BAB II

TINJAUAN PUSTAKA

II.1 Implementasi

Implementasi dalam Kamus Bahasa Besar Indonesia diartikan sebagai pelaksanaan atau penerapan. Yang berarti bahwa hal-hal yang telah terencana sebelumnya dalam tataran ide, akan diusahakan untuk dijalankan sepenuhnya, agar hal yang dimaksud dapat tersampaikan. Dalam pengertian secara sederhana, yang dimaksud implementasi adalah juga suatu proses yang dilakukan dalam rangka evaluasi atas aspek-aspek yang dikenainya.

Dalam teori organisasi dan implementasi, Browne dan Wildavsky (dalam Nurdin dan Usman, 2004) mengemukakan bahwa implementasi adalah perluasan aktivitas yang saling menyesuaikan. Pengertian tersebut diadaptasi dari hal yang dikemukakan oleh Mc Laughin mengenai hal yang sama. Dari sumber yang sama, implementasi juga diartikan oleh Schubert, yakni implementasi adalah sebuah sistem rekayasa. (*http://www.anneahira.com/implementasi-adalah.htm,* diakses pada tanggal 03 Agustus 2015).

II.2 Kriptografi

II.2.1 Sejarah Kriptografi

Kriptografi mempunyai sejarah yang menarik dan panjang. Kriptografi sudah digunakan 4000 tahun yang lalu yang diperkenalkan oleh orang-orang Mesir untuk mengirim pesan ke pasukan militer yang berada di lapangan dan

supaya pesan tersebut tidak bisa dibaca oleh pihak musuh walaupun kurir pembawa pesan tersebut tertangkap oleh musuh (Donny ariyus, 2006).

Pada zaman Romawi kuno dikisahkan pada suatu saat, ketika Julius Caesar ingin mengirimkan satu pesan rahasia kepada seorang Jendral di medan perang. Pesan tersebut harus dikirimkan melalui seorang kurir, tetapi karena pesan tersebut mengandung rahasia, Julius Caesar tidak ingin pesan tersebut terbuka ditengah jalan. Disini Julius Caesar memikirkan bagaimana mengatasinya yaitu dengan cara mengacak pesan tersebut menjadi suatu pesan yang tidak dapat dipahami oleh siapapun kecuali oleh Jendralnya saja. Tentu sang Jendral telah diberi tahu sebelumnya bagaimana cara membaca pesan yang teracak tersebut, karena telah mengetahui kuncinya.yang dilakukan Julius Caesar adalah mengganti semua susunan alfabet dari a, b, c, & yaitu a menjadi d, bmenjadi e, c menjadi f dan seterusnya.

Dari ilustrasi tersebut, beberapa istilah *Cryptography* dipergunakan untuk menandai aktifitas-aktifitas rahasia dalam mengirim pesan. Apa yang dilakukan Julius caesar dengan cara mengacak pesannya, kita sebut sebagai *encryption* dan pada saat Sang Jendral merapikan pesan yang teracak itu, kita sebut dengan *decryption*. Pesan awal yang belum diacak dan yang telah dirapikan, kita sebut *plaintext* sedangkan pesan yang telah diacak, kita sebut *ciphertext*.

II.2.2 Pengertian Kriptografi

Kriptografi berasal dari bahasa Yunani, menurut bahasa dibagi menjadi dua *kripto* dan *graphia*, *kripto* berarti *secret* (rahasia) dan *graphia* berarti *writing*

(menulis). Menurut terminologinya kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ketempat yang lain (Donny Ariyus, 2006).

Krriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang meyediakan keamanan informasi (Rifki Sadikin, 2010).

II.2.3 Algoritma Kriptografi

Menurut Donny Ariyus (2006), Algoritma kriptografi dibagi menjadi tiga bagian berdasarkan dari kunci yang dipakainya :

1. Algoritma Simetri (menggunakan satu kunci untuk enkripsi dan dekripsinya)
2. Algoritma Asimetri (menggunakan kunci yang berbeda untuk enkripsi dan dekripsi)
3. *Hash Function*

II.2.3.1 Algoritma Simetri

Algorima ini sering disebut algoritma klasik, karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsinya. Algoritma ini sudah ada lebih dari

4000 tahun yang lalu. Mengirim pesan dengan menggunakan algoritma ini, si penerima pesan harus diberitahu kunci dari pesan tersebut agar bisa mendekripsi pesan yang dikirim. Keamanan pesan yang menggunakan algoritma ini tergantung pada kunci, jika kunci tersebut diketahui oleh orang lain maka, orang tersebut bisa melakukan enkripsi dan dekripsi pesan tersebut (Donny Ariyus, 2006). Algoritma yang memakai kunci simetri diantaranya :

1. *Data Encryption Standar* (DES)
2. RC2, RC4, RC5, RC6
3. *International Data Encryption Algorithm* (IDEA)
4. *Advanced Encryption Standar* (AES)
5. *One Time Pad* (OTP)
6. A5
7. Dan lain sebagainya.

II.2.3.2 Algoritma Asimetri

Algoritma asimetri sering disebut juga dengan algoritma kunci *public*, dengan arti kata kunci yang digunakan untuk melakukan enkripsi dan dekripsinya berbeda. Menurut Donny Ariyus (2006), pada algoritma asimetri kunci terbagi menjadi dua bagian :

1. Kunci umum (*public key*): Kunci yang boleh semua orang tahu (dipublikasikan)
2. Kunci pribadi (*private key*): Kunci yang dirahasiakan (hanya boleh diketahui oleh satu orang).

Kunci-kunci tersebut saling berhubungan satu dengan lainnya. Dengan kunci *public* orang dapat mengenkripsi pesan tapi tidak bisa mendekripsikanya, hanya orang yang memiliki kunci pribadi yang dapat mendekripsi pesan tersebut. Algoritma asimetri bisa melakukan pengiriman pesan lebih aman dari pada algoritma simetri. Contoh : Bob mengirim pesan ke Alice menggunakan algoritma asimetri, hal yang harus dilakukan adalah :

1. Bob memberitahukan kunci publiknya ke Alice
2. Alice mengenkripsi pesan dengan menggunakan kunci publik Bob
3. Bob mendekripsi pesan dari Alice dengan kunci pribadinya
4. Dan begitu juga sebaliknya jika Bob ingin mengirim pesan ke Alice

Algoritma yang memakai kunci *public* diantaranya:

1. *Digital Signature Algorithm* (DSA)
2. RSA
3. *Diffie-Hellman* (DH)
4. *Elliptic Curve Cryptography* (ECC)
5. Dan lain sebagainya.

II.2.3.3 *Hash Function* (Fungsi *Hash*)

Fungsi *hash* sering disebut dengan fungsi *hash* satu arah (*one way function*), *message digest*, *fingerprint*, fungsi kompresi dan *Message Authentication Code* (MAC), hal ini merupakan suatu fungsi matematika yang mengambil *input* panjang variabel dan mengubahnya ke dalam urutan *biner* dengan panjang yang tetap. Fungsi *hash* biasanya diperlukan bila ingin membuat

sidik jari dari suatu pesan. Sidik jari pada pesan merupakan suatu tanda yang menandakan bahwa pesan tersebut benar-benar dari orang yang diinginkan.

II.2.4 Kriptografi Klasik

Kriptografi klasik merupakan suatu algoritma yang menggunakan satu kunci untuk mengamankan data, teknik ini sudah digunakan bebrapa abad yang lalu (Donny Ariyus, 2006). Dua teknik dasar yang biasa digunakan pada algoritma jenis ini, diantaranya adalah :

1. Teknik Subtitusi: Penggantian setiap karakter plaintext dengan karakter lain
2. Teknik Transposisi (Permutasi): Teknik ini menggunakan permutasi karakter

II.2.5 Kriptografi Modern

Enkripsi modern berbeda dengan enkripsi konversional dikarenakan pada enkripsi modern sudah menggunakan komputer dalam pengoperasiannya, yang berfungsi mengamankan data baik yang ditransfer melalui jaringan komputer maupun tidak, hal ini sangat berguna untuk melindungi, *privacy*, integeritas data, *Authentication* dan *non repudiation* dibawah ini akan digambarkan bagaimana enkripsi modern saling mendukung satu dengan lainnya (Donny Ariyus, 2006).

II.3 Steganografi

II.3.1 Sejarah Steganografi

Pertama kali metode ini dipakai pada masa pemerintahan Yunani kuno dan Persia, Caesar menulis pesan dengan menggunakan papan, dan papan tersebut ditulis, kemudian dilapiskan dengan lilin, sehingga pesan tidak bisa dibaca. Pada masa itu teknik ini hanya diketahui oleh orang-orang yang mempunyai jabatan di dalam istana (Ariyus, 2006).

Papan yang berisi pesan tersebut dikirim dengan menggunakan jasa kurir. Lama-lama metode menyembunyikan pesan ini diketahui oleh pihak lawan, sehingga Caesar tidak menggunakan metode ini untuk mengirimkan pesan tetapi dengan metode *Caesar cipher*.

Steganografi juga digunakan di Cina, metode yang digunakan di Cina berbeda dengan yang digunakan Yunani dan Persia, di Cina pesan yang akan dikirim, ditulis di atas kepala kurir teknik ini disebut dengan *Histaleus* (pesan yang di*tatto* di atas kepala kurir yang telah digunduli), tapi metode ini membutuhkan waktu yang lama, karena pesan yang telah di*tatto* diatas kepala kurir bisa dikirim jika rambut sudah tumbuh. Pada zaman sekarang teknik steganografi sudah semakin moden dan tidak perlu membutuhkan waktu lama.

II.3.2 Pengertian Steganografi

Steganografi merupakan suatu cabang ilmu yang mempelajari tentang bagaimana menyembunyikan suatu informasi “rahasia” di dalam suatu informasi lainnya (Donny Ariyus, 2006).

Steganografi adalah ilmu menyembunyikan teks pada media lain yang telah ada sedemikian sehingga teks yang tersembunyi menyatu dengan media itu. Media tempat penyembunyian pesan tersembunyi dapat berupa media teks, gambar, *audio*, atau *video*. Steganografi yang kuat memiliki sifat media yang telah tertanam teks tersembunyi sulit dibedakan dengan media asli namun teks tersembunyi tetap dapat di ekstrasi (Rifki Sadikin, 2010).

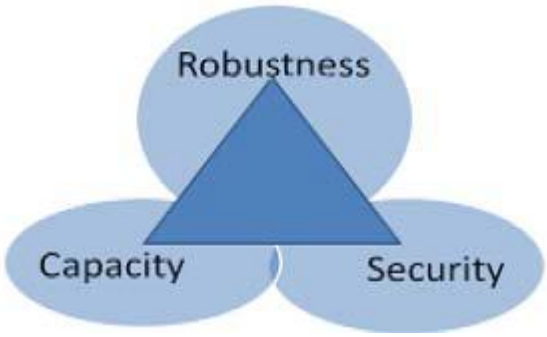
Steganografi merupakan seni penyembunyian pesan ke dalam pesan lainnya sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu di dalam pesan tersebut. Kata steganografi berasal dari bahasa Yunani yaitu *steganos* yang artinya tersembunyi atau terselubung dan *graphein*, yang artinya menulis, sehingga kurang lebih artinya adalah "menulis tulisan yang tersembunyi atau terselubung" (Tri Prasetyo Utomo, 2011).

Steganography merupakan sebuah ilmu dan seni untuk menuliskan pesan tersembunyi sedemikian rupa sehingga pihak selain yang berhak menerima tidak mengetahui keberadaan pesan tersebut. *Steganography* berbeda dengan kriptografi, dimana kriptografi keberadaan pesannya jelas tetapi maknanya dikaburkan (M. Miftakul Amin, 2013).

II.3.3 Kriteria Steganografi Yang Baik

Beberapa *property steganography* yang perlu diperhatikan yang biasa dikenal dengan segitiga *steganography*. Satu *property* dengan *property* yang lain saling bergantung, seperti dapat dilihat pada Gambar II.1. *Robustness* dapat diartikan sebagai kemampuan pesan yang disisipkan supaya tetap utuh terjaga,

jika *stego image* mengalami perubahan karena proses edit. *Security* mengacu pada proses untuk melindungi supaya penyadap/orang yang tidak berhak, tidak mampu untuk mendeteksi informasi yang tersembunyi. *Capacity* bertujuan supaya ukuran informasi yang dapat disembunyikan relatif terhadap *cover image* tidak menyebabkan kualitasnya berkurang (M. Miftakul Amin, 2013).



Gambar II.1 Segitiga *Steganography*.

(Sumber : M. Miftakul Amin, 2013:5)

II.3.4 Algoritma Steganografi

Menurut M. Miftakul Amin (2013), Secara umum algoritma *steganography* dikelompokkan berdasarkan 2 pendekatan, yaitu:

1. *Spatial domain based steganography*, merupakan algoritma yang termasuk dalam kategori ini adalah LSB (*Least Significant Bit*) sebagai pendekatan sederhana untuk menyisipkan informasi berupa *bit* ke dalam *bit* terakhir yang terdiri dari 8 *bit* (1 *byte*) yang ada dalam *cover image*. Sebagai contoh *Pixel*:

   (10101111 11101001 10101000)

   (10100111 01011000 11101001)

(11011000 10000111 01011001)

Secret message :

01000001

Result:

(10101110 11101001 10101000)

(10100110 01011000 11101000)

(11011000 10000111 01011001)

Algoritma untuk menyisipkan pesan rahasia berupa teks menggunakan LSB dapat dilakukan dengan langkah-langkah sebagai berikut:

1. Langkah 1, baca *cover image* dan pesan rahasia yang akan disisipkan dalam *cover image*.
2. Langkah 2, konversi *cover image* setiap *pixel*nya dari representasi desimal menjadi *biner*.
3. Langkah 3, konversi pesan rahasia menjadi *biner*, jika pesan berupa *file* teks maka setiap karakter dapat diketahui kode ASCII nya berupa bilangan desimal untuk kemudian dikonversi menjadi *biner*.
4. Langkah 4, hitung LSB dari setiap *pixel cover image*.
5. Langkah 5, ganti LSB dari *cover image* dengan setiap *bit* dari pesan rahasia satu per satu.
6. Langkah 6, tulis *stego image* yang merupakan hasil akhir dari proses penyisipan, biasanya dengan cara menyimpan ulang *file* gambar yang telah disisipi pesan rahasia.

Sedangkan algoritma untuk membaca pesan rahasia langkah-langkahnya sebagai berikut:

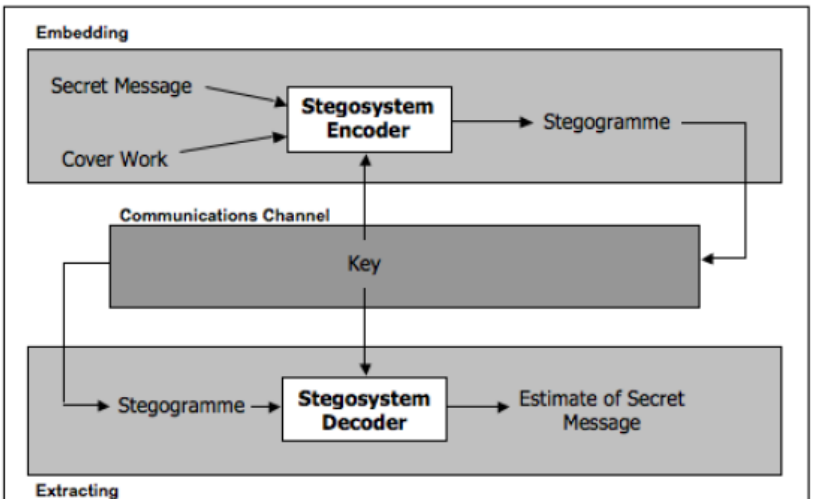
1. Langkah 1, baca *stego image*.
2. Langkah 2, hitung LSB dari setiap *pixel* dalam *stego image*.
3. Langkah 3, ambil *bit-bit* yang diperoleh pada step 2 untuk selanjutnya dikonversi menjadi karakter.

2. *Transform domain based steganography,* Algoritma jenis ini beroperasi dengan cara mentransformasikan *image* ke domain frekuensi. Contoh algoritma pendekatan transform domain diantaranya *Discrete Cosine Transform, KL Transform* dan *Wavelet Transform*.

II.3.5 Proses Steganografi

Dalam pengembangan *steganography* terdapat dua algoritma penting yaitu untuk melakukan *embedding* dan satu lagi untuk melakukan *extracting*. Proses *embedding* merupakan proses untuk menyisipkan pesan rahasia (*secret message*) ke dalam *cover work* yang berupa file *image*, *vide*o, *audio* maupun teks sebagai media untuk menyisipkan pesan. Output dari proses *embedding* disebut sebagai *Stegograme* yang berisi *cover work* dan pesan tersembunyi. Sedangkan *extracting* adalah proses untuk memunculkan kembali pesan yang tersembunyi dari *cover work.* Keseluruhan proses dalam *steganograph*y dapat dilihat pada Gambar II.2 (M. Miftakul Amin, 2013).

.



Gambar II.2 Skema Proses *Steganography*

(Sumber : M. Miftakul Amin, 2013:3)

II.4 *File* Gambar

Banyak sekali jenis-jenis gambar yang dapat kita temukan diinternet maupun komputer kita sendiri, tapi tidak semuanya kita gunakan. Beberapa gambar digunakan oleh perangkat lunak yang terpasang dikomputer kita, sehingga tidak bisa dilihat langsung tanpa menggunakan perangkat lunak khusus. Berikut ini beberapa jenis-jenis gambar yang sering digunakan.

1. JPG / JPEG (*Joint Photographic Expert Group*) *Format file* ini mampu mengkompres objek dengan tingkat kualitas sesuai dengan pilihan yang disediakan. *Format file* sering dimanfaatkan untuk menyimpan gambar yang akan digunakan untuk keperluan halaman *web*, multimedia, dan publikasi elektronik lainnya. *Format file* ini mampu menyimpan gambar dengan mode warna *RGB, CMYK*, dan *Grayscale*. *Format file* ini juga mampu menyimpan *alpha channel*, namun karena orientasinya ke

publikasi elektronik maka *format* ini berukuran relatif lebih kecil dibandingkan dengan *format file* lainnya.

2. GIF (*Graphic Interchange Format*) *Format file* ini hanya mampu menyimpan dalam 8 *bit* (hanya mendukung mode warna *Grayscale*, *Bitmap* dan *Indexed Color*). *Format file* ini merupakan format standar untuk publikasi elektronik dan *internet*. *Format file* mampu menyimpan animasi dua dimensi yang akan dipublikasikan pada *internet*, desain halaman *web* dan publikasi elektronik. *Format file* ini mampu mengkompresi dengan ukuran kecil menggunakan kompresi LZW.
3. PNG (*Portable Network Graphic*) *Format file* ini berfungsi sebagai alternatif lain dari *format file* GIF. Format *file* ini digunakan untuk menampilkan objek dalam halaman *web*. Kelebihan dari *format file* ini dibandingkan dengan GIF adalah kemampuannya menyimpan *file* dalam *bit depth* hingga 24 *bit* serta mampu menghasilkan latar belakang (*background*) yang transparan dengan pinggiran yang halus. *Format file* ini mampu menyimpan alpha channel (*http://www.zainalhakim.web.id/posting/jenis-jenis-file-gambar.html*, diakses pada tanggal 03 Agustus 2015).

II.5 *Vigenere Cipher*

II.5.1 Pengertian *Vigenere Cipher*

Vigenere cipher terdiri dari beberapa sandi *Caesar* dalam urutan dengan nilai pergeseran yang berbeda. Dalam *Caeser cipher* setiap huruf digeser sepanjang beberapa tempat. Misalnya untuk pergeseran dari 5 A akan menjadi f, b

akan memetakan ke g dan sebagainya. *Vigenere cipher* menggunakan urutan nilai pergeseran yang berbeda dan menggunakan tabel disebut *tabula recta*, *Vigenere square*, atau *Vigenere table*. Tabelnya adalah 26 * 26 *matriks* dimana abjad bahasa Inggris yang ditulis 26 kali dalam baris yang berbeda mewakili pergeseran yang berbeda pula. Tabel digunakan dan substitusi dibuat sesuai dengan pergeseran variasi nilai-nilai yang berasal dari kunci (Md. Khalid Imam Rahmani, et al, 2012).

Sandi *vigenere* merupakan sistem sandi poli-alfabetik yang sederhana. Sistem sandi poli-alfabetik mengenkripsi sekaligus sebuah teks yang terdiri dari beberapa huruf. Sandi *vigenere* menggunakan subtitusi dengan fungsi *shift* seperti pada sandi *Caesar* (Rifki Sadikin, 2010).

II.5.2 Deskripsi Aljabar *Vigenere Cipher*

Menurut Md. Khalid Imam Rahmani, et al (2012), *Vigenere cipher* juga dapat dilihat dari aljabarnya. Jika huruf A - Z yang diambil untuk menjadi nomor 0-25 kemudian enkripsi *Vigenere* E menggunakan kunci K dapat ditulis sebagai berikut:

Ci = EK (Pi) = (Pi + Ki) mod 26

dan dekripsi D menggunakan kunci K,

Pi = DK (Ci) = (Ci - Ki) mod 26

dimana

P = P0 ... Pn adalah pesan,

C = C0 ... .Cn adalah *ciphertext* dan

K = K0 ..... Kn adalah kunci yang digunakan.

II.5.3 Pembacaan Sandi dari *Vigenere Cipher*

Vigenere cipher menyembunyikan karakteristik frekuensi surat *plaintext* bahasa Inggris, tetapi beberapa pola tetap. Misalnya, jika P adalah huruf yang paling sering muncul dalam *ciphertext* yang mana *plaintext* dalam bahasa Inggris, orang mungkin menduga bahwa P sesuai dengan E, karena E adalah huruf yang paling sering digunakan dalam bahasa Inggris. Namun, menggunakan *Vigenere cipher*, E dapat dienkripsi sebagai *ciphertext* yang berbeda huruf di berbagai titik dalam pesan, sehingga mengalahkan analisis frekuensi sederhana. Kelemahan utama dari *Vigenere cipher* adalah sifat berulang dari kunci. Jika seorang *cryptanalyst* benar menebak panjang kunci, maka *ciphertext* dapat dibilang sama dengan *Caesar cipher*, yang masing-masing mudah dibobol. Pengujian *Kasiski* dan *Friedman* dapat membantu menentukan panjang kunci. Penelitian *Kasiski*, juga disebut *Kasiski Test*, mengambil keuntungan dari kemungkinan kata yang muncul berulang, dengan peluang, kadang-kadang dienkripsi menggunakan huruf kunci yang sama, yang mengarah ke berulang kelompok dalam *ciphertext*. Analisis frekuensi: Jika panjang kunci telah diketahui atau diduga, *ciphertext* ditulis ulang ke dalam banyak kolom. Setiap kolom terdiri dari *plaintext* yang telah dienkripsi oleh satu *Caesar Cipher*. Dengan menggunakan metode serupa yang digunakan untuk memecahkan sandi *Caesar*, surat-surat yang di enkripsi dapat dipecahkan. Perbaikan penelitian *Kasiski*, dikenal sebagai metode *Kerckhoffs*, mencocokkan setiap kolom frekuensi surat ke frekuensi *plaintext* yang

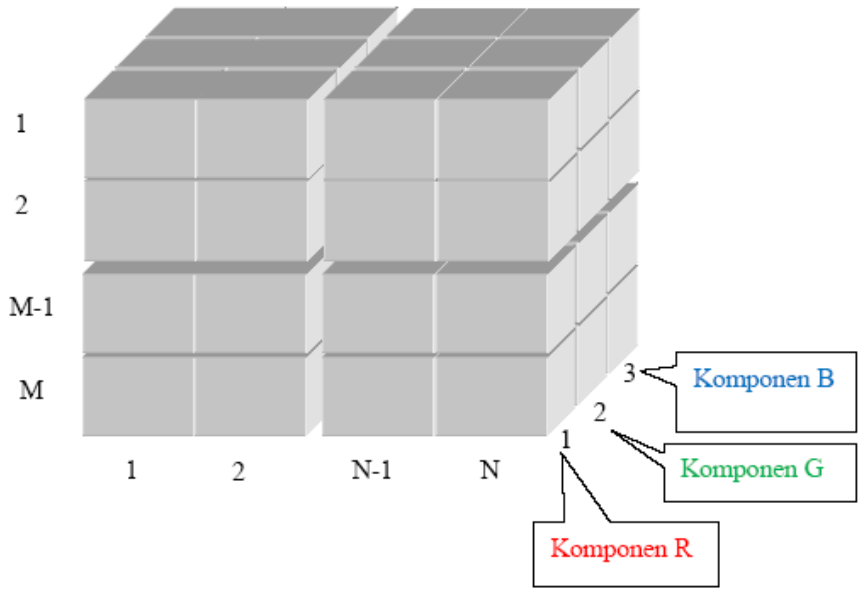
bergeser untuk menemukan huruf kunci untuk kolom itu. Setelah setiap huruf kunci diketahui, kriptanalis dengan mudah dapat mendekripsi *ciphertext* dan mengungkap *plaintext*. Metode *Kerckhoffs* tidak berlaku ketika tabel *Vigenere* telah diubah (Md. Khalid Imam Rahmani, et al, 2012).

II.6 *Least Significant Bit* (LSB)

Metode penyisipan LSB (*Least Significant Bit*) ini adalah menyisipi pesan dengan cara mengganti bit ke 8, 16 dan 24 pada representasi biner *file* gambar dengan representasi biner pesan rahasia yang akan disembunyikan. Dengan demikian pada setiap *pixel file* gambar BMP 24 *bit* dapat disisipkan 3 *bit* pesan (Tri Prasetyo Utomo, 2011).

Metode *Least Significant Bit* (LSB) adalah teknik penyembunyian pesan dengan cara menyisipkan pesan pada *bit* rendah atau *bit* paling kanan pada *file cover work* sebagai media untuk menyembunyikan pesan. Pada uji coba ini digunakan media citra digital *true colour* 24 *bit* dengan model warna RGB. Pada citra digital nantinya terdapat 3 *bit* yang dapat disisipi dalam 1 *pixel*. Hal ini dikarenakan dalam 1 *pixel* warna tersusun dari 3 komponen warna, yaitu *Red, Green,* dan *Blue* yang masing-masing disusun oleh 8 *digit* bilangan *biner* dari rentang nilai 0 sampai dengan 255 dalam desimal atau 00000000 sampai 11111111 dalam representasi *biner* (M. Miftakul Amin, 2013).

Representasi *pixel* citra digital 24 *bit* dengan model warna RGB dapat dilihat pada Gambar II.3.



Gambar II.3 Model Citra Warna RGB

(Sumber : M. Miftakul Amin, 2013:7)

II.7 *Java*

Java adalah nama salah satu bahasa pemrograman komputer yang berorientasi objek, diciptakan oleh satu tim dari perusahaan *Sun Microsystems*, perusahaan *workstation UNIX* (*Sparc*) yang cukup terkenal. *Java* diciptakan berdasarkan bahasa C++, dengan tujuan *platform independent* (dapat dijalankan pada berbagai jenis *hardware* tanpa kompilasi ulang), dengan slogan *Write Once Run Anywhere* (WORA). Dibanding bahasa C++. *Java* pada hakikatnya lebih sederhana dan memakai objek secara murni (Suarga, 2009).

II.8 *Netbeans*

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir

100 mitra (dan terus bertambah!). *Sun Microsystems* mendirikan proyek kode terbuka *NetBeans* pada bulan Juni 2000 dan terus menjadi sponsor utama. Saat ini terdapat dua produk : *NetBeans IDE* dan *NetBeans Platform*. *NetBeans IDE* adalah sebuah lingkungan pengembangan sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. *Netbeans IDE* ditulis dalam *Java* namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas *Netbeans IDE*. *Netbeans IDE* adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan. Tersedia juga *NetBeans Platform* sebuah fondasi yang modular dan dapat diperluas yang dapat digunakan sebagai perangkat lunak dasar untuk membuat aplikasi *desktop* yang besar. Mitra ISV menyediakan *plug-in* bernilai tambah yang dapat dengan mudah diintegrasikan ke dalam *Platform* dan dapat juga digunakan untuk membuat kakas dan solusi sendiri. Kedua produk adalah kode terbuka (*open source*) dan bebas (*free*) untuk penggunaan komersial dan non komersial. Kode sumber tersedia untuk guna ulang dengan lisensi *Common Development and Distribution License* (CDDL) (*https://netbeans.org/index_id.html*, diakses pada tanggal 07 Juli 2015).

.