

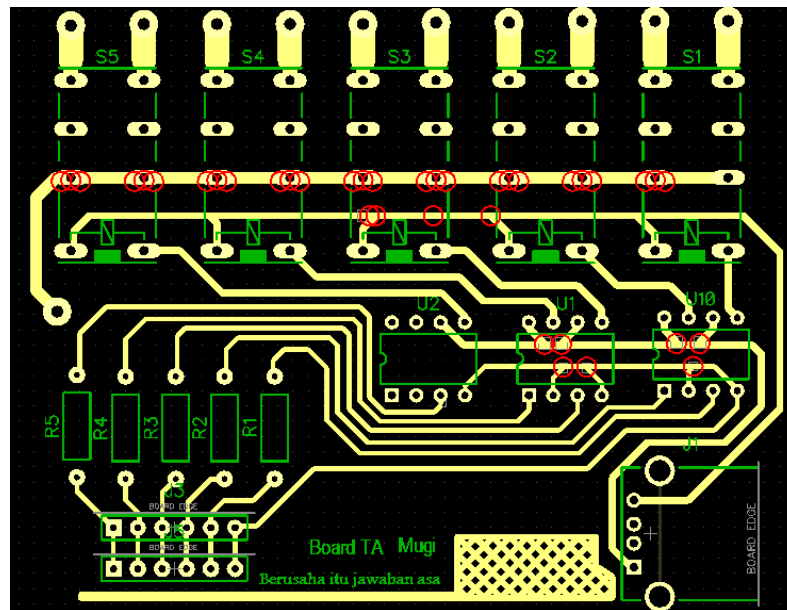
### III.3 Realisasi

Bagian ini adalah perancangan dari konsep blok diagram yang telah dibuat, pada diagram alir yang dirancang, sistemnya terdiri atas *hardware* dan *software* (program).

#### III.3.1 Realisasi Perangkat Keras

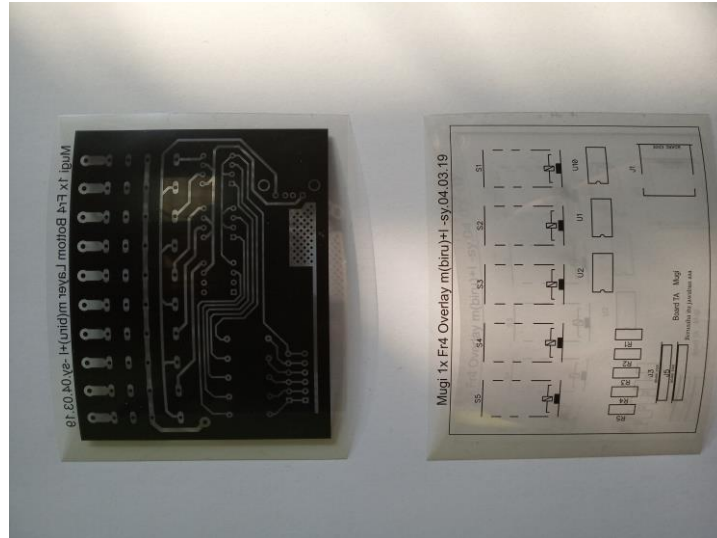
Perangkat keras yang dirakit merupakan sebuah *Printed Circuit Board* (PCB), miniatur rumah, dan kemasan dari perangkat elektronik yang digunakan untuk proyek akhir.

##### III.3.1.1 Realisasi PCB



Gambar III.16 *Layout PCB Board Relay*

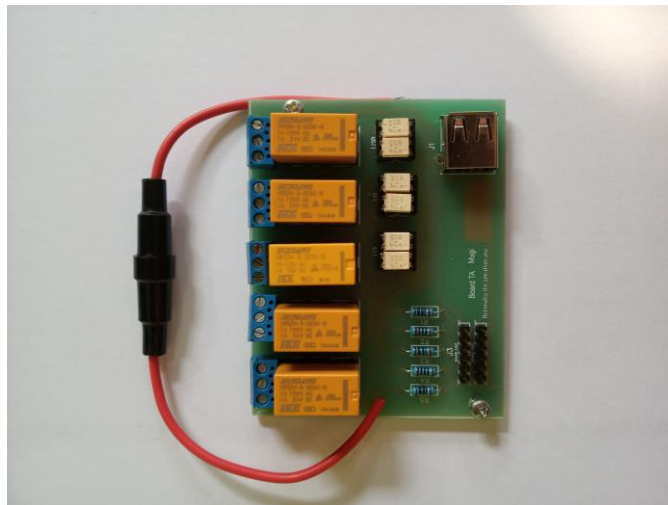
Gambar III.16 menunjukkan *layout* dari *board* relay yang dibuat menggunakan aplikasi Diptrace dengan cara menambahkan komponen yang dibutuhkan ke layer perancangan dan menghubungkan antar kaki komponennya dengan jalur hingga menjadi sebuah rangkaian elektronik dan akan digunakan sebagai aktuator, PCB diatas terdiri dari komponen resistor 100ohm, optocoupler TLP421, relay *Double Pole Double Throw* (DPDT) 5V, port USB *Female* dan pin *Header*.



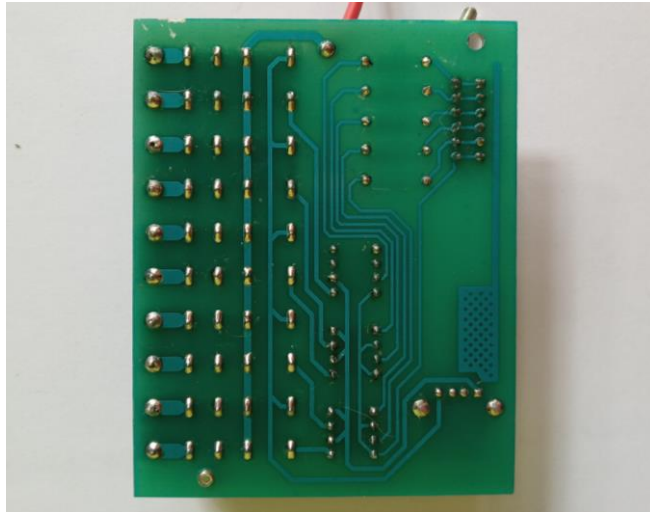
Gambar III.17 *Print Layout* dan Tata Letak PCB

Gambar III.17 merupakan hasil *print* dari *layout* yang dibuat dan gambar *layout* disebelah kiri adalah tampak bawah *board* relay dan tata letak komponen adalah tampak atas dari *board* relay.

### III.3.1.2 Realisasi Perakitan



Gambar III.18 Realisasi PCB *Board* Relay Tampak Atas



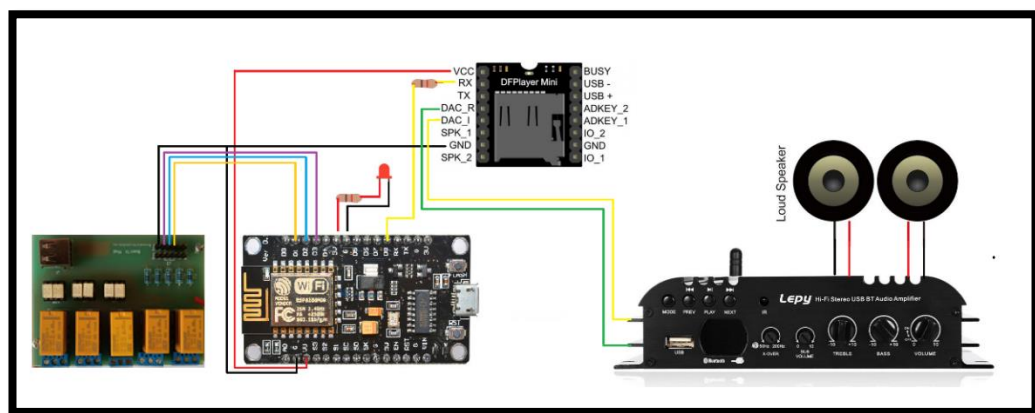
Gambar III.19 Realisasi PCB *Board Relay* Tampak Bawah

Gambar III.18 dan Gambar III.19 adalah hasil cetak PCB dengan ukuran 8,5cm x 7cm dengan tipe *board single* FR4 dan *masking* berwarna biru dengan tebal PCB 2mm yang sudah di solder melalui tahapan penyolderan diawali dengan ukuran tinggi komponen yang paling rendah sampai ke yang paling tinggi agar tiap komponen dipasang dengan sesuai sampai menyentuh ke papan PCB dan tata letak disesuaikan semua untuk membuat estetika yang baik, setelah di solder kaki komponen hendak dipotong sesuai dengan kebutuhan.

### III.3.1.3 Realisasi Pengkabelan

Bagian ini adalah realisasi pengkabelan dari konsep blok diagram yang telah dibuat, pada *wiring* diagram ini memperlihatkan integrasi dari modul yang digunakan pada proyek akhir ini, dan menjelaskan *wiring* dari perangkat yang digunakan.

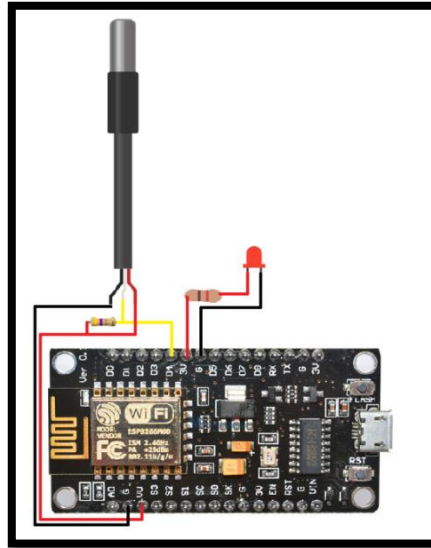
#### III.3.1.3.1 Wiring Diagram Server



Gambar III.20 Wiring Diagram Server

Gambar III.20 menunjukkan integrasi antara pin GPIO 15 (D8) NodeMCU ESP8266 ke pin RX modul DFPlayer, pin Vcc ke *Voltage USB* (VU) dan Gnd ke Gnd dilanjutkan dengan dihubungkan dari pin DAC\_R dan L ke input R dan L *amplifier* dan *output amplifier* ke *speaker*, juga pin Gnd, GPIO D3,D3,D1 (0,4,5) ke *board* relay sebagai aktuator, dan dipasang led sebagai indikator yang dihubungkan seri dengan resistor 330ohm ke 3,3V NodeMCU ESP8266.

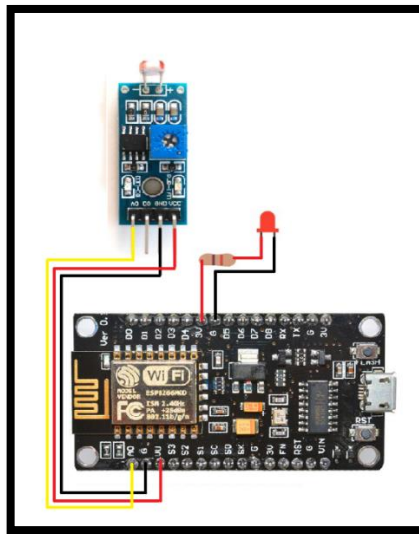
#### III.3.1.3.2 Wiring Diagram Sensor Suhu DS18B20



Gambar III.21 Wiring Diagram Sensor Suhu DS18B20

Gambar III.21 menunjukkan integrasi antara pin GPIO 02(D4) NodeMCU ESP8266 ke pin data dari DS18B20 dan Vcc ke *Voltage USB* (VU), Gnd ke Gnd dan pemasangan resistor 4,7k secara paralel yang berfungsi sebagai *pullup* dari jalur data, dan diperlukan untuk memastikan proses transfer data tetap berjalan dengan baik, dan dipasang led sebagai indikator yang dihubungkan seri dengan resistor 330ohm ke 3,3V NodeMCU ESP8266.

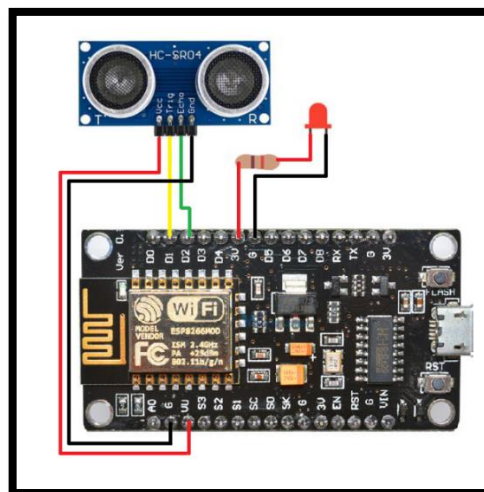
#### III.3.1.3.3 Wiring Diagram Sensor *Light Dependent Resistor* (LDR)



Gambar III.22 *Wiring Diagram Sensor Light Dependent Resistor (LDR)*

Gambar III.22 menunjukkan integrasi antara pin analog pada NodeMCU ESP8266 ke pin analog pada sensor LDR dan Vcc ke *Voltage USB (VU)*, Gnd ke Gnd dan dipasang led sebagai indikator yang dihubungkan seri dengan resistor 330ohm ke 3,3V NodeMCU ESP8266.

#### III.3.1.3.4 *Wiring Diagram Sensor Ultrasonik*



Gambar III.23 *Wiring Diagram Sensor Ultrasonik*

Gambar III.23 menunjukkan integrasi antara pin Vcc, Gnd, *Echo* dan *Trigger* pada sensor ultrasonik dengan *Voltage USB*, Gnd, GPIO D1, D2 (pin 5,4) NodeMCU ESP8266 dimana sinyal dari *trigger* akan dipancarkan dan dipantulkan kembali ke penerima (*echo*).

#### III.3.2 *Realisasi Perangkat Lunak*

Bagian ini membahas realisasi dari program pada proyek ini yang merupakan penjelasan dari diagram alir yang dibuat di prancangan realisasi.

### III.3.2.1 Realisasi Program

Berikut ini merupakan bagian-bagian dari program yang direalisasikan dari NodeMCU ESP8266 menggunakan aplikasi IDE Arduino berdasar dari diagram alir yang dirancang.

```
const char* ssid = "AA_Daus.net";      // Nama AP/Hotspot
const char* password = "netdausaa";    // Password AP/Hotspot

WiFiServer server(80);
IPAddress ip(192, 168, 1, 80);          // IP address of the server
IPAddress gateway(192,168,1,1);         // gateway of your network
IPAddress subnet(255,255,255,0);        // subnet mask of your network
```

Gambar III.24 Program Konfigurasi WiFi

Pada baris 1 terdapat konfigurasi *ssid* dan *password* dengan tipe data yang dimasukan berupa string disesuaikan dengan dari nama jaringan yang akan digunakan, pada baris selanjutnya didapati pengaturan *port* yaitu di *port* 80, karena port 80 biasanya digunakan untuk web server, yang akan berfungsi ketika *user* mengetikan alamat IP atau *hostname* maka akan terbaca pada *port* 80 tersebut.

Baris selanjutnya adalah mengatur IP yang akan digunakan dan itu merupakan IP *static*, gateway yang digunakan untuk menghubungkan suatu jaringan ke jaringan yang lain agar dapat saling berkomunikasi, dan subnetmask yang hendak membedakan *network* ID dengan *host* ID.

```
void setup() {
  myDFPlayer.begin (9600);
  mp3_set_serial (myDFPlayer); //set Serial for DFPlayer-mini mp3 module
  mp3_set_volume (25);
  Serial.begin(115200);
  WiFi.config(ip, gateway, subnet);
  delay(10);
}
```

Gambar III.25 Program Konfigurasi DFPlayer

Selanjutnya pada Gambar III.25 adalah program pada perangkat server yang terhubung ke modul DFPlayer dan aktuator, pada baris pertama didapati pengaturan *baudrate* pada komunikasi serial DFPlayer yang bekerja pada 9600 baud, lalu ada pengaturan *volume* yang diatur pada level 25 dengan spesifikasi yang tertera adalah 0-30, dibaris selanjutnya ada pengaturan *baudrate* NodeMCU ESP8266 di 115200 baud, karena *baudrate* dari kedua perangkat berbeda maka dibuat 2 pengaturan *baudrate*.selanjutnya adalah konfigurasi IP,gateway, dan subnetmask yang telah diatur.

```

if (req.indexOf("KamarON") != -1){
    digitalWrite(Kamar,HIGH);
    mp3_play (2);
    delay (2000);
if (req.indexOf("Lampu_On") != -1){
    //Serial.print("Lampu Kamar ON");
    Serial.print(" ");
    }
}

if (req.indexOf("KamarOFF") != -1){
    digitalWrite(Kamar,LOW);
    mp3_play (3);
    delay (2000);
if (req.indexOf("Lampu_Off") != -1){
    //Serial.print("Lampu Kamar OFF");
    Serial.print(" ");
    }
}

if (req.indexOf("PompaON") != -1){
    digitalWrite(Pompa,HIGH);
    mp3_play (4);
    delay (2000);
    //Serial.print("Pompa ON");
    Serial.print(" ");
}

```

Gambar III.26 Program *request* Perintah dan Pemutar Mp3

Pada Gambar III.26 itu menunjukkan kondisi *request* perintah yang harus dipenuhi di perangkat server, jika *request* perintah sesuai maka mikrokontroler akan mengaktifkan pin *output* digitalnya dan memberikan notifikasi suara berupa Mp3, dan pengaturan delay waktu itu disesuaikan dengan durasi rekaman suara yang diputar pada modul DFPlayer.

```

WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
}
// We now create a URI for the request
Serial.print("Requesting URL: ");
Serial.println(url);
Serial.println(" ");
// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
             "Host: " + host + "\r\n" +
             "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 1000) {
        //Serial.println(">>> Client Timeout !");
        Serial.print(" ");
        client.stop();
        return;
    }
}
// Read all the lines of the reply from server and print them to Serial
while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
}
Serial.println();
Serial.println("closing connection");

```

Gambar III.27 Program Komunikasi WiFi Menggunakan URL

Pada Gambar III.27 menunjukkan program proses pengiriman *request* data dari tiap Node dimana komunikasi akan dilakukan melalui *port* yang sudah ditentukan yaitu *port* 80, maka langkah berikutnya adalah mengirim data ke alamat yang dituju dengan print *request* ditambah IP yang dituju, jika tidak ada masukan *request* maka akan ada indikator *request timeout* dan terus dilakukan pengecekan berulang.

```

const int trigPin = 5;
const int echoPin = 4;

long duration;
int distance;

```

Gambar III.28 Konfigurasi Sensor Ultrasonik



Pada Gambar III.28 ditunjukkan sebuah konfigurasi dari pin sensor ultrasonik yang mana pin *trigger* sebagai *transmitter* di hubungkan ke pin 5 sebagai *output* di NodeMCU ESP8266 dan pin *echo* di hubungkan ke pin 4 sebagai *input*, pada baris selanjutnya ada inisialisasi durasi dan pengukuran.

```
void loop(){
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delay(500);
  digitalWrite(trigPin, LOW);
  // baca echopin dari sinyal yang dipantulkan
  duration = pulseIn(echoPin, HIGH);
  // kalkulasi jarak
  distance= duration*0.034/2;
  // Print jarak di serial monitor

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println("cm");

  const char* url;
  if (distance > 10) {
    url="Pompa_On";
  }
  else {
    url="Pompa_Off";
  }
  int value = 0;

  delay(200);
  ++value;
```

Gambar III.29 Program Parameter dan Pengukur Jarak

Pada Gambar III.29 menunjukkan program dari sensor ultrasonik dalam mengatur pengukuran suatu jarak dengan mengatur proses pengiriman data melalui pin *trigger* dengan delay yang singkat dengan fungsi waktu yang ditentukan oleh pulsa masukan yang aktif dari pin mikrokontroler ke pin *trigger* dan pengukuran jarak yang akan dikalkulasikan dengan dasar, jarak = kecepatan suara x waktu, karena kecepatan suara = 0,034 cm/us. Maka dikarenakan waktu tempuh gelombang suara adalah dua kali yaitu saat pertama dikeluarkan dan dipantulkan dari objek kembali ke sensor maka rumusnya = **durasi x 0.034/2**.

Setelah terukur maka hasil akan di *print* ke serial monitor dengan satuan centimeter (cm), pengiriman data akan dilakukan secara otomatis untuk mengaktifkan aktuator melalui server setelah parameter terpenuhi > 10cm, yang kemudian url dengan string "Pompa\_On" atau "Pompa\_Off" akan dikirimkan sebagai *request* ke server.

```

void loop(){
lampu = analogRead(lampu_1);
Serial.print("Analog read:");
Serial.println(lampu);

    const char* url;
    if (lampu < 700) {
        url="Lampu_On";
    }
    else {
        url="Lampu_Off";
    }

int value = 0;

    delay(500);
    ++value;

```

Gambar III.30 Program Parameter Sensor LDR

Pada Gambar III.30 menunjukkan program dari sensor LDR yang mana terdapat inisialisasi lampu sebagai *analogread* yang digunakan sebagai *input* pembacaan data masukan yang diterima oleh pin analog dari mikrokontroler, yang kemudian akan di *print* ke serial monitor.

Pengiriman data akan dilakukan secara otomatis sebagai data notifikasi lampu secara *real time* melalui server ke aplikasi setelah parameter terpenuhi < 700 untuk perintah lampu menyala dan mati jika parameter tidak dipenuhi, yang didasari data analog memiliki *range* data dari 0-1024 disesuaikan oleh pengaturan *variable resistor* yang terdapat pada modul LDR yang kemudian url dengan string “Lampu\_On” atau “Lampu\_Off” akan dikirimkan sebagai *request* ke server.

```

sensors.requestTemperatures();

Celsius = sensors.getTempCByIndex(0);
Serial.print(Celsius);
Serial.println(" C ");
delay(1000);

const char* url;
if (Celsius > 37) {
    url="Kipas_On";
}
else {
    url="Kipas_Off";
}
int value = 0;

    delay(1000);
    ++value;

```

Gambar III.31 Program Pengukur dan Parameter Suhu

Pada Gambar III.31 yang menunjukkan program dari sensor DS18B20 sebagai pengukur suhu ruangan, dimana pengukuran suhu ruangan diawali dengan *request temperature* yang sebelumnya sudah di inisialisasi dengan awalan “0” agar pembacaan dimulai dari 0, suhu yang terukur akan di *print* ke serial monitor dengan satuan Celcius sebagai satuan standar internasional untuk suatu ukuran suhu.

Pengiriman data akan dilakukan secara otomatis untuk mengaktifkan aktuator melalui server setelah parameter terpenuhi  $> 37^{\circ}\text{C}$  untuk perintah menyalakan pin kipas, yang kemudian url dengan string “Kipas\_On” atau “Kipas\_Off” akan dikirimkan sebagai *request* ke server.

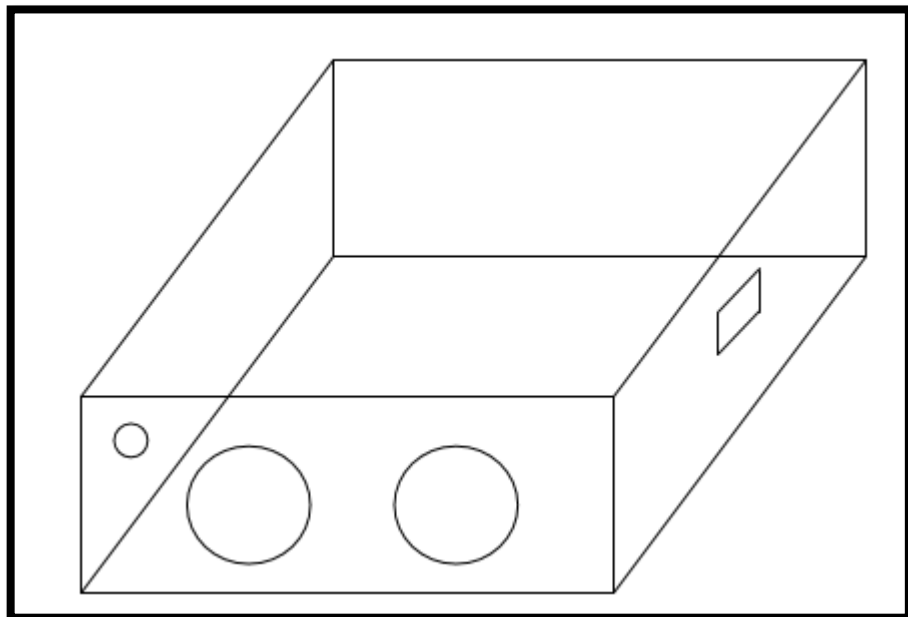
### III.3.3 Realisasi Mekanik

Bagian ini akan menjelaskan tentang realisasi kemasan perangkat elektronik dan miniatur rumah yang sudah dirancang beserta tahapan yang dilakukan dalam proses perealisasiian proyek akhir tersebut.

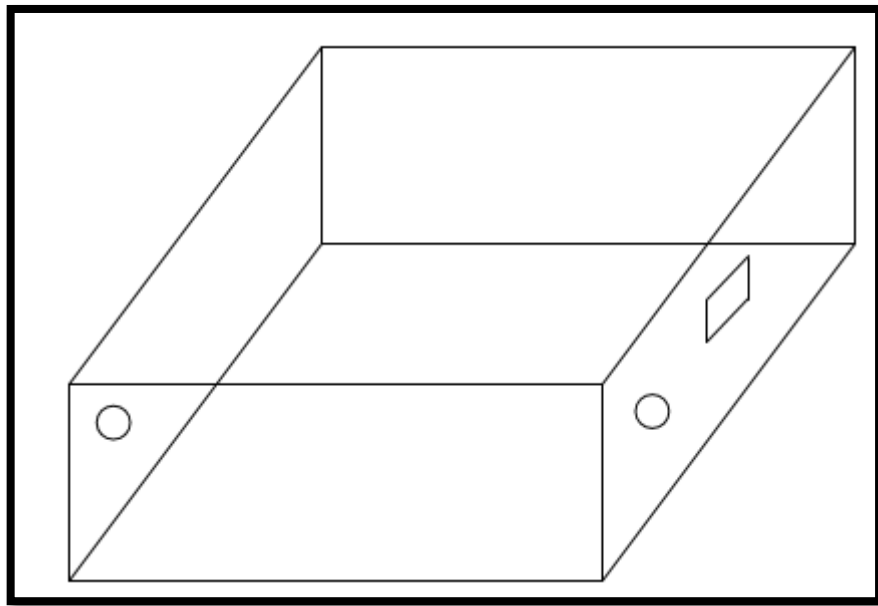
#### III.3.3.1 Realisasi Mekanik Pendukung

Pada realisasi desain perangkat sensor yang terhubung kepada mikrokontroler seperti LDR, DS18B20 dan Ultrasonik semuanya menggunakan *black box* berukuran  $P \times L \times T = 8,5\text{cm} \times 7,5\text{cm} \times 3,5\text{cm}$  dan desain dibuat menggunakan *software visio*.

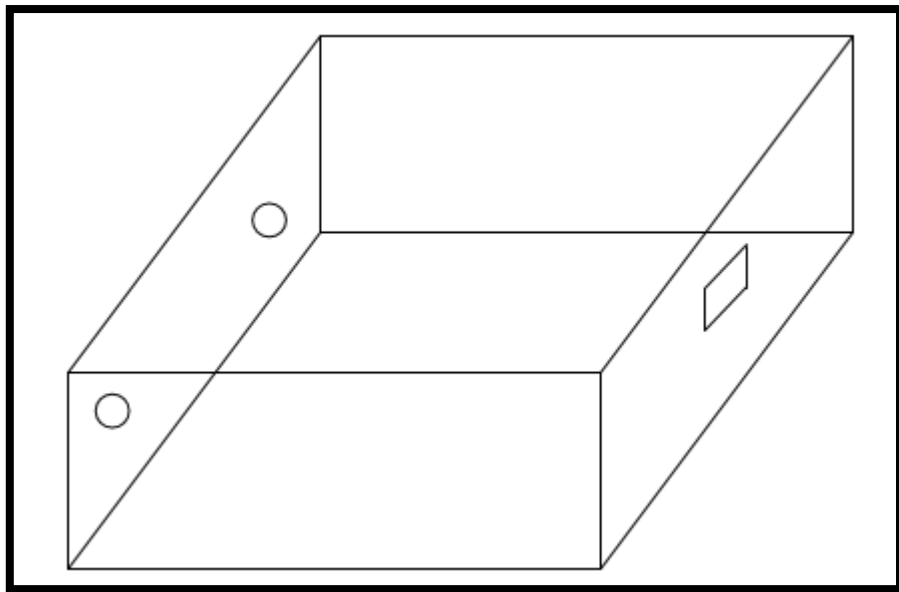
##### 1. Desain dan Konsep Kemasan



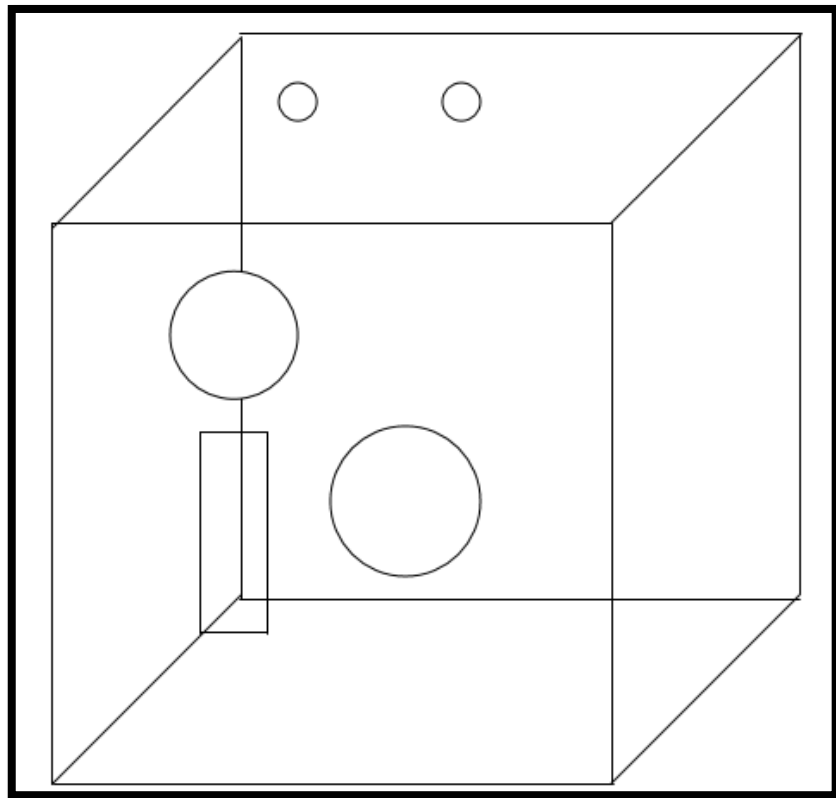
Gambar III.32 Desain *Black Box* Rangkaian Sensor Ultrasonik



Gambar III.33 Desain *Black Box* Rangkaian Sensor DS18B20

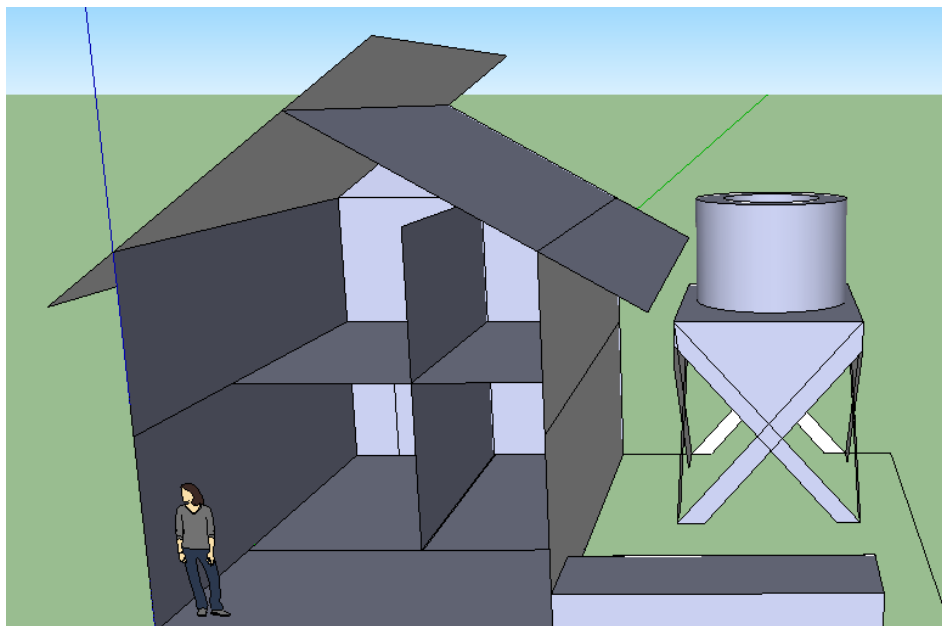


Gambar III.34 Desain *Black Box* Rangkaian Sensor Ultrasonik



Gambar III.35 Desain *Box* Panel Server

Pada Gambar III.35 menunjukkan desain panel server yang terdiri dari kunci panel, lampu indikator aktif yang berwarna merah dan sebuah saklar putar untuk mengaktifkan atau nonaktifkan panel server.



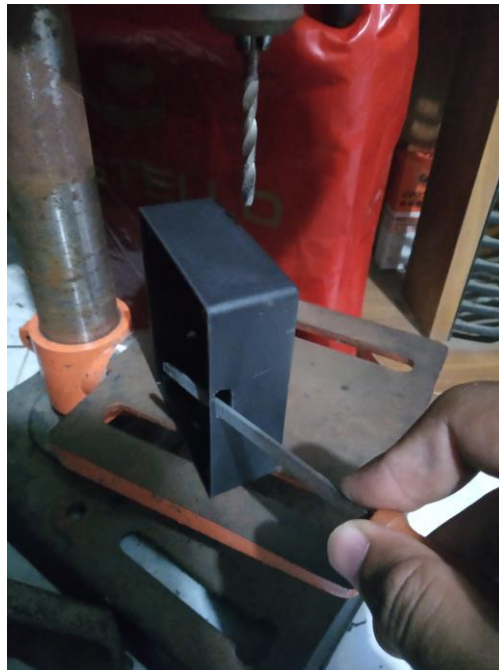
Gambar III.36 Desain dari Miniatur Rumah atau Prototipe

Gambar III.36 diatas merupakan prototipe yang akan digunakan sebagai penunjang pengujian alat dari proyek akhir yang direalisasikan ini, pembuatan desain diatas menggunakan *software* sketchup dengan ukuran realisasi adalah  $P \times L \times T = 55\text{cm} \times 45\text{cm} \times 60\text{cm}$ .

## 2. Proses Pembuatan Kemasan



Gambar III.37 Proses *Drilling Black Box*

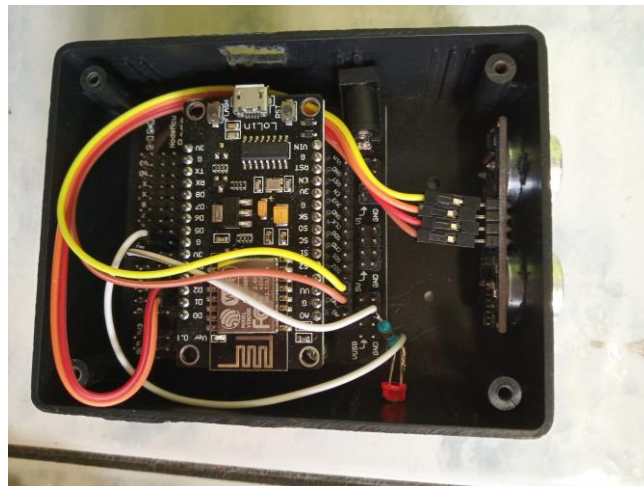


Gambar III.38 Proses Pengikiran *Black Box*

### III.3.3.2 Realisasi Kemasan Alat



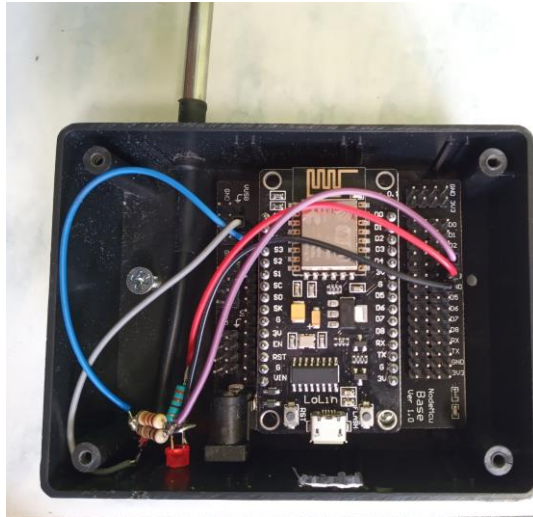
Gambar III.39 Kemasan Rangkaian Sensor Ultrasonik



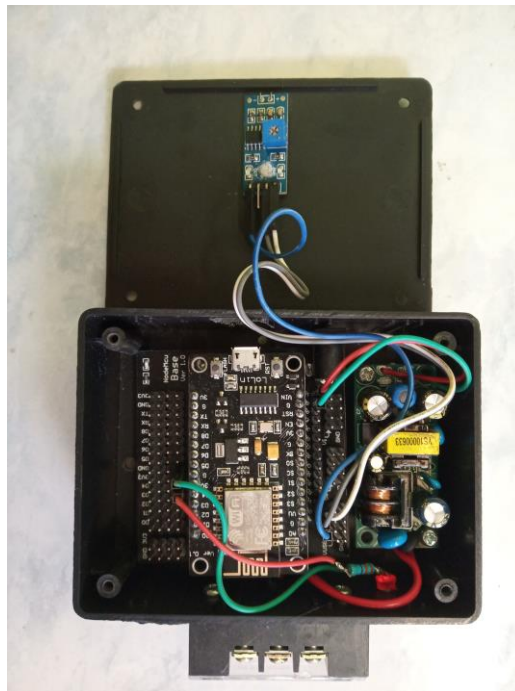
Gambar III.40 Isi Kemasan Rangkaian Sensor Ultrasonik



Gambar III.41 Kemasan Rangkaian Sensor DS18B20



Gambar III.42 Isi Kemasan Rangkaian Sensor DS18B20



Gambar III.43 Kemasan Rangkaian Sensor LDR





Gambar III.44 Miniatur Rumah atau Prototipe