

BAB IV

HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan pembahasan sistem kontroler *Air Conditioner* menggunakan ESP8266 sebagai webserver, yang terdiri dari hasil perancangan dan pengujian tiap sensor suhu, inframerah dan warna, serta pengujian modul ESP8266-01 dengan Arduino Mega 2560. Hasil pengujian berupa data-data yang diperoleh untuk memperlihatkan kemampuan perangkat keras dan perangkat lunak yang dirancang dapat bekerja dengan baik atau tidak. Dari data-data tersebut dapat dilakukan analisis kinerja perangkat secara keseluruhan berdasarkan pengujian dari masing-masing sistem untuk mendapatkan hasil yang lebih baik dan lebih akurat serta kemudian dapat digunakan untuk mengambil kesimpulan akhir terhadap perangkat tersebut.

4.1 Hasil Perancangan *Hardware*

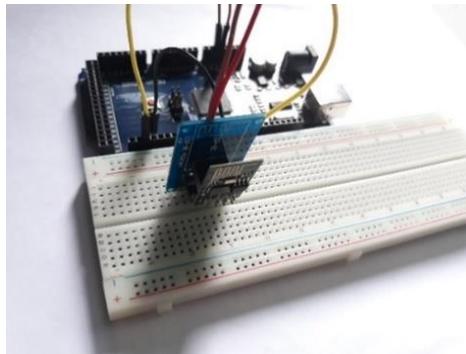
Hasil perancangan *hardware* ini terdiri dari Arduino Mega 2560, Modul ESP8266-01, Sensor suhu, Sensor warna sebagai input dan Inframerah sebagai keluarannya. Hasil dari rangkaian *hardware* digunakan untuk sistem kontroler dan monitoring ditunjukkan pada gambar 4.1



Gambar 4. 1 Rangkaian *Hardware*

4.2 Pengujian Wi-Fi ESP8266-01

Proses pengiriman data sensor suhu dan Warna ke *web browser* menggunakan modul Wi-Fi ESP8266, dalam penelitian ini menggunakan Wi-Fi ESP8266-01. ESP8266-01 ini dihubungkan dengan arduino Mega 2560 untuk menjalankan perintah ESP8266 sebagai *Station* dalam hal ini ESP8266 sebagai *Web Server*. Pada gambar 4.2 menunjukkan ESP8266-01 yang terhubung dengan Arduino Mega 2560.



Gambar 4. 2 ESP8266-01 yang terhubung dengan Arduino Mega 2560

4.2.1 *Station Mode* (Mode Penerima)

Pada mode ini ESP8266 akan difungsikan sebagai *Station* (Penerima) yang akan menangkap sinyal dari *Access Point* (AP). Untuk menggunakan mode *Station* pada Skrip harus disetel terlebih dahulu *Access Point* mana yang akan di sambungkan. Berikut adalah skema *station mode* ESP8266 dan *sketch* pada Arduino IDE



Gambar 4. 3 Skema station mode ESP8266

ESP01

```

#include <WiFiEsp.h>
//=====SSID&PASS=====
char ssid[] = "Dota 1";
char pass[] = "oke1234567";

//=====INISIALISASI I/O =====
WiFiEspServer server(80);
WiFiEspClient klien;
//Network : Static IP address
IPAddress ip(192,168,43,6);
int status = WL_IDLE_STATUS;

void setup() {
  // put your setup code here, to run once:
  Serial1.begin(9600);
  Serial.begin(115200);
  //=====KONEKSI KE JARINGAN=====
  WiFi.init(&Serial1);
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial1.println("Tidak ada modul WiFi");
    while (true);
  }
  while (status != WL_CONNECTED) {
    delay(500);
    Serial1.print("Koneksi ke SSID: ");
    Serial1.println(ssid);
    WiFi.config(ip);
    status = WiFi.begin(ssid,pass);
    delay(100);
  }
}

```

(a) Bagian Pertama

```

//=====Start the Server=====
server.begin();
Serial.println("Server started");
// Print the IP address
Serial.print("Use this URL to connect: ");
Serial.print("http://");
Serial.print(WiFi.localIP());
Serial.println("/");
delay(100);
}
void loop() {
  WiFiEspClient client = server.available();
  if (!client){
    return;
  }
  //tunggu client sends data
  Serial.println("new client");
  String request = "";
  while (client.connected()) {
    if (client.available()>0 ) {
      char c=client.read();
      request += c;
      if(c == '\n')
      {
        break;
      }
    }
  }
  Serial.println(request);
  client.flush();
  delay(100);
}

```

(b) Bagian Kedua

```

// Return the response
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // do not forget this one
client.println("<!DOCTYPE HTML>");
client.println("<html>");
delay(1);
client.stop(); //*****
Serial.println("Client disconnected");
Serial.println("");
}

```

(c) Bagian Ketiga

Gambar 4. 4 Sketch arduino IDE ESP8266-01

Pada gambar 4.4 bagian pertama menunjukkan bahwa arduino membutuhkan berkas WiFiEsp untuk menjalankan *web server*. Dapat diketahui juga bahwa ESP8266 akan terkoneksi dengan *Access Point* “Dota 1” dengan Password “oke1234567” dan IP address yang digunakan sebagai *web server* adalah 192.168.43.6. Komunikasi ESP8266 dan Arduino Mega 2560 menghubungkan pin Tx dan pin Rx pada ESP8266 masing – masing terhubung pada pin 18 dan pin 19 (terlihat pada Sketsa program Serial1) arduino Mega 2560. Pada proses koneksi jaringan ESP8266 program Arduino IDE, Program tersebut bekerja ketika Rx Tx membaca bahwa ESP8266 terhubung, saat ESP8266 sudah terhubung maka ESP8266 akan menangkap sinyal *Access Point* (AP) dan mengkoneksikannya sesuai SSID dan PASS *Access Point* yang sudah disetel sebelumnya pada Arduino IDE. Setelah ESP8266 dan *Access Point* terkoneksi maka akan muncul perintah untuk mengakses *web server* melalui *IP address* yang sudah diatur pada Arduino IDE.

Pada bagian kedua script tersebut menjelaskan mengenai proses kerja server, di awal bagian kedua skrip menjelaskan mengenai *IP address* yang dijadikan sebagai *web server* data yaitu *IP address* lokal yang telah disetel pada bagian pertama. Selanjutnya setiap *client* yang mengakses website akan diberi penamaan sesuai urutan akses.

Pada bagian ketiga skrip menjelaskan mengenai proses pada tampilan *web*, ketika proses akses website selesai maka ESP8266 akan otomatis melakukan *disconnect*. Berikut adalah hasil respon serial monitor dari skrip pada gambar 4.5.



```

[WiFiEsp] Initializing ESP module
[WiFiEsp] Initialization successful - 1.3.0
[WiFiEsp] IP address set 192.168.43.6
[WiFiEsp] Connected to Dota 1
[WiFiEsp] Server started on port 80
Server started
Use this URL to connect: http://192.168.43.6/
[WiFiEsp] New client 0
new client
GET / HTTP/1.1

[WiFiEsp] Disconnecting 0
Client disconnected

[WiFiEsp] New client 1
new client
GET /favicon.ico HTTP/1.1

[WiFiEsp] Disconnecting 1
Client disconnected

```

Gambar 4. 5 Respon ESP8266-01 pada serial monitor

4.3 Pengujian Sensor

Pengujian sensor dilakukan untuk mendapatkan nilai baca sensor yang mendekati nilai akurasi yang baik. Untuk hasil baca sensor suhu dapat dibandingkan dengan thermometer, untuk mengetahui keakurasian sensor.

4.3.1 Sensor Suhu

Sensor suhu yang digunakan dalam penelitian ini menggunakan modul sensor DS18B20 dengan tegangan 5 volt. Sensor ini dapat digunakan untuk mengetahui suhu ruangan dengan kualitas yang baik dan pembacaan data yang cepat. Pada gambar 4.6 menunjukkan sensor suhu dan pada gambar 4.7 menunjukkan program pada Arduino IDE untuk inialisasi nilai masukan.



Gambar 4. 6 Pengujian sensor suhu

```

DS18B20
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 3

OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

void setup(void)
{
  // start serial port
  Serial.begin(9600);

  // Start up the library
  sensors.begin();
}

void loop(void)
{
  sensors.requestTemperatures(); // kirim command untuk mendapatkan data
  Serial.println("Temperature is: ");
  Serial.print(sensors.getTempCByIndex(0)); // 0 untuk IC pertama pada wire
  delay(1000);
}

```

Gambar 4. 7 Inisialisasi program DS18B20 Arduino IDE

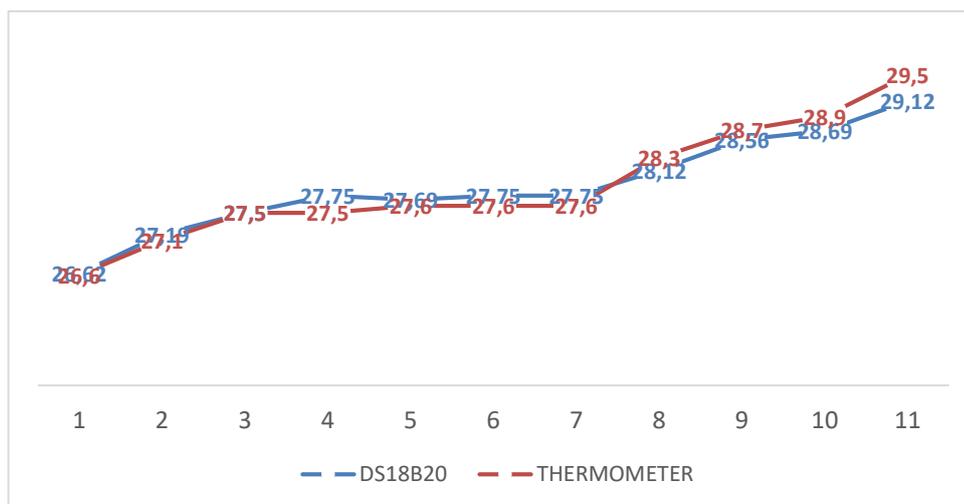
Pada penelitian ini rangkaian sensor suhu menggunakan Arduino Mega 2560 sebagai input untuk menjalankan program pada Arduino IDE. Dari pengujian pengukuran suhu dengan DS18B20 maka dapat dibandingkan dengan *thermometer*. Pada tabel 4.1 Hasil sensor DS18B20 dengan *thermometer*. Dimana hasil pengujian ini menggunakan suhu ruangan ber-AC.

Tabel 4. 1 Hasil pembacaan sensor DS18B20 dan *Thermometer*

No	Suhu keluaran AC	Suhu sensor DS18B20	Suhu thermometer	Error %
1	20 °C	26.62 °C	26.6 °C	0,08
2	21 °C	27.19 °C	27.1 °C	0,33
3	22 °C	27.50 °C	27.5 °C	0,00
4	23 °C	27.61 °C	27.5 °C	0,91
5	24 °C	27.69 °C	27.6 °C	0,33
6	25 °C	27.75 °C	27.6 °C	0,54
7	26 °C	27.75 °C	27.6 °C	0,54
8	27 °C	28.12 °C	28.3 °C	0,64
9	28 °C	28.56 °C	28.7 °C	0,49
10	29 °C	28.69 °C	28.9 °C	0,73
11	30 °C	29.12 °C	29.5 °C	1,29

Tabel 4.1 merupakan data hasil pengujian temperatur pada ruangan ber-AC. Pengambilan data berlangsung selama 88 menit dengan pengambilan data setiap 5 menit setelah pergantian suhu, 3 menit menunggu suhu normal kembali dan membandingkan langsung dengan thermometer. Pada pengujian ini didapat persentase error suhu tertinggi adalah 1,29% dan terendah 0%. Perbedaan tersebut dikarenakan sensitivitas serta keakuratan pada tiap sensor berbeda-beda.

Dari hasil pengukuran diatas dibuat dalam bentuk grafik dapat dilihat pada gambar 4.8

Gambar 4. 8 Grafik hasil pembacaan sensor DS18B20 dan *Thermometer*

Dari hasil data dan grafik data dapat dilihat bahwa *error* yang terjadi tidak terlalu besar, hal ini menunjukkan bahwa sensor dapat bekerja dengan baik. Sensor dikatakan bekerja dengan baik apabila perbandingan sensor suhu dengan thermometer tidak terlalu jauh.

4.3.2 Sensor Warna

Sensor warna yang digunakan pada penelitian ini menggunakan modul sensor TCS3200 dengan tegangan masukan 5 volt. Cara kerja sensor ini yaitu dengan cara mengkonversi warna cahaya menjadi nilai frekuensi. Keluaran dari sensor ini berupa output digital yang berbentuk pulsa-pulsa hasil pembacaan warna RGB. Pada gambar 4.9 menunjukkan sensor warna.



Gambar 4. 9 Sensor Warna

Pada prinsipnya pembacaan warna pada TCS3200 dilakukan secara bertahap yaitu membaca frekuensi warna dasar secara simultan dengan cara memfilter pada tiap-tiap warna dasar. Warna dasar tersebut meliputi RGB (*Red-Green-Blue*). Pada penelitian ini indikator LED AC (*Air Conditioner*) yang diperlukan hanyalah kondisi saat ON dan OFF, dimana saat ON indikator LED yang menyala adalah warna hijau dan saat OFF indikator LED tersebut mati. Maka dalam penelitian ini yang dilakukan konversi warna hanyalah warna hijau pada AC (*Air Conditioner*). Pada gambar 4.10 menunjukkan inisialisasi Program TCS3200 pada Arduino IDE dan gambar 4.11 menunjukkan skema proses konversi warna TCS3200.

TCS3200warna

```
#include <WiFiEsp.h>

//=====TCS3200=====
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8

int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;
int clearFrequency = 0;

int redColor = 0;
int greenColor = 0;
int blueColor = 0;
int clearColor = 0;

//=====SSID&PASS=====
char ssid[] = "Dota 1";
char pass[] = "oke123456";

//=====INISIALISASI I/O =====
WiFiEspServer server(80);
WiFiEspClient klien;

//Network : Static IP address
IPAddress ip(192,168,43,6);
int status = WL_IDLE_STATUS;
```

(a) Bagian Pertama

```
void setup() {
  // Setting output TCS3200
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);

  // Setting sensorOut sebagai input
  pinMode(sensorOut, INPUT);

  // Setting frekuensi scaling
  digitalWrite(S0,HIGH);
  digitalWrite(S1,HIGH);

  // put your setup code here, to run once:
  Serial1.begin(9600);
  Serial.begin(115200);

  //=====KONEKSI KE JARINGAN=====
  WiFi.init(&Serial1);

  if (WiFi.status() == WL_NO_SHIELD) {
    Serial1.println("Tidak ada modul WiFi");
    while (true);
  }
  while (status != WL_CONNECTED) {
    delay(500);
    Serial1.print("Koneksi ke SSID: ");
    Serial1.println(ssid);
    WiFi.config(ip);
    status = WiFi.begin(ssid,pass);
```

(b) Bagian Kedua

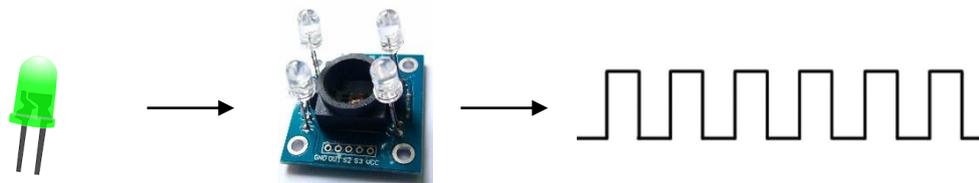
```

//=====BACA SENSOR TCS3200=====
// Setting warna hijau
digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
// baca frekuensi
greenFrequency = pulseIn(sensorOut, LOW);
delay(100);
// Return the response
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // do not forget this one
client.println("<html><head><title>Air Conditioner Controller UMY</title><meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1' ></head><body>");
client.println("<doctype html>");
client.println("<html>");
client.println("<h1> AC WEBSERVER CONTROLLER </h1>");
|
client.println("<br><br>");
client.println("<br>");
//print frekuensi pada web
client.print("G= "); //printing name
client.print(greenFrequency); //printing GREEN color frequency
client.print(" ");
client.println("</html>");
delay(1);
client.stop(); //*****
Serial.println("Client disconnected");
Serial.println("");
}

```

(c) bagian ketiga

Gambar 4. 10 inialisasi Program TCS3200 pada Arduino IDE



Gambar 4. 11 Skema proses konversi warna TCS3200

Sampling dilakukan 10x pada saat kondisi LED indikator AC *on* dan saat kondisi LED indikator AC *off*. Sampling dilakukan agar mendapat nilai frekuensi yang tepat dan memperkecil error dalam menentukan frekuensi indikator LED keadaan on atau off. Dalam hal ini sampling warna yang dibutuhkan hanyalah warna hijau yang menandakan indikator LED power pada AC. Berikut adalah data sampling sensor warna TCS3200 :

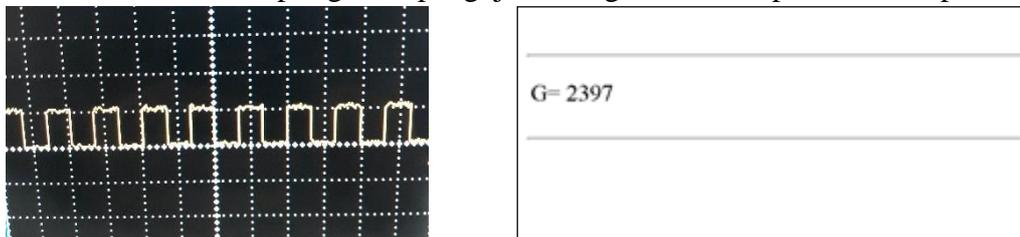
Tabel 4. 2 Data sampling sensor warna TCS3200

No	Frekuensi LED ON	Frekuensi LED OFF
1	2397	3211
2	2912	2930
3	2910	2943
4	2676	2955
5	2771	3100
6	2859	3256

Lanjutan Tabel 4.2 Data sampling sensor warna TCS3200

No	Frekuensi LED ON	Frekuensi LED OFF
7	2830	2925
8	2911	2935
9	2907	3012
10	2615	2924

Dari hasil sampling dan pengujian dengan osiloskop maka didapat data



Gambar 4. 13 Hasil konversi TCS3200 kondisi LED ON

bahwa TCS3200 memiliki frekuensi gelombang kotak. pada saat TCS3200 terkena



Gambar 4. 12 Hasil konversi TCS3200 kondisi LED OFF

cahaya LED maka frekuensinya akan berubah-ubah. Pada penelitian ini pada saat LED AC mati TCS3200 mengkonversi warna putih pada chasing AC.

Dari hasil data maka dapat disimpulkan ketika Frekuensi kurang dari 2924 maka kondisi AC dalam keadaan ON dan ketika Frekuensi lebih dari atau sama dengan 2924 maka AC dalam keadaan OFF.

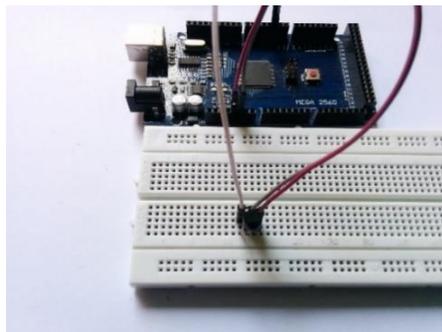
4.4 Pengujian Inframerah

Pengujian inframerah dilakukan untuk mendapatkan nilai *decode remote* AC menggunakan Inframerah *receiver* dan mencoba melakukan pengiriman hasil

decode inframerah *remote* menggunakan inframerah transmitter dengan arduino Mega 2560.

4.4.1 Inframerah Receiver

Pada penelitian ini inframerah receiver yang digunakan adalah tipe TSOP 4838 dengan tegangan masukan 5 volt. Untuk mendapatkan data hasil *decode remote* yang akurat maka hal yang harus diperhatikan adalah kondisi ruangan yang jauh dari noise frekuensi inframerah dan pengiriman frekuensi *remote* dilakukan pada tempat yang tertutup rapat sehingga seluruh frekuensi masuk pada inframerah *receiver*. Gambar 4.14 menunjukkan rangkaian *decode remote AC* dengan Inframerah *receiver* pada arduino Mega 2560, gambar 4.15 Inisialisasi IR *receiver* pada pemrograman Arduino IDE.



Gambar 4. 14 rangkaian *decode remote AC* dengan arduino mega 2560


```

DRY (26°C)
Raw: (439) 3452, 1696, 436, 420, 464, 1272, 432, 424, 460, 400, 464, 408, 440, 432, 440, 436, 436, 412, 432, 420, 436, 420, 460, 404,
432, 432, 432, 432, 456, 1296, 436, 436, 436, 416, 440, 408, 436, 428, 436, 420, 436, 424, 440, 432, 440, 1308, 460, 1292, 436, 1292,
440, 416, 436, 420, 460, 1272, 440, 424, 440, 428, 460, 412, 432, 440, 432, 424, 432, 420, 436, 420, 436, 428, 432, 436, 428, 432,
432, 436, 440, 432, 440, 416, 440, 408, 460, 400, 440, 416, 440, 424, 436, 440, 432, 432, 460, 420, 432, 412, 436, 420, 436, 424, 432,
432, 428, 436, 428, 440, 432, 440, 432, 440, 416, 436, 416, 440, 1288, 464, 1284, 436, 420, 432, 432, 432, 440, 436, 436, 436,
392, 436, 10124, 3452, 1692, 436, 424, 432, 1300, 436, 420, 432, 432, 432, 440, 432, 432, 436, 444, 436, 408, 440, 416, 436, 420, 436,
424, 440, 424, 440, 424, 464, 1292, 436, 436, 440, 408, 436, 416, 436, 420, 436, 428, 436, 428, 436, 428, 436, 428, 488, 1264, 436, 1312, 460,
1276, 460, 384, 464, 400, 436, 1300, 460, 400, 440, 424, 464, 412, 436, 436, 460, 396, 456, 396, 436, 420, 436, 420, 464, 400, 460,
412, 436, 432, 440, 432, 440, 408, 464, 1272, 464, 1276, 456, 396, 464, 1280, 440, 424, 440, 432, 436, 436, 436, 420, 436, 408, 464,
400, 436, 1304, 432, 432, 460, 1284, 436, 1308, 436, 440, 432, 420, 436, 420, 432, 420, 436, 428, 436, 428, 436, 428, 432, 440, 440,
432, 464, 1264, 460, 1264, 464, 400, 440, 1296, 436, 432, 432, 432, 460, 1292, 436, 436, 436, 1288, 464, 1264, 440, 428, 436, 416, 464,
1284, 432, 432, 436, 432, 440, 432, 440, 416, 440, 408, 460, 400, 440, 416, 440, 424, 436, 436, 436, 428, 468, 416, 436, 408, 436, 420,
464, 1272, 464, 1272, 488, 1256, 464, 404, 432, 436, 436, 436, 440, 416, 440, 408, 432, 428, 440, 416, 436, 428, 440, 432, 440, 1304, 488,
1268, 460, 1268, 460, 396, 432, 420, 460, 396, 488, 376, 488, 384, 436, 436, 436, 436, 436, 408, 464, 392, 464, 392, 464, 400, 460, 400,
440, 424, 464, 408, 440, 436, 464, 392, 484, 1240, 468, 396, 436, 420, 464, 1280, 436, 428, 440, 432, 440, 432, 464, 1264, 436, 416, 440,
416, 440, 424, 464, 400, 436, 428, 436, 440, 432, 440, 432, 420, 436, 420, 460, 396, 460, 404, 456, 404, 436, 428, 436, 436, 440, 432,
464, 1264, 440, 416, 464, 1272, 436, 416, 464, 1280, 440, 428, 464, 1288, 440, 1304, 440, 400, 436,
OFF
Raw: (440) 21636, 3456, 1692, 440, 416, 464, 1272, 436, 420, 460, 400, 464, 416, 432, 436, 436, 436, 436, 412, 432, 420, 436, 420, 464, 400,
432, 432, 432, 432, 460, 1292, 436, 436, 436, 416, 440, 408, 436, 428, 436, 416, 464, 400, 440, 436, 436, 1308, 464, 1292, 436, 1288, 440,
416, 464, 388, 492, 1244, 464, 404, 436, 428, 464, 408, 432, 440, 440, 416, 436, 416, 440, 416, 436, 428, 436, 420, 436, 424, 440, 432, 440,
432, 440, 416, 440, 408, 460, 404, 436, 416, 440, 424, 440, 432, 436, 432, 464, 416, 436, 412, 432, 420, 436, 420, 432, 436, 428, 432, 432,
428, 436, 436, 440, 432, 440, 416, 440, 416, 436, 1292, 460, 1284, 436, 424, 440, 424, 440, 436, 436, 436, 436, 388, 440, 10124, 3456, 1688,
436, 420, 436, 1300, 436, 420, 436, 424, 440, 432, 440, 424, 440, 444, 436, 408, 436, 416, 440, 416, 440, 416, 440, 424, 440, 428, 432, 432, 464, 1288,
436, 436, 436, 408, 440, 416, 440, 416, 436, 428, 436, 428, 436, 424, 464, 1288, 440, 1308, 464, 1272, 440, 408, 464, 400, 436, 1300, 436,
424, 440, 424, 440, 432, 440, 436, 436, 420, 432, 424, 432, 420, 436, 420, 464, 400, 460, 412, 436, 432, 440, 432, 440, 408, 436, 420, 460,
1272, 440, 424, 440, 1300, 440, 432, 440, 424, 436, 464, 392, 436, 416, 440, 416, 464, 1272, 436, 424, 472, 1276, 464, 1292, 436, 432,
440, 408, 436, 420, 436, 416, 464, 400, 440, 424, 436, 432, 464, 408, 436, 436, 460, 1272, 440, 1288, 440, 416, 464, 1276, 432, 428, 436,
436, 440, 1304, 440, 436, 464, 1264, 464, 1272, 436, 416, 440, 424, 440, 1296, 436, 436, 436, 432, 432, 440, 460, 392, 440, 416, 436, 420,
436, 424, 440, 424, 440, 424, 436, 436, 436, 436, 416, 440, 420, 436, 1292, 464, 1280, 436, 1296, 440, 428, 464, 408, 440, 432, 436,
420, 436, 416, 440, 416, 440, 424, 436, 428, 436, 428, 460, 1284, 464, 1288, 464, 1264, 436, 420, 436, 416, 440, 424, 440, 424, 440, 432, 432,
432, 440, 436, 432, 440, 416, 440, 416, 436, 420, 436, 424, 440, 424, 440, 424, 436, 440, 432, 440, 436, 416, 440, 1288, 440, 416, 436, 428,
436, 1300, 436, 432, 440, 428, 464, 416, 440, 1288, 436, 420, 436, 416, 440, 416, 464, 400, 464, 408, 436, 432, 440, 432, 412, 432, 424,
432, 420, 464, 400, 440, 424, 436, 428, 464, 408, 436, 432, 464, 1272, 440, 1292, 440, 416, 436, 424, 440, 1296, 440, 432, 440, 1304, 468,
1288, 440, 392, 432,

```

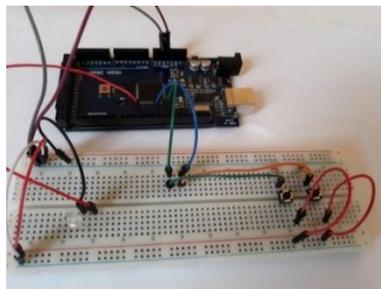
(b) Bagian kedua

Gambar 4. 18 Hasil konversi decode *remote* AC

Pada saat decode didapat nilai Raw, yang dimaksud “Raw” disini adalah frekuensi tersebut tidak diketahui *library IRremote* atau nilai kasaran dari suatu frekuensi yang didapat dan nilai pada Raw : (439) adalah jumlah data yang didapat.

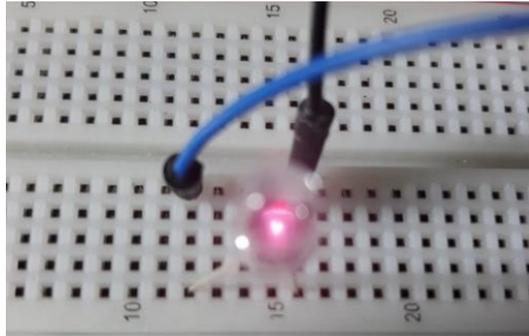
4.4.2 Inframerah Transmitter

Pada pengujian ini inframerah yang digunakan adalah inframerah 38 Khz dengan tegangan masukan 5 volt. Inframerah *transmitter* disini digunakan untuk mengirim sinyal inframerah hasil *decode* dari inframerah *receiver*. Pengujian dilakukan untuk mengetahui apakah sinyal yang didapat sudah dapat digunakan untuk menghidupkan dan mematikan *Air Conditioner*. Pada gambar 4.19 menunjukkan rangkaian *transmitter remote Air Conditioner* menggunakan tombol.



Gambar 4. 19 Rangkaian transmitter menggunakan tombol

Cahaya Sensor inframerah tidak dapat dilihat oleh mata manusia secara langsung untuk itu diperlukan perangkat yang dapat membantu melihat cahaya yang ditransmisikan sensor inframerah, dalam hal ini untuk melihat cahaya yang ditransmisikan Inframerah pada saat Inframerah menyala dapat dilihat menggunakan kamera *handphone*.



Gambar 4. 20 Kondisi infrared LED saat menyala

```

TESIRAC §
#include <IRremote.h>
//ON COOL
//OFF
IRsend irsend;

unsigned ON[] = {3516, 1680, 468, 396, 488, 1264, 464, 396, 468, 404, 468, 412, 464, 420, 464,
416, 464, 388, 464, 400, 464, 396, 468, 404, 468, 408, 464, 404, 492, 1280, 464, 416, 464, 388,
496, 368, 464, 408, 468, 396, 464, 404, 468, 412, 468, 1296, 496, 1272, 468, 1276, 468, 396, 464,
396, 468, 1288, 464, 404, 468, 404, 464, 420, 464, 416, 464, 396, 468, 396, 464, 396, 468, 408,
468, 404, 464, 404, 468, 412, 468, 416, 468, 396, 464, 388, 464, 408, 464, 396, 468, 404, 464, 420,
464, 404, 468, 420, 468, 388, 464, 396, 472, 396, 468, 404, 464, 404, 468, 404, 464, 420, 468, 412,
464, 396, 468, 396, 464, 1280, 492, 1268, 468, 404, 464, 408, 464, 416, 472, 408, 468, 368, 468,
10228, 3508, 1684, 464, 388, 496, 1256, 468, 404, 464, 412, 464, 408, 464, 412, 468, 412, 468, 396,
468, 396, 468, 396, 464, 400, 464, 404, 468, 412, 464, 1296, 468, 416, 468, 396, 464, 396, 468, 396,
464, 408, 464, 404, 468, 408, 492, 1264, 496, 1276, 496, 1248, 468, 396, 464, 396, 496, 1260, 464,
404, 468, 416, 468, 412, 468, 412, 464, 388, 468, 396, 468, 392, 472, 404, 464, 408, 464, 404, 468,
412, 472, 412, 464, 400, 488, 1256, 492, 1256, 468, 408, 464, 1284, 496, 1264, 496, 1276, 468, 416,
464, 388, 464, 400, 488, 1264, 464, 408, 464, 404, 468, 404, 496, 1264, 472, 412, 492, 372, 464, 396,
468, 396, 464, 408, 464, 408, 468, 404, 464, 416, 464, 416, 492, 1252, 496, 1256, 464, 1280, 492, 1264,
472, 1284, 468, 416, 464, 1296, 464, 416, 496, 1248, 464, 396, 468, 396, 496, 1256, 492, 1272, 464, 404,
468, 412, 468, 416, 464, 396, 468, 396, 464, 396, 468, 404, 472, 400, 472, 404, 464, 412, 468, 412, 472,
384, 496, 364, 496, 1260, 492, 1260, 496, 1264, 468, 404, 472, 412, 464, 416, 464, 396, 468, 396, 464,
396, 472, 392, 496, 380, 488, 388, 468, 1296, 492, 1276, 492, 1248, 472, 396, 464, 404, 460, 404, 468,
404, 464, 408, 468, 416, 464, 412, 468, 396, 464, 388, 496, 376, 468, 392, 472, 404, 468, 412, 464, 408,
496, 396, 464, 388, 492, 1260, 468, 396, 464, 408, 464, 1284, 468, 408, 488, 392, 464, 412, 496, 1260,
468, 384, 468, 404, 468, 396, 468, 404, 464, 416, 468, 400, 472, 420, 468, 388, 464, 400, 464, 396, 464,
408, 468, 400, 472, 404, 464, 416, 464, 416, 468, 392, 472, 1272, 472, 392, 496, 1264, 472, 400, 468, 1284,
496, 388, 468, 412, 492, 352, 468,
};
unsigned OFF[] = {3516, 1680, 468, 396, 464, 1288, 464, 400, 464, 404, 472, 412, 464, 408, 464, 424, 468, 384,
468, 400, 464, 396, 468, 404, 464, 408, 464, 404, 496, 1276, 468, 416, 464, 388, 464, 400, 464, 396, 464, 408,

```

(a) Bagian pertama

```

464, 404, 472, 404, 492, 1280, 464, 1292, 496, 1256, 472, 384, 496, 380, 464, 1284, 468, 404, 468, 404, 464,
416, 472, 412, 464, 396, 468, 396, 464, 396, 468, 396, 496, 376, 496, 384, 468, 412, 468, 412, 472, 380, 472,
396, 468, 396, 488, 380, 468, 404, 468, 400, 496, 388, 468, 412, 468, 396, 468, 392, 472, 392, 468, 404, 468,
404, 464, 408, 468, 416, 464, 412, 468, 396, 464, 388, 496, 1260, 492, 1260, 464, 408, 464, 412, 472, 412, 468,
412, 464, 372, 464, 10196, 3508, 1684, 464, 396, 472, 1272, 468, 404, 468, 404, 468, 404, 468, 412, 468, 412,
468, 392, 472, 396, 464, 400, 464, 404, 468, 404, 464, 408, 492, 1268, 472, 412, 492, 368, 468, 396, 464, 396,
468, 404, 468, 404, 468, 408, 488, 1276, 472, 1288, 496, 1252, 468, 392, 472, 396, 488, 1268, 472, 404, 464, 404,
468, 412, 472, 412, 464, 400, 464, 388, 464, 408, 464, 396, 472, 400, 468, 416, 464, 404, 468, 420, 472, 384, 468,
396, 492, 1256, 472, 396, 496, 1260, 496, 1268, 496, 1264, 472, 412, 464, 400, 464, 396, 464, 1288, 468, 396, 464,
408, 468, 412, 468, 1292, 468, 416, 464, 396, 464, 400, 464, 396, 472, 392, 468, 404, 468, 412, 468, 412, 468, 416,
464, 1276, 496, 1248, 496, 1256, 496, 1260, 488, 1272, 468, 404, 496, 1264, 496, 392, 472, 1272, 468, 396, 464, 396,
496, 1260, 492, 1268, 464, 408, 468, 408, 472, 412, 468, 396, 464, 388, 468, 404, 468, 392, 472, 404, 468, 412, 464,
408, 464, 424, 468, 388, 464, 400, 488, 1260, 468, 1284, 496, 1256, 496, 388, 468, 412, 468, 412, 464, 388, 496, 368,
468, 392, 496, 376, 472, 404, 464, 416, 464, 1296, 464, 1308, 464, 1280, 464, 396, 468, 396, 464, 396, 472, 400, 468,
416, 464, 416, 464, 412, 472, 384, 468, 392, 472, 392, 468, 408, 464, 404, 468, 404, 468, 416, 464, 416, 464, 396, 496,
1248, 468, 404, 464, 396, 496, 1268, 464, 408, 468, 416, 464, 412, 496, 1244, 472, 396, 464, 396, 468, 404, 472, 400, 468,
408, 464, 412, 468, 412, 472, 392, 468, 392, 472, 396, 464, 396, 496, 376, 496, 384, 468, 416, 464, 416, 464, 388, 496,
364, 472, 400, 472, 1280, 472, 400, 468, 1292, 472, 404, 468, 420, 468, 368, 468,
};
void setup() {
  // put your setup code here, to run once:
  pinMode(7,INPUT); // button 1
  pinMode(8,INPUT); // button 2
}

void loop() {
  //for (int i = 0; i< 3; i++){
  if (digitalRead(7) == HIGH){

    delay(50);
    irsend.sendRaw(ON, sizeof(ON) / sizeof(int), 38);
  }

  if (digitalRead(8) == HIGH){

    delay(50);
    irsend.sendRaw(OFF, sizeof(OFF) / sizeof(int), 38);
  }
  delay(1);
}

```

(b) Bagian kedua

Gambar 4. 21 Inisialisasi IR *transmitter* pada pemrograman Arduino IDE

4.5 Pembuatan tampilan web server

Proses pembuatan tampilan *web server* ini menggunakan HTML (*Hypert Text Markup Language*) dimana pada proses coding HTML ini yang biasanya didesain pada *web server*, langsung dimasukan pada Arduino melalui program Arduino IDE.

```

Desain_webserver

client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // do not forget this one
client.println("<!DOCTYPE html>");
client.println("<html lang='en'>");
client.println("<head>");
client.println("<title>Air Conditioner Controller webserver</title>");
client.println("<meta charset='utf-8'>");
client.println("<meta name='viewport' content='width=device-width; initial-scale=1.0; maximum-scale=1.0;'>");
client.println("<style>{ box-sizing: border-box;}");

client.println("body { font-family: Arial, Helvetica, sans-serif; background: black;}");

/* Style the header */
client.println("<.header { height:60px; background: black; padding: 3px; text-align: center; font-size: 14px; font-family: Times New Roman; color: white;}");

/* Create three equal columns that floats next to each other */

client.println("<.box{ width:250px; height:100px; background:#205081; padding: 15px; border-radius: 10px; display: inline-block; margin-left: 18%;-webkit-box-shadow: inset 0px 0px 120px -33px");
client.println("<.box2{ width:250px; height:100px; background:#205081; padding: 15px; border-radius: 10px; display: inline-block; margin-left: 50px;-webkit-box-shadow: inset 0px 0px 120px -33px");
client.println("<.box3{ width:500px; height:100px; background:#205081; padding: 15px; border-radius: 10px;display: inline-block; margin-left: 18%;-webkit-box-shadow: inset 0px 0px 120px -33px ;");

client.println("<.button {background-color: #4CAF50; /* Green */border: none;color: white;padding: 16px 32px;text-align: center;text-decoration: none;display: inline-block;font-size: 16px;-webkit-tz");
client.println("<.button1 {background-color: #205081; color: black;margin-left: 238;}");

client.println("<.button1:hover {background-color: #008CBA; color: white;}");

client.println("<.button2 {background-color: #205081; color: black; }");

client.println("<.button2:hover {background-color: #008080;color: white;}");

client.println("<.button3 {background-color: #205081; color: black; }");

client.println("<.button3:hover {background-color: #444336;;color: white;}");

/* Clear floats after the columns */
client.println("<.row:after {content: '';display: table;clear: both;}");

```

(a) Bagian Pertama

```

/* Style the footer */
client.println("<.footer {padding: 4px;}");

/* Responsive layout - makes the three columns stack on top of each other instead of next to each other */
client.println("<@media (max-width: 600px) {.column {width: 100%;}}");
client.println("</style>");
client.println("</head>");
client.println("<body>");

client.println("<div class='header'>");
client.println("<h2>Air Conditioner Controller webserver</h2>");
client.println("</div>");
client.println("<br>");
client.println("<div class='row'>");
client.println("<div class='box'>");
client.println("<div");
if(tempC > 25){
    client.println("<style='font-size: 30px; color: #930000; text-align: center; font-family: Times New Roman;'>");
}
else if(tempC <=25 && tempC >=5){
    client.println("<style='font-size: 30px; color: #00FF00; text-align: center; font-family: Times New Roman;'>");
}
else if(tempC <5){
    client.println("<style='font-size: 30px; color: #009191; text-align: center; font-family: Times New Roman;'>");
}
client.println("</div>");
client.println(tempC);
client.println("</b>");
client.println("<sup style='font-size: 25px'>C</sup>");
client.println("</div>");
client.println("<p style='text-align: center; font-family: Times New Roman; font-size: 18px; color: black'>");
client.println("<b>Celcius</b></p>");
client.println("</div>");

client.println("<div class='box2'> ");
client.println("<div");
if(tempF >= 77){
    client.println("<style='font-size: 30px; color: #930000; text-align: center; font-family: Times New Roman;'>");
}

```

(b) Bagian Kedua

```

else if(tempF <77 && tempC >=41){
  client.println("style='font-size: 30px; color: #00FF00; text-align: center; font-family: Times New Roman;'");
}
else if(tempF <41){
  client.println("style='font-size: 30px; color: #009191; text-align: center; font-family: Times New Roman;'");
}

client.println("<b>");
client.println(tempF);
client.println("</b>");
client.println("  <sup style='font-size: 25px;'>F</sup>");
client.println("</div>");
client.println("  <p style='text-align: center; font-family: Times New Roman; font-size: 18px; color: black;'>");
client.println("    <b>Fahrenheit</b></p>");
client.println("</div>");
client.println("<br>");
client.println("  <div class='row'>");
client.println("    <br>");
client.println("    <div class='box3'> ");
client.println("    <div");
client.println("      style='font-size: 30px; color: white; text-align: center; font-family: Times New Roman;'>");
client.println("<br>");
if(greenFrequency < 2520){
  client.println("<div style='color: white;'>");
  client.println("ON");
}
else
{
  client.println("<div style='color: black;'>");
  client.println("OFF");
}
client.println("</b>");
client.println("  </div>");
client.println("  <p style='text-align: center; font-family: Times New Roman; font-size: 18px; color: black;'>");
client.println("    <b>Status AC</b></p>");
client.println("</div>");
client.println("<br> ");
client.println("<br> ");
client.println("  <div class='row'> ");

```

(c) Bagian Ketiga

```

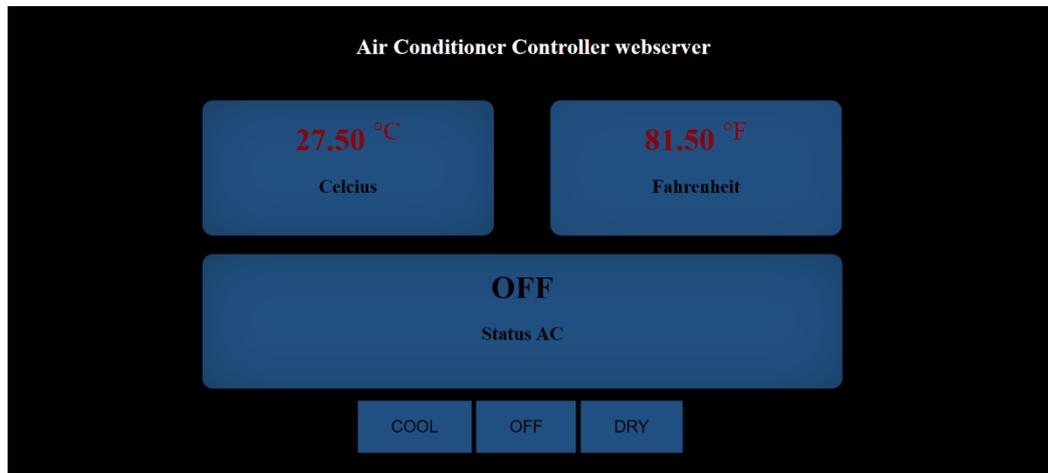
}
client.println("</b>");
client.println("  </div>");
client.println("  <p style='text-align: center; font-family: Times New Roman; font-size: 18px; color: black;'>");
client.println("    <b>Status AC</b></p>");
client.println("</div>");
client.println("<br> ");
client.println("<br> ");
client.println("  <div class='row'> ");
client.println("    <a href='/COOL' class='button button1'>COOL</a>");
client.println("    <a href='/OFF' class='button button2'>OFF</a>");
client.println("    <a href='/DRY' class='button button3'>DRY</a>");
if (request.indexOF("COOL") > 0) {
  irsend.sendRaw(COOL, sizeof(COOL) / sizeof(int), 38);
}
if (request.indexOF("OFF") > 0) {
  irsend.sendRaw(OFF, sizeof(OFF) / sizeof(int), 38);
}
if (request.indexOF("DRY") > 0) {
  irsend.sendRaw(DRY, sizeof(DRY) / sizeof(int), 38);
}
client.println("</div>");
client.println("</section>");
client.println("  </div>");
client.println("  <br><br><br><br><br><br>");
client.println("<br>");
client.println("  <footer class='footer'>");
client.println("    <div style='font-size: 12px; font-family: Arial; color: white;'>");
client.println("      <strong>Copyright copyy 2018-2022 <a href='https://www.instagram.com/martadhidendra/'>Armed</a></strong> All rights reserved.");
client.println("      <span style='float:right'><b>Credit</b> @b Waradhi Dendra Permata </span>");
client.println("    </div>");
client.println("  </div>");
client.println("  </div>");
client.println("</body>");
client.println("</html>");

```

(d) Bagian Keempat

Gambar 4. 22 Inisialisasi HTML Web Server pada Arduino IDE

Pada pembuatan tampilan ini didapatkan hasil dimana untuk melihat tampilan keseluruhan *Web Server* membutuhkan waktu 10 detik.



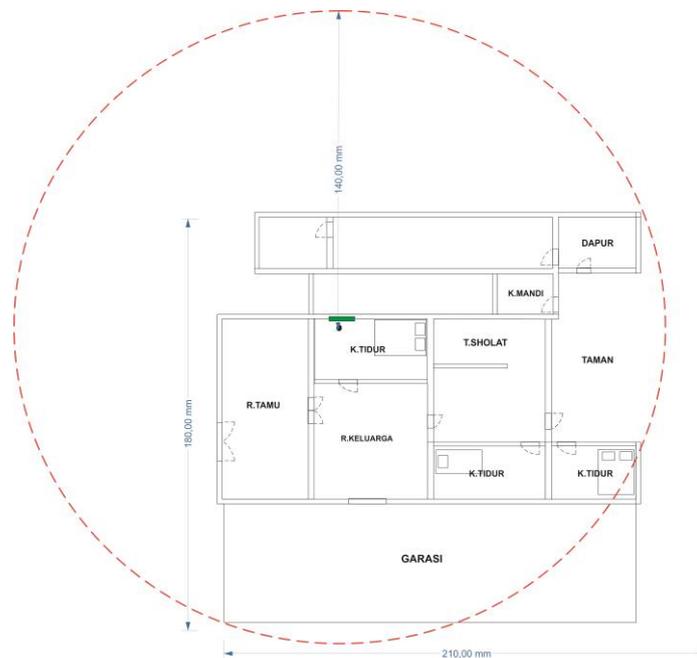
Gambar 4. 23 Tampilan keseluruhan hasil sensor pada *web browser (Off)*



Gambar 4. 24 Tampilan keseluruhan hasil sensor pada *web browser (On)*

4.6 Pengujian jangkauan alat

Pada pengujian ini adalah pengujian *hotspot* dari *handphone* yang akan diterima oleh modul ESP8266-01. Jangkauan jarak pancar ini dipengaruhi oleh lokasi pengambilan data, lokasi pengambilan dilakukan didalam ruangan yang dipenuhi dengan sekat-sekat dinding bata sesuai dengan penempatan perangkat yang akan digunakan untuk kontrol *Air conditioner*. Gambar 4.25 menunjukkan skema jangkauan wifi.

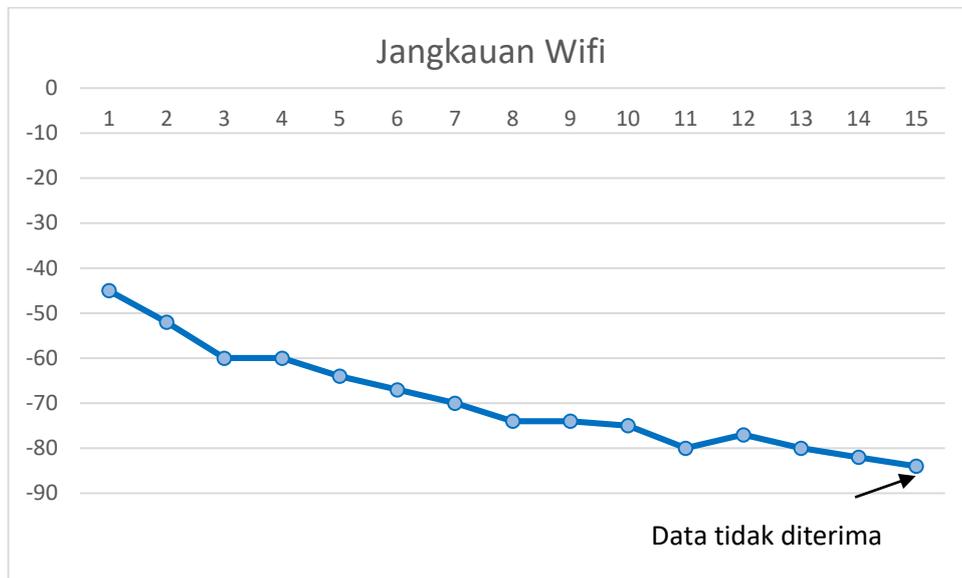


Gambar 4. 25 Skema jangkauan wifi

Dari skema diatas didapat hasil seperti pada tabel 4.3 dibawah, dimana jarak maksimal wi-fi yang dapat ditangkap ESP8266-01 adalah 14 meter. Sehingga, apabila jarak penerima sinyal lebih dari 15 meter dari titik *access point* maka klient tidak dapat terhubung dengan *access point*.

Tabel 4. 3 Hasil jangkauan jarak wifi dan kekuatan wifi

No	Jarak (meter)	Kekuatan Wifi (dbm)	Data kirim	Sinyal
1	1	-45	Data diterima	4
2	2	-52	Data diterima	4
3	3	-60	Data diterima	3
4	4	-60	Data diterima	3
5	5	-64	Data diterima	3
6	6	-67	Data diterima	3
7	7	-70	Data diterima	2
8	8	-74	Data diterima	2
9	9	-74	Data diterima	2
10	10	-75	Data diterima	2
11	11	-80	Data diterima	1
12	12	-77	Data diterima	2
13	13	-80	Data diterima	1
14	14	-82	Data diterima	1
15	15	-84	Data tidak diterima	0



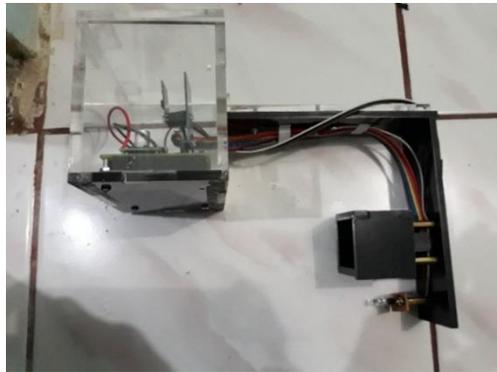
Gambar 4. 26 Grafik jarak dan kekuatan wifi

4.6 Pengujian Kinerja Alat secara Keseluruhan

Dari penelitian ini bahwa kinerja setiap perangkat keras dan perangkat lunak sangat baik. Pada sensor DS18B20 dapat mendeteksi suhu ruangan dengan *error* rata-rata 0.04% , untuk sensor TCS3200 dapat mengkonversi warna sensor pada indikator LED AC dan selanjutnya frekuensi tersebut digunakan untuk menentukan pada *web server* apakah AC tersebut ON atau OFF dan Inframerah LED dapat bekerja untuk mematikan dan mengaktifkan AC pada kondisi COOL maupun DRY. Untuk modul ESP8266 dapat bekerja dengan baik saat mengirim dan menerima data dari sensor menuju *web server*. Dalam akses *web server* membutuhkan waktu sekitar 10 detik untuk menampilkan keseluruhan data. Gambar 4.27 menunjukkan perangkat keras keseluruhan.



(a) Perangkat keseluruhan saat digunakan pada AC



(a) Perangkat keseluruhan saat tidak menempel pada AC

Gambar 4. 27 Perangkat keras keseluruhan

