JIMU: Jurnal Ilmiah Multidisipliner
Vol. 02, No. 03, Tahun 2024, Hal. 794– 815, ISSN: 3031-9498 (Online)

Protokol REST API pada Sistem Peringatan Dini Bencana Banjir

Habib Maulana Yusuf[1], Nurfiana[2], Dodi Yudo Setyawan[3]
[1,2,3] Institut Informatika Dan Bisnis Darmajaya Bandar Lampung
Email: hmyusuf101@gmail.com

Article Info

Article history:

Received August 15, 2024
Revised August 19, 2024
Accepted August 21, 2024

Keywords:

Flood, IoT, ultrasonic sensor, REST API

ABSTRACT

This research aims to develop an Internet of Things (IoT)-based water level monitoring system using ultrasonic sensors and REST API as an early warning tool against potential flooding. The system detects water levels in real-time and sends data to the server for processing and display. Tests conducted both in the laboratory and in the field, show that this system is able to provide early warnings with a fast response and fairly good accuracy, although there is a margin of error of ± 1 cm. The results of this research show that the system is effective in detecting and providing early warnings of potential flooding, but still requires improving sensor accuracy and adding backup resources to increase system reliability. With further development, this system has the potential to be widely applied in flood risk mitigation.

Article Info

Article history:

Received August 15, 2024
Revised August 19, 2024
Accepted August 21, 2024

Keywords:

Banjir, IoT, Sensor Ultrasonik REST API.

ABSTRACT

Penelitian ini bertujuan untuk mengembangkan sistem monitoring ketinggian air berbasis Internet of Things (IoT) menggunakan sensor ultrasonik dan REST API sebagai alat peringatan dini terhadap potensi banjir. Sistem ini mendeteksi ketinggian air secara real-time dan mengirimkan data ke server untuk diproses dan ditampilkan. Pengujian dilakukan baik di laboratorium maupun di lapangan, menunjukkan bahwa sistem ini mampu memberikan peringatan dini dengan respon cepat dan akurasi yang cukup baik, meskipun terdapat margin error sebesar ±1 cm. Hasil penelitian ini menunjukkan bahwa sistem ini efektif dalam mendeteksi dan memberikan peringatan dini terhadap potensi banjir, namun masih memerlukan peningkatan akurasi sensor dan penambahan sumber daya cadangan untuk meningkatkan keandalan sistem. Dengan pengembangan lebih lanjut, sistem ini memiliki potensi untuk diterapkan secara luas dalam mitigasi risiko bencana banjir.

Corresponding Author:

Nama Penulis: Habib Maulana Yusuf
IIB Darmajaya Bandar Lampung
Email: hmyusuf101@gmail.com

Pendahuluan

Pada era teknologi yang semakin berkembang, sistem monitoring berbasis Internet of Things (IoT) telah banyak diterapkan dalam berbagai bidang, salah satunya untuk sistem peringatan dini bencana banjir. Mengingat Indonesia sebagai negara dengan curah hujan yang tinggi, banjir menjadi salah satu bencana alam yang sering terjadi. Oleh karena itu, diperlukan sebuah sistem yang dapat memantau dan memberikan peringatan dini untuk mengurangi dampak dari bencana banjir tersebut. Dalam penelitian ini, sistem monitoring ketinggian air menggunakan sensor ultrasonik dan REST API diimplementasikan sebagai solusi untuk memberikan peringatan dini terhadap potensi banjir.

Badan Penanggulangan Bencana Daerah (BPBD) Lampung mendata sepanjang bulan Februari 2024 terdapat 16 kejadian bencana. Di mana, dari jumlah tersebut bencana banjir mendominasi. Kepala Pelaksana BPBD Lampung, Rudy Sjawal Sugiarto mengatakan, sepanjang bulan Februari 2024 ini kejadian bencana yang ada di Lampung juga menyebabkan ratusan bangunan mengalami kerusakan. "Telah terjadi 16 kejadian bencana dalam satu bulan, tidak ada korban jiwa. Bencana juga mengakibatkan 235 unit bangunan terdampak bencana. BPBD Lampung mencatat sepanjang Februari 2024 bencana banjir terjadi sebanyak 10 kejadian yang tersebar di Bandar Lampung, Lampung Selatan, Pesawaran dan Tulang Bawang. Sementara dari 16 kejadian bencana yang terjadi selama bulan Februari 2024 juga mengakibatkan kerugian uang sekitar 155 juta rupiah (Galih Prihantoro, 2024).

Sistem peringatan dini bencana banjir bertujuan untuk memberikan informasi yang cepat dan akurat mengenai potensi terjadinya banjir kepada masyarakat dan pihak berwenang. Informasi ini mencakup data curah hujan, ketinggian air sungai, prediksi cuaca, dan faktor-faktor lain yang berpotensi menyebabkan banjir. Salah satu tantangan utama dalam pengembangan sistem ini adalah integrasi dan distribusi data dari berbagai sumber secara real-time penelitan system monitoing bencana banjir sudah pernah dilakukan oleh beberapa peneliti diantaranya yaitu

(Ulum, 2023) Sistem Monitoring Cuaca Dan Peringatan Banjir Berbasis IOT Dengan Menggunakan Aplikasi Mit App Inventor. menggunakan Aplikasi Mit App Inventor yang dimana diketahuin jika protokol ini cenderung memiliki kinerja yang lebih rendah sehhingga dapat mempengaruhi kecepatan dan efisiensi komunikasi protokol, terutama jika aplikasi harus menangani data dalam jumlah besar atau komunikasi real-time.

(Bustomi, 2023) Rancang bangun sistem monitoring peringatan dini bencana banjir berbasis IOT menggunakan protokol MQTT dengan notifikasi bot telegram. Menggunakan protocol MQTT namun protokol MQTT memiliki kelemahan, seperti kompleksitas dalam implementasi dan kebutuhan untuk infrastruktur broker MQTT yang handal, yang dapat menambah biaya dan upaya pengelolaan.

Sistem monitoring bencana banjir menggunakan protocol REST API. REST API adalah arsitektur yang memungkinkan pertukaran data antar sistem secara sederhana dan cepat melalui HTTP. (Muhammad Kevin Naufal Faza Affrianto, 2023) Dengan menggunakan REST API, berbagai komponen sistem peringatan dini, seperti sensor cuaca, server prediksi, dan aplikasi notifikasi, dapat saling berkomunikasi dan bertukar data dengan efisien. REST API juga mendukung interoperabilitas, memungkinkan berbagai pihak seperti pemerintah daerah, lembaga penanggulangan bencana, dan masyarakat umum untuk mengakses data dan informasi peringatan dengan mudah. Penelitian Fish Feeding System in Aquaponic Media Using Firebase and Internet of Things Based Codular, masih berkaitan dengan perancangan perangkat lunak dan perancangan data sensor yang dibaca oleh mikrokontroler akan dikirimkan ke Firebase real-

time database kemudian diolah menggunakan REST API untuk ditampilkan oleh aplikasi.(Santoso & Rosmalia, 2023).

Penelitian dengan judul Earthquake Early Warning System Real Time Design Using Total Electron Content and Geomagnetism with Fuzzy Logic. Perancangan sistem peringatan dini gempa bumi ini memanfaatkan data real-time geomagnetisme yang diperoleh dari sensor magnetometer fluxgate dan Total Electron Content (TEC) pada lapisan ionosfer, yang diukur melalui pemancar dan penerima gelombang FM. Data yang dikumpulkan dari kedua tipe sensor ini diproses menggunakan logika fuzzy untuk menentukan hasil pengukuran. Informasi yang dihasilkan meliputi prediksi kekuatan gempa bumi yang akan terjadi, serta waktu dan lokasi pusat gempa. Data ini kemudian dikirimkan kepada pengguna melalui Layanan Pesan Singkat (SMS) ke nomor telepon yang telah didaftarkan. (Dodi Yudo Setyawan, 2014).

Penelitian ini memiliki sistem otomasi yang digunakan untuk Bencana tanah longsor. Merupakan peristiwa gerakan tanah atau batuan yang terjadi dalam jumlah massa tertentu dan dapat memiliki berbagai tipe serta jenis, termasuk longsoran translasi, longsor rotasi, dan rayapan tanah. Tanah longsor terjadi ketika gaya pendorong pada lereng melebihi gaya penahan, sering mengakibatkan banyak korban karena sifatnya yang tiba-tiba. Untuk mengatasi masalah ini, diperlukan alat peringatan dini yang efektif. Salah satu solusi teknologi yang dapat digunakan adalah mikrokontroler, yang terdiri dari dua bagian utama: sensor potensiometer dan LDR (Light Dependent Resistor) untuk mendeteksi pergeseran tanah, serta modem Wavecom yang berfungsi sebagai sistem peringatan dini. Sistem ini dirancang untuk memberikan informasi awal guna mencegah dampak bencana tanah longsor yang tiba-tiba dan merugikan.(Sudibyo & Ridho, 2015).

Untuk membangun sistem peringatan dini gempa bumi, metode telemetri berbasis IoT diterapkan untuk mengukur flux magnet bumi dan getaran pada koordinat lintang dan bujur tertentu (-5.640274, 104.3050093) dan (-5.600941, 104.7788183) menggunakan sensor MAG3110 dan ADXL345, yang diintegrasikan dalam Raspberry Pi 3 sebagai server. Aplikasi di handphone Android digunakan untuk menampilkan data pengukuran. Hasil pengukuran menunjukkan bahwa rata-rata waktu yang dibutuhkan untuk dampak timbulnya anomali getaran pada koordinat yang sama adalah 400 detik, dengan rentang anomali flux magnet antara 395 μT hingga 404 μT. Temuan ini memberikan dasar untuk pengembangan sistem peringatan gempa bumi secara real-time. (D. Y. Setyawan et al., 2021).

Implementasi REST API dalam sistem peringatan dini bencana banjir memiliki beberapa keunggulan. Pertama, fleksibilitas dan skalabilitasnya memungkinkan penambahan atau pengurangan komponen sistem tanpa mengganggu keseluruhan sistem. Kedua, kemudahan integrasi dengan berbagai platform dan perangkat, termasuk aplikasi mobile dan web, memudahkan distribusi informasi kepada pengguna akhir. Ketiga, protokol ini mendukung format data yang umum digunakan seperti JSON dan XML, sehingga mempermudah proses pengolahan dan analisis data (Kaniya Pradnya Paramitha et al., 2022).

Pengembangan dan implementasi REST API dalam sistem peringatan dini bencana banjir di Bandar Lampung menjadi langkah yang strategis untuk meningkatkan efektivitas penanggulangan bencana banjir di kota ini. Hal ini diharapkan dapat memberikan kontribusi signifikan dalam mengurangi risiko dan dampak yang ditimbulkan oleh bencana banjir, serta meningkatkan kesiapsiagaan dan respons masyarakat terhadap ancaman banjir.

Dasar Teori

a**.** *Internet of Things (IoT)*

Internet of Things (IoT) adalah konsep yang menggambarkan jaringan perangkat fisik yang tertanam dengan teknologi seperti sensor, perangkat lunak, dan konektivitas untuk bertukar data dengan perangkat dan sistem lainnya melalui internet. IoT memungkinkan objek-objek ini untuk mengumpulkan dan berbagi data, menciptakan lebih banyak peluang untuk integrasi sistem fisik dengan dunia digital. Hal ini memungkinkan pengambilan keputusan yang lebih baik dan tindakan otomatis berdasarkan data real-time (Windiastik et al., 2019).

b**.** Protokol IoT

Protokol IoT adalah seperangkat aturan dan standar yang memungkinkan perangkat IoT untuk berkomunikasi dan bertukar data satu sama lain melalui jaringan. Protokol ini memastikan bahwa perangkat yang berbeda dari berbagai produsen dapat bekerja sama secara efisien. Ada beberapa protokol yang umum digunakan dalam ekosistem IoT, masing-masing dengan kelebihan dan kekurangan tertentu. Beberapa protokol IoT yang populer antara lain:

- MQTT (Message Queuing Telemetry Transport)
- *CoAP (Constrained Application Protocol)*
- *AMQP (Advanced Message Queuing Protocol)*
- REST API

c. REST API

REST (Representational State Transfer) adalah sebuah arsitektur yang digunakan untuk mengembangkan layanan web. REST API (Application Programming Interface) adalah antarmuka yang memungkinkan berbagai aplikasi berkomunikasi satu sama lain melalui protokol HTTP. REST dirancang untuk bekerja dengan sumber daya yang dapat diidentifikasi oleh URI (Ibarreche et al., 2020).

REST API memungkinkan sistem untuk mengirim dan menerima data dalam format tertentu (biasanya JSON atau XML) melalui metode HTTP standar seperti GET, POST, PUT, DELETE, dan PATCH. Pendekatan ini memfasilitasi interaksi antara berbagai aplikasi dan layanan dengan cara yang sederhana dan efisien (Choirudin & Adil, 2019).

- Prinsip-Prinsip Dasar REST

REST memiliki beberapa prinsip dasar yang membuatnya menjadi pilihan populer untuk mengembangkan API:

• Stateless: Setiap permintaan dari klien ke server harus berisi semua informasi yang diperlukan untuk memahami dan memproses permintaan tersebut. Server tidak menyimpan konteks antara permintaan.

• Client-Server: Pemisahan antara klien dan server meningkatkan portabilitas antarmuka pengguna di berbagai platform dan meningkatkan skalabilitas server.

• Cacheable: Respons dari server harus menyatakan apakah mereka dapat di-cache atau tidak untuk meningkatkan kinerja klien dengan menghindari pengambilan data yang sama berulang kali.

• Uniform Interface: Antarmuka yang konsisten dan seragam memungkinkan interaksi yang mudah dan efisien antara klien dan server.

• Layered System: Arsitektur dapat diatur dalam lapisan, dimana setiap lapisan hanya tahu tentang interaksi dengan lapisan di sebelahnya.

• Code on Demand (opsional): Server dapat memperluas fungsionalitas klien secara sementara dengan mengirimkan kode eksekusi (misalnya, applet Java atau skrip JavaScript).

d. Sensor Ultrasonik HC-SR04

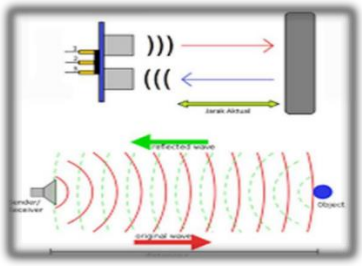
Modul Ultrasonic HC-SR04 dapat digunakan untuk mengukur jarak yang menggunakan gelombang ultrasonik adalah pemancar (Transmitter) mengirimkan gelombang ultrasonik dan jika di pantulkan oleh suatu benda di depannya. Lalu akan diterima oleh penerima (Receiver). Pengukuran pada sensor ultrasonik ini adalah dengan cara panjang gelombang yang dipantulkan sampai diterima dan dibagi 2 dan Dikalikan 0.0034. (Andalanelektro.id, 2018)

Waktu tempuh = waktu tempuh gelombang dari transmitter menuju reciver /2 Jarak = waktu tempu * 343.2

Gambar 1. Sensor Ultrasonic (Sumber sensor pengukur jarak)

Spesifikasi dari sensor ultrasonik HY-SRF05 sebagai berikut:

1. Bekerja pada tegangan DC 5 Volt
2. Beban arus sebesar 30 mA – 50 mA
3. Menghasilkan gelombang dengan frekuensi 40KHz
4. Jangkauan jarak yang dapat dideteksi 3cm

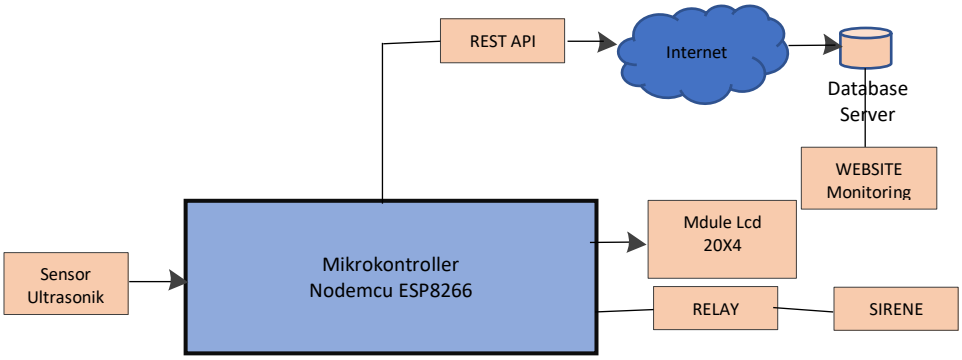


Gambar 2. Cara Kerja Modul Ultrasonik (Sumber sensor pengukur jarak)

Metode

Perancangan sistem merupakan suatu hal yang dilakukan untuk mempermudah proses pembuatan alat. Konsep Implementasi protokol REST API pada sistem peringatan dini bencana

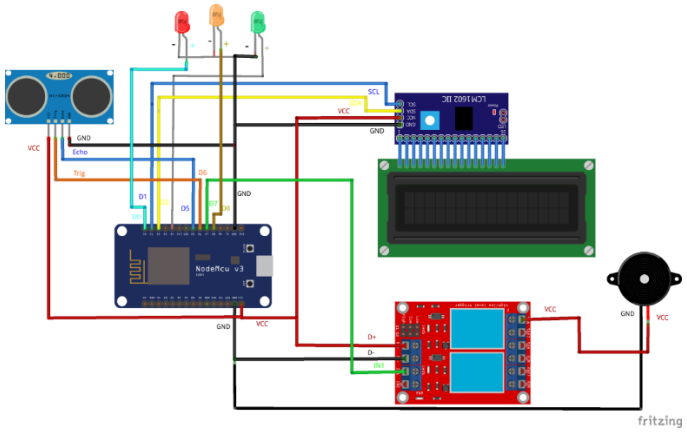
banjir digambarkan pada diagram blok dapat dilihat pada gambar 3 Blok diagram menjelaskan gambaran umum mengenai cara kerja dari sistem monitoring peringatan dini bencana banjir yang akan dibuat.



Gambar 3. Blok Diagram Sistem

Dari gambar 3, blok diagram sistem dapat diketahui sistem kerja dari alat yaitu inputan dari Sensor ultrsaonik memiliki 3 Level yaitu Level yaitu Level 1, Level 2 dan Level 3 yang akan diproes oleh nodemcu, jika sensor dalam Level 1 (Aman) maka sirine tidak akan berbunyi dan akan mengirimkan notifikasi (Sungai dalam kondisi Waspada), jika sensor dalam Level 2 (Siaga) maka sirine akan berbunyi 2 kali dan akan mengirimkan notifikasi (Sungai dalam kondisi Siaga). Sedangakan jika sensor dalam Level 3 (Bahaya) maka sirine berbunyi akan mengirimkan notifikasi (Sungai dalam kondis bahaya). Hasil dari pembacaan sensor akan ditampikan pada LCD 20x4 dan aplikasi.

Rangkaian keseluruhan merupakan tahap terakhir dari perancangan yang telah dilakukan. Dalam tahap ini seluruh komponen dipasang sesuai dengan sistem yang telah dibuat, Adapun rangkaian keseluruhan dapat dilihat pada gambar 4 berikut ini.

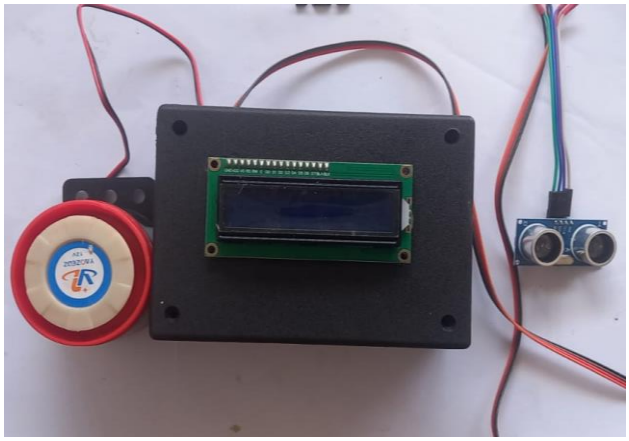


Gambar 4. Rangkaian Keseluruhan

Hasil

Uji coba dilakukan untuk memastikan rangkaian yang dihasilkan mampu bekerja sesuai dengan yang diharapkan. maka terlebih dahulu dilakukan langkah pengujian dan mengamati langsung rangkaian serta komponen. Hasil pengukuran ini dapat diketahui rangkaian telah

bekerja dengan baik atau tidak, sehingga apabila terdapat kesalahan dan kekurangan akan terdeteksi. Gambar 5 berikut ini merupakan gambar dari bentuk fisik alat yang telah dibuat.



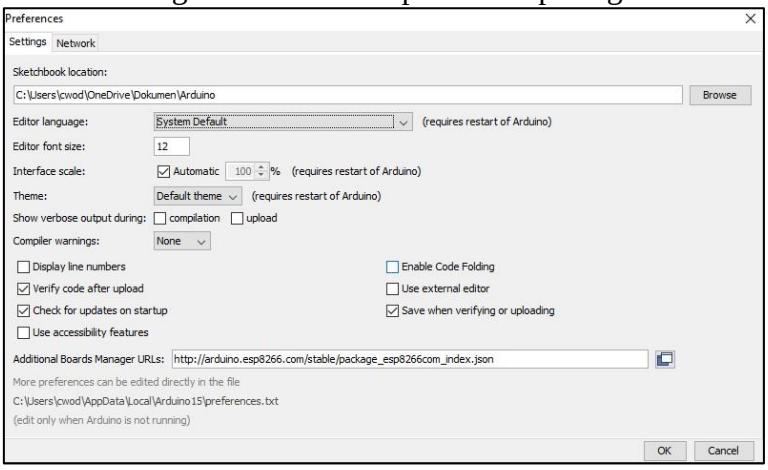
Gambar 5 Bentuk Fisik Alat

Pada pengujian ini meliputi pungujian sensor ultrasonik, web dan pengujian sistem keseluruhan. Pengujian ini dilakukan agar peneliti dapat mengetahuin kelebihan dan kekurangan sistem yang telah di buat hasil pengujian sebagai berikut:

Implementasi Arduino

1. Preferences Setting Arduino IDE

Preferences digunakan untuk menentukan isi menu Board terdapat di boards.txt di hardware atau sub-direktori dari direktori aplikasi Arduino. Definisi untuk menu Burn Bootloader ada di file programmers.txt di direktori yang sama. Untuk membuat papan baru atau definisi pemrograman, salin link yang sudah ada, ubah awalan yang digunakan dalam tombol preferensi (contoh “diecimilia. atau “avrisp.”) dan ubah nilainya agar sesuai dengan perangkat yang akan digunakan. Untuk additional boards manager URLs diisikan untuk mencari board dari komponen nodeMCU ESP8266 dikarenakan board belum muncul di aplikasi. Oleh karena itu, harus dilakukan pencarian board terlebih dahulu agar dapat melakukan proses pemrograman selanjutnya. Preferences Setting Arduino IDE dapat dilihat pada gambar 6.

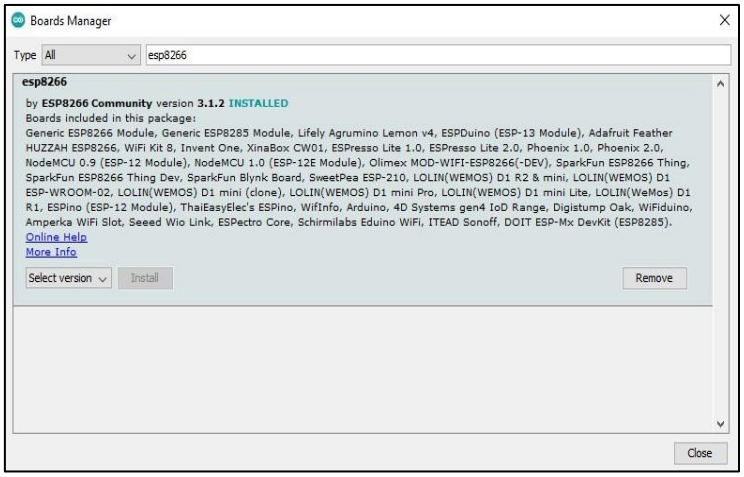


Gambar 6 Preferences Setting Arduino IDE

2. Board Manager

Pada saat preferences sudah diisi maka selanjutnya melakukan pencarian board manager di tools yang ada di Arduino IDE. Untuk type pilih All, dan pada kolom kosong sebelahnya diisikan ESP8266, jika sudah diisi maka otomatis board yang di download melalui link di

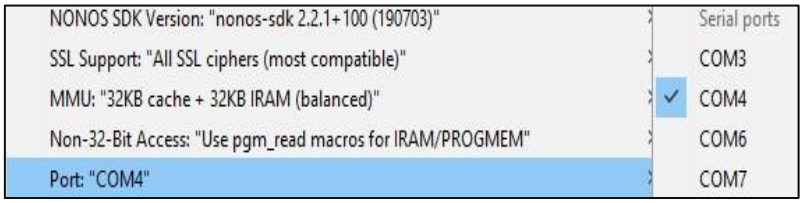
preferences akan tersedia dan dapat digunakan. Instal Board Manager pada Arduino IDE dapat diliat pada gambar 7 dibawah ini.



Gambar 7 Instal Board Manager

2. Port Arduino IDE

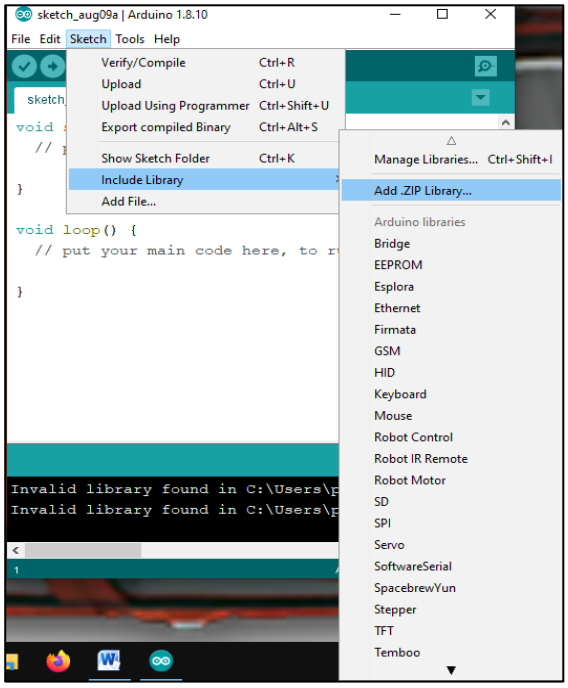
Port digunakan untuk mengkomplikasi sketsa dan protocol yang digunakan saat mengunggah sketsa. Pemilihan port mengacu pada papan yang terhubung pada komputer. Port yang dikenali oleh Arduino IDE akan secara otomatis dikenali dan ditampilkan di menu. Berikut penggunaan Port Arduino Uno pada gambar 8.



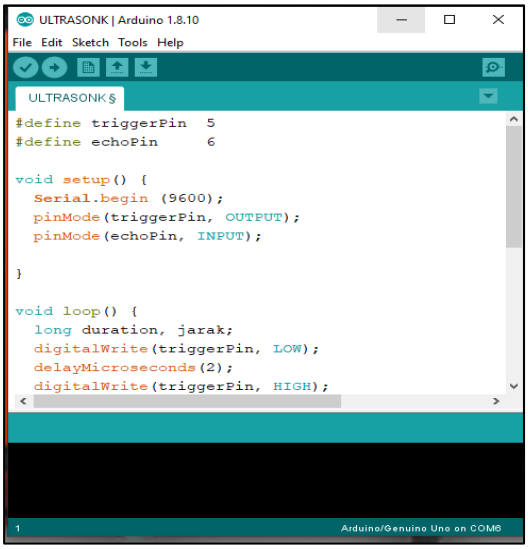
Gambar 8 Port Arduino IDE

3. Include Library

Berguna untuk menambahkan fungsionalitas ekstra ke dalam Modul Arduino. Terdapat daftar pustaka yang dapat ditambahkan dengan menekan tombol sketsa. Untuk menambahkan fungsionalitas ekstra ke dalam modul Arduino, buka Arduino IDE dan akses menu `Sketch`. Dari submenu `Include Library`, pilih `Manage Libraries` untuk membuka Library Manager, di mana Anda dapat mencari dan menginstal pustaka baru dari repositori Arduino dengan mengklik tombol `Install` pada pustaka yang diinginkan. Alternatifnya, jika Anda sudah memiliki pustaka dalam format .ZIP, pilih `Add .ZIP Library...` dan arahkan ke file tersebut untuk menambahkannya langsung. Setelah pustaka diinstal, sertakan pustaka dalam sketsa Anda dengan memilih `Include Library` dari menu yang sama dan memilih pustaka yang telah diinstal, yang akan menambahkan baris `#include <NamaPustaka.h>` di bagian atas sketsa. Anda dapat menggunakan fungsionalitas pustaka sesuai dokumentasi atau contoh kode yang disediakan. Verifikasi sketsa dengan menekan `Verify`, dan jika tidak ada kesalahan, unggah ke papan Arduino dengan menekan `Upload`. Penjelasan berikut terdapat pada gambar 9.

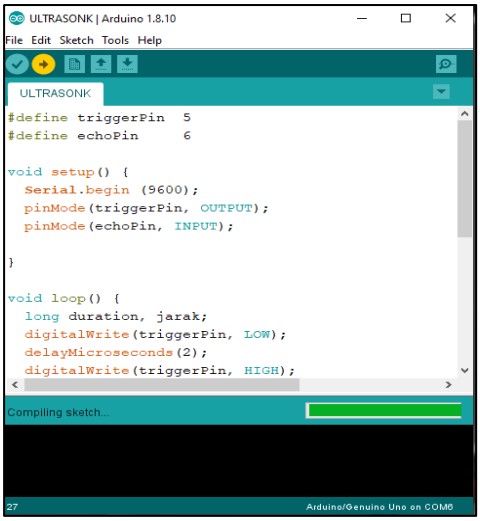


Gambar 9 Include Library.



Gambar 10 Program Yang Telah Dibuat

Kemudian setelah program selesai dibuat. Pada gambar 10, langkah selanjutnya adalah mengunggah program tersebut ke dalam perangkat Arduino. Proses ini penting agar Arduino dapat menjalankan instruksi yang telah diprogram untuk mengoperasikan sistem otomatis yang dirancang. Program yang telah diunggah akan berfungsi. Pengunggahan program dilakukan melalui perangkat lunak Arduino IDE. Setelah program berhasil diunggah, Arduino akan secara otomatis menjalankan program tersebut setiap kali perangkat dinyalakan. Proses upload program dapat terlihat pada gambar 11 dibawah ini.



Gambar 11 Proses Upload Program Ke NodeMCU

1. Hasil Pengujian REST API

Pengujian bertujuan untuk menguji REST API yang dirancang khusus untuk memonitoring ketinggian air secara real-time menggunakan sensor ultrasonik. Sensor Utrasonik akan mengukur jarak antara permukaan air dan sensor, kemudian data tersebut dikirimkan melalui REST API ke sistem pemantauan yang terintegrasi. Melalui pengujian ini, stabilitas dan keandalan API akan diuji, termasuk bagaimana API menangani berbagai kondisi jaringan, volume data, serta respon terhadap perubahan cepat dalam ketinggian air. Pengujian juga mencakup evaluasi terhadap kecepatan respon dan akurasi data yang dikirimkan oleh sensor ultrasonik. Tabel 1 di bawah ini memberikan rincian lebih lanjut mengenai berbagai aspek yang diuji selama proses pengujian REST API untuk memonitoring ketinggian air.

Tabel 1 Pengujian REST API

No	Endpoint	Metode HTTP	Input Data	Expected Output	Actual Output	Status	Keterangan
1	/api/waterlevel	GET	N/A	Daftar semua data ketinggian air	Daftar semua data ketinggian air	Pass	Berhasil mendapatkan semua data ketinggian air
2	/api/waterlevel/latest	GET	N/A	Data ketinggian air terbaru	Data ketinggian air terbaru	Pass	Data terbaru sesuai
3	/api/waterlevel	POST	{"height": 150, "timestamp": "2024-08-01T12:00:00Z"}	Data ketinggian air baru	Data ketinggian air baru	Pass	Data baru berhasil disimpan
4	/api/waterlevel/1	GET	N/A	Data ketinggian air dengan ID 1	Data ketinggian air dengan ID 1	Pass	Data ketinggian air sesuai
5	/api/waterlevel/1	PUT	{"height": 160, "timestamp": "2024-08-01T12:00:00Z"}	Data ketinggian air dengan ID 1 yang diperbarui	Data ketinggian air dengan ID 1 yang diperbarui	Pass	Data berhasil diperbarui

6	/api/waterlevel	POST	{"height": -10, "timestamp": "2024-08-01T12:00:00Z"}	Error: Data tidak valid	Error: Data tidak valid	Pass	Validasi input berfungsi
7	/api/waterlevel/100	GET	N/A	Error: Data tidak ditemukan	Error: Data tidak ditemukan	Pass	Data tidak ada
8	/api/waterlevel	GET	N/A	Daftar semua data ketinggian air	Error: Server tidak tersedia	Fail	Server down
9	/api/waterlevel	OPTIONS	N/A	Metode yang didukung oleh endpoint	Metode yang didukung oleh endpoint	Pass	Berhasil mendapatkan metode yang didukung

Tabel diatas ini menjelaskan secara rinci mengenai komponen-komponen penting yang diperhatikan selama proses pengujian REST API, terutama ketika menguji endpoint tertentu. Berikut penjelasan lebih rinci tentang setiap komponen yang terlibat:

1. Endpoint : Ini adalah URL spesifik yang menjadi target pengujian. Setiap endpoint pada API mewakili fungsi tertentu, seperti mendapatkan data ketinggian air dari sensor atau mengirimkan data pengukuran. Dalam pengujian, setiap endpoint diakses untuk memastikan bahwa ia merespon dengan benar sesuai dengan fungsinya.

2. Metode HTTP : Yang digunakan untuk mengakses endpoint. HTTP menyediakan berbagai metode untuk berinteraksi dengan server, dan dalam pengujian REST API, metode yang paling umum digunakan adalah:

- GET : Untuk mengambil data dari server, seperti mendapatkan data ketinggian air saat ini.
- POST : Untuk mengirimkan data ke server, misalnya, mengirimkan hasil pengukuran terbaru dari sensor.
- PUT : Untuk memperbarui data yang ada di server, seperti memperbarui konfigurasi sensor.
- DELETE : Untuk menghapus data di server jika diperlukan.

Pengujian melibatkan memastikan bahwa setiap metode HTTP berfungsi sesuai dengan spesifikasinya untuk setiap endpoint yang diuji.

3. Input Data : Ini merujuk pada data yang dikirimkan dalam permintaan HTTP ke API, jika diperlukan. Misalnya, saat menggunakan metode POST atau PUT, data seperti hasil pengukuran sensor atau konfigurasi tambahan mungkin harus dikirimkan ke server. Pengujian ini memastikan bahwa data input yangp diberikan diproses dengan benar oleh API.

4. Expected Output : Ini adalah keluaran yang diharapkan dari API berdasarkan spesifikasi atau dokumentasi API. Misalnya, jika permintaan GET dikirimkan ke endpoint untuk mendapatkan ketinggian air, expected output-nya mungkin berupa data ketinggian air dalam format JSON. Expected output harus ditentukan sebelumnya berdasarkan kebutuhan sistem dan spesifikasi API.

5. Actual Output : Actual output adalah respons nyata yang diterima dari API setelah permintaan dikirim. Ini dibandingkan dengan expected output untuk menentukan apakah API berfungsi sebagaimana mestinya. Misalnya, jika expected output adalah data ketinggian air

dalam format tertentu, actual output yang diterima dari server harus sesuai dengan format dan nilai yang diharapkan.

6. Status: Ini adalah hasil pengujian yang menunjukkan apakah output aktual sesuai dengan output yang diharapkan. Hasil pengujian diberi status **Pass** jika actual output sesuai dengan expected output, menunjukkan bahwa API berfungsi dengan benar. Jika actual output tidak sesuai, statusnya adalah **Fail**, menunjukkan bahwa ada masalah dalam API yang memerlukan perbaikan.

7. Keterangan : Bagian ini mencakup informasi tambahan yang relevan tentang hasil pengujian. Misalnya, jika pengujian gagal, keterangan dapat berisi detail kesalahan, seperti pesan error yang dikembalikan oleh server, atau kondisi lingkungan yang mungkin mempengaruhi hasil pengujian, seperti masalah jaringan atau server yang tidak tersedia. Keterangan ini penting untuk analisis lebih lanjut dan membantu dalam proses debugging.

Berdasarkan tabel diatas digunakan untuk mengorganisir dan melacak pengujian REST API yang memonitoring ketinggian air menggunakan sensor ultrasonik. Pengujian yang berhasil memastikan bahwa API dapat diandalkan untuk memantau ketinggian air dengan akurat. Pengujian yang gagal membantu mengidentifikasi masalah yang perlu diperbaiki untuk memastikan API berfungsi dengan baik.

2. Hasil Pengujian REST API Update Ketinggian

Pada monitoring banjir, terdapat beberapa endpoint yang digunakan untuk berbagai tujuan dalam sistem, masing-masing dengan metode HTTP tertentu yang diperlukan untuk mengakses atau memanipulasi data yang berkaitan dengan monitoring banjir. Daftar lengkap dari endpoint-endpoint tersebut, beserta dengan metode HTTP yang digunakan untuk setiap operasi, ditampilkan secara rinci pada Tabel 2 di bawah ini.

Tabel 2 Daftar Endpoint Modul Data Monitoing Banjir

Method	***Endpoint***
updateKetinggian	/ganti
getStatus	/status

Method update Ketinggian digunakan untuk menampilkan keseluruhan data ketinggian air yang diproleh dari hasil pembacaan sensor ultrasonic. Kode program merupakan kode program pada resource class yang digunakan pada method updateKetinggian. Method updateKetinggian menggunakan HTTP method GET dengan endpoint yaitu API//ganti. Adapun gambar dari kode program resource update ketinggian adalah seperti gambar 12.

```
app.get('/ganti',(req, res)=>{
  const nilaiStr = req.query.nilai;

    // Check if nilai is provided
    if (nilaiStr === undefined) {
        return res.status(400).send('Nilai tidak boleh kosong');
    }

    // Convert nilai to an integer
    const nilai = parseInt(nilaiStr, 10);

    // Check if conversion was successful
    if (isNaN(nilai)) {
        return res.status(400).send('Nilai parameter must be an integer');
    }
  const query = 'UPDATE status SET ketinggian = ?';
  db.query(query,[nilai],(error,result)=>{
    if(error){
      return res.status(500).send('Database error');
    }

    res.send(`
    <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Success Message</title>
    <style>
```

Gambar 12 Kode Program Resource Update Ketinggian

Method getStatus digunakan untuk hasil status ketinggian air sungai yang diproleh dari hasil pembacaan sensor ultrasonic. Kode program merupakan kode program pada resource class yang digunakan pada method getStatus. Method getStatus menggunakan HTTP method GET dengan endpoint yaitu api/ status. Kode program resource update status dapat dilihat pada 13.

```
app.get('/status', (req, res) => {
  db.query('SELECT ketinggian FROM status', (err, results) => {
    if (err) return res.status(500).send('Database query error.');

    var hasil;

    const ketinggian = results[0].ketinggian;
    if(ketinggian <= aman){
      hasil = 'Aman';
    }else if(ketinggian > aman && ketinggian <= siaga){
      hasil = 'Siaga';
    }else if(ketinggian > siaga){
      hasil = 'Bahaya';
    }

    res.send(`
      <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0
    <title>Status</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f9f9f9;
            display: flex;
```

Gambar 13 Kode Program Resource Update Status

3. Hasil Pengujian REST API Update Ketinggian

Pengujian dilakukan dalam bentuk functional testing terhadap hasil yang didapatkan dengan menggunakan metode blackbox testing untuk memastikan fitur–fitur yang dirancang dapat berjalan sesuai dengan yang diharapkan. Pengujian REST API dilakukan dengan menggunakan sensor ultrasonic. Berikut adalah pengujian REST API pada fitur Update Ketinggian, yang mencakup proses mendapatkan dan mengubah ketinggian:

1. Mendapatkan Ketinggian

Hal ini ialah fungsi utama dari endpoint, yaitu untuk mengambil data ketinggian saat ini yang diukur oleh sistem monitoring.

Endpoint:

GET https://pentagonal-crawling-shovel.glitch.me/status

Bagian ini memperkenalkan URL endpoint, yaitu alamat spesifik pada server tempat permintaan data dikirimkan. Endpoint ini berfungsi sebagai pintu akses ke data ketinggian air. Ini adalah URL lengkap dan metode HTTP (GET) yang digunakan untuk mengakses endpoint tersebut. Dalam kasus ini, metode GET digunakan untuk mengambil atau membaca data, bukan untuk mengirim atau mengubah data.

Deskripsi:

Endpoint ini berfungsi untuk mengambil data terbaru mengenai ketinggian air yang diukur oleh sistem. Setiap kali permintaan dikirim ke endpoint ini, server akan merespons dengan nilai ketinggian air saat ini yang diperoleh dari sensor, memungkinkan pengguna atau sistem untuk memantau kondisi air secara real-time.

Permintaan:

- **Metode:** GET

Ini menunjukkan bahwa metode HTTP yang digunakan untuk meminta data dari server adalah metode GET. Metode GET biasanya digunakan untuk mengambil data dari server tanpa mengubah status atau konten server tersebut.

Contoh Permintaan:

http
GET https://pentagonal-crawling-shovel.glitch.me/status

Ini adalah contoh URL yang digunakan untuk membuat permintaan GET. URL ini adalah endpoint API yang akan dikunjungi untuk mengambil data.

Respon:

- **Status:** aman OK
  Menunjukkan bahwa status dari respon adalah "aman OK", yang berarti permintaan berhasil diproses dengan baik oleh server.
- **Body:**
  Menggambarkan struktur data JSON yang akan dikembalikan oleh server sebagai respons dari permintaan GET tersebut. JSON (JavaScript Object Notation) adalah format data yang umum digunakan untuk bertukar data antara server dan klien.

Json

{
"ketinggian": <nilai_ketinggian>
}

Contoh Respon:

Ini adalah contoh respons dalam format JSON yang menunjukkan bahwa nilai dari parameter "ketinggian" yang dikembalikan oleh server adalah 30.

Json

{
"ketinggian": 30
}

Gambar 14 Hasil Ujicoba REST API Mendapatkan Nilai Ketinggian

Gambar 14 ini menampilkan informasi mengenai "Ketinggian". Pada contoh yang ditampilkan, ketinggian air adalah 30 (kemungkinan dalam satuan cm atau meter). Dan menampilkan "Status" kondisi berdasarkan ketinggian air, statusnya adalah Aman, yang berarti ketinggian air masih dalam batas aman dan belum membahayakan.

2. Mengubah Ketinggian

Hal ini ialah yang menunjukkan bahwa menjelaskan cara untuk mengubah nilai ketinggian menggunakan endpoint API yang disediakan.

Endpoint:
POST https://pentagonal-crawling-shovel.glitch.me/status
Ini adalah URL endpoint yang digunakan untuk melakukan permintaan POST. Dalam konteks API, endpoint ini digunakan untuk mengubah atau memperbarui data (dalam hal ini, nilai ketinggian).

Deskripsi:

Endpoint ini digunakan untuk memperbarui atau mengubah data ketinggian yang tersimpan di server. Dengan menggunakan endpoint ini, pengguna dapat mengirimkan nilai ketinggian yang baru melalui permintaan HTTP-POST. Nilai ketinggian yang dikirim akan menggantikan nilai yang ada, memungkinkan sistem untuk menyesuaikan atau memperbarui informasi terkait ketinggian sesuai dengan input yang diberikan. Proses ini penting untuk memastikan bahwa data yang ditampilkan atau digunakan dalam aplikasi selalu terkini dan akurat.

Permintaan:

- **Metode:** POST
  Menunjukkan bahwa metode HTTP yang digunakan adalah POST. Metode POST umumnya digunakan untuk mengirim data ke server untuk membuat atau memperbarui sumber daya.
- **Header:**
  - Content-Type: application/json

  Header ini menunjukkan bahwa data yang dikirimkan dalam permintaan ini berformat JSON (JavaScript Object Notation). Content-Type digunakan untuk memberi tahu server jenis data apa yang dikirim oleh klien.
- **Body:**
  Menunjukkan struktur dari data JSON yang perlu dikirim dalam body permintaan POST. Struktur JSON yang perlu dikirim dalam body permintaan. "<nilai_ketinggian>" adalah nilai yang ingin diubah atau diperbarui, dan akan digantikan dengan angka yang sesuai, misalnya 30.

Json

{
"ketinggian": <nilai_ketinggian>
}

- **Parameter:**
  - ketinggian (int): Nilai ketinggian yang ingin diubah.

  Parameter yang harus disertakan dalam permintaan. Parameter ini bernama ketinggian dan nilainya). Nilai ini menunjukkan ketinggian yang ingin Anda atur atau ubah melalui permintaan HTTP.

Contoh Permintaan:

http

POST https://pentagonal-crawling-shovel.glitch.me
Content-Type: application/json
 	
{
"ketinggian": 50
}

- **Respon:**
  - **Status: Siaga OK**
    Bagian dari respons yang menunjukkan bahwa permintaan Anda berhasil diproses. Status ini mungkin merujuk pada kondisi atau status dari server atau sistem setelah menerima dan memproses permintaan.
  - **Body:**
    Body pada respons ini berisi data dalam format JSON yang memberikan informasi mengenai hasil dari permintaan yang Anda kirimkan. Berikut adalah struktur dan penjelasan dari setiap elemen di dalamnya:

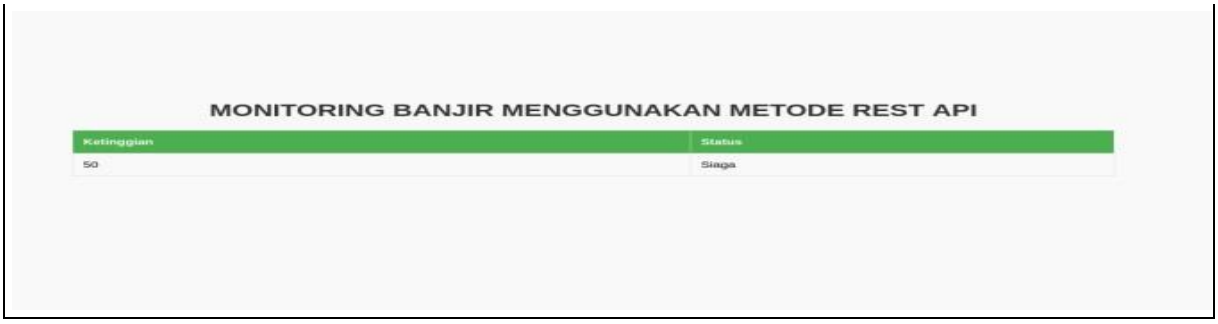
Json

{
"message": "Ketinggian berhasil diubah",
"ketinggian": <nilai_ketinggian>
}

- **Contoh Respon:**
  Berikut adalah contoh respons yang diberikan oleh server setelah memproses permintaan :
  Server mengirimkan pesan ini sebagai tanda bahwa ketinggian telah berhasil diubah seperti yang diminta. Ini menandakan bahwa operasi yang Anda minta telah dilakukan dengan sukses tanpa error. Nilai ketinggian ini menunjukkan angka 50, yang berarti server telah berhasil mengubah ketinggian menjadi 50, sesuai dengan yang diminta dalam permintaan Anda. Nilai ini sesuai dengan yang Anda kirim dalam body permintaan.

Json

{
"message": "Ketinggian berhasil diubah",
"ketinggian": 50
}



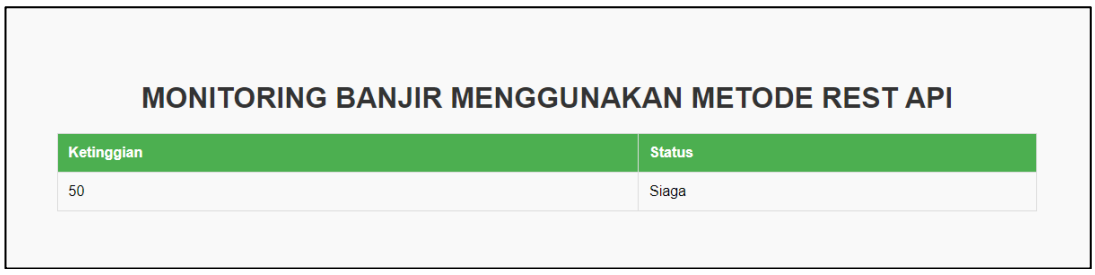
Gambar 15 Hasil Ujicoba REST API Update Nilai Ketinggian

Gambar 15 ini menampilkan informasi mengenai "Ketinggian". Pada contoh yang ditampilkan, ketinggian air adalah 50 (kemungkinan dalam satuan cm atau meter). Dan menampilkan "Status" kondisi berdasarkan ketinggian air, statusnya adalah Siaga, yang berarti ketinggian air dalam tidak aman dan akan membahayakan.

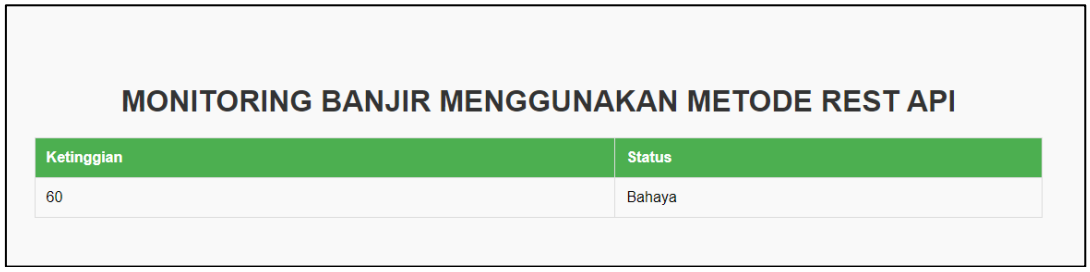
4. Hasil Pengujian Tampilan

Pengujian ini dilakukan agar peneliti mengetahui apakah program REST API dan program arduino yang dibuat untuk memnitoring ketinggian air sudah bekerja. Hasil pengujian tampilan peringatan dini banjir dengan menggunakan REST API dapat dilihat pada gambar 16 - 18 dibawah ini.

Gambar 16 Tampilan Dalam Kondisi Aman



Gambar 17 Tampilan Dalam Kondisi Siaga



Gambar 18 Tampilan Dalam Kondisi Bahaya

4. Pengujian Sensor Ultrasonik

Sensor ultrasonik HC-SR04 merupakan sensor yang dapat mengukur jarak atau tinggi dari 2 cm sampai 400 cm. Sensor ini menerima masukan tegangan mulai dari 1V sampai 5V. Keluaran sensor ultrasonik ini sebagai masukan bagi mikrokontroler berupa data analog yang akan diproses menjadi nilai jarak atau tinggi sebenarnya oleh mikrokontroler. Dilakukan perbandingan dalam pengukuran rangkaian sensor ultrasonik dengan mistar 30cm. Berikut tabel 3 pengukuran sensor ultrasonik HC-SR04.

Tabel 3 Perbandingan Pengukuran Oleh Mistar dan OlehSensor Ultrasonik

No	Pengukuran Oleh Mistar (cm)	Pengukuran Oleh Sensor Ultrasonik (cm)	Selisih *Error*(Cm)
1	0 Cm	0 Cm	0 Cm
2	1 Cm	2 Cm	1 Cm
3	2 Cm	3 Cm	1 Cm
4	4 Cm	4 Cm	0 Cm
5	6 Cm	6 Cm	0 Cm
6	8 Cm	8 Cm	0 Cm
7	10 Cm	10 Cm	0 Cm
8	12 Cm	12 Cm	0 Cm
9	14 Cm	14 Cm	0 Cm
10	16 Cm	17 Cm	1 Cm

Dari hasil pengujian didapat bahwa jarak hasil pengujian pada alat tidak sama dengan jarak hasil perhitungan dengan persentase kesalahan antara 0 Cm hingga 1 Cm. Scrip program sensor ultrasonik.

```
int trigPin = 13;
int echoPin = 15;
long duration, distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

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

```
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  Serial.print("Jarak: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);
}
```

Penjelasan kode program di atas adalah sebagai berikut:

Hubungkan kabel GND sensor dengan pin GND pada nodemcu. Hubungkan kabel VCC sensor dengan pin 5V pada nodemcu. Hubungkan kabel Trigger sensor dengan pin digital misal pin 13 pada nodemcu. Hubungkan kabel Echo sensor dengan pin digital misal pin 15 pada nodemcu nodemcu. Pada baris pertama, dideklarasikan variabel trigPin dan echoPin sebagai pin output dan input yang akan digunakan untuk mengirim dan menerima sinyal ultrasonik. Pada baris ketiga dan keempat, dideklarasikan variabel duration dan distance sebagai variabel yang akan digunakan untuk menyimpan durasi sinyal ultrasonik kembali dan jarak yang diukur. Pada fungsi setup, dilakukan pengaturan pin trigPin dan echoPin sebagai output dan input, serta inisialisasi Serial untuk komunikasi dengan komputer melalui port serial.

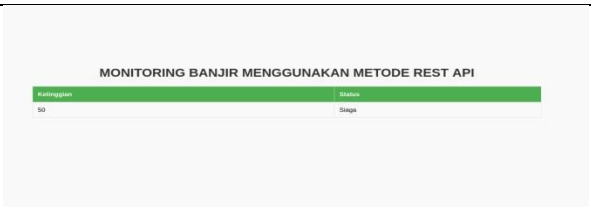
Pada fungsi loop, pertama-tama sinyal di pin trigPin dikirimkan dengan memberikan sinyal HIGH selama 10 mikrodetik, kemudian sinyal dikembalikan ke LOW. Kemudian, durasi sinyal yang diterima di pin echoPin dihitung dengan menggunakan fungsi pulseIn(). Setelah itu, jarak dihitung berdasarkan durasi tersebut dengan rumus jarak=(durasi/2)/29.1(dalam cm). Hasil jarak dioutputkan ke Serial Monitor dengan menggunakan fungsi Serial.print() dan Serial.println(), dan ditampilkan dalam satuan cm. Terakhir, dilakukan delay selama 500 milidetik sebelum mengulangi proses pengukuran jarak.

5. Pengujian Sistem Secara Keseluruhan

Pengujian sistem secara keseluruhan dilakukan untuk menguji kinerja Rancang, dilakukan ujicoba sistem agar peneliti dapat mengetahui apakah sistem yang telah dibuat dapat berkerja dengan baik. Hasil dari pengujian, termasuk berbagai aspek yang diuji dan hasil yang diperoleh, ditampilkan secara rinci pada Tabel 4 di bawah ini.

Tabel 4 Pengujian Sistem Keseluruhan

NO	Sensor Ultrasonik Cm	Status Buzzer	Hasil Tampilan
1	< 50cm	Tidak Aktif	 Sungai dalam kondisi aman

2	>50 dan <60 cm	Aktif	 Sungai dalam kondisi siaga
3	> 60 cm	Aktif	 Sungai dalam Kondisi Bahaya

Berdasarkan hasil pengujian diatas dapat diketahui jika hasil pembacaan sensor ultrasonic < 50cm maka dikatagorikan sungai dalam kondisi aman, sedangkan jika sensor ultrasonic >50 dan <60 cm maka dikatagorikan sungai siaga dan jika hasil pembacaan sensor >60cm maka dikatagorikan sugai berstatus bahaya. Sehingga dari hasil uicoba system keseluruhan dapat disimpulka jika alat yang dibuat dapat berkerja dengan baik dalam memonitoring ketiggian air sungai.

Pembahasan

Dari hasil pengujian dan analisa sistem, dapat disimpulkan bahwa REST API yang dikembangkan telah berhasil memenuhi tujuan utamanya dalam sistem peringatan dini banjir. REST API ini efektif dalam memonitor ketinggian air secara real-time, yang merupakan komponen kunci untuk memberikan peringatan dini yang tepat waktu. Keberhasilan ini menunjukkan bahwa REST API dapat berfungsi dengan baik dalam merespons data dari Sensor Ultrasonik, yang digunakan untuk memantau ketinggian air secara akurat.

Penggunaan metode GET dan POST dalam REST API juga terbukti optimal, menunjukkan bahwa sistem ini mampu menangani berbagai jenis data dengan baik. Selain itu, integrasi data dengan web monitoring sederhana menunjukkan bahwa REST API dapat beroperasi dengan baik dalam konteks sistem peringatan dini banjir, memberikan hasil yang sesuai dengan ekspektasi dalam hal kecepatan dan fungsionalitas. Namun, meskipun Sensor Ultrasonik berfungsi dengan baik, terdapat sedikit kesalahan dalam pengukuran yang perlu diperhatikan. Kesalahan ini, meskipun kecil, harus diminimalkan untuk memastikan akurasi data yang lebih tinggi dalam sistem peringatan dini. Untuk meningkatkan kinerja dan keandalan sistem, beberapa pengembangan perlu dilakukan. Salah satu langkah penting adalah menerapkan caching pada endpoint yang menyajikan data statis atau jarang berubah. Caching ini akan mengurangi beban server dan meningkatkan kecepatan respons, yang sangat krusial dalam situasi darurat di mana akses cepat dan andal terhadap informasi dapat mempengaruhi keselamatan masyarakat.

Selain itu, mempertimbangkan pembaruan dan peningkatan reguler pada sensor dan perangkat keras lainnya dapat membantu mengurangi kesalahan pengukuran dan meningkatkan akurasi data. Pengembangan lebih lanjut dari REST API dan integrasi dengan teknologi canggih

dapat memperkuat sistem peringatan dini banjir, memastikan bahwa sistem ini tetap efektif dalam menghadapi pengujian yang akan datang.

Kesimpulan

Dalam penilitian, telah diuraikan berbagai aspek yang berkaitan dengan perkembangan dan implementasi Protokol REST API pada sistem peringatan dini banjir. Berdasarkan pengujian dan analisa sistem yang telah dilakukan, dapat di simpulkan bahwa:

1. Pada sistem peringatan dini banjir dengan menggunakan Protokol REST API yang dikembangkan berhasil untuk memonitor ketinggian air secara real-time.
2. REST API bekerja dengan baik dalam merespons data dari Sensor Ultrasonik untuk memantau ketinggian air pada sistem peringatan dini banjir.
3. REST API berfungsi secara optimal dalam sistem peringatan dini banjir dengan memanfaatkan tipe data melalui metode GET dan POST.
4. Sensor Ultrasonik beroperasi dengan baik, meskipun terdapat sedikit kesalahan dalam pengukurannya.
5. Protokol REST API mampu mengintegrasikan data dengan aplikasi web monitoring sederhana dan beroperasi secara optimal dalam sistem peringatan dini banjir.

5.2 Saran

Alat ini masih terdapat kekurangan sehingga perlu dilakukan pengembangan. Seperti gunakan caching pada endpoint yang memberikan data statis atau jarang berubah, karena untuk mengurangi beban server dan meningkatkan kecepatan respons. hal ini sangat penting untuk memastikan informasi penting tersedia dengan cepat dan dapat diakses oleh semua pengguna yang membutuhkan, terutama dalam situasi darurat di mana kecepatan dan keandalan informasi dapat berdampak signifikan pada keselamatan masyarakat.

Daftar Pustaka

Choirudin, R., & Adil, A. (2019). Implementasi Rest Api Web Service Dalam Membangun Aplikasi Multiplatform Untuk Usaha Jasa. Matrik : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer, 18(2), 284–293. Https://Doi.Org/10.30812/Matrik.V18i2.407

Darmawan, A. K. (2021). Sistem Peringatan Dini Banjir Menggunakan Arduino Dan Sms Gateway Untuk Daerah Pamekasan.

Hanggara, F. D., & Eka Putra, R. D. (2021). Purwarupa Perangkat Deteksi Dini Banjir Berbasis Internet Of Things. Jurnal Informatika Dan Rekayasa Elektronik, 4(1), 87–94. Https://Doi.Org/10.36595/Jire.V4i1.349

Ibarreche, J., Aquino, R., Edwards, R. M., Rangel, V., Pérez, I., Martínez, M., Castellanos, E., Álvarez, E., Jimenez, S., Rentería, R., Edwards, A., & Álvarez, O. (2020). Flash Flood Early Warning System In Colima, Mexico. Sensors, 20(18), 5231. Https://Doi.Org/10.3390/S20185231

Implementasi,Rest Api Pada Fitur Rencana Strategis Dalam Aplikasi Website E-Government Studi Kasus Cv. Atsoft Teknologi.Pdf. (N.D.).

Kaniya Pradnya Paramitha, I. A., Wiharta, D. M., & Arsa Suyadnya, I. M. (2022). Perancangan Dan Implementasi Restful Api Pada Sistem Informasi Manajemen Dosen Universitas Udayana. Jurnal Spektrum, 9(3), 15. Https://Doi.Org/10.24843/Spektrum.2022.V09.I03.P3

Pratama, N. Z., Rismawan, T., & Suhardi, S. (2022). Penerapan Metode Regresi Linear Pada Sistem Peringatan Dini Banjir Berbasis Internet Of Things (Iot). Jurikom (Jurnal Riset Komputer), 9(5), 1414. Https://Doi.Org/10.30865/Jurikom.V9i5.4849

Priatim, R. A., Asri, M., & Abdussamad, S. (2023). Rancang Bangun Prototipe Peringatan Dini Banjir Menggunakan Raspberry Pi Berbasis Iot.

Santoso, P. P., & Rosmalia, L. (2023). Fish Feeding System In Aquaponic Media Using Firebase And Internet Of Things Based Codular. Proceeding International Conference On Information Technology And Business (ICITB).

Setyawan, D. (2014). Earthquake Early Warning System Real Time Design Using Total Electron Content And Geomagnetism With Fuzzy Logic. IJCSI (Internasional Journal Of Computer Science Issues), 11(6). Www.Ijcsi.Com

Setyawan, D. Y., Nurfiana, N., Rosmalia, L., & Setiawati, M. G. (2021). Gempa Bumi: Hubungan Data Sensor Mag3110 Dengan Data Sensor Adxl345 Berbasis IOT. Jurnal Teori Dan Aplikasi Fisika, 9(2), 185. Https://Doi.Org/10.23960/Jtaf.V9i2.2802

Sudibyo, N. H., & Ridho, M. (2015). Pendeteksi Tanah Longsor Menggunakan Sensor Cahaya. Jurnal Tim Dar majaya Vol. 01 No, 02, 218.

Ulum, M. B. (2023). Sistem Monitoring Cuaca Dan Peringatan Banjir Berbasis Iot Dengan Menggunakan Aplikasi Mit App Inventor. Jurnal Informatika Dan Teknik Elektro Terapan, 11(3). Https://Doi.Org/10.23960/Jitet.V11i3.3088

Windiastik, S. P., Ardhana, E. N., & Triono, J. (2019). Perancangan Sistem Pendeteksi Banjir Berbasis Iot (Internet Of Thing).