

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Hendrikus, dkk pada tahun (2022). Implementasi sistem monitoring dan pengontrolan pada kandang ayam broiler berhasil dilaksanakan dengan membuat sebuah *website* yang dapat memantau dan mengontrol suhu udara, kelembapan udara, tinggi air, lampu dan pakan ayam broiler. Perangkat keras yang digunakan masing-masing memiliki nilai akurasi meliputi pembacaan sensor suhu udara adalah 99,23%, kelembapan udara 98,67%, tinggi air 93,46%, dan pembacaan tinggi pakan 93,66%, dapat disimpulkan bahwa implementasi sistem monitoring berhasil dilaksanakan. Implementasi pengaruh sistem terhadap perkembangan ayam broiler sudah dilakukan dengan pengamatan dan perbandingan pada kedua jenis kandang yang digunakan. Berat ayam broiler pada umur 0 hari 100 gram dan umur 1-7 hari menjadi 200 gram terjadinya perbedaan pertumbuhan berat badan di minggu kedua. Pada kandang berbasis *internet of things* berat rata-rata 417 gram pada ke lima ekor ayam broiler. Sedangkan pada kandang konvensional berat badan rata-rata hanya mencapai 278,8 gram pada ke lima ekor ayam broiler [4].

Penelitian yang dilakukan oleh Putu Adi S, dkk pada tahun (2021). Penerapan metode fuzzy mamdani pada sistem kontrol otomatis suhu,

kelembaban, dan gas amonia yang diterapkan pada source code arduino. Yang dapat dimonitoring pada website dengan Firefox version 84.0.2, Brave version 88.1.19.86, Google Chrome version 87.0.4280.141 dan Microsoft Edge version 87.0.664.75 dengan tampilan login dan semua fungsi di halaman utama dan tampilan logout berfungsi dengan baik. Dari pengujian sistem dapat berjalan dengan baik menggunakan bahasa pemrograman PHP, MySQL sebagai database Dari pengujian sensor rata-rata presentase error pada sensor suhu 0,03 %, sensor kelembaban 0,13 % dan sensor ultrasonic 0,08% Sensor mq137 dapat mendeteksi gas ammonia dengan baik sesuai dengan real time yakni 100% .Berdasarkan penelitian yang telah dilakukan, maka peneliti dapat memberikan saran-saran untuk pengembangan selanjutnya yaitu untuk monitoring bisa dikembangkan menjadi aplikasi android dengan tampilan yang lebih mudah, efektif dan efesien [5].

Penelitian yang dilakukan oleh Nur Aeni Fahila, dkk pada tahun (2024). Dari hasil pengujian yang telah dilaksanakan, sistem yang telah dikembangkan beroperasi dengan baik. Setiap sensor yang diintegrasikan berhasil mentransmisikan data secara akurat ke situs web. Rekomendasi yang dapat diberikan adalah meningkatkan tampilan situs web dan menyediakan fitur sederhana untuk menampilkan data secara real-time. Selain itu, sistem masih dapat dikembangkan lebih lanjut dengan penambahan fitur atau peningkatan tampilan situs web. Implementasi

metode ini juga dapat dieksplorasi lebih lanjut dengan mempertimbangkan metode lain, seperti sistem pakar [6].

Penelitian yang dilakukan oleh Deni Kurnia, dkk pada tahun (2019). Hasil penelitian menunjukkan bahwa pengontrolan pemberian pakan ideal mampu memberikan hasil yang signifikan terhadap penghematan pakan ayam. Terbukti, dengan pemberian porsi yang tepat, bobot ayam memiliki hasil yang sama dengan kondisi awal saat belum dilakukan pengontrolan. Kondisi ini memungkinkan untuk meningkatkan keuntungan peternak ayam melalui penghematan pakan. Kemudahan lainnya sistem ini adalah, peternak menjadi mudah melakukan monitoring stok pakan karena data dapat ditampilkan secara realtime melalui web. Namun, dalam implementasinya dibutuhkan edukasi secara bertahap kepada para peternak tentang penggunaan sistem ini. Faktor lainnya yang harus menjadi perhatian adalah dukungan infrastruktur terutama koneksi internet yang cepat dan stabil. Sistem ini masih terbuka lebar untuk diteliti dan dikembangkan lagi untuk implementasi yang lebih besar dan kompleks [7].

Penelitian yang dilakukan oleh Muhamad Ripa Nawa Syarip, dkk pada tahun (2020). Setelah melakukan serangkaian kegiatan yang meliputi: Perancangan, Pembuatan alat, pengujian, dan analisa, maka didapatkan beberapa kesimpulan dalam tugas akhir ini, diantaranya yaitu Dengan menggunakan kandang ayam type clouse house maka peternak tidak perlu lagi untuk mengatur suhu/kelembaban secara manual. Pada

sensor suhu DS18B20 dan Sensor kelembaban DHT11 terdapat kesalahan nilai pembacaan atau error reading yaitu : sensor suhu DS18B20 paling besar itu 3,3% dan untuk sensor kelembaban DHT11 paling besar itu 3,07692% [8].

Penelitian yang dilakukan oleh Sofian Efendi pada tahun (2024) Pada peternakan ayam boiler ada beberapa faktor yang dapat mempengaruhi tumbuh kembang ayam. Selain ketersediaan makanan, faktor lingkungan pun menjadi faktor penting dalam tumbuh kembang ayam. Salah satu faktor lingkungan itu adalah suhu dari pada kandang tersebut. Suhu bisa jadi yang terpenting dikarenakan suhu yang dirasakan oleh ayam dapat mempengaruhi nafsu makan ayam, misalnya jika suhu kandang terlalu panas maka ayam akan lebih banyak minum, dan jika suhu kandang terlalu dingin maka ayam akan cenderung lebih banyak makan. Penyesuaian suhu pada kandang ayam dapat diklasifikasikan berdasarkan umur ayam, ayam pada umur 1-7 hari suhu idealnya 34°C - 32°C [9].

Tabel 2. 1 Tabel Penelitian Sebelumnya

No	Peneliti	Judul Penelitian	Teknologi Yang Digunakan	GAP/ Pembaruan dari Penelitian Sebelumnya
1.	Hendrikus dkk (2022)	Sistem Pengontrolan Dan Monitoring Pada Kandang Ayam Broiler Berbasis	Sensor suhu, kelembapan, ultrasonik, <i>load cell</i> , <i>Website</i>	Tidak menggunakan <i>framework</i> modern dan real-time. Penelitian ini menggunakan CodeIgniter dan JavaScript untuk efisiensi dan

		<i>Internet Of Things (IoT)</i>		keamanan serta tampilan <i>real-time</i> .
2.	Putu Adi S, dkk (2021)	Penerapan Logika Fuzzy Pada Sistem Monitoring Dan Kontrol Kandang Ayam Otomatis Berbasis IoT	Arduino, MQ137, sensor suhu & kelembaban, <i>PHP</i> , <i>MySQL</i>	Belum mengatur stok pakan atau tampilan dinamis. Penelitian Anda menambahkan analisis stok serta JavaScript <i>real-time</i> .
3.	Nur Aeni Fahila, dkk (2024)	Implementasi Fuzzy Mamdani Pada Sistem Automasi Dan Monitoring Ayam Broiler Berbasis Internet Of Things (IoT)	<i>Website</i> , sensor suhu, kelembaban, air, pakan	Penelitian pembaruan sudah mengimplementasikan <i>real-time</i> JavaScript, UI lebih baik, dan sistem analisis stok pakan.
4.	Deni Kurnia, dkk (2019)	Implementasi NodeMCU Dalam Prototipe Sistem Pemberian Pakan Ayam Otomatis Dan Presisi Berbasis Web	<i>Website</i> monitoring, kontrol otomatis pakan	Belum menggunakan framework. Penelitian sekarang menggunakan CodeIgniter untuk keamanan dan modularitas, serta MySQL untuk pengelolaan data.
5.	Muhamad Ripa Nawa Syarip, dkk (2020)	Perancangan Dan Pembuatan Sistem Monitoring Suhu Ayam, Suhu Dan Kelembaban Kandang Untuk Meningkatkan	Sensor DS18B20, DHT11	Tidak mencakup pakan dan analisis stok. Penelitian Sekarang menggabungkan kontrol suhu, kelembaban, dan manajemen pakan otomatis.

		Produktifitas Ayam Broiler.		
6.	Sofian Efendi (2024)	Penerapan Sistem Monitoring Kandang Ayam Broiler Closed House Berbasis IoT Pada Studi Kasus Moldovar Farm	NodeMCU ESP8266, Sensor DHT22, Sensor Ultrasonik, Motor Servo	Penelitian Sekarang mengembangkan otomatisasi pengaturan suhu berdasarkan usia ayam dan monitoring real-time.

Penelitian ini menghadirkan beberapa pembaruan dibandingkan penelitian sebelumnya. Penggunaan *CodeIgniter* membuat sistem lebih efisien dan aman dibandingkan PHP biasa. JavaScript memungkinkan pemantauan real-time tanpa perlu refresh halaman, Database MySQL digunakan untuk menyimpan dan menganalisis data pakan, membantu monitoring stok secara lebih akurat.

## 2.2 Landasan Teori

### 2.2.1 Sistem Monitoring

Sistem monitoring atau sistem pengawasan adalah suatu proses sistematis yang bertujuan menetapkan standar kinerja dalam perencanaan, merancang sistem umpan balik informasi, membandingkan kinerja aktual dengan standar yang telah ditetapkan, serta menentukan apakah terdapat penyimpangan. Jika terjadi penyimpangan, sistem ini juga berfungsi untuk mengambil

tindakan perbaikan yang diperlukan agar semua sumber daya perusahaan atau organisasi dapat digunakan secara efektif dan efisien guna mencapai tujuan yang telah ditetapkan [10].

### 2.2.2 PHP

PHP adalah bahasa pemrograman berbasis *website* yang dirancang khusus untuk pengembangan aplikasi *website*. Bahasa ini pertama kali dibuat oleh *Rasmus Lerdorf*, seorang pengembang perangkat lunak sekaligus anggota tim *Apache*, dan dirilis pada akhir tahun 1994. Awalnya, PHP dikembangkan hanya untuk mencatat pengunjung pada situs web pribadi Lerdorf. Selain bersifat *open source* dan tersedia secara gratis, PHP juga dikenal mudah dipelajari oleh siapa saja, sehingga banyak digunakan dalam pembuatan berbagai aplikasi *website* [11].



Gambar 2. 1 Gambar logo php

### 2.2.3 MySQL

*MySQL* merupakan sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multiuser serta menggunakan perintah standar SQL (*Structured Query Language*) [12].



Gambar 2. 2 Gambar logo MySQL

#### 2.2.4 Visual Studio Code

Visual Studio Code merupakan sebuah aplikasi editor *code open source* yang dikembangkan oleh Microsoft untuk sistem operasi *Windows, Linux, dan MacOS*. Visual Code memudahkan dalam penulisan code yang mendukung beberapa jenis pemrograman, seperti *C++, C#, Java, Python, PHP, GO*. Visual Code memiliki kemampuan untuk mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian code tersebut. Visual Studio Code juga telah terintegrasi ke Github. Selain itu fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di Visual Studio Code [13].



Gambar 2. 3 Gambar Visual Studio Code

### 2.2.5 Database

Database merupakan suatu koleksi terstruktur dari data yang saling terkait, disimpan dalam media penyimpanan komputer, dan dapat diakses serta dikelola menggunakan perangkat lunak khusus. Database digunakan untuk menyimpan, mengelola, dan mengorganisir data dengan tujuan memberikan akses yang efisien, aman, dan terstruktur terhadap informasi [14].

### 2.2.6 Codeigniter

*CodeIgniter* adalah Sebuah *framework* php yang bersifat open source dan menggunakan metode MVC (Model,*View*,*Controller*) untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal [15].



Gambar 2. 4 Gambar logo *CodeIgniter*

### 2.2.7 Java Script

JavaScript adalah sekumpulan skrip yang dijalankan di dalam dokumen HTML. Awalnya dikembangkan khusus untuk penggunaan pada *website*, JavaScript berfungsi sebagai bahasa pemrograman yang memberikan fungsi tambahan pada halaman *website*. Perintah-perintah JavaScript dieksekusi di sisi pengguna (*client-side*), yaitu langsung di dalam browser, bukan di server *website*. Dengan demikian, JavaScript memungkinkan interaktivitas dan dinamika pada halaman *website* tanpa harus mengirim permintaan ke server secara terus-menerus.



Gambar 2. 5 Gambar logo Java Script

### 2.2.8 UML (*Unified modeling language*)

*Unified Modeling Language* (UML) adalah salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan perangkat lunak berorientasi objek. UML berfungsi sebagai standar dalam penulisan atau blueprint yang mencakup berbagai aspek, seperti proses bisnis, diagram kelas, serta elemen-elemen lain yang spesifik untuk mendeskripsikan struktur dan

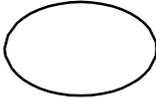
perilaku sistem secara terorganisir dan mudah dipahami. Dengan UML, pengembang dapat memvisualisasikan, mendokumentasikan, dan merancang sistem secara efektif sebelum implementasi.

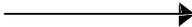
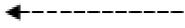
Dalam perancangan sistem terdapat UML yang sering digunakan sebagai berikut:

1. *Use Case Diagram*

Dalam pengembangan sistem informasi, use case diagram berfungsi sebagai pemodelan untuk menggambarkan perilaku atau aktivitas sistem informasi. Diagram ini bersifat statis karena menunjukkan fungsi-fungsi utama sistem dan interaksi antara pengguna (aktor) dengan sistem, tanpa menjelaskan detail prosesnya. Untuk simbol *Use Case Diagram* disajikan pada Tabel 2.1.

Tabel 2. 2 Use Case Diagram

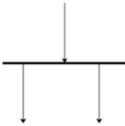
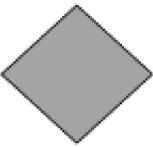
No	Simbol	Keterangan
1.		<i>Use case</i> aktivitas yang dapat dilakukan <i>actor</i> pada sistem
2.		<i>Aktor</i> entitas yang berinteraksi dengan sistem

3.	 —	<i>Association</i> hubungan antara <i>actor</i> dengan <i>usecase</i>
4.	 -- -- -- -- -- <<include>>	<i>Includes</i> suatu <i>use case</i> termasuk bagian dari <i>use case</i> lain
5.	 —————>	<i>Dependency</i> ketergantungan dengan elemen-elemen <i>use case</i> lain
6.	 ←----- <<extends >>	<i>Extend</i> satu <i>use case</i> dapat diperluas dengan <i>use case</i> lain

## 2. Activity Diagram

Activity Diagram atau Diagram Aktivitas adalah diagram yang bersifat statis dan digunakan untuk memvisualisasikan rangkaian aktivitas dalam suatu sistem bisnis. Simbol-simbol yang digunakan dalam diagram aktivitas ini dapat dilihat pada Tabel 2.2.

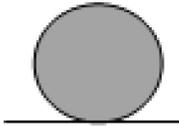
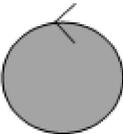
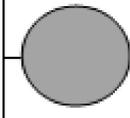
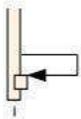
Tabel 2. 3 Activity Diagram

No	Simbol	Keterangan
1.		<i>End point</i> menunjukkan bagian akhir dari aktivitas
2.		<i>Start Point</i> menandakan Tindakan awal atau titik pada aktivitas pada <i>activity diagram</i>
3.		<i>Fork</i> berfungsi memecah <i>behaviour</i> menjadi <i>activity</i> atau <i>action</i> yang paralel
4.		<i>Activity</i> merupakan aktivitas 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 menggambarkan interaksi berupa pengiriman dan penerimaan pesan antar objek. Simbol-simbol yang digunakan dalam diagram urutan ini disajikan pada Tabel 2.3.

Tabel 2. 4 Squence Diagram

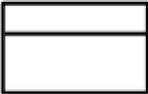
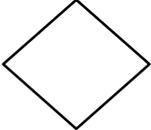
No	Simbol	Keterangan
1.		<i>Entity Class</i> gambaran sistem sebagai landasan dalam Menyusun basis data
2.		<i>ControlClass</i> bertanggung jawab terhadap kelas objek yang berisi logika
3.		<i>Boundary Class</i> menangani komunikasi antar lingkungan sistem
4.		Pesan atau <i>message</i> menunjukkan pengiriman pesan antar <i>class</i>
5.		<i>Self message</i> menunjukkan pengiriman pesan untuk diri sendiri
6.		<i>Activation</i> mewakili durasi aktivitas sebuah operasi

7.		<i>Lifeline</i> komponen yang digambarkan garis putus terhubung dengan objek
----	---	--

#### 4. *Class Diagram*

Class Diagram adalah diagram statis yang menampilkan kumpulan kelas, antarmuka, serta hubungan di antara mereka. Simbol-simbol yang digunakan dalam Class Diagram dapat dilihat pada Tabel 2.4.

Tabel 2. 5 Class Diagram

No	Simbol	Keterangan
1.		<i>Generallization</i> , hubungan dimana objek berbagi perilaku dan struktur data dari objek yang ada di atasnya
2.		<i>Class</i> , adalah kumpulan objek yang saling berbagi.
3.		<i>Nary Assocation</i> , digunakan untuk asosiasi terhindar dengan objek lainnya.
4.		<i>Collaboration</i> 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.		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri
7.		<i>Association</i> menghubungkan antara objek satu dengan objek yang lain