

SISTEM SMART CLASS ROOM BERBASIS SMART CARD DAN BAHASA PEMROGRAMAN C++

Gintoro, Adrian Tjahjana, Dedy Irawan, William

ABSTRAKS

Tujuan penulisan skripsi ini adalah menganalisa dan merancang sebuah sistem pengelolaan sumber daya kelas pada Universitas Bina Nusantara yang terintegrasi dengan sistem Smart Class Room. Metodologi analisa yang dipergunakan dalam perancangan sistem meliputi tiga bagian pokok yaitu metode analisis, perancangan, dan studi pustaka. Metode analisis meliputi survei atas sistem yang sedang berjalan, analisis terhadap hasil survei, dan identifikasi kebutuhan informasi. Metode perancangan menggunakan pendekatan Object Oriented Analysis and Design (OOAD) dengan notasi Unified Modelling Language (UML). Hasil yang dicapai adalah mengoptimalkan waktu persiapan alat didalam ruang kelas dan adanya pengontrolan ruang kelas yang dilakukan oleh pihak akademis. simpulan yang didapat dari penulisan skripsi ini adalah sistem dapat mempercepat waktu untuk melakukan persiapan kelas yang dilakukan dosen, mengurangi terjadinya kesalahan yang dilakukan oleh kelalaian manusia, dan mempersingkat proses bisnis yang dilakukan seperti pemeriksaan ruang kelas sedang digunakan atau tidak serta keadaan alat didalam ruang kelas.

Kata Kunci - Sumber daya kelas, Smart Card, C++ , programming language.

I. PENDAHULUAN

Dengan semakin berkembangnya proses pembelajaran di sebuah universitas maka dibutuhkan pula sebuah sistem untuk mendukung proses pembelajaran tersebut. Sistem tersebut haruslah dapat digunakan untuk menyelesaikan berbagai persoalan yang sering muncul seiring dengan semakin berkembangnya proses pembelajaran. Oleh karena itu diperlukannya sebuah sistem yang terintegrasi dengan baik untuk mendukung sebuah proses pembelajaran.

Universitas Bina Nusantara merupakan sebuah institusi pendidikan tinggi yang saat ini semakin berkembang sehingga membutuhkan sebuah sistem yang dapat mengakomodasi peningkatan kinerja pembelajaran. Salah satu bagian yang sedang dikembangkan oleh Universitas Bina Nusantara yaitu tentang pengelolaan sumber daya ruang kelas untuk mempercepat pengoperasian alat ketika proses perkuliahan dimulai. Saat ini, persiapan ruang kelas dan fasilitasnya seperti membuka dan menutup ruang kelas oleh petugas serta menyalakan atau mematikan fasilitas alat di dalam ruang kelas oleh dosen belum menggunakan teknologi yang terintegrasi secara otomatis. Karena persiapan ruang kelas masih dilakukan oleh manusia maka akan terdapat kemungkinan terjadinya berbagai kesalahan akibat kelalaian manusia itu sendiri yang mengakibatkan proses persiapan alat di dalam ruang kelas menjadi memakan waktu lama dan mengganggu proses perkuliahan.

Melihat hal tersebut maka perlu dikembangkan sebuah sistem pengelolaan sumber daya kelas yang terintegrasi dengan sistem absensi dosen yang sudah tersedia dan berbasis aplikasi.

Salah satu sistem yang dapat digunakan untuk memenuhi kebutuhan tersebut ialah *Smart Class*

Room. Smart Class Room merupakan sebuah sistem yang dipakai untuk mempermudah pengelolaan sumber daya kelas.

II. TINJAUAN PUSTAKA

A. Model TCP/IP Layer

Model TCP/IP merupakan protokol yang paling luas digunakan dalam arsitektur jaringan. TCP/IP merupakan hasil penelitian yang dibuat dan dikembangkan oleh DARPA (*Defence Advanced Research Project Agency*) yang digunakan pada jaringan paket. *Layer* TCP/IP dibagi menjadi 4 *layer*, yaitu:

1. *Host-to-Network Layer*

Layer ini berfungsi agar sebuah *host* bisa terhubung ke sebuah jaringan menggunakan protokol sehingga dapat mengirim paket IP melalui protokol tersebut.

2. *Internet Layer*

Layer ini berfungsi untuk mengijinkan *hosts* untuk mengirimkan paket ke jaringan mana pun dan paket berjalan secara independen ke tujuannya atau jaringan yang berbeda.

3. *Transport Layer*

Layer ini bertugas untuk mengkoordinasikan semua data yang diterima maupun data yang dikirim. *Layer* ini berfungsi sama dengan *transport layer* di OSI *Layer*.

4. *Application Layer*

Layer ini merupakan *layer* tertinggi yang digunakan oleh banyak aplikasi untuk komunikasi jaringan. *Layer* ini juga berisi segala aplikasi yang dimiliki oleh pengguna (Tanenbaum, 2003).

B. Pengertian Smart Card

Smart card adalah sebuah kartu yang terbuat dari plastik (PVC) yang digabung dengan *Integrated Circuit* (IC) yang berukuran sebesar kartu kredit dan

bersifat *tamper resistant*, yaitu usaha ilegal pengambilan data dari dalam kartu yang tidak dimungkinkan. *Smart card* didesain untuk menyimpan data yang bersifat pribadi dengan tingkat keamanan yang tinggi serta kartu mudah untuk dibawa kemana saja. Penyimpanan dan pemrosesan informasi dalam *smart card* dilakukan melalui sirkuit elektronik yang digabungkan dalam silikon pada bahan plastik (PVC) dari kartu.

Integrated Circuit (chip) berukuran sangat kecil dan *printed circuit* berbentuk plat emas yang tipis. *Printed circuit* ini memberikan kontak elektrik dengan lingkungan luar dan juga melindungi *chip* dari kerusakan mekanik dan gangguan listrik.

Untuk pemrosesan dan penyimpanan data yang aman digunakan *public-key* atau *shared-key algorithm*. Beberapa *smart card* memiliki *cryptographic coprocessors* yang terpisah dan mendukung algoritma seperti RSA (*Rivest-Shamir-Adleman's algorithm*), AES (*Advanced Encryption Standar*) dan Triple DES (*Triple Data Encryption Standard*).

Cara komunikasi yang dipakai teknologi *smart card* adalah *half duplex*, yaitu komunikasi dua arah, namun tidak dilakukan secara bersamaan. Fungsi ini bekerja dengan cara proses mengirim dan menerima harus dilakukan secara bergantian. Data yang dikirimkan dan diterima dari *smart card* disimpan dalam *buffer* yang terdapat dalam RAM *smart card* (Ortiz, 2003).

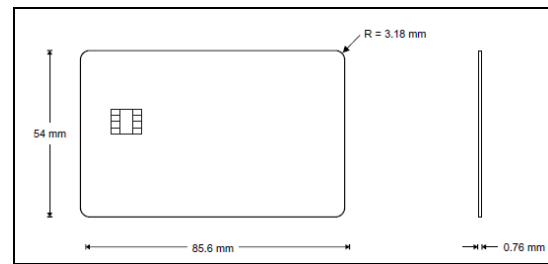
C. Standar Internasional Smart Card

Standar ISO/IEC merupakan suatu standar internasional untuk menentukan standarisasi dari *smart card*. Kepanjangan dari ISO adalah *International Organization for Standardization*, sedangkan kepanjangan dari IEC adalah *International Electrotechnical Commission* (Rankl, et al.,2003).

ISO/IEC 7816 dan ISO/IEC 7810 adalah sebuah standar untuk *smart card* untuk menentukan hal berikut ini:

1. Bentuk fisik dari kartu.
2. Karakteristik dari sirkuit elektronik.
3. Posisi dan ukuran kartu dari konektor elektrikal kartu.
4. Protokol komunikasi.
5. Ketahanan kartu.
6. Fungsionalitas kartu.

Ukuran kartu *smart card* dengan format ID-1 berdasarkan ISO/IEC 7810 adalah 85.60mm x 54mm dan memiliki ketebalan kartu 0.76mm ± 0.08mm serta jari-jari sudut 3.18mm (lihat Gambar 1) (Rankl, et al.,2003).



Gambar 1 Ukuran Standar Smart Card Format ID-1

D. Tipe Smart Card

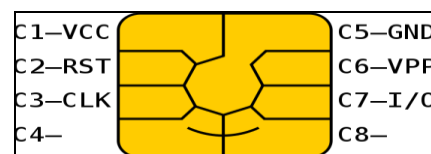
Secara umum ada dua macam tipe dari *smart card* yaitu *memory card* dan *microprocessor card*. *Memory card* hanya menyimpan dan memproteksi data secara lokal, namun tidak mengandung sebuah prosesor untuk melakukan perhitungan komputer pada data. Sedangkan *microprocessor card* adalah sebuah kartu yang memiliki memori dan *microprocessor* yang dapat melakukan kalkulasi pada data dan menyimpan data di dalam kartu secara aman.

Sebuah *memory card* dapat menyimpan data sebesar 4K dan keuntungan dari *memory card* terletak pada teknologinya yang sederhana. Sedangkan *microprocessor card*, menawarkan sekuritas dan kemampuan multifungsional. Data yang disimpan dalam *microprocessor card* tidak dapat diakses secara langsung melalui aplikasi di luar kartu. *Microprocessor* mengontrol data dan memori yang mengatur pengaksesan data menurut kondisi yang diberikan seperti *password*, enkripsi dan lainnya. Keuntungan *microprocessor card* adalah dapat diintegrasikan ke lebih dari satu aplikasi (Rankl, et al.,2003).

Secara fisik *smart card* dapat dibedakan menjadi dua jenis yaitu:

1. Contact Smart Card

Contact Smart Card bekerja dengan cara berkomunikasi secara fisik antara *card reader* dan *smart card pin contact* yang berbentuk segi empat berukuran ± 1 cm (Lihat gambar 2). *Contact smart card* tidak membutuhkan baterai dan akan aktif ketika terhubung dengan *card reader*. Saat terhubung dengan *card reader*, maka *chip* akan menunggu perintah *request* dari klien dari aplikasi untuk membaca informasi dari *chip* atau menulis informasi ke *chip*.



Gambar 2 Smart Card's Pin Contact

Berikut adalah fungsi dari masing-masing *contact* menurut ISO/IEC 7816, yaitu:

a. VCC (*Supply Voltage*)

Berfungsi sebagai penyedia tegangan listrik, tegangan listrik yang tersedia sebesar 3 atau 5 volt dengan toleransi maksimum $\pm 10\%$.

b. RST (*Reset Input*)

Berfungsi untuk mengirimkan sinyal untuk me-
reset microprocessor.

c. CLK (*Clock Input*)

Berfungsi sebagai *timing* atau *clocking signal* yang mengatur frekuensi waktu dan kecepatan *microprocessor*.

d. GND (*Ground*)

Merupakan *reference voltage* yang berlawanan dengan VCC di mana energi potensialnya diukur dan nilainya dianggap 0 volt.

e. VPP (*Programming Voltage*)

Berfungsi menyediakan tegangan listrik yang berbeda dari VCC. Biasa digunakan untuk aplikasi lain seperti konektivitas USB.

f. I/O (*Input/Output*)

Berfungsi untuk memonitor semua komunikasi yang keluar maupun yang masuk ke kartu.

g. C4 dan C8

Contact C4 dan C8 merupakan *contact* tambahan yang digunakan bila sewaktu-waktu diperlukan untuk operasi pada kartu atau penggunaan *interface* di masa yang akan datang (Rankl, et al.,2003).

2. *Contactless Smart Card*

Contactless smart card berkomunikasi dengan terminal melalui sinyal frekuensi radio. *Contactless smart card* tidak memiliki baterai sehingga kartu ini memiliki *inductor* yang dapat menangkap sinyal frekuensi radio sebagai daya elektronik bagi kartu. *Contactless card* memerlukan jarak tertentu untuk melakukan pertukaran data dengan *card reader* (Rankl, et al.,2003).

Standar komunikasi *contactless smart card* adalah ISO/IEC 14443. Berikut adalah tabel standarisasi ISO/IEC 14443.

Tabel 1 Standarisasi ISO/IEC *Contactless Smart Card*

Standar	Tipe <i>Contactless Smart Card</i>	Jarak Komunikasi
ISO/IEC 10536	<i>Close-coupling card</i>	± 1 cm
ISO/IEC 14443	<i>Proximity coupling card (PICC)</i>	± 10 cm
ISO/IEC 15693	<i>Vicinity coupling card (VICC)</i>	± 1 m

E. **Sistem Memori Smart Card**

Smart card terdiri dari 3 macam jenis memori, yaitu:

1. ROM (*Read-Only Memory*)

Merupakan tipe memori yang hanya dapat dibaca (*read*) dan tidak bisa ditulis (*write*), dan tidak membutuhkan tenaga listrik untuk menyimpan dan menjaga keutuhan data. ROM tidak dapat ditulis kembali setelah kartu dibuat, dan berisikan sebuah sistem operasi. ROM dari *smart card* berisikan data dan aplikasi pengguna yang bersifat permanen (Rankl, et al.,2003).

2. EEPROM (*Electrical Erasable Programmable Read Only Memory*)

EEPROM hampir serupa dengan ROM yang dapat menyediakan data ketika tenaga dari memori dimatikan. Perbedaan di antara keduanya adalah isi dari memori yang dapat dimodifikasi selama penggunaan kartu secara normal. EEPROM dapat menerima setidaknya 100.000 kali penulisan, dan dapat menyimpan data selama 10 tahun. Membaca dari EEPROM sama cepatnya dengan membaca dari RAM, namun menulis pada EEPROM lebih lama dibandingkan menulis dari RAM (Rankl, et al.,2003).

3. RAM (*Random Access Memory*)

RAM adalah sebuah memori yang digunakan untuk menampung data yang disimpan selama sebuah proses berlangsung. Jumlah akses pada suatu RAM tidak terbatas dan membutuhkan tenaga listrik untuk beroperasi. RAM merupakan memori yang bersifat *nonpersistent* yaitu informasi yang disimpan pada RAM akan hilang ketika tenaga listrik dari RAM hilang (Rankl, et al.,2003).

F. **Client dan Server Side Scripting**

PHP merupakan sebuah bahasa *scripting* yang bersifat *open source* dan dirancang untuk pengembangan situs *web*. Awalnya PHP dimulai dari peralatan personal *home page* yang dikembangkan oleh Rasmus Lerdorf. Nama lain dari PHP adalah *Hypertext Preprocessor* yang mempunyai kemampuan untuk memproses halaman *web* sebelum ditampilkan (Valade, 2004).

Kelebihan-kelebihan dari PHP antara lain:

1. Cepat

Pada sebuah situs *web*, karena menggunakan kode HTML, waktu proses untuk menampilkan sebuah halaman *web* menjadi lebih pendek.

2. Mudah digunakan

PHP memiliki sintaks sederhana dan mudah dimengerti dan digunakan bahkan bukan untuk seorang programmer. Untuk digunakan di dalam situs *web*, kode PHP dirancang agar dapat dimasukkan dengan mudah dalam kode HTML.

3. Berjalan pada banyak sistem operasi

PHP dapat berjalan di banyak sistem operasi seperti Windows, Linux, Mac OS dan jenis Unix lainnya.

4. Bisa di-*customise*

Sesuai aturan *open source* yaitu mengizinkan para programmer untuk mengubah perangkat lunak PHP, menambah atau memodifikasi fitur jika diperlukan untuk mencocokkan lingkungan yang ada.

5. Gratis

PHP bersifat gratis akan tetapi tetap dapat diandalkan.

G. Pengertian Socket

Socket adalah sebuah abstraksi di mana suatu aplikasi mengirim dan menerima data seperti dengan membuka *file* melalui aplikasi untuk membaca dan menulis data. Sebuah *socket* memperbolehkan suatu aplikasi untuk melakukan '*plug in*' kepada suatu jaringan dan melakukan komunikasi dengan aplikasi lainnya yang berada pada jaringan yang sama. Informasi yang ditulis ke sebuah *socket* oleh suatu aplikasi didalam suatu mesin dapat dibaca oleh aplikasi lainnya di dalam mesin yang berbeda dan seterusnya.

Socket datang dalam bentuk yang berbeda-beda tergantung oleh protokol yang digunakan dan beda *stack* pada sebuah protokol. Bentuk utama *socket* dalam TCP/IP adalah *stream sockets* dan *datagram sockets*. *Stream sockets* didalam TCP (di bawah IP) digunakan sebagai *end-to-end protocol* dan menyediakan kehandalan dalam pengiriman byte. *Datagram sockets* menggunakan UDP (di bawah IP) dan menyediakan kemampuan untuk mengirim data individu sampai 65.500 bytes. *Stream* dan *datagram sockets* juga didukung oleh protokol lainnya.

Sebuah *socket* yang menggunakan protokol TCP/IP dapat secara unik dikenali oleh alamat internet, *end-to-end protocol* (TCP atau UDP) dan nomor *port*. Ketika pertama kali *socket* dibuat, *socket* sudah mempunyai protokol yang terkait tetapi bukan alamat internet atau nomor *port*. Sebelum sebuah *socket* terikat oleh sebuah nomor *port*, *socket* tidak bisa menerima pesan dari aplikasi di luar jaringan (Donahoo, et al.,2001).

H. TCP Client

Tugas dari TCP *Client* adalah untuk memulai suatu komunikasi dengan sebuah *server* yang secara pasif menunggu untuk dihubungi.

Pada dasarnya TCP *Client* memiliki 4 tahap dasar:

1. Membuat TCP *socket* menggunakan `socket()`.
2. Membentuk suatu koneksi ke *server* menggunakan `connect()`.
3. Berkomunikasi menggunakan `send()` dan `recv()`.
4. Mengakhiri koneksi dengan `close()`.

Sebuah TCP *socket* harus terhubung dengan *socket* lainnya sebelum berbagai data dapat dikirim. *Client* memulai koneksi ketika sebuah *server* secara pasif menunggu untuk seorang klien terhubung (Donahoo, et al.,2001).

I. TCP Server

Tugas dari TCP *Server* adalah untuk mengatur komunikasi akhir dan secara pasif menunggu untuk sebuah koneksi dari klien.

Pada dasarnya TCP *Server* memiliki 4 tahap dasar:

1. Membuat TCP *Socket* menggunakan `socket()`.
2. Menetapkan nomor *port* ke *socket* dengan `bind()`.
3. Memberitahu sistem untuk memperbolehkan adanya koneksi dibuat pada *port* tersebut menggunakan `listen()`.
4. Secara berulang-ulang:
 - a. Memanggil `accept()` untuk mendapatkan *socket* baru untuk masing-masing koneksi klien.
 - b. Berkomunikasi dengan klien melalui *socket* baru tersebut menggunakan `send()` dan `recv()`.
 - c. Menutup koneksi klien menggunakan `close()`.

Sebuah server harus melakukan *binding* kepada sebuah alamat pada sebuah *socket* dan kemudian menggunakan *socket* tersebut sebagai sebuah *channel* untuk menerima *socket* lainnya yang terhubung ke klien (Donahoo, et al.,2001).

J. QT Framework

Qt *framework* pertama kali dipublikasikan pada bulan Mei 1995. Dikembangkan oleh Haavard Nord (Trolltech's CEO) dan Eirik Chambe-Eng (Trolltech's Chief Troll). Haavard dan Eirik bertemu di Norwegian Institute of Technology di Trondheim dan mereka berdua lulus dengan gelar master di bidang *computer science*. Haavard mempunyai ketertarikan dalam pengembangan C++ UI pada tahun 1988 ketika ia disewa oleh perusahaan Swedia untuk mengembangkan suatu *framework* untuk C++ GUI. Pada tahun 1990, Haavard dan Eirik bekerja bersama pada suatu aplikasi C++ *database* untuk ultrasound images. Sistem tersebut harus dapat berjalan dengan GUI pada Unix, Macintosh, dan Windows.

Pada tahun 1991, Haavard mulai menulis *class* yang akhirnya menjadi Qt. Pada tahun 1993, Haavard dan Eirik membangun grafik kernel pertama untuk Qt dan bisa diimplementasi dengan *widget* mereka. Huruf Q dipilih sebagai awalan *class* karena huruf itu terlihat bagus di Haavard's Emacs font. Huruf "t" ditambahkan dari kata *toolkit*. Pada tahun 2005, Qt versi 4 dirilis dengan 500 *class* dan lebih dari 9.000 fungsi. Qt 4 merupakan Qt edisi pertama yang tersedia sebagai *open source* di semua *platform* yang mendukungnya (Blanchette, et al.,2004).

III. TINJAUAN PUSTAKA

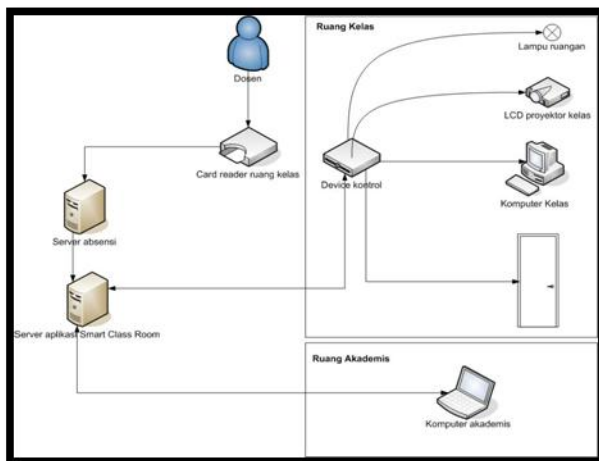
A. Model Konseptual

Aplikasi yang akan menangani inputan *smart card* dosen akan diletakan pada bagian *server*. Aplikasi ini akan dibangun dengan menggunakan Qt *toolkit* yang berbasiskan bahasa pemograman C++.

Aplikasi ini akan menangani 3 inputan data secara bersamaan, antara lain inputan dari kartu *BINUSIAN Card* dosen, inputan dari komputer akademis sebagai pengontrol, inputan dari perangkat kelas yang dibuat untuk mengirimkan *signal* menyalakan atau mematikan fasilitas-fasilitas kelas seperti lampu, komputer dan LCD monitor ataupun membuka dan mengunci pintu.

Aplikasi untuk mengontrol dan memonitor pemakaian ruang akan diletakkan di bagian akademis. Ini bertujuan agar akademis dapat secara langsung mengontrol dan memonitor pemakaian ruang secara langsung. Aplikasi ini dibuat dengan berbasis web dengan bahasa pemrograman *PHP (Hypertext Preprocessor)*.

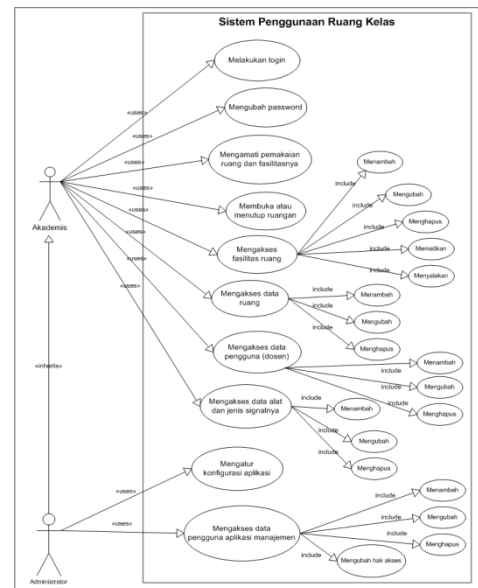
Gambaran mengenai proses sistem pemecahan masalah yang telah dijelaskan dapat dilihat pada Gambar 3



Gambar III Model Desain Dari Sistem Solusi Yang Akan Dibangun

B. Desain Model

Aplikasi yang dibangun digambarkan dengan menggunakan Use Case Diagram seperti pada Gambar 4.



Gambar 4 Use Case Proses Bisnis Penggunaan Ruang Kelas

C. Kebutuhan Perangkat Lunak

Untuk mengimplementasikan rancangan aplikasi dan basis data yang telah dibangun, diperlukan perangkat lunak dengan spesifikasi tertentu.

1. Komputer Server

Spesifikasi perangkat lunak komputer *server* yang diperlukan untuk implementasi rancangan aplikasi dan basis data pada aplikasi *server smart class room* yang berperan sebagai pemeriksa, pengirim dan pencatat status dan perintah yang terjadi ketika proses penggunaan ruang sedang berlangsung sebagai berikut:

- Sistem operasi Linux Ubuntu v.9.10 atau Microsoft Windows Server 2003
- MySQL 5.1.22
- Qt SDK 2009.03
- Apache v.2.2.14
- PHP v.5.2.12

2. Komputer Client

Spesifikasi perangkat lunak komputer *client* yang diperlukan untuk implementasi rancangan aplikasi dan basis data pada aplikasi bagian akademis yang berperan dalam pengaturan data dan informasi sebagai berikut:

- Sistem operasi Linux, Unix, Microsoft, Mac dan lainnya
- Browser* Internet Explorer, Mozilla Firefox, Opera, dan lainnya

D. Kebutuhan Perangkat Keras

Untuk mengimplementasikan rancangan aplikasi dan basis data yang telah dibangun, diperlukan perangkat keras dengan spesifikasi tertentu.

1. Komputer Server

Spesifikasi perangkat keras pada komputer *server* yang diperlukan untuk implementasi rancangan aplikasi dan basis data untuk aplikasi *server smart class room* yang berperan sebagai pemeriksa, pengirim dan pencatat status dan perintah yang terjadi ketika proses penggunaan ruang sedang berlangsung sebagai berikut:

- a. Processor Core™ 2 Duo CPU T6600 2,20 GHz
- b. Memory 1 GB
- c. Harddisk drive dengan kapasitas 80 GB atau lebih
- d. Keyboard dan mouse
- e. Monitor
- f. LAN Card

2. Komputer *Client*

Spesifikasi perangkat keras komputer *client* yang diperlukan untuk implementasi rancangan aplikasi dan basis data pada aplikasi bagian akademis yang berperan dalam pengaturan data dan informasi sebagai berikut:

- a. Processor 1 GHz
- b. Memory 512 MB
- c. Harddisk drive dengan kapasitas 40 GB
- d. Keyboard dan mouse
- e. Monitor
- f. LAN Card

3. Perangkat Keras Pengontrol

Perangkat keras pengontrol yang digunakan untuk menyalakan dan mematikan alat pada ruang kelas adalah ARM2386 *Controller*.

IV. EVALUASI

A. Evaluasi Waktu

Untuk melakukan evaluasi waktu pada sistem yang sedang berjalan, maka dilakukan observasi sistem penggunaan ruang kelas beserta fasilitas alatnya yang dilakukan dosen ketika proses belajar mengajar. Observasi ini dilakukan pada tanggal 12 Desember 2009, mulai dari pukul 7.20 sampai dengan pukul 13.00 (3 *shift*) di Kampus Anggrek Universitas Bina Nusantara. Tabel 2 menggambarkan hasil evaluasi waktu yang dilakukan.

Tabel 2 Evaluasi Waktu

No	Rata - rata waktu yang dibutuhkan	Sistem yang sedang berjalan	Sistem yang diusulkan
1	Melakukan <i>taping</i> kartu <i>smart card</i> , menyalakan fasilitas lampu, <i>LCD projector</i> , komputer.	± 4 menit	± 33 detik
2	Mematikan fasilitas lampu, <i>LCD projector</i> , komputer.	± 2 menit	± 10 detik
3	Waktu dosen menunggu kunci lemari komputer, dan <i>remote control</i> ke petugas akademis.	± 9 detik	0
4	Waktu dosen mengembalikan kunci lemari komputer, dan <i>remote control</i> .	± 5 detik	0
6	Selisih waktu yang dapat dihemat (sistem yang sedang berjalan – sistem yang diusulkan)		± 5 menit 31 detik

Hasil evaluasi waktu diambil dengan pengamatan lapangan terhadap ± 40 kelas dengan dosen yang bervariasi. Rata-rata dosen yang dihitung pada evaluasi ini adalah ± 40 orang. Jumlah waktu yang dapat dihemat dari penggunaan sistem yang diusulkan adalah ± 5 menit 31 detik.

B. Evaluasi Sumber Daya Manusia

Bagian ini menggambarkan evaluasi sumber daya manusia yang dibutuhkan baik pada sistem yang sedang berjalan maupun yang diusulkan. Evaluasi sumber daya manusia yang dibutuhkan pada tiap-tiap sistem dapat dilihat pada Tabel 3

Tabel 3 Evaluasi Sumber Daya Manusia

No	Sistem yang sedang berjalan	Sistem yang diusulkan
1	Petugas bagian ISS yang mengurus kunci ruang yang harus dibuka dan ditutup untuk perkuliahan.	-
2	Petugas yang membuka dan menutup pintu ruang	-

	pada saat perkuliahan dimulai.	
3	Petugas akademis yang mengontrol dan mengatur pemakaian kelas oleh dosen	Petugas akademis yang mengontrol dan mengatur pemakaian kelas oleh dosen

Berdasarkan hasil evaluasi diketahui bahwa sumber daya manusia yang dibutuhkan pada sistem yang diusulkan lebih sedikit daripada sistem yang sedang berjalan karena sistem yang diusulkan menghilangkan proses pembukaan ruang kelas secara manual oleh petugas pembuka ruang.

C. Evaluasi Kesalahan Manusia

Bagian ini menggambarkan evaluasi kesalahan manusia yang dapat terjadi pada sistem yang sedang berjalan dan sistem yang diusulkan dalam proses penggunaan ruang. Evaluasi kesalahan manusia yang dapat terjadi pada setiap sistem dapat dilihat pada Tabel 4

Tabel IV Evaluasi Kesalahan Manusia Yang Dapat Terjadi

No	Kesalahan Manusia yang dapat terjadi	Pada sistem yang sedang berjalan	Sistem yang diusulkan
1	Kesalahan akademis memberikan kunci dan <i>remote controller</i> kepada dosen	± 2 kali - ± 5 kali dalam satu hari	-
2	Kesalahan Dosen lupa mengembalikan kunci komputer dan <i>remote control LCD proyektor</i> .	± 2 kali dalam satu bulan	-
3	Kesalahan dalam membuka atau menutup ruang oleh petugas pada awal perkuliahan.	± 5 kali dalam satu hari	-

Berdasarkan evaluasi kesalahan yang disebabkan oleh manusia pada sistem yang sedang berjalan dan yang diusulkan didapatkan bahwa sistem yang baru dapat mengurangi kesalahan yang disebabkan oleh kelalaian manusia dalam proses penggunaan kelas. Ini dikarenakan sistem yang baru membuat proses penggunaan kelas dilakukan secara otomatis dalam mengaktifkan ataupun menonaktifkan ruang kelas dan fasilitas alat yang ada di dalamnya. Akibatnya baik dosen, petugas bagian akademis dan pekerja yang membuka ruang tidak perlu secara langsung menangani proses menyiapkan kunci dan *remote control*, membuka ruang, menutup ruang, mematikan dan menyalakan fasilitas alat sehingga

kesalahan yang disebabkan manusia dari tiap proses dapat dihilangkan.

PUSTAKA

- Blanchette, Jasmin and Summerfield, Mark. *C++ GUI Programming with Qt 4, Second Edition*. s.l.: Prentice Hall, 2004. 0-13-235416-0.
- Donahoo, Michael J and Calvert, Kenneth L. *TCP/IP Sockets in C : Practical Guide for Programmers*. s.l.: Morgan Kaufmann, 2001.
- Ortiz, C.E. *An introduction to java card technology*. 2003.
- Rankl, Wolfgang and Effing, Wolfgang. *Smart Card Handbook Third Edition*. s.l.: John Wiley & Sons, Ltd, 2003.
- Tanenbaum, Andrew S. *Computer Networks, Fourth Edition*. s.l.: Prentice Hall, 2003. 0-13-066102-3.
- Valade, Janet. *PHP 5 For Dummies*. s.l.: For Dummies, 2004. 978-0764541667.