BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem bertujuan untuk mengidentifikasikan permasalahan-permasalahan yang terdapat pada sistem serta menentukan kebutuhan-kebutuhan dari sistem yang dibangun. Analisis tersebut meliputi analisis masalah, analisis aplikasi sejenis, analisis arsitektur sistem, analisis kebutuhan non fungsional dan analisis data.

Dalam membangun sebuah aplikasi yang sesuai dengan kebutuhan, tentu dibutuhkan analisis terhadap sistem yang sedang berjalan saat ini atau sistem yang berjalan secara umumnya. Tujuannya adalah agar sistem yang akan dibangun tidak keluar dari sistem inti yang sudah berjalan saat ini.

3.1.1 Analisis Masalah

Pembangunan aplikasi kartu nama ini menggunakan *smartphone android* yang merupakan suatu mekanisme yang dirancang sedemikian rupa yang digunakan untuk mempermudah para pelaku bisnis dan pemilik kartu nama. Adapun berdasarkan penelitian yang dilakukan terdapat permasalahan yang terjadi sehingga dibuatnya sistem ini adalah sebagai berikut :

1. Pengguna merasa kesulitan berkomunikasi dengan orang banyak karena mengeluarkan beban biaya lebih untuk tarif komunikasi. Hal ini menyebabkan tidak efektif dalam berkomunikasi.
2. Kesulitan mengatasi dalam mengingat untuk membawa kartu nama, karena umumnya orang tidak selalu mengingat kartu namanya sendiri. Hal seperti ini menjadi kendala jika melakukan pertukaran kartu nama.
3. Pengguna merasa kesulitan dalam mengelola kartu nama karena bentuk kartu nama berukuran kecil sehingga terkadang mengalami kehilangan. Hal ini menjadi tidak efisien jika kita ingin menghubungi seseroang.

4. Tidak adanya informasi pada kartu nama mengenai wajah pemilik kartu nama, karena umumnya kartu nama hanya memberikan informasi data pribadi. Hal ini menyebabkan sulitnya mengenal wajah seseorang.

3.1.2 Analisis Aplikasi Sejenis

Analisis aplikasi sejenis dimaksudkan untuk menganalisis fungsionalitas dan alur dari aplikasi yang telah ada pada domain yang sama dengan perangkat lunak pada penelitian ini. Hal ini bertujuan untuk memilah fungsionalitas dan alur yang akan diadopsi pada sistem yang diteliti berdasarkan manfaat yang sudah dihasilkan dari aplikasi yang dianalisis dan mencari keunikan dari perangkat lunak yang akan dibangun pada penelitian ini.

Pada penelitian ini terdapat dua aplikasi sejenis yang akan diteliti diantaranya adalah Profy dan Kwartu dengan pertimbangan kedua aplikasi tersebut berada pada *platform mobile android*. Paramater pembanding yang digunakan mengacu kepada komponen-komponen sistem informasi.

1. Proffy

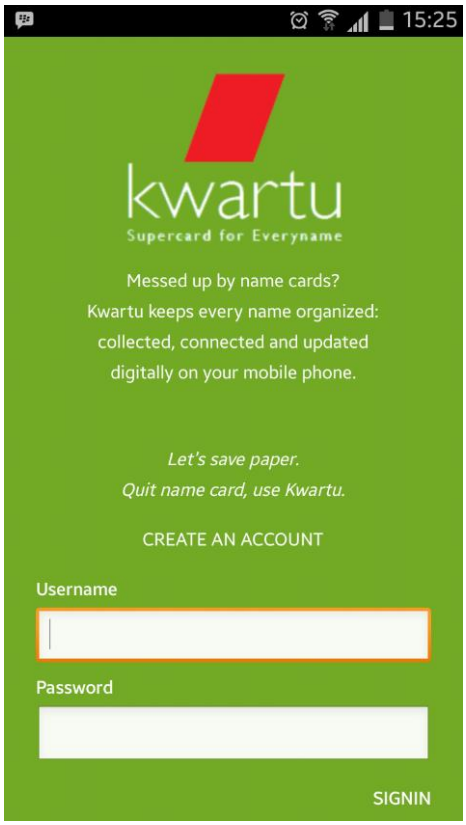
   Proffy adalah aplikasi *mobile* yang diciptakan dengan berfokus pada *kartu nama*. Penggunaan aplikasi ini untuk pertukaran kartu nama dengan pengguna lain dan mengatur kartu nama. Berikut adalah gambar *screenshot* aplikasi proffy.



Gambar 3.1 *Screenshoot* Aplikasi Proffy

2. Kwartu

Kwartu adalah aplikasi *mobile* yang diciptakan dengan berfokus pada *design* kartu nama. Penggunaan aplikasi ini bersifat men*design* kartu nama bagi pengguna yang belum mempunyai kartu nama. Berikut adalah gambar *screenshot* aplikasi kwartu.



Gambar 3.2 *Screenshot* Aplikasi Kwartu

Berdasarkan fungsi pada masing-masing aplikasi tersebut maka dapat dilakukan analisis perbandingan dari kedua aplikasi diatas dengan menggunakan parameter komponen sistem informasi. Berikut adalah Tabel 3.1 Analisis perbandingan aplikasi sejenis.

Tabel 3.1 Analisis Perbandingan Aplikasi Sejenis

Komponen Pembanding	**Proffy**	**Kwartu**
Input	Masukan berupa data pribadi identitas diri sendiri dan *profile* kartu nama.	Masukan berupa data *user*name dan *password* dan men*design* kartu nama
Proses	Data masukan diproses menjadi beberapa fungsionalitas seperti *NFC, QrCode , search*	Data masukan diproses hanya untuk *login* pada akun aplikasi tersebut
Output	Data kontak kartu nama berupa foto *profile* dan identitas pribadi. list Kontak di atur sesuai abjad	Hasil *design* kartu bisa di simpan di dalam aplikasi.Informasi kartu nama berupa data pribadi
Teknologi	Teknologi NFC, *QrCode* untuk mendukung pertukaran kartu nama	Teknologi NFC untuk mendukung berbagi kartu nama
Software	Aplikasi ini menggunakan sistem operasi *android* dengan minimal version 4.1 Jelly Bean	Aplikasi ini menggunakan sistem operasi *android* dengan minimal version 4.1 Jelly Bean
Hardware	Aplikasi ini menggunakan beberapa perangkat keras yang sudah terpasang default disetip perangkat *android* diantaranya: Kamera, GPS, NFC	Aplikasi ini menggunakan beberapa perangkat keras yang sudah terpasang default disetip perangkat *android* diantaranya: NFC
Data	menggunakan penyimpanan internal dan external pada perangkat *mobile*.	menggunakan penyimpanan internal dan external pada perangkat *mobile*.
Komponen Kendali	Aplikasi ini akan menampilkan warning untuk	Aplikasi ini akan menampilkan warning

Komponen Pembanding	**Proffy**	**Kwartu**
	melakukan update jika pengguna yang menggunakan aplikasi ini tidak melakukan update secara berkala.	untuk melakukan update jika pengguna yang menggunakan aplikasi ini tidak melakukan update secara berkala.
Kelebihan	Aplikasi ini mendukung penggunaan nearby dan *login* menggunakan via linkedn	Aplikasi ini mampu men*design* kartu nama keinginan kita sendiri.
Kekurangan	Aplikasi ini tidak mendukung fitur *chatting* dan hanya berupa icon sms untuk diteruskan ke menu sms	Aplikasi ini tidak mendukung *QrCode* dan tidak mendukung fitur *chatting*

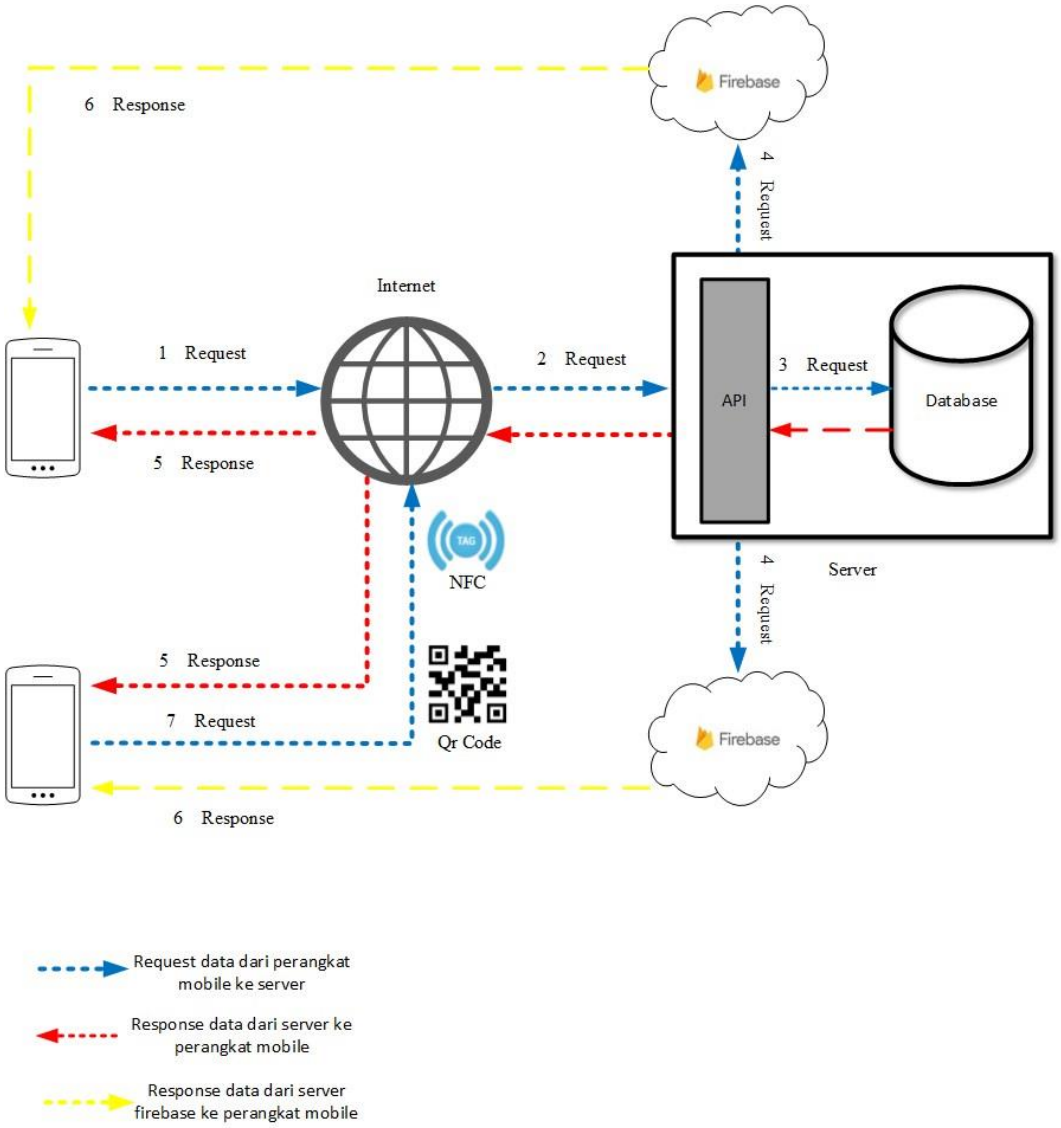
Dari hasil analisis kedua aplikasi sejenis, aplikasi yang akan dibangun akan memiliki beberapa fungsionalitas diantaranya:

1. *Chatting* untuk berkomunikasi langsung dalam aplikasi.
2. Menampilkan informasi alamat secara otomatis saat sedang melakukan pertukaran kartu nama

Berdasarkan analisis tersebut juga dapat disimpulkan keunikan perangkat lunak yang dibangun dibandingkan dengan aplikasi sejenis yaitu adanya fitur *chatting* langsung dalam aplikasi dan memberikan informasi alamat secara otomatis saat sedang melakukan pertukaran kartu nama.

3.1.3 Analisis Arsitektur Sistem

Analisis Arsitektur Sistem merupakan sistem yang akan dibangun, aplikasi Kartu nama yang berbasis internet ini akan berkomunikasi oleh web *service* dengan menggunakan pertukaran bahasa JSON, adapun rancanganan sistem yang akan dibangun dapat dilihat di gambar berikut.



Gambar 3.3 Arsitektur Sistem Perangkat Lunak

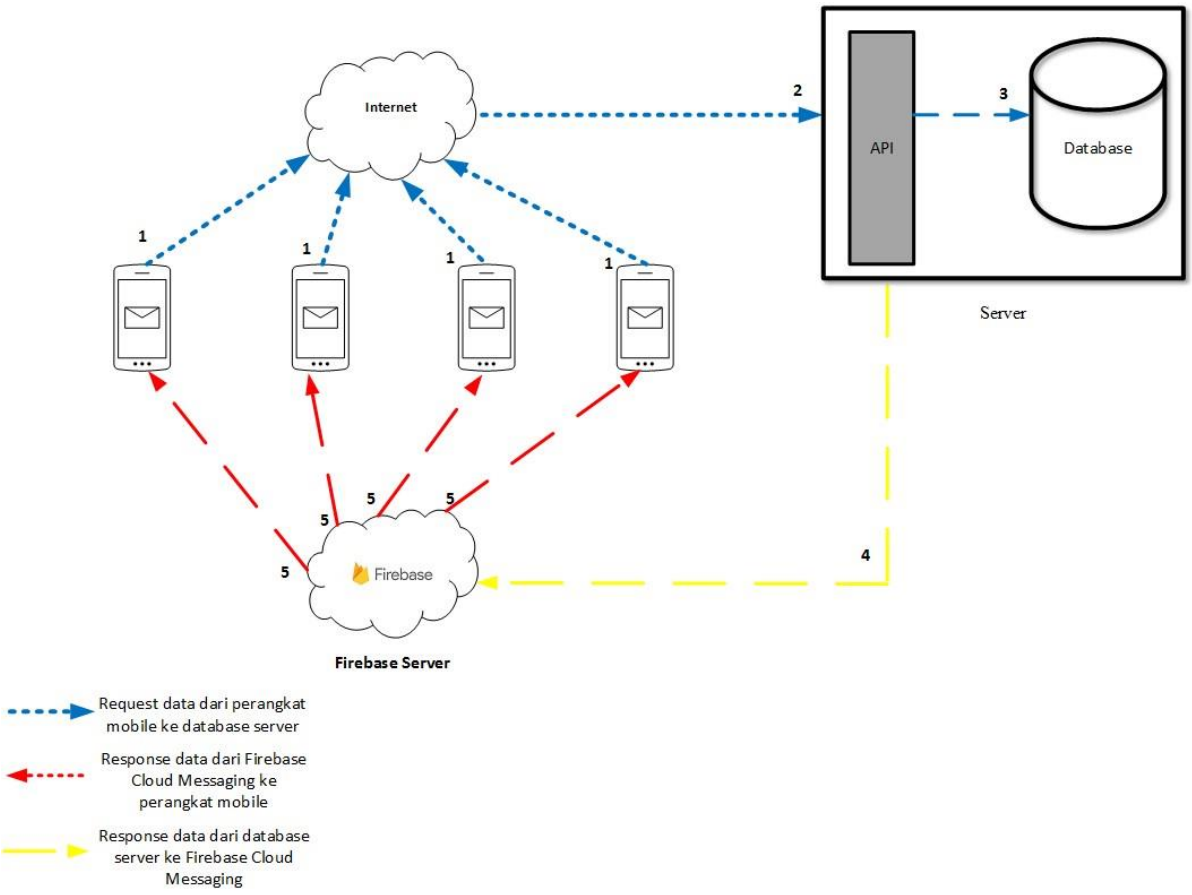
Berikut adalah deskripsi dari gambar 3.3 Arsitektur Sistem Perangkat lunak:

1. Perangkat *mobile* pengguna melakukan *request* data menggunakan jaringan internet ke *server* melalui API. Pengguna bisa melakukan pendaftaran akun dan menyimpan data pribadi, gambar kartu nama, menambah kontak pertemanan serta berinteraksi secara langsung di dalam aplikasi.

2. *Server* menerima *request* data dari *server* dan menentukan jenis *request* yang diminta. *Server* akan mengecek jenis *request* yang disampaikan. Jenis *request* berupa data text, data gambar dan data kirim pesan.
3. Jika *server* menerima permintaan data text dan data gambar maka *server* akan langsung menyimpan atau mengambil data yang ada di dalam database. Database akan mengecek ketersedian data text dan data gambar milik pengguna. Bila tidak tersedia, database akan memberikan notifikasi ke *server* dan diteruskan ke pengguna.
4. Perangkat *mobile* pengguna melakukan *request* data pesan menggunakan jaringan internet ke *server* firebase melalui API. Pengguna melakukan kirim pesan kepada pengguna lain dimana pesan tersebut akan dikirim ke *server* firebase. Firebase adalah layanan DbaaS (Database as a Service) dengan konsep realtime. Firebase merupakan penyedia layanan cloud dengan backend sebagai servis
5. Setelah *server* menerima data yang diminta berupa data text dan data gambar, maka data tersebut akan dikembalikan dalam bentuk JSON untuk diproses perangkat *mobile* pengguna. Pengguna dapat melakukan modifikasi data pribadi dan dapat melakukan pergantian data gambar.
6. Setelah *server* firebase menerima data pesan yang diminta, maka data tersebut akan dikembalikan dalam bentuk JSON untuk diproses dan menampilkan notifikasi terima pesan ke perangkat *mobile* pengguna.
7. Perangkat *mobile* melakukan pertukaran kartu nama dengan menggunakan fitur NFC atau *QrCode*. Pengguna dapat melakukan 2 cara untuk dapat melakukan pertukaran kartu nama yaitu dengan NFC atau *QrCode*. Jika 2 pengguna mempunyai fitur NFC pada *smartphone*nya maka pengguna bisa menggunakan fitur NFC pada aplikasi dan jika tidak mempunyai fitur NFC, pengguna dapat menggunakan fitur *QrCode* didalam aplikasi. Inputan didalam NFC dan *QrCode* berupa Id *User* pengguna aplikasi dan output yang dihasilkan berupa penambahan data kontak, memberikan informasi data pribadi pengguna lainnya dan dapat melakukan interaksi secara langsung ke pengguna lainnya.

3.1.4 Analisis *Chatting*

Platform mobile adalah salah satu subsistem yang dipilih untuk pembangunan *chatting* dari perangkat lunak ini. Arsitektur *chatting* perangkat lunak pada *platform mobile* mengambarkan bagaimana perangkat lunak saling berinteraksi seperti diilustrasikan pada gambar 3.4



Gambar 3.4 Arsitektur Sistem *Chatting*

Berikut adalah deskripsi dari gambar 3.4 Arsitektur *Chatting* pada Sistem Perangkat lunak:

1. Perangkat *mobile* pengguna melakukan *request* data kirim pesan menggunakan jaringan internet ke database *server* dan firebase *server* melalui API. Pengguna melakukan kirim pesan kepada pengguna lain dimana pesan tersebut akan dikirim ke *server* firebase.
2. *Server* menerima data kirim pesan yang diminta, data kirim pesan berupa teks dan data tersebut diteruskan ke dalam database.

3. Database menerima data pesan dari *server*. Data pesan tersimpan didalam database ke tabel *message*s
4. API me*response* data kirim pesan tersebut dan diteruskan ke *server* firebase cloud messaging.
5. *Server* Firebase Cloud Messaging meneruskan pesan ke pengguna perangkat *mobile* lainnya dan Perangkat *mobile* mendapatkan notifikasi terima pesan.

Firebase adalah layanan DbaaS (*Database as a Service*) dengan konsep realtime. Firebase merupakan penyedia layanan cloud dengan backend sebagai servis yang berbasis di San Fransisco, California yang sekarang dimiliki oleh *google*. Firebase terdiri dari fitur pelengkap yang bisa dipadukan sesuai dengan kebutuhan.

3.1.5 Analisis NFC

Sistem yang akan dibangun adalah aplikasi untuk pertukaran kartu nama dengan menggunakan teknologi NFC pada fitur *Smartphone* yang di miliki oleh *user*. Sistem pertukaran kartu nama yang dilakukan yaitu dengan menghidupkan fitur NFC pada *smartphone* kedua *user* dan mendekatkan kedua *smartphone* tersebut

Proses penyampaian informasi dilakukan menggunakan *Tag* NFC dimana pengguna akan melakukan write and read dengan aplikasi yang akan dibangun. Informasi yang disediakan yaitu berupa data kartu nama. Inputan didalam *tag* NFC berupa id *user* yang akan dimasukkan kedalam NDEF *Record* dan output yang dihasilkan adalah bertambah data kontak pertemanan, mendapatkan informasi langsung alamat saat pertukaran kartu nama, dapat melihat informasi data pribadi pengguna lainnya, dan dapat melakukan interaksi secara langsung didalam aplikasi. Berikut Gambar Prosedur Pertukaran Kartu Nama.



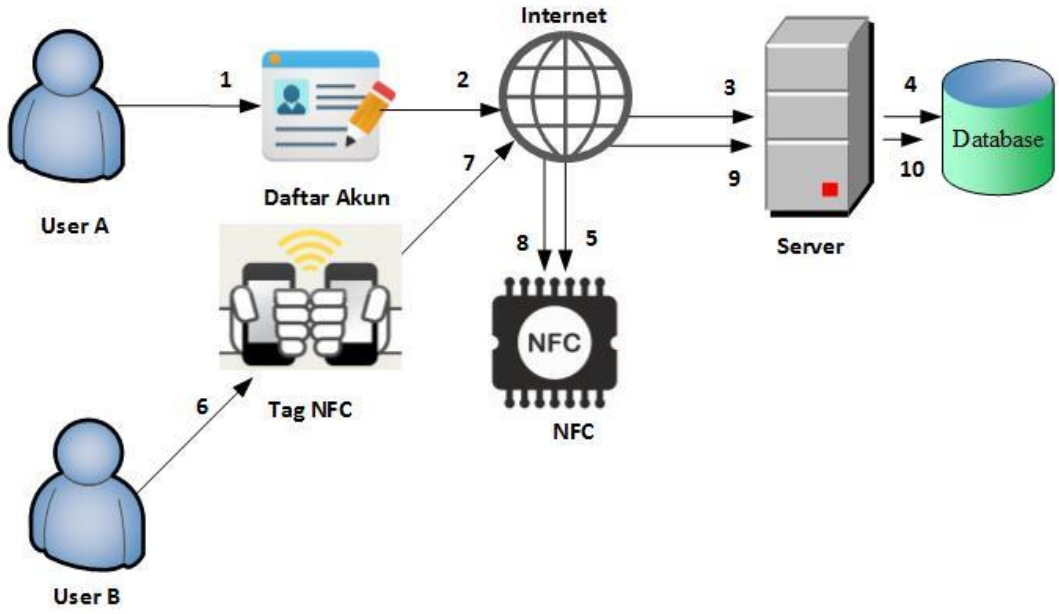
Gambar 3.5 Prosedur Pertukaran Kartu Nama dengan NFC

Berikut adalah deskripsi dari gambar 3.5 Gambar Pertukaran Kartu Nama dengan NFC sebagai berikut:

1. Pengguna dengan pengguna lainnya bersiap untuk mengoperasikan aplikasi dan aplikasi akan memberikan notifikasi untuk menghidupkan fitur NFC pada *smartphone* kedua pengguna.
2. Pengguna saling mendekatkan *smartphone* dengan pengguna lainnya untuk proses pembacaan NFC. Ilustrasi mekanisme pembacaan, pengguna pertama bertindak sebagai read untuk membaca id *user* pengguna kedua didalam NFC. Pengguna kedua mendekatkan smartphonnya dengan pengguna pertama.
3. Setelah selesai terbaca, pengguna akan mendapatkan notifikasi berhasil dalam melakukan pertukaran kartu nama lalu pengguna akan mendapatkan informasi alamat secara langsung lokasi pertukaran kartu nama, pengguna dapat melihat informasi data pribadi pengguna lainnya dan siap untuk saling berinteraksi secara langsung.

3.1.5.1 Deskripsi Umum Baca dan Tulis *Tag* NFC

Sistem write dan reader digunakan oleh untuk pengguna aplikasi, sistem write untuk menginputkan data *user* ke dalam *tag* NFC pada aplikasi. Sistem reader digunakan untuk melakukan pertukaran kartu nama atau menambah kontak pertemanan didalam aplikasi Berikut ini adalah deskripsi sistem yang akan dibangun di tunjukkan pada Gambar 3.6



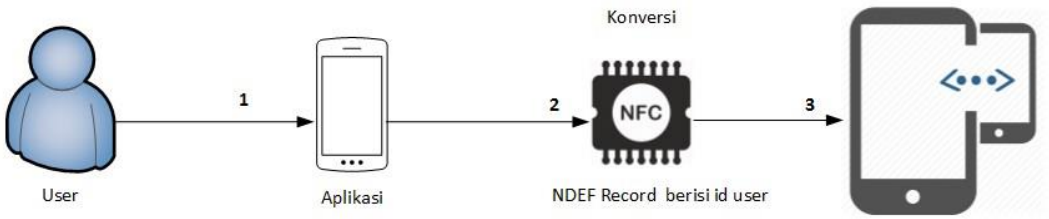
Gambar 3.6 Deskripsi Umum Baca dan Tulis *Tag* NFC

Berikut adalah deskripsi dari gambar 3.6 Deskripsi Umum Baca dan Tulis *Tag* NFC:

1. Pengguna A melakukan pendaftaran akun pada aplikasi. Pendaftaran berisi *email*, nama depan , nama belakang dan *password*.
2. Data yang sudah didaftarkan akan diteruskan ke *server* dan disimpan didalam database melalui jaringan internet.
3. *Server* menerima data yang akan diteruskan ke dalam database.
4. Data telah tersimpan didalam database
5. Data yang sudah tersimpan akan dikonversi ke NDEF *Record* yang berisi Id *User* A.
6. *User* B bersiap melakukan mendekatkan *smartphone*nya didalam aplikasi.
7. *User* B dan *user* A saling mendekatkan *smartphone*nya menggunakan jaringan internet.
8. Proses berhasil dan hasilnya akan di konversi menjadi id kontak pada tabel kontak.
9. *Server* menerima data yang akan diteruskan ke dalam database.
10. Data telah tersimpan didalam database.

3.1.5.2 Analisis Mekanisme Penulisan NFC

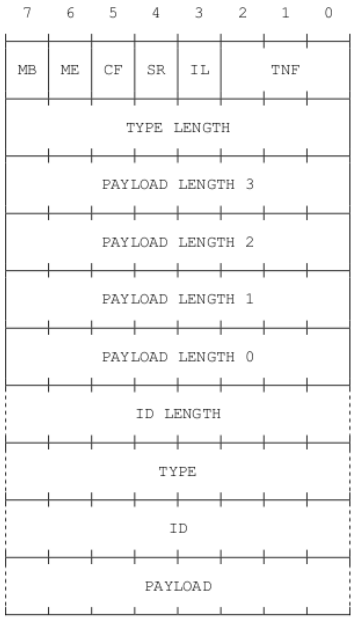
Mekanisme proses Pembacaan NFC melalui perangkat *mobile android* ditunjukan pada gambar 3.7



Gambar 3.7 Mekanisme Penulisan NFC

Berikut adalah proses alur pembacaan NFC:

1. *User* melakukan pendaftaran di aplikasi dan mendapatkan id *user* dengan contoh “1”
2. Sistem mengkonversi id *user* ke dalam bilangan heksadesimal.
3. Sistem akan mengkonversi id *user* kedalam NDEF *record*. Berikut adalah format NDEF *record*.



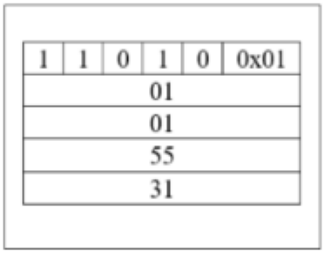
Gambar 3.8 NDEF *Record*

Proses konversi id member kedalam NDEF *record* adalah sebagai berikut:

a) Variabel MB (*Message Begin*) : variabel ini merupakan awal dari NDEF *Message* dan akan bersi nilai 1 sesuai dengan yang telah ditentukan oleh Forum NFC.

b) Variabel ME (*Message End*) : variabel ini merupakan akhir dari NDEF *Message* dan akan selalu berisi nilai 1 karena nilai dalam variabel ini telah ditentukan oleh Forum NFC.
c) Variabel CF (*Chunk Flag*) : variabel ini berfungsi untuk mengidentifikasi potongan *record* dan akan berisi nilai 0 karena nilai dalam variabel ini sudah ditentukan oleh Forum NFC.
d) Variabel SR (*Short Record*) : variabel ini berfungsi untuk menentukan payload (muatan data) yang berukuran kecil dan akan berisi nilai 1 sesuai dengan yang telah ditentuakn oleh Forum NFC.
e) Variabel IL (*Identification Length*) : variabel ini diisi 0 karena tidak akan di set.
f) Variabel TNF (*Type Name Format*) : variabel ini akan diisi oleh nilai 0x01. Nilai tersebut didapatkan dari jenis TNF NFC Forum well-known type [NFC RTD].
g) Type Length : berisi bilangan heksadesimal 01 yang didapatkan dari jenis TNF NFC Forum *well-known type* [NFC RTD].
h) *Payload length* : berisi bilangan heksadesimal 01 yang didapatkan dari panjang *payload* = 1.
i) *ID Length* : variabel ini diabaikan karena IL diisi 0.
j) *Type* : berisi bilangan heksadesimal 55 yang didapatkan dari nilai “U” yang merupakan standar dari Forum NFC.
k) *ID* : variabel ini diabaikan karena IL diisi 0.
l) *Payload* : berisi bilangan heksadesimal 31 yang didaptkan dari nilai *payload* = “1”.

Sehingga NDEF *record* ditunjukan pada Gambar 3.9



Gambar 3.9 NDEF *Record* Id *User* = “1”

Berikut contoh *Source Code* dalam proses menulis NFC pada aplikasi kuartu.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_nfc);
    mToolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(mToolbar);
    actionBar = getSupportActionBar();
    if (actionBar != null) {
        actionBar.setTitle("Via NFC");
        actionBar.setDisplayHomeAsUpEnabled(true);
    }
    aq = new AQuery(this);
    session = new SessionManager(getApplicationContext());
    HashMap<String, String> user = session.getUserDetails();
    id_user = user.get(SessionManager.KEY_IDUSER);
    email = user.get(SessionManager.KEY_EMAIL);
    nama_depan = user.get(SessionManager.KEY_NAMADEPAN);
    nama_belakang = user.get(SessionManager.KEY_NAMABELAKANG);
    foto_kartu = user.get(SessionManager.KEY_FOTOKARTU);
    GPSTracker tracker = new GPSTracker(this);
    if (!tracker.canGetLocation()) {
        tracker.showSettingsAlert();
    } else {
        latitude = tracker.getLatitude();
        longitude = tracker.getLongitude();
        Geocoder geocoder = new Geocoder(this, Locale.getDefault());
        List<Address> addresses = null;
        try {
            addresses = geocoder.getFromLocation(latitude, longitude, 1);
            alamat_event = addresses.get(0).getAddressLine(0) + " " + addresses.get(0).getLocality();
            //aq.id(R.id.edAlamat).text(alamat_tempat);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    aq.id(R.id.imgKartuNama).image("http://www.kuartu.web.id/kuartu_api/gambar/" + foto_kartu, true, true, 400, 0);
```

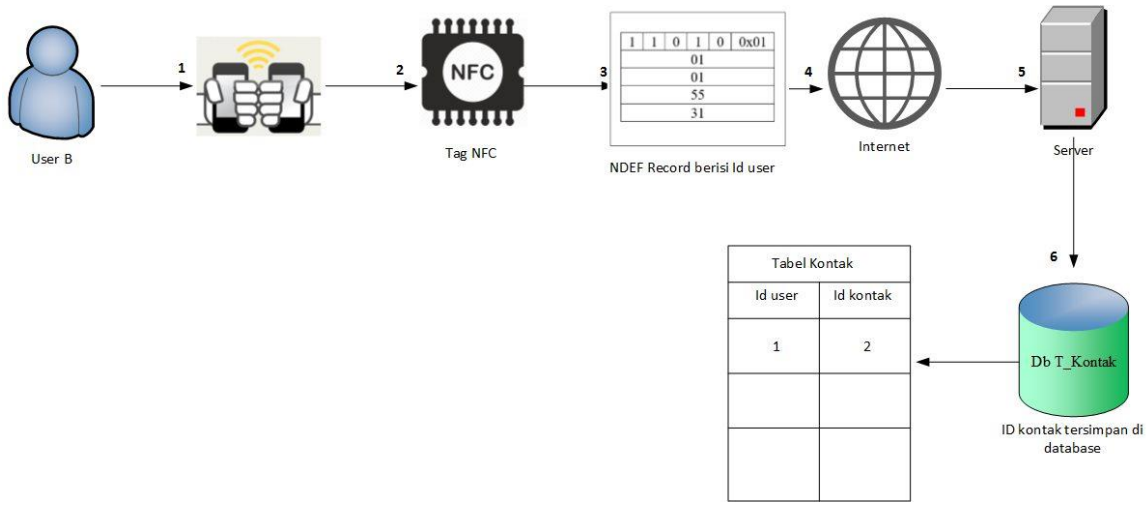
```
    textInfo = (TextView)findViewById(R.id.info);
    textOut = (EditText)findViewById(R.id.textout);

    nfcAdapter = NfcAdapter.getDefaultAdapter(this);
    if(nfcAdapter==null){
        Toast.makeText(NFCActivity.this,
                "nfcAdapter==null, no NFC adapter exists",
                Toast.LENGTH_LONG).show();
    }else{
        Toast.makeText(NFCActivity.this,
                "Set Callback(s)",
                Toast.LENGTH_LONG).show();
        nfcAdapter.setNdefPushMessageCallback(this, this);
        nfcAdapter.setOnNdefPushCompleteCallback(this, this);
    }
}
```

Gambar 3.10 Contoh *Source Code* Dalam Proses Penulisan NFC

3.1.5.3 Analisis Mekanisme Pembacaan NFC

Mekanisme proses pembacaan NFC melalui perangkat *mobile android* ditunjukan pada gambar 3.11



Gambar 3.11 Mekanisme Pembacaan NFC

Berikut adalah proses alur *encoding QrCode*:

1. Proses baca *tag* NFC dilakukan oleh *user* lain contoh "*User* B".
2. Sistem *mobile* membaca NDEF *Message* yang tersimpan didalam aplikasi .
   Proses membaca data NDEF *message* pada *tag* NFC ditangani oleh *tag* dispatch sistem, yang menganalisis *tag* NFC saat NDEF *message* ditemukan.
   Perangkat *android* akan mencari *tag* NFC meskipun layar perangkat dalam keadaan terkunci, terkecuali saat NFC dinonaktifkan.
   Ketika sebuah perangkat *android* memindai *tag* NFC yang berisi data NDEF, maka perangkat *android* akan mengurai pesan dan mencari tahu tipe data, apakah MIME atau URI. Untuk melakukan hal ini, pertama sistem membaca NDEF *record* didalam NDEF *message* untuk menterjemahkan seluruh pesan.
3. Di dalam *tag* NFC terdapat NDEF *Record* yang telah diisi.
4. Data tersebut akan diteruskan ke *server* dan database melalui jaringan internet.
5. *Server* menerima data *request*
6. Data di simpan kedalam database didalam tabel kontak.

Berikut contoh *Source Code* dalam proses menulis NFC pada aplikasi kuartu.

```
@Override
protected void onResume() {
    super.onResume();
    Intent intent = getIntent();
    if(intent.getAction() != null){
        String action = intent.getAction();
        if(action.equals(NfcAdapter.ACTION_NDEF_DISCOVERED)){
            Parcelable[] parcelables =
                    intent.getParcelableArrayExtra(
                            NfcAdapter.EXTRA_NDEF_MESSAGES);
            NdefMessage inNdefMessage = (NdefMessage)parcelables[0];
            NdefRecord[] inNdefRecords = inNdefMessage.getRecords();
            NdefRecord NdefRecord_0 = inNdefRecords[0];
            String inMsg = new String(NdefRecord_0.getPayload());
            textInfo.setText(inMsg);
        }
    }

}
```

```
    @Override
    protected void onNewIntent(Intent intent) {
        setIntent(intent);
    }

    @Override
    public void onNdefPushComplete(NfcEvent event) {

        final String eventString = "onNdefPushComplete\n" + event.toString();
        runOnUiThread(new Runnable() {

            @Override
            public void run() {
                Toast.makeText(getApplicationContext(),
                        eventString,
                        Toast.LENGTH_LONG).show();
            }
        });

    }

    @Override
    public NdefMessage createNdefMessage(NfcEvent event) {

        String stringOut = textOut.getText().toString();
        byte[] bytesOut = stringOut.getBytes();

        NdefRecord ndefRecordOut = new NdefRecord(
                NdefRecord.TNF_MIME_MEDIA,
                "text/plain".getBytes(),
                new byte[] {},
                bytesOut);

        NdefMessage ndefMessageout = new NdefMessage(ndefRecordOut);
        return ndefMessageout;
    }
}
```

Gambar 3.12 Contoh *Source Code* Proses Pembacaan NFC

3.1.6 Analisis *QrCode*

Untuk mengganti fisik sebuah kartu nama menjadi *QrCode*, diperlukan aplikasi yang terdiri dari *server* yang juga berfungsi sebagai encoder dan scanner yang berfungsi juga sebagai decoder. Pada penelitian ini, encoder dibangun pada *platform mobile* karena kartu nama yang dikelola merupakan kartu nama digital yang dikoleksikan dalam sebuah aplikasi. Decoder pada sistem ini berupa aplikasi scanner yang dibangun pada *platform Android*. Aplikasi scanner ini harus terinstal pada telepon seluler bersistem operasi *Android* yang memiliki kamera serta dapat terhubung dengan jaringan internet. Inputan dalam *image QrCode* berisi id *user* dan di konversi menjadi id kontak dan output yang dihasilkan adalah bertambah data kontak pertemanan, mendapatkan informasi langsung alamat saat pertukaran kartu nama, dapat melihat informasi data pribadi pengguna lainnya, dan dapat melakukan interaksi secara langsung didalam aplikasi. Berikut Gambaran Sistem Umum Proses *QrCode*:



Gambar 3.13 Prosedur Pertukaran Kartu nama dengan *QrCode*

Berikut adalah deskripsi dari gambar 3.13 Prosedur Pertukarana Kartu Nama dengan *QrCode*:

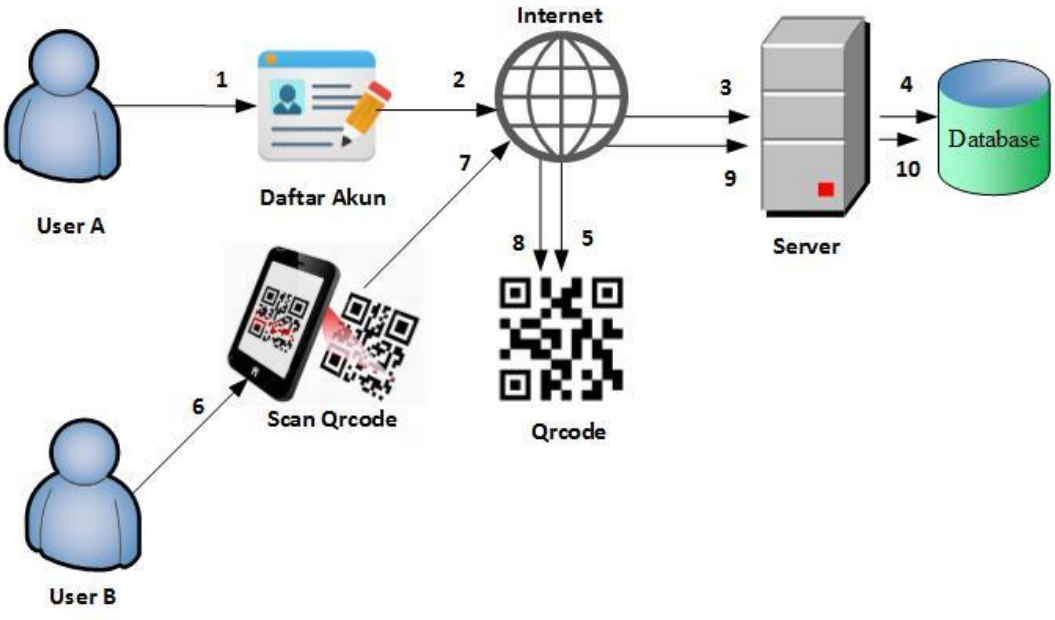
1. Pengguna bersiap mengoperasikan aplikasi untuk melakukan pertukaran kartu nama menggunakan *QrCode*.
2. Pengguna mengarahkan kamera ke *image QrCode* didalam aplikasi untuk melakukan *scanning* pembacaan qrcode. Ilustrasi mekanisme pembacaan *QrCode*, Pengguna pertama bertindak sebagai membaca *image QrCode* milik pengguna kedua. Inputan dalam *image QrCode* berisi id *user* pengguna kedua. Pengguna kedua mengarahkan *image QrCode*nya yang ada didalam aplikasi ke kamera pengguna pertama.

3. Setelah selesai terbaca, pengguna akan mendapatkan notifikasi berhasil dalam melakukan pertukaran kartu nama lalu pengguna akan mendapatkan informasi alamat secara langsung lokasi pertukaran kartu nama, pengguna dapat melihat informasi data pribadi pengguna lainnya dan siap untuk saling berinteraksi secara langsung.

3.1.6.1 Deskripsi Umum *Encoding* dan *Decoding QrCode*

Aplikasi yang dibangun merupakan aplikasi yang melakukan *encoding* dan *decoding* kartu nama menjadi *QrCode*. Pada proses *encoding*, terjadi perubahan data dari id *user* dalam database aplikasi yang bertipe integer dengan panjang 4 digit menjadi sebuah *QrCode* yang berupa citra. Dalam mengkonversi inputan berupa id *user* menjadi output berupa citra *QrCode* tersebut dilakukan dalam beberapa proses yang merupakan tahapan dari proses *encoding*.

Sedangkan untuk proses *decoding* dibutuhkan citra *QrCode* yang akan diambil secara realtime menggunakan kamera perangkat *Android* untuk kemudian dari input citra tersebut diolah lebih lanjut oleh sistem untuk mendapatkan id kontak, kemudian id kontak dikirim ke *server* guna masuk ke dalam database aplikasi dan berhasil menambah kontak pada akun kita.



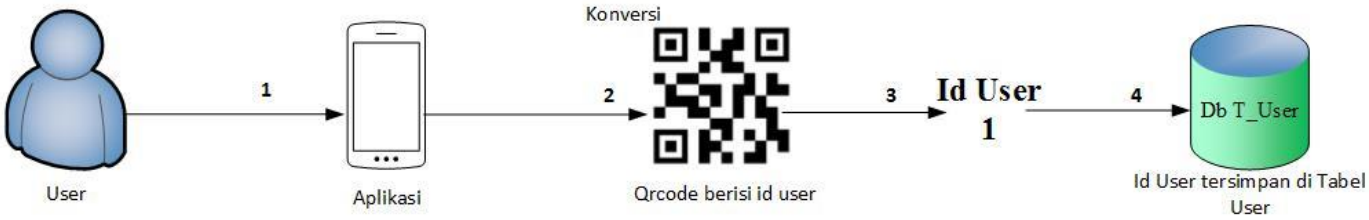
Gambar 3.14 Deskripsi Umum Proses *Encoding* dan *Decoding QrCode*

Berikut adalah deskripsi dari gambar 3.14 Deskripsi Umum Proses *Encoding* dan *Decoding QrCode*:

1. Pengguna A melakukan pendaftaran akun pada aplikasi. Pendaftaran berisi *email*, nama dan *password*
2. Data yang sudah didaftarkan akan diteruskan ke *server* dan disimpan didalam database melalui jaringan internet.
3. *Server* menerima data yang akan diteruskan ke dalam database.
4. Data telah tersimpan didalam database
5. Data yang sudah tersimpan akan dikonversi menjadi citra *QrCode* yang berisi Id *User* A.
6. *User* B bersiap melakukan *scanning QrCode* didalam aplikasi.
7. *User* B melakukan *scanning QrCode* milik *user* A melalui jaringan internet.
8. Proses *scanning* berhasil dan hasil *scanning* di konversi menjadi id kontak.
9. *Server* menerima data yang akan diteruskan ke dalam database.
10. Data telah tersimpan didalam database.

3.1.6.2 Analisis Mekanisme *Encoding* Qrcode

Mekanisme proses *encoding QrCode* melalui perangkat *mobile android* ditunjukan pada gambar 3.15



Gambar 3.15 Mekanisme *Encoding QrCode*

Berikut adalah proses alur *encoding QrCode*:

1. *User* melakukan pendaftaran di aplikasi dan mendapatkan id *user* dengan contoh “1”
2. Sistem akan mengkonversi id *user* berupa citra *QrCode*.
3. Sistem menampilkan citra qrcode yang berisi id *user*.
4. Sistem akan menyimpan id *user* kedalam database dan sistem akan menampilkan citra *QrCode* didalam aplikasi.

Berikut contoh *Source Code* dalam proses generate qrcode atau proses *encoding QrCode*

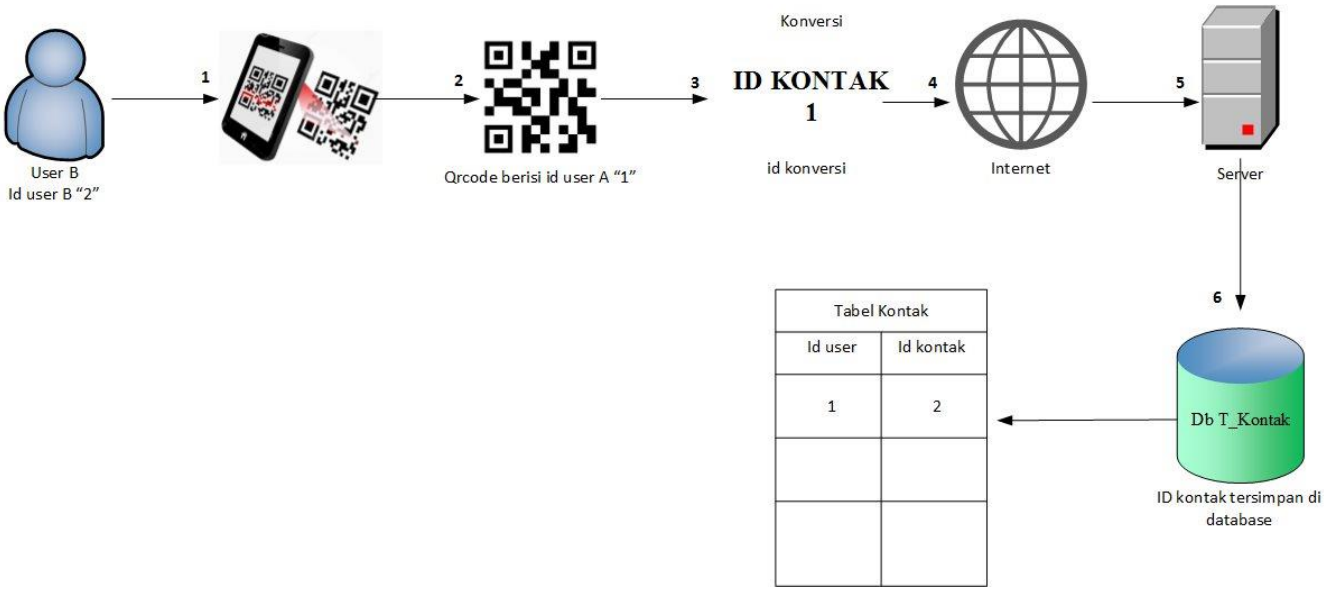
```
VCard kartuNama = new VCard(id_user + " " + nama_depan); // masukkan data id_user ke dalam qr code
QRCode.from(kartuNama).file(); // proses pembentukan qr code

Bitmap myBitmap = QRCode.from(kartuNama).bitmap(); // ubah qr code menjadi gambar bitmap
ImageView myImage = (ImageView) findViewById(R.id.imageView);
myImage.setImageBitmap(myBitmap); // tempel gambar qr code di imageview
```

Gambar 3.16 Contoh *Source Code* Proses *Encoding*

3.1.6.3 Analisis Mekanisme *Decoding* Qrcode

Mekanisme proses *decoding QrCode* melalui perangkat *mobile android* ditunjukan pada gambar 3.17



Gambar 3.17 Mekanisme *Decoding QrCode*

Berikut adalah proses alur *encoding QrCode*:

1. Proses *decoding QrCode* dilakukan oleh *user* lain contoh "*User* B" dimana *User* B mempunyai Id *User* "2".
2. Sistem melakukan pembacaan *QrCode* milik *user* lain contoh "*User* A". *QrCode* tersebut berisi id *user* A "1"
3. Setelah berhasil pembacaan, sistem mengkonversi id *user* A menjadi id kontak yang akan di simpan didalam database.
4. Data tersebut akan diteruskan ke *server* dan database melalui jaringan internet.

5. *Server* menerima data *request*.
6. Data di simpan kedalam database didalam tabel kontak.

Berikut contoh *Source Code* dalam proses *decoding* atau reader *QrCode*

```
private void scanBarcode(){
    qrEader = new QREader.Builder(this, surfaceView, new QRDataListener() {
        @Override public void onDetected(final String data) {
            Log.d("QREader", "Value : " + data);
            textView_qrcode_info.post(new Runnable() {
                @Override public void run() {
                    //textView_qrcode_info.setText(data);
                    detail_kontak = data;
                    //nama_kontak =
                    if (!detail_kontak.equals("")){
                        showDialogTambah();
                        qrEader.stop();
                        qrEader.releaseAndCleanup();
                    }
                }
            });
        }
    }).facing(QREader.BACK_CAM)
            .enableAutofocus(true)
            .height(800)
            .width(800)
            .build();

    qrEader.init();
}
```

Gambar 3.18 Contoh *Source Code Decoding QrCode*

3.1.7 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan erangkat lunak terbagi kedalam dua kebutuhan yaitu kebutuhan non fungsional dan fungsional. Kebutuhan fungsional dapat dilihat pada tabel 3.2 Spesifikasi Kebutuhan Fungsional

Tabel 3.2 Spesifikasi Kebutuhan Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-001	Subsistem *mobile* menyediakan fasilitas bagi pengguna untuk registrasi kedalam sistem
SKPL-F-002	Subsistem *mobile* menyediakan fasilitas bagi pengguna untuk *login* kedalam sistem
SKPL-F-003	Subsistem *mobile* menyediakan fasilitas bagi pengguna untuk mengisi identitas pribadi kedalam sistem
SKPL-F-004	Subsistem *mobile* menyediakan fasilitas fitur NFC bagi pengguna untuk pertukaran kartu nama dengan pengguna lain
SKPL-F-005	Subsistem *mobile* menyediakan fasilitas *QrCode* bagi pengguna untuk menambahkan kontak kartu nama dengan pengguna lain
SKPL-F-006	Subsistem *mobile* menyediakan fasilitas *chatting* bagi pengguna untuk berinteraksi dengan pengguna lain

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-007	Subsistem *mobile* menyediakan fasilitas notifikasi *chatting* bagi pengguna untuk melihat pemberitahuan pesan masuk yang dikirim oleh pengguna lain
SKPL-F-008	Subsistem *mobile* menyediakan fasilitas bagi pengguna untuk melihat *profile* pengguna lain.
SKPL-F-009	Subsistem *mobile* menyediakan fasilitas bagi pengguna untuk merubah *profile* pengguna
SKPL-F-010	Subsistem *mobile* menyediakan fasilitas lupa *password* bagi pengguna untuk masuk akun jika kita lupa dengan *password* yang didaftar

Kebutuhan nonfungsional pada perangkat lunak yang dibangun dapat dilihat pada tabel 3.3 Spesifikasi Kebutuhan Non Fungsional

Tabel 3.3 Spesifikasi Kebutuhan Non Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-NF-001	Sistem yang dibangun bersifat *mobile*
SKPL-NF-002	Sistem yang dibangun minimal menggunakan sistem operasi *android* versi 4.0 *Ice Cream Sandwich*
SKPL-NF-003	Sistem yang dibangun menggunakan perangkat kamera dan NFC
SKPL-NF-004	Sistem dibangun dengan spesifikasi hardware yang memenuhi standar minimum kebutuhan
SKPL-NF-006	Desain *user interface* pada perangkat lunak yang akan dibangun mengacu kepada *design google*
SKPL-NF-007	Sistem tidak mengizinkan konten yang mengandung unsur sara, Pornografi dan norma-norma kesusilaan.
SKPL-NF-008	Perangkat lunak yang dibangun harus menghasilkan format file standar (.json) yang bisa digunakan oleh pihak luar yang berkepentingan.

3.1.8 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang akan mempengaruhi perilaku pada sistem yang akan dibangun, diantaranya kebutuhan pengguna, analisi kebutuhan perangkat keras dan perangkat lunak sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.9 Analisis Kebutuhan Pengguna

Analisis pengguna adalah merupakan ketentuan pengguna yang dapat mengoprasionalkan atau menjalankan aplikasi kartu nama, adapun ketentuan pengguna adalah sebagai berikut:

1. Pengguna dapat mengoprasionalkan perangkat *smartphone android*
2. Perangkat *smartphone android* yang digunakan pengguna harus terkoneksi dengan internet.
3. Pengguna harus memiliki aplikasi kartu nama agar dapat menjalin komunikasi dan saling bertukar kartu nama dengan orang lain.

Dengan karakteristik pengguna diatas dapat disimpulkan bahwa pengguna yang dapat menggunakan aplikasi ini minimal harus memahami perangkat *smartphone android* dan dapat menjalankan aplikasi *android*.

3.1.10 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang akan digunakan berdasarkan kebutuhan minimal yang haru terpenuhi yaitu :

Tabel 3.4 Kebutuhan Perangkat Keras *Mobile*

No	Perangkat Keras	Kebutuhan Minimal
1	*Sistem Operasi*	*Jelly Bean*
2	*CPU*	800 Mhz
3	*Memory Internal*	1 GB
4	*RAM*	512 MB
5	*Konektifitas*	Koneksi Internet

3.1.11 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan proses analisis yang lebih menekankan kepada aspek pemanfaatan *software*. Kebutuhan minimal spesifikasi perangkat lunak pada perangkat *smartphone* yang akan menggunakan perangkat lunak ini dapat dilihat pada Tabel 3.5 Kebutuhan Perangkat Lunak.

Tabel 3.5 Kebutuhan Perangkat Lunak

No	Nama Perangkat Lunak	Keterangan
1	*Windows 10 Pro 64 bit*	Sistem operasi yang digunakan dalam membangun aplikasi.
2	*Java Development Kit versi 1.8.0*	Keperluan kompilasi kode-kode program.
3	*Android Studio*	Membangun aplikasi *android*.

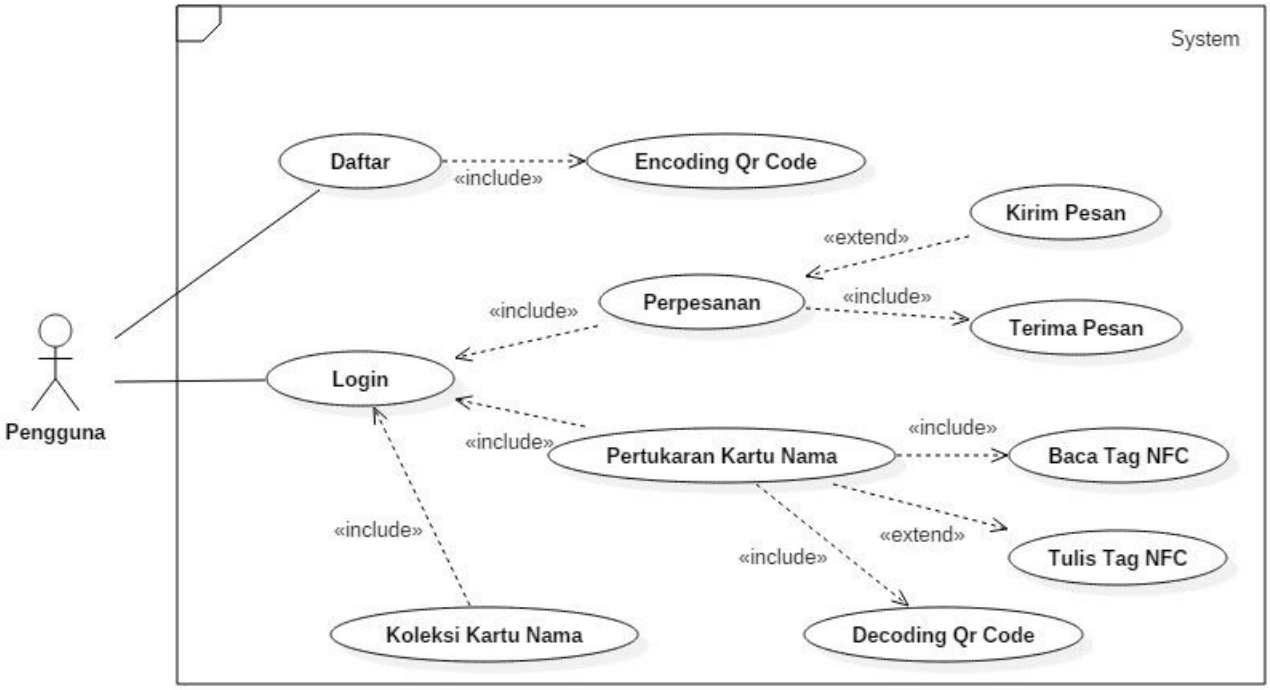
No	Nama Perangkat Lunak	Keterangan
4	*Android SDK*	Pengembnagan aplikasi *android*
5	*Android Development Tool*	Mengintegrasikan *Android* SDK pada *android* studio.

3.1.12 Analisis Kebutuhan Fungsional

Kebutuhan fungsional meliputi beberapa diagram UML, diantaranya: diagram *UseCase*, diagram *activity*, diagram *class* dan diagram *sequence*. Diagram tersebut dipergunakan sebagai desain rancangan untuk mempermudah proses pengembangan aplikasi khususnya untuk aplikasi yang dikembangkan menggunakan pemrograman berorientasi objek.

3.1.12.1 *UseCase* Diagram

UseCase diagram adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat pengguna. Dari identifikasi aktor yang terlibat di atas maka *UseCase Diagram mobile* dapat dilihat pada gambar 3.19



Gambar 3.19 *UseCase* Diagram

1. Deskripsi Aktor

Aktor adalah abstraksi dari orang dan sistem yang mengaktifkan fungsi dari target sistem. Berdasarkan *UseCase Diagram* pada Gambar 3.17, deskripsi aktor dapat dilihat pada Tabel 3.6

Tabel 3.6 Deskripsi Aktor

No.	Aktor	Deskripsi
1.	Pengguna	Merupakan aktor yang dapat berperan sebagai orang yang memakai aplikasi *mobile*.

2. Deskripsi *UseCase*

Deskripsi *UseCase* didalam sistem dapat dilihat pada Tabel 3.7.

Tabel 3.7 Deskripsi *UseCase*

No.	*UseCase*	Deskripsi
1.	Daftar	Sistem menampilkan form register untuk pengguna mendaftar akunnya di aplikasi.
2.	*Login*	Sistem menampilkan form *login* untuk pengguna untuk masuk ke akun aplikasi form ini terdiri dari *email* dan *password*.
3.	Perpesanan	Fungsionalitas ini digunakan oleh pengguna untuk melakukan interaksi dan komunikasi kepada akun lainnya.
4.	Kirim Pesan	Fungsionalitas ini digunakan oleh pengguna untuk mengirim pesan teks ke pengguna lainnya.
5.	Terima Pesan	Fungsionalitas ini digunakan untuk menerima pesan teks .
6.	Koleksi Kartu Nama	Fungsionalitas ini digunakan untuk mengelola kartu nama berdasarkan alphabet.

No.	*UseCase*	Deskripsi
7.	Pertukaran Kartu Nama	Fungsionalitas ini digunakan untuk menambah pertemanan atau pertukaran kartu nama dalam aplikasi
8.	Baca *Tag NFC*	Fungsionalitas ini digunakan untuk membaca data dan mengakses data pada *tag* NFC.
9.	Tulis *Tag NFC*	Fungsionalitas ini digunakan untuk menulis data inputan ke *tag* NFC.
10.	*Encoding QrCode*	Fungsionalitas ini digunakan untuk mengkonversi data inputan ke dalam heksadesimal.
11.	*Decoding QrCode*	Fungsionalitas ini digunakan untuk mengkonversi data dalam bentuk heksadesimal ke bentuk awal.

3. *UseCase* Skenario

UseCase scenario bertujuan untuk mendeskripsikan urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berdasarkan *UseCase Diagram* pada gambar 3.4 maka *UseCase scenario* dijelaskan sebagai berikut :

1) *UseCase* Daftar

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario daftar pada tabel 3.8

Tabel 3.8 Skenario *UseCase* Daftar

UseCase Name	Daftar
Goal Context	Pengguna baru berhasil terdaftar dalam sistem
Preconditions	Form Daftar untuk calon pengguna akun baru ditampilkan
Successful End Condition	Sistem menampilkan halaman utama aplikasi
Failed End Condition	Menampilkan pesan kesalahan daftar
Primary Actor	Pengguna

Trigger	Pengguna menekan tombol daftar	
Main Flow	*Step*	*Action*
	1	Pengguna menekan tombol belum memiliki akun
	2	Pengguna menekan klik disini
	3	Sistem melakukan validasi form masukan pengguna
	4	Sistem melakukan pemeriksaan data register
	5	Sistem menampilkan halaman utama aplikasi
Extension	*Step*	*Branching Action*
	3.1	Nama depan, nama belakang, *email*, *password* dan ulangi *password* kosong
	3.2	Nama depan, nama belakang, *email*, *password* dan ulangi *password* terisi
	4.1	*Email* sudah terdaftar
	4.2	*Email* belum terdaftar

2) *UseCase login*

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario *login* pada tabel 3.9

Tabel 3.9 Skenario *UseCase Login*

UseCase Name	*Login*	
Goal Context	Melakukan *login* kedalam perangkat lunak	
Preconditions	Form *login* perangkat lunak ditampilkan	
Successful End Condition	Sistem menampilkan halaman utama aplikasi	
Failed End Condition	Menampilkan pesan kesalahan *login*	
Primary Actor	Pengguna	
Trigger	Pengguna menekan tombol *login*	
Main Flow	*Step*	*Action*
	1	Sistem menampilkan form *login* perangkat lunak
	2	Pengguna memilih *login* via media sosial atau *login* via *email*
	3	Pengguna menekan tombol *login*
	4	Sistem melakukan validasi field *login*
	5	Sistem melakukan autentikasi data *login*
	6	Sistem menampilkan *activity* utama perangkat lunak
Extension	*Step*	*Branching Action*
	4.1	*Email* dan *password* kosong

	4.2	*Email* dan *password* terisi
	5.1	*Email* dan *password* tidak terdaftar
	5.2	*Email* dan *password* tidak valid.

3) *UseCase* Perpesanan

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario perpesanan pada tabel 3.10

Tabel 3.10 Skenario *UseCase* Perpesanan

UseCase Name	Perpesanan	
Goal Context	Melakukan komunikasi dengan pengguna lain	
Preconditions	Form list contact perangkat lunak ditampilkan	
Successful End Condition	Menampilkan *chat window* sesuai dengan *user* yang di tuju	
Failed End Condition	-	
Primary Actor	Pengguna	
Trigger	Pengguna memilih list contact akun *user* lain	
Main Flow	*Step*	*Action*
	1	Sistem menampilkan form contact list perangkat lunak
	2	Pengguna memilih *user* lain
	3	Pengguna menekan akun *user* lain
	4	Sistem menampilkan chat window
Extension	*Step*	*Branching Action*
	-	-

4) *UseCase* Kirim Pesan

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario Kirim Pesan pada tabel 3.11

Tabel 3.11 Skenario *UseCase* Kirim Pesan

UseCase Name	Kirim Pesan	
Goal Context	Mengirim pesan kepada pengguna lain	
Preconditions	Menampilkan c*hat window*	
Successful End Condition	Pesan terkirim ke web *server*	
Failed End Condition	Menampilkan pesan kesalahan Connection internet	
Primary Actor	Pengguna	
Trigger	Pengguna menekan tombol send	
Main Flow	Step	*Action*
	1	Sistem menampilkan chat window
	2	Pengguna mengirim pesan
	3	Pesan terkirim ke web *server*

Extension	Step	*Branching Action*
	3.1	Koneksi Internet tidak terhubung

5) *UseCase* Terima Pesan

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario Terima Pesan pada tabel 3.12

Tabel 3.12 Skenario *UseCase* Terima Pesan

UseCase Name	Terima Pesan	
Goal Context	Menampilkan pesan yang terkirim	
Preconditions	Menampilkan halaman chat	
Successful End Condition	Pesan di terima oleh pengguna lainny	
Failed End Condition	-	
Primary Actor	Web *Server*	
Trigger	-	
Main Flow	*Step*	*Action*
	1	Sistem menampilkan halaman chat
	2	Sistem menerima pesan dari web *server* di halaman chat
Extension	*Step*	*Branching Action*
	-	-

6) *UseCase* Koleksi Kartu Nama

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario Koleksi Kartu Nama pada tabel 3.13

Tabel 3.13 Skenario *UseCase* Koleksi kartu Nama

UseCase Name	Koleksi Kartu Nama	
Goal Context	Menampilkan data koleksi kartu nama	
Preconditions	Menampilkan halaman utama	
Successful End Condition	Menampilkan Halaman Koleksi kartu nama	
Failed End Condition	-	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu koleksi kartu	
Main Flow	*Step*	*Action*
	1	Sistem menampilkan halaman utama
	2	Pengguna memilih menu koleksi kartu nama

	3	Sistem menampilkan halaman koleksi kartu nama
Extension	*Step*	*Branching Action*
	-	-

7) *UseCase* Pertukaran Kartu Nama

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario Pertukaran Kartu Nama pada tabel 3.14

Tabel 3.14 Skenario *UseCase* Pertukaran Kartu Nama

UseCase Name	Pertukaran Kartu Nama	
Goal Context	Menambah pertemanan dan pertukaran kartu nama	
Preconditions	Menampilkan halaman utama	
Successful End Condition	Berhasil menambah pertemanan	
Failed End Condition	-	
Primary Actor	Pengguna	
Trigger	Pengguna menekan tombol add	
Main Flow	*Step*	*Action*
	1	Sistem menampilkan halaman utama
	2	Pengguna menekan tombol add
	3	Pengguna memilih QrCode atau NFC
Extension	*Step*	*Branching Action*
	-	-

8) *UseCase* Baca *Tag* NFC

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario Baca *Tag NFC* pada tabel 3.15

Tabel 3.15 Skenario *UseCase* Baca *Tag* NFC

UseCase Name	Baca*Tag* NFC	
Goal Context	Membaca data inputan *tag* NFC di *smartphone*	
Preconditions	Menampilkan halaman add Account	
Successful End Condition	Berhasil menambah pertemanan atau pertukaran kartu nama	
Failed End Condition	Gagal membaca data *tag* NFC	
Primary Actor	Sistem	
Trigger	Mendekatkan *smartphone* ke smarphone lainnya	
Main Flow	*Step*	*Action*
	1	*Tag* NFC sudah terisi data
	2	Sistem menampilkan halaman add account
	3	Pengguna menekan tombol NFC

	4	Pengguna mendekatkan smarphone dengan smarphone lainnya
Extension	*Step*	*Branching Action*
	-	-

9) *UseCase* Tulis *Tag* NFC

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario Tulis *Tag NFC* pada tabel 3.16

Tabel 3.16 Skenario *UseCase* Tulis *Tag* NFC

UseCase Name	Tulis *Tag* NFC	
Goal Context	Menyimpan data inputan *tag* NFC di *smartphone*	
Preconditions	Menampilkan Halaman Utama	
Successful End Condition	Berhasil menyimpan data inputan	
Failed End Condition	Gagal menyimpan data *tag* NFC	
Primary Actor	Sistem	
Trigger	Mendekatkan *smartphone* ke smarphone lainnya	
Main Flow	*Step*	*Action*
	1	Menyimpan data inputan *tag* NFC
Extension	*Step*	*Branching Action*
	-	-

10) *UseCase Encoding* QrCode

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario *Encoding QrCode* pada tabel 3.17

Tabel 3.17 Skenario *UseCase Encoding* QrCode

UseCase Name	*Encoding* QrCode	
Goal Context	Menyimpan data inputan QrCode di *smartphone*	
Preconditions	Menampilkan Halaman Utama	
Successful End Condition	Berhasil menyimpan data inputan	
Failed End Condition	Gagal menyimpan data inputan	
Primary Actor	Sistem	
Trigger		
Main Flow	*Step*	*Action*
	1	Sistem membangkitkan QrCode
	2	Sistem Menampilkan QrCode
Extension	*Step*	*Branching Action*
	-	-

11) *UseCase Decoding* QrCode

Interaksi antara aktor pengguna dengan *UseCase* di jelaskan dalam *UseCase* skenario *Decoding QrCode* pada tabel 3.18

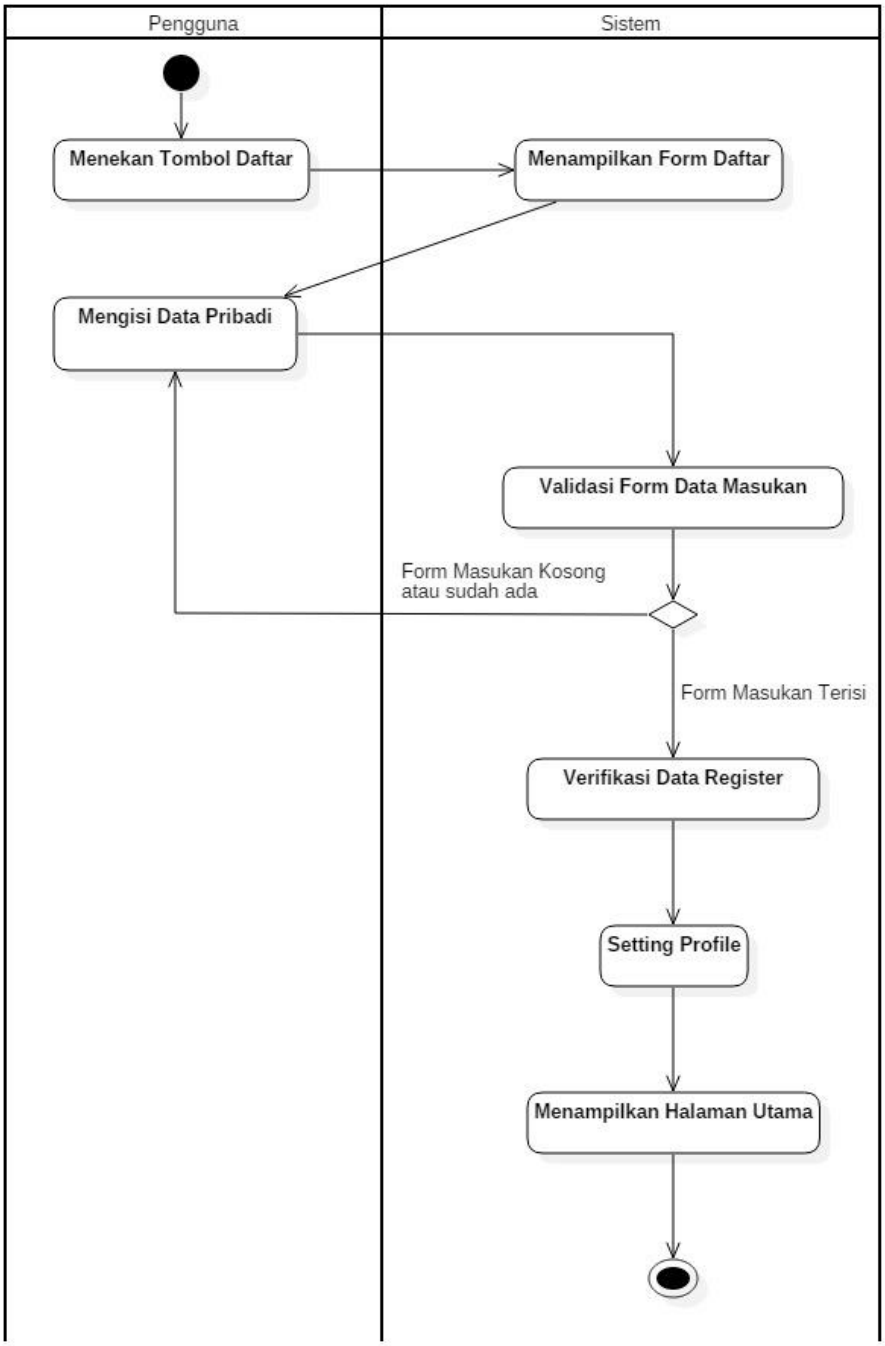
Tabel 3.18 Skenario *UseCase Decoding* QrCode

UseCase Name	*Decoding* QrCode	
Goal Context	Membaca QrCode *user* lain	
Preconditions	Menampilkan halaman add Account	
Successful End Condition	Berhasil menambah pertemanan atau pertukaran kartu nama	
Failed End Condition	Gagal *Scanning* QrCode	
Primary Actor	Pengguna	
Trigger	Mengarahkan kamera ke QrCode	
Main Flow	*Step*	*Action*
	1	Sistem menampilkan halaman add account
	2	Pengguna menekan tombol QrCode
	3	Pengguna mengarahkan kamera ke QrCode
	4	Sistem Mendecode QrCode
Extension	*Step*	*Branching Action*
	-	-

3.1.12.2 *Activity* Diagram

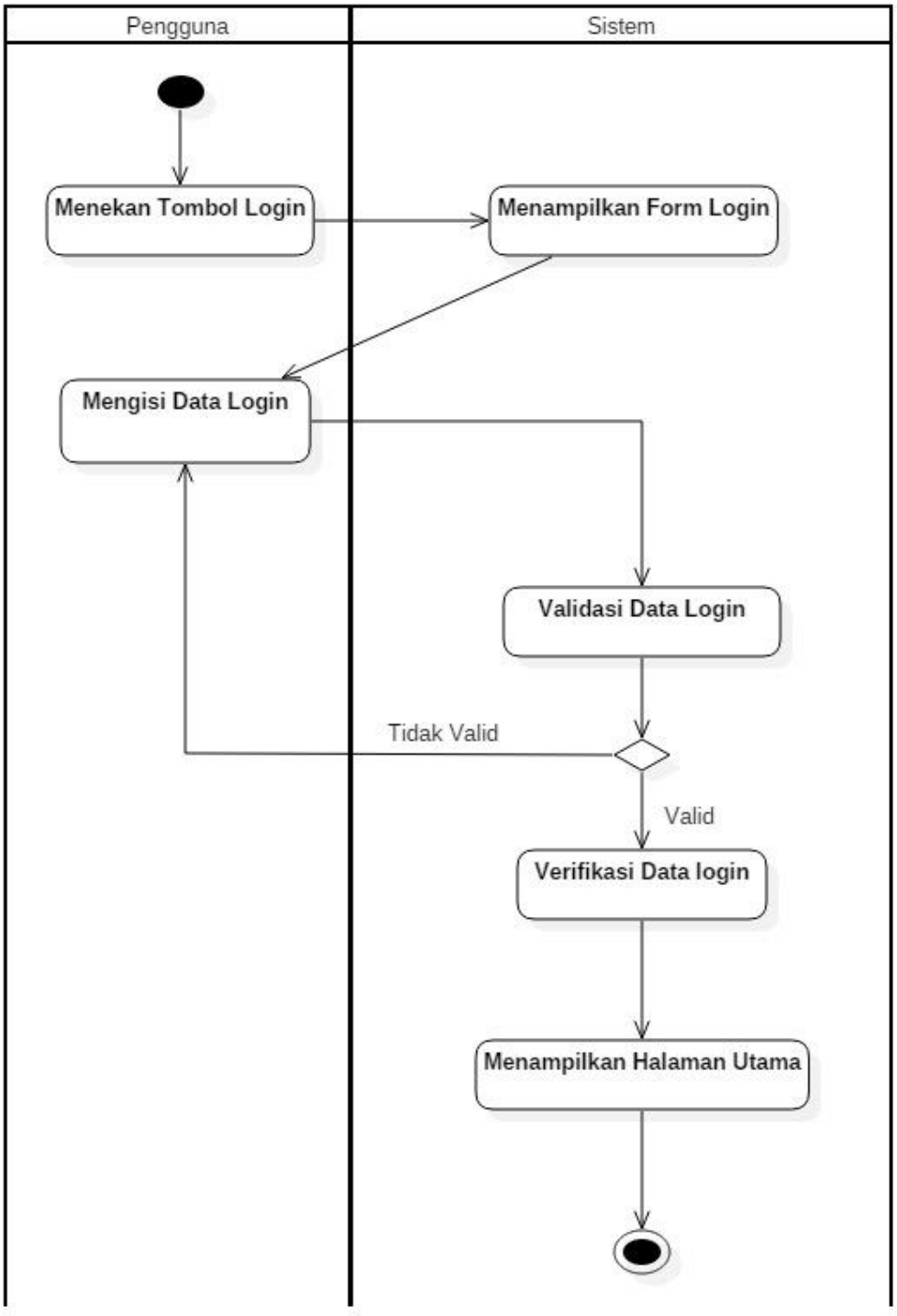
Activity diagram menjelaskan aliran kerja aktor pada perangkat lunak yang akan dibangun

1) *Activity* Diagram Daftar



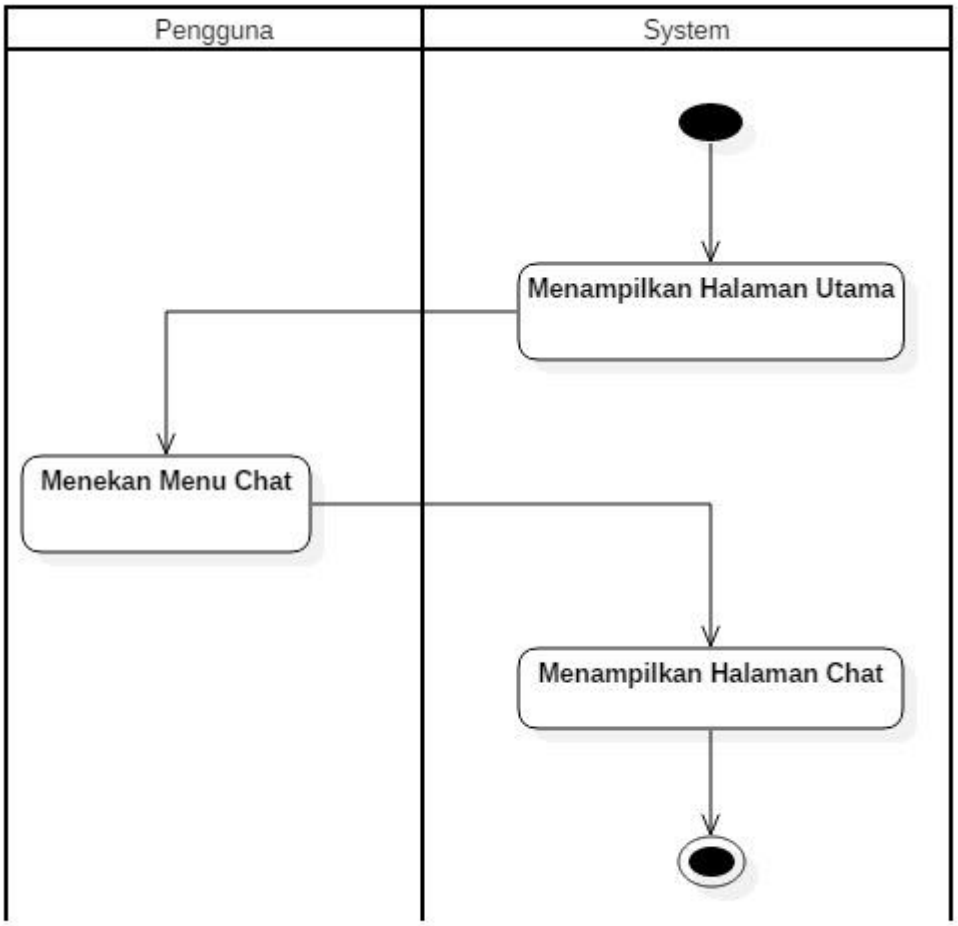
Gambar 3.20 *Activity* Diagram Register

2) *Activity* Diagram *Login*



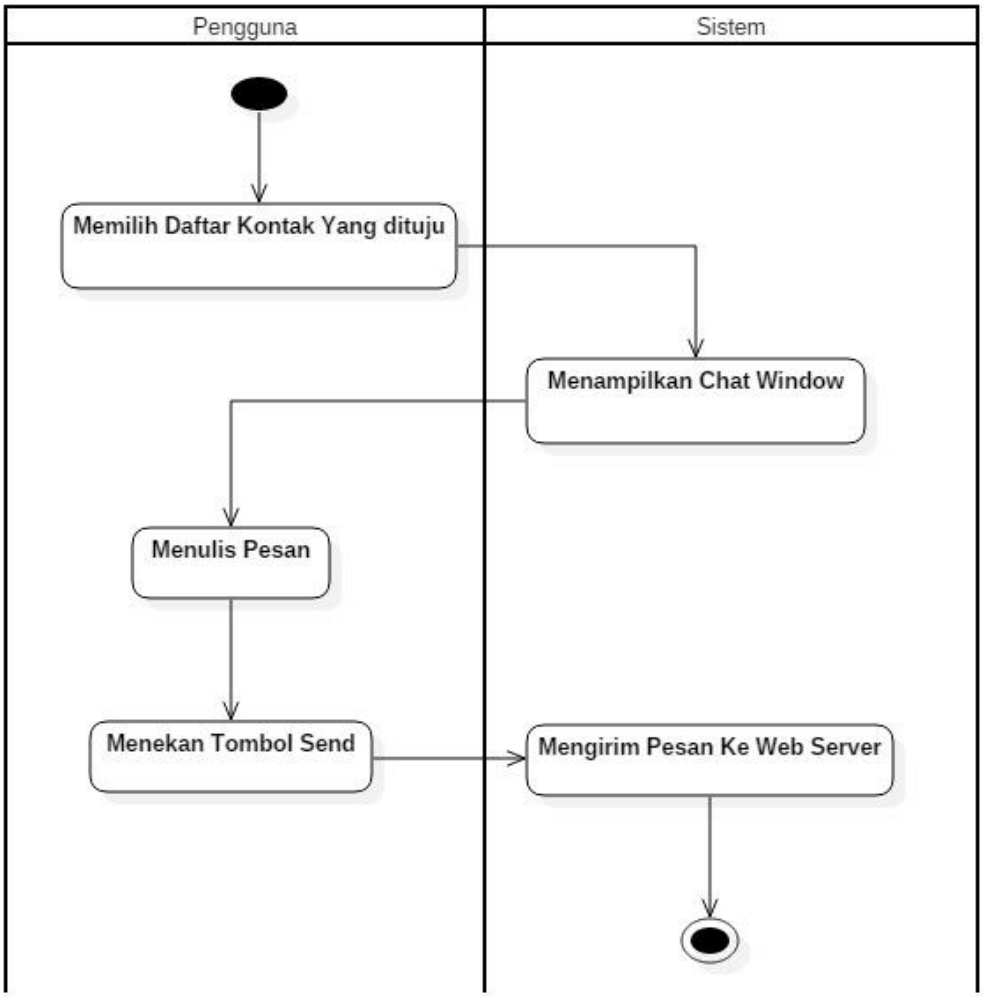
Gambar 3.21 *Activity* Diagram *Login*

3) *Activity* Diagram Perpesanan



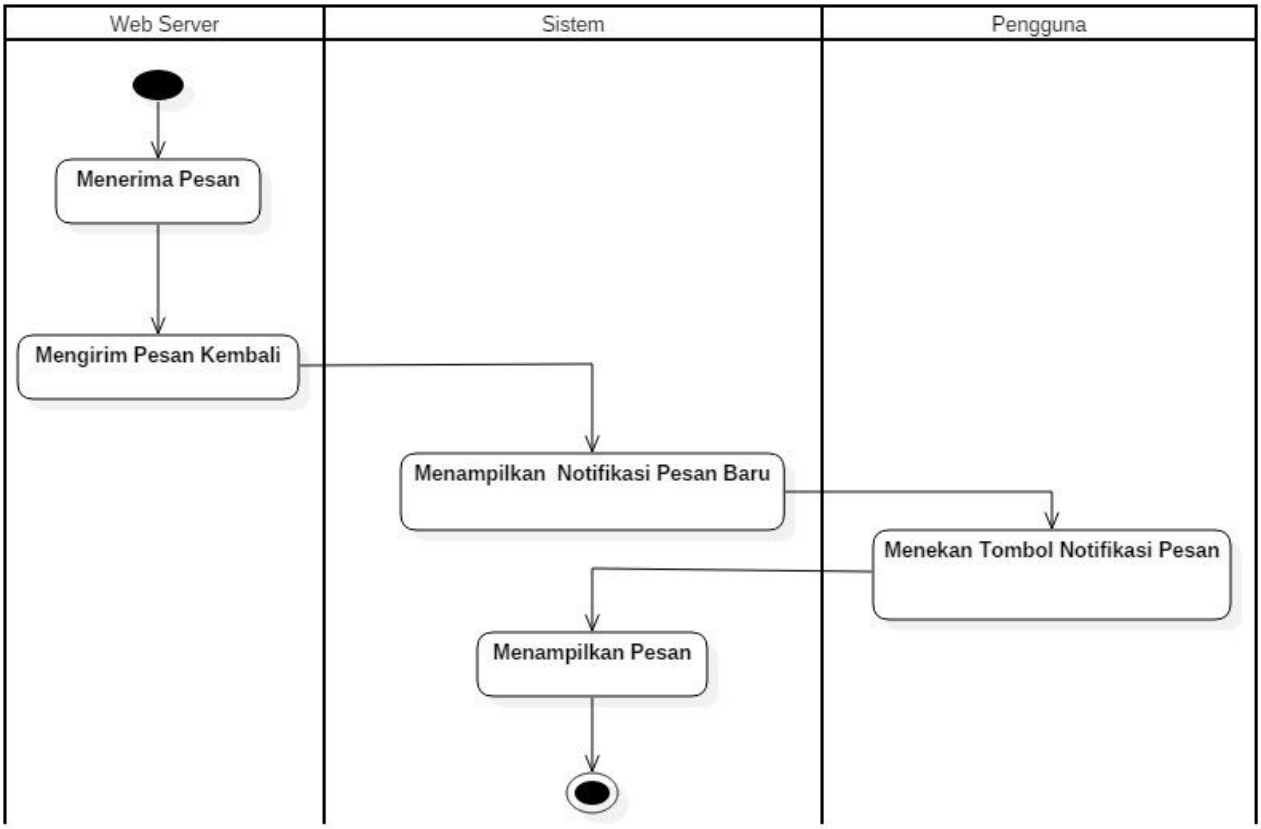
Gambar 3.22 *Activity* Diagram Perpesanan

4) *Activity* Diagram Kirim Pesan



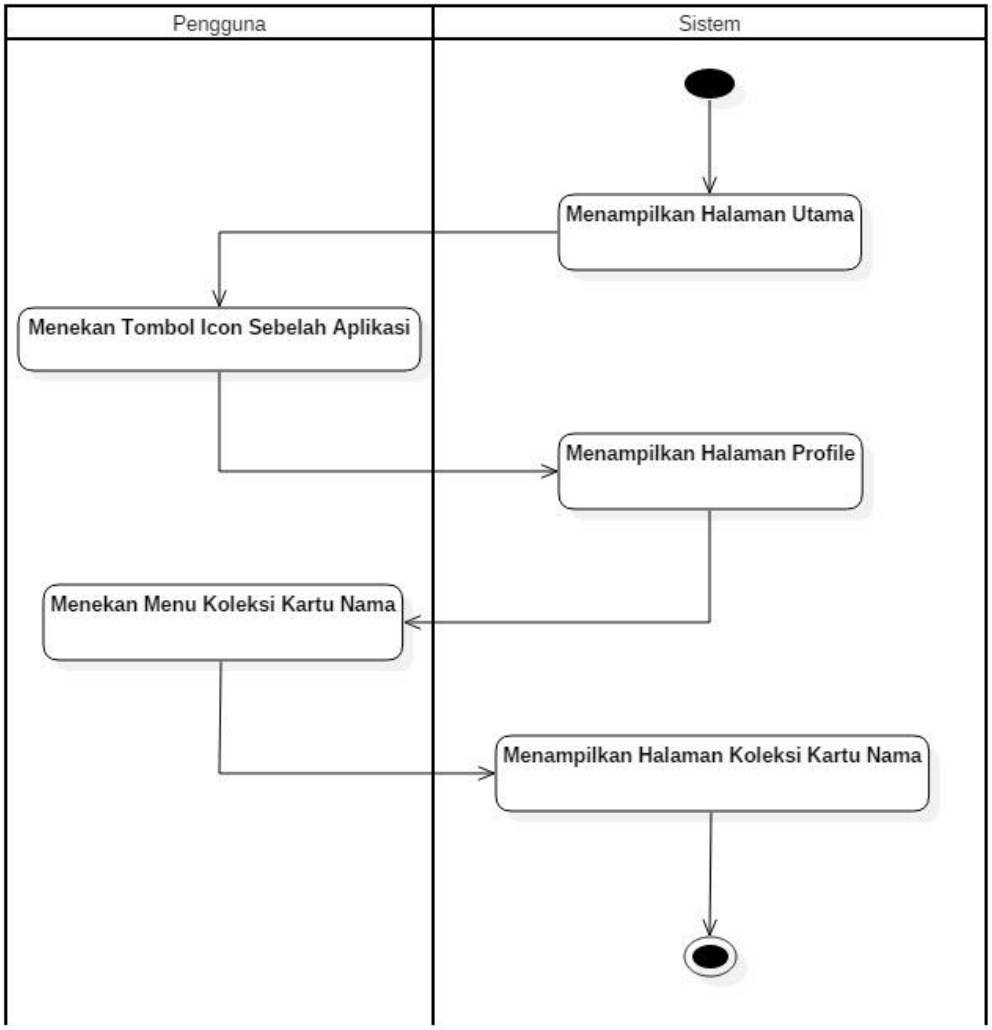
Gambar 3.23 *Activity* Diagram Kirim Pesan

5) *Activity* Diagram Terima Pesan



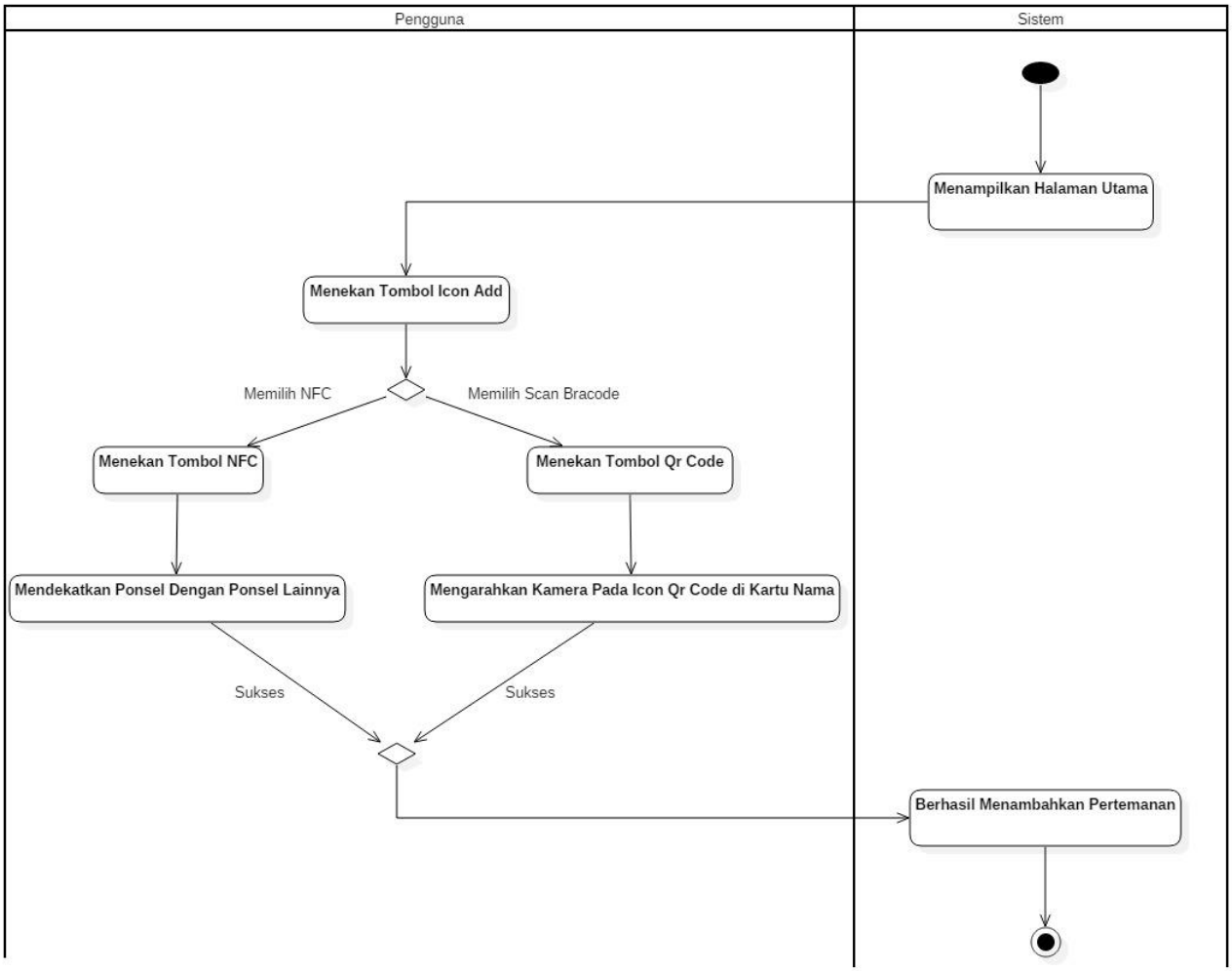
Gambar 3.24 *Activity* Diagram Terima Pesan

6) *Activity* Diagram Koleksi Kartu Nama



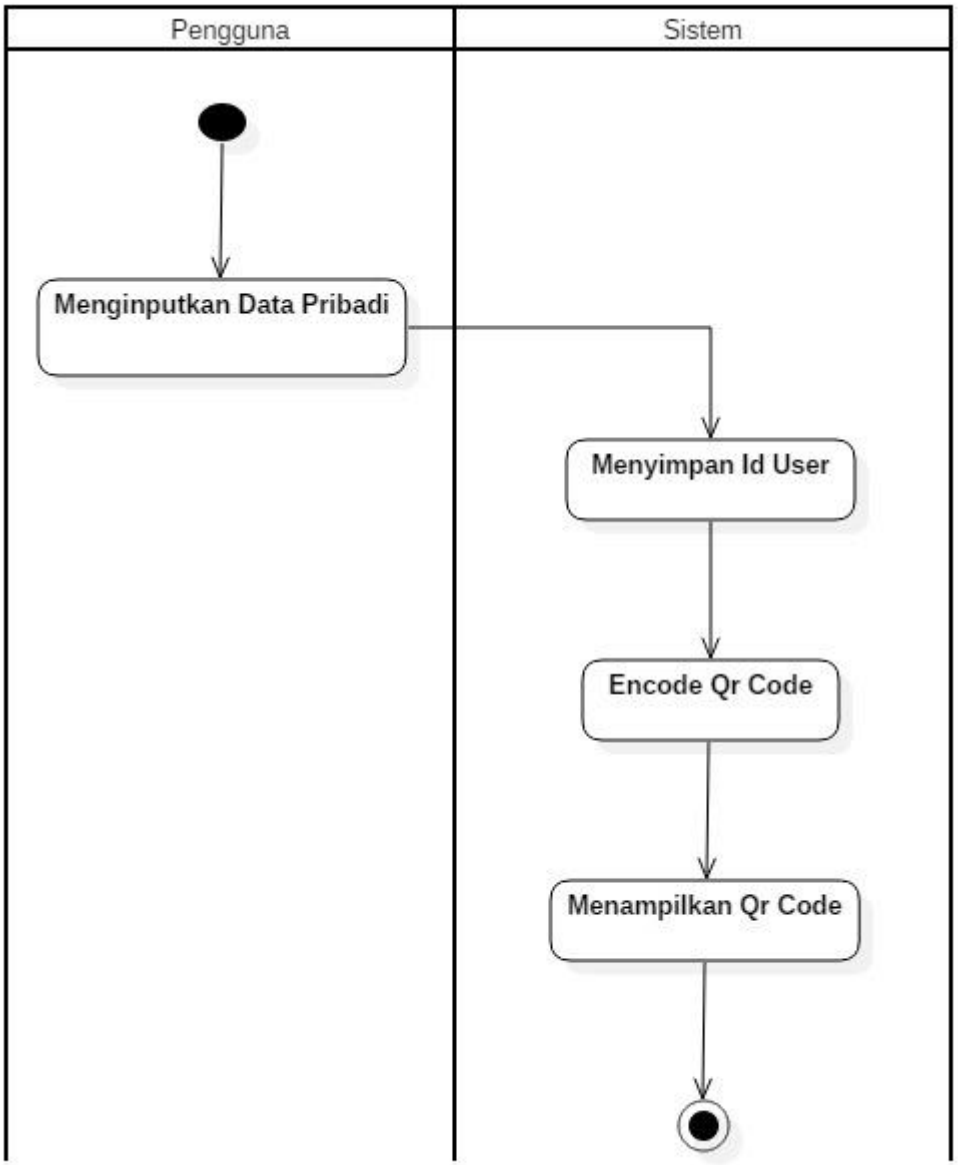
Gambar 3.25 *Activity* Diagram Koleksi Kartu Nama

7) *Activity* Diagram Pertukaran Kartu Nama



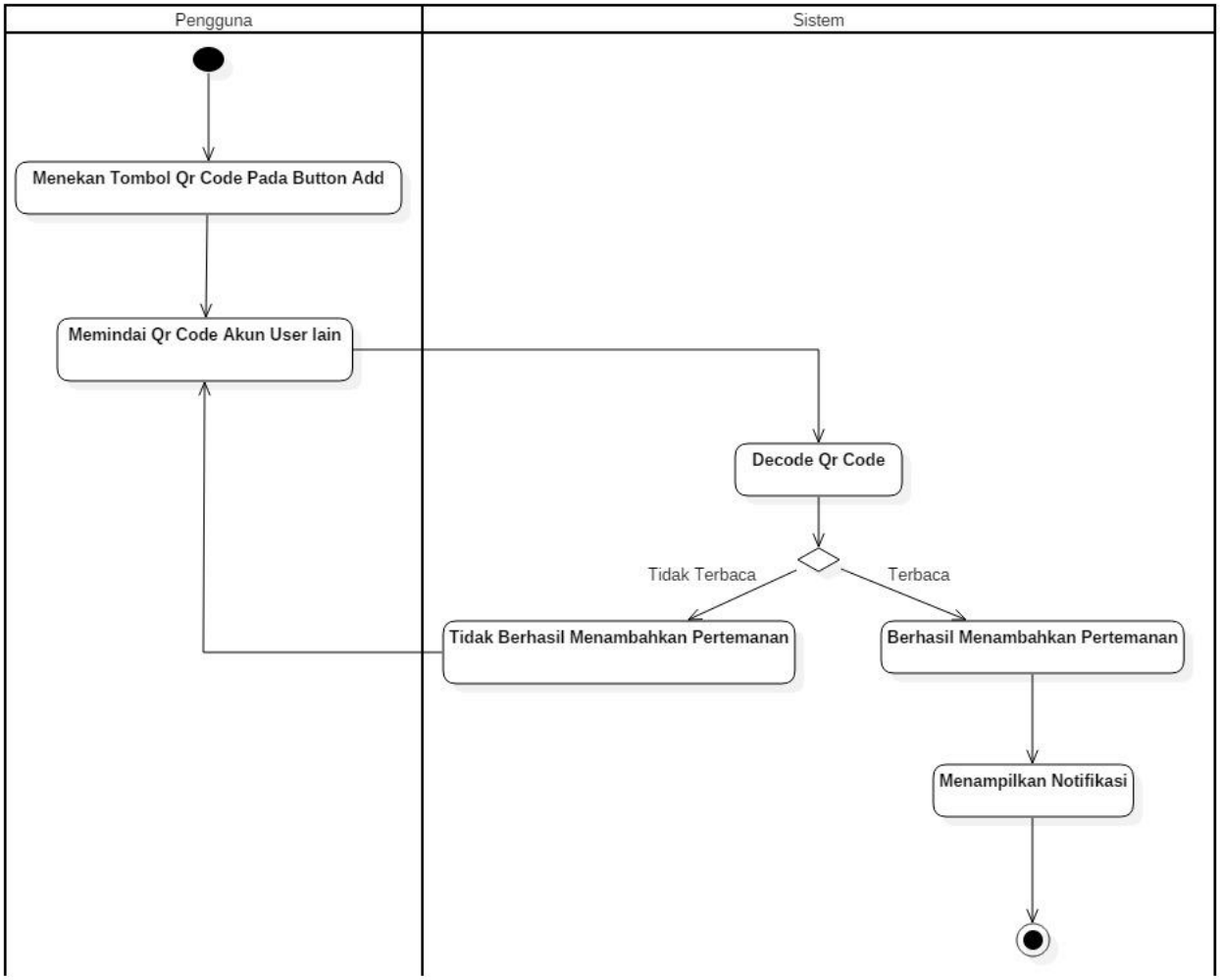
Gambar 3.26 *Activity* Diagram Pertukaran Kartu Nama

8) *Activity* Diagram *Encoding QrCode*



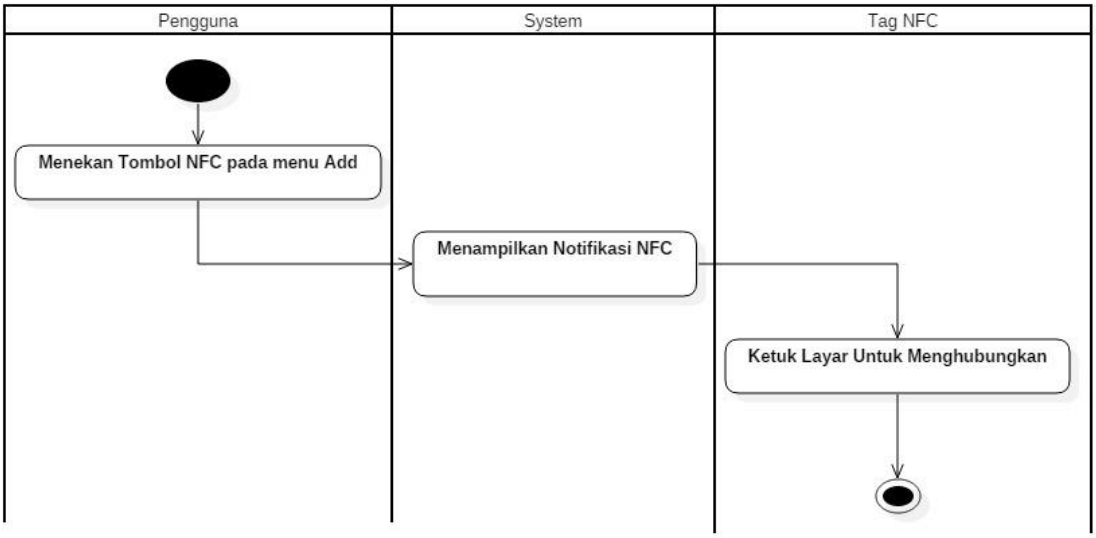
Gambar 3.27 *Activity* Diagram *Encoding QrCode*

9) *Activity* Diagram *Decoding QrCode*



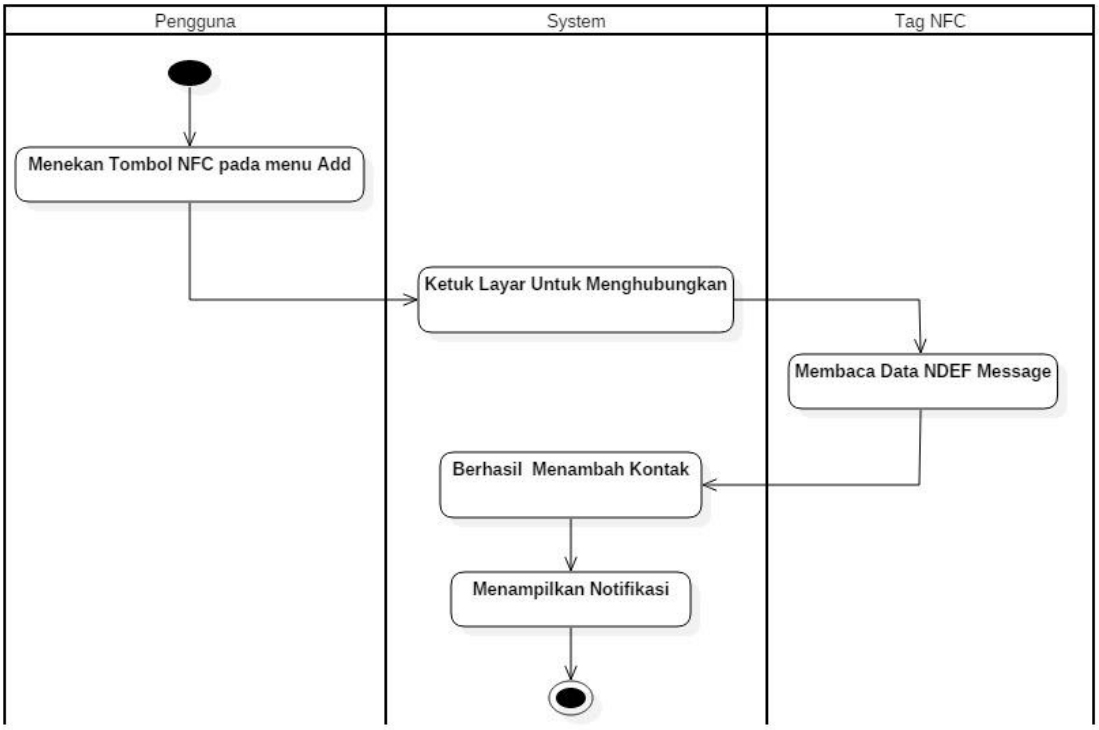
Gambar 3.28 *Activity* Diagram *Decoding QrCode*

10) *Activity* Diagram Tulis *Tag* NFC



Gambar 3.29 *Activity* Digaram Tulis *Tag* NFC

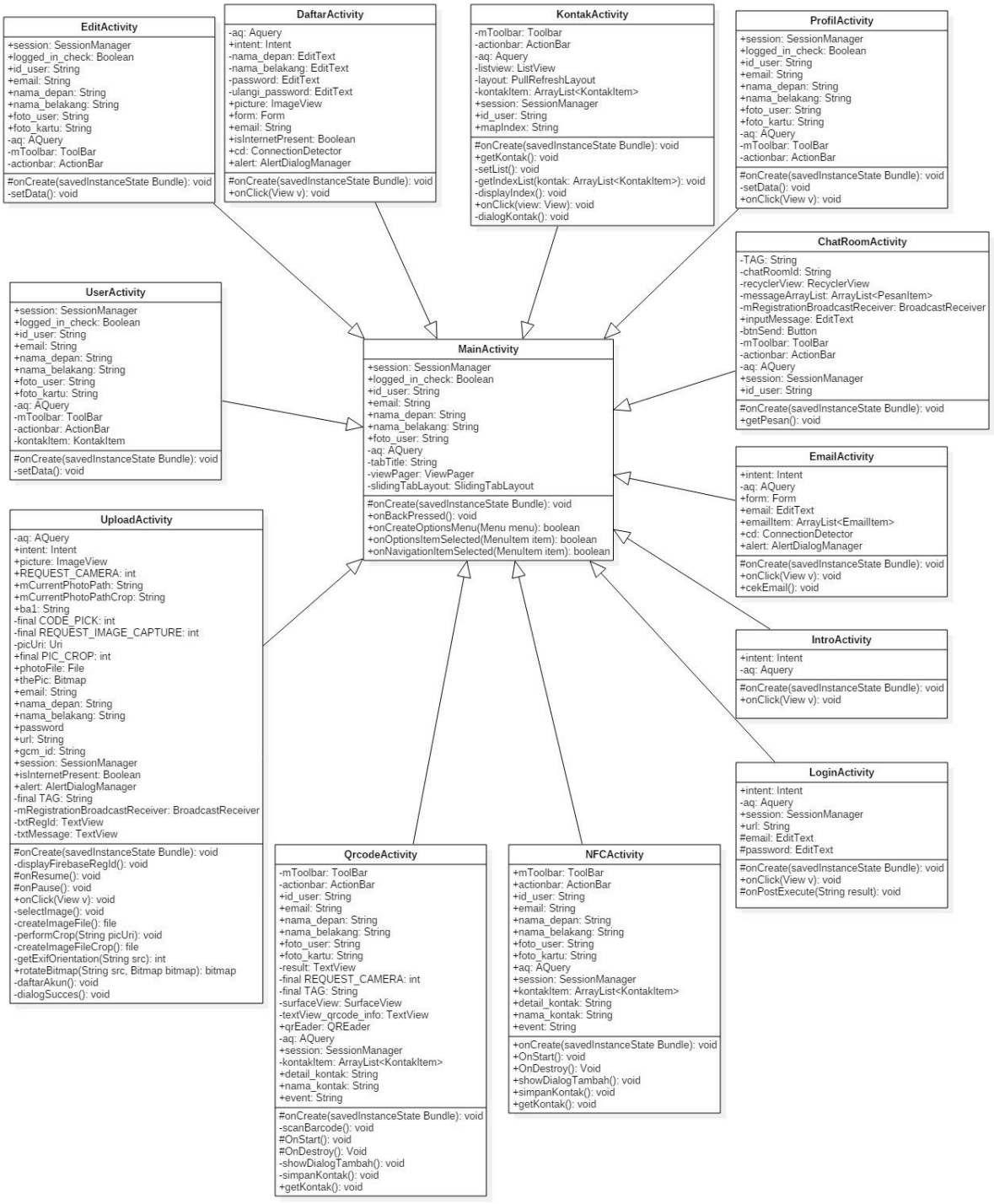
11) *Activity* Diagram Baca *Tag* NFC



Gambar 3.30 *Activity* Diagram Baca *Tag* NFC

3.1.12.3 *Class* Diagram

Diagram *Class* digunakan untuk menggambarkan secara abstrak struktur dari aplikasi yang akan dibangun, *class-class* yang terlibat, serta hubungan antar*class* untuk saling berkomunikasi satu sama lain.

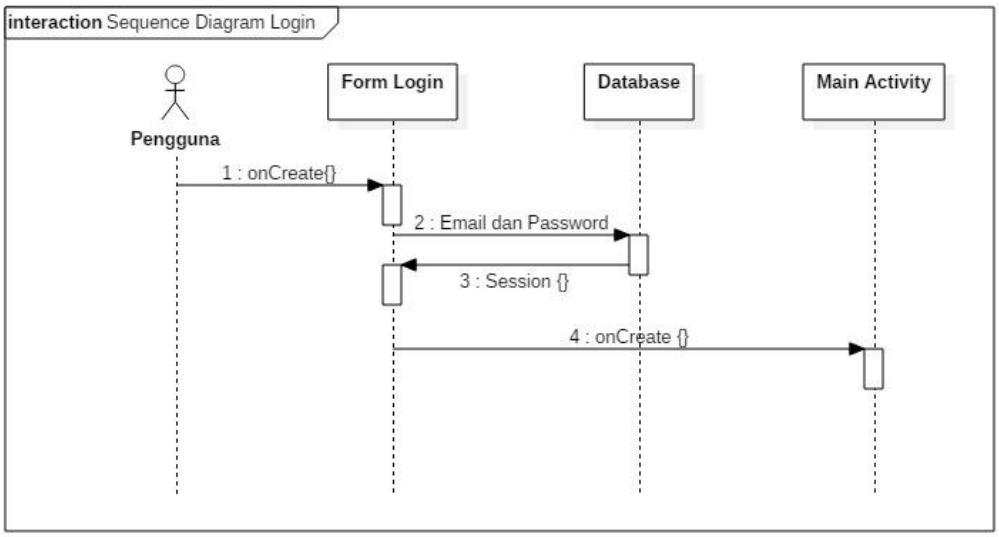


Gambar 3.31 *Class* Diagram

3.1.12.4 *Sequence* Diagram

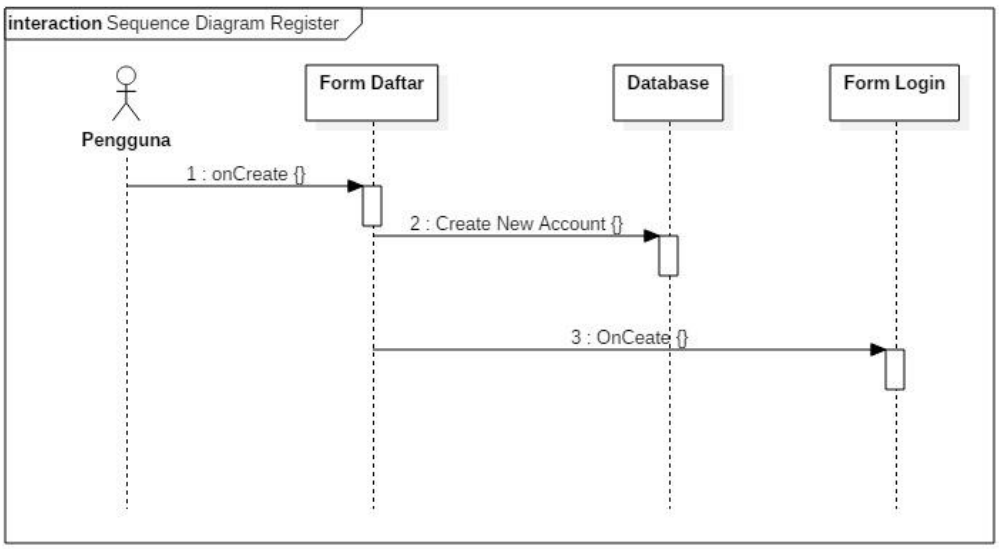
Diagram *sequence* menggambarkan urutan aktifitas yang terjadi di dalam sistem. Diagram ini menunjukkan pengiriman pesan-pesan yang melewati objek yang terlibat di dalam sistem.

1) *Sequence* Diagram *Login*



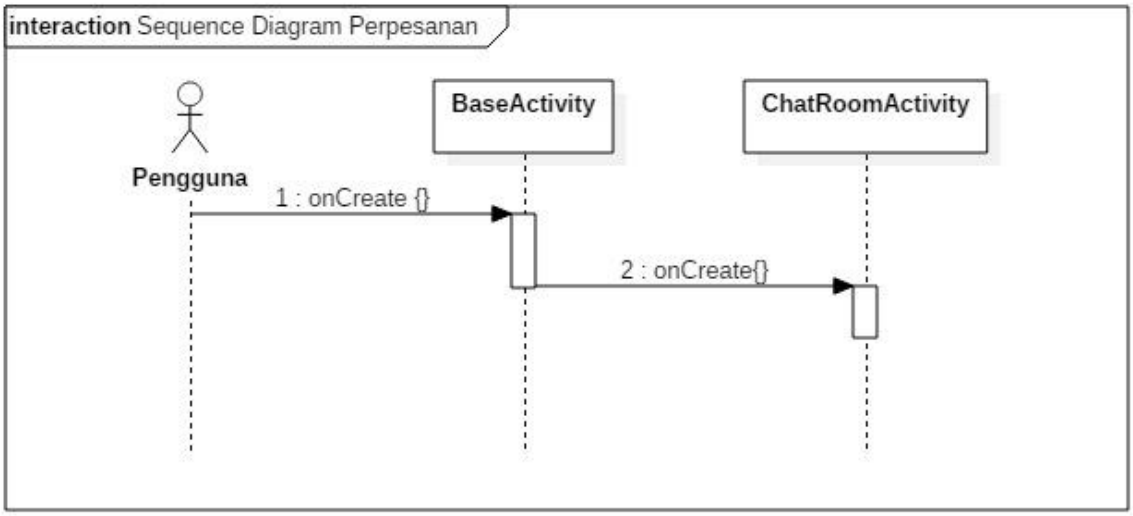
Gambar 3.32 Diagram *Sequence Login*

2) *Sequence* Diagram Register



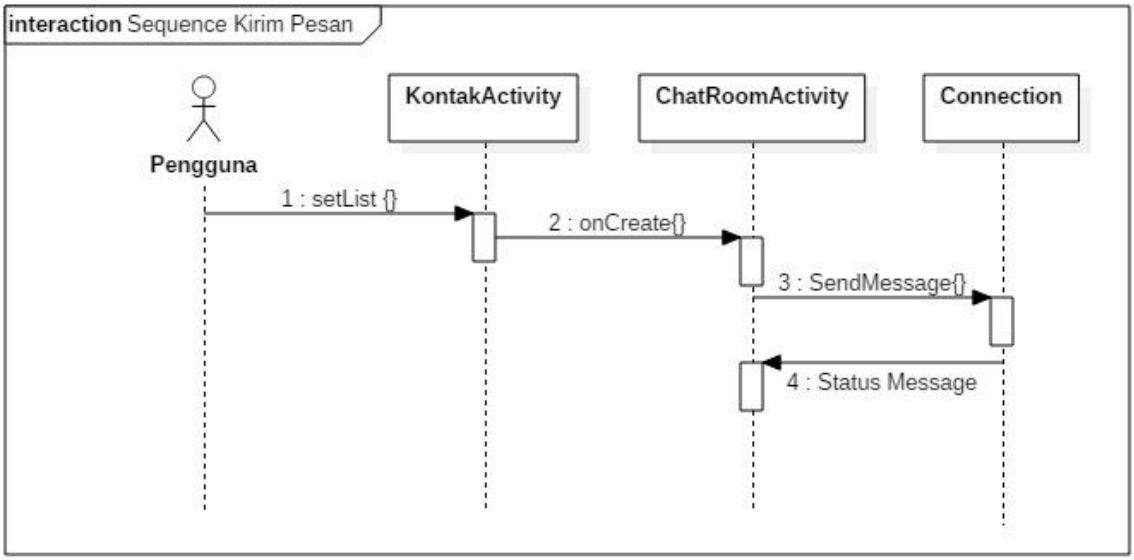
Gambar 3.33 Diagram *Sequence* Register

3) *Sequence* Diagam Perpesanan



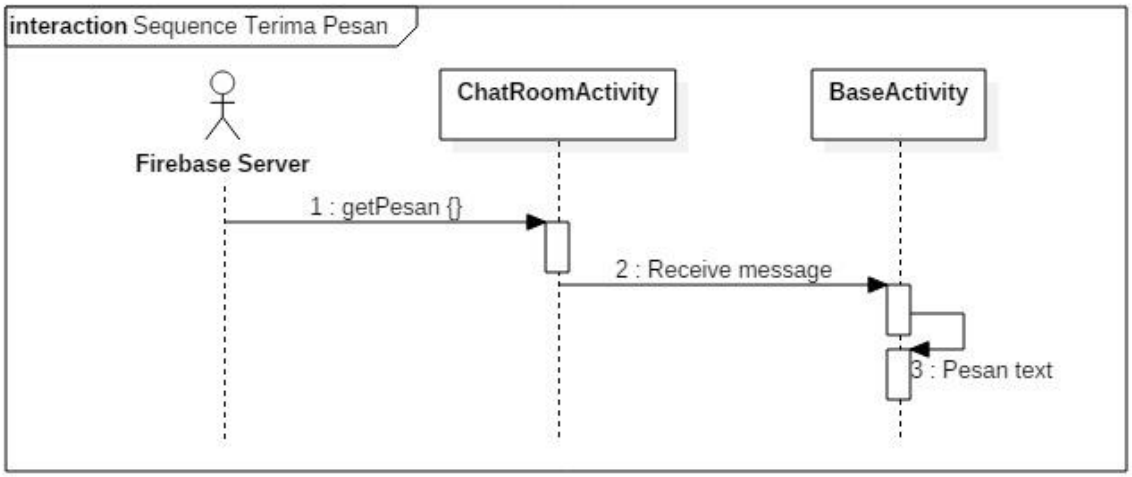
Gambar 3.34 Diagram *Sequence* Perpesanan

4) *Sequence* Diagam Kirim Pesan



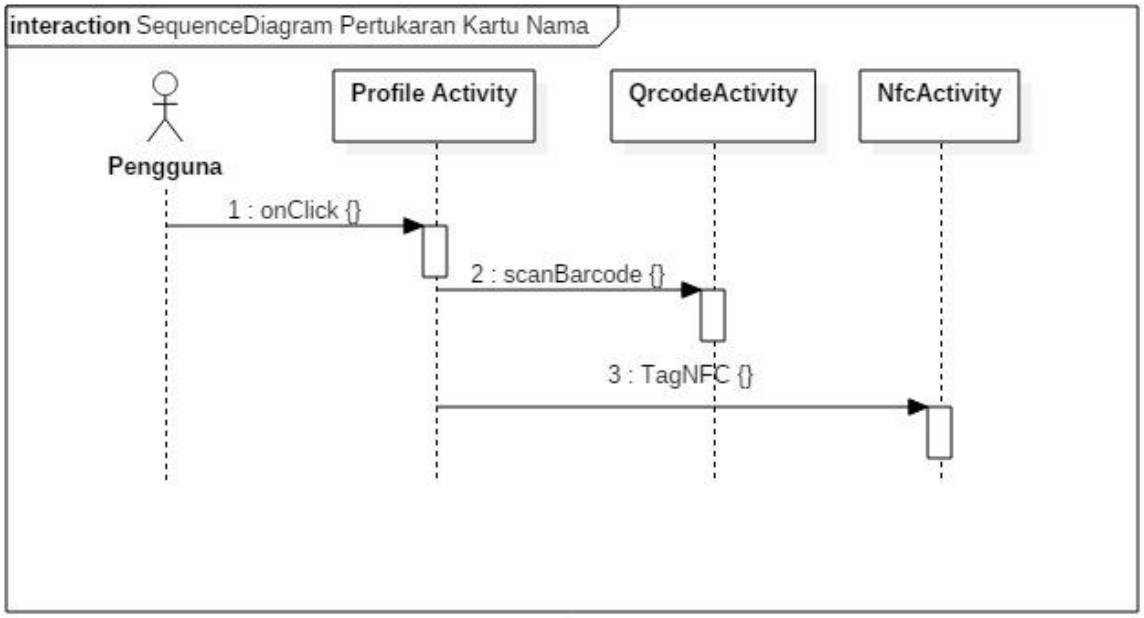
Gambar 3.35 Diagram *Sequence* Kirim Pesan

5) *Sequence* Diagam Terima Pesan



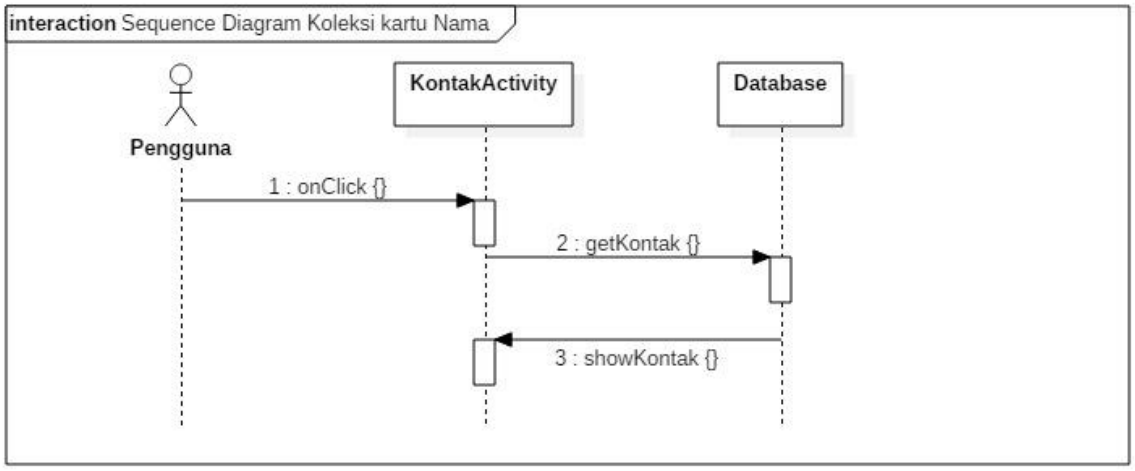
Gambar 3.36 Diagram *Sequence* Terima Pesan

6) *Sequence* Diagam Pertukaran Kartu Nama



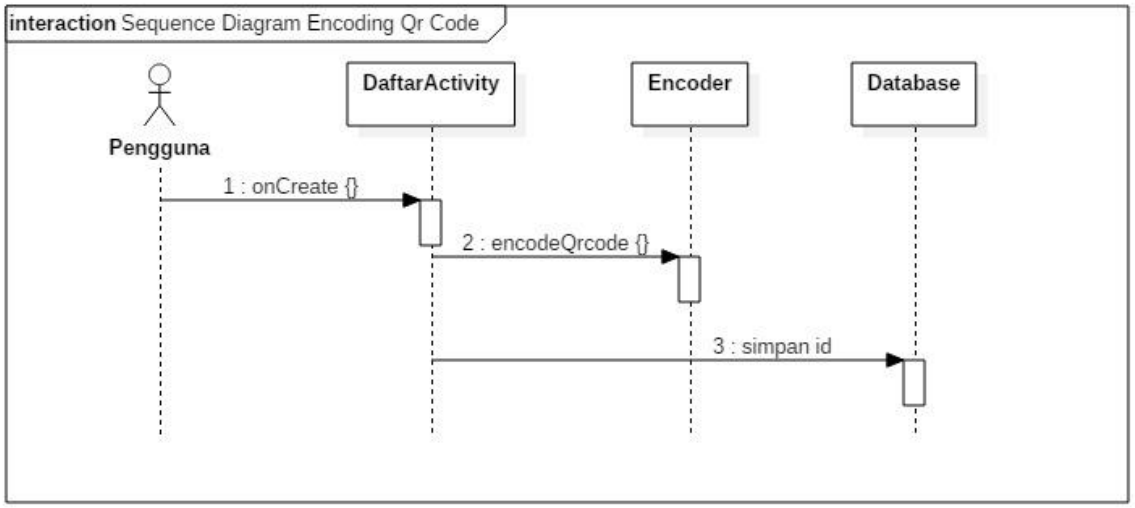
Gambar 3.37 Diagram *Sequence* Pertukaran Kartu Nama

7) *Sequence* Diagam Koleksi Kartu Nama



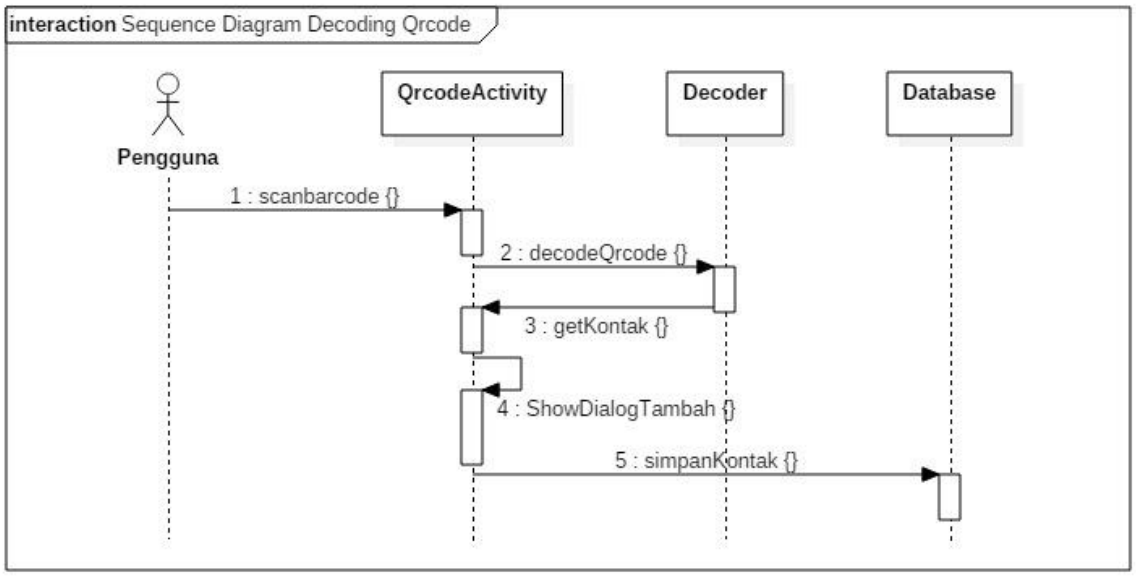
Gambar 3.38 Diagram *Sequence* Koleksi Kartu Nama

8) *Sequence* Diagam *Encoding QrCode*



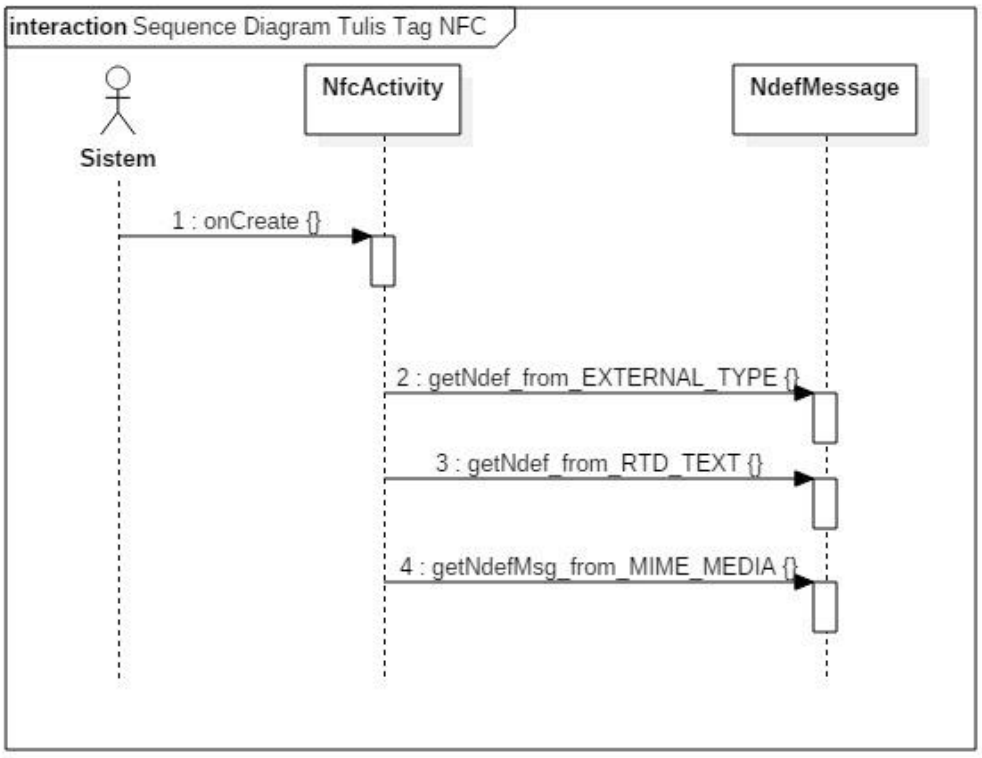
Gambar 3.39 Diagram *Sequence Encoding* QrCode

9) *Sequence* Diagam *Decoding QrCode*



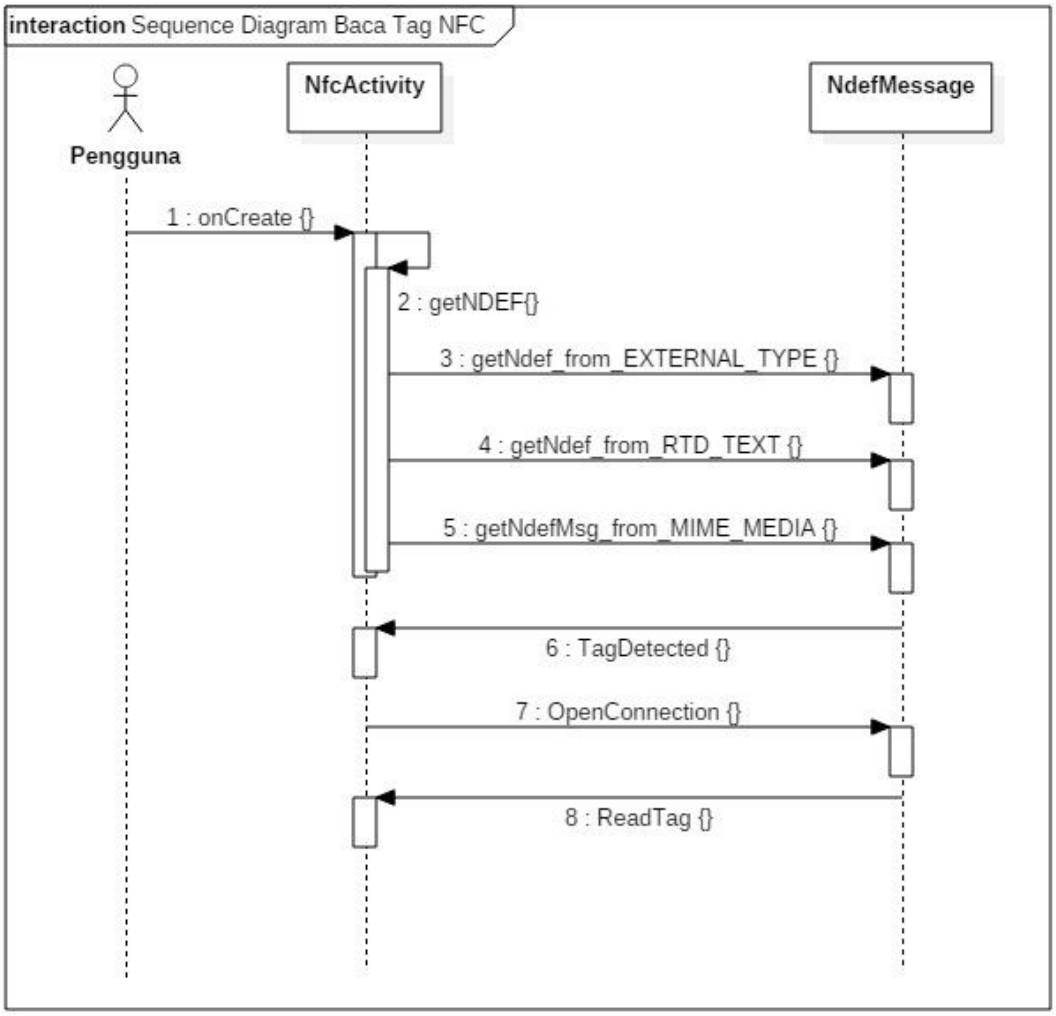
Gambar 3.40 Diagram *Sequence Decoding* QrCode

10) *Sequence* Diagram Tulis *Tag* NFC



Gambar 3.41 Diagram *Sequence* Tulis *Tag* NFC

11) *Sequence* Diagram Baca *Tag* NFC



Gambar 3.42 Diagram *Sequence* Baca *Tag* NFC

3.2 Perancangan Sistem

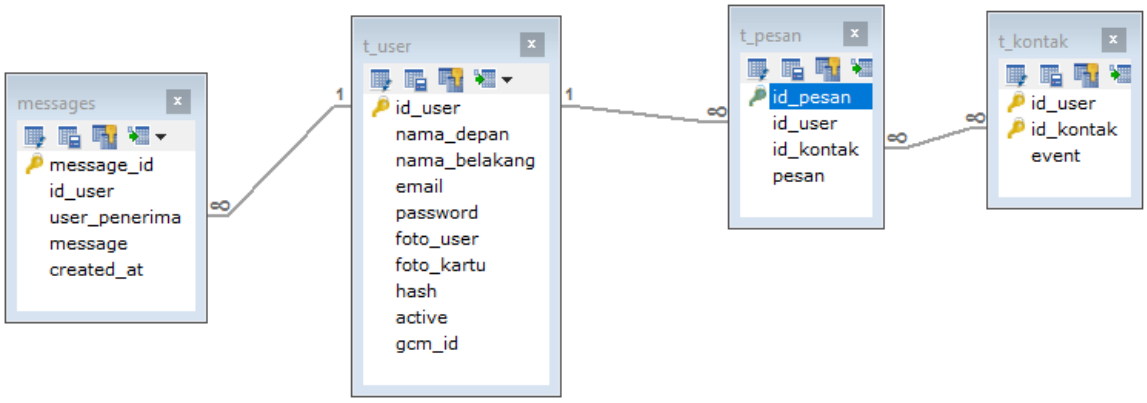
Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh.

3.2.1 Perancangan Data

Berikut adalah perancangan data dari perangkat lunak yang dibangun

1. Diagram Relasi

Berikut adalah gambar 3.41 Diagram relasi dari perangkat lunak yang akan dibangun



Gambar 3.43 Diagram Relasi

2. Struktur Tabel

Berikut adalah struktur table dari perangkat lunak yang dibangun

a) Tabel t_*user*

Berikut adalah Tabel 3.19 Struktur Tabel t_*user* yang mendeskripsikan deskripsi tabel.

Nama File : t_*user*.db

Tempat Penyimpanan : *Harddisk*

Tabel 3.19 Struktur Tabel t_*user*

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_*user*	int	11	primary key	not null, auto increment
nama_depan	varchar	50		not null
Nama_belakang	varchar	50		not null
email	varchar	100		not null

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
password	varchar	50		not null
foto_*user*	varchar	200		not null
foto_kartu	varchar	200		not null
hash	varchar	32		not null
active	int	1		null
gcm_id	varchar	200		null,

b) Tabel t_kontak

Berikut adalah Tabel 3.20 Struktur Tabel t_kontak yang mendeskripsikan deskripsi tabel.

Nama File : t_kontak.db

Tempat Penyimpanan : *Harddisk*

Tabel 3.20 Struktur Tabel t_kontak

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_*user*	int	11	primary key	not null
id_kontak	int	11	primary key	not null
event	varchar	200		not null

c) Tabel t_pesan

Berikut adalah Tabel 3.21 Struktur Tabel t_pesan yang mendeskripsikan deskripsi tabel.

Nama File : t_pesan.db

Tempat Penyimpanan : *Harddisk*

Tabel 3.21 Struktur Tabel t_pesan

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_pesan	int	11	primary key	not null, auto increment
id_*user*	int	11		not null
id_kontak	int	11		not null
pesan	text			

d) Tabel *message*s

Berikut adalah Tabel 3.22 Struktur Tabel *message*s yang mendeskripsikan deskripsi tabel.

Nama File : *message*s.db

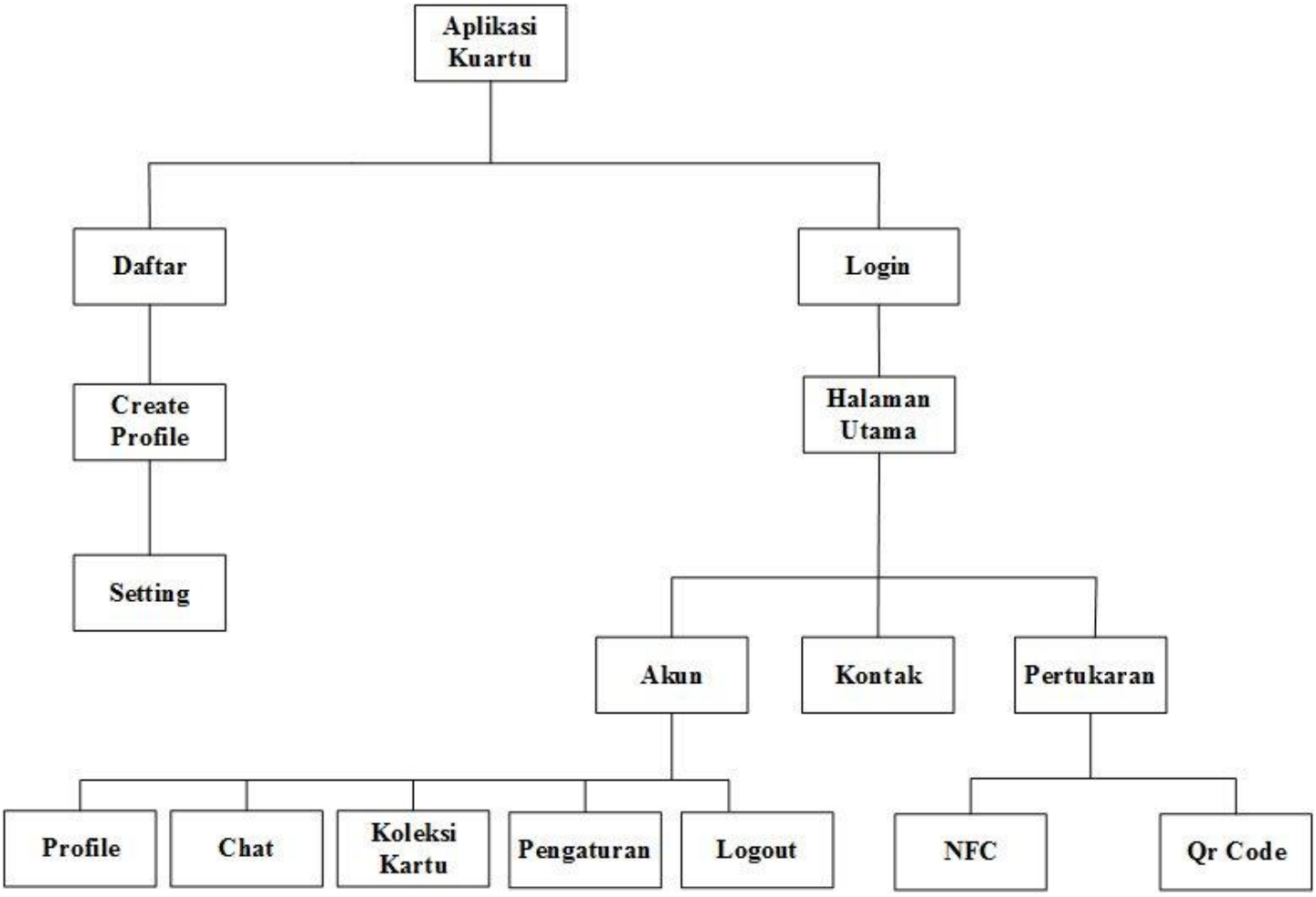
Tempat Penyimpanan : *Harddisk*

Tabel 3.22 Struktur Tabel *message*s

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
*message*_id	int	11	primary key	not null, auto increment
id_*user*	int	11		not null
*user*_penerima	int	11		not null
message	text			not null
created_at	timestamp			not null

3.2.2 Perancangan Arsitektur Menu

Perancangan arsitektur menyediakan *software engineer* satu gambaran dari stuktur menu program atau blueprint dari perangkat lunak yang akan dibuat. Tujuan perancangan ini adalah untuk membangun struktur program secara moduler dan menggambarkan hubungan kendali diantara modul program. Berikut adalah gambaran perancangan arsitektura menu.



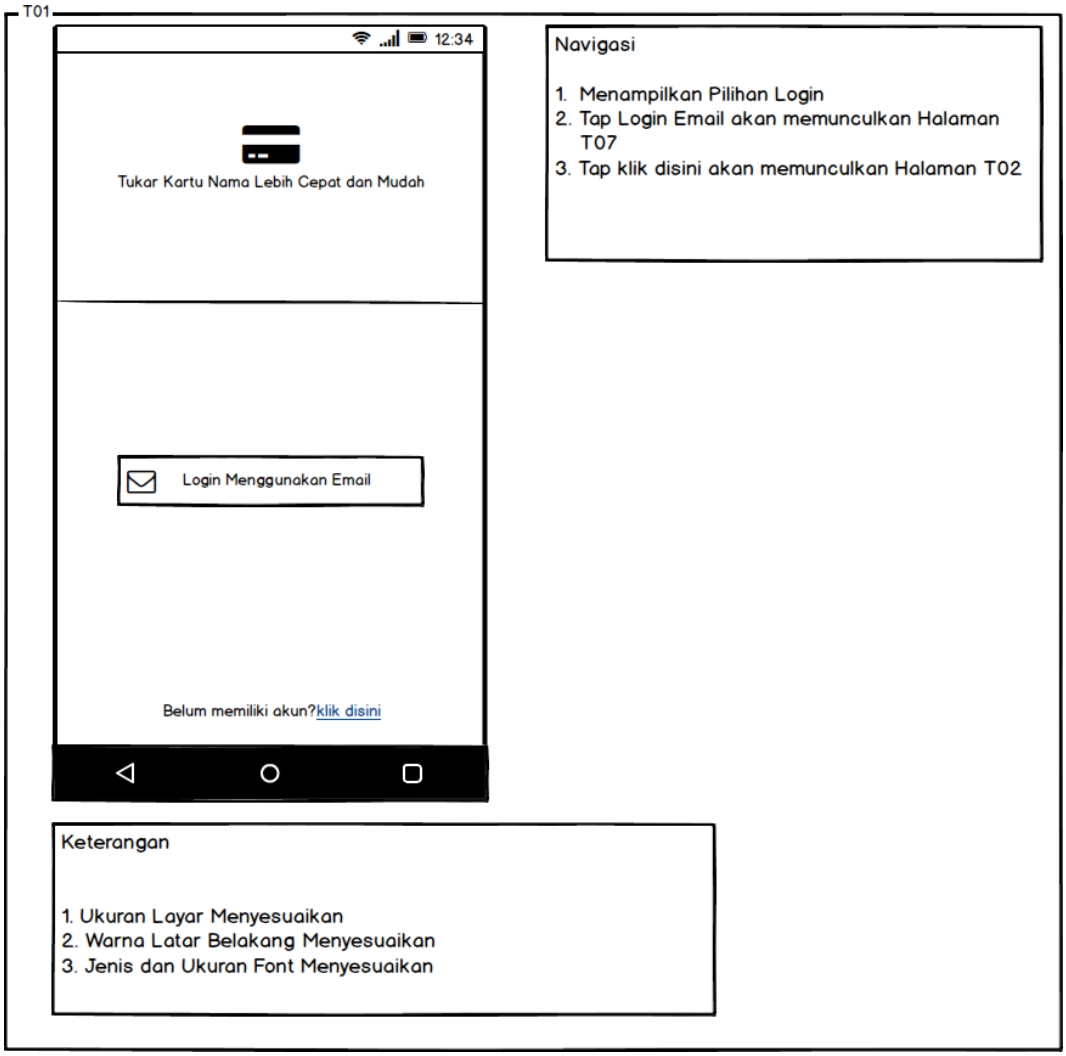
Gambar 3.44 Perancangan Arsitektur Menu

3.2.3 Perancangan Antarmuka

Antarmuka (*interface*) merupakan bentuk tampilan dari program yang tampil pada perangkat *android* yang bertujuan memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan memudahkan pengimplementasikan aplikasi sesuai ukuran layar dan memudahkan dalam pembuatan aplikasi.

1) Halaman *Login*

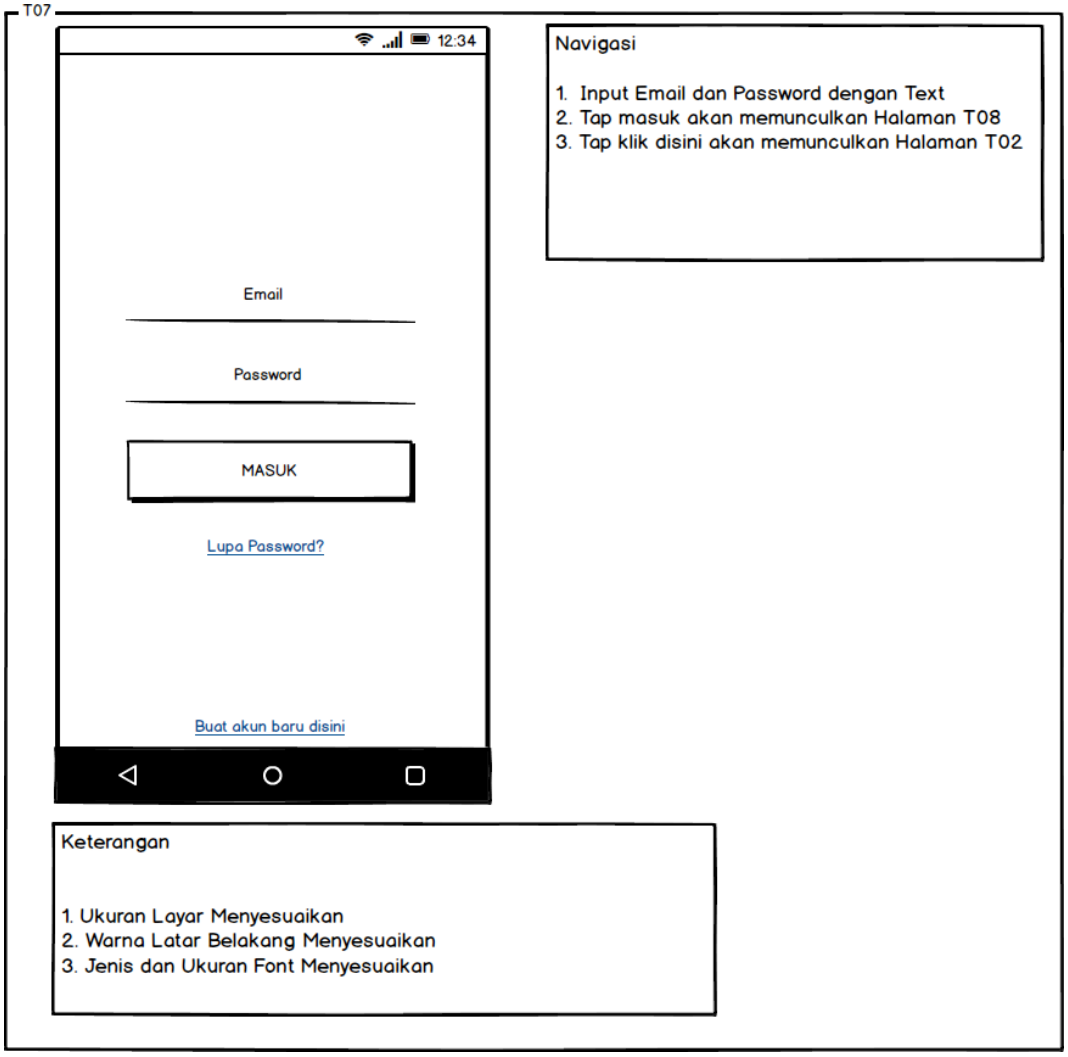
Berikut Gambar 3.45 Antar Muka Halaman *login* adalah yang mengambarkan rancangan antarmuka halaman *login*.



Gambar 3.45 Antarmuka Halaman *Login*

2) Halaman *Login* via *Email*

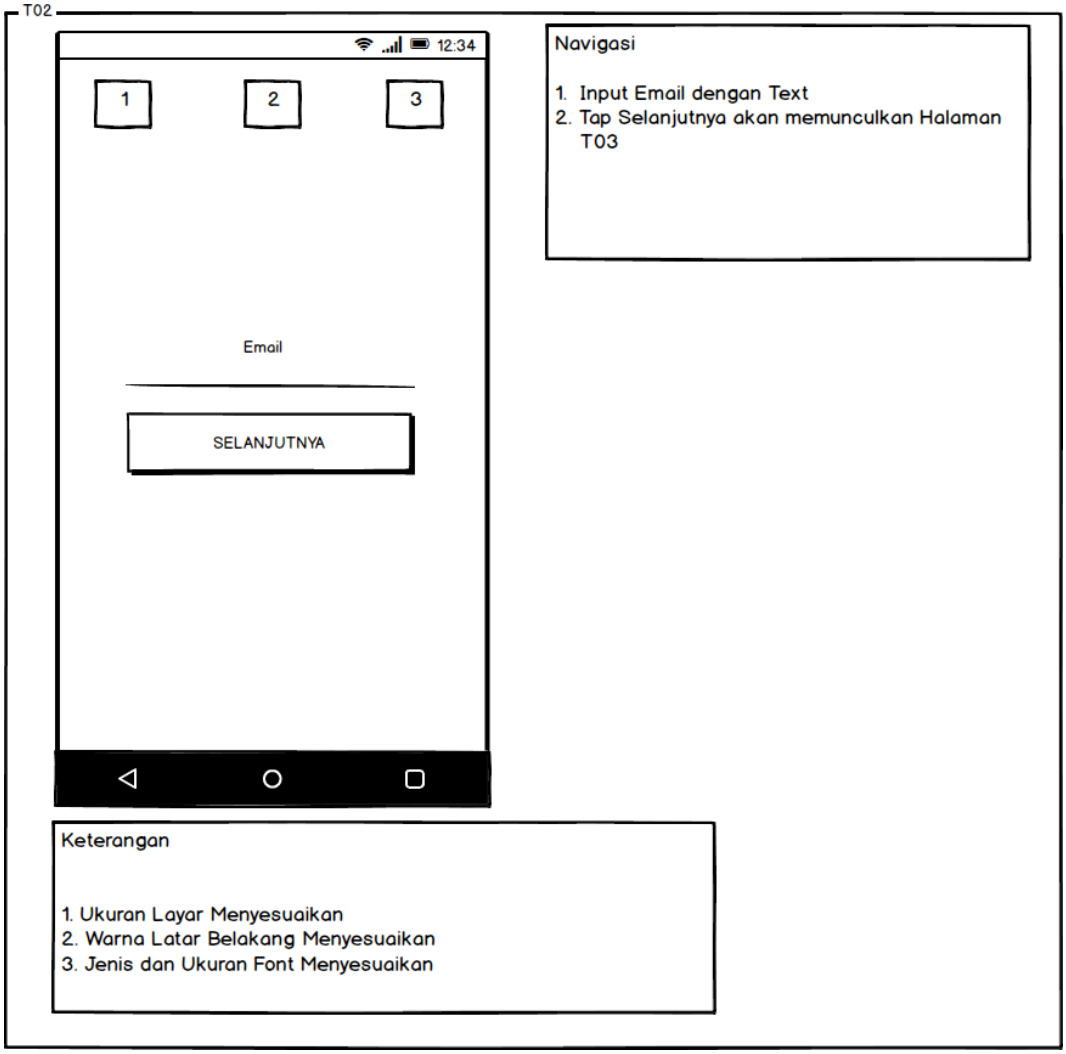
Berikut Gambar 3.46 Antar Muka Halaman *Login* via *email* adalah yang mengambarkan rancangan antarmuka halaman *login* via *email.*



Gambar 3.46 Antarmuka Halaman *Login* Via *Email*

3) Halaman Daftar

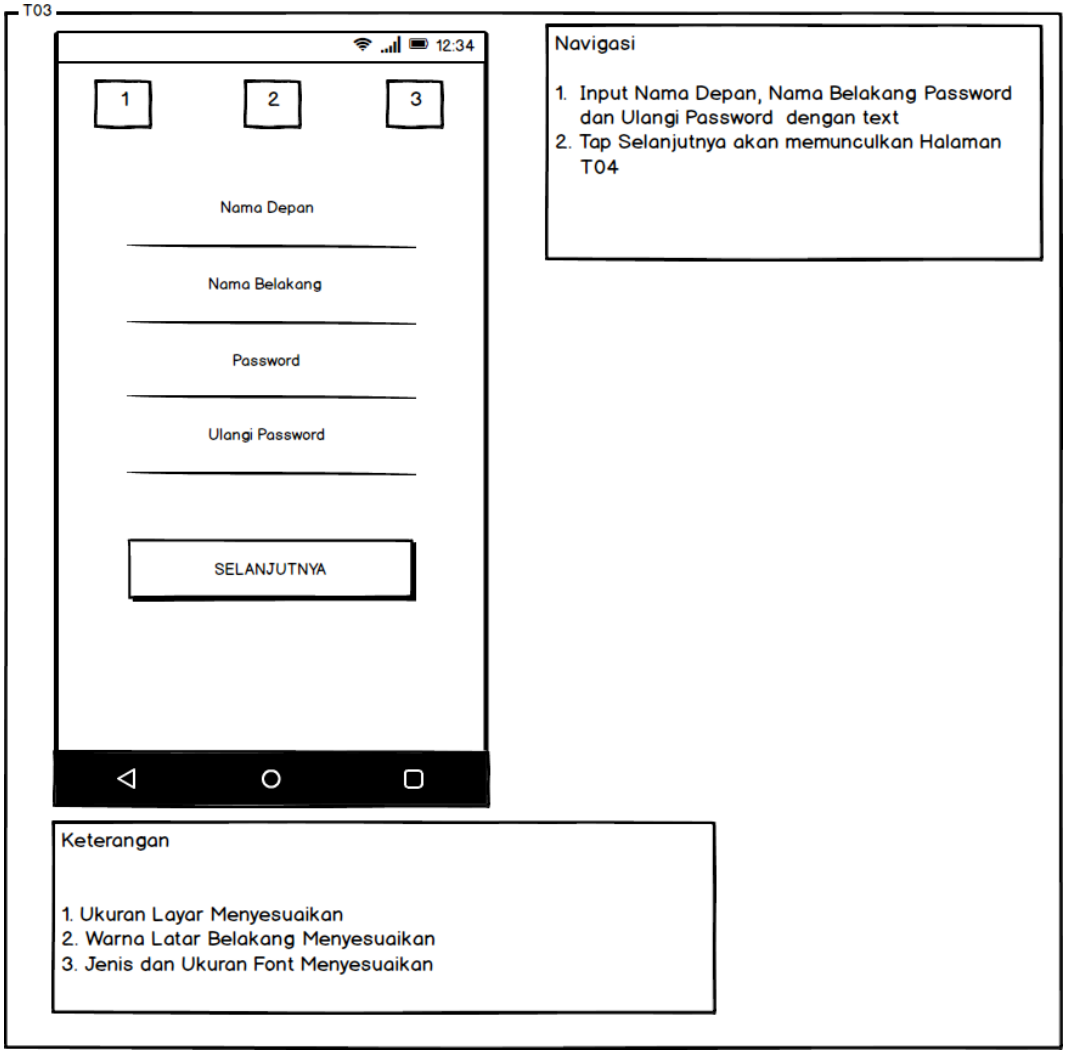
Berikut Gambar 3.47 Antar Muka Halaman Daftar I. Halaman dalam menu ini harus menginputkan *email user*.



Gambar 3.47 Antarmuka Halaman Daftar I

4) Halaman Daftar II

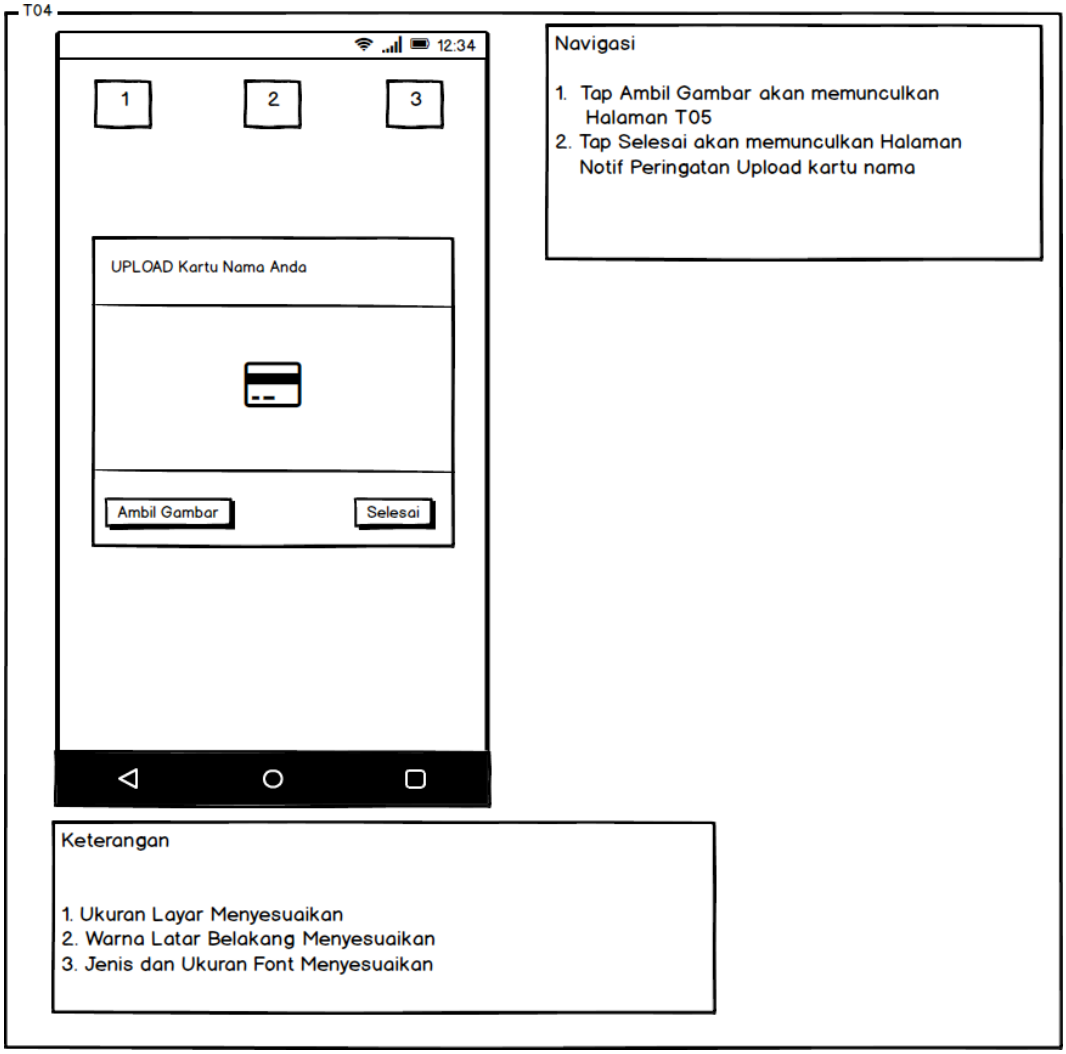
Berikut Gambar 3.48 Antar Muka Halaman Daftar II. Halaman dalam menu ini harus menginputkan nama *user* dan *password* akun.



Gambar 3.48 Antarmuka Halaman Daftar II

5) Halaman Daftar III

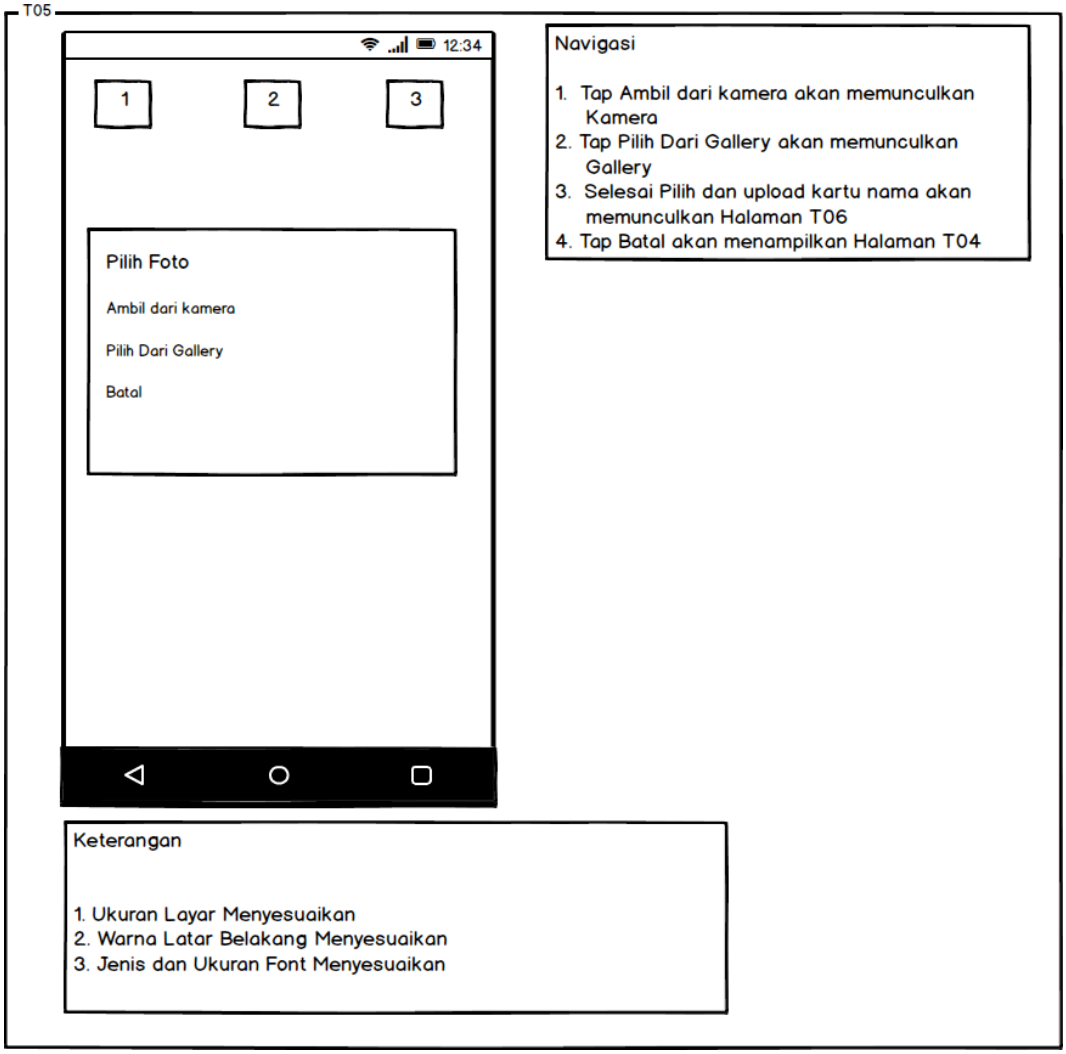
Berikut Gambar 3.49 Antar Muka Halaman Daftar III. Halaman dalam menu ini *user* harus mengupload atau men*capture* kartu nama *user*.



Gambar 3.49 Antarmuka Halaman Daftar III

6) Halaman Daftar IV

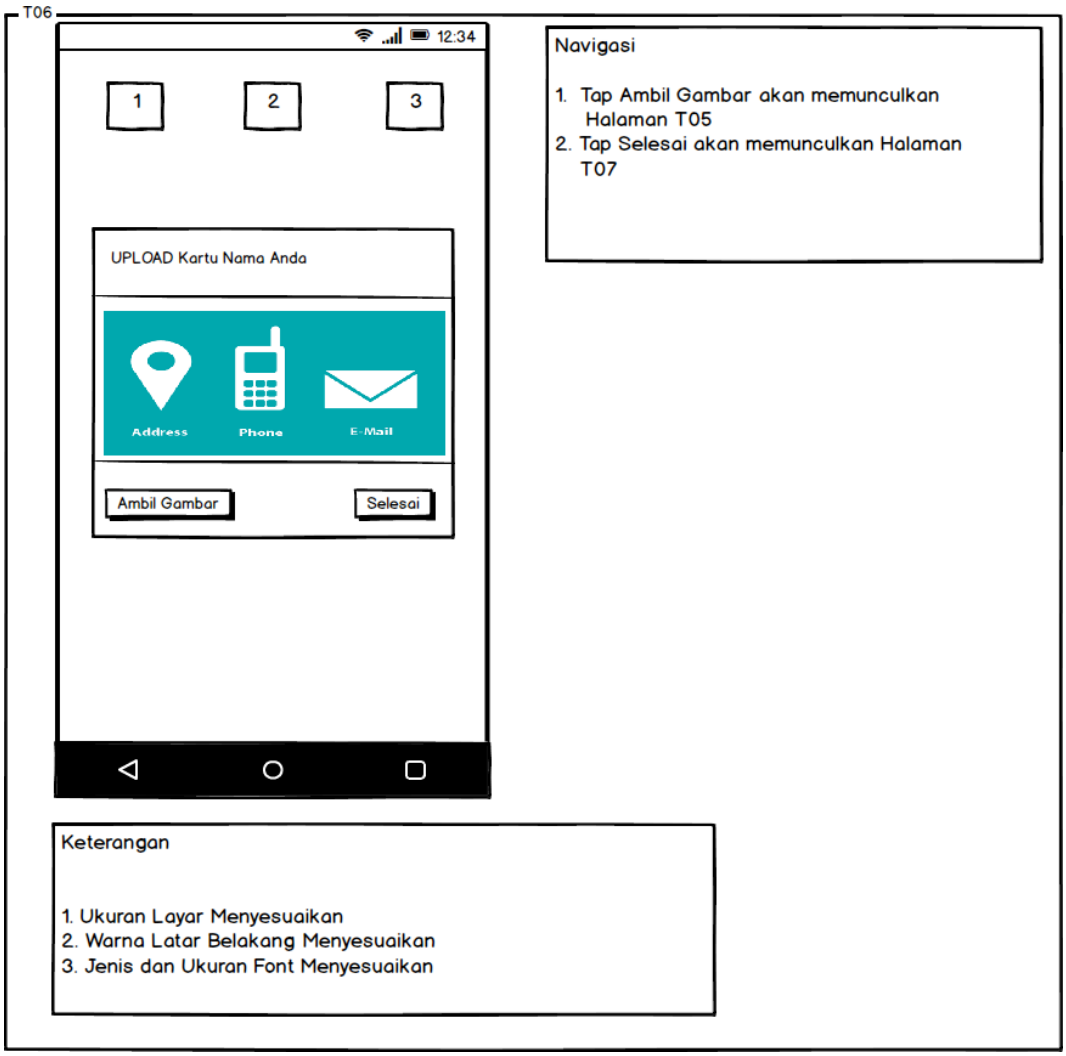
Berikut Gambar 3.50 Antar Muka Halaman Daftar IV. Halaman dalam menu ini *user* memilih kartu nama untuk di *capture* atau ambil dari *gallery*.



Gambar 3.50 Antarmuka Halaman Daftar IV

7) Halaman Daftar V

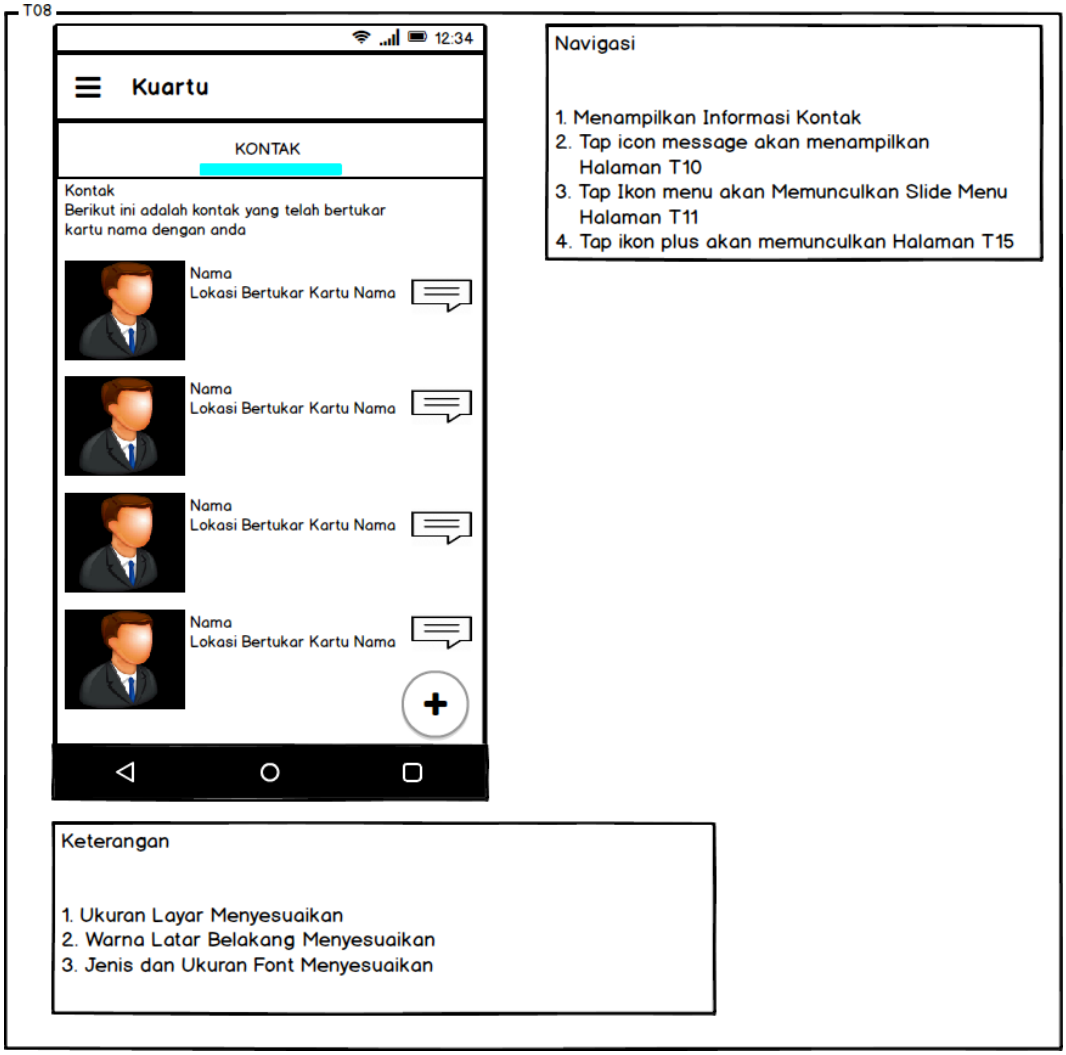
Berikut Gambar 3.51 Antar Muka Halaman Daftar V. Halaman dalam menu ini *user* sudah memilih kartu nama untuk di *capture* atau ambil dari *gallery*



Gambar 3.51 Antarmuka Halaman Daftar V

8) Halaman Kontak

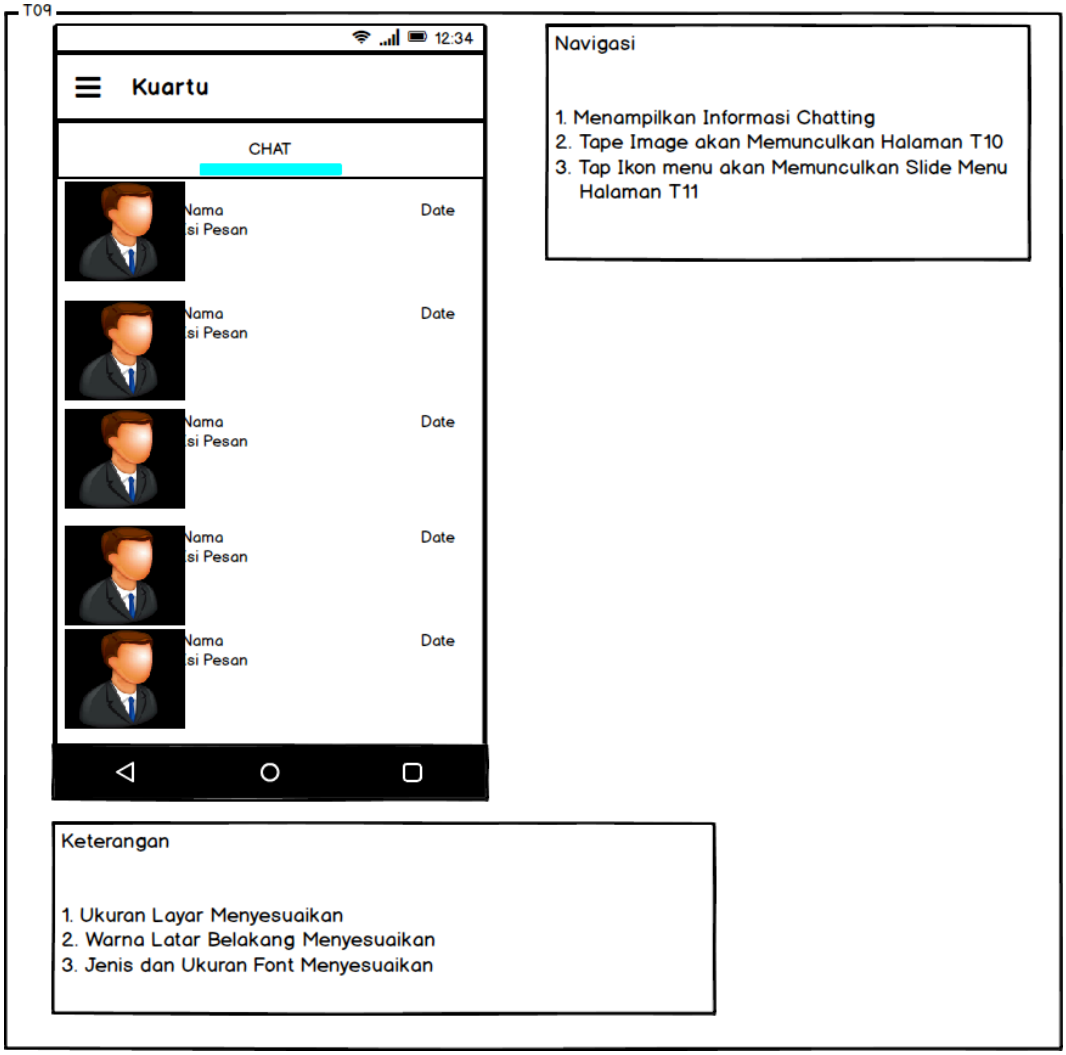
Berikut Gambar 3.52 Antar Muka Halaman Kontak adalah yang mengambarkan rancangan antarmuka halaman kntak



Gambar 3.52 Antarmuka Halaman Kontak

9) Halaman *Chatting*

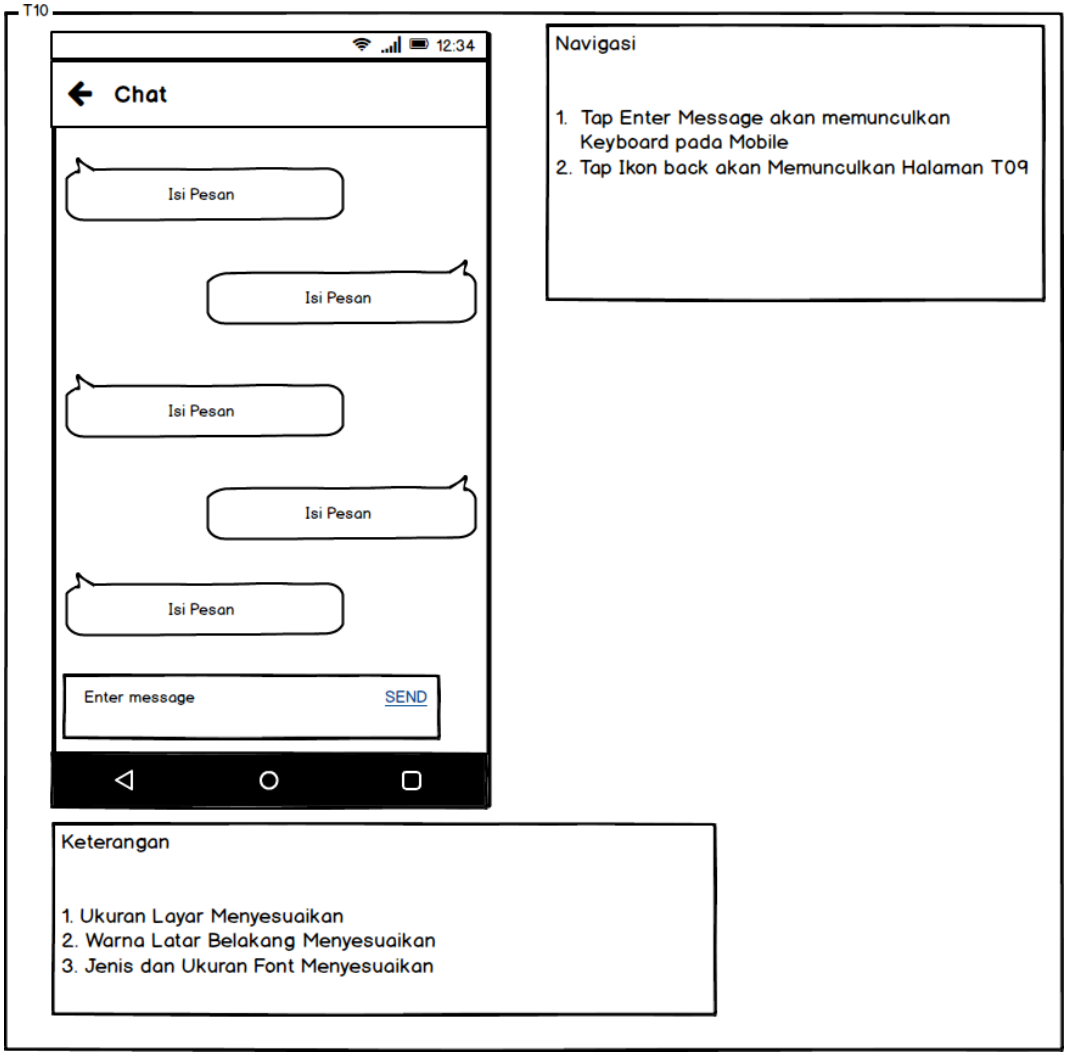
Berikut Gambar 3.53 Antar Muka Halaman *Chatting* adalah yang mengambarkan rancangan antarmuka halaman *chatting*.



Gambar 3.53 Antarmuka Halaman *Chatting*

10) Halaman View *Chatting*

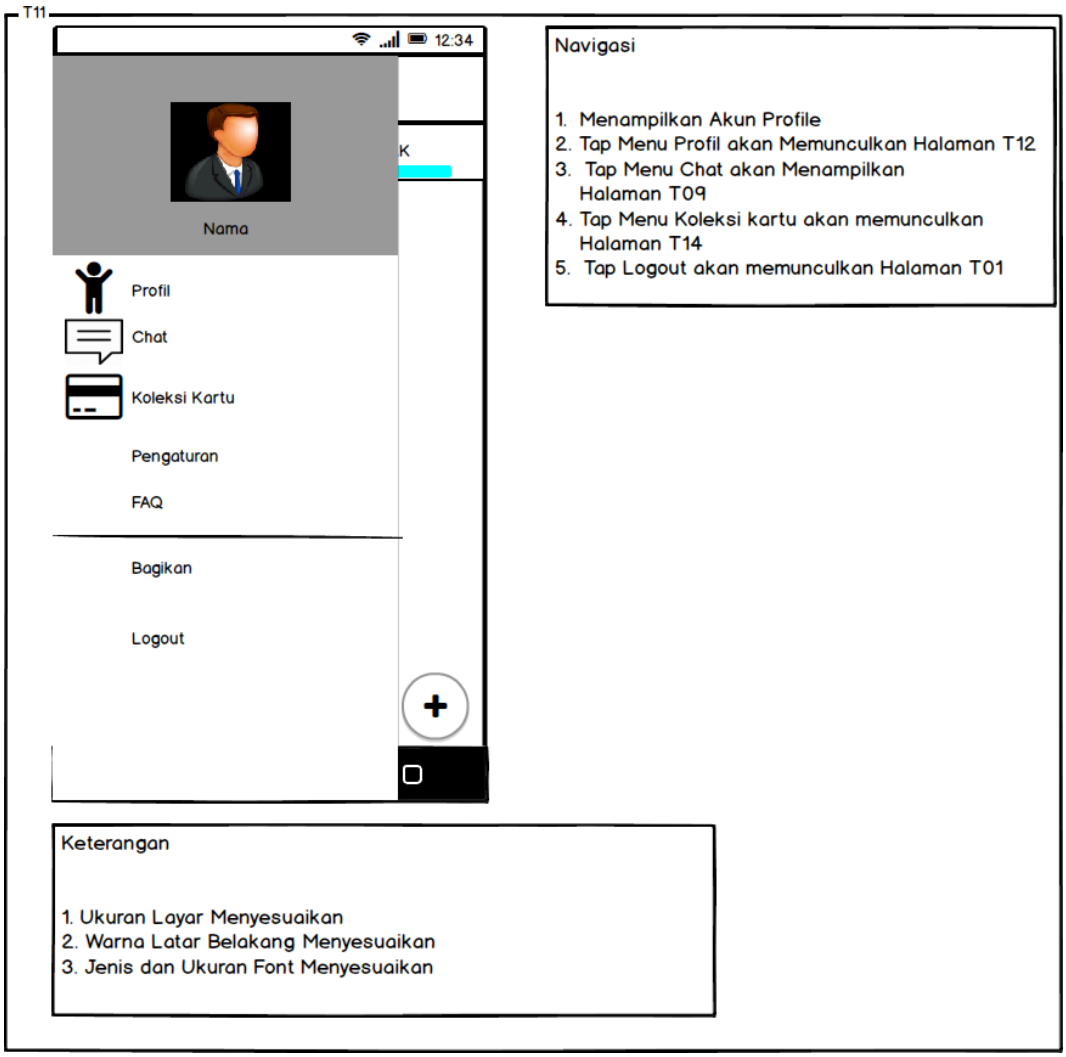
Berikut Gambar 3.54 Antar Muka Halaman View *Chatting* adalah yang mengambarkan rancangan antarmuka View *Chatting*



Gambar 3.54 Antarmuka Halaman *Chatting* I

11) Halaman Akun

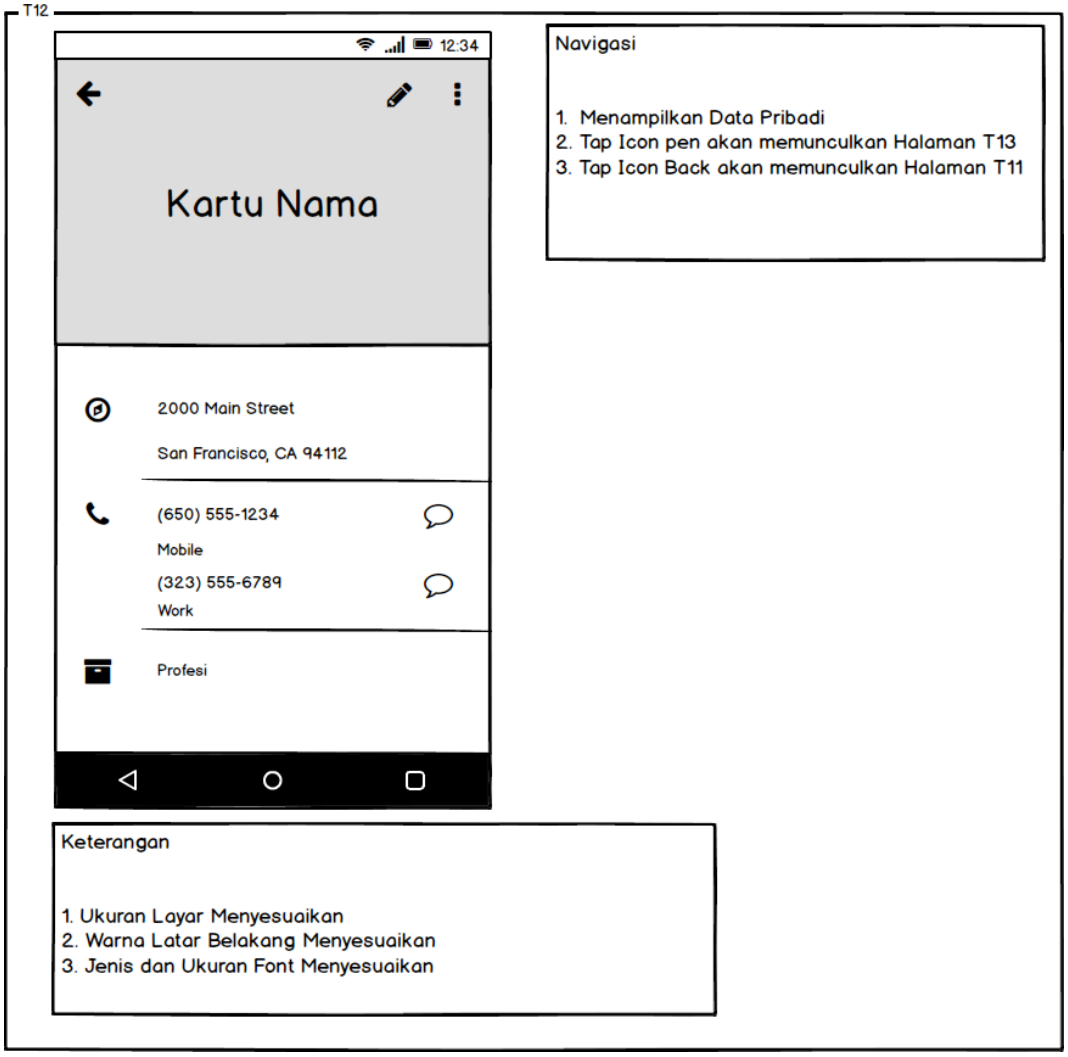
Berikut Gambar 3.55 Antar Muka Halaman Akun adalah yang mengambarkan rancangan antarmuka halaman akun



Gambar 3. 55 Antarmuka Halaman Akun

12) Halaman Profil

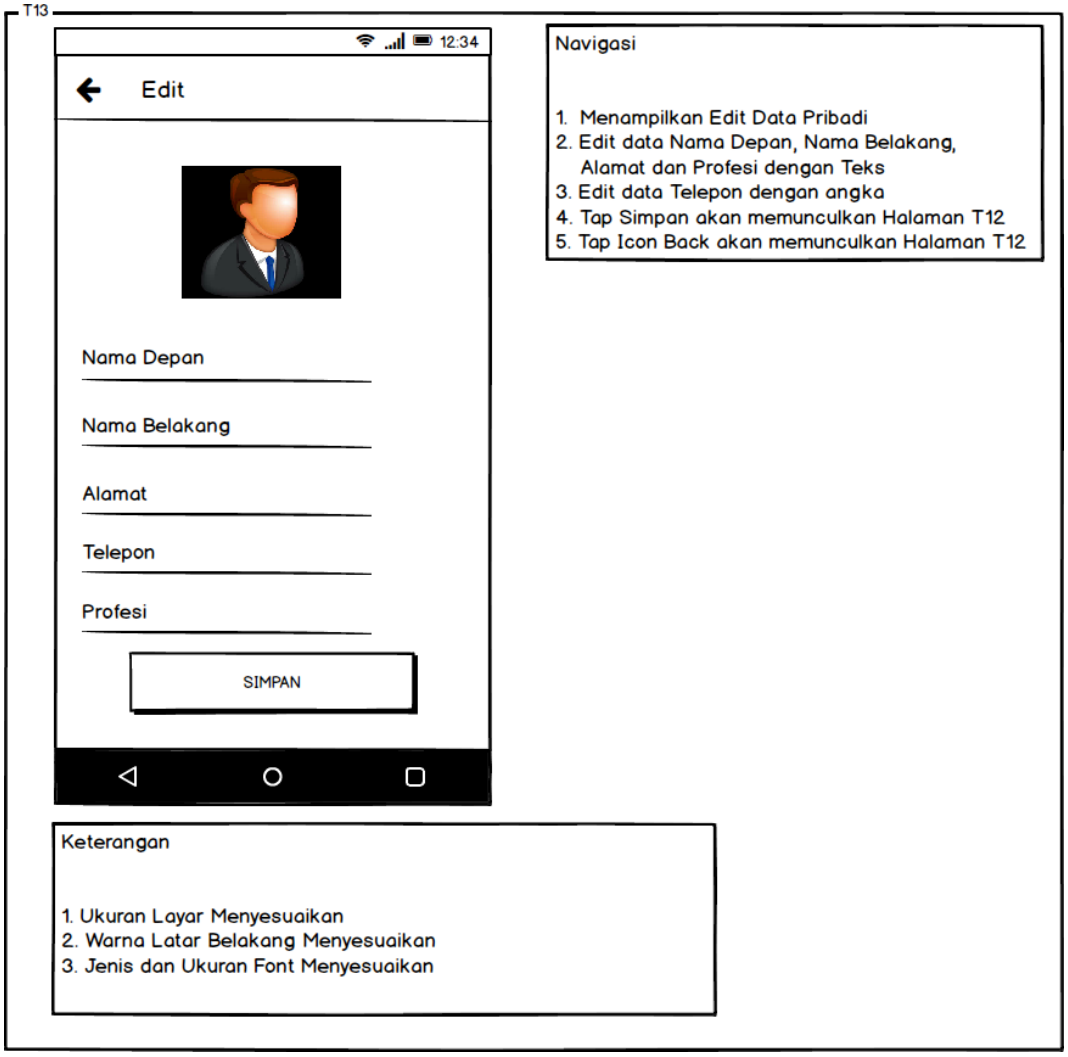
Berikut Gambar 3.56 Antar Muka Halaman *Profile* adalah yang mengambarkan rancangan antarmuka halaman *profile*



Gambar 3.56 Antarmuka Halaman *Profile*

13) Halaman Edit *Profile*

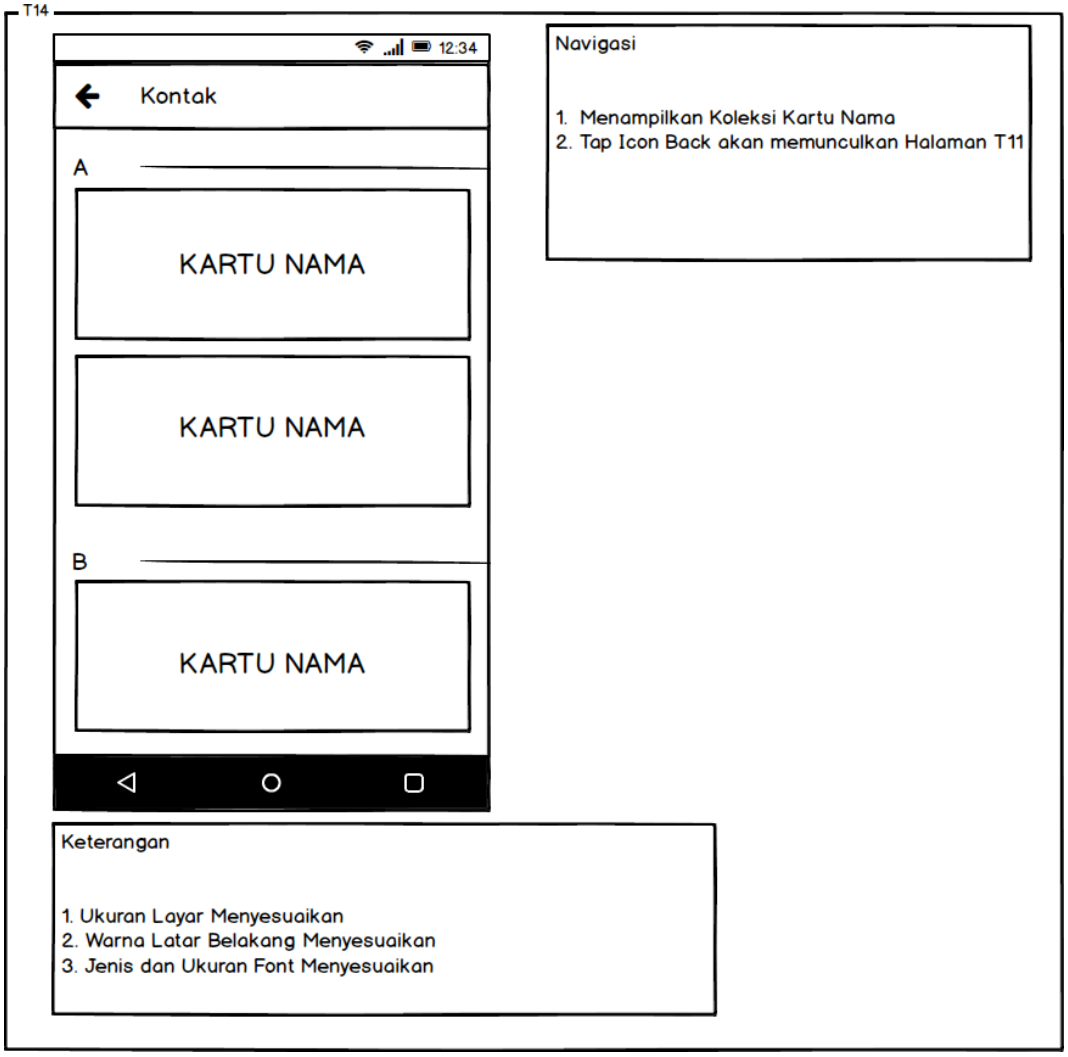
Berikut Gambar 3.57 Antar Muka Halaman Edit *Profile* adalah yang mengambarkan rancangan antarmuka halaman edit *profile*.



Gambar 3.57 Antarmuka Halaman Edit *Profile*

14) Halaman Koleksi Kartu

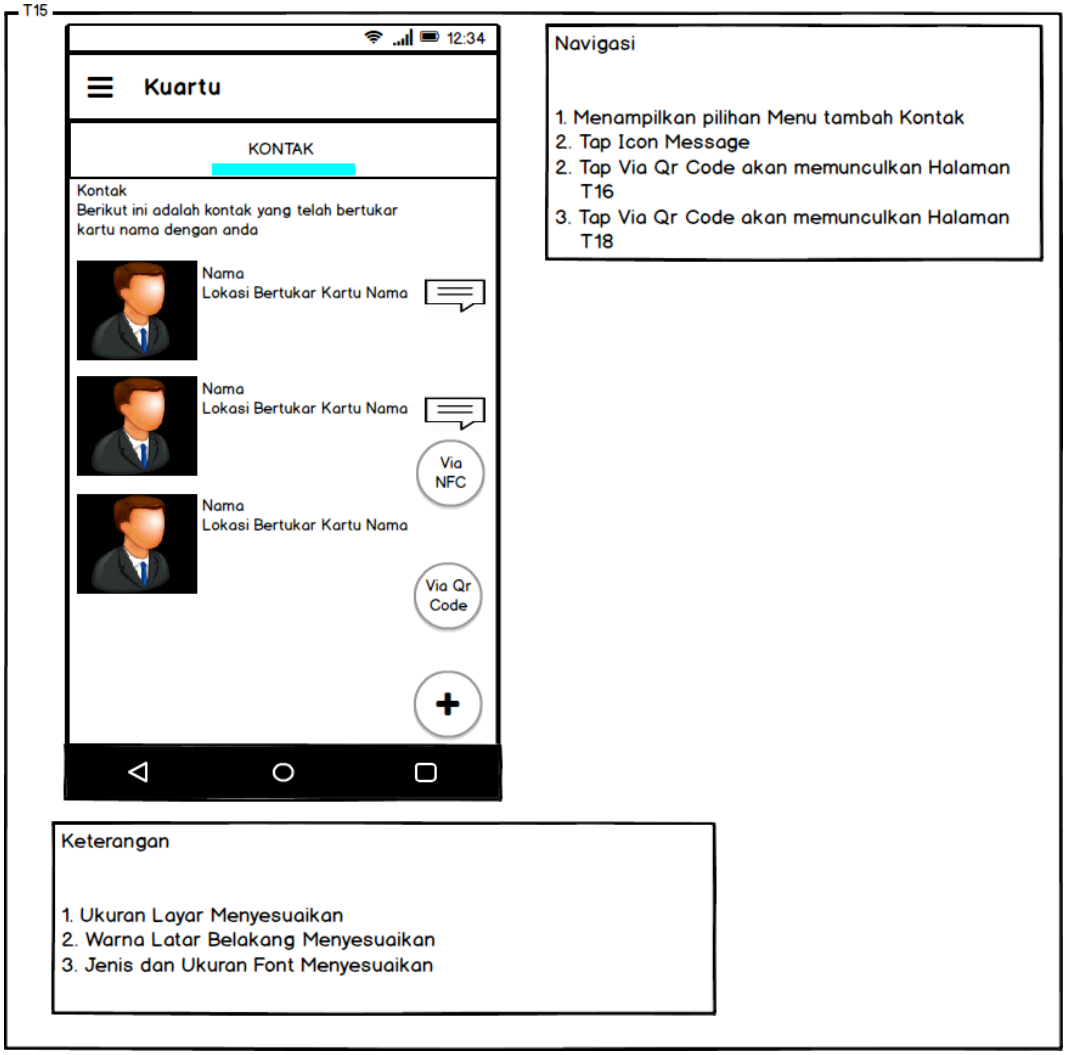
Berikut Gambar 3.58 Antar Muka Halaman Koleksi Kartu adalah yang mengambarkan rancangan antarmuka halaman koleksi kartu.



Gambar 3.58 Antarmuka Halaman Koleksi Kartu

15) Halaman Pertukaran Kartu Nama

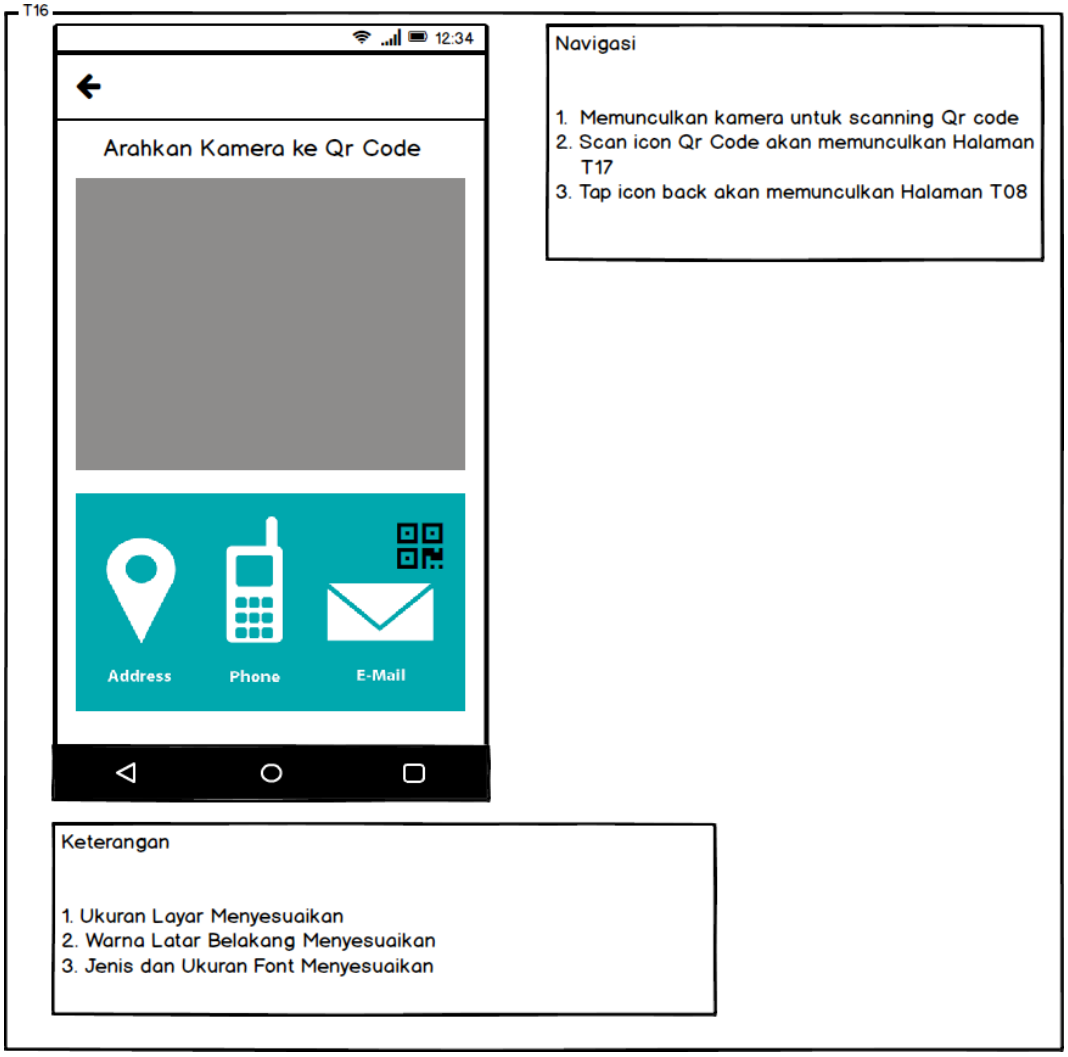
Berikut Gambar 3.59 Antar Muka Halaman Pertukaran Kartu Nama adalah yang mengambarkan rancangan antarmuka halaman pertukaran kartu nama.



Gambar 3.59 Antarmuka Halaman Pertukaran Kartu Nama

16) Halaman Scan *QrCode*

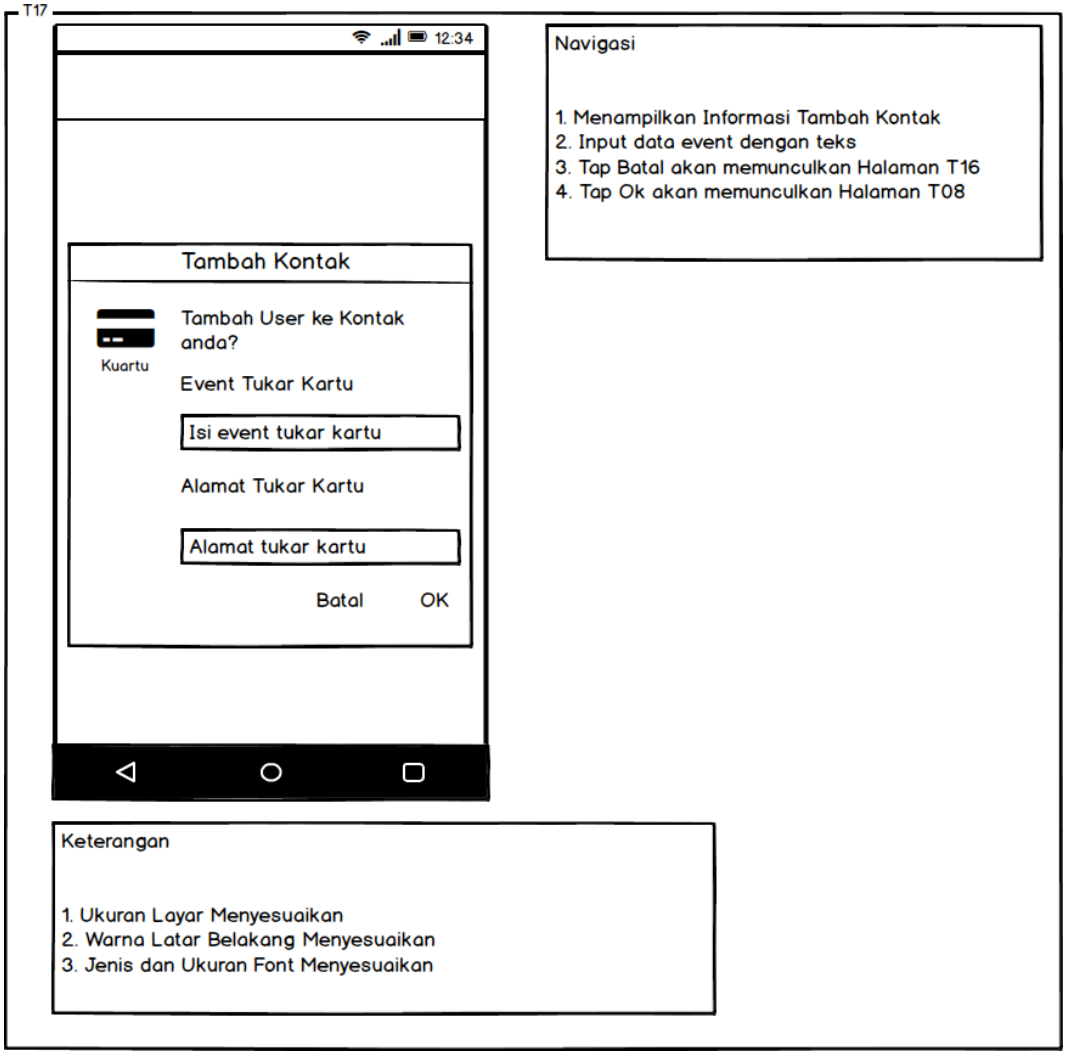
Berikut Gambar 3.60 Antar Muka Halaman Scan *QrCode* adalah yang mengambarkan rancangan antarmuka halaman scan *QrCode*.



Gambar 3.60 Antarmuka Halaman Scan *QrCode*

17) Halaman Tambah Kontak

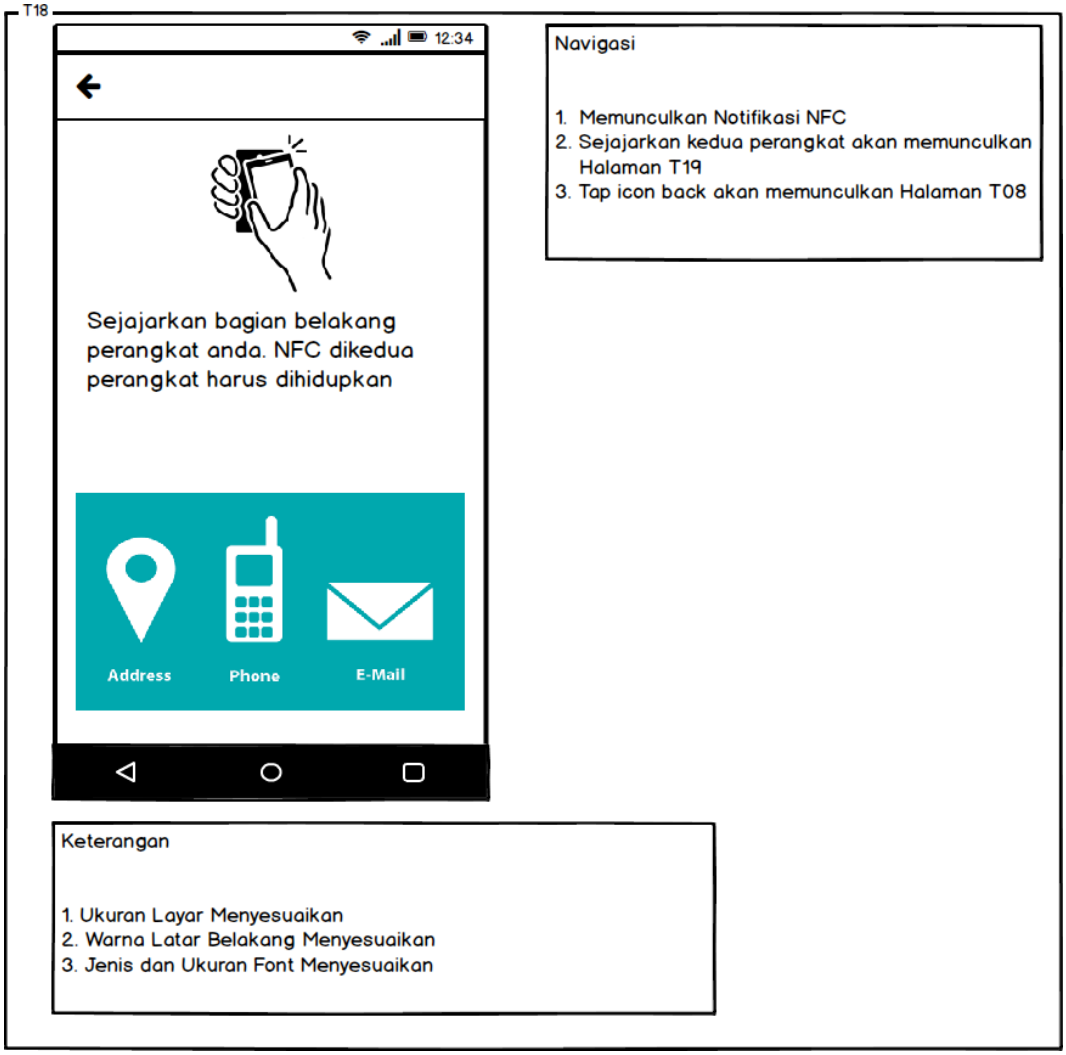
Berikut Gambar 3.61 Antar Muka Halaman Tambah Kontak adalah yang mengambarkan rancangan antarmuka halaman tambah kontak.



Gambar 3.61 Antarmuka Halaman Tambah Kontak

18) Halaman NFC

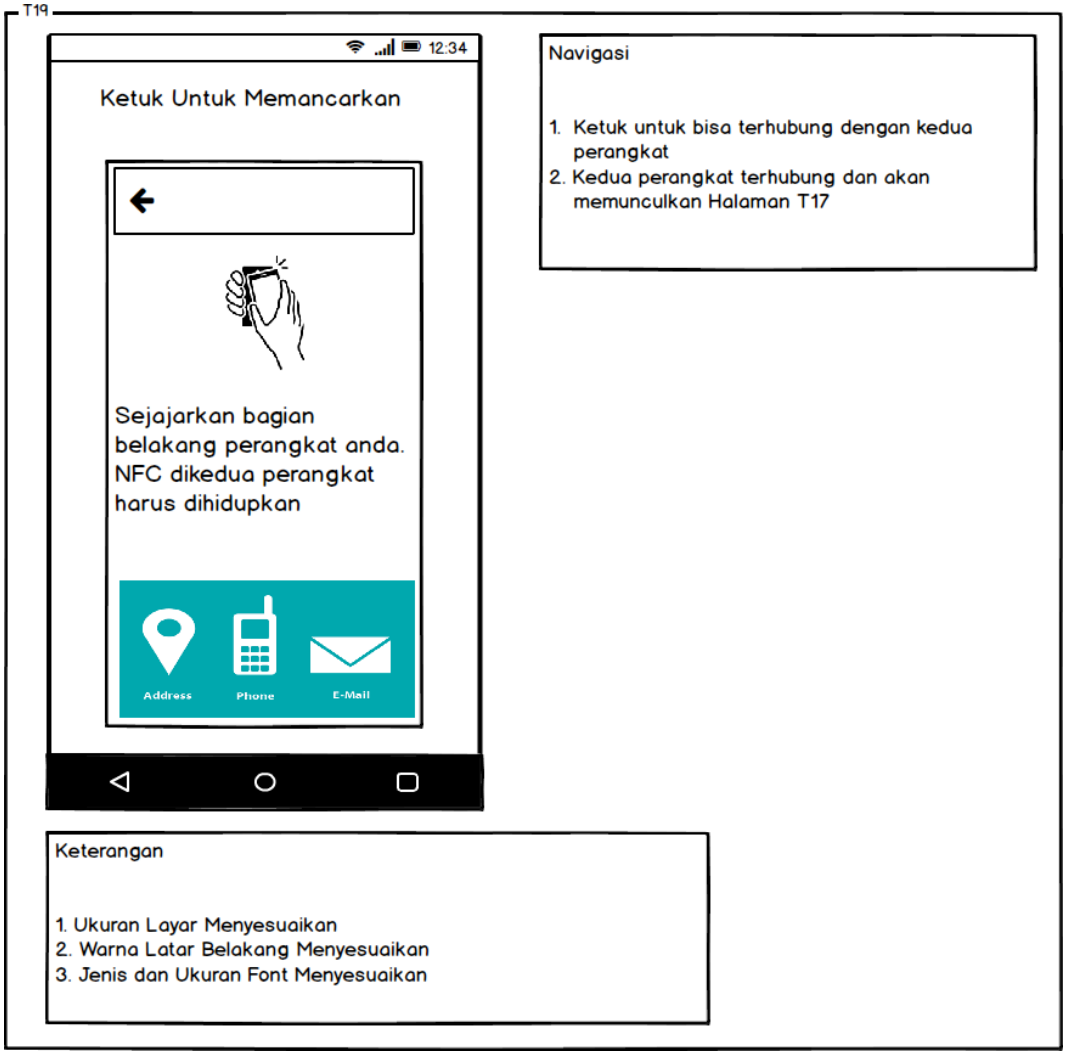
Berikut Gambar 3.62 Antar Muka Halaman NFC adalah yang mengambarkan rancangan antarmuka halaman NFC



Gambar 3.62 Halaman Antarmuka NFC

19) Halaman Proses NFC

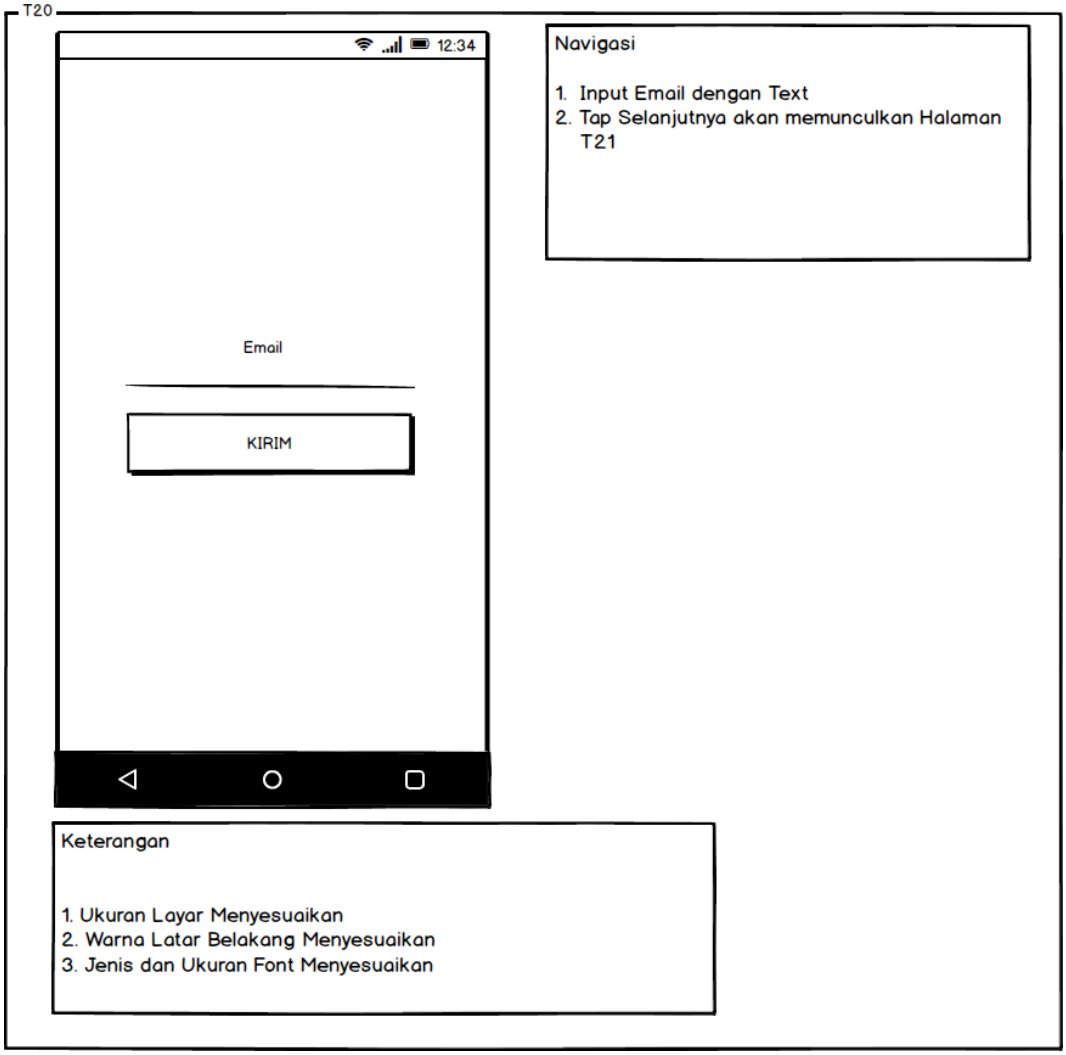
Berikut Gambar 3.63 Antar Muka Halaman Proses NFC adalah yang mengambarkan rancangan antarmuka halaman Proses NFC.



Gambar 3.63 Halaman Antarmuka Proses NFC

20) Halaman Lupa *Password*

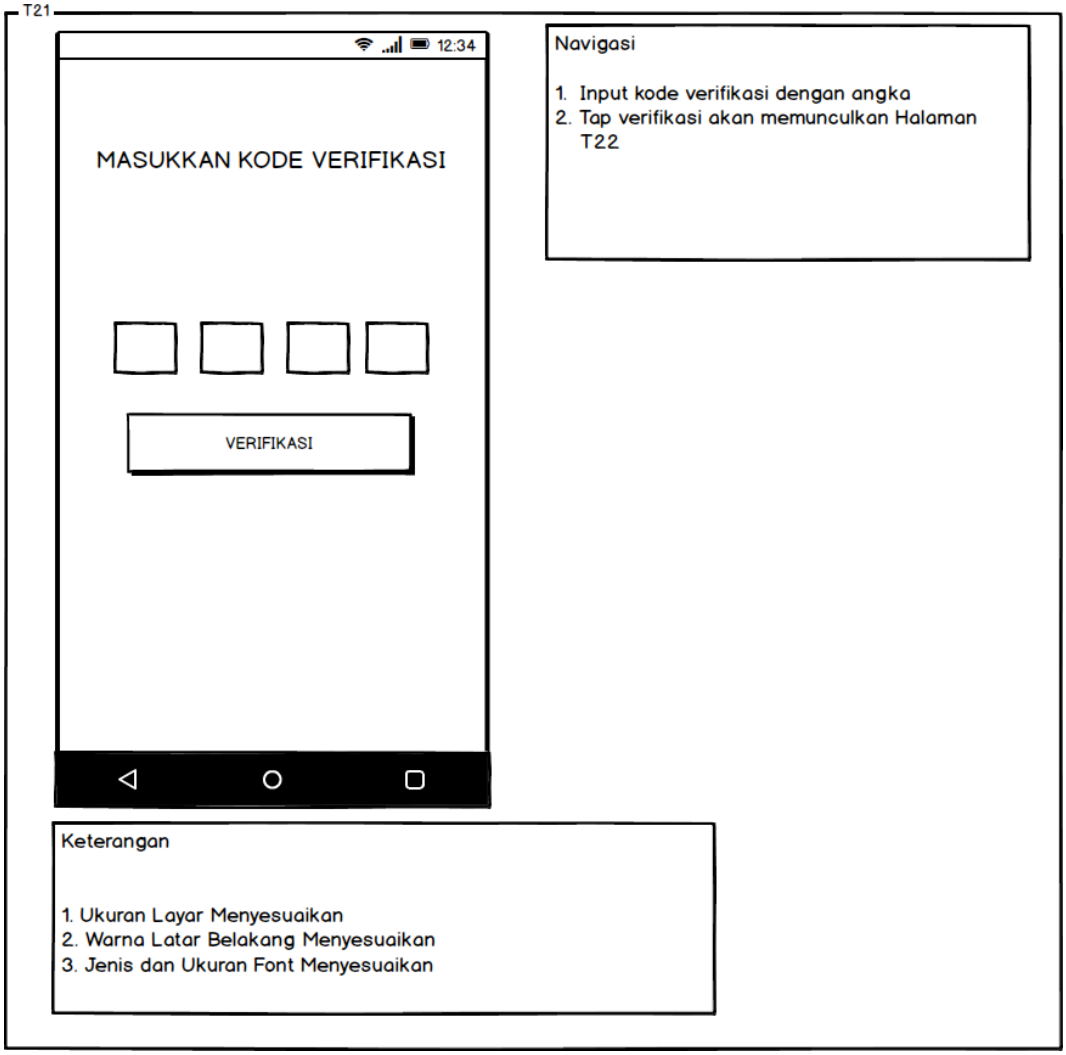
Berikut Gambar 3.64 Antar Muka Halaman Lupa *Password* adalah yang mengambarkan rancangan antarmuka halaman Lupa *Password*.



Gambar 3.64 Antarmuka Halaman Lupa *Password*

21) Halaman Lupa *Password* I

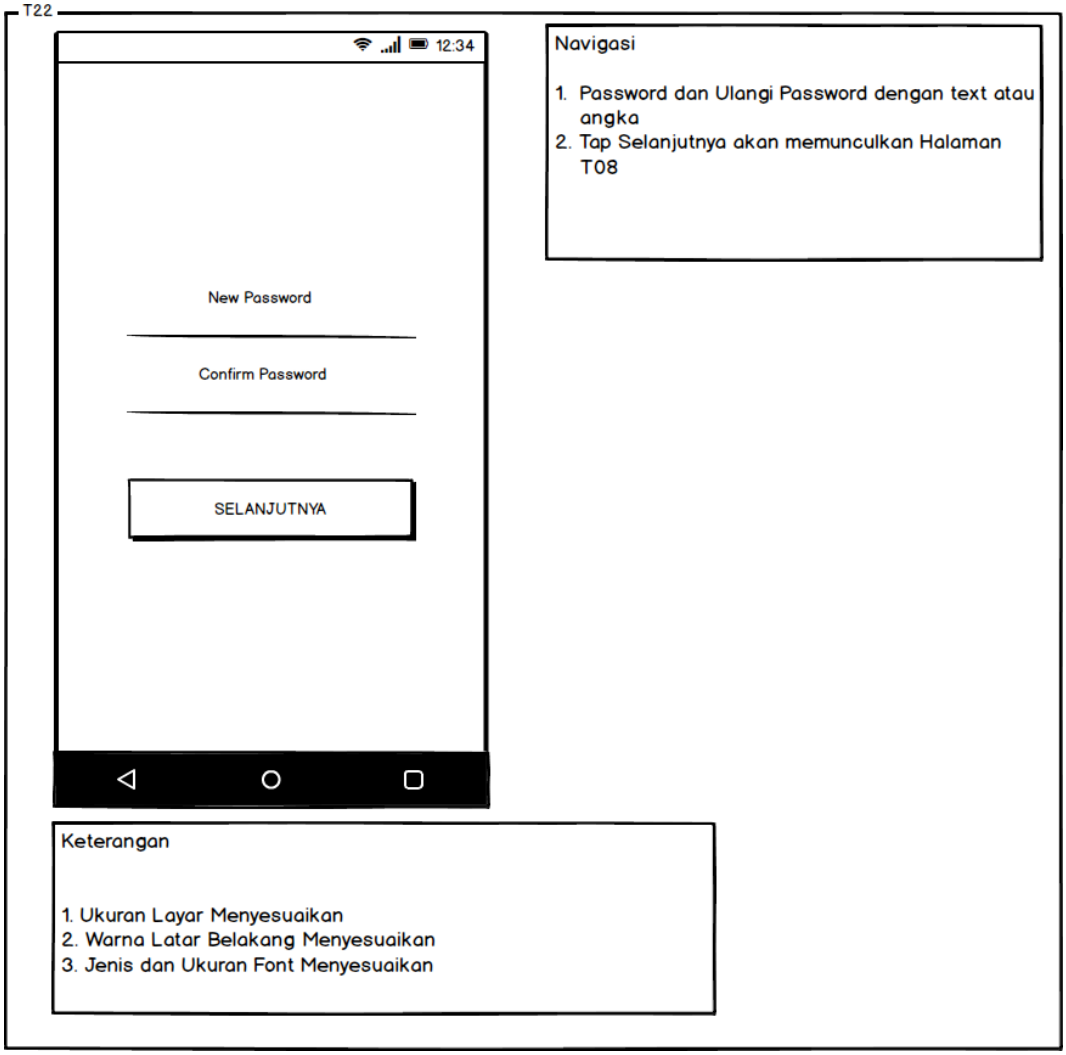
Berikut Gambar 3.65 Antar Muka Halaman Lupa *Password* I adalah yang mengambarkan rancangan antarmuka halaman Lupa *Password* I.



Gambar 3.65 Antarmuka Halaman Lupa *Password* I

22) Halaman Lupa *Password* II

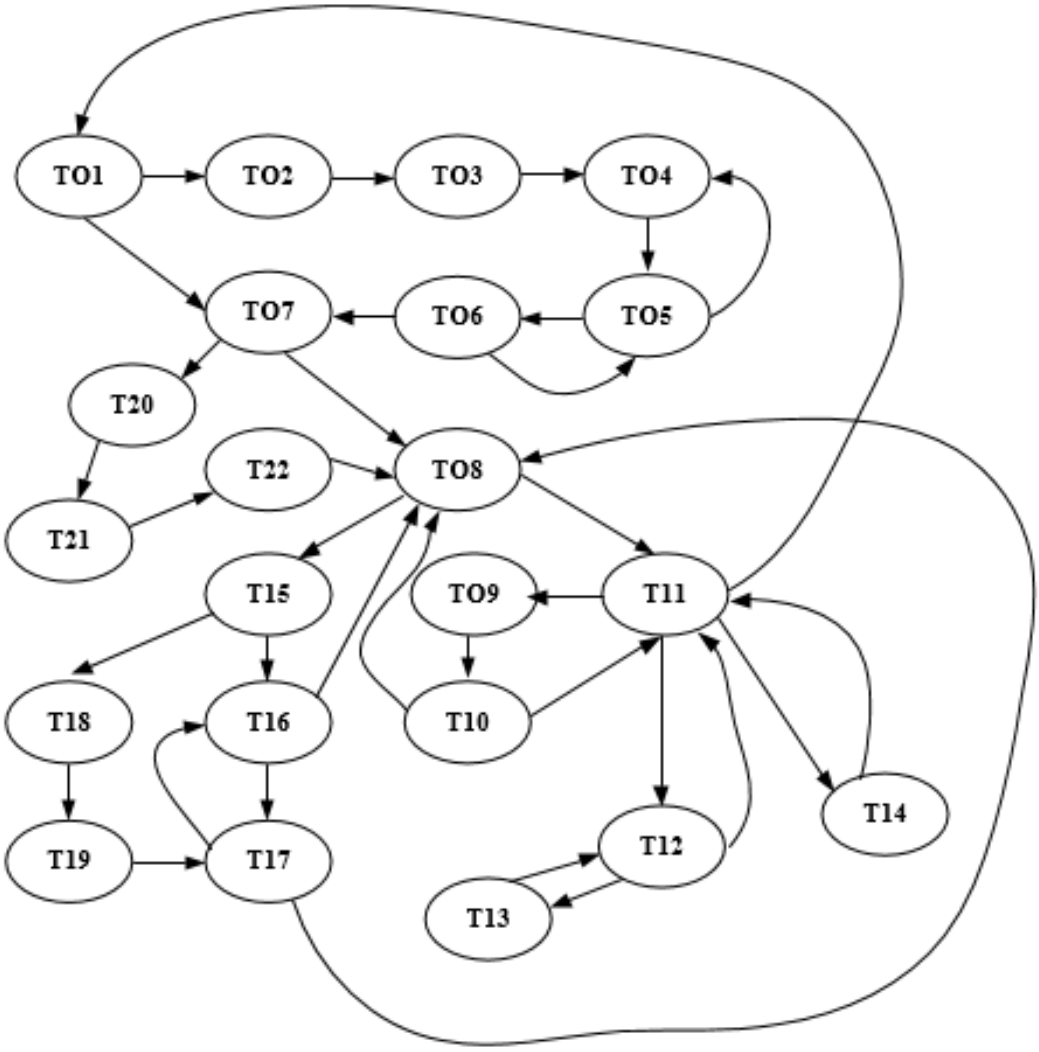
Berikut Gambar 3.66 Antar Muka Halaman Lupa *Password* II adalah yang mengambarkan rancangan antarmuka halaman Lupa *Password* II.



Gambar 3.66 Antarmuka Halaman Lupa *Password* II

3.2.4 Jaringan Semantik

Jaringan Semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar objek (lingkaran) dan garis yang menggambarkan formasi antarobjek tersebut. Adapun jaringan semantik untuk perangkat yang dibangun dapat dilihat pada Gambar 3.67 Jaringan Semantik



Gambar 3.67 Jaringan Semantik