

```
#define trigger_kiri 22
#define echo_kiri 23
#define trigger_tengah 24
#define echo_tengah 25
#define trigger_kanan 26
#define echo_kanan 27

#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR 0x27 //LCD (jika tidak bisa 3F ganti 27F)
#define BACKLIGHT_PIN 3
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7

LiquidCrystal_I2C lcd(I2C_ADDR, En_pin, Rw_pin, Rs_pin, D4_pin, D5_pin,
D6_pin, D7_pin);

#include "HX711.h"
HX711 scale(A1, A0);
```

```
#include <SoftwareSerial.h>
#include "DFPlayer_Mini_Mp3.h"

long jarak_kiri;
long jarak_kanan;
long jarak_tengah;

float berat;
float berat_rata = 0 ;
long tinggi_max = 190;
long tinggi_diambil = 0;
long tinggi_sebelumnya = 0;

int kounter = 0;
int kounter_berat = 0;

int range_error = 2;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial3.begin(9600);
  pinMode(trigger_kiri, OUTPUT);
  pinMode(echo_kiri, INPUT);
  pinMode(trigger_tengah, OUTPUT);
  pinMode(echo_tengah, INPUT);
```

```
pinMode(trigger_kanan, OUTPUT);
pinMode(echo_kanan, INPUT);
mp3_set_serial (Serial3); //set Serial for DFPlayer-mini mp3 module
mp3_set_volume (100);
lcd.begin (16, 2);
lcd.setBacklightPin(BACKLIGHT_PIN, POSITIVE); //LCD
lcd.setBacklight(HIGH);
lcd.home ();
lcd.print("Kalibrasi Sensor");
lcd.setCursor(0, 1);
lcd.print("Harap Tunggu");

Serial.println("HX711 Demo");

Serial.println("Before setting up the scale:");
Serial.print("read: \t\t");
Serial.println(scale.read()); // print a raw reading from the ADC

Serial.print("read average: \t\t");
Serial.println(scale.read_average(20)); // print the average of 20 readings from
the ADC

Serial.print("get value: \t\t");
Serial.println(scale.get_value(5)); // print the average of 5 readings from the
ADC minus the tare weight (not set yet)

Serial.print("get units: \t\t");
```

```
Serial.println(scale.get_units(5), 1); // print the average of 5 readings from the
ADC minus tare weight (not set) divided

// by the SCALE parameter (not set yet)

scale.set_scale(25500.f);           // this value is obtained by calibrating the
scale with known weights; see the README for details

scale.tare();           // reset the scale to 0

Serial.println("After setting up the scale:");

Serial.print("read: \t\t");

Serial.println(scale.read());           // print a raw reading from the ADC

Serial.print("read average: \t\t");

Serial.println(scale.read_average(20)); // print the average of 20 readings
from the ADC

Serial.print("get value: \t\t");

Serial.println(scale.get_value(5)); // print the average of 5 readings from the
ADC minus the tare weight, set with tare()

Serial.print("get units: \t\t");

Serial.println(scale.get_units(5), 1); // print the average of 5 readings from
the ADC minus tare weight, divided

// by the SCALE parameter set with set_scale

Serial.println("Readings:");

lcd.clear();

lcd.home ();
```

```
lcd.print("Silahkan Coba");

}

void loop() {
  // put your main code here, to run repeatedly:
  sKiri();
  sTengah();
  sKanan();

  tinggi_diambil = jarak_kiri;
  if (tinggi_diambil > jarak_tengah)tinggi_diambil = jarak_tengah;
  if (tinggi_diambil > jarak_kanan)tinggi_diambil = jarak_kanan;
  if (tinggi_diambil < 100) {
    //ada orang
    if (tinggi_sebelumnya == 0) {
      tinggi_sebelumnya = tinggi_diambil;
    } else {
      if (tinggi_diambil - tinggi_sebelumnya <= range_error) {
        kounter++;
        if (kounter >= 6) {
          lcd.setCursor(0, 0);
          lcd.print("Kalkulasi");
          berat_rata = 0;
          lcd.setCursor(0, 1);
        }
      }
    }
  }
}
```

```

for (int i = 0 ; i < 4 ; i++) {
    lcd.print(">>>>");
    baca_berat();
    if (berat_rata == 0){
        berat_rata = berat;
    }else{
        berat_rata += berat;
    }
    //delay(100);
}
berat_rata/=4;
float tinggi_orang = (float)(tinggi_max - tinggi_diambil)/100;
float BMI = berat_rata / (tinggi_orang * tinggi_orang);
Serial.print("Tinggi = ");Serial.println(tinggi_orang);
Serial.print("BMI = ");Serial.println(BMI);
lcd.clear();
lcd.home();
lcd.print("BMI = ");lcd.print(BMI);
delay(1000);
if (BMI <= 17){
    kurus();
}
else if (BMI > 17 && BMI <=27){
    sedang();
}
else if (BMI > 27){
    gemuk();
}
}

```

```
kounter = 0;

lcd.clear();
lcd.home();
lcd.print("Silahkan Coba");

}

} else {
    kounter = 0;
}

tinggi_sebelumnya = tinggi_diambil;
}

} else {
    //katek wonk
    kounter = 0;
}

delay(100);

}

void sKiri() {
    digitalWrite(trigger_kiri, 0);
```

```
delayMicroseconds(2);
digitalWrite(trigger_kiri, 1);

delayMicroseconds(5);
digitalWrite(trigger_kiri, 0);

jarak_kiri = pulseIn (echo_kiri, HIGH);
jarak_kiri = jarak_kiri / 58;

Serial.print(jarak_kiri); Serial.print(" ");
}

void sTengah() {
  digitalWrite(trigger_tengah, 0);
  delayMicroseconds(2);
  digitalWrite(trigger_tengah, 1);
  delayMicroseconds(5);
  digitalWrite(trigger_tengah, 0);

  jarak_tengah = pulseIn (echo_tengah, HIGH);
  jarak_tengah = jarak_tengah / 58;
  Serial.print(jarak_tengah); Serial.print(" ");
}

void sKanan() {
  digitalWrite(trigger_kanan, 0);
```

```

delayMicroseconds(2);
digitalWrite(trigger_kanan, 1);
delayMicroseconds(5);
digitalWrite(trigger_kanan, 0);

jarak_kanan = pulseIn (echo_kanan, HIGH);
jarak_kanan = jarak_kanan / 58;
Serial.println(jarak_kanan);
}

void baca_berat() {
  Serial.print("read average: \t\t");
  Serial.println(scale.read_average(20)); // print the average of 20 readings from
the ADC
  Serial.print("one reading:\t");
  Serial.print(scale.get_units(), 1);
  Serial.print("\t| average:\t");
  berat = scale.get_units(10);
  Serial.println(berat, 1);
  scale.power_down(); // put the ADC in sleep mode
  delay(500);
  scale.power_up();
}

void kurus() {
  lcd.clear();
  lcd.home();
}

```

```
lcd.print("Anda Kurus");  
mp3_play (1); //play 0001.mp3  
delay (10000); //10 sec, time delay to allow 0001.mp3 to finish playing  
}  
void sedang() {  
  lcd.clear();  
  lcd.home();  
  lcd.print("Anda Normal");  
  mp3_play (2); //play 0001.mp3  
  delay (10000); //10 sec, time delay to allow 0001.mp3 to finish playing  
}  
void gemuk() {  
  lcd.clear();  
  lcd.home();  
  lcd.print("Anda Obesitas");  
  mp3_play (3); //play 0001.mp3  
  delay (10000); //10 sec, time delay to allow 0001.mp3 to finish playing  
}
```