

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Hidroponik adalah metode budidaya tanaman tanpa menggunakan tanah sebagai media tanam, melainkan dengan memanfaatkan air yang telah diperkaya dengan larutan nutrisi. Metode ini menawarkan sejumlah keunggulan seperti efisiensi penggunaan lahan, pengendalian lingkungan yang lebih optimal, serta peningkatan hasil produksi. Sistem ini sangat cocok diterapkan di wilayah perkotaan atau area dengan keterbatasan lahan pertanian. Berdasarkan kajian oleh Naresh et al. (2024) dalam *International Journal of Environment and Climate Change*, hidroponik terbukti menghemat hingga 90 % penggunaan air dibandingkan pertanian konvensional melalui sistem tertutup yang mendukung daur ulang larutan nutrisi [4].

Penelitian oleh Muhammad Hidayatullah dkk. (2022) yang dipublikasikan dalam *Jurnal Altron* mengembangkan sistem monitoring larutan hidroponik berbasis Arduino tanpa menggunakan *buzzer* sebagai media peringatan. Prinsip yang diadopsi dalam sistem ini juga sejalan, yaitu menjaga kestabilan pH larutan dalam instalasi hidroponik melalui dua pompa otomatis pompa *pH Up* dan pompa *pH Down* yang dikendalikan oleh sistem berbasis IoT. Sistem ini tidak menggunakan sensor *waterflow* maupun *buzzer*, namun memanfaatkan sensor pH untuk membaca kondisi larutan secara berkala. Saat nilai pH berada di luar rentang ideal, sistem akan mengaktifkan salah satu dari dua pompa untuk menambahkan larutan penyeimbang pH,

kemudian berhenti ketika nilai kembali ke kisaran optimal. Informasi kondisi pH dan status pompa dikirim secara otomatis ke sistem berbasis *website*, sehingga pengguna dapat memantau dan mengontrol proses ini secara *real-time* dari jarak jauh melalui koneksi internet. Dengan pendekatan ini, sistem menjadi lebih efisien, akurat, dan sangat relevan untuk diterapkan sebagai sarana edukasi teknologi terapan di lingkungan sekolah kejuruan. [5].

Sementara itu, penelitian oleh Fadillah Tarigan, Relita Buaton, dan Suci Ramadani (2023) dalam *Journal of Artificial Intelligence and Engineering Applications* merancang sistem *monitoring* hidroponik berbasis IoT menggunakan ESP32. Sistem tersebut memanfaatkan beberapa sensor seperti TDS, DHT11 (untuk suhu dan kelembaban). Data yang diperoleh dikirim secara *real-time* ke *website* melalui koneksi internet, sehingga pengguna dapat memantau kondisi hidroponik secara langsung dari perangkat seluler. Walaupun sistem ini menggunakan berbagai sensor tambahan, prinsip pengiriman data *real-time* ke platform *website* memberikan gambaran bahwa teknologi IoT sangat efektif diterapkan dalam sistem *monitoring* hidroponik. Dalam konteks ini, sistem yang dikembangkan juga mengusung pendekatan serupa dengan menggunakan ESP32, namun lebih disederhanakan dengan hanya melibatkan sensor pH, TDS, dan suhu tanpa kelembaban dan sensor ultrasonik, serta mengandalkan dua pompa otomatis untuk penyesuaian pH (*pH Up* dan *pH Down*) [6].

Penelitian lain oleh Hamdan Sulaiman, Ahmad Anas Yusof, dan Mohd Khairi Mohamed Nor (2025) dalam jurnal *AgriEngineering* menyoroti

pentingnya sistem dosing nutrisi otomatis berbasis pH dan EC melalui *scoping review* terhadap 89 penelitian. Salah satu rekomendasinya adalah penggunaan algoritma kontrol adaptif yang berjalan pada mikrokontroler seperti Arduino maupun ESP32, yang mampu merespons fluktuasi lingkungan dengan lebih presisi. Hal ini mendukung penerapan sistem yang dibangun dalam penelitian ini, karena penggunaan dua mikrokontroler secara bersama-sama memungkinkan proses pemantauan dan kontrol berjalan lebih akurat dan responsif. [7].

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan untuk merancang dan menggambarkan sistem perangkat lunak secara visual. UML memudahkan pengembang dan analis sistem dalam memahami struktur, perilaku, serta alur proses sistem sebelum tahap implementasi dilakukan. Diagram yang umum digunakan dalam perancangan sistem meliputi *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*, yang masing-masing berfungsi menggambarkan interaksi pengguna, alur aktivitas, urutan proses, serta struktur kelas dalam sistem. Penelitian oleh Sari dan Prasetyo (2023) menunjukkan bahwa penggunaan UML dalam tahap awal pengembangan perangkat lunak dapat meningkatkan akurasi dokumentasi kebutuhan sistem serta mempercepat proses desain dan implementasi sistem berbasis aplikasi, khususnya dalam proyek pengembangan sistem informasi layanan publik [8].

2.2 Landasan Teori

2.2.1 Sistem *Monitoring* Terintegrasi Berbasis *Website*

Sistem *monitoring* terintegrasi berbasis *website* adalah sistem informasi yang dirancang untuk memantau dan mengelola data dari berbagai komponen secara *real-time* melalui platform web yang dapat diakses secara daring. Sistem ini menggabungkan perangkat keras seperti sensor dan mikrokontroler dengan antarmuka berbasis web untuk menyajikan data yang akurat, terkini, dan mudah dipantau dari jarak jauh. Dalam konteks *monitoring* hidroponik, sistem ini berfungsi untuk memantau kondisi lingkungan tanaman seperti suhu lingkungan, tingkat kepekatan nutrisi (TDS), dan tingkat keasaman larutan (pH). Sistem ini dirancang menggunakan sensor pH, sensor TDS, dan sensor suhu (DHT) yang terhubung ke mikrokontroler ESP32 dan Arduino Uno, serta dikendalikan melalui relay dan dua pompa otomatis untuk menyesuaikan kadar pH dengan cara menambahkan larutan *pH Up* atau *pH Down* sesuai kebutuhan. Menurut penelitian oleh Fadillah Tarigan, Relita Buaton, dan Suci Ramadani (2023), sistem hidroponik berbasis IoT dan *website* yang terintegrasi dengan sensor-sensor lingkungan mampu menyajikan data secara *real-time* dan mempermudah pengguna dalam melakukan pemantauan tanpa harus hadir secara fisik. Konsep ini sejalan dengan sistem yang dikembangkan dalam penelitian ini, yang memungkinkan pengguna memantau dan mengendalikan sistem hidroponik melalui

website dari perangkat mana pun yang terhubung ke internet [9].

Menurut Menurut Muhammad Hidayatullah, Paris Ali Topan, dan Titi Andriani (2022), sistem pemantauan hidroponik otomatis berbasis Arduino Uno dan sensor TDS terbukti mampu mengendalikan kualitas larutan secara mandiri dengan mengaktifkan *dosing pump* secara otomatis saat konsentrasi nutrisi menyimpang dari ambang batas. Meskipun sistem mereka belum terintegrasi dengan platform web dan masih menggunakan *buzzer* sebagai alat peringatan lokal, penelitian ini menunjukkan pentingnya pemantauan digital dalam meningkatkan efisiensi pertanian hidroponik modern [10].

Dengan demikian, penerapan sistem *monitoring* hidroponik berbasis *website* di lingkungan pendidikan seperti SMK Negeri 3 Tegal sangat tepat untuk menjawab tantangan digitalisasi pertanian. Tidak hanya mendukung proses pembelajaran praktis, tetapi juga memperkenalkan konsep *Internet of Things* (IoT), otomasi, dan antarmuka pengguna secara langsung kepada peserta didik. Menurut Hamdan Sulaiman, Ahmad Anas Yusof, dan Mohd Khairi Mohamed Nor (2025), pendekatan berbasis *automated nutrient dosing* dan kontrol adaptif sangat penting untuk menjaga konsistensi hasil tanam, terutama jika dikembangkan ke dalam sistem berbasis web dengan loop umpan balik dan analisis prediktif, yang akan menjadi masa depan pertanian presisi di era Industri 4.0 [11].

2.2.2 SMK Negeri 3 Tegal

SMK Negeri 3 Tegal merupakan salah satu Sekolah Menengah Kejuruan Negeri yang berada di Kota Tegal, Provinsi Jawa Tengah, yang berfokus pada pendidikan vokasi berbasis keterampilan dan teknologi. Sekolah ini memiliki berbagai kompetensi keahlian, termasuk jurusan yang mendukung bidang pertanian, teknologi informasi, dan elektronika. Salah satu bentuk kegiatan pembelajaran yang dikembangkan adalah praktik budidaya tanaman hidroponik sebagai media pembelajaran kontekstual berbasis proyek (*project-based learning*).

Sebagai lembaga pendidikan kejuruan, SMK Negeri 3 Tegal memiliki tanggung jawab dalam menciptakan lulusan yang tidak hanya menguasai teori, tetapi juga memiliki keterampilan praktis yang sesuai dengan kebutuhan dunia industri dan teknologi. Dalam rangka mendukung hal tersebut, penerapan teknologi *monitoring* berbasis *Internet of Things* (IoT) dalam sistem hidroponik menjadi salah satu upaya untuk mengintegrasikan pembelajaran dengan perkembangan teknologi digital terkini. Berdasarkan Permendikbud Nomor 34 Tahun 2018 tentang Standar Nasional Pendidikan Kejuruan, sekolah kejuruan didorong untuk mengembangkan lingkungan belajar yang inovatif dan kolaboratif melalui integrasi teknologi dalam kegiatan belajar mengajar. Oleh karena itu, pengembangan sistem *monitoring* hidroponik terintegrasi *website* di SMK Negeri 3 Tegal menjadi

langkah strategis dalam meningkatkan efektivitas pembelajaran, efisiensi pemeliharaan tanaman, serta sebagai sarana pengenalan dan penerapan teknologi berbasis IoT di dunia pendidikan [12].



Gambar 2. 1 SMK Negeri 3 Tegal

2.2.3 Website

Website atau lebih dikenal dengan sebutan Web, dapat diartikan sebagai suatu kumpulan halaman yang menampilkan berbagai macam informasi seperti teks, data, gambar diam maupun bergerak, animasi, suara, video, maupun gabungan dari semuanya. Halaman-halaman tersebut dapat bersifat statis maupun dinamis dan saling terhubung melalui jaringan tautan atau *hyperlink*.

Tujuan dari penggunaan *web service* adalah agar client yang menggunakan berbagai macam platform, baik itu desktop, *website*, maupun *mobile apps*, dapat mengakses layanan yang berada di server web secara terintegrasi (Berners-Lee et al., 2001) [13].

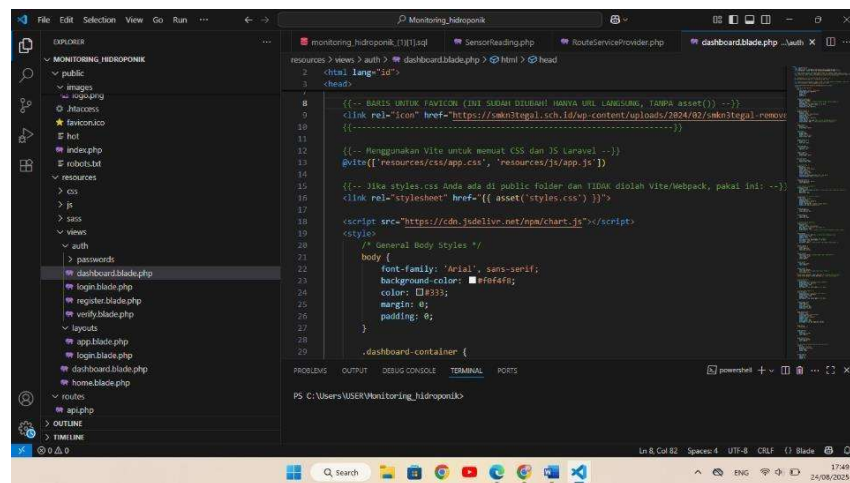
2.2.4 Perangkat Lunak (*Software*)

Perangkat lunak (*software*) adalah serangkaian instruksi atau program yang dirancang untuk menjalankan fungsi tertentu pada suatu sistem komputer. Dalam konteks pengembangan *website*, perangkat lunak mencakup berbagai *tools* dan *platform* yang digunakan untuk membangun, mengelola, dan menjalankan aplikasi web.

Perangkat lunak tidak memiliki bentuk fisik, melainkan terdiri dari kode-kode program yang dijalankan oleh perangkat keras. *Website* sebagai produk perangkat lunak bekerja dengan menggabungkan berbagai komponen seperti *frontend* (antarmuka pengguna), *backend* (logika bisnis), serta *database* (penyimpanan data) (Sommerville, 2016) [14].

2.2.5 *PHP*

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang sering digunakan dalam pengembangan aplikasi web dinamis. Keunggulan *PHP* terletak pada kemudahan integrasinya dengan berbagai sistem basis data, seperti *MySQL* dan *PostgreSQL*, serta kompatibilitas yang tinggi dengan server web seperti *Apache*. Selain itu, *PHP* juga mendukung berbagai *framework* modern seperti *Laravel* dan *CodeIgniter* yang mempercepat proses pengembangan aplikasi web (Welling & Thomson, 2003) [15].

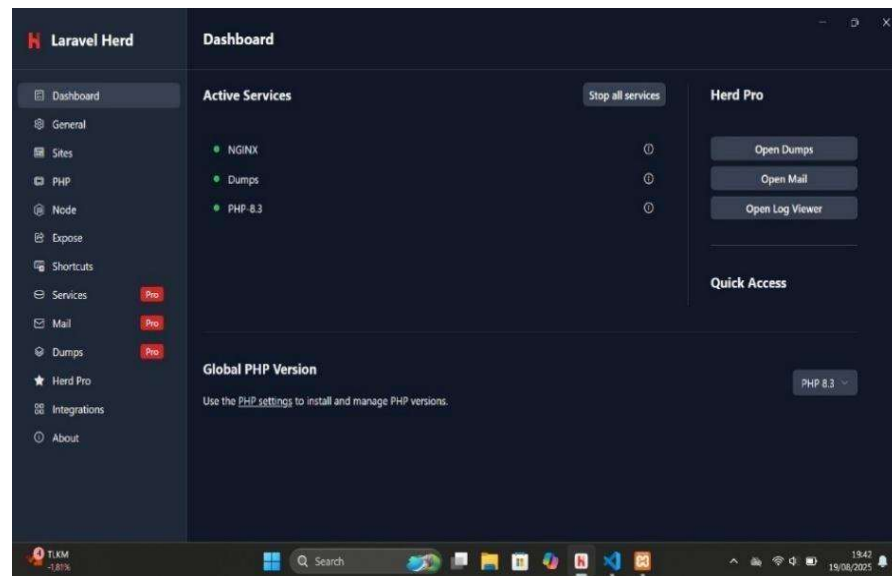


Gambar 2. 2 Php My Admin

2.2.6 *Laravel*

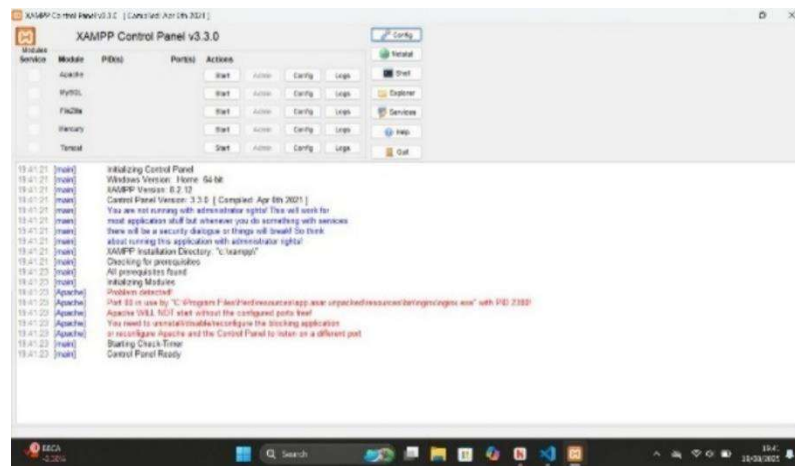
Laravel adalah *framework PHP open-source* yang mengadopsi pola arsitektur MVC (*Model-View-Controller*) untuk mempermudah pengembangan aplikasi web. *Laravel* menyediakan berbagai fitur seperti *routing*, *authentication*, *templating*, serta *ORM (Eloquent)* yang memudahkan *developer* dalam membangun sistem yang aman, cepat, dan terstruktur. *Framework* ini juga mendukung pengembangan berbasis modular dan berorientasi objek, sehingga cocok digunakan untuk aplikasi skala menengah hingga besar (Syafitri, A., Angraeni, A., & Wibowo, A. (2025).

Dalam proyek ini, *Laravel* digunakan untuk membangun sistem *monitoring hidroponik terintegrasi website* agar mudah dikembangkan dan dikelola secara berkelanjutan[16].

Gambar 2. 3 *Laravel*

2.2.7 *XAMPP*

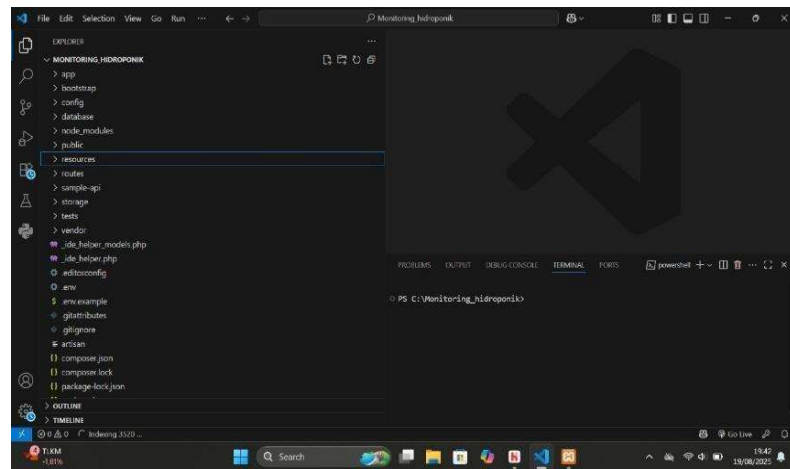
XAMPP adalah paket perangkat lunak *open-source* yang berfungsi sebagai local web server untuk kebutuhan pengembangan dan pengujian aplikasi berbasis web. *XAMPP* dikembangkan oleh *Apache Friends* dan terdiri dari beberapa komponen utama yaitu Apache (web server), MySQL (basis data), PHP, dan *Perl*, yang dikemas dalam satu paket instalasi sehingga memudahkan pengguna dalam membangun lingkungan server lokal tanpa konfigurasi yang kompleks (Nugroho, 2021) [17].



Gambar 2. 4 Xampp

2.2.8 Visual Studio Code

Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. Teks editor ini secara langsung mendukung bahasa pemrograman *JavaScript*, *TypeScript*, dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang melalui *marketplace Visual Studio Code* seperti *C++*, *C#*, *Python*, *Go*, *Java*, dan *PHP* (*Microsoft*, 2023; Nugroho, 2021) [18].



Gambar 2. 5 *Visual Studio Code*

2.2.9 UML (Unified Modeling Language)

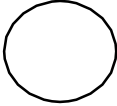


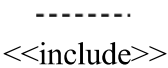

Unified Modeling Language (UML) adalah standar pemodelan visual yang digunakan untuk mendesain dan mendokumentasikan sistem perangkat lunak. UML digunakan untuk membuat Diagram seperti *Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, dan *Activity Diagram*, yang membantu dalam memahami struktur dan alur kerja sistem sebelum pengembangan dilakukan. Penggunaan UML dalam penelitian ini bertujuan untuk menyusun perencanaan sistem secara lebih sistematis dan mendalam sebelum tahap implementasi. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem yaitu *Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, dan *Activity Diagram* (Nugroho, 2010) [19].

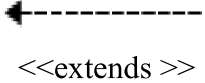
1. *Use Case Diagram*

Use Case Diagram digunakan untuk menggambarkan interaksi antara aktor (pengguna sistem) dengan sistem itu sendiri.

Diagram ini menampilkan fungsi-fungsi utama (*use case*) yang tersedia dalam sistem dan siapa saja yang dapat mengaksesnya. Dalam sistem *monitoring* hidroponik terintegrasi *website* ini, aktor seperti pengguna, admin dan ESP32 akan memiliki peran masing-masing dalam menjalankan modul sistem yang saling terintegrasi.

Tabel 2. 1 Simbol *Usecase*





No	Simbol	Keterangan
1.		<i>Use case</i> merupakan deskripsi fungsional yang telah disediakan oleh sistem sebagai entitas yang menghasilkan hasil yang terukur untuk suatu actor.
2.		<i>Actor</i> merupakan himpunan peran untuk berinteraksi dengan <i>use case</i>
3.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek yang lain
4.		<i>Include</i> merupakan gambaran jika <i>Use case</i> dipanggil oleh <i>Use case</i> lain.
5.		<i>Dependecy</i> merupakan garis panah yang menunjukkan jika <i>actor</i> berinteraksi secara pasif

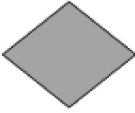
6.		<i>Extend</i> merupakan gambar jika memperluas <i>Use case</i> target.
----	---	--

2. Activity Diagram

Activity Diagram memodelkan alur aktivitas atau proses bisnis yang terjadi dalam sistem. Diagram ini menunjukkan bagaimana suatu proses dimulai, bagaimana keputusan dibuat, dan bagaimana alur kerja berpindah dari satu aktivitas ke aktivitas lainnya.

Tabel 2. 2 Simbol *Activity Diagram*

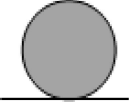
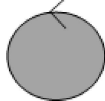
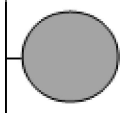
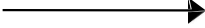
No	Simbol	Keterangan
1.		<i>End point</i> atau <i>final Node</i> merupakan gambaran akhir dari suatu aktivitas
2.		<i>Start Point</i> merupakan awal dari suatu aktivitas yang peletakannya pada pojok kiri atas
3.		<i>Fork</i> atau <i>join</i> digunakan untuk memarallelkan suatu kegiatan atau penggabungan 2 kegiatan parallel menjadi satu
4.		<i>Activity</i> merupakan gambaran dari suatu proses

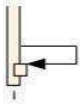


5.		<i>Decision</i> merupakan pilihan pengambilan suatu keputusan <i>false</i> or <i>true</i> .
----	---	---

3. Sequence Diagram

Sequence Diagram digunakan untuk memodelkan komunikasi antar objek dalam sistem berdasarkan urutan waktu. Diagram ini memperlihatkan bagaimana pesan dikirim dari satu objek ke objek lain dalam suatu proses.

Tabel 2. 3 Simbol *Sequence Diagram*

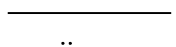
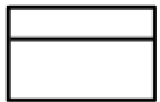
No	Simbol	Keterangan
1.		<i>Entity Class</i> merupakan bagian sistem yang membentuk deskripsi awal sistem dan berisi kumpulan kelas dalam bentuk entitas yang mendasari untuk membuat <i>Database</i>
2.		<i>Control Class</i> merupakan gambaran penghubung antara <i>Boundary class</i> dengan suatu <i>table</i>
3.		<i>Boundary Class</i> merupakan gambaran dari penggambaran <i>table</i>
4.		Pesan atau <i>message</i> menunjukkan pengiriman pesan antar <i>class</i>

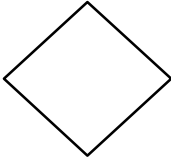

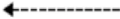


No	Simbol	Keterangan
5.		<i>Self message</i> menunjukkan pengiriman suatu pesan yang akan dikirim ke objek itu sendiri.
6.		<i>Activation</i> menggambarkan suatu objek yang melakukan sebuah aksi/eksekusi operasi
7.		<i>Lifeline</i> garis titik yang terhubung ke objek disepanjang garis <i>lifeline</i> memiliki aktivitas

4. *Class Diagram*

Class Diagram menjelaskan struktur statis sistem, termasuk kelas, atribut, metode, dan hubungan antar kelas. Diagram ini menjadi acuan utama dalam implementasi kode program di Java, karena setiap *class* dalam Diagram akan direpresentasikan dalam bentuk kode yang mengatur fungsionalitas dan alur data di dalam sistem.

Tabel 2. 4 Simbol *Class Diagram*

No	Simbol	Keterangan
1.		<i>Generallization</i> , merupakan dimana objek <i>descendant</i> membagikan perilaku dan struktur data objek induknya
2.		<i>Class</i> , adalah kumpulan objek yang saling berbagi.

No	Simbol	Keterangan
3.		<i>N-ary Association</i> , digunakan untuk <i>asosiasi</i> terhubung dengan objek lainnya.
4.		<i>Collaboration</i> merupakan deskripsi urutan aksi yang ditampilkan suatu sistem yang memiliki konsekuensi terukur bagi <i>actor</i>
5.		Merupakan operasi yang valid dilakukan oleh suatu objek
6.		Merupakan garis panah yang menunjukkan jika actor berinteraksi secara pasif
7.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek

2.2.10 Pengujian Sistem *Black box* dan *White box*

Pengujian sistem merupakan tahap penting dalam proses pengembangan perangkat lunak untuk memastikan bahwa sistem yang dibangun telah berfungsi sesuai dengan kebutuhan dan spesifikasi

yang telah ditentukan. Menurut Pressman (2018), pengujian sistem bertujuan untuk menemukan kesalahan sebelum sistem diimplementasikan kepada pengguna, serta memastikan kualitas perangkat lunak dari sisi fungsionalitas dan keandalan [20].

Dalam penelitian ini, metode pengujian yang digunakan terdiri dari dua pendekatan utama, yaitu pengujian *Black Box* (Kotak Hitam) dan pengujian *White Box* (Kotak Putih). Kedua metode ini memiliki tujuan yang berbeda namun saling melengkapi dalam menjamin kualitas sistem *monitoring* hidroponik berbasis *website*.

5. Pengujian *Black Box*

Pengujian *Black Box* adalah metode pengujian perangkat lunak yang berfokus pada fungsi sistem tanpa memperhatikan struktur internal kode program. Penguji hanya melihat hasil keluaran berdasarkan input yang diberikan, apakah sudah sesuai dengan kebutuhan fungsional sistem. Menurut Sommerville (2016), pengujian *Black Box* digunakan untuk memvalidasi fungsi sistem berdasarkan spesifikasi yang telah ditetapkan tanpa mengetahui bagaimana proses logika internal berjalan [21].

Dalam konteks sistem *monitoring* hidroponik terintegrasi *website*, pengujian *Black Box* dilakukan dengan cara memberikan berbagai skenario input seperti data sensor pH, TDS, dan suhu, serta aksi pengguna pada tampilan *website* (misalnya tombol

login, refresh data, dan kontrol pompa). Hasil yang diharapkan adalah keluaran sesuai rancangan sistem, seperti data tampil pada *dashboard* secara *real-time* atau status pompa otomatis berubah ketika nilai pH di luar batas ideal.

Tabel 2. 5 Skenario Pengujian *Black Box*

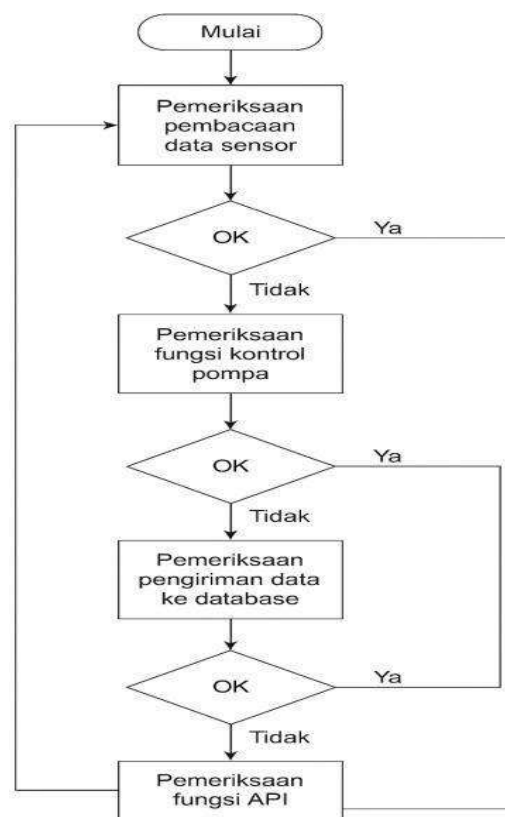
No	Modul yang diuji	Input	Proses	Output yang diharapkan	Hasil uji
1.	<i>Login user</i>	<i>User name & password</i>	Validasi akun di <i>database</i>	Masuk ke halaman <i>dashboard</i>	Berhasil
2.	<i>Monitoring sensor</i>	Pembacaan data pH, TDS, suhu	Sistem menampilkan data ke web	Data tampil <i>real-time</i> di <i>dashboard</i>	Berrhasil
3.	Kontrol Pompa	Nilai pH < 5,5 atau > 6,5	Sistem menyalakan pompa pH <i>up/down</i>	pH kembali ke kondisi ideal	Berhasil

6. Pengujian *White Box*

Pengujian *White Box* merupakan metode pengujian perangkat lunak yang dilakukan dengan melihat struktur internal sistem, termasuk alur logika, kondisi, dan proses yang terjadi di dalam kode program. Menurut Myers, Sandler, dan Badgett (2011), *white box testing* bertujuan untuk memastikan bahwa setiap jalur logika, perulangan, dan kondisi dalam program telah berjalan sesuai dengan yang diharapkan [22].

Dalam penelitian ini, pengujian *white box* dilakukan pada bagian-bagian program yang berkaitan langsung dengan logika utama sistem *monitoring* hidroponik. Pengujian dimulai dari proses pembacaan data sensor pH, TDS, dan suhu untuk memastikan bahwa setiap sensor mampu mengirimkan data dengan akurat dan konsisten ke sistem tanpa terjadi kesalahan logika dalam pembacaan nilai. Selanjutnya, pengujian dilakukan pada fungsi kontrol otomatis pompa *pH Up* dan *pH Down*, guna memverifikasi bahwa algoritma pengendalian bekerja dengan benar sesuai kondisi nilai pH yang terdeteksi, serta memastikan pompa aktif dan nonaktif pada waktu yang tepat. Tahap berikutnya adalah pengujian proses pengiriman data dari mikrokontroler ke *database*, yang bertujuan untuk memastikan bahwa data yang dikirim dari perangkat keras tersimpan dengan baik di basis data tanpa kehilangan atau duplikasi. Selain itu,

dilakukan pula pengujian pada fungsi API yang menghubungkan mikrokontroler dengan *website Laravel*, untuk menjamin bahwa komunikasi data antara sisi perangkat keras dan antarmuka web berjalan stabil, aman, dan dapat menampilkan informasi secara *real-time* pada *dashboard* pengguna.



Gambar 2. 6 Pengujian *White Box*

Tabel 2. 6 Hasil Pengujian *White Box*

No	Fungsi yang Diuji	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil Uji	Keterangan
1.	Fungsi pembacaan sensor pH	Memeriksa apakah sensor membaca nilai pH dengan benar dan mengirim ke server	Nilai pH terbaca dan tampil di <i>database</i>	Berhasil	Fungsi berjalan normal
2.	Fungsi kontrol pompa otomatis	Menguji logika pengaktifan pompa berdasarkan kondisi pH	Pompa aktif sesuai kondisi (<i>Up/Down</i>)	Berhasil	Logika kontrol benar
3.	Fungsi koneksi <i>database</i>	Mengecek konektivitas antara ESP32 dan <i>database</i> MySQL	Data tersimpan dengan benar tanpa error	Berhasil	Sinkronisasi data baik
4.	Fungsi API	Menguji pengiriman data JSON dari mikrokontroler	Data diterima oleh <i>Laravel</i> tanpa kehilangan paket	Berhasil	Integrasi stabil

		ler ke <i>website</i>			
5.	Fungsi tampi lan data	Mengecek alur query data dari database ke tampilan web	Data tampil sesuai pembacaan sensor	Berha sil	Alur logika benar