

LAMPIRAN A

Program ARDUINO

```
#include <HX711_ADC.h>
```

```
#include <EEPROM.h>
```

```
// Pin assignment untuk HX711
```

```
const int HX711_dout = 4; // MCU > pin dout HX711
```

```
const int HX711_sck = 5; // MCU > pin sck HX711
```

```
// Pin Relay
```

```
const int relay1Pin = 6; // Relay 1 terhubung ke pin 6
```

```
const int relay2Pin = 7; // Relay 2 terhubung ke pin 7
```

```
const int relay3Pin = 8; // Relay 3 terhubung ke pin 8
```

```
const int relay4Pin = 9; // Relay 3 terhubung ke pin 9
```

```
// Konstruktor HX711
```

```
HX711_ADC LoadCell(HX711_dout, HX711_sck);
```

```
const int calVal_eeepromAdress = 0;
```

```
unsigned long t = 0;
```

```
void setup() {
```

```
    Serial.begin(57600);
```

```
delay(10);

Serial.println();

Serial.println("Memulai...");

// Membaca nilai kalibrasi dari EEPROM

float savedCalFactor;

EEPROM.get(calVal_eeepromAdress, savedCalFactor);

// Jika tidak ada nilai yang disimpan di EEPROM, gunakan nilai
default 1.0

if (isnan(savedCalFactor) || savedCalFactor == 0) {

    savedCalFactor = 1.0;

}

// Inisialisasi load cell

LoadCell.begin();

unsigned long stabilizingtime = 2000; // Waktu stabilisasi setelah
power-up

boolean _tare = true; // Lakukan tare saat memulai

LoadCell.start(stabilizingtime, _tare);

if (LoadCell.getTareTimeoutFlag() ||
LoadCell.getSignalTimeoutFlag()) {
```

```

Serial.println("Timeout, periksa pengkabelan MCU>HX711 dan
penunjukan pin");

while (1);

} else {

LoadCell.setCalFactor(savedCalFactor); // Gunakan nilai kalibrasi
yang disimpan

Serial.print("Nilai kalibrasi digunakan: ");

Serial.println(savedCalFactor);

Serial.println("Startup selesai");

}

// Pengaturan pin relay

pinMode(relay1Pin, OUTPUT);
pinMode(relay2Pin, OUTPUT);
pinMode(relay3Pin, OUTPUT);
pinMode(relay4Pin, OUTPUT);

// Status awal relay

digitalWrite(relay1Pin, HIGH);
digitalWrite(relay2Pin, HIGH);
digitalWrite(relay3Pin, HIGH);
digitalWrite(relay4Pin, HIGH);

```

```

while (!LoadCell.update());

// kalibrasi(); // Hanya dilakukan jika perlu kalibrasi baru
}

void loop() {

static boolean newDataReady = 0;

const int serialPrintInterval = 0; // Interval untuk cetak serial

// Periksa data baru/mulai konversi berikutnya
if (LoadCell.update()) newDataReady = true;

// Dapatkan nilai yang telah dihaluskan dari dataset
if (newDataReady) {
if (millis() > t + serialPrintInterval) {
float berat = LoadCell.getData();
Serial.print("Nilai output Load_cell: ");
Serial.println(berat);

// Kontrol relay berdasarkan berat
if (berat >= 200 && berat < 1200) {
digitalWrite(relay1Pin, LOW); // Relay 1 ON
digitalWrite(relay2Pin, HIGH); // Relay 2 OFF
}
}
}
}

```



```
digitalWrite(relay3Pin, HIGH); // Relay 3 OFF
} else if (berat >= 1200 && berat < 1350) {
digitalWrite(relay1Pin, HIGH); // Relay 1 OFF
digitalWrite(relay2Pin, LOW); // Relay 2 ON
digitalWrite(relay3Pin, HIGH); // Relay 3 OFF
} else if (berat >= 1350 && berat < 2000) {
digitalWrite(relay1Pin, HIGH); // Relay 1 OFF
digitalWrite(relay2Pin, HIGH); // Relay 2 OFF
digitalWrite(relay3Pin, LOW); // Relay 3 ON
} else if (berat >= 2000) {
digitalWrite(relay1Pin, HIGH); // Relay 1 OFF
digitalWrite(relay2Pin, HIGH); // Relay 2 OFF
digitalWrite(relay3Pin, HIGH); // Relay 3 ON
} else if (berat <= 0) {
digitalWrite(relay1Pin, HIGH); // Relay 1 OFF
digitalWrite(relay2Pin, HIGH); // Relay 2 OFF
digitalWrite(relay3Pin, HIGH); // Relay 3 ON
digitalWrite(relay4Pin, LOW); // Relay 4 ON
}
}
```

```

    newDataReady = 0;

    t = millis();
}
}

// Terima perintah dari terminal serial
if (Serial.available() > 0) {
    char inByte = Serial.read();

    if (inByte == 't') LoadCell.tareNoDelay(); // Tare
    else if (inByte == 'r') kalibrasi(); // Kalibrasi
    else if (inByte == 'c') ubahFaktorKalibrasiTersimpan(); // Ubah nilai
    kalibrasi secara manual
}

// Periksa jika operasi tare terakhir telah selesai
if (LoadCell.getTareStatus() == true) {
    Serial.println("Tare selesai");
}
}

void kalibrasi() {
    Serial.println("*");
    Serial.println("Mulai kalibrasi:");
}

```

```
Serial.println("Letakkan load cell di permukaan yang stabil dan rata.");  
Serial.println("Hapus semua beban yang diterapkan pada load cell.");  
Serial.println("Kirim 't' dari monitor serial untuk mengatur offset tare.");
```

```
boolean _resume = false;  
while (_resume == false) {  
    LoadCell.update();  
    if (Serial.available() > 0) {  
        char inByte = Serial.read();  
        if (inByte == 't') LoadCell.tareNoDelay();  
    }  
    if (LoadCell.getTareStatus() == true) {  
        Serial.println("Tare selesai");  
        _resume = true;  
    }  
}
```

```
Serial.println("Sekarang, letakkan beban yang diketahui di load cell.");  
Serial.println("Kemudian kirim berat dari beban ini (misalnya 100.0) dari monitor serial.");
```

```
float known_mass = 0;
```



```

_resume = false;
while (_resume == false) {
  LoadCell.update();
  if (Serial.available() > 0) {
    known_mass = Serial.parseFloat();
    if (known_mass != 0) {
      Serial.print("Beban yang diketahui adalah: ");
      Serial.println(known_mass);
      _resume = true;
    }
  }
}

LoadCell.refreshDataSet(); // Segarkan dataset untuk memastikan
pengukuran yang akurat

float nilaiKalibrasiBaru = LoadCell.getNewCalibration(known_mass);
// Dapatkan nilai kalibrasi baru

Serial.print("Nilai kalibrasi baru telah disetel ke: ");
Serial.print(nilaiKalibrasiBaru);

Serial.println(", gunakan ini sebagai nilai kalibrasi (calFactor) di sketsa
proyek Anda.");

Serial.print("Simpan nilai ini ke alamat EEPROM ");

```

```

Serial.print(calVal_eeepromAdress);

Serial.println("? y/n");

_resume = false;
while (_resume == false) {
  if (Serial.available() > 0) {
    char inByte = Serial.read();
    if (inByte == 'y') {
#if defined(ESP8266)|| defined(ESP32)
      EEPROM.begin(512);
#endif
      EEPROM.put(calVal_eeepromAdress, nilaiKalibrasiBaru);
#if defined(ESP8266)|| defined(ESP32)
      EEPROM.commit();
#endif
      EEPROM.get(calVal_eeepromAdress, nilaiKalibrasiBaru);
      Serial.print("Nilai ");
      Serial.print(nilaiKalibrasiBaru);
      Serial.print(" disimpan ke alamat EEPROM: ");
      Serial.println(calVal_eeepromAdress);
      _resume = true;
    } else if (inByte == 'n') {

```

```

Serial.println("Nilai tidak disimpan ke EEPROM");

_resume = true;
}
}
}

Serial.println("Kalibrasi selesai");

Serial.println("*");

Serial.println("Untuk mengkalibrasi ulang, kirim 'r' dari monitor
serial.");

Serial.println("Untuk mengedit nilai kalibrasi secara manual, kirim 'c'
dari monitor serial.");

Serial.println("*");
}

void ubahFaktorKalibrasiTersimpan() {
float nilaiKalibrasiLama = LoadCell.getCalFactor();

boolean _resume = false;

Serial.println("*");

Serial.print("Nilai saat ini adalah: ");

Serial.println(nilaiKalibrasiLama);

Serial.println("Sekarang, kirim nilai baru dari monitor serial, misalnya
696.0");
}

```

```

float nilaiKalibrasiBaru;
while (_resume == false) {
  if (Serial.available() > 0) {
    nilaiKalibrasiBaru = Serial.parseFloat();
    if (nilaiKalibrasiBaru != 0) {
      Serial.print("Nilai kalibrasi baru adalah: ");
      Serial.println(nilaiKalibrasiBaru);
      LoadCell.setCalFactor(nilaiKalibrasiBaru);
      _resume = true;
    }
  }
}
_resume = false;
Serial.print("Simpan nilai ini ke alamat EEPROM ");
Serial.print(calVal_eeepromAdress);
Serial.println("? y/n");
while (_resume == false) {
  if (Serial.available() > 0) {
    char inByte = Serial.read();
    if (inByte == 'y') {
#ifdef ESP8266 || defined(ESP32)
      EEPROM.begin(512);
#endif
    }
  }
}

```

```

#endif

    EEPROM.put(calVal_eeepromAdress, nilaiKalibrasiBaru);
#if defined(ESP8266)|| defined(ESP32)

    EEPROM.commit();
#endif

    EEPROM.get(calVal_eeepromAdress, nilaiKalibrasiBaru);
    Serial.print("Nilai ");
    Serial.print(nilaiKalibrasiBaru);
    Serial.print(" disimpan ke alamat EEPROM: ");
    Serial.println(calVal_eeepromAdress);
    _resume = true;
} else if (inByte == 'n') {
    Serial.println("Nilai tidak disimpan ke EEPROM");
    _resume = true;
}
}
}
Serial.println("Ubah nilai kalibrasi selesai");
Serial.println("*");
}

```

BIODATA PENULIS



Nama : Amar Baskara
Tempat/Tanggal Lahir : Cilacap, 6 April 2003
Alamat : Jl. Benggala Timur 007/009 Mertasinga,
kec. Cilacap Utara - 53232
Email : baskaraamar@gmail.com
Telepon/HP : 088233183047
Hobi : Berjualan
Motto : Siapapun bisa jadi apapun banyak orang
sukses berawal dari Helper

Riwayat Pendidikan

- SD Negeri 06 Mertasinga Tahun 2008-2014
- SMP Negeri 7 Cilacap Tahun 2014-2017
- SMA Budi Utomo Jombang Tahun 2017-2020
- Politeknik Negeri Cilacap Tahun 2021-2024
Prodi D3 Teknik Listrik

Penulis telah mengikuti seminar hasil pada tanggal 2 Agustus 2024 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md).