Implementasi Digital Signature pada Soal Ujian Akademik

Dimas Lucky Mahendra - 13518003
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13518003@std.stei.itb.ac.id

***Abstrak*—Pada masa pandemi sekarang ini, pelaksanaan ujian akademik untuk setiap jenjang pendidikan dilakukan secara daring. Para tenaga pengajar pun pada akhirnya akan membagikan soal ujian kepada para murid/mahasiswanya secara daring pula. Hal ini menyebabkan resiko soal tersebut tersebar menjadi lebih tinggi. Untuk menjamin integritas dari soal tersebut, dan menjamin juga bahwa penerima soal tersebut merupakan orang-orang yang seharusnya, tanda tangan digital dapat diimplementasikan. Dengan diterapkannya tanda tangan digital pada soal ujian yang akan disebarkan, diharapkan dapat menghindari penyalahgunaan dari soal ujian tersebut.**

Keywords—Digital Signature; RSA; SHA-256; Hash; Kriptografi

I. Pendahuluan

Pengiriman pesan merupakan sebuah hal yang sudah menjadi kegiatan sehari-hari manusia. Saat ini, pengiriman pesan dari satu pihak ke pihak lain perlu untuk dijaga keamanannya, dengan tujuan untuk menjaga privasi. Keamanan pesan yang dikirim memerlukan sebuah sistem verifikasi antara pengirim dan penerima, agar dapat memastikan bahwa pesan tersebut tidak bocor dan diterima oleh orang-orang yang seharusnya tidak dapat menerima pesan tersebut.

Pada masa pandemi sekarang ini, sistem akademik tetap harus berjalan. Meskipun sekarang semua hal dilakukan dengan cara daring, aktivitas-aktivitas akademik harus tetap dilaksanakan. Ujian yang biasanya dilakukan secara luring terpaksa harus diadakan secara daring juga. Hal ini menimbulkan masalah baru, yaitu meningkatnya risiko dari soal-soal ujian yang dibagikan untuk bocor dan diterima oleh orang-orang yang tidak bertanggung jawab. Untuk menangani masalah tersebut, diperlukan sebuah sistem untuk meningkatkan keamanan soal-soal tersebut.

Digital signature merupakan sebuah sistem keamanan pesan yang telah sering digunakan sejak dahulu kala. *Digital signature* bukan merupakan sebuah tanda tangan yang didigitisasi, namun merupakan sebuah nilai kriptografis yang bergantung pada isi pesan dan kunci yang digunakan. Teknik *digital signature* memanfaatkan metode *hash* dan juga metode kriptografi kunci-publik. Dengan mengimplementasikan *digital signature*, pengirim pesan dapat memastikan bahwa pesan yang dikirim hanya dapat diterima oleh penerima yang diinginkan.

Dua teknik yang digunakan untuk mengimplementasikan *digital signature*, *hash* dan kriptografi kunci-publik, memiliki banyak cara untuk diimplementasikan. Salah satu algoritma *hash* yang sudah sering digunakan adalah SHA-256. Untuk kriptografi kunci-publik, salah satu algoritma yang dapat digunakan adalah algoritma RSA. Pada implementasinya, *digital signature* mengkombinasikan kedua algoritma ini untuk membuat sebuah tanda tangan digital yang dapat diverifikasi dengan menggunakan kunci publik yang dihasilkan.

II. Dasar Teori

A. Digital Signature

Digital signature merupakan sebuah metode yang umum digunakan sebagai sistem keamanan pengiriman pesan. Sesuai dengan konsep kriptografi, sistem *digital signature* memberikan semua aspek yang ada dari konsep kriptografi, yaitu *confidentiality*, *authentication*, *data integrity*, dan *nonrepudiation*. *Confidentiality*, atau kerahasiaan pesan, diterapkan dengan cara enkripsi dan dekripsi pesan [1]. *Authentication*, *data integrity* (keaslian pesan), dan *nonrepudiation* (anti-penyangkalan) diterapkan dengan cara memberikan tanda tangan digital.

Tanda tangan digunakan karena sifat-sifat yang dimiliki oleh tanda tangan tersebut. Sebuah tanda tangan merupakan sebuah bukti yang otentik, tidak dapat dilupakan, dan tidak dapat disangkal. Selain itu, tanda tangan tidak dapat dipindah untuk digunakan ulang dan juga dokumen yang sudah ditandatangani tidak dapat diubah lagi. Kelebihan dari tanda tangan digital jika dibandingkan dengan tanda tangan tertulis terletak pada perbedaannya. Tanda tangan digital selalu berbeda-beda antara satu dokumen dengan yang lain, tidak seperti tanda tangan tertulis yang selalu sama.

Digital signature menggunakan teknik *hash*, dimana isi dari dokumen tersebut diproses. *Output* dari proses *hashing* adalah sebuah nilai *hash* yang digunakan sebagai tanda tangan yang akan ditambahkan ke dalam dokumen. Pada teknik kriptografi kunci-publik, hasil nilai dari proses *hash* akan diproses dengan cara dienkripsi sesuai algoritma yang digunakan. *Output* dari proses kriptografi kunci-publik merupakan sebuah tanda tangan dan kunci publik yang dapat digunakan oleh penerima untuk kepentingan verifikasi.

Untuk proses verifikasi, sistem akan membandingkan hasil dari proses *hash*. Plainteks akan dihitung nilai hashnya (h), kemudian *digital signature* akan didekripsi (h') menggunakan kunci publik. Nilai h dan h' akan dibandingkan. Apabila bernilai sama, maka tanda tangan valid.

B. Hash

Hashing merupakan sebuah teknik yang memanfaatkan sebuah fungsi matematika yang dapat mengubah *input* menjadi sebuah *output* yang terenkripsi. Dengan menggunakan *hash*, *input* dapat diubah menjadi sebuah pesan yang tidak dapat terbaca oleh sembarang orang.

Salah satu algoritma dari *hash* adalah SHA-256. SHA merupakan fungsi *hash* satu arah yang dibuat oleh NIST dan digunakan bersama DSS (*Digital Signature Standard*). SHA dibuat berdasarkan algoritma MD4 yang dibuat oleh Ronald L. Rivest dari MIT. Algoritma SHA memiliki beberapa varian yang memiliki perbedaan pada ukuran dari *input*nya. Dengan kemampuan untuk menerima masukan berukuran maksimum 2^{64} bit, SHA menghasilkan *message digest* yang memiliki panjang 160 bit [2].

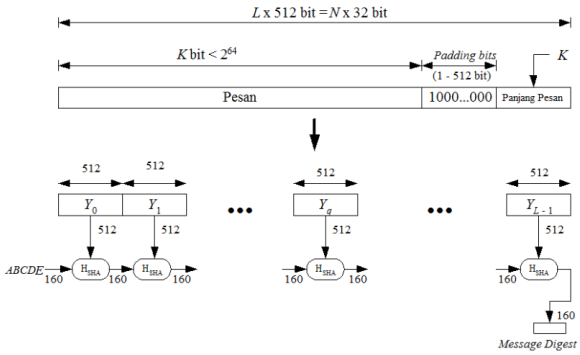

Fig. 1. Gambaran umum proses SHA-1

Pada makalah ini, varian SHA yang akan digunakan adalah SHA-256. SHA-256 memiliki kemampuan untuk mengeluarkan *output* dengan ukuran 256 bit. Proses SHA-256 diawali dengan menambahkan bit-bit pengganjal pada pesan yang ingin diproses. Kemudian setelah diberi bit-bit pengganjal, pesan kemudian ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula. SHA membutuhkan 8 buah penyangga (*buffer*) yang memiliki panjang 32 bit untuk masing-masing *buffer*. *Buffer* ini diberi nama h0, h1, h2, h3, h4, h5, h6, dan h7 yang diinisialisasi dengan nilai hex. Nilai-nilai tersebut adalah sebagai berikut:

- h0 = 0x6a09e667
- h1 = 0xbb67ae85
- h2 = 0x3c6ef372
- h3 = 0xa54ff53a
- h4 = 0x510e527f
- h5 = 0x9b05688c
- h6 = 0x1f83d9ab
- h7 = 0x5be0cd19

Selain *buffer*, konstanta lain dibuat untuk merepresentasikan nilai dari akar dari 64 bilangan prima pertama (2-311). Kemudian hasil dari pesan akan dipecah menjadi beberapa bagian dengan tiap bagian memiliki panjang 512 bit. Kemudian, sebuah *message schedule* akan dibuat berdasarkan *input* yang diterima. Kompresi akan dilakukan dengan cara membuat variabel baru yaitu A-H dengan nilai yang sama dengan konstanta *buffer*. Variabel tersebut akan diubah dengan memutasikan value dari nilai A-H sebanyak 64 kali. Setelah dikompresi, nilai final akan didapatkan dengan cara menambahkan nilai *buffer* dengan nilai A-H (h0 + A, h1 + B, dst.) dan menggabungkan seluruh hasil nilai tersebut.

C. Kriptografi Kunci-Publik

Kriptografi kunci publik merupakan sebuah sistem kriptografi yang menggunakan sepasang kunci, kunci publik dan kunci privat. Kunci publik merupakan kunci yang dapat disebarkan sementara kunci privat merupakan sebuah kunci yang hanya dimiliki oleh pemilik. Kunci publik digunakan untuk mengenkripsi pesan, sementara kunci privat digunakan untuk mendekripsi pesan. Maka dari itu, pesan yang terenkripsi hanya dapat didekripsi oleh penerima yang seharusnya.

Algoritma kriptografi kunci-publik memiliki banyak jenis algoritma, salah satunya adalah algoritma RSA. RSA merupakan algoritma kunci-publik yang paling terkenal dan paling banyak aplikasinya. Algoritma ini ditemukan oleh tiga peneliti dari MIT (Massachussets Institute of Technology), yaitu Ronald Rivest, Adi Shamir, dan Len Adleman pada thaun 1976 [3]. Keunggulan algoritma ini terletak pada keamanan algoritma tersebut, dimana menemukan faktor prima dari bilangan bulat yang besar adalah hal yang sulit untuk dilakukan.

Algoritma RSA memiliki beberapa properti, yaitu:

- p dan q (bilangan prima) (rahasia)
- n = p x q (tidak rahasia)
- $\phi(n) = (p-1)(q-1)$ (rahasia)
- e (kunci enkripsi) (tidak rahasia)
- d (kunci dekripsi) (rahasia)
- m (plainteks) (rahasia)
- c (cipherteks) (tidak rahasia)

Kunci enkripsi didapatkan dengan mencari bilangan e, dimana pembagi bersama terbesar dari e dan $\phi(n)$ adalah 1. Kunci dekripsi didapatkan dengan mencari bilangan d, dimana d adalah bilangan yang kongruen dengan.

$$e^{-1} \, mod \, (\phi(n)) \quad (1)$$

Proses RSA dilakukan dengan cara membangkitkan kunci publik dan juga kunci privat. Kemudian pesan akan dipecah menjadi beberapa blok dengan panjang yang sama. Tiap blok pesan (b) kemudian dienkripsi dengan menggunakan kunci enkripsi (e) dengan cara:

$$b^e \ mod \ n \quad (2)$$

Hasil enkripsi tersebut kemudian digabungkan untuk menjadi sebuah *ciphertext*.

Proses dekripsi RSA dilakukan dengan menggunakan kunci privat. Proses dekripsi dilakukan dengan membagi *ciphertext* menjadi beberapa blok pesan (b). Blok pesan tersebut kemudian didekripsi dengan cara menghitung b^d mod n. Hasil dekripsi tiap blok kemudian digabungkan untuk menjadi plainteks awal.

III. Implementasi

Metode yang digunakan akan memanfaatkan metode *hash* SHA-256 dan juga algoritma RSA sebagai metode kriptografi kunci-publik.

A. Prosedur Konversi File PDF to Text File

Pertama-tama agar dapat memproses file dengan format PDF, file harus dikonversi terlebih dahulu menjadi *Text File*. Proses konversi ini memanfaatkan *library* PyPDF2 yang dapat diinstall pada python. Isi dari file PDF akan dibaca dan akan membuat *Text File* baru dengan isi sesuai dengan file PDF yang digunakan.

UJIAN AKHIR SEMESTER - UAS
DK 3014 - PSIKOLOGI PERSEPSI
Semester I 2021/2022

Kelas	: K01 - K02 - K03
Tanggal	: **Senin, 20 Desember 2021**
Waktu Pengerjaan	: **12.30 – 15.30 WIB**
Dosen	: 1. Dra. Lies Neni Budiarti, Psi., M. Si
	2. Miranti Sari Rahma, S.T., M. Ds

SOAL

Setelah mengikuti perkuliahan selama satu semester secara daring, saudara telah mengetahui mengenai definisi psikologi persepsi beserta fenomena – fenomenanya. Untuk mengetahui sejauh mana pemahaman materi yang telah disampaikan anda diminta untuk menjelaskan pertanyaan – pertanyaan dibawah ini secara komprehensif.

1. Dari hukum-hukum pengorganisasian sistem persepsi dalam mengolah stimulus dapat disimpulkan secara umum hukum/prinsip2 persepsi sbb, al:
   a. *selective functional*
   b. medan perseptual dari struktur dan sub nya (sub struktut) selalu diorganisasikan dan diberi arti/ makna

   Jelaskan pendapat saudara dan berikan contoh prinsipnya
2. Terkait fenomena Sensasi - Persepsi Warna, jelaskan mengapa perlu hati2 dalam mengaplikasikan warna. Apakah ada perbedaan preferensi warna berdasarkan gender/ usia/ latar belakang seseorang? Berikan bukti dari literatur/ hasil riset ataupun penelitian yang telah ada.
3. Terkait fenomena Persepsi Jarak, jelaskan mengapa sistem penglihatan manusia menggunakan dua mata serta apa yang menjadi kelebihan penglihatan dengan kedua mata tersebut. Saudara dapat menjelaskannya dengan membandingkan syarat jarak binokuler dengan monokuler. Berikan contoh konkritnya.
4. Terkait fenomena Ilusi persepsi uraikan bagaimana pendapat saudara mengenai *Seeing is beliefing*.
5. Bahasan Persepsi Interpersonal merupakan inti dari Komunikasi Interpersonal. Persoalan -persoalan keakuratan/ kecermatan Persepsi interpersonal pada gilirannya dapat mempengaruhi keselarasan komunikasi interpersonal/ social. Terdapat beberapa faktor yang dapat mempengaruhinya seperti parabahasa/ vocal, *stereotyping*, *hallo effect*, atribusi dan *implicit personality theory*. Uraikan penjelasan Anda tentang implikasi dari jalinan komunikasi interpersonal seseorang terhadap kesuksesan dirinya baik dalam berkarir khususnya, maupun kehidupan sosial secara umum.

Fig. 2. Contoh file PDF soal ujian

```
UJIAN AKHIR SEMESTER
-

UAS

DK 3014
-

PSIKOLOGI PERSEPSI

Semester I

202
1
/202
2


Kelas


: K01
-

K02
-

K03

Tanggal


:
Senin, 20 Desember 2021

Waktu Pengerjaan
```

Fig. 3. Contoh hasil konversi

B. Prosedur Hashing

Prosedur *hashing* dilakukan dengan algoritma SHA-256. Pertama-tama akan diinisialisasi nilai-nilai *buffer* h0-h7. Lalu pesan akan diubah menjadi bentuk bit dan ditambahkan bit-bit pengganjal. Pesan tersebut kemudian akan dipecah menjadi beberapa bagian sepanjang 512 bit. Setelah itu, *message schedule* akan dibuat dan akhirnya akan dilakukan proses kompresi. Kemudian *buffer* akan ditambahkan dengan hasil dari kompresi dan seluruh hasil penambahan tersebut akan dikonkatenasi. Hasil konkatenasi tersebut merupakan hasil *hash* yang akan dipakai sebagai tanda tangan digital.

C. Prosedur RSA

Proses enkripsi RSA dilakukan dengan menginisialisasi nilai p, q, dan e. Nilai p dan q didapatkan secara acak dari sebuah file yang berisi bilangan-bilangan prima. Nilai e didapatkan dari nilai acak dari 2 sampai ϕ(n) dimana nilai PBB e dan ϕ(n) adalah 1.

Proses enkripsi dengan menggunakan RSA diawali dengan mengubah pesan plainteks menjadi angka sesuai dengan urutan alfabet. Kemudian pesan yang sudah diubah akan dipecah menjadi beberapa blok pesan dengan panjang yang sama. Tiap blok tersebut kemudian akan dienkripsi dengan cara:

$$e^{-1} \ mod \ (\phi(n)) \quad (3)$$

Hasil dari enkripsi tiap blok kemudian dikonkatenasi sehingga menjadi sebuah *ciphertext*.

Proses dekripsi dengan menggunakan RSA diawali dengan membagi *ciphertext* menjadi blok-blok pesan dengan panjang yang sama. Setiap blok pesan ini kemudian didekripsi dengan cara:

$$b^e \ mod \ n \quad (4)$$

Hasil dari dekripsi tiap blok kemudian dikonkatenasi sehingga menjadi plainteks awal.

D. *Digital Signature*

Proses penandatanganan digital dilakukan dengan mengambil nilai dari proses *hash* dan enkripsi. Hasil dari kedua proses tersebut menjadi sebuah tanda tangan yang kemudian ditambahkan di akhir file. Tanda tangan dibatasi dengan template pemisah seperti "**Digital Signature**".

```
hello world

**Digital Signature**0x6ld**Di
gital Signature**
```

Fig. 4. Contoh hasil file yang ditandatangan

E. *Proses Verifikasi*

Proses verifikasi tanda tangan dilakukan dengan mengambil tanda tangan yang sudah diimbuhkan ke dalam file plainteks awal. Tanda tangan tersebut kemudian didekripsi dengan cara dekripsi RSA dengan menggunakan kunci publik.

Hasil dari dekripsi tersebut kemudian dibandingkan dengan nilai *hash* dari plainteks. Apabila hasilnya sama, maka tanda tangan tervalidasi.

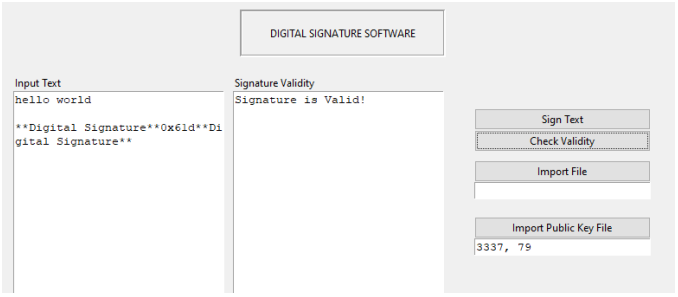

Fig. 5. Contoh hasil validasi tanda tangan

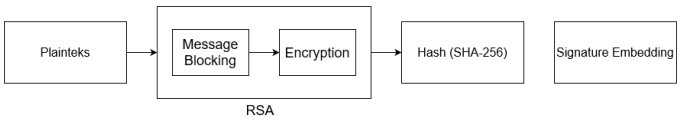

Fig. 6. Skema pembuatan tanda tangan digital

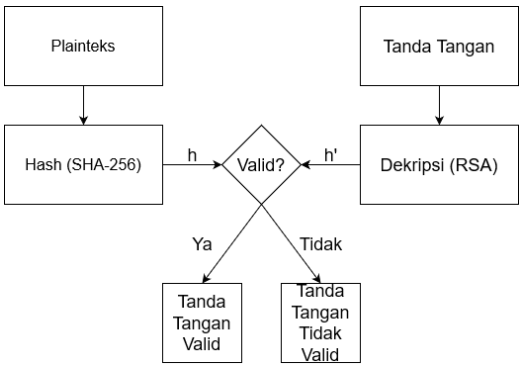

Fig. 7. Skema validasi tanda tangan

IV. HASIL DAN ANALISA

Bagian ini berisi pengujian dari implementasi yang dilakukan.

A. *Pengujian*

Pengujian dilakukan dengan cara mengkonversi file pdf "soal uas psiper.pdf" ke dalam format *Text File*. File teks tersebut kemudian di-*import* ke dalam program untuk diproses. Kemudian akan dicoba untuk menambahkan tanda tangan digital ke dalam file.

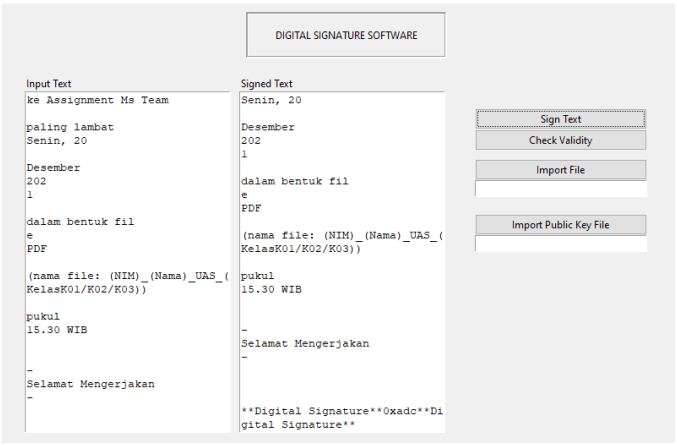

Fig. 8. Contoh hasil penambahan tanda tangan digital

Hasil file yang sudah ditandatangan kemudian di-*import* kembali ke dalam program untuk dicek validaitas dari tanda tangan tersebut. Kemudian akan di-*import* juga kunci publik yang sudah di-*generate* pada saat proses penambahan tanda tangan.

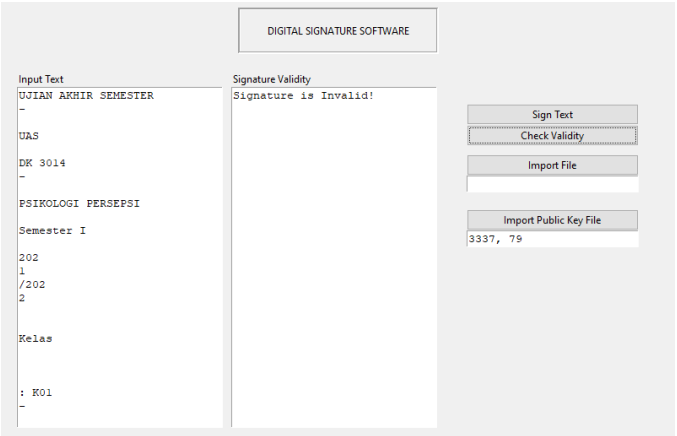

Fig. 9. Proses validasi tanda tangan menggunakan kunci publik

Berikut merupakan pengujian pada file teks biasa

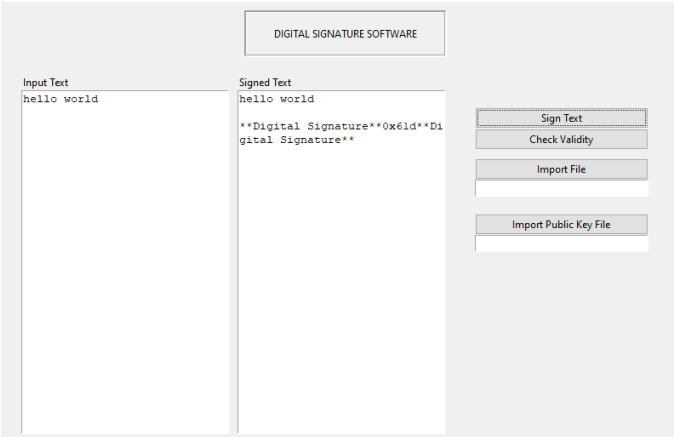

Fig. 10. Hasil pengujian penandatanganan pada file teks biasa

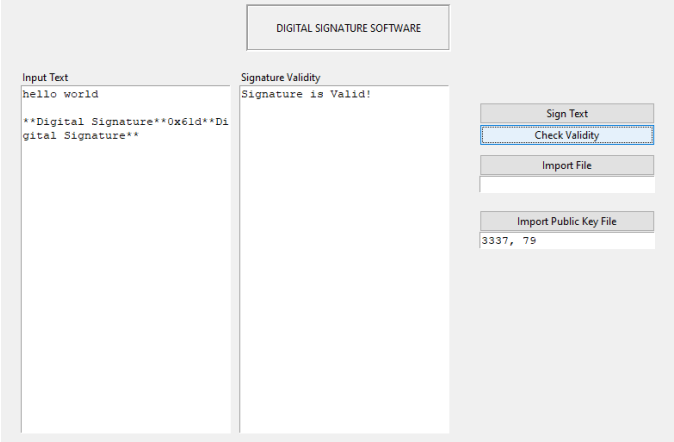

Fig. 11. Hasil pengujian validasi tanda tangan pada file teks biasa

B. Analisa

Pada hasil pengujian yang menggunakan file PDF, validasi menghasilkan hasil gagal, dikarenakan belum ditanganinya apabila ada simbol "\n". Hal ini dibuktikan pada pengujian file teks langsung, dimana validasi mengembalikan hasil valid.

V. KESIMPULAN

Berdasarkan hasil pengujian, dapat disimpulkan bahwa implementasi *digital signature* dapat menjamin keaslian data dan juga dapat menjamin bahwa penerima data merupakan penerima yang seharusnya. Secara aspek keamanan, dengan menggunakan algoritma RSA untuk proses enkripsi, kesulitan berada pada menemukan faktor prima dari bilangan bulat yang besar. Namun, perlu diperhatikan bahwa segala *testcase* seperti adanya symbol "\n" seperti pada pengujian harus ditangani agar program bekerja dengan baik.

Dengan kata lain, dapat disimpulkan bahwa penerapan *digital signature* pada soal ujian akademik merupakan hal yang baik untuk dilakukan. Hanya saja, implementasi dari *digital signature* ini harus diterapkan dengan baik terlebih dahulu sebelum digunakan untuk kepentingan akademik yang nyata.

UCAPAN TERIMA KASIH

Dalam proses penulisan makalah dengan judul "Implementasi Digital Signature pada Soal Ujian Akademik", penulis menerima banyak bantuan seperti dari teman-teman Saya. Tidak lupa Saya juga berterima kasih kepada Pak Rinaldi Munir dalam membimbing saya pada mata kuliah ini. Saya memohon maaf apabila ada kesalahan kata. Saya harap makalah Saya ini dapat membantu dan memberikan dampak baik kepada siapapun yang membaca makalah ini.

REFERENSI

[1] Munir, Rinaldi. "Tanda-tangan Digital", https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Tanda-tangan-digital-2020.pdf (terakhir diakses pada 22.55 WIB, 20 Desember 2021)

[2] Munir, Rinaldi. "Secure Hash Algorithm (SHA)", https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Fungsi-hash-SHA.pdf (terakhir diakses pada 22.55 WIB, 20 Desember 2021)

[3] Munir, Rinaldi. "Algoritma RSA", https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Algoritma-RSA-2020.pdf (terakhir diakses pada 22.55 WIB, 20 Desember 2021)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Desember 2021

Dimas Lucky Mahendra, 13518003