

SKRIPSI

**RANCANG BANGUN SISTEM MONITORING
KAPASITAS MULTIPLE TANGKI DAN
PENGISIAN AIR OTOMATIS BERBASIS IOT**



POLITEKNIK NEGERI BALI

Oleh :

I Putu Agastia Kama Suika

NIM. 1815344011

**PROGRAM STUDI D4 TEKNIK OTOMASI
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI BALI
2022**

LEMBAR PERSETUJUAN UJIAN SKRIPSI

RANCANG BANGUN SISTEM MONITORING KAPASITAS MULTIPLE TANGKI DAN PENGISIAN AIR OTOMATIS BERBASIS IOT

Oleh :

I Putu Agastia Kama Suika

NIM. 1815344011

Skripsi ini telah Melalui Bimbingan dan Pengujian Hasil, disetujui untuk
diujikan pada Ujian Skripsi

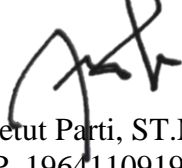
di

Program Studi D4 Teknik Otomasi
Jurusan Teknik Elektro - Politeknik Negeri Bali

Bukit Jimbaran, 22 September 2022

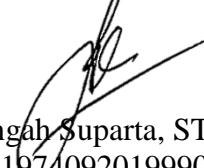
Disetujui Oleh :

Dosen Pembimbing 1:



I Ketut Parti, ST.MT.
NIP. 196411091990031002

Dosen Pembimbing 2:



I Nengah Suparta, ST.MT.
NIP. 197409201999031002

LEMBAR PENGESAHAN SKRIPSI

RANCANG BANGUN SISTEM MONITORING KAPASITAS MULTIPLE TANGKI DAN PENGISIAN AIR OTOMATIS BERBASIS IOT

Oleh :

I Putu Agastia Kama Suika

NIM. 1815344011

Skripsi ini sudah melalui Ujian Skripsi pada tanggal 22 September 2022,
dan sudah dilakukan Perbaikan untuk kemudian disahkan sebagai Skripsi

di

Program Studi D4 Teknik Otomasi
Jurusan Teknik Elektro - Politeknik Negeri Bali

Bukit Jimbaran, 29 September 2022

Disetujui Oleh :

Tim Penguji :

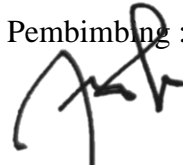


1. I Gede Suputra Widharma, ST.MT
NIP. 197212271999031004

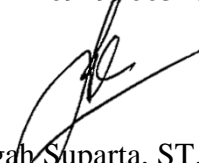


2. I Made Adi Yasa, S.Pd.M.Pd.
NIP. 198512102019031008

Dosen Pembimbing :



1. I Ketut Parti, ST.MT.
NIP. 196411091990031002



2. I Nengah Suparta, ST.MT.
NIP. 197409201999031002

Disahkan Oleh:



Ketua Jurusan Teknik Elektro

I I Wawan Raka Ardana, MT.

NIP. 196705021993031005

HALAMAN PERNYATAAN KEASLIAN KARYA SKRIPSI

Saya yang bertanda tangan di bawah ini, menyatakan bahwa Skripsi dengan judul:

RANCANG BANGUN SISTEM MONITORING KAPASITAS MULTIPLE TANGKI DAN PENGISIAN AIR OTOMATIS BERBASIS IOT

adalah asli hasil karya saya sendiri.

Dengan ini saya menyatakan bahwa dalam naskah Skripsi ini tidak terdapat karya orang lain yang pernah diajukan untuk memperoleh gelar di suatu perguruan tinggi, dan atau sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah Skripsi ini, dan disebutkan dalam daftar pustaka.

Apabila saya melakukan hal tersebut di atas, dengan ini saya menyatakan menarik Skripsi yang saya ajukan sebagai hasil karya saya.



ABSTRAK

Kegiatan monitoring persediaan air pada bak penampungan menjadi hal yang sangat penting mengingat ketersediaan air yang terbatas. Kegiatan Monitoring saat ini masih banyak yang menggunakan cara manual, sehingga akan memerlukan banyak tenaga dan waktu. Mengatasi kegiatan monitoring yang masih dilakukan secara manual maka penulis membuat suatu alat monitoring dan pengisian air otomatis berbasis *Internet Of Things*. Alat ini menggunakan sensor HC-SR04 yang berfungsi untuk mengatur level air secara otomatis pada suatu bak penampungan air serta *NodeMCU ESP8266* sebagai mikrokontrolernya. Cara kerja alat adalah apabila bak penampungan air dibawah 50% volume bak, maka sensor ultrasonic HC-SR04 akan mendeteksi ketinggian air dan memberikan sinyal ke *NodeMCU ESP8266* untuk menghidupkan pompa pengisi bak penampungan air dan mengirimkan data ketinggian air pada *Webserver*. Apabila bak penampungan air diatas 90% bak penampung maka sensor ultrasonic HC-SR04 akan mendeteksi ketinggian air dan memberikan sinyal ke *NodeMCU ESP8266* untuk mematikan pompa pengisi bak penampungan air secara otomatis dan mengirimkan data ketinggian air pada *Webserver*, sehingga dapat memudahkan dalam memonitoring ketersediaan air.

Kata kunci : NodeMCU ESP8266,HC-SR04, Internet Of Things, Webserver

ABSTRACT

Water supply monitoring activities in reservoirs become very important given the limited availability of water. Monitoring activities currently still mostly use manual methods, so it will require a lot of energy and time. Overcoming monitoring activities that are still carried out manually, the author wishes to create an Internet of Things-based automatic monitoring and water filling tool. This tool uses the HC-SR04 sensor which functions to adjust the water level automatically in the water reservoir and the NodeMCU ESP8266 as the microcontroller. The working of the tool is that if the water tank is under 50%, the ultrasonic sensor HC-SR04 will detect the water level and give a signal to the NodeMCU ESP8266 to turn on the water reservoir filling pump and send the water level data to the Webserver. If the water reservoir is above 90%, the ultrasonic sensor HC-SR04 will detect the water level and give a signal to the NodeMCU ESP8266 to turn off the water reservoir filling pump automatically and send water level data to the Webserver, making it easier to monitoring of water inventory .

Keywords : NodeMCU ESP8266, HC-SR04, Internet Of Things, Webserver

KATA PENGANTAR

Puji syukur penulis panjatkan kehadapan Ida Sang Hyang Widhi Wasa TuhanYang Maha Esa, karena berkat rahmat-Nya yang telah memberikan nikmat kesehatan dan hikmat kepada penulis sehingga dapat menyelesaikan Skripsi dengan baik sesuai dengan waktu yang telah direncanakan yang berjudul : **“Rancang Bangun Sistem Monitoring Kapasitas Multiple Tangki Dan Pengisian Air Otomatis Berbasis IoT”**.

Penulis menyadari bahwa Skripsi ini tidak dapat tersusun tanpa mendapat bantuan dan bimbingan dari berbagai pihak oleh karena itu dalam kesempatan yang baik ini penulis menyampaikan ucapan terima kasih setinggi-tingginya dan tak terhingga kepada yang terhormat:

1. Bapak I Nyoman Abdi, SE, M.eCom selaku Direktur Politeknik Negeri Bali.
2. Bapak Ir. I Wayan Raka Ardana, MT. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Bali.
3. Bapak Ida Bagus Irawan Purnama, ST., M.Sc., Ph.D selaku Ketua Program Studi Teknik Otomasi Politeknik Negeri Bali.
4. Bapak I Ketut Parti, ST., MT selaku Dosen pembimbing I dan Bapak I Nengah Suparta, ST., MT. selaku Dosen pembimbing II yang telah meluangkan waktu untuk memberikan masukan, bimbingan, dan motivasi yang membangun kepada penulis hingga Skripsi ini dapat terselesaikan dengan baik.
5. Staf pengajar Program Teknik Otomasi Politeknik Negeri Bali atas segala ilmu, masukan dan bantuan yang telah diberikan kepada penulis. Bapak dan Ibu Dosen Program Teknik Otomasi Politeknik Negeri Bali yang telah memberikan pengajaran mata kuliah dari semester I sampai semester VII, serta seluruh staff Program Teknik Otomasi yang telah membantu kelancaran proses perkuliahan.
6. Kepada keluarga yang sangat penulis cintai dan hormati yang tak henti-hentinya memberikan dukungan, doa, nasehat, dan motivasi hingga sampai pada detik ini penulis tetap kuat dan bersemangat dalam menyelesaikan studi.
7. Kepada sahabat-sahabat penulis yang membantu dan selalu memberikan dukungan dan motivasi. Terima kasih atas kasih sayang dan dukungan yang diberikan hingga saat ini.
8. Rekan-rekan mahasiswa dan semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah membantu dalam penyelesaian skripsi.

Penulis menyadari skripsi ini masih banyak kekurangan dan jauh dari kesempurnaan, karena itu kritik dan saran yang bersifat membangun sangat diharapkan guna penyempurnaan skripsi ini dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Badung, 29 September 2022

I Putu Agastia Kama Suika

DAFTAR ISI

LEMBAR PERSETUJUAN UJIAN SKRIPSI.....	ii
LEMBAR PENGESAHAN SKRIPSI	iii
HALAMAN PERNYATAAN KEASLIAN KARYA SKRIPSI.....	iii
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	4
2.1 Penelitian Sebelumnya	4
2.2 Landasan Teori.....	5
2.2.1 Internet Of Things (IoT)	5
2.2.2 <i>Web Server</i>	6
2.2.3 NodeMCU.....	6
2.2.4 Sensor.....	7
2.2.5 Solenoid <i>Valve</i> DC 12V.....	9
2.2.6 Relay	9
2.2.7 Pompa Air DC 12V.....	10

BAB III METODE PENELITIAN	11
3.1 Rancangan Sistem	11
3.1.1 Rancangan Bentuk Alat	11
3.1.2 Rancangan <i>Hardware</i>	12
3.1.3 Diagram Blok.....	15
3.1.4 Kebutuhan Fungsional Sistem	15
3.2 Implementasi Sistem	16
3.2.1 Perancangan <i>Hardware</i>	16
3.2.2 Perancangan <i>Software</i>	16
3.2.3 Diagram Alir Sistem	17
BAB IV HASIL DAN ANALISA	19
4.1 Hasil Implementasi.....	19
4.1.1 Implementasi Alat.....	19
4.1.2 Implementasi Sistem.....	19
4.1.3 Implementasi Program	20
4.2 Hasil Pengujian	24
4.2.1 Pengujian Sensor HC-SR04.....	24
4.2.2 Pengujian Sensor Flow water	25
4.2.3 Pengujian Monitoring Level Air Otomatis	26
4.2.4 Pengujian Kontrol Valve	27
BAB V PENUTUP	28
5.1 Kesimpulan	28
5.2 Saran.....	29
DAFTAR PUSTAKA	30
LAMPIRAN.....	31

DAFTAR GAMBAR

Gambar 2.1 <i>Internet Of Things</i>	5
Gambar 2.2 <i>Web Server</i>	6
Gambar 2.3 <i>NodeMCU ESP ESP8266</i>	7
Gambar 2.4 Sensor HC-SR04	8
Gambar 2.5 Sensor <i>Flow water</i>	8
Gambar 2.6 Solenoid <i>valve</i> DC 12V	9
Gambar 2.7 Modul Relay	10
Gambar 2.8 Pompa Air DC 12V	10
Gambar 3.1 Rancangan Bentuk Alat	11
Gambar 3.2 Blok Diagram Hardware	12
Gambar 3.3 Rangkaian Dasar Transmitter sensor HC-SR04	12
Gambar 3.4 Rangkaian Dasar Receiver sensor HC-SR04	13
Gambar 3.5 Schematic <i>NodeMCU ESP8266</i>	13
Gambar 3.6 Rangkaian Modul Relay 4 Channel	14
Gambar 3.7 Blok Diagram	15
Gambar 3.8 Perancangan Hardware	16
Gambar 3.9 Software Arduino IDE	17
Gambar 3.10 <i>Web Server</i>	17
Gambar 3.11 Diagram Alir Sistem	18
Gambar 4.1 Implementasi Alat	19
Gambar 4.2 Implementasi Sistem	20

DAFTAR TABEL

Tabel 3.1 Daftar Kebutuhan Hardware	15
Tabel 3.2 Daftar Kebutuhan Software	16
Tabel 4.1 Hasil Pengujian Sensor HC-SR04	24
Tabel 4.2 Hasil Pengujian 1 Water Flow Sensor	25
Tabel 4.3 Hasil Pengujian 2 Water Flow Sensor	25
Tabel 4.4 Hasil Pengujian 3 Water Flow Sensor	26
Tabel 4.5 Hasil Pengujian Monitoring Level Air Otomatis Pada Alat	26
Tabel 4.6 Hasil Pengujian Kontrol Valve Pada Alat	27

DAFTAR LAMPIRAN

Lampiran 1 Pengujian Sensor	31
Lampiran 2 Pengujian Sistem	32
Lampiran 3 Program	33

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi semakin terus meningkat. Kita sebagai masyarakat harus lebih pintar dalam menggunakan teknologi agar bisa tetap mengikuti perkembangan jaman dan tidak tertinggal jauh. Seperti yang kita lihat selama ini, proses kemajuan teknologi sangat membantu manusia untuk meringankan pekerjaannya. Kemajuan teknologi bukan berarti semata-mata dapat menghilangkan produk lama yang sudah dianggap ketinggalan jaman, namun dengan adanya teknologi dapat meningkatkan nilai tambah dari produk itu sendiri. Sehingga tentu saja teknologi sangat penting kaitannya dengan kehidupan kita sehari-hari, karena selain dapat memudahkan segala hal yang kita sedang lakukan, teknologi juga membuat semuanya menjadi praktis dan lebih efisien waktu.

Semakin hari diiringi dengan perkembangan teknologi dan penambahan penduduk, kebutuhan akan air juga semakin meningkat sehingga ketersediaan air tetap harus selalu ada baik di rumah tangga, tempat umum, perkantoran ataupun industri. Ini menyebabkan peran penampung air menjadi penting dan diperlukan suatu mekanisme pengukuran untuk mengetahui ketersediaan air pada wadah tersebut. Seringkali mekanisme tersebut masih berupa cara-cara manual, misalnya dengan mendatangi, melihat atau melakukan pengukuran langsung pada tempat penampung air tersebut. Cara ini merupakan cara yang mudah dan murah, tetapi akan sedikit sulit jika misalnya letak penampungan air tersebut jauh dan sulit dijangkau, misalnya di puncak bangunan atau di tebing sungai[1].

Permasalahan umum yang sering terjadi adalah kita tidak mengetahui kapasitas air yang terdapat pada tangki, apalagi biasanya masyarakat meletakkan atau memasang tangki air di atas rumah atau pada tempat yang sulit dijangkau. Sehingga ketika ingin menggunakan air dan kita tidak mengetahui air pada tangki habis, maka dari itu kita harus menunggu beberapa saat agar tangki air terisi kembali, hal tersebut menghabiskan waktu yang cukup lama. Untuk saat ini, masyarakat di Indonesia khususnya di Bali hanya menggunakan 1 tangki air untuk memenuhi semua kebutuhan air dirumahnya. Namun jika dipikirkan lebih jauh lagi, justru ini dapat menjadi masalah. Masalah lainnya adalah dapat terjadinya pemborosan air dan listrik yang dikarenakan kita lupa untuk mematikan pompa air sehingga air akan meluap dan terbuang percuma. Maka dari itu akan menghabiskan banyak waktu dan tenaga untuk selalu mengecek tangki apakah masih

terisi air atau tidak. Dari permasalahan-permasalahan tersebut, membuat saya tertarik untuk membuat prototipe alat monitoring dan pengisian tangki air otomatis yang berjudul “Rancang Bangun Sistem Monitoring Kapasitas Multiple Tangki dan Pengisian Air Otomatis Berbasis IOT” guna untuk meminimalisir terjadinya masalah-masalah yang dapat merugikan masyarakat. Sistem monitoring dan pengisian air otomatis tersebut menggunakan komponen-komponen seperti mikrokontroler *NodeMCU ESP8266*, solenoid *valve*, sensor HC-SR04, dan sensor *flow water*. Kegiatan monitoring dapat dilaksanakan didalam ruangan rumah sehingga kita tidak akan menghabiskan banyak waktu dan tenaga untuk selalu mengecek kapasitas tangki air dan menghidupkan pompa secara manual. Tampilan hasil datanya akan ditampilkan di *Web Server* yang outputnya berupa sebuah grafik maupun angka.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan di atas, maka rumusan masalah yang dapat ditarik adalah sebagai berikut:

- a. Bagaimana cara merancang sistem monitoring kapasitas multiple tangki dan pengisian air otomatis berbasis IoT?
- b. Bagaimana cara memonitoring *level* air dan kontrol menggunakan *web server*?
- c. Bagaimana mengevaluasi cara kerja sistem alat?

1.3 Batasan Masalah

Sesuai dengan penelitian yang dilakukan, maka penulis memberikan batasan masalah sebagai berikut:

- a. Membahas tentang sistem hanya difungsikan untuk memonitoring 2 tangki air.
- b. Membahas tentang mikrokontroler yang digunakan adalah *NodeMCU ESP8266*.
- c. Sensor yang digunakan dalam sistem adalah sensor ultrasonik.
- d. Merancang sistem monitoring alat menggunakan *Web server*.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari penulisan dan penyusunan skripsi ini adalah sebagai berikut:

- a. Dapat mengetahui cara merancang sistem monitoring kapasitas multiple tangki dan pengisian air otomatis berbasis IoT.
- b. Dapat mengetahui cara memonitoring *level* air dan kontrol menggunakan *web server*.
- c. Dapat mengevaluasi sistem cara kerja alat.

BAB IV

HASIL DAN ANALISA

4.1 Hasil Implementasi

4.1.1 Implementasi Alat

Setelah melakukan perancangan *hardware* dan *software*, bentuk alat dari “Rancang Bangun Sistem Monitoring Kapasitas Multiple Tangki Dan Pengisian Air Otomatis Berbasis IoT” dapat dilihat pada gambar 4.1. Alat ini dibuat dengan bahan besi dan 2 buah galon yang berukuran 12 liter. Komponen sudah terangkai sesuai dengan perencanaan *hardware*.

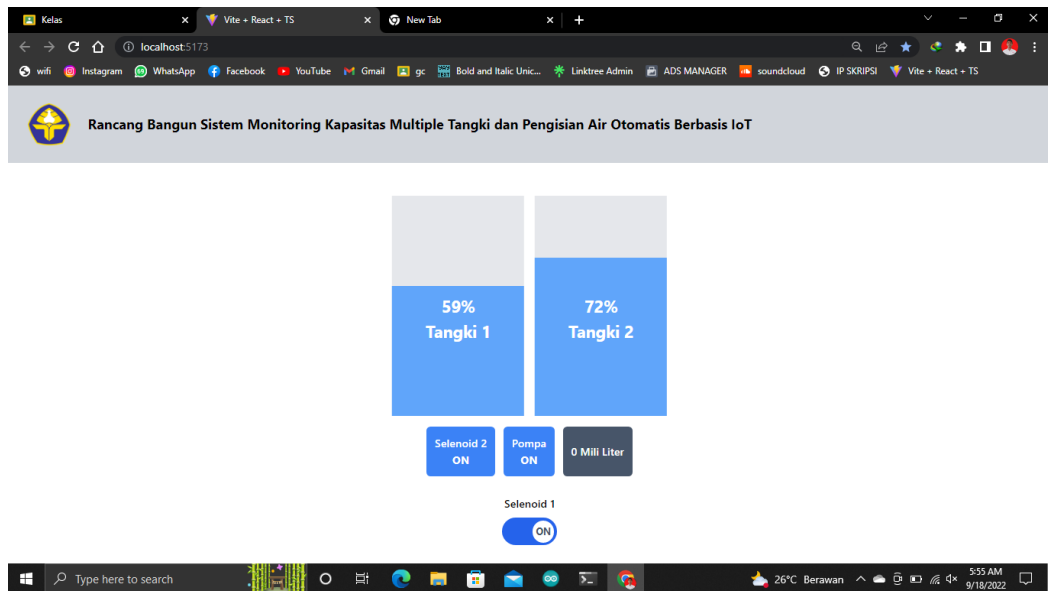


Gambar 4.1 Implementasi Alat

4.1.2 Implementasi Sistem

Sistem kerja dari alat ini dimulai ketika perangkat dinyalakan, kemudian sensor akan membaca keadaan yang akan di kirim ke *NodeMCU ESP8266*. Sensor HC-SR04 pada tangki 1 dan tangki 2 akan mendeteksi level air pada bak penampung, ketika level air pada tangki 1 di bawah 50% volume bak penampung maka solenoid *valve* 2 akan menyala. Begitu juga, ketika level air pada tangki 2 dibawah 50% volume bak penampung maka pompa akan menyala. Ataupun sebaliknya ketika ketinggian air pada tangki 1 sudah diatas 50% volume bak penampung maka solenoid *valve* 2 akan mati. Begitu juga pada tangki 2 jika ketinggian air sudah diatas 50% volume bak penampung maka pompa akan otomatis mati. Solenoid *valve* 1 pada keluaran tangki 1 bisa dihidupkan secara

manual melalui *web server*. Sensor *Flow water* berfungsi untuk menghitung jumlah debit air yang digunakan. Setelah *NodeMCU ESP8266* selesai melakukan proses, selanjutnya data hasil pengukuran sensor akan ditampilkan pada *Web server* agar dapat dimonitoring dan dikontrol secara *real time*.



Gambar 4.2 Implementasi Sistem

4.1.3 Implementasi Program

A. Program 1

Pada Program ini menjelaskan tentang library yang digunakan. Tujuan dari memasukan *library* adalah untuk menginstall *code* dari komponen dan fungsi pada program.

```
#include <Arduino.h>
#include <ArduinoJson.h>
#include <ESP8266WebServer.h>
#include <ESP8266WiFi.h>
#include <ESP8266mDNS.h>
```

B. Program 2

Pada program ini menjelaskan tentang pin-pin yang digunakan untuk komponen pada *NodeMCU ESP8266*.

```
#define HallS_PIN 4 // D2
#define trig_1 0 // D3
#define echo_1 12 // D6
#define trig_2 15 // D8
#define echo_2 13 // D7

int selenoid_1 = 5; // D1
int selenoid_2 = 2; // D4
int pompa = 14; //D5
```

C. Program 3

Pada program ini menjelaskan tentang menghubungkan *NodeMCU ESP8266* dengan koneksi internet

```
#define SSID "Multiverse"
#define PASSWORD "Strange05"

    WiFi.begin(SSID, PASSWORD);
    Serial.print("Connecting to WiFi ..");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print('.');
        delay(1000);
    }
    Serial.println(WiFi.localIP());
    // The ESP8266 tries to reconnect automatically when the connection is lost
    WiFi.setAutoReconnect(true);
    WiFi.persistent(true);
}
```

D. Program 4

Pada program ini menjelaskan tentang mendeklarasikan variable pin yang digunakan agar sesuai dengan *input* ataupun *output*.

```
void setup()
{
    Serial.begin(115200);
    Serial.println("Serial Initialized");

    // deklarasi pin
    pinMode(solenoid_2, OUTPUT);
    pinMode(solenoid_1, OUTPUT);
    pinMode(pompa, OUTPUT);

    pinMode(trig_1, OUTPUT);
    pinMode(echo_1, INPUT);

    pinMode(trig_2, OUTPUT);
    pinMode(echo_2, INPUT);

    pinMode(HallS_PIN, INPUT_PULLUP);
}
```

E. Program 5

Pada program ini menjelaskan tentang kontrol lever air secara otomatis pada kedua bak penampung air serta berisi kalibrasi dari kedua sensor HC-SR04.

```
int calculateLevel1()
{
    int duration_1, distance_1;

    duration_1 = pulseIn(echo_1, HIGH);
    distance_1 = (duration_1 / 2) / 29.1;
    jarakPandang_1 = jarakSensor_1 - distance_1;
    // RUMUS MERUBAH PERSEN
    return (jarakPandang_1 / 25) * 100; //(22 MERUPAKAN BATAS BISA DIUBAH)
}
void sensorHandler_1()
{
    digitalWrite(trig_1, HIGH);
    delay(50);
    digitalWrite(trig_1, LOW);

    level_1 = calculateLevel1();
    if (level_1 >= 0) {
        Serial.print("Level Air 1 : ");
        Serial.print(level_1);
        Serial.println("%");

        // KONTROL LEVEL AIR 1
        if (level_1 <= 50) {
            Serial.println("Solenoid 2 : ON");
            digitalWrite(solenoid_2, LOW);
        } else if (level_1 >= 90) {
            Serial.println("Solenoid 2: OFF");
            digitalWrite(solenoid_2, HIGH);
        }
    }
}
int calculateLevel2()
{
    int duration_2, distance_2;

    duration_2 = pulseIn(echo_2, HIGH);
    distance_2 = (duration_2 / 2) / 29.1;
    jarakPandang_2 = jarakSensor_2 - distance_2;

    // RUMUS MERUBAH PERSEN
    return (jarakPandang_2 / 25) * 100;
}
void sensorHandler_2()
{
    digitalWrite(trig_2, HIGH);
    delay(50);
    digitalWrite(trig_2, LOW);

    level_2 = calculateLevel2();
    if (level_2 >= 0) {
        Serial.print("Level Air 2 : ");
        Serial.print(level_2);
        Serial.println("%");
        // KONTROL LEVEL AIR 2
        if (level_2 <= 50) {
            Serial.println("Pompa : ON");
            digitalWrite(pompa, LOW);
        } else if (level_2 >= 90) {
            Serial.println("Pompa : OFF");
            digitalWrite(pompa, HIGH);
        }
    }
}
```

F. Program 6

Pada program ini menjelaskan tentang cara mendapatkan data dari sensor agar dapat di tampilkan pada *web server*.

```
void getData()
{
    setCrossOrigin();
    DynamicJsonDocument doc(512);
    doc["kapasitas_tangki_1"] = level_1 >= 0 ? level_1 : 0;
    doc["kapasitas_tangki_2"] = level_2 >= 0 ? level_2 : 0;
    doc["flow_meter"] = flow;
    doc["solenoid_1_status"] = digitalRead(solenoid_1) == 0 ? true : false;
    doc["solenoid_2_status"] = digitalRead(solenoid_2) == 0 ? true : false;
    doc["pompa_status"] = digitalRead(pompa) == 0 ? true : false;

    Serial.print(F("Stream..."));
    String buf;
    serializeJson(doc, buf);
    server.send(200, "application/json", buf);
    Serial.println(F("done."));
}
```

G. Program 7

Pada Program ini menjelaskan respon apabila *router* yang diminta pada *client* tidak tersedia.

```
// Manage not found URL
void handleNotFound()
{
    String message = "File Not Found\n\n";
    message += "URI: ";
    message += server.uri();
    message += "\nMethod: ";
    message += (server.method() == HTTP_GET) ? "GET" : "POST";
    message += "\nArguments: ";
    message += server.args();
    message += "\n";
    for (uint8_t i = 0; i < server.args(); i++) {
        message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
    }
    server.send(404, "text/plain", message);
}
```

H. Program 8

Pada program ini menjelaskan tentang setup server agar bisa mengirimkan *request* dari *client* ke *NodeMCU ESP8266 RestApi*

```
void setupServer()
{
  // Activate mDNS this is used to be able to connect to the server
  // with local DNS hostmane esp8266.local
  if (MDNS.begin("esp8266")) {
    Serial.println("MDNS responder started");
  }

  // Set server routing
  restServerRouting();
  // Set not found response
  server.onNotFound(handleNotFound);
  // Start server
  server.begin();
  Serial.println("HTTP server started");
}
void sendCrossOriginHeader()
{
  Serial.println(F("sendCORSHeader"));
  server.sendHeader(F("access-control-allow-credentials"), F("false"));
  setCrossOrigin();
  server.send(204);
}
void setCrossOrigin()
{
  server.sendHeader(F("Access-Control-Allow-Origin"), F("*"));
  server.sendHeader(F("Access-Control-Max-Age"), F("600"));
  server.sendHeader(F("Access-Control-Allow-Methods"), F("PUT,POST,GET,OPTIONS"));
  server.sendHeader(F("Access-Control-Allow-Headers"), F("*"));
};
```

4.2 Hasil Pengujian

4.2.1 Pengujian Sensor HC-SR04

Pengujian dengan membandingkan data pengukuran secara manual dengan pengukuran *sensor*. Data *error* dan presentase akurasi yang di dapat bisa menggunakan perhitungan presentase kesalahan dan ketepatan, dengan menggunakan persamaan (1) dan (2), memperoleh rata – rata *error* sensor 3.86% dan presentase akurasi dari sensor 95.47

Tabel 4.1 Hasil Pengujian Sensor HC-SR04

No	Pengukuran (Manual)/cm	Pengukuran (Sensor)/cm	Nilai Error (%)	Presentase Akurasi (%)
1	7	6.5	7.1	92.9
2	10	9.3	7	93
3	15	14.4	4	96
4	17	16.1	5.2	94.8
5	20	19.5	2.5	97.5
6	25	24.3	2.8	97.2
7	27	26.1	3.3	96.7
8	30	29.2	2.6	90.4
9	35	34.4	1.7	98.3
10	37	36.1	2.4	97.9
Rata - Rata			3.86	95.47

4.2.2 Pengujian Sensor Flow water

Pengujian dengan membandingkan data *supply* air manual dengan pengukuran *sensor*. Data *error* dan presentase akurasi yang di dapat bisa menggunakan perhitungan persentase kesalahan dan ketepatan, dengan menggunakan persamaan (3) dan (4).

Tabel 4.2 Hasil Pengujian 1 Water Flow Sensor

No	Supply Air Manual(ml)	Hasil Pengukuran Sensor (ml)	Nilai Error (%)	Presentase Akurasi (%)
1	1 000	951	4.9	95.1
2	1 000	955	4.5	95.5
3	1 000	943	5.7	94.3
4	1 000	950	5	95
5	1 000	953	4.7	95.3
6	1 000	944	5.6	94.4
7	1 000	953	4.7	95.3
8	1 000	957	4.3	95.7
9	1 000	965	3.5	96.5
10	1 000	959	4.1	95.9
Rata - Rata			4.7	95.3

Tabel 4.3 Hasil Pengujian 2 Water Flow Sensor

No	Supply Air Manual(ml)	Hasil Pengukuran Sensor (ml)	Nilai Error (%)	Presentase Akurasi (%)
1	2 000	1890	5.5	94.5
2	2 000	1897	5.1	94.9
3	2 000	1901	4.9	95.1
4	2 000	1896	5.2	94.8
5	2 000	1891	5.4	94.6
6	2 000	1905	4.7	95.3
7	2 000	1912	4.4	95.6
8	2 000	1898	5.1	94.9
9	2 000	1907	4.6	95.4
10	2 000	1905	4.7	95.3
Rata - Rata			4.96	95.04

Tabel 4.4 Hasil Pengujian 3 Water Flow Sensor

No	Supply Air Manual(ml)	Hasil Pengukuran Sensor (ml)	Nilai Error (%)	Presentase Akurasi (%)
1	3 000	2982	0.6	99.4
2	3 000	2987	0.4	99.6
3	3 000	2981	0.6	99.4
4	3 000	2970	1	99
5	3 000	2985	0.5	99.5
6	3 000	2971	0.9	99.1
7	3 000	2987	0.4	99.6
8	3 000	2981	0.6	99.4
9	3 000	2989	0.3	99.7
10	3 000	2979	0.7	99.3
Rata - Rata			0.6	99.4

4.2.3 Pengujian Monitoring Level Air Otomatis

Dari pengujian kinerja sistem alat yang dilakukan, pembacaan dari sensor yang diuji sebagai monitoring level air sudah berjalan dan sudah menampilkan nilai dari objek yang diteliti. Jika dilihat dari hasil pengujian kinerja alat akan berkerja jika kondisi air 0% - 90% maka solenoid *valve* 2 dan pompa menyala, selanjutnya jika kondisi air diatas 90% maka solenoid *valve* 2 dan pompa akan mati. Sistem yang telah dibuat sudah berjalan sesuai dengan perancangan awal.

Tabel 4.5 Hasil Pengujian Monitoring Level Air Otomatis Pada Alat

No	Level Air Tangki 1	Level Air Tangki 2	Selenoid Valve 2	Pompa	Keterangan
1	15%	15%	ON	ON	Berhasil
2	25%	25%	ON	ON	Berhasil
3	35%	35%	ON	ON	Berhasil
4	45%	45%	ON	ON	Berhasil
5	50%	50%	ON	ON	Berhasil
6	60%	60%	ON	ON	Berhasil
7	80%	80%	ON	ON	Berhasil
8	90%	90%	OFF	OFF	Berhasil
9	93%	93%	OFF	OFF	Berhasil
10	95%	95%	OFF	OFF	Berhasil

4.2.4 Pengujian Kontrol Valve

Dari pengujian kinerja sistem alat yang dilakukan, kontrol dari komponen yang diuji sebagai kontrol solenoid sudah berjalan dan sudah menampilkan *output* dari objek yang diteliti. Jika dilihat dari hasil pengujian kinerja alat akan berkerja jika push button pada tampilan *web server* ditekan maka solenoid akan terbuka dan dapat mengaliri air. Hasil yang ditampilkan pada *web server* sudah berjalan dengan baik dan sistem sudah berjalan sesuai perencanaan awal.

Tabel 4.6 Hasil Pengujian Kontrol Valve Pada Alat

No	Tombol Valve1 Websever	Relay 3	Keterangan
1	ON	ON	Berhasil
2	OFF	OFF	Berhasil

BAB V

PENUTUP

5.1 Kesimpulan

Dari pengujian yang telah dilakukan menunjukkan bahwa penelitian tentang sistem monitoring dan pengisian air otomatis sesuai dengan yang diinginkan baik dari alat yang digunakan dan sistem monitoring dan kontrol menggunakan *web server* sudah bisa dikatakan berjalan dengan baik. Berdasarkan proses implementasi dan pengujian alat dapat disimpulkan bahwa:

1. Sistem monitoring dan pengisian air otomatis ini menggunakan dua sensor HC-SR04 untuk mengetahui kapasitas air disetiap tangki, *NodeMCU ESP8266* sebagai mikrokontrolernya, sensor *flow water* untuk menghitung debit air yang keluar. Hasil dari pembacaan sensor akan ditampilkan pada *web server* yang datanya berupa angka maupun grafik secara *realtime*.
2. Perancangan monitoring level air menggunakan sensor HC-SR04 yang dimana sensor akan membaca kapasitas air pada setiap tangki, dan kemudian hasil pengukuran sensor akan dikirimkan ke *web server*. Untuk memonitoring debit air yang keluar menggunakan sensor *flow water* dimana jika sensor dialiri air maka sensor akan langsung membaca aliran air yang melewati sensor kemudian hasil pembacaan akan dikirimkan ke *web server*. Kemudian untuk melakukan sistem kontrol menggunakan relay yang dimana *output* dari relay berisi solenoid *valve* untuk supply air. Monitoring dan kontrol dapat dilihat pada tampilan *web server*.
3. Hasil pengujian sistem secara keseluruhan sudah berjalan sangat baik karena saat keadaan level air yang belum terpenuhi sistem mampu melakukan pengisian air secara otomatis sampai mencapai level air yang diinginkan. Pembacaan penggunaan air dapat terbaca dengan baik ketika water flow dialiri air pada saat solenoid terbuka kemudian hasilnya dapat di lihat pada tampilan *web server*. Untuk sistem kontrol solenoid juga sudah berjalan dengan baik dimana sistem dapat melakukan kontrol dari tampilan *web server*.

5.2 Saran

Berdasarkan analisis hasil berbagai pengujian, adapun saran dari penulis untuk pengembangan sistem monitoring dan kontrol menggunakan SCADA ini bahwa:

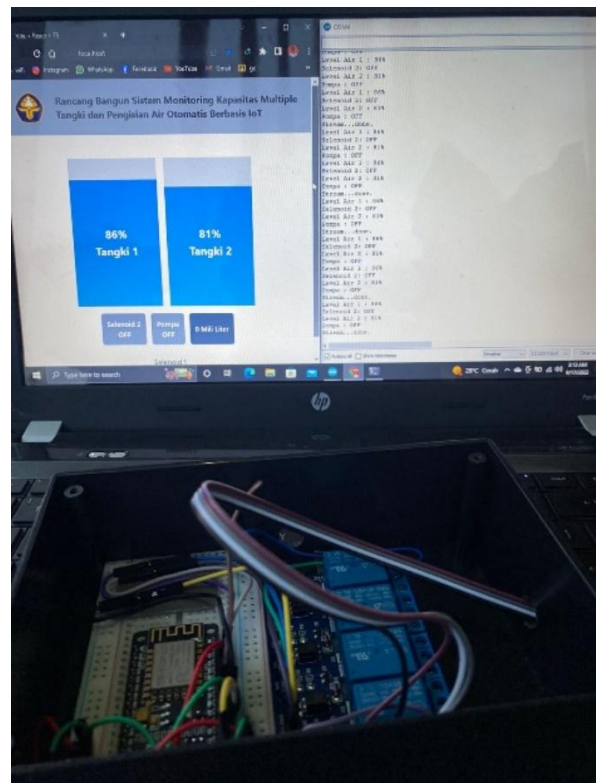
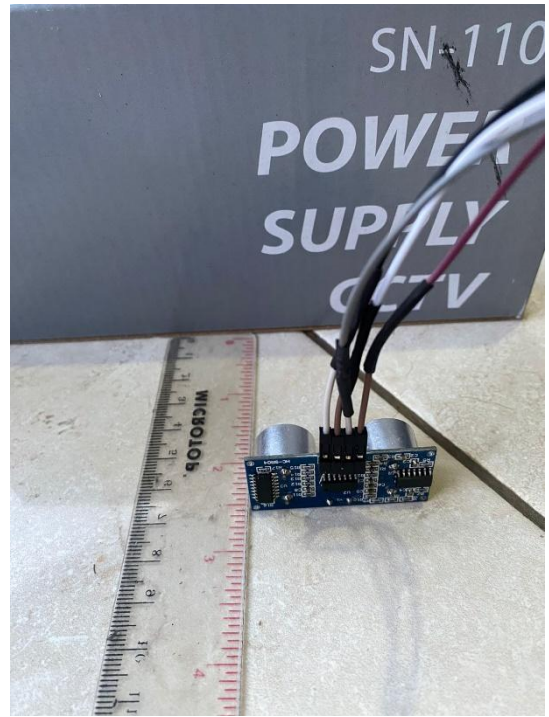
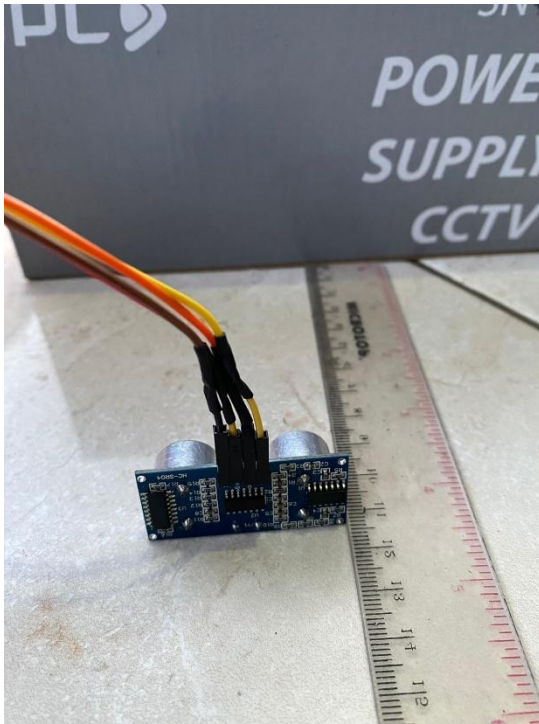
1. Perlu ditambahkan penyimpanan data hasil dari pembacaan sensor *flow water* agar dapat mengetahui pemakaian air yang di gunakan pada hari – hari sebelumnya.
2. Menambahkan baterai atau panel surya sebagai alternatif daya listrik saat pasokan daya listrik tidak ada.
3. Perlu di tambahkan *ball valve* pada keluaran tangki untuk mencegah saat *solenoid valve* sewaktu – waktu rusak atau tidak bekerja.

DAFTAR PUSTAKA

- [1] V. Yuliaminuddin, J. Bintoro, and F. Teknik, "Prototipe Sistem Kontrol dan Monitoring pada Tangki Air Berbasis Internet of Things," *J. Autocracy*, vol. 1, pp. 27–34, 2020.
- [2] I. Muklisin, "Pendeteksi Volume Tandon Air Secara Otomatis Menggunakan Sensor Ultrasonic Berbasis Arduino Uno R3," *J. Qua Tek.*, vol. 7, no. 2, pp. 55–65, 2017, doi: 10.35457/quateknika.v7i2.242.
- [3] I. Gunawan, T. Akbar, and M. Giyandhi Ilham, "Prototipe Penerapan Internet Of Things (Iot) Pada Monitoring Level Air Tandon Menggunakan Nodemcu Esp8266 Dan Blynk," *Infotek J. Inform. dan Teknol.*, vol. 3, no. 1, pp. 1–7, 2020, doi: 10.29408/jit.v3i1.1789.
- [4] R. H. Hardyanto, "Konsep Internet Of Things Pada Pembelajaran Berbasis Web," *J. Din. Inform.*, vol. 6, no. 1, pp. 87–97, 2017.
- [5] E. B. Lewi, U. Sunarya, and D. N. Ramadan, "Sistem Monitoring Ketinggian Air Berbasis Internet of Things Menggunakan Google Firebase," *Univ. Telkom, D3 Tek. Telekomun.*, vol. 1, no. 1, pp. 1–8, 2017.
- [6] N. H. L. Dewi, M. F. Rohmah, and S. Zahara, "Prototype Smart Home Dengan Modul Nodemcu Esp8266 Berbasis Internet of Things (Iot)," *J. Tek. Inform.*, p. 3, 2019.
- [7] A. Amin, "Monitoring Water Level Control Berbasis Arduino Uno Menggunakan Lcd Lm016L," *J. Ilm. Tek. Elektro*, vol. 1, no. 2, pp. 41–52, 2018.
- [8] A. Suharjono, L. N. Rahayu, and R. Afwah, "Aplikasi Sensor Flow Water Untuk Mengukur Penggunaan Air Pelanggan Secara Digital Serta Pengiriman Data Secara Otomatis Pada PDAM Kota Semarang," *Tek. Elektro, Politek. negeri Semarang*, vol. Vol.13, no. 1, pp. 7–12, 2015, [Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiH7I3Om7DwAhUObn0KHZGYC0sQFjABegQIAhAD&url=https%3A%2F%2Fjurnal.polines.ac.id%2Findex.php%2Ftele%2Farticle%2Fview%2F151%2F143&usg=AOvVaw1cjVXdPv7BTpCqce_sf4P
- [9] F. Fadhilillah., A. G. Putrada., and S. Prabowo., "Water Level Controller pada Pemandian Pintar Menggunakan Fuzzy Logic dan Solenoid Valve," *e-Proceeding Appl. Sci.*, vol. 6, no. 2, pp. 9658–9666, 2019.
- [10] O. T. Kwaar and X. B. N. Najooan, "Rancang Bangun Aplikasi Pengendalian Saluran Air dalam Rumah Berbasis Internet of Things," *J. Tek. Elektro dan Komput.*, pp. 1–10, 2020.
- [11] K. B. Kusuma, C. G. I. Partha, and I. W. Sukerayasa, "Perancangan Sistem Pompa Air Dc Dengan Plts 20 kWp Tianyar Tengah Sebagai Suplai Daya Untuk Memenuhi Kebutuhan Air," *J. SPEKTRUM*, vol. 7, no. 2, pp. 46–56, 2020.

LAMPIRAN

Lampiran 1 Pengujian Sensor



Lampiran 2 Pengujian Sistem



Lampiran 3 Program

```
#include <Arduino.h>

#include <ArduinoJson.h>

#include <ESP8266WebServer.h>

#include <ESP8266WiFi.h>

#include <ESP8266mDNS.h>

#define SSID "RedmiNote8"

#define PASSWORD "09090909"

#define Halls_PIN 4 // D2

#define trig_1 0 // D3

#define echo_1 12 // D6

#define trig_2 15 // D8

#define echo_2 13 // D7

int selenoid_1 = 5; // D1

int selenoid_2 = 2; // D4

int pompa = 14; //D5

int valueLS = HIGH;

int jumlahPulse = 0;

float flow = 0;
```

```

long distance_1, duration_1;

long distance_2, duration_2;

float jarakSensor_1, jarakPandang_1;

float jarakSensor_2, jarakPandang_2;

int level_1 = 0;

int level_2 = 0;

ESP8266WebServer server(80);

// pembacaan flow

void int_handler()

{

    int tombolCS = digitalRead(Halls_PIN);

    if (valueLS == HIGH && tombolCS == LOW) {

        jumlahPulse++;

        Serial.print("Jumlah Pulse: ");

        Serial.println(jumlahPulse);

        flow = .00225 * jumlahPulse;

        Serial.print(flow);

        Serial.println("L");

        Serial.println();

    }

    valueLS = tombolCS;

```

```
}

void setup()
{
  Serial.begin(115200);

  Serial.println("Serial Initialized");

  // deklarasi pin

  pinMode(solenoid_2, OUTPUT);

  pinMode(solenoid_1, OUTPUT);

  pinMode(pompa, OUTPUT);

  pinMode(trig_1, OUTPUT);

  pinMode(echo_1, INPUT);

  pinMode(trig_2, OUTPUT);

  pinMode(echo_2, INPUT);

  pinMode(Halls_PIN, INPUT_PULLUP);

  setupWifi();

  setupServer();

  // JARAK SENSOR KE GALON

  jarakSensor_1 = 26;
}
```



```

    jarakSensor_2 = 26;

    digitalWrite(solenoid_1, HIGH);

    digitalWrite(solenoid_2, HIGH);

    digitalWrite(pompa, HIGH);
}

void loop()
{
    int_handler();

    sensorHandler_1();

    sensorHandler_2();

    server.handleClient();

    MDNS.update();
}

int calculateLevel1()
{
    int duration_1, distance_1;

    duration_1 = pulseIn(echo_1, HIGH);

    distance_1 = (duration_1 / 2) / 29.1;

    jarakPandang_1 = jarakSensor_1 - distance_1;

    // RUMUS MERUBAH PERSEN

    return (jarakPandang_1 / 25) * 100; //(22 MERUPAKAN BATAS BISA DIUBAH)
}

```

```

}

void sensorHandler_1()

{

    digitalWrite(trig_1, HIGH);

    delay(50);

    digitalWrite(trig_1, LOW);

    level_1 = calculateLevel1();

    if (level_1 >= 0) {

        Serial.print("Level Air 1 : ");

        Serial.print(level_1);

        Serial.println("%");

        // KONTROL LEVEL AIR 1

        if (level_1 <= 50) {

            Serial.println("Solenoid 2 : ON");

            digitalWrite(solenoid_2, LOW);

        } else if (level_1 >= 90) {

            Serial.println("Solenoid 2: OFF");

            digitalWrite(solenoid_2, HIGH);

        }

    }

}

```

```

int calculateLevel2()
{
    int duration_2, distance_2;

    duration_2 = pulseIn(echo_2, HIGH);
    distance_2 = (duration_2 / 2) / 29.1;
    jarakPandang_2 = jarakSensor_2 - distance_2;

    // RUMUS MERUBAH PERSEN
    return (jarakPandang_2 / 25) * 100;
}

void sensorHandler_2()
{

    digitalWrite(trig_2, HIGH);
    delay(50);
    digitalWrite(trig_2, LOW);

    level_2 = calculateLevel2();
    if (level_2 >= 0) {
        Serial.print("Level Air 2 : ");
        Serial.print(level_2);
        Serial.println("%");

        // KONTROL LEVEL AIR 2
    }
}

```

```

    if (level_2 <= 50) {
        Serial.println("Pompa : ON");
        digitalWrite(pompa, LOW);
    } else if (level_2 >= 90) {
        Serial.println("Pompa : OFF");
        digitalWrite(pompa, HIGH);
    }
}

void setupWifi()
{
    WiFi.mode(WIFI_STA);
    WiFi.begin(SSID, PASSWORD);
    Serial.print("Connecting to WiFi ..");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print('.');
        delay(1000);
    }
    Serial.println(WiFi.localIP());
    // The ESP8266 tries to reconnect automatically when the connection is lost
    WiFi.setAutoReconnect(true);
    WiFi.persistent(true);
}

```

```

// Serving Hello world

/*

LOW = ON = 0

HIGH = OFF = 1

*/

void handleSolenoidSwitch(int pompa)

{

    setCrossOrigin();

    DynamicJsonDocument jsonBuffer(512);

    DynamicJsonDocument doc(512);

    Serial.println(F("handlePompaSwitch"));

    String postBody = server.arg("plain");

    Serial.println(postBody);

    // Parse JSON object

    auto jsonData = deserializeJson(jsonBuffer, postBody);

    if (jsonData) {

        Serial.println(F("Parsing failed!"));

        doc["error"] = "parsing failed";

        String buf;

        serializeJson(doc, buf);

```

```

        return server.send(200, "application/json", buf);
    }

    // Decode JSON/Extract values
    bool status = jsonBuffer["status"].as<bool>();

    if (status) {
        digitalWrite(pompa, LOW);
    } else {
        digitalWrite(pompa, HIGH);
    }

    doc["status"] = digitalRead(pompa);
    String buf;
    serializeJson(doc, buf);

    return server.send(200, "application/json", buf);
}

void getData()
{
    setCrossOrigin();
    DynamicJsonDocument doc(512);

```

```

doc["kapasitas_tangki_1"] = level_1 >= 0 ? level_1 : 0;
doc["kapasitas_tangki_2"] = level_2 >= 0 ? level_2 : 0;
doc["flow_meter"] = flow;
doc["solenoid_1_status"] = digitalRead(solenoid_1) == 0 ? true : false;
doc["solenoid_2_status"] = digitalRead(solenoid_2) == 0 ? true : false;
doc["pompa_status"] = digitalRead(pompa) == 0 ? true : false;

Serial.print(F("Stream..."));

String buf;

serializeJson(doc, buf);

server.send(200, "application/json", buf);

Serial.println(F("done.));

}

// Define routing

void restServerRouting()

{
    server.on("/", HTTP_GET, []() { server.send(200, F("text/html"), F("Welcome to the
REST Web Server")); });

    server.on(F("/handle-solenoid-1"), HTTP_OPTIONS, sendCrossOriginHeader);

    server.on(F("/handle-solenoid-1"), HTTP_POST, []() {
handleSolenoidSwitch(solenoid_1); });

    server.on(F("/get-data"), HTTP_GET, getData);

}

```

```

// Manage not found URL

void handleNotFound()

{

    String message = "File Not Found\n\n";

    message += "URI: ";

    message += server.uri();

    message += "\nMethod: ";

    message += (server.method() == HTTP_GET) ? "GET" : "POST";

    message += "\nArguments: ";

    message += server.args();

    message += "\n";

    for (uint8_t i = 0; i < server.args(); i++) {

        message += " " + server.argName(i) + ": " + server.arg(i) + "\n";

    }

    server.send(404, "text/plain", message);

}

void setupServer()

{

    // Activate mDNS this is used to be able to connect to the server

    // with local DNS hostmane esp8266.local

    if (MDNS.begin("esp8266")) {

        Serial.println("MDNS responder started");

    }

}

```



```

// Set server routing

restServerRouting();

// Set not found response

server.onNotFound(handleNotFound);

// Start server

server.begin();

Serial.println("HTTP server started");
}

void sendCrossOriginHeader()

{

Serial.println(F("sendCORSHeader"));

server.sendHeader(F("access-control-allow-credentials"), F("false"));

setCrossOrigin();

server.send(204);

}

void setCrossOrigin()

{

server.sendHeader(F("Access-Control-Allow-Origin"), F("*"));

server.sendHeader(F("Access-Control-Max-Age"), F("600"));

server.sendHeader(F("Access-Control-Allow-Methods"),
F("PUT,POST,GET,OPTIONS"));

server.sendHeader(F("Access-Control-Allow-Headers"), F("*"));

};

```