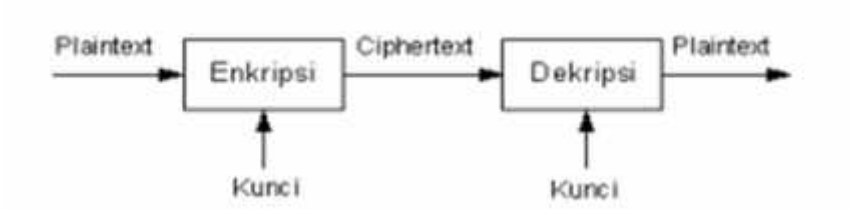
BAB II

TINJAUAN PUSTAKA

II.1. Kriptografi

Kata kriptografi (*Cryptography*) berasal dari bahasa Yunani yaitu dari kata Cryptos yang artinya tersembunyi dan Graphein yang artinya menulis. Kriptografi dapat diartikan sebagai suatu ilmu ataupun seni yang mempelajari bagaimana sebuah data dikonversi ke bentuk tertentu yang sulit untuk dimengerti (Doni, 2012). Kriptografi bertujuan untuk menjaga kerahasiaan informasi atau data supaya tidak dapat diketahui oleh pihak yang tidak berhak (*unauthorized person*).

Suatu data yang tidak disandikan disebut *plaintext* atau *cleartext*. Sedangkat data yang telah tersandikan disebut *ciphertext*. Proses yang dilakukan untuk mengubah *plaintext* menjadi *ciphertext* disebut enkripsi (*encryption*) atau *encipherment*. Sedangkan proses untuk mengubah *ciphertext* kembali ke *plaintext* disebut dekripsi (*decryption*) atau *decipherment*. Dalam kriptografi diperlukan parameter yang digunakan untuk proses konversi data yaitu suatu set kunci. Enkripsi dan dekripsi data dikontrol oleh sebuah kunci atau beberapa kunci. Secara sederhana istilah – istilah diatas dapat digambarkan sebagai berikut:



Gambar II.1 Proses Enkripsi/Dekripsi Sederhana

II.2. Aspek-Aspek Kriptografi

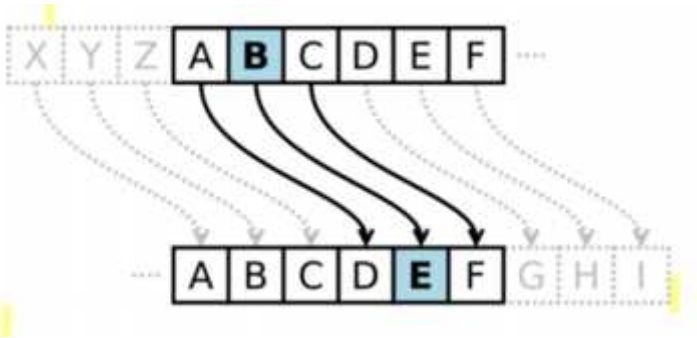
Kriptografi tidak hanya memberikan kerahasiaan dalam telekomunikasi, namun, juga memiliki sejumlah aspek (Doni, 2012), yaitu:

1. Kerahasiaan data, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
2. Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
3. Autentifikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentifikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
4. Non-repuidasi, atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman suatu informasi oleh yang mengirimkan.

II.3. Algoritma Caesar Cipher

Dalam kriptografi Caesar cipher atau sandi Caesar, kode Caesar atau sandi geser adalah salah satu teknik enkripsi paling sederhana dan paling terkenal. Sandi ini termasuk sandi subtitusi dimana setiap huruf pada teks terang (*plaintext*) digantikan oleh huruf lain yang memiliki selisih posisi tertentu dalam alfabet. Misalnya, jika menggunakan geseran 3, B akan menjadi E, U menjadi X, dan K menjadi N sehingga plaintext “BUKU” akan menjadi “EXNX” 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 Vigenere. Pada saat ini, seperti halnya sandi substitusi alfabet tunggal lainnya, sandi Caesar dapat dengan mudah dipecahkan dan praktis tidak memberikan kerahasiaan bagi pemakainya. (Doni, 2012)



Gambar II.2 Sandi Caesar Dengan Geseran Tiga

II.4. Cara Kerja Sandi Caesar

Cara kerja sandi dapat diilustrasikan dengan membariskan dua set 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:

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

Alfabet Sandi:

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

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:

Alfabet Biasa:

DONI

Alfabet Sandi:

GRQL

Proses penyandian (enkripsi) dapat secara matematis menggunakan operasi modulus dengan mengubah huruf-huruf menjadi angka, A = 0, B = 1, ..., Z = 25. Sandi (E_n) dari "huruf" *x* dengan geseran *n* secara matematis dituliskan dengan:

$$\mathbf{E_n(x) = (x + n) \bmod 26}$$

Sedangkan, pada proses pemecahan kode (dekripsi), hasil dekripsi (Dn) adalah:

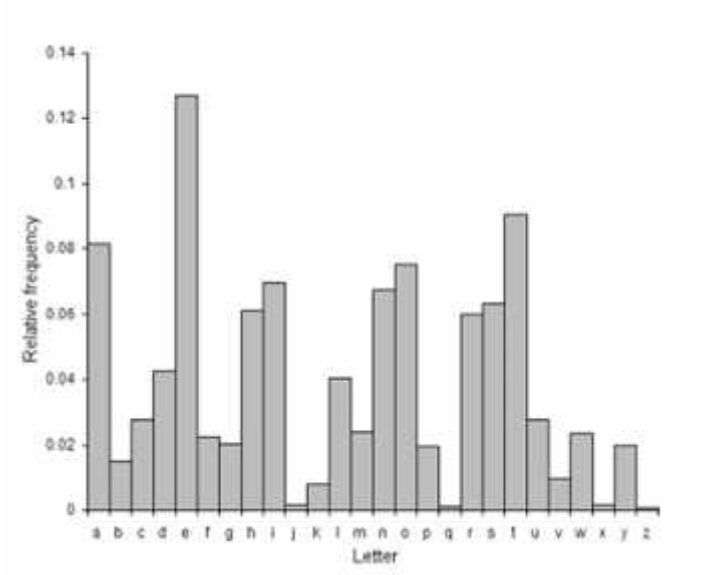
$$D_n(x) = (x - n) \bmod 26$$

Setiap huruf yang sama digantikan oleh huruf yang sama disepanjang pesan, sehingga sandi Caesar digolongkan kepada *substitusi monoalfabetik* yang berlawanan dengan *substitusi poliaflabetik* (Doni, 2012).

II.5. Dekripsi Sandi Caesar

Proses membaca teks tersandi menjadi plaintext disebut dekripsi. Sandi Caesar dapat dipecahkan bahkan jika seseorang hanya memiliki teks tersandi tanpa mengetahui nilai geserannya, ataupun bahwa sandi Caesar telah digunakan.

Jika pihak pemecah sandi hanya mengetahui bahwa digunakan substitusi monoalfabetik dalam suatu sandi, sandi tersebut dipecahkan dengan cara analisis frekuensi. Setiap bahasa memiliki huruf yang sering digunakan atau jarang digunakan. Misalnya huruf *a* sering sekali digunakan dalam bahasa Indonesia, dan *q* atau *x* jarang sekali muncul. Setiap bahasa memiliki pola frekuensi tertentu, yang menunjukkan frekuensi relatif dari digunakannya huruf-huruf dalam bahasa tersebut. Pola frekuensi huruf dalam bahasa Inggris ditunjukkan dalam gambar sebagai berikut:



Gambar II.3 Frekuensi Kemunculan Huruf Dalam Bahasa Inggris

Cara kedua yang lebih mudah, dapat dilakukan jika sang pemecah sandi mengetahui bahwa pengirim sandi menggunakan sandi Caesar. Sandi tersebut akan dipecahkan dengan menggunakan *brute force attack* adalah sebuah teknik serangan terhadap sebuah sistem keamanan komputer yang menggunakan percobaan terhadap semua kunci yang mungkin, yaitu mencoba ke-26 kemungkinan geseran yang digunakan. Biasanya hanya satu dari ke-26 kemungkinan ini yang dapat dibaca. Misalkan suatu teks tersandi “EXXEGOEXSRGI”.

Deskripsi pergeseran	Kandidat plaintext
0	exxegoexsrgi
1	dwwdfndwrqfh
2	cvvcemcvqpeg
3	buubdlbupodf
4	attackatonce
5	zsszbjzsnmbd
6	yrryaiyrmlac
	.....
23	haahjrhavujl
24	gzzgiqgzutik
25	fyyfhpfytshj

Gambar II.4 Tabel Brute Force Attack

Pada tabel diatas ditunjukkan hasil percobaan yang dilakukan, dan hanya satu kali yang dapat dibaca, yaitu *attackatonce*. Hal ini berarti pesan yang disandikan adalah pesan berbahasa Inggris “attack at once”, yang berarti “serang sekarang juga”.

Dengan kemajuan komputer dan teknologi informasi, kedua cara diatas dapat dijalankan dengan mudah dan cepat, sehingga saat ini sandi Caesar sama sekali tidak berguna untuk menyembunyikan atau menyandikan dokumen-dokumen atau perintah-perintah penting dan rahasia. (Doni, 2012)

II.6. SMS

Telepon seluler merupakan salah satu hasil dari perkembangan teknologi komunikasi. Terdapat beberapa layanan komunikasi yang dapat digunakan pada telepon seluler, diantaranya: layanan telepon, *video call*, SMS, dan MMS. *Short Message Service* (SMS) atau pesan singkat merupakan fungsi komunikasi yang banyak digunakan oleh pengguna telepon seluler. Salah satu alasan layanan SMS menjadi salah satu layanan yang paling penting dan dibutuhkan dikarenakan SMS mudah digunakan dan biaya yang dikeluarkan untuk mengirim SMS relatif murah (Triyuswoyo, 2012).

II.7. UML

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan iteratif, yaitu: identifikasi kelas-kelas dan obyek-

obyek, identifikasi semantik dari hubungan obyek dan kelas tersebut, perincian interface dan implementasi. Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur dan pemodelan entity-relationship. Tahapan utama dalam metodologi ini adalah analisis, design sistem, design obyek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep Object Oriented. Metode OOSE dari Jacobson lebih memberi penekanan pada use case. OOSE memiliki tiga tahapan yaitu membuat model requirement dan analisis, design dan implementasi, dan model pengujian (test model). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Dengan UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif daripada metode lainnya (Munawar, 2005).

II.8. Use Case

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan

antara pengguna dan sistem disebut scenario. Setiap scenario mendekripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan use case adalah serangkaian scenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

Dalam pembicaraan tentang use case, penggunanya biasanya disebut dengan actor. Actor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem.

Model use case adalah bagian dari model requirement. Termasuk disini adalah problem domain object model dan penjelasan tentang user interface. Use case memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif user (Munawar, 2005).

II.9. Java

Java adalah sebuah bahasa pemrograman yang populer dikalangan para akademisi dan praktisi komputer. Java pertama kali dikembangkan untuk memenuhi kebutuhan akan sebuah bahasa komputer yang ditulis satu kali dan dapat dijalankan dibanyak sistem komputer berbeda tanpa perubahan kode berarti. Pada umumnya, para pakar pemrograman berpendapat bahwa bahasa Java memiliki konsep yang konsisten dengan teori pemrograman objek dan aman untuk digunakan.

Java sampai saat ini masih merupakan bahasa pemrograman yang masih sangat di minati dan banyak digunakan oleh para programmer dan software

developer untuk mengembangkan berbagai tipe aplikasi, mulai dari aplikasi *console*, aplikasi *desktop*, game, dan applet (aplikasi yang berjalan di lingkungan *web browser*), sampai ke aplikasi-aplikasi yang berskala *enterprise*. Untuk memenuhi kebutuhan tipe aplikasi yang beragam tersebut, Java dikategorikan menadi tiga edisi, yaitu: J2SE (*Java 2 Platform Standart Edition*) untuk membuat aplikasi-aplikasi desktop dan applet, J2EE (*Java 2 Platform Enterprise Edition*) untuk membuat aplikasi-aplikasi *multitier* berskala *enterprise*, dan J2ME (*Java 2 Micro Edition*) untuk membuat aplikasi-aplikasi yang dapat dijalankan dilingkungan perangkat-perangkat mikro seperti handphone, PDA, dan Smartphone (Wardhani, 2013).

II.10. Android Studio

Android Studio adalah sebuah lingkungan pengembangan terpadu (IDE) untuk mengembangkan pada platform Android. Hal itu disampaikan pada tanggal 16 Mei 2013 di Google I/O konferensi dengan Product Manager Google, Katherine Chou. Android Studio tersedia secara bebas di bawah Lisensi Apache 2.0. Android Studio berada di awal tahap preview akses mulai dari versi 0.1 Mei 2013, kemudian memasuki tahap beta mulai dari versi 0.8 yang dirilis pada bulan Juni 2014. Yang pertama membangun stabil dirilis pada bulan Desember 2014, mulai dari versi 1.0. Berdasarkan software IDEA JetBrains IntelliJ, Android Studio dirancang khusus untuk pengembangan Android. Ini tersedia untuk di-download pada Windows, Mac OSX dan Linux, dan mengganti Eclipse

Pengembangan Android Tools (ADT) sebagai IDE utama Google untuk pengembangan aplikasi Android asli.