BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi

Hasil dari penelitian ini adalah perangkat lunak tandatangan digital sebagai implementasi dari penggabungan metode tandatangan manual dan RSA. Perangkat lunak yang dihasilkan terdiri dari tiga proses inti yaitu proses pembuatan kunci (*create Key*), pembuatan tandatangan digital (*signing*) dan proses verifikasi (*verify*) yang masing-masing tampilan jendela untuk ketiga proses tersebut diberikan pada gambar 4.2, gambar 4.3 dan gambar 4.4. Sebagai masukan tandatangan *offline* yang digunakan pada penelitian ini, diberikan pada tabel 4.1.

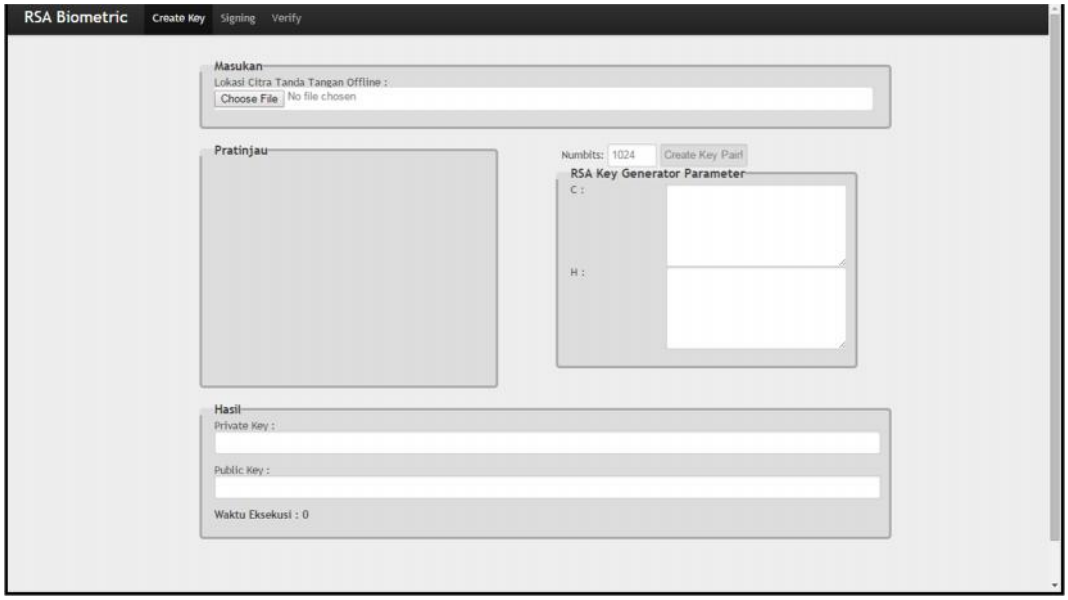


Gambar 4.1 Citra Ttd1.jpg

Tabel 4.1 Citra Tandatangan yang Digunakan dalam Penelitian

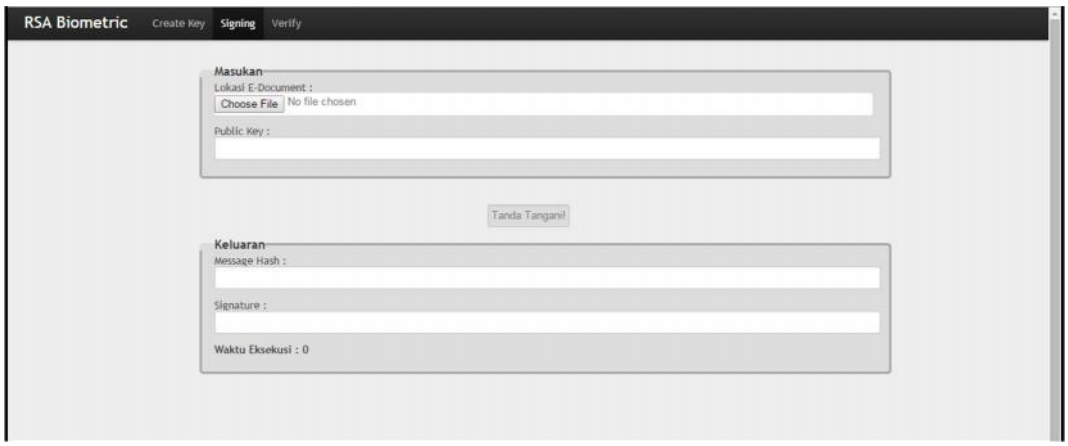
No	Nama file	Keterangan
1	Ttd1.jpg	*Signer* sah

Gambar 4.2 menunjukkan jendela "*Create* RSA *Private* and *Public Key using offline signature* as seed" untuk membuat kunci privat dan kunci publik dengan masukan adalah tandatangan *offline*.



Gambar 4.2 Tampilan Jendela Create Key

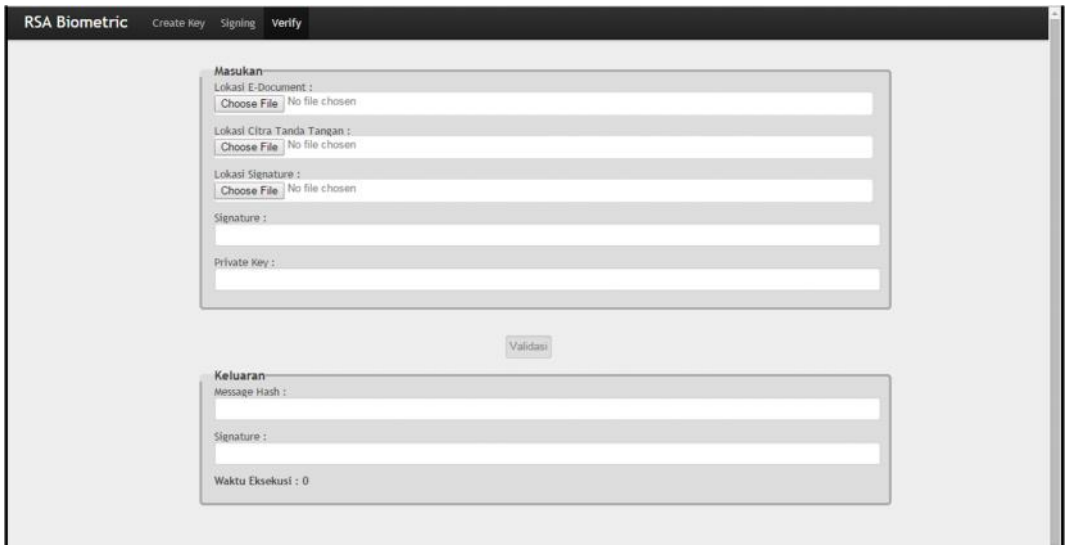
Gambar 4.3 menunjukkan jendela Signing untuk membuat tandatangan digital dari e-dokumen. Sebagai masukan pada proses pembuatan tandatangan digital adalah lokasi e-dokumen dan lokasi kunci privat. Tandatangan digital yang dihasilkan, ditulis pada file *signature* dengan ekstensi sig. Sebagai keluarannya adalah massage hash e-dokumen, *signature* dan lokasi *signature*.



Gambar 4.3 Tampilan Jendela Signing

Gambar 4.4 menunjukkan jendela "verifikasi tandatangan digital" untuk melakukan verifikasi atas keutuhan/ keaslian e-dokumen dan keabsahan signer. Sebagai masukan adalah lokasi e-dokumen, lokasi kunci publik dan lokasi

signature. Sedangkan keluarannya adalah message hash e-dokumen, *signature* dan hasil verifikasi.



Gambar 4.4 Tampilan Jendela Verify

Keunggulan dari perangkat lunak yang dihasilkan pada penelitian ini adalah sebagai konsep baru pada penerapan tandatangan digital yaitu:
Sebagai salah satu solusi dalam aspek manajemen kunci dengan membangkitkan kunci secara dinamis. Hal ini karena dengan masukan tandatangan *offline* yang sama dan panjang kunci juga sama, dapat membangkitkan kunci yang berbeda. Sehingga sifat pemakaian kunci adalah sekali pakai untuk satu tandatangan digital e-dokumen.

4.2 Pengujian Sistem

Pengujian Sistem ini dilakukan dengan hasil simulasi terhadap satu signer. Signer adalah pihak pengirim via e-mail: e- dokumen, tandatangan digital, dan berkas signature, sedangkan verifier adalah pihak penerima e-mail yang melakukan proses verifikasi terhadap file lampiran yang dikirim via e-mail dari signer.

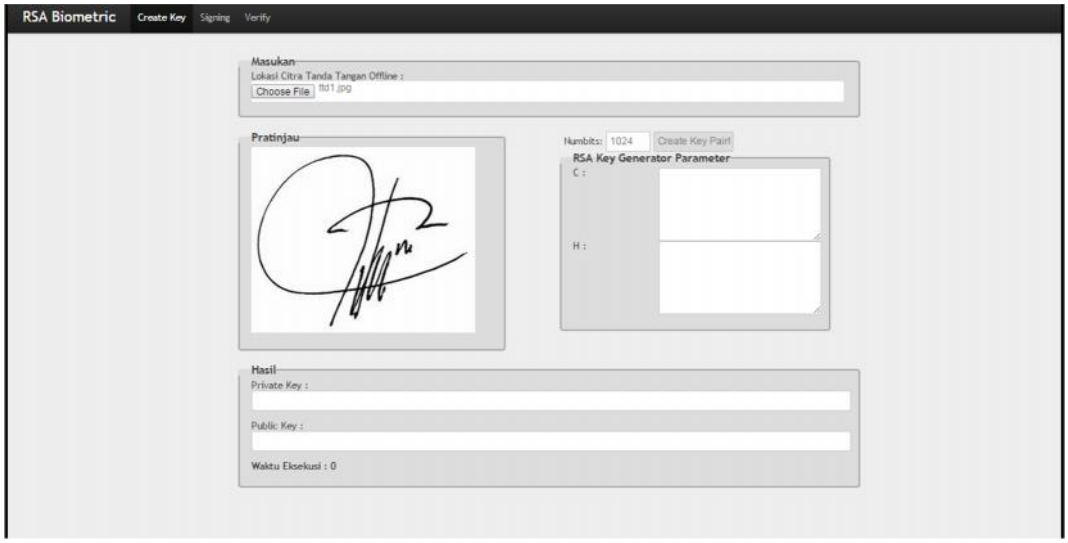
Pengujian dilakukan dengan e-dokumen yang utuh/ otentik dan yang telah berubah, pasangan kunci yang dipakai dari signer yang sah/ sebenarnya dan

signer tidak sah/ bukan sebenarnya juga dengan kunci dan atau *signature* yang dirusak pihak *attacker*.

4.2.1 Simulasi Tandatangan Digital dengan Satu Signer.

Berikut ini disimulasikan tandatangan digital dengan mengambil satu tandatangan *offline* yaitu ttd1.jpg sebagai penandatangan (signer) sah e-dokumen.

4.2.1.1 Proses Pembuatan Kunci Privat dan Publik serta Proses Signing



Gambar 4.4 Menampilan hasil proses membuat kunci dengan masukan ttd1.jpg

Tampilan nilai kunci publik dan kunci privat pada "RSA *Key generator* parameter" adalah pada format hexa.

Sedangkan file yang dihasilkan yaitu kunci publik dan kunci privat adalah dengan menggunakan format mode base 64 [4]. Hal ini untuk menunjukkan adanya perbedaan hasil pada tampilan keluaran nilai dan isi file kunci privat dan kunci publik hanya pada formatnya saja sedangkan nilainya sama. Selanjutnya pasangan kunci tersebut siap digunakan untuk menandatangani/ membuat tandatangan digital e- dokumen.

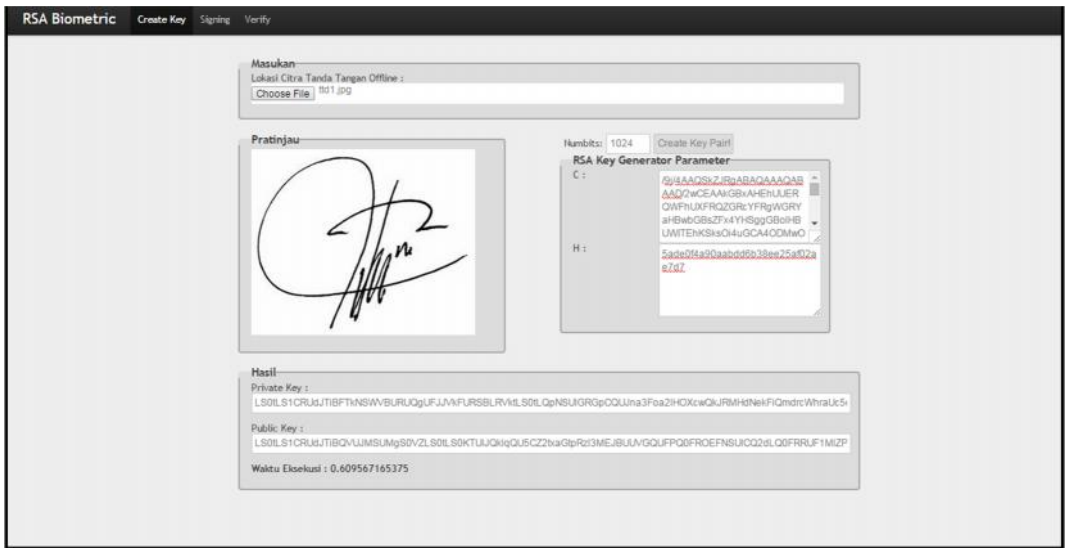
Hasil keluaran pada proses "*create Key* pair" adalah file parameter RSA, kunci privat dan kunci publik yang tersimpan di lokasi yang sama dimana

tandatangan manual yang digunakan untuk membuat kunci disimpan. Contoh tampilan sebagai hasil proses pembuatan kunci dari ttd1.jpg adalah parameter RSA, kunci privat dan publik yang dihasilkan berdasarkan gambar 4.5, masing-masing ditunjukkan pada gambar 4.6.

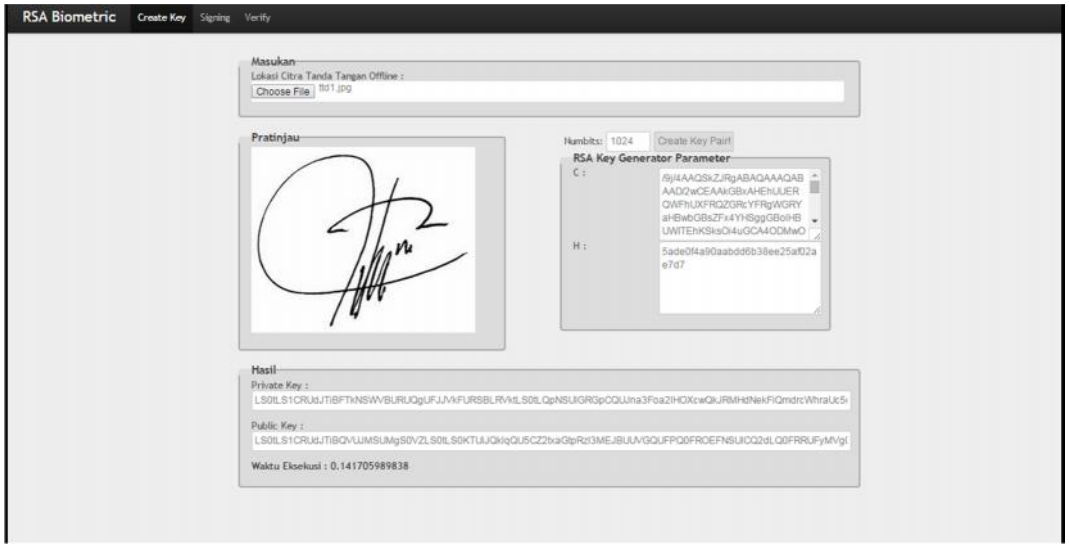
Citra Signature	Public Key	Private Key
	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA014HrnepEQPB2Fo9i0w4 HBM6uKHUlNVd5cRIJLBBAyxZrnT/2u026R9t2s9jEzHlcOdePWrrZVs+8yQ61FE6 EwWa1KJrLF8pKg4Pyyf6lp9hJwbTf6WrScSrMOiID0zpyjLmQf6FCQfSEbj5agKr 3cSkFX7PpjIdgkzYroO/7jdvgE/QA/hAc	-----BEGIN ENCRYPTED PRIVATE KEY----- MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQI/zUe9Hvl6u4CAggA MBQGCCqGSIb3DQMHBAik0aAkyCPfYQSCBMjzrITEIGNd0ObB02x/ioZp8O87xH3e qnuVoi9+kX2e0MlOSgMt19OT1dcaeM7OcJPmTnkLhDSQylhPMhxCwNvnB124u+sHN GVWZJ12DhtgmIHIsaLXlor

Gambar 4.5 Contoh Tampilan Kunci Privat dan Kunci Publik yang Dihasilkan dari Ttd1.jpg dalam Mode Base 64

Pada masukan yang sama yaitu ttd1.jpg dan panjang kunci 1024 dapat menghasilkan kunci publik dan privat yang berbeda. Hal ini terjadi karena hasil/ nilai bilangan prima adalah acak. Sebagai contoh dari masukan ttd1.jpg dan panjang kunci 1024 bit, selain hasil kunci pada gambar 4.6 dan 4.7 dapat menghasilkan kunci yang lain yaitu kunci publik 1 dan 2, yang ditunjukkan masing-masing pada gambar 4.6 dan 4.7.



Gambar 4.6 Kunci Privat dan Publik (1) yang Dihasilkan dari Ttd1.jpg



Gambar 4.7 Kunci Privat dan Publik (2) yang Dihasilkan dari Ttd1.jpg.

Perbedaan pasangan kunci (1) dan (2) yang dihasilkan gambar 4.6 dan 4.7 diberikan pada gambar 4.8

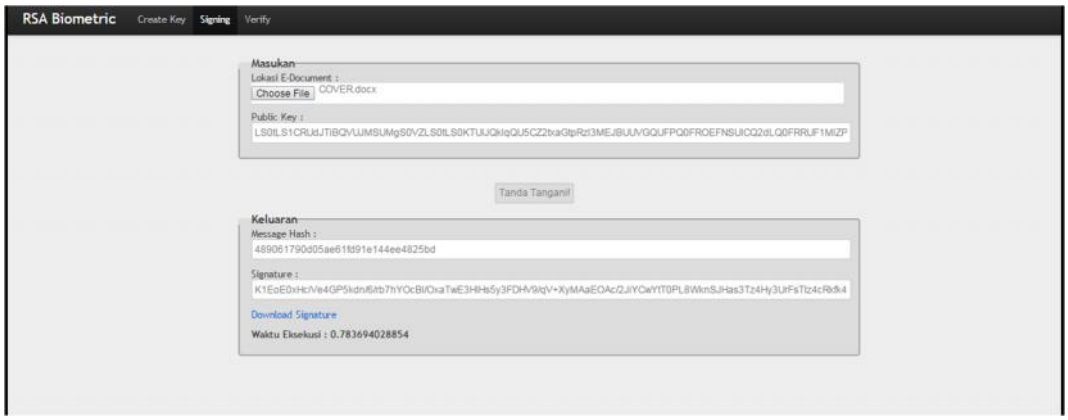
Citra Signature	Public Key	Private Key
	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA51zuSgMUwzr7f7we1YUQ Wkx+PYTotpoAJ25FulcIY0kflVYdFKXyG7U0tFtCAnb3NpTvxcDmozsA6+d/LTMa n6Oc/4dQcMmpwNS4+c1Fb/mB2+BIAE1mFT0fVs4sYosR0R/1dzn6oS8ab2ssaOVm 2LZHIOMO+l6jpsyvCMkrtkT8PbBm3oakM	-----BEGIN ENCRYPTED PRIVATE KEY----- MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQIKlHWZqMbpGICAggA MBQGCCqGSIb3DQMHBAj9S1VNr65jKASCBMy7Cv7lBeU/HopR9EMwarju0ftG62E ZzhlNpmiuts9OXsHpGzOMBsCpzae2AsWYKX8CaVi7o4QS+rQ9KcnYfWciIUl1yOg BGFtPvOzF4GwIRBEYom4vG
	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAr1X4p5pEGkCSlujZN7o+ d+cvgikOmxBX8mNNFi7y+V2hv5P3SBL4eZvAG3gXpcm99QKfsz8v0oceWJbbnkUV tehUA65StgSzCSaF6vHhWXNJKSmxf1v1izViauDlNYYZAAue7Z+U1X57uwvJEyA+ rmvFSow5ZShLVeLStZ9yNxwrECQdupah	-----BEGIN ENCRYPTED PRIVATE KEY----- MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQIrN02ebwBjOECAggA MBQGCCqGSIb3DQMHBAiqXxqJbyrBKgSCBMglVaZgX/R/BgxNAeaCextpU8Ibvi8W NyXs2qqzr+Xo2SpKpWYz0qScHnsCoz+YvMLrC/BqzTHgaT11GqMYjzSP0U4yltE5 f+c0n83bD7G9p8t3kZJCv2

Gambar 4.8 Perbedaan kunci yang dihasilkan berdasarkan gambar 4.6 dan 4.7

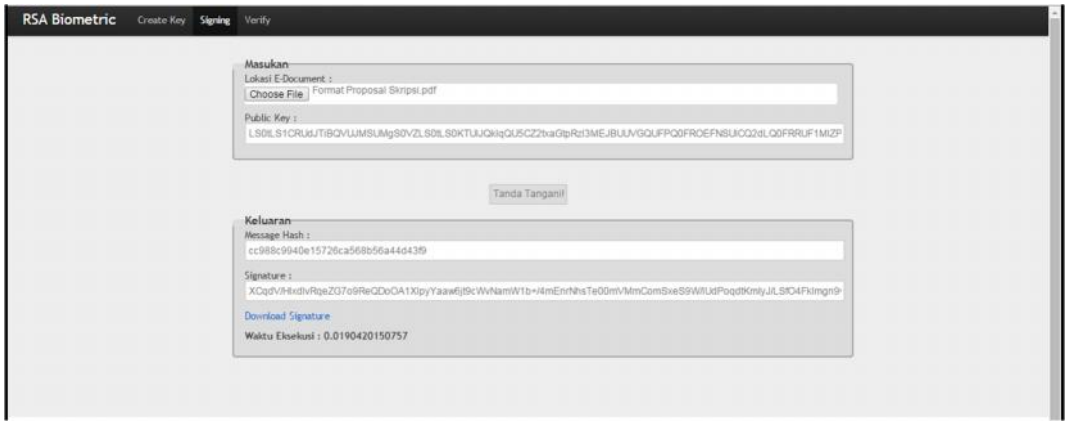
Berdasarkan gambar 4.8 dapat dijelaskan bahwa dari tandatangan yang sama (satu tandatangan *offline*) dengan panjang kunci juga sama, dapat digunakan untuk membuat pasangan kunci yang berbeda sehingga dapat memenuhi kemungkinan tandatangan seseorang yang tidak unik artinya ada beberapa orang yang mempunyai tandatangan yang sama/ hampir sama tetapi kunci yang dihasilkan berbeda. Selain itu hasil pembuatan kunci dari satu masukan yang sama dapat menghasilkan lebih dari satu pasang kunci yang berbeda, artinya kapanpun diperlukan dapat menghasilkan kunci yang berbeda dari satu masukan yang sama. Hal ini menunjukkan konsep baru pada

pembangkitan kunci yang bersifat dinamis, sehingga dapat menjadi salah satu solusi pada aspek manajemen kunci. Jadi sifat penggunaan kunci adalah sekali pakai artinya setiap kali diperlukan dapat membuat pasangan kunci yang baru/berbeda walaupun dari masukan "*Create Pair Key*" yang sama. Selain itu dapat pula menggunakan pasangan kunci yang sama/yang sudah ada beberapa kali untuk membuat tandatangan digital beberapa e-dokumen seperti pada konsep penggunaan kunci. Keputusan menggunakan kunci sekali pakai atau beberapa kali pakai ada di pihak pengguna, namun dari sisi keamanan jauh lebih baik jika menggunakan kunci satu kali pakai karena sangat perlu kunci diubah jauh sebelum ia dapat ditemukan dengan cara exhaustive search [1].

Pada proses signing atau pembuatan tandatangan digital, dalam penelitian ini disebut *signature*, satu pasang kunci dapat digunakan untuk membuat *signature* dari satu e-dokumen (kunci sekali pakai) atau lebih dari satu e-dokumen (kunci beberapa kali pakai). Gambar 4.9 dan 4.10 untuk menunjukkan masukan lokasi kunci publik dan numbit yang sama masing-masing untuk menandatangani e-dokumen COVER.doc dan Format Proposal Skripsi.pdf.



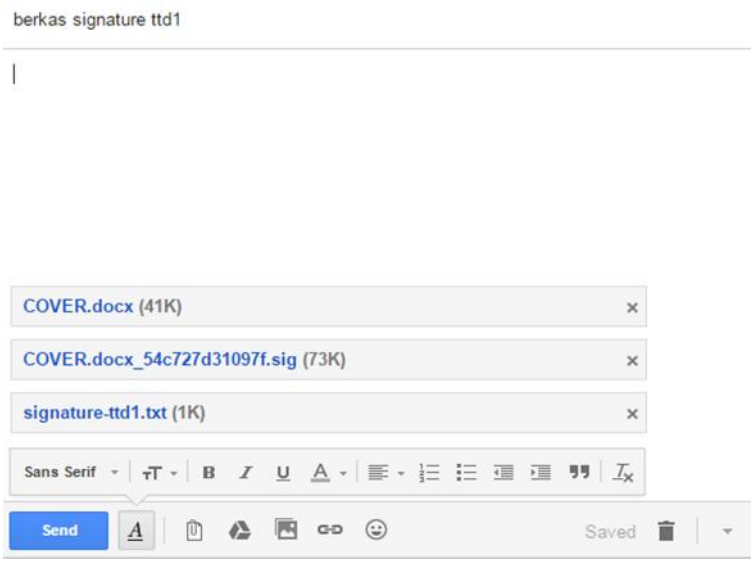
Gambar 4.9 Hasil *Signature* dari COVER.doc dengan Kunci Ttd1



Gambar 4.10 Hasil *Signature* dari Format Proposal Skripsi.pdf dengan Kunci Ttd1

4.2.2 Proses Verifikasi

Pada penelitian ini, disimulasikan proses verifikasi berdasarkan file lampiran yang dikirim via e-mail. Diasumsikan seluruh file lampiran yang dikirim signer diterima oleh verifier tanpa ada satu file pun yang hilang (dicuri) dalam perjalanan transmisi. File yang dilampirkan berupa file e-dokumen, *signature*: (e-dokumen).sig, dan tandatangan *offline*. File lampiran dapat dikirim bersama-sama atau terpisah. Contoh tampilan e-mail yang diterima verifier diberikan pada gambar 4.12. Pada gambar 4.12 file lampiran berupa file e-dokumen adalah COVER.doc, file *signature* ttd1, berkas *signature* COVER.docx_54c727d31097f.



Gambar 4.11 Contoh E-mail yang dikirim Beserta File Lampiran (yang Sah)

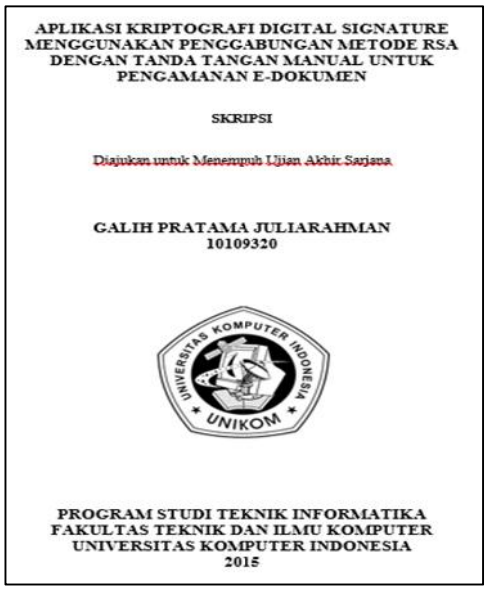


Gambar 4.12 Contoh E-mail yang Diterima Verifier Beserta File Lampiran (yang Sah)

Pada contoh kasus 2 sampai 8 diasumsikan ada *attacker* yang mengirimkan file lampiran yang tidak sah yaitu dengan isi file e-dokumen yang sudah diubah/tidak sah dan atau penggantian kunci publik dan atau penggantian pada *signature*. Diasumsikan pula *attacker* yang tidak merusak kunci publik, privat dan *signature*, tapi hanya mengganti dengan kunci lain dan atau *signature* lain. Kunci atau *signature* rusak didefinisikan sebagai hasil peng-edit-an isi file kunci atau *signature*. Tampilan isi kunci privat dan publik pada mode base64 dari ttd1.jpg diberikan masing-masing pada gambar 4.6 dan 4.7. Hasil verifikasi berdasarkan beberapa kemungkinan kasus yang dapat terjadi, sebagai berikut:

- Kasus 1 : e-dokumen sah/ otentik, *signature* dan kunci publik sah.

  Gambar 4.13 menunjukkan isi sebagian halaman pertama file "COVER.doc" yang sah/masih utuh.



APLIKASI KRIPTOGRAFI DIGITAL SIGNATURE MENGGUNAKAN PENGGABUNGAN METODE RSA DENGAN TANDA TANGAN MANUAL UNTUK PENGAMANAN E-DOKUMEN

SKRIPSI

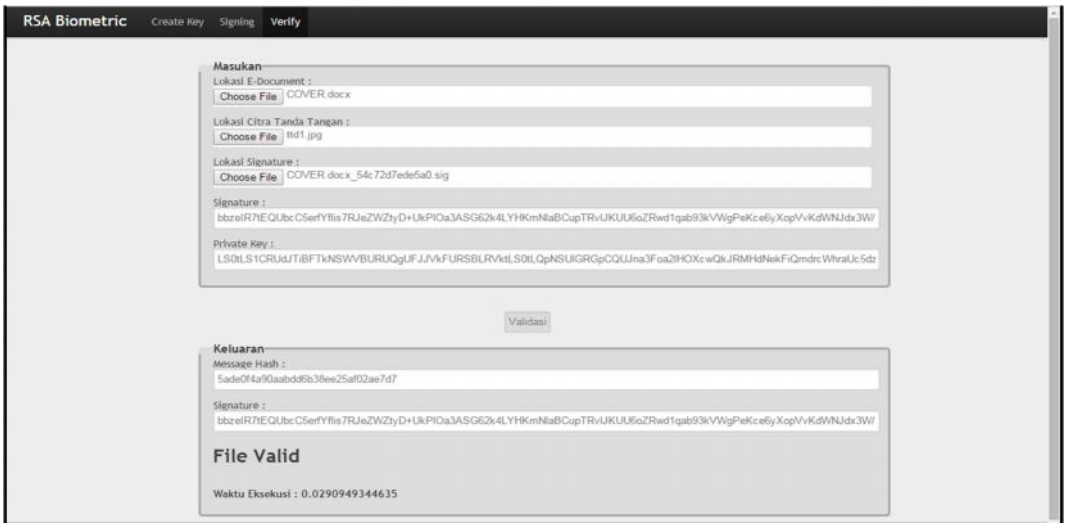
Diajukan untuk Menempuh Ujian Akhir Sarjana

GALIH PRATAMA JULIARAHMAN
10109320

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA
2015

Gambar 4.13 Tampilan Potongan Halaman Pertama File COVER.doc yang Diunduh Verifier (File Otentik/ Sah)

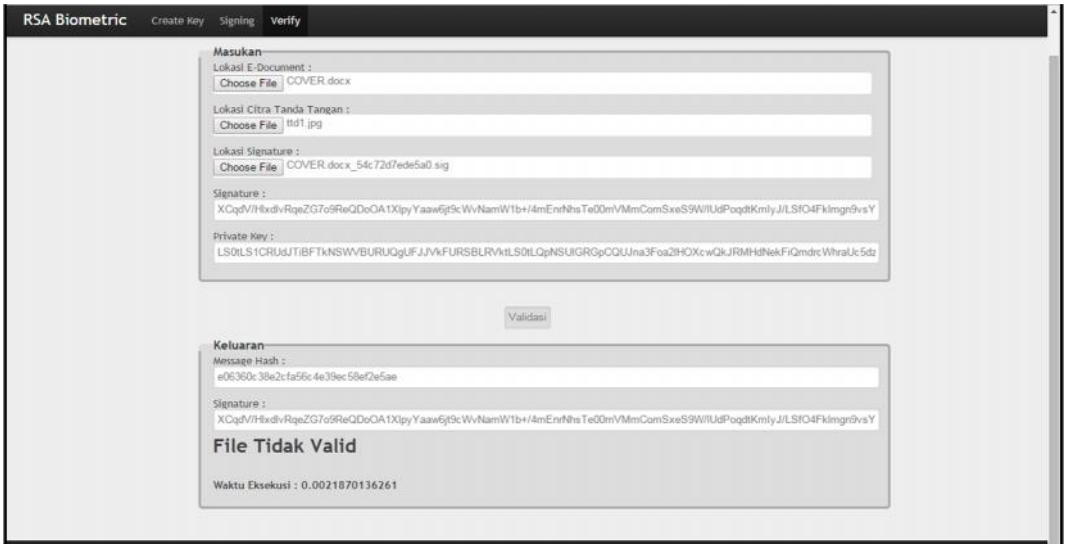
Gambar 4.14 menunjukkan hasil verifikasi oleh verifier terhadap file lampiran yang diunduh berdasarkan gambar 4.12



Gambar 4.14 Hasil Verifikasi Terhadap File Lampiran yang Diunduh Berdasarkan Gambar 4.12

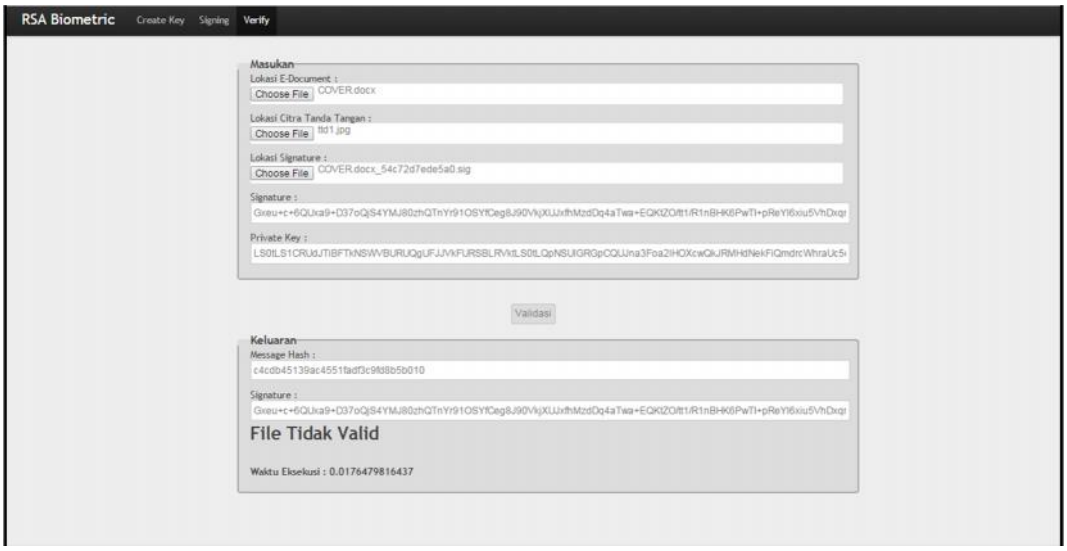
Berdasarkan gambar 4.17, jika e-dokumen masih utuh/ asli dan signer adalah signer sebenarnya (*signature* dan kunci private yang sah) maka hasil verifikasi adalah “Valid”.

- Kasus 2: e-dokumen otentik, *signature* yang sah dan kunci private tidak sah.



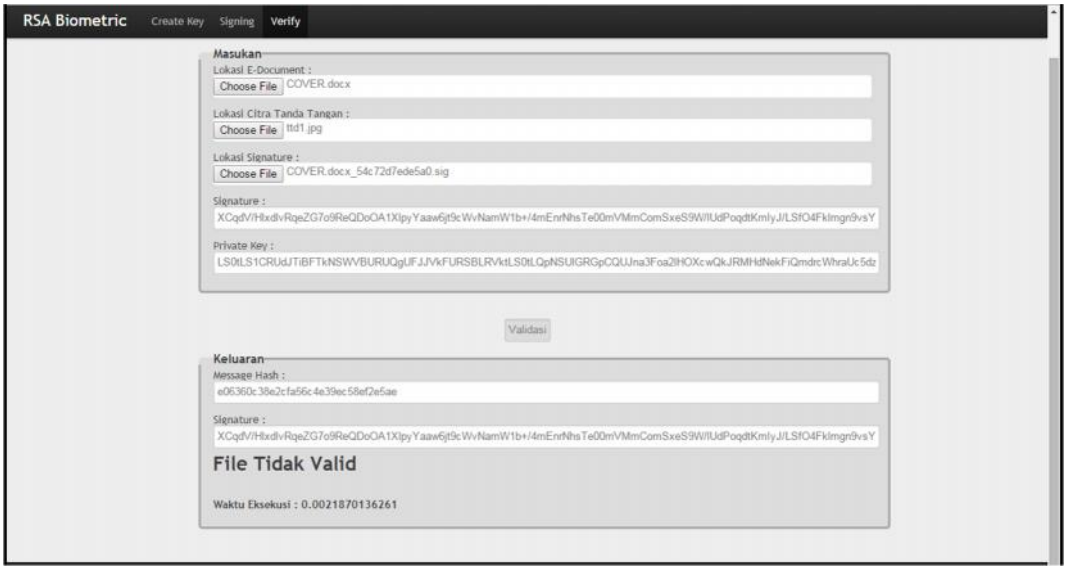
Gambar 4.15 Hasil Verifikasi Kasus 2

- Kasus 3: e-dokumen otentik, *signature* yang tidak sah dan kunci private sah



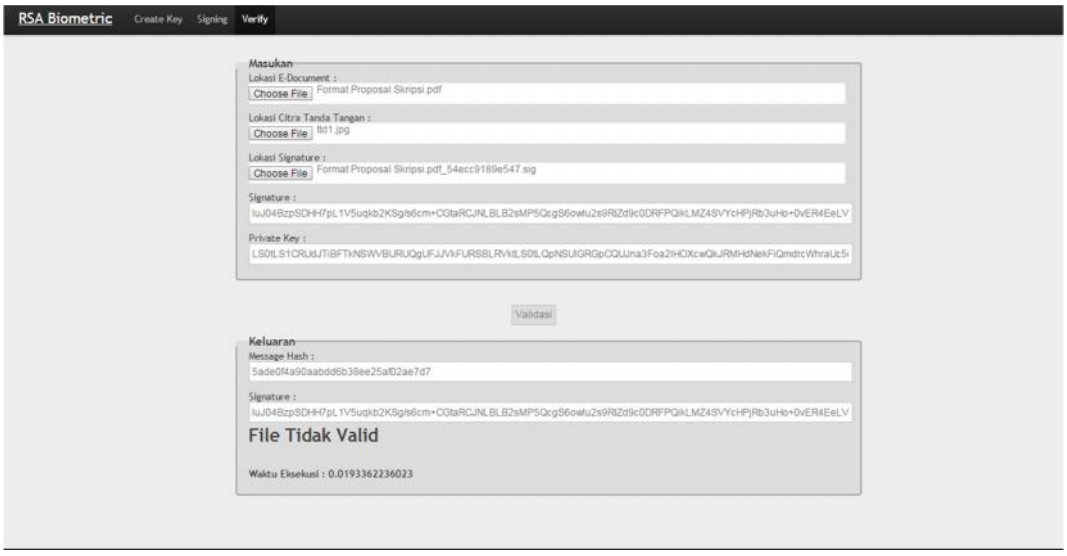
Gambar 4.16 Hasil Verifikasi Kasus 3

- Kasus 4: e-dokumen otentik, *signature* yang tidak sah dan kunci private tidak sah.



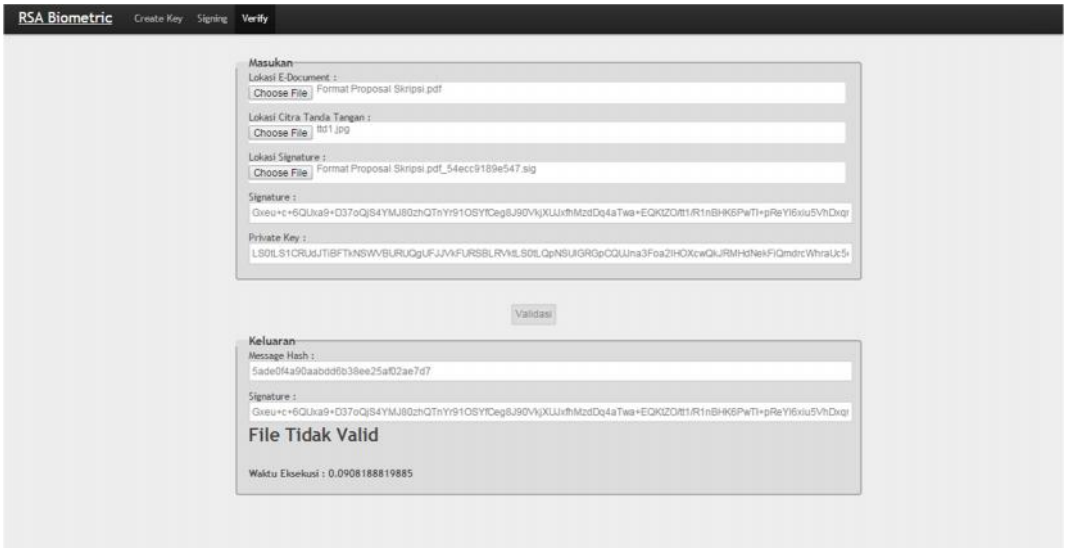
Gambar 4.17 Hasil Verifikasi Kasus 4

- Kasus 5 : e-dokumen tidak sah/ tidak otentik, *signature* sah dan kunci private sah.



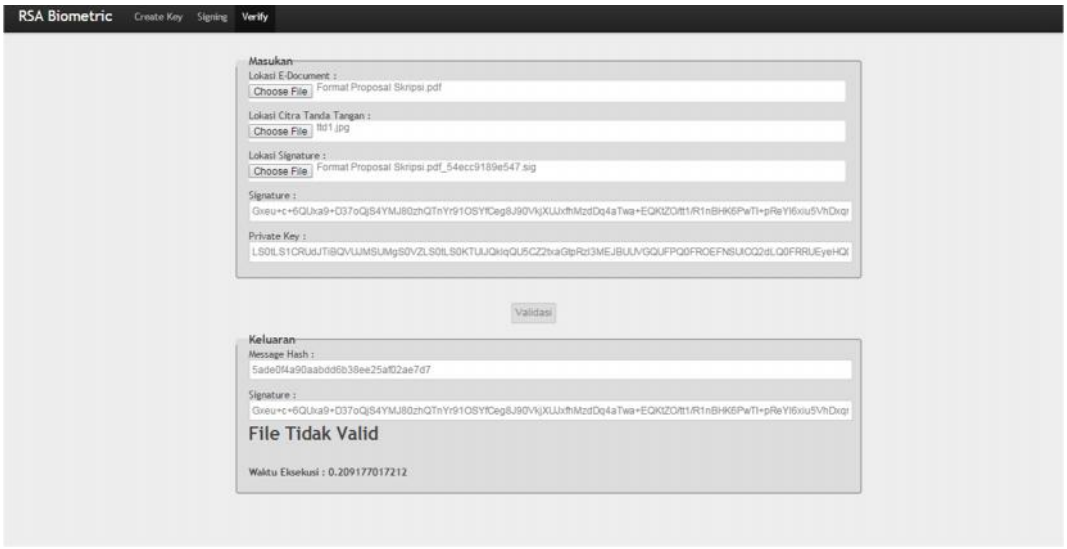
Gambar 4.18 Hasil Verifikasi Kasus 5

- Kasus 6 : e-dokumen tidak sah/ tidak otentik, *signature* sah dan kunci private tidak sah.



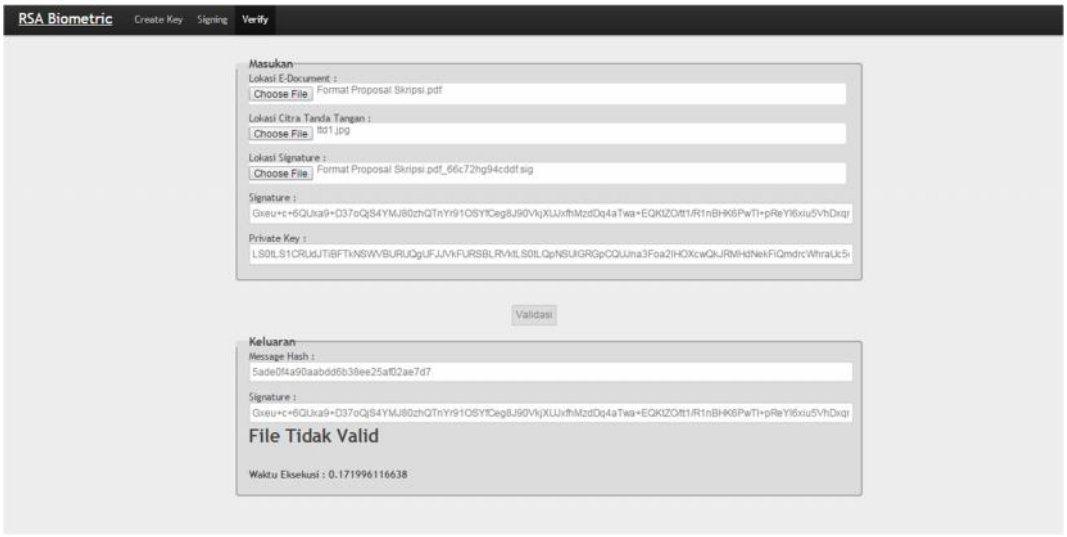
Gambar 4.19 Hasil Verifikasi Kasus 6

- Kasus 7 : e-dokumen tidak sah/ tidak otentik, *signature* tidak sah dan kunci private sah.



Gambar 4.20 Hasil Verifikasi Kasus 7

- Kasus 8 : e-dokumen tidak sah/ tidak otentik, *signature* tidak sah dan kunci private tidak sah.



Gambar 4.21 Hasil Verifikasi Kasus 8

Berdasarkan hasil pengujian yang berbeda ada pada nilai message hash. Hal ini berarti e-dokumen tidak sah/ tidak otentik. Sedangkan nilai *signature*-nya juga tidak sama, karena *signature* dan kunci *Private*-nya tidak sah.

Dari hasil verifikasi kasus 1 sampai 8 dapat diringkas pada tabel 4.2.

Tabel 4.2 Rekapitulasi Hasil Verifikasi dari Kasus 1 sampai 8

Kasus No	e-dokumen		*Signature*		Kunci *Private*		Hasil verifikasi
	S	TS	S	TS	S	TS	
1	1	0	1	0	1	0	Valid
2	1	0	1	0	0	1	Tidak valid
3	1	0	0	1	1	0	Tidak valid
4	1	0	0	1	0	1	Tidak valid
5	0	1	1	0	1	0	Tidak valid
6	0	1	1	0	0	1	Tidak valid
7	0	1	0	1	1	0	Tidak valid
8	0	1	0	1	0	1	Tidak valid

Keterangan: S = sah, TS = tidak sah, 1 = ya, 0 = tidak.

Dari tabel 4.2 dapat disimpulkan bahwa hasil verifikasi valid hanya jika e- dokumen, *signature* dan kunci publik-nya sah. Sah atau otentik dan tidak sah atau tidak

Hal ini membuktikan dengan operasi logika "e-dokumen and *signature* and kunci *Private*" dimana hanya menghasilkan true jika ketiganya true atau dalam penelitian ini disebut "sah".

Berdasarkan simulasi pada kasus 2 sampai 8, jika hasil verifikasinya adalah "Tidak Valid" maka salah satu atau kedua kemungkinan di bawah ini terjadi, yaitu :

1. E-dokumen mengalami perubahan, penambahan atau pengurangan isinya atau sudah tidak utuh/ otentik, dijamin dengan nilai massage hash e-dokumen.
2. Penandatangan adalah tidak sah yang berarti bukan orang sebenarnya sebagai signer, dijamin dengan nilai *signature* dan atau kunci publiknya. Kemungkinan lain jika hasil verifikasi "Tidak valid" adalah terjadi kesalahan pengiriman file e-dokumen dan atau *signature* dan atau kunci publiknya. Jadi jika hasil verifikasi "tidak valid" maka verifier dapat meyakini bahwa file e-dokumen yang diterima tidak sah/ tidak otentik dan atau signer tidak sah.
3. Dimungkinkan juga oleh *attacker*, *signature* yang dikirim oleh signer dirusak. Sebagai contoh kunci dan *signature* yang rusak didapat dengan mengganti (me-edit manual) isi *signature* yang sah lewat text editor/ notepad dan disimpan dalam file untuk kunci dan sig untuk *signature*, atau disimpan dengan format txt dan di-rename oleh *attacker* kunci dan atau *signature* yang dirusak dikirimkan kepada verifier.