PERANCANGAN DAN IMPLEMENTASI DSA *(DIGITAL SIGNATURE ALGORITHM)* MENGGUNAKAN BAHASA PEMROGRAMAN JAVA

Nora Herawati[1], R. Rizal Isnanto[2], Adian Fatchurrohim[3]

Abstract: The digital signature is a value of cryptography which depends on message and sender. Using digital signature, the integrity of data can be guaranted, Beside that, it can be used for authetification of message (legality of sender) and for non repudiation purposes. Thereby, this research is performed to make an application of public key algorithm that is used especially for digital signature, that is DSA (Digital Signature Algorithm) algorithm using Java programming language. DSA algorithm is using two keys, those are private key and public key. The function of private key is to make formation of the signature and to verify signature using public key. Large integer that is used in DSA make DSA very difficult to be eavesdropped. The stages of research are requirement analysis of system, designing system, and testing system. Designing system of DSA using object oriented method. Design system produces a class planning, package and user interface to implementation program with the java programming language. The SimDSA application as a result of the research has an ability to generate key, generate digital signature, and verification of digital signature. This application also can show that the message influence which is modified by another person towards validity of the digital signature. This application can give the security service in the source of communication process using data integrity checking and it's authentication of message source.

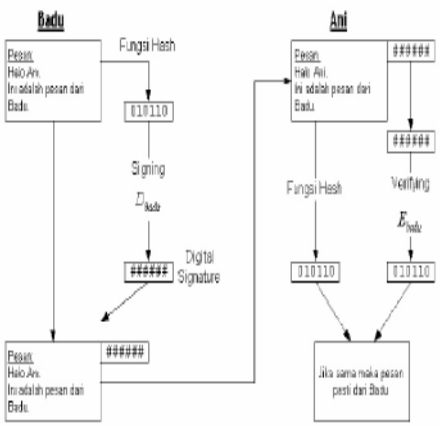
***Kata kunci** : Crypthography, DSA, Java*

Seiring dengan perkembangan teknologi informasi, pada tahun 1990 keamanan informasi menjadi bahan pembicaraan bagi banyak kalangan antaranya pemerintah, bisnis komersial, dan individu. Informasi disimpan dalam bentuk elektronik karena medium ini lebih sederhana, ukurannya kompak, dan melayani transfer data yang cepat. Namun dengan terjadinya revolusi elektronik maka 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 pengirim asli tentunya akan sangat merugikan.

Tujuan pembuatan Tugas Akhir ini adalah dapat menentukan *private key* dan *public key* dari serta dapat menentukan keabsahan tanda tangan digital. Pembatasan masalah untuk Tugas Akhir ini adalah *pertama*, perancangan sistem ini dengan memanfaatkan Java sebagai bahasa pemrogramannya. *Kedua*, pada aplikasi ini format pesan yang dikirim berupa teks. *Ketiga*, *a*lgoritma yang digunakan untuk tanda tangan digital pada perancangan aplikasi ini adalah algoritma DSA (Digital Signature Algorithm)

Tanda-tangan digital

Teknologi tanda-tangan digital memanfaatkan teknologi kunci publik. Sepasang kunci publik-privat dibuat untuk keperluan seseorang. Kunci privat disimpan oleh pemiliknya, dan digunakan untuk membuat tanda-tangan digital. Sedangkan kunci publik dapat diserahkan kepada siapa saja yang ingin memeriksa tanda-tangan digital yang bersangkutan pada suatu dokumen. Proses pembuatan dan pemeriksaan tanda-tangan ini melibatkan sejumlah teknik kriptografi seperti *hashing* (membuat 'sidik – jari' dokumen) dan enkripsi asimetris. Fungsi hash ini menghasilkan nilai hash atau *message digest* dari suatu pesan. Intisari pesan ini berfungsi sebagai sidik – jari (*finger print*) dari suatu pesan sehingga setiap pesan yang berbeda akan menghasilkan *message digest* yang berbeda pula.



Gambar 1. *Tanda-tangan digital* dengan menggunakan fungsi *hash*

Dari Gambar 1 dapat dilihat bahwa tanda-tangan digital adalah message digest yang dienkripsi dengan kunci privat sender. Pada praktiknya para ahli kriptografi mengkombinasikan enkripsi dan tanda-tangan digital sehingga banyak bermunculan cipher-cipher berjenis kriptografi kunci publik. Antaranya ada RSA (Rivest, Shamir, Adleman), DSA (Digital Signature Algorithm) dan ECC (Elliptic Curve Cryptography). DSA inilah yang akan dibahas.

[1] Mahasiswa Teknik Elektro Universitas Diponegoro
[2] Dosen Teknik Elektro Universitas Diponegoro

DSA *(Digital Signature Algorithm)*

DSA menggunakan fungsi hash SHA (*Secure Hash Algorithm*) untuk mengubah pesan menjadi intisari pesan yang berukuran 160 bit. DSA dan algoritma tanda-tangan digital lainnya mempunyai tiga proses utama yaitu:

1. Pembangkitan pasangan kunci (*Key Pair Generation)*
2. Pembangkitan tanda-tangan digital (*Digital Signature Generation)*
3. Verifikasi tanda-tangan digital (*Digital Signature Verification)*

Parameter DSA

DSA dikembangkan dari algoritma ElGamal. DSA mempunyai properti berupa parameter sebagai berikut.

1. p, adalah bilangan prima dengan panjang L bit, yang dalam hal ini 512 ≤ L ≤ 1024 dan L harus kelipatan 64. Parameter p bersifat publik dan dapat digunakan bersama oleh orang di dalam kelompok.
2. q, bilangan prima 160 bit, merupakan faktor dari p – 1. Dengan kata lain, (p – 1) mod q = 0. Parameter q bersifat publik.
3. $g = h^{(p-1)/q} \bmod p$, yang dalam hal ini h < p – 1 sedemikian sehingga $h^{(p-1)/q} \bmod p > 1$. Parameter g bersifat publik.
4. x, adalah bilangan bulat kurang dari q. Parameter x adalah kunci privat.
5. $y = g^x \bmod p$, adalah kunci publik.
6. m, pesan yang akan diberi tanda-tangan.

Prosedur Pembangkitan Sepasang Kunci

1. Pilih bilangan prima p dan q, yang dalam hal ini $(p-1) \bmod q = 0$
2. Hitung g = $h^{(p-1)/q}$ mod p, yang dalam hal ini $1 < h < p-1$ dan $h^{(p-1)/q} \bmod p > 1$
3. Tentukan kunci privat x, yang dalam hal ini x < q.
4. Hitung kunci publik y = g^x mod p.

Jadi, prosedur di atas menghasilkan:

kunci publik dinyatakan sebagai (p, q, g, y);
kunci privat dinyatakan sebagai (p, q, g, x).

Prosedur Pembangkitan Tanda-tangan *(Signing)*

1. Ubah pesan m menjadi intisari pesan dengan fungsi *hash* SHA, H.
2. Tentukan bilangan acak k < q.
3. Tanda-tangan dari pesan m adalah bilangan r dan s. Hitung r dan s sebagai berikut:

$$r = (g^k \bmod p) \bmod q$$
$$s = (k^{-1}(H(m) + x * r)) \bmod q$$

4. Kirim pesan m beserta tanda- tangan r dan s.

Prosedur Verifikasi Keabsahan Tanda-tangan (Verifying)

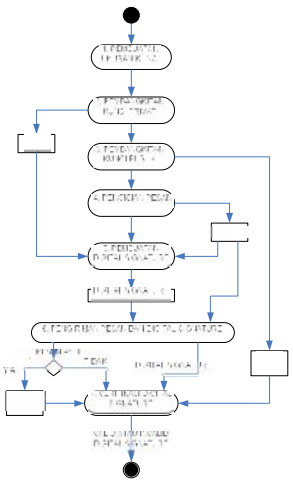
1. Hitung

$$w = s^{-1} \bmod q$$
$$u_1 = (H(m) * w) \bmod q$$
$$u_2 = (r * w) \bmod q$$
$$v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$$

2. Jika v = r, maka tanda-tangan sah, yang berarti bahwa pesan masih asli dan dikirim oleh pengirim yang benar.

DIAGRAM ALIR SISTEM

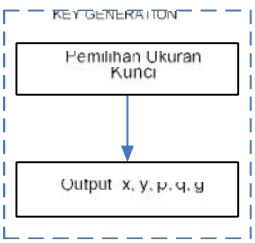
Pada tahap perancangan digunakan diagram alir dari skenario Simulasi DSA. Diagram alir sistem pada Gambar 2 menggambarkan aliran proses dari awal sistem, 3 proses simulasi DSA yang meliputi pembangkitan kunci, pembuatan tanda-tangan digital dan proses verifikasi tanda-tangan digital yang merupakan proses akhir dari sistem.



Gambar 2. Diagram alur dari skenario simulasi DSA

DIAGRAM ALIR PEMBANGKITAN KUNCI DSA

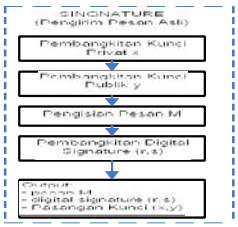
Diagram alir ini memperlihatkan pembangkitan ukuran kunci yang akan digunakan. Semakin panjang ukuran kunci maka akan semakin sulit untuk disadap. Proses dari pembangkitan kunci ini dapat dilihat pada Gambar 3.



Gambar 3. Alur dari bagian pembangkitan kunci

DIAGRAM ALIR *SENDER* (PEMBUATAN TANDA-TANGAN DIGITAL)

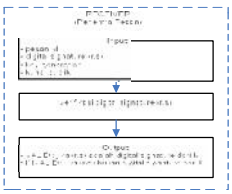
Pada bagian ini dilakukan pembangkitan pasangan kunci yaitu kunci publik dan privat. Kemudian pengisian pesan M untuk kemudian dikirim ke receiver beserta tanda-tangan digitalnya. Proses dari pembangkitan kunci ini dapat dilihat pada Gambar 4.



Gambar 4. Alur bagian pengirim

DIAGRAM ALIR RECEIVER (VERIFIKASI TANDA-TANGAN DIGITAL)

Diagram alir seperti yang ditunjukkan pada Gambar 5, memperlihatkan bagian terakhir dari simulasi DSA yaitu *receiver* sebagai penerima pesan M. Pada bagian ini dilakukan verifikasi terhadap *signature* (r,s) dari pesan M. Namun penerima harus mendapatkan kunci publik yang digunakan oleh pengirim. Setelah itu dilakukan verifikasi terhadap tanda-tangan digital (r,s) dari pesan M.

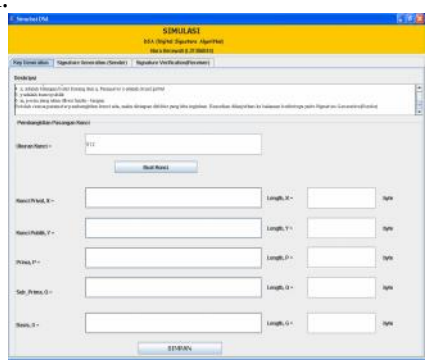


Gambar 5. Alur dari bagian penerima

HASIL

Tampilan *Key Generation*

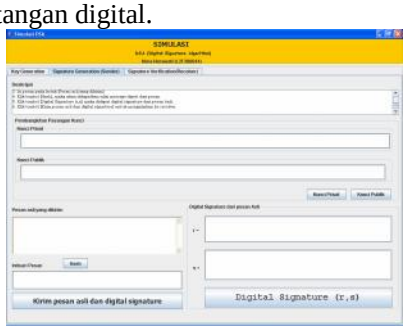
Dalam Menu *Key Generation* ini adalah pemilihan ukuran kunci yang akan digunakan. Tujuan pada bagian ini adalah untuk memilih ukuran kunci yang nantinya digunakan pada bagian berikutnya. Gambar 6 menunjukkan tampilan *Key Generation* dari aplikasi SimDSA.



Gambar 6. Panel *Key Generation.*

Tampilan *Signature Genereation (Sender)*

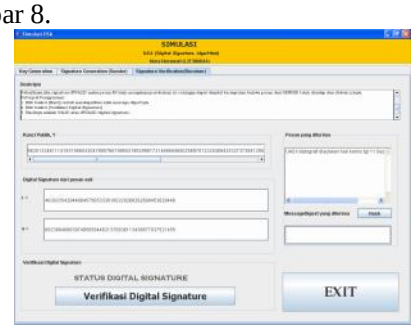
Dalam Panel *sender* menunjukkan pembentukan tanda-tangan digital dengan menggunakan dua buah kunci yaitu kunci publik dan kunci privat. Selain itu masukan yang berupa pesan ini diproses sesuai dengan tipe hash yang dipilih untuk kemudian dihasilkan nilai *message digest*-nya. Pada keluarannya diperkirakan data menjadi teracak dan jumlah bit-nya mengecil sesuai dengan tipe fungsi hash yang digunakan. Gambar 7 menunjukkan tampilan pembangkitan tanda-tangan digital.



Gambar 7. Tampilan pada panel *Signature Generation(Sender)*

Tampilan Verifikasi Tanda-tangan Digital

Tampilan ini merupakan tempat untuk proses verifikasi tanda-tangan digital. Dimana terdapat proses keabsahan pesan yang dikirim oleh pengirim. Adapun tempat untuk memproses tanda-tangan digital tersebut ditunjukkan pada Gambar 8.



Gambar 8. Tampilan untuk tahap verifikasi *digital signature*

PEMBAHASAN

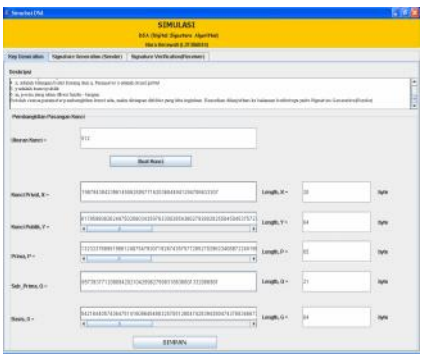
Program ini memperlihatkan bagaimana pengirim memproses pesan sehingga didapat tanda-tangan digital dengan menggunakan algoritma DSA. Kemudian penerima melakukan verifikasi tanda-tangan digital tadi. Juga diperlihatkan pengaruh pada tanda-tangan digital jika pesan asli yang diterima penerima telah dimodifikasi pada saat pengiriman pesan. Untuk contoh kasus, pesan dan ukuran kunci yang digunakan adalah sebagai berikut:

1. Pesan asli:
   ```
   UAS Kriptografi diadakan  hari
   kamis tgl 11 September 2008
   ```

2. Pesan Modifikasi
   `UAS Kriptografi diadakan hari kamis tgl 18 September 2008`
3. `Ukuran Kunci : 512`

Dari hasil pengujian didapatkan:

1. Tampilan pada **Pemilihan parameter DSA.** Dimana nilai parameter-parameter DSA yaitu kunci privat, x, kunci publik, y, prima, p; sub_prima, q; dan basis, g. Tampilan program untuk memilih ukuran kunci yang akan digunakan dapat dilihat pada Gambar 9.



Gambar 9. Tampilan *user interface* untuk pemilihan kunci

Pada panel *key generation*, untuk ukuran kunci 512 maka didapatkan hasil seperti pada Tabel 1.

Tabel 1. Pengujian DSA dengan ukuran kunci 512

Parameter	Nilai	Byte
x	697400905103959132549927686893854335526802766055	20
y	1046205293675265067170877748813035420363977733374690205708914966185801566099086656747312518001142688167011281201971313784023851960888875507124792727315043 5	64
p	1323237689519861240754793071826743575772852702962340887224515603975771302903636871914645218604120423735052178524033704875207146279827300393564623677745922 3	65
q	8573937712080942021042596279903186366013320860981	21
g	5421644057436475141609648488325705128047428394380474376834667300766108262613900542681289080713724597310673074119355136085795982097390670890367185141189796	64

Dari tabel diatas dapat dilihat bahwa:

1. Nilai parameter kunci privat, *x* = 20 byte = 20 x 8 = **160 bit**

Dari hasil pengujian nilai *x* sebesar 48 digit. Hal ini benar, jika nilai kunci harus berada dalam kisaran $0 \le$ nilai kunci $\le 2^{\text{panjang kunci}}$ -1. Hal ini dapat dibuktikan dengan:

$2^{160} = 1{,}46150164...x\ 10^{48} = 48$ digit

dimana

$0 \le$ nilai kunci $\le 2^{\text{panjang kunci}} - 1 \approx 0 \le 2^{160} \le 2^{160} - 1$

Nilai x

$$= 1{,}46150164..x10^{48} \le \left(1{,}46150164..x10^{48}\right) - 1$$

$\therefore$ Hal ini benar karena nilai *x* hasil pengujian 48 digit $\le$ 48 digit (nilai parameter *x*)

2. Nilai parameter kunci publik, *y* = 64byte=64 x 8 = **512 bit**

Berdasarkan hasil pengujian nilai *y* sebesar 154 digit. Hal ini benar, apabila nilai kunci berada dalam kisaran0 $\le$ nilai kunci $\le 2^{\text{panjang kunci}} - 1$. Hal ini dapat dibuktikan dengan:

$2^{512} = 1{,}34078079...x\ 10^{154} = 154$ digit

dimana $0 \le$ nilai kunci $\le 2^{\text{panjang kunci}} - 1$

$\approx 0 \le 2^{512} \le 2^{512} - 1$

Nilai y

$$= 1{,}3408079...x10^{154} \le \left(1{,}34078079...x10^{154}\right) - 1$$

$\therefore$ Hal ini benar karena nilai *y* hasil pengujian 154 digit $\le$ 154 digit (nilai parameter *y*)

3. Nilai parameter *p* = 65 byte = 65 x 8 = **520 bit.**

Dari hasil pengujian nilai *p* sebesar 154 digit. Setiap digit desimal yang tertampilkan, sebenarnya adalah 65 byte atau 520 bit. Hal ini benar,jika nilai kunci harus berada dalam kisaran $0 \le$ nilai kunci $\le 2^{\text{panjang kunci}} - 1$. Hal ini dapat dibuktikan dengan:

$2^{520} = 3{,}43239883...x\ 10^{156} = 156$ digit

dimana $0 \le$ nilai kunci $\le 2^{\text{panjang kunci}} - 1$

$\approx 0 \le 2^{520} \le 2^{520} - 1$

Nilai x

$$= 3{,}43239883\ \ ...x\ 10^{156} \le \left(3{,}43239883\ \ ...x\ 10^{156}\right) - 1$$

Hal ini benar karena nilai *p* hasil pengujian 154 digit $\le$ 156 digit (nilai parameter *p*)

4. Nilai parameter *q* = 21 byte = 21 x 8 = **168 bit**

Berdasarkan hasil pengujian dengan ukuran kunci

sebesar 512 bit, maka dihasilkan nilai q sebesar 48 digit. Hal ini benar, jika nilai kunci harus berada dalam kisaran $0 \leq$ nilai kunci $\leq 2^{\text{panjang kunci}} - 1$. Hal ini dapat dibuktikan dengan:

$2^{168} = 3{,}74144419...x\ 10^{50} = 50$ digit

dimana $0 \leq$ nilai kunci $\leq 2^{\text{panjang kunci}} - 1$

$$\approx 0 \leq 2^{168} \leq 2^{168} - 1$$

$$\text{Nilai x} = 3{,}74144419\ldots x\ 10^{50} \leq \left(3{,}74144419\ldots x\ 10^{50}\right) - 1$$

$\therefore$ Hal ini benar karena nilai *q* hasil pengujian 48 digit $\leq$ 50 digit (nilai parameter *q*)

5. Nilai parameter *g* = 64 byte = 64 x 8 = **512 bit**

Dari hasil pengujian nilai *g* sebesar 154 digit. Sehingga setiap digit desimal yang tertampilkan, sebenarnya adalah 64 byte atau 512 bit. Hal ini dapat dibuktikan dengan:

$2^{512} = 1{,}34078079...x\ 10^{154} = 154$ digit

dimana $0 \leq$ nilai kunci $\leq 2^{\text{panjang kunci}} - 1$

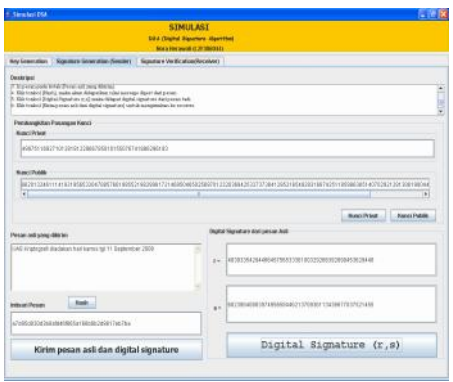
$$\approx 0 \leq 2^{512} \leq 2^{512} - 1$$

$$\text{Nilai x} = 1{,}3408079..x10^{154} \leq \left(1{,}34078079..x10^{154}\right) - 1$$

$\therefore$ Hal ini benar karena nilai g hasil pengujian 154 digit $\leq$ 154 digit (nilai parameter g)

2. Tampilan pada **Signature Generation**.

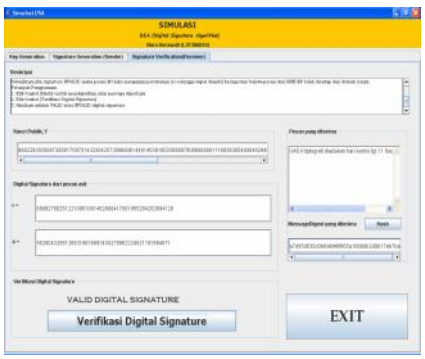
Pada tahap ini sender mengambil pasangan kunci privat dan publik yang telah disimpan. Untuk kemudian sender mengisi pesan yang akan dikirm. Dimana pesan tersebut akan diperoleh nilai *message digest*nya menggunakan fungsi *hash* SHA-1. Ketika pesan dengan sebarang panjang < 2^{64} bit dimasukkan, SHA-1 menghasilkan 160 bit keluaran yang disebut sebagai *message digest*. *Message digest* ini kemudian dimasukkan ke dalam DSA. Setelah didapat kunci privat dan kunci publik maka tahap selanjutnya adalah membangkitkan *digital signature* dari pesan yang akan dikirim. Seperti yang ditunjukkan pada Gambar 10.



Gambar 10. Tampilan untuk tahap pembangkitan pasangan kunci dan *digital signature*

3. Tampilan **Signature Verification.**

Disisi penerima, tanda-tangan diverifikasi untuk dibuktikan keotentikannya dengan cara sebagai berikut: *pertama*, Tanda-tangan digital (r,s) didekripsi dengan menggunakan kunci publik pengirim pesan, menghasilkan *message digest* semula, MD. *Kedua, p*enerima kemudian mengubah pesan M menjadi *message digest* MD' menggunakan fungsi *hash* satu arah yang sama dengan fungsi *hash* yang digunakan oleh pengirim.*Ketiga, J*ika MD = MD', berarti tanda-tangan yang diterima otentik dan berasal dari pengirim yang benar.



Gambar 11. Tampilan untuk tahap verifikasi *digital signature* dengan hasil *valid digital signature*

Jika hasilnya adalah "VALID DIGITAL SIGNATURE" seperti pada Gambar 11 di atas, maka receiver mendapatkan jaminan bahwa pesan dikirim oleh sender. Apabila sender tidak mengakui bahwa pesan tersebut dikirim olehnya maka receiver dapat membuktikannya dengan digital signature dari pesan tersebut.

Sangat mungkin terjadi penyadapan pesan oleh *eavesdropper* sewaktu pesan M ditransmisikan dari sender ke receiver. Pesan M dimodifikasi menjadi M' dan kemudian bersama tanda-tangan digital, pesan asli M dikirimkan kembali ke penerima. Jika terjadi hal seperti itu maka dengan tanda-tangan digital bisa diketahui apakah pesan yang dikirim masih utuh atau sudah dimodifikasi. Sebagai contoh, jika pesan asli dimodifikasi menjadi "`UAS Kriptografi diadakan hari kamis tgl 11 September 2008`" kemudian diverifikasi maka outputnya adalah "INVALID DIGITAL SIGNATURE". Hal ini terjadi karena tanda-tangan digital tadi tidak valid untuk pesan yang dimodifikasi. Hasil tampilan program dengan pesan yang dimodifikasi dapat dilihat pada Gambar 12

Dengan begitu maka tanda-tangan digital pada DSA juga bisa memeriksa keutuhan pesan asli yang dikirim oleh sender. Dari simulasi DSA ini dapat disimpulkan bahwa tanda-tangan digital pada DSA memenuhi tiga layanan keamanan informasi yaitu keaslian data

autentikasi, keutuhan data dan tak-penyangkalan.

Gambar 12. Tampilan untuk tahap verifikasi *digital signature* dengan hasil *valid digital signature*

KESIMPULAN

Setelah dilakukan peose perencanaan, pembuatan dan pengujian sistem maka dapat dibuat kesimpulan, *pertama*, telah dapat dikembangkan aplikasi DSA yang diberi nama SimDSA menggunakan bahasa pemrograman Java dengan kemampuan untuk membangkitkan pasangan kunci, membangkitkan tanda-tangan digital dan verifikasi tanda-tangan digital. *Kedua*, Program SimDSA dapat memperlihatkan pengaruh pesan yang dimodifikasi oleh pihak ketiga terhadap kesahihan tanda-tangan digital. Hal ini membuktikan bahwa tanda-tangan digital pada DSA dapat memberikan layanan keamanan dalam proses komunikasi dengan cara memeriksa keutuhan pesan dan keaslian sumber pesan. *Ketiga*, Tingkat keamanan pada aplikasi DSA sudah dapat diperlihatkan. Hal ini ditunjukkan oleh adanya pembangkitan pasangan kunci yaitu kunci publik dan kunci privat secara semi acak (*pseudorandom)*. Olah sebab itu, penyusup akan mengalami kesulitan dalam menebak ukuran kunci maupun kunci yang digunakan pada proses DSA. *Keempat*, Aplikasi yang dibangun dapat membuktikan bahwa algoritma DSA dapat berjalan dengan baik, dilihat dari proses yang terjadi pada sisi pengirim dan penerima. Pada sisi pengirim *message digest* dienkripsi dengan algoritma kunci publik dan menggunakan kunci privat pengirim. Hasil enkripsi ini yang disebut dengan tanda-tangan digital, kemudian tanda-tangan digital dilekatkan pada pesan. Pada sisi penerima tanda-tangan digital didekripsi menggunakan kunci publik pengirim pesan. Hasil pesan diubah menjadi *message digest* menggunakan fungsi *hash* satu arah yang sama dengan yang digunakan oleh pengirim. Jika *message digest* pengirim = *message digest* penerima, berarti tanda-tangan yang diterima adalah asli dan berasal dari pengirim yang benar.

SARAN

Setelah aplikasi ini dibuat maka ada beberapa hal yang dapat digunakan sebagai bahan penelitian selanjutnya, *pertama*, Perlu dilakukan penelitian lebih lanjut terhadap dua konsep tanda-tangan digital yaitu, tanda-tangan digital menggunakan enkripsi pesan dan tanda-tangan digital menggunakan fungsi *hash*. Selanjutnya hasilnya dibandingkan dengan hasil penelitian ini agar dapat ditentukan metode tanda-tangan digital yang optimal. *Kedua*, Diharapkan dengan adanya aplikasi tanda-tangan digital dapat diterapkan di lingkungan intranet Universitas Diponegoro seperti untuk pengiriman email antar fakultas, ataupun kegiatan akademik yang lainnya, sehingga keamanan pesan yang terkirim terjaga keasliannya.

DAFTAR PUSTAKA

Hariyanto, Bambang, *Esensi-esensi Bahasa Pemrograman Java*, Informatika, Bandung, 2003

Munir Rinaldi, *Kriptografii*, Informatika, Bandung, 2006

Indrajadi dan Martin, *Pemrograman Berorientasi Objek dengan Java*, Elex Media Komputindo, Jakarta, 2003

Hartati, Sri dkk, *Pemrograman GUI Swing Java dengan Netbeans 5*, Andi Yogyakarta, Yogyakarta, 2006

....., *Java™ Cryptography Architecture API Specification and Reference*, http://java.sun.com/product/jdk/1.2/docs/guide/security/crytospec.html, 1999

Dageforde, Mary, *Security in Java 2 SDK 1.2*, http://java.sun.com/docs/books/tutorial.html, Sun Mycrosystem, Inc

Nora Herawati
(L2F306044)
Mahasiswa Jurusan Teknik elektro Ekstensi 2006, Bidang Konsentrasi Teknik Informatika dan Komputer. Universitas Diponegoro
Email:nr_herawati@yahoo.com

Semarang, Desember 2008

Menyetujui

Pembimbing I

R. Rizal Isnanto, S.T, M.M, M.T
NIP. 132 288 515
Tanggal:

Pembimbing II

Adian Fatchurrohim, S. T, M. T
NIP. 132 205 680
Tanggal: