

REPUBLIC INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202384256, 22 September 2023

Pencipta

Nama : **Muhammad Fatih Qodri, S.T., M.Eng., Bagus Gilang Pratama, S.T., M.Eng. dkk**

Alamat : Jalan Piyungan Prambanan KM 01 Piyungan RT10,
Piyungan, Bantul, DI Yogyakarta, 55792

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **LPPMI INSTITUT TEKNOLOGI NASIONAL YOGYAKARTA**

Alamat : Jl Babarsari Tambak Bayan,
Depok, Sleman, Di Yogyakarta 55281

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **“WAMOT” Sistem Pemantau Limbah Cair Terintegrasi (Dissolved Oxygen, Turbidity, PH, Suhu, Dan Total Dissolved Solids) Berbasis Internet Of Things Terproteksi Wireguard**

Tanggal dan tempat diumumkan untuk pertama kali : 12 September 2023, di Sleman
di wilayah Indonesia atau di luar wilayah Indonesia

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000517209

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
Direktur Hak Cipta dan Desain Industri



Anggoro Dasananto
NIP. 196412081991031002

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Muhammad Fatih Qodri, S.T., M.Eng.	Jalan Piyungan Prambanan KM 01 Piyungan RT10
2	Bagus Gilang Pratama, S.T., M.Eng.	Jalan Pahlawan No. 107, LK. II, RT 15, Surabaya
3	Ir. Oni Yuliani, ST., M.Kom	Prayan Kulon No.96-A RT 005/RW 037 Suropadan Condong Catur
4	Irvan Ardhi Permana	Villa Intan 2 Blok L.5 No.8 017/RW 006 Klayan

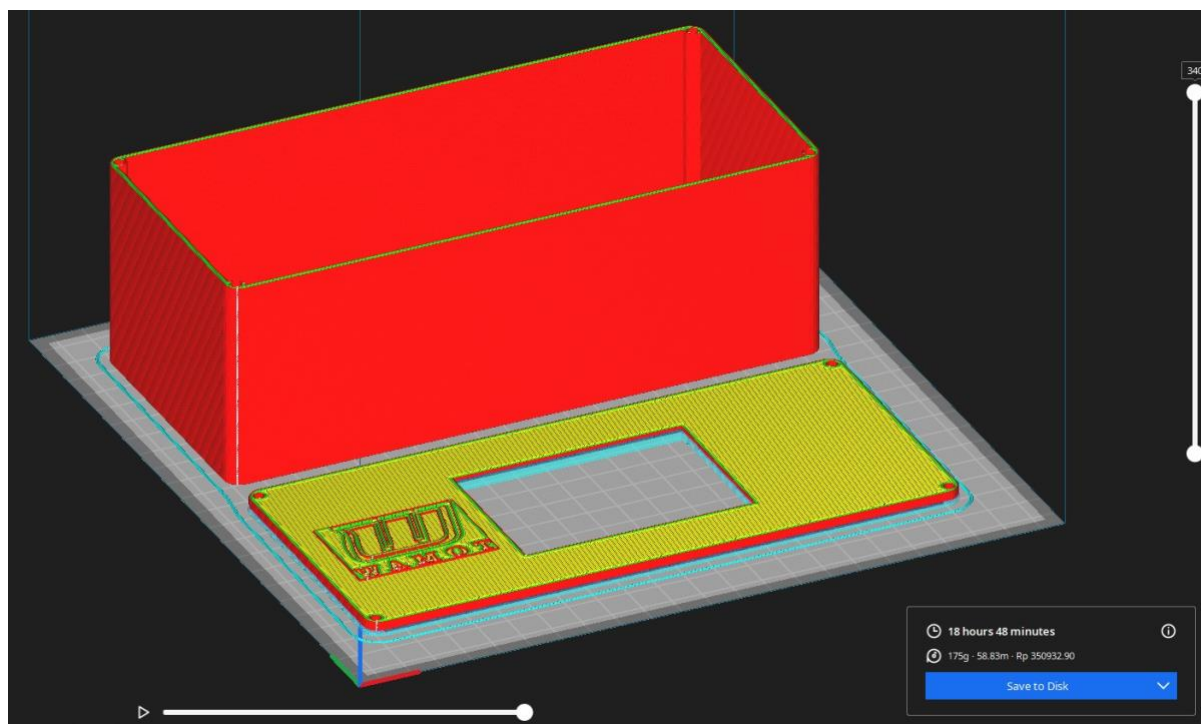


DOKUMEN SPESIFIKASI WAMOT

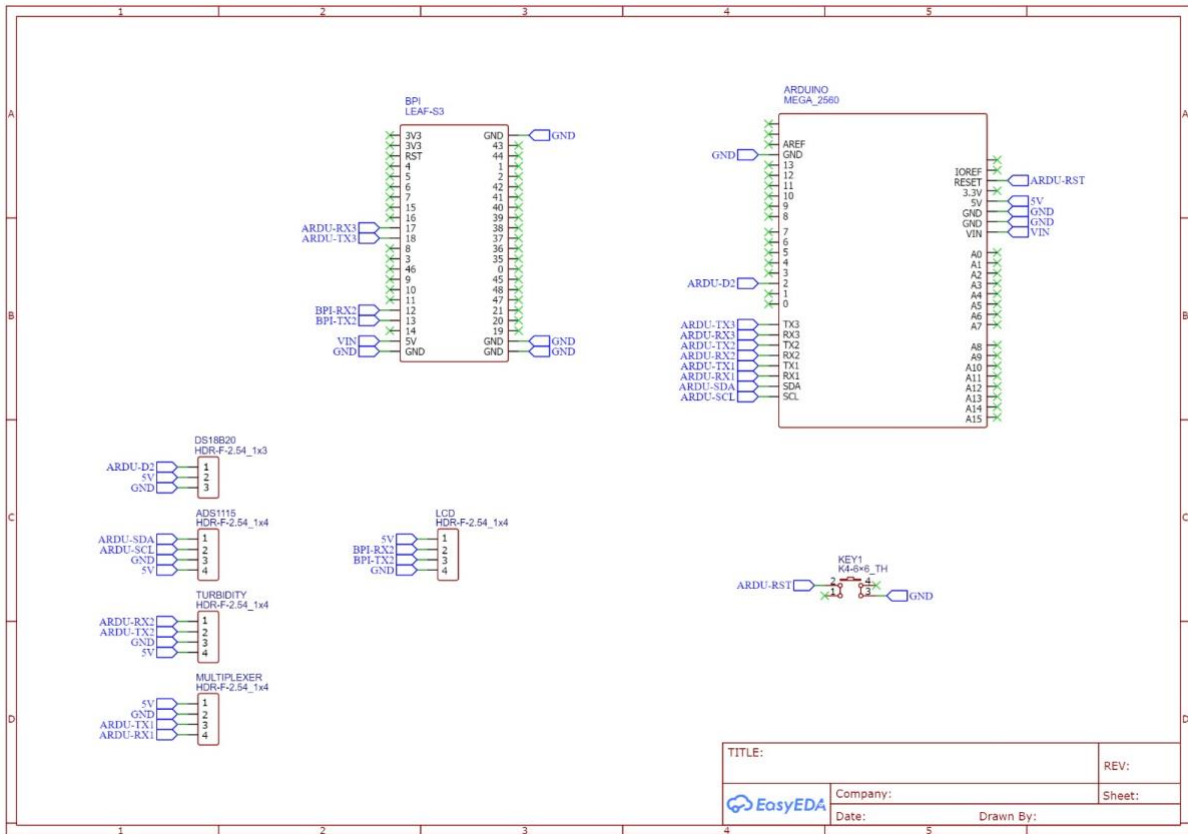
Sistem Pemantauan Otomatis terintegrasi pada Limbah Cair Berbasis Internet of Things (IoT) dengan terproteksi *Wireguard* merupakan sebuah sistem teknologi masa kini yang memungkinkan dapat menjadikan solusi praktis untuk melakukan monitoring terhadap sistem limbah. Sistem ini diberi nama WAMOT dengan kepanjangan Water Monitoring. Sensor-sensor yang terintegrasi akan mengirimkan data untuk melakukan monitoring melalui jaringan internet pada industri. Sensor-sensor yang digunakan dalam perangkat ini ditunjukkan pada Tabel 1.

Tabel 1. Sensor yang digunakan

No	Sensor	Keterangan	Fungsi
1	pH Sensor	Seeed Studio Industrial pH Meter	Membaca derajat pH pada cairan
2	eC Sensor	Gravity Analog Electrical Conductivity Sensor	Membaca electrical conductivity pada cairan yang berkorelasi dengan TDS (Total Dissolved Solid)
3	Dissolved Oxygen Sensor	Gravity Analog Dissolved Oxygen Sensor	Membaca oksigen terlarut pada cairan
4	Temperature Sensor	Waterproof DS18B20 Digital Temperature Sensor	Mengukur suhu yang berguna untuk referensi eC sensor dan pH sensor
5	Turbidity Sensor	Non contact Optical Turbidity Sensor	Membaca kekeruhan pada cairan

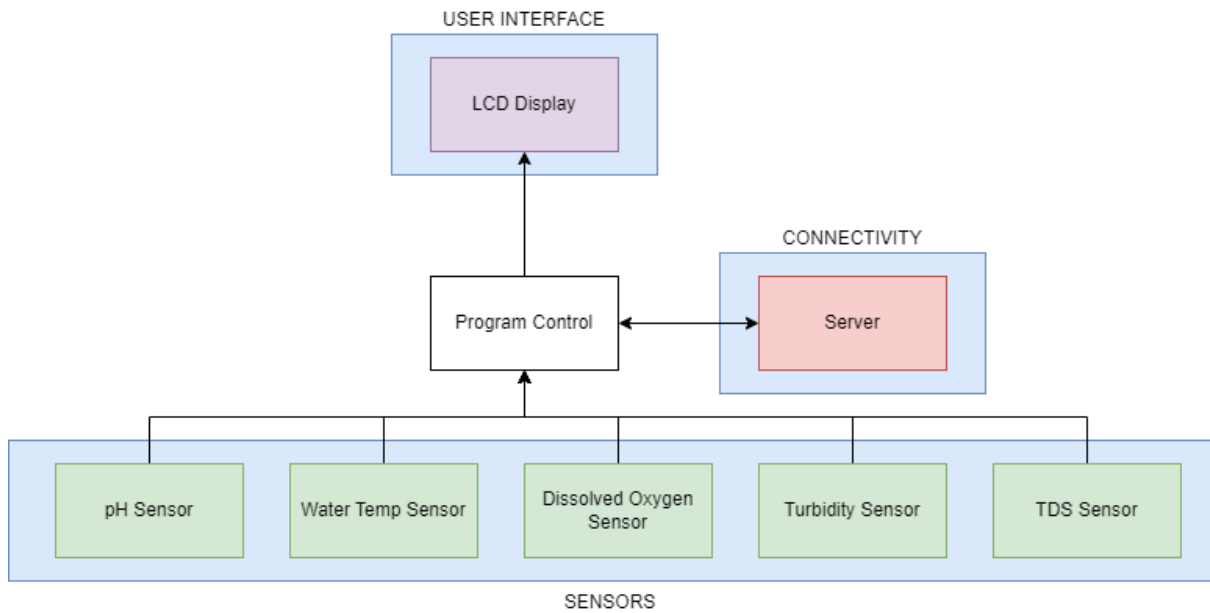


Gambar 1. Tampilan 3D Case Wamot



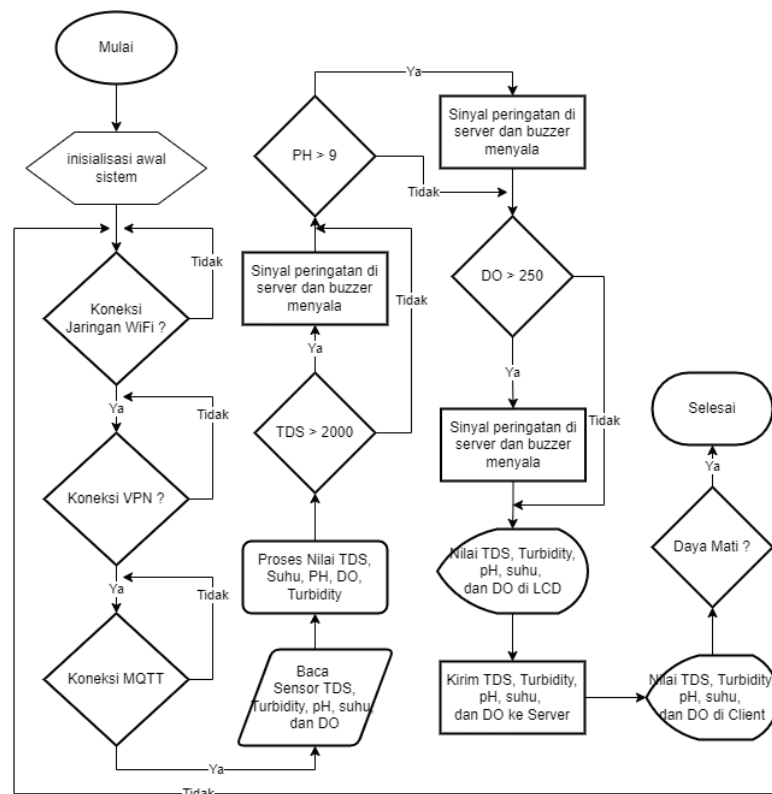
Gambar 2. Skematik Alat Wamot

Sistem ini menggunakan ESP32-S3 sebagai mikrokontroler utama dalam melakukan komunikasi dengan server dan mikrokontroler Arduino Mega untuk melakukan pembacaan terhadap kualitas TDS, suhu, Dissolved Oxygen (DO), turbidity, dan pH limbah cair yang ada di industri. Hasil pembacaan kemudian akan diproses di dalam ESP32-S3 dan akan dilakukan perbandingan apakah melebihi ambang batas yang telah ditentukan oleh pengguna. Jika hasil pembacaan TDS, DO, turbidity, dan/atau pH melebihi ambang batas yang telah ditetapkan, maka sistem ESP32-S3 akan memberikan peringatan ke server. Lebih lanjut, seluruh informasi mengenai hasil pembacaan TDS, suhu, Dissolved Oxygen (DO), turbidity, dan pH akan dikirimkan kepada smartphone pengguna, sehingga pengguna dapat memantau kualitas limbah cair secara real time. Agar sistem ESP32 dapat mengirimkan data pembacaan TDS dan pH ke smartphone pengguna, kami akan menggunakan server MQTT sebagai terminal data dan penghubung antara sistem ESP32 dan smartphone pengguna.

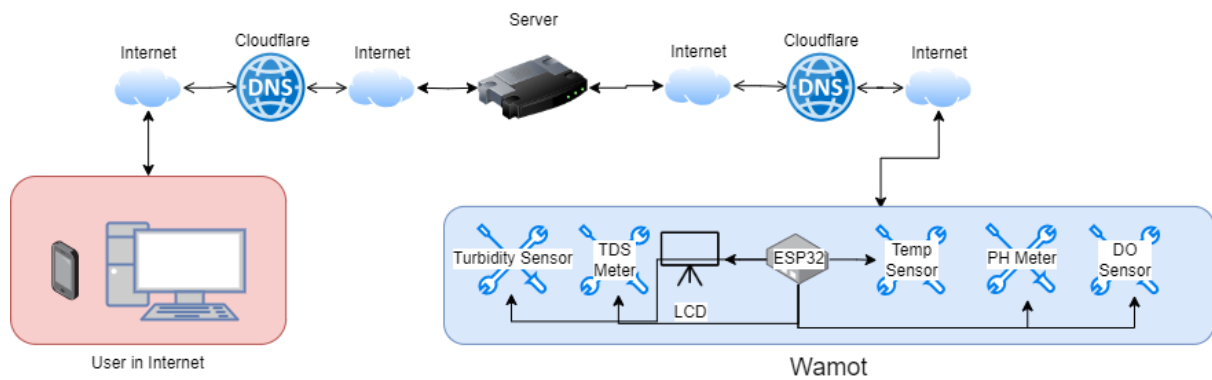


Gambar 3. Blok Diagram WAMOT

Alur pemrograman dirancang untuk melakukan pemantauan parameter fisika limbah cair di pabrik Gambar 3 & 4. Sistem dirancang akan melakukan pembacaan terhadap sensor yang dipasang di limbah cair. Ketika nilai TDS pada limbah cair tersebut melebihi 2000 dan/atau nilai pH melebihi 9, maka sistem akan membunyikan buzzer peringatan dan mengirimkan peringatan kepada operator melalui *smartphone*. Selanjutnya akan dikembangkan parameter-parameter lain seperti DO dll. Sebelumnya pengembangan skala kecil sudah dibuat yang hanya berdasarkan pada parameter suhu dan TDS dan didaftarkan HKI.



Gambar 4. Diagram Alir WAMOT



Gambar 5. Aliran Data WAMOT

Pengiriman peringatan ke *smartphone* operator harus melalui MQTT *server* yang bertindak sebagai terminal dan penghubung data antara sistem pemantau limbah cair dan *smartphone* operator. Tampilan *smartphone* dapat ditunjukkan pada Gambar 3. Sistem yang dibangun karena berbasis IoT memungkinkan operator untuk memantau kualitas dari TDS dan pH limbah cair dimanapun dan kapanpun. Operator akan menerima info secara *real time* selama terhubung dengan jaringan internet (Gambar 5).

Listing Program

```
#include <WiFi.h>
#include <ESP32Ping.h>

#include <SoftwareSerial.h>

#include <PubSubClient.h>
#include <WireGuard-ESP32.h>
#include <WiFiClient.h>
#include <ESPmDNS.h>
#include <Update.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <AsyncElegantOTA.h>
#include <EEPROM.h>
#include <HTTPClient.h>

const byte rxPin = 12;
const byte txPin = 13;

// Set up a new SoftwareSerial object
SoftwareSerial mySerial (rxPin, txPin);

// const char* ssid      = "HDIRFAN_ext";
// const char* password = "Hasa246810";

#define RXp2 18
#define TXp2 17

//#define EEPROM_SIZE 512
#define ESPADC      4096.0 //the esp Analog Digital Conversion value
#define ESPVOLTAGE  3300   //the esp voltage supply value

#define MSG_BUFFER_SIZE (50)
#define MQTT_USER      "sistemtds"
#define MQTT_PASS      "sistemtds"

#define WIFI_SSID      "SamsungZFlip3"
#define WIFI_PASS      "bagus1993"

#define SOFT_APP_SSID  "sistemtds"
#define SOFT_APP_PASS  "sistemtds"

#define HOST           "wamot"

#define mqtt_server    "192.168.88.210"
#define port            1883
```

```

#define local_IP          "10.6.0.3"

WiFiClient espClient;
PubSubClient client(espClient);
AsyncWebServer server(80);

const IPAddress remote_ip(192, 168, 88, 210);

static WireGuard wg;
bool connectedOrNot = false;
char private_key[] = "KNvFtgbt5q/Pb0gkHa63gBA17nTNIo06R+wIMCvs9VU=";
char preshared_key[] = "Ut0qV1Ib/VWNs/BgmwUvPS6Mf/NeedeRIs0cPbFefjM=";
IPAddress local_ip(10, 244, 179, 5);
char public_key[] = "tKzCE8CKdfL45EpGIMyEtII/YfIr1eNfry+PKt0t1A8=";
char endpoint_address[] = "arducamp.duckdns.org";
int endpoint_port = 51820;
bool wifiConnection = false;

unsigned long lastTime = 0;
unsigned long timerDelay = 5000;

float temperature = 0;
float tdsValue = 0;
float kValue = 0;
float EcValue = 0;
float voltage = 0;
float pHValue = 0;
float DOValue = 0;
float TurbidityValue = 0;

String TDSMeterVar;
String TDSWarningVar;
String TempVar;
String pHVar;
String idVar;
String timeVar;
String EcVar;
String DOVar;
String TurbidityVar;

String TDSMeterTopic      = "sensors/TDSMeter/Number";
String TDSWarningTopic    = "sensors/TDSMeter/Warning";
String TempTopic          = "sensors/Temp/Number";
String PHTopic            = "sensors/PHMeter/Number";
String availabilityTopic  = "system/available";
String DoTopic            = "sensors/DO/Number";
String TurbidityTopic     = "sensors/Turbidity/Number";
String EcTopic            = "sensors/EC/Number";

```



```

String TDSWarning = "Normal";

bool firstConnectionWifi = false;
bool firstExecution = false;

String data;

String valueString1 = "0";
String valueString2 = "0";
String valueString3 = "0";
String valueString4 = "0";
String valueString5 = "0";
String valueString6 = "0";
String valueString7 = "0";
String valueString8 = "0";
String valueString9 = "0";

String strArr[9]; // Set the size of the array to equal the number of values
you will be receiving.

unsigned long prevMillis = 0;

```

```

void sendNumberTopic(float TDS, String Temp, String TDSWarning, String PH,
String DO, String Turbidity, String Ec) {
    String TDSString = String(TDS);
    String TDSWarningString = String(TDSWarning);
    String TempString = String(Temp);
    String PHString = String(PH);
    String DoString = String(DO);
    String TurbidityString = String(Turbidity);
    String EcString = String(Ec);

    client.publish(TDSMeterTopic.c_str(), TDSString.c_str());
    client.publish(TempTopic.c_str(), TempString.c_str());
    client.publish(TDSWarningTopic.c_str(), TDSWarningString.c_str());
    client.publish(PHTopic.c_str(), PHString.c_str());
    client.publish(DoTopic.c_str(), DoString.c_str());
    client.publish(TurbidityTopic.c_str(), TurbidityString.c_str());
    client.publish(EcTopic.c_str(), EcString.c_str());
}

void checkMessage(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] = ");
    Serial.println();
    String topicStr(topic);

```

```

}

void reconnectMQTT() {
  if (!wg.is_initialized()) {
    Serial.println("Attempting Wireguard Connection...");
    if (wg.begin(
      local_ip,
      private_key,
      endpoint_address,
      public_key,
      endpoint_port)) {
      Serial.println("Connected to Wireguard Server...");
    }
  }
}

if (!connectedOrNot or !client.connected()) {
  Serial.print("Ping MQTT Server ");
  Serial.println(remote_ip);
  if (Ping.ping(remote_ip)) {
    Serial.println("Success Pinging MQTT Server!");
    Serial.print("Attempting MQTT Connection...");
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);
    if (!client.connected()) {
      if (client.connect(clientId.c_str(), MQTT_USER, MQTT_PASS)) {
        Serial.println("connected");
        Serial.println("Sending Availability Confirmation");
        client.publish(availabilityTopic.c_str(), "online");
        connectedOrNot = true;
      } else {
        Serial.print("failed, rc=");
        Serial.print(client.state());
        Serial.println(" try again in 5 seconds");
        delay(5000);
        connectedOrNot = false;
        wg.end();
      }
    }
  }
} else {
  Serial.println("Error Pinging MQTT Server!");
  connectedOrNot = false;
  wg.end();
}
}
}
}

```

```

void setup() {

```

```

Serial.begin(115200);
Serial2.begin(115200, SERIAL_8N1, RXp2, TXp2);
mySerial.begin(9600);

WiFi.mode(WIFI_MODE_APSTA);
WiFi.softAP(SOFT_APP_SSID, SOFT_APP_PASS);

Serial.println();
Serial.println("Connecting to WiFi");

Serial.print("AP IP address: ");
IPAddress myIP = WiFi.softAPIP();

Serial.println(myIP);
if (!MDNS.begin(HOST)) {
  Serial.println("Error setting up MDNS responder!");
}
Serial.println("mDNS responder started");

server.on("/", HTTP_GET, [](AsyncWebServerRequest* request) {
  request->send(200, "text/plain", "Welcome to WaMoT");
});

AsyncElegantOTA.begin(&server, "wamot", "wamot"); // Start AsyncElegantOTA
server.begin();

//MQTT Parameter
client.setServer(mqtt_server, port);
client.setCallback(checkMessage);

firstConnectionWifi = false;
firstExecution = false;
//EEPROM.begin(EEPROM_SIZE); //Initialize EEPROM
delay(2500);
}

void loop() {
  // check if data is available
  if (Serial2.available() > 0) {
    valueString1 = "";

    // read the incoming string:
    String incomingString = Serial2.readStringUntil('\n');

    // prints the received data
    Serial.print("I received: ");
    Serial.println(incomingString);
  }
}

```

```

int stringStart = 0;
int arrayIndex = 0;

for (int i = 0; i < incomingString.length(); i++) {
  // Get character and check if it's our "special" character.
  if (incomingString.charAt(i) == '+') {
    // Clear previous values from array.
    strArr[arrayIndex] = "";

    // Save substring into array.
    strArr[arrayIndex] = incomingString.substring(stringStart, i);

    // Set new string starting point.
    stringStart = (i + 1);
    arrayIndex++;
  }
}

// Put values from the array into the variables.
valueString1 = strArr[0];
valueString2 = strArr[1];
valueString3 = strArr[2];
valueString4 = strArr[3];
valueString5 = strArr[4];
valueString6 = strArr[5];
valueString7 = strArr[6];
valueString8 = strArr[7];
valueString9 = strArr[8];
}

if (millis() - prevMillis > 1000) {
  prevMillis = millis();

  NextionData();
}

if (WiFi.status() != WL_CONNECTED && !firstConnectionWifi &&
!wg.is_initialized() && !connectedOrNot && !firstExecution) { //Registers the
connection
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  Serial.print("Connecting to ");
  Serial.println(WIFI_SSID);
  firstConnectionWifi = true;
} else if (WiFi.status() != WL_CONNECTED && firstConnectionWifi &&
!wg.is_initialized() && !connectedOrNot && !firstExecution) { //attempts
connection to WiFi
  delay(150);
  Serial.print(".");
}

```

```

static unsigned long timePoint = millis();
if (millis() - timePoint > 300000) {
    timePoint = millis();
    ESP.restart();
}
} else if (WiFi.status() == WL_CONNECTED && firstConnectionWifi &&
!wg.is_initialized() && !connectedOrNot && !firstExecution) { //Initializes
WiFi Connection
    Serial.println();
    Serial.println("WiFi Connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    delay(2500);
    configTime(9 * 60 * 60, 0, "ntp.jst.mfeed.ad.jp", "ntp.nict.jp",
"time.google.com");
    reconnectMQTT();
} else if (WiFi.status() == WL_CONNECTED && firstConnectionWifi &&
wg.is_initialized() && connectedOrNot && !firstExecution) {

    Serial.print("Temperature : ");
    Serial.print(valueString1);
    Serial.println(" °C");

    Serial.print("TDS          : ");
    Serial.print(tdsValue);
    Serial.println(" ppm");

    Serial.print("EC          : ");
    Serial.print(valueString9);
    Serial.print(F(" µS/cm"));
    Serial.println();

    Serial.print("pH          : ");
    Serial.print(valueString2);

    Serial.print("DO          : ");
    Serial.print(valueString5);
    Serial.println(" ppm");

    Serial.print("Turbidity   : ");
    Serial.print(valueString6);
    Serial.println(" ppm");

    Serial.println();

    delay(2500);

    if (tdsValue > 300) TDSWarning = "Exceed";

```

```

    else TDSWarning = "Normal";
    sendNumberTopic(tdsValue, valueString1, TDSWarning, valueString2,
valueString5, valueString6, valueString9);
    firstExecution = true;
    //postHTTPSensor(tdsValue, temperature, TDSWarning, pHValue);
} else { //WiFi gets disconnected
    WiFi.disconnect();
    firstConnectionWifi = firstExecution = connectedOrNot = false;
    wg.end();
}
}

void NextionData() {
    data = "tempWater.txt=\"" + String(valueString1) + "\"";
    mySerial.print(data);
    mySerial.write(0xff);
    mySerial.write(0xff);
    mySerial.write(0xff);

    data = "pHWater.txt=\"" + String(valueString2) + "\"";
    mySerial.print(data);
    mySerial.write(0xff);
    mySerial.write(0xff);
    mySerial.write(0xff);

    data = "doWater.txt=\"" + String(valueString5) + "\"";
    mySerial.print(data);
    mySerial.write(0xff);
    mySerial.write(0xff);
    mySerial.write(0xff);

    data = "turbidWater.txt=\"" + String(valueString6) + "\"";
    mySerial.print(data);
    mySerial.write(0xff);
    mySerial.write(0xff);
    mySerial.write(0xff);

    data = "ecWater.txt=\"" + String(valueString9) + "\"";
    mySerial.print(data);
    mySerial.write(0xff);
    mySerial.write(0xff);
    mySerial.write(0xff);
}

```

“WAMOT” Sistem Pemantau Limbah Cair Terintegrasi (Dissolved Oxygen, Turbidity, pH, Suhu, dan Total Dissolved Solids) berbasis Internet of Things terproteksi Wireguard

[Detail](#)[Pencipta](#)[Lampiran](#)[Approval History](#)[Pasca History](#)

Permohonan

Sertifikat (https://e-hakcipta.dgip.go.id/index.php/print_sertifikat/1ebabd069d83757943e73a88255fdd3d)

Nomor Aplikasi

EC00202384256

Nomor Sertifikat

000517209

Judul Ciptaan

“WAMOT” Sistem Pemantau Limbah Cair Terintegrasi (Dissolved Oxygen, Turbidity, pH, Suhu, dan Total Dissolved Solids) berbasis Internet of Things terproteksi Wireguard

Deskripsi

Sistem Pemantauan Otomatis terintegrasi pada Limbah Cair Berbasis Internet of Things (IoT) dengan terproteksi Wireguard merupakan sebuah sistem teknologi masa kini yang memungkinkan dapat menjadikan solusi praktis untuk melakukan monitoring terhadap sistem limbah. Sistem ini diberi nama WAMOT dengan kepanjangan Water Monitoring. Sensor-sensor yang terintegrasi akan mengirimkan data untuk melakukan monitoring melalui jaringan internet pada industri yang menghasilkan limbah cair. Sistem ini menggunakan ESP32-S3 sebagai mikrokontroler utama dalam melakukan komunikasi dengan server dan mikrokontroler Arduino Mega untuk melakukan pembacaan terhadap kualitas Total Dissolve Solid (TDS), suhu, Dissolved Oxygen (DO), turbidity, dan pH.

Pengguna

Sentra Kekayaan Intelektual STTNAS Yogyakarta

Tanggal Pengajuan

22-09-2023 08:44:19

Jenis permohonan

UMK, Lembaga Pendidikan, Lembaga Litbang Pemerintah

Jenis Ciptaan

Program Komputer

Tanggal Diumumkan

2023-09-12

Negara

Indonesia

Kota

SLEMAN

Biaya

Rp. 300.000

Status Pembayaran

Lunas Pengajuan Pencatatan Ciptaan

Status Penerimaan

Diterima

Billing Code

820230922173985



Persetujuan

Waktu Perlindungan

Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Catatan Penerimaan

Auto approve by system

