

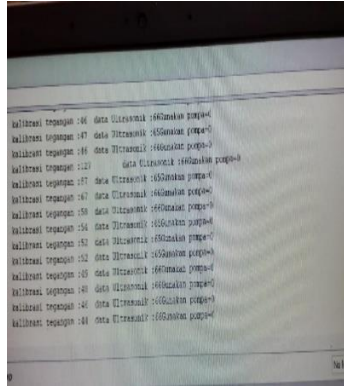
LAMPIRAN A

a. Pembuatan Box Panel 50x50x85 cm



b. Kalibrasi Sensor



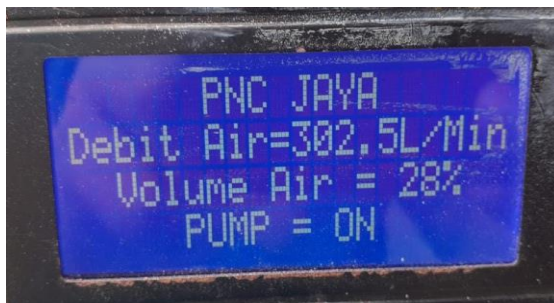


c. Pengukuran Tegangan, Arus, Debit, dan Volume Air





d. Pembacaan pengukuran Tampilan LCD



LAMPIRAN B

Berikut kode pemograman

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define adcTeg A0
#define relay 10
#define echoPin1 9
#define initPin1 8
#define ADClowBat 787 // batas bawah tegangan batre digunakan
#define ADCfullBat 797 // batas atas tegangan batre mulai digunakan
saat adc 791 yaitu 24v
bool gunakan_pompa = true;
unsigned long current_ultrasonik;
unsigned long flagRun;
int dataTeg;
float voltage;
float liter;
int distance1 = 0;
int jarakAir;
byte statusLed = 13;
byte sensorInterrupt = 0;
byte sensorPin = 2;
String kondisi = "ON";
volatile long pulseCount;
```

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

```
void setup()
```

```
{
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  pinMode(initPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(A0, INPUT);
  pinMode(relay, OUTPUT);
  digitalWrite(relay, LOW);
  pinMode(statusLed, OUTPUT);
  digitalWrite(statusLed, HIGH);
```

```

pinMode(sensorPin, INPUT);
digitalWrite(sensorPin, HIGH);
pulseCount    = 0;
attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
digitalWrite(relay, HIGH);

}

/**
 * Main program loop
 */
float totalMilliLitres_kalibrasi;

void loop() {
  if (millis() - current_ultrasonik > 100) {
    current_ultrasonik = millis();
    distance1 = bacaUltrasonik();
  }
  ////////////////////////////////////////////////////////////////////nilai depan//////////////////////////////////////////////////////////////////nilai asli
  liter//////////////////////////////////////////////////////////////////
  liter = maping(pulseCount, 0, 4897, 0, 15);
  unsigned long penambahan = 0;
  int sampling = 1000;
  for (int i = 0; i < sampling; i++) {
    penambahan += analogRead(adcTeg);
  }
  dataTeg = penambahan / sampling;
  ////////////////////////////////////////////////////////////////////kalib tegangan//////////////////////////////////////////////////////////////////nilai asli
  tegangan//////////////////////////////////////////////////////////////////
  // voltage = maping(dataTeg, 0, 793.0, 0.0, 24.2);
  // voltage = hitungTeg(dataTeg);
  ////////////////////////////////////////////////////////////////////tinggi//
  jarakAir = map(distance1 - 20, 119, 0, 0, 100); //140 diganti tinggi
  dasar ke sensor
  Serial.print("kalibrasi waterflow :");
  Serial.print(pulseCount);
  Serial.print("\t kalibrasi tegangan :");
  Serial.print(dataTeg);
  Serial.print("\t data Ultrasonik :");

```

```

Serial.print(distance1);
Serial.println("Gunakan pompa=" + String(gunakan_pompa));
//=====LOGIKA UNTUK
POMPA=====

if (dataTeg > ADCfullBat) {
  gunakan_pompa = true;
}
else if (dataTeg < ADClowBat) {
  gunakan_pompa = false;
}

if (gunakan_pompa == true) {
  if (jarakAir <= 75) { // batas atas tandon air
    digitalWrite(relay, HIGH);
    kondisi = "ON ";
  }
  else if (jarakAir >= 100) {
    digitalWrite(relay, LOW);
    kondisi = "OFF";
  }
}
else {
  digitalWrite(relay, LOW);
  kondisi = "OFF";
}
// printing();
printLcd();

//=====
=====
}

void printLcd() {
  if ((millis() - flagRun) > 1500) {
    lcd.init();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" Baterai ADC = ");
  }
}

```

```

    lcd.print(dataTeg);
    lcd.setCursor(0, 1);
    lcd.print("  Liter = " + String(liter) + "L ");
    lcd.setCursor(0, 2);
    lcd.print("  Volume Air = " + String(jarakAir) + "% ");
    lcd.setCursor(0, 3);
    lcd.print("  PUMP = ");
    lcd.print(kondisi);
    flagRun = millis();
  }
}

void printing() {
  Serial.print("ACD_Teg=" + String(dataTeg));
  Serial.print("Flow rate: ");
  Serial.print(" Liter=");
  Serial.print(liter);
  Serial.print("L");
  Serial.print("DISTANCE=" + String(distance1));
  Serial.println("Gunakan pompa=" + String(gunakan_pompa));
}

int bacaUltrasonik() {
  digitalWrite(initPin1, LOW);
  delayMicroseconds(5);
  digitalWrite(initPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(initPin1, LOW);

  long duration = pulseIn(echoPin1, HIGH);
  int distance = duration * 0.034 / 2;
  return distance;
}

float mapping(float x, float x1, float x2, float y1, float y2) {
  return (x - x1) * (y2 - y1) / (x2 - x1) + y1;
}

void pulseCounter()

```



```
{  
    pulseCount++;  
}  
double hitungTeg(int adc) {  
    double R1 = 47.0;  
    double R2 = 10.0;  
    double vout = (float)adc * 5.0 / 1023.0;  
    double vin = vout * (R1 + R2) / R2;  
    return vin;  
}
```