PESAN RAHASIA DENGAN METODE KRIPTOGRAFI *ELGAMAL* PADA PERANGKAT *ANDROID MOBILE*

SKRIPSI

Oleh

Hasan Jindan
NIM. 102410101039

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015

PESAN RAHASIA DENGAN METODE KRIPTOGRAFI *ELGAMAL* PADA PERANGKAT *ANDROID MOBILE*

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Program Studi Sistem Informasi (S1) dan mencapai gelar Sarjana Sistem Informasi

Oleh:

Hasan Jindan
NIM. 102410101039

PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2015

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang senantiasa melimpahkan kenikmatan, kelancaran serta kemudahan dalam pengerjaan skripsi ini.
2. Keluarga tercinta; my father Alm. Ali Djindan, my mom Anisah Jindan, kakak Muhammad (Mamad) dan Soraiya (Ayak) serta adik Fatimah (Iim).
3. Bapak dan Ibu Dosen Pembimbing; Bapak Prof. Drs. Slamin, M.Comp.Sc.,Ph.D dan Ibu Nelly Oktavia Adiwijaya, S.Si.,M.T yang selalu memberikan bantuan, bimbingan, dukungan dan semangat dalam mengerjakan skripsi ini, Bapak Yanuar selaku Dosen Pembimbing Akademik. Dan Alm. Bapak Puguh yang saya senangi atas kelembutannya saat mengajar.
4. Keluarga dari awal kuliah; Yusuf (Ucup), Apyu (Emmak), Alvin (Lancip), Brill (Wahid), Andre (De Bamba), Yoga (Ganyok), Yudha (Kentus), Hendri (Jorok), Friendy (Singo), Iwan, Dymas (Doyok), Gomay, Rio, Arbi (Maji), Hawwin (Wingswood), Gayatri (Poyeng), Kikik Hawwin, Brian (Bom), Aang, Ruroh, Kebal, Saddam, Adong, Faizal, Joe, Doci, Hamdan, Syafiq, Yusa, Zizi, Nay, Erik, Angga, Dhevi, Nindi, Affan, Levi, Kresna, Brilly, Brian, Glenn, Afendi, Ocha, Riski Vadilla, Marcelli, Mbak Ifrina, Mas Vanda, Roni, Roqib, Musa, Peter, Dido, Lely, Kadek, Awang, Ridwan, Umam, Mas Sugeng, Mas Holili, Mas Chandra, Pak Khobir, Mas Imam, Mas Wahyu, Mas Dwi, Mas Dika, Mas Novi, Mas Darwis, Rusdi, Spesialis H., dan teman sekampus yang lain yang tidak dapat saya sebutkan semua.
5. Teman seperjuangan di kosan lama; Mas Angga, A'ak, Mas Baim, Mas Joe, Mas Sandy, Jono, Cebok, Bejo, Gepeng, Japrax, Rival, Fauzi, Rama, dan teman yang lain.
6. Semua orang yang pernah berkenalan yang memberi banyak pelajaran dan mendewasakan saya pentingnya bersosialisasi dengan memperbanyak pertemanan.
7. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTTO

"Success is always accompanied with failure."

(Annonymous)

"Tidak perlu menunggu untuk bisa menjadi cahaya bagi orang-orang di sekelilingmu. Lakukan kebaikan, sekecil apapun sekarang juga."

(Andy F. Noya)

"You may never know what results come of your action, but if you do nothing, there will be no results."

(**Mahatma Gandhi)**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Hasan Jindan

NIM : 102410101039

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Pesan Rahasia dengan Metode Kriptografi *Elgamal* pada Perangkat *Android Mobile*" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada intitusi mana pun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, November 2015

Yang menyatakan,

Hasan Jindan

NIM. 102410101039

SKRIPSI

PESAN RAHASIA DENGAN METODE KRIPTOGRAFI *ELGAMAL* PADA PERANGKAT *ANDROID MOBILE*

Oleh:
Hasan Jindan
NIM. 102410101039

Menyetujui

Pembimbing Utama,	Pembimbing Anggota,
Prof. Drs. Slamin, M.Comp.,Sc.,Ph.D	Nelly Oktavia Adiwijaya, S.Si.,M.T.
NIP. 196704201992011001	NIP. 198410242009122008

PENGESAHAN

Skripsi berjudul "Pesan Rahasia dengan Metode Kriptografi Elgamal pada Perangkat Android Mobile", telah diuji dan disahkan pada:

hari, tanggal : Senin, 2 November 2015

tempat : Program Studi Sistem Informasi Universitas Jember

Penguji 1,	Penguji 2,
Dr. Saiful Bukhori, ST.,M.Kom	Yanuar Nurdiansyah ST,.M.Cs.
NIP. 196811131994121001	NIP. 198201012010121004

Mengesahkan

Ketua Program Studi Sistem Informasi,

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D

NIP. 196704201992011001

RINGKASAN

Pesan Rahasia dengan Metode Kriptografi *Elgamal* pada Perangkat *Android Mobile*; Hasan Jindan, 102410101039; 2015; 80 halaman; Program Studi Sistem Informasi Universitas Jember.

Sering terjadinya kerumitan dalam pengiriman dan penyimpanan pesan terkirim yang tidak ingin diketahui oleh pihak tertentu. Hal ini dikarenakan karena tidak sedikit juga pihak-pihak yang tidak bertanggung jawab yang ingin mengetahui isi dari pesan rahasia tersebut dan disalah gunakan terhadap hal yang tidak baik. Tentunya memerlukan cara untuk mengatasi permasalahan dalam menyimpan pesan rahasia. Pada penelitian ini Pesan Rahsia berbasis *Android mobile* ini menggunakan "Metode Kriptografi Elgamal" karena Algoritma ini akan memberikan *public key* serta *private key* yang digunakan dalam proses Enkripsi dan Dekripsi. Dalam proses pembentukan *public key* dan *private key*, akan dibutuhkan suatu bilangan prima yang bernilai besar agar menjadi aman. Berdasarkan penelitian dan pembuatannya aplikasi ini mampu membantu mengamankan pesan rahasia agar tidak mudah diakses dan digunakan pihak lain.

PRAKATA

Alhamdulillah, segala puji kepada Allah SWT atas segala nikmat dan karunia-Nya sehingga penulis mampu menyelesaikan skripsi yang berjudul "Pesan Rahasia dengan Metode Kriptografi Elgamal pada Perangkat Android Mobile". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, penulismenyamaikan terimakasih kepada:

1. Prof. Drs. Slamin, M.Comp.Sc.,Ph.D selaku Ketua Program Studi Sistem Informasi serta Dosen Pembimbing Utama dan Nelly Oktavia Adiwijaya, S.Si.,M.T selaku Dosen Pembimbing Anggota dan Dosen Pembimbing Akademik yang telah memberikan bantuan, dukungan, dan semangat dalam pengerjaan skripsi ini.
2. Bapak Ibu Dosen beserta staf karyawan Program Studi Sistem Informasi Universitas Jember.
3. Keluarga tercinta; my father Alm. Ali Djindan, my mom Anisah Jindan, kakak Muhammad (Mamad) dan Soraiya (Ayak) serta adik Fatimah (Iim).
4. Keluarga dari awal kuliah; Yusuf (Ucup), Apyu (Emmak), Alvin (Lancip), Brill (Wahid), Andre (De Bamba), Yoga (Ganyok), Yudha (Kentus), Hendri (Jorok), Friendy (Singo), Iwan, Dymas (Doyok), Gomay, Rio, Arbi (Maji), Hawwin (Wingswood), Gayatri (Poyeng), Kikik Hawwin, Brian (Bom), Aang, Ruroh, Kebal, Saddam, Adong, Faizal, Joe, Doci, Hamdan, Syafiq, Yusa, Zizi, Nay, Erik, Angga, Dhevi, Nindi, Affan, Levi, Kresna, Brilly, Brian, Glenn, Afendi, Ocha, Riski Vadilla, Marcelli, Mbak Ifrina, Mas Vanda, Roni, Roqib, Musa, Peter, Dido, Lely, Kadek, Awang, Ridwan, Umam, Mas Sugeng, Mas Holili, Mas Chandra, Pak Khobir, Mas Imam, Mas Wahyu, Mas Dwi, Mas Dika, Mas Novi, Mas Darwis, Rusdi, Spesialis H., dan teman sekampus yang lain yang tidak dapat saya sebutkan semua.

5. Teman seperjuangan di kosan lama; Mas Angga, A'ak, Mas Baim, Mas Joe, Mas Sandy, Jono, Cebok, Bejo, Gepeng, Japrax, Rival, Fauzi, Rama, dan teman yang lain.
6. Semua orang yang pernah berkenalan yang memberi banyak pelajaran dan mendewasakan saya pentingnya bersosialisasi dengan memperbanyak pertemanan.
7. Almamater Program Studi Sistem Informasi Universitas Jember.Program Studi Sistem Informasi.
8. Semua pihak yang memberikan dorongan dan semangat yang tidak dapat disebutkan satu per satu.

Dengan harapan bahwa penelitian ini nantinya akan terus dapat dikembangkan, penulis menerima kritik dan saran dari semua pihak demi perbaikan dan kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, November 2015

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1. PENDAHULUAN

Bab 1 merupakan bab yang berisi latar belakang suatu penelitian itu diambil, rumusan masalah, batasan masalah, tujuan serta manfaat dari penelitian yang dilakukan.

1.1 Latar Belakang

Pada era modern saat ini BBM merupakan salah satu media berkirim pesan yang sangat digandrungi dan banyak digunakan terutama oleh masyarakat Indonesia. Blackberry Messenger tercatat digunakan oleh sebanyak 79 persen orang di Indonesia. Jumlah ini menjadi yang terbanyak setelah WhatsApp dengan mendapatkan 54 persen pengguna. Demikian hasil riset dari *On Device Meter* oleh *Nielsen* pada Februari 2014 lalu. Dengan adanya teknologi ini memudahkan kita dalam hal berkomunikasi antara satu sama lain dengan jarak jauh dan waktu yang singkat.

Seiring dengan banyaknya orang yang menggunakan BBM sebagai alat komunikasi, beberapa orang berpikir pesan mereka boleh diketahui pihak lain, tapi tidak sedikit orang yang mengirimkan pesan yang sifatnya rahasia dan hanya boleh diketahui oleh orang-orang tertentu saja. Tapi tidak sedikit juga pihak-pihak yang tidak bertanggung jawab yang ingin mengetahui isi dari pesan rahasia tersebut dan disalah gunakan terhadap hal yang tidak baik. Tentunya memerlukan cara untuk mengatasi permasalahan dalam menyimpan pesan rahasia.

Proses pengamanan pesan rahasia telah dilakukan penelitian sebelumnya dengan metode kriptografi dalam jurnal "Jurnal Analisis dan Perbandingan Kecepatan Algoritma RSA dan Algoritma *Elgamal*", Nikolaus Indra, Mei 2011. Bahwa terdapat algoritma simetris yang hanya menggunakan sebuah kunci dalam melakukan proses enkripsi dekripsi dapat memungkinkan informasi yang dirahasiakan dapat bocor, dan algoritma asimetris dengan kunci *public* yang lebih menjamin keamanan informasi.

Ketidakamanannya penyebaran kunci melalui saluran komunikasi membuat para kriptografer berpikir lebih keras untuk menemukan algoritma-algoritma baru yang dapat digunakan tanpa harus mempermasalahkan pengiriman melalui saluran komunikasi yang tidak aman tersebut. Hingga akhirnya pada tahun 1976 muncul suatu sistem kriptografi baru, yaitu kriptografi kunci publik. Hingga saat ini ada tiga algoritma kriptografi kunci publik yang sering digunakan yaitu RSA, *Elgamal*, dan Rabin (Jurnal Perbandingan Algoritma Kriptografi Kunci Publik RSA, Rabin, dan *Elgamal* oleh Maureen Linda Caroline).

Jurnal-jurnal tersebut, menyatakan bahwa untuk melindungi isi pesan rahasia tersebut agar tidak mudah terbaca oleh orang lain adalah lebih aman dengan menggunakan sebuah algoritma yang dapat menyembunyikan pesan dengan menggunakan *public key* dan *private key* dari pada hanya menggunakan *private key* saja. Salah satu algoritma yang dapat menyembunyikan pesan dengan *public key* adalah algoritma *Elgamal*. Algoritma ini akan memberikan *public key* serta *private key* yang digunakan dalam proses Enkripsi dan Dekripsi. Dalam proses pembentukan *public key* dan *private key*, akan dibutuhkan suatu bilangan prima yang bernilai besar agar menjadi aman. Pada penelitian ini implementasi metode *Elgamal* akan dibangun pada perangkat mobile berbasis Android.

1.2 Rumusan Masalah

Dalam proses penerapan pesan rahasia dengan metode kriptografi *Elgamal* pada *android mobile*, ditemukan beberapa permasalahan sebagai berikut:

1. Bagaimana penerapan metode *Elgamal* dengan proses enkripsi dan deskripsi pesan.
2. Bagaimana membangun aplikasi berbasis android yang dapat mengamankan pesan rahasia dengan metode *Elgamal*.

1.3 Batasan Masalah

Beberapa batasan masalah dalam penerapan pesan rahasia dengan metode kriptografi *Elgamal* pada *android mobile*:

1. Dataset yang digunakan yakni berupa data sms yang berisi pesan.
2. Aplikasi ini dibuat pada mobile berbasis android.

1.4 Tujuan

Tujuan dari penerapan pesan rahasia dengan metode kriptografi *Elgamal* pada *android mobile* yaitu:

1. Menerapkan algoritma *Elgamal* pada proses enkripsi dan deskripsi pesan.
2. Mengembangkan aplikasi yang dapat menyimpan pesan.

1.5 Manfaat

Manfaat dari penerapan pesan rahasia dengan metode kriptografi *Elgamal* pada *android mobile*:

1. Manfaat Akademis

   Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.
2. Manfaat bagi peneliti dan objek penelitian
   a) Mengetahui bagaimana proses penerapan metode kriptografi *Elgamal*.
   b) Membantu mengamankan pesan rahasia agar tidak mudah diakses dan digunakan pihak lain.
   c) Memberikan inovasi baru kepada siapa saja mengenai pengamanan pesan rahasia.

1.6 Sistematika Penulisan

Sistematika penulisan untuk tugas akhir ini adalah sebagai berikut:

1. Pendahuluan

   Bab pendahuluan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat dari penelitian serta sistematika penulisan.

2. Tinjauan pustaka

   Berisi materi dan informasi yang digunakan dalam penyelesaian penelitian. Selain itu juga terdapat teori-teori yang digunakan dalam penulisan penelitian.

3. Metodologi penelitian

   Berisi metode yang akan digunakan dalam penelitian mulai dari tahap pengumpulan data, pengembangan desain sistem, implementasi dan evaluasi sistem.

4. Perancangan Sistem

   Tahapan perancangan dan desain yang kan dikembangkan untuk membuat sistem dijelaskan pada bab ini.

5. Hasil dan pembahasan

   Menjelaskan tentang hasil dan pembahasan dari penelitian yang sudah dikembangkan.

6. Kesimpulan dan Saran

   Berisi tentang kesimpulan dari penelitian yang telah dikembangkan serta saran untuk penelitian yang akan dilakukan selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Bab ini berisi tinjaun pustaka atau kajian teori yang melandasi penelitian yang dilakukan oleh penulis.

2.1 Kriptografi

Kriptografi, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita [Bruce Schneier - Applied Cryptography]. Selain pengertian tersebut terdapat pula pengertian ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data [A. Menezes, P. van Oorschot and S. Vanstone - Handbook of Applied Cryptography]. Tidak semua aspek keamanan informasi ditangani oleh kriptografi.

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-repudiasi, atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/ terciptanya suatu informasi oleh yang mengirimkan/membuat.

Ilmu kriptografi suatu pesan yang akan dirahasiakan atau disandikan disebut dengan plaintext, sedangkan pesan yang telah disandikan sehingga tidak memiliki nilai dan arti lagi dengan tujuan agar pesan tidak dapat dibaca oleh pihak yang tidak berhak disebut *chipertext*. Dalam ilmu kriptografi juga terdapat istilah enkripsi dan deskripsi. Enkripsi merupakan proses menyandikan plaintext menjadi *chipertext* dengan menggunakan algoritma tertentu. Sedangkan proses mengembalikan *chipertext* menjadi plaintext disebut sebagai Deskripsi.

2.2 Algoritma Kriptografi

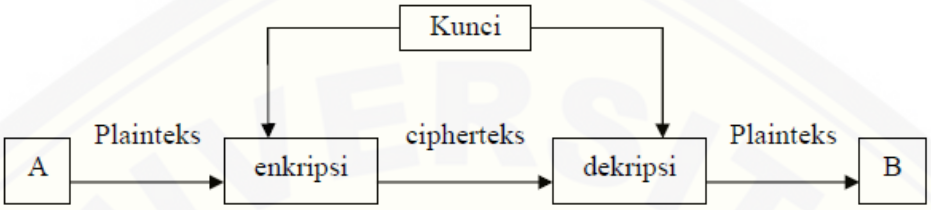
Algoritma kriptografi atau sering disebut dengan *cipher* adalah suatu fungsi matematis yang digunakan untuk melakukan enkripsi dan dekripsi (Schneier, 1996). Ada dua macam algoritma kriptografi, yaitu *algoritma simetris* (*symmetric algorithms*) dan *algoritma asimetris* (*asymmetric algorithms*).

2.2.1 Algoritma Simetris

Algoritma simetris adalah algoritma kriptografi yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya. Algoritma ini mengharuskan pengirim dan penerima menyetujui suatu kunci tertentu sebelum mereka saling berkomunikasi. Keamanan algoritma simetris tergantung pada kunci, membocorkan kunci berarti bahwa orang lain dapat mengenkripsi dan mendekripsi pesan. Agar komunikasi tetap aman, kunci harus tetap dirahasiakan.

Algoritma simetris sering juga disebut dengan algoritma kunci rahasia, algoritma kunci tunggal, atau algoritma satu kunci. Sifat kunci yang seperti ini membuat pengirim harus selalu memastikan bahwa jalur yang digunakan dalam pendistribusian kunci adalah jalur yang aman atau memastikan bahwa seseorang yang ditunjuk membawa kunci untuk dipertukarkan adalah orang yang dapat dipercaya. Masalahnya akan

menjadi rumit apabila komunikasi dilakukan secara bersama-sama oleh sebanyak *n* pengguna dan setiap dua pihak yang melakukan pertukaran kunci, maka terdapat sebanyak kunci rahasia yang harus dipertukarkan secara aman. Alur algoritma sismetris ini dapat dilihat pada gambar 2.1.



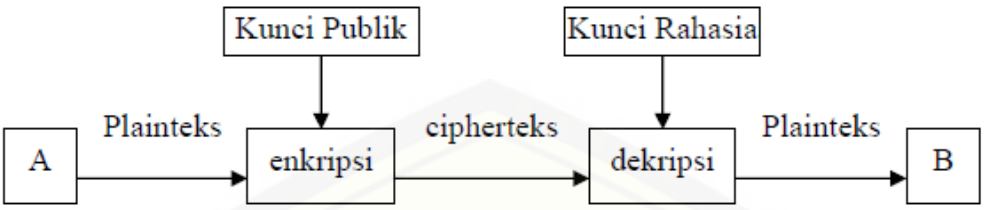
Gambar 2.1: Skema algoritma simetris (Muhamad Zaki, 2007)

Contoh dari algoritma kriptografi simetris adalah Cipher Permutasi, Cipher Substitusi, Cipher Hill, OTP, RC6, Twofish, Magenta, FEAL, SAFER, LOKI, CAST, Rijndael (AES), Blowfish, GOST, A5, Kasumi, DES dan IDEA.

2.2.2 Algoritma Asimetris

Algoritma asimetris, sering juga disebut dengan *algoritma kunci publik*, menggunakan dua jenis kunci, yaitu *kunci publik* (*public key*) dan *kunci rahasia* (*secret key*). Kunci publik merupakan kunci yang digunakan untuk mengenkripsi pesan. Sedangkan kunci rahasia digunakan untuk mendekripsi pesan. Kunci publik bersifat umum, artinya kunci ini tidak dirahasiakan sehingga dapat dilihat oleh siapa saja. Sedangkan kunci rahasia adalah kunci yang dirahasiakan dan hanya orang-orang tertentu saja yang boleh mengetahuinya.

Keuntungan utama dari algoritma ini adalah memberikan jaminan keamanan kepada siapa saja yang melakukan pertukaran informasi meskipun di antara mereka tidak ada kesepakatan mengenai keamanan pesan terlebih dahulu maupun saling tidak mengenal satu sama lain. Alur algoritma asismetris ini dapat dilihat pada gambar 2.2.



Gambar 2.2: Skema algoritma asimetris (Muhamad Zaki, 2007)

Algoritma asimetris pertama kali dipublikasikan oleh Diffie dan Hellman pada tahun 1976 dalam papernya yang berjudul *"New Directions in Cryptography"*. Menurut Diffie dan Hellman, ada beberapa syarat yang perlu diperhatikan pada algoritma asimetris, yaitu:

1. Penerima B membuat pasangan kunci, yaitu kunci publik *pB k* dan kunci rahasia *rB k* .
2. Pengirim A dengan kunci publik B dan pesan *x*, pesan dienkripsi dan diperoleh cipherteks.
3. Penerima B untuk mendekripsi cipherteks menggunakan kunci privat B untuk mendapatkan kembali pesan aslinya.
4. Dengan mengetahui kunci publik *pB k*, bagi penyerang akan kesulitan dalam melakukan untuk mendapatkan kunci rahasia.

Contoh dari algoritma asimetris adalah RSA, *Elgamal*, McEliece, LUC dan DSA (*Digital Signature Algorithm*). Dalam melakukan proses enkripsi, sering digunakan plainteks berupa data ataupun pesan yang besar, sehingga membutuhkan waktu yang lama apabila dilakukan proses sekaligus pada plainteks tersebut. Oleh karena itu, plainteks dapat dipotong-potong menjadi beberapa blok-blok yang sama panjang. Kemudian dari blok-blok yang diperoleh tersebut dilakukan proses enkripsi, dan hasil cipherteksnya dapat didekripsi dan digabungkan kembali menjadi plainteks. Algoritma kriptografi yang menggunakan mekanisme seperti ini disebut dengan cipher blok (*block cipher*).

2.3 Algoritma *Elgamal*

Algoritma *Elgamal* ditemukan oleh ilmuwan Mesir, yaitu Taher *Elgamal* pada tahun 1985, merupakan algoritma kriptografi kunci publik. Algoritma *Elgamal* terdiri atas tiga proses, yaitu proses pembentukan kunci, enkripsi, dan dekripsi. Algoritma *Elgamal* mendasarkan kekuatannya pada fakta matematis kesulitan menghitung logaritma diskrit.

1. Pembentukan Kunci

Skema *Elgamal* memerlukan sepasang kunci yang dibangkitkan dengan memilih sebuah bilangan prima p dan dua buah bilangan random g dan x. Nilai g dan x lebih kecil dari p yang memenuhi persamaan :

$$y = g^x \bmod p$$

Dari persamaan tersebut y, g dan p merupakan kunci publik dan x adalah kunci rahasia.

2. Proses Enkripsi

Proses enkripsi merupakan proses mengubah pesan asli (plaintext) menjadi pesan rahasia (ciphertext). Pada proses ini digunakan kunci publik (p, g, y). Langkah-langkah dalam mengenkripsi pesan adalah sebagai berikut.

i. Potong plaintext menjadi blok-blok m1, m2,…., nilai setiap blok di dalam selang [0, p – 1].
ii. Ubah nilai blok pesan ke dalam nilai ASCII (*American Standard Code for Information Interchange*).
iii. Pilih bilangan acak k, dengan syarat $1 \leq k \leq p - 2$.
iv. Setiap blok m dienkripsi dengan rumus sebagai berikut.
    gamma (γ) = $g^k \bmod p$
    delta (δ) = $y^k m \bmod p$
v. Susun ciphertext dengan urutan $\gamma_1, \delta_1, \gamma_2, \delta_2, \ldots, \gamma_n, \delta_n$.

Pasangan γ dan δ adalah cipherteks untuk blok pesan m. Hasil yang didapat dari proses enkripsi berupa pesan rahasia (ciphertext).

3. Proses Dekripsi

Proses dekripsi merupakan proses mengubah pesan rahasia (ciphertext) menjadi pesan asli (plaintext). Pada proses ini menggunakan kunci pribadi (x, p).

Langkah-langkah dalam mendekripsi pesan adalah sebagai berikut.

i. Penentuan nilai gamma dan delta. Nilai gamma (γ) diperoleh dari ciphertext dengan urutan ganjil sedangkan delta (δ) dengan urutan genap.

ii. Hitung plaintext m dengan persamaan rumus berikut.

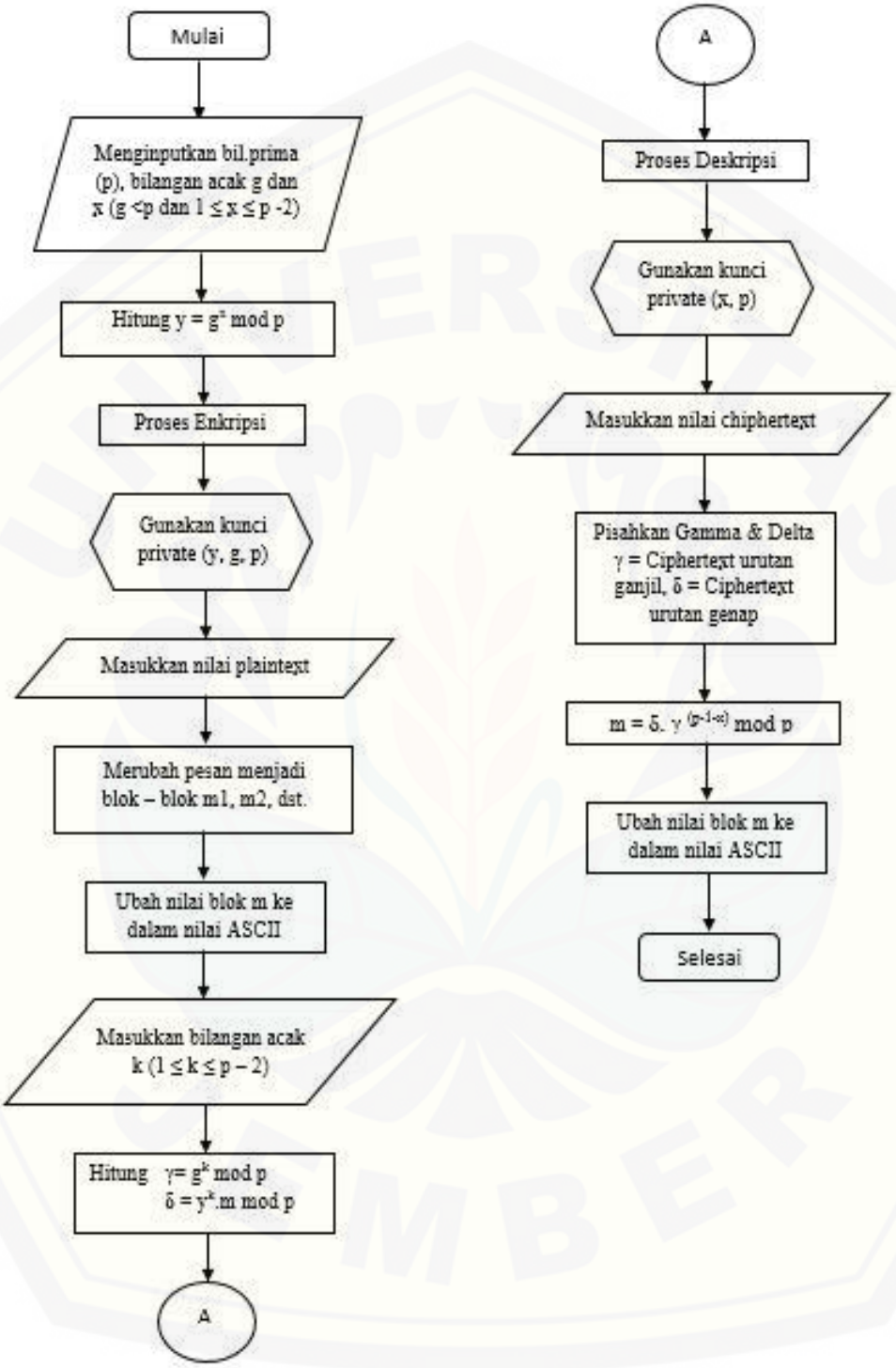
$m = \delta . \gamma^{(p-1-x)} \bmod p$.

iii. Ubah nilai m yang didapat kedalam nilai ASCII (*American Standard Code for Information Interchange*).

iv. Susun plaintext dengan urutan $m_1, m_2, ..., m_n$.

v. Hasil yang didapat dari proses deskripsi tersebut berupa pesan asli (plaintext).

Flowchart algoritma *elgamal* dengan tahap – tahap pembentukan kunci, proses enkripsi, proses deskripsi dapat dilihat pada gambar 2.3.

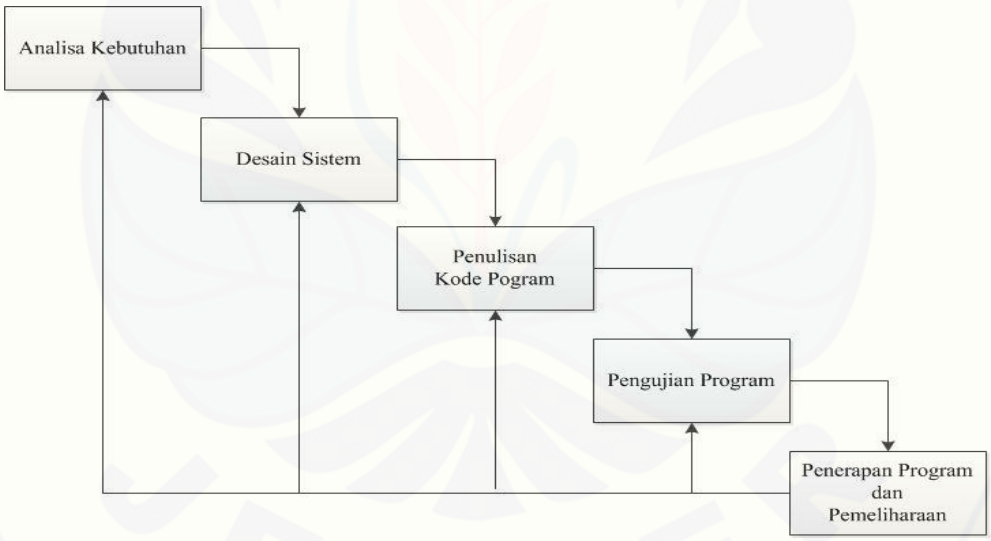


Gambar 2.3: Flowchart Algoritma *Elgamal*

2.4 Android

Menurut Nazrudin Safaat H, Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan elekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh *Android Inc*, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh *Google Inc*.



Gambar 2.4: Tahapan Metode *Waterfall* (Kadir, 2003)

2.5 Model Waterfall

Pembuatan perancangan perangkat lunak ini menggunakan metode *waterfall*. Metode *waterfall* merupakan metode yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada anailisia kebutuhan, desain sistem, penulisan kode program, pengujian program, dan penerapan program serta pemeliharaan (Kadir, 2003). Pembuatan desain sistem menggunakan *Unified Modeling Languange* (*UML*). Pemodelan UML yang digunakan pada penelitian ini antara lain, *Use Case Diagram*, *Use Case Scenario*, *Sequence Diagram*, *Activity Diagram*, dan *Class diagram* seperti pada gambar 2.4.

BAB 3. METODE PENELITIAN

Pada bab ini akan dibahas tentang metode yang digunakan oleh penulis dalam pembuatan Sistem Pesan Rahasia pada *Android mobile* menggunakan Metode Kriptografi *Elgamal*. Terdiri dari tahapan pengumpulan data dan studi literatur, jenis penelitian yang digunakan, model perancangan sistem, analisis kebutuhan, desain perancangan dan pengkodean serta implementasi sistem.

3.1 Jenis Penelitian

Pada penelitian ini termasuk jenis penelitian kualitatif karena penelitian ini diperoleh melalui studi literatur mengenai metode *elgamal*, android, dan kriptografi. Dengan menganalisa studi literatur yang berhubungan dengan indikator untuk membuat pesan rahasia.

3.2 Analisa Data

Dalam penerapan metode kriptografi *Elgamal* ini, dilakukan beberapa tahapan-tahapan seperti gambar 3.1:

Gambar 3.1: Skema algoritma *elgamal*

1. Pembentukan Kunci

   Tahap awal dari proses algoritma *Elgamal* dengan pembentukan kunci *public key* dan *private key*, dimana memerlukan proses penentuan bilangan prima p serta nilai g dan x untuk memperoleh nilai y sesuai rumus yang terdapat dalam metode ini. Dalam proses ini nilai x merupakan kunci rahasia dan nilai p, g, dan y merupakan kunci publik.

2. Proses Enkripsi

   Proses enkripsi merupakan proses mengubah pesan asli (plaintext) menjadi pesan rahasia (ciphertext). Pada proses ini menggunakan kunci publik (p, g, y) untuk merubahnya.

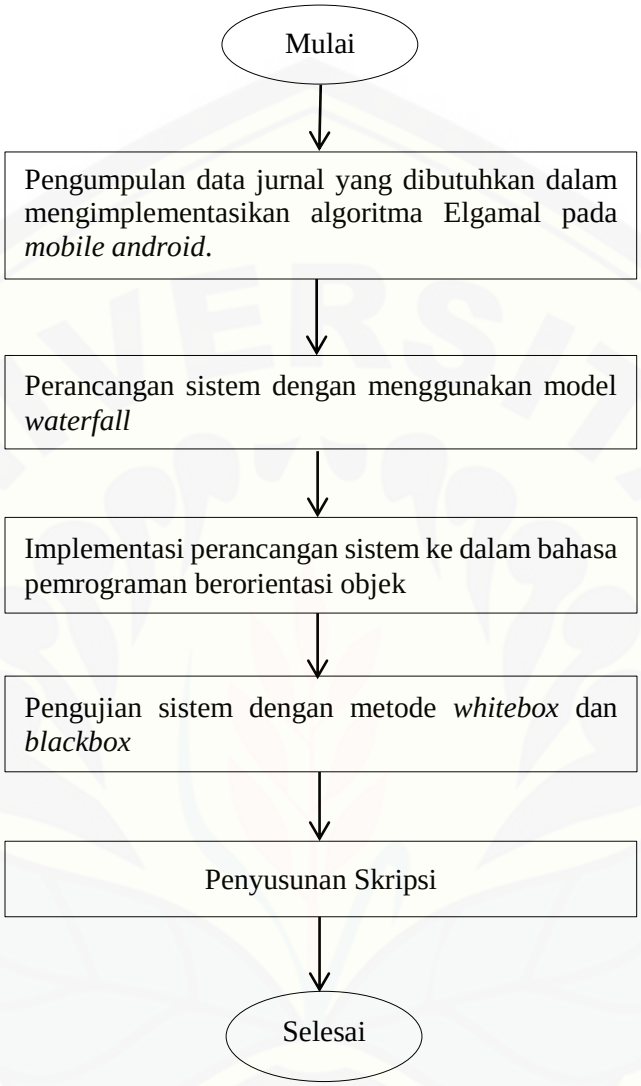
3. Proses Deskripsi

   Proses deskripsi merupakan proses mengubah pesan rahasia (ciphertext) menjadi pesan asli (plaintext). Pada proses ini menggunakan kunci *private* (x, p) untuk merubahnya kembali menjadi pesan asli.

3.3 Alur Penelitian

Alur Penelitian menjelaskan urutan penelitian yang akan dilakukan yaitu tahap pengumpulan data, tahap perancangan, tahap implementasi, tahap pengujian dan tahap penyusunan skripsi.

Diagram alir tahapan yang akan dilakukan dalam penelitian pengimplementasian algoritma Kriptografi *Elgamal* pada *android mobile* dapat dilihat pada Gambar 3.2.



Gambar 3.2: Diagram Alur Penelitian

3.3.1 Tahapan Pengumpulan Data dan Studi Literatur

Tahapan pengumpulan data dan studi literatur merupakan tahapan awal yang akan dilakukan dalam penelitian ini. Diawali dengan proses pencarian permasalahan pada objek penelitian, yang selanjutnya akan diberikan solusi-solusi dari permasalahan tersebut.

Pada tahap ini, diketahui bahwa permasalahan yang dihadapi pengguna pesan di bbm dalam penyimpanan atau pengiriman pesan rahasia masih belum dapat memudahkan pengguna dalam mengamankan informasi tersebut.

Dalam mengatasi permasalahan tersebut, maka peneliti berusaha untuk membangun *android mobile*. Adapun data yang dibutuhkan ialah data – data tentang kriptografi, android, dan algoritma *elgamal*. Data tersebut diperoleh dari jurnal – jurnal, buku yang terkait, website, artikel, dan penelitian terdahulu.

Studi literatur dibutuhkan guna mendapatkan gambaran menyeluruh mengenai apa yang telah dipelajari untuk menunjang pemahaman dan pengetahuan penulis tentang teori, konsep ilmu, dan metode yang digunakan dalam penelitian ini. Diharapkan aplikasi tersebut dapat membantu pengguna dalam mengamankan informasi pada pesan tersebut.

3.3.2 Tahapan Perancangan

Tahap perancangan merupakan tahapan yang dilakukan setelah tahapan pengumpulan data selesai dilakukan. Pada tahapan ini peneliti akan mulai membuat rancang bangun *android mobile* dengan menggunakan metode kriptografi *Elgamal*. Pada proses perancangan *android mobile* ini, akan digunakan model perancangan *waterfall*.

Perancangan sistem yang digunakan menggunakan konsep berbasis objek dengan pemodelan *Unified Modelling Language* (UML). Pemodelan UML yang digunakan pada penelitian ini antara lain, *Usecase Diagram*, *Usecase Scenario*, *Sequence Diagram*, *Activity Diagram*, *Class diagram* dan *Entity Relationship Diagram* (ERD). Perangkat lunak yang akan dibangun ini menggunakan bahasa pemrograman Java XML pada perangkat *android mobile* dengan menggunakan Eclipse.

3.3.3 Tahapan Implementasi

Tahapan implementasi yang dilakukan yaitu dengan cara memasang aplikasi mobile yang telah dibuat ke perangkat android, sehingga dapat digunakan langsung oleh penggunanya.

3.3.4 Tahapan Pengujian

Tahap pengujian dilakukan apabila aplikasi yang dibuat telah selesai dan siap untuk digunakan pengguna. Pengujian yang dilakukan berguna untuk mengetahui sejauh mana pengimplementasian algoritma Kriptografi *Elgamal* pada aplikasi *android mobile*.

Tahapan pengujian dilakukan dengan mencari kesalahan-kesalahan yang mungkin terjadi, serta melakukan perbaikan untuk menyempurnakan aplikasi *android mobile* dalam mengimplementasikan algoritma Kriptografi *Elgamal*. Proses pengujian dilakukan dengan metode *whitebox* oleh pengembang dan *blackbox* oleh pengguna. Pengujian *whitebox* dilakukan untuk mengetahui apakah aplikasi yang dibangun dari segi desain dan program sesuai dengan kebutuhan. Sedangkan untuk pengujian *blackbox* dilakukan hanya dengan memperhatikan masukan/keluaran (I/O) yang dihasilkan oleh aplikasi *android mobile.*

3.3.5 Tahapan Penyusunan Skripsi

Tahap penyusunan skripsi merupakan langkah akhir pada penelitian ini. Pada tahap ini akan dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam skripsi ini serta hasil dari implementasi algoritma Kriptografi *Elgamal* pada *android mobile*.

BAB 4. PERANCANGAN SISTEM

Pada bab ini akan diuraikan tentang desain perancangan serta implementasi dari perancangan Pesan Rahasia Dengan Metode Kriptografi *Elgamal* pada Perangkat *Android Mobile*.

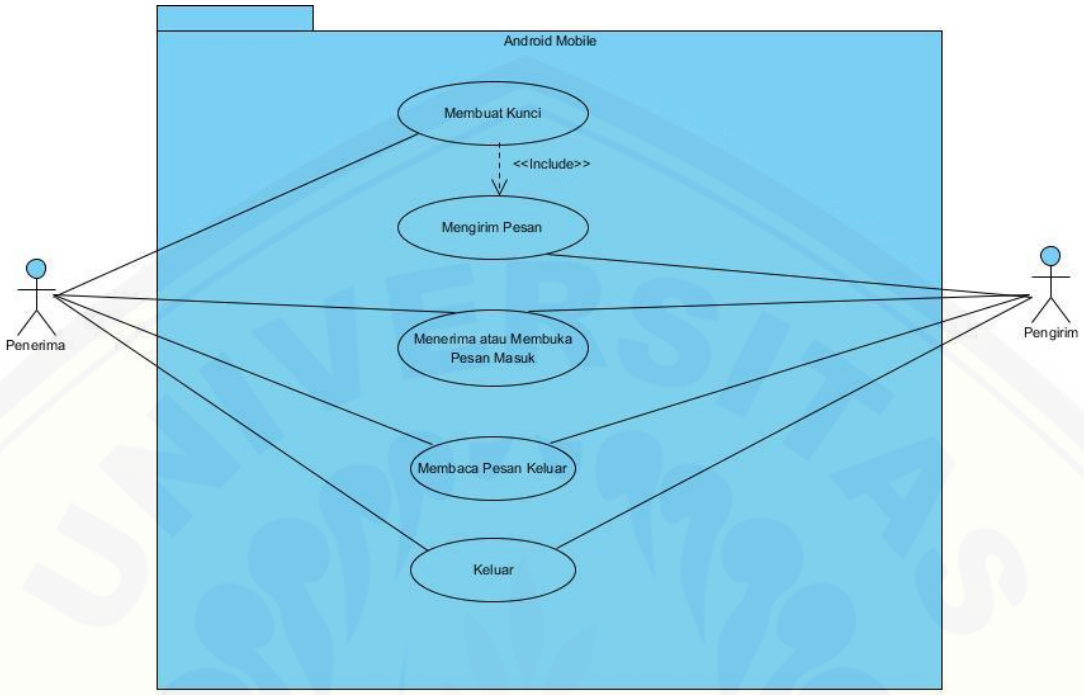
4.1 Deskripsi Umum Sistem

Sistem yang dibangun dalam penelitian ini adalah sistem berbasis *android mobile* untuk pengamanan pesan yang tidak ingin diketahui oleh pihak lain. Sistem ini difungsikan untuk membantu penerima untuk pengamanan pesannya dengan pengirim. Sistem ini diharapkan benar-benar efisien dalam proses pengamanan kuncinya yang dimana terdapat 2 kunci yaitu *private key* yang dimiliki oleh penerima dan *public key* yang dimiliki oleh pengirim.

4.2 Usecase Diagram

Usecase diagram pada pengimplementasian algoritma *elgamal* di *android mobile*. *Usecase android mobile* yang berfungsi untuk menggambarkan fitur apa saja yang akan dijalankan pada aplikasi *android mobile* yang akan diimplementasikan algoritma *elgamal* di dalamnya. *Usecase android mobile* dapat dilihat pada Gambar 4.1.

Definisi *usecase* pada *usecase android mobile* dapat dilihat pada Tabel 4.1 sedangkan untuk definisi aktor yang ada pada *usecase android mobile* dapat dilihat pada Tabel 4.2.



Gambar 4.1: Usecase *android mobile*

Tabel 4.1: Definisi *usecase android mobile*

No	*Usecase*	Deskripsi
1	Membuat Kunci	Proses untuk melakukan buat kunci pengguna *android mobile*
3	Mengirim Pesan	Membuat pesan yang akan dikirim oleh pengguna *android mobile*
4	Menerima atau Membuka Pesan Masuk	Menampilkan daftar pesan masuk yang ada di pengguna *android mobile*
5	Membuka Pesan Keluar	Menampilkan daftar pesan keluar yang ada di pengguna *android mobile*
6	Keluar	Proses untuk mengakhiri aplikasi *android mobile*

Tabel 4.2: Definisi aktor *usecase android mobile*

No	Aktor	Deskripsi
1	Penerima	User yang mengoperasikan aplikasi *android mobile* untuk menerima pesan rahasia
2	Pengirim	User yang mengoperasikan aplikasi *android mobile* untuk mengirim pesan rahasia

4.3 *Usecase Scenario*

Usecase Scenario merupakan uraian dari *Usecase Diagram* sistem, dimana setiap *usecase* memiliki fungsi dan uraian dari setiap fitur yang ada pada sistem. Adapun *Usecase Scenario* pada Pesan Rahasia dengan Metode Kriptografi *Elgamal* pada Perangkat *Android mobile* ini ditunjukkan pada tabel berikut:

a. *Usecase scenario* Membuat Kunci

Usecase scenario buat kunci menjelaskan alur pembentukan kunci yang dilakukan penerima. Aktor melakukan aksi dan sistem akan melakukan reaksi. Berikut merupakan gambaran alur dari *usecase* membuat kunci:

Tabel 4.3: *Usecase scenario* Membuat Kunci

Nama	Membuat Kunci
Participating Actor	Penerima
Entry Condition	Sistem menampilkan form buat kunci
Exit Condition	Sistem menampilkan hasil *public key* dan *private key*
Skenario Normal	
1. Penerima memilih *button* buat kunci	
	2. Sistem menampilkan form buat kunci
3. Penerima menginputkan nilai p, g, x dan memilih *button* generate	
	4. Sistem menampilkan hasil *public key* dan *private key*
5. Penerima menginputkan nomer tujuan dan mengirim *public key*	
	6. Sistem mengirim pesan sesuai nomer tujuan
Skenario Alternatif (Membuat Kunci)	

3. Penerima tidak mengisi form p, g, dan x	
	4. Sistem menampilkan pesan "data belum terisi"
3. Penerima menginputkan nilai p bukan bilangan prima dan nilai p kurang dari 255	
	4. Sistem menampilkan pesan "p bukan bilangan prima" dan menampilkan pesan "p harus lebih besar dari 255"
3. Penerima mengitputkan nilai g dan x lebih besar dari nilai p	
	4. Sistem menampilkan pesan "g dan x harus lebih kecil dari p"

b. *Usecase scenario* Mengirim Pesan

Usecase scenario mengirim pesan menjelaskan alur pembuatan pesan dan pengiriman pesan yang dilakukan Pengirim. Aktor melakukan aksi dan sistem akan melakukan reaksi. Berikut merupakan gambaran alur dari *usecase* mengirim pesan:

Tabel 4.4: *Usecase scenario* Mengirim Pesan

Nama	Mengirim Pesan
Participating Actor	Pengirim
Entry Condition	Sistem menampilkan form pesan *public key* pada page pesan masuk
Exit Condition	Sistem menenkripsi pesan dan mengirimkan pesan pada penerima
Skenario Normal	
1. Pengirim memilih *button* pesan masuk	

	2. Sistem menampilkan page pesan masuk
3. Pengirim memilih pesan *public key* dari penerima	
	4. Sistem menampilkan form pesan *public key* dan menampilkan *public key* otomatis pada text box
5. Pengirim menulis pesan yang akan dikirim dan memilih *button* kirim	
	6. Sistem mengekripsi pesan dan mengirim pesan pada penerima
Skenario Alternatif (Mengirim Pesan)	
5. Pengirim memilih tombol button dan tidak mengisi pesan pada text box pesan	
	6. Sistem menampilkan pesan "pesan belum terisi"

c. *Usecase scenario* Menerima atau Membuka Pesan Masuk

Usecase scenario menerima atau membuka pesan masuk menjelaskan tentang pesan-pesan di pesan masuk dan pesan keluar *android mobile* aktor. Aktor melakukan aksi dan sistem akan melakukan reaksi. Berikut merupakan gambaran alur dari *usecase* menerima atau membuka pesan masuk:

Tabel 4.5: *Usecase scenario* Menerima atau Membuka Pesan Masuk

Nama	Menerima atau Membuka Pesan Masuk
Participating Actor	Penerima
Entry Condition	Sistem menampilkan page pesan masuk

Exit Condition	Sistem mendeskripsi pesan yang telah dipilih
Skenario Normal	
1. Penerima memilih *button* pesan masuk	
	2. Sistem menampilkan page pesan masuk
3. Penerima memilih pesan yang akan dibuka	
	4. Sistem menampilkan isi pesan yang dipilih oleh penerima
5. Penerima mengitputkan *private key* (p dan x) dan memilih *button decrypt*	
	6. Sistem menampilkan *plaintext* yang telah di deskripsi
Skenario Alternatif (Menerima atau Membuka Pesan)	
5. Penerima tidak mengisi nilai *private key* (p dan x)	
	6. Sistem menampilkan pesan “kunci masih kosong”

d. *Usecase scenario* Membaca Pesan Keluar

Usecase scenario membaca pesan keluar menjelaskan tentang pesan-pesan di kotak keluar *android mobile* aktor. Aktor melakukan aksi dan sistem akan melakukan reaksi. Berikut merupakan gambaran alur dari *usecase* pesan keluar:

Tabel 4.6: *Usecase scenario* Membaca Pesan Keluar

Nama	Membaca Pesan Keluar
Participating Actor	Penerima dan Pengirim
Entry Condition	Sistem menampilkan page pesan keluar
Exit Condition	Sistem menampilkan isi pesan yang telah dipilih
Skenario Normal	
1. Aktor memilih *button* pesan keluar	
	2. Sistem menampilkan page pesan keluar
3. Aktor memilih pesan yang akan dibaca atau dibuka	
	4. Sistem menampilkan isi pesan yang dipilih oleh aktor

e. *Usecase scenario* Keluar

Usecase scenario keluar menjelaskan tentang alur untuk keluar dari aplikasi *android mobile* aktor. Aktor melakukan aksi dan sistem akan melakukan reaksi. Berikut merupakan gambaran alur dari *usecase* keluar:

Tabel 4.7: *Usecase scenario* Keluar

Nama	Keluar
Participating Actor	Penerima dan Pengirim
Entry Condition	Sistem menampilkan pop up *button* keluar
Exit Condition	Semua aktifitas aplikasi *android mobile* ditutup
Skenario Normal	
1. Aktor memilih *button* keluar	

	2. Sistem menampilkan pop up *button* keluar
3. Aktor memilih *button* ya	
	4. Sistem menutup semua aktifitas pada aplikasi *android mobile*
5. Aktor memilih *button* tidak	
	6. Sistem menampilkan kembali aplikasi *android mobile*

4.4 *Activity Diagram*

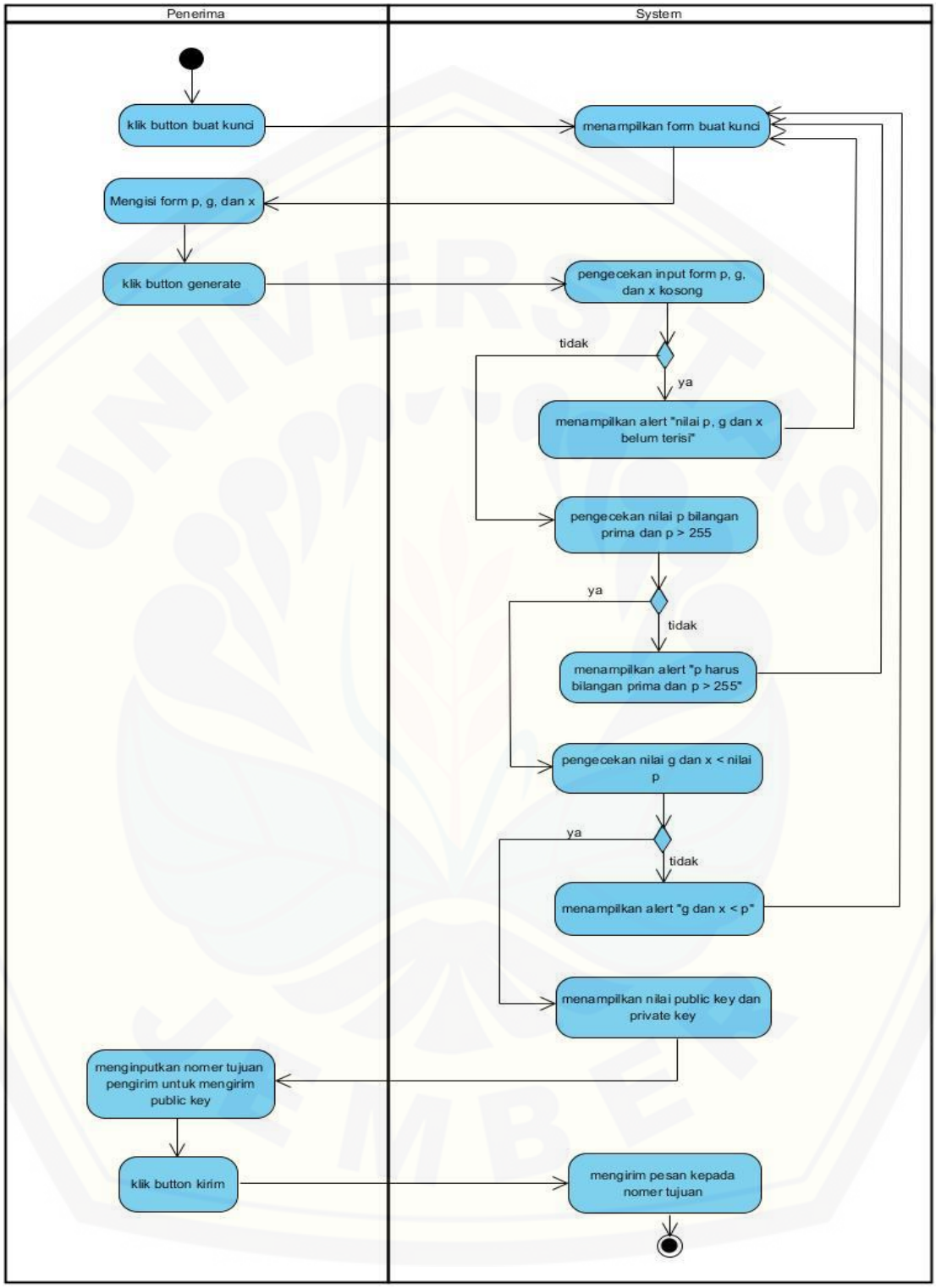
Activity Diagram pada pembangunan aplikasi algoritma *elgamal* di *android mobile* ini terbagi atas beberapa fitur yang digambarkan sesuai dengan *usecase diagram* dan *usecase scenario* yang telah dibuat. *Activity diagram* juga merupakan gambaran alur dari aksi *user* dan reaksi sistem pada saat aplikasi dijalankan.

a. *Activity Diagram* Membuat Kunci

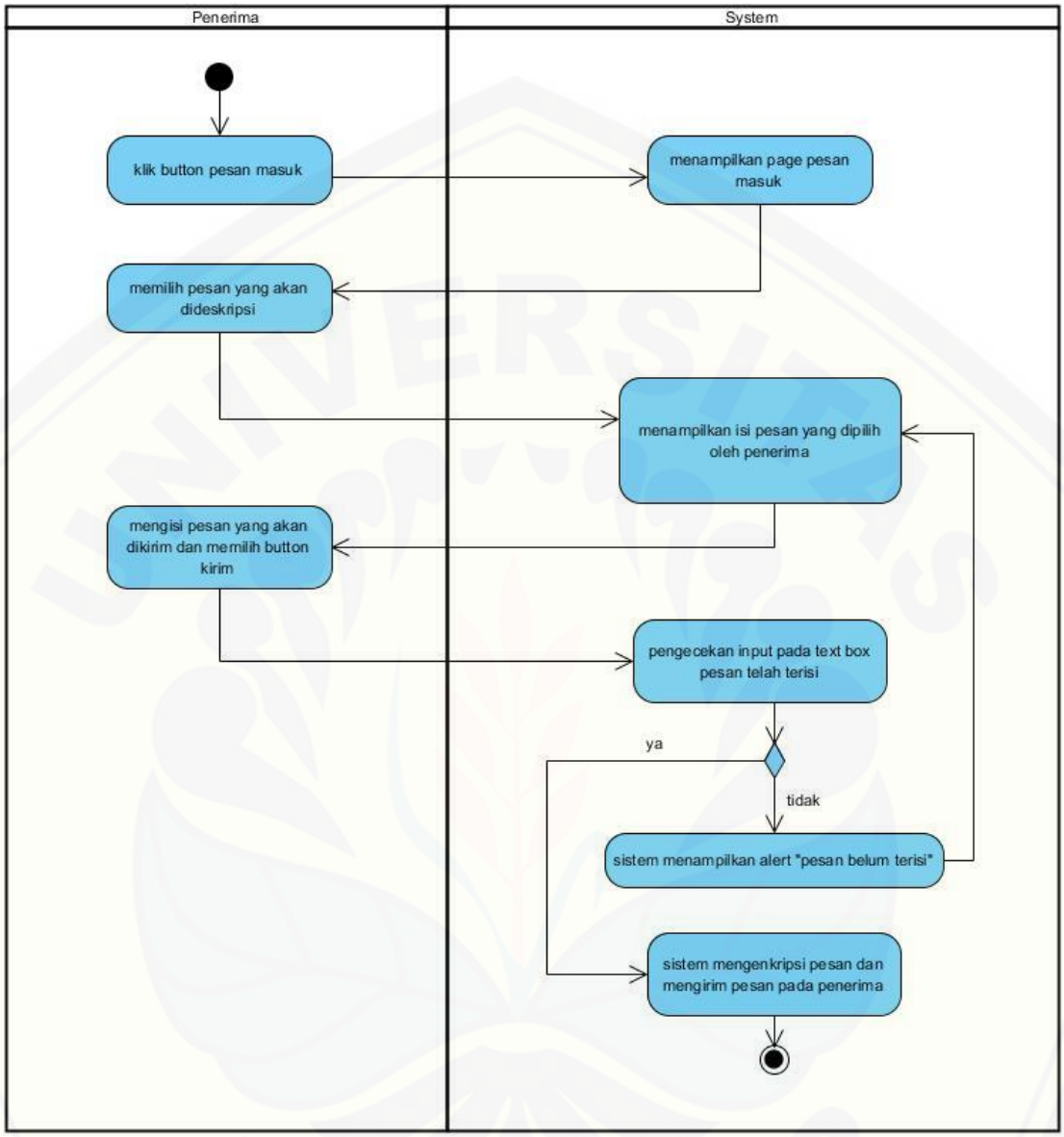
Seperti Gambar 4.2 di bawah ini menggambarkan mengenai proses Membuat Kunci yang terdapat dalam *android mobile*. Diawali dengan penerima memilih tombol buat kunci lalu menginputkan nilai p, g, dan x untuk menentukan nilai y. Kemudian sistem melakukan pengecekan dan jika sukses sistem menampilkan *private key* dan *public key*, lalu penerima mengirim *public key* pengirim dengan menginputkan nomer tujuan pengirim dan memilih tombol kirim. Setelah itu sistem mengirim pesan *public key* tersebut kepada nomer tujuan yang telah diinputkan penerima.

b. *Activity Diagram* Mengirim Pesan

Seperti Gambar 4.3 di bawah ini menggambarkan mengenai proses Mengirim Pesan yang terdapat dalam *android mobile*. Diawali dengan pengirim membuka isi pesan berupa *public key* dari penerima dengan memilih *button* pesan masuk dan memilih pesan tersebut. Kemudian sistem menampilkan form pesan dengan *public key* serta mereply nomer tujuan. Dan pengirim mengisi pesan tersebut dan memilih *button* kirim. Lalu sistem akan melakukan pengecekan dan jika sukses sistem mengenkripsi pesan dan mengirim pesan tersebut pada penerima.

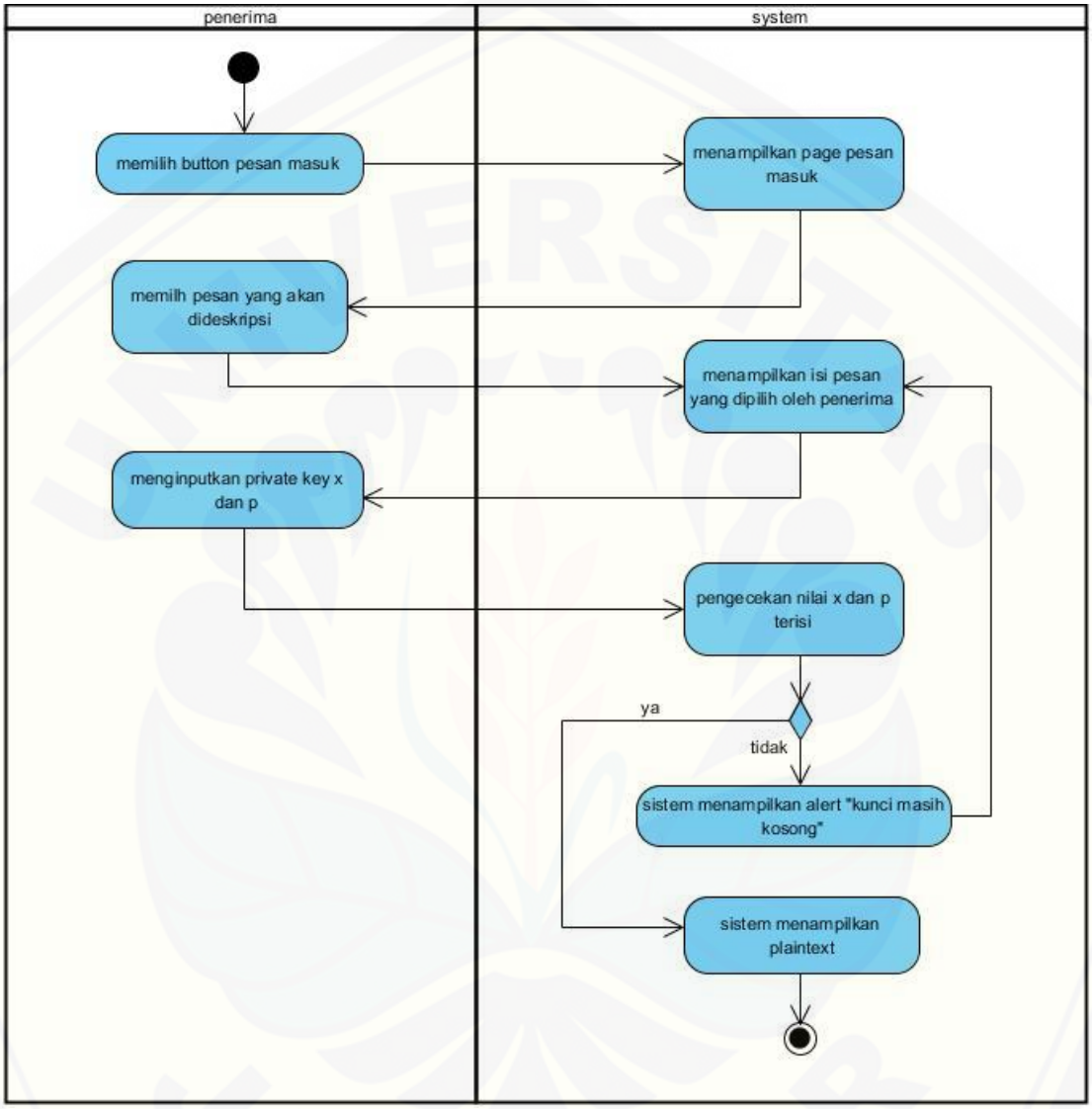


Gambar 4.2: *Activity Diagram* Membuat Kunci



Gambar 4.3: *Activity Diagram* Mengirim Pesan

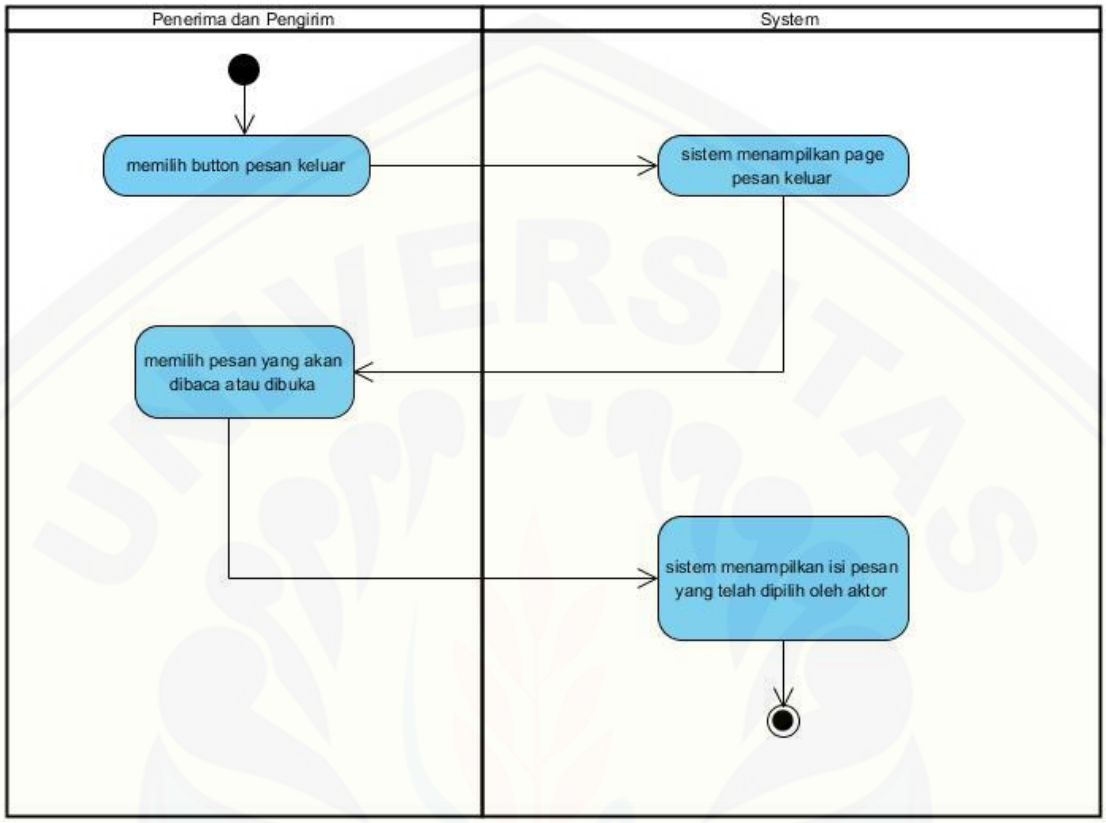
c. *Activity Diagram* Menerima atau Membuka Pesan Masuk



Gambar 4.4: Menerima atau Membuka Pesan Masuk

Gambar 4.4 menggambarkan mengenai proses Menerima atau Membuka Pesan Masuk yang terdapat dalam *android mobile*. Diawali dengan penerima membuka isi pesan yang akan dideskripsi dan menginputkan nilai x dan p. Kemudian sistem akan melakukan pengecekan dan jika sukses sistem mendeskripsi pesan dan menampilkan hasil plaintext.

d. *Activity Diagram* Membaca Pesan Keluar

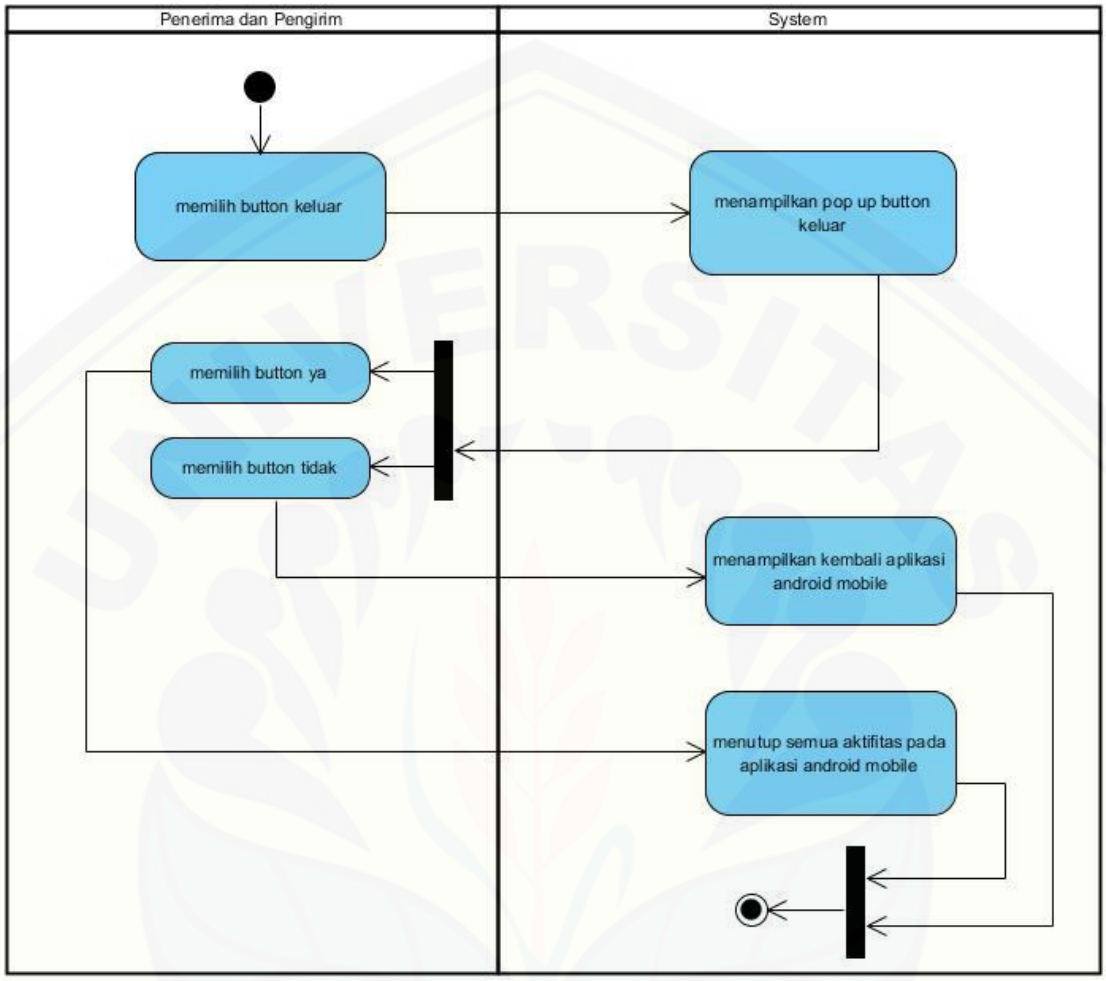


Gambar 4.5: *Activity Diagram* Membaca Pesan Keluar

Gambar 4.5 menggambarkan mengenai proses Membaca Pesan Keluar yang terdapat dalam *android mobile*. Diawali dengan penerima atau pengirim memilih *button* pesan keluar lalu memilih pesan yang akan dibaca atau dibuka. Kemudian sistem akan menampilkan isi pesan yang telah dipilih oleh penerima atau pengirim.

e. *Activity Diagram* Keluar

Gambar 4.6 menggambarkan mengenai proses Keluar yang terdapat dalam *android mobile*. Diawali dengan penerima atau pengirim memilih *button* keluar. Setelah itu sistem akan menampilkan pop up keluar. Lalu penerima atau pengirim memilih *button* ya atau *button* tidak. Kemudian sistem akan melakukan pengecekan dan jika penerima memilih *button button* ya maka sistem akan menutup semua aktifitas pada aplikasi *android mobile*. Jika aktor memilih *button* tidak maka sistem akan menampilkan kembali aplikasi *android mobile*.

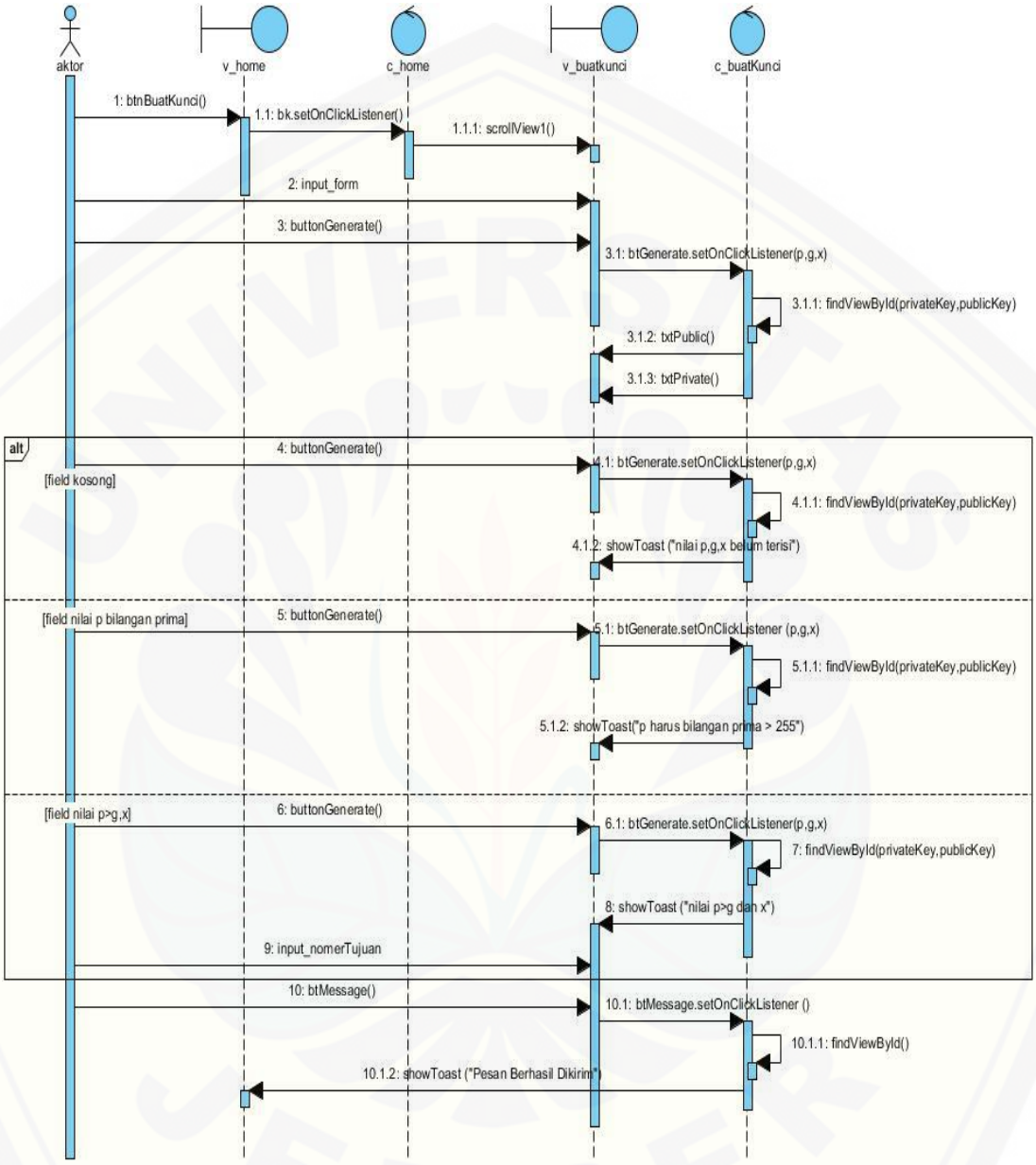


Gambar 4.6: *Activity Diagram* Keluar

4.5 *Sequence Diagram*

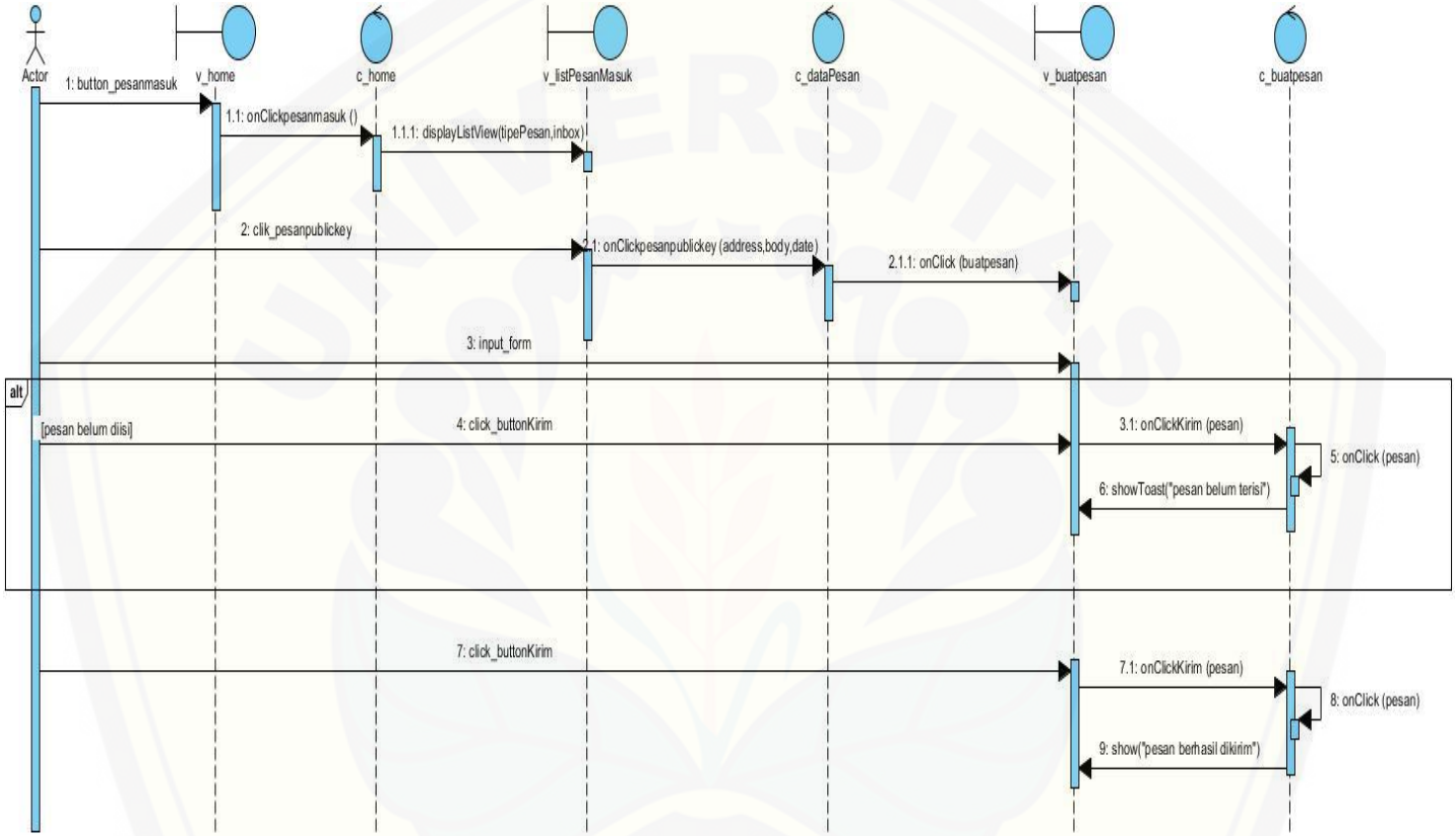
Proses *perancangan* selanjutnya yaitu dengan membuat *sequence diagram*. *Sequence diagram* ini dibuat sesuai dengan *activity diagram* yang sudah ada dan juga disesuaikan dengan fitur pada aplikasi yang dibangun.

a. *Sequence Diagram* Membuat Kunci



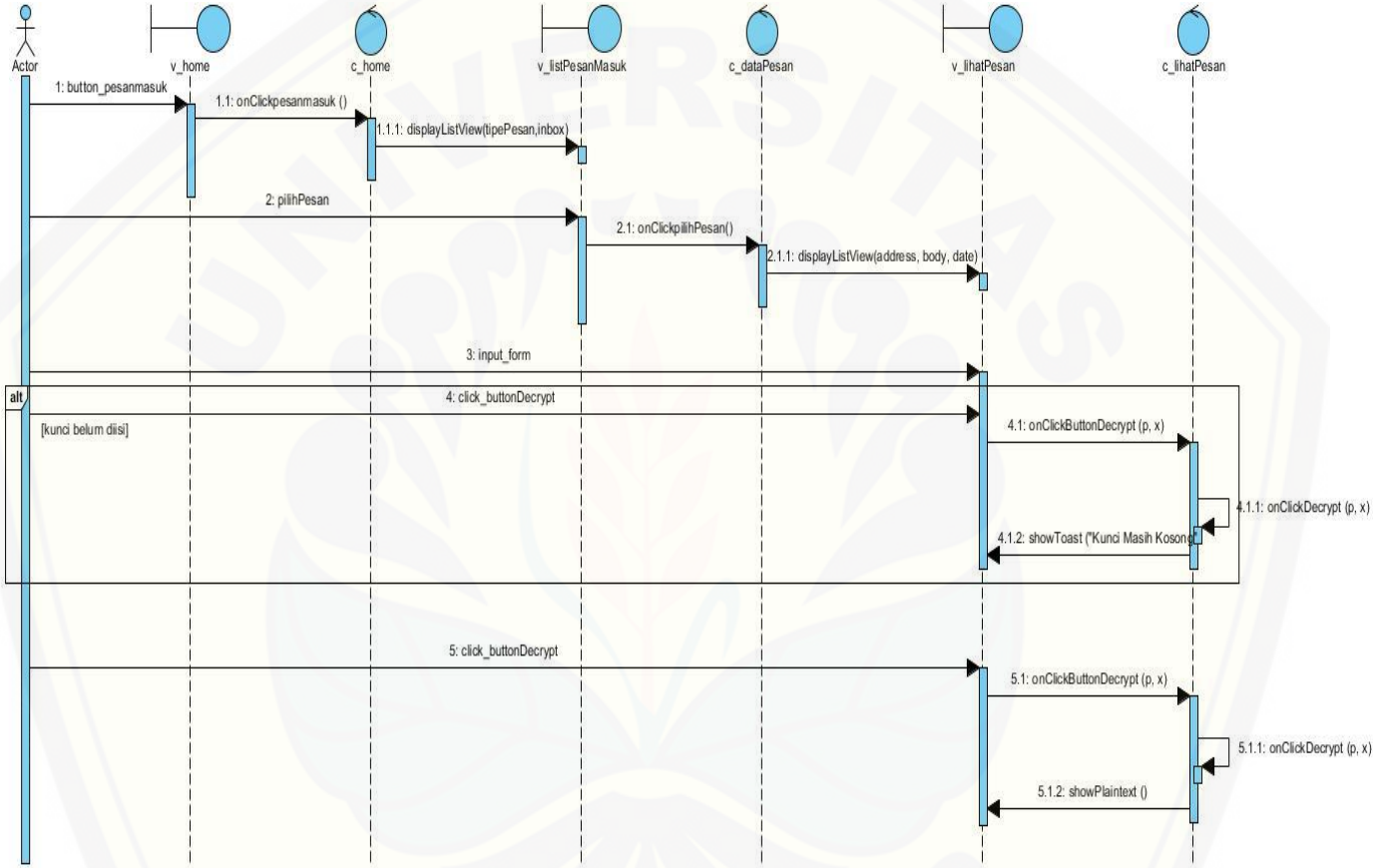
Gambar 4.7: *Sequence* Membuat Kunci

b. *Sequence Diagram* Mengirim Pesan



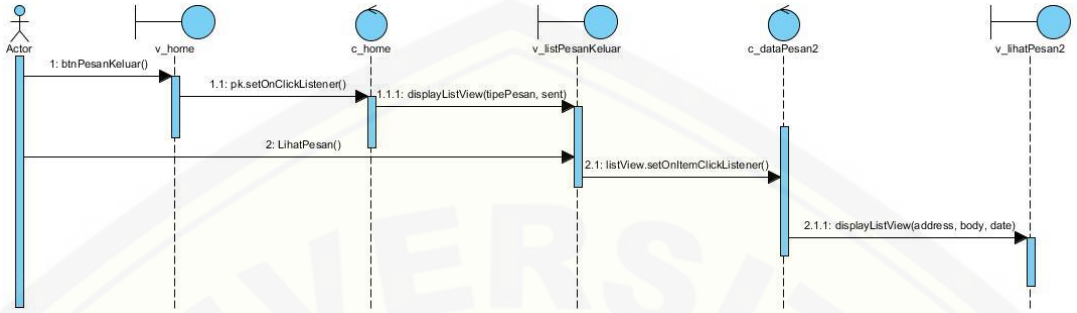
Gambar 4.8: *Sequence* Mengirim Pesan

c. *Sequence Diagram* Menerima atau Membuka Pesan Masuk



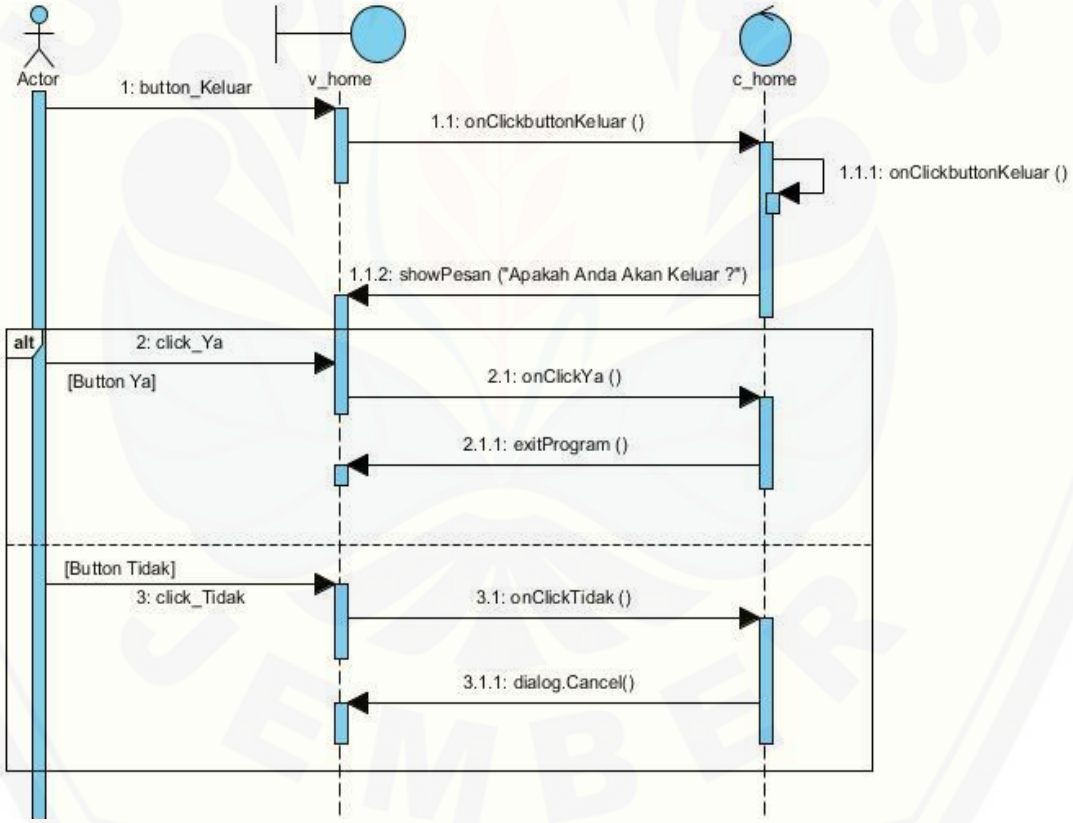
Gambar 4.9: *Sequence* Menerima atau Membuka Pesan Masuk

d. *Sequence Diagram* Membaca Pesan Keluar



Gambar 4.10: *Sequence* Membaca Pesan Keluar

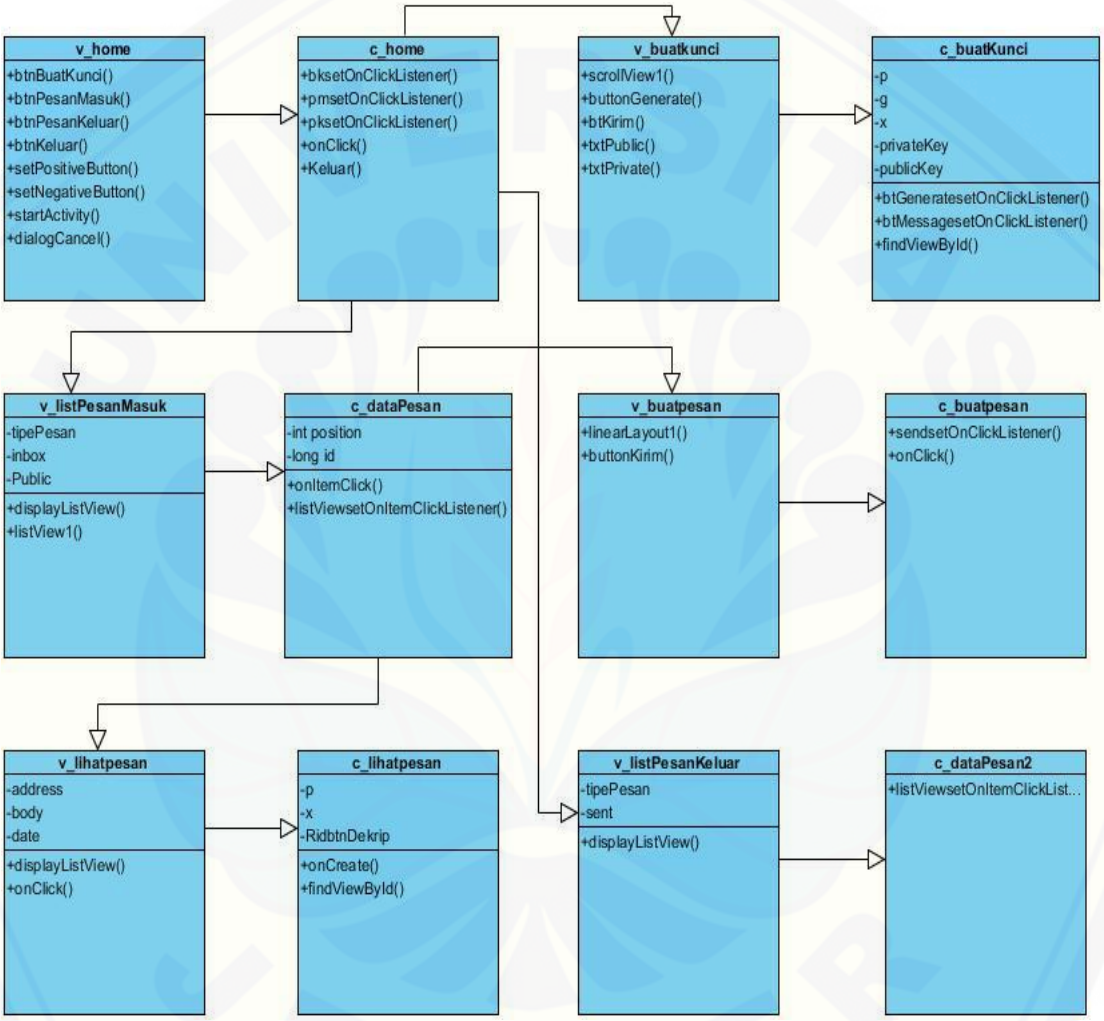
e. *Sequence Diagram* Keluar



Gambar 4.11: *Sequence* Keluar

4.6 *Class Diagram*

Setelah melalui tahap pembuatan desain dengan *sequence diagram*, tahap selanjutnya yaitu membuat desain perancangan *class diagram*. Untuk *class diagram android mobile* ini dapat dilihat pada Gambar 4.12



Gambar 4.12: *Class Diagram* Android Mobile

4.7 Implementasi Perancangan

Setelah tahap desain perancangan selesai, tahap selanjutnya dalam penelitian ini yaitu tahap pengimplementasian desain perancangan ke dalam bahasa pemrograman. Bahasa pemrograman yang dipakai adalah bahasa pemrograman Java yang dapat diakses pada perangkat mobile.

a. Alur Pemrograman Buat Kunci

```
public class BuatKunciEl {

    private BigInteger y;
    private boolean cek;
    private int bilanganPrima;

    public BigInteger getKunci(BigInteger p, BigInteger g, BigInteger x) {
        y = g.modPow(x, p);
        return y;
    }

    public void setPrima(int bilanganPrima) {
        this.bilanganPrima = bilanganPrima;
    }

    public boolean isPrima() {
        for (int i = 3; i < bilanganPrima; i += 2) {
            if (bilanganPrima % i == 0) {
                cek = false;
                break;
            } else {
                cek = true;
            }
        }

        return cek;
    }
}
```

Gambar 4.13: Pembuatan Kunci

Gambar 4.13 ini menjelaskan tentang pengkodingan saat pembuatan kunci dan penginputan nilai p, g, dan x.

b. Alur Pemrograman Proses Enkripsi

```
public String getEnkripsi(String chrASCII, String rnd, BigInteger g,
        BigInteger p, BigInteger y, String pesan) {

    for (int i = 0; i < pesan.length(); i++) {
        BigInteger m = new BigInteger(chrASCII);
        BigInteger k = new BigInteger(rnd);

        gamma = g.modPow(k, p);

        delta = y.pow(k.intValue()).multiply(m).mod(p);

    }
    return gamma.toString() + " " + delta.toString()+" ";
}
```

Gambar 4.14: Proses Enkripsi

Gambar 4.14 ini menjelaskan tentang pengkodingan pada saat dimana pesan *plaintext* akan di enkripsi menjadi pesan *chiphertext*.

c. Alur Pemrograman Proses Deskripsi

```
public char getDekripsi(String nGamma, String nDelta,
        BigInteger p, BigInteger x, String pesan) {

    for (int i = 0; i < pesan.length(); i++) {

        BigInteger a = new BigInteger(nGamma);
        BigInteger b = new BigInteger(nDelta);

        BigInteger m = b.multiply(a.pow(p.intValue() - 1 - x.intValue())).mod(p);
        int ma = m.intValue();
        chr = (char) ma;
    }
    return chr;
}
```

Gambar 4.15: Proses Deskripsi

Gambar 4.15 ini menjelaskan tentang pengkodingan pada saat dimana pesan *chiphertext* akan di deskripsi menjadi pesan *plaintext*.

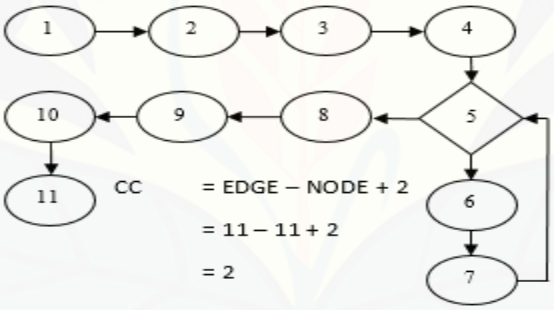
4.8 Pengujian

Pada tahapan ini pengujian dapat dilakukan terhadap sistem melalui 2 cara yaitu, *white box* dan *black box.* Untuk pengujian kali ini dilakukan dengan cara white box yang dapat dilakukan pada function class.

4.8.1 *White Box Testing*

Pengujian *white box* dilakukan dengan menggunakan metode *Cyclomatic Complexcity* (CC). Pengujian *white box* ini dapat dilihat pada gambar dibawah ini dan lebih lengkapnya pada halaman lampiran.

```
public class BilanganAcak {

    @SuppressWarnings("rawtypes")
    private final ArrayList listNumber = new ArrayList();

    @SuppressWarnings({ "rawtypes", "unchecked" })
    public ArrayList getBilanganAcak(String pesan, BigInteger p) {

        int pp = p.intValue() - 2;
        for (int i = 0; i < pesan.length(); i++) {
            int rand = (int) (pp * Math.random());

            listNumber.add(rand);
        }
        return listNumber;
    }
}
```



Gambar 4.16: Whitebox Bilangan Acak

4.8.2 Black Box Testing

Tahapan pengujian *black box* dilakukan untuk menguji apakah kebutuhan yang dibutuhkan oleh *user* atau kebutuhan yang tertera pada kebutuhan fungsional sudah sesuai atau tidak, sehingga pengujian ini dilakukan pada *form* untuk setiap *usecase* yang dilakukan *sample* pengguna aplikasi *android mobile* yang dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil Pengujian *Black Box* Perhitungan Rute

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Membuat Kunci	Memilih *button* "Buat Kunci"	Menampilkan form "Buat Kunci"	[√] Berhasil [ ] Gagal
		a) Menginputkan nilai p, g, x b) Klik *button* "generate"	Menampilkan hasil *public key* dan *private key*	[√] Berhasil [ ] Gagal
		a) Menginputkan nomer tujuan b) Klik *button* "kirim"	Mengirim pesan sesuai nomer tujuan	[√] Berhasil [ ] Gagal
2.	Mengirim Pesan	Memilih *button* "Pesan Masuk"	Menampilkan form "Pesan Masuk"	[√] Berhasil [ ] Gagal
		Memilih pesan *public key* dari penerima	a) menampilkan form pesan *public key* b) menampilkan *public key* otomatis pada text box	[√] Berhasil [ ] Gagal
		a) Menulis pesan yang akan dikirim b) Memilih *button* "kirim"	Mengekripsi pesan dan mengirim pesan pada penerima	[√] Berhasil [ ] Gagal

Berlanjut

Lanjutan

No.	Fitur	Aksi	Hasil	Kesimpulan
3.	Menerima atau Membuka Pesan Masuk	Memilih *button* "Pesan Masuk"	Menampilkan form "Pesan Masuk"	[√] Berhasil [ ] Gagal
		Memilih pesan yang akan dibuka	Menampilkan isi pesan yang dipilih oleh penerima	[√] Berhasil [ ] Gagal
		a) Mengitputkan *private key* (p dan x) b) Memilih *button decrypt*	Menampilkan *plaintext* yang telah di deskripsi	[√] Berhasil [ ] Gagal
4.	Membaca Pesan Keluar	Memilih *button* "Pesan Keluar"	Menampilkan form "Pesan Keluar"	[√] Berhasil [ ] Gagal
		Memilih pesan yang akan dibaca atau dibuka	Menampilkan isi pesan yang dipilih oleh aktor	[√] Berhasil [ ] Gagal
5.	Keluar	Memilih *button* "Keluar"	Menampilkan pop up *button* "Keluar"	[√] Berhasil [ ] Gagal
		Memilih *button* "Ya"	Menutup semua aktifitas pada aplikasi *android mobile*	[√] Berhasil [ ] Gagal
		Memilih *button* "Tidak"	Menampilkan kembali aplikasi *android mobile*	[√] Berhasil [ ] Gagal

BAB 5. HASIL DAN PEMBAHASAN

Pada bab ini akan dipaparkan hasil dan pembahasan sistem selama dilakukannya penelitian yang mencakup setiap tahap implementasi dan pengujian Aplikasi Pesan Rahasia dengan Metode Kriptografi *Elgamal* pada Perangkat *Android mobile*.

5.1 Analisis Data Penerapan Metode Elgamal

Pada sub bab ini pengujian dilakukan dengan cara membandingkan hasil proses enkripsi dan dekripsi dari program aplikasi yang telah dibuat dengan hasil penghitungan enkripsi dan dekripsi secara manual. Data yang digunakan untuk pengujian ini adalah seperti pada Tabel 5.1.

Tabel 5.1: Data Pengujian Program

Keterangan	Nilai
Pesan	hello android
Nilai (p,g,y,x)	(383, 148, 295, 338)
Nilai k	k_1 = 319, k_2= 259, k_3 = 353, k_4 = 105, k_5 = 267, k_6= 279, k_7 = 190, k_8 = 252, k_9 = 60, k_{10} = 87, k_{11} = 360, k_{12} =139, k_{13} = 48

5.1.1 Proses Pembentukan Kunci

Berikut contoh manual pembentukan kunci untuk proses enkripsi dan dekripsi. Misalkan pengirim memilih p = 383, g = 148, dan x = 338. Kemudian menghitung: $y = g^x \bmod p = 148^{338} \bmod 383 = 295$

Diperoleh kunci publik (y, g, p) = (295, 148, 383) dan kunci privatnya x = 338. Kunci publik (295, 148, 383) inilah yang diberikan penerima kepada pemberi pesan. Kunci rahasia tetap dipegang oleh penerima dan tidak boleh ada yang mengetahui selain dirinya sendiri.

5.1.2 Proses Enkripsi Secara Manual

Langkah-langkah penyelesaian proses enkripsi secara manual adalah sebagai berikut:

Diketahui: Plaintext: "hello android"

Nilai p = 383, g = 148 dan y = 295.

Nilai k_1 = 319, k_2 = 259, k_3 = 353, k_4 = 105, k_5 = 267, k_6 = 279, k_7 = 190, k_8 = 252, k_9 = 60, k_{10} = 87, k_{11} = 360, k_{12} = 139, k_{13} = 48

Jawab:

a. Ubah pesan asli (plaintext) ke dalam ASCII
h=104, e=101, l=108, l=108, o=111, spasi=32, a=97, n=110, d=100, r=114, o=111, i=105, d=100
sehingga nilai pesan ASCII adalah sebagai berikut :
m_1=104, m_2=101, m_3=108, m_4=108, m_5=111, m_6=32, m_7=97, m_8=110, m_9=100, m_{10}=114, m_{11}=111, m_{12}=105, m_{13}=100

Tabel 5.2: Konversi ASCII

No.	Karakter	Plaintext mi	ASCII
1	h	m_1	104
2	e	m_2	101
3	l	m_3	108
4	l	m_4	108
5	o	m_5	111
6	<spaci>	m_6	32
7	a	m_7	97
8	n	m_8	110
9	d	m_9	100
10	r	m_{10}	114
11	o	m_{11}	111
12	i	m_{12}	105
13	d	m_{13}	100

b. Hitung gamma (γ) dengan rumus $\gamma = g^k \bmod p$

$\gamma_1 = 148^{319} \bmod 383$ $\gamma_2 = 148^{259} \bmod 383$ $\gamma_3 = 148^{353} \bmod 383$
$\gamma_4 = 148^{105} \bmod 383$ $\gamma_5 = 148^{267} \bmod 383$ $\gamma_6 = 148^{279} \bmod 383$
$\gamma_7 = 148^{190} \bmod 383$ $\gamma_8 = 148^{252} \bmod 383$ $\gamma_9 = 148^{60} \bmod 383$
$\gamma_{10} = 148^{87} \bmod 383$ $\gamma_{11} = 148^{360} \bmod 383$ $\gamma_{12} = 148^{139} \bmod 383$
$\gamma_{13} = 148^{48} \bmod 383$

Sehingga Hasil $\gamma_1 = 197$, $\gamma_2 = 122$, $\gamma_3 = 85$, $\gamma_4 = 379$, $\gamma_5 = 340$, $\gamma_6 = 269$, $\gamma_7 = 339$, $\gamma_8 = 31$, $\gamma_9 = 168$, $\gamma_{10} = 37$, $\gamma_{11} = 38$, $\gamma_{12} = 356$, $\gamma_{13} = 144$

c. Hitung delta dengan rumus $\delta = y^k * m \bmod p$

$\delta_1 = 295^{319}. 104 \bmod 383$
$= 158$

$\delta_2 = 295^{259}. 101 \bmod 383$
$= 2$

$\delta_3 = 295^{353}. 108 \bmod 383$
$= 300$

$\delta_4 = 295^{105}. 108 \bmod 383$
$= 336$

$\delta_5 = 295^{267}. 111 \bmod 383$
$= 250$

$\delta_6 = 295^{279}. 32 \bmod 383$
$= 98$

$\delta_7 = 295^{190}. 97 \bmod 383$
$= 99$

$\delta_8 = 295^{252}. 110 \bmod 383$
$= 153$

$\delta_9 = 295^{60}. 100 \bmod 383$
$= 292$

$\delta_{10} = 295^{87}. 114 \bmod 383$
$= 113$

$\delta_{11} = 295^{360}. 111 \bmod 383$
$= 367$

$\delta_{12} = 295^{139}. 105 \bmod 383$
$= 345$

$\delta_{13} = 295^{48}. 100 \bmod 383$
$= 8$

Hasil nilai $\delta_1 = 158$, $\delta_2 = 2$, $\delta_3 = 300$, $\delta_4 = 336$, $\delta_5 = 250$, $\delta_6 = 98$, $\delta_7 = 99$, $\delta_8 = 153$, $\delta_9 = 292$, $\delta_{10} = 113$, $\delta_{11} = 367$, $\delta_{12} = 345$, $\delta_{13} = 8$

d. Susun hasil perhitungan gamma (γ) dan delta (δ)
Ciphertext: 197, 158, 122, 2, 85, 300, 379, 336, 340, 250, 269, 98, 339, 99, 31, 153, 168, 292, 37, 113, 38, 367, 356, 345, 144, 8.

Tabel 5.3: Proses Enkripsi plaintext ke *chipertext*

i	m_i	k_i	$\gamma = 148^k \bmod 383$	$\delta = 295^k . m \bmod 383$
1	104	319	197	158
2	101	259	122	2
3	108	353	85	300
4	108	105	379	336
5	111	267	340	250
6	32	279	269	98
7	97	190	339	99
8	110	152	31	153
9	100	60	168	292
10	114	87	37	113
11	111	360	38	367
12	105	139	356	345
13	100	48	144	8

5.1.3 Proses Dekripsi Secara Manual

Langkah-langkah penyelesaian proses dekripsi secara manual adalah sebagai berikut:

Diketahui:
Ciphertext: 197, 158, 122, 2, 85, 300, 379, 336, 340, 250, 269, 98, 339, 99, 31, 153, 168, 292, 37, 113, 38, 367, 356, 345, 144, 8
Nilai p = 383, x = 338.

a. Pisahkan nilai gamma dan delta pada pesan rahasia (ciphertext).
γ = Ciphertext urutan ganjil.
δ = Ciphertext urutan genap
Nilai gamma $\gamma_1 = 197$, $\gamma_2 = 122$, $\gamma_3 = 85$, $\gamma_4 = 379$, $\gamma_5 = 340$, $\gamma_6 = 269$, $\gamma_7 = 339$, $\gamma_8 = 31$, $\gamma_9 = 168$, $\gamma_{10} = 37$, $\gamma_{11} = 38$, $\gamma_{12} = 356$, $\gamma_{13} = 144$

Nilai delta $\delta_1 = 158$, $\delta_2 = 2$, $\delta_3 = 300$, $\delta_4 = 336$, $\delta_5 = 250$, $\delta_6 = 98$, $\delta_7 = 99$, $\delta_8 = 153$, $\delta_9 = 292$, $\delta_{10} = 113$, $\delta_{11} = 367$, $\delta_{12} = 345$, $\delta_{13} = 8$

b. Pisahkan nilai gamma dan delta pada pesan rahasia (ciphertext).
Hitung m (pesan asli) dengan rumus:

$m = \delta . \gamma^{(p-1-x)} \bmod p$

$m_1 = 158.197^{(383-1-338)} \bmod 383$
$= 104$

$m_2 = 2.122^{(383-1-338)} \bmod 383$
$= 101$

$m_3 = 300.85^{(383-1-338)} \bmod 383$
$= 108$

$m_4 = 336.379^{(383-1-338)} \bmod 383$
$= 108$

Sehingga hasilnya sebagai berikut:
m_1=104, m_2=101, m_3=108, m_4=108, m_5=111, m_6=32, m_7=97, m_8=110, m_9=100, m_{10}=114, m_{11}=111, m_{12}=105, m_{13}=100

c. Hasil dari penyusunan inilah yang merupakan pesan asli (plaintext) yang dihasilkan pada proses dekripsi plaintext: “hello android”.

Hasil proses perhitungan enkripsi dekripsi dengan program aplikasi dan secara manual adalah sama. Selain itu plaintext setelah dekripsi sama dengan nilai plaintext sebelum di enkripsi.

Tabel 5.4: Proses Dekripsi *chipertext* ke plaintext

i	Δ	γ	$m_i = \delta_i . \gamma_i^{(383-1-338)} \bmod 383$	Karakter m_i
1	158	197	104	h
2	2	122	101	e
3	300	85	108	l
4	336	379	108	l
5	250	340	111	o
6	98	269	32	<spasi>
7	99	339	97	a
8	153	31	110	n
9	292	168	100	d
10	113	37	114	r
11	367	38	111	i
12	345	356	105	o

5.1.4 Proses *Screenshoot* Aplikasi

Gambar 5.1: Alur di *Android Mobile*

5.2 Hasil Implementasi *Android mobile*

Hasil implementasi aplikasi *android mobile* yang dibangun pada penelitian ini terdiri atas beberapa fitur yang dapat diakses oleh aktor pengirim maupun penerima. Dengan adanya sistem ini dapat membantu mengamankan pesan rahasia agar tidak mudah diakses dan digunakan pihak lain.

5.2.1 Halaman Awal atau Dashboard

Gambar 5.2: Halaman Dashboard

5.2.2 Halaman Buat Kunci

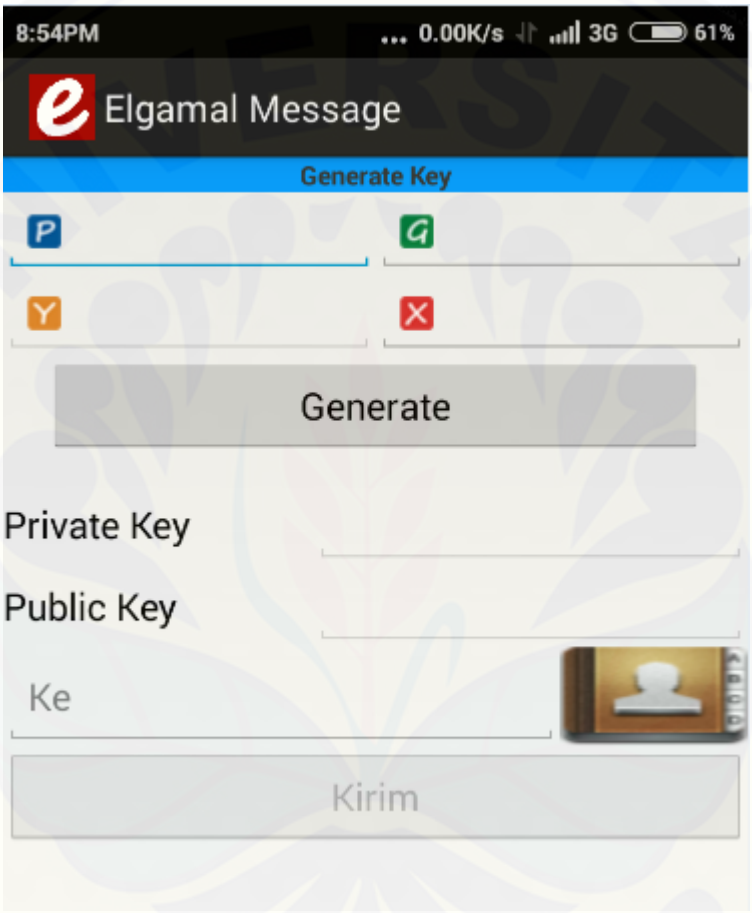
Halaman buat kunci ini dikhususkan untuk proses penginputan nilai (p,g,dan x) untuk mencari y. Lalu proses pengiriman *public key* dengan menginputkan nomer tujuan. Tampilan buat kunci ini dapat dilihat pada gambar 5.3:

5.2.3 Halaman Pesan Masuk

Halaman pesan masuk merupakan fitur untuk menampilkan isi pesan yang diterima yang berisi pesan *public key*, pesan yang akan di deskripsi, dan pesan-pesan lainnya yang terdapat di *android mobile* pemiliknya. Tampilan pesan masuk ini dapat dilihat pada gambar 5.4:

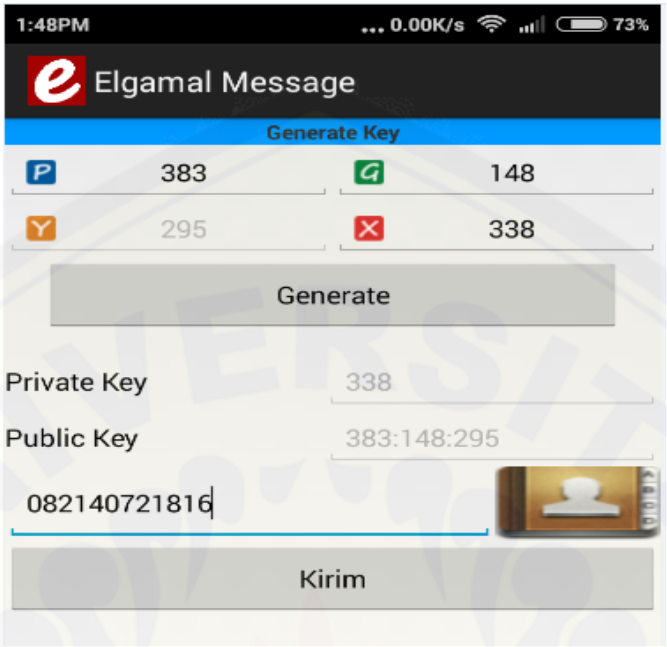
5.2.4 Halaman Pesan *Public key*

Halaman ini merupakan fitur untuk mengirim pesan yang sifatnya rahasia yang secara otomatis *public key* terisi pada label. Lalu pengirim mengisi isi pesan pada text field yang telah tersedia. Tampilan pesan *public key* ini dapat dilihat pada gambar 5.5:



Gambar 3 Gambar 5.3: Halaman Buat Kunci

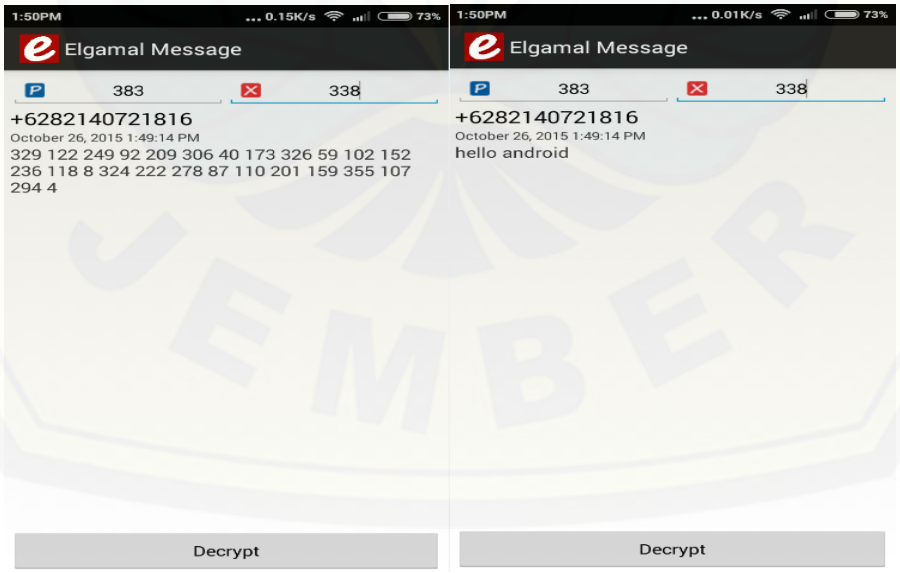
Gambar 5.4: Halaman Pesan Masuk



Gambar 5.5: Halaman Pesan *Public key*

5.2.5 Halaman Lihat Pesan

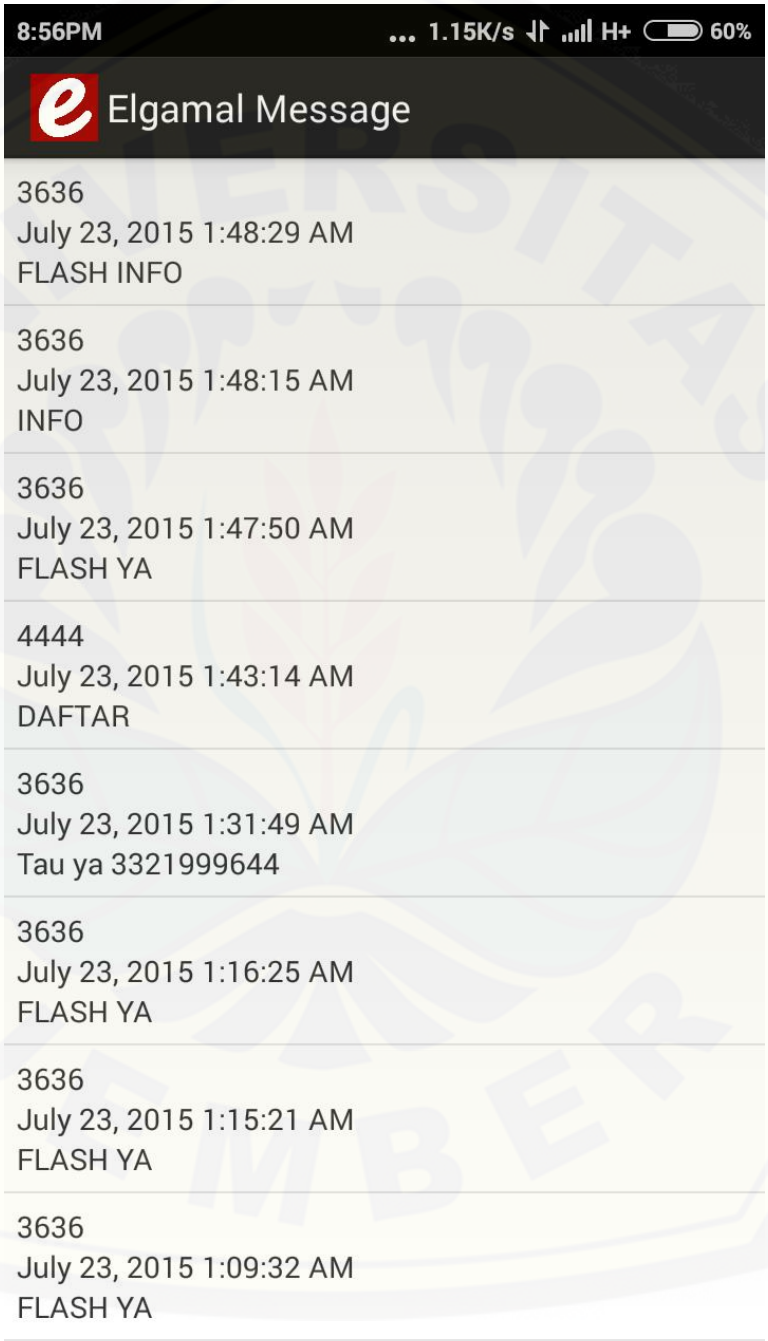
Halaman ini merupakan fitur untuk melihat pesan *chipertext* yang akan di deskripsi menjadi pesan plaintext dan hasil pesan deskripsi. Tampilan lihat pesan ini dapat dilihat pada gambar 5.6:



Gambar 5.6: Halaman Lihat Pesan

5.2.6 Halaman Pesan Keluar

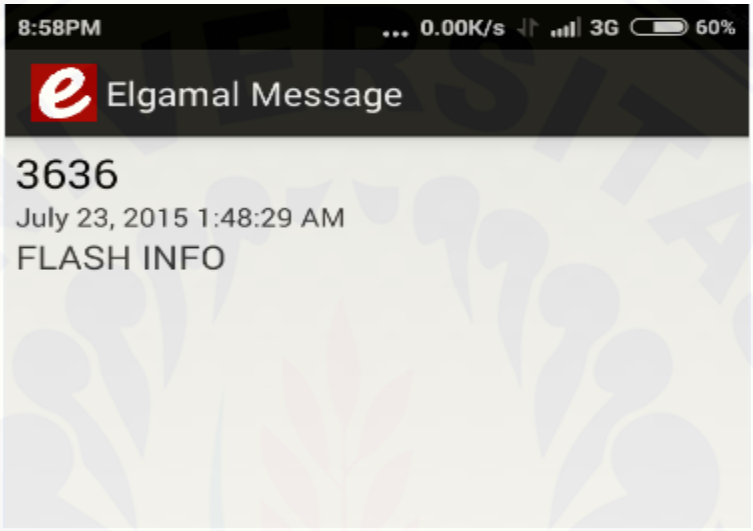
Halaman ini merupakan fitur untuk menampilkan pesan yang telah dikirim. Tampilan pesan keluar ini dapat dilihat pada gambar 5.7:



Gambar 5.7: Halaman Pesan Keluar

5.2.7 Halaman Lihat Pesan 2

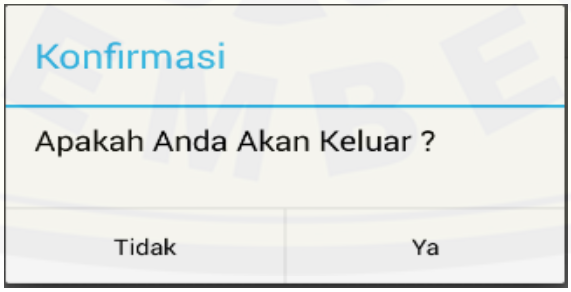
Halaman ini merupakan fitur untuk melihat isi pesan yang terdapat pada pesan terkirim tetapi tidak terdapat fitur tambahan di dalamnya seperti fitur deskripsi yang terdapat pada halaman lihat pesan. Tampilan lihat pesan 2 ini dapat dilihat pada gambar 5.8:



Gambar 5.8: Halman Lihat Pesan 2

5.2.8 Halaman atau Pop Up Keluar

Halaman atau pop up keluar ini merupakan fitur untuk keluar dari aplikasi *android mobile*. Dengan memilih ya untuk keluar dari aplikasi *android mobile* dan memilih tidak untuk tetap pada aplikasi tersebut. Tampilan pop up ini dapat dilihat pada gambar 5.9:



Gambar 5.9: Pop Up Keluar

BAB 6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan oleh peneliti, dapat diambil kesimpulan sebagai berikut:

1. Algoritma ini melalui proses enkripsi dengan menggunakan *public key* untuk membuat pesan dan merubah pesan ke dalam nilai ASCII serta mengirim pesan kepada penerima. Sedangkan untuk membaca pesan melalui proses deskripsi dengan menggunakan *private key*.
2. Proses perhitungan pada algoritma Elgamal ini menggunakan perumusan gamma dan delta. Pada saat proses enkripsi hasil dari gamma dan delta diurutkan sesuai dengan jumlah nilai m sehingga terbentuk nilai *chipertext*. Dan pada saat proses deskripsi dipisahkan kembali urutan *chipertext* menjadi gamma dan delta untuk menghitung kembali nilai m dengan perumusan gamma dan delta sesuai jumlah gamma dan delta yang tertera, sehingga terbentuk kembali urutan nilai m untuk merubahnya kembali menjadi pesan *plaintext* dengan melihat nilai ASCII.
3. Aplikasi ini bisa digunakan untuk mengamankan pesan yang bersifat rahasia, dimana penggunanya tidak ingin pesan tersebut dibaca oleh pihak tertentu.

6.2 Saran

Pengembangan lebih lanjut untuk penelitian ini dapat dilakukan dengan membangun aplikasi *android mobile* pesan rahasia dengan menggunakan paket data dan disarankan menggunakan metode kriptografi lainnya untuk menciptakan perbandingan antar metode yang satu dengan yang lain.

DAFTAR PUSTAKA

A., Y. T., Idris, W. S., & Kholid, F. (2006). Enkripsi Email Dengan Menggunakan Metode Elgamal Pada Perangkat Mobile. *Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember*.

Caroline, M. L. (2011). Perbandingan Algoritma Kriptografi Kunci Publik RSA, Rabin, dan Elgamal. *Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung*.

H., N. S. (2011). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android.* Bandung: Informatika Bandung.

Indra, N. (2011). Jurnal Analisis dan Perbandingan Kecepatan Algoritma RSA dan Algoritma ElGamal. *Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung*.

Kadir, A. (2003). *Pengenalan Sistem Informasi.* Yogyakarta: ANDI.

Nielsen. (2014, September 6). *On Device Meter*. Diambil kembali dari http://www.dreamersradio.com/article/31331

P., H. (2013). *Android Programming with Eclipse.* Yogyakarta: ANDI.

Riyanto, M. (2007). Pengamanan pesan rahasia menggunakan algoritma kriptografi ElGamal atas grup pergandaan Zp. *Jurusan Matematika FMIPA Universitas Gadjah Mada Yogyakarta*.

W., D., & M., H. (1976). *New Directions in Cryptography.* Nopember: IEEE Transactions on Information Theory.

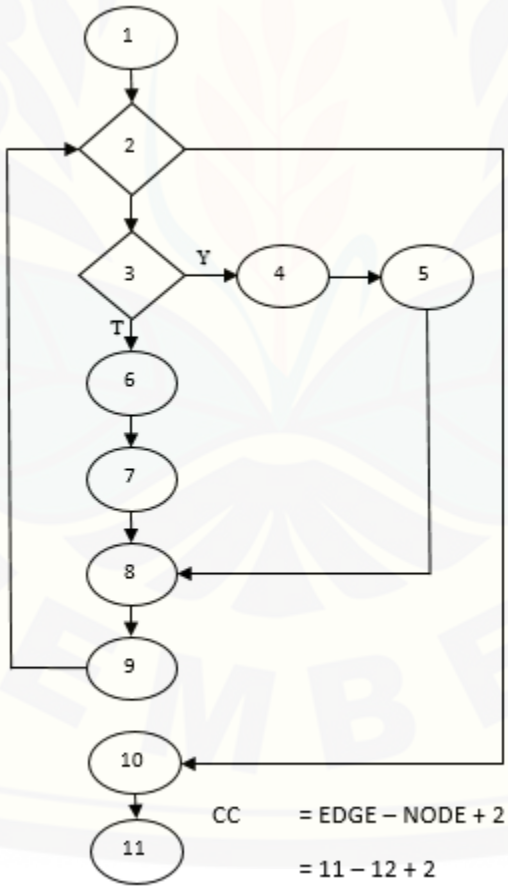
Widiantoro, S. (2014, September 6). *Implementasi logika deduktif pada aplikasi sistem pakar*. Diambil kembali dari http://issuu.com/esq-bs/docs/jbsm-201402s

LAMPIRAN

A. Pengujian *White Box*

1. *Class* boolean isPrima

```
public boolean isPrima() {
    for (int i = 3; i < bilanganPrima; i += 2) {
        if (bilanganPrima % i == 0) {
            cek = false;
            break;
        } else {
            cek = true;
        }
    }

    return cek;
}
```



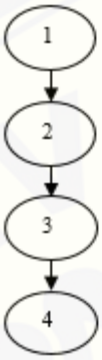
CC = EDGE – NODE + 2

= 11 – 12 + 2

= 1

2. Big integer

```
public BigInteger getKunci(BigInteger p, BigInteger g, BigInteger x) {
    y = g.modPow(x, p);
    return y;
}
```



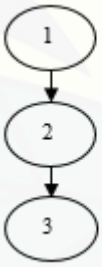
CC = EDGE – NODE + 2

= 4 – 4 + 2

= 2

3. Void setPrima

```
public void setPrima(int bilanganPrima) {
    this.bilanganPrima = bilanganPrima;
}
```



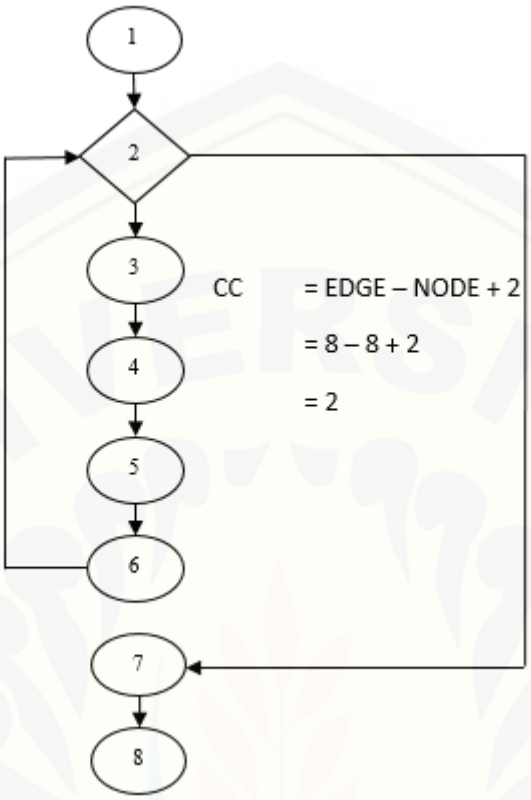
CC = EDGE – NODE + 2

= 2 – 3 + 2

= 2

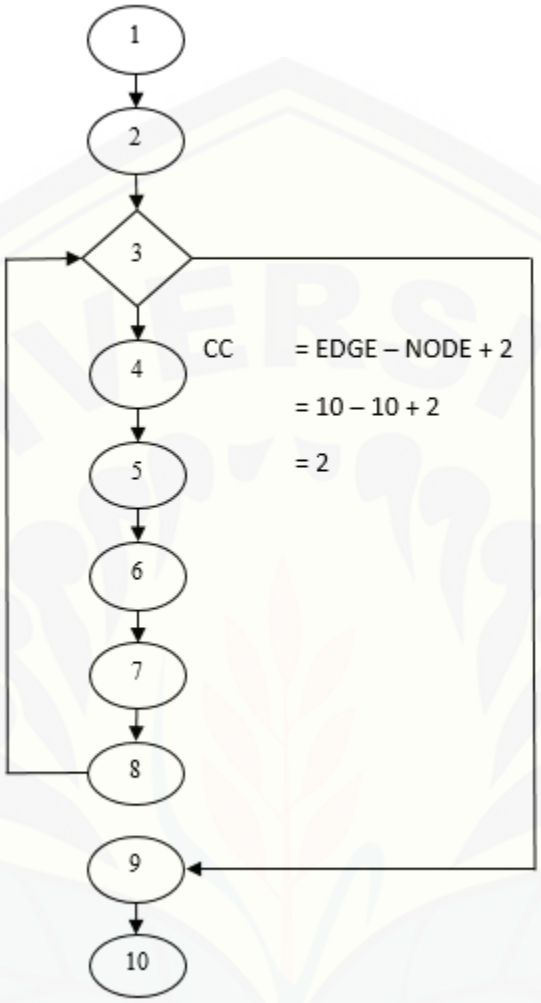
4. Get char ASCII

```
public ArrayList getCharASCII(String pesan) {

    for (int i = 0; i < pesan.length(); i++) {
        char chr = pesan.charAt(i);
        int in = chr;

        listChar.add(in);
    }

    return listChar;
}
```



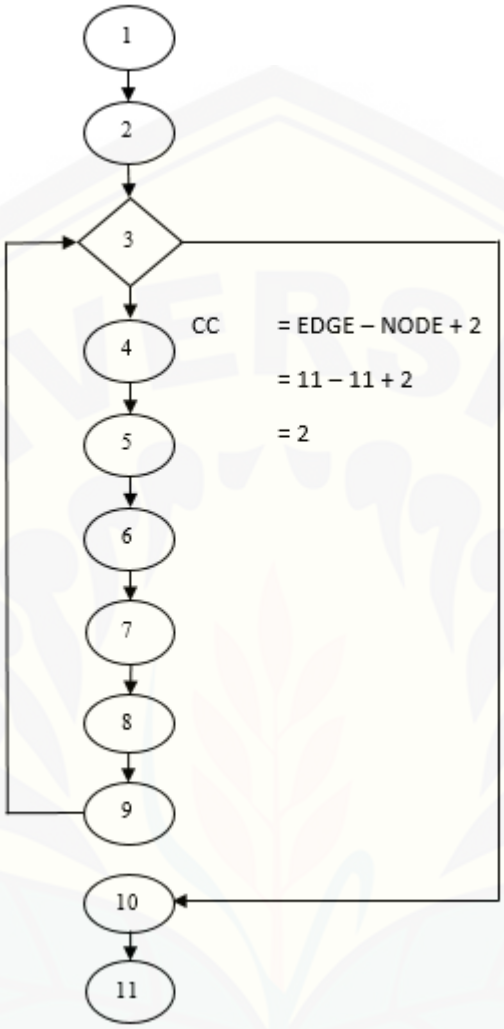
5. Enkripsi

```
public String getEnkripsi(String chrASCII, String rnd, BigInteger g,
        BigInteger p, BigInteger y, String pesan) {

    for (int i = 0; i < pesan.length(); i++) {
        BigInteger m = new BigInteger(chrASCII);
        BigInteger k = new BigInteger(rnd);

        gamma = g.modPow(k, p);

        delta = y.pow(k.intValue()).multiply(m).mod(p);

    }
    return gamma.toString() + " " + delta.toString()+" ";
}
```



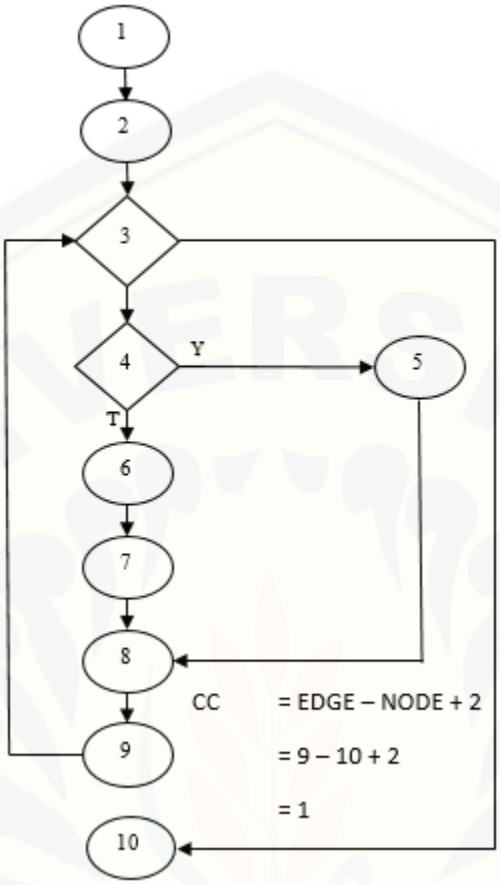
6. Deskripsi

```
public char getDekripsi(String nGamma, String nDelta,
        BigInteger p, BigInteger x, String pesan) {

    for (int i = 0; i < pesan.length(); i++) {

        BigInteger a = new BigInteger(nGamma);
        BigInteger b = new BigInteger(nDelta);

        BigInteger m = b.multiply(a.pow(p.intValue() - 1 - x.intValue())).mod(p);
        int ma = m.intValue();
        chr = (char) ma;
    }
    return chr;
}
```



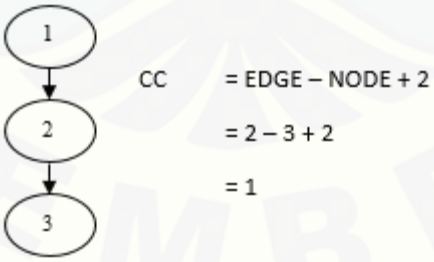
7. Pecah chiper text

```
public void setChiper(String chiper) {
    pecah = chiper.split(" ");

    for (int i = 0; i < pecah.length; i++) {

        if (i % 2 == 0) {
            pGamma.add(pecah[i]);
        } else {
            pDelta.add(pecah[i]);
        }
    }
}
```



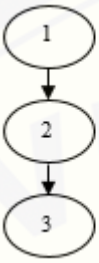
8. Get gamma

```
public ArrayList getGamma() {
    return pGamma;
}
```



9. Get delta

```
public ArrayList getDelta() {
    return pDelta;
}
```



CC = EDGE – NODE + 2

= 2 – 3 + 2

= 1

B. Bilangan ASCII

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	!!	DC3	051	3	083	S	115	s
020	¶	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	|
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂