

**PERANCANGAN SISTEM *MICROPAYMENT* MENGGUNAKAN
TEKNOLOGI *NEAR FIELD COMMUNICATION***

Draft Laporan Tugas Akhir II

**Disusun sebagai syarat kelulusan mata kuliah
IF4092/Tugas Akhir II**

**Oleh
Arina Listyarini Dwiastuti
NIM: 13512006**



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO & INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
Agustus 2017**

**PERANCANGAN SISTEM *MICROPAYMENT* MENGGUNAKAN
TEKNOLOGI *NEAR FIELD COMMUNICATION***

Draft Laporan Tugas Akhir II

Oleh

Arina Listyarini Dwiastuti

NIM: 13512006

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Bandung, 8 Agustus 2017

Mengetahui,

Pembimbing,

Riza Satria Perdana

NIP. 19700609 199512 1 002

DAFTAR ISI

| | |
|---|----------|
| BAB I PENDAHULUAN..... | 1 |
| I.1 Latar Belakang..... | 1 |
| I.2 Rumusan Masalah..... | 2 |
| I.3 Tujuan | 2 |
| I.4 Batasan Masalah | 2 |
| I.5 Metodologi..... | 2 |
| BAB II STUDI LITERATUR | 4 |
| II.1 <i>Near Field Communication</i> (NFC)..... | 4 |
| II.1.1 Karakteristik NFC | 4 |
| II.1.2 Mode Operasi NFC | 5 |
| II.1.3 Arsitektur NFC..... | 6 |
| II.2 Perangkat Keras NFC | 9 |
| II.2.1 <i>Tag</i> NFC..... | 9 |
| II.2.2 <i>Smartphone</i> dengan NFC | 12 |
| II.2.3 <i>NFC Reader/Writer</i> | 12 |
| II.3 Pengembangan Aplikasi NFC..... | 13 |
| II.3.1 CommandAPDU dan ResponseAPDU | 13 |
| II.3.2 Pengembangan Aplikasi NFC pada Platform Java | 14 |
| II.3.3 Pengembangan Aplikasi NFC pada Platform Android | 17 |
| II.4 Kriptografi | 19 |
| II.4.1 Enkripsi Simetrik dan Asimetrik..... | 19 |
| II.4.2 <i>Digital Signature</i> | 20 |

| | | |
|---|---|-----------|
| II.4.3 | Algoritma AES (<i>Advanced Encryption Standard</i>)..... | 21 |
| II.4.4 | Algoritma ECC (<i>Elliptic Curve Cryptography</i>) dan ECDSA (<i>Elliptic Curve Digital Signature Algorithm</i>)..... | 21 |
| II.4.5 | Algoritma PBKDF2 | 21 |
| BAB III ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM MICROPAYMENT DENGAN TEKNOLOGI NFC..... | | 22 |
| III.1 | Analisis Kebutuhan dan Keamanan SISMIC | 22 |
| III.1.1 | Analisis Kebutuhan SISMIC | 22 |
| III.1.2 | Analisis Kebutuhan Keamanan SISMIC | 24 |
| III.2 | Perancangan SISMIC | 25 |
| III.2.1 | Operasi Aplikasi SISMIC | 25 |
| III.2.2 | Penyimpanan dan Struktur Data SISMIC | 39 |
| III.2.3 | Rancangan Keamanan SISMIC | 57 |
| BAB IV IMPLEMENTASI | | 61 |
| IV.1 | Arsitektur SISMIC | 61 |
| IV.2 | Pemilihan Kakas dan <i>Library</i> | 62 |
| IV.3 | Lingkungan Implementasi | 62 |
| IV.4 | Hasil Implementasi | 62 |
| IV.4.1 | Tampilan Hasil Implementasi | 63 |

DAFTAR LAMPIRAN

| | |
|---|-----------|
| Lampiran A. Contoh Judul Lampiran..... | 74 |
| A.1 Membaca Data <i>Reader/Writer Block</i> dari Tag NFC | 74 |
| A.2 Membaca Data <i>Value Block</i> dari Tag NFC | 74 |
| A.3 Menulis Data ke <i>Reader/Writer Block Tag</i> NFC | 75 |
| A.4 Menulis Data ke <i>Value Block Tag</i> NFC | 76 |
| Lampiran B. Diagram Komunikasi SISMIC | 77 |
| B.1 Transaksi <i>Top-Up</i> Saldo Melalui ATM | 77 |
| B.2 Transaksi <i>Top-Up</i> Saldo Melalui <i>Merchant</i> | 77 |
| B.3 Transaksi Pembelian Melalui <i>Merchant</i> | 78 |
| B.4 Lihat Saldo Melalui ATM | 78 |
| B.5 Lihat Saldo Melalui <i>Merchant</i> | 78 |
| B.6 Lihat Masa Berlaku Kartu SISMIC Melalui ATM | 79 |
| B.7 Lihat Masa Berlaku Kartu SISMIC Melalui <i>Merchant</i> | 79 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar II-1 Arsitektur Mode <i>Peer-to-Peer</i> | 6 |
| Gambar II-2 Arsitektur Mode <i>Reader/Writer</i> | 8 |
| Gambar II-3 Arsitektur Mode <i>Card Emulation</i> | 9 |
| Gambar III-1 Diagram <i>Use Case</i> SISMIC | 23 |
| Gambar III-2 Diagram Aktivitas Transaksi <i>Top-Up</i> Saldo Melalui <i>Smartphone</i> . | 26 |
| Gambar III-3 Diagram Aktivitas Transaksi <i>Top-Up</i> Melalui Mesin ATM..... | 28 |
| Gambar III-4 Diagram Aktivitas Transaksi <i>Top-Up</i> Melalui <i>Merchant</i> | 30 |
| Gambar III-5 Diagram Aktivitas Transaksi Pembelian Melalui <i>Smartphone</i> | 31 |
| Gambar III-6 Diagram Aktivitas Transaksi Pembelian Melalui <i>Merchant</i> | 33 |
| Gambar III-7 Diagram Aktivitas Transaksi Pembelian dengan Parameter Melalui <i>Merchant</i> | 34 |
| Gambar III-8 Diagram Aktivitas Lihat Saldo SISMIC Melalui <i>Smartphone</i> | 35 |
| Gambar III-9 Diagram Aktivitas Lihat Saldo SISMIC Melalui ATM..... | 36 |
| Gambar III-10 Diagram Aktivitas Lihat Saldo SISMIC Melalui <i>Merchant</i> | 36 |
| Gambar III-11 Diagram Aktivitas Lihat Riwayat Transaksi..... | 37 |
| Gambar III-12 Diagram Aktivitas Lihat Masa Berlaku Kartu SISMIC Melalui <i>Smartphone</i> | 38 |
| Gambar III-13 Diagram Aktivitas Lihat Masa Berlaku SISMIC Melalui ATM .. | 38 |
| Gambar III-14 Diagram Aktivitas Lihat Masa Berlaku Kartu SISMIC Melalui <i>Merchant</i> | 39 |
| Gambar III-15 Diagram <i>Entity-Relationship</i> SISMIC | 45 |
| Gambar III-16 Basisdata SISMIC | 46 |

| | |
|---|----|
| Gambar III-17 Basisdata <i>Merchant</i> | 47 |
| Gambar III-18 Basisdata <i>Payment Gateway</i> | 47 |
| Gambar III-19 Diagram Kelas ATM..... | 49 |
| Gambar III-20 Diagram Kelas <i>Merchant</i> | 51 |
| Gambar III-21 Diagram Kelas SISMIC | 53 |
| Gambar III-22 Diagram Kelas <i>Payment Gateway</i> | 55 |
| Gambar III-23 Diagram Kelas <i>Merchant</i> | 56 |
| Gambar IV-1 Arsitektur SISMIC..... | 61 |
| Gambar IV-2 Menu Utama Aplikasi ATM..... | 64 |
| Gambar IV-3 Menu Lihat Saldo Aplikasi ATM | 64 |
| Gambar IV-4 Menu Lihat Masa Berlaku Aplikasi ATM..... | 65 |
| Gambar IV-5 Menu Top-Up Saldo Aplikasi ATM..... | 65 |
| Gambar IV-6 Menu Top-Up Saldo Aplikasi ATM Setelah Top-Up | 66 |
| Gambar IV-7 Menu Utama Aplikasi <i>Merchant</i> | 67 |
| Gambar IV-8 Menu Beli Barang I Aplikasi <i>Merchant</i> | 68 |
| Gambar IV-9 Menu Beli Barang I Aplikasi <i>Merchant</i> Setelah Transaksi | 68 |
| Gambar IV-10 Menu Beli Barang II Aplikasi <i>Merchant</i> | 69 |
| Gambar IV-11 Menu Beli Barang II Aplikasi <i>Merchant</i> Setelah Transaksi..... | 69 |
| Gambar IV-12 Menu Top-Up Saldo Aplikasi <i>Merchant</i> | 70 |
| Gambar IV-13 Menu Top-Up Saldo Aplikasi <i>Merchant</i> Setelah <i>Top-Up</i> | 70 |
| Gambar IV-14 Menu Lihat Saldo Aplikasi <i>Merchant</i> | 71 |
| Gambar IV-15 Menu Lihat Masa Berlaku Aplikasi <i>Merchant</i> | 71 |
| Gambar B-1 Diagram Komunikasi Transaksi <i>Top-Up</i> Saldo Melalui ATM..... | 77 |
| Gambar B-2 Diagram Komunikasi Transaksi <i>Top-Up</i> Saldo Melalui <i>Merchant</i> . | 77 |

| | |
|--|----|
| Gambar B-3 Diagram Komunikasi Transaksi Pembelian Saldo Melalui <i>Merchant</i> | 78 |
| Gambar B-4 Diagram Komunikasi Lihat Saldo Melalui ATM | 78 |
| Gambar B-5 Diagram Komunikasi Lihat Saldo Melalui <i>Merchant</i> | 79 |
| Gambar B-6 Diagram Komunikasi Masa Berlaku Kartu Melalui ATM..... | 79 |
| Gambar B-7 Diagram Komunikasi Masa Berlaku Kartu Melalui <i>Merchant</i> | 80 |

DAFTAR TABEL

| | |
|---|----|
| Tabel II-1 Perbandingan Tipe <i>Tag</i> NFC | 9 |
| Tabel II-2 Format <i>Value Block</i> | 11 |
| Tabel II-3 Struktur Instruksi APDU..... | 13 |
| Tabel II-4 Instruksi APDU <i>Load</i> Kunci A atau Kunci B..... | 14 |
| Tabel II-5 Instruksi APDU Autentikasi <i>Block</i> | 15 |
| Tabel II-6 Penjabaran <i>Bytes</i> Autentikasi Data | 15 |
| Tabel II-7 Program Kecil Membaca <i>Reader/Writer Block</i> | 16 |
| Tabel III-1 Konfigurasi Sektor dan Blok Kartu SISMIC..... | 40 |
| Tabel III-2 Hak Akses Kartu SISMIC | 43 |
| Tabel A.1 Instruksi APDU Baca <i>Reader/Writer Block</i> | 74 |
| Tabel A.2 Instruksi APDU Baca <i>Value Block</i> | 75 |
| Tabel A.3 Instruksi APDU Tulis Data ke <i>Reader/Writer Block</i> | 75 |
| Tabel A.4 Instruksi APDU Tulis Data ke <i>Value Block</i> | 76 |

BAB I

PENDAHULUAN

I.1 Latar Belakang

Manusia memiliki kebutuhan hidup yang harus dipenuhi. Kebutuhan hidup dapat dipenuhi dengan melakukan transaksi dengan pihak lain. Semakin banyak orang yang melakukan transaksi, semakin rumit transaksi terjadi. Di zaman yang terus berubah, perkembangan transaksi semakin canggih dan mudah. Transaksi dengan jumlah yang kecil namun sering dilakukan dapat disebut sebagai transaksi *micropayment* (WhatIs.com, 2016).

Micropayment biasanya terhubung dengan koneksi internet, walaupun bisa saja tanpa terhubung koneksi internet. Di Indonesia, penggunaan *micropayment* sudah banyak dijumpai. Contohnya adalah pembayaran tol dengan e-toll Mandiri, pembayaran belanja di mini market dengan flazz BCA, serta pembayaran tiket transportasi *commuter line* di Jabodetabek dengan kartu Commet. Kartu-kartu tersebut menggunakan teknologi *Near Field Communication* (NFC).

Di era teknologi yang semakin canggih, teknologi yang mendukung dan mempermudah komunikasi semakin banyak, salah satunya adalah teknologi NFC. NFC adalah teknologi koneksi jarak pendek tanpa kabel yang membuat kebutuhan hidup lebih nyaman dan mudah (NFC Forum, 2016). NFC memiliki manfaat yang dapat digunakan untuk membantu memenuhi kebutuhan sehari-hari seperti berbagi kartu nama, berbagi data, untuk alat pembayaran, sebagai tiket transportasi, membagikan suatu informasi, dan lain-lain.

Transaksi *micropayment* membutuhkan keamanan dan kehandalan yang baik. Begitu juga jika *micropayment* tersebut memanfaatkan teknologi NFC, NFC tersebut harus aman dan handal. Jika tidak aman, pihak yang tidak berhak melakukan transaksi dapat melakukan transaksi dan merugikan pihak yang lain. Jika tidak handal, transaksi tersebut dapat gagal dilakukan.

I.2 Rumusan Masalah

Ruang lingkup untuk tugas akhir ini adalah perancangan sistem *micropayment*. Rumusan masalah yang akan dibahas pada tugas akhir ini adalah bagaimana rancangan sistem *micropayment* menggunakan NFC yang aman. Aman yang dimaksud adalah tidak mudah disadap oleh penyerang.

I.3 Tujuan

Tujuan dari tugas akhir ini adalah dapat membuat rancangan sistem *micropayment* menggunakan NFC yang aman. Harapannya, rancangan sistem *micropayment* menggunakan NFC pada tugas akhir ini dapat diimplementasikan pada kehidupan sehari-hari dan dapat mengatasi persoalan yang ada.

I.4 Batasan Masalah

Batasan masalah untuk tugas akhir ini adalah sebagai berikut.

1. Perancangan sistem *micropayment* menggunakan NFC.
2. Menerapkan metode pengamanan yang sesuai pada transaksi *micropayment* yang dirancang.

I.5 Metodologi

Metodologi yang digunakan pada Tugas Akhir ini adalah:

1. Studi Literatur

Studi literatur dilakukan dengan mempelajari berbagai teori mengenai transaksi *micropayment* dan teknologi NFC melalui jurnal, buku, situs, dan artikel.

2. Penentuan Kebutuhan

Penentuan kebutuhan dilakukan untuk menentukan kebutuhan sistem yang akan dibuat.

3. Eksplorasi perangkat keras yang mendukung NFC

Eksplorasi dilakukan dengan mempelajari perangkat keras yang mendukung NFC. Membaca dan memahami berbagai jurnal, buku, situs, dan artikel dapat membantu eksplorasi perangkat keras ini. Dengan melakukan eksplorasi ini diharapkan dapat menemukan teknik terbaik untuk memprogram NFC.

4. Eksplorasi teknik *programming* NFC

Eksplorasi dilakukan dengan mempelajari dan mencoba berbagai teknik *programming* yang dapat digunakan untuk memprogram NFC. Buku, jurnal, situs, dan artikel dapat membantu eksplorasi teknik *programming* NFC. Dengan melakukan eksplorasi ini diharapkan dapat menemukan teknik terbaik untuk memprogram NFC.

5. Perancangan

Perancangan dilakukan untuk menentukan bagaimana sistem *micropayment* yang akan dibuat bekerja, serta perangkat keras apa saja yang akan digunakan.

6. Implementasi

Implementasi dilakukan untuk menerapkan hasil perancangan sistem *micropayment*.

7. Pengujian

Pengujian dilakukan untuk menguji hasil implementasi sistem *micropayment*.

BAB II

STUDI LITERATUR

II.1 *Near Field Communication* (NFC)

Pada subbab NFC ini, akan dibahas mengenai karakteristik NFC, mode operasi NFC, dan arsitektur NFC.

II.1.1 Karakteristik NFC

NFC memiliki manfaat yang dapat digunakan untuk membantu memenuhi kebutuhan sehari-hari seperti berbagi kartu nama, berbagi data dalam bentuk gambar, pembayaran, dan lain-lain. Manfaat-manfaat NFC tersebut memiliki tujuh karakteristik sebagai berikut (NFC Forum, 2015):

1. Intuitif: NFC dapat digunakan dengan sekali sentuh. Pengguna NFC tidak membutuhkan banyak langkah untuk melakukan beberapa hal seperti pembayaran, berbagi data, konfigurasi *smartphone*, dan lain-lain. Maka dari itu, penggunaan NFC membuat hal-hal tersebut menjadi lebih praktis.
2. Serba guna: NFC tidak hanya digunakan pada bidang teknologi, tapi juga dapat digunakan pada bidang lainnya seperti industri, lingkungan, dan kesehatan. Salah satu contoh penggunaan NFC pada bidang kesehatan adalah penggunaan NFC untuk merekam data medis pasien (Jara, Zamora, & S, 2009).
3. Terbuka dan memiliki standar: NFC memiliki standar ISO, ECMA, ETSI, dan NFC Forum sehingga NFC memiliki standardisasi yang sama secara internasional (Coskun, Ok, & Ozdenizci, 2013).
4. *Technology-enabling*: NFC dapat digunakan untuk mengonfigurasi teknologi lain seperti pengaturan konfigurasi untuk Bluetooth dan WiFi pada telepon selular secara cepat tanpa membutuhkan banyak tahap.

5. Aman: Transmisi data melalui NFC dilakukan dengan jarak pendek secara aman dan dapat diandalkan.
6. Dapat dioperasikan: NFC dapat dioperasikan menggunakan teknologi *contactless smart card* atau kartu tanpa sentuh yang sudah ada untuk pembayaran, konfigurasi telepon selular, dan lain-lain.
7. *Security-ready*: NFC sudah mendukung fitur keamanan pada pengembangan aplikasi NFC.

II.1.2 Mode Operasi NFC

Perangkat keras NFC dapat saling berinteraksi satu sama lain. NFC memiliki tiga mode operasi agar perangkat keras NFC dapat berkomunikasi dengan perangkat lainnya, yaitu mode *peer-to-peer*, mode *reader/writer*, dan mode *card emulation* (Coskun, Ok, & Ozdenizci, 2013).

Pada mode *peer-to-peer*, perangkat keras NFC seperti *smartphone* ber-NFC, dapat saling bertukar data seperti nomor telepon, gambar, video, audio, dan data-data lainnya, dengan *smartphone* ber-NFC lainnya. Pertukaran data ini terjadi dengan pengiriman data melalui kanal *bidirectional half-duplex*. Pengiriman data dengan *bidirectional half-duplex* adalah ketika satu perangkat sedang mengirim data, perangkat yang lain harus mendengarkan dan menunggu data yang dikirim lawannya sampai selesai sebelum melakukan pengiriman data sehingga pengiriman data tidak dapat dilakukan secara bersamaan, tetapi harus saling menunggu satu sama lain sampai selesai.

Sedangkan pada mode *reader/writer*, perangkat keras NFC dapat berkomunikasi dengan *tag* NFC untuk membaca atau menulis data. Ketika perangkat keras NFC berada dalam mode *reader*, perangkat keras NFC yang berperan sebagai *reader* akan membaca data pada *tag* NFC dan menentukan bentuk apa data itu sehingga aplikasi NFC akan mengetahui hal apa yang harus dilakukan selanjutnya. Pada mode *writer*, perangkat keras NFC yang berperan sebagai *writer* akan menulis data ke *tag* NFC.

Terakhir pada mode *card emulation*, perangkat keras NFC dapat berfungsi seperti *contactless smart card* sehingga perangkat keras tersebut dapat digunakan untuk transaksi pembelian tiket dan akses transportasi umum. Dalam mode ini, *reader/writer* NFC dapat berkomunikasi dengan membaca atau menulis data pada perangkat keras NFC yang berada pada mode *card emulation* sehingga pada akhirnya terjadilah transaksi.

II.1.3 Arsitektur NFC

Masing-masing mode operasi NFC yang ada pada sub-bab sebelumnya memiliki arsitektur dengan spesifikasi teknis untuk berkomunikasi dengan perangkat NFC, baik mode *peer-to-peer*, *reader/writer*, maupun *card emulation* (Coskun, Ok, & Ozdenizci, 2013) .

II.1.3.1 Arsitektur Mode *Peer-to-Peer*

Mode *peer-to-peer* memiliki spesifikasi teknis untuk level aplikasi, protokol analog, protokol digital, *Logical Link Control Protocol* (LLCP), protokol NFC Forum, *protocol bindings*, dan *Simple NDEF Exchange Protocol* (SNEP) (NFC Forum, 2015) dengan arsitektur seperti pada Gambar II-1 Arsitektur Mode *Peer-to-Peer* berikut ini (Coskun, Ok, & Ozdenizci, 2013).

| | |
|------------------|--------------------------|
| Aplikasi | |
| SNEP | Protokol NFC Forum |
| | <i>Protocol Bindings</i> |
| LLCP | |
| Protokol Digital | |
| Protokol Analog | |

Gambar II-1 Arsitektur Mode *Peer-to-Peer*

Arsitektur mode *peer-to-peer* memiliki rincian sebagai berikut:

- Protokol digital mengatur pemetaan sinyal digital untuk komunikasi antar perangkat NFC dan mengikatnya ke protokol LLCP dengan cara bit *level*

coding, *bit rates*, *frame formats*, dan kumpulan perintah-perintah pada level digital. Sedangkan protokol analog menspesifikasikan pemetaan sinyal analog pada perangkat NFC.

- LLCP menyediakan lima layanan penting pada perangkat NFC yaitu *connectionless transport*; *connection-oriented transport*; *link activation*, *supervision*, dan *deactivation*; *asynchronous balanced communication*; dan *protocol multiplexing*. Cara kelima layanan ini bekerja adalah seperti berikut ini:
 - *Connectionless transport* kurang dapat diandalkan dibandingkan *connection-oriented transport* karena *connectionless transport* menyediakan data yang belum dikenal. Sedangkan *connection-oriented transport* menjamin data yang dikirimkan sudah terurut dan dikontrol menggunakan protokol *sliding window*.
 - *Link activation*, *supervision*, dan *deactivation* membuat sebuah hubungan LLCP *link* agar komunikasi antar perangkat keras NFC dapat terjadi dalam jarak yang dapat ditangani oleh LLCP. Setelah itu, komunikasi yang telah dibuat akan diawasi dan di non-aktifkan jika komunikasi telah selesai dilakukan.
 - *Asynchronous balanced communication* membuat perangkat keras NFC dapat mengirim dan meminta data dari perangkat lainnya. Hal lainnya adalah layanan ini dapat menginisiasi, mengawasi, dan memulihkan diri dari kesalahan yang terjadi. Sedangkan *protocol multiplexing* dapat membuat LLCP menangani protokol-protokol lain pada waktu yang sama.
 - *Protocol bindings* mengelola bagaimana protokol NFC Forum dapat saling terikat dengan protokol-protokol NFC lainnya.
- SNEP adalah bagaimana cara perangkat NFC saling bertukar pesan dalam bentuk NFC *Data Exchange Format* (NDEF). Protokol ini menggunakan layanan *connection-oriented transport* agar pertukaran data dapat diandalkan.

- Level aplikasi menjalankan SNEP, protokol NFC Forum, ataupun protokol-protokol lainnya.

II.1.3.2 Arsitektur Mode *Reader/Writer*

Mode *reader/writer* memiliki spesifikasi teknis untuk protokol analog, protokol digital, operasi *tag* NFC Forum, aplikasi NFC *Data Exchange Format* (NDEF), dan aplikasi non-NDEF dengan susunan arsitektur seperti pada Gambar II-2 berikut ini (Coskun, Ok, & Ozdenizci, 2013).

| | |
|------------------------------|-------------------|
| NDEF | Aplikasi Non-NDEF |
| Operasi <i>Tag</i> NFC Forum | |
| Protokol Digital | |
| Protokol Analog | |

Gambar II-2 Arsitektur Mode *Reader/Writer*

Arsitektur Mode *Reader/Writer* memiliki rincian sebagai berikut:

- Protokol analog ditentukan oleh karakteristik pada masing-masing perangkat keras NFC. Protokol analog menentukan jarak yang dapat ditangani oleh perangkat tersebut untuk berkomunikasi. Sedangkan protokol digital akan membuat blok-blok yang digunakan untuk komunikasi.
- Operasi *tag* NFC Forum digunakan untuk mengoperasikan *tag-tag* NFC Forum. *Tag* NFC Forum terdiri dari empat tipe, yaitu tipe 1, tipe 2, tipe 3, dan tipe 4. Semua *tag* ini dapat dibaca atau ditulis menggunakan data yang berformat NDEF.
- Aplikasi NDEF adalah aplikasi yang menangani data berformat NDEF, sedangkan aplikasi non-NDEF menangani *tag* NFC yang memiliki data dalam bentuk non-NDEF seperti tiket transportasi.

II.1.3.3 Arsitektur Mode *Card Emulation*

Mode *card emulation* memiliki spesifikasi teknis untuk protokol analog, protokol digital, dan level aplikasi dengan arsitektur seperti pada Gambar II-3 berikut ini (Coskun, Ok, & Ozdenizci, 2013). Perangkat keras NFC yang bermode *card emulation* menggunakan protokol analog dan digital yang mirip dan sepadan dengan protokol yang dimiliki oleh *smart card*.

| |
|------------------|
| Aplikasi |
| Protokol Digital |
| Protokol Analog |

Gambar II-3 Arsitektur Mode *Card Emulation*

II.2 Perangkat Keras NFC

Perangkat keras NFC yang digunakan pada tugas akhir ini adalah *tag* NFC, *smartphone* yang memiliki fitur NFC, dan NFC *reader/writer*.

II.2.1 *Tag* NFC

Seperti yang sudah dijelaskan sebelumnya, *tag* NFC terdiri dari empat tipe yaitu tipe 1, tipe 2, tipe 3, dan tipe 4. *Tag* NFC ini mudah didapatkan karena telah dijual secara bebas. Perbedaan *tag* NFC Forum tipe 1, 2, 3, dan 4 ini terletak pada standar ISO/IEC yang digunakan, konfigurasi, kapasitas memori, dan kecepatan data transfer. Perbedaan tipe *tag* NFC beserta contoh *tag* dapat dilihat pada Tabel II-1 berikut ini (Chen, Lin, & Yang, 2014).

Tabel II-1 Perbandingan Tipe *Tag* NFC

| | <i>Tag</i> Tipe 1 | <i>Tag</i> Tipe 2 | <i>Tag</i> Tipe 3 | <i>Tag</i> Tipe 4 |
|---------|-------------------|-------------------|----------------------------|-------------------|
| Standar | ISO/IEC 14443A | ISO/IEC 14443A | JIS-X 6319-4 (Sony FeliCa) | ISO/IEC 14443A/B |

| | <i>Tag Tipe 1</i> | <i>Tag Tipe 2</i> | <i>Tag Tipe 3</i> | <i>Tag Tipe 4</i> |
|----------------------------|---|---|---|--|
| Konfigurasi | <i>Read-only</i> atau <i>rewritable</i> | <i>Read-only</i> atau <i>rewritable</i> | Ditentukan oleh manufaktur apakah <i>read-only</i> atau <i>rewritable</i> | Ditentukan oleh manufaktur apakah <i>read-only</i> atau <i>rewritable</i> |
| Kapasitas Memori | 96 B sampai 2 KB | 48 B sampai 2 KB | 1 MB per <i>service</i> | Maksimal 32 KB per <i>service</i> |
| Kecepatan data transfer | 106 kbps | 106 kbps | 212 kbps dan 424 kbps | 106kbps |
| Contoh <i>Tag</i> NFC | Innovision Topaz | NXP MIFARE (selain DESFire) | Sony FeliCa | NXP MIFARE DESFire |

Tag NFC tipe 1 dan 2 bebas dikonfigurasi oleh pengguna apakah *tag* tersebut *read-only* atau *rewritable*. Sedangkan *tag* NFC tipe 3 dan 4 harus dikonfigurasi oleh manufaktur *tag* NFC tersebut apakah *tag* tersebut *read-only* atau *rewritable*. Pada tugas akhir ini, *tag* NFC yang digunakan adalah NXP MIFARE Classic yang berkapasitas 1 KB (MIFARE Classic EV1 1K)

II.2.1.1 *Tag* NFC MIFARE Classic EV 1K

MIFARE Classic EV1 1K ini mendukung *anti-collision* dan dapat ditulis ulang sampai 100.000 kali. Walaupun MIFARE Classic EV1 1K bukan termasuk dalam *tag* NFC standar NFC Forum (tipe 1, tipe 2, tipe 3, atau tipe 4), MIFARE Classic EV 1K tetap dapat digunakan untuk berkomunikasi dengan perangkat keras NFC yang lain (Igoe, Coleman, & Jepson, 2014).

MIFARE Classic EV1 1K memiliki memori dalam bentuk sektor, di mana satu kartu MIFARE Classic EV1 1K memiliki 16 sektor. 1 sektor pada kartu terdiri dari 4 blok di mana 1 bloknya berukuran 16 *bytes*. Sektor ke-0 blok ke-0 merupakan informasi UID kartu dan manufaktur pembuat kartu, sehingga blok ini tidak dapat diubah isinya. UID ada dua versi, yang satu berukuran 4 *bytes*, yang satu berukuran 7 *bytes*. Blok ke-3 dari masing-masing sektor adalah *sector trailer*, yaitu blok yang berisi kunci A, kunci B, dan *bits* untuk kondisi akses dari blok-blok pada sektor tersebut apakah blok ke-0 dari sektor dapat ditulis, atau blok ke-1 dari sektor tersebut dapat dibaca, dan lain-lain. Kunci A dan kunci B berfungsi untuk autentikasi blok sebelum blok tersebut dibaca ataupun ditulis.

Blok pada Mifare Classic dapat berfungsi sebagai *read/write blocks* atau *value blocks*. *Read/write blocks* dapat diisi dengan bilangan hexa berukuran 16 *bytes* yang bisa dikonversi menjadi bilangan desimal biasa maupun karakter ASCII. Sedangkan *value blocks* merupakan blok untuk menyimpan bilangan yang dapat di-*increment* ataupun *decrement*. *Value blocks* dapat dibentuk dengan menuliskan suatu blok dengan format seperti Tabel II-2 berikut ini.

Tabel II-2 Format *Value Block*

| 1 (4 <i>bytes</i>) | 2 (4 <i>bytes</i>) | 3 (4 <i>bytes</i>) | 4 (1 <i>byte</i>) | 5 (1 <i>byte</i>) | 6 (1 <i>byte</i>) | 7 (1 <i>byte</i>) |
|---|---|---------------------|---|----------------------------------|---------------------|---------------------|
| 4 <i>bytes</i> bilangan <i>signed long integer</i> yang berupa nilai <i>value block</i> | Bilangan hasil <i>invert</i> dari nomor 1 | Sama dengan nomor 1 | <i>Address</i> berukuran 1 <i>byte</i> untuk <i>back-up</i> | Hasil <i>invert</i> dari nomor 4 | Sama dengan nomor 4 | Sama dengan nomor 6 |

Setelah suatu blok berisi seperti format pada gambar x, blok tersebut sudah menjadi *value blocks* dan dapat di-*increment* maupun *decrement*.

II.2.2 *Smartphone* dengan NFC

Sudah banyak *smartphone* yang dapat mendukung teknologi NFC dan beredar di pasar. Namun, tidak semua *smartphone* dapat mendukung semua *tag* NFC. *Smartphone* yang digunakan untuk pengerjaan tugas akhir ini adalah Samsung Galaxy Note II yang dapat mendukung hampir semua *tag* NFC seperti Topaz, MIFARE, Sony FeliCa, dan lain-lain.

Smartphone dengan NFC dapat beroperasi dengan mode *peer-to-peer*, yaitu saling bertukar data dengan *smartphone* lain. Ketika *smartphone* ber-NFC beroperasi dengan mode *reader/writer*, *smartphone* dapat membaca maupun menulis sebuah *tag* NFC. Sedangkan *smartphone* yang berada pada mode *card emulation* dapat dibaca dan ditulis oleh sebuah NFC *Reader/Writer*.

Di Indonesia, *smartphone* sudah dimanfaatkan dengan teknologi NFC oleh bank. Beberapa contohnya adalah aplikasi Mandiri E-Money Isi Ulang dari PT Bank Mandiri (Persero) Tbk untuk isi ulang kartu e-money dari Bank Mandiri dan aplikasi BCA Mobile dari PT Bank Central Asia Tbk untuk melihat saldo kartu flazz BCA dari Bank BCA. Kartu e-money dan flazz BCA tersebut dapat digunakan untuk melakukan berbagai macam transaksi seperti membeli barang di mini market dan bayar tol. Selain bank, salah satu operator telekomunikasi seluler di Indonesia, Telkomsel, membuat aplikasi T-Wallet yang dapat digunakan untuk transaksi pembayaran dan isi ulang TCASH, layanan uang elektronik dari telkomsel.

II.2.3 NFC *Reader/Writer*

NFC *reader/writer* adalah suatu perangkat keras yang dapat digunakan untuk membaca data pada suatu perangkat keras NFC dan menuliskan suatu data pada perangkat keras NFC. NFC *reader/writer* dapat membaca *tag* NFC dan dengan sedikit modifikasi, NFC *reader/writer* juga dapat membaca data dari *smartphone* ber-NFC. Lalu, NFC *reader/writer* dapat menulis data ke *tag* NFC dan dengan sedikit modifikasi, NFC *reader/writer* juga dapat menulis data ke *smartphone* ber-NFC.

NFC *Reader/Writer* yang digunakan untuk pengerjaan tugas akhir adalah ACR122U yang diproduksi oleh Advanced Card Systems Holdings Limited (ACS), perusahaan teknologi yang bergerak di bidang kartu dan *reader*. ACR122U tidak hanya dapat membaca ataupun menulis *tag* NFC MIFARE Classic, tetapi juga semua tipe *tag* NFC lainnya (tipe 1, tipe 2, tipe 3, dan tipe 4).

II.3 Pengembangan Aplikasi NFC

Pengembangan aplikasi NFC pada tugas akhir ini meliputi CommandAPDU, ResponseAPDU, dan pengembangan aplikasi pada platform Java dan Android.

II.3.1 CommandAPDU dan ResponseAPDU

Application Protocol Data Unit atau APDU adalah sebuah unit komunikasi antara *smart card reader* dan *smart card* yang digunakan untuk pertukaran data atau komunikasi antara NFC *reader* dan *tag* NFC (Rankl & Effing, 2010). Pada umumnya, instruksi APDU memiliki struktur seperti Tabel II-3 berikut ini dengan ukuran yang berbeda-beda. APDU menggunakan bilangan *hexadecimal*. Instruksi APDU berbeda-beda untuk tiap operasi maupun jenis *smart card reader*. Untuk tugas akhir ini, instruksi APDU yang digunakan adalah APDU untuk NFC *reader* ACR122U dan MIFARE Classic 1K.

Tabel II-3 Struktur Instruksi APDU

| Command Header | | | | Command Body | | |
|----------------|-----|----|----|--------------|------|----|
| CLA | INS | P1 | P2 | Lc | Data | Le |

CommandAPDU adalah sebuah kelas yang terdapat pada API Java Smart Card I/O. CommandAPDU digunakan untuk mengirimkan instruksi ke *tag* NFC dan setelah itu, *tag* NFC akan memberikan respon. Pada tugas akhir ini, CommandAPDU digunakan untuk membaca data dari *tag* NFC dan menuliskan data ke *tag* NFC pada platform Java.

ResponseAPDU adalah sebuah kelas yang terdapat pada API Java Smart Card I/O. ResponseAPDU digunakan untuk mendapatkan respon dari *tag* NFC yang

dikirimkan instruksi CommandAPDU. Pada tugas akhir ini, ResponseAPDU digunakan untuk mendapatkan respon dari *tag* NFC setelah dikirimkan instruksi CommandAPDU.

II.3.2 Pengembangan Aplikasi NFC pada Platform Java

Pengembangan aplikasi NFC pada platform Java akan dilakukan pada kakas Netbeans IDE 8.0. Sebelum membaca data dari atau menulis data ke suatu *reader/writer block* atau *value block tag* NFC, *reader/writer block* atau *value block* tersebut harus diautentikasi terlebih dahulu menggunakan kunci A atau kunci B. Autentikasi dapat dilakukan menggunakan commandAPDU. Untuk melakukan autentikasi *reader/writer block* atau *value block*, kunci A atau kunci B yang akan dipakai untuk autentikasi harus di-load menggunakan commandAPDU. APDU untuk *load* kunci A atau kunci B yang berukuran 11 *bytes* dicantumkan pada Tabel II-4 berikut ini.

Tabel II-4 Instruksi APDU *Load* Kunci A atau Kunci B

| CLA (1 <i>byte</i>) | INS (1 <i>byte</i>) | P1 (1 <i>byte</i>) | P2 (1 <i>byte</i>) | Lc (1 <i>byte</i>) | Data (6 <i>bytes</i>) |
|----------------------|----------------------|-----------------------|---------------------|---------------------|------------------------|
| 0xFF | 0x82 | Struktur kunci (0x00) | Lokasi kunci (0x01) | 0x06 | Kunci A atau kunci B |

APDU tersebut akan memberikan respon berukuran 2 *bytes* yang berisi SW1 (1 *bytes*) dan SW2 (1 *bytes*). Apabila SW1 bernilai 0x90 dan SW2 bernilai 0x00, *load* kunci A atau kunci B telah berhasil. Namun apabila SW1 bernilai 0x63 dan SW2 bernilai 0x00, *load* kunci A atau kunci B gagal.

Setelah berhasil *load* kunci A atau kunci B, autentikasi *block* dapat dilakukan. Instruksi APDU untuk autentikasi *block* ada pada Tabel II-5 berikut.

Tabel II-5 Instruksi APDU Autentikasi *Block*

| CLA (1 byte) | INS (1 byte) | P1 (1 byte) | P2 (1 byte) | Lc (1 byte) | Data (5 bytes) |
|--------------|--------------|-------------|-------------|-------------|---|
| 0xFF | 0x86 | 0x00 | 0x00 | 0x05 | Autentikasi data (penjabaran di Tabel II-6) |

Tabel II-6 Penjabaran *Bytes* Autentikasi Data

| Byte ke-1 | Byte ke-2 | Byte ke-3 | Byte ke-4 | Byte ke-5 |
|-----------|-----------|--|--|---------------------|
| 0x01 | 0x00 | <i>Block</i> yang akan diautentikasi (0x00 ~ 0x3F) | 0x60 untuk autentikasi kunci A atau 0x61 untuk autentikasi kunci B | Lokasi kunci (0x01) |

Seperti APDU *load* kunci, APDU autentikasi *block* juga akan memberikan respon berukuran 2 *bytes* yang berisi SW1 (1 *bytes*) dan SW2 (1 *bytes*). Apabila SW1 bernilai 0x90 dan SW2 bernilai 0x00, autentikasi telah berhasil. Namun apabila SW1 bernilai 0x63 dan SW2 bernilai 0x00, autentikasi gagal. Setelah autentikasi berhasil, baru *tag* NFC dapat dibaca atau ditulis. Lihat lampiran A yang berjudul “Membaca dan Menulis *Tag* NFC” untuk mengetahui bagaimana cara membaca data *reader/writer block* dan *value block* dari *tag* NFC serta menulis data ke *reader/writer block* dan *value block* *tag* NFC.

II.3.2.1 Membaca Data *Reader/Writer Block* dari *Tag* NFC

Instruksi *CommandAPDU* untuk membaca data *reader/writer block* dari *tag* NFC dapat dilihat di lampiran subbab A.1. Contoh program kecil untuk membaca data *reader/writer block* ke-0 dari sector ke-0 dengan *key* A yang bernilai 0x000000000000 dapat dilihat pada Tabel II-7 berikut ini.

Tabel II-7 Program Kecil Membaca *Reader/Writer Block*

```
// Load Key A atau Key B
byte[] load_auth_key_apdu = {
    (byte) 0xFF, (byte) 0x82,
    (byte) 0x00,
    (byte) 0x01,
    (byte) 0x06,
    (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00,
    (byte) 0x00 // key A atau key B, 6 bytes
};
CommandAPDU cmd = new CommandAPDU(load_auth_key_apdu);
ResponseAPDU transmit = channel.transmit(cmd);

int loadKeyResult = transmit.getSW1();

//jika berhasil load key
if (loadKeyResult == 0x90) {
    // Autentikasi Block
    byte[] authenticate_block_apdu = {
        (byte) 0xFF, (byte) 0x86, (byte) 0x00, (byte) 0x00, (byte) 0x05,
        (byte) 0x01, (byte) 0x00
        (byte) 0x00,
        (byte) 0x60, // 0x60 untuk Key A dan 0x61 untuk Key B
        (byte) 0x01
    };
    CommandAPDU cmd = new CommandAPDU(authenticate_block_apdu);
    ResponseAPDU transmit = channel.transmit(cmd);

    int authBlockResult = transmit.getSW1();

    // jika autentikasi block berhasil
    if (authBlockResult == 0x90) {
        //baca isi blocknya
        byte[] read_block_apdu = new byte[] {
            (byte) 0xFF,
            (byte) 0xB0,
            (byte) 0x00,
            (byte) 0x00,
            (byte) 0x10
        };
        read_block_apdu = Operation.concat3Bytes(read_block_apdu, blockB,
        read_block_apdu_last);

        CommandAPDU cmd = new CommandAPDU(read_block_apdu);
        ResponseAPDU transmit = channel.transmit(cmd);

        if (transmit.getSW1() == 0x90) {
            System.out.println("Isi block ke-0 sektor ke-0 adalah " +
```

```

result);
    }
    else{
        System.out.println("Gagal baca isi block");
    }
}
else{
    System.out.println("Gagal autentikasi block");
}
}
else {
    System.out.println("Gagal load key");
}
}

```

II.3.2.2 Membaca Data *Value Block* dari *Tag NFC*

Instruksi CommandAPDU untuk membaca data *value block* dari *tag NFC* dapat dilihat di lampiran subbab A.2.

II.3.2.3 Menulis Data ke *Reader/Writer Block Tag NFC*

Instruksi CommandAPDU untuk menulis data ke *reader/writer block tag NFC* dapat dilihat di lampiran subbab A.3.

II.3.2.4 Menulis Data ke *Value Block Tag NFC*

Instruksi CommandAPDU untuk menulis data ke *value block tag NFC* dapat dilihat di lampiran subbab A.4.

II.3.3 Pengembangan Aplikasi NFC pada Platform Android

Pengembangan Aplikasi NFC pada platform Android akan menggunakan kakas pengembangan Anrdoid Studio. Untuk mengembangkan aplikasi NFC pada platform Android, perlu mengaktifkan *permission* NFC terlebih dahulu pada *manifest* Android. Pada *AndroidManifest.xml*, tambahkan *permission* NFC dengan menambahkan `<uses-permission android:name="android.permission.NFC" />`.

Sama seperti platform Java, sebelum melakukan operasi baca atau tulis ke *blocks tag NFC*, *block* yang ingin dioperasikan harus diautentikasi terlebih dahulu. API yang digunakan adalah `android.nfc.tech` dengan kelas `MifareClassic`. Seperti yang sudah disebutkan sebelumnya, pada tugas akhir ini, *NFC reader* yang

digunakan adalah ACR122U dan *tag* NFC yang digunakan adalah MIFARE Classic 1K. Sehingga, penjelasan studi literatur akan mengacu pada ACR122U dan MIFARE Classic 1K.

Autentikasi dilakukan dengan *method* `authenticateSectorWithKeyA(int sectorIndex, byte[] key)` untuk autentikasi dengan kunci A atau `authenticateSectorWithKeyB(int sectorIndex, byte[] key)` untuk autentikasi dengan kunci B. `sectorIndex` adalah lokasi sektor yang ingin diautentikasi, bisa dicari dengan memasukkan *block* yang akan diautentikasi ke *method* `blockToSector(int blockIndex)`, dan `key` adalah kunci A atau B yang digunakan untuk autentikasi. *Methods* ini mengembalikan nilai *boolean*. Jika mendapat nilai *true*, dapat dilanjutkan menulis data ke atau membaca data dari *tag* NFC.

II.3.3.1 Membaca Data *Reader/Writer Block* dari *Tag* NFC

Setelah berhasil autentikasi *block*, membaca data *reader/writer block* dapat dilakukan menggunakan *method* `readBlock(int blockIndex)` dengan `blockIndex` adalah *reader/writer block* yang ingin dibaca, dimulai dari 0. *Method* `readBlock` mengembalikan nilai dari *block* tersebut dalam *array of bytes* yang berukuran 16 *bytes*. *Method* ini akan berhasil jika *block* tersebut berhasil diautentikasi.

II.3.3.2 Membaca Data *Value Block* dari *Tag* NFC

Sama seperti membaca *reader/writer block*, setelah berhasil autentikasi *block*, membaca data *value block* dapat dilakukan menggunakan *method* `readBlock(int blockIndex)` dengan `blockIndex` adalah *value block* yang ingin dibaca, dimulai dari 0. *Method* `readBlock` mengembalikan nilai dari *block* tersebut dalam *array of bytes* yang berukuran 16 *bytes* yang berbentuk bilangan *hexadecimal*. *Method* ini akan berhasil jika *block* tersebut berhasil diautentikasi.

II.3.3.3 Menulis Data ke *Reader/Writer Block Tag* NFC

Sebelum menulis data ke *reader/writer block*, *reader/writer block* yang ingin ditulis harus berhasil diautentikasi terlebih dahulu. Setelah berhasil autentikasi *reader/writer block*, gunakan *method* `writeBlock(int blockIndex, byte[] data)`. `blockIndex` adalah *reader/writer block* yang akan ditulis dan `data` adalah bilangan *hexadecimal* berukuran 16 *bytes* yang akan ditulis ke *reader/writer block*.

II.3.3.4 Menulis Data ke *Value Block Tag* NFC

Menulis data ke *value block* dapat berupa *increment* atau *decrement*. *Method* `increment(int blockIndex, int value)` digunakan untuk *increment* sedangkan *method* `decrement(int blockIndex, int value)` digunakan untuk *decrement*. `blockIndex` adalah *value block* yang akan ditulis dan `value` adalah bilangan *integer* yang digunakan untuk *increment* atau *decrement value block*. Sama seperti menulis data ke *reader/writer block*, *value block* yang ingin di-*increment* atau di-*decrement* harus berhasil diautentikasi terlebih dahulu.

II.4 Kriptografi

Kriptografi adalah ilmu untuk keamanan dengan menulis secara rahasia (Knudsen, 1998). Kriptografi memastikan sesuatu dilakukan oleh orang yang tetap, terutama ketika pengguna komputer memasukkan nama pengguna beserta kata sandi ke suatu sistem. Kriptografi mencakup enkripsi dengan berbagai algoritma.

II.4.1 Enkripsi Simetrik dan Asimetrik

Enkripsi adalah proses mengubah data yang disebut *plaintext* secara matematis menjadi data tidak terbaca yang disebut *chipertext* (Knudsen, 1998). Proses mengubah *chipertext* menjadi data yang terbaca (*plaintext*) disebut dekripsi. Enkripsi ada dua jenis, yaitu enkripsi simetrik dan enkripsi asimetrik.

Enkripsi simetrik adalah enkripsi di mana proses enkripsi dan dekripsi menggunakan satu kunci yang sama, yaitu *private key* atau bisa juga disebut

secret key. Contoh algoritma simetrik adalah DES, AES (Rijndael), *twofish*, dan lain-lain. Tugas akhir ini menggunakan algoritma AES untuk enkripsi simetrik.

Enkripsi asimetrik adalah enkripsi di mana proses enkripsi dan dekripsi tidak menggunakan kunci yang sama. Untuk enkripsi, pihak pengirim pesan menggunakan *public key* untuk mengenkripsi pesan. Setelah pesan dikirim, pihak penerima pesan akan mendekripsi menggunakan *private key* miliknya sehingga penerima pesan dapat membaca isi pesan. Contoh algoritma asimetrik adalah RSA, ECC, dan lain-lain. Pada tugas akhir ini, enkripsi asimetrik tidak digunakan. Namun algoritma dari enkripsi asimetrik akan digunakan, yaitu algoritma ECDSA.

Masing-masing jenis enkripsi memiliki kelemahan dan kelebihan. Enkripsi simetrik jika *private key* diketahui oleh penyerang, penyerang tersebut langsung mengetahui bagaimana cara mengenkripsi dan mendekripsi pesan. Akan tetapi, enkripsi simetrik ini dapat dilakukan dengan cepat dan efisien, tidak seperti enkripsi asimetrik yang membutuhkan waktu lebih lama.

II.4.2 Digital Signature

Digital signature adalah sebuah cara autentikasi yang membuat penulis pesan menyisipkan sebuah kode sebagai tanda tangan atau *signature* (Stallings, 2005). *Signature* pada *digital signature* ini dibentuk dengan melakukan operasi *hash* pada pesan yang akan diberi *digital signature* atau mengenkripsi pesan tersebut dengan kunci privat dari pengirim pesan.

Digital signature ini adalah bukti otentik yang tidak dapat digunakan ulang dan tidak dapat diubah. Sehingga, *digital signature* tidak dapat disangkal dan dapat membantu aspek keamanan *non-repudiation*. *Digital signature* membantu penerima pesan mengetahui bahwa pesan yang dia terima berasal dari pihak yang benar. Ada banyak algoritma untuk *digital signature*, namun algoritma *digital signature* yang akan digunakan pada tugas akhir ini adalah algoritma ECDSA.

II.4.3 Algoritma AES (*Advanced Encryption Standard*)

Algoritma AES adalah algoritma DES yang dimodifikasi menjadi lebih baik oleh Rijndael. Panjang kunci pada algoritma AES bisa 128 bit sampai 256 bit dengan step 32 bit. Algoritma akan tahan terhadap serangan *exhaustive key search* karena membutuhkan waktu bertahun-tahun untuk memecahkan *key* yang digunakan pada algoritma AES (Munir, 2004).

II.4.4 Algoritma ECC (*Elliptic Curve Cryptography*) dan ECDSA (*Elliptic Curve Digital Signature Algorithm*)

Algoritma ECC termasuk dalam kriptografi kunci publik atau asimetrik. Prinsip dari algoritma ECC adalah memiliki kekuatan enkripsi dan dekripsi yang aman dengan kunci yang berukuran lebih kecil dengan proses seefektif mungkin (Stallings, 2005). Sehingga, tidak diperlukan daya dan tempat penyimpanan yang banyak. Algoritma ECC mendasarkan keamanan kriptografi pada analogi kurva elips.

Algoritma ECDSA adalah algoritma ECC yang digunakan untuk memberikan *digital signature* pada pesan yang akan dikirim. Untuk memberikan *digital signature* pada suatu pesan, pesan tersebut akan dibubuhi *digital signature* yang menggunakan kunci privat dari pengirim pesan. Penerima pesan menggunakan kunci public dari pengirim pesan untuk memverifikasi apakah pesan tersebut berasal dari pihak yang benar.

II.4.5 Algoritma PBKDF2

Algoritma PBKDF2 adalah algoritma yang digunakan untuk menurunkan kunci dan terlindung dari serangan *brute-force* (Society, 2000). PBKDF2 memiliki masukan berupa kunci master yang akan diturunkan, salt yang dapat berupa bilangan random, jumlah perulangan *hash* yang dilakukan, dan panjang kunci turunan yang ingin dihasilkan, dapat sebesar 128, 192, atau 256 bit. Masukan-masukan tersebut akan menghasilkan kunci turunan.

BAB III

ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

MICROPAYMENT DENGAN TEKNOLOGI NFC

III.1 Analisis Kebutuhan dan Keamanan SISMIC

Subbab ini meliputi pembahasan tentang analisis kebutuhan SISMIC dan analisis kebutuhan keamanan SISMIC.

III.1.1 Analisis Kebutuhan SISMIC

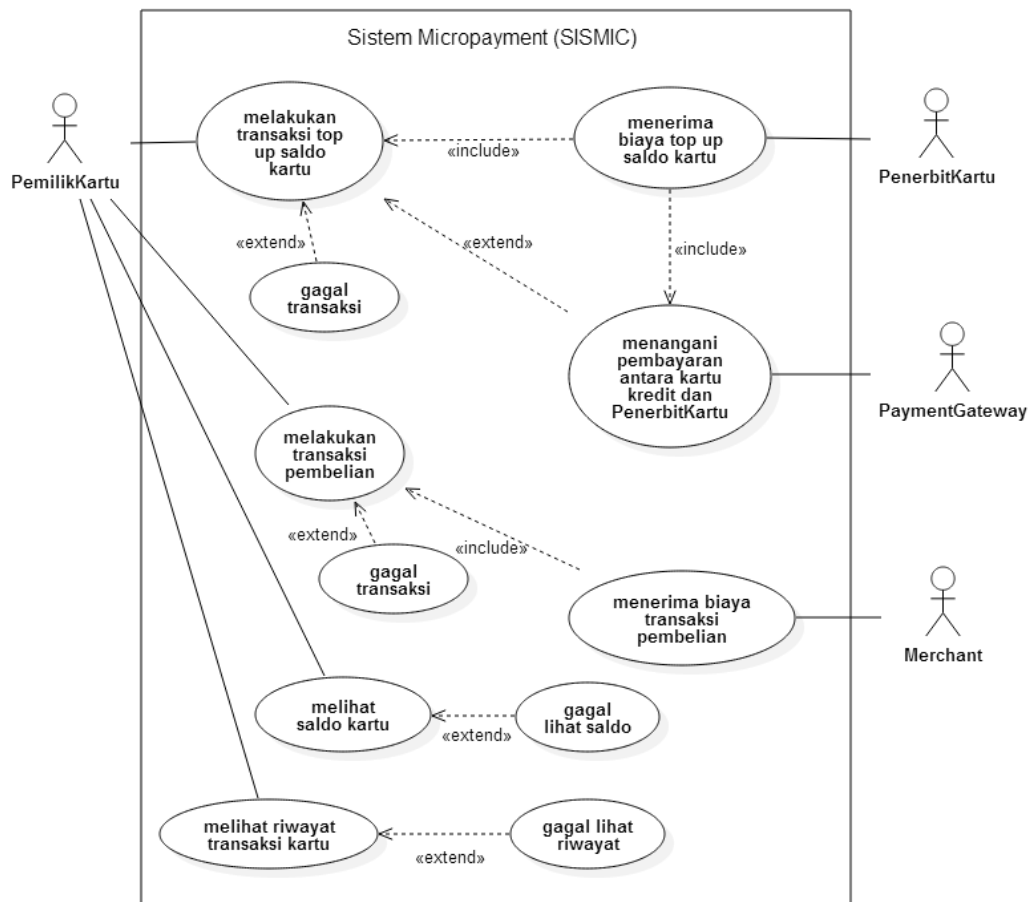
Tugas akhir ini membuat perancangan untuk sebuah sistem transaksi *micropayment* yang disebut SISMIC. SISMIC harus memenuhi kebutuhan-kebutuhan sebagai berikut:

1. SISMIC dapat melakukan transaksi *top-up* saldo kartu.
 - 1.1. SISMIC dapat menambah saldo kartu.
 - 1.2. SISMIC dapat memberikan biaya pembayaran *top-up* saldo kartu ke penerbit kartu.
 - 1.3. SISMIC dapat menyimpan riwayat transaksi *top-up* saldo kartu ke dalam kartu.
 - 1.4. SISMIC dapat menyimpan riwayat transaksi *top-up* saldo kartu ke dalam basisdata.
2. SISMIC dapat melakukan transaksi pembelian menggunakan kartu.
 - 2.1. SISMIC dapat mengurangi saldo kartu.
 - 2.2. SISMIC dapat memberikan biaya transaksi pembelian pemilik kartu ke *merchant*.
 - 2.3. SISMIC dapat menyimpan riwayat transaksi pembelian ke dalam kartu.
 - 2.4. SISMIC dapat menyimpan riwayat transaksi pembelian saldo kartu ke dalam basisdata.
3. SISMIC dapat menunjukkan saldo kartu pada pemilik kartu.

4. SISMIC dapat menunjukkan riwayat transaksi kartu pada pemilik kartu.
5. Kartu SISMIC memiliki masa berlaku (tanggal kadaluarsa).

Benda-benda yang ada pada SISMIC adalah sebagai berikut:

1. Kartu SISMIC yang merupakan sebuah *tag* NFC.
2. *Smartphone* yang mendukung fitur NFC dan berplatform Android.
3. Mesin EDC, yang dimiliki oleh *merchant*.
4. Mesin ATM, yang dimiliki oleh penerbit kartu.



Gambar III-1 Diagram *Use Case* SISMIC

Gambaran umum apa saja yang dapat dilakukan SISMIC dapat dilihat pada Gambar III-1 di atas yang berupa diagram *use case*. Pada Gambar III-1, ada beberapa aktor pada SISMIC. Aktor-aktor yang ada pada SISMIC dapat dilihat sebagai berikut:

1. Penerbit kartu. Penerbit kartu bisa berasal dari bank, *provider* kartu telepon selular, dan lain-lain. Pada umumnya di Indonesia, penerbit kartu berasal dari bank. Pada tugas akhir ini, penerbit kartu adalah sebuah bank.
2. Pemilik kartu, yaitu orang yang membeli kartu dari penerbit kartu dan dapat menggunakan kartu tersebut untuk melakukan berbagai transaksi dengan memanfaatkan teknologi NFC.
3. *Merchant*, yaitu tempat di mana pemilik kartu dapat melakukan transaksi pembelian. *Merchant* dapat berupa supermarket, restoran, dan lain-lain. *Merchant* ini telah bekerja sama dengan penerbit kartu.
4. *Payment gateway*, yaitu perantara transaksi yang terjadi antara penerbit kartu dan bank kartu kredit yang dimiliki pemilik kartu. *Payment gateway* adalah layanan transaksi perdagangan menggunakan teknologi informasi (Gulati & Srivastava, 2007). *Payment gateway* melayani proses pembayaran menggunakan kartu kredit atau pembayaran langsung. Layanan *payment gateway* dapat disediakan oleh bank kepada pelanggannya atau penyedia layanan keuangan lainnya sebagai layanan terpisah oleh penyedia layanan keuangan elektronik.

Seperti yang dapat dilihat pada Gambar III-1, pemilik kartu dapat melakukan transaksi *top-up* saldo kartu, transaksi pembelian, melihat saldo kartu SISMIC, dan melihat riwayat transaksi kartu SISMIC. Penerbit kartu dapat menerima uang *top-up* saldo kartu SISMIC dari pemilik kartu. *Merchant* dapat menerima biaya uang transaksi pembelian oleh pemilik kartu SISMIC. *Payment gateway* menangani pembayaran antara kartu kredit dan penerbit kartu jika pemilik kartu melakukan *top-up* kartu SISMIC menggunakan kartu kredit melalui *smartphone*.

III.1.2 Analisis Kebutuhan Keamanan SISMIC

Aspek keamanan yang harus dicapai oleh SISMIC adalah berikut:

- *Confidentiality*
 - Data yang ada pada SISMIC tidak dapat diakses oleh orang yang tidak berhak.

- *Integrity*
 - Data SISMIC tidak boleh berubah tanpa izin dari pihak yang berhak.
- *Availability*
 - Ketika orang, baik penerbit kartu, pemilik kartu, ataupun *merchant*, menggunakan SISMIC, layanan SISMIC harus dapat digunakan.
- *Authentication*
 - Transaksi SISMIC hanya dapat dilakukan pada *hardware* dan *software* yang resmi dikeluarkan oleh penerbit kartu.
- *Authorization*
 - Penerbit kartu, pemilik kartu, dan *merchant* melakukan transaksi secara legal sesuai tugasnya masing-masing pada SISMIC.
- *Accountability*
 - Segala aktivitas yang terjadi di SISMIC ada catatannya.
- *Non-repudiation*
 - Tidak ada pihak dari penerbit kartu, pemilik kartu, maupun *merchant* yang dapat menyanggah suatu transaksi yang telah terjadi.

Pada subbab berikutnya, akan dijelaskan operasi apa saja yang dapat dilakukan SISMIC.

III.2 Perancangan SISMIC

Subbab ini meliputi pembahasan tentang apa saja operasi yang ada pada aplikasi SISMIC, bagaimana penyimpanan dan struktur data pada SISMIC, dan bagaimana rancangan keamanan SISMIC.

III.2.1 Operasi Aplikasi SISMIC

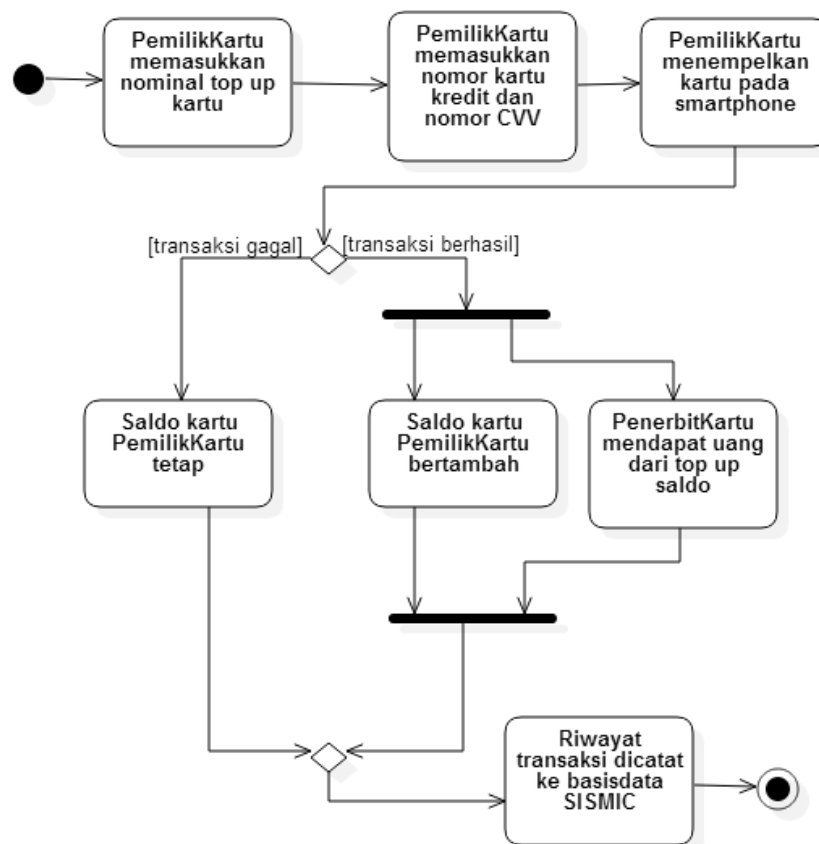
Operasi Aplikasi SISMIC meliputi pembahasan tentang bagaimana terjadinya transaksi *top-up* kartu SISMIC, terjadinya transaksi pembelian, proses melihat saldo, proses melihat riwayat transaksi, dan proses melihat masa berlaku kartu.

III.2.1.1 Transaksi *Top-Up* Kartu SISMIC

Transaksi *Top-Up* Kartu SISMIC meliputi pembahasan tentang transaksi *top-up* kartu SISMIC melalui *smartphone*, ATM, dan *merchant*.

III.2.1.1.1 Melalui *Smartphone*

Transaksi *top-up* kartu SISMIC dapat dilakukan secara *online* melalui *smartphone* yaitu melalui aplikasi SISMIC. Diagram aktivitas untuk transaksi *top-up* kartu SISMIC melalui *smartphone* dapat dilihat pada gambar III.2 berikut ini.



Gambar III-2 Diagram Aktivitas Transaksi *Top-Up* Saldo Melalui *Smartphone*

Pada Gambar III-2, dapat dilihat bagaimana alur kerja transaksi *top-up* saldo pada SISMIC melalui *smartphone*. Pertama, pemilik kartu memasukkan nominal *top-up* saldo kartu SISMIC pada *smartphone* melalui aplikasi SISMIC. Setelah itu, pemilik kartu memasukkan nomor kartu kredit dengan nomor CVV dan pemilik kartu menempelkan kartu pada *smartphone*. Jika transaksi berhasil, saldo kartu

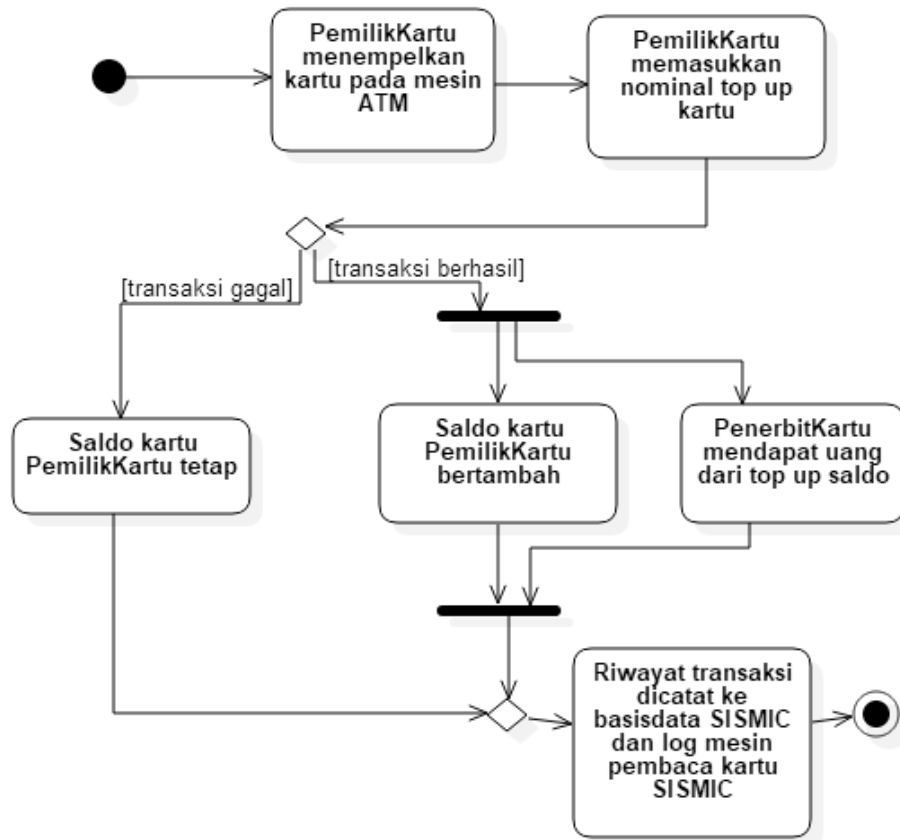
SISMIC akan bertambah dan penerbit kartu akan menerima uang dari biaya *top-up* saldo pemilik kartu. Jika transaksi gagal, saldo kartu SISMIC akan tetap seperti sebelumnya. Terakhir, riwayat transaksi *top-up*, baik yang berhasil ataupun yang gagal, akan dicatat ke basisdata SISMIC. Hal yang menyebabkan transaksi *top-up* gagal adalah nominal *top-up* yang membuat kartu SISMIC memiliki jumlah saldo melebihi batas maksimal saldo dan masa berlaku kartu SISMIC sudah habis.

Untuk melakukan transaksi pada *smartphone*, pemilik kartu harus memiliki kartu kredit. Pembayaran dilakukan dengan pemilik kartu memasukkan nomor kartu kredit dan nomor CVV. Di balik proses yang ada di gambar III.5, penerbit kartu akan bekerja sama dengan *payment gateway* untuk melakukan transaksi dengan kartu kredit. Berikut ini adalah proses bagaimana penerbit kartu dapat menerima pembayaran transaksi *top-up* dari pemilik kartu yang melakukan transaksi *top-up* melalui *smartphone* dengan kartu kredit:

1. Pemilik kartu melalui aplikasi SISMIC mengirimkan identitas kartu kredit kepada *payment gateway*.
2. *Payment gateway* mengirimkan identitas kartu kredit kepada bank kartu kredit untuk dikonfirmasi apakah identitas kartu kredit benar atau salah dan apakah masih dapat digunakan atau tidak.
3. Bank kartu kredit memberikan konfirmasi keabsahan identitas kartu kredit kepada *payment gateway*.
4. *Payment gateway* memberikan konfirmasi kepada penerbit kartu.
5. Jika kartu kredit sah untuk digunakan pembayaran, penerbit kartu melalui aplikasi SISMIC memberikan konfirmasi kepada pemilik kartu bahwa kartu kredit sah untuk digunakan.
6. Saldo kartu pemilik kartu bertambah.
7. *Payment gateway* menangani pembayaran transaksi *top-up* antara penerbit kartu dan bank kartu kredit.

III.2.1.1.2 Melalui ATM

Selain melalui *smartphone*, transaksi *top-up* kartu SISMIC dapat dilakukan secara *offline* melalui mesin ATM yang dikeluarkan oleh penerbit kartu. Mesin ATM yang dapat digunakan untuk transaksi *top-up* kartu SISMIC harus memiliki slot untuk membaca kartu SISMIC. Mesin ATM pada tugas akhir ini akan disimulasikan dengan NFC reader dan *desktop*.



Gambar III-3 Diagram Aktivitas Transaksi *Top-Up* Melalui Mesin ATM

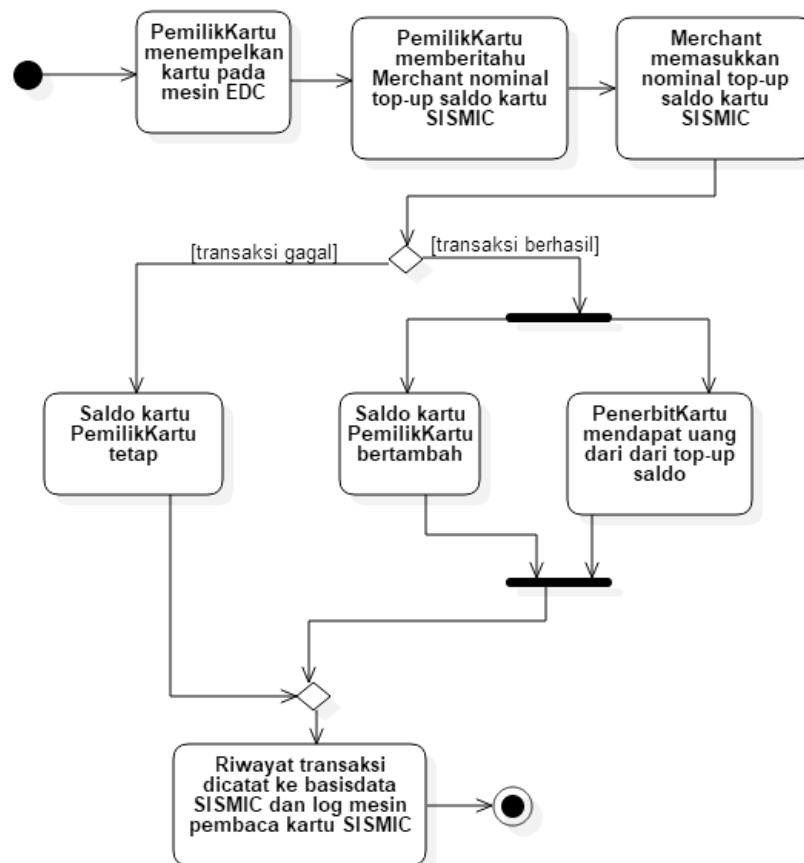
Pada Gambar III-3 di atas, dapat dilihat bagaimana alur kerja transaksi *top-up* saldo pada SISMIC melalui mesin ATM. Sedangkan untuk diagram komunikasi transaksi *top-up* saldo pada SISMIC melalui mesin ATM dapat dilihat di lampiran B. Pemilik kartu mendatangi langsung mesin ATM. Pertama, pemilik kartu menempelkan kartu SISMIC pada mesin ATM. Setelah itu, pemilik kartu memasukkan nominal *top-up* saldo kartu SISMIC. Jika transaksi berhasil, saldo kartu SISMIC akan bertambah dan penerbit kartu akan menerima uang dari biaya

top-up saldo pemilik kartu. Jika transaksi gagal, saldo kartu SISMIC akan tetap seperti sebelumnya. Terakhir, riwayat transaksi *top-up*, baik yang berhasil ataupun yang gagal, akan dicatat ke basisdata SISMIC dan dicatat ke log di dalam mesin pembaca kartu SISMIC pada mesin ATM. Hal yang menyebabkan transaksi *top-up* gagal adalah nominal *top-up* yang membuat kartu SISMIC memiliki jumlah saldo melebihi batas maksimal saldo dan masa berlaku kartu SISMIC sudah habis.

Untuk melakukan transaksi *top-up* pada mesin ATM, pemilik kartu harus memiliki kartu debit diterbitkan oleh penerbit kartu dan melakukan pengisian pada mesin ATM yang dikeluarkan oleh penerbit kartu. Kartu debit dan mesin ATM yang dikeluarkan oleh bank lain selain penerbit kartu tidak dapat digunakan. Setelah transaksi *top-up* melalui mesin ATM dilakukan, saldo kartu debit milik pemilik kartu akan berkurang sesuai biaya transaksi pembelian dan uang dari pengurangan saldo tersebut akan diterima oleh penerbit kartu.

III.2.1.1.3 Melalui *Merchant*

Selain melalui *smartphone* dan ATM, transaksi *top-up* kartu SISMIC dapat dilakukan secara *offline* dengan mendatangi *merchant* yang bekerja sama dengan penerbit kartu. Masing-masing *merchant* memiliki mesin EDC yang digunakan untuk membaca kartu SISMIC. Mesin EDC pada tugas akhir ini akan disimulasikan dengan NFC *reader* dan *desktop*.



Gambar III-4 Diagram Aktivitas Transaksi *Top-Up* Melalui *Merchant*

Pada Gambar III-4 di atas, dapat dilihat bagaimana alur kerja transaksi *top-up* saldo pada SISMIC melalui *merchant*. Sedangkan untuk diagram komunikasi transaksi *top-up* saldo pada SISMIC melalui *merchant* dapat dilihat di lampiran B. Pemilik kartu mendatangi langsung *merchant* yang bekerja sama dengan penerbit kartu. Pertama, pemilik kartu menempelkan kartu SISMIC pada mesin EDC. Lalu, pemilik kartu memberitahu *merchant* nominal *top-up* saldo kartu SISMIC. Setelah itu, *merchant* akan memasukkan nominal *top-up* saldo pada mesin EDC. Jika transaksi berhasil, saldo kartu SISMIC akan bertambah dan penerbit kartu akan menerima uang dari biaya *top-up* saldo pemilik kartu. Jika transaksi gagal, saldo kartu SISMIC akan tetap seperti sebelumnya. Terakhir, riwayat transaksi *top-up*, baik yang berhasil ataupun yang gagal, akan dicatat ke basisdata SISMIC dan dicatat ke log di dalam mesin EDC. Hal yang menyebabkan transaksi *top-up* gagal

adalah nominal *top-up* yang membuat kartu SISMIC memiliki jumlah saldo melebihi batas maksimal saldo dan masa berlaku kartu SISMIC sudah habis.

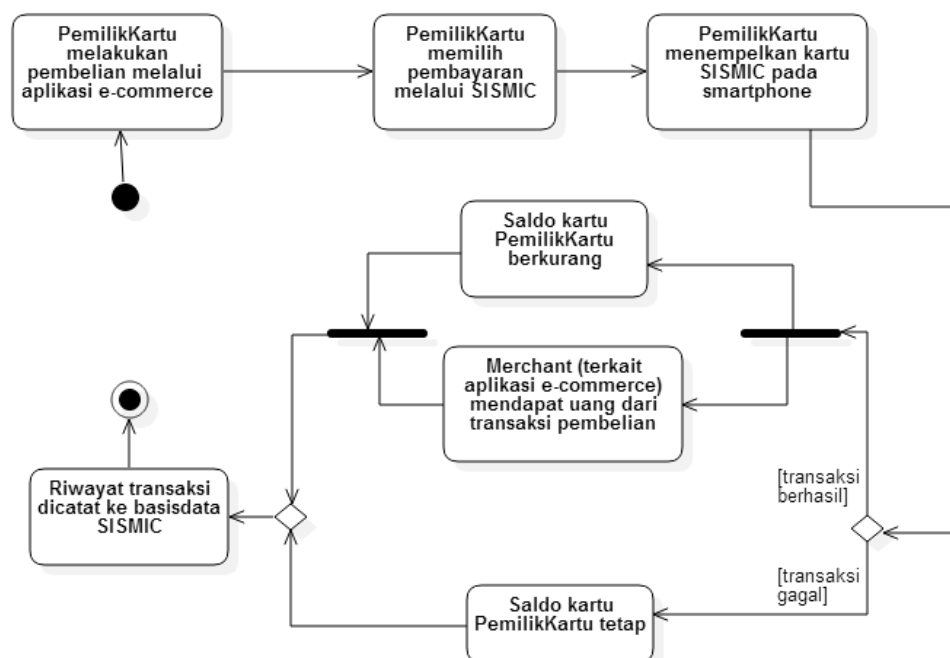
Untuk melakukan transaksi *top-up* pada *merchant*, pemilik kartu harus mendatangi *merchant* yang telah bekerja sama dengan penerbit kartu. Semua *merchant* yang bekerja sama dengan penerbit kartu memiliki mesin EDC. Mesin EDC dapat digunakan untuk melakukan transaksi *top-up* atau transaksi pembelian.

III.2.1.2 Transaksi Pembelian

Sub subbab ini meliputi pembahasan tentang transaksi pembelian melalui *smartphone* tanpa parameter dan *merchant* baik tanpa parameter ataupun dengan parameter.

III.2.1.2.1 Melalui *Smartphone* Tanpa Parameter

Transaksi pembelian dalam SISMIC dapat dilakukan melalui *smartphone* secara *online* melalui *smartphone* dengan aplikasi *e-commerce* yang telah bekerja sama dengan penerbit kartu. Diagram aktivitas untuk transaksi pembelian melalui *smartphone* dapat dilihat pada Gambar III-5 berikut ini.



Gambar III-5 Diagram Aktivitas Transaksi Pembelian Melalui *Smartphone*

Pada Gambar III-5 di atas, dapat dilihat bagaimana alur kerja transaksi pembelian melalui *smartphone*. Pertama, pemilik kartu melakukan pembelian melalui aplikasi *e-commerce* pada *smartphone* yang telah bekerja sama dengan penerbit kartu. Lalu, pemilik kartu memilih menu pembayaran melalui SISMIC. Dari tahap ini, aplikasi akan ditangani oleh SISMIC untuk melakukan pembayaran. Jika transaksi berhasil, saldo kartu SISMIC akan berkurang dan *merchant* (pihak *e-commerce*) di mana pemilik kartu membeli barang akan menerima uang dari biaya transaksi pembelian melalui penerbit kartu. Jika transaksi gagal, saldo kartu SISMIC akan tetap seperti sebelumnya. Terakhir, riwayat transaksi pembelian, baik yang berhasil ataupun yang gagal, akan dicatat ke basisdata SISMIC. Setelah itu, aplikasi akan kembali ke aplikasi *e-commerce*. Hal yang menyebabkan transaksi pembelian gagal adalah saldo kartu yang tidak mencukupi nominal transaksi pembelian dan masa berlaku kartu SISMIC sudah habis.

III.2.1.2.2 Melalui Merchant Tanpa Parameter

Selain melalui *smartphone*, transaksi pembelian dapat dilakukan dengan mengunjungi *merchant* yang telah bekerja sama dengan penerbit kartu secara *offline*. Transaksi pembelian di *merchant* menggunakan kartu SISMIC yang dibaca oleh mesin EDC. Diagram aktivitas untuk transaksi pembelian melalui *merchant* tanpa parameter dapat dilihat pada gambar Gambar III-6 berikut ini. Transaksi pembelian melalui mesin EDC ada dua jenis, ada pembelian tanpa parameter, ada pembelian dengan parameter. Yang dimaksud dengan pembelian tanpa parameter adalah pembelian barang yang cukup hanya dengan satu kali menempelkan kartu SISMIC.

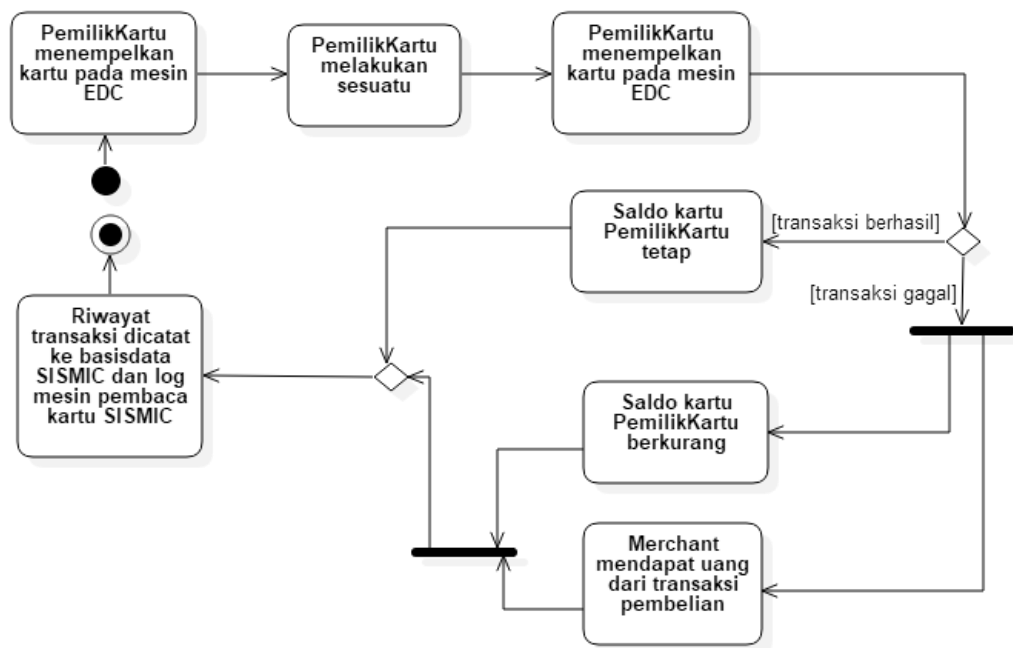


33

Merchant akan menerima uang hasil transaksi pembelian dari penerbit kartu. Penerbit kartu akan memberikan uang tersebut ke *merchant* dengan mengirimkan uang tersebut melalui kartu debit yang dimiliki *merchant*. *Merchant* yang bekerja sama dengan penerbit kartu akan memiliki kartu debit yang diterbitkan oleh penerbit kartu untuk menerima uang dari berbagai macam transaksi pembelian dengan SISMIC.

III.2.1.2.3 Melalui *Merchant* dengan Parameter

Pada dasarnya, transaksi pembelian melalui *merchant* dengan parameter sama saja dengan transaksi pembelian melalui *merchant* tanpa parameter. Hanya saja, transaksi pembelian melalui *merchant* dengan parameter pembayarannya berdasarkan jarak ataupun waktu. Diagram aktivitas transaksi pembelian dengan parameter melalui *merchant* dapat dilihat pada Gambar III-7 berikut ini.



Gambar III-7 Diagram Aktivitas Transaksi Pembelian dengan Parameter Melalui *Merchant*

Pada gambar Gambar III-7 di atas, dapat dilihat bagaimana alur kerja transaksi pembelian di *merchant* melalui mesin EDC tanpa parameter. Pemilik kartu akan mendatangi *merchant* di mana pembayarannya dihitung berdasarkan waktu atau

jarak seperti tempat bermain anak yang tarifnya dihitung per-jam, tempat bermain *ice skating* yang tarifnya dihitung per-jam, tempat olahraga yang tarifnya dihitung per-jam, transportasi seperti bus ataupun kereta. Pertama, pemilik kartu menempelkan kartu SISMIC pada mesin EDC. Setelah itu, pemilik kartu melakukan sesuatu sesuai *merchant* yang ia datangi, misal jika pemilik kartu ke tempat *ice skating*, pemilik kartu akan bermain *ice skating*. Setelah selesai, pemilik kartu akan menempelkan kartu SISMIC pada mesin EDC lagi. Apa yang terjadi selanjutnya sama dengan transaksi pembelian dengan parameter melalui *merchant*.

III.2.1.3 Lihat Saldo

Sub subbab ini meliputi pembahasan tentang bagaimana proses melihat saldo melalui *smartphone* dan melalui ATM.

III.2.1.3.1 Melalui *Smartphone*

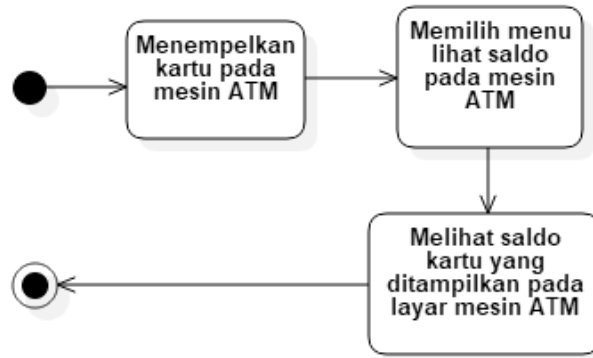


Gambar III-8 Diagram Aktivitas Lihat Saldo SISMIC Melalui *Smartphone*

Pada Gambar III-8 di atas, dapat dilihat bagaimana alur kerja pemilik kartu melihat saldo kartu SISMIC melalui aplikasi SISMIC pada *smartphone*. Pertama, pemilik kartu menempelkan kartu pada *smartphone*. Setelah itu, saldo kartu

SISMIC akan ditampilkan di layar *smartphone*. Diagram komunikasi untuk proses melihat saldo kartu SISMIC dapat dilihat pada lampiran B.

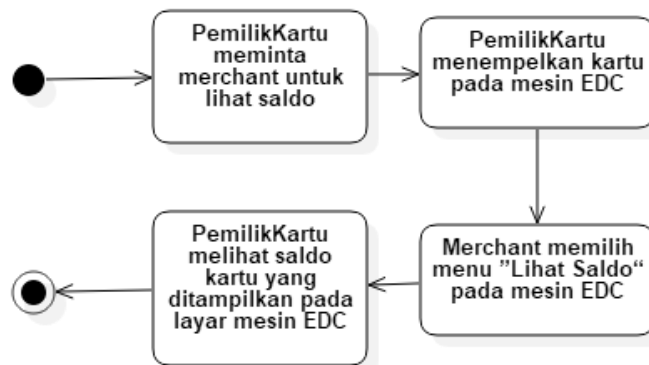
III.2.1.3.2 Melalui ATM



Gambar III-9 Diagram Aktivitas Lihat Saldo SISMIC Melalui ATM

Selain melalui *smartphone*, saldo kartu dapat dilihat secara *offline* melalui ATM. Sedangkan untuk diagram komunikasi lihat saldo SISMIC melalui mesin ATM dapat dilihat di lampiran B. Pada Gambar III-9 di atas, dapat dilihat bagaimana alur kerja pemilik kartu melihat saldo kartu SISMIC melalui mesin ATM. Pertama, pemilik kartu menempelkan kartu pada mesin ATM. Setelah itu, pemilik kartu memilih menu untuk melihat saldo kartu SISMIC pada mesin ATM. Terakhir, saldo kartu SISMIC akan ditampilkan di layar mesin ATM.

III.2.1.3.3 Melalui Merchant



Gambar III-10 Diagram Aktivitas Lihat Saldo SISMIC Melalui Merchant

Selain melalui *smartphone* dan ATM, saldo kartu dapat dilihat secara *offline* melalui *merchant* menggunakan mesin EDC. Sedangkan untuk diagram komunikasi lihat saldo SISMIC melalui *merchant* dapat dilihat di lampiran B. Pada Gambar III-10 di atas, dapat dilihat bagaimana alur kerja pemilik kartu melihat saldo kartu SISMIC melalui *merchant*. Pertama, pemilik kartu meminta *merchant* untuk melihat saldo kartu SISMIC milik pemilik kartu. Lalu, pemilik kartu akan menempelkan kartu pada mesin EDC. Setelah itu, *merchant* akan memilih menu "Lihat Saldo dan Masa Berlaku" pada mesin EDC. Terakhir, saldo dan masa berlaku kartu SISMIC akan ditampilkan pada layar mesin EDC.

III.2.1.4 Lihat Riwayat Transaksi



Gambar III-11 Diagram Aktivitas Lihat Riwayat Transaksi

Pada Gambar III-11 di atas, dapat dilihat bagaimana alur kerja pemilik kartu melihat riwayat transaksi SISMIC. Melihat riwayat transaksi SISMIC dilakukan melalui aplikasi SISMIC pada *smartphone*. Pertama, pemilik kartu menempelkan kartu pada *smartphone*. Setelah itu, riwayat transaksi SISMIC akan ditampilkan di layar *smartphone*, baik transaksi yang berhasil ataupun yang gagal.

III.2.1.5 Lihat Masa Berlaku Kartu SISMIC

Sub subbab ini meliputi pembahasan tentang bagaimana proses melihat masa berlaku kartu SISMIC melalui *smartphone* dan ATM.

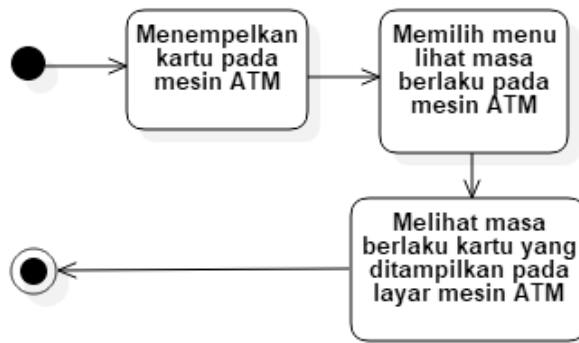
III.2.1.5.1 Melalui *Smartphone*



Gambar III-12 Diagram Aktivitas Lihat Masa Berlaku Kartu SISMIC Melalui *Smartphone*

Pada Gambar III-12 di atas, dapat dilihat bagaimana alur kerja pemilik kartu melihat masa berlaku kartu SISMIC melalui aplikasi SISMIC pada *smartphone*. Pertama, pemilik kartu menempelkan kartu pada *smartphone*. Setelah itu, masa berlaku kartu SISMIC akan ditampilkan di layar *smartphone*.

III.2.1.5.2 Melalui ATM



Gambar III-13 Diagram Aktivitas Lihat Masa Berlaku SISMIC Melalui ATM

Selain melalui *smartphone*, masa berlaku kartu dapat dilihat secara *offline* melalui ATM. Pada Gambar III-13 di atas, dapat dilihat bagaimana alur kerja pemilik kartu melihat masa berlaku kartu SISMIC melalui mesin ATM. Sedangkan untuk diagram komunikasi lihat masa berlaku SISMIC melalui mesin ATM dapat dilihat di lampiran B. Pertama, pemilik kartu menempelkan kartu pada mesin ATM.

Setelah itu, pemilik kartu memilih menu untuk melihat masa berlaku kartu SISMIC pada mesin ATM. Terakhir, masa berlaku kartu SISMIC akan ditampilkan di layar mesin ATM.

III.2.1.5.3 Melalui *Merchant*



Gambar III-14 Diagram Aktivitas Lihat Masa Berlaku Kartu SISMIC Melalui *Merchant*

Selain melalui *smartphone* dan ATM, masa berlaku kartu dapat dilihat secara *offline* melalui *merchant* menggunakan mesin EDC. Pada Gambar III-14 di atas, dapat dilihat bagaimana alur kerja pemilik kartu melihat masa berlaku kartu SISMIC melalui *merchant*. Sedangkan untuk diagram komunikasi lihat masa berlaku SISMIC melalui *merchant* dapat dilihat di lampiran B. Pertama, pemilik kartu meminta *merchant* untuk melihat masa berlaku kartu SISMIC milik pemilik kartu. Lalu, pemilik kartu akan menempelkan kartu pada mesin EDC. Setelah itu, *merchant* akan memilih menu "Lihat Saldo dan Masa Berlaku" pada mesin EDC. Terakhir, saldo dan masa berlaku kartu SISMIC akan ditampilkan pada layar mesin EDC.

III.2.2 Penyimpanan dan Struktur Data SISMIC

Subbab ini meliputi pembahasan tentang bagaimana struktur penyimpanan dan hak akses kartu SISMIC, basisdata, dan diagram kelas pada SISMIC.

III.2.2.1 Struktur Penyimpanan dan Hak Akses Kartu SISMIC

Kartu SISMIC memiliki teknologi NFC dan memori berukuran 1 KB dengan 16 sektor, di mana 1 sektornya terdiri dari 4 blok dengan ukuran 16 *bytes* per blok. Saat pertama kali kartu diterbitkan, beberapa sektor kartu SISMIC akan dikonfigurasi seperti pada Tabel III-1 berikut ini.

Tabel III-1 Konfigurasi Sektor dan Blok Kartu SISMIC

| Sektor | Blok | Jenis Blok | Data |
|--------|------|----------------------------|---|
| 0 | 0 | - | Nomor kartu dari manufaktur |
| | 1 | <i>Reader/writer block</i> | Tanggal masa berlaku kartu |
| | 2 | <i>Value block</i> | Saldo Kartu |
| 1-5 | 0 | <i>Reader/writer block</i> | Riwayat waktu transaksi pembelian atau <i>top-up</i> dalam bentuk epoch |
| | 1 | <i>Value block</i> | Riwayat nominal transaksi pembelian atau <i>top-up</i> |
| | 2 | <i>Reader/writer block</i> | Riwayat jenis transaksi apakah transaksi merupakan transaksi pembelian atau <i>top-up</i> |
| 6-14 | 0 | <i>Reader/writer block</i> | Parameter jarak atau waktu pertama kartu disentuh pada <i>reader</i> |
| | 1 | <i>Reader/writer block</i> | Iv hasil dari enkripsi blok ke-0 di sektor 6 sampai 14 |
| | 2 | - | - |

| Sektor | Blok | Jenis Blok | Data |
|--------|------|----------------------------|--|
| 15 | 0 | - | - |
| | 1 | <i>Reader/writer block</i> | Iv hasil dari enkripsi blok ke-1 di sektor 0 |
| | 2 | <i>Reader/writer block</i> | Iv hasil dari enkripsi blok ke-2 di sektor 0 |

Penjelasan lebih lengkap mengenai Tabel III-1 pada uraian berikut ini:

1. Sektor ke-0 blok ke-1 yang berupa *reader/writer block* menyimpan tanggal masa berlaku kartu. Blok ini pada awalnya ditentukan dan ditulis oleh penerbit kartu. Lalu, blok ini dapat dibaca oleh pemilik kartu dan tidak dapat ditulis ulang oleh siapapun kecuali ada wewenang dari penerbit kartu. Data yang tersimpan di blok ini dienkripsi menggunakan algoritma AES.
2. Sektor ke-0 blok ke-2 yang berupa *value block* menyimpan saldo kartu. Blok ini dapat ditulis ketika kartu digunakan untuk transaksi pembelian atau *top-up* oleh pemilik kartu pada *smartphone*, mesin EDC, atau mesin ATM yang telah dipastikan keamanannya. Selain itu, blok ini dapat dibaca oleh pemilik kartu. Nominal saldo pada blok ini dibatasi, tergantung dari kebijakan penerbit kartu. Data yang tersimpan di blok ini dienkripsi menggunakan algoritma AES.
3. Sektor ke-1 sampai sektor ke-5 akan menyimpan 5 transaksi pembelian atau *top-up* terakhir dengan konfigurasi blok berikut ini:
 - a. Blok ke-0 yang berupa *reader/writer block* menyimpan waktu transaksi pembelian atau *top-up* dalam bentuk epoch
 - b. Blok ke-1 yang berupa *value block* menyimpan nominal transaksi pembelian atau *top-up*. Transaksi pembelian tidak dapat dilakukan jika saldo *kartu* lebih kecil dari nominal transaksi pembelian.

- c. Blok ke-2 yang berupa *reader/writer block* menyimpan jenis transaksi apakah transaksi pembelian atau *top-up*
4. Sektor ke-6 sampai sektor ke-14 blok ke-0 digunakan untuk menangani kasus di mana transaksi pembelian membutuhkan parameter seperti waktu dan jarak. Masing-masing *merchant* menempati satu sektor. Jumlah maksimal *merchant* yang bekerja sama dengan penerbit kartu adalah 10 *merchant*. Lebih dari itu, penerbit kartu harus mengganti kartu Mifare Classic 1KB dengan kapasitas yang lebih besar. Contoh parameter jarak adalah pembayaran yang dihitung berdasarkan jarak seperti ketika kartu SISMIC digunakan sebagai tiket transportasi untuk pembayaran tol, tiket bus, ataupun tiket kereta. Contoh parameter waktu adalah pembayaran yang dihitung berdasarkan waktu seperti ketika kartu SISMIC digunakan untuk pembayaran tempat bermain anak ataupun *ice skating* yang dihitung per-jam. Perhitungan nominal pembayaran akan ditangani oleh aplikasi *merchant*. Data yang tersimpan di blok ini dienkripsi menggunakan algoritma AES. Sektor ke-6 sampai sektor ke-14 blok ke-1 menyimpan iv yang digunakan untuk mendekripsi blok ke-0 menggunakan algoritma AES.
5. Sektor ke-15 blok ke-1 menyimpan iv yang digunakan untuk mendekripsi sektor ke-0 blok ke-1 menggunakan algoritma AES.
6. Sektor ke-15 blok ke-2 menyimpan iv yang digunakan untuk mendekripsi sektor ke-0 blok ke-2 menggunakan algoritma AES.

Pada tabel Tabel III-2 berikut, dapat dilihat bagaimana hak akses dari masing-masing blok pada kartu SISMIC. Pengaturan hak akses pada kartu SISMIC akan disimpan pada *Secure Access Module* (SAM). Pihak *merchant* akan diberikan *key* kartu SISMIC sesuai dengan hak aksesnya.

Tabel III-2 Hak Akses Kartu SISMIC

| Sektor | Blok | Data | Hak Akses | | | |
|--------|------|---|---------------|----------------|-------------|-----------------|
| | | | Pemilik Kartu | Penerbit Kartu | Merchant | Payment Gateway |
| 0 | 0 | Nomor kartu dari manufaktur | Baca | Baca | - | - |
| | 1 | Tanggal masa berlaku kartu | Baca | Baca | Baca | - |
| | 2 | Saldo Kartu | Baca | Baca, tulis | Baca, tulis | - |
| 1-5 | 0 | Riwayat waktu transaksi pembelian atau <i>top-up</i> dalam bentuk epoch | Baca | Baca, tulis | - | - |
| | 1 | Riwayat nominal transaksi pembelian atau <i>top-up</i> | Baca | Baca, tulis | - | - |
| | 2 | Riwayat jenis transaksi apakah transaksi merupakan transaksi pembelian atau <i>top-up</i> | Baca | Baca, tulis | - | - |

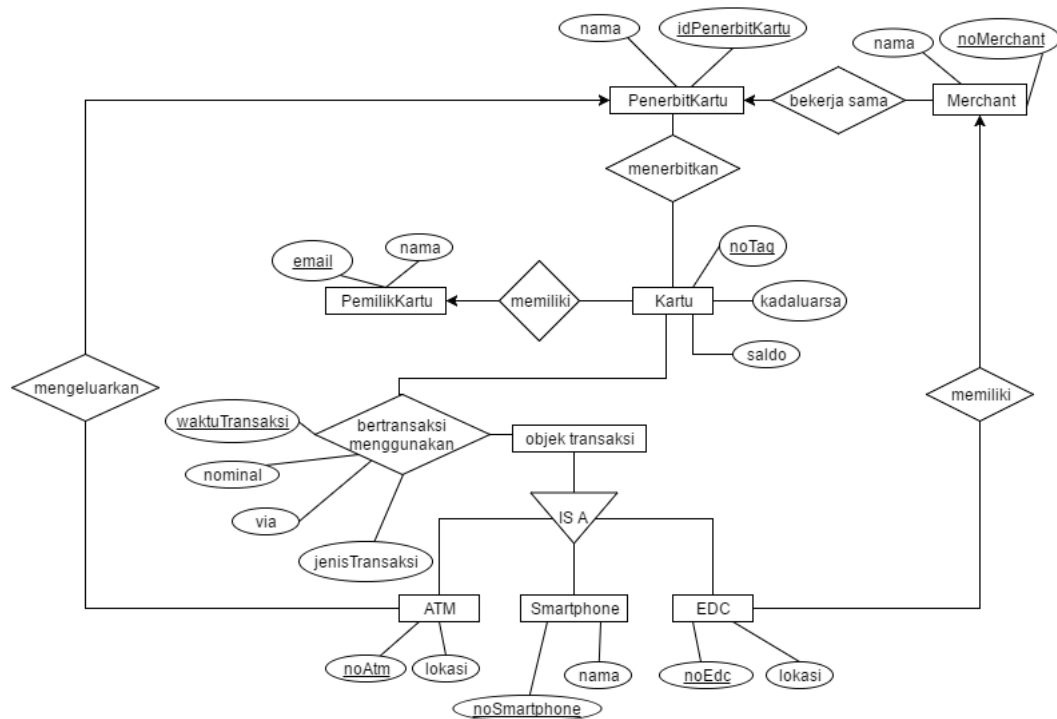
| Sektor | Blok | Data | Hak Akses | | | |
|--------|------|--|---------------|----------------|-------------|-----------------|
| | | | Pemilik Kartu | Penerbit Kartu | Merchant | Payment Gateway |
| 6-14 | 0 | Parameter jarak atau waktu pertama kartu disentuh pada <i>reader</i> | - | - | Baca, tulis | - |
| | 1 | Iv hasil dari enkripsi blok ke-0 di sektor 6 sampai 14 | - | - | - | - |
| | 2 | - | - | - | - | - |
| 15 | 1 | Iv hasil dari enkripsi blok ke-1 di sektor 0 | - | - | - | - |
| | 2 | Iv hasil dari enkripsi blok ke-2 di sektor 0 | - | - | - | - |

Blok-blok ini dapat dilihat oleh pemilik kartu dan ditulis menggunakan algoritma *round-robin* ketika kartu digunakan untuk transaksi pembelian atau *top-up*. Algoritma *round-robin* yang dimaksud adalah di mana transaksi pembelian atau *top-up* yang paling lama disimpan akan ditimpa dengan yang paling baru sampai memenuhi kapasitas penyimpanan riwayat transaksi, yaitu lima buah. *Payment gateway* tidak memiliki hak akses apapun ke kartu SISMIC.

III.2.2.2 Basisdata

Sub subbab ini meliputi pembahasan tentang basisdata pada server SISMIC, *merchant*, dan *payment gateway*.

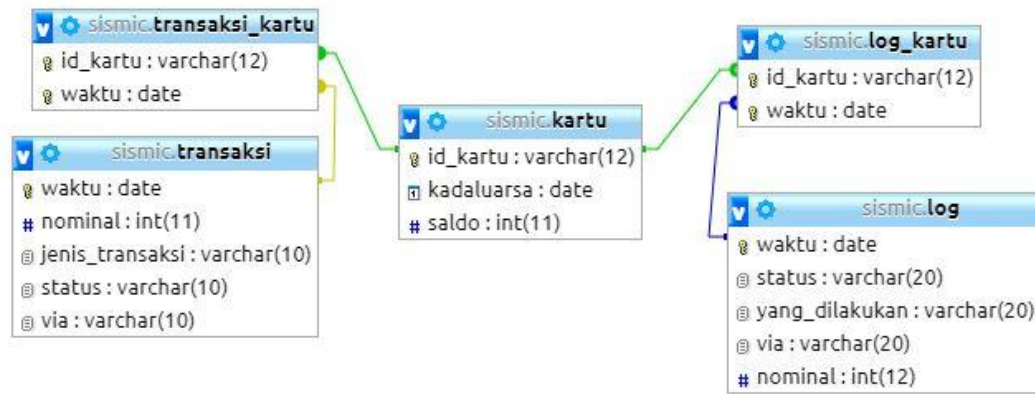
III.2.2.2.1 Model Data SISMIC



Gambar III-15 Diagram *Entity-Relationship* SISMIC

SISMIC memiliki tempat penyimpanan untuk menyimpan data. Penyimpanan data SISMIC disimpan pada basisdata dan kartu SISMIC yang berupa *tag* NFC. Basisdata SISMIC dan kartu SISMIC akan saling bersinkronisasi ketika kartu NFC ditempelkan pada *smartphone* dan mesin EDC. Diagram *Entity-Relationship* (Diagram ER) untuk basisdata SISMIC dapat dilihat pada Gambar III-15 di atas. Ada tujuh entitas yang terlibat di basisdata SISMIC, yaitu Kartu, PenerbitKartu, Merchant, serta objek transaksi yang terdiri dari ATM, Smartphone, dan EDC. Karena kartu tidak ada kepemilikan, entitas pemilik kartu tidak akan dilibatkan pada level basisdata berikutnya. Basisdata yang akan digunakan di SISMIC ada tiga, yaitu basisdata SISMIC, basisdata *merchant*, dan basisdata *payment gateway*.

III.2.2.2.2 Basisdata SISMIC

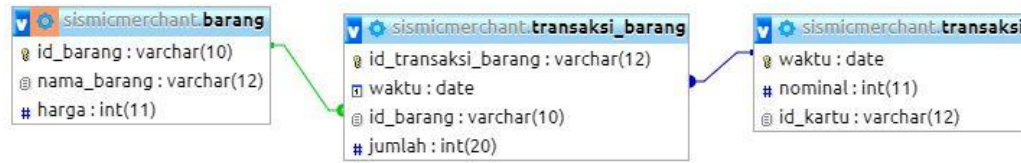


Gambar III-16 Basisdata SISMIC

Skema basisdata relasional untuk basisdata SISMIC dapat dilihat pada Gambar III-16 di atas. Basisdata SISMIC digunakan untuk menyimpan segala riwayat transaksi yang terjadi pada SISMIC. Riwayat transaksi yang disimpan seperti waktu transaksi, nominal transaksi, jenis transaksi apakah transaksi yang dilakukan transaksi pembelian atau *top-up*, status transaksi apakah transaksi yang dilakukan gagal atau berhasil, dan melalui apa transaksi dilakukan, apakah ATM, *merchant*, atau *smartphone*. Selain itu, basisdata SISMIC juga menyimpan informasi kartu yang digunakan untuk transaksi seperti ID kartu, masa berlaku kartu, dan saldo kartu. Basisdata SISMIC dapat dilihat secara lengkap oleh penerbit kartu.

Selain itu, basisdata SISMIC juga mencatat log apa saja yang dilakukan oleh pemilik kartu (lihat saldo, lihat masa berlaku, melakukan transaksi pembelian, dan melakukan transaksi *top-up*), kapan hal tersebut dilakukan, melalui apa transaksi dilakukan, nominal transaksi, dan apakah transaksi berhasil atau gagal.

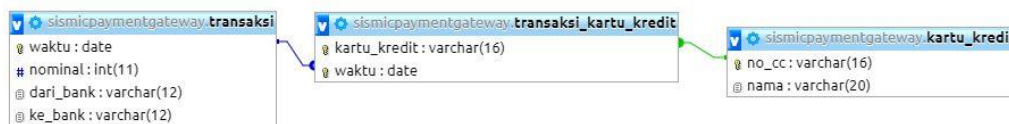
III.2.2.2.3 Basisdata *Merchant*



Gambar III-17 Basisdata *Merchant*

Skema basisdata relasional untuk basisdata SISMIC dapat dilihat pada Gambar III-17 di atas. Basisdata *merchant* yang dibuat pada tugas akhir ini hanyalah yang berpengaruh pada SISMIC saja, bukan basisdata *merchant* keseluruhan. Basisdata *merchant* digunakan untuk menyimpan segala riwayat transaksi pemilik kartu yang terjadi pada *merchant*. Riwayat transaksi yang disimpan adalah ID kartu yang melakukan transaksi, waktu transaksi, nominal transaksi, dan barang apa saja yang dibeli oleh pemilik kartu. Selain riwayat transaksi, basisdata *merchant* juga menyimpan harga barang beserta ID barang yang dijual di *merchat*. Basisdata *merchant* dapat dilihat secara lengkap oleh *merchant*.

III.2.2.2.4 Basisdata *Payment Gateway*



Gambar III-18 Basisdata *Payment Gateway*

Skema basisdata relasional untuk basisdata *payment gateway* dapat dilihat pada Gambar III-18 di atas. Basisdata *payment gateway* yang dibuat pada tugas akhir ini hanyalah yang berpengaruh pada SISMIC saja, bukan basisdata *payment gateway* keseluruhan. Basisdata *payment gateway* digunakan untuk menyimpan segala riwayat transaksi pembayaran yang harus ditangani *payment gateway*. Riwayat transaksi yang disimpan adalah waktu transaksi, nomor kartu kredit yang melakukan transaksi, nominal transaksi, dari bank apa kartu kredit milik pemilik kartu yang digunakan untuk *top-up*, dan ke bank mana transaksi *top-up* dilakukan

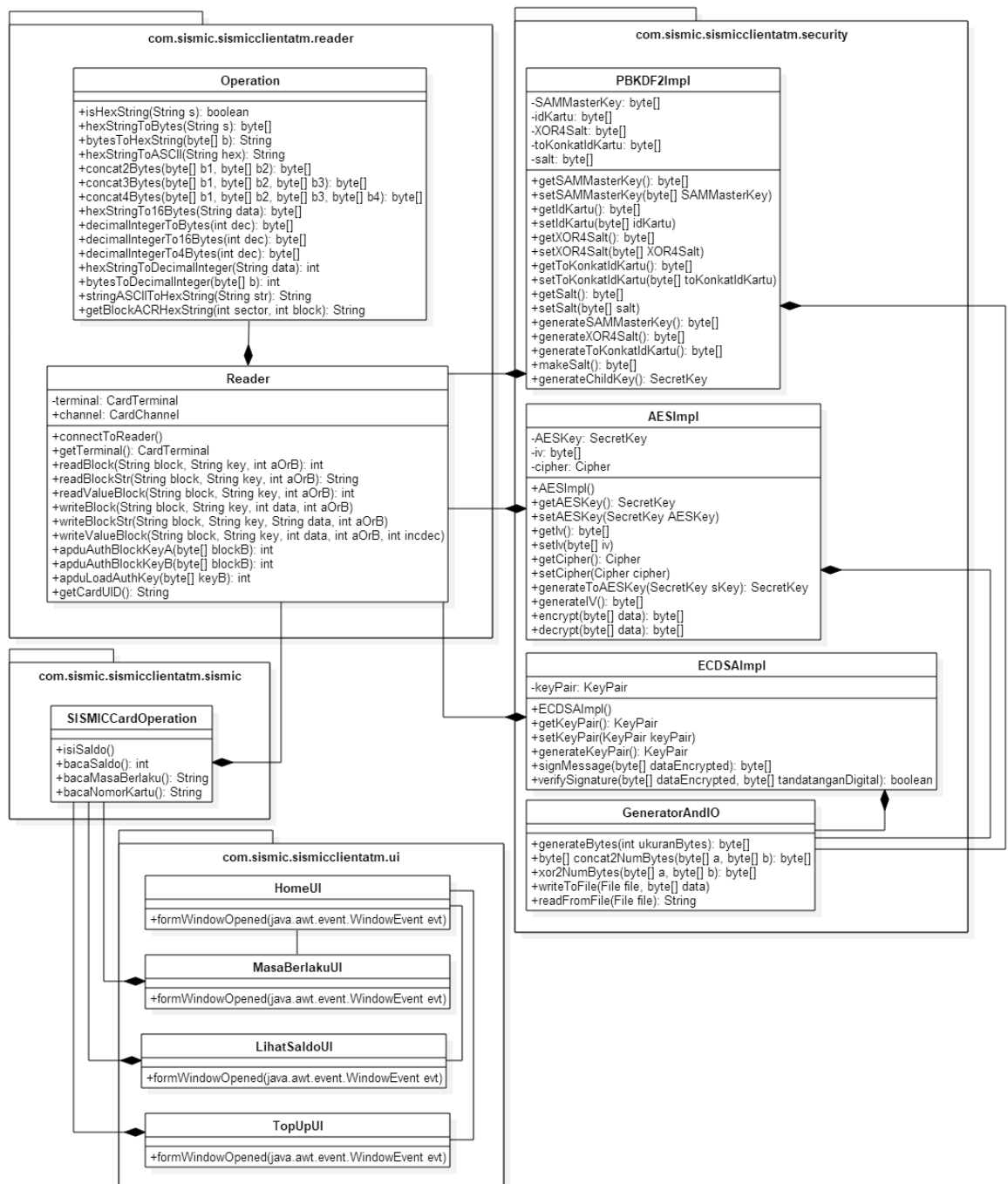
atau bisa disebut juga bank penerbit kartu. Basisdata *merchant* dapat dilihat secara lengkap oleh *payment gateway*.

III.2.2.3 Diagram Kelas

Sub subbab ini meliputi pembahasan tentang diagram kelas yang digunakan untuk aplikasi ATM, aplikasi EDC, aplikasi *Smartphone*, *web service* SISMIC, *web service payment gateway*, dan *web service merchant*.

III.2.2.3.1 Diagram Kelas untuk ATM

Diagram kelas pada Gambar III-19 di bawah adalah diagram kelas yang digunakan untuk aplikasi ATM. Aplikasi ATM ini akan terhubung ke *web service* SISMIC. Diagram kelas ATM memiliki empat *package*, yaitu *package* `com.sismic.sismicclientatm.reader`, `com.sismic.sismicclientatm.security`, `com.sismic.sismicclientatm.sismic`, dan `com.sismic.sismicclientatm.ui`.



Gambar III-19 Diagram Kelas ATM

Package com.sismic.sismicclientatm.reader berisi dua kelas, yaitu kelas Reader dan Operation. *Package* ini berhubungan dengan mesin pembaca kartu SISMIC. Kelas Reader memiliki *methods* untuk membuat koneksi dengan mesin pembaca kartu, membaca kartu, dan menulis kartu SISMIC. Kelas Operation memiliki

methods yang diperlukan untuk operasi-operasi di kelas Reader seperti mengubah bilangan hexa menjadi bytes, menggabungkan dua sampai empat bilangan bytes, dan operasi-operasi lain yang berhubungan dengan bilangan hexa.

Package com.sismic.sismicclientatm.security berisi empat kelas, yaitu kelas PBKDF2Impl, AESImpl, ECDSAImpl, GeneratorAndIO. Kelas PBKDF2Impl memiliki *methods* untuk membuat kunci SAM, membuat salt, dan membuat kunci turunan yang akan digunakan untuk mengenkripsi kartu SISMIC. Kelas AESImpl memiliki *methods* untuk membuat kunci AES dari kunci turunan, membuat IV, mengenkripsi data, dan mendekripsi data. Kelas ECDSAImpl memiliki *methods* untuk membuat kunci publik dan kunci privat, memberi *digital signature* pada data, dan memverifikasi *digital signature*. Kelas GeneratorAndIO memiliki *methods* yang dibutuhkan kelas PBKDF2Impl, AESImpl, dan ECDSAImpl untuk membuat bilangan random, menggabungkan dua bilangan, melakukan operasi XOR, menulis ke *file* txt, dan membaca *file* txt.

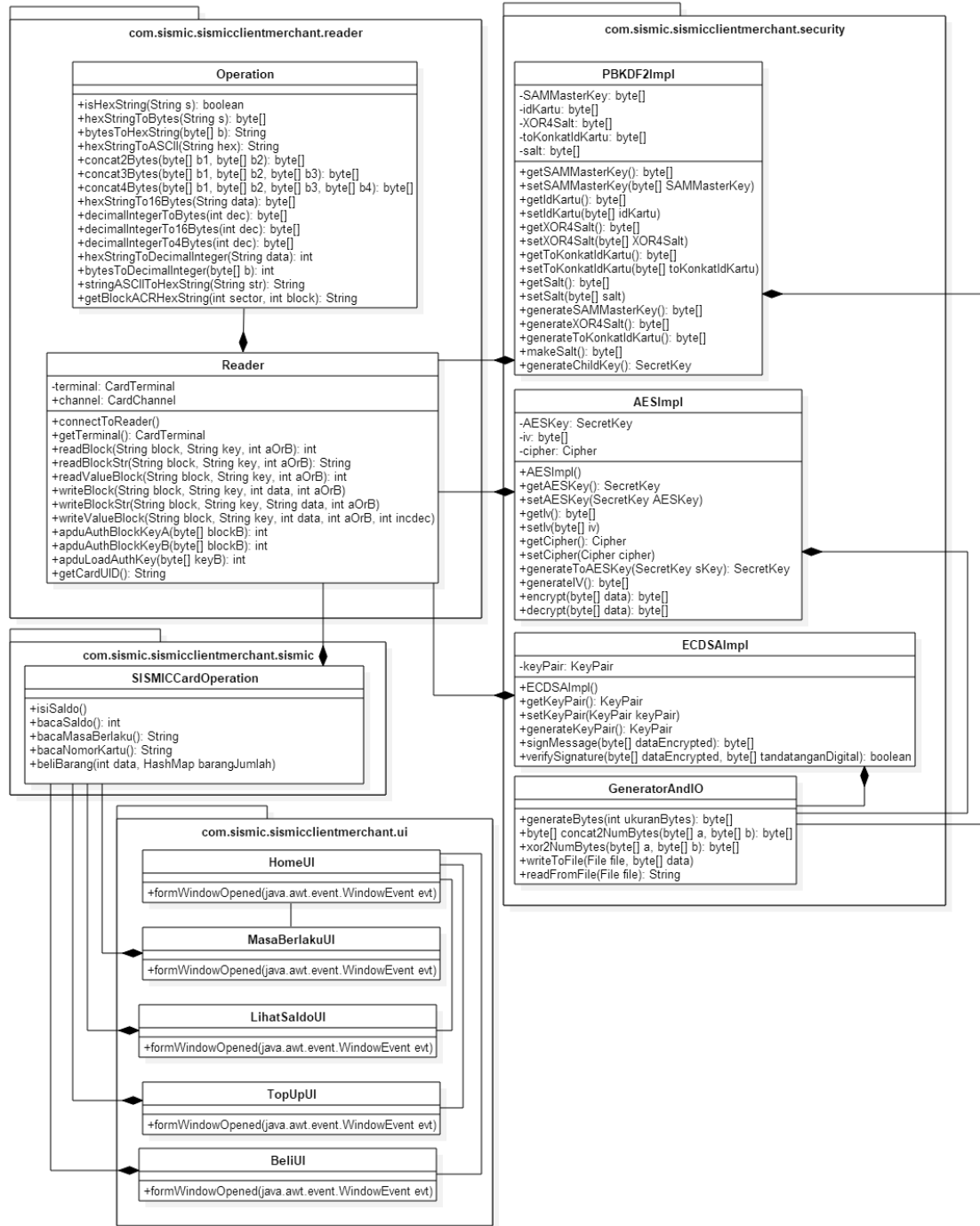
Package com.sismic.sismicclientatm.sismic berisi satu kelas saja, yaitu kelas SISMICCardOperation. Kelas ini memiliki *methods* untuk mengisi saldo kartu, melihat saldo kartu, melihat masa berlaku kartu, dan membaca nomor kartu SISMIC.

Package com.sismic.sismicclientatm.ui berisi empat kelas, yaitu kelas HomeUI, MasaBerlakuUI, LihatSaldoUI, dan TopUpUI. Kelas-kelas ini digunakan untuk menampilkan tampilan aplikasi ATM. Kelas HomeUI menampilkan tampilan menu utama dari aplikasi ATM, kelas MasaBerlaku menampilkan tampilan untuk melihat masa berlaku kartu SISMIC, kelas LihatSaldoUI menampilkan tampilan untuk melihat saldo kartu SISMIC, dan kelas TopUpUI menampilkan tampilan untuk melakukan transaksi *top-up* saldo kartu SISMIC.

III.2.2.3.2 Diagram Kelas Merchant

Diagram kelas pada Gambar III-20 di bawah adalah diagram kelas yang digunakan untuk aplikasi *merchant*. Aplikasi *merchant* ini akan terhubung ke *web service* SISMIC dan *web service merchant*. Diagram kelas *Merchant* memiliki

empat *package*, yaitu *package* com.sismic.sismicclientmerchant.reader, com.sismic.sismicclientmerchant.security, com.sismic.sismicclientmerchant.sismic, dan com.sismic.sismicclientmerchant.ui.



Gambar III-20 Diagram Kelas *Merchant*

Package com.sismic.sismicclientmerchant.reader berisi dua kelas, yaitu kelas Reader dan Operation. *Package* ini berhubungan dengan mesin pembaca kartu SISMIC. Kelas Reader memiliki *methods* untuk membuat koneksi dengan mesin pembaca kartu, membaca kartu, dan menulis kartu SISMIC. Kelas Operation memiliki *methods* yang diperlukan untuk operasi-operasi di kelas Reader seperti mengubah bilangan hexa menjadi bytes, menggabungkan dua sampai empat bilangan bytes, dan operasi-operasi lain yang berhubungan dengan bilangan hexa.

Package com.sismic.sismicclientmerchant.security berisi empat kelas, yaitu kelas PBKDF2Impl, AESImpl, ECDSAImpl, GeneratorAndIO. Kelas PBKDF2Impl memiliki *methods* untuk membuat kunci SAM, membuat salt, dan membuat kunci turunan yang akan digunakan untuk mengenkripsi kartu SISMIC. Kelas AESImpl memiliki *methods* untuk membuat kunci AES dari kunci turunan, membuat IV, mengenkripsi data, dan menkripsi data. Kelas ECDSAImpl memiliki *methods* untuk membuat kunci publik dan kunci privat, memberi *digital signature* pada data, dan memverifikasi *digital signature*. Kelas GeneratorAndIO memiliki *methods* yang dibutuhkan kelas PBKDF2Impl, AESImpl, dan ECDSAImpl untuk membuat bilangan random, menggabungkan dua bilangan, melakukan operasi XOR, menulis ke *file* txt, dan membaca *file* txt.

Package com.sismic.sismicclientmerchant.sismic berisi satu kelas saja, yaitu kelas SISMICCardOperation. Kelas ini memiliki *methods* untuk mengisi saldo kartu, melihat saldo kartu, melihat masa berlaku kartu, melakukan transaksi pembelian, dan membaca nomor kartu SISMIC.

Package com.sismic.sismicclientmerchant.ui berisi empat kelas, yaitu kelas HomeUI, MasaBerlakuUI, LihatSaldoUI, BeliUI, dan TopUpUI. Kelas-kelas ini digunakan untuk menampilkan tampilan aplikasi ATM. Kelas HomeUI menampilkan tampilan menu utama dari aplikasi ATM, kelas MasaBerlaku menampilkan tampilan untuk melihat masa berlaku kartu SISMIC, kelas LihatSaldoUI menampilkan tampilan untuk melihat saldo kartu SISMIC, kelas BeliUI menampilkan tampilan untuk melakukan transaksi pembelian, dan kelas

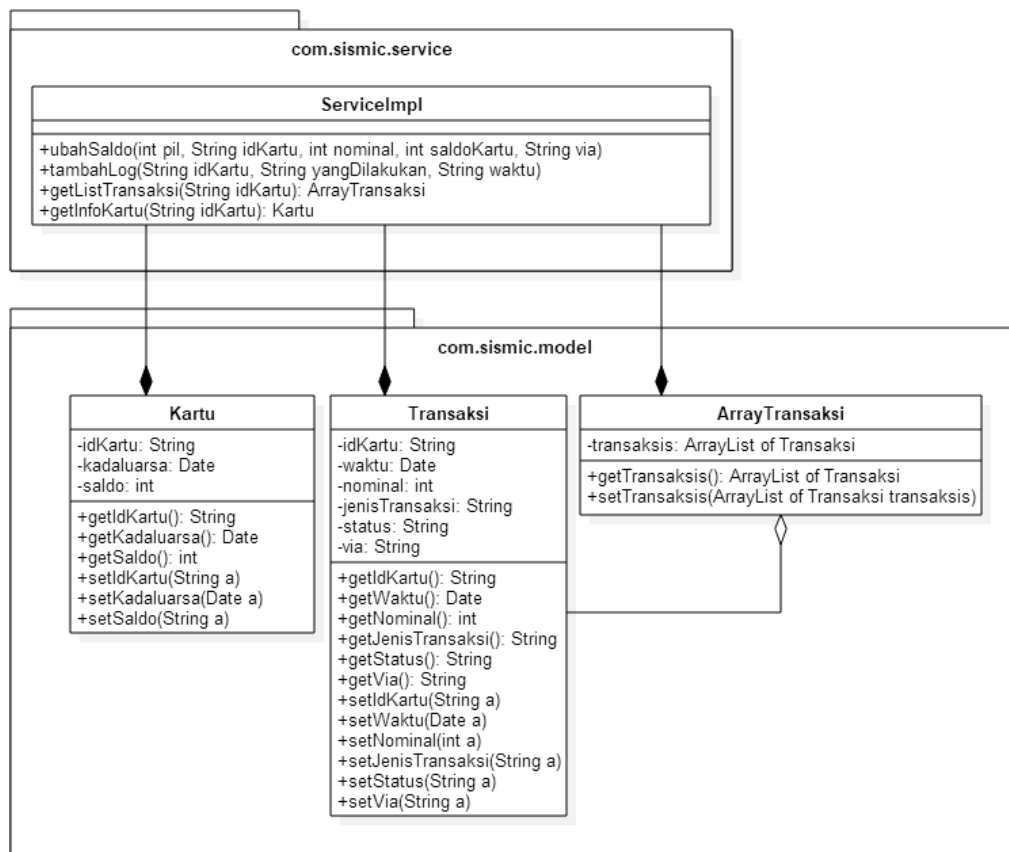
TopUpUI menampilkan tampilan untuk melakukan transaksi *top-up* saldo kartu SISMIC.

III.2.2.3.3 Diagram Kelas *Smartphone*

Aplikasi *smartphone* pada tugas akhir ini tidak diimplementasi.

III.2.2.3.4 Diagram Kelas *Web Service SISMIC*

Diagram kelas SISMIC digunakan untuk *web service* SISMIC yang terhubung ke basisdata SISMIC. Basisdata SISMIC adalah basisdata paling penting agar sistem *micropayment* bisa berjalan dan akan terhubung ke *web service* SISMIC. *Web service* SISMIC akan digunakan oleh seluruh aplikasi yang ada pada SISMIC yaitu aplikasi ATM, aplikasi *merchant*, dan aplikasi *smartphone*. *Web service* SISMIC menangani operasi-operasi yang berhubungan dengan basisdata SISMIC.



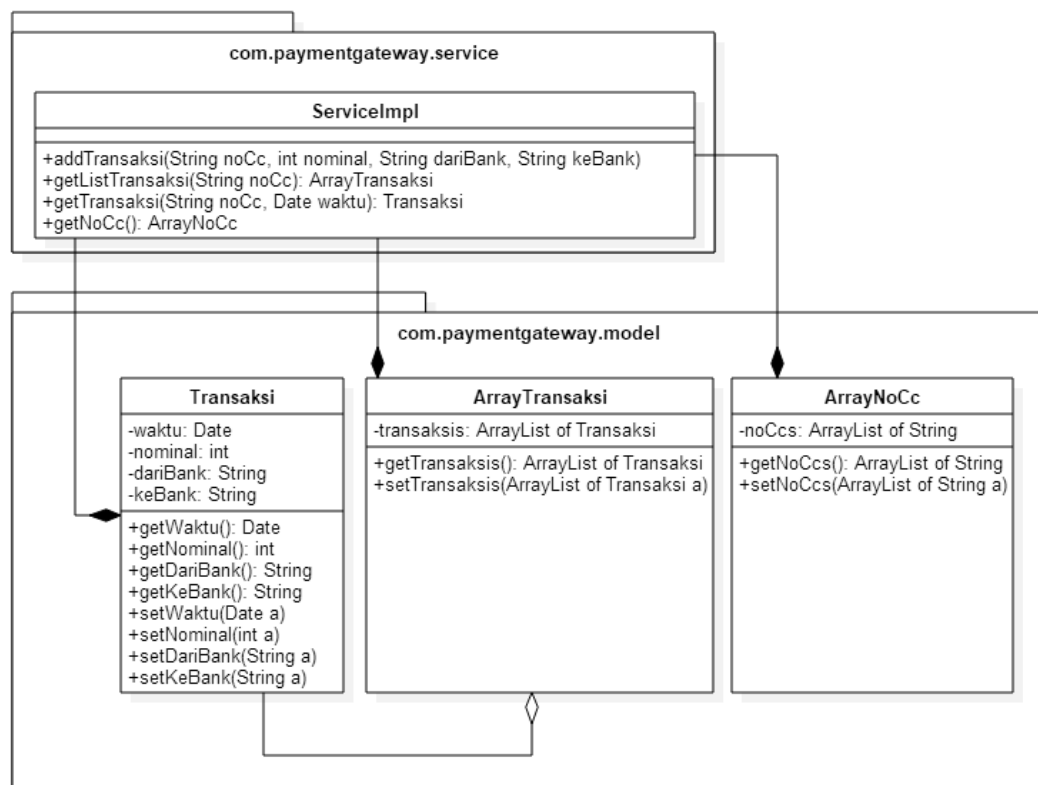
Gambar III-21 Diagram Kelas SISMIC

Bisa dilihat pada Gambar III-21 di atas, diagram kelas SISMIC memiliki dua *package*, yaitu *package* com.sismic.service dan com.sismic.model. com.sismic.service adalah *package* yang berisi *methods* yang ada di *web service*. com.sismic.service hanya memiliki satu kelas, yaitu kelas ServiceImpl yang mengimplementasi *web service*. *Methods* ini akan menulis, mengubah, atau membaca basisdata SISMIC.

Package com.sismic.model adalah *package* yang berisi kelas Kartu dan Transaksi, yaitu kelas yang memodelkan entitas di basisdata SISMIC. Selain kelas Kartu dan Transaksi, *package* com.sismic.model memiliki kelas ArrayTransaksi, yaitu kelas untuk merepresentasikan array dari kelas Transaksi.

III.2.2.3.5 Diagram Kelas Web Service Payment Gateway

Diagram kelas *payment gateway* digunakan untuk *web service payment gateway* yang terhubung ke basisdata *payment gateway*. *Web service payment gateway* akan digunakan oleh aplikasi *smartphone*. *Web service payment gateway* menangani operasi-operasi yang berhubungan dengan basisdata *payment gateway*.



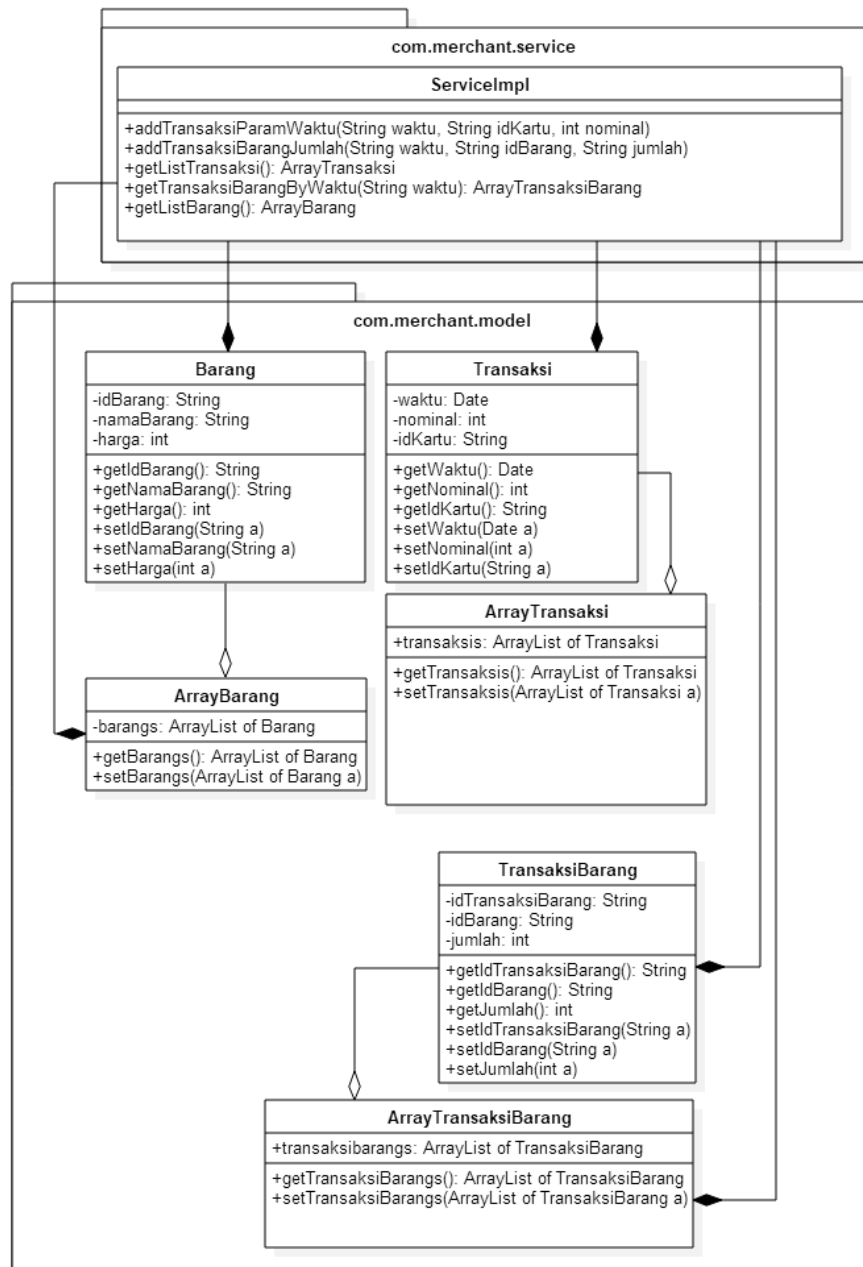
Gambar III-22 Diagram Kelas *Payment Gateway*

Bisa dilihat pada Gambar III-22 di atas, diagram kelas *Payment Gateway* memiliki dua *package*, yaitu *package* `com.paymentgateway.service` dan `com.paymentgateway.model`. `com.paymentgateway.service` adalah *package* yang berisi *methods* yang ada di *web service*. *Package* `com.paymentgateway.service` hanya memiliki satu kelas, yaitu kelas `ServiceImpl` yang mengimplementasi *web service*. *Methods* ini akan menulis, mengubah, atau membaca basisdata *payment gateway*.

Package `com.paymentgateway.model` adalah *package* yang berisi kelas `Transaksi`, yaitu kelas yang memodelkan entitas di basisdata *payment gateway*. Selain kelas `Transaksi`, *package* ini juga memiliki kelas `ArrayTransaksi` yang merepresentasikan array dari kelas `Transaksi`, dan kelas `ArrayNoCc` yang merepresentasikan array dari nomor kartu kredit yang berupa `String`.

III.2.2.3.6 Diagram Kelas *Web Service Merchant*

Diagram kelas *merchant* digunakan untuk *web service merchant* yang terhubung ke basisdata *merchant*. *Web service merchant* akan digunakan oleh aplikasi *smartphone* dan *merchant*. *Web service merchant* menangani operasi-operasi yang berhubungan dengan basisdata *merchant*.



Gambar III-23 Diagram Kelas *Merchant*

Bisa dilihat pada Gambar III-23 di atas, diagram kelas *Merchant* memiliki dua *package*, yaitu *package* `com.merchant.service` dan `com.merchant.model`. `com.merchant.service` adalah *package* yang berisi *methods* yang ada di *web service*. `com.merchant.service` hanya memiliki satu kelas, yaitu kelas `ServiceImpl` yang mengimplementasi *web service*. *Methods* ini akan menulis, mengubah, atau membaca basisdata *merchant*.

Package `com.merchant.model` adalah *package* yang berisi kelas `Barang`, `Transaksi`, dan `TransaksiBarang`, yaitu kelas yang memodelkan entitas di basisdata *merchant*. Selain itu, *package* ini juga memiliki kelas `ArrayBarang` yang merepresentasikan array dari kelas `Barang`, `ArrayTransaksi` yang merepresentasikan array dari kelas `Transaksi`, dan kelas `ArrayTransaksiBarang` yang merepresentasikan array dari kelas `TransaksiBarang`.

III.2.3 Rancangan Keamanan SISMIC

Subbab ini meliputi pembahasan tentang aspek keamanan SISMIC, enkripsi dan dekripsi pada SISMIC, dan manajemen dan distribusi kunci pada SISMIC.

III.2.3.1 Aspek Keamanan SISMIC

Seperti yang disebutkan pada subbab sebelumnya, ada beberapa aspek keamanan yang harus dicapai oleh SISMIC agar SISMIC dapat disebut sebagai sistem *micropayment* yang aman. Berikut ini adalah cara bagaimana agar aspek keamanan SISMIC dapat terpenuhi:

- *Confidentiality*
 - Mengubah *key* A dan *key* B dari semua sektor di kartu SISMIC dengan manajemen kuncinya diatur di SAM
 - Melindungi basisdata SISMIC dengan enkripsi dan *password*
- *Integrity*
 - Menerapkan enkripsi dan dekripsi pada SISMIC
 - Pencatatan (*logging*) segala aktivitas yang terjadi di SISMIC
- *Availability*

- Membuat sistem cadangan redundan sehingga jika sistem utama ada gangguan, ada sistem cadangan yang dapat menggantikannya
- *Authentication*
 - Mengubah *key A* dan *key B* dari semua sektor di kartu SISMIC dengan manajemen kuncinya diatur di SAM
 - SISMIC menyimpan identitas mesin EDC yang dikeluarkan resmi oleh penerbit kartu
- *Authorization*
 - Mengubah *key A* dan *key B* dari semua sektor di kartu SISMIC dengan manajemen kuncinya diatur di SAM
 - *Key A* atau *key B* yang digunakan untuk melakukan transaksi dengan kartu SISMIC berbeda-beda tiap pihak (*merchant*, penerbit kartu, dan pemilik kartu)
- *Accountability dan Non-repudiation*
 - Membuat riwayat transaksi dari segala jenis transaksi yang terjadi, baik yang berhasil maupun yang gagal
 - Pencatatan (*logging*) segala aktivitas yang terjadi di SISMIC

III.2.3.2 Manajemen dan Distribusi Kunci

Pada SISMIC, manajemen dan distribusi kunci akan diatur pada *Secure Access Module* (SAM). SAM akan menyimpan dan mengolah kunci-kunci yang ada pada SISMIC seperti *key A* untuk membaca dan menulis kartu dan kunci AES untuk enkripsi dan dekripsi kartu. SAM sudah memiliki sistem kriptografi sendiri sehingga kunci-kunci yang tersimpan di dalam SAM akan aman. SAM memiliki kunci master yang dapat diturunkan menjadi kunci-kunci lain yang dapat digunakan untuk baca atau tulis kartu, yaitu *key A*, dan untuk enkripsi, yaitu kunci AES.

Kunci AES diturunkan dari kunci master menggunakan algoritma PBKDF2. Kunci AES pada tugas akhir ini berukuran 16 bytes dan diturunkan dengan algoritma PBKDF2 dengan pengulangan 1000 kali. Salt yang digunakan untuk penurunan kunci menggunakan algoritma PBKDF2 adalah ID kartu SISMIC

berukuran 8 bytes disatukan dengan bilangan random berukuran 12 bytes. Setelah itu, hasil konkatenasi tersebut akan di-XOR dengan bilangan random berukuran 20 bytes.

Key A juga diturunkan dari kunci master menggunakan algoritma PBKDF2, sama seperti kunci AES. *Key A* tiap sektor pada kartu SISMIC akan berbeda-beda, *key A* satu kartu SISMIC dengan kartu SISMIC yang lain juga akan berbeda. *Key A* berukuran 6 bytes, namun kunci turunan yang dihasilkan akan berukuran 16 bytes. Kunci turunan ini akan dipotong menjadi 6 bytes sebagai *key A*. *Key A* diturunkan dari kunci master menggunakan algoritma PBKDF2 dengan pengulangan 1000 kali. Salt yang digunakan adalah konkatenasi antara ID kartu SISMIC berukuran 8 bytes, posisi sektor kartu yang akan dibuat kuncinya yang berukuran 1 bytes, dan bilangan random berukuran 11 bytes yang di-XOR dengan bilangan random berukuran 20 bytes. Pada tugas akhir ini, SAM akan disimulasikan dengan disimpan pada aplikasi karena keterbatasan alat.

III.2.3.3 Enkripsi dan Dekripsi pada SISMIC

Enkripsi dan dekripsi pesan yang akan digunakan pada SISMIC menggunakan algoritma AES. Pesan yang dienkripsi akan diberi *digital signature* dengan menggunakan algoritma ECDSA. Tujuan pemberian *digital signature* ini adalah agar sistem mengetahui apakah pesan berasal dari pihak yang benar atau tidak, jika berasal dari pihak yang tidak seharusnya, pengiriman pesan akan digagalkan sehingga tidak akan terjadi transaksi apapun.

Pesan yang akan dikirim ke mesin pembaca kartu SISMIC akan dienkripsi dahulu menggunakan algoritma AES. Enkripsi ini akan membutuhkan kunci AES. Kunci AES didapatkan dari kunci turunan yang disimpan di dalam SAM, seperti yang telah dijelaskan di sub subbab sebelumnya. Kunci AES ini akan digunakan untuk mengenkripsi dan juga mendekripsi pesan.

Setelah dienkripsi, pesan tersebut akan diberi *digital signature* menggunakan kunci privat yang dibuat dari aplikasi SISMIC. Aplikasi SISMIC ini akan membuat kunci privat dan kunci publik. Lalu, pesan akan dikirim ke mesin

pembaca kartu. Mesin pembaca kartu akan memverifikasi pesan dengan memeriksa *digital signature* yang berada di pesan yang terenkripsi. Verifikasi akan menggunakan kunci publik. Jika *digital signature* terbukti berasal dari pihak yang benar, pesan akan didekripsi menggunakan kunci AES dan instruksi seperti baca saldo, transaksi pembelian, ataupun transaksi *top-up* saldo dapat dilakukan. Aspek keamanan yang dilindungi dengan enkripsi dan dekripsi ini adalah *integrity*.

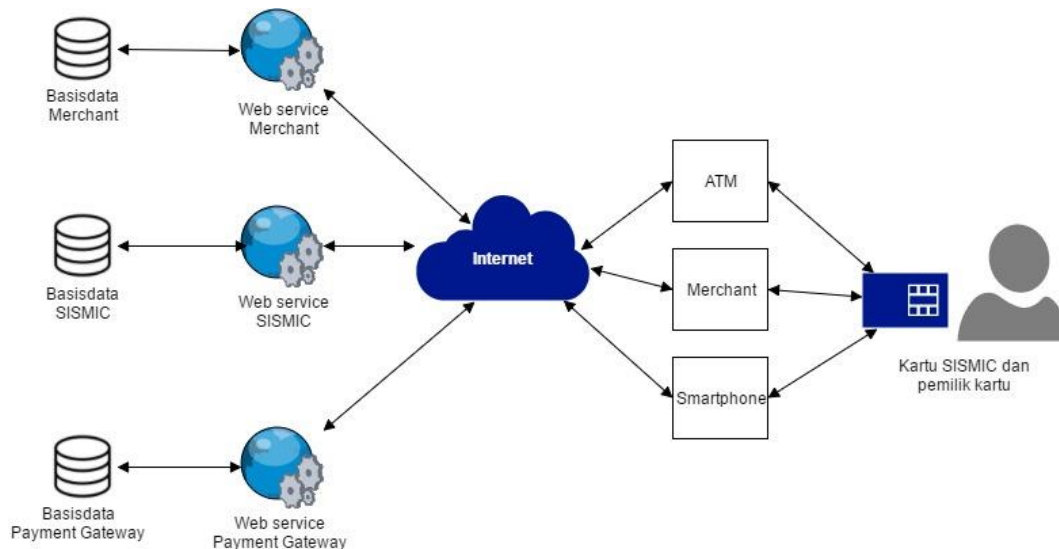
III.2.3.4 Log Offline di Mesin Pembaca Kartu SISMIC

Pada mesin EDC yang berada di *merchant* ataupun mesin ATM, mesin pembaca kartu SISMIC menyimpan log transaksi kartu SISMIC milik pemilik kartu, baik transaksi *top-up* atau pembelian. Log ini bersifat *offline* tersimpan di dalam mesin pembaca kartu SISMIC. Log ini dilindungi dengan enkripsi menggunakan algoritma AES, dan kuncinya akan tersimpan di SAM, sehingga pihak *merchant* ataupun bank tidak bisa sembarangan mengutak-atik log ini. Isi log ini adalah ID kartu dari pemilik kartu yang melakukan transaksi, waktu transaksi, status, transaksi yang dilakukan, via, dan nominal. Tidak hanya transaksi *top-up* ataupun pembelian yang disimpan di log, operasi baca saldo dan lihat masa berlaku juga akan disimpan. Log ini berfungsi agar *merchant* dapat meminta pembayaran hasil transaksi pembelian pemilik kartu ke penerbit kartu dan begitu juga sebaliknya, agar penerbit kartu dapat meminta pembayaran hasil transaksi *top-up* kartu ke *merchant*. Log *offline* ini akan disimpan dengan diekripsi terlebih dahulu menggunakan algoritma AES yang kuncinya disimpan di SAM.

BAB IV

IMPLEMENTASI

IV.1 Arsitektur SISMIC



Gambar IV-1 Arsitektur SISMIC

Gambar IV-1 adalah arsitektur SISMIC. Arsitektur SISMIC terdiri dari tiga basisdata yang berbeda, tiga *web service* yang berbeda, aplikasi ATM, aplikasi mesin EDC, aplikasi *smartphone* milik SISMIC, aplikasi *smartphone* milik *merchant*, dan kartu SISMIC. Tiga basisdata adalah basisdata milik SISMIC, basisdata milik *merchant*, dan basisdata milik *payment gateway*. Masing-masing basisdata akan dioperasikan oleh *web service SISMIC*, *web service merchant*, dan *web service payment gateway*. Ada empat aplikasi yang ada pada SISMIC, yaitu aplikasi *smartphone SISMIC* dan *merchant*, ATM, dan *merchant*. *Web service SISMIC* digunakan oleh seluruh aplikasi yang ada di SISMIC. Sedangkan *Web service merchant* digunakan oleh aplikasi *smartphone* milik *merchant* dan aplikasi *merchant*. *Web service payment gateway* digunakan oleh aplikasi *smartphone SISMIC*. Kartu SISMIC milik pemilik kartu dapat digunakan untuk membeli barang ataupun *top-up* saldo dengan menggunakan aplikasi *smartphone SISMIC* dan *merchant*, ATM, dan *merchant*.

IV.2 Pemilihan Kakas dan *Library*

Implementasi tugas akhir menggunakan beberapa kakas dan *library*. Berikut ini adalah kakas dan *library* yang digunakan pada tugas akhir:

1. IDE NetBeans yang digunakan untuk mengimplementasi tugas akhir menggunakan bahasa Java.
2. Basisdata SISMIC, basisdata *merchant*, dan basisdata *payment gateway* menggunakan NoSQL dan disimpan *online* di Firebase. *Library* Firebase yang digunakan pada tugas akhir ini adalah versi 2.5.2.
3. CFX JAXWS untuk Heroku yang dibuat oleh Chamerling (<https://github.com/chamerling/heroku-cxf-jaxws>). *Library* ini digunakan agar *web service* SISMIC, *merchant*, dan *payment gateway* dapat berjalan di Heroku.
4. API Java Smart Card I/O untuk kartu SISMIC dan mesin pembaca kartu SISMIC. API ini digunakan agar program dapat membaca ataupun menulis kartu SISMIC.
5. *Library* Bouncy Castle untuk implementasi enkripsi dan dekripsi menggunakan algoritma PBKDF2, AES, dan ECDSA.

IV.3 Lingkungan Implementasi

SISMIC akan diimplementasi pada lingkungan sebagai berikut:

1. Sistem operasi: Windows 10 64-bit untuk aplikasi yang disimulasikan di *desktop* dan Android untuk aplikasi *smartphone*
2. Bahasa pemrograman: Java
3. Basisdata: NoSQL dengan disimpan di Firebase
4. *Web Service*: CXF JAXWS di Heroku

IV.4 Hasil Implementasi

Sub subbab ini meliputi pembahasan tentang tampilan hasil implementasi.

IV.4.1 Tampilan Hasil Implementasi

Sub subbab ini meliputi pembahasan tentang tampilan hasil implementasi dan analisis hasil implementasi pada aplikasi *smartphone* SISMIC dan *merchant*, ATM, dan *merchant*.

IV.4.1.1 Aplikasi *Smartphone* SISMIC

Aplikasi *smartphone* SISMIC dapat melakukan hal-hal berikut ini:

1. Melakukan *top-up* saldo kartu SISMIC
2. Melihat saldo kartu SISMIC
3. Melihat masa berlaku kartu SISMIC
4. Melihat riwayat transaksi kartu SISMIC

Aplikasi *smartphone* SISMIC pada tugas akhir ini tidak diimplementasi.

IV.4.1.2 Aplikasi *Smartphone Merchant*

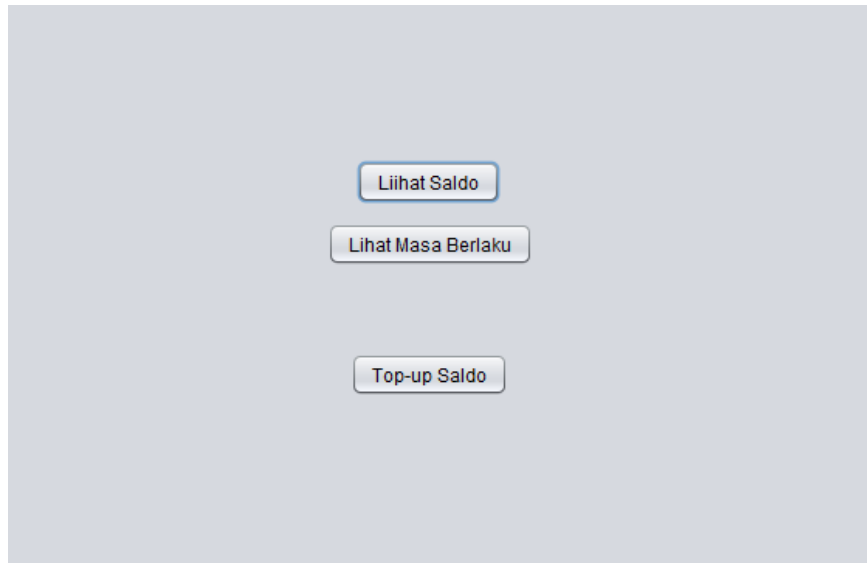
Aplikasi *smartphone merchant* dapat melakukan transaksi pembelian menggunakan kartu SISMIC. Barang-barang yang dapat dibeli adalah barang yang dijual oleh *merchant* yang bersangkutan. Aplikasi *smartphone* pada tugas akhir ini tidak diimplementasi.

IV.4.1.3 Aplikasi ATM

Aplikasi ATM yang disimulasikan melalui *desktop* dapat melakukan hal-hal berikut ini:

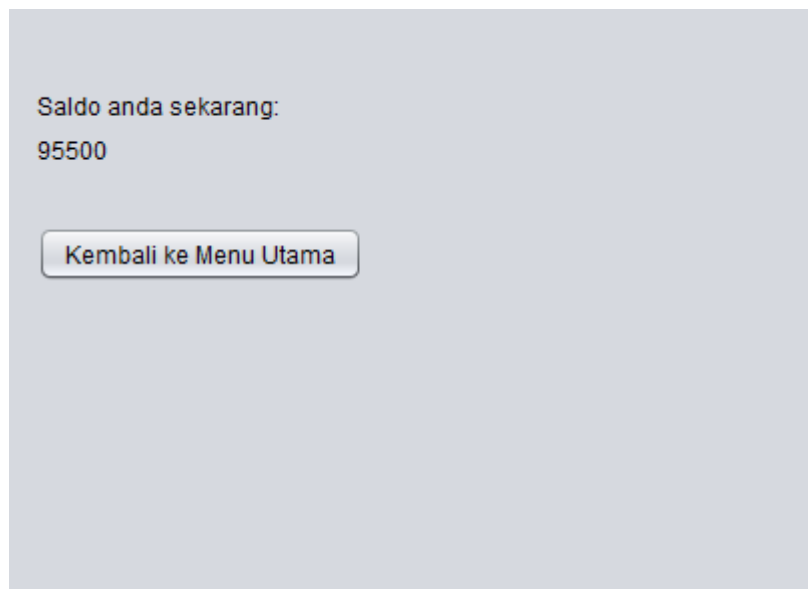
1. Melakukan *top-up* saldo kartu SISMIC
2. Melihat saldo kartu SISMIC
3. Melihat masa berlaku kartu SISMIC

Berikut ini adalah tampilan-tampilan yang ada pada aplikasi ATM. Gambar IV-2 berikut ini adalah tampilan menu utama dari aplikasi ATM. Bisa dilihat, ada tiga pilihan menu, yaitu menu “Lihat Saldo” untuk melihat saldo SISMIC, menu “Lihat Masa Berlaku” untuk melihat masa berlaku kartu SISMIC, dan menu “Top-Up Saldo” untuk *top-up* saldo kartu SISMIC.



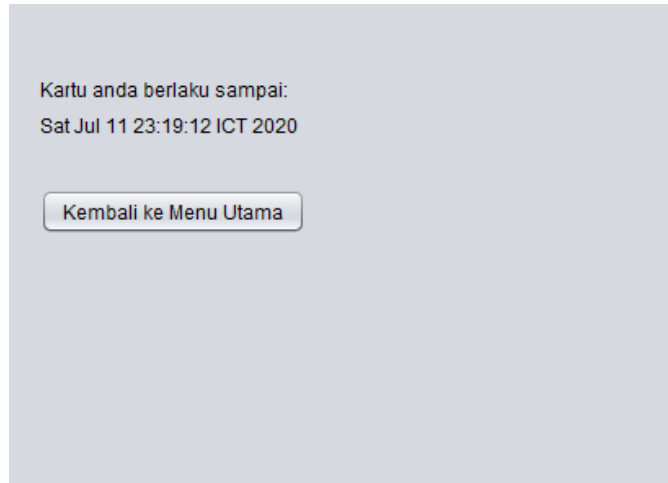
Gambar IV-2 Menu Utama Aplikasi ATM

Gambar IV-3 di bawah ini adalah menu “Lihat Saldo” pada aplikasi ATM. Pada menu ini, saldo kartu SISMIC ditampilkan pada layar. Tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi ATM.



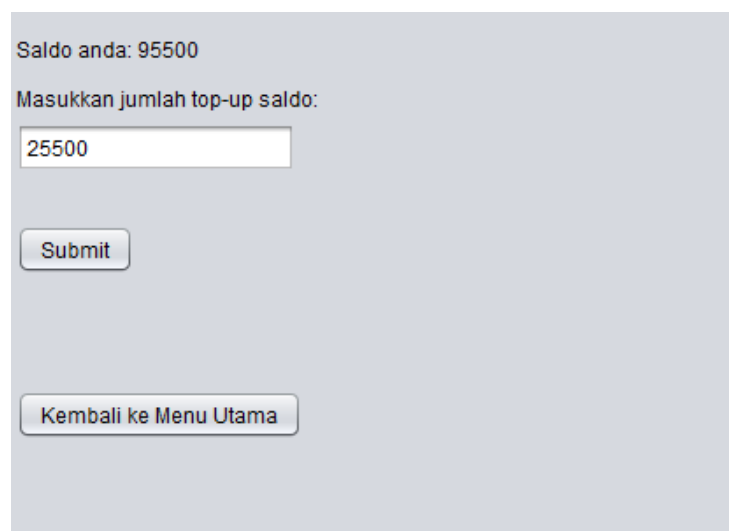
Gambar IV-3 Menu Lihat Saldo Aplikasi ATM

Gambar IV-4 di bawah ini adalah menu “Lihat Masa Berlaku”. Pada menu ini, tanggal masa berlaku kartu SISMIC ditampilkan pada layar. Tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi ATM.



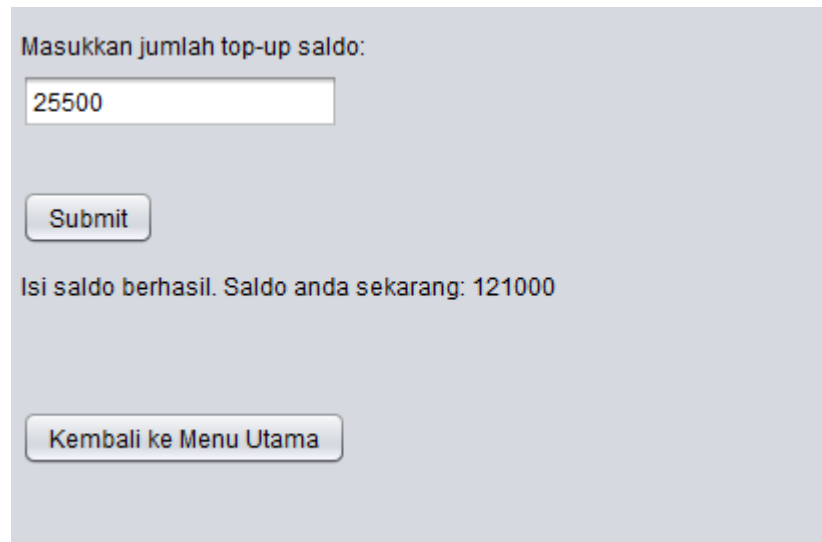
Gambar IV-4 Menu Lihat Masa Berlaku Aplikasi ATM

Gambar IV-5 di bawah ini adalah menu “Top-Up Saldo”. Pada menu ini, saldo kartu SISMIC ditampilkan pada layar. Selain itu, form *top-up* saldo juga ditampilkan. Tombol “Submit” adalah tombol untuk melakukan *top-up* saldo. Lalu, tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi ATM.



Gambar IV-5 Menu Top-Up Saldo Aplikasi ATM

Gambar IV-6 adalah tampilan layar setelah berhasil melakukan *top-up* saldo SISMIC. Tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi ATM.



The screenshot shows a light gray background with a white text box at the top containing the instruction "Masukkan jumlah top-up saldo:". Below this is a white input field with the number "25500" entered. Under the input field is a rounded rectangular button labeled "Submit". Below the button, the text "Isi saldo berhasil. Saldo anda sekarang: 121000" is displayed. At the bottom of the screen is another rounded rectangular button labeled "Kembali ke Menu Utama".

Gambar IV-6 Menu Top-Up Saldo Aplikasi ATM Setelah Top-Up

IV.4.1.4 Aplikasi *Merchant*

Aplikasi *merchant* yang disimulasikan melalui *desktop* dapat melakukan hal-hal berikut ini:

1. Melakukan transaksi pembelian menggunakan kartu SISMIC
2. Menunjukkan saldo kartu SISMIC setelah melakukan transaksi pembelian
3. Melakukan *top-up* saldo kartu SISMIC
4. Melihat saldo kartu SISMIC
5. Melihat masa berlaku kartu SISMIC

Berikut ini adalah tampilan-tampilan yang ada pada aplikasi *merchant*. Gambar Gambar IV-7 berikut ini adalah tampilan menu utama dari aplikasi *merchant*. Bisa dilihat, ada enam pilihan menu, yaitu menu “Beli Barang I” untuk melakukan transaksi pembelian, menu “Beli Barang II” untuk melakukan transaksi pembelian, menu “Ice Skating” untuk melakukan transaksi pembelian dengan parameter, menu “Top-Up Saldo” untuk *top-up* saldo kartu SISMIC, menu “Lihat

Saldo” untuk melihat saldo SISMIC, dan menu “Lihat Masa Berlaku” untuk melihat masa berlaku kartu SISMIC



Gambar IV-7 Menu Utama Aplikasi *Merchant*

Gambar IV-8 di bawah ini adalah menu “Beli Barang I”. Menu “Beli Barang I” adalah menu untuk melakukan transaksi pembelian yang barangnya tidak tercantum di aplikasi *merchant*. Pada menu ini, saldo kartu SISMIC ditampilkan pada layar. Selain itu, form untuk menuliskan nominal transaksi pembelian yang akan dilakukan juga ditampilkan. Tombol “Submit” adalah tombol untuk melakukan transaksi pembelian. Lalu, tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi *merchant*.

Saldo anda: 121000

Masukkan nominal transaksi:

Submit

Kembali ke Menu Utama

Gambar IV-8 Menu Beli Barang I Aplikasi *Merchant*

Gambar IV-9 adalah tampilan layar setelah berhasil melakukan transaksi pembelian. Tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi *merchant*.

Saldo anda: 100000

Masukkan nominal transaksi:

Submit

Transaksi berhasil. Saldo anda sekarang: 100000

Kembali ke Menu Utama

Gambar IV-9 Menu Beli Barang I Aplikasi *Merchant* Setelah Transaksi

Gambar IV-10 adalah menu “Beli Barang II”. Menu “Beli Barang II” adalah menu untuk melakukan transaksi pembelian yang barangnya tercantum di aplikasi *merchant*. Pada menu ini, saldo kartu SISMIC ditampilkan pada layar. Selain itu, nama barang dan form untuk menuliskan jumlah barang yang akan dibeli juga ditampilkan. Tombol “Submit” adalah tombol untuk melakukan transaksi

pembelian. Lalu, tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi *merchant*.



Saldo anda: 100000

A01 Sewa Sepatu

A02 Sewa Loker

A03 Kaos Kaki

Beli

Kembali ke Menu Utama

Gambar IV-10 Menu Beli Barang II Aplikasi *Merchant*

Gambar IV-11 adalah tampilan layar setelah berhasil melakukan transaksi pembelian. Tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi *merchant*.



Saldo anda: 65000

A01 Sewa Sepatu

A02 Sewa Loker

A03 Kaos Kaki

Beli

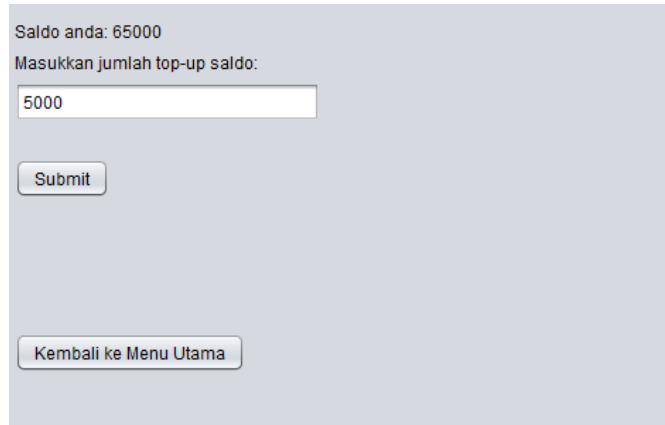
Transaksi berhasil. Saldo anda sekarang: 65000

Kembali ke Menu Utama

Gambar IV-11 Menu Beli Barang II Aplikasi *Merchant* Setelah Transaksi

Gambar IV-12 di bawah ini adalah menu “Top-Up Saldo”. Pada menu ini, saldo kartu SISMIC ditampilkan pada layar. Selain itu, form *top-up* saldo juga

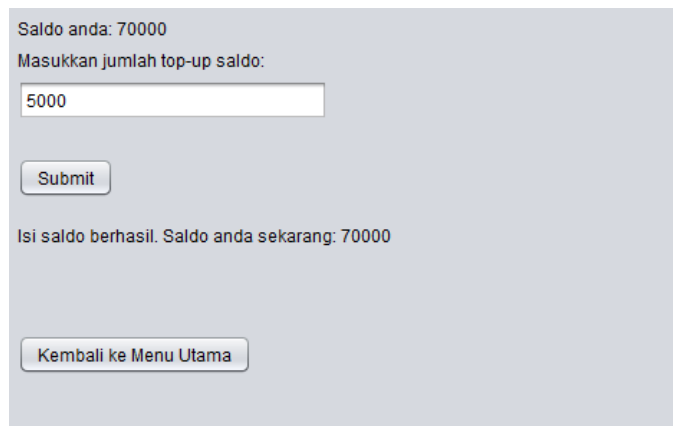
ditampilkan. Tombol “Submit” adalah tombol untuk melakukan *top-up* saldo. Lalu, tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi *merchant*.



Saldo anda: 65000
Masukkan jumlah top-up saldo:

Gambar IV-12 Menu Top-Up Saldo Aplikasi *Merchant*

Gambar IV-13 adalah tampilan layar setelah berhasil melakukan *top-up* saldo SISMIC. Tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi *merchant*.

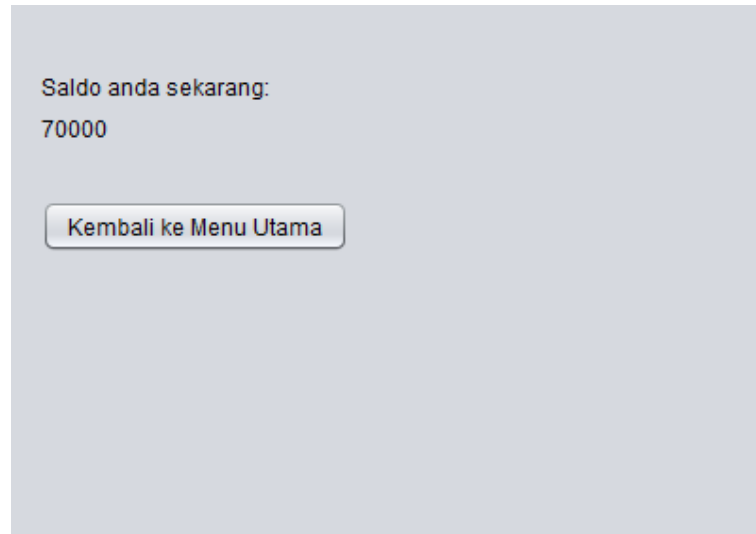


Saldo anda: 70000
Masukkan jumlah top-up saldo:

Isi saldo berhasil. Saldo anda sekarang: 70000

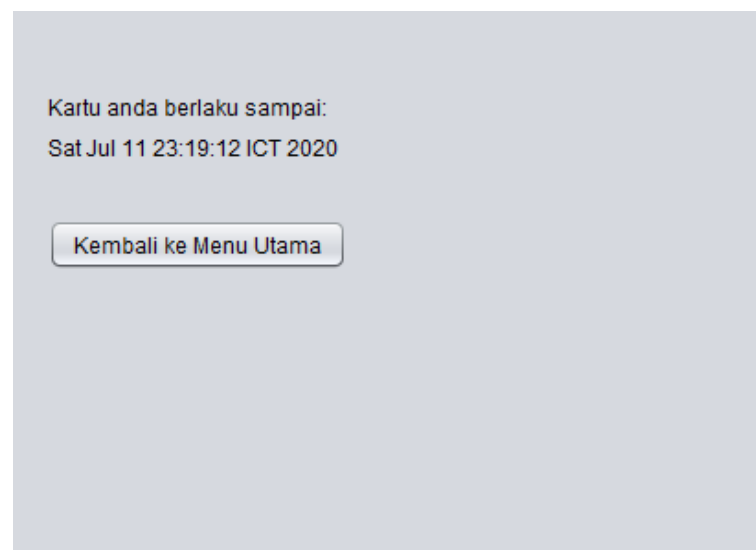
Gambar IV-13 Menu Top-Up Saldo Aplikasi *Merchant* Setelah *Top-Up*

Gambar IV-14 di bawah ini adalah menu “Lihat Saldo” pada aplikasi *merchant*. Pada menu ini, saldo kartu SISMIC ditampilkan pada layar. Tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi *merchant*.



Gambar IV-14 Menu Lihat Saldo Aplikasi *Merchant*

Gambar IV-15 di bawah ini adalah menu “Lihat Masa Berlaku”. Pada menu ini, tanggal masa berlaku kartu SISMIC ditampilkan pada layar. Tombol “Kembali ke Menu Utama” adalah tombol untuk kembali ke menu utama aplikasi *merchant*.



Gambar IV-15 Menu Lihat Masa Berlaku Aplikasi *Merchant*

DAFTAR PUSTAKA

- Chen, C.-H., Lin, I.-C., & Yang, C.-C. (2014). NFC Attacks Analysis and Survey.
- Coskun, V., Ok, K., & Ozdenizci, B. (2013). *Professional NFC Application Development For Android™*. United Kingdom: John Wiley & Sons Ltd.
- Gulati, V. P., & Srivastava, S. (2007). The Empowered Internet Payment Gateway. *Towards Next Generation E-government*, 98-107.
- Igoe, T., Coleman, D., & Jepson, B. (2014). *Beginning NFC*. Sebastopol: O'Reilly.
- Jara, A. J., Zamora, M. A., & S, A. F. (2009). Secure use of NFC in medical environments. 8.
- Knudsen, J. (1998, May). *Java Cryptography*. O'Reilly.
- Munir, R. (2004). Diklat Kuliah Kriptografi. Bandung, Indonesia.
- NFC Forum. (2015). *NFC Forum - NFC Forum Specification Architecture*. Retrieved 12 3, 2015, from NFC Forum: <http://nfc-forum.org/our-work/specifications-and-application-documents/specifications/>
- NFC Forum. (2015). *NFC Forum - NFC in Action*. Retrieved 12 3, 2015, from NFC Forum: <http://nfc-forum.org/what-is-nfc/nfc-in-action/>
- NFC Forum. (2016, Desember 27). *What is NFC?* Retrieved from NFC Forum: <http://nfc-forum.org/what-is-nfc/>
- Rankl, W., & Effing, W. (2010). *Smart Card Handbook Fourth Edition*. United Kingdom: John Wiley & Sons, Ltd.
- Society, T. I. (2000). *RFC 2898 - PKCS #5: Password-Based Cryptography Specification Version 2.0*. Retrieved from IETF Tools: <https://tools.ietf.org/html/rfc2898#section-5.2>

Stallings, W. (2005). *Cryptography and Network Security: Principles and Practice, Fourth Edition*. New Jersey: Prentice Hall.

WhatIs.com. (2016, December 28). *What is micropayment? - Definition from WhatIs.com*. Retrieved from WhatIs.com:
<http://whatis.techtarget.com/definition/micropayment>

Lampiran A. Contoh Judul Lampiran

A.1 Membaca Data *Reader/Writer Block* dari Tag NFC

Membaca *reader/writer block* dari *tag* NFC dapat dilakukan dengan mengirimkan instruksi APDU dari NFC *reader* ke *tag* NFC menggunakan CommandAPDU seperti *load* kunci atau autentikasi *block*, hanya berbeda instruksi APDU-nya saja. Instruksi APDU untuk membaca *reader/writer block* ada pada Tabel A.1 berikut yang berukuran 5 *bytes*.

Tabel A.1 Instruksi APDU Baca *Reader/Writer Block*

| CLA (1 byte) | INS (1 byte) | P1 (1 byte) | P2 (1 byte) | Le (1 byte) |
|--------------|--------------|-------------|---|--|
| 0xFF | 0xB0 | 0x00 | <i>Reader/writer block</i> yang akan dibaca (0x00 ~ 0x3F) | Ukuran <i>bytes</i> yang akan dibaca (maksimal 16 <i>byte</i> , 0x00 ~ 0x10) |

Respon *tag* NFC setelah dikirim instruksi APDU untuk membaca *reader/writer block* terdiri dari 0 sampai 16 *bytes* data yang di baca, 1 *byte* SW1, dan 1 *byte* SW2. Apabila SW1 bernilai 0x90 dan SW2 bernilai 0x00, pembacaan *block* berhasil. Namun apabila SW1 bernilai 0x63 dan SW2 bernilai 0x00, pembacaan *block* gagal.

A.2 Membaca Data *Value Block* dari Tag NFC

Sama seperti *reader/writer block*, membaca *value block* dari *tag* NFC dapat dilakukan dengan mengirimkan instruksi APDU dari NFC *reader* ke *tag* NFC menggunakan CommandAPDU. Instruksi APDU untuk membaca *value block* ada pada Tabel A.2 berikut.

Tabel A.2 Instruksi APDU Baca *Value Block*

| CLA (1 byte) | INS (1 byte) | P1 (1 byte) | P2 (1 byte) | Le (1 byte) |
|--------------|--------------|-------------|---|-------------|
| 0xFF | 0xB1 | 0x00 | <i>Value block</i> yang akan dibaca (0x00 ~ 0x3F) | 0x04 |

Respon *tag* NFC setelah dikirim instruksi APDU pada tabel x terdiri dari 4 *bytes* data yang di baca (berupa bilangan *signed long integer*), 1 *byte* SW1, dan 1 *byte* SW2. Apabila SW1 bernilai 0x90 dan SW2 bernilai 0x00, pembacaan *block* berhasil. Namun apabila SW1 bernilai 0x63 dan SW2 bernilai 0x00, pembacaan *block* gagal.

A.3 Menulis Data ke *Reader/Writer Block Tag* NFC

Menulis data ke *reader/writer block tag* NFC dapat dilakukan dengan mengirimkan instruksi APDU dari NFC *reader* ke *tag* NFC menggunakan CommandAPDU. Instruksi APDU untuk menulis *reader/writer block* ada pada Tabel A.3 berikut.

Tabel A.3 Instruksi APDU Tulis Data ke *Reader/Writer Block*

| CLA (1 byte) | INS (1 byte) | P1 (1 byte) | P2 (1 byte) | Lc (1 byte) | Data (6 bytes) |
|--------------|--------------|-------------|--|--|------------------------|
| 0xFF | 0xD6 | 0x00 | <i>Reader/writer block</i> yang akan ditulis (0x00 ~ 0x3F) | Jumlah <i>bytes</i> yang akan ditulis (0x10) | Data yang akan ditulis |

Respon *tag* NFC setelah dikirim instruksi APDU untuk menulis *reader/writer block* terdiri dari 1 *byte* SW1 dan 1 *byte* SW2. Apabila SW1 bernilai 0x90 dan SW2 bernilai 0x00, penulisan *reader/writer block* berhasil. Namun apabila SW1 bernilai 0x63 dan SW2 bernilai 0x00, penulisan *reader/writer block* gagal.

A.4 Menulis Data ke *Value Block* Tag NFC

Menulis data ke *value block* dapat berupa *increment* atau *decrement*. *Increment* atau *decrement value block* dapat dilakukan dengan mengirimkan instruksi APDU dari NFC *reader* ke *tag* NFC menggunakan CommandAPDU. Instruksi APDU untuk menulis *reader/writer block* ada pada Tabel A.4 berikut.

Tabel A.4 Instruksi APDU Tulis Data ke *Value Block*

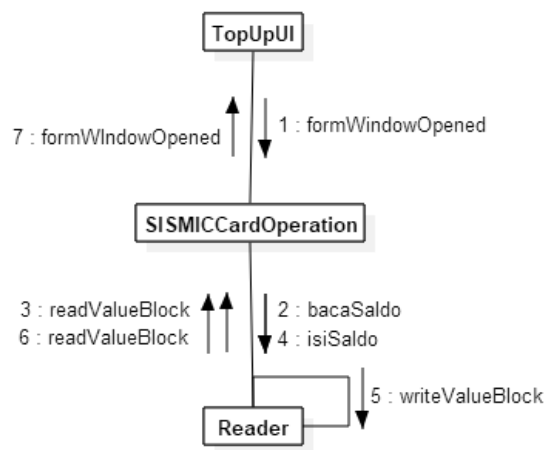
| CLA (1 <i>byte</i>) | INS (1 <i>byte</i>) | P1 (1 <i>byte</i>) | P2 (1 <i>byte</i>) | Lc (1 <i>byte</i>) | Data (5 <i>bytes</i>) | |
|-------------------------|-------------------------|------------------------|--|------------------------|---|---|
| 0xFF | 0xD7 | 0x00 | <i>Value block</i> yang akan ditulis (0x00 ~ 0x3F) | 0x05 | 0x01 untuk <i>increment</i> dan 0x02 untuk <i>decrement</i> | Bilangan <i>signed long integer</i> yang berukuran 4 <i>bytes</i> |

Respon *tag* NFC setelah dikirim instruksi APDU untuk menulis *value block* terdiri dari 1 *byte* SW1 dan 1 *byte* SW2. Apabila SW1 bernilai 0x90 dan SW2 bernilai 0x00, penulisan *value block* berhasil. Namun apabila SW1 bernilai 0x63 dan SW2 bernilai 0x00, penulisan *value block* gagal.

Lampiran B. Diagram Komunikasi SISMIC

B.1 Transaksi *Top-Up* Saldo Melalui ATM

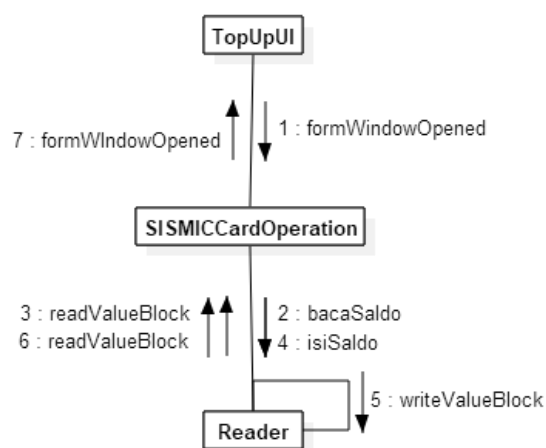
Gambar B-1 berikut ini adalah diagram komunikasi untuk transaksi *top-up* saldo pada SISMIC melalui mesin ATM.



Gambar B-1 Diagram Komunikasi Transaksi *Top-Up* Saldo Melalui ATM

B.2 Transaksi *Top-Up* Saldo Melalui *Merchant*

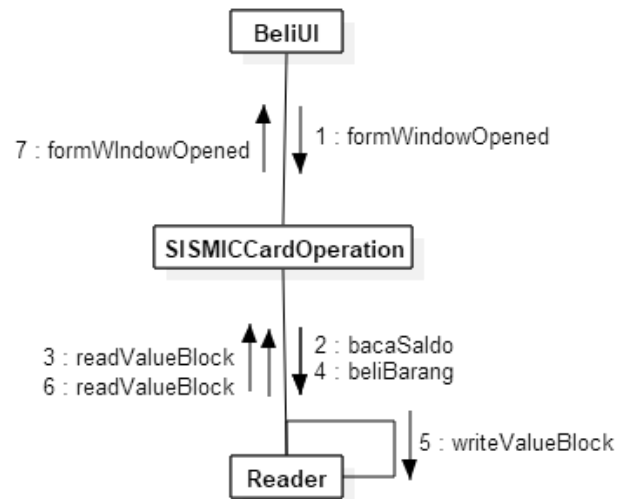
Gambar B-2 berikut ini adalah diagram komunikasi untuk transaksi *top-up* saldo pada SISMIC melalui *merchant*.



Gambar B-2 Diagram Komunikasi Transaksi *Top-Up* Saldo Melalui *Merchant*

B.3 Transaksi Pembelian Melalui *Merchant*

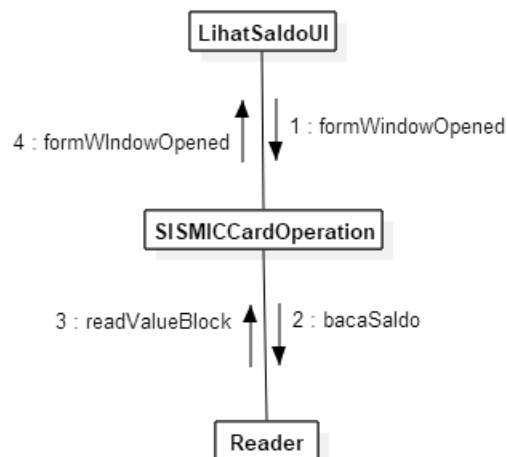
Gambar B-3 berikut ini adalah diagram komunikasi untuk transaksi pembelian melalui *merchant*.



Gambar B-3 Diagram Komunikasi Transaksi Pembelian Saldo Melalui *Merchant*

B.4 Lihat Saldo Melalui ATM

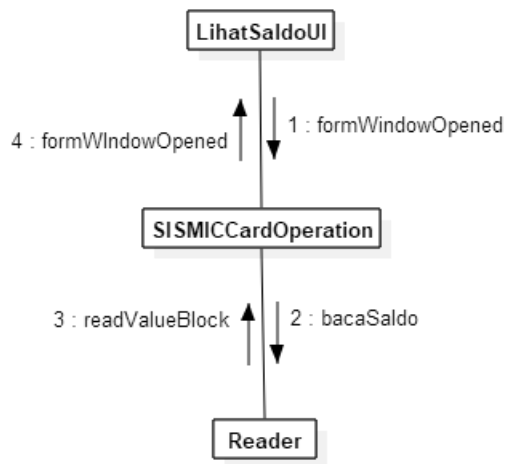
Gambar B-4 berikut ini adalah diagram komunikasi untuk melihat saldo kartu SISMIC melalui mesin ATM.



Gambar B-4 Diagram Komunikasi Lihat Saldo Melalui ATM

B.5 Lihat Saldo Melalui Merchant

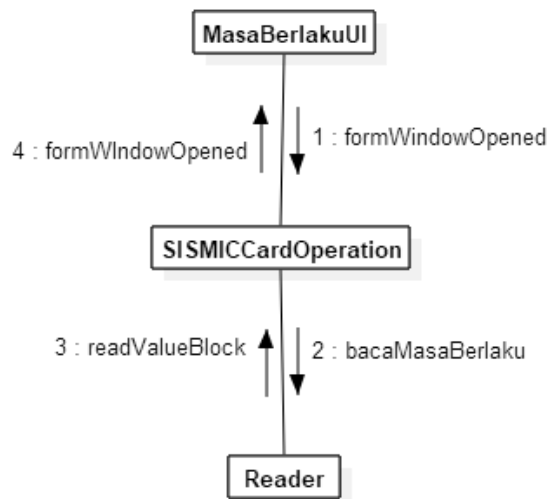
Gambar B-5 berikut ini adalah diagram komunikasi untuk melihat saldo kartu SISMIC melalui *merchant*.



Gambar B-5 Diagram Komunikasi Lihat Saldo Melalui *Merchant*

B.6 Lihat Masa Berlaku Kartu SISMIC Melalui ATM

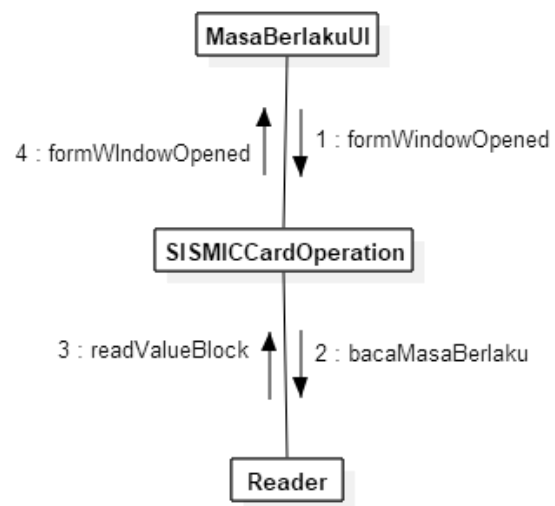
Gambar B-6 berikut ini adalah diagram komunikasi untuk melihat masa berlaku kartu SISMIC melalui mesin ATM.



Gambar B-6 Diagram Komunikasi Masa Berlaku Kartu Melalui ATM

B.7 Lihat Masa Berlaku Kartu SISMIC Melalui *Merchant*

Gambar B-7 berikut ini adalah diagram komunikasi untuk melihat masa berlaku kartu SISMIC melalui *merchant*.



Gambar B-7 Diagram Komunikasi Masa Berlaku Kartu Melalui *Merchant*