

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Penelitian yang dilakukan oleh (Asniati , Ery Muchyar Hasiri ,Fitriani, 2021) tujuan penelitian ini membuat sebuah alat yang mempermudah pengguna dalam melakukan perawatan tanaman, dengan menggunakan NodeMCU ESP8266 dan sensor soil moisture untuk mendeteksi kelembaban tanah, serta sensor DHT-11 yang dapat di monitoring secara realtime dengan memanfaatkan firebase dan aplikasi android yang terhubung dengan jaringan internet, lalu ada relay 4 channel yang akan digunakan untuk menyalakan pompa jika kondisi kelembaban tanah kurang dari 700 RH jika keadaan suhu lebih dari 32°C relay akan otomatis aktif, fungsi dari relay ini sebagai alat output untuk mengaktifkan dan menonaktifkan alat seperti lampu, kipas dan pompa[5].

Penelitian lainnya dilakukan (Asniati , Ery Muchyar Hasiri ,Fitriani, 2021) penelitian ini menghasilkan alat yang dapat menyiram tanaman menggunakan perangkat android untuk memantau dan kontrol. Alat ini menggunakan Arduino yang terhubung dengan ESP8266 untuk berkomunikasi dengan Android melalui aplikasi Blynk dan menggunakan wifi. Pengguna dapat mengontrol dan memantau alat penyiraman tanaman dengan mengirimkan perintah on/off melalui Blynk, serta menerima data suhu dan kelembaban dari sensor DHT22. Hasil penelitian ini menunjukkan bahwa alat penyiram tanaman mampu bekerja dengan baik,

mampu mengontrol penyiraman secara manual dan otomatis. Alatakan menyiram tanaman bila suhu lebih dari 31°C[6].

Dalam penelitian yang juga dilakukan oleh (Asniati , Ery Muchyar Hasiri ,Fitriani, 2021) mengembangkan sebuah sistema kontrol otomatis penyiraman tanaman menggunakan *sprayer* berbasis Android. Sistem ini diberi sumber arus listrik 5-9 volt dan melakukan pemindaian jaringan Wi-Fi yang telah ditentukan dalam memori EPROM. Setelah terhubung, NodeMCU membaca data dari sensor kelembaban tanah (sensor Moisture) untuk mengoperasikan relay yang mengontrol pompa penyiraman: relay mengaktifkan pompa jika kelembaban tanah di bawah 70% dan mematikannya jika di atas 70%. Hasil pengujian menunjukkan efektivitas sistem ini dalam mengontrol penyiraman tanaman secara otomatis berdasarkan kelembaban tanah[7].

Penelitian lain oleh (Maharani, Ayasha Ninda, Handaga, Bana, 2022) penelitian ini menghasilkan aplikasi android yang dapat memonitor nilai kelembapan tanah pada tanaman dalam satuan waktu dan tombol on/off yang dapat menghidupkan dan mematikan pompa air penyiram tanaman. Pengguna dapat melihat nilai kelembapan tanah secara realtime melalui aplikasi android dengan menggunakan koneksi internet. Penggunaan ini menggunakan server firebase sebagai database dengan memanfaatkan fitur *realtime* firebase *database* milik *server* firebase. dari peneliatian yang meraka lakukan mendapatkan hasil bahawa aplikasi berfungsi sebagaimana semestinya dan mendapat kategori “Sangat Baik” sebesar 94,44% dari responden[8].

## 2.2 Landasan Teori

### 2.2.1 *Internet Of Things*

*Internet Of Things* merupakan teknologi yang ditemukan oleh Kevin Ashton pada tahun 1999. Cara kerja *internet of things* adalah setiap bendayang sudah terpasang sensor atau modul internet of things mengirimkan data atau informasi ke pengguna melalui internet dan bisa diakses kapan saja dan dimana saja tanpa terbatas oleh jarak. Fungsi dari *internet of things* sendiri adalah untuk memudahkan *monitoring* dan *kontrolling* suatu benda pada kehidupan sehari-hari. Selain itu informasi yang didapat bisa setiap waktu pada *internet of things*[9].

### 2.2.2 **Android Studio**

Android Studio adalah salah satu IDE untuk pengembangan aplikasi android, yang didasarkan pada IntelliJ IDEA. Selain sebagai code editor, Android Studio juga menawarkan banyak fitur yang dapat meningkatkan produktivitas seorang *developer* dalam mengembangkan aplikasi Android[10].

### 2.2.3 **Figma**

Figma adalah salah satu *design tool* yang biasanya digunakan untuk membuat tampilan aplikasi *mobile*, *desktop*, *website* dan lain-lain[11]. Figma bisa digunakan di sistem operasi windows, linux ataupun mac dengan terhubung ke internet. Figma memiliki keunggulan yaitu untuk pekerjaan yang sama dapat dikerjakan oleh lebih dari satu orang secara

bersama-sama walaupun ditempat yang berbeda. Hal tersebut bisa dikatakan kerja kelompok dan karena kemampuan aplikasi figma tersebut lah yang membuat aplikasi ini menjadi pilihan banyak UI/UX design untuk membuat *prototy website* atau aplikasi dengan waktu yang cepat dan efektif[12].

#### **2.2.4 Firebase**

Firebase adalah *platform* seluler yang membantu *developer* mengembangkan aplikasi berkualitas tinggi secara cepat, berbasis pengguna, dan dapat menghasilkan uang lebih banyak. Firebase terdiri dari fitur pelengkap yang bisa dipadukan sesuai dengan kebutuhan (Google). Digunakannya firebase, karena *platform* yang telah dikembangkan oleh Google ini memiliki fitur cukup mumpuni untuk dimanfaatkan pada beberapa aplikasi[13].

#### **2.2.5 Firebase *Realtime Database***

*Realtime Database* adalah *database cloud*. Data disimpan di dalam format JSON dan disinkronkan ke setiap klien yang terhubung secara *realtime*. Firebase biasanya digunakan pada aplikasi *realtime*. Ketika terjadi pembaruan data, maka aplikasi yang terhubung pada Firebase akan secara otomatis menerima pembaruan data. Firebase *realtime database* adalah *databses nonSQL* dan karena itu memiliki optimalisasi dan fungsionalitas yang berbeda dibandingkan dengan basis data relasiona[14].

### 2.2.6 Kotlin

Kotlin adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM) yang dikembangkan oleh JetBrains. Bahasa pemrograman ini bersifat pragmatis untuk android yang mengkombinasikan *object oriented* (OO) dan pemrograman fungsional. Kotlin juga bahasa pemrograman yang *interoperabilitas* yang membuat bahasa ini dapat digabungkan dengan bahasa pemrograman Java pada suatu *project* yang sama. Kotlin juga dapat digunakan untuk mengembangkan aplikasi berbasis desktop, web dan bahkan untuk *backend*[15][16].

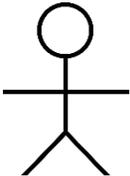
### 2.2.7 Unified Modeling Language (UML)

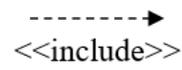
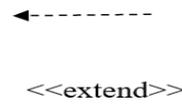
Alat berorientasi objek untuk desain sistem adalah *Unified Modeling Language* (UML)[17]. Perangkat lunak didokumentasikan, ditentukan, dan dibangun menggunakan UML, bahasa spesifikasi standar. Selain itu, UML menawarkan pedoman untuk membuat sistem cetak biru yang mencakup skema database, kelas penulisan dalam bahasa pemrograman tertentu, ide proses bisnis, dan elemen lain yang diperlukan untuk membuat sistem perangkat lunak[18]. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

## 1. Use Case Diagram

*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[19]. Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat pada Table 2.1.

Tabel 2.1 *Use Case Diagram*

No	Simbol	Keterangan
1.		Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
2.		<i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
3.		Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.

No	Simbol	Keterangan
4.		Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk Mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
5.		Merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
6.		Merupakan perluasan dari <i>use case</i> lain jika kondisi syarat terpenuhi.

## 2. Diagram Aktivitas (*Activity Diagram*)

*Diagram Aktivitas* menggambarkan proses atau kegiatan dalam sebuah sistem. Simbol-simbol yang digunakan dalam *Diagram Aktivitas* dapat dilihat pada Tabel 2.2.

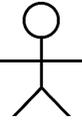
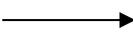
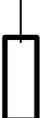
Tabel 2.2 *Diagram Activity*

No	Simbol	Keterangan
1.		Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		Bagaimana objek dibentuk dan dihancurkan.
3.		Bagaimana objek dibentuk atau diawali.
4.		Satu aliran yang tahap tertentu berubah menjadi beberapa aliran.

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence Diagram* menampilkan interaksi antara objek dengan sistem sesuai urutan kejadian. Diagram ini berfungsi untuk menggambarkan kolaborasi dinamis dalam penyelesaian tugas. Simbol-simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 *Sequence Diagram*

No	Simbol	Keterangan
1.		Menggambarkan pengguna yang sedang menggunakan sistem.
2		Hubungan antara antarmuka pengguna dan tabel data.
3		Pengiriman pesan yang tidak memerlukan respon langsung.
4		Penghubung antara antarmuka pengguna dan tabel data.
5		Menggambarkan sebuah gambaran dari form antar penguana.
		Penghubung antara antarmuka pengguna dan tabel data.