

# **PROTOTIPE REMOTE CONTROL AC BERBASIS IOT**

## **TUGAS AKHIR**



**Disusun Oleh :**

**FARIS BANU ANWARI**  
**5223165335**

**MUCH GILANG RAMADHAN**  
**5223165439**

Sebagai salah satu syarat untuk memperoleh gelar Ahli Madya (A.Md)  
pada Program Studi Diploma III Teknik Elektronika

**PROGRAM STUDI DIPLOMA III TEKNIK ELEKTRONIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI JAKARTA  
2020**

## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Peneliti ini menyatakan dengan sesungguhnya bahwa Tugas Akhir dengan judul :

### **PROTOTIPE REMOTE CONTROL AC BERBASIS IOT**

Dirancang untuk memenuhi salah satu syarat kelulusan Program Studi Diploma Tiga Teknik Elektronika Fakultas Teknik Universitas Negeri Jakarta. Dengan ini kami menyatakan bahwa :

1. Karya tulis akhir ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik D3 (Diploma Tiga), Baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis ini adalah gagasan murni, rumusan dan penelitian kami berdua dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain kecuali, secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini kami buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini kami bersedia menerima sanksi. Berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, 31 Januari 2020  
Yang Membuat Pernyataan



Faris Banu Anwari  
(5223165335)

## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Peneliti ini menyatakan dengan sesungguhnya bahwa Tugas Akhir dengan judul :  
**PROTOTIPE REMOTE CONTROL AC BERBASIS IOT**

Dirancang untuk memenuhi salah satu syarat kelulusan Program Studi Diploma Tiga Teknik Elektronika Fakultas Teknik Universitas Negeri Jakarta. Dengan ini kami menyatakan bahwa :

1. Karya tulis akhir ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik D3 (Diploma Tiga), Baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis ini adalah gagasan murni, rumusan dan penelitian kami berdua dengan arahan dosen pebiimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain kecuali, secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini kami buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini kami bersedia menerima sanksi. Berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, 31 Januari 2020  
Yang Membuat Pernyataan



Much Gilang Ramadhan

(5223165439)

## LEMBAR PENGESAHAN

NAMA DOSEN

TANDA TANGAN

TANGGAL

Koordinator Program Studi

Syufrijal, ST., MT.  
NIP : 19760327 200112 1 001



13/2/2020

Dosen Pembimbing

Dr. Muhammad Rif'an, MT.  
NIP : 19741022 200112 1 001



14/2/2020

Dosen Ahli

Syufrijal, ST., MT.  
NIP : 19760327 200112 1 001



13/2/2020

Dosen Pengaji

Drs. Rimulyo Wicaksono, MM.  
NIP : 19631001 198811 1 001



14/2/2020

Heri Firmansyah, ST., MT.  
NIP : 19840214 201903 1 011



14/02/2020

## KATA PENGANTAR

Dengan menyebut nama Allah SWT yang maha pengasih lagi maha penyayang, kami panjatkan puji syukur kehadirat-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada kami, sehingga kami dapat menyelesaikan tugas akhir yang berjudul “PROTOTIPE REMOTE CONTROL AC BERBASIS IOT”.

Penelitian tugas akhir dilakukan untuk memenuhi salah satu syarat kelulusan dan mendapat gelar Ahli Madya Teknik Elektronika Universitas Negeri Jakarta. Tanpa bantuan dan bimbingan dari beberapa pihak, sangat sulit bagi kami untuk menyelesaikan penelitian tugas akhir. Oleh karena itu, kami mengucapkan terima kasih kepada:

1. Syufrijal S.T, M.T selaku Koordinator Program Studi Diploma Tiga Teknik Elektronika.
2. Dr. Muhammad Rif'an, MT. selaku dosen pembimbing atas segala kesabaran, ketulusan, motivasi, dan bimbingannya untuk menyelesaikan penelitian.
3. Kedua orang tua dan keluarga kami yang telah memberikan semangat dan doa yang tidak pernah terhenti diucapkan.
4. Rekan-rekan mahasiswa/i Diploma Tiga Teknik Elektronika angkatan 2016 yang telah membantu dan memberi dukungan dalam penelitian tugas akhir.

Akhir kata, semoga Allah Subhanahu Wa Ta’ala membalas segala kebaikan semua pihak yang telah membantu penelitian tugas akhir ini. Semoga tugas akhir membawa manfaat yang besar bagi pengembangan ilmu pengetahuan dan teknologi.

Jakarta, 31 Januari 2020

Peneliti

## **ABSTRAK**

Pada pembuatan tugas akhir bertujuan merancang dan mengembangkan sistem prototipe remote remote ac berbasis IOT. Tujuan pembuatan alat ini adalah agar memudahkan kita dalam kehidupan sehari-hari. Sistem ini memanfaatkan sensor infrared untuk memancarkan kode-kode ke AC tertentu sama halnya seperti remote AC pada umumnya tetapi bisa dikendalikan dari jarak jauh. Alat ini juga menampilkan kodisi suhu sebenarnya di ruangan di mana tempat alat ini digunakan melalui smartphone maupun melihat langsung dari alat tersebut. NodeMCU ESP8266 sebagai mikrokontroler. Sensor DHT11 untuk mendeteksi suhu ruangan setiap 2 detik. Hasil dari sensor DHT11 ditampilkan pada aplikasi Blynk dan OLED SSD1306, dibutuhkan koneksi internet untuk mengirimkan data yang diperoleh ESP8266 ke aplikasi Blynk. Hasil pengujian menunjukan bahwa setiap komponen berfungsi dengan baik, persentase error pada sensor DHT11 dalam mendeteksi suhu  $\pm 0.01\%$  dan dalam mendeteksi kelembaban  $\pm 7.6\%$ .

**Kata kunci :** Remote AC, Infrared, NodeMCU ESP8266, Blynk

## DAFTAR ISI

<b>PERNYATAAN KEASLIAN TUGAS AKHIR.....</b>	Error! Bookmark not defined.
<b>LEMBAR PENGESAHAN .....</b>	Error! Bookmark not defined.
<b>KATA PENGANTAR .....</b>	v
<b>ABSTRAK .....</b>	vi
<b>DAFTAR ISI.....</b>	vii
<b>DAFTAR GAMBAR.....</b>	ix
<b>DAFTAR TABEL.....</b>	x
<b>BAB I PENDAHULUAN.....</b>	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah .....	2
1.3 Perumusan Masalah.....	2
1.4 Pembatasan Masalah .....	2
1.5 Tujuan Penelitian.....	3
1.6 Manfaat Penelitian.....	3
<b>BAB II LANDASAN TEORI .....</b>	4
2.1 Internet of Things .....	4
2.1.1 Cara Kerja .....	4
2.2 Mikrokontroler .....	5
2.3 Blynk .....	6
2.3.1 Sketch Program Blynk .....	7
2.4 Sensor Infra Merah .....	8
2.4.1 Sketch Tester Receiver Infrared.....	9
2.4.2 Simbol Infrared .....	11
2.4.3 Skematik Diagram Infrared.....	12
2.5 NodeMCU ESP8266 .....	12
2.5.1 Spesifikasi .....	13
2.6 Sensor Humidity DHT11.....	14
2.6.1 Sketch Tester DHT11 .....	15
2.6.2 Skematik Diagram DHT11 .....	16
2.7 OLED SSD1306 .....	17

2.7 Software .....	18
2.7.1 Arduino IDE.....	18
<b>BAB III PEMBAHASAN .....</b>	<b>20</b>
3.1 Deskripsi Alat .....	20
3.2 Analisis Kebutuhan Alat.....	20
3.3 Blok Diagram Alat .....	21
3.4 Flowchart Sistem .....	22
3.5 Perancangan Perangkat Keras .....	23
3.5.1 Perancangan Sistem Mekanik.....	23
3.5.2 Perancangan Sistem Elektrik .....	24
3.5.3 Perancangan WIFI .....	26
3.6 Deskripsi Kerja Alat.....	26
3.7 Perancangan Perangkat Lunak .....	27
3.7.1 Arduino IDE .....	27
3.7.2 Blynk .....	28
3.8 Pengujian Alat .....	29
3.8.1 Pengujian Data Setiap Komponen .....	29
3.8.2 Pengujian Serial Monitor Pada Board Mikrokontroler.....	32
3.8.3 Pengujian WIFI .....	34
<b>BAB IV KESIMPULAN DAN SARAN.....</b>	<b>36</b>
4.1 Kesimpulan.....	36
4.2 Saran .....	36
<b>DAFTAR PUSTAKA .....</b>	<b>37</b>
<b>LAMPIRAN – LAMPIRAN.....</b>	<b>39</b>
Lampiran 1. Program Utama .....	40
Lampiran 2. Library Infrared ESP8266.....	49
Lampiran 3. Program Penerima Infrared ESP8266.....	51

## DAFTAR GAMBAR

Gambar 2.1 Blynk .....	6
Gambar 2.2 Halaman utama Blynk .....	7
Gambar 2.3 Infrared .....	9
Gambar 2.4 Infrared Transmitter .....	11
Gambar 2.5 Photodioda Sebagai Penerima Inframerah .....	11
Gambar 2.6 Skematik Infrared Transmitter .....	12
Gambar 2.7 Skematik Photodioda.....	11
Gambar 2.8 NodeMCU ESP8266 .....	13
Gambar 2.9 DHT11.....	14
Gambar 2.10 Skematik DHT11.....	16
Gambar 2.11 OLED SSD1306.....	17
Gambar 2.12 Tampilan Software Arduino IDE .....	18
Gambar 3.1 Blok Diagram Alat .....	20
Gambar 3.2 Flowchart Alat.....	21
Gambar 3.3 Desain Depan .....	22
Gambar 3.4 Skematik Diagram Sistem ESP8266 .....	23
Gambar 3.5 Skematik Diagram ESP8266 Penerima Inframerah .....	24
Gambar 3.6 Wiring Diagram Sistem Utama ESP8266 .....	24
Gambar 3.7 Wiring Diagram Sistem ESP8266 Penerima Inframerah.....	25
Gambar 3.8 Tampilan awal Arduino IDE .....	27
Gambar 3.9 Tampilan <i>Log In</i> Blynk .....	28
Gambar 3.10 Data Pertama Hasil Perbandingan Sensor DHT11 .....	30
Gambar 3.11 Data Kedua Hasil Perbandingan Sensor DHT11 .....	30
Gambar 3.12 Data Ketiga Hasil Perbandingan Sensor DHT11 .....	31
Gambar 3.13 Serial Monitor Pada ESP8266 Program Utama .....	32
Gambar 3.14 Serial Monitor Pada ESP8266 Penerima Inframerah.....	33
Gambar 3.15 Kode Menghubungkan Dan Mensetting Nama WIFI .....	33
Gambar 3.16 Serial Monitor Pada ESP8266 Penerima WIFI .....	34
Gambar 3.17 Tampilan Blynk Pada Smartphone Saat Terhubung ESP8266 .....	34

## **DAFTAR TABEL**

Tabel 3.1 Peralatan yang dibutuhkan .....	19
Tabel 3.2 Bahan yang dibutuhkan.....	20
Tabel 3.3 Pengujian Sensor DHT11 .....	29

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Saat ini IOT (Internet Of Things) banyak dipergunakan untuk banyak aplikasi - aplikasi tertentu seperti untuk memantau CCTV jalan raya yang bisa diakses oleh khalayak publik. Perkembangan IOT telah banyak berkontribusi untuk memudahkan memantau maupun mengendalikan alat dari jarak yang cukup jauh bahkan sangat jauh.

Menurut C. Wangetal (2013) dalam jurnal Gunawan Hendro Cahyono, dari semua kegiatan yang ada dalam IOT adalah untuk mengumpulkan data mentah yang benar dengan cara yang efisien; tapi yang lebih penting adalah untuk menganalisis dan mengolah data mentah menjadi informasi yang lebih berharga.

Melihat kondisi tersebut maka dalam tugas akhir ini ingin menciptakan suatu alat pengatur suhu otomatis bagi perusahaan agar lebih mudah mengatur suhu darimanapun, dengan begitu bisa mengefisienkan waktu dan menimilisir kerusakan barang-barang yang tidak tahan suhu panas. Alat ini juga dapat dijadikan pengatur suhu rumah bagi yang memiliki peliharaan agar selalu dalam suhu yang cukup nyaman maupun menghindari kerusakan barang ataupun obat – obat yang tidak boleh terkena suhu tinggi.

Alat ini menggunakan *platform* blynk sebagai penghubung dengan alat yang bisa diakses kapan saja memakai internet melalui *smartphone*.

## 1.2 Identifikasi Masalah

Permasalahan yang peneliti ajukan dapat diidentifikasi permasalahannya sebagai berikut:

1. Bagaimana cara menghubungkan remote ir ke blynk?
2. Bagaimana cara mengetahui kode-kode khusus pada remote AC?
3. Bagaimana cara mengetahui suhu yang diatur oleh remote AC?

## 1.3 Perumusan Masalah

Berdasarkan dari latar belakang masalah diatas, didapatkan beberapa rumusan masalah, yaitu:

1. Bagaimana cara mengetahui apakah AC sudah menyala saat dinyalakan oleh remote melalui blynk?
2. Apakah remote ini akan sesuai dengan apa yang diperintahkan oleh blynk?

## 1.4 Pembatasan Masalah

Agar penelitian dapat dilakukan lebih fokus dan mendalam, maka Peneliti memandang permasalahan penelitian yang diangkat perlu dibatasi variabelnya. Adapun Batasan masalah dalam tugas akhir, yaitu:

1. Alat ini baru bisa untuk AC tertentu
2. Merancang dan medesain tempat yang pas agar bisa dideteksi oleh AC
3. Mengetahui suhu real dan suhu yang diatur melalui remote AC
4. Pembangunan aplikasi IOT menggunakan platfrom Blynk

## **1.5 Tujuan Penelitian**

Untuk memperjelas arah dan tujuan dari perancangan, peneliti merumuskan tujuannya, sebagai berikut:

1. Memahami cara kerja sistem Remote Control AC Berbasis IOT menggunakan modul NodeMCU V3 ESP8266 *Microcontroller*.
2. Membantu mempermudah pengguna AC dalam mengatur suhu saat sedang tidak di dekat ruangan.

## **1.6 Manfaat Penelitian**

Bisa mengatur suhu bagi perusahaan atau pengguna agar lebih mudah darimanapun, dengan begitu bisa mengefisienkan waktu dan menimilisir kerusakan barang-barang yang tidak tahan suhu tinggi.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Internet of Things**

Menurut Yasha (2018), Internet of Things (IOT) adalah suatu konsep dimana objek tertentu punya kemampuan untuk mentransfer data lewat jaringan tanpa memerlukan adanya interaksi dari manusia ke manusia maupun dari manusia ke perangkat komputer. IOT ini juga kerap diidentifikasi dengan sensor sebagai metode komunikasi informasi. Walaupun begitu, IOT bisa mencakup teknologi-teknologi lainnya, semacam teknologi nirkabel maupun kode QR yang sering kita temukan.

Menurut Ashton (2009), definisi awal IOT adalah memiliki potensi untuk mengubah dunia seperti pernah dilakukan oleh Internet, bahkan mungkin lebih baik.

Internet of Things (IOT) menjadi minat utama sebagai hasil pengembangan teknologi dan revolusi industri 4.0. Aplikasi IOT telah banyak diimplementasikan di setiap sektor seperti sistem keamanan, industri, pertanian, dan kedokteran. Beberapa penelitian telah mengembangkan rumah pintar berbasis IOT seperti sistem keamanan rumah menggunakan internet.

Alat ini menggunakan *platform blynk* sebagai penghubung dengan alat yang bisa diakses kapan saja memakai internet melalui *smartphone*.

#### **2.1.1 Cara Kerja**

Menurut Alexandromeo (2017), salah satu contoh dari cara kerja Internet of Things ini seperti lupa mengunci pintu rumah, sedangkan

pemilik rumah tersebut berada diluar rumah. Logika untuk Internet of Things dalam mengunci pintu :

1. Buka aplikasi yang mendukung IOT seperti Blynk.
2. Diterima perangkat IOT.
3. Perangkat IOT memberikan perintah untuk modul solenoid doorlock/servo.
4. Kunci bergerak.
5. Pintu terkunci.

## **2.2 Mikrokontroler**

Menurut Yuliana (2011), mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya adalah membaca dan menulis data.

Sedangkan menurut Ramdhani (2012) mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori dan input/output. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus secara khusus.

Menurut Bejo (2008), Sistem yang menggunakan mikrokontroler sering disebut sebagai embedded system atau dedicated system. Embedded system adalah sistem pengendali yang tertanam pada suatu produk, sedangkan dedicated system adalah sistem pengendali yang dimaksudkan hanya untuk suatu fungsi tertentu. Sebagai contoh, printer adalah suatu embedded system karena di dalamnya terdapat

mikrokontroler sebagai pengendali dan juga dedicated system karena fungsi pengendali tersebut berfungsi hanya untuk menerima data dan mencetaknya. Hal ini berbeda dengan suatu PC yang dapat digunakan untuk berbagai macam keperluan, sehingga mikroprosesor pada PC sering disebut sebagai general purpose microprocessor (mikroprosesor serba guna). Pada PC berbagai macam software yang disimpan pada media penyimpanan dapat dijalankan, tidak seperti mikrokontroler hanya terdapat satu software aplikasi.

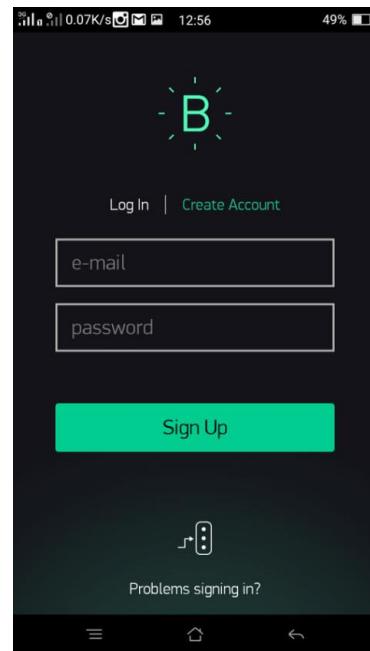
### **2.3 Blynk**

Menurut Agus (2018), Blynk adalah platform untuk aplikasi OS Mobile yang bertujuan untuk mengendalikan modul mikrokontroler seperti Arduini, Raspberry Pi, Wemos dan modul sejenisnya melalui jaringan internet.

BLYNK adalah platform untuk aplikasi OS Mobile (iOS dan Android) yang bertujuan untuk kendali module Arduino, Raspberry Pi, ESP8266, WEMOS D1, dan module sejenisnya melalui Internet. Aplikasi ini merupakan wadah kreatifitas untuk membuat antarmuka grafis untuk proyek yang akan diimplementasikan hanya dengan metode drag and drop widget. Blynk tidak terikat pada papan atau module tertentu. Dari platform aplikasi inilah dapat mengontrol apapun dari jarak jauh, dimanapun kita berada dan waktu kapanpun. Dengan catatan terhubung dengan internet dengan koneksi yang stabil dan inilah yang dinamakan dengan sistem Internet of Things (IOT). Berikut adalah tampilan blynk yang dapat dilihat pada gambar 2.1 dan 2.2



Gambar 2.1 Blynk



Gambar 2.2 Halaman utama Blynk

Aplikasi ini merupakan wadah kreatifas untuk membuat antarmuka grafis untuk proyek yang akan diimplementasikan hanya dengan metode drag and drop widget.

### 2.3.1 Sketch Program Blynk

```
#define BLYNK_PRINT Serial // Enables Serial
Monitor
```

```

// Following includes are for Arduino Ethernet
Shield (W5100)
// If you're using another shield, see Boards_*
examples
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";
void setup()
{
    // See the connection status in Serial Monitor
    Serial.begin(9600);
    // Here your Arduino connects to the Blynk
    Cloud.
    Blynk.begin(auth);
}

void loop()
{
    // All the Blynk Magic happens here...
    Blynk.run();

    // You can inject your own code or combine it
    with other sketches.
    // Check other examples on how to communicate
    with Blynk. Remember
    // to avoid delay() function! }

```

## 2.4 Sensor Infra Merah

Sinar Inframerah adalah sinar electromagnet yang panjang gelombangnya lebih daripada cahaya nampak yaitu di antara 700 nm dan 1 mm. Sinar infra merah merupakan cahaya yang tidak tampak. Jika dilihat dengan spektroskop cahaya maka radiasi cahaya inframerah akan nampak pada spectrum elektromagnet dengan panjang gelombang di atas panjang gelombang cahaya merah. Kita bisa memanfaatkan infrared untuk memancarkan sinyal tertentu.

Dalam sensor Infrared, ada komponen elektronika yang dapat mendeteksi benda ketika cahaya infra merah terhalangi oleh benda. Sensor infared terdiri dari

led infrared sebagai pemancar dan fototransistor sebagai penerima cahaya infra merah.

Proses pemancaran cahaya akibat adanya energi listrik yang diberikan terhadap suatu bahan disebut dengan sifat elektroluminesensi. (Sutrisno.1987).

Berikut adalah contoh infrared yang dapat dilihat pada gambar 2.3



Gambar 2.3 Infrared

#### 2.4.1 Sketch Tester Receiver Infrared

```
#include <Arduino.h>
#include <IRrecv.h>
#include <IRremoteESP8266.h>
#include <IRac.h>
#include <IRtext.h>
#include <IRutils.h>

const uint16_t kRecvPin = 14;
const uint32_t kBaudRate = 115200;
const uint16_t kCaptureBufferSize = 1024;
#if DECODE_AC
const uint8_t kTimeout = 50;
#else // DECODE_AC
const uint8_t kTimeout = 15;
#endif // DECODE_AC
const uint16_t kMinUnknownSize = 12;
#define LEGACY_TIMING_INFO false
IRrecv irrecv(kRecvPin, kCaptureBufferSize,
kTimeout, true);
```

```

decode_results results; // Somewhere to store the
results

// This section of code runs only once at start-up.
void setup() {
#if defined(ESP8266)
    Serial.begin(kBaudRate, SERIAL_8N1,
SERIAL_TX_ONLY);
#else // ESP8266
    Serial.begin(kBaudRate, SERIAL_8N1);
#endif // ESP8266
    while (!Serial) // Wait for the serial connection
to be established.
    delay(50);
    Serial.printf("\n" D_STR_IRRECVDUMP_STARTUP "\n",
kRecvPin);
#if DECODE_HASH
    // Ignore messages with less than minimum on or
off pulses.
    irrecv.setUnknownThreshold(kMinUnknownSize);
#endif // DECODE_HASH
    irrecv.enableIRIn(); // Start the receiver
}

// The repeating section of the code
void loop() {
    // Check if the IR code has been received.
    if (irrecv.decode(&results)) {
        // Display a crude timestamp.
        uint32_t now = millis();
        Serial.printf(D_STR_TIMESTAMP " : %06u.%03u\n",
now / 1000, now % 1000);
        // Check if we got an IR message that was too
big for our capture buffer.
        if (results.overflow)
            Serial.printf(D_WARN_BUFFERFULL "\n",
kCaptureBufferSize);
        // Display the library version the message was
captured with.
        Serial.println(D_STR_LIBRARY " : " v"
_IRREMOTEESP8266_VERSION "\n");
        // Display the basic output of what we found.

        Serial.print(resultToHumanReadableBasic(&results));
        // Display any extra A/C info if we have it.
        String description =
IRAcUtils::resultAcToString(&results);
        if (description.length())
Serial.println(D_STR_MSGDESC ": " + description);
    }
}

```

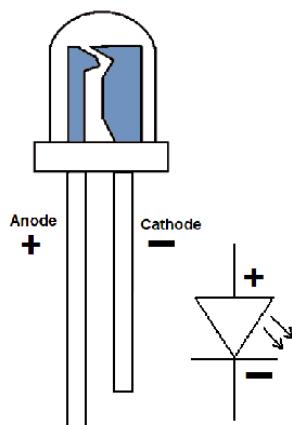
```

        yield(); // Feed the WDT as the text output
can take a while to print.
#if LEGACY_TIMING_INFO
    // Output legacy RAW timing info of the result.
    Serial.println(resultToTimingInfo(&results));
    yield(); // Feed the WDT (again)
#endif // LEGACY_TIMING_INFO
    // Output the results as source code
    Serial.println(resultToSourceCode(&results));
    Serial.println(); // Blank line between
entries
    yield(); // Feed the WDT (again)
}
}

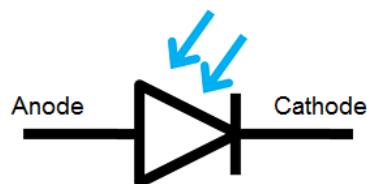
```

#### 2.4.2 Simbol Infrared

Berikut adalah symbol infrared transmitter dan receiver yang dapat dilihat pada gambar 2.4 dan 2.5



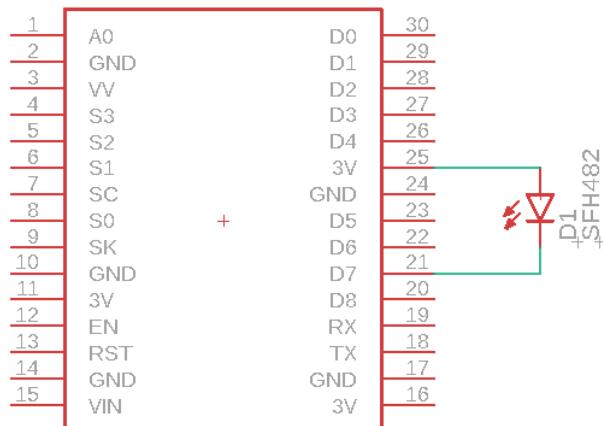
Gambar 2.4 Infrared transmitter



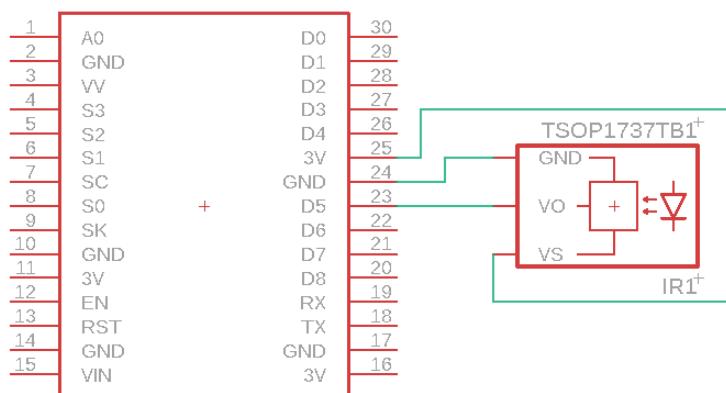
Gambar 2.5 Infrared receiver sebagai penerima inframerah

### 2.4.3 Skematik Diagram Infrared

Berikut adalah skematik diagram infrared yang dapat dilihat pada gambar 2.6 dan 2.7



Gambar 2.6 Skematik Infrared Transmitter

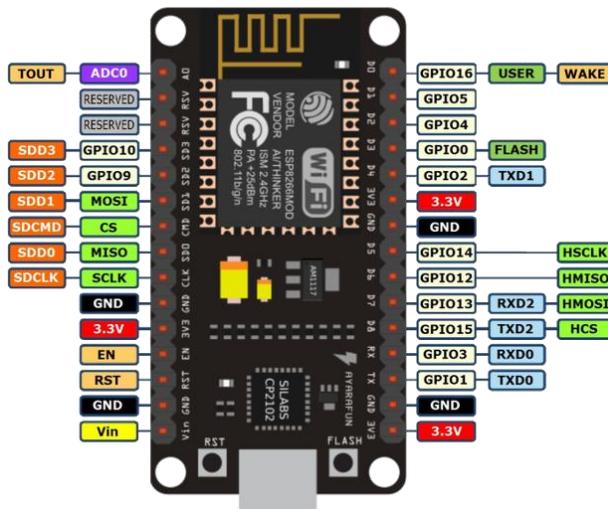


Gambar 2.7 Skematik Photodioda

### 2.5 NodeMCU ESP8266

Menurut Widiyaman (2016), NodeMCU ESP8266 merupakan modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti Arduino agar dapat terhubung langsung dengan wifi dan membuat koneksi TCP/IP. Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode wifi yaitu Station,

Access Point dan Both (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis **ESP8266** yang kita gunakan. Sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler. Firmware default yang digunakan oleh perangkat ini menggunakan AT Command, selain itu ada beberapa Firmware SDK yang digunakan oleh perangkat ini berbasis opensource. Berikut adalah contoh NODEMCU ESP8266 dapat dilihat pada gambar 2.8



Gambar 2.8 NodeMCU ESP8266

### 2.5.1 Spesifikasi

NodeMCU (ESP8266) ini memiliki spesifikasi sebagai berikut :

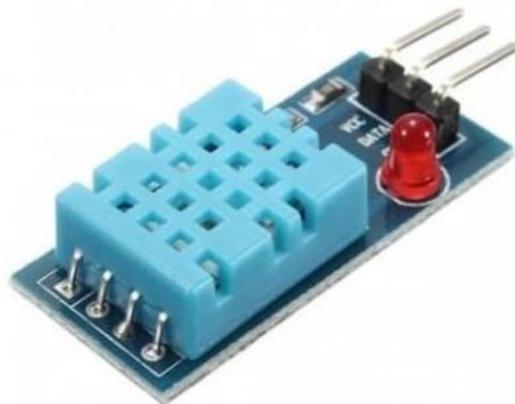
1. Mikrokontroler : ESP8266
2. Ukuran Board : 57mm x 30mm
3. Tegangan Input : 3.3V ~ 5V
4. GPIO : 13 PIN
5. Kanal PWM : 10 Kanal

6.	10 bit ADC Pin	: 1 Pin
7.	Flash Memori	: 4MB
8.	Clock Speed	: 40/26/24 MHz
9.	WiFi	: IEEE 802.11 b/g/n
10.	Frekuensi	: 2.4 GHz – 22.5 GHz
11.	USB Port	: Micro USB
12.	Card Reader	: Tidak Ada
13.	USB to Serial Converter	: CH340G

## 2.6 Sensor Humidity DHT11

Water Menurut Baskara (2013), merupakan konverter yang diprogram untuk mengubah warna menjadi frekuensi yang tersusun atas konfigurasi silicon photodiode dan konverter arus ke frekuensi dalam IC CMOS.

Sensor DHT11 merupakan sensor dengan kalibrasi sinyal digital yang mampu memberikan informasi suhu dan kelembaban. Sensor ini tergolong komponen yang memiliki tingkat stabilitas yang sangat baik, apalagi digandeng dengan kemampuan mikrokontroler. DHT11 memiliki fitur kalibrasi yang sangat akurat. Koefisien kalibrasi ini disimpan dalam OTP program memory, sehingga ketika internal sensor mendeteksi sesuatu suhu atau kelembaban, maka module ini membaca koefisien sensor tersebut. Ukurannya yang kecil, dengan transmisi sinyal hingga 20 meter, membuat produk ini cocok digunakan untuk banyak aplikasi-aplikasi pengukuran suhu dan kelembaban. Berikut adalah contoh DHT11 dapat dilihat pada gambar 2.9.



Gambar 2.9 DHT11

### 2.6.1 Sketch Tester DHT11

```
#include "DHT.h"
#define DHTPIN 14      // Digital pin connected to
the DHT sensor
#define DHTTYPE DHT11    // DHT 11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));

  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about
  250 milliseconds!
  // Sensor readings may also be up to 2 seconds
  'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit
= true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to
try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
```

```

        Serial.println(F("Failed to read from DHT
sensor!"));
        return;
    }

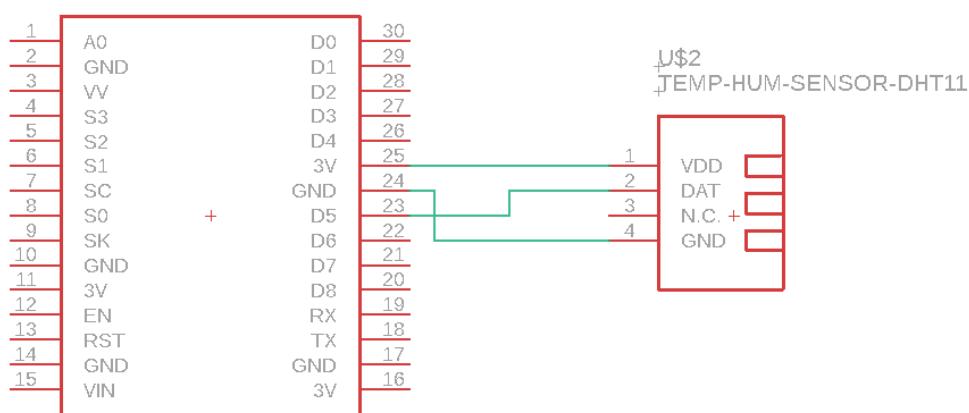
    // Compute heat index in Fahrenheit (the
    default)
    float hif = dht.computeHeatIndex(f, h);
    // Compute heat index in Celsius (isFahreheit =
    false)
    float hic = dht.computeHeatIndex(t, h, false);

    Serial.print(F("Humidity: "));
    Serial.print(h);
    Serial.print(F("% Temperature: "));
    Serial.print(t);
    Serial.print(F("°C "));
    Serial.print(f);
    Serial.print(F("°F Heat index: "));
    Serial.print(hic);
    Serial.print(F("°C "));
    Serial.print(hif);
    Serial.println(F("°F"));
}

```

## 2.6.2 Skematik Diagram DHT11

Berikut adalah contoh skematik diagram DHT11 yang dapat dilihat pada gambar 2.10.



Gambar 2.10 Skematik Diagram DHT11

## 2.7 OLED SSD1306

OLED SSD1306 adalah driver CMOS OLED / PLED chip tunggal dengan *controller* untuk *system* tampilan grafis dot-matrix *light emitting diode*. Terdiri dari 128 *segment* dan 64 *commons*. IC OLED SSD1306 dirancang untuk panel OLED tipe *Common Cathode*.

OLED SSD1306 memiliki fitur sebagai berikut :

1. Layar LCD memiliki lebih banyak fungsi.
2. Kontras tinggi, sehingga tidak membutuhkan cahaya latar.
3. Tegangan kerja : 2.7V – 5.5V; Ukuran PCB: 2,8 cm x 3,2 cm.
4. Papan sirkuit cetak dua sisi standar, tebal 1, 16mm, lubang 3-mm di dua sudut untuk memudahkan pemasangan.
5. Konsumsi daya rendah 0,04W selama operasi normal.

Berikut adalah contoh OLED SSD1366 yang dapat dilihat pada gambar 2.11



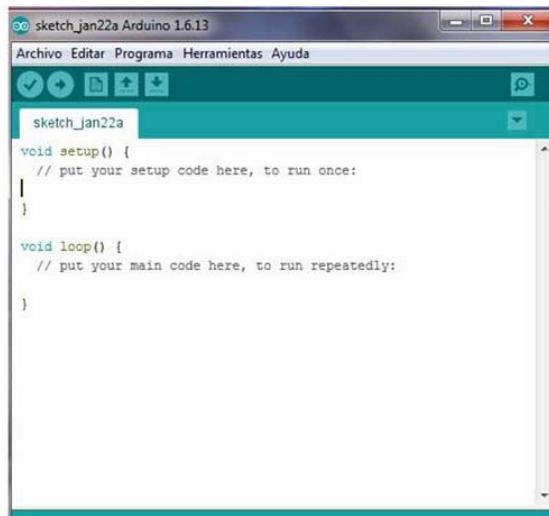
Gambar 2.11 OLED SSD1306

## 2.7 Software

Pengertian software adalah suatu perintah program dalam sebuah komputer yang apabila dieksekusi oleh usernya akan memberikan fungsi dan unjuk kerja seperti yang diharapkan oleh user-nya. Dengan kata lain, perangkat lunak berfungsi untuk memberi perintah kepada komputer agar dapat berfungsi secara optimal sesuai dengan perintah user.

### 2.7.1 Arduino IDE

Arduino IDE (Integrated Development Environment) adalah *software* pemrograman board mikrokontroler untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. *Software* ini menggunakan Bahasa pemrograman sendiri yang meyerupai Bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam pemrograman dari bahasa aslinya. Arduino IDE dibuat dari Bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino. Berikut adalah tampilan software Arduino yang dapat dilihat pada gambar 2.12.



```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Gambar 2.12 Tampilan Software Arduino IDE

## **BAB III**

### **PEMBAHASAN**

#### **3.1 Deskripsi Alat**

Alat pengatur suhu dan kelembaban di buat untuk mengetahui suhu ruang setiap 2 detik, alat ini juga bias memerintahkan dari jarak jauh melalui blynk, dibutuhkan koneksi internet untuk mengirimkan data yang diperoleh ke aplikasi Blynk. Terdapat sebuah sensor yaitu sensor DHT11. Sensor dan OLED dipasang pada tempat dan di tempelkan dekat penerima infrared AC, Digunakan *powersupply* dengan tegangan 3.7V untuk *supply*, kemudian diletakan di AC

#### **3.2 Analisis Kebutuhan Alat**

Alat dan bahan yang dibutuhkan dalam penggerjaan Prototipe Remote AC Berbasis IOT dapat dilihat pada Tabel 3.1 berikut:

Tabel 3.1 Peralatan yang dibutuhkan

No	Nama Peralatan	Jumlah
1	Solder	1
2	Penyedot Timah	1
3	Tang jepit	1
4	Multimeter	1
5	Cutter	1
6	Obeng plus	1
7	Bor tangan	1
8	Bor beton	1

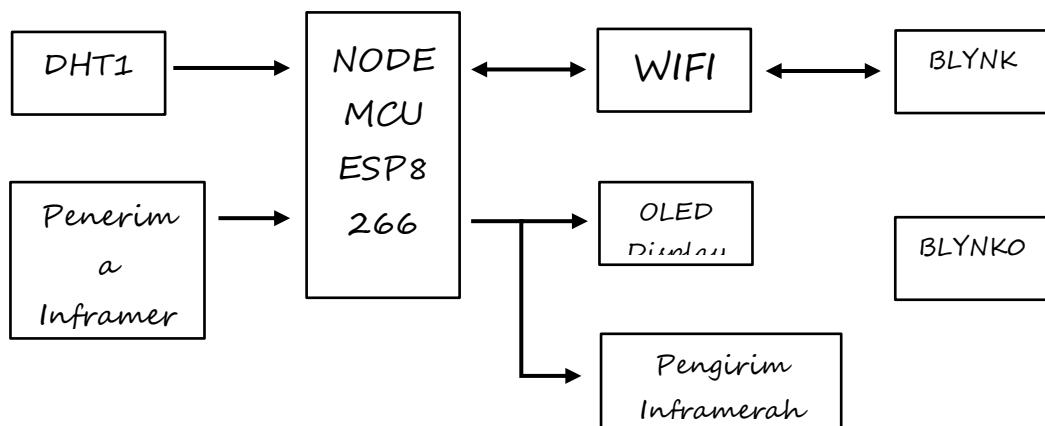
Sedangkan untuk bahan yang dibutuhkan dalam penggerjaan alat dapat dilihat pada Tabel 3.2 berikut.

Tabel 3.2 Bahan yang dibutuhkan

No	Nama Bahan	Jumlah
1	<i>Casing</i>	1
2	NodeMCU ESP 8266	2
3	<i>Infrared Transmitter</i>	1
4	<i>Infrared Receiver</i>	1
5	Kabel jumper	25
6	DHT11	1
7	Smartphone + Blynk	1
8	OLED SSD1366	1
9	Remote AC asli	1

### 3.3 Blok Diagram

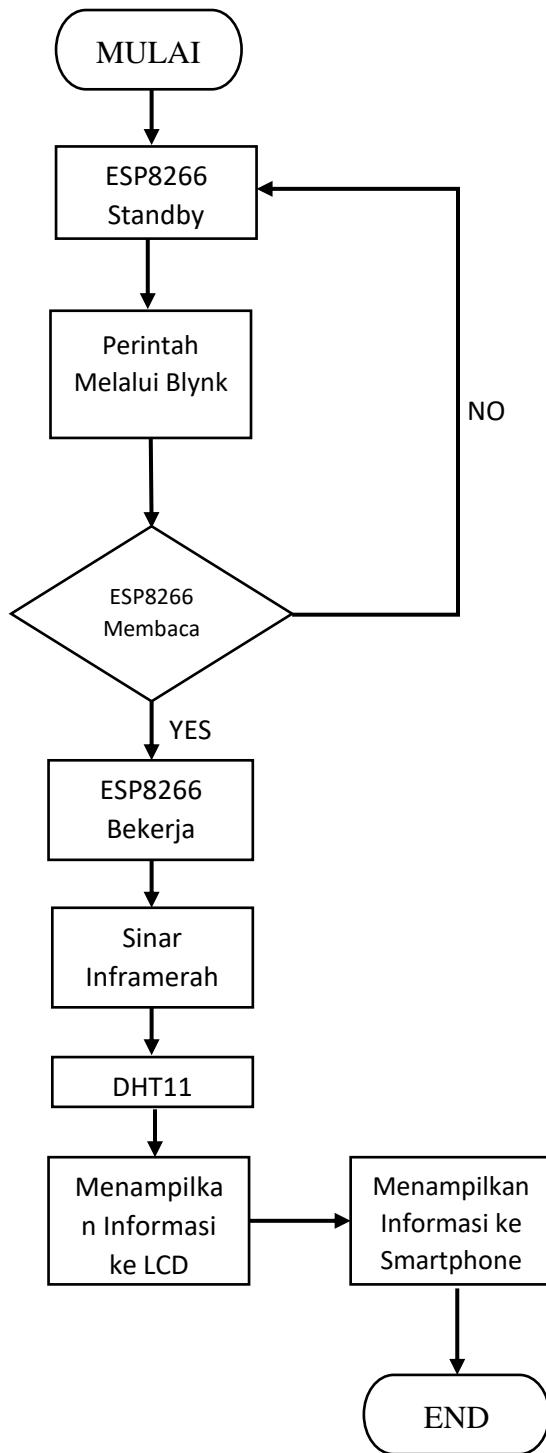
Blok diagram sistem dari Prototipe Remote Control AC Berbasis IOT dapat dilihat pada Gambar 3.1 berikut.



Gambar 3.1 Blok Diagram Alat

### 3.4 Flowchart Sistem

Flowchart sistem dari Prototipe Remote Control AC Berbasis IOT dapat dilihat pada Gambar 3.2 berikut.



Gambar 3.2 Flowchart

### 3.5 Perancangan Perangkat Keras

Perancangan perangkat keras dibagi 2 bagian yaitu perancangan sistem mekanik dan perancangan sistem elektrik.

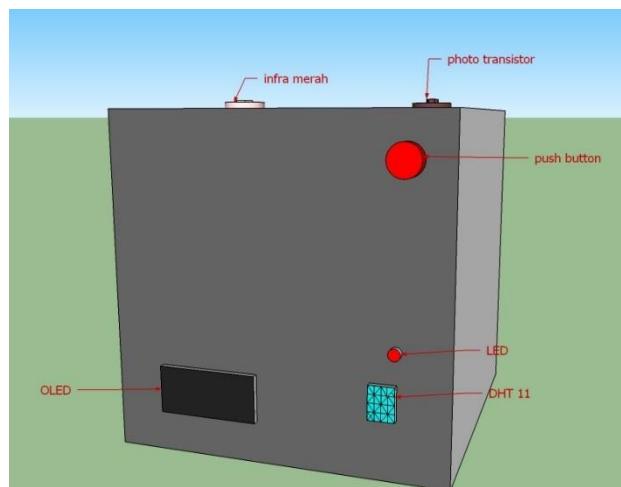
#### 3.5.1 Perancangan Sistem Mekanik

Perancangan mekanik merupakan tahap utama dalam pembuatan alat tugas akhir. Karena pada tahap perancangan mekanik, akan menentukan bagaimana mekanisme cara kerja alat, mulai dari desain kerangka alat dan juga penempatan setiap komponen alat.

Berikut merupakan tahapan perancangan mekanik :

1. Desain alat menggunakan aplikasi Sketch Up

Pada desain alat menggunakan aplikasi Sketch Up. Contoh desain dari Prototipe Remote Control AC Berbasis IOT dapat dilihat pada Gambar 3.3 berikut.



Gambar 3.3 Desain depan

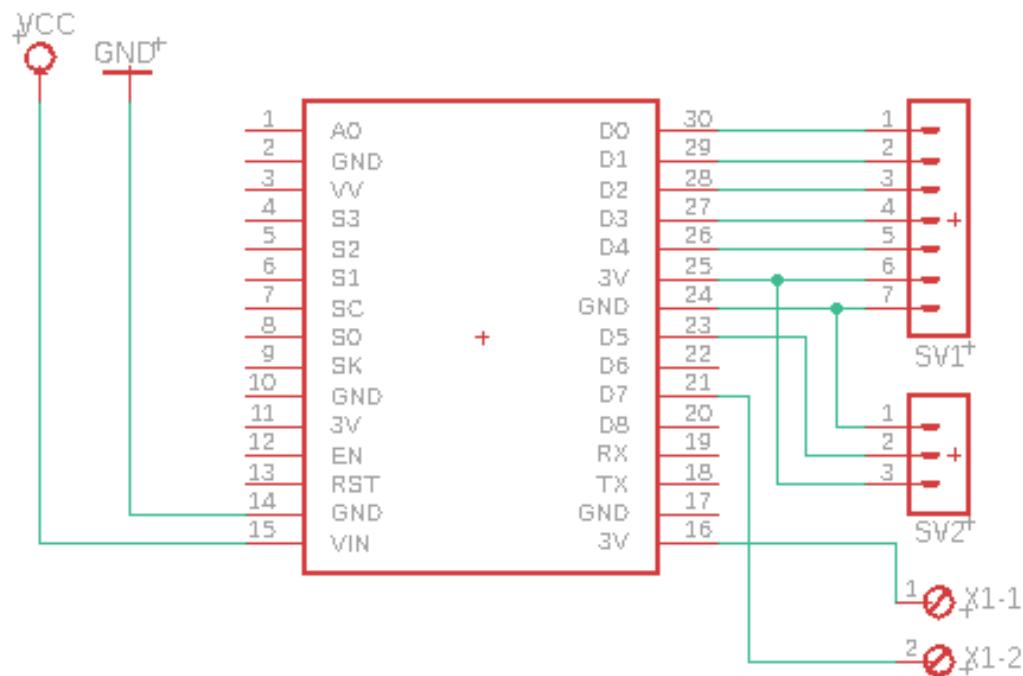
### 3.5.2 Perancangan Sistem Elektrik

Perancangan sistem elektrik merupakan salah satu bagian penting pada pembuatan alat. Penempatan komponen-komponen yang tepat dapat menambah efisiensi dan mengurangi biaya yang dibutuhkan. Membuat perancangan sistem elektrik sangat penting dalam perancangan kerangka pada alat.

Berikut merupakan desain perancangan elektrik pada alat :

#### 1. Skematik Diagram Sistem ESP8266

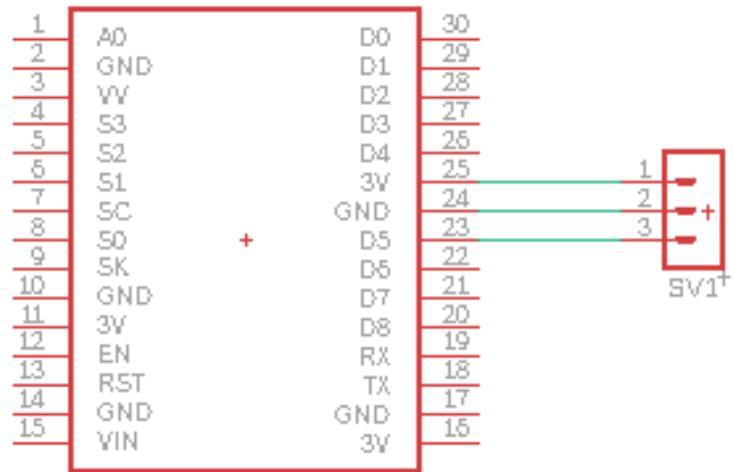
Skematik Diagram Sistem ESP8266 untuk program utama ditunjukan pada gambar 3.4



Gambar 3.4.Skematik Diagram Sistem ESP8266

## 2. Skematik Diagram Sistem ESP8266 Penerima Infrared

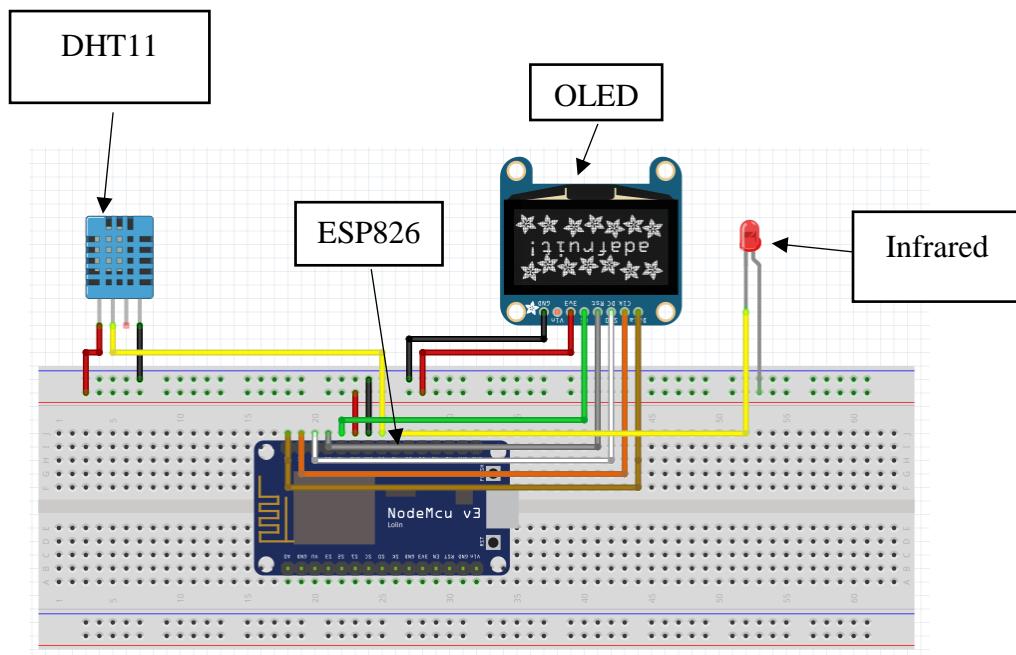
Skematik Diagram Sistem ESP8266 untuk penerima inframerah ditunjukan pada gambar 3.5



Gambar 3.5 Skematik Diagram ESP8266 Penerima Inframerah

## 3. Wiring Diagram Sistem ESP8266

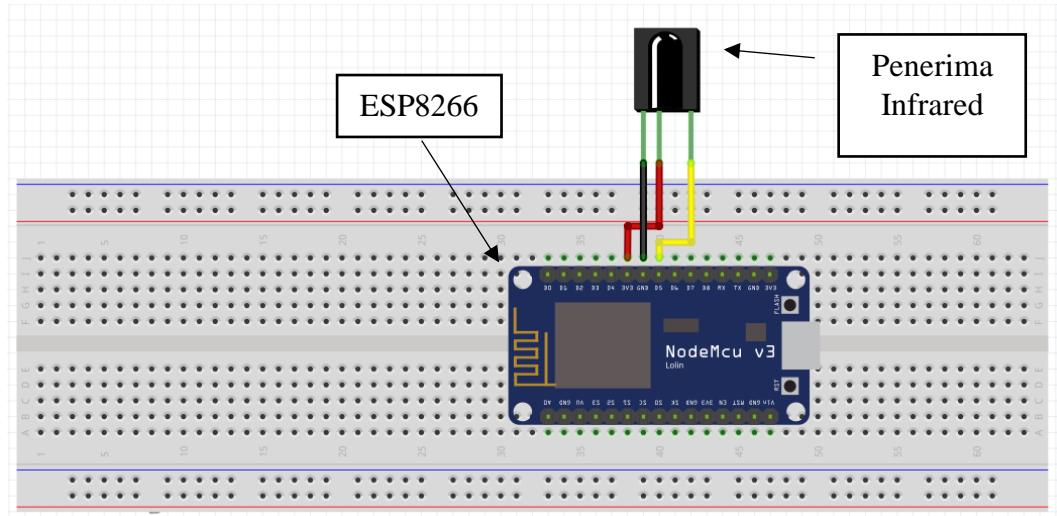
Wiring Diagram Sistem ESP8266 ditunjukan pada gambar 3.6



Gambar 3.6 Wiring Diagram Sistem Utama ESP8266

#### 4. Wiring Diagram Sistem ESP8266

Wiring Diagram Sistem ESP8266 ditunjukan pada gambar 3.6



Gambar 3.7 Wiring Diagram Sistem ESP8266 Penerima Inframerah

#### 3.5.3 Perancangan WIFI

Perancangan WIFI merupakan salah satu bagian penting pada pembuatan alat. Penempatan pemancar WIFI yang tepat dapat menimbulkan error karena tidak terkirimnya data ke ESP8266 maupun ke *smartphone*. Membuat perancangan system WIFI sangat penting dalam perancangan kerangka pada alat.

Dalam tugas akhir ini menggunakan *smartphone* sebagai router karena bersifat mobile dan memudahkan saat diuji di manapun dan kapanpun.

#### 3.6 Deskripsi Kerja Alat

Awal kerja alat yaitu, alat harus terhubung dengan internet, jika tidak, alat tidak akan bekerja sesuai apa yang diperintahkan. Saat menekan salah satu *icon* di blynk, blynk akan mengirimkan data ke NodeMCU sesuai pilihan dari aplikasi

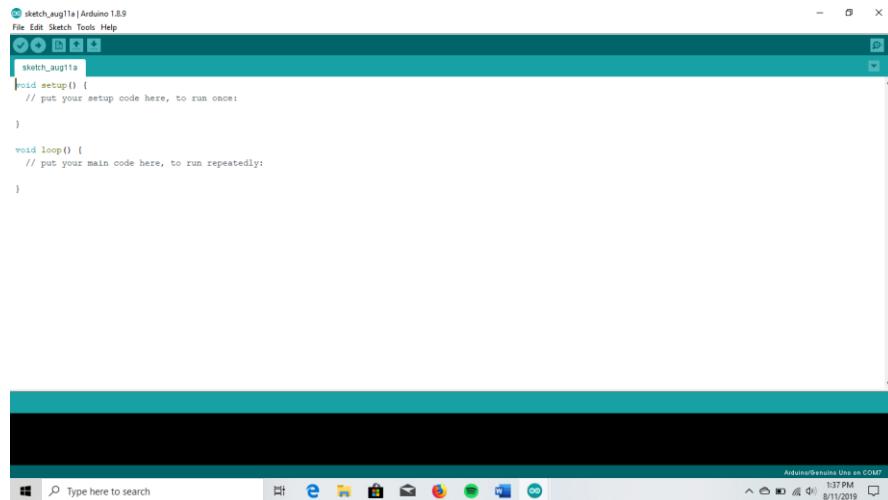
blynk di *smartphone*. NodeMCU akan memprosesnya lalu mengirimkan kode ke AC melalui *IR Transmitter* yang sesuai dengan icon yang di sentuh. DHT11 akan mendeteksi suhu yang ada diruangan tempat alat ini ditempatkan. Selanjutnya NodeMCU akan mendeteksi suhu dan kelembaban ruangan dan mengirimkannya ke blynk agar bisa ditampilkan suhu terkini dari ruangan tempat alat ini ditempatkan.

### **3.7 Perancangan Perangkat Lunak**

Perangkat lunak (*software*) yang di gunakan dalam pembuatan alat Remote AC berbasis IOT ini menggunakan 1 *software*, yaitu Arduino IDE (*Integrated Development Environment*). Arduino IDE digunakan untuk memprogram NodeMCU ESP8266 untuk menjalankan DHT11, OLED SSD1366, *IR Receiver* serta *IR Transmitter* dan mengirimkan data – data yang diambil oleh sensor ke aplikasi Blynk secara *wireless* berbasis *Internet of Things* (IOT).

#### **3.7.1 Arduino IDE**

Alat remote AC berbasis IOT menggunakan *software* Arduino IDE (*Integrated Development Environment*) untuk memprogram 3 sensor dan 1 layar LED. Arduino IDE digunakan dalam Prototipe Remote AC Berbasis IOT untuk memprogram NodeMCU ESP8266 yang digunakan sebagai mikrokontroler untuk sistem *monitoring* dan mengatur alat berbasis *Internet of Things* (IOT). Berikut adalah tampilan awal Arduino IDE pada Gambar 3.8.

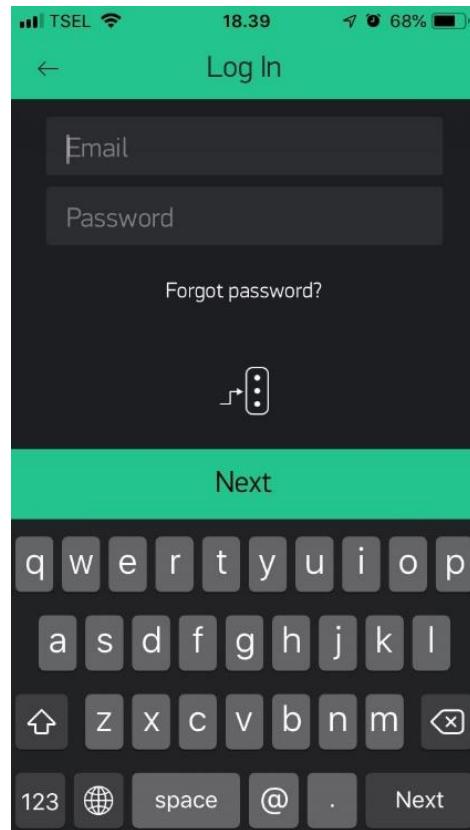


Gambar 3.8 Tampilan awal Arduino IDE

### 3.7.2 Blynk

Blynk merupakan sebuah aplikasi pada *smartphone* yang dapat menampilkan *interface* suatu alat atau proyek untuk memantau kelangsungan dari alat atau proyek tersebut. Pada tugas akhir ini, Blynk digunakan untuk memantau suhu ruangan dan mengatur suhu AC. Berikut adalah langkah – langkah perancangan pada aplikasi Blynk sebagai *platform* untuk menampilkan *interface* alat berbasis IoT.

Langkah awal dalam merancang *interface* Protitpe Remote AC Berbasis IOT pada aplikasi Blynk yaitu dengan mengunduh aplikasi Blynk pada *smartphone*. Selanjutnya *log in* jika sudah memiliki akun atau *create new account* untuk membuat akun baru seperti pada Gambar 3.9.



Gambar 3.9 Tampilan *Log In* Blynk

Setelah *log in*, buat *project* baru dan tuliskan nama *project* yang akan dibuat, pilih *device* yang digunakan pada alat dan pilih tema yang ingin diterapkan pada *interface* proyek. Dalam Prototipe Robot Lengan 3 DOF, *device* yang digunakan adalah ESP8266

### 3.8 Pengujian Alat

Pengujian alat dibagi menjadi 2 yaitu, pengujian data setiap data hasil komponen dan pengujian serial monitor setiap board mikrokontroler.

#### 3.8.1 Pengujian Data Setiap Komponen

Komponen-komponen yang di uji untuk pengambilan data adalah sensor DHT11

## 1. Pengujian Pada Sensor DHT11

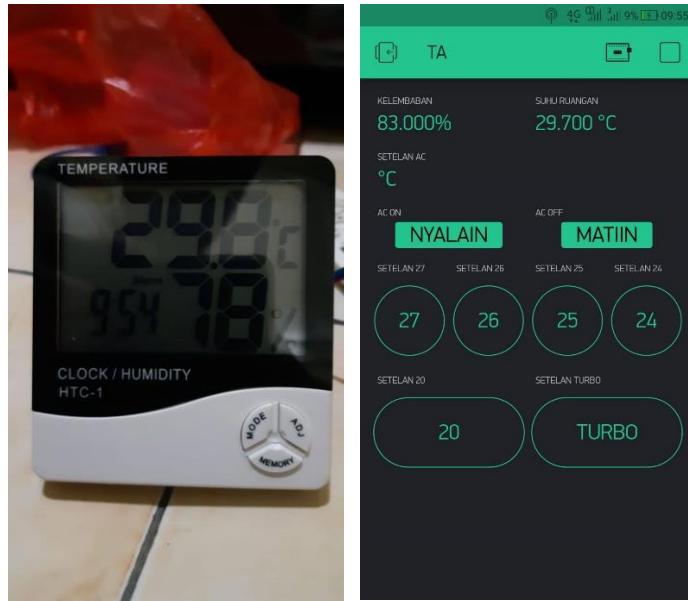
Pada pengujian sensor DHT11 ini dilakukan juga perbandingan data hasil dari sensor DHT11 dengan sensor suhu dan kelembaban ruangan, berikut data hasil pengujian ditunjukan pada gambar 3.8 – 3.10 dan pada tabel 3.1.

Tabel 3.1 Pengujian Sensor DHT11

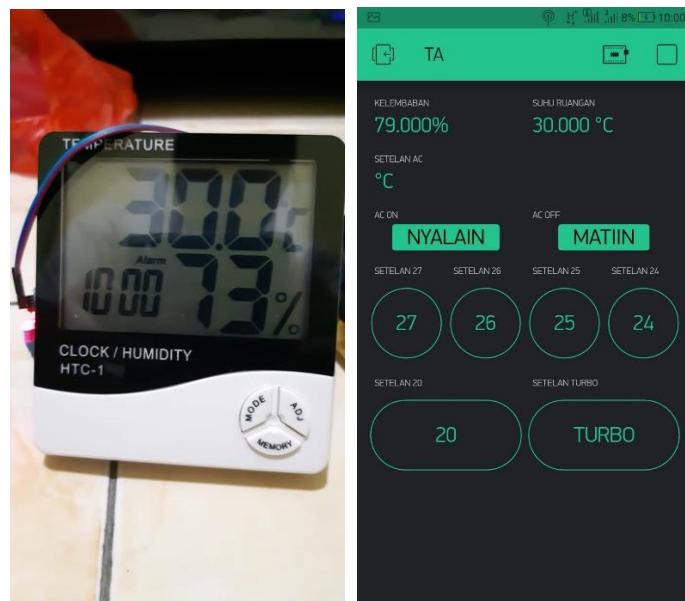
No	Sensor DHT11		Sensor Suhu Ruangan		Error	
	Suhu	Kelembaban	Suhu	Kelembaban	Suhu	Kelembaban
1	29.7°C	83.%	29.8°C	78%	0.003%	5%
2	30.0°C	79.%	30.0°C	73%	0%	6%
3	29.1°C	79.%	29.1°C	67%	0°%	12%
Nilai rata-rata Error					0.001%	7.6%

$$\text{Rumus error pada Suhu} = \frac{|Sensor\ Ruangan - Sensor\ DHT11|}{Sensor\ Ruangan} \times 100\%$$

$$\begin{aligned} &\text{Rumus} && \text{error} && \text{pada} && \text{Kelembaban} \\ &= |sensor\ Ruangan - sensor\ DHT11| \end{aligned}$$



Gambar 3.8 Data Pertama Hasil Perbandingan Sensor DHT11



Gambar 3.9 Data Kedua Hasil Perbandingan Sensor DHT11



Gambar 3.10 Data Ketiga Hasil Perbandingan Sensor DHT11

Berdasarkan data yang diperoleh, maka dapat disimpulkan hasil dari sensor DHT11 yang dibandingkan dengan sensor Suhu ruangan menampilkan perbandingan dengan selisih hasil yang tidak terlalu jauh, itu berarti sensor DHT11 berfungsi dengan baik.

### 3.8.2 Pengujian Serial Monitor Pada Board Mikrokontroler

Komponen-komponen yang di uji untuk pengambilan data adalah sensor DHT11 DAN IR Receiver

1. Pengujian Serial Monitor Pada Board Mikrokontroler ESP8266  
Program Utama

Pengujian serial monitor pada board mikrokontroler ESP8266 ditunjukan pada gambar 3.11.

The screenshot shows a Windows-style application window titled "COM7". The main area displays a continuous stream of text output from an ESP8266 microcontroller. The text includes connection logs, sensor readings (Temperature and Humidity), and AC control commands. The output starts with connecting to a Blynk cloud server and then repeatedly outputs sensor data and AC control commands. The bottom of the window contains standard serial monitor controls: "Autoscroll" (checked), "Newline", "115200 baud" (selected), and "Clear output".

```
[2588] Connecting to blynk-cloud.com:80
[2783] Ready (ping: 95ms).
Temperature: 34.10C
Humidity: 66.00%
AC Temp: 0C
Temperature: 34.00C
Humidity: 66.00%
AC Temp: 0C
Temperature: 34.00C
Humidity: 66.00%
AC Temp: 0C
Temperature: 34.10C
Humidity: 66.00%
AC Temp: 0C
Temperature: 34.20C
Humidity: 65.00%
AC Temp: 0C
```

Gambar 3.11 Serial Monitor Pada Board Mikrokontroler ESP8266 Program

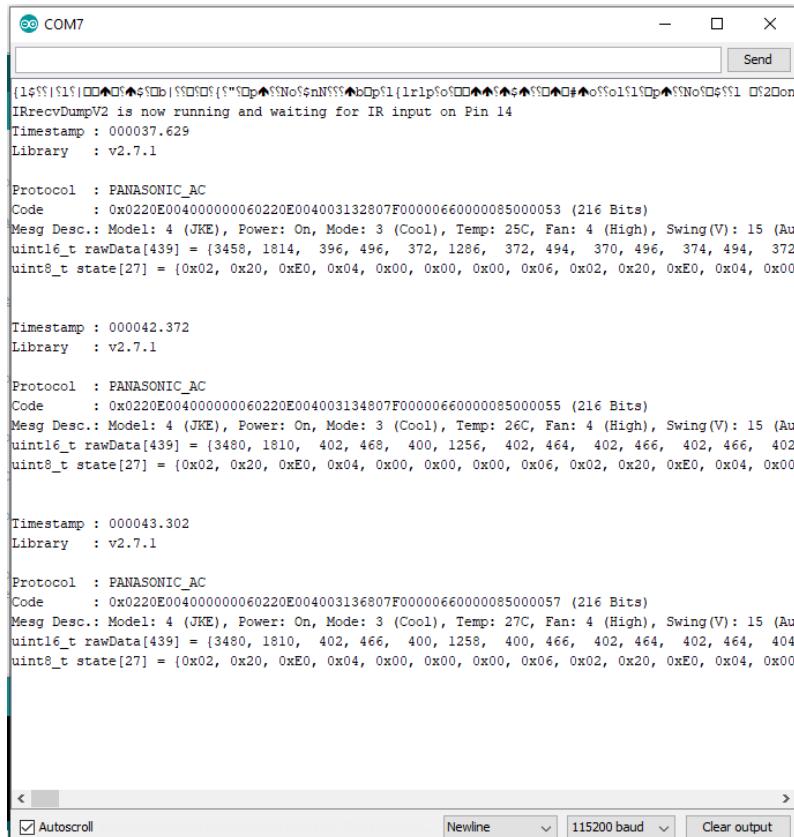
Utama

## 2. Pengujian Serial Monitor Pada Board Mikrokontroler ESP8266

Penerima Inframerah

Pengujian serial monitor pada board mikrokontroler ESP8266

ditunjukan pada gambar 3.12.



Gambar 3.12 Serial Monitor Pada Board Mikrokontroler ESP8266 Penerima Inframerah

### 3.8.3 Pengujian WIFI

Komponen NODEMCU ESP8266 yang di uji untuk penerima wifi lalu mengirimkan data DHT11 ke blynk. Pengujian WIFI ditunjukan pada gambar 3.13 – 3.15

```
char auth[] = "oCrUxxaSpWlcgP_ulrxAA2B7ZEKfwS6_";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "asuka98";
char pass[] = "asukal998";
BlynkTimer timer;
```

Gambar 3.13 Kode Menghubungkan Dan Mengatur Nama WIFI

```
/-\) / /  
/_ / / / / \ /'  
/_ / / \ , / / / / /\ \  
/_/ v0.6.1 on NodeMCU
```

```
[2593] Connecting to blynk-cloud.com:80  
[2781] Ready (ping: 118ms).  
Temperature: 28.70C  
Humidity: 67.00%  
AC Temp: 0C  
Temperature: 28.50C  
Humidity: 67.00%  
AC Temp: 0C  
Temperature: 28.60C  
Humidity: 67.00%  
AC Temp: 0C
```

Gambar 3.14 Serial Monitor Pada Board ESP8266 Penerima Wifi



Gambar 3.15 Tampilan Blynk Pada Smartphone Saat Terhubung ESP8266

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Berdasarkan hasil dari pengujian dan pengambilan data dari alat, maka dapat disimpulkan sebagai berikut :

1. Sensor DHT11 dapat berfungsi dengan baik dalam mendekripsi detak suhu dan kadar kelembaban dalam dalam ruangan dengan persentase error  $\pm 0.01\%$  untuk suhu dan 7.6% untuk kelembaban
2. Inframerah dapat mengirimkan kode ke AC yang dituju dengan benar.
3. OLED SSD1306 dapat menampilkan hasil data dari Sensor DHT11 dan pembacaan penyetelan suhu AC.
4. Alat dapat berfungsi dengan baik dan sesuai dengan yang direncanakan.

#### **4.2 Saran**

Pada tugas akhir dengan judul Prototype Remote Control AC Berbasis IOT, terdapat beberapa saran untuk pengembangan alat :

1. Diperlukan Modul internet tersendiri agar tidak membutuhkan WiFi lagi.
2. Terbatas jarak penempatan alat ke AC karena sinyal infrared mempunyai batasan jarak yang tidak jauh.

## **DAFTAR PUSTAKA**

1. Bambang Tri Wahjo Utomo. (2017). Rancang Bangun Aplikasi Sistem Parkir Mobil Menggunakan Sensor Infra Red. Malang
2. Bejo, Firmansyah Safitri, (2015), “Proyek Robotik Keren dengan Arduino”, PT Elex Media Komputindo, Jakarta
3. Faudin Agus, (2018), *Mengenal Aplikasi Blynk Untuk Fungsi IOT.*  
[https://www.nyebarilmu.com/mengenal-aplikasi-blynk-untuk-fungsi-IOT,](https://www.nyebarilmu.com/mengenal-aplikasi-blynk-untuk-fungsi-IOT)  
[diakses tanggal 21 November 2019]
4. Faudin, Agus. (2017). Mengenal aplikasi BLYNK untuk fungsi IOT.  
[https://www.nyebarilmu.com/mengenal-aplikasi-blynk-untuk-fungsi-IOT/.](https://www.nyebarilmu.com/mengenal-aplikasi-blynk-untuk-fungsi-IOT/)  
[Diakses pada 19 Agustus 2019]
5. Hendro Cahyono (2013). Internet Of Things (Sejarah,Teknologi Dan Penerapannya).  
[http://pusdiklatmigas.esdm.go.id/file/t1internet\\_of\\_things\\_\(gunawan\\_hc\).pdf,](http://pusdiklatmigas.esdm.go.id/file/t1internet_of_things_(gunawan_hc).pdf)  
[diakses tanggal 4 Januari 2020]
6. *Infra Red (IR) Detektor (Sensor Infra Merah).*  
[http://elektronika-dasar.web.id/infra-red-ir-detektor-sensor-infra-merah,](http://elektronika-dasar.web.id/infra-red-ir-detektor-sensor-infra-merah)  
[diakses tanggal 18 November 2019]
7. Muhammad Yan Eka Adiptya, Hari Wibawanto. (2013). Sistem Pengamatan Suhu dan Kelembaban. Journal Teknik Elektro, Semarang.
8. Pressman, Roger S. (2002). Pengertian Software.  
[https://www.maxmanroe.com/vid/teknologi/pengertian-software.html.](https://www.maxmanroe.com/vid/teknologi/pengertian-software.html)  
[Diakses pada 4 Agustus 2019]

9. Sinauarduino. (2016). Mengenal Arduino Software.  
<https://www.sinauarduino.com/artikel/mengenal-arduino-software-ide/>.  
[Diakses pada 4 Agustus 2019]
10. Sulistyanto, M. T., Nugraha, D. A.; dkk. (2015). Impelemtasi IOT (Internet of Things) dalam pembelajaran di Universitas Kanjuruhan Malang. SMARTICS Journal, 1(1), 20-23
11. Widiyaman, Tresna (2016). *Pengertian Modul Wifi ESP8266*. [online].  
<https://www.warriornux.com/pengertian-modul-wifi-esp8266/>,  
[diakses tanggal 3 Januari 2020]

**LAMPIRAN – LAMPIRAN**

## Lampiran 1. Program Utama

```

#include <IRremoteESP8266.h>
#include <IRsend.h>
#include <SPI.h>
#include <Wire.h>
#include <DHT.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Arduino.h>
#include <IRrecv.h>
#include <IRAc.h>
#include <IRtext.h>
#include <IRutils.h>

#define OLED_MOSI 0 // D3
#define OLED_CLK 2 // D4
#define OLED_DC 5 // D1
#define OLED_CS 16 // D0
#define OLED_RESET 4 // D2
Adafruit_SSD1306 display(OLED_MOSI, OLED_CLK,
OLED_DC, OLED_RESET, OLED_CS);
#define DHTTYPE DHT11
#define DHTPIN 14 // D5
DHT dht(DHTPIN, DHTTYPE);
#define SSD1306_LCDHEIGHT 64
#define Desired_temperature 27

uint16_t ACoff[211] = {3818, 1906, 454, 526, 430,
1428, 430, 498, 456, 1402, 456, 500, 454, 1404,
454, 500, 456, 1428, 430, 500, 454, 1402, 456,
500, 454, 1428, 428, 1402, 454, 500, 456, 1402,
456, 500, 454, 1402, 456, 1402, 454, 1402, 456,
1428, 430, 498, 456, 526, 430, 1402, 454, 1402,
454, 500, 454, 500, 456, 500, 456, 498, 456,
1402, 454, 526, 430, 526, 430, 498, 456, 500,
456, 502, 454, 1402, 456, 1402, 454, 500, 456,
526, 430, 1402, 456, 1402, 454, 1428, 430, 500,
454, 500, 454, 500, 454, 498, 456, 1402, 454,
500, 454, 500, 454, 526, 430, 1402, 454, 500,
456, 498, 456, 1428, 428, 1404, 456, 1400, 456,
526, 430, 500, 454, 500, 454, 498, 456, 498,
456, 500, 456, 498, 456, 500, 456, 526, 430,
524, 430, 502, 454, 524, 430, 1402, 454, 500,
454, 526, 430, 500, 454, 500, 456, 498, 456,
}

```

```

502, 454, 498, 456, 498, 456, 500, 454, 500,
456, 500, 456, 1428, 430, 498, 456, 498, 456,
500, 456, 500, 454, 500, 456, 500, 454, 526,
428, 500, 456, 498, 456, 502, 454, 498, 456,
500, 456, 1402, 454, 1428, 430, 1402, 454, 1402,
456, 1402, 454, 500, 456, 500, 454, 500, 456,
1402, 456, 500, 456, 1400, 456, 500, 468}; //  

SHARP_AC  

uint16_t ACon[211] = {3874, 1852, 508, 470, 430,
1404, 508, 444, 510, 1374, 484, 446, 476, 1406,
484, 470, 486, 1372, 484, 472, 482, 1374, 484,
446, 510, 1346, 456, 1404, 510, 444, 508, 1348,
456, 524, 430, 1404, 508, 1348, 454, 1404, 454,
1402, 510, 444, 454, 502, 454, 1402, 454, 1404,
454, 500, 452, 500, 454, 500, 510, 446, 454,
1402, 454, 524, 430, 500, 454, 500, 454, 502,
454, 500, 456, 1426, 430, 1402, 472, 482, 456,
500, 456, 1428, 430, 1428, 430, 1402, 456, 498,
456, 500, 456, 500, 454, 1402, 456, 502, 454,
498, 456, 500, 456, 500, 454, 1402, 454, 500,
454, 500, 454, 1402, 454, 1402, 456, 1428, 430,
500, 454, 500, 454, 524, 430, 502, 452, 500,
454, 526, 428, 526, 430, 524, 430, 500, 454,
498, 456, 500, 454, 500, 454, 1402, 456, 526,
428, 500, 456, 500, 454, 500, 454, 526, 428,
500, 454, 502, 454, 500, 454, 500, 454, 500,
456, 500, 454, 1402, 456, 502, 454, 500, 454,
500, 454, 500, 454, 500, 454, 498, 456, 500,
454, 500, 456, 500, 456, 524, 430, 500, 454,
500, 456, 1402, 456, 1402, 454, 1402, 454, 1428,
430, 1404, 454, 498, 456, 500, 456, 500, 454,
500, 454, 1404, 452, 1402, 456, 502, 492}; //  

SHARP_AC  

uint16_t Temp24[211] = {3804, 1904, 458, 498, 456,
1402, 454, 498, 456, 1402, 456, 498, 456, 1404,
534, 422, 456, 1400, 458, 496, 458, 1402, 456,
498, 458, 1400, 456, 1400, 456, 498, 456, 1402,
456, 500, 454, 1402, 456, 1400, 456, 1400, 458,
1400, 458, 498, 458, 496, 458, 1400, 454, 1402,
456, 498, 458, 496, 456, 526, 430, 498, 456,
1402, 456, 498, 456, 498, 456, 500, 454, 1402,
456, 500, 456, 500, 454, 1404, 454, 500, 456,
498, 456, 1400, 456, 1402, 456, 1428, 508, 422,
456, 498, 456, 500, 456, 1402, 454, 1402, 456,
498, 456, 498, 456, 498, 456, 1402, 456, 500,
454, 500, 456, 1400, 456, 1402, 456, 1402, 454,
500, 456, 498, 456, 500, 456, 500, 454, 500,
456, 498, 456, 500, 456, 498, 456, 498, 456,
524, 430, 498, 456, 500, 456, 1400, 456, 498,

```

```

456, 500, 456, 498, 458, 498, 458, 498, 456,
498, 456, 498, 456, 500, 454, 500, 456, 498,
456, 498, 456, 1402, 456, 500, 456, 498, 456,
1400, 456, 500, 454, 526, 430, 498, 456, 500,
456, 498, 456, 498, 456, 500, 454, 498, 458,
524, 430, 1402, 456, 1400, 456, 1402, 454, 1402,
456, 1402, 454, 498, 456, 498, 456, 498, 458,
1404, 454, 498, 458, 1402, 454, 502, 468}; //
SHARP_AC
uint16_t Temp25[211] = {3820, 1908, 454, 500, 454,
1404, 454, 502, 452, 1402, 456, 500, 454, 1402,
456, 500, 454, 1404, 454, 500, 454, 1404, 454,
502, 454, 1404, 454, 1406, 452, 500, 454, 1404,
454, 500, 454, 1404, 454, 1402, 454, 1404, 454,
1402, 454, 502, 454, 500, 454, 1404, 454, 1404,
454, 500, 456, 500, 454, 502, 454, 502, 454,
1402, 454, 500, 454, 500, 454, 500, 454, 500,
454, 1402, 454, 500, 454, 1402, 454, 502, 454,
500, 456, 1402, 456, 1402, 456, 1402, 454, 502,
454, 502, 454, 500, 454, 1404, 454, 1402, 454,
500, 454, 500, 454, 500, 454, 1402, 456, 500,
454, 500, 454, 1402, 454, 1402, 456, 1402, 454,
500, 454, 500, 454, 498, 456, 500, 454, 502,
454, 500, 456, 498, 456, 500, 452, 502, 454,
500, 454, 500, 454, 500, 454, 1404, 454, 500,
456, 500, 454, 500, 456, 500, 456, 500, 454,
500, 454, 500, 454, 504, 452, 500, 456, 500,
454, 500, 454, 1404, 454, 500, 454, 498, 456,
1402, 454, 500, 454, 502, 454, 500, 454, 500,
454, 500, 454, 500, 454, 502, 454, 502, 452,
502, 454, 1402, 454, 1404, 452, 1402, 454, 1404,
454, 1404, 454, 502, 454, 502, 454, 502, 452,
502, 452, 1404, 454, 1402, 456, 500, 520}; //
SHARP_AC
uint16_t Temp26[211] = {3802, 1908, 454, 502, 454,
1428, 430, 526, 428, 1402, 454, 526, 428, 1402,
454, 500, 454, 1428, 428, 526, 430, 1404, 508,
444, 454, 1404, 452, 1428, 430, 500, 454, 1428,
428, 500, 454, 1404, 452, 1428, 430, 1428, 430,
1402, 454, 526, 428, 502, 452, 1404, 454, 1428,
430, 526, 430, 526, 428, 500, 454, 526, 428,
1430, 428, 526, 428, 502, 454, 500, 454, 1400,
456, 1402, 456, 526, 430, 1402, 456, 500, 454,
502, 454, 1428, 428, 1428, 430, 1428, 428, 526,
428, 526, 428, 526, 430, 1402, 454, 1402, 454,
502, 452, 526, 428, 500, 454, 1428, 430, 500,
454, 526, 430, 1402, 456, 1402, 454, 1404, 454,
526, 428, 502, 452, 502, 454, 524, 430, 526,
428, 500, 454, 526, 428, 500, 454, 500, 454,

```

```

526, 428, 526, 430, 500, 454, 1428, 430, 500,
456, 500, 454, 526, 430, 524, 430, 500, 454,
526, 430, 500, 454, 500, 454, 500, 456, 500,
454, 500, 456, 1402, 456, 500, 454, 526, 428,
1402, 454, 526, 428, 500, 454, 502, 454, 500,
454, 526, 430, 502, 452, 502, 454, 502, 452,
526, 430, 1384, 472, 1428, 430, 1402, 454, 1404,
454, 1404, 454, 502, 452, 500, 454, 500, 456,
1404, 454, 1428, 428, 1428, 430, 526, 442}; //
SHARP_AC
uint16_t Temp27[211] = {3820, 1906, 454, 500, 454,
1430, 430, 526, 428, 1428, 430, 524, 430, 1428,
430, 526, 430, 1402, 456, 500, 454, 1428, 428,
526, 430, 1404, 454, 1402, 454, 500, 456, 1428,
430, 498, 456, 1402, 456, 1402, 456, 1404, 454,
1404, 454, 500, 454, 500, 454, 1428, 430, 1402,
454, 526, 428, 526, 428, 500, 454, 500, 454,
1428, 430, 500, 454, 526, 428, 500, 454, 500,
454, 500, 456, 1402, 454, 1404, 454, 500, 454,
526, 430, 1402, 454, 1404, 454, 1402, 454, 500,
456, 498, 456, 498, 454, 1402, 456, 1402, 454,
526, 430, 500, 454, 500, 454, 1402, 456, 500,
454, 500, 456, 1402, 454, 1430, 428, 1404, 454,
500, 454, 500, 454, 500, 454, 502, 454, 500,
454, 502, 454, 502, 454, 526, 428, 526, 428,
498, 456, 526, 428, 500, 454, 1402, 456, 500,
456, 500, 454, 500, 456, 500, 454, 500, 454,
500, 454, 500, 454, 526, 428, 502, 454, 502,
454, 500, 454, 1428, 430, 498, 456, 500, 454,
1402, 454, 500, 454, 500, 454, 500, 454, 500,
456, 500, 454, 500, 454, 498, 456, 500, 456,
500, 456, 1428, 430, 1402, 454, 1428, 428, 1428,
430, 1404, 454, 500, 454, 500, 454, 500, 454,
524, 430, 500, 454, 502, 454, 500, 494}; //
SHARP_AC
uint16_t Temp20[211] = {3876, 1834, 558, 398, 502,
1356, 502, 452, 500, 1358, 500, 454, 472, 1386,
504, 452, 500, 1356, 504, 452, 472, 1386, 500,
454, 502, 1356, 502, 1356, 502, 452, 500, 1358,
500, 454, 502, 1354, 500, 1358, 502, 1358, 472,
1384, 502, 452, 504, 454, 526, 1330, 504, 1354,
500, 456, 498, 454, 502, 454, 500, 454, 502,
1356, 500, 456, 500, 454, 498, 458, 530, 426,
498, 1358, 530, 1326, 494, 462, 500, 454, 498,
456, 502, 1356, 532, 1326, 472, 1384, 474, 480,
530, 426, 498, 456, 502, 1354, 502, 1356, 474,
482, 500, 454, 500, 456, 472, 1384, 498, 456,
528, 428, 500, 1356, 500, 1356, 528, 1330, 502,
454, 530, 424, 502, 454, 472, 482, 470, 484,

```

```

502, 454, 502, 452, 500, 456, 500, 454, 530,
424, 498, 456, 498, 456, 498, 1360, 528, 426,
476, 480, 498, 456, 474, 482, 472, 482, 502,
454, 472, 482, 530, 424, 502, 454, 500, 454,
498, 456, 530, 1328, 502, 452, 528, 428, 500,
1356, 472, 484, 500, 454, 502, 454, 500, 454,
474, 480, 498, 456, 528, 428, 532, 422, 472,
482, 532, 1328, 472, 1384, 502, 1356, 472, 1384,
502, 1356, 528, 426, 530, 424, 500, 456, 474,
480, 498, 1360, 472, 482, 502, 1356, 438}; //  

SHARP_AC  

uint16_t Temp18[211] = {3842, 1892, 500, 454, 414,  

1442, 538, 418, 532, 1326, 412, 540, 534, 1324,  

476, 478, 506, 1352, 414, 540, 470, 1388, 470,  

484, 470, 1388, 470, 1388, 476, 480, 470, 1386,  

504, 450, 502, 1356, 504, 1354, 470, 1388, 526,  

1332, 440, 516, 500, 454, 506, 1352, 472, 1386,  

502, 452, 536, 420, 468, 486, 414, 540, 504,  

1356, 472, 482, 536, 420, 534, 420, 494, 1364,  

504, 452, 472, 482, 532, 422, 500, 456, 470,  

484, 478, 1380, 528, 1330, 444, 1414, 414, 542,  

506, 448, 470, 486, 414, 1444, 472, 1386, 476,  

478, 538, 418, 534, 420, 474, 1384, 478, 478,  

414, 540, 472, 1386, 500, 1358, 414, 1442, 474,  

482, 478, 478, 478, 412, 540, 478, 478,  

476, 480, 470, 484, 470, 484, 536, 420, 562,  

394, 500, 456, 414, 542, 500, 1356, 414, 540,  

478, 478, 414, 540, 538, 416, 478, 478, 414,  

540, 476, 480, 506, 448, 438, 518, 414, 540,  

508, 448, 472, 1386, 414, 540, 506, 450, 532,  

1324, 474, 482, 414, 540, 470, 484, 504, 450,  

472, 484, 414, 540, 508, 448, 532, 424, 506,  

448, 508, 1350, 508, 1350, 414, 1444, 532, 1324,  

472, 1388, 470, 484, 414, 542, 500, 456, 414,  

1444, 414, 542, 412, 1444, 506, 1352, 442}; //  

SHARP_AC

```

```

char auth[] = "oCrUxxaSpWlcgP_u1rxAA2B7ZEKfwS6_";  

// Your WiFi credentials.  

// Set password to "" for open networks.  

char ssid[] = "asuka98";  

char pass[] = "asuka1998";  

BlynkTimer timer;  

const uint16_t kIrLed = 13; // D7  

IRsend irsend(kIrLed); // Set the GPIO to be used  

to sending the message.

```

```

int AC_Temp;

const uint16_t kRecvPin = 15; // D8
const uint32_t kBaudRate = 115200;
const uint16_t kCaptureBufferSize = 1024;

#if DECODE_AC
const uint8_t kTimeout = 50;
#else
const uint8_t kTimeout = 15;
#endif

const uint16_t kMinUnknownSize = 12;
#define LEGACY_TIMING_INFO false

IRrecv irrecv(kRecvPin, kCaptureBufferSize,
kTimeout, true);
decode_results results;

void setup()
{
  irsend.begin();
#if ESP8266
  Serial.begin(115200, SERIAL_8N1, SERIAL_TX_ONLY);
#else // ESP8266
  Serial.begin(115200, SERIAL_8N1);
#endif // ESP8266
while (!Serial) // Wait for the serial connection
to be established.
  delay(50);
  Serial.printf("\n" D_STR_IRRECVDUMP_STARTUP "\n",
kRecvPin);
#if DECODE_HASH
  // Ignore messages with less than minimum on or
off pulses.
  irrecv.setUnknownThreshold(kMinUnknownSize);
#endif // DECODE_HASH
  irrecv.enableIRIn(); // Start the receiver
display.begin(SSD1306_SWITCHCAPVCC);
display.clearDisplay();
dht.begin();
Blynk.begin(auth, ssid, pass);
}

void loop(){
Suhu();
}

```

```

IR1();
}

void Suhu() {
    Blynk.run();
    // Reading temperature or humidity takes about 250
    milliseconds!
    // Sensor readings may also be up to 2 seconds
    'old' (its a very slow sensor)
    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit =
    true)
    float f = dht.readTemperature(true);

    // Check if any reads failed and exit early (to
    try again).
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println(F("Failed to read from DHT
sensor!"));
        return;
    }

    // text display tests
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.print("Temperature: ");
    display.print(t);display.println("C");
    Serial.print("Temperature: ");
    Serial.print(t);Serial.println("C");
    display.setCursor(0,10);
    display.print("Humidity: ");
    display.print(h);display.println("%");
    Serial.print("Humidity: ");
    Serial.print(h);Serial.println("%");
    display.setCursor(0,20);
    display.print("AC Temp: ");
    display.print(AC_Temp);display.println("C");
    Serial.print("AC Temp: ");
    Serial.print(AC_Temp);Serial.println("C");
    Blynk.virtualWrite(V2, h);
    Blynk.virtualWrite(V1, t);
    Blynk.virtualWrite(V8, AC_Temp);
    delay(2000);
    display.display();
    display.clearDisplay();
}

```

```

void IR1() {
    // Check if the IR code has been received.
    if (irrecv.decode(&results)) {
        // Display a crude timestamp.
        uint32_t now = millis();
        Serial.printf(D_STR_TIMESTAMP " : %06u.%03u\n",
now / 1000, now % 1000);
        // Check if we got an IR message that was to big
for our capture buffer.
        if (results.overflow)
            Serial.printf(D_WARN_BUFFERFULL "\n",
kCaptureBufferSize);
        // Display the library version the message was
captured with.
        Serial.println(D_STR_LIBRARY " : v"
_IRREMOTEESP8266_VERSION_);
        // Display the basic output of what we found.

        Serial.print(resultToHumanReadableBasic(&results));
        // Display any extra A/C info if we have it.
        String description =
IRAcUtils::resultActToString(&results);
        if (description.length())
Serial.println(D_STR_MESGDESC ": " + description);
        yield(); // Feed the WDT as the text output can
take a while to print.
#if LEGACY_TIMING_INFO
        // Output legacy RAW timing info of the result.
        Serial.println(resultToTimingInfo(&results));
        yield(); // Feed the WDT (again)
#endif // LEGACY_TIMING_INFO
        // Output the results as source code
        Serial.println(resultToSourceCode(&results));
        Serial.println(); // Blank line between
entries
        yield(); // Feed the WDT (again)
    }
}

BLYNK_WRITE(V3) {
    if (param.asInt()) {
        irsend.sendRaw(ACon, 211, 38); delay(2000);
//Send Signal to Turn On the AC
        irsend.sendRaw(Temp27, 211, 38);
delay(2000); //Send signal to set 27*C
        AC_Temp = 27;
    }
}

```

```

        irsend.sendRaw(Temp27, 211, 38); //Send
signal to set 27*C
        AC_Temp = 27; }

BLYNK_WRITE(V4) {
    if (param.asInt()) {
        irsend.sendRaw(Temp27, 211, 38);
delay(2000); //Send signal to set 27*C
        AC_Temp = 27;
        else
    {
        irsend.sendRaw(Temp27, 211, 38); //Send
signal to set 27*C
        AC_Temp = 27;
    }

BLYNK_WRITE(V5) {
    if (param.asInt()) {
        irsend.sendRaw(Temp26, 211, 38);
delay(2000); //Send signal to set 26*C
        AC_Temp = 26;
        else
    {
        irsend.sendRaw(Temp26, 211, 38); //Send
signal to set 26*C
        AC_Temp = 26;
    }

BLYNK_WRITE(V6) {
    if (param.asInt()) {
        irsend.sendRaw(Temp25, 211, 38);
delay(2000); //Send signal to set 25*C
        AC_Temp = 25;
        else
    {
        irsend.sendRaw(Temp25, 211, 38); //Send
signal to set 25*C
        AC_Temp = 25;
    }

BLYNK_WRITE(V7) {
    if (param.asInt()) {
        irsend.sendRaw(Temp24, 211, 38);
delay(2000); //Setel suhu 24*C
        AC_Temp = 24;
        else
    {

```

```

        irsend.sendRaw(Temp24, 211, 38); //Setel suhu
24*C
        AC_Temp = 24; }

BLYNK_WRITE(V9) {
    if (param.asInt()) {
        irsend.sendRaw(ACoff, 211, 38);
delay(2000); //Matiin AC
        AC_Temp = 0; }
    else
    {
        irsend.sendRaw(ACoff, 211, 38); //Matiin AC
        AC_Temp = 0; }
    }

BLYNK_WRITE(V10) {
    if (param.asInt()) {
        irsend.sendRaw(Temp20, 59, 38);
delay(2000); //Setel suhu 20*C
        AC_Temp = 20; }
    else
    {
        irsend.sendRaw(Temp20, 59, 38); //Setel suhu
20*C
        AC_Temp = 20; }
    }

BLYNK_WRITE(V11) {
    if (param.asInt()) {
        irsend.sendRaw(Temp18, 59, 38);
delay(2000); //Setel suhu 18*C
        AC_Temp = 18; }
    else
    {
        irsend.sendRaw(Temp18, 59, 38); //Setel suhu
18*C
        AC_Temp = 18; }
    }
}

```

## Lampiran 2. Library Infrared ESP8266

Library dapat di download  
<https://github.com/crankyoldgit/IRremoteESP8266>

```
{
  "name": "IRremoteESP8266",
  "version": "2.7.1",
```

```

    "keywords": "infrared, ir, remote, esp8266,
esp32",
    "description": "Send and receive infrared signals
with multiple protocols (ESP8266/ESP32)",
    "repository":
    {
        "type": "git",
        "url":
"https://github.com/crankyoldgit/IRremoteESP8266.gi
t"
    },
    "authors": [
        {
            "name": "David Conran",
            "url":
"https://plus.google.com/+davidconran",
            "maintainer": true
        },
        {
            "name": "Mark Szabo",
            "url": "http://nomartini-
noparty.blogspot.com/",
            "maintainer": true
        },
        {
            "name": "Sebastien Warin",
            "url": "http://sebastien.warin.fr",
            "maintainer": true
        },
        {
            "name": "Ken Shirriff",
            "email": "zetoslab@gmail.com"
        },
        {
            "name": "Roi Dayan",
            "url": "https://github.com/roidayan/",
            "maintainer": true
        },
        {
            "name": "Massimiliano Pinto",
            "url": "https://github.com/pintomax/",
            "maintainer": true
        }
    ],
    "frameworks": "arduino",
    "platforms": ["espressif8266", "espressif32"]
}

```

### Lampiran 3. Program Penerima Infrared ESP8266

```

#include <Arduino.h>
#include <IRrecv.h>
#include <IRremoteESP8266.h>
#include <IRAc.h>
#include <IRtext.h>
#include <IRutils.h>

const uint16_t kRecvPin = 14;
const uint32_t kBaudRate = 115200;
const uint16_t kCaptureBufferSize = 1024;
#if DECODE_AC
const uint8_t kTimeout = 50;
#else // DECODE_AC
const uint8_t kTimeout = 15;
#endif // DECODE_AC
const uint16_t kMinUnknownSize = 12;
#define LEGACY_TIMING_INFO false

IRrecv irrecv(kRecvPin, kCaptureBufferSize,
kTimeout, true);
decode_results results;

void setup() {
#if defined(ESP8266)
    Serial.begin(kBaudRate, SERIAL_8N1,
SERIAL_TX_ONLY);
#else // ESP8266
    Serial.begin(kBaudRate, SERIAL_8N1);
#endif // ESP8266
    while (!Serial) // Wait for the serial connection
to be established.
        delay(50);
    Serial.printf("\n" D_STR_IRRECVDUMP_STARTUP "\n",
kRecvPin);
#if DECODE_HASH
    // Ignore messages with less than minimum on or
off pulses.
    irrecv.setUnknownThreshold(kMinUnknownSize);
#endif // DECODE_HASH
    irrecv.enableIRIn(); // Start the receiver
}

// The repeating section of the code
void loop() {
    // Check if the IR code has been received.
}

```

```
if (irrecv.decode(&results)) {
    // Display a crude timestamp.
    uint32_t now = millis();
    Serial.printf(D_STR_TIMESTAMP " : %06u.%03u\n",
now / 1000, now % 1000);
    if (results.overflow)
        Serial.printf(D_WARN_BUFFERFULL "\n",
kCaptureBufferSize);
    Serial.println(D_STR_LIBRARY " : v"
_IRREMOTEESP8266_VERSION "\n");

Serial.print(resultToHumanReadableBasic(&results));
    String description =
IRAcUtils::resultActToString(&results);
    if (description.length())
Serial.println(D_STR_MESGDESC ": " + description);
    yield();
#ifndef LEGACY_TIMING_INFO
    Serial.println(resultToTimingInfo(&results));
    yield();
#endif
    Serial.println(resultToSourceCode(&results));
    Serial.println();
    yield();
}
}
```



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS NEGERI JAKARTA  
TEKNIK ELEKTRONIKA**

*Building Future Leaders*  
 Gedung L Kampus A UNJ, Jalan Rawamangun Muka, Jakarta 13220  
 Telepon : ( 62-21 ) 4712137 Fax. 4712137  
 Laman: <http://ft.unj.ac.id> email: [elektronikad3@unj.ac.id](mailto:elektronikad3@unj.ac.id)

**LEMBAR PERSETUJUAN PERBAIKAN UJIAN  
TUGAS AKHIR\***

Nama Mahasiswa : Faris Banu Anwari  
 Nomor Registrasi : 5223165335  
 Program Studi : D3 Teknik Elektronika  
 Judul : "Prototipe Remote Control AC Berbasis IOT"  
 Tanggal Ujian : 07 Februari 2020  
 Batas Akhir Perbaikan : 14 Februari 2020

**PERSETUJUAN:**

NO.	PENGUJI	TANDA TANGAN	TANGGAL
1	Drs. Rimulyo Wicaksono, MM.		18/2/2020
2	Heri Firmansyah, ST., MT.		14/2/2020
3	Syufrijal, ST., MT.		14/2/2020
4	Dr. Muhammad Rif'an, ST., MT.		14/2/2020

Jakarta, .....  
 Koordinator Program Studi  
 D3 Teknik Elektronika

Syufrijal, ST., MT.  
 NIP. 19760327 200112 1 001



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS NEGERI JAKARTA  
TEKNIK ELEKTRONIKA

*Building Future Leaders*  
Gedung L Kampus A UNJ, Jalan Rawamangun Muka, Jakarta 13220  
Telepon : ( 62-21 ) 4712137 Fax. 4712137  
Laman: <http://ft.unj.ac.id> email: [elektronikad3@unj.ac.id](mailto:elektronikad3@unj.ac.id)

**LEMBAR PERSETUJUAN PERBAIKAN UJIAN  
TUGAS AKHIR\***

Nama Mahasiswa	: Much. Gilang Ramadhan
Nomor Registrasi	: 5223165439
Program Studi	: D3 Teknik Elektronika
Judul	: "Prototipe Remote Control AC Berbasis IOT"
Tanggal Ujian	: 07 Februari 2020
Batas Akhir Perbaikan	: 14 Februari 2020

**Persetujuan:**

NO.	PENGUJI	TANDA TANGAN	TANGGAL
1	Drs. Rimulyo Wicaksono, MM.		18/2/2020
2	Heri Firmansyah, ST., MT.		14/2/2020
3	Syufrijal, ST., MT.		14/2/2020
4	Dr. Muhammad Rif'an, ST., MT.		14/2/2020

Jakarta, .....  
Koordinator Program Studi  
D3 Teknik Elektronika

Syufrijal, ST., MT.  
NIP. 19760327 200112 1 001