BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasikan permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis dapat juga diartikan sebagai penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbarui.

Dalam proses pembuatan suatu sistem harus melakukan penelitian dan penganalisaan tentang sistem yang akan dibangun, berikut adalah beberapa analisis yang di lakukan untuk membangun aplikasi yang memanfaatkan teknologi NFC sebagai media transfer *file* dan *service initiation* pada *platform mobile* berbasis android.

3.2 Analisis Masalah

Dengan makin berkembangnya teknologi terutama dibidang IT (*Information and Technology*) maka semakin banyak pula tuntutan para pengguna untuk melakukan sesuatu dengan mudah, seperti pengiriman *file* antar ponsel, berbagi informasi, dan pengaturan perangkat elektronik seperti ponsel sangatlah dibutuhkan. Perangkat elektronik seperti ponsel untuk sekarang ini sudah banyak terpasang teknologi untuk melakukan pengiriman *file* didalam perangkat ponsel tersebut, seperti teknologi *bluetooth* dan teknologi *wifi*, namun teknologi tersebut memiliki beberapa kelemahan seperti memerlukan waktu yang sedikit lama untuk melakukan *pairing* untuk menghubungkan terhadap perangkat lainnya. Selain itu untuk melakukan pengaturan ponsel pengguna perlu membuka menu *setting* dan memilih sistem yang akan di *setting*, apabila hal seperti ini sering dilakukan dan

pengaturan sistem yang diatur merupakan sistem yang sama maka hal seperti ini akan membutuhkan waktu yang cukup banyak.

Tabel 3. 1 Perbandingan NFC, *Bluetooth* dan *WiFi [20]*

	NFC	***Bluetooth***	***WiFi***
Waktu Pairing	<0.1ms	± 6 s	± 0.5 s
Jarak	Up to 10cm	Up to 30m	± 60m tergantung standar frekuensi
Keamanan	Tingkat keamanan tinggi, sistem keamanan sudah ditentukan	Tingkat keamanan tergantung pengguna, tidak ditentukan oleh sistem	Tingkat keamanan tergantung pengguna, tidak ditentukan oleh sistem
Cara penggunaan oleh user	Sangat mudah tinggal disentuh dan didekatkan dengan perangkat lain	Memerluakan konfigurasi	Memerlukan konfigurasi

Untuk mengatasi kendala-kendala tersebut, maka dibutuhkan sebuah aplikasi yang memiliki metode *peer-to-peer* dengan teknologi nirkabel yang sama dengan teknologi *blueetooth* namun memiliki kemampuan yang lebih dari teknologi sebelumnya, seperti lebih cepatnya melakukan proses *pairing* data kesesama perangkat dengan jarak yang lebih pendek untuk melakukan konektifitas pada sesama perangkat yang berkemampuan teknlogi yang sama. Selain itu dibutuhkan juga aplikasi yang mampu melakukan *service initiation* untuk melakukan pengaturan terhadap perangkat elektronik seperti ponsel yang lebih cepat. Dengan memanfaatkan teknologi yang sedang berkembang sekarang ini yaitu teknologi NFC yang memiliki metode yang sama dengan teknologi sebelumnya yaitu *bluetooth* dan *wifi* maka aplikasi yang akan dibangun akan mengimplementasikan teknologi NFC untuk menangani kendala tersebut, selain sebagai media pengiriman *file* aplikasi ini mampu melakukan pengaturan perangkat elektronik dengan menggunakan media *tag* NFC.

Berdasarkan uraian yang dikemukakan, maka terdapat masalah yaitu bagaimana menerapkan metode *peer-to-peer* dan *service initiation* yang mudah untuk melakukan *pairing* pada aplikasi pengiriman *file* dengan memanfaatkan

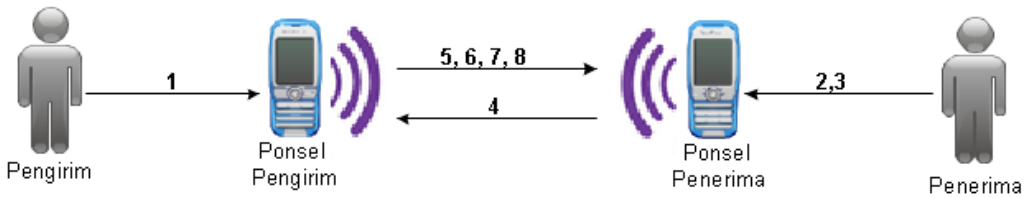
teknologi terbaru yaitu NFC (*Near Field Communication*), serta bagaimana cara untuk membuat aplikasi yang mampu untuk melakukan *read/write* pada media *tag* NFC sebagai media untuk berbagi informasi.

3.3 Analisis Prosedur yang Sedang Berjalan

Analisis prosedur yang sedang berjalan merupakan proses penguraian terhadap sistem yang telah ada atau sedang berjalan dengan tujuan untuk melakukan evaluasi yang ada, merumuskan tujuan yang ingin dicapai dengan sistem yang baru, serta menyususun rencana pembangunan sistem.

Terdapat beberapa cara/prosedur yang dapat digunakan untuk melakukan pengiriman *file* dan pengaturan sistem diantaranya penggunaan teknologi *bluetooth* dan *wifi* yang telah tertanam didalam perangkat ponsel sebagai media untuk pengiriman *file*, dan pengaturan sistem biasanya dilakukan secara manual dengan langsung mengakses perangkat tersebut.

1. Penggunaan teknologi *bluetooth* untuk saat ini masih menjadi media pengiriman *file* antar perangkat ponsel yang masih sering dipergunakan. Pada gambar 3.1 akan menjelaskan langkah-langkah untuk mengirim file menggunakan *bluetooth* pada ponsel android.



Gambar 3. 1 Pengiriman file menggunakan *Bluetooth* pada ponsel

1. Pengirim menyalakan *bluetooth* pada ponsel android dengan cara memilih menu setting lalu memilih *wireless & network* dan beri centang pada menu *bluetooth* untuk menghidupkan .
2. Penerima menyalakan *bluetooth* pada ponsel android dengan cara memilih menu setting lalu memilih *wireless & network* dan beri centang pada menu *bluetooth* untuk menghidupkan.

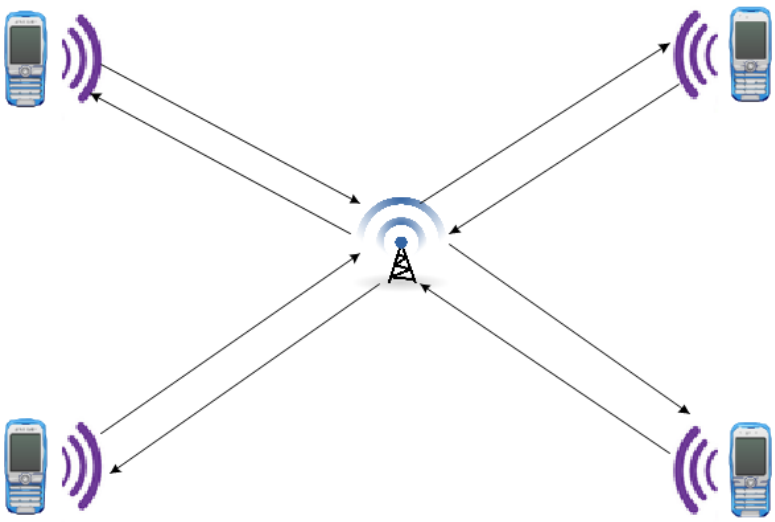
3. Kemudian setelah menyalakan *bluetooth*, pengirim mengaktifkan *discoverable* agar perangkat *bluetooth* penerima bisa terditeksi oleh *bluetooth* ponsel pengirim.
4. Perangkat *bluetooth* penerima sudah bisa terditeksi oleh perangkat *bluetooth* yang lain.
5. Apabila *bluetooth* penerima sudah bisa terditeksi oleh perangkat *bluetooth* yang lain maka, perangkat *bluetooth* pengirim akan melakukan *scaning* untuk mencari perangkat *bluetooth* yang sudah aktif.
6. Setelah *bluetooth* pengirim terditeksi maka lakukan *pairing* untuk menghubungkan kedua *bluetooth* tersebut.
7. Setelah kedua perangkat *bluetooth* tersebut terhubung maka pengirim tinggal mengirim file yang akan di kirimkan.
8. Perangkat *bluetooth* pengirim melakukan pengiriman *file* kepada perangkat *bluetooth* penerima.

Kekurangan dari sistem *Bluetooth* adalah:

a. Sistem ini menggunakan frekuensi yang sama dengan gelombang LAN standar
b. Apabila dalam suatu ruangan terlalu banyak koneksi *Bluetooth* yang digunakan, akan menyulitkan pengguna untuk menemukan penerima yang diharapkan
c. Banyak mekanisme keamanan *Bluetooth* yang harus diperhatikan untuk mencegah kegagalan pengiriman atau penerimaan informasi.
d. Sudah banyak beredar virus-virus yang disebarkan melalui *bluetooth* dari *handphone*.

2. Penggunaan teknologi *wifi* untuk proses pengiriman *file* saat ini sangat jarang sekali dipergunakan. Dalam proses penggunaannya teknologi ini

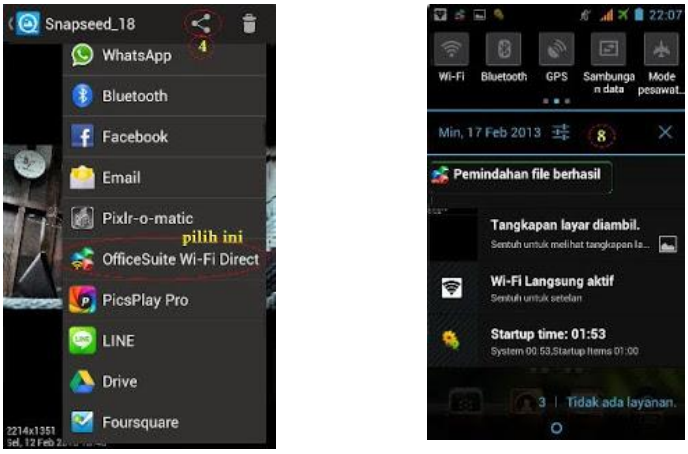
lebih cepat di bandingkan teknologi *bluetooth*, walaupun proses transfer datanya lebih cepat dibandingkan teknologi transfer yang lainnya proses pengiriman menggunakan teknologi ini sangat rentan sekali dalam sisi keamanannya. Pada gambar 3.2 akan menjelaskan prosedur pengiriman file menggunakan *wifi* antar ponsel.



Gambar 3. 2 Pengiriman file menggunakan wifi

Transfer file menggunakan *wifi* ini hanya bisa dilakukan apabila ponsel yang berbasis android yang telah tersedia fiture *wifi derect*. Jika di perangkat android ada fiture *wifi direct* tetapi pada saat ingin mengirim file tidak ada pilihan *wifi direct* maka ini harus memerlukan aplikasi tambahan seperti *officesuite*. Pada proses transfer menggunakan *wifi* semua ponsel harus terkoneksi pada satu jaringan *hotspot* yang memiliki SSID yang sama, misalkan SSID yang berada di kampus UNIKOM. Pengguna ponsel yang akan melakukan pengiriman menggunakan *wifi* maka harus terkoneksi terlebih dahulu pada SSID yang ada di UNIKOM, apabila kedua ponsel tersebut telah terhubung pada SSID yang sama maka pengiriman akan bisa melanjutkan pengiriman melalui *wifi* tersebut menggunakan aplikasi yang tersedia di google *play*. Pada gambar 3.3 menggambarkan beberapa antar muka pada aplikasi *wifi*

direct dan aplikasi tambahan seperti *officesuite* yang digunakan sebagai pengiriman *file* dengan *wifi*.



Gambar 3. 3 Antar muka aplikasi *wifi direct*

Kekurangan dari sistem Wifi adalah:

a. Sistem ini menggunakan frekuensi yang sama dengan gelombang LAN standar

b. Apabila dalam suatu ruangan terlalu banyak yang menggunakan pasilitas wifi tersebut, akan menyulitkan pengguna untuk menemukan penerima yang diharapkan

c. Banyak mekanisme keamanan wifi yang harus diperhatikan untuk mencegah kegagalan pengiriman atau penerimaan informasi.

3.4 Analisis Sistem yang Dibangun

Analisis sistem yang akan dibangun merupakan penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan sebagai perbaikan-perbaikannya.

3.4.1 Deskripsi Sistem

Sistem yang akan dibangun adalah aplikasi NFC *Tasks Initiation* berbasis *mobile* yang dapat dioperasikan menggunakan *platform* Android. Dalam pembangunannya, aplikasi ini dibangun dengan mengimplementasikan teknologi NFC (*Near Field Communication*) yang bertujuan untuk mempermudah pengguna dalam mempergunakan aplikasi ini, karena teknologi NFC merupakan seperangkat teknologi konektivitas nirkabel berbasis teknologi RFID (*Radio Frequency Identification*) yang menggunakan induksi medan magnet untuk berkomunikasi antar perangkat hanya dengan menyentuhkan perangkat tersebut. Aplikasi yang dibangun bisa digunakan sebagai *reader* maupun *writer* pada *tag* NFC tipe 1 atau tipe 2 dan bisa digunakan sebagai aplikasi untuk berbagi file antar ponsel.

Dalam penggunaan *reader*, pengguna tidak perlu menjalankan aplikasi ataupun fitur-fitur yang terdapat didalam aplikasi yang akan dibangun. Melainkan pengguna tinggal mengaktifkan fasilitas NFC yang terdapat pada ponsel yang dimilikinya, dengan demikian fasilitas *reader* yang terdapat pada aplikasi akan otomatis berfungsi. Setelah fasilitas NFC *reader* aktif maka pengguna tinggal mendekatkan perangkat ponselnya dengan perangkat NFC yang lainnya seperti *tag* NFC.



Gambar 3. 4 Gambaran proses *reader*

Keterangan :

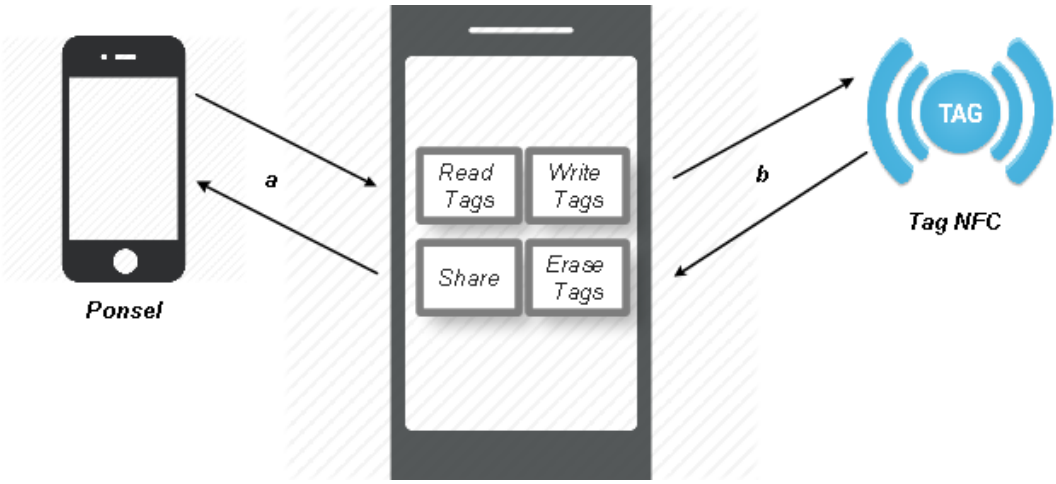
a. Pengguna mengaktifkan terlebih dahulu fasilitas NFC didalam perangkat ponselnya, setelah fasilitas NFC sudah aktif maka pengguna tinggal mendekatkan ponselnya tersebut dengan *tag* NFC.
b. *Tag* NFC akan memberikan data atau informasi yang terdapat didalamnya kepada ponsel yang memiliki pasilitas NFC dengan gelombang magnetik.

Kemudian informasi yang terdapat pada *tag* NFC akan tampil pada ponsel pengguna.

Dalam penggunaan sebagai *writer,* pengguna dapat menjalankan fitur NFC *Write Tags.* Dalam menu tersebut terdapat sub menu yang memiliki fungsi untuk melakukan *write* terhadap *tag* NFC, seperti fungsi untuk melakukkan *write URL* pada *tag* NFC, melakukan fungsi konfigurasi *wifi* yang akan di *write* pada *tag* NFC dan fungsi untuk melakukan konfigurasi pengaturan ponsel seperti *silent ringer* yang nantinya akan di *write* pada *tag* NFC.

Pada proses *peer-to-peer* pada aplikasi yang akan dibangun di implementasikan pada menu *Share* dimana pada menu tersebut memiliki beberapa fungsi yaitu untuk melakukan *sharing Text, sharing URL* dan *sharing* file bisa berupa file musik, file gambar dan file video. Pada proses *peer-to-peer* ini teknologi NFC hanya dipergunakan sebagai media untuk melakukan *pairing* terhadap teknologi yang sama agar bisa terhubung sesama perangkat yang memiliki kemampuan NFC, karena teknologi NFC ini memiliki proses *pairing* yang sangat cepat dibandingkan teknologi sebelumnya, namun untuk melakukan pengiriman file yang cukup besar melakukan menggunakan *bluetooth* karena teknologi NFC memiliki kelemahan transfer *rate* yang sangat kecil dibandingkan teknologi sebelumnya. Namaun untuk melakukan *sharing* text dan url menggunakan NFC dikarenakan *size* text dan url sangat tidak terlalu besar.

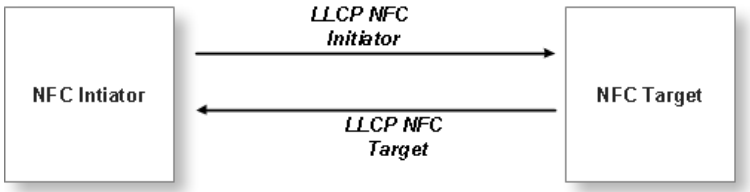
Pada menu *Erase Tags* menu ini berfungsi untuk menghapus semua perintah yang telah di inputkan pada *tag* NFC. Gambar 3.5 menunjukan alur kerja sistem yang akan dibangun.



Gambar 3. 5 Alur Kerja Sistem Yang Akan Dibangun

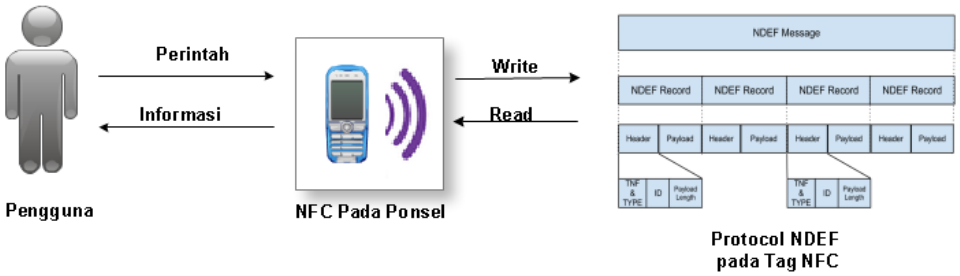
Keterangan :

a. Pada tahap ini pengguna akan melakukan *share* antar perangkat ponsel yang telah tertanam teknologi NFC. Data yang dapat di *share* pada aplikasi ini yaitu *text*, *url* dan file dimana *file* ini bisa berupa musik, gambar dan video. Pada proses ini pengguna tinggal mendekatkan kedua ponsel tersebut untuk melakukan *sharing*. Pada proses ini terjadi komunikasi antar dua ponsel yang mendukung komunikasi *peer-to-peer* yang melalui protokol LLCP pada NFC yang memungkinkan mentransfer unit informasi atau data antar dua perangkat ponsel tersebut. Proses transfer tersebut apabila *file* yang akan di *share* memiliki *size* yang cukup besar maka proses *share* ini akan dilakukan oleh *bluetooth* sedangkan teknologi NFC nya digunakan untuk menghubungkan kedua perangkat tersebut dikarenakan teknologi NFC memiliki transfer rate yang sangat kecil.



Gambar 3. 6 Peer to Peer

b. Pada tahap ini ponsel akan melakukan proses *read/write* pada perangkat NFC *tag*. Pada proses *read* pengguna hanya mendekatkan perangkat ponselnya kepada perangkat NFC *tag* sehingga terjadi proses komunikasi dengan mode pasif, dimana hanya perangkat ponsel saja yang menghasilkan medan magnet untuk berkomunikasi.

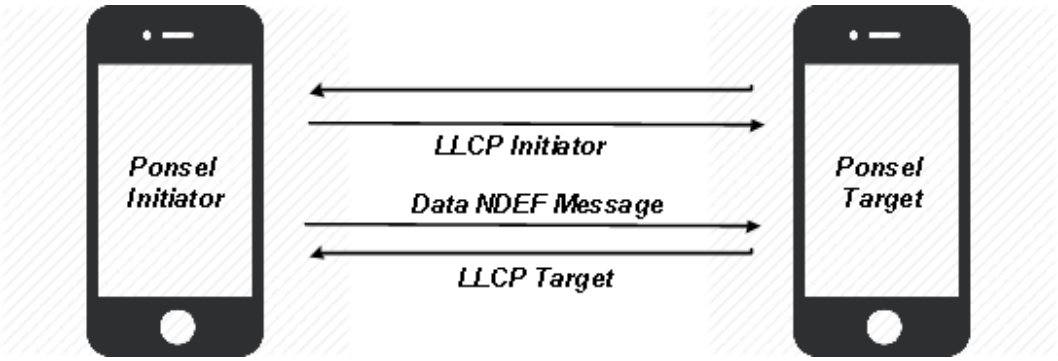


Gambar 3. 7 *Proses Read Write*

Pada proses ini medan magnet yang dikeluarkan oleh perangkat ponsel akan membaca protokol NDEF yang berada pada perangkat NFC *tag*, apabila pada protokol tersebut terdapat sebuah aksi atau informasi maka akan dibaca oleh gelombang magnet tersebut dan akan disampaikan kepada pengguna melalui ponselnya. Pada proses *write* pengguna tinggal memasukan perintah yang sudah terdapat pada aplikasi misalkan, pengguna akan menginputkan *url* unikom.ac.id pada NFC *tag* maka pengguna tinggal menulisakan *url* yang akan dibuka pada menu *url* lalu mendekatkan ponselnya pada NFC *tag*, maka akan terjadi proses *writer* antara ponsel kepada NFC *tag*. Semua inputan yang dilakukan oleh pengguna akan dikonversikan oleh sistem pada bilangn UTF (*Unicode Transformation Format*) yaitu pada bilangan UTF 8 atau sering disebut bilangan oktal dan UTF 16 atau sering disebut bilangan heksadesimal dan akan disimpan pada NDEF. Pada proses ini ponsel akan menghasilkan medan magnet yang membawa data yang telah di atur oleh pengguna dan akan disimpan pada NFC *tag* dalam protokol NDEF.

3.4.2 Analisis *Peer-to-Peer*

Dalam *mode peer-to-peer*, dua ponsel yang didalamnya terdapat perangkat teknologi NFC yang sudah di aktifkan membuat sambungan dua arah untuk pertukaran informasi, seperti digambarkan pada Gambar 3.8. Kedua perangkat tersebut dapat bertukar file, digital foto, atau data jenis lain. ISO / IEC 18092 mendefinisikan standar untuk operasi *peer-to-peer mode*.



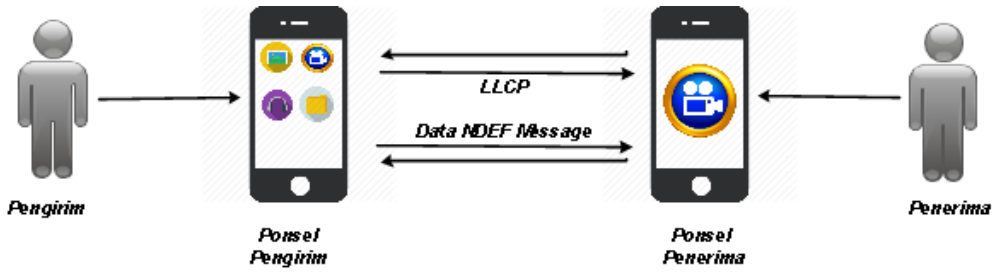
Gambar 3. 8 Analisis *Peer-to-Peer*

Layanan *peer-to-peer* diterapkan pada aplikasi ini pada fitur *share*. Layanan *peer-to-peer* digunakan untuk pertukaran data antar dua ponsel yang telah terpasang teknologi NFC. Agar kedua ponsel tersebut terhubung maka kedua ponsel tersebut harus memenuhi beberapa pengaturan didalam ponselnya seperti :

1. Android *beam* dari pengaturan OS android harus diaktifkan di kedua ponsel.
2. Layar ponsel yang akan menerima berkas tidak terkunci layarnya.
3. Ketika akan melakukan pengiriman berkas kedua ponsel saling bersentuhan.

Pada proses *peer-to-peer* ini hanya terdapat satu inputan saja dari pengguna yaitu sebuah aksi atau perintah, misalkan sebuah aksi untuk mengirimkan sebuah file video. Sedangkan *output* yang dihasilkan dari proses ini berupa bilangan heksadesimal yang telah di konversi oleh protokol NDEF dan disimpan pada NDEF *message*. Hasil konversi tersebut nantinya akan digunakaan pada tahapan berikutnya yaitu konversi dari bilangan heksadesimal ke bentuk file yang dikirimkan, proses pengkonversian tersebut terjadi pada ponsel pengirim pada

protokol NDEF yang ada pada ponsel tersebut. Proses pengiriman file berupa musik, gambar dan video akan dilakuka oleh *bluetooth* NFC hanya sebagai media untuk melakukan *pairing* kepada ponsel penerima karena teknologi NFC memiliki kelemahan dalam proses transfer dibandingkan teknologi sebelumnya, namun pada proses *share text* dan *share URL* bisa menggunakan NFC karena data tersebut tidak memiliki *size* yang terlalu besar. Misalkan pengguna akan mengirimkan file video menggunakan aplikasi ini. Cara untuk melakukan *share* tersebut bisa di lihat pada gambar 3.9.



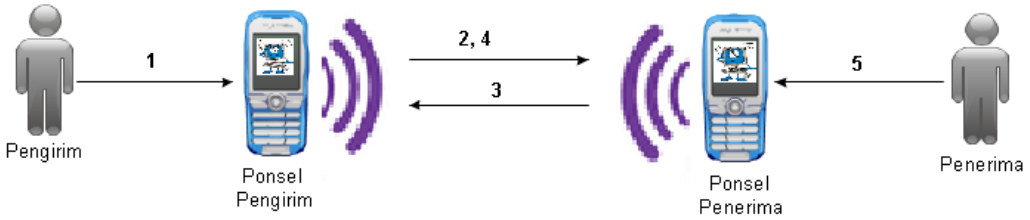
Gambar 3. 9 Skenario Pengiriman *File* Video

Keterangan gambara 3.9 :

1. Pengirim akan memilih menu *share* pada apliksi.
2. Sistem akan menampilkan menu halaman *share*.
3. Pengirim akan memilih menu share file dan sistem akan menampilkan file apa saja yang akan di share.
4. Pengirim memilih file video untuk di *share* pada ponsel penerima.
5. Sebelum melakukan *sharing* kedua ponsel pengirim dan penerima harus sudah aktif pasilitas NFC dan pasilitas *beam* yang terdapat pada ponselnya.
6. Setelah itu ponsel pengirim dan penerima saling didekatkan. Layar pada ponsel pengirim di touch untuk mendorong file yang akan dikirimkan.
7. Teknologi NFC akan melakukan pairing terhadap kedua ponsel tersebut.

8. Setelah itu *file* akan dikirimkan melalui *bluetooth*. *Bluetooth* akan menyala secara otomatis dan akan mati secara otomatis apabila *file* telah selesai di kirim.
9. File yang di *share* akan tampil pada ponsel penerima.

Langkah-langkah proses pengkonversian aksi atau perintah tersebut terdapat pada setiap ponsel yang terdapat teknologi NFC. Langkah-lagkah tersebut dapat di lihat pada gambar 3.9.



Gambar 3. 10 Alur proses *peer-to-peer*

Keterangan :

1. Pengguna melakukan perintah atau aksi pada aplikasi misalkan untuk mengirimkan sebuah file gambar. Aksi dan file gambar tersebut akan disimpan pada NDEF *message* yang berada pada ponsel pengirim, sebuah perintah atau aksi akan disimpan pada struktur NDEF *message PAYLOAD,* sedangkan file atau data akan disimpan pada NDEF *Record* yaitu berupa bilangan heksadesimal hasil konversi.
2. Pada ponsel pengirim akan mengeluarkan gelombang magnetik yang di muat oleh LLCP untuk saling berkomunikasi antar dua perangkat ponsel yang memiliki teknologi NFC.
3. Pada ponsel penerima akan mengeluarkan gelombang magnetik yang di muat oleh LLCP untuk saling berkomunikasi antar dua perangkat ponsel yang memiliki teknologi NFC. Gelombang yang dikeluarkan oleh LLCP pada ponsel penerima akan membaca NDEF *Message* yang berada di ponsel pengirim.

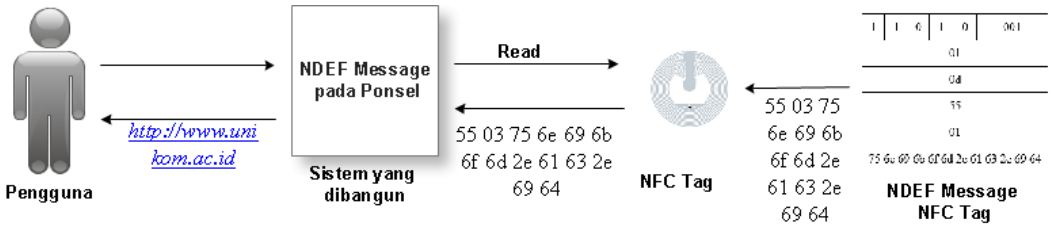
4. Apabila didalam NDEF *Message* terdapat aksi atau data maka akan dikirimkan kembali kepada ponsel pengirim melalui gelombang magnetik berupa bilangan heksadesimal hasil konverter dari NDEF pada ponsel pengirim.
5. Apabila hasil konverter dari ponsel pengirim telah diterima oleh ponsel penerima maka hasil konverter dari ponsel pengirim akan di konverterkan kembali kebentuk semula yaitu file gambar oleh NDEF yang berada pada ponsel penerima.

3.4.3 Analisis *Service Initiation*

Service Initiation merupakan kemampuan yang dimiliki oleh teknologi NFC terhadap perangkat yang memiliki kemampuan khusus penerima teknologi NFC yang bisa melakukan transformasi pesan antar keduanya dengan cara menyentuhkan atau menempelkan perangkat tersebut[16]. Dalam proses *Service Initiation* terdapat dua proses yaitu *Read* dan *Write*. Proses Write merupakan proses dimana perangkat teknologi NFC seperti *Tag* NFC atau *smart* NFC akan di *write* atau di inputkan sebuah informasi seperti meng*input*kan sebuah url kedalam *tag* NFC. Sedangkan pada proses *Read,* teknologi NFC akan me*read* atau membaca isi informasi yang telah di *write* atau di inputkan kedalam teknologi NFC lainnya, seperti perangkat ponsel yang didekatkan dengan *tag* NFC maka ponsel tersebut akan membaca isi informasi yang terdapat dalam *tag* tersebut. Berikut adalah analisis *Read* dan Write pada aplikasi yang akan dibangun.

3.4.3.1 Analisis *Read*

Layanan *read* pada aplikasi ini tidak memiliki fitur khusus karena layanan ini akan berjalan secara otomatis apabila aplikasi ini dijalankan. Pada proses *read* hanya terdapat satu inputan yaitu aksi atau informasi yang terbaca pada *tag* NFC oleh aplikasi, ketika aplikasi didalam ponsel menemukan *tag* NFC, maka disitulah dimulai proses pembacaan pada *tag* NFC.



Gambar 3. 11 Proses *Read*

Dalam proses *read* akan menjelaskan cara untuk membaca data NDEF dari *tag*, dalam proses tersebut memungkinkan operasi yang berbeda-beda berdasarkan *tag* yang ditemukan. Pada gambar 3.12 adalah operasi pada proses *read* dari *tag* yang ditampilkan atau di temukan pada NDEF PAY LOAD.

```
Procedure aksiIntent (Intent intent)

Kamus :

   Aksi : string

   Intent,i :integer

Algoritma

   Aksi : intent.getAksi()

   if (NfcAdapter.ACTION_TAG_DISCOVERED.equals(action)
      NfcAdapter.ACTION_TECH_DISCOVERED.equals(action)
      NfcAdapter.ACTION_NDEF_DISCOVERED.equals(action)) {
      Parcelable[] rawMsgs =
intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
      NdefMessage[] msgs;
      if (rawMsgs != null) {
          msgs = new NdefMessage[rawMsgs.length]
          for ( i = 0; i < rawMsgs.length i++)
              msgs[i] = (NdefMessage) rawMsgs[i]
      endIf
   elseIf {
                byte[] empty = new byte[0];
                byte[] id =
intent.getByteArrayExtra(NfcAdapter.EXTRA_ID);
                Parcelable tag =
intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
                byte[] payload = dumpTagData(tag).getBytes();
                NdefRecord record = new
NdefRecord(NdefRecord.TNF_UNKNOWN, empty, id, payload);
                NdefMessage msg = new NdefMessage(new
NdefRecord[])
```

```
                msgs = new NdefMessage[]
    endIf

endProdecure
```

Gambar 3. 12 NDEF PAYLOAD

Pertama aplikasi akan memeriksa maksud dari interaksi antara aplikasi dan *tag* melalui NDEF_DISCOVERED. Kemudian, *tag* NFC terditeksi oleh aplikasi dan aplikasi memperoleh informasi yang terdapat didalam *tag* tersebut. Setelah aplikasi mendapatkan informasi dari NDEF *message tag* NFC, maka aplikasi akan memproses apapun yang terdapat pada *tag* NFC tersebut.

Untuk memperoleh informsai dari NDEF *message pada tag* NFC aplikasi menggunakan *lib* yang terdapat pada *eclipse* yaitu *NdefMessage* untuk menyimpan pesan dan memanggil *getNdefMessage* lib *get* ini digunakan untuk mengambil informasi dari tag. Pada Gambar 3.12 adalah code untuk pemanggilan dan penyimpanan pada *tag.*

```
Procedure TagViews(NdefMessage[])
   Kamus
      Msgs, size, i = integer
     mTagContent = content
   Algoritma
        If
          (msgs == null || msgs.length == 0)
        endIf
        LayoutInflater inflater = LayoutInflater.from(this)
        Date now = new Date();
        List<ParsedNdefRecord> records =
NdefMessageParser.parse(msgs[0])
        for ( i = 0; i < size; i++)
            timeView.setText(TIME_FORMAT.format(now))
            content.addView(timeView, 0)
            ParsedNdefRecord record = records.get(i)
            content.addView(record.getView(this, inflater,
content, i), 1 + i)
            content.addView(inflater.inflate(false), 2 + i)
endProcedure
```

Gambar 3. 13 NDEF Message

Dalam lib metode *getNdefMessage* aplikasi akan mengambil data yang dimaksud pada *tag* NFC dengan menggunakan metode *intent.getParcelableArrayExtra*. metode ini akan mendapatkan data yang terdapat pada *tag* yang dimaksud, setelah itu aplikasi akan mengecek *array rawMessages* apakah *null* atau kosong apabila array tersebut kosong maka *tag* NFC tersebut beris informasi pada NDEF Message. Apabila *rawMessage null* berarti *tag* tersebut dari jenis yang tidak diketahui atau *tag* tersebut masih kosong sehingga aplikasi akan kembali lagi pada halaman pertama.

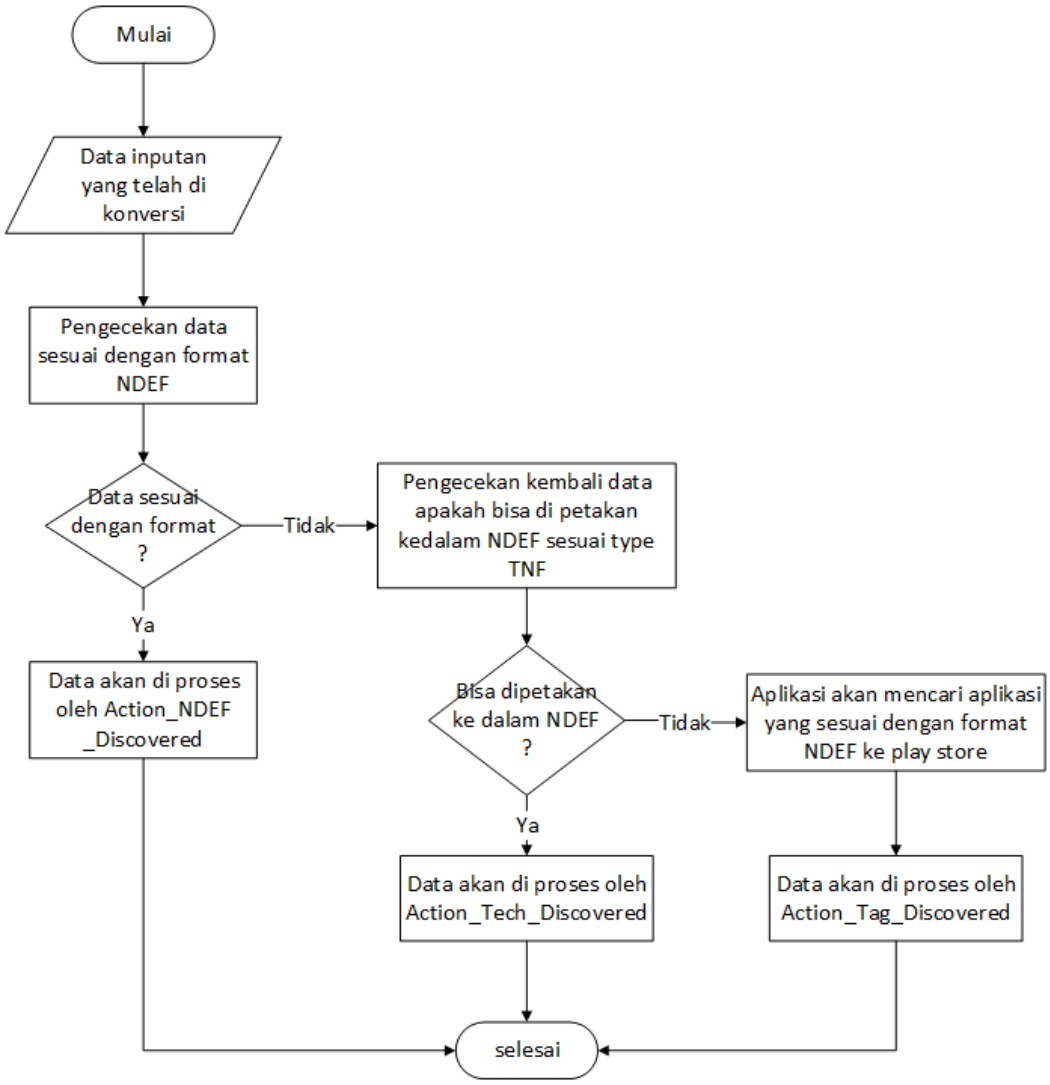
Pada mode operasi read/write, ponsel NFC dengan status aktif menginisiasi komunikasi wireless, setelah itu data dibaca dan diubah kemudian di simpan pada NFC tag. Pada mode ini, ponsel NFC berhak membaca tipe label yang diperintahkan oleh NFC. NFC melakukan standarisasi terhadap beberapa data yang diperintahkan untuk dibaca berdasarkan NFC *Data Exchange Format* (NDEF).

Membaca data NDEF dari tag NFC ditangani oleh *tag dispatch system*, yang menganalisis *tag* NFC saat ditemukan, mengkategorikan data secara tepat, dan memulai aplikasi yang tertarik pada data yang dikategorikan. Sebuah aplikasi yang ingin menangani NFC *tag* dapat mendeklarasikan sebuah *intent filter* dan permintaan untuk menangani data. Sebuah intent adalah sekumpulan informasi yang menjelaskan tindakan yang akan dilakukan oleh aplikasi, termasuk data yang akan diproses, kategori komponen yang harus melakukan tindakan tersebut, dan instruksi lainnya yang berhubungan dengan aplikasi tersebut.

Ketika tag *dispatch system* selesai membuat *intent* yang mneggunakan tag NFC dan informasi didalamnya, sistem tersebut akan mengirimkan intent untuk aplikasi yang mendukung intent tersebut. Jika lebih dari satu aplikasi dapat menangani intent, *Activity Chooser* ditampilkan sehingga aplikasi dapat memilih sistem activity yang ingin dijalankan. Berdasarkan prioritasnya, *tag dispatch system* mendefinisikan *tag* yang ditemukan menjadi tiga bagian, yaitu:

1. *ACTION_NDEF_DISCOVERED*: Ketika *Tag* NFC didekatkan dengan ponsel yang telah dipasang aplikasi maka *intent* ini digunakan untuk memulai kegiatan ketika tag yang berisi data NDEF ditemukan dan merupakan proses pertama yang akan membaca isi dari NDEF tersebut. Ini adalah intent dengan prioritas tertinggi, dan jika memungkinkan, *tag dispatch system* akan mencoba untuk memulai *activity* dengan intent ini terlebih dahulu sebelum *intent* yang lainnya.
2. *ACTION_TECH_DISCOVERED*: Jika tidak ada activity yang terdaftar untuk menangani *intent ACTION_NDEF_DISCOVERED*, *tag dispatch system* mencoba untuk memulai aplikasi dengan intent ini. Intent ini juga langsung dimulai (tanpa memulai *ACTION_NDEF_DISCOVERED* terlebih dahulu) jika tag yang ditemukan berisi data NDEF yang tidak dapat dipetakan ke tipe MIME (*Multipurpose Internet Mail Extension*) yaitu tipe yang digunakan sebagai mekanisme untuk mengirim berbagai informasi seperti text, aplikasi, gambar, suara, video, dll agar tidak salah menerjemahkan konten yang salah, jika tag tidak berisi data NDEF dari teknologi *tag* yang dikenal. Maka sistem akan mulai menciptakan sesuatu yang sesuai dengan isi data NDEF tersebut, misalkan bisa membuka *play store* untuk mengunduh aplikasi yang dimaksud dalam NDEF.
3. ACTION_TAG_DISCOVERED: intent ini dimulai jika tidak ada activity yang menangani ACTION_NDEF_DISCOVERED atau intent ACTION_TECH_DISCOVERED.

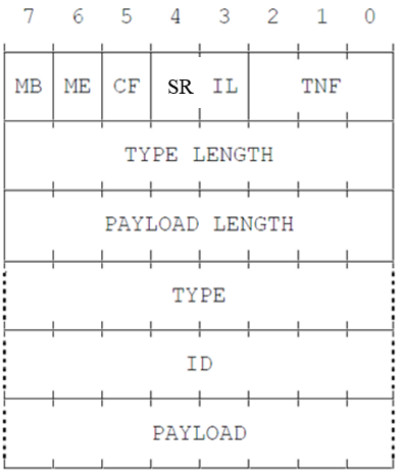
Pada gambar 3.13 akan menjelaskan skema *read* pada *Tag* NFC



Gambar 3. 14 Distribusi Intent pada *Tag Dispatch System*

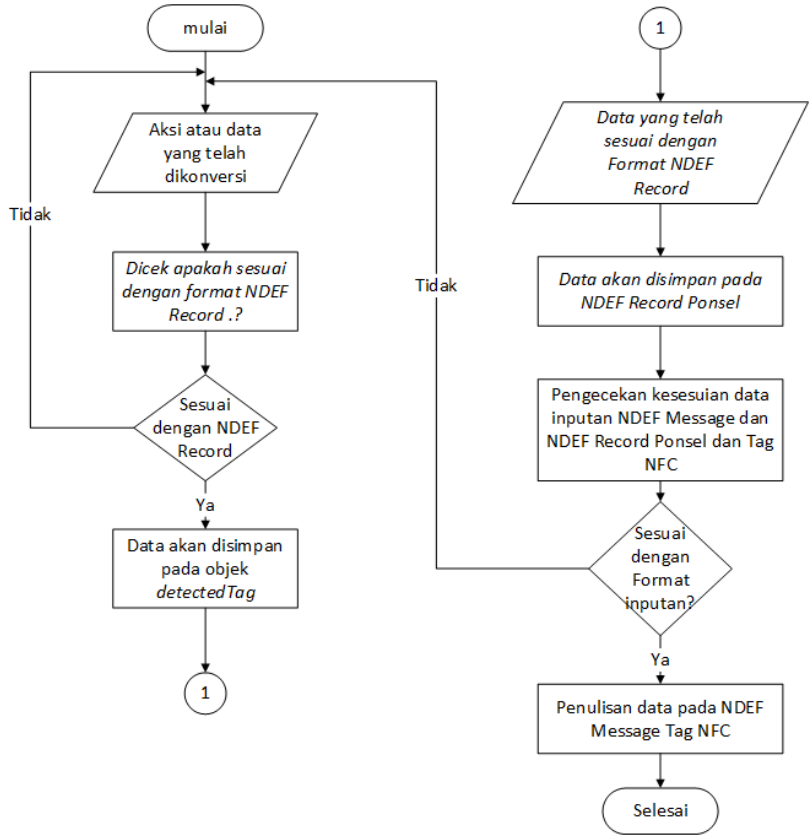
3.4.3.2 Analisis *Write*

Pada proses *write* sistem akan melakukan proses ketika perangkat ponsel yang sudah dipasangkan aplikasi didekatkan dengan *Tag* NFC, maka sistem akan memulai aksi yang telah di inputkan oleh pengguna. Inputan tersebut akan di konversi oleh sistem kedalam bentuk heksadesimal. Hasil dari inputan tersebut akan disimpan pada NDEF *Message* tepatnya pada bagian NDEF *Record.* Hasil tersebut akan disaimpan sesuai dengan format yang telah ditetapkan oleh NDEF *Record.* Gambar 3.14 merupakan gambar format dari NDEF *Record*.



Gambar 3. 15 Format NDEF *Record*

Pada gambar 3.15 menjelaskan proses input pada *Tag* NFC.



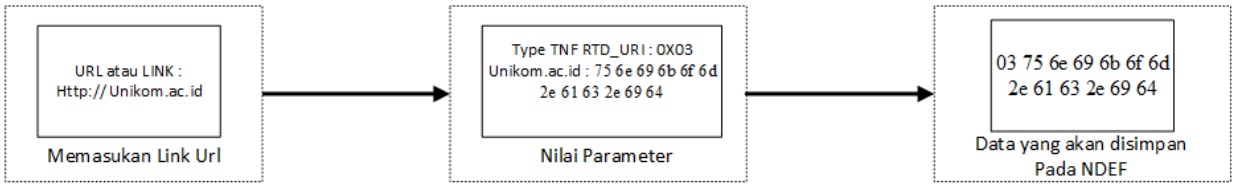
Gambar 3. 16 Proses Input

Pertama tujuan dari aksi yang akan diinputkan dicek jika diproses dari hasil interaksi NFC dari NDEF_DISCOVERED. *Tag* menampilkan kondisi *Tag* NFC pada saat ditemukan dan mengecek apakah data dari hasil konversi sesuai dengan format NDEF. Selanjutnya *Tag* yang ditemukan diterima dan disimpan pada objek `detectedTag`. Selain itu `NfcAdapter.EXTRA_TAG` digunakan sebagai parameter untuk mendapatkan informasi *Tag*. Setelah mendapatkan kondisi dari *Tag* yang terdeteksi, pertama-tama NDEF *Message* perlu disiapkan dan kemudian ditulis pada NDEF *Record* ponsel. Setelah itu menginputkan NDEF *Message* ke *Tag*, NDEF *Record* dan NDEF *Message* ponsel dan *tag* harus dipersiapkan berdasarkan jenis data inputan sesuai berdasarkan standar NFC Forum. Setelah dipersiapkan, data akan tertulis pada *Tag*.

Untuk melakuakan inputan pada *tag* NFC, setiap *value* yang akan di inputkan pada NDEF harus sesuai dengan data NDEF *Record* dan NDEF *Message* yang telah di tetapkan oleh NFC. Setiap inputan akan di cek sesuai dengan TNF (*Type Name Format*) yang telah ditetapkan, setiap *type* akan memiliki *value* yang berbeda-beda. Berikut adalah proses pengecekan *type* data sesuai dengan TNF.

a. TNF_WELL_KNOWN dengan RTD_URI

TNF_WELL_KNOWN adalah salah satu type TNF yang memiliki value 0x01. Sedangkan RTD_URI adalah record yang terdapat pada poster NFC yang memiliki kemampuan untuk menyimpan text, audio dan url. Untuk membuat TNF_WELL_KNOW *Record* dengan RTD_URI maka diperlukan sebuah URL dengan prefiks (awalan). Contoh prefiks bisa berupa "http://www.", "http://", atau "https://". Setiap prefiks memiliki kode byte yang berbeda-beda, misalnya "http://www." bernilai 1, "http://" bernilai 3 dan tidak ada prefiks bernilai 0. Untuk setiap data inputan, awalan harus diinputkan sebelum data URI. Pada gambar 3.16 menjelaskan proses TNF_WELL_KNOW *Record* dengan RTD_URI dibuat dengan prefiks 0x01 atau 1.



Gambar 3. 17 Proses TNF RTD URI

Berikut adalah penjelasan dari gambar 3.16:

a. Mula-mula URI dibuat dan disimpan ke dalam array byte `uriField` sesuai dengan pengkodean dengan karakter US-ASCII.

b. Array byte muatan data digunakan untuk menyimpan seluruh muatan data yang dibuat dengan ukuran URI dan tambahan byte untuk menyimpan

prefiks. Karena prefiks dari contoh adalah "http://www." maka muatan data diset ke 0x01

c. Kemudian data URI ditambahkan ke array *payload* menggunakan *method* `System.arraycopy`. *Method* ini hanya salinan dari elemen `uriField.length` dari array `uriField` mulai dari 0 menuju array muatan data mulai dari 1.

d. Terakhir membuat NDEF *Message* dengan membuat NDEF *Record* dengan RTD_URI. Untuk melakukan hal tersebut diperlukan membuat `NdefRecord`. `NdefRecord` yang tidak dibangun dari byte mentah harus memiliki persyaratan berikut :

1. 3 bit *Type Name Format* yang mengidentifikasi bagaimana menjelaskan tipe field
2. *Variable Length Type* yang menjelaskan format *record*.
3. *Variable Length ID* yang mengidentifikasi *record.*
4. *Variable Length Payload* yang merupakan data actual.

b. TNF_WELL_KNOWN dengan RTD_TEXT

TNF_WELL_KNOWN adalah tipe TNF_WELL_KNOWN nama pada tipe ini record yang disimpan pada poster ataupun tag hanya berupa text saja. Pada RTD_TEXT *record* teks dapat dikodekan atau di konversikan pada UTF-8 atau UTF-16 yang terdefinisi dalam status byte di teks *record*. Selain itu status byte juga termasuk ke dalam panjang dari bahasa pengkodean yang digunakan untuk mengidentifikasi kode bahasa teks. UTF-8 teridentifikasi oleh byte 0x00 sedangkan UTF-16 teridentifikasi oleh byte 0x80. Pengkodean atau konversi untuk menentukan UTF-8 atau UTF-16 di lakukan sesuai dengan kebutuhan atau panjang teks yang diinputkan.

c. TNF_ABSOLUTE_URI

TNF_ABSOLUTE_URI menunjukan bentuk mutlak dari URI dari aturan yang didefiniskan oleh RFC 3986 Array byte URI mengandung URI mutlak yang dibuat oleh pengkodean dengan set karakter US-ASCII dan kemudian digunakan untuk membuat TNF_ABSOLUTE_URI *record* dengan memilih jenis rekor sebagai NdefRecord.TNF_ABSOLUTE_URI.

d. TNF_MIME_MEDIA

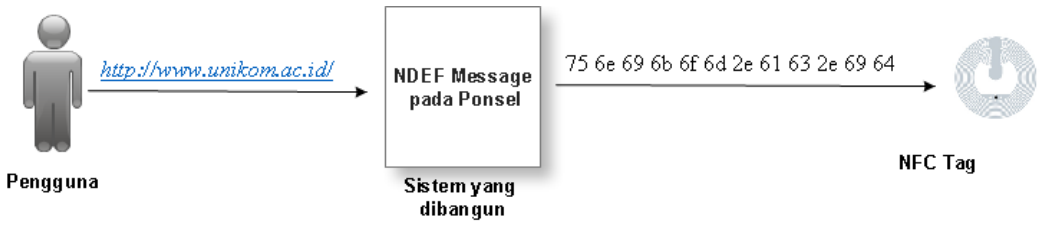
Record memiliki tipe MIME `application/nfclabapp` yang disimpan dalam string `mimeType`. Untuk membuat `NdefRecord`, TNF membuat *record* yang didefinisikan sebagai `NdefRecord.TNF_MIME_MEDIA` dan *Variable Length Type* dari NDEF *record* yang didefinisikan sebagai byte dari tipe MIME. *Record* diberi nilai 0 karena akan menjadi *record* pertama dalam `NdefMessage` dan *payload* yang akan ditulis ke *Tag* yang diambil dari *payload* string.

e. TNF_EXTERNAL_TYPE

TNF_EXTERNAL_TYPE adalah tipe TNF yang menunjukan bidang tipe TNF berisi nilai yang mengikuti TNF RTD yang didefinisikan untuk nama jenis eksternal. Layanan terdaftar untuk *record* diberikan dari string `externalType` sedangkan *payload* terdaftar diberikan dari string *payload*. TNF membuat *record* didefinisikan sebagai TNF_EXTERNAL_TYPE dan *Variable Length Type* yang didefinisikan sebagai byte dari layanan daftar.

3.4.3.2.1 Analisis *Write Url*

Layanan *Write Url* diterapkan pada layanan aplikasi ini pada fitur *Write Tags* dimana pengguna akan menginputkan halaman *Url* pada aplikasi ini dan akan didekatkan pada *tag* NFC sehingga terjadi proses *writer* antara ponsel pengguna yang telah di inputkan *text url* atau *link* terhadap *tag* NFC.



Gambar 3. 18 *Write Url*

Input pada proses ini merupakan text *url* atau *link* dan *output* yang dihasilkan pada proses ini berupa bilangan heksadesimal yang nantinya akan disimpan pada NFC *Data Exchange Format* (NDEF). Dalam pemrosesannya untuk dapat mengkonversi dari inputan *text* tersebut terdapat beberapa tahapan, diantaranya :

1. Inputan dari pengguna pada sistem yaitu berupa *text url* atau *link*, misalkan pengguna meng*input*akn *url* http://www.unikom.ac.id/. Maka sistem akan melakukan pengecekan terhadap inputan tersebut. Pertama sistem akan mengecek URI (*Uniform Resource Identifier*) *Identifier Code* dari inputan url tersebut sesuai pada tabel URI yang telah di tetapkan pada NFC untuk mendapatkan *value* dari URI tersebut. Tabel 3.2 merupakan tabel URI yang telah di tetapkan pada NFC.

Tabel 3. 2 Format Tabel URI

Value	**Protocol**
0x00	URI *Field*
0x01	*http://www.*
0x02	*https://www.*
0x03	*http://*
0x04	*https://*
0x05	*Tel:*
0x06	*mailto:*
0x07	*ftp://anonymous:anonymous@*
0x08	*ftp://ftp.*

0x09	*ftps://*
0x0A	*sftp://*
0x0B	*smb://*
0x0C	*nfs://*
0x0D	*ftp://*
0x0E	*dav://*
0x0F	*news:*
0x10	*telnet://*
0x11	*imap:*
0x12	*rtsp://*
0x13	*urn:*
0x14	*pop:*
0x15	*sip:*
0x16	*sips:*
0x17	*tftp:*
0x18	*btspp://*
0x19	*btl2cap://*
0x1A	*btgoep://*
0x1B	*tcpobex://*
0x1C	*irdaobex://*
0x1D	*file://*
0x1E	*urn:epc:id:*
0x1F	*urn:epc:tag:*
0x20	*urn:epc:pat:*
0x21	*urn:epc:raw:*
0x22	*urn:epc:*
0x23	*urn:nfc:*

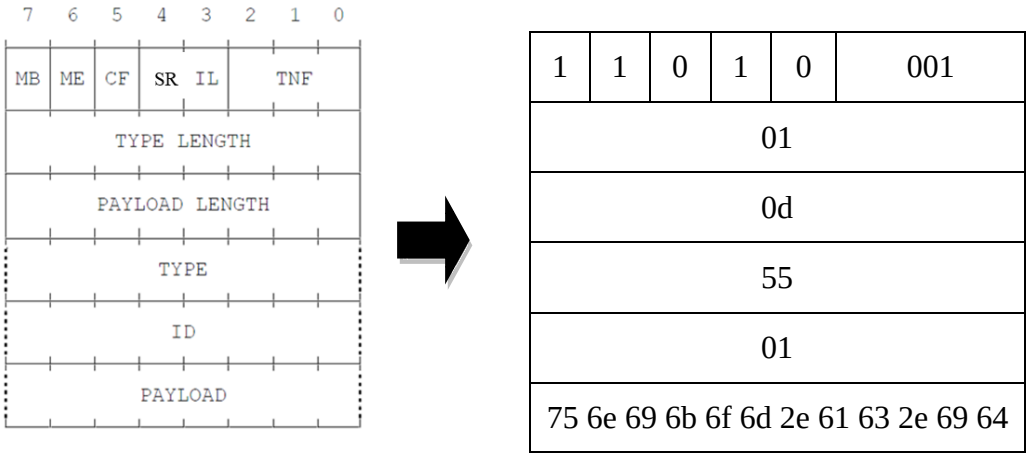
Sesuai dengan tabel URI yang telah ditetapkan maka url dari *http://www.unikom.ac.id* adalah *http://www*. maka *value* URI dari url tersebut adalah 0x01 atau 01.

2. Setelah sistem melakukan pengecekan terhadap URI pada inputan, selanjutnya sistem akan melakukan konversi pada sisa *text url* kedalam bentuk heksadesimal. Misalkan url yang telah di inputkan adalah *http://www.unikom.ac.id* sesuai dari inputan tersebut maka sisa dari URI adalah *unikom.ac.id* maka sisa *text* tersebut akan di konversikan kedalam bentuk heksadesimal. Maka hasil dari konversi *unikom.ac.id* adalah 75 6e 69 6b 6f 6d 2e 61 63 2e 69 64.
3. Setelah menentukan *value* URI dan melakukan konversi kedalam bentuk heksadesimal kemudian hasilnya akan disimpan pada format NDEF *Record.*
4. Setelah *text* inputan tersebut diketahui *value* dari URI nya dan diketahui nilai konversinya maka sistem akan melakukan penempatan hasil tersebut pada *variabel* sesuai dengan format NDEF *Record* pada gambar 3.14. Maka *value* akhir dari *http://www.unikom.ac.id* adalah sebagai berikut :
   a. *Value* URI dari *http://www*. adalah 01
   b. *Value* dari hasil konveris text *unikom.ac.id* kedalam bentuk heksadesimal adalah 75 6e 69 6b 6f 6d 2e 61 63 2e 69 64

   Setelah itu *value* tersebut akan di simpan pada *variabel-variabel* yang telah ditetapkan pada format NDEF *Record*. Berikut adalah isi dari *variabel* pada format NDEF *Record* dengan inputan *http://www.unikom.ac.id :*

   1. *Variabel* MB (*Message Begin*) : *variabel* ini akan selalu berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC. Sebagai awal NDEF *Message.*
   2. *Variabel* ME (*Message End*) : *variabel* ini akan selasu berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC. Sebagai akhir NDEF *Message*.

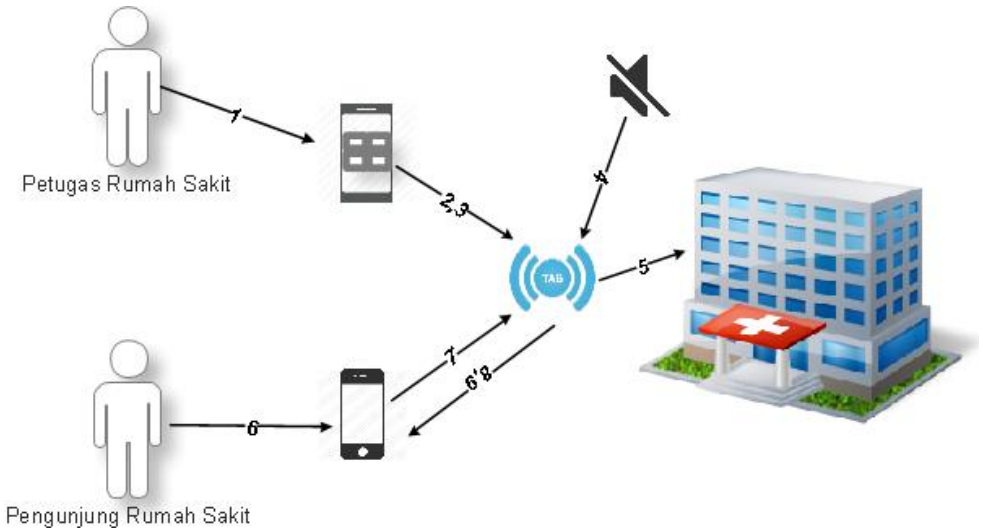
3. *Variabel* CF (*Chunk Flag*) : *variabel* ini berisi *value* 0 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC untuk mengidentifikasi potongan record.
4. *Variabel* SR (*Short Record*) : *variabel* ini berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC untuk menentukan *Payload* yang berukuran kecil.
5. *Variabel* IL (*Identification Length*) : *variabel* ini berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC. Sebagai awal NDEF *Message*.
6. *Variabel* TNF (*Type Name Format*) : variabel ini akan di isi oleh *value* bit yaitu 001, value tersebut merupakan standar dari *value* TNF pada NDEF *Record*.
7. *Type Length :* berisi *value* 01 nilai 01 berasal dari value URI *http://www*.
8. *Payload Length :* berisi bilangan heksadesimal 0d nilai 0d merupakan nilai dari panjang oktet dari *Payload* dan SR.
9. *Type :* berisi *value* 55, nilai tersebut didapat dari Type TNF yaitu URI.
10. *ID :* 01 merupakan nilai dari ID_LENGHT NFC *Record*
11. *Payload :* 55 03 75 6e 69 6b 6f 6d 2e 61 63 2e 69 64 merupakan bilangan heksadesimal dari *unikom.ac.id*



Gambar 3. 19 *Variabel* NDEF *Record* *http://www.unikom.ac.id*

3.4.3.2.2 Analisis *Write* Pengaturan Sistem

Pada layanan *Other Action* diterapkan pada layanan aplikasi ini pada fitur *Write Tags* dimana isi dari layanan ini adalah *Enable/Disable Airplane Mode, Enable/Disable WiFi, Enable/Disable Bluetooth, Narmal/Silent Ringer, Enable Alarm,* dan *Enable Launching App.* Didalam layanan ini pengguna akan memberikan aksi perintah untuk mengaktifkan atau menonaktifkan layanan-layanan tersebut. Berikut adalah contoh kasus penerapan analisis Write Pengaturan Sistem yang diterapkan pada sebuah *Rumah Sakit*. Misalkan seorang pengunjung rumah sakit akan melakukan *Silent* untuk ponselnya.



Gambar 3. 20 Penerapan *Write* Untuk *Silent*

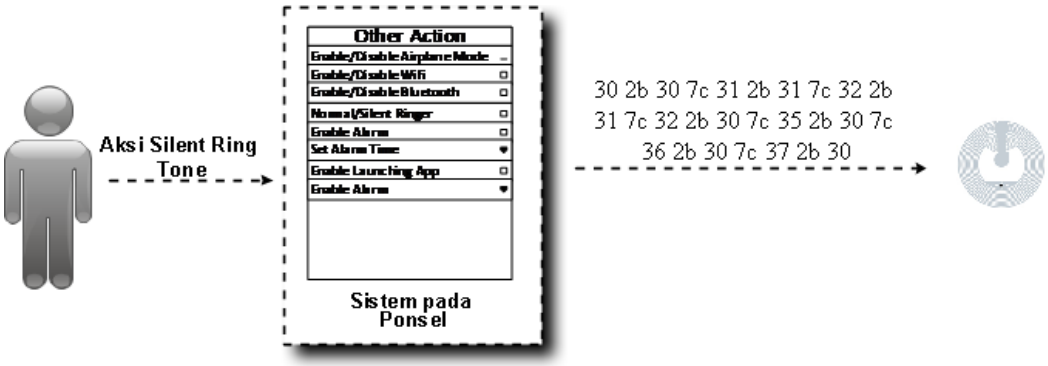
Sekenario Gambar :

1. Petugas Rumah Sakit akan melakuan inputan sebuat pengaturan sistem pada ponsel yaitu untuk meng*silent*kan ponsel setiap pengunjung yang akan mengunjungi ruangan yang memerlukan ketenangan khusus seperti ruangan Icu
2. Sistem pada ponsel yang telah diinstal aplikasi akan menampilkan halaman untuk melakukan sebuah pengaturan sistem *silent*.
3. Sistem akan menyimpan aksi yang telah di inputkan oleh petugas Rumah Sakit tersebut ke dalam *tag* NFC.
4. Pengaturan sisitem yang telah diinputkan oleh petugas Rumah Sakit akan disimpan pada *tag* NFC dalam bentuk bilangan sesuai dengan format yang telah ditetapkan oleh NFC.
5. *Tag* NFC akan disimpan di Rumah Sakit di tempat-tempat yang khusus yang memerlukan ketenangan untuk pasien.
6. Setiap pengunjung yang akan memasuki ruangan tersebut harus meng*silent*kan ponsel yang dimilikinya, yaitu dengan cara tinggal mengaktifkan teknologi NFC yang dimiliki oleh ponselnya lalu tinggal

didekatkan pada NFC yang telah diisi pengaturan oleh petugas Rumah Sakit nya.

7. Teknologi NFC yang terpasang pada ponsel pengunjung akan melakukan pairing terhadap *tag* NFC yang telah terpasang didalam runagn tersebut dengan cara didekatkan ponselnya pada *tag* NFC tersebut.
8. Tag NFC akan melakuakan pengiriman inforamsi melalu medan magnet yang telah terhubung dengan ponsel pengunjung tadi, informasi yang dikirimkan yaitu berupa bilangan okta atau bilang heksadesimal.
9. Kemudian ponsel pengguna akan melakukan konversi informasi tersebut kedalam bentuk semula yaitu berupa informasi untuk melakukan pengaturan *silent* ada ponselnya.

Berikut adalah bagai mana cara sistem akan melakukan *Write* sebuah aksi untuk pengaturan sistem yang ada pada ponsel.



Gambar 3. 21 Proses *Write Action*

Input pada proses ini merupakan aksi perintah berupa *check box* untuk menandakan sebuah aksinya, dan yang dihasilkan pada proses ini berupa bilangan heksadesimal yang nantinya akan disimpan pada NFC *Data Exchange Format* (NDEF). Dalam pemrosesannya untuk dapat mengkonversi dari inputan *text* tersebut terdapat beberapa tahapan, diantaranya :

1. Inputan dari pengguna pada sistem yaitu berupa aksi, sesuai contoh kasus diatas pengguna meng*input*akn aksi untuk *Silent* . Maka sistem

akan melakukan pengecekan terhadap inputan tersebut. Pertama sistem akan mengecek tipe TNF nya terlebih dahulu, pengecekan tipe TNF ini disesuaikan dengan tabel 2.2 *Value Type Name Format* yang telah di tetapkan oleh NFC. Sesuai dengan contoh kasus maka inputan dari pengguna merupakan untuk *Silent*, tipe TNF nya adalah MIME Media *type* maka value dari tipe TNF ini adalah 0x02 atau 02.

2. Setelah sistem melakukan pengecekan terhadap tipe TNF aksi yang telah diinputkan, selanjutnya sistem akan melakukan pengecekan aksi tersebut sesuai dengan tabel 3.3 untuk mengetahui aksi apa yang telah di inputkan. Kode-kode untuk mengaktifkan atau untuk menonaktifkan aksi-aksi yang terdapat pada aplikasi ini dapat di lihat pada tabel 3.3.

Tabel 3. 3 value aksi

No	Nama Aksi	*Value*
1	ENABLE_TOGGLE_WIFI	0
2	ENABLE_TOGGLE_BLUETOOTH	1
3	ENABLE_TOGGLE_RINGTONE	2
4	ENABLE_TOGGLE_AIRPLANE_MODE	3
5	ENABLE_SET_ALARM	4
6	ALARM_TIME	5
7	ENABLE_START_EXTERNAL_APP	6
8	EXTERNAL_APP_PACKAGE_NAME	7

Format yang akan dihasilkan dari pengecekan dari aksi tersebut adalah 0+0|1+0|2+0|3+0|4+0|5+0|6+0|7+0. Nilai 0,1,2,3,4,5,6 dan 7 merupakan *value* dari tabel 3.3 yang telah di tetapkan pada sistem untuk membedakan aksi yang diinputkan oleh pengguna. Sedangkan nilai 0 adalah kode untuk memberikan aksi *enable* atau *disable* pada sistem, apabila aksi itu *disable* maka *value*nya adalah 0 dan *enable* memiliki *value* 1. Tanda + dan tanada | merupakan VALUE_SEPARATOR dan

PREFERENCE_SEPARATOR yang berguna sebagai pemisah dari aksi tersebut.

Aksi yang telah diinputkan adalah *Silent* sesuai dari inputan tersebut maka akan di rubah terlebih dahulu kepada bilangan yang telah ditentukan oleh sistem. Bilangan yang didapatkan dari aksi *Silent* tersebut adalah 0+0|1+0|2+1|3+0|4+0|5+0|6+0|7+0.

3. Setelah menentukan bilangan aksi *Silent* maka sistem akan melakukan konversi kedalam bentuk heksadesimal kemudian hasilnya akan disimpan pada format NDEF *Record.* Hasil konversi dari aksi *Silent* kebilangan heksadesimal adalah 30 2b 30 7c 31 2b 31 7c 32 2b 31 7c 32 2b 30 7c 35 2b 30 7c 36 2b 30 7c 37 2b 30

4. Setelah itu *value* tersebut akan di simpan pada *variabel-variabel* yang telah ditetapkan pada format NDEF *Record.* Berikut adalah isi dari *variabel* pada format NDEF *Record* dengan inputan aksi untuk mengaktifkan *bluetooth :*

    1. *Variabel* MB (*Message Begin*) : *variabel* ini akan selalu berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC. Sebagai awal NDEF *Message.*
    2. *Variabel* ME (*Message End*) : *variabel* ini akan selasu berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC. Sebagai akhir NDEF *Message*.
    3. *Variabel* CF (*Chunk Flag*) : *variabel* ini berisi *value* 0 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC untuk mengidentifikasi potongan record.
    4. *Variabel* SR (*Short Record*) : *variabel* ini berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC untuk menentukan *Payload* yang berukuran kecil.

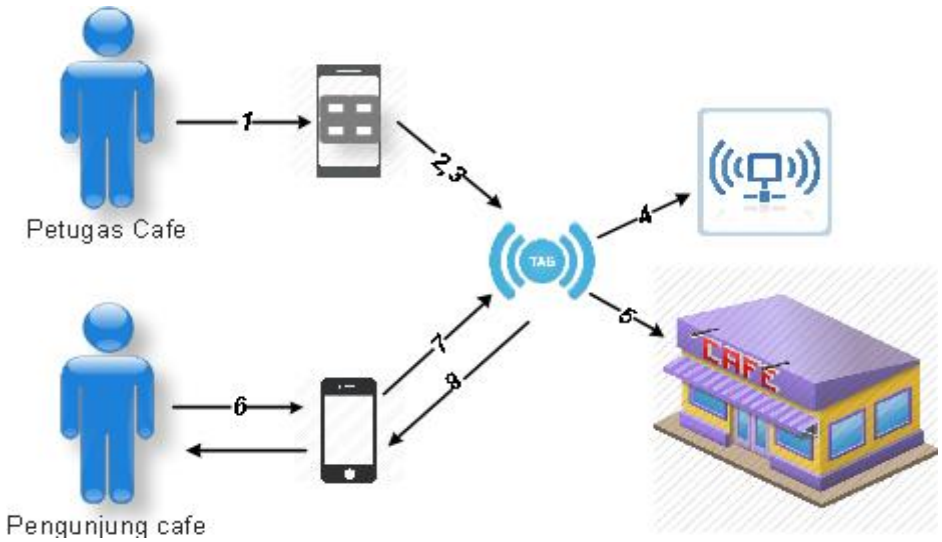
5. *Variabel* IL (*Identification Length*) : *variabel* ini berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC. Sebagai awal NDEF *Message*.
6. *Variabel* TNF (*Type Name Format*) : variabel ini akan di isi oleh *value* bit yaitu 002, value tersebut merupakan standar dari *value* TNF pada NDEF *Record* dengan type TNF MIME Media *Type*.
7. *Type Length :* berisi *value* 02 nilai 02 berasal dari value TNF MIME Media *Type*
8. *Payload Length :* berisi bilangan heksadesimal 0d nilai 0d merupakan nilai dari panjang oktet dari *Payload* dan SR.
    1. *Type :* berisi *value* 56, nilai tersebut didapat dari Type TNF yaitu *Signature*.
    2. *ID :* 01 merupakan nilai dari ID_LENGHT NFC *Record*
    3. *Payload :* 30 2b 30 7c 31 2b 31 7c 32 2b 31 7c 32 2b 30 7c 35 2b 30 7c 36 2b 30 7c 37 2b 30 merupakan bilangan heksadesimal dari aksi *Silent.*

1	1	0	1	0	022
01					
17					
56					
01					
30 2b 30 7c 31 2b 31 7c 32 2b 31 7c 32 2b 30 7c 35 2b 30 7c 36 2b 30 7c 37 2b 30					

Gambar 3. 22 Variabel NDEF *Record Silent*

3.4.3.2.3 *Analisis Koneksi WiFi*

Layanan *Koneksi WiFi* diterapkan pada layanan aplikasi ini pada fitur *Write Tags* dimana pengguna akan menginputkan *WiFi* SSID dari jaringan yang akan disambungkan, type jaringan yang akan di hubungkan dan menginputkan *WiFi Key* atau password dari jaringan *WiFi* tersebut. Pada aplikasi ini dan akan didekatkan pada *tag* NFC sehingga terjadi proses *writer* antara ponsel pengguna yang telah di inputkan *WiFi* SSID, *type WiFi* dan *WiFi Key* terhadap *tag* NFC. Berikut adalah contoh kasus penerapan analisis koneksi *WiFi* yang diterapkan pada sebuah *cafe*. Misalkan seorang pengunjung *cafe* akan melakukan koneksi terhadap jaringan WiFi yang disediakan oleh *cafe* tersebut. Tetapi pengunjung harus melakukan konfigurasi terlebih dahulu agar bisa terkoneksi pada jaringan WiFi tersebut, pengunjung harus menyalakan wifi ponselnya terlebih dahulu lalu lalu sistem WiFi yang terdapat pada ponsel tersebut akan melakukan *scanning* pada jaringan *WiFi* yang tersedia, setelah tesedia jaringan *WiFi* nya pengunjung harus menginputkan password dari jaringan WiFi tersebut. Hal seperti ini memerlukan waktu yang cukup lama oleh karena itu diterapkan sebuah analisis *write* pada *tag* NFC untuk mengkonfigurasi jaringan *WiFi*. Berikut adalah sekenario konfigurasi WiFi menggunakan teknologi NFC disebuah cafe.

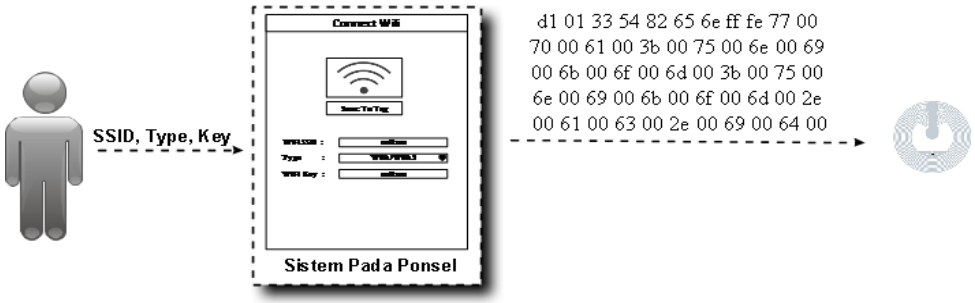


Gambar 3. 23 Skema konfigurasi WiFi pada Cafe

Sekenario Gambar :

1. Petugas *cafe* akan memasukan SSID, type dan key dari jaringan WiFi yang ada pada *cafe* tersebut.
2. Sistem akan melakukan konversi data yang telah diinputkan oleh petugas *cafe* tersebut.
3. Hasil konversi tersebut akan disimpan pada *Tag* NFC yang telah disediakan oleh *cafe*.
4. Tag NFC akan menyimpan konfigurasi dari jaringan WiFi yang ada pada *cafe* tersebut.
5. Tag NFC yang sudah terisi oleh konfigurasi jaringan WiFi tersebut akan di tempelakan di meja pengunjung atau di pintu masuk cafe aga mudah diakses oleh pengunjung.
6. Ketika pengunjung akan melakukan koneksi terhadap jaringan WiFi yang ada di *cafe* tersebut maka pengunjung tinggal mendekatkan ponsel yang telah memiliki teknologi NFC pada Tag NFC yang telah diisi oleh konfigurasi tadi.
7. Ponsel pengguna akan melakukan *reader* terhadap tag NFC yang telah didekatkan.
8. Setelah itu ponsel pengguna akan melakukan aksi sesuai dengan isi yang telah diinputkan.

Berikut adalah proses *write* untuk konfigurasi *WiFi.*



Gambar 3. 24 Proses Koneksi WiFi

1. Inputan dari pengguna pada sistem yaitu berupa *text* , misalkan pengguna meng*input*akn *WiFi* SSID adalah unikom, *type WiFi* adalah WPA/WPA2 dan *WiFi Key* adalah unikom.ac.id. Maka sistem akan melakukan pengecekan terhadap inputan tersebut. Pertama sistem akan mengecek tipe TNF nya terlebih dahulu, pengecekan tipe TNF ini disesuaikan dengan tabel 2.2 *Value Type Name Format* yang telah di tetapkan oleh NFC. Sesuai dengan contoh kasus maka inputan dari pengguna merupakan untuk *WiFi,* tipe TNF nya adalah MIME Media *type* maka value dari tipe TNF ini adalah 0x02 atau 02.
2. Setelah sistem melakukan pengecekan terhadap aksi yang telah diinputkan, maka sistem akan melakukan pemisahan kode SSID, type dan key dengan tanda " ; ", setelah itu setiap kode tersebut akan di pisahkan dengan tanda " . " untuk mengetahui value dari inputan tersebut. Maka dapat di peroleh hasil sebagai berikut sesuai inputan dari pengguna.

   ..'T.en.w.p.a.;.u.n.i.k.o.m.;.u.n.i.k.o.m.

   selanjutnya sistem akan melakukan konversi pada sisa *aksi tersebut* kedalam bentuk heksadesimal.
3. Kemudian hasilnya akan disimpan pada format NDEF *Record.* Hasil konversi dari aksi *inputan* tersebut kebilangan heksadesimal adalah d1 01 33 54 82 65 6e ff fe 77 00 70 00 61 00 3b 00 75 00 6e 00 69 00 6b 00 6f 00 6d 00 3b 00 75 00 6e 00 69 00 6b 00 6f 00 6d 00 2e 00 61 00 63 00 2e 00 69 00 64 00
4. Setelah itu *value* tersebut akan di simpan pada *variabel-variabel* yang telah ditetapkan pada format NDEF *Record.* Berikut adalah isi dari *variabel* pada format NDEF *Record* dengan inputan aksi untuk mengaktifkan *bluetooth :*

1. *Variabel* MB (*Message Begin*) : *variabel* ini akan selalu berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC. Sebagai awal NDEF *Message.*
2. *Variabel* ME (*Message End*) : *variabel* ini akan selasu berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC. Sebagai akhir NDEF *Message.*
3. *Variabel* CF (*Chunk Flag*) : *variabel* ini berisi *value* 0 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC untuk mengidentifikasi potongan record.
4. *Variabel* SR (*Short Record*) : *variabel* ini berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC untuk menentukan *Payload* yang berukuran kecil.
5. *Variabel* IL (*Identification Length*) : *variabel* ini berisi *value* 1 karena *value* dalam *variabel* ini sudah ditentukan oleh NFC. Sebagai awal NDEF *Message.*
6. *Variabel* TNF (*Type Name Format*) : variabel ini akan di isi oleh *value* bit yaitu 002, value tersebut merupakan standar dari *value* TNF pada NDEF *Record* dengan type TNF MIME Media *Type*.
7. *Type Length :* berisi *value* 02 nilai 02 berasal dari value TNF MIME Media *Type*
8. *Payload Length :* berisi bilangan heksadesimal 0d nilai 0d merupakan nilai dari panjang oktet dari *Payload* dan SR.
9. *Type :* berisi *value* 54, nilai tersebut didapat dari Type TNF yaitu *Text*.
    1. *ID :* 01 merupakan nilai dari ID_LENGHT NFC *Record*
2. *Payload :*

    d1 01 27 54 82 65 6e ff

fe 77 00 70 00 61 00 3b

00 75 00 6e 00 69 00 6b

00 6f 00 6d 00 3b 00 75

00 6e 00 69 00 6b 00 6f

00 6d 00

merupakan bilangan heksadesimal dari aksi koneksi *WiFi*.

1	1	0	1	0	002
01					
27					
54					
01					
d1 01 33 54 82 65 6e ff fe 77 00 70 00 61 00 3b 00 75 00 6e 00 69 00 6b 00 6f 00 6d 00 3b 00 75 00 6e 00 69 00 6b 00 6f 00 6d 00 2e 00 61 00 63 00 2e 00 69 00 64 00					

Gambar 3. 25 Variabel NDEF Record Wifi

3.5 Analisis Kebutuhan Non-fungsional

Ananlisis kebutuhan non-fungsional merupakan penguraian kebutuhan-kebutuhan non fungsional (batasan sistem) menjadi kebutuhan yang lebih detail berdasarkan kebutuhan yang ada. Spesifikasi non-fungsional melibatkan beberapa kebutuhan, perangkat lunak, perangkat keras dan pengguna.

3.5.1 Analisis Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembangunan aplikasi ini adalah:

1. Perangkat lunak PC:
   a) Sistem Operasi yang digunakan adalah Windows 8.1 64-bit
   b) ADT (*Android Development Tools*) *Bundle* v23.0.3-1085508
   c) Java Development Kit v7.0.510.13

2. Perangkat lunak *Smartphone* yang digunakan adalah sistem operasi Android versi 4.0 *Ice Cream Sandwich.*

Spesifikasi minimum perangkat lunak *Smartphone* yang dibutuhkan untuk dapat menjalankan aplikasi ini adalah sistem operasi yang berdasarkan minimum sistem operasi pada NFC yaitu Android versi 4.0 *Ice Cream Sandwich.*

3.5.2 Analisis Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pembangunan aplikasi ini adalah:

1. Perangkat keras PC:
   a) *Processor* Intel Core i5 2.6 GHz.
   b) RAM yang digunakan 4 GB.
   c) *Hardisk* 750 G.
2. Perangkat keras *Smartphone*:
   a) *Processor* 1,5 GHz.
   b) RAM 1 Gb.
   c) Media penyimpanan internal 32 GB.
   d) Terpasang teknologi NFC.

Spesifikasi minimum perangkat keras *Smartphone* yang dibutuhkan untuk dapat menjalankan aplikasi ini adalah:

a) *Processor* 600 MHz.
b) RAM 256 Mb.
c) Media penyimpanan internal *free space* 15 MB.
d) Media penyimpanan eksternal 1 GB.
e) Terpasang teknologi NFC

3.5.3 Analisis Pengguna

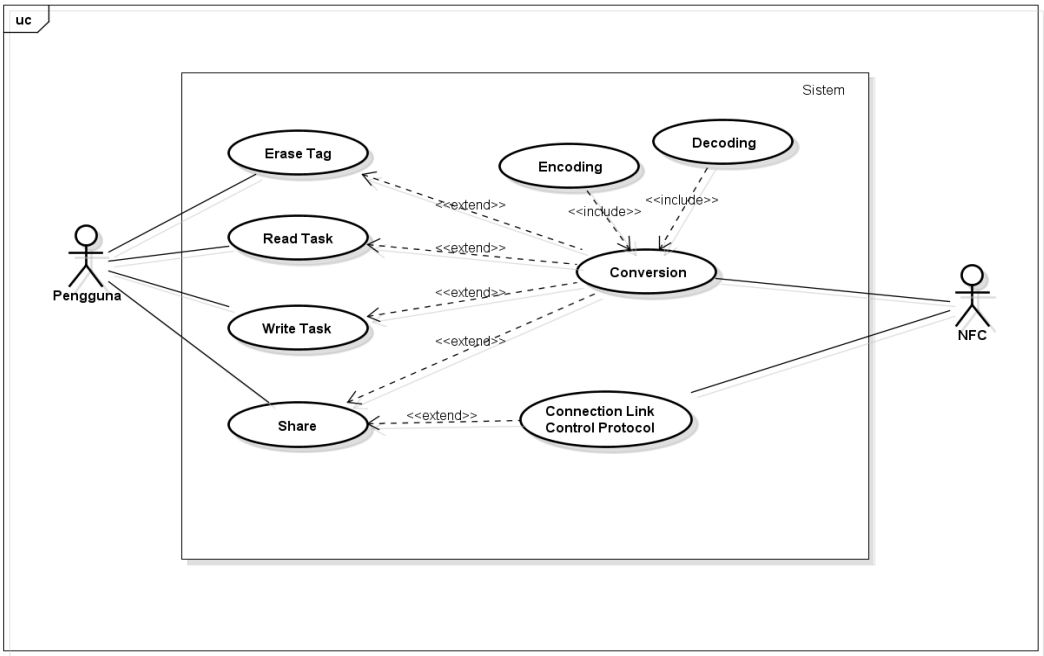
Analisis pengguna sistem dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat dalam menjalankan sistem. Pengguna pada aplikasi ini yaitu publik, semua orang yang dapat mengoperasikan *smartphone* khususnya *platform* android.

3.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik. Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain *Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram*.

3.6.1 Use Case Diagram

Use Case Diagram mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. *Use Case* terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi *use case* dan skenario *use case*. Analisis *Use Case* Diagram pada sistem yang akan bibangun dapat dilihat pada Gambar.



Gambar 3. 26 Diagram Use Case

1. Identifikasi Aktor

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibangun. Berikut adalah aktor-aktor yang terdapat pada sistem yang akan dibangun.

Tabel 3. 4 Identifikasi Aktor

No	Aktor	Deskripsi
A-01	Pengguna	Merupakan aktor dari aplikasi yang dibangun yang berperan sebagai pengguna aplikasi
A-02	NFC	Merupakan aktor yang berperan sebagai penyimpan semua informasi yang diinputkan oleh sistem dan aktor yang menghubungkan antara perangkat pengguna.

2. Identifikasi Use Case

Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Pada tabel 3.4 adalah deskripsi use case yang terdapat pada sistem yang akan dibangun.

Tabel 3. 5 Identifikasi Use Case

No	Usecase	Deskripsi
UC-01	*Write Task*	Fungsionalitas untuk membuat sebuah *task* atau perintah yang akan digunakan untuk di *share* ataupun untuk disimpan pada *tag* NFC
UC-02	*Read Task*	Fungsionalitas untuk membaca sebuah *task* yang telah di inputkan pada *Tag* NFC
UC-02	*Share*	Fungsionalitas untuk membagikan *task* yang telah dibuat atau untuk membagikan text, url dan file yang terdapat pada ponsel
UC-03	*Erase Tag*	Fungsionalitas untuk menghapus *task* yang telah di inputakan ke *tag* NFC
UC-04	*Conversion*	Fungsionalitas untuk merubah data inputan menjadi bilangan heksadesimal
UC-05	*Connection Link Control Protocol*	Fungsionalitas untuk mendukung komunikasi *peer-to-peer* antara dua perangkat yang berkemampuan NFC
UC-06	*Encoding*	Fungsional untuk melakukan proses konversi dari data inputan pengguna ke bilangan okta atau bilangan heksadesimal sesuai dengan format

		NDEF *Record*
UC-07	*Decoding*	Fungsional untuk melakukan proses konversi dari bilangan okta atau bilangan heksadesimal ke dalam bentuk semula.

3. Skenario *Use Case*

Skenario *Use Case* mendeskripsikan urutan langkah-langkah dalam proses bisnis, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor.

a. Skenario *Use Case Write*

Tabel 3. 6 Skenario Use Case *Write Task*

Identifikasi	
Nama *Use Case*	*Write Task*
Aktor	Pengguna
Tujuan	Membuat *task* yang baru pada *tag* NFC
Kondisi Awal	Menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*Write Task*"	
	2. Menampilkan halaman *Write Task*.
3. Memilih fungsional yang telah disediakan pada halaman *Write Task*	
	4. Menampilkan halaman sesuai fungsional yang telah dipiliih oleh pengguna
5. Memasukan informasi atau perintah sesuai dengan inputan yang diinginkan oleh pengguna	
6. Menekan button *Go*	
	7. Mengecek inputan dari pengguna
	8. Mengelompokan inputan sesuai dengan format yang telah ditentukan oleh sistem
	9. Merubah inputan kedalam

	bilangan heksadesimal
	10. Menyimpan hasil bilangan heksadesiaml pada NDEF *Record*
	11. Menampilkan pesan "*Please tap a NFC tag to write*"
12. Mendekatkan ponselnya dengan *tag* NFC	
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*Go"*	
	2. Gagal menyimpan *task* yang dibuat pada *tag* NFC
Kondisi Akhir	Menampilkan halaman sesuai dengan fungsional yang telah dipilih

b. Skenario *Use Case Read*

Tabel 3. 7 Skenario *Use Case Read*

Identifikasi	
Nama *Use Case*	*Read*
Aktor	*User, Tag NFC*
Tujuan	Menampilkan informasi yang terdapat pada *tag* NFC
Kondisi Awal	Menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu "*Read Task*"	
	2. Mengecek *tag* NFC yang akan dibaca
	3. Membaca NDEF *Record* yang beada pada *tag* NFC
	4. Merubah bilangan heksadesimal kedalam bentuk semula
	5. Menampilkan informasi yang ada pada *tag* NFC
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**

Kondisi Akhir	Menampilkan halaman *My Task*

c. Skenario *Use Case Share*

Tabel 3. 8 Skenario *Use Case Share*

Identifikasi	
Nama *Use Case*	*Share*
Aktor	*User, Ponsel*
Tujuan	Menampilkan halaman *share*
Kondisi Awal	Menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*share*"	
	2. Menampilkan halaman *share*
	3. Menampilkan menu jenis file yang akan di *share*
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*share*"	
	2. Gagal menampilkan halaman *share*
	3. Gagal menu jenis file yang akan di *share*
Kondisi Akhir	Menampilkan halaman *Share File*

d. Skenario *Use Case Erase Tag*

Tabel 3. 9 Skenario *Erase Tag*

Identifikasi	
Nama *Use Case*	*Erase Tag*
Aktor	*NFC Tag*
Tujuan	Menghapus *task* yang telah di inputkan pada *tag* NFC
Kondisi Awal	Menampilkan halaman *setting*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan tombol "*erase tag*"	

	2. Mencari perangkat *tag* NFC
3. Mendekatkan *tag* NFC yang akan dihapus	
	4. Menghapus *task* yang telah di inputkan pada *tag* NFC
	5. Menampilkan pesan "*tag erase successfully*"
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
1. Menekan tombol "*erase tag*"	
	2. Menampilkan pesan "*Tag erase not successfully*"
Kondisi Akhir	Menampilkan halaman Utama

e. Skenario *Use Case* Convertion

Tabel 3. 10 Skenario Use Case Convertion

Identifikasi	
Nama *Use Case*	*Convertion*
Aktor	*Mobile Phone, NFC Tag*
Tujuan	Merubah data inputan dari pengguna kedalam bilangan heksadesimal
Kondisi Awal	Menampilkan halaman *write tags*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih fungsional ynagbterdapat pada menu *write tags*	
2. Menginputkan data sesuai dengan kebutuhan dari fungsional tersebut	
	3. Merubah inputan data dari pengguna kedalam benuk heksadesimal
	4. Menyimpan hasil dari konversi data tersebut kedalam NDEF *Message* tepatnya pada NDEF *Record*
	5. Membaca bilangan yang ada pada NDEF *Record*
	6. Merubah isi dari NDEF

	Record kedalam bentuk data semula
	7. Menampilkan hasil data konversi dari bentuk heksadesimal ke dalam bentuk data semula
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
Kondisi Akhir	Menampilkan informasi yang telah di konversi

f. Skenario *Use Case Connection Control Link Protocol*

Tabel 3. 11 Skenario Use Case Connection

Identifikasi	
Nama *Use Case*	*Connection Control Link Protocol*
Aktor	*Mobile Phone*
Tujuan	Menghubungkan komunikasi *peer-to-peer* antar dua perangkat yang terdapat teknologi NFC
Kondisi Awal	Menampilkan *Task* yang akan di *share* sudah dibuat
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mencari perangkat yang sama yang mampu melakukan komunikasi menggunakan NFC	
	2. Menghubungkan komunikasi *peer-to-peer* dua perangkat tersebut
	3. Mengirimkan *task* yang akan di *share*
	4. Menampilkan dialog "*Waiting for tag/share*"
	5. Menampilkan dialog "*Succes write for tag/share*"
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Menghubungkan komunikasi *peer-to-peer* dua perangkat tersebut

	2. Menampilkan dialog "*Place your connecting for device*."
Kondisi Akhir	Menampilkan halaman *share*

g. Sekenario *Use Case Encoding*

Tabel 3. 12 sekenario Use Case Encoding

Identifikasi	
Nama *Use Case*	*Encoding*
Aktor	Ponsel
Tujuan	Merubah data inputna pengguna kedalam bentuk heksadesimal
Kondisi Awal	-
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menginputkan data sesuai dengan kebutuhan dari fungsional tersebut	
	2. Merubah inputan data dari pengguna kedalam benuk heksadesimal
	3. Menyimpan hasil dari konversi data tersebut kedalam NDEF *Message* tepatnya pada NDEF *Record*
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
Kondisi Akhir	Menampilkan hasil ari konversi data kedalam bentuk heksadesimal

h. Sekenario *Use Case Decoding*

Tabel 3. 13 Sekenario Use Case Decoding

Identifikasi		
Nama *Use Case*	*Decoding*	
Aktor	*Ponsel*	
Tujuan	Merubah bilangan heksadesimal kedalam bentuk semula	
Kondisi Awal	-	
Skenario Utama		
Aksi Aktor		**Reaksi Sistem**
1. Mendekatkan sistem pada *tag* NFC		
		2. Akan membaca isi informasi yang tersimpan di dalam NDEF *Record*
		3. Akan merubah isi informasi yang terdapat pada NDEF *Record* Kedalam bentuk semula
		4. Menampilkan hasil konversi dalam bentuk data semula
Skenario Alternatif		
Aksi Aktor		**Reaksi Sistem**
Kondisi Akhir	Menampilkan halaman data yang sudah di konversi kedalam bentuk semula	

3.6.2 Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity* diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behaviour* internal sebuah sistem (interaksi antar subsistem)

secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

1. Aktivitas *Write Task*

   Aktivitas yang terdapat dalam *Write Task* adalah melakukan inputan berupa informasi atau perintah kedalam *Tag* NFC. Dalam aktivitas ini, terdapat tiga sub aktivitas inputan terhadap *tag* NFC, yaitu *Write Url, Connect WiFi,* dan *Other Action.*

   Tahapan-tahapan yang terjadi dalam sistem adalah sebagai berikut.

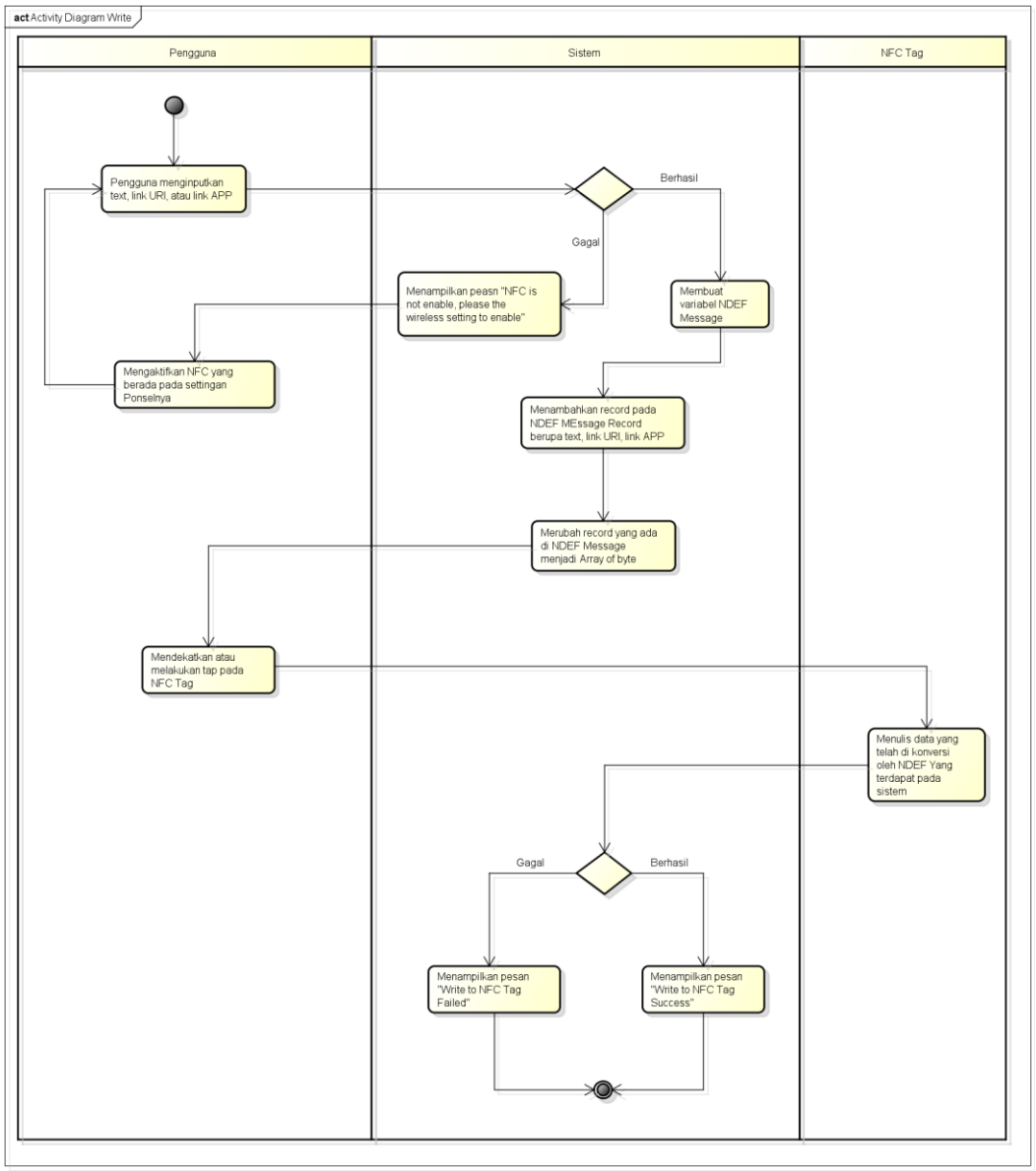
   a. Memilih menu *Write Tags,* aktivitas diawali dengan memilih menu *Write Tags*.
   b. Menampilkan sub menu dari *Write Tags* yaitu, *Write Url*, *Connect WiFi* dan *Other Action*
   c. Pengguna memilih fungsional dari Write Tags yaitu Write Url, Connect WiFi, dan Other Action.
   d. Jika memilih fungsional *Write Url*, maka sistem akan menampilkan halaman untuk menginputkan *Url.* Dan pengguna akan menginputkan Url atau Link kedalam sistem.
   e. Jika memilih fungsional *Connect WiFi*, maka sistem akan menampilkan halaman *Connect WiFi*. Pengguna akan meng inputkan SSID, *type WiFi* dan *Key*.
   f. Jika memilih fungsional Other Action, maka sistem akan menampilkan halaman list action seperti *Airplane Mode*, *WiFi*, *Bluetooth*, *Silent Ringer*, *Alarm* dan *Launching App.*
   g. Menampilkan pesan untuk mengaktifkan terlebih dahulu *settingan* NFC bila ponsel NFC Pengguna belum diaktifkan.
   h. Jika pengguna telah memilih salah satu fungsional dan telah menginputkan sesuai dengan inputan fungsionalnya maka sistem akan

menampilkan pesan untuk mendekatkan ponsel dengan telnologi NFC yang lainnya seperti *Tag* NFC.

i. Jika NFC yang lainnya tidak terditeksi maka sistem akan menampilkan pesan *tag* NFC tidak terditek.

j. Menampilkan halaman sesuai fungsional yang telah dipilih oleh pengguna.

Seluruh aktifitas *Write Task* dapat dilihat tahapannya pada diakram Activity Write Task pada Gambar 3.26.



Gambar 3. 27 Diagram Aktivitas *Write Task*

2. Aktivitas *Read Task*

Aktivitas yang terdapat dalam *Read Task* adalah melakukan pembacaan terhadap informasi atau perintah yang ada didalam *Tag* NFC. Dalam aktivitas ini sistem secara otomastis akan membaca setiap di dekatkan dengan perangkat NFC.

Tahapan-tahapan yang terjadi dalam sistem adalah sebagai berikut.

a. Pengguna akan medekatkan perangkat NFC seperti tag NFC dengan sistem.

b. Sistem akan melakukan pembacaan informasi yang terdapat pada setiap NDEF *Record* perangkat NFC yayng didekatkan.

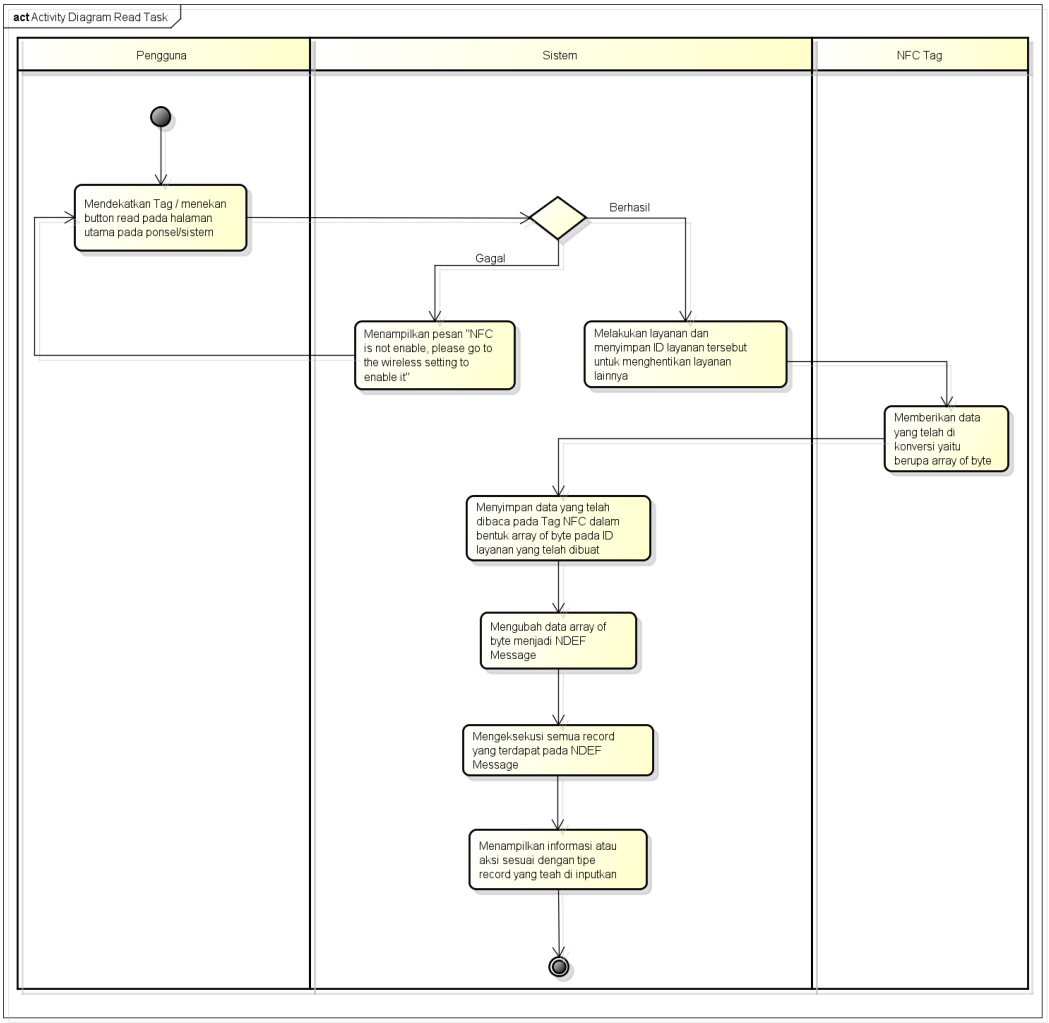
c. NFC Tag akan memberikan data yang telah di konversikan yaitu berupa *array of byte*

d. Sistem akan menyimpan array tersebut pada format NDEF *Message*.

e. Sistem akan mengeksekusi semua record yang terdapat pada NDEF *Message.*

f. Sistem akan menampilkan informasi yang didapatkannya pad NDEF *Message.*

Seluruh aktifitas *Read Task* dapat dilihat tahapannya pada diakram Activity *Write Task* pada Gambar 3.27.



Gambar 3. 28 Diagram Aktivitas *Read Task*

3. Aktivitas *Share*

Aktivitas yang terdapat dalam *Share* adalah melakukan inputan perintah berupa pilihan *file* yang akan di *share* kan.

Tahapan-tahapan yang terjadi dalam sistem adalah sebagai berikut.

a. Pengguna akan memilih menu *share* dan sistem akan menampilkan halaman *share*.

b. Pengguna akan memilih *file* yang ajan di bagikan.

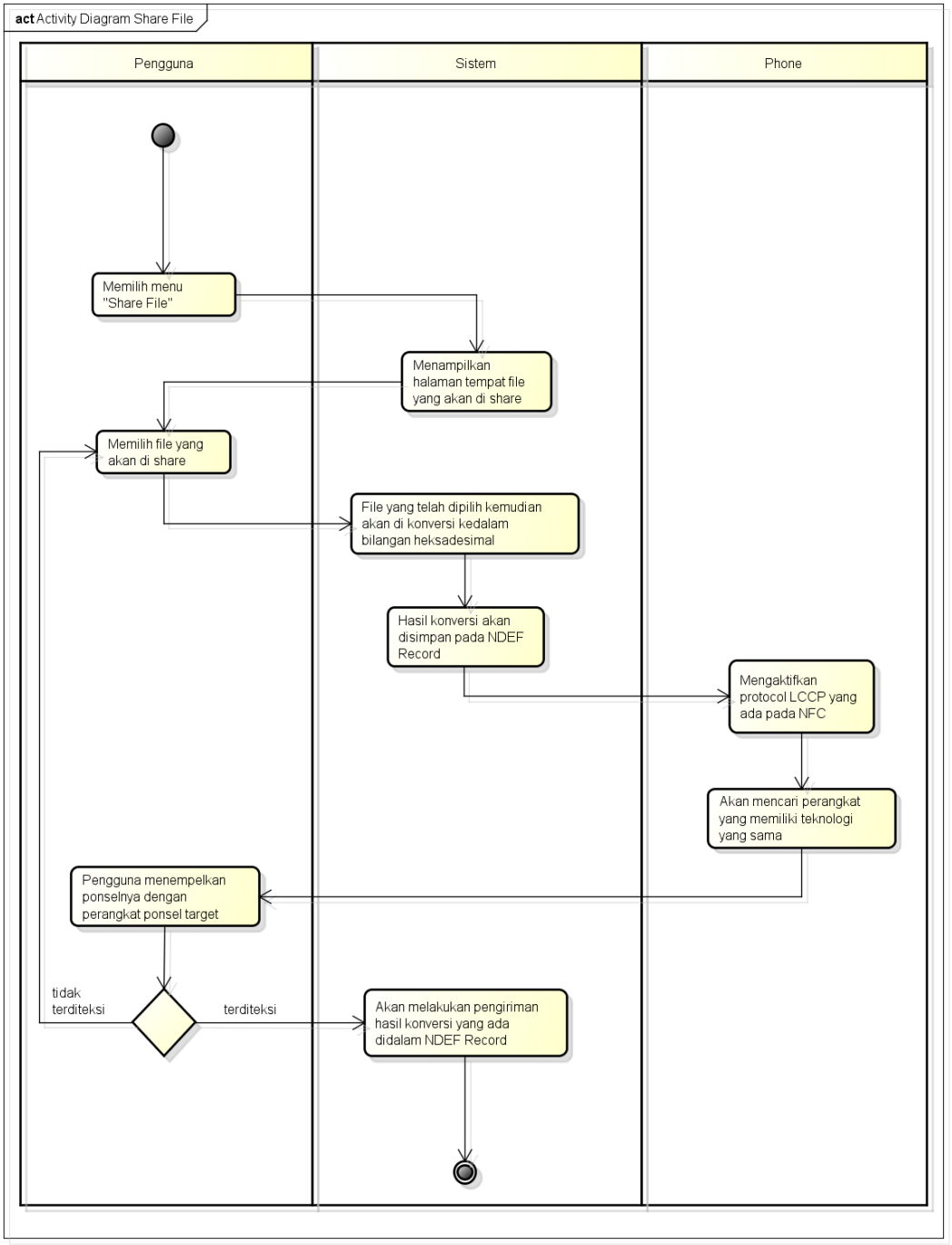
c. File yangtelah di pilih kemudian akan dikonversikan oleh sistem kedalam bentuk heksadesimal.

d. Sistem akan menyimpan hasil konversi tersebut dalam NDEF *Record.*

e. Sistem akan mengaktifkan protocol LCCP yang ada pada perangkat ponsel pengguna.

f. Sistem akan melakukan pengiriman hasil konversi tersebut sesama perangkat ponsel yang telah terhubung.

Seluruh aktifitas *Share* dapat dilihat tahapannya pada diakram Activity Write Task pada Gambar 3.28.



Gambar 3. 29 Diagram Aktivitas *Share*

4. Aktivitas *Erase Tag*

Aktivitas yang terdapat dalam *Write Task* adalah melakukan penghapusan terhadap task atau informasi yang telah di inputkan kedalam *tag* NFC.

Tahapan-tahapan yang terjadi dalam sistem adalah sebagai berikut.

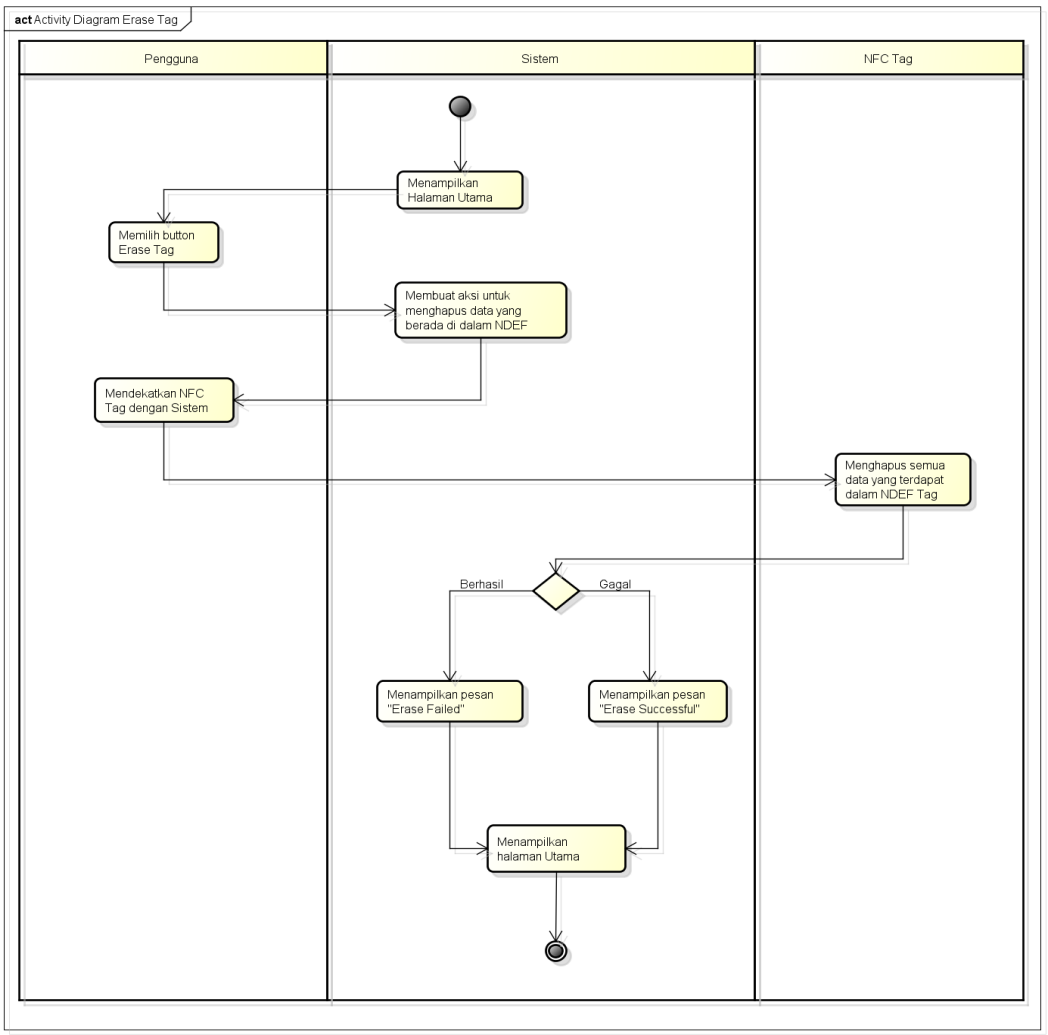
a. Pengguna memilih menu erase tags

b. Sistem akan membuat aksi untuk menghapus semua aksi atau informasi yang ada pada NDEF *Mesasge*

c. Sistem akan mengirimkan semua aksi untuk *erase* tersebut pada NDEF *Message* tag NFC.

d. NDEF *Message* pada tag telah kosong dan aplikasi akan menampilkan halaman *erase*

Seluruh aktifitas *Erase Tag* dapat dilihat tahapannya pada diakram Activity Write Task pada Gambar 3.29.



Gambar 3. 30 Diagram Aktivitas *Erase Tag*

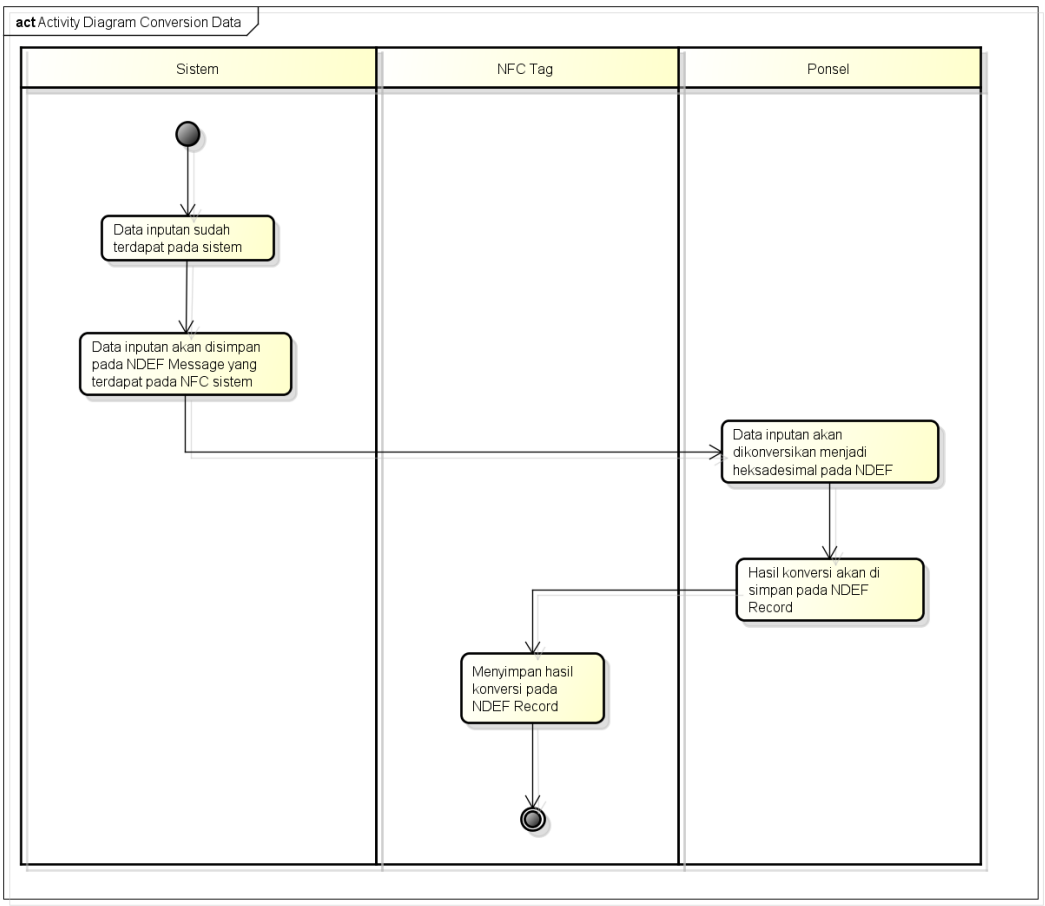
5. Aktivitas *Convertion*

Aktivitas yang terdapat dalam *Convertion Data* adalah melakukan inputan konveri dari data inputan pengguna kedalam bilangan heksadesimal yang telah ditetapkan pada format NDEF *Message*.

Tahapan-tahapan yang terjadi dalam sistem adalah sebagai berikut.

a. Data inputan sudah terdapat pada sistem
b. Data inputan tersebut akan disimpan sementara pada NDEF Message sistem.
c. Kemudian data tersebut akan di konversi oleh sistem kedalam bentuk hekasdesimal.
d. Hasil konversi tersebut akan disimpan pada NDEF *Record* ponsel dan NDEF *Record Tag* NFC.

Seluruh aktifitas *Convertion data* dapat dilihat tahapannya pada diakram Activity Write Task pada Gambar 3.30.



Gambar 3. 31 Diagram Aktivitas *Convertion*

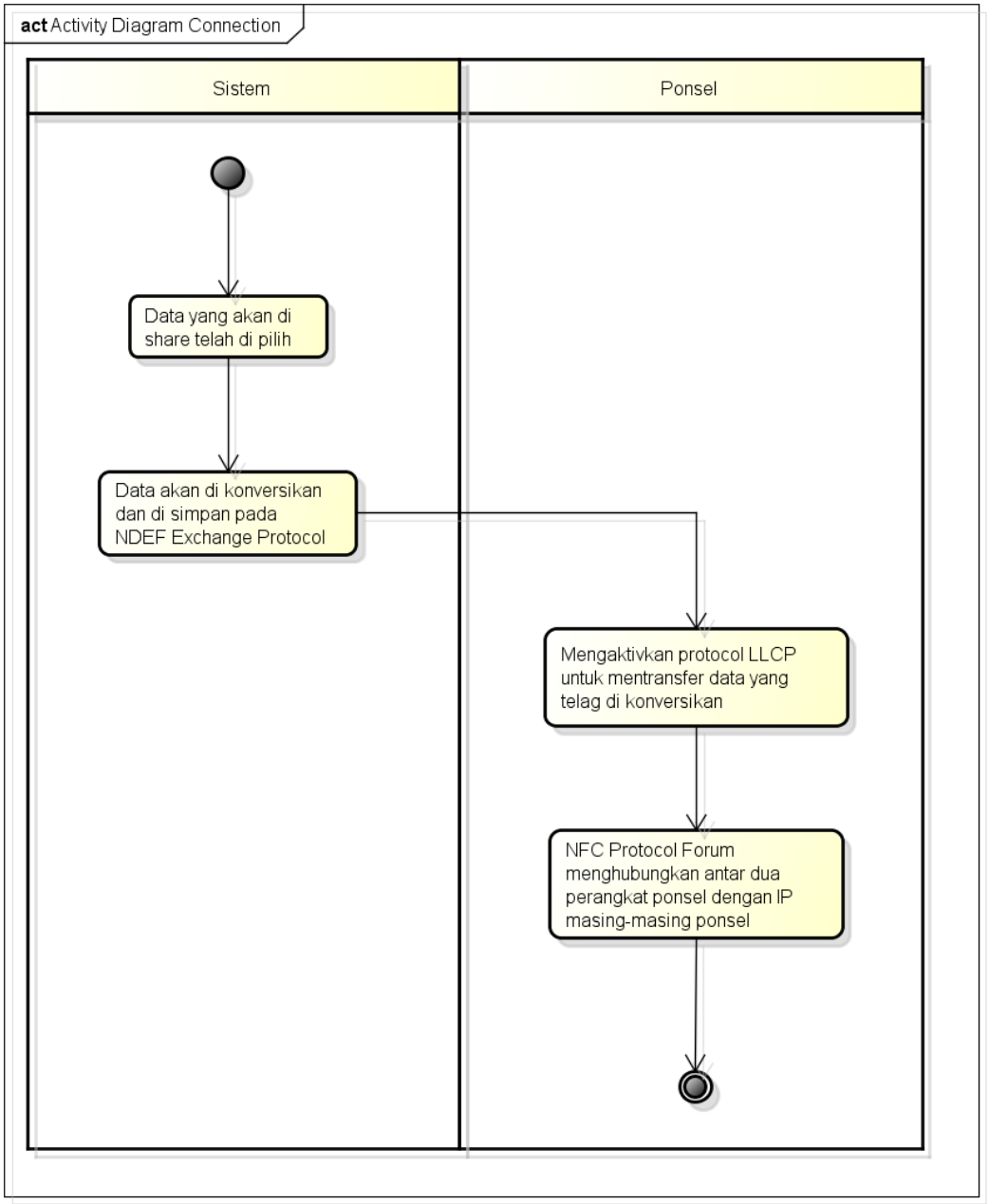
6. Aktivitas *Connection Link Control Protocol*

Aktivitas yang terdapat dalam *Connection* adalah melakukan *conektivitas* antar perankat ponsel yang memiliki kemampuan teknologi NFC.

Tahapan-tahapan yang terjadi dalam sistem adalah sebagai berikut.

a. Data yang akan di bagikan telah dipilih
b. Kemudian data tersebut akan dikonversi kedalam formant NDEF *Message*
c. Ponsel yang memiliki kemampuan NFC akan mengaktipkan protocol LLCP untuk menghubungkan antar kedua perangkat ponsel tersebut
d. Sistem akan melakuka pengiriman file tersebut.

Seluruh aktifitas *Connection* dapat dilihat tahapannya pada diakram Activity Write Task pada Gambar 3.31.

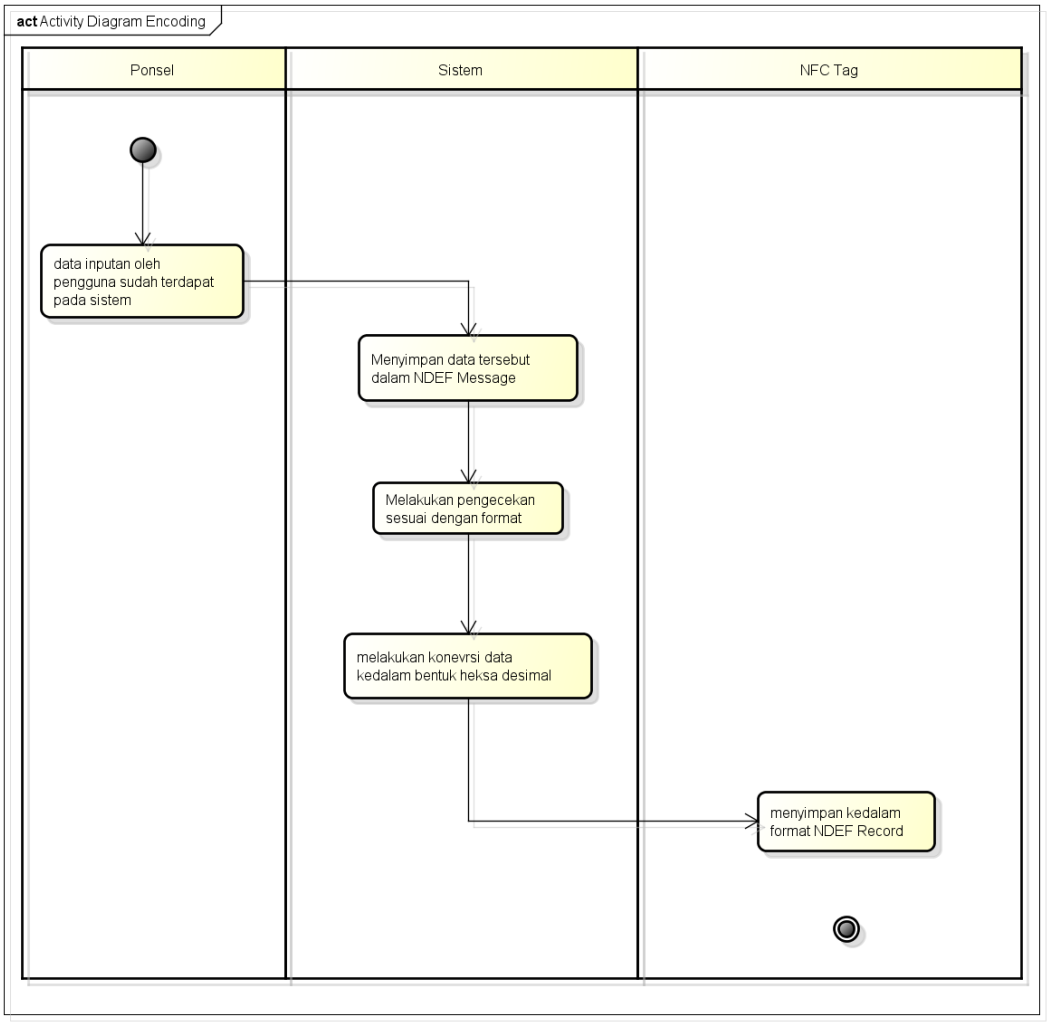


Gambar 3. 32 Diagram Aktivitas *Connection*

7. Aktivitas *Encoding*

Aktivitas yang terdapat dalam *Encoding* adalah melakukan konversi dari data inputan pengguna kedalam bentuk heksadesimal sesuia dengan format NDEF *Message*.

Seluruh aktifitas *Encoding* dapat dilihat tahapannya pada diakram Activity Write Task pada Gambar 3.32.

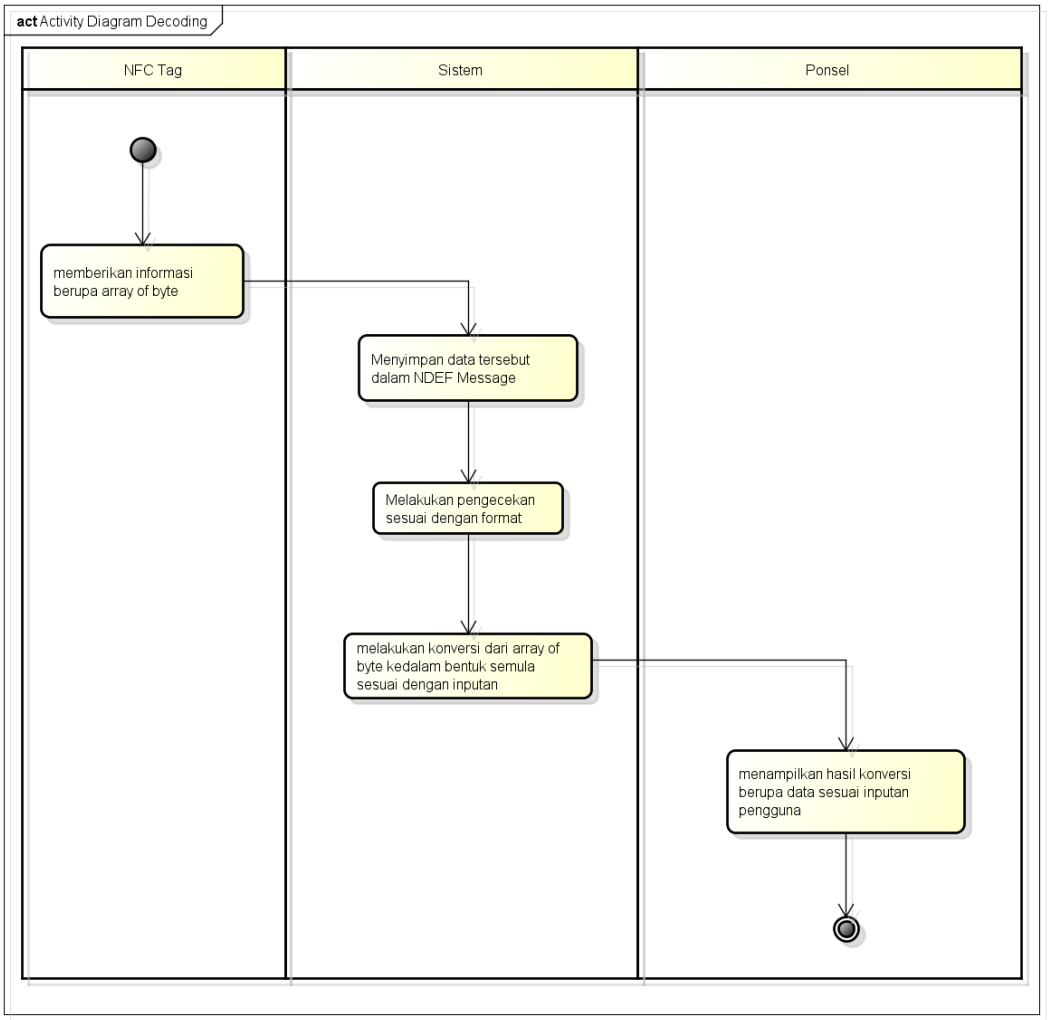


Gambar 3. 33 Diagram Aktivitas *Encoding*

8. Aktivitas *Decoding*

Aktivitas yang terdapat dalam *Decoding* adalah melakukan konversi dari format NDEF *Message* kedalam bentuk date semula yang telah diinputkan .

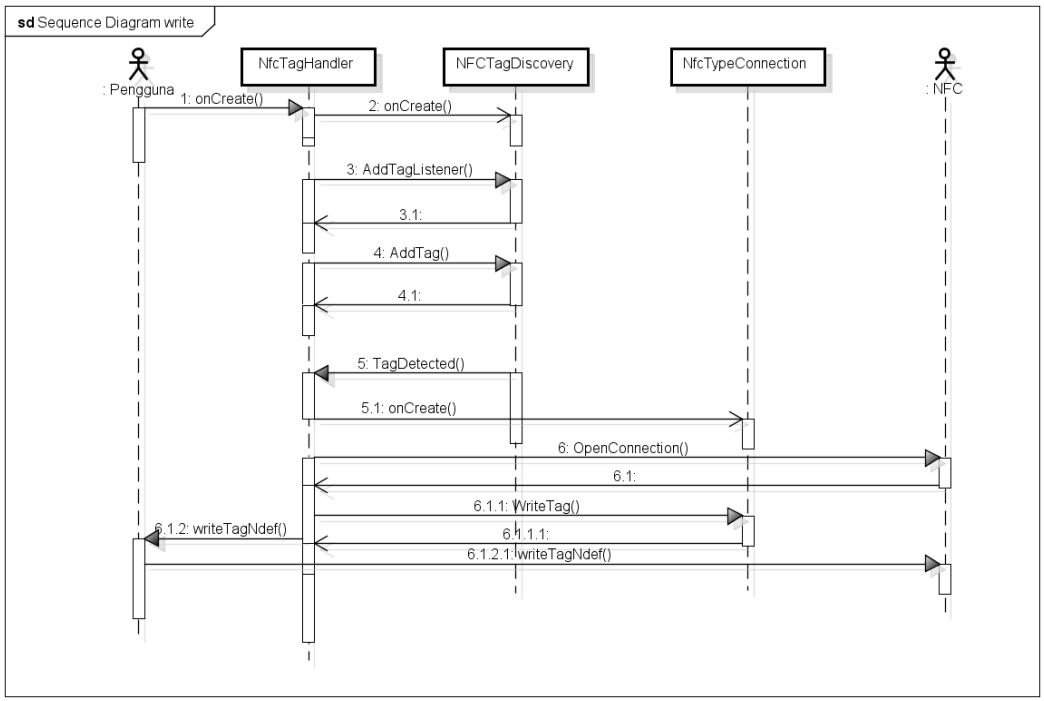
Seluruh aktifitas *Decoding* dapat dilihat tahapannya pada diakram Activity Write Task pada Gambar 3.33.



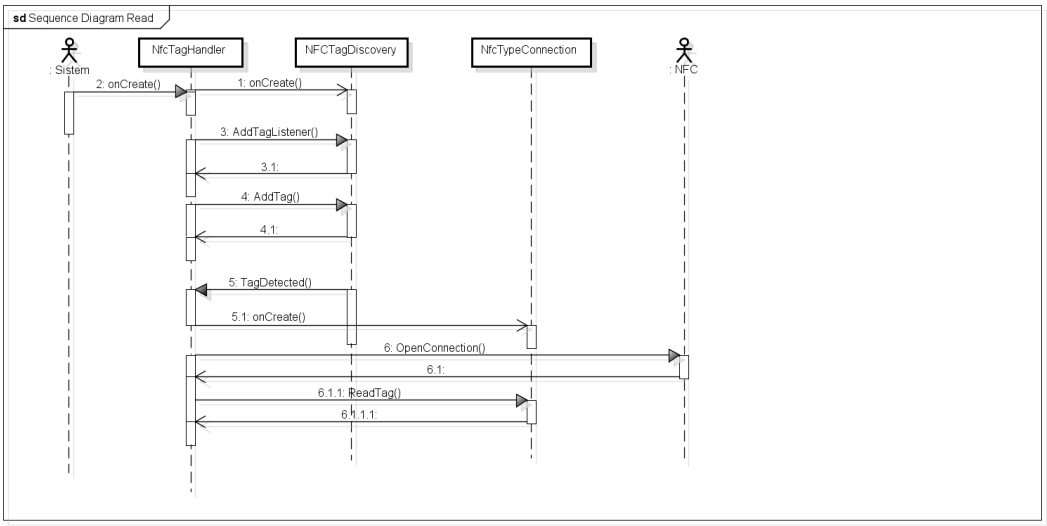
Gambar 3. 34 Diagram Aktivitas *Decoding*

3.6.3 Squence Diagram

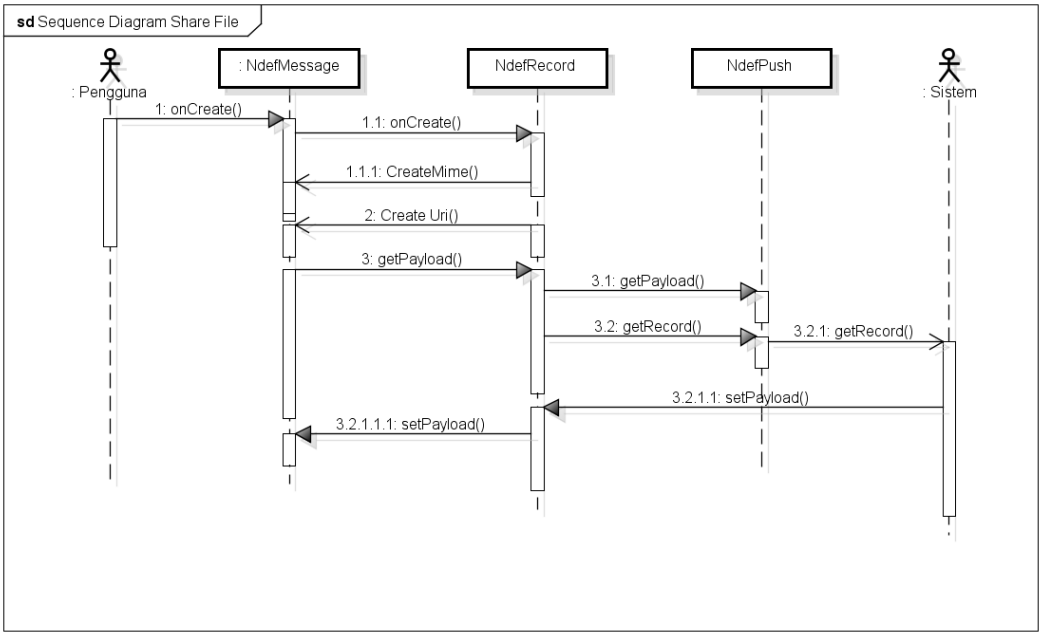
Sequence Diagram, menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.



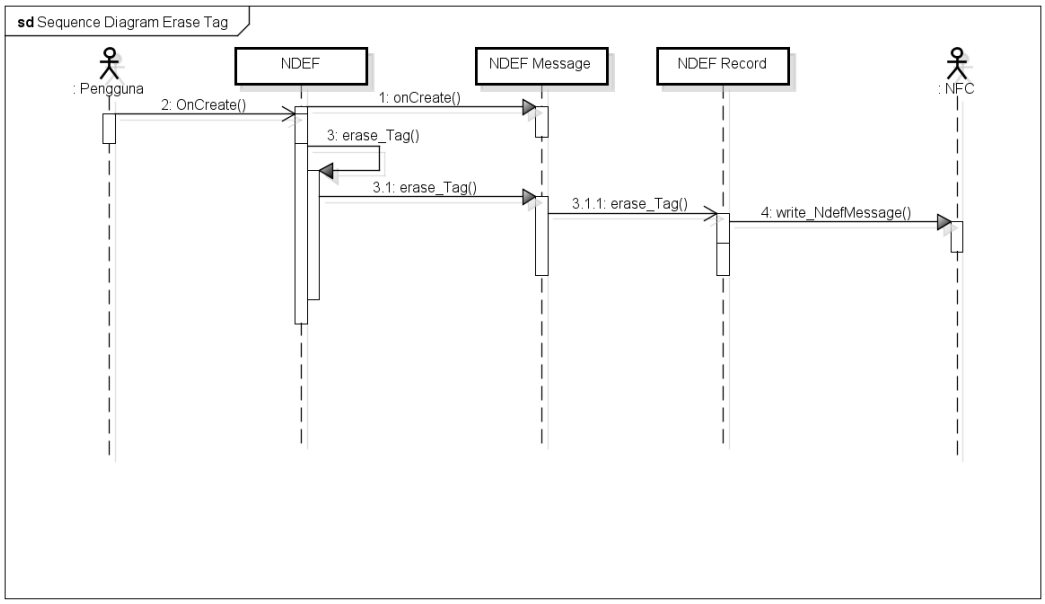
Gambar 3. 35 *Sequence* Diagram *Write*



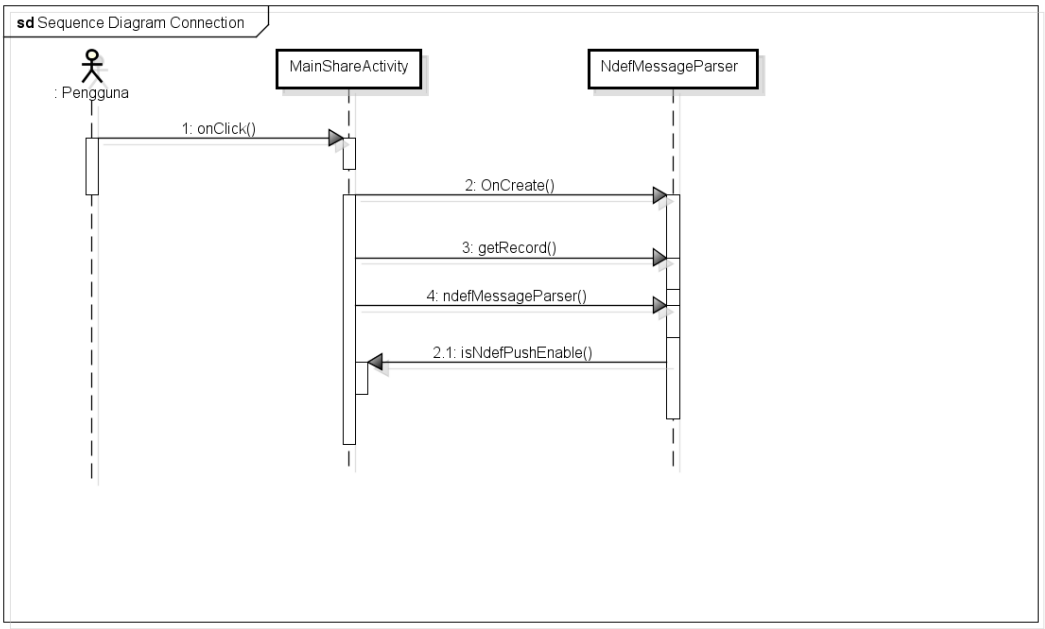
Gambar 3. 36 *Sequnce* Diagram *Read*



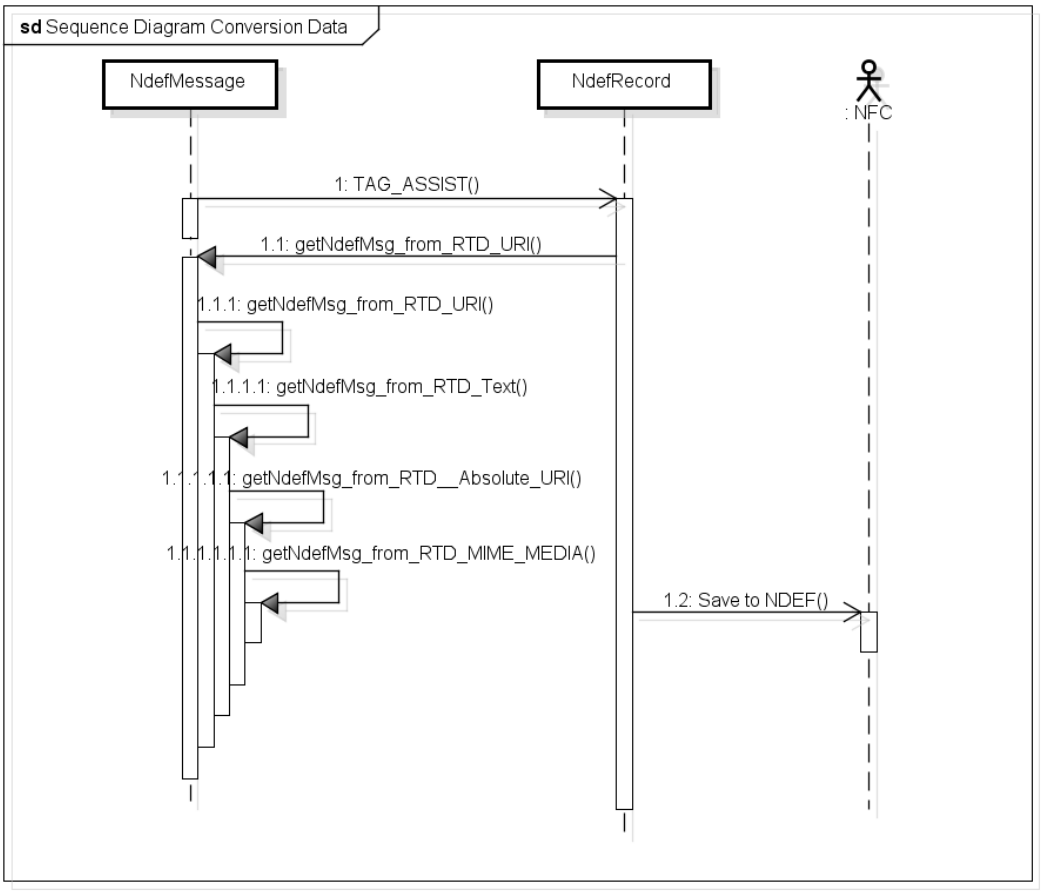
Gambar 3. 37 *Sequence* Diagram *Share*



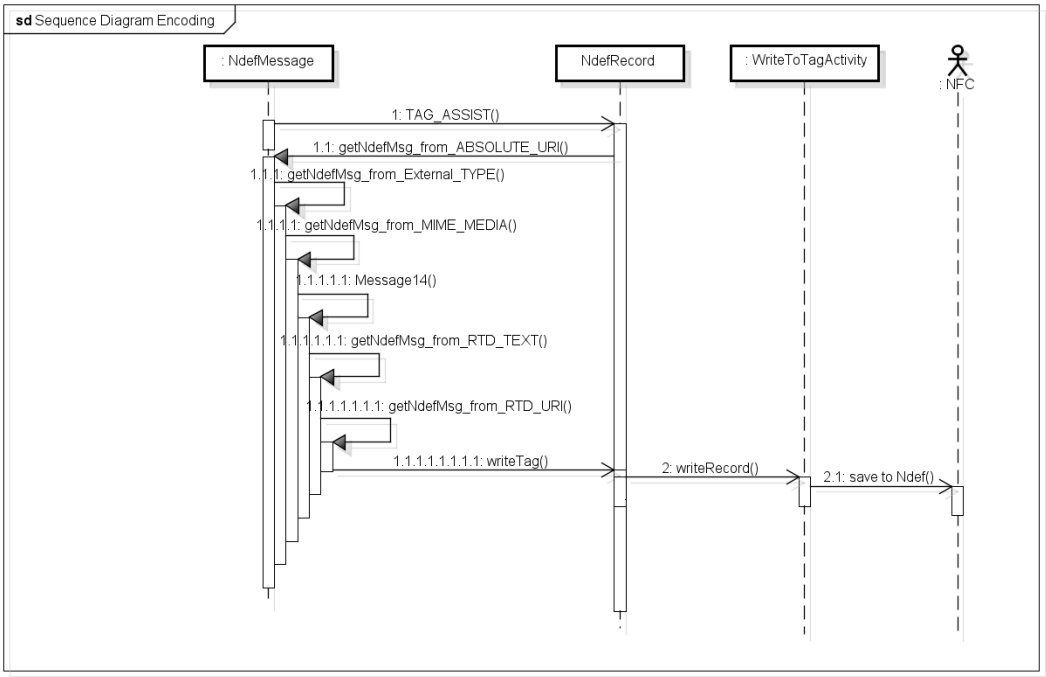
Gambar 3. 38 *Sequence* Diagram *Erase Tag*



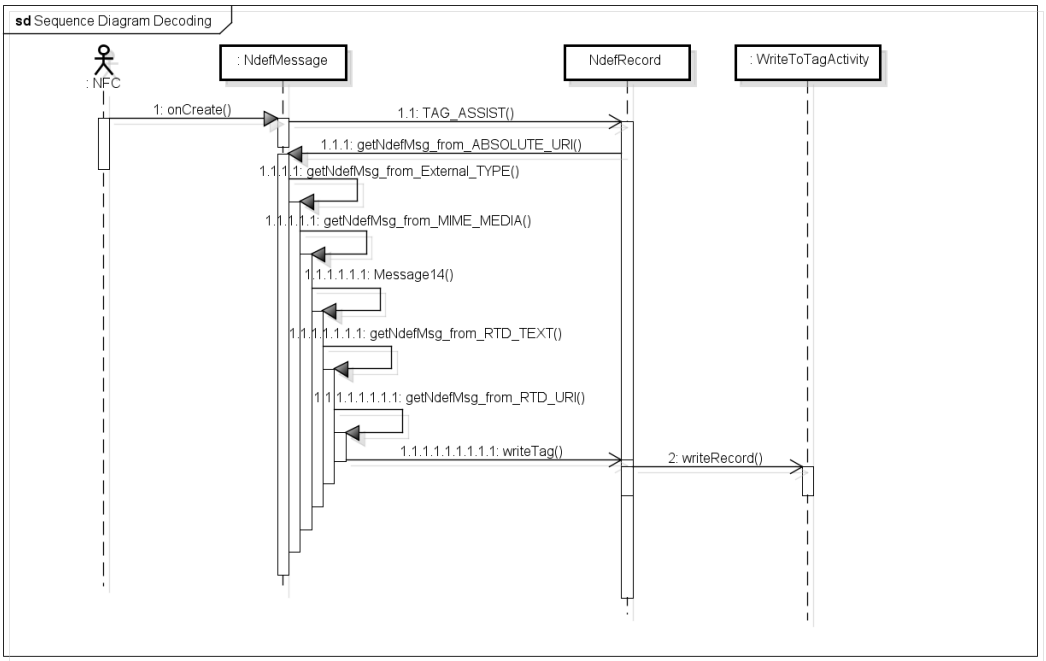
Gambar 3. 39 *Sequence* Diagram *Connection Link Control Protocol*



Gambar 3. 40 Sequence Diagram Convertion data



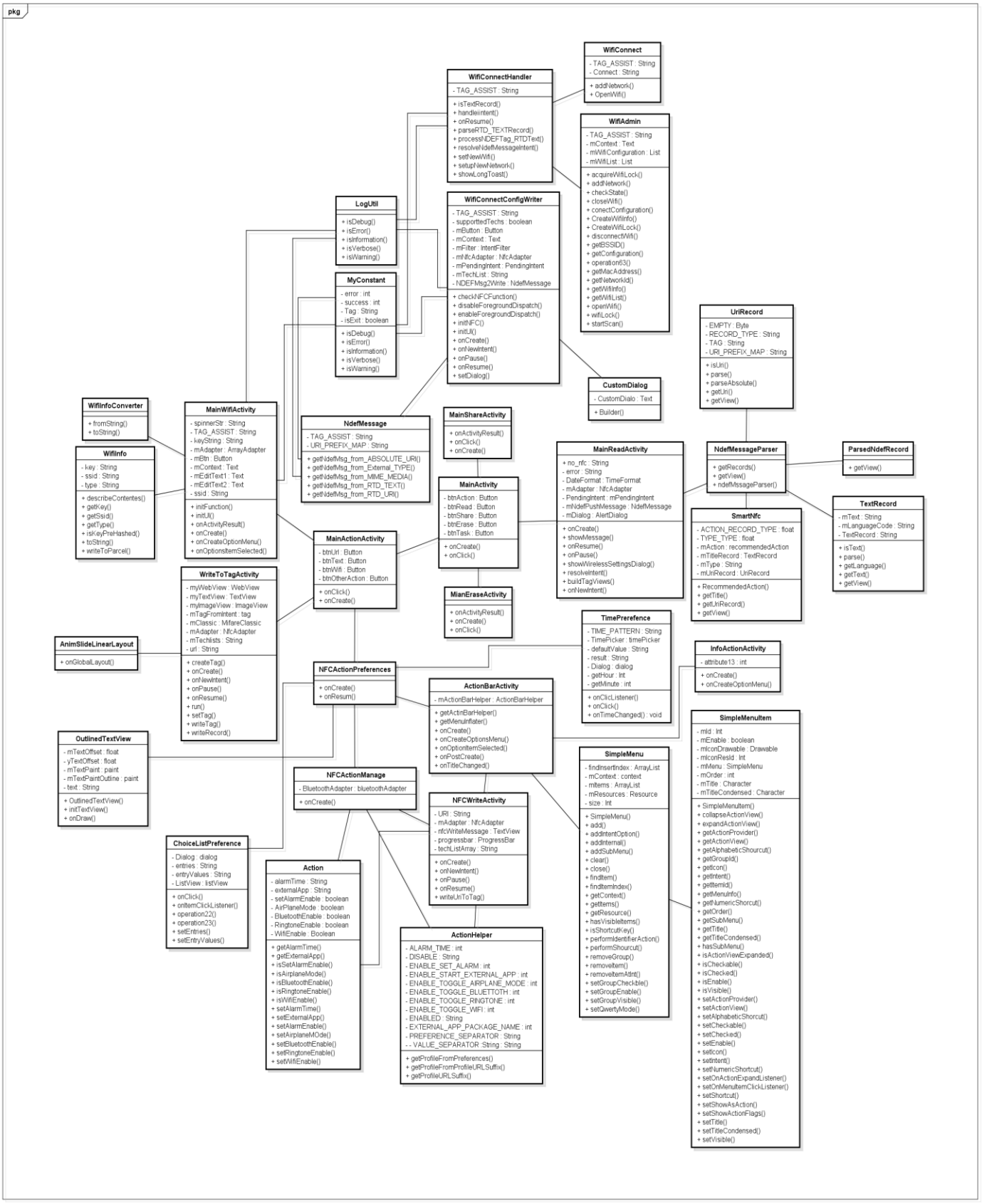
Gambar 3. 41 Squence Diagram *Encoding*



Gambar 3. 42 Squence Diagram *Decoding*

3.6.4 Class Diagram

Terdapat tiga jenis kelas dalam sistem, yaitu kelas *Boundary*, kelas *Control*, dan kelas *Entity*. Kelas *Boundary* merupakan kelas yang digunakan untuk membangun tampilan antarmuka sistem. Kelas *Control* merupakan kelas yang digunakan sebagai tempat terjadinya seluruh proses yang ada dalam sistem dan juga validator yang memeriksa validasi data yang masuk, dan kelas *Entity* merupakan kelas yang digunakan sebagai penampung data atau representasi data itu sendiri. Seluruh kelas-kelas yang ada dalam sistem dapat dilihat pada Gambar 3.43.



Gambar 3. 43 Class Diagram

Tabel 3. 14 Keterangan Class Diagram

No	Nama *Class*	Jenis *Class*	Keterangan
1	MainActivity	*Boundary*	Merupakan *class* yang digunakan untuk menampilkan halaman utama
2	MainActionActivity	*Boundary*	Merupakan *class* yang digunakan untuk menampilkan halaman *Action*
3	MainReadActivity	*Boundary*	Merupakan *class* yang digunakan untuk menampilkan halaman *Read*
4	MainShareActivity	*Boundary*	Merupakan *class* yang digunakan untuk menampilkan halaman *Share File*
5	MainEraseActivity	*Boundary*	Merupakan *class* yang digunakan untuk menampilkan halaman *Erase*
6	NFCActionPreferences	*Control*	Merupakan *class* yang berisi *method-method* untuk menangani proses *reader* pada NFC *tag*
7	NFCActionManage	*Control*	Merupakan *class* yang berisikan method-method untuk menangani proses yang ada pada menu *action*
8	InfoActionActivity	*Entity*	Merupakan *Class* untuk memproses info yang akan disampaikan oleh sistem
9	CustumDialog	*Entity*	Merupakan *Class* untuk menampilkan dialog pada sistem
10	NFCWriteAction	*Entity*	Merupakan *class* data yang digunakan untuk memproses segala pengaksesan *writer* pada NFC *Tag*
11	OutlinedTextView	*Entity*	Merupakan *class* data yang digunakan untuk mengatur tampilan pada menu action
12	TimePreference	*Entity*	Merupakan *class* data yang digunakan untuk memproses waktu *writer* sistem pada NFC *Tag*

13	ChoiceListPreference	*Entity*	Merupakan *class* data yang digunakan untuk memproses tampilan *list* pada menu *action*
14	ActionBarActivity	*Control*	Merupakan *class* data yang digunakan untuk memproses tampilan *bar* pada sistem
15	SimpleMenu	*Entity*	Merupakan *class* data yang digunakan untuk memproses tampilan menu *list* pada sistem
16	SimpleMenuItem	*Entity*	Merupakan *class* data yang digunakan untuk memproses tampilan menu sub*list* pada sistem
17	Action	*Entity*	Merupakan *class* data yang digunakan untuk memproses pengaksesan *Enabled* didalam menu *action*
18	ActionHelper	*Entity*	Merupakan *class* data yang digunakan untuk memproses pengaksesan seluruh aksi yang di inputkan pada menu *action*
19	NdefMessageParse	*Control*	Merupakan *class* data yang dugunakan untuk menyimpan dan membaca hasil data konversi dari NDEF
20	ParsedNdefRecord	*Entity*	Merupakan *class* yang digunakan untuk memproses penampilan hasil bacaan dari Ndef *message*
21	SmartNfc	*Control*	Merupakan class yang digunakan untuk memproses perintah-perintah pembacaan record pada NFC *tag*
22	TextRecord	*Entity*	Merupakan *class* yang digunakan untuk memproses konversi data pada NDEF
23	UriRecord	*Entity*	Merupakan *class* data yang digunakan untuk memproses penulisan *url* pada NFC *tag* yang di simpan pada NDEF Message
24	WifiInfo	*Entity*	Merupakan *class* data yang digunakan untuk memproses dan menyimpan pengaturan SSID, *Type* dan *Key* wifi yang telah di inputakan oleh

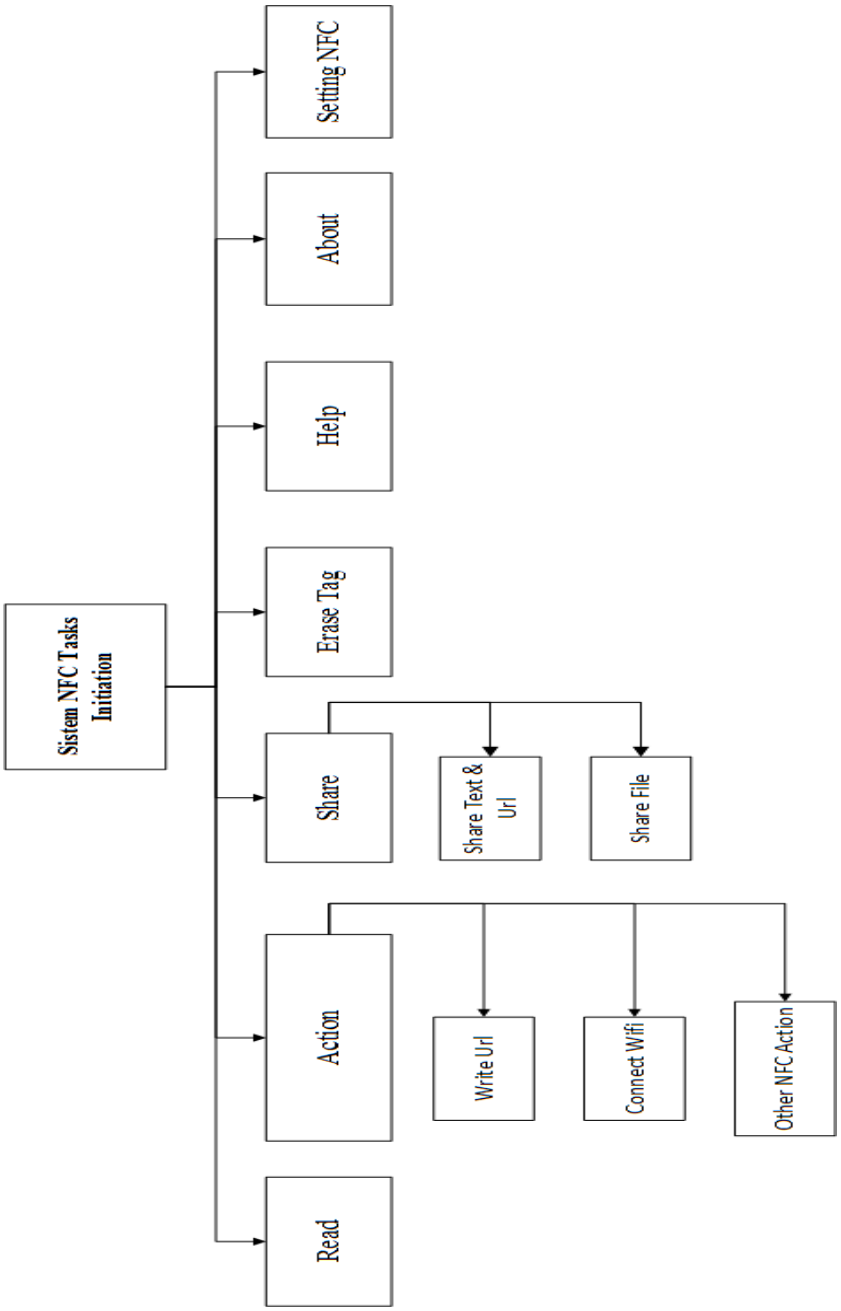
			pengguna
25	WifiInfoConverter	*Entity*	Merupakan *class* data yang digunakan untuk melakukan *converter* terhadap pengaturan wifi yang telah di inputkan oleh pengguna
26	WifiConnectConfigWriter	*Entity*	Merupakan *class* data yang digunakan untuk melakukan proses konfigurasi wifi dan melakukan *writer* pada *tag* NFC
27	WifiConnectHandler	*Control*	Merupakan class untuk memproses semua konfigurasi data pada koneksi wifi
28	NdefMessage	*Entity*	Merupakan *class* untuk memproses konversi data ke heksadesimal berupa data konfigurasi yang diinputan oleh pengguna
29	LogUtil	*Entity*	Merupakan *class* data yang digunakan untuk memberikan informasi kepada pengguna
31	MyConstant	*Control*	Merupakan *class* data yang berisikan *method-method* pemberitahuan
32	WifiAdmin	*Control*	Merupakan *class* data yang memproses semua pengaturan dalam menu *connect wifi*
33	WifiConnect	*Entity*	Merupakan *class* data yang berguna untuk memproses koneksi antara ponsel dan jaringan *wifi* yang SSID, *type* dan *key* nya di inputkan oleh penguuna
34	AnimSlideLinerarLayout	*Entity*	Merupakan *class* data yang berguna untuk memproses animasi *layout*
35	WriteToTagActivity	*Control*	Erupakan *class* data yang berguna untuk memproses *writer url* pada tag NFC

3.7 Perancangan Sistem

Perancangan sistem bertujuan untuk menspesifikasikan aspek-aspek teknik yang menjadi solusi dalam perencanaan. Pada tahap ini perancangan akan didefinisikan secara detail untuk mengatasi masalah-masalah yang lebih teknis, berkaitan dengan kegiatan implementasi seperti perancangan menu, perancangan antarmuka, dan jaringan semantik.

3.7.1 Perancangan Struktur Menu

Perancangan struktur menu merupakan gambaran alur pemakaian aplikasi. Perancangan struktur menu dari aplikasi ini dapat dilihat pada Gambar 3.44

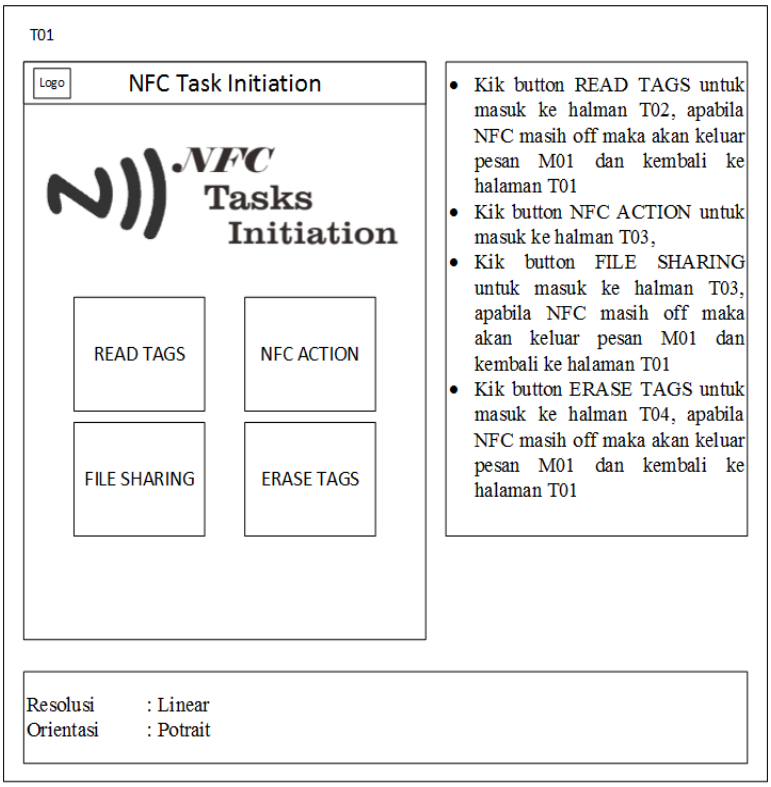


Gambar 3. 44 Struktur Menu

3.7.2 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga mempermudah dalam mengimplementasikan aplikasi serta memudahkan dalam pembuatan aplikasi.

1. Perancangan Antarmuka *Main Menu*



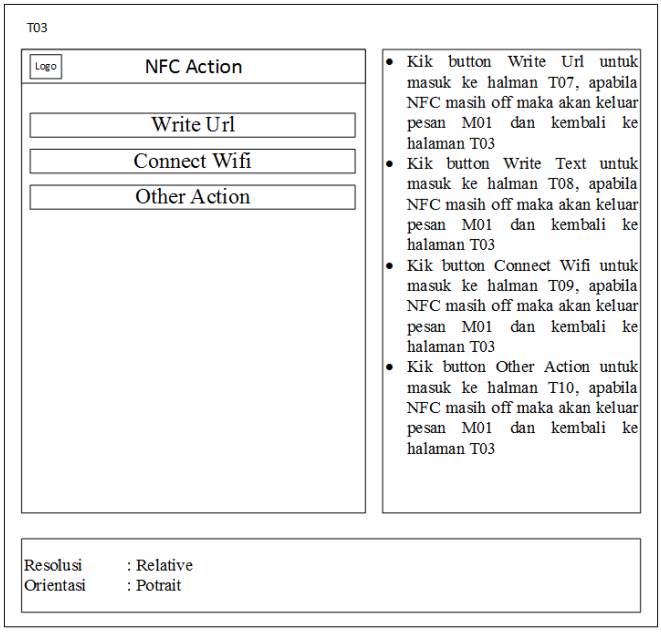
Gambar 3. 45 Antarmuka *Main Menu*

2. Perancangan Antarmuka *Read*



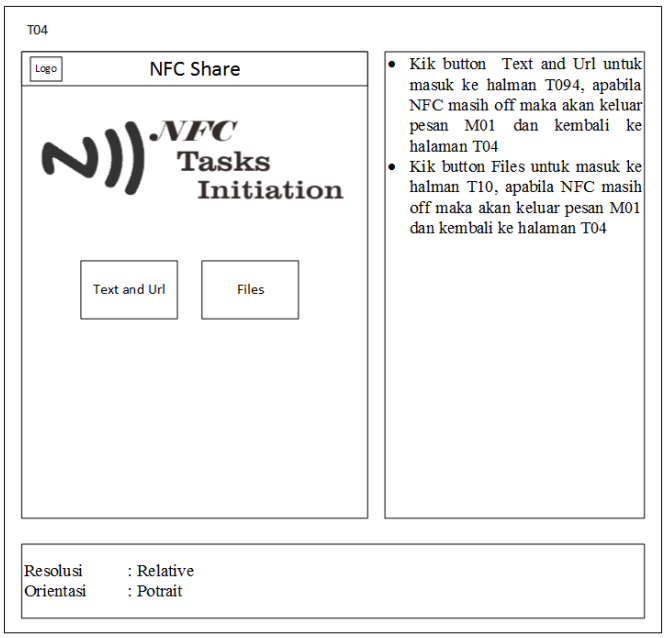
Gambar 3. 46 Antarmuka *Read*

3. Perancangan Antarmuka *NFC Action*



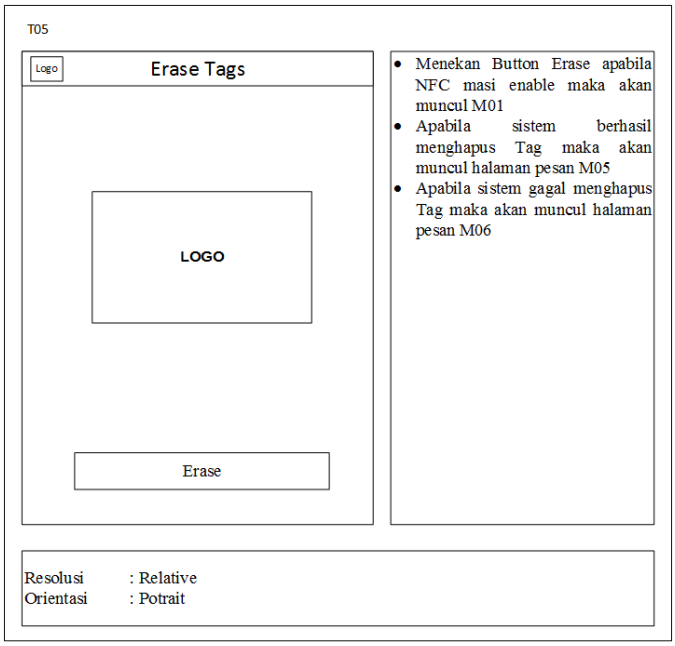
Gambar 3. 47 Antarmuka NFC *Action*

4. Perancangan Antarmuka *File Sharing*



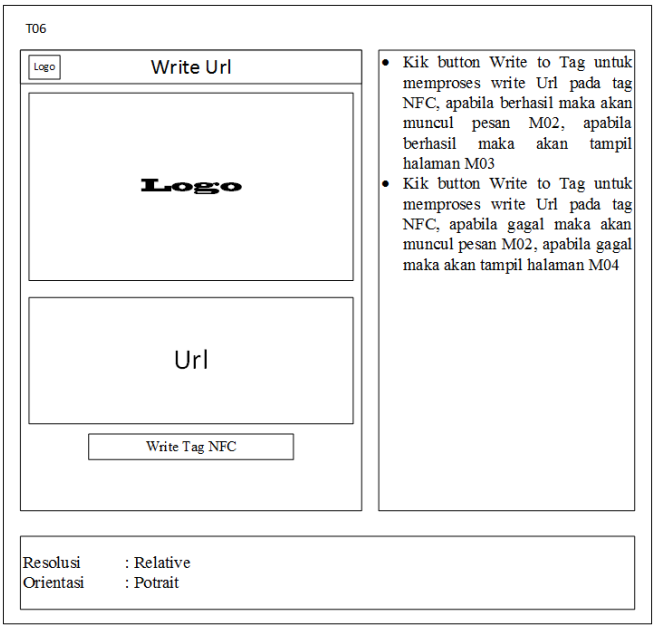
Gambar 3. 48 Antarmuka *File Sharing*

5. Perancangan Antarmuka *Erase Tags*



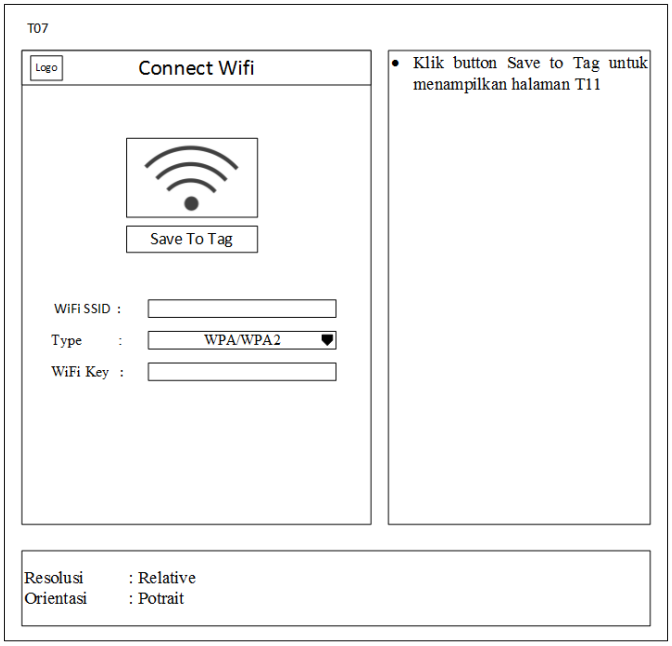
Gambar 3. 49 Antarmuka Erase Tags

6. Perancangan Antarmuka *Write Url*



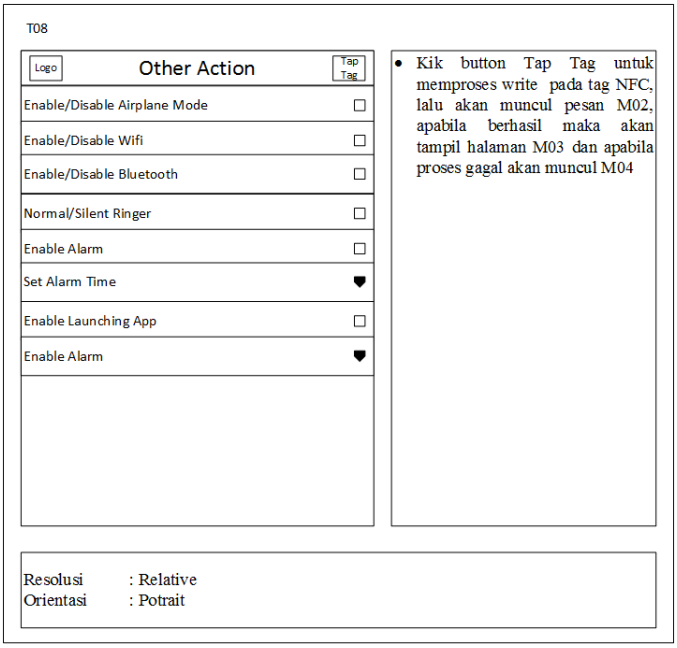
Gambar 3. 50 Antarmuka *Write Url*

7. Perancangan Antarmuka *Connect Wifi*



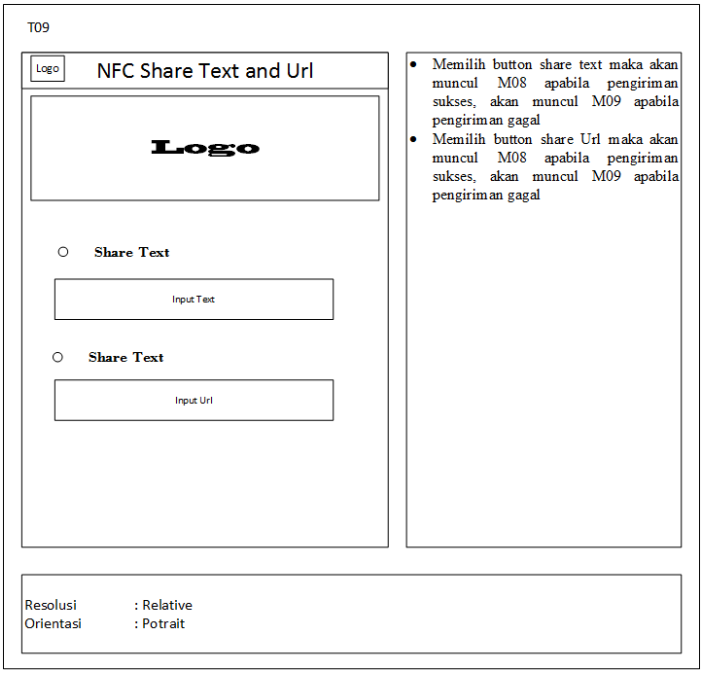
Gambar 3. 51 Antarmuka *Connect WiFi*

8. Perancangan Antarmuka *Other Action*



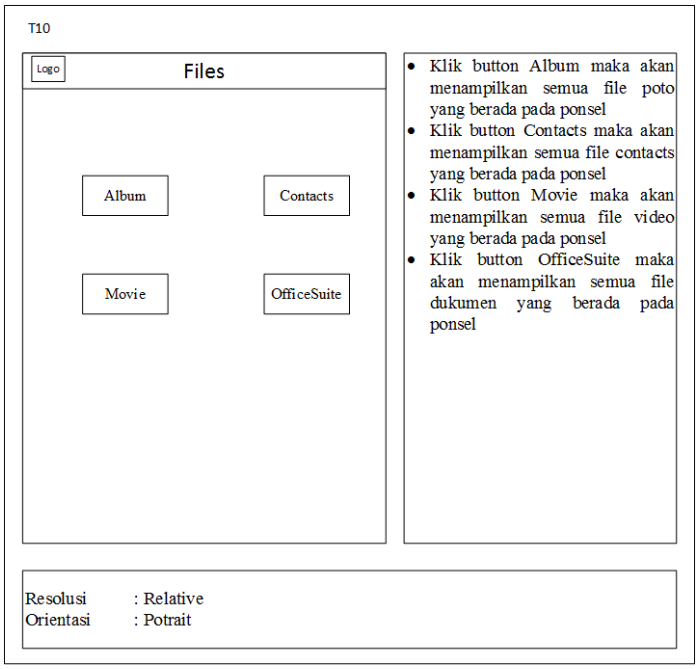
Gambar 3. 52 Antarmuka *Other Action*

9. Perancangan Antarmuka *Share Text and Url*



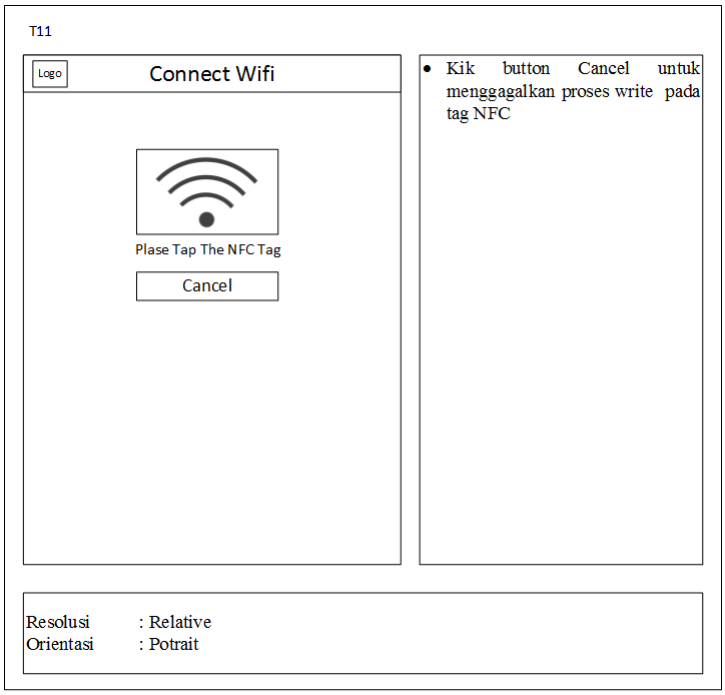
Gambar 3. 53 Antarmuka *Share Text and Url*

10. Perancangan Antarmuka *Share FIle*



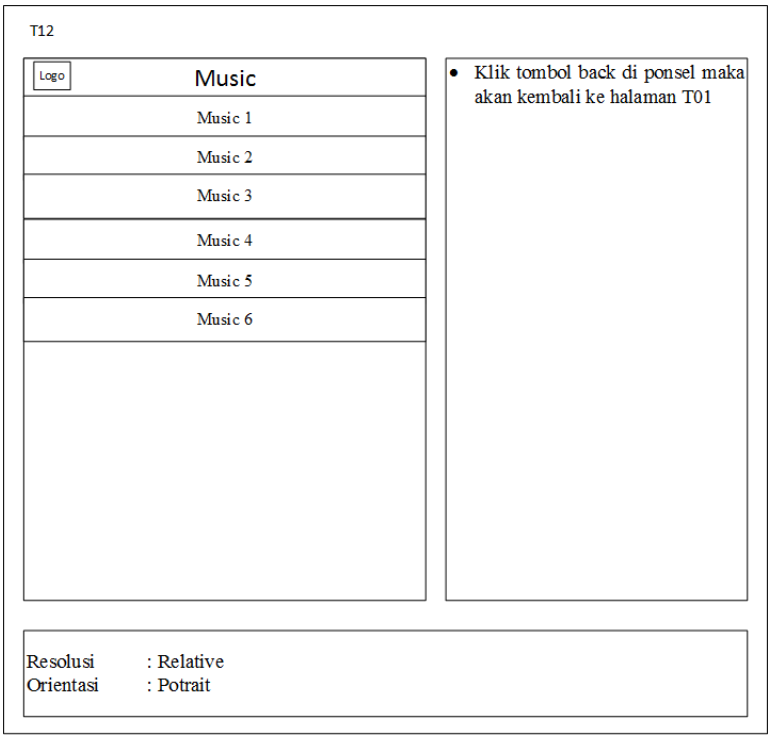
Gambar 3. 54 Antarmuka *File Share*

11. Perancangan Antarmuka *Tap WiFi*



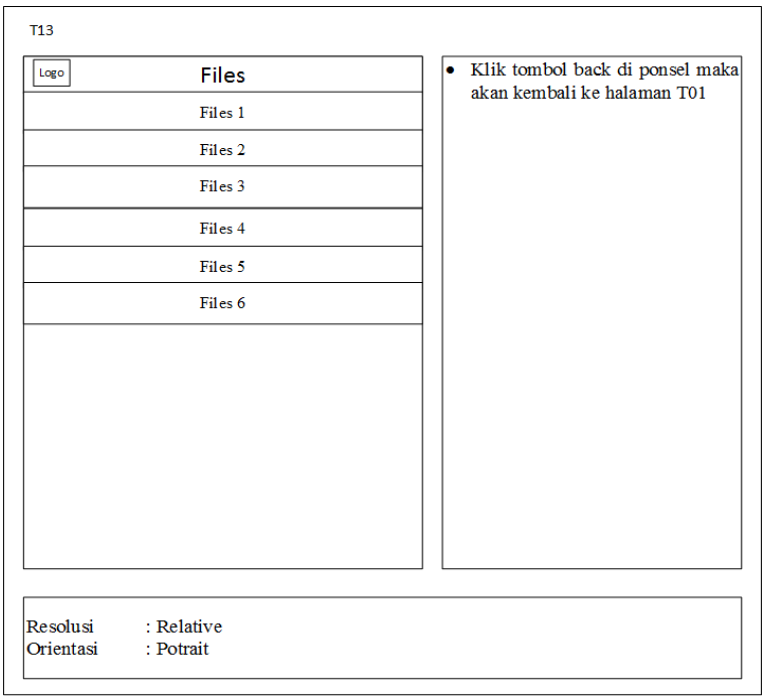
Gambar 3. 55 Antarmuka *Videos*

12. Perancangan Antarmuka *Music*



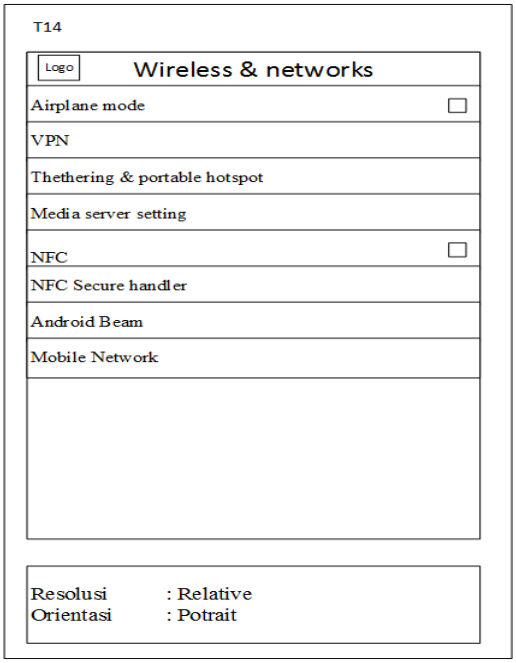
Gambar 3. 56 Antarmuka *Music*

13. Perancangan Antarmuka *Files*



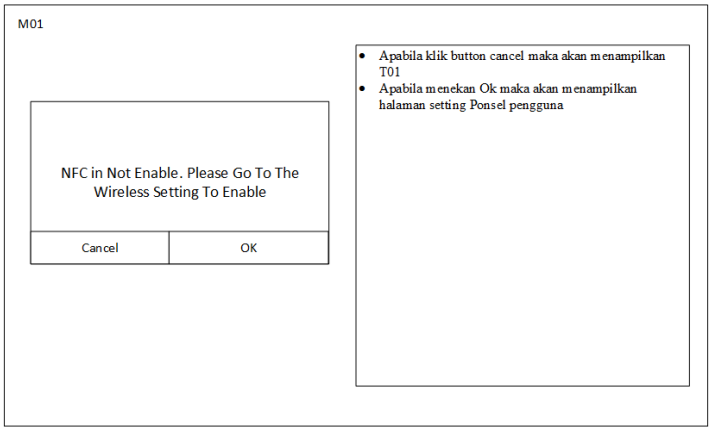
Gambar 3. 57 Antarmuka *Files*

14. Perancangan Antarmuka *Setting* NFC



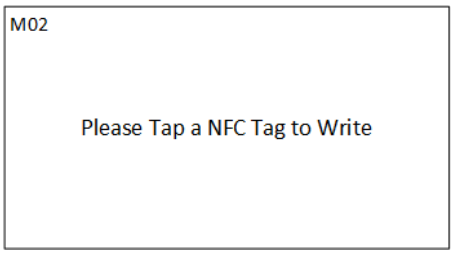
Gambar 3. 58 *Setting* NFC

15. Perancangan Antarmuka Pesan *NFC not Enable*



Gambar 3. 59 Antarmuka *NFC not Enable*

16. Perancangan Antarmuka Pesan *Tap a NFC Tag*



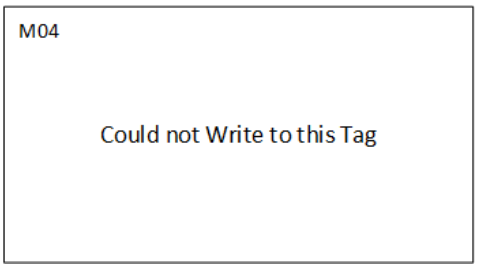
Gambar 3. 60 Antarmuka *Tap a NFC Tag*

17. Perancangan Antarmuka Pesan *Tag Written Successfull*



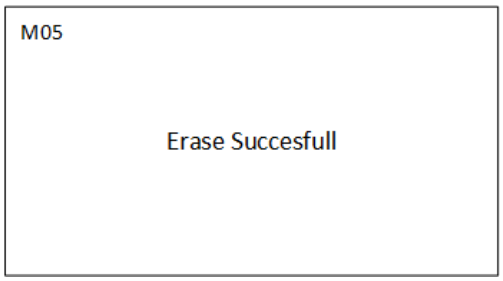
Gambar 3. 61 Antarmuka *Tag Written Successfull*

18. Perancangan Antarmuka Pesan *No Write*



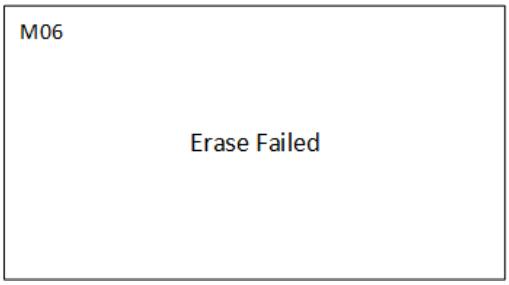
Gambar 3. 62 Antarmuka *No Write*

19. Perancangan Antarmuka Pesan *Erase Success*



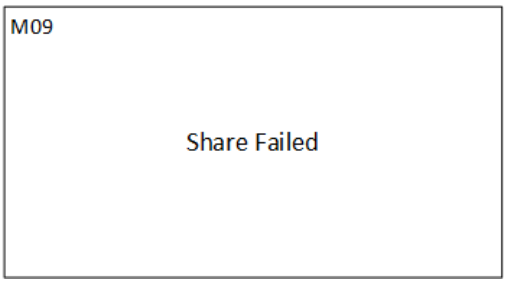
Gambar 3. 63 Antarmuka *Erase Success*

20. Perancangan Antarmuka Pesan *Erase Failed*



Gambar 3. 64 Antarmuka *Erase Failed*

21. Perancangan Antarmuka Pesan *Share Success*



Gambar 3. 65 Antarmuka *Share Success*

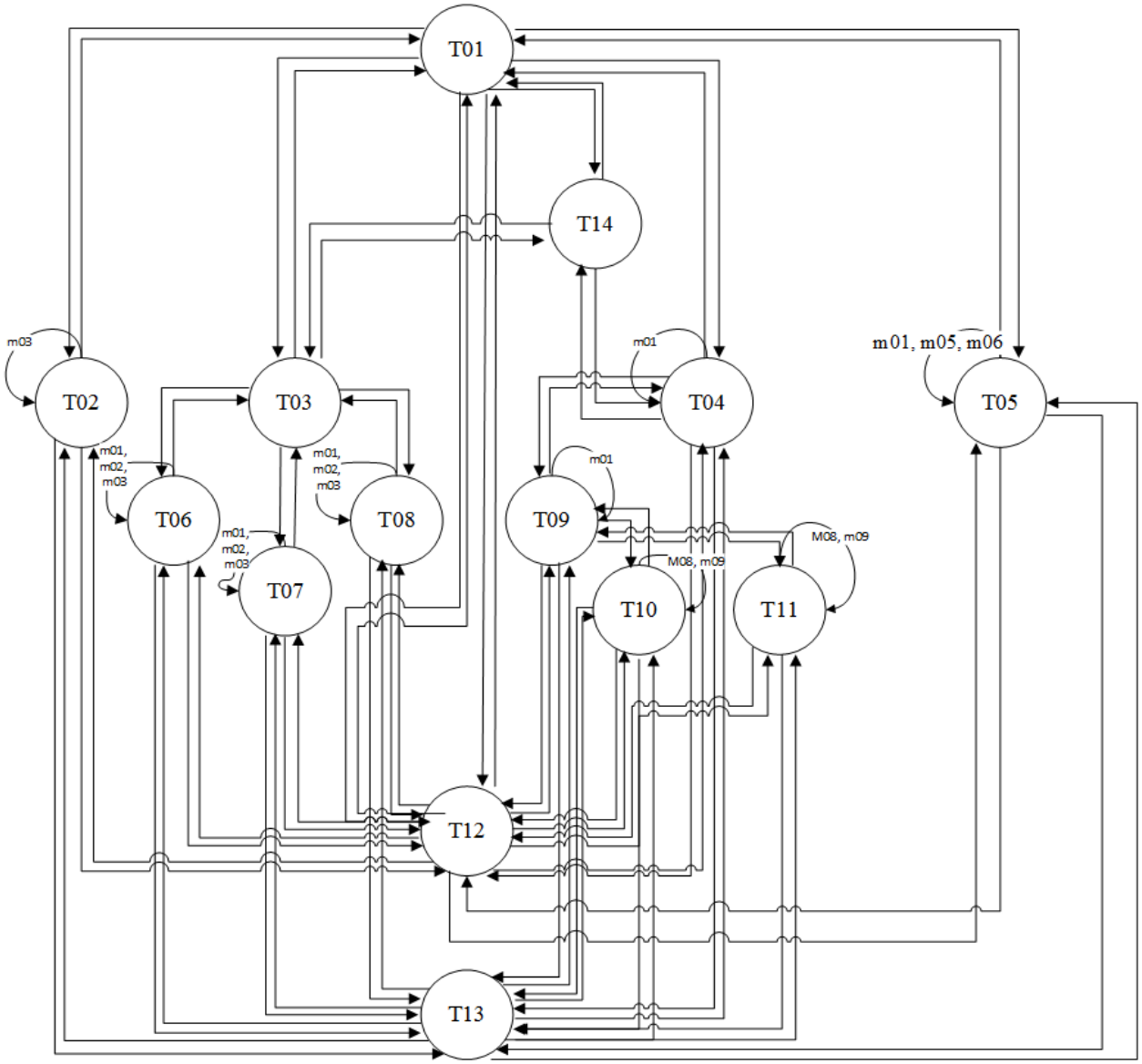
22. Perancangan Antarmuka Pesan *Share Failed*



Gambar 3. 66 Antarmuka *Share Success*

3.7.3 Jaringan Semantik

Jaringan semantik merupakan gambaran tentang relasi dari masing-masing halaman yang bisa diakses oleh pengguna. Jaringan semantik ini, digunakan sebagai panduan penggunaan menu-menu dalam sebuah aplikasi. Perancangan semantik dapat dilihat pada Gambar 3.63.



Gambar 3. 67 Jaringan Semantik