

Analisis dan Perancangan Sistem Personalisasi dan Identifikasi Absensi Dosen Berbasis *Smartcard Mifare* Menggunakan *Java Card Applet*

Nofita Rismawati¹, Muhamad Femy Mulya²

¹Program Studi Informatika, Universitas Indraprasta PGRI, Jakarta, Indonesia
novi.9001@gmail.com

²Program Studi Sistem Informasi, Tanri Abeng University, Jakarta, Indonesia
femy.mulya@tau.ac.id

Diterima : 01 Maret 2019

Disetujui : 20 April 2019

Abstract— *Smartcard* is an invention in the field of IT that has simplified human life because it has many functions and capabilities to store data. The advantages of a *smartcard* are rewritable data, extensive data capacity, support for reading multiple data carriers (*support for multiple tag reads*), robust physicality and smart task. The purpose of this study is to analyze and design a system of personalization and identification for the attendance of lecturers by utilizing *Mifare's smartcard* technology using a *java card applet* that can facilitate the lecture attendance process on a campus. Besides that, it also helps interested parties (in this case, campus management) in evaluating the performance of a lecturer. This study uses the *Prototype / Prototyping* model research method with the *Unified Modeling Language* (UML) approach. The results of this study are a prototype design personalization system and identification of lecturer attendance based on *Mifare smartcard* technology using *java card applets*, as well as the results of testing the accuracy of *prototype* readings from the *Mifare smartcard* personalization and identification system with *smartcard readers* using *Java card applets*.

Index Term—*Smartcard, Java Card Applet, Prototype*

I. PENDAHULUAN

Pencatatan kehadiran sering dikenal dengan istilah presensi. Pada awalnya, kebutuhan absensi dipenuhi secara manual semisal dengan menulis di buku catatan. Hal ini hanya efektif dilakukan jika individu yang akan diabsen berjumlah sedikit. Pada penerapan nyata saat ini, kebanyakan lingkungan memiliki banyak item yang perlu dicatat, sehingga absensi manual sulit diandalkan. Otomasi pun mulai banyak dilakukan seiring dengan penemuan-penemuan baru di bidang teknologi. Dengan berjalannya waktu, pada beberapa tahun terakhir sistem absensi dengan memanfaatkan sistem identifikasi *biometric* banyak digunakan, identifikasi ini mengandalkan elemen yang terintegrasi dalam tubuh pengguna semisal sidik jari. Integritas data yang tinggi, yaitu pendataan tidak bisa dititipkan, menjadi keunggulan utama sistem absensi ini. Salah satu kekurangan sistem ini yang cukup mengganggu adalah waktu yang dibutuhkan untuk mengenali sidik jari, satu kali

identifikasi memakan waktu 5 hingga 10 detik. Pada kebanyakan kasus, pengguna harus meletakkan jarinya pada sensor beberapa kali karena sistem kesulitan mengenali sidik jari dari database yang dimiliki. Kesulitan ini biasanya terjadi karena jari pengguna atau sensor yang kotor. Kesulitan ini sering membuat antrian panjang orang-orang yang harus melakukan absensi.

Dari beberapa teknologi identifikasi, *smartcard* menjadi teknologi yang menarik perhatian saat ini. Seperti halnya penemuan-penemuan lain, *smartcard* memiliki kelebihan dan kekurangan. Kelebihan-kelebihan yang dimiliki *smartcard* adalah data dapat ditulis ulang (*rewritable data*), kapasitas data yang luas, mendukung pembacaan banyak pembawa data (*support for multiple tag reads*), fisik yang kokoh dan dapat melakukan tugas pintar (*smart task*). Dalam penerapannya, *smartcard* akan diintegrasikan dengan perangkat lunak yang mengelola data hasil identifikasi.

Java Card adalah suatu generasi bahasa pemrograman *smartcard* yang baru. Dimana suatu *java applet* ditulis dalam bahasa tingkat tinggi, dikompilasi kedalam *byte code*, yang selanjutnya akan disimpan kedalam EEPROM, dan dijalankan pada suatu *smartcard*. Hal lain yang membedakan dengan kartu dari generasi sebelumnya adalah kemampuan *smartcard* untuk menampung lebih dari satu aplikasi (*applet*) dan menambah atau menghapus *applet* ke dan dari *smartcard*. Penelitian ini penting dilakukan, karena pada penelitian-penelitian yang sudah dilakukan tidak ada parameter uji coba akurasi pembacaan sistem personalisasi dan identifikasi *smartcard mifare* dengan *smartcard reader (Smartcard Reader OMNIKEY 3121)* dalam bentuk persentase.

Dari uraian yang telah diberikan, melalui penelitian ini akan menjawab permasalahan personalisasi dan identifikasi kartu *smartcard* untuk absensi dosen. Perangkat lunak ini diharapkan dapat membantu pihak yang berkepentingan (dalam hal ini, pihak manajemen kampus) dalam mengambil keputusan berkaitan dengan penilaian kinerja dosen. Perangkat lunak akan dikembangkan menggunakan bahasa pemrograman JSP (*Java Server Page*).

II. LANDASAN TEORI

A. Pengertian Smartcard

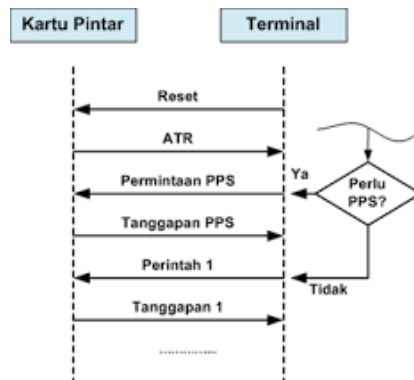
Smartcard adalah suatu miniatur komputer yang memiliki *Input/Output* yang terbatas (tidak ada *keyboard*, *mouse*, layar, *harddisk*, dan seterusnya) dan juga *resource-resource* yang terbatas pula (10MHz, 16K ROM, 1K RAM, 16K EEPROM)[1]. Teknologi ini di negara-negara maju sudah banyak digunakan untuk membangun suatu sistem-sistem berteknologi multiguna, contohnya adalah *bank card*, *mobile phone SIM*, *TV digital membership*, dan seterusnya.

Pada dasarnya, *smartcard* adalah suatu *microprocessor* yang mampu menyimpan dan memproses informasi secara aman. Hal ini yang membedakan antara *smartcard* dengan jenis-jenis kartu lainnya, seperti kartu *magnetic stripe* dan *memory-only chip*. *Smartcard* dianggap lebih aman karena informasi dan program pada suatu *smartcard* tidak dapat (dengan mudah) dibaca atau diubah. Dengan *smartcard*, pengguna dapat menyimpan kunci *cryptology* sehingga *smartcard* mampu untuk melakukan enkripsi maupun dekripsi

terhadap data yang dipertukarkan. Dengan demikian *smartcard* memungkinkan untuk mewujudkan suatu aplikasi yang canggih dan *portable* dalam pemrosesan data.

B. Transmisi Data Smartcard

Proses transmisi data antara *smartcard* dengan *smartcard reader*-nya terjadi secara *Half Duplex*, dengan demikian proses pengiriman dan penerimaan data dilakukan bergantian. Setelah kartu dimasukkan kedalam *slot* pada *reader* maka proses pertama yang terjadi adalah *reader* me-*reset* kartu, setelah kartu ter-*reset* maka kartu akan mengirimkan kode ATR (*Answer To Reset*) yang berisi beberapa parameter kartu untuk diperiksa oleh *reader*, setelah kode ATR diterima dan dimengerti oleh *reader*, maka *reader* akan mengirimkan perintah pertama pada kartu dan kartu akan mengirimkan respon dari perintah tersebut. Pada selang waktu antara kode ATR diterima dan perintah pertama dikirim, jika diperlukan *reader* juga dapat mengirim PPS (*Protocol Parameter Selection*). Proses-proses diatas dapat diilustrasikan dalam gambar berikut.



Gambar 1. Transmisi Data *Smartcard* [1].

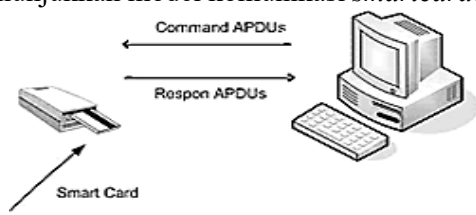
C. Format APDU

Smartcard tidak berarti tanpa adanya *smartcard reader*, yang berfungsi sebagai perantara komunikasi antara *smartcard* dengan peralatan lain seperti komputer. Komputer membaca atau menulis data melalui *smartcard reader*, kemudian *smartcard reader* mengubah perintah membaca/menulis tersebut ke dalam bahasa yang dimengerti *smartcard*. Masing-masing perusahaan menyediakan protokol yang berbeda untuk berkomunikasi dengan *smartcard* *reader*.

Komunikasi dengan *smartcard* berdasarkan format APDU (*Application Protocol Data Unit*). APDU merupakan unit dasar untuk pertukaran paket di dalam *smartcard*. Komunikasi antara kartu dengan *smartcard reader* dilakukan dengan APDU. APDU dinyatakan sebagai data paket yang berisi perintah lengkap atau respon yang lengkap dari kartu. Untuk menyediakan fungsionalitas seperti ini, APDU mendefinisikan struktur yang didefinisikan dalam beberapa dokumen ISO 7816 [3].

ISO mendefinisikan standar bagaimana aplikasi berkomunikasi dengan *smartcard*. Sayangnya, ISO tidak mendefinisikan standar untuk berkomunikasi dengan *smartcard reader*. Sehingga untuk mengirim perintah ke kartu, pertama pemrogram perlu menemukan *command* yang dimengerti oleh kartu, kemudian membungkus *command* tersebut dengan ISO *command* paket, kemudian dibungkus lagi dengan pembungkus yang diperlukan oleh *smartcard reader*.

Model *master-slave* digunakan di mana *smartcard* selalu memainkan posisi yang pasif. Dengan kata lain, *smartcard* selalu menunggu perintah APDU dari terminal. Kemudian *smartcard* mengeksekusi aksi yang ditentukan didalam APDU dan mengembalikannya ke terminal dengan *response APDU*. *Command APDU* dan *response APDU* dipertukarkan antara kartu dan terminal. Gambar 2 menunjukkan model komunikasi *smartcard*.



Gambar 2. Model Komunikasi *Smartcard*

Gambar 3 dan Gambar 4 di bawah adalah format *command* dan *response APDU*. Struktur APDU didefinisikan dalam ISO 7816-4.

Command APDU							
Mandatory Header				Conditional Body			
CLA	INS	P1	P2	Lc	Data field	Le	

Gambar 3. *Command APDU*

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*. Di beberapa *smart card* digunakan untuk mengidentifikasi aplikasi.
2. INS: *Instruction byte*. *Byte* ini menyatakan kode instruksi/perintah.
3. P1 dan P2: *Parameter byte*. Menyediakan kualifikasi lebih lanjut untuk perintah APDU.
4. *Conditional body* terdiri dari 3 *field*, yaitu Lc, *data field* dan Le.
5. Lc menyatakan jumlah *byte* di dalam *data field* dari *command APDU*.
6. *Data field* menyatakan data yang diperlukan oleh *command APDU*.
7. Le menyatakan jumlah maksimal dari *byte* yang diharapkan di dalam *data field* dari respon APDU.

Response APDU		
Conditional Body	Mandatory Trailer	
Data field	SW1	SW2

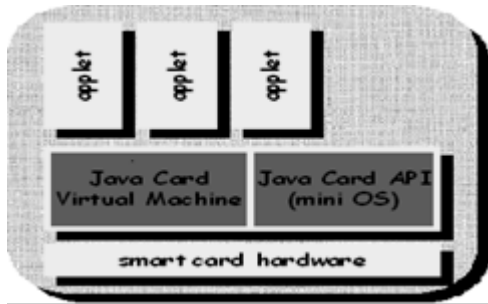
Gambar 4. *Response APDU*

Response APDU terdiri dari *conditional body* dan *mandatory trailer*.

1. *Conditional body* berisi *data field* yang menyatakan data yang diperlukan oleh *response APDU*.
2. *Mandatory trailer* terdiri dari *status byte* SW1 dan SW2 menyatakan status proses dari *command APDU* di dalam kartu.

D. Java Card

Java Card adalah suatu generasi bahasa pemrograman *smartcard* yang baru. Dimana suatu *java applet* ditulis dalam bahasa tingkat tinggi, dikompilasi kedalam *byte code*, disimpan kedalam EEPROM, dan dijalankan pada suatu *smartcard*. Hal lain yang membedakan dengan kartu dari generasi sebelumnya adalah kemampuan *smartcard* untuk menampung lebih dari satu aplikasi (*applet*) dan menambah atau menghapus *applet* ke dan dari *smartcard* [4].



Gambar 5. Arsitektur Sistem Java Card [4].

Gambar 5 menunjukkan sistem arsitektur Java Card. Java card merupakan subset dari Java dengan sejumlah keterbatasan, yaitu:

1. Tidak ada *real*, *double*, *string*, dan multi-dimensi *array*.
2. Tidak ada *threads*.
3. Ops untuk melakukan *garbage collection*.

Java Card menggunakan 16 bit aritmatika dan *class file* yang telah dioptimisasi, yang disebut *cap-files*. API untuk Java Card mendukung I/O *smartcard* dengan menggunakan APDU's menggunakan ISO 7816. API Java Card juga mendukung *persistensi*, *transient data* dan *transaction*.

E. Java Card Applet

Java Card applet adalah program java mengikuti sebuah set konvensi yang memungkinkan itu untuk berjalan dalam Java Card runtime environment. Java Card applet tidak ditujukan untuk berjalan dalam lingkungan browser. Nama applet dipilih untuk aplikasi Java Card adalah karena applet Java Card dapat di-load ke dalam Java Card runtime environment setelah kartu di-manufaktur. Tidak seperti aplikasi pada banyak sistem *embedded*, applet tidak perlu untuk di-*burned* ke dalam ROM selama manufaktur, tetapi dapat secara dinamis di *download* ke kartu pada saat berikutnya [4].

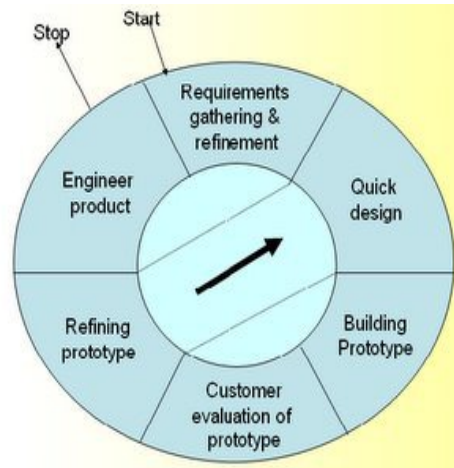
Sebuah *applet* harus diperluas dari `javacard.framework.applet.class`. Kelas *base applet* merupakan *superclass* untuk semua *applet* yang disimpan dalam *smartcard*. Kelas *applet* merupakan *blueprint* yang mendefinisikan variabel dan metoda dari sebuah *applet*. *Applet* yang berjalan dalam kartu merupakan *instance* dari sebuah *applet* sebuah objek dari kelas *applet*. Seperti objek *persisten* lainnya, satu kali di-*create*, sebuah *applet* hidup dalam *applet* selamanya.

Java Card runtime environment mendukung lingkungan multiaplikasi. *Multiple applet* dapat hidup bersama dalam sebuah *java smartcard*, dan sebuah *applet* dapat memiliki *multiple instances*.

Contoh: satu *wallet applet instance* dapat di-*create* untuk mendukung *U.S dollar* dan lainnya dapat di-*created* untuk *British pound*.

III. METODE PENELITIAN

A. Metode Penelitian



Gambar 6. Model Prototype/Prototyping

Dalam menganalisa dan merancang sistem personalisasi dan identifikasi absensi dosen berbasis *smartcard mifare* menggunakan *java card applet*, maka pada penelitian ini mengikuti model *Prototype/Prototyping* [2]. Adapun tahapan *Prototyping* pengembangan sistem yang terdiri dari:

1. Pengumpulan Kebutuhan.
Pelanggan dan pengembang bersama-sama mendefinisikan format dan kebutuhan keseluruhan perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. Membangun *Prototype/Prototyping*.
Membangun *prototyping* dengan membuat perancangan sementara yang berpusat pada penyajian kepada pelanggan (misalnya dengan membuat *input* dan contoh *output*-nya).
3. Evaluasi *Prototyping*.
Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah keempat akan diambil. Jika tidak, maka *prototyping* diperbaiki dengan mengulang langkah 1, 2, dan 3.

4. Mengkodekan Sistem.
Dalam tahap ini *prototyping* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.
5. Menguji Sistem.
Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *White Box, Black Box, Basis Path*, pengujian arsitektur dan lain-lain.
6. Evaluasi Sistem.
Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika sudah, maka langkah ketujuh dilakukan, jika belum maka mengulangi langkah 4 dan 5.
7. Menggunakan Sistem.
Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

B. Metode Analisa Data Kuantitatif

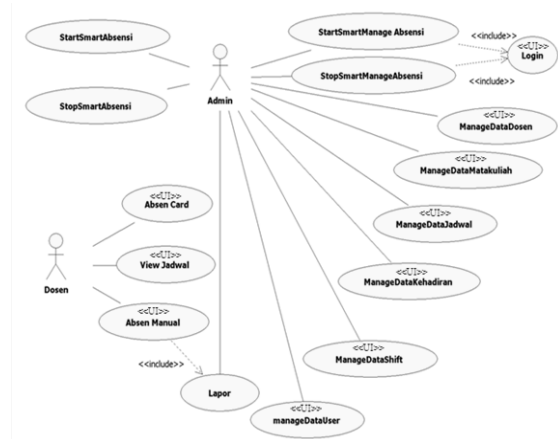
Pada metode kuantitatif dilakukan menggunakan metode penelitian eksperimental, yaitu dengan melakukan eksperimen terhadap variabel-variabel *input* untuk menganalisis *output* yang dihasilkan. Penelitian Eksperimental merupakan bentuk penelitian dimana peneliti (*eksperimenter*) dengan sengaja menguji coba objek yang terdapat pada perangkat lunak personalisasi dan identifikasi absensi dosen berbasis *smartcard mifare* menggunakan *java card applet* (uji coba akurasi pembacaan sistem personalisasi *smartcard* dengan *smartcard reader*), selanjutnya mengamati dan mencatat hasil ujicoba yang dilakukan, kemudian melihat hubungan diberikan dan reaksi yang muncul dari Proses tersebut [2].

IV. HASIL DAN PEMBAHASAN

A. Diagram Sistem

Fungsi utama dari sistem ini adalah melakukan aktifitas absensi oleh dosen. Berikut ini merupakan diagram yang menjelaskan secara singkat mengenai alur dan kerja dari aplikasi sistem absensi dosen berbasis *smartcard mifare* menggunakan *Java Card Applet*:

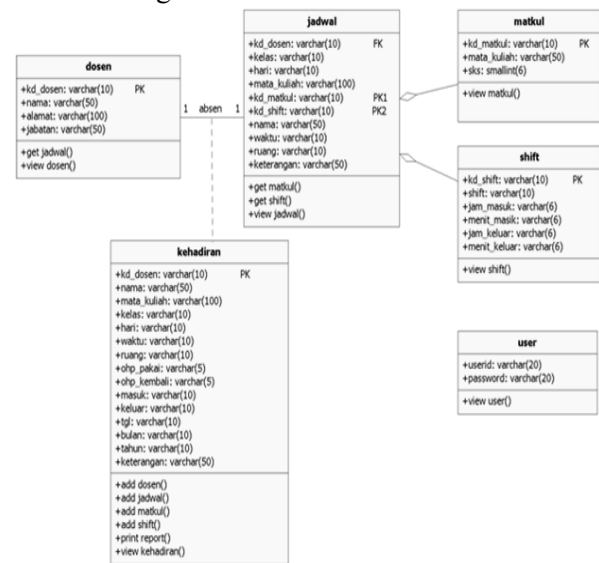
1. Usecase Diagram



Gambar 7. Use Case Diagram

Pada gambar *usecase* diagram diatas menjelaskan secara menyeluruh tentang aktifitas yang dapat dilakukan oleh aktor pada sistem absensi ini.

2. Class Diagram

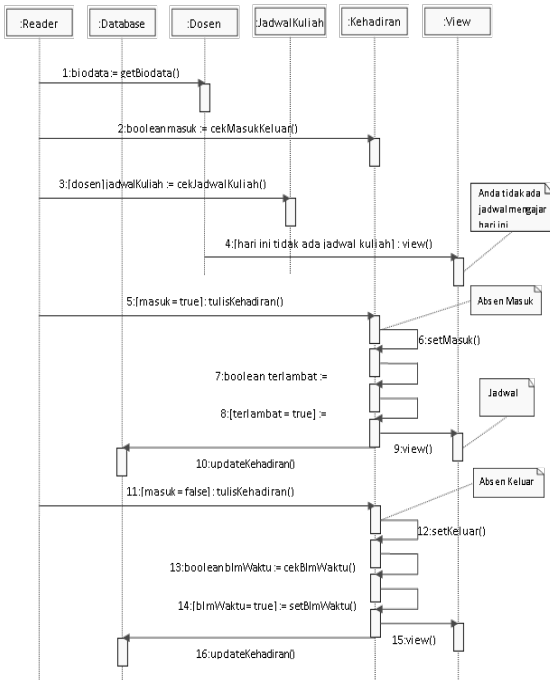


Gambar 8. Class Case Diagram

Pada gambar *Class Diagram* diatas menjelaskan tentang hubungan antar table pada *database* yang digunakan oleh sistem absensi ini serta atribut apa saja yang terdalem didalamnya.

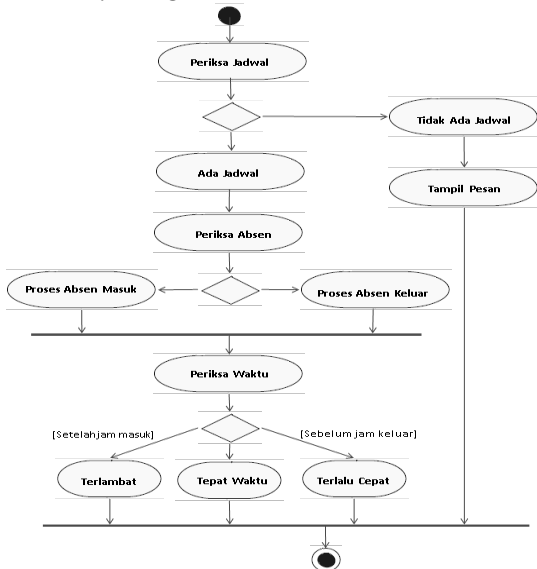
3. Sequence Diagram

Pada diagram *sequence* diatas menggambarkan *Sequence* ketika dosen melakukan proses absensi masuk (*check-in*) dan absensi keluar (*check-out*), ditunjukkan pada gambar 9.



Gambar 9. Sequence Diagram Absensi

4. Activity Diagram



Gambar 10. Activity Diagram Absensi

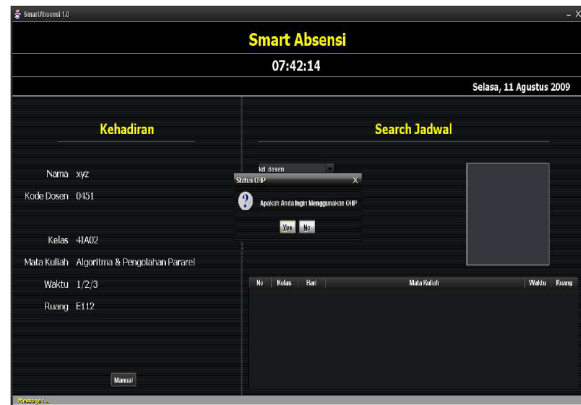
Pada diagram activity diatas menggambarkan Activity ketika dosen melakukan proses absensi masuk (check-in) dan absensi keluar (check-out) dengan status absensi yaitu: Terlambat, Tepat Waktu dan Terlalu Cepat.

B. Hasil

Berikut ini merupakan hasil dari pengujian aplikasi Sistem Personalisasi dan Identifikasi Absensi Dosen Berbasis Smartcard Mifare Menggunakan Java Card Applet:

1. Antarmuka Halaman Smart Absensi (Check-In).

Pada Gambar 11, merupakan tampilan pada saat dosen melakukan absensi sebelum mengajar (check-in). Untuk melakukan absensi dosen hanya perlu memasukkan smartcard ke dalam smartcard reader, maka akan ditampilkan identitas dosen pada ruas kiri (kehadiran) berupa nama, kode dosen, kelas, mata kuliah, waktu dan ruang. Apabila telah selesai maka akan muncul messagebox penggunaan OHP/Proyektor. Jika setuju maka tekan command yes dan jika tidak tekan command no. Berikut adalah antarmuka halaman Smart Absensi (Check-In) pada aplikasi ini:

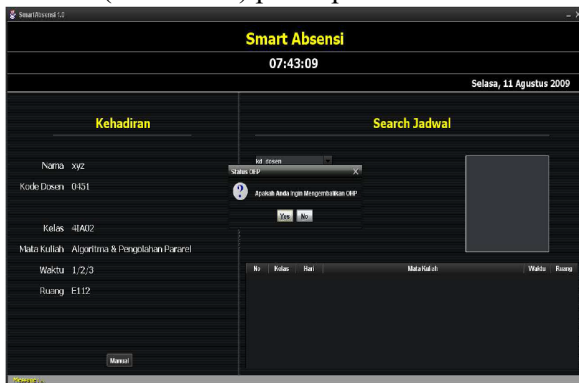


Gambar 11. Antarmuka Halaman Smart Absensi (Check-In)

2. Antarmuka Halaman Smart Absensi (Check-Out).

Pada Gambar 12, merupakan tampilan pada saat dosen melakukan absensi setelah mengajar (check-out). Untuk melakukan proses check-out dosen hanya perlu memasukkan kembali smartcard ke dalam smartcard reader, maka akan ditampilkan identitas dosen pada ruas kiri (kehadiran) berupa nama, kode dosen, kelas, mata kuliah, waktu dan ruang. Apabila telah selesai maka akan muncul messagebox pengembalian OHP/Proyektor. Jika setuju maka tekan command yes dan jika tidak tekan command no. Setelah itu, akan muncul Terima Kasih pada bagian ruas kiri bawah.

Berikut adalah antarmuka halaman *Smart Absensi (Check-Out)* pada aplikasi ini:



Gambar 12. Antarmuka Halaman *Smart Absensi (Check-Out)*

3. Antarmuka Halaman *Smart Manage Absensi*

Pada Gambar 13, merupakan tampilan utama *smart manage absensi*. Pada tampilan ini, data yang di manage berupa data dosen, mata kuliah, jadwal, kehadiran, shift dan user. Berikut adalah antarmuka halaman *smart manage absensi* pada aplikasi ini:



Gambar 13. Antarmuka Halaman *Smart Manage Absensi*

4. Hasil Pengujian *BlackBox*

Pengujian terhadap sistem yang dibuat dilakukan dengan metode pengujian *black box* pada aplikasi sistem personalisasi dan identifikasi absensi dosen berbasis *smartcard mifare* menggunakan *java card applet*. Pengujian *black box* ini penting dilakukan, karena pada pengujian ini, dapat mengidentifikasi kesalahan pada persyaratan fungsional dengan mengabaikan mekanisme internal atau komponen dari suatu aplikasi. Berikut ini tabel dari hasil pengujian *BlackBox*:

Tabel 1. Hasil Pengujian *BlackBox*

No	Test Case	Skenario Uji	Hasil yang Diharapkan	Hasil Uji
1	Menampilkan halaman <i>login smart absensi</i>	Melakukan <i>login</i> pada aplikasi	Menampilkan isi dari halaman utama <i>smart</i>	Berhasil

			absensi	
2	Membuka menu <i>install smartcard</i>	Melakukan proses <i>install</i> kartu <i>smartcard</i>	Menampilkan isi dari halaman <i>install smart absensi</i>	Berhasil
3	Membuka menu <i>uninstall smartcard</i>	Melakukan proses <i>uninstall</i> kartu <i>smartcard</i>	Menampilkan isi dari halaman <i>uninstall smart absensi</i>	Berhasil
4	Membuka halaman absensi masuk (<i>check-in</i>)	Melakukan proses <i>read data smartcard (check-in)</i>	Menampilkan isi dari halaman menu <i>check-in</i>	Berhasil
5	Membuka halaman absensi keluar (<i>check-out</i>)	Melakukan proses <i>read data smartcard (check-out)</i>	Menampilkan isi dari halaman menu <i>check-out</i>	Berhasil
6	Membuka halaman <i>save data smartcard</i>	Melakukan proses <i>save data smartcard</i>	Menampilkan isi dari halaman manu <i>save data to smartcard</i>	Berhasil
7	Membuka halaman <i>delete data smartcard</i>	Melakukan proses <i>delete data smartcard</i>	Menampilkan isi dari halaman manu <i>delete data from smartcard</i>	Berhasil
9	Menampilkan halaman <i>login smart manage absensi</i>	Melakukan proses <i>login</i> pada aplikasi	Menampilkan isi dari halaman utama <i>smart manage absensi</i>	Berhasil
10	Membuka halaman <i>smart management data dosen dan absensi</i>	Melakukan proses manajemen data dosen (<i>input, update dan delete</i>)	Menampilkan isi dari halaman <i>management data dosen dan absensi</i>	Berhasil

5. Uji Coba Akurasi Pembacaan *smartcard* dengan *smartcard Reader*.

Pada penelitian ini dikembangkan suatu *prototype* untuk sistem personalisasi dan identifikasi absensi dosen berbasis *smartcard mifare*, sehingga dapat diketahui persentase tingkat akurasi pembacaan *process smartcard access* untuk setiap prosesnya yang terdiri dari, proses *install smartcard, uninstall smartcard, read data smartcard, save data smartcard* dan *delete data smartcard*.

Untuk menghitung persentase akurasi Pembacaan Sistem Personalisasi dan identifikasi *smartcard* dengan *smartcard Reader*, dilakukan uji coba *sampling* pembacaan *process smartcard access* yang masing-masing sebanyak 100 kali *sampling* pembacaan data untuk setiap prosesnya, maka dari hasil uji coba didapat hasil rata-rata persentase tingkat akurasi pembacaan *process smartcard access* sebagai berikut [2]:

Tabel 2. Hasil Uji Coba Akurasi Pembacaan *Smartcard* dengan *Smartcard Reader*.

Process <i>Smartcard Access</i>	Uji Coba sebanyak 100 kali Percobaan		Total Persentase	
	Total Sukses	Total Gagal	Persentase Total Sukses	Persentase Total Gagal
Proses <i>Install Smartcard</i>	97	3	97%	3%
Proses <i>Uninstall Smartcard</i>	96	4	96%	4%
Proses <i>Read Data Smartcard</i>	99	1	99%	1%
Proses <i>Save Data Smartcard</i>	95	5	95%	5%
Proses <i>Delete Data Smartcard</i>	94	6	94%	6%
Rata-Rata Pembacaan Process <i>Smartcard Access</i>	96,20	3,80	96,20%	3,80%

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Tujuan utama dari aplikasi sistem personalisasi dan identifikasi absensi dosen berbasis *smartcard mifare* menggunakan *java card applet* adalah untuk menggantikan sistem absensi dosen secara manual, sehingga akan mengurangi kecurangan, dan meningkatkan tingkat keefektifitasan pada sistem absensi dosen pada suatu kampus. Sistem absensi yang dibuat ini dapat memecahkan sebagian besar permasalahan dari tujuan di atas dengan beberapa keunggulan, yaitu:

1. Perancangan Aplikasi sistem personalisasi dan identifikasi absensi dosen berbasis *smartcard mifare* menggunakan *java card applet* telah dibuat sesuai dengan perancangan yang sudah di buat.
2. Hasil pengujian aplikasi menggunakan metode pengujian *black box* pada sistem menunjukkan bahwa sistem mempunyai tingkat fungsionalitas yang baik, semua fungsi yang berada pada sistem dapat bekerja sangat baik. Sistem sudah dapat memenuhi spesifikasi kebutuhan.
3. Berdasarkan hasil uji coba pembacaan sistem personalisasi dan identifikasi *smartcard mifare* dengan *smartcard reader*

yang dilakukan sebanyak 100 kali uji coba, maka diperoleh rata-rata pembacaan *process smartcard access* sebesar 96,20% untuk persentasi total sukses dan 3,80% untuk persentase total gagal. Dari perbandingan persentase rata-rata pembacaan *process smartcard access*, bisa dikatakan hasilnya akurat, dan patut dijadikan bahan pertimbangan untuk kampus dalam mengimplementasikan sistem ini.

B. Saran

Berdasarkan hasil dari penelitian yang telah dilakukan oleh peneliti, berikut ini beberapa saran untuk pengembangan lebih lanjut pada aplikasi system personalisasi dan identifikasi absensi dosen berbasis *smartcard mifare* menggunakan *java card applet*:

1. Sistem pendataan *smartcard* dapat diganti dengan teknologi identifikasi lain jika integritas absensi (masalah penitipan absen) merupakan masalah yang kritis. *smartcard reader* dapat diganti dengan sensor *fingerprint* atau sensor-sensor lain. *Software* cukup melakukan modifikasi minor dan sistem akan berjalan hampir tanpa perubahan signifikan.
2. Sebagai bahasa pemrograman yang menekankan netralitas arsitektur, penulis sangat mendorong untuk dilakukan penelitian lanjutan untuk melakukan perubahan sistem ini ke sistem operasi lain semisal Linux.
3. Mengimplementasikan sistem *smartcard reader* yang lebih canggih sehingga dapat menghilangkan masalah kemungkinan penitipan absen (data).

DAFTAR PUSTAKA

- [1] Fakhruddin, R. and Harsoyo, A. *Implementasi Portable Smartcard Reader untuk Absensi*. In, Konferensi Nasional Teknologi Informasi dan Komunikasi untuk Indonesia, Aula Barat Institut Teknologi Bandung: 2006
- [2] Rismawati, Nofita and Femy, Muhamad. *Analisis dan Perancangan Sistem Personalisasi dan Monitoring SLA (Service Level Agreement) Berbasis NFC (Near Field Communication) Studi Kasus Akses Kontrol Vendor Ke Perangkat BTS (Base Transceiver Station)*. Jurnal Faktor Exacta. Vol. 11 No.4. Universitas Indraprasta PGRI. Jakarta: 2018.
- [3] ISO/IEC 7816, *Interindustry Commands for Interchange*, ISO/IEC: 1995.
- [4] Alfitri, Nadia. *Akses Database Menggunakan Java Card*. Tugas akhir, Jurusan Teknik Informatika, Fakultas Teknik, Institut Teknologi Bandung, Bandung: 2006.
- [5] A. Dayumi and M. Femy. *Sistem Absensi Karyawan Berbasis Location Based Services (LBS) Menggunakan Platform Android Studi Kasus: PT.NoXus Ideata Prima*. Jurnal Sistem Komputer & Kecerdasan Buatan. Vol. 2 No.1. Universitas Tanri Abeng. Jakarta: 2018.
- [6] Alfatwa, Fathony, D. *Kartu cerdas (smart card) dan hubungannya dengan kriptografi*. Tugas akhir, Jurusan Teknik Informatika, Fakultas Teknik, Institut Teknologi Bandung, Bandung: 2003.
- [7] Budiarto, A. *Otentikasi pada kartu pintar (smartcard authentication)*. Tugas akhir, Jurusan Teknik Komputer, Fakultas Teknik, Institut Teknologi Bandung, Bandung: 2003.
- [8] A. Rokhman, *Prospek dan tantangan penerapan e-voting di indonesia*, in Seminar Nasional Peran Negara dan Masyarakat dalam Pembangunan dan Masyarakat Madani di Indonesia: 2011, pp. 1–11.
- [9] Larman, C. *Applying UML and Patterns 3rd Edition*. Addison-Wesley, 2004.
- [10] Lamida, J. K. *Pengembangan perangkat lunak pengelola data kehadiran*. Tugas akhir, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta, Jakarta: 2008.
- [11] Sommerville, Ian. *Software Engineering (Rekayasa Perangkat Lunak)*. Erlangga. Jakarta: 2011.
- [12] McGraw, G. and Felten, E. *How Smart Card and Java Mix*. Willey: 1999.