

BAB II

TINJAU PUSTAKA

2.1. TEORI TERKAIT

Dalam rangka mendukung sistem peringatan dini banjir di area bendungan, dibutuhkan sebuah perangkat yang mampu membuka dan menutup pintu air secara otomatis. Selain itu, dibutuhkan juga media informasi yang dapat menyampaikan data ketinggian air dengan cepat dan akurat agar masyarakat sekitar dapat meningkatkan kewaspadaan serta mempersiapkan diri lebih awal menghadapi potensi banjir. Perancangan alat ini melibatkan beberapa tahapan, seperti pembuatan diagram blok, *flowchart*, desain mekanik, perancangan perangkat lunak, serta perancangan rangkaian elektronik. Hasil pengujian menunjukkan bahwa ketika sumber daya diaktifkan, sensor *ultrasonic* langsung mulai mendeteksi tinggi permukaan air. Jika air mencapai ketinggian tertentu, maka NodeMCU yang telah terhubung dengan jaringan *WiFi* akan mengirimkan notifikasi data ketinggian melalui aplikasi *Blynk*. Saat ketinggian air mencapai ≤ 8 cm, data tinggi air ditampilkan secara *realtime* di *Blynk*. Jika air naik ke kisaran 9–11 cm, sensor akan mendeteksi perubahan tersebut dan *Blynk* akan memperbarui data ketinggian, sementara motor DC akan membuka pintu air 1 secara penuh untuk mengalirkan air. Jika tinggi air mencapai 12 cm atau lebih, maka motor DC akan mengangkat pintu air 1 dan 2, serta sistem akan mengirimkan notifikasi *realtime* melalui *blynk* berdasarkan data yang

diterima sensor. Langkah ini bertujuan memberikan sinyal peringatan kepada warga sekitar bahwa ketinggian air telah mencapai ambang bahaya, sehingga masyarakat dapat segera melakukan evakuasi untuk menghindari risiko banjir. [3]

Bendungan memiliki peranan *vital* dalam kehidupan manusia dan memberikan berbagai manfaat penting, seperti mendukung sistem irigasi serta mengendalikan banjir. Namun, kurangnya pengawasan terhadap kondisi bendungan dapat menimbulkan dampak serius, termasuk ancaman keselamatan bagi warga di sekitar dan potensi kerusakan pada lahan pertanian akibat luapan air. Dalam kondisi saat ini, petugas bendungan masih harus melakukan pemantauan secara manual dengan bolak-balik dari lokasi bendungan ke pos jaga untuk mengatur buka tutup pintu air demi menjaga kestabilan debit air. Untuk mengatasi hal tersebut, dikembangkanlah sistem monitoring otomatis untuk pintu air bendungan yang berbasis *Internet of Things*. Penelitian ini menggunakan pendekatan kualitatif, dengan metode pengembangan sistem berbasis model prototype, serta pengujian dilakukan dengan metode *blackbox*. Hasil dari penelitian ini adalah sebuah sistem *prototype* yang mampu mengatur pintu air secara otomatis. Ketika sensor ultrasonik mendeteksi ketinggian air lebih dari atau sama dengan 5 cm, dan sensor aliran air *water flow* mencatat kecepatan aliran lebih dari 4 liter/menit, maka ESP32 akan memerintahkan motor servo untuk membuka pintu air. Sebaliknya, jika ketinggian air melebihi 5 cm namun kecepatan aliran di bawah 4 liter/menit, maka pintu air akan

tertutup secara otomatis. Seluruh data ketinggian dan aliran air ini ditampilkan secara *real-time* melalui antarmuka *website* yang telah dikembangkan.

Pada musim hujan, tidak sedikit wilayah seperti perumahan, lahan perkebunan, maupun sawah yang terdampak banjir. Oleh karena itu, diperlukan sistem pengelolaan air yang efektif untuk meminimalisir risiko banjir. Salah satu upaya pengendalian debit air yang umum dilakukan adalah pembangunan bendungan, waduk, atau pintu-pintu air. Bendungan dan pintu air merupakan struktur yang dirancang untuk menahan serta mengatur aliran air. Selain sebagai pengendali banjir, bendungan juga memiliki manfaat lain seperti sebagai sumber pembangkit listrik dan penyedia air bagi berbagai sektor yang membutuhkan. Agar manfaat dari bendungan ini dapat terus dirasakan secara optimal, maka pengawasan terhadap kondisi dan fungsi bendungan harus dilakukan secara berkelanjutan. Sebagian besar bendungan dilengkapi dengan pintu air yang berfungsi mengalirkan kelebihan air secara bertahap, disesuaikan dengan volume air yang tersimpan di dalamnya. Akan lebih efisien jika sistem pengendalian pintu air tersebut dapat berjalan secara otomatis, mengingat volume air dalam bendungan bersifat fluktuatif dan sulit diprediksi. Oleh karena itu, dibutuhkan alat otomatis yang mampu mengukur ketinggian air serta mengatur buka-tutup pintu air secara mandiri. Dengan adanya sistem otomatis ini, potensi kesalahan manusia dalam pengoperasian pintu air dapat diminimalisir. Pasalnya, masih sering ditemukan kasus di mana

petugas pintu air lalai dalam menjalankan tugasnya, sehingga menyebabkan ketidakstabilan volume air dan menimbulkan kerusakan lingkungan. Dengan sistem otomatis ini, pengaturan volume air menjadi lebih efisien, akurat, dan stabil.[4]

Penelitian ini menunjukkan bahwa ketelitian dalam merancang perangkat keras sangat penting, terutama dalam menghitung besaran tegangan dan arus yang dihasilkan oleh sensor. Tegangan yang tidak sesuai dapat memengaruhi kinerja mikrokontroler secara signifikan. Hasil pengujian pada alat yang dirancang masih bersifat kualitatif. Oleh karena itu, agar sistem dapat berfungsi secara maksimal, diperlukan penggunaan sensor yang mampu menghasilkan sinyal dalam bentuk kuantitatif sehingga perubahan ketinggian air dapat diamati dengan lebih akurat. Dalam implementasinya, penelitian ini menggunakan mikrokontroler ATMega 8535. Bahasa pemrograman yang digunakan adalah *CVAVR*, sementara proses pemrograman dan pengunduhan program dilakukan menggunakan perangkat lunak *CodeVisionAVR* versi 2.03.9. Program interface yang dikembangkan meliputi fungsi untuk mendeteksi sinyal dari sensor, mengolah data yang diterima, serta menampilkan hasil informasi melalui tampilan LCD dan indikator LED.

Penelitian yang dilakukan oleh Febri Agustian, dkk pada tahun 2024 dengan judul "SISTEM MONITORING LUAPAN AIR SUNGAI BERBASIS IoT (INTERNET OF THINGS) STUDI KASUS SUNGAI PEUDADA KABUPATEN BIREUEN" menjelaskan bahwa sungai

berperan sebagai jalur vital bagi masyarakat di sepanjang tepinya karena menyediakan sumber daya penting seperti air untuk memancing, irigasi, dan aktivitas sehari-hari, namun saat musim hujan permukaan air dapat naik signifikan hingga meluap dan menimbulkan banjir yang berdampak pada kerusakan rumah, ketidakstabilan ekonomi, serta terganggunya pendidikan anak-anak, sehingga penggunaan alat pendeksi banjir dinilai penting untuk membantu masyarakat memprediksi waktu terjadinya banjir. Mengembangkan sistem pemantauan ketinggian air menggunakan sensor dengan penyimpanan data pada database yang dapat diakses dalam jangka panjang, serta dilengkapi notifikasi bahaya berupa indikator siaga 1 (1 cm), siaga 2 (2 cm), dan siaga 3 (3 cm) yang ditampilkan melalui speaker, lampu indikator, LCD, dan pesan Telegram sebagai peringatan dini banjir [5].

2.2. LANDASAN TEORI

2.2.1 Sistem Monitoring

Sistem *monitoring* merupakan suatu mekanisme yang berfungsi untuk melakukan pemantauan secara berkelanjutan terhadap kondisi suatu objek tertentu. Tujuan dari sistem ini adalah untuk memperoleh informasi secara *real-time* atau tepat waktu, sehingga kondisi objek yang dipantau dapat diketahui dengan akurat saat itu juga.[6]

2.2.2 PHP

PHP merupakan bahasa pemrograman berbasis skrip yang terintegrasi dengan HTML dan berjalan di sisi *server*. Artinya, seluruh kode atau sintaks yang dituliskan akan diproses sepenuhnya oleh server, sementara yang diteruskan ke browser pengguna hanyalah output atau hasil akhirnya saja.[7]

2.2.3 MySQL

Structured Query Language atau yang lebih dikenal dengan MySQL adalah bahasa yang dirancang khusus untuk mengelola dan memproses data dalam sebuah basis data. MySQL sendiri merupakan sistem manajemen *database* yang bersifat open source. Selain itu, MySQL termasuk dalam kategori Relational Database Management System (RDBMS), yang berarti pengelolaan data dilakukan dengan menempatkannya ke dalam tabel-tabel terpisah. Pendekatan ini memungkinkan proses manipulasi data menjadi lebih efisien dan cepat.[8]

2.2.4 Laravel

Laravel merupakan kerangka kerja pengembangan webberbasis PHP yang menggunakan pola arsitektur Model-View-Controller(MVC). Laravel dirancang untuk meningkatkan kualitas perangkat lunak dengan cara mengurangi biaya pengembangan dan pemeliharaan, serta memberikan sintaks yang ekspresif dan efisien dalam membangun aplikasi . Dengan fleksibilitas tersebut,

Laravel menjadi pilihan populer dalam pengembangan sistem informasi, termasuk sistem monitoring [9].

2.2.5 Visual Studio Code

Visual Studio Code adalah editor teks yang ringan namun memiliki kinerja yang andal. Dikembangkan oleh Microsoft, editor ini dirancang untuk dapat digunakan di berbagai sistem operasi (multiplatform). *Visual Studio Code* secara bawaan mendukung berbagai bahasa pemrograman seperti *JavaScript*, *Node.js*, dan *TypeScript*. Selain itu, tersedia beragam ekstensi yang memungkinkan dukungan tambahan untuk bahasa lain seperti C++, Python, C#, serta PHP. [10]

2.2.6 Database

Database adalah suatu sistem yang dirancang untuk mengatur, menyimpan, dan mengakses data dengan cara yang efisien. Sistem ini berisi kumpulan data yang tersusun secara terorganisir dan ditujukan untuk satu atau lebih kebutuhan dalam format digital, sehingga memudahkan pengguna dalam mengelola serta mengambil informasi sesuai kebutuhan.[11]

2.2.7 UML (Unified Modeling Language)

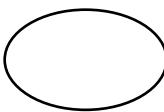
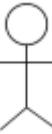
Unified Modeling Language, atau yang biasa disingkat UML, merupakan salah satu metode dalam Rekayasa Perangkat Lunak (RPL) yang digunakan untuk memvisualisasikan cara kerja sebuah sistem, termasuk fungsi, alur, tujuan, serta mekanisme kontrolnya.

UML memiliki empat jenis model yang paling umum digunakan dalam perancangan sistem, yaitu *use case diagram*, *class diagram*, dan *sequence diagram*. Keempat model ini dikenal sebagai teknik dasar dalam pemodelan UML, dan sangat dominan penggunaannya dalam proyek yang berorientasi objek.[12] Dalam perancangan UML yang sering digunakan sebagai berikut:

1. Use Case Diagram

Dalam pembuatan sistem informasi, *use case diagram* ini bersifat statis. Diagram ini berfungsi sebagai pemodelan perilaku (behavior) dalam sistem informasi yang akan dikembangkan, membantu dalam memahami fungsi utama sistem dari perspektif pengguna [13]. Untuk simbol *UseCase Diagram* disajikan pada Tabel 2.1.

Tabel 2. 1 Use Case Diagram

No	Simbol	Keterangan
1.		<i>Use case</i> Merupakan representasi dari fungsi yang diberikan oleh sistem yang menghasilkan output terukur untuk seorang aktor.
2.		<i>Actor</i> Adalah entitas yang menggambarkan peran tertentu yang berinteraksi langsung dengan use case dalam sistem.

3.		<i>Association</i> merupakan Garis penghubung yang menunjukkan relasi atau keterkaitan antara satu objek dengan objek lainnya dalam diagram.
4.		<i>Include</i> Menunjukkan bahwa suatu use case menyisipkan atau menyertakan perilaku dari use case lain sebagai bagian dari prosesnya
5.		<i>Dependency</i> Panah yang menggambarkan adanya keterkaitan pasif atau ketergantungan antara aktor dengan elemen dalam sistem.
6.		<i>Extend</i> Digunakan untuk menunjukkan bahwa suatu use case memperluas perilaku atau fungsi dari use case utama secara opsional.

2. Activity Diagram

Activity Diagram atau *Diagram Aktivitas* adalah jenis diagram statis yang digunakan untuk memvisualisasikan rangkaian aktivitas dalam sebuah sistem bisnis. Diagram ini memberikan gambaran alur proses kerja dari suatu sistem secara terstruktur. Simbol-simbol yang digunakan dalam diagram aktivitas dapat dilihat pada Tabel 2.2.

Tabel 2. 2 *Activity Diagram*

No	Simbol	Keterangan
1.		<i>End point</i> atau <i>final Node</i> Menunjukkan titik akhir dari sebuah aktivitas dalam diagram.
2.		<i>Start Point</i> merupakan simbol yang menunjukkan permulaan dari aktivitas, biasanya ditempatkan di bagian kiri atas diagram.
3.		<i>Fork</i> atau <i>join</i> Digunakan untuk memecah aktivitas menjadi beberapa proses paralel, atau menggabungkan beberapa proses paralel menjadi satu jalur.
4.		<i>Activity</i> Melambangkan proses atau langkah kerja yang terjadi dalam suatu sistem.
5.		<i>Decision</i> Menandakan titik pengambilan keputusan dengan dua kemungkinan hasil, yaitu benar (true) atau salah (false).

3. Sequence Diagram

Sequence Diagram, atau disebut juga *Diagram Urutan*, merupakan jenis diagram interaksi yang menggambarkan proses pengiriman dan penerimaan pesan antar objek dalam sistem. Diagram ini menunjukkan urutan komunikasi antar objek secara

kronologis. Simbol-simbol yang digunakan dalam sequence diagram ditampilkan pada Tabel 3.3.

Tabel 2. 3 *Sequence Diagram*

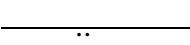
No	Simbol	Keterangan
1.		<i>Entity Class</i> Merupakan bagian dari sistem yang berperan sebagai dasar deskripsi sistem dan berisi kumpulan kelas entitas yang digunakan untuk membangun struktur database.
2.		<i>ControlClass</i> Berfungsi sebagai penghubung antara kelas <i>boundary</i> dengan tabel dalam sistem.
3.		<i>Boundary Class</i> Menggambarkan antarmuka atau batas interaksi sistem, biasanya mewakili tampilan atau tabel.
4.		Pesan atau <i>message</i> Menandakan proses pengiriman informasi atau instruksi antar objek dalam diagram.
5.		<i>Self message</i> Menunjukkan bahwa suatu objek mengirimkan pesan atau instruksi kepada dirinya sendiri.

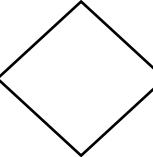
No	Simbol	Keterangan
6.		<i>Activation</i> Mengilustrasikan bahwa sebuah objek sedang menjalankan atau mengeksekusi sebuah operasi.
7.		<i>Lifeline</i> Adalah garis vertikal yang menunjukkan keberadaan objek selama interaksi berlangsung, di mana sepanjang garis tersebut objek dapat melakukan aktivitas.

4. Class Diagram

Class Diagram adalah representasi visual dari hubungan antar kelas serta detail setiap kelas dalam model desain suatu sistem[14]. Diagram ini menampilkan kumpulan kelas, antarmuka, serta hubungan di antara keduanya dalam suatu sistem. Simbol-simbol yang digunakan dalam *Class Diagram* dapat dilihat pada Tabel 2.4.

Tabel 2. 4 Class Diagram

No	Simbol	Keterangan
1.		<i>Generalization</i> , Menunjukkan bahwa objek turunan mewarisi perilaku dan struktur data dari objek induknya.

No	Simbol	Keterangan
2.		<i>Class</i> , adalah Sekelompok objek yang memiliki atribut dan metode yang sama serta saling berbagi struktur dan perilaku.
3.		<i>Nary Assocation</i> , Digunakan untuk menunjukkan hubungan antara lebih dari dua objek dalam satu asosiasi.
4.		<i>Collaboration</i> Menggambarkan urutan aksi atau interaksi dalam sistem yang memberikan hasil terukur bagi aktor.
5.		Menunjukkan tindakan atau fungsi yang dapat dijalankan oleh sebuah objek.
6.		Adalah panah yang menunjukkan bahwa aktor terlibat dalam interaksi secara tidak langsung atau pasif.
7.		<i>Association</i> Garis yang menghubungkan dua objek untuk menggambarkan hubungan atau komunikasi antar objek tersebut.