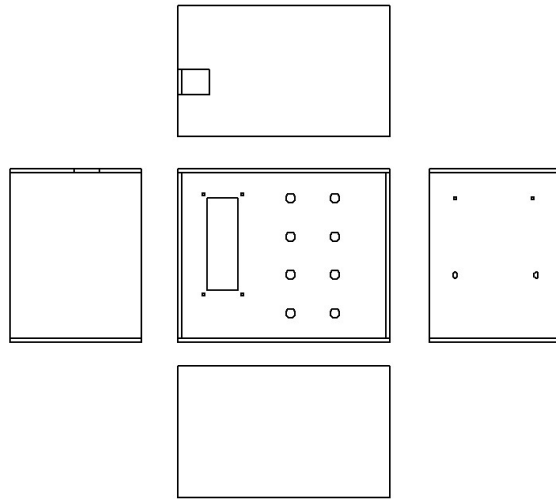
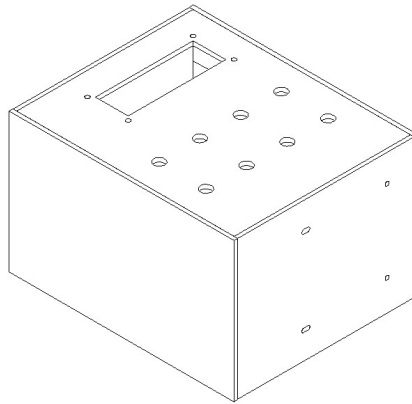


LAMPIRAN

Lampiran 1. Desain 2D Akrilik



Lampiran 2. Desain 3D Akrilik



Lampiran 3. Pengujian Keseluruhan



Lampiran 4. Kode Program

```
#include "ACS712.h"
#include <SPI.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
SoftwareSerial esp8266(4,5);
String AP = "AndroidAP1"; // CHANGE ME
String PASS = "20111998"; // CHANGE ME
String API = "TQ6BCW303Y16J26W"; // CHANGE ME
String HOST = "api.thingspeak.com";
String PORT = "80";
String field1 = "field1";
String field2 = "field2";
String field3 = "field3";
String field4 = "field4";
String field5 = "field5";

//TEGANGAN DC
const int voltageSensor = A3;
float vOUT = 0.0;
float VDC = 0.0;
float R1 = 30000.0;
float R2 = 7500.0;
int value = 0;

//RPM
float rev=0;
```

```

int rpm;
int oldtime=0;
int time;

//SENSOR ARUS
ACS712 sensorac(ACS712_20A, A0);
ACS712 sensordc(ACS712_20A, A1);
int countTrueCommand;
int countTimeCommand;
boolean found = false;

void isr() //interrupt service routine
{
rev++;
}

void setup() {
  Serial.begin(9600);
  attachInterrupt(0,isr,RISING); //attaching the interrupt
  Serial.println("Running");
  lcd.init();
  lcd.init();
  lcd.backlight();
  sensorac.calibrate();
  sensordc.calibrate();
  esp8266.begin(9600);
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\", \""+ PASS +"\",20,"OK");
}

```

```

void loop() {
  //Arus AC
  float IAC = sensorac.getCurrentAC();

  //Tegangan AC
  double a = analogRead(A2);
  double VAC = a * (225.0/1023.0);
  double c = a * (5/1023.0);
  delay(1500);

  //Arus DC
  float IDC = sensordc.getCurrentDC();

  //Tegangan
  value = analogRead(voltageSensor);
  vOUT = (value * 5.0) / 1024.0;
  VDC = vOUT / (R2/(R1+R2));

  //RPM
  detachInterrupt(0); //detaches the interrupt
  time=millis()-oldtime; //finds the time
  rpm=(rev/time)*6000*3; //calculates rpm for blades
  oldtime=millis(); //saves the current time
  rev=0;
  attachInterrupt(0,isr,RISING);

  //Serial monitor
  Serial.println(String("I AC = ") + IAC + " A");
}

```

```

Serial.println(String("I DC = ") + IDC + " A");
Serial.println(String("V AC = ") + VAC + " V");
Serial.println(String("V DC = ") + VDC + " V");
Serial.println(String("Kecepatan = ") + rpm + " RPM");

// Upload Data
String getData2 = "GET /update?api_key="+ API +"&"+ field1
+"="+String(IAC)
+""+"GET /update?api_key="+ API +"&"+ field2 +"="+String(IDC)
+""+"GET /update?api_key="+ API +"&"+ field3 +"="+String(VAC)
+""+"GET /update?api_key="+ API +"&"+ field4 +"="+String(VDC)
+""+"GET /update?api_key="+ API +"&"+ field5 +"="+String(rpm);
sendCommand("AT+CIPMUX=1",5,"OK");
sendCommand("AT+CIPSTART=0,\"TCP\", \"\"+ HOST +"\", "+
PORT,15,"OK");
sendCommand("AT+CIPSEND=0,"
+String(getData2.length()+4),4,">");
esp8266.println(getData2);countTrueCommand++;
sendCommand("AT+CIPCLOSE=0",5,"OK");
delay(1000);
}

void sendCommand(String command, int maxTime, char
readReplay[]) {
Serial.print(countTrueCommand);
Serial.print(". UPLOAD DATA => ");
Serial.print(command);
Serial.print(" ");
while(countTimeCommand < (maxTime*1))
{

```

```
    esp8266.println(command);//at+cipsend
    if(esp8266.find(readReplay))//ok
    {
        found = true;
        break;
    }

    countTimeCommand++;
}

if(found == true)
{
    Serial.println("DONE");
    countTrueCommand++;
    countTimeCommand = 0;
}

if(found == false)
{
    Serial.println("FAIL");
    countTrueCommand = 0;
    countTimeCommand = 0;
}

found = false;
}
```