

**SISTEM PENDAFTARAN, REKAM MEDIS DAN  
SISTEM PEMBAYARAN PASIEN RUMAH SAKIT  
BERBASIS TEKNOLOGI JAVA CARD**

**TUGAS AKHIR**

oleh

**DWI KURNIA PUTRA**

**06 06 04 249 4**



**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
GENAP 2007/2008**

## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Saya menyatakan dengan sesungguhnya bahwa tugas akhir dengan judul:

### **SISTEM PENDAFTARAN, REKAM MEDIS DAN SISTEM PEMBAYARAN PASIEN RUMAH SAKIT BERBASIS TEKNOLOGI JAVA CARD**

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Pendidikan Sarjana Teknik Ekstensi Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari tugas akhir yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 7 Juli 2008

Dwi Kurnia Putra

NPM 06 06 04 249 4

# PENGESAHAN

Tugas Akhir dengan judul :

**SISTEM PENDAFTARAN, REKAM MEDIS  
DAN SISTEM PEMBAYARAN PASIEN RUMAH SAKIT  
BERBASIS TEKNOLOGI JAVA CARD**

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Pendidikan Sarjana Teknik Ekstensi Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Tugas akhir ini telah diujikan pada sidang ujian tugas akhir pada tanggal 7 Juli 2008 dan dinyatakan memenuhi syarat/sah sebagai tugas akhir pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, 14 Juli 2008  
Dosen Pembimbing

## UCAPAN TERIMA KASIH

Puji dan syukur kehadiran ALLAH SWT atas berkat dan rahmat-Nya sehingga tugas akhir ini dapat diselesaikan dengan baik. Tak lupa penulis juga mengucapkan terima kasih kepada:

**Prof. Dr. Ir. Harry Sudibyo S, DEA**

**F. Astha Ekadiyanto ST, MSc**

selaku dosen pembimbing dan penasehat yang telah meluangkan waktu untuk memberi pengarahan, diskusi dan bimbingan, serta persetujuan sehingga tugas akhir ini dapat selesai dengan baik. Terima kasih pula kepada kedua orang tua dan seluruh anggota keluarga atas dukungan yang telah diberikan. Tidak lupa terima kasih kepada semua rekan-rekan yang tidak dapat disebutkan satu per satu.

Prof. Dr. Ir. Harry Sudibyo S, DEA

NIP 130 891 668

Dwi Kurnia Putra  
NPM 06 06 04 249 4  
Departemen Teknik Elektro

Dosen Pembimbing  
Prof. Dr. Ir. Harry Sudibyo S, DEA

**SISTEM PENDAFTARAN, REKAM MEDIS  
DAN SISTEM PEMBAYARAN PASIEN RUMAH SAKIT  
BERBASIS TEKNOLOGI JAVA CARD**

**ABSTRAK**

Informasi rekam medis seseorang merupakan salah satu faktor yang menentukan kualitas pelayanan yang diberikan oleh pusat pelayanan kesehatan kepada pasiennya, oleh sebab itu informasi rekam medis ini harus selalu ada ketika dibutuhkan [PerMen89].

*Smart card* merupakan suatu teknologi kartu yang di dalamnya terdapat sebuah *chip* komputer. *Smart card* dapat diprogram untuk menciptakan berbagai macam aplikasi dan sebagai tempat penyimpanan data. Salah satu jenis *smart card* adalah *java card* yang merupakan pemrograman berbasis *java*.

Tugas akhir ini bertujuan untuk memberikan gambaran umum tentang rancang bangun untuk mengimplementasikan suatu aplikasi *java card* yang terintegrasi dengan sistem *database* rumah sakit sehingga dapat dikembangkan menjadi aplikasi yang diterapkan menjadi layanan kartu akses *smart card* kesehatan.

Program aplikasi *java card* yang terintegrasi dengan sistem *database* rumah sakit ini terdiri dari empat aplikasi *host* yaitu aplikasi rumah sakit sebagai *administrator* dan *supervisor*, aplikasi dokter, aplikasi pasien, dan aplikasi bagian keuangan serta *applet* pada kartu. Program aplikasi ini dimodifikasi dan disimulasikan menggunakan simulator *Java Card Workstation Development Environment* (JCWDE) untuk dilakukan analisa dan pengujian.

Hasil dari tugas akhir ini berupa rancangan suatu sistem kartu akses *smart card* kesehatan yang dapat memenuhi peraturan kesehatan Indonesia, menjamin keamanan dan kerahasiaan data, mempercepat dan meningkatkan pelayanan kesehatan, serta dapat digunakan oleh berbagai perangkat lunak aplikasi *smart card* kesehatan (*interoperability*).

Tugas akhir ini berhasil mensimulasikan suatu *applet java card* dengan platform *java card* yang berinteraksi dengan suatu program *host* dengan platform *Microsoft Windows XP SP 2*.

**Kata kunci : Administrator, Applet, Chip, Database, Host, Interoperability, Java Card, Smart Card, Supervisor.**

Dwi Kurnia Putra  
NPM 06 06 04 249 4  
Departemen Teknik Elektro

Counsellor  
Prof. Dr. Ir. Harry Sudibyo S, DEA

**REGISTRY SYSTEM, MEDICAL RECORD  
AND FINANCIAL SYSTEM OF PATIENT ON HOSPITAL  
BASED JAVA CARD TECHNOLOGY**

**ABSTRACT**

Information about a patient's medical history is one of the main factor that determine the quality of service of a health service center, such that information must be available any time needed [ministry degree no. 89].

Smart card is a technology card that incorporate a computer chip in a card. Smart card can be programmed to produce many applications and can be used to store the data. One kind of smart card is java card which is based on java technology.

This final project has the purpose to give general explanation about the design, contruction and implementation of an application in java card which is integrated with the hospital's database system for accessing health services using medical smart card .

The application program which is integrated with the hospital's database system consists of four host application programs host which consist of function such as administrator and supervisor, doctor application, patient application, financial application and applet in card, which is modified by this application program and simulated using Java Card Workstation Development Environment (JCWDE) for analysis and testing.

The result of this final project is to design an access card for smart card health system which can comply with health regulation in Indonesia, assuring security and confidentiality of data, and can be speed up and improve health services, and can be used to any software application for health smart card (interoperability).

This final project successfully simulate a java card applet on a java card platform which can interact with a host program running on Microsoft Windows XP SP 2 platform.

**Keywords : Applet, Chip, Database, Host, Interoperability, Java Card, Smart Card.**

# DAFTAR ISI

	Halaman
<b>JUDUL</b>	I
<b>PERNYATAAN KEASLIAN TUGAS AKHIR</b>	ii
<b>PENGESAHAN</b>	iii
<b>UCAPAN TERIMA KASIH</b>	iv
<b>ABSTRAK</b>	v
<b>ABSTRACT</b>	vi
<b>DAFTAR ISI</b>	vii
<b>DAFTAR GAMBAR</b>	ix
<b>DAFTAR TABEL</b>	xii
<b>DAFTAR LAMPIRAN</b>	xiii
<b>DAFTAR SINGKATAN</b>	xiv
<b>BAB 1 PENDAHULUAN</b>	1
1.1 Latar Belakang Masalah	1
1.2 Tujuan Penulisan	3
1.3 Ruang Lingkup	3
1.4 Pembatasan Masalah	3
1.5 Metode Perancangan	4
1.6 Sistematika Penulisan	5
<b>BAB 2 LANDASAN TEORI</b>	6
2.1 Rekam Medis	6
2.1.1 Kegunaan Rekam Medis	7
2.1.2 Nilai Informasi Rekam Medis	8
2.1.3 Kepemilikan Rekam Medis	8
2.1.4 Dasar Hukum Rekam Medis di Indonesia	9
2.1.5 Lama Penyimpanan Rekam Medis	10
2.2 <i>Smart Card</i>	10
2.2.1 Komunikasi antara <i>Smart Card</i> dan Aplikasi	11
2.2.2 Jenis Memori pada <i>Smart Card</i>	13
2.3 Teknologi <i>Java Card</i>	14

2.3.1 Elemen-Element dari Aplikasi <i>Java Card</i>	15
2.3.2 Operasi JCRE selama Sesi CAD ( <i>Card Acceptance Device</i> )	17
2.3.3 Komunikasi <i>Applet Java Card</i>	18
2.4 Komunikasi <i>Smart Card</i>	19
<b>BAB 3 PERANCANGAN SISTEM PENDAFTARAN, REKAM MEDIS DAN SISTEM PEMBAYARAN PASIEN BERBASIS TEKNOLOGI JAVA CARD</b>	23
3.1 <i>Use Case</i> Diagram Aplikasi Kartu akses Pasien Rumah Sakit	23
3.2 Paket dan Kelas yang digunakan	29
3.3 Langkah-langkah Pembuatan	30
3.4 Aliran Diagram Aplikasi Kartu Akses Pasien Rumah Sakit	32
3.5 Sekuensial Diagram Aplikasi Kartu Akses Pasien	33
3.6 Konfigurasi Sistem Rumah Sakit	44
<b>BAB 4 PENGUJIAN DAN ANALISIS</b>	46
4.1 Simulator Kartu pada CAD	46
4.2 Pengujian Program	47
4.3 Penilaian Pengguna Terhadap Aplikasi	53
4.3.1 Penilaian Pengguna Terhadap Aplikasi Rumah Sakit	54
4.3.2 Penilaian Pengguna Terhadap Aplikasi Pasien	56
4.3.3 Penilaian Pengguna Terhadap Aplikasi Dokter	58
4.3.4 Penilaian Pengguna Terhadap Aplikasi Bagian Keuangan	60
4.4 Analisa Hasil Pengujian Program	62
<b>BAB 5 KESIMPULAN</b>	64
<b>DAFTAR ACUAN</b>	65
<b>DAFTAR PUSTAKA</b>	67
<b>LAMPIRAN</b>	68



## DAFTAR GAMBAR

	Halaman
<b>Gambar 2.1</b>	Bentuk fisik smart card Contact 11
<b>Gambar 2.2</b>	Bentuk fisik smart card Contactless 12
<b>Gambar 2.3</b>	Delapan titik kontak dengan dunia luar 12
<b>Gambar 2.4</b>	Java Card Virtual machine (JCVM) 14
<b>Gambar 2.5</b>	Arsitektur sistem on-card 15
<b>Gambar 2.6</b>	Elemen-elemen aplikasi Java Card 16
<b>Gambar 2.7</b>	Gambar smart card dan reader 17
<b>Gambar 2.8</b>	Komunikasi I/O APDU 17
<b>Gambar 2.9</b>	Proses pengembangan applet 18
<b>Gambar 2.10</b>	Transmisi half duplex 19
<b>Gambar 2.11</b>	Transfer data awal antara Smart Card dan terminal 20
<b>Gambar 2.12</b>	Struktur sebuah karakter untuk transmisi data 20
<b>Gambar 2.13</b>	Model Komunikasi Smart Card 21
<b>Gambar 3.1</b>	Use case Diagram Aplikasi Pengguna (Pasien) 25
<b>Gambar 3.2</b>	Use case Diagram Aplikasi Pengguna (Dokter) 26
<b>Gambar 3.3</b>	Use Case Diagram Aplikasi Supervisor 27
<b>Gambar 3.4</b>	<i>Use Case</i> Diagram Aplikasi Administrator 28
<b>Gambar 3.5</b>	<i>Use Case</i> Diagram Aplikasi Bagian Keuangan 29
<b>Gambar 3.6</b>	<i>Class</i> Diagram Kartu Akses Pasien Rumah Sakit 31
<b>Gambar 3.7</b>	Bagian-bagian aplikasi kartu akses pasien rumah sakit 32
<b>Gambar 3.8</b>	Aliran diagram aplikasi kartu akses pasien rumah sakit 32
<b>Gambar 3.9</b>	Diagram Sekuensial <i>Login</i> 34
<b>Gambar 3.10</b>	Diagram Sekuensial Pilih Dokter 35
<b>Gambar 3.11</b>	Diagram Sekuensial Lihat Rekam Medis Pasien (dapat dilakukan oleh dokter yang merawat dan pasien) 36
<b>Gambar 3.12</b>	Diagram Sekuensial Ganti PIN 37
<b>Gambar 3.13</b>	Diagram Sekuensial Selesai 38
<b>Gambar 3.14</b>	Diagram Sekuensial Tambah Rekam Medis Pasien oleh Dokter 39
<b>Gambar 3.15</b>	Sekuensial Diagram Program Aplikasi Supervisor 41

<b>Gambar 3.16</b>	Sekuensial Diagram Program Aplikasi Administrator	42
<b>Gambar 3.17</b>	Sekuensial Diagram Program Aplikasi Bagian Keuangan	43
<b>Gambar 3.18</b>	Konfigurasi Sistem Smart Card Kesehatan	44
<b>Gambar 4.1</b>	C-languge Java Card Runtime Environment	47
<b>Gambar 4.2</b>	Diagram alir pengujian aplikasi administrator	48
<b>Gambar 4.3</b>	Diagram alir pengujian aplikasi <i>supervisor</i>	48
<b>Gambar 4.4</b>	Diagram alir pengujian aplikasi pasien	49
<b>Gambar 4.5</b>	Diagram alir pengujian aplikasi dokter	50
<b>Gambar 4.6</b>	Diagram alir pengujian aplikasi bagian keuangan	50
<b>Gambar 4.7</b>	Grafik Penilaian Terhadap Aplikasi Rumah Sakit	55
<b>Gambar 4.8</b>	Grafik Penilaian Terhadap Aplikasi Pasien	57
<b>Gambar 4.9</b>	Grafik Penilaian Terhadap Aplikasi Dokter	59
<b>Gambar 4.10</b>	Grafik Penilaian Terhadap Aplikasi Bagian Keuangan	61

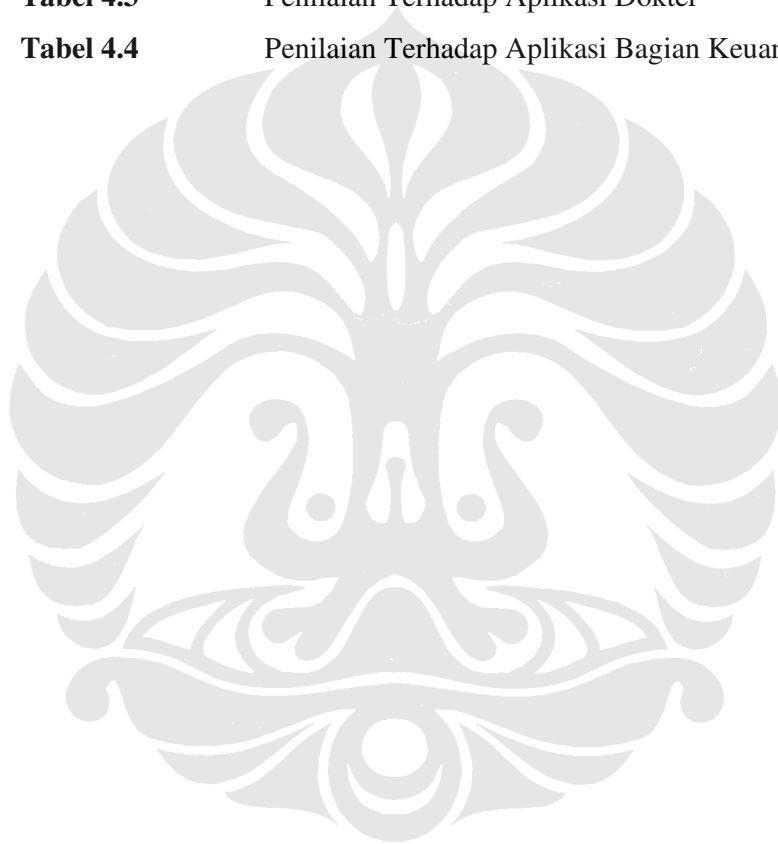
#### LAMPIRAN

<b>Gambar L1</b>	Start Aplikasi Rumah Sakit	68
<b>Gambar L2</b>	Form Aplikasi Administrator	68
<b>Gambar L3</b>	Form Konfirmasi Penyimpanan Data Umum Pasien	69
<b>Gambar L4</b>	Konfirmasi Berhasil Menyimpan Data Umum Pasien	69
<b>Gambar L5</b>	Form Aplikasi Supervisor	70
<b>Gambar L6</b>	Form Lihat Identitas Umum Pasien pada Aplikasi Pasien	70
<b>Gambar L7</b>	Mereset Kartu yang Terblokir	71
<b>Gambar L8</b>	Start Aplikasi Pasien	71
<b>Gambar L9</b>	Form Aplikasi Pasien	71
<b>Gambar L10</b>	Form Cek PIN	72
<b>Gambar L11</b>	Lihat Rekam Medis pada Aplikasi Pasien	72
<b>Gambar L12</b>	Pesan jika PIN tidak Sesuai	73
<b>Gambar L13</b>	Pesan jika PIN tidak Sesuai lebih dari 3 kali	73
<b>Gambar L14</b>	Kondisi Pasien ketika Belum Memilih Status Dokter	73
<b>Gambar L15</b>	Form Pilih Dokter	74

<b>Gambar L16</b>	Kondisi mendapat Nomor Urut Pasien Setelah Mendaftar	74
<b>Gambar L17</b>	Form Ganti Pin	75
<b>Gambar L18</b>	Ganti PIN berhasil	75
<b>Gambar L19</b>	Start Aplikasi Dokter	75
<b>Gambar L20</b>	Form Aplikasi Dokter	76
<b>Gambar L21</b>	Frame Administrator	76
<b>Gambar L22</b>	Cek PIN Pasien pada Form Aplikasi Dokter	77
<b>Gambar L23</b>	Form Lihat Rekam Medis Pasien pada Aplikasi Dokter	77
<b>Gambar L24</b>	Form Simpan Rekam Medis	78
<b>Gambar L25</b>	Form Konfirmasi Simpan Rekam Medis	79
<b>Gambar L26</b>	Simpan Rekam Medis Berhasil	79
<b>Gambar L27</b>	Konfirmasi Segera Melunasi Biaya Pengobatan setelah Berobat	80
<b>Gambar L28</b>	Start Aplikasi Bagian Keuangan	80
<b>Gambar L29</b>	Form Aplikasi Bagian Keuangan	81
<b>Gambar L30</b>	Form Transaksi Pelunasan Biaya Pengobatan	81
<b>Gambar L31</b>	Form Pelunasan Biaya Pengobatan	81

## DAFTAR TABEL

		Halaman
<b>Tabel 2.1</b>	Elemen data pada ATR	21
<b>Tabel 2.2</b>	Perintah APDUs	22
<b>Tabel 2.3</b>	Respon APDUs	22
<b>Tabel 4.1</b>	Penilaian Terhadap Aplikasi Rumah Sakit	54
<b>Tabel 4.2</b>	Penilaian Terhadap Aplikasi Pasien	56
<b>Tabel 4.3</b>	Penilaian Terhadap Aplikasi Dokter	58
<b>Tabel 4.4</b>	Penilaian Terhadap Aplikasi Bagian Keuangan	60



## DAFTAR LAMPIRAN

		Halaman
<b>Lampiran</b>	Tampilan Aplikasi Kartu Akses Smart Card Pasien Rumah Sakit	68



## DAFTAR SINGKATAN

APDU	: Application Protocol Data Unit
API	: Application Programming Interface
CAD	: Card Acceptance Device
CAP	: Converted Applet
CPU	: Central Processing Unit
EEPROM	: Electrically Erasable Programmable Read Only Memory
IEC	: International Electronics Commitee
ISO	: International Standardization Organization
JCDK	: Java Card Development Kit
JCRE	: Java Card Runtime Environment
JCRMI	: Java Card Remote Method Invocation
JCWDE	: Java Card Workstation Development Environment
JCVM	: Java Card Virtual Machine
J2SE	: Java 2 Standard Edition
OCF	: Open Card Framework
PIN	: Personal Identification Number
RAM	: Read Access Memory
RF	: Radio Frequency
RFID	: Radio Frequency IDentification
RMI	: Remote Method Invocation
SW	: Status Word
VM	: Virtual Machine

# BAB I

## PENDAHULUAN

Bab ini menjelaskan latar belakang masalah, tujuan, ruang lingkup, pembatasan masalah, metode penelitian yang dilakukan, dan sistematika penulisan tugas akhir ini.

### 1.1 LATAR BELAKANG MASALAH

Dalam era globalisasi yang terus berkembang saat ini, kehidupan manusia modern membutuhkan sistem dan informasi yang dapat diakses dengan cepat dan mudah sehingga dapat berjalan dengan efisien dari segi waktu, dan dapat menjamin keamanan dari informasi-informasi tersebut. Salah satu teknologi yang dapat diimplementasikan yaitu teknologi *smart card*.

Teknologi *smart card* memberikan pelayanan dalam hal kemudahan dan keamanan penyimpanan data karena adanya mekanisme enkripsi data sebelum data tersebut disimpan dalam memori yang terdapat pada kartu, serta adanya pin (kode rahasia) sehingga dapat dijaga kerahasiaan data tersebut. Sistem operasi pada *smart card* menyediakan bahasa/perintah yang dapat dimengerti oleh *smart card* itu sendiri, serta fleksibilitasnya yang tinggi seperti halnya bahasa pemrograman yang berorientasi objek, *package-package* yang terdapat di dalamnya dapat dimanfaatkan untuk pengembangan lebih lanjut sesuai kondisi dan keinginan.

Salah satu bahasa/perintah yang digunakan pada *smart card* adalah pemrograman *java card* yang merupakan *subset* dari bahasa pemrograman *java* yang dikembangkan oleh *Sun Microsystems* khusus untuk divais-divais dengan keterbatasan sumber daya seperti *smart card*. *Tools* pengembangan *java card* ini disebut *Java Card Development Kit (JCDK)*.

Secara umum *smart card* merupakan suatu kartu elektronik yang dapat menyimpan dan memproses informasi dan memiliki nilai lebih dibandingkan kartu magnetik (*magnetic card*) yaitu:

- *smart card* dapat diandalkan daripada *magnetic card* karena *smart card* tidak terpengaruh oleh medan listrik dan magnet dari luar,
- lebih aman karena dilengkapi dengan sistem keamanan yang baik yaitu dengan penggunaan pin dan enkripsi data,
- dapat dijamin kerahasiaan data pemegang *smart card*.

Teknologi *smart card* dengan pemrograman berbasis *java card* mendorong penulis untuk menggunakan teknologi ini. *Sun microsystems* menyertakan beberapa contoh aplikasi yang cukup bermanfaat yang diterapkan dalam kehidupan sehari-hari, seperti aplikasi kartu pembayaran, kartu anggota, kartu akses, dan sebagainya.

Salah satu penerapan teknologi *smart card* dengan pemrograman *java card* yang akan dikembangkan oleh penulis adalah kartu akses *smart card* kesehatan yang dirancang untuk dapat menyimpan id dan pin pasien sebagai tanda pengenal pasien rumah sakit. Dengan adanya kartu akses *smart card* kesehatan tersebut dapat mempercepat penyaluran informasi dan penyimpanan data yang terorganisir dalam *database* rumah sakit. Selain itu, dokter dapat melihat riwayat rekam medis pasien secara langsung ketika pasien membawa kartu akses *smart card* sehingga dapat dengan cepat diperoleh informasi rekam medis pasien sehingga memudahkan dokter dalam melakukan tindak lanjut terhadap pengobatan sebelumnya maupun memberikan diagnosa untuk pengobatan yang baru. Dokter berwenang untuk menuliskan data rekam medis pasien ke *database* rumah sakit.

Di lain pihak, pasien dapat melakukan pendaftaran secara *on line* sehingga pasien dapat menentukan dokter yang akan merawatnya dengan melakukan pendaftaran berobat pada mesin pendaftaran pasien. Pasien dapat dengan mudah membawa dan menggunakan kartu akses *smart card* tersebut saat berobat ke rumah sakit yang telah dilengkapi sistem pembacaan (*card reader*) dan perangkat lunak aplikasi *smart card* kesehatan (*interoperability*).



## 1.2 TUJUAN PENULISAN

Tujuan tugas akhir ini adalah sebagai berikut.

1. Mengimplementasikan teknologi *java* yang berfungsi sebagai bagian integrasi dari *database* pasien rumah sakit sebagai sarana penyimpanan catatan rekam medis dalam penerapan aplikasi kartu akses *smart card*. Aplikasi ini akan dioperasikan dengan menggunakan *Java Card Reference Implementation, C-JCRE*.
2. Memahami arsitektur *java card* dan mengerti bagaimana membuat suatu aplikasi dalam *java card*.
3. Melakukan evaluasi terhadap kinerja aplikasi kartu akses *smart card* kesehatan yang dibuat.
4. Menerapkan sistem pendaftaran pasien secara *on line* dan interaktif.
5. Menerapkan sistem pembayaran tunai dari pengobatan pasien.

## 1.3 RUANG LINGKUP

Ruang Lingkup Tugas Akhir ini adalah sebagai berikut.

1. Mempelajari proses pengembangan aplikasi kartu akses *smart card* berbasis teknologi *java card*.
2. Mempelajari dan menganalisa *applet java card* yang berfungsi sebagai kartu akses pasien rumah sakit.
3. Memodifikasi, menguji, dan menjalankan *applet java card* tersebut menggunakan *simulator Java Card Workstation Development Environment (JCWDE)*.

## 1.4 PEMBATASAN MASALAH

Pembatasan masalah pada tugas akhir ini adalah sebagai berikut.

1. Tugas akhir ini merupakan analisa suatu program untuk aplikasi kartu akses pasien sebagai bagian integrasi dari *database* rekam medis pasien. Prinsip-prinsip pengembangan aplikasi kartu akses *Smart Card* berbasis *Java Card* berupa proses pengembangannya, *use case diagram*, *class*

*diagram, sequence*, modifikasi program, analisa program, pengujian, dan demo program.

2. Pembahasan mencakup aplikasi *host* (aplikasi rumah sakit yaitu *administrator* dan *supervisor*, aplikasi dokter, aplikasi pasien, dan aplikasi bagian keuangan) dan *applet* pada kartu akses pasien, sementara aplikasi *back end* tidak dibahas. *Applet* pada kartu akses pasien berisi identifikasi pemilik kartu akses *smart card* atau data personal yang meliputi id dan pin.
3. Aplikasi yang telah dimodifikasi hanya sampai *off-card installation*, dan aplikasi tidak disimpan ke dalam kartu, tetapi dijalankan dengan *simulator Java Card Workstation Development Environment (JCWDE)*.
4. Sistem *database* rumah sakit yang diterapkan merupakan sistem *database* sederhana. Rekam medis yang dicatat berupa rekam medis untuk pasien rawat jalan.

## 1.5 METODE PERANCANGAN

Perancangan dimulai dengan melakukan penelitian terhadap sistem rekam medis di rumah sakit, dan mempelajari peraturan-peraturan pemerintah tentang rekam medis di Indonesia, serta mempelajari tentang penggunaan JCDK sebagai alat untuk mengembangkan aplikasi kartu akses *smart card* berbasis teknologi *java card* yang membutuhkan beberapa perangkat lunak untuk dapat mengembangkan aplikasi *java card* dan penggunaannya.

Kemudian perancangan dilanjutkan dengan mempelajari prinsip-prinsip dasar pengembangan aplikasi kartu akses *smart card* dengan menggunakan teknologi *java card*. Selanjutnya mempelajari dan memahami aplikasi yang telah dikembangkan dalam program-program yang ada untuk dianalisa, dimodifikasi dan diuji.

## 1.6 SISTEMATIKA PENULISAN

Sistematika penulisan tugas akhir ini adalah sebagai berikut :

**Bab I;** Pendahuluan; membahas latar belakang masalah, tujuan penelitian, ruang lingkup dan pembatasan masalah, metode penelitian, dan sistematika penulisan tugas akhir ini.

**Bab II;** Landasan Teori; membahas landasan-landasan teori yang meliputi rekam medis rumah sakit, landasan teori dalam perancangan aplikasi kartu akses *smart card* berbasis teknologi *java card* yang memenuhi spesifikasi kebutuhan rekam medis di Indonesia. Pertama-tama bab ini membahas tentang rekam medis, setelah itu bab ini membahas tentang *smart card* yang berisi penjelasan tentang prinsip kerja *smart card*. Kemudian dilanjutkan dengan teknologi *Java Card*, yang berisi tentang elemen-elemen teknologi *java card* dan bagaimana komunikasi antara aplikasi *host* dengan *applet java card*.

**Bab III;** perancangan sistem pendaftaran, rekam medis dan sistem pembayaran pengobatan pasien rumah sakit dengan menggunakan kartu akses *smart card* pasien rumah sakit; membahas tentang pengembangan aplikasi, sistem yang akan diimplementasikan, *use case diagram*, *class diagram*, *sequence* dari program aplikasi kartu akses *smart card* sebagai kartu akses pasien rumah sakit.

**Bab IV;** Analisis dan Pengujian Program; menganalisa sistem yang diimplementasikan, tahap-tahap pengembangan dan ujicoba aplikasi.

**Bab V;** Kesimpulan; berisi kesimpulan hasil perancangan dan saran pengembangan di masa depan terhadap rancangan yang dihasilkan.

## BAB II

### LANDASAN TEORI

#### 2.1 REKAM MEDIS

Definisi rekam medis adalah data yang berisi keterangan riwayat medis pasien baik yang tertulis maupun terekam, dan memuat informasi yang cukup dan akurat tentang identitas pasien, anamnesis, pemeriksaan, penentuan fisik, perjalanan penyakit, laboratorium, diagnosis, segala pelayanan dan tindakan medis serta proses pengobatan yang diberikan kepada pasien, dan dokumentasi hasil pelayanan; baik yang dirawat inap, rawat jalan maupun pelayanan gawat darurat di suatu sarana pelayanan kesehatan [1].

Definisi *electronic medical record* adalah data rekam medis yang diatur oleh suatu badan tertentu seperti : rumah sakit, klinik, atau suatu jaringan komputer sehingga antar badan tersebut dapat saling beroperasi [2]. *Electronic medical record* memiliki fungsi-fungsi penting sebagai berikut :

- terdapat sistem yang dapat mengidentifikasi semua informasi pasien yang ada dalam ruang lingkup suatu badan tertentu,
- menjamin semua informasi pasien tersedia bagi para petugas kesehatan dalam ruang lingkup suatu badan tertentu; termasuk harmonisasi data, penyimpanan data, teknik data *mining*, mesin antar muka, jaringan, dsb,
- mengimplementasikan ketentuan perangkat lunak, struktur dan antar muka sistem,
- membuat keamanan sistem dengan mendefinisikan ketentuan hak (kerahasiaan, akses data rekam medis dan mengubah informasi pasien).

Keamanan sistem yang harus diimplementasikan :

- kontrol akses menggunakan *password* atau *biometric* untuk mengotentikasi dan mengklasifikasikan pengguna sesuai dengan otorisasi mereka dalam mengakses informasi dan menggunakan fungsi tertentu,

- integritas data setelah proses perbaikan, tidak boleh ada informasi yang hilang atau diubah dengan cara apapun, perbaikan dibuat berdasarkan persetujuan,
- pemeriksaan lengkap terhadap akses ke suatu data dan tambahan lain yang dibuat dalam data,
- ketersediaan sistem harus dirancang untuk tersedia setiap waktu.

### **2.1.1 Kegunaan Rekam Medis**

Kegunaan rekam medis bagi rumah sakit dan pasien [3] adalah sebagai berikut.

#### **1. Administrasi**

Isinya menyangkut kewenangan dan tanggung jawab tenaga medis dan paramedis untuk mencapai tujuan perawatan pasien. Dalam hal ini, rekam medis merupakan sumber informasi pasien yang berobat / dirawat di sebuah rumah sakit.

#### **2. Medis**

Sebagai dasar untuk merencanakan pengobatan atau perawatan pasien, termasuk untuk alat komunikasi antar dokter dan antara dokter dengan petugas kesehatan lainnya dan untuk evaluasi kualitas pelayanan rumah sakit.

#### **3. Hukum**

Melindungi kepentingan hukum bagi pasien, rumah sakit dan tenaga kesehatan.

#### **4. Keuangan**

Sebagai dasar dalam perhitungan biaya pembayaran pelayanan medis terhadap pasien.

#### **5. Penelitian**

Sebagai bahan untuk analisa, penelitian, dan dasar untuk menyusun laporan rutin rumah sakit.

#### **6. Pendidikan**

Menyediakan data untuk penelitian dan pendidikan.

#### **7. Dokumentasi**

Menjadi sumber ingatan yang harus didokumentasikan, sebagai laporan, dan dapat dipertanggungjawabkan.

### **2.1.2 Nilai Informasi Rekam Medis**

Nilai informasi yang terdapat dalam rekam medis [4] adalah sebagai berikut.

1. Informasi yang mengandung nilai kerahasiaan.

Laporan / catatan yang terdapat dalam berkas rekam medis sebagai hasil pemeriksaan, pengobatan, observasi, atau wawancara dengan pasien tidak boleh disebarluaskan kepada pihak-pihak yang tidak berwenang karena menyangkut individu langsung dari pasien. Pemberitahuan kepada pasien, keluarga pasien dilakukan oleh dokter yang merawat pasien.

2. Informasi yang tidak mengandung nilai kerahasiaan.

Identitas umum pasien seperti id, nama, jenis kelamin, alamat, dan tanggal lahir. Untuk kasus-kasus tertentu tidak boleh disebarluaskan (untuk ketenangan dan keamanan rumah sakit).

### **2.1.3 Kepemilikan Rekam Medis**

Pihak-pihak yang menjadi pemilik rekam medis [4] adalah sebagai berikut.

1. Milik rumah sakit atau tenaga kesehatan:
  - a. Sebagai penanggungjawab integritas dan kesinambungan pelayanan.
  - b. Sebagai tanda bukti rumah sakit terhadap segala upaya dalam penyembuhan pasien.
  - c. Rumah sakit memegang berkas rekam medis asli. Direktur RS bertanggungjawab atas:
    - i. Hilang, rusak, atau pemalsuan rekam medis.
    - ii. Penggunaan oleh badan atau orang yang tidak berhak.
2. Milik pasien, pasien memiliki hak legal maupun moral atas isi rekam medis. Rekam medis adalah milik pasien yang harus dijaga kerahasiaannya.
3. Milik umum, pihak ketiga boleh memiliki (asuransi, pengadilan, dsb).

Semua informasi yang terkandung dalam rekam medis adalah rahasia oleh karena itu, pemaparan isi rekam medis harus seijin pasien, kecuali:

- 1) keperluan hukum,
- 2) rujukan ke pelayanan lain untuk kepentingan pasien/keluarganya,
- 3) evaluasi pelayanan di institusi sendiri,
- 4) riset/edukasi,
- 5) kontrak badan atau organisasi pelayanan.

Berkas rekam medis sebuah rumah sakit tidak boleh dikirimkan ke tempat perawatan lain jika seandainya pasien dirujuk untuk mendapat perawatan lanjutan di institusi / rumah sakit lain. Berkas yang dikirimkan cukup resume (kesimpulan) saja atau untuk kepentingan lain oleh pasien. Semua kebijaksanaan tersebut harus dengan persetujuan dokter yang merawat pasien dan direktur rumah sakit. Kelalaian dalam pengelolaan dan pemanfaatan rekam medis dapat dikenakan sanksi oleh Dirjen Yan Medik atau oleh direktur rumah sakit yang bersangkutan [3].

#### **2.1.4 Dasar Hukum Rekam Medis di Indonesia**

Hal-hal yang mendasari hukum rekam medis di Indonesia [4] adalah sebagai berikut.

1. Peraturan pemerintah No. 10 Tahun 1966 tentang Wajib Simpan Rahasia Kedokteran.
2. Peraturan pemerintah No. 32 Tahun 1996 tentang Tenaga Kesehatan.
3. Keputusan menteri kesehatan No. 034/Birhub/1972 tentang Perencanaan dan Pemeliharaan Rumah Sakit yang mewajibkan rumah sakit untuk:
  - a. Mempunyai dan merawat statistik yang *up to date*.
  - b. Membina rekam medis yang berdasarkan ketentuan yang telah ditetapkan.
4. Peraturan menteri kesehatan No. 749a/Menkes/Per/xii/89 tentang Rekam Medis.

### 2.1.5 Lama Penyimpanan Rekam Medis

Berpedoman kepada PERMENKES tentang rekam medis tahun 1989, pasal 7 [5] dinyatakan :

- 1) lama penyimpanan rekam medis sekurang-kurangnya 5 (lima) tahun dihitung tanggal terakhir pasien berobat,
- 2) lama penyimpanan rekam medis yang berkaitan dengan hal-hal yang bersifat khusus dapat ditetapkan tersendiri.

Rekam medis yang tidak aktif dapat disimpan di ruangan lain atau dibuat mikro film.

## 2.2 SMART CARD

*Smart card* adalah kartu pintar yang di dalamnya terdapat *chip* silikon yang disebut *microcontroller*. *Chip* tersebut merupakan *integrated circuit* yang terdiri dari prosesor dan memori. *Chip* bertindak sebagai CPU (*Central Processing Unit*) yang bertugas melaksanakan perintah dan menyediakan *power* ke *smart card*. *Smart card* mempunyai kemampuan untuk memproses dan menginterpretasikan data, serta menyimpan data.

Informasi yang disimpan dalam *smart card* dapat dibawa-bawa (*portable*). Dengan teknologi *java card* kita dapat membawa informasi yang berharga dan bersifat personal di dalamnya, seperti catatan medis, nomor kartu kredit, atau kartu pembayaran elektronik.

Dengan perkembangan algoritma kriptografi, data yang disimpan dalam *smart card* akan dienkripsi terlebih dahulu, yang dilakukan untuk menambahkan autentikasi dan mengamankan akses ke sistem informasi yang mensyaratkan tingkat keamanan yang tinggi sehingga hal ini akan mempersulit pemalsuan *smart card* dan penggunaan oleh pihak yang tidak berkepentingan/berwenang.

*Smart card* hanya akan aktif bila *smart card* terhubung dengan *card reader* karena *smart card* tidak memiliki catu daya sendiri. Pada saat *card reader* membaca *smart card* dan dilakukan proses *reset*, maka *smart card* akan tetap pasif sambil menunggu *command request* dari aplikasi *client (host)*. Dalam hal



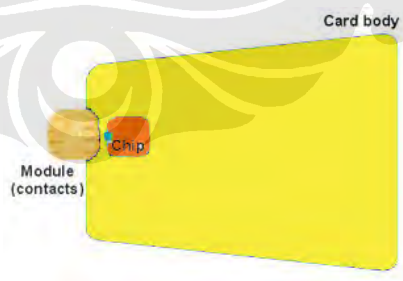
ini, *command request* diperlukan untuk dapat dilakukan proses pembacaan maupaun penulisan ke dalam *smart card*.

### 2.2.1 Komunikasi antara *Smart card* dan Aplikasi

Aplikasi pengguna berkomunikasi dengan *reader* (yang kemudian akan berkomunikasi dengan *smart card*) menggunakan protokol yang standar, yaitu protokol *International Standard Organization* (ISO) 7816. Standar internasional ISO/IEC 7816, yang diperkenalkan tahun 1983 dan terakhir diperbaharui tahun 2003, mendefenisikan berbagai aspek dari *smart card*, termasuk karakteristik fisik, kontak fisik, sinyal elektronik dan protokol-protokol transmisi, perintah-perintah, arsitektur keamanan, pengidentifikasi aplikasi, dan elemen-elemen data yang umum. *Smart card* merupakan *personal hardware* yang melakukan komunikasi dengan perangkat lainnya untuk mengakses perangkat *display* atau jaringan komputer.

*Smart card* dapat berkomunikasi dengan *reader* dengan 2 cara, yaitu :

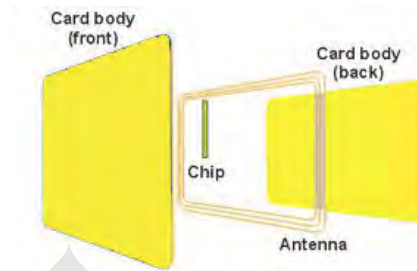
- *contact smart card* - koneksi dimulai ketika *reader* bersentuhan dengan *chip* yang ada di *smart card*. *Smart card contact* berkomunikasi dengan *card reader* menggunakan kontak secara fisik antara *card reader* dan 8 kontak pin dari *smart card*. Gambar 2.1 memperlihatkan bentuk fisik *smart card contact*.



Gambar 2.1. Bentuk fisik *smart card Contact*

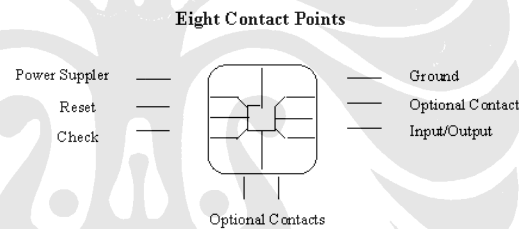
- *contact less smart card* – dapat berkomunikasi melalui antena, mengurangi keperluan untuk memasukkan dan mengambil *smart card*. Dengan *contact less*, yang harus dilakukan hanya mendekatkan *smart card* ke *reader*, dan selanjutnya *smart card* akan berkomunikasi melalui sinyal frekuensi radio

dengan jarak sekitar 60 cm berdasarkan teknologi yang menyerupai *tag* RFID (*Radio Frequency ID*) [6]. Gambar 2.2 memperlihatkan bentuk fisik *smart card contactless*.



Gambar 2.2. Bentuk fisik *smart card Contactless*

*Smart card* berinteraksi dengan dunia luar menggunakan antarmuka komunikasi serial melalui delapan titik kontak atau *contact point* yang ada seperti terlihat pada Gambar 2.3.



Gambar 2.3. Delapan titik kontak dengan dunia luar

Dalam beberapa area, penggunaan *smart card* hanya sebagai kartu memori yang hanya memberikan media penyimpanan *non-volatile* yang terproteksi. *Smart card* yang lebih canggih memiliki mikroprosesor dan memori, untuk proses dan penyimpanan yang aman, dan dapat digunakan untuk aplikasi keamanan yang menggunakan *public-key* atau *shared-key algorithm*. Memori *non-volatile* pada *smart card* merupakan sumber yang sangat berharga karena dapat dipakai untuk menyimpan kunci rahasia dan sertifikat digital.

Di sisi lain, aplikasi ini juga melakukan otentikasi terhadap pemakai. Otentikasi dilakukan terhadap *smart card* dengan cara mengetahui apakah *smart card* tersebut asli, dalam arti :

- apakah *smart card* memang berasal dari badan/perusahaan pemberi layanan aplikasi tersebut,

- setiap perusahaan menggunakan kode yang unik untuk menandai *smart card* yang dikeluarkannya,
- apakah *smart card* tersebut tidak ditiru atau digandakan. Dalam arti lain data yang disimpan di dalam *smart card* dapat ditiru jika tidak mendapat proteksi yang layak. Untuk itu pihak pemberi layanan perlu memeriksa apakah *smart card* yang dipakai benar-benar asli,
- apakah pemakai adalah pemilik *smart card* yang asli. *Smart card* bisa hilang atau mungkin dicuri sehingga bisa digunakan oleh orang yang tidak berwenang. Oleh karena itu perlu mekanisme untuk mengetahui bahwa pemegang *smart card* adalah orang berhak mengaksesnya.

Dalam rangka mengembangkan aplikasi berbasis *smart card*, diperlukan beberapa perangkat yang meliputi *smart card reader*, perangkat lunak untuk berkomunikasi dengan *reader* maupun perangkat lunak yang berkomunikasi dengan kartu dan *smart card*. *Reader* menyediakan *path* untuk aplikasi, untuk mengirim dan menerima *command* dari kartu.

### 2.2.2 Jenis Memori Pada *Smart card*

Memori yang terdapat pada *smart card* terbagi menjadi 3 bagian [ISO7816-95] yaitu:

1. ROM (*Read Only Memory*), berfungsi untuk menyimpan program utama dan bersifat permanen.
2. RAM (*Random Access Memory*), berfungsi untuk menyimpan data sementara ketika proses sedang berjalan atau hasil penghitungan selama mengeksekusi perintah. Data yang disimpan di dalamnya akan hilang begitu *smart card* dicabut (*power* hilang).
3. EEPROM (*Electrically Erasable Programmable Read Only Memory*) yang berfungsi untuk menyimpan program dan data yang sewaktu-waktu bisa diubah. Seperti halnya *hard disk* pada komputer, jenis memori ini akan tetap menyimpan data meskipun tidak ada *power* (permanen).

### 2.3 TEKNOLOGI JAVA CARD

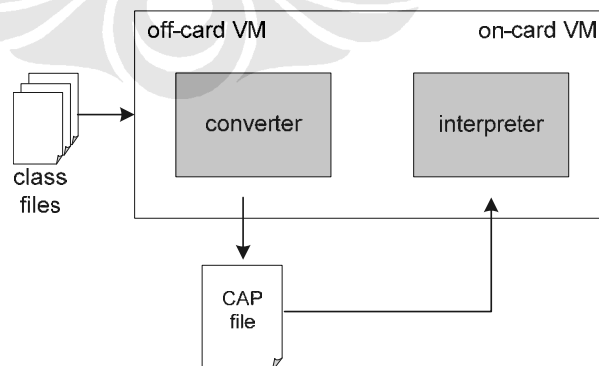
*Java Card* mendefinisikan sebuah *platform* dan aplikasinya ditulis dengan bahasa pemrograman *java*. Aplikasi yang ditulis untuk *platform Java Card* tersebut merupakan *applet* [7].

Secara umum divais *Java Card* memiliki ukuran 8-bit atau 16-bit CPU yang memiliki kecepatan 3,7Mhz, 1K RAM, dan lebih dari 16K EEPROM dan 24K ROM. Teknologi *smart card* memiliki unjuk kerja yang tinggi karena dilengkapi dengan sebuah prosesor terpisah dan *chip cryptographic* dan memori untuk enkripsi.

Spesifikasi teknologi *Java Card* yang sekarang merupakan versi 2.2, terdiri dari tiga bagian utama [8] yang meliputi:

1. *Java Card Virtual Machine (JCVM)*

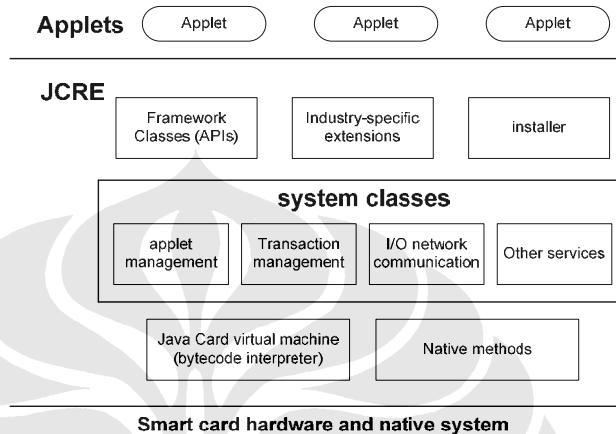
*Java Card Virtual Machine* mendefinisikan sebuah *subset* dari bahasa pemrograman *Java* dan definisi *virtual machine (VM)* untuk *smart card* [8]. *Virtual Machine* untuk *platform Java Card* diimplementasikan dalam dua bagian, dengan satu bagian di luar kartu dan satu lagi di dalam kartu. Bagian *Java VM* eksternal adalah *tool* pengembangan, biasanya disebut *tool Java Card Converter* yang mengambil, memverifikasi, dan persiapan lanjutan *class Java* pada *applet* kartu untuk eksekusi *on-card*. Keluaran dari *converter tool* ini adalah file *Converted Applet (CAP)* yang berisi semua *class* dalam sebuah paket (*package*) *Java* yang dapat di-load yang mewakili *binary* yang dapat dieksekusi.



Gambar 2.4. *Java Card Virtual machine (JCVM)*

## 2. *Java Card Runtime Environment specification (JCRE)*

Arsitektur sistem *on-card Java Card Runtime Environment (JCRE)* seperti terlihat pada Gambar 2.5. meliputi *Java Card Runtime environment (JCRE)* yang mendeskripsikan *Java Card runtime behaviour*, seperti manajemen memori, manajemen *applet*, komunikasi I/O, dan layanan lainnya [8].



Gambar 2.5. Arsitektur sistem *on-card*

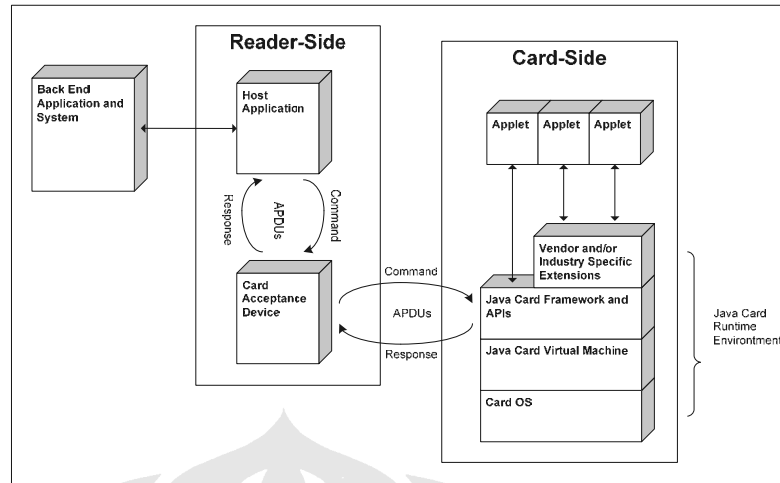
## 3. *Java Card Application Programming Interface (API)*

*Java Card Application Programming Interface (API)* mendefinisikan suatu *subset* kecil dari *Application Programming Interface (API)* dari bahasa pemrograman *Java* dan menambahkan *Java Card Framework* yang mendefinisikan *semua* sendiri dari *class* inti (*core classes*) yang secara khusus mendukung aplikasi *Java Card*. Akses *applet* ke layanan JCRE melalui kelas API [9].

*Sun Microsystems* menyediakan *Java Card Development Kit (JCDK)*, yang meliputi suatu referensi dari implementasi *Java Card RE* dan *Java Card Virtual Machine*, dan *tool* lain yang membantu pengembangan *Applet Java Card*.

### 2.3.1 Elemen-elemen dari Aplikasi Java Card

Aplikasi *java card* terdiri dari beberapa elemen yang membangun satu kesatuan fungsi dari teknologi *java card*. Elemen-elemen tersebut meliputi aplikasi *card side*, *reader side*, dan *back end* yang diperlihatkan pada Gambar 2.6.



Gambar 2.6. Elemen-elemen aplikasi Java Card

Bagian-bagian dari aplikasi *java card* terdiri dari:

1. Aplikasi *Back end* dan Sistem.  
Aplikasi *back-end* merupakan layanan yang dapat mendukung *in-card java applet*. Contoh aplikasi *back end* yaitu hubungan antara *smart card* dengan sistem keamanan.
2. Aplikasi *Host (Reader Side, off card)*.  
Aplikasi *host* merupakan terminal seperti PC, dan *notebook*. Aplikasi *host* dapat melayani komunikasi antar pengguna, *applet java card*, dan aplikasi *back end*.
3. *Card Acceptance Device (Reader-Side)*.  
*Card Acceptance Device* (CAD) adalah perangkat antarmuka (*interface device*) antara perangkat *java card* dengan aplikasi *host*. CAD (*Card Acceptance Device*) menyediakan daya ke kartu baik secara elektrik maupun dengan komunikasi RF. Sebuah CAD dapat berupa sebuah *card reader* yang dihubungkan ke komputer personal dengan menggunakan *port serial*, atau bisa saja terintegrasi dengan sebuah terminal. *Interface device* mengirimkan perintah *Application Protocol Data Unit* (APDU) dari aplikasi *host* ke kartu, dan mengirimkan respon dari kartu ke aplikasi *host*.
4. *On Card Applet dan Environment*.  
*Platform java card* merupakan *environment* yang dapat dimanfaatkan menjadi multi-aplikasi seperti aplikasi satu atau lebih *applet java card*

dapat disimpan dalam kartu, bersama dengan perangkat lunak pendukung seperti sistem operasi dari kartu dan *Java Card Runtime Environment* (JCRE). JCRE terdiri dari *Java Card VM*, *Java Card Framework* dan API, dan beberapa *extend* dari API.

### 2.3.2 Operasi JCRE selama Sesi CAD (Card Acceptance Device)

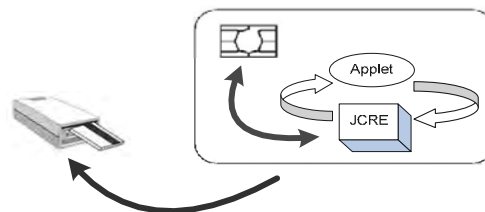
Sesi merupakan periode dari saat kartu dimasukkan ke *card acceptance device* (CAD) berupa card reader yang diberi daya listrik sampai kartu tersebut dikeluarkan lagi dari card reader.



Gambar 2.7. Gambar *smart card* dan *reader*

Pada Gambar 2.8 terlihat terjadinya proses komunikasi I/O APDU (*Application Protocol Data Unit*) yang melakukan pertukaran paket-paket data antara *applet* dan aplikasi *host*. Masing-masing APDU mengandung *command*/perintah dari *host* ke *applet* atau respon dari *applet* ke *host* [10].

Setelah JCRE reset, maka JCRE masuk ke *loop* dan menunggu APDU *command* dari *host*. *Host* mengirim APDU *command*/peintah ke *platform Java Card* dengan menggunakan *interface*/antarmuka komunikasi serial melalui *contact point input/output* dari kartu. *Applet* yang dipilih akan memproses perintah APDU. Ketika proses selesai dilakukan maka *applet* akan mengirim respon ke aplikasi *host* dan menyerahkan kontrol ke JCRE [11].

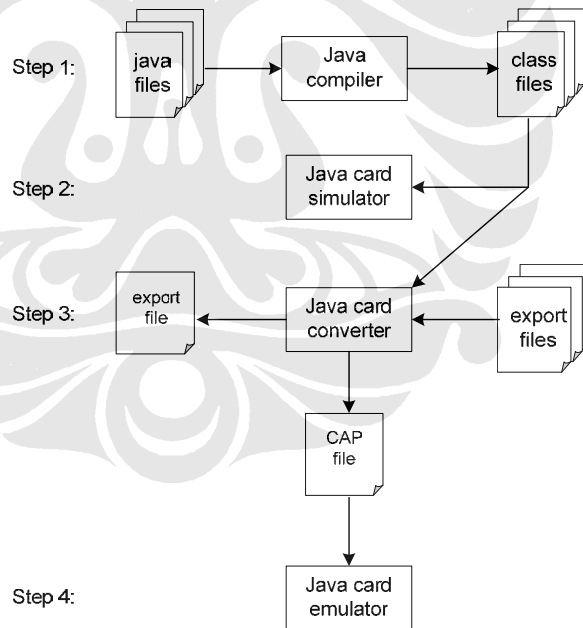


Gambar 2.8. Komunikasi I/O APDU

### 2.3.3 Komunikasi Applet Java Card

*Java Card applet* merupakan integrasi dari program *java* yang mengikuti set konvensi yang dapat dioperasikan melalui *Java Card runtime environment* dan secara dinamis dapat didownload ke dalam kartu. Sebuah *applet* harus diperluas dari *javacard.framework.applet.class*. Kelas *base applet* merupakan *superclass* untuk semua *applet* yang disimpan dalam *smart card*. Kelas *applet* merupakan *blueprint* yang mendefinisikan variabel dan metoda dari sebuah *applet*. *applet* yang berjalan dalam kartu merupakan *instance* dari sebuah *applet* sebuah objek dari kelas *applet*. Seperti objek persisten lainnya, satu kali di-*create*, sebuah *applet* hidup dalam *applet* selamanya [12].

Pada Gambar 2.9 diperlihatkan proses pengembangan *Java Card applet* yang terdiri dari satu atau beberapa kelas *java* dan kemudian di-*compile source code*-nya dengan menggunakan *java compiler* sehingga menghasilkan satu atau beberapa file kelas.



Gambar 2.9. Proses pengembangan *java card applet*

Setelah dihasilkan beberapa file kelas maka *applet* dijalankan. Dilakukan *testing* dan *debug* pada lingkungan simulasi JCWDE yang dapat mensimulasikan *Java Card runtime environment* pada sebuah PC dengan *applet* beroperasi pada



*java virtual machine*, dan file kelas dari *applet* dieksekusi. Selama proses ini dilakukan, keseluruhan aspek fungsional dari *applet* diuji.

Setelah itu file kelas dari *applet* dikonversi menjadi file CAP dengan menggunakan *Java Card converter*. Kemudian file CAP di-load dan diuji pada lingkungan emulator yang meliputi implementasi dari sebuah *Java Card virtual machine*. Perilaku dari *applet* yang dieksekusi dalam emulator sama dengan perilaku saat *applet* berjalan pada kartu yang asli. Akhirnya, ketika *applet* selesai diuji dan siap untuk di-*download* ke kartu, maka *applet* ini diwakili oleh satu atau beberapa file CAP yang dapat di-load dan di-instal ke dalam *java smart card* yang sebenarnya[13].

## 2.4 KOMUNIKASI SMART CARD

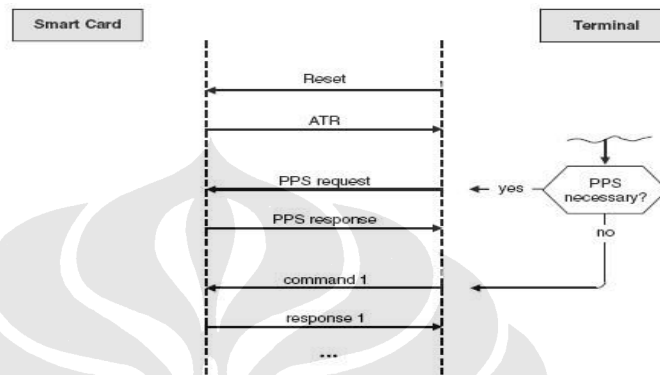
*Smart card* melakukan komunikasi dengan cara komunikasi secara bidirectional/dua arah untuk melakukan interaksi antara *smart card* dan *reader*. Koneksi antara keduanya merupakan tempat pertukaran data digital antara *smart card* dan terminalnya. Prosedur pengiriman data tersebut dinamakan transmisi *half duplex* yang memungkinkan perangkat yang satu menjadi pengirim dan perangkat lainnya menjadi penerima.

Pada Gambar 2.10, terlihat adanya komunikasi antara *smart card* dan terminal. Ketika *Smart Card* dimasukkan pada *reader* maka akan terjadi kontak secara fisik antara pin kontak *Smart Card* itu sendiri dengan pin-pin soket kartu *reader*. *Smart Card* secara otomatis akan mengaktifkan suatu *power on reset* dan mengirimkan suatu data yang dibutuhkan untuk transmisi data yaitu *Answer to Reset (ATR)* pada terminal setelah terminal tersebut meminta data *reset* kartu. Sebelum mengirimkan perintah atau *command* yang pertama pada *smart card*, terminal melakukan verifikasi terhadap ATR tersebut untuk mengecek kelayakannya[14].



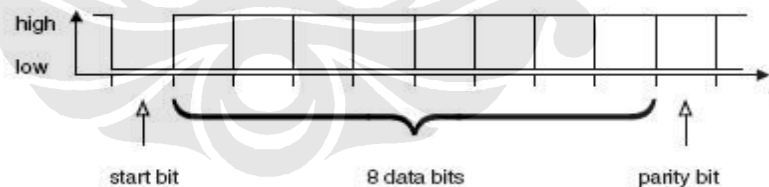
Gambar 2.10. Transmisi *half duplex*

Pada Gambar 2.11, terlihat hubungan antara ATR dan perintah pertama pada *smart card*, yang menunjukkan terminal mengirimkan suatu perintah *Protocol Parameter Selection* (PPS). Perintah ini dikeluarkan jika terminal menginginkan modifikasi pada parameter yang telah ditentukan secara *default* sebelumnya oleh ATR [13].



Gambar 2.11. Transfer data awal antara *smart card* dan terminal

Komunikasi antara *Smart Card* dan terminal adalah secara serial. Data ditangani oleh prosesor dalam bentuk urutan byte ditunjukkan pada Gambar 2.12. Tiap byte dipisahkan masing-masing kedalam delapan bit secara mandiri. Suatu *start bit* ditambahkan pada awal setiap transmisi byte untuk menandai awal urutan transmisi bagi si penerima. Pada akhir setiap byte ditambahkan *parity bit* untuk deteksi error dan satu atau dua *stop bit*.



Gambar 2.12. Struktur sebuah karakter untuk transmisi data

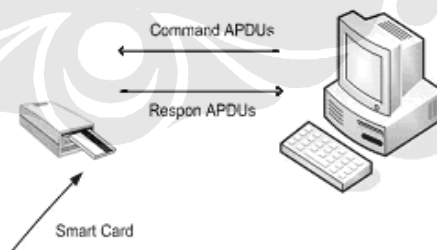
Setelah tegangan *supply*, sinyal *clock* dan sinyal *reset* tersedia maka *smart card* akan mengirim suatu *Answer to Reset* (ATR) melalui jalur data I/O. Elemen penyusun informasi ATR dapat digolongkan menjadi lima elemen yaitu *initial character*, *format character*, *interface character*, *historical character* dan *check character*. Masing-masing elemen tersebut merepresentasikan fungsi-fungsi

transmisi data tertentu pada *smart card*. Elemen data dan struktur dasar ATR diberikan pada Tabel 2.1.

Tabel 2.1. Elemen data pada ATR

Elemen Data ATR	
TS	Initial character
T0	Format character
TA1, TB1, TC1, TD1, ...	Interface character
T1, T2, ..., Tk	Historical character
TCK	Check character

Ketika dua buah komputer berkomunikasi satu sama lain, maka akan terjadi pertukaran paket-paket data, yang dibangun mengikuti satu set protokol-protokol. Demikian pula dengan *smart card* berhubungan dengan dunia luar dengan menggunakan paket datanya sendiri yang dinamakan dengan APDU (*Application Protokol Data units*). APDU mengandung perintah atau sebuah pesan respon. Sebuah *smart card* selalu menunggu sebuah perintah APDU dari terminal. *Smart card* tersebut kemudian mengeksekusi aksi-aksi yang terdapat pada APDU dan memberi balasan ke terminal dengan sebuah respon APDU. Perintah-perintah dan respon APDU bertukar secara bergantian antara sebuah kartu dengan sebuah terminal. Gambar 2.13 menunjukkan model komunikasi *smart card*.



Gambar 2.13. Model Komunikasi *Smart Card*

Komunikasi *smart card* dengan standar *protocol APDUs* didefinisikan dalam ISO 7816-4, pesan perintah dikirim dari layar aplikasi kemudian dikembalikan oleh *smart card* ke layar aplikasi. *Command APDUs* diberikan pada Tabel 2.2.

Tabel 2.2 Perintah APDUs

<b>Command APDU</b>						
Mandatory Header				Conditional Body		
CLA	INS	P1	P2	Lc	Data field	Le

**Keterangan :**

*Header* terdiri dari 4 *field*: *class* (CLA), perintah (INS) serta parameter 1 dan 2 (P1 dan P2). Masing-masing *field* berukuran 1 *byte*.

1. CLA: *class byte*. Pada beberapa *smart card*, CLA digunakan untuk mengidentifikasi aplikasi.
2. INS : *Instruction byte*. *Byte* ini menyatakan kode perintah.
3. P1 dan P2: *Parameter byte*. Menyediakan kualifikasi lebih lanjut untuk *command APDUs*.
4. *Condition body* terdiri dari 3 *field*, yaitu Lc, data *field* dan Le.
5. Lc menyatakan jumlah *byte* di dalam data *field* dari *command APDUs*.
6. *Data Field* menyatakan data yang diperlukan oleh *command APDUs*.
7. Le menyatakan jumlah maksimal dari *byte* yang diharapkan di dalam data *field* dari *respons APDUs*.

*Respons APDUs* diberikan pada Tabel 2.3.

Tabel 2.3 Respon APDUs

<b>Response APDU</b>		
Conditional Body	Mandatory Trailer	
Data field	SW1	SW2

**Keterangan:**

1. *Respons APDUs* terdiri dari *conditional body* dan *mandatory trailer*.
2. *Conditional body* berisi *data field* yang menyatakan data yang diperlukan oleh *respon APDUs*. Panjang variabel ditentukan oleh Le pada *command APDU*.
3. *Mandatory trailer* terdiri dari status *byte* SW1(status word 1) dan SW2 (status word 2) yang menyatakan status proses dari *command APDUs* di dalam kartu.

# BAB III

## PERANCANGAN SISTEM PENDAFTARAN, REKAM MEDIS DAN PEMBAYARAN PASIEN BERBASIS TEKNOLOGI JAVA CARD

Metode perancangan aplikasi *smart card* sebagai kartu akses pasien rumah sakit meliputi beberapa metode yaitu *use case* diagram dengan mendeskripsikan pihak-pihak yang terlibat dan hubungannya, sekuensial diagram yang menjelaskan sekuensial atau urutan dari program, kelas diagram dari *applet* dan aliran diagram komunikasi antara aplikasi pengguna dan *applet* pada kartu. Aplikasi *smart card* sebagai kartu akses pasien menggunakan bahasa pemrograman *Java Card* RMI, dengan menggunakan pendekatan model yang berorientasi objek. Perancangan aplikasi kartu akses pasien ini menggunakan model RMI dengan membuat aplikasi *server*, *remote objek* dan aplikasi pengguna, dan membuat *remote reference* pada *server remote objek*. Pada aplikasi kartu akses pasien rumah sakit ini, *applet* pada kartu akses pasien ini adalah *server* dan *client* adalah aplikasi pengguna yang digunakan oleh pemegang kartu akses pasien untuk melakukan akses terhadap *database* rumah sakit.

### 3.1. USE CASE DIAGRAM APLIKASI KARTU AKSES PASIEN RUMAH SAKIT

Rancang bangun sistem pendaftaran dan rekam medis pasien rumah sakit dengan menggunakan kartu akses *smart card* yang dibuat terdiri dari aplikasi rumah sakit (*administrator* dan *supervisor*), aplikasi dokter, aplikasi pasien, aplikasi bagian keuangan, dan *applet* pada kartu. Penjelasan mengenai aplikasi-aplikasi tersebut akan dijelaskan pada pembahasan bab ini.

Kartu akses pasien adalah sebuah aplikasi *smart card* yang dapat digunakan oleh pemegang kartu dalam hal ini adalah pasien. Kartu ini menyimpan data yang meliputi status kartu, id, dan PIN pasien yang digunakan sebagai *argument* untuk melakukan akses terhadap sistem *database* rumah sakit.

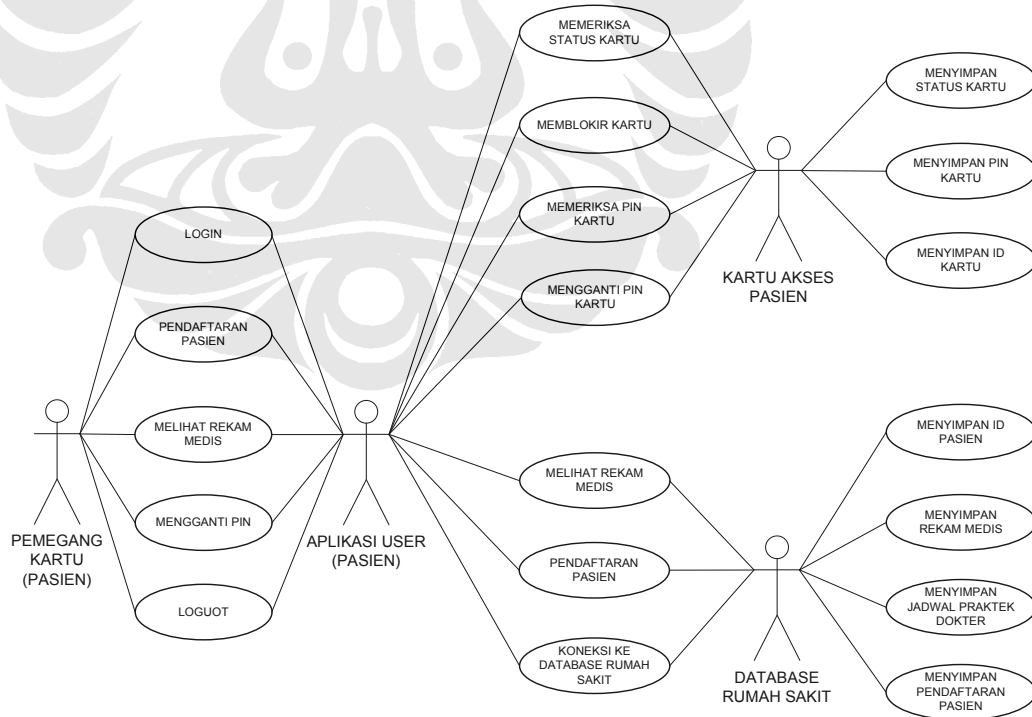
Pemegang kartu adalah pasien dalam aplikasi kartu akses pasien. Pasien yang memegang kartu memiliki akses terhadap dirinya sendiri terhadap rumah sakit sehingga dokter yang merawatnya dapat melihat riwayat rekam medis pasien yang bersangkutan sehingga dapat mempercepat dan membantu dalam memberikan pengobatan selanjutnya kepada pasien. Dalam hal ini ada kalanya rekam medis pasien terdahulu memiliki keterkaitan terhadap pengobatan selanjutnya seperti adanya hal-hal yang dianggap penting dari dokter yang merawat terdahulu seperti pasien alergi terhadap obat tertentu maka dokter yang merawat selanjutnya tidak akan memberikan obat tersebut kepada pasien dalam pengobatannya. Dokter yang merawat pasien dapat menuliskan rekam medis yang dapat dilakukan dengan menggunakan kartu akses pasien tersebut sebagai *argument* untuk perekaman data baru ke *database* rumah sakit. Hal ini terdapat dalam aplikasi dokter.

Aplikasi pasien adalah program yang digunakan oleh pemegang kartu dalam hal ini adalah pasien untuk melakukan *login* karena id pada kartu digunakan sebagai *argument* untuk melakukan koneksi terhadap *database* rumah sakit. Program dapat melakukan beberapa kegiatan seperti memeriksa status kartu, memblokir kartu jika kartu tidak sesuai dengan sistem yang telah dibuat oleh pihak rumah sakit, memeriksa PIN kartu, koneksi ke *database* rumah sakit sehingga pasien melihat rekam medis yang tersimpan dalam *database* rumah sakit, memilih dokter yang akan merawat secara *on line*, dan mengganti PIN kartu bagi pasien.

*Database* rumah sakit yang dibuat merupakan sebuah aplikasi sistem *database* rumah sakit sederhana yang meliputi beberapa *database* sebagai tempat penyimpanan data. *Database* pertama yaitu *database* umum pasien yang digunakan untuk menyimpan beberapa informasi dari data umum pasien pada saat pendaftaran yang meliputi id, pin, nama, jenis kelamin, alamat, tanggal lahir, telepon darurat 1 dan 2, asuransi, dan jenis asuransi. Untuk asuransi dan jenis asuransi diisi tanda *strip* jika tidak ada. *Database* kedua yaitu *database* dokter yang berfungsi untuk menyimpan beberapa informasi data pribadi dokter yang meliputi kode dokter, nama dokter, jenis kelamin, alamat, tanggal lahir, nomor

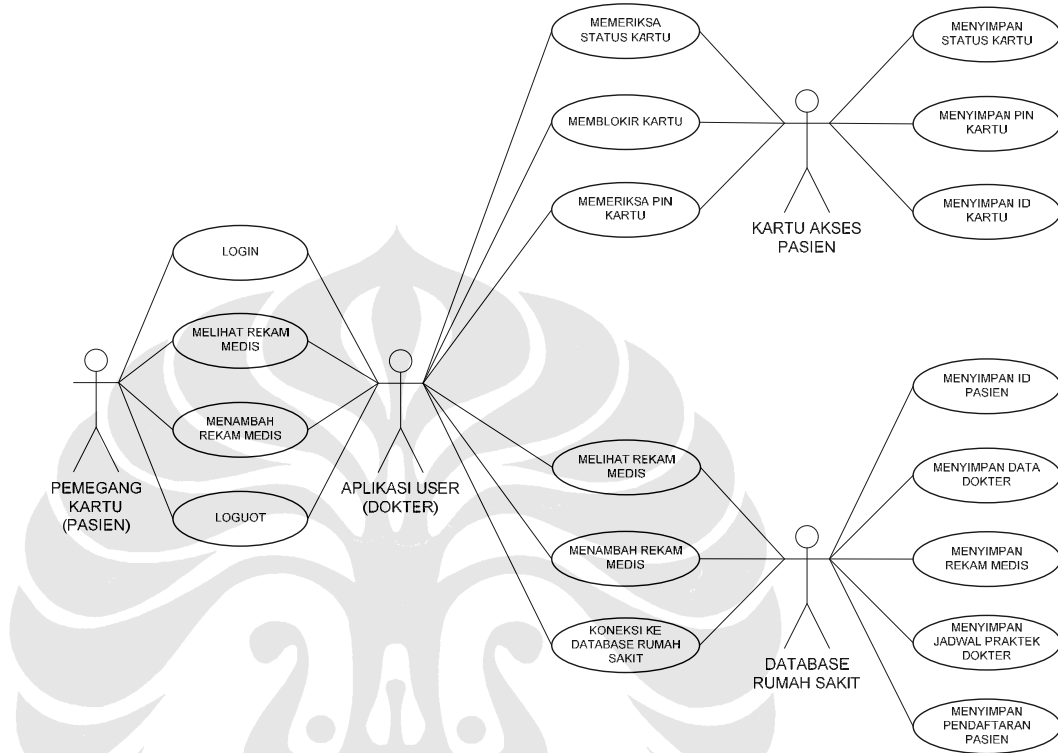
telepon dan status dokter pada rumah sakit tersebut. *Database* ketiga adalah *database* pasien rumah sakit yang berfungsi untuk menyimpan beberapa informasi rekam medis yang meliputi kode dokter, anamnesa, diagnosa penyakit, dan terapi yang diisi oleh dokter yang merawat. *Database* keempat adalah *database* jadwal praktek dokter yang berisi informasi kode, nama, status dan jadwal praktek dokter selama satu minggu yang diisi oleh seksi penjadwalan dokter rumah sakit. *Database* kelima adalah *database* pendaftaran pasien yang dibuat sebagai tempat penyimpanan data pasien yang mendaftar untuk berobat yang berfungsi untuk menyimpan hari/tanggal pemeriksaan, kode pasien, nama pasien, kode dokter, nama dokter dan status dokter. Dan *database* yang terakhir adalah *database* bagian keuangan yang merupakan sebuah aplikasi untuk proses pembayaran oleh pasien yang telah berobat.

Pemegang kartu akses pasien rumah sakit dapat melakukan proses dengan menggunakan aplikasi pengguna (pasien) untuk mengakses kartu tersebut dengan *database* rumah sakit. Hubungan pihak-pihak yang terlibat pada simulasi aplikasi kartu akses pasien ditunjukkan pada Gambar 3.1.



Gambar 3.1. Use case Diagram Aplikasi Pengguna (Pasien)

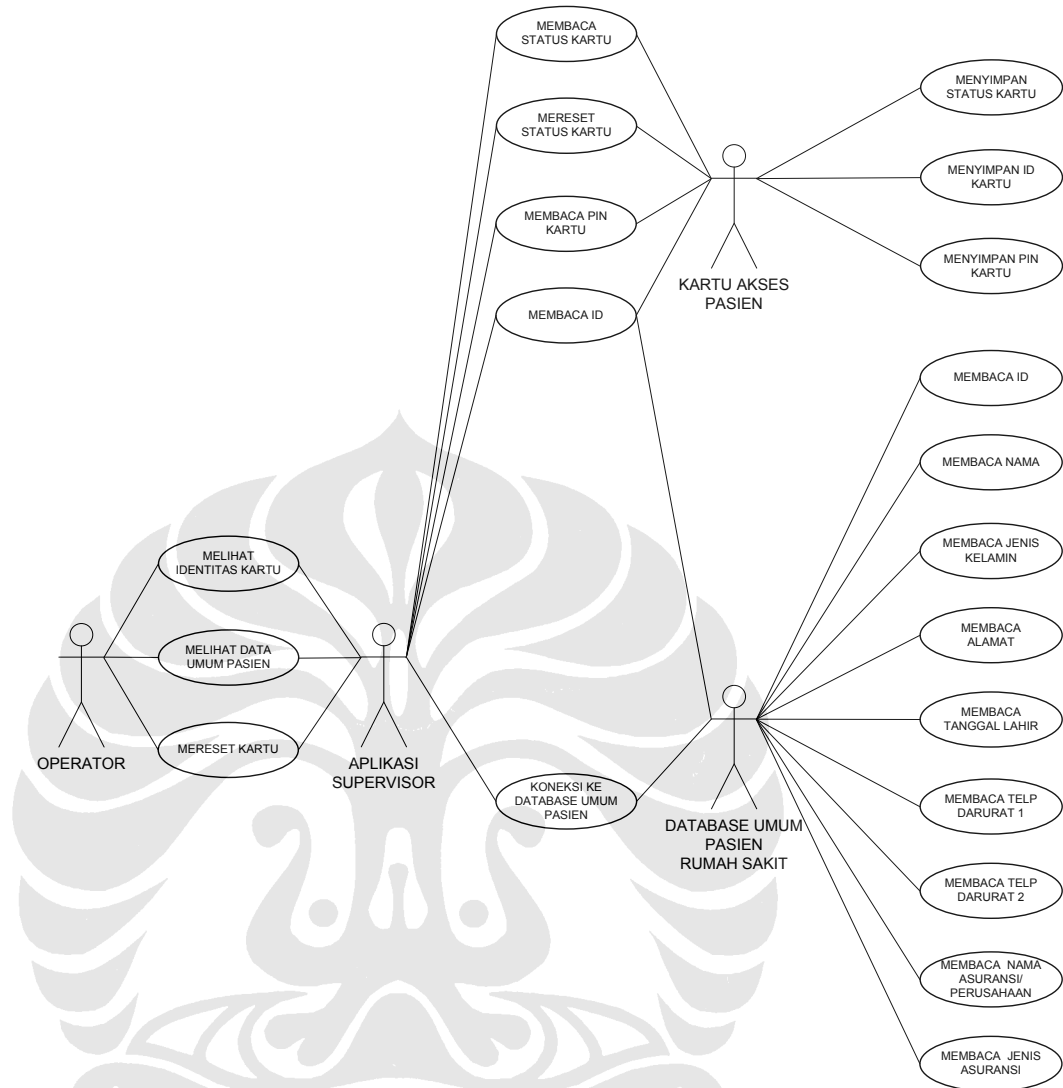
Dokter yang merawat pasien menjalankan aplikasi pengguna (dokter) untuk dapat melihat dan menyimpan rekam medis pasien. Hubungan pihak-pihak yang terlibat pada aplikasi pengguna (dokter) ditunjukkan pada Gambar 3.2.



Gambar 3.2. Use case Diagram Aplikasi Pengguna (Dokter)

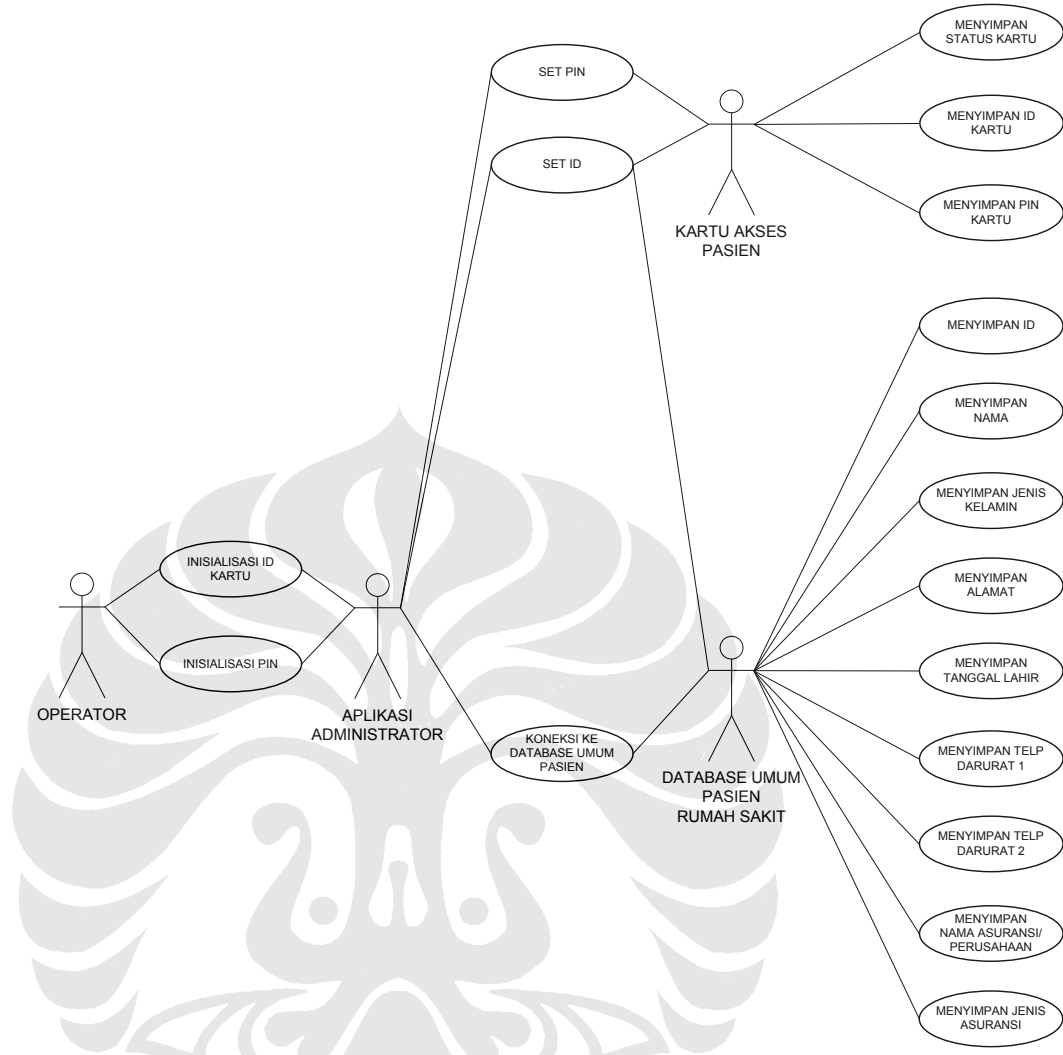
Aplikasi *supervisor* adalah program yang dijalankan oleh seorang operator atau petugas rumah sakit untuk melihat beberapa informasi pasien yaitu identitas kartu yang meliputi nomor id dan PIN kartu pasien, melihat informasi data umum pasien yang meliputi id, nama, jenis kelamin, alamat, tanggal lahir, telepon darurat 1, telepon darurat 2, nama asuransi dan jenis asuransi milik pasien, membaca status dari kartu serta dapat me-*reset* status kartu yang telah terblokir. Use case diagram dari aplikasi *supervisor* ditunjukkan pada Gambar 3.3.





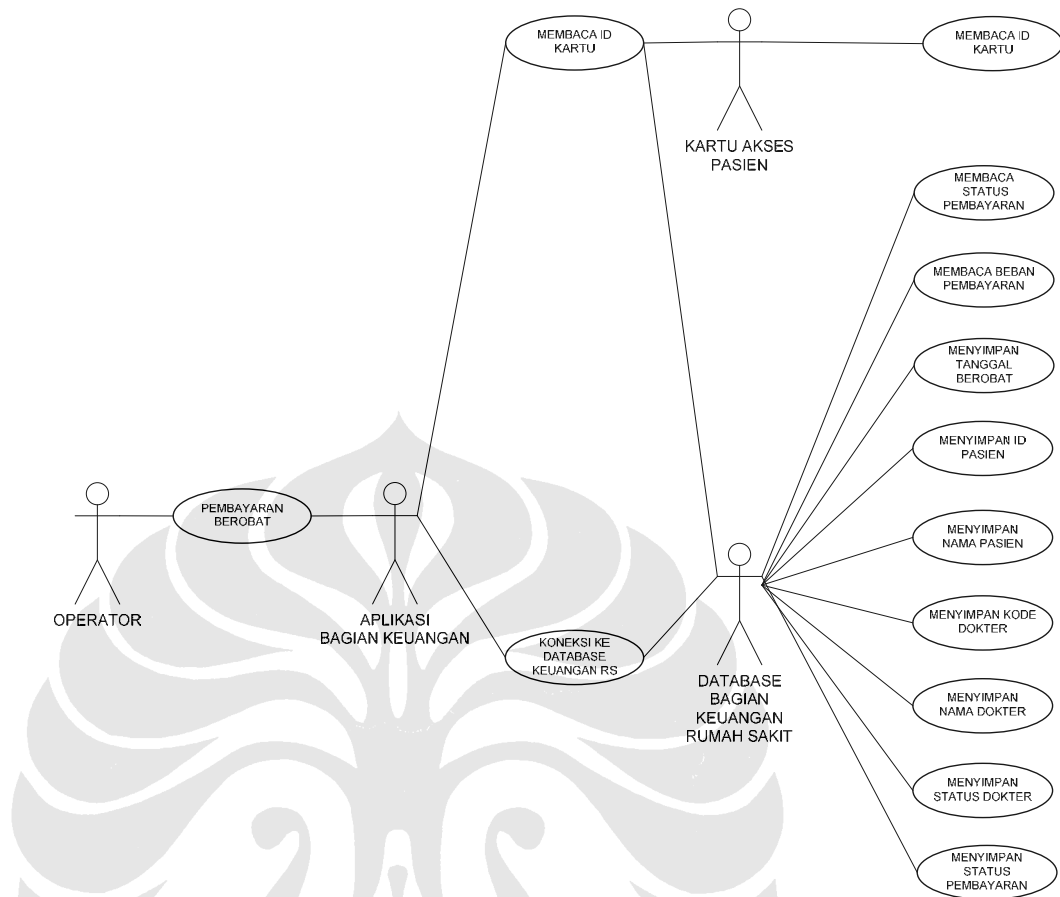
Gambar 3.3. Use Case Diagram Aplikasi Supervisor

Untuk dapat digunakan oleh pengguna kartu akses pasien, maka kartu perlu diinisialisasi terlebih dahulu. Aplikasi *administrator* adalah program yang dijalankan oleh operator atau petugas rumah sakit untuk melakukan inisialisasi sebuah kartu baru. Seorang operator akan melakukan inisialisasi terhadap id dan PIN kartu. Data yang diisikan sesuai dengan data dari pasien yang meminta aplikasi kartu akses pasien. Use case diagram aplikasi *administrator* ditunjukkan pada Gambar 3.4.



Gambar 3.4. Use Case Diagram Aplikasi Administrator

Proses pembayaran dapat dilakukan pada aplikasi pengguna bagian keuangan rumah sakit. ID pasien digunakan sebagai *argument* untuk mendapatkan beban biaya berobat pasien tersebut. Pada aplikasi ini beban biaya pengobatan dihitung berdasarkan status dokter yang merawat. Use case diagram aplikasi bagian keuangan ditunjukkan pada Gambar 3.5.



Gambar 3.5. Use Case Diagram Aplikasi Bagian Keuangan

### 3.2. PAKET DAN KELAS YANG DIGUNAKAN

Java.rmi mendefinisikan remote interface dan kelas RemoteException. javacardframework.service mendefinisikan kelas-kelas Java Card service termasuk kelas-kelas RMI service, CardRemoteObject dan RMIService.

Kelas CardRemoteObject mendefinisikan dua method untuk membuat dan memutuskan akses remote object dari luar kartu. Kelas RMIService memproses RMI request (menterjemahkan perintah APDU yang diterima menjadi panggilan ke remote method). Kelas JavaCardRMISConnect yang digunakan aplikasi pengguna untuk menginisialisasi JCRMI dan inialisasi remote reference.

`Javacard.framework.service` mendefinisikan `ServiceException`, yaitu eksepsi yang berhubungan dengan layanan *framework*.

### 3.3. LANGKAH-LANGKAH PEMBUATAN

Langkah-langkah pembuatan aplikasi kartu akses pasien :

1. Mendefinisikan *interface*, *behavior* dari kelas *remote*.
2. Membuat implementasi *server* dari kelas *remote* dan kelas-kelas yang berhubungan dengan aplikasi.
3. Membuat program aplikasi pengguna yang menggunakan *remote service* dan kelas-kelas yang berhubungan dengan aplikasi.

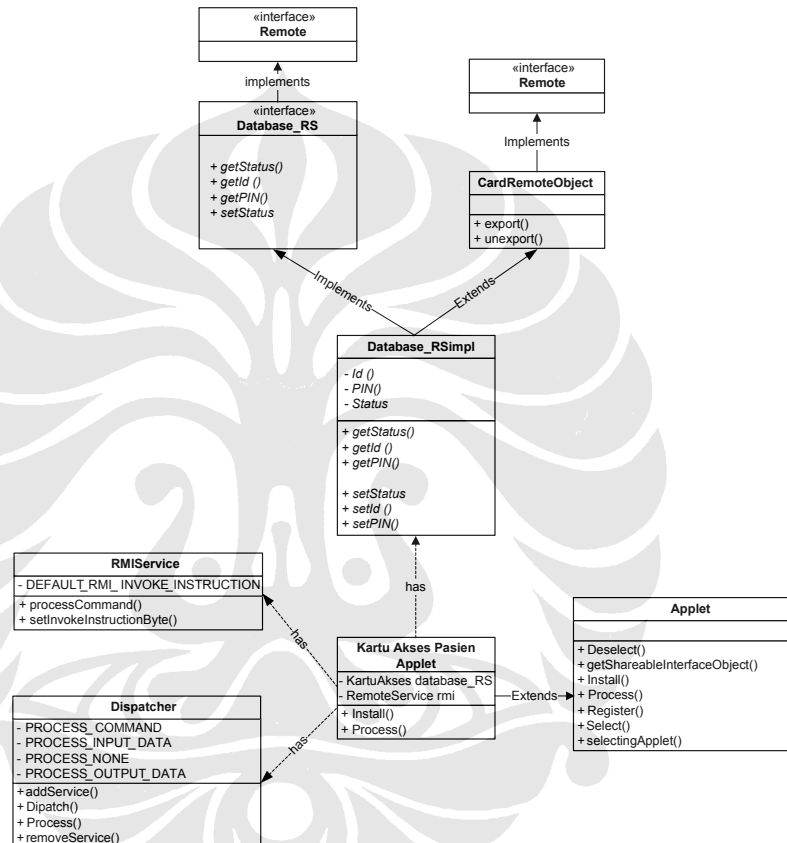
Langkah pertama yang dilakukan adalah mendefinisikan *behavior remote service*. *Remote interface* mendefinisikan layanan yang disediakan oleh *applet*. Mengikuti standar J2SE RMI, semua *interface-interface remote Java Card RMI* diturunkan dari `java.rmi.remote interface`.

Langkah kedua adalah mengimplementasi *behavior* dari *server*. Implementasi *server* pada *database\_RS applet* merupakan implementasi *remote interface* yang telah definisikan dan implementasi kelas-kelas yang berhubungan dengan aplikasi. *Database\_RS Applet* adalah *JCRMI server* yang memiliki *remote object* yang dapat digunakan pada aplikasi pengguna.

*Database\_RS applet* merupakan *object container*, seperti yang terlihat pada kelas diagram Gambar 3.6, *database\_RS applet* mempunyai *remote objek*, *APDU dispatcher* dan *RMIService* yang menerima *APDU-APDU* dan menerjemahkannya menjadi panggilan-panggilan ke *remote method*, kelas *remote* diturunkan dari kelas `CardRemoteObject`.

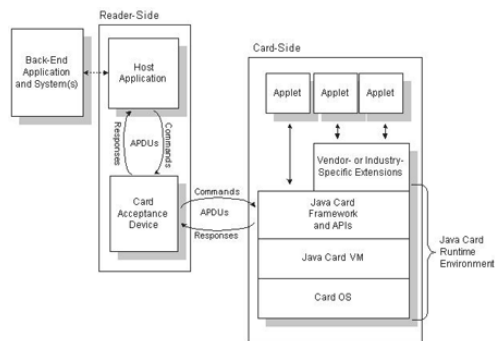
*Database\_RS Applet* diturunkan `Javacard.framework.Applet`, mengikuti struktur standar *applet*, dan mendefinisikan *life-cycle method* yang sesuai yang dapat meng-*install* dan me-*register* sendiri, serta mengirimkan *APDU-APDU*. *Method Process()* *applet* menerima perintah *APDU* dan mengirimkan pada *RMIService*, yang akan memproses perintah dengan cara

menterjemahkan menjadi RMI *call* dan *respons* secara berurutan. Mengimplementasikan JCRMI *remote object*. Implementasi diturunkan dari *CardRemoteObject*. *CardRemoteObject* mendefinisikan dua *method*, yaitu *export()* dan *unexport()*, untuk membuat hubungan atau memutuskan akses ke objek dari luar kartu, dengan menurunkan dari *CardRemoteObject*, secara otomatis dapat menggunakan semua metode dari kelas *CardRemoteObject*.



Gambar 3.6. Class Diagram Kartu Akses Pasien Rumah Sakit

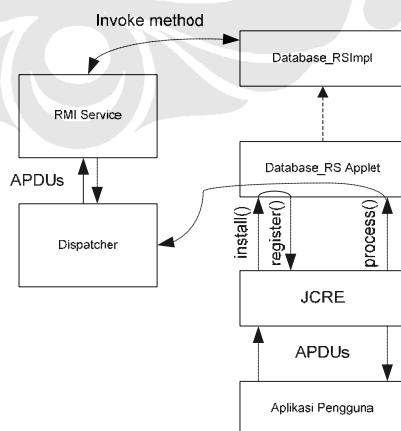
Langkah ketiga yaitu membuat program aplikasi pengguna. Untuk mengakses *smart card*, maka program aplikasi pengguna memerlukan *card-terminal* dan *service API* seperti *OpenCard Framework (OCF)*. Bagian aplikasi pengguna ditunjukkan pada Gambar 3.7 di dalam bagian *reader*. Aplikasi *database\_RS* berhubungan dengan *database* rumah sakit sederhana sebagai aplikasi *back-end*.



Gambar 3.7. Bagian-bagian aplikasi kartu akses pasien rumah sakit

### 3.4. ALIRAN DIAGRAM APLIKASI KARTU AKSES PASIEN

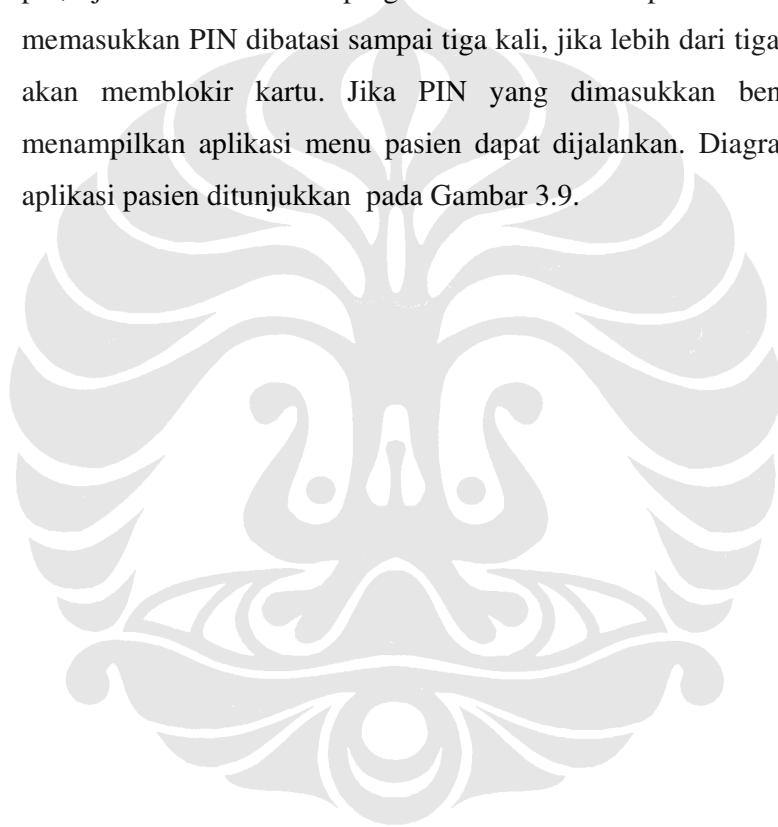
Aplikasi pengguna melakukan RMI *call* dengan cara melewati APDU-APDU RMI pada *on-card* JCRE, yang diteruskan ke *database\_RS applet*, setelah itu *applet* mengirimkan APDU yang diterima pada *RMIService*, APDU diproses dan diterjemahkan menjadi RMI *call*. JCRMI menyediakan mekanisme model objek yang terdistribusi, pada bagian atas dari diagram model pengiriman pesan dengan berbasis APDU. Pesan JCRMI diubah menjadi *format* pesan APDU yang dilewatkan ke *RMIService*, yang bertanggung jawab mengartikan perintah-perintah APDU dan menerjemahkan menjadi *metode invocation* dan *respons* sehingga *server* dan *client* dapat berkomunikasi, dengan cara melewati informasi *method*, *argument* dan mengirimkan nilai balik. Aliran diagram aplikasi kartu akses pasien ditunjukkan pada Gambar 3.8.

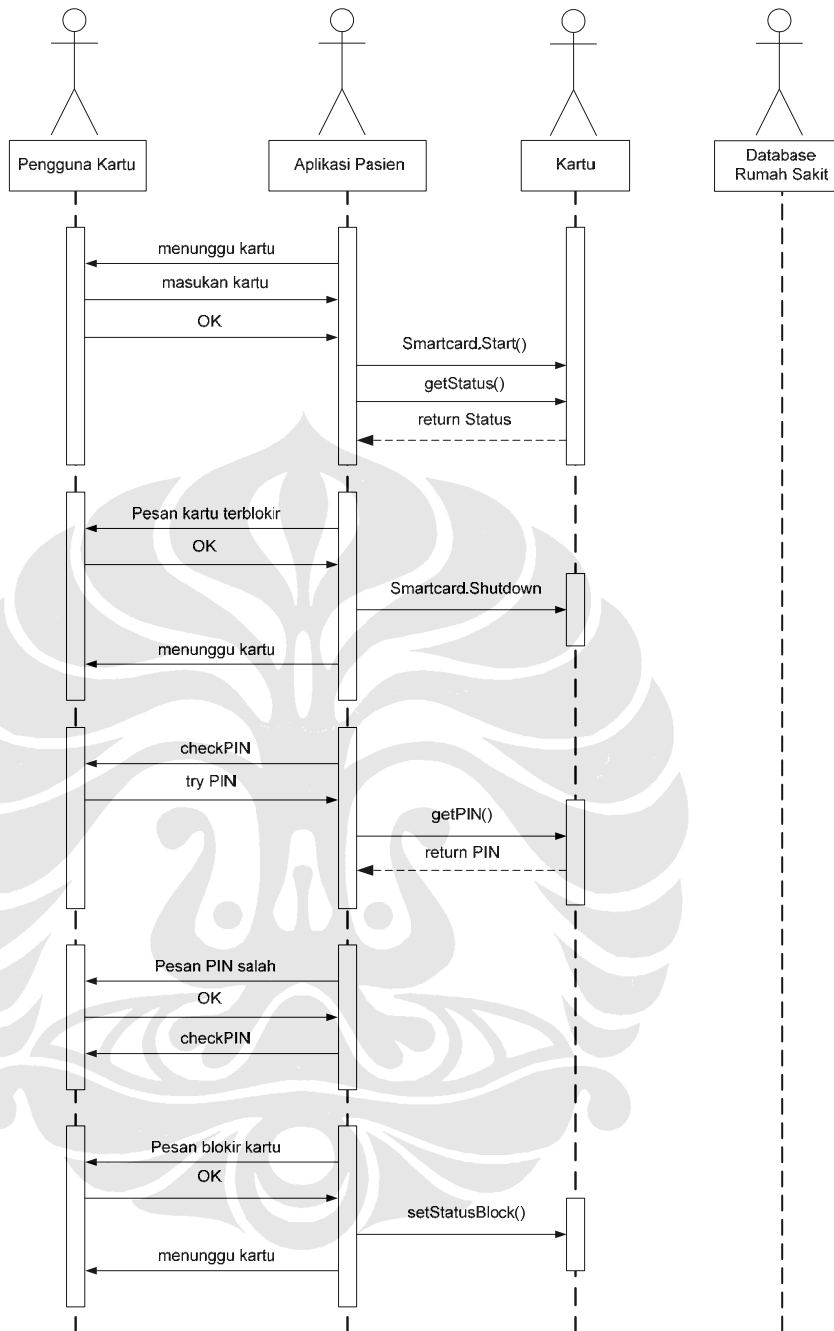


Gambar 3.8. Aliran diagram aplikasi kartu akses *smart card* pasien rumah sakit

### 3.5. SEKUENSIAL DIAGRAM APLIKASI KARTU AKSES *SMART CARD* PASIEN

Sekuensial diagram *login* pada aplikasi pengguna, pertama-tama program menunggu pengguna untuk memasukkan kartu dan setelah kartu dimasukkan maka program akan membaca status dari kartu, jika terblokir program memberi pesan bahwa kartu telah terblokir dan kartu ditolak. Jika kartu *valid* maka dilanjutkan dengan pemeriksaan terhadap PIN kartu. Pasien memasukkan nomor pin, jika salah maka program memberikan pesan kesalahan, percobaan memasukkan PIN dibatasi sampai tiga kali, jika lebih dari tiga kali maka program akan memblokir kartu. Jika PIN yang dimasukkan benar maka program menampilkan aplikasi menu pasien dapat dijalankan. Diagram sekuensial *login* aplikasi pasien ditunjukkan pada Gambar 3.9.



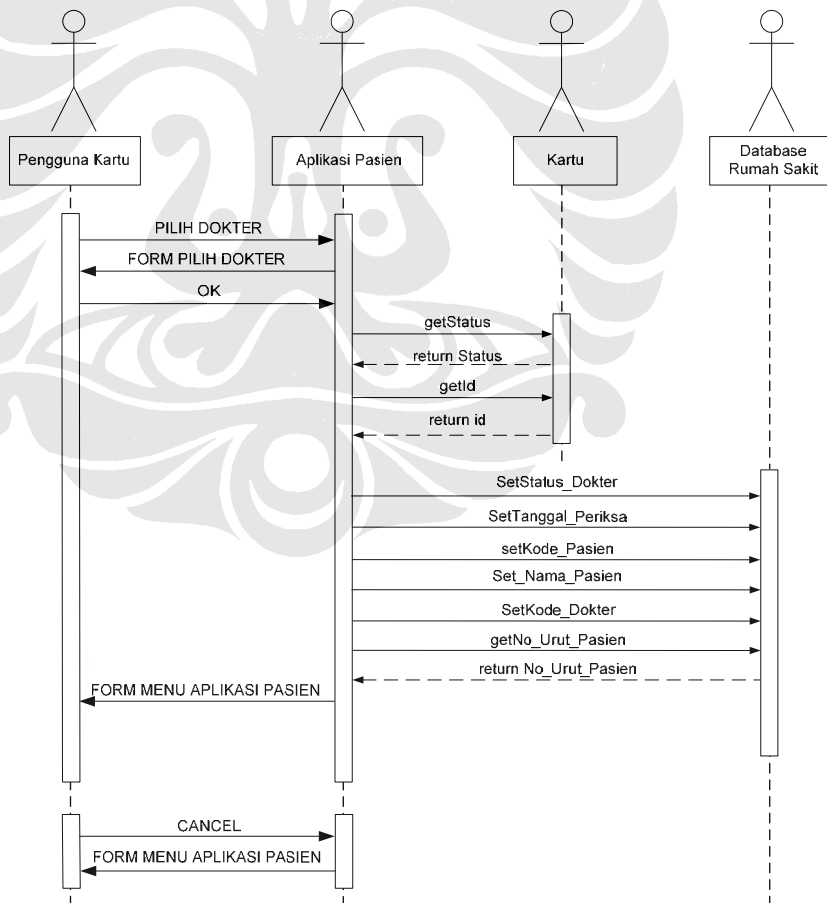


Gambar 3.9. Diagram Sekuensial *Login*

*Frame* aplikasi menu pasien menyediakan pilihan pada pasien untuk melakukan beberapa proses, yaitu melihat rekam medis miliknya sendiri, memilih dokter yang akan memeriksa, dan dapat mengganti PIN kartu. Setelah *login* berhasil dilakukan pengguna maka akan tampil menu aplikasi pasien dapat dijalankan. Jika pengguna menentukan untuk memilih dokter yang merawat maka

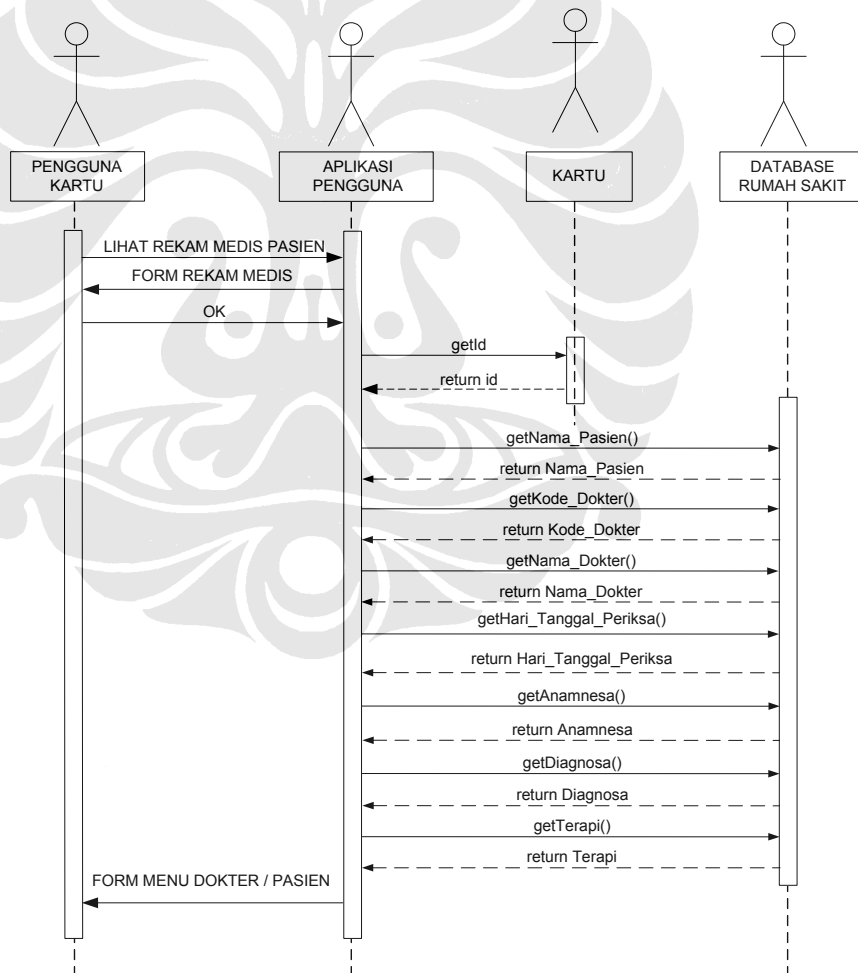


aplikasi pengguna akan mengakses kartu dengan memanggil beberapa *method* yang diperlukan seperti `GetStatus()`, `getID()`, `SetStatus_Dokter()`, `SetTanggal_Periksa()`, `SetKode_Pasien()`, `SetNama_Pasien()`, `SetKode_Dokter()`, dan `getNo_Urut_Pasien`. Setelah menentukan status dokter maka pada *form* pilih dokter tersebut akan tampil informasi yang berisi identitas dokter berdasarkan status dokter yang dipilih pasien. Setelah itu pasien dapat menentukan dokter yang akan merawatnya berdasarkan jadwal yang telah dibuat oleh bagian pembuatan jadwal praktek dokter rumah sakit. Kemudian pasien dapat melihat lokasi tempat dokter yang merawat dan mendapatkan nomor urut pasien. Setelah itu pengguna dapat mengakhiri proses dengan memilih menu kembali ke menu aplikasi pasien dan mengakhiri proses. Kemudian pasien menunggu di ruang tunggu yang telah disediakan. Diagram sekuensial memilih dokter ditunjukkan pada Gambar 3.10.



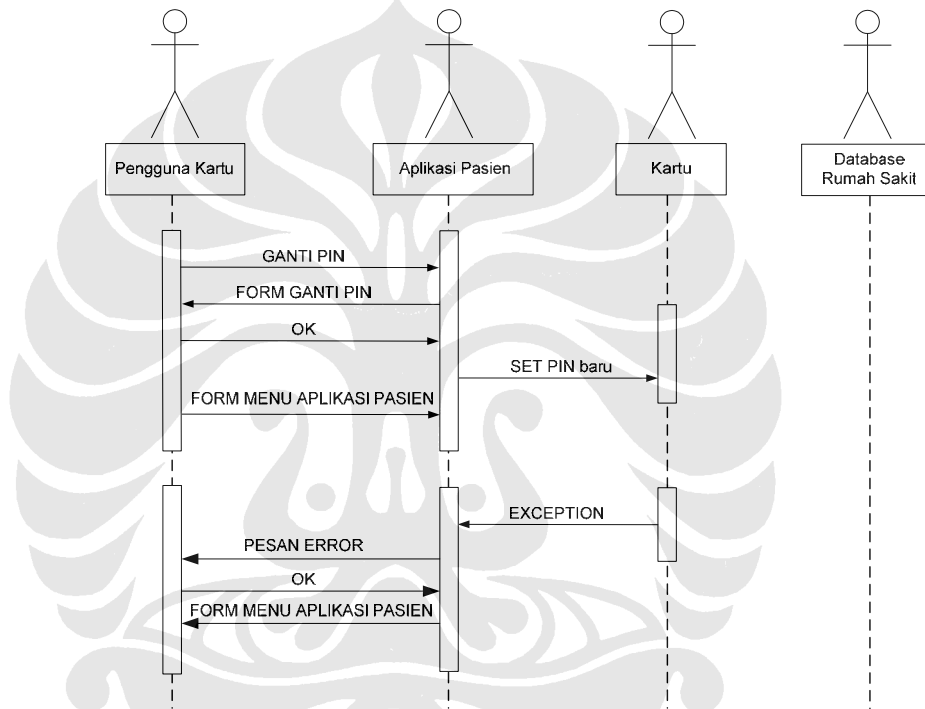
Gambar 3.10. Diagram Sekuensial Pilih Dokter

Aplikasi pengguna pasien dan dokter dapat memilih proses untuk melihat rekam medis. Pada aplikasi pengguna ini akan dibaca id kartu yang digunakan sebagai *argument*. Setelah kartu mengirimkan nilai balik id maka program akan mengakses *database* rumah sakit untuk membaca rekam medis pasien. Aplikasi pengguna ini akan mengakses kartu dengan memanggil beberapa *method* yang diperlukan seperti `getId()`, `getNama_Pasien()`, `getKode_Dokter()`, `getNama_Dokter()`, `getHari_Tanggal_Periksa()`, `getAnamnesa()`, `getDiagnosa()`, dan `getTerapi()`. Setelah melihat rekam medis pasien maka aplikasi akan kembali ke *frame* aplikasi pengguna pasien atau dokter. Gambar 3.11 menggambarkan diagram sekuensial lihat rekam medis pasien tersebut.



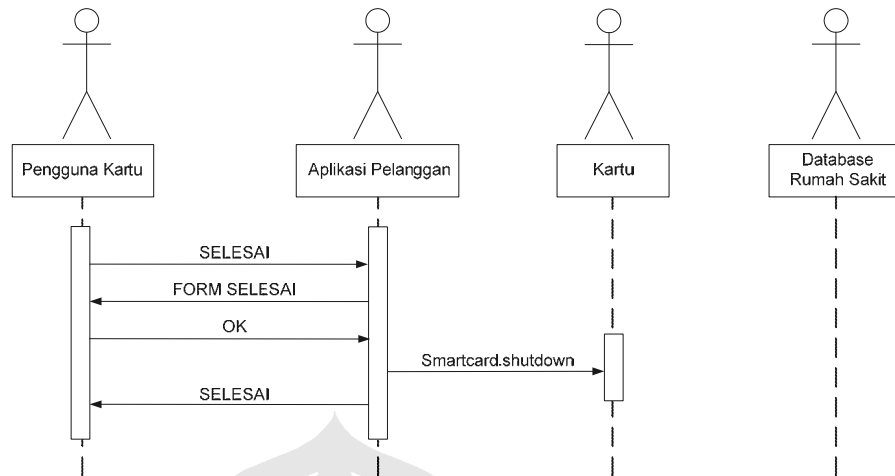
Gambar 3.11. Diagram Sekuensial Lihat Rekam Medis Pasien  
(dapat dilakukan oleh dokter yang merawat dan pasien)

Jika pengguna memilih melakukan proses untuk mengganti PIN kartu maka aplikasi pengguna menampilkan *frame* untuk mengganti PIN kartu, kemudian pasien dapat memasukkan PIN baru dan menekan tombol OK, kemudian aplikasi pasien mengakses kartu dengan memberikan pesan memanggil *method* `setPin()`, kartu akan menyimpan nomor PIN baru. PIN yang dimasukkan adalah 4 digit, jika terjadi kesalahan masukan maka program akan memberikan pesan terjadi kesalahan dan proses dibatalkan. Sekuensial diagram ganti PIN ditunjukkan pada Gambar 3.12.



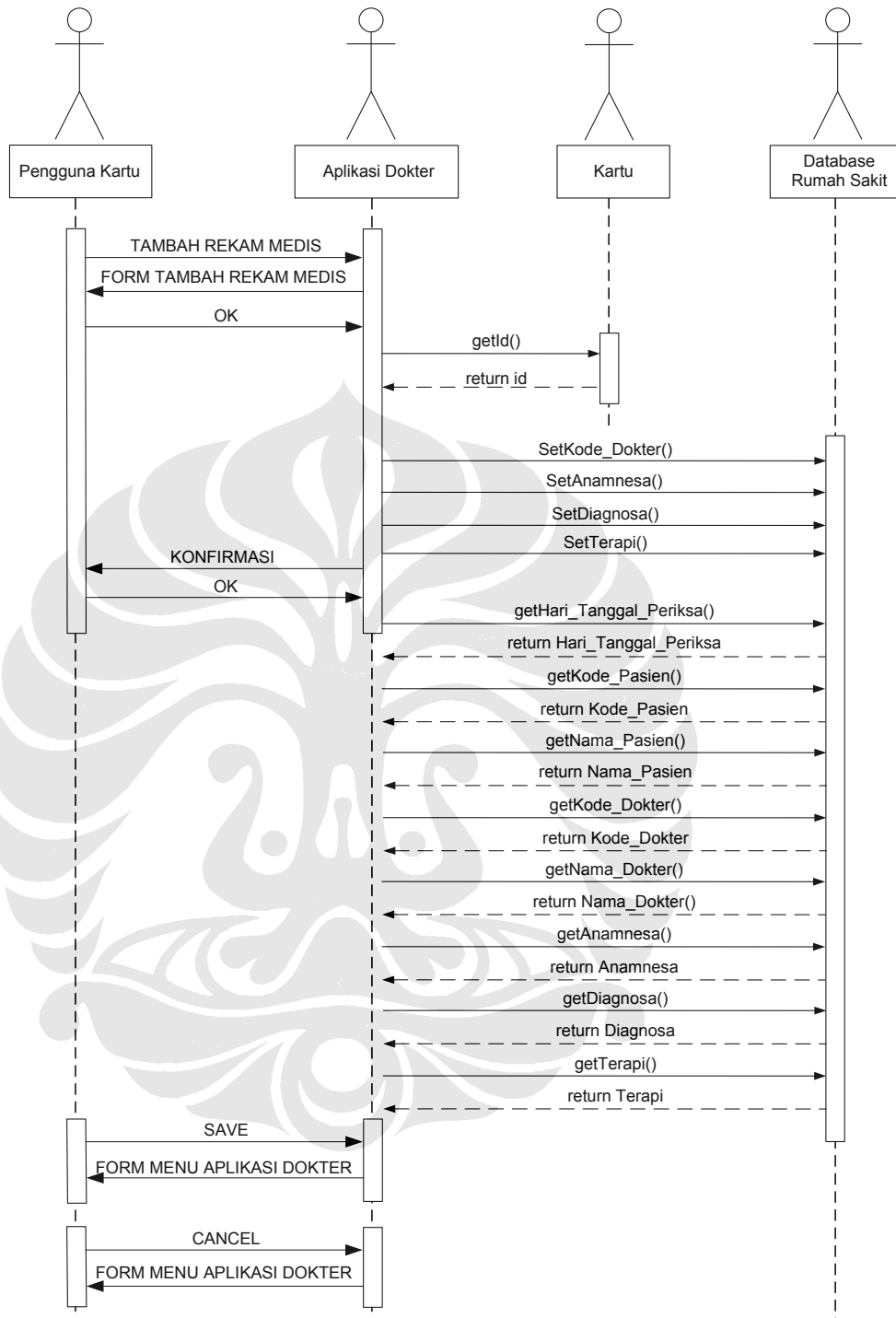
Gambar 3.12. Diagram Sekuensial Ganti PIN

Jika pasien selesai melakukan proses kemudian dapat menekan tombol selesai, maka aplikasi pengguna akan mematikan *power* kartu dengan memberikan pesan `SmartCard.shutdown()`. Aplikasi pengguna akan menampilkan *frame end* simulasi aplikasi pengguna. Sekuensial diagram selesai proses ditunjukkan pada Gambar 3.13.



Gambar 3.13. Diagram Sekuensial Selesai

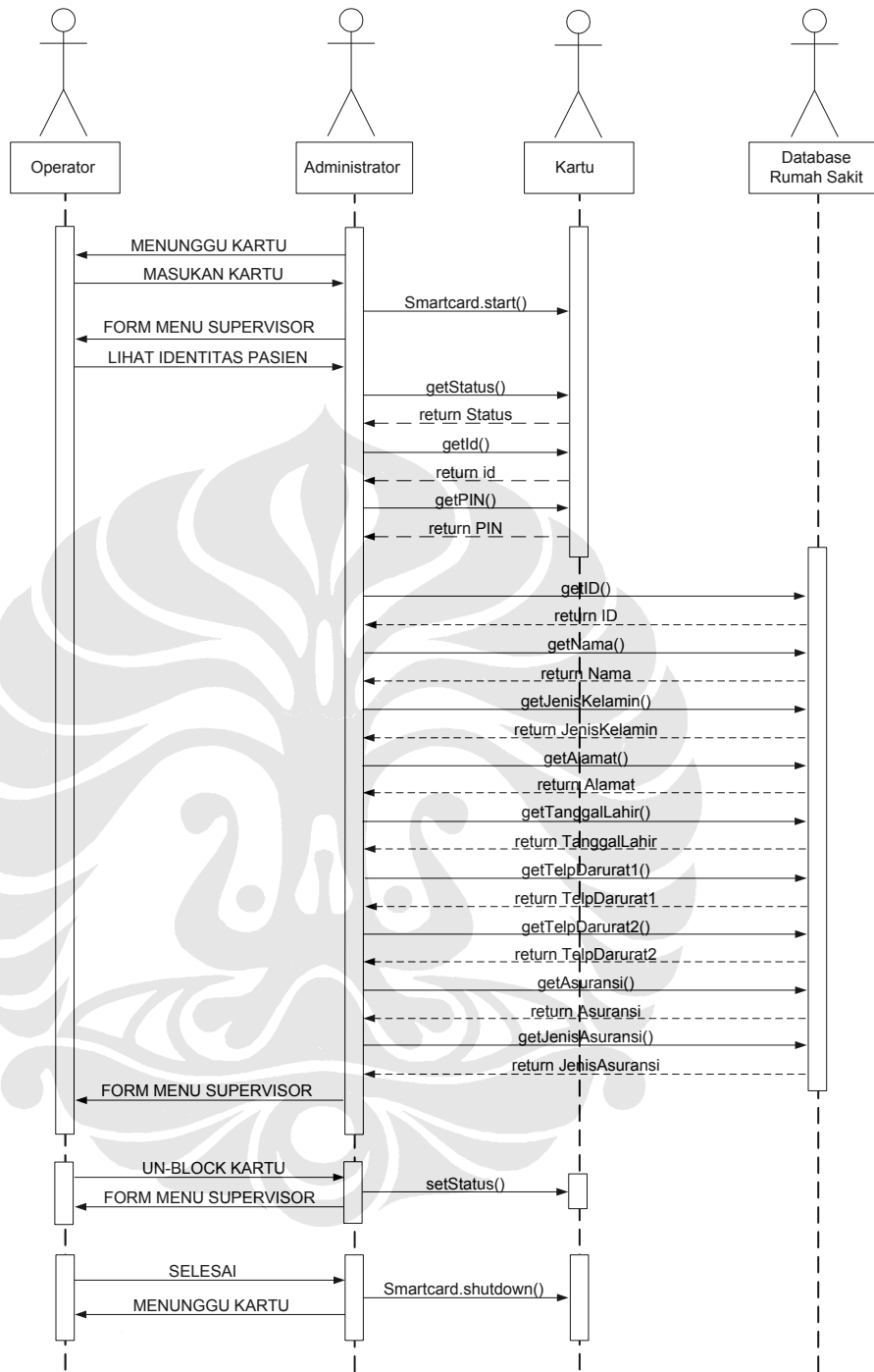
Pada aplikasi dokter, pertama-tama akan dilakukan pengecekan terhadap status kartu dan juga status pembayaran jika pasien telah berobat sebelumnya. Jika dokter memilih proses menambah rekam medis pasien ke dalam *database* rumah sakit maka program akan menampilkan *frame* tambah rekam medis, selanjutnya dokter diminta untuk memasukkan beberapa parameter yang diperlukan dengan memanggil *method* `SetKode_Dokter()`, `SetAnamnesa()`, `SetDiagnosa_Penyakit()`, dan `SetTerapi()`. Id kartu digunakan sebagai *argument* untuk mengakses *database* pasien yang bersangkutan. Sebelum dokter memasukkan data rekam medis pasien yang baru ke dalam *database* pasien rumah sakit, terdapat tombol untuk konfirmasi data rekam medis yang telah dibuat. Setelah dianggap bahwa data rekam medis pasien tersebut ditulis benar maka dokter dapat melakukan proses penyimpanan rekam medis. Jika terjadi kesalahan dalam memasukkan input seperti salah memasukkan kode dokter, dan tidak ada kesesuaian dengan *database* dokter rumah sakit maka pada layar tidak akan tampil data dokter yang merawat, maka dokter dapat membatalkan proses penyimpanan rekam medis dengan menekan tombol *cancel*. Sekuensial diagram tambah rekam medis pasien oleh dokter ditunjukkan pada Gambar 3.14.



Gambar 3.14. Diagram Sekuensial Tambah Rekam Medis Pasien oleh Dokter

Program aplikasi *supervisor* adalah program yang dijalankan oleh seorang operator atau petugas rumah sakit untuk melihat beberapa informasi dengan

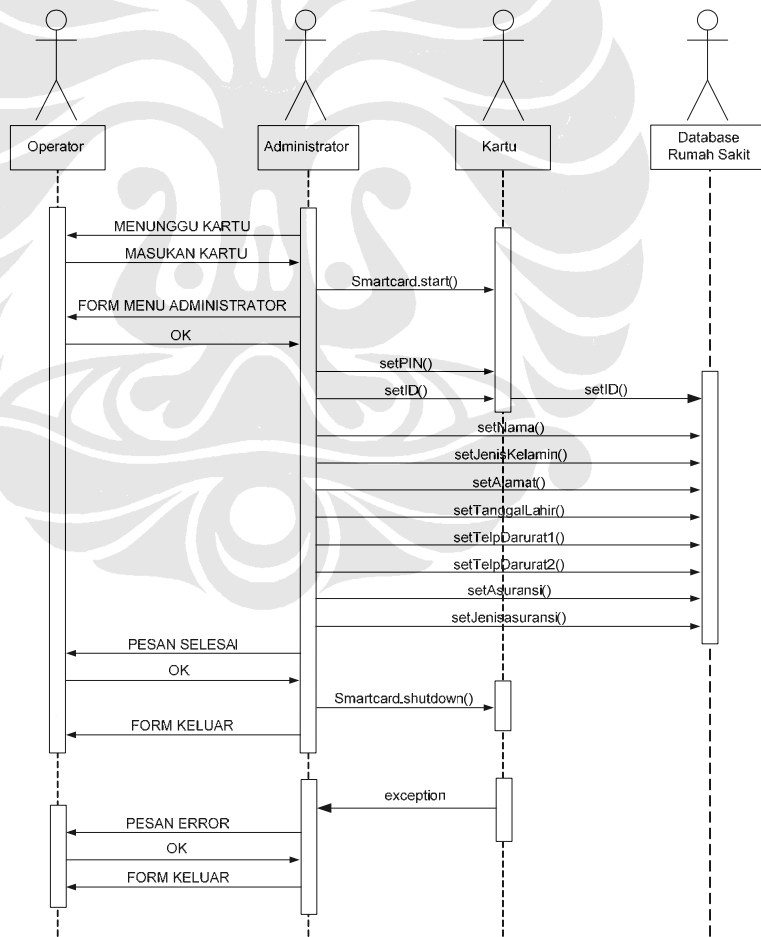
memanggil *method* `getstatus()`, `getid()`, dan `getPIN()`. Selain itu petugas dapat melihat data umum pasien dengan memanggil *method* `getid()`, `getNama()`, `getJenis_Kelamin()`, `getAlamat()`, `getTanggal_Lahir()`, `getTelepon_Darurat1()`, `getTelepon_Darurat2()`, `getAsuransi()`, `getJenisAsuransi()`, dan dapat me-*reset* status kartu. Sekuensial dari program, pertama kali dijalankan, program menampilkan *frame* aplikasi rumah sakit, setelah dipilih menu *supervisor* maka dilakukan proses menunggu kartu hingga dimasukkan kemudian program akan menampilkan *frame* menu aplikasi *supervisor* yang kemudian operator dapat memilih menu yang diberikan. Jika operator memilih untuk melihat identitas pasien, maka program akan mengakses kartu untuk membaca id kartu, dan PIN kartu serta dapat melihat identitas umum pasien yang meliputi id, nama, jenis kelamin, alamat, tanggal lahir, telepon darurat 1 dan telepon darurat 2, asuransi dan jenis asuransi yang tersimpan dalam *database* data umum pasien. Jika status kartu tidak *valid*, operator dapat me-*reset* status kartu dengan memilih menu `unblock` sehingga kartu dapat digunakan kembali oleh pasien. Kartu akan keluar jika operator menekan tombol selesai dan program pada keadaan menunggu kartu keluar dari *card reader*. Sekuensial diagram aplikasi *supervisor* ditunjukkan pada Gambar 3.15.



Gambar 3.15. Sekuensial Diagram Program Aplikasi *Supervisor*

Program aplikasi *administrator* dijalankan oleh operator atau petugas rumah sakit untuk menginisialisasi kartu baru, variabel yang diinisialisasi adalah id, dan PIN kartu. Selain itu juga dilakukan pengisian data umum pasien ke dalam *database* rumah sakit seperti mengisi *form* pendaftaran pasien rumah sakit dengan

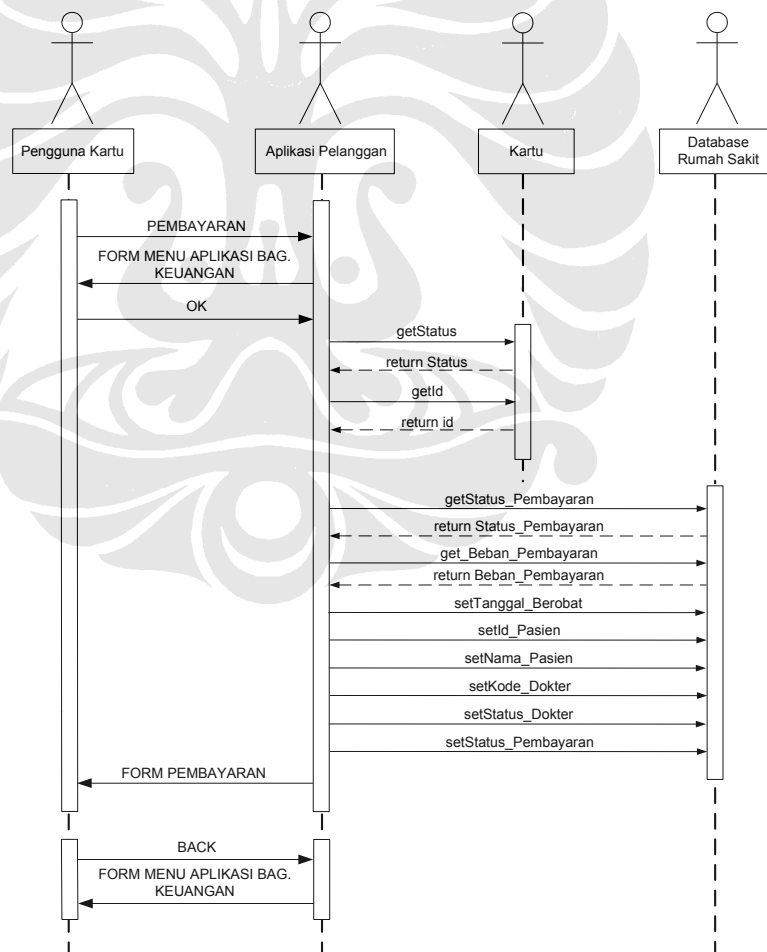
memanggil *methode* `setPIN()`, `setid()`, `setNama()`, `setJenis_Kelamin()`, `setAlamat()`, `setTanggal_Lahir()`, `setTelepon_Darurat1()`, `set_Telepon_Darurat2()`, `setAsuransi()`, dan `setJenis_Asuransi()`. Sekuensial dari program aplikasi *administrator*, pertama kali dijalankan program menampilkan *frame* simulasi *administrator* dan *supervisor*, setelah itu dipilih menu *administrator* maka program akan berada dalam keadaan menunggu kartu, dan setelah kartu dimasukkan maka program memberikan *frame administrator*, operator memasukkan *input* nilai variabel yang diinisialisasi, setelah itu program memberikan *frame* dialog yang menyatakan inialisasi telah selesai, jika terjadi kesalahan input maka program akan memberikan pesan kesalahan, sekuensial diagram aplikasi *administrator* ditunjukkan pada Gambar 3.16.



Gambar 3.16. Sekuensial Diagram Program Aplikasi *Administrator*



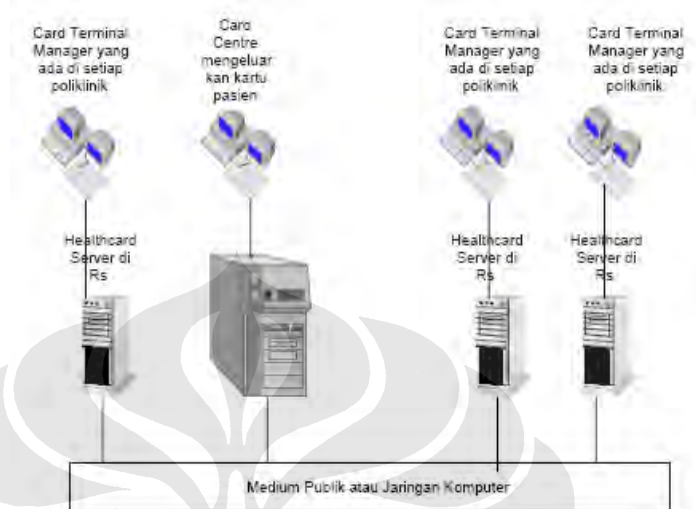
Program aplikasi bagian keuangan dijalankan oleh petugas rumah sakit bagian administrasi untuk melakukan transaksi pembayaran yang dilakukan oleh pasien. Pada awalnya, aplikasi bagian keuangan akan membaca status kartu kemudian membaca id kartu yang digunakan sebagai *argument*. Aplikasi bagian keuangan akan mengakses kartu dengan memanggil beberapa *method* seperti `getStatus_Pembayaran()`, dan `getBeban_Pembayaran()` kemudian menampilkan jumlah beban pembayaran pada *frame* pembayaran. Jika pasien telah membayar biaya pengobatan maka program aplikasi akan secara otomatis mengeset status pembayaran menjadi "LUNAS" dan jika Jika pasien belum membayar biaya pengobatan maka program aplikasi akan mengeset status pembayaran menjadi "BELUM LUNAS". Gambar 3.17 menggambarkan diagram sekuensial lihat rekam medis pasien tersebut.



Gambar 3.17. Sekuensial Diagram Program Aplikasi Bagian Keuangan

### 3.6. KONFIGURASI SISTEM RUMAH SAKIT

Konfigurasi sistem rumah sakit yang dirancang ditunjukkan oleh Gambar 3.18. yang menunjukkan infrastruktur yang umum ada antar rumah sakit.



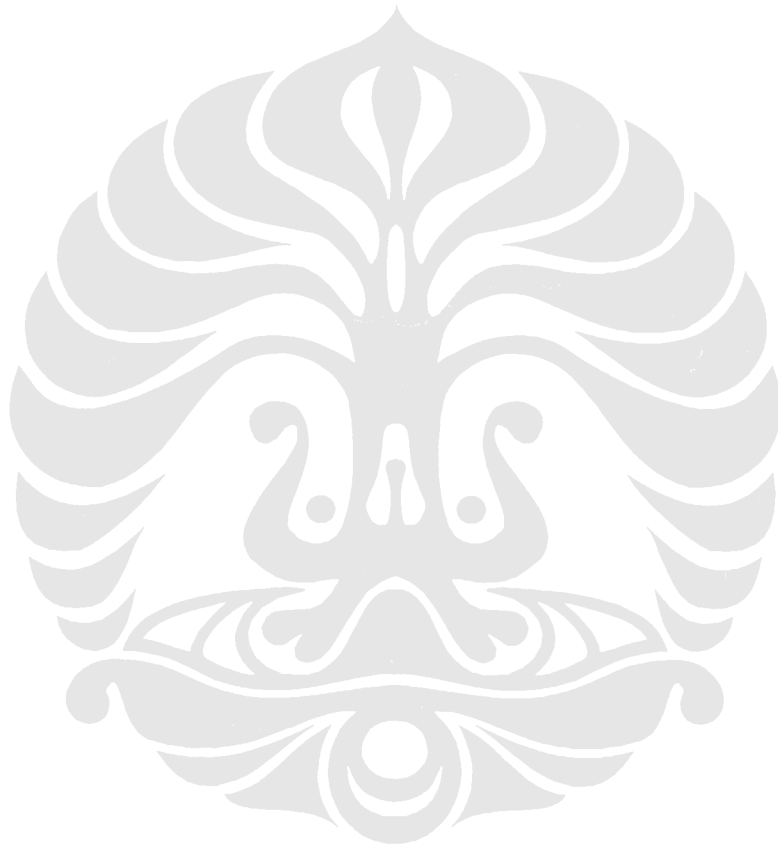
Gambar 3.18. Konfigurasi Sistem *Smart Card* Kesehatan

Secara garis besar sistem pendaftaran dan rekam medis pasien rumah sakit yang dirancang dapat dijelaskan sebagai berikut :

- Pihak-pihak yang terlibat adalah : dokter, pasien, *card centre* dan rumah sakit.
- *Card Centre* dan setiap rumah sakit bertugas untuk mengeluarkan kartu akses pasien dan juga berfungsi dalam aplikasi rumah sakit yaitu *administrator* dan *supervisor*.
- Setiap rumah sakit dan *card centre* terhubung satu sama lain melalui *medium* publik atau jaringan komputer. Jaringan komputer digunakan jika dibutuhkan pengiriman data dari satu pihak kepada pihak lain secara *on-line*.
- *Card Terminal* mengecek jenis *smart card* yang digunakan dan me-load perangkat lunak *Healthcard Server* yang sesuai dengan jenis *smart card* dan membaca data dari *smart card*. Data yang dibaca dari *smartcard* dipetakan sesuai dengan struktur data yang *interoperability* oleh *Healthcard Server* sehingga data tersebut siap digunakan oleh aplikasi dari berbagai *vendor*. Jika data yang sudah *interoperability* tersebut ingin

disimpan ke dalam *smart card*, maka perangkat lunak aplikasi dapat mengubah data yang *interoperability* menjadi data yang sesuai dengan struktur data *smart card* miliknya.

- Setiap pembacaan, penambahan, dan koreksi terhadap data rekam medis disimpan di dalam *database* oleh pihak rumah sakit melalui fasilitas pencatatan.



## BAB IV

### PENGUJIAN DAN ANALISA

#### 4.1. SIMULATOR KARTU PADA CAD

*Card Accept Device* (CAD) pada *Java Card Reference Implementation* dan *Java Card Development Kit* (JCDK) digunakan untuk mensimulasikan *smart card* di dalam *card reader*. *Java Card Reference Implementation* ditulis dalam bahasa pemrograman C sehingga disebut sebagai *C-language Java Card Runtime Environment* (*C-language Java card RE*) yang merupakan *simulator* untuk membuat ROM *mask*, C-JCRE mempunyai kemampuan untuk mensimulasikan *persistant memory* (EEPROM), menyimpan dan memakai kembali isi dari EEPROM dan *file disk*. *Applet* dapat diinstall pada C-JCRE. Pada JCDK versi 2.2.1, C-JCRE merupakan implementasi 32-bit, 64 KB akses memori.

Isi dari EEPROM dapat disimpan dan dapat digunakan kembali pada *invocation* berikutnya dari C-JCRE. Hal ini sama sifatnya dengan *smart card*. Menjalankan C-JCRE menggunakan perintah `cref` ditambah `option -i` dan diikuti dengan nama *file*, merupakan *file image* EEPROM pada saat sebelum *Java Card Virtual Machine bytecode* dieksekusi. Sedangkan perintah `cref` ditambah `option -o` dilakukan untuk menyimpan dan meng-*update* EEPROM pada nama *file* yang disebutkan.

Sebagai contoh nama *file* yang akan dijadikan sebagai simulator *smart card* berbasis *java card* dengan menggunakan C-JCRE yang akan dilakukan inisialisasi EEPROM bernama `pasien1`. Pada saat proses awal berjalan, `id` dan `pin` akan disimpan pada kartu dalam hal ini EEPROM `pasien1`. *C-language Java Card Runtime Environment* ditunjukkan pada Gambar 4.1.

```

KARTU PASIEN 1
32-bit Address Space implementation - with cryptography support
Copyright 2003 Sun Microsystems, Inc. All rights reserved.

Memory configuration
  Type   Base      Size      Max Addr
  RAM    0x0      0x800    0x7ff
  ROM    0x2000   0xb000   0xcfff
  E2P    0x10020   0xffe0   0x1ffff

  ROM Mask size =                0x8e35 =          36405 bytes
  Highest ROM address in mask = 0xae34 =          44596 bytes
  Space available in ROM =      0x21cb =           8651 bytes
EEPROM (0xffe0 bytes) restored from file "pasien1"
Using a pre-initialized Mask

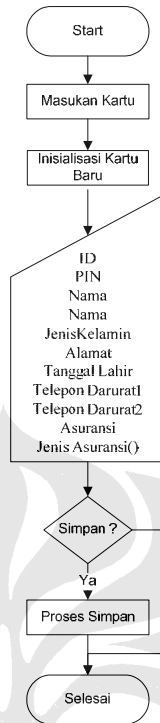
```

Gambar 4.1. *C-language Java Card Runtime Environment*

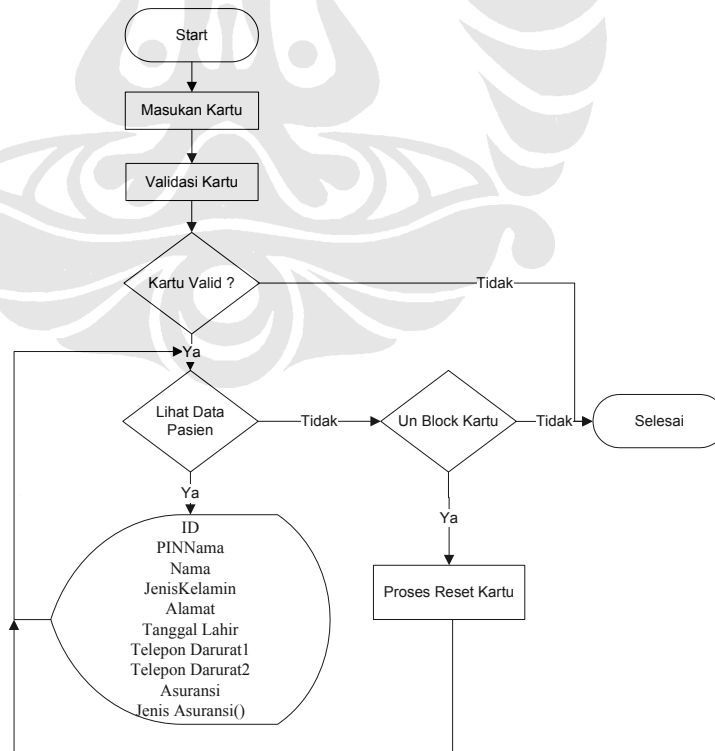
*Java Card reference implementation* versi 2.2.1 menyediakan versi 32-bit dari *C-language Java Card RE* dijalankan pada *Microsoft Windows XP platform*. Sedangkan *database* rumah sakit disimpan pada *database MySQL Control Center 0.9.2-beta-[Console Manager]*.

#### 4.2. **PENGUJIAN PROGRAM**

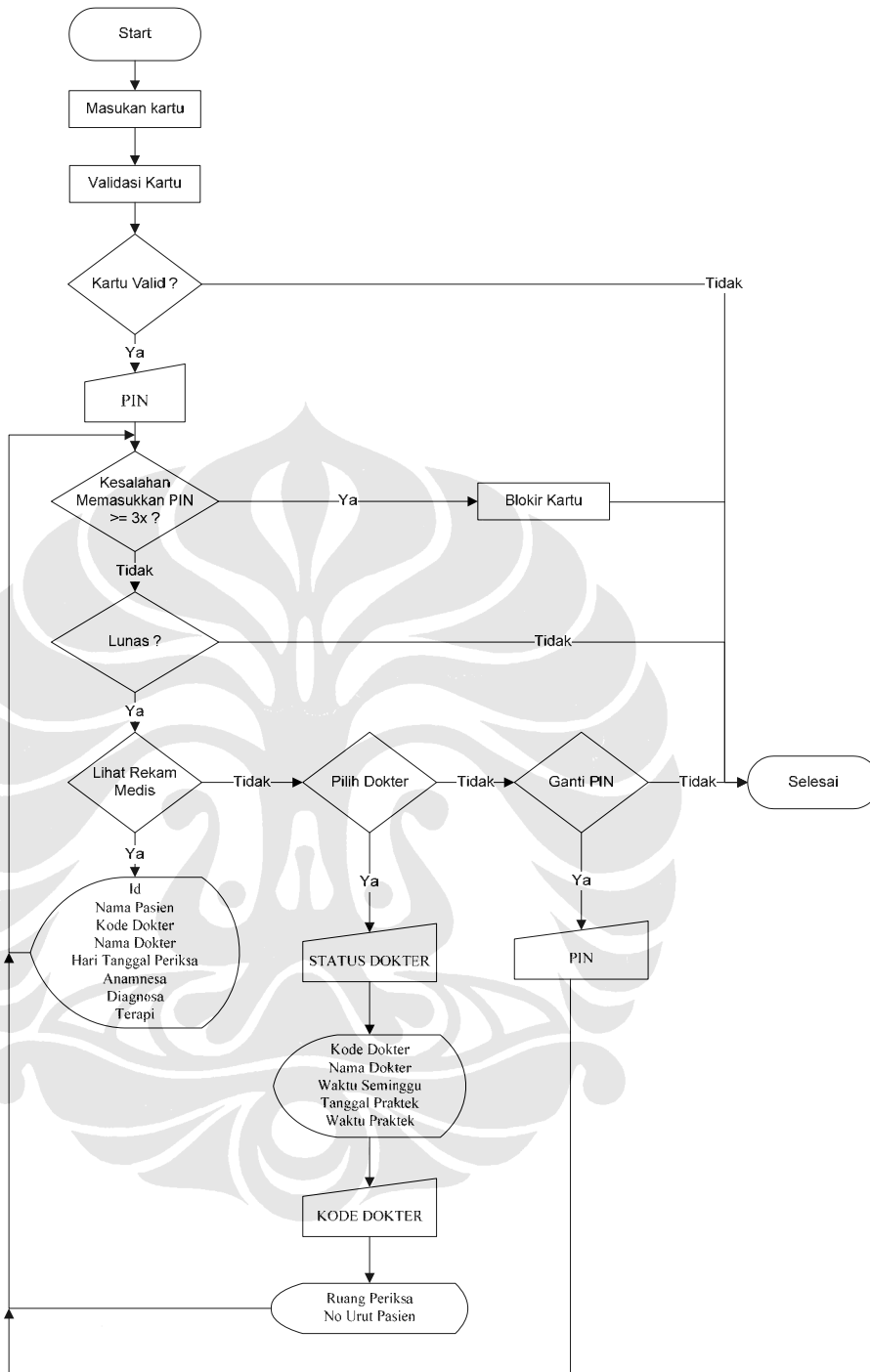
Berikut ditampilkan pengujian program aplikasi yang dibuat yaitu aplikasi rumah sakit yang meliputi *administrator* dan *supervisor*, aplikasi pasien, aplikasi dokter, dan aplikasi bagian keuangan yang mengikuti aliran diagram. Setiap aplikasi yang dijalankan, menggunakan kartu akses pasien sebagai *argument* untuk dilakukan proses inisialisasi kartu, dan pengenalan id pasien. Untuk pengujian terhadap aplikasi *supervisor*, dan aplikasi dokter dilakukan pengujian dengan memasukkan *input* secara *manual*, aliran diagram pengujian masing-masing aplikasi ditunjukkan pada Gambar 4.2 – 4.6 berikut ini.



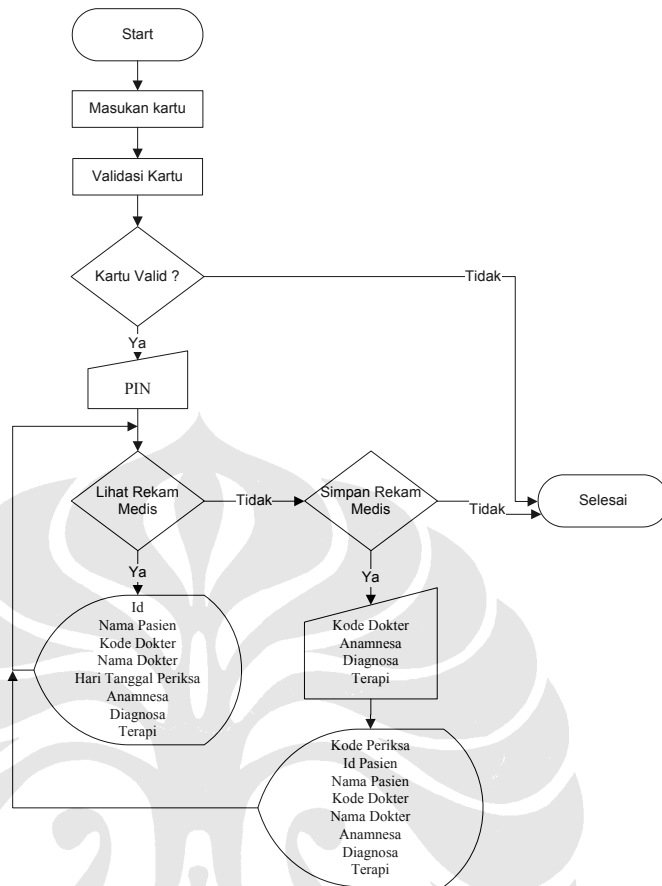
Gambar 4.2. Diagram alir pengujian aplikasi *administrator*



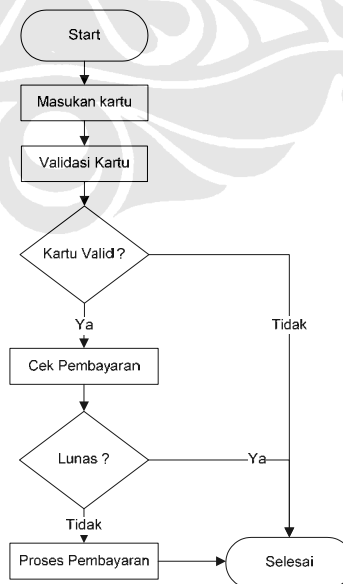
Gambar 4.3. Diagram alir pengujian aplikasi *supervisor*



Gambar 4.4. Diagram alir pengujian aplikasi pasien



Gambar 4.5. Diagram alir pengujian aplikasi dokter



Gambar 4.6. Diagram alir pengujian aplikasi bagian keuangan



Pengujian program aplikasi yang dibuat dilakukan dengan membuat sebuah skenario pengobatan pasien di rumah sakit (diperlihatkan pada Gambar 4.2 – Gambar 4.6 di atas) sebagai berikut:

- 1) suatu rumah sakit mempunyai fasilitas untuk mengeluarkan kartu pasien,
- 2) seorang pasien meminta dibuatkan kartu pasien,
- 3) seorang petugas rumah sakit memeriksa kartu baru dengan menjalankan aplikasi *supervisor*,
- 4) dari Gambar 4.2, petugas rumah sakit menjalankan aplikasi *administrator* untuk melakukan inisialisasi id (id kartu sama dengan id pasien pada *database* rumah sakit) dan PIN untuk pembuatan kartu baru,
- 5) kemudian petugas mengisi *form* pendaftaran pasien rumah sakit yang meliputi id, PIN, nama, jenis kelamin, alamat, tanggal lahir, telepon darurat 1, telepon darurat 2, nama perusahaan asuransi dan jenis asuransi. Untuk nama dan jenis asuransi diisi tanda strip (-) jika pasien tidak memiliki asuransi,
- 6) dari Gambar 4.3, petugas menjalankan aplikasi *supervisor* untuk memeriksa identitas kartu yang akan dikeluarkan,
- 7) kemudian petugas mengecek *form* pendaftaran pasien yang berisi data umum pasien yang telah dimasukkan pada aplikasi *administrator*,
- 8) kartu diberikan pada pasien. Dari Gambar 4.4, Pemegang kartu (pasien) kemudian mencoba *login* pada aplikasi pasien, dengan tidak menggunakan kartu,
- 9) memasukkan kartu dan *login* kembali, setelah berhasil *login* maka pemegang kartu (pasien) dapat menggunakan kartu tersebut ketika berkunjung ke rumah sakit karena pasien dapat melihat rekam medis miliknya sendiri, melakukan pendaftaran dokter yang akan merawat secara *on-line* dan di lain pihak dokter dapat melakukan pengisian rekam medis pasien rawat jalan, melihat riwayat rekam medis pasien,

- 10) memasukkan kartu dan *login*, setelah berobat tetapi belum melunasi biaya pengobatan,
- 11) memasukkan kartu dan *login*, setelah berobat setelah melunasi biaya pengobatan,
- 12) melihat rekam medis oleh pasien,
- 13) memilih dokter yang akan merawat,
- 14) mengganti PIN, melakukan kesalahan memasukkan PIN baru,
- 15) memasukkan lebih dari 4 digit, memasukkan input kurang dari 4 digit,
- 16) melakukan pemilihan terhadap dokter yang akan merawat,
- 17) pemegang kartu memasukkan kartu kembali dan melakukan tiga kesalahan percobaan PIN,
- 18) mencoba memasukkan kartu yang telah terblokir dan menjalankan aplikasi,
- 19) pemegang kartu menghubungi rumah sakit kemudian petugas rumah sakit menjalankan aplikasi *supervisor* melihat identitas kartu dan mereset kartu tersebut sehingga dapat digunakan kembali,
- 20) pemegang kartu mencoba *login* kembali,
- 21) keluar dari aplikasi pasien,
- 22) dari Gambar 4.5, melihat rekam medis oleh dokter yang merawat,
- 23) melakukan penambahan rekam medis oleh dokter yang merawat dengan memasukkan beberapa parameter yaitu kode dokter, anamnesa, diagnosa, dan terapi,
- 24) melakukan penambahan rekam medis oleh dokter yang merawat dengan kode dokter yang tidak terdaftar pada database rumah sakit,

- 25) dari Gambar 4.6, melakukan proses pembayaran oleh pasien setelah melakukan pengobatan tanpa kartu,
- 26) melakukan proses pembayaran oleh pasien setelah melakukan pengobatan dengan memasukkan kartu,
- 27) dari Gambar 4.4, menjalankan aplikasi pasien setelah melakukan pengobatan, jika belum melunasi biaya pengobatan maka akan tampil pada layar yang menyatakan pasien belum melunasi biaya pengobatan,
- 28) uji coba selesai.

#### 4.1 PENILAIAN PENGGUNA TERHADAP APLIKASI

Penilaian pengguna terhadap aplikasi rumah sakit, aplikasi pasien, aplikasi dokter, dan aplikasi bagian keuangan bertujuan untuk mengetahui pendapat dari pengguna terhadap aplikasi yang dibuat. Penilaian dibagi menjadi empat bagian, yaitu penilaian terhadap aplikasi rumah sakit, aplikasi pasien, aplikasi dokter, dan aplikasi bagian keuangan.

Penilaian diberikan berdasarkan bobot *point* penilaian yaitu :

- 1 = Sangat tidak setuju
- 2 = Tidak setuju
- 3 = Ragu-ragu
- 4 = Setuju
- 5 = Sangat setuju

Pada penilaian ini digunakan standar deviasi. Standar deviasi ini mengacu kepada seberapa lebar sebaran data dari nilai rata-ratanya. Selanjutnya dari standar deviasi ini dapat diperoleh nilai batas bawah dan batas dari *point* penilaian, yang diperoleh dengan rumus :

$$\text{Rata - rata} \pm 1,96 * \frac{\text{STDEV}}{\sqrt{\text{populasi}}} \qquad \text{Stdev} = \frac{\sqrt{\sum (\text{nilai} - \text{mean})^2}}{\text{populasi} - 1}$$

Stdev = standar deviasi

Populasi = banyaknya responden

Nilai rata-rata diperoleh dari hasil perkalian antara bobot nilai (1 sampai 5) dibagi dengan jumlah responden.

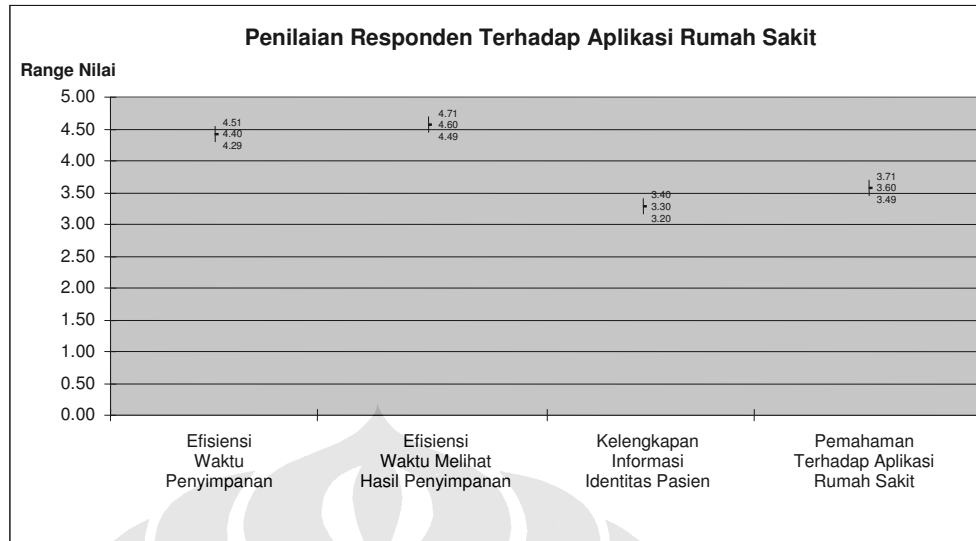
Persamaan perhitungan di atas merupakan perhitungan berdasarkan tingkat kepercayaan 95 %.

#### 4.1.1 Penilaian Pengguna Terhadap Aplikasi Rumah Sakit

Responden adalah orang yang berprofesi sebagai praktisi kesehatan dan bagian IT dari rumah sakit. Responden terdiri dari 10 orang. Parameter penilaian aplikasi rumah sakit meliputi efisiensi waktu penyimpanan, efisiensi waktu melihat hasil penyimpanan, kelengkapan informasi identitas pasien, dan pemahaman terhadap program aplikasi yang dibuat pada tabel 4.1.

Tabel 4.1. Penilaian Terhadap Aplikasi Rumah Sakit

No	Point Penilaian	Bobot Penilaian					Standar Deviasi	Batas Bawah	Mean	Batas Atas
		1	2	3	4	5				
1	Efisiensi Waktu Penyimpanan	0	0	0	6	4	0.17	4.29	4.40	4.51
2	Efisiensi Waktu Melihat Hasil Penyimpanan	0	0	0	4	6	0.17	4.49	4.60	4.71
3	Kelengkapan Informasi Identitas Pasien	0	0	7	3	0	0.16	3.20	3.30	3.40
4	Pemahaman Terhadap Aplikasi Rumah Sakit	0	0	4	6	0	0.17	3.49	3.60	3.71



Gambar 4.7. Grafik Penilaian Terhadap Aplikasi Rumah Sakit

Bedasarkan penilaian di atas dapat disimpulkan bahwa :

- Pada penilaian dari sisi efisiensi waktu penyimpanan, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 4,4, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, setuju bahwa penilaian dari sisi efisiensi waktu penyimpanan dari aplikasi rumah sakit dapat menambah efisiensi waktu.
- Pada penilaian dari sisi efisiensi waktu melihat hasil penyimpanan, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 4,6, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, setuju bahwa penilaian dari sisi efisiensi waktu melihat hasil penyimpanan dari aplikasi rumah sakit dapat menambah efisiensi waktu dalam melihat hasil penyimpanan. Hal ini karena responden dapat mencari atau melihat secara langsung identitas yang telah diisi dan tersimpan langsung ke dalam *database* identitas pasien rumah sakit.
- Pada penilaian dari sisi kelengkapan informasi identitas pasien, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 3,3, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, cukup setuju bahwa penilaian dari sisi kelengkapan informasi identitas pasien dari aplikasi rumah sakit merupakan identitas yang dapat mewakili identitas

pasien. Beberapa responden mengharapkan adanya penambahan informasi pasien lainnya agar dapat melengkapi identitas pasien yang biasa diterapkan oleh rumah sakit pada saat pendaftaran seperti agama, status perkawinan, pendidikan, pekerjaan, nomor induk karyawan, telepon kantor, dan lain-lain.

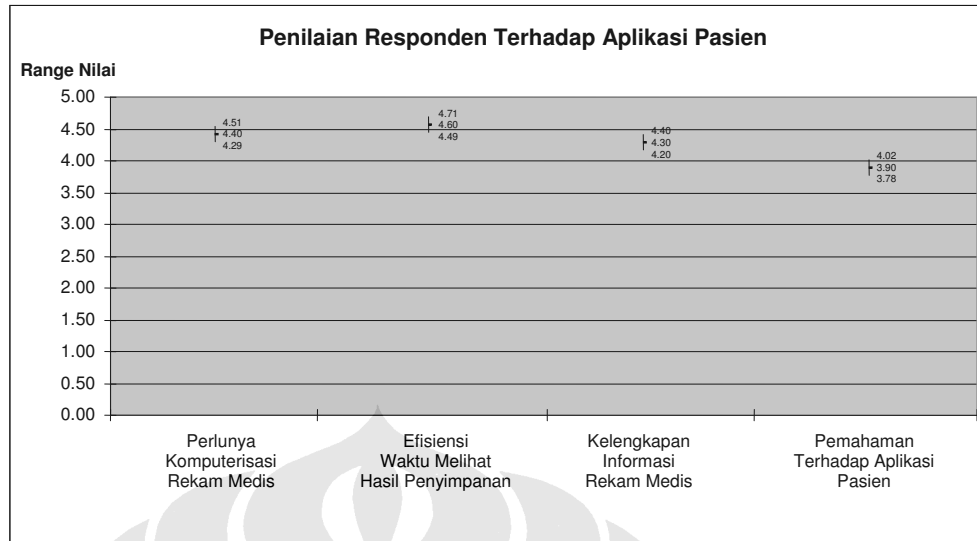
- Pada penilaian dari sisi pemahaman terhadap aplikasi rumah sakit, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 3,6, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, cukup memahami program yang dibuat oleh penulis.

#### 4.1.2 Penilaian Pengguna Terhadap Aplikasi Pasien

Responden adalah mahasiswa dan masyarakat umum. Responden terdiri dari 10 orang. Parameter penilaian setiap aplikasi didasarkan pada beberapa parameter penilaian yang meliputi perlunya komputerisasi rekam medis, efisiensi waktu melihat hasil penyimpanan rekam medis, kelengkapan informasi rekam medis, dan pemahaman terhadap program aplikasi yang dibuat pada tabel 4.2.

Tabel 4.2. Penilaian Terhadap Aplikasi Pasien

No	Point Penilaian	Bobot Penilaian					Standar Deviasi	Batas Bawah	Mean	Batas Atas
		1	2	3	4	5				
1	Perlunya Komputerisasi Rekam Medis	0	0	0	6	4	0.17	4.29	4.40	4.51
2	Efisiensi Waktu Melihat Hasil Penyimpanan Rekam Medis	0	0	0	4	6	0.17	4.49	4.60	4.71
3	Kelengkapan Informasi Rekam Medis	0	0	0	7	3	0.16	4.20	4.30	4.40
4	Pemahaman Terhadap Aplikasi Pasien	0	0	2	7	1	0.19	3.78	3.90	4.02



Gambar 4.8. Grafik Penilaian Terhadap Aplikasi Pasien

Bedasarkan penilaian diatas didapatkan bahwa :

- Pada penilaian dari sisi perlunya komputerisasi rekam medis, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 4,4, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, setuju bahwa rekam medis perlu untuk dilakukan komputerisasi seperti pendaftaran pasien secara *on-line*.
- Pada penilaian dari sisi efisiensi waktu melihat hasil penyimpanan, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 4,6, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, setuju bahwa penilaian dari sisi efisiensi waktu melihat hasil penyimpanan dari aplikasi pasien dapat menambah efisiensi waktu dalam melihat hasil penyimpanan. Hal ini karena responden secara langsung dapat mencari atau melihat rekam medis yang tersimpan dalam *database*.
- Pada penilaian dari sisi kelengkapan informasi rekam medis, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 4,3, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, setuju bahwa kelengkapan informasi rekam medis dari aplikasi pasien telah terpenuhi. Beberapa responden mengharapkan adanya penambahan rekam medis untuk rawat inap, hasil laboraturioum, dan lain-lain untuk dikembangkan berikutnya.

- Pada penilaian dari sisi pemahaman terhadap aplikasi pasien, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 3,9, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, cukup memahami program yang dibuat oleh penulis.

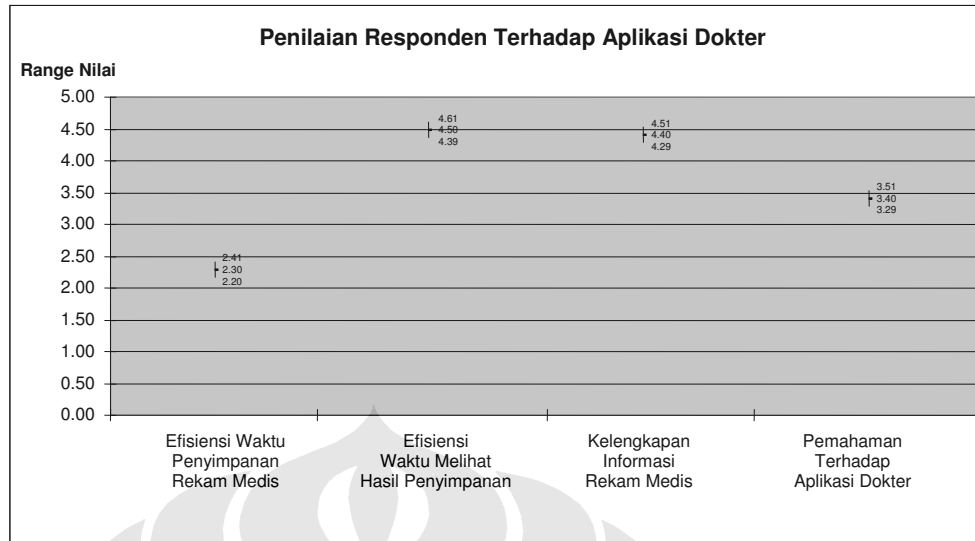
#### 4.1.3 Penilaian Pengguna Terhadap Aplikasi Dokter

Responden adalah orang yang berprofesi sebagai dokter, praktisi kesehatan, dan mahasiswa kedokteran yang sedang koas. Responden terdiri dari 10 orang. Parameter penilaian aplikasi pasien meliputi efisiensi waktu penyimpanan rekam medis, efisiensi waktu melihat hasil penyimpanan rekam medis, kelengkapan informasi rekam medis, dan pemahaman terhadap program aplikasi yang dibuat pada tabel 4.3.

Tabel 4.3. Penilaian Terhadap Aplikasi Dokter

No	Point Penilaian	Bobot Penilaian					Standar Deviasi	Batas Bawah	Mean	Batas Atas
		1	2	3	4	5				
1	Efisiensi Waktu Penyimpanan Rekam Medis	0	7	3	0	0	0.16	2.20	2.30	2.40
2	Efisiensi Waktu Melihat Hasil Penyimpanan	0	0	0	5	5	0.18	4.39	4.50	4.61
3	Kelengkapan Informasi Rekam Medis	0	0	0	6	4	0.17	4.29	4.40	4.51
4	Pemahaman Terhadap Aplikasi Dokter	0	0	6	4	0	0.17	3.29	3.40	3.51





Gambar 4.9. Grafik Penilaian Terhadap Aplikasi Dokter

Pada hasil penilaian pengguna, dapat ditarik kesimpulan sebagai berikut :

- Pada penilaian dari sisi efisiensi waktu penyimpanan rekam medis, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 2,3, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, kurang setuju bahwa aplikasi yang dibuat dapat menambah efisiensi waktu penyimpanan. Hal ini dikarenakan dokter lebih memiliki efisiensi terhadap waktu dalam membuat rekam medis secara manual yaitu ditulis dalam kertas.
- Pada penilaian dari sisi efisiensi waktu melihat hasil penyimpanan, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 4,5, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, setuju bahwa penilaian dari sisi efisiensi waktu melihat hasil penyimpanan dari aplikasi dokter dapat menambah efisiensi waktu dalam melihat hasil penyimpanan. Hal ini karena responden dapat mencari atau melihat secara langsung identitas yang telah diisi dan tersimpan dalam *database* identitas pasien rumah sakit.
- Pada penilaian dari sisi kelengkapan informasi identitas pasien, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 4,4, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, cukup

setuju bahwa kelengkapan informasi rekam medis pasien dari aplikasi dokter telah terpenuhi untuk pasien rawat jalan.

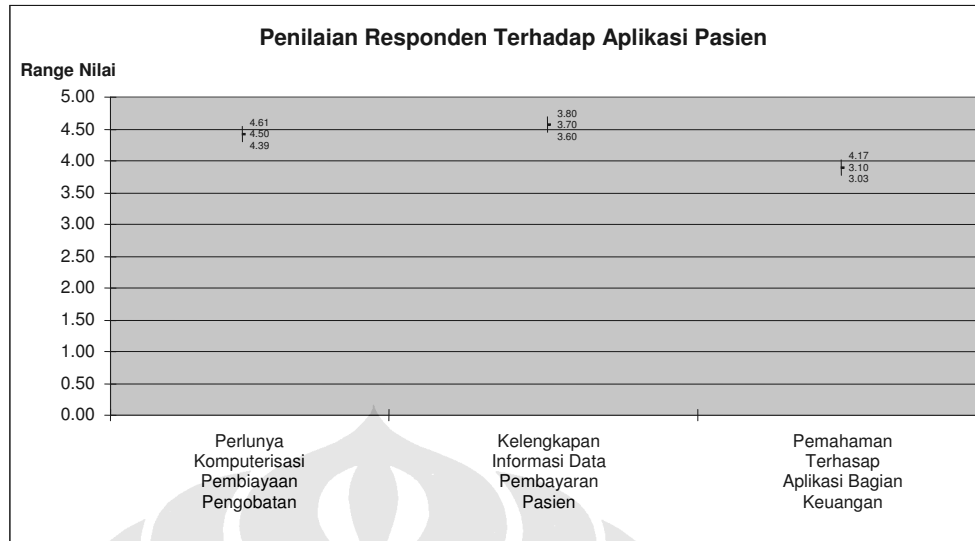
- Pada penilaian dari sisi pemahaman terhadap aplikasi dokter, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 3,4, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, cukup memahami program yang dibuat oleh penulis.

#### 4.1.4 Penilaian Pengguna Terhadap Aplikasi Bagian Keuangan

Responden adalah orang yang berprofesi sebagai staf bagian keuangan. Responden terdiri dari 10 orang. Parameter penilaian dari aplikasi bagian keuangan meliputi perlunya komputerisasi pembiayaan pengobatan, kelengkapan informasi data pembayaran pasien, dan pemahaman terhadap aplikasi yang dibuat.

Tabel 4.4. Penilaian Terhadap Aplikasi Bagian Keuangan

No	Point Penilaian	Bobot Penilaian					Standar Deviasi	Batas Bawah	Mean	Batas Atas
		1	2	3	4	5				
1	Perlunya Komputerisasi Pembiayaan Pengobatan	0	0	0	5	5	0.18	4.39	4.50	4.61
2	Kelengkapan Informasi Data Pembayaran Pasien	0	0	3	7	0	0.16	3.60	3.70	3.80
3	Pemahaman Terhadap Aplikasi Bagian Keuangan	0	0	0	9	1	0.11	4.03	4.10	4.17



Gambar 4.10. Grafik Penilaian Terhadap Aplikasi Bagian Keuangan

Pada hasil penilaian pengguna, dapat ditarik kesimpulan sebagai berikut :

- Pada penilaian dari sisi perlunya komputerisasi pembiayaan pengobatan, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 4,5, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, setuju bahwa penilaian dari pembiayaan pengobatan perlu dilakukan pencatatan dan penyimpanan pada komputer. Hal tersebut dapat membantu dalam membuat pembukuan atau laporan keuangan rumah sakit.
- Pada penilaian dari sisi kelengkapan informasi data pembayaran pasien, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 3,7, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, cukup setuju bahwa informasi data pembayaran pasien dari aplikasi bagian keuangan telah lengkap. Hal yang perlu ditambah adalah sistem pembayaran jika pasien melakukan pembayaran biaya pengobatan menggunakan asuransi.
- Pada penilaian dari sisi pemahaman terhadap aplikasi bagian keuangan, dari skala 1 – 5, point penilaian ini memiliki hasil penilaian rata-rata 3,6, hal ini berarti bahwa secara umum, dari 10 responden dengan tingkat keyakinan 95%, cukup memahami program yang dibuat oleh penulis.

### 4.3. ANALISA HASIL PENGUJIAN PROGRAM

Setelah program diujicobakan diketahui bahwa program berjalan sesuai dengan yang diharapkan pada perancangan, hasil yang didapat sebagai berikut:

- 1) kartu (C-JCRE) diinisialisasi dahulu pada program aplikasi *administrator* sebelum digunakan oleh pasien,
- 2) masukkan yang diberikan pada aplikasi *administrator* terdiri dari 4 karakter atau 4 *digit* dan untuk *input* id dan PIN,
- 3) kartu yang telah diinisialisasi dapat dilihat atau diperiksa dengan menjalankan program aplikasi *supervisor*,
- 4) aplikasi *supervisor* *mereset* status kartu yang telah terblokir (*not valid*),
- 5) apabila pasien menjalankan program dan mencoba *login* tanpa memasukkan kartu (menjalankan C-JCRE) program akan memberikan pesan bahwa kartu belum dimasukan,
- 6) pasien dapat mencoba memasukkan PIN hingga tiga kali percobaan, jika pasien melakukan kesalahan sebanyak tiga kali maka kartu akan diblokir sehingga menjadi tidak *valid* untuk pasien,
- 7) jika pasien memasukkan kartu yang telah terblokir, maka program akan menolak kartu dan memberikan pesan bahwa kartu telah terblokir,
- 8) jika pasien belum melunasi biaya pengobatan setelah berobat maka aplikasi pasien tidak dapat dijalankan,
- 9) jika pasien berhasil *login* dengan memasukkan PIN kartu yang benar program memberikan *form* aplikasi pasien yaitu melihat rekam medis, melakukan pemilihan dokter yang merawat secara *online*, dan mengganti PIN baru,
- 10) pasien dapat melihat rekam medis miliknya sendiri pada aplikasi pasien,
- 11) pasien dapat memilih dokter yang akan merawat secara *online*,

- 12) dokter dapat melihat riwayat rekam medis pasien dan menambahkan rekam medis. Jika dokter akan menambahkan rekam medis pasien maka akan ada tampilan mengenai data yang akan diisikan pada database pasien sebagai konfirmasi pengisian rekam medis, dan jika terdapat kesalahan dalam pengisian rekam medis maka proses dapat dibatalkan,
- 13) pengisian rekam medis pasien dilakukan dengan menambahkan rekam medis pasien baru pada *database* pasien dengan menggunakan id pasien,
- 14) sebelum proses pengisian rekam medis dijalankan, maka program akan memberikan *form* konfirmasi pengisian rekam medis, yang meliputi kode dokter, anamnesa, diagnosa penyakit, dan terapi yang diisi oleh dokter yang merawat,
- 15) pasien dapat melakukan penggantian nomor PIN dengan memberi 4 digit PIN baru. Kartu akan dikeluarkan (C-JCRE tertutup) jika pasien menekan tombol selesai pada aplikasi,
- 16) jika pasien telah berobat maka pasien wajib membayar biaya pengobatan sebelum berobat lagi,
- 17) dari hasil pengamatan penulis terhadap responden pada aplikasi dokter, dapat diperoleh kesimpulan bahwa dokter lebih efisien terhadap waktu pada pencatatan rekam medis secara manual daripada dengan komputer, karena dalam kondisi banyak pasien maka diperlukan diagnosa yang cepat pula sehingga lebih cepat pencatatan rekam medis secara manual.
- 18) menurut penulis, pencatatan rekam medis dilakukan oleh petugas rekam medis sedangkan dokter mengisi form pemeriksaan fisik yang telah dilengkapi pertanyaan untuk rekam medis secara manual sehingga kecepatan pemeriksaan pada saat banyak pasien dapat dilayani lebih baik setelah melakukan analisa terhadap responden dari dokter,
- 19) perlunya sosialisasi terhadap sistem rekam medis yang baru,
- 20) butuh waktu untuk memahami program aplikasi yang baru.

## BAB V

### KESIMPULAN

Setelah program dianalisa dan diuji, maka dapat disimpulkan sebagai berikut.

1. Tugas akhir ini mengimplementasikan teknologi *Java Card* sebagai aplikasi kartu pasien rumah sakit dengan menggunakan C-JCRE sebagai pengganti JCRE pada *Smart Card* yang sesungguhnya yang dijalankan dalam *simulator JCWDE*.
2. Aplikasi yang dijalankan dapat digunakan sebagai kartu akses pasien rumah sakit yang terintegrasi dengan database pasien rumah sakit yang dapat melakukan aktivitas sebagai berikut:
  - Membuat kartu akses pasien, mengecek kartu, dan mereset kartu yang terblokir oleh petugas administrasi rumah sakit.
  - Melihat riwayat rekam medis, menambah dan menyimpan rekam medis oleh dokter/petugas rekam medis.
  - Melihat riwayat rekam medis, melakukan pemilihan dokter yang merawat secara *on line*, dan mengganti PIN oleh pasien.
  - Melakukan pembayaran biaya pengobatan.
3. Hasil pengamatan penulis dari pendapat beberapa responden untuk mengatasi kurang efisiensinya waktu pencatatan rekam medis, maka pencatatan rekam medis dapat dilakukan oleh petugas rekam medis sedangkan dokter mengisi form pemeriksaan fisik yang telah dilengkapi pertanyaan untuk rekam medis secara manual untuk memberikan waktu yang lebih efisien, dan diperlukan sosialisasi terhadap sistem rekam medis yang baru.

## DAFTAR ACUAN

- [1] Peraturan Menteri Kesehatan Republik Indonesia Nomor : 749a/MENKES/PER/XII/1989 tentang Rekam Medis / *Medical Records*. Diakses pada tanggal 8 Maret 2008.
- [2] Medical Records Institute: Homepage, 1999. <http://www.medrecinst.com/resources/levels.html>. Diakses pada tanggal 8 Maret 2008.
- [3] Gde, A.a. Muninjaya. Manajemen Kesehatan edisi 2. 2004. Jakarta : Penerbit Buku Kedokteran EGC.
- [4] Sanjoyo, Raden. Aspek Hukum Rekam Medis. D3 Rekam Medis FMIPA Universitas Gadjah Mada. <http://www.yoyoke.web.ugm.ac.id>. Diakses pada tanggal 8 Maret 2008.
- [5] Jusuf, M. Hanafiah dan Amir, Amri. *Etika Kedokteran & Hukum Kesehatan edisi 3*. 1999. Jakarta : Penerbit Buku Kedokteran EGC.
- [6] Enrique C. Ortiz. *An Introduction to Java Card Technology*. 2003. <http://developers.sun.com/techtopics/mobility/javacard/articles/javacard1/>. Diakses pada tanggal 10 Maret 2008.
- [7] Java Card™ 2.2, *Application Programming Interface, Revision 1.1 for the 2.2\_01 Release*, Sun Microsystems, Inc., 2002. <http://www.JavaWorld.com>. Diakses pada tanggal 10 Maret 2008.
- [8] Alfitri, Nadia, Akses Database menggunakan Java Card, ITB, Bandung, 2006.
- [9] Ed Ort, *Developing a Java Card Applet*, 2001, <http://developers.sun.com/techtopics/mobility/javacard/articles/applet/Wallet.java>. Diakses pada tanggal 10 Maret 2008.
- [10] Ed Ort, *Writing a Java Card Applet*, 2001, <http://developers.sun.com/techtopics/mobility/javacard/articles/applet/Wallet.java>. Diakses pada tanggal 10 Maret 2008.
- [11] Zhiquan Chen, Rinaldo Di Giorgio, *Understanding Java Card 2.0, Learn the Inner Workings of the Java Card Architecture, API, and Runtime Environment*, 03/01/98, <http://www.JavaWorld.com>. Diakses pada tanggal 10 Maret 2008.
- [12] Zhiquan Chen, *How To Write A Java Card Applet, A Developer's Guide, Learn the Programming Concepts and Major Steps of Creating Java Card*

*Applets*, 07/01/99, <http://www.JavaWorld.com>. Diakses pada tanggal 10 Maret 2008.

[13] *Java Card Applet Developer's Guide*, Sun Microsystems, Inc., Revision 1.12,1998. Diakses pada tanggal 10 Maret 2008.

[14] Di Giorgio, By Rinaldo, *How to Write a CardTerminal Class for Simple and Complex Readers in an OpenCard Environment*, 1999, <http://www.JavaWorld.com>. Diakses pada tanggal 10 Maret 2008.





## DAFTAR PUSTAKA

- Bergner, Klaus, Rausch, Andreas, Sihling, Marc, *Using UML for Modeling a Distributed Java Application*, 30<sup>th</sup> July 1997 <http://www.informatik.tu-muenchen.de> , diakses pada tanggal tanggal 10 Maret 2008.
- Di Giorgio, By Rinaldo, *A primer, Develop on the Java platform of the Future*, 1997, <http://www.JavaWorld.com>, diakses pada tanggal tanggal 10 Maret 2008.
- Di Giorgio, By Rinaldo, *How to Write a CardTerminal Class for Simple and Complex Readers in an OpenCard Environment*, 1999, <http://www.JavaWorld.com>, diakses pada tanggal tanggal 10 Maret 2008.
- Eckel, Bruce, *Thinking in Java 3<sup>rd</sup> Edition, Revision 4*, President, MindView, Inc.2002.
- Gde, A.a. Muninjaya. *Manajemen Kesehatan edisi 2*. 2004. Jakarta : Penerbit Buku Kedokteran EGC.
- Hermawan, Benny. *Menguasai Java 2 & Object Oriented Programming*. 2004. Yogyakarta : ANDI.
- Java Card™ 2.2, *Off-Card Verifier*, Sun Microsystems, Inc., 2002
- Java Card™ Platform, Version 2.2.2, *RMI Client Application Programming Interface*, Sun Microsystems, Inc., Revision 1.12, 2006
- Java Card Applet Developer's Guide*, Sun Microsystems, Inc., Revision 1.12, 1998
- Jusuf, M. Hanafiah dan Amir, Amri. *Etika Kedokteran & Hukum Kesehatan edisi 3*. 1999. Jakarta : Penerbit Buku Kedokteran EGC.
- Kadir, Abdul, Dasar *Pemrograman Java 2*, 2003.
- Martin, Hugues , du Bousquet, Lydie, *Automatic Test Generation for Java-Card Applets*
- Supardi, Yuniar Ir.. *Pemrograman Database dengan Java dan MySQL*. 2007. Jakarta : PT Elex Media Komputindo.
- Website. <http://www.phcsg.org/main/pastconf/camb96>.
- Website. [http://en.wikipedia.org/wiki/medical\\_record](http://en.wikipedia.org/wiki/medical_record).

## LAMPIRAN

### I. Tampilan Aplikasi Rumah Sakit

#### a. Aplikasi Administrator



Gambar L1. Start Aplikasi Rumah Sakit



ID (4 digit)	0001
PIN (4 digit)	1234
NAMA	DWI KURNIA PUTRA
JENIS KELAMIN	LAKI-LAKI
ALAMAT	JL. YAHYA NUIH NO 36 PONDOK CINA DEPOK
TANGGAL LAHIR	21-04-1984
TELEPON DARURAT 1	0813 21397860
TELEPON DARURAT 2	0761 26087
NAMA ASURANSI / NOMOR POLIS (*)	-
JENIS ASURANSI (*)	-

\* = jika tidak ada isi dengan (-)

Gambar L2. Form Aplikasi Administrator

SIMULASI REKAM MEDIS

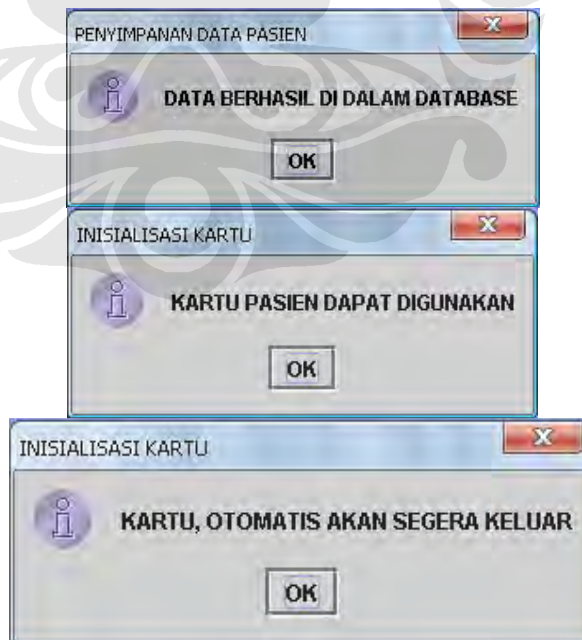
DEPARTEMEN TEKNIK ELEKTRO

KONFIRMASI PENYIMPANAN DATA PASIEN

<b>KARTU</b> 	ID KARTU	0001
	PIN (4 digit)	1234
<b>DATABASES</b> 	ID	0001
	NAMAPASIE	DWI KURNIA PUTRA
	JENIS KELAMIN	LAKI-LAKI
	ALAMAT	JL. YAHYA NUIH NO 36 PONDOK CINA DEPOK
	TANGGAL LAHIR	21-04-1984
	TELEPON DARURAT 1	0813 21397860
	TELEPON DARURAT 2	0761 26087
	NAMA ASURANSI / NOMOR POLIS	-
	JENIS ASURANSI	-

**CANCEL** **SAVE**

Gambar L3. Form Konfirmasi Penyimpanan Data Umum Pasien



Gambar L4. Konfirmasi Berhasil Menyimpan Data Umum Pasien

## b. Aplikasi Supervisor



Gambar L5. Form Aplikasi Supervisor

KODE PASIEN	NAMA PASIEN	JENIS KELAMIN	TANGGAL LAHIR
0001	DWI KURNIA PUTRA	LAKI-LAKI	21-04-1984

KLIK DALAM TABEL UNTUK KETERANGAN LEBIH LANJUT

DATA KARTU

KODE PASIEN: 0001

PIN: 1234

KARTU

Card Centre menggebuur kan kartu pasien

DATA PASIEN DALAM DATABASE RUMAH SAKIT

KODE PASIEN: 0001

NAMA PASIEN: DWI KURNIA PUTRA

ALAMAT: JL. YAHYA NUIH NO 36 POCIN DEPOK

JENIS KELAMIN: LAKI-LAKI

TANGGAL LAHIR: 21-04-1984

TELEPON DARURAT 1: 081 321 397860

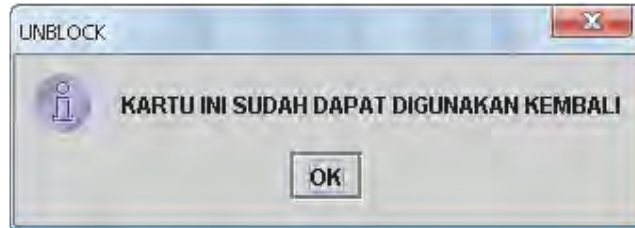
TELEPON DARURAT 2: 0761 26087

ASURANSI PASIEN: -

JENIS ASURANSI: -

BACK EDIT HAPUS

Gambar L6. Form Lihat Identitas Umum Pasien pada Aplikasi Pasien



Gambar L7. Mereset Kartu yang Terblokir

## II. Aplikasi Pasien



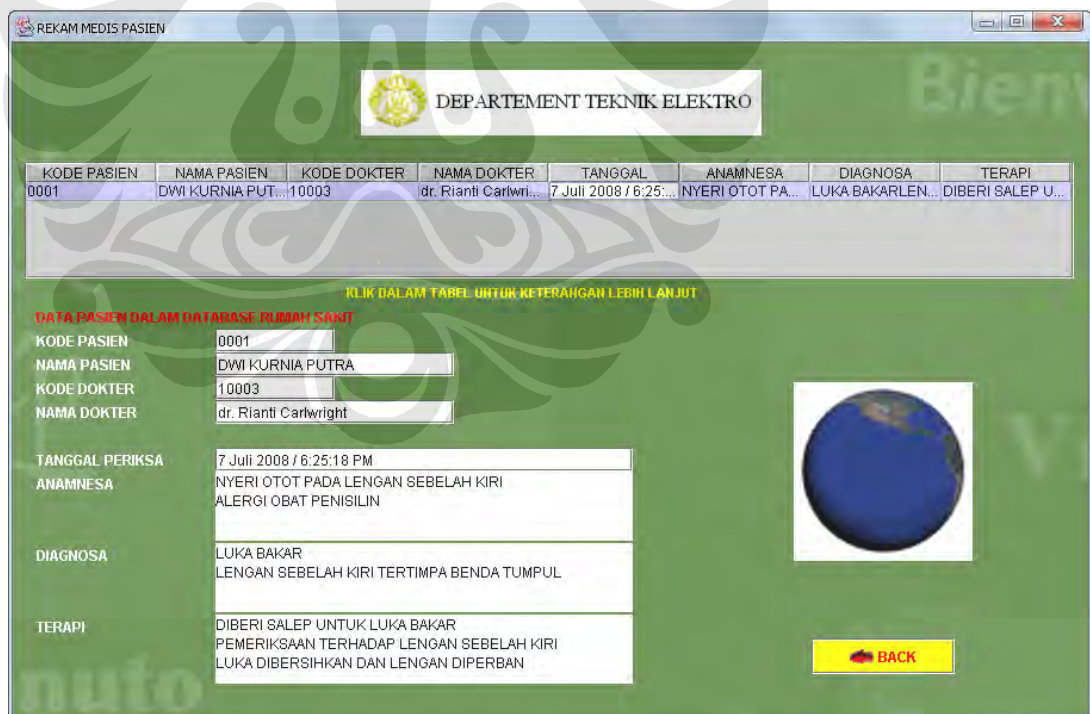
Gambar L8. Start Aplikasi Pasien



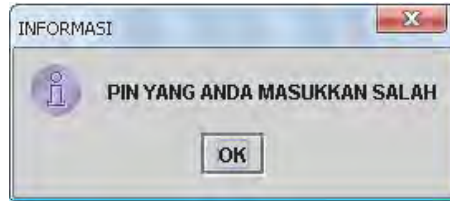
Gambar L9. Form Aplikasi Pasien



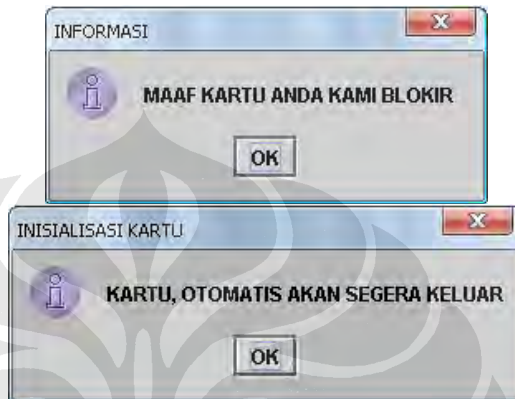
Gambar L10. Form Cek PIN



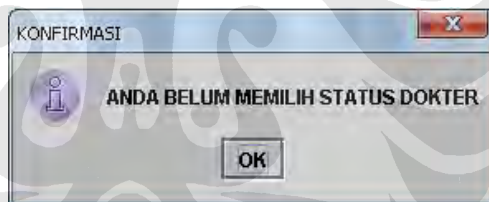
Gambar L11. Lihat Rekam Medis pada Aplikasi Pasien



Gambar L12. Pesan jika PIN tidak Sesuai



Gambar L13. Pesan jika PIN tidak Sesuai lebih dari 3 kali



Gambar L14. Kondisi Pasien ketika Belum Memilih Status Dokter

DATA UMUM PASIEN

DEPARTEMENT TEKNIK ELEKTRO

KODE DOKTER	NAMA DOKTER	STATUS DOKTER	TANGGAL PRAKTEK
10001	dr. Ade Rustanto	DOKTER UMUM	6 s/d 12 Juli 2008
10002	dr. Jefry Budiman	DOKTER UMUM	6 s/d 12 Juli 2008
10003	dr. Rianti Carlwright	DOKTER UMUM	6 s/d 12 Juli 2008

KLIK DALAM TABEL UNTUK KETERANGAN LEBIH LANJUT

**JADWAL PRAKTEK DOKTER UMUM SAKIT**

KODE DOKTER:

NAMA DOKTER:

TANGGAL:

TANGGAL PRAKTEK:

SENIN   
 SELASA   
 RABU   
 KAMIS   
 JUMAT   
 SABTU   
 MINGGU

PILIH KODE DOKTER:



DENAH RUANG PRAKTEK DOKTER

Gambar L15. Form Pilih Dokter

DATA UMUM PASIEN

DEPARTEMENT TEKNIK ELEKTRO

KODE DOKTER	NAMA DOKTER	STATUS DOKTER	TANGGAL PRAKTEK
10001	dr. Ade Rustanto	DOKTER UMUM	6 s/d 12 Juli 2008
10002	dr. Jefry Budiman	DOKTER UMUM	6 s/d 12 Juli 2008
10003	dr. Rianti Carlwright	DOKTER UMUM	6 s/d 12 Juli 2008

KLIK DALAM TABEL UNTUK KETERANGAN LEBIH LANJUT

**JADWAL PRAKTEK DOKTER UMUM SAKIT**

KODE DOKTER: 10001

NAMA DOKTER: dr. Ade Rustanto

TANGGAL: 7-6-2008

TANGGAL PRAKTEK: 6 s/d 12 Juli 2008

SENIN 08.00-12.00  
 SELASA 08.00-12.00  
 RABU 08.00-12.00  
 KAMIS LIBUR6  
 JUMAT 12.00-18.00  
 SABTU 12.00-18.00  
 MINGGU LIBUR

PILIH KODE DOKTER: 10001



RUANG PRAKTEK DOKTER BERATAP HIJAU  
SILAHKAN MENUNGGU DI RUANG TUNGGU

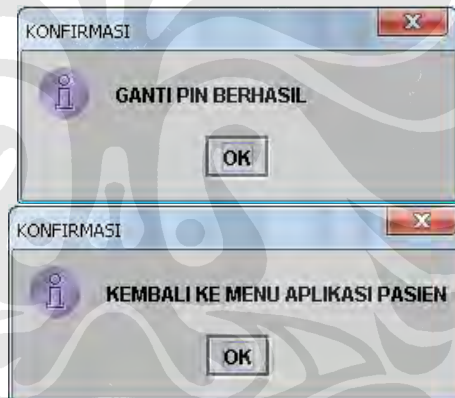
NO URUT PASIEN

Gambar L16. Kondisi mendapat Nomor Urut Pasien Setelah Mendaftar





Gambar L17. Form Ganti Pin



Gambar L18. Ganti PIN berhasil



Gambar L19. Ganti PIN tidak berhasil

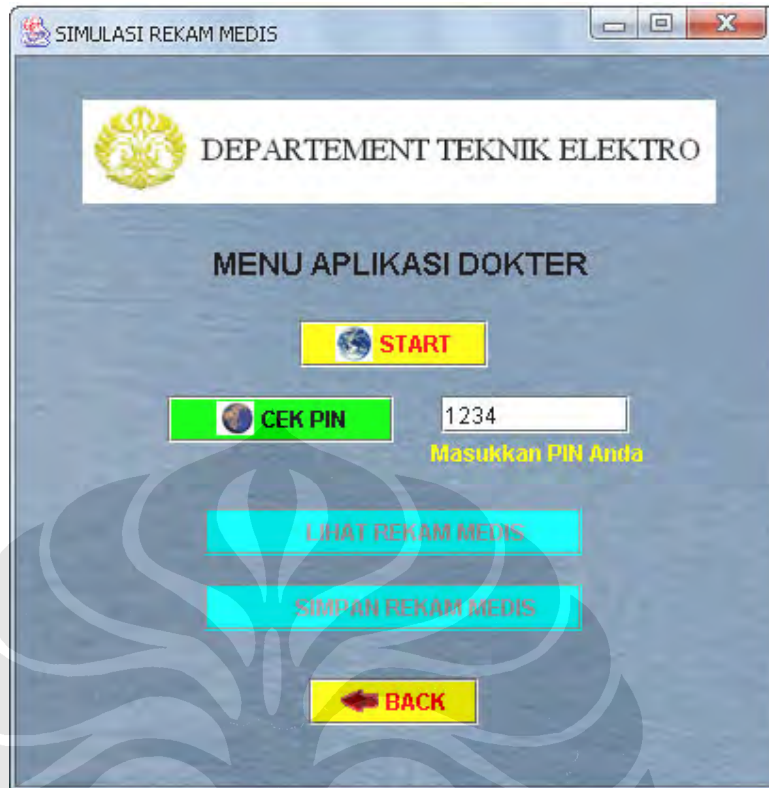
### III. Aplikasi Dokter



Gambar L20. Start Aplikasi Dokter



Gambar L21. Form Aplikasi Dokter




Gambar L22. Cek PIN Pasien pada Form Aplikasi Dokter



Gambar L23. Form Lihat Rekam Medis Pasien pada Aplikasi Dokter


**SIMULASI REKAM MEDIS**

 **DEPARTEMEN TEKNIK ELEKTRO**

**PENYIMPANAN REKAM MEDIS**

**KODE DOKTER**

**ANAMNESA**

 **DIAGNOSA**

**TERAPI**

**KONFIRMASI** **KONFIRMASI PENYIMPANAN REKAM MEDIS**

**TANGGAL PERIKSA**

**KODE PASIEN**

**NAMA PASIEN**

**KODE DOKTER**

**NAMA DOKTER**

**ANAMNESA**

**DIAGNOSA**

**TERAPI**

Gambar L24. Form Simpan Rekam Medis

**SIMULASI REKAM MEDIS**

**DEPARTEMEN TEKNIK ELEKTRO**

**PENYIMPANAN REKAM MEDIS**

**KODE DOKTER** 10003

**ANAMNESIS** NYERI OTOT PADA LENGAN SEBELAH KIRI  
ALERGI OBAT PENISILIN

**DIAGNOSA** LUKA BAKAR  
LENGAN SEBELAH KIRI TERTIMPA BENDA TUMPUL

**TERAPI** DIBERI SALEP UNTUK LUKA BAKAR  
Pemeriksaan terhadap lengan sebelah kiri  
Luka dibersihkan dan lengan diperban

**KONFIRMASI** **KONFIRMASI PENYIMPANAN REKAM MEDIS**

**TANGGAL PERIKSA** 7 Juli 2008 / 6:25:18 PM

**KODE PASIEN** 0001

**NAMA PASIEN** DWI KURNIA PUTRA

**KODE DOKTER** 10003

**NAMA DOKTER** dr. Rianti Carlwright

**ANAMNESIS** NYERI OTOT PADA LENGAN SEBELAH KIRI  
ALERGI OBAT PENISILIN

**DIAGNOSA** LUKA BAKAR  
LENGAN SEBELAH KIRI TERTIMPA BENDA TUMPUL

**TERAPI** DIBERI SALEP UNTUK LUKA BAKAR  
Pemeriksaan terhadap lengan sebelah kiri  
Luka dibersihkan dan lengan diperban

**CANCEL** **SAVE**

Gambar L25. Form Konfirmasi Simpan Rekam Medis

**PENYIMPANAN DATA PASIEN**

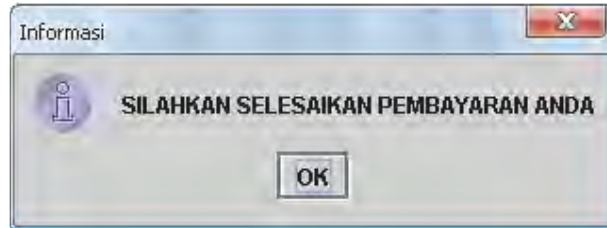
**DATA BERHASIL DISIMPAN DALAM DATABASE**

**OK**

**KEMBALI KE MENU APLIKASI DOKTER**

**OK**

Gambar L26. Simpan Rekam Medis Berhasil



Gambar L27. Konfirmasi Segera Melunasi Biaya Pengobatan setelah Berobat

#### IV. Aplikasi Bagian Keuangan



Gambar L28. Start Aplikasi Bagian Keuangan

The 'MENU BAGIAN KEUANGAN' (Financial Department Menu) form. The window title is 'SIMULASI REKAM MEDIS'. At the top, it says 'DEPARTEMENT TEKNIK ELEKTRO'. Below that is the menu title 'MENU BAGIAN KEUANGAN'. A green button labeled 'CEK PEMBAYARAN' (Check Payment) is at the top. Below it are several labels for data entry: 'TANGGAL BEROBAT', 'KODE PASIEN', 'NAMA PASIEN', 'KODE DOKTER', 'NAMA DOKTER', 'STATUS DOKTER', 'BEBAN PEMBAYARAN', and 'STATUS PEMBAYARAN'. At the bottom, there are three input fields for 'PEMBAYARAN OLEH PASIEN', 'BEBAN PEMBAYARAN PASIEN', and 'PENGEMBALIAN'. There are two buttons at the bottom: 'BACK' (yellow) and 'PROSES' (pink).

Gambar L29. Form Aplikasi Bagian Keuangan

SIMULASI REKAM MEDIS

DEPARTEMENT TEKNIK ELEKTRO

MENU BAGIAN KEUANGAN

CEK PEMBAYARAN

TANGGAL BEROBAT: 7-6-2008 / 10:46:8 AM

KODE PASIEN: 0001

NAMA PASIEN: DWM KURNIA PUTRA

KODE DOKTER: 10001

NAMA DOKTER: dr. Ade Rustanto

STATUS DOKTER: DOKTER UMUM

BEBAN PEMBAYARAN: 30000

STATUS PEMBAYARAN: BELUM LUNAS

PEMBAYARAN OLEH PASIEN:

BEBAN PEMBAYARAN PASIEN:

PENGEMBALIAN:

BACK PROSES

Gambar L30. Form Transaksi Pelunasan Biaya Pengobatan

TERIMA KASIH

BEBAN PEMBAYARAN LUNAS

OK

Gambar L31. Form Pelunasan Biaya Pengobatan