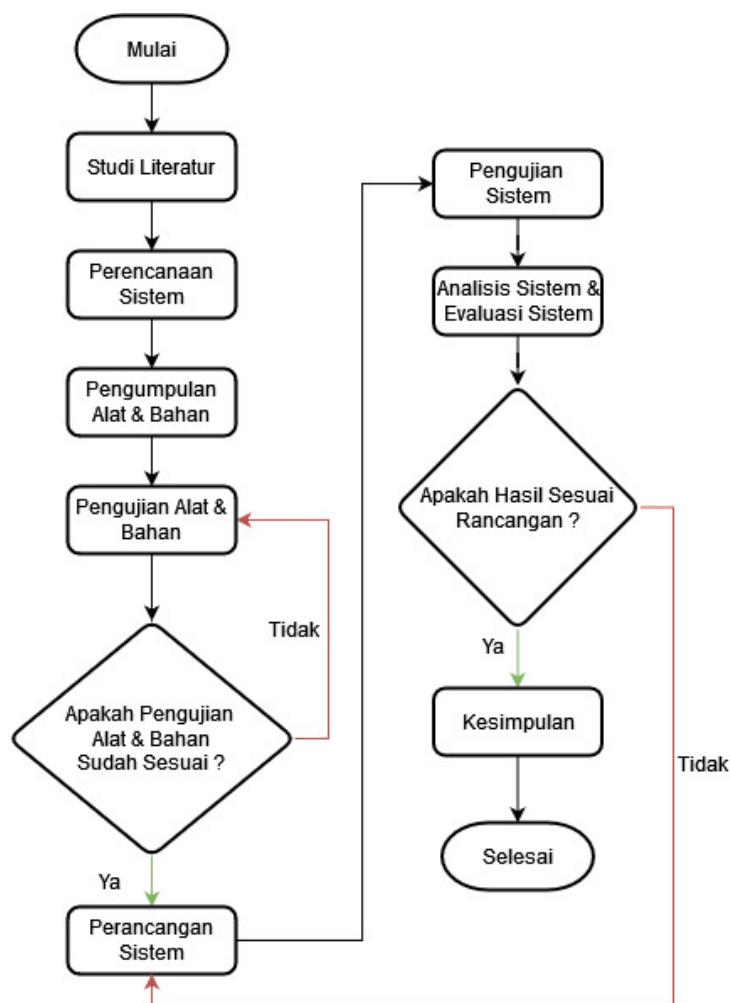


## BAB III

### METODE PENELITIAN

#### 3.1. *Flowchart Penelitian*

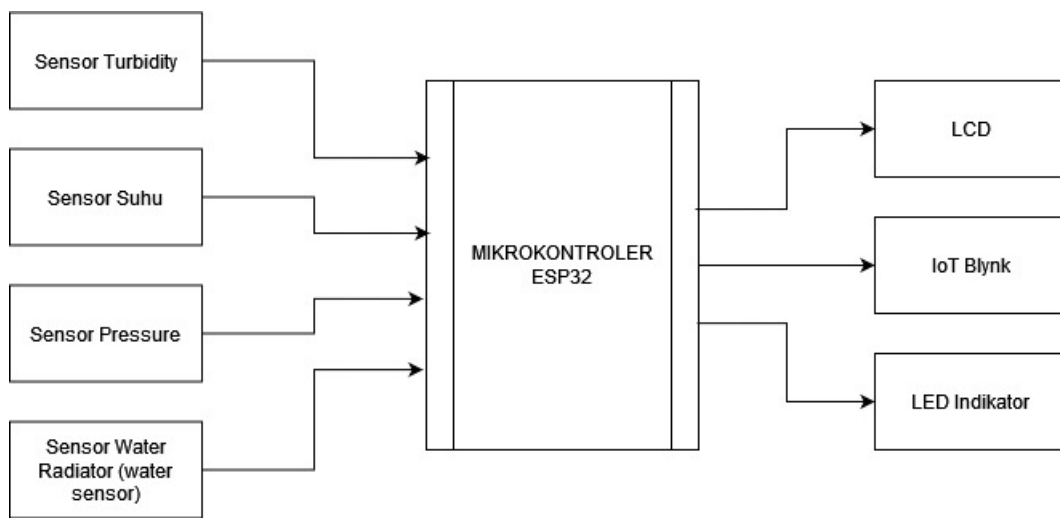
*Flowchart* pada penelitian ini meliputi beberapa tahapan penelitian kerja, ada beberapa tahapan yang harus dilakukan secara berurut sesuai dengan tujuan yang akan dilakukan agar penelitian dapat dijalankan sesuai dengan harapan. Berikut tahapan penelitian kerja dapat dilihat pada Gambar 3.1.



Gambar 3. 1. *Flowchart Penelitian*

Pada Gambar 3.1. Menunjukkan alur dari tahapan penelitian yang akan dilakukan, pada setiap langkah – langkah alur dari tahapan penelitian tersebut menggambarkan alur sesuai dengan urutan yang tercantum, agar proses penelitian dapat tercapai sesuai dengan perencanaan yang diharapkan peneliti.

### 3.1.1. Perencanaan Sistem



Gambar 3. 2. Diagram Arsitektur Perencanaan Sistem

Pada Gambar 3.2. Diagram ini menggambarkan arsitektur sebuah sistem yang terdiri dari beberapa komponen:

1. Sensor *Turbidity*: Sensor untuk mengukur kekeruhan oli.
2. Sensor *Temperature*: Sensor untuk mengukur suhu oli.
3. Sensor *Pressure*: Sensor untuk mengukur tekanan oli.
4. Sensor *Water Radiator (Water Sensor)*: Sensor untuk mengukur air atau cairan pada radiator.

Semua sensor tersebut terhubung ke MIKROKONTROLER ESP32, yang berfungsi sebagai pusat pengolahan data dari sensor-sensor.

Dari mikrokontroler ESP32, data kemudian dikirimkan ke:

1. LCD: Untuk menampilkan informasi secara langsung pada layar moonitor.
2. IoT Blink: Untuk menghubungkan ke *internet of things*, dan menampilkan informasi secara jauh atau informasi dapat berjalan di server.
3. LED Indikator: Untuk memberikan indikasi / indikator visual dengan cahaya warna dari LED.

Secara keseluruhan, diagram ini menggambarkan arsitektur sistem yang mengintegrasikan berbagai sensor dengan mikrokontroler ESP32 sebagai pusat pengolahan data, dan mengirimkan informasi ke berbagai output seperti LCD, IoT, dan LED indikator.

### 3.1.2. Kebutuhan Alat dan Bahan

Pada penelitian ini akan dilakukan pengumpulan, pemilihan alat dan bahan seperti komponen – komponen sensor dan lainnya sesuai dengan kebutuhan pada penelitian ini. Adapun pengumpulan dan pemilihan komponen-komponen yang dibutuhkan, seperti pada tabel 3.1 berikut.

Tabel 3. 1. Daftar Komponen – komponen

| No | Nama                     |
|----|--------------------------|
| 1  | Mikrokontroller ESP32    |
| 2  | LCD Oled 0.96 inch       |
| 3  | Sensor <i>Water</i>      |
| 4  | Sensor <i>Temperatur</i> |
| 5  | Sensor <i>Turbidity</i>  |
| 6  | Sensor <i>Pressure</i>   |

### 3.1.3. Pengujian Alat dan Bahan

Pada tahap ini yaitu pengujian alat dan bahan yang telah disiapkan untuk setiap komponen – komponen di uji, apakah komponen tersebut berfungsi sebagaimana mestinya berdasarkan standar pabrikan ataupun referensi lainnya untuk membantu saat proses uji komponen.

#### 3.1.3.1 Pengujian Mikrokontroler ESP32



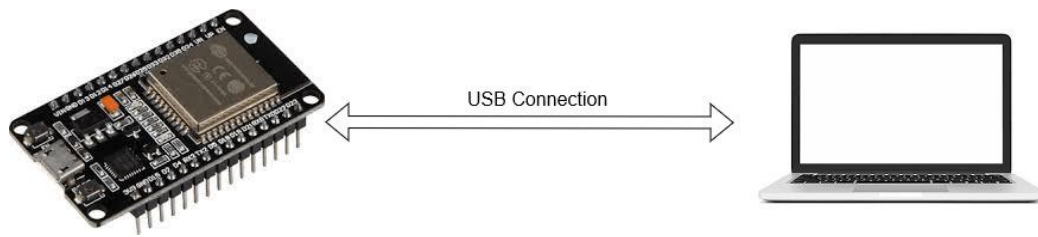
Gambar 3. 3. Pengujian Mikrokontroler ESP32

Pada Gambar 3.3. menggambarkan proses alur pengujian ESP32. Berikut penjelasan dan langkah-langkahnya:

1. Mulai: Proses dimulai, yaitu menyiapkan komponen yang akan digunakan.

2. Wiring ESP32 ke PC/Laptop: Langkah pertama adalah menghubungkan ESP32 ke komputer atau laptop.
3. Menjalankan Code: Setelah terhubung, kode program diupload dan dijalankan pada ESP32 menggunakan aplikasi arduino IDE dari PC/Laptop.
4. Apakah ESP32 berfungsi dengan Baik?: Ini adalah tahap pengujian untuk memeriksa apakah ESP32 berfungsi sebagaimana mestinya.
5. Jika Ya: Jika ESP32 berfungsi dengan baik, alur akan menuju ke "Selesai", menandakan bahwa proses telah berhasil diselesaikan atau pengujian selesai.
6. Jika Tidak: Jika ESP32 tidak berfungsi dengan baik, alur kembali ke langkah "Wiring ESP32 ke PC/Laptop". Ini menunjukkan bahwa proses harus diulang dari awal, untuk memeriksa koneksi atau melakukan *troubleshooting*.
7. Selesai: Proses berakhir setelah ESP32 berfungsi dengan baik.

Pada gambar 3.3. ini mengilustrasikan siklus sederhana untuk menyiapkan, menguji, dan memastikan ESP32 berfungsi dengan benar, dengan opsi untuk mengulangi proses jika diperlukan, agar mencapai hasil sesuai dengan pengujian.



Gambar 3. 4. Koneksi Esp32 PC/Laptop

Gambar 3.4. ini menunjukkan skema koneksi sederhana antara dua perangkat:

1. Di sebelah kiri terdapat sebuah mikrokontroler, yang tampak seperti ESP32 atau perangkat sejenis. Ini adalah papan pengembangan kecil dengan *chip mikrokontroler* dan pin-pin I/O.
2. Di sebelah kanan ada gambar laptop atau komputer.
3. Di antara kedua perangkat tersebut terdapat tanda panah dua arah dengan label "USB Connection", yang menunjukkan bahwa mikrokontroler dan komputer terhubung melalui kabel USB.

Gambar 3.4. ini mengilustrasikan cara dasar menghubungkan mikrokontroler ke komputer, untuk tujuan pengujian komponen dengan pemrograman, debugging, atau transfer data antara kedua perangkat tersebut.

*Script/Program Dasar Untuk Mikrokontroller ESP32*



```

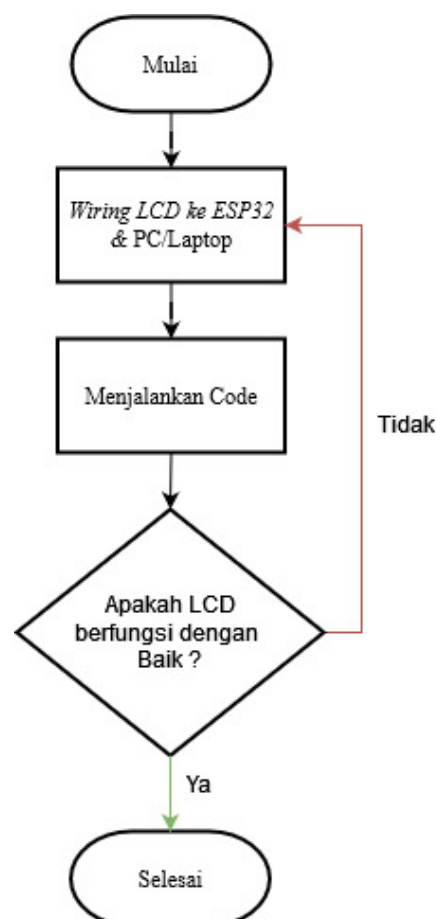
1 void setup() {
2   Serial.begin(115200);
3   Serial.print("Hallo");
4   delay(2000);
5 }
6
7 void loop() {
8   Serial.println("Nama : Thea Dadan Hoeruman");
9   Serial.println("NPM : 187002001");
10  Serial.println("Tugas Akhir Teknik Elektro");
11  delay(1000);
12 }
13

```

Gambar 3. 5. *Script* Mikrokontroller ESP32

Gambar 3.5. Menunjukkan kode uji coba awal (pengujian komunikasi serial) pada mikrokontroler ESP32 menggunakan Arduino IDE versi 2.1.0.

### 3.1.3.2 Pengujian LCD Liquid Crystal Display

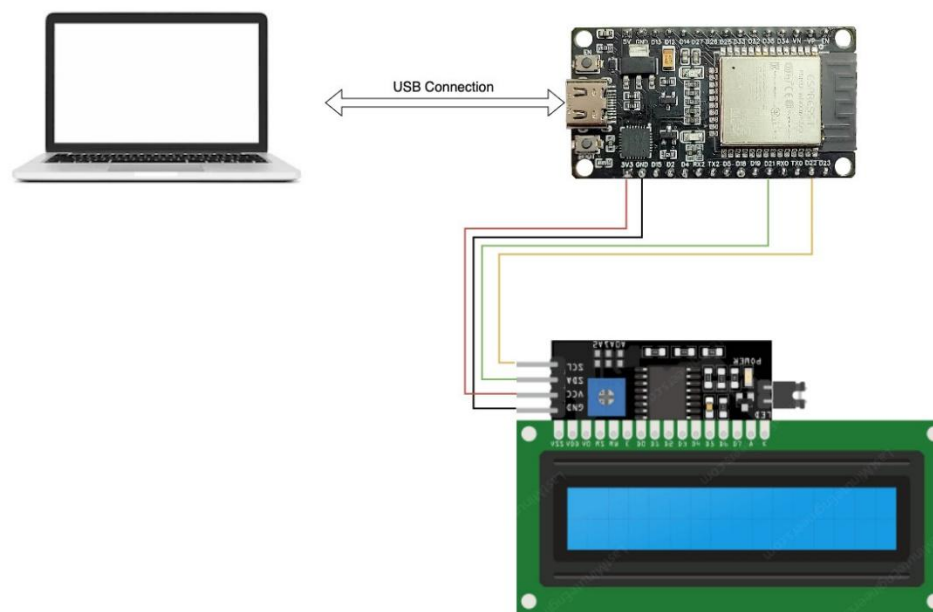


Gambar 3. 6. Pengujian LCD (*Liquid Crystal Display*)

Pada Gambar 3.6. Menggambarkan proses alur pengujian LCD. Berikut penjelasan dan langkah-langkahnya:

1. Mulai: Proses dimulai, yaitu menyiapkan komponen yang akan digunakan.
2. Wiring LCD ke ESP32 dan PC/Laptop: Langkah pertama adalah menghubungkan LCD ke ESP32 dan komputer atau laptop.
3. Menjalankan Code: Setelah terhubung, kode program diupload dan dijalankan pada ESP32 menggunakan aplikasi arduino IDE dari PC/Laptop.
4. Apakah LCD berfungsi dengan Baik?: Ini adalah tahap pengujian untuk memeriksa apakah LCD berfungsi sebagaimana mestinya.
5. Jika Ya: Jika LCD berfungsi dengan baik, alur akan menuju ke "Selesai", menandakan bahwa proses telah berhasil diselesaikan atau pengujian selesai.
6. Jika Tidak: Jika LCD tidak berfungsi dengan baik, alur kembali ke langkah "Wiring LCD ke ESP32 dan PC/Laptop". Ini menunjukkan bahwa proses harus diulang dari awal, untuk memeriksa koneksi atau melakukan *troubleshooting*.
7. Selesai: Proses berakhir setelah LCD berfungsi dengan baik.

Pada gambar 3.6. ini mengilustrasikan siklus sederhana untuk menyiapkan, menguji, dan memastikan LCD berfungsi dengan benar, dengan opsi untuk mengulangi proses jika diperlukan, agar mencapai hasil sesuai dengan pengujian.



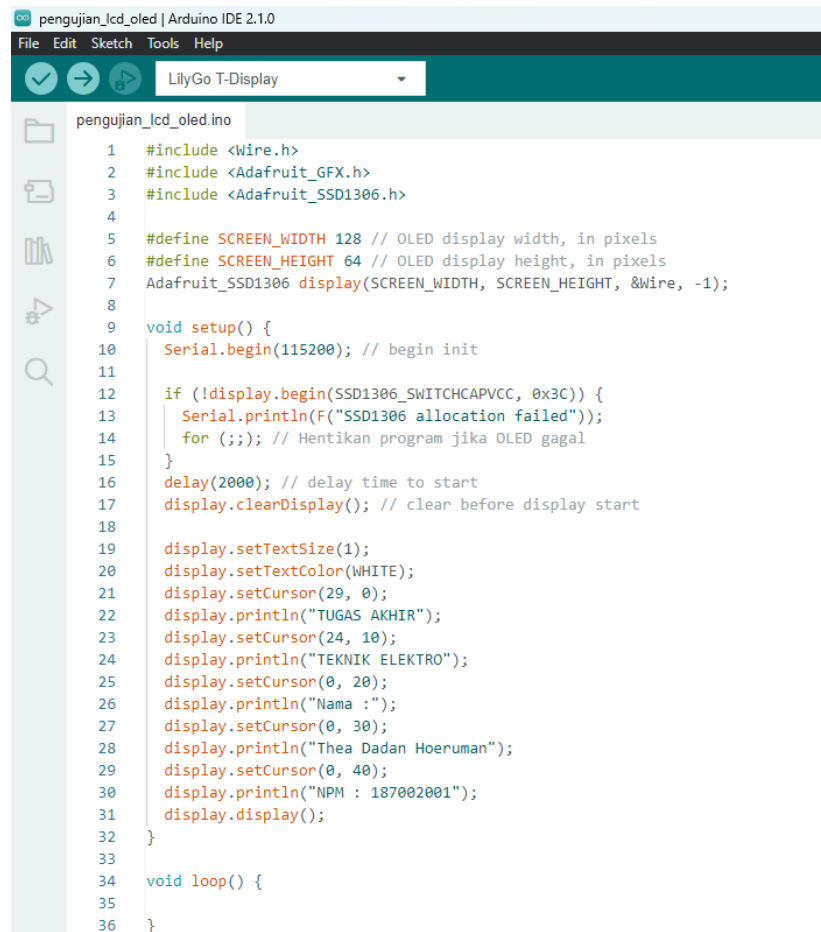
Gambar 3. 7. Koneksi LCD (*Liquid Crystal Display*) dan PC/Laptop

Gambar 3.7. Menunjukkan skema koneksi antara beberapa perangkat elektronik:

1. Di sebelah kiri terdapat laptop atau komputer.
2. Di tengah ada sebuah mikrokontroler, kemungkinan ESP32 atau board sejenis.
3. Di bagian bawah terdapat modul *display* LCD.
4. Laptop dan mikrokontroler terhubung melalui koneksi USB, ditunjukkan dengan panah dua arah berlabel "*USB Connection*".
5. Mikrokontroler terhubung ke *display* LCD melalui beberapa kabel berwarna (merah, kuning, hitam dan hijau) merah untuk sumber tegangan, kuning untuk data scl, hijau untuk data sda dan hitam adalah *ground* atau negatif.

Skema ini menggambarkan setup tipikal untuk pengujian komponen dimana mikrokontroler diprogram melalui komputer dan mengendalikan display LCD untuk menampilkan informasi atau data.

### Script/Program Dasar untuk LCD (*Liquid Crystal Display*)



```

pengujian_lcd_oled | Arduino IDE 2.1.0
File Edit Sketch Tools Help
LilyGo T-Display

pengujian_lcd_oled.ino
1  #include <Wire.h>
2  #include <Adafruit_GFX.h>
3  #include <Adafruit_SSD1306.h>
4
5  #define SCREEN_WIDTH 128 // OLED display width, in pixels
6  #define SCREEN_HEIGHT 64 // OLED display height, in pixels
7  Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
8
9  void setup() {
10     Serial.begin(115200); // begin init
11
12     if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
13         Serial.println(F("SSD1306 allocation failed"));
14         for (;;); // Hentikan program jika OLED gagal
15     }
16     delay(2000); // delay time to start
17     display.clearDisplay(); // clear before display start
18
19     display.setTextSize(1);
20     display.setTextColor(WHITE);
21     display.setCursor(29, 0);
22     display.println("TUGAS AKHIR");
23     display.setCursor(24, 10);
24     display.println("TEKNIK ELEKTRO");
25     display.setCursor(0, 20);
26     display.println("Nama :");
27     display.setCursor(0, 30);
28     display.println("Thea Dadan Hoeruman");
29     display.setCursor(0, 40);
30     display.println("NPM : 187002001");
31     display.display();
32 }
33
34 void loop() {
35
36 }

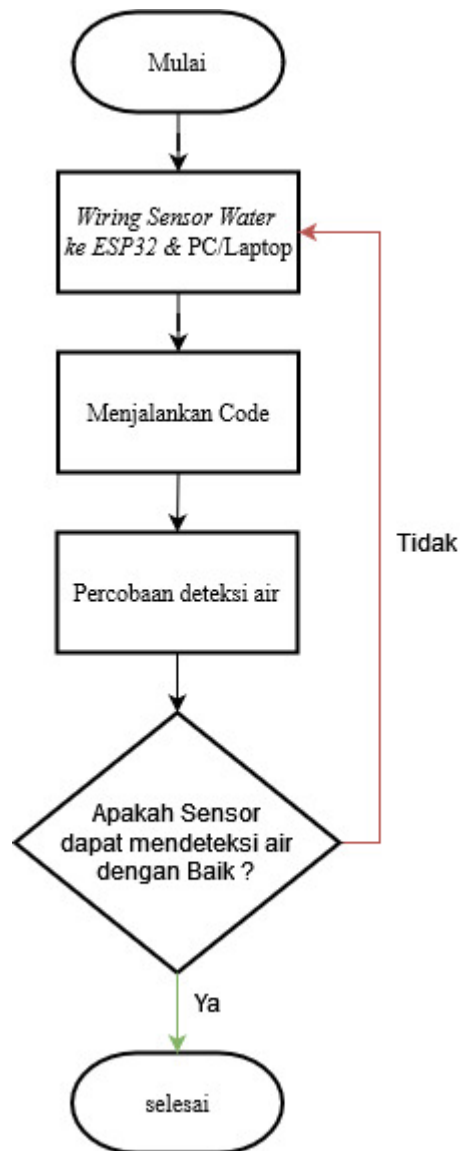
```

Gambar 3. 8. Script LCD (*Liquid Crystal Display*)

Gambar 3.8. Menunjukkan kode pengujian LCD (*Liquid Crystal Display*)

dengan ESP32 menggunakan Arduino IDE 2.1.0. Menghubungkan ESP32 dengan LCD (*Liquid Crystal Display*) menampilkan teks ke layar LCD (*Liquid Crystal Display*) saat startup.

### 3.1.3.3 Pengujian Sensor *Water*



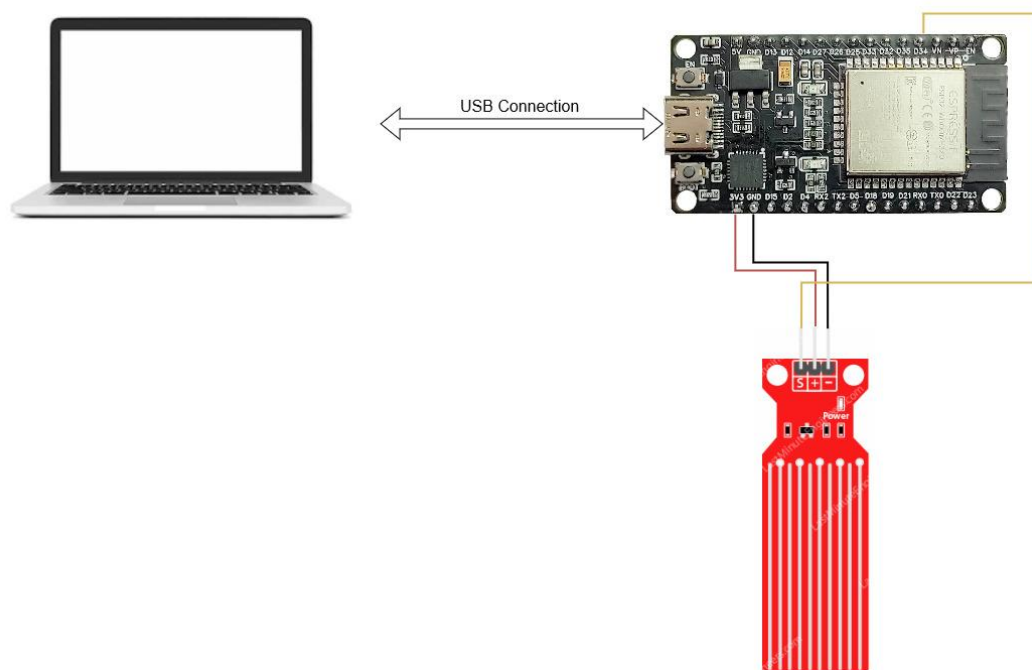
Gambar 3. 9. Pengujian Sensor *Water*

Pada Gambar 3.9. Ini menggambarkan alur proses penggunaan *water* sensor, untuk pengujian deteksi air menggunakan ESP32 dan PC/laptop. Berikut deskripsi singkat alur:

1. Proses dimulai
2. Melakukan *wiring* (pengkabelan) *water* sensor ke ESP32 dan PC/laptop
3. Menjalankan kode program

4. Melakukan percobaan deteksi air oleh *water* sensor
5. Memeriksa apakah sensor dapat mendeteksi air dengan baik
6. Jika ya, proses selesai. Jika tidak, kembali ke langkah percobaan deteksi air

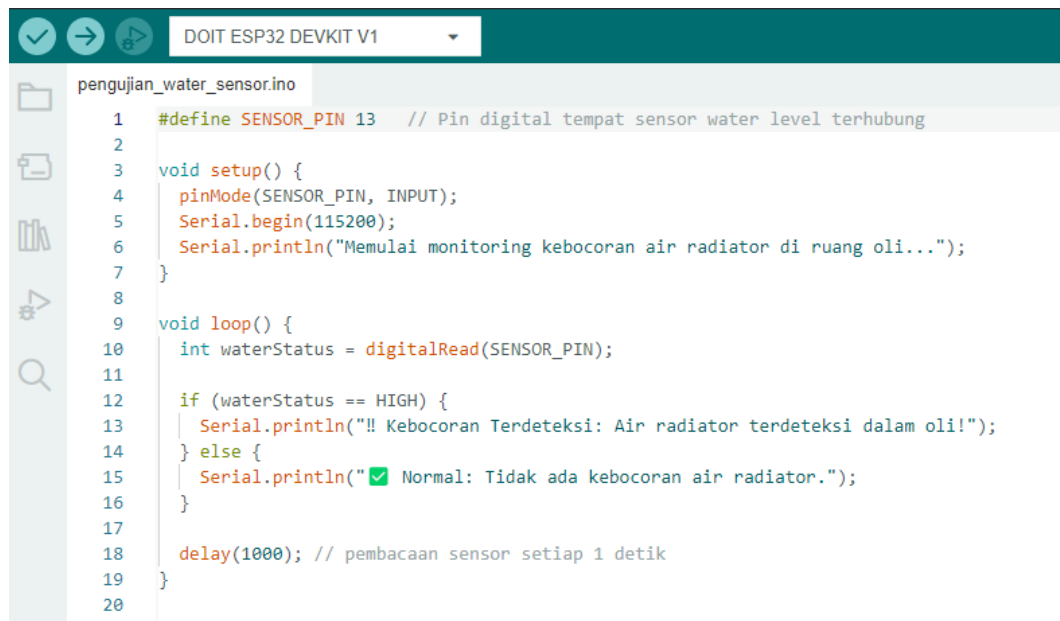
Pada *Flowchart* ini menunjukkan langkah-langkah dasar untuk menyiapkan, mengkonfigurasi, dan menguji sensor air dalam suatu sistem, dengan penekanan pada keberhasilan deteksi air sebagai kriteria penyelesaian proses.



Gambar 3. 10. Koneksi *Water Sensor* ke ESP32 dan PC/Laptop

Gambar 3.10. Ini menunjukkan skema koneksi antara laptop, papan sirkuit, dan modul sensor air (*water* sensor) berwarna merah, mengilustrasikan bagaimana data dapat mengalir dari sensor air ke komputer melalui mikrokontroler. Laptop digambarkan secara sederhana di sebelah kiri, sedangkan papan sirkuit kompleks terletak di tengah, dan modul sensor air berwarna merah terletak di sebelah kanan. Koneksi antara laptop dan papan sirkuit dilakukan melalui koneksi USB, sedangkan papan sirkuit terhubung ke modul sensor air menggunakan kabel. Skema ini menunjukkan bagaimana data dapat dikumpulkan dari sensor air dan diproses oleh

mikrokontroler sebelum dikirim ke laptop untuk analisis atau pengolahan lebih lanjut, seperti monitoring kualitas air atau deteksi kelembaban.



```

1  #define SENSOR_PIN 13 // Pin digital tempat sensor water level terhubung
2
3  void setup() {
4      pinMode(SENSOR_PIN, INPUT);
5      Serial.begin(115200);
6      Serial.println("Memulai monitoring kebocoran air radiator di ruang oli...");
7  }
8
9  void loop() {
10     int waterStatus = digitalRead(SENSOR_PIN);
11
12     if (waterStatus == HIGH) {
13         Serial.println("!! Kebocoran Terdeteksi: Air radiator terdeteksi dalam oli!");
14     } else {
15         Serial.println("✅ Normal: Tidak ada kebocoran air radiator.");
16     }
17
18     delay(1000); // pembacaan sensor setiap 1 detik
19 }
20

```

Gambar 3. 11. *Script Sensor Water*

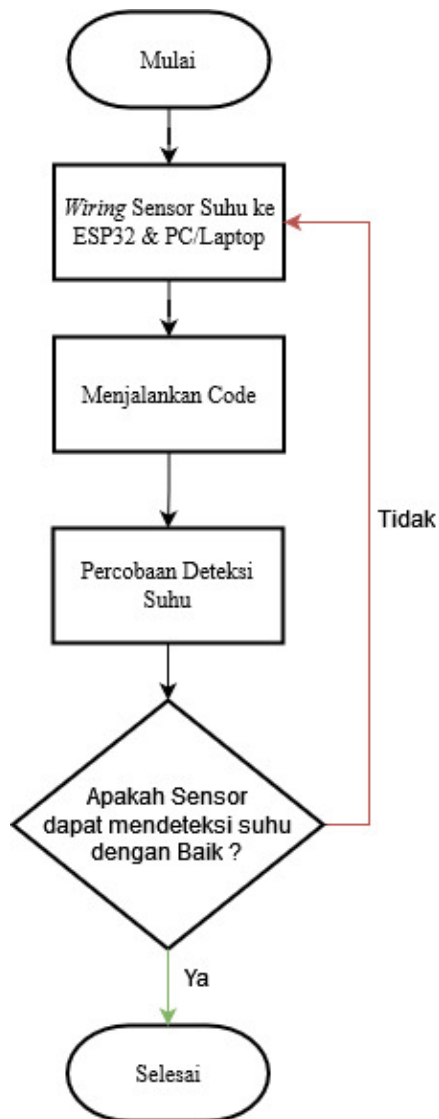
Gambar 3.11. Menunjukkan kode Arduino untuk pengujian sensor kebocoran air (*water level sensor*) menggunakan ESP32.

Kalibrasi Sensor *Water* memiliki tahapan yaitu:

- Metode Kalibrasi: Kalibrasi dilakukan dengan melakukan pencampuran air kedalam oli secara bertahap mulai yaitu pada kandungan air 0.5 ml, 1 ml, 5 ml, 10 ml, 20 ml, 30 ml. Untuk setiap tahapan pencampuran, diberikan waktu selama 1 menit untuk sensor melakukan pembacaan sehingga diketahui bagaimana sensor melakukan pembacaan, dari hasil kalibrasi ini kemudian dilakukan penyesuaian terhadap sensor sehingga keluaran dari sensor adalah bernilai 0 untuk kondisi aman, 1 untuk kondisi berpotensi dan 2 untuk kondisi terjadi kebocoran.

- Jenis Output: Output sensor adalah digital dimana sensor akan memberikan output 1 ketika sensor mendeteksi air dan output 0 ketika sensor tidak mendeteksi air. Sensor dapat mendeteksi air dengan syarat bahwa air tersebut menyentuh bagian dari sensor itu sendiri. Pada penelitian ini, keluaran dari sensor disesuaikan dengan kebutuhan sehingga memiliki keluaran 0 untuk kondisi ketika sensor tidak mendeteksi air, 1 untuk kondisi ketika keluaran sensor tidak stabil, dan 2 ketika keluaran sensor stabil mendeteksi adanya air.
- Logika Uji: Pengujian dilakukan dengan melakukan pencampuran air dengan oli secara bertahap pada kadar air 0.5 ml, 1 ml, 5 ml, 10 ml, 20 ml, 30 ml. Untuk tiap tahapan pencampuran terdapat 6 kali pembacaan data dilakukan oleh sensor.

### 3.1.3.4 Pengujian Sensor *Temperature*



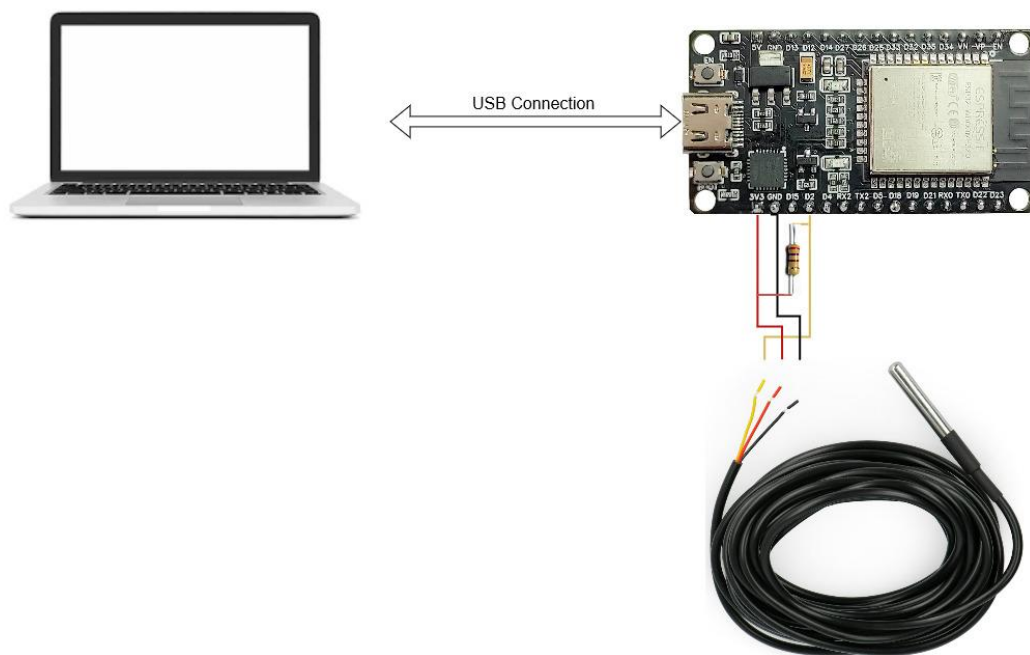
Gambar 3. 12. Pengujian Sensor *Temperature*

Pada Gambar 3.12. Ini menggambarkan alur proses penggunaan sensor air, untuk deteksi suhu menggunakan ESP32 dan PC/laptop. Berikut deskripsi singkat alurnya:

1. Proses dimulai
2. Melakukan wiring (pengkabelan) sensor suhu ke ESP32 dan PC/laptop
3. Menjalankan kode program

4. Melakukan percobaan deteksi suhu
5. Memeriksa apakah sensor dapat mendeteksi suhu dengan baik
6. Jika ya, proses selesai. Jika tidak, kembali ke langkah percobaan deteksi suhu

Pada *Flowchart* ini menunjukkan langkah-langkah dasar untuk menyiapkan, mengkonfigurasi, dan menguji sensor air dalam suatu sistem, dengan penekanan pada keberhasilan deteksi air sebagai kriteria penyelesaian proses.



Gambar 3. 13. Koneksi sensor suhu ke ESP32 dan PC/Laptop

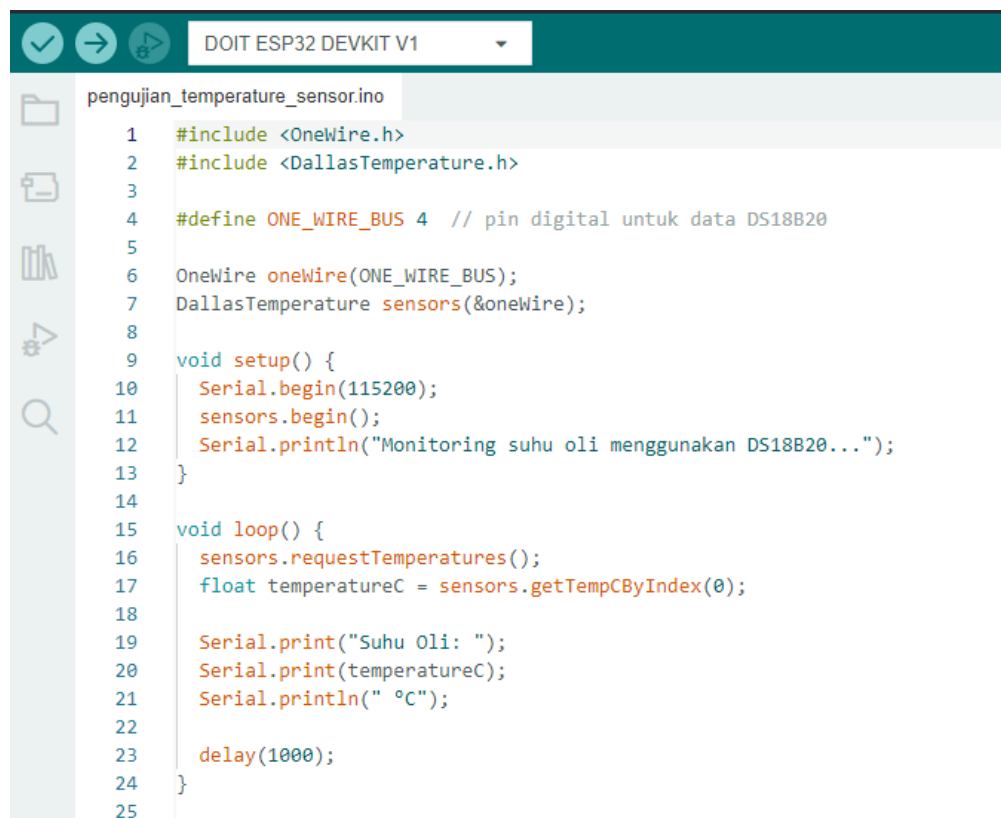
Gambar 3.13. Rangkaian ini menggambarkan koneksi sederhana untuk penelitian atau pengembangan yang melibatkan mikrokontroler dan sensor:

1. Laptop/komputer: Digunakan untuk memprogram, mengirim perintah, atau menerima data dari mikrokontroler.
2. Mikrokontroler (ESP32): Berfungsi sebagai otak dari sistem, dapat diprogram untuk mengontrol sensor atau perangkat lain.

3. Koneksi USB: Menghubungkan laptop dengan mikrokontroler untuk pemrograman dan komunikasi data.
4. Sensor suhu (DS18B20): Berfungsi untuk mengukur suhu.
5. Kabel jumper: Menghubungkan sensor ke mikrokontroler, memungkinkan pengambilan data suhu.

Pengujian ini dilakukan untuk menguji sensor dalam pemantauan suhu, pengolahan data sensor, atau pengembangan sistem kontrol berbasis mikrokontroler. Peneliti dapat memprogram mikrokontroler melalui laptop, mengambil data suhu dari sensor, dan menganalisis atau memanfaatkan data tersebut sesuai kebutuhan penelitian.

#### *Script/Program Dasar untuk Sensor Temperatur*



```

1  #include <OneWire.h>
2  #include <DallasTemperature.h>
3
4  #define ONE_WIRE_BUS 4 // pin digital untuk data DS18B20
5
6  OneWire oneWire(ONE_WIRE_BUS);
7  DallasTemperature sensors(&oneWire);
8
9  void setup() {
10     Serial.begin(115200);
11     sensors.begin();
12     Serial.println("Monitoring suhu oli menggunakan DS18B20...");
13 }
14
15 void loop() {
16     sensors.requestTemperatures();
17     float temperatureC = sensors.getTempCByIndex(0);
18
19     Serial.print("Suhu Oli: ");
20     Serial.print(temperatureC);
21     Serial.println(" °C");
22
23     delay(1000);
24 }
25

```

Gambar 3. 14. *Script Sensor Temperatur*

Gambar 3.14. Menunjukkan kode Arduino untuk membaca suhu menggunakan sensor DS18B20 dengan ESP32.

#### Kalibrasi Sensor *Temperatur*

- Metode Kalibrasi: Perbandingan terhadap termometer digital dalam media cair (oli yang dipanaskan).
- Rumus Kalibrasi Linear:

$$T_{\text{aktual}} = a \cdot T_{\text{sensor}} + b \quad (3.1.)$$

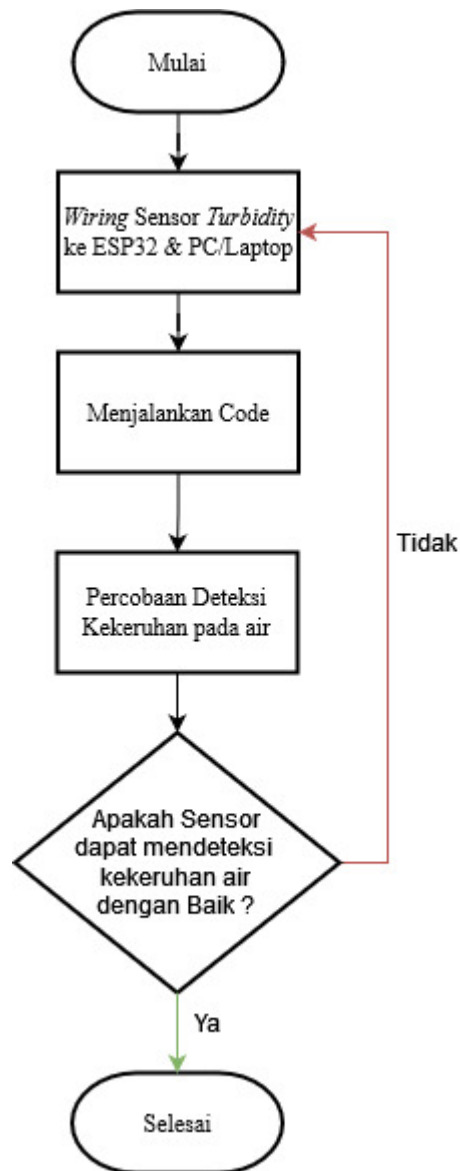
Di mana :

$T_{\text{sensor}}$  : Merupakan nilai suhu yang dibaca langsung dari sensor suhu. Nilai ini sering kali memiliki deviasi (penyimpangan) terhadap suhu sebenarnya karena toleransi sensor atau pengaruh lingkungan.

$T_{\text{actual}}$  : Merupakan nilai suhu yang sebenarnya, yang diukur menggunakan alat ukur standar seperti termometer laboratorium atau termometer digital yang sudah dikalibrasi pabrik (berfungsi sebagai acuan pembandingan).

$a, b$  : Koefisien hasil regresi linier dari data kalibrasi.

### 3.1.3.5 Pengujian Sensor *Turbidity*



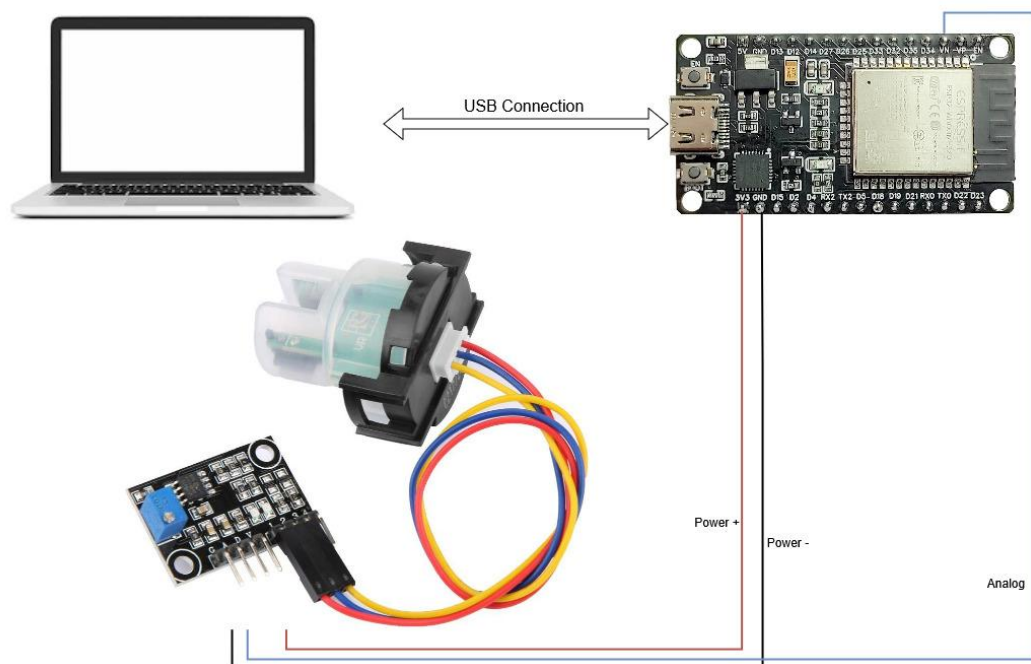
Gambar 3. 15. Pengujian Sensor *Turbidity*

Pada Gambar 3.15. ini menggambarkan proses penggunaan sensor *turbidity* (kekeruhan) air. Berikut deskripsi alurnya:

1. Mulai: Proses dimulai.
2. Wiring Sensor Turbidity ke ESP32 & PC/Laptop: Melakukan pengkabelan sensor turbidity ke perangkat ESP32 dan komputer.

3. Menjalankan Code: Mengeksekusi program yang telah disiapkan untuk sensor.
4. Percobaan Deteksi Kekeruhan pada air: Melakukan uji coba untuk mendeteksi tingkat kekeruhan air menggunakan sensor.
5. Evaluasi: Memeriksa apakah sensor dapat mendeteksi kekeruhan air dengan baik.
  - Jika Ya: Proses berlanjut ke tahap selesai.
  - Jika Tidak: Kembali ke langkah percobaan deteksi kekeruhan.
6. Selesai: Proses berakhir setelah sensor berhasil mendeteksi kekeruhan air dengan baik.

Flowchart ini menunjukkan langkah-langkah sistematis untuk mengatur, menguji, dan memverifikasi fungsi sensor turbidity dalam mengukur kekeruhan air, dengan fokus pada keakuratan deteksi sebagai syarat penyelesaian proses.



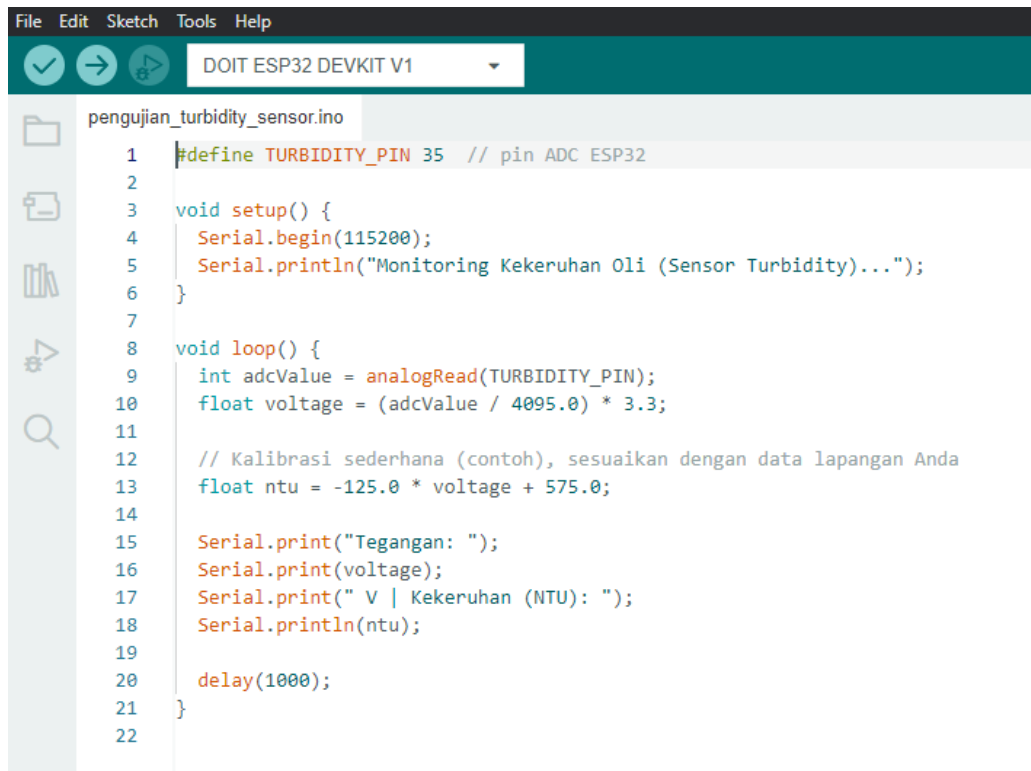
Gambar 3. 16. Koneksi sensor *turbidity* ke ESP32 dan PC/Laptop

Gambar 3.16. ini menunjukkan koneksi rangkaian pada penelitian yang melibatkan beberapa komponen elektronika:

1. Laptop: Terletak di sebelah kiri, digunakan untuk pemrograman dan pengolahan data.
2. Mikrokontroler: ESP32, berada di tengah atas. Ini adalah otak dari sistem.
3. Koneksi USB: Menghubungkan laptop dengan mikrokontroler untuk pemrograman dan transfer data.
4. Modul sensor *Turbidity*: Terletak di bagian bawah, modul ini memiliki beberapa pin koneksi.
5. Kabel penghubung: Menghubungkan mikrokontroler dengan modul sensor, pengambilan data.

Peneliti dapat menggunakan laptop untuk memprogram mikrokontroler, mengumpulkan data dari sensor kekeruhan, dan melakukan analisis lebih lanjut. Sistem ini memungkinkan pengukuran parameter kekeruhan secara *real-time*.

*Script/Program Dasar untuk Sensor Turbidity*



```

File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1

pengujian_turbidity_sensor.ino
1  #define TURBIDITY_PIN 35 // pin ADC ESP32
2
3  void setup() {
4      Serial.begin(115200);
5      Serial.println("Monitoring Kekeruhan Oli (Sensor Turbidity)...");
6  }
7
8  void loop() {
9      int adcValue = analogRead(TURBIDITY_PIN);
10     float voltage = (adcValue / 4095.0) * 3.3;
11
12     // Kalibrasi sederhana (contoh), sesuaikan dengan data lapangan Anda
13     float ntu = -125.0 * voltage + 575.0;
14
15     Serial.print("Tegangan: ");
16     Serial.print(voltage);
17     Serial.print(" V | Kekeruhan (NTU): ");
18     Serial.println(ntu);
19
20     delay(1000);
21 }
22

```

Gambar 3. 17. *Script Sensor Turbidity*

Gambar 3.17. Menunjukkan kode arduino untuk membaca kekeruhan menggunakan sensor *turbidity* dengan ESP32.

#### Kalibrasi Sensor *Turbidity*

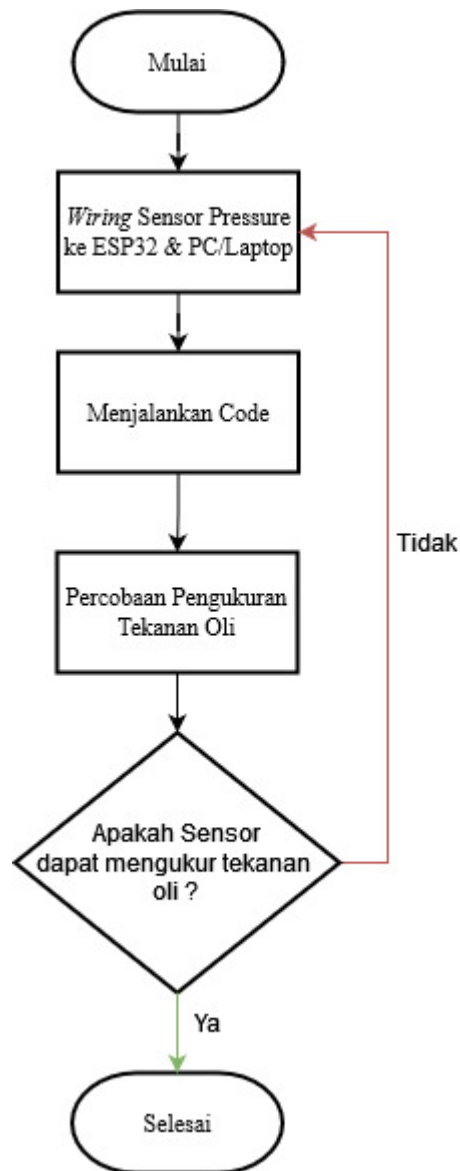
- Metode Kalibrasi: Menggunakan sampel oli dalam 3 kondisi: bersih, setengah pakai, dan bekas. Data dibandingkan dengan pengamatan visual dan nilai ADC sensor.
- Rumus Kalibrasi:

$$NTU = a \cdot V_{ADC} + b \quad (3.2.)$$

Di mana:

- $NTU$ : Nephelometric Turbidity Unit (tingkat kekeruhan)
- $V_{ADC}$ : Tegangan analog dari sensor
- $a, b$ : Hasil kalibrasi regresi linier

### 3.1.3.6 Pengujian Sensor *Pressure*



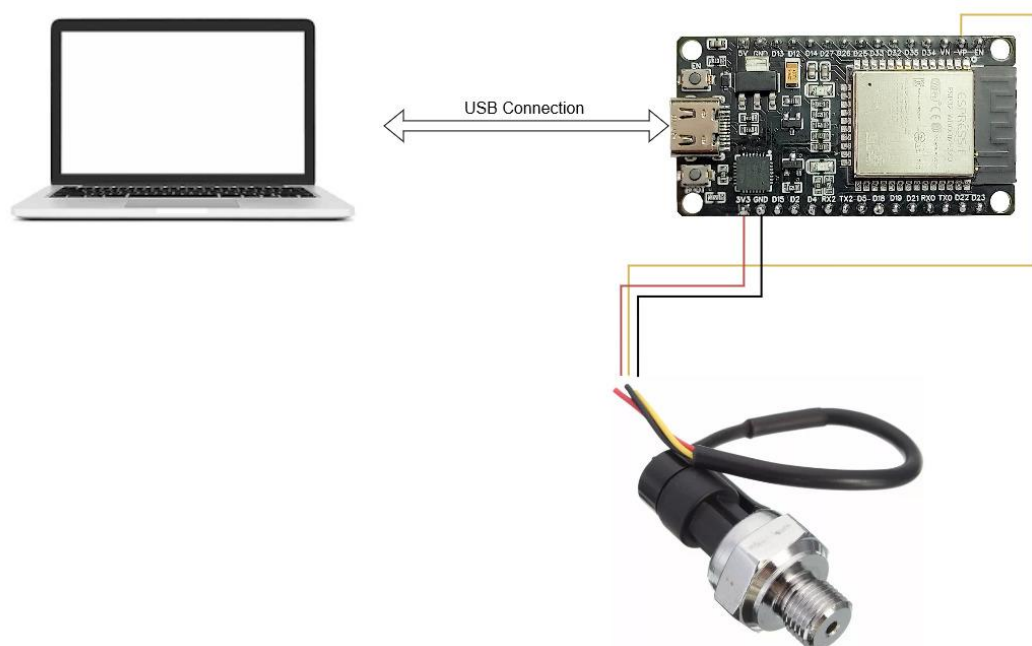
Gambar 3. 18. Pengujian Sensor *Pressure*

Gambar 3.18. Menggambarkan proses penggunaan sensor tekanan untuk mengukur tekanan oli. Berikut deskripsi alurnya:

1. Mulai: Proses dimulai.
2. Wiring Sensor Pressure ke ESP32 & PC/Laptop: Melakukan pengkabelan sensor tekanan ke perangkat ESP32 dan komputer.

3. Menjalankan Code: Mengeksekusi program yang telah disiapkan untuk sensor tekanan.
4. Percobaan Pengukuran Tekanan Oli: Melakukan uji coba untuk mengukur tekanan oli menggunakan sensor.
5. Evaluasi: Memeriksa apakah sensor dapat mengukur tekanan oli dengan baik.
  - Jika Ya: Proses berlanjut ke tahap selesai.
  - Jika Tidak: Kembali ke langkah percobaan pengukuran tekanan oli.
6. Selesai: Proses berakhir setelah sensor berhasil mengukur tekanan oli dengan baik.

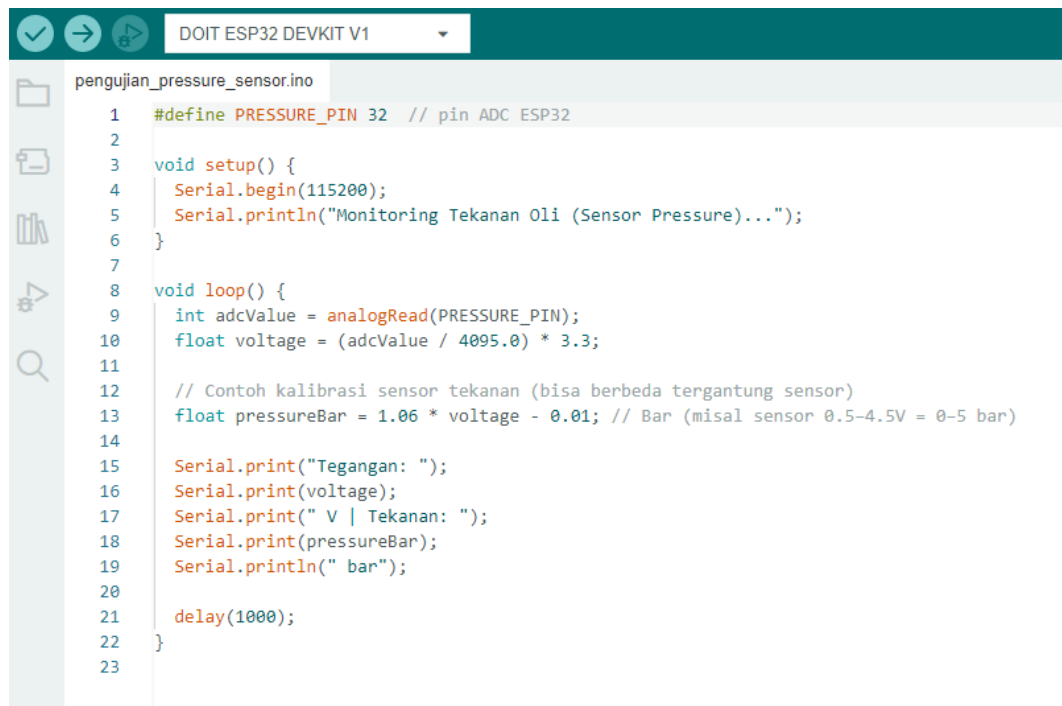
Flowchart ini menunjukkan langkah-langkah sistematis dalam menyiapkan, mengkonfigurasi, dan menguji sensor tekanan untuk mengukur tekanan oli. Fokus utamanya adalah memastikan sensor dapat melakukan pengukuran dengan akurat sebelum menyelesaikan proses pengujian komponen.



Gambar 3. 19. Koneksi sensor pressure ke ESP32 dan PC/Laptop

Gambar 3.19. ini menunjukkan skema koneksi antara laptop, papan ESP32, dan sensor tekanan (pressure sensor), mengilustrasikan bagaimana data tekanan dapat dikumpulkan dan diproses oleh ESP32 sebelum dikirim ke laptop. Papan ESP32 digambarkan secara sederhana di tengah, dengan sensor tekanan terletak di sebelah kanan dan laptop di sebelah kiri. Koneksi antara laptop dan ESP32 dilakukan melalui koneksi USB, sedangkan ESP32 terhubung ke sensor tekanan menggunakan tiga kabel berwarna (merah, kuning, dan hitam). Skema ini menunjukkan bagaimana ESP32 dapat membaca data tekanan dari sensor, memprosesnya, dan mengirimkannya ke laptop untuk analisis atau pengolahan lebih lanjut.

#### *Script/Program Dasar untuk Sensor Pressure*



```

1  #define PRESSURE_PIN 32 // pin ADC ESP32
2
3  void setup() {
4      Serial.begin(115200);
5      Serial.println("Monitoring Tekanan Oli (Sensor Pressure)...");
6  }
7
8  void loop() {
9      int adcValue = analogRead(PRESSURE_PIN);
10     float voltage = (adcValue / 4095.0) * 3.3;
11
12     // Contoh kalibrasi sensor tekanan (bisa berbeda tergantung sensor)
13     float pressureBar = 1.06 * voltage - 0.01; // Bar (misal sensor 0.5-4.5V = 0-5 bar)
14
15     Serial.print("Tegangan: ");
16     Serial.print(voltage);
17     Serial.print(" V | Tekanan: ");
18     Serial.print(pressureBar);
19     Serial.println(" bar");
20
21     delay(1000);
22 }
23

```

Gambar 3. 20. *Script Sensor Pressure*

Gambar 3.20. Menunjukkan kode arduino untuk membaca tekanan menggunakan sensor *pressure* dengan ESP32.

### Kalibrasi Sensor *Pressure*

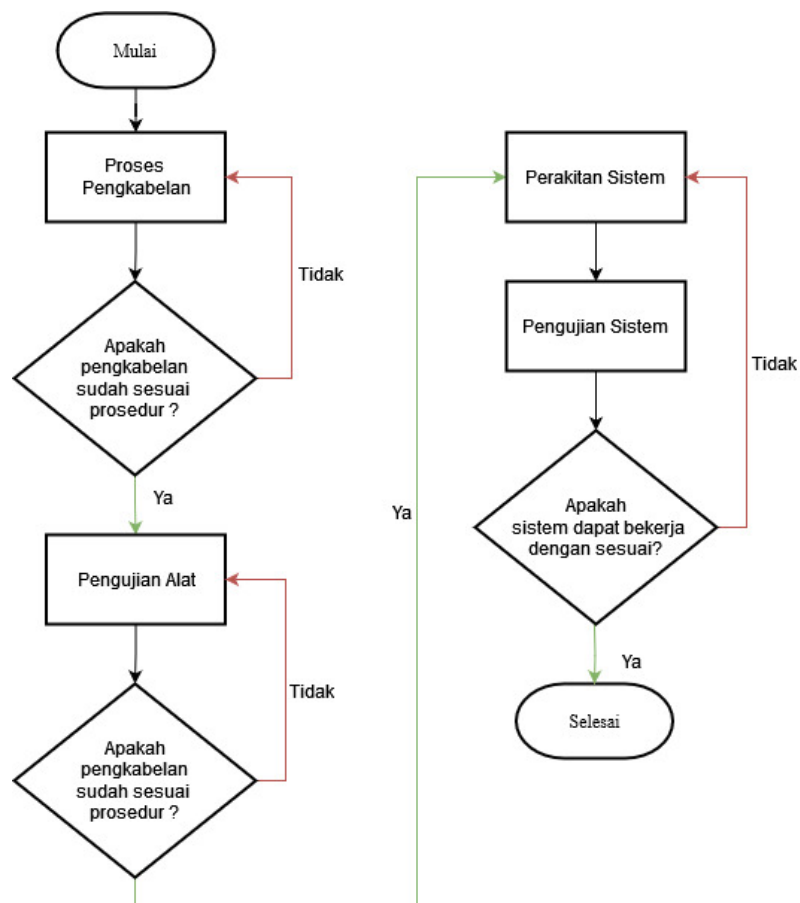
- Metode Kalibrasi: Uji terhadap alat ukur tekanan manual (pressure gauge), dengan memberikan tekanan menggunakan pompa udara.
- Rumus Kalibrasi:

$$P_{\text{aktual}} = a \cdot V_{\text{sensor}} + b \quad (3.3.)$$

Di mana:

- $V_{\text{sensor}}$ : Tegangan output dari sensor
- $P_{\text{aktual}}$ : Tekanan aktual (bar atau psi)
- $a, b$ : Koefisien kalibrasi

### 3.1.4. Perancangan & Pengujian Sistem



Gambar 3. 21. *Flowchart* Perancangan Sistem

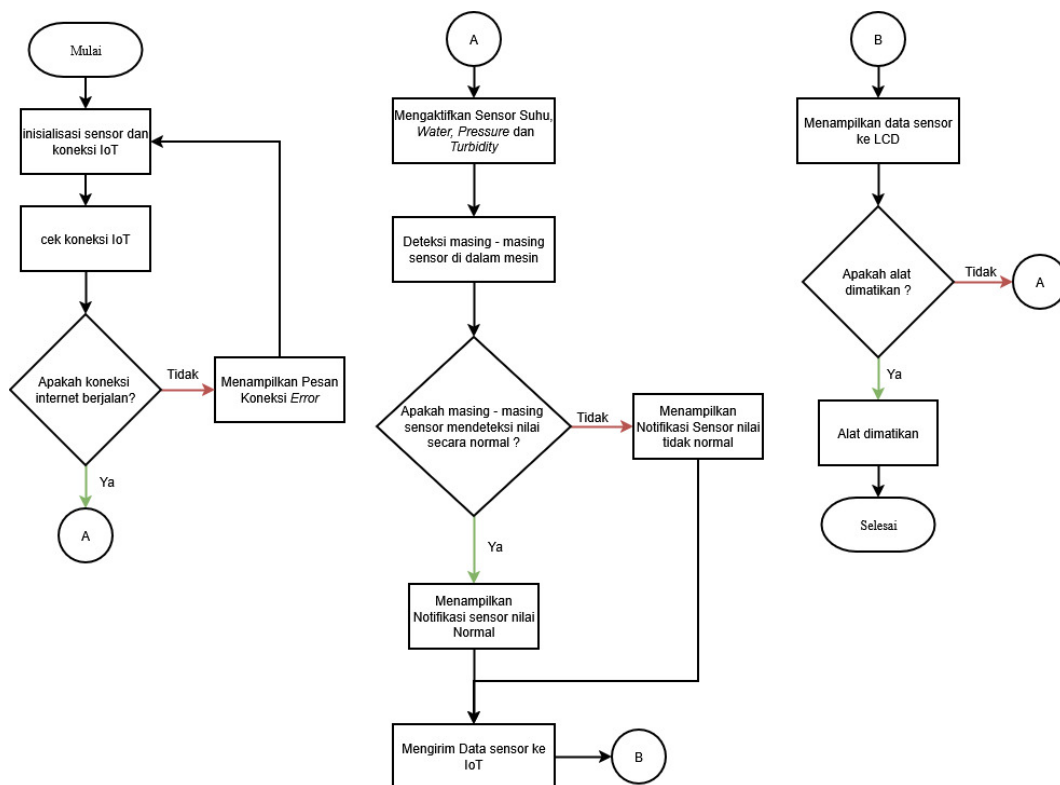
Gambar 3.21. ini menggambarkan alur perancangan dan pengujian suatu sistem. Berikut deskripsi alurnya:

1. Mulai: Proses dimulai.
2. Proses Pengkabelan: Melakukan pengkabelan komponen-komponen sistem.
3. Evaluasi Pengkabelan: Memeriksa apakah pengkabelan sudah sesuai prosedur.
  - Jika Tidak: Kembali ke proses pengkabelan.
  - Jika Ya: Lanjut ke pengujian alat.
4. Pengujian Alat: Melakukan tes pada peralatan individual.
5. Evaluasi Pengujian Alat: Memeriksa apakah hasil pengujian alat sudah sesuai prosedur.
  - Jika Tidak: Kembali ke proses pengkabelan.
  - Jika Ya: Lanjut ke perakitan sistem.
6. Perakitan Sistem: Menggabungkan semua komponen menjadi satu sistem utuh.
7. Pengujian Sistem: Melakukan tes pada keseluruhan sistem yang telah dirakit.
8. Evaluasi Kinerja Sistem: Memeriksa apakah sistem dapat bekerja dengan sesuai.
  - Jika Tidak: Kembali ke perakitan sistem.
  - Jika Ya: Proses selesai.
9. Selesai: Perancangan sistem berhasil dan selesai.

Pada Alur ini menunjukkan pendekatan sistematis dalam membangun dan menguji suatu sistem, dengan penekanan pada verifikasi di setiap tahap untuk memastikan kualitas dan fungsionalitas yang diharapkan pada penelitian ini.

### 3.1.4.1 Pengujian Sistem

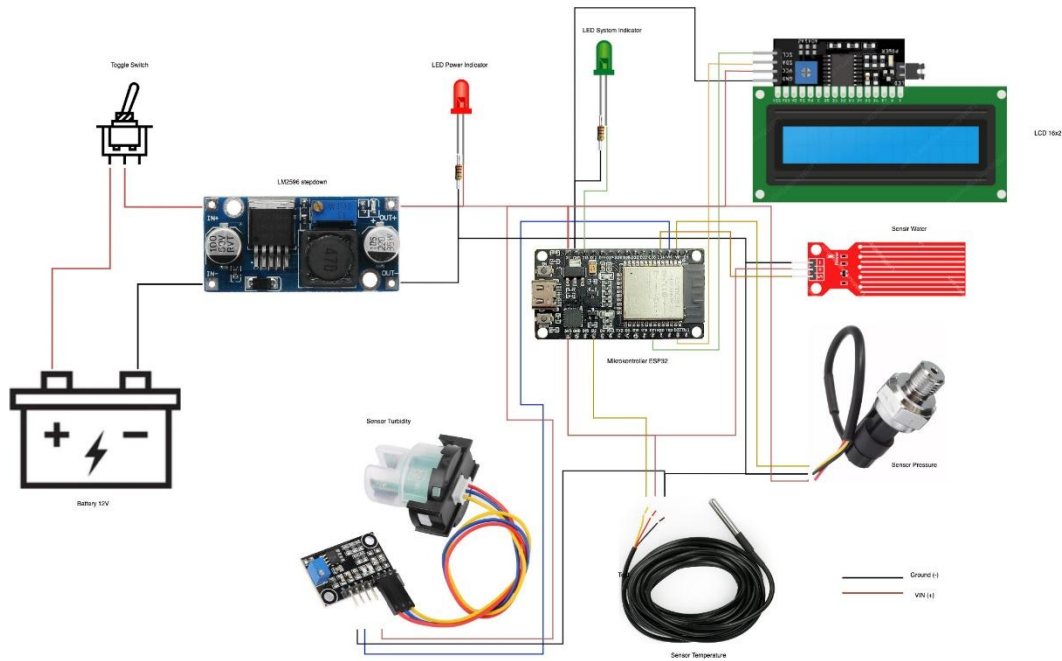
Pada pengujian sistem ini dilakukan untuk menguji dari beberapa bagian seperti uji deteksi kebocoran, pemantauan secara lokal dan *online*. Pengujian ini dilakukan untuk mengetahui kinerja sensor dan pemantauan data yang dihasilkan oleh sensor.



Gambar 3. 22. *Flowchart* Pengujian Sistem

Gambar 3.22. Menunjukkan alur pengujian sistem pada penelitian ini. Pada bagian pengujian ini dilakukan untuk menguji kesesuaian kerja sistem atau alat yang dirancang. Berdasarkan gambar 3.16. pada alur pengujian sistem ini meliputi

- Mulai
- Menginisialisasi sensor dan koneksi ke IoT
- Cek koneksi IoT
- Apakah koneksi internet dapat berjalan, jika tidak maka akan menampilkan pesan koneksi Error, dan mengulangi ke proses inisialisasi sensor dan koneksi IoT, jika koneksi internet dapat berjalan maka lanjut ke proses berikut
- Mengaktifkan sensor suhu, *water*, *pressure* dan *turbidity*
- Deteksi masing – masing sensor di dalam mesin
- Apakah sensor dapat mendeteksi nilai secara normal, jika tidak maka artinya nilai melebihi batas maksimum atau nilai tidak normal maka akan menampilkan notifikasi sensor nilai tidak normal atau dengan kondisi pesan per masing-masing sensor dan mengirim data sensor ke IoT, jika nilai sensor normal maka akan menampilkan notifikasi sensor nilai normal dan mengirim data sensor ke IoT. Pada dua kondisi tersebut tetap akan mengirim data ke IoT dan lanjut ke proses selanjutnya
- Menampilkan data sensor ke LCD
- Apakah alat dimatikan, jika tidak maka akan masuk ke proses pengaktifan sensor yang di tunjukan pada simbol huruf A dan melakukan proses itu berulang, jika ya atau alat dimatikan maka proses pengujian sistem selesai.



Gambar 3. 23. *Wiring* Pengujian Sistem

Gambar 3.23. Merupakan diagram rangkaian sistem monitoring sensor menggunakan mikrokontroler ESP32.

#### 3.1.4.2 Pengujian Sistem Deteksi Kebocoran Air Radiator

Pengujian sistem deteksi kebocoran air radiator bertujuan untuk mengetahui apakah keberadaan air radiator di dalam ruang oli mesin dapat dideteksi secara akurat oleh sensor *water level*. Pengujian dilakukan dalam dua kondisi, yaitu saat mesin dalam keadaan mati dan saat mesin menyala, guna mensimulasikan potensi kebocoran pada berbagai situasi operasional kendaraan.

Metode pengujian dilakukan dengan mendekatkan sensor *water* ke media uji berupa oli murni, kemudian mencampurkan air radiator secara bertahap untuk mengamati perubahan nilai pembacaan sensor. Nilai yang dihasilkan sensor akan dibandingkan antara kondisi tanpa air radiator dan kondisi dengan keberadaan air radiator.

Jika sensor memberikan perubahan nilai yang signifikan saat terdeteksi adanya air, maka sistem dianggap mampu melakukan deteksi kebocoran. Hasil pembacaan sensor ini dijadikan acuan dalam sistem monitoring, baik untuk pemicu alarm maupun tampilan data pada LCD dan platform IoT.

Dengan demikian, keberadaan air radiator yang masuk ke dalam ruang oli mesin dan menyentuh sensor akan menyebabkan perubahan nilai digital atau analog pada pembacaan, dan dapat ditandai sebagai indikasi kebocoran. Keakuratan sensor dalam mendeteksi air di antara fluida oli merupakan parameter penting dalam validasi fungsi sistem.

#### **3.1.4.3 Pengujian Sistem *Monitoring***

Pada pengujian sistem *monitoring* ini dilakukan untuk menguji data yang dihasilkan oleh sensor dan pemantauan data tersebut diukur dari segi suhu, kekeruhan, tekanan, dan juga deteksi kebocoran oleh sensor *water*. Data yang harus ditampilkan pada monitoring ini ada 4 data yaitu: Suhu, Kekeruhan, Tekanan dan Juga *Water Coolant Detect* (deteksi air radiator didalam ruang mesin). Pemantauan data ini dilakukan dengan kondisi saat mesin menyala dan mati, untuk mengetahui apakah tiap sensor dapat menghasilkan nilai sesuai dengan kondisinya seperti perbedaan suhu saat mesin menyala ataupun mati.

#### **3.1.4.4 Pengujian Sistem IoT**

Pengujian sistem iot yaitu untuk menguji pengiriman data dan menampilkan data di server menggunakan aplikasi *Blynk* apakah data yang dihasilkan sensor dapat dikirim oleh ESP32 ke server, data yang dikirimkan berupa data suhu, kekeruhan, tekanan dan deteksi kebocoran.

### **3.2. Subjek Dan Objek Penelitian**

Subjek pada penelitian ini yaitu terkait dengan sensor yang digunakan seperti sensor *water*, sensor *pressure*, sensor turbidity, dan sensor suhu untuk digunakan sebagai deteksi dan juga pembacaan data untuk monitoring. Pada objek yaitu adalah mesin kendaraan roda dua (motor), oli dan air radiator sebagai objek yang akan di teliti berdasarkan subjek yang digunakan dari sensor – sensor yang menampilkan datanya. Fokus penelitian ini adalah pemantauan kelayakan oli dan juga kebocoran air radiator pada ruang oli mesin kendaraan roda dua sebagai perbandingan antara subjek dan objek yang diteliti.

### **3.3. Implementasi Rancangan Alat**

Implementasi rancangan alat adalah tahap penting dalam proses pengembangan dan pembuatan alat yang dapat mempengaruhi kinerja dan kualitas alat. Pada implementasi ini merupakan realisasi dari hasil kajian teoritis dan perencanaan sistem, hal ini bertujuan untuk mengetahui keterhubungan antara hasil kerja alat yang diteliti dari teori yang berlaku.

### **3.4. Analisa Data Sistem**

Pada Analisa data sistem ini dilakukan untuk menyesuaikan hasil penelitian dari kajian dan teori yang berlaku. Sehingga mendapatkan hasil implementasi yang sesuai dengan rancangan secara ilmiah. Selain itu juga pada Analisa data ini dapat diketahui kelebihan dan kekurangan kerja alat. Pada hasil implementasi alat ini ada beberapa Data yang akan dianalisa terdiri dari pengujian Sistem Deteksi Kebocoran, Sistem Monitoring, dan Sistem IoT.

### 3.5. Jadwal Kegiatan Penelitian

Tabel 3. 2. Jadwal Kegiatan Penelitian

[illegible]