

PERANCANGAN APLIKASI ABSENSI LABORATORIUM KOMPUTER DENGAN MENGGUNAKAN SISTEM VERIFIKASI PENGGUNA DI UNIVERSITAS KRISTEN MARANATHA

Radiant Victor Imbar¹⁾, Robby Kurniawan²⁾

Jurusan Sistem Informasi

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri No. 65 Bandung 40164

email: radiant.vi@eng.maranatha.edu¹⁾, sylvirus@yahoo.com²⁾

Abstract

Maranatha Christian University as one of the universities in Bandung has computer laboratory for practical activities. Although, Maranatha Christian University has owned an absence application which record user who accessing the computer in laboratory, but the system can't track whether the user sit on its place correctly or not. For that reason UKM need a system to control and minimize the risk of lost and damage keyboard, mouse and computer table because the data have been recorded in database.

The main purpose of this research is to make a laboratory absent tracking system software to record the use of computer laboratory facilities.

Keywords: *absence tracking system, absence application, database*

I. Pendahuluan

Pada saat ini teknologi berkembang dengan pesat. Perkembangan teknologi ini menyebabkan meningkatnya permintaan sumber daya manusia (SDM) yang memiliki kompetensi di bidangnya dan menguasai teknologi untuk menyelesaikan masalah yang dihadapinya.

Universitas Kristen Maranatha (UKM) sebagai salah satu perguruan tinggi di Indonesia telah menyadari hal tersebut sehingga menyediakan fasilitas – fasilitas yang membantu mahasiswa dan mahasiswi dalam belajar. Salah satu fasilitas yang disediakan oleh UKM adalah laboratorium komputer. Penyediaan fasilitas ini memerlukan suatu pengawasan. Pengawasan ini bertujuan agar fasilitas tersebut dapat terpelihara dengan baik.

Kenyataannya banyak komputer serta fasilitas di dalam laboratorium komputer sering mengalami kerusakan. Kerusakan ini merupakan bukti masih lemahnya sistem pengawasan yang dilakukan. Kerusakan ini umumnya disebabkan oleh mahasiswa atau mahasiswi yang tidak bertanggungjawab. Identifikasi terhadap mahasiswa atau mahasiswi ini sangat sulit dilakukan karena belum adanya suatu pencatatan pemakaian komputer di laboratorium komputer yang dapat mencatat serta menolak akses masuk komputer apabila mahasiswa atau mahasiswi tidak berhak menggunakan komputer tersebut.

Oleh sebab itu penulis mencoba untuk membangun suatu aplikasi pengontrolan pemakaian komputer di laboratorium komputer yang dapat mencatat serta menolak akses masuk komputer apabila pengguna tersebut tidak berhak menggunakan komputer tersebut. Dengan adanya aplikasi ini pemakaian laboratorium itu dapat dikontrol

dengan baik, sehingga kerusakan fasilitas laboratorium komputer yang disebabkan oleh mahasiswa atau mahasiswi yang tidak bertanggungjawab dapat dicegah serta pengaksesan komputer oleh mahasiswa atau mahasiswi yang tidak berhak dapat diatasi.

Aplikasi absensi ini akan menangani:

1. Pencatatan pemakaian komputer yang digunakan mahasiswa atau mahasiswi.
2. Laporan pemakaian komputer yang dapat digunakan sebagai bukti penggunaan komputer oleh mahasiswa atau mahasiswi.
3. Proteksi terhadap komputer laboratorium oleh mahasiswa atau mahasiswi yang tidak berhak menggunakan komputer tersebut.

II. Landasan Teori

II.1 UML (*Unified Modeling Language*)

UML adalah suatu permodelan yang digunakan pada saat ini untuk merepresentasikan suatu kesatuan konsep – konsep dan notasi yang digunakan untuk memodelkan suatu permasalahan. Tujuan dari UML ini adalah untuk menjadi suatu bahasa umum yang digunakan untuk membuat model yang berorientasi objek pada pengembangan rekayasa komputer.

UML memodelkan suatu program dalam bentuk kelas dan objek.

Kelas adalah “pola” dari sebuah objek. Dari sebuah kelas dapat dibuat banyak objek yang sejenis.

Contoh:

Dari sebuah pola baju merek xxx, dapat dibuat banyak(objek) baju bermerek xxx, yang mempunyai warna merah ,biru, hijau, dan lain sebagainya.

Objek adalah sesuatu yang mempunyai sifat.

Contoh:

Objek :Sepeda

Sifat :Warna

Aksi :Rem

Objek :Batu

Sifat :Warna

Aksi :-

Dalam program objek adalah gabungan variable dan fungsi yang saling terkait. Sifat objek sering disebut sebagai property dan aksi sering disebut method. Perbedaan yang paling mencolok antara kelas dengan objek adalah sifat kelas belum terdefinisi sedangkan objek sudah terdefinisi.

Tujuan dari diagram kelas ini adalah untuk memodelkan kelas. Dalam aplikasi berorientasi objek, kelas memiliki *atribut*, operasi dan hubungan dengan kelas lainnya. Diagram kelas dalam UML dapat memodelkan hal ini dengan sangat mudah.

Sebuah kelas dimodelkan dengan sebuah persegi panjang yang memiliki tiga bagian. Bagian paling atas adalah nama kelas itu kemudian bagian yang ditengah adalah *atribut – atribut* dari kelas itu, dan yang paling bawah adalah operasi yang dilakukan oleh kelas itu.

II.2 Basis Data (*DataBase*)

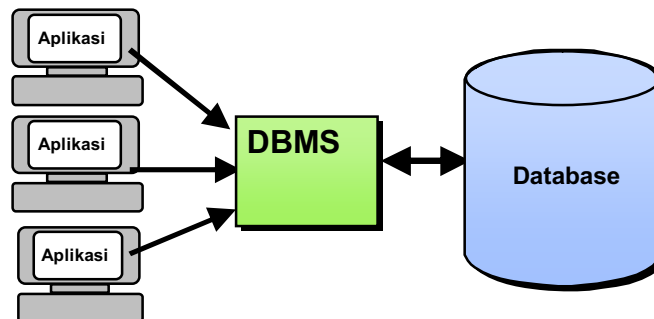
Basis data terdiri dari dua kata, yaitu basis dan data. Basis dapat diartikan sebagai tempat penyimpanan sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu obyek yang direkam dalam bentuk angka, huruf, simbol, teks. Prinsip utama basis data adalah untuk pengaturan data dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data.

Basis data digunakan untuk menyimpan, memanipulasi dan mengambil data hampir semua tipe perusahaan termasuk bisnis, pendidikan, rumah sakit, pemerintahan dan perpustakaan.

Basis data dapat menghindari adanya *data redundancy*. *Data redundancy* adalah duplikasi data dimana beberapa file memiliki data yang sama sehingga menyebabkan pemborosan kapasitas media penyimpanan. Dengan menggunakan basis data, perubahan, penambahan dan penghapusan suatu data lebih mudah dilakukan. Basis data juga mendukung pengaksesan suatu *resource* secara bersama – sama oleh lebih dari sebuah aplikasi.

Sistem basis data ini lebih dikenal dengan istilah *DataBase Management System* (DBMS).

DBMS adalah sistem penyimpanan dan pengambilan data yang mengizinkan beberapa aplikasi untuk mengakses data tersebut dalam sebuah sistem terpadu sehingga tidak terjadi *data redundancy* dan integritas data dapat dipertahankan. Berikut ini adalah gambar dari sebuah DBMS yang mengatur penyimpanan data di dalam database serta bertugas sebagai media penghubung antara aplikasi dengan *database*.



Gambar 1. DBMS

Contoh-contoh dari DBMS yaitu:

- Oracle
- SQL Server
- Informix
- DB2

II.3 *Client Server Programming*

Client-Server Programming adalah suatu jenis teknologi yang menghubungkan *Client* dan *server* dalam suatu proses pada aplikasi.

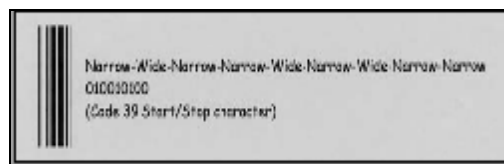
Langkah – langkah dalam *Client Server Programming* untuk *server* adalah sebagai berikut:

1. *Socket*
Socket digunakan sebagai alamat dari server. *Socket* ini memiliki atribut alamat *IP (Internet Protocol)* dan *port* yang dipakai oleh *server*.
2. *Bind*
Setelah *socket* dibangun, alamat *IP* dan *port* akan disatukan menjadi satu bagian dan di *bind* ke sistem.
3. *Listen*
Setelah di *bind server* akan memasuki fase *listen* dimana pada fase ini *server* akan mendengarkan apakah ada *client* terhubung dengannya.
4. *Accept*
Apabila ada permintaan dari *client* maka *server* akan menanggapi dan membangun hubungan dengan *client* tersebut.
5. *Read*
Server kemudian akan melakukan proses pembacaan paket data yang dikirim oleh *client*
6. *Write*
Server juga dapat melakukan pengiriman paket data melalui koneksi tersebut.
7. *Close*
Apabila *server* menerima pesan bahwa *client* terputus atau memutuskan hubungan maka *server* akan memutuskan koneksi dan menutup *port* yang digunakan untuk koneksi *client* tersebut.

Langkah – langkah dalam *Client Server Programming* untuk *client* hampir sama dengan *server* perbedaannya adalah pada *client* tidak terdapat proses *listen*.

II.4 *Barcode Scanner*

Pembacaan barcode dilakukan dengan menggunakan *photosensor* untuk mengkonversikan *barcode* yang dibaca ke sinyal elektrik. Pembacaan dilakukan dari awal kemudian sampai ke akhir dimulai dari kiri ke kanan atau sebaliknya. Cara pembacaan adalah dengan mengukur lebar dari pola berwarna hitam dan pola yang berwarna putih, dan menerjemahkan perbedaan pola tersebut ke dalam bentuk karakter dan mengirimkannya ke komputer.



Gambar 2. Gambar Contoh *Barcode*

Setiap *barcode* dimulai dengan *special start character* dan *special stop character* yang membantu *scanner* mendeteksi *barcode* dan mengetahui pembacaan *barcode* dimulai dari kiri (awal) atau dari kanan (akhir).

III. Analisa dan Desain Aplikasi

Analisa dan desain aplikasi ini dibagi atas 2 bagian yaitu analisa tujuan dan analisa proses aplikasi.

III.1 Analisa Tujuan

Tujuan dari aplikasi ini adalah :

1. Mencatat setiap transaksi pemakaian komputer di laboratorium IT Universitas Kristen Maranatha.
2. Mengetahui apakah suatu user itu menempati tempatnya dengan benar.
3. Melakukan tindakan pencegahan terhadap penggunaan komputer oleh user yang tidak menempati tempatnya dengan benar.

NO	Kejadian (<i>Event</i>)	NO	Respon aplikasi
1	Staf memasukan nomor mahasiswa dan nomor komputer mahasiswa		
		2	Memeriksa nomor mahasiswa dan komputer dengan syarat : Apabila nrp atau nomor komputer tidak terdaftar munculkan pesan kesalahan
3	Pesan "nomor mahasiswa tidak terdaftar" atau "Nomor komputer tidak ada"		
		4	Apabila nomor mahasiswa sudah masuk atau komputer sedang dipakai munculkan pesan kesalahan
5	Pesan "nomor mahasiswa sudah masuk" atau "Komputer sedang dipakai"		
		6	Apabila nomor mahasiswa sudah masuk dan komputer sedang dipakai, ubah status user menjadi 'K'. restart komputer client.
		7	Apabila nomor mahasiswa belum masuk dan komputer belum dipakai tambah data pemakaian dengan status 'I'

III.2 Analisa Proses Aplikasi

Sebelum menggunakan aplikasi ini, staf laboratorium harus memasukan data komputer kedalam *database*. Data ini berupa nomor komputer dan *IP Address* dari komputer.

Untuk memasukan data komputer tersebut, staf dapat menggunakan *form* data komputer.



	IPAddress	Komputer
*		

Reload Update Cancel

Gambar 3. *Form Data Komputer*

Sebelumnya menggunakan laboratorium komputer, pengguna harus mendaftarkan diri kepada staf laboratorium sehingga informasi pengguna ada didalam *database*.

Untuk memasukan informasi pengguna ini staf dapat menggunakan *form* data mahasiswa.

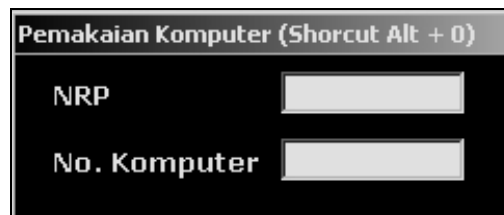


	NRP	Nama
*		

Load Update Cancel All

Gambar 4. *Form Data Mahasiswa*

Ketika pengguna ingin menggunakan komputer didalam laboratorium, Staf akan memasukan data pengguna kedalam *form* pemakaian. Data pengguna ini terdiri atas nomor mahasiswa dan nomor komputer yang akan dipakai.



Gambar 5. Form Pemakaian Komputer

Data yang dimasukan tersebut akan diperiksa oleh aplikasi. Apabila pengguna atau nomor komputer tidak terdaftar dalam *database*, aplikasi akan memunculkan pesan pengguna atau nomor komputer tidak terdaftar. Apabila pengguna atau nama komputer sudah masuk maka aplikasi akan mengeluarkan pesan bahwa pengguna atau nomor komputer tersebut sudah masuk. Apabila nomor mahasiswa dan nomor komputer belum masuk maka data mahasiswa, data komputer, waktu masuk, tanggal masuk dan status yang bernilai 'I' akan ditambahkan ke dalam *database*.

Ketika pengguna selesai menggunakan komputer, staf akan memasukan kembali data mahasiswa ke dalam *form* pemakaian. Aplikasi akan secara otomatis mengupdate *database* pengguna tersebut dan menambahkan waktu keluar sekaligus merubah status menjadi 'K'.

Ketika pengguna melakukan akses masuk, informasi pengguna akan di ambil dan dikirim ke *server* oleh aplikasi yang ada di komputer *client*.

Staf dapat melihat status pengguna yang melakukan tindakan akses masuk komputer pada *form server*.



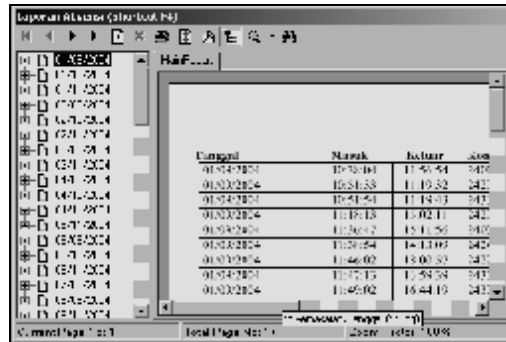
Gambar 6. Form Status Server

Form server ini juga secara otomatis memeriksa apakah pengguna tersebut memiliki hak untuk memakai komputer tersebut dengan cara membandingkan data yang ada dalam *database* dengan informasi pengguna yang dikirimkan *client*.

Jika pengguna menempati komputer yang sesuai dengan yang tercatat dalam *database* maka *server* akan menampilkan pesan bahwa pengguna berhak melakukan akses masuk. Jika tidak maka *server* akan mengirim sinyal *restart* ke komputer *client* dan menampilkan pesan bahwa terjadi akses masuk oleh pengguna yang tidak berhak.

Aplikasi ini juga dapat menampilkan laporan yang berisi informasi pemakaian komputer dalam bentuk *hardcopy* maupun *softcopy*.

Fasilitas untuk melihat laporan pemakaian komputer ini tersedia pada *form* laporan



Tanggal	Masuk	Keluar	Jam
01/04/2004	10:30:00	11:50:54	84%
01/04/2004	10:31:33	11:19:32	24%
01/04/2004	10:51:51	11:18:31	34%
01/04/2004	11:18:13	11:02:11	24%
01/04/2004	11:30:17	11:11:56	34%
01/04/2004	11:31:54	11:13:03	24%
01/04/2004	11:46:07	11:00:37	24%
01/04/2004	11:57:15	11:58:38	34%
01/04/2004	11:49:02	11:44:19	24%

Gambar 7. Form Laporan Pemakaian

IV. Kesimpulan dan Saran

Aplikasi “Absensi Lab 2004” ini memiliki kemampuan sebagai berikut:

1. Menyimpan informasi mahasiswa yang terdiri dari nomor mahasiswa dan nama mahasiswa.
2. Menyimpan informasi komputer yang terdiri dari *IP* komputer dan nama komputer.
3. Menyimpan informasi pemakaian komputer .
4. Menghalangi akses masuk oleh pengguna yang tidak berhak atau yang tidak terdaftar.

Selain memiliki kemampuan tersebut aplikasi ini juga memiliki beberapa kekurangan yang harus diperhatikan yaitu:

1. Proses instalasi harus memakai *default path*. Hal ini disebabkan aplikasi ini belum memiliki kemampuan untuk menentukan *path directory*.
2. Akses masuk sistem operasi *windows* dengan menggunakan ‘*safe mode*’ masih diijinkan sehingga pengguna yang mengetahui *password administrator* lokal dapat mengakses komputer.oleh karena itu disarankan *administrator* dapat menyimpan *password*nya dengan baik dan melakukan perubahan *password* secara berkala.
3. Konfigurasi aplikasi masih dalam bentuk teks yang tidak tersandikan (tidak ada *encryption*).

V. Daftar Pustaka

- [1] Bunyamin, H.(2003), '*Aplikasi POS (Point of Sales)*'. Universitas Kristen Maranatha, Bandung.
- [2] Deitel, H. M., P. J. Deitel, J.Listfield, T. R. Nieto, C.Yaeger, M. Zlatkina. *C# How To Program* .Upper Saddle River,NJ:Prentice Hall, 2002.
- [3] Imbar, R.V.(2004), '*Pengembangan Perangkat Lunak Sistem Absensi Laboratorium Dengan Menggunakan Barcode Scanner di Universitas Kristen Maranatha*', Universitas Kristen Maranatha.
- [4] Manfre, M. *Creating A Windows Service With C#* .
<http://www.devhood.com>, Tanggal Akses: 25 Oktober 2004.
- [5] Mayo, J. *Beginners's C# Tutorial*. CSharp Station.
<http://www.csharp-station.com>, Tanggal Akses: 22 September 2003.
- [6] Mössenböck, H. *Advanced C#*. University of Linz, Austria 2004.
- [7] *Object-Oriented Frameworks for Network Programming*.
Tanggal Akses: 27 Oktober 2002.
- [8] Schult, W. and A. Polze. *Aspect-Oriented Programming with C# and .NET*. Hasso-Plattner-Institute. University Potsdam, Germany 2004.
- [9] Setiawan, E.(2003) '*Modul Praktikum Basis Data*'. Universitas Kristen Maranatha, Bandung.
- [10] SoftTech Developer.*Get Current Logged On User*.
<http://www.planetsourcecode.com>, Tanggal Akses: 25 Oktober 2004.
- [11] Subrata, L. R.(2003) '*Object Oriented and Design*'. Universitas Kristen Maranatha, Bandung.