

BAB III
METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu dan tempat penelitian merupakan serangkaian gambaran umum yang menjelaskan lokasi serta waktu dalam mengumpulkan data dalam sebuah penelitian atau riset.

3.1.1 Waktu Penelitian

Penelitian dilaksanakan pada semester ganjil dan genap tahun akademik 2021/202, tabel waktu penelitian sebagai berikut :

Tabel 3.1 Jadwal Penelitian

Kegiatan	Tahun 2021-2022					
	Septemb er	Oktober	Novem ber	Desem ber	Januar i	Februa ri
Pengumpulan Data						
Perancangan Alat						
Pengumpulan Alat dan Bahan						
Perancangan Perangkat Lunak						

Uji Coba Alat					
---------------	--	--	--	--	--

3.1.2 Tempat Penelitian

Dalam penelitian ini, pengumpulan data hingga uji coba alat dilakukan di perusahaan SHATOMEDIA Yogyakarta.

3.2 Alat dan Bahan

Pada perancangan dan pembuatan sistem ini dibutuhkan beberapa alat dan bahan untuk membuat sistem antrian online. Daftar alat dan bahan yang digunakan dalam melakukan perancangan dan pembuatan sistem ini sebagai berikut :

a) Kebutuhan Perangkat Keras

Tabel 3.2 Perangkat Keras

N o.	Kebutuhan Perangkat Keras	Spesifikasi
1.	Laptop	RAM 8GB, <i>processor</i> intel [®] core [™] i5-9400F CPU @ 2.90 GHz
2.	Modul P10	Resolusi 16 LED (T) x 32 LED (L), tegangan <i>input</i> DC 5V, jarak antar LED 10mm, jumlah LED 512 per modul
3.	Modul <i>Speaker</i>	<i>Speaker</i> ACR 4 inch full range KSV 20WATT, diameter 4 inch, <i>impedence</i> 8 ohm, tinggi 4.2 cm
4.	DFPlayer Mini	Output 24-bit DAC, dukungan rentang dinamis 90dB, SNR mendukung 85dB, mendukung sistem file FAT16 dan FAT32
5.	NodeMCU ESP8266	Tegangan <i>input</i> 3.3 – 5V, GPIO 13 PIN, frekuensi 2.4 GHz – 22.5 Ghz, USB <i>port</i> micro USB

6.	<i>Smartphone</i>	RAM 2GB, memori <i>internal</i> 16GB
----	-------------------	--------------------------------------

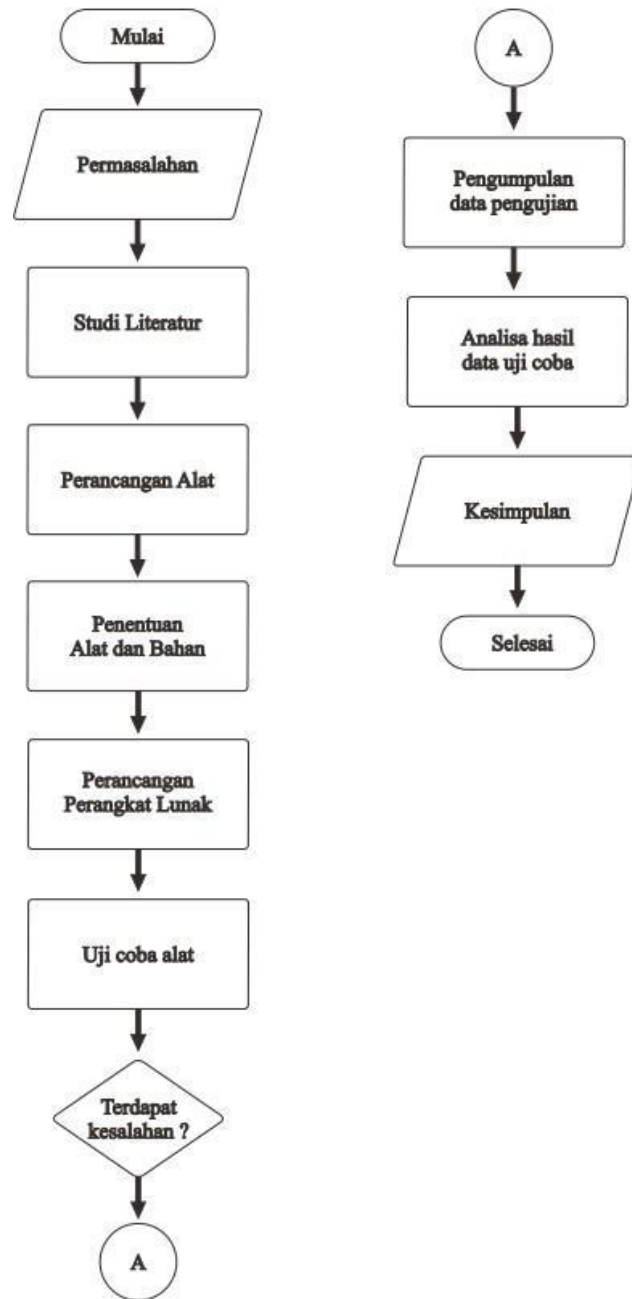
b) Kebutuhan Perangkat Lunak

Tabel 3.3 Perangkat Lunak

N o.	Kebutuhan Perangkat Lunak	Spesifikasi
1.	<i>Windows</i>	Versi <i>windows</i> 10 pro
2.	Arduino IDE	Versi arduino 1.8.15
3.	Android Studio	Android studio chipmunk 2021.2.1 patch 1
4.	Firefox	Versi 103.0.1 (64-bit)
5.	Android OS	Versi android 5.0 <i>lollipop</i>

3.3 Alur Penelitian

Dalam melakukan pembuatan sistem ini dilakukan perancangan setelah mengetahui latar belakang dari sistem yang dibuat. Setelah itu menentukan alat dan bahan apa saja yang dibutuhkan untuk membuat sistem, mulai dari cara pengambilan nomor antrian, melihat nomor antrian, memanggil dan mereset nomor antrian oleh admin. Dalam menyusun tugas akhir ini, terdapat beberapa tahapan yang dapat dilihat pada diagram alir dibawah ini:



Gambar 3.1 Diagram Alur Penelitian

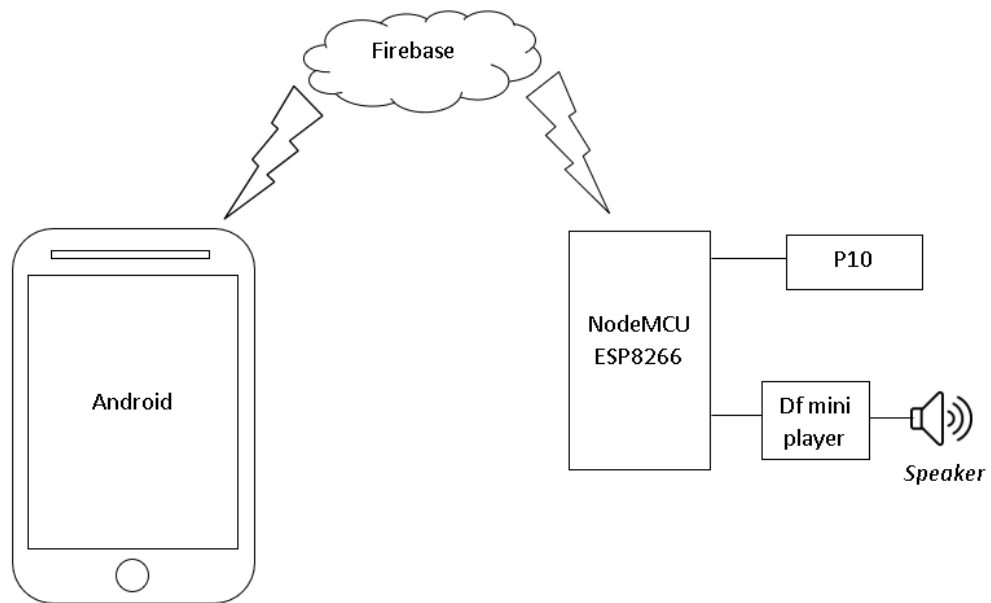
3.4 Perancangan Sistem

Setelah menentukan alat dan bahan yang dibutuhkan pada penelitian ini, langkah selanjutnya adalah merancang sistem. Adapun tahap perancangan sistem yang akan dilakukan meliputi perancangan *hardware*, perancangan *software* alat, dan perancangan *software* aplikasi.

3.4.1 Perancangan *Hardware*

Pada perancangan ini digunakan NodeMCU ESP8266 sebagai pengendali dari *hardware*. NodeMCU ESP8266 dipilih karena sudah terdapat modul wifi didalamnya, sehingga bisa bertugas sebagai *client* maupun *access point*. NodeMCU ESP8266 ini akan membaca apakah data nomor antrian telah berganti atau masih dengan data yang sama.

Ketika data yang di ambil oleh NodeMCU ESP8266 terdapat perubahan, maka data tersebut akan dikirim ke Modul P10 dan DFPlayer Mini untuk mengeksekusi program yang telah diatur. Setelah mendapatkan data dari NodeMCU ESP8266, *Speaker* akan memutar file audio yang diambil dari DFPlayer Mini yang berupa suara pemanggil nomor antri. Pada waktu yang sama, Modul P10 juga akan menampilkan nomor antrian yang sedang berjalan.

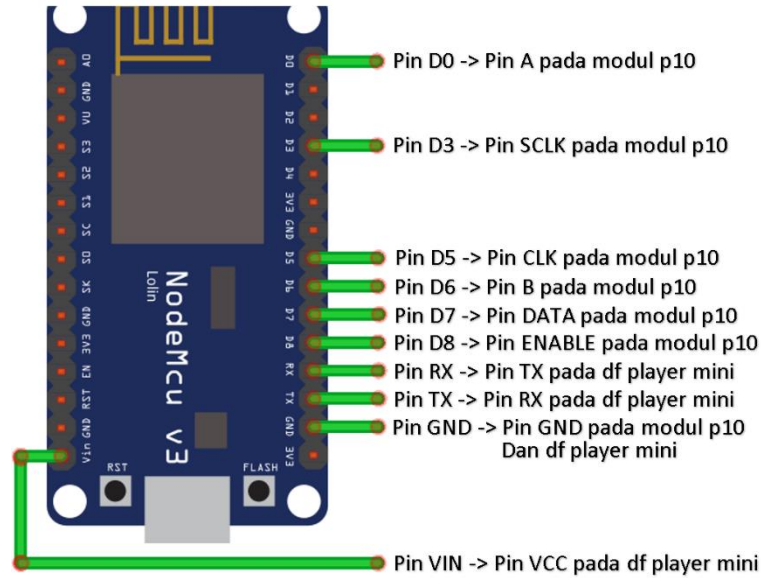


Gambar 3.2 Diagram Blok Sistem

Gambar 3.2 merupakan blok diagram alat yang akan dibuat. NodeMCU ESP8266 akan selalu membaca data dari *database* Firebase secara *Realtime*. Ketika terdapat perubahan pada Firebase yang didapat dari Aplikasi Antrian, NodeMCU akan langsung membaca dan memberikan perintah ke DFPlayer Mini dan Modul P10. Setelah DFPlayer Mini menerima perintah, DFPlayer Mini akan mengeluarkan file *audio* yang kemudian akan diputarkan oleh *Speaker*. Fungsi dari alat dan bahan yang digunakan yaitu:

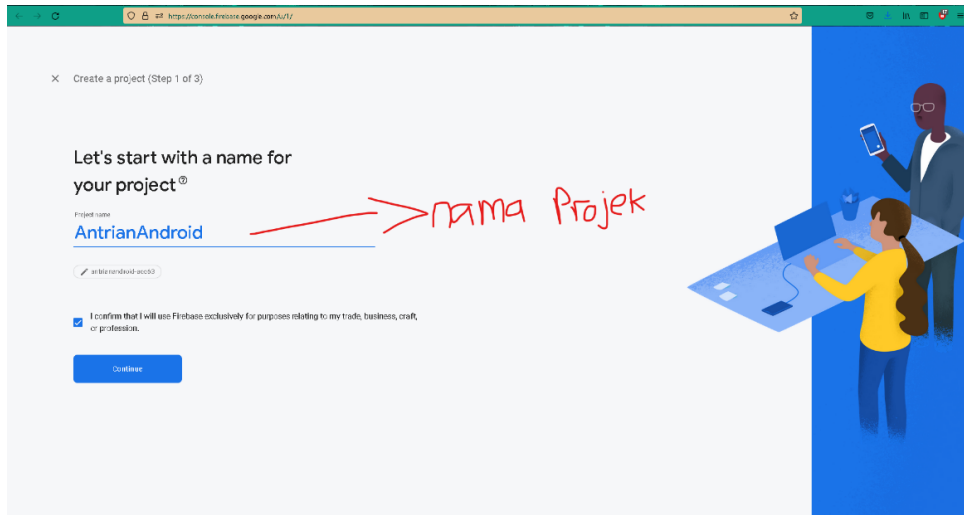
a. NodeMCU ESP8266

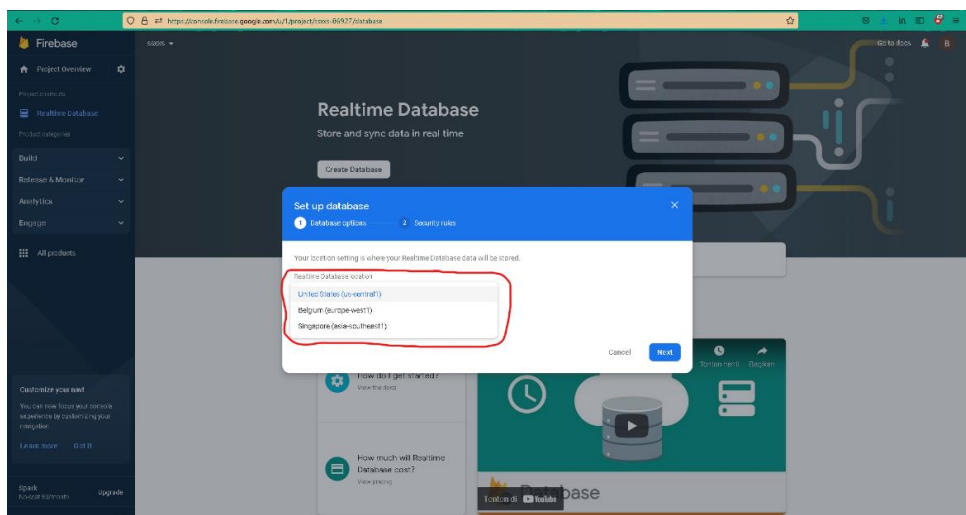
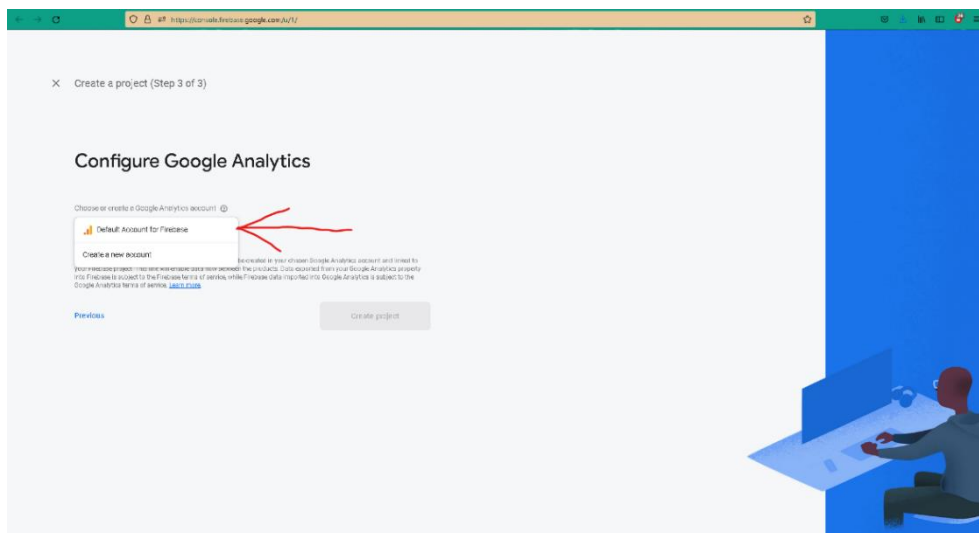
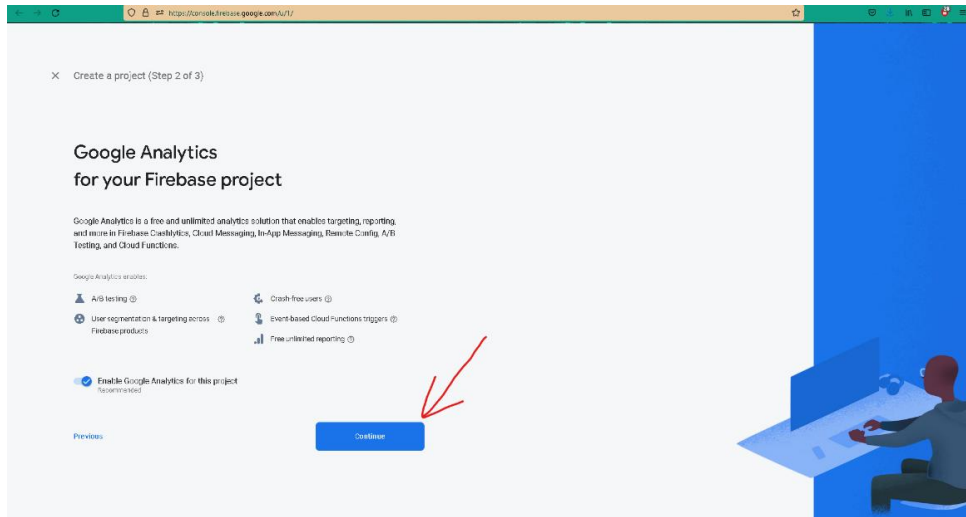
NodeMCU ES8266 pada konsep ini digunakan sebagai *client* dari *database* firebase. NodeMCU akan selalu membaca data dari firebase secara terus menerus sehingga ketika terdapat suatu perubahan data, NodeMCU akan membaca perubahan data tersebut dan menjadikan perintah untuk DFPlayer Mini dan Modul P10. Alasan menggunakan mikrokontroler jenis NodeMCU ESP8266 karena sudah terdapat modul wifi didalamnya.

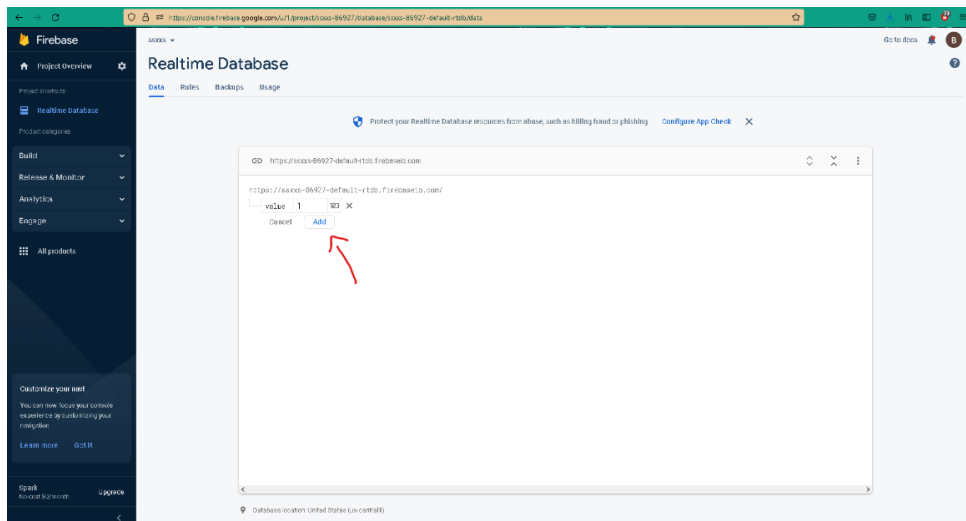
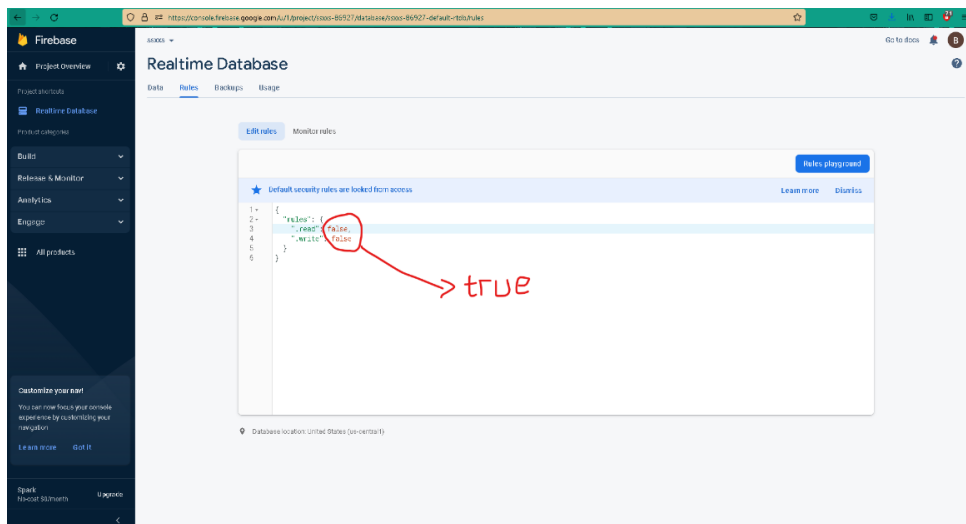
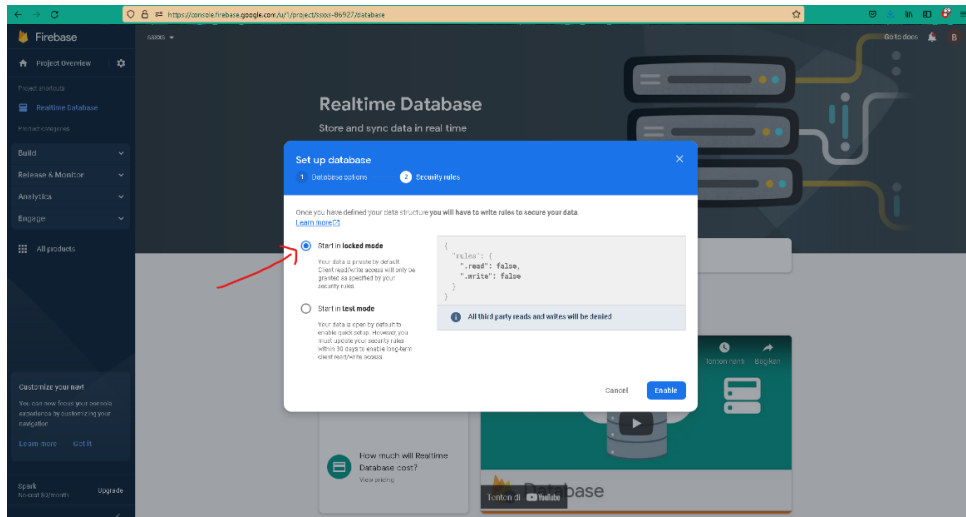


b. Firebase

Firebase digunakan sebagai tempat penyimpanan data pengguna, admin, maupun data antrian. Firebase akan menerima data dari Aplikasi secara *realtime*, dan data dari firebase akan dibaca oleh NodeMCU ESP8266 yang kemudian data dalam bentuk antrian tersebut ditampilkan dalam bentuk nomor antrian pada Modul P10, dan data antrian ini akan dikonversikan kedalam bentuk suara yang diperintahkan ke DFPlayer Mini untuk diteruskan ke *Speaker*.

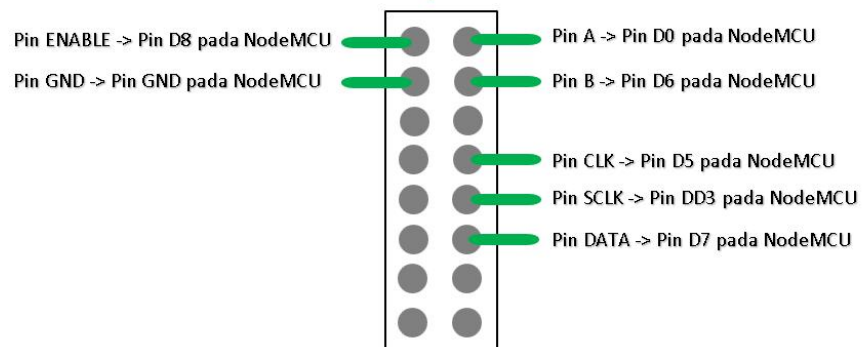
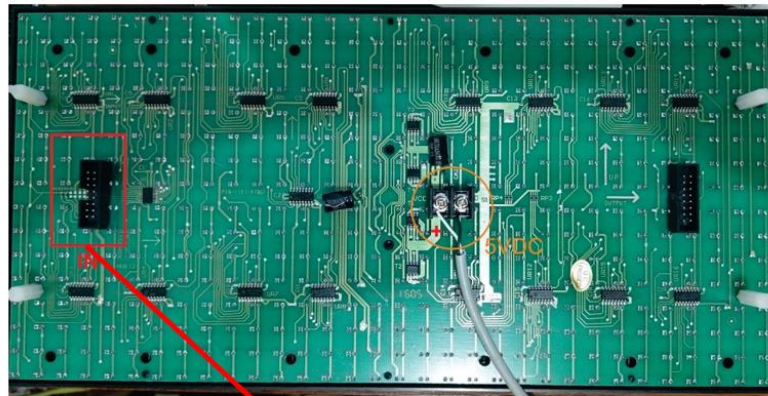






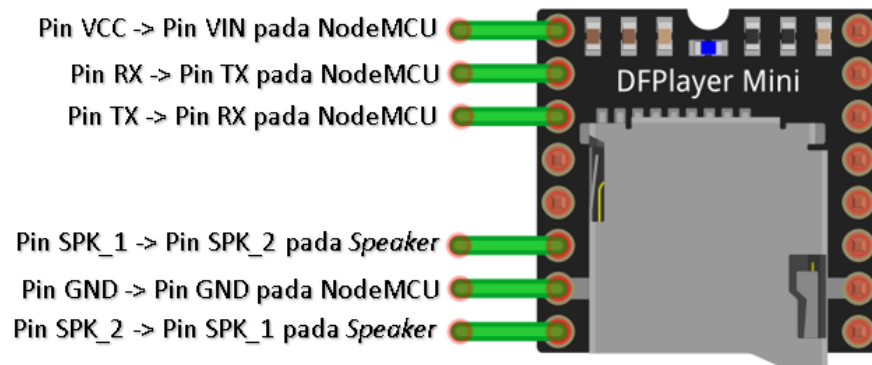
c. Modul P10

Modul P10 merupakan komponen yang dapat menampilkan LED dalam jumlah banyak sehingga dapat diatur sesuai keinginan pengguna. Modul P10 pada sistem antrian ini berfungsi sebagai papan informasi yang menampilkan nomor antrian yang sedang berlangsung secara *realtime*. Modul P10 akan menampilkan berupa angka antrian dan berkedip setiap 5 detik sekali.



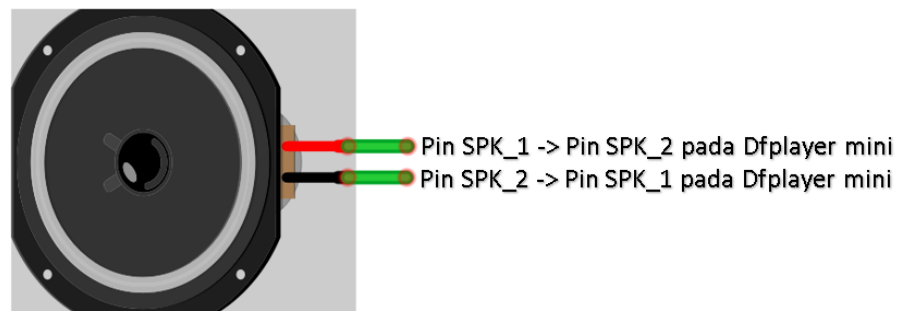
d. DFPlayer Mini

DFPlayer Mini berfungsi sebagai komponen yang akan memutar file audio sesuai perintah dari NodeMCU ESP8266. DFPlayer Mini akan mengirim file audio ketika nomor antrian berubah, dan file audio tersebut akan di eksekusi oleh *Speaker* untuk mengeluarkan suara.



e. *Speaker*

Speaker merupakan komponen yang dapat merubah sinyal listrik menjadi sinyal suara. Pada sistem antrian ini *Speaker* akan berperan sebagai komponen yang mengeluarkan audio sesuai dengan file audio pada DFPlayer Mini.



f. *Smartphone Android*

Smartphone Android digunakan sebagai pengirim dan penerima data dari Aplikasi Antrian. Baik untuk *user* maupun admin, harus menggunakan *Smartphone* Android untuk mengakses. Pada *smartphone* yang telah *terinstall* aplikasi antrian, *user* bisa mulai menggunakan aplikasi tersebut. Langkah pertama *user* harus registrasi dahulu sebelum masuk ke aplikasi. Pada tahap ini aplikasi akan mengirim data *user* ke firebase untuk disimpan. Setelah registrasi *user* akan ditunjukan ke halaman *Login* untuk masuk ke aplikasi.

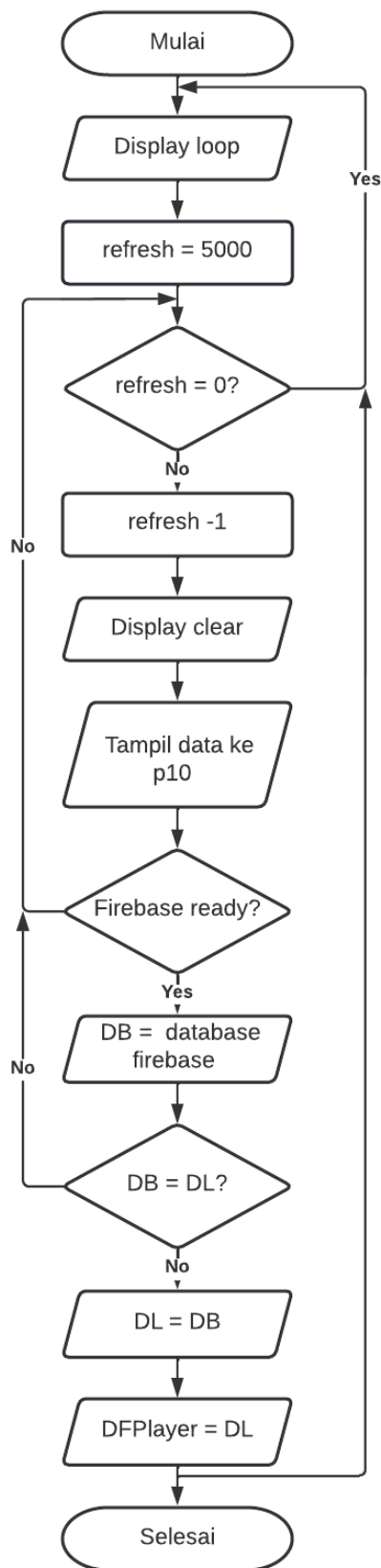
Setelah mengisi semua form yang ada pada halaman *Login*, maka aplikasi akan membaca data dari firebase apakah data tersebut ada dalam firebase dan apakah dia *user* atau admin. Jika tidak ada, maka *user* tidak bisa masuk. Jika ada, maka aplikasi akan meneruskan ke halaman *dashboard user* jika dia *user*, dan meneruskan ke *dashboard admin* jika admin. Pada *dashboard user* akan terdapat 3 menu yaitu ambil antrian, lihat antrian, dan *logout*. Ketika *user* masuk ke menu ambil antrian, maka akan mendapati 2 form untuk diisi yaitu nama dan nomor hp sebelum mengambil nomor antrian. Setelah form diisi dengan benar dan *user* menekan tombol ambil antrian, aplikasi akan mengirim data tersebut ke *database* firebase. Kemudian aplikasi akan meneruskan ke halaman lihat antrian. Pada halaman lihat antrian terdapat 2 nomor antrian yang ditampilkan, pertama nomor antrian yang sedang berjalan dan yang kedua nomor antrian si *user*. Dan ketika *user* menekan menu *logout*, maka aplikasi akan mengeluarkan akun *user* dari aplikasi kemudian akan diteruskan ke halaman *Login*.

3.4.2 Perancangan Software Alat

Setelah perancangan *Hardware* maka selanjutnya dilanjutkan dengan perancangan *Software* atau code agar sensor dapat bekerja. Kode ditulis dengan menggunakan Bahasa C, Penulisan kode dilakukan pada aplikasi Arduino IDE. Kode dibuat untuk dapat memfungsikan Modul P10, DFPlayer dan keseluruhan alat agar alat dapat selalu membaca data dari Firebase secara *realtime*.

NodeMCU ESP8266 membaca data dari Firebase melalui jaringan Wifi, kemudian hasil pembacaan data tersebut akan diproses. Jika nomor antrian berubah, maka Modul P10 akan menampilkan nomor antrian sesuai data Firebase, dan DFPlayer Mini mengeluarkan file suara yang diteruskan ke

Speaker sesuai data yang didapat. Berikut dalah gambar *flowchart* program sistem.



Keterangan:
 antrian = Data Baru = DB
 angka = Data Lama = DL
 Tampil data = data pembacaan
 dari firebase yang disimpan
 pada memori NodeMCU

Gambar 3.3 Flowchart Program

Gambar 3.3 merupakan *flowchart* program pada sistem yang akan dirancang. Tujuan dari *flowchart* ini untuk menjadikan tanda bahwa program tersebut dapat dioperasikan dan siap dijalankan. Pada proses ini terdapat dua keterangan yaitu antrian = Data Baru = DB dan angka = Data Lama = DL.

Program berjalan diawali dengan start yang kemudian menjalankan perintah *display loop*, yang berarti menampilkan data yang ada pada memori NodeMCU ESP8266 ke LED ke Modul P10. Setelah menampilkan data, selanjutnya membuat sebuah variabel dengan nama *refresh* dan memiliki nilai 5000. Variabel ini akan diperiksa, jika nilai variabel *refresh* 0 maka kembali ke proses *display loop* atau proses menampilkan data yang ada pada memori NodeMCU ESP8266 ke LED pada Modul P10, jika nilai variabel *refresh* tidak 0 maka nilai variabel tersebut akan dikurangi -1. Kemudian menjalankan *display clear*, yang berfungsi untuk menghapus data yang ada pada memori sekunder NodeMCU ESP8266, dan menampilkannya pada Modul P10. Selanjutnya proses pembacaan apakah Firebase siap? Jika iya, maka NodeMCU ESP8266 akan mengambil data dari firebase dan menyimpannya pada variabel DB yang disimpan ke memori NodeMCU ESP8266. Jika firebase tidak siap, maka proses akan kembali ke pembacaan apakah nilai variabel *refresh* bernilai 0.

Setelah NodeMCU ESP8266 mengambil data dari firebase dan menyimpannya dalam variabel DB, kemudian melakukan pemeriksaan apakah DB sama dengan DL? Jika sama, maka proses akan kembali ke pemeriksaan apakah variabel *refresh* bernilai 0? Dan mengulang proses yang sudah dilakukan sebelumnya sesuai prosedur. Jika DB tidak sama dengan DL, maka nilai variabel DB akan disimpan sebagai nilai variabel DL. Kemudian DFPlayer Mini sama dengan variabel DL, yang berarti DFPlayer Mini akan selalu mengikuti perubahan data yang terjadi pada DL.

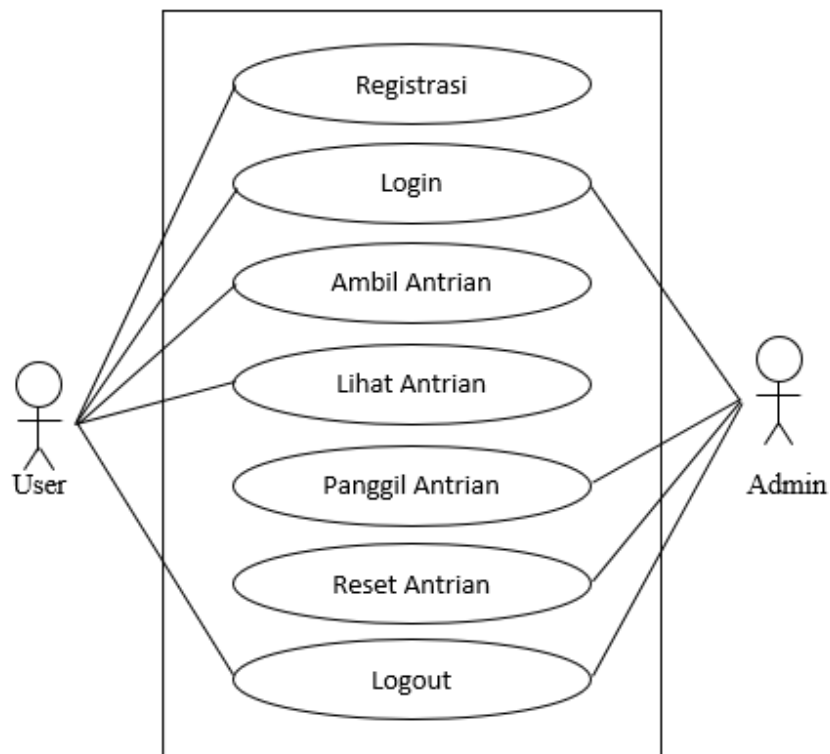
3.4.3 Perancangan Software Aplikasi

Setelah perancangan *software* alat, maka selanjutnya dengan perancangan *software* aplikasi. Tahapan perancangan sistem merupakan dianalisis ke bentuk yang mudah dimengerti oleh pemakai (*user*). Adapun perancangan pada tahap ini meliputi:

1. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case Diagram* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini 2016).

Dalam penelitian ini penulis membuat *use case diagram* pada aplikasi antrian pada gambar berikut:



Gambar 3.4 Use Case Diagram Aplikasi Antrian

a) Registrasi

Nama *Use Case* : Registrasi

Aktor : *User*

Deskripsi : Proses ini adalah tahap awal yang dilakukan oleh *user* untuk menggunakan aplikasi

Pre-Condition : *User* membuka aplikasi kemudian mengisi form registrasi dengan nama lengkap, *email*, dan *password* dengan benar

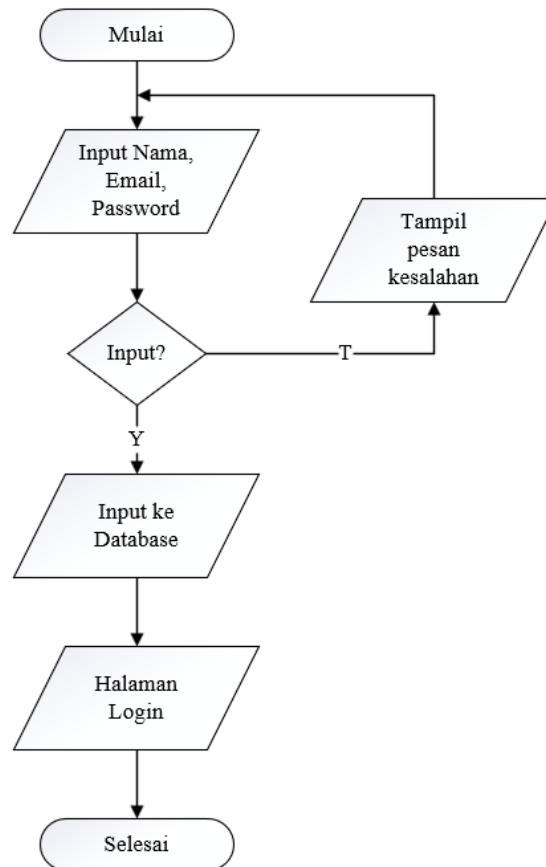
Post-Condition : Sistem akan memeriksa apakah data diisi semua dan

apakah data yang dimasukan valid. Jika valid maka akan terdaftar pada aplikasi

Actor	Reaksi Sistem
<i>User</i> menekan tombol "Register"	
	Menampilkan form register
Memasukan <i>fullname</i> , <i>email</i> dan <i>password</i>	
	Validasi data

Tabel 3.4 Skenario *Use Case Diagram* Registrasi

Berikut juga *gambar* diagram *flowchart* pada form registrasi :



Gambar 3.5 *Flowchart* Registrasi

b) *Login*

Nama *Use Case* : *Login*

Aktor : *User, Admin*

Deskripsi : Setelah terdaftar pada aplikasi, maka harus masuk atau *Login* terlebih dahulu untuk dapat mengakses

aplikasi

Pre-Condition : *User* atau *admin* membuka aplikasi kemudian mengisi *form Login* dengan *username* dan *password* dengan benar

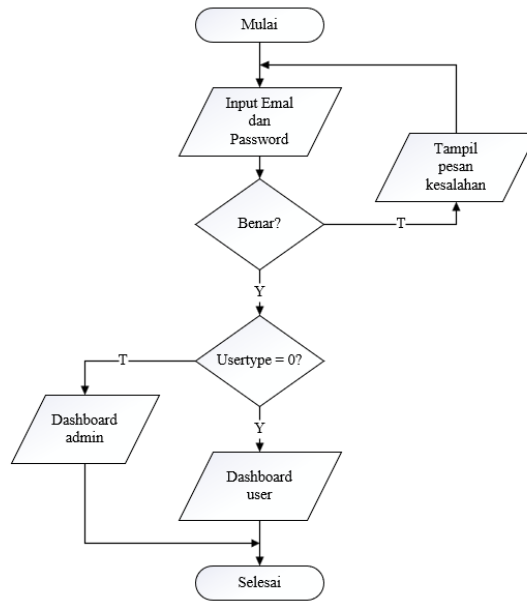
Post-Condition : Sistem akan memeriksa validasi data yang dimasukan. Apabila valid maka *User* akan masuk sebagai pengguna pada aplikasi tersebut dan Admin

sebagai admin pada aplikasi tersebut

Tabel 3.5 Skenario *Use Case Diagram Login*

Actor	Reaksi Sistem
<i>User</i> atau admin membuka aplikasi	
	Menampilkan form <i>Login</i>
Memasukan email dan <i>password</i>	
	Validasi <i>User</i> atau Admin

Berikut juga gambar diagram *flowchart* pada form *Login* :



Gambar 3.6 *Flowchart Login*

c) Ambil Antrian

Nama *Use Case* : Ambil Antrian

Aktor : User

Deskripsi : User mengisi form Ambil Antrian dengan nama dan

nomor *handphone* yang benar

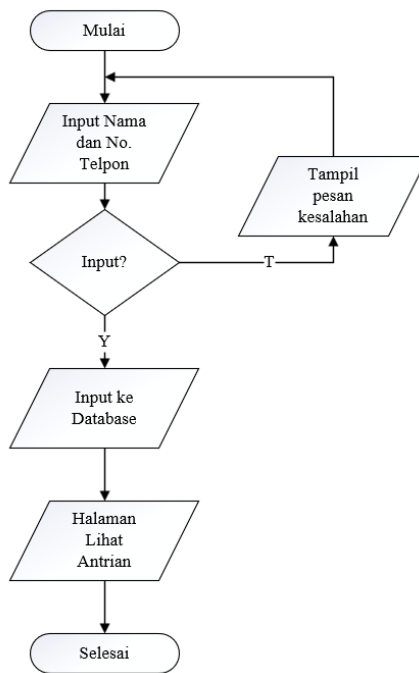
Pre-Condition : User membuka aplikasi kemudian mengisi form ambil antrian dengan nama dan nomor *handphone* yang benar

Post-Condition : Sistem akan memeriksa validasi data yang dimasukan. Apabila valid maka akan mendapatkan nomor antrian

Tabel 3.6 Skenario *Use Case Diagram* Ambil Antrian

Actor	Reaksi Sistem
<i>User</i> atau admin menekan tombol ambil antrian	
	Menampilkan form ambil antrian
Memasukan nama dan nomor <i>handphone</i>	
	Menampilkan nomor antrian

Berikut juga gambar diagram *flowchart* Ambil Antrian :



Gambar 3.7 Flowchart Ambil Antrian

d) Lihat Antrian

Nama Use Case : Lihat Antrian

Aktor : User

Deskripsi : User dapat melihat nomor antri yang telah di ambil

Pre-Condition : User melihat nomor antrian yang telah diambil pada

halaman ambil antrian

Post-Condition : User dapat mengecek nomor antriannya kapanpun

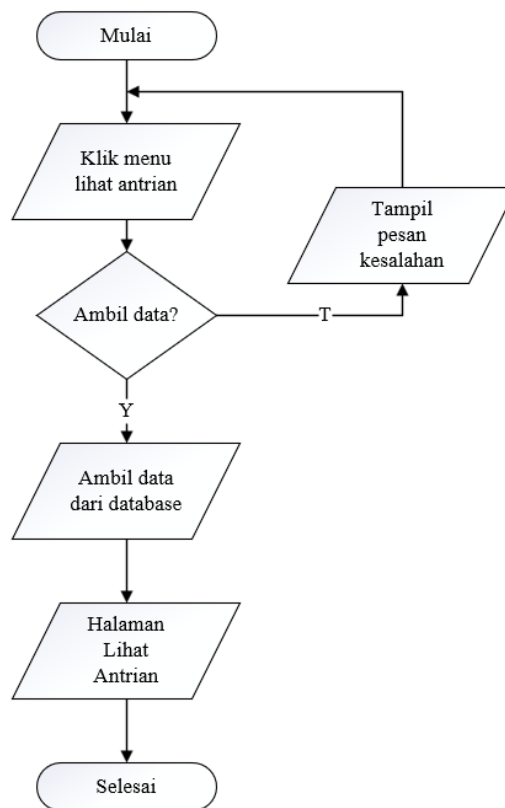
selama dalam hari yang sama

Tabel 3.7 Skenario Use Case Diagram Lihat Antrian

Actor	Reaksi Sistem
-------	---------------

User atau admin menekan tombol lihat antrian	
	Menampilkan nomor antrian
Menekan tombol "OK"	
	Kembali ke halaman awal

Berikut juga gambar diagram *flowchart* Lihat Antrian :



Gambar 3.8 *Flowchart* Lihat Antrian

e) Panggil Antrian

Nama Use Case : Panggil Antrian

Aktor : Admin

Deskripsi : Hanya admin yang dapat memanggil nomor antrian

Pre-Condition : Admin melakukan pemanggilan nomor antrian

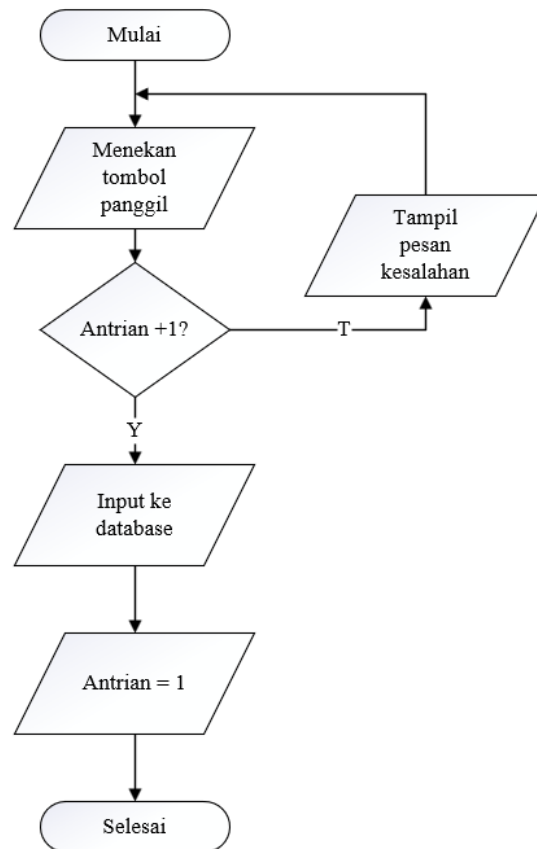
Post-Condition : Nomor antrian bertambah satu, orang yang menunggu

antrian setelahnya menuju loket

Tabel 3.8 Skenario *Use Case Diagram* Panggil Antrian

Actor	Reaksi Sistem
Admin menekan tombol "panggil"	
	Menambahkan jumlah antrian +1
	Menampilkan nomor antrian

Berikut juga gambar diagram *flowchart* Panggil Antrian :



Gambar 3.9 *Flowchart* Panggil Antrian

f) Reset Antrian

Nama *Use Case* : Reset Antrian

Aktor : Admin

Deskripsi : Hanya admin yang dapat mereset nomor antrian
maupun pengambilan nomor antri

Pre-Condition : Admin mereset nomor antrian dan pengambilan nomor

antri menjadi 0

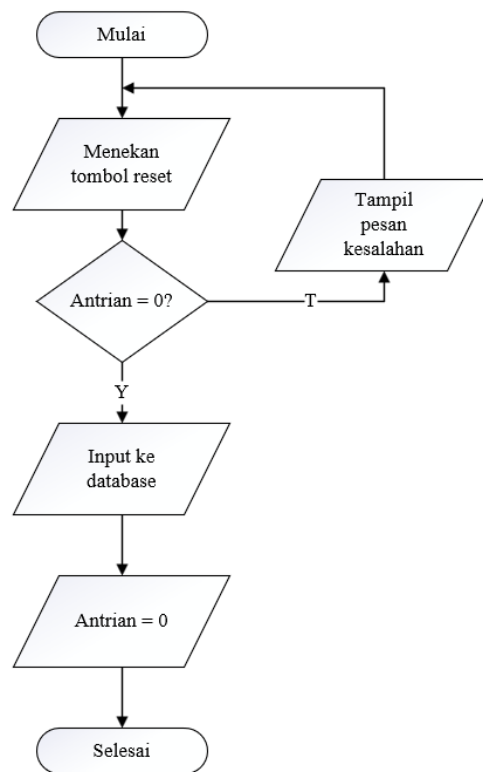
Post-Condition : Sistem akan mereset nomor antrian dan pengambilan

nomor antri menjadi 0

Tabel 3.9 Skenario *Use Case Diagram* Reset Antrian

Actor	Reaksi Sistem
Admin menekan tombol "reset"	
	Mengatur ulang nomor antrian dan pengambilan nomor antri menjadi 0
Menampilkan nomor antrian 0	

Berikut juga gambar diagram *flowchart* Reset Antrian :



Gambar 3.10 *Flowchart* Reset Antrian

g) *Logout*

Nama Use Case : *Logout*

Aktor : *User, Admin*

Deskripsi : Proses ini merupakan tahap akhir yang dilakukan *user* atau dalam menggunakan aplikasi

Pre-Condition : *User* atau admin menekan tombol *logout*, maka akan keluar dari aplikasi dan kembali ke halaman *Login*

Post-Condition : Sistem akan memutuskan koneksi dengan *user* atau admin. Apabila berhasil keluar, maka sistem akan

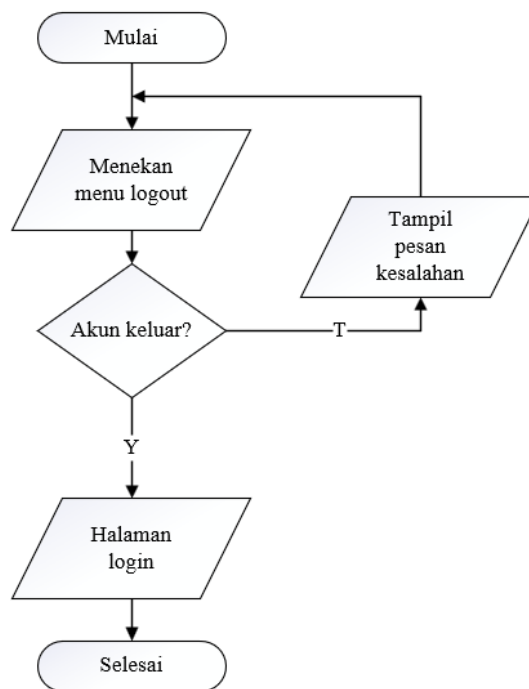
menunjukkan ke halaman *Login*. Jika gagal akan tetap

dihalaman tersebut

Tabel 3.10 Skenario *Use Case Diagram Logout*

Actor	Reaksi Sistem
Admin menekan tombol " <i>Logout</i> "	
	Memutuskan koneksi dengan <i>user</i> atau admin dan menampilkan halaman <i>Login</i>

Berikut juga gambar diagram *flowchart Logout* :



Gambar 3.11 *Flowchart Logout*

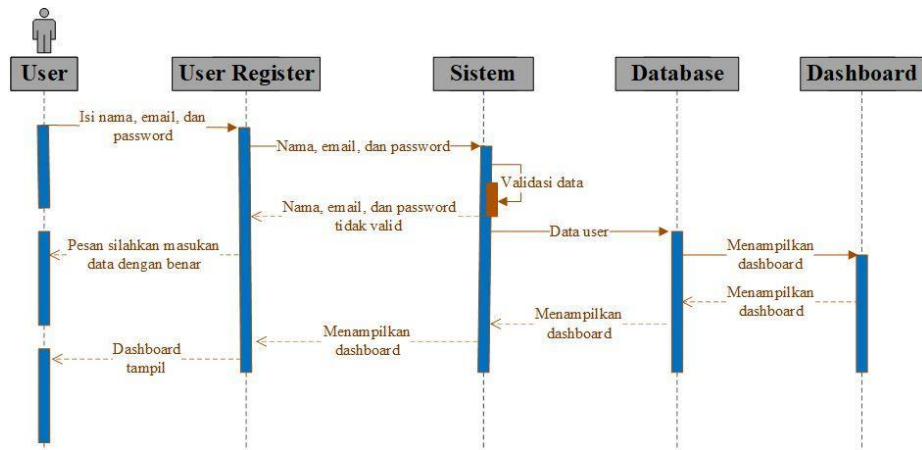
2. *Sequence Diagram*

Sebuah *sequence diagram* menunjukkan urutan interaksi *object* dan *class-class* yang terlibat dalam skenario dan urutan pesan yang diperlukan antara *object* yang dibutuhkan untuk melaksanakan fungsi skenario (Rosad, Yudhana, and Fadlil 2019).

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan antar objek. Berikut merupakan *sequence diagram* pada aplikasi antrian :

a) Registrasi

Pada gambar diagram 3.12 terdapat satu aktor dan empat objek, yaitu *user registrasi*, *sistem*, *database*, dan *dashboard*.

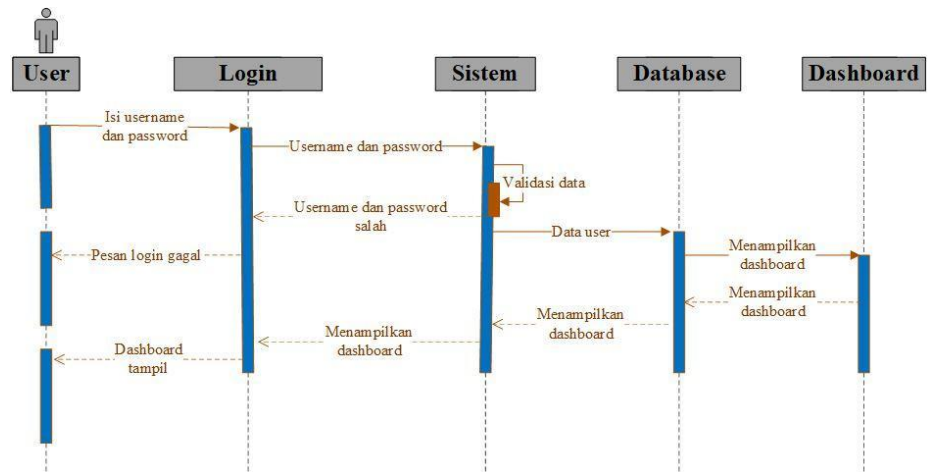


Gambar 3.12 *Sequence Diagram* Registrasi

Pertama-tama *user* akan masuk ke tampilan *user register* dengan mengisi form nama lengkap, email, dan *password*. Sebelum data dikirim ke *database*, data akan divalidasi terlebih dahulu. Jika data yang dimasukan tidak valid, maka pada halaman tersebut akan menampilkan pesan bahwa data yang dimasukan tidak valid. Jika data yang dimasukan valid, maka data akan dikirim ke *database* dan sistem akan menampilkan halaman *Login* pada aplikasi.

b) *Login*

Pada gambar diagram 3.13 terdapat satu aktor dan empat objek, yaitu *Login*, sistem, *database*, dan *dashboard*. Pertama *user* akan masuk ke tampilan *Login* dengan menggunakan *username* dan *password*.

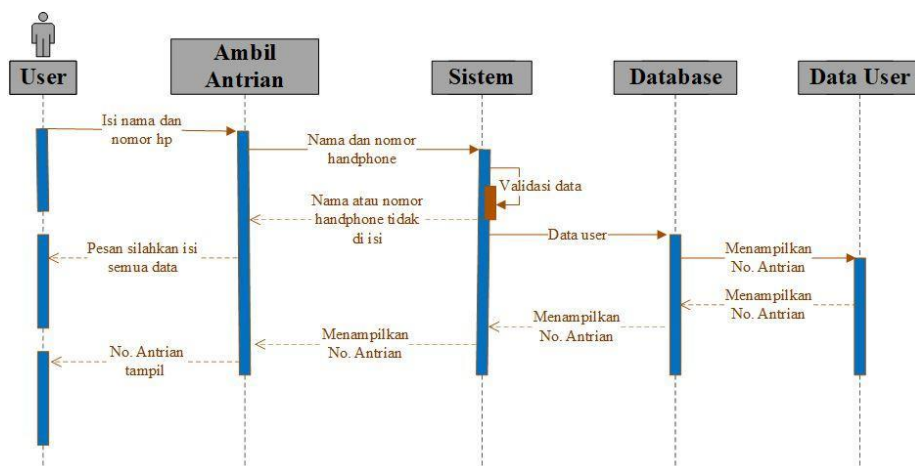


Gambar 3.13 Squence Diagram Login

Kemudian, sistem akan mengirim data tersebut ke *database* untuk divalidasi. Di dalam *database* data *user* akan divalidasi. Jika data yang dimasukan salah, maka akan menampilkan pesan bahwa *username* atau *password* salah. Sedangkan jika data yang dimasukan benar dan valid, maka sistem akan menampilkan halaman home pada aplikasi.

c) Ambil Antrian

Pada gambar diagram 3.14 terdapat satu *actor* (*user*) dan empat objek, yaitu ambil antrian, sistem, *database*, dan data *user*.

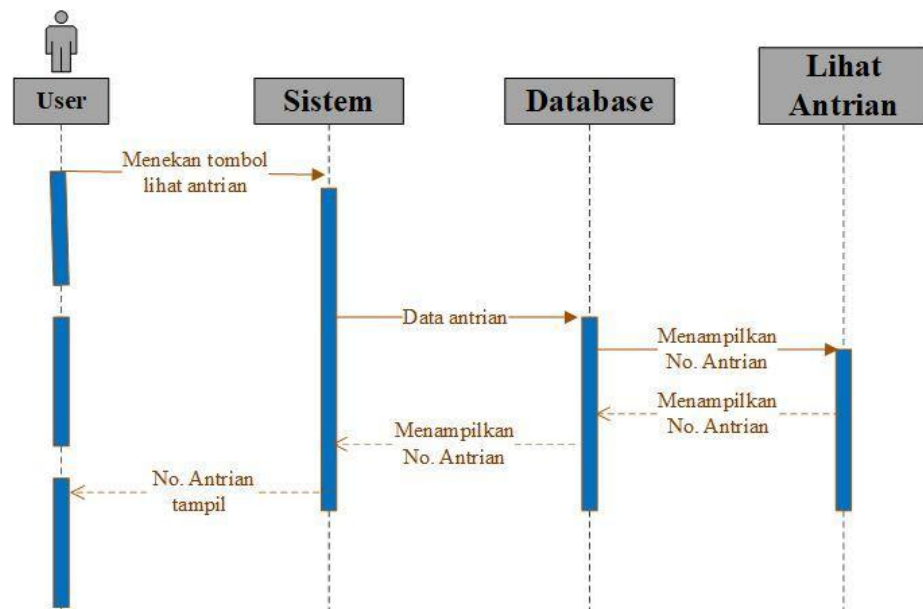


Gambar 3.14 *Sequence Diagram* Ambil Antrian

Pertama *user* akan masuk ke form ambil antrian dengan klik tombol “ambil antrian”. Kemudian *user* mengisi form ambil antrian dengan nama dan nomor handphone. Kemudian sistem akan memvalidasi form tersebut apakah di isi atau tidak. Jika data tidak di isi, maka akan menampilkan pesan bahwa form harus di isi. Sedangkan jika data sudah di isi dan valid, maka data akan di kirimkan ke *database* dan sistem akan menampilkan nomor antrian.

d) Lihat Antrian

Pada gambar diagram 3.15 terdapat *actor (user)* dan tiga objek, yaitu sistem, *database*, dan lihat antrian.

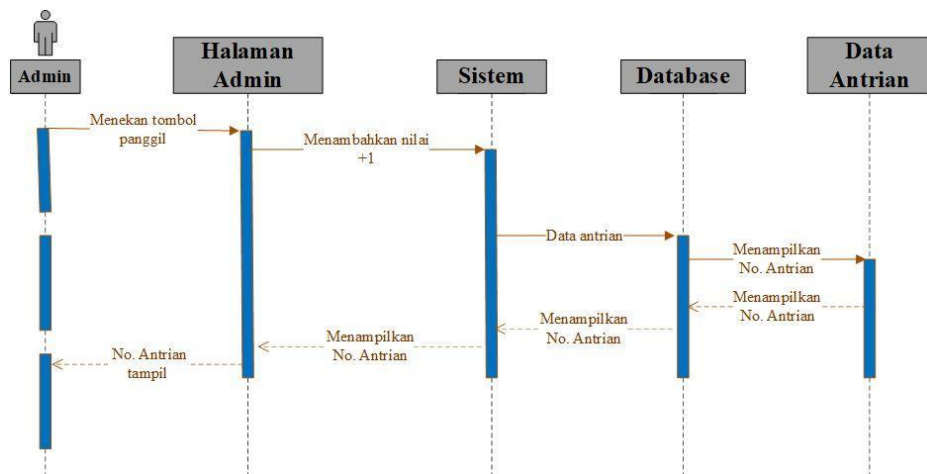


Gambar 3.15 *Sequence Diagram* Lihat Antrian

Pertama *user* akan menekan tombol “lihat antrian”. Kemudian sistem mengambil data antrian dari *database*. Kemudian sistem akan menampilkan nomor antrian *user*.

e) Panggil Antrian

Pada gambar diagram 3.16 terdapat *actor* (admin) dan empat objek, yaitu halaman admin, sistem, *database*, dan data antrian. Pertama admin akan masuk ke tampilan admin dengan klik tombol “halaman admin”.

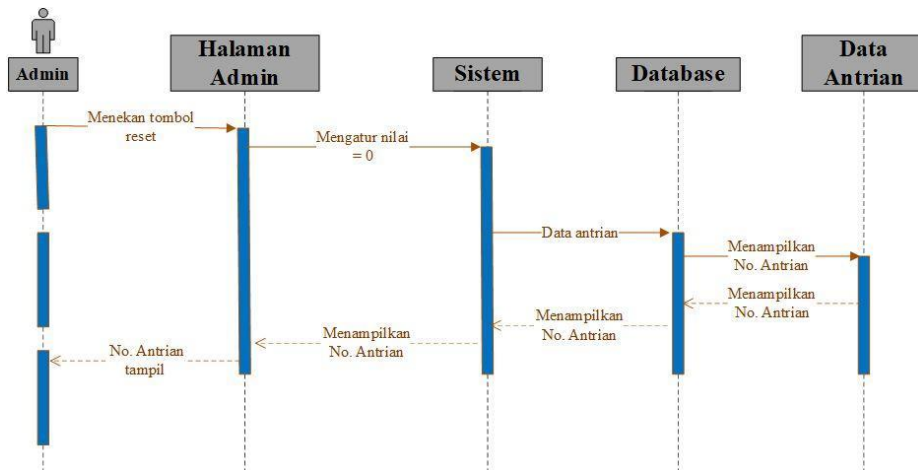


Gambar 3.16 *Sequence Diagram* Panggil Antrian

Kemudian ketika admin memanggil nomor antrian, data panggilan tersebut bernilai +1 dan akan ditambahkan dengan data yang ada pada *database* kemudian dimasukkan ke dalam *database*. Kemudian sistem akan menampilkan nomor antrian.

f) Reset Antrian

Pada gambar diagram 3.17 terdapat satu aktor (admin) dan empat objek, yaitu halaman admin, sistem, *database*, dan data antrian.

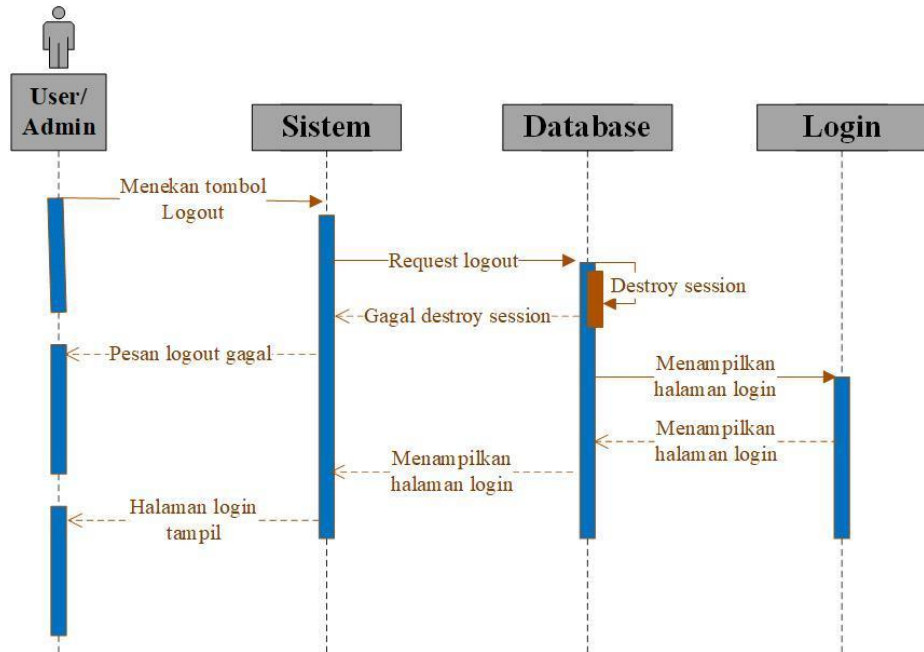


Gambar 3.17 *Sequence Diagram* Reset Antrian

Pertama admin akan masuk ke tampilan halaman admin dengan klik “halaman admin”. Kemudian ketika admin menekan tombol reset pada halaman admin, sistem akan mengatur data antrian menjadi 0 dan kemudian data dimasukan ke dalam *database*. Selanjutnya sistem akan menampilkan data antrian.

g) *Logout*

Pada gambar diagram 3.18 terdapat dua aktor (*user*, admin) dan tiga objek, yaitu sistem, *database*, dan *Login*.

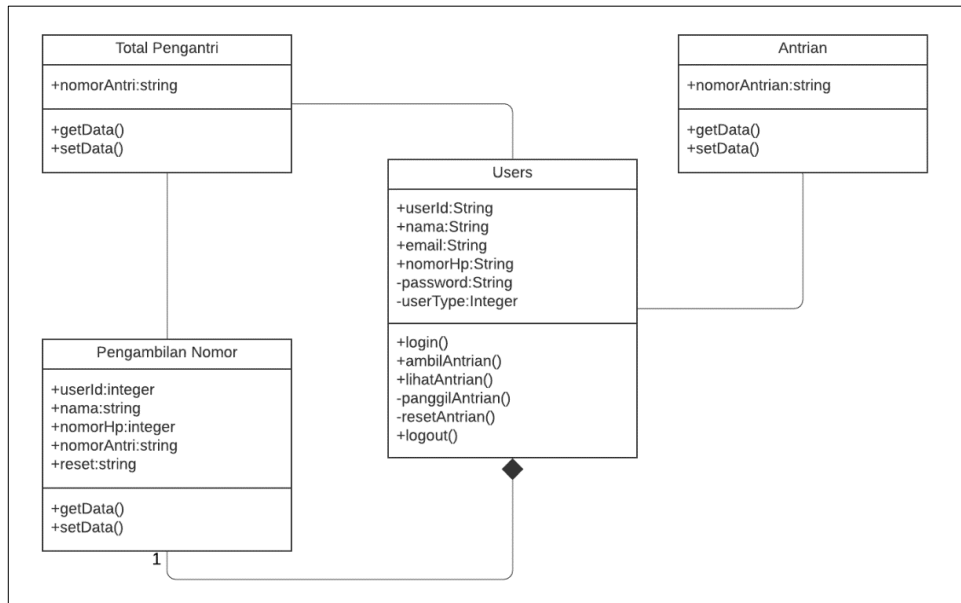


Gambar 3.18 *Sequence Diagram Logout*

Pertama *user* atau *admin* menekan “Logout”. Kemudian sistem akan mengirim permintaan untuk keluar ke *database*. Kemudian *database* akan memproses permintaan tersebut. Jika permintaan gagal diproses, maka akan menampilkan pesan *logout* gagal. Jika berhasil maka akun akan dikeluarkan dan kembali ke halaman *Login*.

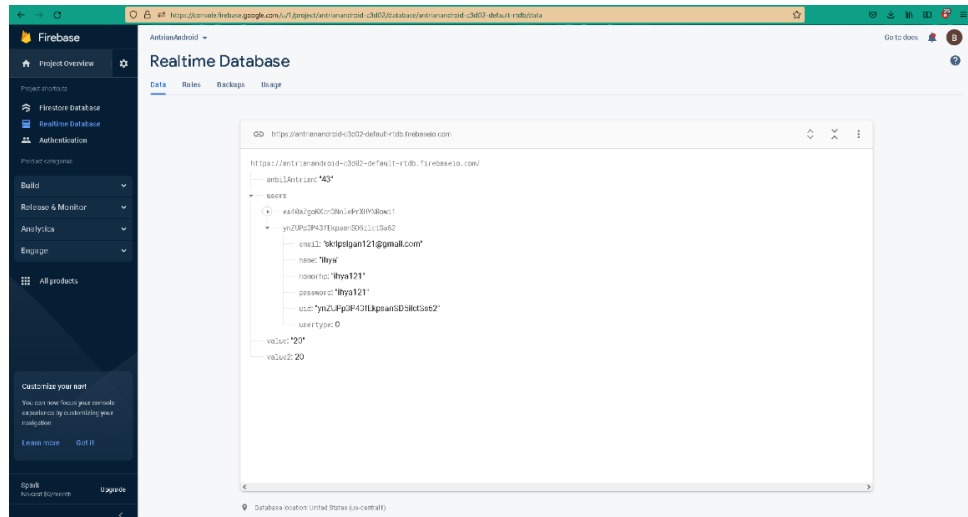
3. *Class Diagram*

Berikut merupakan gambar class diagram dari *software* aplikasi yang akan dirancang :



Gambar 3.19 Class Diagram Aplikasi

Class diagram perancangan perangkat lunak aplikasi pada sistem antrian ini terdiri dari 4 *class* yaitu *Users*, *Antrian*, *Total Pengantri*, *Pengambilan Nomor*. Pada gambar 3.10 juga dijelaskan hubungan antar *class*, seperti pada *class* *Pengambilan Nomor* yang tidak bisa berdiri sendiri tanpa ada *class* *User*. Pada *class* *Pengambilan Nomor* juga terdapat *multiplicity*, yang berarti setiap *user* hanya bisa mempunyai satu nomor antri. *Class* total pengantri juga memiliki hubungan dasar dengan *class* *users* dan *class* pengambilan nomor antri, yang artinya *class* total pengantri dapat mengakses atribut yang ada pada *class* *users* dan pengambilan nomor. Sedangkan pada *class* *Antrian* hanya mempunyai hubungan netral dengan *class* *Users*, yang artinya antara *class* *users* dan *class* antrian bisa saling mengakses atribut dari keduanya.



Gambar 3.20 Database

4. Prototype

Prototype sebagai pemodelan dasar gambaran dari suatu pengembangan program. *Prototype* digunakan sebagai contoh gambaran dari suatu rancangan aplikasi. Berikut merupakan *prototype* pada aplikasi yang akan dibuat pada penelitian ini :

a) *Splash Screen*

Splash Screen merupakan halaman pertama yang ditampilkan saat *user* atau admin membuka aplikasi.

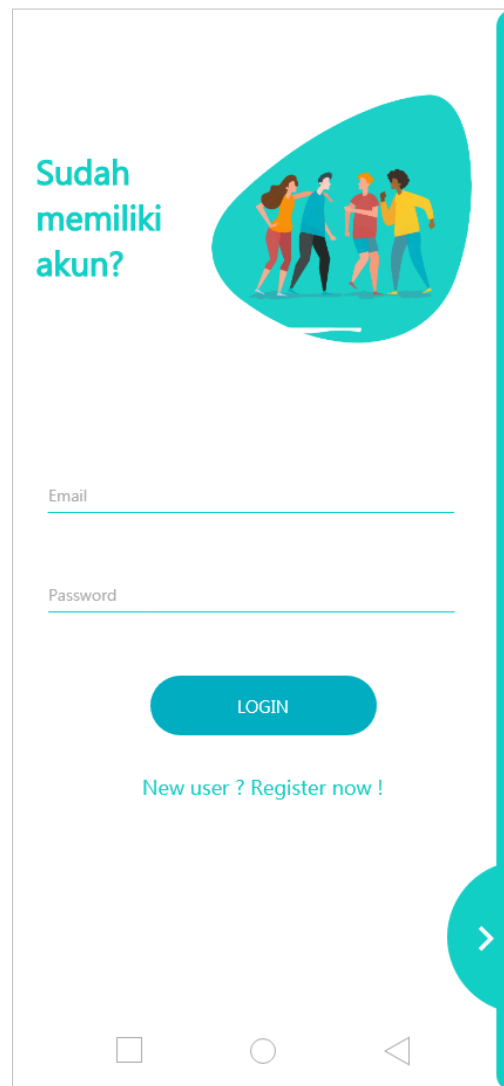


Gambar 3.20 *Prototype Splash Screen*

Splash screen akan tampil selama 3 detik dan kemudian menuju halaman selanjutnya. *Splash screen* pada aplikasi ini berfungsi untuk menampilkan nama atau slogan *aplikasi*, dan juga sebagai hiasan pada aplikasi agar terlihat lebih menarik.

b) Login

Pada gambar *prototype Login* menunjukan form yang harus di isi agar pengguna bisa mengakses aplikasi.

The image shows a mobile application login screen. At the top left, the text "Sudah memiliki akun?" is displayed in a teal color. To the right of this text is an illustration of four people (two women and two men) standing together. Below the text and illustration are two input fields: "Email" and "Password", each with a horizontal line underneath. A teal rounded rectangular button labeled "LOGIN" is positioned below the input fields. Underneath the button, the text "New user ? Register now !" is written in a smaller teal font. At the bottom of the screen, there is a navigation bar with three icons: a square, a circle, and a triangle. A teal vertical bar is on the right side of the screen, featuring a white right-pointing arrow icon.

Gambar 3.21 *Prototype Login*

Pada form terdapat 2 bagian yang harus di isi, pertama *email* yang harus di isi sesuai karakter penulisan *email* dan yang ke dua adalah *password* yang telah didapat setelah mendaftar. Setelah mengisi form dengan benar maka langkah selanjutnya menekan tombol *Login* yang berada di bawah form. Jika belum mempunyai akun, pengguna juga bisa mendaftar dahulu dengan cara menekan

tombol Register now atau menekan tombol arah ke kanan yang ada di kanan bawah halaman.

c) Registrasi

Pada gambar *prorotype* registrasi terdapat 4 form yaitu form nama lengkap, *email*, *password* dan ulangi *password*.

Ini langkah
pertamamu
bersama
kami !

Nama lengkap

Email

Password

Ulangi password

REGISTER

Already have an account ?

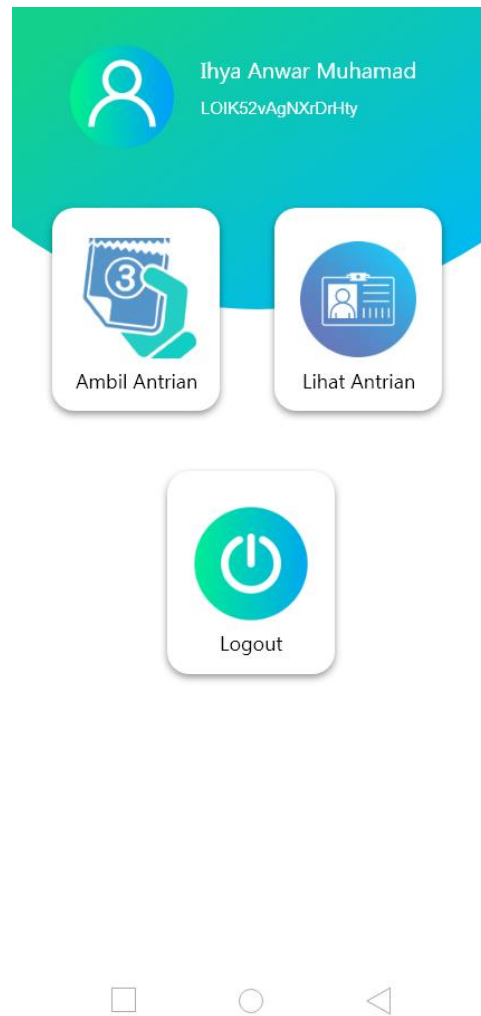
< Isi data dengan benar

Gambar 3.22 *Prototype* Registrasi

Selain form juga terdapat 2 tombol, yaitu tombol register dan tombol kembali yang ada pada kiri bawah halaman. Juga terdapat gambar dan teks pada halaman tersebut. Setelah pengguna mengisi semua form dengan benar, kemudian menekan tombol register dan *data* akan disimpan ke *database*. Jika berhasil maka akan menampilkan halaman *Login*, tetapi jika gagal maka akan menampilkan pesan kesalahan pada halaman registrasi tersebut.

d) Dashboard User

Pada gambar 3.23 menunjukkan bahwa *dashsboard user* mempunyai 3 menu utama yaitu Ambil antrian, Lihat antrian, dan *Logout*.

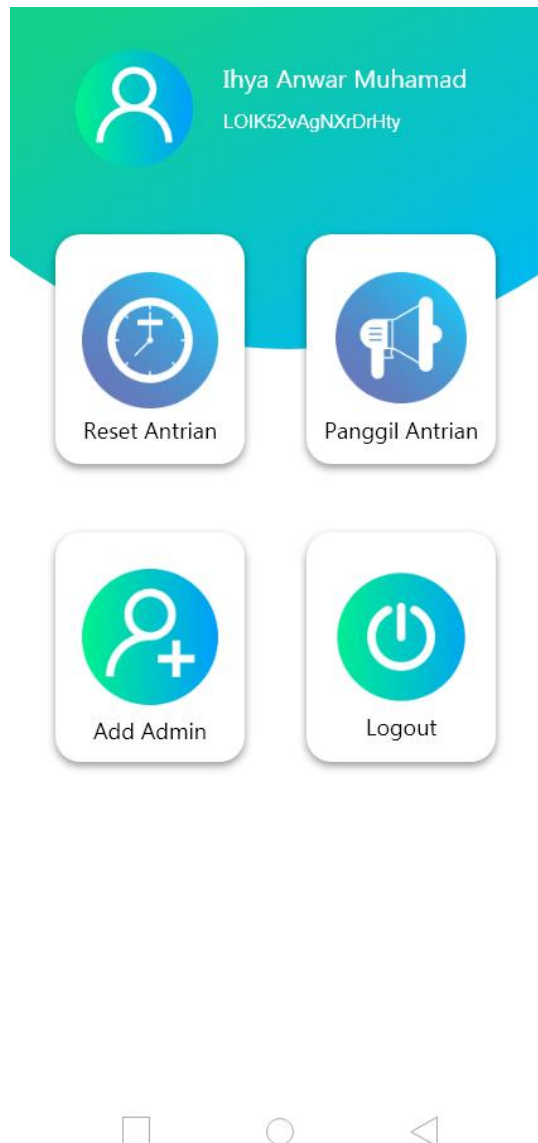


Gambar 3.23 *Prototype Dashboard User*

Pada gambar *prototype Dashboard user* juga terdapat nama dan id *user* pada bagian atas halaman. Ketika *user* ingin mengambil antrian maka tinggal menekan menu ambil antrian dan akan ditujukan ke halaman ambil antrian, jika ingin melihat antrian juga cukup menekan menu lihat antrian. Setelah selesai dan ingin keluar dari aplikasi maka *user* harus menekan menu *logout*.

e) *Dashboard Admin*

Pada gambar *prorotype dashboard* admin memiliki beberapa bagian yang sama dengan *dashboard user*. Pada bagian atas halaman sama terdapat nama dan id dari pengguna tersebut.



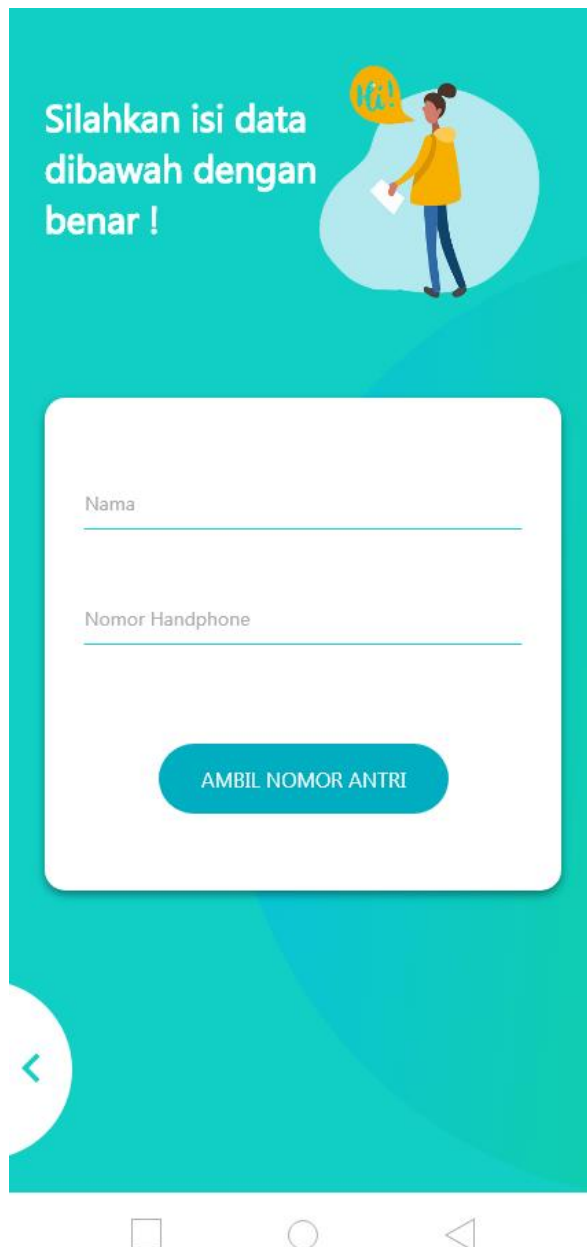
Gambar 3.24 *Prototype Dashboard Admin*

Pada *dashboard* admin terdapat 4 menu yaitu Reset antrian, Panggil antrian, Menambahkan admin, dan *Logout*. Ke empat menu tersebut hanya bisa diakses oleh admin, jika admin ingin *mereset* antrian maka harus masuk ke menu reset antrian, jika ingin memanggil

antrian maka harus masuk ke menu panggil antrian, jika akan menambahkan admin baru maka harus masuk ke menu tambahkan admin, dan ketika ingin keluar maka harus menekan menu *logout*.

f) Ambil Antrian

Pada *prototype* ambil antrian menampilkan teks pengantar kemudian form yang terdiri dari nama dan nomor *handphone*.



Silahkan isi data dibawah dengan benar !

Nama

Nomor Handphone

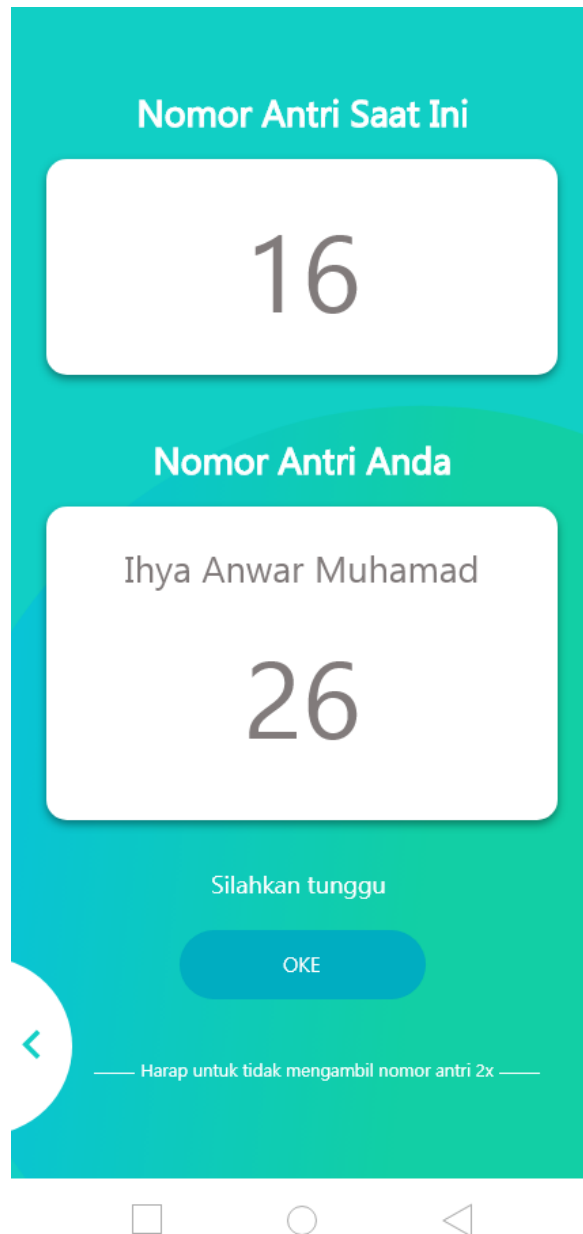
AMBIL NOMOR ANTRI

Gambar 3.25 *Prototype* Ambil Antrian

Pada gambar 3.25 juga terdapat dua tombol aktif yaitu tombol ambil nomor antri dan tombol kembali pada bagian kiri bawah halaman. Ketika pengguna ingin mengambil antrian maka harus mengisi form nama dan nomor *handphone* terlebih dahulu dengan benar, *kemudian* menekan tombol ambil antrian untuk mengambil.

g) Lihat Antrian

Pada gambar *prototype* Lihat antrian berfungsi untuk menampilkan informasi nomor antrian *yang* sedang berlangsung, nama *user* dan nomor antri *user*.



Gambar 3.26 *Prototype* Lihat Antrian

Pada halaman ini *user* tidak bisa merubah informasi yang telah ditampilkan. jika pada *halaman* ini tidak menampilkan informasi nama dan nomor antri, maka *user* harus mengambil nomor antri terlebih dahulu pada halaman ambil antrian untuk mendapatkan nomor antri.

h) Panggil Antrian

Pada *prototype* Panggil antrian menunjukkan informasi berupa angka antrian yang sedang *berlangsung* dan 3 buah tombol yang mempunyai fungsi berbeda.



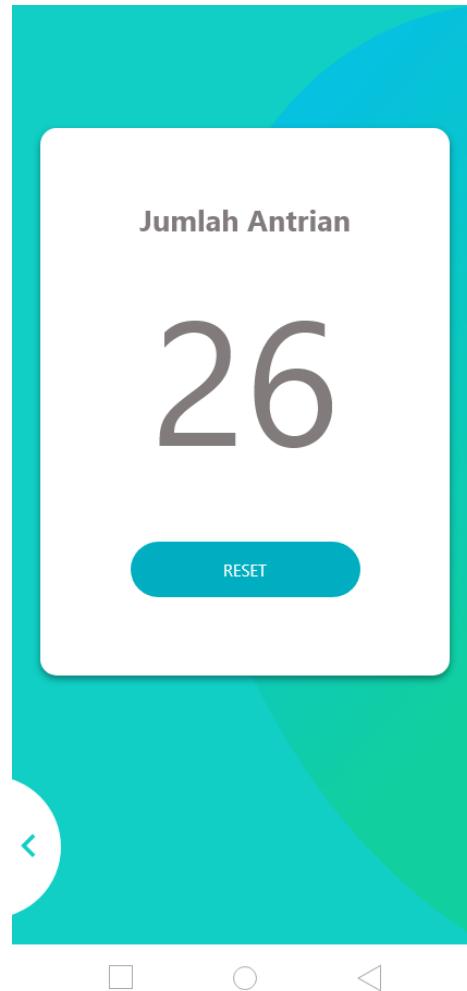
Gambar 3.27 *Prototype* Panggil Antrian

Tombol *tambah* berfungsi sebagai penambahan nomor antrian yang sedang *berlangsung*, ketika admin menekan tombol tambak maka

angka antrian akan bertambah 1. Tombol kurang berfungsi sebagai pengurangan terhadap angka antrian yang sedang berlangsung, misal admin tidak sengaja menekan tombol tambah maka bisa dikembalikan dengan cara menekan tombol kurang. Berikutnya tombol reset yang berfungsi sebagai pengatur ulang nomor antrian yang sedang berlangsung menjadi 0, jika admin menekan tombol reset maka nomor antrian yang sedang berlangsung berubah menjadi 0.

i) Reset Antrian

Pada *prototype* Reset antrian menampilkan informasi jumlah nomor yang telah diambil serta terdapat tombol reset pada bagian bawah informasi.

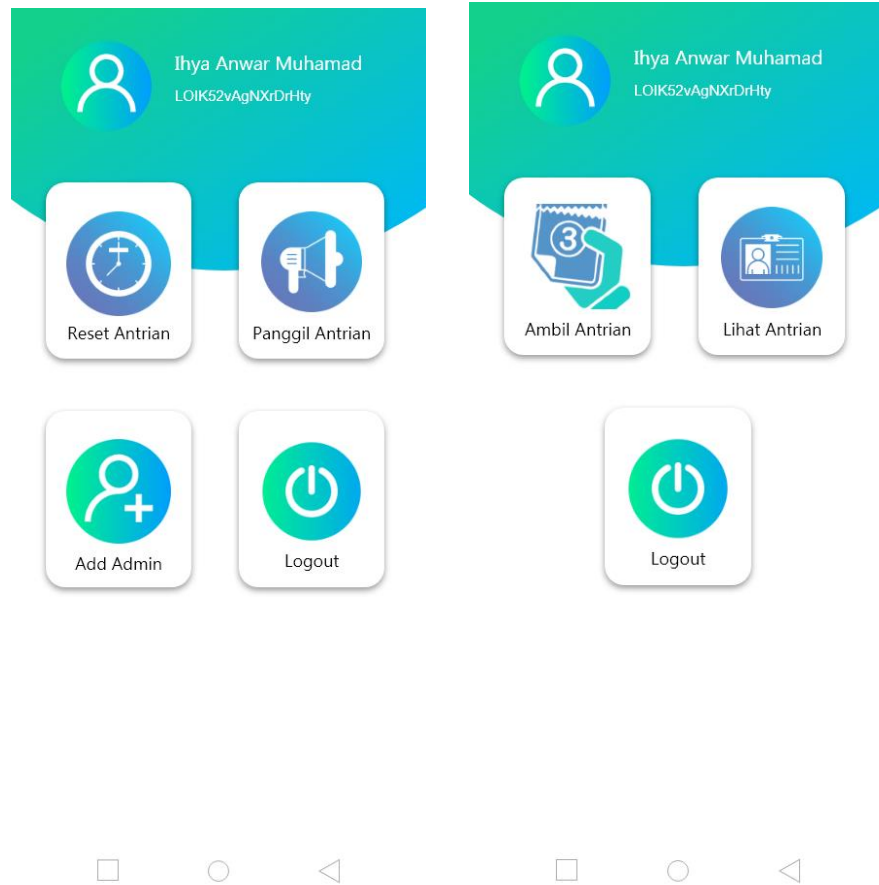


Gambar 3.28 *Prototype* Reset Antrian

Tombol *reset berfungsi* untuk mengatur ulang nomor urut yang telah diambil menjadi 0 kembali. Jadi ketika admin menekan tombol reset, maka jumlah antrian akan berubah menjadi 0.

j) Logout

Pada *gambar prototype logout* sebenarnya terdapat pada *prototype dashboard user* dan *dashboard admin*.



Gambar 3.29 *Prototype Dashboard User dan Admin*

Menu *logout* pada *masing-masing* halaman pada *dashboard user* dan *dashboard admin* mempunyai fungsi yang sama, yaitu mengeluarkan sesi penggunaan aplikasi pada *user* maupun *admin*.

