

LAPORAN PRAKTIKUM 4 IOT

Transfer Data Terformat



Disusun Oleh

Nama : Rangga Prabaswara

NIM : 3.34.21.0.21

Kelas : IK-2A

POLITEKNIK NEGERI SEMARANG

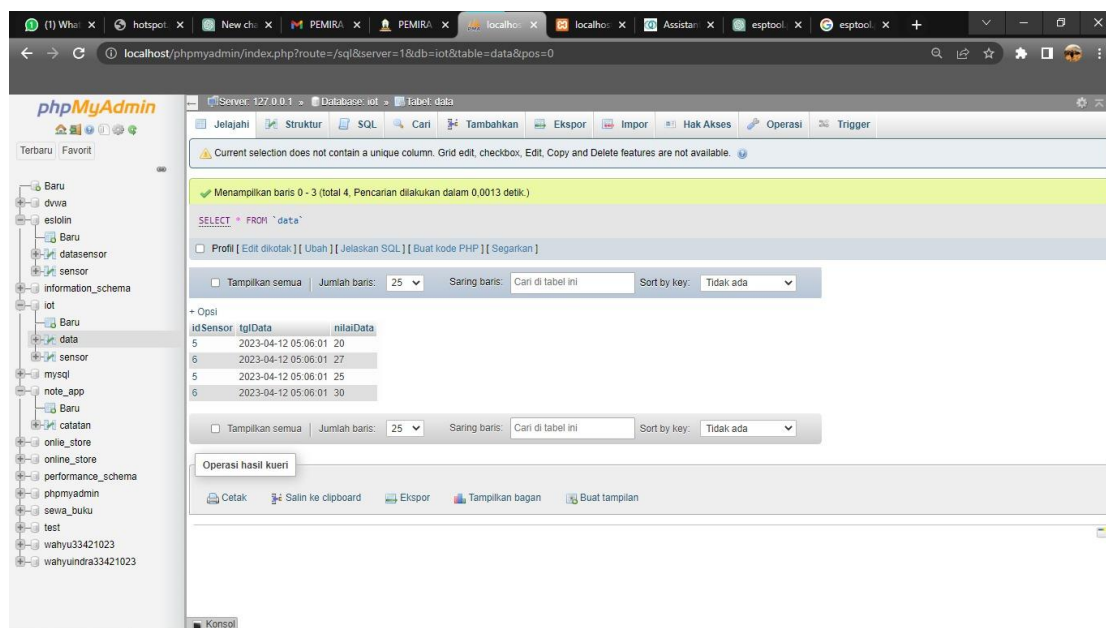
Transfer Data Terformat

Tujuan

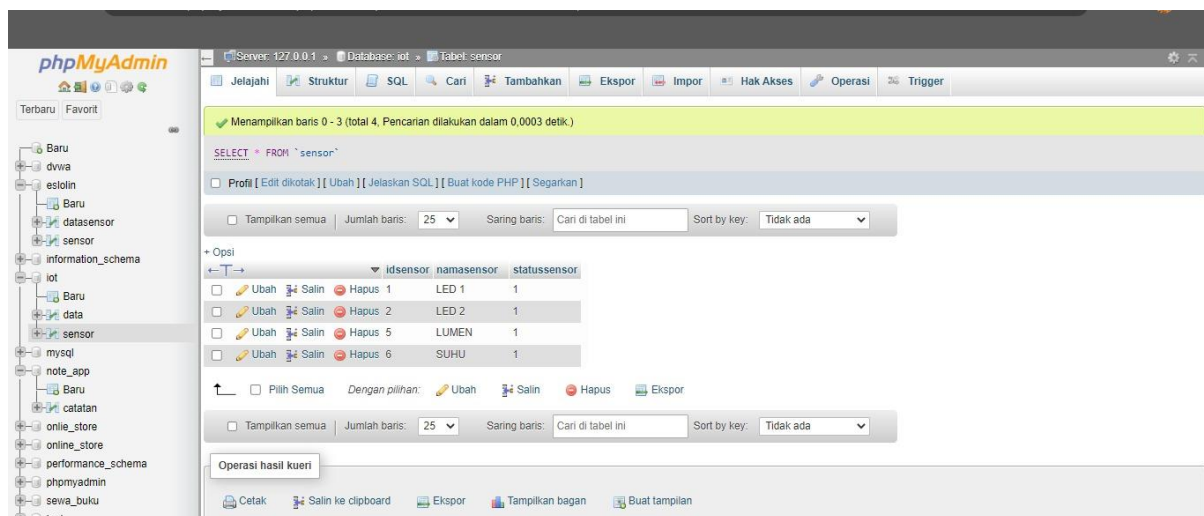
1. Mahasiswa mampu mentransferkan data dengan format yang bisa dipahami dan diterapkan diberbagai bahasa pemrograman.
2. Mahasiswa mampu memprogram untuk read dan write data sesuai format dari dan ke mikrokontroller.
3. Mahasiswa mampu membuat aplikasi dengan mikrokontroler dengan format data dan mengimplementasikan pada

Menyiapkan database dan tabel

Tabel Data



Tabel Sensor



Source code sensormulti-json.php

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "eslolin";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$conn) {
    die("Koneksi gagal: " . mysqli_connect_error());
}

$sql = "SELECT idSensor, namaSensor, statusSensor FROM sensor where idSensor=1";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {

    $objData = new stdClass;

    while($row = mysqli_fetch_assoc($result)) {

        $objData->idSensor = $row["idSensor"];
        $objData->namaSensor = $row["namaSensor"];
        $objData->statusSensor = $row["statusSensor"];

    }
} else {
    echo "0 results";
}

```

```

}
$objData->add = [36,'40']; // [suhu, lumen]
mysqli_close($conn);

//konversi data menjadi json
$dataJSON = json_encode($objData,JSON_FORCE_OBJECT);
echo $dataJSON; //menampilkan data sebagai json

?>

```

Source Code program arduino yang sudah diberikan kondisi dimana if led 1 on dan 0 mati.

```

#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>

WiFiClient wifiClient;

// Variabel global
const char* ssid = "MITTO";
const char* password = "123456789";
int pinLED = 2;
const int ON = HIGH;
const int OFF = LOW;

//tambahan
int firstVal, secondVal;

void setup() {
    // put your setup code here, to run once:

```

```

// Konfigurasi Modul

Serial.begin(115200);

// Connect to WiFi network

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);


while (WiFi.status() != WL_CONNECTED) {

    //Wait till connects

    //Wait till connects

    delay(500);

    Serial.print(".");

}


Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

// --- End of Connect wifi


// Deklarasi PIN GPIO 1

pinMode(pinLED, OUTPUT);

}


void loop() {

    delay(5000);


    Serial.print("connecting to ");


    while (WiFi.status() != WL_CONNECTED) {

```

```

delay(500);

Serial.print(".");

}

Serial.println(WiFi.localIP());


// put your main code here, to run repeatedly:

if(WiFi.status() == WL_CONNECTED){

  HTTPClient http;

  Serial.print("[HTTP] begin...\n");

  //IP menuju ke server web
  http.begin(wifiClient,"http://192.168.136.248/eslolin/sensormulti-json.php");


  Serial.print("[HTTP] GET...\n");
  int httpCode = http.GET();
  Serial.print(httpCode);
  if(httpCode > 0){
    // HTTP header has been send and Server response header has been handled
    Serial.printf("[HTTP] GET... code: %d\n", httpCode);

    // file found at server
    //If(200 == 200)
    if(httpCode == HTTP_CODE_OK) {
      String answer = http.getString();

      //awal

      // Convert to JSON
      String jsonAnswer;
      int jsonIndex;

      for (int i = 0; i < answer.length(); i++) {
        if (answer[i] == '{') {
          jsonIndex = i;

```

```

        break;
    }
}

// Get JSON data
jsonAnswer = answer.substring(jsonIndex);
Serial.println();
Serial.println("JSON answer: ");
Serial.println(jsonAnswer);
jsonAnswer.trim();

// Allocate the JSON document
// Use https://arduinojson.org/v6/assistant to compute the capacity.
// const size_t capacity = JSON_OBJECT_SIZE(3) + JSON_ARRAY_SIZE(2) + 60;
// DynamicJsonDocument doc(capacity);

// Stream& input;
StaticJsonDocument<192> doc;
DeserializationError error = deserializeJson(doc, jsonAnswer);

if (error) {
    Serial.print(F("deserializeJson() failed: "));
    Serial.println(error.f_str());
    return;
}

const char* idSensor = doc["idSensor"]; // "1"
const char* namaSensor = doc["namaSensor"]; // "LED 1"
const char* statusSensor = doc["statusSensor"]; // "1"

```

```

int add_0 = doc["add"]["0"]; // 36
const char* add_1 = doc["add"]["1"]; // "40

// Extract values
Serial.println(F("Response:"));
    Serial.println(namaSensor);
    Serial.println(idSensor);
    Serial.println(statusSensor);
    Serial.println(add_0);
    Serial.println(add_1);
    if(strcmp(statusSensor, "1") == 0){ // check if statusSensor is "1"
        digitalWrite(pinLED, ON); // turn on the LED
    } else {
        digitalWrite(pinLED, OFF);
    }
} else {
    Serial.printf("[HTTP] GET... failed, error: %s\n",
        http.errorToString(httpCode).c_str());
}
delay(50);
}

http.end();

} else {
    Serial.println("Delay...");
}
delay(1000);
}

```

Penjelasan menambahkan code untuk menyalakan lampu


```
if(strcmp(statusSensor, "1") == 0){ // check if statusSensor is "1"
    digitalWrite(pinLED, ON); // turn on the LED
} else {
    digitalWrite(pinLED, OFF);
}
```

Jika diberikan nilai 1 lampu akan menyala dan diberikan nilai 0 lampu akan mati.

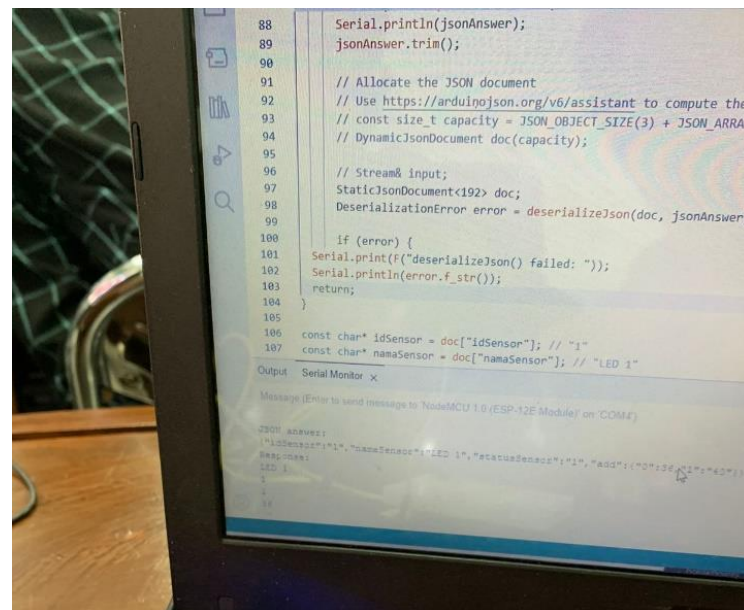
Kode tersebut adalah contoh program untuk Arduino yang menggunakan fungsi `strcmp` untuk membandingkan nilai variabel `statusSensor` dengan string "1". Jika nilai `statusSensor` sama dengan "1", maka akan mengaktifkan LED dengan menggunakan fungsi `digitalWrite` dan mengatur nilai pin LED menjadi HIGH (ON). Jika nilai `statusSensor` tidak sama dengan "1", maka akan mematikan LED dengan mengatur nilai pin LED menjadi LOW (OFF) menggunakan fungsi `digitalWrite`.

Dalam konteks Arduino, fungsi `digitalWrite` digunakan untuk mengendalikan status pin digital, yang dapat berupa pin digital yang terhubung ke LED atau komponen elektronik lainnya. Nilai HIGH (ON) akan mengaktifkan pin digital dan nilai LOW (OFF) akan mematikan pin digital.

`strcmp` merupakan fungsi standar pada bahasa C/C++ yang digunakan untuk membandingkan dua string. Dalam kode tersebut, `strcmp(statusSensor, "1")` akan membandingkan string yang disimpan dalam variabel `statusSensor` dengan string "1". Jika kedua string tersebut sama, maka `strcmp` akan mengembalikan nilai 0, dan pernyataan `strcmp(statusSensor, "1") == 0` akan bernilai benar, sehingga LED akan dinyalakan. Jika kedua string tersebut tidak sama, maka pernyataan `strcmp(statusSensor, "1") == 0` akan bernilai salah, dan LED akan dimatikan.

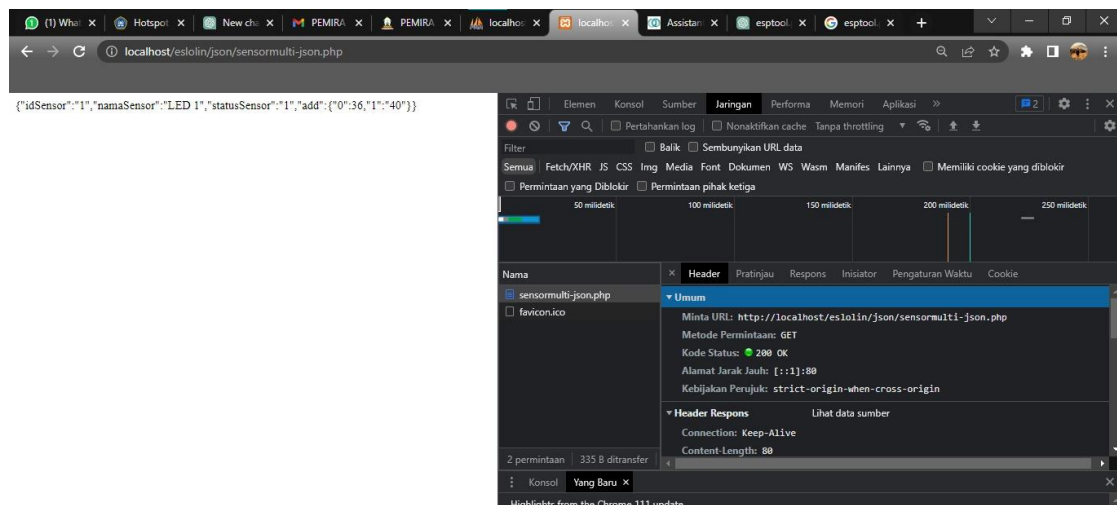
FOTO PRAKTIKUM

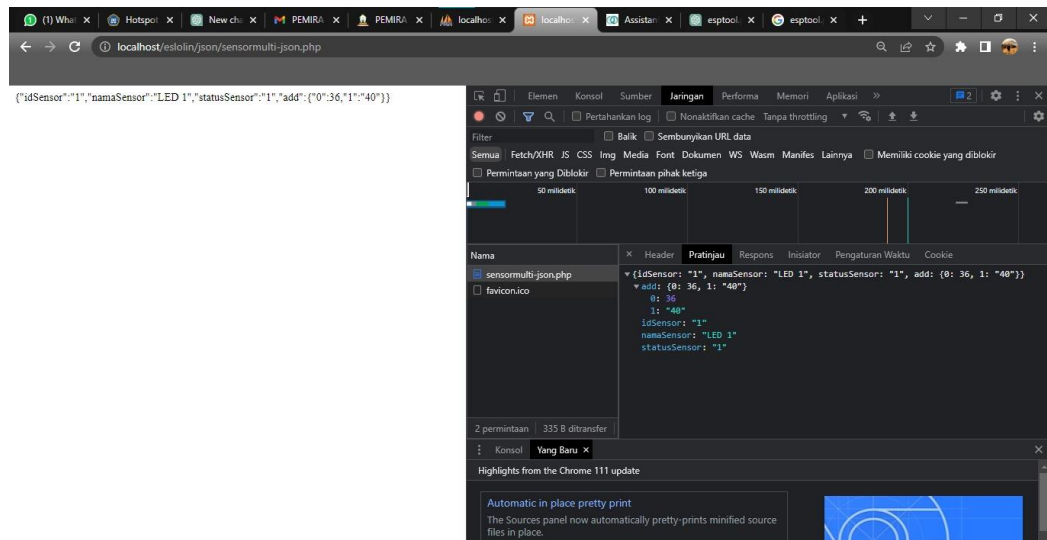
Hasil command line di arduino file



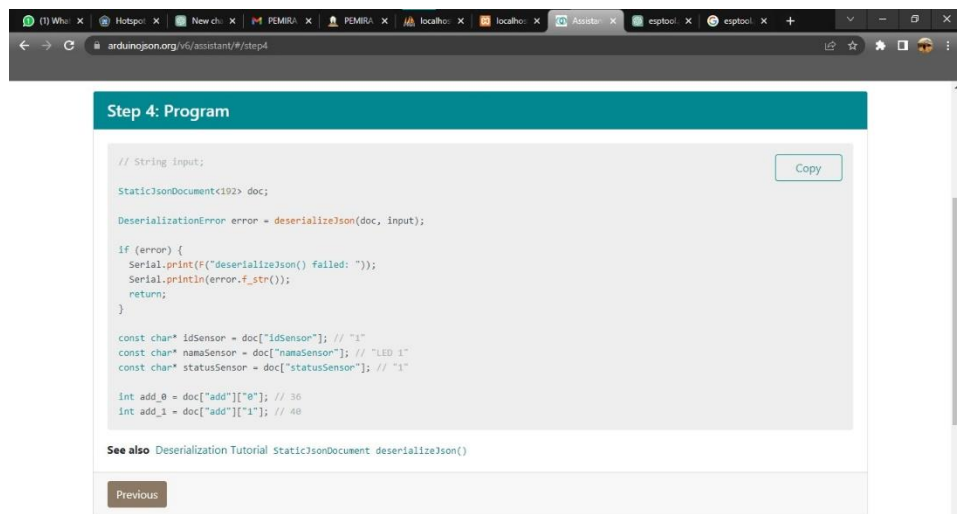
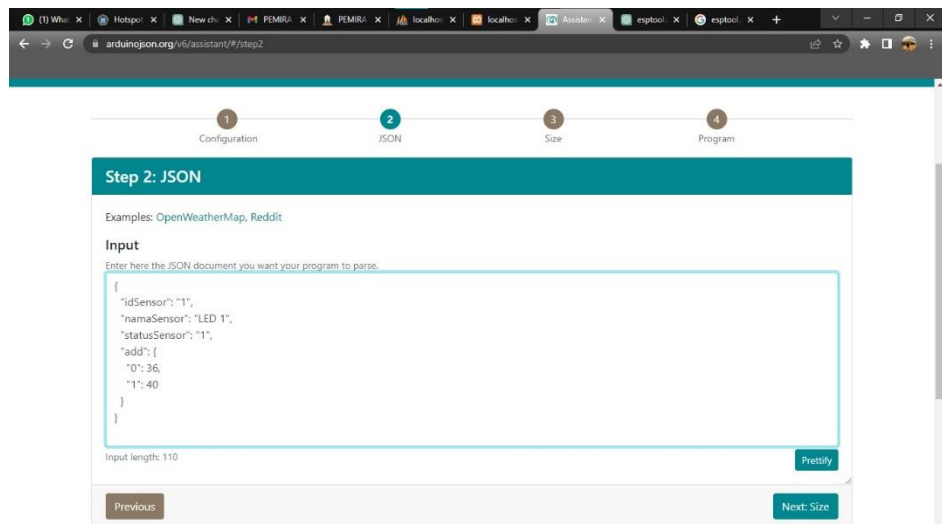
Hasil tampilan website dari hasil localhost httdocs file sensormulti-json.php

status get terbaca dan OK





Tampilan saat diwebsite arduinojson.org



Tampilan Lampu menyala



Analisis sederhana dan kesimpulan mempelajari praktikum datasensor

Praktikum Transfer Data Terformat dengan LED menggunakan format JSON dan database PHP merupakan sebuah eksperimen atau latihan yang melibatkan pengiriman data dari suatu perangkat mikrokontroler atau Arduino ke server berbasis PHP melalui protokol komunikasi tertentu, dan data tersebut diorganisir dalam format JSON. Selanjutnya, data yang diterima oleh server akan diproses dan disimpan dalam database menggunakan bahasa pemrograman PHP.

Analisis Sederhana:

1. Format Data JSON: Data dikirimkan dalam format JSON, yang merupakan format data yang ringan, mudah dibaca, dan dapat diinterpretasikan oleh banyak bahasa pemrograman.
2. Pengiriman Data: Data JSON dikirimkan dari perangkat mikrokontroler ke server menggunakan protokol komunikasi seperti HTTP atau MQTT.
3. Pengolahan Data di Server: Data JSON di-parse di server PHP untuk mengambil nilai-nilai dari pasangan kunci-nilai dan dapat digunakan untuk berbagai operasi, seperti penyimpanan data ke dalam database atau pengiriman notifikasi.
4. Penyimpanan Data ke Database: Data JSON yang sudah di-parse akan disimpan ke dalam database menggunakan bahasa pemrograman PHP yang menyediakan fungsi-fungsi untuk berinteraksi dengan database.

Kesimpulan:

Praktikum Transfer Data Terformat dengan LED menggunakan format JSON dan database PHP merupakan pengenalan tentang bagaimana data dapat dikirim dari perangkat mikrokontroler ke server menggunakan format JSON, dan kemudian diproses dan disimpan ke dalam database menggunakan bahasa pemrograman PHP. Dalam praktikum ini, Anda akan mempelajari bagaimana mengirim dan menerima data JSON, mengelola data JSON di server, dan melakukan penyimpanan data ke dalam database. Praktikum ini memberikan pemahaman

dasar tentang bagaimana pengiriman data terformat dapat dilakukan dalam proyek berbasis mikrokontroler, serta pengelolaan data menggunakan format JSON dan database PHP.