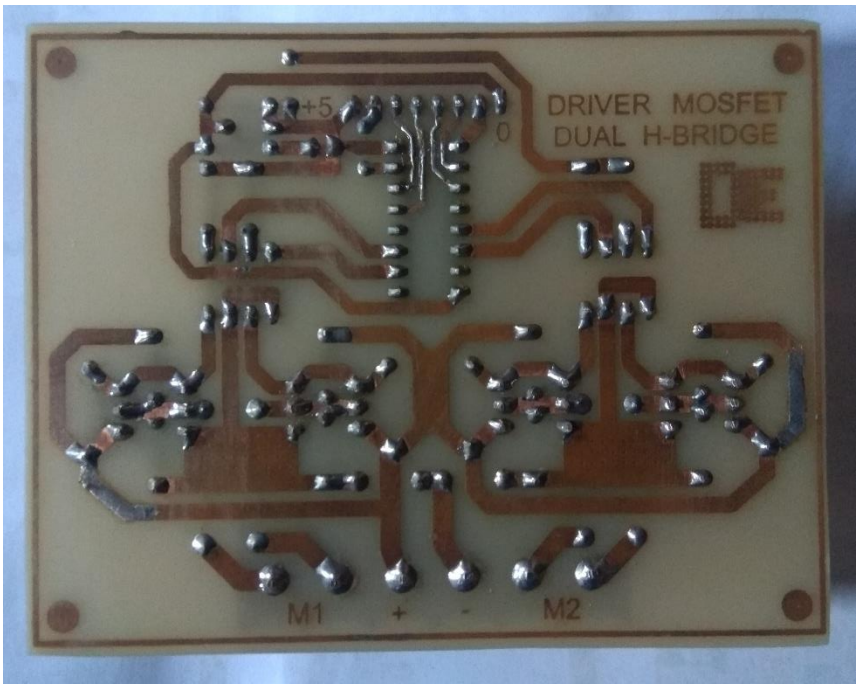
## 3.3. Realisasi

### 3.3.1. Realisasi Perangkat Keras

#### 3.3.1.1. Realisasi PCB

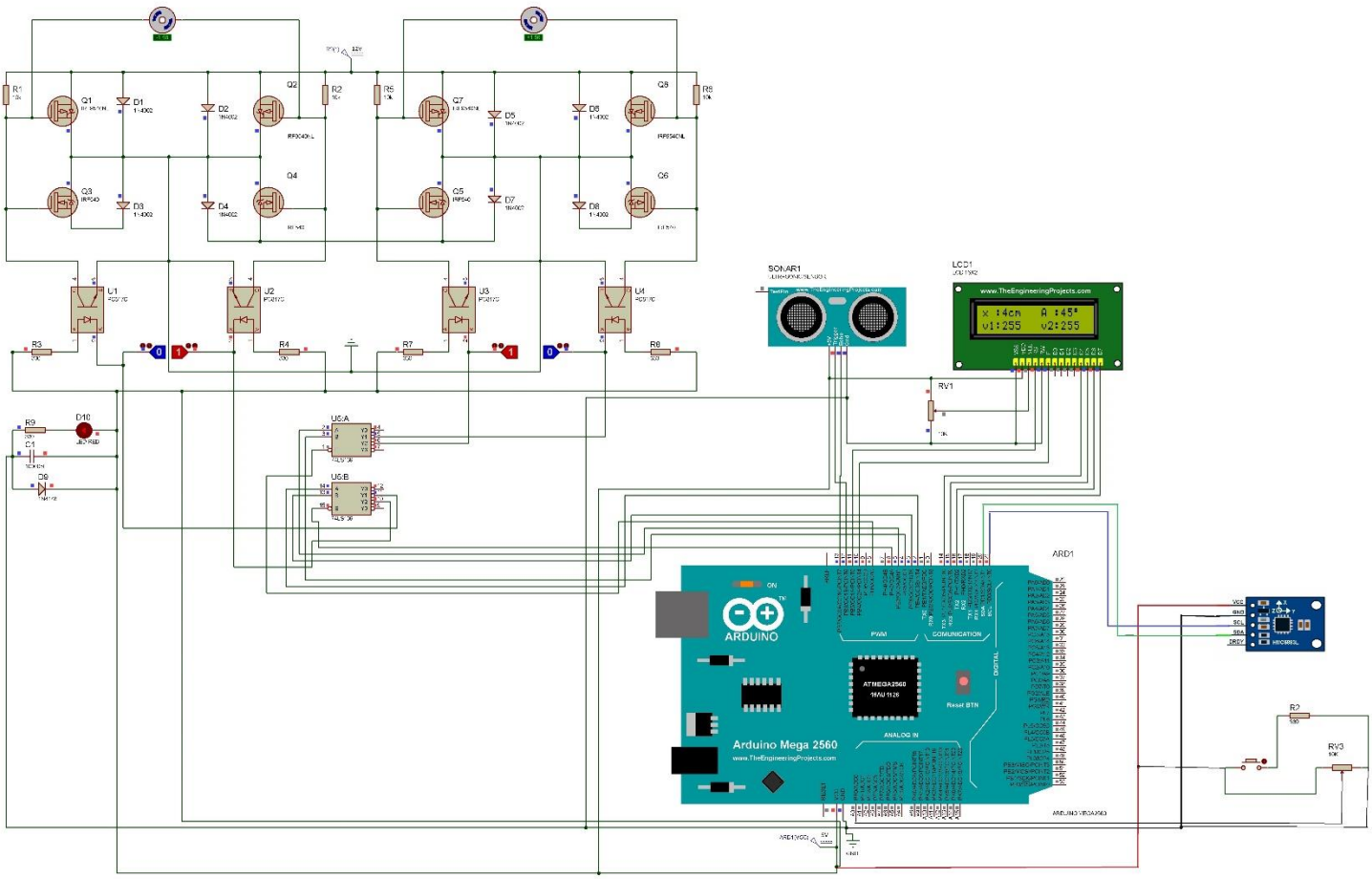
Di bawah ini merupakan realisasi dari desain PCB yang dibuat untuk modul driver motor mosfet h-bridge:



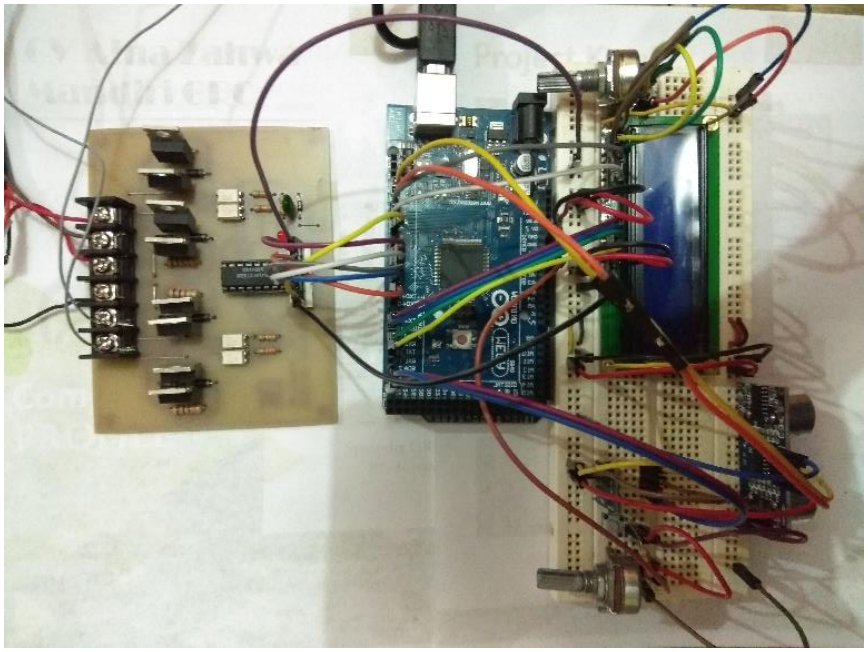

**Gambar 3.11. Rangkaian Keseluruhan**

#### 3.3.1.2. Realisasi Perkabelan

Perangkat keras yang digunakan pada kursi roda elektrik dengan perintah suara adalah menggunakan Arduino Mega 2560, sensor ultrasonik HC-SR04 sebelah kiri, sensor ultrasonik HC-SR04 depan dan belakang, *speech recognition kit (microphone dan raspberry)*, HMC5883L, *Driver* Motor H-Bridge Mosfet, dan Motor *DC*. Pada Gambar 3.4 adalah rancangan *hardware* keseluruhan yang dihubungkan pada Arduino Mega 2560




**Gambar 3.12. Rangkaian Keseluruhan**



**Gambar 3.13. Realisaasi Perkabelan**

### 3.3.2. Realisasi Perangkat Lunak

#### 3.3.2.1.Realisasi Program Kendali Suara / Voice Control

Untuk kendali suara atau voice control, program dikerjakan pada mikroprosesor raspberry. Program digunakan untuk pergerakan maju perlahan, maju normal, maju cepat, berhenti, belok kanan, berhenti, yang mana kata perintah tersebut diubah kedalam biner dan divisualisasikan pada LED untuk selanjutnya nanti akan dikirim ke mikrokontroler untuk sistem kendali pergerakan motornya.

```
import speech_recognition as sr
import RPi.GPIO as GPIO
from os import path
from subprocess import call

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.cleanup();
kesatu= 17
kedua = 22
ketiga = 9
keempat = 11
kelima = 7
keenam = 21
ketujuh = 5
kedelapan =13

GPIO.setup(kesatu, GPIO.OUT)
GPIO.setup(kedua, GPIO.OUT)
GPIO.setup(ketiga, GPIO.OUT)
GPIO.setup(keempat, GPIO.OUT)
GPIO.setup(kelima, GPIO.OUT)
GPIO.setup(keenam, GPIO.OUT)
GPIO.setup(ketujuh, GPIO.OUT)
GPIO.setup(kedelapan, GPIO.OUT)

GPIO.setup(25, GPIO.IN, pull_up_down=GPIO.PUD_UP)

def turnon(led):
    GPIO.output(led, 1)
def turnoff(led):
    GPIO.output(led, 0)

while True :
    input_state = False
    print(input_state)
    if input_state == False:
        print ('Button Pressed')

        AUDIO_FILE                                                    =
path.join(path.dirname(path.realpath(__file__)), "test.wav")
        call(["arecord"   ,"-D"   ,   "plughw:1,0"   ,"-d"   ,   "5"
,"test.wav"])

        r = sr.Recognizer()
```

```
        with sr.AudioFile(AUDIO_FILE) as source:
            audio = r.record(source) # read the entire audio file

            try:
                # for testing purposes, we're just using the
default API key
                txt = r.recognize_google(audio, language='id-
ID')
                print("User Say : " + txt )
                if "berhenti" in txt:
                    turnoff(kesatu)
                    turnoff(kedua)
                    turnoff(ketiga)
                    turnoff(keempat)
                    turnoff(kelima)
                    turnoff(keenam)
                    turnoff(ketujuh)
                    turnoff(kedelapan)
                if "maju" in txt:
                    if "perlahan" in txt:
                        turnon(kesatu)
                        turnoff(kedua)
                        turnoff(ketiga)
                        turnoff(keempat)
                        turnoff(kelima)
                        turnoff(keenam)
                        turnoff(ketujuh)
                        turnoff(kedelapan)
                    if "normal" in txt:
                        turnoff(kesatu)
                        turnon(kedua)
                        turnoff(ketiga)
                        turnoff(keempat)
                        turnoff(kelima)
                        turnoff(keenam)
                        turnoff(ketujuh)
                        turnoff(kedelapan)
                    if "cepat" in txt:
                        turnon(kesatu)
                        turnon(kedua)
                        turnoff(ketiga)
                        turnoff(keempat)
                        turnoff(kelima)
                        turnoff(keenam)
                        turnoff(ketujuh)
                        turnoff(kedelapan)
                if "mundur" in txt:
                    if "perlahan" in txt:
                        turnoff(kesatu)
                        turnoff(kedua)
                        turnon(ketiga)
                        turnoff(keempat)
                        turnoff(kelima)
                        turnoff(keenam)
                        turnoff(ketujuh)
                        turnoff(kedelapan)
                    if "normal" in txt:
                        turnon(kesatu)
                        turnoff(kedua)
```

```
                        turnon(ketiga)
                        turnoff(keempat)
                        turnoff(kelima)
                        turnoff(keenam)
                        turnoff(ketujuh)
                        turnoff(kedelapan)
                if "belok" in txt:
                    if "kanan" in txt:
                        turnoff(kesatu)
                        turnon(kedua)
                        turnon(ketiga)
                        turnoff(keempat)
                        turnoff(kelima)
                        turnoff(keenam)
                        turnoff(ketujuh)
                        turnoff(kedelapan)
                    if "kiri" in txt:
                        turnon(kesatu)
                        turnon(kedua)
                        turnon(ketiga)
                        turnoff(keempat)
                        turnoff(kelima)
                        turnoff(keenam)
                        turnoff(ketujuh)
                        turnoff(kedelapan)

            except sr.UnknownValueError:
                print("Google Speech Recognition could not
understand audio")
            except sr.RequestError as e:
                print("Could not request results from Google
Speech Recognition service {0}".format(e))
"""
finally:
GPIO.cleanup()
"""
```

### 3.3.2.2.Realisasi Program Pengujian Driver Mosfet Dual H-Bridge

Program ini digunakan sebagai program pengujian rangkaian driver mosfet untuk memastikan kinerjanya. Program dieksekusi melalui mikrokontroler Arduino mega 2560 yang diintegrasikan dengan 2 buah motor DC dan catu daya. Berikut listing dari programnya:

```
//output
#define DIR1 2 //direction motor kiri
#define DIR2 3 //direction motor kiri
#define DIR3 4 //direction motor kanan
#define DIR4 5 //direction motor kanan
#define PWM1  6 //pwm motor kiri -> pilih pin PWM bertanda ~
#define PWM2  9 //pwm motor kanan -> pilih pin PWM bertanda ~

//output
#define led 13

//variabel
```

```
unsigned char speed1;
unsigned char speed2;

void setup()
{
//set output
pinMode(DIR1,OUTPUT);
pinMode(DIR2,OUTPUT);
pinMode(DIR3,OUTPUT);
pinMode(DIR4,OUTPUT);
pinMode(PWM1,OUTPUT);
pinMode(PWM2,OUTPUT);
pinMode(led,OUTPUT);

//inisialisasi
digitalWrite(PWM1, HIGH); //stop motor kiri. Sistem PWM aktif LOW
digitalWrite(PWM2, HIGH); //stop motor kanan. Sistem PWM aktif
LOW
}

void maju(unsigned char speed_motor1,unsigned char speed_motor2)
{// 0=stop, 100=full speed

// menggerakkan motor kiri maju
digitalWrite(DIR1, HIGH);
digitalWrite(DIR2, LOW);

// menggerakkan motor kanan maju
digitalWrite(DIR3, HIGH);
digitalWrite(DIR4, LOW);

  speed1=map(speed_motor1,0,100,100,0);  //  konversi  dari  PWM
aktif LOW ke aktif HIGH
  analogWrite(PWM1, speed1);

  speed2=map(speed_motor2,0,100,100,0);  //  konversi  dari  PWM
aktif LOW ke aktif HIGH
  analogWrite(PWM2, speed2);
}

void    mundur(unsigned    char    speed_motor1,unsigned    char
speed_motor2) {// 0=stop, 100=full speed

// menggerakkan motor kiri mundur
digitalWrite(DIR1, LOW);
digitalWrite(DIR2, HIGH);

// menggerakkan motor kanan mundur
digitalWrite(DIR3, LOW);
digitalWrite(DIR4, HIGH);

  speed1=map(speed_motor1,0,100,100,0);  //  konversi  dari  PWM
aktif LOW ke aktif HIGH
  analogWrite(PWM1, speed1);

  speed2=map(speed_motor2,0,100,100,0);  //  konversi  dari  PWM
aktif LOW ke aktif HIGH
  analogWrite(PWM2, speed2);
}
```

```
void putarkiri(unsigned char speed_motor1,unsigned char
speed_motor2) {// 0=stop, 100=full speed

// menggerakkan motor kiri mundur
digitalWrite(DIR1, LOW);
digitalWrite(DIR2, HIGH);

// menggerakkan motor kanan maju
digitalWrite(DIR3, HIGH);
digitalWrite(DIR4, LOW);

  speed1=map(speed_motor1,0,100,100,0);  //  konversi  dari  PWM
aktif LOW ke aktif HIGH
  analogWrite(PWM1, speed1);

  speed2=map(speed_motor2,0,100,100,0);  //  konversi  dari  PWM
aktif LOW ke aktif HIGH
  analogWrite(PWM2, speed2);
}

void putarkanan(unsigned char speed_motor1,unsigned char
speed_motor2) {// 0=stop, 100=full speed

// menggerakkan motor kiri maju
digitalWrite(DIR1, HIGH);
digitalWrite(DIR2, LOW);

// menggerakkan motor kanan mundur
digitalWrite(DIR3, LOW);
digitalWrite(DIR4, HIGH);

  speed1=map(speed_motor1,0,100,100,0);  //  konversi  dari  PWM
aktif LOW ke aktif HIGH
  analogWrite(PWM1, speed1);

  speed2=map(speed_motor2,0,100,100,0);  //  konversi  dari  PWM
aktif LOW ke aktif HIGH
  analogWrite(PWM2, speed2);
}

void berhenti()
{
// now turn off motors
digitalWrite(DIR1, LOW);
digitalWrite(DIR2, LOW);
digitalWrite(DIR3, LOW);
digitalWrite(DIR4, LOW);

//matikan motor kiri dan motor kanan
digitalWrite(PWM1,HIGH);
digitalWrite(PWM2,HIGH);
}


// main program
void loop(){

//gerakan maju
```

```
//jika ada motor yang mundur maka tukar posisi kabel pada motor
yang mundur tersebut
maju(1,1); // speed kiri 100, kanan 100
digitalWrite(led, HIGH); // indikator bergerak
delay(2000);
digitalWrite(led, LOW);

//gerakan berhenti
berhenti();
delay(1000);

//gerakan mundur
mundur(1,1); // speed kiri 128, kanan 128
digitalWrite(led, HIGH); // indikator bergerak
delay(2000);
digitalWrite(led, LOW);

//gerakan berhenti
berhenti();
delay(1000);

//gerakan putar kiri
putarkiri(1,1); // speed kiri 64, kanan 64
digitalWrite(led, HIGH); // indikator bergerak
delay(2000);
digitalWrite(led, LOW);

//gerakan berhenti
berhenti();
delay(1000);

//gerakan putar kanan
putarkanan(1,1); // speed kiri 64, kanan 64
digitalWrite(led, HIGH);  // indikator bergerak
delay(2000);
digitalWrite(led, LOW);

//gerakan berhenti
berhenti();
delay(1000);

}//end
```

### 3.3.2.3.Realisaasi Program Komunikasi

Program komunikasi antara mikrokontroler arduino dan raspberry belum terealisasi.

### 3.3.2.4.Realisasi Program Pergerakan Motor Tanpa Metode

Pada program ini digunakan sensor ultrasonik dan sensor kompas digital yaitu HMC5883L. Besarnya kecepatan dari motor diatur oleh sebuah potensiometer yang memiliki nilai 0-255. Rangkaian dan program ini diimplementasikan terlebih

dahulu ada sebuah prototype menggunakan motor DC dengan tegangan kecil. Selanjutnya akan diterapkan pada motor untuk kursi roda pintar dan akan dikembangkan kembali untuk program kendalinya agar perintah yang dapat dikendalikan lebih variatif lagi. Berikut listing programnya:

```
#include <Wire.h> /* Include the standard Wire library */
#define HMC5803L_Address 0x1E /* The I2C address of the module */
#define X 3 /* Register address for the X Y and Z data */
#define Y 7
#define Z 5
double angle;
#include <LiquidCrystal.h>
LiquidCrystal lcd(11, 10, 14, 15, 16, 17); //Rs, E, D4, D5, D6,
D7
//output
#define DIR1 2 //direction motor kiri
#define DIR2 3 //direction motor kiri
#define DIR3 4 //direction motor kanan
#define DIR4 5 //direction motor kanan
#define PWM1 6 //pwm motor kiri -> pilih pin PWM bertanda ~
#define PWM2 9 //pwm motor kanan -> pilih pin PWM bertanda ~

#define button 52
int rotDirection = 0;
int pressed = false;

const int pinT = 12; // pin Trigger
const int pinE = 13; // pin Echo
//output
#define led 13

int durasi, jarak;

//variabel
unsigned char speed1;
unsigned char speed2;

void setup()
{
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  lcd.print("hello, world!");
  delay(500);
  lcd.clear();
  Wire.begin(); /* Initialise the Wire library */
 Init_HMC5803L(); /* Initialise the module */

//set output
pinMode(DIR1,OUTPUT);
pinMode(DIR2,OUTPUT);
pinMode(DIR3,OUTPUT);
pinMode(DIR4,OUTPUT);
pinMode(PWM1,OUTPUT);
pinMode(PWM2,OUTPUT);
pinMode(led,OUTPUT);
pinMode(button, INPUT);
```

```
pinMode(pinT, OUTPUT);
 pinMode(pinE, INPUT);

 //inisialisasi
digitalWrite(PWM1, HIGH); //stop motor kiri. Sistem PWM aktif LOW
digitalWrite(PWM2, HIGH); //stop motor kanan. Sistem PWM aktif
LOW
}

// main program
void loop(){

  int potValue = analogRead(A0); // Read potentiometer value
  int pwmOutput = map(potValue, 0, 1023, 0, 255); // Map the
potentiometer value from 0 to 255
  analogWrite(PWM1, pwmOutput); // Send PWM signal to L298N Enable
pin
    analogWrite(PWM2, pwmOutput); // Send PWM signal to L298N
Enable pin
  // Read button - Debounce
  if (digitalRead(button) == true) {
    pressed = !pressed;
  }
  while (digitalRead(button) == true);
  delay(20);
  // If button is pressed - change rotation direction
  if (pressed == true  & rotDirection == 0) {
    digitalWrite(DIR1, HIGH);
    digitalWrite(DIR2, LOW);
     digitalWrite(DIR3, HIGH);
    digitalWrite(DIR4, LOW);
    rotDirection = 1;
    delay(20);
  }
  // If button is pressed - change rotation direction
  if (pressed == false & rotDirection == 1) {
    digitalWrite(DIR1, LOW);
    digitalWrite(DIR2, HIGH);
     digitalWrite(DIR3, HIGH);
    digitalWrite(DIR4, LOW);
    rotDirection = 0;
    delay(20);
  }

  //speed1=constrain(speed1, 0, 100);
  //speed1 = 100;// konversi dari PWM aktif LOW ke aktif HIGH
  //analogWrite(PWM1, speed1);

  //speed2=constrain(speed2, 0, 100);
 // speed2 = 100;// konversi dari PWM aktif LOW ke aktif HIGH
  //analogWrite(PWM2, speed2);

   // ----------- mengaktifkan sensor ultrasonic --------------
//
 // mengaktifkan pin trigger
 digitalWrite(pinT, HIGH);
 // delay 10 mikrodetik
 delayMicroseconds(10);
```

```
 // mematikan pin trigger
 digitalWrite(pinT, LOW);
 delayMicroseconds(2);

// mendapat data durasi pantulan gelombang ultrasonic
 durasi = pulseIn(pinE, HIGH);
 // konversi durasi ke jarak dalam satuan centimeter(cm)
 jarak = ((durasi * 0.034) / 2);

 angle=
(atan2((double)HMC5803L_Read(X),(double)HMC5803L_Read(Y)) * (180
/
 3.14159265)) + 180; // angle in

 Serial.print(HMC5803L_Read(X)); /* Read each sensor axis data
and output to the
 serial port */
 Serial.print(", ");
 Serial.print(HMC5803L_Read(Y));
 Serial.print(", ");
 Serial.print(HMC5803L_Read(Z));
 Serial.print(", ");
 Serial.print(angle,0);
 Serial.print(", ");

  lcd.clear();

 if((angle <= 22.1) || (angle >= 337.1 )){
 Serial.print("North");
 //lcd.print("North");
 }
 if((angle >= 22) && (angle <=67.1 )){
 Serial.print("North-East");
 //lcd.print("North-East");
 }
 if((angle >= 67) && (angle <= 112.1 )){
 Serial.print("East");
 //lcd.print("East");
 }
 if((angle >= 112) && (angle <= 157.1 )){
 Serial.print("South-East");
 //lcd.print("South-East");
 }
 if((angle >= 157) && (angle <= 202.1 )){
 Serial.print("South");
 //lcd.print("South");
 }
 if((angle >= 202) && (angle <= 247.1 )){
 Serial.print("South-West");
 //lcd.print("South-West");
 }
 if((angle >= 247) && (angle <= 292.1 )){
 Serial.print("West");
// lcd.print("West");
}
 if((angle >= 292) && (angle <= 337 )){
 Serial.print("North-West");
 //lcd.print("North-West");
 }
```

```
lcd.setCursor(0,0);
lcd.print("x :");
lcd.print(jarak);
lcd.print("cm ");
lcd.setCursor(9,0);
lcd.print("A :");
lcd.print(angle,0);
lcd.print((char)223);

lcd.setCursor(0,1);
lcd.print("v1:");
lcd.print(pwmOutput);
lcd.setCursor(8,1);
lcd.print(" ");
lcd.print("v2:");
lcd.print(pwmOutput);


delay(500); //delay 500ms

//gerakan maju
//jika ada motor yang mundur maka tukar posisi kabel pada motor
yang mundur tersebut
// ------------ mengatur pergerakan robot --------------- s//
 // jika jarak lebih besar atau sama dengan 20 cm
 if (jarak >= 20)
 {
 // jalan maju
 digitalWrite(DIR1, HIGH); //searah jarum jam <CW>
 digitalWrite(DIR2, LOW);
 digitalWrite(DIR3, HIGH);
 digitalWrite(DIR4, LOW);
 }


 // jika jarak kurang dari atau sama dengan 19 cm
 else if (jarak <= 15)
 {
 // jalan mundur selama 300 milidetik
 digitalWrite(DIR1, LOW);
 digitalWrite(DIR2, LOW);
 digitalWrite(DIR3, LOW);
 digitalWrite(DIR4, LOW);
 delay(300);
digitalWrite(led, HIGH); // indikator bergerak
delay(2000);
digitalWrite(led, LOW);
 }
}//end

void Init_HMC5803L(void)
{
 /* Set the module to 8x averaging and 15Hz measurement rate */
 Wire.beginTransmission(HMC5803L_Address);
 Wire.write(0x00);
 Wire.write(0x70);

 /* Set a gain of 5 */
```

```
 Wire.write(0x01);
 Wire.write(0xA0);
 Wire.endTransmission();
}
/* This function will read once from one of the 3 axis data
registers
and return the 16 bit signed result. */
int HMC5803L_Read(byte Axis)
{
 int Result;

 /* Initiate a single measurement */
 Wire.beginTransmission(HMC5803L_Address);
 Wire.write(0x02);
 Wire.write(0x01);
 Wire.endTransmission();
 delay(6);

 /* Move modules the resiger pointer to one of the axis data
registers */
 Wire.beginTransmission(HMC5803L_Address);
 Wire.write(Axis);
 Wire.endTransmission();

 /* Read the data from registers (there are two 8 bit registers
for each axis) */
 Wire.requestFrom(HMC5803L_Address, 2);
 Result = Wire.read() << 8;
 Result |= Wire.read();
 return Result;
}
```

### 3.3.3. Realisasi Mekanik

Bahan yang digunakan untuk *casing* sistem pengendalian kecepatan ini yaitu *plastic casing*. *Casing* yang penulis rancang yaitu sebanyak satu buah, yang sudah mencakup semua komponen yang diperlukan yaitu memuat mikrokontroller, rangkaian driver motor dengan mosfet, dan LCD serta sensor berada di bagian luar. Modul dan sensor ini akan diletakan di bagian bawah kursi roda beserta baterai di bagian belakang.

*Casing* modul sistem yang penulis rancang berdimensi 20cm x 16cm x 5cm. Lebih jelasnya perancangan disain *casing* ditunjukkan pada **Gambar III.35**. Berikut tampilan dari casing yang dibuat

Gambar 3.14. Contoh Gambar untuk Casing