

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Banjir merupakan salah satu bencana alam yang paling sering melanda Indonesia, khususnya pada musim hujan ketika curah hujan tinggi melebihi kapasitas sungai maupun sistem drainase. Dampaknya mencakup kerusakan infrastruktur, kerugian ekonomi, hingga ancaman terhadap keselamatan jiwa (Ratmini et al., 2025). Kota Makassar sebagai salah satu kota metropolitan di kawasan timur Indonesia juga memiliki kerentanan tinggi terhadap banjir. Setiap tahun, wilayah ini terdampak banjir pada bulan Desember–Februari ketika curah hujan mencapai puncaknya, dengan catatan banjir besar pada 1967, 1976, 1999, dan 2000. Kawasan rawan meliputi aliran Sungai Tello, Jenneberang, Pampang, serta daerah Antang, Minasa Upa, hingga sekitar Pelabuhan (BPBD Makassar, 2023).

Menurut Arvi et al. (2025), literatur terbaru menunjukkan bahwa banjir di kota-kota besar Indonesia umumnya dipengaruhi dua faktor, yaitu faktor alam seperti curah hujan ekstrem, sedimentasi sungai, serta kondisi topografi dataran rendah, dan faktor manusia seperti urbanisasi yang cepat, pembangunan permanen yang menutup daerah resapan air, serta buruknya pengelolaan drainase. Kasus di Makassar sejalan dengan temuan ini, di mana sistem drainase yang masih bercampur antara air hujan dan limbah rumah tangga membuat aliran semakin terhambat (BPBD Makassar, 2023). Kondisi ini menegaskan pentingnya sistem monitoring dan peringatan dini banjir yang efektif agar masyarakat dapat memperoleh informasi secara cepat dan tepat. Upaya tersebut tidak hanya sebatas perbaikan fisik infrastruktur, melainkan juga memerlukan dukungan teknologi untuk meningkatkan akurasi deteksi dini dan penyebaran informasi. Seiring berkembangnya teknologi, konsep Internet of Things (IoT) menjadi solusi potensial dalam mitigasi bencana, khususnya pemantauan lingkungan secara *real-time* (Narayana et al., 2024).

Internet of Things (IoT) memiliki kontribusi signifikan dalam pengembangan sistem monitoring dan peringatan dini bencana banjir. Teknologi ini memungkinkan proses akuisisi, transmisi, dan pengolahan data dilakukan secara otomatis serta *real-time*, sehingga informasi yang dihasilkan dapat segera dimanfaatkan oleh masyarakat maupun pemerintah untuk melakukan tindakan mitigatif. Keunggulan utama IoT terletak pada kemampuannya mengintegrasikan berbagai sensor dan perangkat dalam suatu jaringan terhubung internet, yang memungkinkan pemantauan kondisi lingkungan dilakukan secara fleksibel dan tanpa batasan lokasi (Baharuddin et al., 2024). Hal ini memungkinkan pemerintah maupun masyarakat melakukan tindakan cepat sebelum banjir meluas. Dengan demikian, penerapan IoT dapat menjadi pelengkap strategi struktural dan non-struktural yang telah ada, sekaligus menyadarkan masyarakat terhadap ancaman banjir.

Berbagai penelitian membuktikan bahwa sistem monitoring berbasis IoT dengan sensor ultrasonik, mikrokontroler seperti ESP32, serta integrasi dengan aplikasi maupun bot pesan instan dan mampu memberikan notifikasi dini dengan akurasi. Hal ini memperkuat bahwa penggunaan IoT dalam sistem peringatan dini banjir memiliki peluang besar untuk membantu masyarakat dan pemerintah dalam upaya mitigasi bencana, khususnya di wilayah rawan seperti Kota Makassar.

Penelitian terbaru menunjukkan implementasi IoT dalam sistem monitoring banjir memberikan hasil yang signifikan:

- Ratmini et al., (2025) mengembangkan sistem monitoring banjir berbasis ESP32, Blynk, dan Telegram dengan akurasi deteksi ketinggian air hingga 95% dan latensi notifikasi 1,2 detik.
- Narayana et al., (2024) membuktikan bahwa IoT mampu memantau parameter lingkungan seperti curah hujan, kelembaban, dan ketinggian air melalui web monitoring berbasis HTTP secara *real-time*.

- Priatim et al., (2023) merancang prototipe peringatan dini banjir berbasis Raspberry Pi dan IoT, yang mampu mengirim informasi level air (normal, waspada, siaga, awas) ke aplikasi Blynk secara *real-time*.
- Hendra et al., (2023) menerapkan ESP32 dan sensor ultrasonik pada Bendungan Pamarayan untuk mendeteksi luapan air sungai serta mengirim peringatan dini secara otomatis.

Berdasarkan latar belakang tersebut, penelitian ini difokuskan pada perancangan prototipe sistem monitoring dan peringatan dini banjir berbasis IoT menggunakan ESP32 dan sensor ultrasonik yang terintegrasi dengan web monitoring dan early warning system (EWS) berupa notifikasi chatbot. Walaupun implementasi dilakukan dalam skala laboratorium sederhana, seperti simulasi genangan di ember, sistem ini berfungsi sebagai proof of concept yang dapat diterapkan di berbagai wilayah rawan banjir. Dengan demikian, hasil penelitian ini diharapkan dapat mendukung pengembangan sistem peringatan dini yang lebih luas sekaligus memberikan kontribusi dalam edukasi pemanfaatan IoT untuk mitigasi bencana.

## 1.2 Tujuan dan Manfaat

Tujuan dari penelitian ini adalah:

1. Merancang dan membangun **prototipe sistem monitoring dan peringatan dini banjir berbasis ESP32** dengan memanfaatkan konsep Internet of Things (IoT).
2. Mengintegrasikan **sensor ultrasonik** untuk mengukur ketinggian air dan **sensor hujan** untuk mendeteksi kondisi cuaca secara *real-time*, **sehingga dapat dianalisis apakah kenaikan muka air disebabkan oleh hujan lokal atau faktor lain.**
3. Mengembangkan **web monitoring** yang dapat menampilkan informasi ketinggian air, kondisi hujan, dan lokasi sensor secara akurat dan mudah diakses.
4. Menerapkan mekanisme **notifikasi otomatis melalui bot Telegram** sehingga masyarakat dapat memperoleh informasi dini ketika ketinggian air mencapai level berbahaya **tanpa harus melakukan pengecekan manual setiap waktu.**

Hasil penelitian ini diharapkan dapat bermanfaat:

1. Bagi Peneliti  
Penelitian ini menjadi sarana untuk menerapkan ilmu yang diperoleh selama perkuliahan ke dalam praktik nyata melalui perancangan dan implementasi **prototipe sistem monitoring ketinggian air dan peringatan dini berbasis IoT**. Dengan demikian, peneliti memperoleh pengalaman dalam mengintegrasikan perangkat keras, perangkat lunak, dan analisis data secara ilmiah untuk memecahkan permasalahan banjir, meskipun masih dalam skala simulasi laboratorium.
2. Bagi Lembaga/Pemerintah Daerah  
Hasil penelitian ini diharapkan dapat memberikan masukan dan pertimbangan bagi lembaga terkait, khususnya **BPBD atau instansi penanggulangan bencana di berbagai daerah**, dalam mengembangkan sistem peringatan dini banjir yang lebih efektif. Prototipe yang dihasilkan dapat dijadikan dasar pengembangan sistem operasional guna mendukung kebijakan mitigasi bencana dan memberikan informasi cepat kepada masyarakat.
3. Bagi Perkembangan Ilmu Pengetahuan  
Penelitian ini diharapkan dapat menambah wawasan dan memperkaya literatur dalam bidang **Internet of Things (IoT) untuk mitigasi bencana, khususnya banjir**. Hasil penelitian ini juga dapat menjadi referensi bagi penelitian selanjutnya dalam mengembangkan sistem serupa dengan skala lebih luas atau integrasi teknologi tambahan seperti prediksi curah hujan, sistem otomatisasi pintu air, maupun integrasi dengan platform resmi pemerintah.

### 1.3 Landasan Teori

#### 1.3.1 Banjir

Banjir merupakan salah satu bencana alam yang paling sering melanda wilayah Indonesia dan menimbulkan kerugian besar baik secara fisik, sosial, maupun ekonomi. Menurut Taslim et al. (2024), banjir didefinisikan sebagai tergenangnya suatu daerah akibat meluapnya air yang melebihi kapasitas saluran drainase atau sungai, sehingga mengakibatkan kerusakan infrastruktur dan mengganggu aktivitas masyarakat. Banjir juga termasuk bencana yang bersifat musiman, khususnya pada musim penghujan dengan curah hujan tinggi.

Penyebab banjir dapat dibagi ke dalam dua kelompok besar, yaitu faktor alam dan faktor manusia. Duykers et al. (2023) menjelaskan bahwa faktor alam meliputi intensitas curah hujan ekstrem, kondisi topografi dataran rendah, serta sedimentasi sungai yang menghambat aliran air. Di sisi lain, faktor manusia meliputi urbanisasi yang pesat, berkurangnya daerah resapan air, pembangunan permukiman di sempadan sungai, dan buruknya pengelolaan drainase. Arvi et al. (2025) menambahkan bahwa urbanisasi tanpa perencanaan tata ruang yang baik memperparah risiko banjir di kota-kota besar di Indonesia, termasuk Jakarta dan Makassar.

Dampak banjir sangat luas, mulai dari kerugian ekonomi akibat rusaknya infrastruktur, hilangnya mata pencaharian, hingga ancaman terhadap keselamatan jiwa. Suwarman et al. (2025) menekankan bahwa keterlambatan dalam penanganan banjir dapat memperbesar risiko korban jiwa dan kerugian material, karena banjir sering datang tiba-tiba dan meluas dalam waktu singkat. Oleh karena itu, diperlukan sistem peringatan dini yang mampu memberikan informasi secara cepat dan akurat.

Mitigasi banjir menjadi sangat penting untuk mengurangi risiko tersebut. Taslim et al. (2024) menyebutkan bahwa strategi mitigasi tidak hanya berfokus pada pembangunan fisik seperti tanggul dan drainase (strategi struktural), tetapi juga mencakup strategi non-struktural seperti edukasi masyarakat, penyebaran informasi kebencanaan, dan penerapan sistem peringatan dini. Dalam konteks ini, perkembangan teknologi menghadirkan peluang baru, salah satunya melalui penerapan Internet of Things (IoT) berbasis mikrokontroler seperti ESP32. Sistem ini dapat diwujudkan dalam bentuk prototype monitoring banjir yang dilengkapi sensor ketinggian air, kemudian diintegrasikan dengan web monitoring untuk pemantauan jarak jauh, serta early warning system (EWS) yang mampu memberikan notifikasi dini kepada masyarakat dan pihak terkait. Upaya mitigasi berbasis teknologi tersebut diharapkan dapat mempercepat penyampaian informasi banjir secara *real-time*, meningkatkan akurasi deteksi, serta memperkuat kesiapsiagaan masyarakat dalam menghadapi bencana.

#### 1.3.2 Hypertext Transfer Protokol

Web monitoring adalah sistem pemantauan berbasis web yang memungkinkan pengguna mengakses data sensor secara *real-time* melalui jaringan internet. Sistem ini bekerja dengan mengirimkan data dari perangkat sensorik atau mikrokontroler ke server, kemudian ditampilkan dalam antarmuka berbasis web sehingga dapat diakses kapan saja dan di mana saja melalui browser atau aplikasi seluler. Teknologi ini banyak digunakan dalam berbagai bidang, termasuk pemantauan lingkungan, pertanian cerdas, industri, dan mitigasi bencana. Pada konteks penelitian ini, web monitoring dimanfaatkan untuk memantau ketinggian air secara langsung sebagai upaya mendukung sistem peringatan dini banjir. ESP32 merupakan mikrokontroler yang dilengkapi modul Wi-Fi dan Bluetooth terintegrasi, sehingga ideal untuk aplikasi Internet of Things (IoT). Keunggulan ESP32 dibandingkan mikrokontroler generasi sebelumnya adalah kemampuannya menghubungkan sensor dengan server secara langsung tanpa membutuhkan modul tambahan. ESP32 mampu menjalankan komunikasi berbasis protokol HTTP maupun MQTT untuk mengirim dan menerima data dari server. Dengan kemampuan ini, ESP32 sering digunakan untuk membangun sistem pemantauan jarak jauh berbasis web karena proses pengiriman data menjadi lebih cepat dan hemat biaya (Priatim et al., 2023; Hendra et al., 2023).

Menurut Agusnadi et al., (2024), Hypertext Transfer Protocol (HTTP) adalah protokol komunikasi yang bekerja pada lapisan aplikasi dan digunakan untuk mengirim dan menerima berbagai jenis informasi di World Wide Web. HTTP memungkinkan pertukaran data berupa teks, gambar, suara, video, maupun berkas lainnya antara klien (misalnya browser atau perangkat IoT) dengan server melalui pola **request-response**. Klien mengirimkan permintaan HTTP kepada server. HTTP mendefinisikan tindakan spesifik yang diambil pada sumber daya di server melalui metode-metode tertentu seperti GET, POST, PUT, DELETE, HEAD, dan OPTIONS (Suharyanto & Simanjuntak, 2017).

- **GET** digunakan untuk meminta atau mengambil data dari server (misalnya menampilkan data sensor pada web monitoring).
- **POST** digunakan untuk mengirim data sensor dari ESP32 ke server atau database untuk disimpan dan diproses.
- **PUT** digunakan untuk memperbarui sumber daya tertentu jika dibutuhkan.
- **DELETE** digunakan untuk menghapus sumber daya tertentu.

Sistem monitoring dan peringatan dini banjir berbasis ESP32 ini, konsep Hypertext Transfer Protocol (HTTP) digunakan sebagai protokol komunikasi utama pada sistem monitoring dan peringatan dini banjir berbasis ESP32 ini. HTTP bekerja dengan pola permintaan dan tanggapan (request-response) antara klien dan server. Pada sistem yang dibangun, konsep ini diterapkan sebagai berikut. Pertama, ESP32 berperan sebagai klien HTTP. Perangkat ini membaca data sensor ketinggian muka air, sensor hujan, dan koordinat GPS, lalu mengemasnya dalam format JSON dan mengirimkannya ke server Flask menggunakan metode HTTP POST ke endpoint /update. Metode POST dipilih karena data yang dikirim bersifat baru (input) dan memicu logika di server untuk menentukan status banjir.

- Server Flask berperan sebagai server HTTP. Server ini menyediakan beberapa endpoint untuk menangani permintaan dari ESP32 dan browser, yaitu:
- Endpoint /update menerima permintaan HTTP POST dari ESP32 berisi data sensor, menambahkan stempel waktu (timestamp), menentukan status banjir, menyimpan data ke riwayat, dan meneruskan informasi ke layanan n8n jika status banjir terdeteksi.
- Endpoint /data melayani permintaan HTTP GET untuk mengirimkan data terbaru dalam format JSON ke browser atau antarmuka pengguna.
- Endpoint /history melayani permintaan HTTP GET untuk mengirimkan riwayat data dalam format JSON.
- Endpoint /download melayani permintaan HTTP GET untuk mengunduh riwayat dalam format CSV.
- Endpoint / melayani permintaan HTTP GET untuk menampilkan dashboard berbasis web (index.html).

*Front-end* (index.html) berperan sebagai klien web yang berjalan di web *browser*. Menggunakan JavaScript `fetch()` (GET) ke endpoint /data dan /history, halaman web ini menampilkan data *real-time* seperti tinggi muka air, kondisi cuaca, status banjir, riwayat pengukuran, dan peta lokasi sensor. Semua komunikasi antara perangkat keras, server, dan halaman web pengguna berlangsung melalui protokol HTTP sehingga sistem ini sederhana, fleksibel, dan mudah diintegrasikan dengan layanan pihak ketiga seperti n8n untuk pengiriman notifikasi otomatis.

### 1.3.3 Internet of Things

Internet of Things (IoT) merupakan konsep yang menghubungkan berbagai perangkat fisik seperti sensor, aktuator, dan mikrokontroler ke jaringan internet agar dapat saling berkomunikasi dan bertukar data secara *real-time*. Konsep ini memungkinkan integrasi antara perangkat keras dengan layanan berbasis web atau cloud untuk menghasilkan sistem yang cerdas, otomatis, dan terdistribusi (Pratama et al., 2022). Dalam penerapannya, IoT memungkinkan pengambilan data lingkungan secara kontinu,

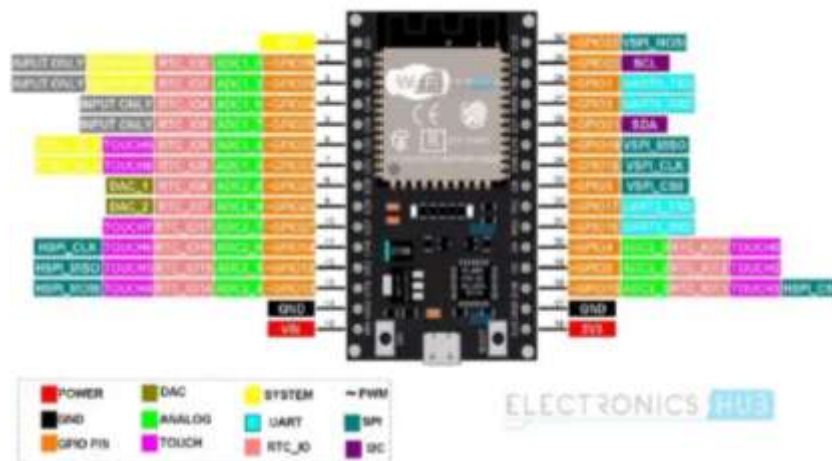
pengiriman data tersebut ke server untuk diproses dan dianalisis, serta penyajian informasi yang dihasilkan kepada pengguna atau sistem lain untuk mendukung pengambilan keputusan yang lebih cepat dan tepat.

Tujuan utama penerapan IoT adalah mempermudah pemantauan, pengendalian, dan pengelolaan sistem secara jarak jauh sehingga lebih efisien dan responsif. Pada konteks sistem monitoring banjir, IoT bertujuan untuk mengumpulkan data tinggi muka air, curah hujan, dan lokasi sensor secara otomatis dan *real-time*, mengirimkan data tersebut ke server web untuk dianalisis dan divisualisasikan sehingga mudah dipahami pengguna, serta memberikan peringatan dini atau notifikasi otomatis ketika kondisi berbahaya terdeteksi. Dengan memanfaatkan IoT, sistem monitoring dapat meningkatkan efektivitas, kecepatan respon, dan ketepatan informasi dibandingkan metode pengukuran manual, sekaligus memudahkan integrasi dengan berbagai layanan pihak ketiga untuk mendukung sistem peringatan dini yang lebih luas dan handal (Choudhary et al., 2025).

Pemanfaatan Internet of Things (IoT) pada sistem monitoring dan peringatan dini banjir berbasis ESP32 memberikan berbagai keuntungan dibandingkan metode konvensional. Dengan IoT, sensor ketinggian air, sensor hujan, dan modul GPS yang dipasang di lokasi rawan banjir dapat terhubung langsung ke jaringan internet sehingga data lingkungan dapat dikumpulkan dan dikirim secara *real-time* tanpa keterlibatan manusia secara langsung. Data yang masuk ke server web kemudian dianalisis dan divisualisasikan secara otomatis melalui dashboard, sehingga pengguna dapat memantau kondisi banjir kapan saja dan dari mana saja hanya dengan perangkat yang terhubung ke internet. Selain itu, integrasi IoT memungkinkan sistem untuk memberikan peringatan dini secara otomatis ketika ketinggian air mencapai ambang batas berbahaya. Informasi tersebut dapat dikirim ke berbagai pihak melalui layanan notifikasi seperti Telegram yang diotomasi melalui platform n8n. Pendekatan ini meningkatkan kecepatan respon masyarakat atau pihak berwenang dalam mengambil tindakan mitigasi. Sistem menjadi lebih efisien dan hemat biaya karena tidak memerlukan petugas untuk melakukan pengukuran manual secara terus-menerus. Data historis yang tersimpan di server juga dapat digunakan untuk analisis tren dan prediksi banjir di masa depan. Dengan kata lain, penerapan IoT pada sistem monitoring banjir tidak hanya memberikan kemudahan pemantauan dan peringatan dini, tetapi juga mendukung pengambilan keputusan berbasis data yang lebih akurat dan cepat, serta memperluas jangkauan informasi ke seluruh pemangku kepentingan secara instan.

#### **1.3.4 Modul ESP32**

ESP32 adalah mikrokontroler yang dikembangkan oleh Espressif Systems sebagai generasi penerus dari ESP8266. Modul ini dirancang dengan arsitektur 32-bit berbasis prosesor Tensilica Xtensa LX6 dual-core yang mampu berjalan hingga 240 MHz (Hendra & Pradana, 2019). Keunggulan ESP32 adalah sudah dilengkapi dengan modul Wi-Fi 802.11 b/g/n dan Bluetooth v4.2 BLE sehingga dapat langsung terkoneksi dengan jaringan internet tanpa memerlukan modul tambahan. Fitur ini menjadikan ESP32 sangat populer dalam pengembangan sistem berbasis Internet of Things (IoT) karena mendukung komunikasi nirkabel secara efisien dan hemat biaya.



**Gambar 1.**ESP32 Pinout Parts

**Tabel 1.** Fungsi Pinout ESP32

| Warna            | Fungsi           | Penjelasan                                                            |
|------------------|------------------|-----------------------------------------------------------------------|
| Merah            | Power            | Pin untuk daya (VIN, 3V3)                                             |
| Hitam            | Gnd              | Ground                                                                |
| Oranye           | Gpio Pin         | Pin input/output biasa ( <i>General Purpose Input Output</i> )        |
| Hijau            | Analog / Adc     | Pin yang bisa membaca tegangan analog (ADC1/ADC2)                     |
| Coklat           | Dac              | Pin <i>Digital-to-Analog Converter</i> (menghasilkan tegangan analog) |
| Kuning           | Sistem           | Pin sistem seperti EN, BOOT                                           |
| Pink / Ungu Muda | Touch            | Pin sensor sentuh kapasitif                                           |
| Biru Muda        | Uart             | Pin komunikasi serial (TX, RX)                                        |
| Biru Tua         | Spi              | Pin komunikasi SPI (MOSI, MISO, CLK, CS)                              |
| Ungu             | I <sup>2</sup> C | Pin komunikasi I <sup>2</sup> C (SCL, SDA)                            |

Secara spesifikasi, ESP32 memiliki komponen dan fitur sebagai berikut (Pratama & Kiswantono, 2022):

- **Memori internal:** 448 kB ROM, 520 kB SRAM, 8 kB RTC memory
- **Flash memory:** 4 MB
- **GPIO:** hingga 34 pin
- **ADC:** 18 channel dengan resolusi 12-bit
- **UART:** 3 unit
- **I2C:** 2 unit
- **SPI:** 4 unit
- **Prosesor:** Dual-core Xtensa LX6, hingga 240 MHz
- **Konektivitas:** Wi-Fi 802.11 b/g/n dan Bluetooth v4.2 BLE
- **Varian modul:** ESP32-WROOM-32, ESP32-WROVER, ESP32-PICO-KIT

Dengan kombinasi memori besar, prosesor cepat, serta dukungan banyak antarmuka komunikasi, ESP32 sangat fleksibel digunakan pada berbagai aplikasi monitoring maupun kontrol otomatis.

Fungsi utama ESP32 adalah sebagai pengendali sistem elektronik sekaligus penghubung perangkat dengan internet. ESP32 mampu membaca data dari sensor, mengolah data tersebut, dan mengirimkannya ke server melalui protokol komunikasi seperti HTTP atau MQTT (Hendra & Pradana, 2019). Dalam implementasi praktis, ESP32 sering digunakan untuk sistem monitoring lingkungan, smart home, industri, hingga mitigasi bencana. Misalnya, modul ini dapat membaca data dari sensor suhu, kelembaban, atau ketinggian air, kemudian mengirimkan informasi tersebut ke aplikasi web atau mobile agar dapat diakses secara *real-time*.

Penelitian ini menggunakan **ESP32 dipilih sebagai mikrokontroler utama pada sistem monitoring ketinggian air dan peringatan dini banjir**. Perangkat ini berfungsi untuk membaca data dari sensor ultrasonik, sensor hujan, serta modul GPS, kemudian mengolah dan mengirimkan data tersebut secara *real-time* melalui jaringan Wi-Fi. Dengan kemampuannya mengintegrasikan berbagai sensor dan mendukung konektivitas nirkabel, ESP32 menjadikan sistem ini efisien, hemat biaya, serta mampu memberikan informasi dan peringatan dini ketika ketinggian air mencapai ambang batas bahaya, sehingga mendukung upaya mitigasi banjir secara lebih cepat dan tepat.

1.3.5 Modul GPS Neo 6M

Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang digunakan untuk mengetahui letak suatu objek di permukaan bumi. Sistem ini memanfaatkan sinyal gelombang mikro yang dikirimkan oleh satelit dan diterima oleh perangkat penerima di darat. Sinyal tersebut kemudian diproses untuk menghasilkan informasi mengenai koordinat posisi, kecepatan, arah pergerakan, serta waktu. Satelit berfungsi sebagai elemen utama dalam sistem GPS yang bertugas menyediakan dan mengirimkan data navigasi. Data yang diterima selanjutnya diolah oleh penerima untuk menentukan lokasi dan informasi pendukung lainnya.

Modul GPS Neo-6M merupakan produk Ublox generasi ke-6 seri 6M yang terdiri dari sebuah receiver dan antena berukuran 20 × 20 mm serta bersifat stand-alone sehingga dapat beroperasi secara mandiri. Modul ini mampu menghasilkan data posisi dan kecepatan dengan tingkat akurasi yang tinggi, namun memiliki kelemahan karena rentan terhadap gangguan lingkungan yang dapat menyebabkan sinyal GPS menjadi tidak stabil hingga hilang. Dalam pengoperasiannya, modul GPS Neo-6M membutuhkan tegangan kerja maksimum sebesar 3,6 V dengan arus maksimum 67 mA dan dilengkapi dengan empat pin utama, yaitu VCC, GND, TX (transmitter), dan RX (receiver). Pin VCC berfungsi sebagai masukan tegangan sumber, pin GND sebagai ground, sedangkan pin TX dan RX digunakan sebagai jalur komunikasi serial antara modul GPS dan perangkat pendukung lainnya seperti mikrokontroler. Adapun spesifikasi Modul GPS seri Neo-6M, adalah:

Tabel 2. Spesifikasi GPS Neo 6M

| Parameter                 | Spesifikasi |
|---------------------------|-------------|
| Receiver                  | 50 Channels |
| Horizontal accuracy 2.5 m | 2.5 m       |
| TTFF                      |             |
| - Cold Start              | 30 s        |
| - Hot Start               | 1.5 s       |

Berdasarkan tabel di atas, Time to First Fix (TTFF) merupakan salah satu parameter penting yang digunakan untuk menganalisis kinerja modul GPS. TTFF didefinisikan sebagai waktu yang diperlukan oleh receiver GPS sejak dinyalakan hingga mampu menerima sinyal dari satelit, memperoleh data navigasi, dan menentukan posisi awal. Apabila receiver telah menyimpan informasi navigasi dan waktu sebelumnya, maka proses perolehan posisi akan berlangsung lebih cepat, kondisi ini dikenal sebagai hot-start. Sebaliknya, jika data navigasi belum tersedia, receiver membutuhkan waktu yang lebih lama karena harus melakukan pencarian satelit dari awal, yang disebut sebagai cold-start. Selain itu, parameter HDOP (Horizontal Dilution of Precision) menunjukkan tingkat kesalahan presisi posisi horizontal yang dipengaruhi oleh jumlah satelit yang terdeteksi, kondisi lingkungan, topografi, atmosfer, serta kualitas antena GPS. Nilai HDOP yang semakin kecil menandakan tingkat akurasi posisi horizontal yang semakin baik (Kusuma et al., 2024).

Tabel 3. Kategori Nilai HDOP

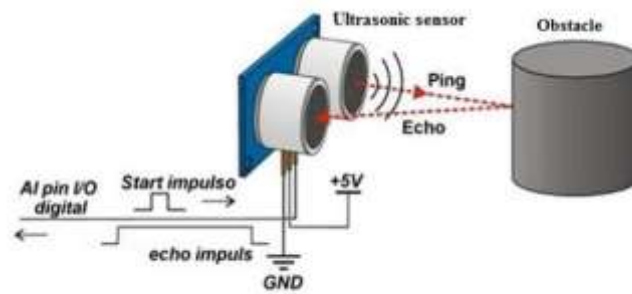
| DOP   | Kategori    | Keterangan                    |
|-------|-------------|-------------------------------|
| 0 – 1 | Ideal       | Tingkat akurasi sangat tinggi |
| 1 – 2 | Sangat Baik | Posisi tergolong akurat       |

|         |        |                                   |
|---------|--------|-----------------------------------|
| 2 – 5   | Baik   | Akurasi cukup, namun tidak tinggi |
| 5 – 10  | Sedang | Tingkat akurasi kurang memadai    |
| 10 – 20 | Cukup  | Posisi tidak akurat               |
| > 20    | Buruk  | Data tidak dapat diandalkan       |

Berdasarkan tabel di atas, nilai HDOP dipengaruhi oleh konfigurasi posisi relatif satelit terhadap receiver, sehingga setiap perangkat GPS dapat menghasilkan nilai HDOP yang berbeda. Perbedaan tersebut dipengaruhi oleh beberapa faktor, antara lain kondisi lingkungan sekitar dan kualitas antena yang digunakan (Kusuma et al., 2024).

### 1.3.6 Modul Sensor Ultrasonik

Sensor ultrasonik HC-SR04 merupakan salah satu jenis sensor yang digunakan untuk mengukur jarak dengan memanfaatkan gelombang ultrasonik. Sensor ini memiliki dua komponen utama, yaitu transmitter sebagai pemancar gelombang dan receiver sebagai penerima pantulan gelombang. HC-SR04 mampu melakukan pengukuran dengan jangkauan antara 2 cm hingga 400 cm dengan tingkat akurasi  $\pm 3$  mm, sehingga banyak digunakan dalam berbagai aplikasi seperti robotika, sistem keamanan, dan pemantauan lingkungan (Amin, 2020).



**Gambar 2.** Prinsip Kerja Sensor Ultrasonik

Prinsip kerja HC-SR04 didasarkan pada pemancaran gelombang ultrasonik dengan frekuensi 40 kHz melalui pin Trigger ketika diberi pulsa selama 10  $\mu$ s. Gelombang ini merambat di udara, kemudian dipantulkan kembali oleh suatu objek dan diterima oleh pin Echo. Selanjutnya, waktu tempuh gelombang dihitung dan digunakan untuk menentukan jarak menggunakan rumus :

$$s = v \times \frac{t}{2} \quad (1)$$

Dimana  $s$  adalah jarak objek,  $v$  adalah kecepatan rambat suara di udara ( $\pm 344$  m/s), dan  $t$  adalah waktu tempuh sinyal. Dengan metode ini, HC-SR04 dapat mengukur jarak secara cepat dan relatif akurat tanpa dipengaruhi oleh warna maupun intensitas cahaya objek (Hakim & Alamsyah, 2023).

Keunggulan HC-SR04 dibandingkan sensor lain adalah harganya yang relatif murah, mudah diperoleh, serta kemudahan integrasi dengan berbagai mikrokontroler seperti Arduino maupun ESP32. Selain itu, sensor ini memiliki konsumsi daya rendah, akurasi cukup tinggi, serta tidak terpengaruh oleh kondisi cahaya sehingga cocok digunakan untuk aplikasi monitoring berbasis Internet of Things (IoT). Faktor-faktor tersebut menjadikan HC-SR04 banyak digunakan dalam penelitian maupun implementasi praktis, khususnya di bidang pemantauan lingkungan.



**Gambar 3.** Bentuk Fisik Sensor Ultrasonik

Sensor ultrasonik HC-SR04, terdapat dua pin utama yang berperan penting dalam proses pengukuran jarak, yaitu pin Trigger (Trig) dan pin Echo. Pin Trigger berfungsi untuk memberikan sinyal pemicu agar sensor memancarkan gelombang ultrasonik. Mikrokontroler (misalnya ESP32) mengirimkan pulsa logika tinggi (HIGH) dengan durasi minimal 10 mikrodetik ( $\mu s$ ) ke pin ini. Saat menerima pulsa tersebut, transmitter pada HC-SR04 akan memancarkan delapan gelombang ultrasonik dengan frekuensi 40 kHz ke arah depan sensor (Hakim & Alamsyah, 2023). Sementara itu, pin Echo berfungsi untuk menerima sinyal pantulan (echo) dari objek yang terdeteksi. Setelah gelombang dipancarkan, pin Echo akan berubah menjadi logika tinggi (HIGH) selama waktu tempuh sinyal ultrasonik dari transmitter menuju objek dan kembali ke receiver. Durasi logika HIGH pada pin Echo inilah yang digunakan oleh mikrokontroler untuk menghitung jarak objek menggunakan persamaan di atas.

Penelitian ini memanfaatkan sensor ultrasonik HC-SR04 digunakan untuk mengukur ketinggian air sebagai parameter utama sistem monitoring banjir. Pada program yang dirancang, pin Trigger (GPIO12) digunakan untuk memancarkan gelombang ultrasonik, sedangkan pin Echo (GPIO13) menerima pantulan gelombang yang digunakan ESP32 untuk menghitung jarak permukaan air dengan sensor. Hasil perhitungan tersebut kemudian diproses lebih lanjut untuk menentukan status kondisi, misalnya “aman” atau “bahaya”. Jika kondisi bahaya terdeteksi, ESP32 akan mengaktifkan buzzer (GPIO14) sebagai alarm peringatan dini. Data hasil pembacaan HC-SR04 juga dikombinasikan dengan sensor hujan dan modul GPS untuk memperkuat sistem peringatan dini berbasis IoT, sehingga mendukung mitigasi banjir secara lebih efektif dan *real-time*.

### 1.3.7 Modul Sensor Hujan FC-37

Sensor hujan adalah jenis sensor yang berfungsi untuk mendeteksi terjadinya hujan atau tidak, yang dapat difungsikan dalam berbagai macam aplikasi kehidupan sehari-hari. Modul sensor hujan FC-37 adalah alat yang digunakan sebagai pendeteksi ketika air hujan jatuh melalui papan hujan, sekaligus untuk mengukur intensitas curah hujan. Modul ini terdiri dari papan hujan (rain board) dan papan kontrol (control board) yang terpisah, dilengkapi indikator LED, serta sensitivitasnya dapat disesuaikan menggunakan potensiometer (Ichsan et al., 2022).



**Gambar 4.** Modul Sensor Hujan

Prinsip kerja sensor hujan FC-37 adalah ketika air hujan mengenai panel sensor, maka terjadi proses elektrolisis karena air hujan bersifat sebagai cairan elektrolit yang dapat menghantarkan arus listrik. Kondisi ini menyebabkan resistansi antar jalur elektroda pada papan sensor menurun. Modul sensor ini dilengkapi dengan IC komparator yang menghasilkan output berupa logika digital (D0) dan tegangan analog (A0). Pada output digital, nilai akan bernilai logika tinggi (HIGH) ketika papan sensor kering, dan logika rendah (LOW) ketika papan sensor terkena air. Sedangkan pada output analog, sensor menghasilkan perubahan tegangan yang sebanding dengan intensitas curah hujan dan dapat dihubungkan ke pin ADC (Analog Digital Converter) pada mikrokontroler (Syarmuji et al., 2022).

Dalam penelitian ini, digunakan output analog (A0) dari rain sensor yang dihubungkan ke GPIO34 (ADC1) pada ESP32 untuk membaca variasi tegangan sesuai intensitas hujan. Pembacaan sinyal analog ini memungkinkan sistem membedakan tingkat hujan, bukan sekadar ada atau tidak adanya hujan. Selain itu, modul sensor ini juga dilengkapi dengan LED indikator yang menyala ketika air terdeteksi pada papan sensor, sehingga dapat berfungsi sebagai pemberitahuan visual secara langsung. Data dari sensor hujan kemudian diproses oleh ESP32 bersama dengan data ketinggian air dan koordinat GPS, lalu ditampilkan pada dashboard web monitoring untuk mendukung sistem peringatan dini banjir berbasis IoT secara *real-time*.

### 1.3.8 Modul Liquid Crystal Display

Liquid Crystal Display (LCD) adalah sebuah teknologi tampilan yang menggunakan lapisan kristal cair di antara dua lembar substrat kaca. Ketika diberikan tegangan listrik, kristal cair tersebut mengatur cahaya yang melewati mereka, sehingga dapat menampilkan gambar atau teks. LCD 16x2 dan mampu menampilkan 32 karakter yang terdiri dari 2 baris, masing-masing 16 karakter per baris. Dengan tambahan modul I<sup>2</sup>C, LCD dapat dihubungkan ke mikrokontroler hanya melalui dua pin data (SDA dan SCL), sehingga lebih sederhana dan hemat pin dibandingkan koneksi paralel. Kemampuan LCD ini membuatnya banyak digunakan pada sistem berbasis mikrokontroler untuk menampilkan informasi secara langsung kepada pengguna, seperti status sensor, data hasil pembacaan, maupun pesan peringatan. Peran LCD adalah sebagai antarmuka pengguna (user interface) yang memperlihatkan data atau status sistem secara *real-time*. Contohnya pada penelitian Ahmad Hunaepi et al., (2023), LCD I<sup>2</sup>C digunakan untuk menampilkan hasil pengenalan wajah yang diproses oleh ESP32CAM pada sistem kehadiran pendidik dan tenaga kependidikan berbasis mikrokontroler. Dengan LCD, pengguna dapat langsung melihat apakah wajah sudah terdaftar dan tercatat dalam data kehadiran tanpa harus membuka dashboard web.

LCD juga memiliki beberapa keterbatasan seperti ukuran layar kecil yang mampu menampilkan teks sederhana atau karakter terbatas, sehingga kurang cocok untuk menampilkan grafik kompleks. Selain itu, LCD kurang terlihat jelas di bawah cahaya matahari langsung dan membutuhkan suplai daya tambahan untuk backlight. Pada sistem monitoring dan peringatan dini banjir berbasis ESP32, LCD digunakan untuk menampilkan informasi penting secara *real-time* di lapangan seperti ketinggian air, kondisi hujan, status banjir, dan koordinat GPS. Dengan begitu petugas di lokasi bisa langsung memantau kondisi tanpa perlu membuka laptop atau ponsel. Penggunaan LCD ini memperkuat fungsi sistem sebagai perangkat peringatan dini yang mudah dibaca dan dioperasikan.

### 1.3.9 Platform Node-to-Node (N8N)

N8N merupakan sebuah platform otomatisasi alur kerja berbasis *open-source* dan *fair-code* yang dikembangkan untuk mempermudah integrasi berbagai layanan, aplikasi, maupun perangkat tanpa memerlukan penulisan kode yang kompleks. Konsep utama dari **n8n** adalah *node to node*, yaitu setiap node mewakili suatu fungsi tertentu seperti penerimaan data, pemrosesan logika, maupun pengiriman output, yang kemudian dapat dirangkai menjadi sebuah alur kerja (*workflow*) yang terstruktur. Dengan pendekatan ini, N8N tidak hanya berperan sebagai alat otomatisasi, tetapi juga mampu menghubungkan berbagai sistem dalam satu kesatuan alur kerja digital. Menurut Ramadhani et al. (2025), N8N dapat digunakan dalam pembuatan chatbot cerdas yang terintegrasi dengan kecerdasan buatan untuk menganalisis data percakapan, menyimpan interaksi, dan menghasilkan laporan otomatis. Sementara itu, Wali et al. (2025) menekankan bahwa N8N sebagai platform *low-code/no-code* terbukti meningkatkan efisiensi operasional, mengurangi kesalahan manual, serta mempercepat pengolahan data pada organisasi skala kecil maupun menengah.

Kelebihan N8N dibandingkan platform otomasi lainnya terletak pada fleksibilitas dan ketersediaannya yang *open-source*, sehingga pengguna dapat memodifikasi serta menyesuaikan alur kerja sesuai dengan kebutuhan. Platform ini mendukung integrasi dengan lebih dari 200 aplikasi melalui API, webhook, dan protokol standar, sehingga mampu beradaptasi dengan berbagai konteks penggunaan. Penelitian Wali et al. (2025) menunjukkan bahwa penerapan N8N dapat mengurangi waktu pemrosesan hingga 85% dan menekan tingkat kesalahan input data lebih dari 90%, yang membuktikan kemampuannya dalam meningkatkan efisiensi kerja. Selain itu, N8N memiliki keunggulan dalam skalabilitas karena alur kerja yang dirancang bersifat modular dan dapat digunakan kembali, sehingga sesuai untuk sistem yang memerlukan pengembangan berkelanjutan. Hal ini membuat n8n banyak digunakan pada berbagai sektor seperti layanan keuangan, pendidikan daring, maupun sistem berbasis *Internet of Things* (IoT) yang membutuhkan integrasi data secara *real-time* dan responsif (Ramadhani et al., 2025).

Penelitian ini memanfaatkan N8N untuk mendukung sistem *Early Warning System* (EWS) banjir berbasis IoT dengan mengintegrasikan data sensor dari ESP32 agar dapat diteruskan menjadi notifikasi otomatis. Data yang dikirim dari sensor ultrasonik, sensor hujan, dan modul GPS ke server diproses melalui n8n menggunakan beberapa node utama.

**Tabel 4.** Komponen n8n yang digunakan dalam sistem Early Warning Banjir

| Komponen n8n      | Fungsi                                                                         | Integrasi dalam Sistem Early Warning Banjir                                                                      |
|-------------------|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Scheduler Node    | Menjalankan alur kerja sesuai jadwal tertentu.                                 | Digunakan untuk memicu workflow setiap 5 detik agar data sensor selalu diperbarui.                               |
| HTTP Request Node | Mengambil data dari server atau layanan eksternal.                             | Mengakses server Flask untuk memperoleh data tinggi muka air, kondisi hujan, dan lokasi sensor.                  |
| Switch Node       | Mengecek kondisi atau menentukan jalur alur kerja berdasarkan logika tertentu. | Membandingkan tinggi muka air dengan ambang batas untuk menentukan status banjir (normal, waspada, siaga, awas). |
| Telegram Node     | Mengirimkan pesan notifikasi secara otomatis ke aplikasi Telegram.             | Mengirim peringatan dini kepada pengguna ketika kondisi banjir terdeteksi.                                       |

Integrasi komponen ini menjadikan sistem peringatan dini bekerja secara cepat, akurat, dan *real-time*, sehingga mendukung tujuan penelitian dalam merancang prototipe sistem monitoring dan peringatan dini banjir berbasis ESP32 yang efektif serta bermanfaat dalam mitigasi bencana.

## **BAB 2 METODE PENELITIAN**

### **2.1 Bahan dan Alat**

**Spesifikasi alat dan bahan yang digunakan dalam penerapan perancangan dan pembuatan alat pendeteksi banjir berbasis IoT ini adalah:**

#### **2.1.1 Bahan**

**Bahan yang dibutuhkan:**

- a. Timah (untuk penyolderan komponen)
- b. karton board (papan rangkaian permanen)
- c. Adaptor 5 V (sumber daya untuk sistem)
- d. Kabel jumper (penghubung antar komponen)
- e. LCD I2C 16x2 (untuk menampilkan status banjir)
- f. Sensor Ultrasonik HC-SR04 (untuk mengukur ketinggian air)
- g. Sensor Hujan (Rain Sensor FC-37) (untuk mendeteksi kondisi hujan)
- h. Modul GPS NEO-6M (untuk memperoleh koordinat lokasi alat)
- i. Buzzer (sebagai alarm peringatan dini)
- j. LED (indikator kondisi sistem)
- k. Modul Wi-Fi ESP32 (mikrokontroler utama dengan konektivitas internet)
- l. ESP32 (mikrokontroler utama dengan modul Wi-Fi dan Bluetooth bawaan)

#### **2.1.2 Alat**

**Alat yang dibutuhkan:**

- a. Laptop (untuk pemrograman dan konfigurasi)
- b. Kabel Micro USB / Type-C (untuk upload program ke ESP32)
- c. Solder (untuk penyambungan komponen ke PCB)
- d. Breadboard (untuk perakitan rangkaian sementara)
- e. Ember
- f. Penggaris

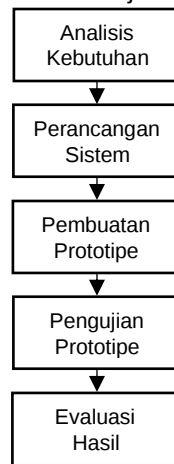
### **2.2 Pelaksanaan penelitian**

Pelaksanaan penelitian ini dilakukan melalui beberapa tahapan yang dilakukan secara berurutan mulai dari analisis kebutuhan, perancangan sistem, pembuatan prototipe, hingga tahap pengujian dan evaluasi hasil.

#### **2.2.1 Model pengembangan**

Model pengembangan yang digunakan pada penelitian ini adalah model penelitian dan pengembangan (Research and Development/R&D). Model R&D adalah model pengembangan yang dilakukan dengan cara merancang, membuat, dan menguji suatu produk tertentu sehingga dapat dimanfaatkan secara optimal. Menurut Waruwu (2024), penelitian dan pengembangan merupakan metode yang memadukan kegiatan riset untuk menemukan pengetahuan baru dengan kegiatan pengembangan untuk menghasilkan produk yang aplikatif. Tujuan dari pengembangan model R&D adalah menciptakan suatu produk inovatif yang mampu memecahkan masalah, menyempurnakan produk yang sudah ada, serta menguji tingkat efektivitas produk melalui serangkaian uji coba. Dalam penelitian ini, model R&D dipilih karena sesuai untuk merancang prototipe sistem *Early Warning System* (EWS) banjir berbasis

ESP32 yang terintegrasi dengan n8n dan Telegram, sehingga penelitian tidak hanya berhenti pada pembuatan prototipe, tetapi juga mencakup evaluasi performa sensor, kecepatan notifikasi, dan efektivitas sistem dalam memberikan peringatan dini banjir.



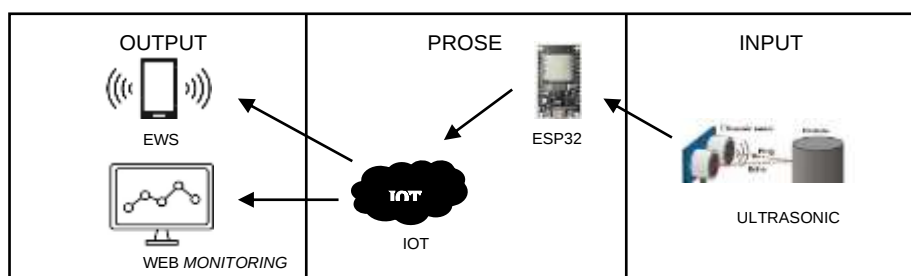
**Gambar 5.** Model Research and Development (R&D) dengan Pendekatan Prototyping

Model penelitian dan pengembangan (**Research and Development/R&D**) dengan pendekatan prototyping merupakan salah satu metode pengembangan produk yang digunakan untuk menghasilkan prototipe awal, kemudian dilakukan pengujian, perbaikan, dan evaluasi sehingga produk yang dihasilkan mendekati kebutuhan sebenarnya. Metode ini bersifat iteratif, artinya proses pengembangan dilakukan secara berulang dengan penyempurnaan pada setiap tahapannya. Tujuan utama dari pendekatan ini adalah memastikan produk yang dikembangkan sesuai dengan kebutuhan pengguna sekaligus dapat diuji keefektifannya dalam kondisi nyata. Tahapan-tahapan dalam pengembangan R&D dengan prototyping meliputi:

1. **Analisis kebutuhan**, yaitu mengidentifikasi masalah banjir dan menentukan kebutuhan sistem deteksi berbasis sensor serta integrasi notifikasi.
2. **Perancangan sistem**, yang mencakup desain rangkaian perangkat keras (ESP32, sensor ultrasonik, sensor hujan, GPS, LCD, dan buzzer) serta alur perangkat lunak (server Flask dan workflow n8n).
3. **Pembuatan prototipe**, yaitu merakit perangkat keras dan mengimplementasikan kode program pada ESP32 serta server pendukung.
4. **Pengujian prototipe**, dengan cara menguji akurasi sensor, pengiriman data, dan integrasi n8n untuk memicu notifikasi.
5. **Evaluasi hasil**, yaitu menilai performa sistem, kecepatan respon, serta efektivitas sistem dalam memberikan peringatan dini banjir secara *real-time*.

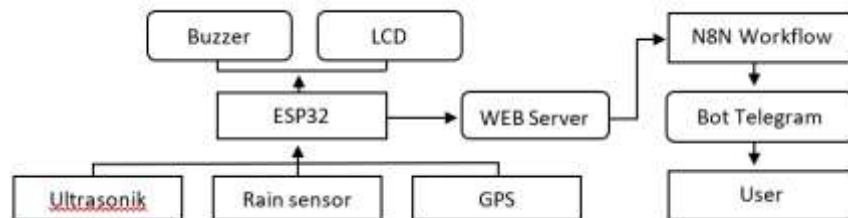
Melalui tahapan ini, prototipe sistem yang dihasilkan tidak hanya memberikan gambaran awal, tetapi juga dapat diuji kelayakan dan efektivitasnya. Dengan demikian, pendekatan R&D berbasis prototyping sangat sesuai diterapkan dalam penelitian ini karena mampu menghasilkan sistem peringatan dini banjir yang praktis, responsif, serta bermanfaat dalam mitigasi bencana.

### 2.2.2 Desain Sistem



**Gambar 6.** Desain Arsitektur Sistem

Tahap desain bertujuan untuk merumuskan rancangan tampilan, interaksi, serta prosedur kerja sistem yang akan digunakan dalam proses pengembangan dan pembuatan alat. Desain ini disusun berdasarkan hasil analisis kebutuhan pengguna yang telah dilakukan pada tahap sebelumnya. Gambaran umum rancangan sistem ditampilkan pada Gambar di atas yang menunjukkan alur kerja mulai dari input sensor hingga keluaran sistem.



**Gambar 7.** Diagram Arsitektur Sistem Monitoring Ketinggian Air Berbasis ESP32

Bagan di atas menjelaskan hubungan antara **sensor ultrasonik**, **sensor hujan**, dan **GPS** yang mengirim data ke **ESP32** sebagai pusat kendali. ESP32 kemudian memproses dan mengirimkan data ke **web server**, yang diintegrasikan dengan **n8n workflow** untuk diteruskan ke **bot Telegram** sebagai notifikasi kepada pengguna. Desain pengembangan sistem meliputi dua bagian utama, yaitu perancangan perangkat keras dan perangkat lunak:

#### 1. Perancangan Alat

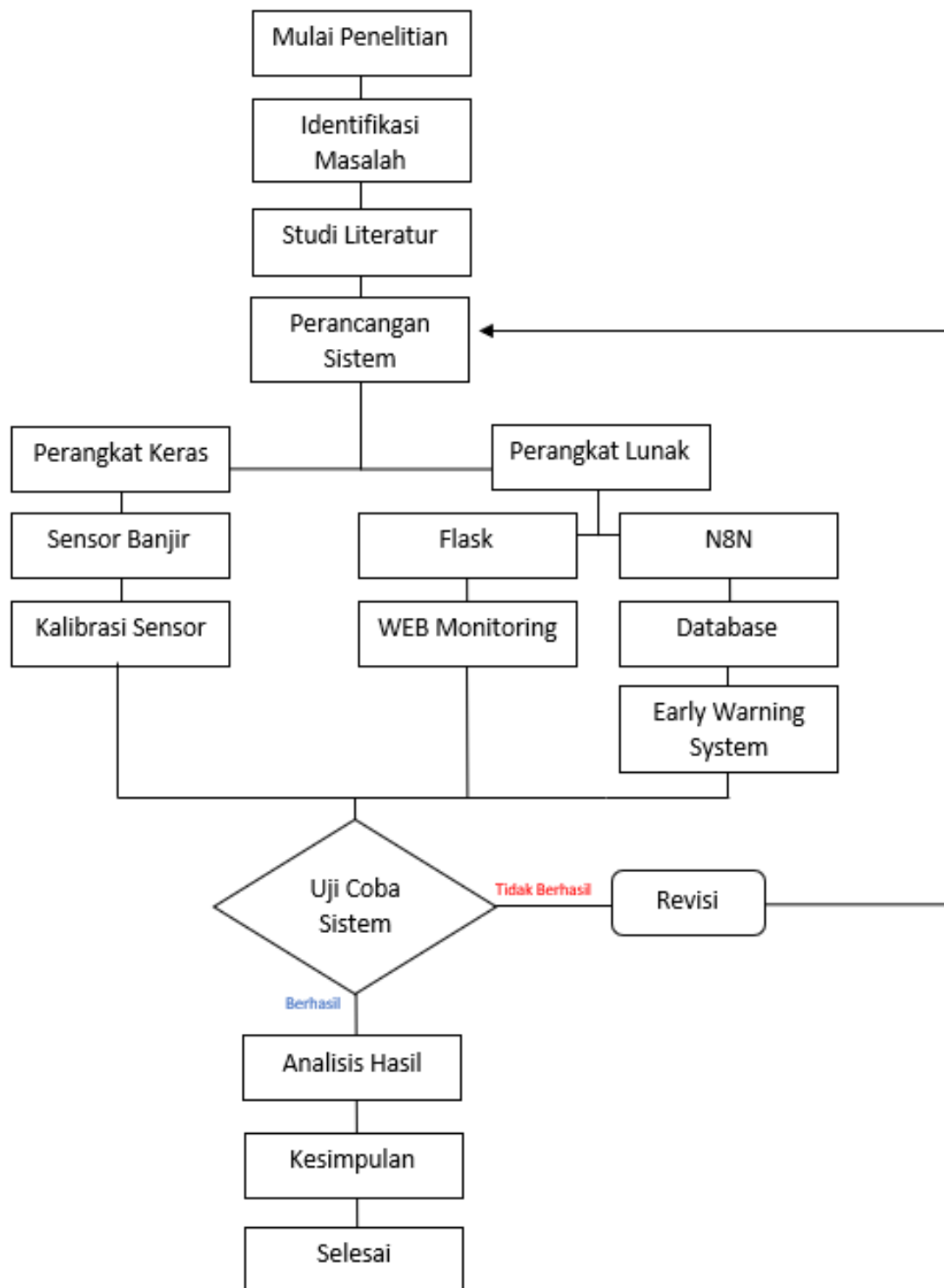
Perancangan perangkat keras menjadi komponen penting dalam pembangunan sistem. Mikrokontroler **ESP32** digunakan sebagai pusat kendali yang dihubungkan dengan **sensor ultrasonik HC-SR04** untuk mendeteksi ketinggian air, **sensor hujan FC-37** untuk mengetahui kondisi cuaca, serta **modul GPS NEO-6M** untuk menentukan koordinat lokasi. Data yang diperoleh dari sensor diproses oleh ESP32 kemudian ditampilkan melalui **LCD 16x2** serta diperkuat dengan **buzzer** sebagai alarm peringatan dini.

#### 2. Perancangan Aplikasi

Perangkat lunak dirancang untuk mendukung fungsi monitoring jarak jauh. Sistem ini menggunakan **Flask berbasis HTTP** sebagai server yang menerima data dari ESP32, lalu menampilkannya dalam bentuk dashboard web secara *real-time*. Selain itu, sistem diintegrasikan dengan **platform n8n** untuk menghubungkan server dengan **Telegram bot**. Melalui integrasi ini, informasi kondisi banjir dapat disampaikan langsung kepada pengguna melalui notifikasi otomatis di perangkat seluler.

### 2.2.3 Prosedur Pengembangan

Metode penelitian yang digunakan adalah metode eksperimen dengan pendekatan langsung di lapangan. Penelitian ini difokuskan pada pengujian sistem monitoring ketinggian air berbasis Internet of Things (IoT) dengan mikrokontroler ESP32 sebagai pusat kendali. Rentang ketinggian air yang diuji berada pada 2–25 cm, dengan ambang batas banjir ditetapkan pada 20 cm. Sistem dirancang menggunakan sensor ultrasonik HC-SR04 untuk mengukur ketinggian air, sensor hujan untuk mendeteksi kondisi curah hujan, serta modul GPS NEO-6M untuk menentukan koordinat lokasi pemasangan alat. Hasil pengukuran dikirimkan ke server Flask melalui jaringan WiFi dan ditampilkan secara *real-time* pada halaman web monitoring, serta dikirimkan sebagai notifikasi otomatis ke aplikasi Telegram melalui integrasi dengan platform n8n. Desain sistem alat pendeteksi banjir berbasis IoT dengan langkah-langkah sebagai berikut:

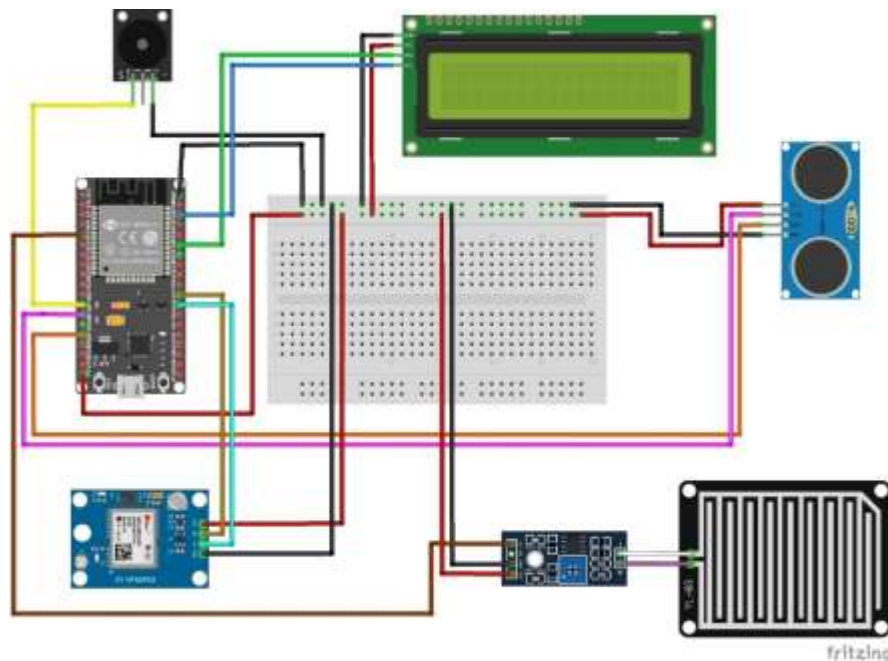


**Gambar 8.** Alur Kerja Sistem Pendeteksi Banjir

Bagan alir penelitian pada Gambar 8 menunjukkan tahapan yang dilakukan mulai dari identifikasi masalah, studi literatur, hingga perancangan sistem. Perancangan dibagi menjadi dua bagian utama, yaitu perangkat keras dan perangkat lunak. Perangkat keras meliputi sensor banjir dan proses kalibrasi, sedangkan perangkat lunak mencakup penggunaan Flask untuk web monitoring serta integrasi dengan n8n sebagai early warning system yang terhubung dengan database. Setelah sistem selesai dirancang, dilakukan uji coba untuk memastikan kinerjanya. Apabila uji coba tidak berhasil, dilakukan revisi hingga sistem berjalan sesuai harapan. Jika berhasil, penelitian dilanjutkan dengan analisis hasil, penyusunan kesimpulan, kalibrasi sensor akhir, dan diakhiri pada tahap penyelesaian.

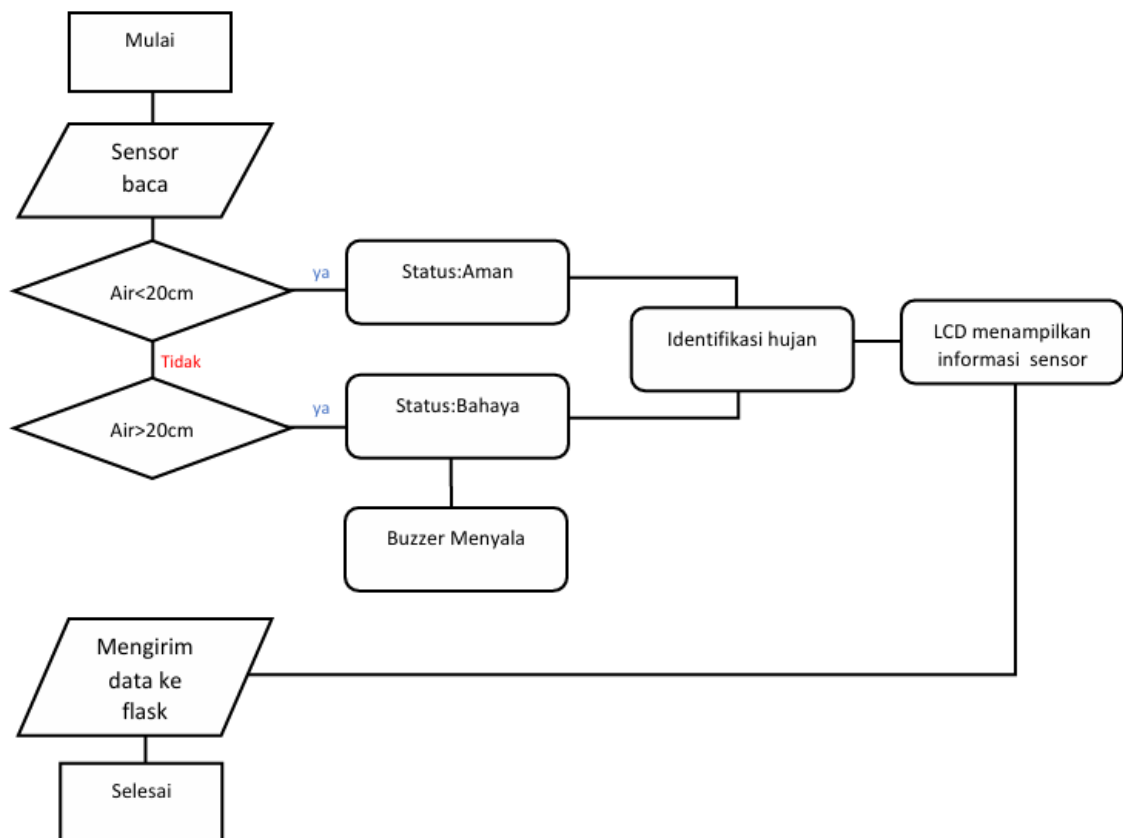
#### 2.2.4 Desain Perangkat Keras

Pada sistem ini, sensor ultrasonik HC-SR04 berperan untuk memantau ketinggian permukaan air dengan memanfaatkan pin *Trigger* dan *Echo* yang berfungsi mengirimkan serta menerima pantulan gelombang ultrasonik dari permukaan air. Jarak yang dihasilkan dari proses tersebut digunakan untuk menentukan tinggi muka air. Sensor hujan FC-37 berfungsi untuk mendeteksi adanya hujan, sehingga sistem dapat mengidentifikasi apakah peningkatan tinggi air disebabkan oleh curah hujan atau oleh faktor lain seperti limpasan dari daerah sekitarnya. Modul GPS digunakan untuk menentukan koordinat lokasi perangkat, sehingga ketika terjadi kenaikan muka air, data lokasi dapat diketahui secara akurat. Seluruh hasil pengukuran dari sensor ditampilkan pada LCD 16x2, yang menampilkan informasi berupa koordinat lokasi, tinggi muka air, serta kondisi cuaca (hujan atau tidak) secara *real-time*. Selain itu, sistem juga dilengkapi dengan buzzer yang akan memberikan peringatan suara apabila ketinggian air mencapai batas tertentu. Dalam penelitian ini, batas ambang yang ditetapkan untuk kondisi siaga atau berbahaya adalah tinggi air mencapai 20 cm dari dasar wadah. Dengan demikian, sistem ini tidak hanya mampu mengukur dan menampilkan data lingkungan, tetapi juga memberikan peringatan dini (*early warning*) terhadap potensi banjir berdasarkan kombinasi data tinggi air, curah hujan, dan lokasi pengukuran. Adapun perancangan perangkat keras dapat dilihat pada gambar sebagai berikut:



**Gambar 9.** Perancangan Alat

Gambar 9 menunjukkan konfigurasi keseluruhan rangkaian sistem. Berdasarkan gambar tersebut, dapat dijelaskan bahwa sistem terdiri dari beberapa komponen yang saling terhubung secara terpadu. Data yang diterima dari sensor ultrasonik, GPS, dan sensor hujan akan diproses oleh ESP32, kemudian hasilnya ditampilkan pada LCD.



**Gambar 10.** Bagan Alir Sistem Perancangan Alat

Bagan Alir sistem perancangan alat ini menggambarkan **alur kerja logika program dan proses pengambilan keputusan** pada sistem pemantauan ketinggian air berbasis Internet of Things (IoT). Flowchart berfungsi untuk menjelaskan urutan langkah kerja dari mulai sistem dinyalakan hingga data dikirimkan ke server dan ditampilkan pada LCD.

Proses dimulai ketika sistem diaktifkan (**Mulai**). Setelah itu, **sensor ultrasonik** akan membaca jarak antara sensor dan permukaan air untuk menentukan **tinggi muka air**. Hasil pembacaan tersebut kemudian dibandingkan dengan **batas ambang yang telah ditentukan**, yaitu **20 cm**.

#### 1. Kondisi Air < 20 cm (Aman)

Jika hasil pengukuran menunjukkan bahwa ketinggian air **kurang dari 20 cm**, maka sistem akan menetapkan status "**Aman**". Dalam kondisi ini, **buzzer tidak menyala**, dan sistem tetap menampilkan informasi sensor pada **LCD** berupa tinggi air, kondisi hujan, dan koordinat lokasi. Selain itu, sistem juga mengirimkan data ke server Flask untuk disimpan dalam database atau ditampilkan di halaman monitoring web.

#### 2. Kondisi Air > 20 cm (Bahaya)

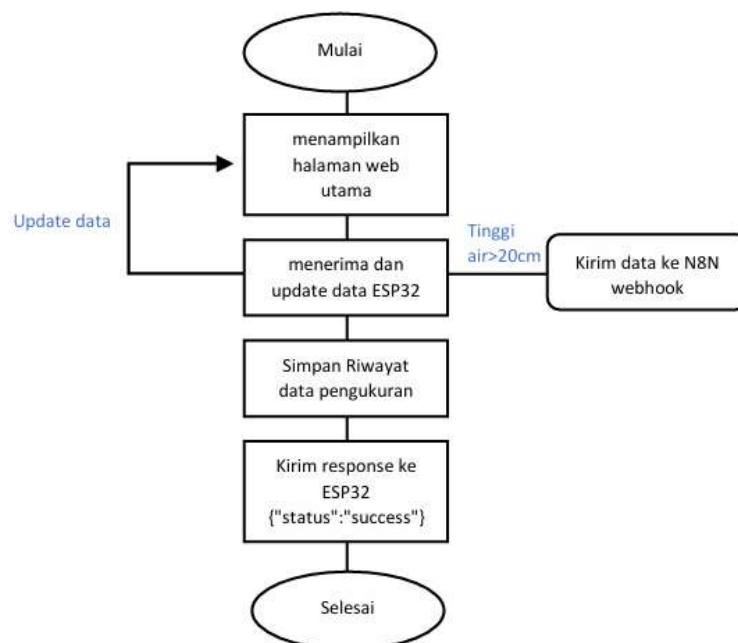
Jika hasil pengukuran menunjukkan bahwa ketinggian air **melebihi 20 cm**, maka sistem akan mengubah status menjadi "**Bahaya**". Pada kondisi ini, **buzzer akan menyala sebagai tanda peringatan**. Setelah itu, data hasil pengukuran tetap dikirimkan ke **server Flask**, dan sistem melakukan **identifikasi kondisi hujan** menggunakan sensor FC-37 untuk mengetahui apakah kenaikan muka air diakibatkan oleh curah hujan atau bukan. Selanjutnya, informasi ketinggian air, kondisi hujan, serta status bahaya akan ditampilkan pada **LCD** secara *real-time*.

Proses ini berlangsung secara **berulang (looping)** selama sistem masih menyala. Dengan alur kerja seperti pada flowchart, sistem dapat **memantau kondisi air secara otomatis, memberikan**

peringatan dini ketika air mencapai ambang batas bahaya, dan mengirimkan data ke server untuk keperluan pemantauan jarak jauh melalui jaringan internet.

### 2.2.5 Desain Perangkat Lunak

Perancangan perangkat lunak pada sistem pemantauan ketinggian air berbasis *Internet of Things* (IoT) menggunakan ESP32 melibatkan beberapa perangkat lunak pendukung, yaitu Arduino IDE, Visual Studio Code (VSC), Python (Flask), HTML, dan N8N *Workflow Automation*. Integrasi dari perangkat lunak ini bertujuan untuk membangun sistem yang dapat melakukan pemantauan, penyimpanan data, serta pengiriman notifikasi secara otomatis dan *real-time*. Perangkat lunak Arduino IDE digunakan untuk menulis dan mengunggah program ke mikrokontroler ESP32 menggunakan bahasa pemrograman C/C++. ESP32 berfungsi sebagai pusat kendali yang menerima data dari sensor ultrasonik, sensor hujan, dan modul GPS. Data hasil pembacaan sensor tersebut diolah kemudian dikirimkan melalui koneksi WiFi menuju web server yang dibangun menggunakan Flask di bahasa pemrograman Python.



Gambar 11. Bagan Alur Kerja Web Server

Alur kerja web server dapat dilihat pada **Gambar di atas**. Web server dikembangkan melalui **Visual Studio Code (VSC)** dengan kombinasi bahasa **Python (Flask)** dan **HTML**. Flask berperan sebagai *backend* yang menangani komunikasi data dari ESP32 sedangkan HTML digunakan untuk menampilkan antarmuka web yang berisi informasi tinggi air dan status kondisi cuaca secara *real-time*. Pembuatan server diawali dengan inisialisasi aplikasi Flask dan pengaktifan *Cross-Origin Resource Sharing (CORS)* agar ESP32 dapat berkomunikasi dengan server tanpa batasan domain kemudian menentukan alamat webhook yang digunakan untuk menghubungkan Flask dengan sistem **N8N**. Pada tahap pengujian, server dijalankan secara lokal. Data hasil pembacaan dari sensor disimpan dalam dua struktur utama, yaitu *latest\_data* untuk menyimpan data terbaru dan *history\_data* untuk merekam seluruh riwayat data selama sistem berjalan. Data ini mencakup informasi *latitude*, *longitude*, *tinggi\_air*, *status hujan*, serta waktu pengambilan data (*timestamp*). *Endpoint* berikutnya adalah */update*, yang menjadi pusat komunikasi utama antara ESP32 dan server. *Endpoint* ini menerima data sensor melalui metode **POST** dan kemudian memprosesnya. Apabila tinggi air yang diterima melebihi 20 cm, sistem akan menandai status banjir sebagai *True* dan secara otomatis mengirimkan data tersebut ke **N8N webhook** menggunakan metode *requests.post()*. Selain itu, server juga menyediakan *endpoint /history* untuk menampilkan seluruh riwayat data dalam format JSON, serta *endpoint /download* yang mengubah data tersebut ke dalam format **CSV** agar dapat diunduh untuk kebutuhan analisis lebih lanjut. Ketika sistem dijalankan, **ESP32** akan mengirimkan data ke *endpoint /update*. Server Flask akan menampilkan

data terbaru pada halaman HTML melalui /data, menyimpan seluruh riwayat ke dalam memori, dan mengirimkan notifikasi ke **N8N workflow** apabila kondisi banjir terdeteksi. Pengguna dapat memantau tinggi air melalui **web browser** secara *real-time* serta mengunduh riwayat data dalam bentuk file CSV untuk dokumentasi dan analisis.



**Gambar 12.** Bagan Alir N8N

Selanjutnya, **Gambar 12** menampilkan **proses pada sistem N8N Workflow**. Proses dimulai dengan *schedule trigger* yang dijalankan secara otomatis untuk mengambil data dari Flask melalui *endpoint* tertentu. Setelah data diterima, sistem melakukan pemeriksaan terhadap nilai tinggi air. Jika hasil pemeriksaan menunjukkan potensi banjir, data akan disimpan ke dalam **database MySQL** dan sistem akan mengirimkan notifikasi peringatan ke pengguna melalui **Bot Telegram**. Apabila kondisi masih aman, sistem tidak mengirimkan notifikasi dan menunggu jadwal pengambilan data berikutnya.



**Gambar 13.** Workflow N8N Early Warning System

Implementasi otomatisasi tersebut divisualisasikan pada Gambar 13 yang memperlihatkan tampilan *workflow* di platform N8N dan chat Bot telegram. *Workflow* ini terdiri dari beberapa node utama, yaitu *Schedule* Tiap 5 Detik, Ambil Data Flask, Cek Banjir, Simpan ke *Database MySQL*, dan Kirim Telegram. Masing-masing node bekerja berurutan untuk memastikan aliran data dari pengambilan hingga pengiriman pesan peringatan berjalan dengan baik. Node *Cek Banjir* berfungsi sebagai logika utama yang menentukan apakah sistem perlu menyimpan data dan mengirimkan peringatan banjir pada Bot telegram.

Dengan integrasi antara ESP32, Flask, HTML, dan N8N Workflow, sistem ini mampu melakukan proses pemantauan, penyimpanan, dan notifikasi secara otomatis tanpa perlu intervensi manual. Desain perangkat lunak ini memastikan pengguna dapat memperoleh informasi tinggi air secara *real-time* melalui *web monitoring* serta menerima peringatan dini melalui Telegram ketika kondisi air mulai melebihi ambang batas aman.

## 2.3 Teknik Analisis Data

### 1. Analisis Akurasi sensor

Analisis akurasi sensor dilakukan untuk mengetahui sejauh mana hasil pembacaan sensor mendekati nilai sebenarnya. Akurasi menunjukkan tingkat ketepatan sensor dalam menghasilkan data yang sesuai dengan kondisi nyata di lapangan. Pengujian akurasi biasanya dilakukan dengan membandingkan hasil pembacaan sensor dengan alat ukur pembanding (alat referensi) yang memiliki tingkat ketelitian lebih tinggi (Amrullah, 2022). Akurasi dapat dihitung dengan rumus:

$$\text{Akurasi} = \left[ 1 - \frac{\Delta x}{\bar{x}} \right] \cdot 100\% \quad (2)$$

dengan keterangan:

$\Delta x$  = Standar deviasi hasil pengukuran sensor,

$\bar{x}$  = Nilai rata-rata hasil pengukuran.

Nilai akurasi akan semakin tinggi apabila selisih antara hasil pengukuran sensor dengan nilai acuan semakin kecil. Nilai akurasi mendekati 100% menunjukkan bahwa sensor memiliki kemampuan pembacaan yang sangat baik dan dapat diandalkan untuk digunakan dalam sistem pemantauan ketinggian air.

### 2. Analisis Presisi sensor

Presisi menggambarkan tingkat konsistensi atau kestabilan hasil pengukuran yang dilakukan secara berulang pada kondisi yang sama. Semakin kecil variasi hasil pengukuran antar percobaan, semakin tinggi nilai presisinya (Amrullah, 2022). Berdasarkan persamaan (3), nilai presisi dapat dihitung menggunakan rumus:

$$\text{Presisi} = \left[ 1 - \frac{|H - \bar{x}|}{H} \right] \cdot 100\% \quad (3)$$

dengan keterangan:

H = Nilai sebenarnya

$\bar{x}$  = Nilai rata-rata hasil pengukuran sensor.

Nilai presisi yang tinggi menunjukkan bahwa sensor mampu menghasilkan pembacaan yang stabil dan seragam walaupun dilakukan pengukuran berulang-ulang. Analisis presisi ini penting untuk memastikan kestabilan sensor ultrasonik dalam mendeteksi perubahan ketinggian air secara konsisten.

### 3. Analisis Kesalahan Relatif Error sensor

Analisis kesalahan relatif bertujuan untuk mengetahui **besarnya tingkat kesalahan sensor terhadap nilai sebenarnya**. Nilai error ini menunjukkan seberapa jauh data hasil pengukuran sensor menyimpang dari data acuan (Priadi et al., 2025). Rumus yang digunakan untuk menghitung error relatif adalah:

Keterangan:

$$\text{Error Relatif \%} = \frac{\left( \frac{\sum f x}{n} \right) - s}{s} \cdot 100\% \quad (4)$$

$\Sigma x$  = Jumlah seluruh hasil pengukuran sensor

n = Jumlah percobaan yang dilakukan

s = Jarak nyata (diukur secara manual)

Semakin kecil nilai error relatif, maka semakin baik kinerja sensor yang digunakan. Analisis ini membantu untuk mengetahui seberapa besar kesalahan yang terjadi dalam pembacaan jarak oleh sensor ultrasonik terhadap nilai sebenarnya.