

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Banjir dan Penyebabnya**

##### **2.1.1 Banjir**

Banjir terjadi ketika suatu daerah terendam karena volume air yang meningkat, biasanya terjadi saat musim hujan dengan curah hujan yang tinggi dan berkelanjutan. Pada bulan Januari 2024 beberapa daerah mengalami kejadian banjir, setidaknya ada 101 kabupaten/kota di 28 provinsi yang telah terkena bencana banjir.

Banjir pada bulan Januari paling banyak terjadi di Pulau Sumatera, kemudian Pulau Kalimantan, Jawa, Sulawesi, Maluku dan Papua. Ada 125 laporan kejadian banjir yang terjadi selama bulan Januari, paling banyak terjadi di Provinsi Riau dengan 19 kejadian, Sumatera Selatan 14 kejadian, Jawa Barat dan Kalimantan Barat masing-masing 9 kejadian, Jawa Tengah 8 kejadian, Kalimantan Selatan dan Kalimantan Tengah 7 kejadian, Jambi dan Sumatera Utara 6 kejadian, Sulawesi Tengah dan Sumatera Barat 5 kejadian, Jawa Timur dan Sulawesi Selatan 4 kejadian, DKI Jakarta 3 kejadian, Banten, Maluku Utara, Sulawesi Tenggara, Papua Barat, dan Aceh masing-masing 2 kejadian, Lampung, Sulawesi Barat, Kepulauan Riau, Papua Tengah, Bengkulu, Kepulauan Bangka Belitung, Papua, Kalimantan Utara dan Sulawesi Utara masing-masing kejadian. Berikut gambar 2.1 adalah grafik jumlah korban akibat bencana banjir bulan januari.



Gambar 2. 1 Grafik Jumlah Korban Bencana Banjir Bulan Januari (BNPB, 2024)

Berdasarkan data yang dihimpun oleh Pusat Pengendalian Operasi BNPB banjir yang terjadi pada bulan Januari mengakibatkan 7 orang meninggal, 2 orang luka, dan 1.150.771 orang terdampak dan mengungsi. Selain korban jiwa, banjir juga mengakibatkan dampak kerusakan diantaranya 46 unit rumah rusak berat, 41 unit rumah rusak sedang, 9.290 unit rumah rusak ringan dan 244.679 unit rumah terendam. Berikut tabel 2.1 jumlah bencana pada bulan januari 2024.

Tabel 2. 1 Kejadian Bencana di Bulan Januari 2024 (BNPB, 2024)

NO	Jenis Bencana	Jumlah Kejadian	Meninggal	Luka/Sakit	Mengungsi	Kerusakan Rumah			
						Berat	Sedang	Ringan	Terendam
				Jiwa					Unit
1	Gempa Bumi	2	-	-	110	-	4	21	-
2	Erupsi Gunung Api	1	4	-	13.118	-	-	-	-
3	Kebakaran Hutan	1	-	-	-	-	-	-	-
4	Banjir	125	7	2	1.150.771	46	41	9.290	244.679
5	Tanah Longsor	11	3	13	2.255	42	14	20	-
6	Cuaca Ekstrem	57	6	25	8.581	102	223	1.189	-
	Total	197	20	40	1.174.835	190	282	10.520	244.679

Pada bulan Januari 2024, lima provinsi dengan kejadian bencana terbanyak adalah Provinsi Jawa Barat sebanyak 59 kejadian, Jawa Tengah sebanyak 28 kejadian, Riau sebanyak 19 kejadian, Jawa Timur sebanyak 17 kejadian, dan Sumatera Selatan sebanyak 14 kejadian. Di Provinsi Jawa Barat dan Jawa Tengah,

kejadian bencana yang mendominasi adalah cuaca ekstrem dengan masing-masing terjadi 14 kejadian dan 16 kejadian. Sedangkan di Provinsi Riau selama Januari terjadi bencana banjir dengan 19 kejadian.

Selain itu, 8 Januari 2024 hujan dengan intensitas tinggi melanda Kabupaten Bandung hingga menyebabkan beberapa wilayah terdampak banjir yang menyebabkan 74.067 orang terdampak dan lebih dari 677 orang mengungsi, sedangkan dampak kerusakan yang terjadi diantaranya 41 unit rumah rusak dan 9.186 unit rumah terendam akibat banjir. Banjir yang terjadi selain akibat curah hujan yang tinggi faktor lain diantaranya tanggul jebol, luapan Sungai, dan saluran drainase yang tidak berfungsi dengan baik (Muhari dkk., 2023).

### **2.1.2 Faktor Penyebabnya**

#### **1. Curah Hujan Tinggi**

Curah hujan yang tinggi dalam waktu singkat adalah penyebab utama kebanjiran. Ketika curah hujan yang besar terjadi dalam periode singkat, sistem drainase alami dan buatan sering kali tidak mampu menangani volume air yang datang dengan cepat, akibatnya air menggenangi daerah sekitarnya, terutama jika drainase tidak efektif.

#### **2. Penyempitan Aliran Sungai**

Pembangunan di sepanjang sungai sering kali menyebabkan penyempitan aliran sungai. Bangunan, jalan, dan infrastruktur lainnya dapat menghalangi aliran air dan mengurangi kapasitas sungai untuk menampung volume air yang tinggi.

#### **3. Topografi Wilayah**

Wilayah dengan topografi curam atau dataran rendah rentan terhadap banjir. Dataran rendah cenderung menjadi tempat genangan air saat hujan lebat terjadi. Di

sisi lain, lereng curam dapat mempercepat aliran air permukaan, meningkatkan risiko banjir di daerah yang lebih rendah.

#### 4. Pembangunan yang Tidak Terencana

Pembangunan perkotaan yang tidak terencana sering kali mengabaikan sistem drainase yang memadai. Tanpa infrastruktur drainase yang cukup, air hujan tidak dapat dialirkan dengan baik, meningkatkan risiko banjir di kawasan perkotaan (Muhari dkk., 2023).

## **2.2 Kebakaran dan Faktor Penyebabnya**

### **2.2.1 Kebakaran**

Kebakaran akibat pembakaran sampah sembarangan adalah praktik yang tidak aman dan merugikan yang melibatkan pembakaran sampah di lokasi yang tidak terkontrol atau tidak diizinkan. Praktik ini sering kali dilakukan oleh individu atau kelompok yang tidak memiliki akses atau akses terbatas ke layanan pengelolaan sampah yang formal, seperti pengumpulan sampah atau tempat pembuangan sampah yang terkendali. Penyebab utama dari kebakaran akibat pembakaran sampah sembarangan dapat bervariasi, tetapi praktik ini umumnya dipicu oleh beberapa faktor yang saling terkait.

Pada hari Minggu (29/10) sekitar pukul 14.15 WIB, terjadi kebakaran di Tempat Pengolahan Sampah Terpadu (TPST) Bantargebang, yang berlokasi di Kelurahan Cikiwul, Kelurahan Sumur Batu, Kecamatan Bantargebang, Kota Bekasi, Jawa Barat. Berikut gambar 2.2 kebakaran tempat pengolahan sampah terpadu.



Gambar 2. 2 Kebakaran Tempat Pengolahan Sampah Terpadu (BNPB, 2023)

Di zona 2 TPST Bantargebang, di sekitar *Power House* dan Pembangkit Listrik Tenaga Sampah (PLTSA), kebakaran pertama kali terjadi. Meskipun tidak ada korban jiwa, insiden ini menghasilkan asap hitam pekat yang menyebar di langit dan menyelimuti Bantargebang.

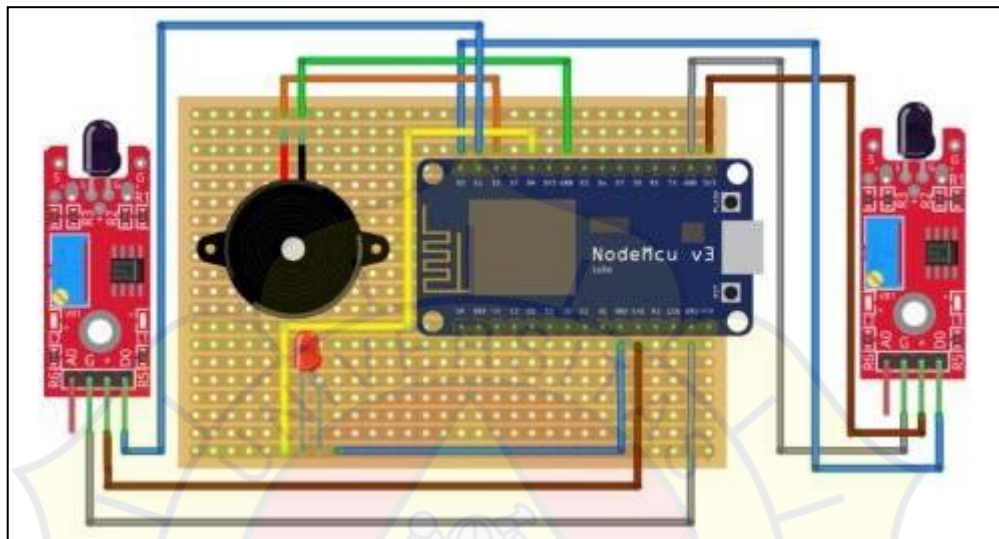
Karsono, sebagai Koordinator Pusat Pengendali Operasi (Pusdalops) Badan Penanggulangan Bencana Daerah (BPBD) Kota Bekasi, mengungkapkan bahwa tim gabungan telah dikerahkan dengan mobil pemadam kebakaran dan tangki air guna memadamkan api tersebut. Selain itu, ia juga menyatakan bahwa BPBD setempat telah mendistribusikan masker kepada penduduk di sekitar area untuk mengurangi dampak kabut asap yang menyebar ke pemukiman (BNPB, 2023).

### 2.2.2 Faktor Penyebabnya

Banyak faktor yang dapat menyebabkan kebakaran, seperti korsleting listrik, ledakan kompor gas, atau puntung rokok. Kebakaran sering kali terdeteksi saat api mulai berkobar, asap menjadi tebal, atau asap sudah keluar dari bangunan. Karena

risiko kebakaran dapat terjadi kapan saja, sistem keamanan sangat penting untuk mengurangi kemungkinan kebakaran serta dampak kerugian yang lebih besar. (Isyanto dkk., 2020).

### 2.2.3 Arsitektur Perangkat *IoT*



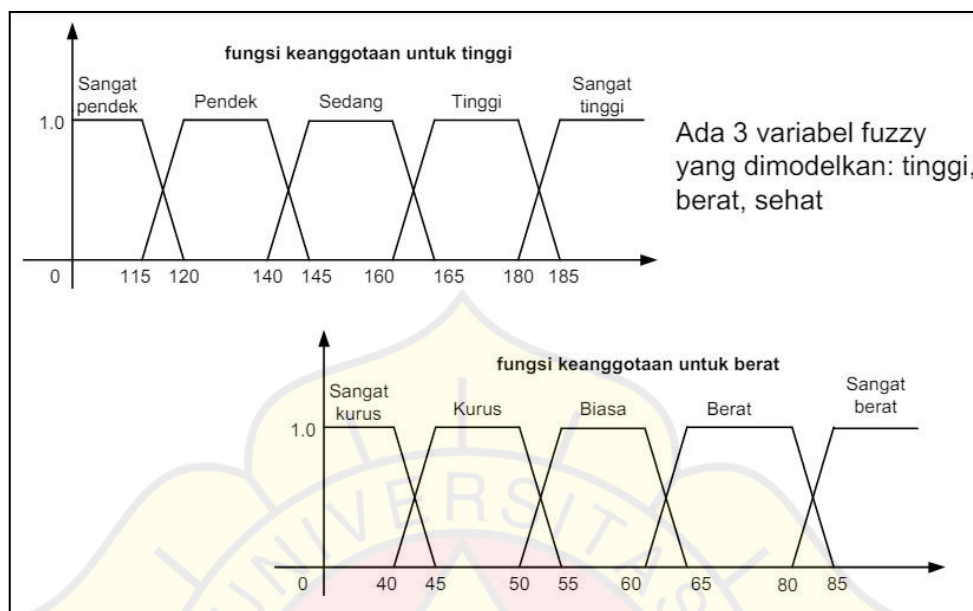
Gambar 2. 3 Rangkaian Prototipe Alat Pendeteksi Kebakaran (Fritzing)

Berdasarkan gambar 2.3 desain prototipe perangkat deteksi kebakaran bertujuan untuk mendeteksi api dan asap saat terjadi kebakaran. Ketika perangkat mendeteksi kebakaran, akan mengaktifkan bunyi alarm untuk memberikan peringatan awal, yang memungkinkan evakuasi secara cepat.

## 2.3 Metode *Fuzzy*

Metode *Fuzzy* atau Logika *fuzzy* adalah cabang dari logika matematika yang memungkinkan pengambilan keputusan dengan mengakomodasi ketidakpastian dan ketidakjelasan dalam sistem. Konsep dasar dari logika *fuzzy* adalah memperluas

gagasan "benar" atau "salah" ke dalam rentang nilai yang lebih luas daripada hanya 0 dan 1.



Gambar 2. 4 Contoh Keanggotaan Fuzzy (Saludin Muis, 2018)

Sebagai contoh dalam kasus ini, penulis ingin membuat sistem cerdas untuk mengontrol ketinggian air pada saluran air yang ada di Pasar *Family*. Dalam logika klasik, suatu pernyataan hanya bisa benar atau salah. Misalnya, jika ketinggian air lebih dari 1 meter, Anda mungkin mengatakan "ketinggian air ini bahaya" dan jika tidak, Anda mengatakan "ketinggian air ini tidak bahaya". Namun, dengan logika *fuzzy*, Anda mempertimbangkan bahwa ada tingkat ketinggian air yang berbeda-beda. Misalnya, ketinggian air 1 meter sedikit bahaya, sedang, atau sangat bahaya tergantung pada preferensi Anda. Dalam logika *fuzzy*, Anda bisa menggambarkan ini sebagai derajat kebenaran yang fleksibel, bukan hanya benar atau salah.

Metode *fuzzy* memungkinkan pemodelan sistem yang kompleks dan tidak linier dengan menggabungkan pengetahuan manusia dengan data empiris. Dalam pengendalian ketinggian air, variabel-variabel seperti tingkat curah hujan, debit air

masuk, dan karakteristik saluran air dapat dimodelkan sebagai variabel *fuzzy* yang memiliki nilai yang tidak pasti.

Metode *fuzzy* memungkinkan pengambilan keputusan yang fleksibel dan dapat disesuaikan dengan tingkat ketidakpastian yang ada. Dalam konteks pengendalian ketinggian air, *Fuzzy Inference System (FIS)* dapat digunakan untuk menentukan keputusan tentang kapan dan seberapa banyak pompa atau pintu air harus diaktifkan atau dinonaktifkan untuk menjaga ketinggian air dalam batas yang diinginkan.

Dengan memanfaatkan metode *fuzzy* dalam pengendalian ketinggian air, sistem pengendalian dapat menjadi lebih adaptif, responsif, dan mampu mengatasi ketidakpastian yang melekat dalam lingkungan alamiah. Hal ini dapat membantu dalam menjaga ketinggian air pada level yang aman dan optimal, serta mengurangi risiko banjir atau kekurangan air yang berpotensi merugikan.

#### **2.4 *Internet of Things***

*Internet of Things (IoT)* adalah konsep di mana semua perangkat terhubung ke internet. Ini merupakan teknologi terbaru yang sedang berkembang pesat dan menjadi tren utama di masa depan. IoT telah mengalami pertumbuhan yang signifikan sebagai platform komunikasi antar perangkat. Dalam sektor biomedis atau kesehatan, ekonomi yang berkembang pesat diperkirakan akan meningkatkan jumlah pengguna IoT secara dramatis. Selain itu, aplikasi IoT juga sedang mengalami pertumbuhan yang pesat di berbagai bidang lainnya.

Menurut laporan *Institute of Electrical and Electronics Engineers (IEEE)* tahun 2012, *Internet of Things (IoT)* meliputi tiga elemen utama: sensor, jaringan,

dan konektivitas internet. *International Telecommunication Union* (ITU) mendefinisikan IoT sebagai infrastruktur publik yang digunakan untuk masyarakat informasi, menyediakan layanan interkoneksi fisik dan virtual berdasarkan perkembangan teknologi informasi dan komunikasi. Secara ringkas, IoT terdiri dari sensor, jaringan, dan aplikasi yang mendukung konektivitas perangkat ke internet menggunakan berbagai teknologi komunikasi.

#### **2.4.1 Manfaat *Internet of Things* (IoT)**

Peran *Internet of Things* (IoT) dalam pembuatan alat pendeteksi banjir dan kebakaran sangat penting karena memiliki banyak manfaat sebagai berikut:

1. **Deteksi Dini:** *IoT* memungkinkan deteksi dini terhadap potensi bahaya seperti banjir dan kebakaran dengan menggunakan sensor-sensor yang sensitif. Hal ini memungkinkan tindakan pencegahan atau evakuasi dapat diambil tepat waktu, mengurangi risiko kerugian nyawa dan properti.
2. **Pemantauan Real-Time:** *IoT* memungkinkan pengumpulan data secara real-time dari sensor-sensor yang terpasang di lokasi-lokasi yang rentan terhadap bencana. Ini memungkinkan pemantauan kondisi lingkungan dengan akurat dan cepat, sehingga memungkinkan respons yang lebih cepat dan tepat dalam situasi darurat.
3. **Pemantauan Jarak Jauh:** Melalui koneksi internet, *IoT* memungkinkan pemantauan kondisi lingkungan dari jarak jauh, memungkinkan petugas darurat atau otoritas terkait untuk mengambil tindakan yang diperlukan tanpa harus berada di lokasi secara fisik.
4. **Koordinasi Respons Darurat:** *IoT* memungkinkan pertukaran data secara real-time antara berbagai pihak terkait seperti petugas pemadam kebakaran, polisi,

dan layanan darurat lainnya. Hal ini memungkinkan koordinasi respons darurat yang lebih efisien dan terkoordinasi.

5. Peringatan dan Notifikasi: IoT memungkinkan generasi peringatan dan notifikasi kepada pengguna atau pihak yang berwenang melalui pesan teks, email, atau aplikasi seluler. Ini memungkinkan orang-orang untuk mengambil tindakan pencegahan atau evakuasi tepat waktu (Isyanto dkk., 2020).

## **2.5 UML (Unified Modelling Language)**

*Unified Modeling Language* (UML) diadopsi seiring dengan kemajuan teknik pemrograman berbasis objek, sebagai respons terhadap kebutuhan dalam memodelkan visual, menjabarkan, membangun, dan mendokumentasikan sistem perangkat lunak.









UML adalah sebuah bahasa visual yang digunakan untuk memodelkan sistem dan komunikasi dengan menggunakan diagram serta teks pendukung (Rosa & Shalahuddin, 2016).

### **2.5.1 Use Case Diagram**

Diagram *use case*, yang sering disebut sebagai "diagram kebutuhan", adalah alat pemodelan yang digunakan untuk mengilustrasikan perilaku sistem yang sedang dikembangkan. Diagram ini memvisualisasikan bagaimana aktor-aktor berinteraksi dengan sistem dan menjelaskan skenario-skenario yang mungkin terjadi. Diagram *use case* membantu dalam memahami dengan jelas fungsi-fungsi yang ada dalam sistem serta aktor-aktor yang terlibat dalam penggunaan fungsi-

fungsi tersebut. Tabel 2.2 menunjukkan simbol dalam *Use Case Diagram* (Sugiarti, 2013).







Tabel 2. 2 Simbol *Use Case Diagram* (Sugiarti, 2013)

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Actor</i>	Mengidentifikasi peran-peran yang dimainkan oleh pengguna saat berinteraksi dengan use case.
2		<i>Dependency</i>	Hubungan di mana elemen dependen mengacu pada elemen induk ketika elemen induk mengalami perubahan.
3		<i>Generalization</i>	Relasi di mana objek turunan membagi perilaku dan struktur data dengan objek induk di atasnya.
4		<i>Include</i>	Menentukan contoh sumber.
5		<i>Extend</i>	Menyatakan bahwa use case target mengembangkan perilaku dari use case sumber pada titik tertentu.
6		<i>Association</i>	Koneksi antara objek-objek.
7		<i>Use case</i>	Deskripsi urutan tindakan yang dilakukan oleh sistem untuk mencapai hasil yang dapat diukur untuk seorang aktor.
8		<i>Collaboration</i>	Perilaku yang dihasilkan oleh aturan interaksi dan elemen-elemen lainnya yang lebih besar dari jumlah dan elemen individu.

### 2.5.2 Activity Diagram

Diagram aktivitas menggambarkan urutan aktivitas atau proses yang akan dilakukan oleh sistem. Selain itu, aliran tampilan sistem dapat digambarkan dalam diagram aktivitas. Dalam diagram aktivitas, tiap elemen terhubung oleh panah yang menunjukkan urutan aktivitas dari awal hingga akhir. Tabel 2.3 menunjukkan simbol dalam *Activity Diagram*.

Tabel 2. 3 Simbol *Activity Diagram* (www.dicoding.com)

NO	SIMBOL	NAMA	KETERANGAN
1		Status Awal	Sebuah awalan ditunjukkan pada diagram aktivitas.
2		Aktivitas	Setiap aktivitas yang dilakukan oleh sistem umumnya dimulai dengan kata kerja.
3		Percabangan / <i>Decision</i>	Dalam percabangan, terdapat beberapa pilihan aktivitas yang dapat dipilih.
4		Penggabungan / <i>Join</i>	Penggabungan adalah saat dua atau lebih hal digabungkan menjadi satu.
5		Status Akhir	Diagram aktivitas menampilkan status akhir dari aktivitas yang dilakukan oleh sistem.
6		<i>Swimlane</i>	<i>Swimlane</i> membedakan berbagai organisasi bisnis yang bertanggung jawab atas berbagai tindakan yang terjadi.

### 2.5.3 *Sequential Diagram*

Diagram yang mengilustrasikan interaksi antara objek dan komunikasi di antara mereka melalui petunjuk atau simbol adalah *Sequence diagram*. Diagram ini menggambarkan perilaku dalam suatu skenario dan menunjukkan bagaimana entitas dan sistem saling berinteraksi, termasuk urutan pesan yang digunakan selama interaksi tersebut. Setiap pesan dijabarkan dalam urutan eksekusi yang jelas. *Sequence diagram* terkait erat dengan *Use Case Diagram*, di mana setiap *Use Case* menghasilkan satu *Sequence Diagram*.

### 2.5.4 *Class Diagram*

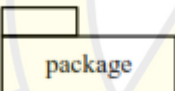
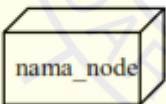


Diagram kelas menggambarkan struktur sistem program berdasarkan jenis-jenis entitas yang terbentuk. *Class Diagram* menjelaskan proses database dalam

suatu program dan menunjukkan alur jalan database di dalam sistem. Diagram kelas, yang sering disebut sebagai "*class diagram*", merupakan bagian integral dari laporan sistem karena menguraikan struktur sistem dari perspektif definisi kelas-kelas yang akan digunakan dalam pembangunan sistem (Satzinger, 2011).

### 2.5.5 *Deployment Diagram*

Diagram deployment menggambarkan hubungan antara perangkat lunak dengan node pemrosesan. Diagram ini menampilkan konfigurasi elemen pemroses dan perangkat lunak yang dijalankan pada saat *runtime*. Sebelum tahap *coding* dimulai, diagram ini dianggap krusial dalam tahap implementasi perangkat lunak. Tabel 2.4 menjelaskan simbol dalam *Deployment Diagram*.

Tabel 2. 4 Simbol *Deployment Diagram* (sis.binus.ac.id)

Simbol	Deskripsi
<i>Package</i> 	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih <i>node</i>
<i>Node</i> 	Biasanya mengacu pada perangkat keras (hardware), perangkat lunak yang tidak dibuat sendiri (software), jika di dalam node disertakan komponen mengkonsistenkan rancangan maka komponen yang telah didefinisikan sebelumnya pada diagram komponen
Kebergantungan/ <i>dependency</i> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
<i>Link</i> 	Relasi antar <i>node</i>

## 2.6 *Software* dan Bahasa Pemrograman Terkait

### 2.6.1 *Software*

Istilah "software" atau "perangkat lunak" merujuk kepada program-program yang digunakan untuk menampilkan dan mengelola data serta instruksi yang terstruktur di dalam komputer. Perangkat lunak memiliki tanggung jawab untuk mengoordinasikan, mengatur, dan menyatukan komponen-komponen perangkat keras sistem komputer dengan maksud untuk mengeksekusi tugas atau instruksi yang diberikan (Dhaifullah dkk., 2022).

#### 2.6.1.1 *Arduino IDE*

*Arduino IDE* adalah lingkungan pengembangan terintegrasi (*Integrated Development Environment*) yang dirancang khusus untuk membuat, memprogram, dan mengunggah kode ke papan pengembangan mikrokontroler Arduino. Berikut adalah penjelasan lebih rinci tentang *Arduino IDE*:

1. *Text Editor*: *Arduino IDE* menyediakan editor teks yang sederhana tetapi fungsional yang memungkinkan pengguna untuk menulis kode program dalam bahasa Arduino. Editor ini dilengkapi dengan fitur penyorotan sintaksis, yang mempermudah pengguna untuk memahami struktur dan sintaks kode program.
2. *Built-in Examples*: *Arduino IDE* dilengkapi dengan banyak contoh program (*sketch*) yang mencakup berbagai fungsi dan komponen Arduino. Contoh-contoh ini dapat digunakan sebagai referensi atau dasar untuk membuat program baru.
3. *Compiler*: *Arduino IDE* memiliki kompiler (pengompilasi) yang mengonversi kode program Arduino yang ditulis dalam bahasa pemrograman C/C++

menjadi kode biner (bahasa mesin) yang dapat dipahami oleh mikrokontroler Arduino.

4. *Serial Monitor: Arduino IDE* menyediakan *Serial Monitor* yang memungkinkan pengguna untuk berkomunikasi dengan papan Arduino secara serial. Ini memungkinkan pengguna untuk membaca data yang dikirimkan oleh Arduino, serta mengirimkan data ke Arduino untuk diolah atau direspon.
5. *Upload Tool: Arduino IDE* menyediakan alat untuk mengunggah (*upload*) kode program yang telah dikompilasi ke papan Arduino. Proses ini biasanya dilakukan melalui koneksi USB antara komputer dan papan Arduino.
6. *Library Manager: Arduino IDE* memiliki *Library Manager* yang memudahkan pengguna untuk mengelola dan menginstal perpustakaan (*library*) tambahan yang diperlukan untuk proyek mereka. Perpustakaan ini bisa mencakup berbagai fungsi dan modul yang dapat digunakan untuk memperluas kemampuan Arduino.
7. *Platform Support: Arduino IDE* mendukung berbagai jenis papan Arduino, termasuk Arduino Uno, Arduino Mega, Arduino Nano, dan lain-lain. Selain itu, *Arduino IDE* juga mendukung berbagai papan mikrokontroler berbasis Arduino dari produsen pihak ketiga.

#### **2.6.1.2 MySQL**

MySQL, yang disebut juga *My Structure Query Language*, merupakan perangkat lunak sistem manajemen basis data *SQL Database* yang berperan sebagai DBMS. Menurut prinsip *open source*, MySQL, yang merupakan sebuah *Database*

*Management System* (DBMS), menyediakan *source code* yang terbuka (Agus Prayitno, 2015).

### 2.6.1.3 *Blynk*

*Blynk* adalah *platform* yang memungkinkan kita untuk dengan mudah membuat aplikasi perangkat *Internet of Things (IoT)* tanpa perlu pengetahuan pemrograman yang mendalam. Dengan *Blynk*, kita dapat mengontrol dan memonitor perangkat keras dari jarak jauh melalui ponsel pintar atau tablet. Berikut adalah beberapa fitur utama dan penjelasan tentang perangkat lunak *Blynk*:

1. *User Interface Builder*: *Blynk* menyediakan *User Interface (UI) Builder* yang intuitif dan mudah digunakan. Kita dapat dengan cepat membuat antarmuka pengguna untuk aplikasi *IoT* kita dengan menambahkan *widget* seperti tombol, *slider*, grafik, tampilan nilai, dan banyak lagi.
2. Komunikasi Dua Arah: *Blynk* memungkinkan komunikasi dua arah antara perangkat keras dan aplikasi *Blynk* yang kita buat. Ini berarti kita tidak hanya dapat mengontrol perangkat dari aplikasi, tetapi juga menerima data dari perangkat dan menampilkannya dalam aplikasi.
3. Koneksi ke Perangkat Keras: *Blynk* mendukung berbagai jenis perangkat keras, termasuk mikrokontroler seperti Arduino, ESP8266, ESP32, *Raspberry Pi*, dan banyak lagi. Kita dapat menggunakan *library Blynk* yang tersedia untuk perangkat keras yang kita pilih agar dapat terhubung dengan *platform Blynk*.
4. Protokol Komunikasi Aman: *Blynk* menggunakan protokol komunikasi yang aman antara aplikasi *Blynk* dan perangkat keras. Ini memastikan bahwa data yang dikirimkan antara perangkat dan aplikasi tetap aman dan terlindungi.

5. Integrasi dengan Layanan *Cloud*: *Blynk* memungkinkan kita untuk mengintegrasikan aplikasi *IoT* kita dengan layanan *cloud* lainnya seperti *Google Sheets*, *Twitter*, dan banyak lagi. Ini memberikan fleksibilitas tambahan dalam mengelola dan menganalisis data dari perangkat *IoT*.
6. *Blynk Cloud* atau *Server Sendiri*: Kita memiliki pilihan untuk menggunakan layanan *cloud Blynk* yang di-host oleh *Blynk*, atau kita juga dapat mengatur *server Blynk* sendiri untuk mengendalikan data dan menjaga privasi.
7. Dokumentasi Lengkap dan Komunitas yang Aktif: *Blynk* menyediakan dokumentasi yang lengkap dan mudah diakses, serta komunitas pengguna yang aktif di forum dan saluran komunikasi lainnya. Ini membuatnya mudah bagi pengguna untuk memulai, belajar, dan memecahkan masalah saat menggunakan platform *Blynk*.

## **2.6.2 Bahasa Pemrograman**

### **2.6.2.1 PHP (*PHP: Hypertext Preprocessor*)**

PHP, yang merupakan singkatan dari *PHP: Hypertext Preprocessor*, mengartikan *hypertext preprocessor* sebagai "pengolah *hypertext*" atau "pengolah HTML". Ini merujuk pada fakta bahwa PHP adalah bahasa pemrograman web yang digunakan untuk menghasilkan kode HTML atau membuat halaman web secara dinamis (Pratama, A., 2019).

### **2.6.2.2 HTML (*HyperText Markup Language*)**

*HTML* adalah bahasa markup yang digunakan untuk membuat struktur dasar sebuah halaman web. Dengan *HTML*, Anda dapat menentukan elemen-elemen dasar seperti teks, gambar, *hyperlink*, tombol, formulir, tabel, dan banyak lagi.

Struktur dasar dari sebuah halaman web terdiri dari berbagai elemen *HTML* yang ditandai dengan tag-tag khusus. Contohnya, tag <h1> hingga <h6> digunakan untuk judul, <p> untuk paragraf, <img> untuk gambar, <a> untuk *hyperlink*, <table> untuk tabel, dan seterusnya.

Setiap elemen *HTML* dapat memiliki atribut yang menunjukkan informasi tambahan. Misalnya, atribut `src` pada tag <img> digunakan untuk menginformasikan di mana gambar akan ditampilkan (Pratama, A., 2019).

### **2.6.2.3 CSS (*Cascading Style Sheets*)**

CSS adalah gaya yang digunakan untuk mengatur tampilan dan tata letak elemen *HTML* di halaman web. Ini memungkinkan anda mengubah warna, ukuran, jenis *font*, *margin*, *padding*, dan banyak lagi.

CSS memungkinkan pemisahan antara struktur dan presentasi halaman web, sehingga memungkinkan perubahan tata letak atau gaya secara konsisten di seluruh situs web hanya dengan mengubah satu file *CSS*.

Untuk menerapkan *CSS*, Anda dapat menggunakan aturan *CSS* yang terdiri dari selektor dan deklarasi. Selektor digunakan untuk memilih elemen *HTML* yang akan diberi gaya, sedangkan deklarasi berisi properti-properti gaya dan nilainya.

### **2.6.2.4 JavaScript**

*JavaScript* adalah bahasa pemrograman tingkat tinggi yang digunakan untuk membuat dan mengontrol perilaku dinamis halaman web. Berikut adalah beberapa poin penting tentang *JavaScript*.:

1. *Client-Side Scripting: JavaScript* dijalankan di sisi klien (*client-side*), artinya kode *JavaScript* dieksekusi di *browser* pengguna daripada di server. Ini

memungkinkan pengguna untuk berinteraksi dengan halaman web tanpa melakukan permintaan ke server.

2. *Multi-Platform: JavaScript* adalah bahasa pemrograman yang sangat fleksibel dan populer dalam pengembangan web karena dapat digunakan di banyak platform dan browser.
3. *Dynamic Content: JavaScript* memungkinkan Anda untuk menambah, menghapus, atau mengubah konten halaman web secara dinamis tanpa memuat ulang halaman. Ini dapat digunakan untuk membuat efek animasi, menanggapi interaksi pengguna, atau memperbarui data secara *real-time*.
4. *Event-Driven Programming: JavaScript* adalah bahasa pemrograman yang berbasis peristiwa (*event-driven*), yang berarti Anda dapat menentukan tindakan yang akan dilakukan ketika suatu peristiwa terjadi. Contohnya adalah menanggapi klik tombol, mengirimkan formulir, atau menggulir halaman.
5. *DOM Manipulation: Salah satu kegunaan utama JavaScript* adalah untuk memanipulasi struktur dan konten halaman web melalui *DOM (Document Object Model)*. Anda dapat menambahkan, menghapus, atau memodifikasi elemen-elemen *HTML*, mengubah gaya *CSS*, atau menanggapi interaksi pengguna dengan menggunakan *JavaScript*.
6. *Validasi Formulir: JavaScript* sering digunakan untuk melakukan validasi data input pada formulir di halaman web. Ini memungkinkan Anda untuk memastikan bahwa pengguna memasukkan data yang benar sebelum mengirimkan formulir ke *server*.

7. *Ajax: JavaScript* memungkinkan penggunaan *Ajax (Asynchronous JavaScript and XML)* untuk melakukan permintaan ke *server* secara *asynchronous*, yang memungkinkan halaman web memuat konten tambahan atau memperbarui data tanpa perlu memuat ulang halaman secara penuh.

*JavaScript* merupakan salah satu teknologi paling penting dalam pengembangan web modern, dan banyak kerangka kerja dan perpustakaan *JavaScript* yang tersedia untuk memperluas kemampuannya dan memudahkan pengembangan aplikasi web yang kompleks dan interaktif.

### **2.6.3 Laravel**

Laravel adalah *Framework PHP 5.3 opensource* yang dibuat oleh Taylor Otwell dan dilisensikan di bawah MIT *License*. Tujuannya adalah untuk memfasilitasi para pengembang dalam menciptakan situs web dengan menggunakan sintaksis yang mudah dipahami, elegan, ekspresif, dan menghibur.

Laravel adalah sebuah platform pengembangan web yang menonjol karena sintaksisnya yang ekspresif. Platform ini memungkinkan pengurangan beban kerja bagi desainer dalam sebagian besar proyek web, termasuk dalam hal *routing*, *session*, dan *caching*. Selain itu, Laravel berusaha untuk mengintegrasikan keunggulan pengembangan dari bahasa-bahasa lain seperti *Ruby on Rails*, ASP.NET, MVC, dan Sinatra.

Untuk mengembangkan sebuah proyek, Laravel sangat mudah dikonfigurasi, dan kebutuhan yang harus dipenuhi termasuk:

1. *Text Editor*

Pilihlah editor teks yang cocok dengan preferensi dan kebutuhan kita. Saya pribadi menggunakan *Sublime Text* sebagai *text editor*, yang telah dipasang plugin *Emmet* untuk mempercepat penulisan kode HTML. Alternatif lain yang bisa dipertimbangkan antara lain *PHPStorm*, *Aptana*, *Netbeans*, dan *Notepad++*.

## 2. *Websserver dan Database*

Penulis memanfaatkan XAMPP dalam versi 3.2.1, yang telah menyediakan dukungan untuk PHP 5.4 serta modul Mcrypt, yang diperlukan untuk instalasi Laravel. Penting untuk dicatat bahwa Laravel memerlukan PHP minimal versi 5.3 ke atas..

## 3. *Composer*

Kita akan menggunakan composer untuk menginstal Laravel. *Composer* adalah "*manager dependency*" PHP yang memungkinkan kita untuk secara otomatis menambah *library* untuk website kita tanpa perlu mendownload library secara terpisah.

### 2.6.3.1 Struktur Projek Laravel

Ketika membuka folder Laravel, file dan folder berikut akan muncul:

#### 1. *app/*

Folder ini digunakan sebagai lokasi default untuk menyimpan kode yang sudah ditulis atau dikonfigurasi. Semua kode proyek aplikasi, termasuk kode logic, konfigurasi, dan sebagainya, diletakkan di dalam folder ini.

#### 2. *bootstrap/*

Di dalam folder ini terdapat file-file prosedur dari Framework Laravel, termasuk beberapa file yang hanya dapat diubah oleh pengguna Laravel yang berpengalaman..

3. *vendor/*

Berisikan semua composer yang digunakan untuk aplikasi, serta file framework Laravel.

4. *public/*

berisikan sumber daya yang dapat digunakan dalam menyimpan file CSS, JavaScript, Foto, dan file-file lainnya (biasanya terkait View).

5. *.gitattributes*

Merupakan file konfigurasi untuk sistem kontrol versi Git, yang saat ini memiliki popularitas yang tinggi.

6. *.gitignore*

File ini memuat informasi mengenai folder-folder yang mungkin diabaikan oleh Git.

7. *Artisan*

Digunakan untuk menjalankan atau mengeksekusi *command* artisan CLI di Laravel.

8. *composer.json* dan *composer.lock*

Berisikan info mengenai paket composer yang digunakan aplikasi.

9. *phpunit.xml*

Berisikan konfigurasi unit tes Laravel *default*, serta pemuatan dependensi *composer*. Selain itu, file ini mengeksekusi semua tes yang ada di direktori aplikasi/tes.

## 10. *server.php*

File ini memuat petunjuk tentang cara menjalankan Laravel menggunakan server web internal yang terpasang dengan PHP versi 5.4.

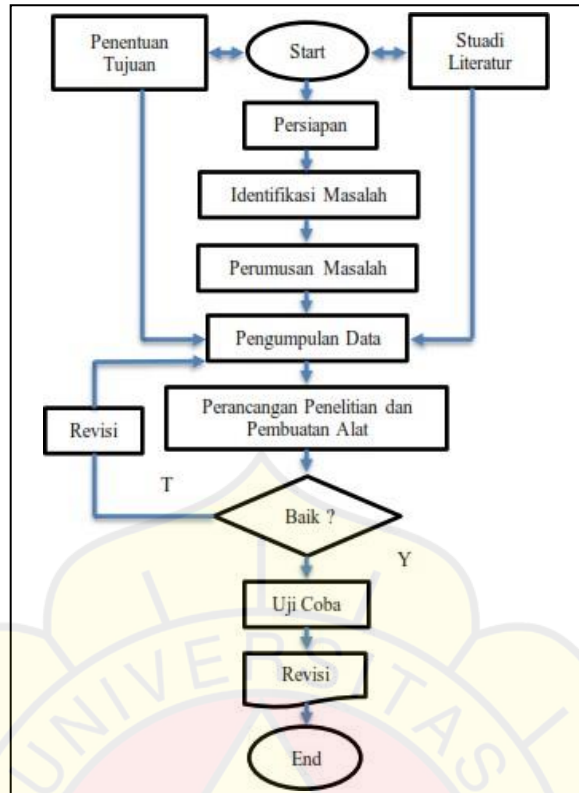
## 2.7 Kajian Penelitian Terdahulu

### 2.7.1 Paper 1

( Rancang Bangun Prototipe Peringatan Dini Banjir Menggunakan Raspberry Pi Berbasis IoT, Restu Adjie Priatim, dkk, *Jambura Journal of Electrical and Electronics Engineering*, 2023, Sinta 3 )

Fokus penelitian ini adalah mengembangkan sistem deteksi awal banjir kemudian memperluasnya dengan berbagai teknik terkait. Tujuan sistem ini adalah untuk mengetahui seberapa besar banjir dan daerah genangannya dalam beberapa hari ke depan. Sistem Peringatan Dini Banjir (FEWS) menggunakan data *real-time* dan prakiraan.

Agar prosedur penelitian lebih mudah dipahami, metode penelitian diuraikan dalam gambar 2.5.



Gambar 2. 5 Diagram Alur Proses Penelitian (Restu Adjie Priatim, 2023)

Pengujian komprehensif prototipe sistem peringatan dini banjir menghasilkan perbedaan signifikan karena adanya perbandingan dan persimpangan, dengan penundaan lima detik untuk setiap perubahan level. Nilai kapasitansi sensor menunjukkan angka 71.9 cm pada kondisi normal, 70.8 cm saat dalam status waspada, 40.5 cm saat dalam status siaga, dan 20.3 cm saat dalam status awas, sebagaimana tercantum dalam tabel 2.5.

Tabel 2. 5 Pengujian Alat Keseluruhan (Restu Adjie Priatim, 2023)

Kenaikan air pada sensor (cm)	Jarak nilai kapasitansi pada ketinggian sensor (cm)	Status LED	Buzzer berbunyi	Status
70	71.9	Hijau	Tidak	Aman
70-40	70.8	Kuning	Tidak	Waspada

40-20	40.5	Biru	Berbunyi	Siaga
20-10	20.3	Merah	Berbunyi	Awat

Berdasarkan hasil diatas, ditarik kesimpulan bahwa sistem peringatan dini banjir menggunakan Raspberry Pi telah berhasil dikembangkan. Berdasarkan pengembangan dan pengujian yang dilakukan, sistem ini mampu mengenali dan melacak ketinggian air untuk mendeteksi potensi banjir. Perangkat ini juga dilengkapi dengan kemampuan untuk mengirim data secara langsung melalui internet, memungkinkan akses data secara real-time di mana saja. Sistem ini didesain untuk membantu masyarakat memitigasi dampak yang ditimbulkan oleh bencana banjir.

### 2.7.2 Paper 2

**( Monitoring Ketinggian Air Sebagai Pendeteksi Banjir Menggunakan Notifikasi Telegram Berbasis *Internet Of Things*, Raditia Vindua & Hasmi Sya'Ban Triaji, Terapan Informatika Nusantara, 2023, Sinta 5 )**

Tujuan penelitian ini adalah mengembangkan sebuah perangkat menggunakan sensor ultrasonik untuk deteksi banjir serta memonitor tingkat air secara online melalui aplikasi Telegram di smartphone, guna memberikan informasi dini tentang potensi banjir dan meningkatkan kesadaran masyarakat terhadap kondisi ketinggian air.

Penelitian ini memanfaatkan observasi, wawancara, dan studi literatur untuk menyusun data yang dianalisis. Tahapan analisis kebutuhan, perancangan ruang kerja (*modeling*), dan implementasi (penerapan) merupakan bagian dari proses

merancang sistem perangkat pemantauan ketinggian air sebagai detektor banjir dengan penerapan metode *Rapid Application Development* (RAD).

Evaluasi dan pengujian sistem alat pemantau ketinggian air sebagai pendeteksi banjir yang memanfaatkan sensor ultrasonik dan notifikasi Telegram menunjukkan bahwa sistem ini mampu melakukan deteksi dini ketinggian air. Informasi pemantauan dan peringatan dikirim langsung ke perangkat *smartphone* melalui Telegram, memungkinkan respons cepat dalam penanganan banjir dan mengurangi potensi kerugian. Uji coba perangkat ini menunjukkan semua komponen beroperasi dengan baik. Sensor ultrasonik mengukur ketinggian air secara akurat, *buzzer* memberikan peringatan dengan intensitas yang sesuai, dan notifikasi Telegram dapat diakses untuk memantau dan memberi peringatan ketinggian air kepada pengguna, bahkan saat mereka tidak berada di rumah.

### **2.7.3 Paper 3**

**( Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis *Internet Of Things*, Frisca Tri Arumsari, dkk, *Infotech Journal*, 2023, Sinta 5 )**

Penelitian ini bertujuan untuk mengembangkan sistem pendeteksi kebakaran yang memberikan peringatan dini dan membantu masyarakat dalam menghadapi risiko kebakaran. Harapannya, perangkat ini akan beroperasi dengan efektif. Menggunakan mikrokontroler Wemos D1 R1 dan sensor detektor api MQ2, sistem *Internet of Things* (IoT) ini dirancang untuk memberikan notifikasi kepada pengguna saat tanda-tanda awal kebakaran muncul dan menyampaikan informasi melalui platform aplikasi Telegram. Hal ini memungkinkan pengguna untuk

mengambil tindakan cepat dalam mengurangi kerugian yang disebabkan oleh kebakaran.

Metode pengembangan sistem SDLC (*System Development Life Cycle*) yang memanfaatkan model air terjun sebagai model prosedur pengembangan perangkat lunak selanjutnya digunakan dalam penelitian ini. Tahapan perencanaan, pemodelan, pelaksanaan, dan pengujian diibaratkan seperti air terjun yang terus turun (Windiastik et al., 2019).

Sistem deteksi kebakaran berbasis *Internet of Things* lulus semua pengujiannya, dan hasilnya menunjukkan bahwa semuanya berfungsi sesuai harapan. Selain itu, peneliti menekankan bahwa sistem ini dapat memberi tahu pengguna atau pemilik properti tentang adanya kebakaran. Ada dua jenis sensor berbeda yang digunakan dalam sistem ini. Sensor *fire finder* mampu mengidentifikasi keberadaan api dalam cakupan jarak 0 hingga 80 cm, sedangkan sensor MQ2 digunakan untuk mengenali gas atau asap dengan konsentrasi lebih dari 400 ppm. Outputnya, sistem ini memiliki pompa air, *buzzer*, dan notifikasi yang dikirimkan melalui aplikasi Telegram.