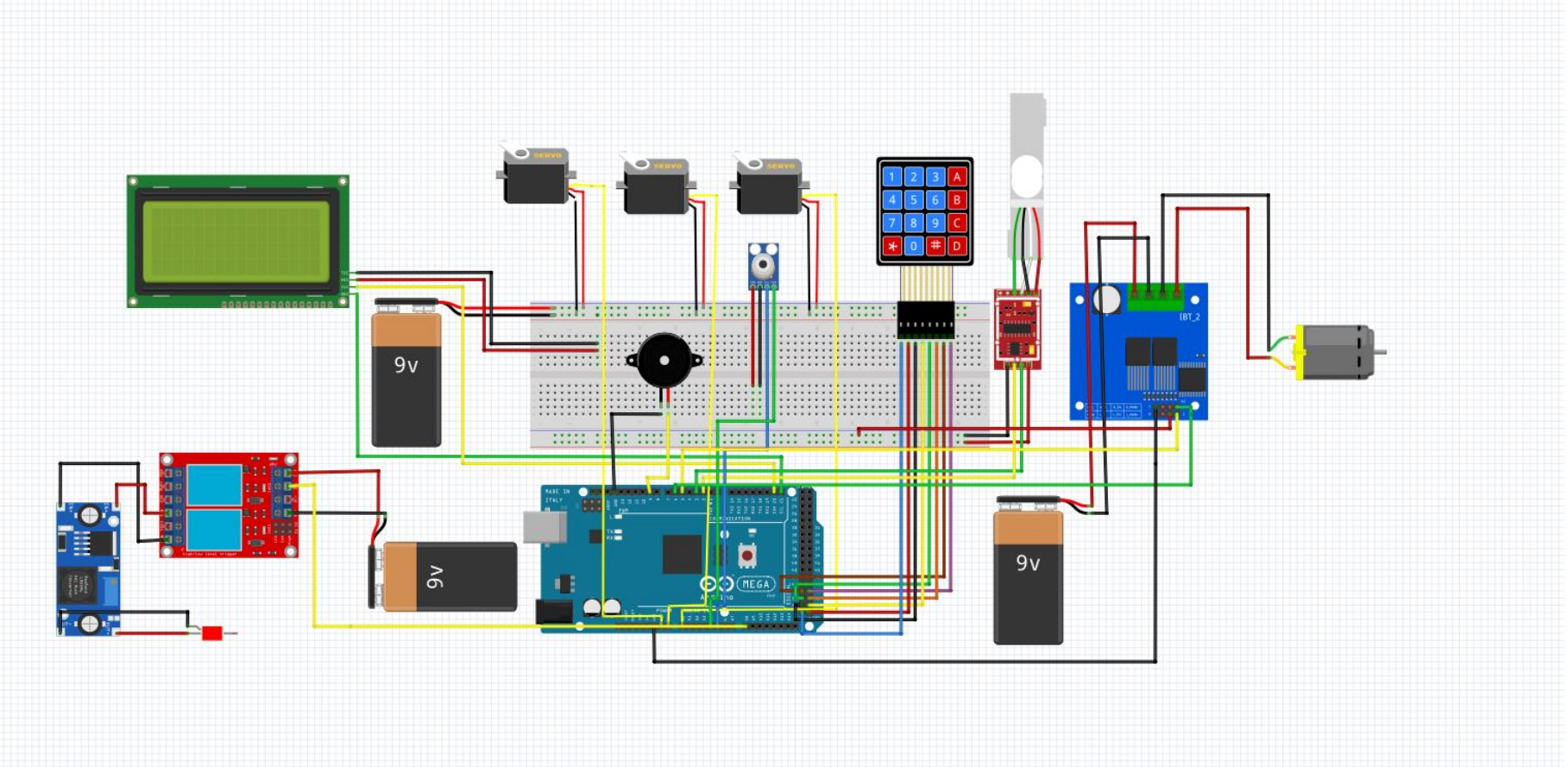
SKEMA KELISTRIKAN MESIN PENYANGRAI KOPI OTOMATIS



NO	KECEPATAN MOTOR 30						60					
	160		190		225		160		190		225	
	X1	$X1^2$	X2	$X2^2$	X3	$X3^2$	X4	$X4^2$	X5	$X5^2$	X6	$X6^2$
1	13.450	180.903	15.700	246.490	45.350	2056.623	11.900	141.610	13.950	194.603	32.000	1024.000
2	13.550	183.603	18.350	336.723	44.600	1989.160	10.950	119.903	12.550	157.503	35.850	1285.223
3	14.100	198.810	16.800	282.240	44.900	2016.010	11.350	128.823	13.550	183.603	32.900	1082.410
ΣX	41.100	563.315	50.850	865.453	134.850	6061.793	34.200	390.335	40.050	535.708	100.750	3391.633
$(\Sigma X)^2$	1689.210	317323.789	2585.723	749008.030	18184.523	36745328.313	1169.640	152361.412	1604.003	286982.526	10150.563	11503171.015

Perhitungan	X1	X2	X3	X4	X5	X6	Total
N	3	3	3	3	3	3	18
ΣX	41.1	50.85	134.85	34.2	40.05	100.75	401.8
ΣXR	13.70000000	16.95	44.95	11.4	13.35	33.58333333	133.9333333
ΣX1.4	41.1			34.2			75.3
ΣX2.5		50.85			40.05		90.9
ΣX3.6			134.85			100.75	235.6
$\Sigma(X^2)$	563.315	865.453	6061.793	390.335	535.708	3391.633	11808.237

Keterangan :

ΣXR = $\frac{\Sigma X}{(N_x)}$

Kodingan Penyangrai Kopi Otomatis

```
#include "max6675.h"
#include "Wire.h"
#include "HX711.h"
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
Servo servoValve;
Servo servoLoad;
Servo servoStove;
HX711 scale;
LiquidCrystal_I2C lcd(0x27, 20, 4);

//=================================== Setting 4 Parameter
int Suhu;
int Speed;
int Timer;
int Load;
int jeda = 5000;
//=========================================================

int timeNow = 0;
int Loading = 0;
int Celcius = 0;
float calibration_factor = -400;
int units;
int thermoDO = 2; //S0
int thermoCS = 3; //cs
```

```
int thermoCLK = 4; //sck
int pemantik = 17;
int relayON = LOW; //relay nyala
int relayOFF = HIGH; //relay mati

MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);
#define pwm1 24
#define pwm2 26

void setup() {
  //============== LCD I2C ==================================
  // Print a message to the LCD.
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("   COFFEE ROASTER   ");
  lcd.setCursor(0, 1);
  lcd.print("      OTOMATIS      ");
  lcd.setCursor(0, 2);
  lcd.print("         BY         ");
  lcd.setCursor(0, 3);
  lcd.print("    SURYA  PALAS     ");
  //============== MOTOR
=====================================
  pinMode(pwm1, OUTPUT);
  pinMode(pwm2, OUTPUT);

  analogWrite(pwm1, 0);
```

```
  analogWrite(pwm2, 0);
  Serial.begin(9600);
  //=============== LoadCell ===================================
  scale.begin(A0, A1);
  scale.set_scale();
  scale.tare();
  long zero_factor = scale.read_average();
  //=============== Thermocouple ================================
  Serial.println("MAX6675 test");
  delay (500);

  //===============Pemantik=====================================
  pinMode (pemantik, OUTPUT);
  digitalWrite(pemantik, relayOFF);


//===============SERVO=========================================
=
  servoStove.attach(7);
  servoLoad.attach(6);
  servoValve.attach(5);

  servoStove.write(180);
  servoValve.write(0);
  servoLoad.write(0);
  delay(3000);
}

void loop() {
```

```
A:
  analogWrite(pwm1, 0);
  analogWrite(pwm2, 0);
  delay(2000);
  digitalWrite(pemantik, relayOFF);
  lcd.clear();
  lcd.noBlink();

  lcd.setCursor(0, 0);
  lcd.print("Suhu : ");
  lcd.setCursor(9, 0);
  lcd.print(Suhu);
  lcd.setCursor(0, 1);
  lcd.print("Waktu :");
  lcd.setCursor(9, 1);
  lcd.print(Timer);
  lcd.setCursor(0, 2);
  lcd.print("Load  :");
  lcd.setCursor(9, 2);
  lcd.print(Load);
  lcd.setCursor(0, 3);
  lcd.print("Speed :");
  lcd.setCursor(9, 3);
  lcd.print(Speed);

  if (Loading == 0) {
    Loadcell();
    if (units < Load) {
```

```
  lcd.clear();
  lcd.noBlink();
  lcd.setCursor(0, 0);
  lcd.print("Suhu : ");
  lcd.setCursor(9, 0);
  lcd.print(Suhu);
  lcd.setCursor(0, 1);
  lcd.print("Waktu :");
  lcd.setCursor(9, 1);
  lcd.print(Timer);
  lcd.setCursor(0, 2);
  lcd.print("Load  :");
  lcd.setCursor(9, 2);
  lcd.print(units);
  lcd.setCursor(17, 2);
  lcd.print("Gr");
  lcd.setCursor(0, 3);
  lcd.print("Speed :");
  lcd.setCursor(9, 3);
  lcd.print(Speed);

  servoValve.write(0);
  servoLoad.write(0);
  servoStove.write(180);
  Loading = 0;
 }
 if (units >= Load) {
  lcd.clear();
```

```
    lcd.noBlink();
    lcd.setCursor(0, 0);
    lcd.print("Suhu : ");
    lcd.setCursor(9, 0);
    lcd.print(Suhu);
    lcd.setCursor(0, 1);
    lcd.print("Waktu :");
    lcd.setCursor(9, 1);
    lcd.print(Timer);
    lcd.setCursor(0, 2);
    lcd.print("Load  :");
    lcd.setCursor(9, 2);
    lcd.print(units);
    lcd.setCursor(17, 2);
    lcd.print("Gr");
    lcd.setCursor(0, 3);
    lcd.print("Speed :");
    lcd.setCursor(9, 3);
    lcd.print(Speed);
    servoValve.write(90);
    delay(5000);
    servoLoad.write(90);
    Loading = 1;
  }
}
if (Loading == 1) {
  delay(jeda);
  servoValve.write(0);
```

```
    servoLoad.write(0);
    servoStove.write(0);
    delay(2000);
    digitalWrite(pemantik, relayON);
    delay(2000);
    digitalWrite(pemantik, relayOFF);
    delay(2000);
    //    Motor();
    Penyangraian();
  }
}

void Penyangraian() {
  Motor();
  readSuhu();
  lcd.clear();
  lcd.noBlink();
  lcd.setCursor(0, 0);
  lcd.print("Suhu : ");
  lcd.setCursor(9, 0);
  lcd.print(Celcius);
  lcd.setCursor(0, 1);
  lcd.print("Waktu :");
  lcd.setCursor(9, 1);
  lcd.print(Timer);
  lcd.setCursor(0, 2);
  lcd.print("Load  :");
  lcd.setCursor(9, 2);
```

```
lcd.print(units);
lcd.setCursor(17, 2);
lcd.print("Gr");
lcd.setCursor(0, 3);
lcd.print("Speed :");
lcd.setCursor(9, 3);
lcd.print(Speed);
int SudutKompor = constrain(SudutKompor, 0, 180);
if (Celcius >= Suhu) {
 SudutKompor++;
 int startingTime = millis();
 if ((startingTime + (Timer * 60000000)) > timeNow) {
  servoStove.write(180);

  lcd.setCursor(0, 0);
  lcd.print("      ROASTING      ");
  lcd.setCursor(0, 1);
  lcd.print("       SELESAI      ");
  lcd.setCursor(0, 2);
  lcd.print("         BY         ");
  lcd.setCursor(0, 3);
  lcd.print("    SURYA  PALAS     ");

  loop();
 }
 else {
  timeNow = millis();
```

```
    lcd.clear();
    lcd.noBlink();
    lcd.setCursor(0, 0);
    lcd.print("Suhu : ");
    lcd.setCursor(9, 0);
    lcd.print(Celcius);
    lcd.setCursor(0, 1);
    lcd.print("Waktu :");
    lcd.setCursor(9, 1);
    lcd.print((startingTime + (Timer * 60000000)) - timeNow);
    lcd.setCursor(0, 2);
    lcd.print("Load  :");
    lcd.setCursor(9, 2);
    lcd.print(units);
    lcd.setCursor(17, 2);
    lcd.print("Gr");
    lcd.setCursor(0, 3);
    lcd.print("Speed :");
    lcd.setCursor(9, 3);
    lcd.print(Speed);
   }
  }
  if (Celcius < Suhu) {
   SudutKompor--;
  }
  servoStove.write(SudutKompor);
}
```

```
void Loadcell() {
  scale.set_scale(calibration_factor);
  units = scale.get_units(), 10;
  if (units < 0)
  {
    units = 0;
  }
  Serial.println(units);
  delay(500);
}

void Motor() {
  analogWrite(pwm1, 0);
  analogWrite(pwm2, 150);
}

void readSuhu() {
  Celcius = thermocouple.readCelsius();
  Serial.print("C = ");
  Serial.println(thermocouple.readCelsius());
  Serial.print("F = ");
  Serial.println(thermocouple.readFahrenheit());
  delay(1000);
}
```