

LAMPIRAN

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);

// ==== PIN ====

#define PH_PIN 34

#define TURBIDITY_PIN 35

#define PH_DIGITAL 25

// ==== Variabel ====

float Voltage_pH = 0.0, ph_act = 0.0;

float turbidityVolt = 0.0, turbidityNTU = 0.0;

// ==== Helper ====

void clearLine(uint8_t row) {

    lcd.setCursor(0, row);

    for (int i = 0; i < 20; i++) lcd.print(' ');

    lcd.setCursor(0, row);

}

void setup() {
```

```

Serial.begin(115200);

analogSetAttenuation(ADC_11db);

pinMode(PH_DIGITAL, INPUT);

Wire.begin();

lcd.init();

lcd.backlight();

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Cek Kualitas Air");

delay(1200);

lcd.clear();
}

void loop() {
  // --- Baca pH (rata-rata 10x) ---
  long sumPh = 0;
  for (int i = 0; i < 10; i++) {
    sumPh += analogRead(PH_PIN);
    delay(8);
  }
  float adcPh = (float)sumPh / 10.0;
  Voltage_pH = adcPh * (3.3 / 4095.0);
}

```

```

// --- Kalibrasi pH ---
ph_act = (-5.8528 * Voltage_pH) + 23.3031;

// --- Baca Turbidity (rata-rata 10x) ---
long sumT = 0;
for (int i = 0; i < 10; i++) {
    sumT += analogRead(TURBIDITY_PIN);
    delay(5);
}
float adcT = (float)sumT / 10.0;
turbidityVolt = adcT * (3.3 / 4095.0);

// === ATURAN NTU BARU BERDASAR TEGANGAN ===
if (turbidityVolt > 2.0) {
    turbidityNTU = 0;    // Bersih
}
else if (turbidityVolt >= 1.0) {
    turbidityNTU = 2;    // NTU sedang (1-3)
}
else {
    turbidityNTU = 3;    // Keruh
}

// === KLASIFIKASI KUALITAS AIR ===
String kualitasAir;

```

```

if (turbidityNTU == 0) kualitasAir = "Jernih";
else if (turbidityNTU == 2) kualitasAir = "SedikitKeruh";
else kualitasAir = "Keruh";

// --- Logika kelayakan ---
bool layak = (kualitasAir == "Jernih" && (ph_act >= 6.5 && ph_act <= 8.5));

// === TAMPILAN LCD ===
// Baris 1: nilai pH & NTU
lcd.setCursor(0,0);
char buf1[21];
snprintf(buf1, sizeof(buf1), "pH:%.2f Turb:%.1fNTU", ph_act, turbidityNTU);
lcd.print(buf1);
for (int i = strlen(buf1); i < 20; ++i) lcd.print(' ');

// Baris 2: kualitas air
lcd.setCursor(0,1);
char buf2[21];
snprintf(buf2, sizeof(buf2), "Kualitas:%-12s", kualitasAir.c_str());
lcd.print(buf2);
for (int i = strlen(buf2); i < 20; ++i) lcd.print(' ');

// Baris 3: tegangan sensor
lcd.setCursor(0,2);
char buf3[21];

```

```

snprintf(buf3, sizeof(buf3), "VpH:%.2fV Vt:%.2fV", Voltage_pH, turbidityVolt);
lcd.print(buf3);
for (int i = strlen(buf3); i < 20; ++i) lcd.print(' ');

// Baris 4: status kelayakan
clearLine(3);
lcd.setCursor(0,3);
if (layak) lcd.print("Status=LayakKonsumsi");
else lcd.print("Status=TidakLayak");

// === SERIAL MONITOR ===
Serial.print("pH: "); Serial.print(ph_act,2);
Serial.print(" | VpH: "); Serial.print(Voltage_pH,3);
Serial.print(" | Vturb: "); Serial.print(turbidityVolt,3);
Serial.print(" | NTU: "); Serial.print(turbidityNTU,1);
Serial.print(" | Kualitas: "); Serial.print(kualitasAir);
Serial.print(" | Layak: "); Serial.println(layak ? "YA" : "TIDAK");

delay(1000);
}

```