

PERANCANGAN APLIKASI TAG WRITER DENGAN TEKNOLOGI NEAR FIELD COMMUNICATION BERBASIS ANDROID

Ferry Satria

Staf Pengajar Jurusan Teknik Elektro Politeknik Negeri Bandung

E-mail: ferrypolban@gmail.com

Abstrak

Perancangan Aplikasi *Tag Writer* dengan Teknologi *Near Field Communication (NFC)* berbasis Android merupakan pengembangan dari aplikasi teknologi *Radio Frequency Identification (RFID)*. Pada teknologi ini dimungkinkan sebuah *device* berinteraksi dengan komponen pasif NFC yang tidak mempunyai tenaga (suplai energi) sendiri. Agar dapat terjadi interaksi dengan komponen pasif tersebut, dibutuhkan *device* berupa *smartphone* yang dapat memancarkan medan elektro magnetik (radio frequency) sehingga mampu mengimbaskan tenaga pada komponen pasif tersebut. Pada penelitian ini telah dibuat perangkat lunak aplikasi untuk *smartphone* android yang dirancang dengan bantuan *perangkat lunak* eclipse sebagai *ide (integrated development environment)*. Sistem ini telah diuji menggunakan 3 macam format yaitu *text*, *contact*, dan URL. Operasi yang digunakan pada proses pengujian tersebut adalah: *read* dan *write*. Sebagai objek target dari proses *write* dan *read* adalah komponen pasif berupa kartu RFID dan *smart_tag* NFC. Masing-masing komponen mempunyai kapasitas memori yang terbatas. Pada *smart tag* NFC hanya ada 46 bytes, sedangkan pada kartu RFID, tersedia 716 bytes. Proses *write* dan *read* memerlukan waktu antara 1 hingga 120 detik tergantung kepada panjang data yang dimasukkan. Manfaat dari penelitian ini adalah, pertama: sebagai media untuk membuat penyimpanan data dimana *file contact* yang sudah di *write* dapat langsung diimport kedalam aplikasi *contact*, ke dua: sebagai alat untuk membuat media promosi (URL), dan ke tiga: sebagai pemberian data identifikasi pemilik (data text) pada benda yang dimilikinya. Hasil pengujian menunjukkan bahwa *file contact*, *text*, dan URL sudah dapat di *write* pada *smart-tag* NFC dan data yang ditulis telah berhasil di *read* dari *smart-tag* NFC tersebut sesuai dengan data yang telah dimasukkan kedalamnya.

Kata kunci : NFC, Read, Write, Smart Tag, RFID, Smartphone, Android

Pendahuluan

Teknologi *Near Field Communication (NFC)* merupakan teknologi pengembangan dari *Radio-frequency identification (RFID)* yang memanfaatkan koneksi jarak dekat menggunakan induksi medan magnet. Para pengguna *smartphone* dapat menikmati berbagai macam fasilitas seperti penyimpanan *file contact*, pembacaan *text*, bahkan dengan menggunakan teknologi ini sambungan langsung ke sebuah *website* dapat dilakukan tanpa harus mengetik alamat tujuan URL yang akan dihubungi dari *smartphone*-nya. Pada aplikasinya di dunia nyata, data berupa *Uniform Resource Locator (URL)* yang dimasukkan dapat dipergunakan sebagai media promosi, contohnya: URL dapat dimasukkan kedalam sebuah poster film yang dipasang pada

tempat umum, sehingga saat peminat film tersebut melakukan *read* data pada poster tersebut, ia akan dapat langsung dihubungkan ke *website* dari film tersebut. Untuk aplikasi *contact*, dapat digunakan sebagai kartu nama elektronik, sehingga data *contact* yang dimiliki pengguna bisa dimasukkan pada kartu RFID ataupun *smart_tag* yang dilekatkan pada gantungan kunci sehingga memudahkan pemberian nomor telepon. Salah satu aplikasi dari data text adalah dapat dipergunakan pada *smart_tag* yang ditempelkan pada suatu gantungan kunci. Data identifikasi dari pemilik kunci berupa data text dapat di *write* pada *smart Tag NFC*. Jika suatu saat kunci hilang, maka orang yang menemukannya dapat melakukan *read* pada *smart Tag* untuk mengetahui identitas si pemilik kunci, dan dapat segera

menghubungi pemilik kunci untuk mengembalikan kunci tersebut.

Tinjauan Pustaka

Aplikasi *tag writer* memerlukan *smartphone android* yang telah dilengkapi dengan teknologi NFC yang berfungsi sebagai media untuk proses *write* dan *read* pada komponen pasif NFC. Penggunaan *smartphone* tersebut adalah sebagai sumber energi agar komponen pasif NFC dapat menerima data yang akan di *write* oleh aplikasi dari *smartphone* tersebut.

Aplikasi *tag writer* ini terbagi menjadi 2 sistem secara umum, yaitu: sistem *write* dan sistem *read*. Proses *write* secara sederhana dapat dijelaskan sebagai proses pengiriman data dari sebuah media (dalam penelitian ini dipergunakan *smartphone android* dengan teknologi NFC) ke media lainnya (dalam penelitian ini menggunakan komponen pasif NFC, kartu RFID dan *smart tag*). Sedangkan proses *Read* adalah proses membaca data yang tersimpan pada komponen pasif NFC. Salah satu aplikasi teknologi NFC, adalah dapat diaplikasikan pada sistem *smart poster*.

Aplikasi sistem *Smart Poster Writer/Reader* menggunakan *smartphone Android* yang telah dilengkapi dengan teknologi NFC. Aplikasi yang dibuat adalah aplikasi untuk melakukan proses *write* data URL pada *smart poster*. Dengan kata lain, *smart poster* dapat diisi data URL sehingga saat dilakukan proses *read* memungkinkan untuk menghubungkan sistem aplikasi ke alamat URL tujuan yang sebelumnya telah diinputkan pada unit NFC yang ditempelkan pada poster tersebut. [7]. Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan *software* dan dapat dijalankan di semua *platform*. Eclipse mempunyai beberapa terget *platform*, di antaranya adalah Windows, Linux, Mac. Eclipse dikembangkan dengan bahasa pengembangan Java, akan tetapi Eclipse juga mendukung pengembangan aplikasi lain berbasis bahasa pemrograman C/C++, Python, PHP dan lainnya[4]. Eclipse pada saat ini merupakan salah satu IDE favorit, karena selain gratis, Eclipse juga *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plugin*. Dalam penelitian ini digunakan *plugin* ADT[4].

Android adalah OS (*Operating System*) yang tersedia pada ribuan dari jutaan *mobile devices* di lebih dari 190 negara diseluruh dunia. Android merupakan OS paling banyak digunakan pada *mobile platform* dan jumlah pengguna terus meningkat dengan cepat. Setiap harinya jutaan pengguna memperbaharui *device android*-nya untuk mulai mencari aplikasi, permainan, serta konten-konten digital lainnya. Android memberikan sebuah *platform* kelas dunia untuk membuat aplikasi dan permainan kepada pengguna android dimanapun, juga merupakan sebuah pusat penjualan untuk mendistribusikannya secara instan. [2]. *Plugin Android Developer Tools* (ADT) untuk eclipse menyediakan sebuah ruang pengembangan dengan tingkat profesional untuk membuat aplikasi android. *Plugin* ini berupa sebuah JAVA IDE dengan fitur untuk membantu dalam proses *build*, *test*, *debug*, dan *package* untuk aplikasi android. *Plugin* ini tersedia secara gratis dan *open source*, serta bisa dijalankan di berbagai macam OS umum seperti Windows, Linux, ataupun Macintosh [3]. Penggunaan teknologi NFC telah meningkat untuk banyak kegunaan. Aplikasi yang ada saat ini salah satunya untuk pembayaran tanpa kontak, pemasaran dan periklanan (ajang promosi suatu produk), keamanan dan kontrol akses, identifikasi produk, aplikasi *task launcher* pada *smartphone*, dan masih banyak lagi. Keunggulan penggunaan teknologi NFC terletak pada kesederhanaannya. Jika sebuah *tag* sudah di *encoded* dengan informasi atau aksi yang diinginkan, *tag* itu sudah siap untuk digunakan[2].

Tag NFC tidak jauh berbeda dengan *tag* RFID, terdiri atas sebuah chip mikro (*microchip*) dan sebuah antena. *Chip* mikro itu sendiri dapat berukuran sekecil butiran pasir, seukuran 0.4 mm. *Chip* tersebut menyimpan nomor seri yang unik atau informasi lainnya tergantung kepada tipe memorinya. Tipe memori itu sendiri dapat *read-only*, *read-write*, atau *write-once read-many*. Antena yang terpasang pada *chip mikro* mengirimkan informasi dari *chip ke reader*. Biasanya rentang pembacaan diindikasikan dengan besarnya antena. Antena yang lebih besar mengindikasikan rentang pembacaan yang lebih jauh. *Tag* tersebut terpasang atau tertanam dalam obyek yang akan diidentifikasi. *Tag* dapat di *scan* dengan *reader* bergerak maupun stasioner menggunakan gelombang radio. [1]

Label NFC atau yang biasa disebut NFC tag, pada dasarnya merupakan suatu *microchip* berantena, yang disertakan pada suatu unit objek. Dengan piranti ini, suatu instansi atau lembaga dapat mengidentifikasi dan melacak keberadaan suatu data.

Seperti halnya barcode, yang memiliki *Universal Product Code* (UPC), sebuah tag NFC memiliki *Electronic Product Code* (EPC) berisi identitas produk tersebut, mulai dari nomor seri, tanggal produksi, lokasi manufaktur, bahkan tanggal kadaluarsa. EPC adalah identifikasi produk generasi baru, mirip dengan UPC atau barcode. Seperti halnya barcode, EPC terdiri dari angka-angka yang menunjukkan kode produsen, produk, versi dan nomor seri. Namun, EPC memiliki digit ekstra untuk mengidentifikasi item yang unik. Ukuran bit EPC yang mencapai 96-bit memungkinkan secara unik mengidentifikasi lebih dari 268 juta produsen, masing-masing memiliki lebih dari satu juta jenis produk, sementara sisanya masih mencukupi untuk melabel seluruh produk individualnya. Informasi EPC inilah yang tersimpan di dalam *chip* NFC[1].

Setiap objek yang akan diidentifikasi oleh sistem NFC memerlukan *tag* di dalamnya. *Tag* NFC didisain dan dimanufaktur menggunakan teknologi yang paling mutakhir dan bentuk geometri terkecil dari *prosessilicon* yang ada. *Tag* NFC/RFID diklasifikasikan menjadi lima kelas, yaitu:

a. CLASS I– Read Only, Factory Program-med

Jenis ini adalah jenis *tag* paling sederhana, dimana data ditulis sekali ketika dimanufaktur. Lalu memori dinonaktifkan dari segala bentuk pembaruan atau *updates*.

b. CLASS II– Write Once Read Only, Factory of User Programed

Pada jenis katagori ini, *tag* diproduksi tanpa adanya data yang tertulis di dalam memori. Data dapat ditulis oleh pamanufaktur *tag*, atau oleh pengguna untuk satu kali. Setelah itu *tag* tidak dapat lagi diprogram akan tetapi hanya dapat dibaca.

c. CLASS III –Read Write

Jenis ini merupakan jenis *tag* yang fleksibel, dimana pengguna mempunyai akses untuk menulis dan membaca data kedalam memori *tag*.

d. CLASS IV –Read Write with Onboard Sensors

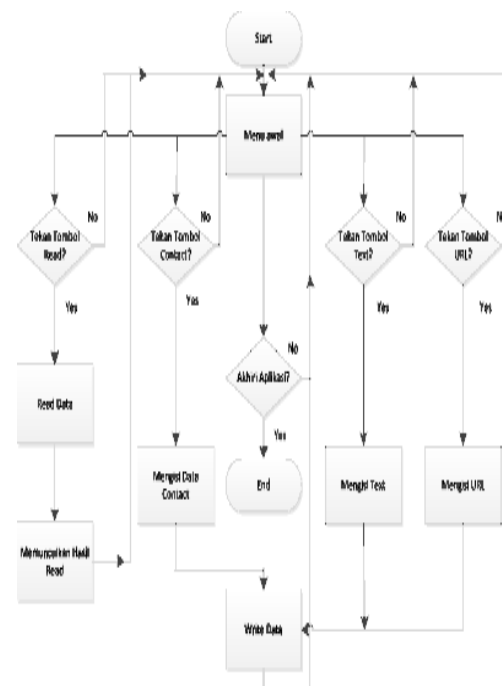
Tag jenis ini mempunyai sensor *onboard* untuk merekam parameter seperti tempe-ratur, tekanan udara dan pergerakan, yang dapat direkam dengan menuliskannya ke dalam memori *tag*. Pembacaan parameter dilakukan ketika terhubung dengan *reader*, *tag* bisa dari jenis aktif atau semi-pasif.

e. CLASS V– Read Write with integrated transmitters

Jenis tag ini seperti *miniature radio*, yang dapat berkomunikasi dengan *tag* dan *device* lain, tanpa harus adanya *reader*. Hal ini berarti mereka aktif dengan *power* dari baterai yang terpasang pada sistem. *Tag* ini bekerja saat antena mendapatkan sinyal dari NFC *reader* dan sinyal tersebut akan dipantulkan lagi, sinyal pantul ini biasanya sudah ditambahkan dengan data yang dimiliki *tag* tersebut.

Metodologi

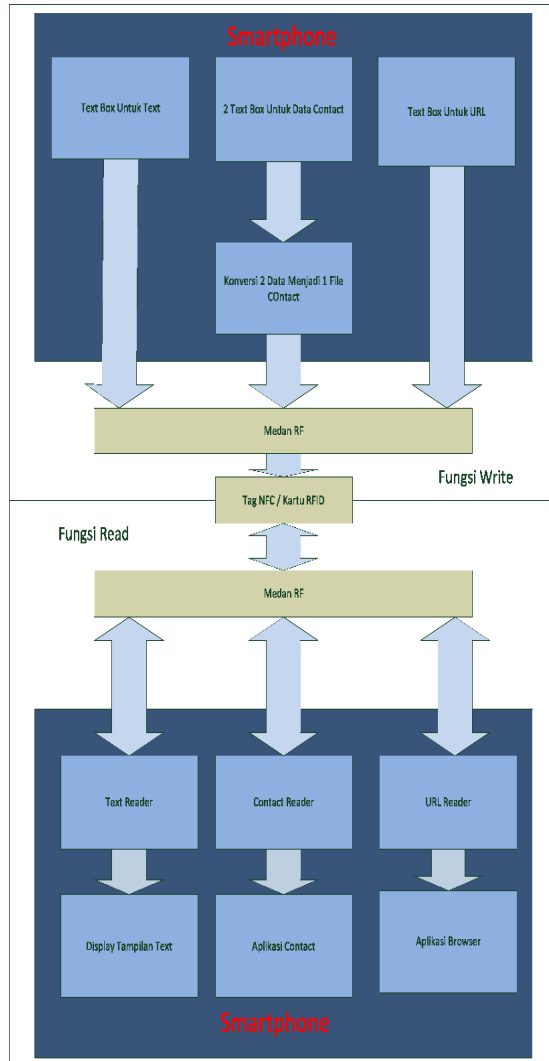
Pada awal penelitian ini, telah dirancang sebuah *flowchart* (lihat gambar 1-1) yang menggambarkan kerja sistem secara keseluruhan sebagai algoritma dasar untuk penulisan program. *Flowchart* ini menjelaskan bagaimana cara kerja program yang ditulis untuk komponen pasif NFC, dimana pada menu awal aplikasi ini terdapat 4 tombol, yaitu read, text, contact, dan URL yang menentukan/mengatur alur kerja aplikasi tersebut dari awal hingga aplikasi ini berakhir.



Gambar 1-1. Flowchart Kerja Sistem.

Selanjutnya pada sistem ini, telah diimplementasikan sistem *read* dan *write* menggunakan NFC pada *smartphone android*. Pada gambar 1-2 diilustrasikan proses *Read* dan *Write*. Implementasi sistem dibuat dalam sebuah SDK (*software development kit*) untuk android yaitu menggunakan perangkat lunak aplikasi eclipse. Dengan menambah plug-in ADT (*android tool*

development), perangkat lunak eclipse akan mampu untuk meng-*compile source code* yang dikhususkan untuk pembuatan *software android*.



Gambar 1-2. Implementasi Proses *Read* dan *Write*.

Pada perancangan dan implementasi dari sistem *read*, diperlukan fungsi-fungsi NFC yang perlu dideklarasikan pada file java agar sistem pada *layout read.xml* bisa melakukan proses *read* terhadap komponen pasif NFC yang sudah memiliki data *text*, *URL*, maupun *contact* didalamnya. Untuk permulaan, aplikasi harus mempunyai izin untuk menggunakan teknologi NFC tersebut. Untuk memenuhi izin tersebut, pada *Android Manifest.xml* perlu ditambahkan *source*

code seperti yang ditunjukkan pada gambar 1-3 (Listing Program-1) .

```
<Button
    android:id="@+id/bContact"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/bText"
    android:layout_alignBottom="@+id/bText"
    android:layout_marginRight="19dp"
    android:layout_toLeftOf="@+id/bRead"
    android:text="Contact" />
<Button
    android:id="@+id/bText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textWrite"
    android:layout_centerVertical="true"
    android:text="Text" />
<Button
    android:id="@+id/bURL"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:text="URL" />
```

Gambar 1-3. Listing Program-1

```
<Button
    android:id="@+id/bContact"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/bText"
    android:layout_alignBottom="@+id/bText"
    android:layout_marginRight="19dp"
    android:layout_toLeftOf="@+id/bRead"
    android:text="Contact" />
<Button
    android:id="@+id/bText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textWrite"
    android:layout_centerVertical="true"
    android:text="Text" />
<Button
    android:id="@+id/bURL"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:text="URL" />
```

Gambar 1-4. Listing Program-2

Dengan *source code* pada gambar 1-3, eclipse akan membuat sebuah tombol dengan nama "bRead" dengan *text* pada tombol yang bertuliskan "Read" pada *layout activity_main.xml*. Tombol *read* ini nantinya akan dihubungkan ke *layout read.xml*. Pada gambar 1-4, diperlihatkan listing program untuk pembentukan tombol: *bContact*, *bText*, dan *bURL* pada sistem.

```

<TextView
    android:id="@+id/textWrite"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textRead"
    android:layout_below="@+id/bRead"
    android:layout_marginTop="14dp"
    android:text="Write"
    android:textAppearance="?android:attr/textAppearanceLarge" />
<TextView
    android:id="@+id/textRead"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="52dp"
    android:text="Read"
    android:textAppearance="?android:attr/textAppearanceLarge" />

```

Gambar 1-5. Listing Program-3

Tombol URL nantinya akan terhubung ke *layout url.xml*, tombol *contact* akan terhubung ke *layout contact.xml*, dan tombol *text* akan terhubung ke *layout text.xml*, masing masing dihubungkan *layout activity_main.xml*. Untuk memisahkan fungsi *write* dan *read*, dipisahkan dengan *text* diatas tombol sebagai penanda fungsi seperti yang ditunjukkan pada *listing program* gambar 1-5.

Hasil dan Pembahasan

Dari hasil pengujian, dapat dikemukakan bahwa penggunaan sistem *read* pada aplikasi penelitian ini sudah bekerja dengan baik. Sistem *read* dapat membaca format *contact*, *text*, dan URL yang dihasilkan oleh *writer* aplikasi ini. Pada *writer*, sistem juga bekerja dengan sempurna, dimana *Output* dari *text* dapat ditampilkan oleh *reader*, *Output contact* dapat langsung *diimport* kedalam aplikasi *contact*, dan *output* dari URL dapat langsung mengaktifkan aplikasi *browser* yang langsung memasuki URL yang diinputkan.

Pada aplikasi penelitian ini, sistem *read* dan *write* bekerja secara baik pada kartu RFID dan *smart tag*. Adapun kekurangan pada sistem yang berjalan, terletak pada keterbatasan *hardware* NFC tag yang tidak mampu menyimpan data *contact* karena kapasitas yang terbatas. Sehingga pada penggunaan aplikasi ini sebagai alat untuk menyimpan *contact*, media promosi *web*, dan pengisi informasi untukantisipasi benda yang hilang dapat dioptimalkan pada kartu RFID. Tetapi tidak demikian untuk *smart tag*, karena tidak dapat menyimpan data *contact* dengan kapasitas yang besar.

Kesimpulan

Setelah melakukan perancangan, realisasi, dan pengujian pada sistem ini dapat disimpulkan sebagai berikut:

Aplikasi *tag writer* telah berhasil dirancang dan digunakan pada *smartphone* android dengan teknologi NFC dimana aplikasi ini dapat melakukan proses *read* dan *write file* dengan format *contact*, *text*, maupun URL dengan baik. Pengembangan penggunaan NFC pada aplikasi *smartphone* android telah berhasil dilakukan, dimana pada penelitian ini, sistem *write* dan *read* NFC digunakan sebagai sistem utamanya. Analisis dari fungsi dasar NFC dapat dilihat dari input dan output yang dihasilkan, pada inputnya, data dari textbox yang dimasukkan menghasilkan output dengan format yang sesuai dengan format yang dipergunakan saat di *write*, sehingga output data dapat di *read* dengan benar oleh aplikasi.

Daftar Pustaka

- [1]. Abu Zairi, Tommy. 2009. *Skripsi: Rancang Bangun Sistem Parkir Otomatis Berbasis Teknologi RFID dengan Fitur Sistem Debit Biaya Parkir Via SMS*. Depok: Universitas Indonesia
- [2]. <http://developer.android.com/about/index.html>
- [3]. <http://developer.android.com/tools/sdk/eclipse-adt.html>
- [4]. http://rapidnfc.com/what_is_nfc
- [5]. <http://rfidepcindonesia.net/artikel/22-nfc/89-nfc>
(Diunduh 1 May 2012)
- [6]. <http://www.eclipse.org/org/>
- [7]. <http://indonesianchipengineer.blogspot.com/2013/05/membuat-aplikasi-smart-poster.html>
(diunduh 22 Mei 2013)