2

LANDASAN TEORI

2.1 Kriptografi

Kriptografi merupakan salah satu ilmu pengkodean pesan memiliki definisi dan memilki teknik-tekniknya tersendiri. Hal itu dapat dilihat sebagai berikut:

2.1.1 Definisi Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani yang terdiri dari dua suku kata yaitu *kryptos* yang artinya tersembunyi dan *graphein* yang artinya tulisan. Jadi kata kriptografi dapat diartikan sebagai frase “tulisan tersembunyi”. Kriptografi merupakan studi teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, autentikasi. Teknik ini digunakan untuk mengubah data ke dalam kode-kode tertentu, dengan tujuan informasi yang disimpan atau ditransmisikan melalui jaringan yang tidak aman (misalnya internet) tidak dapat dibaca oleh siapa pun kecuali orang-orang yang berhak.

Adapun istilah-istilah yang digunakan dalam kriptografi dalam melakukan proses kerjanya adalah sebagai berikut:

a. Plaintext

   Plaintext merupakan pesan asli yang belum disandikan atau informasi yang ingin dikirimkan atau dijaga keamanannya.

b. Ciphertext

   Ciphertext merupakan pesan yang telah disandikan (dikodekan) sehingga siap untuk dikirimkan.

c. Enkripsi

   Enkripsi merupakan proses yang dilakukan untuk menyandikan plaintext menjadi ciphertext dengan tujuan pesan tersebut tidak dapat dibaca oleh pihak yang tidak berwenang.

d. Dekripsi

   Dekripsi merupakan proses yang dilakukan untuk memperoleh kembali plaintext dari ciphertext.

e. Kriptosistem

   Kriptosistem merupakan sistem yang dirancang untuk mengamankan suatu sistem informasi dengan memanfaatkan kriptografi.

Adapun beberapa tuntunan yang terkait dengan isi keamanan data yaitu:

a. Kerahasiaan (*confidentiality*)

   Kerahasiaan berarti data tersebut hanya bisa diakses oleh pihak-pihak tertentu saja.

b. Otentikasi (*authentication*)

Pada saat mengirim atau menerima informasi, kedua belah pihak perlu mengetahui bahwa pengirim dari pesan tersebut adalah orang yang sebenarnya.

c. Integritas data *(integrity)*

   Tuntutan ini berhubungan dengan jaminan setiap pesan yang dikirim pasti sampai pada penerimanya tanpa ada bagian dari pesan tersebut yang diganti, diduplikasi, dirusak, diubah urutannya dan ditambahkan.

d. Ketiadaan penyangkalan *(nonrepudiation)*

   Nonrepudiation mencegah pengirim maupun penerima mengingkari bahwa mereka telah mengirimkan atau menerima suatu pesan/informasi.

2.1.2 Macam-Macam Algoritma Kriptografi

Pada umumnya terdapat dua teknik yang digunakan dalam kriptografi yaitu:

a. Algoritma Simetri

Algoritma simetri atau disebut juga algoritma konvensional adalah algoritma yang menggunakan kunci yang sama pada proses enkripsi dan dekripsi. Algoritma ini mengharuskan pengirim dan penerima menyetujui satu kunci tertentu sebelum dapat berkomunikasi secara aman. Keamanan algoritma simetri tergantung pada rahasia kunci. Pemecahan kunci berarti memungkinkan setiap orang dapat mengenkripsi dan mendekripsi pesan dengan mudah.

Kelebihan algoritma simetri ini adalah kecepatan proses enkripsi dan dekripsinya. Sedangkan kelemahan algoritma ini adalah permasalahan distribusi kunci dan efisiensi jumlah kunci. Contoh algoritma kunci simetri

adalah OTP, DES, RC2, RC4, IDEA, Twofish, Magenta, FEAL, SAFER, LOKI, CAST, AES (Rijndael), Blowfish, GOST, A5, Kasumi, dan lain-lain.

b. Algoritma Asimetri

Algoritma asimetri adalah algoritma yang menggunakan kunci yang berbeda pada proses enkripsi dan dekripsi. Berdasarkan konsep ini, kunci yang didistribusikan adalah kunci publik yang tidak diperlukan kerahasiaannya, sedangkan kunci rahasia tetap disimpan. Jadi setiap orang yang memiliki kunci publik dapat melakukan proses enkripsi yang hanya dapat dibaca oleh orang yang memiliki kunci rahasia.

Beberapa contoh konsep yang menggunakan algoritma ini adalah skema enkripsi Elgamal, RSA, Diffie-Hellman (DH), dan DSA (Digital Signature Algorithm).

2.1.3 Keamanan Algoritma

Suatu algoritma dikatakan aman, bila tidak ada cara untuk menemukan plaintextnya. Sampai saat ini, hanya algoritma One Time Pad (OTP) yang dinyatakan tidak dapat dipecahkan meskipun diberikan sumber daya yang tidak terbatas. Namun algoritma One Time Pad juga terdapat kelemahan yaitu panjang kunci dan panjang plaintext harus sama. Hal ini tentu saja membatasi kemampuan suatu sistem untuk mengolah data rahasia yang panjang dan membuat sistem menjadi kurang efisien.

Karena selalu terdapat kemungkinan ditemukannya cara baru untuk menembus suatu algoritma kriptografi, maka algoritma kriptografi yang dikatakan ”cukup” atau ”mungkin” aman bila memiliki keadaan sebagai berikut:

1. Bila harga untuk menjebol agoritma lebih besar daripada nilai informasi yang dibuka, maka algoritma tersebut cukup aman.
2. Bila waktu yang diperlukan untuk menjebol algoritma tersebut lebih lama daripada lamanya waktu yang diperlukan oleh informasi tersebut harus tetap aman.
3. Bila jumlah data yang dienkrip dengan kunci dan algoritma yang sama lebih sedikit dari jumlah data yang diperlukan untuk menembus algoritma tersebut.

Dari hasil penelitian yang dilakukan dapat dinyatakan bahwa hanya ada dua varians AES, yaitu AES-128 dan AES-256, karena akan sangat jarang pengguna menggunakan kunci yang panjangnya 192bit, karena AES mempunyai panjang kunci paling sedikit 128 bit, maka AES tahan terhadap kriptanalisis dengan teknologi saat ini. Dengan panjang kunci 128-bit, maka terdapat sebanyak:

$2^{128} = 3{,}4 \times 10^{38}$

kemungkinan kunci. Jika digunakan komputer tercepat yang dapat mencoba 1 juta kunci setiap detik, maka akan dibutuhkan waktu $5{,}4 \times 10^{24}$ tahun untuk mencoba seluruh kemungkinan kunci. Jika digunakan komputer tercepat yang dapat mencoba 1 juta kunci setiap milidetik, maka akan dibutuhkan waktu $5{,}4 \times 10^{18}$ tahun untuk mencoba seluruh kemungkinan kunci (Rinaldi, 2006).

2.1.4 Algoritma Rijndael

Karena DES dianggap sudah tidak aman lagi, Agensi Departemen Perdagangan AS, National Institute of Standard and Technology (NIST) yang sebelum tahun 1988 juga dikenal sebagai National Bureau of Standart (NSB), mengusulkan kepada Pemerintah Federal AS untuk merancang sebuah standard kriptografi yang baru.

Untuk menghindari kontroversi mengenai standard yang baru tersebut, sebagaimana terjadi pada pembuatan DES, dimana waktu itu NSA (National Security Agency) yang berperan sebagai penilai kekuatan algoritma DES dicurigai mempunyai cara untuk mengungkap chiphertext yang dihasilkan DES tanpa mengetahui kunci, maka NIST mengadakan sayembara terbuka untuk membuat standard algoritma kriptografi yang baru sebagai pengganti DES. Standar tersebut nanti akan diberi nama Advanced Encryption Standart (AES).

Persyaratan yang diajukan oleh NIST tentang algoritma yang baru tersebut adalah:

a. Algoritma yang ditawarkan termasuk ke dalam kelompok algoritma kriptografi simetri berbasis cipher blok.
b. Seluruh rancangan algoritma harus publik (tidak dirahasiakan)
c. Panjang kunci fleksibel: 128, 192, dan 256 bit.
d. Ukuran blok yang dienkripsi adalah 128 bit.
e. Algoritma yang diimplementasikan baik software maupun hardware.

NIST menerima 15 proposal algoritma yang masuk. Konferensi umum pun diselenggarakan untuk menilai keamanan algoritma yang diusulkan. Pada Agustus 1998, NIST memilih 5 finalis yang didasarkan pada aspek keamanan algoritma:

a. Kemangkusan (*efficiency*)
b. Fleksibilitas
c. Kebutuhan memori

Finalis tersebut adalah:

a. *Rijndael* (dari Vincent Rijmen dan John Daemen – Belgia, 86 suara)
b. *Serpent* (dari Ross Anderson, Eli Biham, dan Lars Knudsen – Inggris, Israel dan Norwegia, 59 suara)
c. *Twofish* (dari tim yang diketuai oleh Bruce Schneier – USA, 31 suara)
d. *RC6* (dari Laboratorium *RSA* – USA, 23 suara)
e. *MARS* (dari IBM, 31 suara)

Pada bulan Oktober 2000, NIST mengumumkan untuk memilih Rinjdael, kemudian pada bulan November 2001, Rinjdael ditetapkan sebagai AES, dan diharapkan Rinjdael menjadi standard kriptografi yang dominan paling sedikit selama 10 tahun. Efektif pada 26 may 2002 AES telah menjadi standard dalam kriptografi kunci simetris modern.

Berdasarkan penelitian yang dilakukan oleh (Aulia Rahma Amin, Pocut Viqorunnisa, Igor Bonny Tua Panggabean, Muhammad Bahari Ilmy dari ITB) maka diperoleh tabel perbandingan algoritma di bawah ini:

Tabel 2.1 Perbandingan Algoritma

No	Jenis Algoritma	Keunggulan Rijndael
1.	a. Serpent	– Algoritma ini memiliki kecepatan yang lebih lambat.
	b. Rijndael	– Algoritma ini memiliki kecepatan yang lebih cepat. – Mampu menangani panjang blok yang berbeda.
2.	a. *Two Fish*	– Waktu proses yang dibutuhkan lebih lama. – Memiliki kode-kode yang lebih rumit.
	b. Rijndael	– Waktu proses yang dibutuhkan lebih singkat. – Memiliki kode-kode yang lebih sederhana.
3.	a.RC6	– RAM yang dibutuhkan lebih besar. – Kinerja algoritma kurang baik Tidak fleksibel.
	b. Rijndael	– RAM yang dibutuhkan lebih kecil. – Kinerja algoritma lebih baik. – Lebih fleksibel karena memiliki kemampuan bekerja yang sangat baik pada platform apapun.
4.	a. MARS	– Waktu yang dibutuhkan pada saat dekripsi dan enkripsi lebih lama.
	b. Rijndael	– Waktu yang dibutuhkan lebih singkat.

Berikut disertakan tabel perbandingan algoritma Rijndael dengan beberapa algoritma lain (Rinaldi, 2006,hal: 170).

Tabel 2.2 Perbandingan Beberapa Cipher

Cipher	**Pembuat**	**Keterngan**
DES	IBM	*Too weak to use now*
Triple Des	IBM	*Second best choice*
GOST	Uni Soviet	*Good*
RC4	Ronald Rivest	*Some keys are weak*
RC5	Ronald Rivest	*Good but pattened*
Rijndael (AES)	Daemen and Rijmen	*Best choice*
Serpent	Anderson, Biham, Knudsen	*Very strong*
Twofish	Bruce Schneier	*Very strong, widely used*
Blowfish	Bruce Schneier	*Old and slow*
IDEA	Massey dan Xuejia	*Good but patened*

Pada algoritma AES, jumlah blok input, blok output, dan *state* adalah 128 bit. Dengan besar data 128 bit, berarti *Nb* = 4 (Nb = panjang blok plaintext dibagi 32 dan Nk = panjang kunci dibagi 32) yang menunjukkan panjang data tiap baris adalah 4 byte. Dengan blok input atau blok data sebesar 128 bit, *key* yang digunakan pada algoritma AES tidak harus mempunyai besar yang sama dengan blok input. *Cipherkey* pada algoritma AES bisa menggunakan kunci dengan panjang 128 bit, 192 bit, atau 256 bit. Perbedaan panjang kunci akan mempengaruhi jumlah *round* yang akan diimplementasikan pada algoritma AES ini. Di bawah ini adalah tabel yang memperlihatkan jumlah *round* (Nr) yang harus diimplementasikan pada masing-masing panjang kunci.

Tabel 2.3 Perbandingan Jumlah Round dan Key

	Jumlah Key (Nk)	Besar Block (Nb)	Jumlah Round (Nr)
AES – 128	4	4	10
AES – 192	6	4	12
AES – 256	8	4	14

Adapun istilah dan operasi yang berkaitan dengan proses enkripsi dan dekripsi pada algoritma Rijndael adalah sebagai berikut:

a. Field GF (2^8)

Pada *finite field* GF (2^8), sebuah byte yang terdiri dari bit $b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0,$ dianggap sebagai polinomial dengan koefisien 0 dan 1:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0x^0$$

Misalnya byte dengan nilai heksadesimal (75) atau dalam biner (01110101) dapat dianggap sebagai polynomial: $x^6 + x^5 + x^4 + x^2 + 1$.

b. Penjumlahan dalam GF (2^8)

Pada representasi polinomial, hasil penjumlahan dua elemen adalah polinomial dengan koefisien-koefisiennya merupakan hasil penjumlahan modulo 2 dari koefisien-koefisien elemen-elemen tersebut. Misalnya (75) $\oplus$ (5B), diubah dalam notasi biner yaitu: (01110101) $\oplus$ (01011011). Jika bilangan biner tersebut dipresentasikan dalam notasi polinomial maka hasilnya adalah sebagai berikut:

$(x^6 + x^5 + x^4 + x^2 + 1) + (x^6 + x^4 + x^3 + x + 1) = x^5 + x^3 + x^2 + x$.

Penjumlahan ini sama dengan operasi XOR (dilambangkan dengan $\oplus$) pada level byte. Penjumlahan ini memenuhi sifat-sifat pada grup Abelian yaitu identitas, assosiatif, invers dan komutatif karena setiap elemen adalah invers penjumlahannya sendiri, maka penjumlahan dan pengurangan dalam GF (2^8) adalah sama.

c. Perkalian dalam GF (2^8)

Perkalian dalam GF (2^8) dengan representasi polinomial adalah hasil perkalian kedua polinomial dimodulo dengan sebuah polinomial berderajat 8 yang *irreducible*. Sebuah polinomial dikatakan *irreducible* jika ia hanya dapat dibagi dengan satu dan dirinya sendiri. Dalam algoritma Rijndael, polinomial *irreducible* ini adalah sebagai berikut:

$m(x) = x^8 + x^4 + x^3 + x + 1$,

Contoh: (57) • (83) = (C1),

atau:

$(x^6 + x^4 + x^2 + x + 1)\ (x^7 + x + 1) = x^{13} + x^{11} + x^9 + x^8 + x^7 + x^7 + x^5 + x^3 + x^2 + x + x^6 + x^4 + x^2 + x$

$= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$

dan $x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$ modulo $x^8 + x^4 + x^3 + x + 1$

$= x^7 + x^6 + 1$

Pengurangan oleh modulo oleh m(x) menyatakan bahwa hasil perkalian akan merupakan sebuah polinomial biner dengan derajat dibawah 8, dan dapat diwakili oleh sebuah byte.

2.1.4.1 Algoritma Enkripsi Rijndael

Langkah-langkah enkripsi untuk algoritma rijndael:

a. Mengekspansi kunci (Key Expansion)

   Pada algoritma Rijndael proses pertama yang dilalui adalah mengekspansi kunci. Kunci hasil ekspansi ini disebut dengan RoundKey yang kemudian digunakan pada tiap-tiap putaran transformasi.

b. Melakukan penjumlahan bit antara blok plaintext dengan kunci yang terekspansi.

c. Melakukan transformasi putaran sebanyak Nr kali sebagai berikut:

   1. SubByte

      Proses mensubstitusi plaintext yang telah diekspansi ke dalam S-Box.

Tabel 2.4 S-Box dalam Rijndael

Hex		**Y**															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	B	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08

	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	A1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

2. ShiftRow

   Rotasi yang dilakukan mulai baris kedua hingga baris ke-4 ke kanan.

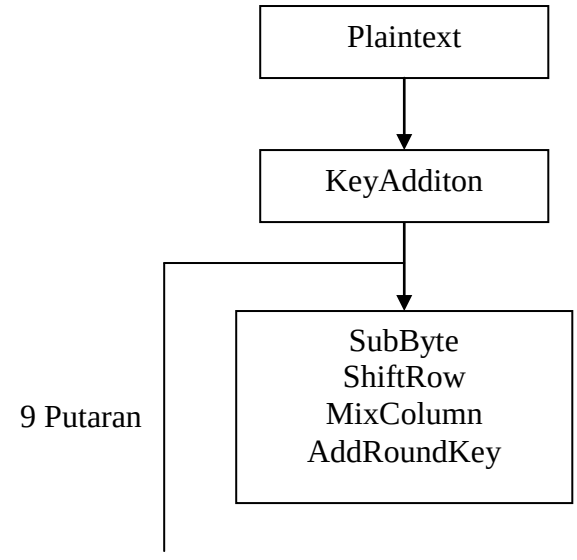
3. MixColumn (untuk putaran ke Nr langkah ini tidak dilakukan)

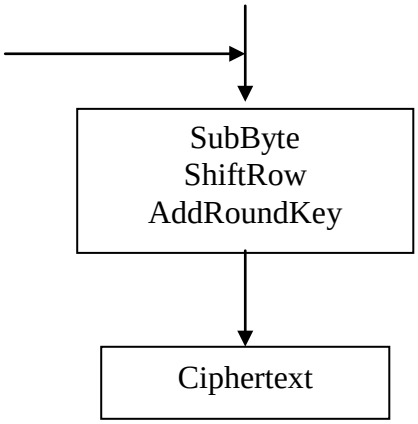
   State yang dihasilkan dari proses ShiftRow di-XOR-kan dengan matriks yang telah ditentukan. Matriks tersebut adalah:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

4. AddRoundKey

   Hasil dari MixColumn di-XOR-kan dengan RoundKey masing-masing putaran. RoundKey diperoleh pada proses ekspansi kunci.





Gambar 2.1 Alir Proses Enkripsi

Langkah-langkah dekripsi untuk algoritma Rijndael:

a. Pada proses dekripsi yang diketahui hanyalah kunci, kunci yang ada diekspansi dahulu, prosesnya sama dengan enkripsi dengan tujuan agar diperoleh RoundKey.
b. Cipher text di-XOR-kan dengan RoundKey terakhir yang diperoleh dari proses Key Schedule. Proses ini disebut Invers of AddRoundKey.
c. Ciphertext hasil yang berasal dari proses AddRoundKey digeser baris keduanya ke kanan 1 langkah, baris ketiga 2 langkah ke kanan, dan seterusnya hingga baris keempat=3 langkah ke kanan. Proses ini disebut dengan invers of ShiftRow.
d. Ciphertext yang dihasilkan dari proses Invers of ShiftRow kemudian ditransformasikan ke dalam kotak Inverse S-Box yang telah ditentukan. Proses ini dinamakan inverse of SubBytes.

e. Ciphertext yang telah ditransformasikan kemudian di XOR kan dengan matriks yang telah ditentukan. Matriks tersebut adalah sebagai berikut:

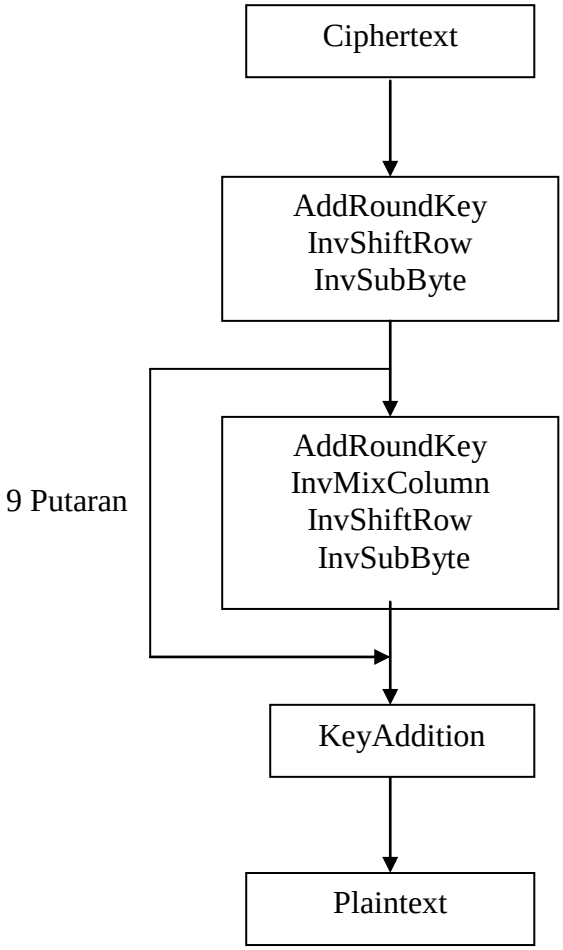
$$\begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix}$$

Pada putaran pertama dalam proses dekripsi ini proses Inverse of MixColumn ini diabaikan.

f.. Hasil dari inverse of MixColumn ini di-XOR-kan dengan RoundKey putaran selanjutnya. Begitu seterusnya hingga putaran terakhir.

Tabel 2.5 Invers S-Box Dalam Rijndael

Hex		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	E3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	17	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d



Gambar 2.2 Diagram Alir Proses Dekripsi

2.2 Steganografi

Steganografi sebagai ilmu menyembunyikan pesan memiliki definisi, *carrier file* serta jenis-jenis algoritma. Berikut disajikan definisi, *carrier file* dan algoritma yang digunakan dalam sistem keamanan data ini.

2.2.1 Definisi Steganografi

Steganografi menurut Stefanus Soehono, adalah ilmu dan seni menyembunyikan pesan rahasia sedemikian sehingga keberadaan pesan tidak dapat dideteksi oleh indera manusia. Steganografi membutuhkan wadah penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya gambar, teks dan video. Data rahasia dapat juga berupa gambar, suara, teks, atau video.

Steganografi memanfaatkan keterbatasan sistem indera manusia seperti mata dan telinga. Dengan adanya keterbatasan inilah, metode steganografi ini dapat diterapkan pada berbagai media digital. Hasi keluaran dari steganografi ini memiliki bentuk persepsi yang sama dengan bentuk aslinya, tentunya di sini sebatas oleh kemampuan indera manusia, tetapi tidak oleh komputer atau perangkat pengolah digital lainnya.

2.2.2 *Carrier File*

Carrier file adalah suatu file yang digunakan oleh file stegano untuk menyembunyikan data. Sekarang ini ada beberapa tipe dari *carrier file* yang mendukung. Tergantung dari masing-masing tipe, ada beberapa pedoman yang harus diikuti, yaitu:

a. Bitmaps

Yang paling penting dari kriteria ini adalah kedalaman warna (berapa banyak bit per pixel yang didefinisikan dari sebuah warna). Bitmap dengan mengikuti kriteria tadi maka dapat dilihat sebagai berikut:

- 4 bit = 16 warna
- 8 bit = 256 warna
- 24 bit = 16.777.216 warna

Secara umum dapat dikatakan: Semakin banyaknya warna, maka akan diperlukan keamanan yang ketat atau tinggi dikarenakan bitmap memiliki area yang sangat luas dalam sebuah warna yang seharusnya dihindarkan.
Manipulasi pada bitmap tidak dapat diubah ke dalam bentuk format grafik yang lain karena data tersembunyi dalam file tersebut akan hilang. Format menggunakan metode komperesi yang lain (seperti JPEG) tidak dapat digunakan.

b. Text files

File stegano dapat memproses text file (*.TXT) sebaik ASCII (dalam DOS), dan ANSI (dalam Windows), wbStego4 menawarkan dua perbedaan metode dalam encoding data dalam text file:

- Metode Standard: Ukuran dari file tetap tidak berubah. Ketika mengimpor manipulasi *carrier file* ke dalam *word processor* (terutama dalam Window), disana akan muncul karakter khusus dalam sebuah text.

- Metode Compatible: File semakin bertambah. Tidak akan ada kemungkinan ketika manipulasi carrier file diimport ke aplikasi lain.

c. Adobe PDF files

Tidak ada batasan dalam menggunakan file PDF. Manipulasi dari file tidak begitu jelas ketika menggambarkan file dengan Adobe Acrobat atau bentuk PDF lainnya. Penyembunyian data dalam file PDF akan meningkatkan ukuran file. Disini tidak ada aturan umum untuk jumlah dari data file PDF yang dapat diambil, tetapi jumlah mengalami pengurangan yang banyak dan penyimpanan objek yang besar.

Penyembunyian informasi rahasia ke dalam media digital mengubah kualitas media tersebut. Kriteria yang harus diperhatikan dalam penyembunyian data diantaranya adalah:

1) ***Fidelity***, maksudnya mutu citra penampung tidak jauh berubah. Setelah penambahan data rahasia, citra hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.
2) ***Recovery***, maksudnya data yang disembunyikan harus dapat diungkapkan kembali. Oleh karena tujuan steganografi adalah *data hiding*, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut.

3) ***Imperceptibility,*** maksudnya keberadaan pesan tidak dapat dipersepsi oleh indrawi. Jika pesan disisipkan ke dalam sebuah citra, citra yang telah disisipkan pesan harus tidak dapat dibedakan dengan citra asli oleh mata. Begitu pula dengan suara, telinga haruslah mendapati perbedaan antara suara asli dan suara yang telah disisipkan pesan.

2.2.3 Steganografi Gambar

Penyembunyian informasi rahasia dalam gambar merupakan implementasi dari steganografi. Dimana informasi rahasia tersebut dibuat sedemikian rupa menempel pada suatu gambar tetapi tidak tertampilkan secara kasat mata.

Steganografi gambar merupakan teknik untuk menyembunyikan suatu informasi dalam gambar digital dengan sedikit perubahan yang tidak tampak pada gambar medianya dan digunakan untuk mengirimkan informasi yang sensitif.

Metode steganografi secara umum ada tiga cara yaitu: metode LSB, masking and filtering, dan metode transformasi. Metode yang paling banyak dan mudah dilakukan adalah metode LSB karena metode ini hanya memberikan pengaruh kecil pada *cover image* karena hanya mempengaruhi satu atau dua bit pada LSB.

Selain itu berdasarkan penelitian Henry, Yus Gias Vembrina tahun 2006 beberapa keunggulan algoritma LSB adalah sebagai berikut:

1. Jika dibandingkan dengan algoritma LSB, algoritma *masking and filtering* lebih lemah terhadap kompresi, *cropping*.
2. Jika dibandingkan algoritma LSB, metode transformasi memiliki sejumlah proses yang lebih rumit dan panjang sehingga waktu proses lebih lama.

2.2.3.1 Kategori Gambar

Secara garis besar gambar dapat dibagi menjadi dua kategori, yaitu vektor dan raster. Vektor adalah serangkaian instruksi matematis yang dijabarkan dalam bentuk, garis, dan bagian-bagian lain yang saling berhubungan dalam sebuah gambar. Ukuran file relatif kecil dan jika diubah ukurannya kualitasnya tetap. File vektor sering juga dipakai dalam membuat logo, animasi, ilustrasi, dan kartun. Sedangkan raster adalah gambar yang terbentuk dari pixel yang mengandalkan jumlah pixel dalam satu satuan tertentu. Semakin rapat pixel, maka semakin baik kualitas gambar. Sebaliknya jika dipaksa diperbesar akan terlihat pecah, karena pixelnya juga bertambah besar. Format gambar bitmap sering dipakai dalam foto dan gambar. Contoh: .bmp, .jpg, .gif.

2.2.3.2 Resolusi Gambar

Resolusi adalah jumlah pixel per satuan luas yang ada pada suatu gambar. Satuan pixel yang sering dipakai adalah dpi (*dot per inch*) atau ppi (pixel per inch). Satuan dpi menentukan jumlah pixel yang ada setiap satu satuan luas. Yang dalam hal ini adalah satu inch kuadrat. Resolusi sangat berpengaruh pada detil dan perhitungan gambarnya.

2.2.3.3 Pixel (Picture Element)

Gambar yang bertipe bitmap tersusun dari pixel-pixel. Pixel disebut juga dengan dot. Pixel berbentuk bujur sangkar dengan ukuran relatif kecil yang merupakan penyusun gambar bitmap.

Banyaknya pixel tiap satuan luas tergantung pada resolusi yang digunakan. Keanekaragaman warna pixel tergantung pada *bit depth* yang dipakai. Semakin banyak jumlah pixel tiap satuan luas, semakin baik kualitas gambar yang dihasilkan dan ukuran file akan semakin besar.

2.2.3.4 Fleck

Fleck merupakan suatu tanda yang menyatakan bahwa dalam gambar menyimpan pesan atau tidak. Dalam penelitian ini, fleck dipilih dalam bentuk karakter #%*. Jika karakter ini terdapat dalam gambar, maka gambar tidak dapat disisipkan pesan. Bentuk ini dapat saja diubah, sesuai dengan keinginan pembuat program.

2.2.4 Algoritma LSB

Algoritma LSB adalah algoritma yang memodifikasi bit terakhir dalam satu byte data. Misalkan data yang diubah adalah warna hijau, maka perubahan pada LSB hanya menyebabkan sedikit perubahan yang tidak dapat dideteksi oleh manusia.

Contohnya pada file gambar, pesan dapat disembunyikan dengan cara menyisipkannya pada bit rendah (LSB) pada data pixel pada gambar tersebut terdiri dari susunan warna merah, hijau, dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Dengan demikian pada setiap pixel file bitmap 24 bit dapat disisipkan 3 bit data. Contohnya huruf A dapat disisipkan dalam 3 pixel, misalkan data raster original adalah sebagai berikut:

00100111	11101001	11001000
00100111	11001000	11101001
11001000	00100111	11101001

Sedangkan representasi huruf A adalah: **01000001**. Dengan menyisipkannya pada data pixel di atas, maka akan dihasilkan:

0010011**0**	1110100**1**	1100100**0**
0010011**0**	1100100**0**	1110100**0**
1100100**0**	0010011**0**	1110100**1**

Dengan demikian prinsip algoritma LSB di atas dapat diterapkan pada steganografi gambar. Jenis file sebagai media pembawa pesan bisa jenis file JPEG,

GIF ataupun BMP. Pada penelitian ini dipakai jenis file BMP karena pada umumnya mempunyai ukuran file yang besar sehingga informasi rahasia yang disembunyikan akan lebih banyak. Jadi semakin besar ukuran file yang ada, maka informasi rahasia yang akan disembunyikan juga akan semakin meningkat banyak.

Format file BMP 24 bit menggunakan model warna RGB. Pada model warna RGB, warna yang ditampilkan di layar monitor disusun oleh tiga buah warna primer, yaitu Red (merah), Green (hijau), Blue (biru). Satu warna sama dengan satu byte sehingga satu pixel pada RGB akan terdiri dari tiga jenis warna, maka 1 pixel akan sama dengan 24 bit atau 3 byte.