BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang berfokus pada keamanan pada pesan yang disampaikan terhadap ancaman seperti SMS spoofing,SMS snoopingdan SMS interception. Untuk mengurangi risiko tersebut, makadibutuhkan sebuah sistem keamanan pada layanan SMS yang mampu menjagaintegritas dan keamanan isi pesan. Dimana tujuannya ialah untuk menutupicelah pada tingkat keamanan SMS. Salah satu penanggulangannya ialahdengan menerapkan algoritma kriptografi, yaitu kombinasi atas algoritmaCipher Disk, Caesar, dan Scytale pada pesan yang akan dikirim. Tujuan daripenelitian ini adalah membangun aplikasi LumaSMS, dengan menggunakankombinasi ketiga algoritma kriptografi tersebut. Dengan adanya aplikasi inidiharapkan mampu mengurangi masalah keamanan dan integritas SMS (Triyuswoyo, 2010).

Penelitian yang berfokus pada pengamanan *filetype text* menggunakan *kriptography*dengan metode caesar. Pengamanan informasi tersebut selain bertujuan untuk meningkatkan keamanan,juga berfungsi untuk melindungi pesan atau informasi agar tidak dapat di akses,menyisipkan dan atau menghapus pesan oleh orang yang tidak berhak.*Kriptography* merupakan seni dan ilmu pengetahuan untuk menjaga keamananinformasi. Kebalikan dari *kriptography* adalah *kryptanalysis*, yaitu seni dan ilmu untukmemecahkan *ciphertext* menjadi *plaintext* tanpa melalui cara yang seharusnya.*Kriptography* menggunakan metode *Caesar Chiper* memiliki kecepatan enkripsiyang cukup baik. Hal ini disebabkan proses enkripsinya cukup sederhana dan hanyamelibatkan beberapa operasi saja setiap *byte* (Praptomo, 1999).

Penelitian yang difokuskan pada penerapan algoritma *caesar cipher* dan algoritma *vigenere* ini bertujuan untuk mengamankan pesan teks. algoritma*caesar cipher* dan *vigenere cipher*,algoritma *caesar cipher* dan *vigenere cipher* termasukkriptografi klasik yang menggunakan *plaintext*,*ciphertext* dan kunci untuk melakukan proses enkripsidan dekripsi dalam pengamanan data.

Algoritma*caesar chiper* adalah teknik kriptografi yangdilakukan dengan mensubtitusi setiap abjad dari pesanyang akan dienkripsi melalui pergeseran susunansebagai kuncinya (Priyono, 2016).

Penelitian yang difokuskan pada pengamanan kerahasiaan informasi terutama yang berisi pesan sensitif. Salah satu cara yangdigunakan untuk pengamanan data adalah menggunakansistem kriptografi yaitu dengan menyediakan isiinformasi (*plaintext*) tersebutmenjadi isi yang tidak dipahami melalui proses enkripsi (*encipher*), danuntukmemperoleh kembali informasi yang asli, dilakukan proses deskripsi (*decipher*), adapun algoritma enkripsiyang digunakan adalah *base*64 yang di mana di dalam proses enkripsi dari *plaintext* ke *ciphertext* melaluitahapan dari ASCII di ubah ke*biner* kemudian di ubah ke desimal yang akan menggunakan *ciphertext*. Untukmembangun aplikasi yang terkomputerisasi, penelitian ini menggunakan Java Eclipce dan notepad++sebagai aplikasi pendukungnya.Aplikasi ini dibuat dalam *Platform* Windows sehingga memudahkan pemakaiuntuk menggunakannya.Dalam pembuatan desain implementasi teknik kriptografi untuk pengamanfile teksdengan menggunakan algoritma kriptografi dan *base*64*alphabet* yang digunakan untuk implementasi enkripsidan deskripsi file teks sebagaikomunikasi yang aman (Hidayatullah, 2012).

Penelitian yang difokuskan pada perbandingan kriptografi RSA dengan *base*64 ini bertujuan untuk membandingkan ukuran dan waktu proses untuk mencari algoritma mana yang lebih baik. Sampai saat ini belum adapenyerangan terhadap RSA yang efektif.Namun apabila pemilihan parameter dan implementasiyang tidak tepat terhadap RSA dapat merupakan titik lemah sistem RSA sehingga rentan untukdiserang. Kriptografi transformasi *base*64 banyak digunakan didunia internet sehingga mediadata format untuk mengirimkan data, ini dikarenakan hasil dari *base*64 berupa *plaintext* makadata ini akan jauh lebih mudah dikirim, dibandingkan dengan format data yang berupa *biner* (Ginting, 2017).

2.2 Landasan Teori

2.2.1 Google Chrome

Google Chrome adalah *web browser* gratis yang dikembangkan oleh Google. Chrome pertama kali dirilis pada bulan September 2008, untuk Microsoft Windows, dan kemudian dikirim ke Linux, macOS, iOS dan Android. Google Chrome juga merupakan komponen utama Chrome OS, yang berfungsi sebagai platform untuk menjalankan aplikasi *web*.

Google merilis sebagian besar sumber kodenya sebagai proyek *open-source*.Salah satu komponen yang bukan *open-source* adalah Adobe Flash Player bawaan (yang telah dinonaktifkan Chrome secara *default* sejak September 2016). Chrome menggunakan*WebKit layout engine* sampai versi 27. Seperti versi 28, semua port Chrome kecuali port iOS menggunakan *Blink*.

Pada tahun 2018, StatCounter memperkirakan bahwa Google Chrome memiliki pangsa penggunaan *browser web* sebesar 66% di seluruh dunia sebagai *browser desktop*. Chrome juga memiliki pangsa pasar 56% di semua *platform* yang digabungkan, karena memiliki lebih dari 50% pangsa pada smartphone. Keberhasilannya telah menyebabkan Google mengembangkan nama merek "Chrome" di berbagai produk lainnya seperti Chromecast, Chromebook, Chromebit, Chromebox dan Chromebase (Paul, 2008).

2.2.2Apache

Server HTTP Apache atau Server Web/WWW Apache adalah *web server*yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell *Netware* serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs *web*. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain.Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah.

Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang di bawah naungan Apache Software Foundation(Apache Software Foundation, 1997).

2.2.3PHP

PHP merupakan singkatan dari PHP *Hypertext Preprocessor*.PHPdigunakan sebagai bahasa *script server side*dalam pengembangan *web* yang disisipkan pada sokumen HTML.Penggunaan PHP memungkinkan *web* dapat dibuat dinamis sehingga maintenance web menjadi lebih mudah dan efisien.PHP ditulis menggunakan bahasa C.

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa scripting lainya. PHP difokuskan pada pembuatan script server side yang bisa melakukan apa saja yang dilakukan oleh CGI, seperti mengumpulkan data dari form, menghasilkan isi halaman web dinamis, dan kemampuan mengirim serta menerima cookies, bahkan lebih daripada kemampuan CGI.

PHP tidak terbatas pada hasil keluaran HTML, namun PHP juga memiliki kemampuan untuk mengolah gambar, file PDF, dan movie flash. PHP juga dapat menghasilkan teks seperti XHTML dan *file* XML lainya. Salah satu fitur yang dapat diandalkan oleh PHP adalah dukungan terhadap data*base*, salah satunya adalah MySQL.

PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP.Apapun di luar pembatas tidak diproses oleh PHP (meskipun teks PHP ini masih mengendalikan struktur yang dijelaskan dalam kode PHP. Pembatas yang paling umum adalah "<?php" untuk membuka dan "?>" Untuk menutup kode PHP.Tujuan dari pembatas ini adalah untuk memisahkan kode PHP dari kode di luar PHP, seperti HTML,Javascript.

Variabel diawali dengan simbol dolar $. Pada versi php PHP 5 diperkenalkan jenis isyarat yang memungkinkan fungsi untuk memaksa mereka menjadi parameter objek dari class tertentu, array, atau fungsi. Namun, jenis petunjuk tidak dapat digunakan dengan jenis skalar seperti angka atau string.Contoh variabel dapat ditulis sebagai $nama_variabel.

Penulisan fungsi, penamaan kelas, nama variabel adalah peka akan huruf besar (Kapital) dan huruf kecil. Kedua kutip ganda "" dari string memberikan kemampuan untuk interpolasi nilai variabel ke dalam string PHP. PHP menerjemahkan baris sebagai spasi, dan pernyataan harus diakhiri dengan titik koma ;.

PHP memiliki 3 jenis sintaks sebagai komentar pada kode yaitu tanda blok / * * / , komentar 2 baris // Serta tanda pagar # digunakan untuk komentar satu baris. Komentar bertujuan untuk meninggalkan catatan pada kode PHP dan tidak akan diterjemahkan ke program.

Ratusan fungsi yang disediakan oleh PHP serta ribuan lainnya yang tersedia melalui berbagai ekstensi tambahan.fungsi-fungsi ini didokumentasikan dalam dokumentasi PHP. Namun, dalam berbagai tingkat pengembangan, kini memiliki berbagai konvensi penamaan. Sintaks fungsi adalah seperti di bawah ini:

```
functiontampilkan($data="")// Mendefenisikan fungsi, "tampilkan"
adalah nama sebuah fungsi
//Diapit oleh tanda kurung kurawal
if($data)return$data;elsereturn'Tidak ada data';// Melakukan
proses pengolahan data, contohnya melalui kondisi
echotampilkan("isi halaman")// Menjalankan fungsi
```

Contoh sebuah halaman web yang ditulis menggunakan Bahasa Pemograman PHP adalah sebagai berikut:

```
<?php
echo"Halo dunia";
?>
```

Beberapa keunggulan bahasa pemrograman PHP diantaranya:

- Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- *Web Server* yang mendukung PHP dapat ditemukan di mana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.

- Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
- Dalam sisi pemahamanan, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
- PHP adalah bahasa *open-source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system (Priyono, 2016).

2.2.4 Visual Studio Code

Visual Studio Code adalah editor *source code* yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk *debugging*, *GIT Control*yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, *shortcut keyboard*, dan preferensi.Visual Studio Code gratis dan *open-source*, meskipun unduhan resmi berada di bawah lisensi *proprietary*.

Kode Visual Studio didasarkan pada *Elektron*, kerangka kerja yang digunakan untuk menyebarkan aplikasi Node.js untuk desktop yang berjalan pada Blink*layout*. Meskipun menggunakan kerangka *Elektron*, Visual Studio Code tidak menggunakan *Atom* dan menggunakan komponen editor yang sama (diberi kode nama "*Monaco*") yang digunakan dalam Visual Studio Team Services yang sebelumnya disebut Visual Studio Online (Lardinois, 2015) .

2.2.5Kriptografi

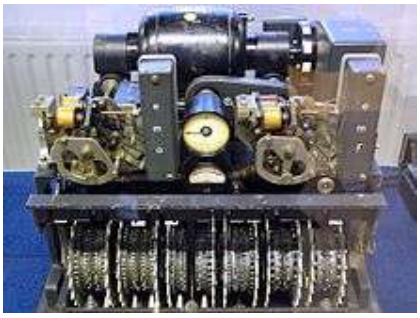
Kriptografi atau sandisastra merupakan keahlian dan ilmu dari cara-cara untuk komunikasi aman pada kehadirannya dipihak ketiga.Secara umum, kriptografi ialah mengenai mengkonstruksi dan menganalisis protokol komunikasi yang dapat memblokir lawan, berbagai aspek dalam keamanan informasi seperti data rahasia, integritas data, autentikasi, dan non-repudansi merupakan pusat dari kriptografi modern.Kriptografi modern terjadi karena terdapat titik temu antara

disiplin ilmu matematika, ilmu komputer, dan teknik elektro.Aplikasi dari kriptografi termasuk ATM, *password* komputer, dan *E-commerce*.

Kriptografi sebelum merupakan sinonim dari "enkripsi", konversi dari kalimat-kalimat yang dapat dibaca menjadi kelihatan tidak masuk akal. Pembuat dari pesan enkripsi membagi teknik pemecahan sandi yang dibutuhkan untuk mengembalikan informasi asli jika hanya dengan penerima yang diinginkan, sehingga dapat mencegah orang yang tidak diinginkan melakukan hal yang sama. Sejak Perang Dunia I dan kedatangan komputer, metode yang digunakan untuk mengelola kriptologi telah meningkat secara komplek dan pengaplikasiannya telah tersebar luar.

Kriptografi modern sangat didasari pada teori matematis dan aplikasi komputer; algoritma kriptografi didesain pada asumsi ketahanan komputasional, membuat algoritma ini sangat sulit dipecahkan oleh musuh.Secara teoretis, sangat sulit memecahkan sistem kriptografi, namun tidak layak melakukannya dengan cara-cara praktis.Skema ini oleh karena itu disebut sangat aman secara komputasional, kemajuan teoretis dapat meningkatkan algoritma faktorisasi *integer*, dan meningkatkan teknologi komputasi yang membutuhkan solusi ini untuk diadaptasi terus-menerus.Terdapat skema keamanan informasi yang benar-benar tidak boleh dapat ditembuh bahkan dengan komputasi yang tak terbatas namun skema ini sangat sulit diimplementasikan.

Teknologi yang berhubungan dengan kriptologi memiliki banyak masalah legal. Di Inggris, penambahan Regulasi Penyelidikan Aksi Wewenang membutuhkan kriminal yang tertuduh harus menyerahkan kunci dekripsinya jika diminta oleh penegah hukum. Jika tidak pengguna akan menghadapi hukum pidana. *Electronic Frontier Foundation* (EFF) terlibat dalam sebuah kasus di Amerika Serikat yang mempertanyakan jika seorang tersangka harus untuk menyerahkan kunci dekripsi mereka kepada pengak hukum merupakan inkonstitusionil.EFF memperdebatkan bahwa regulasi ini merupakan pelanggaran hak untuk tidak dipaksa mencurigai dirinya sendiri, seperti dalam Amendemen Kelima Konsitusi Amerika.Contoh mesin kriptografi dapat dilihat pada Gambar 2.1 (Becket, 1988).



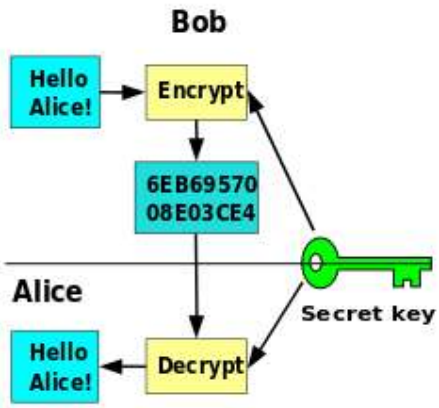
Gambar 2.1 Mesin Kriptografi

2.2.5.1 Kriptografi Kunci-Simetris

Kriptografi kunci-simetris merujuk pada metode enskripsi di mana kedua pengirim dan penerima membagi kunci yang sama (atau, walaupun kuncinya tidak mirip, namun dapat berhubungan dengan cara komputasi sederhana). Hal ini menjadi satu-satunya jenis enkripsi yang ketahui publik hingga Juni 1976.

Standar Enkripsi Data (SED) dan Standar Enkripsi Lanjutan (SEL) merupakan desain *chiper*blok yang telah ditunjuk sebagai standar kriptografi oleh pemerintah Amerika (walaupun penunjukan SED pada akhirnya ditarik setelah SEL diadopsi). Walaupun penarikannya sebagai standar resmi, SED (masih menjadi varian yang masih terbukti dan lebih aman) masih cukup terkenal; Hal ini digunakan oleh banyak penerapan dari enskripsi ATM hingga keamanan e-mail dan akses remote aman. Banyak *chiper*blok lainnya telah didesain dan dirilis, dengan kualitas yang bervariasi.

Beberapa c*hiper*, yang berbeda dengan tipe 'blok', membuat berkas panjang material kunci yang panjang, di mana dikombinasikan dengan bit-bit teks atau karakter-karakter, sedikit mirip dengan one-time pad.Pada *chiper* aliran, aliran keluarannya diciptakan berdasarkan keadaan internal yang tersembunyi yang berubah *saat chiper* bekerja.Keadaan internal mulanya diatur menggunakan bahan kunci rahasia.RC4 sangat luas digunakan sebagai *chiper* aliran.*Chiper* blok dapat digunakan sebagai *chiper* aliran.Diagram kriptografi kunci simetris dapat dilihat pada Gambar2.2.



Gambar 2.2 Kriptografi Kunci Simetris

Fungsi *hash* Kriptografi merupakan algoritma kriptografi tipe ke-tiga. Fungsi ini mengambil segala panjang pesan sebagai input, dan panjang keluaran hash yang pendek dan tetap, yang dapat digunakan sebagai (sebagai contoh) tanda tangan digital. Untuk memiliki fungsi hash yang baik, penyerang tidak dapat mencari dua pesan yang dapat menghasilkan hash yang sama. MD4 merupakan fungsi hash pangjang yang sekarang telah dapat dipecahkan; MD5, varian yang lebih kuat dari MD4, sudah luas digunakan namun dapat dipecahkan saat beroperasi.

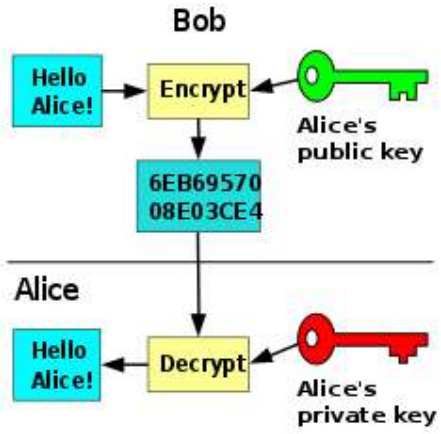
Agensi keamanan nasional Amerika mengembangkan serial Algoritma Hash Aman seperti fungsi hash MD5: SHA-0 ialah algoritma cacat yang kemudian ditarik; SHA-1 digunakan secara luas dan lebih aman dari MD5, namun kriptanalisis telah menemukan serangan padanya; keluarga SHA-2 meningkatkan performa SHA-1, namun belum secara luas digunakan; dan kewenangan Amerika mengatakan hal ini cukup bijaksana dari sudut pandang keamanan untuk mengembangkan standar baru *toolkit* algoritma hash NIST secara keseluruhan untuk peningkatan kekuatan secara signifikan. Sehingga, pada tahun 2012, standar nasional Amerika memilih SHA-3 sebagai standar desain *hash* yang baru.

Message authentication code (MAC) hampir mirip dengan fungsi hash kriptografi, kecuali terdapat kunci rahasia yang dapat digunakan untuk membuktikan nilai hash melalui serangkaian resep, kerumitan tambahan yang

melindungi skema serangan algoritma penyingkat sederhana, dan dianggap cukup menguntungkan (Rivest, 1990).

2.2.5.2 Kriptografi Kunci-Publik

Kriptosistem kunci-simetris menggunakan kunci yang sama untuk enkripsi dan dekripsi sebuah pesan, walaupun pesan atau kelompok pesan dapat memiliki kunci yang berbeda dari yang lain. Kerugian yang paling signifikan dari chiper simetris ialah kebutuhan manajerial kunci untuk menggunakannya secara aman.Setiap sepasang pihak komunikasi yang berbeda harus, idealnya, membagi kunci yang berbeda, dan juga membagi *chipertext* yang berbeda juga.jumlah kunci yang dibutuhkan meningkat dua kali lipat dari jumlah anggota jaringan, yang sangat cepat membutuhkan skema manajemen kunci komplek untuk menjaga semuanya tetap konsisten dan rahasia. Diagram kriptografi kunci public dapat dilihat pada Gambar 2.3 (Rivest, 1990).



Gambar 2.3 Kriptografi Kunci Publik

Kesulitan dari menciptakan kunci rahasia yang aman di antara dua pihak yang saling berkomunikasi, ialah, ketika belum adanya jaringan aman di antara keduanya, juga kehadiran chicken-and-egg problem yang dianggap menjadi tantangan praktikal untuk pengguna kriptografi di dunia nyata.

Kriptografi kunci-publik dapat juga digunakan untuk mengimplementasikan skema tanda tangan digital. Tanda tangan digital

berhubungan dengan tanda tangan pada umumnya; mereka memiliki karateristik yang sama dimana mudah bagi pengguna untuk membuatnya, namun sangat sulit bagi orang lain untuk memalsukannya. Tanda tangan digital dapat juga secara permanen mengikat pada konten pesan yang sedang ditanda tangani; mereka lalu tidak dapat 'dipindahkan' dari satu dokumen ke dokumen yang lain, dan setiap usaha akan dapat terdeteksi. Pada skema tanda tangan digital, terdapat dua algoritma: satu untuk menandatangani, di mana kunci rahasia digunakan untuk memproses pesan (atau hash dari pesan, atau keduanya), dan satu untuk verivikasi, di mana kunci publik yang sesuai digunakan dengan pesan untuk memeriksa validitas tanda tangan. RSA dan DSA merupakan dua skema tanda tangan digital yang paling terkenal.Tanda tangan digital merupakan pusat dari operasi infrastruktur kunci publik dan banyak skema keamanan jaringan lainnya (seperti Transport Layer Security, VPN, dan lain-lain).

Algoritma kunci publik paling sering didasari pada teori masalah kompleksitas komputasional, sering dengan teori bilangan.Sebagai contoh, kekuatan RSA berhubungan dengan masalah faktorisasi integer, sedangkan Diffie-Hellman dan DSA berhubungan dengan masalah logaritma diskrit.Baru-baru saja, kriptografi kurva eliptis telah ditemukan, sistem di mana keamanan yang didasari pada masalah teoretis bilangan yang melibatkan kurva eliptis.

Dikarenakan kesulitan masalah pokok, kebanyakan algoritma kunci-publik melibatkan operasi seperti eksponensial dan perkalian aritmetika modular, di mana teknik ini secara komputasional lebih mahal ketimbang teknik yang digunakan pada banyak chiper blok, khususnya dengan ukuran kunci yang dibutuhkan. Hasilnya, kriptosistem kunci-publik seringnya merupakan kriptosistem hybrid, yang merupakan algoritma enkripsi kunci-simetris berkualitas tinggi digunakan untuk pesan itu sendiri, sedang kunci simetris yang relevan dikirimkan dengan pesan, namun dienkripsikan menggunakan algoritma kunci publik. Hampir sama, skema tanda tangan hybrid sering digunakan, di mana fungsi hash kriptogfari dihitung secara komputer, dan hanya hash hasil yang ditanda tangani secara digital (Rivest, 1990).

2.2.6 Enkripsi dan Dekripsi

Di bidang kriptografi, enkripsi adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus. Dikarenakan enkripsi telah digunakan untuk mengamankan komunikasi di berbagai negara, hanya organisasi-organisasi tertentu dan individu yang memiliki kepentingan yang sangat mendesak akan kerahasiaan yang menggunakan enkripsi. Dipertengahan tahun 1970-an, enkripsi kuat dimanfaatkan untuk pengamanan oleh sekretariat agen pemerintah Amerika Serikat pada domain publik, dan saat ini enkripsi telah digunakan pada sistem secara luas, seperti *internet e-commerce*, jaringan Telepon bergerak dan ATM pada bank.

Enkripsi dapat digunakan untuk tujuan keamanan, tetapi teknik lain masih diperlukan untuk membuat komunikasi yang aman, terutama untuk memastikan integritas dan autentikasi dari sebuah pesan. Contohnya, *Message Authentication Code* (MAC) atau *digital signature*. Penggunaan yang lain yaitu untuk melindungi dari analisis jaringan komputer.

2.2.6.1Cipher

Sebuah *cipher* adalah sebuah algoritme untuk menampilkan enkripsi dan kebalikannya dekripsi, serangkaian langkah yang terdefinisi yang diikuti sebagai prosedur. Alternatif lain ialah *encipherment*. Informasi yang asli disebut sebagai *plaintext,* dan bentuk yang sudah dienkripsi disebut sebagai *chiphertext.*Pesan *chipertext* berisi seluruh informasi dari pesan *plaintext*, tetapi tidak dalam format yang didapat dibaca manusia ataupun komputer tanpa menggunakan mekasnisme yang tepat untuk melakukan dekripsi.

Cipher pada biasanya memiliki parameter dari sebagian dari informasi utama, disebut sebagai kunci. Prosedur enkripsi sangat bervariasi tergantung pada kunci yang akan mengubah rincian dari operasi algoritme. Tanpa menggunakan kunci, *chiper* tidak dapat digunakan untuk dienkirpsi ataupun didekripsi.

2.2.6.2 Cipher dan Code

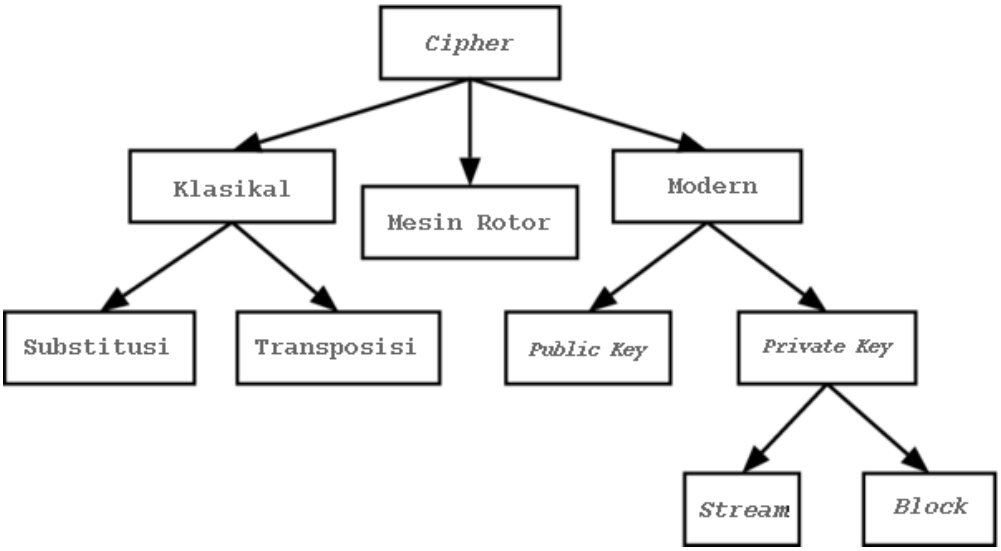
Pada penggunaan non teknis, sebuah secret code merupakan hal yang sama dengan *cipher*. Berdasar pada diskusi secara teknis, bagaimanapun juga, *code* dan

cipher dijelaskan dengan dua konsep. *Code* bekerja pada tingkat pemahaman, yaitu, kata atau frasa diubah menjadi sesuatu yang lain. *Cipher*, dilain pihak, bekerja pada tingkat yang lebih rendah, yaitu, pada tingkat masing-masing huruf, sekelompok huruf, pada skema yang modern, pada tiap-tiap bit. Beberapa sistem menggunakan baik *code* dan *cipher* dalam sistem yang sama, menggunakan superencipherment untuk meningkatkan keamanan.

Menurut sejarahnya, kriptografi dipisah menjadi dikotomi *code* dan *cipher*, dan penggunaan *code* memiliki terminologi sendiri, hal yang sama pun juga terjadi pada cipher: "*encoding, codetext, decoding*" dan lain sebagainya. Bagaimanapun juga, *code* memiliki berbagai macam cara untuk dikembalikan, termasuk kerapuhan terhadap kriptoanalisis dan kesulitan untuk mengatur daftar kode yang susah. Oleh karena itu, *code* tidak lagi digunakan pada kriptografi modern, dan *cipher* menjadi teknik yang lebih dominan.

2.2.6.3 Tipe Cipher

Ada banyak sekali variasi pada tipe enkripsi yang berbeda.Algoritme yang digunakan pada awal sejarah kriptografi sudah sangat berbeda dengan metode modern, dan *cipher* modern dan diklasifikasikan berdasar pada bagaimana *cipher* tersebut beroperasi dan *cipher* tersebut menggunakan sebuah atau dua buah kunci (Goldreich, 2001).



Gambar 2.4 Klasifikasi Cipher

Sejarah Cipher pena dan kertas pada waktu lampau sering disebut sebagai *cipher* klasik. *Cipher* klasik termasuk juga *cipher* pengganti dan *cipher* transposisi. Pada awal abad 20, mesin-mesin yang lebih mutakhir digunakan untuk kepentingan enkripsi, mesin rotor, merupkan skema awal yang lebih kompleks.

Metode enkripsi dibagi menjadi algoritma kunci simetris dan algoritma*asymmetric key*.pada algoritma*symmetric key* (misalkan, DES dan AES), pengirim dan penerima harus memiliki kunci yang digunakan bersama dan dijaga kerahasiaanya. Pengirim menggunkan kunci ini untuk enkripsi dan penerima menggunakan kunci yang sama untuk dekripsi. Pada algoritma kunci asimetris (misalkan, RSA), terdapat dua kunci terpisah, sebuah *public key* diterbitkan dan membolehkan siapapun pengirimnya untuk melakukan enkripsi, sedangkan sebuah private key dijaga kerahasiannya oleh penerima dan digunakan untuk melakukan dekripsi.

Cipher kunci simetris dapat dibedakan dalam dua tipe, tergantung pada bagaimana *cipher* tersebut bekerja pada blok simbol pada ukuran yang tetap *(block ciphers*), atau pada aliran simbol terus-menerus *(stream ciphers*).

Dekripsi adalah proses untuk mengembalikan sebuah nilai menjadi nilai awal sebelum dienkripsi (Goldreich, 2001).

2.2.7 Encoding dan Decoding

Dalam komunikasi dan pemrosesan informasi, pengkodean atau penyandian (*encoding*) adalah proses konversi informasi dari suatu sumber (objek) menjadi data, yang selanjutnya dikirimkan ke penerima atau pengamat, seperti pada sistem pemrosesan data. Pengawakodean atau pengawasandian (*decoding)* adalah proses kebalikannya, yaitu konversi data yang telah dikirimkan oleh sumber menjadi informasi yang dimengerti oleh penerima. Sedangkan decoding adalah proses pengembalian informasi dari hasil encoding menjadi bentuk aslinya (Danesi, 2013).

2.2.8 Algoritma *Caesar*

Dalam kriptografi, sandi *Caesar*, atau sandi geser, kode *Caesar* atau Geseran *Caesar* adalah salah satu teknik enkripsi paling sederhana dan paling terkenal. Sandi ini termasuk sandi substitusi dimana setiap huruf pada teks terang (plaintext) digantikan oleh huruf lain yang memiliki selisih posisi tertentu dalam alfabet. Misalnya, jika menggunakan geseran 3, W akan menjadi Z, I menjadi L, dan K menjadi N sehingga teks terang "wiki" akan menjadi "ZLNL" pada teks tersandi. Nama *Caesar* diambil dari Julius *Caesar*, jenderal, konsul, dan diktator Romawi yang menggunakan sandi ini untuk berkomunikasi dengan para panglimanya.

Langkah enkripsi oleh sandi *Caesar* sering dijadikan bagian dari penyandian yang lebih rumit, seperti sandi Vigenère, dan masih memiliki aplikasi modern pada sistem ROT13.Pada saat ini, seperti halnya sandi substitusi alfabet tunggal lainnya, sandi *Caesar* dapat dengan mudah dipecahkan dan praktis tidak memberikan kerahasiaan bagi pemakainya.

Cara kerja sandi ini dapat diilustrasikan dengan membariskan dua set alfabet; alfabet sandi disusun dengan cara menggeser alfabet biasa ke kanan atau ke kiri dengan angka tertentu (angka ini disebut kunci). Misalnya sandi *Caesar* dengan kunci 3, adalah sebagai berikut:

- Alfabet Biasa : ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Alfabet Sandi : DEFGHIJKLMNOPQRSTUVWXYZABC

Untuk menyandikan sebuah pesan, cukup mencari setiap huruf yang hendak disandikan di alfabet biasa, lalu tuliskan huruf yang sesuai pada alfabet sandi. Untuk memecahkan sandi tersebut gunakan cara sebaliknya. Contoh penyandian sebuah pesan adalah sebagai berikut.

- teks terang: kirim pasukan ke sayap kiri
- teks tersandi: NLULP SDVXNDQ NH VDBDS NLUL

Proses penyandian (enkripsi) dapat secara matematis menggunakan operasi modulus dengan mengubah huruf-huruf menjadi angka, A = 0, B = 1,..., Z

= 25. Sandi ($\mathbf{E_n}$) dari "huruf" ***x*** dengan geseran n secara matematis bisa dilihat dalam persamaan 2.1 (Halsall, 1998).

- $E_n\ (x) = (x+n)$ **mod 26…… 2.1**

Sedangkan pada proses pemecahan kode (dekripsi), hasil dekripsi (D_n) dapat dilihat dalam persamaan 2.2.

- $D_n\ (x) = (x-n)$ **mod 26……..2.2**

Setiap huruf yang sama digantikan oleh huruf yang sama di sepanjang pesan, sehingga sandi *Caesar* digolongkan kepada, *substitusi monoalfabetik*, yang berlawanan dengan *substitusi polialfabetik.*

2.2.9Algoritma *Base*64

*Base*64 adalah kelompok skema pengkodean *biner*-ke-teks yang serupa yang mewakili data *biner* dalam format string ASCII dengan menerjemahkannya ke dalam representasi radix-64 (64 karakter unik).Istilah *Base*64 berasal dari encoding transfer konten MIME tertentu. Setiap digit *base*64 mewakili 6 bit data (IETF, 2006).

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Gambar 2.5 Karakter Base64

Dalam teknik *base*64 jika ada sejumlah string yang akan disandikan ke dalam *base*64 maka caranya adalah :

1. Pecah *string* bytes tersebut ke per-3 bytes.

2. Gabungkan 3 bytes menjadi 24 *bit*. 1 bytes = 8 *bit* sehingga 3 x 8 = 24 *bit*.
3. Lalu 24 *bit* tersebut dipecah menjadi 6 bit sehingga menjadi 4 pecahan.
4. Masing pecahan diubah kedalam nilai desimal, dimana maksimal nilai 6 bit adalah 63 (2^6).
5. Jadikan nilai tersebut menjadi indeks untuk memilih karakter penyusun dari *base*64 dan maksimal adalah 63 atau indeks ke-63.

Jika dalam proses encoding terdapat sisa pembagi maka tambahkan sebagai penggenap sisa tersebut ”=”.

Huruf	R								i								o							
ASCII	82								105								111							
Bit	0	1	0	1	0	0	1	0	0	1	1	0	1	0	0	1	0	1	1	0	1	1	1	1
Index	20				38				37				47											
Base64 Encoded	U				m				l				v											

Gambar 2.6 Teknik *base*64