IMPLEMENTASI ALGORITMA ECDSA UNTUK PENGAMANAN E-MAIL (VERIFIKASI KEASLIAN PESAN)

Pualam Sendi A P[1]; Idris Winarno, S.ST M.Kom[2]; Nur Rosyid M, S.Kom M.Kom[2]
Mahasiswa D4 Lintas Jalur Jurusan Teknik Informatika[1], Dosen Politeknik Elektronika Negeri Surabaya[2]
Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember
Kampus ITS, Keputih Sukolilo, Surabaya 60111
Telp. (+62)-31-5947280 Fax. (+62)-31-5946114
E-mail: pualam_sendi@yahoo.com

Abstrak : Pada proyek akhir ini dibahas masalah implementasi digital signature pada pengiriman email dengan bahasa pemrograman java. Digital signature adalah salah satu layanan keamanan pada kriptografi yang memberikan jaminan kepada pihak penerima pesan (receiver). Jaminan yang diberikan yaitu bahwa pihak pengirim pesan adalah sender, bukan pihak ketiga (eyesdropper) dan pesan yang diterima masih asli. *Elliptic Curve Digital Signature Algorithm* (ECDSA) merupakan salah satu metoda digital signature pada *Elliptic Curve Cryptography* (ECC). ECC adalah public-key cryptography yang menggunakan *Elliptic Curve Discrete Logarithm Problem* (ECDLP) sebagai dasar matematikanya. ECDLP yang digunakan adalah $Q = kP$ dimana Q dan P adalah titik-titik kurva elliptik pada finite field F_{2m} dan k adalah bilangan integer positif. Aplikasi yang dibuat pada proyek akhir ini adalah sebuah mail client yang terintegrasi dengan algoritma tanda tangan digital ECDSA sehingga mampu memberi tanda tangan digital pada pesan yang dikirimkan, melakukan verifikasi tanda tangan digital pada pesan yang diterima, dan memberikan peringatan apabila verifikasi gagal yang berarti email yang diterima sudah tidak asli.

Kata Kunci : email, tanda tangan digital, ECDSA

Abstract : This final project is discussed about implementation of digital signature on email delivery with java programming language. Digital signature is one of the cryptographic security services that provide assurance to the recipient of the message (receiver). Given the assurance that the sender of the message is the sender, not a third party (eyesdropper) and received messages are genuine. Elliptic Curve Digital Signature Algorithm (ECDSA) is one method of digital signatures on Elliptic Curve Cryptography (ECC). ECC is a public-key cryptography using the Elliptic Curve Discrete Logarithm Problem (ECDLP) as the basic math. ECDLP used is Q = kP, where Q and P are the points on the elliptic curve of F2m Finite field and k is positive integers. This final project provide an email client application that integrated with ECDSA algorithm, so it can be able to provide digital signature on sent message, to verify the digital signature on a received message, and give a warning if the verification fails, which means received email was not genuine.

Keyword : email, digital signature, ECDSA

1. Pendahuluan

1.1 Latar Belakang

Seiring dengan perkembangan teknologi informasi, pada tahun 1990 keamanan informasi menjadi bahan pembicaraan bagi banyak orang. Informasi disimpan dalam bentuk elektronik karena medium ini lebih sederhana, ukurannya kompak dan melayani transfer data yang cepat. Kemampuan untuk menyimpan informasi secara efisien mengakibatkan informasi menjadi lebih bernilai. Perkembangan internet yang menjadi jalan raya informasi membuat proses pertukaran informasi menjadi lebih mudah. Banyak kalangan bisnis dan komersial memanfaatkan media ini. Namun, terjadinya revolusi elektronik tersebut, informasi menghadapi masalah yang serius yaitu keamanan informasi pada proses komunikasi.

Proses komunikasi sendiri melibatkan dua pihak yaitu pihak pengirim (sender) dan penerima (receiver). Tentunya yang dikirim adalah informasi atau pesan yang hanya boleh diketahui oleh kedua belah pihak. Namun jika pihak ketiga menyadap dan memodifikasi pesan atau berpura-pura sebagai sender asli tentunya akan sangat merugikan. Masalah ini dikenal dengan nama Man in The middle Attack. Hal ini merupakan salah satu masalah dalam keamanan informasi. Untuk itu dibutuhkan cara agar persoalan tentang keaslian informasi tersebut bisa diatasi.

Authenticity adalah konsep yang dipakai untuk menjaga data yang dikirim agar

tetap utuh dan asli. Authenticity dipakai untuk membuktikan asli atau tidaknya sebuah dokumen atau pesan yang dipakai oleh user (orang yang berhak atas data tersebut). Pembuktian sebuah dokumen atau data ini asli atau tidak juga merupakan dasar untuk pelayanan keamanan pada kepentingan tertentu. Salah satu metode yang ditawarkan dalam konsep authenticity adalah digital signature (tanda tangan digital).

Tanda tangan digital didasarkan pada algoritma kriptografi kunci publik. dimana kunci enkripsi dan kunci deskripsi berbeda. Bagaimanapun tanda tangan digital menggunakan metode yang berlawanan dari yang digunakan oleh algoritma kriprografi kunci publik. Ini berarti bahwa tanda – tanda pengguna dengan menggunakan kunci pribadinya, dan penerima dapat melakukan verifikasi dengan menggunakan kunci publik dari pembuat tanda.

Proyek akhir ini menggunakan algoritma ECDSA(Elliptical Curve Digital Signature) untuk menghasilkan tanda tangan digital, dan diterapkan untuk pengiriman e-mail.

1.2 Tujuan

Proyek akhir ini bertujuan untuk menghasilkan aplikasi mail client yang sudah terintegrasi dengan algoritma tanda tangan digital untuk mengetahui keaslian pesan yang dikirim. Aplikasi tersebut akan memiliki kemampuan:

1.Memberi tanda tangan digital pada pesan yang dikirimkan
2.Melakukan verifikasi tanda tangan digital pada pesan
3.Memberi peringatan apabila verifikasi gagal

1.3 Permasalahan

Permasalahan yang akan diselesaikan pada proyek akhir ini adalah sebagai berikut :

1.Bagaimana cara pembuatan kunci oleh pengirim
2.Bagaimana proses pembuatan tanda tangan digital
3.Bagaimana proses verifikasi tanda tangan digital
4.Bagaimana membuat aplikasi e-mail client dengan menggunakan JAVA
5.Bagaimana mengimplementasikan tanda tangan digital pada e-mail

1.4 Batasan Masalah

Untuk menyederhanakan permasalahan yang akan diselesaikan, diberi batasan masalah sebagai berikut :

1.Pesan yang dikirim berupa plain text
2.Distribusi kunci dilakukan di luar aplikasi yang dibuat
3.Menggunakan database berupa text file untuk menyimpan data

2. Tinjauan Pustaka

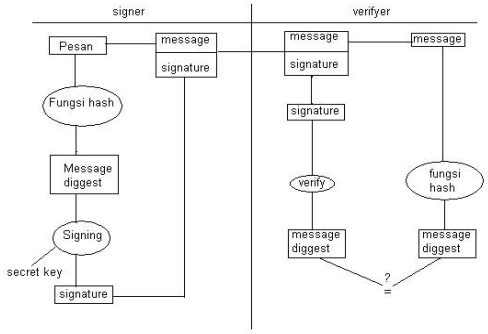
2.1 Keaslian Data Digital

Sebuah pesan, file, dokumen atau kumpulan data yang lainnya dikatakan otentik jika asli dan berasal dari sumber yang terpercaya atau resmi. Otentik sebuah pesan merupakan suatu prosedur yang mengizinkan partisipan untuk memverifikasi bahwa pesan yang diterima otentik atau asli.

2.2 Tanda Tangan Digital

Digital signature disebut juga dengan tanda tangan digital sedangkan handwritten signature adalah tanda tangan tulisan tangan. Walaupun sama-sama mengandung kata signature atau tanda tangan, digital signature tidak sama dengan handwritten signature dalam hal bentuknya namun sama dalam hal fungsinya.

Banyak algoritma digital signature dan semuanya berjenis algoritma public-key dimana untuk menandatangani (signing) pesan menggunakan kunci privat dan untuk verifikasi digital signature menggunakan kunci publik.



Gambar 2.1 skema tanda tangan digital

Secara umum, pada tanda tangan digital terdapat 3 proses utama yaitu : pembangkitan kunci, pemberian tanda tangan digital dan verifikasi keabsahan tanda tangan digital tersebut.

Dalam proyek akhir ini, pemberian tanda tangan akan di lakukan oleh pengirim e-mail, kemudian penerima e-mail akan melakukan verifikasi tanda tangan digital tersebut untuk membuktikan keotentikan pesan.

2.3 ECDSA(Elliptical Curve Digital Signature)

ECDSA adalah salah satu tipe digital signature dari ECC yang memberikan layanan data origin authentication, data integrity dan non-repudiation. ECDSA diperkenalkan

pertama kali pada tahun 1992 oleh Scott Vanstone dan pada tahun 1998 mendapat standar ISO (International Standards Organization) yaitu ISO 14888-3. Tahun 1999 diterima sebagai standar ANSI (American National Standards Institute) ANSI X9.62 dan tahun 2000 sebagai standar IEEE (Institute of Electrical and Electronics Engineers) IEEE 1363-2000 serta standar NIST (National Institute of Standards and Technology) yaitu FIPS 186-2.

Dalam protokol ECDSA, pihak yang akan melakukan tanda tangan digital, mempunyai parameter domain kurva eliptik berupa D = {q,FR,a,b,G,n,h} dan pasangan kunci kunci rahasia dA dan kunci publik QA. Kemudian pihak yang akan melakukan verifikasi terhadap tanda tangan, memiliki salinan dokumen D yang otentik dan kunci publik QA. Proses-proses yang terjadi adalah sebagai berikut.

Key Generation

1. Memilih sebuah bilangan bulat random dA, yang nilainya diantara [1,n-1]
2. Menghitung QA = dA • G = (x1,y1)
3. Kunci rahasia = dA, dan kunci publik = QA

***Signing*(Pemberian tanda tangan)**

1. Memilih sebuah bilangan bulat random k, yang nilainya diantara [1,n-1].
2. Menghitung QA = k • G = (x1,y1) dan r = x1 mod n, jika r = 0, maka kembali ke langkah 1.
3. Menghitung k^{-1} mod n
4. Menghitung e = Hash(m)
5. Menghitung s = k^{-1} {e+dA • r} mod n

Tanda tangan untuk pesan m adalah (r,s).

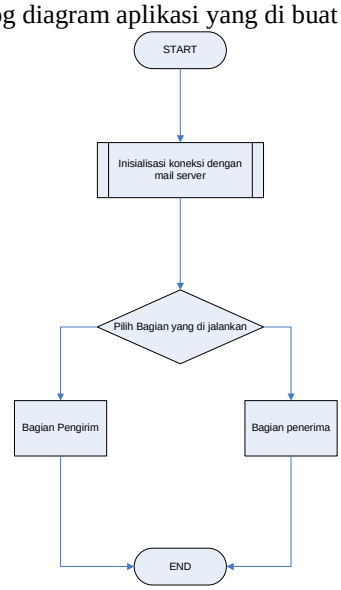
Verifying(Verifikasi tanda tangan digital)

1. Memverifikasi bahwa r dan s adalah bilangan bulat yang antara [1,n-1]
2. Menghitung e = Hash (m)
3. Menghitung w = s^{-1} mod n
4. Menghitung u_1 = ew mod n dan u_2 = rw mod n
5. Menghitung u_1 • G + u_2 • QA = (x_1,y_1)
6. Menghitung v = x_1 mod n
7. Menerima tanda tangan jika dan hanya jika v = r

3. Perancangan Sistem

3.1 Blog diagram Sistem

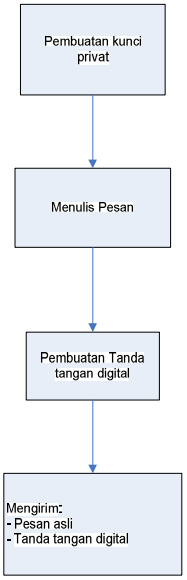
Untuk mengimplementasikan algoritma ECDSA maka terlebih dahulu harus dilakukan perancangan program. Tujuannya adalah untuk penyederhanaan masalah sehingga memudahkan dalam implementasinya. Berikut adalah blog diagram aplikasi yang di buat :



Gambar 3.1 blog diagram sistem secara umum

3.2 Alur Bagian Pengirim

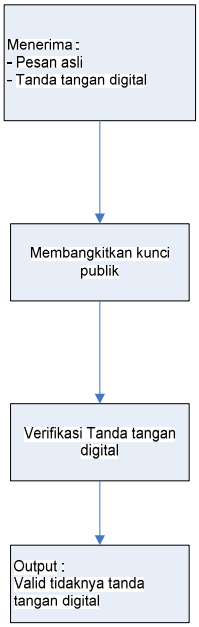
Pada bagian pengirim, yang pertama dilakukan adalah pembuatan kunci privat. Hal ini dilakukan apabila id email yang digunakan belum mempunyai kunci privat. Setelah itu, pengguna dapat menulis email. Ketika mengirimkan email, maka program akan membuatkan tanda tangan digital terlebih dahulu yang kemudian di kirimkan bersama email.



Gambar 3.2 Alur pada bagian pengirim

3.3 Alur Bagian Penerima

Pada bagian penerima, hal utama yang dilakukan adalah verifikasi tanda tangan digital. Verifikasi tanda tangan digital dilakukan secara otomatis ketika email yang akan dibaca di pilih. Program akan menampilkan hasil verifikasi, apakah tanda tangan digitalnya valid yang berarti pesan masih asli, atau apakah tanda tangan digitalnya invalid yang berarti pesan sudah tidak asli.



Gambar 3.3 Alur pada bagian penerima

4. Pengujian dan Analisis

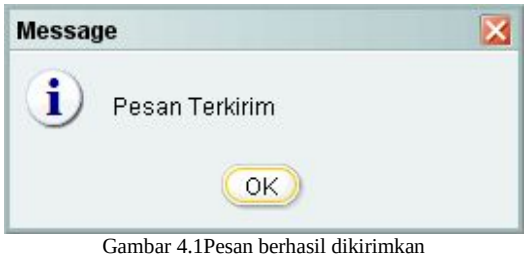
4.1 Skenario Uji Coba

Skenario uji coba yang dilakukan adalah user percobaan2@localhost mengirim pesan pada user percobaan@localhost. Kemudian user percobaan@localhost akan membuka pesan tersebut pada 2 kondisi yaitu saat pesan masih asli, dan saat pesan sudah mengalami perubahan.

4.2 Pengujian 1

Pengujian pertama adalah pengujian dengan isi pesan tidak diubah. Pesan yang dikirim akan sama dengan pesan yang diterima.

Setelah berhasil login, user percobaan2@localhost kemudian melakukan pengiriman pesan ke user percobaan@localhost

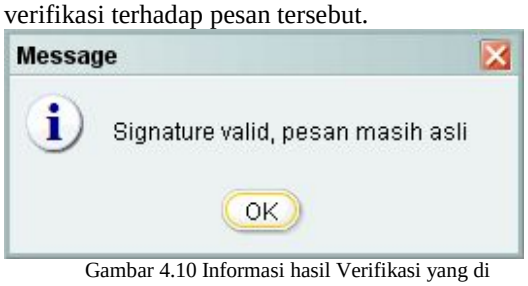


Gambar 4.1Pesan berhasil dikirimkan

Pada jendela output Netbeans akan tampil hasil perhitungan sesuai algoritma ECDSA seperti berikut:

```
560395150366203542734114032468398923
084249375243 dari obyek
buatSignature(kunci privat)
x1 =
840024030525622767069951194587691773
14311626215169
r= x1 mod n =>
215831136203171328758872525849970766
4715821392951
inv(k) mod n =>
387826267790707107333831911466515019
5906436647110
s = inv(k)(e+dr)
=>349219822036810695777597582089722187
871573292439637
215831136203171328758872525849970766
4715821392951 dari obyek
buatSignature(signature)
349219822036810695777597582089722187
1573292439637 dari obyek
buatSignature(signature)
```

Kemudian user percobaan@localhost memilih pesan yang diterima tersebut untuk dibuka. Program akan otomatis melakukan verifikasi terhadap pesan tersebut.



Gambar 4.10 Informasi hasil Verifikasi yang di tampilkan

Karena pesan yang dibuka masih asli, maka informasi yang tampil tampak pada gambar 4.10. Pada jendela output Netbeans akan tampil hasil perhitungan sesuai algoritma ECDSA sebagai berikut :

```
215831136203171328758872525849970766
4715821392951 dari obyek
verifikasiSignature(signature)
349219822036810695777597582089722187
1573292439637 dari obyek
verifikasiSignature(signature)
109b134d461429f7d4aa2b791738ba6b94d4
3296f dari obyek
verifikasiSignature(kunci publik)
58f144c6c3e769feaa37f7932789076c9143
aa13c dari obyek
verifikasiSignature(kunci publik)
w = inv(s) mod n =>
289746883215739261156759527663714054
2814714967009
u1 = ew mod n =>
252098898422106768294090970168418605
3381955644469
u2 = rw mod n =>
141032472179158927987944122188837655
8489577665648
R = u1G + u2Q =>
engine.ECTitikF2m@1c74f37
u = xR mod n =>
215831136203171328758872525849970766
4715821392951
r =
215831136203171328758872525849970766
4715821392951
valid xx= 1
```

4.3 Pengujian 2

Pada pengujian 2 ini diibaratkan ada seseorang yang mampu mengubah pesan yang dikirimkan. Sehingga pesan yang diterima oleh user percobaan@localhost berbeda dari yang dikirimkan oleh user percobaan2@localhost.

Kondisi awal email yang diterima user percobaan@localhost adalah sebagai berikut:

```
Received: from genius ([127.0.0.1])
by localhost
       with SMTP (Code-Crafters
Ability Mail Server 2.70);
       Mon, 18 Jan 2010 22:39:38
+0700
Date: Mon, 18 Jan 2010 22:39:38
+0700 (ICT)
From: percobaan2@localhost
To: percobaan@localhost
Message-ID:
<10040639.0.1263829178187.JavaMail.
Acer@genius>
Subject: pengujian bab 4
MIME-Version: 1.0
Content-Type: text/plain;
charset=us-ascii
Content-Transfer-Encoding: 7bit
signature:
215831136203171328758872525849970766
64715821392951#34921982203681069577
759758208972218715732924396 37

uji coba pesan asli
```

Pada isi email yang ditandai dengan warna merah diatas, akan diubah menjadi “uji coba pesan palsu”, sehingga kondisi email menjadi sebagai berikut :

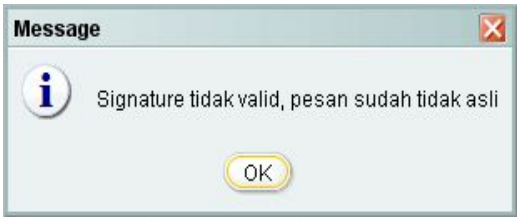
```
Received: from genius ([127.0.0.1])
by localhost
       with SMTP (Code-Crafters
Ability Mail Server 2.70);
       Mon, 18 Jan 2010 22:39:38
+0700
Date: Mon, 18 Jan 2010 22:39:38
+0700 (ICT)
From: percobaan2@localhost
To: percobaan@localhost
Message-ID:
<10040639.0.1263829178187.JavaMail.A
cer@genius>
Subject: pengujian bab 4
MIME-Version: 1.0
Content-Type: text/plain;
charset=us-ascii
Content-Transfer-Encoding: 7bit
signature:
215831136203171328758872525849970766
4715821392951#34921982203681069577 75
975820897221871573292439637

uji coba pesan palsu
```

Aktivitas ini diibaratkan sebagai kegiatan *Man in The Middle Attack*, yaitu melakukan perubahan isi email sehingga isi email yang dikirimkan oleh user percobaan2@localhost tidak akan sama dengan isi email yang diterima oleh user percobaan@localhost.

Pada uji coba ini diumpamakan user percobaan@localhost belum pernah membuka email dari user percobaan2@localhost.

User percobaan@localhost kemudian login pada program yang bertujuan untuk membuka pesan dari user percobaan2@localhost yang ternyata sudah mengalami perubahan.



Gambar 4.2 Informasi hasil verifikasi yang ditampilkan

Karena pesan sudah mengalami perubahan maka informasi verifikasi yang tampil adalah seperti pada gambar 4.13. Pada jendela output Netbeans akan tampil perhitungan sesuai algoritma ECDSA sebagai berikut :

```
215831136203171328758872525849970766
4715821392951 dari obyek
verifikasiSignature(signature)
34921982203681069577759758208972218 7
1573292439637 dari obyek
verifikasiSignature(signature)
109b134d461429f7d4aa2b791738ba6b94d4
3296f dari obyek
verifikasiSignature(kunci publik)
58f144c6c3e769feaa37f7932789076c9143
aa13c dari obyek
verifikasiSignature(kunci publik)
w = inv(s) mod n =>
289746883215739261156759527663714054
2814714967009
u1 = ew mod n =>
231476774062378465089743418446581066
6233656356316
u2 = rw mod n =>
141032472179158927987944122188837655
8489577665648
R = u1G + u2Q =>
engine.ECTitikF2m@c88440
u = xR mod n =>
480813883090653735671904456256818087
3098664516212
r =
215831136203171328758872525849970766
4715821392951
valid xx= 0
```

4.4 Analisis

Pada algoritma ECDSA yang sudah dijelaskan pada bab sebelumnya, telah dijelaskan bahwa signature diterima atau dinyatakan valid jika nilai u = r.

Pada pengujian 1, dari hasil penghitungan sesuai algoritma ECDSA didapat nilai u = r. Sehingga signature dinyatakan valid. Hal ini dapat dilihat dari cuplikan output dari Netbeans berikut ini :

```
u = xR mod n =>
215831136203171328758872525849970766
4715821392951
r =
215831136203171328758872525849970766
4715821392951
valid xx= 1
```

Pada pengujian 2, dari hasil penghitungan sesuai algoritma ECDSA didapat nilai u <> r. Sehingga signature dinyatakan tidak valid. Hal ini dapat dilihat dari cuplikan output dari Netbeans berikut ini :

```
u = xR mod n =>
480813883090653735671904456256818087
3098664516212
r =
215831136203171328758872525849970766
4715821392951
valid xx= 0
```

Pengamatan juga dilakukan terhadap waktu pengiriman dan ukuran email menggunakan tanda tangan digital dibandingkan dengan tanpa tanda tangan digital.

Pada email dengan tanda tangan digital didapat informasi sebagai berikut :

```
Received: from genius ([127.0.0.1])
by localhost
        with SMTP (Code-Crafters
Ability Mail Server 2.70);
        Mon, 18 Jan 2010 22:39:38
+0700
Date: Mon, 18 Jan 2010 22:39:38
+0700 (ICT)
Ukuran : 557
```

Pada email tanpa tanda tangan digital didapat informasi sebagai berikut :

```
Received: from genius ([127.0.0.1])
by localhost
        with SMTP (Code-Crafters
Ability Mail Server 2.70);
        Thu, 21 Jan 2010 12:40:32
+0700
Date: Thu, 21 Jan 2010 12:40:32
+0700 (ICT)
Ukuran : 437
```

Dari data yang didapat, terdapat perbedaan pada ukuran email antara yang menggunakan tanda tangan digital dengan yang tanpa tanda tangan digital. Namun waktu yang dibutuhkan untuk proses pengiriman tidak terdapat perbedaan.

5. Penutup

5.1 Simpulan

Berdasarkan hasil percobaan pada bab sebelumnya, maka dapat disimpulkan bahwa :

- Algoritma ECDSA yang di implementasikan pada email client dalam proyek akhir ini dapat memberikan informasi tentang keaslian pesan yang diterima dengan syarat penerima sudah memiliki kunci publik milik pengirim.
- Penggunaan fungsi trim() pada saat pengambilan data dari luar program sangat penting karena sering terjadi perubahan saat proses pengambilan data dilakukan.
- Penggunaan tanda tangan digital akan mempengaruhi ukuran email

5.2 Saran

Berikut merupakan beberapa saran yang dapat digunakan untuk pengembangan aplikasi kedepannya :

- Menambahkan pilihan domain parameter yang dapat digunakan
- Menambahkan pilihan jenis kurva eliptik yang bisa digunakan
- Algoritma tanda tangan digital hanya dapat melakukan pengecekan terhadap keaslian pesan, namun tidak dapat digunakan untuk menjaga kerahasiaan pesan. Oleh karena itu sebaiknya ditambahkan juga fungsi yang digunakan untuk menjaga kerahasiaan data.

DAFTAR PUSTAKA

[1]. Andy Triwinarko, Elliptic Curve Digital Signature Algorithm (ECDSA), Makalah TA, Departemen Teknik Informatika ITB.

[2]. Abu Bakar Gadi, Perbandingan Sistem Kriptografi Kunci Publik RSA dan ECC, Makalah, Jurusan Teknik Informatika ITB.

[3]. Nana Juhana, Implementasi Elliptic Curves Cryptosystem(ECC) pada Proses Pertukaran Kunci Diffie-Hellman dan Skema Enkripsi ElGamal, Tugas Akhir, Departemen Teknik Elektro Program Pasca Sarjana ITB, Bandung, 2005

[4]. Imam Kholissodin, Penggunaan Kriptosistem Kurva Elliptik untuk Enkripsi dan Dekripsi Data, Skripsi, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Airlangga, Surabaya, 2007

[5]. Dicky Wizanajani R, Perbandingan Algoritma Berbasis Elliptic Curve Cryptography dengan RSA dan DSA pada Tanda Tangan Digital, Makalah, Program studi Teknik Informatika, Sekolah Tinggi Elektro dan Informatika, Institut Teknologi Bandung

[6]. Ade Andri Hendriadi, Desain dan Implementasi Simulasi ECDSA (*Elliptical Curve Digital Signatura Algorithm*) pada keutuhan pesan menggunakan JAVA, Tugas Akhir, Magister Teknologi Informasi, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2006