

BAB II

TINJAUAN PUSTAKA

2.1 Enkripsi dan Dekripsi

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). Enkripsi dapat diartikan sebagai kode atau *chipper*. Sebuah system pengkodean menggunakan suatu tabel atau kamus yang telah didefinisikan untuk mengganti kata dari informasi atau yang merupakan bagian dari informasi yang dikirim. Sebuah *chipper* menggunakan suatu algoritma yang dapat mengkodekan semua aliran data (*stream*) bit dari sebuah pesan menjadi *cryptogram* yang tidak dimengerti (*unintelligible*). Karena teknik *chipper* merupakan suatu system yang telah siap untuk di automasi, maka teknik ini digunakan dalam system keamanan computer dan jaringan. Enkripsi dimaksudkan untuk melindungi informasi agar tidak terlihat oleh orang atau pihak yang tidak berhak. Informasi ini dapat berupa nomor kartu kredit, catatan penting dalam computer, maupun *password* untuk mengakses sesuatu. Enkripsi dibentuk berdasarkan suatu algoritma yang akan mengacak suatu informasi menjadi bentuk yang tidak bisa dibaca atau tak bisa dilihat.

Dekripsi adalah proses dengan algoritma yang sama untuk mengembalikan 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 *ter-enkripsi* yang tidak bisa dikembalikan oleh seseorang, bahkan sekalipun mereka memiliki algoritma yang sama. Sebuah contoh algoritma sederhana seperti algoritma direncanakan, selanjutnya disebut algoritma ($\text{karakter}+3$), agar mampu mengubah setiap karakter menjadi karakter nomor tiga setelahnya. Artinya, setiap menemukan huruf A, maka algoritma akan mengubahnya menjadi D, B menjadi E, C menjadi F, dan seterusnya.

Sebuah pesan asli, disebut *plaintext* dalam bahasa kriptografi, dikonversikan oleh algoritma karakter+3 menjadi *chiphertext* (bahasa kriptografi untuk hasil enkripsi). Sedangkan mendekripsikan pesan digunakan algoritma dengan fungsi kebalikannya yaitu karakter-3.

Metode enkripsi yang lebih umum adalah menggunakan sebuah algoritma dan sebuah kunci. Pada contoh di atas, algoritma bisa diubah menjadi karakter+x, dimana x adalah variabel yang berlaku sebagai kunci. Kunci bisa bersifat dinamis, artinya kunci dapat berubah-ubah sesuai kesempatan untuk lebih meningkatkan keamanan pesan. Kunci harus diletakkan terpisah dari pesan yang terenkripsi dan dikirimkan secara rahasia. Teknik semacam ini disebut sebagai *symmetric (single key)* atau *secret key cryptography*.

2.1.1 Model-Model Enkripsi

Dalam membahas model-model enkripsi beserta algoritma yang akan dipakai untuk setiap enkripsi ada 2 hal yang penting yang dijabarkan, yaitu enkripsi dengan kunci pribadi dan enkripsi dengan kunci public.

a. Enkripsi dengan kunci pribadi

Enkripsi dapat dilakukan jika si pengirim dan si penerima telah sepakat untuk menggunakan metode enkripsi atau kunci enkripsi tertentu. Metode enkripsi atau kuncinya ini harus dijaga ketat supaya tidak ada pihak luar yang mengetahuinya. Kesepakatan cara enkripsi atau kunci dalam enkripsi ini bisa dicapai lewat jalur komunikasi lain yang lebih aman, misalnya dengan bertemu langsung. Cara enkripsi dengan kesepakatan atau kunci enkripsi di atas dikenal dengan istilah enkripsi dengan kunci pribadi, karena cara enkripsi atau kunci yang hanya boleh diketahui oleh dua pribadi yang berkomunikasi tersebut. Cara enkripsi inilah yang umum digunakan pada saat ini baik untuk kalangan pemerintah maupun kalangan bisnis. Cara enkripsi ini juga dikategorikan sebagai kriptografi simetris, karena dua belah pihak mengetahui kunci yang sama. Selain masalah komunikasi awal untuk

penyampaian kunci, cara enkripsi ini juga mempunyai kelemahan yang lain. Kelemahan ini timbul jika terdapat banyak orang yang ingin saling berkomunikasi. Karena setiap pasangan harus menghafal banyak kunci dan harus menggunakannya secara tepat. Sebab, jika tidak, maka si penerima tidak bisa mengartikannya.

b. Enkripsi dengan Kunci Publik

Cara enkripsi ini mempunyai banyak kelebihan, salah satunya adalah tiap orang hanya perlu memiliki satu set kunci, tanpa peduli berapa banyak orang yang akan diajak berkomunikasi. Jadi, jika ada n set kunci saja. Selain itu, cara enkripsi ini tidak membutuhkan saluran yang aman untuk pengiriman kunci, sebab kunci yang dikirimkan ini harus diketahui oleh public. Cara enkripsi sangat praktis sehingga masyarakat umum pun dapat dengan mudah memakainya.

Cara kerja enkripsi ini secara singkat dapat diterangkan sebagai berikut. Setiap orang yang menggunakan enkripsi ini harus mempunyai dua buah kunci, satu disebut kunci rahasia yang hanya boleh diketahui oleh dirinya sendiri dan yang lain disebut kunci public yang disebarakan ke orang lain. Kedua kunci ini dibuat secara acak dengan menggunakan rumus matematika tertentu. Jadi, kedua kunci ini berkaitan erat secara matematika. Jika si A hendak mengirim pesan kepada si B, si A perlu mengenkrip pesan itu dengan kunci public milik si B. Pesan si A yang telah dienkrip dengan menggunakan kunci public si B hanya bisa dibuka dengan kunci public itu sendiri. Si B wajib untuk menjamin keamanan kunci rahasianya.

Karena kunci rahasia ini tidak perlu diketahui pihak si pengirim berita, kunci ini tidak akan pernah dikirim lewat jalur umum. Hal ini membuat cara ini jauh lebih aman daripada enkripsi dengan kunci pribadi. Misalkan si C dapat mengirim ke B dengan menggunakan kunci public si B yang sama. Walaupun mengetahui kunci public si B, pesan yang telah dienkrip dengan itu sangat sulit untuk dibuka. Cara enkripsi ini dikategorikan dalam kriptografi

asimetris, karena kunci yang dipakai untuk mengenkrip dan untuk membuka enkrip adalah dengan menggunakan dua kunci yang berbeda (Wahana Komputer, dkk, 2003:94).

2.2 *Vigenere Chiper*

Vigenere cipher merupakan salah satu algoritma klasik dengan teknik substitusi. Nama vigenere diambil dari seorang yang bernama Blaise de Vigenere. *Vigenere cipher* menggunakan suatu kunci yang memiliki panjang tertentu. Panjang kunci tersebut bisa lebih pendek ataupun sama dengan panjang plainteks. Jika panjang kunci kurang dari panjang plainteks, maka kunci yang tersebut akan diulang secara periodik hingga panjang kunci tersebut sama dengan panjang plainteksnya (Bayu K.N, 2010:12).

Vigenere cipher adalah salah satu algoritma kriptografi klasik yang diperkenalkan pada abad 16 atau kira-kira pada tahun 1586. Algoritma kriptografi ini dipublikasikan oleh seorang diplomat dan juga kriptologis yang berasal dari Prancis, yaitu Blaise de Vigenère, namun sebenarnya algoritma ini telah digambarkan sebelumnya pada buku *La Cifra del Sig. Giovan Batista Belaso*, sebuah buku yang ditulis oleh Giovan Batista Belaso, pada tahun 1553 (Becik G.A, 2011:3).

Cara kerja dari Vigenère cipher ini mirip dengan Caesar cipher, yaitu mengenkripsi plainteks pada pesan dengan cara menggeser huruf pada pesan tersebut sejauh nilai kunci pada deret alphabet. Vigenère cipher adalah salah satu algoritma kriptografi klasik yang menggunakan metode substitusi abjad-majemuk. Substitusi abjad-majemuk mengenkripsi setiap huruf yang ada menggunakan kunci yang berbeda, tidak seperti Caesar cipher yang menerapkan metode substitusi abjad-tunggal yang semua huruf di suatu pesan dienkripsi menggunakan kunci yang sama.

Vigenere cipher yang menerapkan metode substitusi abjad-majemuk tidak memiliki permasalahan tersebut karena setiap huruf pada pesan yang dienkripsi dengan Vigenère cipher ini akan digeser dengan nilai yang berbeda tergantung dengan kunci yang diberikan. Kunci yang digunakan pada Vigenère cipher berbeda

dengan yang digunakan pada Caesar cipher. Jika pada Caesar cipher kuncinya hanya satu nilai saja, maka pada Vigenère cipher kunci yang digunakan berbentuk deretan huruf. Kunci yang berbentuk deretan kata tersebut akan memungkinkan setiap huruf plaintext untuk dienkripsi dengan kunci yang berbeda. Jika panjang kunci yang digunakan lebih pendek dari panjang plaintext maka kunci akan diulang sampai panjang kunci sama dengan panjang plaintext.

Tabel 2.1 Tabel Vigenere Chipper

| | 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 |

Sumber: Wikipedia.com

Sandi Vigenere sebenarnya merupakan pengembangan dari sandi Caesar. Pada sandi Caesar, setiap huruf teks terang digantikan dengan huruf lain yang memiliki perbedaan tertentu pada urutan alfabet. Misalnya pada sandi Caesar dengan geseran 3, A menjadi D, B menjadi E and dan seterusnya. Sandi Vigenère terdiri dari beberapa sandi Caesar dengan nilai geseran yang berbeda, Untuk menyandikan suatu pesan, digunakan sebuah tabel alfabet yang disebut tabel Vigenère (gambar). Tabel Vigenère berisi alfabet yang dituliskan dalam 26 baris, masing-masing baris digeser satu urutan ke kiri dari baris sebelumnya, membentuk ke-26 kemungkinan sandi Caesar. Setiap huruf disandikan dengan menggunakan baris yang berbeda-beda,

sesuai kata kunci yang diulang. Misalnya, teks terang yang hendak disandikan adalah perintah "Serbu Berlin":

serbuberlin

Sedangkan kata kunci antara pengirim dan tujuan adalah "Pizza". "PIZZA" diulang sehingga jumlah hurufnya sama banyak dengan teks terang:

PIZZAPIZZAP

Huruf pertama pada teks terang, S, disandikan dengan menggunakan baris berjudul P, huruf pertama pada kata kunci. Pada baris P dan kolom S di tabel Vigenère, terdapat huruf H. Demikian pula untuk huruf kedua, digunakan huruf yang terletak pada baris I (huruf kedua kata kunci) dan kolom E (huruf kedua teks terang), yaitu huruf M. Proses ini dijalankan terus sehingga:

Tabel 2.2 Tabel hasil chipertext

| | |
|------------|-------------|
| plaintext | serbuberlin |
| keyword | PIZZAPIZZAP |
| chipertext | HMQAUQMQKIC |

Sumber: Wikipedia.com

Proses sebaliknya (disebut dekripsi), dilakukan dengan mencari huruf teks bersandi pada baris berjudul huruf dari kata kunci. Misalnya, pada contoh di atas, untuk huruf pertama, kita mencari huruf H (huruf pertama teks tersandi) pada baris P (huruf pertama pada kata kunci), yang terdapat pada kolom S, sehingga huruf pertama adalah S. Lalu M terdapat pada baris I di kolom E, sehingga diketahui huruf kedua teks terang adalah E, dan seterusnya hingga didapat perintah "serbuberlin".

2.2.1 Perhitungan Vigenere Chiper

Enkripsi (penyandian) dan dekripsi dengan sandi Vigenère juga dapat dituliskan secara matematis, dengan menggunakan penjumlahan dan operasi modulus, yaitu:

Rumus enkripsi *vigenere cipher* :

$$C_i = (P_i + K_i) - 26 \text{ kalau hasil penjumlahan } P_i \text{ dan } K_i \text{ lebih dari } 26$$

Rumus dekripsi *vigenere cipher* :

$$P_i = (C_i - K_i) + 26 \text{ kalau hasil pengurangan } C_i \text{ dengan } K_i \text{ minus}$$

Dimana:

C_i = nilai desimal karakter ciphertext ke- i

P_i = nilai desimal karakter plaintext ke- i

K_i = nilai desimal karakter kunci ke- i

Nilai desimal karakter:

A=0 B=1 C=2 ... Z=25

Sebagai contoh, jika plaintext adalah **STIKOMBALI** dan kunci adalah **KAMPUS** maka proses enkripsi yang terjadi adalah sebagai berikut:

Tabel 2.3 Tabel hasil dekripsi

| | |
|------------|-------------------|
| Plaintext | STIKOMBALI |
| Key | KAMPUSKAMP |
| Ciphertext | CTUZIELAXX |

Sumber: Wikipedia.com

Pada contoh di atas kata kunci **KAMPUS** diulang sedemikian rupa hingga panjang kunci sama dengan panjang plainteksnya. Jika dihitung dengan rumus enkripsi vigenere plainteks huruf pertama **S** (yang memiliki nilai **$P_i=18$**) akan dilakukan pergeseran dengan huruf **K** (yang memiliki **$K_i=10$**) maka prosesnya sebagai berikut:

$$C_i = (P_i + K_i) \text{ mod } 26$$

$$\begin{aligned}
 &= (18 + 10) \bmod 26 \\
 &= 28 \bmod 26 \\
 &= 2
 \end{aligned}$$

$C_i = 2$ maka huruf ciphertext dengan nilai **2** adalah **C** . Begitu seterusnya dilakukan pergeseran sesuai dengan kunci pada setiap huruf hingga semua plainteks telah terenkripsi menjadi ciphertext. Setelah semua huruf terenkripsi maka proses dekripsinya dapat dihitung sebagai berikut:

$$\begin{aligned}
 P_i &= (C_i - K_i) + 26 \\
 &= (2 - 10) + 26 \\
 &= -8 + 26 \\
 &= 18
 \end{aligned}$$

$P_i = 18$ maka huruf plainteks dengan nilai **18** adalah **S**. Begitu seterusnya dilakukan pergeseran sesuai dengan kunci pada setiap huruf hingga semua ciphertext telah terdekripsi menjadi plainteks (Wikipedia.com).

2.3 SMS (*Short Messages Service*)

Short Message Service (SMS) adalah salah satu tipe *Instant Messaging* (IM) yang memungkinkan *user* untuk bertukar pesan singkat kapanpun walaupun *user* sedang melakukan sambungan data/suara. SMS dihantarkan pada *channel signal* GSM (*Global System for Mobile Communication*) spesifikasi teknisi ETSI. SMS diaktifkan oleh ETSI dan dijalankan di *scope* 3GPP. SMS juga digunakan pada teknologi GPRS dan CDMA. SMS menjamin pengiriman pesan oleh jaringan, jika terjadi kegagalan pesan akan disimpan dahulu di jaringan, pengiriman paket SMS bersifat *out of band* dan menggunakan *bandwidth* rendah (Ricky H, 2010).

Dengan adanya perkembangan teknologi yang sangat pesat akhir-akhir ini, teknologi SMS merupakan suatu teknologi yang tidak asing lagi dalam kehidupan masyarakat. Teknologi SMS ini banyak digemari oleh masyarakat karena teknologi ini bersifat praktis, murah, dan mudah untuk digunakan.

SMS merupakan suatu system pengiriman pesan sederhana yang disediakan oleh jaringan telepon seluler. Fitur SMS ini didukung oleh GSM (*Global System for*

Mobile Communication), TDMA (*Time Multiple Digital Access*), CDMA (*Code Multiple Digital Access*).

Semakin pesatnya perkembangan teknologi SMS ini, didukung oleh beberapa faktor, antara lain adalah semakin terjangkaunya harga perangkat keras yang digunakan (telepon seluler). Selain itu, faktor lain yang mempengaruhi adalah banyaknya *provider* penyedia jasa telekomunikasi yang menawarkan jasanya dengan harga yang cukup terjangkau oleh masyarakat saat ini.

2.3.1 Layanan SMS

Layanan pesan singkat atau surat masa singkat (bahasa Inggris: *Short Message Service* disingkat SMS) adalah sebuah layanan yang dilaksanakan dengan sebuah telepon genggam untuk mengirim atau menerima pesan-pesan pendek. Pada mulanya SMS dirancang sebagai bagian daripada GSM, tetapi sekarang sudah didapatkan pada jaringan bergerak lainnya termasuk jaringan UMTS.

Sebuah pesan SMS maksimal terdiri dari 140 bytes, dengan kata lain sebuah pesan bisa memuat 140 karakter 8-bit, 160 karakter 7-bit atau 70 karakter 16-bit untuk bahasa Jepang, bahasa Mandarin dan bahasa Korea yang memakai Hanzi (Aksara Kanji / Hanja). Selain 140 bytes ini ada data-data lain yang termasuk. Adapula beberapa metode untuk mengirim pesan yang lebih dari 140 bytes, tetapi seorang pengguna harus membayar lebih dari sekali. Pesan-pesan SMS dikirim dari sebuah telepon genggam ke pusat pesan (SMSC), di sini pesan disimpan dan mencoba mengirimnya selama beberapa kali. Setelah sebuah waktu yang telah ditentukan, biasanya 1 hari atau 2 hari, lalu pesan dihapus. Seorang pengguna bisa mendapatkan konfirmasi dari pusat pesan ini.

Keterbatasan SMS adalah pada ukuran pesan yang dapat dikirimkan, yaitu maksimal sebesar 160 byte. Keterbatasan ini disebabkan karena mekanisme transmisi SMS itu sendiri. SMS pada awalnya adalah layanan yang ditambahkan pada system GSM yang digunakan untuk mengirimkan data mengenai konfigurasi dari *handset* pelanggan GSM. SMS dikirimkan menggunakan *signaling frame* pada kanal

frekuensi atau *time slot frame* GSM yang biasanya digunakan untuk mengirimkan pesan untuk control dan sinyal setup panggilan telepon, seperti pesan singkat tentang kesibukan jaringan atau pesan CLI (*Caller Line Identification*). *Frame* ini bersifat khusus dan ada pada setiap panggilan telepon serta tidak dapat digunakan untuk membawa *voice* atau data dari pelanggan. Ukuran *frame* pada system GSM sendiri adalah sebesar 1250 bit (kurang lebih sama dengan 160 byte). Karena hanya menggunakan satu *frame* inilah pengiriman pesan SMS menjadi sangat murah, karena beban biaya hanya dihitung dari penggunaan satu *frame* melalui kanal rekuensi. Pengiriman SMS menggunakan *frame* pada kanal frekuensi adalah berarti SMS dikirim oleh pengirim ke nomor telepon tertentu yang bertindak sebagai SMSC (*SMS-center*) dan kemudian SMSC bertugas untuk meneruskannya ke penerima. Pengiriman SMS berlangsung cepat karena SMSC selain terhubung ke LAN aplikasi juga terhubung ke MSC (*Mobile Switching Network*) melalui SS7 (*Signaling System 7*) yang merupakan jaringan khusus untuk menangkap *frame* control dan sinyal. Mekanisme pengiriman pesan singkat SMS yang serupa juga ditemukan dalam system jaringan lain seperti TDMA, PDC, dan cdmaOne. Beda antara system jaringan satu dengan yang lainnya adalah ukuran dari pesan SMS itu sendiri yang bergantung pada ukuran *frame* yang digunakan pada masing-masing system. Pada system TDMA dan PDC ukuran pesan SMS sama dengan system GSM, yaitu 160 byte, dan pada cdma-One ukuran pesan SMS sebesar 256 byte.

2.3.2 Mekanisme Kerja SMS

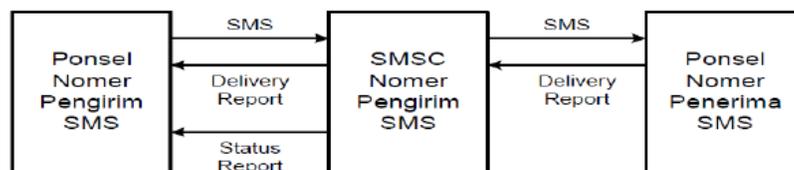
Ketika SMS dikirim ke suatu nomor, SMS tersebut tidak langsung dikirim ke nomor tersebut, melainkan akan masuk ke *SMS center* (SMSC) operator telepon yang digunakan terlebih dahulu. SMS sendiri dapat diartikan sebagai sebuah *server* yang bertanggung jawab pada proses pengiriman SMS pada suatu operator. Setelah SMS tersebut masuk ke SMSC, kemudian barulah diteruskan ke nomor tujuan SMS tersebut. Bila nomor tujuan *offline*, SMSC akan menyimpan SMS tersebut untuk

sementara waktu hingga nomor tujuan *online*, setelah menerima SMS, nomor tujuan akan mengirimkan laporan ke SMSC bahwa SMS telah berhasil diterima.

Mekanisme kerja pengiriman SMS dapat dibagi menjadi 3 macam, yaitu:

1. Pengiriman SMS Intra-Operator (satu operator)

SMS yang dikirimkan oleh nomor pengirim akan dimasukkan terlebih dahulu ke dalam SMSC operator nomor pengirim, kemudian SMSC tersebut akan mengirimkannya ke nomor yang di tuju secara langsung. Nomor penerima akan mengirimkan *delivery report* yang menyatakan bahwa SMS telah diterima kepada SMSC, SMSC kemudian meneruskan report tersebut kepada nomor pengirim SMS, disertai *status report* dari proses pengiriman SMS tersebut.

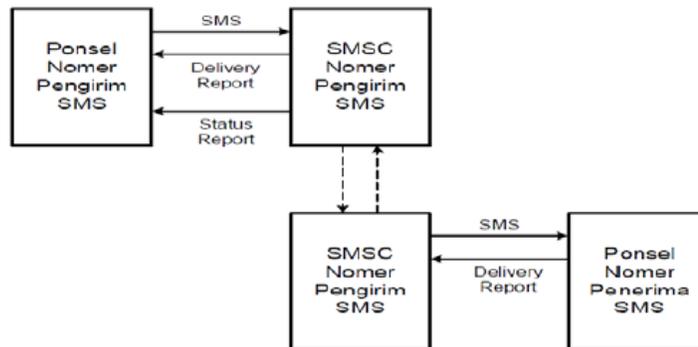


Mekanisme intra-operator SMS

Gambar 2.1 Mekanisme intra-operator SMS
(Ricky H, 2010: 11)

2. Pengiriman SMS Interoperator (operator yang berbeda)

Selain masuk ke SMSC operator pengirim, SMS yang dikirimkan akan diteruskan oleh SMSC operator pengirim ke SMSC operator penerima SMS, kemudian baru diteruskan ke nomor tujuan, *delivery report* yang dihasilkan harus melewati mekanisme yang sama sebelum diterima oleh nomor pengirim.



Mekanisme inter-operator SMS

**Gambar 2.2 Mekanisme inter-operator SMS
(Ricky H, 2010: 11)**

3. Pengiriman SMS dari operator suatu Negara ke Negara lain (SMS Internasional)

Mekanisme yang terjadi tidak jauh berbeda dengan mekanisme pada inter-operator SMS. Perbedaannya hanya pada SMSC nomor penerima, yang tentu saja adalah SMSC operator luar negeri, dan penambahan kode Negara pada nomor tujuan.

2.3.3 Keterbatasan SMS

SMS memiliki beberapa kelemahan. Beberapa kelemahan SMS antara lain:

1. Hanya dapat mengirim pesan berupa text tidak dapat berupa gambar maupun suara
2. Pesan terbatas oleh ukuran, hanya bisa 160 karakter saja (Ricky H, 2010).

2.4 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android juga menyediakan platform terbuka bagi para pengembang guna menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Android merupakan sebuah sistem operasi untuk telepon seluler seperti halnya Symbian pada Nokia, Palm dan Windows Mobile yang sebelumnya sudah

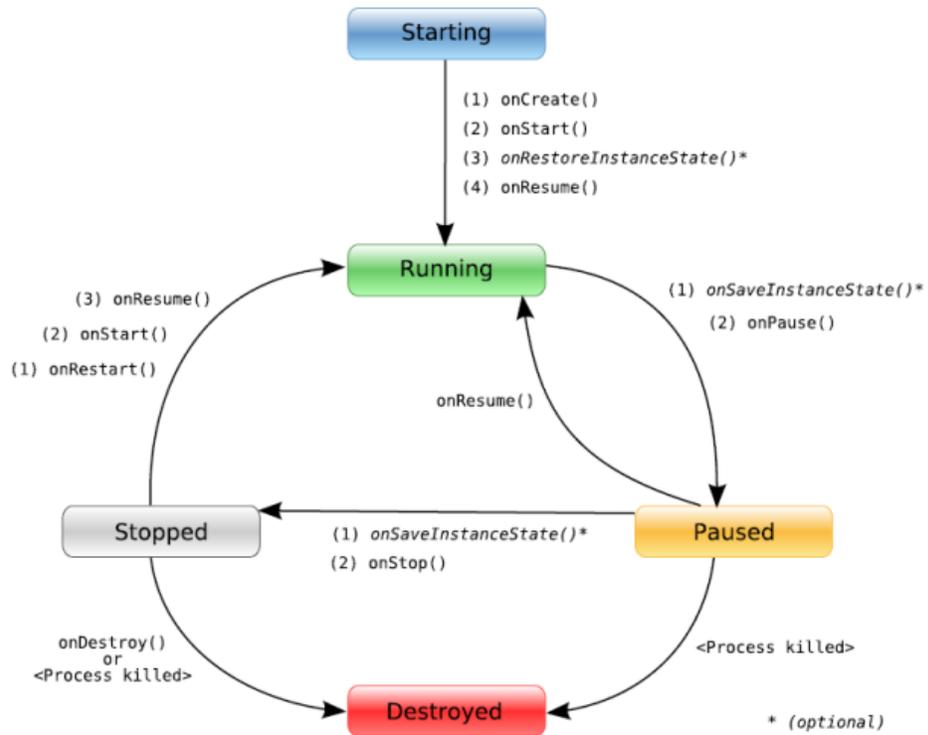
terlebih dahulu kita kenal selama ini. Sistem Android dipakai di telepon pertama kali pada tanggal 22 Oktober 2008 untuk HTC Dream. Pada beberapa bulan kemudian di tahun berikutnya Operating System Android sudah banyak sekali dipakai oleh berbagai jenis telepon seluler. Android disebut sebagai OS yang kuat, cepat dan juga sangat baik.



Gambar 2.3 Logo android

(https://id.wikipedia.org/wiki/Berkas:Android_robot_2014.svg, 17/05/2016)

Perangkat berbasis android hanya mempunyai satu layar *foreground*. Normalnya saat menghidupkan android, yang pertama terlihat adalah home. Kemudian bila menjalankan sebuah aplikasi catur, *User Interfacenya* (UI) akan menumpuk diatas layar sebelumnya (home). Kemudian bila melihat help-nya catur, maka UI help akan menimpa UI sebelumnya (catur), begitu seterusnya. Semua proses diatas direkam di *application stack* oleh sistem *Activity manager*. Menekan tombol back hanya kembali ke halaman sebelumnya, analoginya mirip dengan browser dimana ketika meng-klik tombol back browser akan kembali menampilkan halaman sebelumnya. Setiap *User Interface* diwakili oleh kelas Activity (Activity class). Setiap activity mempunyai siklus, dapat dilihat di gambar 2.4. Sebuah aplikasi dapat terdiri dari satu atau lebih activity yang diproses dalam Linux (Arif A.H, 2012:10).



Gambar 2.4 Siklus *activity*
(Arif A.H, 2012: 10)

Selama siklus ini berjalan, activity bisa mempunyai lebih dari 2 status seperti yang terlihat pada gambar 1.14. Kita tidak bisa mengontrol setiap status karena semuanya sudah ditangani oleh sistem. Namun kita akan mendapat pesan saat terjadi perubahan status melalui method onXX(). Berikut penjelasan setiap status.

Tabel 2.4 Tabel fungsi method onXX()

| | |
|------------------|---|
| onCreate(Bundle) | Dipanggil saat pertama kali aplikasi dijalankan. Kita dapat menggunakan ini untuk deklarasi variabel atau membuat user interface. |
|------------------|---|

| | |
|-----------------------------|--|
| onStart() | Mengindikasikan activity yang ditampilkan ke pengguna (user). |
| onResume() | Dipanggil saat aplikasi kita mulai berinteraksi dengan pengguna. Disini sangat cocok untuk meletakkan animasi ataupun musik. |
| onPause() | Dipanggil saat aplikasi yang Kita jalankan kembali ke halaman sebelumnya atau biasanya karena ada activity baru yang dijalankan. Disini cocok untuk meletakkan algoritma penyimpanan (save). |
| onStop() | Diapnggil saat aplikasi Kita berjalan di belakang layar dalam waktu cukup lama. |
| onRestart() | Activity kembali menampilkan user interface setelah status stop. |
| onDestroy() | Dipanggil saat aplikasi benar-benar berhenti. |
| onSaveInstanceState(Bundle) | Method ini mengijinkan activity untuk menyimpan setiap status intance. Misalnya dalam mengedit teks, kursor bergerak dari kiri ke kanan. |

| | |
|------------------------------------|---|
| onRestoreInstanceState (Bundle) | Diapanggil saat activity kembali meng-inisialisasi dari status sebelumnya yang disimpan oleh onSaveInstanceState(Bundle). |
|------------------------------------|---|

Sumber: Arif A.H, 2012

a. Activity

Normalnya setiap activity menampilkan satu buah *user interface* kepada pengguna. Misalnya sebuah activity menampilkan daftar menu minuman, kemudian pengguna dapat memilih satu jenis minuman. Contoh lainnya pada aplikasi SMS, dimana satu activity digunakan untuk menulis pesan, activity berikutnya untuk menampilkan nomor kontak tujuan, atau activity lainnya digunakan untuk menampilkan pesan-pesan lama. Meskipun activity-activity diatas terdapat dalam satu aplikasi sms, namun masing-masing activity berdiri sendiri. Untuk pindah dari satu activity ke activity lainnya dapat melakukan suatu event misalnya tombol diklik atau melalui trigger tertentu.

b. Service

Service tidak memiliki *user interface*, namun berjalan di belakang layar. Misalnya music player, sebuah activity digunakan untuk memilih lagu kemudian di-play. Agar music player bisa berjalan dibelakang aplikasi lain maka harus menggunakan service.

c. Intents

Intens adalah mekanisme untuk menggambarkan sebuah action secara detail seperti bagaimana cara mengambil sebuah foto.

d. Content Providers

Menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu activity, misalnya kita menggunakan applikasi berbasis peta (MAP). Activity membutuhkan

cara untuk mengakses data kontak untuk prosedur navigasi. Disinilah peran content providers.

e. Resource

Resource digunakan untuk menyimpan file-file non-coding yang diperlukan pada sebuah aplikasi misalnya file icon, file gambar, file audio, file video atau yang lain. Gambar berformat JPG atau PNG sebuah aplikasi biasanya disimpan dalam folder `res/drawable`, icon aplikasi disimpan dalam `res/drawable-ldpi` dan file audio disimpan dalam folder `res/raw`. File XML untuk membentuk sebuah *user interface* disimpan dalam folder `res/layout`.

Android memungkinkan penggunaannya untuk memasang aplikasi pihak ketiga, baik yang diperoleh dari toko aplikasi seperti Google Play, Amazon Appstore, ataupun dengan mengunduh dan memasang berkas APK dari situs pihak ketiga. Aplikasi Android dikembangkan dalam bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak Android (SDK). SDK ini terdiri dari seperangkat perkakas pengembangan, termasuk debugger, perpustakaan perangkat lunak, emulator handset yang berbasis QEMU, dokumentasi, kode sampel, dan tutorial. Didukung secara resmi oleh lingkungan pengembangan terpadu (IDE) Eclipse, yang menggunakan plugin Android Development Tools (ADT). Perkakas pengembangan lain yang tersedia di antaranya adalah Native Development Kit untuk aplikasi atau ekstensi dalam C atau C++, Google App Inventor, lingkungan visual untuk pemrogram pemula, dan berbagai kerangka kerja aplikasi web seluler lintas platform.

2.4.1 Perkembangan Android

1. Android pertama kali dikembangkan oleh sebuah perusahaan bernama android inc.
2. Kemudian pada tahun 2005, Google mengakuisisi perusahaan ini sehingga industri IT ketika itu beranggapan akan muncul istilah iPhone dengan langkah Google tersebut.

3. Pada tahun 2007, Google dan beberapa perusahaan bergabung dengan Open Handset Alliance (Intel, Nvidia, Texas Instrument) mengemb angkkan sistem operasi android dan resmi menjadi open source.
4. Pada tahun 2008, android SDK 1.0 diluncurkandan phone G1 yang diproduksi oleh HTC menggunakan sistem operasi tersebut.
5. Pada tahun 2009, versi terbaru dari sistem android diluncurkan mulai dari versi 1.5 (Cupcake), versi 1.6 (Donut), dan versi 2.0/2.1 (Eclair). Hal ini didukung dengan lebih dari 20 gadget yang menggunakan versi tersebut.
6. Pada tahun 2010 android menjadi sistem operasi blackberry dan mnejadi sistem operasi terbaik pada platform smatphone. Versi 2.2 (Froyo) diluncurkan dan lebih dari 60 gadget menggunakannya.
7. Pada tahun 2011, versi 2.3 (Gingerbread) dan 3.0 (Honeycomb) berturut-turut diluncurkan.
8. Pada November 2011 Android 4.0. (Ice Cream Sandwich) dikenalkan, Android 4.0 ini adalah versi terakhir yang mendukung Flash Player Adobe Systems.
9. Pada tanggal 27 Juni 2012 Android 4.1 Jelly Bean dikenalkan, Jelly Bean adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuka pengguna (UI).Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.
10. Pada tanggal 13 November 2012 dirilis Android 4.2 Jelly Bean, yang pertama kalai digunakan pada Nexus 4 LG dan Nexus 10 Samsung.
11. Pada tanggal 22 Agustus 2013 dirilis Android Jelly Bean 4.3 pada 24 Juli 2013 di San Francisco. Kebanyakan perangkat Nexus menerima pembaruan dengan segera. Nexus 7 generasi kedua adalah perangkat pertama yang menggunakan system operasi ini.
12. Pada tanggal 31 Oktober 2013 dirilis Android 4.4 KitKat (dinamai dengan izin dari Nestlé dan Hershey).

Sehingga saat ini, sistem operasi yang ada di segment pasar smartphone ada 4 macam, yaitu:

- a. Blackberry dengan sistem operasi Blackberry
- b. iPhone dengan sistem operasi MacOS
- c. Nokia dengan operasi sistem Symbian
- d. Berbagai phone dengan sistem Android

Android merupakan sistem operasi *open source* maka sangat memungkinkan untuk para pengembang aplikasi menggunakan sistem operasi android untuk melakukan pengembangan aplikasi yang telah ada. Dalam pengembangan aplikasi android biasanya development android menggunakan eclipse *sebagai integrated development environment (IDE)*.

IDE merupakan program komputer yang memiliki beberapa fasilitas yang diperlukan dalam membangun suatu perangkat lunak. Eclipse tersedia secara bebas untuk merancang *developer* android. Selain itu eclipse juga menjadi sebagai pengembang android, membuat *project* android dimana *source software* langsung dari situs resmi google. Namun, selain eclipse untuk melakukan pengembangan aplikasi diperlukan juga android *software development kit (Android SDK)*, *Java Development kit (JDK)*, Java , PHP, MySQL, dan juga photoshop untuk membuat desain pada aplikasi yang akan dibuat (Edy,dkk,2011).

Antar muka pengguna Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk manipulasi objek di layar. Android adalah system operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2012, ada sekitar

700.000 aplikasi yang tersedia untuk Android, dan sekitar 25 juta aplikasi telah diunduh dari Google Play, took aplikasi utama Android. Sebuah survey pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi seluler.

Faktor-faktor diatas telah memberikan kontribusi terhadap perkembangan android, menjadikannya sebagai sistem operasi telepon pintar yang paling banyak digunakan didunia. Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikostumisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Akibatnya, meskipun pada awalnya sistem operasi ini dirancang khusus untuk telepon pintar dan tablet, Android juga dikembangkan menjadi aplikasi tambahan ditelevisi, konsol permainan, kamera digital, dan perangkat elektronik lainnya. Sifat android yang terbuka telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain.

Tabel berikut ini menunjukkan versi pengembangan sistem operasi Android dan perbandingan fitur-fitur berbagai versi android:

Tabel 2.5 Pengembangan sistem operasi android dan perbandingan fitur-fitur berbagai versi android

| Versi | Gambar | Fitur-fitur |
|-------|--------|-------------|
|-------|--------|-------------|

| | | |
|---|---|---|
| <p>Android versi 1.1</p> |  | <p>Keluaran Android pertama versi 1.1 keluar pada 9 Maret 2009 oleh Google. Android versi ini dilengkapi dengan fitur yang disupport oleh Google Mail Service dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.</p> |
| <p>Android cupcake (versi 1.5)</p> |  | <p>Android Cupcake Versi 1.5 merupakan versi kedua, android ini mendukung berbagai fitur yang lebih lengkap diantaranya adalah upload Video ke Youtube atau Gambar ke Picasa langsung dari telepon selular. Bluetooth A2DP yang sudah terintegrasi.</p> |
| <p>Android donut (versi 1.6)</p> |  | <p>Versi Android 1.6. Android Donut lebih mengembangkan aplikasi – aplikasi standar diantaranya proses searching yang lebih baik, Fitur pada galery yang lebih "user friendly", Mendukung Resolusi Layar WVGA, Peningkatan Android Market dan Aplikasi , juga mampu Upgradeable ke versi 2.1 (Eclair).</p> |
| <p>Android eclair (versi 2.0 / 2.1)</p> |  | <p>Android Eclair yang merupakan urutan generasi ke 4 dari banyak versi Android yang ada, terdapat penambahan google map di versi ini., juga beberapa penambahan fitur yang lebih canggih dan bagus seperti Kamera 3,2 Megapixel yang didukung oleh "flash", Daftar kontak baru yang elegan, HTML5 telah terdukung pada perubahan UI dengan browser</p> |

| | | |
|---|---|---|
| | | baru dan terakhir Bluetooth 2.1 |
| Android frozen yogurt / froyo (versi 2.2) |  | Diluncurkan pada tahun 2010. android versi ini sudah dilengkapi dengan beragam fitur baru seperti adobe flash dan diklaim memiliki kecepatan 3 sampai 5 kali lebih cepat dari versi terdahulunya. Sedangkan fitur terbarunya adalah mampu merekam video dengan HD Quality, Bisa meletakkan aplikasi di dalam MMC/SD Card, Bisa untuk dijadikan Hotspot, Performa yang meningkat, Kemampuan auto update dalam Android Market |
| Android gingerbread (versi 2.3) |  | Android versi 2.3 Gingerbread mampu meningkatkan kinerja dan peforma berbagai macam aplikasi atau fitur - fitur yang umum dalam device android seperti game, audio, video, kamera dll. Fungsi dan penerapan copy paste juga telah dioptimalkan. Android gingerbread juga sudah mendukung User interface hemat energi, Keyboard virtual dengan word selection, Power Management, App control |
| Android honeycomb (versi 3.0 / 3.1) |  | Android versi 3.0/3.1 Honeycomb adalah salah satu versi android yang dikhususkan bagi pengguna tablet, os android versi ini tidak bias digunakan pada ponsel/hp anda. Fitur- fiturnya juga telah disesuaikan bagi pengguna tablet. Sistem ini didesain khusus dengan kecanggihannya membuat tablet anda berjalan |

| | | |
|--|---|--|
| | | dengan kinerja maksimal. |
| Android ice cream sandwich / ics (versi 4.0) |  | Android versi 4.0 Ice Cream Sandwich atau disingkat ICS adalah android pertama yang mempunyai fitur baru membuka kunci dengan pengenalan wajah. Android ini juga diklaim mampu menghasilkan interface yang lebih halus dan bersih. Terdapat juga beberapa penambahan font baru pada android versi ICS ini. |
| Android jelly bean (versi 4.1) |  | Jelly Bean-Android versi 4.1 yang diluncurkan pada acara Google I/O membawa fitur-fitur baru yang menawan, beberapa fitur yang diperbaharui dalam system operasi ini antara lain, pencarian dengan menggunakan Voice Search yang lebih cepat, informasi cuaca, lalu lintas. |
| Android jelly bean (versi 4.2) |  | Android versi 4.2 diklaim lebih pintar dan inovatif dibandingkan dengan pendahulunya. Fitur yang diperbaharui dalam system operasi ini antara lain Notifications, Google Assistant, Face unlock dengan Liveness Check, Barrel Roll, Smart Widget, Google Now, Teknologi Project Butter untuk meningkatkan responsifitas yang sangat baik, Sistem operasi yang cepat dan Ringan, Full Chrome browser menjejalah internet menggunakan Google Chrome seperti pada PC. |

| | | |
|---------------------------------------|--|---|
| <p>Android jelly bean (versi 4.3)</p> |  | <p>ANDROID JELLY BEAN Teknologi smart bluetooth yang membuat Bluetooth aktif tanpa menguras baterai, Mendukung Open GL ES 3.0, performa grafis yang lebih bagus dan realistis. Kecepatan kinerja yang luar biasa cepat serta grafis memukau atau lebih halus.</p> |
| <p>Android kitkat (versi 4.4)</p> |  | <p>Fitur-fitur barunya antara lain Fitur SMS yang terintegrasi langsung kedalam Aplikasi Google Hangouts. Terdapat fasilitas Cloud Printing, dimana pengguna dapat Printing secara nirkabel / mengirim perintah ke Laptop / PC yang terhubung dengan printer. Desain ikon dan tema yang lebih unik dan realistik. Mendengarkan perintah suara dari Google Now tanpa menguras daya baterai. Navigasi dan status bar yang mengalami pembaruan. Interface yang sangat halus. Bisa mengakses aplikasi kamera dari layar yang terkunci..</p> |

Sumber: Wahyu A, 2012

Perkembangan sistem operasi android cukup pesat belakangan ini, terlihat dengan dukungan beberapa vendor yang mengeluarkan produksinya menggunakan sistem operasi android. Terlebih dengan versi terbaru android kitkat yang digunakan untuk smartphone juga tablet.

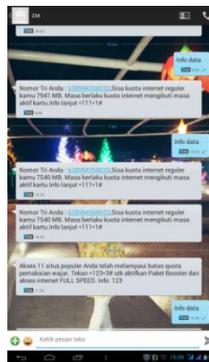
Berikut ini secara umum, antar muka yang biasanya ada pada sistem android, yaitu:

1. *Home screen*, antar muka pertama pada sistem android yang berisi widget dan wallpaper



**Gambar 2.5 Home screen
(Dokumentasi Pribadi)**

2. *Messaging*, berfungsi untuk mengirim dan menerima SMS (*short message service*) yang ditampilkan dalam bentuk percakapan, sehingga kita bisa mengetahui *history* pesan sms sebelumnya.



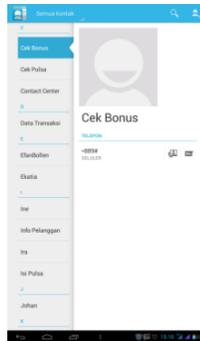
**Gambar 2.6 Messaging
(Dokumentasi Pribadi)**

3. *Dialler*, berfungsi untuk melakukan panggilan dan penerimaan telepon.



**Gambar 2.7 Dialler
(Dokumentasi Pribadi)**

4. *Contact*, berfungsi untuk menyimpan nama dan nomor kontak.



**Gambar 2.8 *Contact*
(Dokumentasi Pribadi)**

5. *Setting*, untuk melakukan *setting device* yang akan digunakan.



**Gambar 2.9 *Setting*
(Dokumentasi Pribadi)**

6. *Silding Drawer*, berfungsi sebagai antar muka untuk menampilkan aplikasi yang telah terinstal. Anda bisa menggunakan touch-screen atau bantuan tangan untuk beralih halaman.



**Gambar 2.10 *Silding drawer*
(Dokumentasi Pribadi)**

2.4.2 Kelebihan Android

Ada beberapa alasan, yang menjadikan dasar mengapa kita memilih pemrograman android, antara lain:

1. Bersifat *open-source*, sehingga dapat mengkostumisasi aplikasi yang berbasis android, dapat mempelajari bahkan membuat sendiri aplikasinya, tanpa harus membayar sejumlah uang tertentu.
2. Implementasinya yang lintas platform. Karena ditulis dengan bahasa pemrograman java, maka dapat dijalankan pada berbagai macam platform, dengan menyertakan java virtual machine yang disebut dengan dalvik virtual machine.
3. Android juga menyediakan SDK dan IDE yang semuanya gratis, sehingga semakin memudahkan kita ketika akan men-devel aplikasi.
4. Dukungan puluhan atau bahkan ratusan aplikasi yang gratis atau berbayar. Bahkan google menyediakan wadah yang para pengguna android dapat *download* dan instal secara langsung aplikasinya, yaitu di android market atau sekarang lebih dikenal *play store*. Didalamnya, terdapat berbagai macam aplikasi yang semuanya telah diseleksi oleh google, dan telah layak digunakan di pasaran. Berbagai aplikasi ditawarkan berdasarkan kategori, dapat kita pilih dalam situs *play store* (Edy,dkk,2011).

2.5 Peranti Pengembangan Aplikasi Android

Pertumbuhan pasar perangkat berbasis Android tentu saja mendorong pertumbuhan pengembangan aplikasi berbasis Android. Bagi sisi pengembang, peranti yang memudahkan pembuatan aplikasi tentu saja diharapkan. Situs Android Developers menyediakan Android SDK (*Software Development Kit*) yang memudahkan siapa pun untuk membuat aplikasi android. Aplikasi yang diperlukan untuk pengembangan aplikasi Android mencakup *Java Development Kit* (JDK), Eclipse, Android SDK, dan *Android Development Tools* (ADT) (Abdul K, 2013:3).

2.5.1 JDK (*Java Development Kit*)

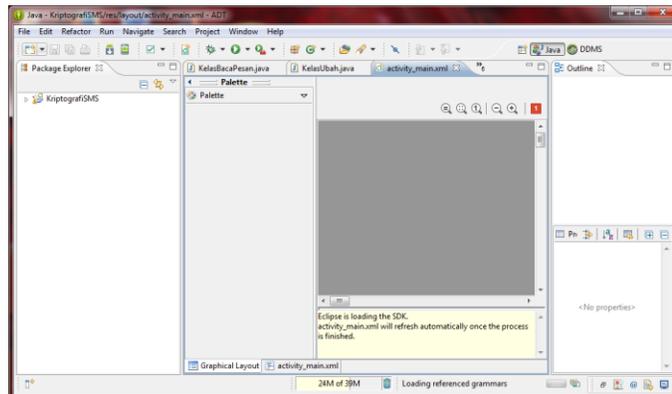
Java Development Kit (JDK) adalah perangkat pengembangan aplikasi java. Perangkat ini mutlak diperlukan untuk membuat aplikasi Android, mengingat aplikasi Android itu berbasis java. Sebagaimana diketahui, java adalah salah satu bahasa pemrograman yang bisa digunakan untuk membuat aplikasi. Namun perlu diketahui, tidak semua pustaka dalam java digunakan di Android. Sebagai contoh, android tidak menggunakan Swing (Abdul K, 2013:4).

JDK adalah semacam kotak peralatan (kit) yang digunakan untuk development. JDK ini berguna untuk menulis kode program saat pembuatan aplikasi baru ataupun pengembangan aplikasi. JDK memuat program-program dan library yang kita butuhkan untuk meng-*compile* dan me-*launch* program Java. Dua program utama yang ada di JDK adalah javac dan java.

Android ini sebenarnya berbasis bahasa java maka kita memerlukan jdk untuk men-*develop* aplikasi android. JDK yang sudah termasuk didalamnya JRE (*Java Runtime Environment*) untuk bisa menjalankan program yang dibuat. (Budi,dkk,2010:12).

2.5.2 Eclipse

Eclipse adalah perangkat pengembangan aplikasi yang tergolong sebagai IDE (*Integrated Development Environment*) karena menyediakan berbagai fasilitas untuk pembuatan aplikasi. Perangkat lunak ini dapat digunakan sebagai peranti pengembangan aplikasi yang menggunakan bahasa seperti java, C++ dan Python (Abdul K, 2013:5). Dengan menggunakan IDE inilah aplikasi android dibangun.



Gambar 2.11 Tampilan eclipse yang dapat digunakan untuk membuat aplikasi android (Dokumentasi Pribadi)

Berikut ini adalah sifat dari Eclipse:

1. *Multi-platform*: Target system operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Multilanguage*: 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.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bias digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak pengembangan IBM Visual Age for Java 4.0. Produk eclipse ini diluncurkan oleh IBM pada tanggal 5 November 2001. Secara standar eclipse selalu dilengkapi dengan JDT (*Java Development Tools*) plug-in yang membuat eclipse kompatibel untuk mengembangkan program java, dan PDE (*Plug-In Development Environment*) untuk mengembangkan plug-in baru. Eclipse beserta *plug-in* nya di implementasikan dalam bahasa pemrograman java. Eclipse tidak saja untuk mengembangkan program java, tetapi untuk berbagai macam keperluan. Perluasan apapun cukup dengan

menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ maka telah terdapat plug-in CDT (C/C++ Development Tools) yang dapat dipasang di eclipse untuk eclipse menjadi perangkat untuk pengembangan C/C++. Pengembangan secara visual bukan hal yang tidak mungkin oleh eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah *kernel*, yang mengangkat *plug-in*. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari *plug-in* yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan *Rich Client Platform* (RCP). Berikut ini adalah komponen yang membentuk RCP:

- *Core platform*
- OSGi
- SWT (*Standard Widget Toolkit*)
- JFace
- *Eclipse Workbench*

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam

keperluan, cukup dengan menginstal *plug-in* yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya. Salah satu situs yang menawarkan *plug-in* secara gratis seperti *Eclipse downloads by project*.



Gambar 2.12 Simbol eclipse
([https://en.wikipedia.org/wiki/Eclipse_\(software\)#/media/File:Eclipse-logo-2014.svg](https://en.wikipedia.org/wiki/Eclipse_(software)#/media/File:Eclipse-logo-2014.svg), 17/05/2016)

2.5.3 Android SDK

Android SDK adalah kumpulan software yang berisi mengenai pustaka, *debugger* (alat pencari kesalahan program), emulator (peniru perangkat bergerak), dokumentasi, kode contoh, dan panduan. Keberadaan emulator dapat membuat dan menguji aplikasi Android, tanpa harus mempunyai perangkat keras berbasis Android. Pembuatan dan pengujian aplikasi Android dapat dilakukan di computer yang tidak berbasis Android bahkan tidak hanya dapat memnguji di Windows, tetapi juga di *platform* lain seperti Mac dan Linux (Abdul Kadir, 2013:5).



Gambar 2.13 Contoh tampilan emulator yang berguna untuk menguji aplikasi android yang sedang dibuat
(Arif A.H, 2012)

Android SDK adalah tools API (Application Programming Interface) yang digunakan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google. Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java sebagai platform aplikasi netral, android memberi kesempatan untuk membuat aplikasi yang dibutuhkan yang bukan merupakan aplikasi bawaan handphone/smartphone (Budi,dkk,2010:12).

2.5.4 Android Development Tools (ADT)

Android Development Tools (ADT) adalah *plugin* untuk Eclipse IDE yang memungkinkan Eclipse digunakan untuk mengembangkan aplikasi android. ADT inilah yang membuat pengembangan aplikasi Android dapat dilakukan dengan mudah (Abdul Kadir, 2013:5).

2.6 Java

Java adalah salah satu bahasa pemrograman yang bersifat *open source* yang merupakan produk dari Sun Microsystem dan sekarang di pegang oleh Oracle. Bahasa Java adalah bahasa modern yang telah diterima masyarakat komputasi dunia. Hampir semua perusahaan perangkat lunak dan komputer besar mendukung dan mengembangkan aplikasi system berbasis Java. Java adalah salah satu bahasa pemrograman *Multiplatform* (bisa berjalan di berbagai macam system operasi) karena pada dasarnya java mempunyai Jre (*java runtime environment*) atau dapat kita artikan sebagai mesin tersendiri untuk mengeksekusi *binary code* hasil dari kompilasi program yang telah kita buat, berbeda dengan bahasa pemrograman vb, C++ yang memanfaatkan komponen sistem dalam Windows untuk mengeksekusi *binary code* hasil kompilasi program.

Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai mesin virtual java (JVM).

Java merupakan bahas pemrograman yang bersifat umum/non-sfesifik (general) purpose, dan secara khusus di desain untuk memanfaatkan dependensi miplementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa *platform* sistem operasi yang berbeda. Saat ini Java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web (Gianto W, 2011:5).



**Gambar 2.14 Logo program java
(Android Programming with Eclipse, 2013:6)**

Pada Tahun 1996, Sun Microsystem secara resmi merilis versi awal Java yang kemudian terus berkembang hingga muncul JDK 1.1 (*Java Development Kit versi 1.1*). Perkembangan terus dilakukan hingga muncul versi baru yang disebut Java 2. Perubahan utama antara versi sebelumnya adalah adanya Swing yang merupakan teknologi *Graphical User Interface* (GUI) yang mampu menghasilkan aplikasi *desktop* yang benar-benar baik. Untuk mengembangkan aplikasi berbasis java, kita memerlukan *Java Development Kit* (JDK), saat ini java telah mencapai versi 2 yang kenal dengan java 2 platform. Java 2 platform, dibagi menjadi tiga kategori yaitu:

- a. Java 2 Standart Edition (J2SE)
- b. Java 2 Enterprice Edition(J2EE)
- c. Java 2 Micro Edition(J2ME)

Karakteristik Java antara lain :

- a. Sederhana (Simple)
- b. Berorientasi Objek (Object Oriented)
- c. Terdistribusi (Distributed)
- d. Interpreted
- e. Robust
- f. Aman (Secure)
- g. Architecture Neutral
- h. Portable
- i. Performance

j. Multithreaded

k. Dinamis

Java mempunyai platform yaitu :

- a. Java Application Programming Interface (Java API)
- b. Java Virtual Machine (Java VM)

2.6.1 Java API

Java API terdiri dari tiga bagian utama :

1. Java Standard Edition (SE), sebuah standar API untuk merancang aplikasi desktop dan applets dengan bahasa dasar yang mendukung grafis, M/K, keamanan, konektivitas basis data dan jaringan.
2. Java Enterprise Edition (EE), sebuah inisiatif API untuk merancang aplikasi server dengan mendukung untuk basis data.
3. Java Micro Edition (ME), sebuah API untuk merancang aplikasi yang jalan pada alat kecil seperti telepon genggam, computer genggam dan pager.

2.6.2 Java Virtual Machine

Java Virtual Machine (JVM) adalah sebuah spesifikasi untuk sebuah computer abstrak. JVM terdiri dari sebuah kelas pemanggil memanggil file, class dari kedua program Java mungkin sebuah perangkat lunak interpreter Java. Interpreter Java mungkin sebuah perangkat lunak interpreter yang menterjemahkan satu kode byte pada satu waktu, atau mungkin sebuah just-in-time (JIT) compiler yang menurunkan byte code arsitektur netral kedalam bahasa mesin untuk host komputer.