Penggunaan Digital Signature untuk Menjamin Integritas Salinan Al Quran Digital

Khairunnisa Rifdah 18218008
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 18218008@std.stei.itb.ac.id

***Abstract*—Al Quran sejak pertama kali diturunkan senantiasa dijaga dan terjaga keasliannya, juga saat Al Quran tersedia dalam salinan digital tentunya harus diterapkan prosedur agar terhindar dari berbagai bentuk pemalsuan. Penerapan tanda tangan digital merupakan salah satu upaya untuk mencegah perubahan terhadap salinan Al Quran tersebut. Dengan menggunakan algoritma kunci publik dan *hash*, salinan Al Quran digital yang tersebar dapat diperiksa keasliannya dengan mudah.**

Keywords—Tanda Tangan Digital, Anti-Penyangkalan, Hash, ElGamal, Algoritma Kunci Publik, Al Quran Digital

I. PENDAHULUAN

Bagi pemeluk agama Islam, Al Quran merupakan kitab suci yang terjaga dan dijaga kesucian dan keasliannya sejak pertama kali diturunkan. Pertama kali diturunkan kepada Nabi Muhammad pada bulan Ramadhan melalui wahyu yang disampaikan oleh Malaikat Jibril, pada awalnya Al Quran hanya berupa hafalan dan seiring berkembangnya zaman dan kebutuhan akan dokumentasi, Al Quran ditulis di atas berbagai media seperti bebatuan dan pelepah kurma. Dan pada saat ini karena kemajuan teknologi, Al Quran dapat disalin dengan mudah melalui media teks bahkan rekamannya dapat disebarkan melalui media audio.

Penggunaan Al Quran digital berkembang pesat seiring perkembangan zaman. Al Quran dalam bentuk digital dapat diakses kapanpun dan dimanapun dengan gratis bahkan saat ini banyak aplikasi yang menyediakan Al Quran digital sehingga pengguna dapat membaca Al Quran melalui ponsel pintar mereka. Al Quran digital pertama kali muncul salinannya pada 19 Maret 2007 dan dapat diunduh dan digunakan di *personal computer*. Pencetusan pembentukan Al Quran digital ini diprakasai oleh Prince Abdul Aziz bin Majed yang saat itu merupakan gubernur dari Kota Madinah. Pada tahun 2006 beliau memerintahkan kepada peneliti muslim untuk memberantas pemalsuan terhadap Al Quran dengan cara yang modern dan ilmiah. Oleh sebab itu, muncullah bentuk digital dari Al Quran untuk kepentingan integritas data dan keamanan, selain untuk keperluan akademis dan kebutuhan umat Muslim untuk membacanya kapanpun.

Al Quran digital terdiri dari 2 jenis yakni jenis berbentuk teks dan berbentuk gambar atau citra. Saat ini berbagai *platform* penyedia layanan Al Quran digital menggunakan salah satu atau kedua dari jenis tersebut. Bentuknya pun bermacam-macam, ada yang berbentuk *raw data*, pdf, gambar format *png*, dan seterusnya. Berbagai aplikasi juga telah diluncurkan seperti aplikasi *mobile* untuk android maupun iOS, aplikasi *web*, ataupun *file* salinan yang dibagikan secara bebas. Terdapat beberapa penyedia layanan Al Quran digital yang telah terpercaya dan terverifikasi, salah satunya adalah *King Fahd Glorious Quran Printing Complex* (2018). *King Fahd Glorious Quran Printing Complex* merupakan Lembaga milik pemerintah Saudi Arabia yang mempublikasikan dan mencetak Al Quran dalam berbagai media, yakni media cetak dan digital [1].

Dengan adanya salinan Al Quran dalam media digital, tentunya hal ini akan menjadi pertimbangan terkait banyak hal salah satunya adalah integritas data. Al Quran sejak pertama kali diturunkan tidak ada padanya satu perbedaan pun dan terjaga dari pemalsuan. Namun seiring dengan berkembangnya teknologi, integritas data dari Al Quran bisa dipertanyakan karena mudahnya memodifikasi data digital. Sehingga dibutuhkan standar dan otentikasi terpadu agar keaslian dari salinan Al Quran ini dapat terjaga tanpa mengurangi kemudahan dalam pengaksesan dan penyebarannya.

Salah satu cara untuk menjaga integritas suatu data adalah dengan diterapkannya tanda tangan digital atau *digital signature*, di mana metode ini dapat membuktikan keaslian dokumen digital yang telah ditandatangani bahwa tidak ada sedikitpun perubahan dari dokumen aslinya. Dokumen digital yang telah ditandatangani tidak dapat disangkal karena tanda tangan digital dibangkitkan berdasarkan isi dokumen dan kunci yang telah dibuat. Tanda tangan digital dapat dilakukan dengan dua langkah, yakni dengan mengenkripsi dokumen atau pesan dan selanjutnya diterapkan kombinasi fungsi *hash* dan kriptografi kunci publik. Namun dalam hal ini, yang dibutuhkan hanyalah keotentikan pesan atau dokumen, sementara kerahasiaan pesan tidak diperlukan. Sehingga penggunaan *digital signature* dengan kriptografi kunci publik dan fungsi *hash* dirasa cocok.

Beberapa algoritma penandatanganan yang paling popular adalah RSA dan ElGamal. Pada makalah ini akan dilakukan algoritma ElGamal terhadap pembangkitan tanda tangan digital. Algoritma tersebut memiliki pasangan kunci publik dan privat di mana kunci publik dapat disebarkan ke khalayak

umum sementara kunci privat harus disembunyikan. Sehingga hal ini menjadikan penggunaan algoritma kunci publik sangat cocok untuk kasus penjaminan integrasi salinan Al Quran digital dengan cara mengenkripsi serta mendekripsi dokumen menggunakan kunci yang telah dibangkitkan tersebut. Sementara fungsi *hash* digunakan untuk melakukan kompresi terhadap pesan sehingga panjangnya akan tetap sesuai dengan standar. *Hash* dilakukan tepat sebelum dienkripsi agar memudahkan proses enkripsi dan untuk keperluan standardisasi.

II. STUDI TERKAIT

Seiring dengan meluasnya perkembangan salinan Al Quran dalam bentuk digital serta ancaman terhadap pemalsuan teks Al Quran yang semakin beragam, banyak pihak meneliti dan berusaha mencari cara terbaik untuk menjamin keaslian dari Al Quran agar sesuai dengan apa yang diturunkan pertama kali. Juga agar sesuai dengan salinan milik lembaga yang telah diverifikasi bahwa salinan milik mereka terjamin keasliannya.

A. Verifikasi dengan *hash function* dan metode kompresi

Metode verifikasi dengan *hash function* untuk memastikan integritas dari data yang ditransmisikan, dalam hal ini adalah Al Quran digital, diusulkan dengan beberapa fungsi yang telah dikenal baik. Salah satu fungsi *hash* yang paling terkenal dan digunakan dalam penelitian ini adalah SHA256 dan RIPEMD160. Setelah dilakukan *hashing* terhadap salinan tersebut, dilakukan juga metode kompresi sehingga dihasilkan salinan Al Quran digital lebih kecil 84.73% dan 90.46% dari ukuran aslinya. [3]

Dalam proses verifikasi, dilakukan pengecekan terhadap *input* yang diterima yakni salinan dari Al Quran digital, apakah sesuai dengan salinan yang telah diverifikasi keasliannya. Salinan yang telah diverifikasi tersebut disimpan di *database* dan akan dicocokkan dengan berbagai salinan yang telah tersebar. Penggunaan fungsi *hash* SHA256 dan RIPEMD160 juga mempertimbangkan keamanan dari data dan ukuran data yang akan dilakukan *hashing.*

B. Otentikasi dari Versi Elektronik Al Quran

Otentikasi yang diusulkan terdiri dari 3 tahap yakni:

1. Inisiasi, yakni tahap di mana pengguna meminta salinan dari Al Quran baik secara gratis ataupun berbayar ke lembaga penyedia.
2. Validasi, yakni tahap mengecek otentikasi dari data yang telah diperoleh. Tahap ini merupakan tahap paling utama dari proses otentikasi karena akan dilakukan komparasi terhadap salinan yang telah terverifikasi yang ada pada *database.*
3. Persetujuan, yakni tahap di mana salinan yang telah diterima sudah disetujui bahwa salinan tersebut sesuai dan terjaga keasliannya [4].

Untuk salinan yang dijadikan acuan, dapat digunakan salinan yang diproduksi oleh *King Fahd Complex of The Printing of The Holy Quran* milik pemerintah Kerajaan Saudi Arabia.

III. DASAR TEORI

A. Tanda Tangan Digital

Tanda tangan digital adalah sebuah skema yang digunakan untuk memastikan pesan yang dikirimkan terjamin keasliannya dan terhindar dari segala bentuk serangan pemalsuan. Sebuah dokumen yang telah ditanda tangani secara digital dan telah terverifikasi terjamin keaslian isinya dan siapa pengirimnya. Cara kerjanya sama seperti tanda tangan biasa, di mana dokumen cetak dapat dijamin keaslian dan otentikasinya dengan melihat tanda tangan yang dibubuhkan oleh orang yang bersangkutan. Jika tanda tangan tersebut tidak valid maka dokumen tersebut tidak valid. Dan jika tanda tangan dipalsukan, dokumen tersebut tidak dapat dijamin keasliannya [3]. Konsep ini sama dengan tanda tangan digital yang digunakan saat ini.

Tanda tangan digital memiliki beberapa karakteristik:

1. Tanda tangan merupakan bukti yang otentik.
2. Tanda tangan tidak dapat dilupakan
3. Tanda tangan tidak dapat dipindah untuk digunakan ulang ke dokumen lain.
4. Dokumen yang telah ditandatangani tidak dapat diubah isinya.
5. Tanda tangan tidak dapat disangkal.

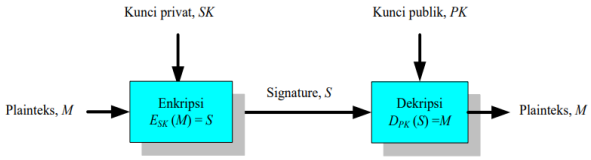
Perlu diperhatikan bahwa tanda tangan digital bukanlah tanda tangan yang ditulis di atas kertas lalu difoto dan disisipkan pada dokumen. Namun tanda tangan digital merupakan skema kriptografis yang sangat bergantung kepada isi pesan atau dokumen serta kunci yang dibangkitkan. Jika isi pesan berubah maka tanda tangan akan berubah.

Terdapat dua cara ketika ingin menandatangani sebuah dokumen atau pesan:

1. Mengenkripsi pesan dengan algoritma enkripsi tertentu. Cara ini dapat diimplementasikan dengan menggunakan skema kriptografi simetri di mana hal ini bisa langsung dibuktikan otentikasinya karena kunci simetri yang dibangkitkan hanya diketahui oleh pengirim dan penerima. Sehingga jika ada orang lain selain pengirim dan penerima yang ingin melakukan enkripsi maupun dekripsi terhadap dokumen yang telah ditandatangani tersebut maka tidak bisa kecuali ia mengetahui kunci simetri yang digunakan. Sehingga metode ini tidak dapat disangkal siapa yang mengirim dan menerima.

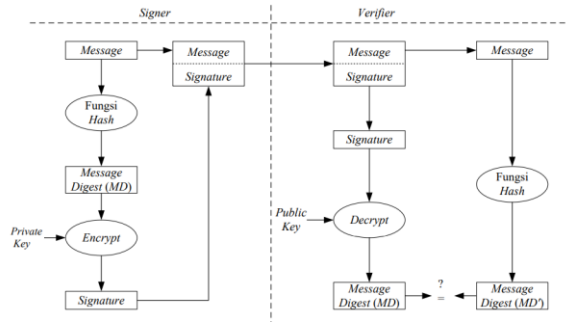
   Cara lain yang dapat digunakan adalah enkripsi pesan dengan menggunakan kriptografi kunci-publik. Di mana pembangkitan kunci menghasilkan 2 kunci yakni kunci publik dan kunci privat. Pesan dapat dienkripsi dengan kunci publik penerima serta dapat didekripsi menggunakan kunci privat penerima.

Namun hal ini tidak dapat memberikan otentikasi karena kunci publik disebarkan secara luas dan dapat diketahui oleh banyak orang. Ide ini ditemukan oleh dua orang yakni Diffie dan Hellman.



Gambar III.1 Ilustrasi Proses *Digital Signature* (sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/Tanda-tangan-digital-2021.pdf)

2. Menggunakan kombinasi fungsi *hash* dan kriptografi kunci-publik. Cara ini dapat menjawab permasalahan anti-penyangkalan pada cara sebelumnya. Karena dengan kombinasi fungsi *hash* dan kriptografi kunci publik menghasilkan penjaminan terhadap kerahasiaan pesan atau dokumen serta otentikasi dari pesan. Namun dalam beberapa kasus, kerahasiaan pesan tidak diperlukan seperti pada kasus pengecekan integritas salinan Al Quran digital ini. Karena yang dibutuhkan hanyalah apakah isi dari salinan tersebut tidak ada yang diubah, bukan kerahasiaan dari isi salinan. Salah satu algoritma yang umum digunakan untuk metode ini adalah Algoritma ElGamal.



Gambar III.2 Ilustrasi *Digital Signature* dengan *Hash* (sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/Tanda-tangan-digital-2021.pdf)

B. Algoritma Kunci Publik ElGamal

Algoritma ElGamal merupakan salah satu algoritma kriptografi kunci publik atau asimetris yang ditemukan pada tahun 1985 oleh seseorang yang bernama Taher Gamal dalam makalahnya yang berjudul "*A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.*" Algoritma ini memiliki keunggulan yang terdapat pada sulitnya menghitung logaritma diskrit sehingga keamanannya cukup tinggi. Logaritma diskrit yang menjadi titik kesulitan adalah jika sebuah bilangan *p* yang merupakan bilangan prima dan bilangan *g* dan *y* yang merupakan bilangan bulat sembarang, carilah nilai *x* agar dapat memenuhi persamaan $g^x \equiv y \pmod p$.

Adapun dalam pembangkitan kunci untuk algoritma ElGamal, terdapat beberapa tahap:

1. Pilih sembaran bilangan prima *p* di mana bilangan ini dapat disebarkan secara luas.
2. Pilih dua bilangan acak yakni *g* dan *y* dengan syarat $g < p$ dan $1 \le y \le p - 2$.
3. Hitung $g^x \equiv y \pmod p$.
4. Hasil dari pembangkitan kunci adalah sepasang kunci publik *(x, g, p)* dan kunci privat *(y, p)* [5].

Dalam proses enkripsi, terdapat beberapa tahap setelah pembangkitan kunci:

1. Susun *message* menjadi beberapa blok dengan nilai setiap blok berada pada selang [0, *p*-1].
2. Pilih sebuah bilangan acak *k*, di mana $1 \le k \le p - 2$.
3. Setiap blok *message* dienkripsi dengan persamaan (1) dan (2).

$$a = g^k \bmod p \quad (1)$$

$$b = y^k m \bmod p \quad (2)$$

Sementara untuk proses dekripsi, berikut tahap pelaksanaannya:

1. Gunakan kunci privat *y*, untuk menghitung $(a^x)^{-1} = a^{p-1-x} \bmod p$.
2. Hitung *plaintext* m dengan cara $m = b/a^x \bmod p = b(a^x)^{-1} \bmod p$.

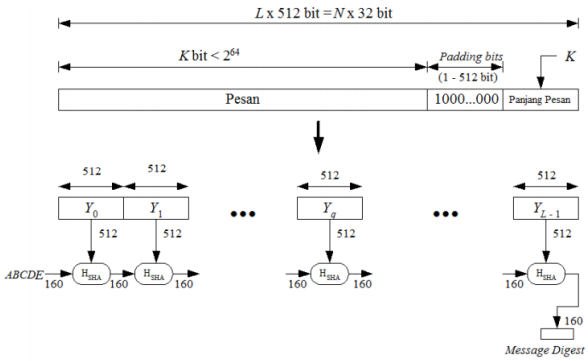
C. Fungsi *Hash* SHA-1

Fungsi *hash* SHA (*Secure Hash Algorithm)*-1 merupakan salah satu fungsi *hash* yang paling populer. SHA merupakan fungsi satu arah yang dibuat oleh *NIST* dan digunakan bersama DSS (*digital signature standard).* Pembuatan SHA didasarkan pada MD4 yang dibuat oleh Ronald L. Rivest dari MIT [6].

Adapun tahap dalam *hashing* menggunakan SHA-1 sebagai berikut:

1. Menambahkan *padding bits* sehingga panjang pesan kongruen dengan 448 (mod 512).
2. Menambahkan nilai panjang pesan sebanyak 64 bit yang menyatakan panjang pesan semula.
3. Menginisiasi penyangga MD (*buffer)* dengan masing-masing panjang 32 bit sebanyak 5 buah sehingga total *buffer* adalah 160 bit.
4. Mengolah pesan dalam blok ukuran 512 bit.

SHA-1 merupakan standar keamanan *hash* yang paling rendah karena telah berhasil diretas. Namun tujuan penggunaan fungsi *hash* SHA-1 dalam makalah ini bukan untuk menyembunyikan isi pesan, namun untuk melakukan kompresi saja. Dan bila SHA-1 ini tidak digunakan akan tidak mengapa, sehingga cukup dengan algoritma kriptografi kunci publik. Sehingga hal ini menjadikan *hash* dengan SHA-1 menjadi opsional atau tambahan.



Gambar III.3 Ilustrasi Proses *Hash*

(sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/Fungsi-hash-SHA-2021.pdf)

IV. RANCANGAN SOLUSI DAN IMPLEMENTASI

Pada bagian ini akan dipaparkan mengenai rancangan solusi dan implementasi yang akan diterapkan. Secara umum terdapat 3 bagian yakni deskripsi umum solusi, rancangan solusi, dan implementasi solusi.

A. *Deskripsi Umum Solusi*

Solusi yang diusulkan dalam makalah ini merupakan prosedur tanda tangan digital menggunakan algoritma kriptografi kunci publik yang sudah populer. Juga prosedur penerapan fungsi *hash* untuk mengompresi pesan atau dokumen agar memiliki standar dalam tanda tangan digital dan memudahkan proses enkripsi untuk menyusun tanda tangan digital.

Algoritma kriptografi kunci publik yang diusulkan adalah ElGamal, namun tidak menutup kemungkinan penggunaan algoritma lain yang lebih kompleks. Algoritma ini digunakan untuk melakukan tanda tangan digital dengan prosedur enkripsi terhadap isi pesan. Kompleksitas algoritma mempengaruhi anti pemalsuan yang ingin diterapkan. Kedua algoritma ini digunakan dengan mempertimbangkan tingkat kesulitan algoritmanya dan keumuman penggunaan algoritma tersebut.

Untuk kompresi, digunakan fungsi *hash* SHA-1 dengan mempertimbangkan kebutuhan untuk mempercepat proses enkripsi. Kerahasiaan pesan bukan menjadi tujuan utama dari penerapan fungsi *hash* ini, karena fungsi ini ditujukan agar panjang pesan seragam dan memudahkan saat proses enkripsi menjadi tanda tangan digital. Fungsi *hash* SHA-1 dipilih dengan pertimbangan keumuman penggunaannya dan merupakan fungsi yang paling tua, meskipun sudah tidak direkomendasikan penggunaannya karena sudah terdapat banyak celah keamanan.

Fungsi *hash* akan diterapkan terlebih dahulu terhadap pesan atau dokumen, setelah itu diterapkan enkripsi dengan algoritma kriptografi kunci publik. Namun jika *hash* tidak digunakan maka tidak mengapa karena *hash* disini bersifat opsional.

B. *Rancangan Solusi*

Algoritma ElGamal dibuat untuk masing-masing prosedur, yakni pembangkitan kunci, fungsi enkripsi, dan fungsi dekripsi. Fungsi enkripsi dibuat untuk membentuk tanda tangan digital yang diinginkan dan fungsi dekripsi dibuat untuk proses verifikasi apakah tanda tangan digital yang diajukan sesuai. Fungsi *hash* SHA-1 dibuat dengan menggunakan *library* yang tersedia. Dalam hal ini semua prosedur solusi dikembangkan dengan bahasa pemrograman Python. Bahasa pemrograman ini dipilih dengan mempertimbangkan kelengkapan *library*, kemampuan untuk memproses bilangan yang besar, dan kemudahan penulisan kode.

Dalam hal ini pembuatan *user interface* tidak dimasukkan ke dalam makalah dengan pertimbangan hanya untuk mengecek dan membandingkan performa algoritma yang dipakai. Selanjutnya pembuatan *user interface* sangat direkomendasikan untuk kemudahan penggunaan pembangkitan tanda tangan digital dan proses verifikasi.

Berikut merupakan diagram alur proses penandatanganan digital yang akan dilakukan pada setiap teks yang di*input*:

Gambar IV.1 Diagram Alur Proses *Digital Signature*

C. *Implementasi Solusi*

Solusinya akan digunakan penandatanganan melalui *command line* karena dalam makalah ini tidak diimplementasikan *user interface*. Solusi diimplementasikan dengan bahasa pemrograman *Python* untuk setiap fungsi. Disediakan *library* untuk *hash* maupun ElGamal namun pada bagian ini akan diimplementasikan *library hash* SHA-1 dan untuk pembangkitan kunci ElGamal tidak menggunakan *library* agar proses pembangkitan kunci dapat terlihat.

Berikut merupakan *pseudo-code* untuk fungsi *hash*, pembangkitan kunci algoritma El Gamal:

1. *Hash* SHA-1

```
# implementasi library SHA-1
def hashFunction(message):
hashed = sha1(message.encode("UTF-8")).hexdigest()
   return hashed
```

2. Algoritma ElGamal

```
def generateKey(number):
   key = random.randint(pow(10,20),number)
   while gcd(number,key)!=1:
      key = random.randint(pow(10,20),number)
   return key
```

V. PENGUJIAN DAN PEMBAHASAN

Pada bagian ini akan dilakukan pengujian terhadap prosedur tanda tangan digital untuk ayat Al Quran dengan membandingkan ayat yang telah diambil dari *King Fahd Quran Digital Copy.*

A. *Pengujian Solusi*

Pengujian dilakukan pada ayat terakhir yang terdapat pada Al Quran yakni ayat ke 6 dari Surat ke 114, An Naas.

Tabel I. Standar Pengujian

Teks Ayat Asli	مِنَ الجِنَّةِ وَالنَّاسِ
Hasil Hash SHA-1	c9a6d87facab9db44ad974baaa5c316abcc8f7b1
Perubahan Terhadap Teks Ayat (untuk pengujian)	مِنَ الجِنَّةِ وَالنَّسِ

Berikut merupakan hasil dari enkripsi untuk algoritma ElGamal:

Tabel II. Hasil Tanda Tangan dengan Algoritma ElGamal

p	1588722213043011404569829109123810258682292734360
x	673596771935050393524409858934376208548439712186 8
y	708167017659133261380752065603632474142243901374 6
g	802874450785366394132676090965724222449673226809400
Tanda tangan digital untuk teks asli tanpa *hash*	[128861349351051306258294512599998737703172552902908 70, 129744511246915209291840456300061034347867193452399 04, 128941636796129842897707780209053101254175202255896 4, 129583936356758136013013921081867889503372588070371 6, 256919824251317246122456349109031751183895432579008, 126452725998695207075896484327101565035823533222480 50, 128781061905972769618881244990902165280927585580227 76, 126854163224087890272962822372584427147048369835885 20, 129744511246915209291840456300061034347867193452399 04, 128941636796129842897707780209053101254175202255896 4, 129824798691993745931253723909157606770112160775079 98, 129583936356758136013013921081867889503372588070371 6, 126613300888852280354723019545294709880313467867842 38, 129744511246915209291840456300061034347867193452399 04, 256919824251317246122456349109031751183895432579008, 129102211686286916176534315427288454969907454870951 52, 129583936356758136013013921081867889503372588070371 6, 126452725998695207075896484327101565035823533222480 50, 128781061905972769618881244990902165280927585580227 76, 128941636796129842897707780209053101254175202255896 4, 129824798691993745931253723909157606770112160775079 98, 126452725998695207075896484327101565035823533222480 50, 127416175339637646748855695636260434102763141094651 78, 129744511246915209291840456300061034347867193452399 04]
Tanda tangan digital untuk teks yang telah diubah tanpa *hash*	[128861349351051306258294512599998737703172552902908 70, 129744511246915209291840456300061034347867193452399 04, 128941636796129842897707780209053101254175202255896 4, 129583936356758136013013921081867889503372588070371 6, 256919824251317246122456349109031751183895432579008, 126452725998695207075896484327101565035823533222480 50, 128781061905972769618881244990902165280927585580227 76, 126854163224087890272962822372584427147048369835885 20, 129744511246915209291840456300061034347867193452399 04, 128941636796129842897707780209053101254175202255896 4, 129824798691993745931253723909157606770112160775079 98, 129583936356758136013013921081867889503372588070371 6, 126613300888852280354723019545294709880313467867842 38, 129744511246915209291840456300061034347867193452399 04, 256919824251317246122456349109031751183895432579008, 129102211686286916176534315427288454969907454870951 52, 129583936356758136013013921081867889503372588070371 6, 126452725998695207075896484327101565035823533222480 50, 127416175339637646748855695636260434102763141094651 78, 129744511246915209291840456300061034347867193452399 04]
Tanda tangan digital untuk teks asli dengan *hash*	[800996245623594064863648605232582436579666725256580, 416844984967380584775980396600629635362887785592710, 800996245623594064863648605232582436579666725256580, 800996245623594064863648605232582436579666725256580, 416844984967380584775980396600629635362887785592710, 425018416045172360948058443592798843899414997074920, 441365278200755913292214537577137260972469420039340, 449538709278547689464292584569306469508996631521550, 809169676701385841035726652224751645116193936738790, 465885571434131241808448678553644886582051054485970, 792822814545802288691570558240413228043139513774370, 408671553889588808603902349608460426826360574110500, 809169676701385841035726652224751645116193936738790, 441365278200755913292214537577137260972469420039340, 809169676701385841035726652224751645116193936738790, 441365278200755913292214537577137260972469420039340, 825516538856969393379882746209090062189248359703210, 809169676701385841035726652224751645116193936738790, 792822814545802288691570558240413228043139513774370, 400498122811797032431824302616291218289833362628290, 449538709278547689464292584569306469508996631521550, 416844984967380584775980396600629635362887785592710, 833689969934761169551960793201259270725775571185420, 392324691734005256259746255624122009753306151146080, 408671553889588808603902349608460426826360574110500, 825516538856969393379882746209090062189248359703210, 425018416045172360948058443592798843899414997074920, 416844984967380584775980396600629635362887785592710, 792822814545802288691570558240413228043139513774370, 433191847122964137120136490584968052435942208557130, 800996245623594064863648605232582436579666725256580, 833689969934761169551960793201259270725775571185420, 817343107779177617207804699216920853652721148221000, 809169676701385841035726652224751645116193936738790, 809169676701385841035726652224751645116193936738790, 392324691734005256259746255624122009753306151146080, 825516538856969393379882746209090062189248359703210, 400498122811797032431824302616291218289833362628290, 457712140356339465636370631561475678045523843003760, 465885571434131241808448678553644886582051054485970, 800996245623594064863648605232582436579666725256580, 400498122811797032431824302616291218289833362628290, 817343107779177617207804699216920853652721148221000, 457712140356339465636370631561475678045523843003760, 817343107779177617207804699216920853652721148221000, 400498122811797032431824302616291218289833362628290, 833689969934761169551960793201259270725775571185420, 800996245623594064863648605232582436579666725256580, 433191847122964137120136490584968052435942208557130, 833689969934761169551960793201259270725775571185420, 800996245623594064863648605232582436579666725256580, 416844984967380584775980396600629635362887785592710, 425018416045172360948058443592798843899414997074920, 465885571434131241808448678553644886582051054485970, 392324691734005256259746255624122009753306151146080, 425018416045172360948058443592798843899414997074920]
Tanda tangan digital untuk teks yang telah diubah dengan *hash*	[794845706277512730191349330056066980225176494541306, 409465969900536861007664806392519353449333345672794, 393408480884829533125011284573204869003398811366 06, 802874450785366394132676090965724222449673226809400, 417494714408390524948991567302176595673830077940888, 441580947931951516772971850031148322347320274745170, 457638436947658844655625371850462806796313739281358, 401437225392683197066338045482862111224836613404700, 385379736376975869183684523663547626775843148868512, 385379736376975869183684523663547626775843148868512, 810903195293220058074002851875381464674169959077494, 433552203424097852831645089121491080122823542477076, 810903195293220058074002851875381464674169959077494, 786816961769659066250022569146409738000679762273212, 433552203424097852831645089121491080122823542477076, 441580947931951516772971850031148322347320274745170, 794845706277512730191349330056066980225176494541306, 794845706277512730191349330056066980225176494541306,

	39340848088482953312501128457320486900033988113660 6, 81893193980107372201532961278503870689866669134558 8, 39340848088482953312501128457320486900033988113660 6, 81090319529322005807400285187538146467416995907749 4, 79484570627751273019134933005606698022517649454130 6, 40143722539268319706633804548286211122483661340470 0, 43355220342409785283164508912149108012282354247707 6, 78681696176965906625002256914640973800067976227321 2, 44960969243980518071429861094080556457181700701326 4, 81090319529322005807400285187538146467416995907749 4, 44960969243980518071429861094080556457181700701326 4, 79484570627751273019134933005606698022517649454130 6, 45763843694765884465562537185046280679631373928135 8, 43355220342409785283164508912149108012282354247707 6, 81893193980107372201532961278503870689866669134558 8, 78681696176965906625002256914640973800067976227321 2, 45763843694765884465562537185046280679631373928135 8, 38537973637697586918368452366354762677584314886851 2, 45763843694765884465562537185046280679631373928135 8, 80287445078536639413267609096572422244967322680940 0, 40946596990053686100766480639251935344933345672794, 78681696176965906625002256914640973800067976227321 2, 40946596990053686100766480639251935344933345672794, 43355220342409785283164508912149108012282354247707 6, 81893193980107372201532961278503870689866669134558 8, 42552345891624418889031832821183383789832681020898 2, 41749471440839052494899156730217659567383007794088 8, 38537973637697586918368452366354762677584314886851 2, 40946596990053686100766480639251935344933345672794, 77878821726180540230869580823675249577618303000511 8, 81090319529322005807400285187538146467416995907749 4, 77878821726180540230869580823675249577618303000511 8, 43355220342409785283164508912149108012282354247707 6, 45763843694765884465562537185046280679631373928135 8, 79484570627751273019134933005606698022517649454130 6, 77878821726180540230869580823675249577618303000511 8, 79484570627751273019134933005606698022517649454130 6, 78681696176965906625002256914640973800067976227321 2]

B. *Pembahasan*

Berdasarkan pengujian pada bagian sebelumnya, diterapkan dua prosedur yakni dengan menggunakan *hash* sebelum membentuk tanda tangan digital dan tanpa *hash.* Dari hasil tersebut didapat bahwa algoritma ElGamal berjalan dengan baik dan dapat diterapkan ke teks yang ingin ditanda tangani. Tanda tangan dari teks asli seharusnya disimpan oleh Lembaga yang bertanggung jawab beserta *private key*. Untuk *public key* dapat disebar ke masyarakat luas untuk digunakan proses penandatanganan.

Dari hasil pengujian tersebut didapat juga hasil bahwa tanpa menggunakan fungsi *hash*, hasil tidak berubah dengan banyak, hanya berubah di bagian yang diubah seperti yang ditunjukkan dengan *highlight* warna hijau. Teks yang telah diubah merupakan teks yang dihilangkan satu huruf padanya sehingga hasil enkripsinya pun ada yang hilang sejumlah huruf tersebut. Teknik ini dapat digunakan untuk mengetahui letak perubahan yang dilakukan pada salinan yang diuji kemudian dicocokkan dengan tanda tangan digital yang dibangkitkan dari dokumen yang asli.

Untuk pengujian dengan penggunaan *hash* sebelum dilakukan proses enkripsi, pengujian ini juga berjalan dengan baik dan akan sangat membantu jika dokumen memiliki ukuran yang cukup besar atau berisi teks yang sangat panjang. Karena pengujian dilakukan terhadap teks yang tidak begitu panjang, tidak terlalu signifikan kompresi dengan fungsi *hash* yang dilakukan. Metode ini sangat efektif untuk mendeteksi adanya perubahan sekecil apapun dalam teks atau dokumen karena fungsi *hash* akan berubah total akibat perubahan tersebut. Tidak seperti metode sebelumnya yang tanpa *hash*, perubahan hanya terlihat pada bagian yang berubah saja. Sementara metode ini mengubah keseluruhan tanda tangan digital.

Sehingga penggunaan kedua metode ini dapat disesuaikan dengan kebutuhan penggunaan. Keduanya bisa digunakan untuk berbagai masalah yang terdapat pada salinan Al Quran digital.

VI. KESIMPULAN DAN SARAN

Solusi yang diajukan yakni dua metode penandatanganan digital dapat membantu mengatasi permasalahan terhadap pemalsuan salinan Al Quran digital, di mana Al Quran merupakan kitab suci yang dijaga keasliannya. Metode penandatanganan digital tanpa *hash* dapat digunakan untuk mendeteksi letak perubahan sementara metode dengan fungsi *hash* dapat menyangkal pemalsuan yang telah dilakukan karena hasil tanda tangan akan sangat berubah. Fungsi *hash* juga akan membantu dalam pemrosesan enkripsi untuk teks atau ayat yang sangat panjang sehingga ukuran dapat dikompresi dan proses enkripsi dapat berjalan lebih cepat.

Fungsi *hash* yang digunakan tidak terbatas pada SHA-1 saja karena SHA-1 telah ditemukan celah keamanan padanya. Kedepannya dapat digunakan fungsi *hash* yang lebih kini, meskipun sebenarnya fungsi *hash* diterapkan hanya untuk proses kompresi dan pengacakan, bukan untuk menyembunyikan isi pesan.

Kedepannya diharapkan solusi ini dapat dikembangkan dan diperbaiki seperti aspek *user interface*, penggunaan fungsi *hash*, dan pemilihan nilai variabel yang lebih besar dan kompleks agar proses penandatanganan menghasilkan tanda tangan yang lebih kompleks.

VII. UCAPAN TERIMA KASIH

Dengan ini penulis memanjatkan syukur sebesar-besarnya karena telah diberikan kesempatan oleh Allah untuk mempelajari topik yang diangkat dalam makalah ini dan dimudahkan dalam proses pembuatannya. Selain untuk memenuhi tugas mata kuliah II4031 – Kriptografi dan Koding, pembuatan makalah ini sangat membantu penulis dalam memahami materi.

Penulis juga ingin mengucapkan terima kasih kepada orangtua dan keluarga yang selalu mendukung setiap langkah dalam mencari ilmu juga atas doa yang senantiasa mereka panjatkan untuk kelancaran studi penulis. Penulis juga ingin berterima kasih kepada dosen mata kuliah II4031 – Kriptografi dan Koding, Bapak Rinaldi Munir yang telah memberikan kesempatan dan ilmu sehingga penulis dapat menyelesaikan pembuatan makalah ini.

Juga untuk teman-teman jurusan Sistem dan Teknologi Informasi angkatan 2018 yang telah membantu penulis dalam menjalani kehidupan di kampus, yang namanya tidak dapat disebutkan satu persatu.

VIDEO LINK AT YOUTUBE

https://youtu.be/5Z5SNXKa2Jw

REFERENCES

[1] M. Almazrooie, A. Samsudin, A. A.-A. Gutub, M. S. Salleh, M. A. Omar, and S. A. Hassan, "Integrity verification for digital Holy Quran verses using cryptographic hash function and compression," Journal of King Saud University - Computer and Information Sciences, vol. 32, no. 1, pp. 24–34, Jan. 2020, doi: 10.1016/j.jksuci.2018.02.006.

[2] M. Almazrooie, A. Samsudin, A. A.-A. Gutub, M. S. Salleh, M. A. Omar, and S. A. Hassan, "Integrity verification for digital Holy Quran verses using cryptographic hash function and compression," Journal of King Saud University - Computer and Information Sciences, vol. 32, no. 1, pp. 24–34, Jan. 2020, doi: 10.1016/j.jksuci.2018.02.006.

[3] Munir, Rinaldi. 2021. Slide Kuliah II4031 Kriptografi dan Koding: Tanda Tangan Digital.

[4] M. F. Hilmi, M. F. Haron, O. Majid, and Y. Mustapha, "Authentication of Electronic Version of the Holy Quran: An Information Security Perspective," presented at the 2013 Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences, Dec. 2013, doi: 10.1109/nooric.2013.24.

[5] Munir, Rinaldi. 2017. Slide Kuliah IF4020 Kriptografi: Algoritma ElGamal.

[6] Munir, Rinaldi. 2021. Slide Kuliah II4031 Kriptografi dan Koding: Secure Hash Algorithm (SHA).

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.

Tangerang Selatan, 25 Mei 2021

Khairunnisa Rifdah 18218008