

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian Putu Denanta Bayuguna dkk. Hidroponik yang diteliti pada tahun 2020 sangat cocok ditanam di perkotaan karena dapat ditanam tanpa memerlukan lahan yang luas. Salah satu teknik yang digunakan dalam hidroponik adalah sistem aeroponik. Namun budidaya hidroponik juga mempunyai kelemahan. Hal ini mencakup perlunya lingkungan yang sangat terkendali untuk menjaga kesehatan tanaman dan menghindari berkurangnya hasil panen. Para petani sibuk di perkotaan, membuat sistem penanaman hidropin tradisional sulit diterapkan. Oleh karena itu, diperlukan solusi untuk mengontrol kondisi tanaman secara otomatis sesuai kebutuhan. Penelitian ini memberikan alternatif solusi menanam tanaman hidroponik dengan menggunakan teknologi berbasis IoT yang membantu dalam pengendalian dan pemantauan tanaman secara otomatis, serta membantu masyarakat luas dalam menanam tanaman hidroponik. Sistem yang dibuat mengintegrasikan perangkat IoT dan *smartphone Android* dan bertindak sebagai sistem kontrol dan pemantauan [3].

Sebuah studi oleh Shafira Rana Rafidah dkk. Berdasarkan hasil survei BPS (Badan Pusat Statistik) tahun 2016, tingkat konsumsi sayur masyarakat Indonesia pada tahun 2020 tergolong tinggi yaitu sebesar 97,29%. Kelimpahan dan kualitas sayuran di Indonesia didukung oleh perkembangan teknologi

sebagai berikut. Banyaknya teknik budidaya yang lebih praktis, efisien, sederhana dan produktif dibandingkan dengan teknik tradisional yang umum digunakan. Teknologi yang populer dan tersebar luas saat ini adalah teknik menanam dengan air yang disebut hidroponik. Budidaya dengan teknik hidroponik tidak hanya untuk konsumsi rumah tangga saja, namun juga menjadi ladang usaha yang sangat menguntungkan. Sistem hidroponik fokus pada penyediaan nutrisi bagi tanaman melalui akarnya, yang menyerap air nutrisi. Faktor-faktor yang perlu diperhatikan dalam hidroponik antara lain kandungan oksigen terlarut, konsentrasi nutrisi terlarut (EC), sinar matahari, keasaman (pH) larutan berair, dan suhu. Pemantauan terhadap semua faktor tersebut harus dilakukan secara berkala untuk memastikan tanaman mendapat nutrisi yang tepat [4].

Diterbitkan oleh Yuga Hadfridar Putra dkk. Sebuah penelitian yang dilakukan pada tahun 2018 umumnya menggunakan air sebagai metode budidaya hidroponik. Kondisi air yang perlu diperhatikan adalah ketersediaan air, oksigen, unsur hara, dan keasaman (pH). Selain itu, suhu dan kelembapan lingkungan harus dijaga pada kondisi yang sesuai untuk tanaman. Pengendalian nutrisi, suhu air, volume air nutrisi, suhu lingkungan, pH, dan kelembapan pada sistem hidroponik masih dilakukan secara manual atau menggunakan cara tradisional. Oleh karena itu, dibutuhkan banyak waktu dan tenaga untuk memeriksa dan menyesuaikan sendiri kondisi air pada sistem hidroponik anda. Pada penelitian ini, guna memudahkan pengelolaan air nutrisi pada budidaya hidroponik, kami mengembangkan sistem yang dapat

memantau dan mengontrol nutrisi, suhu, dan ketinggian air pada budidaya hidroponik dalam sebuah *website*. Di era teknologi dan internet yang berkembang pesat, masyarakat memilih menggunakan aplikasi *website* karena aplikasi berbasis *website* dapat digunakan pada *platform* atau sistem operasi apa pun tanpa perlu instalasi ulang atau memenuhi spesifikasi yang diperlukan kamu membuat. Hanya ketersediaan *browser* dan akses internet yang tinggi [1].

Studi tahun 2018 yang dilakukan Endi Sailul Haq dan Devit Suwardiyanto menemukan bahwa beberapa hal dalam sistem ini masih berjalan seperti biasa hingga saat ini. Artinya, memperhatikan pH air dalam campuran air dan nutrisi, serta menjaga kestabilan suhu dan kelembapan di dalam sistem rumah kaca. Permasalahan yang muncul adalah membutuhkan banyak tenaga dan waktu untuk terus menerus menjaga dan merawat tanaman tersebut. Oleh karena itu, pengelola tidak bisa meninggalkan kegiatan tersebut, yang dapat mengakibatkan metode budidaya hidroponik menjadi tidak efektif dan efisien. Pada penelitian ini, kami membuat sistem semi otomatis untuk pengelolaan tanaman hidroponik di rumah kaca. Sistem yang dibuat dapat memantau suhu dan kelembapan di dalam rumah kaca, serta memantau dan mengontrol pH campuran air dan unsur hara yang disuplai ke tanaman. Pastikan lingkungan stabil yang ideal untuk tanaman. Menggunakan sensor pH dan DHT11 [5].

2.2 Landasan Teori

2.2.1 Sistem Monitoring

Sistem monitoring merupakan suatu proses untuk mengumpulkan data dari berbagai sumber daya. Biasanya data yang dikumpulkan merupakan data yang *realtime*. Secara umum tujuan monitoring adalah untuk mendapatkan data – data atau pandangan agar diperoleh umpan balik bagi kebutuhan tertentu. Secara garis besar tahapan dalam sebuah sistem monitoring terbagi ke dalam tiga proses besar yaitu proses di dalam pengumpulan data monitoring, proses di dalam analisis data monitoring dan proses di dalam menampilkan data hasil monitoring [6].

2.2.2 Framework CodeIgniter

CodeIgniter adalah sebuah *web application framework* yang digunakan untuk membangun aplikasi PHP dinamis yang dibangun menggunakan konsep *Model View Controller development pattern*. CodeIgniter menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan dan termasuk *framework* tercepat dibandingkan dengan *framework* lainnya [7].



Gambar 2. 1 Gambar Logo CodeIgniter

(Sumber: <https://images.app.goo.gl/GJXsTkS9GHZGqYzQ9>)

2.2.3 PHP

Hypertext Preprocessor atau yang disingkat PHP, PHP merupakan script untuk pemrograman *script web server-side*, *script* yang membuat dokumen HTML secara *on the fly*, maksudnya dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. PHP/FI merupakan nama awal dari PHP. PHP adalah *Personal Home Page*, FI adalah *Form Interface*. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam *browser web*. *Software* ini disebar dan dilisensikan sebagai perangkat lunak *Open Source* [8].



Gambar 2. 2 Gambar Logo php

(Sumber: <https://www.php.net/download-logos.php>)

2.2.4 MySQL

Structured Query Language yang sering dikenal dengan MySQL, MySQL merupakan perangkat lunak (*software*) gratis dibawah lisensi GPL (GNU *General Public License*). MySQL sebagai sebuah implementasi dari sistem manajemen basis data relasional (RDBMS-*Relational DataBase Management System*) didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dapat secara bebas menggunakan MySQL, namun dengan batasan tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL merupakan turunan konsep utama dalam basis data, yaitu SQL. SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan input data, yang memungkinkan pengoperasian data dikerjakan dengan mudah [9].



Gambar 2. 3 Gambar Logo MySQL

(Sumber: <https://www.mysql.com/about/legal/logos.html>)

2.2.5 Visual Studio Code

Visual Studio Code merupakan 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 via *marketplace* Visual Studio Code seperti C++, C#, Python, Go, Java dan PHP [8].



Gambar 2. 4 Gambar Logo Visual Studio Code

(Sumber: https://www.researchgate.net/figure/Gambar-10-Logo-Visual-Studio-Code_fig1_369020845)

2.2.6 Database

Database merupakan kumpulan file - file yang saling berkaitan dan berinteraksi, relasi tersebut bila ditunjukkan dengan kunci dari tiap - tiap file yang ada. Satu database menunjukkan suatu kumpulan data yang dipakai dalam suatu lingkup perusahaan, instansi. Pengolahan database merupakan suatu cara yang dilakukan terhadap file-file yang berada di suatu instansi yang mana file tersebut dapat disusun, diurut, diambil sewaktu-waktu serta dapat ditampilkan dalam bentuk suatu laporan sehingga dapat mengolah file-file yang berisikan informasi tersebut secara rapi [9].

2.2.7 Java Script

Sekumpulan script yang fungsinya dijalankan pada dokumen *HTML*. *JavaScript* adalah bahasa skrip yang awalnya dikembangkan khusus untuk penggunaan di *website*. Bahasa ini digunakan sebagai bahasa pemrograman untuk memberikan fungsi tambahan pada halaman *website* menggunakan eksekusi perintah di sisi pengguna, yaitu di dalam browser, bukan di server *website*.



Gambar 2. 5 Gambar Logo Java Script

(Sumber: www.javilazaro.es/programacion/principios-javascript)

2.2.8 UML (Unified Modeling Language)

Unified Modelling Language atau yang disingkat UML, merupakan suatu metode dalam teknik RPL (Rekayasa Perangkat Lunak) yang berfungsi untuk menggambarkan cara kerja sistem, fungsi, alur, tujuan dan juga mekanisme kontrol sistem. UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem [10].

Terdapat empat model UML yang paling sering digunakan untuk menggambarkan suatu desain sistem Usecase diagram, Class diagram, *Behavioral State machine diagram*, dan juga *Sequence diagram*. Teknik-teknik pemodelan *Unified Modeling Language* ini disebut juga dengan 4

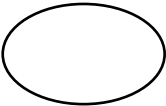


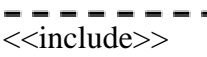
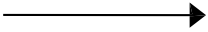
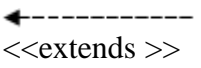
teknik dasar. Dalam proyek berorientasi objek, keempat teknik UML ini sangat mendominasi penggunaannya.

Dalam perancangan sistem terdapat UML yang sering digunakan sebagai berikut:

1. *UseCase Diagram*

Dalam pembuatan sistem informasi, *use case diagram* merupakan pemodelan untuk *behavior* atau kelakuan sistem informasi. Diagram ini juga bersifat statis. Untuk simbol *UseCase Diagram* disajikan pada Tabel 2.1.



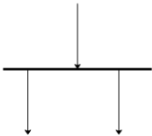

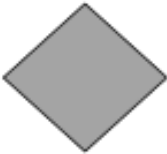
Tabel 2. 1 *Use Case Diagram*

No	Simbol	Keterangan
1.		<i>Use case</i> merupakan deskripsi fungsional yang telah disediakan oleh sistem sebagai enitas yangt menghasilkan hasil yang terukur untuk suatu <i>actor</i> .
2.		<i>Actor</i> merupakan himpunan peran untuk berinteraksi dengan <i>usecase</i>
3.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek yang lain
4.		<i>Include</i> merupakan gambaran jika <i>usecase</i> dipanggil oleh <i>usecase</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>usecase</i> target.

2. Activity Diagram

Activity Diagram atau Diagram Aktivitas merupakan diagram yang bersifat statis, yang menggambarkan aktivitas dari suatu sistem bisnis. Untuk simbol dari diagram aktivitas disajikan pada Tabel 2.2.

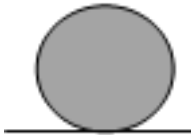
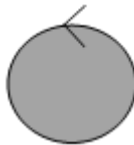
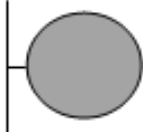
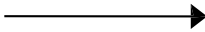
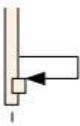


Tabel 2. 2 *Activity Digram*

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 or true</i> .

3. *Sequence Diagram*

Sequence Diagram atau Diagram Urutan mendeskripsikan diagram interaksi yang mengirimkan pesan dan diterima antar objek. Untuk simbol diagram urutan disajikan pada Tabel 2.3

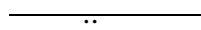
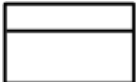
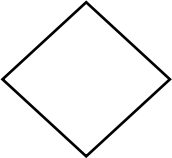




Tabel 2. 3 *Sequence Diagram*

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 database
2.		<i>ControlClass</i> merupakan gambaran penghubung antara <i>Boundaryclass</i> dengan suatu table
3.		<i>Boundary Class</i> merupakan gambaran dari penggambaran table
4.		Pesen atau <i>message</i> menunjukkan pengiriman pesen antar <i>class</i>
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 lifeline memiliki aktivitas

4. *Class Diagram*

Diagram Class merupakan diagram yang bersifat statis, dalam diagram ini memperlihatkan himpunan kelas, antar muka, serta relasi. Simbol dari *Class Diagram* disajikan pada Tabel 2.4.

Tabel 2. 4 *Class Diagram*

No	Simbol	Keterangan
1.		<i>Generallization</i> , merupakan dimana objek <i>descendent</i> membagikan perilaku dan struktur data objek induknya
2.		<i>Class</i> , adalah kumpulan objek yang saling berbagi.
3.		<i>Nary Association</i> , digunakan untuk asosiasi terhindar dengan objek lainnya.
4.		<i>Collaboration</i> merupakan deskripsi urutan aksi yang ditampilkan suatu sistem yang memiliki konsekuensi terukur bagi actor
5.		Merupakan operasi yang valid dilakukan oleh suatu objek
6.		Merupakan garis penah yang menunjukkan jika actor berinteraksi secara pasif
7.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek yang lain