Implementasi One-Time Password dan Digital Signature RSA-PKCS#1 v1.5 pada Physical Gift Card

Rozan Ghosani 18221121
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
18221121@std.stei.itb.ac.id

***Abstract*—Gift card adalah metode pembayaran yang populer di era digital, namun seringkali menghadapi ancaman keamanan seperti pemalsuan, penyalahgunaan, dan pencurian. Untuk mengatasi masalah ini, penulis menyarankan penggunaan OTP dan digital signature. OTP adalah kode unik yang dihasilkan secara acak dan hanya berlaku untuk satu kali transaksi atau dalam jangka waktu yang sangat singkat. Sementara itu, digital signature berfungsi sebagai verifikasi keaslian dan integritas data. Penelitian ini bertujuan untuk mengkaji efektivitas penggunaan OTP dan digital signature dalam meningkatkan keamanan gift card. Fokus utama penelitian ini adalah bagaimana kombinasi kedua metode ini dapat memberikan perlindungan yang lebih baik dibandingkan metode konvensional. Teknologi yang umum digunakan dalam implementasi algoritma tanda tangan digital adalah RSA-PKCS#1 v1.5 dan SHA-3. Kedua algoritma ini dapat memberikan keseimbangan yang baik antara keamanan, kompatibilitas, dan kinerja.**

Keywords—digital signature; rsa; gift card;

I. Pendahuluan

Gift card, atau secara harfiah dapat diterjemahkan sebagai kartu hadiah, merupakan salah satu metode pembayaran yang sangat populer di era digital saat ini. Kemudahan penggunaannya membuat *gift card* sering digunakan sebagai hadiah atau alat alternatif pembayaran dalam bertransaksi. Namun, seperti halnya alat pembayaran lainnya, *gift card* tidak lepas dari ancaman keamanan seperti pemalsuan, penyalahgunaan, dan pencurian. *Gift card* umumnya hanya diamankan dengan menggunakan lapisan penutup goresan pada bagian belakang kartu untuk memastikan bahwa kode pada kartu tersebut belum pernah digunakan sebelumnya [3]. Berdasarkan situasi ini, siapa pun yang memiliki kode pada *gift card* dapat menukarkan uang tersebut. Seiring dengan lemahnya pengamanan pada *gift card*, kebutuhan akan mekanisme keamanan yang lebih kuat menjadi sangat penting.

Permasalahan keamanan ini dapat diatasi dengan menerapkan dua mekanisme keamanan, yaitu One-Time Password (OTP) dan *digital signature*. OTP adalah kode unik yang dihasilkan secara acak dan hanya berlaku untuk satu kali transaksi atau dalam jangka waktu yang sangat singkat [2]. Penerapan OTP pada aktivasi *gift card* diharapkan dapat mencegah penggunaan yang tidak sah atau pencurian pada *gift card*. Di sisi lain, *Digital signature* berfungsi sebagai verifikasi keaslian dan integritas data. Dengan menggunakan digital signature, setiap transaksi atau aktivasi *gift card* dapat diverifikasi keasliannya, sehingga mencegah manipulasi atau pemalsuan data.

Penelitian ini bertujuan untuk mengkaji efektivitas penggunaan OTP dan *digital signature* dalam meningkatkan keamanan *gift card*. Fokus utama penelitian ini adalah bagaimana kombinasi kedua metode ini dapat memberikan perlindungan yang lebih baik dibandingkan metode konvensional. Selain itu, penelitian ini juga akan mengeksplorasi implementasi teknis dan tantangan yang mungkin dihadapi dalam penerapan sistem keamanan ini.

II. Dasar Teori

A. Gift Card

Gift card adalah sebuah kartu yang memuat sejumlah uang yang bisa digunakan untuk melakukan pembelian di tempat-tempat tertentu atau dalam jaringan toko tertentu. Gift card dapat berbentuk fisik atau digital, dan sering kali digunakan sebagai hadiah atau alternatif uang tunai. Untuk gift card konvensional yang berbentuk fisik, biasanya terdapat area yang harus digosok untuk menampilkan kode unik.

Proses penggunaan *gift card* ini melibatkan beberapa langkah. Pertama, bagian belakang kartu perlu digosok terlebih dahulu untuk menampilkan kode. Setelah kode terlihat, pemilik kartu dapat mengunjungi situs *web* berpartisipasi dan memasukkan kode tersebut pada saat pembayaran atau menukarkan kode tersebut untuk menambah saldo dompet digital.

B. One-Time Password

One-Time Password (OTP) adalah mekanisme autentikasi yang menghasilkan kata sandi unik yang hanya berlaku untuk satu sesi atau transaksi tertentu. OTP adalah solusi keamanan yang dirancang untuk mengatasi kelemahan yang ada dalam sistem kata sandi tradisional yang statis [2]. OTP memberikan lapisan keamanan tambahan dengan mengharuskan pengguna untuk memasukkan kata sandi yang berbeda setiap kali mereka mengakses sistem atau layanan tertentu, sehingga mengurangi risiko akses yang tidak sah akibat kata sandi yang dicuri atau ditebak.

C. Digital Signature

Tanda tangan digital atau *digital signature* adalah mekanisme kriptografi yang digunakan untuk memverifikasi keaslian dan integritas pesan, perangkat lunak, atau dokumen digital. Tanda tangan digital memberikan bukti bahwa pesan atau dokumen tersebut dikirim oleh pengirim yang sah dan tidak diubah sejak ditandatangani [1].. Tanda tangan digital biasanya menggunakan algoritma kriptografi untuk menciptakan nilai hash dari pesan yang dikombinasikan dengan kunci pribadi pengirim.

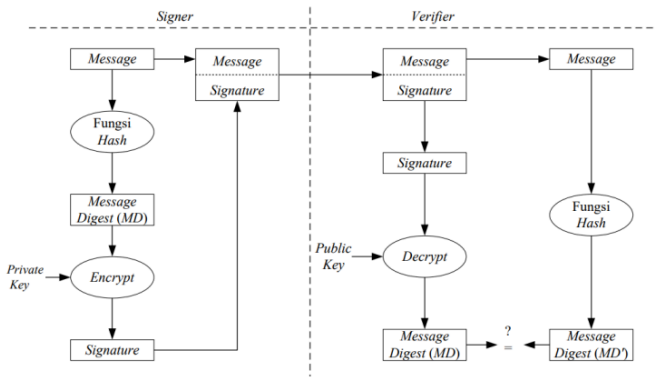

Fig. 1. Mekanisme Tanda Tangan Digital. *(sumber: informatika.stei.itb.ac.id/~rinaldi.munir)*

Proses tanda tangan digital dilakukan dengan melewati pesan pada fungsi *hash* yang unik dan tetap panjangnya. Nilai dari *hash* ini kemudian dienkripsi dengan menggunakan algoritma kriptografi kunci-publik. Enkripsi pesan dilakukan dengan menggunakan kunci privat pengirim pesan. Dengan ini, maka kerahasiaan pesan dan autentikasi pengirim pesan dapat dicapai. Penerima pesan dapat mengautentikasi pengirim pesan karena kunci publik dan privat yang digunakan berpasangan.

Verifikasi pesan dapat dilakukan dengan menggunakan kunci publik pengirim untuk mendekripsi tanda tangan digital menjadi nilai *hash*. Hasil dari dekripsi tanda tangan digital tersebut kemudian dapat dibandingkan dengan nilai *hash* yang didapatkan melalui penghitungan nilai *hash* dari pesan asli yang diterima. Apabila kedua nilai *hash* ini cocok, maka pesan ini dapat dikatakan autentik dan tidak diubah.

Teknologi yang umum digunakan dalam implementasi algoritma tanda tangan digital adalah RSA-PKCS#1 v1.5 dan SHA-3. Kedua algoritma ini dapat memberikan keseimbangan yang baik antara keamanan, kompatibilitas, dan kinerja.

D. RSA-PKCS#1 v1.5

RSA-PKCS#1 v1.5 adalah sebuah skema kriptografi yang digunakan dalam enkripsi dan penandatanganan digital, yang merupakan bagian dari standar PKCS (Public Key Cryptography Standards) yang dikembangkan oleh RSA Laboratories. RSA adalah algoritma kriptografi kunci publik yang digunakan untuk enkripsi dan penandatanganan digital. Algoritma ini berdasarkan pada kesulitan faktorisasi bilangan bulat besar, yang membuatnya aman digunakan dalam praktik. PKCS adalah serangkaian standar yang dikembangkan oleh RSA Laboratories untuk memfasilitasi penggunaan kriptografi kunci publik.

Saat menggunakan RSA untuk enkripsi dengan PKCS#1 v1.5, pesan asli dipad (ditambahkan pengganjal) dengan cara tertentu sebelum dienkripsi [1]. Padding ini terdiri dari byte tetap, byte pengisi acak, dan pesan asli. Ini membantu untuk mencegah serangan yang menggunakan pola-pola dalam data asli.

Untuk menghasilkan kunci, setiap entitas perlu menentukan sebuah nilai bilangan bulat positif e untuk dijadikan eksponen publik. Kemudian setiap entitas harus memilih bilangan dua buah bilangan prima p dan q sehingga $(p-1)$ dengan e tidak mempunyai pembagi persekutuan, dan $(q-1)$ dengan e tidak mempunyai pembagi persekutuan.

1. Pilih dua bilangan prima besar p dan q
2. Hitung n dari hasil kali dari faktor p dan q

   $$n = pq$$

   Di mana n adalah modulus yang digunakan untuk kunci publik dan privat
3. Hitung $\phi(n)$ dengan menggunakan persamaan

   $$\phi(n) = (p-1)(q-1)$$
4. Pilih eksponen publik e sehingga memenuhi

   $$1 < e < \phi(n) \text{ dan } GCD(e, \phi(n)) = 1$$
5. Hitung eksponen privat d

   $$d \equiv e^{-1} (mod\ \phi(n))$$

Panjang modulus n dalam oktet adalah bilangan bulat k yang memenuhi,

$$2^{8(k-1)} \leq n < 2^{8k}$$

Kunci publik terdiri dari pasangan (n, e) dan kunci privat terdiri dari pasangan (n, d).

Untuk mengenkripsi pesan dengan RSA-PKCS#1 v1.5, pesan asli M di-*pad* menjadi m dengan menambahkan *padding* yang terdiri dari byte tetap $\backslash x00$ dan $\backslash x02$ diikuti oleh serangkaian byte acak dan diakhiri dengan byte $\backslash x00$. Panjang blok *padding* harus cukup untuk membuat panjang total m sama dengan panjang modulus n. Format *padding* memenuhi,

$$m = 00||02||PS||00||M$$

Di mana PS adalah string acak non-nol

Pesan kemudian dienkripsi dengan menggunakan kunci publik (n, e) dengan menggunakan persamaan,

$$C = m^e \ (mod\ n)$$

Pesan dapat didapatkan kembali dengan mendekripsi cipherteks C menggunakan kunci privat (n, d) dengan persamaan,

$$m = C^d \ (mod\ n)$$

Setelah dekripsi, *padding* dapat dihapus untuk mendapatkan kembali pesan asli *M*

E. SHA-3 (Keccak)

SHA-3 (Secure Hash Algorithm 3) adalah keluarga fungsi hash kriptografis yang dirancang oleh National Institute of Standards and Technology (NIST). SHA-3 adalah versi terbaru dari keluarga SHA, yang dikenal dengan keamanan yang kuat dan efisiensi [2]. SHA-3 dirancang dengan mempertimbangkan ketahanan terhadap serangan seperti serangan tabrakan (*collision*) dan serangan preimage. Ini membuatnya sangat aman untuk digunakan dalam konteks tanda tangan digital.

SHA-3 menggunakan struktur konstruksi *sponge* dengan menggunakan fungsi non-kompresi untuk menyerap (*absorbing*) dan kemudian memeras (*squeezing*) pesan yang masuk.

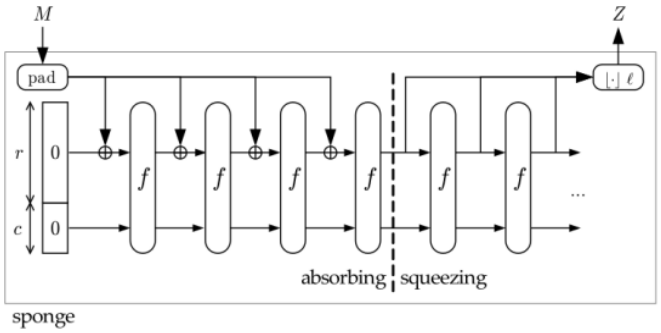

Fig. 2. Konstruksi Spons pada SHA-3 Keccak. *(sumber: informatika.stei.itb.ac.id/~rinaldi.munir)*

Pesan yang masuk, misalkan M akan ditambahkan bit pengganjal menjadi string P sehingga P habis dibagi dengan panjang blok r bit atau $n = length(P) / r$. Pada fase penyerapan, setiap blok masukan P_i berukuran r bit akan dioperasikan dengan *state* S dengan menggunakan operasi XOR. Fungsi permutasi f adalah inti dari Keccak dan terdiri dari beberapa tahap operasi *bitwise* yang diulang beberapa kali. Ini adalah serangkaian operasi non-linier yang dirancang untuk mencampur bit dalam *buffer* internal dengan cara yang kompleks.

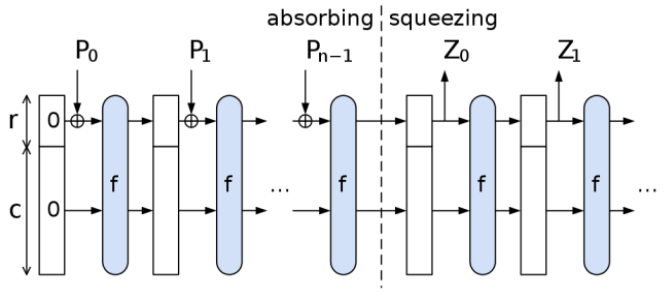

Fig. 3. Proses *Absorbing* dan *Squeezing* pada SHA-3. *(sumber: informatika.stei.itb.ac.id/~rinaldi.munir)*

Setelah seluruh blok masukan selesai diproses, konstruksi spons akan beralih pada fase pemerasan. Pada fase ini hasil *message digest* akan disimpan ke dalam Z yang diinisiasikan dengan string kosong atau *null*. R-bit dari *state* S akan ditambahkan ke dalam Z hingga panjang Z sama dengan d. Jika panjang Z masih belum sama dengan d, maka Z akan terus dimasukan ke fungsi permutasi f hingga panjang Z sama dengan d.

III. DESAIN

Kondisi sistem saat ini atau *as-is* dari proses produksi dan penggunaan *gift card* mencakup proses pembuatan kode *gift card*, pembelian, dan proses penukaran kode *gift card*. Pertama, kode unik diproduksi untuk setiap gift card. Setelah itu, konsumen membeli gift card tersebut. Akhirnya, konsumen memasukkan kode tersebut untuk menggunakannya. Berikut merupakan diagram proses yang menjelaskan proses produksi dan penggunaan *gift card* pada kondisi sistem *as-is*.

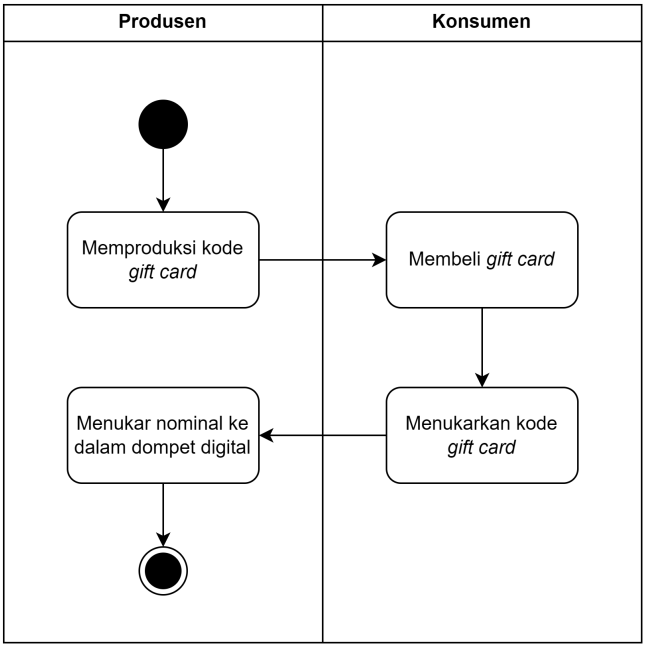

Fig. 4. Proses Produksi dan Penggunaan *Gift Card* pada Sistem *As-is*

Pada sistem *to-be,* langkah-langkah tambahan untuk meningkatkan keamanan dan autentikasi ditambahkan pada sistem. Pertama, seperti sistem *as-is*, kode unik diproduksi untuk setiap *gift card*. Namun, langkah selanjutnya adalah penambahan tanda tangan digital pada kode tersebut, yang berfungsi sebagai lapisan keamanan tambahan. Setelah itu, konsumen membeli *gift card* tersebut seperti biasa.

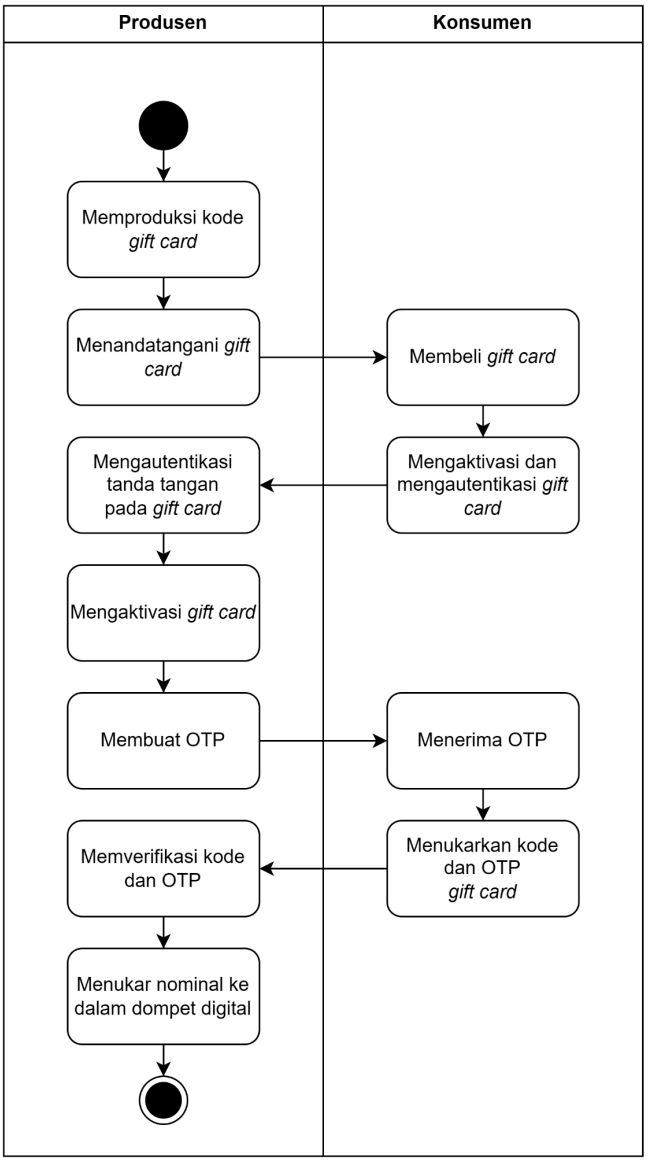

Fig. 5. Proses Produksi dan Penggunaan *Gift Card* pada Sistem *To-be*

Sebelum kode dapat dimasukkan, konsumen harus melakukan aktivasi *gift card*. Ini adalah langkah verifikasi tambahan di mana konsumen menerima OTP (One Time Password), yang mereka dapatkan setelah mengaktifkan *gift card*, yang harus mereka masukkan untuk menukarkan *gift card*. Hal ini dilakukan untuk mencegah pencurian atau *fraud* terjadi dengan memasukan kode yang tidak aktif atau tidak valid pada saat menukarkan *gift card*. Setelah konsumen memasukkan kode gift card tersebut dan memverifikasi dengan menggunakan OTP yang diberikan, barulah nominal dari *gift card* dapat ditukarkan oleh sistem. Dengan demikian, sistem *to-be* ini menambahkan lapisan keamanan dan autentikasi tambahan untuk melindungi konsumen dan perusahaan dari penyalahgunaan *gift card*.

IV. Implementasi

Implementasi dilakukan berdasarkan desain sistem yang telah dibuat. Program dibuat dengan menggunakan bahasa Python dan dengan pustaka Flet untuk membangun aplikasi. Implementasi dibagi menjadi tiga modul utama, yakni modul tanda tangan digital yang menggunakan RSA-PKCS#1 v1.5 dan SHA-3, modul One-Time-Password (OTP), dan modul basis data untuk mengatur penyimpanan *gift card*.

A. Modul Tanda Tangan Digital

Modul ini berfungsi untuk melakukan tanda tangan digital terhadap message apapun yang masuk.

TABLE I. Modul Tanda Tangan Digital

```
from cryptography.hazmat.primitives.asymmetric
import rsa, padding
from cryptography.hazmat.primitives import hashes,
serialization

class Signature:
    def __init__(self):
        self.private_key = None
        self.public_key = None

    def generate_key_pair(self):
        self.private_key =
rsa.generate_private_key(
            public_exponent=65537,
            key_size=2048
        )
        self.public_key =
self.private_key.public_key()

    def sign(self, message):
        return self.private_key.sign(
            message,
            padding.PKCS1v15(),
            hashes.SHA3_256()
        )

    def verify(self, signature, message):
        try:
            self.public_key.verify(
                signature,
                message,
                padding.PKCS1v15(),
                hashes.SHA3_256()
            )
            return True
        except Exception as e:
            return False

    def get_private_key(self):
        return self.private_key.private_bytes(
            encoding=serialization.Encoding.PEM,

format=serialization.PrivateFormat.TraditionalOpen
SSL,

encryption_algorithm=serialization.NoEncryption()
```

```
        )

    def get_public_key(self):
        return self.public_key.public_bytes(
            encoding=serialization.Encoding.PEM,

format=serialization.PublicFormat.SubjectPublicKey
Info
        )
```

Dalam implementasi modul ini, fungsi *hash* yang digunakan adalah SHA3_256 atau SHA-3 Keccak dengan ukuran blok 256 bit. Kelas Signature juga menggunakan modul RSA untuk proses enkripsi dan deksripsi dengan menggunakan padding PCKS1v15. Kedua modul ini didapatkan dengan menggunakan pustaka *cryptography*. Kelas Signautre memiliki beberapa metode sebagai berikut:

- `generate_key_pair(self)`: Metode ini digunakan untuk menghasilkan pasangan kunci publik dan kunci privat RSA yang akan digunakan untuk mengenkripsi dan dekripsi hasil *hash* pesan.
- `sign(self, message)`: Metode ini digunakan untuk menghasilkan tanda tangan berdasarkan masukan pesan yang diterima.
- `verify(self, signature, message)`: Metode ini digunakan untuk memverifikasi pesan yang diterima dengan cara melakukan *hash* pada pesan tersebut. Kemudian *signature* akan didekripsi dengan menggunakan kunci publik RSA. Hasil dari dekripsi dan *hash* pesan ini akan dibandingkan kesamaannya.
- `get_private_key(self)`: Metode ini digunakan untuk mendapatkan private key yang digunakan pada RSA.
- `get_public_key(self)`: Metode ini digunakan untuk mendapatkan public key yang digunakan pada RSA.

B. Modul One-Time Password

TABLE II. MODUL ONE-TIME PASSWORD

```
import datetime
import random

class OTP:
    def __init__(self):
        self.code = None
        self.expiration_time = None

    def generate(self):
        # Generate OTP 6 digits of string
        return str(random.randint(100000, 999999))

    def is_expired(self):
        return datetime.datetime.now() >
self.expiration_time

    def is_valid(self, otp: str):
        return otp == self.code

    def refresh(self):
        self.code = self.generate()
        self.expiration_time =
datetime.datetime.now() +
datetime.timedelta(days=1)
```

Dalam implementasi modul ini, kode OTP yang dibangun dibuat secara acak dengan menggunakan pustaka *random*. Pustaka ini akan menghasilkan kode OTP secara acak berupa bilangan enam digit. Kode OTP ini diimplementasikan untuk valid selama satu hari semenjak *gift card* diaktivasi oleh pengguna. Kelas OTP ini memiliki beberapa metode sebagai berikut:

- `generate(self)`: Metode ini digunakan untuk menghasilkan kode OTP acak berupa bilangan enam digit.
- `is_expired(self)`: Metode ini digunakan untuk mengetahui apakah kode OTP telah kedaluwarsa atau belum.
- `is_valid(self, otp: str)`: Metode ini digunakan untuk mengetahui apakah kode OTP masukan sama dengan kode OTP yang berlaku.
- `refresh(self)`: Metode ini digunakan untuk memperbarui kode OTP dan juga tenggat kedaluwarsa dari kode tersebut.

C. Modul Basis Data

II. MODUL ONE-TIME PASSWORD

```
import random
import base64

from services.signature import Signature
from services.otp import OTP

class GiftCard:
    def __init__(self, value: int, code: str):
        self.value = value
        self.code = code
        self.status = "inactive"
        self.signature = None
        self.otp = OTP()

    def activate(self):
        self.status = "active"
        self.otp.refresh()

        return self.otp.code

    def set_signature(self, signature):
        self.signature = signature

    def validate_signature(self, signature):
        return self.signature == signature

    def redeem(self, otp):
```

```
        if self.otp.is_expired():
            return False

        if not self.otp.is_valid(otp):
            return False

        self.status = "redeemed"
        return True

class Database:
    def __init__(self):
        self.gift_cards = {}
        self.signature = Signature()
        self.signature.generate_key_pair()

    def create_gift_card(self, value: int):
        # Generate a random character code of
length 6
        code =
"".join(random.choices("ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789", k=6))

        # Check if the code is unique
        while self.get_gift_card(code):
            code =
"".join(random.choices("ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789", k=6))

        gift_card = GiftCard(value, code)
        signature = self.signature.sign((code +
str(value)).encode())

gift_card.set_signature(base64.b64encode(signature
).decode())

        self.gift_cards[code] = gift_card
        return gift_card

    def get_gift_card(self, code: str):
        return self.gift_cards.get(code)

    def activate_gift_card(self, code: str):
        gift_card = self.get_gift_card(code=code)
        if gift_card:
            return gift_card.activate()
        return None
```

Dalam implementasi modul ini, terdapat dua kelas yang digunakan. Kelas GiftCard adalah kelas yang digunakan untuk mendefinisikan *gift card*. Setiap *gift card* memiliki kode, nilai, status, tanda tangan digital, dan kode OTP masing-masing. Metode yang terdapat pada kelas ini sebagai berikut:

- `activate(self)`: Metode ini digunakan untuk mengaktifkan *gift card* dengan mengubah statusnya dan membuat kode OTP baru yang berlaku selama satu hari.
- `set_signature(self, signature)`: Metode ini digunakan untuk memberikan identitas tanda tangan digital pada *gift card*.
- `validate_signature(self, signature)`: Metode ini digunakan untuk memvalidasi tanda tangan digital berdasarkan tanda tangan digital masukan.
- `redeem(self, otp)`: Metode ini digunakan untuk menukarkan *gift card* dengan melakukan autentikasi menggunakan kode OTP yang diberikan. Kode OTP akan diperiksa apakah belum kedaluwarsa dan valid.

Kelas Database adalah kelas yang digunakan untuk mendefinisikan basis data yang terdiri dari himpunan *gift card* unik dan objek Signature atau tanda tangan yang digunakan oleh sistem. Kelas ini memiliki metode sebagai berikut:

- `create_gift_card(self, value: int)`: Metode ini digunakan untuk membuat *gift card* dengan nilai sesuai masukan dan kode unik. *Gift card* baru yang dibuat akan ditanda tangani untuk memastikan keasliannya.
- `get_gift_card(self, code: str)`: Metode ini digunakan untuk mencari dan mengembalikan *gift card* berdasarkan kode yang sesuai.
- `activate_gift_card(self, code: str)`: Metode ini digunakan untuk mengaktifkan *gift card* berdasarkan kode yang sesuai.
- `redeem(self, otp)`: Metode ini digunakan untuk menukarkan *gift card* dengan melakukan autentikasi menggunakan kode OTP yang diberikan. Kode OTP akan diperiksa apakah belum kedaluwarsa dan valid.

V. HASIL

Aplikasi yang dibangun memiliki Graphical User Interface (GUI) untuk memudahkan penggunaan dalam interaksi dengan sistem. Aplikasi ini dibangun dengan basis platform *website* untuk memudahkan penggunaan pada berbagai sistem operasi yang beragam. Berikut merupakan tampilan halaman utama dari aplikasi.

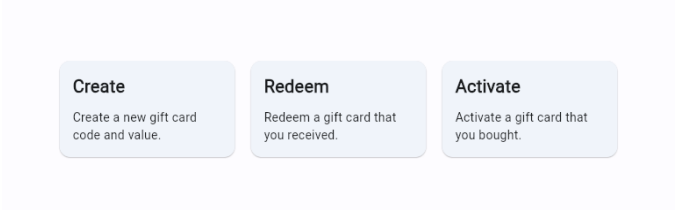

Fig. 6. Tampilan Halaman Utama Aplikasi

Halaman utama aplikasi memberikan pilihan tiga fitur utama dalam sistem ini, yakni pembuatan *gift card*, penukaran *gift card*, dan aktivasi *gift card* dengan menggunakan tanda tangan digital.

A. Halaman Pembuatan Gift Card

Fitur pembuatan *gift card* memfasilitasi pembuatan *gift card* secara otomatis dengan hanya menerima masukan nilai *gift card* yang akan diberikan. Halaman ini hanya menerima masukan berupa nilai *gift card* dalam mata uang rupiah dan juga tombol untuk mengirimkan submisi pembuatan *gift card*. Berikut merupakan tampilan antarmuka pada halaman pembuatan *gift card*.

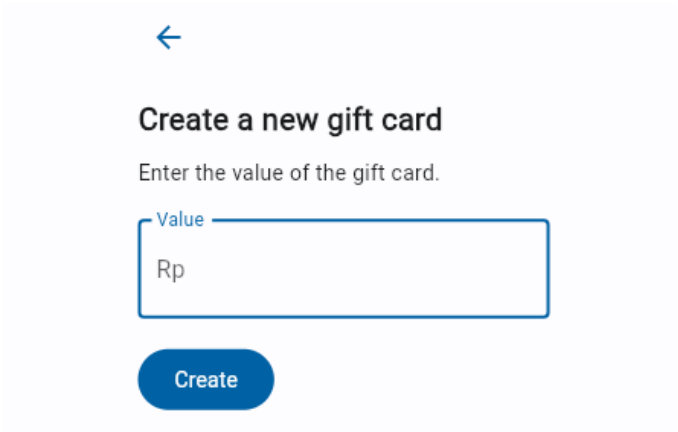

Fig. 7. Halaman Pembuatan *Gift Card*

Setelah pengguna memasukan nominal dan mengirimkan submisi pembuatan *gift card*, pengguna akan dialihkan pada halaman yang menampilkan hasil dari *gift card*. Halaman ini memberikan informasi dari *gift card* yang perlu dicetak atau diproduksi. Pada halaman ini ditampilkan informasi nominal, kode unik yang berlaku, dan kode Quick Response (QR) yang berisikan tanda tangan digital yang digunakan untuk mengaktivasi *gift card*. Berikut merupakan tampilan antarmuka pada halaman ini.

Fig. 8. Halaman Hasil Pembuatan *Gift Card*

B. *Halaman Aktivasi Gift Card*

Fitur aktivasi *gift card* memfasilitasi aktivasi *gift card* dengan memasukan kode unik *gift card* yang akan diaktifkan dan memvalidasi kartu tersebut dengan memindai kode QR berisikan tanda tangan digital yang perlu divalidasi. Halaman ini hanya menerima masukan berupa kode *gift card* yang akan dicari dan menampilkan pembacaan kamera apabila kode tersebut ditemukan pada basis data *gift card*. Berikut merupakan tampilan antarmuka pada halaman pembuatan *gift card*.

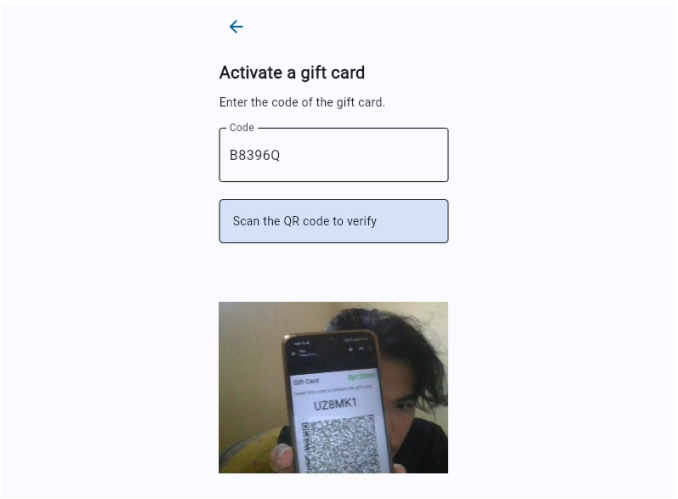

Fig. 9. Halaman Aktivasi *Gift Card*

Proses aktivasi *gift card* yang berhasil, dalam hal ini tanda tangan digital yang terbaca dengan tanda tangan digital yang terdapat dalam basis data sama, akan memberikan tampilan sebagai berikut. Pengguna akan diberikan kode OTP yang dapat digunakan pada saat menukarkan kode *gift card*.

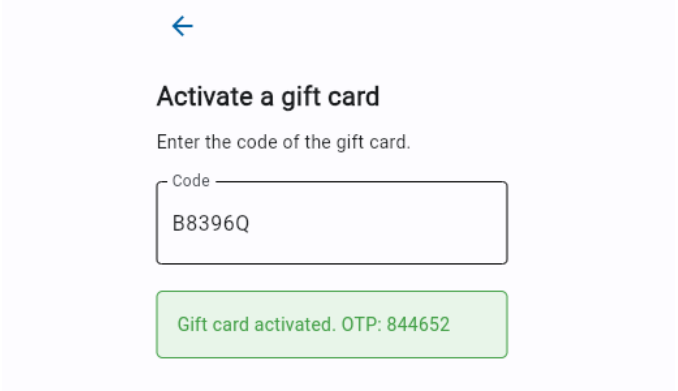

Fig. 10. Halaman Aktivasi *Gift Card* Berhasil

C. *Halaman Penukaran Gift Card*

Fitur penukaran gift card berisikan halaman penukaran gift card yang menerima masukan kode gift card dan kode OTP. Fitur aktivasi gift card memfasilitasi aktivasi gift card dengan memasukkan kode unik gift card yang akan diaktifkan dan memvalidasi kartu tersebut menggunakan OTP. Halaman ini hanya menerima masukan berupa kode gift card yang akan dicari dan divalidasi menggunakan OTP jika kode tersebut ditemukan

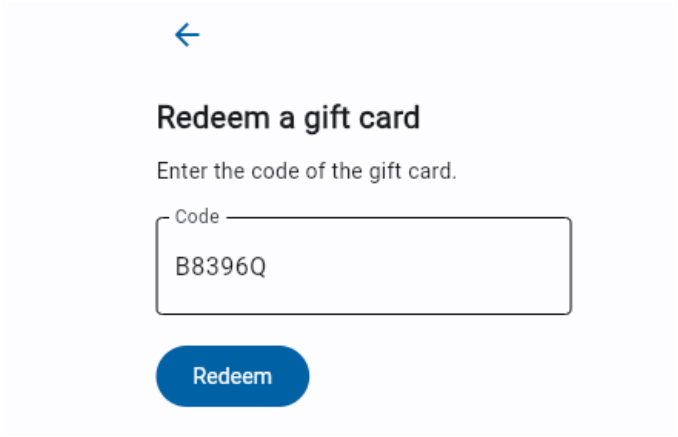

Fig. 11. Halaman Pembuatan *Gift Card*

Berikut merupakan tampilan pada halaman tersebut apabila kode *gift card* yang dimasukan ditemukan. Sistem akan meminta pengguna untuk memasukan kode OTP untuk memvalidasi penukaran *gift card*.

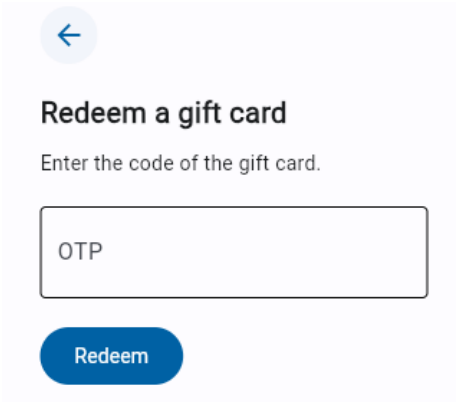

Fig. 12. Halaman Validasi Penukaran *Gift Card*

VI. Pengujian

Pengujian dilakukan terhadap sistem yang sudah diimplementasikan. Pengujian ini dilakukan terhadap fungsionalitas aplikasi pada saat pembuatan, aktivasi, dan penukaran *gift card*.

A. Pengujian Pembuatan Gift Card

TABLE III. Pengujian Pembuatan *Gift Card*

Kasus Uji	***Input***	***Expected Output***	***Actual Output***	**Hasil**
Pengujian nominal 10000	value: 10000	Nominal: Rp10000 Kode unik QR tanda tangan digital	Nominal: Rp10000 Kode unik: AIPADC QR tanda tangan digital:	Diterima
Pengujian nominal 120000	value: 120000	Nominal: Rp120000 Kode unik QR tanda tangan digital	Nominal: Rp120000 Kode unik: 8PMX8R QR tanda tangan digital:	Diterima

Hasil pembuatan *gift card* pada kedua kasus uji menunjukan bahwa pembuatan pembuatan *gift card* berhasil dilengkapi dengan pengamanan tanda tangan digital yang telah diimplementasikan.

B. Pengujian Aktivasi Gift Card

TABLE IV. Pengujian Aktivasi *Gift Card*

Kasus Uji	***Input***	***Expected Output***	***Actual Output***	**Hasil**
Pengujian *gift card* dan tanda tangan digital valid	Code: AIPADC QR tanda tangan digital: (kode kartu AIPADC)	Gift card activated, OTP: {OTP}	Gift card activated, OTP: 333440	Diterima
Pengujian *gift card* valid dan tanda tangan tidak valid	Code: 8PMX8R QR tanda tangan digital: (kode kartu AIPADC)	Invalid signature, gift card cannot be activated	Invalid signature, gift card cannot be activated	Diterima
Pengujian *gift card* sudah pernah diaktivasi	Code: AIPADC	Gift card already activated	Gift card already activated	Diterima
Pengujian *gift card* tidak ditemukan	Code: UZ8MK1	Gift card not found	Gift card not found	Diterima

Hasil aktivasi *gift card* pada kedua kasus uji menunjukan bahwa aktivasi *gift card* berhasil diamankan dengan menggunakan validasi tanda tangan digital. Pengujian aktivasi yang berhasil akan mengembalikan kode OTP yang dapat digunakan oleh pengguna pada saat akan menukarkan *gift card*.

C. Pengujian Penukaran Gift Card

TABLE V. PENGUJIAN PENUKARAN *GIFT CARD*

Kasus Uji	*Input*	*Expected Output*	*Actual Output*	Hasil
Pengujian *gift card* valid dan OTP tidak valid	Code: AIPADC OTP: 215403	Invalid OTP, gift card cannot be redeemed	Invalid OTP, gift card cannot be redeemed	Diterima
Pengujian *gift card* dan OTP valid	Code: AIPADC OTP: 333440	Gift card redeemed successfully	Gift card redeemed successfully	Diterima
Pengujian *gift card* sudah pernah ditukarkan	Code: AIPADC	Gift card already redeemed	Gift card already redeemed	Diterima
Pengujian *gift card* belum diaktivasi	Code: 8PMX8R	Gift card not activated, please activate first	Gift card not activated, please activate first	Diterima
Pengujian *gift card* tidak ditemukan	Code: UZ8MK1	Gift card not found	Gift card not found	Diterima

Hasil penukaran *gift card* pada kedua kasus uji menunjukan bahwa penukaran *gift card* berhasil diamankan dengan menggunakan validasi OTP. Dalam hal ini, OTP berhasil menjadi lapisan keamanan tambahan pada proses penukaran *gift card*.

Secara keseluruhan, hasil pengujian menunjukan bahwa implementasi dari eksperimen ini telah berhasil membuktikan bahwa penggunaan lapisan keamanan tambahan berupa kode One-Time Password (OTP) dan tanda tangan digital (*digital signature*) efektif dalam meningkatkan keamanan *gift card*. Namun, untuk penerapan sistem dalam skala yang lebih besar, diperlukan pengembangan yang lebih lanjut terhadap aksesibilitas dan juga penerapan infrastruktur basis data yang terenkripsi sehingga dapat mengelola penyimpanan *gift card* dengan lebih baik dan aman.

VII. KESIMPULAN

Penelitian ini menunjukkan bahwa implementasi One-Time Password (OTP) dan Digital Signature RSA-PKCS#1 v1.5 pada *physical gift card* memberikan peningkatan signifikan dalam keamanan dibandingkan dengan metode konvensional yang sering digunakan. Penggunaan OTP sebagai lapisan keamanan tambahan terbukti efektif dalam mencegah penggunaan tidak sah atau pencurian gift card. OTP adalah kode unik yang dihasilkan secara acak dan hanya berlaku untuk satu kali transaksi atau dalam jangka waktu yang sangat singkat, sehingga setiap kali pengguna ingin mengakses sistem atau menukarkan gift card, mereka harus memasukkan kata sandi yang berbeda. Ini secara drastis mengurangi risiko akses yang tidak sah akibat kata sandi yang dicuri atau ditebak.

Penerapan tanda tangan digital menggunakan algoritma RSA-PKCS#1 v1.5 dan SHA-3 juga memberikan verifikasi keaslian dan integritas data yang sangat diperlukan untuk mencegah manipulasi atau pemalsuan data *gift card*. *Digital signature* memastikan bahwa setiap transaksi atau aktivasi *gift card* dapat diverifikasi keasliannya, sehingga meningkatkan keamanan dan kepercayaan pengguna terhadap sistem.

Hasil pengujian menunjukkan bahwa sistem yang menggunakan validasi OTP dan tanda tangan digital mampu memberikan perlindungan yang lebih baik terhadap penyalahgunaan dan pemalsuan *gift card*. *Gift card* hanya dapat diaktivasi dan ditukarkan jika OTP dan tanda tangan digitalnya valid, sehingga sistem ini berhasil mencegah pencurian atau penggunaan tidak sah. Namun, penelitian ini juga menemukan bahwa untuk penerapan dalam skala yang lebih besar, diperlukan pengembangan lebih lanjut terhadap aksesibilitas dan infrastruktur basis data yang terenkripsi. Hal ini penting untuk memastikan manajemen penyimpanan *gift card* yang lebih baik dan aman, serta untuk mendukung skala pengguna yang lebih luas.

PRANALA GITHUB

Berikut merupakan pranala yang berisikan kode program pada platform Github.

zshnrg/secured-gift-card: Sistem produksi dan penggunaan gift card yang memiliki lapisan keamanan dan autentikasi tambahan untuk melindungi konsumen dan perusahaan dari penyalahgunaan gift card. (github.com)

ACKNOWLEDGMENT *(Heading 5)*

Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Rinaldi Munir, instruktur kami yang terhormat dalam mata kuliah Kriptografi dan Koding, atas bimbingan dan keahliannya yang sangat berharga. Dengan arahan dan dukungan beliau, penulis dapat menyelesaikan penelitian ini dengan baik.

Penulis juga mengucapkan terima kasih kepada komunitas akademik dan institusi terkait yang telah menyediakan sumber daya dan fasilitas yang diperlukan dalam menyelesaikan penelitian ini dengan sukses. Dukungan mereka sangat berarti bagi kelancaran proses penelitian ini. Tidak lupa, penulis juga ingin memberikan apresiasi kepada komunitas pengembang library, terutama pengembang Flet, yang telah menyediakan *framework* pengembangan perangkat lunak yang mudah digunakan. Kontribusi mereka sangat membantu dalam pengembangan aspek teknis penelitian ini.

Terakhir, penulis menyampaikan apresiasi setinggi-tingginya kepada semua pihak yang telah memberikan kontribusi dan dukungan dalam berbagai bentuknya. Tanpa dukungan dari teman-teman dan pembimbing, makalah ini tidak akan terwujud. Terima kasih atas semua bantuan dan dukungan yang telah diberikan.

References

[1] B. Kaliski, “RFC 2313: PKCS #1: RSA Encryption Version 1.5,” RFC Editor, Mar. 1998. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2313. [Accessed: Jun. 12, 2024].

[2] Munir, R. “SHA-3 (Keccak)”. 2024

[3] Munir, R. “Tanda Tangan Digital”. 2024

[4] R. Steinfeld and P. Hawkes, Eds., Information Security and Privacy: 15th Australasian Conference, ACISP 2010, Sydney, Australia, July 5-7, 2010, Proceedings. Berlin, Germany: Springer, 2010..

[5] Ticketmaster, “How to buy and use a gift card ?” 2024. [Online]. Available: https://help.ticketmaster.fr/hc/en-us/articles/360007186514-How-to-buy-and-use-a-gift-card. [Accessed: Jun. 12, 2024].

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, 12 Juni 2024

Rozan Ghosani 18221121