# BAB II

# TINJAUAN PUSTAKA

## 2.1. Keamanan Data dan Informasi [2]

Keamanan dan kerahasiaan informasi data merupakan hal yang sangat penting yang terus berkembang, karena memiliki peranan yang sangat besar dalam sistem komputer, jaringan komputer, dan penggunaan teknologi komputer. Beberapa kasus menyangkut keamanan data saat ini menjadi sesuatu yang membutuhkan biaya penanganan dan pengamanan yang sedemikian besar.

Ketika kita berbicara tentang keamanan sebuah sistem jaringan, aplikasi, atau apapun yang kita lakukan adalah mengevaluasi aspek *confidentiality, integrity*, dan *availability* atau sering disebut (CIA), adapun prinsip utama dari keamanan data adalah :

1. *Confidentiality*, atau yang sering disebut kerahasiaan adalah aspek yang biasa dipahami tentang keamanan. Aspek confidentiality menyatakan bahwa data hanya dapat diakses atau dilihat oleh orang yang berhak. Biasanya aspek ini yang paling mudah dipahami oleh orang. Jika terkait dengan data pribadi, aspek ini juga dikenal dengan istilah *Privacy*.
2. *Integrity*, Aspek *integrity* mengatakan bahwa data tidak boleh berubah tanpa ijin dari yang berhak. Sebagai contoh, jika kita memiliki sebuah pesan (transfer dari rekening 12345 ke rekening 6789), maka pesan atau data transaksi tersebut tidak dapat diubah seenaknya.
3. *Availability*, Ketergantungan kepada sistem yang berbasis teknologi informasi menyebabkan sistem (beserta datanya) harus dapat diakses ketika dibutuhkan. Jika sistem tidak tersedia, not available, maka dapat terjadi masalah yang menimbulkan kerugian finansial atau bahkan nyawa. Itulah sebabnya aspek *availability* menjadi bagian dari keamanan.

## 2.2. *Cyber Crime*

### 2.2.1. Pengertian *Cyber Crime* [9,10]

*Cyber Crime* merupakan bentuk-bentuk kejahatan yang timbul karena pemanfaatan teknologi internet. Menurut kepolisian Inggris, *cyber crime* adalah segala macam penggunaan jaringan komputer untuk tujuan *criminal* atau *criminal* berteknologi tinggi dengan menyalahgunakan kemudahan teknologi digital dengan menyalahgunakan kemudahan teknologi digital [10]. Beberapa pendapat mengidentikkan *cyber crime* dengan *computer crime*. The U.S. Departement of Justice memberikan pengertian *computer crime* sebagai: "…*any illegal act requiring knowledge of computer technology for its perpetration, investigation, or prosecution*", adapun pendapat dari Andi Hamzah (1898) dalam tulisannya "Aspek-aspek Pidana di Bidang Komputer", mengartikan kejahatan komputer sebagai : "Kejahatan di bidang komputer secara umum dapat diartikan sebagai penggunaan computer secara illegal" [9].

Dari beberapa pengertian di atas, secara ringkas dapat dikatakan bahwa *cyber crime* dapat didefinisikan sebagai perbuatan melawan hukum yang dilakukan dengan menggunakan internet yang berbasis pada kecanggihan teknologi komputer dan telekomunikasi.

### 2.2.2. Karakteritik *Cyber Crime* [9]

Selama ini dalam kejahatan konvensional, dikenal adanya dua jenis kejahatan sebagai berikut:

1. **Kejahatan Kerah Biru (*Blue Collar Crime*)**

   Kejahatan ini merupakan jenis kejahatan atau tindak criminal yang dilakukan secara konvensional seperti misalnya perampokkan, pencurian dan lain-lain.

2. **Kejahatan Kerah Putih (*White Collar Crime*)**

   Kejahatan jenis ini terbagi dalam empat kelompok kejahatan, yakni kejahatan korporasi, kejahatan birokrat, malpraktek dan kejahatan individu.

*Cyber crime* sendiri sebagai kejahatan yang muncul sebagai akibat adanya komunitas dunia maya di internet, memiliki karakteristik tersendiri yang berbeda dengan kedua model di atas. Karakteristik unik dari kejahatan di dunia maya tersebut antara lain menyangkut lima hal yaitu ruang lingkup, sifat kejahatan, pelaku kejahatan, modus kejahatan dan jenis kerugian yang ditimbulkan.

### 2.2.3. Jenis-jenis *Cyber Crime* [10]

Jenis-jenis kejahatan di internet terdapat beberapa versi. Salah satu versi menyebutkan bahwa kejahatan ini terbagi dalam dua jenis, yaitu jenis pertama adalah kejahatan intelektual dengan menimbulkan kerugian dan dilakukan untuk kepuasan pribadi. Jenis kedua adalah kejahatan dengan motif politik, ekonomi atau criminal yang berpotensi menimbulkan kerugian bahkan perang informasi. Versi lain membagi *cyber crime* menjadi tiga bagian, yaitu pelanggaran akses, pencurian data dan penyebaran informasi untuk tujuan kejahatan.

Secara garis besar, ada beberapa *cyber crime* yaitu sebagai berikut :

1. *Joy Computing*, yaitu pemakaian komputer orang lain tanpa izin. Hal ini termasuk pencurian waktu operasi komputer.
2. *Hacking*, yaitu mengakses secara tidak sah atau tanpa izin dengan alat suatu terminal.
3. *The Trojan Horse*, yaitu manipulasi data atau program dengan jalan mengubah data atau instruksi pada sebuah program, menghapus menambah, menjadikan tidak terjangkau dengan tujuan untuk kepentingan pribadi atau orang lain.
4. *Data Leakage*, yaitu menyangkut bocornya data ke luar terutama mengenai data yang harus dirahasiakan. Pembocoran data komputer itu bias berupa rahasia negara, perusahaan, data yang dipercayakan kepada seseorang dan data dalam situasi tertentu.

5. *Data Diddling*, yaitu suatu perbuatan yang mengubah data valid atau sah dengan cara tidak sah, mengubah input data atau output data.
6. *To Frustate Data Communication* atau penyia-nyiaan data computer.
7. *Software Piracy*, yaitu pembajakan perangkat lunak terhadap hak cipta yang dilindungi Hak atas Kekayaan Intelektual (HaKI).

## 2.3. *Cryptography*

### 2.3.1. Pengertian *Cryptography* [2,3,11]

*Cryptography* berasal dari bahasa yunani: "*cryptos*" yang artinya "I" (rahasia) dan "graphein" yang artinya "*writing*" (tulisan). Jadi kriptografi berarti "*secret writing*" (tulisan rahasia) [2]. *Cryptography* merupakan ilmu *mengenai* teknik enkripsi dimana "naskah asli" (*plaintext*) diacak menggunakan suatu kunci enkripsi menjadi "naskah acak yang sulit dibaca" (*ciphertext*) oleh seseorang yang tidak memiliki kunci deskripsi. Dengan menggunakan kunci deskripsi, penerima dapat membaca kembali naskah asli [11]. Ilmu *cryptography* juga adalah suatu teknik untuk mengamankan data atau pesan dengan melakukan proses enkripsi dan proses deskripsi.

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti. Enkripsi dapat diartikan sebagai kode atau *cipher*. Enkripsi merupakan sebuah system dengan pengkodean menggunakan suatu tabel atau kamus yang telah didefinisikan untuk mengganti kata dari informasi yang dikirim. Sebuah *cipher* menggunakan suatu algoritma yang dapat mengkodekan semua aliran data (*stream*) bit dari sebuah pesan menjadi *cryptogram* yang tidak dimengerti (*unintelligible)*. Enkripsi dimaksudkan untuk melindungi informasi agar tidak terlihat oleh orang atau pihak yang tidak berhak.

Dekripsi adalah proses dengan algoritma yang sama untuk mengembalikkan informasi teracak menjadi bentuk aslinya. Algoritma yang digunakan harus terdiri dari susunan prosedur yang direncanakan secara hati-hati yang harus secara efektif

menghasilkan sebuah bentuk terenkripsi yang tidak bisa dikembalikan, bahkan sekalipun dengan algoritma yang sama [3].

Algoritma *cryptography* bukanlah ditentukan oleh kerumitan dalam mengelola data atau pesan yang akan disampaikan. Tetapi, algoritma tersebut memiliki 4 persyaratan sebagai berikut :

1. Kerahasiaan. Pesan (*plaintext*) hanya bisa dibaca antara dua pihak yang memiliki kewenangan.
2. Autentifikasi. Pengirim pesan harus dapat diidentifikasi dengan pasti dan penyusup harus dipastikan tidak bisa berpura-pura menjadi orang lain.
3. Integritas. Penerima pesan harus bisa memastikan bahwa pesan yang dia terima bukanlah dimodifikasi ketika saat melakukan proses transmisi data.
4. *Non-Repudiation*. Pengirim pesan harus tidak bisa menyangkal pesan yang telah dikirim.

### 2.3.2. Sejarah *Cryptography* [2]

*Cryptography* sudah digunakan lebih dari 4000 tahun yang lalu, diperkenalkan oleh orang-orang mesir lewat hieroglyph. Jenis tulisan ini bukanlah bentuk standard untuk menulis pesan. Dikisahkan pada zaman romawi kuno, pada suatu saat Julius Caesar ingin mengirimkan pesan rahasia kepada seorang jenderal di medan perang. Pesan tersebut harus di kirim melalui seorang kurir. Karena pesan tersebut mengandung rahasia, Julius Caesar tidak ingin pesan rahasia tersebut sampai terbuka di jalan. Julius Caesar kemudian memikirkan bagaimana mengatasinya. Ia kemudian mengacak pesan tersebut hingga menjadi suatu pesan yang tidak dapat dipahami oleh siapapun terkecuali oleh Jenderalnya saja. Tentu Sang Jenderal telah diberi tahu sebelumnya bagaiman cara membaca pesan teracak tersebut. Yang dilakukan Julius Caesar adalah mengganti semua susunan alfabet dari a, b, c yaitu a menjadi b, b menjadi c dan c menjadi d dan seterusnya hingga kalimat tersebut tidak bisa dibaca siapapun.

Dari ilustrasi tersebut, beberapa istilah kriptografi dipergunakan untuk menandai aktivitas-aktivitas rahasia dalam mengirim pesan. Apa yang dilakukan Julius Caesar yang mengacak pesan, disebut sebagai enkripsi. Pada saat Sang Jendral merapikan pesan yang teracak itu, proses itu disebut dekripsi. Pesan awal yang belum di acak dan pesan yang telah dirapikan disebut *plaintext*, sedangkan pesan yang telah di acak disebut *ciphertext*.

### 2.3.3. Jenis-jenis *Cryptography*

Algoritma *cryptography* dibagi menjadi tiga bagian berdasarkan 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* (kunci satu arah).

#### 2.3.3.1. Algoritma Simetri [2,4,9,12,13,14,16,15]

Algoritma ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsi. Algoritma ini sudah ada sejak 4000 tahun yang lalu. Bila mengirim pesan dengan menggunakan algoritma ini, si penerima pesan harus diberitahu kunci dari pesan tersebut agar bias mendeskripsikan pesan yang dikirim. Keamanan dari pesan yang menggunakan algoritma ini tergantung pada kunci. Jika kunci tersebut diketahui oleh orang lain, maka orang tersebut akan dapat melakukan enkripsi dan dekripsi terhadap pesan. Algoritma yang memakai kunci simetri diantaranya adalah [2]:

1. Data *Encryption Standard* (DES)

   Algoritma DES termasuk kedalam algoritma simetris karena menggunakan satu buah kunci yang sama dalam mengenkripsi dan mendekripsi data. Algoritma ini tergolong jenis algoritma *block cipher*.

Algoritma DES beroperasi pada ukuran blok data sebesar 64 bit. *Plaintext* sebesar 64 bit di enkripsi menjadi *chipertext* sebesar 64 bit dengan menggunakan 56 bit kunci internal yang dibangkitkan dari kunci eksternal sebesar 64 bit. Panjang kunci eksternal 64 bit (sesuai ukuran blok), tetapi hanya 56 bit yang dipakai yaitu 8 bit paritas tidak digunakan. Setiap blok plainteks atau *ciphertext* dienkripsi dalam 16 putaran. Setiap putaran menggunakan kunci internal berbeda [12].

2. International Data *Encryption Algorithm* (IDEA)

   Algoritma IDEA merupakan suatu algoritma simetris yang dapat bekerja untuk sebuah blok pesan dengan lebar pesan 64 bit dan panjang kuncinya berukuran 128 bit. Algoritma IDEA dapat mengenkripsi dan mendekripsi kan sebuah DOCX menjadi lebih aman dan lebih bersifat rahasia dikarenakan ukuran dari algoritma IDEA sepanjang 64 bit dan dengan menggunakan tiga bentuk operasi aljabar yang berberbeda, yaitu Operasi Perkalian Modulo (216 +1) , Operasi Penjumlahan Modulo (216) dan Operasi XOR. Operasi yang terdapat pada Algoritma IDEA dilakukan pada sebuah subblok 16 bit dan algoritma IDEA bisa melakukan sebanyak 8 iterasi. Algoritma IDEA dapat mengenkripsi pesan terbuka yang dapat berupa citra, audio, ataupun dokumen dengan menggunakan penyandian simetris dimana *key* lebih panjang dibanding pesan[4]. Pada IDEA terdapat arsitektur dasar yang terdiri dari putaran (*round)* IDEA, Penjadwalan Kunci (*key)* dan unit Kontrol [13].

3. *One Time Pad* (OTP)

   Algoritma OTP adalah algoritma berjenis *symmetric key*, artinya kunci yang digunakan untuk melakukan enkripsi dan dekripsi merupakan kunci yang sama. Dalam proses enkripsi, algoritma OTP menggunakan cara *stream cipher* dimana *cipher* berasal dari hasil XOR antara *bit plaintext* dan *bit key*. Prinsip enkripsi pada algoritma OTP adalah dengan mengkombinasikan setiap karakter *plaintext* dengan satu

karakter *key*. Karena itu, panjang *key* harus sama dengan panjang *plaintext*. Secara teoritis, tidak mungkin untuk mendekripsi *ciphertext* tanpa kuncinya karena bila *key* yang digunakan adalah *key* yang salah maka yang diperoleh bukan *plaintext* yang seharusnya. Setiap *key* hanya boleh digunakan untuk sekali pesan, pengambilan dilakukan secara acak agar tidak dapat diterka, dan jumlah karakter *key* harus sebanyak jumlah karakter pesan. Algoritma OTP sering digunakan dalam enkripsi karena prosesnya yang relatif mudah. Fungsi untuk mengenkripsi hanya dengan cara meng-XOR-kan *plaintext* dengan *key* yang telah disiapkan untuk menghasilkan *ciphertext*, yaitu c = p XOR k. Sedangkan fungsi untuk mendekripsi hanya meng-XOR-kan *ciphertext* dengan *key* yang disepakati, yaitu p = c XOR k [14].

4. *Advanced Encryption Standard* (AES)

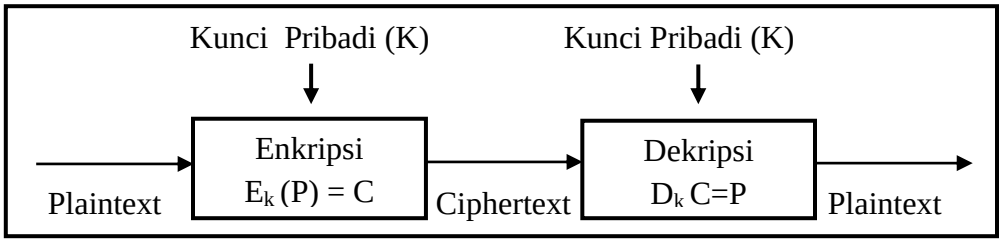
   Algoritma AES adalah algoritma berjenis *symmetric key* AES secara garis besar beroperasi pada blok 128-bit dengan kunci 128-bit sebagai berikut, (1) AddRoundKey yaitu melakukan operasi XOR antara state awal (plaintext) dengan cipher key. Tahap ini juga disebut sebagai initial round, (2) putaran sebanyak Nr – 1 kali. Proses yang dilakukan setiap putaran adalah SubBytes, ShiftRows, MixColumns dan AddRoundKey, (3) final round atau proses untuk putaran terakhir yaitu SubBytes, ShiftRows dan AddRoundKey. Berdasarkan ukuran blok yang tetap, AES bekerja pada matriks berukuran 4 x 4 dimana setiap selnya terdiri atas 1 byte (8 bit). Setiap plaintext akan diubah terlebih dahulu ke dalam blok-blok tersebut dengan bentuk heksadesimal. Setelah proses pengubahan selesai, blok akan diproses dengan metode yang dijelaskan [15].

5. Hill Cipher

   Algoritma ini adalah algoritma berjenis *symmetric key* .Algoritma ini juga dikategorikan sebagain block cipher, karena setiap karakter pada satu blok akan mempengaharui hasil karakter lainnya dalam proses enkripsi dan proses dekripsi. Proses enkripsi pada *Hill Cipher* dilakukan per blok *plaintext*. Ukuran blok tersebut sama dengan ukuran matriks kunci. Sebelum membagi teks menjadi deretan blok-blok, *plaintext* terlebih dahulu dikonversi menjadi angka, masing-masing sehingga A=0, B=1, hingga Z=25 [16].

6. Caesar Cipher

   *Caesar chiper* adalah salah satu teknik enkripsi yang paling sederhana dan paling dikenal. *Caesar chiper* merupakan jenis cipher substitusi di mana setiap huruf dalam plaintext digantikan oleh sebuah huruf dengan beberapa posisi tetap di bawah alfabet. Kriptografi *caesar chiper* hanya menggunakan 26 huruf alfabet, maka pergeseran huruf yang mungkin dilakukan hanya dari 0 sampai 25 [4].




**Gambar 2.1**

Skema Algoritma Simetri [9]

Pada gambar 2.1 merupakan skema enkripsi dan dekripsi untuk algoritma simetri. Adapun proses enkripsi dan dekripsi pada *plaintext* dan *ciphertext* dilakukan dengan menggunakan kunci pribadi (K) yang sama. Sehingga kunci pribadi yang dipakai pada proses enkripsi dan dekripsi hanya satu kunci.

Adapun beberapa kelebihan algoritma simetri adalah :

1. Waktu proses dalam enkripsi dan dekripsi relatif cepat, ini disebabkan karena efisiensi yang terjadi pada pembangkit kunci.
2. Algoritma ini dapat digunakan pada sistem secara real time misalkan pada saluran telepon karena proses enkripsi dan dekripsinya cepat.

Kekurangan algoritma simetri adalah :

1. Untuk tiap pasang pengguna dibutuhkan sebuah kunci yang berbeda, sedangkan sangat sulit untuk menyimpan dan mengingat kunci yang banyak secara aman, sehingga akan menyebabkan kesulitan dalam hal mengelola kunci.
2. Perlu adanya kesepakatan bersama untuk jalur yang khusus untuk kunci, hal ini dapat menimbulkan masalah baru karena tidak mudah untuk menentukan jalur yang aman untuk kunci masalah sering juga disebut dengan key distribution problem.
3. Apabila sebuah kunci sampai hilang atau dapat ditebak maka kriptosistem udah tidak aman lagi.

#### 2.3.3.2. Algoritma Asimetri [2,17,18,19,20]

Algoritma asimetri sering juga disebut dengan algoritma kunci publik, dengan arti kata kunci yang digunakan untuk melakukan enkripsi dan dekripsi berbeda. Pada algoritma asimetri kunci terbagi menjadi dua bagian, yaitu :

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

Kunci – kunci tersebut berhubungan satu sama lain. Dengan kunci *public* orang dapat mengenkripsi pesan tetapi tidak bisa mendekripsikannya. Hanya orang yang memiliki kunci rahasia yang dapat mendekripsi pesan tersebut. Algoritma asimetri bisa mengirimkan pesan dengan lebih aman daripada algoritma simetri. Contoh, Boby mengirim pesan ke Alice menggunkan algoritma asimetri. Hal yang harus dilakukan adalah :

1. Boby memberitahukan kunci publiknya ke Alice.
2. Alice mendekripsi pesan dengan menggunakan kunci publik Boby.
3. Boby mendekripsi pesan dari Alice dengan kunci rahasianya.
4. Begitu juga sebaliknya jika Boby ingin mengirim pesan ke Alice.

Algoritma yang memakai kunci public diantaranya adalah :

1. *Digital Signature Algorithm* (DSA)

   DSA merupakan algoritma kriptografi yang didesain untuk otentikasi pesan dengan menggunakan kunci publik dan *Secure Hash Algorithm*, kemanan ini terletak pada kesulitan di dalam komputasi logaritma diskret. DSA beroperasi pada panjang kunci yang bervariasi dari 512 bit sampai 1024 bit. Tiga proses utama algoritma DSA terdiri atas pembentukan kunci, pembentukan tanda tangan dan proses verifikasi [17].
2. Rivest Shamir Adleman (RSA)

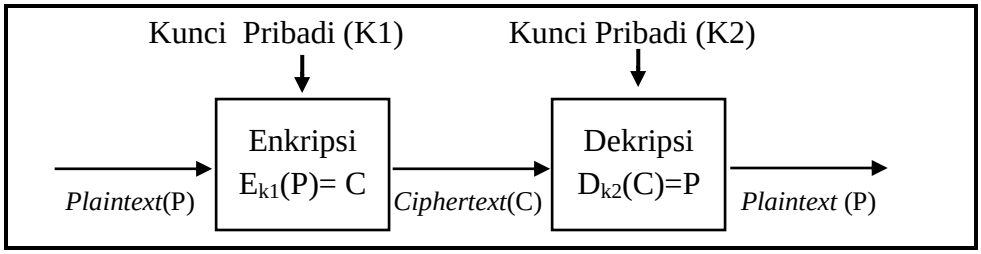
   RSA adalah algoritma untuk enkripsi kunci publik (*public-key encryption*). Algoritma enkripsi dan dekripsi sistem kriptografi RSA bersandar pada asumsi fungsi satu arah (*one-way function*) yang dibangun oleh fungsi eksponensial modular pada grup perkalian (Z*n, x) dan grup perkalian (Z*ø(n),x) dengan n = p x q. dimana p, q adalah bilangan prima dan ø(n) = (p – 1) (q – 1) [18].
3. Diffie-Hellman (DH)

   DH adalah algoritma untuk enkripsi kunci publik (*public-key encryption*). Dasar dari algoritma ini adalah matematika dasar dari aljabar eksponen

dan aritmatika modulus. Jumlah pengguna yang ingin menggunakan pertukaran kunci menggunakan algoritma Diffie-Hellman ini tidak dibatasi [19].

4. *Elliptic Curve Cryptography* (ECC)

   ECC merupakan sistem kriptografi kunci publik yang memanfaatkan persamaan kurva eliptik. Kurva eliptik mempunyai masalah logaritma yang terpisah sehingga sulit untuk dipecahkan. Algoritma kriptografi kurva eliptik mempunyai keuntungan jika dibandingan dengan algoritma kriptografi publik lainnya yaitu dalam hal ukuran panjang kunci yang lebih pendek tetapi memiliki tingkat keamanan yang sama [20].




**Gambar 2.2**

Skema Algoritma Asimetri [2]

Pada gambar 2.2 merupakan tampilan pada skema algoritma asimetri proses enkripsi dan dekripsi pada *plaintext* dan *ciphertext*. Pada proses enkripsi dan dekripsi algotitma asimetri membutuhkan kunci pribadi, kunci pribadi pada asimetri berbeda dengan kunci pribadi simetri. Perbedaan pada kunci pribadi yang digunakan pada proses asimetri adalah kunci proses enkripsi dan dekripsi bernilai berbeda atau tidak sama.

Adapun beberapa kelebihan algoritma asimetri adalah :

1. Masalah keamanan pada distribusi kunci dapat ditangani
2. Manajemen atau pengelolaan kunci pada suatu sistem informasi dengan pengguna yang banyak menjadi lebih mudah, karena jumah kunci yang digunakan lebih sedikit.

Kekurangan algoritma simetri adalah :

1. Kecepatan proses tergolong lambat bila dibandingkan dengan kriptografi simetris.
2. Dalam tingkat keamanan yang sama, rata-rata ukuran kunci harus lebih besar bila dibandingkan dengan ukuran kunci yang dipakai pada kriptografi simetris.

#### 2.3.3.3. Fungsi Hash [2]

Fungsi hash sering disebut dengan fungsi Hash satu arah (*one-way function*), *message digest, finger print,* fungsi kompresi dan *message authentication code* (MAC), merupakan suatu fungsi matematika yang mengambil masukan panjang *variable* dan mengubahnya kedalam 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 bahwa pesan tersebut benar-benar berasal dari orang yang diinginkan.

Kelebihan fungsi hash dibandingkan fungsi enkripsi pada umumnya :

1. Hasil dari fungsi hash panjangnya tetap, panjang masukan tidak akan mempengaruhi panjang nilai hash.
2. Karena tidak merubah data asli, tidak diperlukan proses dekripsi
3. Perubahan sekecil apapun pada data asli akan membuat nilai hash yang sangat jauh berbeda, sehingga cukup mudah untuk memeriksa keaslian data

Kekurangan fungsi hash diantaranya :

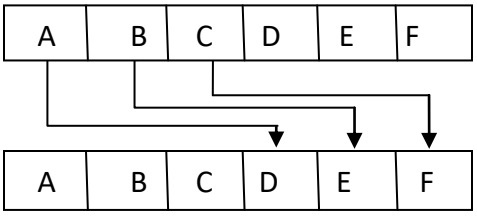
1. Memiliki kemungkinan untuk terjadi bentrokan. Hal ini tidak dapat dihindari untuk semua fungsi hash, namun ada beberapa fungsi hash dibuat khusus untuk menghindari terjadinya bentrokan.
2. Fungsi hash adalah fungsi satu arah, jadi jika kita hanya mendapat sebuah nilai hash, kita tidak bisa mengembalikannya menjadi data yang asli. Hal ini dipersulit dengan kemungkinan terjadinya bentrokan.

3. Tingkat keamanan suatu fungsi hash dinilai berdasarkan jumlah kemungkinan nilai hash, yaitu 2n , dengan n adalah panjang nilai hash dalam bit. Jadi semakin panjang nilai hash semakin aman

### 2.3.4. *Caesar Cipher* [1,2,4,16]

*Caesar cipher* adalah salah satu teknik enkripsi yang paling sederhana dan paling dikenal. *caesar cipher* adalah algoritma yang digunakan oleh sistem *cryptography* simetri dan digunakan jauh sebelum sistem *cryptography public key* ditemukan. *Caesar cipher* merupakan jenis *cipher* substitusi dimana setiap huruf dalam *plaintext* digantikan oleh sebuah huruf dengan beberapa posisi tetap dibawah alphabet, teknik ini juga dikenal sebagai *single cipher alphabet* [4].

*Caesar cipher* pertama kali digunakan oleh Julius Caesar. Caesar mengkodekan informasi dengan mengubah setiap huruf dalam informasi menjadi tiga huruf di setelah informasi asli dalam urutan alphabet. Inti dari algoritma *cryptography* ini menggeser semua karakter dalam *plaintext* dengan nilai pergeseran yang sama. Langkah yang ditempuh untuk membangun *ciphertext* dengan *Caesar cipher* adalah menentukan besarnya pergeseran karakter serta menukar karakter berdasarkan pergeseran yang telah ditentukan.




**Gambar 2.3.**

Pergeseran dalam *Caesar cipher* [2]

Pada gambar 2.3 merupakan tampilan pada pergeseran yang dalam algoritma *caesar cipher* dengan nilai pergeseran bernilai tiga, sehingga tata letak pada alphabet akan bergerak berdasarkan jumlah nilai pergeseran.

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |

**Gambar 2.4**

Pergeseran alphabet dalam *Caesar cipher* [1]

Pada gambar 2.4 merupakan contoh pergeseran karakter alphabet dengan metode yang digunakan dalam *Caesar cipher* yang dicontohkan pada pergeseran bernilai 3. Pergeseran ini adalah dengan mempertukarkan setiap huruf dari *plaintext* dengan huruf lain dengan interval 3 huruf dari huruf *plaintext*.

Untuk menyandikan sebuah pesan, cukup mencari setiap huruf yang hendak disandikan pada alfabet biasa, lalu tuliskan huruf yang sesuai pada alphabet sandi. Untuk memecahkan sandi tersebut gunakan cara sebaliknya.

Contoh penyandian sebuah pesan adalah sebagai berikut :

- Teks terang : POLITEKNIK NEGERI SRIWIJAYA
- Teks tersandi : SROLWHNQLN QHJHUL VUZLMDBD

Proses Penyandian (enkripsi) dapat secara matematis menggunakan operasi modulus dengan merubah huruf-huruf menjadi angka, A=1, B=2,…, Z=26. Secara matematis dituliskan dengan persamaan,

$$C = E(P) = (P+K) \bmod (26) \quad \ldots\ldots 1)$$

Sedangkan proses pemecahan kode (deskripsi), mengacu kepada persamaan:

$$P = D(C) = (C-K) \bmod (26) \quad \ldots\ldots 2)$$

Setiap huruf yang sama digantikan oleh huruf yang sama disepanjang pesan, sehingga sandi *Caesar* digolongkan kepada substitusi *monoalfabetik* atau *cipher* abjad tunggal yang berlawanan dengan substitusi *polialfabetik* yang berarti setiap karakter alfabet bisa dipetakan ke lebih dari satu macam [16].

### 2.3.5. Jenis-Jenis Serangan *Cryptography* [21]

Walaupun telah dilakukannya pengaman data dengan teknik *crypotography*, namun masih terdapat beberapa kemungkinan serangan yang dapat dilakukan terhadap pesan yang sudah di enkripsi berdasarkan ketersediaan data yang ada dan tingkat kesulitannya bagi penyerang, adalah :

1. *Ciphertext only attack*, penyerang hanya mendapatkan *ciphertext* dari sejumlah pesan yang seluruhnya telah dienkripsi menggunakan algoritma yang sama. Sehingga, metode yang digunakan untuk memecahkannya adalah *exhaustive key search*, yaitu mencoba semua kemungkinan yang ada untuk menemukan kunci.
2. *Known plaintext attack*, dimana penyerang selain mendapatkan sandi, juga mendapatkan pesan asli. Terkadang disebut pula *clear-text attack.*
3. *Choosen plaintext attack*, sama dengan *known plaintext attack*, namun penyerang bahkan dapat memilih penggalan mana dari pesan asli yang akan disandikan. Serangan jenis ini lebih hebat daripada *known-plaintext attack*, karena kriptoanalis dapat memilih *plaintext* tertentu untuk dienkripsikan, yaitu *plaintext-plaintext* yang lebih mengarahkan penemuan kunci.
4. *Chosen-ciphertext attack*. Pada tipe ini, kriptoanalis dapat memilih *ciphertext* yang berbeda untuk didekripsi dan memiliki akses atas *plaintext* yang didekripsi.
5. *Chosen-key attack*. Kriptoanalis pada tipe penyerangan ini memiliki pengetahuan tentang hubungan antara kunci-kunci yang berbeda dan memilih kunci yang tepat untuk mendekripsi pesan.
6. *Rubber-hose cryptanalysis*. Pada tipe penyerangan ini, kriptoanalis mengancam, menyiksa, memeras, memaksa, atau bahkan menyogok seseorang hingga mereka memberikan kuncinya. Ini adalah cara yang paling ampuh untuk mendapatkan kunci.

7. *Adaptive – chosen – plaintext attack*. Penyerangan tipe ini merupakan suatu kasus khusus *chosen-plaintext attack*. Kriptoanalis tidak hanya dapat memilih *plaintext* yang dienkripsi, ia pun memiliki kemampuan untuk memodifikasi pilihan berdasarkan hasil enkripsi sebelumnya. Dalam *chosen-plaintext attack*, kriptoanalis mungkin hanya dapat memiliki *plaintekxt* dalam suatu blok besar untuk dienkripsi dalam *adaptive-chosen-plaintext attack* ini ia dapat memilih blok *plaintext* yang lebih kecil dan kemudian memilih yang lain berdasarkan hasil yang pertama, proses ini dapat dilakukannya terus menerus hingga ia dapat memperoleh seluruh informasi.

## 2.4. *Steganography*

### 2.4.1. Pengertian *Steganography* [5,22]

*Steganography* adalah suatu teknik untuk menyembunyikan informasi yang bersifat pribadi dengan sesuatu yang hasilnya akan tampak seperti informasi normal lainnya. Media yang digunakan umumnya merupakan suatu media yang berbeda dengan media pembawa informasi rahasia, dimana fungsi dari teknik *steganography* yaitu sebagai teknik penyamaran menggunakan media lain yang berbeda sehingga informasi rahasia dalam media awal tidak terlihat secara jelas.

*Steganography* biasanya sering disalah-kaprahkan dengan kriptografi karenanya keduanya sama-sama bertujuan untuk melindungi informasi yang berharga. Perbedaan yang mendasar antara keduanya yaitu steganografi berhubungan dengan informasi tersembunyi sehingga tampak seperti tidak ada informasi tersembunyi sama sekali. Jika seseorang mengamati objek yang menyimpan informasi tersembunyi tersebut, ia tidak akan menyangka bahwa terdapat pesan rahasia dalam objek tersebut, dan karenanya ia tidak akan berusaha memecahkan informasi (dekripsi) dari objek tersebut [5].

Kata *steganography* berasal dari bahasa Yunani, yaitu dari kata *Stegan*os (tersembunyi) dan *Graptos* (tulisan). *Steganography* di dunia modern biasanya mengacu pada informasi atau suatu arsip yang telah disembunyikan ke dalam suatu arsip citra digital, audio, atau video. Teknik *Steganography* ini telah banyak digunakan dalam strategi peperangan dan pengiriman sandi rahasia sejak zaman dahulu kala. Dalam perang dunia II, teknik *steganography* umum digunakan oleh tentara Jerman dalam mengirimkan pesan rahasia dari atau menuju Jerman**.** Semakin pentingnya nilai dari sebuah informasi, maka semakin berkembang pula metode-metode yang dapat digunakan untuk melakukan penyisipan informasi yang didukung pula dengan semakin berkembangnya media elektronik. Berbagai macam media elektronik kini telah dapat digunakan untuk melakukan berbagai fungsi *steganography* dengan berbagai macam tujuan dan fungsi yang diharapkan.

Berdasarkan definisi *steganography*, maka dapat diambil beberapa *point* atau unsur yang berkaitan dengan metode atau cara menyembunyikan pesan, yaitu [22]:

1. Pesan yang dirahasiakan atau disembunyikan (*Hidden text* atau *embedded message*)
2. Media yang digunakan untuk menyembunyikan pesan (*Cover text* atau *Cover-object*)
3. Media yang telah mengandung pesan rahasia (*Stegotext* atau *Stego-object*)

### 2.4.2. Sejarah *Steganography* [5]

Penggunaan *steganography* sebetulnya telah digunakan berabad-abad yang lalu. Berikut adalah contoh penggunaan steganografi di masa lalu:

- Selama terjadinya Perang Dunia ke-2, tinta yang tidak tampak (*invisible ink*) telah digunakan untuk menulis informasi pada lembaran kertas sehingga saat kertas tersebut jatuh di tangan pihak lain hanya akan tampak seperti lembaran kertas kosong biasa. Cairan seperti air kencing (urine), susu, vinegar, dan jus buah digunakan sebagai media penulisan sebab bila salah satu elemen tersebut dipanaskan, tulisan akan menggelap dan tampak melalui mata manusia.

- Pada sejarah Yunani kuno, masyarakatnya biasa menggunakan seorang pembawa pesan sebagai perantara pengiriman pesan. Pengirim pesan tersebut akan dicukur rambutnya, untuk kemudian dituliskan suatu pesan pada kepalanya yang sudah botak. Setelah pesan dituliskan, pembawa pesan harus menunggu hingga rambutnya tumbuh kembali sebelum dapat mengirimkan pesan kepada pihak penerima. Pihak penerima kemudian akan mencukur rambut pembawa pesan tersebut untuk melihat pesan yang tersembunyi.
- Metode lain yang digunakan oleh masyarakat Yunani kuno adalah dengan menggunakan lilin sebagai media penyembunyi pesan mereka. Untuk melihat pesan yang disampaikan oleh pihak pengirim pihak penerima harus memanaskan agar lilin tersebut mencair dan pesan rahasia akan keluar.

### 2.4.3. Jenis-jenis *Steganography* [5,22]

Teknik *steganography* gambar atau pada media penampung berbentuk data citra saat ini telah dikembangkan dalam beberapa teknik, yaitu [5,22]:

1. *Domain Spatial Technique*, teknik ini bekerja dengan menyembunyikan informasi pada pixel-pixel yang membentuk sebuah citra yang disebut domain spasial. Teknik ini juga dikenal sebagai teknik substitusi. Salah satu metode yang terkenal dalam teknik ini adalah metode *least significant bit* (LSB).
2. *Transform Domain Technique*, teknik ini bekerja dengan menyisipkan data rahasia ke dalam domain frekuensi, yaitu pada koefesien frekuensi hasil transformasi data cover. Adapun jenis transformasi yang telah dikembangkan dalam *steganography* pada teknik ini, diantaranya adalah :
   a. *Discrete Cosine Transform*
   b. *Fourier Transform*
   c. *Wavelet Transform*

### 2.4.4. *Least Significant Bit-2* (LSB-2) [7,23]

Metode *Least Significant Bit* (LSB) tergolong pada teknik ranah spasial *(spatial/time domain),* yaitu teknik yang memodifikasi langsung nilai *byte* dari *covertext* atau dengan kata lain nilai *byte* yang dapat merepresentasikan intensitas/warna pixel atau amplitudo.

*Least Significant Bit* (LSB) adalah bagian dari barisan data biner (basis dua) yang mempunyai nilai paling tidak berarti/paling kecil. Letaknya adalah paling kanan dari barisan bit. Sedangkan *most significant bit* adalah sebaliknya, yaitu angka yang paling berarti/paling besar dan letaknya disebelah paling kiri. Seperti kita ketahui untuk *file* bitmap 24 bit maka setiap pixel (titik) pada gambar tersebut terdiri dari susunan tiga warna merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (byte) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Dengan demikian pada setiap pixel *file* bitmap 24 bit kita dapat menyisipkan 3 bit data [23].

Sebagai contoh 3 buah *pixel* (sembilan *bytes*) citra digital 24 *bit* dijadikan sebagai medium pesan yang akan disembunyikan dengan pesan teks rahasia adalah karakter T dengan nilai RGB sebagai berikut [7]:

11110101 00010110 10101010
11000100 11111001 00000001
00000001 11110001 00011101

Karakter T dalam biner adalah 01010100, maka dengan menggunakan metode LSB, maka akan dihasilkan citra hasil dengan urutan *bit* akhir sebagai berikut :

1111010**0** 0001011**1** 1010101**0**
1100010**1** 1111100**0** 0000000**0**
0000000**0** 1111000**1** 00011101

Ekstraksi data dilakukan dengan cara memisahkan pesan dari wadah penampung dan tahap-tahap ekstraksi yang dilakukan hanya diizinkan untuk diketahui oleh orang yang dituju. Apabila metode penyembunyian pesan pada wadah dilakukan dengan LSB, maka teknik ekstraksi dilakukan dengan cara yang

sama pula yaitu mengumpulkan kembali *bit-bit* yang tersimpan pada *bit* akhir setiap *byte pixel* wadah penampung kemudian bit-bit tersebut dikonversikan menjadi teks yang memiliki arti seperti teks aslinya.

Perkembangan era digitalisasi yang memungkinkan semakin banyaknya penyerang pada pesan rahasia metode *Least Significant Bit* melakukan beberapa modifikasi. Salah satu modifikasi dari metode *Least Significant Bit* (LSB) adalah *Least Significant Bit-2* (LSB-2).

Metode *Least Significant Bit-2* (LSB-2) ini menukar *bit pixel* medium penampung yang dengan setiap bit pesan yang akan disembunyikan. Apabila pada LSB menukarkan *bit* akhir (*bit* ke-8) maka pada LSB-2 akan menukarkan *bit* ke-8 dikurangi dengan 2 (nilai LSB - 2) atau sama dengan *bit* ke-6.

11110101 00010110 10101010
11000100 11111001 00000001
00000001 11110001 00011101

Karakter T dalam biner adalah 01010100, maka akan dihasilkan citra hasil dengan urutan *bit* akhir sebagai berikut :

1111**0**001 0001**1**110 1010**1**010
1100**1**100 1111**0**001 0000**0**001
0000**0**001 1111**1**001 00011101

Uraian contoh penerapan modifikasi metode *Least Significant Bit-2* (LSB-2) di atas, dapat disimpulkan bahwa *bit-bit* yang ditukarkan tidak lagi berada pada *bit* akhir *byte pixel*, sehingga dapat mempersulit para penyusup untuk mengungkap pesan rahasia yang ada di dalam medium pesan.

### 2.4.5. Jenis-Jenis Serangan *Steganography* [24]

*Watermarking* yang merupakan bagian dari *steganography* adalah suatu cara penyembunyian atau penanaman data atau informasi tertentu (baik hanya berupa catatan umum maupun rahasia) kedalam suatu data digital lainnya, tetapi tidak diketahui kehadirannya oleh indera manusia (indera penglihatan atau indera

pendengaran), dan mampu menghadapi proses-proses pengolahan sinyal digital sampai pada tahap tertentu [24]. Walaupun telah dilakukannya pengaman data dengan teknik *watermarking* pada teknik *steganography*, namun masih terdapat beberapa kemungkinan serangan yang dapat dilakukan diantaranya adalah :

1. *Scrambling Attack.* Serangan dengan melakukan pengacakan pada file yang telah diberi *watermark*. *Scramble* (pengacakan) dilakukan sebelum dilakukan *detection* dan dilakukan *de-scrambled* setelah *detection*. Dengan cara ini *detector* bisa tidak mendeteksi adanya *watermark* pada file. Contoh serangan ini adalah *Mosaic Attack* oleh PetitColas, pada *Mosaic Attack* sebuah file dipecah menjadi banyak bagian kecil yang masing-masing terlalu kecil untuk detektor mendeteksi *watermark*.
2. *Synchronization Attack.* Serangan dengan melakukan transformasi bentuk pada file yang telah diberi *watermark*. Dengan cara ini detektor tidak dapat mendeteksi *watermark*. Contoh serangan ini adalah *StirMark Attack* oleh Petit Colas, Transformasi pada *StirMark Attack*
3. *Linear Filtering dan Noise Removal*, Melakukan serangan pada file yang telah diberi *watermark* dengan berusaha mengurangi/menghapus *noise*/*frequency* yang tidak diinginkan pada file (dalam hal ini adalah *watermark* pada file). Contoh aplikasi serangan ini adalah *Host Data Estimation* yang dilakukan oleh Kutter.
4. *Copy Attack,* Serangan yang dilakukan terhadap *fragile watermark.* Serangan ini dilakukan dengan cara mencari *pattern watermark* pada sebuah file dan melakukan *embed pattern watermark* tersebut kedalam file lain. Hal ini menyebabkan *watermark* yang valid di-*embed* kepada file yang berbeda. Contoh serangan ini adalah *Collage* Attack oleh Holliman.

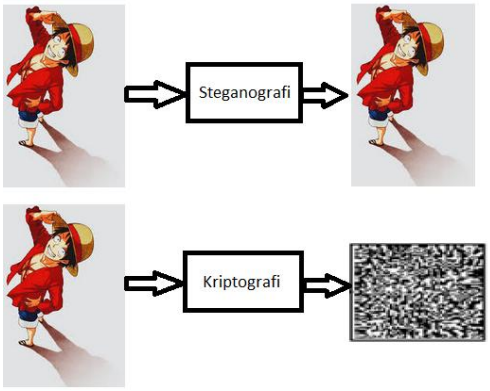
## 2.5. Perbedaan *Cryptography* dan *Steganography* [5]

**Tabel 2.1**

Perbedaan *Cryptography* dan *Steganography*

| *Cryptography* | *Steganography* |
|---|---|
| Hasil keluaran dari *cryptography* biasanya berupa data yang berbeda dari bentuk aslinya dan biasanya data seolah-olah berantakan sehingga tidak dapat diketahui informasi apa yang terkandung didalamnya (namun sesungguhnya dapat dikembalikan ke bentuk semula lewat proses dekripsi) | Hasil keluaran dari *steganography* memiliki bentuk persepsi yang sama dengan bentuk aslinya. Kesamaan persepsi tersebut adalah oleh indera manusia (khususnya visual), namun bila digunakan komputer atau perangkat pengolah digital lainnya dapat dengan jelas dibedakan antara sebelum proses dan setelah proses |

Pada tabel 2.1 merupakan penjelasan dari perbedaan antara algoritma *cryptography* dengan algoritma *steganography*. Perbedaan pada hasil keluaran dari algoritma *cryptography* adalah berupa data yang berbeda dengan bentuk aslinya, sedangkan untuk hasil keluaran algoritma *steganography* adalah berupa data yang sama dengan file aslinya. Adapun contoh gambar pada hasil keluaran steganografi dan kriptografi ditampilkan pada gambar 2.5 berikut ini :




**Gambar 2.5**

Perbedaan Steganografi dengan Kriptografi [5].

## 2.6. Kode ASCII [16]

Kode ASCII (*American Standard Code for Information Interchange*) merupakan representasi numerik dari suatu karakter seperti ‘a’ atau ‘@’ atau karakter yang tidak tercetak, misalnya ‘Σ’. ASCII merupakan kombinasi kode 8 bit, yang terdiri atas 7 bit data dan 1 bit parity, sehingga mempunyai $2^7$ atau 128 kode karakter yang berbeda dan unik yang terdiri dari bit 0 dan bit 1.

**Tabel 2.2**

Kode ASCII [2]

| DEC | OCT | HEX | BIN | CHAR |
|---|---|---|---|---|
| 1 | 1 | 1 | 00000001 | |
| 2 | 2 | 2 | 00000010 | ┐ |
| 3 | 3 | 3 | 00000011 | └ |
| 4 | 4 | 4 | 00000100 | ┘ |
| 5 | 5 | 5 | 00000101 | │ |
| 6 | 6 | 6 | 00000110 | ─ |
| 7 | 7 | 7 | 00000111 | • |
| 8 | 10 | 8 | 00001000 | ◘ |
| 9 | 11 | 9 | 00001001 | |
| 10 | 12 | A | 00001010 | |
| 11 | 13 | B | 00001011 | ♂ |
| 12 | 14 | C | 00001100 | ♀ |
| 13 | 15 | D | 00001101 | |
| 14 | 16 | E | 00001110 | ♫ |
| 15 | 17 | F | 00001111 | ☼ |
| 16 | 20 | 10 | 00010000 | ┼ |
| 17 | 21 | 11 | 00010001 | ◄ |
| 18 | 22 | 12 | 00010010 | ↕ |
| 19 | 23 | 13 | 00010011 | ‼ |
| 20 | 24 | 14 | 00010100 | ¶ |
| 21 | 25 | 15 | 00010101 | ┴ |
| 22 | 26 | 16 | 00010110 | ┬ |
| 23 | 27 | 17 | 00010111 | ┤ |
| 24 | 30 | 18 | 00011000 | ↑ |
| 25 | 31 | 19 | 00011001 | ├ |
| 26 | 32 | 1A | 00011010 | → |
| 27 | 33 | 1B | 00011011 | ← |
| 28 | 34 | 1C | 00011100 | |
| 29 | 35 | 1D | 00011101 | |
| 30 | 36 | 1E | 00011110 | |
| 31 | 37 | 1F | 00011111 | |
| 32 | 40 | 20 | 00100000 | |
| 33 | 41 | 21 | 00100001 | ! |
| 34 | 42 | 22 | 00100010 | " |
| 35 | 43 | 23 | 00100011 | # |
| 36 | 44 | 24 | 00100100 | $ |
| 37 | 45 | 25 | 00100101 | % |
| 38 | 46 | 26 | 00100110 | & |
| 39 | 47 | 27 | 00100111 | ' |
| 40 | 50 | 28 | 00101000 | ( |
| 41 | 51 | 29 | 00101001 | ) |
| 42 | 52 | 2A | 00101010 | * |
| 43 | 53 | 2B | 00101011 | + |
| 44 | 54 | 2C | 00101100 | , |
| 45 | 55 | 2D | 00101101 | - |
| 46 | 56 | 2E | 00101110 | . |
| 47 | 57 | 2F | 00101111 | / |
| 48 | 60 | 30 | 00110000 | 0 |
| 49 | 61 | 31 | 00110001 | 1 |
| 50 | 62 | 32 | 00110010 | 2 |
| 51 | 63 | 33 | 00110011 | 3 |
| 52 | 64 | 34 | 00110100 | 4 |

| DEC | OCT | HEX | BIN | CHAR |
|---|---|---|---|---|
| 86 | 126 | 56 | 01010110 | V |
| 87 | 127 | 57 | 01010111 | W |
| 88 | 130 | 58 | 01011000 | X |
| 89 | 131 | 59 | 01011001 | Y |
| 90 | 132 | 5A | 01011010 | Z |
| 91 | 133 | 5B | 01011011 | [ |
| 92 | 134 | 5C | 01011100 | \ |
| 93 | 135 | 5D | 01011101 | ] |
| 94 | 136 | 5E | 01011110 | ^ |
| 95 | 137 | 5F | 01011111 | _ |
| 96 | 140 | 60 | 01100000 | ` |
| 97 | 141 | 61 | 01100001 | a |
| 98 | 142 | 62 | 01100010 | b |
| 99 | 143 | 63 | 01100011 | c |
| 100 | 144 | 64 | 01100100 | d |
| 101 | 145 | 65 | 01100101 | e |
| 102 | 146 | 66 | 01100110 | f |
| 103 | 147 | 67 | 01100111 | g |
| 104 | 150 | 68 | 01101000 | h |
| 105 | 151 | 69 | 01101001 | i |
| 106 | 152 | 6A | 01101010 | j |
| 107 | 153 | 6B | 01101011 | k |
| 108 | 154 | 6C | 01101100 | l |
| 109 | 155 | 6D | 01101101 | m |
| 110 | 156 | 6E | 01101110 | n |
| 111 | 157 | 6F | 01101111 | o |
| 112 | 160 | 70 | 01110000 | p |
| 113 | 161 | 71 | 01110001 | q |
| 114 | 162 | 72 | 01110010 | r |
| 115 | 163 | 73 | 01110011 | s |
| 116 | 164 | 74 | 01110100 | t |
| 117 | 165 | 75 | 01110101 | u |
| 118 | 166 | 76 | 01110110 | v |
| 119 | 167 | 77 | 01110111 | w |
| 120 | 170 | 78 | 01111000 | x |
| 121 | 171 | 79 | 01111001 | y |
| 122 | 172 | 7A | 01111010 | z |
| 123 | 173 | 7B | 01111011 | { |
| 124 | 174 | 7C | 01111100 | \| |
| 125 | 175 | 7D | 01111101 | } |
| 126 | 176 | 7E | 01111110 | ~ |
| 127 | 177 | 7F | 01111111 | ⌂ |
| 128 | 200 | 80 | 10000000 | € |
| 129 | 201 | 81 | 10000001 | |
| 130 | 202 | 82 | 10000010 | ‚ |
| 131 | 203 | 83 | 10000011 | ƒ |
| 132 | 204 | 84 | 10000100 | „ |
| 133 | 205 | 85 | 10000101 | … |
| 134 | 206 | 86 | 10000110 | † |
| 135 | 207 | 87 | 10000111 | ‡ |
| 136 | 210 | 88 | 10001000 | ˆ |
| 137 | 211 | 89 | 10001001 | ‰ |

| DEC | OCT | HEX | BIN | CHAR |
|---|---|---|---|---|
| 171 | 253 | AB | 10101011 | « |
| 172 | 254 | AC | 10101100 | ¬ |
| 173 | 255 | AD | 10101101 | - |
| 174 | 256 | AE | 10101110 | ® |
| 175 | 257 | AF | 10101111 | ¯ |
| 176 | 260 | B0 | 10110000 | ° |
| 177 | 261 | B1 | 10110001 | ± |
| 178 | 262 | B2 | 10110010 | ² |
| 179 | 263 | B3 | 10110011 | ³ |
| 180 | 264 | B4 | 10110100 | ´ |
| 181 | 265 | B5 | 10110101 | µ |
| 182 | 266 | B6 | 10110110 | ¶ |
| 183 | 267 | B7 | 10110111 | · |
| 184 | 270 | B8 | 10111000 | ¸ |
| 185 | 271 | B9 | 10111001 | ¹ |
| 186 | 272 | BA | 10111010 | º |
| 187 | 273 | BB | 10111011 | » |
| 188 | 274 | BC | 10111100 | ¼ |
| 189 | 275 | BD | 10111101 | ½ |
| 190 | 276 | BE | 10111110 | ¾ |
| 191 | 277 | BF | 10111111 | ¿ |
| 192 | 300 | C0 | 11000000 | À |
| 193 | 301 | C1 | 11000001 | Á |
| 194 | 302 | C2 | 11000010 | Â |
| 195 | 303 | C3 | 11000011 | Ã |
| 196 | 304 | C4 | 11000100 | Ä |
| 197 | 305 | C5 | 11000101 | Å |
| 198 | 306 | C6 | 11000110 | Æ |
| 199 | 307 | C7 | 11000111 | Ç |
| 200 | 310 | C8 | 11001000 | È |
| 201 | 311 | C9 | 11001001 | É |
| 202 | 312 | CA | 11001010 | Ê |
| 203 | 313 | CB | 11001011 | Ë |
| 204 | 314 | CC | 11001100 | Ì |
| 205 | 315 | CD | 11001101 | Í |
| 206 | 316 | CE | 11001110 | Î |
| 207 | 317 | CF | 11001111 | Ï |
| 208 | 320 | D0 | 11010000 | Ð |
| 209 | 321 | D1 | 11010001 | Ñ |
| 210 | 322 | D2 | 11010010 | Ò |
| 211 | 323 | D3 | 11010011 | Ó |
| 212 | 324 | D4 | 11010100 | Ô |
| 213 | 325 | D5 | 11010101 | Õ |
| 214 | 326 | D6 | 11010110 | Ö |
| 215 | 327 | D7 | 11010111 | × |
| 216 | 330 | D8 | 11011000 | Ø |
| 217 | 331 | D9 | 11011001 | Ù |
| 218 | 332 | DA | 11011010 | Ú |
| 219 | 333 | DB | 11011011 | Û |
| 220 | 334 | DC | 11011100 | Ü |
| 221 | 335 | DD | 11011101 | Ý |
| 222 | 336 | DE | 11011110 | Þ |

| | | | | |
|---|---|---|---|---|
| 52 | 64 | 34 | 00110100 | 4 |
| 53 | 65 | 35 | 00110101 | 5 |
| 54 | 66 | 36 | 00110110 | 6 |
| 55 | 67 | 37 | 00110111 | 7 |
| 56 | 70 | 38 | 00111000 | 8 |
| 57 | 71 | 39 | 00111001 | 9 |
| 58 | 72 | 3A | 00111010 | : |
| 59 | 73 | 3B | 00111011 | ; |
| 60 | 74 | 3C | 00111100 | < |
| 61 | 75 | 3D | 00111101 | = |
| 62 | 76 | 3E | 00111110 | > |
| 63 | 77 | 3F | 00111111 | ? |
| 64 | 100 | 40 | 01000000 | @ |
| 65 | 101 | 41 | 01000001 | A |
| 66 | 102 | 42 | 01000010 | B |
| 67 | 103 | 43 | 01000011 | C |
| 68 | 104 | 44 | 01000100 | D |
| 69 | 105 | 45 | 01000101 | E |
| 70 | 106 | 46 | 01000110 | F |
| 71 | 107 | 47 | 01000111 | G |
| 72 | 110 | 48 | 01001000 | H |
| 73 | 111 | 49 | 01001001 | I |
| 74 | 112 | 4A | 01001010 | J |
| 75 | 113 | 4B | 01001011 | K |
| 76 | 114 | 4C | 01001100 | L |
| 77 | 115 | 4D | 01001101 | M |
| 78 | 116 | 4E | 01001110 | N |
| 79 | 117 | 4F | 01001111 | O |
| 80 | 120 | 50 | 01010000 | P |
| 81 | 121 | 51 | 01010001 | Q |
| 82 | 122 | 52 | 01010010 | R |
| 83 | 123 | 53 | 01010011 | S |
| 84 | 124 | 54 | 01010100 | T |
| 85 | 125 | 55 | 01010101 | U |

| | | | | |
|---|---|---|---|---|
| 137 | 211 | 89 | 10001001 | ‰ |
| 138 | 212 | 8A | 10001010 | Š |
| 139 | 213 | 8B | 10001011 | ‹ |
| 140 | 214 | 8C | 10001100 | Œ |
| 141 | 215 | 8D | 10001101 | |
| 142 | 216 | 8E | 10001110 | Ž |
| 143 | 217 | 8F | 10001111 | |
| 144 | 220 | 90 | 10010000 | |
| 145 | 221 | 91 | 10010001 | ‘ |
| 146 | 222 | 92 | 10010010 | ’ |
| 147 | 223 | 93 | 10010011 | “ |
| 148 | 224 | 94 | 10010100 | ” |
| 149 | 225 | 95 | 10010101 | • |
| 150 | 226 | 96 | 10010110 | – |
| 151 | 227 | 97 | 10010111 | — |
| 152 | 230 | 98 | 10011000 | ˜ |
| 153 | 231 | 99 | 10011001 | ™ |
| 154 | 232 | 9A | 10011010 | š |
| 155 | 233 | 9B | 10011011 | › |
| 156 | 234 | 9C | 10011100 | œ |
| 157 | 235 | 9D | 10011101 | |
| 158 | 236 | 9E | 10011110 | ž |
| 159 | 237 | 9F | 10011111 | Ÿ |
| 160 | 240 | A0 | 10100000 | |
| 161 | 241 | A1 | 10100001 | ¡ |
| 162 | 242 | A2 | 10100010 | ¢ |
| 163 | 243 | A3 | 10100011 | £ |
| 164 | 244 | A4 | 10100100 | ¤ |
| 165 | 245 | A5 | 10100101 | ¥ |
| 166 | 246 | A6 | 10100110 | ¦ |
| 167 | 247 | A7 | 10100111 | § |
| 168 | 250 | A8 | 10101000 | ¨ |
| 169 | 251 | A9 | 10101001 | © |
| 170 | 252 | AA | 10101010 | ª |

| | | | | |
|---|---|---|---|---|
| 222 | 336 | DE | 11011110 | Þ |
| 223 | 337 | DF | 11011111 | ß |
| 224 | 340 | E0 | 11100000 | à |
| 225 | 341 | E1 | 11100001 | á |
| 226 | 342 | E2 | 11100010 | â |
| 227 | 343 | E3 | 11100011 | ã |
| 228 | 344 | E4 | 11100100 | ä |
| 229 | 345 | E5 | 11100101 | å |
| 230 | 346 | E6 | 11100110 | æ |
| 231 | 347 | E7 | 11100111 | ç |
| 232 | 350 | E8 | 11101000 | è |
| 233 | 351 | E9 | 11101001 | é |
| 234 | 352 | EA | 11101010 | ê |
| 235 | 353 | EB | 11101011 | ë |
| 236 | 354 | EC | 11101100 | ì |
| 237 | 355 | ED | 11101101 | í |
| 238 | 356 | EE | 11101110 | î |
| 239 | 357 | EF | 11101111 | ï |
| 240 | 360 | F0 | 11110000 | ð |
| 241 | 361 | F1 | 11110001 | ñ |
| 242 | 362 | F2 | 11110010 | ò |
| 243 | 363 | F3 | 11110011 | ó |
| 244 | 364 | F4 | 11110100 | ô |
| 245 | 365 | F5 | 11110101 | õ |
| 246 | 366 | F6 | 11110110 | ö |
| 247 | 367 | F7 | 11110111 | ÷ |
| 248 | 370 | F8 | 11111000 | ø |
| 249 | 371 | F9 | 11111001 | ù |
| 250 | 372 | FA | 11111010 | ú |
| 251 | 373 | FB | 11111011 | û |
| 252 | 374 | FC | 11111100 | ü |
| 253 | 375 | FD | 11111101 | ý |
| 254 | 376 | FE | 11111110 | þ |
| 255 | 377 | FF | 11111111 | ÿ |

Pada tabel 2.2 menunjukkan 255 karakter ASCII dan 32 karakter yang tidak tercetak dengan informasi desimal, oct, hex, biner pada tiap tiap karakternya. Pada tampilan biner karakter ASCII disediakan biner dengan 8 bit tiap karakternya. Kode ini digunakan dalam Personal Computer (PC). Piranti yang menggunakan kode ini perlu menterjemahkan 1 bit didepan sebagai parity. Bit parity berfungsi sebagai tanda kesalahan dalam pengiriman data selama komunikasi, yang terdiri atas parity genap (bit 1 apabila jumlah bit 1 dalam 7 deretan bit data berjumlah genap) dan parity ganjil (bit 1 apabila jumlah bit 1 dalam 7 deretan bit data berjumlah ganjil) [16].

## 2.7. Android [2,25]

Dirancang untuk memudahkan pengembahangan aplikasi membuat aplikasi dengan batasan yang minim sehingga kreatifitas pengembangan menjadi lebih berkembang. Android menyediakan *platform* terbuka (*open sources*) bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai macam piranti bergerak.

Pengembangan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Dengan sistem distribusi open sources yang digunakan Android, memungkinkan para pengembang untuk menciptakan berbagai macam aplikasi menarik yang dapat dinikmati oleh para penggunanya, seperti *game, chatting* dan lain-lain, hal ini pulalah yang membuat smartphone berbasis Android ini lebih murah dibanding *gadget* sejenisnya [25].

Adapun aplikasi salah satu platform android adalah android studio. Android Studio adalah sebuah IDE untuk Android *Development* yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangkan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android [2].

Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan *Ant*, Android Studio menggunakan *Gradle* sebagai *build environment*. Fitur-fitur lainnya adalah sebagai berikut:

- Menggunakan *Gradle-based build system* yang fleksibel.
- Bisa mem-*build multiple* APK .
- Template *support* untuk Google *Services* dan berbagai macam tipe perangkat.
- *Layout* editor yang lebih bagus.
- *Built-in support* untuk *Google Cloud Platform,* sehingga mudah untuk integrasi dengan Google *Cloud Messaging* dan *App Engine.*
- *Import library* langsung dari Maven repository, dan lainnya.

## 2.8. Program Java [2,26]

Bahasa pemrograman java digolongkan dalam kategori bahasa pemrograman level tinggi (*High Level Languag*e), karena penggunaan struktur bahasanya yang mudah dimengerti oleh manusia. Namun, tidak jarang para pemula yang ingin mempelajari konsep dasar java masih mengalami kesulitan. Program java dikembangkan oleh Sun Microsystem pada tahun 1995, juga merupakan sebuah bahasa berorientasi objek. Digunakan untuk menuliskan program-program yang padat dan bisa diunduh melalui 3 internet dan dengan segera dieksekusi pada banyak komputer.

File kode sumber java (file dengan eksistensi java) dikompilasi kedalam format yang disebut *byte code* (file dan eksistensi *class*), yang nantinya bisa dieksekusi oleh penerjemah (*interpreter*) java. Kode java yang telah dikompilasi bisa berjalan pada sebagian besar komputer karena penerjemah java dan lingkungan untuk menjalankannya, Java Virtual *Mechines* (JVMs), telah tersedia di sebagian besar Sistem Operasi. Pada dasarnya, java dibagi menjadi 3 kategori, yaitu [26]:

a. J2SE (Standart Edition) : Pemrograman Java yang berbasis *Desktop Programming*.
b. J2ME (Micro/Mobile Edition) : Pemrograman Java yang berbasis *Mobile*, biasanya digunakan untuk handphone dan *chip* pada kartu tertentu.
c. J2EE (Enterprise Edition) : Pemrograman Java yang berbasis jaringan/network, sehingga dapat diakses melalui *web browser*.

## 2.9. Program XML [1,4,22]

XML (*eXtensible Markup Language*) merupakan bahasa web turunan dari SGML (*Standard Generalized Markup Language*) yang ada sebelumnya. XML hampir sama dengan HTML, dimana kedua-duanya diturunkan dari SGML.

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa metamarkup yang menyediakan format tertentu untuk dokumen-dokumen yang menyediakan format tertentu untuk dokumen-dokumen yang mempunyai data terstruktur. Bahasa Markup adalah mekanisme untuk mengenal suatu struktur didokumen [1].

Sebuah dokumen XML terdiri dari bagian bagian yang disebut dengan node. Node-node itu diantaranya adalah [22]:

a. Root node yaitu node yang melingkupi keseluruhan dokumen. Dalam satu dokumen XML hanya ada satu root node.
b. Element node yaitu bagian dari dokumen XML yang ditandai dengan tag pembuka dan tag penutup, atau bisa juga sebuah tag tunggal elemen kosong seperti Root node biasa juga disebut root element.
c. Attribute note termasuk nama dan nilai atribut ditulis pada tag awal sebuah elemen atau pada tag tunggal.
d. Text node, adalah text yang merupakan isi dari sebuah elemen, ditulis diantara tag pembuka dan tag penutup.

## 2.10. Matlab [27]

Matlab (Matrix Laboratory) merupakan merel dagang yang dikembangkan oleh Math Works Inc. Perangkat lunak ini luas digunakan dalam bidang sains dan rekayasa. Matlab didukung oleh sistem operasi Unix, Macintosh dan Windows.

Matlab mengintegrasikan komputasi matematik, visualisasi dan bahasa pemrograman untuk memberikan lingkungan fleksibel bagi komputasi teknis. Arsitektur terbukanya membuat pengguna mudah dalam mengeksplorasi data, menciptakan algoritma dan menciptakan beberapa perangkat grafik (GUI). Matlab dikenal dengan perhitungan vektor dan matriks dengan kecepatan tinggi, Matlab menawarkan solusi terhadap permasalahan secara matematik dan secara visual.

Matlab juga merupakan paket perangkat lunak yang memampukan anda untuk melakukan komputasi matematik, menganalisa data, mengembangkan algoritma, melakukan simulasi dan pemodelan, dan menghasilkan tampilan grafik dan antar-muka grafikal yang dikembangkan dari bahasa C, yang sangat mudah digunakan. Matlab untuk matematika simbolik juga akan disajikan. Kegunaan dari fungsi-fungsi tersebut adalah untuk melakukan operasi-operasi simbolik dalam menyelesaikan persamaan aljabar, persamaan diferensial biasa dan lain-lain. Matematika simbol juga dapat dipakai untuk menentukan ekspresi analitik pada persamaan diferensial dan persamaan integral.

## 2.11. Penelitian Terdahulu

Sebagai bahan pendukung penyusunan laporan tugas akhir ini, penulis mengambil beberapa referensi yang diambil dari berbagai laporan penelitian terdahulu serta jurnal yang berhubungan dengan keamanan data informasi menggunakan algoritma *caesar cipher cryptography* dan least significant bit *steganography*. Adapun beberapa penelitian terdahulu ditampilkan pada tabel 2.3 di bawah ini.

**Tabel 2.3**

Penelitian Terdahulu

| No | Nama Peneliti | Tahun | Judul | Metode | Hasil |
|---|---|---|---|---|---|
| 1 | Taronisokhi Zebua | 2015 | Penerapan Metode LSB-2 Untuk Menyembunyikan Ciphertext Pada Citra Digital | Steganografi LSB-2 dan Kriptografi Triangle Chain Cipher | Penerapan kombinasi kemanan dengan algoritma Triangle Chain Cipher untuk menghasilkan stegano image. Citra cover yang digunakan adalah file gambar dengan jenis bitmap menggunakan aplikasi visual studio 2008 |
| 2 | Bonifacius Vicky Indriyono | 2016 | Implementasi Sistem Keamanan File dengan Metode Steganografi EOF dan Enkripsi Caesar Cipher | Steganografi EOF dan Kriptografi Caesar Cipher | Aplikasi computer dengan menggunakan tampilan GUI menggunakan kombinasi metode Steganografi EOF dan Kriptografi Caesar Cipher media yang digunakan berupa gambar, video dan audio |

| | | | | | |
|---|---|---|---|---|---|
| 3 | Imam Wahyu Utomo, Retnani Latifah dan Rita Dewi Risanty | 2017 | Aplikasi Kriptografi Berbasis Android Menggunakan Algoritma Caesar Cipher dan Vigenere Cipher | Kriptografi Caesar Cipher dan Vigenere Cipher | Aplikasi android dengan melakukan kombonasi antar 2 metode kriptografi dengan tingkat kesuksesan 75% dengan menggunakan karakter ASCII dengan nomor index antara 32 sampai 125 (total 94) |
| 4 | Primaningtyas Nur Arifah dan Windi Agustiar Basuki | 2017 | Implementasi Kriptografi *Caesar Cipher* Menggunakan Matlab R2013a | Kriptografi Caesar Cipher | Modifikasi Metode kemanan Kriptografi Caesar Cipher dengan dilakukannya perbaikan pada proses enkripsi dekrispi dengan menambahkan tanda spasi, beberapa tanda baca, angka dan penggunaan huruf capital dan huruf kecil, dengan jumlah mod adalah 41 |
| 5 | Muhammad Ridwan Rambe, Edy Victor Haryanto dan Adil Setiawan | 2018 | Aplikasi Pengamanan Data dan Disisipkan Pada Gambar dengan Algoritma RSA dan Modified LSB Berbasis Android | Kriptografi Algoritma RSA dan Steganografi LSB | Aplikasi kombinasi metode keamanan menggunakan netode kriptografi RSA dan Steganografi *modified* LSB pada perangkat android yang dibuat dengan pemrograman Eclipse |

| | | | | | |
|---|---|---|---|---|---|
| 6 | Rizky Maynarda, Ahmad Setiadi, Pas Mahyu Akhirianto, Marianus Lase, dan Agus Junaidi | 2018 | Implementasi *Steganografi* dengan Metode *Least Significant Bit* Menggunakan Bahasa Android | Steganografi LSB | Steganografi Aplikasi android dengan menggunakan metode LSB ini menggunakan *Software Development kit* (SDK) yang umum dan ringan, *Eclipse luna* dengan hasil aplikasi berupa *file apk* |
| 7 | M. Ali Fahroni | 2018 | Sistem Keamanan Transmisi Data Menggunakan Steganography Dengan Berbasis Android | Steganografi LSB | Aplikasi Android yang digunakan adalah data berupa biner gambar yang telah disisipkan pesan dengan maksimal pesan yang disisipkan dibawah 8MB. Penggunaan media gambar sebagai data masukan media pembawa pesan rahasia berupa gambar dengan format JPG |

| | | | | | |
|---|---|---|---|---|---|
| 8 | Indra Gunawan dan Sumarno | 2018 | Penggunaan Algoritma Kriptografi Steganografi *Least Significant Bit* Untuk Pengamanan Pesan Teks dan Data Video | Steganografi LSB dan EOF | Pengaplikasian peningkatan kemanan dengan metode steganografi menggunakan computer berupa teks dan video dengan cara menyisipkan teks ke dalam video, mengacak video sehingga tidak dapat dilihat |
| 9 | Aliy Hafiz | 2019 | Steganografi Berbasis Citra Digital untuk Menyembunyikan Data Menggunakan Metode *Least Significant Bit* (LSB) | Steganografi LSB | Aplikasi Komputer penyisipan pesan tersembunyi berupa data dapat dilakukan ke dalam wadah citra digital berformat JPEG dan format citra digital lainnya, mengekstraksi kembali data tersembunyi tersebut dari dalam citra digital |
| 10 | Lekso Budi Handoko dan Chaerul Umam | 2019 | Penyembunyian Pesan Menggunakan Steganografi dengan Metode LSB dan Enkripsi Kriptografi | Steganografi LSB dan Kriptografi Caesar Cipher | Analisa dari Proses Penyembunyian pesan menggunakan kriptografi dan steganografi terbagi menjadi empat, yaitu *Encode, Decode, Enkripsi Dekripsi*, dengan tujuan utama untuk mengamankan substansi agar sulit teridentifikasi |

| | | | | | |
|---|---|---|---|---|---|
| 11 | Andre Hernandes, Hartini dan Dewi Sartika | 2019 | Steganografi Citra Menggunakan Metode *Least Significant Bit* (LSB) dan *Linear Congruental Generator* (LCG) | Steganografi LSB dan LCG | Aplikasi steganografi dengan bahasa pemrograman java dan menguji kualitas stego image yang menghasilkan nilai rata-rata PSNR mencapai 51 *dB*, dan hasilnya berupa image stego dengan kualitas baik dan tidak mengalami perubahan |
| 12 | Muhammad Arif Ismirianda, Setia Juli Irzal Ismail dan Anang Sularsa | 2019 | Implementasi Teknik Kriptografi dan Steganografi pada Aplikasi Android | Steganografi LSB dan Kriptografi AES | Aplikasi Android menggunakan kombinasi metode Steganografi LSB dan Kriptografi AES menggunakan file gambar |
| 13 | Muhammad Syahril dan Hendra Jaya | 2019 | Aplikasi Steganografi Pengamanan Data Nasabah di Standard Chartered Bank Menggunakan Metode Least Significant Bit dan RC4 | Steganografi LSB dan RC4 | Aplikasi computer menggunakan metode Steganografi LSB dan RC4 keperluan data nasabah pada *Standard Chartered Bank* |

| | | | | | |
|---|---|---|---|---|---|
| 14 | Hermansa, Rusyidi Umar dan Anton Yudhana | 2019 | Analisis Sistem Keamanan Teknik Kriptografi dan Steganografi pada Citra Digital (*BITMAP*) | Steganografi LSB-2 dan Kriptografi Hill Cipher | Analisis metode keamanan pesan menggunakan Steganografi LSB-2 dan Kriptografi Hill Cipher pada citra digital bitmap, sehingga tidak mengalami perubahan setelah proses penyisipan biner teks ke dalam biner bitmap. |
| 15 | Rindy Febrianingsih dan Aliy Hafiz | 2019 | Implementasi Kriptografi Berbasis Caesar Cipher Untuk Kemanan Data | Kriptografi Caesar Cipher | Aplikasi Kriptografi Caesar Cipher mod 26 menggunakan *extreme programming* DELPHI, aplikasi ini dapat mengacak file sehingga tidak bias dibaca membuat data menjadi lebih aman |