Lampiran 3 Source Code:

```
#include <SPI.h>
#include <Ethernet.h>
#include <DHT.h>
#include <Servo.h>

// Sensor and actuator pins
#define DHTPIN 2               // Pin untuk sensor DHT11
#define DHTTYPE DHT11          // Tipe sensor DHT11
#define SOIL_MOISTURE_PIN A0   // Pin analog untuk sensor kelembaban
tanah
#define LDR_PIN A1             // Pin untuk sensor LDR
#define RELAY_PIN 3            // Pin untuk relay lampu LED
#define WATERPUMP_PIN 4        // Pin untuk relay pompa air

DHT dht(DHTPIN, DHTTYPE);
Servo servo;

// Ethernet setup
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 177);
IPAddress server_ip(192, 168, 1, 100); // Ubah sesuai dengan IP
server Anda
EthernetServer server(80);
EthernetClient client;

// Servo control variables
int servoPos = 0;        // Current position of the servo
int servoDir = 1;        // Direction of servo movement (1 for
right, -1 for left)
unsigned long lastMoveTime = 0; // Timestamp of the last servo
movement
const int moveInterval = 1000; // Interval between servo movements
in milliseconds (1 second)

// Status variables
bool lampuStatus = LOW;  // Status lampu LED
bool pompaStatus = LOW;  // Status pompa air
bool manualControlLampu = false; // Status pengendalian manual untuk
lampu
bool manualControlPompa = false; // Status pengendalian manual untuk
pompa
```

```
void setup() {
  Serial.begin(9600);
  dht.begin();
  servo.attach(5); // Menghubungkan servo ke pin digital 5

  pinMode(SOIL_MOISTURE_PIN, INPUT);
  pinMode(LDR_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);
  pinMode(WATERPUMP_PIN, OUTPUT);

  digitalWrite(RELAY_PIN, lampuStatus);
  digitalWrite(WATERPUMP_PIN, pompaStatus);
  servo.write(0); // Pastikan servo berada di posisi awal

  // Initialize Ethernet
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("Server is at ");
  Serial.println(Ethernet.localIP());
}

void loop() {
  // Membaca data dari sensor
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  int soilMoistureAnalog = analogRead(SOIL_MOISTURE_PIN); //
Membaca nilai analog dari sensor kelembaban tanah
  int lightIntensity = analogRead(LDR_PIN);

  // Mengkonversi nilai analog kelembaban tanah menjadi persentase
  float soilMoisturePercent = map(soilMoistureAnalog, 1023, 0, 0,
100);

  // Debugging: Print data sensor ke Serial Monitor
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print(" C, Humidity: ");
  Serial.print(humidity);
  Serial.print(" %, Soil Moisture: ");
  Serial.print(soilMoisturePercent);
  Serial.print(" %, Light Intensity: ");
  Serial.println(lightIntensity);
```

```
if (!manualControlLampu) {
    if (lightIntensity < 100) {
      digitalWrite(RELAY_PIN, HIGH); // Menyalakan lampu jika cahaya
kurang
      digitalWrite(WATERPUMP_PIN, HIGH);
      lampuStatus = HIGH;
    } else {
      digitalWrite(RELAY_PIN, LOW); // Mematikan lampu jika cahaya
cukup
      lampuStatus = LOW;
    }
  } else {
    digitalWrite(RELAY_PIN, lampuStatus);
  }

  // Mengontrol pompa air dan servo berdasarkan sensor kelembaban
tanah atau suhu ekstrim
  if (!manualControlPompa) {
    if (soilMoisturePercent < 1.00) { // Menyiram jika kelembaban
tanah kurang dari 30%
      digitalWrite(WATERPUMP_PIN, HIGH); // Menyalakan pompa
      pompaStatus = HIGH;
      moveServo(); // Menggerakkan servo
    } else {
      digitalWrite(WATERPUMP_PIN, LOW); // Mematikan pompa jika
kelembaban tanah lebih dari atau sama dengan 30%
      pompaStatus = LOW;
      servo.write(0); // Pastikan servo berada di posisi awal
    }
  } else {
    digitalWrite(WATERPUMP_PIN, pompaStatus);
  }

  // Mengontrol pompa air berdasarkan suhu ekstrim
  if (temperature < 1.10 ) { // Menyiram jika suhu di bawah 1.10
derajat Celsius (dikonfigurasi berdasarkan kondisi ekstrim yang Anda
tentukan)
    digitalWrite(WATERPUMP_PIN, HIGH); // Menyalakan pompa
    //moveServo(); // Menggerakkan servo
  } else {
    digitalWrite(WATERPUMP_PIN, LOW); // Mematikan pompa jika suhu
tidak terlalu ekstrim
    //servo.write(0); // Pastikan servo berada di posisi awal
  }
```

```
 // Mengirim data sensor ke server
  sendSensorData(temperature, humidity, soilMoisturePercent,
lightIntensity);

  // Terima dan proses permintaan HTTP
  EthernetClient client = server.available();
  if (client) {
    String request = client.readStringUntil('\r');
    Serial.println(request);
    client.flush();

    // Kontrol relay berdasarkan permintaan HTTP
    if (request.indexOf("/lampu_on") != -1) {
      lampuStatus = HIGH;
      manualControlLampu = true;  // Set manual control to true
      digitalWrite(RELAY_PIN, HIGH);
    }
    if (request.indexOf("/lampu_off") != -1) {
      lampuStatus = LOW;
      manualControlLampu = true;  // Set manual control to true
      digitalWrite(RELAY_PIN, LOW);
    }
    if (request.indexOf("/pompa_on") != -1) {
      pompaStatus = HIGH;
      manualControlPompa = true;  // Set manual control to true
      digitalWrite(WATERPUMP_PIN, HIGH);
    }
    if (request.indexOf("/pompa_off") != -1) {
      pompaStatus = LOW;
      manualControlPompa = true;  // Set manual control to true
      digitalWrite(WATERPUMP_PIN, LOW);
    }
    if (request.indexOf("/otomatis") != -1) {
      manualControlLampu = false;  // Set manual control to false
      manualControlPompa = false;  // Set manual control to false
    }

    // Kirim respons HTTP
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("Connection: close");
    client.println();
    client.stop();
  }
```

```
 delay(10000); // Interval antara siklus pembacaan sensor (5 detik)
}

void moveServo() {
  unsigned long currentTime = millis();
  if (currentTime - lastMoveTime >= moveInterval) {
    // Update the servo position
    servoPos += servoDir * 40; // Menggerakkan servo sebesar 40
derajat
    // Change direction if the servo hits the end
    if (servoPos >= 180 || servoPos <= 0) {
      servoDir = -servoDir;
    }
    // Move the servo to the new position
    servo.write(servoPos);
    // Update the last move time
    lastMoveTime = currentTime;
  }
}
void sendSensorData(float temperature, float humidity, float
soilMoisturePercent, int lightIntensity) {
  if (client.connect(server_ip, 80)) {
    String data = "temperature=" + String(temperature) +
"&humidity=" + String(humidity) + "&soilMoisture=" +
String(soilMoisturePercent) + "&lightIntensity=" +
String(lightIntensity);

    // Debugging: Print data yang akan dikirim
    Serial.println("Sending data: " + data);
    client.println("POST /save_data.php HTTP/1.1");
    client.print("Host: ");
    client.println(server_ip);
    client.println("Content-Type: application/x-www-form-
urlencoded");
    client.println("Connection: close");
    client.print("Content-Length: ");
    client.println(data.length());
    client.println();
    client.println(data);
    client.stop();
  } else {
    Serial.println("Connection failed");
  }
}
```