

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian yang dilakukan oleh Dwiarga Sholahuddin dan Agung Setia Budi (2023) pada artikel jurnal berjudul “Purwarupa Sistem Monitoring dan Otomasi Hidroponik Berbasis IoT Menggunakan Aplikasi Android” mengembangkan sebuah sistem berbasis Internet of Things (IoT) yang mampu memantau dan mengendalikan kondisi lingkungan dalam sistem pertanian hidroponik. Sistem ini menggunakan mikrokontroler ESP8266 yang terhubung dengan sensor pH dan suhu, serta dikontrol melalui aplikasi Android sebagai antarmuka pengguna. Hasil penelitian menunjukkan bahwa sistem mampu memberikan pembacaan data secara real-time, sehingga sangat membantu dalam meningkatkan efisiensi dan kontrol budidaya tanaman hidroponik secara berkelanjutan [5].

Penelitian selanjutnya dilakukan oleh tim dari Universitas Trunojoyo Madura (2023) dengan judul “Rancang Bangun Sistem Monitoring Suhu, pH dan Kejernihan Air Pada Kolam”. Sistem yang dikembangkan menggunakan sensor suhu DS18B20, sensor pH, dan sensor kejernihan air yang digabungkan dalam satu sistem monitoring. Data yang diperoleh dikirim secara otomatis ke platform IoT dan dapat diakses melalui aplikasi Android. Sistem ini membantu pemilik kolam dalam mengawasi kualitas air untuk memastikan ekosistem air tetap stabil [6].

Selanjutnya, Nur Rachmat (2024) melalui penelitian berjudul “Implementasi LLM Gemini pada Pengembangan Aplikasi Chatbot Berbasis Android” mengembangkan sistem chatbot cerdas berbasis Android menggunakan Gemini API. Sistem ini memungkinkan pengguna untuk berinteraksi langsung dengan chatbot melalui antarmuka aplikasi mobile guna memperoleh informasi sesuai kebutuhan. Penelitian ini menggunakan metode prototyping, dengan pemanfaatan LLM Gemini untuk melakukan parsing input pengguna, menghasilkan respons dinamis, serta menyusun skenario percakapan yang kontekstual. Aplikasi ini menunjukkan efektivitas Gemini API dalam menghadirkan percakapan interaktif dan responsif, serta relevan untuk diterapkan dalam sistem monitoring berbasis mobile yang mengutamakan efisiensi komunikasi antara pengguna dan sistem[7].

1. Tabel Penelitian Terkait

Tabel 2. 1 Tabel Penelitian Terkait

Penulis	Judul	Fokus	Tools
Dwiarga Sholahuddin	Mengembangkan aplikasi sistem Monitoring Kualitas Air Berbasis IoT.	Monitoring kualitas air secara real-time.	Android Studio, Firebase .
Affan Bachri	Rancang Bangun Sistem Monitoring Suhu, pH dan Kejernihan Air Pada Kolam.	Monitoring suhu, pH, dan kejernihan air kolam, data dikirim ke cloud dan diakses lewat smartphone.	Android Studio, Firebase

Nur Rachmat (2024)	Implementasi LLM Gemini pada Pengembangan Aplikasi Chatbot Berbasis Android	Membuat chatbot interaktif yang dapat menjawab pertanyaan dan merespons perintah pengguna secara cerdas	Android Studio, API Gemini AI
--------------------	---	---	-------------------------------

2. Tabel yang diteliti

Tabel 2. 2 Tabel Yang Di teliti

Penulis	Judul	Fokus	Tools
Moch Ahdi Al Wafa. Ega Ardhi Kusuma.	Sistem Monitoring Tanaman Tomat Hidroponik dengan Pemanfaatan Solar Panel sebagai Sumber Energi Berbasis Mobile	Monitoring serta suhu air, kualitas air, dan ph air melalui aplikasi android.	Android Studio, Firebase

2.2 Landasan Teori

2.2.1. Android

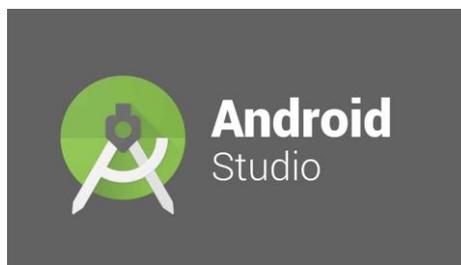
Android adalah sistem operasi seluler yang dirancang untuk perangkat seluler terutama seperti HP dan dibuat secara komersial oleh Open Handset Alliance. Itu pertama kali dirilis di bulan September 2008 [8].



Gambar 2. 1 Android

2.2.2. Android Studio

Android Studio, yang dibangun di atas IntelliJ IDEA, adalah Integrated TEMATICS (Technology Management and Informatics Research) Vol. 5 No. 2 2021 29 Programming Environment (IDE) resmi untuk mengembangkan aplikasi Android. Selain sebagai editor kode serbaguna dan perangkat lunak pengembangan IntelliJ, Android Studio memiliki sejumlah fitur yang membantu Anda membuat aplikasi Android lebih cepat [9]. Sebagai IDE resmi untuk pengembangan aplikasi Android, Android Studio menawarkan banyak fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant sebagai build environment, Android Studio menggunakan Gradle, yang lebih fleksibel .



Gambar 2. 2 Android Studio

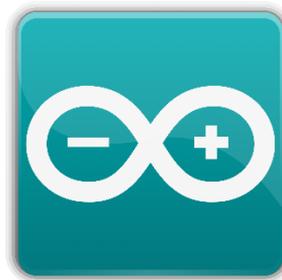
2.2.3. Android SDK

Android SDK merupakan sebuah API (application programming interface) yang digunakan sebagai awalan dalam mengembangkan aplikasi yang ada pada platform Android menggunakan bahasa pemrograman Java. Android SDK telah tersedia sebagai API dan alat untuk memulai pengembangan aplikasi

Android menggunakan bahasa pemrograman Java. SDK berisi beberapa perangkat alat untuk pengembangan yang lengkap[10].

2.2.4. Arduino IDE

Arduino IDE adalah software yang digunakan untuk membuat sketch pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman pada board yang ingin diprogram. Arduino IDE ini berguna untuk mengedit, membuat, meng-upload ke board yang ditentukan, dan meng-coding program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi dengan library C/C++(wiring), yang membuat operasi input/output lebih mudah [11].



Gambar 2. 3 Arduino IDE

2.2.5. Kotlin

Kotlin di resmikan sebagai Bahasa kedua yang didukung untuk di pakai membuat aplikasi Android di Android Studio. Saat ini, Posisi Java sebagai Bahasa utama memang belum tergantikan oleh Kotlin. Namun tidak dapat disangkal pula bahwa, perkembangan Kotlin di komunitas Developer Android sangat pesat[12].



Gambar 2. 4 Kotlin

2.2.6. *Firestore*

Firestore memiliki SDK real-time dan memprioritaskan aplikasi seluler dan mendukung penyimpanan data local secara off-line. Tujuan utama dari realtime database firestore adalah kinerja waktu akses data. Realtime database mengoptimalkan waktu akses, sehingga akses data berada dikisaran mikrodetik bahkan nanodetik, sehingga biaya akses data dapat diminimalisir[13].

Dengan menggunakan Firestore, apps developer bisa fokus dalam mengembangkan aplikasi tanpa memberikan effort yang besar untuk urusan backend. Mengenai segi layanan, dulu Firestore memberikan service trial (percobaan), namun saat ini kamu bisa memanfaatkan dan menggunakan layanan Firestore secara free (gratis). Tentu saja dengan adanya batasan-batasan tertentu.



Gambar 2. 5 Firestore

2.2.7. Bahasa Java

Java adalah bahasa pemrograman yang biasa digunakan untuk mengembangkan bagian back-end dari software, aplikasi Android, dan juga website. Java juga dikenal memiliki moto “Write Once, Run Anywhere”. Artinya, Java mampu dijalankan di berbagai platform tanpa perlu disusun ulang menyesuaikan platformnya. Misalnya, berjalan di Android, Linux, Windows, dan lainnya. Hal itu dapat terjadi karena Java memiliki sistem syntax atau kode pemrograman level tinggi. Di mana ketika dijalankan, syntax akan di-compile dengan Java Virtual Machine (JVM) menjadi kode numeric (bytecode) platform[14].



Gambar 2. 6 Bahasa Java

2.2.8. (UML) *Unified Modeling Language*

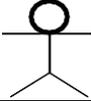
(UML) *Unified Modeling Language* adalah bahasa visual yang digunakan untuk menjelaskan, memvisualisasikan, membangun, dan mendokumentasikan sistem yang kompleks. UML digunakan untuk memahami, menganalisis, memelihara, dan mengendalikan informasi tentang suatu sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai

berikut:

1. Use Case Diagram

Merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

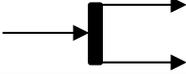
Tabel 2. 3 Simbol Use Case Diagram

Simbol	Keterangan
1. Aktor 	Aktor adalah Perangkat, manusia yang berperan dalam sistem yang dibangun
2. Use Case 	Hubungan aktor dan sistem yang terdapat didalamnya
Simbol	Keterangan
3. Association 	Penghubung sistem dan aktor dengan use case
4. Generalisasi 	Pengkhususan aktor yang bisa bergabung dengan use case
5. <<Include>> 	Untuk menunjukan seluruh sistem yang dibangun dari fungsional sistem lainnya
6. <<Extend>> 	Untuk menunjukan bagian dari sistem diagram tambahan dengan yang lainnya jika kondisi terpenuhi

2. Activity Diagram

Activity Diagram menunjukkan alur kerja atau *activity* suatu sistem atau proses bisnis. Simbol-simbol yang digunakan dalam diagram *activity* adalah sebagai berikut:

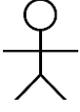
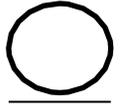
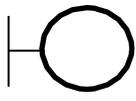
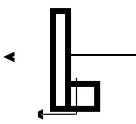
Tabel 2. 4 Simbol Activity Diagram

Simbol	Keterangan
1. <i>Activity</i> 	Untuk memvisualisasikan <i>UI</i> dari setiap kelas yang saling terhubung
2. <i>Action</i> 	Keadaan suatu sistem yang menggambarkan implemementasi dari suatu aksi
3. <i>Start</i> 	Penggambaran dari suatu sistem untuk membentuk objek
4. <i>End</i> 	Penggambaran dari suatu sistem untuk membentuk objek dan dihancurkan
5. State Transitio 	Aksi selanjutnya
6. Fork 	Cabang dari aliran pada diagram activity
7. Join 	Arah aliran diagram activity yang digabung
8. 	Pilihan pada sistem untuk mengambil suatu keputusan
9.Flow Final 	Akhir dari aliran

3. Sequence Diagram

Sequence Diagram menggambarkan perilaku objek dalam *use case* dengan menggambarkan waktu hidup objek dan pesan yang dikirim antar objek.

Tabel 2. 5 Simbol Squence Diagram

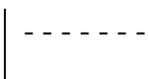
Simbol	Keterangan
1. Actor 	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2. <i>Entity Class</i> 	Gambaran sistem sebagai landasan dalam Menyusun basis data
3. <i>Boundary Class</i> 	Menangani komunikasi antar lingkungan sistem
4. <i>Control Class</i> 	Bertanggung jawab terhadap kelas – kelas terhadap objek yang berisi logika.
5. <i>Recursive</i> 	Pesan untuk dirinya
6. <i>Activation</i> 	Mewakili proses durasi aktivasi sebuah operasi

<p>7. <i>Life Line</i></p> 	<p>Komponen yang digambarkan garis putus terhubung dengan objek</p>
--	---

4. Class Diagram

Class diagram merupakan penghubung antara kelas dan penjelasan rinci setiap kelas dalam model perancangan suatu sistem. Hal ini juga menunjukkan karakteristik yang melekat pada sistem dan tanggung jawab entitas.

Tabel 2. 6 Simbol Class Diagram

Simbol	Keterangan
<p>1. <u><i>Generalization</i></u></p>	<p>Hubungan antara objek (<i>descendent</i>) dan struktur data objek induk (<i>ancestor</i>)</p>
<p>2. <i>Nary Association</i></p> 	<p>Suatu Aksi untuk terhindar dari asosiasi yang memiliki objek lebih dari 2</p>
<p>3. <i>Class</i></p> 	<p>Kumpulan objek atribut dan operasi yang sama</p>
<p>4. <i>Realization</i></p> 	<p>Tindakan dari suatu objek</p>
<p>5. <i>Dependency</i></p> 	<p>Suatu perubahan pada elemen yang mempengaruhi elemen yang tidak mandiri.</p>
<p>6. <u><i>Association</i></u></p>	<p>hubungan objek 1 dengan yang lainnya</p>