

**RANCANG BANGUN APLIKASI KRIPTOGRAFI BERBASIS
ANDROID DENGAN MENGGUNAKAN ALGORITMA
*BLOWFISH***

SKRIPSI



**Disusun Oleh:
Mochamad Chakimin
09.18.140**



**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNIK INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014**

LEMBAR PERSETUJUAN DAN PENGESAHAN
RANCANG BANGUN APLIKASI KRIPTOGRAFI BERBASIS
ANDROID DENGAN MENGGUNAKAN ALGORITMA
BLOWFISH

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Komputer Strata Satu (S-1)*

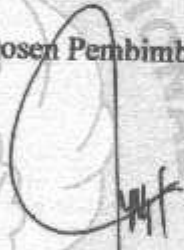
Disusun Oleh :
Mochamad Chakimin
09.18.140

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II


Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002


Yosep Agus Pranoto, ST
NIP. P. 1031000432

Program Studi Teknik Informatika S-1
Ketua




Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
RANCANG BANGUN APLIKASI KRIPTOGRAFI BERBASIS ANDROID
DENGAN MENGGUNAKAN ALGORITMA *BLOWFISH***

Mochamad Chakimin (0918140)

Jurusan Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jln.Raya Karanglo Km 2 Malang
Email: hakam_hakim@rocketmail.com

Abstrak

Teknologi kini telah mengalami perkembangan yang sangat pesat. Teknologi sudah mampu mengembangkan beberapa aplikasi di dunia, khususnya perkembangan teknologi *mobile*. Android adalah sistem operasi yang kini menjadi target utama bagi para pengembang untuk menciptakan beberapa aplikasi *mobile*. Salah satu fasilitas *mobile* yang sangat populer dan banyak digemari masyarakat adalah SMS (*Short Message Service*). Fasilitas SMS dapat digunakan seseorang untuk menyampaikan pesan singkat dan padat dalam bentuk tulisan dengan media *mobile*. Aplikasi SMS sendiri memiliki banyak manfaat yang bisa dipergunakan untuk kepentingan di semua bidang. SMS dapat dijadikan sarana untuk menyampaikan informasi yang lebih praktis. Pada kesempatan ini penulis mencoba untuk menerapkan teknologi SMS dengan suatu algoritma untuk pengamanan pesan pada sebuah *mobile* berbasis android.

Aplikasi *Kriptografi* merupakan aplikasi *enkripsi* data dengan menggunakan Algoritma *Blowfish* dengan mengubah dokumen berformat *txt* kedalam dokumen terenkripsi dengan format *.enc* sehingga isi dokumen tidak terbaca selain itu hasil pengubahan kembali dokumen yang terenkripsi kedokumen aslinya isi dokumen tetap sama dan tidak berubah dari dokumen aslinya.

Kata Kunci: SMS (*short message service*), android, enkripsi, Algoritma *Blowfish*.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas karunia, rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan penelitian yang berjudul “**RANCANG BANGUN APLIKASI KRIPTOGRAFI BERBASIS ANDROID DENGAN MENGGUNAKAN ALGORITMA *BLOWFISH***” Skripsi ini dapat terselesaikan tidak terlepas dari dukungan berbagai pihak. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Ir. Soeparno Djiwo, MT. selaku Rektor ITN Malang.
2. Bapak Ir. H. Anang Subardi, MT. selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Bapak Joseph Dedy Irawan, ST, MT selaku Kepala Program Studi Teknik Informatika S-1 Institut Teknologi Nasional Malang.
4. Bapak Joseph Dedy Irawan, ST, MT selaku Dosen Pembimbing I yang telah memberikan saran dan bimbingannya dalam penyusunan laporan ini.
5. Bapak Yosep Agus Pranoto, ST. selaku Dosen Pembimbing II yang telah memberikan saran dan bimbingannya dalam penyusunan laporan ini.
6. Bapak dan Ibu Dosen yang telah mengajar penulis selama studi di Institut Teknologi Nasional Malang.
7. Rekan-rekan Teknik Informatika Institut Teknologi Nasional Malang serta berbagai pihak yang turut membantu dalam penyelesaian laporan ini.
8. Untuk Orang Tua dan Keluarga yang telah mendukung penulis sampai sekarang.

Semoga apa yang telah disajikan dapat memberikan manfaat dan pengetahuan bagi para pembaca. Segala kritik dan saran yang bersifat membangun, diterima dengan senang hati sebagai tambahan ilmu pengetahuan.

Malang, 15 Agustus 2013

Penulis

DAFTAR ISI

	Halaman
Halaman Judul	1
Lembar Persetujuan dan Pengesahan	ii
Lembar Keaslian	iii
Abstrak	iv
Kata Pengantar	v
Daftar Isi	vi
Daftar Gambar	ix
Daftar Tabel	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian.....	3
1.5.1 <i>Library research</i>	3
1.5.2 Perancangan Sistem	3
1.5.3 Implementasi	3
1.5.4 Verifikasi	3
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Kriptografi	5
2.2 Algoritma Sandi	6
2.3 Algoritma <i>Blowfish</i>	8
2.4 Struktur Algoritma <i>Blowfish</i>	9
2.4.1 Enkripsi Algoritma <i>Blowfish</i>	9
2.5 SMS (<i>Short Message Servive</i>)	11
2.6 Sistem Operasi Android	12
2.6.1 Produk Awal	13
2.6.2 Android Versi 1.1	13

2.6.3	Android Versi 1.5 (Cupcake)	13
2.6.4	Android Versi 1.6 (Donut)	14
2.6.5	Android Versi 2.0/2.1 (Éclair)	14
2.6.6	Android Versi 2.2 (Froyo : Frozen Yoghurt)	15
2.6.7	Android Versi 2.3 (Gingerbread)	15
2.6.8	Android Versi 3.0/3.1 (Honeycomb)	15
2.6.9	Android Versi 4.0 (ICS: Ice Cream Sandwich)	15
2.6.10	Android Versi 4.1 (Jelly Bean)	16
2.6.11	Android Versi 4.2 (Jelly Bean)	16
2.7	Intens	17
2.8	Java	17
2.9	Eclipse (IDE)	18
2.10	YouWave	20
2.10.1	Fitur YouWave Android	20
2.10.2	Kebutuhan Minimal Sistem	21
BAB III ANALISA DAN PERANCANGAN		22
3.1	Analisa Sistem	22
3.1.1	Identifikasi Masalah	22
3.1.2	Analisis Kebutuhan Hardware dan Software	23
3.1.2.1	Komponen Hardware	23
3.1.3	Sumber Informasi	24
3.2	Perancangan Sistem	24
3.2.1	Use Case Diagram	24
3.2.2	Flowchart Prosedur Enkripsi dan Deskripsi	24
3.3	Flowchart Sistem	28
3.4	Perancangan Antar Muka	29
3.4.1	Rancangan Splash Screen	30
3.4.2	Rancangan Menu Utama	30
3.4.3	Rancangan Menu Tulis Pesan	31
3.4.4	Rancangan Menu Kotak Masuk	31

3.4.5 Rancangan Menu Tentang	32
3.4.6 Rancangan Menu Bantuan	32
3.4.7 Rancangan Menu Keluar	33
BAB IV HASIL IMPLEMENTASI DAN PENGUJIAN SISTEM	34
4.1 Implementasi Sistem	34
4.1.1 Hasil Implementasi Splash Screen	34
4.1.2 Hasil Implementasi Menu Utama	35
4.1.3 Hasil Implementasi Menu Tulis Pesan	35
4.1.4 Hasil Implementasi Menu Kotak Masuk	36
4.1.5 Hasil Implementasi Menu Tentang	36
4.1.6 Hasil Implementasi Menu Bantuan	37
4.1.7 Hasil Implementasi Menu Keluar	38
4.2 Pengujian	38
4.2.1 Pengujian Fungsional Sistem.....	38
4.2.2 Pengujian Tulis Pesan	39
4.2.3 Pengujian Kotak Masuk	39
4.2.4 Pengujian Pesan Terenkripsi	40
4.3 Pengujian Terhadap Versi Operating Sistem Android Lain	40
4.4 Pengujian Terhadap Berbagai Ukuran Layar	41
BAB V Penutup	42
5.1 Kesimpulan	42
5.2 Saran	42
Daftar Pustaka.....	44
Lampiran	45

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Kriptografi Kunci Simetris	7
Gambar 2.2 Jaringan Fiestel Untuk Algoritma <i>Blowfish</i>	10
Gambar 2.3 Fungsi <i>f</i> dalam Algoritma <i>Blowfish</i>	11
Gambar 2.4 Tampilan Eqlipse Juno (IDE)	20
Gambar 2.5 Tampilan YouWave For Android	22
Gambar 3.1 Use Case Diagram Aplikasi Kriptografi	25
Gambar 3.2 Flowchart Prosedur Enkripsi	26
Gambar 3.3 Flowchart Prosedur Deskripsi	27
Gambar 3.4 Flowchart Program Aplikasi Kriptografi.....	28
Gambar 3.5 Rancangan Splash Screen	30
Gambar 3.6 Rancangan Menu Utama	30
Gambar 3.7 Rancangan Menu Tulis Pesan	31
Gambar 3.8 Rancangan Menu Kotak Masuk	31
Gambar 3.9 Rancangan Menu Tentang	32
Gambar 3.10 Rancangan Menu Bantuan	32
Gambar 3.11 Rancangan Menu Keluar	33
Gambar 4.1 Tampilan Splash Screen	34
Gambar 4.2 Tampilan Menu Utama	35
Gambar 4.3 Tampilan Tulis Pesan	35
Gambar 4.4 Tampilan Kotak Masuk	36
Gambar 4.5 Tampilan Menu Tentang	37
Gambar 4.6 Tampilan Menu Bantuan	37
Gambar 4.7 Tampilan Menu Keluar	38
Gambar 4.8 Tampilan Pengujian Tulis Pesan	39
Gambar 4.9 Tampilan Pengujian Kotak Masuk	40
Gambar 4.10 Tampilan Pengujian Pesan Terenkripsi	40

DAFTAR TABEL

	Halaman
Tabel 4.1 Tabel Pengujian Fungsional Sistem	38
Tabel 4.2 Tabel Pengujian Terhadap Versi Android	40
Tabel 4.3 Tabel Pengujian Terhadap Berbagai Ukuran Layar	41

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi kini telah mengalami perkembangan yang sangat pesat. Teknologi sudah mampu mengembangkan beberapa aplikasi di dunia, khususnya perkembangan teknologi *mobile*. Android adalah sistem operasi yang kini menjadi target utama bagi para pengembang untuk menciptakan beberapa aplikasi *mobile*.

Pengguna teknologi *mobile* telah menyebar pada semua tingkatan umur, jenis kelamin, pendidikan, dan pekerjaan. Mereka menggunakan teknologi *mobile* untuk memperlancar sarana komunikasi, baik dalam urusan bisnis, politik, cinta, keluarga, perkuliahan, maupun sekedar untuk berbasa-basi. Selain berfungsi sebagai sarana komunikasi, *mobile* juga dilengkapi dengan fasilitas-fasilitas yang berfungsi sebagai penyimpan data, kalkulator, internet, radio, *game*, musik, layanan pesan singkat atau SMS (*Short Message Service*), dan sebagainya.

Salah satu fasilitas teknologi *mobile* yang sangat populer dan banyak digemari masyarakat adalah SMS. Fasilitas SMS dapat digunakan seseorang untuk menyampaikan pesan singkat dan padat dalam bentuk tulisan dengan media *mobile*. Melalui SMS pula seseorang tentunya akan menghemat biaya karena beban biaya lebih murah dibandingkan menelepon.

Dalam *kriptografi* terdapat berbagai algoritma sandi yang bisa digunakan untuk mengamankan suatu data. Salah satunya adalah algoritma *Blowfish* yang biasa digunakan untuk mengenkripsi suatu data. *Blowfish* termasuk algoritma kunci simetris yang memiliki kunci yang sama untuk mengenkripsi dan mendeskripsi sebuah data. Algoritma *Blowfish* termasuk cipher blok dan sampai saat ini masih dianggap aman karena belum ada *attack* yang benar-benar mematahkan algoritma *Blowfish*.

Pada penelitian ini akan dibuat sebuah aplikasi SMS yang disertai dengan kode keamanan pesan yang di telah enkripsi, yaitu aplikasi *Kriptografi* yang dapat membantu pengguna *mobile* untuk keamanan menyampaikan pesan penulis kepada penerima.

Metode *kriptografi* merupakan salah satu metode yang digunakan untuk meningkatkan keamanan data. *Kriptografi* menggunakan algoritma dalam melakukan proses enkripsi dan deskripsi data. Algoritma *blowfish* merupakan salah satu algoritma enkripsi data yang dapat digunakan untuk mengamankan data pada aplikasi *SMS*, sehingga data pada aplikasi *SMS* lebih terjaga keamanannya. *Algoritma Blowfish* termasuk dalam algoritma simetris, proses enkripsi dan deskripsi lebih cepat karena menggunakan kunci yang sama.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, maka penulis akan merumuskan masalah yang ada. Adapun perumusan masalah yang akan dibahas adalah :

1. Bagaimana membuat aplikasi *Kriptografi* berbasis android?
2. Bagaimana menerapkan algoritma *blowfish* pada aplikasi *Kriptografi* berbasis android?
3. Bagaimana menganalisis pengamanan pesan melalui aplikasi *Kriptografi* berbasis android?

1.3 Tujuan Penelitian

Tujuan dari penulisan ini adalah :

1. Melakukan penguncian pesan pada aplikasi *Kriptografi* dengan menggunakan algoritma *blowfish* .
2. Menganalisis pengiriman pesan untuk mengetahui proses *Kriptografi* yang digunakan dalam penyisipan teks yang dienkripsi pada algoritma *blowfish*.

1.4 Batasan Masalah

Batasan masalah yang diberikan berdasarkan permasalahan yang telah dikemukakan diatas adalah sebagai berikut:

1. Metode yang digunakan pada *kriptografi* ini adalah algoritma *blowfish*.
 2. Media yang digunakan untuk pengaman pesan adalah *mobile* berbasis android.
 3. Pesan yang disisipkan atau disembunyikan berupa pesan teks .txt dengan ukuran maksimal 200 karakter.
-

4. Software sebagai alat bantu dalam pembuatan aplikasi *mobile* ini adalah Eqlipse Juno win32 dan Youwave_Android-2-3-4.

1.5 Metode Penelitian

Metode yang digunakan oleh penulis dalam penyusunan tugas akhir ini adalah sebagai berikut :

1. *Library Research*

Library research adalah suatu metode pengumpulan data yang diperlukan dengan jalan membaca literatur dan sumber data lainnya didalam perpustakaan yang berkaitan dengan *kriptografi* yang menggunakan algoritma *Blowfish*.

2. Perancangan Sistem

Perancangan Sistem adalah suatu metode yang digunakan untuk pembuatan alur proses yang tergambar dalam bentuk diagram alir (*flowchart*) dan pembuatan rancangan antarmuka (*user interface*).

3. Implementasi

Mengimplementasikan suatu simulasi sistem yang telah terdisain ke dalam program komputer dengan menggunakan bahasa pemrograman android Eqlipse juno win32.

4. Verifikasi

Menganalisis simulasi sistem aplikasi mobile yang telah dibuat berdasarkan algoritma yang digunakan untuk mengetahui keberhasilan program yang dibuat berdasarkan teori-teori yang ada.

1.6 Sistematika Penulisan

Untuk mencapai maksud dan tujuan dari sistematika penulisan adalah untuk memperoleh suatu penyusunan masalah yang berkaitan langsung dengan yang lainnya dengan menggunakan metode penulisan sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini membahas tentang latar belakang, perumusan masalah, batasan masalah, tujuan, dan manfaat dan sistematika penulisan.

BAB II : LANDASAN TEORI

Pada bab ini dijelaskan landasan teori yang merupakan teori dasar dari teori yang dipakai untuk menyelesaikan masalah.

BAB III : ANALISA DAN PERANCANGAN

Pada bab ini berisi tentang deskripsi sistem serta desain rancangan sistem

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Pada bab ini menyajikan hasil implementasi serta pengujian mengenai cara kerja dari sistem.

BAB V : PENUTUP

Pada bab ini dijelaskan mengenai kesimpulan yang diambil serta saran-saran yang mungkin digunakan untuk mengembangkan sistem selanjutnya.

BAB II

LANDASAN TEORI

2.1 Kriptografi

Kriptografi, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita [Bruce Schneier – *Applied Cryptography*, 1994, 1998]. Selain pengertian tersebut terdapat pula pengertian ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integrasi data, serta autentikasi [A. Menezes, P. van Oorschot and S. Vanstone - *Handbook of Applied Cryptography*, October 1996]. Tidak semua aspek keamanan informasi ditangani oleh kriptografi (Ariyus, Dony, 2008). Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu:

1. Kerahasiaan, 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. Autentikasi, 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 diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
4. Non-reoudiasi, atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu :

1. Enkripsi

Enkripsi adalah merubah *plaintext* menjadi *ciphertext*. Sebuah pesan yang belum mengalami enkripsi disebut sebagai *plaintext*. Sedangkan pesan yang

sudah disandikan dengan sebuah algoritma enkripsi disebut sebagai *ciphertext*.

2. Dekripsi

Dekripsi merupakan kebalikan dari enkripsi. Dekripsi adalah proses merubah *ciphertext* menjadi *plaintext*.

3. Kunci

Yang dimaksud kunci disini adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi.

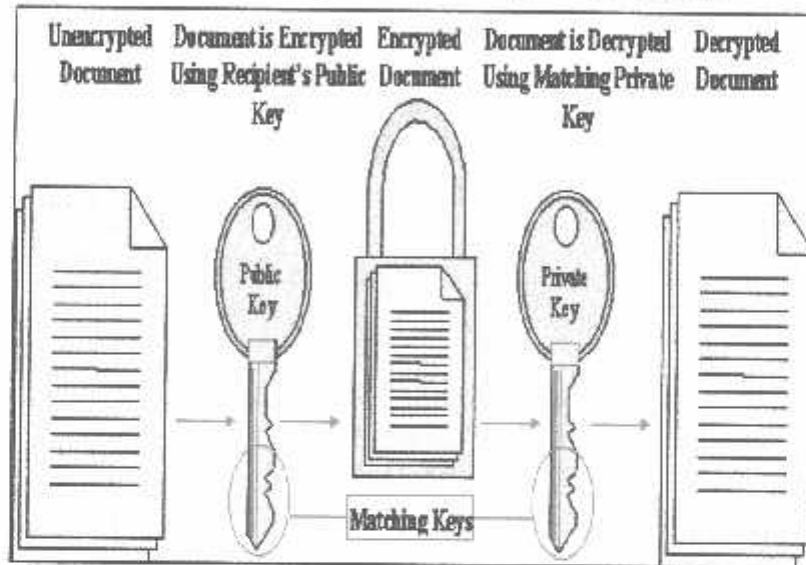
Penulis ini bertujuan untuk mempelajari algoritma Blowfish secara keseluruhan agar dapat memahami cara kerja dan struktur algoritmanya, kemudian mempelajari strategi perancangan sehingga bisa optimal. Dengan mempelajari algoritma Blowfish bisa di buat sebuah aplikasi yang memanfaatkan algoritma ini dan biasa digunakan untuk mengenkripsi data yang diinginkan.

2.2 Algoritma Sandi

Algoritma sandi adalah algoritma yang berfungsi untuk melakukan tujuan kriptografis. Algoritma tersebut harus memiliki kekuatan untuk melakukan (dikemukakan oleh Shannon): pemingungan dan peleburan, sehingga dapat digunakan untuk mengamankan informasi. Algoritma sandi yang handal adalah algoritma sandi yang kekuatannya terletak pada kunci, bukan pada kerahasiaan algoritma itu sendiri. Teknik dan metode untuk menguji kehandalan algoritma sandi adalah kriptanalisis.

Algoritma sandi berdasarkan kesamaan kunci dibagi menjadi dua, yaitu kunci asimetris dan kunci simetris. Pada sistem kunci-asimetris digunakan sepasang kunci yang berbeda, umumnya disebut kunci public (*public key*) dan kunci pribadi (*private key*), digunakan untuk proses enkripsi dan proses dekripsinya. Jadi kunci untuk membuat pesan yang disandikan berbeda dengan kunci untuk membuka pesan yang disandikan itu. Beberapa contoh algoritma yang menggunakan kunci-asimetris : Knapsack, RSA (Rivert-Shamir-Adelman), Diffie-Hellman. Sedangkan pada skema kunci-simetris, digunakan sebuah kunci rahasia yang sama untuk melakukan proses enkripsi dan dekripsinya. Algoritma sandi kunci simetris adalah jenis kriptografi yang paling umum dipergunakan. Kunci untuk membuat pesan yang disandikan

sama dengan kunci untuk membuka pesan yang disandikan itu. Siapapun yang memiliki kunci tersebut, dapat membuat dan membongkar rahasia ciphertext. Problem yang paling jelas disini terkadang bukanlah masalah pengiriman ciphertext-nya, melainkan masalah bagaimana menyampaikan kunci simetris tersebut kepada pihak yang diinginkan. Beberapa contoh algoritma yang menggunakan kunci-simetris: DES (*Data Encryption Standard*), Blowfish, Twofish, MARS, IDES, 3DES -DES diaplikasikan 3 kali, AES (*Advanced Encryption Standard*).



Gambar 2.1 Kriptografi kunci simetris

Sebuah skema algoritma sandi akan disebut kunci simetris apabila untuk setiap proses enkripsi maupun dekripsi data secara keseluruhan digunakan kunci yang sama. Skema ini berdasarkan jumlah data per proses dan alur pengolahan data didalamnya dibedakan menjadi dua kelas, yaitu *stream-cipher* dan *block-cipher*. *Stream-cipher* merupakan sebuah algoritma sandi yang mengenkripsi data persatuan data, seperti bit, byte, nibble atau per lima bit (saat data yang di enkripsi berupa data Boudout). Setiap mengenkripsi satu satuan data di gunakan kunci yang merupakan hasil pembangkitan dari kunci sebelum. *Block-cipher* adalah skema algoritma sandi yang akan membagi-bagi teks terang yang akan dikirimkan dengan ukuran tertentu (disebut blok) dengan panjang t , dan setiap blok dienkripsi dengan menggunakan kunci yang sama. Semenjak pertama kali ditemukan, telah banyak penemuan-penemuan baru dalam penerapan *cipher* blok. Salah satunya adalah algoritma Blowfish yang dirancang oleh Bruce Schneier pada tahun 1993. Sejak saat itu, telah

dilakukan berbagai macam analisis, dan perlahan-lahan mulai mendapat penerimaan sebagai algoritma enkripsi yang kuat.

2.3 Algoritma Blowfish

Blowfish atau yang disebut juga "OpenPGP.Cipher.4" adalah algoritma kunci simetrik cipher blok yang dirancang pada tahun 1993 oleh Bruce Schneier untuk menggantikan DES (*Data Encryption Standard*) (Alim Sutanto, Candra, 2010). Algoritma Blowfish dibuat untuk digunakan pada komputer yang mempunyai microprosesor besar (32-bit keatas dengan cache data yang besar). Pada saat itu banyak sekali rancangan algoritma yang ditawarkan, namun hampir semua terhalang oleh paten atau kerahasiaan pemerintah Amerika. Schneier menyatakan bahwa blowfish bebas paten dan akan berada pada domain publik. Dengan pernyataan Schneier tersebut blowfish telah mendapatkan tempat di dunia kriptografi, khususnya bagi masyarakat yang membutuhkan algoritma kriptografi yang cepat, kuat, dan tidak terhalang oleh lisensi. Blowfish dirancang dan diharapkan mempunyai kriteria perancangan yang diinginkan sebagai berikut :

1. Cepat, Blowfish melakukan enkripsi data pada microprocessor 32-bit dengan rate 26 clock cycles per byte.
2. *Compact*, Blowfish dapat dijalankan pada memory kurang dari 5K.
3. Sederhana, Blowfish hanya menggunakan operasi – operasi sederhana, Blowfish hanya menggunakan operasi – operasi sederhana, seperti : penambahan, XOR, dan lookup tabel pada operan32- bit.
4. Memiliki tingkat keamanan yang bervariasi, panjang kunci yang digunakan oleh Blowfish dapat bervariasi dan bisa sampai sepanjang minimal 32-bit, maksimal 448 -bit, Multiple 8 bit, default 128 bit.

Namun, dalam penerapannya sering kali algortima ini menjadi tidak optimal. Karena strategi implementasi yang tidak tepat. Algoritma Blowfish akan lebih optimal jika digunakan untuk aplikasi yang tidak sering berganti kunci, seperti jaringan komunikasi atau enkripsi file otomatis.

2.4 Struktur Algoritma Blowfish

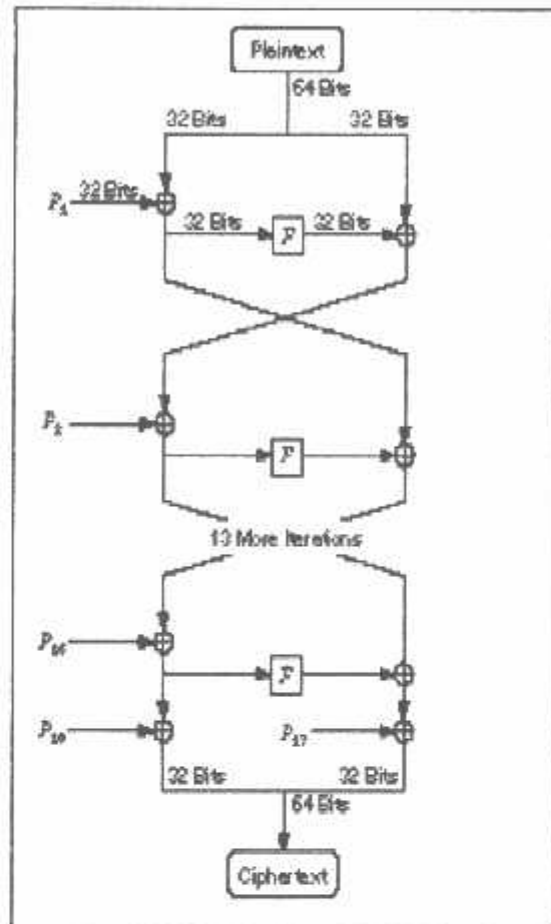
Blowfish merupakan blok chipper 64-bit dengan panjang kunci variable. Algoritma terdiri dari dua bagian: *key expansion* atau perluasan kunci dan enkripsi data. *Key Expansion* Berfungsi merubah kunci (Minimum 32-bit, Maksimum 448-bit) menjadi beberapa array sub kunci (*subkey*) dengan total 4168 byte.

2.4.1 Enkripsi Algoritma Blowfish

Blowfish menggunakan subkunci besar yang harus dihitung sebelum enkripsi dan dekripsi data. Algoritma Blowfish menerapkan jaringan Feistel (*Feistel Network*) yang terdiri dari 16 putaran. Setiap putaran terdiri dari permutasi kunci dan data dependent. Semua operasi adalah penambahan (*addition*) dan XOR pada variable 32-bit. Operasi tambahan lainnya hanyalah empat penelusuran tabelo (*table lookup*) array berindeks untuk setiap putaran. Untuk alur algoritma enkripsi dengan metode Blowfish dijelaskan sebagai berikut [Schneier, 1993]:

1. Bentuk inisial P-array sebanyak 18 buah (P1,P2,.....P18) masing-masing bernilai 32-bit. Array P terdiri dari delapan belas kunci 32-bit subkunci : P1,P2,.....,P18
 2. Bentuk S-box sebanyak 4 buah masing-masing bernilai 32-bit yang memiliki masukan 256.
Empat 32-bit S-box masing-masing mempunyai 256 entri :
S1,0,S1,1,.....,S1,255
S2,0,S2,1,.....,S2,255
S3,0,S3,1,.....,S3,255
S4,0,S4,1,.....,S4,255
 3. Plaintext yang akan dienkripsi diasumsikan sebagai masukan, Plaintext tersebut diambil sebanyak 64-bit, dan apabila kurang dari 64-bit maka kita tambahkan bitnya, supaya dalam operasi nanti sesuai dengan datanya.
 4. Hasil pengambilan tadi dibagi 2, 32-bit pertama disebut XL, 32-bit yang kedua disebut XR.
 5. Selanjutnya lakukan operasi $XL = XL \text{ xor } P_i$ dan $XR = F(XL) \text{ xor } XR$.
 6. Hasil dari operasi diatas ditukar XL menjadi XR dan XR menjadi XL.
 7. Lakukan sebanyak 16 kali, perulangan yang ke-16 lakukan lagi proses penukaran XL dan XR.
-

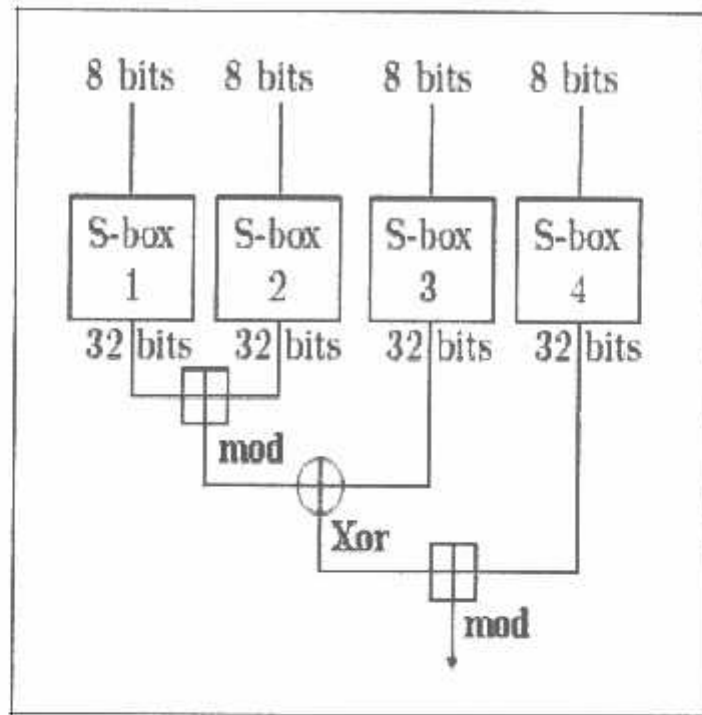
8. Pada proses ke-17 lakukan operasi untuk $XR = XR \text{ xor } P_{17}$ dan $XL = XL \text{ xor } P_{18}$.
9. Proses terakhir satukan kembali XL dan XR sehingga menjadi 64-bit kembali.



Gambar 2.2 Jaringan Feistel untuk Algoritma Blowfish

Algoritma *Blowfish* memiliki keunikan dalam proses deskripsi, yaitu proses deskripsi yang dilakukan dengan urutan yang sama persis dengan enkripsi, hanya saja pada proses deskripsi P_1, P_2, \dots, P_{18} digunakan dalam urutan terbalik.

Dalam Algoritma *Blowfish* juga terdapat fungsi f . Berikut ini merupakan gambar fungsi f tersebut.



Gambar 2.3 Fungsi f dalam Algoritma Blowfish

2.5 SMS (Short Message Service)

Short Message Service (SMS) adalah kemampuan untuk mengirim dan menerima pesan dalam bentuk teks dari dan kepada ponsel. Teks tersebut bisa terdiri dari kata-kata atau nomor atau kombinasi alphanumeric. SMS diciptakan sebagai standar pesan (*message*) oleh ETSI (*European Telecommunication Standards Institute*), yang juga membuat standar GSM yang diimplementasikan oleh semua operator GSM. SMS yang pertama dikirimkan pada Desember 1992 dari PC ke sebuah ponsel melalui jaringan GSM Vodafone di UK. Setiap Pesan maksimal terdiri dari 160 karakter jika menggunakan alphabet Latin, dan 70 karakter jika menggunakan alphabet non-Latin seperti huruf Arab atau China. *Short Message Service* atau biasa disingkat SMS merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel (*wireless*), memungkinkan dilakukannya pengiriman pesan dalam bentuk alphanumeric antara terminal pelanggan atau antar terminal pelanggan dengan sistem eksternal, seperti e-mail, paging, voice mail, dan lain-lain.

Aplikasi SMS merupakan aplikasi yang paling banyak peminat dan penggunaannya. Hal ini dapat dibuktikan dengan munculnya berbagai jenis aplikasi yang memanfaatkan fasilitas SMS. Teknologi SMS memiliki beberapa keunggulan, yaitu

harganya murah, merupakan "*Delivered Oriented Service*", artinya pesan akan selalu diusahakan untuk dikirimkan ke tujuan. Jika suatu saat nomor tujuan sedang tidak aktif atau di luar jaringan, maka pesan akan disimpan di SMSC (*SMS Center*) server dan akan dikirimkan segera setelah nomor tujuan aktif kembali. Pesan juga akan terkirim ke tujuan walaupun nomor tujuan sedang melakukan pembicaraan (sibuk). Cara kerja SMS : Saat kita menerima pesan SMS/MMS dari *handphone (mobile originated)*, pesan tersebut tidak langsung dikirimkan ke *handphone* tujuan (*mobile terminated*), akan tetapi dikirim terlebih dahulu ke *SMS Center (SMSC)* yang biasanya berada di kantor operator telepon, baru kemudian pesan tersebut diteruskan ke *handphone* tujuan. Dengan adanya SMSC, kita dapat mengetahui status dari pesan SMS yang telah dikirim, apakah telah sampai atau gagal.

2.6 Sistem Operasi Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet (Arif. 2012). Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

Pada Juli 2005, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White.

Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

2.6.1 Produk awal

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC corporation dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*mobile*) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada oktober 2008. Pada penghujung tahu 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

2.6.2 Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2.6.3 Android versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur

dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picassa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

2.6.4 Android versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada bulan September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA/ EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi *text to change spect* (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

2.6.5 Android versi 2.0/2.1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini bert hadiah \$25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

2.6.6 Android versi 2.2 (Froyo: Frozen Yoghurt)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 Java Script engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WifiHotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

2.6.7 Android versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

2.6.8 Android versi 3.0/3.1 (Honeycomb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee pad transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

2.6.9 Android versi 4.0 (ICS: Ice Cream Sandwich)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline,

dan berbagi informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

2.6.10 Android versi 4.1 (Jelly Bean)

Android Jelly Bean yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat. Tak ketinggalan Google Now juga menjadi bagian yang diperbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yakni Google Nexus 7.

2.6.11 Android versi 4.2 (Jelly Bean)

Fitur photo sphere untuk panorama, daydream sebagai screensaver, power control, lock screen widget, menjalankan banyak user (dalam tablet saja), widget terbaru. Android 4.2 Pertama kali dikenalkan melalui LG, yakni Google Nexus 4. Fitur yang tersedia di Android adalah:

- 1 Kerangka aplikasi: itu memungkinkan penggunaan dan penghapusan komponen yang tersedia.
 - 2 Dalvik mesin virtual: mesin virtual dioptimalkan untuk perangkat telepon seluler.
 - 3 Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
 - 4 SQLite: untuk penyimpanan data.
 - 5 Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
 - 6 GSM, Bluetooth, EDGE, 3G, 4G dan WiFi (tergantung piranti keras)
 - 7 Kamera, Global Positioning System (GPS), kompas, NFC dan *accelerometer* (tergantung piranti keras).
-

2.7 Intens

Intens merupakan inti aplikasi android yang dibangun. Sebuah intens dapat terdiri atas berbagai aksi, seperti view, edit, dial, dan sebagainya. Intens digunakan untuk memulai sebuah *activity* dan berinteraksi dengan komponen pada aplikasi android. Sebuah aplikasi bias jadi mengirim (*broadcast*) atau menerima (*receive*) intens.

Ketika melakukan *broadcast* intens maka akan dikirim pesan ke sistem android apa yang akan dilakukan sehingga android akan memulai *activity* baru. Jika semua penerima dapat menerima intens yang dikirim tersebut maka akan muncul pilihan menu atau aplikasi mana yang akan dijalankan.

2.8 Java

Java adalah bahasa pemrograman yang dapat dijalankan oleh berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat ini masih bergabung dengan *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin arus bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi kedalam p-code (*bytecode*) dan dapat dijalankan pada berbagai mesin virtual java (JVM).

Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*) dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya "Tulis sekali, jalankan dimana pun". 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.

2.9 Eclipse Juno (IDE)

Eclipse Integrated Development Environment (IDE) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat-sifat dari Eclipse antara lain:

- 1 Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- 2 Multi-language dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP dan sebagainya.
- 3 Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse dapat digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, uji perangkat lunak, pengembangan web dan sebagainya.

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*.

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age For Java. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US\$ 40 juta untuk pengembangannya. Semenjak itu konsorsium *Eclipse Foundation* mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

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:

- 1 *Core platform*
- 2 OSGi
- 3 SWT(*Standard Widget Toolkit*)
- 4 JFace
- 5 *Eclipse Workbench*

Secara standar Eclipse selalu dilengkapi dengan JTD(*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 UML* 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 download project*.



Gambar 2.4 Tampilan Eqlipse Juno (IDE)

2.10 YouWave

YouWave Android adalah software / aplikasi yang berguna untuk menjalankan aplikasi android di PC / Laptop kita. Software ini biasa disebut dengan *Emulator* Android. Dengan software ini anda dapat menjalankan jutaan aplikasi android dan juga anda pun dapat menambahkan / menginstall program baru yang berplatform android. Untuk anda yang ingin belajar dan mendalami tentang sistem operasi android maka Youwave adalah solusinya.

Software ini selain dapat menjalankan aplikasi yang ada di android market juga dapat menjalankan aplikasi yang telah anda buat. Jadi untuk anda yang ingin menjadi developer aplikasi android anda dapat dengan mudah mengetes aplikasi yang telah anda buat dengan Youwave Android ini. Sangat bermanfaat sekali bukan? tanpa membeli Smartphone Android yang mungkin mahal itu anda dapat menjalankan ribuan aplikasinya.

Grafis di software ini juga sudah bersahabat dan nyaman sekali untuk dipandang mata. Jadi untuk anda yang baru menggunakannya pun mungkin akan sangat cepat sekali mengerti tentang cara pemakaian software Youwave Android ini.

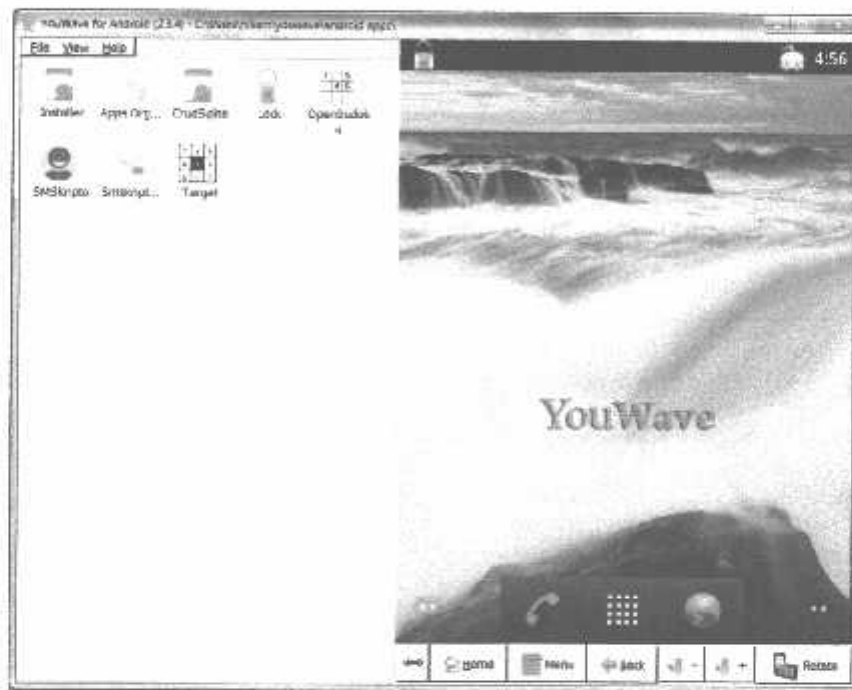
2.10.1 Fitur Youwave Android

1. Sudah Mendukung Android 2.3 Gingerbread.
 2. Simulasi fungsi SD card yaitu memungkinkan anda untuk menyimpan *game*.
 3. Memungkinkan untuk me-restart cepat.
 4. Memungkinkan multi-player untuk game online.
 5. Memungkinkan perputaran rotasi layar yang Dinamis, seperti respon instant.
 6. Ada tombol untuk mengontrol volume.
 7. Kontrol panel dapat ditarik masuk.
-

8. Dapat Berjalan pada Sistem Operasi Windows XP, Vista dan Windows 7 32/64 bit. Untuk dapat menjalankan Software Youwave Android disarankan memiliki minimum spesifikasi seperti dibawah ini

2.10.2 Kebutuhan Minimal Sistem :

1. IntelPentium1.6GHz
2. 1GBDRAM
3. 500MB diskpace
4. OS Windows XP, Vista atau Windows 7 32/64 bit.



Gambar 2.5 Tampilan YouWave For Android

BAB III

ANALISA DAN PERANCANGAN

3.1 Analisa Sistem

3.1.1 Identifikasi Masalah

Mengamankan sebuah data biasanya kita melakukannya lewat komputer dengan menggunakan software aplikasi. Hal tersebut kadang sangat kurang efisien mengingat perkembangan teknologi jaman sekarang salah satunya *handphone*, dimana didalam *handphone* sudah terdapat fitur-fitur seperti layaknya komputer atau *notebook*, misalnya: SMS (*Short Message Service*) browser, email, aplikasi *chat*, *games*, kamus, dan aplikasi lainnya.

Berdasarkan analisis diatas melalui tugas akhir ini dibuatlah rancang bangun aplikasi kriptografi berbasis android dengan menggunakan algoritma *blowfish* untuk kemudahan user mengamankan pengiriman pesan dari user lain yang tidak bertanggung jawab yang memungkinkan dilakukan dimana saja dengan menggunakan teknologi *mobile* berbasis android.

Dalam membangun sebuah aplikasi kriptografi berbasis android dengan menggunakan algoritma *blowfish* dengan menggunakan *Eclipse Juno* diperlukan beberapa tahap analisis, yaitu:

1. Menentukan masalah dan batasan masalah yang akan dibuat untuk sebuah aplikasi kriptografi berbasis android dengan menggunakan algoritma *blowfish* guna kemudahan user.
2. Mengumpulkan data yang diperlukan , yaitu berupa aplikasi yang sudah ada, buku/modul sebagai *referensi* dari masalah serta metode yang akan digunakan kedalam aplikasi.
3. Mempresentasikan pengetahuan kedalam pembuatan aplikasi dengan memanfaatkan AVD (*Android Virtual Device*) untuk menampilkan proses jalannya aplikasi.

3.1.2 Analisis Kebutuhan Hardware dan Software

Analisis kebutuhan komponen-komponen yang dibutuhkan dalam pembuatan aplikasi *mobile message lock*, dalam hal ini dibutuhkan hardware dan software, diantaranya:

3.1.2.1 Komponen Hardware:

Komputer yang digunakan:

1. Processor intel core i3 2100@3.1 Ghz
2. MotherBoard ASRock P67 Pro 3 SE
3. VGA ATI Radeon HD 4670 1Gb 128 bit
4. Memory 4096
5. HardDisk 500Gb

Spesifikasi Perangkat Android:

1. Processor 600MHz ARMv6
2. Resolusi 320 x 480 pixels, 3,0 inches, capacitive touchscreen 16 jt warna
3. Memory 168 Mb
4. Memory Internal 128 Mb
5. *Operating System* Gingerbread 2.3
6. Jaringan HSDPA 900 / 2100, GSM 850 / 900 / 1800 / 1900

3.1.2.2 Spesifikasi Software yang digunakan:

1. Sistem Operasi Windows 7 Ultimate 64-bit
 2. *Eclipse Juno*
 3. JDK 7u11 Windows 64-bit
 4. Android SDK 21.01
-

5. AVD (*Android Virtual Device*)

3.1.3 Sumber Informasi

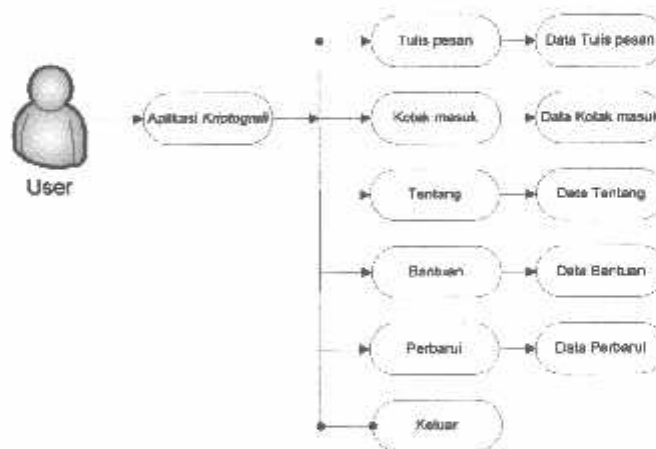
Data mengenai rancang bangun aplikasi kriptografi berbasis android dengan menggunakan algoritma *blowfish* yang berupa konsep dasar aplikasi android dan *resource* pendukung untuk membuat suatu aplikasi di *smartphone* android yang didapat dari buku dan *literature* di internet.

3.2 Perancangan Sistem

Dalam perancangan aplikasi kriptografi berbasis android dengan menggunakan algoritma *blowfish* menjelaskan *flowchart* yang berisi proses-proses interaksi user terhadap aplikasi dan rancangan *interface* aplikasi.

3.2.1 Alur Penggunaan Aplikasi

Alur dari penggunaan aplikasi oleh user digambarkan dengan use case diagram dibawah ini.

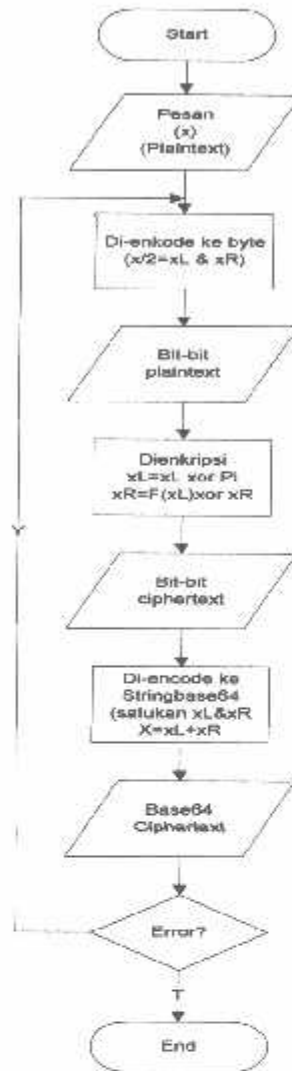


Gambar 3.1 Alur Penggunaan Aplikasi *Kriptografi*

3.2.2 Flowchart Prosedur Enkripsi dan Deskripsi

Proses kerja untuk enkripsi dan dekripsi menggunakan kunci dan proses yang serupa. Hanya berbeda pada inputan pesannya (*plaintext-ciphertext*) digunakan dengan urutan terbalik atau di inverskan. Oleh karena

itu, flowchart program untuk proses enkripsi dan dekripsi ditunjukkan pada gambar sebagai berikut:

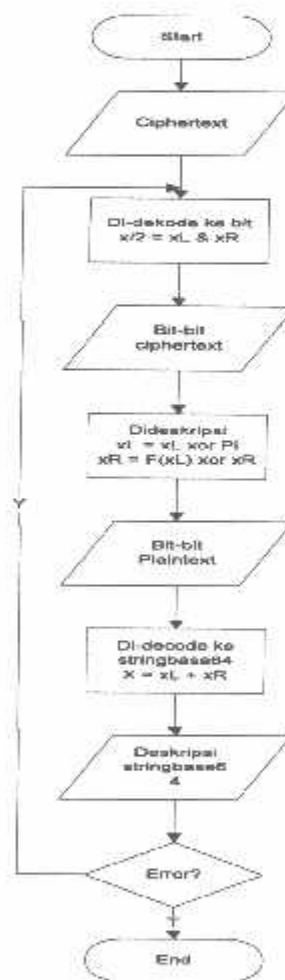


Gambar 3.2 Flowchart Prosedur Enkripsi

Keterangan:

1. Inputan pesan / x / *plaintext* yang akan dienkrpsi diasumsikan sebagai masukan, *plaintext* tersebut diambil sebanyak 64-bit dan apabila kurang dari 64-bit maka kita tambahkan bit-nya, supaya dalam operasi nanti sesuai dengan datanya.
2. Pesan tersebut diencode kedalam bit, $x/2 = xL \ \& \ xR$; hasil pengambilan tadi dibagi 2, 32-bit pertama disebut xL , 32-bit kedua disebut xR .

3. Menghasilkan bit-bit plaintext
4. Kemudian bit-bit plaintext dienkripsi, lakukan operasi $xL = xL \text{ xor } P_i$ dan $xR = F(xL) \text{ xor } xR$
5. Menghasilkan bit-bit ciphertext, hasil dari operasi diatas ditukar xL menjadi xR dan xR menjadi xL , lakukan sebanyak 16 kali, perulangan yang ke-16 lakukan lagi proses penukaran xL dan xR .
6. Pada proses ke-17 lakukan operasi untuk $xR = xR \text{ xor } P_{17}$ dan $xL = xL \text{ xor } P_{18}$.
7. Selanjutnya di-encode ke stringbase64, $x = xL + xR$ agar bit-bit tersebut tetap dapat dipetakan dalam karakter-karakter txt.
8. Dan menghasilkan ciphertext base64.



Gambar 3.3 Flowchart Prosedur Deskripsi

Keterangan:

1. Inputan pesan / x / *plaintext* yang akan dienkripsi diasumsikan sebagai masukan, *plaintext* tersebut diambil sebanyak 64-bit dan apabila kurang dari 64-bit maka kita tambahkan bit-nya, supaya dalam operasi nanti sesuai dengan datanya.
2. Pesan tersebut didekode kedalam bit, $x/2 = xL \ \& \ xR$: hasil pengambilan tadi dibagi 2, 32-bit pertama disebut xL , 32-bit kedua disebut xR .
3. Menghasilkan bit-bit *plaintext*
4. Kemudian bit-bit *plaintext* dideskripsi, lakukan operasi $xL = xL \ \text{xor} \ P_i$ dan $xR = F(xL) \ \text{xor} \ xR$
5. Menghasilkan bit-bit *ciphertext*, hasil dari operasi diatas ditukar xL menjadi xR dan xR menjadi xL , lakukan sebanyak 16 kali, perulangan yang ke-16 lakukan lagi proses penukaran xL dan xR .
6. Pada proses ke-17 lakukan operasi untuk $xR = xR \ \text{xor} \ P_{17}$ dan $xL = xL \ \text{xor} \ P_{18}$.
7. Selanjutnya didekode ke *stringbase64*, $x = xL + xR$ agar bit-bit tersebut tetap dapat dipetakan dalam karakter-karakter *txt*.
8. Menghasilkan *ciphertext base64*.
9. Dan menghasilkan deskripsi *base64*.

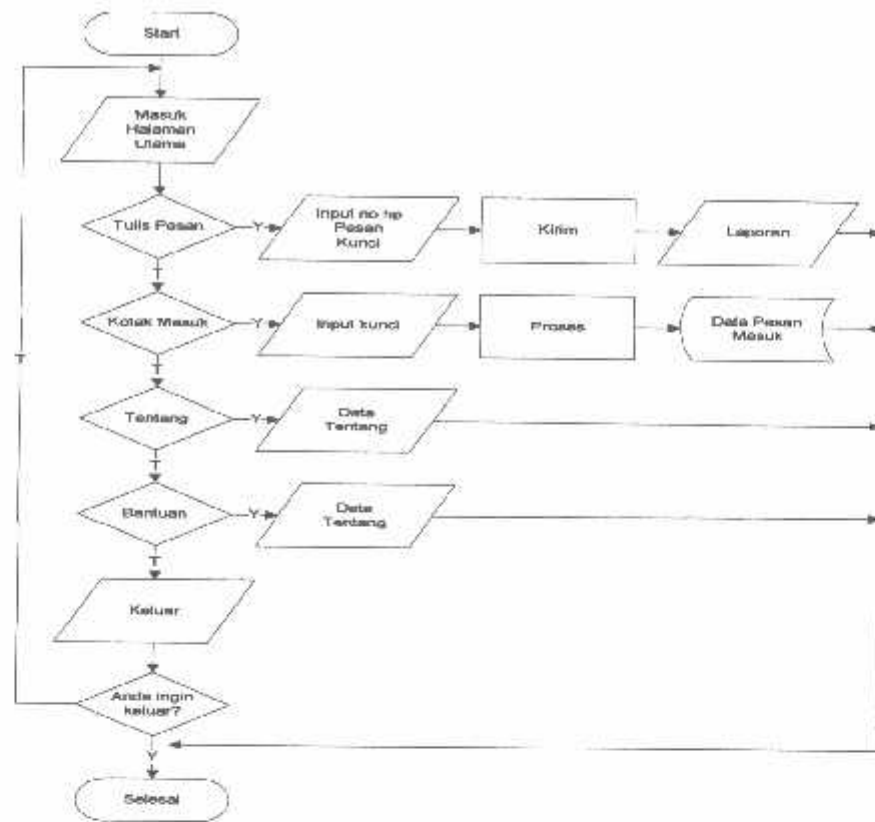
Plaintext: Text asli

Ciphertext: Text terenkripsi

Stringbase64: Format bit-bit yang telah di-encode atau di-dekode

3.3 Flowchart Sistem

Diagram alir (*Flowchart*) merupakan suatu gambaran secara grafik yang terdiri dari simbol-simbol dari algoritma dalam suatu program yang digunakan untuk menyatakan alur program. Flowchart program ditunjukkan pada gambar 3.4 berikut:



Gambar 3.4 Flowchart Program Aplikasi Kriptografi

Penjelasan alur flowchart program sebagai berikut:

1. Start, masuk halaman utama aplikasi diantaranya: tulis pesan, kotak masuk, tentang, bantuan, perbarui, dan menu keluar.
2. Menu tulis pesan, apabila user setuju aplikasi akan menampilkan halaman tulis pesan diantaranya : input nomor telepon/hp penerima, pesan, dan kata kunci yang akan dienkripsi yang berfungsi untuk membuka pesan tersebut bagi penerima.
3. Menu kotak masuk, user dapat menampilkan halaman kotak masuk dimana terdapat: input kata kunci yang dienkripsi untuk membuka pesan dikotak masuk.
4. Tentang, user dapat membuka menu tentang yang berisi identitas pembuat aplikasi.

5. Bantuan, user dapat mempelajari penggunaan aplikasi melalui halaman yang akan ditampilkan aplikasi di menu bantuan.
6. Keluar, menu yang berfungsi untuk keluar dari aplikasi, apabila setuju maka user akan keluar dari aplikasi, begitu juga sebaliknya apabila user membatalkan user akan pada halaman menu utama.

3.4 Perancangan Antar Muka

Desain antarmuka (*interface*) adalah sketsa awal tampilan dari program yang bertujuan untuk mempermudah programmer dalam tahap implementasi. Desain tampilan program inilah yang akan menjadi penghubung antara pengguna dan system, agar pengguna mudah berinteraksi dengan sistem.

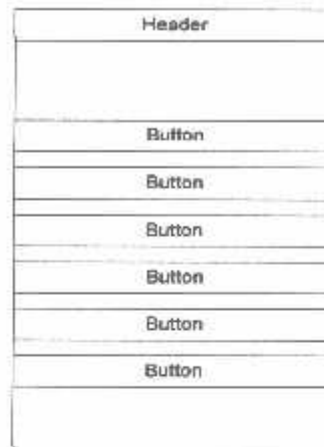
3.4.1 Rancangan Splash Screen



Gambar 3.5 Rancangan Splash Screen

Pada gambar 3.5 menunjukkan rancangan tampilan splash screen saat pengguna antarmuka aplikasi yang berjalan sekitar 3 detik untuk membuka halaman tampilan menu utama aplikasi.

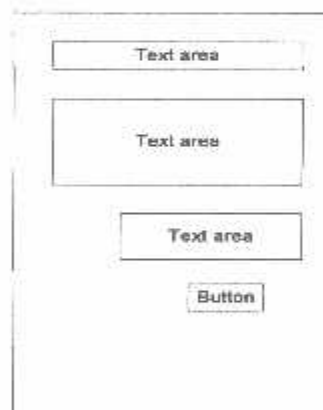
3.4.2 Rancangan Menu Utama



Gambar 3.6 Rancangan Menu Utama

Gambar 3.6 menunjukkan menu utama terdiri dari *header* dan 6 buah *button* dibagian bawah yang memiliki fungsi masing-masing diantaranya: Tulis pesan, Kotak masuk, Tentang, Bantuan, Perbarui, dan keluar.

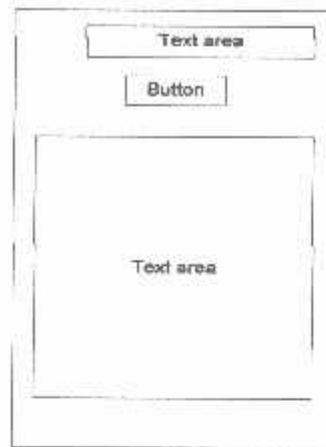
3.4.2 Rancangan Menu Tulis Pesan



Gambar 3.7 Rancangan Menu Tulis Pesan

Gambar 3.7 merupakan menu tulis pesan yang berisi diantaranya : 3 *text area* dan 1 *button*. Fungsi masing-masing diantaranya : Text area 1 berfungsi untuk inputan nomor telepon penerima, text area 2 berfungsi untuk penulisan pesan, text area 3 berfungsi untuk inputan kata kunci untuk dienkripsi, dan button untuk mengirim pesan.

3.4.3 Rancangan Menu Kotak Masuk



Gambar 3.8 Rancangan Menu Kotak Masuk

Gambar 3.8 menunjukkan tampilan menu kotak masuk yang berisi diantaranya : *text area*, *button* dan *text view*, yang memiliki fungsi masing-masing diantaranya: *text area* berfungsi sebagai inputan kata kunci yang dienkripsi untuk membuka menu kotak masuk, *button* berfungsi sebagai proses dari inputan kata kunci yang akan di enkripsi, *text view* merupakan tampilan kotak masuk pesan setelah melakukan enkripsi kata kunci.

3.4.4 Rancangan Menu Tentang



Gambar 3.9 Menu Tentang

Gambar 3.9 menunjukkan tampilan menu tentang aplikasi yang berisi *background* dan *button*, yang dapat dijelaskan memiliki fungsi sebagai berikut: *background* berfungsi untuk menampilkan keterangan tentang aplikasi dibuat dan *button* berfungsi sebagai tombol kembali pada menu utama.

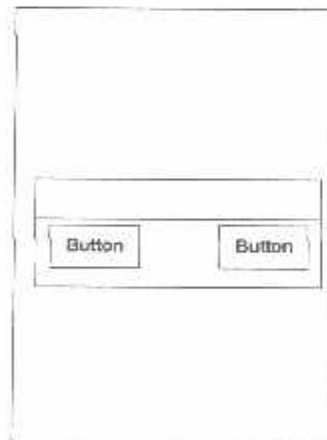
3.4.5 Rancangan Menu Bantuan



Gambar 3.10 Rancangan Menu Bantuan

Gambar 3.10 menunjukkan tampilan menu tentang aplikasi yang berisi *text view* dan *button*, yang memiliki beberapa fungsi masing-masing diantaranya: *text view* berfungsi untuk memberikan keterangan pemakaian aplikasi dan *button* berfungsi untuk tombol kembali pada menu utama.

3.4.6 Rancangan Menu Keluar



Gambar 3.12 Rancangan Menu Keluar

Gambar 3.12 merupakan tampilan menu keluar yang terdiri atas 2 button yang memiliki fungsi untuk mengkonfirmasi perintah menu.

BAB IV

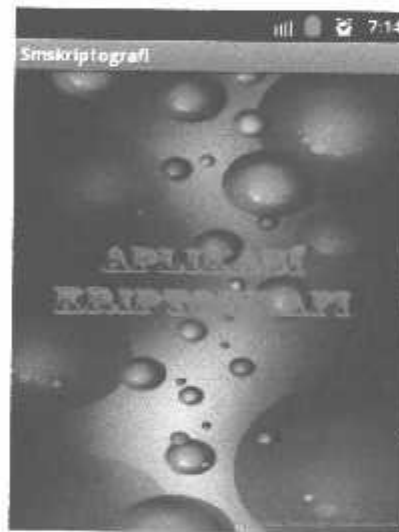
HASIL IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Hasil Implementasi Sistem

Pada proses implementasi ke dalam aplikasi *Kriptografi* yaitu mengubah implementasi sistem menjadi sistem yang dapat dijalankan. Pada tahap ini merupakan lanjutan dari tahap perancangan dan membutuhkan beberapa tahapan-tahapan lagi seperti tahap pemrograman spesifikasi aplikasi dengan desain sistem.

4.1.1 Hasil Implementasi Splash Screen

Pada menu splash screen yang berjalan sekitar 3 detik akan menampilkan logo dari aplikasi. Seperti gambar 4.1.



Gambar 4.1 Tampilan *Splash Screen*

4.1.2 Hasil Implementasi Menu Utama

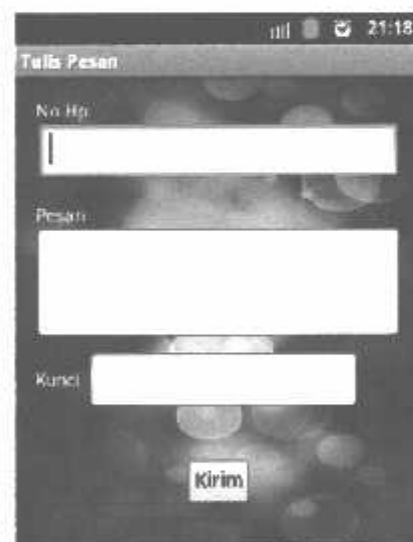
Pada tampilan menu utama terdapat 6 buah *button* yang memiliki fungsi masing-masing diantaranya: tulis pesan, kotak masuk, tentang, perbarui, bantuan, dan keluar. Seperti gambar 4.2.



Gambar 4.2 Tampilan Menu Utama

4.1.3 Hasil Implementasi Menu Tulis Pesan

Pada tampilan menu tulis pesan terdapat 3 buah *text area* dan 1 buah *button* yang memiliki fungsi masing-masing diantaranya : tulis no tlp/hp, tulis pesan, tulis kata kunci dan tombol kirim. Seperti gambar 4.3.



Gambar 4.3 Tampilan Tulis Pesan

4.1.4 Hasil Implementasi Menu Kotak masuk

Pada tampilan menu kotak masuk terdapat 1 buah *text area*, 1 buah *button*, dan 1 buah *text view* yang memiliki fungsi masing-masing diantaranya: input kata kunci, tombol proses, dan tampilan pesan masuk. Seperti gambar 4.4.



Gambar 4.4 Tampilan Menu Kotak Masuk

4.1.5 Hasil Implementasi Menu Tentang

Pada tampilan menu tentang terdapat 1 buah *text view* yang memiliki fungsi untuk menampilkan keterangan pembuat dan aplikasi. Seperti gambar 4.5



Gambar 4.5 Tampilan Menu Tentang

4.1.6 Hasil Implementasi Menu Bantuan

Pada tampilan menu bantuan terdapat 1 buah *text view* yang memiliki fungsi untuk menampilkan keterangan pembuat dan aplikasi. Seperti gambar 4.6



Gambar 4.6 Tampilan Menu Bantuan

4.1.7 Hasil Implementasi Menu Keluar

Pada tampilan menu keluar terdapat 1 buah *rectangle* dan 2 buah *button* yang memiliki fungsi untuk menampilkan keterangan keluar atau tidaknya dari aplikasi. Seperti gambar 4.8



Gambar 4.7 Tampilan Menu Keluar

4.2 Pengujian

Pada bab ini dilakukan pengujian pada aplikasi *kriptografi* berbasis Android. Pengujian ini meliputi pengujian fungsional sistem, pengujian waktu, pengujian keakuratan lokasi serta pengujian terhadap *operating system* android versi lain.

4.2.1 Pengujian Fungsional Sistem

Pengujian pertama yaitu fungsional sistem yang dilakukan terdiri dari beberapa pengujian. Diantaranya dari setiap pengujian sistem terdapat pada tabel 4.1.

Tabel 4.1 Pengujian Fungsional Sistem

Prosedur	Hasil	Keterangan
Mengirimkan pesan user dengan kata kunci yang dienkripsikan kepada penerima pesan	User dapat mengirim dan penerima dapat membuka pesan dengan kata kunci tersebut	sesuai
Mengirimkan pesan user tanpa kata kunci yang dienkripsikan kepada penerima pesan	User dapat mengirim pesan dan penerima tidak dapat menampilkan pesan	Tidak sesuai
Mengirimkan pesan user dengan kata kunci yang dienkripsikan kepada penerima pesan yang tidak memiliki aplikasi yang sama	User dapat mengirim pesan dan penerima dapat membuka pesan tetapi pesan tidak terbaca (angka dan huruf secara acak)	Tidak sesuai

Terdapat tiga kondisi saat pengujian mengirimkan pesan user dengan kata kunci yang dienkripsikan kepada penerima pesan, yaitu:

1. Pengujian pertama, User dapat mengirim dan penerima dapat membuka pesan dengan kata kunci tersebut.
2. Pengujian kedua User dapat mengirim pesan dan penerima tidak dapat menampilkan pesan. Ini dikarenakan user mengirimkan pesan tanpa kata kunci yang akan dienkripsi oleh aplikasi.

- Pengujian ketiga, User dapat mengirim pesan dan penerima dapat membuka pesan tetapi pesan tidak terbaca (angka dan huruf secara acak). Ini dikarenakan pengiriman pesan user dengan kata kunci yang dienkripsikan kepada penerima psan yang tidak memiliki aplikasi yang sama digunakan.

4.2.2 Pengujian Tulis Pesan

Pengujian tulis pesan aplikasi dilakukan pada *mobile* android versi 2.3.6 dengan ukuran layar 240 x 320 pixel yang ditunjukkan pada gambar 4.9:



Gambar 4.8 Tampilan Pengujian Tulis Pesan

4.2.3 Pengujian Kotak Masuk

Pengujian kotak masuk aplikasi dilakukan pada *mobile* android versi 2.3.6 dengan ukuran layar 240 x 320 pixel yang ditunjukkan pada gambar 4.10:



Gamabr 4.9 Tampilan Pengujian Kotak Masuk

4.2.4 Pengujian Pesan Terenkripsi

Pengujian pesan terenkripsi aplikasi dilakukan pada *mobile* android versi 2.3.6 dengan ukuran layar 240 x 320 pixel yang ditunjukkan pada gambar 4.11:



Gambar 4.10 Tampilan Pengujian Pesan Terenkripsi

4.3 Pengujian terhadap versi *Operating System* Android Lain

Pengujian ini adalah pengujian terhadap beberapa perangkat bergerak atau ponsel Android dari berbagai tipe, seperti pada tabel 4.2 berikut :

Tabel 4.2 Pengujian terhadap versi Android

Versi Android	Operating Sistem	Berjalan	Tidak Berjalan
2.3.6	Gingerbread	√	-
4.0.4	Ice Cream Sandwich	-	√
4.1.2	JellyBean	√	-

4.4 Pengujian Terhadap Berbagai Ukuran Layar

Pengujian yang terakhir adalah pengujian terhadap beberapa perangkat bergerak atau *smartphone* dari berbagai tipe, seperti pada tabel 4.3 berikut :

Tabel 4.3 Pengujian terhadap berbagai ukuran resolusi layar smartphone.

Ukuran Resolusi Layar Smartphone	Tampilan Aplikasi	
	Sesuai	Tidak Sesuai
240 x 320 Pixel	√	-
320 x 480 Pixel	√	-
430 x 850 Pixel	√	-
540 x 960 Pixel	√	-

Berdasarkan beberapa hasil pengujian aplikasi yang telah dilakukan, aplikasi dapat dijalankan pada beberapa versi operating sistem android terkecuali pada operating sistem android versi 4.0 Ice cream sandwich, aplikasi juga telah dilakukan pengujian terhadap berbagai ukuran resolusi layar smartphone dan menghasilkan penyesuaian tampilan yang sesuai dengan berbagai ukuran layar yang telah diujikan.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan pembahasan pada bab sebelumnya dapat disimpulkan sebagai berikut:

1. Berdasarkan pengujian fungsional sistem yang telah dilakukan pada aplikasi kriptografi, user dapat mengirim dan penerima dapat membuka pesan dengan kata kunci yang sama.
2. Berdasarkan pengujian fungsional sistem yang kedua pada aplikasi kriptografi, user dapat mengirim pesan dan penerima tidak dapat menampilkan pesan, ini disebabkan karena user mengirim pesan tanpa mencantumkan kata kunci yang akan dienkripsi oleh aplikasi.
3. Berdasarkan pengujian fungsional sistem yang ketiga pada aplikasi kriptografi, user dapat mengirim pesan dan penerima dapat membuka pesan tetapi pesan terbaca berupa ciphertext (text yang terenkripsi), ini dikarenakan pengiriman pesan user dilengkapi dengan kata kunci yang dienkripsi oleh aplikasi dan penerima pesan tidak menggunakan aplikasi yang sama digunakan.

5.2. Saran

Saran yang didapatkan dari percobaan dan evaluasi yang telah dilakukan adalah sebagai berikut:

1. Aplikasi *Kriptografi* masih banyak kekurangannya sehingga diharapkan kedepannya aplikasi ini lebih berkembang dengan memiliki fitur yang lebih lengkap, seperti: kotak keluar, kotak terkirim, konsep, dan input kontak yang tersimpan pada *mobile* dalam pengiriman pesan dapat ditampilkan.
2. Pada aplikasi *Kriptografi* data yang menjadi inputan untuk enkripsi diharapkan tidak lagi sebatas dokumen berformat .txt dapat juga dengan format JPG, Png atau dengan format lainnya.

3. Aplikasi *Kriptografi* terutama pada fungsionalitas dari aplikasi enkripsi itu sendiri tidak terbatas hanya pada saat melakukan pengiriman pesan SMS (*Short Message Service*) tapi juga dapat dilakukan pada saat mengirim pesan MMS (*Multimedia Messaging Service*) atau pengiriman pesan multimedia.
 4. Aplikasi Kriptografi ini masih menggunakan inputan nomor kontak secara manual, untuk pengembangan selanjutnya diharapkan inputan nomor kontak dapat langsung melalui daftar nomor telepon yang tersimpan pada *mobile* atau *sim card*.
-

DAFTAR PUSTAKA

1. Ariyus, Dony. 2008 *Pengantar Ilmu Kriptografi*, Yogyakarta, Andy Offset.
 2. Akbarul Huda, S.Si, Arif. 2012. *24 Jam Pintar Pemrograman*, Yogyakarta, Andi Publisher.
 3. Alim Sutanto, Candra *Penggunaan Algoritma Blowfish Dalam Kriptografi*. (diakses tanggal 2 Juli 2013 dari informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2009-2010/Makalah0910/MakalahStrukdis0910-070.pdf)
 4. Wahyu Prasetyo, Galih *Bab II Landasan Teori* (diakses tanggal 2 juli 2013 dari <http://www.google.com/url?sa=t&rct=j&q=Bab+II+pengertian+SMS+ShortMessagesService+adalah>)
 5. Erikawaty, Shanty *Implimentasi Algoritma Kriptografi Blowfish Untuk Keamanan Dokumen*. (diakses tanggal 2 Juli 2013 dari http://repository.amikom.ac.id/index.php/add_downloader/Publikasi_06.11.1189.pdf/2213)
 6. Gilang Kautzar, Mohammad *Bab IV Analisa Dan Perancangan*. (diakses tanggal 2 Juli 2013 dari <http://www.google.com/url?sa=t&rct=j&q=Bab+IV+penerapan+algoritma+blowfish&source>)
-

LAMPIRAN



FORMULIR PERBAIKAN SKRIPSI

Nama : Mochamad Chakimin
NIM : 0918140
Jurusan : Teknik Informatika S-1
Judul : Rancang Bangun Aplikasi Kriptografi Berbasis Android Dengan Menggunakan Algoritma Blowfish

Tanggal	Penguji	Uraian	Paraf
20 Agustus 2013	I	Revisi Program : Hilangkan Menu Perbarui Revisi Laporan :Perbaiki laporan & Isi laporan sesuai dengan catatan-catatan saya pada laporan anda	
20 Agustus 2013	II	Pengaturan spasi, flowchart algoritma, flowchart aplikasi, penutup, citasi	

Anggota Penguji :

Penguji Pertama

Karina Auliasari, ST, M.Eng
NIP.P. 1031000426

Penguji Kedua

Nurlaily Vendyansyah, ST.

Mengetahui

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT
NIP. 197404162005021002

Dosen Pembimbing II

Yosep Agus Pranoto, ST
NIP. P.1031000432



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Raya Karanglo Km. 2 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

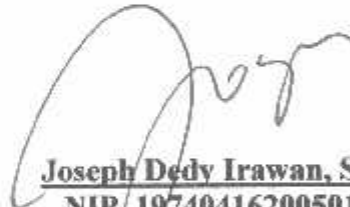
Nama : Mochamad Chakimin
NIM : 0918140
Jurusan : Teknik Informatika S-1
Judul : Rancang Bangun Aplikasi Kriptografi Berbasis Android Dengan
Menggunakan Algoritma *Blowfish*

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Selasa
Tanggal : 20 Agustus 2013
Nilai : 80,65 (A)

Panitia Ujian Skripsi :

Ketua Majelis Penguji



Joseph Dedy Irawan, ST, MT
NIP/197404162005011002

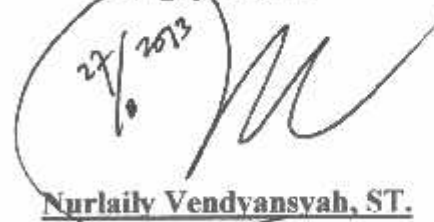
Anggota Penguji :

Penguji Pertama



Karina Auliasari, ST, M.Eng
NIP.P. 1031000426

Penguji Kedua



Nurlaily Vendyansyah, ST.



FORMULIR BIMBINGAN SKRIPSI

Nama : Mochamad Chakimin
Nim : 09.18.140
Masa Bimbingan : 11 Mei 2013 s/d 11 Nopember 2013
Judul Skripsi : Rancang Bangun Aplikasi Kriptografi Berbasis Android
Dengan Menggunakan Algoritma *Blowfish*

No	Tanggal	Uraian	Paraf Pembimbing
1	20-06-2013	Rancangan Program	
2	24-06-2013	Revisi Bab I,II,III	
3	28-06-2013	Revisi Penulisan	
4	02-07-2013	Konsultasi Program	
5	23-07-2013	Revisi Bab I,II,III,IV	
6	24-07-2013	Revisi Bab I,II,III,IV,V,Kuisonaire	
7	29-07-2013	Revisi Flowchart dan Kesimpulan	
8	22-07-2013	ACC Makalah Seminar Hasil	
9	16-08-2013	ACC Laporan Ujian Skripsi	
10			

Malang, 16 Agustus 2013
Dosen Pembimbing II


Yosep Agus Pranoto, ST

NIP.P. 1031000432

Form S-4b

Source Code Splash Screen

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/splash3"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity" >

</RelativeLayout>
```

Source Code Halaman Menu

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/windows2"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity" >

<TextView
    android:id="@+id/txtNoHP"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:text="Aplikasi Kriptografi"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textColor="#ffffff" />

<Button
    android:id="@+id/btnKey"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/txtNoHP"
    android:layout_marginTop="26dp"
    android:text="Tulis Pesan"
    android:textColor="#000000" />
```

```

<Button
    android:id="@+id/btnSend"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/btnKey"
    android:layout_alignRight="@+id/btnKey"
    android:layout_below="@+id/btnKey"
    android:text="Pesan Masuk"
    android:textColor="#000000" />

<Button
    android:id="@+id/button6"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/btnSend"
    android:layout_alignRight="@+id/btnSend"
    android:layout_below="@+id/btnSend"
    android:text="Tentang"
    android:textColor="#000000" />

<Button
    android:id="@+id/button1"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button6"
    android:layout_alignRight="@+id/button6"
    android:layout_below="@+id/button6"
    android:text="Bantuan"
    android:textColor="#000000" />

<Button
    android:id="@+id/button5"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button1"
    android:layout_alignRight="@+id/button1"
    android:layout_below="@+id/button1"
    android:text="Perbarui"
    android:textColor="#000000" />

<Button
    android:id="@+id/button4"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button5"
    android:layout_alignRight="@+id/button5"
    android:layout_below="@+id/button5"
    android:text="Keluar"

```

```
android:textColor="#000000" />
```

```
</RelativeLayout>
```

Source Code Menu Tulis Pesan

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/windows3"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity" >

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:text="Phone Number"
    android:textColor="#ffffff" />

<EditText
    android:id="@+id/edtNoTujuan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/textView1"
    android:ems="10"
    android:inputType="phone" >

    <requestFocus />
</EditText>

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/edtNoTujuan"
    android:layout_below="@+id/edtNoTujuan"
    android:layout_marginTop="14dp"
    android:text="Message"
    android:textColor="#ffffff" />
```

```

<EditText
    android:id="@+id/edtPesan"
    android:layout_width="20dp"
    android:layout_height="90dp"
    android:layout_alignParentLeft="true"
    android:layout_alignRight="@+id/edtNoTujuan"
    android:layout_below="@+id/textView2"
    android:ems="10"
    android:inputType="textMultiLine" />

<EditText
    android:id="@+id/edtKey"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/textView3"
    android:layout_alignBottom="@+id/textView3"
    android:layout_toRightOf="@+id/textView1"
    android:ems="10" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/edtPesan"
    android:layout_below="@+id/edtPesan"
    android:layout_marginTop="20dp"
    android:text="Key"
    android:textColor="#ffffff" />

<TextView
    android:id="@+id/txtChiper"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView3"
    android:layout_alignParentBottom="true"
    android:text="." />

<Button
    android:id="@+id/btnSend"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/edtKey"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="14dp"
    android:text="Send" />

</RelativeLayout>

```

Source Code Kotak Masuk

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/sunset"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".BlankActivity" >

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="18dp"
    android:text="Key"
    android:textColor="#ffffff" />

<EditText
    android:id="@+id/edtKey2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/textView1"
    android:layout_marginLeft="28dp"
    android:layout_toRightOf="@+id/textView1"
    android:ems="10" >

    <requestFocus />
</EditText>

<ListView
    android:id="@+id/list"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_below="@+id/btnKey" >

</ListView>

<Button
    android:id="@+id/btnKey"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/edtKey2"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="21dp"
    android:text="Proses" />

</RelativeLayout>

```

Source Code Menu Tentang

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/tentang" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="26dp"
        android:text="Back"
        android:textColor="#ffffff" />

</RelativeLayout>
```

Source Code Menu Bantuan

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bantuan"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".BlankActivity" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="26dp"
        android:text="Back" />

</RelativeLayout>
```

Java Script Splash Screen

```
package com.example.smskriptografi;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;

public class SplashActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);

        Thread timer = new Thread() {
            public void run() {
                try {
                    //berapalama splashscreen akan
                    ditampilkan dalam milisecond
                    sleep(3000);
                } catch (InterruptedException e) {
                    // TODO: handle exception
                    e.printStackTrace();
                } finally {
                    //activity yang akan
                    dijalankan setelah splashscreen selesai
                    Intent i = new
                    Intent(SplashActivity.this, MainActivity.class);
                    startActivity(i);
                    finish();
                }
            }
        };
        timer.start();
    }
}
```


Java Script Activity Menu

```
package com.example.smskriptografi;

import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu);

        Button Tulis_Pesan = (Button)
        findViewById(R.id.btnKey);
        Button Kotak_Masuk = (Button)
        findViewById(R.id.btnSend);
        Button Ferbarui = (Button)
        findViewById(R.id.button5);
        Button Keluar = (Button)
        findViewById(R.id.button4);
        Button Bantuan = (Button)
        findViewById(R.id.button1);
        Button Tentang = (Button)
        findViewById(R.id.button6);

        Tulis_Pesan.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent i = new
                Intent(getApplicationContext(), Tulis_Pesan.class);
                startActivity(i);
            }
        });

        Kotak_Masuk.setOnClickListener(new
        View.OnClickListener() {
            @Override
```

```

        public void onClick(View view) {

            Intent i = new
Intent(getApplicationContext(), Kotak_Masuk.class);
            startActivity(i);
        }
    });

    Bantuan.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {

            Intent i = new
Intent(getApplicationContext(), Bantuan.class);
            startActivity(i);
        }
    });

    Tentang.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {

            Intent i = new
Intent(getApplicationContext(), Tentang.class);
            startActivity(i);
        }
    });

    Perbarui.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {

            Intent i = new
Intent(getApplicationContext(), Perbarui.class);
            startActivity(i);
        }
    });

    Keluar.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {

```

```

        AlertDialog keluar=new
AlertDialog.Builder(MainActivity.this).create();

        keluar.setMessage("Apakah anda
ingin keluar?");
        keluar.setButton(-1,"Ya", new
DialogInterface.OnClickListener() {

            @Override
            public void
onClick(DialogInterface cobatok, int arg1) {

                MainActivity.this.finish();

            }

        });
        keluar.setButton(-2,"Tidak", new
DialogInterface.OnClickListener() {

            @Override
            public void
onClick(DialogInterface cobatok, int arg1) {
                // TODO Auto-generated
method stub
                cobatok.cancel();

            }

        });

        keluar.show();

    }

}

public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the
action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

```

Java Script SMS Kripto

```
package com.example.smskriptografi;

import android.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

public class Kripto {
    public static String enkripsi(String pesan, String
key){
        try {
            SecretKeySpec KS = new
SecretKeySpec(key.getBytes(), "Blowfish");
            Cipher cipher =
Cipher.getInstance("Blowfish");
            cipher.init(Cipher.ENCRYPT_MODE, KS);
            byte[] encrypted =
cipher.doFinal(pesan.getBytes());
            return Base64.encodeToString(encrypted,
Base64.NO_PADDING);
        } catch (Exception e) {
            return "ERROR:"+e.getMessage();
        }
    }

    public static String deskripsi(String chipertext,
String key){
        try {
            SecretKeySpec KS = new
SecretKeySpec(key.getBytes(), "Blowfish");
            Cipher cipher =
Cipher.getInstance("Blowfish");
            cipher.init(Cipher.DECRYPT_MODE, KS);
            byte[] decrypted =
cipher.doFinal(Base64.decode(chipertext, Base64.NO_PADDING));
            return new String(decrypted);
        } catch (Exception e) {
            return "ERROR";
        }
    }
}
```

Java Script Tulis Pesan

```
package com.example.smskriptografi;

import android.os.Bundle;
import android.app.Activity;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class Tulis_Pesan extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tulis);
        final EditText edtPhone = (EditText) findViewById(R.id.edtNoTujuan);
        final EditText edtPesan = (EditText) findViewById(R.id.edtPesan);
        final Button btnKirim = (Button) findViewById(R.id.btnSend);
        final TextView txtChiper = (TextView) findViewById(R.id.txtChiper);
        final EditText edtKey = (EditText) findViewById(R.id.edtKey);

        btnKirim.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                String noHP= edtPhone.getText().toString();
                String pesan = edtPesan.getText().toString();

                String strKey=edtKey.getText().toString().trim();
                String chiperText= Kripto.enkripsi(pesan, strKey);

                if(noHP.length(>0){
                    kirimSMS(noHP, chiperText);
```

```

        txtChiper.setText(chiperText);
    }else {
        Toast.makeText(getBaseContext(), "Write your
phone number", Toast.LENGTH_LONG).show();
    }
}

});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

public void kirimSMS(String noHP, String pesan){
    String SENT = "SMS_SENT";
    String DELIVERED = "SMS_DELIVERED";
    PendingIntent sentPI = PendingIntent.getBroadcast(this, 0, new Intent(SENT),
0);
    PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0, new
Intent(DELIVERED), 0);
    //ketika SMS SENT
    registerReceiver(new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            // TODO Auto-generated method stub
            switch (getResultCode()) {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(),
"Message Sent", Toast.LENGTH_LONG).show();

                    break;
                case
android.telephony.SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                    Toast.makeText(getBaseContext(), "Error",
Toast.LENGTH_LONG).show();

                    break;
                case

```

```

android.telephony.SmsManager.RESULT_ERROR_NO_SERVICE:
    Toast.makeText(getBaseContext(), "No
Signal", Toast.LENGTH_LONG).show();
    break;
    case
android.telephony.SmsManager.RESULT_ERROR_NULL_PDU:
    Toast.makeText(getBaseContext(), "PDU
NULL", Toast.LENGTH_LONG).show();
    break;
    case
android.telephony.SmsManager.RESULT_ERROR_RADIO_OFF:
    Toast.makeText(getBaseContext(), "Error,
GSM MATI", Toast.LENGTH_LONG).show();
    break;

    default:
        Toast.makeText(getBaseContext(), "Sorry",
Toast.LENGTH_LONG).show();
    }
}
}, new IntentFilter(SENT));

//ketika SMS DELIVERED
registerReceiver(new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO Auto-generated method stub
        switch (getResultCode()) {
            case Activity.RESULT_OK:
                Toast.makeText(getBaseContext(),
"Message Delivered", Toast.LENGTH_LONG).show();

                break;
            case Activity.RESULT_CANCELED:
                Toast.makeText(getBaseContext(),
"Message Panding..", Toast.LENGTH_LONG).show();
                break;

            default:
                Toast.makeText(getBaseContext(), "Sorry",
Toast.LENGTH_LONG).show();

```

```

        }
        Toast.makeText(getApplicationContext(),
        "hakam_hakim@rocketmail.com", Toast.LENGTH_LONG).show();
    }

    }, new IntentFilter(DELIVERED));

    android.telephony.SmsManager sms =
    android.telephony.SmsManager.getDefault();
    sms.sendTextMessage(noHP, null, pesan, sentPI, deliveredPI);

    }
}

```

Java Script Kotak Masuk

```

package com.example.smskriptografi;

import java.util.ArrayList;
import java.util.List;

import android.net.Uri;
import android.os.Bundle;
import android.app.Activity;
import android.database.Cursor;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;

public class Kotak_Masuk extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.kotak_masuk);
        final ListView list = (ListView) findViewById(R.id.list);
        final Button btnKey = (Button) findViewById(R.id.btnKey);
        btnKey.setOnClickListener(new View.OnClickListener() {

```



```

        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            final EditText edtKey= (EditText)
findViewById(R.id.edtKey2);
            List<String> msgs = getSMS(edtKey.getText().toString().trim());
            if(msgs.isEmpty()){
                msgs.add("Tidak ada SMS yang bisa dipecahkan dengan key
tsbt");
            }
            ArrayAdapter<String> smsAdapter = new
ArrayAdapter<String>(getBaseContext(),
                android.R.layout.simple_list_item_1 ,msgs);
            list.setAdapter(smsAdapter);
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

public List<String> getSMS(String strkey) {
    List<String> list = new ArrayList<String>();
    Uri uri = Uri.parse("content://sms/inbox");
    Cursor c = null;
    try{
        c = getApplicationContext().getContentResolver().query(uri, null, null
,null,null);
    }catch(Exception e){
        e.printStackTrace();
    }
    try{
        for (boolean hasData = c.moveToFirst(); hasData; hasData = c.moveToNext()) {

            final String noHP = c.getString(c.getColumnIndex("address"));
            final String msg = c.getString(c.getColumnIndexOrThrow("body"));

```

```

String plainTeks= Kripto.deskripsi(msg,strkey);
if(!plainTeks.equalsIgnoreCase("ERROR")){
    list.add(noHP + "\nText: " + plainTeks);
}
}
}
}catch(Exception e){
    e.printStackTrace();
}
}
c.close();
return list;
}
}
final EditText edtKey= (EditText) findViewById(R.id.edtKey);

btnKirim.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        String noHP= edtPhone.getText().toString();
        String pesan = edtPesan.getText().toString();

        String strKey=edtKey.getText().toString().trim();
        String chiperText= Kripto.enkripsi(pesan, strKey);

        if(noHP.length()>0){
            kirimSMS(noHP, chiperText);
            txtChiper.setText(chiperText);
        }else {
            Toast.makeText(getApplicationContext(), "Write your
phone number", Toast.LENGTH_LONG).show();
        }
    }

});
}

@Override

```

```

public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

public void kirimSMS(String noHP, String pesan){
    String SENT = "SMS_SENT";
    String DELIVERED = "SMS_DELIVERED";
    PendingIntent sentPI = PendingIntent.getBroadcast(this, 0, new Intent(SENT),
0);
    PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0, new
Intent(DELIVERED), 0);
    //ketika SMS SENT
    registerReceiver(new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            // TODO Auto-generated method stub
            switch (getResultCode()) {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(),
"Message Sent", Toast.LENGTH_LONG).show();

                    break;
                case
android.telephony.SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                    Toast.makeText(getBaseContext(), "Error",
Toast.LENGTH_LONG).show();
                    break;
                case
android.telephony.SmsManager.RESULT_ERROR_NO_SERVICE:
                    Toast.makeText(getBaseContext(), "No
Signal", Toast.LENGTH_LONG).show();
                    break;
                case
android.telephony.SmsManager.RESULT_ERROR_NULL_PDU:
                    Toast.makeText(getBaseContext(), "PDU
NULL", Toast.LENGTH_LONG).show();
                    break;
                case
android.telephony.SmsManager.RESULT_ERROR_RADIO_OFF:
                    Toast.makeText(getBaseContext(), "Error,
GSM MATI", Toast.LENGTH_LONG).show();

```

```

                break;
            default:
                Toast.makeText(getApplicationContext(), "Sorry",
Toast.LENGTH_LONG).show();
            }
        }, new IntentFilter(SENT));

//ketika SMS DELIVERED
registerReceiver(new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO Auto-generated method stub
        switch (getResultCode()) {
            case Activity.RESULT_OK:
                Toast.makeText(getApplicationContext(),
"Message Delivered", Toast.LENGTH_LONG).show();

                break;
            case Activity.RESULT_CANCELED:
                Toast.makeText(getApplicationContext(),
"Message Panding..", Toast.LENGTH_LONG).show();
                break;

            default:
                Toast.makeText(getApplicationContext(), "Sorry",
Toast.LENGTH_LONG).show();
        }
        Toast.makeText(getApplicationContext(),
"hakam_hakim@rocketmail.com", Toast.LENGTH_LONG).show();
    }

}, new IntentFilter(DELIVERED));

        android.telephony.SmsManager sms =
        android.telephony.SmsManager.getDefault();
        sms.sendTextMessage(noHP, null, pesan, sentPI, deliveredPI);
    }
}

```

Java Script Tentang

```
package com.example.smskriptografi;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class Tentang extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tentang);

        Button exit = (Button) findViewById(R.id.button1);

        exit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                finish();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

Java Script Bantuan

```
package com.example.smskriptografi;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class Bantuan extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bantuan);

        Button exit = (Button) findViewById(R.id.button1);

        exit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

Source Code Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.smskriptografi"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="10" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.smskriptografi.SplashActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="MainActivity"
            android:label="Menu" ></activity>
        <activity android:name="Kripto" android:label="Kripto"
        ></activity>
        <activity android:name="Tulis_Pesan" android:label="Tulis
Pesan" ></activity>
        <activity android:name="Kotak_Masuk" android:label="Kotak
Masuk" ></activity>
        <activity android:name="Bantuan" android:label="Bantuan"
        ></activity>
        <activity android:name="ReceiveSMS" android:label="Receive"
        ></activity>
        <activity android:name="Perbarui" android:label="Perbarui"
        ></activity>
        <activity android:name="getApplicationContext"
            android:label="" ></activity>
        <activity android:name="Tentang"
            android:label="Tentang" ></activity>

    </application>
</manifest>
```