

APLIKASI PRESENSI KELAS KULIAH DENGAN *NEAR FIELD COMMUNICATION* (NFC) PADA ANDROID

Andreas Handojo, Julius Wonodihardjo, Justinus Andjarwirawan
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) - 8417658
E-mail: handojo@petra.ac.id²

ABSTRAK:

Saat ini, presensi kelas masih menggunakan kertas pada setiap perkuliahan. Daftar presensi sering kali diberikan kepada ketua kelas untuk mengedarkan presensi. Dosen hanya memberikan paraf hanya sebagai bukti jalannya perkuliahan tersebut dan tidak melakukan pengecekan kehadiran peserta. Sistem presensi yang dijalankan tidak menggunakan teknologi sebagai alat bantu.

Berdasarkan latar belakang permasalahan itu, sistem presensi dirancang dengan menggunakan NFC sebagai alat bantu dosen untuk mengecek kehadiran mahasiswa dengan menggunakan perangkat Android. Sistem pengecekan dilakukan dalam bentuk total presensi pada tanggal tertentu maupun siapa saja yang hadir atau tidak hadir dalam perkuliahan tersebut. Mahasiswa dapat mengirimkan informasi presensi melalui NFC dan dapat mengecek jumlah kehadiran mahasiswa tersebut.

Hasil yang diperoleh dari pengujian yang telah dilakukan, *prototype* ini dapat menjalankan sistem presensi melalui NFC, mengecek peserta yang hadir, dan mengecek jumlah kehadiran mahasiswa.

Kata Kunci: *Near Field Communication*, Presensi, JSON, Teknologi, Android.

ABSTRACT: *Currently, class attendance is still using paper in every courses. Attendance list is often granted to the class representative to distribute the attendance list. Lecturer just give their initials only as evidence of the courses of the lecture and does not check the presence of the students. Attendance system that run does not use technology as a tool.*

Based on the background, the attendance system designed using Near Field Communication (NFC) as a tools to checking the presence of students using Android devices. Checking system carried out in the form of attendance totals in certain dates and who else was present or not present in the lecture. Students can submit their attendance information via NFC and can checking their attendance.

The results of the testing has been done, the prototype system can be running through the NFC, attendees check, and checking the number of attendances.

Keywords: *Near Field Communication, Attendance, JSON, Technology, Android.*

1. PENDAHULUAN

Pada saat ini, dunia teknologi telah berkembang begitu cepat. Segala macam teknologi dibuat untuk memberikan kemudahan kepada pengguna teknologi. Perkembangan teknologi ini memudahkan setiap manusia untuk berkomunikasi satu sama lain, dimana ketika mengirim pesan jarak jauh hanya membutuhkan waktu tidak lebih dari satu menit untuk sampai tujuan. *Smartphone* yang saat ini telah berhasil menggeser kedudukan *personal computer* dan *notebook*. Secara tidak langsung, banyak *vendor-vendor* bersaing untuk memberikan teknologi terkini untuk ditanamkan pada *smartphone* dan *tablet*. Begitu pula dengan teknologi *infrared* yang telah digantikan oleh *bluetooth* dan setelah itu berkembang NFC yang sekarang ditanamkan pada hampir setiap *smartphone*.

Presensi kelas yang saat ini masih menggunakan kertas setiap kali mahasiswa melakukan presensi. Pengecekan presensi setiap mata kuliah jarang untuk dilakukan oleh dosen karena presensi kelas tersebut dilakukan dengan mengedarkan presensi dari satu mahasiswa ke mahasiswa lain.

Dari permasalahan tersebut, maka dibuatlah sebuah aplikasi presensi kelas berbasis Android dengan menggunakan teknologi *Near Field Communication* (NFC). Aplikasi ini akan memberikan informasi kelas yang diambil oleh mahasiswa. Informasi tersebut dikirimkan ke NFC penerima dan dikirim ke *server* untuk hasil presensi kelas.

2. LANDASAN TEORI

2.1 Pengertian NFC

NFC (*Near Field Communication*) merupakan spesifikasi standar untuk *smartphone* dan *device* yang serupa untuk membangun komunikasi radio antar *device* dengan cara menempelkan atau mendekatkan kedua *device* tersebut. Berdasarkan teknologi RFID, NFC menyediakan media untuk mengidentifikasi protokol yang aman saat digunakan untuk mengirim data. NFC didesain untuk dapat mengirimkan berbagai macam tipe informasi, seperti nomor telepon, gambar, *file* MP3, atau *digital authorization* antar dua NFC (antar *mobile phone* atau antara *mobile phone* dengan kartu

RFID *chip / reader* yang memiliki jarak yang cukup dekat satu sama lain) [1] [2].

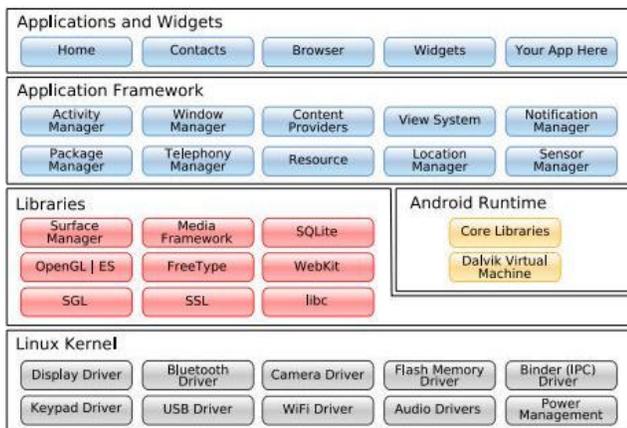
Aplikasi – aplikasi yang dapat dikembangkan untuk kehidupan sehari – hari [3]:

- Pembayaran *mobile*
 - Membayar dengan NFC *phone* untuk tiket atau biaya taksi.
 - Menyimpan *voucher* di NFC *phone*.
- Autentikasi, akses kontrol – kunci toko elektronik, legitimasi pada NFC *phone*.
 - Keamanan akses gedung.
 - Keamanan *log-in* pada *personal computer* (PC).
 - Membuka pintu mobil.
- Pengiriman data antar unit NFC yang berbeda (pertukaran data secara *peer-to-peer*) seperti NFC *smartphone*, kamera digital, *notebook*.
 - Pertukaran kartu bisnis elektronik.
 - Mencetak foto dengan mendekati kamera ke printer.
- Membuka *service* lain (seperti membuka *communication link* untuk mengirim data).
 - Menyiapkan Bluetooth, WLAN *link*.

2.2 Pengertian Android

Android merupakan sistem operasi berbasis Linux dan menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Pada awalnya, Google Inc. membeli Android Inc., yang membuat *software* untuk ponsel. Kemudian, untuk mengembangkan Android, dibentuklah Open Handset Alliance, yang terdiri dari 34 perusahaan *hardware*, *software*, dan telekomunikasi, termasuk Google Inc., HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [4].

Arsitektur Android terdiri dari *Applications*, *Application Framework*, *Libraries*, *Android Runtime* dan *Kernel Linux* dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur Android

2.3 The Dalvik Virtual Machine (DVM)

Salah satu elemen kunci dari Android adalah *Dalvik Virtual Machine* (DVM). Android berjalan didalam DVM, bukan pada *Java Virtual Machine* (JVM). DVM adalah “*register based*”, sementara JVM adalah “*stack based*”. DVM menggunakan *kernel Linux* untuk menangani fungsionalitas tingkat rendah, termasuk keamanan, *threading*, proses, serta manajemen memori.

Semua *hardware* yang berbasis Android dijalankan dengan menggunakan *Virtual Machine* untuk menjalankan aplikasi [4].

2.4 Android SDK (Software Development Kit)

Android SDK adalah *tools API* (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh Google. Beberapa fitur-fitur Android yang paling penting adalah [4]:

- *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
- *Virtual Dalvik Machine* dioptimalkan untuk perangkat *mobile*.
- *Integrated browser* berdasarkan *engine open source WebKit*.
- Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, 3D berdasarkan spesifikasi OpenGL 1.0.
- SQLite untuk penyimpanan data.
- *Media support* yang mendukung *audio*, *video*, dan gambar (MPEG4, MP3, JPG, PNG, GIF), GSM *Telephony* (tergantung *hardware*).
- Bluetooth, EDGE, 3G, dan WiFi (tergantung *hardware*).
- Kamera, GPS, kompas, *accelerometer* (tergantung *hardware*).

2.5 Android Development Tools (ADT)

Android Development Tools (ADT) adalah *plugin* yang didesain untuk IDE Eclipse yang memberikan kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse, akan lebih mudah dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya. Selain itu, dengan ADT, dapat membuat *package* Android (.apk) yang digunakan untuk mendistribusikan aplikasi Android yang telah dibuat [4].

2.6 JavaScript Object Notation (JSON)

JSON (*JavaScript Object Notation*) merupakan format untuk pertukaran data seperti halnya XML. Penggunaan JSON mudah dipahami oleh manusia karena formatnya yang cukup sederhana. Terdapat kesamaan dan perbedaan dengan XML. Kesamaan dengan XML adalah sebagai berikut [5]:

- JSON merupakan *plain text*.
- JSON mudah dibaca oleh manusia.
- Dapat diparsing oleh *JavaScript*.

Perbedaan dengan XML adalah sebagai berikut [5]:

- Tidak terdapat *tag*.
- Lebih pendek.
- Lebih cepat untuk dibaca dan ditulis
- Menggunakan *array*.

2.7 IMEI (*International Mobile Station Equipment Identity*)

IMEI merupakan nomor unik yang dialokasikan pada setiap peralatan *mobile* dalam PLMN (*Public Land Mobile Network*) dan harus diimplementasikan tanpa syarat oleh produsen MS. Nomor IMEI digunakan oleh jaringan GSM untuk mengidentifikasi validasi perangkat dan dapat digunakan untuk menghentikan akses jaringan pada perangkat yang dicuri sehingga tidak dapat digunakan kembali.

Nomor IMEI terdiri dari 15 angka (14 angka dan 1 angka pengecekan) mengandung informasi tentang asal-usul, model, dan nomor serial dari perangkat. Struktur dari IMEI ditentukan dalam 3GPP TS 23.003 [6]. Format dari IMEI adalah AA-BBBBBB-CCCCCC-D [7]:

- AA : Dua digit ini untuk *Reporting Body Identifier* (TAC).
- BBBBBB : Sisa dari TAC (FAC).
- CCCCCC : Urutan serial dari model (SNR).
- D : Sebuah *check digit* berdasarkan formula Luhn.

3. DESAIN SISTEM

3.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem didasarkan pada relasi antar perangkat Android dan *server*.



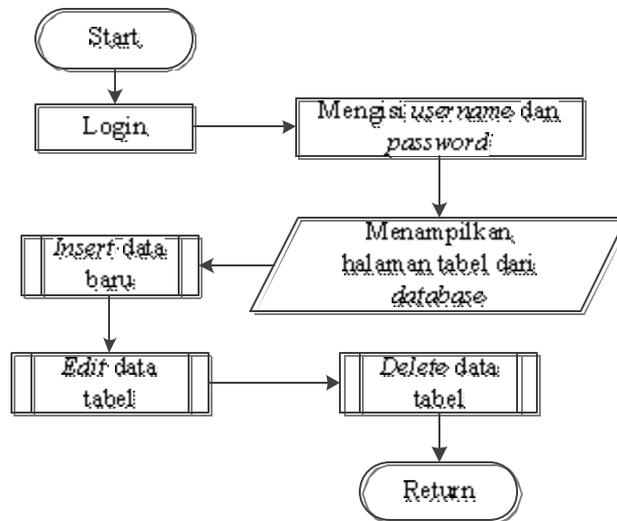
Gambar 2. Perancangan Arsitektur Sistem

Dapat dilihat pada Gambar 2, relasi antara Android dosen dengan *server* adalah meminta autentikasi *login* dan mengirimkan data presensi ke *server* ketika Android dosen menerima informasi presensi dari Android mahasiswa. *Device* Android dosen juga meminta informasi tentang presensi, mata kuliah, dan peserta ke *server* sehingga informasi tersebut dapat ditampilkan pada *device* Android dosen.

Relasi antara Android mahasiswa dengan *server* adalah meminta autentikasi *login* dan menerima informasi tentang presensi maupun mata kuliah dari mahasiswa tersebut. Hubungan antara Android mahasiswa dengan dosen adalah Android mahasiswa mengirimkan informasi presensi mahasiswa ke Android dosen melalui NFC.

3.2 Flowchart Sistem Pada Web Admin

Sistem pada *web admin* ini dimulai dengan proses *login* dari *admin* dengan cara mengisi *username* dan *password*. Setelah menekan tombol *login*, *admin* akan masuk pada halaman yang berisi tabel-tabel pada *database*. Tabel-tabel tersebut adalah tabel *Login*, *Dosen*, *Mahasiswa*, *Mata Kuliah*, *Jadwal*, *Peserta*, dan *Presensi*. *Admin* dapat mengakses tabel *Login*, *Dosen*, *Mahasiswa*, *Mata Kuliah*, *Jadwal* dengan bantuan tombol navigasi. Tabel *Peserta* dan tabel *Presensi* dapat diakses pada kolom detail dari halaman jadwal. *Admin* dapat melakukan *insert*, *update*, dan *delete* data dari tabel-tabel tersebut. *Flowchart* sistem pada *web admin* dapat dilihat pada Gambar 3.

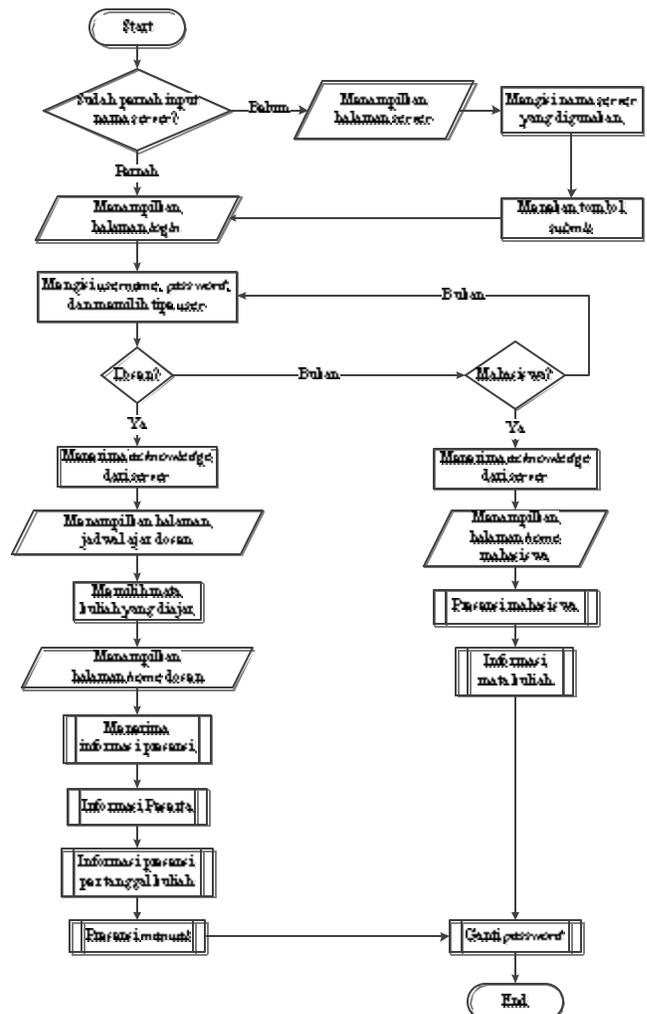


Gambar 3. Flowchart Sistem Pada Web Admin

3.3 Flowchart Sistem Pada Android

Sistem dari *prototype* ini dimulai dengan pengecekan apakah *user* pernah memasukkan nama *server* yang digunakan. Jika belum, maka *user* diminta untuk mengisi nama *server* yang akan digunakan. Jika pernah, maka *user* akan langsung masuk kedalam *menu login* dan mengisi *username*, *password*, dan tipe *user*. Setelah mengisi *username*, *password*, dan tipe *user*, *user* menekan tombol *login* untuk dapat masuk ke *menu* berikutnya. Jika tipe dari *user* adalah dosen, maka akan ditampilkan halaman yang berisikan *list view* dari mata kuliah yang diajarkan oleh *user* tersebut.

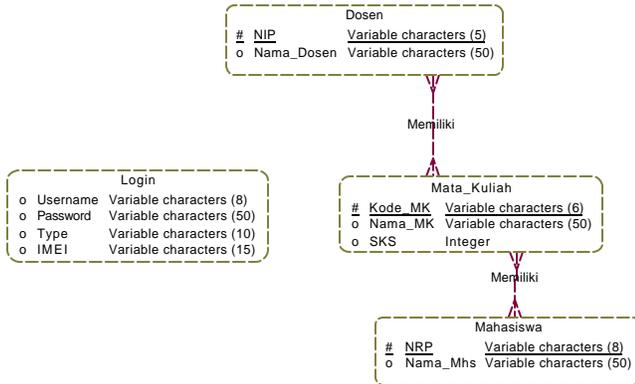
Dalam tipe *user* dosen, dibagi menjadi lima sub proses, yaitu sub proses menerima informasi presensi, sub proses informasi peserta, sub proses informasi presensi per tanggal kuliah, sub proses presensi *manual*, dan sub proses ganti *password*. Jika tipe *user* adalah mahasiswa, maka akan ditampilkan halaman *home* mahasiswa. Dalam tipe *user* mahasiswa, dibagi menjadi tiga sub proses, yaitu sub proses presensi mahasiswa, sub proses informasi mata kuliah, dan sub proses ganti *password*. Flowchart sistem pada Android dapat dilihat pada Gambar 4.



Gambar 4. Flowchart Sistem Pada Android

3.4 Desain Struktur Database

Dalam mendesain sebuah sistem dari *database*, diperlukan *Entity-Relationship Diagram* (ERD) untuk membuat *entity* yang dibutuhkan dalam penyimpanan data dan relasi antara *entity* satu dengan *entity* yang lain. *Entity Relationship Diagram* yang dirancang memberikan gambaran detail tentang *database* yang digunakan dalam pembuatan *prototype* ini. Desain struktur *database* yang digunakan dapat dilihat pada Gambar 5.



Gambar 5. Entity-Relationship Diagram

4. IMPLEMENTASI SISTEM

4.1 Implementasi Pada Server

Implementasi pada *server* berperan sebagai perantara komunikasi antara *prototype* Android dengan *database* yang terdapat dalam *server*. Seluruh proses *read*, *insert*, *update*, dan *delete* terhadap *database* dilakukan dalam bentuk fungsi pada PHP. *Prototype* Android hanya melakukan *request* dan *response* menggunakan fungsi JSON. Adapun komunikasi antara *server* dengan Android dilakukan dengan menggunakan metode HTTP GET.

4.2 Implementasi Prototype Pada Android

Implementasi dari *prototype* yang dijelaskan pada bagian ini adalah bagian NFC. Pada bagian ini, fungsi *NdefMessage createNdefMessage* digunakan untuk mengirimkan *message*. *Message* yang dikirim harus dienkapsulasi menjadi *byte* menggunakan fungsi *NdefRecord createMime*. Parameter dari fungsi *NdefRecord* berupa *mimeType* dan *message* yang akan dikirim. *Message* tersebut dikirim melalui NFC dengan fungsi *onNdefPushComplete* ke NFC penerima. Dalam fungsi tersebut, dijalankan juga fungsi *handler* untuk menampilkan pesan “*Message sent*”. Fungsi *createNdefMessage* dapat dilihat pada *Pseudocode 1*.

Pseudocode 1. createNdefMessage

```

public NdefMessage
createNdefMessage(NfcEvent arg0) {

send_msg_nfc = Get NRP, Nama Mata
                Kuliah, Kelas, NIP Dosen,
                Tanggal from TextView;
    
```

```

NdefMessage msg = new NdefMessage(
    new NdefRecord[] { createMime(
        "application/packages_name",
        Get send_msg_nfc bytes)
    });
return msg;
    
```

Pada perangkat Android dosen, informasi NFC diterima dalam bentuk *byte* dikembalikan menjadi *string*. Informasi diterima oleh fungsi *processIntent* berupa *raw message* dan dikonversi kedalam bentuk *string*. *String* tersebut kemudian pisahkan dengan menggunakan fungsi *split*, dimasukkan kedalam *array*, dan dimasukkan kedalam variabel yang tersedia untuk dikirimkan langsung ke *database*. Presensi berhasil dilakukan jika muncul *toast* dengan pesan presensi sukses dan disusul oleh suara presensi sukses. Fungsi *processIntent* dapat dilihat pada *Pseudocode 2*.

Pseudocode 2. processIntent

```

void processIntent(Intent intent) {
Parcelable[] rawMsgs =
intent.getParcelableArrayExtra(
    Extra containing an array of
    NdefMessage present on the
    discovered tag);

NdefMessage msg = get array 0 from the
rawMsgs;

String payload = new String(
    Get array 0 from the message
record);

String[] separated =
    Split payload string to define
    nrp, kode mata kuliah, kelas,
    nip, tanggal, and status based
    on ":";

Set result of the separated to the new
variable based on array index;

AbsenTask task = new AbsenTask();
try {
    Get string from the variable that
    have been separated, encode the
    strings, and set to the new variable;

String url = "URL to send the
variables to the server";

task.execute(url);
    
```

```

} catch(
UnsupportedEncodingException e) {
    e.printStackTrace();
}
}

```

5. PENGUJIAN SISTEM

Pengujian ini dilakukan dengan menggunakan *smarphone* Sony Xperia Z, Sony Xperia Sola, Samsung Galaxy Note II, dan Samsung Galaxy Nexus untuk melakukan pengujian *prototype*. Spesifikasi dari *device* yang digunakan saat pengujian dapat dilihat pada Tabel 1.

Tabel 1. Spesifikasi Device Pengujian

Device	Operating System	Display size	CPU	Internal Memory
Sony Xperia Z	Android 4.1.2 (Jelly Bean)	1080 x 1920 pixels, 5.0 inches	Quad-core 1.5 GHz Krait	16 GB storage, 2 GB RAM
Sony Xperia Sola	Android 4.0.4 (Ice Cream Sandwich)	480 x 854 pixels, 3.7 inches	Dual-core 1.0 GHz Cortex-A9	8 GB storage, 512 MB RAM
Samsung Galaxy Note II	Android 4.1.2 (Jelly Bean)	720 x 1280 pixels, 5.5 inches	Quad-core 1.6 GHz Cortex-A9	16 GB storage, 2 GB RAM
Samsung Galaxy Nexus	Android 4.2.1 (Jelly Bean)	720 x 1280 pixels, 4.65 inches	Dual-core 1.2 GHz Cortex-A9	16 GB storage, 1 GB RAM

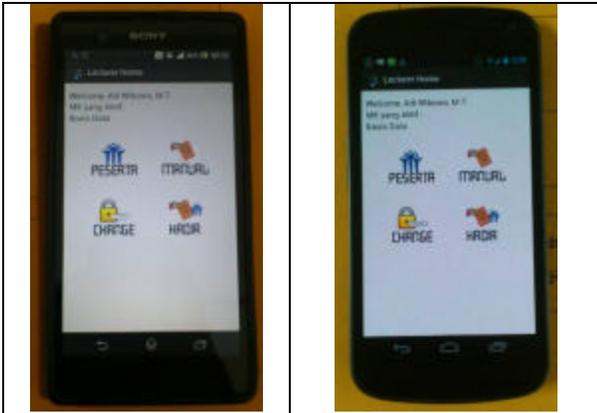
5.1 Pengujian Pada Prototype Android

Proses pengujian ini dilakukan pada *device* Android untuk melihat hasil yang telah didesain. Pengujian yang dilakukan pada Android adalah sebagai berikut.

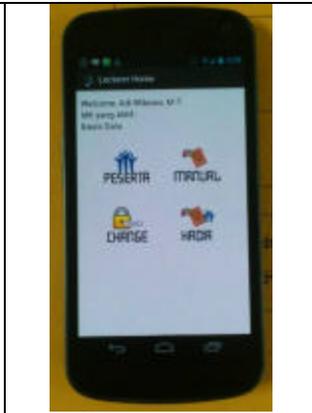
5.1.1 Pengujian *Prototype* dengan Hak Akses Dosen
 Dosen harus melakukan *login* untuk dapat menerima presensi dari mahasiswa, melihat mata kuliah yang diajar, dan melihat peserta kuliah. Pengujian *prototype* pada halaman *list* jadwal dari dosen dapat dilihat pada Gambar 6, tampilan dosen dapat dilihat pada Gambar 7, tampilan *list* kehadiran sebelum *device* dosen menerima presensi pada Gambar 8, tampilan saat menerima hasil presensi pada Gambar 9, dan tampilan *list* kehadiran saat *device* dosen telah menerima informasi dari *device* mahasiswa pada Gambar 10.



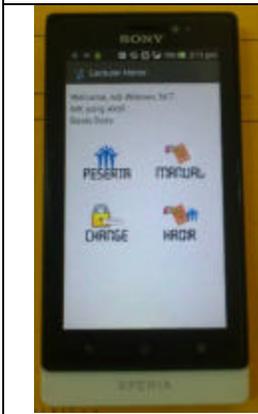
Gambar 6. Pengujian Halaman List Jadwal Dosen



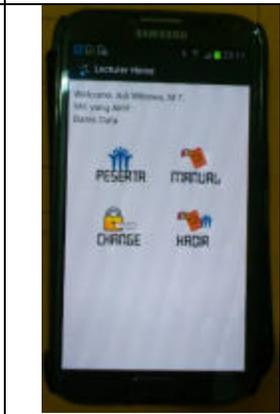
Sony Xperia Z



Samsung Galaxy Nexus

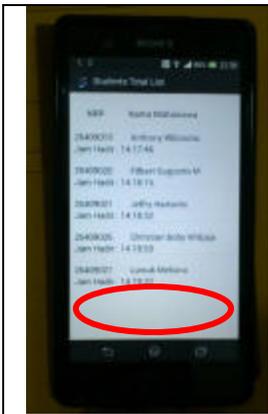


Sony Xperia Sola

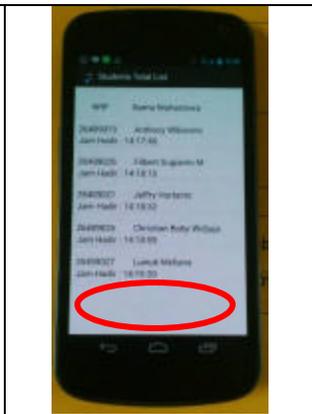


Samsung Galaxy Note II

Gambar 7. Pengujian Halaman Dosen



Sony Xperia Z



Samsung Galaxy Nexus

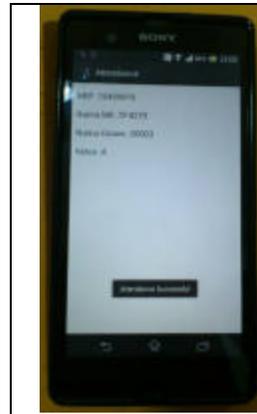


Sony Xperia Sola

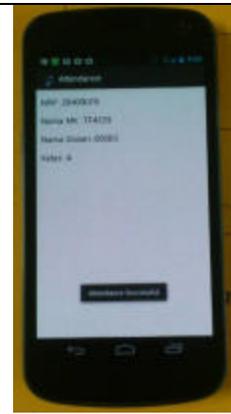


Samsung Galaxy Note II

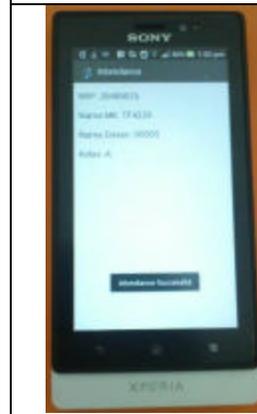
Gambar 8. Pengujian Halaman List Kehadiran



Sony Xperia Z



Samsung Galaxy Nexus

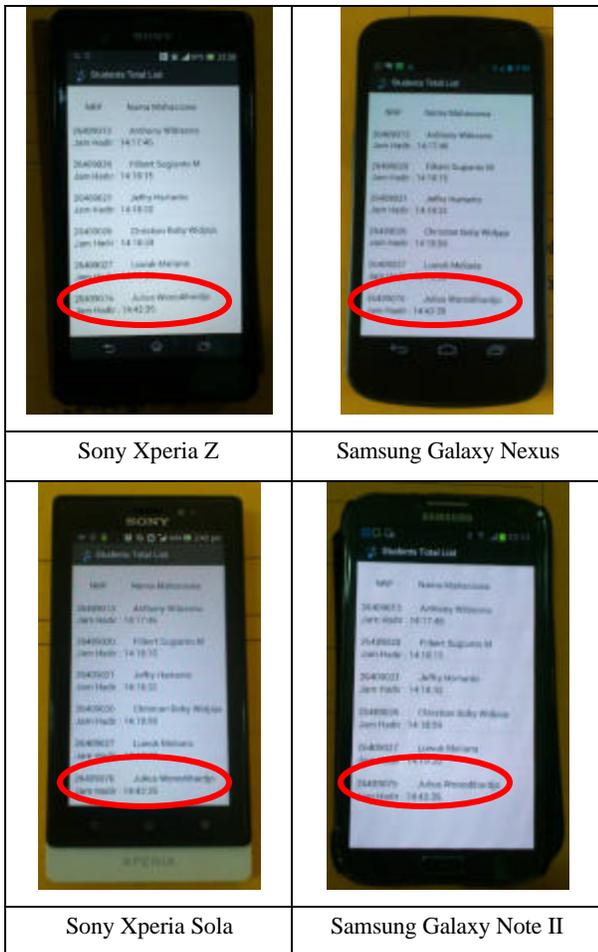


Sony Xperia Sola



Samsung Galaxy Note II

Gambar 9. Pengujian Halaman Presensi Penerima



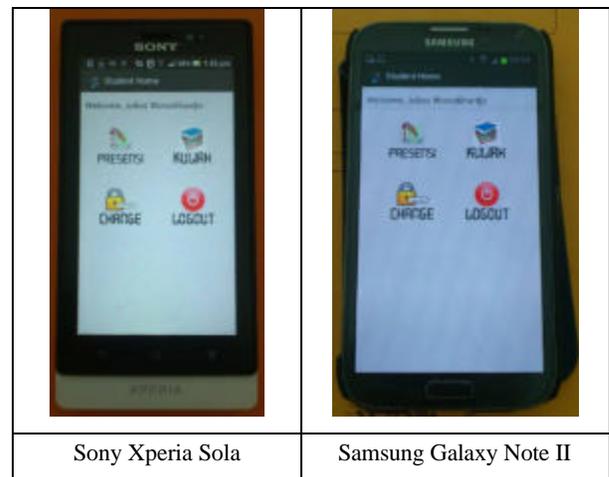
Gambar 10. Pengujian Halaman *List Kehadiran (Data Baru)*

5.1.2 Pengujian *Prototype* dengan Hak Akses Mahasiswa

Mahasiswa harus melakukan *login* untuk dapat melakukan presensi, melihat mata kuliah yang diambil, dan melihat jumlah presensi mata kuliah. Pengujian *prototype* pada tampilan mahasiswa dapat dilihat pada Gambar 11, tampilan mata kuliah dapat dilihat pada Gambar 12, tampilan informasi presensi yang akan dikirim pada Gambar 13, dan tampilan *interface* saat *device* dari mahasiswa didekatkan pada *device* Android dosen pada Gambar 14.

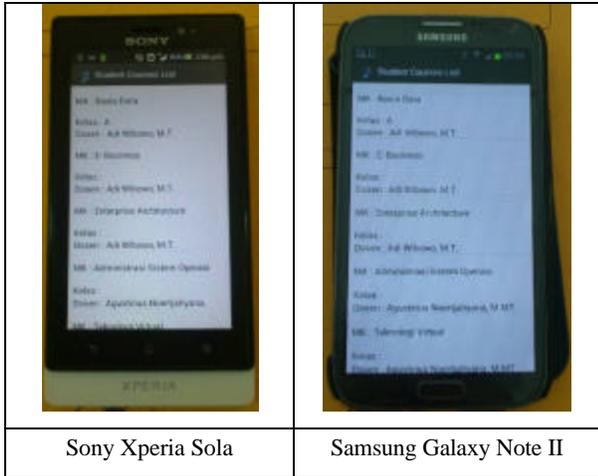


Gambar 11. Pengujian Halaman Mahasiswa



Gambar 11. Pengujian Halaman Mahasiswa (sambungan)

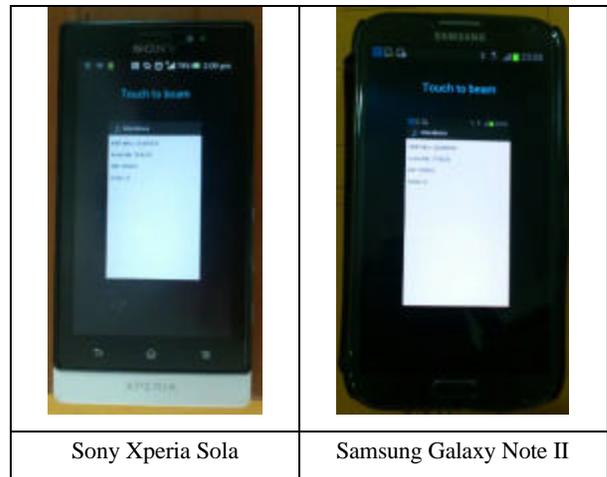




Gambar 12. Pengujian Halaman Mata Kuliah



Gambar 14. Pengujian Halaman Presensi saat akan dikirim



Gambar 14. Pengujian Halaman Presensi saat akan dikirim (sambungan)



Gambar 13. Pengujian Halaman Presensi Pengirim

6. KESIMPULAN

Dari proses perancangan sistem hingga pengujian *prototype* yang telah dilakukan, dapat ditarik beberapa kesimpulan yaitu sebagai berikut.

- NFC pada *prototype* dapat digunakan untuk mengirimkan informasi presensi dengan baik.
- Kecepatan *internet* yang buruk mempengaruhi kinerja dari *prototype*, seperti mengirimkan informasi presensi ke *server*.
- Berdasarkan hasil pengujian terhadap beberapa *smartphone*, fitur presensi melalui NFC dapat berjalan dengan baik pada sistem operasi Android versi 4.0.4 (*Ice Cream Sandwich*) dan Android versi 4.1.1 – 4.1.2 (*Jelly Bean*).
- Sistem presensi yang dirancang telah berjalan dengan baik.

7. REFERENSI

- [1] NFC Forum. (2012). *About the Forum*. Retrieved December 20, 2012, from <http://www.nfc-forum.org/aboutus/>

- [2] NFC World. (2012). *About NFC*. Retrieved November 28, 2012, from <http://www.nfc-world.com/en/about/01.html>
- [3] Rohde & Schwarz *NFC technology and measurements*. (2011). *White Paper*. Retrieved January 18, 2013
- [4] Safaat, Nazruddin. (2012). *Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Penerbit Informatika.
- [5] w3school.com. (2012). *JSON Tutorial*. Retrieved May 23, 2013, from <http://www.w3schools.com/json/>
- [6] 3GPP. (2009). *Technical Specification Group Services and System Aspects (Release 9)*. Retrieved June 27, 2013, from http://www.3gpp.org/ftp/Specs/archive/22_series/22.016/22016-900.zip
- [7] GSM Association. (2010). *IMEI Allocation and Approval Guidelines Version 5.0*. Retrieved June 27, 2013, from <http://imeidb.gsm.org/imei/DG06.pdf>