

Lampiran 3 Source Code Arduino ke ESP8266

```
#include <SoftwareSerial.h>
#include <OneWire.h>
#include <DallasTemperature.h>

SoftwareSerial ss(2,3);
#define ONE_WIRE_BUS 4
#define relay 9

int LED1 = 5;
int LED2 = 6;
int buzzer_cont = 7;
int buzzer_bip = 8;
int Sensor_Value;
float Voltage;

int turbidity_pin = A0;
const int ph_pin = A1;

// Setup a oneWire instance to communicate with any OneWire
device
OneWire oneWire(ONE_WIRE_BUS);

// Pass oneWire reference to DallasTemperature library
DallasTemperature sensors(&oneWire);

// hasil kalibrasi Sensor pH
float PH4 = 4.68;
float PH7 = 3.70;

void setup() {
    // ===== Inisialisasi Input / Output =====
    Serial.begin (9600);
    ss.begin (115200);
```

```

pinMode(ph_pin, INPUT);
pinMode(relay, OUTPUT);
pinMode(buzzer_cont, OUTPUT);
pinMode(buzzer_bip, OUTPUT);
pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
}

void loop() {
    // ===== Setting Pembacaan Output Sensor =====

    // pH Sensor
    int analog_ph = analogRead(ph_pin);
    double tegangan_sensor = ((5 / 1023.0) * analog_ph);
    float ph_step = ((PH4 - PH7) / 3);
    float ph_output = (7.00 + ((PH7 - tegangan_sensor) /
    ph_step));

    // Turbidity Sensor
    int analog_turbidity = analogRead(turbidity_pin);
    int turbidity_output = map(analog_turbidity, 100,500,
    100, 0);

    // Temperatures Sensor (by library)
    sensors.requestTemperatures();

    // ===== Print Dalam Serial Monitor =====
    // Print Suhu air dalam satuan Celsius
    Serial.print("Temperature  : ");
    Serial.print(sensors.getTempCByIndex(0));
    Serial.print("^C"); Serial.print("\n");
}

```

```

// Print nilai pH air
Serial.print("pH           : ");
Serial.println(ph_output);
// Serial.print("Sensor Voltage: ");
// Serial.println(tegangan_sensor);

// Print kekeruhan air dalam satuan NTU
// Serial.print("sensor value : ");
// Serial.println(analog_turbidity);
Serial.print("NTU           : ");
Serial.println(turbidity_output);
Serial.println();

// ===== Perintah Alarm Suhu =====
if(sensors.getTempCByIndex(0) > 28.0 &&
sensors.getTempCByIndex(0) <= 32.0)
{
    digitalWrite(LED1, LOW);           // LED OFF
    digitalWrite(buzzer_cont, LOW);    // Buzzer tidak aktif
}
else
{
    digitalWrite(LED1, HIGH);          // LED ON
    digitalWrite(buzzer_cont, HIGH);   // Buzzer aktif
    sebagai alarm
}

// ===== Perintah Alarm pH =====
if(ph_output > 6.20 && ph_output < 7.0)
{
    digitalWrite(LED2, LOW);           // LED OFF
    digitalWrite(buzzer_bip, LOW);     // Buzzer tidak aktif
}

```

```

else
{
    digitalWrite(LED2, HIGH);          // LED ON
    digitalWrite(buzzer_bip, HIGH);    // Buzzer aktif
    sebagai alarm
}

// ===== Perintah Buka dan Tutup Katup Filter Air =====
if(turbidity_output > 25) // Air terbaca keruh, Katup
terbuka
{
    digitalWrite(relay, LOW);
}
}
else // Air terbaca jernih, Katup tertutup
{
    digitalWrite(relay, HIGH);
}

// ===== Kirim Hasil Pembacaan Sensor ke ESP8266 =====
String sendToESP = "";
sendToESP += sensors.getTempCByIndex(0);
sendToESP += ";";
sendToESP += ph_output;
sendToESP += ";";
sendToESP += turbidity_output;
ss.println(sendToESP);

delay(2000);
}

```

Lampiran 4 Source Code ESP8266 ke Database

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

float ph = 0, temp = 0;
int turbidity = 0;

// Konfigurasi WiFi
const char *ssid = "Cinnamon";
const char *password = "12345678";

// URL server
const char *serverName = "http://discusplus.tif-
skripsi.my.id/kirimdata.php";

void setup()
{
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }

  // Jika koneksi berhasil, maka akan muncul address di
  serial monitor
  Serial.println("");
  Serial.print("Connected to ");
```

```

    Serial.print("Connected to ");
    Serial.println(ssid);
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
}
String splitString(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = {0, -1};
    int maxIndex = data.length() - 1;

    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i + 1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0],
strIndex[1]) : "";
}

void loop()
{
    if (Serial.available()) {
        String msg = "";
        while (Serial.available()) {
            msg += char(Serial.read());
            delay(50);
        }
        temp = splitString(msg, ';', 0).toFloat();
        ph = splitString(msg, ';', 1).toFloat();
        turbidity = splitString(msg, ';', 2).toInt();
    }
}

```

```
Serial.print(msg);

    Serial.print("temp: ");
    Serial.print(temp);
    Serial.print(", pH: ");
    Serial.print(ph);
    Serial.print(", turbidity: ");
    Serial.println(turbidity);
    Serial.print("\n");
}

if (WiFi.status() == WL_CONNECTED) {
    WiFiClient client;
    HTTPClient http;

    // Construct the URL
    String url = String(serverName) + "?suhu=" + temp +
"&ph=" + ph + "&kekeruhan=" + turbidity;

    http.begin(client, url); // Specify the URL and
WiFiClient

    int httpCode = http.GET(); // Make the request

    // Check the returning code
    if (httpCode > 0) {
        // Get the request response payload
        String payload = http.getString();

        // Serial.println(httpCode); // Comment this line if
you don't want to print the HTTP code

        Serial.println(payload);
    } else {
        Serial.println("Error on HTTP request");
    }
}
```

```

    http.end();
} else {
    Serial.println("WiFi Disconnected");
}

delay(5000); // Delay for 5 seconds before next loop
}

```

Lampiran 5 Source Code Konek PHP dengan Database

```

<?php
// Database connection
$konek = new mysqli("localhost", "root", "",
"grafiksensor");

// Check connection
if ($konek->connect_error) {
    die("Connection failed: " . $konek->connect_error);
}

// Get the input values
$suhu = $_GET['suhu'];
$ph = $_GET['ph'];
$kekeruhan = $_GET['kekeruhan'];

// Validate inputs
if (is_numeric($suhu) && is_numeric($ph) &&
is_numeric($kekeruhan)) {
    // Reset auto increment (optional, but make sure it's
    necessary)

    $konek->query("ALTER TABLE tb_sensor
    AUTO_INCREMENT=1");
}

```

```
// Prepare and bind
$stmt = $koneksi->prepare("INSERT INTO tb_sensor (suhu,
ph, kekeruhan) VALUES (?, ?, ?)");

$stmt->bind_param("ddd", $suhu, $ph, $kekeruhan); //
"ddd" means three double parameters

// Execute the statement
if ($stmt->execute()) {
    echo "Berhasil disimpan";
} else {
    echo "Gagal tersimpan: " . $stmt->error;
}

// Close the statement
$stmt->close();
} else {
    echo "Invalid input";
}

// Close the connection
$koneksi->close();
?>
```