



**REAL TIME SYSTEM AND INTERNET OF THINGS FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA**

WATER QUALITY MONITORING SYSTEM

GROUP B7

FAHREZY H	2106731466
FARRAS RAFI PERMANA	2106700990
MUHAMMAD RIZKY UTOMO	2106731320
ZULFIKAR HADZALIC	2106636224

PREFACE

Kualitas air adalah aspek krusial dalam menjaga keberlangsungan lingkungan dan kesehatan masyarakat. Dalam upaya untuk memahami dan mengontrol faktor-faktor yang memengaruhi kualitas air, proyek ini menghadirkan solusi inovatif melalui pemanfaatan teknologi Real-time System dan IoT untuk memonitor kualitas air di lingkungan.

Proyek ini bertujuan untuk menciptakan sistem pemantauan kualitas air yang dapat memberikan informasi real-time tentang kondisi air di lingkungan. Melalui penggunaan perangkat ESP32 dan sensor kualitas air yang terhubung dengan platform Thingsboard, sistem ini akan memungkinkan pengukuran yang akurat terhadap parameter-parameter kritis seperti suhu, kekeruhan, dan jumlah zat terlarut.

Laporan ini menguraikan secara rinci tentang perancangan, pengembangan, serta implementasi sistem pemantauan kualitas air yang dilengkapi dengan sensor-sensor yang mampu membaca parameter suhu, Total Dissolved Solids (TDS), dan kekeruhan air. Proyek ini menggunakan ESP32 sebagai pusat pengendali, mendukung koneksi WiFi untuk mentransmisikan data ke platform IoT ThingsBoard. Laporan ini juga menyajikan detail mengenai desain hardware, pengembangan software, serta integrasi antara hardware dan software yang diterapkan.

Dengan adanya laporan ini, diharapkan dapat memberikan gambaran yang jelas dan komprehensif tentang implementasi teknologi untuk memantau kualitas air secara real-time, serta manfaat yang diperoleh dari sistem monitoring kualitas air pada lingkungan masyarakat.

Depok, December 10, 2023

Group B7

TABLE OF CONTENTS

CHAPTER 1	4
INTRODUCTION	4
1.1 PROBLEM STATEMENT	4
1.3 ACCEPTANCE CRITERIA	5
1.4 ROLES AND RESPONSIBILITIES	5
1.5 TIMELINE AND MILESTONES	6
CHAPTER 2	7
IMPLEMENTATION	7
2.1 HARDWARE DESIGN AND SCHEMATIC	7
2.2 SOFTWARE DEVELOPMENT	8
2.3 HARDWARE AND SOFTWARE INTEGRATION	11
CHAPTER 3	13
TESTING AND EVALUATION	13
3.1 TESTING	13
3.2 RESULT	13
3.3 EVALUATION	19
CHAPTER 4	20
CONCLUSION	20

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Air merupakan unsur bumi yang menjadi kebutuhan hidup bagi seluruh makhluk. Dalam kegiatan-kegiatan yang dilakukan manusia, air memainkan peran sentral sebagai cairan serbaguna dan memiliki berbagai kegunaan dalam berbagai bidang, seperti dalam pertanian, industri, pertambangan, dan, yang paling diketahui, sebagai sumber air minum. Kebanyakan kegiatan ini membutuhkan air bersih untuk memastikan efektivitas pekerjaan dan mencegah adanya zat-zat yang tidak diinginkan sehingga dianggap aman.

Agar air yang digunakan dianggap sebagai air yang aman untuk sebuah kegunaan, air tersebut memiliki karakteristik-karakteristik yang harus dipenuhi. Dalam air minum sendiri, karakteristik air yang aman untuk diminum adalah tidak memiliki bau, warna, dan rasa. Selain dari karakteristik fisik, air tersebut harus memiliki atribut-atribut kimiawi tertentu. Dalam Peraturan Menteri Kesehatan Nomor 2 Tahun 2023, 2 dari 19 parameter yang disebutkan untuk air minum adalah berada dalam suhu udara dan memiliki benda solid kurang dari 300 mg/L [1]. Masalahnya, atribut-atribut seperti ini sering kali tidak dapat dideteksi melalui observasi biasa. Hal ini menyulitkan masyarakat awam, terutama mereka yang masih harus mengambil air dari sumur.

Dari masalah seperti ini, para penulis menawarkan sebuah solusi, yaitu Water Quality Monitoring System (Sistem Pemantauan Kualitas Air). Sistem ini mengintegrasikan sebuah perangkat IoT yang memeriksa kualitas air dengan beberapa sensor dan memberikan peringatan jika kualitas air yang ada tidak sesuai dengan yang diinginkan. Selain itu, sistem ini memudahkan pemantauan dengan memberikan antarmuka tempat melihat kualitas air secara *real-time*.

1.2 PROPOSED SOLUTION

Solusi yang diusulkan adalah pengembangan Water Quality Monitoring System yang terkoneksi secara langsung dengan platform IoT, menggunakan perangkat keras ESP32 sebagai pusat pengendali dan pembacaan sensor. Sistem ini akan dilengkapi dengan sensor-sensor

khusus seperti SEN0244 TDS Sensor untuk mengukur *Total Dissolved Solids* (TDS), SEN0189 Turbidity Sensor untuk mengukur tingkat kekeruhan air, dan DS18B20 Temperature Sensor untuk memantau suhu air.

Melalui koneksi WiFi dan MQTT, ESP32 akan mengumpulkan data dari sensor-sensor tersebut secara terus-menerus. Data-data ini akan dikirimkan ke ESP32 pengatur melalui MQTT, kemudian akan dikirimkan pada platform IoT Thingsboard untuk pengolahan data lebih lanjut. Di sini, data akan dianalisis dan divisualisasikan sehingga dapat diakses secara real-time melalui antarmuka pengguna.

Solusi ini diharapkan dapat memberikan pemantauan kualitas air yang lebih akurat, real-time, dan mudah diakses. Hal ini akan membantu dalam pengambilan keputusan yang cepat terkait kualitas air dan memungkinkan adopsi tindakan yang tepat waktu untuk menjaga kesehatan lingkungan dan masyarakat.

1.3 ACCEPTANCE CRITERIA

Kriteria keberhasilan dari proyek ini ditentukan oleh faktor berikut:

1. Keberhasilan koneksi dan komunikasi, yang mencakup stabilitas koneksi antar perangkat keras (ESP32) dan antara perangkat keras (ESP32) dan platform IoT (Thingsboard), dan respons dalam mentransmisikan data sensor ke platform IoT dalam batas waktu yang dapat diterima.
2. Integrasi perangkat keras (ESP32 dan sensor) dengan perangkat lunak (Thingsboard), sehingga data dari sensor dapat diakses dan diolah dengan baik oleh platform IoT.
3. Keberhasilan pembacaan sensor pembaca kualitas air (TDS, turbidity, temperature) dalam mengukur parameter-parameter yang ditentukan, konsistensi pembacaan dan respons sensor.

1.4 ROLES AND RESPONSIBILITIES

Peran dan tanggung jawab masing-masing anggota kelompok terbagi menjadi berikut:

Roles	Responsibilities	Person
-------	------------------	--------

Brainstorming dan Design	Memikirkan ide desain, membuat prototype, dan meng-spesifikasi komponen yang dibutuhkan pada proyek.	Farras, Zulfikar, Rizki, Fahrezy
Perancang perangkat keras	Menyusun komponen-komponen perangkat keras yang ditentukan menjadi sebuah kesatuan dengan struktur yang didokumentasikan.	Zulfikar, Fahrezy, Farras
Kode dan Software	Menulis kode dan merancang perangkat lunak.	Farras, Zulfikar
Laporan dan Dokumentasi	Membuat laporan yang komprehensif, membuat user manual, dan mengumpulkan dokumentasi	Rizki, Farras
Testing dan Evaluasi	Melakukan pengujian terhadap solusi yang telah dibuat dan memastikan bahwa semua fungsionalitas bekerja dan memenuhi kriteria yang telah disepakati.	Farras, Zulfikar, Rizki, Fahrezy

Tabel 1. Peran dan Tanggung Jawab

1.5 TIMELINE

Berikut adalah Gantt Chart untuk proyek ini:

Tahap		November										Desember									
		21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10
Brainstorming																					
Perancangan perangkat keras	Penentuan komponen																				
	Penyusunan komponen																				
Pengembangan software	Kode master																				
	Kode slave dan sensor																				
	Setup Thingsboard																				
Integrasi perangkat keras dan software																					
Pengujian dan evaluasi																					
Dokumentasi																					

Tabel 2. Linimasa dan Target

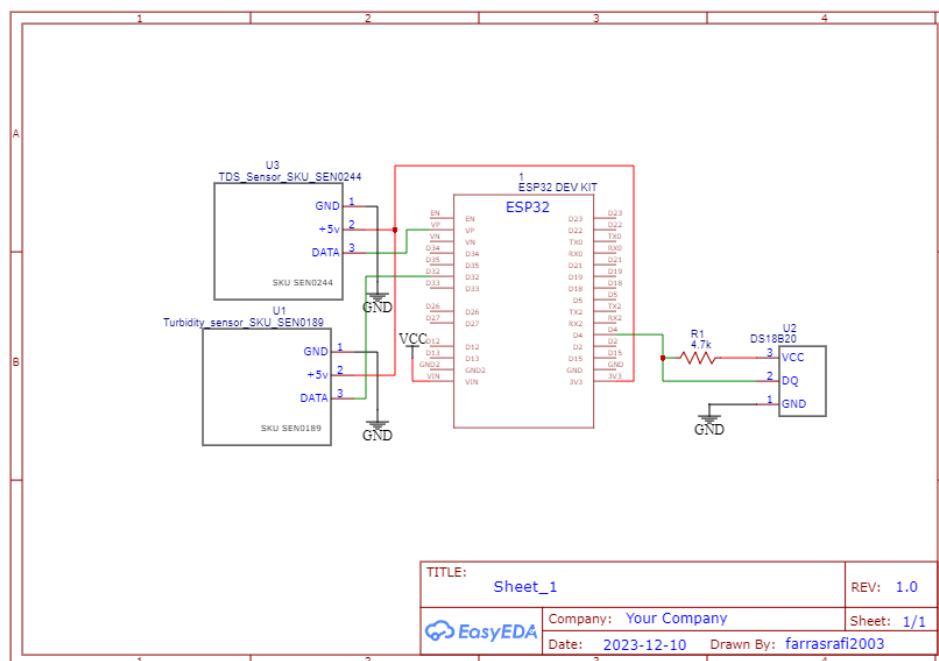
CHAPTER 2

IMPLEMENTATION

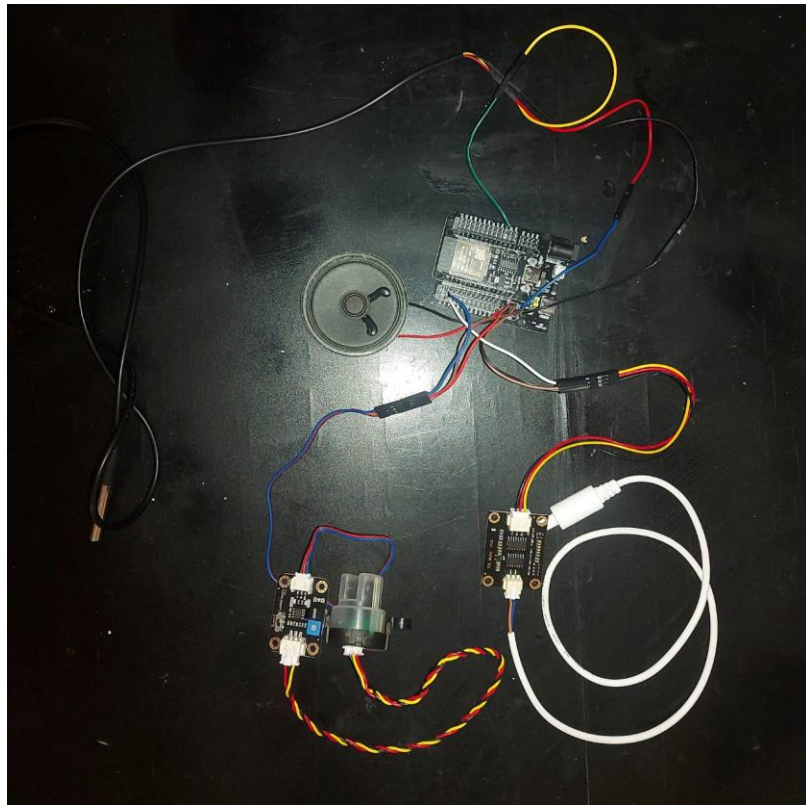
2.1 HARDWARE DESIGN AND SCHEMATIC

Dalam pembuatan sistem monitoring kualitas air, dibutuhkan banyak komponen perangkat keras yang perlu diselaraskan. Sistem ini menggunakan dua buah ESP32, yang terdiri dari ESP32 untuk pembacaan sensor, dan ESP32 untuk menerima data dan mengirimkan data ke Thingsboard.

Pada ESP32 untuk pembacaan sensor, sensor DS18B20 Temperature Sensor bertugas mengukur suhu pada air. Selain itu, sebuah sensor SEN0244 TDS Sensor bertugas mengukur total dissolved Solid (TDS) dalam air, dan sebuah SEN0189 Turbidity Sensor bertugas mengukur tingkat kekeruhan dalam air. Data yang dibaca oleh ESP32 ini akan diteruskan dengan menggunakan protokol MQTT ke ESP32 server yang berfungsi untuk menerima dan mengirimkan data ke Thingsboard. Kedua ESP32 yang digunakan akan terkoneksi dengan internet melalui jaringan Wi-Fi.



Gambar 1. Skematik Rangkaian ESP32 Sensor



Gambar 2. Foto Rangkaian

2.2 SOFTWARE DEVELOPMENT

Pada proyek ini, para penyusun menggunakan bahasa pemrograman C++ untuk membangun program dari kedua ESP32 yang terdiri dari dua bagian utama:

a. ESP32 Master

Pada ESP32 yang berperan sebagai master atau sebagai penerima dan pengirim data, terdapat beberapa fungsionalitas utama seperti mengumpulkan data dari sensor-sensor melalui MQTT, dan mengirimkannya ke platform ThingsBoard. Koneksi Wi-Fi akan membuat dua client, yang pertama akan meng-handle koneksi dari MQTT dan yang kedua akan meng-handle koneksi ke Thingsboard.

Perangkat akan melakukan koneksi ke Wi-Fi menggunakan WiFiSecure yang dilanjutkan dengan menjalankan task untuk koneksi ke MQTT setelah koneksi WiFi berhasil. Setelah koneksi MQTT berhasil dan telah berlangganan dengan topik yang

dispesifikasikan, perangkat akan mengirimkan pesan pembuka dan siap untuk membaca data dari ESP32 sensor yang ditransmisikan dengan format JSON.

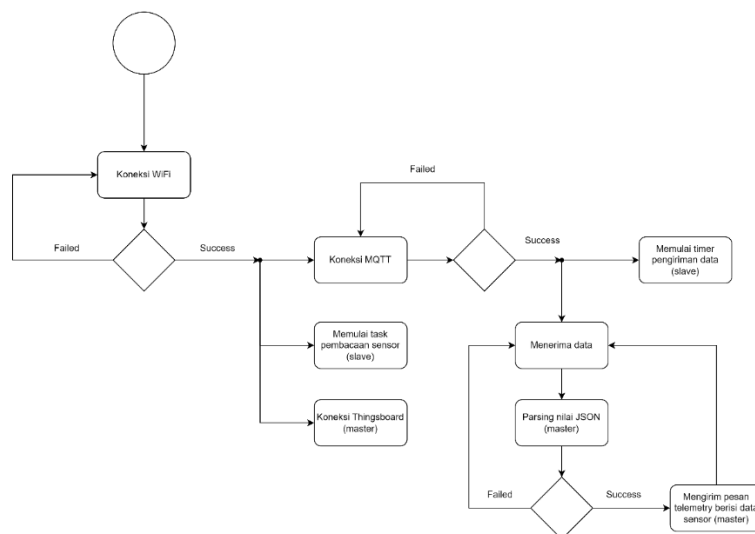
Koneksi ke Thingsboard akan dijalankan pada fungsi loop dan akan melakukan reconnection bila koneksi terputus. Untuk pengiriman data (telemetry), terdapat tiga parameter yang dikirimkan, yaitu temperature, tds, dan turbidity. Data dikirimkan sesaat setelah menerima data dari MQTT melalui fungsi callback, dan hanya dikirimkan jika data yang diterima valid (sukses di parse dari JSON).

b. ESP32 Slave

Pada ESP32 yang berperan sebagai pembaca sensor, terdapat dua fungsionalitas utama yaitu membaca sensor dan mengirimkan data ke MQTT.

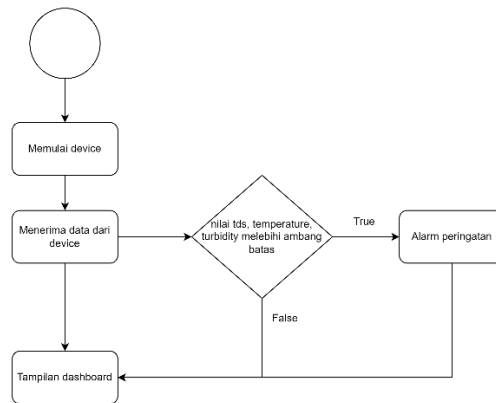
Perangkat akan melakukan koneksi ke Wi-Fi menggunakan WiFiSecure, setelah itu, dilanjutkan dengan menjalankan task untuk koneksi ke MQTT setelah koneksi WiFi berhasil. Setelah koneksi MQTT berhasil dan telah berlangganan dengan topik yang dispesifikasikan, perangkat akan mengirimkan pesan pembuka dan siap untuk membaca data dari ESP32 sensor yang ditransmisikan dengan format JSON.

Setelah koneksi MQTT berhasil, perangkat akan memulai task untuk pembacaan sensor. Setiap sensor akan melakukan pembacaan dengan waktu yang bervariasi, bergantung pada spesifikasi dan respons dari sensor. Untuk pengiriman data ke MQTT, akan menggunakan timer callback yang akan dipanggil setiap lima detik untuk memastikan data yang diterima tersinkronisasi dan tidak overloaded.

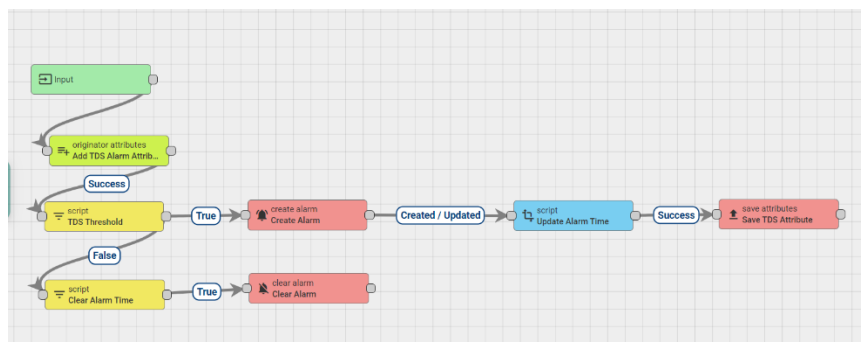


Gambar 3. Flowchart Perangkat Master dan Slave

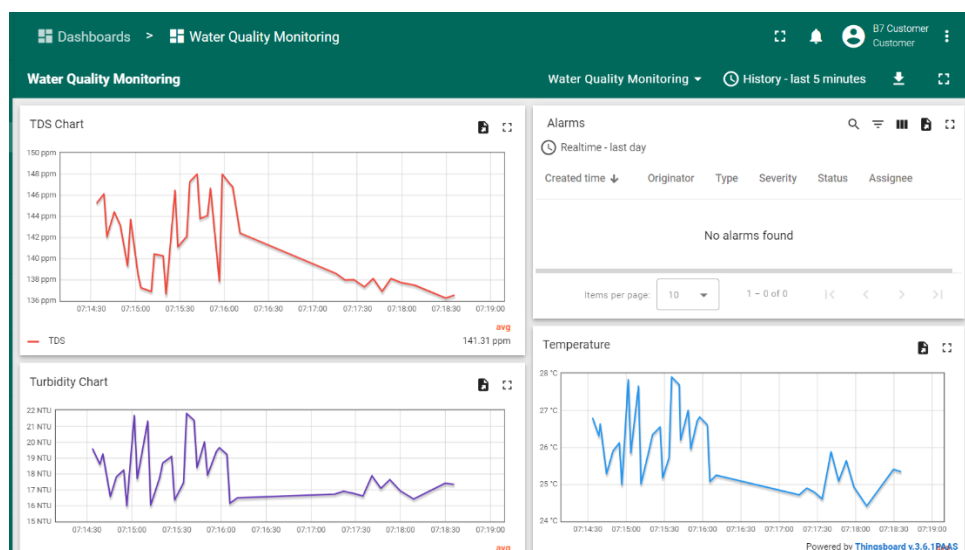
Thingsboard digunakan untuk menampilkan dan monitoring hasil dari pembacaan sensor yang dikirimkan dari ESP32 master. Dashboard untuk user akan terdiri dari line chart pembacaan sensor dan notifikasi alarm yang akan terlihat jika nilai dari TDS, temperature, dan turbidity melebihi ambang batas yang ditetapkan pada rule chains Thingsboard.



Gambar 4. Flowchart Thingsboard



Gambar 5. Alur Rule Chain Alarm

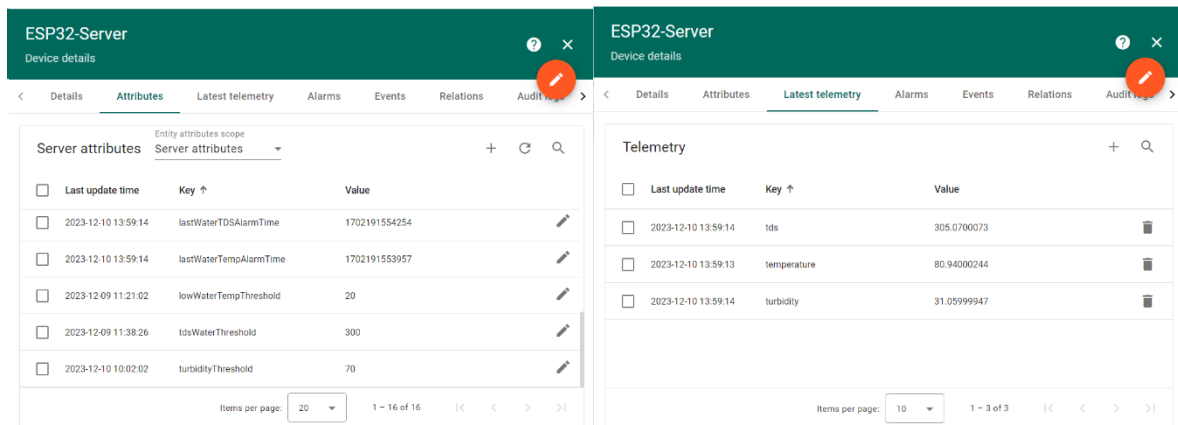


Gambar 6. Tampilan Dashboard User

2.3 HARDWARE AND SOFTWARE INTEGRATION

Dalam integrasi perangkat keras dan perangkat lunak, fokus utamanya adalah mengintegrasikan kode program yang telah dikembangkan, untuk diimplementasikan ke perangkat kerasnya seperti ESP32 dan sensor-sensor pembaca kualitas air. Integrasi ini bertujuan untuk memastikan bahwa semua elemen bekerja bersama secara efektif dan sesuai dengan tujuan proyek.

Integrasi antara perangkat keras ESP32 dengan perangkat lunak Thingsboard dilakukan dengan menspesifikasikan kredensial dari perangkat keras yang terdaftar pada Thingsboard, kemudian melakukan koneksi pada ESP32 menggunakan kredensial tersebut. Data yang dikirimkan dari ESP32 server ke Thingsboard disebut dengan Telemetry, dan pada proyek ini, ESP32 server memiliki tiga data yaitu tds, temperature, dan turbidity. Selain Telemetry, perangkat keras yang terhubung dengan Thingsboard akan memiliki attribute yang digunakan untuk menjalankan logika pada rule chain.

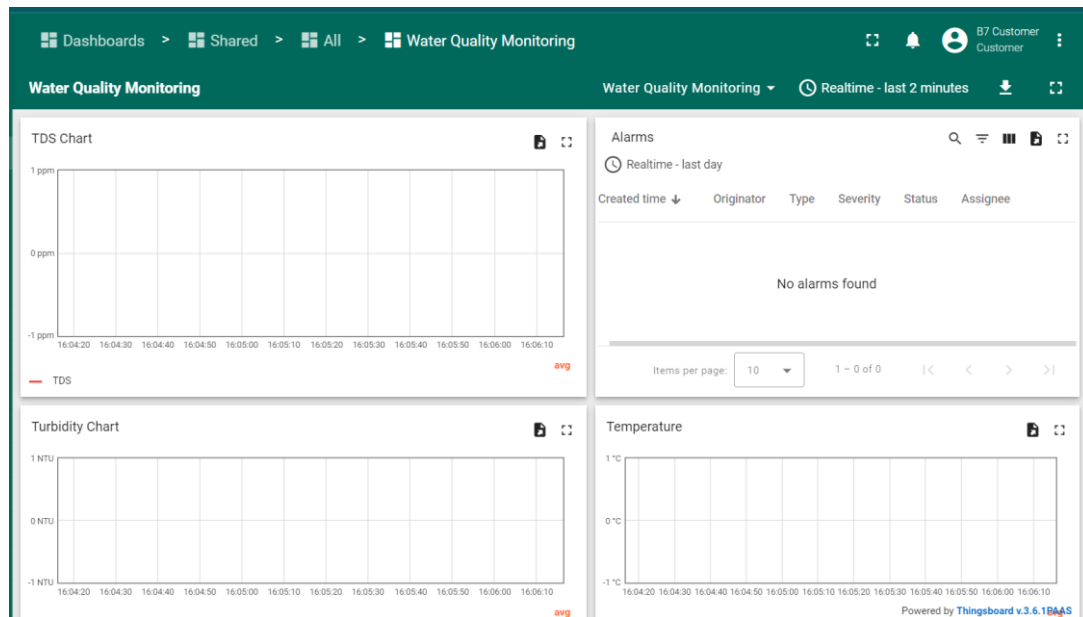


Last update time	Key	Value
2023-12-10 13:59:14	lastWaterTDSAlarmTime	1702191554254
2023-12-10 13:59:14	lastWaterTempAlarmTime	1702191553957
2023-12-09 11:21:02	lowWaterTempThreshold	20
2023-12-09 11:38:26	tdsWaterThreshold	300
2023-12-10 10:02:02	turbidityThreshold	70

Last update time	Key	Value
2023-12-10 13:59:14	tds	305.0700073
2023-12-10 13:59:13	temperature	80.94000244
2023-12-10 13:59:14	turbidity	31.05999947

Gambar 6. Tampilan Data Pada ESP32 server

Terdapat akun khusus untuk customer yang digunakan untuk memantau perangkat yang bekerja dan didesain sebagai akun yang bersifat read-only, sehingga akun ini dijadikan sebagai akun untuk end user dan tidak memerlukan pengaturan lebih lanjut.



Gambar 7. Tampilan Dashboard untuk end user

Setelah perangkat lunak diimplementasikan pada ESP32, serangkaian pengujian dilakukan untuk memastikan bahwa interaksi antara perangkat keras dan perangkat lunak bekerja seperti yang diharapkan. Hal ini melibatkan proses "trial and error", sehingga para penyusun dapat memodifikasi kode untuk memenuhi kriteria proyek ini.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

Pada tahap pengujian, para penyusun melakukan serangkaian tes untuk memastikan bahwa program berjalan sesuai dengan kriteria yang sudah ditentukan sebelumnya. Tahapan pengujian meliputi keberhasilan koneksi dan pengiriman data, baik antara ESP32 pembaca sensor dengan ESP32 server, maupun antara ESP32 server dengan software Thingsboard. Tahapan selanjutnya adalah memastikan integrasi antara ESP32 server dengan Thingsboard, sehingga software Thingsboard dapat membaca data dan melakukan aksi yang telah ditentukan dengan benar. Tahapan terakhir adalah menguji keberhasilan pembacaan sensor kualitas air yang terdiri dari sensor TDS, temperature, dan turbidity.

3.2 RESULT

Berdasarkan testing yang sudah ditentukan, para penyusun mendapatkan hasil sebagai berikut sesuai dengan masing-masing kriteria keberhasilan. Untuk testing pertama yaitu keberhasilan koneksi antara perangkat dan software, dapat dilihat bahwa kedua ESP32 dapat terhubung dengan MQTT dan Wi-Fi, sedangkan ESP32 server dapat terhubung dengan Thingsboard dengan baik.

```
13:59:30.209 -> Connecting to WiFi network: ZAYDA
13:59:30.848 -> ..
13:59:31.353 -> WiFi connected successfully:
13:59:31.353 -> IP address:
13:59:31.353 -> 192.168.1.37
13:59:31.353 -> Menghubungkan ke ThingsBoard node ...Connecting to broker.hivemq.com[DONE]
13:59:34.703 ->
13:59:34.703 -> Subscribed to FinalProject/B7/water_quality
13:59:34.703 -> Connected to hive MQTT broker
13:59:35.016 -> Pesan Diterima: Hello from master
```

Gambar 8. Hasil Serial Monitor ESP32 server

```
14:04:56.517 -> Connected
14:04:56.517 -> Connected! Local IP: 192.168.1.37
14:04:56.517 -> Connecting to broker.hivemq.com
14:04:58.479 -> Subscribed to FinalProject/B7/water_quality
14:04:58.479 -> Connected to hive MQTT broker
14:04:58.698 -> Pesan Diterima: Inital Message from Device 1
```

Gambar 9. Hasil Serial Monitor ESP32 sensor reader

Devices Device Filter Include customer entities							
<input type="checkbox"/>	Created time ↓	Name	Device profile	Label	State	Customer name	Groups
<input type="checkbox"/>	2023-12-09 09:08:31	ESP32-Server	default		Active	Project Customer	Project-Group

Gambar 10. Tampilan list devices pada Thingsboard

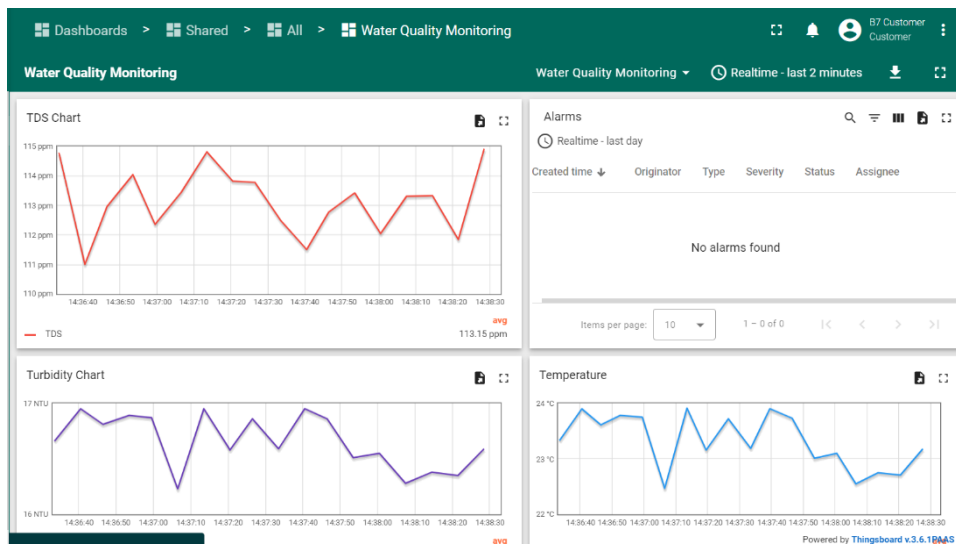
Untuk testing kedua yaitu integrasi antar ESP32 dan antara ESP32 server dengan Thingsboard, dapat dilihat bahwa ESP32 server dapat menerima data dari ESP32 sensor, dan ESP32 server dapat mengirimkan data ke Thingsboard dan menampilkan hasilnya.

```

14:36:00.217 -> Pesan Diterima: Initial Message from Device 1
14:36:00.217 -> deserializeJson() failed: InvalidInput
14:36:06.864 -> Pesan Diterima: {"temperature":22.64,"tds":114.93,"turbidity":16.32}
14:36:13.132 -> Pesan Diterima: {"temperature":23.84,"tds":114.24,"turbidity":16.92}
14:36:19.983 -> Pesan Diterima: {"temperature":23.11,"tds":113.06,"turbidity":16.56}
14:36:26.229 -> Pesan Diterima: {"temperature":23.75,"tds":112.33,"turbidity":16.88}
14:36:32.773 -> Pesan Diterima: {"temperature":23.33,"tds":114.78,"turbidity":16.66}
14:36:39.242 -> Pesan Diterima: {"temperature":23.90,"tds":111.01,"turbidity":16.95}
14:36:45.475 -> Pesan Diterima: {"temperature":23.61,"tds":112.96,"turbidity":16.81}
14:36:51.800 -> Pesan Diterima: {"temperature":23.78,"tds":114.04,"turbidity":16.89}
14:36:58.489 -> Pesan Diterima: {"temperature":23.75,"tds":112.36,"turbidity":16.87}
14:37:05.228 -> Pesan Diterima: {"temperature":22.47,"tds":113.43,"turbidity":16.23}
14:37:12.399 -> Pesan Diterima: {"temperature":23.91,"tds":114.81,"turbidity":16.95}
14:37:19.170 -> Pesan Diterima: {"temperature":23.16,"tds":113.82,"turbidity":16.58}
14:37:25.799 -> Pesan Diterima: {"temperature":23.72,"tds":113.78,"turbidity":16.86}
14:37:32.482 -> Pesan Diterima: {"temperature":23.19,"tds":112.49,"turbidity":16.59}
14:37:39.146 -> Pesan Diterima: {"temperature":23.90,"tds":111.51,"turbidity":16.95}
14:37:45.859 -> Pesan Diterima: {"temperature":23.73,"tds":112.78,"turbidity":16.86}

```

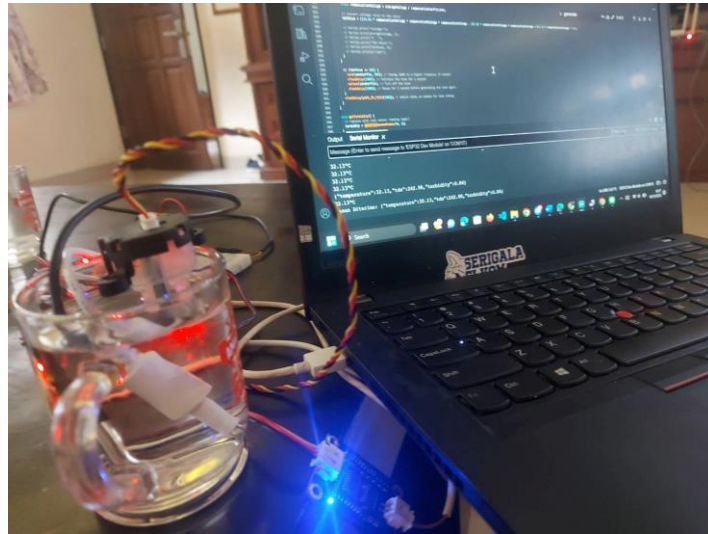
Gambar 11. Hasil pengiriman data dari ESP32 sensor



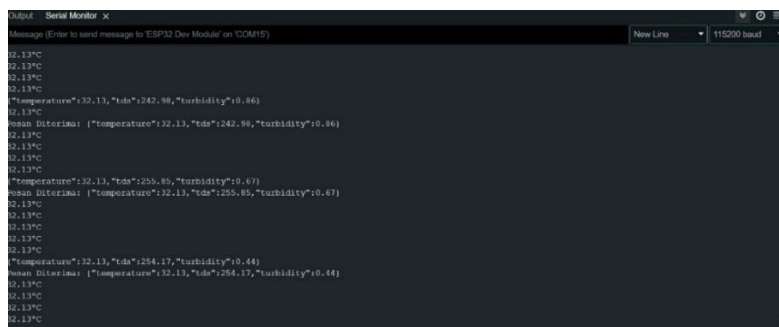
Gambar 12. Hasil pengiriman dari ESP32 server

Untuk testing berikutnya adalah menguji keberhasilan pembacaan sensor kualitas air yang terdiri dari sensor TDS, temperature, dan turbidity. Testing ini akan terbagi menjadi beberapa bagian dengan menggunakan bahan air yang berbeda-beda.

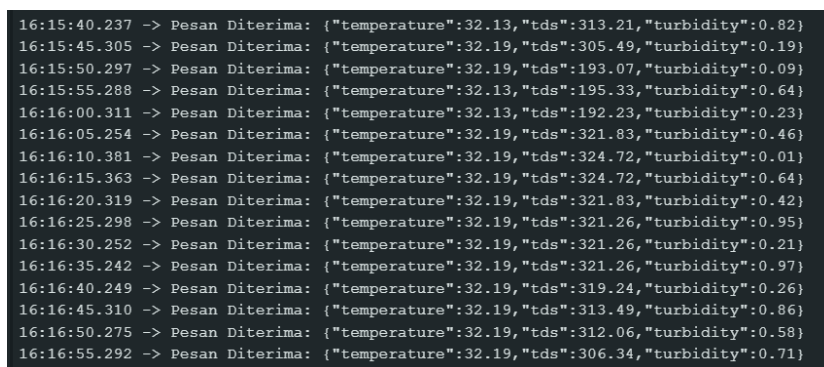
Kasus pertama adalah menguji sensor pada kondisi air yang bersih dan pada suhu ruangan.



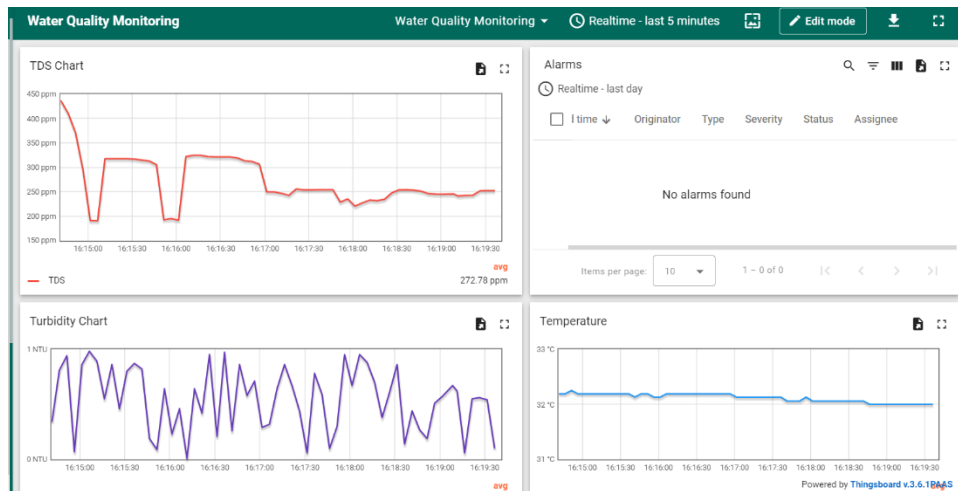
Gambar 13. Rangkaian untuk uji coba kasus pertama



Gambar 14. Serial Monitor ESP32 sensor untuk uji coba kasus pertama

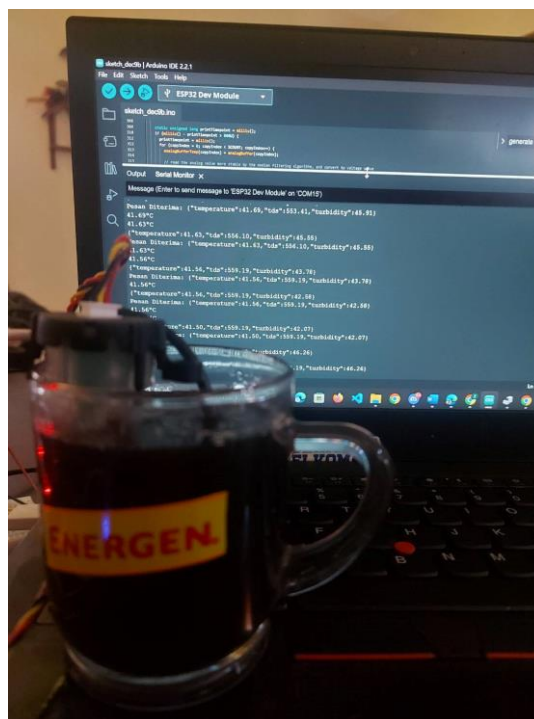


Gambar 15. Serial Monitor ESP32 server untuk uji coba kasus pertama



Gambar 16. Dashboard Thingsboard untuk uji coba kasus pertama

Kasus kedua adalah menguji sensor pada air keruh berwarna kecoklatan, dan ditambahkan dengan 1 sendok garam.



Gambar 17. Rangkaian untuk uji coba kasus kedua

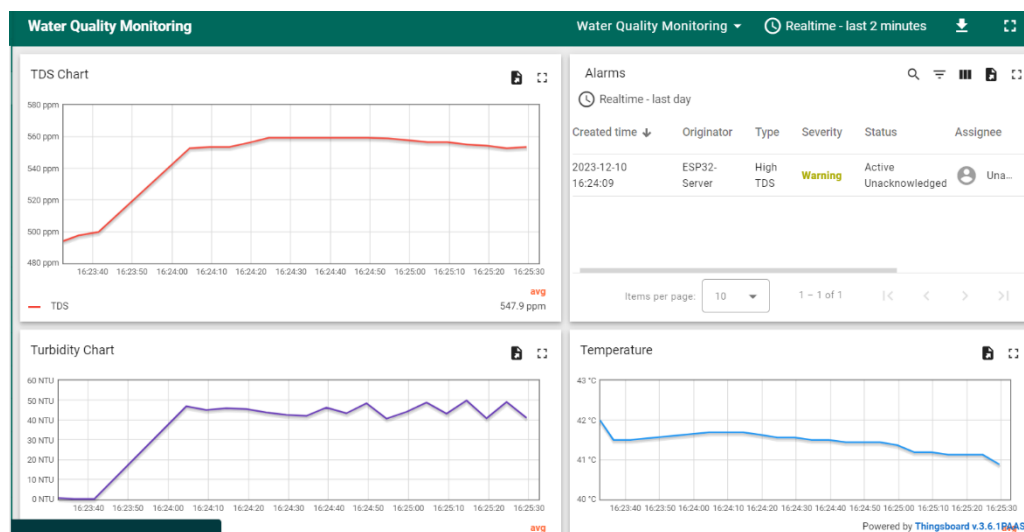

```
Output Serial Monitor x
Message (Enter to send message to 'ESP32 Dev Module' on 'COM15')
New Line 115200 baud

{"temperature":41.56,"tds":559.19,"turbidity":43.78}
Pesan Diterima: {"temperature":41.56,"tds":559.19,"turbidity":43.78}
41.56°C
{"temperature":41.56,"tds":559.19,"turbidity":42.58}
Pesan Diterima: {"temperature":41.56,"tds":559.19,"turbidity":42.58}
41.56°C
41.50°C
{"temperature":41.50,"tds":559.19,"turbidity":42.07}
Pesan Diterima: {"temperature":41.50,"tds":559.19,"turbidity":42.07}
41.50°C
{"temperature":41.50,"tds":559.19,"turbidity":46.26}
41.50°C
Pesan Diterima: {"temperature":41.50,"tds":559.19,"turbidity":46.26}
41.44°C
{"temperature":41.44,"tds":559.19,"turbidity":43.40}
Pesan Diterima: {"temperature":41.44,"tds":559.19,"turbidity":43.40}
41.44°C
41.44°C
{"temperature":41.44,"tds":559.19,"turbidity":40.44}
Pesan Diterima: {"temperature":41.44,"tds":559.19,"turbidity":40.44}
41.44°C
{"temperature":41.44,"tds":558.81,"turbidity":40.69}
Pesan Diterima: {"temperature":41.44,"tds":558.81,"turbidity":40.69}
41.44°C
```

Gambar 18. Serial Monitor ESP32 sensor untuk uji coba kasus kedua

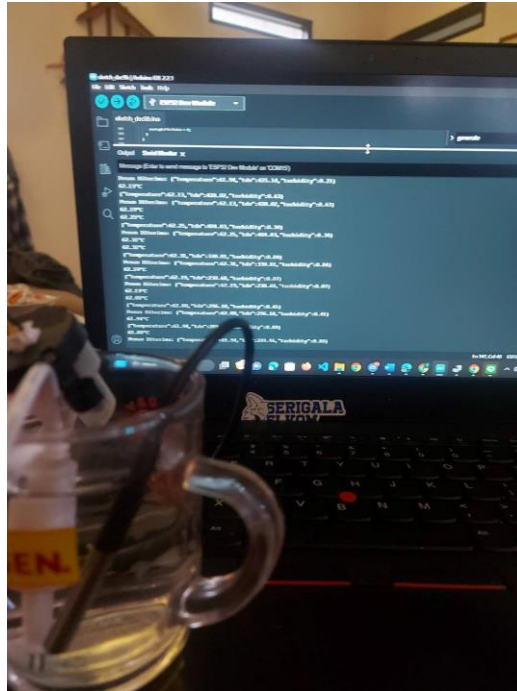
```
16:24:03.661 -> Pesan Diterima: {"temperature":41.69,"tds":552.64,"turbidity":46.90}
16:24:08.681 -> Pesan Diterima: {"temperature":41.69,"tds":553.41,"turbidity":45.05}
16:24:13.694 -> Pesan Diterima: {"temperature":41.69,"tds":553.41,"turbidity":45.91}
16:24:18.703 -> Pesan Diterima: {"temperature":41.63,"tds":556.10,"turbidity":45.55}
16:24:23.840 -> Pesan Diterima: {"temperature":41.56,"tds":559.19,"turbidity":43.78}
16:24:28.755 -> Pesan Diterima: {"temperature":41.56,"tds":559.19,"turbidity":42.58}
16:24:33.656 -> Pesan Diterima: {"temperature":41.50,"tds":559.19,"turbidity":42.07}
16:24:38.689 -> Pesan Diterima: {"temperature":41.50,"tds":559.19,"turbidity":46.26}
16:24:43.668 -> Pesan Diterima: {"temperature":41.44,"tds":559.19,"turbidity":43.40}
16:24:48.728 -> Pesan Diterima: {"temperature":41.44,"tds":559.19,"turbidity":48.44}
16:24:53.845 -> Pesan Diterima: {"temperature":41.44,"tds":558.81,"turbidity":40.69}
16:24:58.748 -> Pesan Diterima: {"temperature":41.37,"tds":557.65,"turbidity":44.10}
```

Gambar 19. Serial Monitor ESP32 server untuk uji coba kasus kedua



Gambar 20. Dashboard Thingsboard untuk uji coba kasus kedua

Kasus ketiga adalah menguji sensor pada air panas.



Gambar 21. Rangkaian untuk uji coba kasus ketiga

```

Output Serial Monitor x
Message (Enter to send message to ESP32 Dev Module on COM15)
Pesan Diterima: {"temperature":62.31,"tds":330.81,"turbidity":0.84}
62.19°C
{"temperature":62.19,"tds":230.68,"turbidity":0.07}
Pesan Diterima: {"temperature":62.19,"tds":230.68,"turbidity":0.07}
62.13°C
62.00°C
{"temperature":62.00,"tds":296.10,"turbidity":0.41}
Pesan Diterima: {"temperature":62.00,"tds":296.10,"turbidity":0.41}
61.94°C
{"temperature":61.94,"tds":209.96,"turbidity":0.88}
61.88°C
Pesan Diterima: {"temperature":61.94,"tds":209.96,"turbidity":0.88}
61.81°C
{"temperature":61.81,"tds":222.85,"turbidity":0.32}
Pesan Diterima: {"temperature":61.81,"tds":222.85,"turbidity":0.32}
61.79°C

```

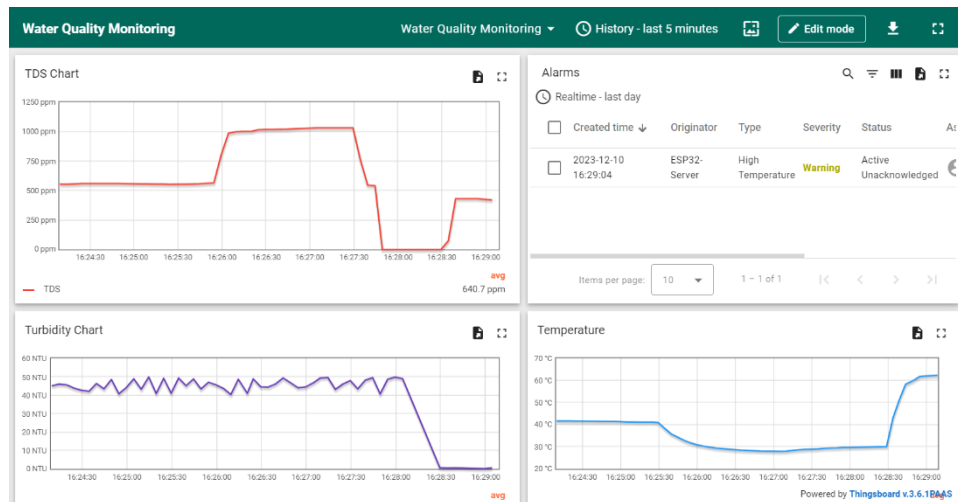
Gambar 22. Serial monitor ESP32 sensor untuk uji coba kasus ketiga

```

16:28:53.253 -> Pesan Diterima: {"temperature":61.69,"tds":430.95,"turbidity":0.26}
16:28:58.164 -> Pesan Diterima: {"temperature":61.94,"tds":425.14,"turbidity":0.21}
16:29:03.332 -> Pesan Diterima: {"temperature":62.13,"tds":420.02,"turbidity":0.63}
16:29:08.191 -> Pesan Diterima: {"temperature":62.25,"tds":404.83,"turbidity":0.34}
16:29:13.311 -> Pesan Diterima: {"temperature":62.31,"tds":330.81,"turbidity":0.84}
16:29:18.209 -> Pesan Diterima: {"temperature":62.19,"tds":230.68,"turbidity":0.07}
16:29:23.243 -> Pesan Diterima: {"temperature":62.00,"tds":296.10,"turbidity":0.41}
16:29:28.173 -> Pesan Diterima: {"temperature":61.94,"tds":209.96,"turbidity":0.88}
16:29:33.184 -> Pesan Diterima: {"temperature":61.81,"tds":222.85,"turbidity":0.32}
16:29:38.298 -> Pesan Diterima: {"temperature":61.69,"tds":272.39,"turbidity":0.98}
16:29:43.217 -> Pesan Diterima: {"temperature":61.69,"tds":242.98,"turbidity":0.09}
16:29:48.229 -> Pesan Diterima: {"temperature":60.50,"tds":308.63,"turbidity":0.91}
16:29:53.164 -> Pesan Diterima: {"temperature":60.06,"tds":312.06,"turbidity":0.10}
16:29:58.268 -> Pesan Diterima: {"temperature":59.94,"tds":320.11,"turbidity":0.92}

```

Gambar 23. Serial monitor ESP32 server untuk uji coba kasus ketiga



Gambar 24. Dashboard Thingsboard untuk uji coba kasus ketiga

3.3 EVALUATION

Untuk tes pertama yang menguji koneksi antar hardware dan software, screenshot yang ditunjukkan menandakan bahwa koneksi ketiganya terjadi dengan sempurna. ESP32 yang membaca sensor mendapatkan pesan dari ESP32 master, sedangkan master terhubung ke MQTT dan ESP32 sensor serta Thingsboard. Platform tersebut mengonfirmasi hal ini dengan screenshot connected devices.

Untuk tes kedua yang menguji pengiriman data dari ESP32 sensor ke Thingsboard, screenshot yang ditunjukkan menandakan bahwa pekerjaan tersebut terjadi dengan sempurna. ESP32 server menerima data dari ESP32 sensor, sedangkan Thingsboard menunjukkan data yang baru masuk pada grafiknya.

Tes ketiga menguji kemampuan sensor. Untuk kasus pertama di mana air bersifat jernih dengan suhu normal, grafik berada di sisi tengah yang menandakan bahwa kondisi air adalah normal. Untuk kasus kedua di mana air bersifat keruh, berisi garam, dan bersuhu normal, grafik TDS dan kekeruhan mengalami peningkatan. Untuk kasus ketiga di mana air bersifat jernih dengan suhu panas, grafik suhu mengalami peningkatan. Ketiga kasus tersebut menunjukkan bahwa sensor berhasil mendeteksi keadaan air dengan akurat.

CHAPTER 4

CONCLUSION AND FUTURE WORK

Air bersih merupakan kebutuhan sentral dalam kehidupan manusia. Jika sumber air menjadi keruh, air tidak dapat digunakan. Oleh karena itu, sistem peringatan dan monitoring air keruh perlu dirancang. Para penyusun mengusulkan sebuah solusi bernama Water Quality Monitoring System (WQMS).

WQMS mengaplikasikan konsep IoT dengan menghubungkannya ke Wi-Fi dan sebuah platform agar para pengguna dapat melakukan monitoring jarak jauh. Perangkat keras yang digunakan melibatkan dua buah mikrokontroler untuk pengiriman ke platform dan mengambil data, serta tiga buah sensor yang masing-masing mengukur TDS, suhu, dan kekeruhan. Kedua mikrokontroler berkomunikasi secara nirkabel dengan ketiga sensor terhubung ke salah satu variabel.

Dengan WQMS, dua tujuan telah tercapai: mengaplikasikan IoT pada kebutuhan sehari-hari dan menyediakan solusi pengukuran kualitas air sederhana. WQMS menjadi solusi bagi mereka yang membutuhkan integrasi IoT dalam memantau kualitas air.

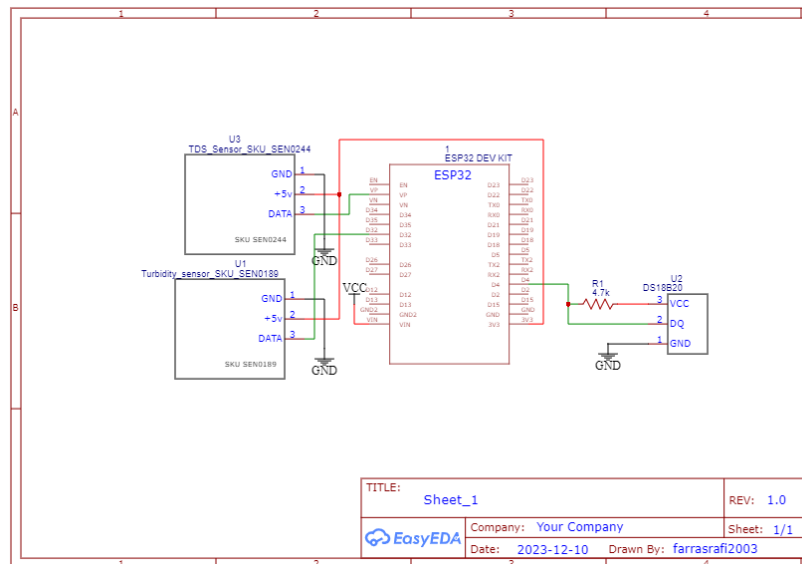
Untuk pengembangan selanjutnya, interface yang dibuat sendiri perlu dirancang untuk mempermudah penambahan komponen kontrol dan pemantauan. Selain itu, dan yang paling utama, sensor yang diintegrasikan ke ESP32 pembaca sensor ditambahkan untuk menambah parameter pengukuran. Peraturan Menteri Kesehatan memberi 19 parameter wajib ditambah dengan parameter khusus untuk air minum.

REFERENCES

- [1] Jakarta Selatan, Jakarta: Kementerian Kesehatan Republik Indonesia, 2023, pp. 29–30.
- [2] "Water Quality Monitoring Device (Prototype) - Projects made with Blynk - Blynk Community," community.blynk.cc. [Online]. Available: <https://community.blynk.cc/t/water-quality-monitoring-device-prototype/53868>
- [3] N. H. Omer, Water Quality Parameters. IntechOpen, 2019. [Online]. Available: <https://www.intechopen.com/chapters/69568>
- [4] "ESP32 with TDS Sensor (Water Quality Sensor) | Random Nerd Tutorials," Apr. 06, 2022. [Online]. Available: <https://randomnerdtutorials.com/esp32-tds-water-quality-sensor/>
- [5] Thingsboard.cloud, 2023. <https://thingsboard.cloud/> (accessed Dec. 10, 2023).

APPENDICES

Appendix A: Project Schematic



Skematik Rangkaian ESP32 Sensor

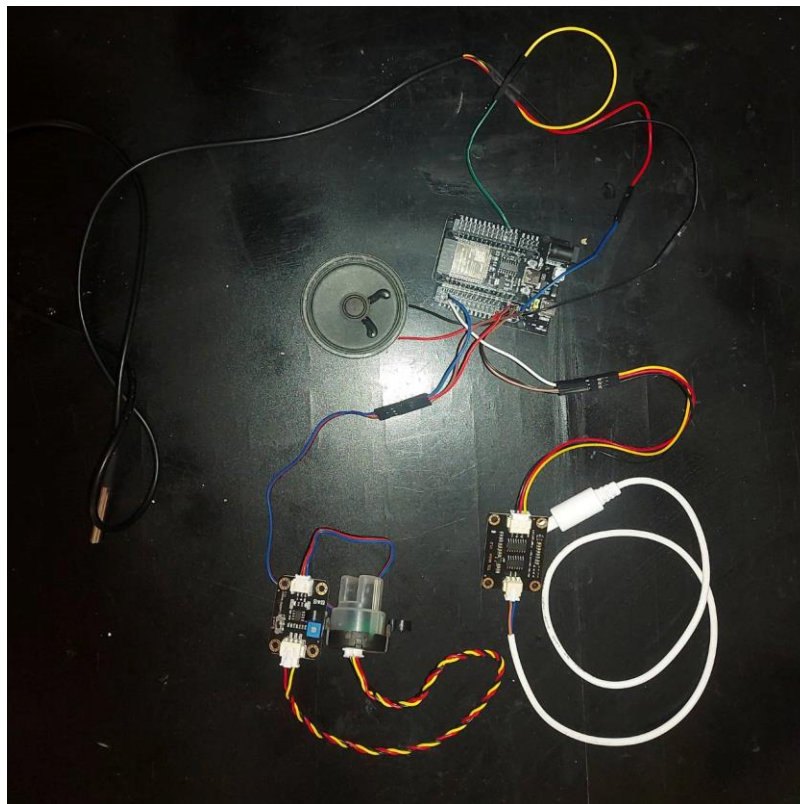


Foto Rangkaian

Appendix B: Documentation

Put the documentation (photos) during the making of the project