

BAB III METODOLOGI PENELITIAN

A. 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.

B. Waktu Penelitian

Penelitian dilaksanakan pada semester ganjil genap tahun akademik, tabel waktu penelitian sebagai berikut :

Tabel 1 Jadwal Penelitian

No	Kegiatan	Bulan Ke-					
		1	2	3	4	5	6
1	Pengumpulan data	■					
2	Penentuan Alat dan Bahan		■				
3	Perancangan Alat			■			
4	Perancangan Perangkat Lunak			■	■		
5	Uji Coba					■	
6	Pembuatan Laporan						■

C. Tempat Penelitian

Dalam penelitian ini, pengumpulan data hingga coba alat dilakukan di Bank Sampah Nusantara Al Ihya.

D. Alat dan Bahan

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

1. Kebutuhan Perangkat Keras

Tabel 2 Perangkat Keras

No.	Kebutuhan Perangkat Keras	Keterangan
1.	Laptop	Digunakan untuk mencari referensi, membuat diagram, membuat program <i>hardware</i> maupun program aplikasi.
2.	Sensor Load Cell	Perangkat keras yang digunakan untuk mendeteksi tekanan atau berat sebuah beban
3.	Module HX711	Module hx711 ini berfungsi sebagai amplifier atau penguat hasil pembacaan sensor berat loadcell
4.	Keypad	Digunakan untuk menentukan data teks yang akan dipilih oleh pengguna
5.	Lcd 20x4 i2c	Digunakan untuk menampilkan dan mengirim data dan sebagai interface antara mikrokontroler dengan usernya
6.	Node MCU ESP8266	Mikrokontroler ini digunakan sebagai otak dari alat yang akan dirancang, sebagai pengendali dari perangkat keras yang lain

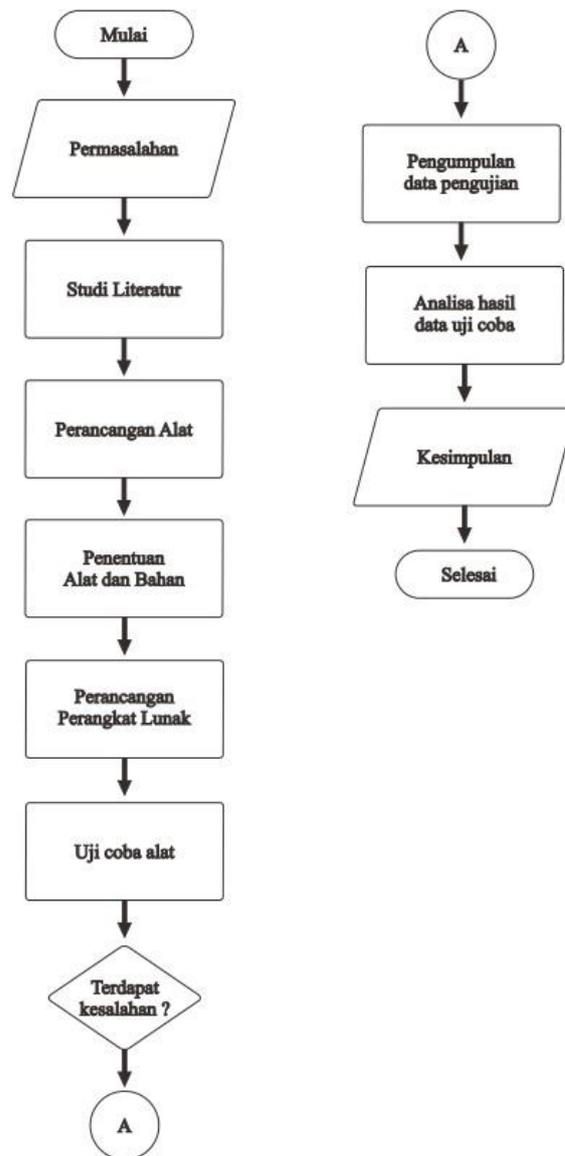
2. Kebutuhan Perangkat Lunak

Tabel 3 Perangkat Lunak

No.	Kebutuhan Perangkat Lunak	Keterangan
1.	Windows 10 64-bit	<i>Operating System</i> pada laptop digunakan untuk pengembangan aplikasi
2.	Firebase	Digunakan untuk menyimpan dan mensinkronkan data secara realtime
2.	Arduino IDE	Digunakan untuk membuat program pada mikrokontroler
3.	Android Studio	Digunakan untuk membuat aplikasi android yang akan digunakan
4.	Firefox	Digunakan sebagai <i>browser</i> utama

3. Alur Penelitian

Dalam melakukan pembuatan sitem 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 menambahkan data nasabah dan melihat data nasabah. Dalam menyusun tugas akhir ini, terdapat beberapa tahapan yang dapat dilihat pada diagram alir dibawah ini:



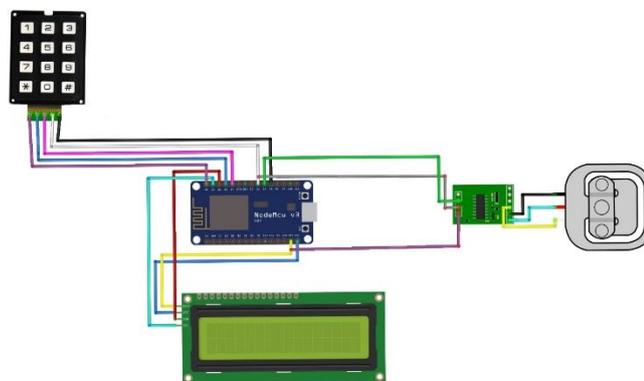
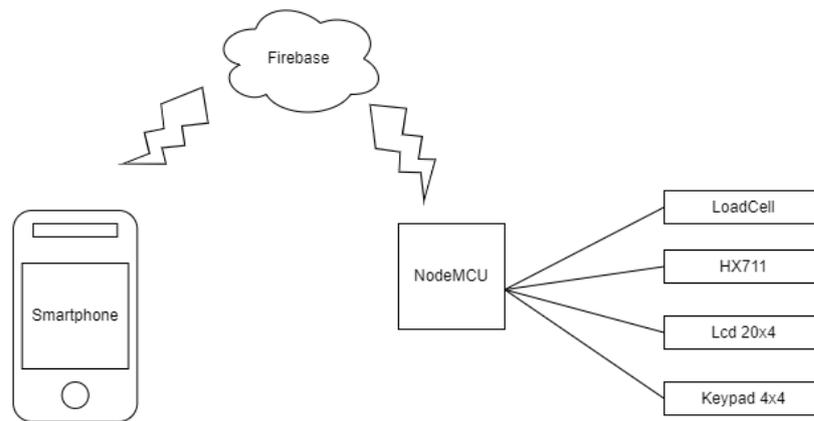
Gambar 1 Diagram Alur

4. Perancangan Sistem

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

5. Perancangan Hardware

Pada perancangan ini digunakan NodeMCU ESP8266 sebagai pengendali dari hardware. NodeMCU dipilih karena sudah terdapat wifi modul didalamnya, sehingga bisa bertugas sebagai client maupun access point.



Gambar 2 Diagram Blok Sistem dan Rangkaian Konektifitas

Gambar 10 merupakan blok diagram alat yang akan dibuat. Ketika sampah diletakan pada sensor Load Cell maka NodeMCU ESP8266 akan membaca data dan dikirimkan ke *Firebase*. Kemudian data akan tampil pada aplikasi bank sampah. Fungsi dari alat dan bahan yang digunakan yaitu:

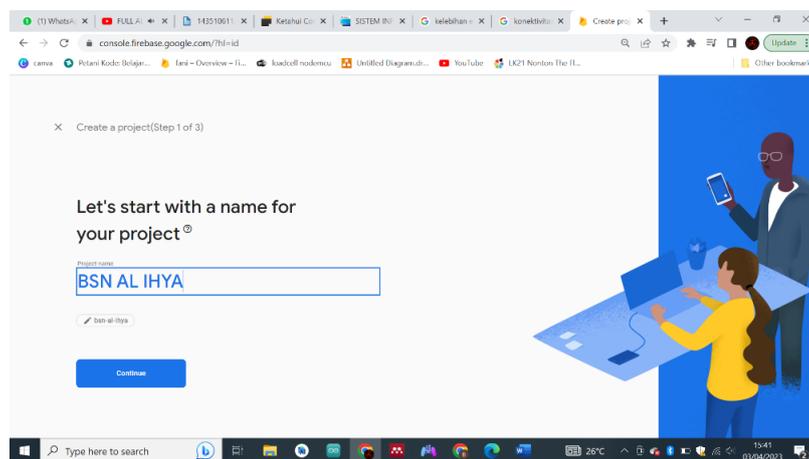
a. NodeMCU ESP8266

NodeMCU ESP8266 pada konsep ini digunakan sebagai *client* dari database firebase. NodeMCU akan selalu membaca data dari sensor *Load Cell*. Alasan menggunakan mikrokontroler jenis NodeMCU ESP8266 karena sudah terdapat modul *wifi* didalamnya.

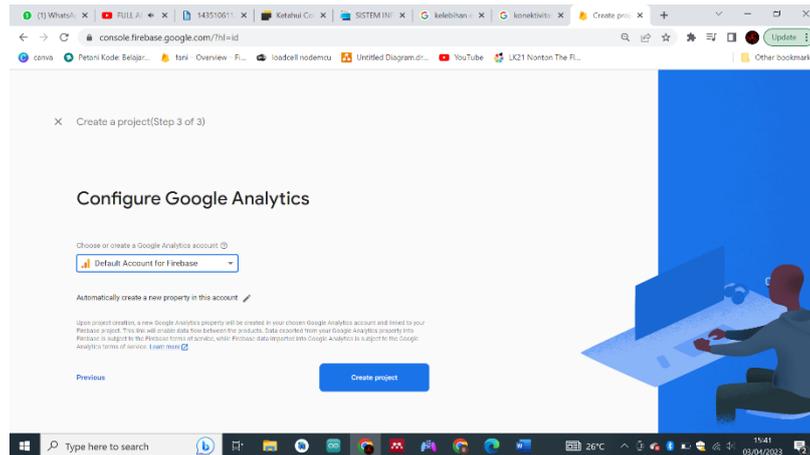
b. Firebase

Firebase digunakan sebagai tempat penyimpanan data pengguna, admin, maupun data nasabah. *Firebase* akan menerima data dari NodeMCU ESP8266 yang kemudian data tersebut ditampilkan pada aplikasi bank sampah. Pada saat menimbang dan ketika data yang berhasil ditimbang dan di simpan, maka *firebase* akan memperbaharui data tersebut dan tampil pada aplikasi.

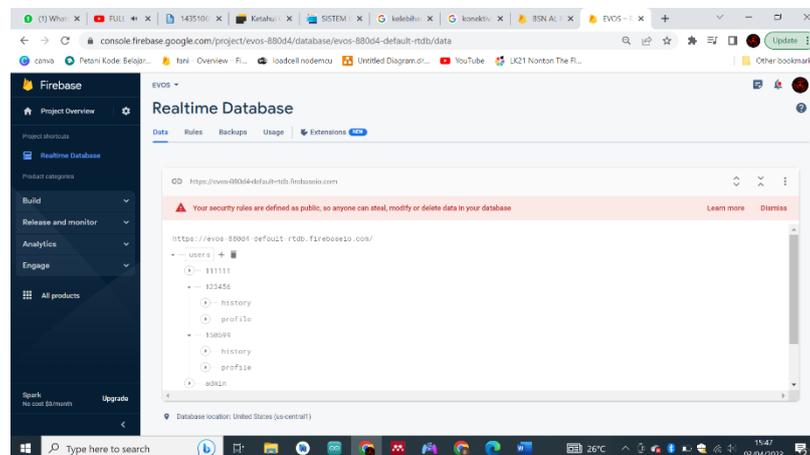
Pertama yang dilakukan untuk membuat database firebase yaitu membuat nama databasenya di *firebase* yang ingin digunakan. Ketikkan nama database di your project kemudian klik *continue*.



Setelah membuat nama database kemudian pilih *default account for firebase* database yang ingin dituju dan selanjutnya klik create project. Kemudian tunggu beberapa menit sesuai jaringan internet lalu klik *continue*.



Setelah memilih default database kemudian ada menu realtime database didalamnya. Kemudian dilanjutkan ke security rules pilih start in locked mode dan klik enable. Kemudian tampilan masuk ke databse. Dan database pun siap untuk digunakan.



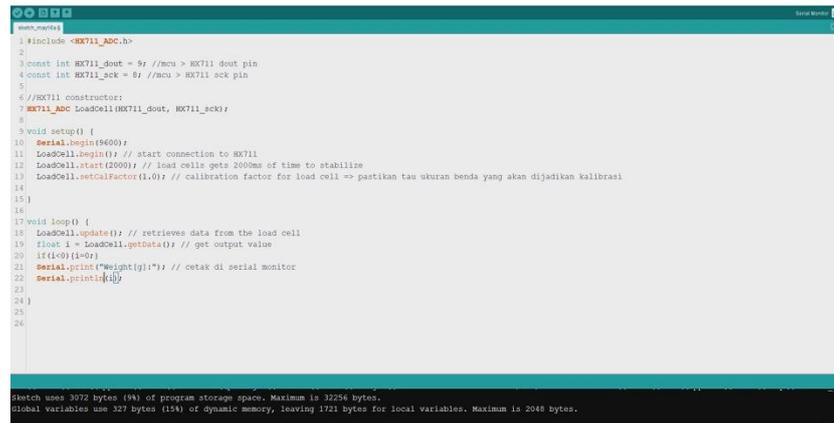
c. *Sensor Load Cell*

Sensor Load Cell merupakan komponen yang berfungsi untuk mendeteksi tekanan atau berat sebuah beban. Secara umum prinsip cara kerja loadcell mirip dengan sensor tekanan, yaitu untuk mengukur tekanan suatu zat. Beban yang diberikan akan mengakibatkan reaksi terhadap elmen logam pada loadcell. Pada timbangan digital loadcell itu sendiri bekerja berdasarkan prinsip mengubah gaya mekanik menjadi energi listrik. Loadcell juga memiliki karakteristik yang dapat diukur tergantung pada jenis logam yang digunakan, dari aspek tersebut maka harus dipastikan untuk memilih sesuai dengan kebutuhan agar mendapatkan parameter yang diinginkan. Salah satu keuntungan menggunakan loadcell yaitu sinyal yang dihasilkan oleh

loadcell terhadap output berbanding lurus dengan eksitasi dan beban yang diterapkan.

Sensor *Load Cell* pada penelitian ini yaitu digunakan untuk menimbang sampah para nasabah.

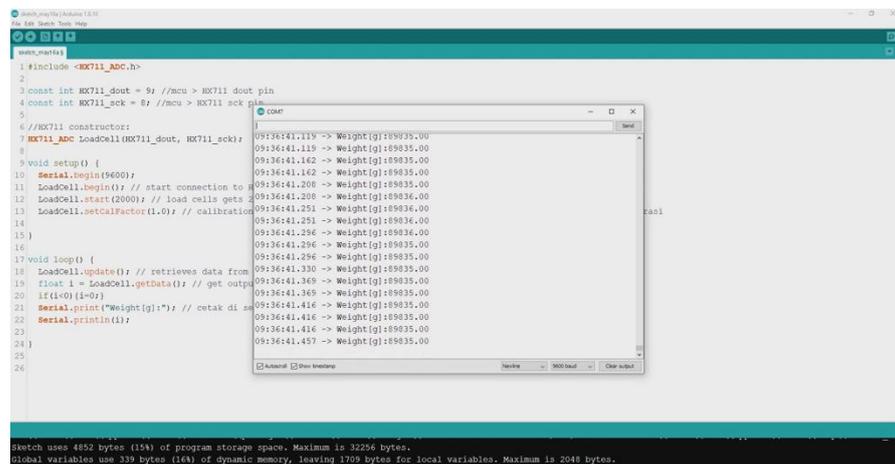
Namun sebelum digunakan loadcell harus dikalibrasi terlebih dahulu, yang pertama disiapkan adalah benda untuk kalibrasinya dengan catatan harus mengetahui berat sebenarnya dari benda tersebut. Untuk kalibrasi ini saya menggunakan smartphone sony dengan berat 198 gram. Selanjutnya yaitu menyiapkan library hx711 pada arduino ide. Sebelum menjalankan program hubungkan sensor loadcell dengan laptop terlebih dahulu.



```
1 #include <HX711_ADC.h>
2
3 const int HX711_dout = 9; //mcu > HX711 dout pin
4 const int HX711_sck = 8; //mcu > HX711 sck pin
5
6 //HX711 constructor:
7 HX711_ADC LoadCell(HX711_dout, HX711_sck);
8
9
10 void setup() {
11   Serial.begin(9600);
12   LoadCell.begin(); // start connection to HX711
13   LoadCell.start(2000); // load cells gets 2000ms of time to stabilize
14   LoadCell.setCalFactor(1.0); // calibration factor for load cell => pastikan tau ukuran benda yang akan dijadikan kalibrasi
15 }
16
17 void loop() {
18   LoadCell.update(); // retrieves data from the load cell
19   float i = LoadCell.getData(); // get output value
20   if(i<0 || i>0)
21     Serial.print("Weight[g]:"); // cetak di serial monitor
22     Serial.println(i);
23 }
24
25
26
```

Sketch uses 3072 bytes (94% of program storage space. Maximum is 32256 bytes.
Global variables use 327 bytes (15% of dynamic memory, leaving 1721 bytes for local variables. Maximum is 2048 bytes.

Kemudian upload program tersebut, ketika sudah berhasil di upload lalu buka serial monitor untuk melihat pembacaan sensor. Kemudian letakan barang yang sudah diketahui beratnya. Maka muncul angka-angka pada serial monitor. Angka tersebut nantinya digunakan untuk menghitung kalibrasi.



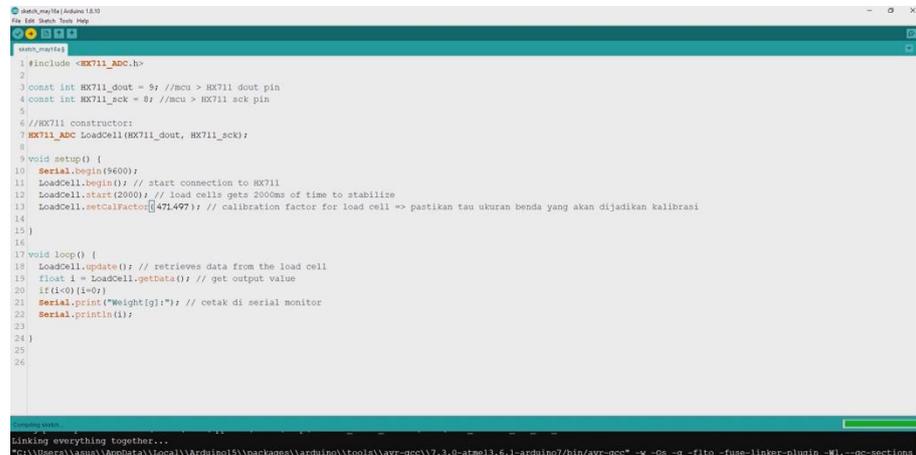
```
1 #include <HX711_ADC.h>
2
3 const int HX711_dout = 9; //mcu > HX711 dout pin
4 const int HX711_sck = 8; //mcu > HX711 sck pin
5
6 //HX711 constructor:
7 HX711_ADC LoadCell(HX711_dout, HX711_sck);
8
9
10 void setup() {
11   Serial.begin(9600);
12   LoadCell.begin(); // start connection to HX711
13   LoadCell.start(2000); // load cells gets 2000ms of time to stabilize
14   LoadCell.setCalFactor(1.0); // calibration
15 }
16
17 void loop() {
18   LoadCell.update(); // retrieves data from
19   float i = LoadCell.getData(); // get output
20   if(i<0 || i>0)
21     Serial.print("Weight[g]:"); // cetak di
22     Serial.println(i);
23 }
24
25
26
```

Serial Monitor Output:

```
09:36:41.119 -> Weight[g]:89035.00
09:36:41.119 -> Weight[g]:89035.00
09:36:41.162 -> Weight[g]:89035.00
09:36:41.208 -> Weight[g]:89035.00
09:36:41.208 -> Weight[g]:89036.00
09:36:41.251 -> Weight[g]:89036.00
09:36:41.251 -> Weight[g]:89036.00
09:36:41.296 -> Weight[g]:89036.00
09:36:41.296 -> Weight[g]:89035.00
09:36:41.330 -> Weight[g]:89035.00
09:36:41.369 -> Weight[g]:89035.00
09:36:41.416 -> Weight[g]:89035.00
09:36:41.416 -> Weight[g]:89035.00
09:36:41.416 -> Weight[g]:89035.00
09:36:41.457 -> Weight[g]:89035.00
```

Sketch uses 4852 bytes (15% of program storage space. Maximum is 32256 bytes.
Global variables use 339 bytes (16% of dynamic memory, leaving 1709 bytes for local variables. Maximum is 2048 bytes.

Langkah selanjutnya yaitu buka kalkulator kemudian angka yang tertera pada serial monitor dibagi dengan nilai berat yang sudah diketahui sebelumnya, (91783/198). Jumlah dari pembagian tersebut (471.497) kemudian dimasukkan pada library lagi lalu upload ulang. Jika sudah maka bisa dipastikan nilai yang terbaca akan sesuai dengan berat yang diketahui.



```
1 #include <HX711_ADC.h>
2
3 const int HX711_dout = 9; //mcu > HX711 dout pin
4 const int HX711_sck = 8; //mcu > HX711 sck pin
5
6 //HX711 constructor:
7 HX711_ADC LoadCell(HX711_dout, HX711_sck);
8
9 void setup() {
10   Serial.begin(9600);
11   LoadCell.begin(); // start connection to HX711
12   LoadCell.start(2000); // load cells gets 2000ms of time to stabilize
13   LoadCell.setCalFactor(471.497); // calibration factor for load cell => pastikan tau ukuran benda yang akan dijadikan kalibrasi
14
15 }
16
17 void loop() {
18   LoadCell.update(); // retrieves data from the load cell
19   float i = LoadCell.getData(); // get output value
20   if(i<0){i=0;}
21   Serial.print("Weight[g]:"); // cetak di serial monitor
22   Serial.println(i);
23
24 }
25
26
```

d. Module HX711

HX711 adalah modul timbangan, yang memiliki prinsip kerja mengkonversi perubahan yang terukur dalam perubahan resistansi dan mengkonversinya ke dalam besaran tegangan melalui rangkaian yang ada (Vin : DC 5V, Arus : 10mA)

Prinsip kerja modul ini adalah membuat cara penampilan hasil pengukuran loadcell pada sebuah tampilan LCD 20X4 dan mengatur timbangan agar dapat menimbang berat sesuai dengan berat yang diinginkan.

e. Lcd Display 20x4 I2c

Penampil data Liquid crystal display (LCD) 20x4 merupakan komponen elektronika, mempunyai fungsi sebagai penampil karakter, angka, huruh bahkan grafik. CMOS logic adalah salah satu teknologi yang digunakan dalam membuat LCD, di mana teknologi ini memantulkan cahaya yang ada pada sekelilingnya dan tidak menghasilkan cahaya (back-lit). Beberapa campuran organik yang berada pada lapisan kaca bening dan elektroda yang transparan berbentuk sevent segment merupakan komponen dasar dalam pembuatan LCD. Saat di trigger tegangan, maka elektroda aktif dengan medan listrik dan molekul-molekul organic yang beebentuk panjang dan silindris secara otomatis menyesuaikan dengan elektroda pada

sevent segmen. LCD ini juga berfungsi untuk menampilkan data dan sebagai interface antara mikrokontroler dengan user-nya.

f. Keypad 4x4

Konstruksi matrik keypad 4×4 diatas cukup sederhana, yaitu terdiri dari 4 baris dan 4 kolom dengan keypad berupa saklar push button yang diletakan disetiap persilangan kolom dan barisnya. Rangkaian matrik keypad diatas terdiri dari 16 saklar push buton dengan konfigurasi 4 baris dan 4 kolom. 8 line yang terdiri dari 4 baris dan 4 kolom tersebut dihubungkan dengan port mikrokontroler 8 bit. Sisi baris dari matrik keypad ditandai dengan nama Row1, Row2, Row3 dan Row4 kemudian sisi kolom ditandai dengan nama Col1, Col2, Col3 dan Col4. Sisi input atau output dari matrik keypad 4×4 ini tidak mengikat, dapat dikonfigurasikan kolom sebagai input dan baris sebagai output atau sebaliknya.

Keypad 4x4 berfungsi untuk menentukan data teks yang akan dipilih oleh pengguna. Selanjutnya, data masukan tersebut akan diolah oleh mikrokontroler. Tombol-tombol pada keypad dilapisi dengan lempengan logam yang telah diberikan kode sesuai nomor pada Keypad 4x4 biasa.

g. *Smartphone Android*

Smartphone Android digunakan sebagai pengirim dan penerima data dari Aplikasi Bank Sampah. Baik untuk *user* maupun admin, harus menggunakan *Smartphone* Android untuk mengakses. Pada *smartphone* yang telah *terinstall* aplikasi Bank Sampah, *user* bisa mulai menggunakan aplikasi tersebut. Langkah pertama *user* harus login terlebih dahulu dengan memasukkan username dan password.

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 login, halaman/dashboard user, dan *logout*. Ketika *user* sudah login aplikasi akan mengirim data tersebut ke *database* firebase.

Kemudian aplikasi akan meneruskan ke halaman user. Pada halaman user terdapat nama pemilik akun dan jumlah total tabungan yang telah diperoleh. Dan ketika *user* menekan menu *logout*, maka aplikasi akan mengeluarkan akun *user* dari aplikasi kemudian akan diteruskan ke halaman *Login*.

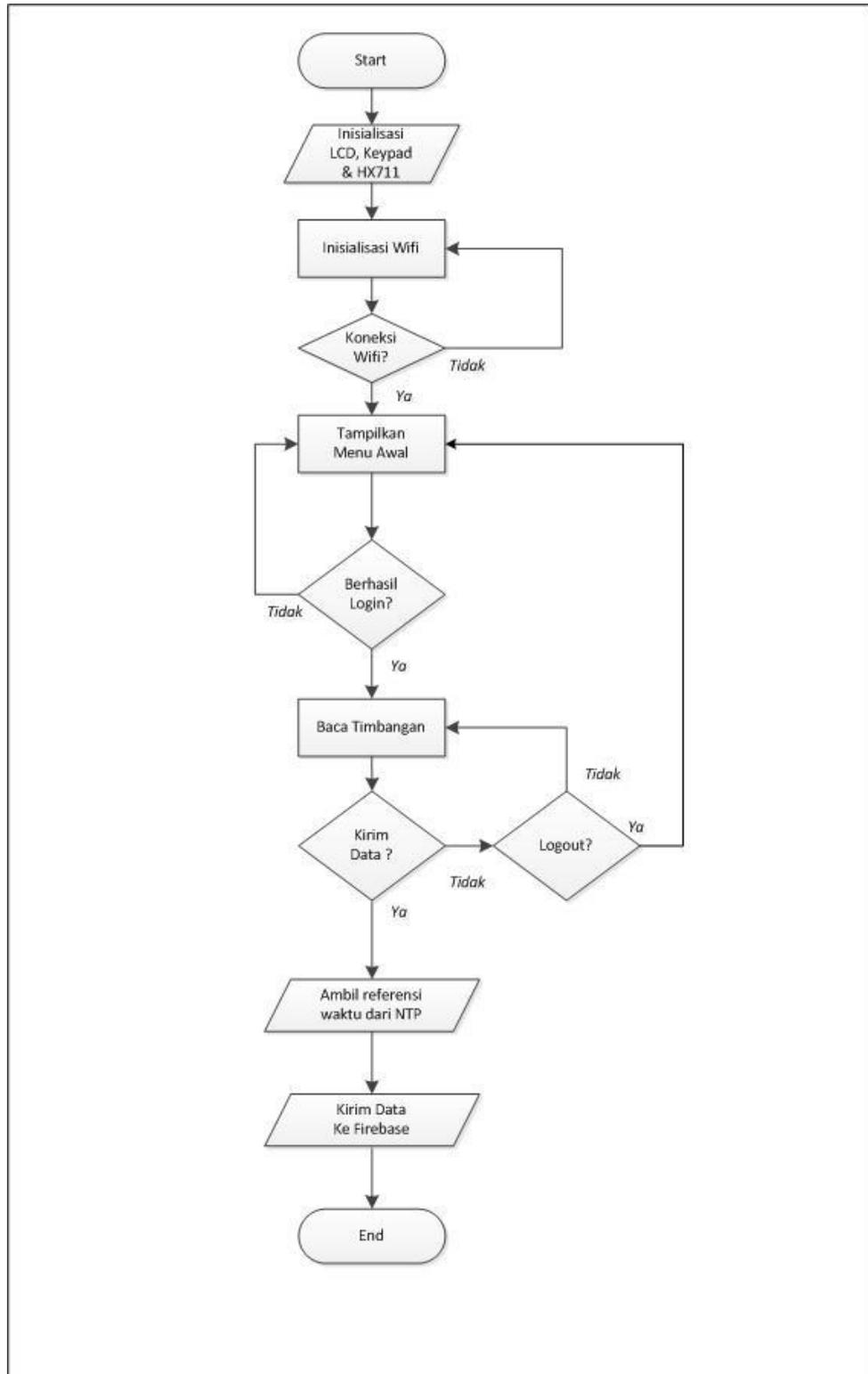
Sedangkan untuk dashboard terdapat beberapa menu diantaranya yaitu *price*, *price* pada aplikasi ini untuk mengganti harga sampah sesuai dengan yang dibutuhkan admin. Kemudian terdapat juga menu lihat dan tambah user. Jika admin ingin melihat data nasabah yang telah menabung, admin harus menekan menu *users*. Dan jika admin ingin menambahkan nasabah baru maka admin menekan menu *add new user*. Kemudian mengisi form yang terdapat pada menu tersebut lalu simpan. Ketika admin ingin keluar dari aplikasi, maka admin harus menekan menu *logout*. Dan aplikasi pun akan mengeluarkan akun admin tersebut kemudian akan diteruskan ke halaman *login*

6. 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 Android IDE. Kode dibuat untuk dapat memfungsikan sensor *Load Cell* dengan NodeMCU ESP8266 agar dapat selalu membaca data dari sensor *Load Cell* tersebut. Kemudian hasil pembacaan data tersebut akan diproses.

Flowchart program pada sistem yang akan dirancang bertujuan untuk menjadikan tanda bahwa program tersebut dapat dioperasikan dan siap untuk dijalankan. Diawali dengan mulai kemudian inisialisasi hardware yang digunakan yaitu *Lcd*, *Keypad* dan *Module Hx711*. Selanjutnya yaitu inisialisasi *Wifi* apakah sudah terkoneksi atau belum, jika belum maka akan terus mengulang koneksi, dan jika sudah maka akan menampilkan menu awal. Setelah menu awal tampil langkah selanjutnya yaitu login dengan password yang telah diperoleh, apabila berhasil login maka selanjutnya yaitu membaca hasil dari timbangan sampah yang telah diletakan diatas sensor. Kemudian setelah membaca timbangan apakah hasil dari pembacaan timbangan tersebut akan dikirim atau tidak, jika tidak maka *logout* dengan menekan tombol "0" pada *keypad*, jika iya selanjutnya akan mengambil referensi waktu dari *NTP (Network Time Protokol)*. *NTP* digunakan untuk memberikan history yang berupa jam, tanggal, bulan, tahun pada saat menimbang agar lebih detail. Selanjutnya

data dikirimkan kedalam firebase, kemudian selesai. Berikut adalah gambar *flowchart* program sistem.



Gambar 3 Flowchart Program

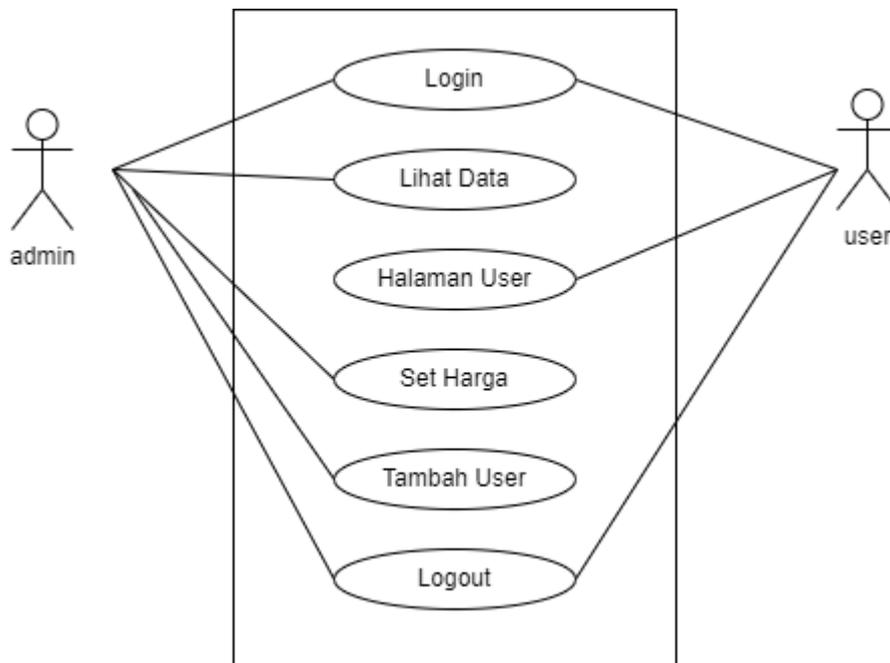
7. Perancangan Software Aplikasi

Setelah perancangan *software* alat, maka selanjutnya dengan perancangan *software* aplikasi. Adapun perancangan pada tahap ini meliputi:

a. 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 bank sampah pada gambar berikut:



Gambar 4 Use Case Diagram Aplikasi Bank Sampah

1) Login

Nama *Use Case* : Login

Aktor : User, Admin

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

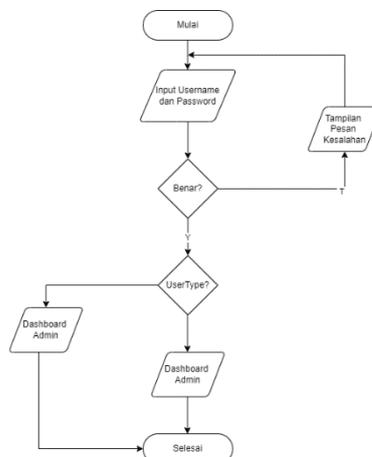
Pre-Condition : User atau admin membuka aplikasi kemudian mengisi form login dengan *username* dan *password* dengan benar

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 4 Skenario *Use Case* Diagram Login

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

Berikut juga gambar diagram *flowchart* pada form login:



Gambar 5 *Flowchart* Login

2) Lihat Data

Nama *Use Case* : Lihat Data

Aktor : Admin

Deskripsi : Admin dapat melihat data nasabah yang telah mendaftar pada aplikasi bank sampah

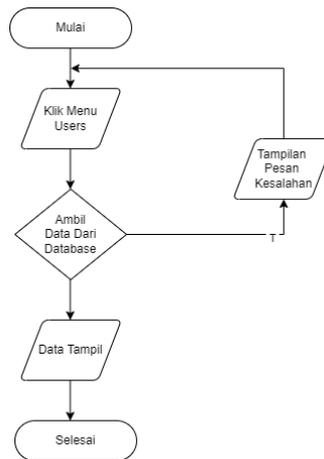
Pre-Condition : Admin melihat data nasabah pada halaman Users

Post-Condition : Admin dapat melihat data nasabah kapanpun selama berada didalam aplikasi

Tabel 5 Skenario *Use Case* Diagram lihat data

Actor	Reaksi Sistem
admin menekan menu Users	
	Menampilkan data nasabah

Berikut juga gambar diagram *flowchart* lihat data :



Gambar 6 *Flowchart* Lihat Data

3) Halaman User

Nama *Use Case* : Halaman User

Aktor : User

Deskripsi : User dapat melihat history tabungan yang telah dikumpulkan atau ditabung pada bank sampah

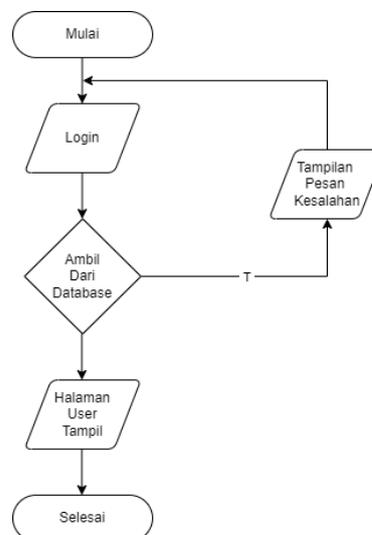
Pre-Condition : User melihat tabungan yang telah ditabung

Post-Condition : Sistem akan menampilkan history tabungan User yang telah diperoleh

Tabel 6 Skenario Use Case Diagram Halaman User

Actor	Reaksi Sistem
Setelah user login maka akan otomatis menampilkan halaman user	
	Menampilkan halaman user

Berikut juga gambar diagram *flowchart* halaman user :



Gambar 7 *Flowchart* Halaman User

4) Set Harga

Nama *Use Case* : Set Harga

Aktor : Admin

Deskripsi : Admin dapat mengganti harga sampah pada aplikasi sesuai dengan yang dibutuhkan

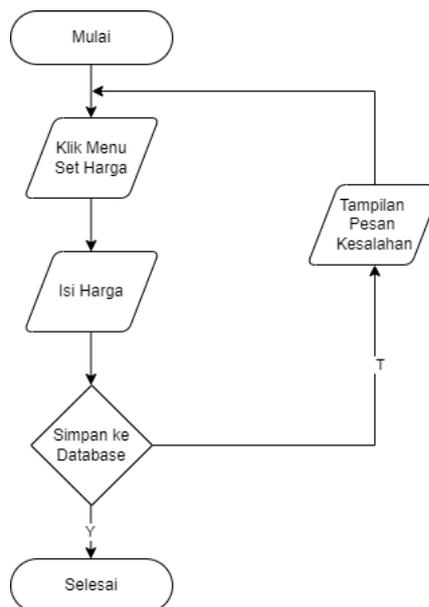
Pre-Condition : Admin membuka aplikasi kemudian menekan menu price

Post-Condition : Sistem akan menampilkan halaman set harga

Tabel 7 Skenario *Use Case* diagram Set Harga

Actor	Reaksi Sistem
Admin menekan menu “set harga”	
	Menampilkan halaman set harga

Berikut juga gambar diagram *flowchart* set harga :



Gambar 8 *Flowchart* Set Harga

5) Tambah User

Nama *Use Case* : Tambah User

Aktor : Admin

Deskripsi : Admin dapat menambahkan user baru ketika ada nasabah yang akan menabung

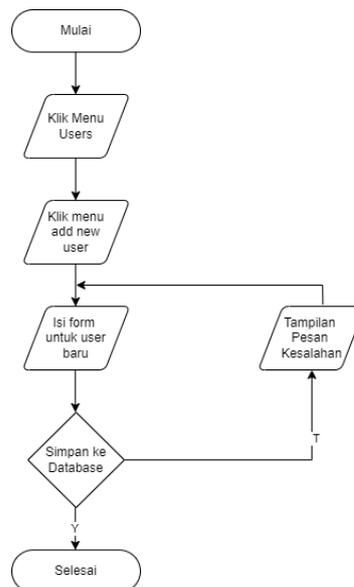
Pre-Condition : Admin membuka aplikasi kemudian menekan menu users kemudian klik menu add new user

Post-Condition : Sistem akan menampilkan halaman users

Tabel 8 Skenario Use Case Diagram tambah user

Actor	Reaksi Sistem
Admin menekan menu “Users”, kemudian klik menu add new user	
	Menampilkan halaman form untuk user baru

Berikut juga gambar diagram *flowchart* tambah user :



Gambar 9 *Flowchart* Tambah User

6) Logout

Nama *Use Case* : Logout

Aktor : User, Admin

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

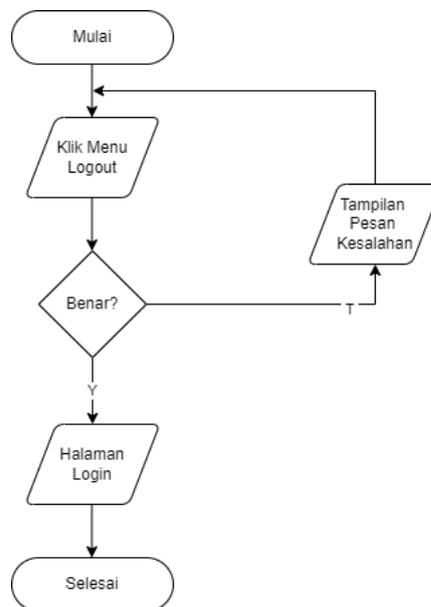
Pre-Condition : User atau Admin menekan tombol logout, maka akan keluar dari aplikasi dan kembali ke halaman login

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 9 Skenario *Use Case diagram Logout*

Actor	Reaksi Sistem
User atau Admin menekan tombol “Logout”	
	Memutuskan koneksi dengan user atau admin dan menampilkan halaman login

Berikut juga gambar diagram *flowchart* logout :



Gambar 10 *Flowchart Logout*

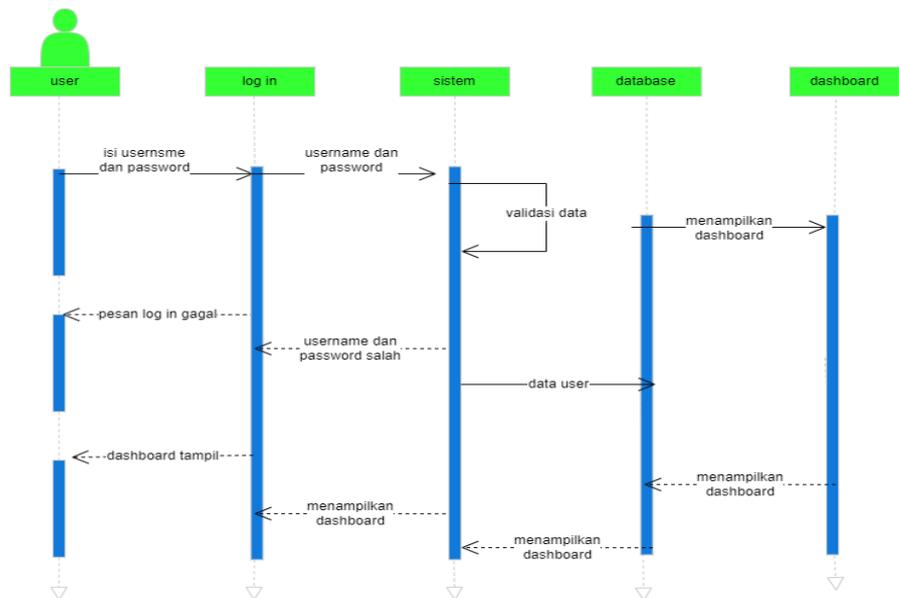
b. Sequence Diagram

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 (Rosyad, 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 bank sampah :

1) Login

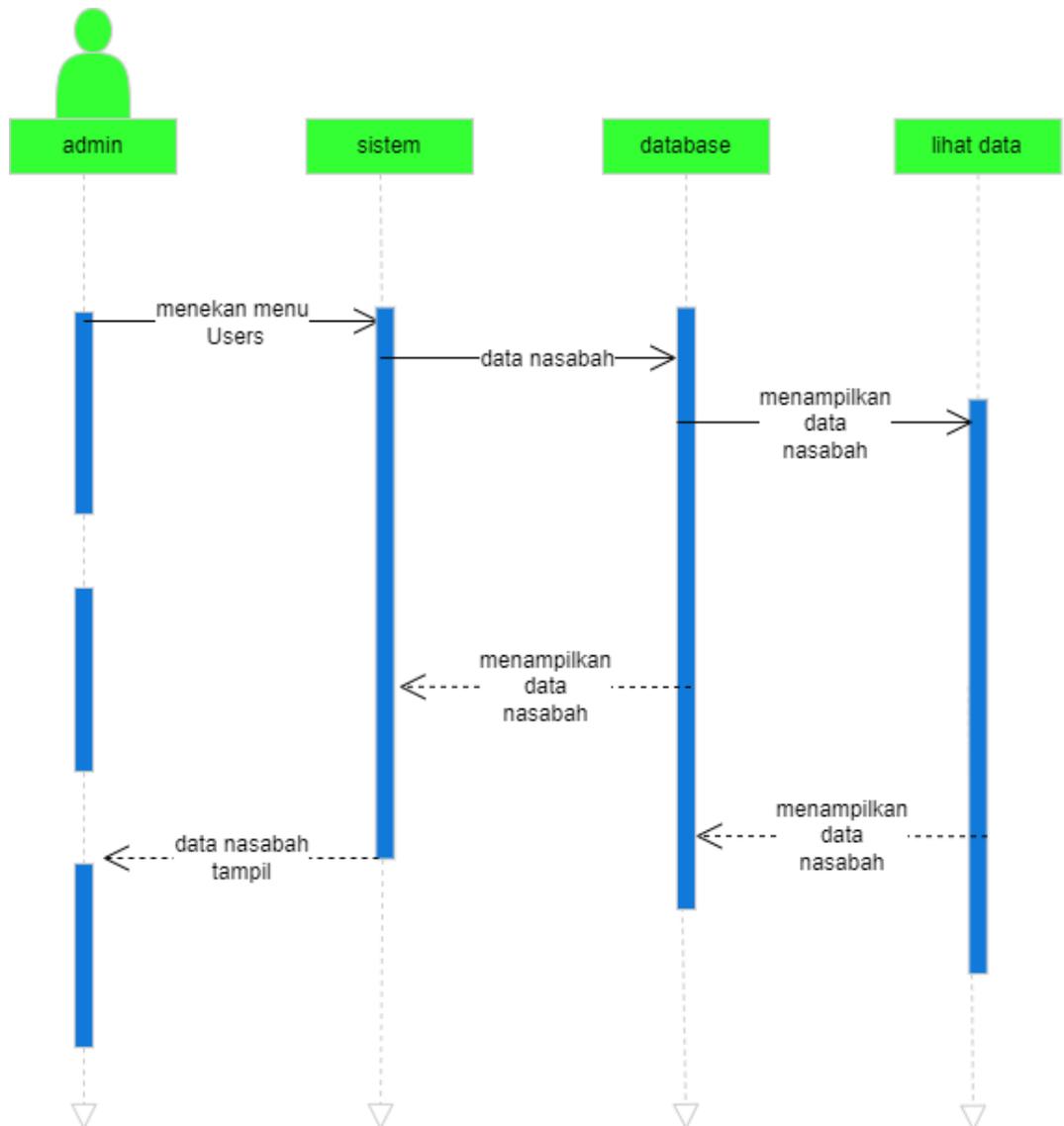
Pada gambar diagram 3.11 terdapat satu aktor dan empat objek, yaitu login, sistem, sistem, databse dan *dashboard*. Pertama user akan masuk ke tampilan login dengan menggunakan *username* dan *password*. Kemudian, sistem akan mengirim data tersebut ke database untuk divalidasi. Di dalam database data dan 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.



Gambar 3.11 *Sequence Diagram* Login

2) Lihat Data

Pada gambar diagram 3.12 terdapat aktor (admin) dan tiga objek yaitu sistem, database, dan lihat data. Pertama admin akan menekan tombol “lihat data”. Kemudian sistem mengambil data nasabah dari database, lalu sistem akan menampilkan data nasabah.

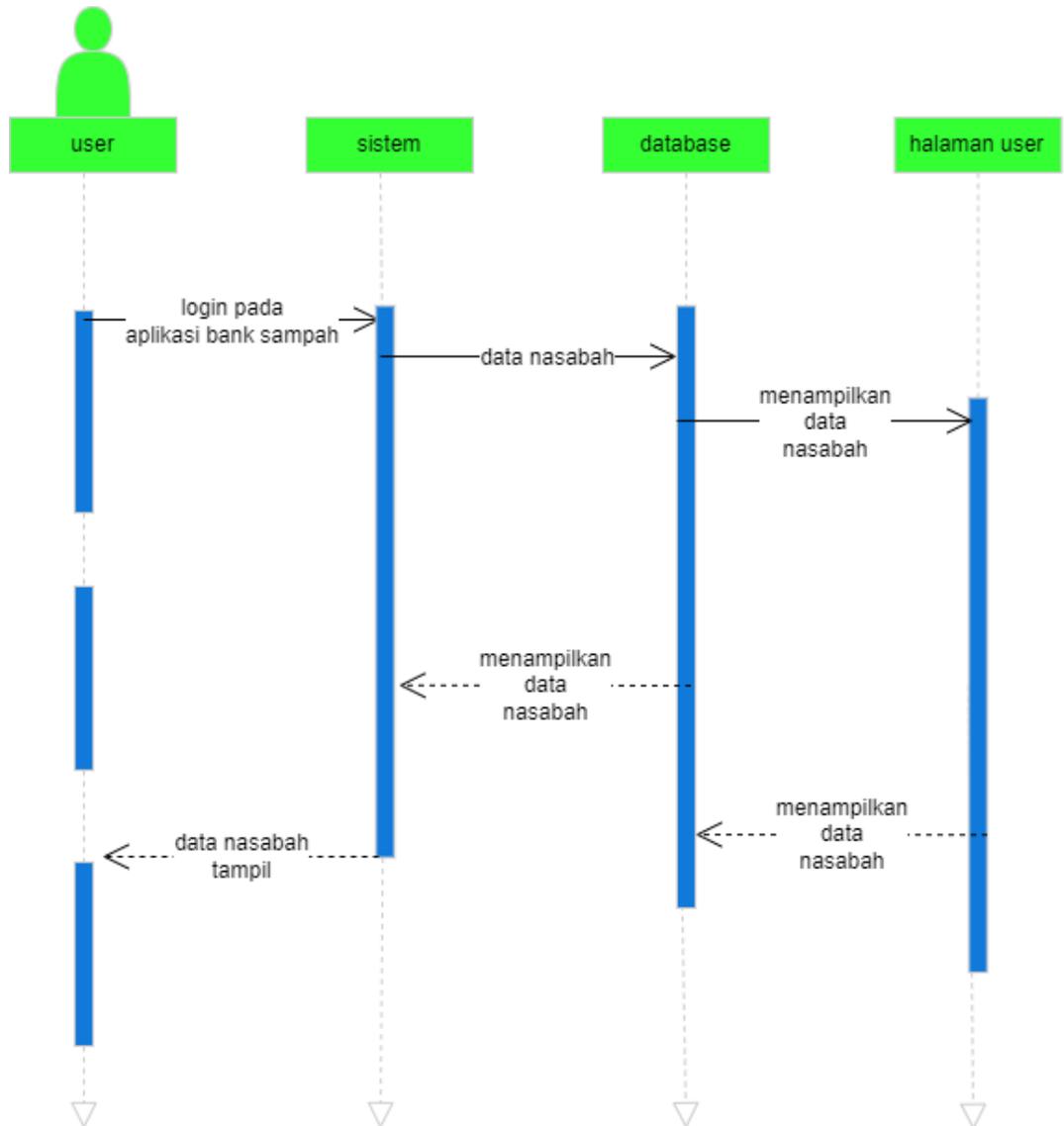


Gambar 3.12 *Sequence Lihat Data*

3) Halaman User

Pada gambar 3.13 terdapat aktor (user) dan tiga objek, yaitu sistem, database, dan halaman user. Pada saat user

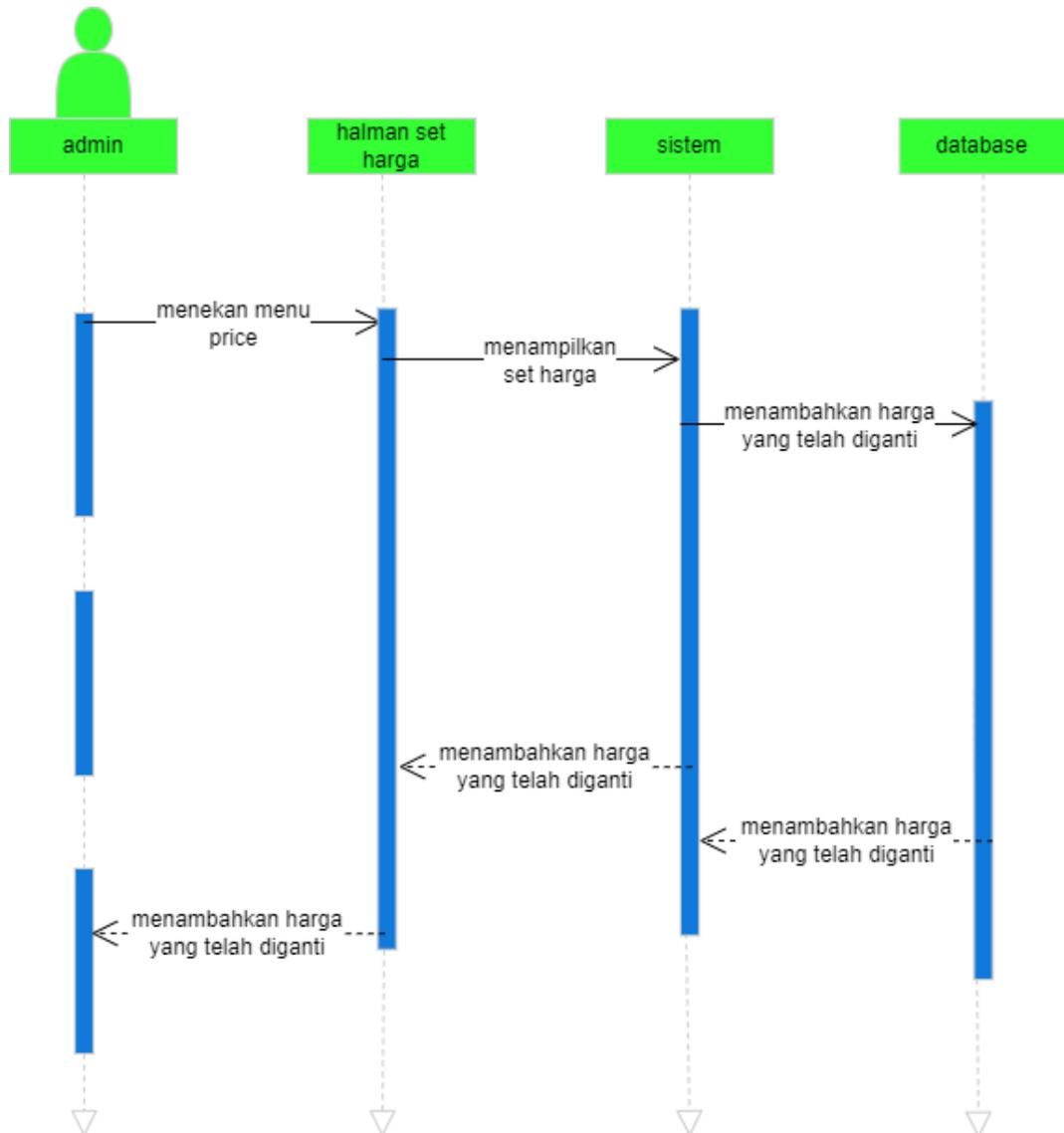
membuka aplikasi bank sampah, maka langsung muncul halaman user sesuai dengan username masing-masing.



Gambar 3.13 *Sequence Halaman User*

4) Set Harga

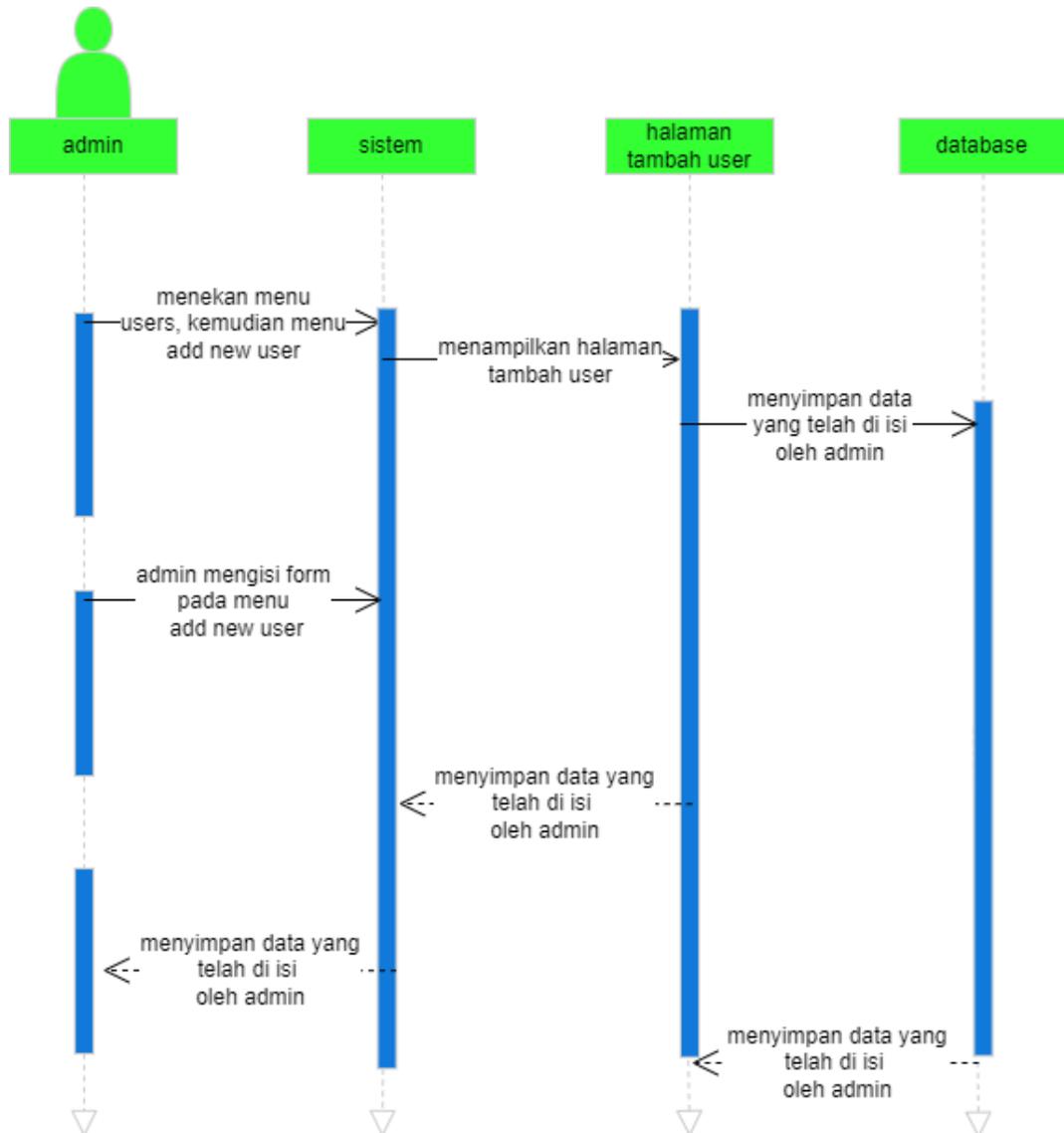
Pada gambar diagram 3.14 terdapat aktor (user) dan tiga objek, yaitu halaman set harga, sistem dan database. Admin menekan menu price pada aplikasi, kemudian memasukkan harga yang dibutuhkan lalu simpan.



Gambar 3.14 *Sequence Diagram* Set Harga

5) Tambah User

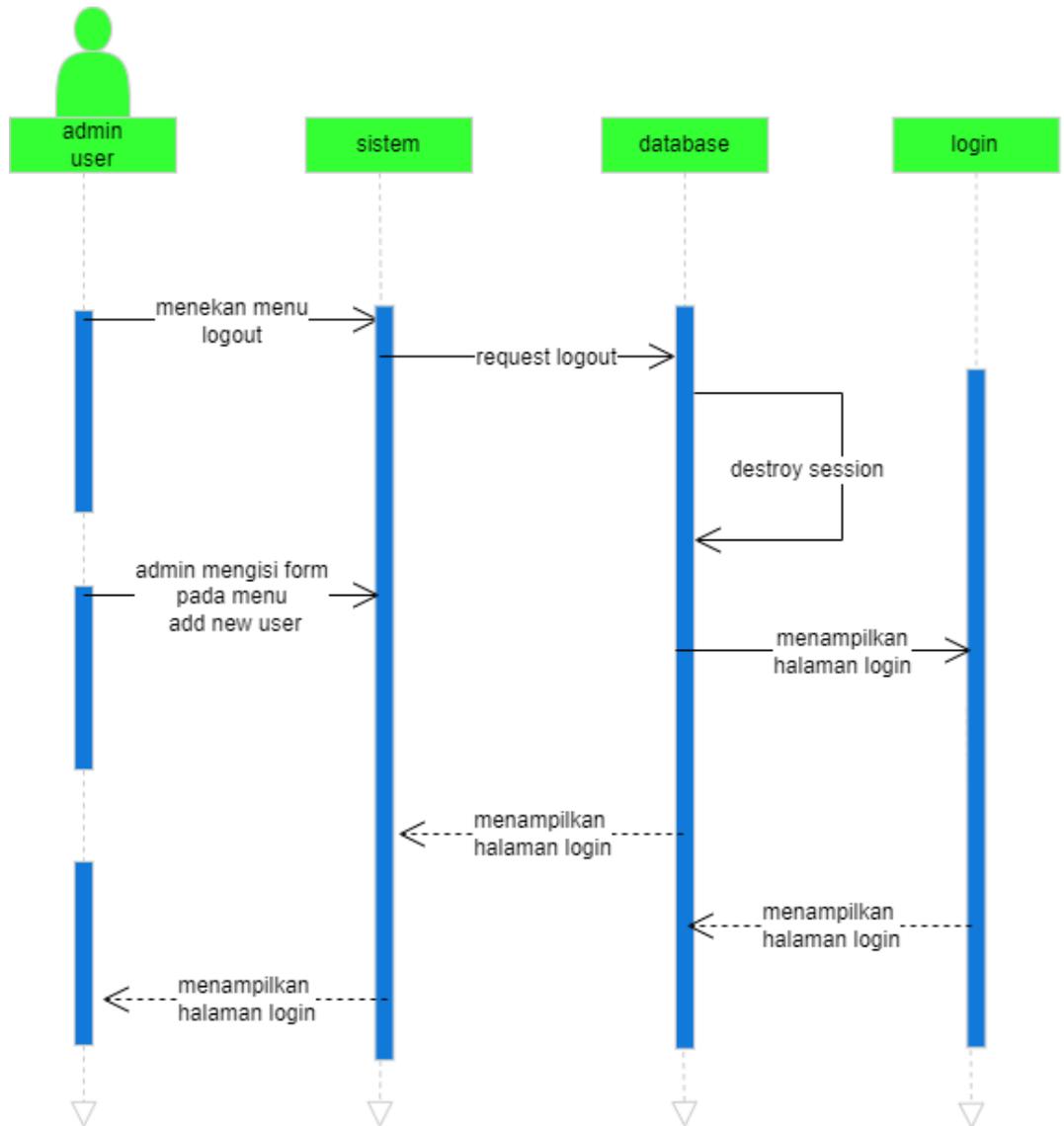
Pada gambar diagram 3.15 terdapat aktor (admin) dan tiga objek, yaitu sistem, halaman tambah user, database. Admin menekan menu Users, kemudian menekan menu *add new user* dan selanjutnya mengisi form yang tersedia untuk user baru.



Gambar 3.15 *Sequence Diagram* Tambah User

6) Logout

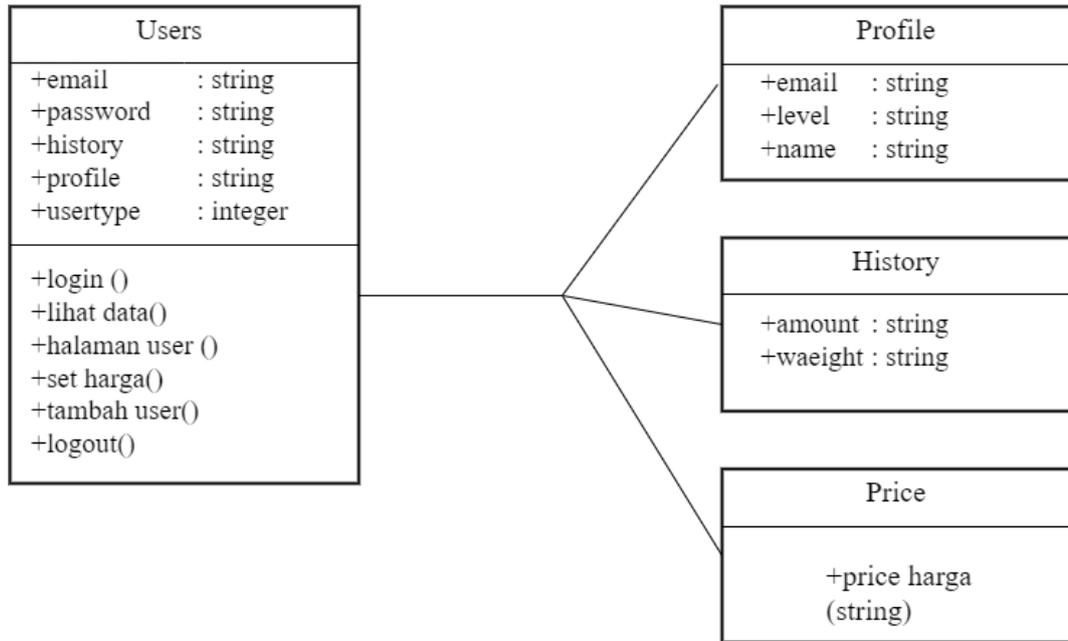
Pada gambar diagram 3.16 terdapat dua aktor (admin, user) dan tiga objek, yaitu sistem, database, dan login. Pertama admin atau user menekan tombol “logout”. Kemudian sistem akan mengirim permintaan untuk keluar ke database. Kemudian database akan memproses permintaan tersebut. Jika permintaan gagal di proses, maka akan menampilkan pesan logout gagal. Jika berhasil maka akan dikeluarkan dan kembali ke halaman login.



Gambar 3.16 *Sequence Diagram Logout*

c. Class Diagram

Berikut adalah merupakan *class diagram* dari *software* aplikasi yang akan di rancang :



Gambar 11 *Class Diagram* Aplikasi

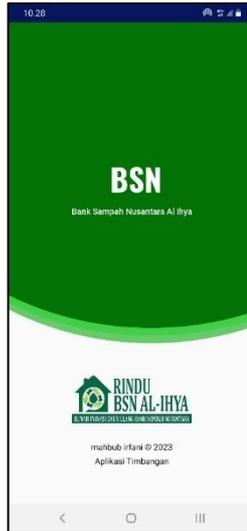
Class Diagram perancangan perangkat lunak aplikasi pada sistem bank sampah terdiri dari 4 class yaitu Users, Profile, History, Price. Pada gambar 19 juga dijelaskan hubungan antar class, seperti pada class Users yang tidak bisa berdiri sendiri tanpa adanya Profile, History dan Price. Artinya semua class tersebut saling berkaitan.

d. 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*

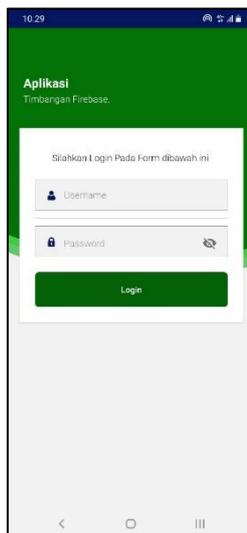
Splash Screen merupakan halaman pertama yang terbuka saat user atau admin membuka aplikasi. *Splash Screen* akan tampil selama 3 detik dan kemudian menuju halaman selanjutnya. *Splash Screen* pada aplikasi ini berfungsi untuk menampilkan logo aplikasi bank sampah, dan juga sebagai hiasan pada aplikasi agar terlihat lebih menarik.



Gambar 12 *Prototype Splash Screen*

b) Login

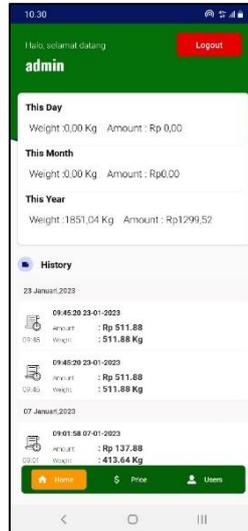
Pada tampilan login menunjukkan form yang harus diisi agar bisa mengakses aplikasi. Pada form terdapat 2 bagian yang harus diisi, pertama *username* dan yang ke dua adalah *password* yang telah didapat setelah mendaftar. Setelah mengisi form dengan benar langkah selanjutnya yaitu menekan tombol login yang berada di bawah form.



Gambar 13 *Prototype Login*

c) Dashboard admin

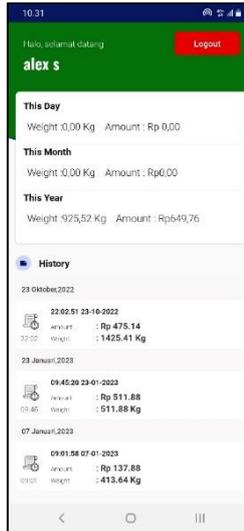
Pada *prototype* dashboard admin memiliki 3 menu yaitu price, users dan log out. Ketika admin ingin mengubah harga sesuai yang dibutuhkan maka admin menekan menu price. Ketika admin ingin menambahkan atau melihat data nasabah maka admin menekan menu users. Dan ketika admin ingin keluar dari aplikasi maka admin menekan menu logout.



Gambar 14 *Prototype Dashboard Admin*

d) Dashboard user

Berbeda dengan admin, setiap masing-masing user hanya dapat melihat data yang telah di simpan pada saat menabung di menu utama itu sendiri. Jika user ingin keluar dari aplikasi mak user harus menekan menu log out pada aplikasi tersebut.



Gambar 15 *Prototype* Dashboard User

e) Set Harga

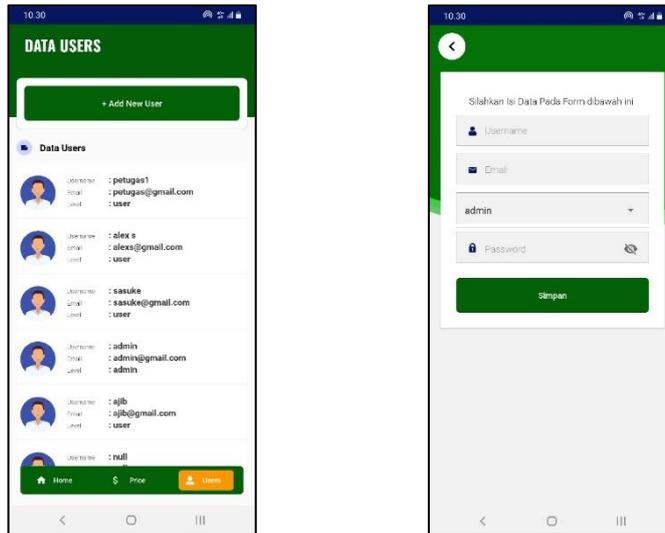
Jika admin ingin mengubah harga sampah pada timbangan, mak admin harus menekan menu price, kemudian menuliskan harga sesuai yang di dibutuhkan, lalu klik simpan.



Gambar 16 *Prototype* Set Harga

f) Tambah User

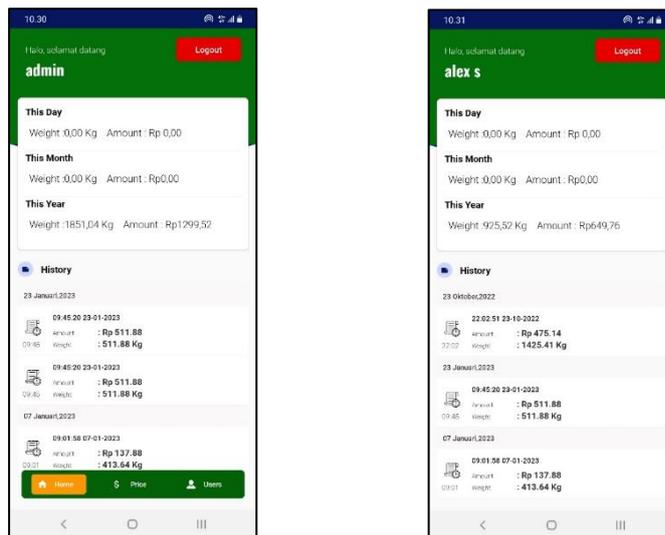
Apabila admin ingin menambahkan nasabah baru, maka admin harus menekan menu Users, kemudian klik menu add new user, lalu isikan form pada menu tersebut dan simpan.



Gambar 17 Prototype Tambah User

g) Logout

Pada menu logout tiap dashboard antara admin dan user keudanya memiliki fungsi yang sama yaitu mengeluarkan akun atau memutuskan sesi interaksi dengan aplikasi.



Gambar 18 Prototype Logout