

BAB II

KAJIAN PUSTAKA

A. KAJIAN TEORITIS

1. Rancang Bangun

Rancang bangun adalah suatu proses yang dilakukan untuk mengimplmentasikan hasil analisis ke dalam bentuk rancangan perangkat lunak, yang kemudian digunakan untuk membangun sistem baru atau meningkatkan sistem yang telah ada sebelumnya (Sitanggang et al., 2022). Menurut Gunawan (2021) rancang bangun dapat diartikan sebagai suatu kegiatan yang mencakup tahap menggambarkan, merencanakan, dan menyusun sketsa untuk mengatur berbagai komponen yang sebelumnya terpisah agar dapat tersusun menjadi satu kesatuan sistem yang terpadu dan dapat berfungsi secara optimal. Ariansyah & Wijaya (2021) berpendapat bahwa rancang bangun bermula dari proses perancangan dalam bentuk visualisasi gambaran kasar suatu sistem yang sebelumnya belum pernah dibuat, kemudian dikembangkan hingga menjadi suatu rancangan yang memiliki fungsi sesuai dengan tujuan yang diharapkan.

Berdasarkan uraian pendapat para ahli tersebut, dapat disimpulkan bahwa rancang bangun merupakan proses awal dalam perancangan suatu sistem yang meliputi kegiatan perencanaan, penyusunan konsep, serta pembuatan sketsa atau rancangan dari berbagai komponen yang sebelumnya

terpisah, sehingga dapat membentuk suatu sistem yang utuh dan memiliki fungsi yang optimal.

2. *Soil moisture*

Kelembapan tanah (*Soil moisture*) adalah salah satu komponen lingkungan yang berperan penting dalam menunjang pertumbuhan pada tanaman, mendukung pemanfaatan air secara optimal, serta mempengaruhi keberhasilan sistem irigasi pada bidang pertanian (Ella et al., 2026). Menurut Isnanto & Suprayogi (2023) kelembapan tanah juga dapat diartikan sebagai kandungan air yang terdapat dalam ruang pori-pori tanah, terletak pada lapisan selain permukaan seperti yang terdapat di sungai atau danau,. Besarnya tingkat kelembapan tanah ini dipengaruhi oleh berbagai faktor eksternal, seperti keadaan iklim atau cuaca.

Berdasarkan kedua pendapat ahli tersebut, dapat disimpulkan bahwa kelembapan tanah merupakan kandungan air yang tersimpan didalam pori-pori tanah yang keberadaannya dapat berubah-ubah karena dipengaruhi oleh berbagai faktor lingkungan, seperti kondisi cuaca.

3. *Irrigation Controller*

Irigasi atau pengairan adalah suatu upaya untuk menyalurkan air ke lahan pertanian melalui pembangunan saluran atau infrastruktur tertentu yang diatur secara sistematis, serta mengalirkan kembali air yang sudah tidak diperlukan setelah dimanfaatkan secara optimal (Wahyudin et al., 2024). Penerapan otomatisasi dalam sistem irigasi diperlukan untuk meningkatkan efisiensi pemanfaatan air sekaligus mengurangi

ketergantungan pada pemantauan manual oleh petani. Kemajuan teknologi *Internet of Things* (IoT) membuka peluang penerapan sistem irigasi otomatis berbasis sensor yang mampu memonitor kondisi tanah secara *real-time* serta mengatur penyiraman sesuai dengan kebutuhan tanaman.

Dalam dunia pertanian mikrokontroller seperti ESP32 dimanfaatkan untuk mengendalikan penyiraman secara jarak jauh dengan memanfaatkan sensor kelembapan tanah (Lestari et al., 2025).

4. *Internet of Things* (IoT)

Internet of Things atau yang sering disingkat dengan *IoT* merupakan teknologi modern yang dikembangkan untuk memperluas pemanfaatan konektivitas internet yang selalu terhubung. Teknologi ini menghubungkan berbagai perangkat atau objek di lingkungan sekitar sehingga dapat berkomunikasi satu sama lain dan membantu mempermudah berbagai aktivitas manusia secara lebih efisien. Saat ini, penggunaan *Internet of Things* (IoT) semakin berkembang dan banyak diterapkan dalam berbagai aspek kehidupan (Selay et al., 2022).

Istilah *Internet of Things* terdiri dari dua kata utama, yaitu *internet* yang berfungsi sebagai media koneksi jaringan dan *things* yang mengacu pada objek atau perangkat yang terhubung dalam sistem tersebut. Dengan konsep ini, perangkat mampu mengumpulkan data, mengirimkannya melalui jaringan internet, serta memungkinkan perangkat lain untuk mengakses informasi tersebut tanpa memerlukan interaksi langsung antara manusia ataupun dari manusia ke komputer (Selay et al., 2022).

Menurut Ariwibowo (2025) *Internet of Things (IoT)* merupakan teknologi yang memungkinkan berbagai perangkat elektronik atau sistem rancangan awal terhubung melalui jaringan internet untuk melakukan pengelolaan serta pemantauan data secara jarak jauh sehingga dapat meningkatkan efisiensi waktu dalam berbagai aktivitas yang memerlukan pengawasan dari lokasi yang berbeda. *Internet of Things* juga diartikan sebagai suatu konsep teknologi yang memungkinkan berbagai perangkat fisik, seperti sensor, actuator, perangkat elektronik, kendaraan, maupun peralatan rumah tangga, terhubung dengan internet sehingga mampu menerima dan mengirim data tanpa memerlukan interaksi manusia secara langsung. Perangkat tersebut dilengkapi dengan sensor, perangkat lunak, dan jaringan komunikasi sehingga dapat :

- a. Mengumpulkan informasi dari lingkungan sekitar.
- b. Mengirimkan informasi tersebut melalui jaringan internet ke sistem lain.
- c. Menerima perintah dari sistem secara otomatis untuk menjalankan fungsi tertentu (Saptadi et al., 2025).

Berdasarkan pendapat tersebut, dapat disimpulkan bahwa *Internet of Things (IoT)* adalah suatu teknologi yang memungkinkan berbagai perangkat fisik yang terhubung melalui jaringan internet untuk mengumpulkan, mengirimkan, dan menerima data secara otomatis sehingga dapat mendukung proses pemantauan serta pengelolaan sistem secara lebih efektif dan efisien.

5. Durian

Pohon durian (*Durio zibethinus*) merupakan salah satu tanaman tropis penghasil buah durian. Tanaman ini termasuk dalam *famili Malvaceae* yang berasal dari wilayah Asia Tenggara, tetapi saat ini telah tersebar dan dibudidayakan di berbagai daerah tropis lainnya. Secara morfologi, pohon durian dapat tumbuh hingga mencapai tinggi sekitar 25-50 meter, memiliki daun berbentuk memanjang, bunga berwarna putih atau kuning pucat serta buah berbentuk bulat dengan kulit keras yang dilapisi duri tajam (Sumarsono et al., 2024).

Sebagian besar tanaman durian belum dibudidayakan secara intensif dalam skala besar, melainkan masih banyak ditanam di pekarangan maupun kebun dengan luas terbatas. Selain itu, penggunaan bibit yang umumnya berasal dari biji menyebabkan karakteristik tanaman menjadi beragam, sehingga produktivitas buah durian cenderung rendah dan bervariasi (Wibowo & Zahrah, 2022).

6. Sensor *Soil moisture*

Sensor kelembapan tanah (*Soil moisture Sensor*) merupakan salah satu teknologi irigasi modern yang mampu mendeteksi kandungan air dalam tanah secara *real-time*. Sensor ini bekerja dengan mengukur tingkat resistansi atau kapasitansi pada media tanah yang kemudian diolah menjadi nilai kelembapan, sehingga dapat dijadikan dasar dalam menentukan kebutuhan penyiraman tanaman (M. J. Saputra & Suryono, 2024).

Sensor *soil moisture* berfungsi untuk melakukan *inisialisasi* dan mendeteksi kondisi kelembapan tanah. Hasil pembacaan tersebut kemudian diterima oleh mikrokontroler untuk diproses dan selanjutnya dikirimkan ke database guna penyimpanan data (Mahfud et al., 2023). Untuk mendukung proses pembacaan data tersebut, sensor *soil moisture* dilengkapi dengan beberapa pin yang memiliki fungsi masing-masing. Adapun terdapat empat pin utama, diantaranya yaitu A0, D0, VCC, dan GND. Luaran dari pin A0 berupa signal analog, sedangkan pin D0 luarannya berupa signal digital, kemudian pin VCC berfungsi sebagai sumber tegangan positif, dan pin GND digunakan sebagai jalur ground atau massa (Matdoan, 2022).

Berdasarkan beberapa penjelasan tersebut, dapat disimpulkan bahwa sensor *soil moisture* merupakan perangkat yang digunakan untuk mendeteksi tingkat kelembapan tanah secara *real-time* dengan memanfaatkan pengukuran resistansi atau kapasitansi pada tanah. Sensor ini bekerja melalui proses inisialisasi dan pembacaan kondisi tanah, kemudian data yang dihasilkan diproses oleh mikrokontroler dan dapat dikirimkan ke sistem penyimpanan seperti database. Dalam penggunaannya, sensor ini dilengkapi dengan beberapa pin utama seperti A0, D0, VCC, dan GND yang berfungsi untuk menghasilkan sinyal analog maupun digital serta mendukung proses koneksi dalam rangkaian sistem pemantauan kelembapan tanah.

7. ESP32

Menurut Alrasyid et al (2025) ESP32 adalah salah satu platform yang efisien dan ekonomis untuk pengembangan aplikasi *Internet of Things* (IoT). Mikrokontroler ini dikembangkan oleh *Espressif Systems Company* dan memiliki berbagai fitur yang mendukung implementasi sistem berbasis *Internet of Things* (IoT). Beberapa keunggulan yang dimiliki ESP32 antara lain prosesor *dual-core*, konektivitas Wi-Fi dan Bluetooth yang telah terintegrasi, jumlah pin *General Purpose Input Output* (GPIO) yang cukup banyak, serta konsumsi daya yang *relative* rendah.

8. Relay

Relay adalah salah satu komponen elektronika yang berfungsi sebagai saklar yang dikendalikan menggunakan arus listrik. Komponen ini terdiri dari dua bagian utama, yaitu elektromagnet (*coil*) yang berfungsi menghasilkan medan magnet dan bagian mekanik berupa kontak saklar (*switch*) yang bekerja untuk menghubungkan atau memutus rangkaian (Dhaifulloh et al., 2024). *Relay* juga didefinisikan sebagai perangkat yang bekerja dengan memanfaatkan gaya elektromagnet dan mekanikal untuk menggerakkan kontak saklar dalam suatu rangkaian. Secara umum, struktur dasar *relay* terdiri dari lilitan kawat konduktor yang melingkari inti besi sebagai pembentuk medan magnet (Stifvani et al., 2025).

Dalam sistem elektronika, *relay* dapat dimanfaatkan sebagai perangkat pengaman yang bekerja secara otomatis ketika terjadi kenaikan tegangan listrik atau arus berlebih sehingga mampu melindungi komponen

lain pada rangkaian dari potensi kerusakan. Selain itu, *relay* juga berfungsi sebagai pengendali untuk perangkat yang beroperasi pada tegangan tinggi yang tidak memungkinkan dioperasikan secara manual sehingga mempermudah proses pengontrolannya (Maghrobi et al., 2021).

Berdasarkan ketiga pendapat ahli tersebut tersebut, *relay* dapat dipahami sebagai komponen elektronika yang berfungsi sebagai saklar listrik yang bekerja dengan memanfaatkan elektromagnet yang dihasilkan oleh kumparan untuk menggerakkan kontak sakelar. Struktur dasar *relay* terdiri dari kumparan (*coil*), inti besi, serta kontak mekanis yang memungkinkan terjadinya perubahan posisi saklar pada rangkaian listrik. Selain digunakan untuk mengendalikan aliran listrik pada suatu sistem, *relay* juga berperan sebagai perangkat pengaman terhadap lonjakan tegangan atau arus berlebih serta sebagai pengendali perangkat bertegangan tinggi yang sulit dioperasikan secara manual.

9. Laravel

Laravel adalah kerangka kerja pada PHP yang berorientasi secara *open source* dan dirilis di bawah lisensi MIT. *Framework* ini dibangun dengan konsep arsitektur MVC (*Model View Controller*) yang bertujuan untuk mempermudah pengembangan aplikasi berbasis web. Laravel ditulis menggunakan bahasa pemrograman PHP dan dirancang untuk meningkatkan kualitas perangkat lunak sekaligus mengurangi kompleksitas dalam proses pemeliharaan sistem. *Framework* ini dikembangkan oleh Taylor Otwell dan dikenal memiliki sintaks yang rapi serta ekspresif

sehingga dapat membantu mempercepat proses pengembangan aplikasi web (Saefudin et al., 2023).

Adapun menurut Aipina & Witriyono (2022) Laravel adalah salah satu *framework* PHP yang populer di kalangan pengembang dalam perancangan pengembangan web. Dengan menggunakan Laravel, proses pembangunan aplikasi dapat dilakukan secara lebih cepat serta memberikan dukungan fitur yang lebih lengkap. Laravel dilengkapi dengan berbagai fitur modern yang dirancang untuk mempermudah dan mempercepat proses pengembangan aplikasi. Beberapa fitur utama yang tersedia pada *framework* ini antara lain *Bundles*, *Eloquent ORM (Object-Relational Mapping)*, *Query Builder*, *Resource Controller*, *Blade*, *Migration*, dan *Middleware*, serta *Automatic Pagination* (Melyani & Aji, 2023).

Berdasarkan beberapa pendapat ahli tersebut, dapat disimpulkan bahwa Laravel adalah *framework* PHP berbasis *open source* yang banyak digunakan dalam pengembangan aplikasi web karena menyediakan berbagai fitur modern serta menggunakan arsitektur MVC (*Model View Controller*) yang membantu pengembang dalam membangun sistem secara lebih terstruktur, efisien, dan mudah dalam proses pemeliharaan. Dengan dukungan berbagai fitur yang tersedia, laravel mampu mempercepat proses pengembangan aplikasi sekaligus meningkatkan kualitas perangkat lunak yang dihasilkan.

10. Solenoid Valve

Solenoid *valve* adalah komponen pengendali yang berfungsi untuk membuka, menutup, atau mengatur aliran air secara elektromechanical. Perangkat ini bekerja menggunakan kumparan (*coil*) sebagai penggerak. Ketika kumparan tersebut diberikan suplai tegangan listrik, baik AC maupun DC sesuai dengan spesifikasinya, maka akan terbentuk medan magnet yang kemudian menggerakkan piston (*plunger*) di dalam katup (Maghrobi et al., 2021).

Adapun Maghrobi et.al (2021) juga menjelaskan bahwa solenoid *valve* memiliki dua kondisi kerja yang dipengaruhi oleh aliran arus listrik pada kumparan (*coil*), yaitu kondisi aktif (*on*) dan tidak aktif (*off*). Mekanisme kerja masing-masing kondisi tersebut adalah sebagai berikut :

- a. Ketika kumparan magnet dialiri arus listrik (*on*), *solenoid valve* akan tertarik menuju *coil* akibat medan magnet yang terbentuk sehingga lubang inlet terbuka. Pada kondisi ini, air dengan tekanan lebih tinggi akan masuk melalui *inlet hole* dan menekan batang *actuator* sehingga katup bergerak (*actuator on*).
- b. Ketika kumparan magnet tidak dialiri arus listrik (*off*), *solenoid valve* akan terdorong menjauh dari *coil* karena adanya pegas pengembali sehingga lubang *outlet* terbuka. Tekanan air kemudian akan masuk melalui *outlet hole* dan menekan batang *actuator* sehingga katup bergerak kembali ke posisi semula (*actuator off*). Prinsip kerja *coil* pada *solenoid valve* ini memiliki kesamaan dengan mekanisme kerja *relay*.

Secara umum, solenoid *valve* hanya memiliki dua kondisi operasi, yaitu *energized* (aktif) dan *de-energized* (tidak aktif). Agar dapat bekerja dengan baik, *solenoid valve* memerlukan tekanan tertentu untuk menggerakkan actuator katup sesuai dengan spesifikasi [erangkat yang digunakan.

Solenoid *valve* juga didefinisikan sebagai salah satu komponen kontrol yang umum digunakan pada sistem pengaturan aliran cairan. Perangkat ini berfungsi untuk mengatur proses penghentian, pembukaan kembali, pengaturan jumlah aliran, pendistribusian, maupun pencampuran cairan dalam suatu sistem. Solenoid *valve* memiliki berbagai jenis yang disesuaikan dengan tipe serta kebutuhan penggunaannya. Berdasarkan model konstruksinya, solenoid *valve* dibedakan menjadi dua jenis utama, yaitu *solenoid valve single coil* dan *solenoid valve double coil*, yang pada dasarnya keduanya memiliki prinsip kerja yang sama (Hifni et al., 2023).

Solenoid *valve* banyak diterapkan pada berbagai sistem aplikasi karena memiliki sejumlah keunggulan. Perangkat ini mampu melakukan proses *switching* secara cepat dan aman, memiliki tingkat keandalan yang tinggi, serta masa layanan yang cukup tinggi. Selain itu, material yang digunakan memiliki kesesuaian yang baik dengan media yang dialirkan, membutuhkan daya pengendalian yang *relative* kecil, dan dirancang dengan bentuk yang kompak (Hifni et al., 2023).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa solenoid *valve* merupakan komponen pengendali aliran cairan yang bekerja secara elektromekanis dengan memanfaatkan medan magnet dari kumparan untuk membuka atau menutup katup. Perangkat ini banyak digunakan dalam berbagai sistem karena mampu mengatur aliran air secara otomatis, memiliki beberapa jenis seperti *single coil* dan *double coil* dengan prinsip kerja yang serupa, serta menawarkan berbagai keunggulan seperti proses kerja yang cepat, tingkat keandalan tinggi, konsumsi daya yang rendah, dan desain yang kompak.

11. React JS

React JS merupakan komponen dari salah satu library JavaScript bersifat gratis yang banyak dimanfaatkan dalam pengembangan antarmuka pengguna (User Interface/UI), pada pembuatan aplikasi dengan konsep single page. Teknologi ini pertama kali dikembangkan oleh Jordan Walke sekitar tahun 2013. Dalam praktiknya, React JS sering digunakan sebagai kerangka kerja untuk membangun bagian front-end pada suatu aplikasi. Selain itu, React berperan dalam mengelola lapisan tampilan baik pada aplikasi berbasis desktop maupun perangkat mobile. Melalui React, pengembang dapat membuat komponen yang dapat digunakan kembali (reusable component) sehingga tidak perlu membuatnya kembali dari awal setiap kali dibutuhkan (Sulistyorini et al., 2022).

React JS memungkinkan pengembang merancang antarmuka aplikasi secara lebih sederhana pada setiap bagian atau komponen sistem.

Kemampuan tersebut menjadikan React banyak digunakan dalam proses pengembangan aplikasi berbasis web. Tingginya tingkat penggunaan teknologi ini dapat dilihat dari berbagai perusahaan besar yang memanfaatkannya dalam pengembangan layanan digital, seperti Facebook, WhatsApp, Netflix, Instagram, Airbnb, American Express, Dropbox, dan Ebay. Selain perusahaan-perusahaan tersebut, banyak penyedia jasa pengembangan aplikasi web juga memanfaatkan React JS karena kemampuannya dalam mendukung pembuatan tampilan aplikasi yang lebih efisien dan terstruktur (Sulistyorini et al., 2022).

Berdasarkan penjelasan yang telah dipaparkan diatas, dapat dipahami bahwa react js merupakan teknologi berbasis JavaScript yang bersifat open source dan banyak digunakan dalam pengembangan aplikasi berbasis web, khususnya pada bagian *front-end*. Teknologi ini memungkinkan pengembang membangun tampilan aplikasi secara terstruktur melalui pendekatan komponen yang dapat digunakan kembali, sehingga proses pengembangan dan pemeliharaan kode menjadi lebih efisien. Selain itu, React JS mendukung pembuatan tampilan yang sederhana, fleksibel, dan mudah dikembangkan untuk berbagai perangkat, baik desktop maupun mobile. Kemampuannya dalam mengelola komponen secara modular menjadikan React JS banyak dimanfaatkan oleh berbagai perusahaan teknologi dan penyedia layanan pengembangan aplikasi web.

12. *Unified Modelling Language (UML)*

Unified Modeling Language atau UML adalah alat yang digunakan dalam rekayasa perangkat lunak untuk menggambarkan alur kerja, fungsi, tujuan, serta mekanisme pengendalian suatu sistem. Dalam proses analisis dan perancangan sistem informasi, pendekatan yang banyak digunakan saat ini adalah metode yang memadukan konsep pemrograman berorientasi objek dengan teknik pengembangan perangkat lunak. Melalui pendekatan tersebut, sistem dipandang sebagai kumpulan objek yang di dalamnya terdapat data serta proses yang saling berkaitan, atau dapat berjalan secara mandiri dalam suatu kelompok sistem yang disebut package (Putri et al., 2025).

UML merupakan bahasa standar yang digunakan dalam pengembangan perangkat lunak untuk menggambarkan serta membentuk berbagai model sistem. Bahasa ini menjelaskan bagaimana suatu model disusun, namun tidak menentukan jenis model apa yang harus dibuat maupun waktu pembuatannya dalam proses pengembangan perangkat lunak. UML tidak hanya berfungsi sebagai bahasa pemodelan visual, tetapi juga dapat diintegrasikan dengan berbagai bahasa pemrograman seperti Java, C++, dan Visual Basic, bahkan dapat terhubung dengan basis data berorientasi objek. Selain itu, UML juga mendukung proses dokumentasi dalam pengembangan sistem, seperti kebutuhan sistem (requirements), arsitektur, perancangan, kode sumber, rencana proyek, pengujian, serta pembuatan prototipe. Untuk memahami UML dengan baik, diperlukan

pemahaman terhadap konsep bahasa pemodelan serta tiga elemen utama yang dimilikinya, yaitu building block, aturan yang mengatur hubungan antar building block, serta mekanisme umum (D. Saputra et al., 2023).

Pemanfaatan *Unified Modeling Language* (UML) dalam tahap perancangan sistem memiliki peran penting dalam mendukung keberhasilan pengembangan sistem informasi berbasis web. UML menyediakan sarana visual yang membantu menggambarkan alur kerja, komponen, serta hubungan interaksi di dalam sistem. Dengan adanya pemodelan tersebut, kemungkinan terjadinya kesalahan pada tahap perancangan maupun implementasi dapat diminimalkan. Selain itu, pendekatan ini memudahkan pengembang dalam mendokumentasikan setiap bagian sistem secara terstruktur, memperlancar komunikasi antar anggota tim, serta mendukung proses validasi sistem sebelum diterapkan (Putri et al., 2025).

UML memiliki beberapa jenis diagram yang digunakan untuk menggambarkan berbagai aspek dalam suatu sistem, di antaranya *Use Case Diagram*, *Class Diagram*, dan *Sequence Diagram*. *Use Case Diagram* digunakan untuk menjelaskan fungsi sistem berdasarkan sudut pandang pengguna. *Class Diagram* menampilkan struktur objek yang terdapat dalam sistem beserta hubungan antar kelas. Sementara itu, *Sequence Diagram* menggambarkan urutan interaksi antar objek yang terjadi selama proses pemesanan berlangsung (Putri et al., 2025).

Dari beberapa pendapat tersebut, dapat dipahami bahwa *Unified Modeling Language* (UML) merupakan bahasa pemodelan standar dalam

rekayasa perangkat lunak yang digunakan untuk menggambarkan, merancang, dan mendokumentasikan struktur serta proses suatu sistem secara visual. UML mendukung pendekatan pemrograman berorientasi objek dan membantu pengembang memahami alur kerja, komponen, serta interaksi dalam sistem sehingga meminimalkan kesalahan pada tahap perancangan maupun implementasi. Selain itu, UML menyediakan berbagai jenis diagram, seperti *Use Case Diagram*, *Class Diagram*, dan *Sequence Diagram*, yang digunakan untuk merepresentasikan fungsi sistem, struktur objek, serta interaksi antar komponen dalam sistem.

13. MySQL

MySQL merupakan pengembangan proyek dari UNIREG yang selanjutnya diselesaikan oleh Michael Monty Widenius bersama TeX, yaitu perusahaan perangkat lunak asal Swedia. MySQL ini adalah DBMS yang bersifat bebas dapat dimanfaatkan oleh seluruh kalangan yang menyediakan dua jenis lisensi, yaitu Free Software dan Shareware. MySQL termasuk jenis kategori RDBMS (Relational Database Management System), yang memanfaatkan tabel, kolom, dan baris digunakan dalam pengelolaan data. Sebagai contoh, sebuah database di MySQL dapat memuat satu atau beberapa tabel sesuai kebutuhan (Siregar et al., 2024).

MySQL adalah implementasi dari sistem manajemen basis data relasional (RDBMS) yang dapat digunakan secara gratis. Pengguna bebas memanfaatkan MySQL, namun perangkat lunak ini tidak boleh dijadikan produk turunan untuk tujuan komersial. MySQL dikembangkan

berdasarkan salah satu konsep utama dalam basis data sebelumnya, yaitu SQL (*Structured Query Language*). SQL merupakan metode pengelolaan basis data, khususnya untuk seleksi dan input data, yang memungkinkan pengolahan informasi dilakukan dengan mudah dan otomatis. Sebagai server basis data *modern*, MySQL menawarkan sejumlah keunggulan yang membedakannya.

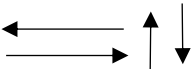
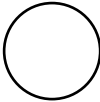
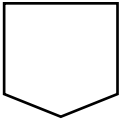
Berdasarkan penjelasan tersebut, dapat dipahami bahwa MySQL adalah implementasi sistem manajemen basis data *relasional* (RDBMS) yang bersifat *open source* dan dapat digunakan secara gratis, baik untuk keperluan pribadi maupun non-komersial. Sistem ini dikembangkan berdasarkan konsep SQL (*Structured Query Language*) yang mempermudah pengolahan data, termasuk proses seleksi dan input, secara otomatis. Sebagai server database modern, MySQL menawarkan keunggulan dalam pengelolaan data, menggunakan struktur tabel, kolom, dan baris, serta dapat mendukung berbagai aplikasi berbasis web dengan efisien.

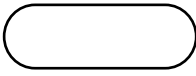
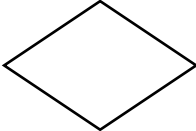
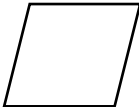
14. Flowchart

Flowchart merupakan representasi visual yang menunjukkan urutan langkah-langkah suatu proses atau prosedur penyelesaian masalah secara sistematis dan logis. Melalui penggunaan simbol-simbol tertentu, *flowchart* memudahkan penyajian setiap tahapan penyelesaian masalah sehingga alur proses dapat dipahami dengan lebih jelas dan terstruktur (Khesya, 2021).

Flowchart adalah diagram yang menyajikan alur proses atau tahapan kerja secara urut dalam menjalankan sebuah program. Melalui diagram ini, proses analisis, perancangan, hingga tahap pengkodean dapat digambarkan secara lebih jelas dan terperinci dalam kegiatan operasional. Penggunaan *Flowchart* umumnya bertujuan untuk membantu mempermudah proses penyelesaian masalah, khususnya ketika dilakukan peninjauan atau evaluasi pada tahap selanjutnya (Listyoningrum et al., 2023).

Tabel 2. 1 Simbol *Flowcart*

Nama	Simbol	Keterangan
<i>Document</i> (Dokumen)		Menyatakan masukan/keluaran berupa dokumen atau laporan tercetak.
<i>Process</i> (Proses)		Menyatakan proses pengolahan atau perhitungan data.
<i>Manual Input</i>		Menyatakan pemasukan data secara manual melalui keyboard.
<i>Flow Line</i> (Garis Alir)		Menunjukkan arah aliran proses dari satu langkah ke langkah berikutnya.
<i>Connector</i> (Penghubung)		Penghubung bagian-bagian flowchart pada halaman yang sama.
<i>Off-Page Connector</i>		Penghubung flowchart yang berlanjut ke halaman yang berbeda

<i>Terminator</i> (Terminal)		Menyatakan awal (Mulai) dan akhir (Selesai) dari suatu program atau proses.
<i>Decision</i> (Keputusan)		Menyatakan percabangan atau pengambilan keputusan berdasarkan kondisi (ya/tidak).
<i>Input/Output</i> (Data)		Menyatakan proses masukan (input) atau keluaran (output) data.
<i>Disk / on-line</i> <i>Storage</i>		Menyatakan input yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i>

(Sinurat et al., 2021)

15. Use Case Diagram

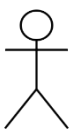
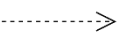
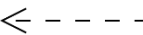
Use Case Diagram terdiri atas aktor, use case, serta hubungan yang menggambarkan interaksi di dalam sistem. Diagram ini berperan penting dalam memvisualisasikan, menjelaskan, dan mendokumentasikan kebutuhan perilaku sistem (Revita et al., 2023).

Use Case Diagram merupakan diagram yang digunakan untuk menggambarkan hubungan antara aktor, use case, dan interaksi yang terjadi dalam suatu sistem. Diagram ini berfungsi untuk memberikan gambaran awal mengenai sistem yang akan dikembangkan (Ramadani, 2025).

Use Case menggambarkan komunikasi antara satu atau lebih aktor dengan sistem informasi yang akan dikembangkan. Use case digunakan untuk mengidentifikasi fungsi-fungsi yang terdapat dalam sistem serta

menentukan aktor yang memiliki hak untuk menggunakan fungsi tersebut (Pradita et al., 2023).

Tabel 2. 2 Simbol *Use Case Diagram*

No	Nama Simbol	Gambar	Keterangan
1	Actor		Menunjukkan peran yang dijalankan oleh pengguna ketika berinteraksi dengan use case dalam suatu sistem.
2	Dependency		Menggambarkan hubungan ketika perubahan pada elemen yang berdiri sendiri dapat memengaruhi elemen lain yang bergantung padanya.
3	Generalization		Menunjukkan hubungan pewarisan, yaitu objek turunan memperoleh perilaku dan struktur data dari objek induknya.
4	Include		Menjelaskan bahwa suatu use case sumber menyertakan use case lain secara langsung dan jelas.
5	Extend		Menunjukkan bahwa use case target dapat memperluas perilaku dari use case sumber pada kondisi atau titik tertentu.
6	Association		Menggambarkan hubungan atau keterkaitan antara satu objek dengan objek lainnya.

7	Sistem		Menunjukkan batasan sistem yang menggambarkan ruang lingkup fungsi atau proses yang terdapat di dalam sistem.
---	--------	---	---

(Malius et al., 2021)

16. *Activity diagram*

Activity diagram adalah salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur aktivitas atau proses kerja dalam suatu sistem. Diagram ini dapat digunakan untuk memodelkan proses bisnis, alur logika program, maupun interaksi antara pengguna dengan sistem. *Activity diagram* menampilkan urutan aktivitas mulai dari awal hingga akhir, termasuk proses pengambilan keputusan, percabangan, dan aktivitas yang berjalan secara paralel. Komponen utama dalam *Activity diagram* meliputi Initial Node sebagai penanda awal proses, Activity sebagai representasi urutan aktivitas, Decision Node sebagai percabangan berdasarkan kondisi tertentu, serta Final Node sebagai penanda akhir proses. Selain itu, Fork Node dan Join Node digunakan untuk menggambarkan aktivitas yang berjalan secara bersamaan dan kemudian bergabung kembali. Untuk memperjelas pembagian tanggung jawab, Swimlanes dapat digunakan untuk mengelompokkan aktivitas berdasarkan aktor atau unit kerja yang terlibat.(Putri et al., 2025).

Activity diagram atau diagram aktivitas merupakan diagram yang digunakan untuk memodelkan proses-proses yang terjadi dalam suatu sistem. Rangkaian proses tersebut umumnya digambarkan secara vertikal

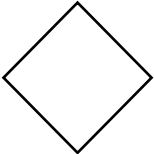
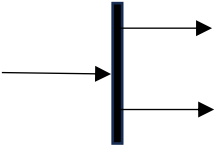
dari awal hingga akhir. Diagram ini merupakan pengembangan dari Use Case karena menampilkan alur aktivitas dalam sistem secara lebih terstruktur (Sutiyono & Ramadan, 2025).

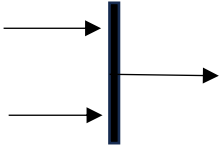
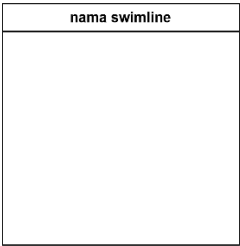
Menurut Amanda & Nopriani (2024) Selain digunakan untuk menggambarkan alur perpindahan aktivitas, *Activity diagram* juga dapat dimanfaatkan untuk menjelaskan perpindahan antar aktivitas yang dikerjakan sistem. Rancangan proses bisnis menggambarkan urutan aktivitas yang menunjukkan alur proses pada sistem yang telah ditentukan. Beberapa aspek yang dijelaskan dalam rancangan ini meliputi:

- a. Urutan atau pengelompokan tampilan sistem maupun antarmuka pengguna (user interface), dengan asumsi bahwa setiap aktivitas memiliki tampilan antarmuka tersendiri.
- b. Rancangan pengujian, karena setiap aktivitas memerlukan kasus uji untuk memastikan fungsionalitasnya berjalan sesuai kebutuhan.
- c. Rancangan menu yang akan ditampilkan pada perangkat lunak.

Activity diagram juga dapat digunakan untuk menjelaskan interaksi proses atau tampilan dalam suatu sistem. Diagram ini terdiri atas beberapa komponen dengan bentuk tertentu yang saling dihubungkan menggunakan tanda panah. Tanda panah tersebut menunjukkan urutan aktivitas yang berlangsung dari awal hingga akhir (Shidqi et al., 2025).

Tabel 2. 3 Simbol *Activity diagram*

Simbol	Nama	Keterangan
	Start State	Menunjukkan titik awal dari suatu alur kerja atau proses. Simbol ini menjadi penanda pertama ketika aktivitas mulai dijalankan.
	Activity	Menggambarkan kegiatan atau langkah yang dilakukan dalam suatu proses. Simbol ini biasanya berbentuk kotak yang berisi nama aktivitas yang sedang berlangsung.
	Decision	Menunjukkan titik ketika sistem harus menentukan pilihan berdasarkan kondisi tertentu. Percabangan ini digunakan apabila terdapat lebih dari satu kemungkinan aktivitas yang dapat dilakukan.
	End State	Menandai akhir dari suatu alur kerja atau proses. Simbol ini menunjukkan bahwa seluruh aktivitas telah selesai dijalankan.
	Fork	Menunjukkan bagian dalam proses ketika dua atau lebih aktivitas dapat berjalan secara bersamaan atau paralel.

	Join	Menunjukkan titik penggabungan kembali dari beberapa jalur aktivitas yang sebelumnya berjalan secara terpisah. Simbol ini merupakan kebalikan dari Fork.
	Swimlane	Digunakan untuk memisahkan area aktivitas dalam <i>Activity diagram</i> berdasarkan peran atau pihak yang terlibat. Swimlane biasanya diberi label agar tanggung jawab setiap bagian proses dapat terlihat dengan jelas.

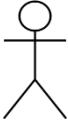
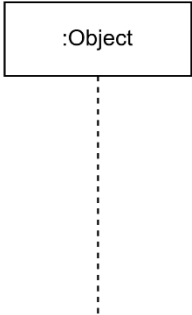
(Tria & Lionie, 2025)

17. *Sequence Diagram*

Sequence Diagram merupakan salah satu diagram UML yang digunakan untuk menggambarkan urutan interaksi antarobjek dalam suatu sistem. Diagram ini menunjukkan alur pesan atau komunikasi yang terjadi di antara objek, sehingga dapat membantu memahami keterkaitan proses yang berjalan di dalam sistem. *Sequence Diagram* juga digunakan untuk memetakan skenario pengguna, menjelaskan cara kerja sistem, serta mengidentifikasi kebutuhan fungsional. Melalui alur pesan antar objek, penerapan suatu fitur atau fungsi dapat diamati secara lebih jelas. Dalam perancangan sistem, diagram ini membantu tim memahami hubungan antar komponen dan menyamakan pemahaman terhadap proses yang dirancang. Selain itu, *Sequence Diagram* dapat menjadi acuan dalam implementasi

kode agar interaksi antar objek berjalan sesuai dengan rancangan. Dengan urutan interaksi yang tersaji secara jelas, diagram ini memudahkan komunikasi antara tim teknis dan pihak bisnis, serta membantu mengidentifikasi kemungkinan masalah sejak tahap awal. Secara umum, *Sequence Diagram* merupakan alat pemodelan yang penting untuk memahami dan mendokumentasikan dinamika interaksi dalam sistem (Putri et al., 2025).

Tabel 2. 4 *Sequence Diagram*

Simbol	Nama	Keterangan
	Aktor	Menunjukkan seseorang, pengguna, atau pihak lain yang berinteraksi dengan sistem yang sedang dikembangkan.
	Objek	Digunakan untuk menampilkan atau menambahkan objek baru yang terlibat dalam <i>Sequence Diagram</i> .
	Aktivasi	Menggambarkan proses atau tahapan yang dilakukan oleh objek selama alur kerja berlangsung.
	Pengulangan	Menunjukkan adanya pesan yang dikirim kembali kepada objek itu sendiri.

	Pesan	Menggambarkan komunikasi atau pengiriman pesan yang terjadi antara dua objek dalam sistem.
---	--------------	--

(Shidqi et al., 2025)

18. *State Diagram*

State Diagram merupakan diagram yang memiliki sifat yang sering terjadi perubahan dan digunakan untuk menggambarkan perubahan suatu kondisi dalam setiap kejadian pada sistem. Diagram ini memuat state, transisi, kejadian, serta aktivitas yang terjadi selama sistem berjalan (Hasanudin, 2020).

Tabel 2. 5 *State Diagram*

Simbol	Nama	Keterangan
	State	Menggambarkan keadaan atau kondisi suatu objek maupun sistem pada saat tertentu. Setiap state dapat diberi label atau uraian untuk menjelaskan sifat, status, atau kondisi yang sedang berlangsung.
	Start State	Berbentuk lingkaran yang menunjukkan titik awal dimulainya proses atau aktivitas suatu objek.
	End State	Berbentuk lingkaran yang menunjukkan titik akhir atau berakhirnya proses suatu objek.

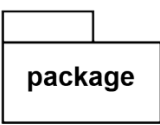
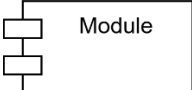
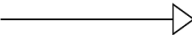
	Transition	Digambarkan dengan panah yang menunjukkan perpindahan dari satu kondisi objek ke kondisi lainnya. Transisi juga dapat mengarah kembali ke state yang sama apabila terjadi pengulangan kondisi.
---	------------	--

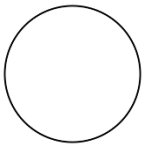
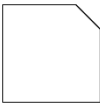
(Salam et al., 2023)

19. *Component diagram*

Component diagram merupakan diagram UML yang digunakan untuk menggambarkan komponen-komponen dalam suatu sistem beserta hubungan antar komponen tersebut. Dalam dokumentasi sistem yang kompleks, diagram ini membantu menguraikan sistem menjadi bagian-bagian yang lebih kecil sehingga lebih mudah dipahami (Gani & Awaludin, 2024).

Tabel 2. 6 *Component diagram*

No	Gambar	Nama	Keterangan
1		Package	Wadah yang berisi satu atau beberapa komponen di dalamnya.
2		Komponen	Bagian atau elemen penyusun yang terdapat dalam sistem.
3		Kebergantungan / Dependency	Hubungan ketergantungan antar komponen, dengan arah panah menuju komponen yang digunakan.

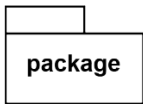
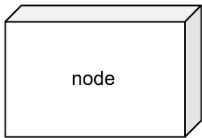
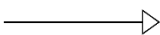
4		Antarmuka / Interface	Memiliki konsep yang sama dengan interface pada pemrograman berorientasi objek, yaitu sebagai penghubung atau antarmuka antar komponen.
5		Link	Hubungan atau relasi yang menghubungkan antar komponen.
6		Dokumen	Dapat berbentuk file, library, atau dokumen pendukung lainnya.

(Sukamto & Shalahuddin, 2018)

20. *Deployment Diagram*

Deployment Diagram adalah salah satu diagram UML yang digunakan untuk menggambarkan penempatan artefak pada node. Diagram ini berfungsi untuk memvisualisasikan hubungan antara perangkat lunak dan perangkat keras. Secara khusus, *Deployment Diagram* menyajikan model fisik mengenai bagaimana komponen perangkat lunak atau artefak diterapkan pada komponen perangkat keras yang disebut node. Diagram ini termasuk diagram penting pada tahap implementasi perangkat lunak dan umumnya dibuat sebelum proses coding dilakukan (Pulungan & Purwanto, 2023).

Tabel 2. 7 Simbol *Deployment Diagram*

No	Gambar	Nama	Keterangan
1		Package	Tempat pengelompokan yang memuat satu atau beberapa node.
2		Node	Biasanya merujuk pada perangkat keras atau perangkat lunak yang berasal dari luar sistem. Jika suatu node memuat komponen untuk menjaga konsistensi rancangan, maka komponen yang ditampilkan harus sesuai dengan komponen yang sudah dijelaskan pada diagram komponen.
3		Kebergantungan / Dependency	Menggambarkan ketergantungan antara node, dengan arah panah mengarah pada node yang dipakai.
4		Link	Relasi yang menghubungkan satu node dengan node lainnya.

(Sukanto & Shalahuddin, 2018)

B. Kajian Empiris

Pertama, penelitian oleh (M. J. Saputra & Suryono, 2024) mengembangkan sistem irigasi tetes otomatis pada tanaman jagung menggunakan sensor kelembapan tanah yang dikendalikan oleh mikrokontroler ESP32. Sistem ini dirancang untuk memantau kadar air tanah secara *real-time* dan melakukan penyiraman otomatis ketika kelembapan tanah berada di bawah ambang batas yang ditentukan. Hasil penelitian menunjukkan bahwa sistem tersebut mampu

meningkatkan efisiensi penggunaan air, menjaga kelembapan tanah tetap optimal, serta mengurangi kebutuhan penyiraman manual dalam budidaya jagung.

Kedua, dalam penelitian oleh Nurhaliza (2025) dikembangkan *sistem irigasi otomatis berbasis Internet of Things (IoT)* untuk optimalisasi penggunaan air pada lahan pertanian kering. Sistem ini memanfaatkan sensor kelembapan tanah, ESP32, dan pompa otomatis yang terhubung ke *dashboard mobile*, serta diuji pada lahan kering seluas 1,5 hektar. Penelitian ini menunjukkan bahwa penggunaan air dapat turun hampir 28,6% dan produktivitas tanaman meningkat dengan pengaturan irigasi otomatis berbasis data sensor *real-time*.

Ketiga, Stiawan & Supardi (2024) melakukan penelitian yang merancang alat penyiram tanaman cabai otomatis berbasis ESP32 dan sensor kelembapan tanah yang mempertimbangkan dua parameter utama, yaitu kelembapan tanah dan waktu penyiraman. Penelitian ini menunjukkan bahwa tingkat kelembapan tanah sangat memengaruhi durasi penyiraman, dan sistem yang dirancang dapat bekerja dengan efektif dalam memberikan air secara otomatis dengan mempertimbangkan kondisi kelembapan.

Keempat, penelitian oleh Wahyudi et al (2025) meneliti *implementasi sistem irigasi otomatis berbasis Internet of Things (IoT)* untuk pertanian greenhouse menggunakan mikrokontroler ESP32 dan sensor kelembapan tanah. Dalam sistem ini, data sensor dikirimkan ke platform monitoring seperti ThingSpeak untuk analisis *real-time*, sementara penyiraman dikontrol

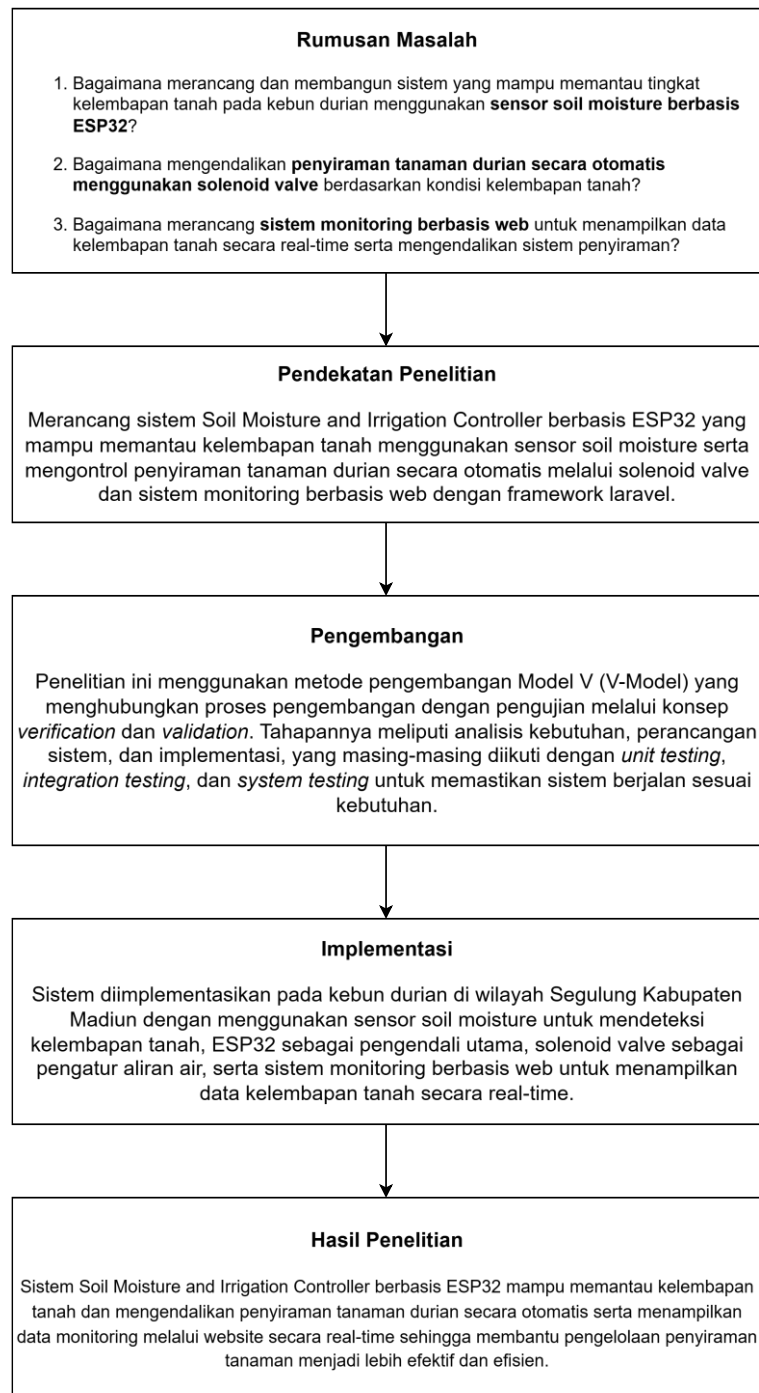
berdasarkan nilai kelembapan tanah yang terukur. Hasil penelitian menunjukkan bahwa pendekatan berbasis *Internet of Things* (IoT) dapat meningkatkan efisiensi penggunaan air dan memungkinkan pemantauan jarak jauh terhadap kondisi kelembapan tanah.

Kelima, penelitian oleh Alrasyid et al. (2025) mengembangkan *sistem irigasi otomatis berbasis Internet of Things (IoT)* yang menggunakan mikrokontroler ESP32, untuk membangun *smart home* dengan menggunakan teknologi *Internet of Things (IoT)* untuk mengontrol pencahayaan rumah guna menciptakan rumah yang hemat energi.

Kajian empiris menunjukkan bahwa penggunaan ESP32 sebagai mikrokontroler dalam sistem irigasi otomatis berbasis sensor kelembapan tanah memiliki potensi untuk meningkatkan efisiensi penggunaan air dan mempermudah pengelolaan penyiraman tanaman secara otomatis. Namun, efektivitas sistem ini dapat dipengaruhi oleh beberapa faktor, seperti jenis tanaman, kondisi tanah, dan lingkungan sekitar. Oleh karena itu, penelitian lebih lanjut diperlukan untuk mengembangkan sistem *soil moisture and Irrigation Controller* yang lebih adaptif, handal, dan sesuai dengan karakteristik kebun durian di wilayah Segulung, Kabupaten Madiun, sehingga dapat menghasilkan manajemen irigasi yang optimal dan berkelanjutan.

C. Kerangka Berpikir

Berdasarkan latar belakang dan kajian teori diatas maka kerangka berfikir penelitian ini adalah sebagai berikut:



Gambar 2.1 Kerangka Berpikir