

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian berjudul "Sistem *Monitoring* Parameter Fisik Air Kolam Ikan menggunakan Jaringan Sensor Nirkabel berbasis Protokol LoRa" (2019) menggunakan jaringan sensor nirkabel untuk memantau kualitas kolam budidaya. Sistem terdiri dari tiga bagian: kumpulan perangkat node sensor, perangkat *gateway*, dan data center. Perangkat node sensor mengukur parameter fisik air secara berkala dengan menggunakan sensornya, setiap node sensor terdiri dari tiga komponen utama: sensor, mikroprosesor, dan modul komunikasi. Selanjutnya, node sensor mengirimkan hasil pengamatan kualitas air ke perangkat *gateway*, yang kemudian mengirimkan data ke data *center* untuk diproses [8].

Studi yang dipublikasikan oleh Hardian Efendi (2020) berjudul "Perancangan Dan Implementasi Alat *Monitoring* Kelayakan Air Pada Kolam Ikan Berbasis *Internet Of Things* (IoT) Menggunakan Mikrokontroler" membahas pengembangan dan pelaksanaan alat yang menggunakan mikrokontroler untuk memantau kelayakan air di kolam ikan. Berdasarkan hasil pengujian fungsionalitas yang dilakukan pada ketiga sensor yang dibuat, dapat disimpulkan bahwa semua fungsinya berjalan sepenuhnya sesuai dengan harapan, yaitu dapat mengolah data dan mengirimkannya ke *database*. Sensor memiliki toleransi kesalahan pH

sebesar 1%, toleransi suhu sebesar 0.99 %, dan toleransi kekeruhan yang diharapkan. *Database* dan *website* sudah terintegrasi dengan baik. Semua data sensor yang diambil dari database dapat ditampilkan di *website* berdasarkan hasil pengujian *monitoring*. Pengujian pengiriman data dilakukan dengan baik[9].

Penelitian yang berjudul “Sistem pemantauan kualitas air budidaya ikan lele menggunakan teknologi IoT” (Nurul, Fahmi, 2020) menggunakan teknologi *Internet of Things* untuk membuat sistem pemantauan kualitas air kolam ikan lele. Mereka memantau pH air dan suhu di kolam ikan lele melalui *website* dan aplikasi Android. Jika mini PC dan Arduino dihubungkan ke satu daya, penelitian ini akan berhasil. dengan menggunakan serial port pada mini pc, data dari arduino akan disambungkan ke *database*. Aplikasi Android dan *website* kemudian dapat membaca data di *database* dan menghasilkan grafik dalam proses. Sistem ini akan mengumpulkan data tentang kualitas air (pH) dan suhu air. Hasil pengujian pH menunjukkan bahwa pH asam adalah 4.45, dan suhu kolam adalah 31.1 derajat Celcius. Kami akan menerapkan komunikasi dengan protokol MQTT untuk penelitian selanjutnya [10].

Penelitian yang berjudul “Rancang Bangun Sistem Otomatis Pemberian Pakan Pada Budidaya Ikan Dalam Ember (Budikdamber) Berbasis Arduino Uno Atemega 328” (Faris Nidaul Haq, Nurfiana 2023). Dalam Penelitian ini bertujuan untuk membangun sistem ketepatan pemberian pakan ikan secara otomatis berdasarkan porsi dan jadwal yang tepat pada Budidaya Ikan Dalam Ember (BUDIKDAMBER) menggunakan Arduino UNO Atemega 328

sebagai kontrol sistem, RTC DS3231 sebagai kontrol waktu dan *Relay* sebagai saklar otomatis yang dapat mengaktifkan Motor DC yang digunakan sebagai pengatur porsi pakan.[11]

Penelitian yang dilakukan oleh Supriadi, yang berjudul "Perancangan Sistem Penjadwalan dan *Monitoring* Pemberian Pakan Ikan Otomatis Berbasis *Internet Of Things*" (2020) membahas mengapa budidaya ikan menjadi salah satu bisnis yang paling diminati oleh orang-orang. Saat ini, hasilnya sangat menjanjikan. Sederhananya, pakan ikan diberikan langsung ke kolam atau tambak setiap hari. Pada saat jadwal pemberian pakan, pengguna atau pemberi pakan biasanya tidak hadir atau tidak dapat diukur dengan tepat sesuai dengan jumlah dan ukuran ikan. Akibatnya, hal yang sering terjadi adalah pengguna atau pemberi pakan tidak berada di area kolam saat jadwalnya[12].

2.2 Landasan Teori

2.2.1 Sistem Monitoring

Sistem *monitoring*, juga dikenal sebagai sistem pengawasan, adalah upaya sistematis untuk membandingkan kinerja sebenarnya dengan standar yang telah ditetapkan saat merancang sistem umpan balik informasi, untuk membandingkan kinerja aktual dengan standar yang telah ditentukan, untuk mengetahui apakah telah terjadi suatu penyimpangan, dan untuk mengambil tindakan perbaikan yang diperlukan untuk memastikan bahwa semua sumber daya yang ada di

perusahaan atau organisasi digunakan dengan cara yang paling efisien dan efektif [13].

2.2.2 PHP

Hypertext Preprocessor, juga dikenal sebagai PHP, adalah program pemrograman web *server-side* yang dapat membuat dokumen HTML secara instan. *Ini* berarti dokumen HTML yang dibuat oleh aplikasi bukan yang dibuat menggunakan editor teks atau editor HTML. Nama awal PHP adalah PHP/FI, dengan "PHP" berarti Personal Home Page dan "FI" berarti *Form Interface*. Rasmus Lerdoff yang pertama kali membuatnya. PHP awalnya dirancang untuk menerima input melalui *form* yang ditampilkan dalam *browser web*. Perangkat lunak ini didistribusikan dan dilisensikan sebagai perangkat lunak *Open Source*. [14].



Gambar 2.1 Logo *Hypertext Preprocessor*
(Sumber: <https://images.app.goo.gl/LYTFhDroeDyBBPzr7>)

2.2.3 MySQL

MySQL, juga dikenal sebagai Bahasa *Query* Terstruktur, adalah perangkat lunak gratis yang tersedia di bawah lisensi GPL (GNU

General Public License). MySQL merupakan implementasi dari sistem manajemen basis data relasional (RDBMS—*Relational DataBase Management System*). Meskipun MySQL dapat digunakan secara gratis, namun tidak diizinkan untuk dijadikan sebagai dasar produk komersial. MySQL adalah turunan dari konsep utama basis data, yaitu SQL. SQL sendiri adalah konsep pengoperasian basis data, terutama untuk input dan pemilihan data, yang memungkinkan pengelolaan data dilakukan dengan mudah dan dapat dijelaskan ulang tanpa terdeteksi oleh Turnitin.[15].



Gambar 2.2 Logo MySQL

(Sumber: <https://images.app.goo.gl/LYTFhDroeDyBBPzr7>)

2.2.4 *CodeIgniter 3*

Struktur yang dibuat oleh Rick Ellis pada tahun 2006 disebut *CodeIgniter3*. *CodeIgniter* adalah platform *open-source* yang digunakan untuk pengembangan aplikasi *web* dan *web*. Untuk pengembang yang menggunakan bahasa pemrograman PHP untuk membuat situs *web* dan aplikasi, *framework* ini dimaksudkan. Tujuan dari *CodeIgniter* ini adalah agar para developer dapat mengembangkan proyek dengan lebih cepat dan lebih mudah.

Karena para developer harus menulis kodenya sendiri dari awal, pekerjaan mereka terkadang membutuhkan waktu yang lebih lama, kata developer *CodeIgniter*. Dengan menggunakan *CodeIgniter* ini, mereka dapat menyediakan *built-in library* dengan ratusan template dan solusi untuk proyek yang biasanya dilakukan oleh para developer. Selain itu, *CodeIgniter* menawarkan user interface (UI) yang sederhana dan struktur logis yang memungkinkan akses ke *built-in library*. [16].



Gambar 2.3 Logo *CodeIgniter 3*
(Sumber: <https://images.app.goo.gl/LYTFhDroeDyBBPzr7>)

2.2.5 *Cascading Style Sheets(CSS)*

Cascading Style Sheet (CSS) adalah bahasa yang biasa digunakan untuk mengatur tampilan elemen yang ditulis dalam bahasa markup, seperti HTML. Fungsinya adalah untuk membedakan tampilan konten dari tampilan visualnya di situs *web*. Untuk tujuan yang sederhana, *World Wide Web Consortium* (W3C) membuat dan mengembangkan CSS pada tahun 1996. Di masa lalu, HTML tidak memiliki tags yang dapat memformat halaman. Setelah tag seperti `` ditambahkan ke HTML versi 3.2, developer menghadapi banyak masalah. Karena berbagai *font*,

warna *background*, dan *style* yang ada di situs *web*, menulis kembali kode memerlukan proses yang panjang dan rumit. Akibatnya, W3C membuat CSS untuk mengatasi masalah ini. HTML dan CSS sangat terkait[17]



Gambar 2.4 *Cascading Style Sheets (CSS)*

(Sumber: <https://images.app.goo.gl/LYTFhDroeDyBBPzr7>)

2.2.6 *Unified Modeling Language*

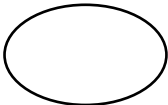


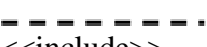
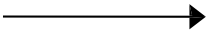
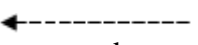
Unified Modelling Language (UML) adalah salah satu metode dalam teknik RPL (Rekayasa Perangkat Lunak). UML mendefinisikan cara kerja sistem, fungsi, alur, tujuan, dan mekanisme kontrol sistem. Sangat populer di industri, dan dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Bahasa pemodelan standar terbuka semakin populer di industri pengembangan sistem dan peranti lunak.

Terdapat empat model UML yang paling sering digunakan untuk menggambarkan suatu desain sistem *Usecase diagram*, *Class*

diagram, *Activity diagram*, dan juga *Sequence diagram*. Teknik-teknik pemodelan UML ini disebut juga dengan 4 teknik dasar. Dalam proyek berorientasi objek, keempat teknik UML ini sangat mendominasi penggunaannya.[18]





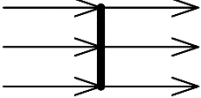
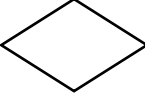
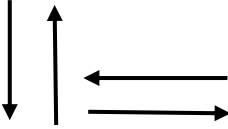
1. *Use case*: Merupakan gambaran dari fungsional yang diharapkan dari sebuah sistem dan mempresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat aktor 6 yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

Tabel 2. 1 Simbol *Use case*

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 <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.	 <<include>>	<i>Include</i> merupakan gambaran jika <i>usecase</i> dipanggil oleh <i>usecase</i> lain
5.		<i>Dependency</i> merupakan garis panah yang menunjukkan jika <i>actor</i> berinteraksi secara pasif
6.	 <<extends >>	<i>Extend</i> merupakan gambar jika memperluas <i>usecase</i> target.



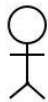
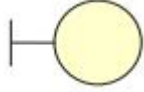




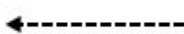
2. *Activity Diagram*: Merupakan gambaran alur dari aktivitas-aktivitas didalam sistem yang berjalan.

Tabel 2. 2 Simbol *Activity Diagram*

Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
	<i>Decision</i>	Pilihan untuk mengambil keputusan.
	<i>Fork/Join</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

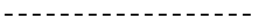
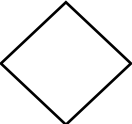




3. *Sequence Diagram*: Menggambarkan interaksi antar objek didalam dan disekitar yang berupa *message* yang digambarkan terhadap waktu.

Tabel 2. 3 Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek <i>entity</i> , antar muka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	<i>Boundary Class</i>	Menggambarkan penggambaran dari form.
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.
	<i>Control Class</i>	Menggambarkan penghubung antara <i>Boundary</i> dengan tabel.
	<i>Activation</i>	Sebagai sebuah objek yang akan melakukan sebuah aksi.
	<i>Message</i>	Mengindikasikan komunikasi antara objek dengan objek.
	<i>Self Mesagge</i>	Mengindikasikan komunikasi kembali kedalam sebuah objek itu <i>sendiri</i> .

4. *Class Diagram*: Merupakan gambaran struktur dan deskripsi dari *class*, *package* dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.

Tabel 2. 4 Simbol *Class Diagram*

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

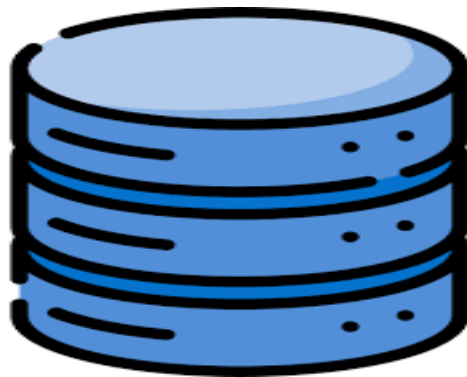
2.2.7 Java Script

Bahasa pemrograman yang digunakan oleh para programmer untuk menciptakan halaman *web* yang menarik dan interaktif adalah *JavaScript*. Fungsi-fungsi *JavaScript* dapat meningkatkan pengalaman pengguna di berbagai situs *web*, mulai dari menyesuaikan media sosial hingga menampilkan grafik animasi dan elemen

interaktif. Sebagai salah satu teknologi *web*, *JavaScript* adalah bahasa skrip yang berjalan di sisi klien. Sebagai contoh, setiap kali mengakses *internet* dan melihat *carousel* gambar yang bergulir, atau menu dengan tombol-tombol yang harus anda klik untuk mengungkapkan informasi, atau perubahan warna pada elemen peringatan di halaman *web*, semuanya adalah hasil dari penggunaan *JavaScript*. [17]

2.2.8 Database

Database adalah kumpulan data yang digunakan dalam suatu instansi atau organisasi. Satu database menunjukkan kumpulan data yang digunakan dalam suatu instansi atau organisasi. Pengolahan database melibatkan menyusun, mengurut, mengambil, dan menampilkan file dalam bentuk laporan. [17].



Gambar 2.5 Ilustrasi *Database*
(Sumber: <https://images.app.goo.gl/LYTFhDroeDyBBPzr7>)

3.2.9 *Visual Studio Code*

Adalah editor kode sederhana dengan dukungan untuk operasi pengembangan seperti *debugging*, menjalankan tugas, dan control versi.



Gambar 2. 6 *Visual Studio Code*

(Sumber: <https://images.app.goo.gl/Fd3BM8xaHpimSFhi8>)

3.2.10 *Balsamic Mockup*

Perangkat lunak yang digunakan untuk pembuatan desain atau *prototyping* dalam pembuatan tampilan *user interface* pada sebuah aplikasi.



Gambar 2. 7 *Balsamic Mockup*

(Sumber: <https://images.app.goo.gl/BTmYwTtq3n6JMhtK6>)