

Aplikasi Presensi via PDA dengan Konektivitas Melalui Jaringan WiFi

Teddy Marcus Zakaria, Fandi

Jurusan Teknik Informatika
Fakultas Teknologi Informasi Universitas Kristen Maranatha
Jl. Prof. drg. Suria Sumantri No. 65, Bandung 40164
Email : teddy.mz@maranatha.edu , fandi_tan@yahoo.com

Abstract

This report explain making of application of presensi student use PDA by using network of wifi. Application of desktop (Presensi Manager) requiring Log In to limit level access user. User acting as operator can addition, distorting, and print officer data, student, subject, majors, semester, class, and also student class. While user acting as lecturer only can to see data exist in in database, without can to alter it.

Application of PDA (PresSender) also require Log In to be able to be used. User able to access application of PDA this only user acting as lecturer. Lecturer can to conduct presensi as according to class and subject taught by pertinent lecturer. Lecturer also can to conduct data synchronization. Operator only can to conduct data synchronization without can do presensi.

Application implementation by using Visual Studio 2005 for the desain of interface. Data storage media is Microsoft SQL server 2005 and SQL CE. Relation between application of PC and application of PDA pass wifi, and used to pass web service.

Keyword : Application of Presensi, wifi, Presensi Manager, PresSender, web service, data synchronization.

1. Pendahuluan

1.1. Latar Belakang

Penggunaan PDA sudah makin sering dijumpai di lapisan masyarakat, khususnya bagi mereka yang banyak berhubungan dengan dunia bisnis dan teknologi. PDA dapat membantu penggunaanya dalam melakukan aktivitas yang sebelumnya sulit untuk dilakukan. Sebagai contoh, pada saat melakukan presensi perkuliahan, mayoritas masih menggunakan cara manual, yaitu dengan mengedarkan kertas presensi yang ditandatangani oleh mahasiswa. Beranjak dari hal itu, akan dibuat suatu aplikasi di dalam suatu PDA yang mampu melakukan presensi mahasiswa oleh dosen yang bersangkutan, dan langsung dikirim ke aplikasi *desktop* di suatu *Personal Computer* (PC) untuk kemudian disimpan ke dalam *database*.

1.2. Tujuan

Aplikasi ini ditujukan untuk menghemat tenaga dan waktu. Aplikasi ini akan menyimpan hasil presensi mahasiswa langsung ke dalam *database*. Dibandingkan dengan sistem presensi manual, yang harus memasukkan data lagi oleh staff Tata Usaha (TU), hal ini tentu akan lebih efektif.

1.3. Rumusan Masalah

Beberapa permasalahan dalam penelitian ini adalah sebagai berikut :

- Bagaimana melakukan presensi mahasiswa di PDA.
- Bagaimana melakukan penyimpanan data presensi di PDA.
- Bagaimana melakukan sinkronisasi data antara PC dan PDA.
- Bagaimana melakukan manajemen data yang berhubungan dengan presensi.

2. Kajian Pustaka

2.1. Sinkronisasi

Sinkronisasi diperlukan untuk menghindari terjadinya ketidak konsisten data akibat adanya akses data secara konkuren. Proses-proses disebut konkuren jika proses-proses itu ada dan berjalan pada waktu yang sama, proses-proses konkuren ini bisa bersifat independen atau bisa juga saling berinteraksi. Proses-proses konkuren yang saling berinteraksi memerlukan sinkronisasi agar terkendali dan juga menghasilkan *output* yang benar. Disini sinkronisasi diperlukan agar data tersebut tetap konsisten.

2.2. Web Service

Web Service merupakan fenomena yang sangat panas saat ini karena, banyak kelebihan yang ditawarkan oleh *Web Service* terutama interoperabilitas tinggi dan penggunaannya yang dapat diakses kapanpun dan dimanapun selama mesin kita terhubung oleh jaringan Internet.

Keuntungan penggunaan *Web Service* :

4. Format penggunaan terbuka untuk semua platform.
5. Mudah dimengerti dan mudah men-debug.
6. Mudah untuk menengahi pesan - pesan proses dan menambahkan nilai.
7. *Routing* and pengiriman.
8. *Security*.
9. *Schema and service design*.
10. Terbuka, standard - standard berbasis teks.
11. Pencapaian modular.
12. Tidak mahal untuk diimplementasikan (relatif).
13. Mengurangi biaya integrasi aplikasi enterprise.

Web Service sepenuhnya berdasarkan pada standard XML dan Web. *Web Service* dapat membantu :

1. Perantara pada integrasi platform sepanjang eksekusi mesin virtual.
2. Integrasi antara Web dan OO *middleware*.
3. Integrasi dari aliran kerja terisolasi dan *service - service* (*Web Services Flow Language – WSFL*).
4. Pertukaran data pada aplikasi yang berbeda-beda (X - *Schema*, XSLT ++)

2.3. WiFi

WiFi merupakan kependekan dari *Wireless Fidelity*, memiliki pengertian yaitu sekumpulan standar yang digunakan untuk [Jaringan Lokal Nirkabel](#) (*Wireless Local Area Networks* – WLAN) yang didasari pada spesifikasi IEEE 802.11. Standar terbaru dari spesifikasi 802.11a atau b, seperti 802.11g, saat ini sedang dalam penyusunan, spesifikasi terbaru tersebut menawarkan banyak peningkatan mulai dari luas cakupan yang lebih jauh hingga kecepatan transfernya.

Awalnya WiFi ditujukan untuk penggunaan perangkat nirkabel dan Jaringan Area Lokal (LAN), namun saat ini lebih banyak digunakan untuk mengakses [internet](#). Hal ini memungkinkan seseorang dengan [komputer](#) dengan kartu nirkabel (*wireless card*) atau [Personal Digital Assistant](#) (PDA) untuk terhubung dengan internet dengan menggunakan titik akses (atau dikenal dengan *hotspot*) terdekat

WiFi dirancang berdasarkan spesifikasi [IEEE 802.11](#). Sekarang ini ada empat variasi dari 802.11, yaitu: 802.11a, 802.11b, 802.11g, and 802.11n. Spesifikasi *b* merupakan produk pertama WiFi.

Spesifikasi	Kecepatan	Frekuensi Band	Cocok dengan
802.11b	11 Mb/s	2.4 GHz	b
802.11a	54 Mb/s	5 GHz	a
802.11g	54 Mb/s	2.4 GHz	b, g
802.11n	100 Mb/s	2.4 GHz	b, g, n

Gambar 1 : Spesifikasi WiFi

Cara Kerja WiFi :

6. Mode Managed

Menggunakan [Access Point](#) yang berfungsi mengatur lalu lintas data. *Access Point* ini berfungsi seperti pusat transfer data. Kelebihan mode ini adalah transfer data lebih cepat, area *coverage* yang lebih luas, pengaturan dan keamanan data lebih terjamin. Mode ini biasa digunakan untuk *hotspot* WiFi dan perkantoran yang memerlukan stabilitas dan keamanan data.

7. Ad – Hoc

Koneksi antar *device* WiFi *peer too peer*. Kelemahan mode ini adalah bila *device* WiFi yang terhubung terlalu banyak, transfer data jadi lambat. Keuntungannya, lebih murah dan praktis bila yang terkoneksi cuma 2 atau 3, tanpa harus membeli *Access Point*.

Hardware WiFi yang ada di pasaran saat ini ada berupa :

6. PCI



Gambar 2 : WiFi PCI

7. USB



Gambar 3 : WiFi USB

8. PCMCIA



Gambar 4 : WiFi PCMCIA

9. Compact Flash



Gambar 5 : WiFi Compact Flash

3. Metode Penelitian

3.1. Analisis Kebutuhan

Bahasa pemrograman yang digunakan dalam implementasi *software* adalah C#. Dalam perancangan aplikasi ini, terdapat beberapa kebutuhan yang harus dipenuhi :

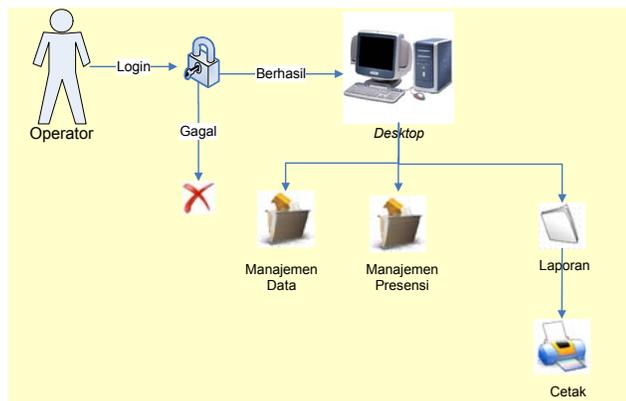
6. Bagi dosen

1. Dosen dapat melakukan presensi mahasiswa di PDA.
2. Dosen dapat memberikan tanda tangan sebagai tanda bahwa presensi sudah dilakukan.
3. Data presensi yang sudah dilakukan akan disimpan pada *database* PDA jika tidak terhubung dengan aplikasi PC (*offline*).
4. Data presensi pada *database* PDA dapat dikirim ke *database* PC jika sudah terhubung (*online*).
5. Dosen dapat melihat data presensi yang sudah pernah dilakukan.
6. Dosen dapat mengubah *password* nya masing – masing.
7. Dosen dapat melakukan sinkronisasi data jika terdapat penambahan, perubahan, atau pengurangan data pada aplikasi PC.

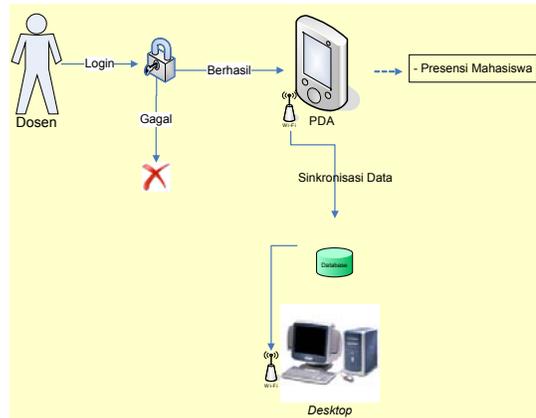
7. Bagi operator

1. Operator dapat melakukan penambahan dan perubahan data mahasiswa, pegawai, mata kuliah, jurusan, semester, kelas, kelas mahasiswa.
2. Operator dapat mengubah data presensi yang sudah dilakukan.
3. Operator dapat mencetak data presensi yang sudah pernah dilakukan, data mahasiswa, pegawai, mata kuliah, dan jurusan.
4. Operator dapat melakukan sinkronisasi data jika terdapat penambahan, perubahan, atau pengurangan data pada aplikasi PC.

Arsitektur dari sistem yang dibuat dapat dilihat pada gambar berikut.



Gambar 6 : Operator pada aplikasi PC

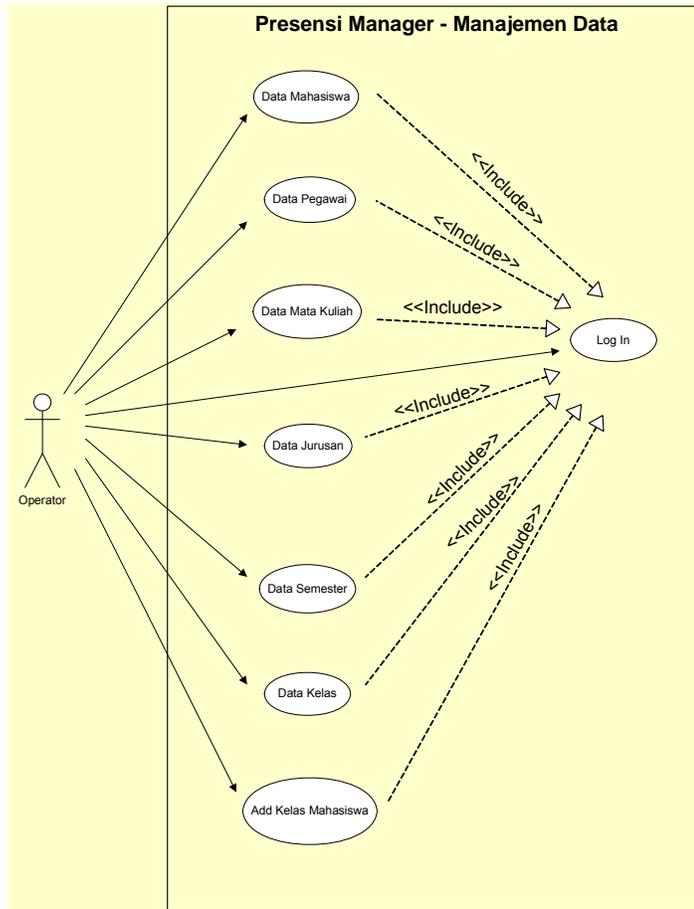


Gambar 7 : Dosen pada Aplikasi PDA

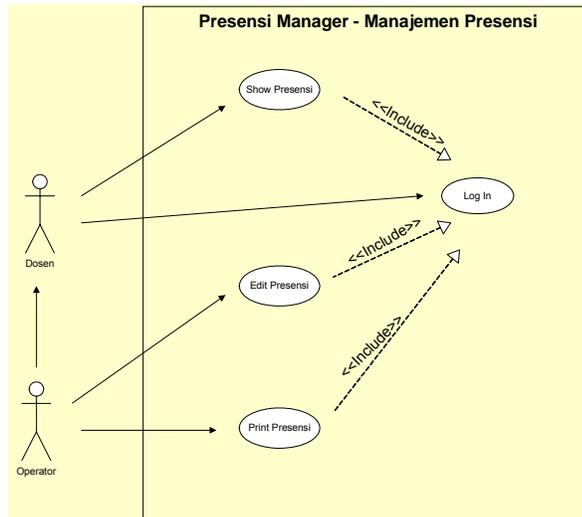
3.2. Desain Sistem

Pembuatan aplikasi presensi via PDA ini memerlukan adanya perancangan proses. Perancangan proses ini menggunakan UML (*Unified Modeling Language*).

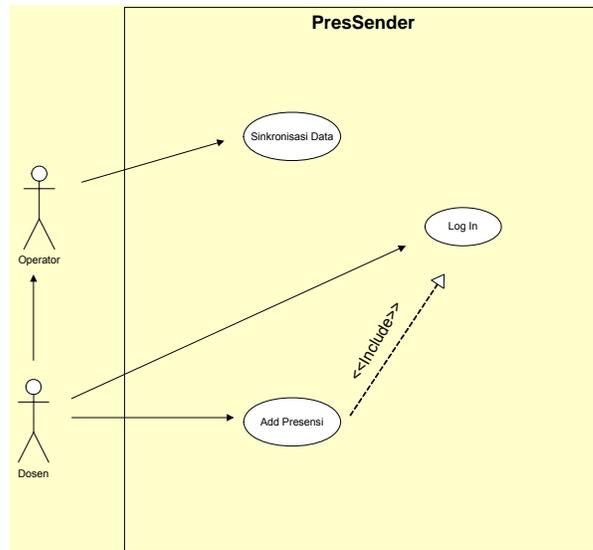
Pengguna aplikasi ini dibagi menjadi 2 tipe, yaitu dosen dan operator. Pada aplikasi PC, operator dapat melakukan penambahan data, perubahan data, dan mencetak data yang ada. Dosen hanya dapat melihat data yang ada tanpa memiliki kemampuan untuk menambah atau mengubahnya. Pada aplikasi PDA, operator hanya mampu untuk melakukan sinkronisasi data, sedangkan dosen juga memiliki kemampuan untuk melakukan presensi mahasiswa.



Gambar 8 : Use Case Diagram manajemen data



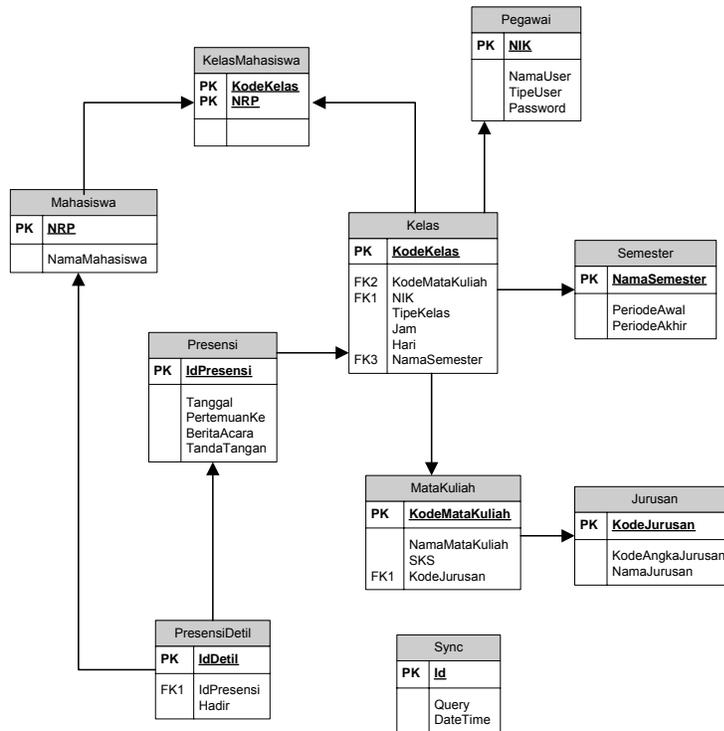
Gambar 9 : Use Case Diagram manajemen presensi



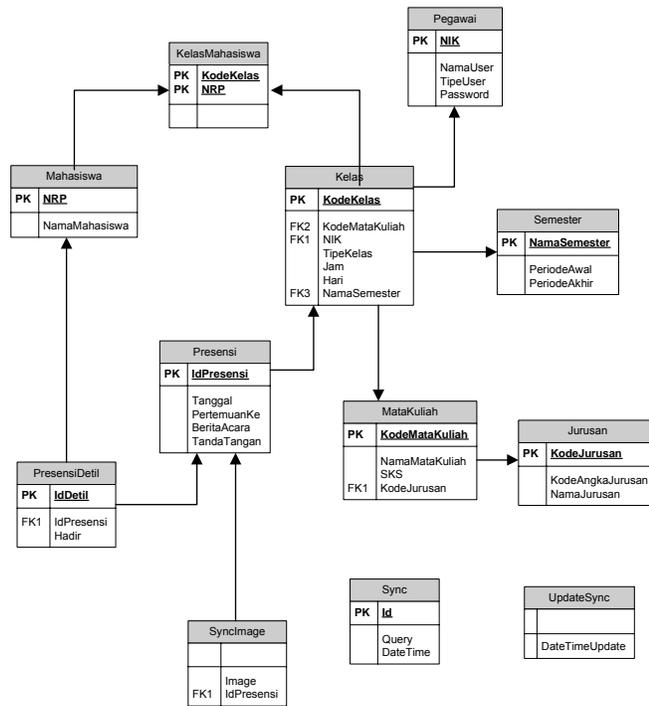
Gambar 10 : Use Case Diagram aplikasi PDA

3.3. Desain Basis Data

Desain basis data yang digunakan untuk menyimpan data tampak seperti pada relasi antar table berikut.



Gambar 11 : Relasi antar tabel aplikasi PC



Gambar 12 : Relasi antar tabel aplikasi PDA

Tabel – tabel yang digunakan adalah :

1. Pegawai : Menyimpan data pegawai.
2. Mahasiswa : Menyimpan data mahasiswa.
3. MataKuliah : Menyimpan data mata kuliah.
4. Jurusan : Menyimpan data jurusan.
5. Semester : Menyimpan data semester.
6. Kelas : Menyimpan data kelas.
7. KelasMahasiswa : Menyimpan data NRP mahasiswa dari suatu kelas.
8. Presensi : Menyimpan data presensi mahasiswa, yaitu PertemuanKe, BeritaAcara, dan TandaTangan.
9. PresensiDetil : Menyimpan data presensi detil, yaitu NRP mahasiswa dari suatu kelas dan kehadiran masing – masing mahasiswa.
10. Sync : Menyimpan query yang dilakukan, yang nantinya akan digunakan untuk melakukan sinkronisasi data.
11. SyncImage : Menyimpan data tanda tangan yang dilakukan oleh dosen.
12. UpdateSync : Menyimpan data tanggal dan waktu terakhir kali melakukan sinkronisasi data pada masing – masing PDA.

3.4. Implementasi

Aplikasi Presensi via PDA diimplementasikan dengan menggunakan C#.Net 2005. Program yang dibuat tampak seperti pada gambar berikut.



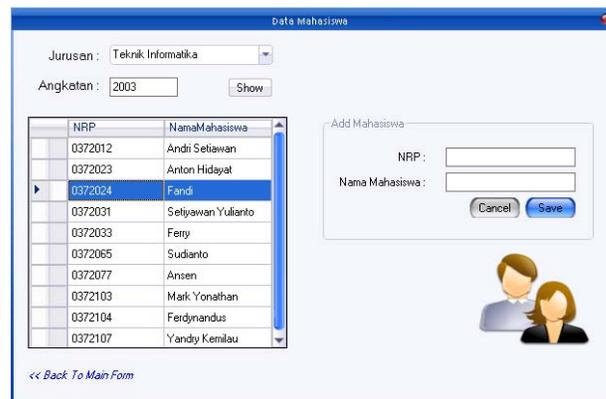
Gambar 13 : *LogInForm*

User yang ingin mengakses aplikasi PC harus melakukan *log in* terlebih dahulu dengan memasukkan NIK dan *password*. Jika data yang dimasukkan benar, maka akan ditampilkan *MainForm*.



Gambar 14 : *MainForm*

Pada *MainForm*, akan ditampilkan menu yang dibagi berdasarkan kategorinya masing – masing. Di bagian tengah, ditampilkan nama *user* yang *log in* terhadap aplikasi PC. Di bagian bawah terdapat informasi tanggal dan jam yang sesuai dengan tanggal dan jam pada komputer.



NRP	Nama Mahasiswa
0372012	Andri Setiawan
0372023	Anton Hidayat
0372024	Fandi
0372031	Setjawan Yulianto
0372033	Ferry
0372065	Sudianto
0372077	Ansen
0372103	Mark Yonathan
0372104	Ferdinandus
0372107	Yandy Kemilau

Gambar 15 : *MahasiswaForm*

Operator dapat melakukan penambahan data. Salah satunya adalah penambahan data mahasiswa. Pada saat akan melakukan penambahan data mahasiswa, dosen diminta untuk memasukkan NRP dan Nama Mahasiswa. Setelah menekan tombol *Save*, maka data mahasiswa tersebut akan tersimpan di dalam database.

The screenshot shows a window titled "Show Presensi". It has two radio buttons: "Semua" (selected) and "Berdasarkan Pertemuan". To the right, there are dropdown menus for "Semester" (Semester Ganjil 2007), "Jurusan" (Teknik Informatika), "Mata Kuliah" (Pemrograman Dasar C), and "Kelas" (A), along with a "Show" button. Below this is a table with columns: Nrp, Nama, and three columns numbered 1, 2, and 3. The table contains two rows of data.

Nrp	Nama	1	2	3
0372024	Fandi	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
0372065	Sudanto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

At the bottom left is a button "<< Back To Main Form" and at the bottom right is a "Print" button.

Gambar 16 : *ShowPresensiForm*

Operator dapat mencetak data presensi yang sudah dilakukan. Sebelum dicetak, data presensi harus ditampilkan terlebih dahulu. Data yang ditampilkan adalah Nama Semester, Jurusan, Mata Kuliah, Kelas, NRP mahasiswa, Nama Mahasiswa, dan kehadiran mahasiswa sesuai dengan jumlah pertemuan yang sudah pernah dilakukan oleh mata kuliah dan kelas tersebut.

The screenshot shows a window titled "Edit Presensi". It has dropdown menus for "Semester" (Semester Ganjil 2007), "Jurusan" (Teknik Informatika), "Mata Kuliah" (Pemrograman Dasar C), "Kelas" (A), and "Pertemuan Ke" (1), along with a "Show" button. Below this is a table with columns: NRP, NamaMahasiswa, and Hadir. The table contains two rows of data.

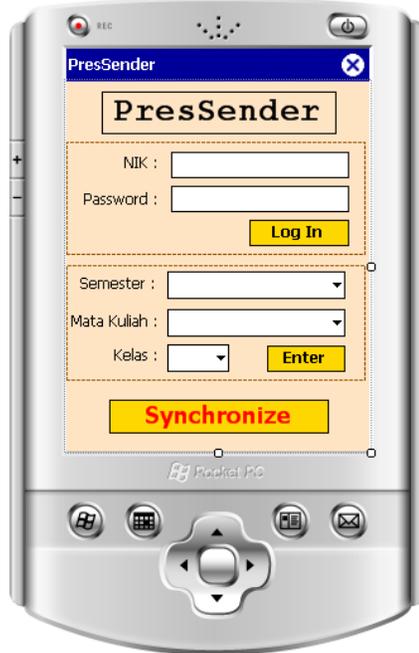
NRP	NamaMahasiswa	Hadir
0372024	Fandi	<input type="checkbox"/>
0372065	Sudanto	<input checked="" type="checkbox"/>

At the bottom left is a button "<< Back To Main Form" and at the bottom right is a "Save" button.

Gambar 17 : *EditPresensiForm*

Operator dapat mengubah data presensi yang sudah dilakukan. Sebelum mengubah data presensi, operator diminta untuk memasukkan data semester, jurusan, mata kuliah, kelas, dan pertemuan ke yang datanya akan diubah.

Setelah data presensi ditampilkan, operator hanya perlu mengubah data sesuai yang diinginkan pada kolom kehadiran. Setelah selesai melakukan perubahan data dan menekan tombol *Save*, maka data presensi di dalam *database* akan ter-*update*.



Gambar 18 : *LogInForm*

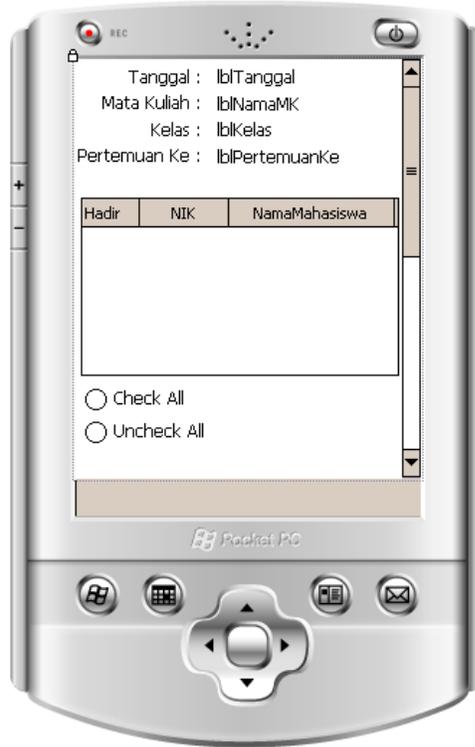
User yang dapat melakukan presensi mahasiswa dari PDA adalah *user* yang bertindak sebagai dosen. Dosen harus memasukkan NIK dan *password* untuk masuk ke dalam aplikasi ini. *Log In* yang berhasil akan menampilkan panel di bawahnya yang berisi nama semester, nama mata kuliah, dan nama kelas dari dosen yang bersangkutan. Dosen memilih data yang ada untuk kemudian melakukan presensi mahasiswa sesuai dengan mata kuliah dan kelas yang dipilih.

Selain untuk melakukan *log in*, pada halaman ini juga memungkinkan *user* untuk melakukan sinkronisasi data. Tidak perlu *log in*, *user* hanya perlu menekan tombol *Synchronize*, yang akan menjalankan fungsi pengiriman data. Data yang dikirim dari *server* ke *client* adalah data – data manajemen data yang berhubungan dengan presensi mahasiswa, seperti data pegawai, mahasiswa, kelas, dan lain – lain. Sedangkan data yang dikirim dari *client* ke *server* adalah data presensi mahasiswa, termasuk di dalamnya berita acara dan tanda tangan.

Pada saat *user* yang akan *log in* memilih *field*, aplikasi akan menampilkan *virtual keyboard* secara otomatis yang memudahkan *user* untuk melakukan *input* data.



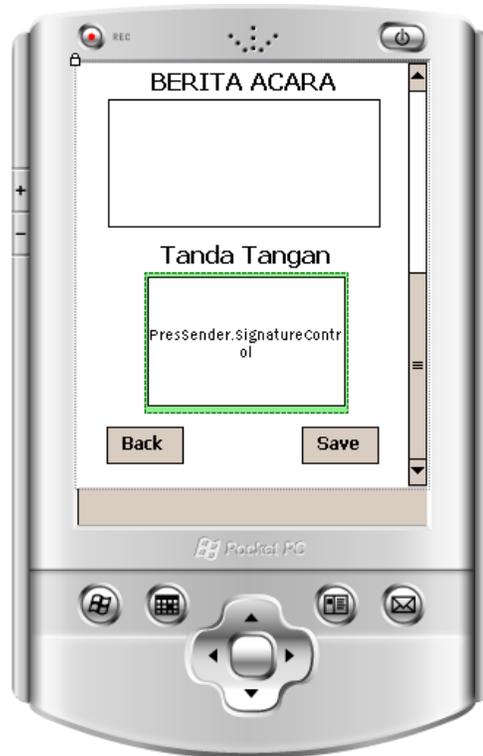
Gambar 19 : Virtual Keyboard



Gambar 20 : PresensiForm (1)

Dosen yang sudah memilih nama mata kuliah dan kelas akan dihadapkan dengan tampilan seperti pada gambar di atas. Bagian atas aplikasi akan menampilkan tanggal hari itu, nama mata kuliah yang akan dilakukan presensi mahasiswa, nama kelas, dan nama pertemuan pada saat itu.

Di bawahnya terdapat *listview* yang berisi *checkbox*, NRP, dan nama mahasiswa yang sesuai dengan nama mata kuliah dan kelas yang sudah dipilih sebelumnya. Dosen hanya perlu melakukan *check* atau *uncheck* di samping NRP dan nama mahasiswa yang ada. Disediakan juga fitur *check all* dan *uncheck all* yang akan membantu dosen dalam memilih nama mahasiswa yang hadir atau tidak hadir.

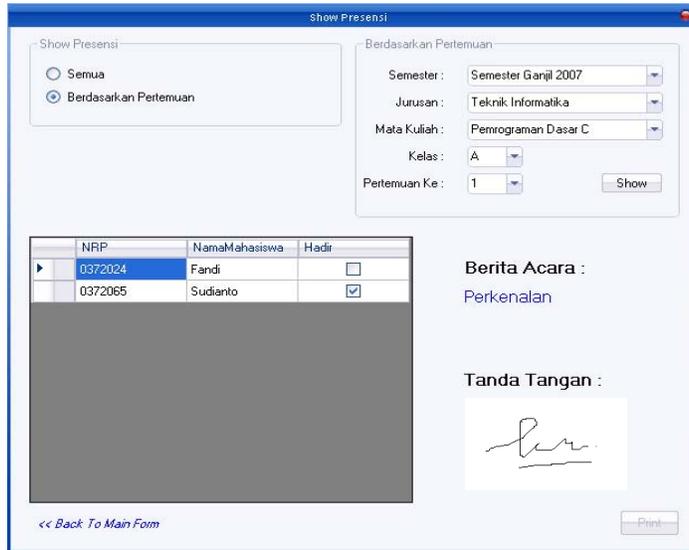


Gambar 21 : PresensiForm (2)

Selanjutnya, terdapat suatu *textarea* yang digunakan oleh dosen untuk memasukkan materi / berita acara pada perkuliahan hari itu. Terdapat juga sebuah *usercontrol* yang dapat digunakan oleh dosen untuk memberikan tanda tangan sebagai tanda bahwa pada hari tersebut dosen sudah mengajar materi tertentu, dan sudah tampak dari tanda tangan yang ada.

Data presensi yang sudah ada kemudian disimpan dengan menekan tombol *Save*. Data akan tersimpan di dalam *database* PDA samapi pada suatu kondisi di mana *user* melakukan sinkronisasi data. Data presensi akan dipindahkan ke *server* dan tersimpan di dalam *database* PC.

Data tanda tangan dosen akan diubah menjadi bentuk *binary* sebelum disimpan ke dalam *database*. Hal ini dilakukan untuk memudahkan dalam proses penyimpanan data ke dalam *database*.



Gambar 22 : ShowPresensiForm

Data presensi, termasuk di dalamnya data berita acara dan tanda tangan dosen, dapat dilihat berdasarkan nama pertemuan yang sudah pernah dilakukan.

Data tanda tangan yang ditampilkan akan diubah kembali menjadi bentuk *image* agar dapat ditampilkan ke dalam suatu *picturebox*.

3.5. Evaluasi

Hasil uji coba terhadap masing - masing fitur di atas menunjukkan bahwa sudah berjalan dengan baik dan menampilkan hasil yang diinginkan sehingga dapat disimpulkan bahwa implementasi ini dengan metode sinkronisasi data sudah berjalan dengan baik antara *database server* dan *database client*. Selain itu, proses presensi mahasiswa sudah dapat dikelola dengan baik, dari proses pembuatan kelas, manajemen data yang berhubungan, presensi mahasiswa itu sendiri, menampilkan data presensi secara keseluruhan maupun berdasarkan pertemuan, maupun pengubahan data presensi jika ada kesalahan dalam melakukan presensi mahasiswa.

4. Kesimpulan dan Saran

4.1. Kesimpulan

Presensi Manager dan *PresSender* adalah aplikasi *desktop* dan PDA yang dikembangkan untuk membantu proses presensi mahasiswa dan manajemen presensi mahasiswa. Proses manajemen yang dimaksud diantaranya adalah data mahasiswa, data pegawai, data mata kuliah, data jurusan, data semester, data kelas, data kelas mahasiswa, dan data presensi itu sendiri. Aplikasi juga dilengkapi dengan laporan – laporan yang memudahkan dalam pencetakan data.

Faktor – faktor yang dianalisa adalah :

1. Kesalahan manusia dapat dikurangi dengan adanya *error handling* pada aplikasi.
2. Pekerjaan yang biasanya menyita banyak waktu seperti proses presensi mahasiswa secara manual dan pembuatan laporan sudah diangani oleh aplikasi ini.
3. Efisiensi waktu dan tenaga akan sangat terasa jika dibandingkan antara sistem yang terkomputerisasi dan sistem yang masih menggunakan cara manual.

4.2. Saran

Saran – saran yang perlu dipertimbangkan agar aplikasi yang dikembangkan dapat berjalan lebih baik lagi di masa mendatang adalah :

4. Kemampuan untuk melakukan *back up database* secara otomatis pada jangka waktu tertentu untuk menghindari kehilangan data yang diakibatkan rusaknya sistem.
5. Aplikasi dapat dikembangkan sedemikian rupa agar dapat menghasilkan informasi seperti :
 1. Laporan data presensi per periode
 2. Laporan data kelas per periode
 3. Data *history*

4.3. Rencana Perbaikan

Rencana perbaikan yang akan dilakukan berdasarkan saran yang diberikan adalah :

1. Penambahan fitur untuk melakukan *back up database* secara otomatis pada jangka waktu tertentu.
2. Penambahan fitur – fitur yang dapat mempermudah *user* dalam menganalisa permasalahan, seperti pembuatan laporan data presensi per periode, laporan data kelas per periode, dan data *history*.

DAFTAR PUSTAKA

- [Ano2003] Koordinasi, Integrasi, Sinkronisasi, Simplifikasi (K-I-S-S). Available: <http://www.hangtuah.ac.id/Sapto/kiss.html>
- [Ano 2002] Sinkronisasi.
Available: <http://bebas.vlsm.org/v06/Kuliah/SistemOperasi/BUKU/SistemOperasi/ch18.html#c41801>
- [Ano 2002] Konsep Interaksi.
Available: <http://bebas.vlsm.org/v06/Kuliah/SistemOperasi/BUKU/SistemOperasi/ch17s06.html>
- [Ano 2007] *Web Service*.
Available: http://id.wikipedia.org/wiki/Web_service
- [Ano 2007] Wi-Fi.

Available: <http://id.wikipedia.org/wiki/Wi-Fi>

- [Ano 2005] *Pocket PC 2003 : Saving the Signature an a Bitmap.*
Available: <http://www.csharpcorner.com/UploadFile/pinto4378/SaveasBitmap12022005011654AM/SaveasBitmap.aspx>
- [Ano 2004] *Pocket PC Signature Application Sample.*
Available: <http://www.codeproject.com/netcf/PPCSignatureApp.asp>
- [Ano 2004] *Database Application Programming for the Pocket PC using SQL CE.*
Available: <http://www.codeproject.com/netcf/DBApplicationsForPocketPC.asp>
- [Ano 2006] *Create a SQL Mobile database part 1.*
Available: <http://netcf2.blogspot.com/2006/01/create-sql-mobile-database-part-1.html>
- [Ano 2006] *Data Binding in .Net CF Application.*
Available: http://msdn.microsoft.com/vstudio/tour/vs2005_guided_tour/VS2005pro/Smart_Client/DataBinder.htm
- [Sha2005] Sharp, John (2005) “*Microsoft Visual CSharp 2005 Step by Step*”, MSPress
- [Amb2006] W. Ambler, Scott (2003 – 2006) “*UML 2.0 Tutorial*”
- [Tro2005] Troelsen, Andrew (2005) “*Pro C# 2005 and .Net 2.0 Platform*”, Apress