

BAB II

KAJIAN PUSTAKA

A. KAJIAN TEORITIS

1. Rancang Bangun

Rancang Secara umum, rancang bangun dapat dipahami sebagai rangkaian kegiatan merencanakan sekaligus membangun suatu sistem guna menjawab kebutuhan tertentu, yang mencakup tahap analisis, perancangan, implementasi, dan pengujian. Arifianto, (2023) mengartikan rancang bangun sebagai upaya menerjemahkan hasil analisis menjadi paket perangkat lunak, lalu mewujudkan sistem baru atau menyempurnakan sistem yang telah ada sebelumnya. Di sisi lain, memandang rancang bangun sebagai rangkaian pengembangan sistem yang berjalan secara bertahap, dimulai dari penggalan kebutuhan hingga evaluasi akhir sistem. Kedua pandangan ini menegaskan bahwa rancang bangun tidak semata soal membangun, melainkan juga menuntut perencanaan yang cermat. Wulandari & Nurmiati,(2022) menambahkan bahwa rancang bangun adalah proses rekayasa yang tersusun rapi untuk menghasilkan sistem yang berfungsi sesuai tujuan yang telah digariskan, dengan memadukan perangkat keras dan perangkat lunak agar tercipta sistem yang saling terhubung dan bekerja secara optimal demi memenuhi kebutuhan penggunaanya. Pandangan serupa juga muncul dalam buku *Rekayasa Perangkat Lunak*,(2024) terbitan PT Mafy Media

LiterasiIndonesia, yang menerangkan bahwa rancang bangun mencakup pemahaman dasar rekayasa perangkat lunak, tahapan pengembangannya. Atas dasar itu, istilah rancang bangun pada penelitian ini merujuk pada upaya mengembangkan kontrol air pada media pembibitan tanaman yang dilengkapi sistem *sprinkler* otomatis berbasis IoT.

2. Kontrol Air Pada Media Pembibitan Tanaman

Kontrol air pada media pembibitan tanaman dapat dimaknai sebagai ruang atau wadah dengan kondisi terkendali yang ditujukan untuk menumbuhkan bibit tanaman sekaligus mempertahankan kestabilan faktor lingkungan seperti suhu, kelembapan udara, kelembapan tanah, dan intensitas cahaya Hadi, (2023) menyebut kontrol air pada media penyemaian sebagai alat yang berperan menjaga kestabilan suhu dan kelembapan lingkungan sehingga proses perkecambahan dan pertumbuhan bibit dapat berlangsung secara optimal. Pendapat ini selaras dengan Maarif et al., (2024) yang memaknai kontrol air pada media pembibitan tanaman sebagai wadah terkendali untuk menciptakan iklimat ideal bagi pertumbuhan bibit pada fase awal. Stabilitas kondisi lingkungan selama masa pembibitan menjadi hal krusial karena berdampak langsung pada daya tumbuh, keseragaman, serta mutu bibit yang dihasilkan. Melalui kontrol air pada media pembibitan, faktor lingkungan yang sulit dikendalikan pada lahan terbuka dapat diatur secara lebih presisi sehingga kemungkinan gagalnya proses pembibitan dapat ditekan.

3. Benih

Benih merupakan biji tanaman yang telah dipersiapkan dan digunakan untuk memperbanyak atau mengembangbiakkan tanaman. Menurut Komarudin, (2026), benih merupakan bahan pertama yang digunakan petani untuk menanam dan sangat menentukan pertumbuhan serta hasil produksi tanaman. Sementara itu, Kushariadi, (2024) menyatakan bahwa kualitas benih sangat memengaruhi pertumbuhan tanaman, tingkat produktivitas, dan efisiensi penggunaan *input* dalam kegiatan budidaya. Benih yang bermutu memiliki mutu fisik, fisiologis, dan genetik yang tinggi sehingga memiliki daya kecambah dan daya tumbuh yang baik. Ashar et al., (2024) dalam buku *Ilmu dan Teknologi Benih*, yang menjelaskan bahwa mutu benih ditentukan melalui proses pembentukan biji, pematangan, hingga pemrosesan dan penyimpanan yang tepat untuk menjaga daya kecambah serta produktivitas tanaman. Dalam penelitian ini, benih menjadi objek yang dipelihara agar memperoleh kondisi lingkungan yang optimal pada masa awal pertumbuhannya.

4. *Internet of Things* (IoT)

Komponen *Internet of Things* (IoT) merupakan gabungan perangkat keras dan perangkat lunak yang saling terhubung untuk mengumpulkan, mengirim, dan mengolah data melalui jaringan internet. Menurut Mahfud, (2023), IoT adalah teknologi yang memungkinkan berbagai perangkat saling terhubung dan bertukar data melalui jaringan internet untuk

mengumpulkan serta memantau informasi dari objek fisik. Sementara itu, Manuhutu, (2025) menyatakan bahwa IoT memungkinkan suatu objek mengirimkan data melalui jaringan tanpa memerlukan interaksi manusia ke manusia maupun manusia ke komputer. Selain itu, menurut Evizal dalam buku *Internet of Things*, IoT merupakan konsep konektivitas antarperangkat yang memungkinkan pengumpulan dan pertukaran data secara otomatis guna mendukung efisiensi di berbagai bidang kehidupan, termasuk pertanian dan otomasi rumah tangga. Dalam penelitian ini, komponen IoT yang digunakan meliputi perangkat berikut.

a. ESP32

ESP32 adalah *system on a chip* (SoC) hemat daya yang telah dilengkapi modul *Wi-Fi* dan *Bluetooth* secara terintegrasi. Nizam Yuana, H., & Wulansari, Z.,(2022) menjelaskan ESP32 sebagai SoC berbasis Wi-Fi yang dirancang khusus untuk aplikasi IoT dengan konsumsi daya rendah dan dapat diprogram sebagai pusat kendali sistem. Sementara Oktafiani, (2023) memaparkan bahwa ESP32 merupakan penyempurnaan dari ESP8266 dengan kemampuan pemrosesan *dual-core* serta jumlah pin yang lebih banyak, menjadikannya sesuai untuk sistem kendali berbasis internet.

b. *Capacitive Soil Moisture Sensor*

Capacitive soil moisture sensor adalah sensor yang berfungsi mengukur kadar air pada media tanam melalui perubahan nilai kapasitansi. Mardiyati Dewi, T., & Oktarina, Y., (2024)

mengungkapkan bahwa sensor kelembapan tanah berjenis kapasitif lebih tahan terhadap korosi dibandingkan jenis resistif sehingga usia pakainya lebih panjang. Adapun Mahfud, (2023) menerangkan bahwa cara kerja sensor ini adalah mendeteksi perubahan konstanta dielektrik tanah seiring kadar airnya, lalu mengonversinya menjadi sinyal analog yang dapat dibaca mikrokontroler.

c. Pompa Sock Drat

Pompa air bertugas mengalirkan air dari penampungan menuju nozzle *sprinkler* untuk proses penyiraman. Driptufany (2024) mendefinisikan pompa sebagai perangkat yang mengonversi energi mekanik menjadi energi fluida guna memindahkan cairan. Yuliana, (2025) menambahkan bahwa pada sistem berbasis IoT, pompa DC mini kerap dipilih sebagai aktuator penyiraman karena kemudahan pengendaliannya melalui relay serta konsumsi dayanya yang rendah.

d. Board ESP32

Board ESP32 merupakan papan pengembangan (*development board*) yang memadukan chip ESP32 dengan port USB, regulator tegangan, dan pin GPIO sehingga mempermudah proses pemrograman dan perakitan rangkaian. Manuhutu, (2025) menyebutkan bahwa *board* NodeMCU ESP32 mempermudah pengembang dalam membangun prototipe IoT karena sudah tersedia konektivitas dan antarmuka siap pakai. Nizam Yuana, H., & Wulansari, Z., (2022) menambahkan

bahwa *board* ini berperan sebagai pusat kendali yang membaca data sensor sekaligus mengendalikan aktuator.

e. Kabel *Jumper*

Kabel *jumper* merupakan kabel penghubung yang digunakan untuk menghubungkan antarkomponen elektronika pada *breadboard* maupun papan mikrokontroler. Menurut Hadriansa., (2025), kabel *jumper* adalah konektor yang memungkinkan aliran listrik dan sinyal antarkomponen tanpa memerlukan penyolderan. Sementara itu, Mukhlison S., & Putra, I. A., (2024) menjelaskan bahwa kabel *jumper* merupakan kabel elektrik dengan konektor pin pada setiap ujungnya yang memungkinkan penghubungan antarkomponen elektronik, termasuk pada rangkaian berbasis Arduino, tanpa memerlukan penyolderan, serta tersedia dalam tipe *male-to-male*, *male-to-female*, dan *female-to-female* sesuai kebutuhan sambungan.

f. Adaptor *Power Supply* 12V

Adaptor *power supply* 12V berfungsi sebagai sumber tegangan untuk menyuplai daya ke komponen sistem, khususnya pompa. Menurut Mukhlison S., & Putra, I. A., (2024), *power supply* merupakan perangkat yang mengubah tegangan listrik AC menjadi tegangan DC yang stabil sesuai kebutuhan perangkat. Sementara itu, Hastaningtyas, (2023) menyatakan bahwa pemilihan adaptor dengan tegangan dan arus yang sesuai sangat penting agar komponen bekerja optimal dan aman.

g. Konektor Jack

Konektor jack (*DC jack*) merupakan komponen penghubung antara adaptor *power supply* dengan rangkaian sistem. Menurut Hadriansa., (2025), konektor jack DC berfungsi menyalurkan tegangan dari adaptor ke rangkaian secara praktis dan mudah dilepas-pasang. Sementara itu, Hastaningtyas, (2023) menjelaskan bahwa penggunaan konektor jack memudahkan proses instalasi serta perawatan perangkat elektronika.

h. *Step Down*

Modul *step down* (*buck converter*) merupakan rangkaian penurun tegangan DC yang berfungsi mengubah tegangan *input* menjadi tegangan *output* yang lebih rendah dan stabil sesuai kebutuhan komponen. Menurut Mahardi Sunuharjo, L., Hendrawan, D., 'Atiq, M., Wahyuadi, R. A., & Nugraha, S. P. A., (2024), IC LM2596 merupakan regulator *switching step-down* yang umum digunakan pada konverter *buck* karena mampu mengatur tegangan keluaran dengan presisi tinggi dan efisiensi yang lebih baik dibandingkan regulator linear. Sementara itu, Fuada dan Fuada & Husnaini, I., 2023) menjelaskan bahwa *buck converter* bekerja berdasarkan prinsip *switching* untuk menurunkan tegangan DC dari level yang lebih tinggi ke level yang lebih rendah dengan rugi daya yang kecil. Dalam penelitian ini, modul *step down* digunakan untuk menyesuaikan

tegangan dari adaptor 12V menjadi tegangan kerja mikrokontroler ESP32 (5V/3,3V) agar seluruh komponen dapat bekerja dengan aman.

5. Sistem *Sprinkler* Otomatis

Sistem *sprinkler* otomatis merupakan sistem penyiraman yang menyalurkan air dalam bentuk semprotan butiran atau kabut (*mist*) ke tanaman dan bekerja secara otomatis berdasarkan pembacaan sensor. Menurut Sirait Santoso, D., Sari, N., Hatta, S., & Hendris., (2022), sistem *sprinkler* adalah metode irigasi yang mendistribusikan air menyerupai hujan melalui nozzle bertekanan sehingga penyebaran air menjadi lebih merata. Sementara itu, Hanafi, (2025) menyatakan bahwa sistem penyiraman otomatis mampu mengatur pemberian air secara mandiri berdasarkan pembacaan sensor kelembapan tanah tanpa memerlukan campur tangan manusia, dengan waktu respons yang cepat serta efisiensi penggunaan air yang lebih baik. Pada penelitian ini, apabila melebihi *threshold* akan menyalurkan air melalui nozzle *sprinkler* dan berhenti otomatis ketika kelembapan telah mencapai nilai yang diinginkan, sehingga pemberian air menjadi lebih tepat jumlah dan efisien.

6. Arduino IDE

Arduino IDE (*Integrated Development Environment*) merupakan perangkat lunak yang digunakan untuk menulis, mengompilasi, dan mengunggah program (*sketch*) ke mikrokontroler. Menurut Kamal & Firdayanti, (2023), Arduino IDE adalah lingkungan pengembangan terintegrasi yang memudahkan pengguna dalam membuat dan

mengunggah program ke papan mikrokontroler. Sementara itu, Mukhlison S., & Putra, I. A., (2024) menyatakan bahwa Arduino IDE menggunakan bahasa pemrograman berbasis C/C++ dan mendukung berbagai jenis papan, termasuk ESP32. Sejalan dengan itu, Adfry Muskhir, M., & Luthfi, A., (2023) menemukan bahwa penggunaan Arduino IDE efektif dalam pembelajaran pemrograman mikrokontroler karena antarmukanya yang sederhana dan mudah dipahami oleh pemula. Arduino IDE juga banyak diterapkan pada berbagai penelitian rancang bangun berbasis mikrokontroler, seperti sistem *smart home* “Rancang Bangun Sistem Smart Home Berbasis Arduino,” (2023) dan kegiatan sosialisasi teknologi Arduino kepada masyarakat “Socialization of Arduino-Based Organic Waste Destruction Tool,” (2024), yang menunjukkan fleksibilitas Arduino IDE untuk berbagai kebutuhan pemrograman perangkat IoT ESP32.

7. *UML (Unified Modeling Language)*

Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang digunakan untuk merancang, menggambarkan, dan mendokumentasikan sistem perangkat lunak. Menurut Nistrina dan Nistrina & Sahidah, L., (2022), UML adalah bahasa pemodelan standar untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak. Sementara itu, Saputra Dharmawan, W. S., Syarif, M., & Risdiansyah, D., (2023) menyatakan bahwa UML terdiri atas berbagai diagram, seperti *use case*

diagram, *activity diagram*, dan *class diagram*, yang membantu menggambarkan alur dan struktur sistem. Pemodelan sistem dengan UML juga memberikan keunggulan dari segi perancangan, pengembangan, serta pemahaman dan komunikasi antara developer dan client Wayahdi & Ruziq, F., (2023). UML membantu memvisualisasikan hubungan antarkelas serta relasi antarobjek dalam sistem berorientasi objek, khususnya melalui *use case diagram*, *activity diagram*, dan *sequence diagram* Ramdany Kaidar, S. A., & Aguchino, B., (2024)). Beberapa penelitian rancang bangun sistem informasi juga memanfaatkan kombinasi diagram-diagram tersebut sebagai alat bantu perancangan sebelum tahap implementasi “Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS),” (2024). Buku Dasar-Dasar Rekayasa Perangkat Lunak Sumirat, (2023) serta Buku Ajar Rekayasa Perangkat Lunak (Umsida Press) turut menjelaskan bahwa UML tidak terikat pada produk tertentu sehingga fleksibel diterapkan pada berbagai jenis proyek pengembangan perangkat lunak. Visual Studio Code

Visual Studio Code (VS Code) merupakan perangkat lunak penyunting kode (code editor) sumber terbuka yang dikembangkan oleh Microsoft. Menurut Murani, (2025), VS Code adalah editor kode ringan yang mendukung banyak bahasa pemrograman serta dilengkapi fitur extension untuk meningkatkan produktivitas. Sementara itu, Nasri Kornelis, & Fauzi, D. M., (2022) menjelaskan bahwa VS Code

menyediakan fitur seperti syntax highlighting, debugging, dan integrasi Git yang memudahkan proses pengembangan aplikasi.

8. Laravel

Laravel merupakan *framework* pengembangan aplikasi web berbasis bahasa pemrograman PHP yang menggunakan pola arsitektur Model-View-Controller (MVC). Menurut Budianto A., & Said, R. M., (2025), Laravel adalah *framework* PHP yang dirancang untuk mempermudah dan mempercepat pengembangan aplikasi web dengan sintaks yang ekspresif. Sementara itu, Ariyanto Rachmad, M. F. F., & Puspitasari, D., (2024) menyatakan bahwa Laravel menyediakan berbagai fitur seperti routing, ORM Eloquent, dan sistem migration yang memudahkan pengelolaan basis data.

9. React.js

React.js merupakan *library* JavaScript sumber terbuka yang digunakan untuk membangun antarmuka pengguna (*user interface*), terutama pada aplikasi web berbasis komponen. Menurut Jonathan dan Supriyadi., (2023), React.js adalah *library* yang dikembangkan oleh Meta (Facebook) untuk membangun antarmuka yang interaktif dan efisien. Sementara itu, Putra Efendi, R. W., (2025) menjelaskan bahwa React.js menggunakan konsep *virtual DOM* dan *component-based* sehingga pembaruan tampilan menjadi lebih cepat dan terstruktur.

10. Web *Browser*

Web *browser* merupakan perangkat lunak yang digunakan untuk mengakses, menampilkan, dan berinteraksi dengan halaman web melalui jaringan internet. Menurut Inggi dan Alam, H. P., (2023), web *browser* adalah aplikasi yang menerjemahkan kode HTML, CSS, dan JavaScript menjadi tampilan yang dapat dibaca pengguna. Peramban menjadi antarmuka utama pengguna dalam memantau data sistem IoT berbasis web secara *real-time*. Pemilihan web *browser* yang tepat turut memengaruhi kenyamanan pengguna dalam mengakses sistem berbasis web, sebagaimana dianalisis menggunakan metode Analytical Hierarchy Process pada penelitian pemilihan web *browser* mobile “Pemilihan Web *Browser* pada Mobile Menggunakan Metode Analytical Hierarchy Process,”. Selain itu, perbandingan kinerja antar *browser* menunjukkan adanya perbedaan efisiensi penggunaan sumber daya sistem yang dapat memengaruhi kenyamanan pengguna (RIGGS: Journal of Artificial Intelligence and Digital Business, 2024).

11. Laragon

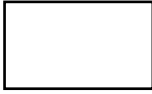
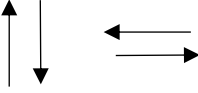
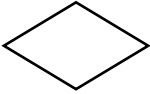
Laragon digunakan menjalankan aplikasi web secara lokal pada komputer sebelum di-*hosting*. Menurut Andarsyah, (2022), Laragon adalah lingkungan pengembangan lokal yang ringan dan cepat serta mendukung PHP, MySQL, dan Apache/Nginx. Sementara itu, Chandra dan Setyaningsih, (2023) menjelaskan bahwa Laragon memudahkan

pengembang dalam membuat dan menguji aplikasi web tanpa memerlukan koneksi internet.

12. Flowchart

Flowchart merupakan representasi grafis dari urutan langkah-langkah atau alur proses suatu sistem dengan menggunakan simbol-simbol tertentu. Menurut Nadiyah, (2023), *flowchart* adalah bagan yang menggambarkan urutan proses secara logis dan sistematis untuk memudahkan pemahaman alur program. Sementara itu, Saputra Dharmawan, W. S., (2023) menyatakan bahwa *flowchart* membantu perancang dalam menganalisis, mendokumentasikan, dan mengomunikasikan alur kerja sistem sebelum implementasi.

Tabel 2. 1 Simbol *Flowcart*

Simbol	Keterangan
	<i>Input/Output</i> Simbol ini digunakan untuk merepresentasikan data masukan maupun keluaran.
	Proses Simbol proses berfungsi untuk menggambarkan suatu aktivitas atau langkah pemrosesan.
	Garis Alir (flow line) Simbol ini digunakan untuk menunjukkan arah aliran dari suatu proses.
	Penghubung (connector) Simbol penghubung berfungsi untuk menandai keterkaitan antarbagian bagan alir yang terpisah, baik dalam halaman yang sama maupun berbeda.
	Keputusan (decision) Simbol ini digunakan untuk menunjukkan proses pemilihan kondisi dalam suatu program.

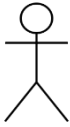
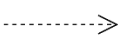
Simbol	Keterangan
	Proses Terdefinisi (predefined process) Simbol ini digunakan untuk menyatakan suatu proses yang penjelasannya dijabarkan di bagian lain.
	Persiapan (preparation) Simbol ini berfungsi untuk memberikan nilai awal pada suatu variabel atau besaran.
	Terminal (terminal point) Simbol ini digunakan untuk menunjukkan titik awal dan akhir dari suatu proses.

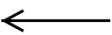
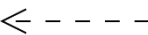
(Smrti et al., 2023).

13. Use Case Diagram

Use case diagram dideskripsikan sebagai interaksi pengguna dengan sistem yang akan dibangun serta dapat menggambarkan fungsi apa saja yang ada pada sebuah sistem informasi yang dibangun (Julianto & Setiawan, dalam Nistrina & Sahidah, L., (2022)). Pada penelitian ini, *use case diagram* menggambarkan interaksi antara aktor Pengguna dengan sistem kontrol air pada media pembibitan tanaman, seperti melihat data kelembapan tanah, memantau status pompa/*sprinkler*, serta melihat riwayat data pemantauan melalui aplikasi web.

Tabel 2. 2 Simbol *Use Case Diagram*

No	Nama Simbol	Gambar	Keterangan
1	Actor		Menunjukkan peran yang dijalankan oleh pengguna ketika berinteraksi dengan <i>use case</i> dalam suatu sistem.
2	Dependency		Menggambarkan hubungan ketika perubahan pada elemen yang berdiri sendiri dapat memengaruhi elemen

No	Nama Simbol	Gambar	Keterangan
			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 <i>use case</i> sumber menyertakan <i>use case</i> lain secara langsung dan jelas.
5	Extend		Menunjukkan bahwa <i>use case</i> target dapat memperluas perilaku dari <i>use case</i> 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)

14. *Activity* Diagram

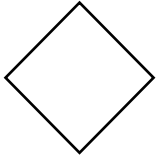
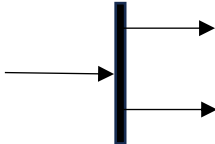
Activity diagram dapat digunakan untuk memodelkan proses bisnis, alur logika program, maupun interaksi antara pengguna dengan sistem. Komponen utama dalam *Activity* diagram meliputi Initial Node sebagai penanda awal proses, *Activity* sebagai representasi langkah-

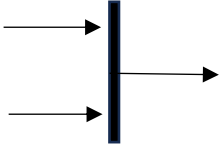
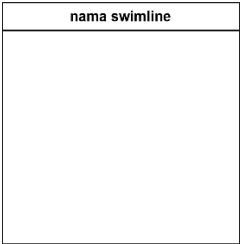
langkah aktivitas, Decision Node sebagai percabangan berdasarkan kondisi tertentu, serta Final Node sebagai penanda akhir proses. Selain itu, Fork Node dan Join Node digunakan untuk menggambarkan aktivitas yang berjalan secara bersamaan dan kemudian bergabung kembali. Untuk memperjelas pembagian tanggung jawab, Swimlanes dapat digunakan untuk mengelompokkan aktivitas berdasarkan aktor atau unit kerja yang terlibat Rahmawati & Kurniawan, (2023).

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 Sulistyorini & Setyowati, (2023).

Activity diagram juga dapat digunakan untuk mendefinisikan atau mengelompokkan alur 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 Hendri & Putra, (2024). Pada penelitian ini, *activity* diagram digunakan untuk menggambarkan alur aktivitas sistem mulai dari pembacaan sensor kelembapan tanah, pengiriman data ke server, hingga pengambilan keputusan mengaktifkan atau mematikan pompa dan *sprinkler* secara otomatis.

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.

Simbol	Nama	Keterangan
	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.

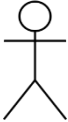
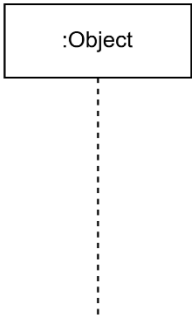
(Tria & Lionie, 2025)

15. *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. Melalui alur pesan antarobjek, 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. Secara umum,

Sequence Diagram merupakan alat pemodelan yang penting untuk memahami dan mendokumentasikan dinamika interaksi dalam sistem Sulistyorini & Setyowati, (2023). Pada penelitian ini, *sequence diagram* menggambarkan urutan interaksi antara sensor, mikrokontroler NodeMCU ESP32, *server*, dan aplikasi web, mulai dari pembacaan data kelembapan tanah hingga penampilan data pada antarmukapengguna.

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.

16. State Diagram

State iagram atau state chart diagram dideskripsikan sebagai diagram yang digunakan untuk menggambarkan perubahan status atau transisi status dari suatu mesin atau sistem dalam bentuk graf berarah, (Saputra Dharmawan, W. S., 2023). Pada penelitian ini, state diagram menggambarkan perubahan status pompa dan

Tabel 2. 5 *State Diagram*

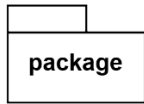
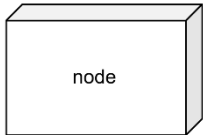
Simbol	Nama	Keterangan
	State	Menggambarkan keadaan atau kondisi suatu objek maupun sistem pada saat tertentu. Setiap state dapat diberi label atau uraian untuk menjelaskan sifat, status, atau kondisi yang sedang berlangsung.
	Start State	Berbentuk lingkaran yang menunjukkan titik awal dimulainya proses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

17. *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. Diagram ini termasuk diagram penting pada tahap implementasi perangkat lunak dan umumnya dibuat sebelum proses coding dilakukan Wijaya & Nugroho, (2023). Pada penelitian ini, *deployment diagram* menggambarkan penempatan komponen sistem secara fisik, meliputi mikrokontroler NodeMCU ESP32 beserta sensor dan aktuator yang terhubung ke jaringan Wi-Fi.

Tabel 2. 6 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.

No	Gambar	Nama	Keterangan
4		Link	Relasi yang menghubungkan satu node dengan node lainnya.

(Sukamto & Shalahuddin, 2018)

B. Kajian Empiris

Penelitian pertama dilakukan oleh Amroni & Wahid, (2025) dengan judul Rancang Bangun Sistem Penyiraman Otomatis Berbasis *Internet of Things* (IoT) pada Penyemaian Bibit Tanaman. Penelitian tersebut bertujuan merancang sistem penyiraman otomatis untuk menjaga kondisi media tanam selama proses penyemaian. Sistem yang dikembangkan mampu melakukan *monitoring* secara *real-time* serta mengaktifkan pompa air secara otomatis. Hasil penelitian menunjukkan bahwa sistem dapat membantu menjaga kelembapan media tanam secara lebih konsisten dibandingkan penyiraman manual. Persamaan penelitian tersebut dengan penelitian ini terletak pada penggunaan teknologi *Internet of Things*, mikrokontroler ESP32, sensor kelembapan tanah, dan sistem penyiraman otomatis. Perbedaannya adalah penelitian ini mengembangkan sebuah kontrol air pada media pembibitan tanaman yang dilengkapi sistem *sprinkler* otomatis serta *website monitoring* berbasis Laravel sehingga pengguna dapat memantau kondisi secara daring.

Penelitian kedua dilakukan oleh Nasrulloh & Sari., (2024) Penelitian tersebut mengembangkan sistem irigasi *sprinkler* yang dapat dikendalikan dari jarak jauh menggunakan teknologi *Internet of Things*. Sistem memungkinkan pengguna memantau kondisi irigasi serta mengendalikan penyiraman melalui

perangkat bergerak sehingga proses penyiraman menjadi lebih efisien. Hasil penelitian menunjukkan bahwa sistem mampu meningkatkan efektivitas pengelolaan irigasi pada tanaman hortikultura. Persamaan penelitian tersebut dengan penelitian ini adalah sama-sama menggunakan sistem *sprinkler* berbasis *Internet of Things* sebagai media penyiraman otomatis. Perbedaannya berfokus pada kontrol air pada media pembibitan tanaman serta menambahkan sistem *monitoring* berbasis web sebagai media pemantauan kondisi tanaman.

Penelitian ketiga dilakukan oleh Nasrulloh & Sari., (2024) dengan judul Algoritma dan Pemrograman *Sprinkler* Irrigation System Terintegrasi IoT. Penelitian ini mengembangkan sistem *sprinkler* yang memanfaatkan sensor kelembapan tanah, sensor suhu, serta komunikasi IoT untuk mengendalikan proses penyiraman secara otomatis. Sistem dirancang agar mampu memberikan penyiraman berdasarkan kondisi lingkungan sehingga penggunaan air menjadi lebih efisien. Hasil penelitian menunjukkan bahwa integrasi sensor dan teknologi IoT mampu meningkatkan efektivitas sistem irigasi otomatis. Persamaan Adalah sistem *sprinkler* dan sensor kelembapan tanah sebagai komponen utama penyiraman otomatis. Perbedaannya adalah penelitian ini menggunakan ESP32 sebagai pengendali utama serta menyediakan *website monitoring* berbasis Laravel untuk menampilkan data hasil pembacaan sensor secara *real-time*.

Penelitian keempat dilakukan oleh Abdiansyah & Maulidda., (2024) Penelitian tersebut mengembangkan sistem *monitoring* yang memanfaatkan sensor untuk mengukur kondisi lingkungan tanaman dan mengendalikan

penyiraman secara otomatis. Hasil penelitian menunjukkan bahwa sistem mampu meningkatkan efisiensi penggunaan air dan mempermudah pengguna dalam memantau kondisi tanaman melalui media digital. Teknologi *Internet of Things* dan sistem *monitoring* sebagai pendukung penyiraman otomatis. *Monitoring* yang digunakan, karena penelitian ini memanfaatkan *website* berbasis Laravel yang terintegrasi dengan *database* MySQL untuk menyimpan dan menampilkan data *monitoring*.

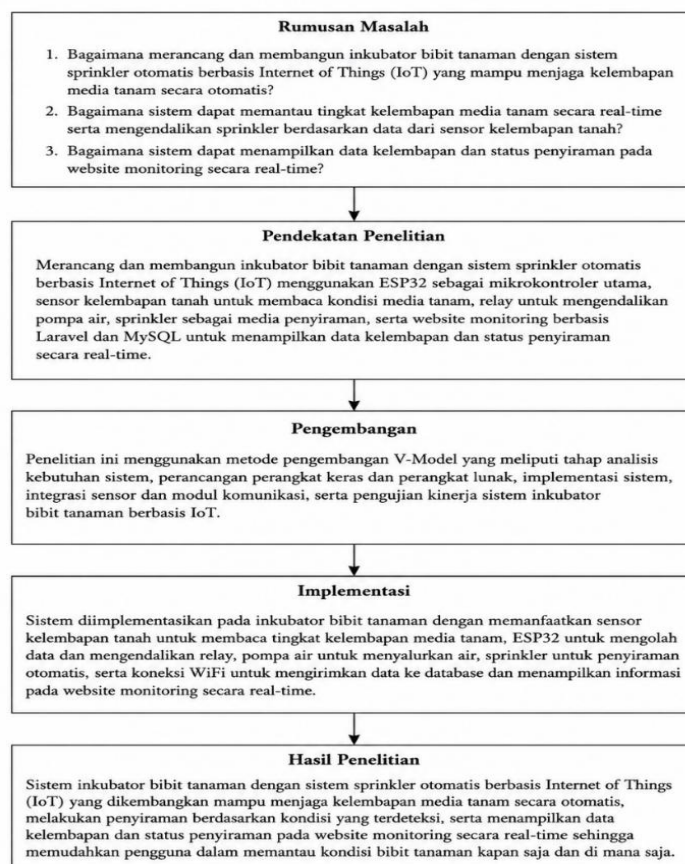
Penelitian kelima dilakukan oleh Sukirno & Nur'aini., (2025) Penelitian tersebut mengembangkan alat penyiraman tanaman otomatis yang memanfaatkan teknologi *Internet of Things* untuk mengoptimalkan penggunaan air dan memudahkan proses pemantauan tanaman secara *real-time*. Sistem yang dikembangkan menggunakan mikrokontroler dan sensor untuk mengatur proses penyiraman secara otomatis. Hasil penelitian menunjukkan bahwa penerapan IoT mampu meningkatkan efisiensi penyiraman serta mempermudah pengguna dalam mengelola tanaman. Persamaan penelitian tersebut dengan penelitian ini adalah sama-sama menerapkan konsep penyiraman otomatis berbasis *Internet of Things*. Perbedaannya adalah penelitian ini secara khusus mengembangkan kontrol air pada media pembibitan tanaman dengan sistem *sprinkler* otomatis serta menyediakan sistem *monitoring* berbasis web.

Berdasarkan hasil kajian terhadap beberapa penelitian terdahulu tersebut, dapat disimpulkan bahwa teknologi *Internet of Things* telah banyak diterapkan pada sistem *monitoring* dan penyiraman tanaman otomatis. Namun,

penelitian mengenai rancang bangun kontrol air pada media pembibitan tanaman dengan sistem *sprinkler* otomatis berbasis *Internet of Things* yang terintegrasi dengan *website monitoring* berbasis Laravel masih relatif terbatas. Oleh karena itu, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan teknologi pertanian, khususnya pada proses pembibitan tanaman.

C. Kerangka Berpikir

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



Gambar 2. 1 Kerangka Berpikir