Tanda Tangan Digital Majemuk dengan Kunci Publik Tunggal dengan Algoritma RSA dan El Gamal

Muhamad Fajrin Rasyid [1)]
1) Program Studi Teknik Informatika ITB, Bandung 40132, email: if14055@students.if.itb.ac.id

Abstract *– Integritas dan otentikasi suatu pesan merupakan hal yang tidak dapat dihindari dalam pengiriman dan penerimaan dokumen. Kedua hal tersebut mutlak diperlukan terutama oleh dokumen-dokumen penting seperti perjanjian bisnis, keputusan pihak eksekutif dan legislatif, pernyataan seseorang, dan sebagainya. Hal ini juga berlaku dalam pertukaran data digital. Tanda tangan digital secara umum digunakan untuk otentikasi data digital, seperti pesan yang dikirim melalui saluran komunikasi dan dokumen elektronis yang disimpan dalam komputer. Tanda tangan digital merupakan konsep dalam kriptografi untuk membubuhkan suatu nilai pada pesan yang bergantung pada pengirim pesan. Dengan tanda tangan digital, maka integritas data dapat dijamin. Selain itu, tanda tangan digital juga digunakan untuk membuktikan asal pesan (keabsahan pengirim), dan nirpenyangkalan.*
Tanda tangan digital majemuk merupakan konsep perluasan tanda tangan digital sehingga satu pesan dapat ditandatangani oleh lebih dari satu pihak. Konsep tanda tangan digital majemuk yang akan dibahas pada makalah ini adalah tanda tangan digital majemuk dengan kunci publik tunggal. Pada metode ini, dengan satu kunci publik saja, penerima pesan dapat memeriksa validitas seluruh (sejumlah lebih dari satu) tanda tangan yang ada di dalam pesan tersebut. Konsep ini dapat dilakukan dengan menggunakan algoritma kunci publik RSA dan El Gamal dalam melakukan enkripsi terhadap message digest pesan.

Kata Kunci*: Tanda tangan digital, kunci publik tunggal, RSA, El Gamal*

1. PENDAHULUAN

Tanda tangan digital merupakan suatu nilai kriptografis yang bergantung pada pesan dan pengirim pesan [1]. Pesan yang berbeda akan menghasilkan tanda tangan digital yang berbeda pula. Selain itu, pengirim pesan yang berbeda akan memiliki kunci privat masing-masing yang besar kemungkinan berbeda dengan kunci yang dibuat oleh pengirim pesan lainnya. Tanda tangan digital menjamin integritas data – pesan masih bersifat asli atau utuh dan belum pernah dimanipulasi selama pengiriman. Selain itu, tanda tangan digital juga digunakan untuk membuktikan asal pesan – keabsahan pengirim – dan nirpenyangkalan – pengirim pesan tidak dapat menyangkal melakukan pengiriman dan penerima pesan tidak dapat menyangkal bahwa dirinya telah menerima pesan.

Saat ini kebanyakan aplikasi mendukung tanda tangan digital yang bersifat tunggal, yaitu hanya satu buah tanda tangan yang dapat diverifikasi dengan satu buah kunci publik. Konsep tanda tangan digital majemuk dengan kunci publik tunggal memungkinkan sejumlah n orang ($n > 1$) untuk menandatangani sebuah pesan dan kemudian keseluruhan tanda tangan digital tersebut diverifikasi oleh penerima pesan dengan sebuah kunci publik saja. Konsep ini amat bermanfaat dalam memverifikasi dokumen yang memerlukan keabsahan dari sejumlah pihak yang berbeda, misalnya akta jual beli tanah yang perlu disahkan oleh pihak pembeli dan penjual tanah tersebut.

Tanda tangan digital majemuk dengan kunci publik tunggal bekerja berdasarkan teori bahwa balikan dari suatu bilangan bulat tidak bersifat tunggal. Padahal, konsep balikan bilangan bulat inilah yang digunakan dalam pembangkitan beberapa algoritma kunci publik seperti RSA dan El Gamal yang menjadi dasar penyusunan tanda tangan digital. Dengan demikian, skema untuk penyusunan sejumlah lebih dari satu tanda tangan untuk diverifikasi oleh satu kunci publik saja menjadi mungkin dengan menganalogikan kunci publik dengan suatu bilangan bulat dan sejumlah tanda tangan digital dengan sejumlah balikan bilangan bulat tersebut.

Makalah ini akan membahas penyusunan tanda tangan digital majemuk dengan menggunakan kedua algoritma yang sudah dijelaskan di atas.

2. DASAR TEORI

2.1. Tanda Tangan Digital

Tanda tangan digital diterapkan dengan menggunakan fungsi *hash* dengan sistem kriptografi kunci publik. Dengan cara seperti ini, nirpenyangkalan dapat dihindari [1]. Hal ini disebabkan tanda tangan digital yang dihasilkan bersifat unik untuk tiap pengirim pesan karena bergantung pada kunci privat masing-masing.

2.1.1. Pemberian Tanda Tangan Digital

Proses pemberian tanda tangan digital berlangsung melalui tahapan-tahapan sebagai berikut [5]:

1) Dari pesan *m* yang akan dikirimkan, hitung nilai *message digest* pesan tersebut dengan fungsi *hash* seperti MD5. Misalkan nilai *message digest* pesan

tersebut adalah *MD*.

2) Lakukan enkripsi terhadap *MD* dengan algoritma kriptografi kunci publik yang digunakan (RSA atau El Gamal) dan kunci privat yang dimiliki oleh pengirim pesan (misal *PrivKey*). Hasil dari enkripsi inilah yang menjadi tanda tangan digital (misal *DS*) yang disisipkan pada pesan.
3) Kirimkan *m* dan DS sebagai satu kesatuan pesan baru (misal *m_DS*) yang bertandatangan pengirim pesan.

2.1.2. Verifikasi Tanda Tangan Digital

Proses verifikasi tanda tangan digital dilakukan melalui langkah-langkah sebagai berikut [5]:

1) Ekstraksi kedua elemen pesan *m* dan tanda tangan digital *DS* dari pesan bertanda tangan *m_DS*.
2) Hitung nilai *message digest* dari elemen pesan yang sudah diekstraksi *m* dengan fungsi *hash* sehingga menghasilkan *MD'*.
3) Lakukan dekripsi terhadap elemen tanda tangan yang sudah diekstraksi *DS* dengan kunci publik yang telah diumumkan oleh pengirim pesan (misal *PubKey*). Hasil dari dekripsi ini adalah nilai *message digest* pesan awal yang dikirimkan oleh pengirim pesan – misal *MD*.
4) Bandingkan *MD'* dan *MD*. Apabila *MD' = MD*, maka tanda tangan digital tersebut lolos verifikasi – yang berarti bahwa pesan tersebut otentik dan sah. Sebaliknya, jika *MD'* tidak sama dengan *MD*, berarti tanda tangan digital tersebut tidak lolos verifikasi – pesan tersebut sudah diubah atau elemen kunci yang digunakan tidak benar.

2.2. Teori Balikan Modulo Bilangan Bulat

Dua buah bilangan bulat *a* dan *b* yang memenuhi $a = b \bmod m$ untuk suatu bilangan bulat *m* berarti bahwa *a* dan *b* bersisa sama jika dibagi *m*. Pernyataan $a = b \bmod m$ juga berarti $a = km + b$ untuk suatu bilangan bulat *k* [4]. Sebagai contoh, $9 = 17 \bmod 4$ karena 9 dan 17 sama-sama bersisa 1 jika dibagi dengan 4.

Untuk suatu bilangan bulat *a* dan *b*, bilangan bulat *x* disebut sebagai balikan *a* modulo *b* jika $ax = 1 \bmod b$. Sebagai contoh, balikan dari 8 modulo 5 adalah 2, karena $8.2 = 16 = 1 \bmod 5$ [2]. **Teorema 1** menyatakan bahwa apabila balikan dari *a* modulo *n* adalah *b*, maka demikian pula dengan $kn + b$ untuk setiap bilangan bulat *k* [3]. Hal ini dapat dibuktikan sebagai berikut: $a(kn + b) = akn + ab = akn + ab = 0 + 1 = 1 \pmod n$. Dengan demikian, $kn + b$ adalah balikan dari *a* modulo *n* untuk sembarang *k*. Teorema ini berarti bahwa setiap bilangan bulat *a* memiliki tak hingga balikan modulo *n* yang berulang secara periodik dengan beda *n*.

2.3. Algoritma RSA

Algoritma RSA merupakan algoritma kriptografi kunci publik paling populer yang ditemukan oleh Ron Rivest, Adi Shamir, dan Leonard Adleman [1]. Nama RSA diambil dari inisial nama belakang mereka bertiga.

2.3.1. Prosedur Algoritma RSA

Algoritma RSA diterapkan melalui langkah-langkah sebagai berikut [7]:

1) Tentukan dua buah bilangan prima besar *P* dan *Q* yang bersifat rahasia.
2) Tentukan bilangan asli *E* yang terletak antara 1 dan *PQ* serta relatif prima dengan *(P-1)(Q-1)* – faktor persekutuan keduanya sama dengan 1. Bilangan *E* ini tidak rahasia karena merupakan kunci publik yang digunakan ketika mengenkripsi plaintext.
3) Tentukan balikan dari *E* modulo *(P-1)(Q-1)* – suatu bilangan asli *D* sedemikian hingga $DE = 1 \bmod (P-1)(Q-1)$. Bilangan *D* ini bersifat rahasia karena merupakan kunci privat yang digunakan untuk mendekripsi ciphertext.
4) Bangkitkan fungsi enkripsi $C = T^E \bmod PQ$ dengan *C* adalah ciphertext dan *T* adalah plaintext. Dalam hal ini, *T* harus lebih kecil daripada *PQ*.
5) Bangkitkan fungsi dekripsi untuk menghasilkan plainteks $T = C^D \bmod PQ$.

2.3.2. Karakteristik Algoritma RSA

Algoritma RSA bekerja berdasarkan karakteristik dan fakta sebagai berikut:

1) $t^{o(n)} = 1 \bmod n$, dengan *t* dan *n* adalah dua buah bilangan yang relatif prima dan *o(n)* adalah banyaknya bilangan yang kurang dari *n* yang relatif prima terhadap *n*. Untuk $n = pq$ dengan *p* dan *q* keduanya merupakan bilangan prima, maka $o(n) = (p-1)(q-1)$ [2].
2) Karakteristik 1) digunakan dalam algoritma RSA untuk membangkitkan dua bilangan *d* dan *e* yang saling invers modulo *o(n)*, karena $e.d = 1 \bmod o(n) \leftrightarrow t^{e.d} = t^{k.o(n)+1} = t \bmod n$. Dua buah bilangan *d* dan *e* inilah yang menjadi pasangan kunci publik dan privat dalam algoritma RSA [6].
3) Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan besar menjadi faktor-faktor primanya – sebaliknya, mengalikan faktor-faktor prima menjadi bilangan besar merupakan hal yang cukup mudah. Dengan analogi seperti itu, menghasilkan kunci pada algoritma RSA merupakan hal yang mudah, sebaliknya memecahkannya merupakan hal yang amat sulit [1].

2.4. Algoritma El Gamal

Algoritma El Gamal dibuat oleh Taher El Gamal pada tahun 1984 yang pada awalnya hanya digunakan untuk tanda tangan digital [1].

2.4.1. Prosedur Algoritma El Gamal

Algoritma El Gamal diterapkan melalui langkah-langkah sebagai berikut [8]:

1) Pilih sembarang bilangan prima besar *p* (tidak rahasia).
2) Pilih bilangan acak *k* yang terletak antara 1 dan *p*

– *2* (tidak rahasia).

3) Pilih bilangan acak *g* yang terletak antara 1 dan *p* yang bersifat tidak rahasia karena menjadi kunci publik.
4) Pilih bilangan *x* yang terletak antar 1 dan *p – 2* yang bersifat rahasia karena menjadi kunci privat.
5) Hitung $y = g^x \bmod p$ yang menjadi kunci publik.
6) Proses enkripsi dilakukan sebagai berikut. Untuk setiap plaintext *t*, hitung $a = g^k \bmod p$ dan $b = y^k.t \bmod p$. Gabungan *a* dan *b* inilah yang menjadi ciphertext. Hal ini menyebabkan ciphertext pada El Gamal berukuran dua kali lipat pesan aslinya.
7) Proses dekripsi dilakukan sebagai berikut. Hitung $b / a^x = y^k . t / a^x = g^{xk}.t / g^{xk} = t \bmod p$. Catatlah bahwa $1 / a^x = a^{-x} = a^{p-1-x} \bmod p$ (sebab $a^{p-1} = 1 \bmod p$).

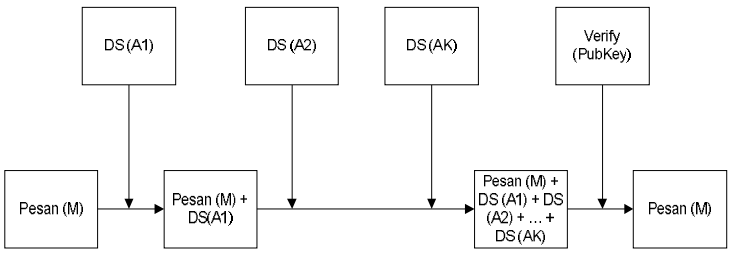
2.4.2. Karakteristik Algoritma El Gamal

Algoritma El Gamal bekerja berdasarkan karakteristik dan fakta sebagai berikut:

1) Keamanan algoritma El Gamal terletak pada *discrete logarithm problem*, yaitu fakta bahwa sulit untuk menemukan bilangan *x* sedemikian hingga $y = g^x \bmod p$ untuk suatu *p* bilangan prima besar, *g* dan *y* sembarang bilangan bulat [9].
2) Tidak seperti algoritma RSA, algoritma El Gamal tidak bekerja secara simetri. Hal ini dapat dilihat pada prosedur mengenkripsi yang berbeda dengan prosedur mendekripsi dan panjang ciphertext yang dua kali panjang plaintext [9].

3. HASIL DAN PEMBAHASAN

Tanda tangan digital majemuk merupakan perluasan tanda tangan digital biasa yang hanya memiliki sepasang kunci privat (digunakan untuk menandatangani pesan) dan kunci publik (digunakan untuk memverifikasi tanda tangan yang terdapat pada pesan). Melalui tanda tangan digital majemuk dengan kunci publik tunggal, seseorang dengan satu kunci publik dapat memeriksa validitas sejumlah lebih dari satu tanda tangan digital (yang berarti ditandatangani oleh sejumlah lebih dari satu pihak) yang terdapat pada pesan. Skema tanda tangan digital majemuk dengan kunci publik tunggal tersebut dapat dilihat pada gambar berikut:



Pada gambar tersebut, dapat dilihat bahwasatu kunci publik (dalam hal ini *PubKey*) dapat memverifikasi seluruh *k* tanda tangan (DS[A1], DS[A2], …, DS[A*k*]). Hal ini dimungkinkan karena algoritma RSA dan El Gamal yang digunakan dalam penerapan tanda tangan digital memanfaatkan konsep kunci publik dan kunci privat berdasarkan teori balikan modulo bilangan bulat yang menyatakan bahwa balikan dari bilangan bulat tidak tunggal melainkan berulang. Fakta inilah yang mendasari bahwa satu kunci publik dapat digunakan oleh sejumlah (tidak tunggal) kunci privat.

3.1. Proses Pemberian Tanda Tangan Digital pada Tanda Tangan Digital Majemuk Kunci Publik Tunggal

Langkah-langkah yang dilakukan untuk memodifikasi proses pemberian tanda tangan digital sehingga menjadi bersifat majemuk adalah sebagai berikut:

1) Pihak yang ingin bersama-sama menandatangani pesan melakukan perjanjian terhadap nilai bersama kunci privat yang ingin digunakan
2) Tiap pengirim pesan menentukan sendiri nilai pribadi kunci privat untuk memperoleh kunci privat masing-masing
3) Ketika tiap pengirim pesan memperoleh pesan, hitung nilai MD dan lakukan enkripsi dengan kunci privat masing-masing, dengan catatan hal itu dilakukan hanya terhadap bagian isi pesan, tidak ditambahkan dengan tanda tangan yang sudah diberikan oleh pihak pengirim pesan sebelumnya
4) Untuk setiap pengirim pesan, setelah nilai tanda tangan digital diperoleh, kirimkan beserta pesan tersebut menjadi pesan baru yang memiliki tanda tangan digital baru dari pengirim pesan tersebut (lihat skema tanda tangan digital majemuk dengan kunci publik tunggal)

3.2. Proses Verifikasi Tanda Tangan Digital pada Tanda Tangan Digital Majemuk Kunci Publik Tunggal

Langkah-langkah yang dilakukan untuk memodifikasi proses verifikasi tanda tangan digital sehingga menjadi bersifat majemuk adalah sebagai berikut:

1) Hitung nilai *message digest* dari elemen pesan yang sudah diekstraksi sehingga menghasilkan *MD'*.
2) Untuk tiap tanda tangan digital yang terdapat pada pesan (DS[A1],DS[A2],…,DS[A*k*]), lakukan dekripsi sehingga menghasilkan *MD[A1],MD[A2],…MD[Ak]*.
3) Bandingkan *MD[Ai]* dan *MD'* untuk $i = 1,2,\ldots,k$. Dengan demikian, validitas seluruh tanda tangan dalam dokumen tersebut sudah diperiksa.

3.3. Karakteristik Tanda Tangan Digital Majemuk Kunci Publik Tunggal dengan Algoritma RSA

Kunci publik *e* dan kunci privat *d* yang digunakan dalam algoritma RSA haruslah memenuhi persamaan $e.d = 1 \bmod (P-1)(Q-1)$ dengan *P* dan *Q* merupakan bilangan prima rahasia yang dipilih. Secara umum, *e*

dan *d* memang memiliki syarat bahwa keduanya harus bernilai lebih kecil dari *PQ*. Namun, hal ini bukanlah syarat mutlak. Secara matematis apabila nilai *d* atau *e* lebih besar daripada *PQ* maka tidak ada prosedur yang dilanggar. Permasalahan yang timbul apabila nilai *d* (atau *e*) lebih besar dari *PQ* adalah sebenarnya terdapat nilai lain yang berpasangan dengan *e* yang merupakan balikan dari *e* modulo *(P-1)(Q-1)* dan berkisar antara 1 dan *PQ*, yaitu $d - k(P-1)(Q-1)$ untuk suatu bilangan bulat *k*. Sebagai akibatnya, dengan *brute force*, nilai *P* dan *Q* dapat ditelusuri. Permasalahan ini dapat diatasi apabila kita memilih nilai *P* dan *Q* yang besar.

Berdasarkan penjelasan tersebut dan **Teorema 1**, dapat dilihat bahwa tiap kunci publik *e* dapat berpasangan dengan sejumlah tak hingga kunci privat $d + k(P-1)(Q-1)$, $k = 0,1,2,\ldots$. Dengan demikian, kunci privat yang digunakan oleh para pengirim pesan haruslah memiliki kesamaan dalam hal nilai *d*, *P*, dan *Q*. Ketiga variabel inilah yang menjadi nilai bersama kunci privat. Di samping itu, kunci privat harus memiliki nilai yang benar-benar dimiliki oleh diri sendiri yaitu *k* (nilai pribadi kunci privat).

3.4. Karakteristik Tanda Tangan Digital Majemuk Kunci Publik Tunggal dengan Algoritma El Gamal

Kunci publik *g* dan *y* dan kunci privat *x* yang digunakan dalam algoritma El Gamal haruslah memenuhi persamaan $y = g^x \bmod p$ dengan *p* merupakan bilangan prima yang dipilih. Seperti pada algoritma RSA, *g*, *y*, dan *x* harus bernilai lebih kecil daripada *p*. Namun, analog dengan algoritma tersebut hal tersebut bukan merupakan syarat mutlak.

Karena $g^{p-1} = 1 \ mod$ p, maka $g^{k(p-1)+x} = g^x \bmod p$. Dengan demikian, untuk suatu nilai kunci publik *g* dan *y*, terdapat sejumlah tak hingga kunci privat $x + k(p-1)$, $k = 0,1,2,\ldots$. Dengan demikian, kunci privat yang digunakan harus memiliki kesamaan dalam hal nilai *x* dan *p* yang menjadi nilai bersama kunci privat. Selain itu, seperti pada RSA, kunci privat memiliki nilai yang benar-benar dimiliki oleh diri sendiri yaitu *k* (nilai pribadi kunci privat).

3.5. Contoh Implementasi Tanda Tangan Digital Majemuk Kunci Publik Tunggal

Tinjau pesan sebagai berikut yang akan ditandatangani oleh tiga orang:

```
Kriptografi
Kriptografi merupakan cabang ilmu
baru meskipun sudah dikenal sejak
ratusan tahun lalu.
```

Pesan tersebut memiliki nilai *message digest* sebesar (dalam heksa) 3016876ED37AAF669EC3FFEF3E9F16D0.

3.5.1. Implementasi dengan Algoritma RSA

Ketiga pengirim pesan melakukan perjanjian untuk menggunakan nilai kunci privat *d* = 131678869943126603234421623066060419456312536058335320382226383069159845762 45, *P* = 333284099561169269844152889400063791187, dan *Q* = 245110699808519556028630632852670780937. Nilai *(P-1)(Q-1)* sebagai basis modulo menjadi 816914988784905051974184433691268221370375967828301198161044389107834536300 96. Dengan demikian, nilai kunci publik *e* = 305350232973451764028255049059837653101.

Selanjutnya, masing-masing pengirim pesan memiliki nilai *k* masing-masing yaitu *k_1* = 683, *k_2* = 941, dan *k_3* = 1283. Dengan demikian, kunci privat orang pertama (privKey1) = *k_1*(P-1)(Q-1)* + *d* = 558084616210033277101602389834202255615423098562788053664375544143720148139318 13,

privKey2 = *k_2*(P-1)(Q-1)* + *d* = 768848683316538780510941973726549456728980098262489762789924996533541458504965 81,

privKey3 = *k_3*(P-1)(Q-1)* + *d* = 104823360948097630828611305004896318843764867925976877256100217760842086991989413.

Dengan kunci publik *e*, ketiga tanda tangan digital yang dihasilkan oleh ketiga orang tersebut dapat diverifikasi sekaligus.

3.5.2. Implementasi dengan Algoritma El Gamal

Ketiga pengirim pesan melakukan perjanjian untuk menggunakan bilangan prima *p* = 181096103647641700716314691561296467673, *g* = 407065770831224180495852259023648 3, dan *x* = 146022131003284771987015118689837 5.

Selanjutnya, masing-masing pengirim pesan menentukan nilai *k* masing-masing yaitu *k_1* = 571, *k_2* = 797, dan *k_3* = 1091. Dengan demikian, kunci privat orang pertama (privKey1) = *k_1*(p-1)* + *x* = 103405876643024721141863408751651469939087,

privKey2 = *k_2*(p-1)* + *x* = 144333596067391745503750529044504471632959,

privKey3 = *k_3*(p-1)* + *x* = 197575850539798405514347048363525633128527.

Dengan kunci publik *g* dan *y* ($g^x \bmod p$), ketiga tanda tangan digital yang dihasilkan oleh ketiga orang tersebut dapat diverifikasi sekaligus.

3.6. Kasus-kasus pada Verifikasi Tanda Tangan Digital Majemuk

Terdapat tiga buah kasus yang mungkin pada saat verifikasi tanda tangan digital majemuk dilakukan. Kasus-kasus ini menentukan kondisi yang terjadi pada pesan maupun kunci-kunci privat yang digunakan oleh masing-masing penandatangan.

3.6.1. Kasus Seluruh Tanda Tangan Digital Valid

Apabila hal ini yang terjadi, berarti tidak ada masalah dengan pesan yang dikirimkan. Pesan – keseluruhan isi dan tanda tangan digital di dalamnya – masih bersifat otentik dan integritasnya masih terjaga. Demikian pula dengan kunci privat yang digunakan oleh masing-masing penanda tangan dan kunci publik

yang digunakan oleh penerima pesan. Keseluruhan kunci tersebut merupakan kunci yang valid yang mmang sudah ditentukan sebelumnya.

3.6.2. Kasus Sebagian Tanda Tangan Digital Valid

Terdapat dua kemungkinan apabila terjadi kasus seperti ini:

1) Apabila DS[A1],DS[A2],…,DS[A*i*] tidak valid, namun DS[A*i+1*],DS[A*i+2*],…,DS[A*k*] bersifat valid, berarti pesan yang diterima oleh penerima pesan sama dengan pesan yang diterima oleh penanda tangan digital ke-*i+1, i+2,…,k*. Dengan demikian, terdapat perubahan isi pesan ketika dikirimkan dari penanda tangan digital ke-*i* ke penanda tangan digital ke-*i+1*.
2) Apabila DS[A1] valid sementara DS[A*i*] tidak valid untuk suatu *i* antara 1 dan *k*, berarti ada yang salah dengan kunci privat yang dimiliki penanda tangan digital ke-*i*.

3.6.3. Kasus Seluruh Tanda Tangan Digital Tidak Valid

Apabila terjadi kasus seperti ini, berarti ada dua kemungkinan yang terjadi. *Pertama*, kunci publik yang digunakan oleh penerima pesan salah. Hal ini disebabkan, apabila pesannya masih otentik sementara kunci publik benar, seharusnya minimal DS[A1] valid. *Kedua*, pesan yang diterima oleh penerima pesan berbeda dengan pesan yang diterima oleh penanda tangan digital ke-*k*. Dengan demikian, *message digest* yang dibaca oleh penerima pesan berbeda dengan *message digest* yang dibaca oleh seluruh penanda tangan.

3.7. Perbandingan Algoritma RSA dan El Gamal pada Tanda Tangan Digital Majemuk

Berikut beberapa faktor yang dibandingkan antara kedua algoritma tersebut dalam kaitannya dengan tanda tangan digital majemuk:

1) Dilihat dari segi keamanan, RSA lebih unggul karena menerapkan keamanan tambahan karena basis modulo yang digunakan (yaitu *(P-1)(Q-1)*) merupakan bilangan yang rahasia. Dengan demikian, apabila salah satu kunci privat sang penandatangan digital diketahui oleh salah seorang penyusup, ia tidak dapat langsung mengetahui kunci privat penandatangan digital yang lain. Hal ini berbeda dengan El Gamal yang menggunakan basis modulo *p* yang diketahui oleh umum. Apabila salah satu kunci privat diketahui, maka dapat dengan mudah menelusuri kunci privat lainnya.
2) Dilihat dari segi variabel yang dipertukarkan, El Gamal lebih unggul karena pihak-pihak penandatangan digital hanya membagi bersama dua variabel (*x* dan *p*) sementara pada RSA, terdapat tiga variabel yang dibagi bersama (*d, P,* dan *Q*). Semakin sedikit variabel yang dipertukarkan, semakin sedikit pula kemungkinan variabel tersebut bocor atau diketahui oleh pihak yang tidak berwenang.
3) Dilihat dari ukuran tanda tangan, RSA unggul karena ukuran tanda tangan yang diperlukan adalah sebesar *message digest* pesan. Sementara itu, pada El Gamal dibutuhkan tanda tangan berukuran dua kali lipat *message digest* pesan. Ukuran tanda tangan yang semakin kecil akan semakin efektif apabila jumlah tanda tangan digital yang terlibat semakin banyak.

4. KESIMPULAN

Dari pembahasan di atas, dapat ditarik kesimpulan sebagai berikut.

1. Tanda tangan digital dapat digunakan untuk menjamin otentikasi data, asal data (keabsahan pengirim), dan nirpenyangkalan.
2. Tanda tangan digital majemuk dengan kunci tunggal memungkinkan sejumlah lebih dari satu orang membubuhkan tanda tangan digital masing-masing pada pesan dan selanjutnya penerima pesan dapat memverifikasi keseluruhan tanda tangan digital tersebut dengan satu buah kunci publik.
3. Tanda tangan digital majemuk kunci tunggal bekerja berdasarkan teori balikan modulo bilangan bulat bahwa balikan tidak bersifat tunggal. Oleh karena itu, hal ini dapat diterapkan dengan algoritma yang berbasiskan teori balikan modulo ini, diantaranya RSA dan El Gamal.
4. Ketika melakukan proses pemberian tanda tangan digital, perlu ditentukan terlebih dahulu nilai bersama kunci privat yang digunakan oleh para penandatangan digital. Selain itu, masing-masing menentukan nilai pribadi kunci privat. Nilai bersama dan nilai pribadi ini yang membentuk kunci privat tiap penandatangan digital.
5. Ketika melakukan verifikasi terhadap tanda tangan digital, ekstraksi isi pesan dan bandingkan message digest dari isi pesan tersebut dan dari tiap tanda tangan untuk memeriksa validitas tiap tanda tangan. Dengan demikian, dapat terjadi kasus seluruh tanda tangan valid, seluruhnya tidak valid, atau sebagian valid sedangkan sebagian tidak valid.
6. RSA dan El Gamal memiliki kelebihan dan kekurangan. RSA unggul dalam hal keamanan tambahan karena kerahasiaan basis modulo yang digunakan dan kebutuhan besar data tanda tangan yang diperlukan. Sementara itu, El Gamal unggul dalam hal sedikitnya kebutuhan variabel yang dipertukarkan.

DAFTAR REFERENSI

[1] R. Munir, "*Diktat Kuliah IF5054 Kriptografi*", Program Studi Teknik Informatika Institut Teknologi Bandung, 2006.

[2] G. Jones, M. Jones, "*Elementary Number Theory*", Springer Verlag London, 1998.
[3] G. Gamble, "*Number Theory*", University of Western Australia, 1997.
[4] R. Munir, "*Diktat Kuliah IF2151 Matematika Diskrit*", Departemen Teknik Informatika Institut Teknologi Bandung, 2005.
[5] http://www.youdzone.com/signature.html, diakses tanggal 11 Januari 2008 pukul 14.00
[6] http://mathworld.wolfram.com/RSA.html, diakses tanggal 11 Januari 2008 pukul 10.00
[7] http://world.std.com/~franl/crypto/rsa-guts.html, diakses tanggal 11 Januari 2008 pukul 10.00
[8] www.informatics.indiana.edu/markus/i400/lecture 7.ppt, diakses tanggal 11 Januari 2008 pukul 10.00
[9] www.ics.uci.edu/~goodrich/teach/ics247/W03/ notes/elgamal.pdf, diakses tanggal 11 Januari 2008 pukul 10.00