BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Firdaus (2023) berjudul "Sistem Manajemen Pengairan pada Budidaya Tanaman Anggrek Berbasis *Internet of Things* (IoT)". Dalam budidaya anggrek di daerah tropis, pengaturan kelembapan tanah sangat penting. Pengendalian kelembapan tanah dapat dilakukan melalui penyiraman air. Namun, penyiraman secara manual membutuhkan pemantauan kelembapan tanah secara terus-menerus, yang tidak efisien. Oleh karena itu, diperlukan sistem berbasis *Internet of Things* (IoT) yang mampu memantau dan mengendalikan penyiraman tanaman secara otomatis. Sistem ini menggunakan ESP32 sebagai mikrokontroler, sensor kelembapan tanah untuk melakukan pengukuran, serta *relay* untuk mengendalikan pompa air. Status sistem dan kontrol penyiraman ditampilkan melalui antarmuka pengguna, di mana pengguna juga dapat mengatur jadwal penyiraman sesuai kebutuhan.[8].

Penelitian yang dilakukan oleh Muhammad Khaerul Mukmin, Nurul Janah, dan Riani Mastita, mahasiswa Politeknik Harapan Bersama Tegal pada tahun 2020, menghasilkan sebuah produk *smart garden* untuk tanaman anggrek berbasis *Internet of Things* (IoT). Sistem ini menggunakan mikrokontroler NodeMCU ESP8266 yang terintegrasi dengan sensor DHT11 dan sensor LDR, serta dilengkapi dengan mekanisme penyiraman otomatis

melalui metode penyemprotan. Sistem bekerja secara otomatis berdasarkan data yang diperoleh dari sensor. Ketika sensor DHT11 mendeteksi suhu atau kelembaban di luar ambang batas yang ditentukan, maka pompa air akan aktif untuk menyiram tanaman. Sementara itu, sensor LDR berfungsi untuk mendeteksi intensitas cahaya, jika cahaya berlebihan atau kurang, sistem akan mengaktifkan *motor stepper* untuk membuka atau menutup atap secara otomatis, guna menjaga kondisi lingkungan yang ideal bagi tanaman anggrek.[9].

Penelitian yang dilakukan oleh Hidayat (2022) membahas sistem monitoring kelembapan, suhu, dan intensitas cahaya pada tanaman anggrek menggunakan mikrokontroler ESP8266 dan Arduino Nano. Sistem ini memanfaatkan beberapa sensor dan komponen utama, antara lain Arduino Nano dan ESP8266 sebagai mikrokontroler, sensor DHT11 untuk mengukur suhu dan kelembapan udara, sensor LDR untuk mendeteksi intensitas cahaya, serta *sensor soil moisture* untuk mengukur kelembapan tanah. Selain itu, sistem dilengkapi dengan pompa air untuk penyiraman otomatis, kipas untuk menurunkan suhu, serta modul *relay* sebagai pengendali arus listrik ke pompa dan kipas. Cara kerja sistem dimulai dengan pembacaan data lingkungan oleh sensor-sensor yang terpasang. Data yang diperoleh kemudian dikirim ke mikrokontroler untuk dianalisis. Jika suhu udara terdeteksi melebihi 30°C, mikrokontroler akan mengaktifkan *relay* untuk menyalakan kipas guna menurunkan suhu. Sebaliknya, jika kelembapan tanah menunjukkan nilai lebih dari 1000 (dalam satuan ADC), maka pompa air akan diaktifkan untuk

menyiram tanaman. Seluruh data hasil pengukuran ditampilkan dalam bentuk grafik dan angka melalui platform *ThingsBoard*, sehingga memudahkan pengguna dalam memantau kondisi tanaman secara *real-time*.[10].

Penelitian yang dilakukan oleh Husdi (2023) berjudul Monitoring Kelembaban Tanah Pertanian Menggunakan *Sensor Soil Moisture* FC-28 dan Arduino UNO. Penelitian ini membahas pemanfaatan teknologi *Internet of Things* (IoT) dalam bidang pertanian, khususnya untuk memantau tingkat kelembapan tanah sebagai media tanam tanaman hortikultura. Sistem yang dikembangkan menggunakan sensor kelembapan tanah FC-28 yang terhubung dengan mikrokontroler Arduino UNO. Hasil pengujian dalam penelitian ini menunjukkan bahwa sistem mampu memantau kondisi kelembapan tanah secara *real-time*, sehingga dapat dijadikan dasar bagi petani dalam merumuskan strategi pengelolaan lahan. Informasi yang diperoleh membantu petani dalam mengambil keputusan terkait waktu dan kebutuhan penyiraman tanaman. Sensor FC-28 menghasilkan data analog yang dikategorikan ke dalam tiga kondisi: basah dengan nilai rentang antara 150 sampai 339, lembab pada rentang 340 sampai 475, dan kering jika nilai sensor berada antara 476 sampai 1023. Klasifikasi ini memberikan gambaran yang jelas mengenai status kelembapan tanah dan dapat digunakan sebagai acuan untuk tindakan selanjutnya di lapangan.[11].

Penelitian yang dilakukan oleh Ahmad Zafrullah (2021) dilaksanakan di Desa Karihkil, Kecamatan Ciseeng, Kabupaten Bogor, Jawa Barat 16120. Penelitian ini berlangsung selama kurang lebih dua bulan, dengan pembagian

waktu yang terstruktur. Pada bulan pertama, yaitu Januari hingga Februari 2024, kegiatan difokuskan pada pengumpulan data secara intensif melalui metode observasi, wawancara, dan dokumentasi. Pemilihan jangka waktu ini bertujuan untuk memperoleh data yang mendalam. Selanjutnya, bulan kedua digunakan untuk proses pengolahan data, yang mencakup analisis hasil, penyusunan laporan penelitian dalam bentuk skripsi, serta bimbingan bersama dosen pembimbing. Durasi penelitian ini dinilai memadai untuk menjamin ketelitian dalam proses analisis dan menghasilkan temuan yang akurat.[12].

Jika dibandingkan dengan penelitian lainnya, penelitian yang saya lakukan menawarkan sejumlah pembaruan. Android Studio digunakan untuk membuat sistem pemantauan dan kontrol tanaman hias anggrek, sehingga mudah digunakan di perangkat seluler. Komunikasi data waktu nyata yang dimungkinkan oleh integrasi dengan *firebase* meningkatkan efektivitas dan kecepatan pemantauan kondisi tanaman tanpa memerlukan penyegaran aplikasi. Selain itu, data lingkungan seperti suhu, kelembapan, dan tingkat irigasi disimpan dan dianalisis menggunakan *firebase realtime* Database, yang sangat memudahkan pemantauan serta proses pengambilan keputusan yang otomatis dan akurat.

2.2 Landasan Teori

2.2.1 Sistem Monitoring

Sistem monitoring, dalam bahasa Indonesia dikenal dengan istilah pemantauan. Monitoring merupakan sebuah kegiatan untuk menjamin akan tercapainya semua tujuan organisasi dan manajemen (Handoko, 1995). Dalam kesempatan lain, monitoring juga didefinisikan sebagai langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan. Dengan kata lain, monitoring merupakan salah satu proses didalam kegiatan organisasi yang sangat penting yang dapat menentukan terlaksana atau tidaknya sebuah tujuan organisasi. Tujuan dilakukannya monitoring adalah untuk memastikan agar tugas pokok organisasi dapat berjalan sesuai dengan rencana yang telah ditentukan[13].

2.2.2 Android Studio

Android Studio merupakan IDE (*Integrated Development Invirontment*) resmi yang digunakan untuk pengembangan aplikasi Android. Android secara sederhana bisa diartikan sebagai sebuah software yang digunakan pada perangkat *mobile* yang mencakup sistem operasi, *middleware*, dan aplikasi kunci yang dirilis oleh

Google. Sehingga Android mencakup keseluruhan sebuah aplikasi, mulai dari sistem operasi sampai pada pengembangan aplikasi itu sendiri[14].

2.2.3 *FireBase*

Firebase adalah suatu layanan dari Google untuk memberikan kemudahan bahkan mempermudah para developer aplikasi dalam mengembangkan aplikasinya. *Firebase* (*Backend as a Service*) merupakan solusi yang ditawarkan oleh Google untuk mempercepat pekerjaan developer[15].

2.2.4 UML *(Unified Modeling Language)*

Unified Modelling Language atau yang disingkat UML, merupakan suatu metode dalam teknik RPL (Rekayasa Perangkat Lunak) yang berfungsi untuk menggambarkan cara kerja sistem, fungsi , alur, tujuan dan juga mekanisme kontrol sistem.Terdapat empat model UML yang paling sering digunakan untuk menggambarkan suatu desain sistem *Usecase diagram*, *Class diagram*, *Behavioral Statemachine diagram*, dan juga *Sequence diagram*. Teknik-teknik pemodelan *Unified Modeling Language* ini disebut juga dengan 4 teknik dasar. Dalam proyek berorientasi objek, keempat teknik UML ini sangat mendominasi penggunaanya. *Unifield Modeling Language* merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah software yang berorientasikan pada objek. UML merupakan sebuah standar

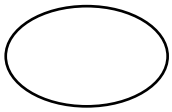
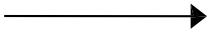
penulisan atau semacam blue print dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik[16].

Dalam perancangan sistem terdapat UML yang sering digunakan sebagai berikut:

1. UseCase Diagram

Dalam pembuatan sistem informasi, *use case diagram* merupakan pemodelan untuk behavior atau kelakuan sistem informasi. *Diagram* ini juga bersifat statis. Untuk simbol *UseCase Diagram* disajikan pada Tabel 2.1.

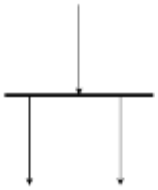
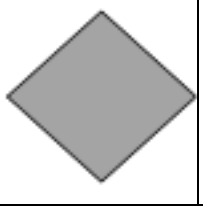
Tabel 2.1 *Use Case Diagram*

No	Simbol	Keterangan
1.		*Use case* merupakan deskripsi fungsional yang telah disediakan oleh sistem sebagai enitas yangt menghasilkan hasil yang terukur untuk suatu *actor*.
2.		*Actor* merupakan himpunan peran untuk berinteraksi dengan *usecase*
3.		*Association* merupakan garis yang menghubungkan objek satu dengan objek yang lain
4.	<<include>>	*Includ*e merupakan gambaran jika *usecase* dipanggil oleh *usecase* lain
5.		*Dependecy* merupakan garis panah yang menunjukan jka *actor* berinteraksi secara pasif
6.	<<extends >>	*Extend* merupakan gambar jika memperluas *usecase* target.

2. *Activity Diagram*

Activity Diagram merupakan diagram yang bersifat statis, yang menggambarkan aktivitas dari suatu sistem bisnis. Untuk simbol dari *diagram* aktivitas disajikan pada Tabel 2.2.

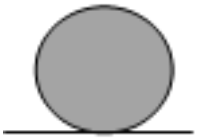
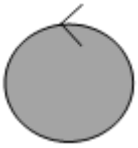
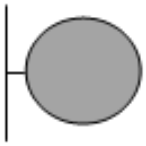
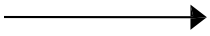
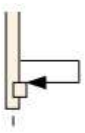
Tabel 2.2 *Activity Diagram*

No	Simbol	Keterangan
1.		*End point* atau *final Node* merupakan gambaran akhir dari suatu aktivitas
2.		*Stari Point* merupakan awal dari suatu aktivitas yang peletakannya pada pojok kiri atas
3.		*Fork* atau join digunakan untuk memarallelkan suatu kegiatan atau penggabungan 2 kegiatan parallel menjadi satu
4.		*Activity* merupakan gambaran dari suatu proses
5.		*Decision* merupakan pilihan pengambilan suatu keputusan *false or true.*

3. *Sequence Diagram*

Sequence Diagram atau *Diagram* Urutan mendeskripsikan diagram interaksi yang mengirimkan pesan dan diterima antar objek. Untuk simbol *diagram* urutan disajikan pada Tabel 2.3.

Tabel 2.3 *Sequence Diagram*

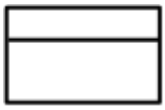
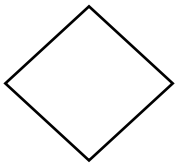
No	Simbol	Keterangan
1.		*Entity Class* merupakan bagian sistem yang membentuk deskripsi awal sistem dan berisi kumpulan kelas dalam bentuk entitas yang mendasari untuk membuat database
2.		*ControlClass* merupakan gambaran penghubung antara *Boundaryclass* dengan suatu table
3.		*Boundary Class* merupakan gambaran dari penggambaran table
4.		Pesen atau *message* menunjukkan pengiriman pesen antar *class*
5.		*Self message* menunjukkan pengiriman suatu pesan yang akan dikirim ke objek itu sendiri.

No	Simbol	Keterangan
6.		*Activation* menggambarkan suatu objek yang melakukan sebuah aksi/eksekusi operasi
7.		*Lifeline* garis titik yang terhubung ke objek disepanjang garis lifeline memiliki aktivitas

4. *Class Diagram*

Diagram Class merupakan *diagram* yang bersifat statis, dalam *diagram* ini memperlihatkan himpunan kelas,antarmuka, serta relasi. Simbol dari *Class Diagram* disajikan pada Tabel 2.4.

Tabel 2.4 *Class Diagram*

No	Simbol	Keterangan
1.		*Generallization*, merupakan dimana objek *descendent* membagikan perilaku dan struktur data objek induknya
2.		*Class*, adalah kumpulan objek yang saling berbagi.
3.		*Nary Assocation,* digunakan untuk asosiasi terhindar dengen objek lainya.

No	Simbol	Keterangan
4.		*Collaboration* merupakan deskripsi urutan aksi yang ditampilkan suatu sistem yang memiliki konsekuensi terukur bagi actor
5.		Merupakan operasi yang valid dilakukan oleh suatu objek
6.		Merupakan garis penah yang menunjukkan jika actor berinteraksi secara pasif
7.		*Association* merupakan garis yang menghubungkan objek satu dengan objek yang lain