

KONTROL PENGGUNAAN LISTRIK PASCABAYAR MENGGUNAKAN ANDROID

Muhammad Risal¹, Andy Lukman Affandy², Subair³, Arsianto⁴, Mursalim Sawawi⁴

^{1,2,3,4,5} STMIK Handayani

¹risal@handayani.ac.id, ²lukman@handayani.ac.id, ⁴mursalim.hsd@gmail.com

ABSTRAK

Listrik pascabayar adalah listrik yang pembayaran tagihannya pada akhir bulan sesuai dengan energi yang digunakan. Meteran listrik pascabayar masih analog yang menunjukkan besarnya daya yang digunakan. Sedangkan listrik Prabayar pembayarannya diawal sebelum digunakan dengan sistem pulsa sehingga pelanggan dapat mengendalikan pemakaian listrik sendiri karena pelanggan dapat membeli pulsa listrik sesuai kebutuhan. Pelanggan dapat mengontrol, melakukan monitoring atau pengecekan terhadap penggunaan listrik setiap hari. Adapun tujuan dari penelitian ini yaitu, 1) Merancang sistem yang dapat mengontrol dan memonitoring penggunaan daya listrik. 2) Mengimplementasikan sistem yang dapat mengontrol dan memonitoring penggunaan daya listrik. Sistem dibuat menggunakan mikrokontroler arduino nano dengan tambahan sensor daya PZEM-004T dan Nodemcu esp8266 untuk komunikasi dengan internet menggunakan aplikasi firebase dan relay sebagai output tegangan listrik. Sistem dapat dimonitoring dan dikontrol melalui smartphone android dimana pelanggan dapat langsung mengetahui total daya yang digunakan dan besar tarif yang akan dibayarkan. Pelanggan dapat menentukan tarif berdasarkan daya yang akan digunakan dan sensor daya akan membaca berapa daya yang sudah terpakai, jika daya melewati batas maksimum pemakaian yang sudah ditentukan, maka secara otomatis aliran listrik akan diputus ke arah terminal relay dan jika ingin menghidupkan kembali dilakukan reset daya pada aplikasi MIT APP Inventor pada handphone android.

Kata kunci: Android, Firebase, Sensor daya, relay, NodeMcu esp8266.

1. PENDAHULUAN

Listrik merupakan kebutuhan utama masyarakat di zaman ini, segala aktifitas kita tidak dapat terlepas dari kebutuhan akan listrik, bahkan ketika mati listrik kita jadi panik dan seakan-akan tidak dapat melakukan aktifitas. Di Indonesia listrik dikuasi oleh pemerintah melalui perusahaan BUMN PT. PLN (Persero) yang memiliki dua jenis layanan listrik yaitu pascabayar dan Prabayar.[1]

Listrik pascabayar adalah listrik yang pembayaran tagihannya pada akhir bulan sesuai dengan energi yang digunakan. Meteran listrik pascabayar masih menggunakan alat analog yang menunjukkan besarnya daya yang telah digunakan. Pembayaran listrik pascabayar dapat dilakukan melalui bank, loket pembayaran dengan biaya admin yang dibebankan sebagai biaya jasa pembayaran tagihan listrik. Sedangkan listrik Prabayar adalah listrik yang pembayarannya berada di awal, yaitu dengan sistem pulsa. Listrik Prabayar juga biasa disebut dengan listrik pintar. Dimana pelanggan dapat mengendalikan pemakaian listrik sendiri. Alat meteran listrik pun tidak lagi berbentuk analog, melainkan digital yang dapat digunakan untuk memasukan kode pengisian listrik. Perbedaan yang mencolok dari listrik pascabayar dengan listrik Prabayar adalah pada listrik Prabayar pelanggan dapat mengontrol, melakukan monitoring atau pengecekan terhadap penggunaan listrik setiap hari dikarenakan sudah menggunakan meteran digital. Sedangkan pada listrik pascabayar alat meteran listriknya masih analog.[2]

Tarif listrik per 1 April 2023 untuk sektor rumah tangga yakni daya 450 Volt Ampere (VA) bersubsidi Rp 415 per kWh, daya 900 VA bersubsidi Rp 605 per kWh, daya 900 VA RTM (Rumah Tangga Mampu) Rp 1.352 per kWh, daya 1.300 VA Rp 1.444,70 per kWh, daya 2.200 VA Rp 1.444,70 per kWh dan daya 3.500 ke atas : Rp 1.699,53 per kWh.[3]



Membahas soal lebih hemat mana antara listrik Prabayar dan Pascabayar, sebenarnya tergantung dari penggunaan listrik oleh pelanggan. Pelanggan listrik berbasis Prabayar dapat memantau penggunaan listrik secara langsung di meteran listrik. Sehingga dapat mengatur penggunaan listrik agar hemat. Sementara itu, pada listrik pascabayar, pelanggan PLN tidak dapat mengetahui secara langsung penggunaan listrik mereka. Rekening listrik hanya dikirim sekali di akhir bulan. Karena ketersediaan listrik yang tidak terbatas, jika tidak berhati-hati dalam menggunakan listrik, biaya kelistrikan mungkin akan membengkak. Penghematan listrik bagi pelanggan PLN berbasis pascabayar dapat dilakukan dengan menerapkan penggunaan listrik seminimal mungkin. Matikan peralatan yang menggunakan listrik jika tidak diperlukan. Sebuah penelitian di Universitas Muhammadiyah Surakarta menunjukkan bahwa biaya kelistrikan dengan daya 450 VA pada listrik pascabayar selisih 6,1 persen lebih tinggi dibandingkan sistem listrik Prabayar. Sedangkan pada kelistrikan dengan daya 900 VA, biaya listrik pascabayar juga lebih tinggi dibandingkan dengan yang Prabayar, yakni 2,7 persen.[4]

Penelitian sebelumnya yang telah dilakukan dengan judul “Monitoring Pemakaian Daya Listrik Secara Realtime Berbasis Internet Of Things”. Pada penelitian tersebut deteksi penggunaan daya listrik digunakan hanya untuk memonitoring penggunaan listrik yaitu pemakaian kWh daya listrik dan biaya pemakaian listrik. Perangkat yang digunakan Wemos D1 ESP8266, sensor arus INA219, dan aplikasi berbasis web.[5]

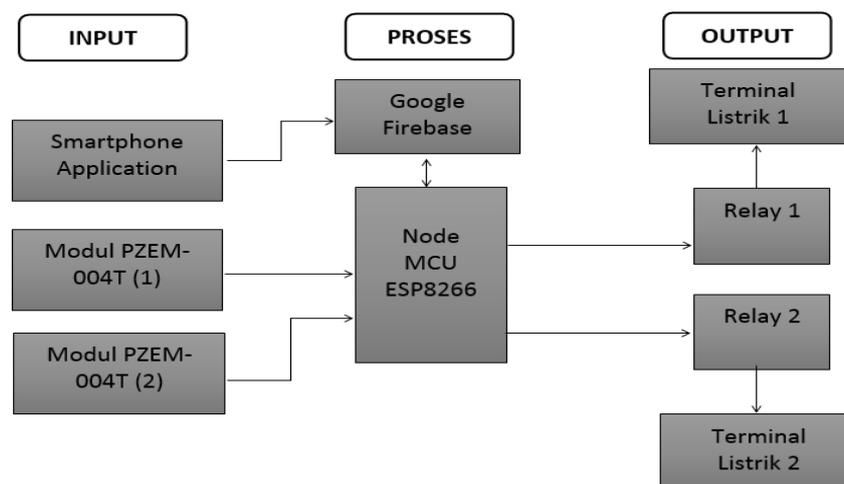
Berdasarkan hal tersebut bagi pelanggan listrik pascabayar agar dapat mengontrol dan memonitoring penggunaan daya listriknya diperlukan sebuah alat tambahan yang dapat menyajikan informasi penggunaan listrik lengkap dengan jumlah rupiah yang harus dibayarkan. Oleh itu Penulis berinisiatif membuat sebuah inovasi yang memanfaatkan mikrokontroler dimana dilakukan modifikasi terhadap terminal listrik yang dilengkapi komponen-komponen elektronika seperti Node MCU dan Relay untuk mengontrol penggunaan listrik. Selain itu, peralatan-peralatan elektronika yang terhubung pada terminal tersebut dikontrol dengan aplikasi yang terdapat pada smartphone dimana kita dapat menentukan besar kWh yang akan digunakan dan memutuskan listrik ketika melewati kWh yang ditentukan.

2. METODE PENELITIAN

2.1 Pemodelan dan Perancangan Sistem

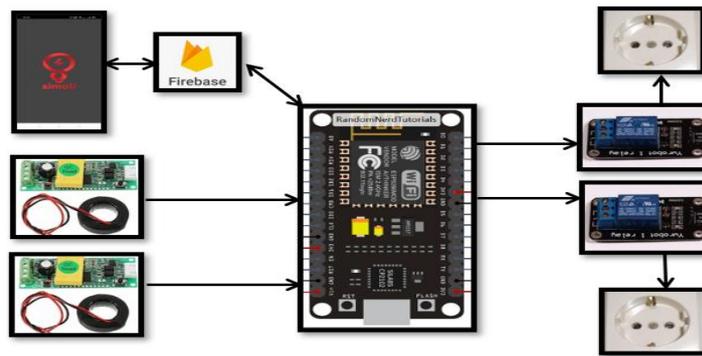
A. Diagram Blok

Rancang bangun produk yang hendak dikembangkan berupa modifikasi terminal listrik yang dilengkapi komponen-komponen elektronika seperti *Node MCU* dan *relay* untuk mengontrol penggunaan listrik pada peralatan-peralatan elektronika rumah tangga yang dikontrol melalui aplikasi *smartphone*. Berikut adalah desain perancangan perangkat keras produk tersebut:



Gambar 2.1. Blok Diagram Perangkat Keras

Gambar di atas merupakan gambar blok diagram dari perangkat keras yang akan dibuat, terdapat tiga bagian utama, yaitu input, proses, dan output. Input berupa sensor atau pembaca data, *nodemcu* (microcontroller arduino nano) sebagai pemroses, dan relay sebagai output kontrol yang dikendalikan. Masing-masing memiliki fungsi tersendiri hingga membentuk sebuah sistem *IoT* (*internet of things*) yang kompleks. Untuk lebih jelasnya berikut disajikan diagram blok perangkat keras :



Gambar 2.2. Hubungan Perangkat Keras

Rancangan perangkat keras terdiri dari beberapa kompone penyusun yaitu :

a). *Arduino Nano* [6]

Sebagai pemroses data sensor yang juga akan mengatur aktif dan matinya *relay* berdasarkan nilai *ADC* sensor yang telah di konversi kedalam bentuk daya atau persen.

b). *Nodemcu esp8266* [7]

Nodemcu akan berfungsi sebagai media antara data sensor dan pengiriman data ke database *cloud firebase cloud messaging* dan juga mengirimkan *push notification*.

c). Sensor Daya *PZEM-004T*

Modul *PZEM-004T* adalah sebuah sistem yang dirancang dan dimanfaatkan untuk membaca jumlah pemakaian energi listrik yang meliputi tegangan, arus, daya aktif dan akumulasi energi [8]. Untuk mengaksesnya, modul ini dihubungkan dengan *mikrokontroler NodeMcu* dan diprogram dengan menggunakan *library* yang telah disediakan oleh *github* ataupun pembuat dari modul sensor itu sendiri. Sensor ini mengeluarkan output dengan komunikasi serial. Jika kita ingin menghubungkan dengan *microcontroller* maka komunikasi yang di gunakan adalah komunikasi serial.

d). *Relay*

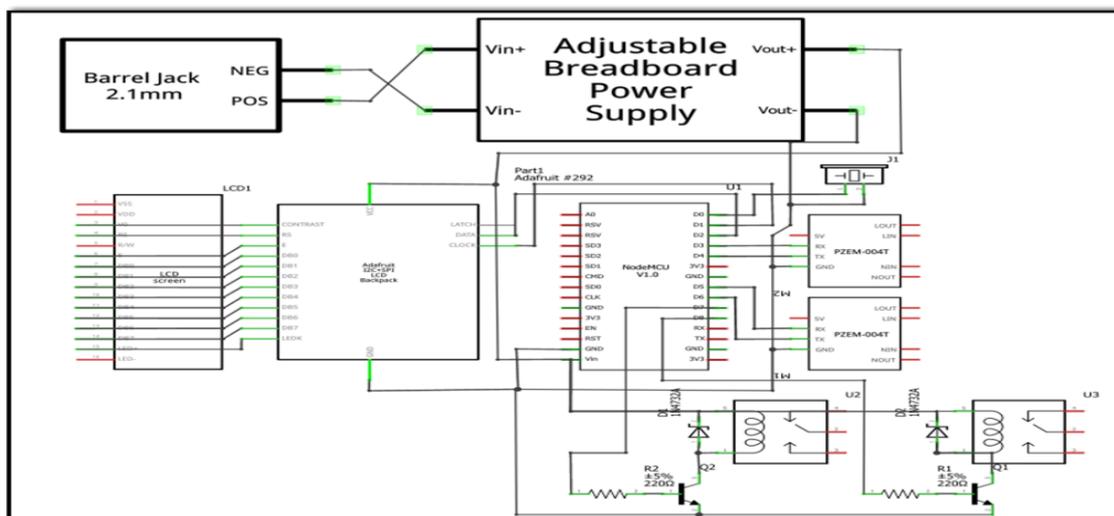
Relay adalah saklar (*switch*) yang dioperasikan secara listrik dan merupakan komponen elektromekanikal yang terdiri dari dua bagian utama yakni elektromagnet (*coil*) dan mekanikal (seperangkat kontak saklar/*switch*). *Relay* menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.[9]

e). Terminal listrik

Untuk menghubungkan ke perangkat yang akan di kontrol penggunaan daya listriknya.

B. Skematik Rangkaian

Rancangan perangkat keras seperti terlihat pada gambar 2.1 dan 2.2. menunjukkan hubungan antar perangkat keras akan tetapi tentu tidak sesederhana itu untuk menghubungkan antara pin pada komponen, untuk itu dibuat skematik rangkaian yang menjelaskan lebih detail hubungan antar pin pada komponen. Skematik ini dibuat dengan menggunakan aplikasi *fritzing* yang disajikan pada gambar berikut:



Gambar 2.3. Skematik Rangkaian

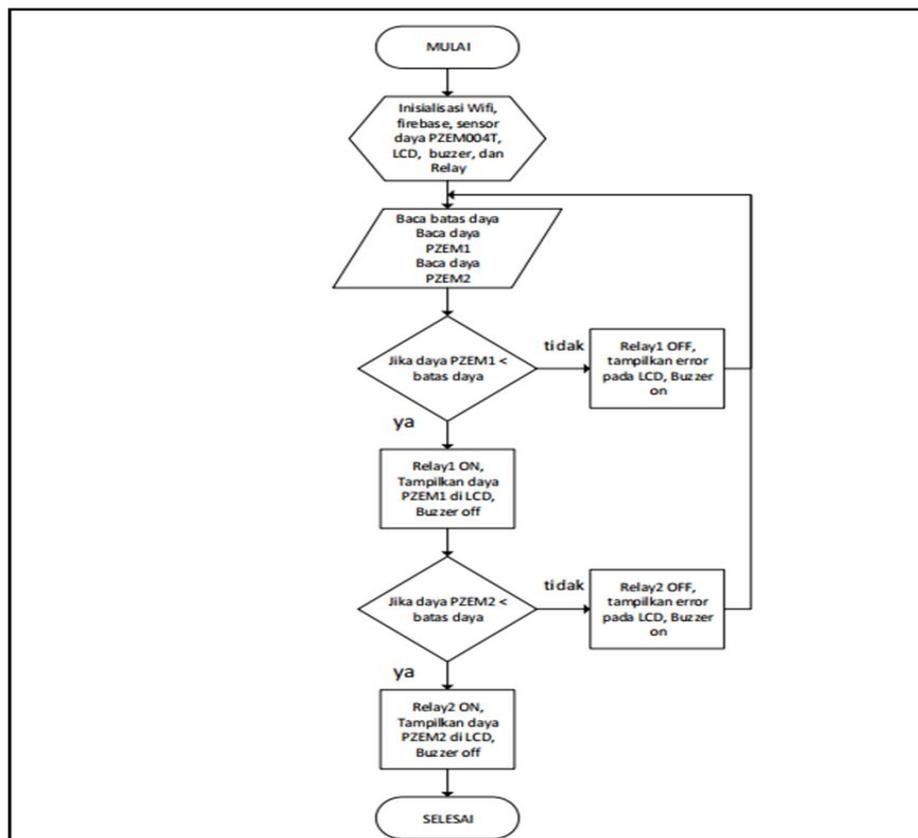
Hubungan antar pin pada setiap komponen perangkat keras elektronika yang membangun sistem ini dapat dilihat pada tabel berikut ini:

Tabel 2.1 Hubungan Antar Pin Pada Komponen Dengan Pin *Ardiono Nano*

No.	Nama Komponen	Pin Pada Komponen	Pin Pada Nodemcu
1	Sensor PZEM004T 1	VCC	Vin
		Gnd	Gnd
		RX	D3
		TX	D4
2	Sensor PZEM004T 2	VCC	Vin
		Gnd	Gnd
		RX	D5
		TX	D6
3	Modul Relay 1	VCC	Vin
		Gnd	Gnd
		Data	D7
4	Modul Relay 2	VCC	Vin
		Gnd	Gnd
		Data	D8
5	LCD 20X4	VCC	Vin
		Gnd	Gnd
		SDA	D2
		SCL	D1
6	Buzzer	Data	D0

C. Diagram Flowchart

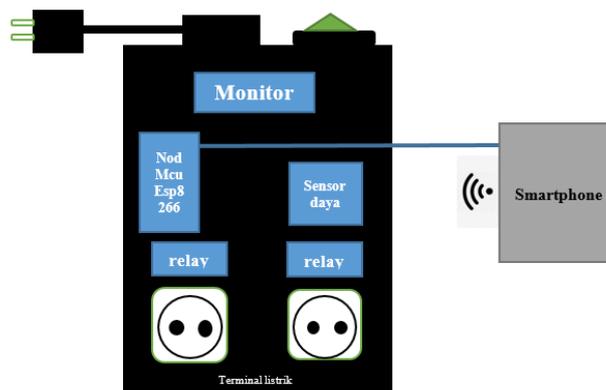
Untuk mengetahui seperti apa proses atau alur kerja yang terjadi pada sistem kontrol penggunaan listrik pascabayar menggunakan *android* dapat dilihat pada diagram flow sebagai berikut:



Gambar 2.4. Flowchart sistem

Untuk mengaktifkan ataupun mematikan terminal listrik maka harus membuka aplikasi *android* terlebih dahulu. Setelah itu pengguna memilih terminal mana yang akan dikendalikan. Saat ingin mematikan terminal maka pengguna harus menekan tombol “Off” pada *relay* yang dipilih, sebaliknya jika pengguna ingin mengaktifkan terminal maka harus menekan tombol “On” pada *relay* yang dipilih. Setelah tombol ditekan maka data akan dikirim ke *web server*. *Web server* menerima data dan mengupdate data *server*. Lalu program pada Modul *ESP8266* *melooping* data yang diterima dari *webserver*, setelah mendapat data maka melakukan proses tindakan melalui *relay* yaitu mengaktifkan atau mematikan terminal sebagai keluaran/*output* produk yang dikendalikan sesuai perintah dari aplikasi *android* tersebut. Dari terminal tersebut, pengguna dapat menghubungkannya ke peralatan-peralatan elektronik yang ingin dikontrol.

D. Arsitektur Sistem

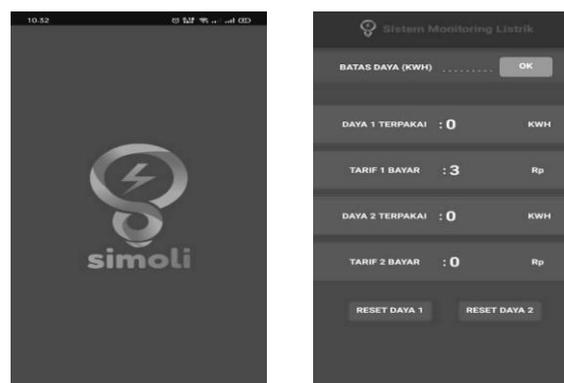


Gambar 2.5. Desain Arsitektur Sistem Manajemen daya listrik

Arsitektur system dari alat yang akan dibuat memiliki beberapa komponen yang saling berkaitan serta saling mendukung dan membentuk sebuah rangkaian manajemen penggunaan daya listrik pada rumah. Adapun komponen-komponen yang digunakan yaitu *nodemcu esp8266*, sensor daya, relay, terminal listrik, monitor *lcd*, dan *smartphone*. Sistem ini menerima input dari sensor daya yang akan membaca data daya penggunaan listrik sehingga dapat disesuaikan dengan tarif yang akan dibayar berdasarkan jumlah maksimum daya yang diberikan. Untuk proses alat ini menggunakan Arduino untuk membaca data sensor dan dibuatkan sebuah logika yang akan mengotomatiskan terminal listrik pada rangkaian tersebut, disini relay akan berfungsi untuk memutuskan aliran listrik yang dikirim ke setiap terminal kemudian data dikirim via *serial communication* ke *Nodemcu esp8266* sebagai media ke database *firebase* sehingga dapat diakses oleh aplikasi *android*.

2.2 Perancangan Perangkat Lunak (Software)

Sebelum membuat aplikasi sistem kontrol penggunaan listrik pascabayar terlebih dahulu dirancang model tampilan *interface* pada *smartphone* yang diharapkan mudah digunakan *user*, tampilan memuat batas daya kwh yang dibolehkan, daya yang digunakan serta nilai rupiah yang harus dibayarkan. Berikut rancangan tampilan *interface* pada *smartphone*.



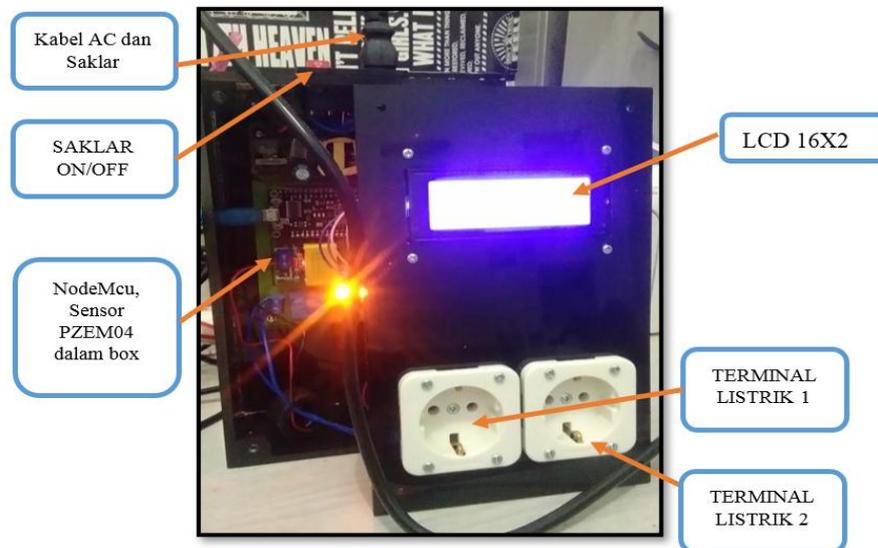
Gambar 2.6. Rancangan Interface Pada Smartphone

3. HASIL DAN PEMBAHASAN

3.1. Hasil Sistem Kontrol Penggunaan Listrik Pascabayar Menggunakan Android

Hasil penelitian yang telah dilakukan adalah sebuah perangkat sistem *kontrol* dan *monitoring* berbasis *microcontroller* dan *smartphone application*. Adapun fungsi dari alat ini terdiri dari 1) Menyalakan dan mematikan terminal listrik secara otomatis dengan sensor daya, 2) membatasi penggunaan daya listrik, 3) memonitoring penggunaan daya listrik, dan 4) Memonitoring pembayaran tagihan listrik pada setiap terminal.

a. Hardware.



Gambar 3.1. Penampakan Produk Secara Keseluruhan

Alat ini terdiri dari komponen input, proses, dan output, perangkat terdiri dari sensor arus *pzem04t*, saklar *on/off*, *microkontroller arduino nano nodemcu*, lampu dan terminal listrik. Sensor *pzem04t* akan membaca daya yang terpakai pada terminal listrik, kemudian data daya tersebut diolah oleh *microkontroller* dan dikirimkan ke *firebase* untuk selanjutnya diterima oleh aplikasi *android*. Output dari sensor daya ini juga akan dikonversi menjadi tarif dalam satuan rupiah dengan menggunakan rumus perkalian tarif dan kwh agar pemilik alat dapat memonitoring penggunaan listrik pada terminal yang dipasang alat tersebut.

Adaptor 5V DC berfungsi untuk mensupply tegangan ke seluruh komponen yang digunakan, baik sensor, rangkaian relay, maupun *microkontroller nodemcu*. Terminal listrik, digunakan sebagai *output* yang akan digunakan untuk mengalirkan listrik ke peralatan-peralatan listrik yang terpasang pada terminal listrik.

b. Pemrograman Arduino IDE

Microcontroller nodemcu tidak dapat berfungsi begitu saja sesuai dengan apa yang diinginkan. Oleh sebab itu kita harus mengisi sebuah bahasa pemrograman ke dalam *chip microcontroller* dengan *algoritma* yang telah dirancang sebelumnya. Bahasa pemrograman ini adalah bahasa C++ yang dibuat dengan menggunakan aplikasi *Arduino IDE*. Untuk mengakses komponen yang digunakan, pengembang aplikasi *arduino ide* tersebut telah menyediakan *library* untuk memudahkan para *programmer* dalam penyusunan *algoritma*-nya. *Programmer* cukup melakukan pengaturan-pengaturan dasar pada listing seperti settingan pin yang digunakan, karakteristik komponen dan lainnya. Berikut adalah tampilan program yang telah dibuat pada aplikasi *arduino ide* :

```
1#include <ESP8266WiFi.h>
2#include <FirebaseESP8266.h>
3#include <PZEM004Tv30.h>
4#include <Wire.h>
5#include <LiquidCrystal_I2C.h>
6#define FIREBASE_HOST "https://monitoring-electric-9bace-default-rtdb.firebaseio.com/"
7#define FIREBASE_AUTH "J5KQe2no0Fhd8nTioeRgdkbp53WgpBfikGEGWjW"
8#define WIFI_SSID "connected"
9#define WIFI_PASSWORD "apasajalah"
```

Gambar 3.2. Scrip Koneksi Wifi

Pada baris 1 hingga 8, merupakan program *library* yang digunakan, *library* tersebut adalah *library* untuk mengakses koneksi *wifi*, *firebase*, *LCD16X2* dan sensor *pzem04t*. Pada program tersebut adalah pengaturan untuk memasukkan alamat *url* dan token dari *firebase* yang telah dibuat sehingga alat dapat mengakses server *firebase* tersebut dan menyimpan datanya disana. Selanjutnya terdapat program untuk mengatur nama *wifi* dan *password* yang nantinya digunakan untuk menghubungkan *microcontroller* dengan *internet*.

```
11 #define buzzer D0
12 #define beban1 D8
13 #define beban2 D7
14
15 LiquidCrystal_I2C lcd(0x27,20,4);
16 PZEM004Tv30 pzem1(D3, D4);
17 PZEM004Tv30 pzem2(D6, D5);
18 FirebaseData firebaseData;
19 float Energy1,Energy2,Harga1,Harga2;
20 String resetDaya1,resetDaya2,batasDaya1,batasDaya2;
21 String data[10]; //array untuk menyimpan
22 int i,dayaMaks;
```

Gambar 3.3. pengaturan pin-pin pada NodeMCU

Program pada baris 11 hingga 13 merupakan program untuk melakukan pengaturan pada pin-pin *nodemcu* yang digunakan, seperti pin yang terhubung ke *buzzer*, saklar dan beban terminal listrik. Selanjutnya pada baris 15 hingga 22 adalah program untuk membuat sebuah *variabel* yang akan digunakan untuk menyimpan data berupa nilai penggunaan daya sensor *pzem04t*, pengaturan batas penggunaan daya listrik, dan tarif yang harus dibayar pada masing-masing terminal listrik.

```
24 void setup(){
25   Serial.begin(9600);
26   lcd.init();
27   lcd.init();
28   lcd.backlight();
29   pinMode(beban1,OUTPUT);]
30   pinMode(beban2,OUTPUT);
31   pinMode(buzzer,OUTPUT);
32
33   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
34   while (WiFi.status() != WL_CONNECTED){Serial.print(".");delay(300);}
35   Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
36   Firebase.reconnectWiFi(true);
37   digitalWrite (buzzer,0);
```

Gambar 3.4. Program *Looping* Pencarian *Wifi*

Pada baris 24 hingga 37 adalah baris program yang berfungsi sebagai *looping* sementara, artinya program hanya akan dieksekusi sekali saja lalu beralih keprogram *full looping*. Selanjutnya adalah penentuan mode pada pin yang digunakan, dimana pin tersebut semuanya menggunakan mode output. Baris 33 hingga 37 adalah program untuk memulai fungsi *wifi* agar melakukan pencarian nama *wifi* dan *password* yang telah diatur sebelumnya. Selain itu juga dilakukan koneksi dengan *firebase* sehingga data akan dapat dikirim ke sana.

```
45 void loop(){
46
47   Energy1 = pzem1.energy();
48   Harga1 = Energy1*1500.00;
49   Energy2 = pzem2.energy();
50   Harga2 = Energy2*1500.00;
51   if (isnan(Energy1)|| isnan(Energy2)){lcd.setCursor(0,0); lcd.print(" DAYA TERBACA ERROR ");delay(3000);}
52   lcd.setCursor(0,0); lcd.print(" DAYA LISTRIK AUTO ");
53   lcd.setCursor(0,1); lcd.print("F1="); lcd.setCursor(3,1); lcd.print(Energy1,3);lcd.setCursor(8,1); lcd.print ("KWh");
54   lcd.setCursor(12,1); lcd.print("Rp="); lcd.setCursor(15,1); lcd.print(Harga1,2);
55   lcd.setCursor(0,2); lcd.print("F2="); lcd.setCursor(3,2); lcd.print(Energy2,3);lcd.setCursor(8,2); lcd.print ("KWh");
56   lcd.setCursor(12,2); lcd.print("Rp="); lcd.setCursor(15,2); lcd.print(Harga2,2);
57   lcd.setCursor(0,3); lcd.print("BATAS DAYA= ");lcd.setCursor(17,3); lcd.print("KWh");
58   if (Firebase.ready()){
59     Firebase.setFloat (firebaseData,"/Monitoring Listrik/Daya1", Energy1);
60     Firebase.setFloat (firebaseData,"/Monitoring Listrik/Daya2", Energy2);
61     Firebase.setFloat (firebaseData,"/Monitoring Listrik/Harga1", Harga1);
62     Firebase.setFloat (firebaseData,"/Monitoring Listrik/Harga2", Harga2);
```

Gambar 3.5. Tarif Pembayaran

Pada program diatas baris 45 hingga 62 adalah program pertama dalam *full looping*, artinya akan dieksekusi secara terus menerus selama alat dalam kondisi on. Pada baris 47 adalah perintah untuk mendapatkan data daya pada sensor *pzem04t* pertama. Baris 48 adalah rumus untuk memperoleh tarif pembayaran dari sensor pertama. Selanjutnya adalah baris 49, sama seperti baris 47 yaitu program untuk mendapatkan data daya dari sensor *pzem04t*.

Baris 51 hingga 57 adalah program untuk menampilkan penggunaan daya listrik beban pertama dan kedua. Selain itu juga ditampilkan tarif pembayaran dan batas penggunaan daya. Pada baris 59 hingga 62 adalah perintah untuk mengisi data pada *server googe firebase*.

```
63 Firebase.getString(firebaseData, "/Monitoring Listrik/ResetDaya1"); resetDaya1=firebaseData.to<const char *>();
64 Firebase.getString(firebaseData, "/Monitoring Listrik/ResetDaya2"); resetDaya2=firebaseData.to<const char *>();
65 Firebase.getString(firebaseData, "/Monitoring Listrik/BatasDaya1"); batasDaya1=firebaseData.to<const char *>();
66 }
67 if (resetDaya1=="1") { pzem1.resetEnergy();Firebase.setString(firebaseData, "/Monitoring Listrik/ResetDaya1", "0");}
68 if (resetDaya2=="1") { pzem2.resetEnergy();Firebase.setString(firebaseData, "/Monitoring Listrik/ResetDaya2", "0");}
69
70 int j=0; //indeks array untuk dt yang dimainkan
71 data[j]="";
72 //array penampung data dimana indeks nya berdasarkan nilai j
73 for (i=0;i<batasDaya1.length();i++) { //mengecek satu persatu data yang diterima, pengecekan dimulai
74     if (batasDaya1[i]=="") {j++; data[j]="";} //jika data berupa karakter " maka indeks array akan berpind
75     else {data[j]=data[j]+batasDaya1[i];} //jika bukan karakter " maka dia akan dimasukkan ke penamp
76 }
77
78
79 dayaMaks = data[1].toInt(); //konversi data array ke int krn data disimpan dalam satu indeks v
80 lcd.setCursor(12,3);lcd.print(dayaMaks);
```

Gambar 3.6. Menerima data dari firebase

Baris program 63-80 adalah bagian program yang berfungsi untuk menerima data dari *firebase*. Perintah reset daya dan batas penggunaan daya semuanya berasal aplikasi *smartphone* yang disimpan pada *server google firebase*. Selain itu, *nodemcu* juga harus melakukan *parsing* data agar lebih mudah membaca data dari *firebase*, proses *parsing* data diawali pada baris 70 hingga 76.

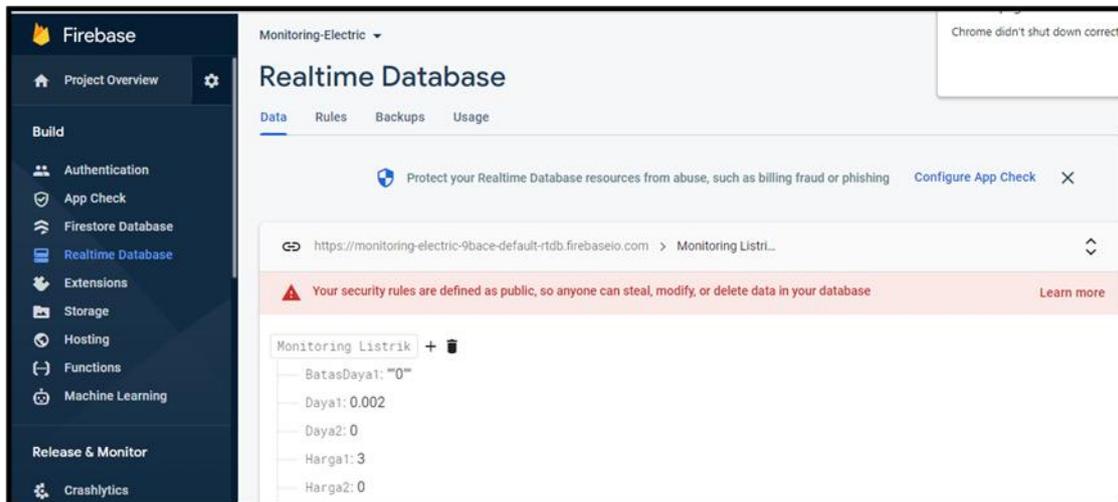
```
79 dayaMaks = data[1].toInt();
80 lcd.setCursor(12,3);lcd.print(dayaMaks);
81 //Serial.println(dayaMaks);
82 if (Energy1<=dayaMaks) {digitalWrite(beban1,1);}
83 if (Energy2<=dayaMaks) {digitalWrite(beban2,1);}
84 if (Energy1>dayaMaks) {digitalWrite(beban1,0);}
85 if (Energy2>dayaMaks) {digitalWrite(beban2,0);}
86 }
```

Gambar 3.7. Pembatasan Daya

Ini adalah inti dari program secara keseluruhan. Program di atas akan melakukan pembatasan penggunaan daya listrik dan diberikan output berupa keputusan dan penyambungan pada *relay* yang ada. Pada *listing* program baris 78 hingga 86 adalah termasuk dalam *looping* tertutup yang terus menerus dieksekusi oleh *microcontroller*. Selama alat dalam kondisi aktif atau terhubung dengan sumber listrik, maka program akan terus dijalankan. Program di atas berisi tentang *algoritma* dari prinsip kerja alat yang telah dirancang. Program selanjutnya adalah akses sensor *pzem04t* hingga dapat mengukur tegangan dan arus listrik pada alat sehingga dapat pula dihitung daya listrik yang digunakan. Nilai tersebut berupa nilai energi yang disimpan pada variabel energi kemudian dikonversi ke satuan rupiah. Data ini selanjutnya dikirim ke database *firebase* menggunakan perintah *firebase setfloat*. Selanjutnya aturan pembatasan untuk penggunaan daya listrik, jika daya yang digunakan masih kurang dari watt yang ditentukan maka alat tetap berfungsi dengan normal, akan tetapi jika penggunaan daya telah melebihi watt yang ditentuka maka aliran listrik pada alat otomatis akan diputuskan.

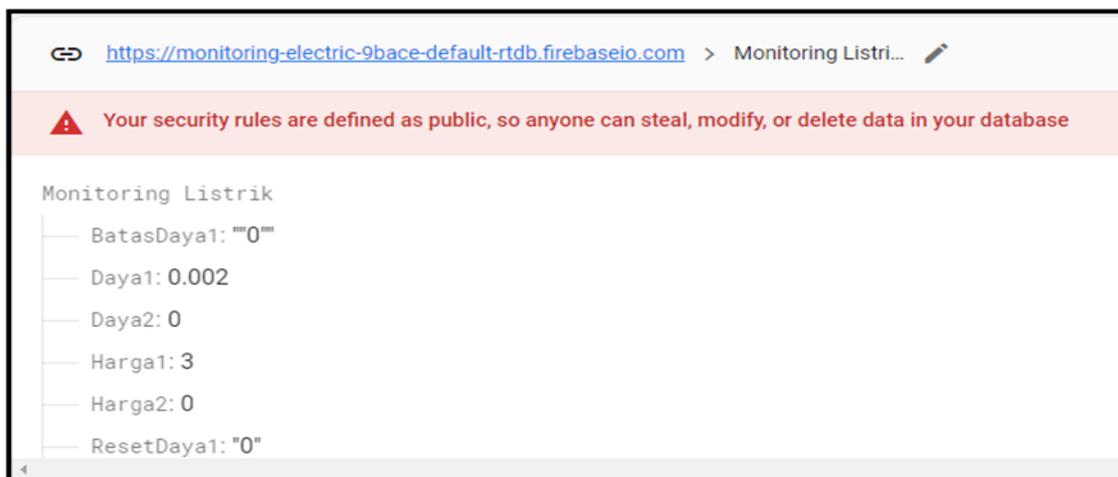
c. *Firebase*

Penghubung antara alat dengan aplikasi *smartphone* adalah *server firebase*, *firebase* dibuat pada halaman tertentu yang telah disediakan oleh *google*. Pada *firebase* terdapat database yang bekerja secara *realtime* sehingga kita dapat mengaksesnya dimanapun dan kapanpun. Berikut adalah tampilan *firebase* yang telah dibuat :



Gambar 3.8. Tampilan *Firebase*

Gambar di atas adalah tampilan secara keseluruhan dari *firebase*, adapun tampilan untuk database tempat menampung data dari alat maupun aplikasi adalah sebagai berikut :

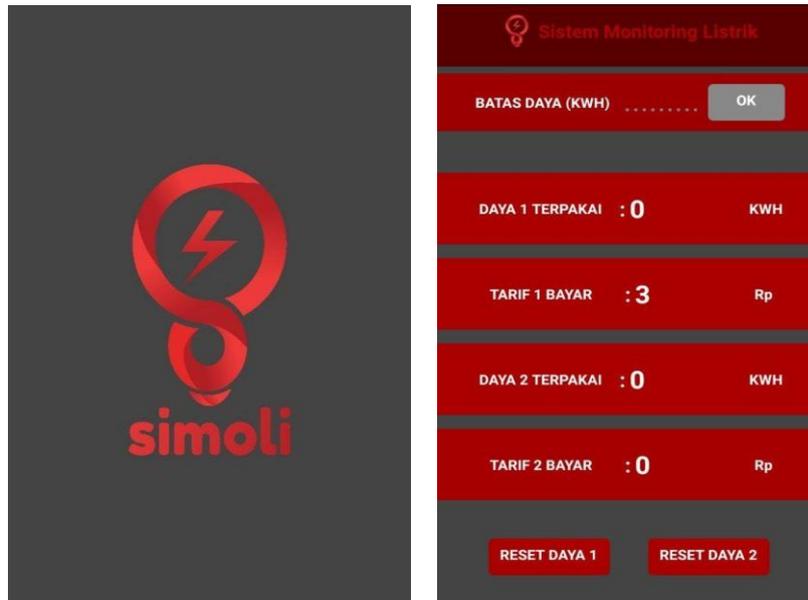


Gambar 3.9. Realtime Database Pada *Firebase*.

Pada gambar di atas terlihat beberapa data yang berasal dari *microcontroller* maupun dari aplikasi *smartphone*, data tersebut adalah *daya1*, *daya2*, *tarif1*, *tarif2*, *resetenergy*, dan *batas daya*. Masing-masing data memiliki pengolahan tersendiri sehingga menghasilkan sebuah sistem monitoring penggunaan daya listrik berbasis aplikasi *smartphone*. Untuk mengakses *firebase* tersebut, yang dibutuhkan hanya ada dua, yaitu alamat *url* yang dapat dilihat pada bagian database *setting*, lalu klik *service account*, dan selanjutnya akan muncul alamat *url* dari database *firebase* yang digunakan. Selanjutnya adalah kode *API* atau token dari *realtime* database, kode tersebut dapat dilihat pada *firebase setting*, kemudian masuk pada *service account*, selanjutnya pilih database *secret*, maka akan ditampilkan kode *API* tersebut. Alamat *URL* dan kode *API* ini akan dimasukkan ke dalam program *arduino ide* dan begitu pula pada desain aplikasi pada *mit app inventor*.

d. Aplikasi *smartphone*.

Pada pembuatan aplikasi ini terdiri dari dua bagian, yaitu bagian *desain* dan *back end* atau blok program. Desain merupakan tampilan aplikasi (*interface*) sedangkan blok program adalah *algoritma* yang membuat aplikasi dapat berfungsi seperti yang diinginkan.



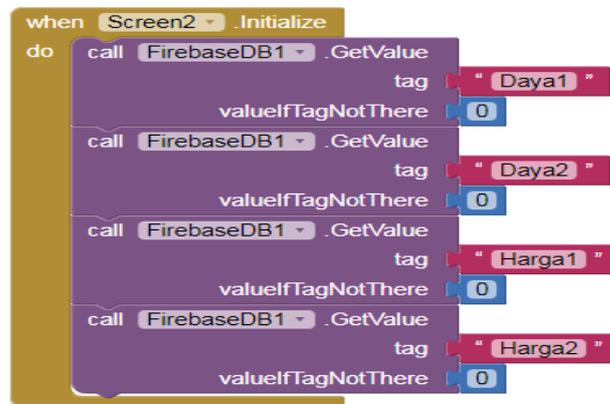
Gambar 3.10. Desain Tampilan Aplikasi *Smartphone* Monitoring Listrik

Tampilan dimulai dengan *splash screen*, yang menampilkan logo dari aplikasi untuk memperkenalkan logo dari aplikasi sehingga user terbiasa mulai dari warna dan *brand* aplikasi layaknya aplikasi-aplikasi lainnya. Selanjutnya tampilan utama dari aplikasi yang merupakan layanan utama dari aplikasi *mitt app inventor*. Yang pertama adalah logo aplikasi kembali ditampilkan di *header* aplikasi beserta kalimat-kalimat *branding* yang akan memberikan manfaat atau kegunaan dari aplikasi ini. Lalu pada tampilan tengah aplikasi terdapat beberapa bagian di antaranya untuk menginput batas daya yang akan di berikan, selanjutnya data di kirim ke *nodemcu* dan akan diproses. Tampilan daya 1 terpakai menunjukkan bahwa terminal 1 sedangkan tampilan daya 2 terpakai menunjukkan pada terminal 2 dan untuk mempertegas tarif yang akan di bayar, terdapat kolom tarif bayar 1 dan tarif bayar 2 menunjukkan jumlah tarif yang akan dibayar berdasarkan pemakaian daya. Pengguna juga dapat mereset daya jika daya sudah habis atau daya sudah mau habis, terdapat reset daya 1 dan reset daya 2.

Tabel 3.1 Spesifikasi Aplikasi *MITT APP Inventor* pada *Android*

No	Function	Spesifikasi
1	Image (logo)	Icon, weight 50%
2	Label (batas daya)	Weight 10%, length 100%,font default, red background,white font
3	Button (ok)	Style butto, Weight 10%, font default, red background,white font
4	Label (daya terpakai 1)	Weight 10%, length 100%, font default, red background,white font
5	Label (tarif 1)	Weight 10%, length 100%, font default, red background,white font
6	Label (daya terpakai 2)	Weight 10%, length 100%, font default, red background,white font
7	Label (tarif 2)	Weight 10%, length 100%, font default, red background,white font
8	Button (Reset1)	Style butto, Weight 10%, font default, red background,white font
9	Button (Reset2)	Style butto, Weight 10%, font default, red background,white font

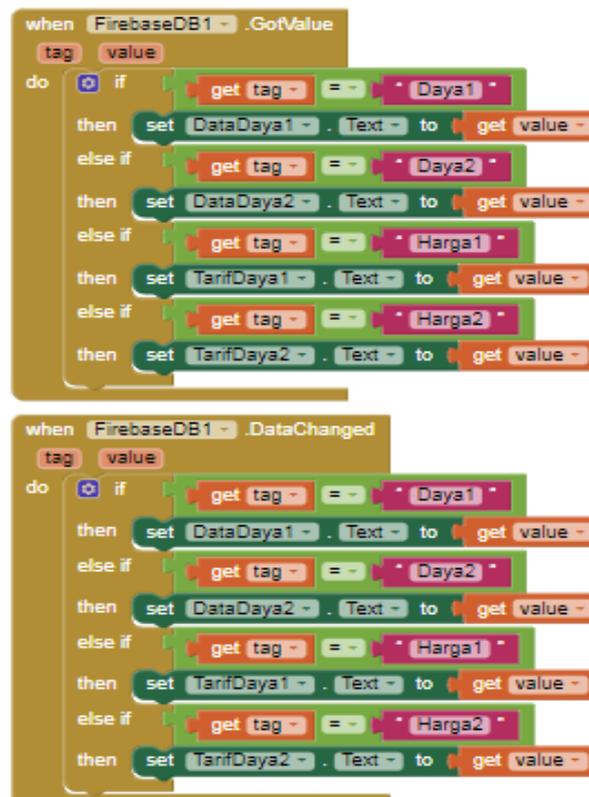
Pada bagian desain, tidak terlalu sulit karena hanya membuat desain dari tampilan aplikasi yang telah dirancang sebelumnya. Hanya membutuhkan konsep desain yang baik agar mudah dipahami dalam penggunaannya. Akan tetapi yang membuat aplikasi ini berfungsi sebenarnya ada pada bagian blok programnya, setiap fitur yang ada pada tampilan aplikasi tidak dapat berfungsi tanpa adanya program ini, berikut adalah blok program dari aplikasinya :



```
when Screen2 .Initialize
do
  call FirebaseDB1 .GetValue
  tag "Daya1"
  valueIfTagNotThere 0
  call FirebaseDB1 .GetValue
  tag "Daya2"
  valueIfTagNotThere 0
  call FirebaseDB1 .GetValue
  tag "Harga1"
  valueIfTagNotThere 0
  call FirebaseDB1 .GetValue
  tag "Harga2"
  valueIfTagNotThere 0
```

Gambar 3.11. Blok 1 Program Aplikasi

Blok 1 adalah program yang berfungsi untuk menerima data dari *firebase* khususnya pada database dengan nama *daya1*, *daya2*, *harga1* dan *harga2*. Semua data dari database tersebut secara *realtime* akan masuk pada aplikasi. Angka 0 artinya jika data tidak tersedia atau tidak terhubung dengan *firebase* maka otomatis aplikasi akan memunculkan angka 0. *Firestore* 1 hingga 4 merupakan jalan masuknya setiap data. Oleh sebab itu tidak boleh tertukar, karena akan menyebabkan data juga tertukar.

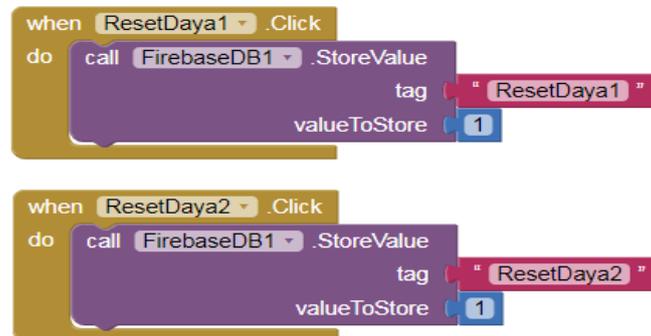


```
when FirebaseDB1 .GetValue
tag value
do
  if get tag == "Daya1"
  then set DataDaya1 .Text to get value
  else if get tag == "Daya2"
  then set DataDaya2 .Text to get value
  else if get tag == "Harga1"
  then set TarifDaya1 .Text to get value
  else if get tag == "Harga2"
  then set TarifDaya2 .Text to get value

when FirebaseDB1 .DataChanged
tag value
do
  if get tag == "Daya1"
  then set DataDaya1 .Text to get value
  else if get tag == "Daya2"
  then set DataDaya2 .Text to get value
  else if get tag == "Harga1"
  then set TarifDaya1 .Text to get value
  else if get tag == "Harga2"
  then set TarifDaya2 .Text to get value
```

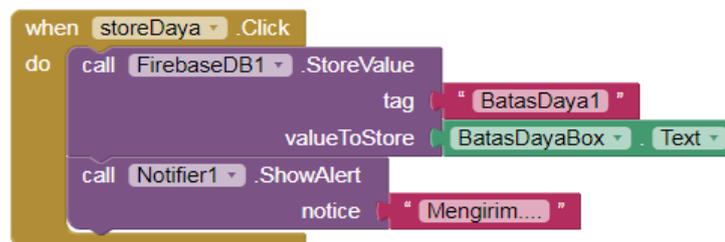
Gambar 3.12. Blok 2 Program Aplikasi

Pada blok 2 ini, merupakan fungsi untuk menyimpan data dari daya dan harga pada variabel dengan nama data daya dan tarif daya dan data Rp, dimana tipe data tersebut adalah tampilan pada bagian pemakaian daya dan tarif pembayaran pada tampilan aplikasi, dari sinilah fungsi yang membuat angka pada tampilan aplikasi bagian daya dan tarif juga dapat berubah-ubah sesuai pemakaian pada alat yang terpasang dalam ruang kontrol. *Firestore* merupakan *firebase* yang digunakan untuk menerima data khusus untuk daya dan tarif dari database *realtime* yang telah dibuat pada *firebase*.



Gambar 3.13. Blok 2 Program Aplikasi Reset Daya

Selanjutnya adalah batas daya. Pada aplikasi *smartphone* nantinya harus dapat menulis batasan penggunaan daya ke *google firebase* dan selanjutnya diakses oleh alat sehingga dapat otomatis terputus listrik pada beban apabila batas daya sudah tercapai. Namun apabila ingin mereset penggunaan daya pada suatu terminal, maka cukup menekan tombol reset, maka penggunaan daya listriknya akan kembali menjadi 0 sehingga beban listrik kembali dapat digunakan. Dibawah adalah programnya:



Gambar 3.14. Blok 3 Program Aplikasi Batas Daya

Pada blok di atas merupakan blok yang berfungsi untuk mengakses pengisian batas penggunaan daya listrik. Pada saat nilai dimasukkan pada kolom *textbox* maka nilai tersebut akan langsung terkirim ke *google firebase* untuk diakses oleh alat.

3.2. Pengujian Sistem

Pengujian dilakukan dengan memberikan daya sebesar 10 kwh pada setiap terminal, kemudian sistem dijalankan dengan memberikan beban hingga 10 kwh terpenuhi. Berikut disajikan tabel pengujian daya listrik:

Tabel 3.2 Pengujian Terminal 1

No	Batas daya (kwh)	Jenis beban 1	Daya terpakai 1	Waktu (menit)	Tarif (rp)	Kondisi beban1
1	10	Dispenser (200Watt) Lampu (15Watt)	0,0	0	0	ON
2			1,0	1,6	1.5	ON
3			2,0	2,4	3.0	ON
4			3,0	4,1	4.5	ON
5			4,0	6,1	6.0	ON
6			5,0	7,2	7.5	ON
7			6,0	8,1	9.0	ON
8			7,0	9,0	10.5	ON
9			8,0	10,2	12.0	ON
10			9,0	11,6	13.5	ON
11			10,0	12,3	15.0	OFF



Berdasarkan penelitian di atas terminal 1 dengan beban dispenser (200 Watt) dan lampu (15 Watt) dengan batasan daya yang di berikan sebesar 10 KWH maka di peroleh lamanya waktu yang didapat untuk menghabiskan daya 10 KWH selama 12,3 menit sedangkan tarif yang dibayar sebesar Rp 15,2. Maka secara otomatis jika daya pemakaian sudah melewati daya maksimal yang diberikan terminal listrik akan mati, gunakan tombol reset pada *android* untuk mengubah daya dan tarif menjadi 0 kembali.

Tabel 3.3 Pengujian Terminal 2

No.	Batas daya (kwh)	Jenis beban 2	Daya terpakai ²	Waktu (menit)	Tarif (rp)	Kondisi beban ²
1	10	Kipas Angin (50Watt) Charger HP (5Watt)	0,0	0	0	ON
2			1,0	4	1.5	ON
3			2,0	9	3.0	ON
4			3,0	13	4.5	ON
5			4,0	17	6.0	ON
6			5,0	22	7.5	ON
7			6,0	27	9.0	ON
8			7,0	31	10.5	ON
9			8,0	36	12.0	ON
10			9,0	41	13,5	ON
11			10,0	44	15.0	OFF

Berdasarkan penelitian di atas terminal 2 dengan beban kipas angin (50 Watt) dan charger HP (5 Watt) dengan batasan daya yang diberikan sebesar 10 KWH maka di peroleh lamanya waktu yang didapat untuk menghabiskan daya 10 KWH selama 44 menit sedangkan tarif yang dibayar sebesar Rp 15,2. Maka secara otomatis jika daya pemakaian sudah melewati daya maksimal yang diberikan terminal listrik akan mati, gunakan tombol reset pada *android* untuk mengubah daya dan tarif menjadi 0 kembali.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, maka peneliti dapat mengambil kesimpulan sebagai berikut:

- Sistem Kontrol Penggunaan Listrik Pascabayar Menggunakan Android terdiri dari Sistem Monitoring yaitu pemantauan penggunaan daya listrik dan tarif pembayaran, dan Sistem Pengontrolan yaitu pemutusan dan penyambungan arus listrik secara otomatis melalui aplikasi *smartphone*.
- Berdasarkan hasil pengujian terminal listrik didapatkan bahwa tarif listrik akan sama ketika menggunakan daya yang sama, namun akan menghasilkan waktu pemakaian yang berbeda menyesuaikan dengan beban yang diberikan. Dalam hal ini untuk daya 1 Kwh tarif yang harus dibayar sebesar 1,5 rupiah dengan lama pemakaian 1,6 menit untuk beban 215 watt dan 4 menit untuk beban 55 watt.

4.2 Saran

Berdasarkan hasil penelitian yang menyatakan bahwa produk ini sudah layak, namun dalam rangka upaya penyempurnaan produk maka penulis menyarankan:

- Pengembangan selanjutnya, diharapkan agar alat dirancang untuk penggunaan pada cakupan ruangan yang luas dengan sistem kelistrikan yang lebih kompleks.
- Diharapkan agar sistem yang dibuat diuji lebih maksimal, yang meliputi tingkat ketahanan alat dan aplikasi yang terstandarisasi dan siap digunakan pada masyarakat.

DAFTAR PUSTAKA

- Tim CNN Indonesia, (14 Nov 2021), *Perbedaan Listrik Prabayar dan Pascabayar: Kelebihan dan Kekurangan*, Available: <https://www.cnnindonesia.com/ekonomi/20211111173849-90-719901/perbedaan-listrik-prabayar-dan-pascabayar-kelebihan-dan-kekurangan>.
- Rinaldi Jasmi, Zahara Putri, Mia Rosmiati, S.Si, M.T., "Smart Metering : Monitoring Calculation Of Electric Current And Load Using Arduino Uno Android Application, e-Proceeding of Applied Science : Vol.6, No.2, Page 2623, ISSN : 2442-5826, Agustus 2020.





- [3] Diva Lufiana Putri, Inten Esti Pratiwi, (2 April 2023) *Daftar Tarif Listrik Per Kwh yang Berlaku Mulai April-Juni 2023*, Available: <https://www.kompas.com/tren/read/2023/04/02/141500765/daftar-tarif-listrik-per-kwh-yang-berlaku-mulai-april-juni-2023>.
- [4] Reporter Tempo.co, Nurhadi, (14 Juni 2022) *Listrik Prabayar dan Pascabayar, Mana yang Lebih Hemat?*, <https://bisnis.tempo.co/read/1601753/listrik-prabayar-dan-pascabayar-mana-yang-lebih-hemat>.
- [5] Tukadi1, Wahyu Widodo, Maretha Ruswiensari, Aryo Qomar, "Monitoring Pemakaian Daya Listrik Secara Realtime Berbasis Internet Of Things", Seminar Nasional Sains dan Teknologi Terapan 2019, Institute Teknologi Adhi Tama Surabaya, ISSN(p) : 2686-0023, ISSN(e)2685-6875, 2019, <http://ejurnal.itats.ac.id/sntekpan/article/view/659>
- [6] Syamsu alam, Figur Muhammad, Muhammad Ikbal M. Rizal "Rancang Bangun Sistem Anti Ngantuk Pada Pengendara Berbasis Arduino Nano", Jurnal IT, Vol 10 No 3, ISSN(p) 2087-6505, ISSN(e) 2550-0511, Desember 2019 .
- [7] M. Emil Bashofi, Rahmat Zainul Abidin, "Implementasi Firebase Pada Sistem Kendali Lampu Jarak Jauh Berbasis Android", Jurnal Explore It!, Vol. 10 No. 2, page: 50-62 ISSN(p) 2086-3489, ISSN(e) 2549-354X, Desember 2018.
- [8] Andriana, Zuklarnain, Hadi Baehaqi, "Sistem kWH Meter Digital Menggunakan Modul PZEM-004T", Jurnal Tiarsie, Vol.16, No. 1, page: 29-34, ISSN 2623-2391, July 2019
- [9] Muhamad Saleh, Munnik Haryanti "Rancang Bangun Sistem Keamanan Rumah Menggunakan Relay", Jurnal Teknologi Elektro, Universitas Mercu Buana, Vol.8, No 2, page: 87-94 ISSN : 2086-9479, Mei 2017.

