

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian yang dilakukan oleh Asniati, Ery Muchyar Hasiri, dan Fitriani (2021) dalam jurnal penelitiannya yang berjudul PENERAPAN SISTEM KONTROL OTOMATIS *SPRAYER* PENYIRAM TANAMAN BERBASIS *ANDROID*. Tujuan penelitian ini adalah untuk membuat sistem kontrol otomatis untuk menyiram tanaman yang berbasis *Android*. Untuk memulai sistem, *NodeMCU* harus diberikan sumber arus listrik sebesar 5-9 *volt*. Setelah terhubung ke arus listrik, *NodeMCU* akan berjalan dan melakukan scan terhadap jaringan *Wifi* yang telah ditentukan. Setelah terhubung ke jaringan, *NodeMCU* membaca hasil dari sensor kelembaban dan menjalankan *relay* [3].

Penelitian yang dilakukan oleh Asniati, Ery Muchyar Hasiri, dan Fitriani (2019) dalam jurnal penelitiannya yang berjudul SISTEM KONTROL SUHU RUANGAN DAN PENYIRAMAN TANAMAN BAWANG MERAH PADA *GREENHOUSE* DENGAN *SMARTPHONE*. Penelitian ini akan menggabungkan teknologi mikrokontroler pertanian. Metode penelitian dimulai dengan merancang model *greenhouse* pintar dan kemudian membuat rangka kayu dan besi untuk rak tanaman. Selanjutnya, tahap berikutnya adalah membuat sistem kontrol otomatis untuk *greenhouse*, membuat formula dengan *Arduino Uno*, membuat pengujian suhu dan kelembaban, dan membuat kinematika manipulator rak tanaman. Tujuan

jangka panjang dari penelitian adalah untuk mencapai tujuan ini. Oleh karena itu, sistem ini dapat digunakan untuk membangun sistem pertanian terpadu dan *greenhouse* pintar [4].

Penelitian yang dilakukan oleh ANDRI SUSANTO & ISMAIL DARISMAN JAUHARI (2019) dalam jurnal penelitiannya yang berjudul RANCANG BANGUN APLIKASI *ANDROID* UNTUK KONTROL LAMPU GEDUNG MENGGUNAKAN MEDIA *BLUETOOTH* BERBASIS *ARDUINO UNO*. Sebuah penelitian yang menggunakan teknik *blackbox* ini menemukan bahwa aplikasi *Android* dapat berinteraksi dengan *arduino* dalam jarak kurang dari 12 meter. Ada tiga perintah yang dapat dilakukan oleh aplikasi yang dibuat: tombol (*ON/OFF*), pengenalan suara (mengidentifikasi suara), dan timer *countdown*. Alat ini telah diuji dan dapat digunakan sebagai pengontrol lampu untuk *smartphone Android* melalui koneksi *Bluetooth* tanpa menekan tombol lampu [5].

Penelitian yang dilakukan oleh Adriel Baruch Lantemona, Andi Patombongi (2019) dalam jurnal penelitiannya yang berjudul SISTEM KENDALI REMOTE KONTROL DENGAN ATMEGA 328 MENGGUNAKAN *SMARTPHONE*. Tujuan dari penelitian ini adalah agar pengguna dapat mengontrol mobil robot dengan bebas. Pembuatan mobil robot melibatkan pembuatan perangkat keras dan perangkat lunak. Beberapa komponen yang digunakan dalam pembuatan perangkat keras adalah *Arduino Uno*, *Modul Bluetooth*, *Modul Driver*, dan *Motor DC*. Perangkat lunak yang digunakan meliputi *Arduino IDE* dan *MIT AppInventor*. Menggunakan media

bluetooth, dua alat dikoneksikan untuk menguji sistem kendali. Aplikasi yang dikembangkan untuk pengendalian mobil robot ini menggunakan sensor akselerometer *smartphone* untuk memberikan perintah kepada mobil robot dengan memiringkannya untuk bergerak maju, mundur, belok kiri, atau kanan, dan berhenti. Dengan demikian, aplikasi remot kontrol dapat mengendalikan mobil robot dengan menggunakan *smartphone* [6].

Penelitian yang dilakukan oleh Andi Chairunnas, Triyoga Ginanjar Pamungkas (2019) dalam jurnal penelitiannya yang berjudul SISTEM KONTROL ROBOT PENYEIMBANG BERBASIS ARDUINO MENGGUNAKAN METODE PID DENGAN KOMUNIKASI *BLUETOOTH HC-05*. Tujuan dari penelitian ini adalah untuk mempelajari sistem kontrol robot penyeimbang berbasis *Arduino* yang menggunakan metode pid dan berkomunikasi dengan *bluetooth hc-05*. Sistem ini berjalan berdasarkan alat *Arduino uno*, *MPU6050*, *driver motor*, dan *bluetooth hc-05*. *Sensor MPU6050* mengumpulkan nilai tengah untuk keseimbangan robot dan menggunakan nilai ini jika nilai keseimbangan tidak sama dengan nil [7].

2.2 Landasan Teori

2.2.1 Sistem Kontrol

Sistem kontrol mengatur atau mengontrol satu atau lebih besaran (*variabel, parameter*) sehingga berada pada suatu harga atau dalam rangkuman harga tertentu [8].

Sistem kontrol adalah kumpulan cara atau metode yang dipelajari dari kebiasaan kerja manusia yang membutuhkan pengamatan kualitas dari apa yang telah mereka kerjakan sehingga memiliki karakteristik yang diharapkan pada awalnya. Menurut penjelasan ini, sistem kontrol adalah kumpulan komponen yang saling berhubungan yang bekerja secara terus menerus untuk memastikan bahwa apa yang telah mereka kerjakan memiliki karakteristik yang diharapkan.

2.2.2 Robot

Kata Kata "robot" berasal dari kata "*robota*" dalam bahasa Cek, yang berarti "pekerja". Menurut Kamus Besar Bahasa Indonesia, robot adalah alat berupa orang orangan dan sebagainya yang dapat bergerak atau berbuat seperti manusia dan dikendalikan oleh mesin. Sementara dalam arti bahasa, robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik dengan pengawasan dan kontrol manusia, atau dengan program yang telah ditetapkan sebelumnya [9].

Berdasarkan *Cangelosi* dan *Schelsinger*, Secara *historis*, *Etimologi* kata robot berasal dari kata slavia yaitu *robota*, yang digunakan untuk merujuk kepada budak atau pekerja paksa. Kata ini pertama kali muncul dalam drama R.U.R (*Rossum Universal Robots*), yang ditulis oleh Karel Capek. *Etimologi* ini menunjukkan bahwa robot dibuat untuk membantu manusia dalam pekerjaan sehari-hari

dan, dalam beberapa kasus, untuk menggantikan manusia di industri robotika.

2.2.3 Bawang Merah

Allium cepa var. *Aggregatum* L, juga dikenal sebagai bawang merah, adalah sayuran umbi yang dapat digunakan untuk berbagai tujuan, seperti sebagai bumbu untuk makanan, sayuran, dan obat tradisional. itu memiliki sifat anti septik karena mengandung senyawa anilin dan alisin. untuk mendapatkan hasil bawang terbaik, beberapa faktor penting adalah lingkungan yang baik, jumlah cahaya, air, dan hara yang memadai. Pengairan yang berlebihan dapat menyebabkan kelembaban tanah menjadi tinggi, yang menghambat pertumbuhan umbi bawang dan menyebabkan bawang merah menjadi busuk [10].

Untuk tumbuh dengan baik, tanaman bawang merah membutuhkan pencahayaan matahari sebanyak 70% atau lebih, suhu udara antara 25–32°C, dan kelembaban nisbi sebanyak 50-70%. Tanaman bawang merah dapat membentuk umbi di tempat dengan suhu udara rata-rata 22 derajat *Celcius*, tetapi jika suhu turun di bawah 22 derajat *Celcius*, tanaman bawang merah tidak akan membentuk umbi. Untuk tanaman sawah Bawang Merah dapat dilihat seperti pada Gambar 2.1



Gambar 2.1 Tanaman Bawang Merah

2.2.4 *Android*

Android, sistem operasi berbasis *Linux*, dimaksudkan untuk mengembangkan *smartphone* dan *tablet* serta perangkat seluler layar sentuh lainnya. *Android* memberi pengembang platform terbuka untuk membuat aplikasi untuk berbagai piranti gerak. Salah satu alasan sistem operasi *Android* begitu populer di pasar dan berkembang dengan cepat adalah penggunaan bahasa pemrograman *Java* dan keunggulannya sebagai *software* yang menggunakan baris kode komputer yang didistribusikan secara terbuka, memungkinkan pengguna membuat aplikasi baru [11]. Untuk logo *Android* dapat dilihat seperti pada Gambar 2.2



Gambar 2.2 *Android*

2.2.5 *Smartphone*

Smartphone adalah telepon genggam yang memiliki fitur dan fungsi yang mirip dengan komputer. Bagi beberapa orang, ponsel cerdas hanyalah sebuah telepon dengan fitur canggih seperti surel (surat elektronik), *internet*, dan kemampuan membaca buku elektronik (*e-book*), atau terdapat papan ketik dan penyambung VGA. Bagi orang lain, ponsel cerdas hanyalah komputer khusus dengan perangkat lunak sistem operasi yang memungkinkan pengembang aplikasi untuk berkomunikasi dengannya [12]. Untuk logo *Smartphone* dapat dilihat seperti pada Gambar 2.3



Gambar 2.3 *Smartphone*

2.2.6 *Bluetooth*

Bluetooth adalah teknologi yang memungkinkan dua perangkat yang kompatibel, seperti telepon dan *PC*, berkomunikasi tanpa kabel tanpa saluran. Teknologi ini mengubah *Bluetooth* menjadi standar industri untuk jaringan wilayah pribadi nirkabel (WPAN). Dengan menggunakan *tranceiver frekuensi hopping*, *Bluetooth* beroperasi

pada frekuensi 2,4Ghz dan memungkinkan *host-host Bluetooth* berkomunikasi satu sama lain secara langsung dengan data dan suara. Salah satu kelemahan teknologi ini adalah jangkauannya yang pendek dan rendahnya kapasitas untuk mengirimkan data [13].

Pada dasarnya *Bluetooth* Pada dasarnya, *Bluetooth* memungkinkan teknologi mobile nirkabel dengan harga terjangkau dan fitur hemat daya. *Smartphone* yang kita gunakan saat ini juga memiliki koneksi *Bluetooth* untuk pengiriman data file, mendengarkan lagu secara nirkabel, dan berkomunikasi dengan *developer* aplikasi. Aplikasi perancangan yang diprogram memerlukan koneksi *Bluetooth* untuk mengirimkan perintah pengiriman data, dan mikrokontroler yang memiliki koneksi *Bluetooth* menerima perintah dan memprosesnya untuk melakukan kontrol. Untuk logo *Bluetooth* dapat dilihat seperti pada Gambar 2.4



Gambar 2.4 *Bluetooth*

2.2.7 *Google Firebase*

Google Firebase adalah platform yang menawarkan layanan yang membuat pengembangan aplikasi berbasis *web* maupun *Android* lebih mudah. Penelitian ini menggunakan *database* *Firebase* karena

tidak memerlukan sistem yang signifikan dan data dapat ditampilkan secara real-time di halaman *website* [14]. Untuk logo *Google Firebase* dapat dilihat seperti pada Gambar 2.5



Gambar 2.5 *Google Firebase*

2.2.8 *MIT App Inventor*

MIT App Inventor adalah aplikasi yang dirilis oleh *Google Labs* yang berjalan pada sistem *Android*. Aplikasi *App Inventor* dibuat pada tanggal 12 Juli 2010, dan dirilis pada tanggal 15 Desember 2010. Pada tanggal 31 Desember 2011, *Google* meninggalkan proyek *App Inventor*, dan sekarang disebut *MIT App Inventor* [15]. Untuk logo *MIT app Inventor* dapat dilihat seperti pada Gambar 2.6



Gambar 2.6 Tampilan awal *Mit App Inventor*

2.2.9 *Unified Modeling Language (UML)*

Bahasa pemodelan perangkat lunak *Unified Modeling Language (UML)* adalah sebuah bahasa yang telah distandardisasi

sebagai media penulisan cetak biru perangkat lunak (*Pressman*). UML dapat digunakan untuk visualisasi, spesifikasi, konstruksi, dan dokumentasi beberapa bagian sistem perangkat lunak yang ada. Dengan kata lain, arsitek perangkat lunak membuat diagram UML untuk membantu programmer dan pengembang membangun perangkat lunak, sama seperti arsitek membuat dokumen cetak biru untuk perusahaan konstruksi [16].

1. *Use Case Diagram*

Sebuah *Use Case diagram* dapat menunjukkan beberapa jalur interaksi manusia dengan sistem, dengan setiap jalur disebut skenario. Untuk simbol *Use Case diagram* dapat dilihat seperti pada Tabel 2.1

Tabel 2.1 Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>)

NO	GAMBAR	NAMA	KETERANGAN
			akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<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>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara

NO	GAMBAR	NAMA	KETERANGAN
			objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menunjukkan alur kerja atau aktivitas sistem dalam perangkat lunak. Untuk simbol *Activity Diagram* dapat dilihat seperti pada Tabel 2.2

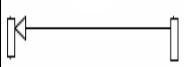
Tabel 2.2 Simbol Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

3. Diagram Urutan (Sequence Diagram)

Sequence Diagram bersifat dinamis dan menampilkan aktifitas objek berdasarkan urutan waktu. *Sequence Diagram* juga menunjukkan objek dan arus pesan antara satu sama lain pada Objek *Use Case*. Untuk simbol *Sequence Diagram* dapat dilihat seperti pada Tabel 2.3

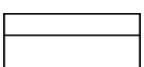
Tabel 2.3 Simbol *Sequence Diagram*

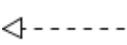
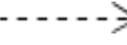
NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

4. *Class Diagram*

Class Diagram adalah model statis yang memungkinkan visualisasi data dan informasi dari sistem secara keseluruhan. *Diagram* ini dikaitkan dengan struktur basis data sistem dan juga dapat digunakan sebagai pengganti ERD dalam proses penggambaran diagram rekayasa perangkat lunak yang biasa. Untuk simbol *Sequence Diagram* dapat dilihat seperti pada Tabel 2.4

Tabel 2.4 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<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>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi

NO	GAMBAR	NAMA	KETERANGAN
			atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		<i>Association</i>	Apa yang menghubungkan antara

NO	GAMBAR	NAMA	KETERANGAN
			objek satu dengan objek lainnya.

