

BAB III

METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Tempat yang digunakan penulis dalam melakukan penelitian ini adalah Boediman.Jr Barbershop Yogyakarta yang beralamat Caturtunggal, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281, Indonesia. Adapun penelitian ini dilaksanakan mulai bulan Juni sampai Juli 2017.

3.2 Alat dan Bahan

Penelitian membutuhkan berbagai perangkat untuk mendukung berjalannya perancangan dan implementasi aplikasi antrian berbasis *android*, antara lain:

3.2.1 Perangkat Keras

yarat minimum Personal computer atau Laptop yang digunakan untuk membangun aplikasi ini adalah:

1. *Processor* intel core i5.
2. RAM 4 GB.
3. *Harddisk* 500 GB.
4. Monitor, *keyboard*, dan *mouse*.

3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam mengembangkan aplikasi antrian boediman.jr barbershop antara lain:

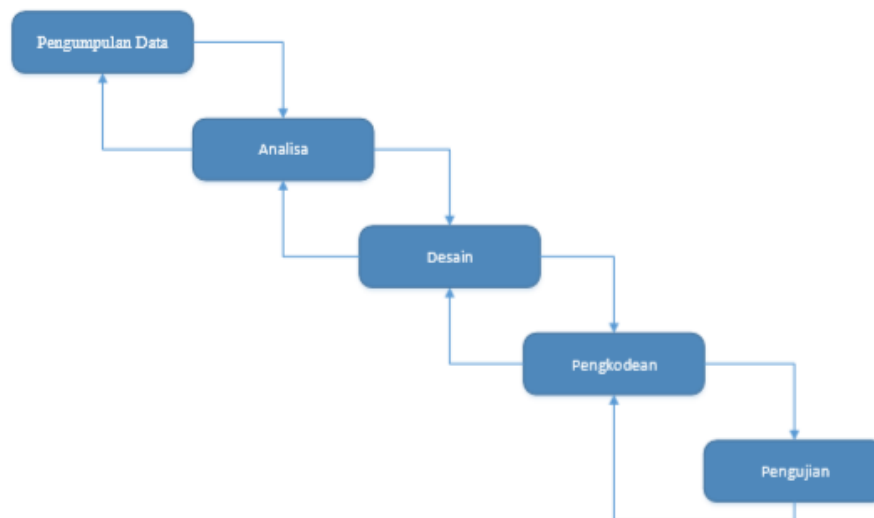
1. *Operating System Windows 10*

2. *Android Studio.*
3. *MySQL Workbench.*
4. *NetBeans.*
5. *Microsoft Visio.*
6. *Web browser: Mozilla firefox dan Internet Explorer.*
7. *Text editor: Notepad++.*

3.3 Alur Penelitian

Tujuan dari penyusunan alur penelitian yang penulis lakukan adalah untuk membantu penulis dalam penyusunan prosedur-prosedur di dalam penelitian sehingga peneliti penelitian dapat dilaksanakan dan diselesaikan dengan baik dan tepat waktu.

Dalam Pembuatan Aplikasi Antrian Boediman-Jr Barbershop Berbasis *Android* yaitu menggunakan model *Waterfall* atau *Classic Life Cycle*, dimana model *Waterfall* merupakan salah satu metode dalam SDLC yang mempunyai ciri khas pengerjaan setiap fase dalam *Waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya. Artinya fokus terhadap masing-masing fase dapat dilakukan maksimal karena tidak adanya pengerjaan yang sifatnya parallel. Model *waterfall* akan digambarkan seperti pada gambar 3.1.



Gambar 3.1 Metode Waterfall.

a Tahap Pengumpulan Data

Pengumpulan data merupakan usaha yang dilakukan untuk memperoleh informasi dalam bentuk data yang dibutuhkan dalam penelitian.

b Tahap Analisis Kebutuhan

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau *study literatur*.

c Tahap Desain

Proses desain akan menterjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding.

d Tahap Pengkodean atau Penulisan *Code* Program

Coding merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan

meterjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan *computer* akan dimaksimalkan dalam tahapan ini.

e Tahap Pengujian Program

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

3.4 Metode Pengumpulan Data

3.4.1 Observasi

Observasi yang dilakukan dalam penelitian bertujuan untuk mengetahui proses kerja antrian yang sedang berjalan pada Boediman.Jr Barbershop. Observasi yang digunakan adalah jenis observasi nonpartisipan. Dalam proses observasi ini, peneliti hanya mengamati tanpa terlibat langsung dalam proses yang sedang diamati. Pada penelitian ini peneliti mendapatkan data dengan mengamati objek yang akan diteliti di tempat penelitian yaitu Boediman.Jr Barbershop.

3.4.2 Studi Literatur

Proses *studi literatur* dilakukan dengan mempelajari bahan-bahan *literatur* tentang pembuatan aplikasi antrian berbasis *android* yang dapat di dapat melalui *internet*, jurnal, *blog* dan buku-buku referensi. Sehingga bahan-bahan *literatur* yang telah dipelajari dapat di implementasikan untuk membuat aplikasi antrian yang baik. Selain itu penulis juga mempelajari berbagai teknologi yang sesuai dengan

kebutuhan sistem dan kebutuhan pengguna yang menjadi acuan pembangunan sistem yang lebih baik.

3.5 Analisis Masalah

Tahap analisis merupakan tahapan tentang merumuskan permasalahan yang timbul dan solusi apa yang dapat menyelesaikan permasalahan tersebut. Pada penelitian ini permasalahan yang timbul bagaimana membuat sebuah aplikasi antrian boerdiman.Jr Barbershop berbasis *Android*.

Fokus penelitian ini adalah bagaimana sistem antrian dapat terkoneksi secara real time sehingga dapat memberikan pelayanan lebih dan konsumen pun dapat dengan mudah melihat dan memantau informasi antrian menggunakan *smartphone* mereka. Dalam hal ini *smartphone* yang digunakan berbasis *android*.

3.6 Analisis Kebutuhan

3.6.1 Analisis Kebutuhan Fungsional

Berikut ini adalah kebutuhan-kebutuhan yang memiliki keterkaitan langsung dengan sistem. Kebutuhan fungsional dari aplikasi ini meliputi:

1. Fasilitas *login* bagi pelanggan agar dapat mengakses aplikasi antrian.
2. Dapat menampilkan informasi nomor antrian yang sedang berjalan.
3. Dapat menampilkan informasi nomor antrian yang dilayani.
4. Dapat menampilkan informasi nomor sisa antrian.
5. Pelanggan dapat mengambil nomor antrian.
6. Dapat menampilkan nomor yang sudah diambil pelanggan.
7. Pelanggan mendapatkan notifikasi.

3.6.2 Analisis Kebutuhan Non-Fungsional

Berikut ini adalah kebutuhan yang tidak secara tidak langsung terkait fitur tertentu di dalam sistem. Kebutuhan non-fungsional dari aplikasi ini meliputi:

1. Kemanan: Mengelola sistem data dalam mengganti, menambah, dan menghapus data konten hanya dapat dilakukan oleh *developer*.
2. Aksesibilitas: Aplikasi Antrian Boediman-Jr Barbershop Berbasis *Android* ini dapat diakses oleh seluruh kalangan.

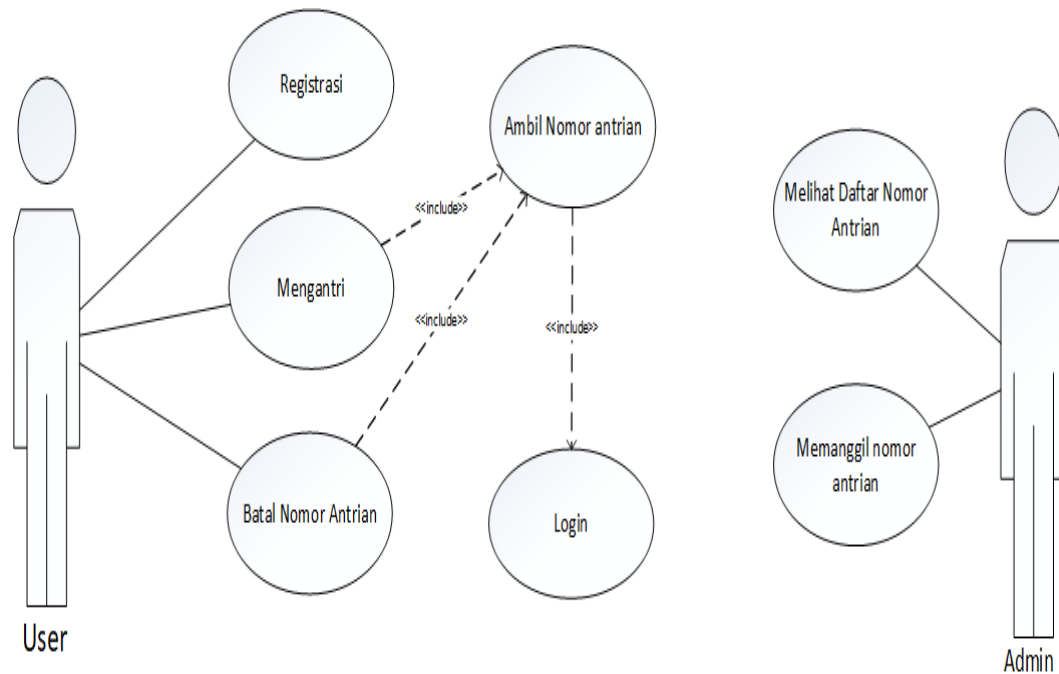
3.7 Rancangan Proses

Pada tahap ini diperlukan untuk membangun desain dari sistem yang akan dikembangkan. Setelah dilakukan analisa, terkumpulah kebutuhan informasi yang dibutuhkan untuk membangun sistem informasi. Untuk pemodelan sistem pada sebuah *database* diperlukan sebuah model analisis terstruktur, yaitu dengan menggunakan *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*. Pada sistem informasi ini, model yang dikembangkan adalah:

3.7.1 Use Case Diagram

Use case diagram merupakan salah satu bagian dari *Unified Markup language* (UML), *use case* diagram adalah suatu diagram yang menggambarkan suatu sistem dan bagaimana sistem tersebut bekerja. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Use case* diagram menggambarkan orang yang akan mengoperasikan atau berinteraksi dengan sistem.

Berikut adalah gambaran *use case diagram* Sistem Aplikasi Antrian Boediman.Jr Barbershop dapat dilihat pada Gambar 3.2



Gambar 3.2 *Use Case Diagram*

Berikut adalah keterangan tentang Gambar 3.2:

1. Terdapat 2 aktor dalam *use case diagram* aplikasi antrian Boediman.Jr Barbershop yaitu *user* dan *admin*.
2. *User* dapat melakukan registrasi.
3. *User* dapat melakukan *login*, melihat info nomor antrian, mengambil nomor antrian, dan membatalkan nomor antrian.
4. *Admin* dapat melakukan *login*, melihat daftar nomor antrian dan *admin* dapat memanggil nomor antrian.

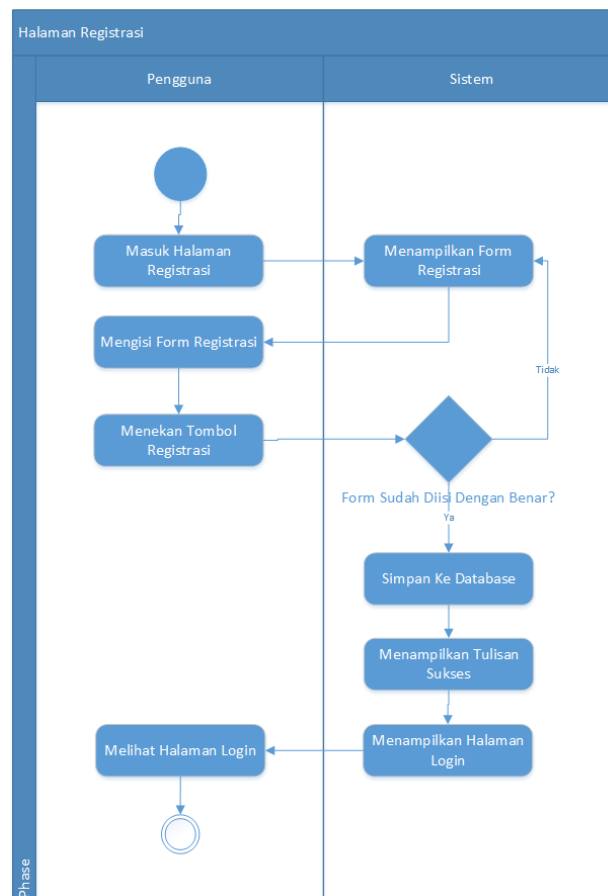
3.7.2 Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas yang ada dalam perangkat lunak yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana alur tersebut berakhir.

Berikut merupakan *activity diagram* yang terdapat pada Aplikasi Antrian Boediman.Jr Barbershop:

1. Activity Diagram Menu Registrasi

Activity Diagram Menu Registrasi pada Aplikasi Antrian Boediman.Jr Barbershop dapat dilihat pada Gambar 3.3.



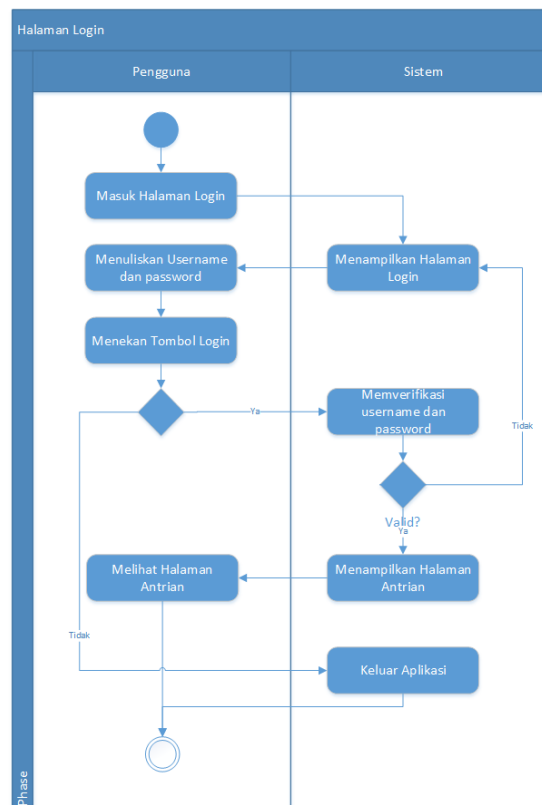
Gambar 3.3 Activity Diagram Menu Registrasi

Berikut adalah keterangan tentang Gambar 3.3:

- a. Pengguna masuk ke menu registrasi.
- b. Sistem akan menampilkan *form* registrasi.
- c. Pengguna mengisi *form* registrasi.
- d. Pengguna dapat memilih menekan button daftar atau Batal, jika pengguna menekan *button* daftar maka sistem akan melakukan *verifikasi*. jika pengguna menekan *button* batal maka pengguna dapat keluar dari aplikasi.
- e. Sistem melakukan *verifikasi* apakah *form* sudah diisi dengan benar, jika benar maka sistem akan menyimpan data ke *database*, jika salah sistem akan menampilkan *form* pendaftaran ulang.
- f. Penyimpanan ke *database* berhasil sistem akan menampilkan pesan sukses.

2. *Activity Diagram* Menu Login

Activity Diagram Menu login pada Aplikasi Antrian Boediman.Jr Barbershop dapat dilihat pada Gambar 3.4.



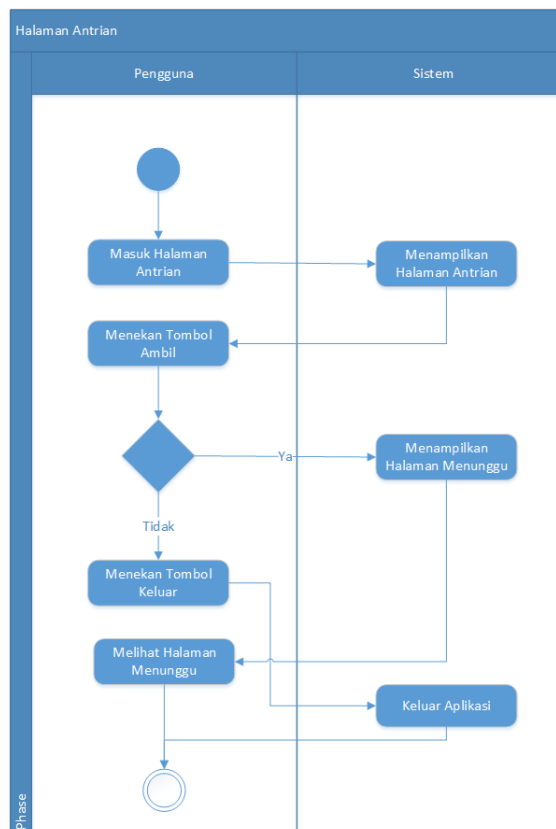
Gambar 3.4 Activity Diagram Menu Login

Berikut adalah keterangan tentang Gambar 3.4:

- Sistem menampilkan halaman *login*.
- Pengguna memasukkan *username* dan *password*.
- Pengguna dapat memilih menekan *button* masuk atau Batal, jika pengguna menekan *button* masuk maka sistem akan melakukan *verifikasi username* dan *password*. jika pengguna menekan *button* batal maka pengguna dapat keluar dari aplikasi.
- Sistem akan melakukan *verifikasi*, jika salah sistem akan meminta pengguna untuk memasukkan kembali *username* dan *password*. Jika benar *user* akan menampilkan halaman antrian.

3. *Activity Diagram* Halaman Antrian

Activity Diagram Halaman antrian pada Aplikasi Antrian Boediman.Jr Barbershop dapat dilihat pada Gambar 3.5



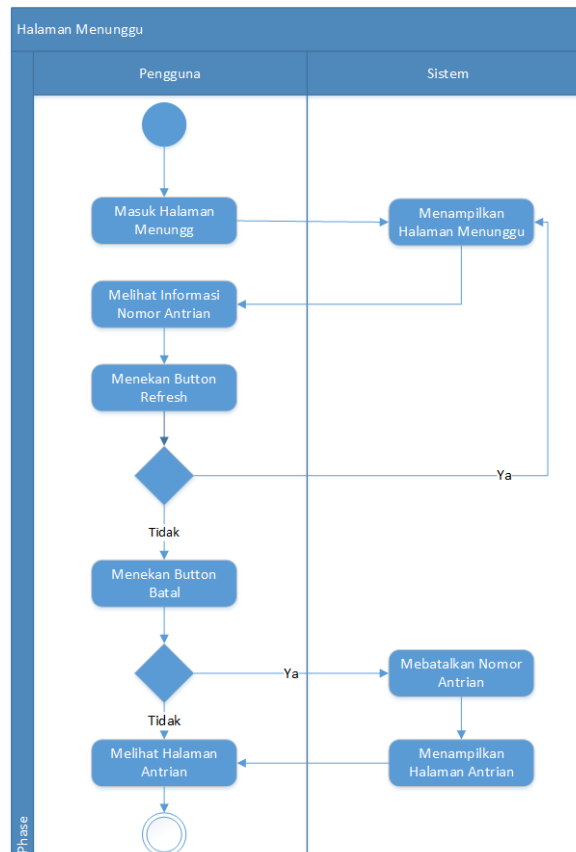
Gambar 3 5 *Activity Diagram* Halaman Antrian

Berikut adalah keterangan tentang Gambar 3.5:

- a. Pengguna masuk kehalaman antrian.
- b. Sistem menampilkan halaman antrian.
- c. Pengguna dapat memilih menekan *button* ambil atau Batal, jika pengguna menekan *button* ambil maka sistem akan menampilkan halaman menunggu, jika pengguna menekan tombol batal makan pengguna dapat keluar dari aplikasi.

4. Activity Diagram Halaman Menunggu

Activity Diagram Halaman Menunggu pada Aplikasi Antrian Boediman.Jr Barbershop dapat dilihat pada Gambar 3.6.



Gambar 3.6 Activity Diagram Halaman Menunggu

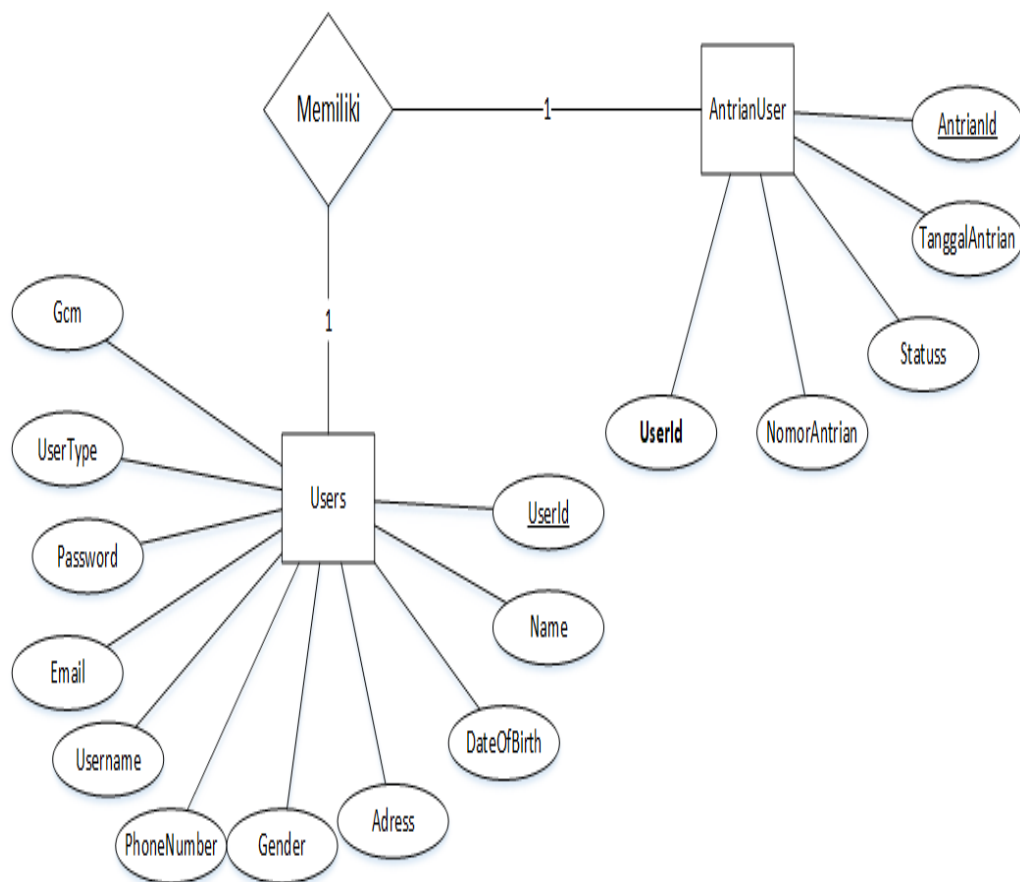
Berikut adalah keterangan tentang Gambar 3.6:

- a. Pengguna masuk ke halaman menunggu.
- b. Sistem akan menampilkan halaman menunggu.
- c. Penggun dapat melihat informasi nomor antrian.
- d. Penguana dapat menekan *button refresh* untuk melakukan *refresh* halaman menunggu.

- e. Pengguna dapat menekan *button* batal untuk membatalkan nomor antrian yang sudah diambil, sistem akan menampilkan halaman antrian.

3.7.3 ER Diagram

Gambaran *ER Diagram* yang digunakan pada *database* dapat dilihat pada Gambar 3.7



Gambar 3.7 Rancangan *ER Diagram*

3.7.4 Kamus Basis Data

Tabel 3.1 Struktur Tabel *Users*

No	Field Name	Key Type	Data Type	Keterangan
1	UserId	PK	<i>Int</i>	Berfungsi untuk id dari tabel <i>users</i> , contoh data : 1, 2, 3
2	Name		<i>Varchar(20)</i>	Berfungsi untuk nama <i>user</i> , contoh data : faisal azhari, faisal faras, admin
3	DateOfBirth		<i>Datetime</i>	Berfungsi untuk tanggal lahir <i>user</i> , contoh data : 1995-01-01, 1992-02-10
4	<i>Adress</i>		<i>Varchar(20)</i>	Berfungsi untuk alamat <i>user</i> , contoh data : gamping
5	<i>Gender</i>		<i>Varchar(9)</i>	Berfungsi untuk jenis kelamin <i>user</i> , contoh data : laki-laki, perempuan
6	PhoneNumber		<i>Varchar(12)</i>	Berfungsi untuk nomor telpon <i>user</i> , contoh data : 081321488933

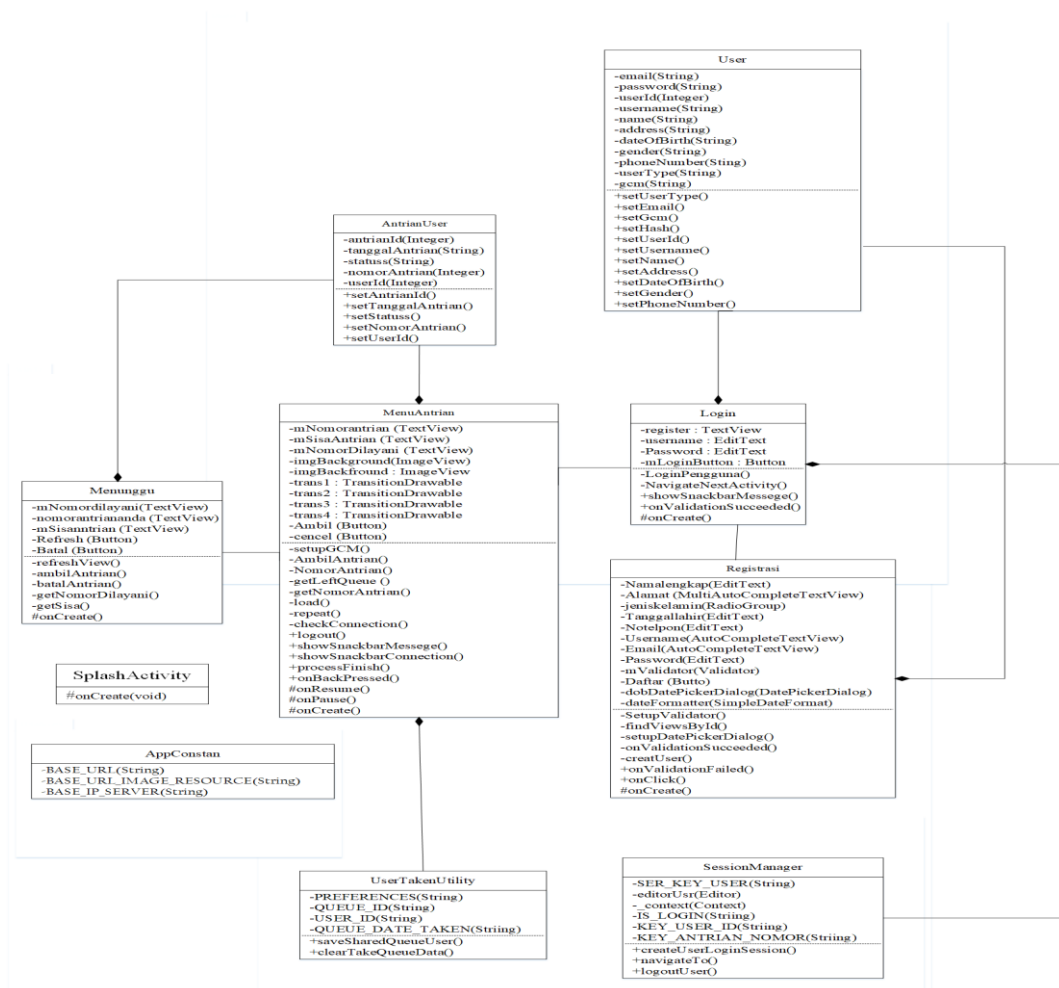
No	Field Name	Key Type	Data Type	Keterangan
7	<i>Username</i>		<i>Varchar(15)</i>	berfungsi untuk <i>username user</i> , contoh data : faisal, aldi, ridwan
8	<i>Passwords</i>		<i>Varchar(15)</i>	Berfungsi untuk <i>password user</i> , contoh data : akuaku12345, faisal123, aldo12345
9	<i>UserType</i>		<i>Varchar(5)</i>	Berfungsi untuk tipe <i>user</i> , contoh data : <i>User, Admin</i>
10	<i>Gcm</i>		<i>Varchar(160)</i>	<i>Google Cloud Messaging</i> berfungsi untuk notifikasi contoh : foQTKKDx_SM:APA91bH2 endZmrmbMtNm0mPvoG1D BuvhWZxWCnFl667jDp970 ogMxL82sTPBS0jQKc9b077 GQkycof6gsKuixRACHs1hC uLefij1HsQdx592Oq0Byctbz iMXrUkCbREYrbITRcRpDn YC

Tabel 3.2 Stuktur Tabel AntrianUser

No	Field Name	Key Type	Data Type	Keterangan
1	<i>AntrianId</i>	<i>PK</i>	<i>Int</i>	Id dari tabel antrianuser, contoh data : 1, 2, 3
2	TanggalAntrian		<i>Datetime</i>	Sebagai tanggal antrian , contoh data : 2017-10-24 00:00:00
3	Statuss		<i>Varchar(10)</i>	Sebagai status antrian <i>user</i> , contoh data : Menunggu, Dilayani, Selesai, Cencel
4	NomorAntrian		<i>Int</i>	Sebagai Nomor antrian <i>user</i> , contoh data : 1, 2, 3, 4, 5
5	<i>UserId</i>		<i>Int</i>	<i>Foreign keys</i> tabel <i>users</i>

3.7.5 Class Diagram

Rancangan *Class* diagram yang akan digunakan dalam proses pembuatan aplikasi dapat dilihat pada gambar 3.8.



Gambar 3.8 Rancangan *Class* Diagram

Berikut penjelasan fungsi dari *class* diagram yang di gambarkan pada gambar 3.8 adalah sebagai berikut:

1. *Class AppConstan* berfungsi sebagai jembatan antara aplikasi dengan *server*.
2. *Class SplashActivity* berfungsi menampilkan halaman *splashscreen* pada saat aplikasi dibuka.

3. *Class* Menunggu berfungsi untuk menampilkan informasi nomor antrian *user*, nomor yang sedang dilayani, dan nomor sisa antrian.
4. *Class* MenuAntrian berfungsi untuk menampilkan informasi nomor antrian sekarang, nomor antrian dilayani, nomor sisa antrian dan mengambil nomor antrian.
5. *Class Login* berfungsi untuk mengakses fungsi yang lain, dengan cara memasukkan *username* dan *password*.
6. *Class* registrasi berfungsi untuk membuat akun sebagai inisialisasi.
7. *Class* SeesionManager berfungsi untuk menyimpan akun *user* ketika *login* pada *class login*, dan ketika *user* melakukan *logout* data akun *user* tersimpan akan di *clear* oleh *method* `logoutUser()`
8. *Class* `UserTakenUtility` berfungsi untuk menyimpan nomor antrian ketika *user* mengambil nomor antrian.
9. *Class* `AntrianUser` adalah data *object* yang merupakan inisialisasi data yang diambil dari *server*.
10. *Class* `User` adalah data *object* yang merupakan inisialisasi data yang diambil dari *server*.

Berikut adalah penjelasan relasi antar kelas pada *class diagram* yang digambarkan pada gambar 3.6:

1. *Class* `login` memiliki relasi *association* dengan *class* `MenuAntrian` dan *class* `Registrasi`, karena *class* `Login` dapat mengakses *class* `MenuAntrian` dan *class* `Registrasi`, melalui *Button* `mLoginButton` dan *TextView* `register`.

2. *Class MenuAntrian* memiliki relasi *association* dengan *class Menunggu*, karena *class MenuAntrian* dapat mengakses *class Menunggu* melalui *Button Ambil*.
3. *Class User* memiliki relasi *composition* dengan *class Login*, artinya *class user* merupakan bagian dari *class Login*, *class User* tidak bisa berdiri sendiri apabila *class Login* tidak ada.
4. *Class AntrianUser* memiliki relasi *composition* dengan *class MenuAntrian*, artinya *class AntrianUser* merupakan bagian dari *class MenuAntrian*, *class AntrianUser* tidak bisa berdiri sendiri apabila *class MenuAntrian* tidak ada.
5. *Class SessionManager* memiliki relasi *composition* dengan *class Login*, artinya *class SessionManager* merupakan bagian dari *class Login*, *class SessionManager* tidak bisa berdiri sendiri apabila *class Login* tidak ada.
6. *Class UserTakenUtility* memiliki relasi *composition* dengan *class MenuAntrian*, , artinya *class UserTakenUtility* merupakan bagian dari *class MenuAntrian*, *class UserTakenUtility* tidak bisa berdiri sendiri apabila *class MenuAntrian* tidak ada.

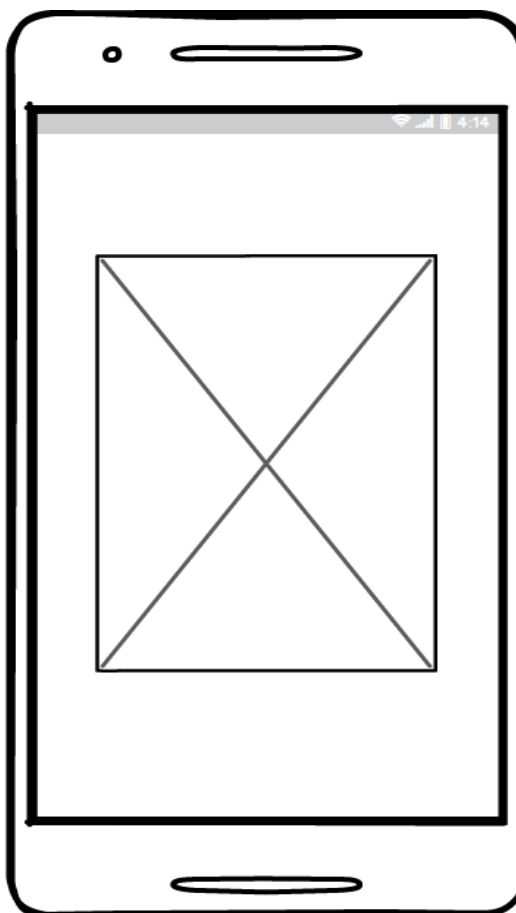
3.8 Rancangan Antar Muka

Perancangan antarmuka merupakan tampilan dari suatu perangkat lunak yang berperan sebagai media komunikasi antara perangkat lunak dan pengguna. perancangan ini merupakan sebuah penggambaran, perencanaan dan pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh. Perancangan antarmuka diharapkan memudahkan pengguna dalam melakukan proses interaksi terhadap sistem.

Berikut adalah perancangan antar muka yang ada pada Aplikasi Antrian Boediman.Jr Barbershop:

3.8.1 Antarmuka *Splash Screen*

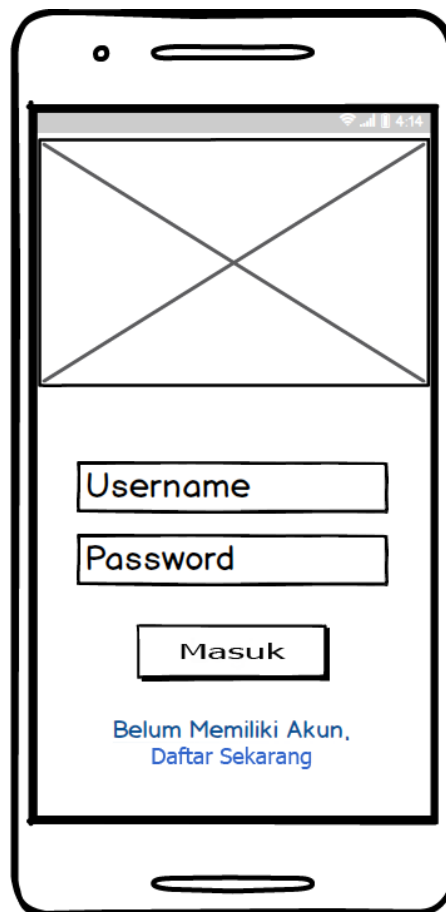
Halaman *splash screen* merupakan halaman pertama saat pertama kali membuka aplikasi antrian boediman.jr barbershop. Berikut ini merupakan perancangan antarmuka splash screen pada aplikasi antrian Boediman.jr barbershop seperti terlihat pada Gambar 3.9.



Gambar 3. 9 Rancangan antarmuka *Splash Screen*

3.8.2 Antarmuka Halaman *Login*

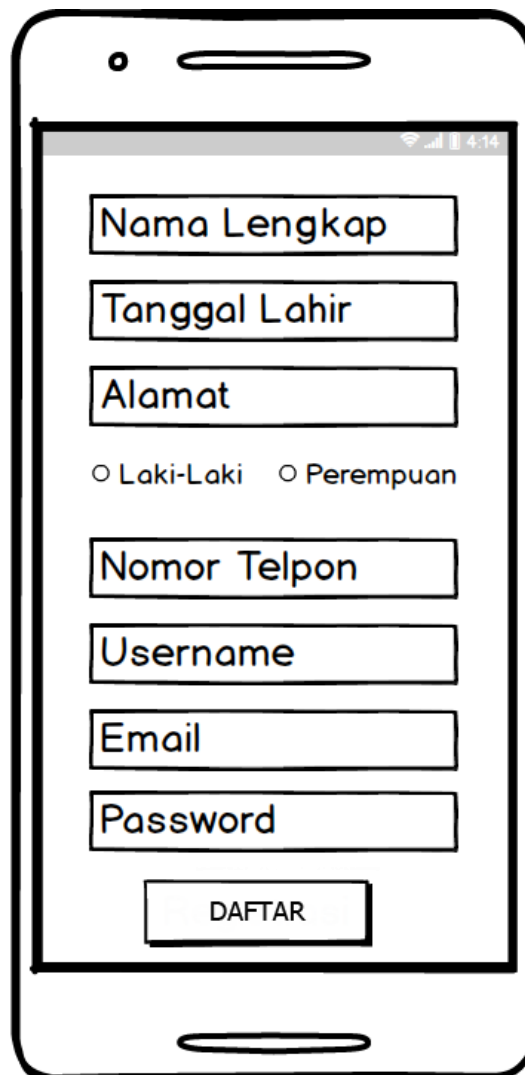
Untuk masuk ke halaman menu utama, pengguna harus melakukan *login* terlebih dahulu. Untuk melakukan *login* pengguna harus memasukkan *username* dan *password*. Gambaran rancangan antarmuka *login* dapat dilihat pada Gambar 3.9



Gambar 3.10 Rancangan antarmuka Login

3.8.3 Antarmuka Halaman Registrasi

Antarmuka Registrasi merupakan tampilan apabila pengguna akan melakukan pendaftaran, pada tampilan ini terdapat *form* yang harus di isi pengguna yaitu nama, tempat, tanggal lahir, jenis kelamin, nomor telepon, *email*, *username*, katasandi. rancangan antarmuka *registrasi* dapat dilihat pada Gambar 3.11.

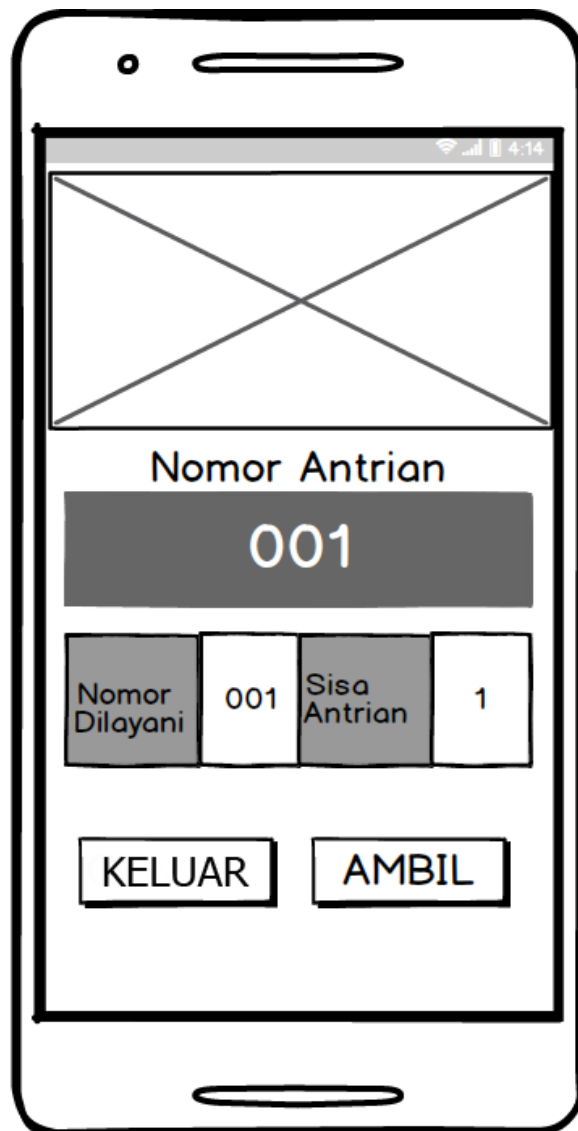


The image shows a mobile registration form layout. At the top, there are status icons for signal strength, Wi-Fi, and battery, along with the time 4:14. The form consists of the following elements from top to bottom: a text input field labeled 'Nama Lengkap', a date input field labeled 'Tanggal Lahir', a text input field labeled 'Alamat', two radio buttons labeled 'Laki-Laki' and 'Perempuan', a text input field labeled 'Nomor Telpon', a text input field labeled 'Username', a text input field labeled 'Email', a text input field labeled 'Password', and a button labeled 'DAFTAR'.

Gambar 3.11 Rancangan antarmuka *Registrasi*

3.8.4 Antarmuka Halaman Antrian

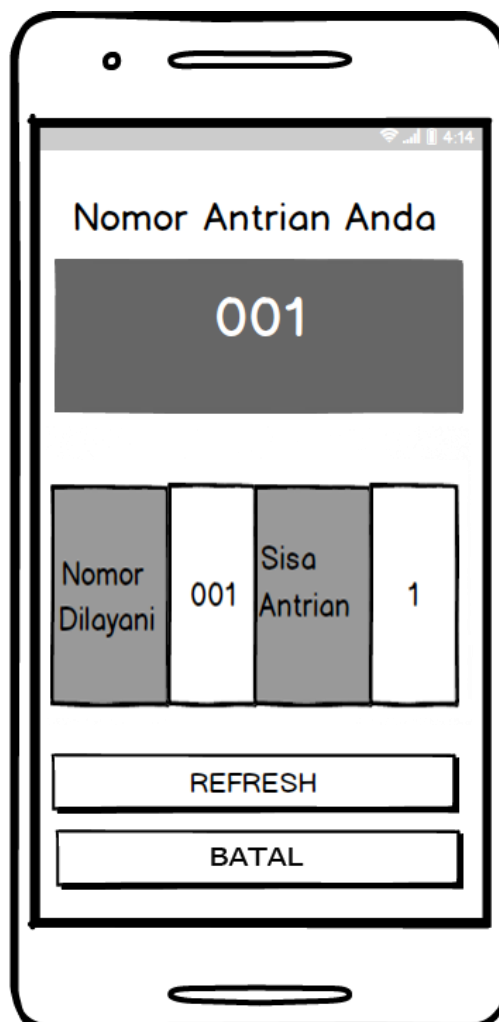
Antarmuka halaman antrian menampilkan informasi nomor antrian sekarang, nomor antrian dilayani, nomor sisa antrian, tombol ambil untuk mengambil nomor antrian, dan tombol *cancel* untuk keluar dari aplikasi. rancangan antarmuka halaman antrian dapat dilihat pada Gambar 3.12.



Gambar 3.12 Rancangan Antarmuka Antrian

3.8.5 Antarmuka Halaman Menunggu

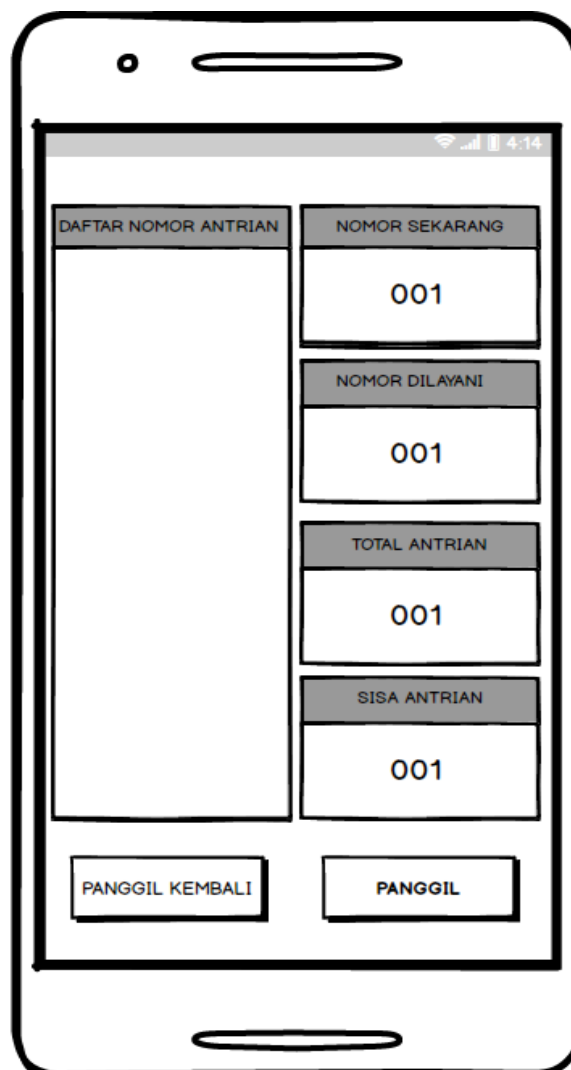
Antarmuka halaman menunggu merupakan halaman disaat *user* mengambil nomor antrian. pada halaman menunggu menampilkan informasi nomor antrian *user*, nomor yang sedang dilayani, dan nomor sisa antrian. *button refresh* untuk melakukan *refresh* ulang halaman menunggu, *button* batal untuk membatalkan nomor antrian yang sudah diambil. rancangan antarmuka halaman menunggu dapat dilihat pada Gambar 3.13.



Gambar 3.13 Rancangan Antarmuka Menunggu

3.8.6 Antarmuka Halaman Admin

Rancangan halaman antarmuka admin merupakan tampilan yang hanya bias diakses oleh admin. Pada halaman admin menampilkan informasi nomor sekarang, nomor dilayani, total antrian, sisa antrian dan daftar nomor antrian. *Button* panggil berfungsi memanggil nomor antrian pengguna.



Gambar 3.14 Rancangan Antarmuka *Admin*