

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Andhika Bayu (2022) dalam artikel yang berjudul "Sistem Pemantau Kualitas Air Kolam Ikan Koi Berbasis IoT", penelitian ini merancang suatu sistem yang dapat digunakan sebagai memonitor dan kontrol kualitas air kolam Ikan Koi dengan tujuan dapat membantu pemeliharaan Ikan Koi meningkatkan pertumbuhan pada Ikan Koi itu sendiri. Pada penelitian ini terdapat empat parameter penting yang dijadikan sebagai acuan untuk mengamati kualitas air Ikan Koi, parameter suhu ideal Ikan Koi adalah 25-30°C, parameter keasaman ideal Ikan Koi adalah berkisar antara 7,0-8,7 pH, parameter minimum oksigen terlarut (*Dissolved Oxygen*) Ikan Koi adalah 5 ppm, sedangkan batas maksimum parameter kekeruhan atau *turbidity* Ikan Koi adalah 2000 NTU. Pada penelitian ini didapatkan nilai akurasi kinerja sistem 100% dengan sepuluh kali percobaan [5].

Menurut Imam Erlangga Prasetya et al. (2022) dalam artikel yang berjudul "Penerapan IoT untuk Sistem Monitoring Air dan *Controlling* pada Kolam Ikan Gurami Berbasis *Website*", penelitian ini bertujuan untuk melakukan monitoring kolam ikan terhadap suhu air, ph air, ketinggian air, pemberian makan ikan, dan pengurasan air kolam. Untuk monitoringnya sendiri menggunakan dua media yaitu *website* dan aplikasi *Whatsapp*.

Pengujian pompa apabila pada halaman *website* pompa dalam kondisi *on* maka pompa akan menyala dan apabila dalam kondisi *off* pompa akan mati [6].

Menurut Nur Ilham et al. (2023), dalam artikel yang berjudul “Rancang Bangun System Monitoring dan *Controlling* Alat Pemberi Pakan Ikan dan Pengganti Air Otomatis”, dalam peraturan Menteri Kesehatan, Menurut Keputusan No.416/MENKES/PER/IX/1990 tentang permintaan dan pengendalian air, batas maksimum kekeruhan air bersih adalah 25 NTU (*Nephelometric Turbidity Unit*). Sensor *turbidity* mendeteksi kekeruhan air untuk menentukan kapan harus mengganti air. Jika data pada sensor *turbidity* mendeteksi kekeruhan air lebih dari 25 NTU, *relay* 1 terbuka dan *relay* 2 tertutup sehingga pompa pengurasan air di akuarium beroperasi [7].

Serta penelitian yang dilakukan oleh Yohanes Anton Nugroho et al. (2020) dalam artikel yang berjudul “Perancangan *Mobile Application* Alat Pantau Kualitas Air Kolam Berbasis *Internet of Things*”, penelitian ini menerapkan pemantauan kualitas air dari kolam beton yang dirancang berbasis teknologi IoT (*Internet of Things*). Peralatan pemantau tersebut tidak akan berfungsi dengan optimal tanpa adanya aplikasi yang berfungsi untuk menerima data pantauan dan selanjutnya melakukan tindakan. Alat pemantau kualitas air kolam dihubungkan ke *cloud* menggunakan koneksi jaringan GSM. Data yang terekam selanjutnya ditampilkan pada aplikasi pemantau kualitas air yang dirancang menggunakan *operating system* Android. Desain aplikasi dikembangkan menggunakan pendekatan *User Centered Design*,

dimana proses perancangannya mempertimbangkan sejumlah variabel, yaitu kemudahan penggunaan (*easy for use*), kejelasan penyampaian informasi, pemenuhan kebutuhan dan tampilan [8].

2.2 Landasan Teori

2.2.1 *Website*

Sebuah *website* yaitu sekumpulan informasi yang dapat diakses melalui jaringan internet, yang memungkinkan siapa saja yang terhubung ke jaringan internet untuk mengunjunginya. *Website* terdiri dari berbagai elemen seperti teks, gambar, suara, dan animasi yang membuatnya menjadi media informasi yang lebih menarik. Halaman *website* dibuat menggunakan bahasa standar yang dikenal sebagai HTML. *Web browser* menerjemahkan skrip HTML ini sebagai informasi yang dapat ditampilkan dalam bentuk yang dapat dibaca oleh semua orang [9]. Arsitektur *website* merujuk pada struktur dasar pada *website* tersebut, yang terdiri dari elemen-elemen *frontend* dan *backend*.

Keamanan dan privasi data merupakan aspek penting dari sistem ini. Data kualitas air dan aktivitas kontrol dienkripsi untuk mencegah akses yang tidak sah, fitur tambahan seperti visualisasi data dalam bentuk grafik memudahkan pengguna untuk memahami perubahan parameter kualitas air dari waktu ke waktu. Dengan kombinasi fungsionalitas, kemudahan akses dan keamanan, *website*

ini diharapkan dapat menjadi solusi yang efektif dan efisien dalam pengelolaan kolam Ikan Koi.

2.2.2 Visual Studio Code

Aplikasi *editor code open source* Visual Studio Code oleh Microsoft untuk sistem operasi Windows, Linux, dan MacOS [10]. Aplikasi mendukung berbagai jenis pemrograman, seperti C++, C#, JavaScript, Python, PHP, dan lainnya.

2.2.3 Arduino IDE

Arduino IDE (*Intergrated Development Enviroment*) adalah aplikasi yang digunakan untuk menyusun *sketch* pemrograman yang sebagai alat untuk pemograman pada *board* yang ingin diprogram. JavaScript adalah bahasa pemrograman yang digunakan dalam Arduino IDE, dan dilengkapi dengan *library* C/C++ untuk mempermudah *input* dan *output* [11].

2.2.4 Laragon

Laragon merupakan perangkat lunak gratis yang mendukung berbagai sistem operasi dan dapat digunakan sebagai server diri sendiri atau *localhost* [12].

2.2.5 CodeIgniter 4

CodeIgniter merupakan kerangka kerja pengembangan aplikasi PHP dengan model MVC (*Model, View, Controller*). Dalam CodeIgniter, *browser* berinteraksi melalui *Controller*. *Controller* akan menerima dan membalas semua permintaan dari *browser*,

ketika *Controller* membutuhkan data, maka *Controller* akan meminta ke *Model*. Sedangkan untuk tampilan pada *user* ditangani oleh *View* [13].

2.2.6 JavaScript

JavaScript adalah bahasa pemrograman untuk memberikan kemampuan tambahan ke dalam bahasa pemrograman HTML atau juga digunakan untuk menjelaskan tampilan dalam halaman *website* [14].

2.2.7 MySQL

MySQL merupakan salah satu jenis *database* paling umum digunakan untuk membuat aplikasi *website* yang dinamis. MySQL adalah *software* sistem manajemen DBMS (*Database Management System*) yang *multiuser* karena sifatnya yang *open source* dan kapasitasnya yang luar biasa, Akibatnya, MySQL menjadi *database* [15].

2.2.8 PhpMyAdmin

PhpMyAdmin mendukung berbagai operasi MySQL, seperti mengelola *database*, tabel-tabel, bidang (*fields*), relasi (*relation*), *index*, pengguna (*users*), perijinan (*permissions*), dan sebagainya [16].

2.2.9 HyperText Markup Language (HTML)

HTML berfungsi sebagai kerangka dari suatu halaman web. Elemen-elemen dalam HTML *ditulis* menggunakan *tag-tag* yang mencakup konten tertentu. Salah satu fungsi HTML adalah menyusun

struktur halaman *website*, menempatkan setiap komponen *layout* yang diinginkan [17].

2.2.10 *Cascading Style Sheet (CSS)*

CSS yaitu bahasa untuk mengatur tampilan dan gaya dari sebuah dokumen web yang ditulis dalam HTML. CSS digunakan untuk memisahkan konteks utama dengan tampilan dokumen seperti *layout*, warna dan *font* [18].

2.2.11 UML (*Unified Modeling Language*)

UML yaitu salah satu alat untuk merancang *software* yang berbasis *object-oriented* [19]. Perancangan UML bertujuan untuk memvisualisasikan dari proses analisis dan desain berorientasi objek. Ada beberapa diagram UML yang biasa digunakan dalam pengembangan sistem, yaitu:

1. *Use Case Diagram*: Menggambarkan dari fungsionalitas yang dibutuhkan menurut sebuah sistem, dan merepresentasikan sebuah hubungan antara *actor* dan sistem.

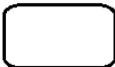
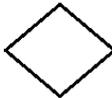
Tabel 2. 1 Simbol *Use Case Diagram*

Simbol	Keterangan
Actor 	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
Use Case 	Abstraksi dan interaksi antara sistem dan <i>actor</i> .
Association 	Abstraksi dari penghubung antara <i>actor</i> dan <i>use case</i> .

Simbol	Keterangan
Generalisasi ----->	Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i> .
<i>include</i> ←<<include>>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
<i>extend</i> →<<extend>>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

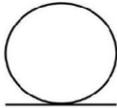
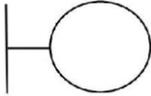
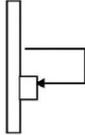
2. *Activity Diagram*: Menggambarkan bagaimana suatu sistem atau bagian dari dirinya bertindak secara terus menerus melalui aliran kontrol yang berbentuk antara tindakan yang dilakukan sistem.

Tabel 2. 2 Simbol *Activity Diagram*

Simbol	Keterangan
<i>Start</i> 	Sebuah diagram aktivitas memiliki sebuah status awal.
<i>Action</i> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
<i>Decision</i> 	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
<i>Join</i> 	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
<i>End</i> 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

3. *Sequence Diagram*: Biasanya digunakan untuk menunjukkan skenario atau tindakan yang diambil sebagai tanggapan terhadap sebuah peristiwa tertentu untuk mencapai *output* tertentu.

Tabel 2. 3 Simbol *Sequence Diagram*

Simbol	Keterangan
<p><i>Entity Class</i></p> 	Gambaran sistem sebagai landasan dalam menyusun basis data.
<p><i>Boundary Class</i></p> 	Menangani komunikasi antar lingkungan sistem.
<p><i>Control Class</i></p> 	Bertanggung jawab terhadap <i>class-class</i> terhadap objek yang berisi logika.
<p><i>Recursive</i></p> 	Pesan untuk dirinya.
<p><i>Activation</i></p> 	Mewakili proses durasi aktivasi sebuah operasi.
<p><i>Life Line</i></p> 	Komponen yang digambarkan garis putus terhubung dengan objek.