# APLIKASI *MOBILE MESSENGER* MENGGUNAKAN METODE ALGORITMA ECC (*ELLIPTIC CURVE CRYPTOGRAPHY*)

**SKRIPSI**


Oleh :

**EKO HARTANTO**

**NIM. 06550043**




**JURUSAN INFORMATIKA**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS ISLAM NEGERI**
**MAULANA MALIK IBRAHIM**
**MALANG**


**2013**

**APLIKASI *MOBILE MESSENGER* MENGGUNAKAN METODE ALGORITMA ECC (*ELLIPTIC CURVE CRYPTOGRAPHY*)**

**SKRIPSI**


Oleh :

**EKO HARTANTO**

**NIM. 06550043**




**JURUSAN INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI**

**MAULANA MALIK IBRAHIM**

**MALANG**


**2013**

**APLIKASI *MOBILE MESSENGER* MENGGUNAKAN METODE ALGORITMA ECC (*ELLIPTIC CURVE CRYPTOGRAPHY*)**

**SKRIPSI**

**Diajukan Kepada:**
**Fakultas Sains dan Teknologi**
**Universitas Islam Negeri Maulana Malik Ibarahim Malang**
**Untuk Memenuhi Salah Persyaratan Dalam**
**Memperoleh Gelar Sarjana Komputer (S.Kom)**

Oleh:
**EKO HARTANTO**
**NIM. 06550043**

**JURUSAN INFORMATIKA**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS ISLAM NEGERI**
**MAULANA MALIK IBRAHIM**
**MALANG**
**2013**

# HALAMAN PERSETUJUAN

**APLIKASI *MOBILE MESSENGER* MENGGUNAKAN METODE ALGORITMA ECC (*Elleptic Curve Cryptography*)**

**SKRIPSI**

Oleh

**EKO HARTANTO**
**NIM.06550043**

Telah Disetujui
Malang, 7 Juli 2013

| Dosen Pembimbing 1 | Dosen Pembimbing II |
|---|---|
| **RIRIEN KUSUMAWATI, M.Kom** | **YUNIFA MIFTACHUL ARIF, M.T** |
| **NIP. 197203092005012002** | **NIP. 198306162011011004** |

Mengetahui
Ketua Jurusan Teknik Informatika

**RIRIEN KUSUMAWATI, M.Kom**
**NIP. 197203092005012002**

**HALAMAN PENGESAHAN**

**APLIKASI *MOBILE MESSENGER* MENGGUNAKAN METODE ALGORITMA ECC (*ELLIPTIC CURVE CRYPTOGRAPHY*)**

**SKRIPSI**

**Oleh:**
**Eko Hartanto**
**NIM. 06550043**

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar sarjana Komputer (S.Kom)

Malang 10 Juli 2013

| Susunan Dewan Penguji | | | Tanda Tangan |
|---|---|---|---|
| 1. Penguji Utama | : | Hani Nurhayati, M.T<br>NIP. 197806252008012006 | ( ) |
| 2. Ketua Penguji | : | Fresy Nugroho, M.T<br>NIP. 1970722 2011011001 | ( ) |
| 3. Sekretaris Penguji | : | Ririen Kusumawati, M.Kom<br>NIP. 197203092005012002 | ( ) |
| 4. Anggota Penguji | : | Yunifa Miftachul Arif, M.T<br>NIP. 198306162011011004 | ( ) |

Mengetahui:
Ketua Jurusan Teknik Informatika

**RIRIEN KUSUMAWATI, M.Kom**
**NIP. 197203092005012002**

# *PERSEMBAHAN*

*Kenikmatan akan terasa dengan adanya berbagai macam ujian dan cobaan. Manjalani PKLI, pengajuan judul, seminar proposal, penelitian, hafalan ayat-ayat suci, ujian kompeherensif dan mengerjakan skripsi, telah saya lalui dengan berbagai macam kisah.*

*Sebuah karya hasil jerih payahku telah berhasil kususun kupersembahkan kepada:*

*Beliau mutiara hidupku Ayah (Sunardi) dan mama (Sulastri) dengan cinta, kasih sayang dan doa beliau, saya selalu optimis untuk menuju gerbang kesuksesan yang penuh gemilang dalam hidup saya.*

*Para dosen Universitas Islam Negeri Maulana Malik Ibrahim Malang yang sangat saya hormati di jurusan teknik informatika terutama Ibu Ririen Kusumawati M.Kom dan bapak Yunifa Miftachul Arif M.T selaku pembimbing skripsi, berkat didikan, motivasi, kritik dan saran beliau, saya berhasil menyelesaikan studi di Perguruan Tinggi dengan gelar strata satu, yaitu dengan berhasilnya karya ini.*

*Teruntuk Rizki Wannur Asmara S.H.I yang selalu membangkitkan semangat, harapan dan cintanya. Semoga Allah memberikan jalan terbaik buat kita.Amin.*

# *MOTTO*

وَأَنِ ٱحۡكُم بَيۡنَهُم بِمَآ أَنزَلَ ٱللَّهُ وَلَا تَتَّبِعۡ أَهۡوَآءَهُمۡ وَٱحۡذَرۡهُمۡ أَن يَفۡتِنُوكَ عَنۢ بَعۡضِ مَآ أَنزَلَ ٱللَّهُ إِلَيۡكَۖ فَإِن تَوَلَّوۡاْ فَٱعۡلَمۡ أَنَّمَا يُرِيدُ ٱللَّهُ أَن يُصِيبَهُم بِبَعۡضِ ذُنُوبِهِمۡۗ وَإِنَّ كَثِيرٗا مِّنَ ٱلنَّاسِ لَفَٰسِقُونَ ٤٩

*"Dan hendaklah kamu memutuskan perkara di antara mereka menurut apa yang diturunkan Allah, dan janganlah kamu mengikuti hawa nafsu mereka. dan berhati-hatilah kamu terhadap mereka, supaya mereka tidak memalingkan kamu dari sebahagian apa yang Telah diturunkan Allah kepadamu. jika mereka berpaling (dari hukum yang Telah diturunkan Allah), Maka Ketahuilah bahwa Sesungguhnya Allah menghendaki akan menimpakan mushibah kepada mereka disebabkan sebahagian dosa-dosa mereka. dan Sesungguhnya kebanyakan manusia adalah orang-orang yang fasik."(Al- Maidah:49)"*

# SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Eko Hartanto

NIM : 06550043

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : Aplikasi *Mobile Messenger* Menggunakan Metode Algoritma ECC ( *Elliptic Curve Cryptography* )

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur penjiplakan maka saya bersedia untuk mempertanggungjawabkan, serta diproses sesuai peratiran yang berlaku.

Malang, 22 Juli 2013

Yang Membuat Pernyataan.

Eko Hartanto

NIM : 06550043

# KATA PENGANTAR

Segala puji bagi Allah SWT Yang Maha Memaafkan segla khilaf, Yang Maha Pengasih terhadap hamba-hambaNya, Yang Maha Pemurah atas doa yang dilantunkan hambaNya. Sebuah karya hasil penelitian dengan judul Aplikasi *Mobile Messenger* Menggunakan Metode ECC (*Elliptic Curve Criptography*)

Shalawat serta salam semoga senantiasa terlimpahkan aatas nabi Muhammad SAW yang telah mengantarkan umatnya menuju jalan yang lurus dengan agama Islam yang dibawanya. Semoga shalawat dan salam juga terlimpahkan aatas keluarga, sahabat dan umat beliau yang mengikuti ajarannya.

Setelah menekuni studi selama ini maka sampailah pada ujung masa studi, yaitu penelitian skripsi yang disusun oleh peneliti. Penelitian skripsi ini tidak lepas dari bantuan berbagai pihak yang dengan ikhlas menyumbangkan ide, saran, motivasi, waktu, bahkan materi demi keberhasilan peneliti dalam menyusun karya ini. Ucapan terima kasih yang sebanyak-banyaknya peneliti haturkan kepada;

1. Prof. Dr. H. Mudjia Rahardjo M.Si selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang yang berusaha keras demi membentuk mahasiswa-mahasiswanya menjadi orang yang berbudi pekerti luhur dan bermanfaat bagi bangsa dan negaranya.
2. Dr. Drh. Bayyinatul Muchtaromah, M.Si. selaku Dekan Fakultas SAINTEK Universitas Islam Negeri Maulana Malik Ibrahim Malang yang senantiasa berusaha membentuk anak-anak didiknya menjadi mahasiswa yang menjunjung tinggi hukum dan mematuhi syariah Islam.

3. Ririen Kusumawati, M.Kom selaku Ketua Jurusan Teknik Informatika, Fakultas SAINTEK Universitas Islam Negeri Maulana Malik Ibrahim Malang yang dengan kesabarannya membantu mahasiswa-mahasiswanya menyelesaikan segala urusan studinya.
4. Mokhamad Amin Hariyadi, M.T selaku dosen wali saya yang berjuang keras mendidik mahasiswanya hingga menyelesaikan studi di Perguruan Tinggi ini.
5. Ririen Kusumawati M.Kom dan Yunifa Miftachul Arif M.T selaku dosen pembimbing skripsi saya yang dengan ketelatenan dan kesabarannya meluangkan waktu untuk mendampingi peneliti dalam menyusun skripsi ini.
6. Ayah dan ibu yang selalu mengiringi saya dengan doa dan memberi dukungan moral, spiritual serta memberi kepercayaan terhadap putrinya merupakan motivasi tersendiri bagi peneliti sehingga ingin segera mempersembahkan karya ini kepada beliau berdua.
7. Seluruh dosen Universitas Islam Negeri Maulanan Malik Ibrahim Malang, khususnya segenap dosen jurusan teknik informatika yang berjuang keras mendidik mahasiswa-mahasiswanya hingga menyelesaikan studi di Perguruan Tinggi.
8. Admin Jurusan Teknik Informatika Ocis ana dan Citra Fidya Atmalia yang selalu sabar mengurusi keperluan administrasi dan sabar menghadapi cobaan dari para mahasiswanya sampai lulus menjadi sarjana.
9. Segenap Karyawan Fakultas Saintek Universitas Islam Negeri Maulana Malik Ibrahim Malang yang membantu dalam administrasi

10. Rizki Wannur Asmara yang sangat membantu peneliti dalam melakukan penelitian. Sahabat-sahabat RC 62 dengan kalian saya punya banyak cerita dan teman-teman kos Ahong juga Kos 49 B yang selalu memberi support.

Peneliti tidak dapat membalas kebaikan dan jasa yang telah diberikan dengan sesuatu yang mewah dan berharga kecuali dengan doa semoga Allah SWT mencatat amal mereka dan menjadi tabungan yang bisa dipanen di akhirat kelak.

Peneliti menyadari bahwa skripsi ini sangat jauh dari kesempurnaan. Oleh karena itu, peneliti menerima kritik dan saran dari para pembaca demi mendapatkan hasil yang jauh lebih baik.

Malang, 22 Juli 2013

Eko Hartanto

# DAFTAR ISI

# DAFTAR TABEL

# DAFTAR GAMBAR

# ABSTRAK

Hartanto, Eko. 2013. **Aplikasi Mobile Messenger Menggunakan Metode Algoritma ECC (Elliptic Curve Cryptography).** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (1) Ririen Kusumawati, M.Kom.(II) Yunifa Miftachul Arif, M.T

**Kata kunci:** *Mobile Messenger, Security,* Algoritma ECC *(elliptical Curve Cryptography).*

Seiring dengan perkembangan telepon seluler yang sangat pesat dan semakin banyak digunakannya layanan *mobile messenger*, maka aspek keamanan (*security*) menjadi sangat penting untuk dipertimbangkan. apalagi jika data yang dikirimkan melalui layanan tersebut merupakan data yang sensitif dan rahasia, tanpa mengesampingkan pengguna terbesar *mobile messenger* yaitu kalangan pribadi atau individual, keamanan merupakan masalah terbesar bagi pengguna *mobile messenger* pada perusahaan atau *enterprise.* Maka dibutuhkan aplikasi *mobile messenger* yang sistem enkripsinya sukar untuk dipecahkan oleh pihak yang tidak berhak menerima pesan dan juga memiliki ukuran kunci yang pendek sehingga sesuai diaplikasikan di telepon seluler, aplikasi *mobile messenger* menggunakan algoritma ECC sebagai metode enkripsi pesannya memenuhi persyaratan tersebut. Algoritma *ECC (Elliptic Curve Cryptography)* termasuk kedalam sistem kriptografi kunci publik yang mendasarkan keamanannya pada permasalahan matematis kurva, Pada penelitian ini Algoritma ECC dalam enkripsi dan dekripsi pada pengiriman dan penerimaan pesan dapat meningkatkan keamanan karena pesan dikirimkan berupa *ciphertext* dan hanya bisa dirubah ke *plaintext* dengan perhitungan matematis kurva.

## مستخلص البحث

هرتنطا, إيكو.٢٠١٣. **تطبيق رسول المحمول طريقة خوارزمية طريق الاهليلجيه تشفير المنحنى**. أطروحة. قسم المعلوماتية، كلية العلوم وهندسة المعلومات من الجامعة الإسلامية مولانا مالك إبراهيم مالانج الدولة.

المشرف: (١) ردرن كوسوماواتى. م.كوم

(٢) نونفا منفتحول اريف. م.ت

**كلمات البحث**: المحمول رسول، الأمن، خوارزميات بيضاوي الشكل تشفير المنحنى.

جنبا إلى جنب مع تطور الهاتف المحمول هو سريع جدا والمزيد والمزيد من استخدام خدمة المحمول، والجوانب الأمنية الأمن يصبح من المهم جدا للنظر. خاصة إذا كانت البيانات المرسلة من خلال الخدمة حساس والبيانات السرية، دون المساس أكبر مستخدمي رسول المحمول هو الشخصية أو الفردية، والأمن هو أكبر مشكلة لمستخدمي الهواتف المتحركة في الشركة أو المؤسسة رسول. يستغرق تطبيق موبايل رسول أن أنظمة التشفير يصعب حلها من قبل أولئك الذين ليسوا مؤهلين لتلقي الرسائل وأيضا لديه أحجام المفاتيح أقصر التي تناسب تطبيقها على الهواتف النقالة، وتطبيق رسول النقالة باستخدام الخوارزمية وطريقة تشفير الرسائل يلبي هذه المتطلبات. خوارزمية الاهليلجيه تشفير المنحنى المضمنة في نظام تشفير المفتاح العام أن تؤسس أمنها على المشاكل الرياضية منحنى، في هذه خوارزمية البحث في التشفير وفك التشفير في إرسال واستقبال الرسائل يمكن تحسين الأمن لأنه يتم إرسال الرسالة في شكل النص المشفر والغير مشفرة ويمكن أيضا أن تتغير إلى الحسابات الرياضية منحنى.

# ABSTRAK

Hartanto, Eko. 2013. **Mobile Messenger Application Algorithm Method Using ECC (Elliptic Curve Cryptography).** Theses. Department of Informatics Engineering. Faculty of Science and Technology The State of Islamic University Maulana Malik Ibrahim Malang.
Promotor: (1) Ririen Kusumawati, M.Kom.
(II) Yunifa Miftachul Arif, M.T

Along with the development of mobile phone is very fast and more use of mobile messenger service, the security aspects becomes very important to consider. especially if the data transmitted through the service is sensitive and confidential data, without prejudice to the biggest users of mobile messenger is the personal or individual, security is the biggest issue for mobile users on the company or enterprise messenger. It takes a mobile messenger application that encryption systems are difficult to be solved by those who are not eligible to receive messages and also has a shorter key sizes that fit applied on mobile phones, mobile messenger application using ECC algorithm as message encryption method meets these requirements. Algorithm ECC (Elliptic Curve Cryptography) included in the public key cryptography system that bases its security on the curve mathematical problems, in this research ECC algorithm in encryption and decryption in sending and receiving messages can improve security because the message is sent in the form of ciphertext and plaintext can also be changed to the curve mathematical calculations.

**Kata kunci:** Mobile Messenger, Security, Algorithm ECC (Elliptical Curve Cryptography).

# BAB I

# PENDAHULUAN

## 1.2 LATAR BELAKANG

Telepon seluler adalah alat komunikasi yang menyediakan media komunikasi beragam dan salah satunya adalah media pengiriman pesan, yaitu layanan komunikasi berupa pengiriman pesan yang berformat text. pesan teks yang dikirimkan melalui telepon seluler ada beberapa cara, melalui sms ataupun aplikasi pengiriman pesan yang lain yang lazim disebut *mobile messenger*.

Seiring dengan perkembangan telepon seluler yang sangat pesat dan semakin banyak digunakannya layanan *mobile messenger*, maka aspek keamanan (*security*) menjadi sangat penting untuk dipertimbangkan. apalagi jika data yang dikirimkan via layanan tersebut merupakan data yang sensitif dan rahasia, tanpa mengesampingkan pengguna terbesar *mobile messenger* yaitu kalangan pribadi atau individual, keamanan merupakan masalah terbesar bagi pengguna *mobile messenger* pada perusahaan atau *enterprise*.

Misalnya pada sebuah perusahaan, beberapa informasi yang sifatnya rahasia dan hanya boleh diketahui oleh orang-orang tertentu dalam sebuah perusahaan tersebut seperti informasi tentang cara membuat produk yang sedang dikembangkan, Seandainya data yang berisi informasi tersebut jatuh kepada pihak lawan bisnis, maka perusahaan akan mengalami kerugian.

Terkait tentang konsep kerahasiaan, kerahasiaan menurut konsep agama Islam bahwa kejadian-kejadian apapun yang ada di alam semesta ini merupakan suatu rahasia Allah yang telah dituliskanNya di alam *lauhul mahfuzh*. Menyangkut tentang hal tersebut, Allah berfirman dalam surah Al- Hadiid pada ayat 22:

مَّا أَصَابَ مِن مُّصِيبَةٍ فِي الْأَرْضِ وَلَا فِي أَنفُسِكُمْ إِلَّا فِي كِتَابٍ مِّن قَبْلِ أَن نَّبْرَأَهَا ۚ إِنَّ ذَٰلِكَ عَلَى اللَّهِ يَسِيرٌ ﴿٢٢﴾

Tiada suatu bencanapun yang menimpa di bumi dan (tidak pula) pada dirimu sendiri melainkan telah tertulis dalam kitab (Lauhul Mahfuzh) sebelum Kami menciptakannya. Sesungguhnya yang demikian itu adalah mudah bagi Allah.

Oleh karenanya kerahasiaan sebuah pesan patutlah kita jaga, tetapi sayangnya, kebanyakan sistem *mobile messenger* saat ini didesain bukan berdasarkan aspek keamanan sebagai pertimbangan utama, melainkan aspek skalabilitas untuk menunjang jumlah pengguna yang begitu besar.. (Symantec, 2006)

Aplikasi-aplikasi *mobile messenger* yang lebih dulu ada yaitu seperti whatsapp, line, ebuddy dan yahoo messenger, tetapi dari beberapa aplikasi diatas tidak banyak literatur yang menerangkan bagaimana aplikasi mereka mengenkripsi dan membuat pesan tersebut aman sampai kepada penerima pesan.

Sebagai contoh berdasarkan rilis terbaru dari situs *The Next Web*, aplikasi populer WhatsApp yang dikenal memiliki fasilitas mengirim pesan ala *blackberry messenger* diketahui memiliki celah keamanan yang memungkinkan orang lain menyadap isi pesan. berdasarkan beberapa laporan dari situs berbahasa Spanyol dan Belanda, proses penyadapan pesan dapat dilakukan dengan menggunakan telepon prabayar dan koneksi WiFi.

Oleh karena itu penulis bermaksud membuat aplikasi mobile messenger yang sistem enkripsinya sukar untuk dipecahkan oleh pihak yang tidak berhak menerima pesan dan juga memiliki ukuran kunci yang pendek sehingga sangat cocok diaplikasikan di telepon seluler, karena alasan tersebut, penulis memilih membuat aplikasi *mobile messenger* menggunakan algoritma ECC sebagai metode enkripsi pesannya.

ECC (*Elliptic Curve Cryptography*) termasuk ke dalam sistem kriptografi kunci publik yang mendasarkan keamanannya pada permasalahan matematis kurva.

## 1.3 RUMUSAN MASALAH

Rumusan masalah skripsi ini sebagai berikut

Bagaimana membangun aplikasi *mobile messenger* berbasis J2ME agar dapat melakukan proses enkripsi dan dekripsi pesan dengan mengimplementasikan algoritma *Elliptic Curve Cryptography*.

## 1.4 BATASAN MASALAH

Batasan masalah skripsi ini sebagai berikut :

1. Membuat Aplikasi *mobile messenger* berbasis J2ME.
2. Aplikasi bersifat *client set* yang terdiri dari dua *client* dan satu *server*.
3. Aplikasi dapat mengimplementasikan algoritma *Elliptic Curve Cryptography* pada proses enkripsi dan dekripsinya.

## 1.5 TUJUAN PENELITIAN

Tujuan pembuatan skripsi ini sebagai berikut :

Membuat aplikasi *mobile messenger* berbasis J2ME yang dapat melakukan proses enkripsi dan dekripsi dengan mengimplementasikan algoritma *Elliptic Curve Cryptography*

## 1.6 MANFAAT PENELITIAN

1. Aplikasi ini berformat *mobile* sehingga sangat sesuai dengan kebutuhan penggunanaan masyarakat pada saat ini dimana telepon seluler telah menjadi bagian yang tak terpisahkan dari kehidupan masyarakat.
2. Pada aplikasi ini pengiriman dan penerimaan pesan menggunakan algoritma ECC sehingga pesan yang terkirim dan diterima dapat terenkipsi sehingga pihak lain dalam hal ini pihak yang tidak berhak menerima tidak dapat membaca pesan tersebut.

## 1.7 METODOLOGI PENELITIAN

Untuk Mencapai tujuan yang telah dirumuskan sebelumnya maka metodologi yang digunakan adalah

1. **Studi Literatur**

   Mempelajari penelitian terkait, buku-buku literature, sumber dari internet yang berhubungan dengan penyelesaian skripsi, adapun topik-topik yang dipelajari meliputi Algoritma ECC, pembuatan aplikasi *mobile*, *client-server*, dan keamanan jaringan

2. **Perancangan Sistem**

   Merancang Pembuatan aplikasi *mobile messenger* menggunakan metode algoritma ECC (*Elleptic Curve Cryptography*)

3. **Implementasi**

   Mengimplementasikan rancangan system dengan cara membangun serta membuat perangkat lunak aplikasi *mobile messenger* menggunakan metode algoritma ECC sesuai dengan perancangan sistem

4. **Uji coba dan evaluasi**

   Menguji perangkat lunak yang sudah dibuat dan kemudian menganalisa hasil *output* apakah sudah sesuai dengan tujan yang dirumuskan.

5. **Penulisan Laporan Penelitian**

Pada tahap ini dilakukan penyusunan laporan sebagai dokumentasi dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil yang telah didapatkan selama pengerjaan skripsi.

## 1.8 SISTEMATIKA PENULISAN

Sistematika yang digunakan dalam penyusunan skripsi ini adalah sebagai berikut :

**BAB I : Pendahuluan**

Berisi tentang gambaran umum latar belakang penulisan skripsi, rumusan masalah, tujuan, batasan masalah, metode penelitian.

**BAB II : Tinjauan Pustaka**

Membahas tentang teori penunjang dari pembahasan masalah sebagai berikut tentang komponen yang digunakan dalam pembuatan program ini.

**BAB III : Analisis Dan Perancangan Sistem**

Berisi tentang rancangan detail tentang perangkat yang akan digunakan serta prinsip kerja dari sistem secara keseluruhan.

**BAB IV : Hasil dan Pembahasan**

Berisi tentang pengujian aplikasi dengan cara melakukan enkripsi dan dekripsi pada pesan dan menganalisa hasil *output* pada program.

**BAB V : Penutup**

Berisi hasil ringkasan dan uraian keseluruhan skripsi, serta saran-saran untuk pengembangan program selanjutnya.

# BAB II

# TINJAUAN PUSTAKA

## 2.1 Keamanan Jaringan

Pada komunikasi data antar *device* yang bersifat global disitu keamanan menjadi hal yang utama. sebagai contoh, dengan berpindahnya data dari titik A ke titik B ia akan melalui beberapa titik lain selama perjalanan, membuka kesempatan bagi pihak lain untuk memotong data, merubah data bahkan merubah tujuan data.

### 2.1.1 Hacker, Cracker dan Motif

*Cracker* merupakan individu maupun kelompok orang yang memanfaatkan hasil penemuan penyusupan terhadap komputer lain untuk melakukan eksploitasi dan mengambil manfaat dari hasilnya. Seorang *cracker* dapat melakukan eksploitasi di mana saja dan kapan saja, tanpa harus mempunyai pengetahuan khusus. *Cracker* jenis ini dikenal sebagai *script kiddies*. Motivasi para *cracker* sangat beragam, diantaranya adalah untuk propaganda *( deface web site / email )*, kriminal murni, penyerangan destruktif (akibat dendam atau ketidaksukaan terhadap suatu insitusi), dan lain-lain. Apapun motif dari *cracker* selalu ada pihak yang dirugikan akibat tindakannya.

Berbeda dengan *Cracker*, *Hacker* adalah entitas yang menemukan kelemahan *(vunerability)* sistem dalam konteks *security incidents*.

Seorang *hacker* bisa menjadi seorang *cracker*, tetapi seorang *cracker* belum tentu menguasai kemampuan yang dipunyai seorang *hacker*.

Motivasi para *hacker* untuk menemukan *vunerability* adalah untuk membuktikan kemampuannya atau sebagai bagian dari kontrol sosial terhadap sistem.

Pada prakteknya suatu pembentukan sistem yang aman akan mencoba melindungi adanya beberapa kemungkinan serangan yang dapat dilakukan pihak lain terhadap kita antara lain :

1. *Intrusion.*

   Pada penyerangan ini seorang penyerang akan dapat menggunakan sistem komputer yang kita miliki. Sebagian penyerang jenis ini menginginkan akses sebagaimana halnya pengguna yang memiliki hak untuk mengakses sistem.

2. *Denial of services.*

   Penyerangan jenis ini mengakibatkan pengguna yang sah tak dapat mengakses sistem. Sebagai contoh adalah *Distributed Denial of Services* (DDOS) yang mengakibatkan beberapa situs Internet tak bisa diakses. Seringkali orang melupakan jenis serangan ini dan hanya berkonsentrasi pada *intrusion* saja..

3. *Joyrider.*

   Pada serangan ini disebabkan oleh orang yang merasa iseng dan ingin memperoleh kesenangan dengan cara menyerang suatu sistem. Mereka masuk ke sistem karena beranggapan bahwa mungkin data yang di

dalamnya menarik. Rata-rata mereka karena rasa ingin tahu, tapi ada juga yang menyebabkan kerusakan atau kehilangan data.

4. *Denial of services*.

   Penyerangan jenis ini mengakibatkan pengguna yang sah tak dapat mengakses sistem. Sebagai contoh adalah *Distributed Denial of Services* (DDOS) yang mengakibatkan beberapa situs Internet tak bisa diakses. Seringkali orang melupakan jenis serangan ini dan hanya berkonsentrasi pada *intrusion* saja.

5. *Vandal*.

   Jenis serangan ini bertujuan untuk merusak sistem. Seringkali ditujukan untuk site-site besar.

6. *Scorekeeper*.

   Jenis serangan in hanyalah bertujuan untuk mendapatkan reputasi dengan cara mengcrack sistem sebanyak mungkin. Sebagian besar dari mereka tertarik pada situs-situs tertentu saja. Sebagian dari mereka tak begitu peduli dengan data yang ada di dalamnya. Saat ini jenis ini lebih dikenal dengan istilah *script kiddies.*

7. Mata-mata.

   Jenis serangan ini bertujuan untuk memperoleh data atau informasi rahasia dari pihak kompetitor. Saat ini semakin banyak perusahaan yang memanfaatkan jasa ini.

Terdapat beberapa macam mata-mata, yaitu :

- *The Curious* (Si Ingin Tahu)

  Tipe penyusup ini pada dasarnya tertarik menemukan jenis sistem dan data yang anda miliki.

- *The Malicious* (Si Perusak)

  Tipe penyusup ini berusaha untuk merusak sistem, atau merubah web page anda, atau sebaliknya membuat waktu dan uang anda kembali pulih.

- *The High-Profile Intruder* (Si Profil Tinggi)

  Tipe penyusup ini berusaha menggunakan sistem untuk memperoleh popularitas dan ketenaran. Dia mungkin menggunakan sistem profil tinggi anda untuk mengiklankan kemampuannya.

8. *The Competition*

   Tipe penyusup ini tertarik pada data dalam sistem. (Ariyus, 2008)

### 2.1.2 Jenis Serangan *Security*

Serangan pada suatu sistem jaringan komputer sendiri pada dasarnya memiliki 3 gelombang trend utama yaitu:

1. Gelombang pertama adalah serangan fisik

   Serangan ini ditujukan kepada fasilitas jaringan, perangkat elektronis dan komputer. Sebagai pertahanan terhadap serangan jenis ini biasanya digunakan sistem backup ataupun sistem komputer yang terdistribusi, sehingga mencegah kesalahan di satu titik mengakibatkan seluruh sistem menjadi tak bekerja. Cara pemecahan terhadap serangan ini telah

diketahui dengan baik. Jaringan Internet sendiri didisain untuk mengatasi permasalahan seperti ini.

2. Gelombang pertama adalah serangan sintatik

   Serangan ini ditujukan terhadap keringkihan (*vulnerability*) pada perangkat lunak, celah yang ada pada algoritma kriptografi atau protokol. Serangan *Denial of Services* (DoS) juga tergolong pada serangan jenis ini. Serangan jenis inilah yang saat ini paling populer. Tetapi relatif cara penanganannya telah diketahui dan biasanya pihak administrator atau pengguna yang lalai menerapkannya.

3. Gelombang pertama adalah serangan semantik

   Serangan jenis ini memanfaatkan arti dari isi pesan yang dikirim. Dengan kata lain adalah menyebarkan disinformasi melalui jaringan, atau menyebarkan informasi tertentu yang mengakibatkan timbulnya suatu kejadian. Pada dasarnya banyak pengguna cenderung percaya apa yang mereka baca. Seringkali keluguan mempercayai berita ini disalah-gunakan pihak tertentu untuk menyebarkan *issue-issue* yang menyesatkan. (Schneier, 2000)

## 2.2 Kriptografi

Berkomunikasi satu sama lain merupakan salah satu sifat dasar manusia sejak ada di muka bumi ini. bagi manusia komunikasi berfungsi sebagai sarana untuk saling memahami satu sama lain. cara manusia berkomunikasi dan zaman dulu sampai sekarang terus mengalami perkembangan.

Salah satu sarana komunikasi manusia adalah tulisan. tulisan berfungsi untuk menyampaikan pesan kepada pembacanya. Pesan itu sendiri merupakan suatu informasi yang dapat dibaca dan dimengerti maknanya. Ketika kertas belum Salah satu sarana komunikasi manusia adalah tulisan, tulisan berfungsi untuk menyampaikan pesan kepada pembacanya.

Pesan terbagi dalam beberapa bagian, seperti:

1. Pesan untuk orang banyak : Suatu Informasi yang ditujukan untuk orang banyak yang tidak mengandung suatu rahasia
2. Pesan untuk suatu kelompok,: suatu informasi untuk beberapa orang (kelompok), kadang bersifat rahasia.
3. Pesan hanya untuk satu orang: pesan hanya untuk satu orang seringkali bersifat rahasia
4. Pesan rahasia: pesan yang tidak boleh diketahui orang lain selain yang berhak menerima pesan.

Pada dasarnya komponen kriptografi terdiri dari beberapa komponen, seperti:

1. Enkripsi: merupakan hal yang sangat penting dalam kriptografi. merupakan cara pengamanan data yang dikirimkan sehingga terjaga kerahasiaannya. Enkripsi bisa diartikan dengan cipher atau kode.
2. Dekripsi: merupakan kebalikan dan enkripsi. Pesan yang telah dienkripsi dikembahikan ke bentuk asalnya. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan yang digunakan untuk enkripsi.
3. Kunci: adaiah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian. yaitu kunci rahasia (private key dan kunci umum (*public key'*).
4. *Chipertext*: merupakan suatu pesan yang telah melalui proses enkripsi. Pesan yang ada pada teks-kode ini tidak bisa dibaca karena berupa karakter-karakter yang tidak mempunyai rnakna (arti).
5. *Plaintext* sering disebut dengan *cleartext*. Teks-asli atau teks-biasa, merupakan pesan yang ditulis atau diketik yang memiliki makna. Teks inilah yang diproses menggunakan algoritma kriptografi untuk menjadi *ciphertext* (teks-kode). mengenkripsi dan mendekripsi data.

Kriptografi menggunakan suatu algoritma (*cipher*) dan kunci (*key*). *Cipher* adalah fungsi matematika yang digunakan untuk mengenkripsi dan mendekripsi. Sedangkan kunci merupakan sederetan bit yang diperlukan untuk mengenkripsi dan mendekripsi data.

Suatu pesan yang tidak disandikan disebut sebagai *plaintext* ataupun dapat disebut juga sebagai *cleartext*. Proses yang dilakukan untuk mengubah *plaintext* ke dalam *ciphertext* disebut *encryption* atau *encipherment*. Sedangkan proses untuk mengubah ciphertext kembali ke plaintext disebut *decryption* atau *decipherment*.

Suatu pesan yang tidak disandikan disebut sebagai *plaintext* ataupun dapat disebut juga sebagai *cleartext*. Proses yang dilakukan untuk mengubah *plaintext* ke dalam *ciphertext* disebut *encryption* atau *encipherment*. Sedangkan proses untuk mengubah *ciphertext* kembali ke plaintext disebut *decryption* atau *decipherment*.

*Cryptographic system* atau *cryptosystem* adalah suatu fasilitas untuk mengkonversikan *plaintext* ke *ciphertext* dan sebaliknya. Dalam sistem ini, seperangkat parameter yang menentukan transformasi pencipheran tertentu disebut suatu set kunci. Proses enkripsi dan dekripsi diatur oleh satu atau beberapa kunci kriptografi. Secara umum, kunci-kunci yang digunakan untuk proses pengenkripsian dan pendekripsian tidak perlu identik, tergantung pada sistem yang digunakan.

Prinsip-prinsip yang mendasari kriptografi yakni:

1. *Confidelity* (kerahasiaan) yaitu layanan agar isi pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima / pihak-pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.

2. Data *integrity* (keutuhan data) yaitu layanan yang mampu mengenali/mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan) data yang tidak sah (oleh pihak lain).
3. *Authentication* (keotentikan) yaitu layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi.
4. Nirpenyangkalan (*non-repudiation*), yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dari dirinya)

Sebagai contoh, misalkan ada seorang agen rahasia Eva yang ingin melaporkan hasil kerjanya kepada pimpinannya Roy. Isi pesan yang dikirimkan Eva kepada Roy tidak boleh diketahui oleh orang lain (kerahasiaan) dan Roy harus dapat memastikan bahwa pengirim pesan adalah Eva, bukan orang lain dari pihak lawan (otentikasi).

Selain itu, isi pesan yang dikirimkan Eva harus sama dengan isi pesan yang diterima Roy, orang lain dari pihak lawan tidak dapat mengganti isi pesan Eva (integritas). Eva juga tidak dapat menyangkal bahwa ia telah mengirimkan pesan kepada Roy (anti penyangkalan). Dalam kenyataan, contoh Eva dan Roy di atas tidak harus manusia. Pihak–pihak yang bertukar informasi mungkin saja merupakan komputer–komputer di suatu jaringan seperti Internet. Salah satu cara mengatasi masalah kerahasiaan adalah kriptografi.

Kriptografi (*cryptography*) berasal dari bahasa Yunani: "*cryptos*" artinya "*secret*" (rahasia) dan "*graphein*" artinya "*writing*" (tulisan). Jadi kriptografi berarti "*secret writing*" (tulisan rahasia). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. Istilah lain yang sering digunakan dalam kriptografi adalah kunci, enkripsi dan dekripsi. Kunci yang dimaksud adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Enkripsi adalah proses mengubah pesan asli (*plainteks*) menjadi data sandi (*cipherteks*) dan dekripsi adalah proses mengembalikan *cipherteks* menjadi *plainteks*. (Ariyus, 2008)

## 2.3 Algoritma ECC

Kriptografi kurva eliptik termasuk kedalam sistem kriptografi kunci publik yang mendasarkan keamanannya pada permasalahan matematis kurva eliptik. Tidak seperti permasalahan matematis logaritma diskrit (*Discrete Logarithm Problem*, DLP) dan pemfaktoran bilangan bulat (*Integer Factorization Problem*, IFP), tidak ada algoritma waktu subeksponensial yang diketahui untuk memecahkan permasalahan matematis logaritma diskrit kurva eliptik (*Elliptic Curve Discrete Logarith* Problem, ECDLP).

Karena alasan tersebut, algoritma kriptografi kurva eliptik mempunyai keuntungan jika dibandingkan dengan algoritma kriptografi kunci publik lainnya yaitu dalam hal ukuran panjang kunci yang lebih pendek tetapi

memiliki tingkat keamanan yang sama. *Elliptic Curve Cryptography* (ECC) adalah salah satu pendekatan algoritma kriptografi kunci publik berdasarkan pada struktur aljabar dari kurva ellips pada daerah finite.

Dari perbandingan Antara algoritma ECC dengan sistem kriptografi lainnya dapat disimpulkan:

- Algoritma ECC membutuhkan ukuran kunci yang lebih sedikit untuk mendapatkan level keamanan yang sama.
- Algoritma ECC sesuai untuk lingkungan yang memiliki resouce terbatas seperti perangkat *mobile*.
- Algoritma ECC merupakan algoritma yang compact dan sangat efisien.
- Implementasi algoritma ECC menggunakan bahasa javascript tidak powerful sesuai dengan penelitian ahli kriptografi

Karena kompleksitas inilah mengapa pada *javascript* kecepatan pemrosesan enkripsi data pada algoritma ecc tidak lebih cepat dibandingkan dengan beberapa algortima kunci publik lainnya.

*Elliptic Curve Cryptography* Pendekatan yang dilakukan untuk menhasilkan algoritma *Elliptic Curve Cryptography* adalah dengan meggunakan struktur matematika yang sangat unik yang memungkinkan pemrosesan titik dengan memiliki dua buah titik dalam kurva eliptik dan menghasilkan titik lain yang ada pada kurva tersebut. Struktur yang nik ini memberikan keuntungan dalam kriptografi dikarenakan kesulitan

Untuk menemukan 2 buah titik yang menentukan titik tertentu tersebut tidak dapat ditemukan dengan mudah. Tingkat kesulitan untuk menemukan 2 buah titik termasuk dalam golongan yang rumit sama seperti kesulitan untuk memperhitungkan variasi *eksponensial* yang digunakan dalam algoritma RSA yang telah banyak diimplementasikan. Untuk memecahkan *Elliptic Curve Cryptography* sendiri dibutuhkan perhitungan matematis yang sangat tinggi.

*Elliptic Curve Cryptography* terdiri dari beberapa operasi *basic* dan juga aturan yang mendefinisikan penggunaan dari operasi operasi *basic* seeperti penambahan, pengurangan, perkalian dan perpangkatan yang didefinisikan sesuai dengan kurva-kurva yang ada

Dasar Matematika pada *Elliptic Curve Cryptography* Operasi matematika yang digunakan pada *Elliptic Curve Cryptography* didefinisikan dengan persamaan:

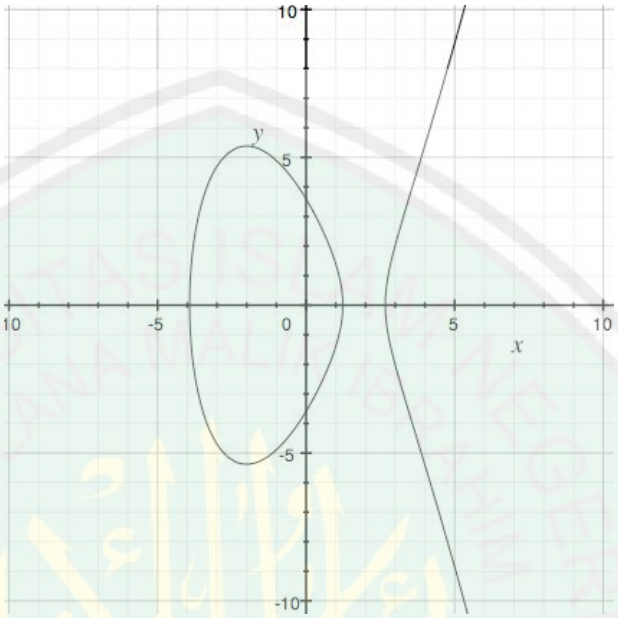
$$y^2 = x^3 + ax + b$$

Tetapi dengan syarat

$$4a^3 + 27b^2 \neq 0$$

Setiap perubahan nilai dari ‘a’ dan’b’ pada persamaan diatas akan menghasilkan *elliptic curve* yang berbeda.

Contoh *Elliptic Curve* yang menggunakan persamaan yang berbeda

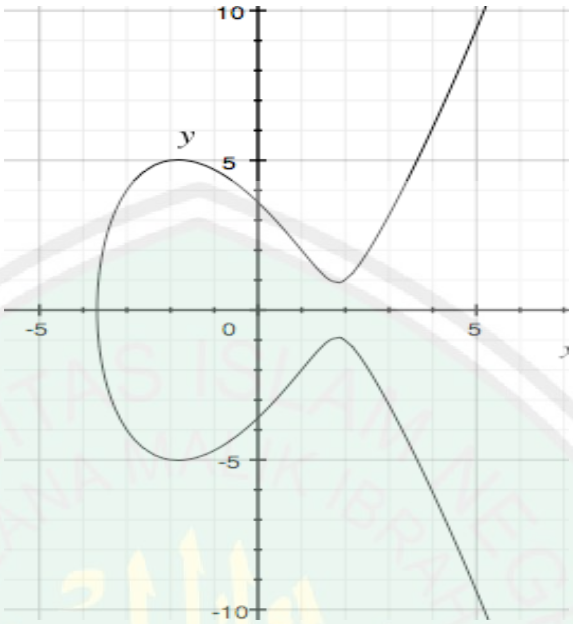



Gambar 2.1 Kurva elliptik dengan persamaan

$$y^2 = x^3 - 12x + 3$$

Pada contoh gambar 2.1 diatas adalah contoh kurva elliptik dari bentuk persamaan tersebut, sedangkan gambar 2.2 dibawah ini adalah kurva elliptik dari bentuk persamaan

$$y^2 = x^3 - 10x + 13$$




Gambar 2.2 Kurve elliptik dengan persamaan

$$y^2 = x^3 - 10x + 13$$

gambar diatas adalah menunjukkan hasil sari perhitungan persamaan

$$y^2 = x^3 - 10x + 13$$

### 2.3.1 Himpunan Pada Kurva Ellips

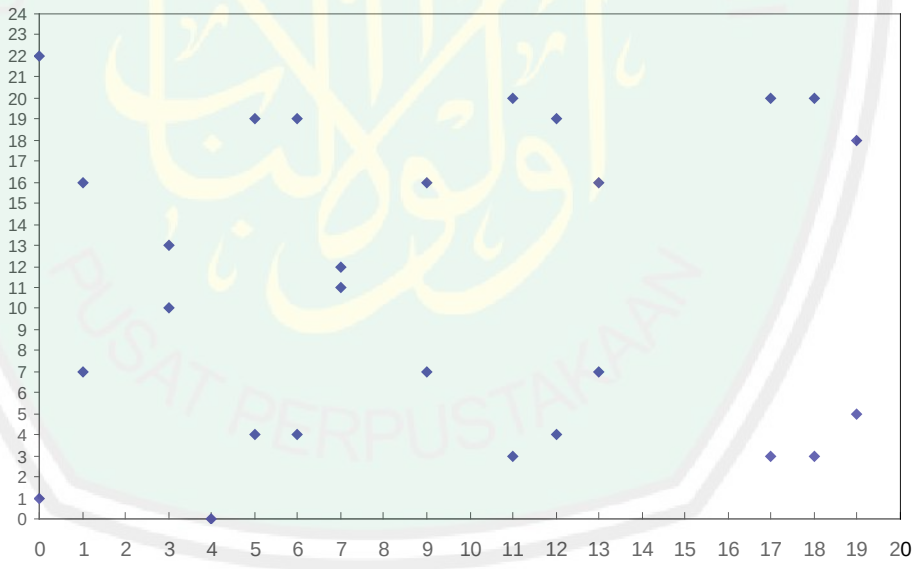
Pada teknik dasar kurva elips dalam grup Zp dimana p adalah bilangan prima yang lebih besar dari 3, Dimana $a,b \in Z_p$ dan $4a^3 + 27b^2 \neq 0 \pmod{p}$, dan titik *O* yang disebut dengan titik *infinity*. Himpunan $E(Z_p)$ adalah semua titik (x,y), untuk x,y $\in Z_p$, yang memenuhi persamaan pada titik *O*.

Berikut ini adalah contoh pencarian himpunan $E(Z_p)$.

Diberikan persamaan kurva elips $E: y2 = x3 + x + 1$ dengan $p$ = 23, yaitu grup $Z_{23}$ ( pada persamaan $a = b = 1$ ).

Maka untuk nilai $E$ ada dalam kurva elips. Titik-titik dalam $E(Z_{23})$ adalah :

(0,1) (6,4) (12,19) (0,22) (6,19) (13,7) (1,7) (7,11) (13,16) (1,16) (7,12) (17,3) (3,10) (9,7) (17,20) (3,13) (9,16) (18,3) (4,0) (11,3) (18,20) (5,4) (11,20) (19,5) (5,19) (12,4) (19,18)




Gambar 2.3 Sebaran titik-titik pada kurva elips E($P_{23}$) dengan persaman

$$y^2 = x^3 + x + 1$$

Pada gambar 2.1 adalah sebaran titik-titik himpunan (E) pada kurva ellips yang didapat dari hasil persamaan

$$y^2 = x^3 + x + 1.$$

### 2.3.2 Bidang Terbatas (*Finite Field*)

Bidang terbatas (*finite field*) atau yang biasa disebut dengan *Galois Field* (GF) adalah bidang yang hanya memiliki elemen bilangan yang terbatas yang ditentukang dengan suatu pembatasan yang abstrak. Derajat atau sering disbuat juga dengan order dari finite field adalah banyaknya elemen yang ada di dalam bidang yang didefinisikan. Jika q adalah pangkat prima (*prime power)*, maka hanya ada satu bidang terbatas dengan derajat q, bidang tersebut dilambangkan dengan Fq atau GF(q).

Banyak cara untuk merepresentasikan elemen dari Fq, jika q=pm , dimana p adalah bilangan prima dan m adalah bilangan integer positif, maka p disebut sebagai karakteristik yang unik dari Fq dan m disebut sebagai derajat perluasan (*extension degree*) dari Fq.

Bidang terbatas yang digunakan dalam *kriptografi* adalah q=p, dimana p adalah bilangan prima ganjil, yang dilambangkan dengan Fp (*odd prime* dimana m adalah integer lebih besar dari satu, yang dilambangkan dengan F2m (*characteristic two or even*).

Bidang Terbatas Fp merupakan bidang yang beranggotakan bilangan integer

$$\{0,1\ , p-1\}$$

dan p merupakan bilangan prima, setiap perhitungan dikalkulasikan dengan *modulo* hasilnya tetap berada dalam daerah Fp.

Bidang terbatas F2m biasa disebut dengan bidang terbatas biner (binary finite field), dapat dipandang sebagai ruang vektor berdimensi m pada F2. Karena itu ada himpunan yang beranggotakan m elemen {α0, α1, αm-1} di dalam F2m sedemikian rupa msehingga setiapa Є F2m dapat ditulis secara unik ke dalam bentuk:

$$a\ =\ a0\ \alpha 0 + a1\alpha 1 + am - 1\alpha m - 1$$

untuk

$$\text{Є}\ \{0,1\}$$

Salah satu cara untuk merepresentasikan elemen-elemen pada F2m adalah dengan representasi basis polinomial. Pada representasi basis polinomial elemen pada F2m merupakan polinomial dengan derajat lebih kecil dari m, dengan koefisien bilangan 0 atau 1

$$\{am - 1xm - 1\ + a2x2\ +\ a1x1\ +\ a0x0 \mid ai\ :\ 0,1\}$$

### 2.3.3 Jenis – Jenis Elliptic Curve Cryptography

Ada beberapa jenis kurva elleptik, yaitu:

1. ECDSA – *Elliptic Curve Digital Signature Algorithm*

   Algoritma pnandatanganan pesan menggunakan ECC yang disbutkan sebadi ECDSA adalah salah satu variasi dari *Digital Signature Algorithm* yang beroperasi dengan kelompok kurva elliptic sebagai basis perhitungan dari proses penandatanganan. Agar dapat menyamakan suatu tandatangan digital dari sbuah psan yang dikirim oleh dua orang, maka kedua orang tersebut harus memiliki kurva elliptik yang sama. Seorang pngirim pesan yang akan ditandatangani akan memiliki kunci pribadi yang merupakan sbuah integer yang dipilih acak kurang dari n yangmerupakan urutan kurva, padametr kurva eliptik domain. Dan kunci publik yang merupakan titik yang digenrasikan dengan kurva eliptik

2. *Elliptic Curve Diffie Hellman*

   ECDH - *Elliptic Curve Diffie Hellman* adalah protokol perjanjian kunci yang memungkinkan dua pihak pngirim dan penrima , yang pada awalnya masing-masing memiliki kurva eliptik sepasang kunci publik-swasta masing –masing, dan mengininkan kunci rahasia bersama melalui saluran yang tidak aman.

Berbagi rahasia ini mungkin langsung digunakan baik lagi, untuk mendapatkan kunci lain yang kemudian dapat digunakan untuk mengenkripsi komunikasi berikutnya menggunakan *cipher* kunci simetris.

Ini adalah varian dari protokol *Diffie-Hellman* yang digunakan untuk menyamakan kunci menggunakan kriptografi kurva eliptik.

Misalkan Pengirim ingin mendirikan kunci bersama dengan penerima, tapi saluran hanya tersedia bagi mereka mungkin melihat oleh pihak ketiga. Awalnya, parameter domain yaitu (p, a, b, G, n, h) dalam kasus perdana atau m, f (x), a, b, G, n, h dalam kasus biner harus disepakati.

Selain itu, masing-masing pihak harus memiliki sepasang kunci yang cocok untuk kriptografi kurva eliptik, terdiri dari kunci pribadi yang dipilih secara acak dalam interval

$$[1\, n\, -\, 1]$$

dan kunci publik dimana

$$Q\, =\, DG$$

Sepasang kunci pengirim akan (dA, QA) dan sepasang kunci Penerima akan (dB, QB). Setiap pihak harus memiliki kunci publik pihak lain (Ariyus, 2009).

# BAB III

# ANALISIS DAN PERANCANGAN SISTEM

## 3.1 Analisis Kebutuhan

Analisis kebutuhan merupakan analisis terhadap komponen-komponen yang digunakan untuk pembuatan aplikasi *mobile messenger* menggunakan metode algoritma Ecc (*Elliptic Curve Cryptography*), dalam hal ini komponen yang dibutuhkan menjadi dua macam, yaitu komponen *software* dan *hardware*

### 3.1.1 *Software*

Software adalah program atau aplikasi komputer lain yang dibutuhkan untuk membangun sebuah sistem. *Software* yang dibutuhkan dalam pembuatan aplikasi *mobile messenger* menggunakan metode algoritma ECC (*Elliptic Curve Cryptography*) sebagai berikut:

1. Windows 7

   Windows 7 merupakan sistem operasi yang menjembatani antara computer dan *user*. Sistem operasi ini digunakan karena support dan *compatible* dengan *software* lain yang dibutuhkan dalam pembuatan aplikasi ini.

2. *Java Development Kit* (JDK) versi 6

JDK merupakan paket *platform* java yang terdiri dari berbagai macam *library, compiler* dan *debugger*. (Raharjo, 2010)

3. Editor Netbeans 7.1.2

   Netbeans adalah sebuah IDE (*Integrated Development Environment*) yang menyediakan berbagai macam platform, khususnya Java. Netbeans mendukung semua jenis pengembangan bahasa Java mulai dari *Java SE*, ME, dan Web. (Raharjo, 2010)

4. Jabber 2.1.6

   Jabber adalah sebuah server yang berbasis protokol *extensible messaging and presence protocol* (XMPP), di sediakan secara gratis atau terbuka (*open source*) yang berfungsi sebagai pengatur pertukaran pesan antara dua *client* atau lebih. (Raharjo, 2010)

### 3.1.2 *Hardware*

Hardware adalah perangkat keras atau *device* yang menunjang dalam pembuatan sebuah sistem, dalam pembuatan aplikasi *mobile messenger* menggunakan metode algoritma ECC (*Elliptic Curve Cryptography*) sebagai berikut :

1. Komputer atau Laptop

   Dalam hal ini komputer merupakan komponen utama yang digunakan untuk membangun aplikasi *mobile messenger* menggunakan metode

algoritma Ecc (*Elliptic Curve Cryptography*). serta sebagai komponen yang digunakan untuk menjalankan aplikasi *client* dan *server*

2. Telepon seluler

   Telepon seluler dalam hal ini digunakan untuk dipasang aplikasi, telepon seluler yang digunakan yang dapat mendukung aplikasi J2ME.

3. *WIFI Adapter*

   Wifi adapter adalah *device* yang memncarkan dan menerima *signal wifi* yang dapat digunakan sebagai penghubung antara *client* pada telepon seluler

## 3.2 Spesifikasi Aplikasi

Spesifikasi aplikasi *mobile messenger* menggunakan metode algoritma ECC (*Elliptic Curve Cryptography*) dibagi menjadi dua bagian yaitu spesifikasi aplikasi *client* dan spesifikasi *server*.

### 3.2.1 Spesifikasi *Client*

Aplikasi yang akan dibangun untuk *client* memiliki kemampuan diantaranya:

1. Menambah *Contact*

2. Menghapus *Contact*

3. Menulis pesan

4. Mengirim pesan

5 Menerima pesan

3.2.2 Spesifikasi *Server*

Server pada aplikasi ini memiliki beberapa kemampuan diantaranya:

1. Mampu mengkoneksikan antar *client*
2. Mampu mengirimkan pesan yang telah dienkripsi

**3.3 Perancangan Sistem**

Sistem yang dirancang agar dapat mengirim pesan yang terenkripsi menggunakan dua buah *mobile phone* dan menggunakan satu laptop sebagai *wifi acces point* dapat dilihat pada gambar 3.1 sebagai berikut


Gambar 3.1 Arsitektur Sistem

Sistem ini terdiri dari dua komponen, yaitu:

1. *Clent Sender* atau *Receiver*

   Komponen ini membuat perancangan mobile messenger yang dapat berjalan di ponsel dan mentransformasikan data *plaintext* ke dalam bentuk *ciphertext. Ciphertext* inilah yang kemudian dikirimkan oleh sender kepada *receive*r, selanjutnya setelah sampai di penerima *ciphertext* ditransformasikan kembali kedalam bentuk *plaintext* agar dapat dibaca.

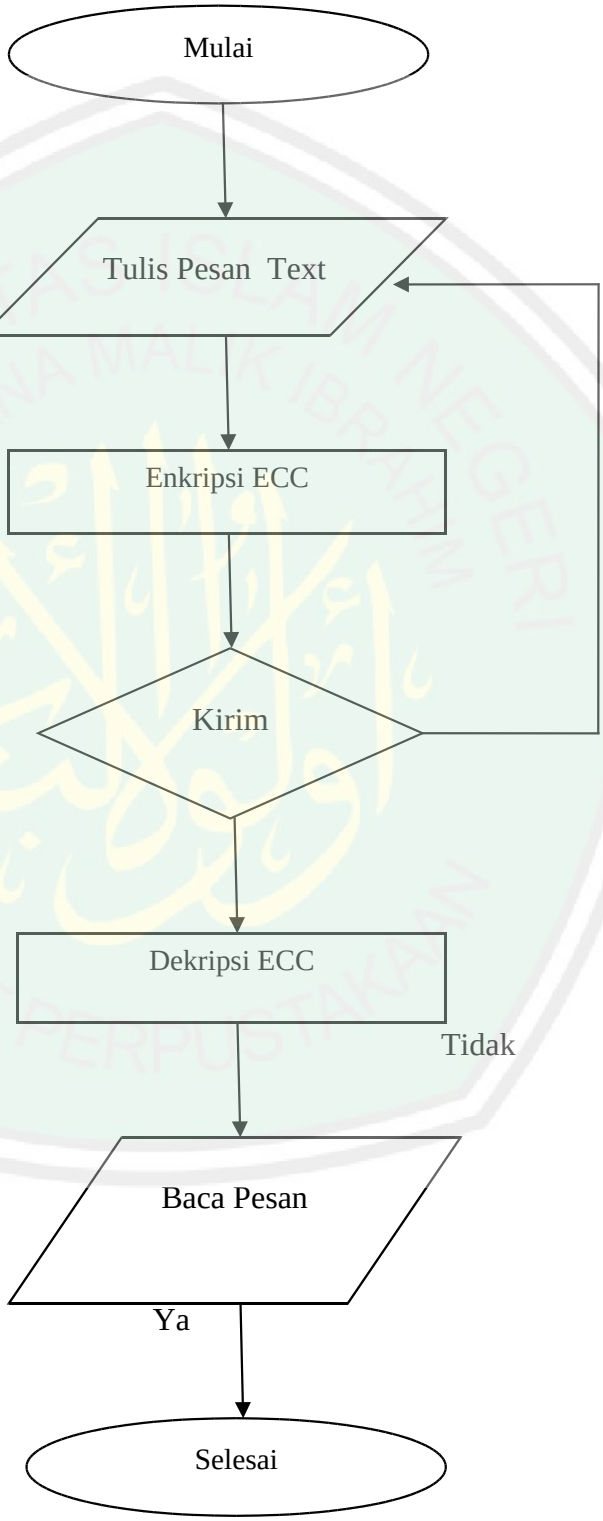
2. Wifi *access point*

   Komponen ini adalah data *center* yang menjadi pertukaran antar sender dan *receiver*, menghubungkan dengan wifi dan berfungsi

meneruskan pesan yang telah dikirim oleh *sender* sehingga samapi kepada *receiver*.

### 3.4 *Flowchart System*

*Flowchart System* aplikasi dapat dijelaskan melalui Gambar 3.2



Gambar 3.2 *Flowchart System*

Pada *flowchart system* yang ada pada gambar 3.2 dapat dijelaskan sebagai berikut.

1. Pesan

   Pada proses ini *client* menulis pesan, pesan berupa text terdiri dari huruf, angka dan simbol, tetapi tidak termasuk gambar dan *emotion*, karena penulisan ini tidak *support* gambar dan *emotion.*

2. Enkripsi ECC

   Pada proses ini pesan dienkripsi, pesan yang berupa *plaintext* diubah menjadi *chipertext* menggunakan algoritma ECC (*Elliptic Curve Cryptography*)

3. Kirim

Saat pesan telah di enkripsi dan telah diubah dari *plaintext* menjadi *chipertext* pesan kemudian dikirim oleh *server* ke penerima, saat pesn tidak terkirim maka pengirim pesan harus menulis pesan kembali, tetapi jika pesan dapat dikirim maka dilanjutkan ke proses selanjutnya.

4. Dekripsi ECC

   Saat pesan yang diterima berupa *chipertext* diproses dekripsi ini pesan diubah kembali menjadi pesan *plaintext*.

5. Baca Pesan

   Pesan yang di[illegible]li *plaintext* dapat dibaca seperti pesan a[illegible]

## 3.5 Proses Enkripsi

### 3.5.1 Enkripsi Pes[illegible]

Adapun proses enk[illegible]ui Gambar 3.3.

Gambar 3.3 Blok Diagram Enkripsi Pesan

Untuk mengenkripsi pesan pada aplikasi ini menggunakan algoritma ECC, terlebih dahulu mencoba melakukan langkah untuk menghitung proses

enkripsi dengan perhitungan algoritma ECC sesuai pada gambar 3.3, proses-proses enkripsi pesan dapat dijelaskan sebagai berikut:

.1 Langkah pertama adalah menentukan persamaan, persamaan yang digunakan adalah $y^2=x^3+x+1$

.2 Langkah kedua adalah menentukan bilangan prima, private key, dan titik awal kurva

- Bilangan prima adalah bilangan asli yang lebih besar dari 1, yang faktor pembaginya adalah 1 dan bilangan itu sendiri.
- *Private key* adalah kunci kunci enkripsi yang hanya boleh diakses oleh pemilik kunci
- Titik awal kurva digunakan sebagai titik awal pengenkripsi pesan

.3 Selanjutnya adalah mencari *publickey*, *publickey* didapat dari hasil perkalian antara *privatekey* dengan titik awal kurva.

.4 Mencari titik kkp sebagai titik pengenkripsi, titik kkp didapat dari perkalian *publickey* dengan *privatekey*

.5 Ambil titik *absis* kkp kemudian di xor kan, titik absis adalah titik awal titik absis adalah unsur pertama dari pasangan terurut

.6 Mengkonversi huruf atau angka menjadi bilangan biner sesuai bilangan *ASCII.*

.7 Kemudian bilangan biner di xor kan dengan titik absis.

### 3.5.2 **Dekripsi Pesan**

Proses dekripsi pesan dapat dijelaskan Gambar 3.4

Gambar 3.4 Blok Diagram Dekripsi Pesan

Pada gambar 3.4 adalah proses dekripsi pesan, proses tersebut dapat dijelaskan sebagai berikut.

.1 Ambil titik kp, titik kp adalah kunci public yang telah didapat dari proses enkripsi pesan

.2 Mencari titik kkp, titik kkp digunakan sebagai titik dekripsi, titik kkp didapat dari kunci privat dengan kunci public.

.3 Ambil titik absis kemudian di xor kan

.4 Konversi pesan yang terenkripsi perhuruf ke bilangan integer sesuai dengan bilangan *ASCII*

.5 Bilangan biner yang didapat dari konversi pesan terenkripsi kemudian di xor kan dengan hasil xor titik absis

.6 Hasil yang didapat berupa bilangan decimal kemudian dirangkai kembali menjadi pesan sesuai bilangan ASCII

### 3.5.3 Perhitungan Algoritma ECC (*Elliptic Curve Cryptography*)

Pada proses enkripsi dan dekripsi uang sesuai dengan blok diagram sesuai dengan algoritma ECC dapat di jelaskan sebagai berikut.

Contoh perhitungan ECC:

Sebagai contoh adalah mengirim pesan ”Bismillah!”, maka pesan saat pesan itu dikirim maka pesan itu juga dienkripsi, adapun langkah perhitungan enkripsi pesan sebagai berikut.

1. Langkah perhitungan pertama kali adalah menentukan persamaan yang dipakai, dan kemudian menenentukan bilangan prima.

- Persamaan yang digunakan $y^2 = x^3+x+1$
- Bilangan Prima (p) = 193

Maka anggota Himpunan di atas dengan batas atas bilangan prima 193 adalah:

(0,1), (0,192), (1,14), (1,179), (3,80), (3,113), (4,29), (4,164), (5,18), (10,25), (10,168), (11,68), (11,125), (12,2), (12,191), (15,66), (15,127), (19,33), (19,160), (20,84), (20,109), (22,40), (22,153), (23,15), (23,178), (25,37), (25,156), (27,5), (27,188), (28,65), (28,128), (29,78), (29,115), (31,78), (31,115), (32,50), (32,143), (35,79), (35,114), (39,84), (39,109), (41,86), (41,107), (45,70), (45,123), (49,82), (49,111), (51,43), (51,150), (52,93), (52,100), (53,30), (53,163), (54,80), (54,113), (55,79), (55,114), (56,64), (56,129), (58,56), (58,137), (61,70), (61,123), (62,6), (62,187), (66,46), 66,147), (67,41), (67,152), (69,38), (69,155), (70,66), (70,127), (71,57), (71,136), (72,32), (72,161), (73,14), (73,179), (74,81), (74,112), (79,2), (79,191), (81,1), (81,192), (82,67), (82,126), (83,28), (83,165), (84,47), (84,146), (85,49), (85,144), (87,70), (87,123), (90,55), (90,138), (91,34), (91,159), (93,42), (93,151), (94,59), (94,134), (95,92), (95,101), (96,182), (100,19), (100,174), (101,13), (102,2), (102,191), (103,79), (103,114), (104,92), (104,101), (108,66), (108,127), (109,63), (109,130), (111,96), (111,97), (112,1), (112,192), (113,23), (113,170), (114,34), (114,159), (115,94), (115,99), (117,105), (119,14), (119,179), (120,81), (120,112), (123,49), (123,144), (127,3), (127,190), (128,23), (128,170), (133,78), (133,115), (134,84), (134,109), (135,95), (135,98), (136,80), (136,113), (138,55), (138,138), (140,35), (140,158), (142,58), (142,135), (145,23), (145,170), (147,85), (147,108), (155,17), (158,55), (158,138), (163,50), (163,143), (165,71), (165,122), (166,39), (166,154), (168,62), (168,131), (169,7), (169,186), (171,77), (171,116), (175,83), (175,110),

(178,49), (178,144), (179,36), (179,157), (180,53), (180,140), (181,34), (181,159), (187,92), (187,101), (188,8), (188,185), (189,60), (189,133), (191,50), (191,143), (192,81), (192,112)

2. Selanjutnya menentukan kunci privat dan menentukan titik awal kurva untuk mencari kunci *publik*.

   - Kunci Privat (k) = 4
   - Titik awal kurva (P) = (133,78)

3. Kunci publik dihitung dengan cara mengalikan kunci privat (k) ke titik awal kurva.

   Kunci *Publik* (kP) = k*P

   = 4*(133,78)

   = [133,78]+[133,78] = [112,192]

   = [112,192]+[112,192]=[163,143]

kP = (163,143)

4. Selanjutnya menentukan titik pengenkripsi yaitu dengan mengalikan kunci privat dengan kunci *publik* .

   Titik kkP = k*kP

   = 4*(163,143)

   = [163,143]+[163,143]=[1,179]

   = [1,179]+[1,179]=[128,170]

   Titik kkP = (128,170)

5. Kemudian ambil titik *absis* (kdP) untuk di xor ke pesan.

   128 → 10000000

6. Setelah titik absis di xor, kemudian konversi pesan per-huruf ke integer sesuai format ASCII, lalu jadikan ke dalam bentuk bilangan *biner* yang selanjutnya di xor dengan *absis* titik kdP. setelah di xor dengan titik kdp maka hasilnya dirangkai kembali menjadi pesan baru yang terenkripsi Sebagai contoh kata yang akan dienkripsi ”Bismillah!” maka ambillah kata perhuruf untuk dienkripsi.

   - Huruf B

➔ B dijadikan bilangan ASCI kemudian diubah menjadi bilangan biner

B = 66

66 = 10000100

➔ Biner dari huruf dixor kan dengan biner titik absis

A = Biner huruf

B = Biner titik absis

Q = Biner hasil Enkripsi

| A | B | Q |
|---|---|---|
| 1 | 1 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |

Dari hasil perhitungan xor maka didapat Q = 00000100

➔ Biner hasil enkripsi kemudian diubah menjadi bilangan ASCII

00000100 = 194

194 = Â

Maka hasil dari enkripsi huruf B adalah Â

Perhitungan selanjutnya dapat dilihat pada tabel 3.1

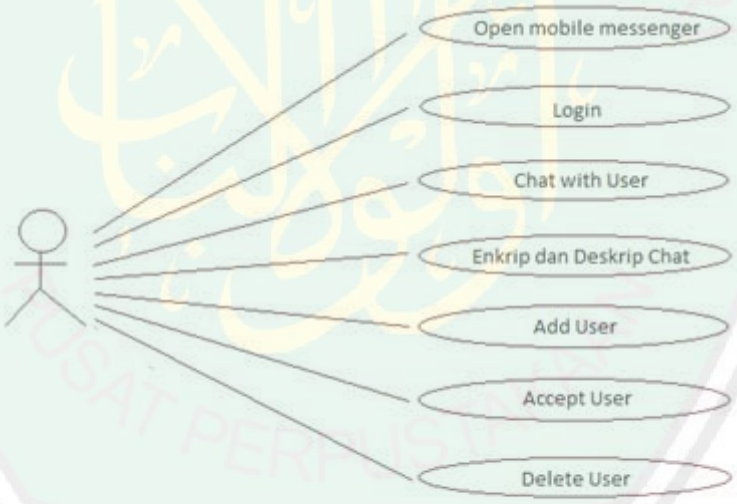
Tabel 3.1 Tabel Hasil Perhitungan Enkripsi ECC

| Huruf | Bilangan Sesuai ASCII | Hasil Bilangan Biner (bb) | Proses | Biner Titik Absis (bta) | Hasil (bb) xor (bta) | Hasil |
|---|---|---|---|---|---|---|
| B | 66 | 10000100 | xor | 10000000 | 194 | Â |
| i | 105 | 11010010 | xor | 10000000 | 233 | é |
| s | 115 | 11100110 | xor | 10000000 | 243 | ó |
| m | 109 | 11011010 | xor | 10000000 | 237 | í |
| l | 105 | 11010010 | xor | 10000000 | 233 | é |
| l | 108 | 11011000 | xor | 10000000 | 236 | ì |
| l | 108 | 11011000 | xor | 10000000 | 236 | ì |
| a | 97 | 11000010 | xor | 10000000 | 225 | á |
| h | 104 | 11010000 | xor | 10000000 | 232 | è |
| ! | 33 | 10000100 | xor | 10000000 | 161 | ¡ |

7. Maka pesan ”Bismillah!” yang terenkripsi adalah Âéóíéììáè¡

### 3.6 *Use Case Diagram*

Gambaran fungsionalitas dari sistem *mobile messenger* sehingga *user* atau pengguna system paham dan mengerti mengenai kegunaan system yang akan dibangun. Gambar diagram *use case* ditunjukkan pada Gambar 3.6




Gambar 3.5 *Use Case diagram*

Pada gambar 3.5 tentang *usecase* adalah menerangkan proses yang dilakukan oleh aplikasi.

1. *Open Mobile Messenger*

Buka aplikasi untuk menjalankan program

2. *Login*

   Proses *login* dilakukan agar dapat masuk sesuai akun yang dimiliki dan agar tak disalahgunakan

3. *Chat with User*

   Komunikasi ini dapat dilakukan oleh user sehingga dapat mengirim dan menerima pesan

4. Enkrip dan Dekrip pesan

   Proses yang dapat dilakukan adalah proses mengenkripsi dan mendekripsi pesan, di aplikasi ini menggunakan metode ecc sebagai metode pengenkripsi dan mendekripsi pesan.

5. *Add User*

   Di proses ini dapat menambah user sesuai contact yang kita tuju

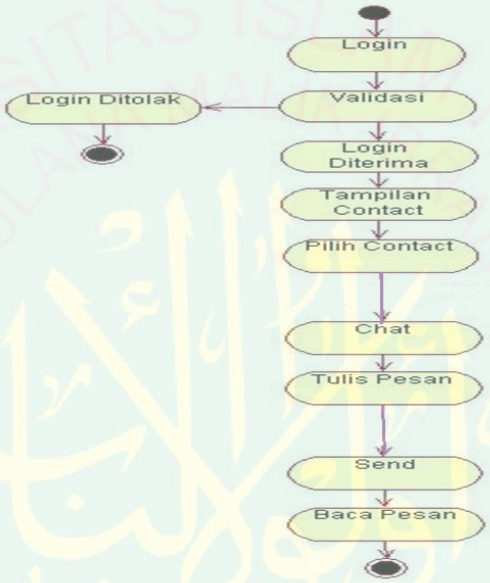
6. *Delete User*

   Menghapus user dapat dilakukan jika diinginkan.

## 3.7 ***Activity Diagram***

### 3.7.1. *Activity Diagram* Pesan

Aktifitas apa saja yang dpat dilakukan oleh *user* dapat dilihat pada gambar 3.6

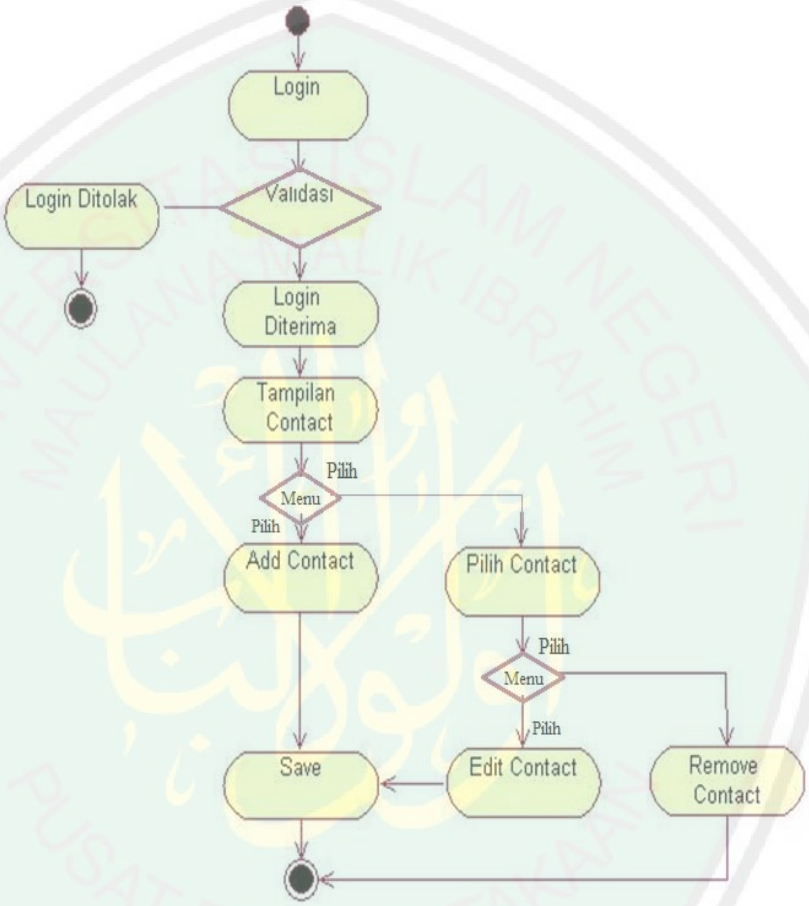



Gambar 3.6 *Activity Diagram* pesan

Aktifitas diagram pesan dimulai dengan *login* yamg kemudian akan divalidasi, jika *login* ditolak, maka harus melakukan *login* kembali. setelah *login* dan validasi dan *login* tersebut diterima akan muncul tampilan *contact* dan kemudian dilanjutkan dengan memilih *contact*. kemudian dilanjutkan dengan memilih menu *chat* dan diteruskan dengan menulis pesan dan dilanjutkan dengan mengirim pesan tersebut dan kemudian pesan tersebut dibaca dan selesai.

### 3.7.2 ***Activity Diagrams Add Contact, Edit Contact, dan Remove Contact***

Sedangkan untuk aktifitas apa saja yang dilakukan oleh *user (client)* dapat dilihat pada gambar 3.7.




Gambar 3.7 *Activity Diagrams Add Contact, Edit Contact,* dan *Remove Contact*

Aktifitas *Diagrams Add Contact*, *Edit Contact*, dan *Remove Contact* pesan dimulai dengan *login* yamg kemudian akan divalidasi, jika *login* ditolak, maka harus melakukan *login* kembali. setelah *login* dan validasi dan *login* tersebut diterima akan muncul tampilan *contact* dan kemudian dilanjutkan dengan memilih *contact*. dalam aktifitas pilih contant ini dapat

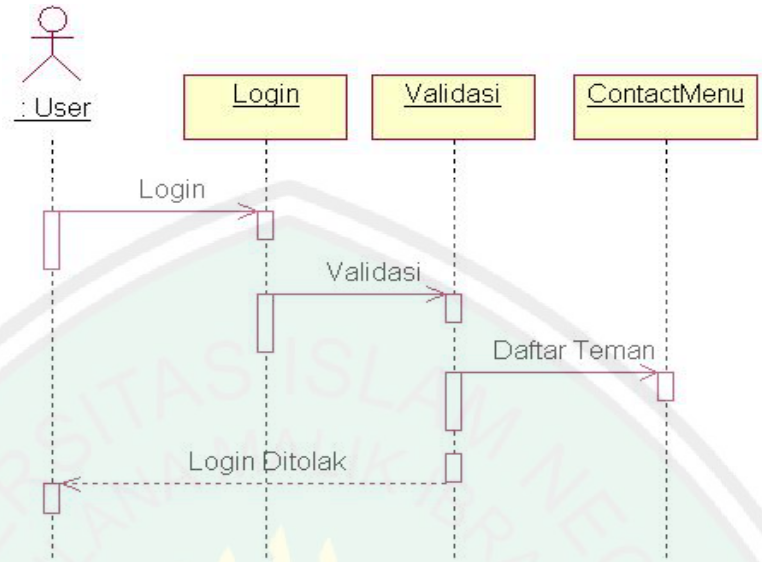
dilakukan beberapa aktifitas edit *contact* yang kemudian dilanjutkan dengan proses *save*, dan untuk *remove contact* akan selesai.

## 3.8 *Sequence Diagram*

Pada diagram dibawah ini di jelaskan urutan kerja aplikasi enkripsi dan *deskripsi* pesan secara urut. Dengan demikian prinsip kerja Pada aplikasi ini dapat kelihatan jelas dan bisa di mengerti. Dari diagram dibawah ini urutan mulai *user*, *login*, menulis pesan, dan pesan dienkripsi sebelum pengiriman pesan. Setelah pesan dienkripsi langkah selanjutnya adalah proses pengiriman pesan. Gambar diagram dibawah ini merupakan urutan proses yang dilakukan oleh *user* saat melakukan pengiriman pesan secara terenkripsi dengan demikian *user* akan melakukan urutan proses dengan benar sehingga tidak terjadi *error*.

### 3.8.1 *Sequence Diagram Login*

Proses Sequence diagram *login* dapat digambarkan pada gambar 3.8

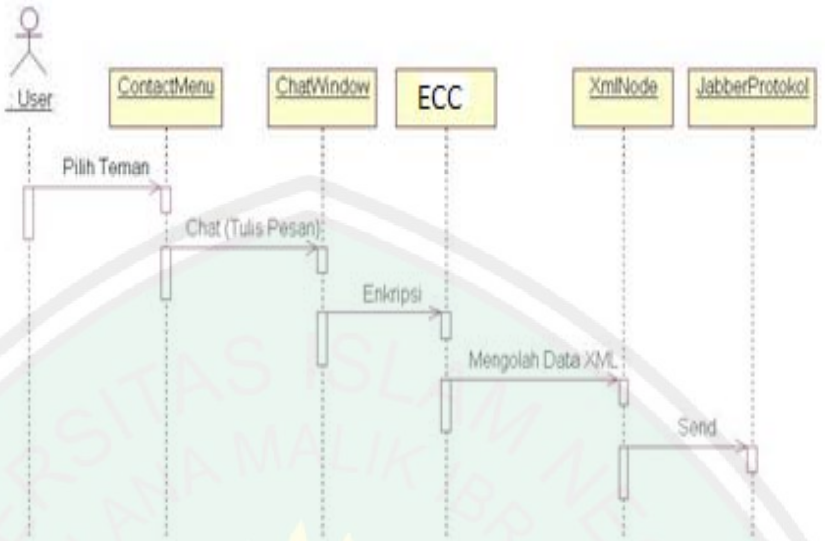



Gambar 3.8 *Diagram Sequence Login*

Saat *user* melakukan *login*, *user* harus memastikan *user name* dan *password* dengan benar. jika salah, maka aktifitas *login* akan ditolak, tetapi jika benar, maka akan muncul daftar *contact* yang ada.

### 3.8.2 *Sequence Diagram* Enkripsi dan Kirim Pesan

*Sequence diagram* enkripsi dan kirim pesan ditunjukkan gambar 3.8

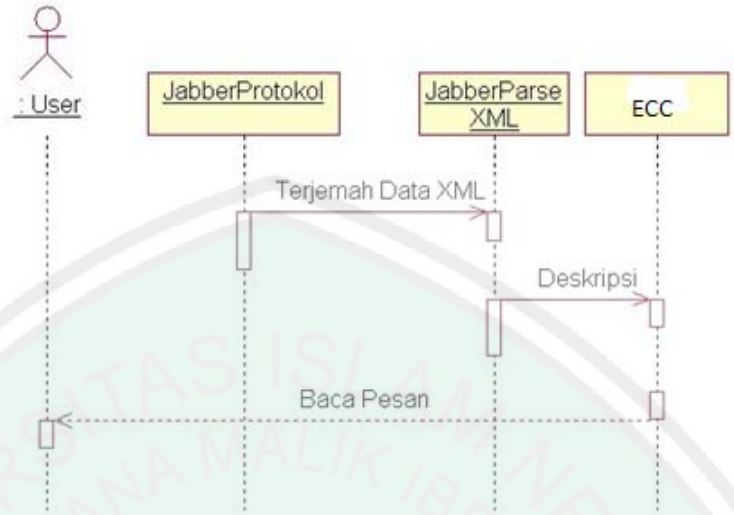



Gambar 3.9 *Diagram Sequence* Enkripsi dan Kirim Pesan

Pada *sequence diagram* enkripsi dan kirim pesan hal yang pertama dilakukan oleh *user* adalah memilih teman dari daftar *contact*, setelah itu menulis pesan dan kemudian pesan dikirimkan, dalam proses pengiriman pesan tersebut pesan dienkripsi kemudian oleh *jabber* pesan dikirimkan ke *contact* tujuan.

### 3.8.3 *Sequence Diagram* Terima dan Dekripsi Pesan

*Sequence diagram* terima dan dekripsi pesan ditunjukkan pada Gambar 3.10




Gambar 3.10 *Diagram Sequence* terima pesan Deskripsi

Saat pesan diterima, prosesnya adalah dari *Jabber* pesan yang diterima didekripsikan oleh Algoritma ECC yang kemudian dijadikan *Plaintext* sehingga pesan tersebut dapat dibaca oleh penerima pesan

# BAB IV

# HASIL DAN PEMBAHASAN

## 4.1 Implementasi

Pada bab ini akan dilakukan pembahasan tentang pengujian dan analisa hasil program yang telah dibuat. Tujuan dari pengujian ini adalah untuk mengetahui apakah aplikasi yang telah dibuat sesusai dengan perancangannya. selain itu juga untuk mengetahui detail jalannya aplikasi serta kesalahan yang ada untuk pengembangan dan perbaikan lebih lanjut. pada proses pengujian ini dibutuhkan beberapa peralatan-peralatan baik berupa perangkat keras dan perangkat lunak.

### 4.1.1 Kebutuhan Perangkat keras

Perangkat keras yang digunakan dalam pembuatan aplikasi *mobile messenger* menggunakan metode algoritma ECC (*Elleptic Curve Cryptography*) ini adalah sebagai berikut.

1. Prosessor AMD Turion X2 2.0 Ghz
2. RAM 1.5 GB
3. Harddisk dengan kapasitas 120 GB
4. LCD 14” dengan resolusi 1280 x 800 *pixels*
5. WIFI *Adapter*
6. Keyboard
7. Mouse

### 4.1.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam pembutan aplikasi *mobile messenger* menggunakan metode algoritma ECC (*Elleptic Curve Cryptography*) sebagai berikut.

1. Sistem Operasi windows 7 Ultimate
2. Netbeans 7.1.2
3. *Java Development Kit* (JDK) 6 u.45
4. Server Jabber 2.1.11

### 4.1.3 Implementasi *Interface*

Pada tahap ini akan dibahas bagaimana implementasi antarmuka. dan dijelaskan proses yang dijalankan.

#### 4.1.3.1 *Interface Jabber Server*

Pada tampilan *add user*, dapat kita buat *user* baru dengan mengisi kolom *user* dan kolom *password*, setelah itu klik tombol *add user* . Contohnya pembutan user dapat dilihat pada gambar 4.1


Gambar 4.1 : *interface jabber server*

Menu yang ada pada *Server Jabber* salah satunya adalah *add user*. Digunakan untuk menambah *user* agar dapat saling berkomunikasi, pada *server* tersebut terdapat juga fasilitas meedit dan menghapus pesan sesuai dengan kebutuhan server.

### 4.1.3.2 *Interface* Pada *Client*

Terdapat beberapa *Interface* pada *client* yang dapat dilihat, tampilan interface tersebut sebagai berikut.

1. *Interface Menu* pada *client*

   Pada tampilan yang tersedia pada emulator terkait dengan menu ditunjukkan pada gambar 4.2


Gambar 4.2 *Interface Menu* pada *client*

Pada tampilan menu di emulator java menunjukkan menu pilihan yang ada yaitu *chat, add contact, remove contact, edit, accounts* dan *about.*

- *Chat yaitu* berfungsi bertukar kirim pesan
- *add contact* berfungsi menambah *contact* pada *client*
- *remove contact* berfungsi menhapus *contact* sesuai yang *client* inginkan
- *edit contact* berfungsi merubah data tentang *contact*
- *about* berisi tentang petunjuk *emulator* itu sendiri

2. *Interface Login* pada *client*

Tampilan *login* pada *client* dapat dilihat pada gambar 4.3


Gambar 4.3 *Interface login*

Pada tampilan *login*, *user* harus memastikan *user name* dan *password* yang sesuai dengan yang didaftarkan pada *server jabber*. Jika tidak sesuai maka *user* tidak akan bisa masuk ke halaman berikutnya.

3. *Interface* Penulisan Pesan Pada *Client*

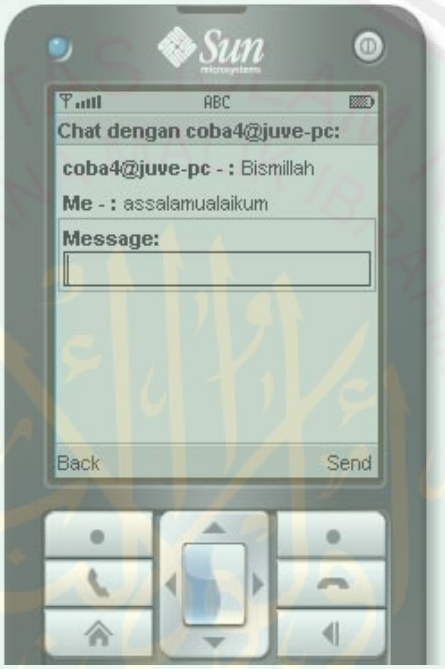
Tampilan penulisan pesan pada *client* dapat dilihat pada gambar 4.4.


Gambar 4.4 *Interface* Penulisan Pesan

*User* dapat menulis pesan dan kemudian pesan tersebut dikirimkan berupa pesan yang telah dienkripsi menggunakan metode algoritma ECC.

4. *Interface* Penerimaan pesan

Pada tampilan saat pesan yang dikirim telah sampai maka tampilan menerima pesan ditunjukkan pada gambar 4.5.

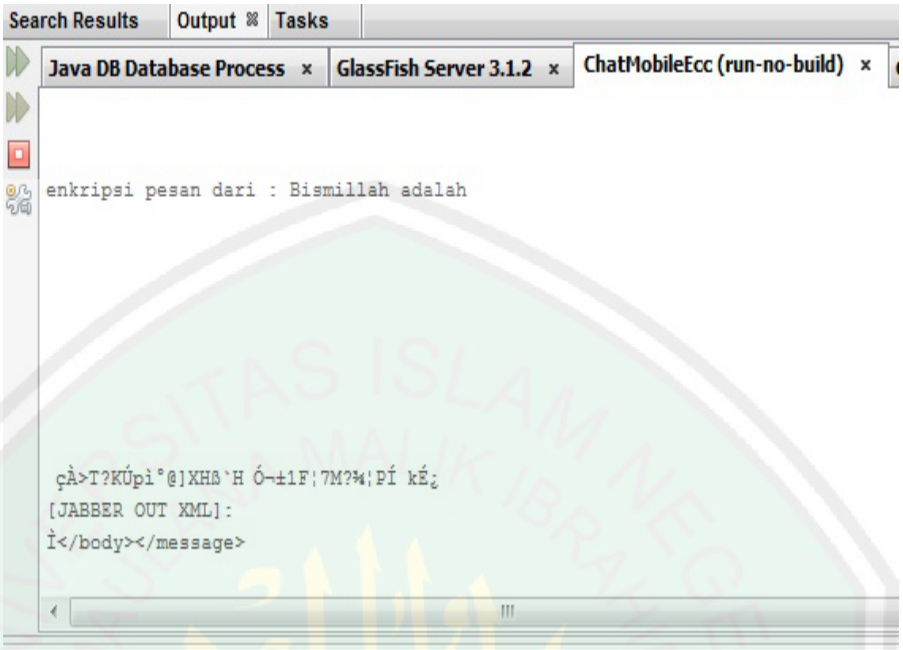



Gambar 4.5 *Interface* Penerimaan pesan

Pesan yang telah diterima oleh *client* kemudian ditampilkan berupa *plaintext* yang telah didekripsi dari pesan yang berupa *chipertext*

### 4.1.3.3 *Interface* Hasil Enkripsi Pesan

Pada bagian ini *interface* hasil dari enkripsi pesan menggunakan algoritma ECC pada pesan yang dikirim oleh *client* dapat dilihat dalam program, hasil enkripsi ditunjukkan padan gambar 4..6




Gambar 4.6 *Interface* Hasil Enkripsi Pesan

Pada gambar 4.6 ditunjukkan pesan yang dikirim oleh *client*, dan diotunjukkan pula hasil enkripsi dari pesan yang telah dikirim.

#### 4.1.3.4 *Interface* Hasil Dekripsi Pesan

*Interface* hasil dekripsi pesan dapat kita lihat dari hasil penerimaan pesan, dari *chipertext* diubah menjadi *plaintext* ditunjukkan pada gambar 4.7


Gambar 4.7 *Interface* Hasil Dekripsi Pesan

Pada gambar 4.7 *client* yang telah mengirimkan pesan akan dikirimkan oleh server berbentuk *chipertext*, sehingga saat pesan itu telah terkirim maka pesan yang diterima seharusnya adalah berupa pesan *chipertext*, tetapi karena ada proses dekirpsi pean maka pesan yang diterima tersebut dirubah dari *chipertext* kedalam bentuk *plaintext* sehingga client yang menerima pesan dapat membaca pesan seperti pesan asli yang dikirim tersebut.

### 4.1.4 Implementasi Algoritma ECC (*Elliptic Curve Cryptography*)

Agar enkripsi pesan menggunakan algoritma ECC dapat berjalan pada aplikasi dibuatlah *source code* sesuai langkah-langkah perhitungan yang telah dilakuklan

Berikut ini adalah *source code* implementasi algoritma ECC ke dalam aplikasi.

1. *Source code* untuk mengenkripsi pesan.

```
public String enkrip(String M, int k){
int[] val = inisialisasi();
int[] P = {val[3],val[4]};
int[] dP = kali(val[0],val[1],val[2],k,P);
int[] kdP = kali(val[0], val[1], val[2], k, dP);
int XkdP = kdP[0];

// Menentukan titik kkP sebagai titik pengenkripsi
// Menentukan titik Awal proses dekripsi

        int[] kP = kali(val[0], val[1], val[2], k, P);
```

```
        String head = (char)kP[0]+"#"+(char)kP[1]+"#";


/*  Header  titik  kP  dijadikan  karakter  kemudian
disertakan dalam pesan untuk dixorkan ke pesan*/

        "+Integer.toBinaryString(XkdP));


 /*Konversi  pesan  ke  integer  sesuai  format  ASCII,
kemudian jadikan biner. Selanjutnya di xor dengan absis
titik kdP, setelah di xor, rangkai lagi menjadi pesan
baru yang terenkripsi*/

        String MM="";

        for(int cd=0; cd<M.length(); cd++)

       {

        "+(int)M.charAt(cd)+"
        Integer.toBinaryString((int)M.charAt(cd));


           int eMx = XkdP ^ M.charAt(cd);

           MM = MM + (char)eMx;}


 /*Angka  desimal  pesan  diatas  kemudian  dikembalikan
menjadi karakter*/

                MM = head + MM

               MM = Base64.encode(MM);

               return MM;


         }
```

2. *Source code* untuk mendenkripsi pesan

```
public String dekrip(String M, int k){
        int[] val = inisialisasi();

/* Pisahkan header dan pesan asli. Kemudian ambil ke
titik kp*/
        String msg = "";
        String MMM = new String(Base64.decode(M));
        String[] dmp = Split(MMM,"#");
        System.out.println("length = "+dmp.length);
        if(dmp.length!=3){
        System.out.println("length NULL= ");
            return M;
        }
        // kp[0]#kp[1]
                int[] kP = {(int)dmp[0].charAt(0),
        (int)dmp[1].charAt(0)};
                //System.out.println("Titik kP :
        ("+kP[0]+","+kP[1]+")");

/* Hitung titik kP dengan k untuk mendapatkan titik-
titik pesan
        int[] dkP = kali(val[0],val[1],val[2],k,kP);

/* Ambil titik absis kkP untuk di-xor-kan ke pesan
terenkripsi.");
        "+Integer.toBinaryString(dkP[0]));

/*Ambil pesan perkarakter, jadikan biner, kemudian
xor-kan ke kkP*/
        M = dmp[2];
        for(int dr=0; dr<M.length(); dr++){
        M.charAt(dr)+" -> "+(int)M.charAt(dr)+" ->
        "+Integer.toBinaryString((int)M.charAt(dr)));
```

```
int eMX = dkP[0] ^ M.charAt(dr);

"+Integer.toBinaryString(dkP[0])+" => "+eMX+"
("+(char)eMX+")");

     msg = msg + (char) eMX;

}

/*Rangkai kembali menjadi pesan asli*/

  return msg;

}
```

## 4.2 Uji Coba Sistem

Pada pengujian sistem ini dilakukan lima pengujian dengan empat pengujian yang berbeda, pengujian meliputi pengujian huruf, *real handset,* hasil enkripsi kata, lama waktu dan jumlah kata yang dapat dienkripsi.

### 4.2.1 Uji Coba Enkripsi Huruf

Pada uji coba enkripsi huruf dapat dijelaskan pada Tabel 4.1

Tabel 4.1 Uji Coba enkripsi huruf

| Input Huruf Abjad | Output Aplikasi | Output Yang Benar | Keterangan |
|---|---|---|---|
| a | á | á | Sesuai |
| b | â | â | Sesuai |
| c | ã | ã | Sesuai |
| d | ä | ä | Sesuai |

| | | | |
|---|---|---|---|
| e | å | å | Sesuai |
| f | æ | æ | Sesuai |
| g | ç | ç | Sesuai |
| h | è | è | Sesuai |
| i | é | é | Sesuai |
| j | ê | ê | Sesuai |
| k | ë | ë | Sesuai |
| l | ì | ì | Sesuai |
| m | í | í | Sesuai |
| n | î | î | Sesuai |
| o | ï | ï | Sesuai |
| p | ð | ð | Sesuai |
| q | ñ | ñ | Sesuai |
| r | ò | ò | Sesuai |
| s | ó | ó | Sesuai |
| t | ô | ô | Sesuai |
| u | õ | õ | Sesuai |
| v | ö | ö | Sesuai |
| w | ÷ | ÷ | Sesuai |
| x | ø | ø | Sesuai |
| y | ù | ù | Sesuai |
| z | ú | ú | Sesuai |

Dari tabel 4.1 dapat diambil kesimpulan bahwa hasil enkripsi huruf dimulai dari huruf a sampai z telah berhasil dienkripsi dan sesuai dengan output yang seharusnya

### 4.2.2 Uji Coba Dekripsi Huruf

Pada uji coba Dekripsi huruf dapat dijelaskan pada Tabel 4.2

Tabel 4.2 Uji Coba Dekripsi huruf

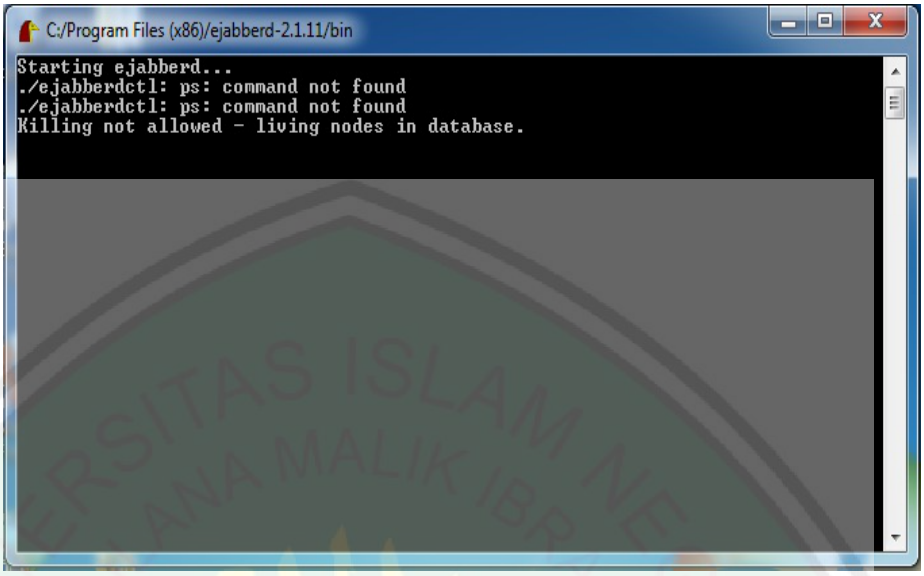
| Enkripsi Huruf | Hasil Dekripsi Huruf | Keterangan |
|---|---|---|
| á | a | Sesuai |
| â | b | Sesuai |
| ã | c | Sesuai |
| ä | d | Sesuai |
| å | e | Sesuai |

| | | |
|---|---|---|
| æ | f | Sesuai |
| ç | g | Sesuai |
| è | h | Sesuai |
| é | i | Sesuai |
| ê | j | Sesuai |
| ë | k | Sesuai |
| ì | l | Sesuai |
| í | m | Sesuai |
| î | n | Sesuai |
| ï | o | Sesuai |
| ð | p | Sesuai |
| ñ | q | Sesuai |
| ò | r | Sesuai |
| ó | s | Sesuai |
| ô | t | Sesuai |
| õ | u | Sesuai |
| ö | v | Sesuai |
| ÷ | w | Sesuai |
| ø | x | Sesuai |
| ù | y | Sesuai |
| ú | z | Sesuai |

Dari tabel 4.2 dapat diambil kesimpulan bahwa huruf yang telah dienkripsi dapat didekripsikan kembali oleh aplikasi menjadi huruf abjad yang sesuai dengan seharusnya.

### 4.2.3 Uji Coba *Real Handset*

Pada uji coba ini dilakukan uji coba pemasangan aplikasi terhadap *real handset*, Uji coba dapat dilihat pada gambar 4.8




Gambar 4.8 Hasil Uji Coba *Real Handset*

Pada uji coba *real handset* dapat dilihat bahwa server tidak dapat berjalan dikarenakan c*rash* dengan koneksi wifi, koneksi wifi dibutuhkan untuk menghubungkan *server* ke *real handset*, tetapi jika wifi dihidupkan maka *server* jabber tidak dapat berjalan, sehingga hasil dari uji coba *real handset* aplikasi tidak dapat dijalankan.

### 4..2.4 Uji Coba Hasil Enkripsi kata

Uji coba hasil enkripsi dapat dilihat pada tabel 4.3.

Tabel 4.3 Tabel Hasil Enkripsi Kata

| Input *Plaintext* | Hasil *Chipertext* | Keterangan |
|---|---|---|
| mencoba | íåîãïâá | Sukses |
| mencoba untuk sukses | íåîãïâá õîôõë óõëóåó | Sukses |
| semoga kita semua dapat menjadi orang yang sukses | óåíïçá ëéôá óåíõá äáðáô íåîêáäé ïòáîç ùáîç óõëóåó | Sukses |
| semoga kita semua selalu dalam lindungan | óåíïçá ëéôá óåíõá óåìáìõ äáìáíîäõîçáîîùá äáî óåíõá áíá | Sukses |

| | | |
|---|---|---|
| nya dan semua amal perbuatan kita menjadi amal dan perbuatan yang berkah | ì ðåòâõáôáî ëéôá íåîêääé áíáì ääî ðåòâõáôáî ùáîç âåòëáè | |

Pada tabel 4.2 didapat kesimpulan bahwa kata-kata berupa *plaintext* telah berhasil dienkripsi menjadi *chipertext* dan hasil enkripsi tersebut adalah sukses yaitu sesuai dengan tabel 4.1 dimana huruf yang ada ditabel hasil enkripsi kata saat dienkripsi sesuai dengan hasil tabel enkripsi huruf .

### 4.2.5 Uji Coba Hasil Dekripsi Kata

Uji coba hasil dekripsi pesan dilakukan dengan mencoba bentuk *chipertext* yang akan dirubah kedalam bentuk *plaintext*. dan hasil dari proses dekripsi pesan ditunjukkan pada tabel 4.4

Tabel 4.4 Tabel Uji Coba Hasil Dekripsi Kata

| Input *Chipertext* | Hasil *Plaintext* | Keterangan |
|---|---|---|
| íåîãïâá | mencoba | Sukses |
| íåîãïâá õîôõë óõë óåó | mencoba untuk sukses | Sukses |
| óåíïçá ëéôá óåíõá ääðáô íåîêääé ïòáîç ùáîç óõëóåó | semoga kita semua dapat menjadi orang yang sukses | Sukses |

Dari hasil uji coba pada tabel hasil dekripsi pesan dapat ditarik kesimpulan bahwa bahwa kata-kata berupa *chipertext* telah berhasil didekripsi menjadi *plaintext* dan hasil dekripsi tersebut adalah sukses yaitu sesuai dengan tabel 4.2 dimana kata yang ada pada tabel hasil dekripsi pesan saat didekripsi sesuai dengan hasil tabel enkripsi huruf sehingga pesan tersebut dapat dibaca oleh penerima pesan.

### 4.2.6 Uji Coba Lama Waktu Terhadap Jumlah Kata Yang Dikirim

Pada uji coba kali ini adalah membandingkan jumlah kata yang dikirim dengan lama waktu yang dibutuhkan dalam proses pengiriman. hasil uji coba dapat dilihat pada tabel 4.5

Tabel 4.5 Uji Coba Lama Waktu Terhadap Jumlah Kata Yang Dikirim

| Jumlah Kata | Lama Waktu (milisecond) |
|---|---|
| 1 | 2 ms |
| 10 | 28 ms |
| 25 | 63 ms |
| 50 | 130 ms |
| 100 | 289 ms |

| | |
|---|---|
| 125 | 395 ms |
| 150 | 473 ms |
| 175 | 534 ms |
| 200 | 682 ms |
| 210 | 778 ms |

Dari tabel 4.5 didapatkanlah hasil bahwa jumlah kata sangat berpengaruh pada lama waktu pengiriman pesan, karena semakin banyak jumlah kata maka semakin lama proses enkripsi dan dekripsi pesan.

Dari tabel 4.5 juga didapatkan kesimpulan bahwa aplikasi ini terbatas pada jumlah karakter yaitu dua ratus sepuluh karakter, sehingga jumlah karakter yang dapat dilakukan pada saat menulis pesan pada aplikasi hanya terbatas sejumlah karakter tersebut.

### 4.3 Tinjauan Sistem Aplikasi Mobile Messenger Menggunakan Metode Algoritma ECC (*Elliptic Curve Cryptography*) Dari Sudut Pandang Islam

Allah SWT memerintahkan kita untuk selalu mencari ilmu dan mengembangkan ilmu, firman Allah SWT dalam Surat Al-Alaq ayat 1 :

خَلَقَ ٱلَّذِى رَبِّكَ بِٱسْمِ ٱقْرَأْ ﴿١﴾

”Bacalah dengan (menyebut) nama Tuhanmu yang Menciptakan”

Ayat tersebut di atas mengandung perintah membaca, membaca berarti berfikir secara teratur atau sistematis dalam mempelajari firman dan ciptaannya, berfikir dengan menkorelasikan antara ayat-ayat Al-Qur'an akan mampu menemukan konsep-konsep sains dan ilmu pengetahuan.

Bahkan perintah yang pertama kali dititahkan oleh Allah kepada nabi Muhammad SAW dan umat Islam sebelum perintah-perintah yang lain adalah mengembangkan sains dan ilmu pengetahuan serta bagaimana cara mendapatkannya. Tentunya ilmu pengetahuan diperoleh diawali dengan cara membaca, karena membaca adalah kunci dari ilmu pengetahuan.

Dari penjelasan tersebut maka penulis di sini berusaha untuk menerapkan apa yang telah diperintahkan oleh Allah SWT, yakni mencari dan mengembangkan ilmu pengetahun. pada prosesnya, penulis mencoba untuk mengembangkan ilmu tentang enkripsi aplikasi *mobile messenger* dan menerapkannya.

### 4.3.1 Enkripsi Pada Pesan Dapat Meningkatakan Sifat Amanah

Terkait tentang konsep kerahasiaan, kerahasiaan menurut konsep agama Islam bahwa kejadian-kejadian apapun yang ada di alam semesta ini merupakan suatu rahasia Allah yang telah dituliskannya di alam *lauhul mahfuzh*. Menyangkut tentang hal tersebut, Allah berfirman dalam surah Al- Hadiid pada ayat 22:

مِّن كِتَٰبٍ فِى إِلَّا أَنفُسِكُمْ فِىٓ وَلَا ٱلْأَرْضِ فِى مُّصِيبَةٍ مِن أَصَابَ مَا
﴿٢٢﴾ يَسِيرٌ ٱللَّهِ عَلَى ذَٰلِكَ إِنَّ ۚ نَّبْرَأَهَآ أَن قَبْلِ

"Tiada suatu bencanapun yang menimpa di bumi dan (tidak pula) pada dirimu sendiri melainkan telah tertulis dalam kitab (Lauhul Mahfuzh) sebelum Kami menciptakannya. Sesungguhnya yang demikian itu adalah mudah bagi Allah."

Oleh karenanya kerahasiaan sebuah pesan patutlah kita jaga, dan supaya kita dapat mengikuti salah satu sifat Rasulullallah SAW yaitu amanah, Secara bahasa, amanah dapat diartikan sesuatu yang dipercayakan atau kepercayaan. Amanah juga berarti titipan (*al-wadî'ah*).

Amanah adalah lawan dari khianat, Amanah terjadi di atas ketaatan ibadah, oleh karena itu sikap amanah merupakan sesuatu yang dipercayakan untuk dijaga, dilindungi, dan dilaksanakan.

Hasil penelitian ini berusaha untuk membuat orang-orang yang memakainya akan melakukan tindakan yang sama sesuai hadist dan firman Allah SWT tersebut, yakni membiasakan untuk membuat kebaikan. Sehingga diharapkan penelitian ini bisa membuat setiap insan yang memakainya mendapatkan manfaat dan kebaikan di dalamnya.

# BAB V

# PENUTUP

## 5.1. Kesimpulan

Setelah melakukan pengujian pada aplikasi ini, dapat disimpulkan bahwa:

- Aplikasi yang dibangun dengan basis J2ME ini dapat mengimplementasikan algoritma ECC dalam proses enkripsi dan dekripsi pesannya, sehingga tingkat keamanan pesan semakin meningkat karena pesan yang dikirimkan berupa *ciphertext* dan hanya bisa dirubah ke *plaintext* jika sampai kepada penerima yang dituju.
- Masih ada kendala dalam pengaplikasian pada penggunaan r*eal handset* dikarenakan *server* jabber mengalami *crash* pada saat wifi komputer dihidupkan, sehingga *server* tidak dapat mengirimkan pesan

## 5.2. Saran

Saran-saran yang diberikan terkait dengan skripsi ini adalah

- *User interface* yang dipakai sebaiknya didesain kembali agar terlihat lebih *user friendly* dan pengembangan aplikasi selanjutnya diharapkan agar dapat mendukung terhadap *real handset* dan juga fitur-fitur seperti *conferencing*, *room chatting*, *smilies* dan pengiriman file
- Perlu pembenahan pada aplikasi *server* karena pada saat pengujian *real handset server* tidak dapat berjalan atau *crash* pada saat wifi diaktifkan.

# DAFTAR PUSTAKA


Ariyus, Doni. 2009, *Keamanan Multimedia.* Yogyakarta, Andi

*Symantec Security Response. (2006). Top Five Instant Messaging Security Risks for* 2006

Ariyus, Doni, 2008, *Pengantar Ilmu Kriptografi*, Yogyakarta, Andi

Schneier, Bruce. *Applied Cryptography*, *2nd edition*, New York: John-Wiley & Sons, 1997

Mardianto, Eko. 2007. *Enkripsi SMS Menggunakan ECC*

Raharjo, Budi, dkk. 2010. *Tuntunan Pemrograman Java untuk Handphone dan Alat Telekomunikasi Mobile lainnya.* Bandung : Informatika Bandung