ISSN: 2252-4983

# ENKRIPTOR-DEKRIPTOR ISI E-MAIL BERBASIS ANDROID DENGAN ALGORITMA BLOWFISH

**Yonathan Gani Panjaitan**
Program Studi Teknik Informatika, Fakultas Ilmu Komputer
Universitas Dian Nuswantoro
Email: jonathanpanjaitan49@gmail.com

**Ajib Susanto**
Program Studi Teknik Informatika, Fakultas Ilmu Komputer
Universitas Dian Nuswantoro
Email: ajib.susanto@dsn.dinus.ac.id

**Wijanarto**
Program Studi Teknik Informatika, Fakultas Ilmu Komputer
Universitas Dian Nuswantoro
Email: wijanarto@dsn.dinus.ac.id

**Ibnu Utomo W.M**
Program Studi Teknik Informatika, Fakultas Ilmu Komputer
Universitas Dian Nuswantoro
Email: ibnu@dsn.dinus.ac.id

## ABSTRAK

Perkembangan teknologi informasi yang pesat diiringi oleh perkembangan dalam hal keamanan data. Terutama informasi atau hal-hal rahasia yang orang lain tidak boleh mengetahui. Salah satu teknologi yang terus semakin berkembang adalah *Email* dan munculnya platform baru yang bernama Android yang digunakan pada *smartphone*. *Email* merupakan salah satu cara untuk bertukar informasi antara satu orang dengan yang lain. Kejahatan teknologi informasi terus berkembang dan dengan berbagai macam model kejahatan, sebagai contohnya adalah *Sniffing*. Algoritma Blowfish merupakan salah satu algoritma algoritma *block cipher 64-bit block* yang cukup aman untuk mengamankan data di dalam sebuah program, tidak dipatenkan, mempunyai ruang kunci yang besar dan panjangnya bermacam-macam dari 32 *bits* sampai 448 *bits* serta cukup kuat keamanannya, sehingga pada bagian kuncinya tidak mudah diserang oleh pihak lain. Metode pendekatan sistem yang digunakan dalam pembangunan aplikasi adalah *Extreme Programming (XP)* untuk menerapkan fungsi dari Enkripsi dan Dekripsi dalam versi aplikasi berbasis Android. Hasil penelitian berupa aplikasi berbasis Android yang dapat mengenkripsi sebagian atau semua isi *E-mail* yang dikirim dan dapat mendeskripsi kembali seperti *text* semula menggunakan algoritma Blowfish dengan kunci tertentu.

**Kata kunci:** *android, email, sniffing, extreme programming, blowfish*.

## *ABSTRACT*

*The rapid development of information technology is accompanied by the development in terms of data security. Specifically, the information or secret things that other people may not know. One technology that keeps growing is E-mail and the emergence of a new platform called Android that is used on smartphones. E-mail is one way to exchange information between one person and another. Crime information technology continues to evolve and with various models of crime, for example is Sniffing. Blowfish algorithm is an algorithm algorithm block cipher 64-bit block that is safe enough to secure data within a program, not patented, has a room key and the length varying from 32 bits to 448 bits, and is strong enough security, so that the the key parts are not easily attacked by the other party. Method approach of system used in application development is the Extreme Programming (XP) to implement the functions of encryption and decryption in the version of Android-based applications. The results of this research were Android-based application that can encrypt part or all of the contents of E-mails sent and can describe again as the original text using the Blowfish algorithm with a particular key.*

***Keywords:*** *android, email, sniffing, extreme programming, blowfish*.

## 1. PENDAHULUAN

Kemudahan dan kecepatan akses komunikasi dan informasi berpengaruh besar terhadap perkembangan pertukaran data dan informasi termasuk di dalamnya keamanan informasinya. Informasi akan menjadi mudah diketahui[1], diambil, dimanipulasi maupun disalahgunakan oleh berbagai pihak lain yang tidak mempunyai akses dan tidak berhak mendapatkan data dan informasi. Penerima informasi harus memverifikasi informasi yang didapat, yakin suatu informasi tersebut adalah dari pengirim atau sumber yang tepat[2], dikenal dan benar isinya. Saat ini media komunikasi yang digunakan atau dimanfaatkan semestinya merupakan media yang mudah, murah, terjangkau dan digunakan oleh kebanyakan orang. Salah satu yang popular untuk berkomunikasi adalah *E-mail*. Beberapa tahun terakhir ini muncul sebuah platform baru yang merupakan teknologi baru, salah satunya adalah telepon seluler (ponsel). Salah satunya yang paling populer sekarang ini adalah ponsel *smartphone* berplatform *android* yang mempunyai beberapa fungsi seperti *e-mail, social media, video streaming, multimedia, transfer data, multiplayer games* dan lain-lain. Namun, akibat dari berkembangnya jumlah pengguna fasilitas *E-mail* di *android* yang digunakan sebagai alat penyimpan data dan informasi, baik informasi pribadi ataupun informasi bisnis perusahaan, tidak sedikit pengguna yang berbuat tidak baik untuk mencari tahu bahkan mengubah-ubah data yang disimpan oleh pengguna tersebut. Apalagi diketahui sekarang bahwa *Android* merupakan *platform open source* yang dikatakan masih baru teknologinya dan juga masih sedikitnya pengamanan di ponsel *Android*[3].

Contoh kasus tahun 2012 dalam hal penyadapan isi dari *E-mail* adalah Skandal Penyadapan Kedua Dalam Kisruh Bumi Plc yang dimana dalam berita ini Samin Tan, *chairman* Bumi Plc merasa aneh dengan sebuah *E-mail* yang dikirimkan kepada dirinya yang berisi bahwa situs *Wikipedia* membutuhkan korespondensi untuk menerbitkan artikel tentang Samin Tan. Ternyata *E-mail* tersebut berisi *malware* yang berhasil mencuri semua isi *E-mail* dari Samin Tan dan membobol info-info penting mengenai Bumi Plc[4].

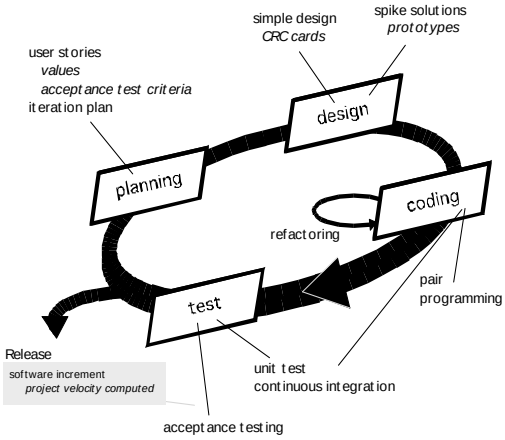
Blowfish[2][5][6] merupakan salah satu algoritma *block cipher 64-bit block* yang tidak dipatenkan mempunyai ruang kunci yang besar dan panjangnya bermacam-macam dari 32 *bits* sampai 448 *bits* serta cukup kuat keamanannya, sehingga pada bagian kuncinya tidak mudah diserang oleh pihak lain. Hasil penelitian Saikumar Manku dan K. Vasant[7], algoritma *Blowfish* menyediakan keamanan yang kuat sehingga tidak ada satu diantara pengiriman dan penerimaan data di *hack*. Begitu juga hasil penelitian Manju Suresh dan Neema M.[8] untuk keamanan transmisi di internet dari semua algoritma kriptografi, algoritma *Blowfish* adalah terbaik dalam hal waktu eksekusi, penggunaan memori, *throughput*, konsumsi daya, dan keamanan[9] dengan demikian cocok untuk *Internet of Things (IOT)*.

Sistem kriptografi dapat dikatakan baik terletak di kerahasiaan kunci, bukan terletak pada kerahasiaan suatu algoritma yang digunakan. *Blowfish* jika diimplementasikan dengan strategi yang tepat dapat lebih optimal, dapat dijalankan dimemori kurang dari 5 (lima) KB dan kesederhanaan pada proses algoritmanya. Sehingga untuk menjamin keamanan pengiriman informasi perlu dibangun aplikasi berbasis *android* untuk pengguna *smartphone* yang dapat digunakan untuk mengamankan data atau informasi sebagian maupun keseluruhan *text* (isi pesan) yang dikirim melalui media *E-mail* dengan menggunakan algoritma *Blowfish* dengan membuat kunci tertentu.

Pembangunan enkriptor-dekriptor ini dilakukan dengan menggunakan dan menerapkan fungsi Algoritma *Blowfish* yang dikerjakan dengan metode pendekatan *Extreme Programming*[10][11] sehingga dihasilkan beberapa versi kecil aplikasi android yang terus ditingkatkan hingga mencapai tujuan optimal yaitu menerapkan Algoritma *Blowfish* yang bisa mengamankan/merahasiakan isi *E-mail* yang dilakukan langsung dalam aplikasi berbasis *Android* dengan spesifikasi minimum *Android* 2.2.

## 2. METODE PENELITIAN

Metode pendekatan sistem berbasis obyek yaitu *Extreme Programming (XP)*[10][11] dengan pemodelam *Unified Modeling Language (UML)* menekankan pada interaksi antar *customer* dengan *developer*. Aplikasi dikerjakan sesuai kesepakatan antar *customer* dan *developer* dalam suatu iterasi yang pada tiap iterasinya dihasilkan suatu unit aplikasi siap pakai (ditingkatkan fungsinya pada iterasi selanjutnya, terus diperbaiki sampai tujuan penelitian sepenuhnya dicapai).



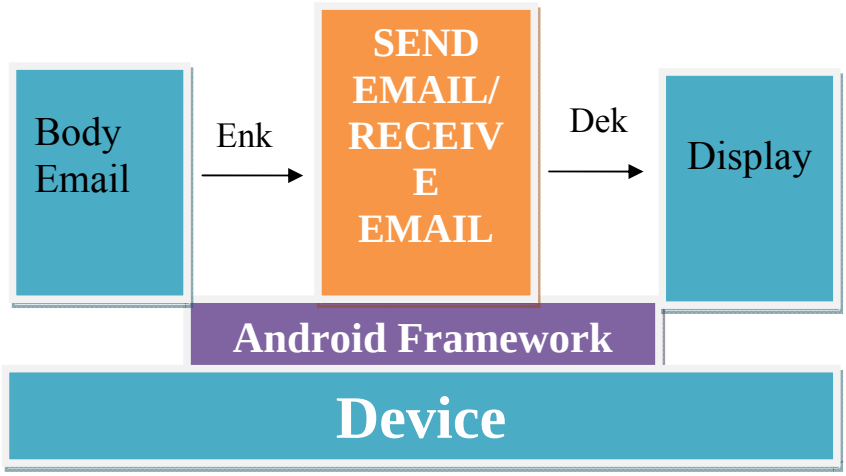

**Gambar 1. Fase *Extreme Programming (XP)*[10]**

Fase pengembangan sistem dengan XP :

a. *Release Planning* (perencanaan penelitian secara utuh)
b. *Iteration* (iterasi penelitian, terdapat perencanaan, pengerjaan, serta pengujian di tiap unit aplikasi yang dihasilkan)
c. *Acceptance Test* (pengujian secara menyeluruh untuk versi terakhir aplikasi)
d. *Small Release* (perilisan aplikasi)

## 3. HASIL DAN PEMBAHASAN

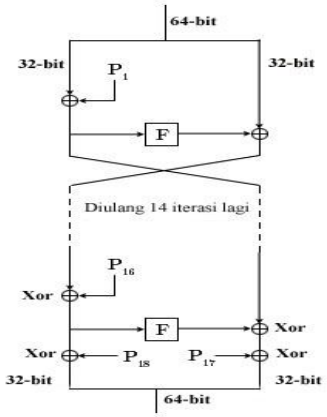
### *3.1 Arsitektur Sistem*




**Gambar 2. Desain Arsitektur Aplikasi**

Gambar 2 menunjukkan beberapa komponen yang terlibat dalam sistem, yaitu isi email (*body email*) pada perangkat (*device*) Android, fungsi Enkripsi (*Enk*), Send *Email/Receive Email*, Dekripsi (*Dek*) beroperasi dalam lingkup sistem Android (*Android Framework*). Masukkan ke aplikasi adalah *Body Email* yang diketik oleh *user*, kemudian aplikasi mengubah isi dari *body email* tersebut (dienkrip) menjadi bentuk atau hasil dari algoritma *Blowfish* yang nantinya hasil tersebut dikirimkan menjadi sebuah *E-mail*. Kemudian jika mendapatkan balasan atau *Receive E-mail* yang mana harus mengakses salah satu *provider E-mail*, akan diubah (didekrip) kembali menjadi *plainteks* atau teks biasa yang dapat ditampilkan dan terbaca.

### *3.2 Penerapan Algoritma Blowfish pada Aplikasi*

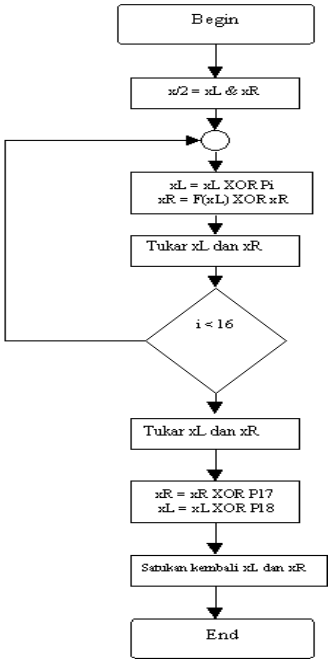



**Gambar 3.** ***Iterasi Algoritma Blowfish***

Gambar 3 menggambarkan di dalam *Blowfish* ada penghitungan iterasi subkunci sebanyak 521 iterasi[12]. Langkah-langkahnya terdiri dari :

1) Inilialisasi P-array, selanjutnya 4 (empat) S-box secara berturut-turut dengan nilai *String* yang tetap. Nilai *String* berisi digit *hexadesimal* dari Pi.
2) $P_1$ di XOR dengan 32-bit pertama dari kunci, $P_2$ di XOR dengan 32-bit kedua dari kunci dan seterusnya untuk setiap bit dari kunci (sampai $P_{18}$). Ulangi terhadap bit kunci sampai keseluruhan P-array di XOR dengan bit kunci.
3) Enkripsi semua *String* nol menggunakan algoritma *Blowfish* dengan subkunci seperti pada langkah (1) dan langkah (2).
4) $P_1$ dan $P_2$ diganti dengan *output* dari langkah (3).
5) Enkripsi *output* dari langkah (3) dengan algoritma *Blowfish* menggunakan subkunci yang sudah dimodifikasi sebelumnya.
6) $P_3$ dan $P_4$ diganti dengan *output* dari langkah (5).
7) Lanjutkan proses, ganti keseluruhan elemen dari P-array, selanjutnya keseluruhan 4 (empat) S-box secara berturut-turut, dengan *output* yang berubah secara kontinyu dari algoritma *Blowfish*.

Langkah-langkah di atas diwakili oleh fungsi *setKey()*, yang mana bagian ini merupakan bagian pengubahan subkey, dan bagian ini berubah seiring key inputan yang digunakan.

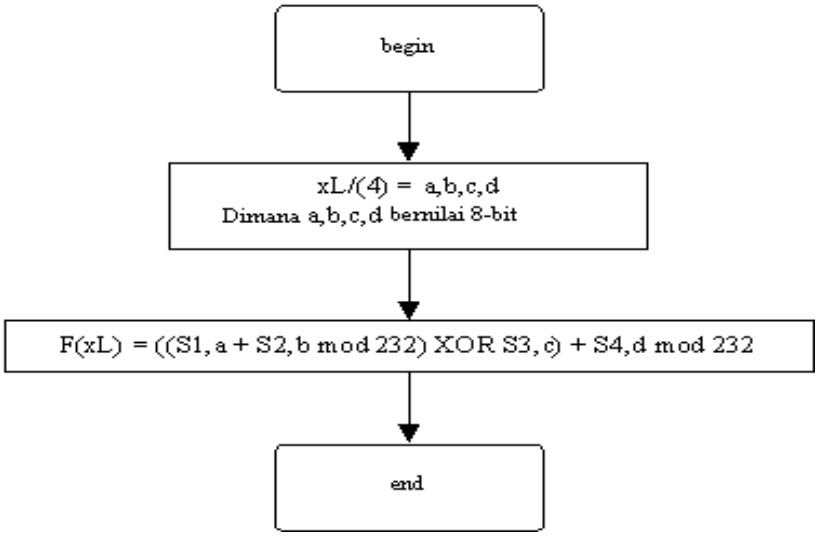



**Gambar 4.** ***Flowchart Blowfish***

Langkah-langkah algoritma *Blowfish* dalam aplikasi enkriptor-dekriptor yang ditunjukkan pada gambar 4 adalah sebagai berikut :

1) User diminta untuk memasukkan inputan *text* pada *body mail* pada aplikasi ini.
   Pada *code program* dimulai pada fungsi *getInit()*, dimana fungsi ini yang menangkap karakter yang diinputkan *user*. Kemudian mengubahnya dalam bentuk *String*.
2) Jika *user* memasukkan inputan kurang dari 64 *bit* maka secara otomatis dilakukan penambahan bit oleh algoritma *Blowfish*.
3) Jika sudah memenuhi kriteria 64 bit data maka *blowfish engine* akan mulai menginisiasi perputaran atau iterasinya yaitu sebanyak 16 kali perputaran[6].
   Pada *code program*, disinilah dimulai fungsi *run()*. Dimana fungsi ini mengubah *string* inputan tadi menjadi dalam bentuk *byte*.
4) Data 64 bit yang sudah diinputkan oleh user akan dibagi 2 (anggap saja xl dan xr), menjadi masing-masing 32 bit yang kemudian diproses dengan menggunakan perhitungan *XOR*.
   Pada *code program*, *engine blowfish (blowfishengine.java)* dipanggil oleh fungsi *run()*, yang mana nantinya *engine* ini akan mulai bergerak dan melakukan proses iterasi. Pada saat ini jika user ingin mengenkrip data maka *run()* akan memanggil fungsi di *blowfishengine.java* yaitu *tryEncryptBlock()* yang merupakan implementasi dari fungsi *encryptBlock()*. Begitupun sebaliknya jika melakukan dekripsi maka yang dipanggil adalah *tryDecryptBlock()* yang merupakan implementasi dari fungsi *decryptBlock()*.
5) Jika sudah sampai ke iterasi ke 16, maka hasil perhitungan xl dan xr ditukar kembali dan di XOR-kan dengan bagian 17 dan 18.
6) Kembali ditukar dan jadilah sebuah *cipherteks* yang berisi 64 *bit data*, sesuai dengan inputan yang diberikan user.

.



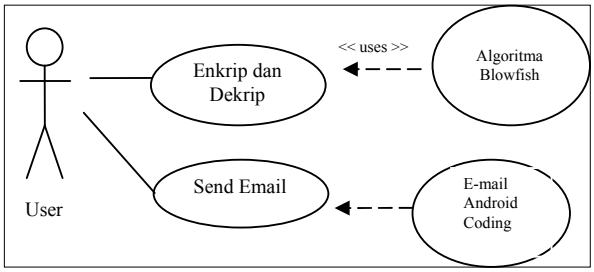

**Gambar 5. *Flowchart Fungsi F*[13]**

Gambar 5 menunjukkan fungsi F yang merupakan salah satu bagian penghitungan dari algoritma *Blowfish*. Penjelasannya adalah sebagai berikut :

1) Salah satu pembagian dari data 64 bit (xl) ini akan dioperasikan dengan hasil dari fungsi F ini.
2) Kemudian setelah mendapatkan hasil dari fungsi F ini maka hasilnya yaitu F(xl) di XOR kan oleh xr.

Kedua langkah ini diwakili oleh fungsi F() dengan S-box yang sudah di-*array*-kan.

### *3.3 Perancangan Aplikasi*

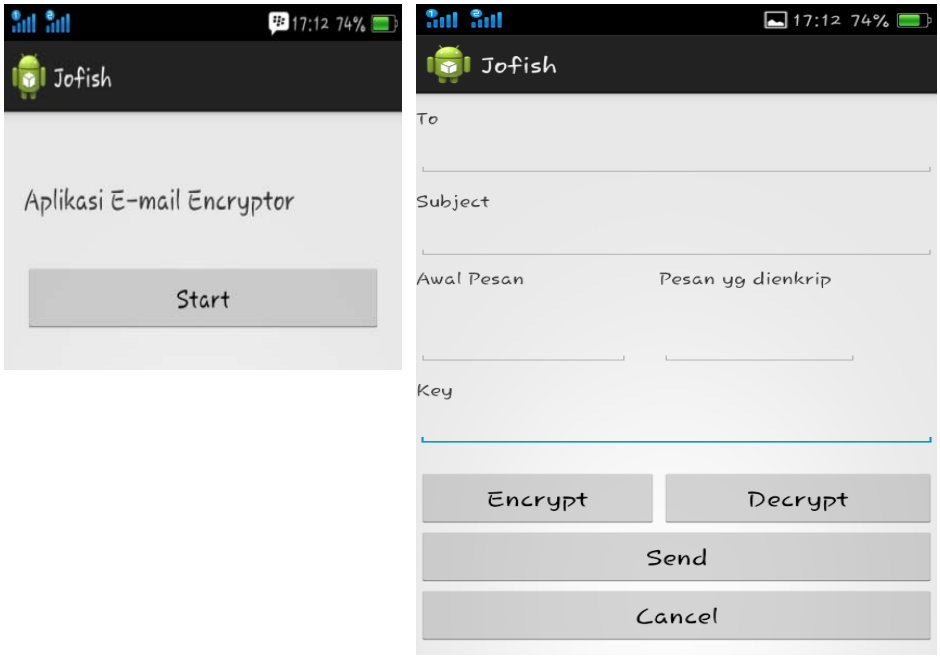



**Gambar 6. Diagram *Use Case***

Gambar 6 menggambarkan pengguna aplikasi ini dapat memilih apakah user ingin mengenkripkan isi dari *E-mail* ataupun tidak. Namun user bisa mengirimkan *E-mail* melalui aplikasi tersebut

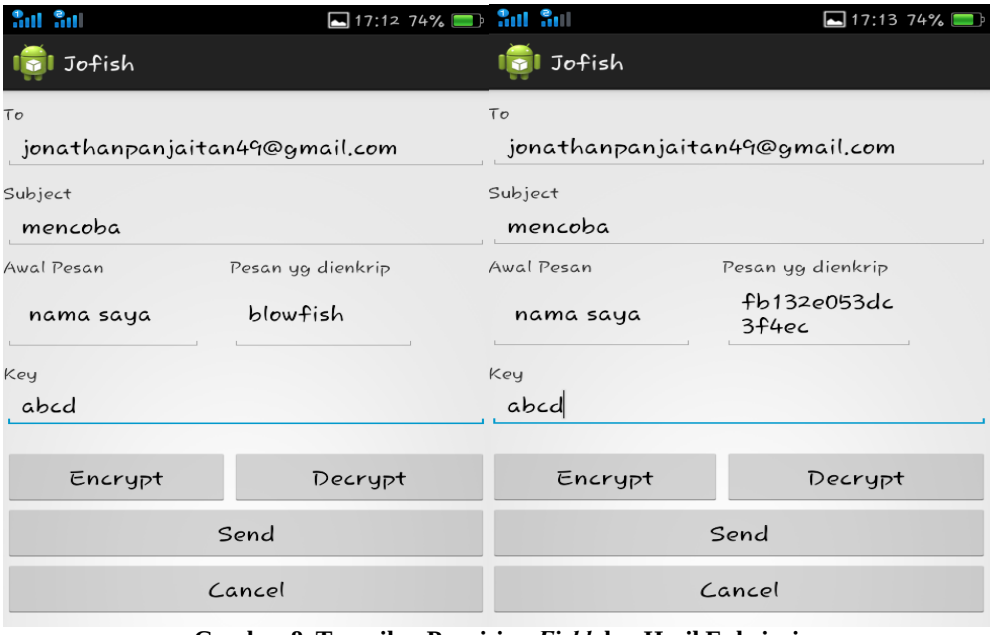
### *3.4 Implementasi Enkriptor-Dekriptor*

Implementasi sistem berupa aplikasi android yang terdiri dari kelas utama sebagai halaman utama serta berfungsi menghubungkan semua fungsi dalam aplikasi ini. Dimana *main.java* menjadi halaman utama untuk memanggil *EnkripsiAktivitas.java* dan *DekripsiAktivitas.java* yang berinti pada *BlowfishEngine.java*. Berikut merupakan hasil implementasi:




**Gambar 7. Tampilan Awal dan Inti Aplikasi (*Field* Kosong)**

Gambar 7 memperlihatkan tampilan awal aplikasi yang terdiri dari satu tombol (*start*) untuk memulai membuka aplikasi. Kemudian ketika sudah memilih aktivasi aplikasi maka aplikasi ini akan dimulai dengan susunan di gambar sebelahnya. Dimana *user* bisa memilih apakah menggunakan sistem enkripsi/dekripsinya.




**Gambar 8. Tampilan Pengisian *Field* dan Hasil Enkripsi**

Pada Gambar 8 ini memperlihatkan *user* bisa memasukkan bagian-bagian dari sebuah *E-mail* seperti alamat *Email*, *subject*, isi pesan yang terbagi 2 (dua) menjadi awal pesan (tidak bisa dienkripsi/dekripsi), yang kedua pesan yang dienkripsi. Sehingga pesan nanti tidak semua harus dienkripsi (kembali pada keinginan *user*)

## 4. KESIMPULAN

a. Fungsi Algoritma *Blowfish* dapat diterapkan dengan baik pada aplikasi *Android* tanpa bersandar pada fungsi *cipher* pada pustaka *android API*.
b. Pengenkripsian dan pendekripsian pesan juga berjalan dengan baik, begitupun dengan pengiriman *E-mail* dari aplikasi ini.
c. Aplikasi ini dapat digunakan dengan *Android* versi *Gingerbread* ke atas, namun jika ada yang menggunakan versi di bawahnya, tetap bisa digunakan.

## DAFTAR PUSTAKA

[1] Rahman, C., Nadhori, I.U., Fathoni, K. 2009. *Studi Dan Implementasi Algoritma Blowfish Untuk Enkripsi Email.* ITS : Surabaya.
[2] Sitinjak, S., Fauziah, Y., Juwairiah. 2010. *Aplikasi Kriptografi File Menggunakan Algoritma Blowfish.* UPN "Veteran" : Yogyakarta.
[3] Taufan A., Y., Winarno, I., Fathoni, K. . *Enkripsi Email Dengan Menggunakan Metode Elgamal Pada Perangkat Mobile.* ITS : Surabaya
[4] Skandal Penyadapan Kedua Dalam Kisruh Bumi PLC. https://harcipto.wordpress.com/2012/12/12/skandal-penyadapan-kedua-dalam-kisruh-bumi-plc/. Diakses 06 Maret 2017 Jam 10:26.
[5] Bruce Schneier. *The Blowfish Encryption Algorithm.* https://www.schneier.com/academic/blowfish/. Diakses 06 Maret 2017 Jam 10:15.
[6] Schneier, Bruce. 1995. *The Blowfish Encryption Algorithm -- One Year Later*. Dr. Dobb's Journal. https://www.schneier.com/academic/archives/1995/09/the_blowfish_encrypt.html. Diakses pada 06 Maret 2017 Jam 10:35.
[7] Saikumar Manku, K. Vasanth, 2015, *Blowfish Encryption Algorithm for Information Security*, ARPN Journal Engineering and Applied Science, Asian Research Publishing Network (ARPN).
[8] Manju Suresh, Neema M, 2016, *Hardware Implementation of Blowfish Algorithm for The Secure DataTransmission in Internet of Things*, Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST 2016), Procedia Technology.

[9] Chaitali Haldankar, Sonia Kulwelkar, 2014, *Implementation of AES and Blowfish Algorithm*, IJRET: International Journal of Research in Engineering and Technology.
[10] Pressman, Roger S., 2010, *Software Engineering : a practitioner's approach Seventh Edition*., New York, The McGraw - Hill Companies, Inc..
[11] *Extreme Programming*. http://www.extremeprogramming.org. Diakses pada 06 Maret 2017 Jam 10:20.
[12] Tri Andriyanto, D.L. Crispina Pardede. 2008. *Studi dan Perbandingan Algoritma Idea dan Algoritma Blowfish*, Gunadarma : Jakarta
[13] Ema Utami, Shanty Erikawaty A.T, 2010, *Penerapan Algoritma Blowfish Untuk Membuat Sebuah Model Kriptosistem Algoritma Dan Menganalisis Kinerja Algoritma Blowfish Dengan Simulasi Data Terbatas*, Jurnal DASI : Yogyakarta.