Implementasi Kriptografi Kunci Publik RSA dan Fungsi *Hash* SHA3 sebagai *Digital Signature* pada Pembayaran Digital

Pandyaka Aptanagi
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517003@std.stei.itb.ac.id

***Abstract*—Teknologi sudah tidak bisa lagi lepas dari kehidupan manusia modern. Saat ini, hampir semua aspek kehidupan sangat dipengaruhi oleh teknologi. Salah satunya adalah teknologi untuk melakukan pembelian dan penjualan secara digital. Untuk mendukung proses tersebut, diperlukan mekanisme pembayaran secara digital yang aman, baik untuk pembeli maupun penjual. Salah satu metode untuk mewujudkan pembelian digital yang aman adalah dengan menggunakan tanda tangan digital untuk setiap proses pembelian. Pada makalah ini akan dibahas mengenai implementasi kriptografi kunci publik RSA dan fungsi *hash* SHA3 sebagai *digital signature* untuk mewujudkan keamanan pembayaran digital.**

***Keywords*—algoritma RSA, kunci publik, fungsi *hash*, algoritma SHA3, pembayaran digital, tanda tangan digital.**

I. PENDAHULUAN

Saat ini, teknologi sudah menjadi aspek yang tidak terpisahkan dalam kehidupan modern. Dengan adanya teknologi, hal seperti sistem pencarian, sistem prediksi saham, sistem robot, hingga sistem untuk melakukan pembelian dan penjualan secara digital dapat diciptakan. Salah satu teknologi yang sangat memudahkan kehidupan adalah sistem pembelian dan penjualan secara digital. Dengan adanya sistem pembelian dan penjualan secara digital, kehidupan manusia sangat dimudahkan karena dapat melakukan penjualan maupun pembelian hanya dengan beberapa sentuhan.

Namun, sistem pembelian dan penjualan secara digital juga memunculkan beberapa persoalan, apalagi komponen yang digunakan merupakan sesuatu yang sensitif dan berharga yaitu uang. Baik proses penjualan maupun proses pembelian harus dipastikan keamanannya untuk memastikan bahwa transaksi akan berjalan dengan tepat. Salah satu cara untuk memastikan transaksi akan berjalan dengan tepat adalah memastikan autentikasi dari pembeli dan penjual.

Oleh karena itu, pada makalah ini akan dibahas mengenai implementasi kriptografi kunci publik RSA dan fungsi *hash* SHA3 sebagai salah satu metode untuk meningkatkan keamanan pembayaran secara digital. Dengan menggunakan *digital signature* pada pembayaran digital, maka transaksi yang berlangsung akan dijamin autentik, asli, dan tidak dapat disangkal sesuai dengan prinsip dari tanda tangan digital.

II. DASAR TEORI

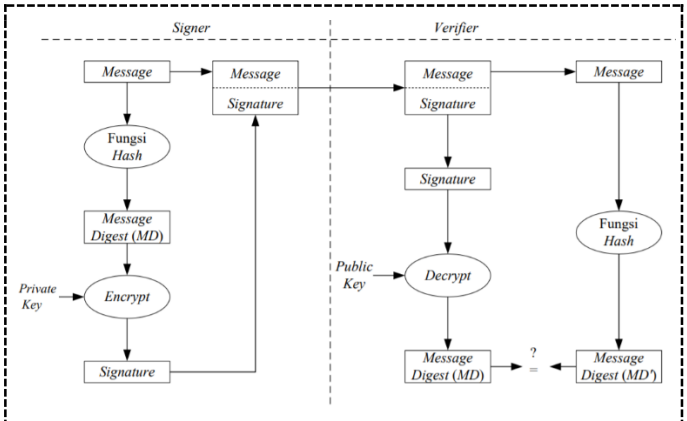
A. Tanda Tangan Digital

Tanda tangan digital (*digital signature*) merupakan tanda tangan yang digunakan untuk data digital. Fungsi tanda tangan secara umum merupakan tanda pada dokumen untuk memastikan autentikasinya. Tanda tangan digital digunakan untuk menyelesaikan aspek keamanan autentikasi (*authentication*), keaslian pesan (*data integrity*), dan anti-penyangkalan (*nonrepudiation*).

Tanda tangan digital memiliki beberapa karakteristik yaitu:

1. Tanda-tangan adalah bukti yang autentik
2. Tanda tangan tidak dapat dilupakan
3. Tanda tangan tidak dapat dipindah untuk digunakan ulang
4. Dokumen yang telah ditandatangani tidak dapat diubah
5. Tanda tangan tidak dapat disangkal

Dalam menandatangani pesan, terdapat dua metode yang dapat dilakukan yaitu dengan cara mengenkripsi pesan dan menggunakan kombinasi fungsi *hash* dan kriptografi kunci-publik. Metode penandatanganan pesan dengan menggunakan kombinasi fungsi *hash* dan kriptografi kunci-publik dapat dilihat pada gambar II.1.



Gambar III.1 Metode Penandatanganan dengan Menggunakan Kriptografi kunci-publik dan Fungsi *Hash*

(sumber: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Tanda-tangan-digital-2020.pdf)

Pada makalah ini, metode tanda tangan digital yang digunakan adalah metode tanda tangan dengan menggunakan kriptografi kunci-publik dan fungsi *hash* dengan pertimbangan lebih sesuai dan lebih aman untuk proses bisnis yang dijalankan.

B. Algoritma Kunci Publik RSA

Algoritma RSA merupakan algoritma yang ditemukan oleh tiga peneliti dari MIT yaitu Ronald Rivest, Adi Shamir, dan Len Adleman pada tahun 1976. Keamanan dari algoritma RSA terletak pada sulitnya memfaktorkan bilangan bulat yang besar menjadi faktor-faktor prima.

Algoritma RSA memiliki persamaan enkripsi seperti ditunjukkan pada persamaan (1) dan persamaan dekripsi seperti ditunjukkan pada persamaan (2).

$$\text{Enkripsi: } E_e(m) = c = m^e \bmod n \quad (1)$$

$$\text{Dekripsi: } D_d(c) = m = c^d \bmod n \quad (2)$$

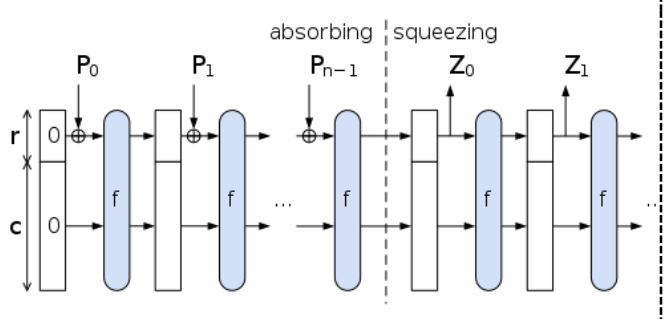
Pada makalah ini, algoritma RSA digunakan untuk melakukan enkripsi pada *message digest* yang dihasilkan dari *payload* yang di-*hash* untuk kemudian menjadi *signature*. Panjang kunci yang dihasilkan oleh algoritma RSA pada makalah ini sebesar 1024-bit dengan pertimbangan keamanan dan juga ukuran *payload* yang besar.

C. Fungsi *Hash* SHA3

Fungsi *Hash* SHA3 atau yang lebih dikenal sebagai Keccak merupakan fungsi *hash* yang dikembangkan oleh Guido Breton, Joan Daemen, Michaël Peeters, dan Gilles Van Assche dan pertama kali dipublikasikan pada tahun 2015. Keccak dapat digunakan untuk melakukan autentikasi, enkripsi, dan sebagai pembangkit *pseudo-random*. Algoritma SHA3 menggunakan mekanisme *sponge construction*, dimana data akan diserap ke dalam suatu *sponge*, untuk kemudian akan diperas dan dihasilkan *message digest*. Konstruksi spons pada algoritma SHA3 dapat dilihat pada gambar III.2.

Secara umum, algoritma SHA3 adalah sebagai berikut:

- Masukkan pesan M pada *padding function*, sehingga dihasilkan *string* P yang memiliki ukuran kelipatan nilai *rate*
- Potong pesan P menjadi blok-blok berukuran *rate*-bit
- Inisialisasi *state* S dengan nilai nol sepanjang panjang blok (b)
- Fase penyerapan. Untuk setiap blok P:
    - Lakukan operasi XOR dengan r-bit pertama dari *state* S
    - Masukkan hasil ke dalam fungsi permutasi f sehingga dihasilkan *state* baru S
- Inisialisasi *string* kosong Z
- Selagi panjang Z belum sama dengan panjang luaran (d), sambungkan r-bit pertama dari *state* S
- Jika panjang Z masih belum sama dengan d, masukkan ke dalam fungsi permutasi *f* sehingga menghasilkan *state* baru S



Gambar III.2 Konstruksi Spons pada SHA3
(sumber: https://en.wikipedia.org/wiki/File:SpongeConstruction.svg)

Pada makalah ini, algoritma SHA3 digunakan untuk melakukan *hashing* terhadap *body payload* dari *request* untuk kemudian dijadikan *signature* dan dilakukan validasi terhadap autentikasi *request*.

III. Rancangan Solusi dan Implementasi

Dalam menyelesaikan permasalahan, terdapat langkah-langkah penyelesaian yaitu Deskripsi Umum, Rancangan, dan Implementasi Solusi.

A. Deskripsi Umum Solusi

Dalam menyelesaikan permasalahan autentikasi pembayaran digital, digunakan pendekatan tanda tangan digital (*digital signature*) dengan metode kombinasi fungsi *hash* dan kriptografi kunci publik. Dengan menggunakan tanda tangan digital, *request* pembayaran digital dapat dipastikan autentik (*authentication*), asli (*data integrity*), dan anti-penyangkalan (*nonrepudiation*).

Fungsi *hash* yang digunakan adalah fungsi *hash* SHA3 dengan ukuran 256-bit (SHA3-256) dengan pertimbangan bahwa SHA3 merupakan fungsi *hash* yang menjadi standar dan masih belum ditemukan kolisi. Panjang *message digest* yang dipilih adalah 256-bit dengan pertimbangan keamanan dan kecepatan pemrosesan.

Algoritma kriptografi kunci publik yang digunakan adalah algoritma kunci publik RSA dengan pertimbangan keamanan dan penggunaan yang sudah popular. Panjang kunci yang digunakan dalah 1024-bit dengan pertimbangan keamanan dan kecepatan pembangkitan dan pemrosesan pasangan kunci.

Solusi yang diimplementasikan memiliki dua sudut pandang yaitu sudut pandang pembeli (*end-user*) sebagai pihak yang

melakukan pembayaran dan sudut pandang *store* sebagai pihak yang menerima pembayaran.

B. Rancangan Solusi

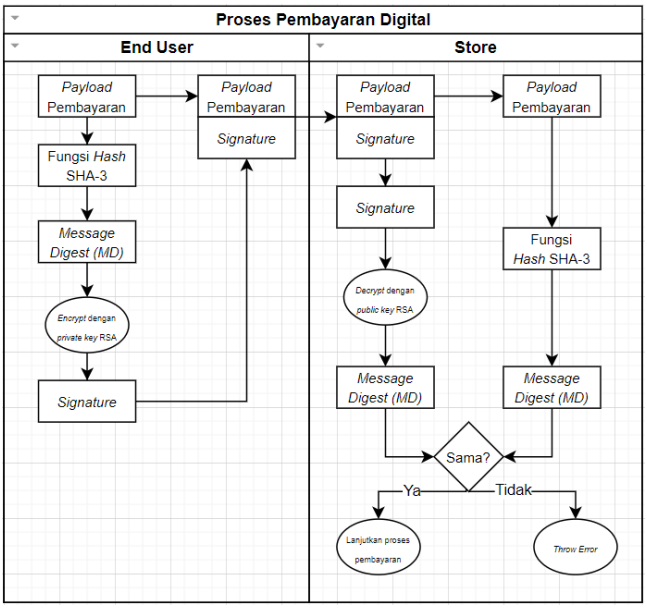
Solusi yang dirancang terdiri dari dua tahap pengguna yaitu tahap pengguna *end user* atau yang melakukan pembayaran dan tahap *store* atau yang menerima pembayaran. Arsitektur rancangan solusi dapat dilihat pada ambar III.1.

Perancangan solusi untuk tahap pengguna *end user* dimulai dengan *end user* mengirimkan *request* untuk melakukan pembayaran. *Request* yang dikirim oleh *end user* akan diproses terlebih dahulu sebelum diterima oleh *store*. Tahapan proses yang akan dilalui adalah sebagai berikut:

- *End-user* membuat pasangan kunci publik dan kunci privat sesuai dengan ketentuan dari *store*.
- Payload pada *request* akan di-*hash* dengan menggunakan fungsi *hash* SHA3 dengan ukuran 256-bit dan menghasilkan *message digest* untuk *payload* tersebut.
- *Message digest* yang dihasilkan akan di-*encrytpt* dengan menggunakan algoritma RSA, sesuai dengan kunci privat yang dimiliki oleh pengguna dan menghasilkan *signature*.
- *Signature* di-*append* ke dalam *payload* pembayaran dan dikirimkan ke *store* sebagai satu *request* yang sama.

Sedangkan perancangan untuk tahap *store* dimulai dari menerima *request payload* pembayaran dari *end user* dan kemudian dilakukan proses validasi sebelum proses pembayaran dimulai. Tahapan proses validasi yang akan dilalui adalah sebagai berikut:

- *Store* menerima *request payload* pembayaran dari *end user*
- *Signature* akan dipisahkan dari *payload body*
- Dilakukan *decryption* pada *signature* dengan menggunakan algoritma RSA, sesuai denan kunci public yang dimiliki oleh pengguna dan *store*
- Bersamaan dengan hal tersebut, dilakukan *hashing* pada *payload* pembayaran
- Dilakukan perbandingan antara *message digest* yang dihasilkan dari *hashing payload* dengan *decryption signature*
- Apabila hasilnya sama, maka proses pembayaran akan dilanjutkan
- Apabila hasilnya berbeda, maka akan dikirimkan *error signature is not valid* kepada *end user* dan proses pembayaran akan di-*terminate*



Gambar III.1 Arsitektur Rancangan Solusi

C. Implementasi Solusi

Solusi diimplementasikan sebagai sebuah *representational state transfer application programming interface* (REST API) untuk memfasilitasi fitur pembayaran dari suatu *store*. Pihak yang ingin melakukan pembayaran harus membangkitkan pasangan kunci publik dan privat terlebih dahulu sebelum mengirimkan *request* pembayaran. Setelah itu, pengguna menggunakan kunci publik yang sudah dibangkitkan untuk membuat *signature* yang akan dilampirkan pada *request payload*. Kemudian, *store* akan melakukan validasi terhadap *request payload* yang dikirimkan pengguna untuk kemudian ditentukan apakah *payload* tersebut autentik atau tidak.

Terdapat beberapa *endpoint* dalam API yang dibangun, yaitu *endpoint* untuk pembangkitan kunci, pembangkitan *signature*, dan pembayaran.

1. Pembangkitan kunci (`/generate`)

   Endpoint untuk pembangkitan kunci menerima *request* dengan HTTP *method* POST dengan *request payload* berupa identitas pengguna *response payload* berupa pasangan kunci publik dan kunci privat RSA.

   Request payload (JSON)

```
{
    "id_user": 105
}
```

   Response payload (JSON)

```
{
    "public_key": "-----BEGIN PUBLIC KEY-
----
```

```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCG
/wrWAdQTOcOl8VF/XjXaRVhR
zG7Szj1m94/WM+2et64Z0Po1l+RKBq0Us6x2MkMj
O+/NC33Eb1I/diShP2jYfzxY
+IrAvW9ATlg/yHob/sn+8QNDEK3UNYAc7LxIjwx3
xq1acijAStr9d6TQEavWJQJk
iq3Lnb/EvALftl5pCQIDAQAB
-----END PUBLIC KEY----
"private_key": "-----BEGIN RSA PRIVATE
KEY-----
MIICWwIBAAKBgQCG/wrWAdQTOcOl8VF/XjXaRVhR
zG7Szj1m94/WM+2et64Z0Po1
l+RKBq0Us6x2MkMjO+/NC33Eb1I/diShP2jYfzxY
+IrAvW9ATlg/yHob/sn+8QND
EK3UNYAc7LxIjwx3xq1acijAStr9d6TQEavWJQJk
iq3Lnb/EvALftl5pCQIDAQAB
AoGAEu+prHVbM0syB4//N5eWhESf6/F6RjXJ5Syq
z4g/N7laQaLPxj7FNiETnEjL
01SfmEyrCZ8oCCr5ZY+D+d50LV2YBmslYReVNGRS
TebH+VLbkQBHB3LFCcbIH4Op
OTovNXgM9Ov4pnqJBwQ+SKbpB/2HnN4ww9AiyPSy
UpQy9pECQQDV+CD1GWyolO72
Jkib8ca05Hzc3djNKl7DuBBixq0VGOdFhR703UzR
EATp26vaU+zFJkPTpKGHVxrb
HKRLT86dAkEAoYOZTputccVZdhVxlpQU8DQY2SMO
Qw3tlSZ4iWHE7fqXdYhyC1Pn
dFk6Lq3MYWbQC7BI/NTPudPtwbsBZdWiXQJAPCez
VFUeiFK+z/M5bKZPoCGwvQOd
5ShpVgsUiECk/o5uzsu7RUpGQ2yGhRzeNtZrdY3p
ecIKK5ugkEOAKTA5vQJAf+Ln
nOp/ZzpQH8PS8NvHyHo3wpaTUFzS5epUPtN/Tr/A
9XtYcce52NJd02AiGG2EBf1L
jYYkBIveEiu0a6HAsQJAWPRYXyDYHCRdiWNgSeAG
hQYgBBqPVcOfBQs9vPTHdbUw
ootYAB1SzBo+TmnGpiMAkLUoByltYdZp/7EZtIco
FQ==
-----END RSA PRIVATE KEY-----
}
```

2. Pembangkitan *signature* (`/signature`)

Endpoint untuk pembangkitan *signature* menerima *request* dengan HTTP *method* POST dengan *request payload* berupa permintaan pembayaran dan *response payload* berupa *signature* yang berhasil dibangkitkan.

Request payload (JSON)

```
{
    "id_user": 105,
    "id_order": 5,
    "total_payment": 10000,
    "va_number": 999102349,
}
```

Response payload (JSON)

```
{
    "signature": "34e96bab82254324631451
9b5c7b35ccebc2cbde0f9e3da1acc566dcab963e
21"
}
```

3. Pembayaran (`/pay`)

Endpoint untuk pembayaran menerima *request* dengan HTTP *method* POST dengan *request payload* berupa permintaan pembayaran dan *signature* yang sesuai dan *response payload* berupa status dari pembayaran. Apabila pembayaran berhasil maka status yang dikembalikan adalah "PAID", namun apabila tidak berhasil maka akan dikembalikan sesuai dengan *error* yang dihasilkan.

Request payload (JSON)

```
{
    "id_user": 105,
    "id_order": 5,
    "total_payment": 10000,
    "va_number": 999102349,
    "encrypted_signature": "MW6+yqTcyxRr
M2YWnSxhqbUlwvlWuPFZWX0GVm9th8FIrasnCJO3
ZgfC5PCzR/XjZLh9NQssJ2wTDNZB+Gpa/SOnFLxh
MMvtrQ3LCrfatedfkN39wQHC7J6LMECv134V0Kue
97a9TdVL2aA+7NF44zBGshtVJ9nY5V0epwgBpeM=
"
}
```

Success response payload (JSON)

```
{
    "id_user": 105,
    "id_order": 5,
    "total_payment": 10000,
    "va_number": 999102349,
    "status": "SUCCESS"
}
```

Failed request payload (JSON)

```
{
    "id_user": 10,
    "id_order": 5,
    "total_payment": 10000,
    "va_number": 999102349,
    "status": "PENDING",
    "error_code": 400,
    "error_message": "Signature is not v
alid"
}
```

IV. PENGUJIAN DAN PEMBAHASAN

A. Pengujian

Terdapat tiga pengujian yang dilakukan yaitu pengujian untuk pembayaran dengan *signature* yang valid, pembayaran dengan *signature* yang tidak valid, dan pembayaran dengan *signature* yang valid namun *request payload* yang dikirim tidak valid.

Lingkungan pengujian yang digunakan ditunjukkan pada tabel berikut:

id_user	1
Id_order	1
total_payment	15000
va_number	9998765412
public_key (hasil pembangkitan)	-----BEGIN PUBLIC KEY----- MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC52slERec7jfUxr22BqyCDl1Rx IrUy3QBg8qcLdKWlyTooiqHBdD6WzXiS+ZZl2yC+TcqkCE+m8Iyh40dNKPvlG6Mz YzaqIXPoAxIvaFVSH5ulqzAfvBh2sIYgibglUbUcP5hIac0/hyoGc0VU66NbCMwH gajPJkj8t7Rmb29YoQIDAQAB -----END PUBLIC KEY-----
private_key (hasil pembangkitan)	-----BEGIN RSA PRIVATE KEY----- MIICXQIBAAKBgQC52slERec7jfUxr22BqyCDl1RxIrUy3QBg8qcLdKWlyTooiqHB dD6WzXiS+ZZl2yC+TcqkCE+m8Iyh40dNKPvlG6MzYzaqIXPoAxIvaFVSH5ulqzAf vBh2sIYgibglUbUcP5hIac0/hyoGc0VU66NbCMwHgajPJkj8t7Rmb29YoQIDAQAB AoGAahwgtwsWBul4qSJYaiTyzk6TFnjTbTPCaJeBicfSX0HxI35q9dFRfrXW9krr dEHUgvJVFKvuFgjWsgLaDZKtpVeVrwQhcHb1hEySiof07zUSjiIU8kzknNvpYXaZ 1gWKELnm4dHpCc92+UU75mCy6EGmZd23fab/KgFOQM4XQSUCQQD/sqCCS007Uq1d +s0x0D0ckDLDAV7YeyYqT28hm993oq/c04LBYHtQGXonryLFRq+F0cSOHTPY3dCQ fau8UHZXAkEAuhMGZaB5pwAj7DFJOShY2TAq3X0DqpFumOdhUsvQ1XQBnN15EnXQ MRDbk6S7BeOJOTfo2AuwjLvn46RuHDCdxwJAcQRcC5sK9OiTNlHPdnALQt+9gBHQ ATFVaigX/6vstu1Se/GE/sINXxcTtgyj+rxGgoh4LzRbIUTRlbvmN2L9GwJBALZb ltVaZF8KgComU0C1Df4sepX+wu0TAnTPrk3aeZv3yOzXEKTK2XJ8tvUHaFLpH773 0DaLnkcOkMBEkbUG4FkCQQCM2mqBLiqMP03D+FWJwkqC5qi/CEvMXu+Tqi0k1vkS 457VZ71ziWKPh7/4Ow1+m4rwfqYhw6w9urnvtk0AwY7f -----END RSA PRIVATE KEY-----
valid signature (hasil pembangkitan)	539eafe4f43ae1927a589b5a3e15eed9b445e2de0bfc664cefb88e5a17646d42
invalid signature (hasil modifikasi valid signature)	539eafe4f43ae1927a589b5a3e15eed9b445e2de0bfc664cefb88e5a17646d41
encrypted valid signature	C8/HrUn82aQFxEi56jiWRhJxPbMywxPbJCByLxH14aVuUEPJRPbDNCaAyuZ87H/10tn5Y69RLCMDkezEQBNERQeT7+CfYvKoBiODFwv5FYBouMVtNMXjxa64pwhBAut2q+1MGuyVNblDh7oHPwMzD91cQY14hBjfLsxum0yq39U=
encrypted invalid signature	V8/L3bZD+QrNaNSezMux7aBUzNJBVYb1Y+Ew4S8en2V0yOQLINTgB0RcLiWM4aIuNRP7XqnUpuDpGwHLBRBglPM/VZeHGgiBIWAcQnX1hDn3rlCBVkYXkneSiL12/6QU9UdWcIMQ8vOG4iHbM0iQmpqmiMyz33971cY+cdyVuxg=

1. Pembayaran dengan *signature* yang valid

Pada pengujian ini, dikirimkan *request* untuk pembayaran dengan *signature* yang valid milik pengguna.

Request payload (JSON)


```
{
    "id_user": 1,
    "id_order": 1,
    "total_payment": 15000,
    "va_number": 9998765412,
    "encrypted_signature": "C8/HrUn82aQF
xEi56jiWRhJxPbMywxPbJCByLxH14aVuUEPJRPbD
NCaAyuZ87H/10tn5Y69RLCMDkezEQBNERQeT7+Cf
YvKoBiODFwv5FYBouMVtNMXjxa64pwhBAut2q+1M
GuyVNblDh7oHPwMzD91cQY14hBjfLsxum0yq39U=
"
}
```


Response payload (JSON)


```
{
    "id_user": 1,
    "id_order": 1,
    "total_payment": 15000,
    "va_number": 9998765412,
    "status": "SUCCESS"
}
```


2. Pembayaran dengan *signature* yang tidak valid

Pada pengujian ini, dikirimkan *request* untuk pembayaran dengan *signature* yang merupakan modifikasi dari *signature* asli dengan hanya berbeda satu angka saja.

Request payload (JSON)


```
{
    "id_user": 1,
    "id_order": 1,
    "total_payment": 15000,
    "va_number": 9998765412,
    "encrypted_signature": "V8/L3bZD+QrN
aNSezMux7aBUzNJBVYb1Y+Ew4S8en2V0yOQLINTg
B0RcLiWM4aIuNRP7XqnUpuDpGwHLBRBglPM/VZeH
GgiBIWAcQnX1hDn3rlCBVkYXkneSiL12/6QU9UdW
cIMQ8vOG4iHbM0iQmpqmiMyz33971cY+cdyVuxg=
"
}
```


Response payload (JSON)


```
{
    "id_user": 1,
    "id_order": 1,
    "total_payment": 15000,
    "va_number": 9998765412,
    "status": "PENDING",
    "error_code": 400,
    "error_message": "Signature is not v
alid"
}
```


3. Pembayaran dengan *signature* yang valid namun *request payload* tidak valid

Pada pengujian ini, dikirimkan *request* untuk pembayaran dengan *signature* yang valid namun ada *field* pada *request payload* yang diubah nilainya.

Request payload (JSON)


```
{
    "id_user": 2,
    "id_order": 1,
    "total_payment": 15000,
    "va_number": 9998765412,
    "encrypted_signature": "C8/HrUn82aQF
xEi56jiWRhJxPbMywxPbJCByLxH14aVuUEPJRPbD
NCaAyuZ87H/10tn5Y69RLCMDkezEQBNERQeT7+Cf
YvKoBiODFwv5FYBouMVtNMXjxa64pwhBAut2q+1M
GuyVNblDh7oHPwMzD91cQY14hBjfLsxum0yq39U=
"
}
```


Response payload (JSON)

```
{
    "id_user": 2,
    "id_order": 1,
    "total_payment": 15000,
    "va_number": 9998765412,
    "status": "PENDING",
    "error_code": 400,
    "error_message": "Signature is not v
alid"
}
```


B. Pembahasan

Berdasarkan pengujian yang telah dilakukan, didapatkan hasil sebagai berikut:

1. Pembayaran dengan *signature* yang valid

   Pengujian pembayaran dengan *signature* yang valid menghasilkan *response* sukses. Hal ini menandakan *request payload* yang dikirimkan dapat dibuktikan validitas dan autentikasinya sehingga proses pembayaran dapat dijalankan dengan baik.

2. Pembayaran dengan *signature* yang tidak valid

   Pengujian pembayaran dengan *signature* yang tidak valid menghasilkan *response* kegagalan. Hal ini dikarenakan pada saat proses validitas, *message digest* yang dibandingkan tidak sama sehingga pengguna akan mendapatkan *response* kegagalan.

3. Pembayaran dengan *signature* yang valid namun *request payload* tidak valid

   Pengujian pembayaran dengan signature *yang valid* namun request payload tidak valid menghasilkan *response* kegagalan. Hal ini dikarenakan pada saat proses pembangkitan *signature*, semua komponen atau *field* menjadi parameter pembangkitan. Namun, ketika dilakukan validasi, *signature* yang dibangkitkan memiliki komponen yang hilang sehingga hasilnya tidak sama dengan *signature* yang dikirimkan.

V. Kesimpulan dan Saran Pengembangan

. Solusi yang diimplementasikan berhasil untuk memastikan pembayaran digital memenuhi tiga aspek, yaitu autentik, asli, dan anti-penyangkalan. Hal ini membuat pembayaran digital menjadi lebih aman baik untuk pihak yang melakukan pembayaran maupun pihak yang menerima pembayaran.

Kedepannya, solusi dapat dikembangkan lebih lanjut pada beberapa bagian seperti ukuran *message digest* yang digunakan, ukuran pasangan kunci public dan privat RSA, penggunaan jenis algoritma kunci public yang lain seperti Elgamal dan AES, penggunaakn algoritma fungsi *hash* yang lain seperti Keccak dan MD5, serta implementasi dengan menggunakan bahasa pemrograman lain yang dapat melakukan pemrosesan dengan lebih cepat seperti WebAsssembly.

Ucapan Terimakasih

Ucapan terimakasih penulis nyatakan kepada Tuhan Yang Maha Esa, karena karunia-Nya penulis bisa diberikan kesempatan untuk menyelesaikan dan bisa memberikan kontribusi nyata dalam memberikan ide yang dituliskan pada makalah ini.

Penulis juga mengucapkan terimakasih kepada Dr. Rinaldi Munir atas dedikasinya dalam memberikan ilmu pengetahuan tentang kriptografi kepada penulis.

Referensi

[1] Munir, Rinaldi. 2020. Slide Kuliah IF4020 Kriptografi: Fungsi Hash

[2] Munir, Rinaldi. 2020. Slide Kuliah IF4020 Kriptografi: Tanda-tangan Digital

[3] Munir, Rinaldi. 2020. Slide Kuliah IF4020 Kriptografi: Algoritma RSA

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, 19 Desember 2020

Pandyaka Aptanagi
13517003