LAMPIRAN

Listing Program Keseluruhan

```
#include <LiquidCrystal.h>
const int rs = 2, en = 3, d4 = 5, d5 = 6, d6 = 7, d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void (*reset)(void)=0x0000;
#define sw1 A0    //menetapkan pin A0 sbg switch 1
#define sw2 A1
#define buz 10    //menetapkan pin 10 sbg buzzer
#define sens A5
#define bl 9
#define  MQ_PIN    (5)     //menetapkan pin mana yg dipakai
#define  RL_VALUE  (10)    //menetapkan RL dalam kilo ohm
#define              RO_CLEAN_AIR_FACTOR              (21)
//RO_CLEAR_AIR_FACTOR=(resistansi    sensor    di    udara
bersih)/RO, dilihat dari grafik di datasheet
#define   CALIBARAION_SAMPLE_TIMES    (100)     //menetapkan
berapa banyak sampel yg diambil dlm fase kalibrasi
#define   CALIBRATION_SAMPLE_INTERVAL   (100)    //menetapkan
interval waktu masing-masing sampel dlm fase kalirasi yaitu
100 ms (0,1 s)
#define  READ_SAMPLE_INTERVAL    (50)    //menetapkan berapa
banyak sampel yg diambil dalam operasi normal
#define  READ_SAMPLE_TIMES    (5)     //menetapkan interval
waktu masing-masing sampel dalam opereasi normal
#define        GAS_ALCOHOL                       (0)
float          AlcoholCurve[3]  =  {2, 2,-1.51};
float          Ro          =  10;                   //Ro is
initialized to 10 kilo ohms
void beep(){
   digitalWrite(buz,HIGH); //fungsi untuk menjadikan buzzer
berlogika high (1) 5volt
   delay(100);
   digitalWrite(buz,LOW);  //fungsi  untuk  menjadikan  buzzer
berlogika low (0) 0volt
```

```
   delay(500);  }
void setup() { //sebuah fungsi yg berjalan pertama kali ketika program dijalankan
  pinMode(sw1,INPUT_PULLUP);
  pinMode(sw2,INPUT_PULLUP);
  pinMode(bl,OUTPUT);
  pinMode(buz,OUTPUT);
  digitalWrite(bl,HIGH);
  lcd.begin(16, 2);
  lcd.setCursor(4,0);
  lcd.print("FARCHANA");
  lcd.setCursor(2,1);
  lcd.print("20153010026");
  delay(2000);
  Ro = MQCalibration(MQ_PIN); //ketika nulis ini, berarti memanggil program yg dibawah (proses warming up)
}
double ppm,mgcc,BAC;
void tampil_hasil(){
  lcd.setCursor(0,0);
  lcd.print(mgcc,1);
  lcd.print(" g/l  ");
  lcd.setCursor(0,1);
  lcd.print(BAC,2);
  lcd.print(" %BAC");
  if (mgcc<0.8){
    lcd.setCursor(9,0);
    lcd.print("Rendah");
    }
  else if (mgcc>0.8){
    lcd.setCursor(9,0);
    lcd.print("Tinggi");
    beep();
   }
  }
void loop() {
  awal:
   float uplimit=0;
   lcd.clear();
  while(digitalRead(sw2)==1){
     lcd.setCursor(0,0);
     lcd.print("PENDETEKSI KADAR");
     lcd.setCursor(4,1);
     lcd.print("ALKOHOL");
    }
  lcd.clear();
  beep();
  for (int ul=40;ul>0;ul--){
    ppm=MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_ALCOHOL);
    if (ppm>uplimit){uplimit=ppm;}
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("00");
    lcd.print(ul/4);
    delay(100);
  }
  beep();
  lcd.clear();
  mgcc= (0.0038*ppm) + 5E-16;
  BAC=mgcc/10.0;
  if (ppm<26){mgcc=0;BAC=0;}
  tampil_hasil();
  delay(1000);
 while(1){
  tampil_hasil();
 if (digitalRead(sw1)==0){
    beep();
    reset(); }
 }
}

//MQResistanceCalculation
float MQResistanceCalculation(int raw_adc)
{
  return ( ((float)RL_VALUE*(1023-raw_adc)/raw_adc));
}

//MQCalibration
float MQCalibration(int mq_pin)
{
  int i;
  float val=0;
  for (i=0;i<CALIBARAION_SAMPLE_TIMES;i++) {
//take multiple samples
    val += MQResistanceCalculation(analogRead(mq_pin));
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("   WARMING UP   ");
    lcd.setCursor(0,1);
    lcd.print(i+1);
    lcd.print(" %");
    delay(CALIBRATION_SAMPLE_INTERVAL);
  }
  val=val/CALIBARAION_SAMPLE_TIMES;
//calculate the average value

  val=val/RO_CLEAN_AIR_FACTOR;
//divided by RO_CLEAN_AIR_FACTOR yields the Ro
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("   WARMING UP   ");
    lcd.setCursor(0,1);
    lcd.print("   COMPLETED    ");
    delay(1000);
    return val;
}

//MQRead
float MQRead(int mq_pin)
{
  int i;
  float rs=0;
  for (i=0;i<READ_SAMPLE_TIMES;i++) {
    rs += MQResistanceCalculation(analogRead(mq_pin));
    delay(READ_SAMPLE_INTERVAL);
  }
  rs = rs/READ_SAMPLE_TIMES;
  return rs;
}

//MQGetGasPercentage
int MQGetGasPercentage(float rs_ro_ratio, int gas_id)
{
  if ( gas_id == GAS_ALCOHOL) {
     return MQGetPercentage(rs_ro_ratio,AlcoholCurve);
  }
  return 0;
}

//MQGetPercentage
int  MQGetPercentage(float rs_ro_ratio, float *pcurve)
{
  return  (pow(10,  (((log(rs_ro_ratio)-pcurve[1])/pcurve[2])
+ pcurve[0])));
}
```