BAB V
HASIL DAN PEMBAHASAN

Bab ini akan memaparkan detail hasil penelitian yang telah dilakukan oleh peneliti sebelumnya yang terdiri melalui dua sub bab, yaitu hasil dan pembahasan. Penjelasan terkait detail bab ini dijabarkan sebagai berikut.

5.1 Hasil

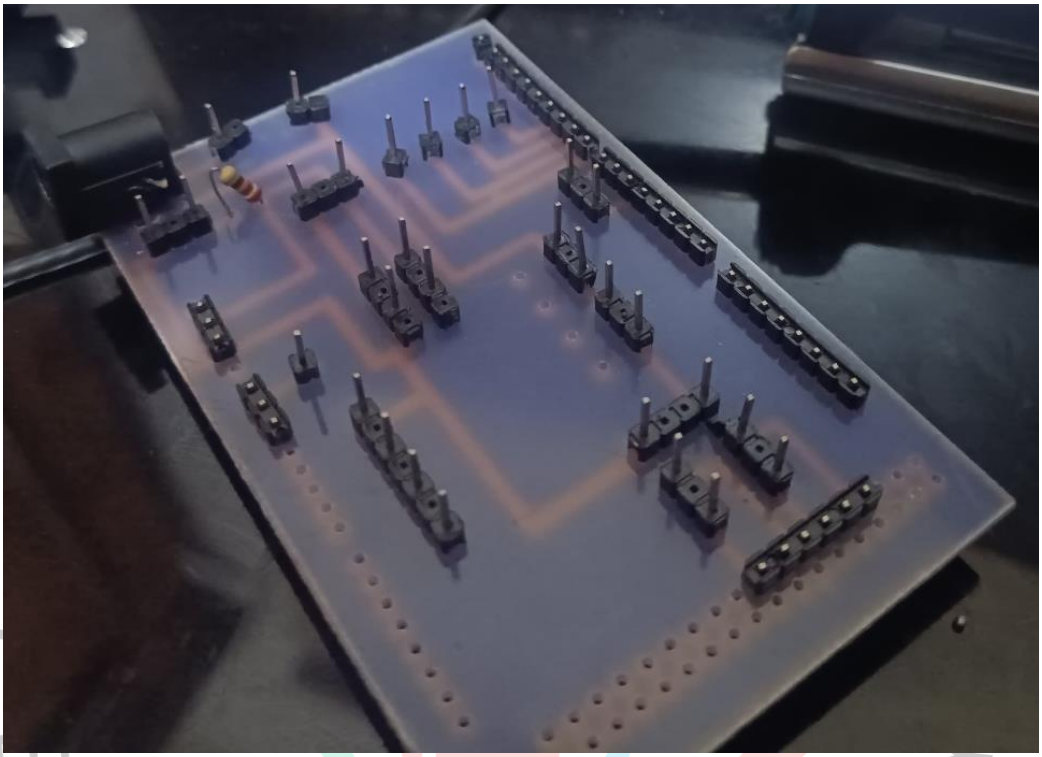
Subbab ini akan menjelaskan hasil penerapan rangkaian sistem sesuai dengan persyaratan dan perancangan yang telah disusun oleh peneliti. Penjabaran hasil implementasi penelitian ini terbagi menjadi dua bagian, yaitu pengembangan rancang fisik dan pengembangan kode program. Kedua aspek ini bertujuan untuk memastikan bahwa rancangan fisik mampu berjalan sesuai dengan prinsip kerja yang telah dijelaskan pada subbab 4.3.1, yang menguraikan prinsip kerja sistem. Berikut adalah hasil dari penelitian ini.

5.1.1 Perakitan

Perakitan adalah tahap implementasi komponen-komponen yang diperlukan untuk mengembangkan Sistem Cerdas Pembersih Toren Berbasis IoT. Pada tahap ini, perakitan dilakukan berdasarkan spesifikasi kebutuhan perangkat keras yang telah dijelaskan sebelumnya di subbab 4.1. Proses perakitan ini mencakup lima bagian utama: desain fisik rangkaian *printed circuit board* (PCB), perakitan rangkaian driver L298N, perakitan rangkaian kran otomatis, penyusunan keseluruhan rangkaian, tampilan akhir rangkaian, dan pembuatan halaman *dashboard website* untuk pengguna. Desain fisik PCB melibatkan pembuatan dan penyusunan jalur untuk menghubungkan berbagai komponen elektronik. Perakitan rangkaian driver L298N meliputi pemasangan driver motor untuk mengendalikan motor pembersih. Rangkaian kran otomatis dirakit untuk mengendalikan aliran air secara otomatis. Penyusunan keseluruhan rangkaian memastikan semua komponen terhubung dengan benar, dan tampilan akhir rangkaian memberikan gambaran visual dari sistem yang telah selesai. Terakhir, halaman *dashboard website* dibuat untuk memberikan antarmuka pengguna yang memungkinkan pemantauan dan

pengendalian sistem pembersih toren secara *real-time*. Penjelasan lebih detail akan dijelaskan di subbab subbab berikut.

(1) Perakitan PCB

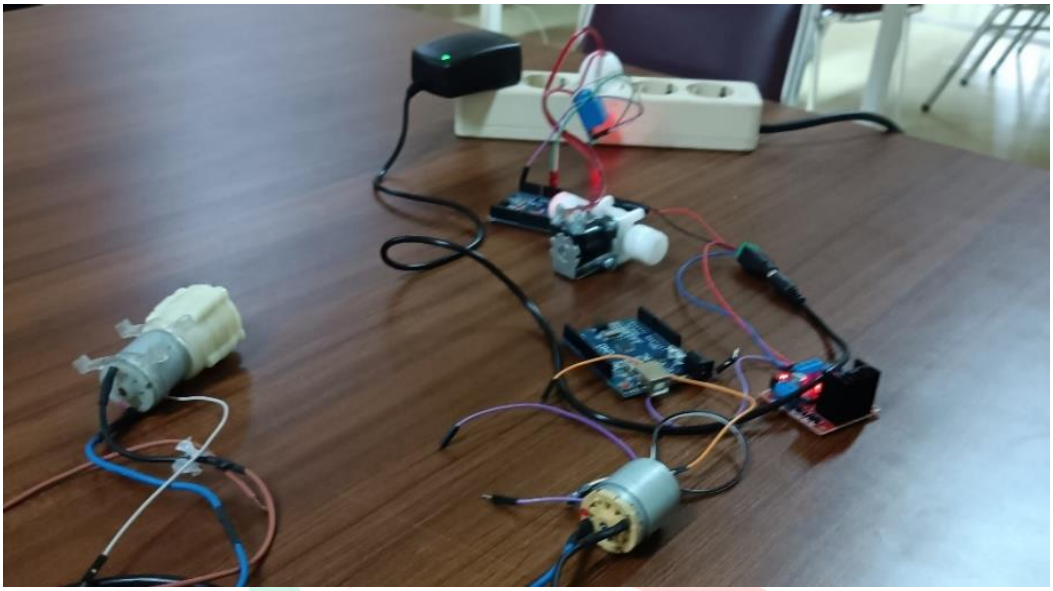


Gambar 5. 1 Perakitan PCB

Gambar 5.1 juga menunjukkan bagaimana bentuk fisik PCB berdasarkan perancangan yang telah dijelaskan sebelumnya pada gambar 4.7. PCB yang ditampilkan pada Gambar 5.1 memiliki tata letak komponen yang diatur dengan cermat untuk mengoptimalkan ruang dan mengurangi interferensi elektromagnetik. Jalur-jalur konduktor yang menghubungkan komponen-komponen pada PCB didesain dengan lebar dan jarak tertentu, sesuai dengan standar desain PCB, untuk memastikan bahwa sinyal dapat mengalir dengan baik tanpa mengalami gangguan atau kehilangan yang signifikan. Selain itu, pada PCB ini juga terdapat Jack DC yang berfungsi untuk mengalirkan tegangan 12V. Tegangan ini digunakan untuk mengaktifkan driver L298N. Driver L298N ini, pada gilirannya, berfungsi untuk mengirimkan tegangan 5V ke Wemos Mega. Desain ini memastikan bahwa setiap komponen mendapatkan tegangan yang sesuai untuk

beroperasi dengan benar, dan pengaturan ini juga meminimalkan risiko kerusakan akibat ketidaksesuaian tegangan.

(2) Perakitan *Driver* L298N

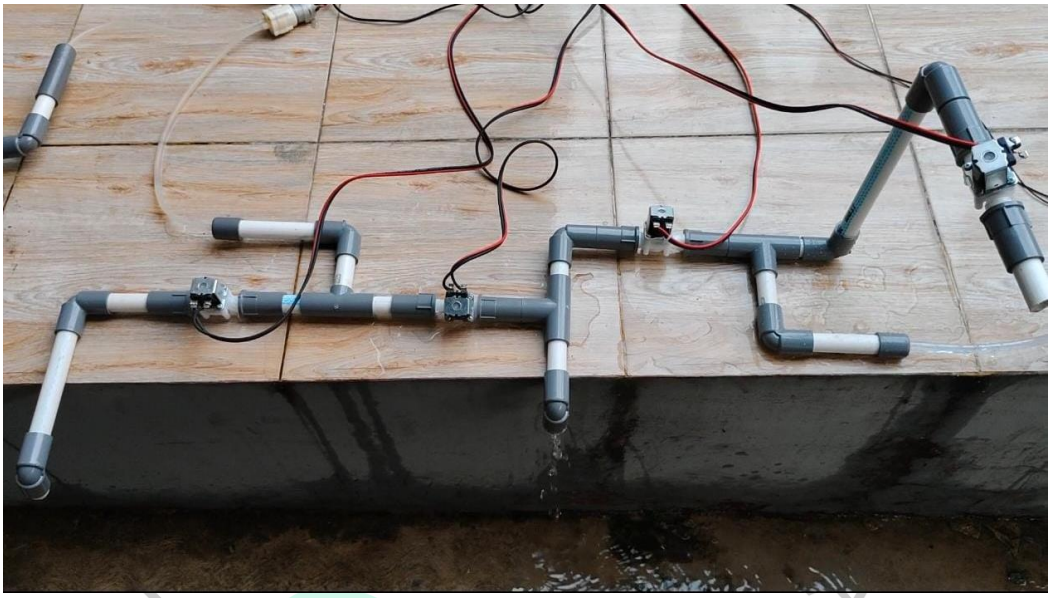


Gambar 5. 2 Perakitan Driver L298N

Gambar 5.2 merupakan hasil dari rangkaian driver L298N untuk menjalankan motor DC dan mesin pompa air, driver L298N membutuhkan tegangan 12V, GND, dan 5V dari Arduino agar dapat agar dapat bekerja dengan baik. Rangkaian driver L298N juga memerlukan sinyal kontrol dari Arduino untuk mengatur arah dan kecepatan motor. Driver ini biasanya memiliki dua saluran yang memungkinkan untuk mengendalikan dua motor DC secara individu.

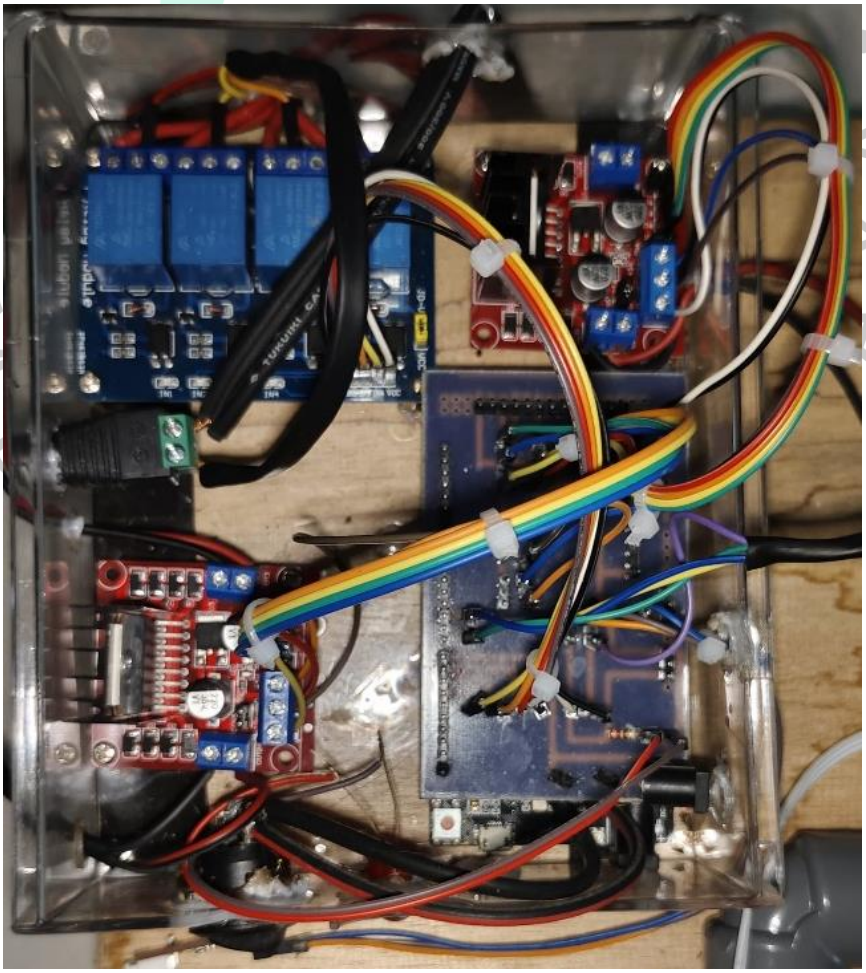
(3) Perakitan Rangkaian Kran Otomatis

Gambar 5.3 merupakan hasil dari rangkaian kran otomatis yang terhubung dengan relay 4 channel. Dalam proses pengoperasian kran otomatis yang mengontrol aliran air. Relay 4 channel digunakan untuk saklar elektronik yang dapat dikendalikan oleh sinyal dari mikrokontroler Arduino. Rangkaian ini bekerja dengan memanfaatkan relay untuk menghubungkan atau memutuskan aliran tegangan 12V ke kran otomatis berdasarkan sinyal yang diterima dari sensor atau kontrol manual dari Wemos Mega 2560.



Gambar 5. 3 Perakitan Rangkaian Kran Otomatis

(4) Tampilan Keseluruhan Rangkaian

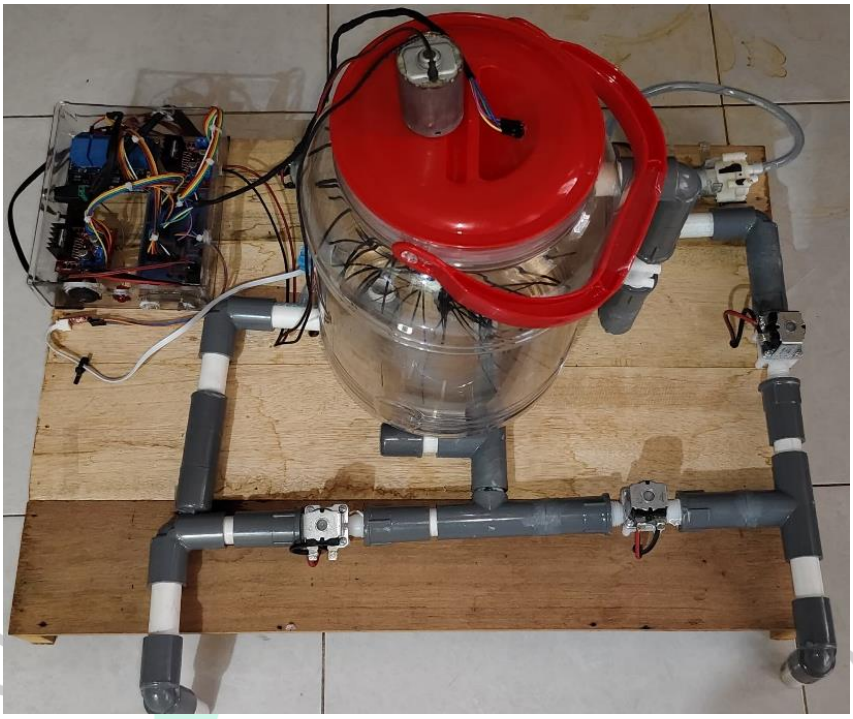


Gambar 5. 4 Tampilan Keseluruhan Rangkaian

Pada Gambar 5.4, ditampilkan keseluruhan rangkaian untuk Pengembangan Sistem Cerdas Pembersih Toren Berbasis IoT, yang telah terhubung dengan Rangkaian PCB yang sebelumnya dijelaskan. Sistem ini mampu mendeteksi tingkat kekeruhan air menggunakan sensor *turbidity* dan mengukur level air dalam toren dengan sensor *ultrasonic*. Simulasi dilakukan menggunakan driver L298N untuk mengendalikan motor DC yang berfungsi sebagai alat pembersih toren. Selain itu, relay 4 channel diaktifkan sesuai kondisi untuk mengalirkan tegangan 12V ke kran otomatis, sehingga memungkinkan aliran air terbuka. Jika sensor *ultrasonic* mendeteksi level air berada di bawah batas minimum, pompa air akan otomatis mengisi toren. Semua data yang diperoleh dari sensor akan ditampilkan di *website* monitoring, memudahkan pengguna dalam memantau kondisi dan kekeruhan air. Pada penelitian ini, digunakan toples berkapasitas 7 liter untuk simulasi pembersihan dan pengisian air guna menciptakan kondisi yang serupa dengan toren sebenarnya.

(5) Tampilan Akhir Sistem

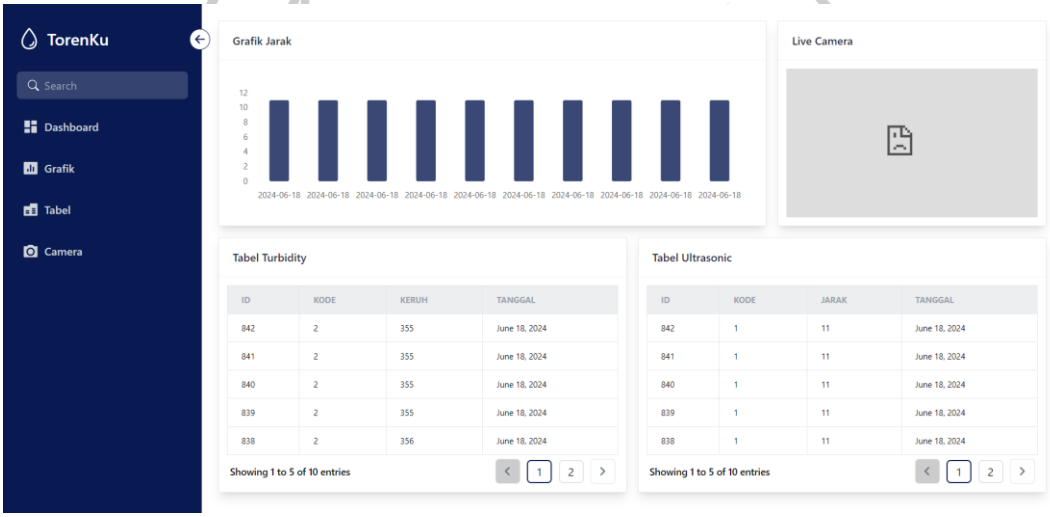
Gambar 5.5 menunjukkan hasil perakitan keseluruhan komponen yang sudah disatukan dengan toren. Komponen tambahan yang digunakan dalam rangkaian ini meliputi 4 kran otomatis (solenoid), motor DC, ESP32-CAM, dan 2 mesin pompa air. Prototipe alat dan toren saling terhubung dengan fungsionalitas masing-masing: motor DC dan pompa air dihubungkan ke driver L298N agar kecepatan dan arah perputaran motor dapat diatur, kran otomatis terhubung ke relay 4 channel untuk menerima perintah ON atau OFF, dan ESP32-CAM memberikan gambaran *real-time* kondisi dalam toren. Motor DC berfungsi menggerakkan mekanisme pembersihan di dalam toren, pompa air mengalirkan air masuk dan keluar toren sesuai kebutuhan, dan kran otomatis mengontrol aliran air secara otomatis. Sistem ini dirancang untuk beroperasi dengan koordinasi yang baik, sehingga proses pembersihan toren menjadi lebih mudah dan efektif.



Gambar 5. 5 Tampilan Akhir Sistem

(6) Halaman *Dashboard Website*

Dashboard Website berfungsi untuk menyajikan seluruh data yang dikumpulkan oleh sensor kepada pengguna. Dengan adanya *dashboard website* ini, pengguna dapat dengan lebih mudah memantau kondisi level air dan tingkat kekeruhan air dalam toren. Selain itu, pengguna juga dapat secara langsung melihat kondisi toren melalui kamera yang menyediakan tampilan secara *real-time*. Hal ini memungkinkan pengguna dalam mendapatkan informasi berupa data yang lebih komprehensif dan sesuai dengan kondisi toren.



Gambar 5. 6 Halaman Dashboard Website

Gambar 5.6 merupakan tampilan dari beberapa informasi yang diberikan kepada pengguna yang terdiri dari jarak, kekeruhan, dan kamera. Tampilan tersebut memberikan informasi untuk mengetahui nilai dari masing – masing variable sehingga pengguna dapat mengetahui data saat ini.

5.1.1 Kode Program

Kode Kode program merupakan elemen penting dalam pengembangan sebuah sistem, karena kode tersebut berfungsi untuk mengontrol perangkat keras (hardware) dan sensor-sensor yang digunakan. Oleh karena itu, diperlukan kode program yang dirancang khusus untuk memenuhi kebutuhan sistem. Berikut ini adalah kode program yang digunakan dalam proyek ini, yang mencakup kontrol perangkat keras dan sensor.

Tabel 5. 1 Kode Program

Potongan Kode Program Ke-1	

Gambar	`// Ultrasonic`
	`#define echoPin 7`
	`#define trigPin 5`
	`long timer;`
	`int ultrasonicValue;`

Keterangan	Definisi variabel pin untuk sensor *ultrasonic*. Dimana trigPin untuk membangkitkan sinyal *ultrasonic* dan echoPin untuk mendeteksi sinyal pantulan *ultrasonic*.
Potongan Kode Program Ke-2	

Gambar	`// turbidityValue`
	`int turbidity = A0;`
	`int turbidityValue;`
	
	`// Relay`
	`#define PIN_RELAY_1  8`
	`#define PIN_RELAY_2  9`
	`#define PIN_RELAY_3  10`
	`#define PIN_RELAY_4  11`

Keterangan	Definisi variabel pin untuk sensor *turbidity* dan relay 4 channel.
Potongan Kode Program Ke-3	

Gambar

```
// L289N (Motor + Pompa)
#define dira 26
#define dirb 32
#define pwma 30
#define pwmb 36

#define dira2 40
#define dirb2 44
#define pwma2 42
#define pwmb2 46
```

Keterangan

Definisi variabel pin untuk dua komponen Driver L298N. Terdapat dira dan dirb untuk membedakan dua kondisi motor dc seperti perputaran dan kecepatan masing-masing motor.

Potongan Kode Program Ke-4

Gambar

```
void motor(int a, int b){
  if (a >= 0){
    digitalWrite(dira, 0);
    analogWrite(pwma, a);
  } else if (a < 0){
    digitalWrite(dira, 1);
    analogWrite(pwma, a+255);
  }
  if (b >= 0){
    digitalWrite(dirb, 0);
    analogWrite(pwmb, b);
  } else if (a < 0){
    digitalWrite(dirb, 1);
    analogWrite(pwmb, b+255);
  }
}
```

```
void motor2(int a, int b){
  if (a >= 0){
    digitalWrite(dira2, 0);
    analogWrite(pwma2, a);
  } else if (a < 0){
    digitalWrite(dira2, 1);
    analogWrite(pwma2, a+255);
  }
  if (b >= 0){
    digitalWrite(dirb2, 0);
    analogWrite(pwmb2, b);
  } else if (a < 0){
    digitalWrite(dirb2, 1);
    analogWrite(pwmb2, b+255);
  }
}
```

Keterangan

Fungsi untuk mengontrol dua motor menggunakan L298N. Fungsi ini menerima 2 parameter yaitu a dan b, yang mengatur masing-masing kecepatan dan arah motor.

Potongan Kode Program Ke-5

Gambar

```
void setup() {
  Serial.begin(9600);
  Serial3.begin(115200);

  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);

  pinMode(dira, OUTPUT);
  pinMode(dirb, OUTPUT);

  pinMode(dira2, OUTPUT);
  pinMode(dirb2, OUTPUT);

  pinMode(PIN_RELAY_1, OUTPUT);
  pinMode(PIN_RELAY_2, OUTPUT);
  pinMode(PIN_RELAY_3, OUTPUT);
  pinMode(PIN_RELAY_4, OUTPUT);
}
```

Keterangan	Persiapan awal ketika wemos mega 2560 pertama kali dinyalakan sehingga perlu inisialisasi awal agar mendapatkan nilai default.

Potongan Kode Program Ke-6

Gambar

```
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  timer = pulseIn(echoPin, HIGH);
  ultrasonicValue = timer * 0.034 / 2;;

  turbidityValue = analogRead(turbidity);

  Serial.print ("Sensor Output (V):");
  Serial.print (turbidityValue);
  Serial.print (" || ");
  Serial.print ("Jarak (cm):");
  Serial.println (ultrasonicValue);
```

Keterangan	Inisiasi value dari sensor *ultrasonic* dan sensor *turbidity* agar dapat digunakan oleh peneliti untuk membuat kondisi.

Potongan Kode Program Ke-7

Gambar

```
// Kondisi 1 (Air Keruh)
if (turbidityValue <= 430) {
  motor(255, 255);
  digitalWrite(PIN_RELAY_1, LOW); // Toren -> Rumah (Aktif)
  digitalWrite(PIN_RELAY_2, HIGH);  // Sumur -> Rumah (Aktif)
  digitalWrite(PIN_RELAY_3, HIGH);  // Toren -> Pembuangan (Tidak Aktif)
  digitalWrite(PIN_RELAY_4, LOW); // Sumur -> Toren (Aktif)

  if (ultrasonicValue >= 13) {
    motor2(255, 0); //Motor DC
    motor(150, 255); //Pompa Toren (Aktif) || Pompa Sumur (Aktif)
    delay(20000);
    motor2(0, 0);
    motor(150, 255); //Pompa Toren (Aktif) || Pompa Sumur (Aktif)
    digitalWrite(PIN_RELAY_1, HIGH); // Toren -> Rumah (Aktif)
    digitalWrite(PIN_RELAY_2, LOW); // Sumur -> Rumah (Tidak Aktif)
    digitalWrite(PIN_RELAY_3, LOW); // Toren -> Pembuangan (Tidak Aktif)
    digitalWrite(PIN_RELAY_4, HIGH); // Sumur -> Toren (Aktif)
  }
  else {
    motor(255, 255); //Pompa Toren (Aktif) || Pompa Sumur (Aktif)
    motor2(0, 0);
    digitalWrite(PIN_RELAY_1, LOW); // Toren -> Rumah (Tidak Aktif)
    digitalWrite(PIN_RELAY_2, HIGH); // Sumur -> Rumah (Aktif)
    digitalWrite(PIN_RELAY_3, HIGH); // Toren -> Pembuangan (Aktif)
    digitalWrite(PIN_RELAY_4, LOW); // Sumur -> Toren (Aktif)
  }
}
```

Keterangan	Kondisi pertama berdasarkan tingkat kekeruhan air dan level air yang terdapat di dalam toren. Kondisi ini akan aktif ketika sensor *turbidity* mendeteksi tingkat kekeruhan apabila mencapai ambang bawah yaitu lebih kecil sama dengan 430.

Potongan Kode Program Ke-8

Gambar

```
// Kondisi Air Tidak Keruh
else if (turbidityValue >= 431) {
  motor(255, 0); //Pompa Toren (Aktif) || Pompa Sumur (Tidak Aktif)
  motor2(0, 0); //Motor DC
  digitalWrite(PIN_RELAY_1, HIGH); // Toren -> Rumah (Aktif)
  digitalWrite(PIN_RELAY_2, LOW);  // Sumur -> Rumah (Tidak Aktif)
  digitalWrite(PIN_RELAY_3, LOW);  // Toren -> Pembuangan (Tidak Aktif)
  digitalWrite(PIN_RELAY_4, LOW); // Sumur -> Toren (Tidak Aktif)
  if (ultrasonicValue >= 13) {
    motor(150, 255); //Motor DC
    motor2(0, 0); //Pompa Toren (Aktif) || Pompa Sumur (Aktif)
    digitalWrite(PIN_RELAY_1, HIGH); // Toren -> Rumah (Aktif)
    digitalWrite(PIN_RELAY_2, LOW); // Sumur -> Rumah (Tidak Aktif)
    digitalWrite(PIN_RELAY_3, LOW); // Toren -> Pembuangan (Tidak Aktif)
    digitalWrite(PIN_RELAY_4, HIGH); // Sumur -> Toren (Aktif)
  } else if (ultrasonicValue <=8) {
    motor(255, 0); //Motor DC
    motor2(0, 0); //Pompa Toren (Aktif) || Pompa Sumur (Tidak Aktif)
    digitalWrite(PIN_RELAY_1, HIGH); // Toren -> Rumah (Aktif)
    digitalWrite(PIN_RELAY_2, LOW); // Sumur -> Rumah (Tidak Aktif)
    digitalWrite(PIN_RELAY_3, LOW); // Toren -> Pembuangan (Tidak Aktif)
    digitalWrite(PIN_RELAY_4, LOW); // Sumur -> Toren (Aktif)
  }
}
```

Keterangan

Kondisi kedua berdasarkan tingkat kekeruhan air dan level air yang terdapat di dalam toren. Kondisi ini akan aktif ketika sensor *turbidity* mendeteksi tingkat kekeruhan diatas ambang bawah yaitu lebih besar sama dengan 431.

Potongan Kode Program Ke-9

Gambar

```
// To Esp8266
String kirim = "";


kirim = "";
kirim += ultrasonicValue;
kirim += ";";
kirim += turbidityValue;

Serial3.println(kirim);

if(Serial3.available()) {
  String msg = "";
  while (Serial3.available()) {
    msg += char(Serial3.read());
    delay (50);
  }
  Serial.println(msg);
}
```

Keterangan

Sumber kode diatas memiliki fungsi sebagai pengirim data yang telah diperoleh oleh mikrokontroler mega 2566 melalui sensor-sensor yang saling terhubung dan hasil data yang diperoleh akan dikirimkan ke mikrokontroler ESP8266.

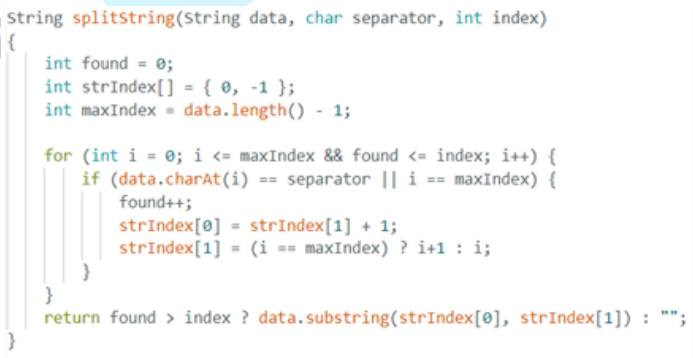
Potongan Kode Program Ke-10

Gambar	
Keterangan	Definisi library koneksi dari ESP8266 agar dapat terkoneksi dengan Wifi yang terlebih dahulu dikonfigurasikan melalui SSID dan password agar dapat mengirim data ke database dan server.

Potongan Kode Program Ke-11

Gambar	
Keterangan	Sebuah fungsi yang digunakan untuk menghubungkan perangkat dengan jaringan Wifi. Fungsi ini antara lain digunakan untuk mengatur mode Wifi dan memulai koneksi jaringan dengan SSID dan password sebelumnya.

Potongan Kode Program Ke-12

Gambar	
Keterangan	Fungsi ini digunakan untuk membagi sebuah string berdasarkan karakter pemisah (separator) dan mengembalikan bagian string yang sesuai dengan indeks yang diberikan (index).

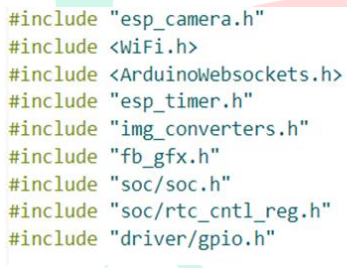
Potongan Kode Program Ke-13

Gambar	

```
void kirimDataKeServer(){
  WiFiClientSecure client;
  client.setInsecure();
  HTTPClient http;
  String postData;
  //Post Data
  postData = "ultrasonic=";
  postData += ultrasonicValue;
  postData += "&turbidity=";
  postData += turbidityValue;

  http.begin(client, "https://torenku.my.id/espRehang/post-data.php");
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  int httpCode = http.POST(postData);
  String payload = http.getString();

  Serial.println(httpCode);
  Serial.println(payload);

  http.end();
}
```

Keterangan	Fungsi ini digunakan mengirimkan data yang didapatkan melalui sensor dan berhasil diolah dengan perangkat ESP8266. Hasil dari data tersebut akan dikirimkan ke server dan akan disimpan ke dalam database.

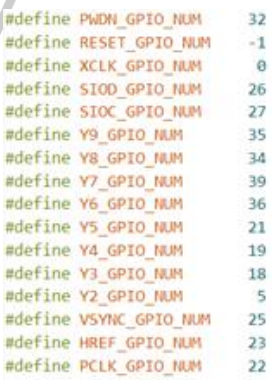
Potongan Kode Program Ke-14

Gambar	

```
#include "esp_camera.h"
#include <WiFi.h>
#include <ArduinoWebsockets.h>
#include "esp_timer.h"
#include "img_converters.h"
#include "fb_gfx.h"
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "driver/gpio.h"
```

Keterangan	Definisi library yang digunakan ESP32-CAM agar dapat mengaktifkan kamera OV7670. Definisi ini digunakan untuk menangkap gambar melalui kamera yang sudah di konfigurasi melalui esp_camera.h.

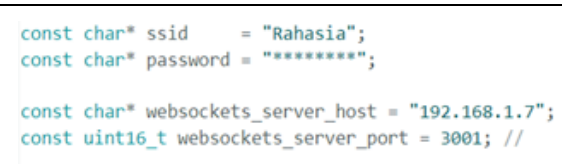
Potongan Kode Program Ke-15

Gambar	

```
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22
```

Keterangan	Definisi varibel yang digunakan ESP32-CAM agar dapat terkoneksi dengan dev board usb

Potongan Kode Program Ke-16

Gambar	
Keterangan	Proses penetapan konfigurasi melalui SSID dan password agar dapat mengirim image ke server

Potongan Kode Program Ke-16

Gambar	
Keterangan	Fungsi ini mengkonfigurasi berbagai pengaturan kamera, memanggil fungsi inisialisasi kamera, dan mengatur ukuran frame.

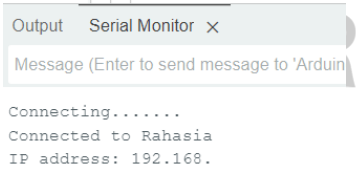
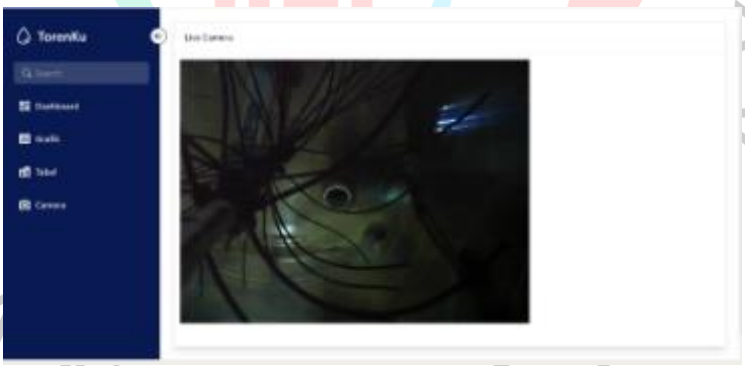
5.2 Pembahasan

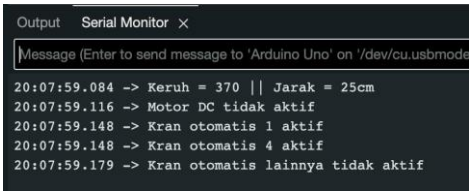
Subbab ini akan memaparkan tentang posisi alat pada saat pengujian, pengujian alat dengan metode *Black Box*.

5.2.1 Hasil Pengujian *Black Box*

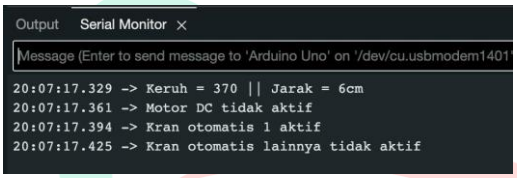
Hasil pengujian dari *Black Box* ini akan ditampilkan gambar dari setiap pengujian yang akan dilakukan. Gambar tersebut dapat mengetahui berhasil atau tidaknya sistem yang telah dibuat. Pengujian tersebut dijelaskan pada tabel 5.2.

Tabel 5. 2 Hasil Pengujian Black Box

Pengujian Black Box 1	
Skenario Pengujian	Rangkaian dihidupkan.
Hasil yang diharapkan	Wemos Mega 2566 dapat terhubung dengan internet dan mendapatkan alamat IP wifi.
Hasil Pengujian	
Pengujian Black Box 2	
Skenario Pengujian	ESP32-CAM dihidupkan.
Hasil yang diharapkan	ESP32-CAM dapat menangkap gambar menggunakan kamera OV7670 dan dapat mengirim ke server.
Hasil Pengujian	
Pengujian Black Box 3	
Skenario Pengujian	Sensor *turbidity* mengukur tingkat kekeruhan dengan nilai >= 350 dan sensor *ultrasonic* mengukur tingkat level air dengan nilai >= 20 cm.
Hasil yang diharapkan	Motor DC pembersih toren tidak aktif, kran 1 dari toren menuju rumah aktif, dan kran 4 dari sumur menuju toren aktif, dan kran lainnya akan tertutup.

Hasil Pengujian	

Pengujian Black Box 4

Skenario Pengujian	Sensor *turbidity* mengukur tingkat kekeruhan dengan nilai >= 350 dan sensor *ultrasonic* mengukur tingkat level air dengan nilai <= 8 cm.
Hasil yang diharapkan	Motor DC pembersih toren tidak aktif dan kran 1 dari toren menuju rumah aktif, dan kran lainnya akan tertutup.
Hasil Pengujian	

Pengujian Black Box 5

Skenario Pengujian	Sensor *turbidity* mengukur tingkat kekeruhan dengan nilai <= 349 dan sensor *ultrasonic* mengukur tingkat level air dengan nilai <=20 cm.
Hasil yang diharapkan	Motor DC pembersih toren aktif, kran 2 dari sumur menuju rumah aktif, dan kran 3 dari toren menuju pembuangan aktif, dan kran lainnya akan tertutup.
Hasil Pengujian