

BAB IV

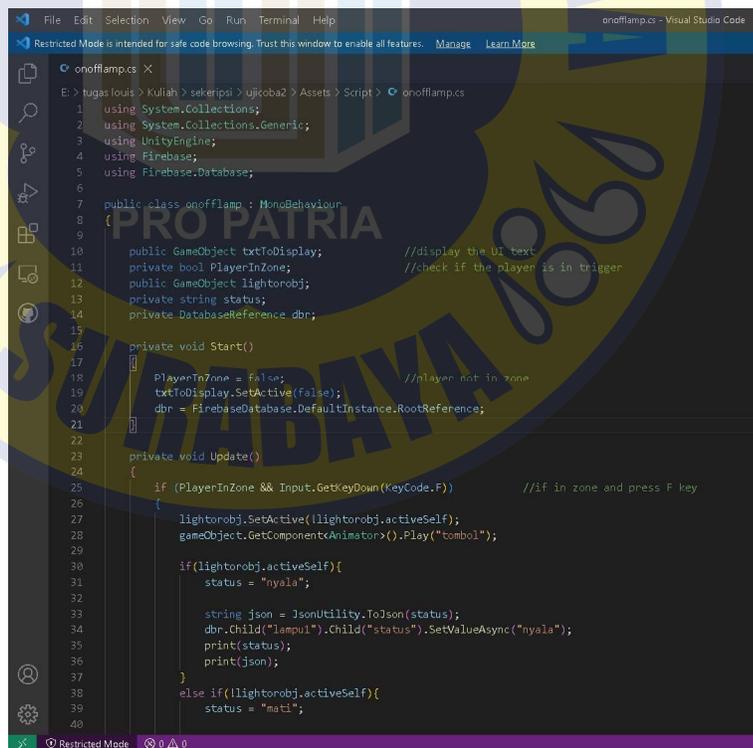
HASIL DAN PEMBAHASAN

4.1 Implementasi

Pada tahap ini akan dilakukan konfigurasi untuk mengintegrasikan IoT dengan metaverse, dimulai dari implementasi pada unity, dilanjut dengan implementasi pada esp32

4.1.1 Implementasi pada unity

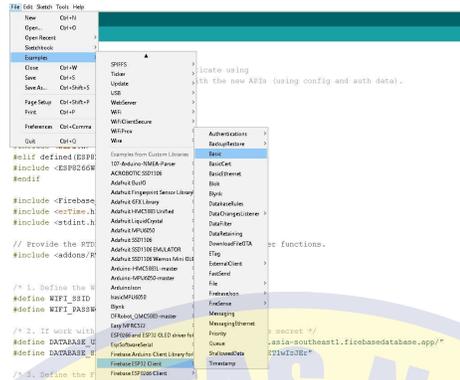
Pada tahap ini dilakukan pemrograman yang di perlukan agar unity dapat terkoneksi ke Firebase



```
File Edit Selection View Go Run Terminal Help onofflamp.cs - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

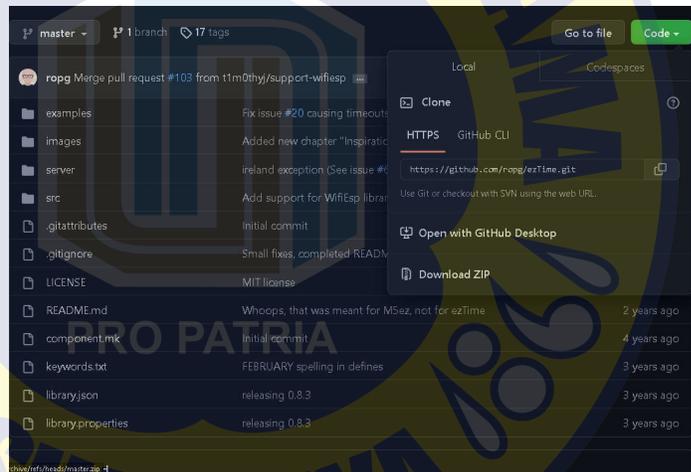
onofflamp.cs X
E:\> tugas\lout\Kuliah\sekeripsi\ujicoba2\Assets> Script > onofflamp.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using Firebase;
5 using Firebase.Database;
6
7 public class onofflamp : MonoBehaviour
8 {
9
10     public GameObject txtToDisplay; //display the UI text
11     private bool PlayerInZone; //check if the player is in trigger
12     public GameObject lightorobj;
13     private string status;
14     private DatabaseReference dbr;
15
16     private void Start()
17     {
18         PlayerInZone = false; //player not in zone
19         txtToDisplay.SetActive(false);
20         dbr = FirebaseDatabase.DefaultInstance.RootReference;
21     }
22
23     private void Update()
24     {
25         if (PlayerInZone && Input.GetKeyDown(KeyCode.F)) //if in zone and press F key
26         {
27             lightorobj.SetActive(!lightorobj.isActiveSelf);
28             gameObject.GetComponent<Animator>().Play("tombol");
29
30             if (lightorobj.isActiveSelf){
31                 status = "nyala";
32
33                 string json = JsonUtility.ToJson(status);
34                 dbr.Child("lampul").Child("status").SetValueAsync("nyala");
35                 print(status);
36                 print(json);
37             }
38             else if (!lightorobj.isActiveSelf){
39                 status = "mati";
40             }
41         }
42     }
43 }
```

Gambar 4. 1 Script lampu



Gambar 4. 3 Konfigurasi ESP32

Pada gambar 4.3 dilakukan pemilihan contoh code yang telah disediakan oleh library firebase yang di install



Gambar 4. 4 Penambahan library eztime

Pada gambar 4.4, dilakukan penambahan library eztime yang berguna untuk mendapatkan data tanggal aktual

Name	Date modified	Type	Size
Adafruit_Circuit_Playground	20/12/2021 8:38	File folder	
Bridge	20/12/2021 8:38	File folder	
Explora	20/12/2021 8:38	File folder	
Ethernet	20/12/2021 8:38	File folder	
ezTime-master	28/12/2022 17:30	File folder	
Firmata	20/12/2021 8:38	File folder	
GSM	20/12/2021 8:38	File folder	
HMC5883L_Simple	21/12/2021 9:05	File folder	
Keyboard	20/12/2021 8:38	File folder	
LiquidCrystal	20/12/2021 8:38	File folder	
Mouse	20/12/2021 8:38	File folder	
Robot_Control	20/12/2021 8:38	File folder	
Robot_Motor	20/12/2021 8:38	File folder	
RobotIRremote	20/12/2021 8:38	File folder	
SD	20/12/2021 8:38	File folder	
Servo	20/12/2021 8:38	File folder	
SpacebrewYun	20/12/2021 8:38	File folder	

Gambar 4. 5 Pemindahan librari eztime

Pada gambar 4.5, file eztime yang telah di unduh dan di ekstrak kemudian di pindah kedalam folder libraries pada folder arduino ide

```
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif

#include <Firebase_ESP_Client.h>
#include <ezTime.h>
#include <stdint.h>
```

Gambar 4. 6 Penggunaan eztime library

Pada gambar 4.6 di tambahkan eztime library pada source code agar dapat mengakses semua fungsi yang ada di dalam library eztime berikut adalah format dari eztime untuk menampilkan tanggal

UTC.dateTime("l, d-M-y H:i:s.v T") akan menampilkan tanggal dengan format Saturday, 25-Aug-18 14:32:53.282 UTC

Tabel 4. 1 Format EZtime

Kode	Arti
d	Hari dalam bulan, 2 digit diawali 0
D	3 huruf pertama hari dalam bahasa inggris
j	Hari dalam bulan tanpa diawali angka 0
l	Menampilkan hari dalam bahasa inggris
m	Menampilkan angka bulan diawali dengan 0
M	Menampilkan nama depan bulan dalam bahasa inggris
n	Menampilkan angka bulan tanpa diawali 0
Y	Menampilkan tahun secara lengkap 4 digit
y	Menampilkan tahun 2 digit dari belakang
H	Format 24 jam diawali dengan 0
i	Menampilkan Menit diawali 0
s	Menampilkan detik diawali 0
v	Menampilkan 3 digit mili detik
T	Menampilkan zona waktu

```

/* 1. Define the WiFi credentials */
#define WIFI_SSID "amatsumagatsuci"
#define WIFI_PASSWORD "lalalala"

/* 2. If work with RTDB, define the RTDB URL and database secret */
#define DATABASE_URL "https://iot-metaverse-default-rtdb.asia-southeast1.firebaseio.com/"
#define DATABASE_SECRET "L6V21gydk8CJUWEQKozEGh7TI4mLiJVETlwI2JBr"

/* 3. Define the Firebase Data object */

```

Gambar 4. 7 Edit kode ESP32

Pada gambar 4.7 dilakukan perubahan pada WIFI_SSID, WIFI_PASSWORD, DATABASE_URL dan DATABASE_SECRET, pada bagian wifi masukkan SSID sesuai nama wifi yang akan di gunakan untuk terhubung ke internet, begitu juga dengan PASSWORD nya, pada bagian DATABASE_URL ubah link sesuai dengan link database yang di gunakan, begitu juga dengan DATABASE_SECRET nya.

```
//===== Firebase =====
float tg11; float tg111; float tg121;
float tg12; float tg112; float tg122;
float tg13; float tg113; float tg123;
float tg14; float tg114; float tg124;
float tg15; float tg115; float tg125;
float tg16; float tg116; float tg126;
float tg17; float tg117; float tg127;
float tg18; float tg118; float tg128;
float tg19; float tg119; float tg129;
float tg110; float tg120; float tg130;
float tg131;
float total;
```

Gambar 4. 8 Penambahan variabel

Pada gambar 4.8 dilakukan penambahan variabel yang akan digunakan untuk menyimpan tanggal untuk dikirim ke firebase.

```
//-----
//===== ezTIME =====
Timezone Surabaya;
//-----
```

Gambar 4. 9 Penambahan variabel

Pada gambar 4.9 dilakukan penambahan variabel yang akan digunakan untuk mengakses isi dari library eztime.

```
//===== Relay =====
const int relay1 = 33;
const int relay2 = 25;
const int relay3 = 26;
const int relay4 = 27;
//-----
```

Gambar 4. 10 Penambahan variabel

Pada gambar 4.9 dilakukan penambahan variabel untuk inialisasi pin yang digunakan untuk output relay.

```
String lampu1;  
String lampu2;
```

Gambar 4. 11 Penambahan variabel

Pada gambar 4.11 dilakukan penambahan variabel lampu yang akan digunakan untuk menyimpan nilai dari firebase

```
pinMode (relay1, OUTPUT) ; //D27  
pinMode (relay2, OUTPUT) ; //D26  
pinMode (relay3, OUTPUT) ; //D25  
pinMode (relay4, OUTPUT) ; //D33
```

Gambar 4. 12 Penambahan kode program

Pada gambar 4.12 dilakukan penambahan kode di dalam *VOID SETUP* untuk inialisasi output pada pin yang telah di tentukan

```
void tanggalan() {  
  waitForSync();  
  
  Surabaya.setPosix("Surabaya-7");  
  Serial.println(Surabaya.dateTime());  
  
  int jam = Surabaya.hour();  
  int mnt = Surabaya.minute();  
  int hri = Surabaya.day();  
  int bln = Surabaya.month();  
  int thn = Surabaya.year();  
  
  Serial.println(jam);  
  Serial.println(mnt);  
  Serial.println(hri);  
  Serial.println(bln);  
  Serial.println(thn);  
}
```

Gambar 4. 13 Penambahan fungsi tanggalan

Pada gambar 4.13 dilakukan penambahan *void* tanggal yang berfungsi untuk mengambil dan menampilkan data tanggal

```
void connectfirebase()
{
  Firebase.FirebaseApp.initializeApp(FirebaseOptions.builder().setApiKey("AIzaSyA...").build());
  FirebaseRTDB firebaseRTDB = FirebaseDatabase.getInstance("https://surabaya.firebaseio.com");
  DatabaseReference databaseReference = firebaseRTDB.reference("lampu");

  Serial.println("Set Tanggal");
  Serial.println("Set Hari");
  Serial.println("Set Bulan");
  Serial.println("Set Tahun");
  Serial.println("Set Jam");
  Serial.println("Set Momen");
  Serial.println("Set Daya");
  Serial.println("Set Log Matri");
  Serial.println("Set Log Daya");
  Serial.println("Set Log Tegangan");
}
```

Gambar 4. 14 Penambahan fungsi connect firebase

Dapat dilihat pada gambar 4.14 dilakukan penambahan kode program untuk mengirim semua data yang berhasil di ambil esp32 ke firebase

```
if (FirebaseRTDB.getString(fbdo, "/lampu/status")) {
  if (fbdo.dataType() == "string") {
    lampu = fbdo.stringData();
    if (lampu == "nyala") {
      digitalWrite(relay1, LOW);
    }
    else if (lampu == "mati") {
      digitalWrite(relay1, HIGH);
    }
  }
}

//=====Jumlah Rwh=====
if (FirebaseRTDB.getFloat(fbdo, ("/log-pln1/Daya"))) {
  if (fbdo.dataType() == "float") {
    tgl1 = fbdo.floatData();
    Serial.println(tgl1);
  }
}

if (FirebaseRTDB.getFloat(fbdo, ("/log-pln2/Daya"))) {
  if (fbdo.dataType() == "float") {
    tgl2 = fbdo.floatData();
    Serial.println(tgl2);
  }
}

if (FirebaseRTDB.getFloat(fbdo, ("/log-pln3/Daya"))) {
  if (fbdo.dataType() == "float") {
    tgl3 = fbdo.floatData();
    Serial.println(tgl3);
  }
}
```

Gambar 4. 15 Penambahan fungsi connect firebase

Pada gambar 4.15 dilakukan penambahan kode program untuk memvalidai data lampu yang di ambil dari metaverse melalui firebase untuk menyalakan lampu pada iot

Disusul dengan penambahan kode program untuk mengirim total Kwh per hari dimulai dari tanggal 1 sampai dengan tanggal 3

```
if (Firebase.RTDB.getInt(fbdo, ("/log-pin4/Daya"))) {
  if (fbdo.dataType() == "float") {
    tg14 = fbdo.floatData();
    Serial.println(tg14);
  }
}

if (Firebase.RTDB.getInt(fbdo, ("/log-pin5/Daya"))) {
  if (fbdo.dataType() == "float") {
    tg15 = fbdo.floatData();
    Serial.println(tg15);
  }
}

if (Firebase.RTDB.getInt(fbdo, ("/log-pin6/Daya"))) {
  if (fbdo.dataType() == "float") {
    tg16 = fbdo.floatData();
    Serial.println(tg16);
  }
}

if (Firebase.RTDB.getInt(fbdo, ("/log-pin7/Daya"))) {
  if (fbdo.dataType() == "float") {
    tg17 = fbdo.floatData();
    Serial.println(tg17);
  }
}

if (Firebase.RTDB.getInt(fbdo, ("/log-pin8/Daya"))) {
  if (fbdo.dataType() == "float") {
    tg18 = fbdo.floatData();
    Serial.println(tg18);
  }
}
```

Gambar 4. 16 Penambahan fungsi connect firebase

Pada gambar 4.16 dilakukan penambahan kode program untuk mengirim total Kwh per hari dimulai dari tanggal 4 sampai dengan tanggal 8

```
if (Firebase.RTDB.getInt(fbdo, ("/log-pin9/Daya"))) {
  if (fbdo.dataType() == "float") {
    tg19 = fbdo.floatData();
    Serial.println(tg19);
  }
}

if (Firebase.RTDB.getInt(fbdo, ("/log-pin10/Daya"))) {
  if (fbdo.dataType() == "float") {
    tg110 = fbdo.floatData();
    Serial.println(tg110);
  }
}

if (Firebase.RTDB.getInt(fbdo, ("/log-pin11/Daya"))) {
  if (fbdo.dataType() == "float") {
    tg111 = fbdo.floatData();
    Serial.println(tg111);
  }
}

if (Firebase.RTDB.getInt(fbdo, ("/log-pin12/Daya"))) {
  if (fbdo.dataType() == "float") {
    tg112 = fbdo.floatData();
    Serial.println(tg112);
  }
}

if (Firebase.RTDB.getInt(fbdo, ("/log-pin13/Daya"))) {
  if (fbdo.dataType() == "float") {
    tg113 = fbdo.floatData();
    Serial.println(tg113);
  }
}
```

Gambar 4. 17 Penambahan fungsi connect firebase

Pada gambar 4.17 dilakukan penambahan kode program untuk mengirim total Kwh per hari dimulai dari tanggal 9 sampai dengan tanggal 13

```
if (Firebase.FTDB.getInt(fbdo, ("/log-pin14/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl14 = fbdo.floatData();  
    Serial.println(tgl14);  
  }  
}  
  
if (Firebase.FTDB.getInt(fbdo, ("/log-pin15/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl15 = fbdo.floatData();  
    Serial.println(tgl15);  
  }  
}  
  
if (Firebase.FTDB.getInt(fbdo, ("/log-pin16/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl16 = fbdo.floatData();  
    Serial.println(tgl16);  
  }  
}  
  
if (Firebase.FTDB.getInt(fbdo, ("/log-pin17/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl17 = fbdo.floatData();  
    Serial.println(tgl17);  
  }  
}  
  
if (Firebase.FTDB.getInt(fbdo, ("/log-pin18/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl18 = fbdo.floatData();  
    Serial.println(tgl18);  
  }  
}
```

Gambar 4. 18 Penambahan fungsi connect firebase

Pada gambar 4.18 dilakukan penambahan kode program untuk mengirim total Kwh per hari dimulai dari tanggal 14 sampai dengan tanggal 18

```
if (Firebase.FTDB.getInt(fbdo, ("/log-pin19/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl19 = fbdo.floatData();  
    Serial.println(tgl19);  
  }  
}  
  
if (Firebase.FTDB.getInt(fbdo, ("/log-pin20/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl20 = fbdo.floatData();  
    Serial.println(tgl20);  
  }  
}  
  
if (Firebase.FTDB.getInt(fbdo, ("/log-pin21/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl21 = fbdo.floatData();  
    Serial.println(tgl21);  
  }  
}  
  
if (Firebase.FTDB.getInt(fbdo, ("/log-pin22/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl22 = fbdo.floatData();  
    Serial.println(tgl22);  
  }  
}  
  
if (Firebase.FTDB.getInt(fbdo, ("/log-pin23/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl23 = fbdo.floatData();  
    Serial.println(tgl23);  
  }  
}
```

Gambar 4. 19 Penambahan fungsi connect firebase

Dapat dilihat pada gambar 4.19 dilakukan penambahan source code pada esp32 untuk mengirim data Kwh pada tanggal 19 sampai dengan tanggal 23

```
if (Firebase.RTDB.getInt(sfbdo, ("/log-pin24/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl24 = fbdo.floatData();  
    Serial.println(tgl24);  
  }  
}  
  
if (Firebase.RTDB.getInt(sfbdo, ("/log-pin25/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl25 = fbdo.floatData();  
    Serial.println(tgl25);  
  }  
}  
  
if (Firebase.RTDB.getInt(sfbdo, ("/log-pin26/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl26 = fbdo.floatData();  
    Serial.println(tgl26);  
  }  
}  
  
if (Firebase.RTDB.getInt(sfbdo, ("/log-pin27/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl27 = fbdo.floatData();  
    Serial.println(tgl27);  
  }  
}  
  
if (Firebase.RTDB.getInt(sfbdo, ("/log-pin28/Daya"))) {  
  if (fbdo.dataType() == "float") {  
    tgl28 = fbdo.floatData();  
    Serial.println(tgl28);  
  }  
}
```

Gambar 4. 20 Penambahan fungsi connect firebase

Pada gambar 4.20 dilakukan penambahan kode program untuk mengirim total Kwh per hari dimulai dari tanggal 24 sampai dengan tanggal 28



```
if (fbdo.dataType() == "float") {  
  tgl28 = fbdo.floatData();  
  Serial.println(tgl28);  
}  
  
if (fbdo.dataType() == "float") {  
  tgl29 = fbdo.floatData();  
  Serial.println(tgl29);  
}  
  
if (fbdo.dataType() == "float") {  
  tgl30 = fbdo.floatData();  
  Serial.println(tgl30);  
}  
  
if (fbdo.dataType() == "float") {  
  tgl31 = fbdo.floatData();  
  Serial.println(tgl31);  
}
```

Gambar 4. 21 Penambahan fungsi connect firebase

Pada gambar 4.21 dilakukan dilakukan penambahan kode program untuk mengirim total Kwh per hari dimulai dari tanggal 29 sampai dengan tanggal 31

Disusul dengan kode program untuk menjumlah total Kwh sebulan penuh, serta mengirim data tersebut ke firebase

```
tanggalan ();  
connectFirebase ();
```

Gambar 4. 22 Penambahan kode pada void loop

Pada gambar 4.22 dilakukan penambahan kode di dalam void loop yang bertujuan untuk mengulangi fungsi tanggalan dan connectFirebase

4.2 Pengujian

Pada tahap ini dilakukan pengujian terhadap semua konfigurasi yang telah di implementasikan yang bertujuan untuk mengetahui jika proses integrasi telah berhasil

4.2.1 Pengujian pada Unity



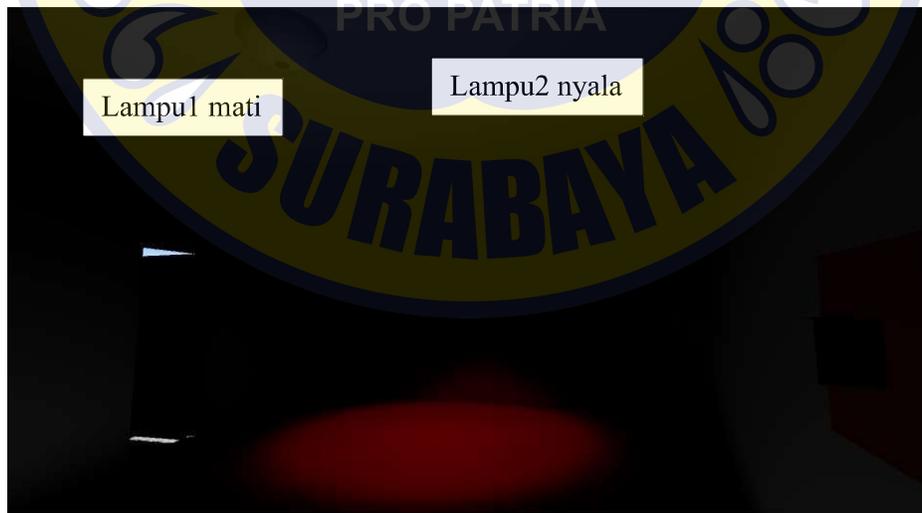
Gambar 4. 23 Lampu1 menyala

Dapat dilihat pada gambar 4.23 lampu1 menyala dan lampu2 mati, lampu1 digambarkan dengan lampu putih yang menerangi ruangan, data lampu akan dikirim ke firebase seperti pada gambar 4.24 dibawah



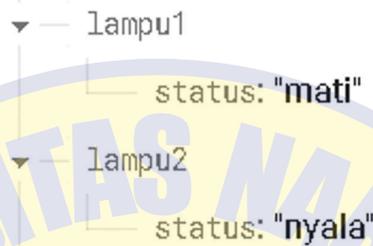
Gambar 4. 24 Status lampu1 pada database

Seperti yang dapat dilihat pada gambar 4.24, saat lampu1 pada unity dinyalakan, data berhasil terkirim ke database dan status berubah nyala



Gambar 4. 25 Lampu2 menyala

Dapat dilihat pada gambar 4.25 bahwa lampu1 mati sedangkan lampu2 menyala, lampu2 digambarkan dengan lampu berwarna merah yang menerangi lantai, lalu data lampu dikirim ke firebase seperti pada gambar 4.26



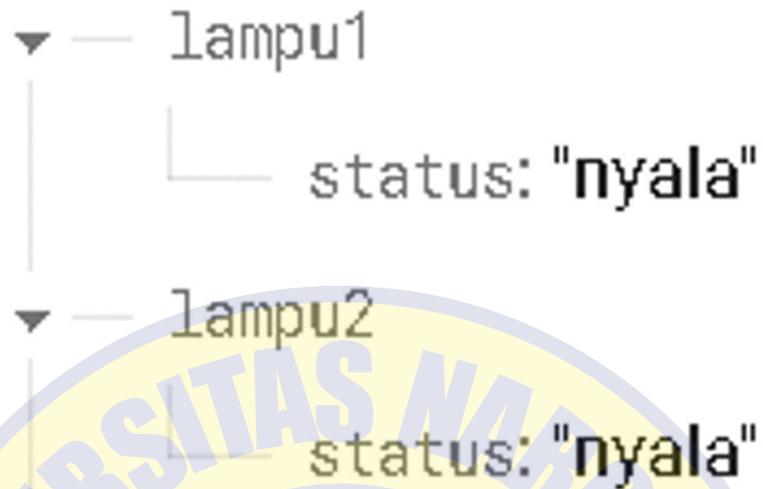
Gambar 4. 26 Status lampu2 pada database

Begitu juga saat menyalakan lampu2, data lampu2 berhasil terkirim ke database dan status lampu2 berubah menjadi menyala



Gambar 4. 27 Kedua lampu menyala

Pada gambar 4.27 dapat dilihat bahwa lampu1 dan lampu2 menyala dan status lampu tersebut berhasil terkirim ke firebase seperti pada gambar 4.28



Gambar 4. 28 Status Lampu pada Database

Dapat dilihat pada gambar 4.28, saat kedua lampu dinyalakan semua status lampu pada database berubah menjadi nyala

4.2.2 Pengujian pada Arduino

```

log-p1n12
log-p1n13
log-p1n14
log-p1n15
log-p1n16
log-p1n17
menit: 46
tanggal
tanggalan: "Tuesday, 17-Jan-2023 23:46:21 Surabaya"
tegangan: 198.89999
watt: 9,1
  
```

Gambar 4. 29 Pengujian Arduino

Pada gambar 4.29 dapat dilihat bahwa setelah melakukan konfigurasi pada esp32, data yang didapat berhasil terkirim ke firebase

Total Kwh Bulan Jan: 2.681
arus: 0.068
daya: 0.565
jam: 23

Gambar 4. 30 Pengujian Arduino

Dapat dilihat pada gambar 4.29 dan 4.30 bahwa setelah melakukan konfigurasi pada esp32, data berhasil terkirim ke database, **Total Kwh Bulan Jan** menandakan kode program untuk mengirim data dari esp32 ke firebase berhasil dan integrasi esp32 dengan firebase sukses

4.2.3 Pengujian Keseluruhan



Gambar 4. 31 Pengujian keseluruhan

Dapat dilihat pada gambar 4.31, saat lampu pada unity dimatikan maka semua lampu pada esp32 akan ikut mati



Gambar 4. 32 Pengujian keseluruhan

Pada gambar 4.32, lampu1 pada unity telah dinyalakan, dan hasilnya lampu1 pada esp32 menyala



Gambar 4. 33 Pengujian Keseluruhan

Pada gambar 4.33, terlihat bahwa lampu2 pada IoT dalam kondisi menyala sedangkan lampu2 pada IoT dalam keadaan mati, hal ini terjadi karena lampu1 pada unity dinonaktifkan sedangkan lampu2 pada unity di aktifkan



Gambar 4. 34 Pengujian Keseluruhan

Pada gambar 4.34, dapat dilihat bahwa semua lampu pada IoT menyala hal ini dikarenakan lampu1 dan lampu2 pada unity dinyalakan, maka esp32 akan merespon dengan menyalakan kedua lampu

Pada pengujian kali ini menggunakan metode *Black Box* yang dimana data kegiatan pengujian dijadikan sebuah tabel untuk memudahkan pendataan kegiatan, dimulai dari langkah pengujian, realisasi yang diharapkan, hasil pengujian, dan kesimpulan seperti yang dapat di lihat pada tabel 4.2 di bawah

Tabel 4. 2 Tabel Pengujian

Langkah Pengujian	Realisasi yang diharapkan	Hasil pengujian	Kesimpulan
Lampu1 pada Unity ON Lampu2 pada Unity OFF	Lampu1 pada IoT ON Lampu2 pada IoT OFF	Lampu1 pada IoT ON Lampu2 pada IoT OFF	Berhasil

Lampu1 pada Unity OFF Lampu2 pada Unity ON	Lampu1 pada IoT OFF Lampu2 pada IoT ON	Lampu1 pada IoT OFF Lampu2 pada IoT ON	Berhasil
Lampu1 pada Unity OFF Lampu2 pada Unity OFF	Lampu1 pada IoT OFF Lampu2 pada IoT OFF	Lampu1 pada IoT OFF Lampu2 pada IoT OFF	Berhasil
Lampu1 pada Unity ON Lampu2 pada Unity ON	Lampu1 pada IoT ON Lampu2 pada IoT ON	Lampu1 pada IoT ON Lampu2 pada IoT ON	Berhasil

