Jurnal Coding Sistem Komputer Untan ISSN 2338-493X
Volume 03, No. 3 (2015), hal 57-66

# APLIKASI ENKRIPSI DAN DEKRIPSI PESAN SINGKAT MENGGUNAKAN ALGORITMA KNAPSACK BERBASIS ANDROID

**[1]Rio Irawan, [2]Ilhamsyah, [3]Yulrio Brianorman.**
[1] [2] [3]Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jalan Prof. Dr. H. Hadari Nawawi, Pontianak
Telp./Fax.: (0561) 577963
e-mail :
[1] rio_irawan@student.untan.ac.id, [2] ilhamsm99@gmail.com
[3] yulrio.brianorman@siskom.untan.ac.id

A B S T R A K

*Perkembangan lalu lintas trafik komunikasi diikuti oleh maraknya aksi penyadapan Short Message Service (SMS) yang dilakukan oleh pihak yang tidak bertanggung jawab. Ada berbagai cara untuk melakukan penyadapan yaitu dengan mengambil info dari Base Transceiver Station (BTS) milik provider atau dengan menggunakan bantuan BTS buatan untuk mencegat SMS maupun voice via On The Air (OTA) sebelum menuju BTS asli. Algoritma Knapsack merupakan algoritma kriptografi asimetris yang menggunakan kunci publik untuk mengenkripsi pesan dan menggunakan kunci pribadi untuk mendeskripsikan pesan. Dalam penelitian ini menggunakan Algoritma Knapsack untuk melakukan enkripsi dan dekripsi SMS dan menerapkan pada smartphone Android. Hasil dari teks pesan dapat berubah ketika penerima tidak menggunakan aplikasi knapsack dan kunci yang sama dengan kunci pengirim sehingga kerahasiaan pesan terjaga dengan baik. Diharapkan penelitian ini dapat diterapkan oleh masyarakat untuk menghindari proses penyadapan data melalui SMS. Pada pada penelitian ini aplikasi Android dengan menggunakan algoritma kriptografi Knapsack dalam pembuatan aplikasi enkripsi dan dekripsi untuk pesan singkat atau Short Message Service (SMS) dapat melakukan enkripsi dan dekripsi dengan proses pengiriman dan penerimaan data secara utuh dan lengkap.*

**Kata Kunci** : android, penyadapan, sms, kriptografi, knapsack

## 1. PENDAHULUAN

Perkembangan lalu lintas trafik komunikasi di Indonesia khususnya penggunaan layanan *Short Message Service (SMS)* sangat pesat. Berdasarkan laporan Siaran Pers No 98/PIH/KOMINFO/2012 yang dilansir Kementrian Kominfo Republik Indonesia, terdapat peningkatan penggunaan trafik layanan *Short Message Service (SMS)* pada perusahaan penyelenggara telekomukasi seperti PT. Telkomsel dan PT. Indosat di saat perayaan Natal tahun 2012 dan dalam pergantian tahun baru tahun 2012 ke 2013. Perkembangan lalu lintas trafik komunikasi ini juga diikuti oleh aksi penyadapan *Short Message Service (SMS)* yang dilakukan oleh pihak yang tidak bertanggung jawab. Penyadapan ini berlangsung banyak cara diantaranya dengan mengambil info dari *Base Transceiver Station (BTS)* milik provider, atau dengan menggunakan bantuan *BTS* buatan untuk mencegat *SMS* maupun *voice* via O*n The Air (OTA*) sebelum menuju *BTS* asli. [1]

Permasalahan dari komunikasi dan informasi yang tersebar melalui *SMS* dapat dengan mudah disadap oleh orang tidak dikenal. Oleh karena itu, dibutuhkan suatu metode atau kajian ilmu pengetahuan yang dapat mengamankan pesan dari tindak penyadapan. Isi pesan yang akan dikirim akan diacak terlebih dahulu (enkripsi) menggunakan algoritma tertentu sebelum dikirim dan hanya akan bisa dibaca oleh orang yang memiliki kunci sebagai penerjemah pesan (deskripsi). Kriptografi merupakan ilmu dan seni untuk menjaga keamanan pesan [2].

 ISSN 2338-493X

Penelitian tentang enkripsi dan dekripsi *SMS* pada sistem operasi Android, sudah pernah dilakukan oleh Pranarelza (2014) menggunakan algoritma Rijndael. Algoritma rijndael menggunakan kunci yang sama untuk mengenkripsi dan mendeskripsikan pesan (kriptografi asimetri), sehingga relatif lebih lemah dibanding kriptografi dengan Algoritma Knapsack menggunakan dua kunci berbeda untuk proses enkripsi deskripsi *SMS*.

Algoritma Knapsack adalah algoritma kriptografi asimetris yang menggunakan kunci publik untuk mengenkripsi pesan dan menggunakan kunci pribadi untuk mendeskripsikan pesan. Penelitian ini menggunakan Algoritma Knapsack untuk proses enkripsi dan dekripsi *SMS* dan diaplikasikan pada *smartphone* Android. Diharapkan dari penelitian ini, dapat diterapkan oleh masyarakat untuk menghindari penyadapan data melalui *SMS*.

## 2. Tinjauan Pustaka

Penelitan ini menggunakan referensi-referensi mengenai teori, konsep dan metode yang digunakan.

### 2.1. Kriptografi

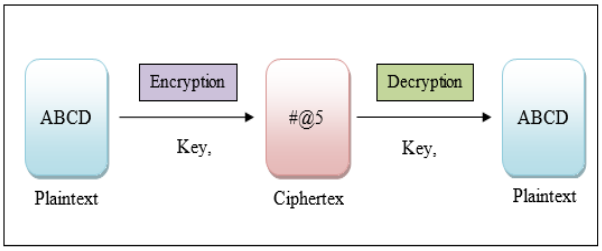
Kriptografi mempunyai sejarah yang sangat panjang. Kriptografi sudah digunakan 4000 tahun yang lalu dan diperkenalkan oleh orang-orang mesir lewat hieroglyph. Jenis tulisan ini bukanlah bentuk standar untuk menulis pesan. [3]

Kriptografi menurut bahasa yaitu kata Kriptografi dibagi menjadi dua, yaitu *kripto* dan *graphia*. *Kripto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Kriptografi adalah sebuah teknik dalam mengamankan dan mengirim data dalam bentuk yang hanya diketahui oleh pihak yang berhak membukanya. Kriptografi merupakan ilmu dan seni dalam memproteksikan informasi dengan mengubahnya ke dalam bentuk himpunan karakter acak yang tidak dapat dibaca. Kriptografi adalah sebuah cara yang efektif dalam mengamankan informasi penting baik yang tersimpan dalam media penyimpanan atau melalui jaringan komunikasi. [3]

Terminology kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain [4]. Pada prinsipnya, kriptogarfi memiliki 4 komponen utama yaitu:

1. *Plaintext*, yaitu pesan yang dapat dibaca secara langsung.
2. *Ciphertext*, yaitu pesan yang disandikan.
3. *Key*, yaitu kunci untuk melakukan enkripsi atau dekripsi.
4. Algoritma, yaitu metode yang digunakan untuk melakukan enkripsi atau dekripsi.

Enkripsi (*encryption*) adalah sebuah proses menjadikan pesan yang dapat dibaca (*plaintext*) menjadi pesan acak yang tidak dapat dibaca (*ciphertext*). Sedangkan dekripsi (*decryption*) merupakan proses kebalikan dari enkripsi dimana proses ini mengubah *ciphertext* menjadi *plaintext*. Proses enkripsi dan dekripsi menggunakan kunci dapat dilihat pada Gambar 1.




Gambar 1. Ilustrasi enkripsi dan dekripsi

### 2.2. Algoritma Knapsack

Algoritma Knapsack adalah algoritma kriptografi kunci publik yang keamanan algoritma ini terletak pada sulitnya memecahkan persoalan Knapsack (Knapsack *Problem*). Knapsack artinya karung atau kantung.karung mempunyai kapasitas muat terbatas. Barang-barang dimasukkan kedalam karung hanya sampai batas kapasitas maksimum karung saja [4]. Tahapan dalam membuat kunci publik dan kunci privat dalam algoritma Knapsack adalah sebagai berikut:

1. Tentukan barisan *superincreasing*.
2. Kalikan setiap elemen di dalam barisan tersebut dengan *n* modulo *m*. Modulus m seharusnya angka yang lebih besar daripada jumlah semua

elemen di dalam barisan, sedangkan pengali *n* seharusnya tidak mempunyai faktor persekutuan dengan *m*.

3. Hasil perkalian akan menjadi kunci publik sedangkan barisan *super-increasing* semula menjadi kunci privat.

2.2.1 Enkripsi

Proses enkripsi dilakukan dengan tahapan sebagai berikut:

1. Enkripsi dilakukan dengan cara yang sama yaitu dengan menggunakan algoritma Knapsack sebelumnya.
2. Mula-mula *plaintext* dipecah menjadi blok bit yang panjangnya sama dengan kardinalitas barisan kunci publik.
3. Kalikan setiap bit di dalam blok dengan elemen yang berkoresponden di dalam kunci publik.

*Plaintext* sebelum proses enkripsi adalah 011000110101101110 dan kunci publik yang digunakan {62, 93, 81, 88, 102, 37}. *Plaintext* dibagi menjadi blok yang panjangnya 6, kemudian setiap bit di dalam blok dikalikan dengan elemen yang berkorepsonden didalam kunci publik :

Blok *plaintext* ke-1 : 011000

Kunci publik : 62, 93, 81, 88, 102, 37
Kriptogram : (1 x 93) + (1 x 81) = 174
Blok *plaintext* ke-2 : 110101

Kunci publik : 62, 93, 81, 88, 102, 37
Kriptogram : (1 x 62) + (1 x 93) + (1 x 88) + (1 x 37) = 280
Blok *plaintext* ke-3 : 101110

Kunci publik : 62, 93, 81, 88, 102, 37
Kriptogram :
(1 x 62) + (1 x 81) + (1 x 88) +
(1 x 102) = 333

Jadi, *ciphertext* yang dihasilkan :
174, 280, 333

2.2.2 Dekripsi

Proses dekripsi dilakukan dengan tahapan sebagai berikut:

1. Dekripsi dilakukan dengan menggunakan kunci privat.
2. Awalnya penerima pesan menghitung $n^{-1}$ , yaitu balikan *n* modulo *m*, sedemikian sehingga $n \cdot n^{-1} \equiv 1 \pmod{m}$. Kekongruenan ini dapat dihitung dengan cara yang sederhana sebagai berikut (disamping dengan cara yang sudah pernah diberikan pada Teori Bilangan Bulat):
   a. $n \cdot n^{-1} \equiv 1 \pmod{m}$
   *b*. $n \cdot n^{-1} = 1 + km$
   c. $n^{-1} = (1 + km)/n$, dengan *k* sembarang bilangan bulat
3. Kalikan setiap kriptogram dengan $n–1 \bmod m$, lalu nyatakan hasil kalinya sebagai penjumlahan elemen-elemen kunci privat untuk memperoleh *plaintext* dengan menggunakan algoritma pencarian menjadi *superincreasing* Knapsack.

Proses dekripsi dimisalkan dengan mendekripsikan *ciphertext* dari Contoh 4 dengan menggunakan kunci rahasia {2, 3, 6, 13, 27, 52}. Di sini, $n = 31$ dan $m = 105$. Nilai $n^{-1}$ diperoleh sebagai berikut:

$$n^{-1} = (1 + 105k)/31$$

Dengan mencoba $k = 0, 1, 2, \ldots$, maka untuk $k = 18$ diperoleh $n^{-1}$ bilangan bulat, yaitu:

$$n^{-1} = (1 + 105 \cdot 18)/31 = 61$$

*Ciphertext* dari proses enkripsi adalah 174, 280, 222. *plaintext* yang berkoresponden diperoleh kembali sebagai berikut:

$174 \cdot 61 \bmod 105 = 9 = 3 + 6$, berkoresponden dengan 011000
$280 \cdot 61 \bmod 105 = 70 = 2 + 3 + 13 + 52$, berkoresponden dengan 011000
$333 \cdot 61 \bmod 105 = 48 = 2 + 6 + 13 + 27$, berkoresponden dengan 101110

Setelah dikorespondensikan dengan kunci maka *plaintext* yang dihasilkan kembali adalah:

011000 011000 101110

### 2.3. Kode ASCII

Kode Standar Amerika untuk pertukaran informasi atau ASCII (American Standard Code for Information Interchange) merupakan suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Kode ini selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 bit. Namun, ASCII disimpan sebagai sandi 8 bit dengan menambakan satu angka 0 sebagai bit significant paling tinggi. Bit tambahan ini sering digunakan untuk uji prioritas. Karakter control pada ASCII di-bedakan menjadi 5 kelompok sesuai dengan penggunaan yaitu berturut-turut meliputi logical communication, Device control, Information separator, Code extention, dan physical communication. Code ASCII ini banyak dijumpai pada papan ketik (*key-board*) computer atau instrument-instrument digital.

Jumlah kode ASCII adalah 255 kode. Kode ASCII 0..127 merupakan kode ASCII untuk manipulasi teks; sedangkan kode ASCII 128. 255 merupakan kode ASCII untuk manipulasi grafik. Kode ASCII sendiri dapat dikelompokkan lagi kedalam beberapa bagian

1. Kode yang tidak terlihat simbolnya seperti Kode 10 (Line Feed), 13(*Carriage Return*), 8 (Tab), 32 (Space)
2. Kode yang terlihat simbolnya seperti abjad (A..Z), numerik (0..9), karakter khusus (~!@#$%^&*()_+?:"{})
3. Kode yang tidak ada di keyboard namun dapat ditampilkan. Kode ini umumnya untuk kode-kode grafik.

Dalam pengkodean kode ASCII memanfaatkan 8 bit. Pada saat ini kode ASCII telah tergantikan oleh kode UNICODE (Universal Code). UNICODE dalam pengkodeannya memanfaatkan 16 bit sehingga memungkinkan untuk menyimpan kode-kode lainnya seperti kode bahasa Jepang, Cina, Thailand dan sebagainya.

### 2.4. *Short Message Service (SMS)*

*Short Message Service (SMS)* merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel/nirkabel. Proses pengiriman pesan dalam bentuk *alphanumeric* antara terminal pelanggan dengan sistem *eksternal* seperti *email*, *paging*, *voice mail*, dan lain-lain. *SMS* pertama kali dikenalkan pada tanggal 3 Desember 1982. *SMS* pertama di dunia dikirimkan menggunakan jaringan *GSM* milik operator telepon bernama Vodafone. *SMS* pertama ini dikirimkan oleh ahli bernama Neil Papwort kepada Richard Jarvis menggunakan computer. [1]

### 2.5. Android

Android adalah sebuah sistem operasi telepon seluler dan komputer *tablet* layar sentuh (*touchscreen*) yang berbasis Linux. Namun seiring perkembangannya Android berubah menjadi sebuah *platform* yang begitu cepat dalam melakukan inovasi dan perkembangan. Hal tersebut tentu saja tidak dapat dilepaskan dari pengembang utama dibelakangnya, yaitu *Google Inc*. Sistem operasi Android tersedia secara bebas *(open-source)* bagi manufaktur perangkat keras untuk memodifikasinya sesuai kebutuhan. Meskipun konfigurasi perangkat Android tidak sama antara satu perangkat dengan perangkat lainnya, namun Android sendiri mendukung *fitur-fitur* berikut [5] :

1. Penyimpanan *(Storage), m*enggunakan SQLite yang merupakan *database relational* yang ringan untuk menyimpan data.
2. Koneksi *(connectivity), m*endukung *GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth* (termasuk *A2DP* dan *AVRCP*), *WiFi*, *LTE* dan *WiMAX*.
3. Pesan *(Messaging), m*endukung *Short Message Service (SMS)* dan *Multi-media Message Service (MMS).*
4. *Web Browser, m*enggunakan *open-source* WebKit termasuk di dalamnya *engine* Google Chrome V8 *Java-Script*.
5. Media yang didukung antara lain : H.263, H.264 (3GP atau MP4 *container*), MPEG-4 SP, AMR, AMR-WB (3GP *container*), AAC,

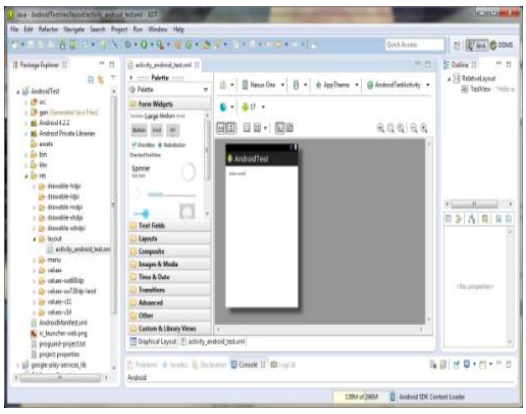
HE-ACC (MP4 atau 3GP *container*), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, BMP.

6. *Hardware* menggunakan, *Accelerometer* Sensor, *Camera*, *Digital Compass*, *Proximity* Sensor dan *GPS*.
7. *Multi-touch yaiut m*endukung layar banyak sentuhan.

### 2.7. Eclipse *IDE*

Sebuah *Integrated Development Environment (IDE)* untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*). Fungsi dari Eclipse *IDE* adalah sebagai berikut:

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



Gambar 2. Halaman Kerja Eclipse

Eclipse pada saat ini merupakan salah satu *IDE* favorit dikarenakan bebas *(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* (Cinar, 2012). Versi terbaru dari Eclipse *IDE* adalah Eclipse 4.4 (Luna) yang sudah mendukung integrasi Java versi 8 dapat dilihat pada Gambar 2.

## 3. METODE PENELITIAN

Dalam penelitian ini yang pertama dilakukan ialah bagaimana mengembangkan konsep sistem yang akan dibuat untuk mengatasi masalah-masalah. Setelah itu baru dilakukan proses perancangan yang meliputi proses pemilihan algoritma yang akan digunakan untuk memenuhi kubutuhan penelitian. Lakukan analisa kebutuhan dalam kebutuhan-kebutuhan *software* untuk menunjang jalannya aplikasi, sehingga dapat memberikan pengamanan yang optimal dan dapat digunakan dengan baik oleh penggunanya, setelah itu lakukan desain atau tampilan antarmuka pada aplikasi berbasis android.

Pengembangan pembuatan aplikasi enkripsi dan dekripsi pesan singkat menggunakan Eclipe IDE untuk membuat tampilan menu dan sistem. Setelah selasai dalam mendisain dan mengembangkannya aplikasi lakukan integrasi dan pengujian. Jika pengunjiannya gagal, maka kembali ketahap desain dan pengembangan. Jika berhasil, maka peneltian yang dibuat telah selesai.

## 4. PERANCANGAN

Perancangan yang akan dilakukan dalam penelitian ini meliputi perancangan perangkat lunak antara lain *use-case diagram*, perancangan antarmuka, perancangan proses enkripsi dan proses dekripsi.

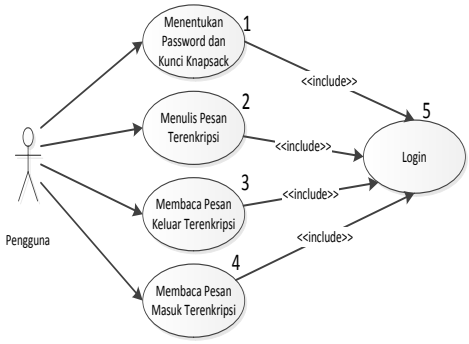
### 4.1. *Use-Case Diagram* pada Sistem

Dalam tahap perancangan digunakan *use-case diagram* untuk menggambarkan fungsionalitas sistem yang dilakukan oleh aktor. Pengguna dapat melakukan beberapa proses dengan syarat telah login yaitu dengan memasukan *password* pada saat aplikasi dimulai. *Use-case*

 ISSN 2338-493X

*diagram* sistem ditunjukan pada Gambar 3. Proses yang dilakukan pengguna adalah:

1. Menentukan password yang digunakan untuk login dan menentukan kunci privat, nilai m dan n sebagai metode kriptografi Knapsack.
2. Pengguna menulis pesan yang akan dienkripsi pada aplikasi.
3. Pengguna dapat membaca pesan terenkripsi yang telah terkirim pada pesan keluar aplikasi.
4. Pengguna dapat membaca pesan terenkripsi yang diterima pada pesan masuk aplikasi.



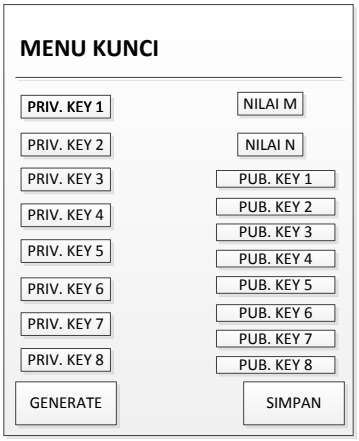

Gambar 3. *Use-case diagram*

## 4.2. Perancangan Antarmuka

Antarmuka merupakan penghubung antara sistem dan pengguna. Pada perancangan antarmuka dapat dibuat sketsa tampilan dari menu aplikasi. Tampilan menu utama pada aplikasi ini dapat dilihat pada Gambar 4. Rancangan tersebut memiliki 4 tombol yaitu tombol kunci, tulis pesan, tombol kotak masuk, tombol kotak keluar dan tombol tentang aplikasi.

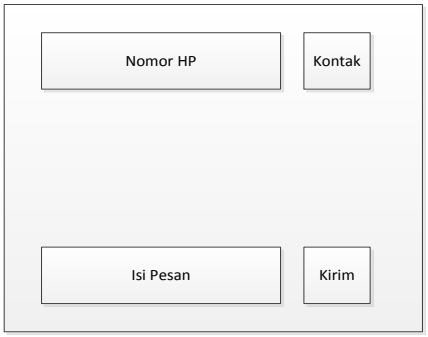

Gambar 4. Perancangan Tampilan Menu Utama

Tombol tulis pesan akan menampilkan halaman untuk membuat pesan. Kotak masuk dan kontak keluar akan berisi pesan yang masuk dan keluar. Tentang Aplikasi akan menampilkan fungsi dari aplikasi ini dibuat. Tombol kunci berfungsi untuk mengatur kunci yang digunakan dalam aplikasi. Halaman dari menu kunci dapat dilihat pada Gambar 5 berisi konfigurasi 8 kunci privat, kunci publik dan ketetapan nilai *m* dan *n* yang berguna sebagai sumber data dalam melakukan enkripsi dan dekripsi.




Gambar 5. Perancangan Tampilan Menu Utama

Ketika pengguna akan menulis pesan maka akan tampil seperti Gambar 6 yaitu form untuk mengirim pesan. Bagian dari form tersebut adalah "Nomor HP" yaitu berfungsi sebagai nomor yang dikirim pesan terenkripsi oleh penerima. Kontak akan memuat Nomor HP yang telah tersimpan pada Android. Pesan akan diisikan pada *field* "Isi Pesan" dan akan dikirim ketika pengguna menekan tombol "Kirim".

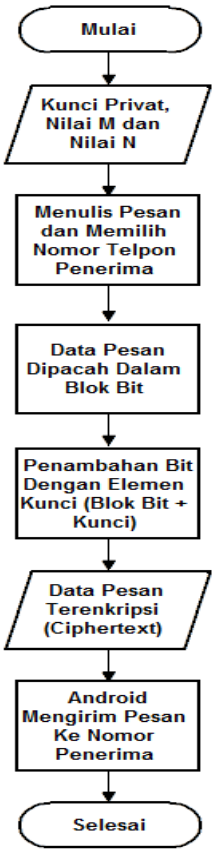



Gambar 6. Perancangan Tampilan Tulis Pesan

 ISSN 2338-493X

### 4.3. Perancangan Proses Enkripsi

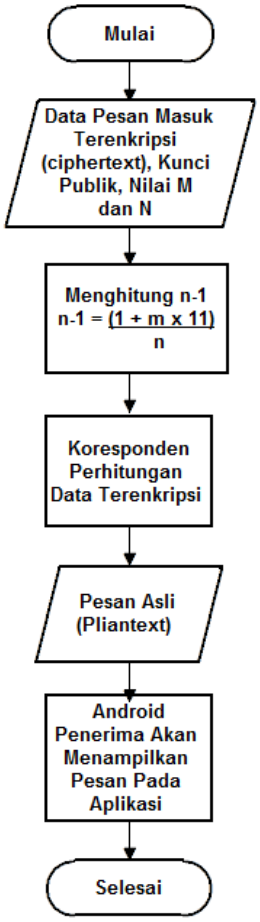
Enkripsi dilakukan sebelum mengirim pesan, saat pesan telah dituliskan dan hendak dikirim maka pesan akan dipecah menjadi blok-blok bit yang panjangnya sama dengan kunci privat yang telah ditetapkan didalam aplikasi. Lalu setiap anggota dari blok bit akan dikalikan dengan element kunci privat sehingga menghasilkan *cipher-text*. Diagram alir dari proses enkripsi dapat dilihat pada Gambar 7.




Gambar 7. Flowchart Enkripsi Pesan

### 4.4. Perancangan Proses Dekripsi

Pesan yang diterima akan masuk dalam kotak masuk dan saat membuka pesan tersebut akan dilakukan proses Dekripsi. Dari kunci privat akan cari nilai pembalik dari n yaitu $n^{-1}$. Lalu pesan yang tersandi dikalikan dengan $n^{-1}$ dan didapat sisa bagi (mod) dengan nilai m yang telah ditetapkan didalam aplikasi. Kemudian hasilnya di-korespondenkan dengan kunci privat yang juga sudah ditentukan. Hasil koresponden berupa blok-blok bit dengan panjang yang sama dengan kunci privat. Dari blok-blok bit dikonversi menjadi pesan asli seperti pada awal enkripsi. Diagram alir dari proses dekripsi dapat dilihat pada Gambar 8.




Gambar 8. Flowchart Dekripsi Pesan

## 5. HASIL DAN PEMBAHASAN

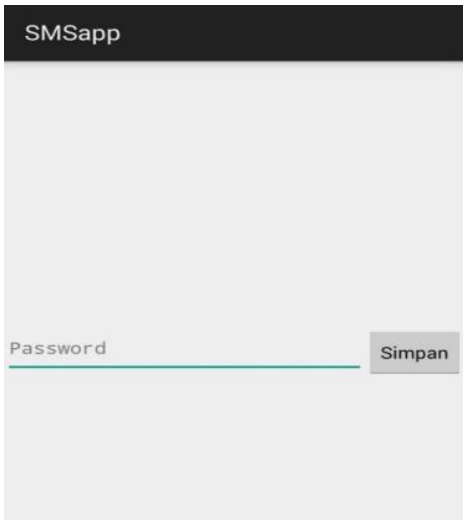
Hasil dan pembahasan yang akan dilakukan dalam penelitian ini meliputi Menentukan password dan kunci knapsack, halaman utama aplikasi, pengiriman dan enkripsi pesan, dan penerimaan dan dekripsi pesan pada aplikasi.

### 5.1. Menentukan password dan kunci knapsack

Setelah proses perancangan, maka dalam penelitian ini dibuatlah aplikasi berbasis Android sebagai proses imple-mentasi. Pada Gambar 9. terdapat tampilan dari menu awal setelah aplikasi berhasil di pasang yaitu meminta password untuk

 ISSN 2338-493X

masuk kedalam aplikasi. Fungsi *password* ini adalah sebagai pengamanan dari penggunaan aplikasi tanpa izin pemilik *smartphone* Android.




Gambar 9. Tampilan Awal Sebagai Pembuat *Password* Masuk

Setelah dilakukan penyimpanan data *password* pada *database*, maka proses selanjutnya pengguna harus memasukan kode kunci *increasing* untuk algoritma Knapsack. Secara mudah pengguna dapat menekan tombol generate sehingga aplikasi dapat membuat kunci increasing secara otomatis. Ketika data telah ditetapkan pengguna, kemudian pengguna harus menekan tombol simpan untuk menyimpan data kunci. Tampilan halaman ini dapat dilihat pada Gambar 10.

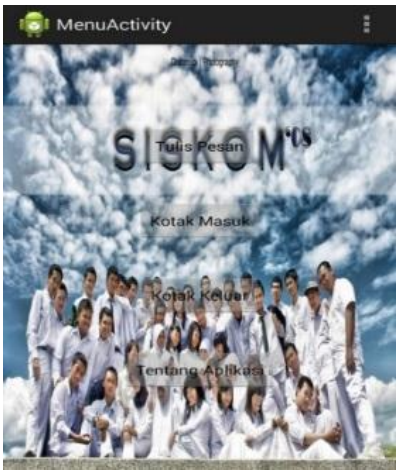

Gambar 10. Tampilan Menu Membuat Kunci Knapsack

## 5.2. Halaman utama

Ketika aplikasi baru dibuka, maka pengguna harus memasukan *password* masuk yang telah dibuat pada Gambar 9. Tetapi pada aplikasi tidak menampilkan pesan *error* sehingga pengguna tidak dapat mengecek apabila terjadi kesalahan *password*. Tampilan dari halaman menu masuk dapat dilihat pada Gambar 11.


Gambar 11. Tampilan Halaman Menu Masuk Aplikasi

Setelah dilakukan proses autentikasi pada *database* password, jika berhasil maka akan tampil halaman menu utama yang dapat dilihat pada Gambar 12.




Gambar 12. Tampilan Halaman Menu Utama

## 5.3. Pengiriman dan enkripsi pesan

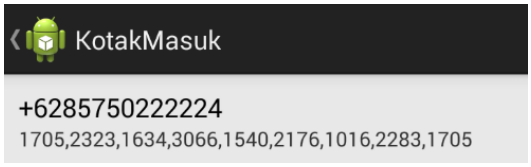
Untuk menggunakan aplikasi sebagai pembuat pesan terenkripsi, maka pengguna harus menekan tombol “Tulis Pesan” pada Gambar 5.4. Kemudian akan tampil halaman

 ISSN 2338-493X

untuk menulis *SMS*. Pengguna harus memasukan nomor telpon dari penerima pesan dan memasukan pesan. Setelah pengguna telah mengisikan nomor dan pesan yang akan di enkripsi, maka pengguna harus menekan tombol "Kirim". Pada halaman ini aplikasi tidak menampilkan atau memuat kontak dari telepon secara otomatis, sehingga pengguna harus memasukan nomor telepon secara manual.

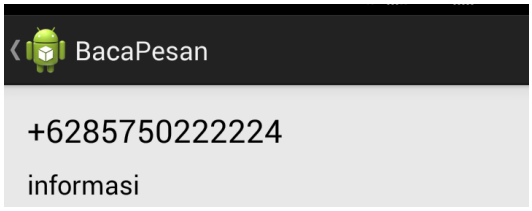
### 5.4. Penerimaan dan dekripsi pesan

Untuk membaca teks yang terenkripsi, sebagai penerima pesan akan masuk kedalam kotak masuk. Tampilan dari halaman menu kotak masuk dapat dilihat pada Gambar 13. Pesan yang terenkripsi berupa kode-kode yang tidak memiliki arti dan tidak dapat dimengerti oleh penerima pesan. Pada pesan yang tidak terenkripsi akan tampil sebagaimana pesan aslinya dan dapat dibaca secara langsung, berbeda dengan pesan yang telah dienkripsi, penerima harus memilih pesan yang berbentuk kode untuk menampilkan pesan asli *(plaintext)* dari pesan terenkripsi *(cipher-text).*




Gambar 13. Tampilan Pesan Keluar Yang Telah Terenkripsi

Pengguna dapat membaca pesan yang telah di enkripsi dengan menekan data pesan yang terenkripsi yaitu Gambar 14. Sehingga akan tampil teks yang dapat dimengerti oleh pengguna dan sama dengan teks yang dikirim pada Gambar 13.




Gambar 14. Tampilan Baca Pesan Yang Telah Terkirim

## 6. KESIMPULAN

Setelah dilakukan proses perancangan dan implementasi dalam penelitian ini, maka dapat disimpulkan bahwa:

1. Pada pada penelitian ini telah berhasil dibuat aplikasi Android dengan menggunakan algoritma kriptografi Knapsack dalam pembuatan aplikasi enkripsi dan dekripsi untuk pesan singkat atau *Short Message Service (SMS).*
2. Aplikasi dapat melakukan enkripsi dan dekripsi dengan dengan baik menggunakan *smartphone* Android. Hal ini dibuktikan dengan proses pengiriman dan penerimaan yang dikirim secara utuh.
3. Hasil dari teks pesan dapat berubah ketika penerima tidak menggunakan kunci yang sama dengan kunci pengirim sehingga kerahasiaan pesan dapat terjaga dengan baik.

## 7. Saran

Berdasakan penelitian yang telah dilakukan pada aplikasi enkripsi dan dekripsi pesan singkat menggunakan Algoritma Knapsack berbasis Android maka diperoleh saran untuk penelitian lebih lanjut yaitu:

1. Memperhatikan desain *User Interface (UI)* dan kelengkapan data berupa waktu dan tanggal pada pesan sehingga lebih mudah digunakan.
2. Membuat notifikasi *error* dan proses verifikasi data ketika pengguna salah memasukan kunci atau *password.*
3. Mengintergrasikan nomor telepon pada aplikasi dengan nomor kontak yang disimpan pada *smartphone* Android, sehingga pengguna dapat dengan mudah mengetahui identitas nomor kontak pada menu pengiriman dan penerimaan.

## DAFTAR PUSTAKA

[1] Pranarelza, R. (2014). Implementasi Algoritma Rijndael Untuk Enkripsi Dan Dekripsi Pesan Sms Pada Smartphone Berbasis Android. Jurnal Teknik Informatika STMIK El Rahma Yogyakarta.

[2] Schneier, B. (1995). Applied Cryptography : Protocols, Algorithms,

and Source Code in C, Second Edition : John Wiley.
[3] Dafid. (2006). *Kriptografi Kunci Simetris Dengan Menggunakan Algoritma Crypton*. STMIK MDP Palembang. Vol 2, No 3, Hal. 20-27.
[4] Munir, R. (2006). *Kriptografi.* Bandung: Informatika.
[5] Suprianto, A. (2012). Pemrograman Aplikasi Android. Yogyakarta: MediaKom.
[6] Ariyus. (2008). *Pengantar Ilmu Kriptografi: Teori, Analisa, dan implementasi*. Yogyakarta: Penerbit Andi.