

**IMPLEMENTASI SISTEM PENDUKUNG KEPUTUSAN
BERBASIS IOT DENGAN METODE TOPSIS UNTUK
PERINGATAN DINI POTENSI BENCANA BANJIR**

SKRIPSI

DISUSUN OLEH

MIRANDA DEWI

NPM. 2109020013



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025**

**IMPLEMENTASI SISTEM PENDUKUNG KEPUTUSAN
BERBASIS IOT DENGAN METODE TOPSIS UNTUK
PERINGATAN DINI POTENSI BENCANA BANJIR**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
(S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu
Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

MIRANDA DEWI

NPM. 2109020013

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN**

2025

LEMBAR PENGESAHAN

Judul Skripsi : IMPLEMENTASI SISTEM PENDUKUNG
KEPUTUSAN BERBASIS IOT DENGAN METODE
TOPSIS UNTUK PERINGATAN DINI POTENSI
BENCANA BANJIR

Nama Mahasiswa : MIRANDA DEWI

NPM : 2109020013

Program Studi : TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing



(Yohanni Syahra S.Si., M.Kom.)
NIDN. 0129108201

Ketua Program Studi



(Fatma Sari Hutagalung S.Kom., M.Kom.)
NIDN. 0109039302

Dekan



(Dr. Al-Khrowafizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

**IMPLEMENTASI SISTEM PENDUKUNG KEPUTUSAN
BERBASIS IOT DENGAN METODE TOPSIS UNTUK
PERINGATAN DINI POTENSI BENCANA BANJIR**

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 17 Juli 2025

Yang membuat pernyataan



Miranda Dewi

NPM. 2109020013

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Miranda Dewi
NPM	: 2109020013
Program Studi	: Teknologi Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

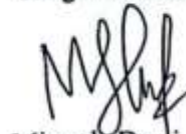
**Implementasi Sistem Pendukung Keputusan Berbasis IOT Dengan Metode
Topsis Untuk Peringatan Dini Potensi Bencana Banjir**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 17 Juli 2025

Yang membuat pernyataan



Miranda Dewi

NPM. 2109020013

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Miranda Dewi
Tempat dan Tanggal Lahir : Medan, 17 Juni 2003
Alamat Rumah : Dusun XIII JL- JATI LUHUR
Telepon/Faks/HP : 085372559478
E-mail : mirandadewi190@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD : Mis Ikhwanul Muslimin II TAMAT: 2015
SMP : Mts Al-Jam'iyatul Washliyah Tembung TAMAT: 2018
SMA : SMAN 11 Medan TAMAT: 2021

KATA PENGANTAR



Assalamualaikum Warrahmatullahi Wabarakatuh Alhamdulillah, Puji dan syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan penulis kesehatan dan kesempatan khususnya kepada penulis dalam menyelesaikan penyusunan skripsi ini dengan judul “Implementasi Sistem Pendukung Keputusan Berbasis IOT dengan Metode Topsis untuk Peringatan Potensi Bencana Banjir” sebagai salah satu syarat kelulusan untuk memperoleh gelar Sarjana pada Program Studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara. Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagaulung S.Kom., M.Kom. Ketua Program Studi Teknologi Informasi.
4. Bapak Muhammad Basri S.Kom., M.Kom. Sekretaris Program Studi Teknologi Informasi

5. Ibu Yohanni Syahra S.Si., M.Kom. Selakku dosen pembimbing yang telah dengan sabar membimbing, mengarahkan, dan memberikan masukan yang sangat berharga dalam proses penyusunan skripsi ini.
6. Seluruh dosen dan staf akademik khususnya di Program Studi Teknologi Informasi yang telah memberikan ilmu, pengetahuan, dan pengalaman selama masa studi.
7. Kedua orang tua yang saya sayangi dan yang saya cintai yaitu Bapak Martoni dan Ibu Dewi Murni yang selalu memberikan doa, motivasi, dan dukungan moral maupun material.
8. Kepada Diri Sendiri yang sudah selalu berusaha untuk menghadapi semua ini
9. Saudara kandung saya yaitu Nayla Syakira yang juga turut ikut memberikan doa, motivasi, dan masukan kepada saya.
10. Ahmad Al Qodri Azizi Dalimunthe selaku pasangan yang telah menemani saya dari mulai awal perkuliahan sampai dengan saat ini yang juga selalu memberikan saran, masukan kepada saya dan selalu sabar dengan saya.
11. Mara Hakim Nasution selaku abang sepupu saya yang telah memberikan saya saran, motivasi dan semangat untuk skripsi saya.
12. Kedua sahabat saya yaitu Nikma dan Bella yang sama sama lagi berjuang skripsi dan selalu menemani saya selama perkuliahan dari semester awal hingga akhir.

13. Teman teman seperjuangan saya “Yok nyusul Skripsi” yaitu nikma,bella,hana,rifdah, dan afridah yang juga sama sama berjuang untuk mendapatkan gelar S.Kom.
14. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa yang akan datang. Semoga skripsi ini dapat memberikan manfaat bagi pembaca dan dapat menjadi referensi bagi penelitian selanjutnya dan saya sangat berharap skripsi saya bisa menjadi landasan pendidikan terkait dengan *Internet of Things* (IOT).

Medan, 17 Juli 2025



Penulis

IMPLEMENTASI SISTEM PENDUKUNG KEPUTUSAN BERBASIS IOT DENGAN METODE TOPSIS UNTUK PERINGATAN DINI POTENSI BENCANA BANJIR

ABSTRAK

Banjir merupakan salah satu bencana alam yang dapat menyebabkan kerugian besar, baik secara material maupun terhadap keselamatan jiwa manusia. Oleh karena itu, dibutuhkan sistem yang mampu memberikan peringatan dini secara otomatis, akurat, dan real-time. Penelitian ini bertujuan untuk mengembangkan sistem monitoring potensi banjir berbasis *Internet of Things* (IoT) yang didukung oleh metode pengambilan keputusan TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*). Sistem ini dibangun menggunakan mikrokontroler ESP32 yang terhubung dengan beberapa sensor, yaitu sensor ultrasonik untuk mengukur tinggi muka air, DHT11 untuk mendeteksi suhu dan kelembapan udara, flow sensor untuk mengukur kecepatan aliran air, serta float sensor untuk mendeteksi level air. Data dari sensor dibaca secara berkala, kemudian dianalisis menggunakan metode TOPSIS untuk menentukan tingkat risiko yang diklasifikasikan ke dalam tiga kategori, yaitu Hujan Berhenti, Perlu Dipantau, dan Potensi Banjir. Sistem memberikan respon lokal melalui LED indikator dan *buzzer*, serta mengirimkan notifikasi ke pengguna melalui Telegram Bot jika terjadi perubahan status risiko. Hasil pengujian menunjukkan bahwa sistem dapat bekerja secara responsif dan *real-time*, dengan akurasi yang baik dalam mendeteksi perubahan kondisi lingkungan. Sistem mampu memberikan peringatan dini yang efektif dan dapat diakses dari jarak jauh. Dengan demikian, solusi ini berpotensi digunakan sebagai sistem mitigasi bencana banjir berbasis teknologi untuk mendukung keselamatan masyarakat dan pengambilan keputusan lebih cepat.

Kata Kunci: IoT, TOPSIS, Banjir, Sistem Peringatan Dini, ESP32, Telegram

IMPLEMENTATION OF IOT-BASED DECISION SUPPORT SYSTEM WITH TOPSIS METHOD FOR EARLY WARNING OF POTENTIAL FLOOD DISASTERS

ABSTRACT

Flooding is a natural disaster that can cause significant losses, both in terms of material and human safety. Therefore, a system capable of providing early warnings automatically, accurately, and in real time is needed. This research aims to develop an Internet of Things (IoT)-based flood monitoring system supported by the TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) decision-making method. This system is built using an ESP32 microcontroller connected to several sensors, namely an ultrasonic sensor to measure water level, a DHT11 to detect air temperature and humidity, a flow sensor to measure water flow velocity, and a float sensor to detect water level. Data from the sensors is read periodically, then analyzed using the TOPSIS method to determine the risk level which is classified into three categories: Rain Stopped, Needs to be Monitored, and Flood Potential. The system provides local responses through LED indicators and a buzzer, and sends notifications to users via Telegram Bot if there is a change in risk status. Test results show that the system can work responsively and in real time, with good accuracy in detecting changes in environmental conditions. The system is capable of providing effective early warnings and can be accessed remotely. Therefore, this solution has the potential to be used as a technology-based flood disaster mitigation system to support public safety and faster decision-making.

Keywords: IoT, TOPSIS, Flood, Early Warning System, ESP32, Telegram

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PERNYATAAN PERSETUJUAN PUBLIKASI	iv
RIWAYAT HIDUP	v
KATA PENGANTAR	vi
ABSTRAK	ix
DAFTAR ISI	xi
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II LANDASAN TEORI.....	7
2.1 Studi Literatur.....	7
2.2 <i>Internet Of Things</i> (IOT)	8
2.2.1 Sejarah <i>Internet Of Things</i> (IOT).....	10

2.2.2	Cara Kerja <i>Internet Of Things</i> (IOT)	11
2.3	Sistem Pendukung Keputusan	12
2.4	Metode TOPSIS	13
2.5	Peringatan Dini Banjir	14
2.6	Prototipe	14
2.7	Mikrokontroler	15
2.7.1	Fungsi Mikrokontroler	15
2.7.2	Cara kerja MikroKontroler	16
2.8	ESP32	17
2.8.1	Spesifikasi ESP32	18
2.8.2	Fitur ESP32.....	20
2.9	Sensor Ultrasonik	21
2.10	Float Sensor	21
2.11	<i>WaterFlow</i>	22
2.12	DHT11.....	23
2.13	<i>Light Emitting Diode</i> (LED).....	23
2.14	<i>Arduino IDE</i>	24
2.14.1	Jenis Fungsi <i>Arduino IDE</i>	25
2.14.2	Fitur Fitur <i>Arduino IDE</i>	26
2.15	Kabel <i>Jumper</i>	27
2.15.1	Jenis Kabel <i>Jumper</i>	28

2.16	<i>Telegram</i>	30
2.16.1	Keunggulan <i>Telegram</i>	31
2.16.2	Telegram Robot	32
BAB III METODOLOGI PENELITIAN		34
3.1	Jenis Penelitian	34
3.2	Metode Pengumpulan Data	34
3.2.1	Studi Pustaka.....	35
3.2.2	Observasi Sistem Simulasi.....	36
3.2.3	Jadwal Penelitian	36
3.2.4	Pengujian Prototipe.....	36
3.3	Penerapan Perancangan Alat Peringatan Dini Banjir.....	37
3.4	Alat dan Bahan	38
3.4.1	<i>Hardware</i>	38
3.4.2	Software	39
3.5	Metode Pengolahan Data.....	40
3.7	Proses Implementasi Alat.....	48
3.8	Diagram Flowchart sistem.....	49
BAB IV HASIL DAN PEMBAHASAN		51
4.1	Langkah Uji Coba	51
4.2	Proses Pembuatan alat Potensi Banjir	53
4.3	Perangkat Keras (<i>Hardware</i>).....	54

4.4 Perangkat Lunak (<i>Software</i>)	55
4.4.1 <i>Software</i> Program	55
4.4.2 Proses Membuat BOT Telegram	56
4.5 Pengoperasian Sistem.....	57
4.6 Proses Pengujian Sistem.....	58
4.7 Proses Pengujian Alat.....	59
4.7.1 Hasil Pengujian 150ml.....	59
4.7.3 Hasil Pengujian 500ml.....	61
4.7.3 Pengujian Sistem & Prediksi TOPSIS	62
4.7.4 Tampilan Notifikasi Telegram.....	64
BAB V PENUTUP.....	68
5.1 Kesimpulan.....	68
5.2 Saran.....	69
DAFTAR PUSTAKA	70
LAMPIRAN.....	74

DAFTAR TABEL

Tabel 2.1 Studi Literatur Penelitian Sebelumnya	7
Tabel 2.2 Fitur Fitur Arduino IDE	26
Tabel 2.3 Keunggulan Telegram.....	31
Tabel 3.1 Jadwal Penelitian.....	36
Tabel 3.2 Matriks Keputusan (Data Sensor)	41
Tabel 3.3 Hasil Normalisasi Matriks	42
Tabel 3.4 Hasil Matriks Keputusan Ternormalisasi.....	43
Tabel 3.5 Nilai Prefensi.....	45
Tabel 3.6 Kategori Resiko.....	46
Tabel 4. 1 Perangkat Keras Alat	54
Tabel 4.2 Hasil Uji Analisis TOPSIS.....	62

DAFTAR GAMBAR

Gambar 2.1 Internet Of Things (IOT).....	9
Gambar 2.2 ESP32	17
Gambar 2.3 Sensor Ultrasonik	21
Gambar 2. 4 Float Sensor.....	21
Gambar 2.5 Water Flow Sensor	22
Gambar 2.6 DH11	23
Gambar 2.7 LED	23
Gambar 2.8 Arduino IDE.....	24
Gambar 2.9 Kabel Jumper.....	27
Gambar 2.10 Kabel Jumper Male to Male	28
Gambar 2.11 Kabel Jumper Male to Female	29
Gambar 2.12 Kabel Jumper Female to Female.....	29
Gambar 2.13 Telegram.....	30
Gambar 3.1 Diagram Waterfall.....	37
Gambar 3.2 Diagram blok sistem	47
Gambar 3. 3 Flowchart Sistem.....	49
Gambar 4.1 Perancangan alat Potensi Banjir.....	54
Gambar 4.2 Kode Program Alat Sensor.....	55
Gambar 4.3 Hasil Uji Coba Alat Sensor	55
Gambar 4.4 BOT Father.....	56
Gambar 4.5 Membuat NamaBOT	56
Gambar 4.6 ID User	57

Gambar 4.7 Nama BOT	57
Gambar 4.8 Data Pengujian 150ml Air.....	59
Gambar 4.9 Data Pengujian 300ml air.....	60
Gambar 4.10 Data Pengujian 500ml air.....	61
Gambar 4.11 Notifikasi Telegram.....	65

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Banjir merupakan bencana alam yang signifikan yang menyebabkan kerusakan besar pada properti dan kehidupan manusia di seluruh dunia. Indonesia merupakan salah satu negara yang rawan terhadap bencana banjir. Sepanjang tahun 2024, beberapa bencana banjir besar telah terjadi, seperti banjir akibat Siklon Tropis Robyn yang melanda Sumatra Utara, Sumatra Barat, Aceh, Jawa Tengah, Jawa Timur, dan Jawa Barat, menyebabkan 41 korban jiwa dan kerusakan infrastruktur yang luas (Binsar Bakkara, 2024). Selain itu, banjir di Maluku Utara pada Agustus 2024 menyebabkan 13 korban jiwa dan merusak 10 rumah (Reuters, 2024). Di Pulau Jawa, khususnya di Kabupaten Sukabumi, banjir dan tanah longsor pada Desember 2024 mengakibatkan 10 korban jiwa, 2 orang hilang, serta lebih dari 3.000 warga mengungsi (Ninie Karmini & Rangga Firmansyah, 2024).

Sistem pemantauan banjir tradisional memiliki keterbatasan dalam menyediakan informasi yang tepat waktu dan akurat, yang dapat menghambat upaya penanggulangan bencana. Kurangnya data akurat dan real-time tentang ketinggian air dan faktor lingkungan lainnya membuat para pengambil keputusan sulit mengambil tindakan cepat, yang menyebabkan peningkatan risiko dan kerusakan yang terkait dengan banjir (Kumar et al., 2023).

Perkembangan teknologi *Internet of Things* (IoT) membuka peluang besar dalam membangun sistem pemantauan lingkungan yang lebih cerdas dan responsif. Dalam sistem ini, berbagai sensor IoT digunakan untuk mengumpulkan data secara *real-time*, seperti kondisi cuaca (sensor *DHT11*), ketinggian air (sensor *ultrasonik*

dan *Float* sensor), serta kecepatan aliran sungai (*flow* sensor). Data yang diperoleh dari sensor-sensor ini dikirim melalui jaringan komunikasi IoT ke sistem pusat untuk dianalisis. Namun, agar data yang dikumpulkan dapat memberikan rekomendasi yang akurat dalam mendeteksi potensi banjir, diperlukan metode pengambilan keputusan yang tepat, seperti *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS), untuk mengolah data dan menentukan tingkat risiko banjir. Apabila tingkat risiko banjir telah teridentifikasi, sistem akan secara otomatis mengirimkan notifikasi melalui *Telegram* kepada masyarakat.

Metode TOPSIS (*Technique for Order of Preference by Similarity to Ideal Solution*) merupakan salah satu metode pengambilan keputusan multi-kriteria yang banyak digunakan di berbagai bidang, termasuk dalam pemilihan alternatif terbaik dari sejumlah opsi berdasarkan beberapa kriteria yang ditentukan. Metode ini berfokus pada prinsip bahwa alternatif terbaik adalah yang memiliki jarak terdekat dari solusi ideal positif dan terjauh dari solusi ideal negatif (Efendy et al., 2023). Penggunaan metode TOPSIS dalam sistem pendukung keputusan memungkinkan pemangku kepentingan untuk membuat keputusan yang lebih objektif dan terukur, karena mengandalkan analisis numerik yang mendetail.

Dalam aplikasinya, metode TOPSIS telah terbukti efektif dalam berbagai studi kasus. Sebagai contoh, penelitian oleh Zainul Efendy et al. (2023) menunjukkan bahwa metode ini dapat digunakan untuk menentukan keluarga miskin yang berhak menerima bantuan dari pemerintah, dengan menghasilkan peringkat preferensi yang jelas. Selain itu, dalam penelitian oleh Edy Prayitno dan Rachmat Ardian Prayoga Putra (2023), metode TOPSIS diterapkan untuk menentukan penerima beasiswa pendidikan dengan presisi yang lebih tinggi

dibandingkan dengan cara tradisional. Demikian juga, Yayas Husni Mubarak et al. (2024) menunjukkan bahwa metode ini dapat membantu dalam pemilihan varietas bibit padi unggul untuk meningkatkan produktivitas pertanian, yang sangat penting bagi ketahanan pangan.

Metode TOPSIS menghadirkan pendekatan yang sistematis dan transparan dalam pengambilan keputusan yang dapat diterapkan di berbagai sektor, dari sosial hingga ekonomi, dan turut berkontribusi pada pengembangan sistem dukungan keputusan yang lebih efisien dan manfaat dalam konteks praktis (Snyders et al., 2024)

1.2 Rumusan Masalah

Perumusan Masalah Penelitian ini ialah :

1. Bagaimana merancang sistem pemantauan banjir berbasis IOT yang mampu mengumpulkan data secara real time ?
2. Bagaimana mengimplementasikan metode topsis dalam sistem pendukung keputusan untuk menentukan tingkat resiko banjir ?
3. Bagaimana mengevaluasi eektivitas sistem prototipe dalam mendeteksi potensi banjir dan mengirim notifikasi peringatan melalui telegram?

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Lingkup penelitian dibatasi pada simulasi menggunakan prototipe miniatur sistem pemantauan banjir.
2. Data yang dikumpulkan hanya mencakup beberapa parameter, yaitu: Kondisi Cuaca (DHT11), ketinggian air (sensor *ultrasonik* dan *float* sensor), serta kecepatan aliran air (*flow* sensor).

3. Metode yang digunakan dalam pengambilan keputusan hanya menggunakan TOPSIS.
4. Notifikasi peringatan hanya dikirim melalui *Telegram*.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk :

1. Merancang dan mengembangkan prototipe sistem pemantauan banjir berbasis *Internet of Things* (IoT) yang mampu mengumpulkan data secara *real-time* untuk dianalisis.
2. Menerapkan metode TOPSIS dalam pengolahan data untuk menentukan tingkat risiko banjir secara objektif.
3. Menguji sistem peringatan dini yang dapat mengirim notifikasi otomatis melalui Telegram berdasarkan hasil analisis data.
4. Menganalisis kinerja prototipe dalam memberikan peringatan dini berdasarkan data lingkungan yang dikumpulkan.

1.5 Manfaat Penelitian

Hasil Penelitian ini diharapkan memberikan manfaat sebagai berikut :

1. Bagi Peneliti

- a. Memberikan pengalaman dalam merancang dan mengembangkan sistem IoT berbasis pengambilan keputusan menggunakan metode TOPSIS
- b. Menyediakan studi kasus tentang penggunaan metode TOPSIS dalam pengolahan data sensor IoT, Sehingga dapat digunakan dalam penelitian serupa yang melibatkan analisis *multi-kriteria*.

- c. Memperluas wawasan dalam pengembangan sistem peringatan dini berbasis IoT, yang berpotensi diadaptasi untuk deteksi bencana lainnya, termasuk tanah longsor, gempa bumi, maupun kebakaran hutan.

2. Bagi Masyarakat

- a. Meningkatkan kesiapsiagaan masyarakat dalam menghadapi potensi bencana banjir, sehingga dapat mengurangi risiko korban jiwa dan kerugian material.
- b. Memudahkan akses informasi terkait potensi banjir melalui notifikasi otomatis via Telegram, sehingga masyarakat dapat menerima peringatan dini secara cepat dan akurat.
- c. Membantu menciptakan lingkungan yang lebih aman dengan meningkatkan kesadaran masyarakat terhadap perubahan kondisi cuaca dan ketinggian air sungai.
- d. Memberikan solusi berbasis teknologi dalam deteksi dini banjir, yang dapat membantu masyarakat di daerah rawan untuk mengambil tindakan pencegahan lebih awal.

3. Bagi Pendidikan

- a. Memberikan contoh penerapan nyata dari konsep *Internet of Things* (IoT), sistem pendukung keputusan (SPK), dan kecerdasan buatan dalam mitigasi bencana.
- b. Meningkatkan keterampilan mahasiswa dalam pemrograman mikrokontroler (ESP8266, ESP32, atau Arduino), pengolahan data sensor, serta integrasi dengan platform komunikasi seperti *Telegram*.

BAB II

LANDASAN TEORI

2.1 Studi Literatur

Tabel 2.1 Studi Literatur Penelitian Sebelumnya

NO	Nama Peneliti/Tahun	Judul Penelitian	Metode	Fokus Penelitian	Perbedaan
1	Sapta Suhardono, Moh Rizal Ngambah Sagara & I Wayan Koko Suryawan ,2024	Sistem Peringatan Dini untuk Banjir Rob dan Sampah Laut: Analisis SWOT.	SWOT	Kajian risiko dan kebijakan banjir rob dan pencemaran laut	Tidak menggunakan sistem atau algoritma pengambilan keputusan. Hanya analisis kualitatif SWOT.
2	Dirga Rama Setiadi, Ernando Rizki Dalimunthe, Novia Utami Putri & Alfath Zain, 2024	Implementasi Metode Fuzzy Sugeno Pada Prototipe Pendeteksi Banjir	Fuzzy Sugeno	Deteksi ketinggian air dan klasifikasi risiko banjir	Sama-sama IoT dan real-time, namun menggunakan metode fuzzy

3	Mufti Ari Bianto & Mala Rosa Aprillya,2023	Sistem Pendukung Keputusan Identifikasi Daerah Potensi Banjir Dengan Metode Multi Attribute Utility Theory (Studi Kasus: Kabupaten Lamongan)	MAUT	Identifikasi wilayah rawan banjir	Tidak menggunakan sensor atau sistem IoT. Berbasis data sekunder/statik dan visualisasi lokasi.
4	Andi Muh Sabran & Untung Suwardoyo,2024	Prototipe Notifikasi Banjir Berbasis IoT	-	Pengiriman notifikasi banjir secara real-time berdasarkan pembacaan ketinggian air	Hanya membaca data dan mengirim notifikasi tanpa metode perhitungan atau klasifikasi multi- kriteria

2.2 *Internet Of Things (IOT)*

Menurut *Selay et al (2022)*, Internet of Things (IoT) adalah teknologi canggih yang bertujuan untuk memperluas dan meningkatkan manfaat konektivitas internet yang terus terhubung. Teknologi ini memungkinkan berbagai objek di sekitar kita saling terhubung, sehingga aktivitas sehari-hari menjadi lebih mudah

dan efisien, serta membantu berbagai pekerjaan manusia. Pentingnya IoT terlihat dari semakin luasnya penerapan teknologi ini dalam berbagai aspek kehidupan. Berdasarkan metode identifikasi RFID (*Radio Frequency Identification*), IoT termasuk dalam kategori metode komunikasi. Namun, IoT juga mencakup berbagai teknologi lain, seperti sensor, jaringan nirkabel, dan kode QR (*Quick Response*).

Menurut *Boimau, Hapsari, dan Lestari (2024)*, IoT memiliki peran penting dalam meningkatkan literasi sains dan teknologi pada siswa sekolah dasar, dengan cara memperkenalkan konsep dan penerapan IoT melalui kegiatan praktis yang melibatkan interaksi langsung dengan alat-alat teknologi. Hal ini tidak hanya meningkatkan pemahaman dasar tentang bagaimana perangkat tersebut berfungsi, tetapi juga memperkuat keterampilan praktis dalam merakit dan mengoperasikan perangkat IoT. Pengenalan sejak dini ini menunjukkan bahwa IoT tidak hanya diterapkan di sektor industri dan profesional, tetapi juga menjadi bagian dari pendidikan berbasis teknologi yang membentuk generasi yang melek digital dan siap menghadapi era otomatisasi.



Gambar 2.1 Internet Of Things (IOT)

(Sumber: <https://docif.telkomuniversity.ac.id/apa-itu-iot/>)

2.2.1 Sejarah *Internet Of Things* (IOT)

Pada awal tahun 1980-an, perangkat pertama yang terhubung ke internet adalah *Vending Machine Coca-Cola* yang dioperasikan oleh para programmer di *Carnegie Mellon University*. Mereka menambahkan *micro switches* pada mesin tersebut dan memanfaatkan internet untuk memantau suhu pendingin serta mengecek ketersediaan stok minuman. Inovasi ini kemudian menginspirasi pengembangan lebih lanjut di berbagai bidang di seluruh dunia. Pada tahun 1990, John Romkey menciptakan perangkat pemanggang roti yang dapat dikendalikan melalui internet. Setahun kemudian, ilmuwan dari *University of Cambridge* mengembangkan konsep penggunaan *web camera* pertama untuk mengamati jumlah kopi di sebuah ruangan, dengan kamera yang mengambil gambar tiga kali per menit dan mengirimkannya ke komputer dalam jaringan lokal agar pengguna dapat mengetahui ketersediaan kopi. Pada tahun 1994, Steve Mann menciptakan *Wear Cam*, sebuah perangkat kamera yang dapat dikenakan.

Pada tahun 1997, Paul Saffo memberikan pandangan awal tentang perkembangan sensor dan perannya di masa depan. Istilah *Internet of Things* (IoT) sendiri baru diperkenalkan pada tahun 1999 oleh Kevin Ashton, seorang anggota Radio Komunitas sekaligus direktur *eksekutif Auto ID Centre*, MIT. Ashton mengembangkan konsep *Radio Frequency Identification (RFID)* yang menjadi dasar bagi IoT modern. Dalam perkembangannya, IoT semakin relevan dengan kehidupan sehari-hari berkat pertumbuhan pesat perangkat seluler, komunikasi tertanam, komputasi awan, serta analisis data yang memungkinkan integrasi dan otomatisasi yang lebih luas (Mubarok et al., n.d.).

Pengembangan IoT adalah respons terhadap kebutuhan untuk mengoptimalkan sistem yang ada dengan menggunakan data dari berbagai perangkat yang saling terhubung. Pendekatan ini tidak hanya meningkatkan efisiensi, tetapi juga memberikan wawasan yang berharga dalam pengambilan keputusan berbasis data di semua sektor. Secara keseluruhan, sejarah IoT menunjukkan perjalanan yang kompleks dan berkembang, mencerminkan inovasi teknologi yang cepat dan adaptasi terhadap kebutuhan masyarakat modern.

2.2.2 Cara Kerja *Internet Of Things* (IoT)

Menurut Ayu Syahfitri (2025) *Internet of Things (IoT)* adalah konsep yang memungkinkan perangkat fisik terhubung ke internet untuk berkomunikasi dan bertukar data secara otomatis. Cara kerja IoT melibatkan beberapa lapisan utama, mulai dari lapisan perangkat fisik yang terdiri dari sensor dan aktuator untuk mendeteksi perubahan lingkungan, lapisan jaringan yang mengirimkan data melalui protokol komunikasi seperti *Wi-Fi*, *Bluetooth*, atau jaringan seluler, hingga lapisan pengolahan data yang memproses informasi menggunakan *cloud computing* atau *edge computing*. Data yang telah diproses kemudian ditampilkan melalui lapisan aplikasi dalam bentuk *dashboard* atau aplikasi mobile, memungkinkan pengguna untuk memantau dan mengontrol perangkat IoT secara *real-time*. Dalam sistem IoT, protokol komunikasi seperti *Message Queuing Telemetry Transport* (MQTT) dan *Constrained Application Protocol* (CoAP) digunakan untuk memastikan efisiensi dan kecepatan dalam pengiriman data.

Penerapan IoT telah meluas ke berbagai sektor, termasuk *smart home*, kesehatan, industri, pertanian, dan transportasi. Dalam *smart home*, perangkat seperti smart lighting dan smart thermostat meningkatkan efisiensi energi,

sementara dalam bidang kesehatan, *smartwatch* dan biosensor memungkinkan pemantauan kondisi pasien secara real-time. Di sektor industri, *Industrial IoT* (IIoT) diterapkan untuk otomatisasi produksi dan *predictive maintenance*, sedangkan dalam pertanian, teknologi IoT digunakan dalam smart farming untuk memantau irigasi dan kondisi tanah. Namun, tantangan utama dalam implementasi IoT adalah keamanan, seperti ancaman *Distributed Denial of Service* (DDoS) dan pencurian data. Oleh karena itu, diperlukan langkah-langkah seperti enkripsi data, autentikasi yang kuat, serta penerapan *firewall* dan *Intrusion Detection/Prevention System* (IDS/IPS) untuk meningkatkan keamanan IoT. Dengan perkembangan teknologi seperti kecerdasan buatan (AI), *blockchain*, dan jaringan 6G, IoT diprediksi akan semakin berkembang, meskipun tantangan dalam standarisasi protokol, privasi, dan interoperabilitas antar perangkat masih perlu diatasi untuk memastikan pertumbuhan yang optimal dan berkelanjutan (Ayu Syahfitri, 2025).

2.3 Sistem Pendukung Keputusan

Menurut *Setiawan & Wiharko, (2023)* Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS) adalah sistem berbasis teknologi yang dirancang untuk membantu menyelesaikan permasalahan kompleks dengan menyediakan solusi yang relevan. SPK berfungsi sebagai alat bantu dalam proses pengambilan keputusan dengan mempertimbangkan berbagai faktor dan kriteria yang dimasukkan ke dalam sistem. Tingkat kepentingan serta kondisi dari setiap kriteria akan memengaruhi hasil analisis yang diberikan, sehingga sistem ini dapat digunakan untuk meningkatkan efektivitas dan akurasi dalam pengambilan keputusan di berbagai bidang.

Dalam konteks manajemen bencana, seperti peringatan dini banjir, SPK dapat membantu pemangku kepentingan menyusun strategi mitigasi dan respons secara efektif berdasarkan data meteorologi, hidrologi, serta data historis banjir. Contoh dari penerapan SPK dalam peringatan dini banjir yang mengembangkan sistem berbasis web menggunakan metode *Technique for Others Reference by Similarity to Ideal Solution* (TOPSIS) untuk menentukan calon penerima bantuan bencana alam. Hasil penelitian ini menunjukkan bahwa penerapan metode TOPSIS dapat memberikan rekomendasi yang akurat dalam menentukan penerima bantuan, yang juga merupakan langkah penting dalam pengelolaan situasi darurat seperti banjir (Hadrianti A & Wajidi, 2023)

2.4 Metode TOPSIS

Menurut *Rahmansyah et al., (2021)* TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) adalah salah satu metode pengambilan keputusan dengan banyak kriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang. Metode ini didasarkan pada prinsip bahwa alternatif yang dipilih seharusnya memiliki jarak terdekat dengan solusi ideal positif dan jarak terjauh dari solusi ideal negatif. Untuk menentukan kedekatan relatif suatu alternatif terhadap solusi optimal, TOPSIS menggunakan jarak Euclidean sebagai ukuran geometris. Metode ini banyak digunakan karena memiliki konsep yang sederhana dan mudah dipahami, proses perhitungannya efisien, serta mampu mengukur kinerja relatif dari berbagai alternatif keputusan dengan model matematis yang sederhana. Keunggulan-keunggulan tersebut menjadikan TOPSIS sebagai metode yang efektif dalam mendukung proses pengambilan keputusan di berbagai bidang.

2.5 Peringatan Dini Banjir

Peringatan dini banjir merupakan sistem yang dirancang untuk mendeteksi, memantau, dan memberikan informasi terkait potensi ancaman banjir sebelum terjadi. Tujuan utama dari sistem ini adalah meminimalkan risiko serta mengurangi kerugian material dan korban jiwa dengan memberikan peringatan secara cepat kepada masyarakat di wilayah rawan banjir. Dengan adanya sistem ini, masyarakat dapat mengambil langkah-langkah preventif seperti evakuasi dini, pengamanan barang berharga, serta persiapan peralatan darurat guna mengurangi dampak bencana (Soselisa & Dasawaty, 2023).

Dalam pengembangannya, sistem peringatan dini banjir dapat menggunakan metode prototipe, yang memungkinkan pengembangan sistem secara bertahap berdasarkan umpan balik dari pengguna. Sistem ini dapat terdiri dari sensor ketinggian air, kondisi cuaca, kecepatan aliran air serta notifikasi berbasis telegram dan *website* sebagai *database* atau sirine peringatan untuk menyampaikan informasi kepada masyarakat secara cepat. Data yang dikumpulkan dari sensor akan dikirim ke pusat pemantauan, di mana operator dapat menganalisisnya dan memberikan peringatan jika batas ketinggian air atau curah hujan yang telah ditentukan terlampaui. Dengan pendekatan berbasis prototipe, pengembang dapat mengadaptasi dan menyempurnakan sistem berdasarkan evaluasi langsung di lapangan, sehingga sistem yang dihasilkan lebih sesuai dengan kebutuhan pengguna dan kondisi lingkungan setempat.

2.6 Prototipe

Prototipe peringatan dini banjir merujuk pada sistem yang dirancang untuk mendeteksi dan memberikan peringatan kepada masyarakat tentang potensi banjir

sebelum kejadian tersebut terjadi. Prototipe ini umumnya dilengkapi dengan berbagai sensor yang mampu memonitor tingkat air, kecepatan aliran, dan faktor lingkungan lainnya, serta menggunakan teknologi komunikasi untuk menginformasikan kepada individu atau komunitas yang berisiko. Dengan demikian, tujuan dari prototipe ini adalah untuk meningkatkan kesiapsiagaan masyarakat dalam mengantisipasi bencana, sehingga dapat mengurangi risiko kerugian yang ditimbulkan oleh banjir.

2.7 Mikrokontroler

Menurut (Ramu *et al.*, 2022) Mikrokontroler adalah sirkuit terpadu yang dirancang untuk mengendalikan perangkat tertentu secara otomatis. Mereka mengumpulkan input, memproses informasi, dan menghasilkan tindakan berdasarkan data yang dikumpulkan.

2.7.1 Fungsi Mikrokontroler

Secara umum, perangkat atau sistem yang memanfaatkan mikrokontroler sebagai unit pemrosesan data dikenal sebagai *embedded system* atau *dedicated system*. *Embedded system* merujuk pada sistem kontrol yang terintegrasi dalam suatu perangkat atau sistem tertentu, sedangkan *dedicated system* adalah sistem kendali yang dirancang khusus untuk menjalankan satu fungsi tertentu. Misalnya, printer dikategorikan sebagai *embedded system* karena memiliki mikrokontroler di dalamnya untuk mengontrol operasionalnya. Pada saat yang sama, printer juga termasuk *dedicated system* karena mikrokontrolernya hanya berfungsi untuk menerima data dan mencetak.

Mikrokontroler memiliki beberapa fungsi utama, di antaranya:

- a. Sebagai timer untuk mengatur waktu dalam suatu sistem.

- b. Sebagai pembangkit osilasi guna menghasilkan sinyal osilasi yang dibutuhkan.
- c. Sebagai *flip-flop* untuk menyimpan status bit dalam operasi logika.
- d. Sebagai ADC (*Analog to Digital Converter*) yang berfungsi mengubah sinyal analog menjadi digital.
- e. Sebagai *counter* atau penghitung dalam berbagai proses numerik.
- f. Sebagai *decoder* dan *encoder* yang digunakan dalam proses pengkodean dan penguraian data.

2.7.2 Cara kerja MikroKontroler

Mikrokontroler bekerja berdasarkan prinsip tertentu agar setiap fungsi di dalamnya dapat berjalan secara optimal. Semua komponen di dalamnya telah dirancang untuk saling terintegrasi sehingga dapat membentuk sebuah sistem kontrol yang efisien. Dengan adanya integrasi ini, mikrokontroler mampu mengontrol perangkat elektronik sesuai dengan tugas yang telah diprogram sebelumnya.

Cara kerja mikrokontroler bergantung pada program yang telah dimasukkan ke dalamnya. Program ini disimpan dalam ROM (*Read-Only Memory*), yang berperan sebagai media penyimpanan instruksi atau perintah yang akan dijalankan. ROM memastikan bahwa program tetap tersedia meskipun perangkat dimatikan, sehingga mikrokontroler dapat bekerja secara konsisten sesuai dengan perintah yang telah ditetapkan.

Ketika beroperasi, mikrokontroler akan mengeksekusi instruksi yang tersimpan dalam ROM. Instruksi-instruksi ini dapat berupa membaca data dari sensor, melakukan perhitungan logis atau matematis, serta mengubah data dari satu

bentuk ke bentuk lain. Sebagai contoh, mikrokontroler dapat mengubah data analog dari sensor menjadi data digital untuk diproses lebih lanjut atau mengaktifkan aktuator sesuai hasil perhitungan tertentu. Dengan proses kerja yang sistematis, mikrokontroler mampu mengendalikan berbagai perangkat elektronik dan sistem otomatisasi dengan efisiensi tinggi.

2.8 ESP32



Gambar 2.2 ESP32

(Sumber: <https://www.aranacorp.com/en/programming-an-esp32-nodemcu-with-the-arduino-ide/amp/>)

ESP32 merupakan mikrokontroler yang dikembangkan oleh *Espressif Systems*, sebuah perusahaan teknologi asal Tiongkok yang berfokus pada solusi komunikasi nirkabel. Sebagai penerus dari ESP8266, ESP32 hadir dengan peningkatan yang signifikan dalam hal performa, fitur, serta efisiensi daya. Mikrokontroler ini mendukung konektivitas *Wi-Fi* dan *Bluetooth* secara bersamaan, menjadikannya pilihan ideal untuk berbagai aplikasi *Internet of Things* (IoT).

ESP32 dilengkapi dengan *prosesor single-core* atau *dual-core Tensilica Xtensa LX6*, serta mendukung berbagai antarmuka komunikasi seperti SPI, I2C, UART, ADC, DAC, dan PWM. Selain itu, perangkat ini memiliki fitur keamanan

yang lebih canggih dibandingkan generasi sebelumnya. ESP32 juga kompatibel dengan berbagai platform pengembangan, seperti *Arduino IDE*, *MicroPython*, dan ESP-IDF (*Espressif IoT Development Framework*), sehingga memudahkan para pengembang dalam merancang dan mengimplementasikan proyek berbasis IoT.

ESP32 lahir sebagai pengembangan dari kesuksesan ESP8266, yang diperkenalkan oleh *Espressif Systems* pada 2014. ESP8266 menjadi sangat populer di kalangan pengembang karena ukurannya yang ringkas, harga yang terjangkau, serta kemampuannya dalam menghubungkan perangkat ke jaringan *Wi-Fi* dengan mudah. Namun, ESP8266 memiliki beberapa keterbatasan, seperti hanya menggunakan prosesor *single-core*, jumlah GPIO yang terbatas, serta tidak memiliki dukungan Bluetooth.

Untuk mengatasi keterbatasan tersebut, *Espressif Systems* merilis ESP32 pada September 2016. Mikrokontroler ini menghadirkan banyak peningkatan, seperti *prosesor dual-core*, dukungan *Bluetooth (BLE dan Bluetooth Classic)*, lebih banyak GPIO, serta fitur keamanan yang lebih mutakhir, termasuk enkripsi perangkat keras dan *secure boot*.

Sejak peluncurannya, ESP32 terus berkembang dengan berbagai varian, seperti ESP32-S2, ESP32-C3, dan ESP32-S3, yang dirancang untuk memenuhi kebutuhan pengembang dalam berbagai proyek IoT. Dengan performa tinggi, fleksibilitas konektivitas, dan efisiensi daya, ESP32 kini menjadi standar utama dalam pengembangan perangkat pintar, termasuk otomasi rumah, pemantauan lingkungan, sistem keamanan, serta berbagai aplikasi IoT lainnya.

2.8.1 Spesifikasi ESP32

Berikut adalah spesifikasi utama dari ESP32:

1. Prosesor :

- a. Tensilica Xtensa LX6
- b. *Dual-core* atau *single-core* (tergantung varian)
- c. Kecepatan clock hingga 240 MHz

2. Memori :

- a. RAM: 520 KB SRAM
- b. ROM: 448 KB
- c. Flash: 4 MB (dapat diperluas)

3. Konektivitas :

- a. *Wi-Fi*: 802.11 b/g/n (2.4 GHz) dengan kecepatan hingga 150 Mbps
- b. *Bluetooth*: BLE (*Bluetooth Low Energy*) dan *Bluetooth Classic*

4. Antarmuka I/O :

- a. GPIO: Hingga 34 pin yang dapat digunakan untuk input/output
- b. Analog: 18 saluran ADC (*Analog to Digital Converter*) dengan resolusi 12-bit
- c. DAC: 2 saluran *Digital to Analog Converter* (DAC)
- d. PWM: Mendukung kontrol sinyal *Pulse Width Modulation*
- e. UART: Hingga 3 port UART
- f. SPI: 4 port SPI untuk komunikasi dengan sensor atau perangkat lainnya
- g. I2C: 2 port I2C untuk komunikasi dengan berbagai sensor
- h. I2S: Mendukung komunikasi audio digital

5. Fitur Tambahan :

- a. RTC (*Real-Time Clock*): Memungkinkan pengelolaan waktu dan mode hemat daya
- b. Sensor internal: Termasuk sensor suhu dan touch sensor (kapasitif)
- c. Keamanan: Mendukung enkripsi AES, RSA, dan *secure boot*
- d. Mode hemat daya: Berbagai mode seperti *Deep Sleep* dan *Light Sleep* untuk efisiensi daya
- e. Perlindungan ESD: Lebih tahan terhadap gangguan listrik statis

2.8.2 Fitur ESP32

1. Dual-Core dengan Kinerja Tinggi

Memungkinkan pemrosesan tugas secara bersamaan dengan efisiensi yang lebih baik dibandingkan pendahulunya, ESP8266.

2. Konektivitas Wi-Fi dan Bluetooth

Bisa digunakan untuk komunikasi nirkabel tanpa perlu tambahan modul eksternal, sangat cocok untuk proyek IoT.

3. Banyak Antarmuka Komunikasi

Mendukung berbagai protokol komunikasi seperti SPI, I2C, UART, ADC, DAC, dan PWM, mempermudah integrasi dengan sensor dan perangkat lain.

4. Fitur Keamanan yang Lebih Baik

Menyediakan enkripsi perangkat keras AES, SHA, RSA, dan fitur *secure boot* untuk melindungi data pengguna.

5. Mode Hemat Daya

Berbagai mode *Deep Sleep*, *Light Sleep*, dan *Modem Sleep* memungkinkan penggunaan daya yang lebih efisien, ideal untuk perangkat berbasis baterai.

6. Kompatibilitas dengan Berbagai Platform

Bisa diprogram menggunakan *Arduino IDE*, *MicroPython*, *ESP-IDF*, dan *PlatformIO*, memudahkan pengembang untuk bekerja dengan ESP32.

2.9 Sensor Ultrasonik



Gambar 2.3 Sensor Ultrasonik

(Sumber: <https://www.arduinoindonesia.id/2022/10/pengertian-dan-cara-kerja-sensor-ultrasonik-HC-SR04.html>)

Menurut (Arifin et al., n.d.) Sensor *ultrasonik* adalah perangkat yang digunakan untuk mengukur jarak suatu objek menggunakan gelombang ultrasonik. Sensor ini bekerja dengan cara memancarkan gelombang suara pada frekuensi tinggi, kemudian mendeteksi pantulan gelombang tersebut dari objek di depannya. Waktu yang dibutuhkan gelombang untuk kembali digunakan untuk menghitung jarak objek dengan sensor. Sensor ini umum digunakan dalam sistem parkir, robotika, dan alat ukur jarak otomatis.

2.10 Float Sensor



Gambar 2. 4 *Float* Sensor

(Sumber: <https://www.gadgetronicx.com/interfacing-float-sensor-arduino/>)

Float sensor merupakan sensor yang digunakan untuk mendeteksi ketinggian atau level suatu cairan dalam wadah atau tangki. Prinsip kerja sensor ini didasarkan pada pelampung yang bergerak naik turun mengikuti permukaan cairan. Ketika level cairan berubah, posisi pelampung juga berubah, sehingga sensor dapat memberikan sinyal apakah cairan berada di atas, di bawah, atau pada tingkat tertentu. Sensor ini banyak digunakan dalam sistem pengendalian air, tangki penampungan, dan peralatan rumah tangga seperti dispenser air otomatis.

2.11 *WaterFlow*



Gambar 2.5 *Water Flow* Sensor

(Sumber: <https://alphatronic.lk/product/waterflow-sensor/>)

Menurut *Rozikhin et al (2024)* Sensor *water flow* merupakan salah satu perangkat yang digunakan untuk mendeteksi serta mengukur total volume air yang mengalir melalui pipa atau jalur terbuka dalam satuan liter. Sensor ini bekerja dengan cara memutar rotor yang ada di dalamnya, di mana kecepatan rotasi rotor bergantung pada laju aliran air yang melewatinya.

Prinsip kerja sensor ini didasarkan pada efek Hall, yang memungkinkan perangkat menghasilkan sinyal pulsa sesuai dengan kecepatan putaran rotor. Sinyal ini kemudian dikirimkan ke mikrokontroler untuk diproses lebih lanjut, sehingga

memungkinkan sistem untuk membaca dan menganalisis jumlah air yang mengalir secara akurat.

2.12 DHT11



Gambar 2.6 DH11

(Sumber: <https://www.aksesoriskomputerlampung.com/2019/04/dht-11-sensor-suhu-humidity.html>)

Menurut *Rangan et al. (2020)* Sensor DHT11 merupakan modul yang digunakan untuk mendeteksi suhu dan kelembaban udara. Sensor ini menghasilkan output berupa tegangan analog yang dapat diproses lebih lanjut menggunakan mikrokontroler. Secara umum, DHT11 memiliki fitur kalibrasi untuk memastikan akurasi dalam pembacaan suhu dan kelembaban. Data hasil kalibrasi ini disimpan dalam memori program OTP (*One-Time Programmable*) yang dikenal sebagai koefisien kalibrasi, sehingga sensor dapat memberikan hasil pengukuran yang lebih stabil dan andal.

2.13 *Light Emitting Diode (LED)*



Gambar 2.7 LED

(Sumber: <https://www.elektronikabersama.web.id/2011/05/led-atau-light-emitting-diode.html>)

Lampu LED dikenal sebagai pencahayaan masa depan yang sangat efisien dalam konsumsi energi, ramah lingkungan, serta memiliki umur pemakaian yang lebih lama dibandingkan lampu konvensional. Nama LED berasal dari teknologi yang digunakannya, yaitu *Light Emitting Diode*, yang berfungsi sebagai sumber cahaya.

LED adalah jenis dioda yang mampu memancarkan cahaya monokromatik ketika diberikan bias tegangan maju (*forward bias*). Sebagai komponen elektronik berbasis dioda, LED dibuat dari bahan semikonduktor. Saat arus listrik mengalir melalui dioda, elektron dalam material semikonduktor melepaskan energi, yang sebagian besar berbentuk panas dan sebagian lainnya dalam bentuk cahaya (Daya et al., 2021).

2.14 *Arduino IDE*



Gambar 2.8 *Arduino IDE*

(Sumber: <https://flathub.org/apps/cc.arduino.arduinoide>)

Menurut Mahanin Tyas et al., (2023) Arduino IDE merupakan perangkat lunak yang digunakan untuk menulis dan mengembangkan program (*sketch*) yang akan dijalankan pada papan mikrokontroler Arduino. Dengan kata lain, Arduino IDE berfungsi sebagai platform utama dalam proses pemrograman *board* Arduino.

Melalui *software* ini, pengguna dapat membuat, mengedit, menulis kode program, dan mengunggahnya langsung ke papan Arduino yang diinginkan.

Arduino IDE dirancang menggunakan bahasa pemrograman Java, serta dilengkapi dengan pustaka (*library*) C/C++ (*wiring*) yang memudahkan dalam pengoperasian fungsi input dan output pada perangkat keras.

2.14.1 Jenis Fungsi *Arduino IDE*

a. *Setup () {}*

Setup ini berperan untuk melakukan pengaturan awal yang hanya dijalankan satu kali saat program mulai berjalan atau ketika papan Arduino dinyalakan. Umumnya, fungsi ini digunakan untuk menetapkan konfigurasi pin sebagai input atau output, mengaktifkan komunikasi serial, serta melakukan pengaturan dasar lainnya sebelum program utama dijalankan.

b. *Loop () {}*

Loop ini akan dieksekusi secara terus-menerus selama papan Arduino dalam keadaan menyala atau mendapat suplai daya. Bagian ini merupakan inti dari program, di mana seluruh logika utama diletakkan dan dijalankan secara berulang tanpa henti. Biasanya, dalam fungsi ini terdapat perintah-perintah untuk membaca data dari sensor, mengendalikan aktuator seperti LED, motor, atau relay, serta menjalankan berbagai kondisi atau keputusan logis yang telah ditentukan sebelumnya berdasarkan masukan dari lingkungan atau pengguna. Dengan kata lain, fungsi ini bertanggung jawab atas jalannya keseluruhan sistem sesuai dengan skenario yang telah diprogram.

2.14.2 Fitur Fitur *Arduino IDE*

Tabel 2.2 Fitur Fitur *Arduino IDE*

NO	Fitur Arduino IDE	Deskripsi
1	Editor Kode	Antarmuka sederhana untuk menulis dan mengedit sketch (kode program).
2	<i>Verify / Compile</i>	Mengecek dan mengkompilasi kode untuk mendeteksi error sebelum diunggah ke <i>board</i> .
3	<i>Upload</i>	Mengirim kode program ke papan mikrokontroler seperti Arduino atau ESP32.
4	Serial Monitor	Menampilkan data keluaran dari <i>board</i> secara <i>real-time</i> dan menerima input dari pengguna.
5	<i>Library Manager</i>	Mengelola dan menginstal pustaka (<i>library</i>) tambahan untuk mempermudah pemrograman.
6	<i>Board Manager</i>	Menginstal dan memilih berbagai jenis papan mikrokontroler yang kompatibel.
7	<i>Auto Detection Port</i>	Mendeteksi port USB secara otomatis saat <i>board</i> terhubung.

8	Serial Plotter	Menyajikan data dalam bentuk grafik secara langsung dari sensor atau output board.
9	Kompatibilitas Multi-Platform	Bisa dijalankan di Windows, macOS, dan Linux.
10	Mendukung Bahasa C/C++	Menggunakan sintaks C/C++ dan wiring yang disesuaikan untuk mikrokontroler Arduino.
11	Dukungan Plugin & Ekstensi (v2.0 ke atas)	Mendukung <i>auto-complete</i> , <i>error checking</i> , dan fitur seperti editor modern.

2.15 Kabel *Jumper*



Gambar 2.9 Kabel *Jumper*

(Sumber: <https://www.arduinoindonesia.id/2022/11/pengertian-jenis-dan-cara-kerja-kabel-jumper-arduino.html>)

Kabel *jumper* adalah jenis kabel elektrik yang digunakan untuk menghubungkan berbagai komponen pada *breadboard* atau papan Arduino tanpa memerlukan proses penyolderan. Kabel ini umumnya sudah dilengkapi dengan pin konektor di setiap ujungnya, sehingga memudahkan dalam menyusun dan

menghubungkan rangkaian elektronik secara praktis dan fleksibel (Tantowi & Kurnia, 2020). Selain itu, kabel jumper tersedia dalam beberapa jenis berdasarkan ujung konektornya, seperti male-to-male, male-to-female, dan female-to-female, yang masing-masing memiliki fungsi tersendiri dalam proses perakitan.

2.15.1 Jenis Kabel *Jumper*

Berikut adalah tiga jenis utama kabel jumper yang umum digunakan dalam proyek elektronik:

1. *Jumper Male to Male* (Pin ke Pin)



Gambar 2.10 Kabel *Jumper Male to Male*

(Sumber: <https://www.arduinoindonesia.id/2022/11/pengertian-jenis-dan-cara-kerja-kabel-jumper-arduino.html>)

- a. Memiliki konektor *male* (laki-laki) di kedua ujungnya.
- b. Digunakan untuk menghubungkan dua titik pada breadboard atau dari modul sensor ke *breadboard*.

2. *Jumper Male to Female* (Pin ke Lubang)



Gambar 2.11 Kabel *Jumper Male to Female*

(Sumber: <https://www.arduinoindonesia.id/2022/11/pengertian-jenis-dan-cara-kerja-kabel-jumper-arduino.html>)

- a. Satu ujung memiliki konektor *male* (laki-laki) dan ujung lainnya *female* (perempuan).
- b. Biasanya digunakan untuk menghubungkan sensor atau modul ke mikrokontroler, seperti Arduino atau ESP32.

3. *Jumper Female to Female* (Lubang ke Lubang)



Gambar 2.12 Kabel *Jumper Female to Female*

(Sumber: <https://www.arduinoindonesia.id/2022/11/pengertian-jenis-dan-cara-kerja-kabel-jumper-arduino.html>)

- a. Memiliki konektor *female* (perempuan) di kedua ujungnya.

- b. Digunakan untuk menghubungkan dua pin male pada komponen elektronik, seperti menghubungkan header pin mikrokontroler dengan modul eksternal

2.16 Telegram



Gambar 2.13 Telegram

(Sumber: https://en.wikipedia.org/wiki/Telegram_%28software%29)

Telegram merupakan salah satu platform komunikasi berbasis internet yang mengalami pertumbuhan pesat sejak peluncurannya pada tahun 2013. Popularitas aplikasi ini tidak lepas dari beragam fitur unggulan yang ditawarkannya, seperti *channel* (saluran), grup dengan kapasitas besar, bot otomatis, serta sistem keamanan yang menggunakan enkripsi *end-to-end*.

Aplikasi ini tidak hanya dimanfaatkan untuk keperluan komunikasi pribadi, tetapi juga telah menjadi alat strategis dalam dunia bisnis digital. Bisnis digital sendiri merujuk pada aktivitas promosi atau pemasaran suatu produk maupun merek yang dilakukan melalui *platform elektronik*.

Di tengah era globalisasi dan percepatan transformasi digital, Telegram telah digunakan secara luas dalam berbagai aktivitas bisnis, mulai dari digital marketing, pengelolaan komunitas, hingga pengembangan usaha berbasis teknologi. Transformasi digital ini membawa dampak besar terhadap cara kerja

dunia bisnis, memungkinkan proses yang lebih cepat, efisien, dan membuka peluang untuk pengambilan keputusan yang lebih strategis (Noviyanti et al., 2024).

2.16.1 Keunggulan *Telegram*

Telegram memiliki beberapa keunggulan yang menjadikannya sangat efektif sebagai media peringatan atau notifikasi, khususnya untuk sistem otomatis dan berbasis IoT:

Tabel 2.3 Keunggulan Telegram

NO	Fitur Telegram	Deskripsi
1	Notifikasi <i>Real-Time</i>	Pesan dikirim dan diterima secara instan, sangat cocok untuk sistem peringatan dini.
2	Dukungan Bot Otomatis	Dapat diintegrasikan dengan mikrokontroler untuk mengirim notifikasi otomatis dari sensor.
3	<i>Multi-Platform</i>	Bisa diakses melalui berbagai perangkat (<i>smartphone</i> , PC, atau <i>browser</i>) tanpa batasan.
4	Grup dan <i>Channel</i>	Memungkinkan pengiriman pesan massal

		ke banyak pengguna lewat grup atau saluran (<i>channel</i>).
5	Aman dan Bebas Iklan	Menggunakan enkripsi <i>end-to-end</i> serta bebas dari iklan yang mengganggu.
6	Gratis dan Mudah Diintegrasikan	Menyediakan API terbuka yang mendukung integrasi dengan sistem pemantauan atau IoT.

2.16.2 Telegram Robot

Menurut Mulyanto (2020) *Bot Telegram* merupakan aplikasi pihak ketiga yang dapat dijalankan di dalam platform *Telegram*. Melalui bot, pengguna dapat mengirimkan pesan, perintah, maupun inline request untuk memperoleh layanan atau informasi tertentu. Bot dikendalikan menggunakan protokol HTTPS yang terhubung ke *Telegram* API, memungkinkan pengembang mengintegrasikan berbagai fungsi otomatis ke dalam bot tersebut. *Bot Telegram* memiliki banyak manfaat, mulai dari memberikan layanan informasi hingga menjalankan perintah khusus secara otomatis.

Beragam penggunaan bot telah diterapkan dalam kehidupan sehari-hari. Sebagai contoh, bot dapat difungsikan sebagai koran pintar yang mengirimkan berita terbaru kepada pelanggan secara berkala. Selain itu, bot dapat menjadi

jemabatan antara *Telegram* dengan layanan eksternal seperti *Gmail*, *YouTube*, *Wikipedia*, *GitHub*, dan lainnya. *Bot Telegram* juga mendukung fitur pembayaran, di mana pengguna dapat melakukan transaksi langsung melalui bot, baik untuk layanan digital, produk virtual, maupun etalase belanja online. Beberapa bot juga menawarkan fitur khusus seperti peringatan cuaca, terjemahan otomatis, pemformatan teks, hingga permainan interaktif, baik single-player maupun *multiplay*

BAB III

METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini termasuk dalam kategori penelitian terapan, yaitu jenis penelitian yang bertujuan untuk menerapkan, menguji, dan mengembangkan pengetahuan atau teknologi tertentu agar dapat digunakan untuk menyelesaikan masalah nyata, khususnya dalam mitigasi bencana banjir.

Pendekatan yang digunakan dalam penelitian ini adalah pendekatan kuantitatif, yang bertujuan untuk mengukur dan menganalisis data numerik yang diperoleh dari berbagai sensor lingkungan seperti ketinggian air, suhu, kelembapan, dan kecepatan aliran air. Data yang dikumpulkan kemudian dianalisis menggunakan metode TOPSIS untuk menghasilkan prediksi kondisi banjir berdasarkan perhitungan matematis secara objektif. Dengan pendekatan ini, hasil penelitian dapat diuji secara statistik dan direproduksi dalam konteks yang serupa.

3.2 Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian ini digunakan untuk memperoleh informasi yang diperlukan dalam proses perancangan dan pengembangan sistem. Data dikumpulkan melalui studi pustaka terhadap sumber-sumber literatur yang relevan, serta melalui simulasi pembacaan sensor dalam lingkungan laboratorium guna menghasilkan data yang akan digunakan dalam proses pengolahan menggunakan metode TOPSIS. Selain itu, dilakukan pengujian berulang untuk memastikan konsistensi data yang diperoleh dan keakuratan hasil analisis dalam menggambarkan kondisi lingkungan yang sebenarnya.

3.2.1 Studi Pustaka

Studi pustaka dilakukan untuk memperoleh landasan teori terkait sistem pendukung keputusan, metode TOPSIS, dan teknologi IoT, dengan sumber dari buku, jurnal, dan penelitian terdahulu yang relevan dengan peringatan dini banjir. Berdasarkan kajian terhadap penelitian-penelitian terdahulu, dapat diidentifikasi beberapa celah (*gap*) yang belum banyak dibahas secara menyeluruh. Penelitian terdahulu umumnya hanya menggunakan satu jenis sensor, seperti sensor ketinggian air, tanpa integrasi berbagai parameter lingkungan seperti debit air, suhu, dan kelembapan secara bersamaan. Selain itu, pendekatan pengambilan keputusan yang digunakan umumnya masih bersifat sederhana atau kualitatif seperti analisis SWOT atau logika fuzzy, tanpa memanfaatkan metode kuantitatif *multi-kriteria* seperti TOPSIS yang dapat menghasilkan keputusan yang lebih objektif dan terukur. Penggunaan media notifikasi otomatis seperti Telegram juga masih jarang ditemukan, padahal hal ini sangat penting dalam mendukung penyebaran informasi cepat ke masyarakat.

Dari penelitian sebelumnya, penelitian ini mengembangkan sistem pemantauan banjir berbasis *Internet of Things* (IoT) yang mampu mengumpulkan data dari beberapa jenis sensor secara *real-time*. Selanjutnya, data tersebut diproses menggunakan metode TOPSIS untuk menentukan tingkat risiko banjir secara sistematis. Output dari sistem akan dikirimkan dalam bentuk notifikasi otomatis melalui *Telegram Bot*. Dengan demikian, penelitian ini menempati posisi penting dalam pengembangan sistem peringatan dini yang tidak hanya terintegrasi dan responsif, tetapi juga berbasis data yang terukur dan bersifat *real-time*.

3.2.2 Observasi Sistem Simulasi

Observasi sistem simulasi dilakukan pada prototipe miniatur sungai di laboratorium, yang mensimulasikan kondisi banjir seperti naiknya permukaan air dan aliran deras. Sensor dipasang untuk memantau kondisi secara *real-time*.

3.2.3 Jadwal Penelitian

Tabel 3.1 Jadwal Penelitian

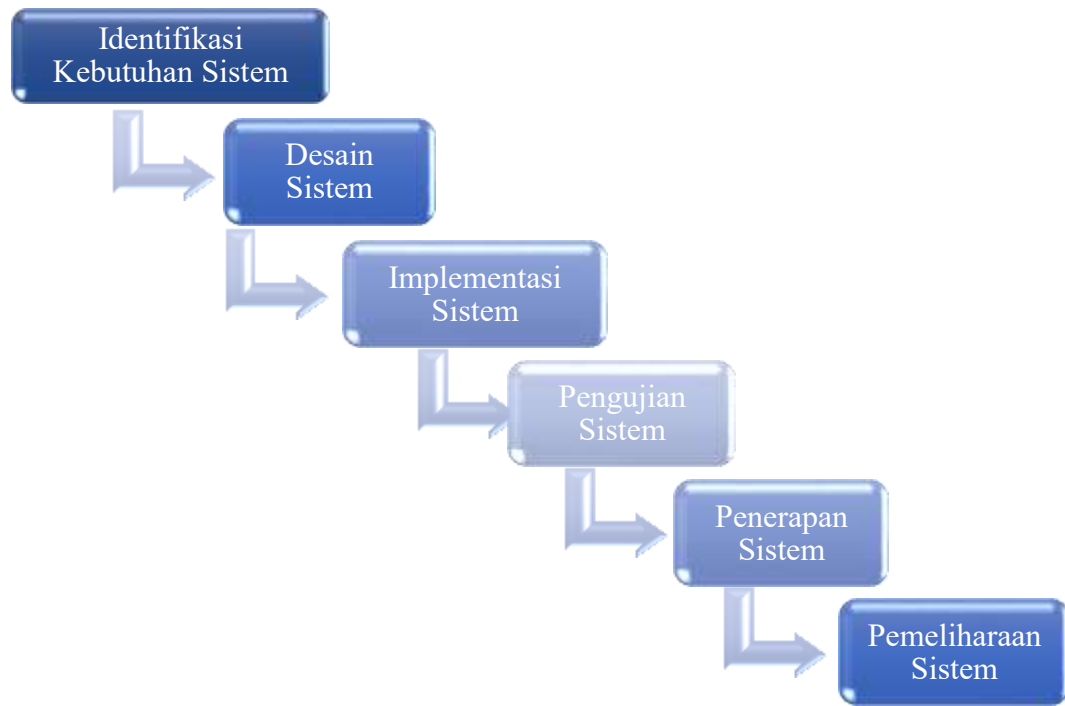
No	Uraian Kegiatan	Maret				April				Mei				Juni				Juli			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Tahap Persiapan Penelitian																				
2.	Tahap Penyusunan Proposal																				
3.	Seminar Proposal																				
3.	Tahap Pelaksanaan Penelitian																				
4.	Tahap Penyusunan Laporan dan Revisi																				
5.	Ujian Skripsi																				

3.2.4 Pengujian Prototipe

Pengujian prototipe dilakukan dengan memberikan input data dari sensor (*ultrasonik, DHT11, float, water flow*) yang dipasang pada miniatur sungai. Data yang diperoleh kemudian diproses menggunakan metode TOPSIS dan hasilnya dikirim serta disimpan ke dalam database *Telegram*, yaitu berupa histori notifikasi otomatis yang dikirim melalui bot. Melalui pengujian ini, diuji keakuratan data sensor, ketepatan klasifikasi tingkat risiko banjir, serta kecepatan respon sistem dalam memberikan peringatan dini.

3.3 Penerapan Perancangan Alat Peringatan Dini Banjir

Tahapan dalam Perancangan alat Peringatan dini Potensi Banjir :



Gambar 3.1 *Diagram Waterfall*

1. *Requirement Analysis* (Analisis Kebutuhan)

Mengumpulkan semua kebutuhan sistem dari pengguna atau stakeholder, baik kebutuhan fungsional maupun non-fungsional.

2. *System Design* (Perancangan Sistem)

Membuat desain sistem yang mencakup:

- a. Arsitektur sistem
- b. Diagram alir, diagram blok, dsb

3. *Implementation* (Implementasi/Koding)

Tahap pembuatan sistem sesuai rancangan yang telah dibuat, berupa penulisan

kode program (Arduino IDE), konfigurasi perangkat keras (ESP32, Sensor, dan Modul Komunikasi) dll.

4. *Testing* (Pengujian)

Sistem diuji untuk memastikan tidak ada kesalahan (bug), serta fungsinya sesuai dengan kebutuhan awal.

5. *Deployment* (Penerapan/Instalasi Sistem)

Sistem mulai digunakan oleh user di lingkungan nyata atau simulasi (misalnya di miniatur atau laboratorium skripsi).

6. *Maintenance* (Pemeliharaan)

Perbaikan jika ada kesalahan saat digunakan, atau pengembangan lebih lanjut bila diperlukan.

3.4 Alat dan Bahan

Dalam Penelitian ini menggunakan alat dan bahan untuk menunjang proses pembuatan dan perakitan, yaitu :

3.4.1 *Hardware*

a. ESP32 (Mikrokontroler)

Mikrokontroler berbasis *Wi-Fi* dan *Bluetooth* yang digunakan untuk membaca data sensor dan mengendalikan sistem secara *real-time*.

b. Sensor Ultrasonik

Sensor yang digunakan untuk mengukur ketinggian air dengan memanfaatkan gelombang *ultrasonik*.

c. Sensor DHT11

Sensor digital yang berfungsi untuk Kondisi Cuaca di lingkungan sekitar.

d. *Water Flow Sensor*

Sensor yang digunakan untuk mengukur kecepatan aliran air (debit) diuji melalui tiga kali percobaan dengan volume air yang berbeda

e. *Float Sensor*

Sensor berbasis pelampung yang mendeteksi tinggi permukaan air dan digunakan untuk mengetahui level air (tinggi atau rendah).

f. *LED*

Dioda pemancar cahaya yang digunakan sebagai indikator visual status sistem, misalnya normal, siaga, atau bahaya.

g. *Kabel Jumper*

Kabel kecil yang digunakan untuk menghubungkan komponen.

h. *Buzzer*

3.4.2 *Software*

a. *Arduino IDE*

Aplikasi pemrograman yang digunakan untuk menulis, mengedit, dan mengunggah kode ke mikrokontroler ESP32.

b. *Telegram*

Antarmuka pemrograman aplikasi dari *Telegram* yang memungkinkan sistem mengirim notifikasi otomatis ke pengguna.

c. *Python*

Digunakan sebagai bahasa pemrograman untuk melakukan perhitungan dan simulasi metode TOPSIS untuk mendapatkan keputusan secara otomatis dan terkomputerisasi.

d. Draw.io

Aplikasi desain skematik dan rangkaian elektronik yang digunakan untuk membuat diagram sistem dan koneksi antar komponen.

3.5 Metode Pengolahan Data

Pengolahan data dalam penelitian ini menggunakan metode **TOPSIS** (*Technique for Order Preference by Similarity to Ideal Solution*), yaitu metode pengambilan keputusan multikriteria yang menentukan alternatif terbaik berdasarkan jaraknya terhadap solusi ideal positif (terbaik) dan solusi ideal negatif (terburuk). Dalam sistem ini, data yang diukur oleh sensor akan dikonversi ke dalam bentuk matriks keputusan untuk dianalisis menggunakan TOPSIS. Setiap sensor berperan sebagai kriteria untuk menentukan tingkat potensi risiko banjir. Adapun sebagai contoh kriteria (parameter) yang digunakan meliputi:

- a. C1 = Ketinggian Air (dari sensor *ultrasonik* + *Float Sensor*)
- b. C2 = Suhu (dari sensor DHT11)
- c. C3 = Kelembapan (dari sensor DHT11)
- d. C4 = Kecepatan Aliran Air (dari *water flow* sensor)

Tahapan Pengolahan Data dengan Metode TOPSIS :

1. Menentukan Matriks Keputusan

- a. Menyusun matriks keputusan berdasarkan alternatif dan kriteria yang telah ditentukan.
- b. Setiap elemen dalam matriks merepresentasikan nilai suatu alternatif terhadap kriteria tertentu.

Sebagai Contoh Perhitungan Data :

Tabel 3.1 Kriteria & skor

Kriteria	Skor	Bobot%
C1	3	3/10
C2	2	2/10
C3	2	2/10
C4	3	3/10
Total	10	100

Tabel 3.2 Matriks Keputusan (Data Sensor)

Alternatif	C1(Ketinggian Air)	C2(Suhu)	C3(Kelembapan)	C4(Kecepatan Aliran Air)
A1	30	29	80	2.5
A2	50	33	70	6.2
A3	40	31	75	4.0

2. Normalisasi Matriks Keputusan

- a. Melakukan normalisasi dengan menggunakan rumus:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

- b. Di mana r_{ij} adalah nilai normalisasi untuk alternatif ke- i dan kriteria ke- j , serta x_{ij} adalah nilai asli dari matriks keputusan.

Langkah Hitung Akar Total Tiap Kolom:

- a. C1: $\sqrt{(30^2 + 50^2 + 40^2)} = \sqrt{(900 + 2500 + 1600)} = \sqrt{5000} = \mathbf{70.71}$

$$r_{(C1,A1)} = 30/70.71 = \mathbf{0.424}$$

$$r_{(C1,A2)} = 50/70.71 = \mathbf{0.707}$$

$$r_{(C1,A3)} = 40/70.71 = \mathbf{0.566}$$

$$\text{b. C2: } \sqrt{(29^2 + 33^2 + 31^2)} = \sqrt{(841 + 1089 + 961)} = \sqrt{2891} = \mathbf{53.77}$$

$$r_{(C1,A1)} = 29/53.77 = \mathbf{0.539}$$

$$r_{(C1,A2)} = 33/53.77 = \mathbf{0.641}$$

$$r_{(C1,A3)} = 31/53.77 = \mathbf{0.576}$$

$$\text{c. C3: } \sqrt{(80^2 + 70^2 + 75^2)} = \sqrt{(6400 + 4900 + 5625)} = \sqrt{16925} = \mathbf{130.09}$$

$$r_{(C1,A1)} = 80/130.09 = \mathbf{0.615}$$

$$r_{(C1,A2)} = 70/130.09 = \mathbf{0.538}$$

$$r_{(C1,A1)} = 75/130.09 = \mathbf{0.576}$$

$$\text{d. C4: } \sqrt{(2.5^2 + 6.2^2 + 4.0^2)} = \sqrt{(6.25 + 38.44 + 16)} = \sqrt{60.69} = \mathbf{7.79}$$

$$r_{(C1,A1)} = 2.5/7.79 = \mathbf{0.321}$$

$$r_{(C1,A2)} = 6.2/7.79 = \mathbf{0.796}$$

$$r_{(C1,A1)} = 4.0/7.79 = \mathbf{0.513}$$

Tabel 3.3 Hasil Normalisasi Matriks

Alternatif	r(C1)	r(C2)	r(C3)	r(C4)
A1	0.424	0.539	0.615	0.321
A2	0.707	0.641	0.538	0.796
A3	0.566	0.576	0.576	0.513

3. Menentukan Matriks Keputusan Ternormalisasi Tertimbang

a. Mengalikan nilai matriks normalisasi dengan bobot masing-masing kriteria:

$$v_{ij} = w_j \times r_{ij}$$

b. Di mana w_j adalah bobot kriteria ke- j .

Bobot :

a. $C1 = 3$

b. $C2 = 2$

c. $C3 = 2$

d. $C4 = 3$

Tabel 3.4 Hasil Matriks Keputusan Ternormalisasi

Alternatif	$v(C1) = 3 \times$ $r(C1)$	$v(C2) = 2 \times$ $r(C2)$	$v(C3) = 2 \times$ $r(C3)$	$v(C4) = 3 \times$ $r(C4)$
A1	$424 \times 3 = \mathbf{1272}$	$539 \times 2 = \mathbf{1078}$	$615 \times 2 = \mathbf{1230}$	$321 \times 3 = \mathbf{963}$
A2	$707 \times 3 = \mathbf{2121}$	$641 \times 2 = \mathbf{1282}$	$538 \times 2 = \mathbf{1076}$	$796 \times 3 = \mathbf{2388}$
A3	$566 \times 3 = \mathbf{1698}$	$576 \times 2 = \mathbf{1152}$	$576 \times 2 = \mathbf{1152}$	$513 \times 3 = \mathbf{1539}$

4. Menentukan Solusi Ideal Positif dan Solusi Ideal Negatif

a. Solusi ideal positif (A^+) :

$$A^+ = (\max y(C1), \max y(C2), \max y(C3))$$

a. $\max y_{(C1)} = \max(1272, 2121, 1698) = \mathbf{1698}$

b. $\max y_{(C2)} = \max(1078, 1282, 1152) = \mathbf{1282}$

c. $\max y_{(C3)} = \max(1230, 1076, 1152) = \mathbf{1230}$

d. $\max y_{(C4)} = \max(963, 2388, 1539) = \mathbf{2388}$

$$A^+ = \{v_1^+, v_2^+, \dots, v_n^+\}$$

Maka :

$$A^+ = (\mathbf{1698, 1282, 1230, 2388})$$

b. Solusi ideal positif (A^+) :

$$A^+ = (\min y(C1), \min y(C2), \min y(C3))$$

$$a. \min y(C1) = \min(1272, 2121, 1698) = \mathbf{1272}$$

$$b. \min y(C2) = \min(1078, 1282, 1152) = \mathbf{1078}$$

$$c. \min y(C3) = \min(1230, 1076, 1152) = \mathbf{1076}$$

$$d. \min y(C3) = \min(963, 2388, 1539) = \mathbf{963}$$

$$A^- = \{v_1^-, v_2^-, \dots, v_n^-\}$$

Maka :

$$A^- = (\mathbf{1272, 1078, 1076, 963})$$

- a. Nilai maksimum digunakan untuk kriteria keuntungan (benefit),
sedangkan nilai minimum untuk kriteria biaya (cost).

c. Menghitung Jarak Alternatif terhadap Solusi Ideal

a. Jarak terhadap solusi ideal positif (D^+) :

$$D_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2}$$

Alternatif A1 :

$$\begin{aligned} D^+ &= \sqrt{(424 - 1698)^2 + (539 - 1282)^2 + (615 - 1230)^2 + (321 - 2388)^2} \\ &= \sqrt{(-1274)^2 + (-743)^2 + (-615)^2 + (-2067)^2} = \sqrt{1.623 + 552 + 4.272} = \sqrt{6.456} \\ &= \mathbf{2.540} \end{aligned}$$

Alternatif A2 :

$$\begin{aligned} D^+ &= \sqrt{(2121 - 1698)^2 + (1282 - 1282)^2 + (1076 - 1230)^2 + (2388 - 2388)^2} \\ &= \sqrt{(423)^2 + (0)^2 + (-154)^2 + (0)^2} = \sqrt{202.645} = \mathbf{14.235} \end{aligned}$$

Alternatif A3 :

$$\begin{aligned}
 D^+ &= \sqrt{(1698-1698)^2 + (1152-1282)^2 + (1152-1230)^2 + (1539-2388)^2} \\
 &= \sqrt{(0)^2 + (-130)^2 + (-78)^2 + (-849)^2} = \sqrt{0 + 16.900 + 6.084 + 720} \\
 &= \sqrt{23.704} = \mathbf{4.87}
 \end{aligned}$$

b. Jarak terhadap solusi ideal negatif (D^-) :

$$D_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}$$

Alternatif A1 :

$$\begin{aligned}
 D^- &= \sqrt{(424-1272)^2 + (539-1078)^2 + (615-1076)^2 + (321-963)^2} = \sqrt{(-848)^2 + (-539)^2 + (-425)^2 + (-642)^2} \\
 &= \sqrt{1.602} = \mathbf{1.266}
 \end{aligned}$$

Alternatif A2 :

$$\begin{aligned}
 D^- &= \sqrt{(2121-1272)^2 + (1282-1078)^2 + (1076-1076)^2 + (2388-963)^2} = \\
 &= \sqrt{(849)^2 + (204)^2 + (0)^2 + (1425)^2} = \sqrt{2.793} = \mathbf{1.671}
 \end{aligned}$$

Alternatif A3 :

$$\begin{aligned}
 D^- &= \sqrt{(1698-1272)^2 + (1152-1078)^2 + (1152-1076)^2 + (1539-963)^2} \\
 &= \sqrt{(426)^2 + (74)^2 + (76)^2 + (576)^2} = \sqrt{524} = \mathbf{22.891}
 \end{aligned}$$

d. Menghitung Nilai Preferensi untuk Setiap Alternatif

a. Menghitung nilai preferensi dengan rumus :

$$P_i = \frac{D_i^-}{D_i^+ + D_i^-}$$

Tabel 3.5 Nilai Prefensi

Alternatif	D+	D-
A1	2.540	1.266
A2	14.235	1.671
A3	4.87	22.891

Alternatif A1 :

$$P_1 = 1.266 / 2.540 + 1.266 = 1.266 / 3.806 = \mathbf{0.33}$$

Kategori : **Perlu di Pantau**

Alternatif A2 :

$$P_2 = 1.671 / 14.235 + 1.671 = 1.671 / 15.906 = \mathbf{0.11}$$

Kategori : **Hujan Berhenti**

Alternatif A3 :

$$P_3 = 22.891 / 4.87 + 22.891 = 22.891 / 27.761 = \mathbf{0.824}$$

Kategori: **Potensi Banjir**

- b. Nilai P_i berkisar antara 0 hingga 1, di mana alternatif dengan nilai P_i terbesar adalah yang paling optimal. Semakin **mendekati 1**, alternatif tersebut **semakin ideal**.

Tabel 3.6 Kategori Resiko

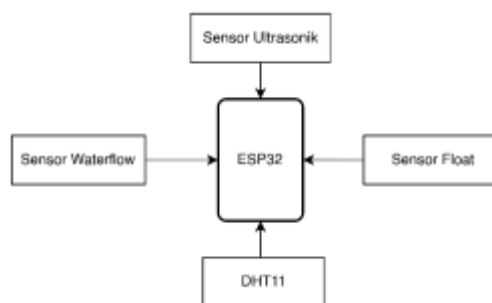
P_i	Kategori Risiko
0 – 0.3	Hujan Berhenti
0.31 – 0.6	Perlu dipantau
> 0.6	Potensi Banjir

Menentukan Peringkat Alternatif

- Alternatif dengan nilai preferensi (P_i) tertinggi menunjukkan risiko banjir tertinggi dan menjadi prioritas penanganan.
- Sistem mengurutkan semua alternatif berdasarkan nilai P_i dari tertinggi ke terendah.

- c. $P_i > 0.6 \rightarrow$ Potensi Banjir: Aktifkan notifikasi Telegram + *LED Red* + buzzer.
- d. $0.31-0.6 \rightarrow$ Perlu Dipantau: Aktifkan *LED Yellow* & buzzer + notifikasi Telegram.
- e. $P_i \leq 0.3 \rightarrow$ Hujan Berhenti: Hanya *LED Green*+ notifikasi Telegram.
- f. Peringkat menunjukkan urutan prioritas risiko, bukan hanya alternatif terbaik.

3.6 Perancangan alat



Gambar 3.2 Diagram blok sistem

Pada Gambar 3.2 Menjelaskan ESP32 yang terhubung dengan beberapa sensor utama. Sensor-sensor ini digunakan untuk memantau kondisi lingkungan secara *real-time*. Sensor *ultrasonik* berfungsi untuk mengukur ketinggian air, yang menjadi indikator utama dalam mendeteksi potensi banjir. Sensor *waterflow* yang digunakan untuk mengukur kecepatan aliran air (debit) diuji melalui tiga kali percobaan dengan volume air yang berbeda. Percobaan pertama menggunakan 150 ml air, kedua 300 ml, dan ketiga 500 ml. Setiap percobaan mengukur jumlah pulsa atau putaran rotor dalam waktu tertentu (dalam satuan menit), yang kemudian dikonversi menjadi nilai debit dalam liter per menit (L/min). Hasil dari ketiga percobaan ini digunakan untuk mengkalibrasi sensor aliran dan memastikan akurasi pengukuran debit air secara *real-time*, yang turut berkontribusi dalam analisis

tingkat risiko. Sensor *float* mendeteksi level permukaan air secara mekanik dan memberikan konfirmasi tambahan terhadap hasil pengukuran dari sensor ultrasonik. Sementara itu, sensor DHT11 berfungsi untuk mengukur suhu dan kelembaban udara di sekitar lingkungan sistem.

Seluruh data dari sensor tersebut dikirim ke ESP32, yang kemudian memproses data menggunakan metode TOPSIS untuk menentukan tingkat risiko banjir. Apabila hasil perhitungan menunjukkan kondisi Potensi Banjir, Hujan Berhenti dan perlu dipantau, sistem akan secara otomatis mengirimkan notifikasi peringatan ke *Telegram* Jika Kondisinya berubah tetapi jika kondisinya tetap sama tidak mengirim notifikasi melalui *Telegram Bot*, yang juga dapat berfungsi sebagai media pencatatan berbasis database *Telegram*. Hal ini memungkinkan pengguna menerima informasi secara cepat dan efisien, sekaligus menyimpan histori notifikasi sebagai referensi untuk pemantauan lanjutan. Integrasi berbagai sensor dengan ESP32 membuat sistem ini bekerja secara otomatis dan responsif terhadap perubahan kondisi lingkungan.

3.7 Proses Implementasi Alat

Proses perakitan dimulai dengan menghubungkan sensor ultrasonik, DHT11, water flow, dan float ke mikrokontroler ESP32 menggunakan *breadboard* dan *kabel jumper*. Sensor-sensor tersebut berfungsi untuk membaca kondisi lingkungan seperti ketinggian air, Kondisi Cuaca, dan Kecepatan aliran debit air. Setelah perakitan selesai, sistem diprogram melalui *Arduino IDE* untuk membaca data sensor, mengolahnya menggunakan metode TOPSIS, dan mengirim notifikasi melalui *Telegram Bot*. Sistem kemudian diuji pada miniatur sungai untuk memastikan semua komponen berfungsi dan notifikasi terkirim sesuai kondisi.

3.8 Diagram Flowchart sistem



Gambar 3. 3 Flowchart Sistem

Pada Gambar 3.3 Menjelaskan proses kerja sistem peringatan dini banjir berbasis *Internet of Things* (IoT) yang terintegrasi dengan metode pengambilan

keputusan TOPSIS dan notifikasi melalui Telegram. Proses dimulai dengan inisialisasi sistem, yaitu mengaktifkan koneksi *Wi-Fi*, sensor-sensor (seperti sensor *ultrasonik*, *flow* sensor, dan *float* sensor). Setelah itu, sistem memeriksa apakah koneksi *Wi-Fi* berhasil. Jika koneksi gagal, sistem akan mencoba menyambung ulang ke jaringan *Wi-Fi*. Jika berhasil, maka sistem akan melakukan pembacaan data dari seluruh sensor yang terpasang.

Data yang dibaca berupa parameter seperti ketinggian air (dari sensor *ultrasonik*), laju aliran air (dari *flow* sensor), dan level air (dari *float* sensor). Setelah data diperoleh, sistem menghitung nilai-nilai tersebut dan mengolahnya menggunakan metode TOPSIS untuk menentukan tingkat risiko banjir secara objektif berdasarkan beberapa kriteria. Hasil dari perhitungan TOPSIS digunakan untuk mengklasifikasikan status kondisi menjadi tiga kategori: Hujan berhenti, Perlu dipantau, dan Potensi Banjir.

Jika status yang terdeteksi adalah *normal*, maka sistem hanya akan mengaktifkan indikator LED hijau sebagai penanda. Namun, jika statusnya *perlu dipantau* atau *potensi banjir*, maka sistem akan mengaktifkan LED kuning atau merah disertai buzzer sebagai peringatan lokal. Selain itu, sistem secara otomatis mengirimkan notifikasi ke pengguna melalui Telegram Bot agar informasi dapat diterima secara real-time. Setelah itu, proses pemantauan akan terus diulang secara berkelanjutan

BAB IV

HASIL DAN PEMBAHASAN

Selanjutnya bab ini menguraikan tentang output uji coba Sistem Pendukung Keputusan berbasis IOT untuk Peringatan dini Potensi Banjir dilakukan untuk memastikan bahwa sistem dapat berfungsi sesuai tujuan, yaitu mendeteksi kondisi lingkungan seperti ketinggian air, suhu, kelembapan, dan kecepatan aliran air, kemudian menganalisisnya menggunakan metode TOPSIS untuk memprediksi potensi banjir dalam 10 menit ke depan. Uji coba ini bertujuan untuk mengevaluasi apakah sistem dapat berjalan dengan baik dalam lingkungan pengujian yang telah ditentukan serta sesuai dengan rancangan perangkat keras dan perangkat lunak yang telah dibuat.

4.1 Langkah Uji Coba

Uji coba dilakukan untuk mengetahui hasil dari implementasi sistem baik dari sisi perangkat keras (*hardware*) maupun perangkat lunak (*software*), serta memastikan bahwa sistem dapat berfungsi sesuai dengan tujuan yang telah dirancang. Pengujian mencakup simulasi pada miniatur sungai untuk mengamati kinerja sensor-sensor yang digunakan, yaitu sensor jarak (ultrasonik), sensor aliran air (flow sensor), sensor suhu dan kelembapan (DHT11), serta sensor untuk ketinggian air (float). Tujuannya adalah menciptakan variasi kondisi lingkungan seperti tinggi muka air, suhu, kelembapan, dan kecepatan aliran air, yang kemudian dianalisis menggunakan metode TOPSIS untuk menghasilkan nilai preferensi (P_i). Untuk memperoleh data tersebut, dilakukan manipulasi volume air sebanyak 150 ml, 300 ml, dan 500 ml dalam tiga kali pengujian dengan 10 data, guna mengamati

perubahan nilai debit dan dampaknya terhadap status risiko banjir yang ditentukan oleh sistem.

Selanjutnya, dilakukan pengujian terhadap performa sistem untuk memastikan bahwa seluruh komponen dapat bekerja secara otomatis dan real-time. Pengujian ini mencakup pemeriksaan LED indikator dan buzzer sebagai sistem peringatan lokal. Status yang dihasilkan oleh sistem dikelompokkan ke dalam tiga kategori, yaitu Hujan Berhenti, Perlu Dipantau, dan Potensi Banjir, yang didasarkan pada hasil perhitungan metode TOPSIS, serta ditampilkan melalui notifikasi yang dikirim ke pengguna menggunakan Telegram Bot.

Setiap kali data sensor berhasil dibaca dan muncul di Serial Monitor, data tersebut juga secara otomatis disimpan dalam bentuk file CSV secara real-time sebagai log pencatatan sistem. Sistem kemudian akan mengambil 10 data terbaru untuk dianalisis menggunakan metode TOPSIS guna menentukan nilai preferensi dan kategori risiko banjir. Setelah status risiko diperoleh, sistem akan mengirimkan notifikasi ke Telegram secara otomatis setiap kali terjadi perubahan status risiko, sehingga pengguna dapat menerima peringatan dini secara cepat dan akurat. Seluruh proses dilakukan secara berulang untuk menilai keakuratan pembacaan sensor, kestabilan sistem, serta kesesuaian respon output terhadap kondisi lingkungan yang berubah.

Melalui rangkaian uji coba ini, dapat dievaluasi seberapa efektif sistem dalam memantau potensi banjir secara otomatis, serta memberikan peringatan dini kepada pengguna secara cepat dan akurat. Pengujian ini sekaligus menjadi validasi

dari keseluruhan proses implementasi sistem pendukung keputusan berbasis IoT dengan metode TOPSIS yang telah dirancang dalam penelitian ini.

4.2 Proses Pembuatan alat Potensi Banjir

Berikut adalah tahapan-tahapan Pembuatan alat Potensi Banjir yaitu :

- a. Tahapan awal yaitu mempersiapkan seluruh komponen, seperti ESP32, sensor DHT11, sensor ultrasonik, flow sensor, float sensor, LED, buzzer, kabel jumper, dan breadboard.
- b. Pasang sensor DHT11, ultrasonik, Flow Sensor dan float sensor dihubungkan ke pin ESP32 sesuai ketentuan Setiap sensor diuji untuk memastikan berfungsi dengan baik.
- c. LED dan buzzer dipasang sebagai output lokal. LED menunjukkan status risiko (normal, siaga, bahaya), sedangkan buzzer aktif saat risiko tinggi dan siaga terdeteksi.
- d. Semua komponen dirakit pada media miniatur aliran sungai. Penempatan sensor disesuaikan untuk pembacaan data yang akurat.
- e. Menggunakan Arduino IDE, dibuat program untuk membaca data sensor.
- f. Proses unggah program ke ESP32 dilakukan dengan menghubungkan perangkat ke laptop menggunakan kabel data, yang sekaligus berfungsi sebagai sumber daya dan jalur komunikasi.
- g. Program diunggah ke ESP32 melalui Arduino IDE. Setelah berhasil, sistem mulai berjalan otomatis saat dinyalakan.
- h. Dilakukan pengujian menyeluruh untuk memastikan sistem membaca data secara real-time, memprosesnya dengan metode TOPSIS, dan memberikan output sesuai kondisi lingkungan.



Gambar 4.1 *Perancangan alat Potensi Banjir*

4.3 Perangkat Keras (*Hardware*)

Berikut adalah beberapa hardware yang digunakan pada riset ini:

Tabel 4. 1 *Perangkat Keras Alat*

NO	Nama Bahan	Jumlah
1.	ESP32	1 pcs
2.	<i>Breadboard</i>	1 pcs
3.	<i>Flow Sensor</i>	1 pcs
4.	<i>Float Sensor</i>	1 pcs
5.	DHT11	1 pcs
6.	Sensor <i>Ultasonic</i>	1 pcs
7.	Kabel <i>Jumper</i>	Secukupnya
8.	Kabel USB	1 pcs

9.	LED	3 pcs
10.	Buzzer	1 pcs
11.	<i>Project IOT Box</i>	1 pcs
12.	Miniatur Sungai	1 Set

4.4 Perangkat Lunak (Software)

Berikut adalah software yang digunakan pada riset ini:

1. Arduino IDE

Software ini dipakai untuk menulis sketch program dengan bahasa pemrograman C++.

4.4.1 Software Program

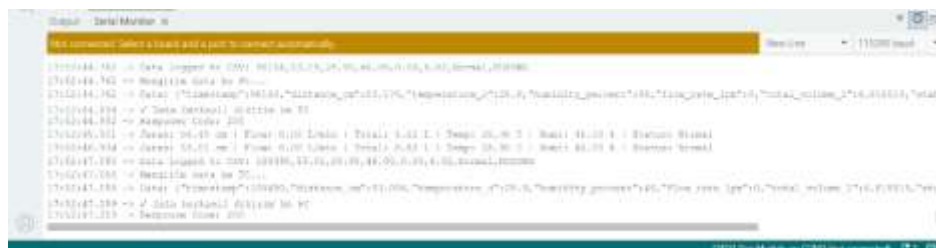
Berikut adalah kode program yang telah dibuat pada Arduino IDE:



```

// ===== PIN ASSIGNMENT =====
10 #define TRIG_PIN 4
11 #define ECHO_PIN 5
12 #define PUMP_MOTOR_PIN 23
13 #define BUZZER_PIN 21
14 #define FLOW_PIN 18
15
16 #define LED_DEFAULT 1
17 #define LED_RED 19
18 #define LED_YELLOW 20
19 #define LED_GREEN 22
20
21 #define SHUTPIN 14
22 #define SHUTTYPE SHUT1
23 #define SHUTMOTOR SHUTTYPE2
24
25 const char* ssid = "Miranda";
26 const char* password = "astralink";
27
28 volatile int flameSensorCount = 0;
29 float flameData = 0.0;
30 float totalVolume = 0.0;
31 const unsigned long interval = 1000;
32
33 #include <Arduino.h>
  
```

Gambar 4.2 Kode Program Alat Sensor



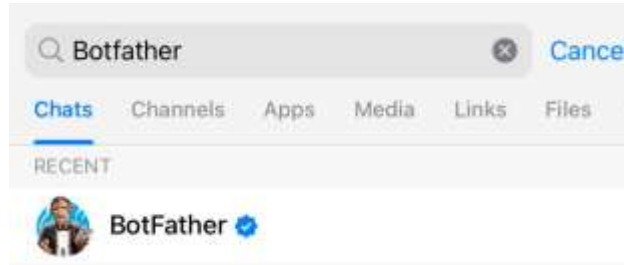
```

17:00:44.160 => Data Logger ke DB: 10.10.10.10, 10.10.10.10, 10.10.10.10, 10.10.10.10
17:00:44.162 => Mengirim Data ke DB...
17:00:44.162 => Data: {"timestamp":16161,"distance_cm":21.77,"temperature_c":25.0,"humidity_percent":65,"flow_rate_l_per_h":0,"total_volume_l":0.0000,"total_volume_cm3":0.0000}
17:00:44.164 => Data (Default) dikirim ke DB
17:00:44.162 => Response Code: 200
17:00:44.311 => Jarak: 21.77 cm | Flow: 0.00 L/h | Total: 0.00 L | Temp: 25.00 C | Hum: 65.00 % | Status: Sukses
17:00:44.314 => Jarak: 21.77 cm | Flow: 0.00 L/h | Total: 0.00 L | Temp: 25.00 C | Hum: 65.00 % | Status: Sukses
17:00:44.316 => Data Logger ke DB: 10.10.10.10, 10.10.10.10, 10.10.10.10, 10.10.10.10
17:00:44.316 => Mengirim Data ke DB...
17:00:44.316 => Data: {"timestamp":16161,"distance_cm":21.77,"temperature_c":25.0,"humidity_percent":65,"flow_rate_l_per_h":0,"total_volume_l":0.0000,"total_volume_cm3":0.0000}
17:00:44.318 => Data (Default) dikirim ke DB
17:00:44.318 => Response Code: 200
  
```

Gambar 4.3 Hasil Uji Coba Alat Sensor

4.4.2 Proses Membuat BOT Telegram

1. Buka aplikasi Telegram dan mencari akun resmi bernama **@BotFather**.



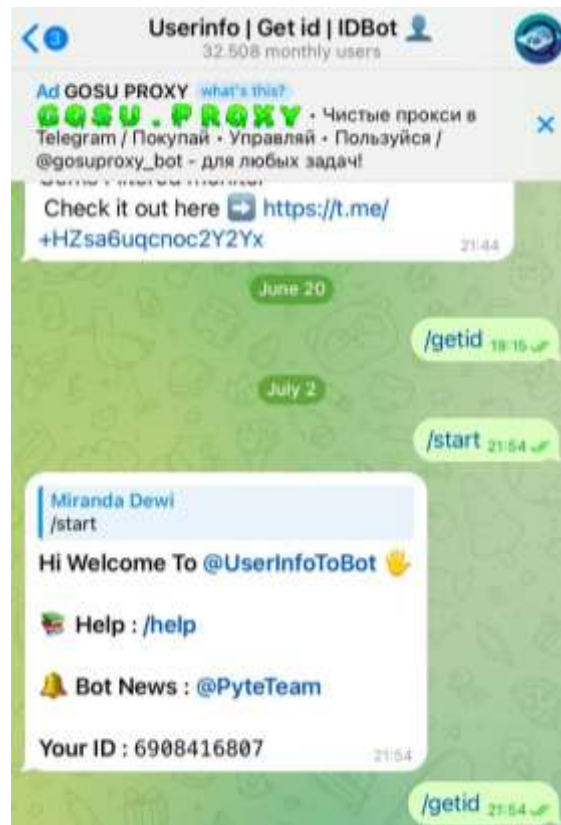
Gambar 4.4 *BOT Father*

2. Memulai interaksi dengan BotFather dengan mengirimkan perintah `/start`, Mengirimkan perintah `/newbot` untuk membuat bot baru dan Memberikan nama bot dan username yang diakhiri dengan kata "bot" atau **"PisbanBot"**.



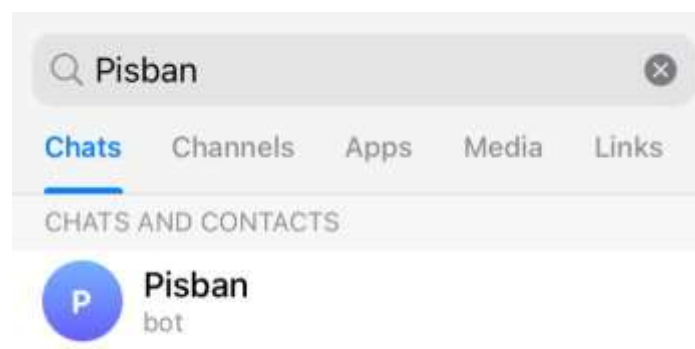
Gambar 4.5 *Membuat NamaBOT*

3. Pengguna harus mengetahui ID telegram pengguna yaitu dengan cara mencari Get My ID Bot kemudian Tekan `/start`, selanjutnya akan menampilkan id telegram pengguna.



Gambar 4.6 ID User

4. Lalu cari nama bot yang sudah dibuat maka akan diarahhkan pada tampilan telegram berikut, kemudian klik /start.



Gambar 4.7 Nama BOT

4.5 Pengoperasian Sistem

Setelah seluruh komponen terpasang dan program berhasil diunggah ke ESP32, sistem siap dijalankan untuk memantau potensi banjir secara otomatis. ESP32 dihubungkan ke sumber daya menggunakan kabel USB dari laptop, adaptor,

atau power bank, lalu terkoneksi ke jaringan WiFi sesuai konfigurasi. Saat aktif, ESP32 membaca data secara berkala dari sensor DHT11, sensor ultrasonik, flow sensor, dan float sensor. Berdasarkan data tersebut, sistem langsung mengaktifkan LED dan buzzer sebagai respons awal. LED menunjukkan kondisi lingkungan (normal, siaga, atau bahaya), sementara buzzer berbunyi saat terdeteksi risiko banjir.

Setiap data sensor disimpan dalam format CSV secara real-time, lalu dianalisis menggunakan metode TOPSIS untuk menentukan kategori risiko: Hujan Berhenti, Perlu Dipantau, atau Potensi Banjir. Jika terjadi perubahan status, sistem akan otomatis mengirimkan notifikasi ke pengguna melalui Telegram Bot. Dengan alur ini, sistem dapat merespons kondisi secara langsung, menganalisis risiko, dan memberikan peringatan dini tanpa memerlukan intervensi manual.

4.6 Proses Pengujian Sistem

Pengujian dilakukan untuk menilai kinerja sistem dalam mendeteksi potensi banjir secara otomatis berdasarkan data sensor dan analisis metode TOPSIS. Simulasi dilakukan menggunakan miniatur sungai dengan tiga variasi volume air, yaitu 150 ml, 300 ml, dan 500 ml, berfungsi mengamati perubahan kondisi lingkungan seperti tinggi air, suhu, kelembapan, dan debit aliran.

Setiap pembacaan sensor disimpan secara otomatis dalam format CSV. Setelah terkumpul 10 data, sistem melakukan analisis menggunakan metode TOPSIS untuk menentukan nilai preferensi dan mengklasifikasikan status risiko menjadi Normal (Hujan Berhenti), Siaga (Perlu Dipantau), atau Bahaya (Potensi Banjir).

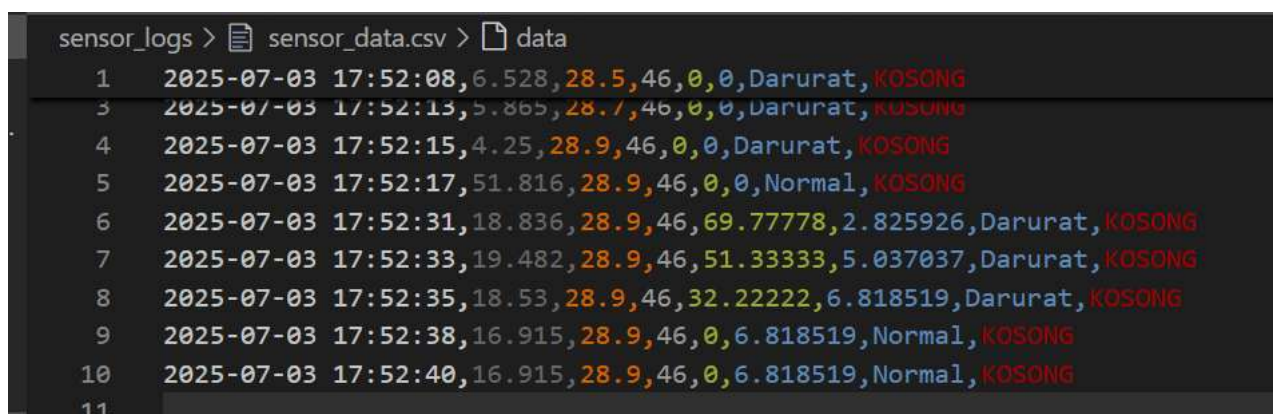
Sebelum analisis dilakukan, sistem akan memberikan respons awal melalui LED dan buzzer. Setelah hasil TOPSIS diperoleh, notifikasi risiko dikirim secara otomatis ke pengguna melalui Telegram Bot.

4.7 Proses Pengujian Alat

Pengujian pertama dilakukan dengan menuangkan air sebanyak 150 ml, 200ml dan 500ml air ke dalam miniatur sungai untuk mensimulasikan kondisi awal peningkatan volume air. Tujuan dari pengujian ini adalah untuk melihat bagaimana sistem merespons kondisi air rendah dan mendeteksi perubahan parameter lingkungan secara real-time.

Setelah Pengujian dilakukan sistem berhasil membaca dan mencatat data dari seluruh sensor secara real-time melalui Serial Monitor pada Arduino IDE. Nilai-nilai seperti *Timestamp* (Waktu dan tanggal), *Distance* (Jarak ketinggian air), *Temperature* (Suhu), *Humidity* (Kelembapan), *FlowRate* (Aliran Air), *TotalVolume* (Total jumlah air), dan *FloatStatus* ditampilkan secara berurutan.

4.7.1 Hasil Pengujian 150ml



```

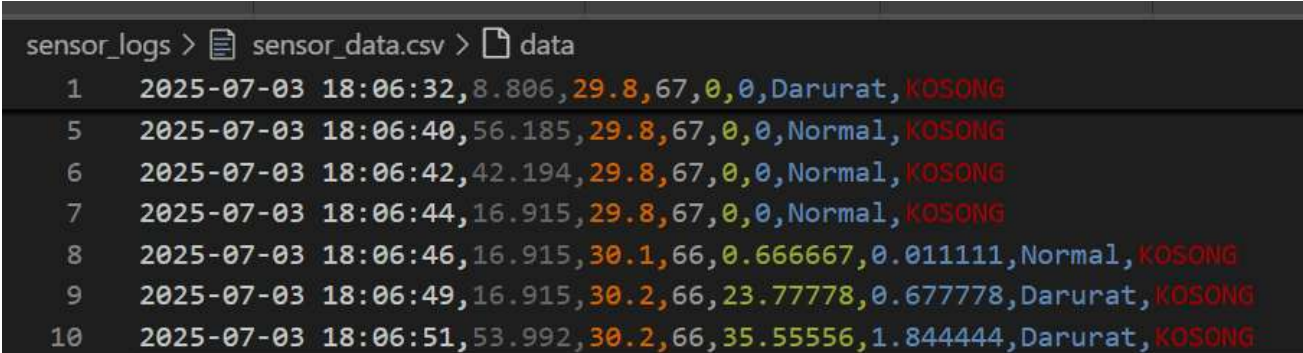
sensor_logs > [icon] sensor_data.csv > [icon] data
1  2025-07-03 17:52:08,6.528,28.5,46,0,0,Darurat,KOSONG
3  2025-07-03 17:52:13,5.865,28.7,46,0,0,Darurat,KOSONG
4  2025-07-03 17:52:15,4.25,28.9,46,0,0,Darurat,KOSONG
5  2025-07-03 17:52:17,51.816,28.9,46,0,0,Normal,KOSONG
6  2025-07-03 17:52:31,18.836,28.9,46,69.77778,2.825926,Darurat,KOSONG
7  2025-07-03 17:52:33,19.482,28.9,46,51.33333,5.037037,Darurat,KOSONG
8  2025-07-03 17:52:35,18.53,28.9,46,32.22222,6.818519,Darurat,KOSONG
9  2025-07-03 17:52:38,16.915,28.9,46,0,6.818519,Normal,KOSONG
10 2025-07-03 17:52:40,16.915,28.9,46,0,6.818519,Normal,KOSONG
11

```

Gambar 4.8 Data Pengujian 150ml Air

Berdasarkan hasil pembacaan sensor, suhu rata-rata tercatat stabil di sekitar 28.9°C dan kelembapan konsisten pada 46.0% , menunjukkan tidak ada perubahan lingkungan yang signifikan selama pengujian. Float sensor menunjukkan status "*Kosong*", menandakan air belum mencapai level pelampung. Namun, *flow rate* dan total volume menunjukkan nilai tinggi pada beberapa titik, seperti 69.77 L/min dan 2.83 liter , yang memicu klasifikasi status *Darurat* sesuai skor TOPSIS. Hasil ini menunjukkan bahwa sistem mampu mendeteksi dan merespons potensi banjir secara efektif dan *real-time*.

4.7.2 Hasil Pengujian 300ml



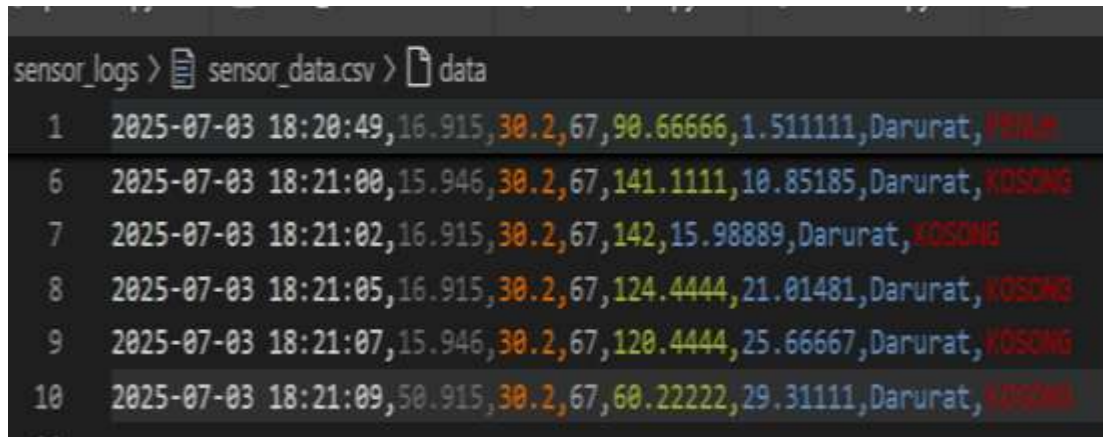
Index	Timestamp	Temp (°C)	Humidity (%)	Flow Rate (L/min)	Volume (L)	Status	Float Sensor
1	2025-07-03 18:06:32	29.8	67	0	0	Darurat	KOSONG
5	2025-07-03 18:06:40	29.8	67	0	0	Normal	KOSONG
6	2025-07-03 18:06:42	29.8	67	0	0	Normal	KOSONG
7	2025-07-03 18:06:44	29.8	67	0	0	Normal	KOSONG
8	2025-07-03 18:06:46	30.1	66	0.666667	0.011111	Normal	KOSONG
9	2025-07-03 18:06:49	30.2	66	23.77778	0.677778	Darurat	KOSONG
10	2025-07-03 18:06:51	30.2	66	35.55556	1.844444	Darurat	KOSONG

Gambar 4.9 Data Pengujian 300ml air

Berdasarkan data sensor yang diperoleh, suhu lingkungan rata-rata tercatat stabil pada angka 29.9°C , sedangkan kelembapan konsisten berada di sekitar 67% . Seluruh data menunjukkan status float sensor *Kosong* yang mengindikasikan bahwa permukaan air belum mencapai batas pelampung. Pada awal pengujian, nilai kecepatan aliran (*flow rate*) dan volume masih nol, namun pada entri berikutnya terdeteksi peningkatan signifikan, seperti flow rate sebesar 35.56 L/min dan volume 1.84 liter . Hal ini direspons sistem dengan mengklasifikasikan status sebagai "*Darurat*". Dari total entri yang diamati, sebagian besar menunjukkan status

“Normal”, sementara status “Darurat” muncul pada saat terjadi lonjakan aliran air. Temuan ini menunjukkan bahwa sistem mampu mengenali perubahan kondisi lingkungan dan memberikan klasifikasi risiko secara akurat berdasarkan data *real-time*.

4.7.3 Hasil Pengujian 500ml



Line	Timestamp	Flow Rate (L/min)	Volume (L)	Status	Remarks
1	2025-07-03 18:20:49	30.2	90.66666	Darurat	PERILAH
6	2025-07-03 18:21:00	30.2	141.1111	Darurat	KOSONG
7	2025-07-03 18:21:02	30.2	142	Darurat	KOSONG
8	2025-07-03 18:21:05	30.2	124.4444	Darurat	KOSONG
9	2025-07-03 18:21:07	30.2	120.4444	Darurat	KOSONG
10	2025-07-03 18:21:09	30.2	60.22222	Darurat	KOSONG

Gambar 4.10 Data Pengujian 500ml air

Berdasarkan data sensor yang ditampilkan, seluruh entri menunjukkan status *Darurat*, yang menandakan bahwa sistem secara konsisten mendeteksi kondisi dengan potensi banjir tinggi. Nilai suhu berada stabil di 30.2°C , dan kelembapan tercatat konstan di 67% , memperlihatkan kestabilan parameter lingkungan dari sisi suhu dan kelembapan.

Kecepatan aliran air (*flow_rate*) dan volume (*total_volume*) tercatat tinggi pada semua entri, dengan *flow_rate* berkisar antara 60.22 hingga 142.15 L/min , serta volume mencapai hingga 29.31 liter , yang secara otomatis mengindikasikan lonjakan debit air signifikan dalam waktu singkat. Hal ini sejalan dengan status darurat yang dihasilkan sistem.

Status float sensor sebagian besar menunjukkan *Kosong*, kecuali satu entri yang mencatat *Penuh*, yang menunjukkan bahwa pada saat tersebut air telah mencapai batas atas pelampung, menguatkan kondisi darurat.

4.7.3 Pengujian Sistem & Prediksi TOPSIS

Pengujian ini bertujuan untuk memastikan bahwa sistem mampu membaca data dari sensor secara real-time, menyimpannya dalam format CSV, serta melakukan analisis menggunakan metode TOPSIS untuk menentukan tingkat risiko banjir secara otomatis dan akurat.

Selama pengujian, sistem membaca data dari berbagai sensor, yaitu sensor DHT11 (suhu dan kelembapan), sensor ultrasonik (ketinggian air), flow sensor (debit dan volume air), dan float sensor (status permukaan). Semua data yang terbaca secara berkala akan ditampilkan melalui Serial Monitor dan disimpan secara otomatis ke dalam file `sensor_data.csv`.

Data yang tersimpan tersebut kemudian diproses menggunakan metode TOPSIS. Setiap kali 10 baris data baru terkumpul, sistem akan menghitung nilai preferensi (P_i) untuk masing-masing entri berdasarkan parameter yang ada. Setelah nilai preferensi dihitung, sistem akan mengambil baris terakhir dari hasil analisis TOPSIS untuk dijadikan sebagai status risiko terkini.

Tabel 4.2 Hasil Uji Analisis TOPSIS

Volume Air	Ketinggian Air	Suhu	Kelembapan	Aliran Air	Score	Status
150ml	83.0 cm	29.0 °C	46.0%	0.00 l/min	0.8500	Potensi Banjir

300ml	46.0 cm	30.0 °C	66.0%	36.00 l/min	0.1900	Hujan telah Berhenti
500ml	49.0 cm	30.0 °C	67.0%	60.00 l/min	0.5100	Perlu diPantau

Berdasarkan hasil tabel pengujian yang menggunakan variasi volume air (150 ml, 300 ml, dan 500 ml), sistem menunjukkan respons yang konsisten dan sesuai dengan kategori risiko yang telah ditentukan oleh metode TOPSIS.

Pada volume 150 ml, meskipun aliran air terdeteksi 0.00 L/min, nilai ketinggian air yang sangat tinggi (83 cm) memicu skor TOPSIS sebesar 0.8500 yang diklasifikasikan sebagai Potensi Banjir. Hal ini menandakan bahwa ketinggian air merupakan indikator dominan dalam kondisi tersebut.

Sebaliknya, **pada volume 300 ml**, meskipun debit air meningkat hingga 36.00 L/min, skor TOPSIS menurun drastis menjadi 0.1900 dan menghasilkan status Hujan Telah Berhenti. Hal ini disebabkan oleh menurunnya ketinggian air ke 46.0 cm dan suhu yang lebih stabil, yang menyebabkan sistem menganggap kondisi tidak kritis.

Pada volume 500 ml, meskipun aliran air tercatat 60.00 L/min, nilai skor TOPSIS yang didapat adalah 0.5100 yang dikategorikan sebagai Perlu Dipantau. Ini menunjukkan bahwa sistem dapat mengenali kondisi antara stabil dan rawan secara akurat.

Dengan demikian, dapat disimpulkan bahwa sistem prediksi yang dibangun mampu mendeteksi dan mengklasifikasikan tingkat risiko potensi banjir secara akurat berdasarkan kombinasi data sensor, dan metode TOPSIS berhasil menjalankan fungsinya dalam pengambilan keputusan otomatis.

4.7.4 Tampilan Notifikasi Telegram

Salah satu fitur utama dari sistem ini adalah kemampuannya dalam mengirimkan notifikasi otomatis melalui Telegram Bot kepada pengguna, sebagai bentuk peringatan dini potensi banjir. Notifikasi ini dikirim setiap kali terjadi perubahan status risiko berdasarkan hasil analisis data sensor menggunakan metode TOPSIS.

Pesan yang dikirimkan mencakup informasi lengkap meliputi:

- Waktu kejadian (timestamp)
- Nilai skor TOPSIS
- Kategori status (Hujan Berhenti, Perlu Dipantau, atau Potensi Banjir)
- Data sensor pendukung seperti ketinggian air, suhu, kelembapan, dan aliran air.



Gambar 4.11 *Notifikasi Telegram*

Pesan dikirim secara *real-time* ke dalam file csv dan hanya muncul ketika ada perubahan status dari data sebelumnya. Jika tidak ada perubahan status, sistem akan tetap siaga dan tidak mengirimkan pesan apapun, guna menghindari spam yang berlebihan. Fitur ini memungkinkan pengguna untuk mendapatkan informasi terkini terkait kondisi lingkungan secara langsung, sehingga dapat melakukan tindakan cepat jika risiko banjir terdeteksi meningkat.

4.7 Evaluasi

Evaluasi dilakukan untuk menilai kinerja sistem peringatan dini potensi banjir yang telah dikembangkan, baik dari segi akurasi klasifikasi risiko maupun keandalan notifikasi dan respon lokal (LED dan buzzer). Evaluasi dilakukan

berdasarkan hasil uji coba sistem dalam lingkungan prototipe dengan beberapa variasi data sensor dan skenario aliran air.

Berdasarkan hasil pengujian yang dilakukan, sistem mampu mengklasifikasikan tingkat risiko secara tepat menggunakan metode TOPSIS, dengan nilai *preference score* (P_i) yang secara konsisten mencerminkan kondisi lingkungan. Misalnya, ketika aliran air (flow rate) dan volume meningkat secara signifikan, serta jarak sensor ultrasonik menunjukkan ketinggian air yang mendekati batas, sistem memberikan nilai $P_i > 0.6$ dan mengklasifikasikan kondisi sebagai "Potensi Banjir". Sebaliknya, ketika seluruh parameter berada dalam ambang aman, nilai P_i cenderung < 0.3 , yang dikategorikan sebagai "Hujan Berhenti".

Selain itu, evaluasi terhadap sistem notifikasi menunjukkan bahwa Telegram Bot berhasil mengirimkan peringatan secara real-time kepada pengguna, terutama saat status "Potensi Banjir" terdeteksi. Indikator lokal berupa LED merah dan buzzer juga aktif secara tepat sesuai status risiko. Hal ini menunjukkan bahwa sistem tidak hanya mampu melakukan klasifikasi risiko, tetapi juga dapat merespon dengan cepat melalui berbagai media output.

Namun demikian, terdapat beberapa catatan dalam proses evaluasi:

- a. Sensor DHT11 menunjukkan sedikit fluktuasi data pada suhu dan kelembapan jika diletakkan terlalu dekat dengan sumber panas atau aliran udara.
- b. Flow sensor perlu dikalibrasi ulang secara berkala untuk mempertahankan keakuratan perhitungan debit dan volume air.
- c. Interval pengiriman data perlu disesuaikan agar tidak terjadi penumpukan request ke server Flask.

Secara keseluruhan, sistem telah bekerja sesuai dengan tujuan awal penelitian, yaitu memberikan peringatan dini terhadap potensi banjir secara otomatis, real-time, dan berbasis IoT. Evaluasi ini menjadi dasar untuk penyempurnaan sistem agar dapat diimplementasikan dalam skala yang lebih luas di masa depan.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian, maka dapat disimpulkan bahwa sistem yang dikembangkan telah berjalan sesuai dengan perancangan dan mampu memberikan informasi kondisi secara real-time. Hasil ini menunjukkan bahwa pendekatan yang digunakan dalam penelitian efektif dalam mendukung tujuan sistem. sebagai berikut:

- a. Sistem yang dikembangkan berhasil membaca dan mengolah data dari sensor ketinggian air (ultrasonik), sensor suhu dan kelembapan (DHT11), sensor aliran air (flow sensor), serta sensor pelampung (float switch) secara real-time.
- b. Metode TOPSIS mampu digunakan secara efektif dalam menentukan tingkat risiko banjir berdasarkan data sensor yang dikumpulkan. Sistem dapat mengklasifikasikan kondisi menjadi tiga kategori: Hujan Berhenti, Perlu Dipantau, dan Potensi Banjir.
- c. Ketika nilai preferensi (P_i) melebihi ambang batas > 0.6 , sistem secara otomatis mengirimkan notifikasi melalui Telegram dan mengaktifkan buzzer serta LED sebagai indikator peringatan lokal. Ini membuktikan sistem dapat memberikan peringatan secara cepat dan tepat.
- d. Evaluasi menunjukkan bahwa sistem mampu merespons perubahan kondisi lingkungan dengan akurat, dan setiap komponen bekerja secara sinergis sesuai dengan perancangan awal.

5.2 Saran

Untuk pengembangan lebih lanjut, disarankan agar sistem dilengkapi dengan kemampuan pemantauan secara real-time berbasis cloud agar data dapat diakses dari berbagai lokasi. Sebagai berikut :

- a. Perlu dilakukan penyesuaian dan kalibrasi sensor secara berkala, terutama flow sensor dan sensor ultrasonik, untuk memastikan keakuratan data yang dikumpulkan.
- b. Pengembangan ke depan sebaiknya menambahkan sumber daya cadangan (misalnya baterai) agar sistem tetap berfungsi saat terjadi pemadaman listrik.
- c. Diperlukan implementasi sistem dalam skala yang lebih luas dan lingkungan nyata (di luar prototipe) untuk menguji kehandalan sistem dalam kondisi cuaca dan aliran sungai yang sebenarnya.
- d. Sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur prediksi cuaca menggunakan API eksternal dan menyimpan log data ke cloud storage untuk pemantauan jangka panjang.
- e. Visualisasi hasil analisis TOPSIS dan status banjir juga bisa ditampilkan melalui dashboard web atau aplikasi mobile agar informasi dapat diakses oleh pengguna dengan lebih mudah dan ce

DAFTAR PUSTAKA

- Adi Gunawan, L., Imam Agung, A., Widyartono, M., & Isnur Haryudo, S. (n.d.). *RANCANG BANGUN PEMBANGKIT LISTRIK TENAGA SURYA PORTABLE*.
- Arifin, T. N., Febriyani Pratiwi, G., & Janrafsasih, A. (n.d.). *SENSOR ULTRASONIK SEBAGAI SENSOR JARAK*. <http://jurnal.undira.ac.id/index.php/jurnaltera/>
- Ayu Syahfitri. (2025). Internet of Things (IoT), Sejarah, Teknologi, dan Penerapannya. *Uranus : Jurnal Ilmiah Teknik Elektro, Sains Dan Informatika*, 3(1), 113–120. <https://doi.org/10.61132/uranus.v3i1.667>
- Bianto, A., & Aprillya, M. R. (n.d.). *Bianto, Sistem Pendukung Keputusan Identifikasi Daerah Potensi Banjir Dengan Metode Multi Attribute Utility Theory (Studi Kasus: Kabupaten Lamongan) 116 Sistem Pendukung Keputusan Identifikasi Daerah Potensi Banjir Dengan Metode Multi Attribute Utility Theory (Studi Kasus: Kabupaten Lamongan)*.
- Binsar Bakkara. (2024, November 26). *Rescuers recover 20 bodies from flash floods and landslides on Indonesia's Sumatra island*. <https://apnews.com/Article/Indonesia-North-Sumatra-Flash-Floods-Landslide-6388a36ab374b62a8d130439ea83e7ba>. <https://apnews.com/article/indonesia-north-sumatra-flash-floods-landslide-6388a36ab374b62a8d130439ea83e7ba>
- Daya, K., Berbagai, D., Rangkaian, K., Dalam, D., Memenuhi, R., Satu, S., & Untuk Menyelesaikan, P. (2021). *PERANCANGAN LAMPU LED BESERTA ANALISIS*.
- Efendy, Z., Rahimullaily, R., & Aini, V. N. (2023). Sistem Pendukung Keputusan Menggunakan Metode TOPSIS Berbasis Aplikasi (Studi kasus Keluarga Miskin di Kelurahan Mata Air Kecamatan Padang Selatan). *Remik*, 7(1), 142–156. <https://doi.org/10.33395/remik.v7i1.11971>

- Hadrianti A, Y. J., & Wajidi, F. (2023). SISTEM PENDUKUNG KEPUTUSAN PENERIMA BANTUAN BENCANA ALAM MENGGUNAKAN METODE TOPSIS (STUDI KASUS: KANTOR BADAN PENANGGULANGAN BENCANA DAERAH (BPBD) KABUPATEN MAJENE). *Jurnal Informatika Dan Komputer*, 185(2), 185–194. <https://doi.org/10.35508/jicon.v11.i2.11178>
- Kumar, Prof. V., Kadnar, Mr. N., Gosavi, Mr. A., & Jadhav, Mr. O. (2023). IOT-Based Flood Monitoring and Alarm System. *International Journal for Research in Applied Science and Engineering Technology*, 11(5), 5142–5146. <https://doi.org/10.22214/ijraset.2023.52858>
- Mahanin Tyas, U., Apri Buckhari, A., Studi Pendidikan Teknologi Informasi, P., & Studi Pendidikan Teknologi dan Kejuruan, P. (2023). *IMPLEMENTASI APLIKASI ARDUINO IDE PADA MATA KULIAH SISTEM DIGITAL* (Vol. 1, Issue 1).
- Mubarok, Y. H., Rahmat Hidayat, C., & Sumaryana, Y. (n.d.). *Sistem Pendukung Keputusan Pemilihan Varietas Bibit Padi Unggul Menggunakan Metode TOPSIS (Studi Kasus : BPP Cibeureum Kota Tasikmalaya) Decision Support System For Selecting Superior Rice Seed Varieties Using The TOPSIS Method (Case Study: BPP Cibeureum Tasikmalaya City)*. 2024(1), 61–72. <https://doi.org/10.51132/teknologika.v14i1>
- Mulyanto, A. D. (2020). Pemanfaatan Bot Telegram Untuk Media Informasi Penelitian. *MATICS*, 12(1), 49. <https://doi.org/10.18860/mat.v12i1.8847>
- Niniekarmini, & Rangga Firmansyah. (2024, December 10). *Landslides and flash floods hit Indonesia's Java island, leaving 10 dead and 2 missing*.

- <https://Apnews.Com/Article/Indonesia-West-Java-Rain-Landslide-Flash-Floods-Cd3ea5fb6ae47b53aa87ba307c73a4c3>.
- Noviyanti, L., Raya Dompok, J., Bukit Bestari, K., Tanjung Pinang, K., & Riau, K. (2024). Pemanfaatan Aplikasi Telegram Sebagai Platform Bisnis Digital : Perspektif Pelajar Sebagai Pengguna Aktif Telegram. *Jurnal Bisnis Kreatif Dan Inovatif*, 1(4). <https://doi.org/10.61132/jubikin.v1i4.409>
- Rahmansyah, N., Kom, S., Kom, M., & Lusinia, S. A. (2021). *Buku Ajar SISTEM PENDUKUNG KEPUTUSAN*. <http://jurnal.pustakagalerimandiri.co.id>
- Ramu, K., Ramachandran, M., & Selvam, M. (2022). Microcontroller Based Sensor Interface and Its Investigation. *Electrical and Automation Engineering*, 1(2), 92–97. <https://doi.org/10.46632/eae/1/2/4>
- Reuters. (2024, August 27). *Floods kill 13 in Indonesia's North Maluku, two injured*. [Http://Reuters.Com/World/Asia-Pacific/Floods-Kill-7-Indonesias-North-Maluku-Some-Missing-2024-08-25/?Utm_source=chatgpt.Com](http://Reuters.Com/World/Asia-Pacific/Floods-Kill-7-Indonesias-North-Maluku-Some-Missing-2024-08-25/?Utm_source=chatgpt.Com).
- Sabran, A. M., & Suwardoyo, U. (2024). *Prototipe Notifikasi Banjir Berbasis IoT* (Vol. 4, Issue 3). <https://jurnal.umpar.ac.id/index.php/sylog>
- Setiadi, D. R., Dalimunthe, E. R., Putri, N. U., & Zain, A. (2024). *Implementasi Metode Fuzzy Sugeno Pada Prototipe Pendeteksi Banjir* (Vol. 18, Issue 03).
- Setiawan, Y., & Wiharko, T. (2023). Implementasi Metode MOORA Pada Sistem Pendukung Keputusan Rekomendasi Destinasi Wisata Pendakian Gunung di Bandung Raya. *Digital Transformation Technology*, 3(2), 515–523. <https://doi.org/10.47709/digitech.v3i2.2924>
- Snyders, S. P., Wibowo, S. A., & Orisa, M. (2024). APLIKASI PENENTUAN SUSU FORMULA UNTUK BAYI BARU LAHIR MENGGUNAKAN METODE

TOPSIS BERBASIS WEB. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 2).

Soselisa, L. A., & Dasawaty, E. S. (2023). Implementasi Sistem Alarm Peringatan Dini untuk Mendeteksi Banjir di Desa Tongute Ternate Berbasis Arduino Uno. In *Jurnal Informatika dan Bisnis* (Vol. 12, Issue 2).

Suhardono, S., Sagara, M. R. N., & Suryawan, I. W. K. (2024). Sistem Peringatan Dini untuk Banjir Rob dan Sampah Laut: Analisis SWOT. *Journal of Marine Research*, 13(3), 419–427. <https://doi.org/10.14710/jmr.v13i3.40850>

Tantowi, D., & Kurnia, Y. (2020). Simulasi Sistem Keamanan Kendaraan Roda Dua Dengan Smartphone dan GPS Menggunakan Arduino. In *JURNAL ALGOR* (Vol. 1, Issue 2). <https://jurnal.buddhidharma.ac.id/index.php/algor/index>

LAMPIRAN

LAMPIRAN 1. SK-2 Penetapan Dosen Pembimbing



**PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 377/IL3-AU/UMSU-09/F/2025**

Assalamu 'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi : Teknologi Informasi
Pada tanggal : 19 Februari 2025

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama : Miranda Dewi
NPM : 2109020013
Semester : VII (Tujuh)
Program studi : Teknologi Informasi
Judul Proposal / Skripsi : Implementasi Sistem Pendukung Keputusan Berbasis IoT dengan Metode TOPSIS untuk Peringatan Dini Potensi Bencana Banjir

Dosen Pembimbing : Yohanni Syahra, S.Si., M.Kom

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. Proyek Proposal / Skripsi dinyatakan " BATAL " bila tidak selesai sebelum Masa Kadaluarsa tanggal : 19 Februari 2026
4. Revisi judul.....

Wassalamu 'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
Pada Tanggal : 20 Sya'ban 1446 H
19 Februari 2025M



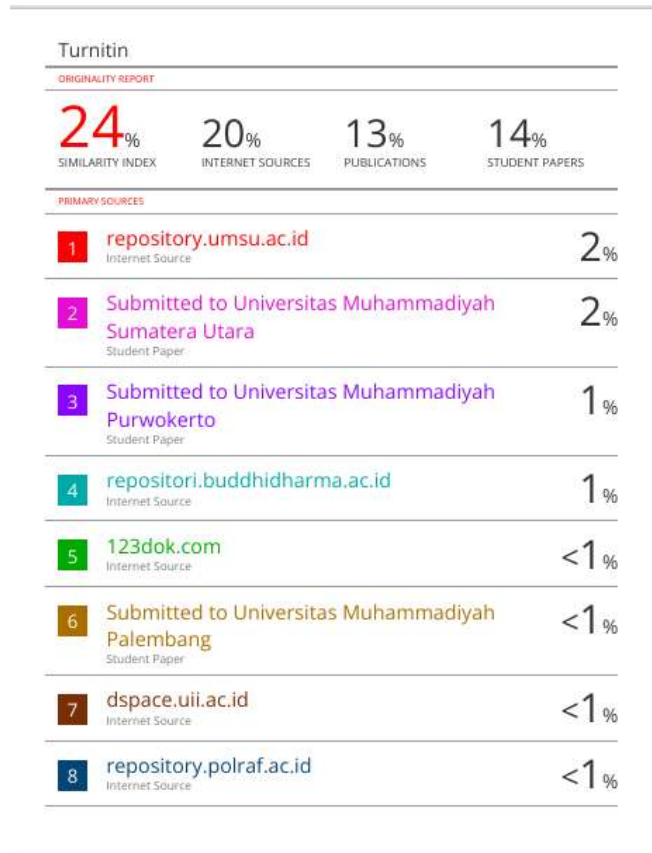
Dekan

Dr. A. Khawarizmi, M.Kom.
NIDN : 0127099201

Cc. File



LAMPIRAN 3. Turnitin



LAMPIRAN 4. Kode Program

```
#include <Wire.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <DHT.h>
#include <SPIFFS.h>
#include <FS.h>

// === PIN KONFIGURASI ===
#define TRIG_PIN 4
#define ECHO_PIN 5
#define FLOW_SENSOR_PIN 23
#define BUZZER_PIN 13
#define FLOAT_PIN 18

#define LED_DEFAULT 2
#define LED_RED 19
#define LED_YELLOW 21
#define LED_GREEN 22

#define DHTPIN 14
#define DHTTYPE DHT11
```

```

DHT dht(DHTPIN, DHTTYPE);

const char* ssid = "Miranda";
const char* password = "adnarim03";

volatile int flowPulseCount = 0;
float flowRate = 0.0;
float totalVolume = 0.0;
const unsigned long interval = 1000;

String status = "Normal";
String lastStatus = "Normal";

// REALTIME
unsigned long lastDataLogTime = 0;
const unsigned long dataLogInterval = 2000; // 2 detik

void setup() {
  Serial.begin(115200);
  delay(1000);

  if (!SPIFFS.begin(true)) {
    Serial.println("Gagal mount SPIFFS");
    return;
  }

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(FLOW_SENSOR_PIN, INPUT_PULLUP);
  pinMode(FLOAT_PIN, INPUT_PULLUP);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(LED_DEFAULT, OUTPUT);
  pinMode(LED_RED, OUTPUT);
  pinMode(LED_YELLOW, OUTPUT);
  pinMode(LED_GREEN, OUTPUT);

  digitalWrite(LED_RED, LOW);
  digitalWrite(LED_YELLOW, LOW);
  digitalWrite(LED_GREEN, LOW);

  attachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN), flowPulseCounter,
    FALLING);
  dht.begin();

  Serial.println("Menghubung WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("\nWiFi OK: " + WiFi.localIP().toString());
  createCSVHeader();

```

```

    delay(2000);
}

void loop() {
    long duration = getUltrasonicDistance();
    float distance = duration * 0.034 / 2;
    float temperature = dht.readTemperature();
    float humidity = dht.readHumidity();

    calculateFlowRate();
    totalVolume += flowRate / 60.0;

    String newStatus = checkStatus(distance);
    status = newStatus;

    if (status == "Darurat") {
        tone(BUZZER_PIN, 1000);
    } else if (status == "Siaga") {
        tone(BUZZER_PIN, 1500);
        delay(150);
        noTone(BUZZER_PIN);
    } else {
        noTone(BUZZER_PIN);
    }

    Serial.print("Jarak: "); Serial.print(distance);
    Serial.print(" cm | Flow: "); Serial.print(flowRate);
    Serial.print(" L/min | Total: "); Serial.print(totalVolume);
    Serial.print(" L | Temp: "); Serial.print(temperature);
    Serial.print(" C | Humi: "); Serial.print(humidity);
    Serial.print(" % | Status: "); Serial.println(status);

    unsigned long currentMillis = millis();

    // LOGGING SETIAP 2 DETIK (REALTIME)
    if (currentMillis - lastDataLogTime >= dataLogInterval) {
        lastDataLogTime = currentMillis;

        // SATU FUNGSI LOGGING SAJA
        logDataToCSV(distance, temperature, humidity, flowRate, totalVolume, status);
        sendDataToPC(distance, temperature, humidity, flowRate, totalVolume, status);
    }

    delay(1000);
}

long getUltrasonicDistance() {U
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    return pulseIn(ECHO_PIN, HIGH);
}

```

```

}

void flowPulseCounter() {
    flowPulseCount++;
}

void calculateFlowRate() {
    const float calibrationFactor = 4.5;
    flowRate = (flowPulseCount / calibrationFactor) / (interval / 1000.0);
    flowPulseCount = 0;
}

String checkStatus(float distance) {
    if (distance < 10 || flowRate > 5 || totalVolume > 20) {
        digitalWrite(LED_RED, HIGH);
        digitalWrite(LED_YELLOW, LOW);
        digitalWrite(LED_GREEN, LOW);
        return "Darurat";
    } else if ((distance >= 10 && distance <= 15) || (flowRate >= 3 && flowRate <= 5) ||
(totalVolume >= 10 && totalVolume <= 20)) {
        digitalWrite(LED_RED, LOW);
        digitalWrite(LED_YELLOW, HIGH);
        digitalWrite(LED_GREEN, LOW);
        return "Siaga";
    } else {
        digitalWrite(LED_RED, LOW);
        digitalWrite(LED_YELLOW, LOW);
        digitalWrite(LED_GREEN, HIGH);
        return "Normal";
    }
}

void createCSVHeader() {
    File file = SPIFFS.open("/sensor_data.csv", FILE_READ);
    if (!file) {
        file.close();
        file = SPIFFS.open("/sensor_data.csv", FILE_WRITE);
        if (file) {
            file.println("Timestamp,Distance_cm,Temperature_C,Humidity_%,FlowRate_L/min,
TotalVolume_L,Status,FloatStatus");
            file.close();
            Serial.println("CSV header dibuat");
        }
    } else {
        file.close();
        Serial.println("CSV file sudah ada");
    }
}

void logDataToCSV(float distance, float temperature, float humidity, float flowRate, float
totalVolume, String status) {
    File file = SPIFFS.open("/sensor_data.csv", FILE_APPEND);
    if (file) {

```

```

unsigned long timestamp = millis();

// PASTIKAN FLOAT STATUS DIBACA DENGAN BENAR
bool isFloatLow = digitalRead(FLOAT_PIN);
String floatStatus = isFloatLow ? "KOSONG" : "PENUH";

String dataLine = String(timestamp) + "," +
    String(distance, 2) + "," +
    String(temperature, 2) + "," +
    String(humidity, 2) + "," +
    String(flowRate, 2) + "," +
    String(totalVolume, 2) + "," +
    status + "," +
    floatStatus;

file.println(dataLine);
file.close();
Serial.println("Data logged to CSV: " + dataLine);
} else {
    Serial.println("Gagal buka file CSV untuk menulis");
}
}

void sendDataToPC(float distance, float temperature, float humidity, float flowRate, float
totalVolume, String status) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin("http://172.20.10.5:5000/log");

        http.addHeader("Content-Type", "application/json");

        StaticJsonDocument<256> json;
        json["timestamp"] = millis();
        json["distance_cm"] = distance;
        json["temperature_c"] = temperature;
        json["humidity_percent"] = humidity;
        json["flow_rate_lpm"] = flowRate;
        json["total_volume_l"] = totalVolume;
        json["status"] = status;
        json["float_status"] = digitalRead(FLOAT_PIN) ? "KOSONG" : "PENUH";

        String requestBody;
        serializeJson(json, requestBody);

        Serial.println("Mengirim data ke PC...");
        Serial.println("Data: " + requestBody);

        int httpResponseCode = http.POST(requestBody);
        if (httpResponseCode > 0) {
            String response = http.getString();
            Serial.println("✓ Data berhasil dikirim ke PC");
            Serial.println("Response Code: " + String(httpResponseCode));
        }
    }
}

```

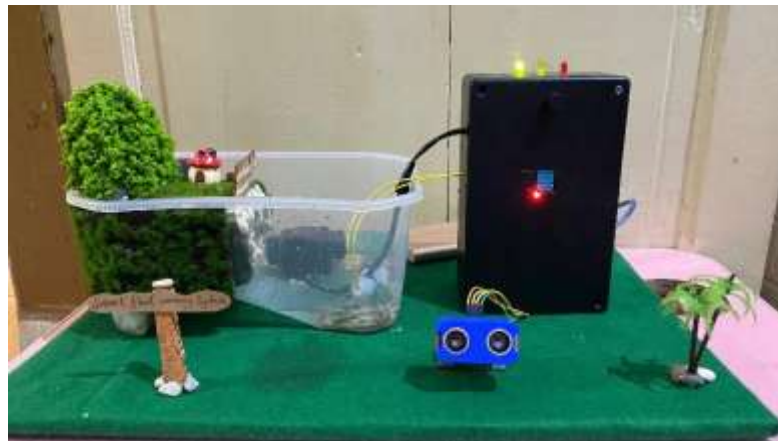
```

    } else {
      Serial.println("X Gagal kirim data ke PC");
      Serial.println("Error Code: " + String(httpResponseCode));
    }

    http.end();
  } else {
    Serial.println("WiFi tidak terhubung!");
  }
}

```

LAMPIRAN 5. Alat Peringatan Potensi Banjir



LAMPIRAN 6. Notifikasi Telegram

