

## BAB II TINJAUAN PUSTAKA

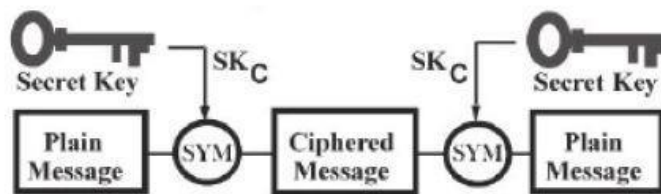
### 2.1 Kriptografi

Kriptografi merupakan kata yang berasal dari bahasa Yunani adalah *Crypto* yang berarti rahasia dan *Grapho* yang berarti menulis. Kriptografi secara umum merupakan ilmu dan seni untuk menjaga keamanan pesan (*Cryptography is the art and science of keeping messages secure*) [7]. Sebagai pembanding, kriptografi merupakan ilmu yang mempelajari teknik-teknik matematika di mana berhubungan dari segi aspek keamanan informasi seperti integritas data, kerahasiaan serta otentikasi. Dalam definisi di atas, kata “seni” berasal dari fakta sejarah bahwa saat masa-masa awal sejarah munculnya kriptografi, setiap orang mungkin mempunyai cara yang unik agar dapat merahasiakan pesan. Berbagai cara unik tersebut mungkin berbeda-beda pada setiap pengguna kriptografi sehingga setiap cara menulis pesan rahasia, pesan tersebut mempunyai nilai estetika tersendiri sehingga kriptografi berkembang menjadi sebuah seni dengan merahasiakan pesan [8].

Dari catatan sejarah yang pernah ada, kriptografi telah digunakan sejak 4000 tahun yang lalu oleh bangsa Mesir yang digunakan oleh raja-raja Mesir pada saat perang untuk menyampaikan pesan rahasia kepada panglima perangnya melalui kurir-kurirnya. Orang yang terlibat melakukan penyandian ini dikenal sebagai *Cryptographer*, sedangkan orang yang terlibat mendalami ilmu dan seni baik dalam membuka atau memecahkan suatu algoritma kriptografi tanpa harus mengetahui kuncinya dikenal sebagai kriptologis.

Kriptografi pada dasarnya memiliki dua cara, yaitu proses enkripsi dan proses dekripsi. Proses enkripsi merupakan proses penyandian terhadap pesan yang terbuka (*plaintext*) menjadi pesan yang rahasia (*ciphertext*). *Ciphertext* tersebut akan dikirimkan melalui saluran komunikasi yang terbuka. Pada saat *ciphertext* tersebut diterima oleh penerima, maka pesan rahasia tersebut diubah kembali menjadi pesan yang terbuka melalui proses dekripsi sehingga pesan

tadi dapat dibaca kembali oleh penerima. Secara umum, proses enkripsi dan dekripsi dapat digambarkan sebagai berikut [9]:



Gambar 2.1 Proses Enkripsi dan Dekripsi [10]

Pada sistem komputer, *plaintext* diberi lambang  $M$  di mana merupakan singkatan dari *message*. *Plaintext* ini dapat berupa tulisan, foto, maupun video yang berbentuk data biner. *Plaintext* inilah yang nantinya akan di enkripsi menjadi sebuah pesan rahasia atau *ciphertext* dan dilambangkan dengan  $C$  (*ciphertext*). Secara matematis, fungsi enkripsi dinotasikan dengan:  $E(M) = C$ . Pada fungsi dekripsi merupakan proses pembalikan dari *ciphertext* menjadi *plaintext* kembali di mana secara matematis dinotasikan sebagai berikut [10]:

$$D(C) = M; D(E(M)) = M$$

### 2.1.1 Algoritma Kriptografi

Jika dilihat dari sejarahnya, kata algoritma memiliki sejarah yang menarik. Kata tersebut muncul di dalam kamus Webster sampai pada akhir tahun 1957. Kata algoritma berasal dari bahasa Arab yaitu *algorism* yang mempunyai arti proses perhitungan. Algoritma berasal dari nama penulis buku Arab yang terkenal, yaitu Abu Ja'far Muhammad Ibnu Musa al-Khuwarizmi. Seiring waktu berjalan, kata *algorism* lambat laun berubah menjadi *algorithm*. Definisi terminologi algoritma yaitu urutan langkah-langkah logis untuk menyelesaikan masalah yang dibuat dan disusun secara sistematis. Algoritma kriptografi yaitu langkah-langkah logis bagaimana cara menyembunyikan pesan dari orang-orang yang tidak seharusnya menerima pesan tersebut.

Algoritma kriptografi yang baik itu bukanlah ditentukan oleh kerumitan dalam mengelola data atau pesan yang akan disampaikan. Tetapi, algoritma tersebut memiliki 4 persyaratan sebagai berikut [11]:

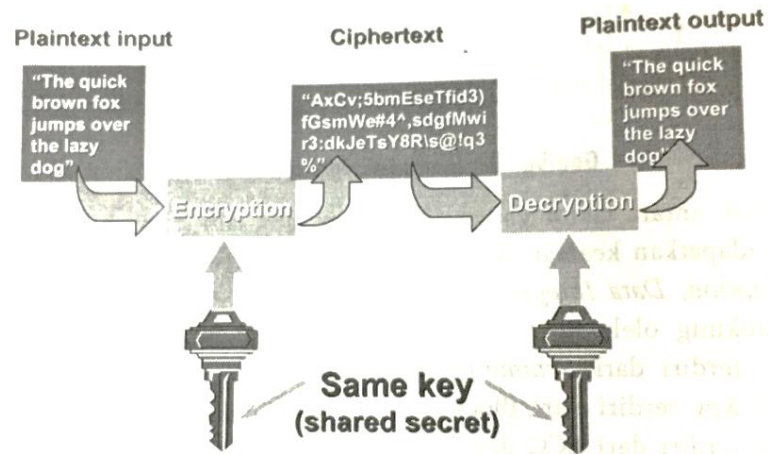
1. Kerahasiaan. Pesan (*plaintext*) hanya bisa dibaca antara dua pihak yang memiliki kewenangan.
2. Autentifikasi. Pengirim pesan harus dapat diidentifikasi dengan pasti dan penyusup harus dipastikan tidak bisa berpura-pura menjadi orang lain.
3. Integritas. Penerima pesan harus bisa memastikan bahwa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

### **2.1.2 Macam-Macam Algoritma Kriptografi**

Algoritma kriptografi dibagi menjadi tiga bagian berdasarkan kunci yang dipakainya yaitu [11]:

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

### 2.1.2.1 Algoritma Simetri

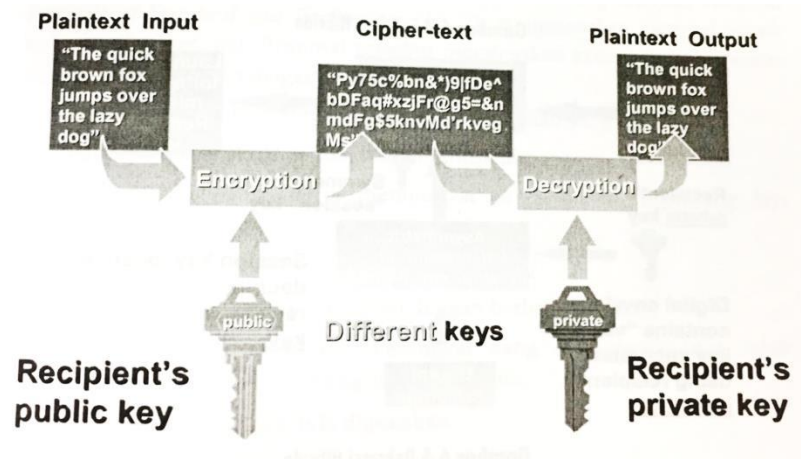


Gambar 2.2 Algoritma Simetris [11]

Algoritma ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsi. Algoritma ini sudah ada sejak lebih dari 4000 tahun yang lalu. Bila mengirim pesan dengan menggunakan algoritma ini, si penerima pesan harus diberitahu kunci dari pesan tersebut agar bisa 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 deskripsi terhadap pesan. Algoritma yang memakai kunci simetri di antaranya adalah [11]:

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

### 2.1.2.2 Algoritma Asimetri



Gambar 2.3 Algoritma Asimetris [11]

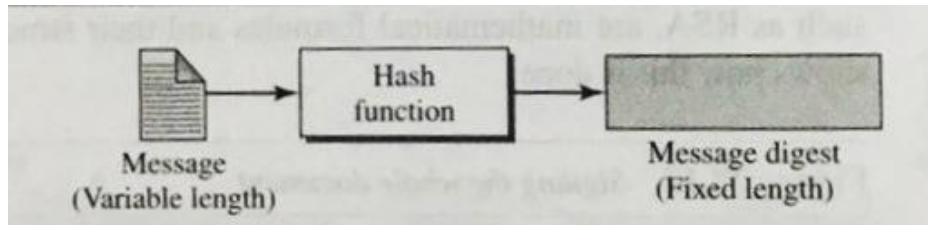
Algoritma asimetri sering juga disebut dengan algoritma kunci publik, dengan arti kata kunci yang digunakan untuk melakukan enkripsi dan deskripsi berbeda. Pada algoritma asimetri kunci terbagi menjadi dua bagian, yaitu [11]:

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

Algoritma yang memakai kunci publik di antaranya adalah [11]:

1. Digital Signature Algorithm (DSA),
2. RSA,
3. Diffie-Hellman (DH),
4. Elliptic Curve Cryptography (ECC),
5. Kriptografi Quantum, dan lain sebagainya.

### 2.1.2.3 Hash Function



Gambar 2.4 Hash Function [12]

*Hash Function* sering disebut dengan fungsi Hash satu arah (*one-way function*), *message digest*, *fingerprint*, fungsi kompresi dan *message authentication code* (MAC), merupakan suatu fungsi matematika yang mengambil masukan panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap. *Hash Function* 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 [12].

## 2.2 Metode Vigenere Cipher

*Vigenere* termasuk dalam kode abjad-majemuk (*polyalphabetic substitution cipher*). Dipublikasikan oleh diplomat (sekaligus seorang kriptologis) Perancis yang bernama Blaise de Vigenere pada abad ke 16, tahun 1586. Sebenarnya pada tahun 1553, Giovan Batista Belaso telah menggambarkannya untuk pertama kali seperti yang ditulis di dalam buku *La Cifra del Sig*. Setelah 200 tahun kemudian, algoritma tersebut baru dikenal luas dengan dinamakan kode *vigenere*. *Vigenere* merupakan pemicu bagi perang sipil di Amerika dan kode *Vigenere* juga digunakan oleh tentara konfederasi (*confederate army*) pada perang sipil Amerika (*American civil war*). Pada pertengahan abad ke 19, kode *vigenere* berhasil dipecahkan oleh *Babbage* dan *Kasiski*.

Bila pada teknik di atas, setiap teks kode harus selalu mengganti nilai setiap teks asli tertentu (tidak peduli apakah jumlah teks kode tersebut ekuivalen dengan teks asli tertentu satu atau lebih). Pada teknik substitusi *vigenere* setiap teks kode

bisa mempunyai banyak kemungkinan teks asli. Teknik dari substitusi *vigenere* dapat dilakukan dengan dua cara, yaitu angka dan huruf [11].

### 2.2.1 Cara Angka Pada *Vigenere Cipher*

Teknik substitusi *vigenere* dengan menggunakan angka dilakukan dengan cara menukarkan huruf dengan angka. Teknik ini hampir sama dengan kode geser.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Gambar 2.5 Cara Angka Pada *Vigenere Cipher* [11]

### 2.2.2 Cara Huruf Pada *Vigenere Cipher*

Ide dasarnya yaitu dengan menggunakan kode kaisar, tetapi jumlah pergeseran pada hurufnya berbeda-beda untuk setiap periode beberapa huruf tertentu. Agar dapat mengenkripsi pesan dengan kode *vigenere*, digunakan tabula recta (disebut juga sebagai bujur sangkar *vigenere*) seperti gambar

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

Gambar 2.6 Tabula Recta [11]

Tabula recta berfungsi untuk memperoleh teks kode dengan menggunakan kunci yang telah ditentukan. Bila panjang kunci lebih pendek daripada panjang teks asli maka penggunaan kunci harus diulang. Secara matematis enkripsi dengan kode *vigenere* harus bisa dinyatakan sebagai [11]:

$$E(p_i) = V(p_i, k(I \bmod m))$$

dengan:

- $p_i$  = huruf ke-I dalam teks asli
- $k_n$  = huruf ke-n dalam kunci
- $m$  = panjang kunci, dan
- $V(x,y)$  = huruf yang tersimpan pada baris x dan kolom y pada tabula recta.



### 2.3 Metode *Alpha-Qwerty Reverse*

*Alpha-Qwerty Reverse* mirip dengan *Alpha-Qwerty Cipher* dalam hal aljabar deskripsi dan juga desain eksperimental yang ditentukan untuk *alpha-qwerty cipher* juga dapat diterapkan pada *alpha-qwerty reverse*.

Namun *alpha-qwerty reverse* berbeda dari *alpha-qwerty cipher* secara normal dalam hal urutan karakter yang digunakan untuk urutan teks biasa. Ini memberikan keuntungan tambahan dari tingkat kebingungan yang lebih besar dalam teks *cipher*.

Karakter set *alpha-qwerty reverse* diberikan pada gambar. Urutan teksnya sebagai berikut [13]:

q-z, Q-Z, ` ~ ! @ # \$ % ^ & \* ( ) \_ - = + { } [ ] | ; : " < > , . ? / , 0-9

urutan teks *cipher* sebagai berikut:

a-z, A-Z, 0-9, ` ~ ! @ # \$ % ^ & \* ( ) \_ - = + { } [ ] | ; : " < > , . ? /



Tabel 2.1 Perbandingan Metode

	Metode <i>Vigenere Cipher</i>	Metode <i>Alpha-Qwerty Reverse</i>	Metode Kombinasi <i>Vigenere Cipher5x</i> dan <i>Vigenere Cipher</i>
Pengenalan	Metode yang menggunakan bentuk bujur sangkar vigenere atau yang dikenal sebagai <i>tabula recta</i>	Metode yang menggunakan bentuk bujur sangkar dengan urutan keyboard <i>qwerty</i> pada sisi atas dan samping	Metode penggabungan antara metode <i>vigenere cipher</i> dan metode <i>alpha-qwerty reverse</i>
Prinsip Kerja	Menggunakan 26 Karakter	Menggunakan 92 Karakter	Menggunakan 93 Karakter
Proses	Penyandian <i>plaintext</i> menggunakan perpanjangan alfabet berurutan untuk diperpanjang menjadi perpanjangan alfabet berurutan	Penyandian <i>plaintext</i> menggunakan urutan keyboard <i>qwerty</i> untuk diperpanjang menjadi perpanjangan alfabet berurutan	Penyandian <i>plaintext</i> menggunakan dengan metode <i>vigenere cipher</i> sebanyak 5 kali proses dan hasil <i>ciphertext</i> nya digunakan sebagai <i>plaintext</i> pada metode <i>alpha-qwerty reverse</i>

## 2.5 Short Message Service (SMS)

*Short Message Service* (SMS) merupakan salah satu jenis *Instant Messaging* (IM) yang dapat memungkinkan pengguna agar bisa menukarkan pesan singkat pada setiap waktu, walaupun pengguna sedang melakukan panggilan data ataupun suara. SMS dihantarkan oleh *channel signal GSM (Global System for Mobile Communication)* dengan spesifikasi teknis ETSI. SMS yang dihidupkan oleh

ETSI dan difungsikan pada *scope* 3GPP. SMS juga difungsikan terhadap teknologi GPRS dan CDMA. SMS menjamin pengiriman pesan yang dilakukan oleh jaringan, bila terjadi kegagalan, maka pesan tersebut akan disimpan terlebih dahulu pada jaringan. Pada pengiriman paket SMS, bersifat *out of band* dengan menggunakan *bandwidth* yang rendah.

Dengan munculnya perkembangan teknologi yang sangat pesat pada akhir-akhir ini, teknologi SMS yaitu suatu teknologi yang sudah tidak asing lagi bagi kehidupan manusia. Teknologi SMS tersebut telah banyak disukai oleh orang karena teknologi tersebut bersifat murah, praktis serta mudah untuk digunakan. SMS juga merupakan suatu sistem pengiriman pesan yang sederhana pada telepon seluler. Fitur pada SMS didukung oleh GSM (*Global System for Mobile Communication*), TDMA (*Time Multiple Digital Access*) dan CDMA (*Code Multiple Digital Access*).

Semakin cepatnya perkembangan teknologi SMS pada telepon seluler saat ini juga didukung oleh beberapa faktor di antara nya adalah harga perangkat keras (telepon seluler) semakin. Selain itu, faktor lain yang mempengaruhi yaitu banyaknya penyedia jasa telekomunikasi yang telah menawarkan jasanya dengan harga yang cukup terjangkau oleh semua orang saat ini [14]. Layanan surat masa singkat atau pesan singkat yang dikenal sebagai SMS adalah sebuah layanan yang dilakukan pada sebuah telepon seluler untuk mengirim maupun menerima pesan-pesan singkat. Pada awalnya SMS dibentuk sebagai bagian daripada GSM, tetapi hingga saat ini sudah didapatkan pada jaringan bergerak lainnya termasuk jaringan UMTS.

Sebuah pesan pada layanan SMS memiliki maksimal yang terdiri dari 140 *bytes*, dengan kata lain sebuah pesan bisa memenuhi 140 karakter 8-bit, 160 karakter 7-bit atau 70 karakter 16-bit untuk yang menggunakan bahasa Jepang, bahasa Mandarin dan bahasa Korea yang memakai Hanzi (Aksara Kanji/Hanja). Selain 140 *bytes*, juga ada data-data lain yang telah termasuk. Adapula beberapa metode untuk melakukan pengiriman pesan jika lebih dari 140 *bytes*, tetapi seorang pengguna harus membayar lebih dari sekali. Pesan-pesan pada layanan SMS dikirim dari sebuah telepon seluler ke pusat pesan (SMSC). Disinilah pesan

akan disimpan dan mencoba mengirimnya kembali selama beberapa kali. Setelah waktu yang sudah ditentukan, biasanya 1 hari atau 2 hari, pesan tersebut akan dihapus. Seorang pengguna bisa memperoleh konfirmasi dari pusat pesan ini.

SMS memiliki keterbatasan yaitu pada ukuran pesan yang dapat dikirimkan maksimal sebesar 160 *byte*. Hal ini disebabkan karena mekanisme jalur transmisi SMS itu sendiri. Pada awalnya, SMS adalah layanan yang telah ditambahkan pada sistem GSM yang difungsikan untuk melakukan pengiriman data mengenai konfigurasi dari *handset* pelanggan GSM. SMS dikirimkan menggunakan *signaling frame* pada kanal frekuensi atau *time slot frame* GSM yang biasanya difungsikan untuk melakukan pengiriman pesan sebagai *control* dan sinyal *setup* panggilan telepon, seperti pesan singkat mengenai kesibukan jaringan maupun pesan CLI (*Caller Line Identification*). *Frame* ini mempunyai sifat yang khusus dan terdapat pada setiap panggilan telepon serta tidak dapat difungsikan untuk membawa *voice* atau data dari pelanggan. *Frame* pada sistem GSM memiliki ukuran sebesar 1250 bit (kurang lebih sama dengan 160 *byte*). Karena dengan menggunakan satu *frame* inilah menyebabkan pengiriman pesan SMS menjadi sangat murah, karena perhitungan pada beban biaya hanya dihitung dari penggunaan satu *frame* melalui kanal frekuensi. Pengiriman SMS yang menggunakan *frame* pada kanal frekuensi berarti SMS yang dikirim oleh pengirim terhadap nomor telepon tertentu bertindak sebagai SMSC (*SMS-center*) dan kemudian SMSC bertugas untuk meneruskannya kepada penerima. Pada pengiriman SMS, berlangsung cepat dikarenakan SMSC selain terhubung dengan LAN aplikasi juga terhubung dengan MSC (*Mobile Switching Network*) melalui SS7 (*Signalling System 7*) yang merupakan jaringan khusus yang berfungsi sebagai penangkap *frame control* dan sinyal. Mekanisme pengiriman pesan singkat pada layanan SMS juga serupa dengan yang ditemukan dalam sistem jaringan lain seperti TDMA, PDC dan cdmaOne. Perbedaan antara sistem jaringan yang satu dengan yang lainnya adalah terletak pada ukuran dari pesan SMS tersebut yang bergantung pada ukuran *frame* yang telah digunakan pada setiap sistem. Ukuran pesan SMS pada sistem TDMA dan PDC sama dengan sistem

GSM sebesar 160 *byte*. Sedangkan pada cdmaOne memiliki ukuran pesan SMS sebesar 256 *byte* [14].

### 2.5.2 Mekanisme Kerja SMS

Saat SMS dilakukan pengiriman ke sebuah nomor tertentu, SMS yang dikirimkan bukan berarti langsung dikirimkan ke nomor tersebut, namun akan masuk terlebih dahulu ke SMS *Center* atau yang biasa disebut SMSC yaitu operator telepon yang sedang digunakan. SMS *Center* tersebut dapat dimaksudkan sebagai sebuah *server* yang memiliki tugas yaitu bertanggung jawab terhadap proses pengiriman SMS dalam suatu operator. SMS yang telah dikirimkan dari suatu telepon seluler akan tiba ke SMSC ini yang kemudian baru dilanjutkan ke nomor tujuan SMS tersebut. Jika nomor yang hendak dituju ternyata sedang mengalami mati/*offline*, Secara otomatis, SMSC ini akan menyimpan SMS tersebut untuk sementara waktu, sampai nomor tujuan tersebut hidup/*online* kembali. Lamanya waktu penyimpanan SMS tersebut sangat tergantung dari lamanya waktu yang sudah ditetapkan oleh operator. Jika nomor tersebut telah menerima SMS, maka secara otomatis akan mengirimkan laporan ke SMSC bahwa SMS tersebut telah diterima. Dari SMSC, laporan tersebut kemudian akan diteruskan kembali ke nomor pengirim SMS [14].

## 2.6 Android










Gambar 2.8 Logo Android [14]









Android yaitu sebuah sistem operasi telepon seluler berbasis Linux. Android juga memiliki *platform* terbuka bagi seseorang yang ingin menciptakan aplikasi untuk digunakan oleh bermacam-macam peranti bergerak. Android juga merupakan sebuah sistem operasi untuk telepon seluler seperti halnya Symbian pada Nokia, *Palm* dan *Windows Mobile* yang sebelumnya sudah dikenal selama ini. Seiring dengan perkembangannya, android berubah menjadi *platform* yang begitu cepat dalam melakukan pembaruan. Pada awalnya, android dikembangkan oleh Android inc, yang didukung finansial dari pihak *google*, yang kemudian membelinya pada tahun 2005. Pada tahun 2007, sistem operasi ini telah resmi yang bersamaan dengan didirikannya *Open handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak serta telekomunikasi yang tidak lain bertujuan untuk memajukan standar terbuka perangkat seluler. Android, inc didirikan pada bulan Oktober tahun 2003 oleh Andy Rubin, Christ White, Rich Miner dan Nick Sears di Palo Alto California yang bertujuan untuk mengembangkan perangkat seluler pintar yang lebih tahu akan lokasi dan preferensi penggunanya [14].

Android juga memberikan pilihan untuk perusahaan di bidang teknologi yang menginginkan sistem operasi yang bisa dikustomasi, berbiaya rendah serta ringan pada perangkat berteknologi tinggi tanpa harus dikembangkan dari nol. Pada tahun 2008, android secara bertahap sudah melakukan beberapa pembaruan untuk menambahkan fitur-fitur yang baru, memperbaiki *bug* maupun meningkatkan sistem kerja operasi yang dimiliki pada versi sebelumnya. Dari versi ke versi, android membuat nama yang unik seperti penamaan berbagai kue atau makanan dengan abjad yang berurutan dan ada juga penamaan yang diberi angka versinya. Ada pula penamaan yang diketahui lewat tingkat *Application Programming Interface (API)* [15]. Tabel berikut ini menunjukkan berbagai versi pengembangan sistem operasi android [15].

Tabel 2.2 Pengembangan Sistem Operasi Android dan Perbandingan Fitur-Fitur Berbagai Versi Android

No	Versi	Gambar	Level API
1	Andoid (Versi 1.1)		1
2	Android <i>Cupcake</i> (Versi 1.5)		3
3	Android <i>Donut</i> (Versi 1.6)		4
4	Android <i>Éclair</i> (Versi 2.0-2.1)		7
5	Android <i>Frozen Yogurt/Froyo</i> (Versi 2.2)		8
6	Android <i>Gingerbread</i> (Versi 2.3)		9-10
7	Android <i>Honeycomb</i> (Versi 3.0-3.2)		12-13



8	<p>Android <i>Ice Cream Sandwich/ics</i> (Versi 4.0)</p>		.15
9	<p>Android <i>Jellybean</i> (Versi 4.1)</p>		16
10	<p>Android <i>Jellybean</i> (Versi 4.2)</p>		17
11	<p>Android <i>Jellybean</i> (Versi 4.3)</p>		18
12	<p>Android <i>Kitkat</i> (Versi 4.4)</p>		19
13	<p>Android <i>Lollipop</i> (Versi 5.0)</p>		21
14	<p>Android <i>Marshmallow</i> (Versi 6.0)</p>		23
15	<p>Android <i>Nougat</i> (Versi 7.0)</p>		24

## 2.7 *Android Jelly Bean*

Sistem Operasi Android *Jelly Bean* dimulai dari versi 4.1 yang dirilis pada tanggal 9 Juli 2012 saat Konferensi *Google I/O* dan diperkenalkan kepada publik beberapa bulan kemudian. Pada tanggal 13 November 2012, Android *Jelly Bean* memulai versi 4.2 dan siap dirilis. Android *Jelly Bean* versi 4.2 diklaim lebih pintar dan inovatif dibandingkan dengan pendahulunya. Fitur – fitur yang ditambahkan dalam sistem operasi ini diantaranya yaitu *Notifications*, *Barrel Roll*, *Google Now*, *Google Assistant*, *Smart Widget*, *Face unlock* dengan *Liveness Check* serta teknologi *Project Butter* yang berfungsi untuk menaikkan responsifitas agar lebih baik. Sistem Operasi Android *Jelly Bean* ini merupakan versi yang terhalus dan tercepat dibandingkan versi android lainnya [16].

## 2.8 **Peranti Pengembangan Aplikasi Android**

Perangkat berbasis android memiliki pertumpuhan pasar yang mendorong pertumbuhan terhadap pengembangan aplikasi berbasis android. Bagi seseorang yang ingin mengembangkan untuk pembuatan aplikasi, peranti yang memudahkan pembuatan aplikasi tentu saja diharapkan. Pada situs android *developers* memudahkan siapa pun untuk membuat aplikasi android yaitu dengan menyediakan Android SDK (*Software Development Kit*). Aplikasi yang diperlukan untuk melakukan pengembangan aplikasi android di antaranya *Java Development Kit (JDK)*, *Eclipse*, *Android SDK* dan *Android Development Tools (ADT)* [14].

### 2.8.1 *Eclipse*

*Eclipse* merupakan sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari *eclipse* [17]:

**Multi-platform:** Target sistem operasi *eclipse* adalah *Microsoft Windows*, *Linux*, *Solaris*, *AIX*, *HP-UX* dan *Mac OS X*.

**Mult-language:** *Eclipse* dikembangkan dengan bahasa pemrograman java, akan tetapi *eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++*, *Cobol*, *Python*, *Perl*, *PHP*, dan lain sebagainya.

**Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, *eclipse* pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.



Gambar 2.9 Logo *Eclipse* [14]

*Eclipse* pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, dengan kata lain, setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, *eclipse* memiliki kelebihan adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in* [14].

### 2.8.2 JDK (*Java Development Kit*)

JDK merupakan aplikasi java yang memiliki perangkat pengembangan. Perangkat tersebut mutlak dibutuhkan guna membuat aplikasi android dikarenakan aplikasi android berbasis java. Dimana yang sudah diketahui, Java yaitu salah satu bahasa pemrograman yang dapat digunakan untuk membuat aplikasi. Tetapi perlu diketahui, tidak seluruh pustaka yang terdapat di dalam java digunakan pada android. JDK yaitu semacam kotak peralatan atau kit yang

difungsikan untuk *development*. JDK tersebut berfungsi untuk membuat kode program untuk pembuatan aplikasi yang baru ataupun melakukan pengembangan aplikasi. JDK telah memuat *library* dan program-program yang diperlukan untuk meng-*compile* dan me-*launch* pada program Java di mana dua program utama yang terdapat di JDK yaitu *javac* dan *java* [14].

### 2.8.3 Android SDK (*System Development Kit*)

Android SDK merupakan peralatan API (*Application Programming Interface*) yang dibutuhkan untuk melakukan pengembangan aplikasi pada *platform* android dengan menggunakan bahasa pemrograman java. Android SDK menyediakan peralatan dan API untuk melakukan pengembangan *platform* aplikasi android dengan menggunakan bahasa pemrograman java. Android merupakan salah satu aplikasi yang dikembangkan dengan berbasis java, sehingga sebelum melakukan *coding* aplikasi berbasis android, PC/komputer harus sudah terinstal program java. Paket untuk instalasi dapat langsung di-*download* secara gratis di *website* [http : // www .oracle . com/ technetwork/ java/ javase/ downloads/ index. html](http://www.oracle.com/technetwork/java/javase/downloads/index.html). Jika menggunakan *platform* Linux seperti Ubuntu, java memiliki program yang sudah termasuk dalam *repository* dalam setiap versi dari Ubuntu dan hanya menghubungkan Ubuntu tersebut ke *repository* lalu melakukan instalasi java [14].

### 2.8.4 Android Development Tools (ADT)

*Android Development Tools* merupakan *plugin* yang dibentuk untuk IDE *Eclipse* yang memberikan kemudahan dalam melakukan pengembangan aplikasi android. Dengan adanya ADT untuk *eclipse*, itu akan memudahkan bagi *develop* dalam proses pembuatan aplikasi *project* android, membuat GUI aplikasi serta menambahkan komponen-komponen lainnya, Selain itu, kita juga dapat melakukan *running* aplikasi menggunakan android SDK melalui *eclipse*. Dengan ADT, kita dapat membuat *package Android* (.apk) yang berfungsi untuk pendistribusian aplikasi android yang telah dirancang [14].

## 2.9 Java



Gambar 2.10 Logo Java [14]

Java merupakan sebuah *platform* dengan bahasa pemrograman tingkat tinggi yang mempunyai kriteria yang sederhana, berorientasi objek, dinamis, terdistribusi, aman dan lain-lain. Bahasa tersebut dikembangkan dengan menggunakan model yang mirip dengan bahasa C++ dan *smalltalk* tetapi lebih mudah untuk digunakan dan juga memiliki *platform* independen yang bisa dijalankan pada seluruh sistem operasi [14].