Implementasi Digital Signature pada Bukti Transfer menggunakan Kriptografi Kunci Publik RSA, Fungsi Hash SHA-256, dan Steganografi.

Inka Anindya Riyadi - 13518038
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13518038@std.stei.itb.ac.id

***Abstract*—Perkembangan industri dan pandemic yang terjadi pada awal 2020 menyebabkan meningkatkanya pembelian dan pembayaran yang dilakukan secara daring. Untuk mengatasi penipuan pada transaksi pembayaran, dibuatlah solusi untuk melakukan verifikasi pembayaran dengan mengunggah citra digital bukti transfer untuk menentukan apakah bukti pembayaran dari suatu transaksi merupakan pembayaran yang valid. Menggunakan *digital signature* yang dienkripsi menggunakan algoritma RSA dan fungsi hash SHA-256 dan yang nantinya akan disisipkan menggunakan steganografi metode *Least Significant Byte* menjamin citra digital bukti transaksi autentik, asli dan anti-penyangkalan.**

***Keywords*—pembayaran digital, tanda tangan digital, algoritma RSA, kunci publik, fungsi hash, SHA-256**

I. PENDAHULUAN

Dengan pesatnya perkembangan industri dalam beberpa dekade terakhir, banyak sektor yang harus mengikuti perkembangan teknologi, salah satunya sektor finansial. Pandemi yang dimulai pada awal tahun 2020 membuat masyarakat harus mengurangi aktivitas di luar rumah termasuk mengurangi frekuensi pembelian barang pokok untuk memenuhi kegiatan sehari-hari. Kedua hal tersebut mendukung pembelian secara daring (*online shopping*) yang dapat dilakukan via *marketplace* ataupun sosial media. Untuk memenuhi suatu transaksi, tentunya perlu dilakukan pembayaran secara *online*. Bagi penjual yang melakukan kegiatan melalui sosial media, mayoritas penjual akan meminta pembeli untuk melakukan pembayaran menggunakan metode transfer dan pembeli harap mengirimkan bukti transfer dalam bentuk foto.

Namun, foto bukti transaksi bisa diedit oleh pihak yang tidak berwenang. Foto bukti transaksi bisa menggunakan foto *screenshot* milik orang lain dengan nominal yang sama, atau mengganti jumlah nominal dengan yang lebih besar. Hal ini tentu akan menjadi sulit khususnya untuk pemegang rekening bisnis, yang menerima sejumlah transaksi dalam satu hari dan harus melakukan pengecekan apakah pembeli benar melakukan pembayaran. Salah satu cara untuk melakukan verifikasi bukti transfer pembayaran dapat menggunakan *digital signature.*

Oleh karena itu, pada makalah ini akan dibahas mengenai implementasi *digital signature* sebagai salah satu metode untuk meningkatkan keamanan pembayaran secara digital. Dengan menggunakan *digital signature*, maka setiap transaksi pembayaran akan terverifikasi, terjamin autentik, dan tidak dapat diduplikasi sesuai dengan prinsip dari tanda tangan digital.

II. DASAR TEORI

A. Tanda Tangan Digital

Tanda tangan merupakan lambang nama yang dituliskan dengan tangan oleh orang itu sendiri sebagai penanda pribadi. Tanda tangan digunakan sebagai bukti persetujuan dari orang yang mempunyai hak / otentikasi dari suatu dokumen.

Karakteristik dari tanda tangan diantaranya:

1. Tanda tangan adalah bukti yang otentik
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

Tanda tangan digital merupakan tanda tangan yang digunakan sebagai otentikasi pada data digital. Tanda tangan digital bukan merupakan tulisan tanda tangan yang didigitisasi dengan cara dipindai atau difoto, melainkan nilai kriptografis yang bergantung pada isi pesan dan kunci. Tanda tangan tertulis akan selalu sama pada setiap dokumen meskipun isi dari dokumen tersebut berbeda, sedangkan tanda tangan digital akan selalu berbeda-beda tergantung dari isi dokumen dari data digital.

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

B. Steganografi Digital

Steganografi merupakan ilmu dan seni menyembunyikan pesan rahasia dengan suatu cara sedemikian sehingga tidak seorang pun mengetahui keberadaan pesan tersebut. Steganografi termasuk dalam *information hiding* yaitu bidang ilmu yang mempelajari cara menyembunyikan pesan sehingga tidak dapat dipersepsi/ditangkap baik secara visual maupun individual.

Steganografi digital merupakan penyembunyian pesan digital di dalam dokumen digital. Pada citra digital, pesan akan disembunyikan salah satunya dengan menggunakan metode *least significant byte* (LSB). Menggunakan metode ini, setiap byte pada citra digital diubah satu lebih tingggi atau satu lebih rendah dari nilai sebelumnya. Penyisipan ini tidak akan berpengaruh terhadap persepsi visual. Ukuran pesan yang dapat disembunyikan pada gambar berwarna (RGB) dapat dihitung menggunakan persamaan (1) dimana S adalah ukuran pesan dalam byte dan P adalah ukuran pixel.

$$S = P \times P \times 3 / 8 \quad (1)$$

C. 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 algoritma RSA terletak pada sulitnya memfaktorkan bilangan bulat yang besar menjadi factor-faktor prima.

Algoritma RSA memiliki persamaan enkripsi seperti ditunjukan pada persamaan (2) dan persamaan dekripsi seperti ditunjukkan pada persamaan (3).

$$\text{Enkripsi: } E_e(m) = c = m^e \bmod n \quad (2)$$

$$\text{Dekripsi : } D_d(c) = m = c^d \bmod n \quad (3)$$

Pada makalah ini, algoritma RSA digunakan untuk melakukan enkripsi pada pesan dari gambar bukti transaksi yang di-*hash* untuk kemudian menjadi signature.

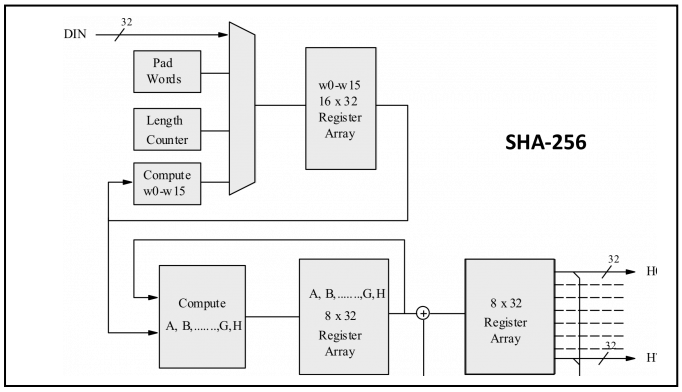
Pesan berisi : ---START--- + Nomor Rekening Pengirim - + Nomor Pengirim Penerima + - + Jumlah amount + ---END---

D. Fungsi *Hash* SHA-256

Secure Hash Algorithm 2 (SHA-2) merupakan sebuah himpunan dari fungsi hash yang dirancang oleh United States National Security Agency (NSA) pada tahun 2001. Keluarga SHA-2 terdiri dari SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256. Fungsi SHA-2 banyak digunakan untuk aplikasi dan protokol keamanan, seperti TLS dan SSL, PGP, SSH, S/MIME dan IPsec.

Secara umum, algoritma SHA-256 adalah sebagai berikut:

- L merupakan panjang pesan
- Inisialisasi *hash values* (h) yang berisi 32 bit pertama dari pecahan akar dari 8 prima pertama (2…19)
- Inisialisasi *array* berisi konstanta bilangan bulat yang berisi 32 bit pertama dari pecahan akar kubik dari 64 prima pertama (2…311)
- Melakukan *padding* dengan menambahkan satu ‘1’ bit.
- Melakukan *padding* dengan menambahkan ‘0’ bit sebanyak K dimana K merupakan angka minimum >=0 yang memenuhi L + 1 + K + 64 % 512 = 0
- Melakukan *padding* dengan menambahkan L sebagai 64-bit big-endian dimana total pesan yang telah ditambahkan padding menjadi kelipatan 512.
- Membagi pesan yang telah di-*padding* menjadi blok berukuran 512
- Untuk setiap blok, melakukan *right rotate* dan *right shift* untuk setiap karakter . Selanjutnya melakukan *compression* secara *looping*
- Nilai yang sudah terkompresi kemudian akan dimodifikasi menjadi 32 bit untuk setiap nilai h dari h0 sampai h7 untuk menghasilkan *hash value* yang final.



Gambar III.I Algoritma SHA-256
(sumber: https://www.cast-inc.com/security/encryption-primitives/sha-256)

Pada makalah ini, algoritma SHA-256 untuk melakukan *hashing* terhadap pesan yang telah digenerasi untuk kemudian dijadikan *signature* dan melakukan validasi terhadap foto bukti transaksi.

III. Rancangan Solusi dan Implementasi

Dalam menyelesaikan permasalahan, dilakukan beberapa langkah yaitu Deskripsi Umum, Rancangan, dan Implementasi Solusi.

A. Deskripsi Umum Solusi

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

Fungsi *hash* yang digunakan adalah fungsi *hash* SHA-2 dengan ukuran 256-bit atau yang dikenal sebagai SHA-256 dengan pertimbangan bahwa fungsi ini terbukti jauh lebih aman jika dibandingkan dengan fungsi *hash* yang lain. Kemungkinan terjadi kolisi juga sangat kecil karena terdapat 2^{256} kemungkinan

dua dokumen secara tidak sengaja memiliki *hash value* yang sama.

Algoritma kriptografi kunci publik yang digunakan adalah algoritma kunci public RSA denga pertimbangan keamanan dan penggunaan yang sudah populer.

Solusi yang diimplementasi memiliki tiga sudut pandang yaitu sudut pandang pembeli (*end-user*) sebagai pihak yang melakukan pembayaran, sudut pandang *store* sebagai pihak yang menerima pembayaran, dan sudut pandang *bank* sebagai pihak yang menyediakan layanan pembayaran.

B. Rancangan Solusi

Solusi yang dirancang terdiri dari dua pihak, yaitu tahap pengguna *end-user* yang melakukan pembayaran dan tahap *store* yang menerima pembayaran.

Perancangan solusi untuk tahap pengguna *end user* dimulai dengan *end user* melakukan proses pembayaran. Tahapan proses yang akan dilalui adalah sebagai berikut:

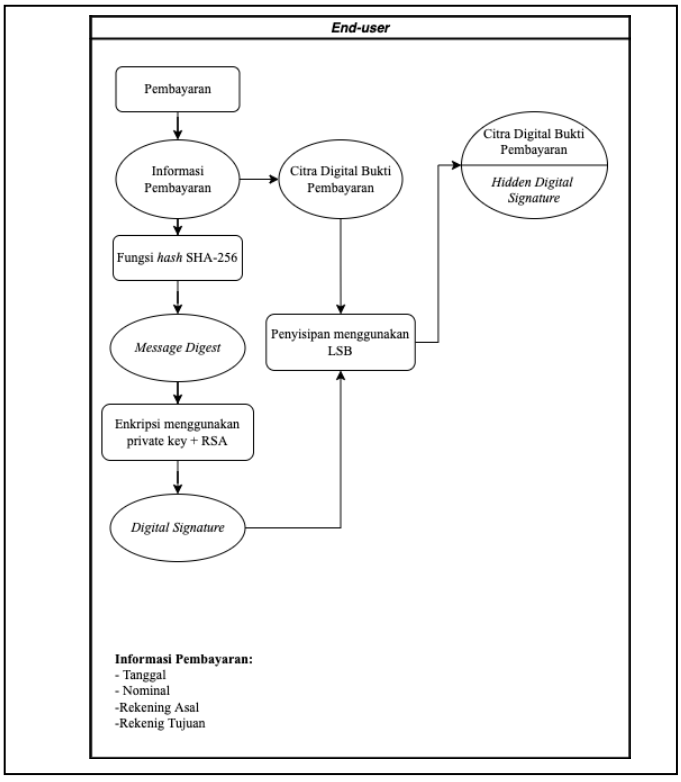
- *End-user* membuat pasangan kunci public dan kunci privat sesuai dengan ketentuan dari *bank*
- *End-user* melakukan pembayaran secara transfer dengan nominal dan tujuan tertentu. Hasil dari pembayaran akan mengeluarkan bukti transaksi digital yang berbentuk citra.
- Informasi mengenai pembayaran akan dibuat suatu pesan dengan format "---START---<tanggal transaksi>-<nomor rekening pengirim>-<nomor rekening pengguna>-<nominal pembayaran>---END---". Format tanggal transaksi menggunakan format DD/MM/YY HH:mm:ss.

  Contoh pesan dari pembayaran dari rekening 1234567890 ke rekening 0123456789 dengan jumlah seratus ribu rupiah pada 18 Desember 2021 pukul satu lewat 5 siang adalah sebagai berikut :

  ---START---
  18/12/21 13:05:00-1234567890-0123456789-100000
  ---END---
- Pesan tersebut akan di-*hash* menggunakan fungsi *hash* SHA-256 dan menghasilkan *message digest*.
- *Message digest* yang dihasilkan akan di-*encrypt* menggunakan algoritma RSA, sesuai dengan kunci private yang dimiliki oleh *end-user*.Hasil dari enkripsi akan menghasilkan *signature*
- *Signature* akan disisipkan ke dalam citra digital bukti transaksi menggunakan steganografi menggunakan LSB.

Sedangkan, perancangan untuk tahap *store* dimulai dari menerima bukti citra digital transaksi pembayaran dan mengisi form berisi tanggal transaksi, nominal, rekening asal, dan rekening tujuan yang bisa didapatkan dari citra digital transaksi pembayaran. Kemudian, pihak *store* melakukan proses validasi terhadap pembayran. Tahapan validasi yang akan dilalui adalah sebagai berikut:

- *Store* menerima bukti citra digital transaksi pembayaran dari *end-user*
- *Store* mengisi form yang berisi tanggal transaksi, nominal, rekening asal, dan rekening tujuan yang bisa didapatkan dari citra digital transaksi pembayaran.
- Melakukan ektraksi pesan tanda tangan digital dari citra digital transakasi pembayaran.
- Melakukan dekripsi pada pesan menggunakan algoritma RSA, sesuai dengan kunci publik yang dimiliki oleh *end-user*, *store,* dan *bank*. Hasil dari dekripsi berupa *message digest*.
- Dilakukan perbandingan antara *message digest* yang dihasilkan dengan pesan yang dibangun dari form.
- Apabila hasilnya sama, maka program akan menginformasikan bahwa citra digital transaksi pembayaran terverifikasi.
- Apabila hasilnya berbeda, maka program akan menginformasikan bahwa citra digital transaksi pembayaran gagal verifikasi.

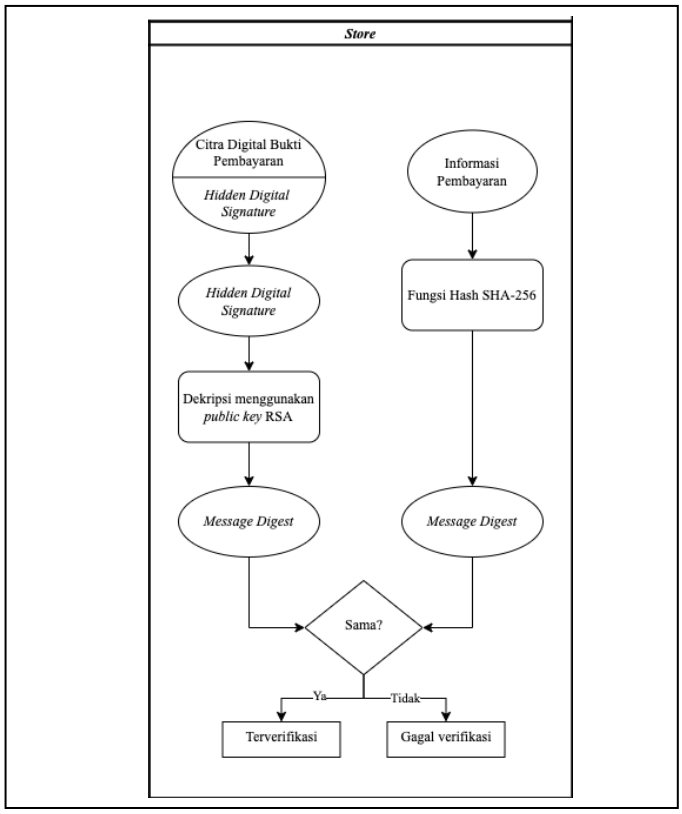


Gambar III.2 Arsitektur Rancangan Solusi *End-user*

Perancangan solusi untuk *end-user* dapat dilihat pada Ganbar III.2. Solusi dimulai dengan *end-user* melakukan pembayaran, mengambil informasi pembayaran yang kemudian di-*hash* menggunakan fungsi hash SHA-256, dienkripsi menggunakan

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors).*

algoritma dan melakukan proses *signing*. *Digital* signature kemudian disipkan ke dalam citra digital bukti pembayaran menggunakan steganografi.



Gambar III.3 Arsitektur Rancangan Solusi *Store*

C. Implementasi

Solusi diimplementasikan sebagai sebuah *representational state transfer application programming interface* (REST API). Penyimpanan file juga menggunakan *firebase storage.* Pihak bank selaku penyedia layanan pembayaran dapat menentukan membangkitkan pasangan kunci publik dan privat untuk *end*-user dan *store.* Pihak *end*-user yang ingin melakukan pembayaran dapat menggunakan kunci public dan privat yang diberikan oleh bank dan dapat melakukan pembayaran. Setelah itu, pihak bank akan membuat citra digital bukti transaksi yang didalamnya terdapat *digital signature* yang tersembunyi (*invisible watermark*). *End-user* kemudian dapat mengirimkan bukti transaksi kepada *store* sebagai pelaporan penyelesaian pembayaran. *Store* dapat melakukan validasi terhadap pembayaran tanpa harus mengecek mutasi akun rekening dengan memasukkan citra digital bukti transaksi dan informasi pembayaran ke suatu *endpoint* untuk menentukan apakah citra digital valid atau tidak.

Dalam implementasi, terdapat beberapa endpoint dalam API yang dibangun, yaitu *endpoint* untuk pembangkitan kunci, proses pembangkitan *signature* / *signing* dan proses *verifying.*

1. Pembangkitan kunci (`/generate`)

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

   Request payload (JSON)

```
{
  "nomor_rekening": 17482653
}
```

   Response payload (JSON)

```
{
  "public_key": 17011,
  "private_key": 560611,
}
```

2. Proses pembangkitan *signature* / *signing* (`/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)

```
{
  "tanggal_pembayaran": "11/12/21 13:01:00",
  "total_pembayaran": 10000,
  "rekening_asal": 17482653,
  "rekening_tujuan: 84728492,
  "private_key": 560611,
}
```

   Response payload (JSON)

```
{
  "transaction_filename": "transaction.png"
}
```

   Servis akan meng-*upload* citra digital bukti transaksi ke *firebase storage. End-user* dapat mengunduh file dengan nama tersebut dari *firebase storage* sehingga *response* hanya perlu berisi nama file dari bukti transaksi. Perlunya menyimpan bukti transaksi pada *storage* yang berada pada *cloud server* sehingga dapat di-*download* oleh *end-user*.

3. Proses *verifying* (`/verifying`)

Endpoint untuk melakukan verifikasi citra digital bukti pembayaran menggunakan HTTP *method* POST dengan *request payload* berupa nama file bukti transaksi yang akan di-*upload* ke *firebase storage* dan informasi pembayaran

Request payload (JSON)

```
{
  "tanggal_pembayaran": "11/12/21 13:01:00",
  "total_pembayaran": 10000,
  "rekening_asal": 17482653,
  "rekening_tujuan: 84728492,
  "public_key": 560611,
  "transaction_filename": "transaction.png"
}
```

Response payload (JSON)

```
{
  "verify": true,
}
```

IV. Pengujian dan Pembahasan

A. Pengujian

Terdapat tiga pengujian yang dilakukan. Pengujian pertama yaitu pembayaran dengan citra digital bukti transaksi yang valid dan informasi pembayaran yang valid saat proses verifying. Pengujian yang kedua yaitu pembayaran dengan citra digital bukti transaksi yang tidak valid dan informasi pembayaran yang valid saat proses *verifying*. Pengujian yang ketiga yaitu pembayaran dengan citra digital bukti transaksi yang valid dan informasi pembayaran yang tidak valid saat proses *verifying*.

Lingkungan pengujian yang digunakan ditujukan pada tabel berikut:

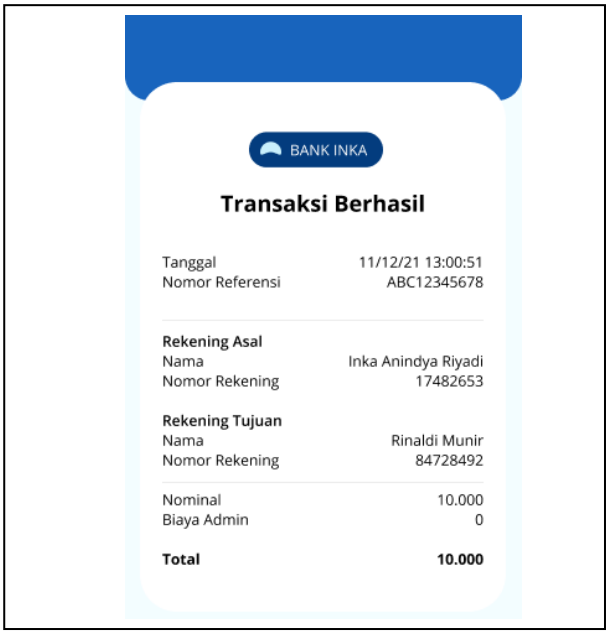
TABLE I. Lingkungan Pengujian

Kasus Pengujian	
Variabel	**Nilai**
tanggal_pembayaran	11/12/21 13:01:00
total_pembayaran	10000
rekening_asal	17482653
rekening_tujuan	84728492
valid_signature (hasil pembangkitan)	1976351 1735057 329936 1331965 1649313 1583993 134733 1123590 1501679 853651 1326874 1890908 625316 1702585 266807 990820 513497 1658040 2335045 1497971 1028910 2388487 762783 250818 2223332 1640603 513497
public_key (hasil pembangkitan)	17011
private_key (hasi pembangkitan)	560611

1. Verifikasi bukti transaksi pembayaran yang valid dengan informasi pembayaran yang valid

   Pada pengujian ini, dikirimkan citra digital bukti transaksi pembayaran yang valid dan informasi pembayaran yang valid. Berikut merupakan *request payload:*

```
{
  "tanggal_pembayaran": "11/12/21 13:00:51",
  "total_pembayaran": 10000,
  "rekening_asal": 17482653,
  "rekening_tujuan: 84728492,
  "public_key": 560611,
  "transaction_filename": "transaction.png"
}
```



Gambar IV.1 Citra Digital Bukti Transaksi "transaction.png"

Hasil *response* dari API verifikasi adalah sebagai berikut:

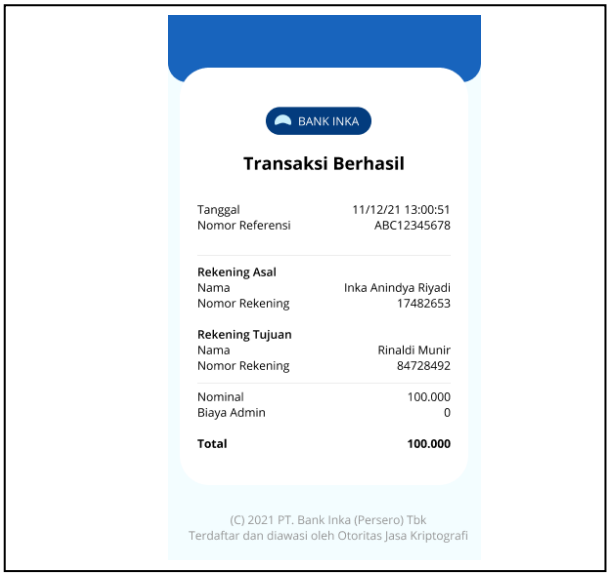
```
{
  "verify": true,
}
```

Berdasarkan hasil *response* dari API, citra digital bukti pembayaran merupakan transaksi yang valid.

2. Verifikasi bukti transaksi pembayaran yang tidak valid dengan informasi pembayaran yang valid

   Pada pengujian ini, dikirimkan citra digital yang telah dimodifikasi, dengan mengubah nominal pembayaran menjadi 100,000 dari 10,000. Berikut merupakan *request payload:*

```
{
  "tanggal_pembayaran": "11/12/21 13:00:51",
  "total_pembayaran": 100000,
  "rekening_asal": 17482653,
  "rekening_tujuan: 84728492,
  "public_key": 560611,
  "transaction_filename": "transaction2.png"
}
```



Gambar IV.2 Citra Digital Bukti Transaksi "transaction2.png"

Hasil *response* dari API verifikasi adalah sebagai berikut:

```
{
  "verify": false,
}
```

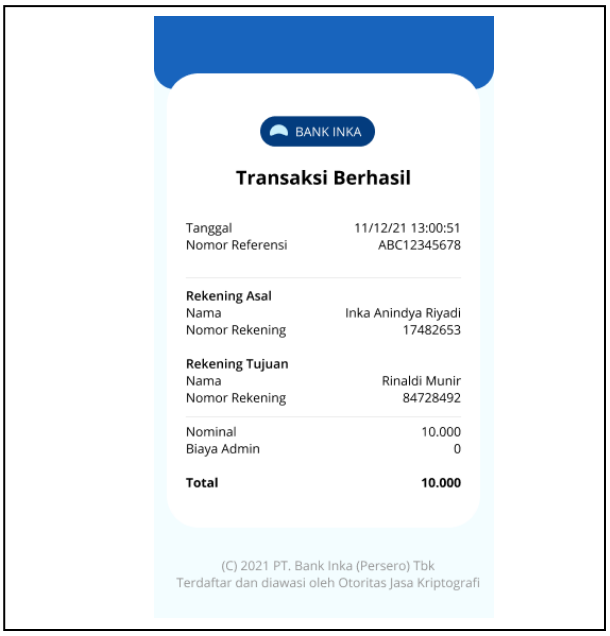
Berdasarkan hasil *response* dari API, citra digital bukti pembayaran merupakan transaksi yang tidak valid.

3. Verifikasi bukti transaksi pembayaran yang valid dengan informasi pembayaran yang tidak valid

   Pada pengujian ini, dikirimkan citra digital bukti transaksi pembayaran yang valid dan informasi pembayaran yang tidak valid dengan memasukkan *public key* yang tidak sesuai yaitu 560617 dari yang seharusnya 560611

   Berikut merupakan *request payload:*

```
{
  "tanggal_pembayaran": "11/12/21 13:00:51",
  "total_pembayaran": 10000,
  "rekening_asal": 17482653,
  "rekening_tujuan: 84728492,
  "public_key": 560617,
  "transaction_filename": "transaction.png"
}
```



Gambar IV.3 Citra Digital Bukti Transaksi "transaction.png"

Hasil *response* dari API verifikasi adalah sebagai berikut:

```
{
  "verify": false,
}
```

Berdasarkan hasil *response* dari API, citra digital bukti pembayaran merupakan transaksi yang tidak valid.

B. Pembahasan

Berdasarkan pengujian yang telah dilakukan, didapatkan hasil sebagai berikut:

1. Verifikasi bukti transaksi pembayaran yang valid dengan informasi pembayaran yang valid

   Berdasarkan gambar IV.1 citra digital bukti pembayaran merupakan transaksi yang valid. Seluruh informasi yang tersedia pada citra digital, yaitu tanggal

pembayaran, nominal, rekening asal, dan rekening tujuan merupakan informasi yang valid

2. Verifikasi bukti transaksi pembayaran yang tidak valid dengan informasi pembayaran yang valid

   Berdasarkan gambar IV.2 citra digital bukti pembayaran gagal terverifikasi. Hal ini dikarenakan *store* menginput nominal pembayaran sejumlah 100,000. Padahal, seharusnya transaksi tersebut bernilai 10,000.

3. Verifikasi bukti transaksi pembayaran yang valid dengan informasi pembayaran yang tidak valid

   Berdasarkan gambar IV.3 citra digital bukti pembayaran gagal terverifikasi karena pihak *store* salah memasukkan *public key* untuk transaksi tersebut. *Public key* untuk transaksi tersebut seharusnya 560611 namun pihak *store* memasukkan *public key* dengan nilai 560617

V. Kesimpulan dan Saran Pengembangan

Solusi yang diimplementasikan berhasil untuk memastikan citra digital bukti pembayaran memenuhi tiga aspek, yaitu autentik, asli, dan anti-penyangkalan. Hal ini memfasilitasi pihak penerima pembayaran untuk memverifikasi pembayaran dari pihak yang melakukan pembayaran tanpa harus melakukan pengecekan mutasi pada rekening.

Kedepannya, solusi dapat dikembangkan lebih lanjut pada beberapa bagian, seperti ekstraksi informasi pembayaran yaitu tanggal, nominal, rekening asal, dan rekening tujuan menggunakan *Artificial Intelligence* untuk mengurangi *human error* pada saat melakukan input informasi pembayaran saat proses verifikasi.

Ucapan Terima kasih

Ucapan terima kasih 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 terima kasih kepada Dr. Rinaldi Munir atas dedikasinya dalam memberikan ilmu pengetahuan terkait mata kuliah kriptografi kepada penulis.

Referensi

[1] Munir, Rinaldi. 2021. Slide Kuliah IF4020 Kriptografi: Steganografi (Bagian 1)

[2] Munir, Rinaldi. 2021. Slide Kuliah IF4020 Kriptografi: Steganografi (Bagian 2)

[3] Munir, Rinaldi. 2021. Slide Kuliah IF4020 Kriptografi: Fungsi Hash

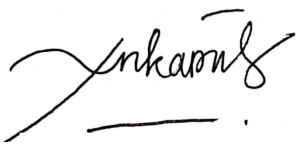
[4] Munir, Rinaldi. 2021. Slide Kuliah IF4020 Kriptografi: Algoritma RSA

[5] Aptanagi, Pandyaka, "Implementasi Kriptografi Kunci Publik RSA dan Fungsi Hash SHA3 sebagai *Digital Signature* pada Pembayaran Digital", Institut Teknologi Bandung, Desember 2020

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



Inka Anindya Riyadi
13518038