

**RANCANG BANGUN SISTEM *MONITORING* DAN *CONTROLLING*
KEBUN ORGANIK BERBASIS INTERNET OF THINGS DENGAN
METODE *PROTOTYPING* DAN *USABILITY TESTING*
(STUDI KASUS: NARA KUPU YOGYAKARTA)**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1
Program Studi Teknik Industri - Fakultas Teknologi Industri
Universitas Islam Indonesia**



Nama : Irfan Rasyid Jatmiko

No. Mahasiswa 19522048

**PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2023

HALAMAN PERNYATAAN KEASLIAN

Saya mengakui bahwa tugas akhir ini adalah hasil karya saya sendiri kecuali kutipan dan ringkasan yang seluruhnya sudah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah maka saya bersedia ijazah yang telah saya terima ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 17 Agustus 2023



(Irfan Rasyid Jatmiko)
19522048

HALAMAN SURAT BUKTI PENELITIAN

SURAT KETERANGAN PENELITIAN

Assalamu'alaikum Warohmatullahi Wabarakaatuh

Dengan hormat,

Yang bertanda tangan dibawah ini, menerangkan bahwa:

Nama : Irfan Rasyid Jatmiko

NIM : 19522048

Jurusan : Teknik Industri

Dosen Pembimbing : Ir. Abdullah 'Azzam S.T., M.T., IPM

Menyatakan bahwa mahasiswa tersebut diatas telah melaksanakan penelitian tugas akhir dengan judul "RANCANG BANGUN SISTEM MONITORING DAN CONTROLLING KEBUN ORGANIK BERBASIS INTERNET OF THINGS DENGAN METODE PROTOTYPING DAN USABILITY TESTING (STUDI KASUS: NARAKUPU YOGYAKARTA)" Mulai pelaksanaan penelitian 11 Februari 2023 sampai 25 Juli 2023.

Demikian surat keterangan penelitian ini kami buat. Atas perhatiannya dan kerja samanya kami mengucapkan terima kasih.

Wassalamu'alaikum Warohmatullahi Wabarokaatuh.

Yogyakarta, 16 Agustus 2023

Pimpinan Nara Kupu Yogyakarta


(Shelli Nara Kupu Orjir)

LEMBAR PENGESAHAN PEMBIMBING

**RANCANG BANGUN SISTEM *MONITORING* DAN *CONTROLLING*
KEBUN ORGANIK BERBASIS INTERNET OF THINGS DENGAN
METODE *PROTOTYPING* DAN *USABILITY TESTING***



Yogyakarta, 15 Maret 2023

Dosen Pembimbing

(Ir. Abdullah Azzam, S.T., M.T., IPM.)

LEMBAR PENGESAHAN DOSEN PENGUJI
RANCANG BANGUN SISTEM *MONITORING* DAN *CONTROLLING* KEBUN
ORGANIK BERBASIS *INTERNET OF THINGS* DENGAN METODE
PROTOTYPING* DAN *USABILITY TESTING
(STUDI KASUS: NARA KUPU YOGYAKARTA)

TUGAS AKHIR

Disusun Oleh :

Nama : Irfan Rasyid Jatmiko

No. Mahasiswa : 19 522 048

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata-1 Teknik Industri Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta, 23 Agustus 2023

Tim Penguji

Ir. Abdullah 'Azzam, S.T., M.T., IPM.

Ketua

Annisa Uswatun Khasanah, S.T., M.sc.

Anggota I

Dr. Eng. Meilinda Fitriani Nur Maghfiroh,

S.T., MBA.

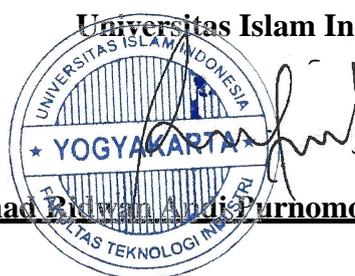
Anggota II

Mengetahui,

Ketua Program Studi Teknik Industri

Fakultas Teknologi Industri

Universitas Islam Indonesia



Ir. Muhammad Ridwan Ardi Purnomo, S.T., M.Sc., Ph.D., IPM.

HALAMAN PERSEMBAHAN*Bismillahirrahmanirrahim*

Puji syukur saya panjatkan kepada Allah SWT atas Rahmat-Nya yang Maha Pengasih Lagi Maha Penyayang saya diberikan kelancaran dalam menyelesaikan tugas akhir. Dengan tulus dan rasa hormat, laporan Tugas Akhir ini disusun sebagai tanda penghormatan kepada keluarga tercinta, Ayah saya Nugroho Widyanto, Ibu saya Erna Hartatiek Subrata, dosen pembimbing yang terhormat Bapak Ir. Abdullah 'Azzam, S.T., M.T., IPM. serta teman-teman sejawat yang telah memberikan dukungan tiada henti selama proses penulisan ini.

HALAMAN MOTTO

"Dan sesungguhnya, setiap kesulitan pasti ada kemudahan."

(Al-Quran, Surah Al-Insyirah: 5-6)

Artinya: Allah menjanjikan bahwa setiap kesulitan yang kita hadapi pasti akan diikuti oleh kemudahan. Hal ini memberikan inspirasi kepada penulis untuk tetap tabah dan optimis dalam menghadapi tantangan.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, Tuhan Yang Maha Esa, yang telah melimpahkan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan laporan Tugas Akhir (TA) ini. Laporan TA ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik dari Fakultas Teknik Industri, Universitas Islam Indonesia.

Penulisan laporan TA ini bertujuan untuk mengimplementasikan dan mengembangkan sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) dengan metode *Prototyping* dan *Usability Testing*. Studi kasus yang diambil adalah Nara Kupu Yogyakarta. Laporan ini mencakup tahap perancangan, pengembangan, dan pengujian sistem yang dilakukan dengan kerja keras dan dedikasi.

Selama pelaksanaan TA ini, penulis tidak dapat menutup mata akan adanya berbagai kesulitan dan tantangan yang dihadapi. Namun, dengan dukungan, bimbingan, dan bantuan dari berbagai pihak, penulis berhasil mengatasi setiap hambatan tersebut. Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M. T., IPU, ASEAN.Eng. selaku Dekan Fakultas Teknik Industri Universitas Islam Indonesia atas kesempatan yang diberikan dan fasilitas yang disediakan.
2. Bapak Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM. selaku Ketua Program Studi Teknik Industri atas arahan dan pengawasannya dalam proses penyelesaian TA ini.
3. Bapak Ir. Abdullah 'Azzam, S.T., M.T., IPM. selaku Dosen Pembimbing Tugas Akhir atas bimbingan, nasihat, dan waktu yang diberikan dalam membimbing penulis selama penyusunan TA ini.
4. Pimpinan perusahaan Nara Kupu Yogyakarta atas izin dan kerjasama dalam melakukan studi kasus di tempat tersebut.
5. Pembimbing lapangan yang telah memberikan petunjuk dan arahan yang berharga selama penelitian dilakukan di lapangan.
6. Kedua orang tua penulis yang selalu memberikan dukungan, doa, dan semangat dalam perjalanan penulisan ini.
7. Orang-orang terdekat dan teman-teman yang memberikan dukungan moral dan motivasi.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan dan keterbatasan. Oleh karena itu, penulis mengharapkan kritik, saran, dan masukan yang membangun untuk perbaikan kedepannya.

Akhir kata, semoga laporan TA ini dapat memberikan manfaat dan kontribusi dalam pengembangan sistem *monitoring* dan *controlling* kebun organik berbasis IoT. Semoga penulisan ini juga dapat menjadi pijakan bagi penelitian-penelitian yang lebih lanjut di bidang yang sama. Penulis berharap agar apa yang telah penulis lakukan dapat bermanfaat bagi semua pihak yang terlibat dan untuk kemajuan ilmu pengetahuan.

Wassalamu'alaikum warahmatullahi wabarakatuh.

Yogyakarta, 17 Agustus 2023

A handwritten signature in black ink, appearing to read 'Irfan Rasyid Jatmiko', written over the printed name below it.

Irfan Rasyid Jatmiko

ABSTRAK

Sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) telah dirancang dan dibangun dengan menggunakan metode *Prototyping* dan *Usability Testing*. Penelitian ini bertujuan untuk memudahkan pengelola kebun organik dalam melakukan proses perawatan yang efisien dan akurat. Studi kasus yang digunakan adalah kebun organik Nara Kupu Yogyakarta. Tantangan pengelolaan kebun organik Nara Kupu Yogyakarta terletak pada metode konvensional yang masih digunakan dalam proses perawatan. Hal ini mengakibatkan rendahnya intensitas perawatan seperti penyiraman air, pemberian pupuk, dan pemberian pestisida. Akibatnya, tanaman menjadi rentan terhadap serangan hama dan penyakit, serta mengalami kekeringan, kehilangan kelembaban, kekurangan nutrisi, dan kerapuhan akar tanaman. Dampaknya, terjadi penurunan hasil panen sebesar 10-25%. Dalam penelitian ini, sistem *monitoring* dan *controlling* berbasis IoT dirancang dengan menggunakan *Microcontroller* ESP32, sensor kelembaban tanah, sensor suhu tanah, sensor *Ultrasonic*, pompa air, LCD, dan aplikasi *Blynk*. Sistem ini memungkinkan pengelola kebun untuk memantau kondisi tanaman secara *real-time* dan mengontrol pengiriman air dan nutrisi secara otomatis. Hasil pengujian menunjukkan bahwa sistem ini mampu memantau kelembaban tanah, suhu tanah, dan tinggi air secara akurat. Selain itu, sistem juga memberikan notifikasi *real-time* melalui aplikasi *Blynk* jika terdapat kondisi yang perlu diperhatikan. Dengan adanya sistem *monitoring* dan *controlling* ini, diharapkan proses pengelola kebun dapat dilakukan dengan efisien dan menyelesaikan hambatan-hambatan yang ada ketika menggunakan metode konvensional.

Kata Kunci: Sistem *monitoring* dan *controlling*, *Internet of Things*, Kebun Organik, *Prototyping*, *Usability Testing*

DAFTAR ISI

HALAMAN JUDUL	1
HALAMAN PERNYATAAN KEASLIAN	ii
HALAMAN SURAT BUKTI PENELITIAN	iii
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO	vii
KATA PENGANTAR	viii
ABSTRAK	x
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xvi
BAB I PENDAHULUAN.....	1
1. 1 Latar Belakang	1
1. 2 Rumusan Masalah	6
1. 3 Tujuan Penelitian	6
1. 4 Manfaat Penelitian	6
1. 5 Batasan Penelitian	7
1. 6 Sistematika Penelitian	7
BAB II.....	9
2. 1 Kajian Literatur	9
2.2 Landasan Teori.....	14
2.2.1 Kebun Organik	14
2.2.2 Monitoring Kebun Organik.....	15
2.2.3 Controlling Kebun Organik.....	16
2.2.4 Internet of Things (IoT)	18
2.2.5 Automasi	20
2.2.6 ESP32.....	21
2.2.7 Arduino IDE.....	23
2.2.8 Pompa Air	24
2.2.9 Sensor.....	24
2.2.10 LCD.....	28
2.2.11 Blynk.....	28
2.2.12 Lingkungan Kerja Fisik (LKF)	29
2.2.13 Black Box Testing.....	31

2.2.14 Metode Prototyping.....	31
2.2.15 Metode Usability Testing.....	33
2.2.16 System Usability Scale (SUS).....	34
BAB III METODE PENELITIAN.....	36
3.1 Kerangka Penelitian	36
3.2 Obyek Penelitian	38
3.3 Subjek Penelitian.....	38
3.4 Alat Bantu Analisis Data.....	39
3.5 Jenis Data	40
3.6 Metode Pengumpulan data.....	40
3.7 Alur Penelitian	41
3.8 Tahapan penelitian	43
3.9 Pembangunan Model dan Perancangan	45
3.10 Rumus dan Teori.....	46
BAB IV PEMBANGUNAN SISTEM.....	48
4.1 Analisis kebutuhan	48
4.1.1 Kebutuhan Pengguna	48
4.1.2 Kebutuhan Dasar Sistem.....	49
4.1.3 Pengambilan Data Lingkungan Kerja Fisik (LKF) Kebun Organik	50
4.1.4 Kebutuhan <i>Software</i>	52
4.1.5 Kebutuhan <i>Hardware</i>	52
4.2 Perancangan <i>Prototype</i>	53
4.3.1 Perancangan Alur <i>Hardware</i> pada <i>Prototype</i>	53
4.3.2 Perancangan Fitur <i>Prototyping</i>	54
4.2.3 Perancangan <i>Input</i>	56
4.2.4 Perancangan <i>Output</i>	56
4.3 Bangun <i>Prototyping</i>	57
4.3.1 Pengaturan Awal <i>Blynk Cloud</i>	57
4.3.2 Pembangunan <i>Hardware</i>	60
4.3.3 Pembangunan <i>Software</i>	71
4.4 Evaluasi Pengguna	87
4.4.1 <i>Black Box Testing</i>	89
4.5 Implementasi Produk	91
4.5.1 Pengaturan Tampilan <i>Blynk</i> pada <i>Smartphone</i>	91
4.5.2 Penyiraman Air Rutin	93

4.5.3	Penyiraman Air ketika Suhu Melebihi Batas	95
4.5.4	Penyiraman Air ketika Kelembaban kurang dari Batas	97
4.5.5	Penyiraman Pupuk Cair.....	98
4.5.6	Penyiraman Pestisida	100
4.5.7	Pemantauan Bak Air	103
4.5.9	Pemantauan Bak Pestisida.....	106
4.5.10	Rangkaian Sistem Keseluruhan.....	108
4.5.11	Pembangunan Tata Letak Sistem	122
4.6	<i>Usability Testing</i>	127
BAB V PENGUJIAN SISTEM DAN PEMBAHASAN.....		132
5.1	Pengujian sistem.....	132
5.1.1	<i>Usability Testing</i> dengan <i>System Usability Scale (SUS)</i>	132
5.2	Pembahasan Lingkungan Kerja Fisik (LKF)	134
5.2.1	Perbandingan Skala <i>Prototype</i>	134
5.2.2	Proyeksi Peletakkan Sensor	135
5.2.3	Rekayasa Perbandingan Hasil Panen	136
BAB VI KESIMPULAN.....		139
6.1	Kesimpulan	139
6.2	Saran.....	139
LAMPIRAN		1

DAFTAR TABEL

Tabel 2. 1 Perbandingan Penelitian Sejenis	13
Tabel 2. 2 Spesifikasi <i>Microcontroller</i> ESP32	23
Tabel 2. 3 Spesifikasi Sensor <i>Ultrasonic</i> HC-SR04	27
Tabel 3. 1 Kerangka Penelitian	36
Tabel 3. 2 Karakteristik dan Kriteria	39
Tabel 3. 3 Kuesioner <i>System Usability Scale</i> (SUS).....	46
Tabel 4. 1 Kebutuhan Pengguna	48
Tabel 4. 2 Pembatasan Suhu dan Kelembaban Tanah	49
Tabel 4. 3 Penjadwalan Waktu Penyiraman Air	49
Tabel 4. 4 Kebutuhan <i>Software</i>	52
Tabel 4. 5 Kebutuhan <i>Hardware</i>	52
Tabel 4. 6 Rangkaian ESP32	63
Tabel 4. 7 <i>Pseudocode</i> Konfigurasi Awal	72
Tabel 4. 8 <i>Pseudocode</i> mendefinisikan variabel.....	73
Tabel 4. 9 <i>Pseudocode</i> pengaturan sistem	74
Tabel 4. 10 <i>Pseudocode</i> aplikasi <i>Blynk</i>	75
Tabel 4. 11 <i>Pseudocode</i> penjadwalan waktu	76
Tabel 4. 12 <i>Pseudocode</i> LCD	77
Tabel 4. 13 <i>Pseudocode</i> Sensor suhu tanah	78
Tabel 4. 14 <i>Pseudocode</i> sensor kelembaban tanah.....	79
Tabel 4. 15 <i>Pseudocode</i> sensor <i>Ultrasonic</i>	79
Tabel 4. 16 <i>Pseudocode</i> fungsi utama	80
Tabel 4. 17 <i>Black Box Testing</i>	89
Tabel 4. 18 Jam penyiraman pupuk cair dan pestisida	90
Tabel 4. 19 Jam penyiraman pupuk cair dan pestisida sesudah di evaluasi.....	90
Tabel 4. 20 Daftar <i>hardware</i> fitur penyiraman air rutin.	93
Tabel 4. 21 <i>Pseudocode</i> fitur penyiraman rutin.....	94
Tabel 4. 22 Daftar <i>hardware</i> fitur penyiraman ketika suhu melebihi batas.	95
Tabel 4. 23 <i>Pseudocode</i> penyiraman air ketika suhu melebihi batas.	96
Tabel 4. 24 Daftar <i>hardware</i> fitur penyiraman air.....	97
Tabel 4. 25 <i>Pseudocode</i> penyiraman air ketika kelembaban kurang dari batas.	97
Tabel 4. 26 Daftar <i>hardware</i> fitur penyiraman pupuk cair.....	98
Tabel 4. 27 <i>Pseudocode</i> penyiraman pupuk cair.	99
Tabel 4. 28 Daftar <i>hardware</i> fitur penyiraman pestisida.....	100
Tabel 4. 29 <i>Pseudocode</i> penyiraman pestisida.	101
Tabel 4. 30 Daftar <i>hardware</i> fitur pemantauan bak air.	103
Tabel 4. 31 <i>Pseudocode</i> pemantauan bak air.....	104
Tabel 4. 32 Daftar <i>hardware</i> fitur pemantauan bak pupuk cair.....	105
Tabel 4. 33 <i>Pseudocode</i> pemantauan bak pupuk cair.....	106
Tabel 4. 34 Daftar <i>hardware</i> fitur pemantauan bak pestisida.....	106
Tabel 4. 35 <i>Pseudocode</i> pemantauan bak pestisida.....	107
Tabel 4. 36 Rangkaian Keseluruhan	108

Tabel 4. 37	109
Tabel 4. 38 Keterangan Proyeksi Peletakkan Sistem	126
Tabel 4. 39 Hasil Kuesioner Responden Pertama.....	127
Tabel 4. 40 Hasil Kuesioner Responden kedua	128
Tabel 4. 41 Hasil Kuesioner Responden Ketiga	129
Tabel 4. 42 Hasil Kuesioner Responden keempat	129
Tabel 5. 1 Analisis rata-rata jawaban responden	132
Tabel 5. 2 Pertanyaan dengan skor terbaik	133
Tabel 5. 3 Pertanyaan dengan skor terburuk.....	134
Tabel 5. 4 Perbandingan Skala <i>Prototype</i>	135
Tabel 5. 5 Perbedaan antara pengelolaan kebun.....	138

DAFTAR GAMBAR

Gambar 2. 1 Elemen Dasar Automasi.....	20
Gambar 2. 2 <i>Microcontroller</i> ESP32.....	22
Gambar 2. 3 <i>Software</i> Arduino IE.....	24
Gambar 2. 4 Sensor Suhu Tanah.....	26
Gambar 2. 5 Sensor Kelembaban Tanah.....	26
Gambar 2. 6 Sensor <i>Ultrasonic</i> HC-SR04.....	27
Gambar 2. 7 Liquid Crystal Display (LCD).....	28
Gambar 2. 8 Logo <i>Software Blynk</i>	29
Gambar 3. 1 Kebun Nara Kupu Yogyakarta.....	38
Gambar 3. 2 Alur Penelitian.....	42
Gambar 4. 1 Hasil Pengambilan Data LKF Suhu Tanah.....	51
Gambar 4. 2 Hasil Pengambilan Data LKF Kelembaban Tanah.....	51
Gambar 4. 3 <i>flowchart</i> Perancangan Sistem.....	54
Gambar 4. 4 Form Pendaftaran <i>Blynk</i>	57
Gambar 4. 5 Form Pendaftaran <i>Blynk</i>	58
Gambar 4. 6 Form Create New Templates.....	58
Gambar 4. 7 Pemilihan <i>New Device</i>	59
Gambar 4. 8 Tambahkan <i>New Device</i>	59
Gambar 4. 9 Mengatur <i>Template</i>	60
Gambar 4. 10 Membuat <i>Datastream</i>	60
Gambar 4. 11 Tampak Tertutup <i>Housing</i> Modul Utama.....	61
Gambar 4. 12 Tampak Terbuka <i>Housing</i> Modul Utama.....	61
Gambar 4.13 <i>Microcontroller</i> ESP32.....	62
Gambar 4. 14 Rangkaian ESP32.....	62
Gambar 4. 15 Tampilan pertama LCD saat sistem dinyalakan.....	63
Gambar 4. 16 Tampilan LCD pertama.....	64
Gambar 4. 17 Tampilan LCD kedua.....	64
Gambar 4. 18 Rangkaian LCD pada sistem.....	65
Gambar 4. 19 Sensor suhu tanah.....	65
Gambar 4. 20 Rangkaian sensor suhu tanah pada sistem.....	66
Gambar 4. 21 Sensor kelembaban tanah.....	66
Gambar 4. 22 Rangkaian sensor kelembaban tanah pada sistem.....	67
Gambar 4. 23 <i>Relay 4 Channel</i>	67
Gambar 4. 24 Pompa Air.....	68
Gambar 4. 25 Rangkaian <i>Relay 4 Channel</i> dan Pompa air pada sistem.....	68
Gambar 4. 26 <i>Adapter 12V</i>	69
Gambar 4. 27 Sensor <i>Ultrasonic</i>	69
Gambar 4. 28 Rangkaian sensor <i>Ultrasonic</i>	70
Gambar 4. 29 Bak Penampung.....	71
Gambar 4. 30 <i>Adapter 5V</i>	71
Gambar 4. 31 Wawancara dengan Pengelola Kebun Organik Nara Kupu Yogyakarta.....	88
Gambar 4. 32 Peneliti melakukan Presentasi <i>Prototype</i> yang telah dibuat.....	88
Gambar 4. 33 Presentasi Alat pada Pengelola Kebun Organik Nara Kupu Yogyakarta.....	89
Gambar 4. 34 Membuka menu pengaturan.....	91
Gambar 4. 35 membuka menu <i>Widget Box</i>	91

Gambar 4. 36 Menambahkan <i>Widget Gauge</i>	92
Gambar 4. 37 Menambahkan <i>Widget Button</i>	92
Gambar 4. 38 Tampilan Aplikasi <i>Blynk</i> di <i>Smartphone</i>	93
Gambar 4. 39 Rangkaian <i>Hardware</i> dalam bentuk <i>flowchart</i>	94
Gambar 4. 40 Rangkaian <i>Hardware</i> dalam bentuk <i>flowchart</i>	96
Gambar 4. 41 Rangkaian <i>Hardware</i> dalam bentuk <i>flowchart</i>	97
Gambar 4. 42 Rangkaian <i>Hardware</i> dalam bentuk <i>flowchart</i>	99
Gambar 4. 43 Rangkaian <i>Hardware</i> dalam bentuk <i>flowchart</i>	101
Gambar 4. 44 Rangkaian <i>Hardware</i> dalam bentuk <i>flowchart</i>	104
Gambar 4. 45 Rangkaian <i>Hardware</i> dalam bentuk <i>flowchart</i>	105
Gambar 4. 46 Rangkaian <i>Hardware</i> dalam bentuk <i>flowchart</i>	107
Gambar 4. 47 Rangkaian Sistem Keseluruhan	108
Gambar 4. 48 Desain 3D <i>Prototype</i>	123
Gambar 4. 49 Gambar Teknik <i>Prototype</i>	124
Gambar 4. 50 Hasil Final <i>Prototype</i>	124
Gambar 4. 51 Proyeksi Pertama	125
Gambar 4. 52 Proyeksi Kedua	125
Gambar 4. 53 Proyeksi Ketiga	125
Gambar 4. 54 Proyeksi Keempat	126
Gambar 4. 55 Pengisian Kuesioner <i>System Usability Scale (SUS)</i>	127
Gambar 5. 1 Proyeksi Peletakkan Sensor Suhu Tanah	135
Gambar 5. 2 Proyeksi Peletakkan Sensor Kelembaban Tanah	136

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengembangan pertanian organik di Indonesia tumbuh pesat dan dianggap sebagai potensi besar untuk meningkatkan ketahanan pangan dan ekonomi negara. Namun menurut Ramakant et al. (2023) perkembangan pertanian organik di Indonesia masih menghadapi hambatan, yang salah satunya adalah kurangnya penerapan teknologi. Di samping aspek teknologi, kendala lainnya adalah tantangan yang melibatkan faktor manusia, seperti keterlambatan dalam tindakan manusia, baik dalam penyiraman, pemberian pupuk, atau penggunaan pestisida, yang akan berdampak langsung pada kondisi tanaman.

Menurut Nayagam et al. (2023), tanaman yang tidak mendapatkan air atau nutrisi yang tepat waktu dapat mengalami pertumbuhan yang terhambat, kerusakan tanaman sehingga berpengaruh pada hasil panen. Oleh karena itu, perlunya menjaga jadwal yang tepat dan konsisten dalam merawat tanaman menjadi suatu keharusan. Dalam rangka mengatasi berbagai tantangan ini, solusi yang ditawarkan oleh *Internet of Things* (IoT) dapat memberikan kontribusi yang signifikan. Konsep ini, sebagaimana dijelaskan oleh Kumar (2021), memungkinkan objek-objek fisik seperti sensor, perangkat elektronik, dan peralatan komunikasi untuk terhubung dan saling berkomunikasi dengan internet. Dengan menerapkan IoT dalam pertanian organik, pengaturan jadwal perawatan tanaman seperti menyiram, pemberian pupuk, dan pemberian pestisida dapat dilakukan secara automasi sesuai dengan pembacaan sensor-sensor IoT yang mengukur kondisi lingkungan dan tanaman secara *real-time*.

Untuk memastikan bahwa tanaman menerima perawatan yang optimal sesuai dengan kondisi aktual, menghindari keterlambatan dan meminimalkan risiko terjadinya kondisi yang tidak menguntungkan bagi pertumbuhan. Selain itu, penggunaan teknologi *Internet of Things* (IoT) dalam melakukan pertanian termasuk dalam pengembangan *Smart Agriculture*. Menurut Abraham (2021), *Smart Agriculture* sendiri merupakan penggunaan teknologi informasi dan komunikasi dalam produksi pertanian untuk meningkatkan efisiensi dan produktivitas. Menurut Ervan Septa et al. (2022), meskipun *Smart Agriculture* memiliki potensi yang besar, penerapannya masih tergolong jarang di Negara Indonesia.

Tantangan tersebut sedang dihadapi oleh kebun organik Nara Kupu Yogyakarta. Berdasarkan dari hasil observasi dan wawancara, para pengelola kebun masih

menggunakan metode konvensional dalam melakukan proses perawatan. Selain itu, para pengelola kebun juga memanen, mengemas, melayani pembeli, dan melayani tamu. Kebun organik Nara Kupu Yogyakarta merupakan kebun edukasi, sehingga sering didatangi oleh tamu yang berjumlah banyak. Hal ini menyebabkan intensitas perawatan seperti penyiraman air, pemberian pupuk, dan pemberian pestisida menurun.

Menurut Nayagam et al. (2023), hal tersebut menyebabkan tanaman menjadi rentan terhadap serangan hama dan penyakit, mengalami kekeringan, kehilangan kelembaban, kekurangan nutrisi, dan akar tanaman menjadi rapuh. Dampak dari hal tersebut adalah setiap melakukan panen terdapat sekitar 15 – 30% hasil yang gagal. Akibatnya, terjadi penurunan pada pemasukan kebun organik Nara Kupu.

Setelah mengetahui hal tersebut, peneliti menerapkan pendekatan Lingkungan Kerja Fisik (LKF) yang menurut Zahwa (2023), digunakan untuk mengkaji semua keadaan berbentuk fisik yang terdapat di fasilitas kerja. Pada penelitian ini, Lingkungan Kerja Fisik (LKF) digunakan untuk mengkaji faktor-faktor yang bisa menyebabkan kegagalan panen tersebut di fasilitas kebun organik Nara Kupu Yogyakarta. Oleh karena, itu peneliti melakukan observasi langsung di Kebun Organik Nara Kupu Yogyakarta selama satu minggu mengenai elemen-elemen fisik yang dapat mempengaruhi produktivitas pertumbuhan tanaman seperti suhu tanah dan kelembaban tanah. Dua elemen tersebut dipilih karena menurut Onwuka (2019), kondisi tanah merupakan hal yang tak tergantikan untuk menjaga kehidupan tumbuhan, memberikan dukungan mekanis, serta menyediakan nutrisi dan air. Menurut Lysenko et al. (2023), suhu dan kelembaban tanah menjadi katalisator bagi banyak proses biologis, mempengaruhi aerasi, dan ketersediaan nutrisi tumbuhan yang penting bagi pertumbuhan tanaman.

Selanjutnya peneliti melakukan evaluasi Lingkungan Kerja Fisik (LKF) yang kemudian dibandingkan dengan kondisi optimal untuk pertumbuhan tanaman. Hasil yang didapatkan adalah pada kebun organik Nara Kupu Yogyakarta, saat jam-jam tertentu (Sekitar pukul 10.00 – 15.00), ketika cuaca sedang cerah dan sinar matahari terik, didapatkan nilai suhu tanah sekitar 33 hingga paling tingginya sebesar 35°C dan nilai kelembaban tanah sekitar 40 hingga 48%. Sedangkan menurut Asih (2018), nilai yang paling optimal untuk pertumbuhan tanaman adalah sekitar 29 hingga 31°C untuk suhu tanah dan dibawah 50% untuk kelembaban tanah. Hasil yang didapat membuktikan bahwa proses perawatan kebun organik secara konvensional masih belum optimal untuk produktivitas pertumbuhan tanaman dan dibutuhkannya sebuah sistem *monitoring* dan

controlling yang bisa menyesuaikan kondisi lingkungan pertumbuhan tanaman secara akurat, *real-time*, dan efisien agar proses pertumbuhan tanaman berjalan secara optimal.

Sebelumnya telah terdapat penelitian terdahulu yang mengangkat topik penerapan teknologi dalam pertanian organik. Salah satunya adalah *Experimental Performance of Smart IoT Enabled Drip Irrigation System Using and Controlled Through Web Based Applications* oleh Kant (2023). Penelitian ini membahas mengenai penerapan *Internet of Things (IoT)* dalam mendukung proses perawatan tanaman dengan memanfaatkan sistem

penyiraman otomatis dan penggunaan *website* sebagai *platform monitoring* sensor kelembaban tanah dan suhu tanah. *Sensor Fusion Method* digunakan untuk menggabungkan fitur dari berbagai sensor. Pada penelitian *Smart Soil Property Analysis Using IoT: A Case Study Implementation in Backyard Gardening* oleh Aarth (2023) juga mengevaluasi penggunaan teknologi dalam pertanian organik. Penelitian ini

menggunakan sensor kelembaban tanah dan suhu tanah untuk mengetahui kondisi lingkungan serta menggunakan aplikasi Android untuk melakukan *monitoring* dan *controlling* dan metode *Prototyping* untuk membuat sistem. Penelitian-penelitian tersebut

hanya berfokus pada sistem penyiraman dan tidak pada faktor lainnya yang dapat mempengaruhi pertumbuhan tanaman seperti pemberian pupuk dan pestisida.

Konsistensi dalam merawat kebun sangat penting untuk mendapatkan hasil panen yang optimal. Dalam konteks *Internet of Things (IoT)*, konsistensi menjadi relevan karena menurut Tang et al. (2021), sistem berbasis IoT memiliki kemampuan untuk melakukan tindakan yang sama secara otomatis dan konsisten berdasarkan waktu atau kondisi yang ditentukan. Dalam penelitian ini, fitur penyiraman otomatis, pemberian pupuk, dan pestisida yang diatur berdasarkan jadwal tertentu mencerminkan konsistensi dalam perawatan. IoT memungkinkan kegiatan perawatan dilakukan dengan tepat waktu dan sesuai jadwal yang telah ditentukan, mengatasi masalah ketidakpastian yang mungkin muncul karena faktor manusia. Dengan kata lain, konsistensi dalam tindakan perawatan dapat ditingkatkan melalui penggunaan IoT yang mampu menjalankan tugas-tugas tersebut secara otomatis.

Maka dari itu, peneliti memutuskan untuk melakukan penelitian perancangan sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things (IoT)* yang berfokus pada seluruh proses perawatan tanaman seperti penyiraman air, pemberian pupuk, dan pestisida yang dilakukan secara automasi. Selain itu, meskipun pupuk dan pestisida diberikan dalam jangka waktu yang lebih panjang, penting untuk memahami

bahwa perawatan tanaman bukan hanya tentang frekuensi pemberian, tetapi juga tentang tepatnya waktu pemberian. Menurut Tudi et al. (2021), pemberian pupuk dan pestisida yang tidak tepat waktu dapat mempengaruhi pertumbuhan tanaman dan mengurangi risiko serangan hama atau penyakit. Dalam situasi di mana pengelola kebun dapat menjadi kewalahan dengan tugas-tugas lain, solusi otomatisasi menggunakan komponen sensor dalam penyiraman, pemberian pupuk, dan pestisida telah terbukti efektif dan terjangkau. Berdasarkan fakta yang ada, teknologi berbasis *Internet of Things* (IoT) memungkinkan sensor memberikan data akurat tentang kondisi tanaman dan lingkungan, yang memungkinkan sistem otomatisasi mengatur waktu pemberian secara optimal. Keuntungan dari penggunaan komponen sensor ini adalah selain efisien juga tidak memerlukan biaya yang relatif tinggi, sehingga masih layak untuk diimplementasikan. Dengan pemanfaatan komponen sensor yang biayanya terjangkau, fitur otomatisasi ini dapat membantu pengelola kebun dalam menjaga konsistensi perawatan, mencegah keterlambatan, dan meningkatkan hasil panen secara efektif.

Sistem ini dibangun menggunakan *Microcontroller* ESP32, sensor kelembaban tanah, sensor suhu tanah, sensor *Ultrasonic*, pompa air, LCD, dan aplikasi *Blynk*. Seluruh proses perawatan tanaman seperti penyiraman, pemupukan, dan pemberian pestisida dilakukan secara automasi berdasarkan waktu dan pembacaan sensor-sensor. Pembacaan sensor-sensor tersebut dapat dikirimkan secara *real-time* ke aplikasi melalui konektivitas dengan internet, sehingga pengguna dapat memantau kondisi kebun organik di *smartphone*. Untuk membangun sistem tersebut, peneliti menggunakan metode *Usability Testing* dan *Prototyping*. Menurut Mishra (2021), metode *Usability Testing* digunakan untuk mengidentifikasi masalah yang muncul selama pengujian sistem, menilai kinerja sistem, serta dapat membuktikan efisiensi dalam penggunaan sistem dengan membandingkan informasi yang diberikan oleh pengguna tentang waktu, usaha, dan tingkat kepuasan dalam menyelesaikan tugas yang diberikan sebelum dan sesudah menggunakan sistem *Internet of Things* (IoT). Menurut Francisco et al. (2022), metode *Prototyping* berfungsi untuk membuat model awal sistem. Sehingga, peneliti dapat menghemat biaya, memprediksi *output*/hasil, dan menguji kinerja *software* dan *hardware* sebelum sistem diimplementasikan secara keseluruhan pada lapangan. Metode ini dipilih berdasarkan pertimbangan fungsionalitas dan kebutuhan pengujian yang kompleks dalam pengembangan sistem IoT. Dalam pengembangan sistem IoT, metode *Prototyping* digunakan untuk membuat model awal sistem yang dapat diuji dalam situasi nyata. Ini

membantu dalam mengidentifikasi masalah atau kesalahan yang mungkin tidak terdeteksi pada tahap perencanaan. Penekanan pada pengujian fungsionalitas dalam metode *Prototyping* sangat krusial dalam menjaga interaksi yang baik antara berbagai komponen dalam sistem.

Adapun perbandingan dengan metode *Quality Function Deployment* (QFD), menurut Ishak et al. (2020), *Quality Function Deployment* (QFD) Ini adalah metode yang digunakan untuk mengidentifikasi apa yang diinginkan oleh konsumen atau pelanggan, kemudian menerjemahkannya dengan tepat menjadi persyaratan teknis, sistem pembuatan, dan rencana produksi yang detail bahwa QFD lebih cenderung fokus pada perencanaan fitur dan spesifikasi berdasarkan kebutuhan dan keinginan konsumen. QFD tidak secara khusus difokuskan pada pengujian interaksi komponen seperti yang dilakukan dalam metode *Prototyping*.

Maka dari itu, pemilihan metode *Prototyping* dalam pengembangan sistem *monitoring* dan *controlling* berbasis IoT di kebun organik merupakan keputusan yang berdasarkan pada perlunya pengujian interaksi yang nyata dan identifikasi masalah yang mungkin terjadi dalam lingkungan yang sesungguhnya. Dengan demikian, pemilihan metode ini didasarkan pada kebutuhan yang spesifik dalam pengembangan sistem IoT yang kompleks.

Peneliti juga menggunakan metode *Black Box Testing* yang menurut Supriyono (2020), merupakan teknik pengujian *Software* yang berfokus pada spesifikasi fungsional perangkat lunak. Metode *Black Box Testing* pada penelitian ini digunakan untuk melakukan evaluasi pengguna.

Oleh karena itu, metode Lingkungan Kerja Fisik, *Black Box Testing*, *Usability Testing*, dan *Prototyping* cocok digunakan dalam penelitian ini karena memungkinkan pengembangan dan pengujian sistem *controlling* dan *monitoring* berbasis *Internet of Things* (IoT) dengan fokus pada kebutuhan pengelola kebun organik di Nara Kupu Yogyakarta. Melalui penelitian ini, diharapkan dapat memudahkan pengelola kebun dalam melakukan proses perawatan kebun organik Nara Kupu Yogyakarta. Dengan adanya sistem *monitoring* dan *controlling* berbasis *Internet of Things* (IoT), pengelola kebun dapat melakukan proses perawatan kebun organik secara efisien, memantau kondisi tanaman secara *real-time* dan akurat.

1.2 Rumusan Masalah

Berikut merupakan rumusan masalah pada penelitian ini:

Bagaimana perancangan sistem *monitoring* dan *controlling* berbasis *Internet of Things* (IoT) dapat membuat proses perawatan kebun organik di Nara Kupu Yogyakarta menjadi efisien, akurat dan *real-time*?

1.3 Tujuan Penelitian

Berikut merupakan rumusan masalah pada penelitian ini:

1. Merancang sistem *monitoring* dan *Controlling* kebun organik berbasis *Internet of Things* (IoT) dengan menggunakan metode *Prototyping* dan *Usability Testing*
2. Menggunakan kombinasi dari berbagai komponen pada sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) untuk melakukan proses perawatan kebun dengan automasi secara efisien, akurat dan *real-time*.
3. Menggunakan kombinasi dari berbagai komponen pada sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) untuk mengirimkan notifikasi mengenai kondisi kebun organik melalui aplikasi *Blynk* secara akurat dan *real-time*.

1.4 Manfaat Penelitian

Berikut merupakan manfaat penelitian ini bagi perusahaan dan dunia akademik:

1.4.1 Bagi Perusahaan

Berikut merupakan manfaat penelitian ini bagi perusahaan:

1. Mendapatkan pengetahuan mengenai alat *monitoring* dan *controlling* berbasis *Internet of Things* (IoT) untuk proses pengelolaan kebun organik yang efisien yang bisa diterapkan kedepannya.
2. Meningkatkan kuantitas dan kualitas hasil panen pada kebun organik Nara Kupu Yogyakarta.

1.4.2 Bagi Dunia Akademik

Berikut merupakan manfaat penelitian ini bagi dunia akademik:

1. Berkontribusi pada pengembangan sektor pertanian dengan penerapan *Internet of Things* (IoT).
2. Berkontribusi pada pengembangan *Smart Agriculture*.

1.5 Batasan Penelitian

Berikut merupakan batasan pada penelitian ini:

1. Penelitian dilakukan pada kebun organik Nara Kupu Yogyakarta.
2. Penelitian dilakukan selama kurang lebih 4 bulan, mulai pada bulan Februari hingga Mei 2023.

1.6 Sistematika Penelitian

Dalam penelitian ini terdapat sistematika penulisan yang tersusun dalam enam bab, berikut merupakan detail isi dari tiap bab:

BAB I PENDAHULUAN

Menjelaskan konteks dan alasan pentingnya pengembangan sistem *monitoring* dan *controlling* kebun organik berbasis IoT. Bab ini juga memaparkan rumusan masalah, tujuan penelitian, manfaat penelitian, serta batasan masalah yang diangkat dalam penelitian ini.

BAB II KAJIAN LITERATUR

Bab 2 menjelaskan landasan teori yang dijadikan sebagai referensi dan konsep-konsep yang mendukung pengembangan sistem *monitoring* dan *controlling* kebun organik berbasis IoT. Terdapat dua pembahasan utama, yaitu kajian induktif yang berisikan penelitian-penelitian terdahulu yang serupa yaitu penerapan *Internet of Things* dalam bidang agrikultur dan kajian deduktif yang berisikan teori-teori yang berkaitan dengan penelitian ini berdasarkan penjelasan para ahli atau penelitian sebelumnya

BAB III METODE PENELITIAN

Bab 3 memuat kerangka rencana penelitian berdasarkan pertanyaan 5W1H, penjelasan subjek dan objek pada penelitian, pemaparan jenis data, metode, instrumen, dan desain eksperimen pada penelitian ini, serta bagan alur penelitian dari awal sampai akhir.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab 4 berisikan metode *Prototyping* sebagai pendekatan dalam Pembangunan sistem secara *hardware* dan *software* serta metode *blackbox* yang digunakan untuk instrumen dalam melakukan evaluasi pengguna. Dilakukan pengumpulan data lingkungan kerja fisik yang akan digunakan untuk bahan evaluasi serta analisis berdasarkan nilai *usability* dari sistem yang dibangun. Dilakukan *Usability Testing* dengan mengolah hasil kuesioner *System Usability Scale* pada sistem *monitoring* dan *controlling* berbasis *Internet of Things*. Dilakukan juga Pembangunan Tata letak sistem untuk pembuatan rekayasa peletakan alat.

BAB V HASIL DAN PEMBAHASAN

Berisikan hasil dan pembahasan dari pengolahan data yang telah dilakukan pada bab sebelumnya. Melakukan analisis *Usability Testing* yang berisikan mengenai tanggapan

pengguna terhadap sistem yang telah dibangun dan pembahasan mengenai nilai *usability* dari sistem tersebut serta pembahasan mengenai Lingkungan Kerja Fisik (LKF) yang berisikan tentang proyeksi peletakkan alat dan rekayasa hasil panen jika sistem yang dibangun diterapkan secara nyata.

BAB VI KESIMPULAN DAN SARAN

Bab 6 berisikan kesimpulan yang menjawab rumusan masalah pada penelitian ini. Peneliti memberikan saran serta rekomendasi sebagai acuan dan bahan evaluasi bagi untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Literatur

Tinjauan terhadap penelitian terdahulu tentang penerapan *Internet of Things* (IoT) pada bidang agrikultur dilakukan untuk membandingkan penelitian ini dengan penelitian sebelumnya, sebagai acuan untuk meningkatkan isi penelitian.

Penelitian yang dilakukan oleh Ratna & Ibrahim (2021) yang berjudul “Perancangan dan Pengujian Alat untuk *monitoring* Kelembaban Tanah dan Pemberian Pupuk Cair pada Tanaman Cabai Berbasis *Internet of Things*”. penelitian ini bertujuan untuk membuat alat *monitoring* kelembaban tanah dan pemberian pupuk cair pada tanaman cabai secara automasi yang berbasis *microcontroller* ESP32, nantinya sistem akan memberikan informasi melalui internet serta menggunakan aplikasi *Blynk* untuk *monitoring*. Sensor yang digunakan adalah suhu tanah dan kelembaban tanah.

Penelitian yang dilakukan oleh Miftahul Walid et al. (2022) yang berjudul “Pengembangan Sistem Irigasi Pertanian Berbasis *Internet of Things* (IoT)”. Tujuan dari penelitian ini adalah membuat sistem irigasi secara automasi berbasis *microcontroller* ESP32 berdasarkan dari waktu dan kelembaban tanah. Alat-alat yang digunakan pada sistem ini adalah pompa air, sensor kelembaban tanah, *solenoid valve*, dan sensor *Ultrasonic*. Sensor *Ultrasonic* digunakan untuk mengetahui jumlah air di tabung untuk proses irigasi.

Penelitian yang dilakukan oleh Sandi & Fatma (2023) yang berjudul “Pemanfaatan Teknologi *Internet of Things* (IoT) pada Bidang Pertanian”. Tujuan dari penelitian ini adalah membuat sistem irigasi secara automasi dan *monitoring* kondisi tanah berbasis Arduino UNO. Alat-alat yang digunakan pada penelitian ini adalah sensor suhu, sensor kelembaban tanah, pompa air, sensor Ph tanah, sensor *Ultrasonic*, dan LCD. Pembacaan sensor akan dikirimkan ke aplikasi *Blynk* di *Smartphone* melalui internet.

Penelitian yang dilakukan oleh Budihartono & Rakhman (2022) yang berjudul “*Monitoring* Suhu dan Kelembaban Tanah pada Budidaya Porang Berbasis Arduino”. Tujuan dari penelitian ini adalah membangun sistem *monitoring* suhu dan kelembaban tanah berbasis *Internet of Things* pada budidaya porang. Alat-alat yang digunakan adalah sensor suhu, sensor kelembaban tanah, dan *Microcontroller* Arduino. Metode yang digunakan adalah *Waterfall*.

Penelitian yang dilakukan oleh Deworo et al. (2023) yang berjudul “Rancang Bangun Alat Penyiraman Otomatis Menggunakan Sensor Kelembaban Tanah Berbasis *Microcontroller* pada Tanaman Kopi”. Penelitian ini bertujuan untuk membangun sistem penyiraman otomatis pada tanaman kopi menggunakan *microcontroller* Arduino, pompa elektrik, *sprayer*, dan sensor kelembaban tanah.

Penelitian yang dilakukan oleh M. S. A. Mahmud et al. (2018) yang berjudul “*Internet of Things Based Smart Environmental Monitoring for Mushroom Cultivation*”. Tujuan dari penelitian ini adalah untuk merancang sebuah sistem kontrol irigasi secara automasi dan *monitoring* level karbon dioksida, suhu, dan kelembaban berbasis *microcontroller* ESP8622 melalui konektivitas *Internet* dan aplikasi *Mobile* untuk menyediakan kondisi yang optimal bagi produksi jamur.

Penelitian yang berjudul Subashini et al. (2018) “*Internet of Things Based Wireless Plant Sensor for Smart Farming*”. Tujuan dari penelitian ini adalah membuat alat irigasi secara automasi dan *monitoring* berbasis *microcontroller* ESP8266. Sensor yang digunakan adalah kelembaban tanah, intensitas Cahaya, kelembaban relatif, suhu udara, dan kelembaban tanah. Pembacaan sensor-sensor akan dikirim melalui konektivitas *Internet*, nantinya informasi tersebut akan digunakan pengguna untuk menganalisis proses pertumbuhan tanaman.

Penelitian yang dilakukan oleh Ardiansah et al. (2020) yang berjudul “*Greenhouse Monitoring and Automation using Arduino: a Review on Precision Farming and Internet of Things (IoT)*”. Tujuan dari penelitian ini adalah membuat sistem *monitoring* dan automasi pada *Greenhouse* yang berbasis Arduino UNO. Alat-alat yang digunakan adalah sensor PH, sensor kelembaban, dan sensor suhu. pembacaan dari sensor-sensor tersebut akan dikirimkan ke *smartphone* pengguna menggunakan aplikasi LINE melalui internet.

Penelitian yang dilakukan oleh Wicaksono et al. (2021) yang berjudul “*Increasing Productivity of Rice Plants Based on IoT (Internet of Things) to Realize Smart Agriculture Using System Thinking approach*”. Tujuan dari penelitian ini adalah menerapkan *Internet of Things* pada kebun padi dengan menggunakan sensor kelembaban, suhu, dan PH untuk meningkatkan produktivitas lahan padi. Metode yang digunakan pada penelitian ini adalah *System Thinking Approach* yang data variabelnya didapatkan dari hasil pembacaan sensor. Penelitian ini dilakukan untuk menerapkan *Smart Agriculture*.

Penelitian yang dilakukan oleh Kiani & Seyyedabbasi (2018) yang berjudul “*Wireless Sensor Network and Internet of Things in Precision Agriculture*”. Tujuan dari penelitian ini adalah membuat sistem penyiraman secara automasi dengan menggunakan pembacaan dari berbagai sensor seperti suhu, kelembaban, dan kelembaban tanah. Metode yang digunakan pada sistem ini adalah *Fuzzy Logic*.

Penelitian yang dilakukan oleh Rehman et al. (2022) yang berjudul “*A Revisit of Internet of Things Technologies for Monitoring and Control Strategies in Smart Agriculture*”. Tujuan dari penelitian ini adalah membangun sistem *monitoring* dan *controlling* pada lahan perkebunan dengan menggunakan berbagai sensor seperti suhu, kelembaban, kelembaban tanah, hujan, PH, dan kekeruhan air. *Monitoring* yang dilakukan pada sistem ini adalah air, tanah, suhu, dan hama. Sedangkan *controlling* yang dilakukan adalah irigasi, pemupukan, pemberian pestisida, dan analisis kondisi tanah.

Penelitian yang dilakukan oleh Supriyanto et al. (2022) yang berjudul “*Smart Greenhouse Coffee Dryer with Fuzzy Algorithm on Internet of Things Platform*”. Tujuan dari penelitian ini adalah membangun sistem *monitoring* dan *controlling* pada kebun kopi dengan menerapkan *Internet of Things*. Proses *monitoring* yang dilakukan adalah suhu ruangan dan *controlling* yang dilakukan adalah proses pengeringan kopi dengan menggunakan cahaya lampu dan kipas. Metode yang digunakan adalah *Fuzzy Algorithm*.

Penelitian yang dilakukan oleh Ali et al. (2019) yang berjudul “*A Smart Monitoring and controlling for Agricultural using LoRa IoT Technology*”. Tujuan dari penelitian ini adalah membangun sistem irigasi yang dapat di *monitor* dan *Control* secara jarak jauh melalui *smartphone* dengan menerapkan *LoRa Communication*. Sensor yang digunakan adalah suhu tanah dan *Microcontroller* yang digunakan pada sistem adalah ESP32.

Penelitian yang dilakukan oleh Al-Obaidi et al. (2020) yang berjudul “*Technique Smart Control Soil Moisture System to Watering Plant Based on IoT with Arduino UNO*”. Tujuan dari penelitian ini adalah membangun sistem penyiraman secara automasi berbasis *Internet of Things* dengan menggunakan konektivitas internet, sensor kelembaban tanah, dan *microcontroller* Arduino UNO.

Penelitian yang dilakukan oleh Visconti et al. (2020) yang berjudul “*IoT-Oriented Software Platform Applied to Sensors-Based Facility with Smartphone Farmer App*”. Tujuan dari penelitian ini adalah membangun sistem *monitoring* dan *controlling* berbasis IoT pada lahan pertanian. *Hardware* yang digunakan adalah pompa air, sensor

kelembaban tanah, sensor suhu, *Microcontroller* ESP8266, panel surya, dll. Proses *monitoring* yang dilakukan adalah pemantauan suhu, kelembaban, cuaca, dll. Proses *controlling* yang dilakukan adalah proses irigasi dan pemupukan.

Berikut adalah perbandingan sensor dan perangkat lainnya yang digunakan dalam penelitian yang disebutkan sebelumnya, seperti yang terlihat dalam Tabel 2.1.

Tabel 2. 1 Perbandingan Penelitian Sejenis

No	Penulis	Sensor Suhu Tanah	Sensor Kelembaban Tanah	Sensor <i>Ultrasonic</i>	Automasi Proses Penyiraman	Automasi Proses Pemupukan	Automasi Proses Pemberian Pestisida	<i>Microcontroller</i> ESP32	Aplikasi <i>Blynk</i>	Metode <i>Prototyping</i>	Metode <i>Usability Testing</i>
1	Ratna & Ibrahim (2021)	√	√		√	√	√	√			
2	Miftahul Walidet al. (2022)		√	√	√		√	√			
3	Sandi & Fatma (2023)	√	√	√	√			√			
4	Budihartono & Rakhman (2022)	√	√								
5	Deworo et al. (2023)		√		√					√	
6	M. S. A. Mahmud et al. (2018)	√	√		√						
7	Subashini et al. (2018)	√	√		√					√	
8	Ardiansah et al. (2020)	√	√	√	√		√				
9	Wicaksono et al. (2021)	√	√		√	√	√				
10	Kiani & Seyyedabbasi (2018)	√	√		√						

No	Penulis	Sensor Suhu Tanah	Sensor Kelembaban Tanah	Sensor <i>Ultrasonic</i>	Automasi Proses Penyiraman	Automasi Proses Pemupukan	Automasi Proses Pemberian Pesticida	<i>Microcontroller</i> ESP32	Aplikasi <i>Blynk</i>	Metode <i>Prototyping</i>	Metode <i>Usability Testing</i>
11	Rehman et al. (2022)	√	√	√	√	√	√				
12	Supriyanto et al. (2022)	√								√	
13	Ali et al. (2019)	√			√		√			√	
14	Obaidi et al. (2020)		√		√						
15	Visconti et al. (2020)	√	√		√	√				√	
	Penelitian yang diajukan	√	√	√	√	√	√	√	√	√	√

Berdasarkan perbandingan dengan penelitian sebelumnya, penelitian ini memiliki perbedaan dalam proses *monitoring*, *controlling* penggunaan sensor, perangkat, dan automasi. Penelitian ini berfokus pada tiga automasi, yaitu penyiraman, pemberian pupuk, dan pestisida. Penelitian ini menggunakan 3 sensor, yaitu sensor kelembaban tanah, sensor suhu tanah, sensor *Ultrasonic*. Perangkat yang digunakan adalah LCD dan *Water pump*. *Microcontroller* yang digunakan adalah ESP32. Hasil pembacaan sensor dapat diakses melalui aplikasi *Blynk* pada *smartphone* yang terhubung dengan internet. Dengan demikian, pengelola kebun dapat memantau dan mengelola kebun organik secara *real-time*.

2.2 Landasan Teori

2.2.1 Kebun Organik

Menurut Ardiansah et al. (2020), kebun organik dapat didefinisikan sebagai teknik budidaya pertanian yang mengandalkan bahan-bahan alami tanpa menggunakan bahan-bahan kimia sintetis. Praktik ini memberikan jaminan kepada masyarakat atau konsumen bahwa produk yang dihasilkan memberikan rasa aman untuk dikonsumsi dikarenakan produk yang dihasilkan berkualitas. Praktik Pertanian Organik mampu menjaga kualitas tanah, air tanah, air permukaan dan atmosfer dengan baik dengan tidak menekan sumber-

sumber yang dapat diperbaharui. Selain itu, Telah terbukti bahwa penerapan praktik ini mampu menciptakan *agricultural sustainability*. Menurut Widjajanto (2021), hal ini sesuai dengan pemahaman bahwa kegiatan pertanian harus mampu memenuhi kebutuhan pangan dan lingkungan yang sehat bagi generasi sekarang tanpa menghambat kemampuan generasi mendatang dalam memenuhi kebutuhannya.

Menurut Mardianah et al. (2022), di Indonesia telah ada sejumlah kebun organik yang ditemukan. Pertanian organik telah meningkat dalam beberapa tahun terakhir sebagai respons terhadap permintaan konsumen yang semakin meningkat akan makanan yang sehat dan ramah lingkungan. Banyak petani di Indonesia mulai beralih ke praktik pertanian organik untuk menghasilkan tanaman dan produk hortikultura yang bebas dari pestisida dan bahan kimia sintetis.

Salah satu cara untuk meningkatkan efisiensi pertanian organik di Indonesia adalah melalui penggunaan *Internet of Things* (IoT). Dengan memanfaatkan sensor dan sistem pemantauan yang terhubung secara digital, IoT dapat digunakan dalam kebun organik untuk meningkatkan efisiensi dan produktivitas. Menurut Cahyani (2023), melalui IoT petani dapat mengontrol penggunaan air, pupuk, dan pestisida secara akurat berdasarkan data sensor kelembaban tanah, suhu, dan kelembaban udara. Dengan demikian, integrasi IoT dalam kebun organik dapat menjadi alat yang berharga dalam mencapai tujuan pertanian organik yang lebih efisien dan berkelanjutan di Indonesia.

2.2.2 Monitoring Kebun Organik

Menurut Darmawan et al. (2022), definisi *monitoring* dalam konteks kebun organik adalah proses pengumpulan, pemantauan, dan analisis data yang berkaitan dengan kondisi lingkungan, pertumbuhan tanaman, dan pengelolaan sumber daya seperti air, pupuk, dan pestisida. Tujuan utama dari *monitoring* adalah untuk memperoleh informasi yang akurat tentang keadaan kebun organik sehingga dapat diambil keputusan yang tepat dalam pengelolaan pertanian organik. Dengan adanya koneksi yang terus-menerus melalui IoT, data dari sensor-sensor yang dipasang di kebun organik dapat dikirim secara real-time ke aplikasi *Blynk* pada *smartphone* pengguna. Ini memungkinkan pengelola kebun untuk mengakses dan menganalisis data dengan mudah dari jarak jauh. Data yang dikumpulkan melalui sensor kelembaban tanah dan suhu tanah dapat memberikan gambaran yang akurat tentang kondisi kebun dan memberikan informasi penting untuk pengambilan keputusan yang lebih baik.

Berdasarkan penelitian terdahulu yang dikaji, dalam upaya *monitoring* perawatan kebun organik di Nara Kupu, peneliti telah mengumpulkan penelitian yang relevan. Tujuan penelitian ini adalah memahami dan mengoptimalkan faktor-faktor penting dalam perawatan kebun organik dengan menerapkan sistem IoT. Aspek-aspek yang diteliti mencakup pengaruh suhu dan kelembaban tanah terhadap pertumbuhan tanaman. Menurut Asih (2018), melalui metode *fuzzification* suhu dan kelembaban tanah, didapatkan bahwa batas suhu tanah yang optimal untuk pertumbuhan tanaman adalah 31°C. Sedangkan untuk kelembaban tanah, batas yang optimal untuk pertumbuhan tanaman adalah 50%.

Oleh karena itu pada proses *monitoring* kebun organik, terdapat beberapa penjadwalan yang dipengaruhi oleh pembacaan sensor suhu dan kelembaban tanah. Pertama adalah ketika suhu tanah di atas 31°C. Jika kondisi tersebut terpenuhi, maka sistem akan melakukan automasi penyiraman hingga suhu tanah kembali normal (dibawah 31°C) serta mengirimkan notifikasi melalui aplikasi *Blynk* untuk memberitahu pengguna bahwa kebun telah disiram karena suhu tanah melebihi batas yang ditentukan. Kedua adalah ketika kelembaban kurang dari 50%. Jika kondisi tersebut terpenuhi, maka sistem akan melakukan automasi penyiraman hingga kelembaban tanah kembali normal (diatas 50%) serta mengirimkan notifikasi melalui aplikasi *Blynk* untuk memberitahu pengguna bahwa kebun telah disiram karena kelembaban tanah kurang dari batas yang ditentukan.

Dengan demikian, penggunaan IoT dalam *monitoring* kebun organik dapat meningkatkan efisiensi pertanian, mengurangi penggunaan sumber daya yang berlebihan, dan mendukung pertanian organik yang berkelanjutan.

2.2.3 Controlling Kebun Organik

Controlling dalam konteks kebun organik merujuk pada proses pengendalian dan pengaturan operasional di kebun untuk memastikan bahwa berbagai faktor yang mempengaruhi pertumbuhan dan kesehatan tanaman organik tetap terkendali. Dalam praktik pertanian organik, kontrol yang efektif melibatkan pengendalian penggunaan air, pupuk, pestisida, dan praktik pertanian lainnya agar sesuai dengan prinsip-prinsip pertanian organik.

Berdasarkan penelitian terdahulu yang dikaji, dalam upaya *controlling* perawatan kebun organik di Nara Kupu, peneliti telah mengumpulkan serangkaian penelitian yang relevan. Tujuan penelitian ini adalah memahami dan mengoptimalkan faktor-faktor

penting dalam perawatan kebun organik dengan menerapkan sistem IoT. Aspek-aspek yang diteliti mencakup penyiraman air, penyiraman pestisida, dan penyiraman pupuk organik cair.

Menurut Seran et al. (2022), penyiraman rutin sehari dua kali pada pagi dan sore hari menyebabkan pertumbuhan tanaman yang optimal. Jumlah air yang dibutuhkan pertanaman adalah 155 ml dalam sekali penyiraman. Penyiraman dilakukan saat pagi hari (saat suhu masih dingin) untuk menyiapkan tanaman dari kenaikan suhu yang akan terjadi pada siang hari. Sedangkan penyiraman pada sore hari dilakukan agar tanaman dapat menyimpan air tersebut hingga penyiraman berikutnya. Peneliti tidak melakukan penyiraman pada siang hari karena air tersebut akan ter-*evaporasi*, sehingga kuantitas air yang didapatkan oleh tanaman tidak akan optimal. Peneliti juga tidak melakukan penyiraman air pada pukul 10.00 – 14.00, karena pada jam tersebut stomata tumbuhan sedang terbuka, sehingga akan mengganggu proses pertumbuhan tanaman jika melakukan penyiraman pada jam-jam tersebut. Selain itu, menurut Gumelar & Yunus (2021) perlakuan frekuensi penyiraman pupuk organik cair sebanyak 5 ml/tanaman pada dua minggu sekali memberikan efek yang paling optimal dalam pertumbuhan tanaman. Selain itu, proses perawatan tanaman akan lebih optimal jika memberikan pestisida agar tanaman terhindar dari hama. Frekuensi dan jumlah penyiraman yang dilakukan pun juga berpengaruh besar pada proses perawatan tanaman, jika frekuensi dan jumlah penyiraman terlalu sedikit dapat menyebabkan masih timbulnya hama, sedangkan jika frekuensi dan jumlah penyiraman terlalu banyak dapat menyebabkan tanaman menjadi layu/mati. Oleh karena itu, menurut Nboyine et al. (2022) jumlah dan frekuensi penyiraman yang memberikan hasil paling optimal adalah 10ml/tanaman pada 1 × seminggu.

Oleh karena itu pada proses *controlling* kebun organik, terdapat beberapa penjadwalan yang berdasarkan dari waktu yang ditentukan. Penyiraman air pada tanaman dilakukan 2 × sehari pada jam 6 pagi dan 6 sore setiap hari agar tanaman tidak mengalami kekeringan. Setelah itu, dilakukan penyiraman pupuk cair setiap 2 minggu sekali agar tanaman selalu mendapatkan nutrisi yang cukup. Selain itu, tanaman juga diberikan pestisida setiap minggu sekali agar tanaman terhindar dari hama.

Dengan menggunakan IoT dalam proses *controlling* di kebun organik, petani dapat mengoptimalkan penggunaan sumber daya dan menjaga kestabilan lingkungan pertanian. Melalui pengendalian yang lebih akurat dan efisien, IoT membantu petani

untuk menghasilkan tanaman organik yang sehat, berkualitas tinggi, dan sesuai dengan prinsip-prinsip pertanian organik yang berkelanjutan.

2.2.4 Internet of Things (IoT)

Menurut Yadav (2023), istilah *Internet of Things* (IoT) pertama kali dibuat pada tahun 1999 oleh Kevin Ashton dari MIT saat ia memperkenalkan teknologi identifikasi frekuensi radio (RFID). Sejak itu, IoT telah menjadi semakin populer sebagai teknologi yang menjanjikan untuk menghubungkan setiap objek menjadi bagian dari jaringan online dan memungkinkan komunikasi antara mesin dan manusia, serta antara mesin dengan dunia fisik.

Definisi *Internet of Things* (IoT) menurut Wang et al. (2022) adalah jenis jaringan yang mengumpulkan informasi secara real-time melalui berbagai perangkat pendeteksi informasi, menghubungkan objek nyata di dunia fisik dengan Internet sesuai dengan protokol yang telah ditentukan sebelumnya. Menurut Gaitan (2021) Kata "*Things*" dalam *Internet of Things* (IoT) adalah berbagai jenis perangkat fisik yang dilengkapi dengan mikrokontroler, *transceiver* komunikasi digital, dan protokol tumpukan. Tujuannya adalah untuk memungkinkan perangkat-perangkat ini berkomunikasi satu sama lain dan dengan pengguna. Jurnal yang telah terbit di *Digital Publishing Institute* (MDPI) tersebut juga menyampaikan tujuan *Internet of Things* (IoT) untuk menghubungkan dunia fisik dengan dunia virtual melalui Internet. Dengan menggunakan Internet sebagai sarana komunikasi dan pertukaran informasi, IoT memungkinkan perangkat komputasi, mesin mekanik dan digital, objek, hewan, atau bahkan manusia yang memiliki identitas unik untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia-ke-manusia atau manusia-ke-komputer. Dalam hal ini, IoT menciptakan sistem yang saling tergantung antara berbagai jenis perangkat untuk memberikan kemudahan dan manfaat bagi pengguna. Sedangkan menurut Standardization (2022), *Internet of Things* (IoT) adalah infrastruktur objek, orang, sistem, dan sumber daya informasi yang saling berhubungan bersama dengan layanan cerdas untuk memungkinkan mereka memproses informasi dari dunia fisik dan virtual dan bereaksi.

Menurut Brous et al. (2020), penerapan IoT pada sebuah organisasi menghasilkan beberapa manfaat, yaitu:

1. Meningkatkan efisiensi waktu karena IoT memungkinkan proses automasi dan pengumpulan data secara *real-time*.

2. Pengurangan biaya operasional karena memungkinkan semua proses untuk dilakukan secara automasi dan tidak membutuhkan tenaga dari manusia.
3. Mendapatkan informasi mengenai strategi atau prosedur yang kurang sesuai dan kurang efisien pada sebuah proses/sistem dikarenakan data didapatkan secara *real-time* dan akurat.
4. Meningkatkan efisiensi dan fleksibilitas proses pekerjaan karena bisa melakukan *monitoring* sistem kapan saja dan dimana saja.
5. Meningkatkan penghasilan dari sebuah sistem karena dapat mengetahui proses kerja yang kurang efisien secara *real-time*.

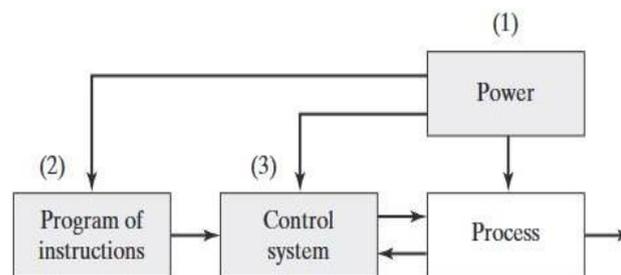
Menurut Adani & Salsabil (2019), dapat dikatakan bahwa *Internet of Things* adalah bagaimana suatu objek yang nyata di dunia ini digambarkan di dunia maya (Internet). Salah satu contoh manfaat dari penerapan *Internet of Things* adalah pada kedai kopi internasional “*Starbucks*”. Dalam beberapa tahun ke depan, dilaporkan berencana menghubungkan kulkas dan mesin kopi milik mereka dengan teknologi *Internet of Things*. Sehingga mereka dapat meningkatkan pelayanan mereka dengan mengetahui apa saja yang lebih disukai konsumen, meramalkan kebutuhan stok barang, dan masih banyak lainnya yang pada akhirnya akan meningkatkan efisiensi dan keuntungan. Menurut Rehman et al. (2022), Contoh penerapan *Internet of Things* (IoT) lainnya meliputi industri *Smart Cities, Smart Homes, Smart Energy, Connected Vehicles, Smart Agriculture, Connected Buildings, Healthcare, dan Logistics*.

Penelitian ini adalah contoh dari penerapan konsep *Smart Agriculture*, dimana sensor-sensor yang terkoneksi dengan jaringan IoT dapat dipasang di lahan pertanian organik. Sensor-sensor ini akan memantau berbagai parameter seperti kelembaban tanah, suhu, tingkat keasaman, dan kualitas udara. Menurut Xu et al. (2022), dengan penerapan *Smart Agriculture* berbasis IoT, pertanian organik dapat menjadi lebih produktif, efisien, dan ramah lingkungan serta bisa mengendalikan sistem pertanian yang rumit dan juga membantu dalam situasi darurat di pertanian. Teknologi IoT memberikan kemampuan untuk mengumpulkan data secara *real-time*, menganalisisnya, dan memberikan informasi yang cerdas kepada petani organik. Dengan demikian, IoT memberikan dukungan yang penting dalam menjaga kualitas dan keberlanjutan pertanian organik, serta membantu memenuhi kebutuhan pangan secara berkelanjutan.

2.2.5 Automasi

Automasi adalah teknologi yang memungkinkan suatu proses dilakukan tanpa bantuan manusia. Hal ini dilakukan melalui program instruksi dan sistem kontrol. Pada dasarnya, automasi membutuhkan daya untuk menjalankan proses dan mengoperasikan program dan kontrol. Meskipun automasi banyak digunakan dalam berbagai bidang, namun terutama terkait dengan industri manufaktur. Istilah "automasi" pertama kali digunakan dalam konteks manufaktur oleh seorang manajer rekayasa di Ford Motor Company pada tahun 1946. Awalnya, automasi mengacu pada perangkat otomatis yang dipasang di pabrik Ford. Saat ini, hampir semua aplikasi automasi modern dikendalikan oleh komputer, meskipun pada awalnya komputer belum tersedia. Menurut Groover (2022), sebuah Sistem otomatis terdiri dari tiga elemen dasar:

- (1) daya untuk menjalankan proses dan mengoperasikan sistem,
- (2) program instruksi untuk mengarahkan proses.
- (3) sistem kontrol untuk menjalankan instruksi tersebut.



Gambar 2. 1 Elemen Dasar Automasi

Hubungan antara ketiga elemen ini diilustrasikan dalam Gambar 4.2. Semua sistem yang memenuhi syarat sebagai otomatis termasuk ketiga elemen dasar ini dalam bentuk yang berbeda. Mereka hadir dalam tiga jenis dasar sistem manufaktur otomatis: automasi tetap, automasi yang dapat diprogram, dan automasi fleksibel. Daya untuk menyelesaikan proses otomatis Sistem otomatis digunakan untuk mengoperasikan suatu proses, dan diperlukan daya untuk menggerakkan proses tersebut serta sistem kontrolnya. Sumber utama daya dalam sistem otomatis adalah listrik. Listrik memiliki banyak keunggulan dalam proses otomatis maupun non-otomatis:

- 1) Listrik tersedia secara luas dengan biaya yang moderat. Ini merupakan bagian penting dari infrastruktur industri.

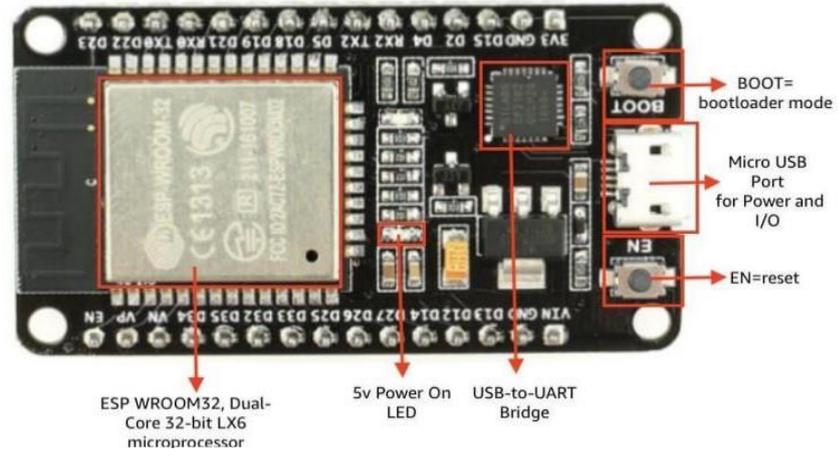
- 2) Listrik dapat dengan mudah dikonversi menjadi bentuk energi alternatif: mekanik, termal, cahaya, akustik, hidrolik, dan pneumatik.
- 3) Daya listrik pada level rendah dapat digunakan untuk melakukan fungsi-fungsi seperti transmisi sinyal, pemrosesan informasi, dan penyimpanan dan komunikasi data.

Selain kebutuhan daya dasar untuk operasi manufaktur, daya tambahan diperlukan untuk automasi. Daya tambahan ini digunakan untuk fungsi-fungsi berikut:

- 1) Unit kontrol. Kontroler industri modern didasarkan pada komputer digital, yang membutuhkan daya listrik untuk membaca program instruksi, melakukan perhitungan kontrol, dan menjalankan instruksi dengan mengirimkan perintah yang tepat ke perangkat yang diaktifkan.
- 2) Daya untuk menjalankan sinyal kontrol. Perintah yang dikirim oleh unit kontrol dilakukan melalui perangkat elektromekanik, seperti saklar dan motor, yang disebut aktuator. Perintah-perintah ini biasanya ditransmisikan dengan menggunakan sinyal kontrol tegangan rendah. Untuk menjalankan perintah tersebut, aktuator membutuhkan daya yang lebih besar, sehingga sinyal kontrol harus diperkuat untuk menyediakan tingkat daya yang sesuai untuk perangkat penggerak.
- 3) Pengumpulan data dan pemrosesan informasi. Dalam sebagian besar sistem kontrol, data harus dikumpulkan dari proses dan digunakan sebagai input untuk algoritma kontrol. Selain itu, untuk beberapa proses, persyaratan hukum meminta adanya catatan kinerja proses dan/atau kualitas produk. Fungsi pengumpulan data dan pencatatan ini membutuhkan daya, meskipun dalam jumlah yang moderat.

2.2.6 ESP32

Menurut Wag yana (2019), ESP32 adalah Mikrokontroler *System on Chip (SoC)* merupakan mikrokontroler kelanjutan dari suatu mikrokontroler ESP8266 yang dapat digunakan untuk implementasi IoT. Modul ESP32 memiliki inti CPU dan *WiFi* yang cepat, GPIO lebih. Menurut G. Santoso et al. (2022), ESP32 memiliki kemampuan untuk mendukung terkoneksi ke *WiFi* secara langsung dan ditambah dengan BLE (*Bluetooth Low Energy*) dalam chip. ESP32 mendukung beberapa jenis pemrograman, jenis pemrograman yang digunakan pada penelitian ini yaitu Arduino IDE. Spesifikasi dan Gambar microcontroller ESP32 dapat dilihat pada tabel 2.2 dan gambar 2.2.



Gambar 2. 2 *Microcontroller ESP32*
 Sumber: Abdulsalam et al. (2023)

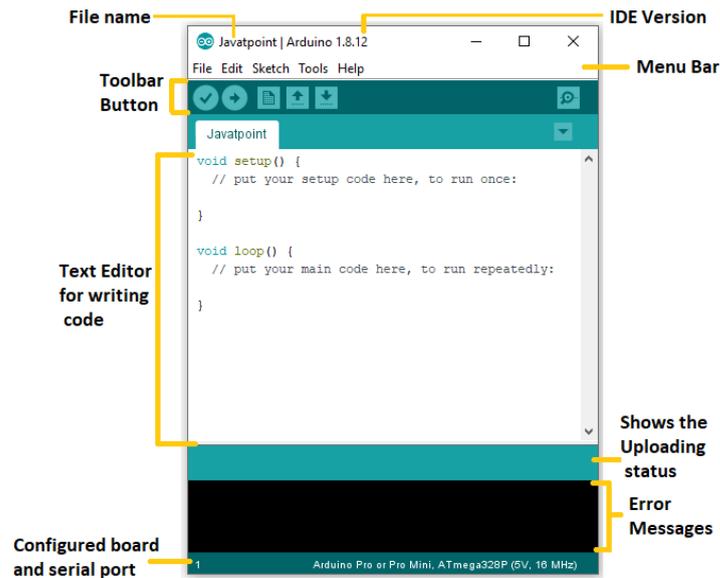
Tabel 2. 2 Spesifikasi *Microcontroller* ESP32

<i>Processor</i>	<i>Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz.</i>
<i>Memory</i>	<i>Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR and BLE (shares the radio with Wi-Fi).</i>
<i>Peripheral I/O</i>	<i>12-bit SAR ADC (up to 18 channels), 2x 8-bit DACs, 10x touch sensors (capacitive sensing GPIOs), 4x SPI, 2x I2S interfaces, 2x I2C interfaces, 3x UART, SD/SDIO/CE-ATA/MMC/eMMC host controller, SDIO/SPI slave controller, Ethernet MAC interface, CAN bus 2.0, infrared remote controller (TX/RX, up to 8 channels), motor PWM, LED PWM (up to 16 channels), hall effect sensor, ultra low power analog preamplifier.</i>
<i>Security</i>	<i>standard security, secure boot, flash encryption, 1024-bit OTP (up to 768-bit for customers), cryptographic hardware acceleration (AES, SHA-2, RSA, ECC), random number generator (RNG).</i>

Dalam penelitian ini, ESP32 digunakan sebagai pusat kontrol dalam sistem *monitoring* dan *controlling* kebun organik berbasis IoT, yang berfungsi untuk mengumpulkan data dari sensor-sensor yang terpasang dan mengirimkan data tersebut ke server atau aplikasi yang digunakan untuk pemantauan dan pengendalian.

2.2.7 Arduino IDE

Menurut Nugroho (2022), Arduino IDE (*Integrated Development Environment*) adalah *software* yang digunakan untuk memprogram di arduino, dengan kata lain Arduino IDE sebagai media untuk memprogram board Arduino. Arduino IDE bisa di download secara gratis di website resmi Arduino IDE. Arduino IDE ini berguna sebagai *text editor* untuk membuat, mengedit, dan juga memvalidasi kode program. bisa juga digunakan untuk meng-upload ke board Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino “*sketch*” atau disebut juga *source code* arduino, dengan ekstensi file *source code*. *Software* ini memiliki dukungan yang luas untuk berbagai jenis mikrokontroler, termasuk ESP32. Selain itu, juga menyediakan banyak *library* yang dapat mempercepat proses pengembangan proyek dengan ESP32. Gambar *software* Arduino IDE dapat dilihat pada gambar 2.3.



Gambar 2. 3 *Software Arduino IE*

Sumber: Ahmed (2023)

2.2.8 Pompa Air

Menurut Hafizh et al. (2018), definisi dan fungsi pompa adalah merupakan pesawat angkat untuk memindahkan zat cair atau fluida zat cair hanya mengalir bila terdapat perbedaan tekanan tertentu. Pompa dapat didefinisikan sebagai penambahan energi untuk menggerakkan zat cair dari suatu tempat ke tempat yang lainnya, penambahan energi yang dilakukan akan menggerakkan atau mengalirkan zat cair melalui pipa atau pindah ke tempat yang lebih tinggi atau tekanan yang lebih tinggi. Pada penelitian ini, pompa air digunakan untuk memindahkan air, pupuk, dan pestisida dari bak penyimpanan ke tanah.

2.2.9 Sensor

Sensor merupakan perangkat yang berfungsi untuk mendeteksi gejala atau sinyal yang berasal dari perubahan energi, seperti energi listrik, energi fisik, energi kimia, energi biologi, energi mekanik, dan lain sebagainya. Sensor-sensor ini memiliki peran penting dalam berbagai aplikasi, termasuk dalam bidang pertanian, industri, kesehatan, lingkungan, dan lain-lain. Keakuratan, sensitivitas, respon waktu, dan stabilitas sensor merupakan faktor-faktor penting yang harus dipertimbangkan saat memilih sensor yang sesuai untuk suatu aplikasi tertentu. Dalam penelitian ini, dilakukan implementasi sensor-sensor seperti sensor suhu tanah, sensor kelembaban tanah, dan sensor *Ultrasonic* dalam sistem *monitoring* dan kontrol kebun organik berbasis IoT. Tujuan dari penggunaan sensor-sensor ini adalah untuk mengukur variabel lingkungan yang relevan dalam kebun organik dan memberikan data yang diperlukan untuk pengambilan keputusan yang tepat

dalam pengendalian dan pemeliharaan kebun organik. Dengan demikian, sistem ini memungkinkan pemantauan yang efektif dan pengelolaan yang optimal dari kebun organik.

Sensor suhu tanah adalah perangkat yang digunakan untuk mengukur suhu dalam tanah. Sensor ini dirancang untuk mendeteksi suhu lingkungan di sekitar akar tanaman atau di dalam tanah pada berbagai kedalaman. Data suhu tanah yang dikumpulkan oleh sensor ini sangat penting dalam pengelolaan kebun organik, karena suhu tanah dapat mempengaruhi pertumbuhan dan kesehatan tanaman. Sensor suhu tanah membantu petani atau pengelola kebun organik untuk memantau dan mengontrol suhu tanah agar sesuai dengan kebutuhan tanaman tertentu. Informasi suhu tanah ini dapat digunakan untuk mengatur pola penyiraman, menjaga kondisi mikroorganisme tanah, dan mengoptimalkan keberhasilan pertumbuhan tanaman organik secara keseluruhan. Dengan memilih penggunaan sensor suhu tanah, pengelola kebun organik dapat melakukan pemantauan yang akurat dan mengambil langkah-langkah yang diperlukan untuk menjaga suhu tanah yang optimal bagi pertumbuhan tanaman organik.

Menurut Ramirez et al. (2020), sensor suhu DS18B20 merupakan suatu komponen yang dapat mengkonversi perubahan suhu lingkungan menjadi besaran listrik. Sensor tersebut berkomunikasi dengan mikrokontroler melewati sensor digital yang menggunakan 1 *wire*. Kode serial yang dimiliki tipe sensor ini memiliki keunikan yaitu masing-masing sensor mempunyai kode serial yang membolehkan untuk menggunakan DS18B20 lebih dari satu pada satu komunikasi 1 *wire*. Dallas Semikonduktor adalah orang yang menciptakan sensor suhu digital DS18B20. Sensor suhu DS18B20 menetapkan protokol 1 *wire* komunikasi untuk pembacaan suhu. Ketelitian sensor ini mampu membaca 9-12 bit. Sensor DS18B20 dapat dilihat pada gambar 2.4.



Gambar 2. 4 Sensor Suhu Tanah
Sumber: Bagus et al. (2022)

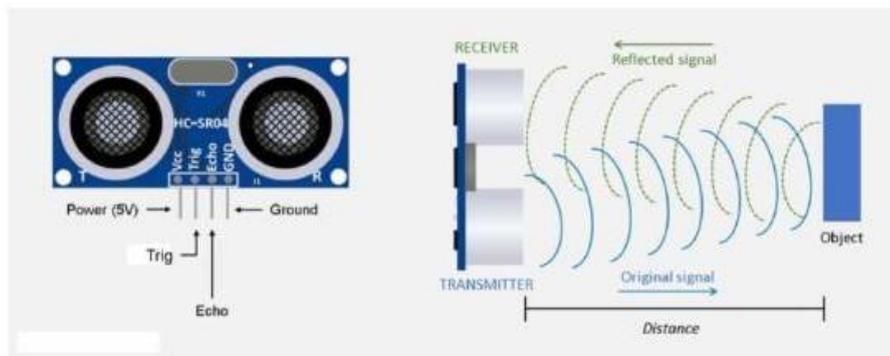
Konfigurasi kaki DS18B20 memiliki 3 pin kaki yang terdiri dari V_s , ground dan data input/output. V_s sendiri berfungsi sebagai tegangan sumber. Tegangan yang dimiliki sensor DS18B20. Sensor suhu tanah digunakan untuk mengukur suhu pada lapisan tanah di kebun organik. Informasi suhu tanah penting dalam memahami kondisi pertumbuhan tanaman, kesehatan mikroorganisme tanah, dan kebutuhan air tanaman. Sensor ini memberikan data suhu yang diperlukan untuk pengendalian suhu yang optimal dalam kebun organik.

Sensor kelembaban tanah adalah perangkat yang digunakan untuk mengukur kadar kelembaban atau tingkat kelembaban dalam tanah. Sensor ini dirancang khusus untuk mendeteksi kandungan air dalam tanah, yang merupakan parameter penting dalam pemantauan dan pengendalian kebun, dan lingkungan. Sensor kelembaban tanah nantinya akan ditempatkan di dalam tanah untuk mengukur kelembaban tanah pada tingkat yang berbeda-beda. Sensor ini memberikan data kelembaban tanah yang diperlukan untuk pengendalian penggunaan air yang efisien dalam kebun organik.



Gambar 2. 5 Sensor Kelembaban Tanah
Sumber: Thombare et al. (2023)

Menurut I. H. Santoso & Irawan (2022), Sensor *Ultrasonic* yang juga dikenal sebagai sensor sonar, adalah jenis sensor yang menggunakan gelombang suara *Ultrasonic* untuk mendeteksi keberadaan objek di depannya dan mengukur jaraknya. Sensor ini sering digunakan dalam aplikasi pengukuran level air dalam tangki atau wadah penyimpanan air salah satunya pada jurnal yang berjudul “Alat Pendeteksi Level Air Otomatis pada Tangki Air Wudhu Masjid Ulil Albab UNSRAT berbasis mikrokontroler” yang disusun oleh Syahputra (2023) jurnal ini bertujuan untuk mengetahui isi air pada tangki stok air yang biasa digunakan untuk wudhu dengan cara memantau ketinggian volume air dengan memanfaatkan sensor *Ultrasonic*. Dengan menggunakan gelombang suara *Ultrasonic*, sensor ini dapat memberikan informasi tentang tingkat atau ketinggian air yang tersedia. Hal ini bermanfaat dalam sistem irigasi otomatis atau pengendalian penggunaan air dalam kebun organik, karena dapat membantu memantau dan mengontrol penggunaan air secara efisien. Pada penelitian ini menggunakan sensor *Ultrasonic* HC-SR04, dapat dilihat pada gambar 2.6.



Gambar 2. 6 Sensor *Ultrasonic* HC-SR04

Sumber: Setiawan et al. (2023)

Spesifikasi Sensor *Ultrasonic* HC-SR04 dapat dilihat pada tabel 2.3.

Tabel 2. 3 Spesifikasi Sensor *Ultrasonic* HC-SR04

Tegangan	5 VDC
Arus	15 mA
Frekuensi Kerja	40 KHz
Jarak Minimum	400 cm (4 meter)
Sudut Pengukuran	15 Derajat
Input Sinyal <i>Trigger</i>	10uS pulsa TTL

Output Sinyal <i>Echo</i>	Sinyal level TTL
Dimensi	45 mm x 20 mm x 15 mm

2.2.10 LCD

LCD adalah jenis tampilan datar atau perangkat optik yang menggunakan teknologi elektronik dan kristal cair untuk mengatur cahaya. Kristal cair tidak menghasilkan cahaya sendiri, tetapi menggunakan lampu latar atau reflektor untuk menciptakan gambar berwarna atau hitam-putih. LCD dapat menampilkan gambar apa pun (seperti pada tampilan komputer) atau gambar tetap dengan sedikit informasi yang dapat ditampilkan atau disembunyikan. Mereka menggunakan teknologi dasar yang sama, hanya beda ukuran elemen tampilannya. LCD dapat berada dalam keadaan menyala (positif) atau mati (negatif), tergantung pada bagaimana polarizer diatur Harini & Lavanya (2022). Dalam sistem kontrol IoT, LCD pada ESP32 memiliki peran penting. LCD ini digunakan sebagai layar yang menampilkan informasi. Dengan menggunakan LCD, ESP32 dapat menampilkan data suhu, kelembaban, status perangkat, atau pesan penting lainnya secara jelas sesuai program yang dibuat di Arduino IDE. Gambar LCD dapat dilihat pada gambar 2.7.



Gambar 2. 7 Liquid Crystal Display (LCD)

Sumber: Harini & Lavanya (2022)

2.2.11 Blynk

Blynk adalah platform IoT untuk *smartphone* iOS atau Android yang digunakan untuk mengendalikan Arduino, Raspberry Pi, dan NodeMCU melalui Internet. Aplikasi ini digunakan untuk membuat antarmuka grafis atau antarmuka mesin manusia (HMI) dengan mengkompilasi dan memberikan alamat yang sesuai pada widget yang tersedia.

Dengan *Blynk*, dapat mengendalikan perangkat keras secara jarak jauh, menampilkan data sensor, menyimpan data, memvisualisasikan-nya. Penggunaan Aplikasi ini dalam konteks kebun organik, untuk memantau dan mengendalikan perangkat seperti sistem irigasi, suhu, kelembaban tanah. Dengan menggunakan *Blynk*, pengguna dapat dengan mudah memonitor kebun organik melalui *smartphone*, di mana pun pengguna berada.



Gambar 2. 8 Logo *Software Blynk*

Sumber: (*Blynk*, 2018)

2.2.12 Lingkungan Kerja Fisik (LKF)

Menurut Sedarmayanti (2019) lingkungan kerja adalah keseluruhan alat perkakas dan bahan yang dihadapi, lingkungan sekitarnya di mana seseorang bekerja, metode kerjanya, serta pengaturan kerjanya baik sebagai perseorangan maupun sebagai kelompok. Jenis lingkungan kerja terbagi menjadi 2 Sedarmayanti (2019) :

1. Lingkungan tempat kerja/Lingkungan kerja fisik (*physical working environment*) adalah semua keadaan berbentuk fisik yang terdapat di sekitar tempat kerja yang dapat mempengaruhi karyawan baik secara langsung maupun secara tidak langsung. Lingkungan kerja fisik sendiri dapat dibagi dalam dua kategori, yakni:
 - a. Lingkungan yang langsung berhubungan dengan karyawan (Seperti: pusat kerja, kursi, meja dan sebagainya).
 - b. Lingkungan perantara atau lingkungan umum dapat juga disebut lingkungan kerja yang mempengaruhi kondisi manusia, misalnya: temperatur, kelembaban, sirkulasi udara, pencahayaan, kebisingan, getaran mekanis, bau tidak sedap, warna, dan lain-lain.
2. Suasana kerja/Lingkungan kerja non fisik (*Non - Physical Working Environment*) adalah semua keadaan yang terjadi yang berkaitan dengan hubungan kerja, baik hubungan dengan atasan maupun hubungan sesama rekan kerja, ataupun hubungan dengan bawahan.

Menurut (Sedarmayanti dalam Wulan, 2011:21) Menyatakan bahwa secara garis besar, jenis lingkungan kerja terbagi menjadi dua faktor yaitu faktor lingkungan kerja fisik dan faktor lingkungan kerja non fisik.

1. Faktor Lingkungan Kerja Fisik

- a. Pewarnaan
- b. Penerangan
- c. Udara
- d. Suara bising
- e. Ruang gerak
- f. Keamanan
- g. Kebersihan

2. Faktor Lingkungan Kerja Non Fisik

- a) Struktur kerja
- b) Tanggung jawab kerja
- c) Perhatian dan dukungan pemimpin
- d) Kerja sama antara kelompok
- e) Kelancaran komunikasi

Indikator-indikator lingkungan kerja menurut Affandi (2018) adalah sebagai berikut :

1. Udara

Mengenai faktor udara ini, yang sering sekali adalah suhu udara dan banyaknya uap air pada udara itu.

2. Warna

Warna merupakan salah satu faktor yang penting untuk memperbesar efisiensi kerja para karyawan, khususnya warna akan mempengaruhi keadaan jiwa mereka dengan memakai warna yang tepat pada dinding ruang dan alat-alat lainnya kegembiraan dan ketenangan bekerja para karyawan akan terpelihara.

3. Pencahayaan

Cahaya penerangan yang cukup memancarkan dengan tepat akan menambah efisiensi kerja para karyawan, karena mereka dapat bekerja dengan lebih cepat lebih sedikit membuat kesalahan dan matanya tak lekas menjadi lelah.

4. Suara

Untuk mengatasi terjadinya kegaduhan, perlu kiranya meletakkan alat-alat yang memiliki suara yang keras, seperti mesin ketik pesawat telepon, parker motor, dan lain-lain. Pada

ruang khusus, sehingga tidak mengganggu pekerja lainnya dalam melaksanakan tugasnya.

2.2.13 Black Box Testing

Black Box Testing adalah teknik pengujian yang berfokus pada spesifikasi fungsional *software*. Salah satu metode *Black Box Testing* adalah *Decision Table Testing* (Pengujian Tabel Keputusan). *Decision Table Testing* (Pengujian Tabel Keputusan) adalah metode pengujian perangkat lunak yang digunakan untuk menguji skenario yang melibatkan kombinasi logika dan keputusan. Dalam metode ini, tabel keputusan digunakan untuk menggambarkan kombinasi dari berbagai kondisi input dan hasil yang diharapkan. Setiap kombinasi kondisi menghasilkan satu atau beberapa aksi atau output.

Pengujian *Black box* mengasumsikan perangkat lunak sebagai "kotak hitam" yang tidak diketahui struktur internalnya. Metode *Black Box* adalah salah satu cara pengujian perangkat lunak yang membahas fungsionalitasnya. Hasil dari pengujian *Black box* berpengaruh pada manajemen sumber daya manusia (SDM) karena hal ini mempengaruhi kepuasan dan kinerja sistem dalam pengambilan keputusan yang baik bagi organisasi atau institusi. Implementasi perangkat lunak membutuhkan rekayasa sistem yang matang dan perencanaan perangkat lunak yang lebih spesifik. Perangkat lunak dalam proses pengujian memerlukan teknik dan optimasi yang cukup intensif agar hasilnya memuaskan.

2.2.14 Metode Prototyping

Menurut Meisak et al. (2022), metode *Prototyping* adalah salah satu pendekatan dalam rekayasa perangkat lunak yang secara langsung mendemonstrasikan bagaimana sebuah perangkat lunak dan keras atau komponen-komponen akan bekerja dalam lingkungannya sebelum tahapan konstruksi aktual dilakukan. *Prototype* merupakan suatu cara yang baik untuk mendapatkan *feedback* mengenai sistem yang diajukan dan menjelaskan bagaimana sistem tersebut tersedia untuk memenuhi kebutuhan informasi pengguna. Adapun beberapa tahapan metode *Prototyping* yaitu Siswidiyanto et al. (2020):

1. *Requirements Gathering and Analysis* (Analisis Kebutuhan)

Pada tahap ini peneliti melakukan analisis terhadap kebutuhan pengguna, baik mengenai *interface*, teknik prosedur maupun jenis alat yang akan digunakan.

2. *Quick Design* (Perancangan *Prototype*)

Pada tahapan ini dilakukan pembuatan desain sederhana berupa gambaran dasar terkait perancangan sistem berdasarkan dari hasil analisis kebutuhan.

3. *Prototype Building* (Bangun *Prototype*)

Pada tahap ini peneliti membangun *prototype* yang telah disesuaikan dengan kebutuhan pengguna.

4. *User Evaluation* (Evaluasi Pengguna)

Pada tahap ini peneliti mengevaluasi kembali *prototype* yang sudah dibangun, mencari tahu apakah *prototype* yang dibangun sudah sesuai dengan kebutuhan pengguna.

5. *Refining Prototype* (Memperbaiki *Prototype*)

pada tahapan ini dilakukan perbaikan *prototype* dari hasil evaluasi pengguna. *Prototype* akan disesuaikan kembali dengan hasil evaluasi pengguna, baik secara *hardware* maupun *software*. Tahapan 4 dan 5 akan terus diulangi hingga *prototype* yang dibangun telah sesuai dengan kebutuhan pengguna.

6. *Product Implementation* (Implementasi Produk)

Pada tahap ini dilakukan pembuatan produk final setelah dilakukannya perbaikan sistem. Dilakukan penggabungan antara *software* dan *hardware* yang telah dibangun agar *prototype* beserta fitur-fiturnya dapat digunakan dan diimplementasikan.

Peneliti memilih *Prototyping* sebagai metode pada penelitian ini karena menurut Pricillia & Zulfachmi (2021), metode ini memiliki berbagai keunggulan. Adapun beberapa keunggulan tersebut, yaitu:

1. Pengguna berpartisipasi aktif dalam pengembangan sistem, sehingga hasil produk pengembangan akan semakin mudah disesuaikan dengan keinginan dan kebutuhan pengguna.
2. Penentuan kebutuhan lebih mudah diwujudkan
3. Mempersingkat waktu pengembangan produk perangkat lunak.
4. Adanya komunikasi yang baik antara peneliti dan pengguna.
5. Peneliti dapat bekerja lebih baik dalam menentukan kebutuhan pengguna.
6. Lebih menghemat waktu pengembangan sistem.
7. Penerapan menjadi lebih mudah karena pengguna mengetahui apa yang diharapkan dari *prototype* yang dibangun.

Oleh karena itu, metode *Prototyping* digunakan dalam penelitian ini untuk merancang dan mengembangkan sistem *monitoring* dan *controlling* kebun organik berbasis IoT secara iteratif.

2.2.15 Metode *Usability Testing*

Usability Testing adalah metode yang digunakan untuk mengukur kinerja, efisiensi, dan kepuasan pengguna terhadap sistem atau produk yang dikembangkan. Dalam konteks penelitian ini, *Usability Testing* dilakukan untuk mengevaluasi kemudahan penggunaan, fungsionalitas, dan kepuasan pengguna terhadap sistem *monitoring* dan *controlling* kebun organik berbasis IoT yang telah dikembangkan. Menurut (Tullis, T. and Albert, B, 2008), hasil pengukuran *usability* memiliki manfaat sebagai berikut:

1. Mendapatkan input yang lebih objektif dari data daripada hanya bergantung pada pendapat subjektif.
2. Memungkinkan perbandingan *usability* antara produk-produk yang berbeda.
3. Mampu mengidentifikasi masalah-masalah yang mungkin timbul.
4. Memberikan perkiraan tentang seberapa baik produk asli akan digunakan.
5. Memberikan contoh konkret kepada manajemen berdasarkan data yang diperoleh.

Adapun beberapa Langkah yang harus dilakukan untuk menerapkan metode *Usability Testing*, yaitu:

1. Komponen *Usability Testing*.

Menurut Krug (2000), komponen *Usability Testing* terdiri dari:

- a. *Learning*: Tingkat keberhasilan dalam menyelesaikan tugas untuk setiap jenis partisipan dan rasio halaman dengan hasil rata-rata kunjungan.
- b. *Efficiency*: Kelompok pengguna dalam mengerjakan tugas yang bermacam-macam.

2. Pemilihan responden *Usability Testing*

Menurut Manik et al. (2021), Jumlah pengguna yang ideal untuk pengujian usability untuk sebuah proyek jika berskala kecil berjumlah tiga atau empat paling banyak. Namun jika proyek tersebut berskala besar maka dibutuhkan minimal 15 pengguna.

3. Pengukuran *usability*

Pengukuran *usability* digunakan untuk menilai apakah interaksi antara pengguna dengan aplikasi atau situs website berjalan dengan baik. Pengukuran dilakukan

dengan mengikuti konsep user testing, dengan menekankan kepada pengukuran bukan kepada pengujian, sebagai berikut:

- a. Menentukan tujuan dan mengeksplorasi pertanyaan
- b. Memilih paradigm dan teknik pengukuran.
- c. Merancang tugas yang akan dijadikan sasaran dalam melakukan pengukuran.
- d. Memilih partisipan dari pengguna untuk mencoba website.
- e. Mempersiapkan kondisi pengukuran.
- f. Merencanakan jalannya pengukuran.
- g. Melakukan evaluasi, analisis dan penyajian data.

4. Tujuan pengukuran *usability*

Pengukuran dilakukan untuk mengidentifikasi masalah dari *usability* yang dapat mempengaruhi antara interaksi sistem dengan pengguna pada hasil perancangan aplikasi. Pengukuran dengan melakukan uji coba perangkat lunak kepada sejumlah partisipan (bertindak sebagai pengguna) dengan melakukan pengamatan. Kemudian partisipan mengisi kuesioner agar dapat memberikan hasil dari gambaran tingkat kepuasan pengguna dalam menggunakan website. Kemudian masukan dari partisipan dapat digunakan sebagai umpan balik untuk melengkapi prasyarat fungsional atau kebutuhan interaksi terhadap pengguna.

5. Teknik pengukuran *usability*

Paradigma untuk melakukan pengukuran adalah *Usability Testing* dengan fokus kepada pengukuran performa dari pengguna melalui sejumlah tugas yang telah dipersiapkan sebelumnya. Teknik pengukuran ini dilakukan oleh pengguna dengan meminta pengguna untuk menjalankan tugas.

Menurut Dumas & Redish (1999), *Usability Testing* dapat mengidentifikasi kelemahan atau masalah dalam desain dan melakukan perbaikan yang diperlukan untuk meningkatkan pengalaman pengguna dan efektivitas sistem tersebut.

2.2.16 System Usability Scale (SUS)

Menurut Lewis (2018), *System Usability Scale (SUS)* merupakan sebuah instrumen kuesioner yang digunakan untuk mengukur tingkat *usability* sebuah sistem atau produk. *System Usability Scale (SUS)* sering digunakan karena penggunaannya yang mudah, kehandalannya, tingkat keakuratan hasilnya (berdasarkan dari koefisien *Cronbach's Alpha* yang melebihi 0,90), dan dapat diterapkan pada berbagai bidang. Terdapat 10 butir pertanyaan pada kuesioner *System Usability Scale (SUS)*, 5 butir dari pertanyaan tersebut

berkonotasi positif (kelebihan pada sistem/produk) dan 5 butir lainnya berkonotasi negatif (kekurangan pada sistem/produk). Respon yang diberikan oleh pengguna akan dikategorikan menjadi skala 1 (Sangat tidak setuju) hingga 5 (Sangat setuju), jangkauan skor pada *System Usability Scale* (SUS) adalah dari 0 hingga 100 pada kelipatan 2,5. Penelitian ini menggunakan *System Usability Scale* (SUS) karena beberapa alasan. Pertama menurut Pal & Vanijja (2020), SUS dapat digunakan secara gratis dan tersedia untuk umum dalam jangka waktu yang lama. Kedua, memiliki sifat psikometri yang sangat baik. Ketiga, telah dilakukan penelitian normatif yang luas terkait SUS, sehingga terdapat berbagai cara untuk menerapkannya.

BAB III

METODE PENELITIAN

3.1 Kerangka Penelitian

Pada penelitian ini akan dilakukan rancang bangun sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT). Agar penelitian dapat berjalan dengan baik dan terarah, maka dibentuk kerangka rencana pada penelitian ini. Kerangka rencana pada penelitian ini terdapat pada tabel 3.1.

Tabel 3. 1 Kerangka Penelitian

Pertanyaan	Penjelasan
Apa	Perancangan dan pembangunan sistem <i>monitoring</i> dan <i>controlling</i> kebun organik berbasis <i>Internet of Things</i> (IoT) menggunakan metode <i>Prototyping</i> dan <i>Usability Testing</i> .
Siapa	Subjek pada penelitian adalah orang yang bertanggung jawab melakukan perawatan kebun organik di Nara Kupu Yogyakarta.
Kapan	<p>Metode kedua adalah metode <i>prototype</i> perancangan. Dalam metode ini, dilakukan perancangan <i>prototype</i> sistem <i>monitoring</i> dan <i>controlling</i> kebun organik berbasis <i>Internet of Things</i> (IoT). Setelah perancangan selesai, dilakukan <i>Usability Testing</i> terhadap <i>prototype</i> yang telah dirancang.</p> <p>Penelitian ini dilakukan dengan menggunakan dua jenis metode. Metode pertama adalah metode <i>Usability Testing</i> yang menggunakan kuesioner <i>Usability</i>. Metode ini digunakan untuk mengukur <i>usability</i> dari <i>prototype</i> yang telah dibangun.</p> <p>Dengan menerapkan kedua metode tersebut, diharapkan dapat diperoleh kesimpulan mengenai <i>usability</i> dan efisiensi penggunaan IoT dalam <i>monitoring</i> dan <i>controlling</i> kebun organik. Skripsi ini berjudul "Rancang Bangun Sistem <i>Monitoring</i> dan <i>Controlling</i> Kebun Organik Berbasis <i>Internet of Things</i> dengan Metode</p>

Pertanyaan	Penjelasan
Dimana	<p><i>Prototyping</i> dan <i>Usability Testing</i> (Studi Kasus: Nara Kupu Yogyakarta)".</p>
Kenapa	<p>Penelitian ini akan dilakukan dalam periode waktu tertentu, yang harus ditentukan oleh peneliti dan pihak terkait. Penelitian ini dilakukan untuk mengoptimalkan pengelolaan kebun organik melalui penerapan teknologi IoT, yang dapat membantu dalam <i>monitoring</i> dan <i>controlling</i> tanaman organik secara efisien.</p>
Bagaimana	<ol style="list-style-type: none"> 1. Membangun prototipe awal sistem <i>monitoring</i> dan <i>controlling</i> kebun organik berbasis IoT, dengan memperhatikan aspek desain dan fungsionalitas. 2. Menggunakan metode <i>Usability Testing</i> untuk mengevaluasi dan memperbaiki sistem berdasarkan feedback dan pengalaman pengguna. 3. Melakukan pengumpulan data mengenai kebutuhan dan preferensi pengguna melalui wawancara, observasi, dan kuesioner. 4. Merancang dan mengimplementasikan sistem <i>monitoring</i> kondisi lingkungan dan tanaman menggunakan sensor-sensor yang sesuai. 5. Membangun sistem pengendalian otomatis berdasarkan data yang dikumpulkan, seperti pengaturan penyiraman, nutrisi, dan pencahayaan. 6. Mengintegrasikan sistem dengan teknologi IoT, sehingga data dapat dikirimkan dan diakses secara online. 7. Melakukan <i>Usability Testing</i> untuk menguji kinerja sistem dan memperbaikinya berdasarkan hasil evaluasi dari pengguna. 8. Menganalisis dan menyajikan hasil penelitian dalam bentuk laporan skripsi, termasuk

Pertanyaan	Penjelasan
	kesimpulan dan rekomendasi untuk pengembangan lebih lanjut.

3.2 Obyek Penelitian

Objek pada penelitian ini adalah Nara Kupu Yogyakarta, sebuah tempat rekreasi berkebun organik yang terletak di kawasan kaki Gunung Merapi, Sleman, DIY. Dengan konsep wisata agrikultur ramah lingkungan, Nara Kupu Yogyakarta menyediakan pengalaman memetik sayur segar secara langsung. Kebun nara kupu Yogyakarta dapat dilihat pada gambar 3.1.



Gambar 3. 1 Kebun Nara Kupu Yogyakarta

Penelitian ini dilakukan untuk merancang sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) dengan menggunakan kombinasi sensor-sensor yang akan mengirimkan hasil pembacaan melalui internet ke aplikasi *Blynk* di *smartphone* secara *real-time*.

3.3 Subjek Penelitian

Subjek penelitian pada penelitian ini terdiri dari tiga responden, yaitu para pengelola kebun organik yang memiliki pengalaman selama 3 tahun dalam mengelola kebun organik. Pemilihan tiga responden yang memiliki pengetahuan mendalam dan pemahaman yang luas tentang kebutuhan *monitoring* dan *controlling* dalam pengelolaan kebun organik memberikan keunggulan dalam mendapatkan wawasan mendalam tentang efektivitas sistem yang dirancang. Meskipun jumlah responden terbatas, fokus pada studi

kasus yang spesifik dan pengambilan data kualitatif, seperti wawancara mendalam, memungkinkan penggalian pemahaman, pengalaman, dan persepsi subjek penelitian secara mendalam. Kemampuan subjek penelitian sebagai pengelola kebun organik yang berpengalaman memberikan perspektif yang berharga dalam mengoptimalkan penggunaan sistem yang dirancang. Penting untuk mengakui keterbatasan jumlah responden dalam laporan penelitian dan menekankan bahwa penelitian ini memberikan wawasan awal dan studi kasus yang spesifik dalam konteks pengelolaan kebun organik. Berikut merupakan karakteristik dan kriteria dari responden yang digunakan pada penelitian ini pada tabel 3.2.

Tabel 3. 2 Karakteristik dan Kriteria

No	Karakteristik	Kriteria
1	Usia	25 - 40 Tahun
2	Jenis	Laki-laki / Perempuan
3	Pengalaman	± 3 Tahun
4	Pekerjaan	Pengelola kebun organik

3.4 Alat Bantu Analisis Data

Digunakan beberapa instrumen dalam penelitian ini yang berfungsi sebagai tools agar mempermudah proses penelitian ini. Adapun instrumen penelitian yang digunakan pada penelitian ini adalah sebagai berikut:

1. Kuesioner *System Usability Scale* (SUS)

Kuesioner *System Usability Scale* (SUS) digunakan untuk mengetahui nilai *usability* pada website M.A. Medical. Sehingga dapat diketahui seberapa optimal rancang bangun sistem *monitoring* dan *controlling* tersebut saat ini seperti yang terdapat dalam lampiran.

2. Website draw.io

Website draw.io digunakan untuk membuat *flowchart* pada penelitian ini.

3. Aplikasi Proteus

Aplikasi Proteus digunakan untuk membuat rangkaian *hardware* pada penelitian ini.

4. Aplikasi *Fusion 360*

Aplikasi *Fusion 360* digunakan untuk membuat desain produk dalam bentuk 3D.

3.5 Jenis Data

Pada penelitian ini digunakan beberapa jenis sumber data. Jenis sumber data yang digunakan yaitu data primer dan data sekunder. Berikut merupakan penjelasan lebih detail terkait kedua jenis sumber data yang digunakan

1. Menurut Hardani et al. (2020), Data primer dalam suatu penelitian diperoleh langsung dari sumbernya dengan melakukan pengukuran, menghitung sendiri dalam bentuk angket, observasi, wawancara dan lain-lain.

2. Data sekunder

Menurut Sugiyono (2018), Data sekunder yaitu sumber data yang tidak langsung memberikan data kepada pengumpul data, misalnya lewat orang lain atau lewat dokumen. Dalam penelitian ini yang menjadi sumber data sekunder adalah sesuai dengan Undang-Undang Ketenagakerjaan, buku, jurnal, artikel yang berkaitan dengan topik penelitian mengenai sistem *monitoring* dan *controlling* berbasis *Internet of Things* (IoT).

3.6 Metode Pengumpulan data

Pada penelitian ini digunakan beberapa metode untuk melakukan pengumpulan data, berikut merupakan penjelasan lebih detail terkait beberapa metode yang digunakan pada penelitian ini:

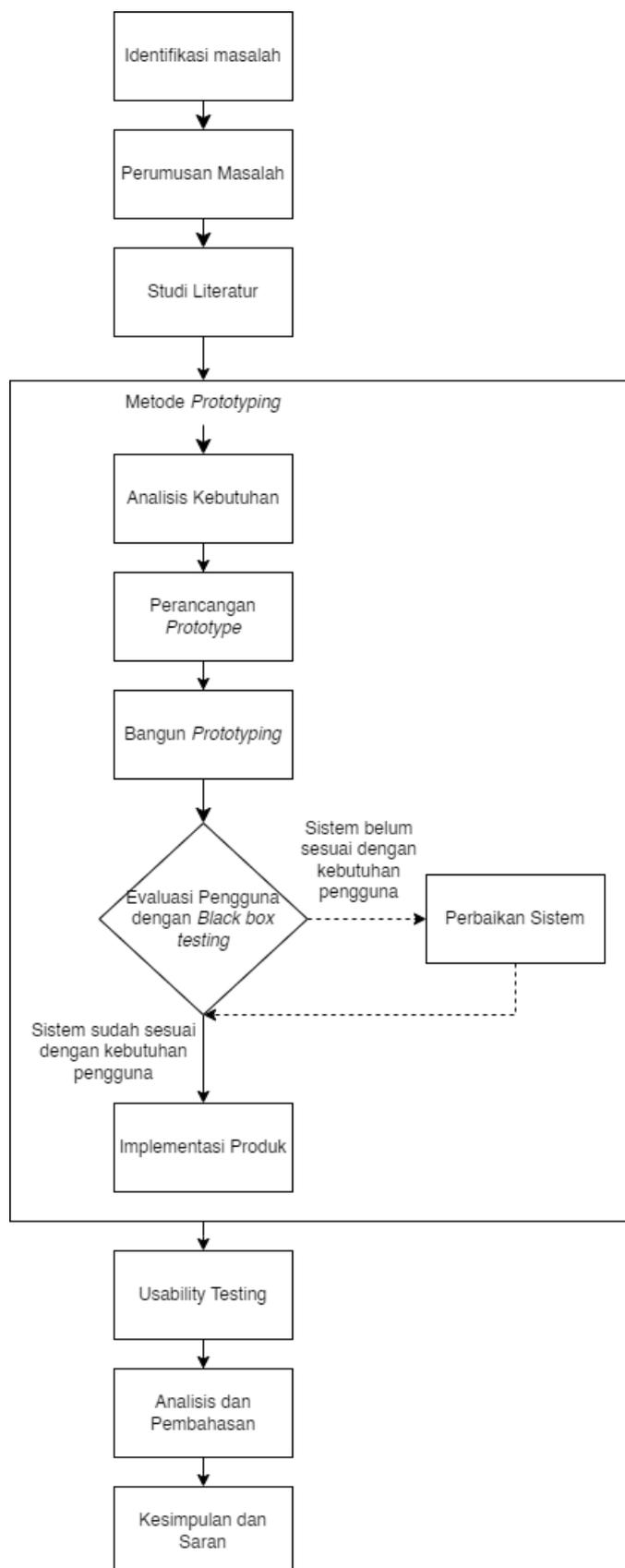
1. Menurut Hardani et al. (2020), Wawancara merupakan situasi sosial antara dua orang, dimana proses psikologis yang terlibat membutuhkan kedua individu secara timbal balik dalam memberikan beragam tanggapan sesuai tujuan penelitian. Dalam wawancara terstruktur, peneliti mengajukan serangkaian pertanyaan standar. Pada penelitian ini wawancara dilakukan sebagai tahap awal untuk mendapatkan informasi secara langsung tentang permasalahan perawatan tanaman, jenis tanaman, proses perawatan, sarana produksi, dan hasil *Usability Testing*. Wawancara dilakukan dengan menggunakan pedoman wawancara yang telah disusun sebelumnya.
2. Observasi menurut Husmaini & Pranomo (2001) adalah pengamatan dengan pencatatan yang sistematis terhadap gejala-gejala yang diteliti. Pada penelitian ini peneliti melakukan observasi secara sistematis pada kebun organik dengan tujuan untuk mengamati dan mencatat gejala-gejala yang diteliti. Observasi ini bertujuan untuk memperoleh pemahaman yang mendalam mengenai interaksi tanaman dengan lingkungan sekitarnya, keseimbangan nutrisi tanah, pertumbuhan

tanaman, serta pengendalian hama dan penyakit. Hasil observasi ini akan menjadi dasar dalam perancangan sistem *monitoring* dan *controlling Internet of Things* yang dapat memantau kondisi kebun organik secara real-time dan mengoptimalkan proses pertanian berkelanjutan.

3. Kuesioner sering menggunakan daftar periksa (checklist) dan skala penilaian. Perangkat ini membantu menyederhanakan dan mengukur perilaku dan sikap responden. Daftar periksa (checklist) adalah daftar perilaku, karakteristik, atau entitas lain yang dicari peneliti. Baik peneliti atau peserta survei hanya memeriksa apakah setiap item dalam daftar diamati, hadir atau benar atau sebaliknya. Menurut Hardani et al. (2020), skala penilaian lebih berguna ketika suatu perilaku perlu dievaluasi biasanya menggunakan skala *Likert*. Pada penelitian ini jenis kuesioner yang digunakan adalah kuesioner *System Usability Scale* (SUS). Kuesioner disusun berdasarkan aspek-aspek yang ingin diteliti, terutama terkait dengan proses merawat tanaman organik dan evaluasi terhadap sistem yang telah diimplementasikan. Kuesioner tersebut terdiri dari 10 pertanyaan yang disesuaikan dengan format kuesioner *System Usability Scale* (SUS). Kuesioner diberikan kepada 4 responden, yaitu seluruh pengelola kebun organik di Nara Kupu Yogyakarta. 4 responden tersebut dipilih karena mereka yang melakukan proses perawatan tanaman setiap harinya.
4. Jurnal ilmiah
Peneliti mencari dan membaca jurnal ilmiah terkait pengembangan sistem *monitoring* dan *controlling* berbasis IoT dalam pertanian organik, fokus pada studi kasus sebelumnya yang telah dilakukan dalam pengaplikasian teknologi IoT dalam pertanian organik, khususnya yang berhubungan dengan pemantauan dan pengendalian perawatan tanaman.
5. *Internet browsing*
Peneliti mencari sumber informasi terkini, artikel, dan laporan terkait penggunaan teknologi IoT dalam pertanian organik serta mengeksplorasi *platform* dan forum online yang berhubungan dengan pertanian organik dan IoT untuk mendapatkan wawasan dan masukan dari komunitas yang terlibat dalam bidang tersebut.

3.7 Alur Penelitian

Alur penelitian merupakan langkah-langkah yang akan dilakukan oleh peneliti untuk mencapai tujuan penelitian. Alur penelitian dapat dilihat pada pada gambar 3.2



Gambar 3. 2 Alur Penelitian

3.8 Tahapan penelitian

Berikut merupakan penjelasan dari tahapan pada penelitian ini:

1. Identifikasi Masalah

Pada tahap ini, peneliti melakukan wawancara dan observasi langsung di Nara Kupu Yogyakarta untuk mengetahui permasalahan yang dialami oleh pengguna. Identifikasi masalah dilakukan kepada 3 responden yang merupakan pengelola kebun organik di Nara Kupu Yogyakarta.

2. Perumusan Masalah

Pada tahap ini masalah yang telah ditemukan sebelumnya akan dikerucutkan agar lebih spesifik, dengan tujuan untuk menyelesaikan rumusan masalah tersebut pada penelitian ini.

3. Studi Literatur

Studi literatur dilakukan untuk mencari informasi yang relevan dengan masalah yang akan diselesaikan pada penelitian ini dan mengkaji beberapa teori dasar yang relevan dengan masalah yang diteliti.

4. Metode *Prototyping*

Metode *Prototyping* untuk menyelesaikan permasalahan yang telah dikumpulkan. Metode *Prototyping* memiliki beberapa tahapan utama, berikut merupakan penjelasan lebih detail terkait tahapan tersebut.

- a. Analisis Kebutuhan

Pada tahap ini peneliti mengumpulkan dan menganalisis kebutuhan dari pengguna melalui wawancara dan observasi langsung di kebun organik Nara Kupu Yogyakarta. Tujuan dilakukan tahapan ini adalah agar *prototype* yang dibangun dapat disesuaikan dengan kebutuhan pengguna.

- b. Perancangan *Prototype*

Pada tahap ini dilakukan perancangan sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) yang akan dibuat. Peneliti akan membuat gambaran sederhana sebelum melakukan pembangunan *Prototyping*.

- c. Bangun *Prototyping*

Pada tahap ini peneliti melakukan pembuatan produk, mulai dari menentukan perangkat keras dan lunak yang akan digunakan,

perancangan sistem kerja *prototype*, pembangunan *hardware prototype* hingga proses pengkodean (*Coding*).

d. Evaluasi Pengguna

Pada tahap ini peneliti melakukan evaluasi pengguna dengan menggunakan metode *Black Box Testing* dengan pengelola kebun Nara Kupu Yogyakarta, dilakukan penyesuaian spesifikasi dan fitur sistem dengan kebutuhan pengguna. Apabila ada hal yang masih kurang sesuai dengan kebutuhan pengguna, maka akan dilakukan perbaikan sistem hingga sistem dapat diterima oleh pengguna.

e. Perbaikan Sistem

Pada tahap ini dilakukan perbaikan sistem untuk menyesuaikan dengan kebutuhan pengguna. Tahapan perbaikan sistem dan evaluasi pengguna akan terus berulang hingga sistem telah sesuai dengan kebutuhan pengguna.

f. Implementasi Produk

Setelah melakukan evaluasi pengguna dan perbaikan sistem, dilakukan pembuatan pembangunan *prototype* secara final, agar sistem beserta fiturnya dapat digunakan dan diimplementasi.

5. Metode *Usability Testing*

Metode *Usability Testing* digunakan untuk mengukur tingkat *usability prototype* yang telah dibuat melalui tanggapan pengguna. Penelitian ini menggunakan pendekatan *System Usability Scale (SUS)* sebagai format untuk kuesioner yang diberikan kepada pengguna. Pengguna yang diberikan kuesioner berjumlah 3.

6. Analisis dan Pembahasan

Pada tahap analisis dan pembahasan, peneliti akan melakukan proses pengumpulan seluruh data dan membandingkannya, sehingga didapatkan hasil yang dapat dijadikan sebagai acuan perkembangan berdasarkan pada pendekatan *Prototyping* dan *Usability Testing*.

7. Kesimpulan dan Saran

Tahap kesimpulan dan saran merupakan tahapan terakhir pada penelitian ini, dimana pada tahap ini peneliti melakukan penarikan kesimpulan berdasarkan hasil data dan analisis yang telah dilakukan pada tahap sebelumnya. Kesimpulan yang dihasilkan berasal dari data dan analisis yang berkaitan dengan tanggapan

pengguna terhadap *prototype* yang telah dibuat serta tingkat kepuasan dan kemudahan penggunaan sebagai solusi masalah yang terdapat di kebun organik Nara Kupu Yogyakarta. Pada tahap ini peneliti sekaligus memberikan saran untuk penelitian selanjutnya yang merupakan kekurangan dari penelitian ini dengan harapan tidak akan terjadi pada penelitian selanjutnya atau sebagai acuan penelitian selanjutnya agar lebih maksimal.

3.9 Pembangunan Model dan Perancangan

Pada penelitian ini, pembangunan model dan perancangan dilakukan dengan menerapkan metode *Prototyping* dan *Usability Testing*. Berikut merupakan tahapan dari pembangunan model dan perancangan.

1. Metode *Prototyping*

Pada tahapan ini, peneliti melakukan analisis kebutuhan, pembangunan *prototype*, evaluasi pengguna, dan implementasi produk. Analisis kebutuhan dilakukan dengan melakukan wawancara dengan pengelola kebun, observasi langsung, dan studi literatur. Pada tahap ini peneliti mengumpulkan dan memilih kebutuhan fitur, *software*, dan *hardware* yang digunakan pada pembangunan sistem. Lalu peneliti akan melakukan pembangunan dan perancangan *hardware* dan dilanjutkan dengan proses pengkodean (*coding*). Setelah itu, peneliti menggunakan metode *Usability Testing* untuk melakukan evaluasi pengguna untuk menyesuaikan spesifikasi *Prototyping* yang dibangun dengan kebutuhan pengguna. Jika pada sistem didapatkan hal-hal yang kurang sesuai, peneliti akan melakukan perbaikan desain. Setelah melakukan evaluasi pengguna, hal selanjutnya adalah implementasi sistem.

2. Metode *Usability Testing*

Metode *Usability Testing* digunakan untuk mengukur tingkat *usability* dari *prototype* yang telah dibangun. Jenis kuesioner yang digunakan adalah *System Usability Scale* (SUS). Peneliti memilih 3 responden yang merupakan pengelola kebun organik Nara Kupu Yogyakarta. Responden tersebut akan diberikan dua kuesioner dengan 10 butir pertanyaan, dan memilih salah satu dari 5 jawaban, 5 jawaban itu adalah sangat tidak setuju, ragu-ragu, setuju, dan sangat setuju. Skor masing-masing jawaban mulai dari 1 (Sangat tidak setuju) hingga 5 (Sangat setuju). Setelah Melakukan pengumpulan data dari responden kemudian data

tersebut dihitung Menggunakan perhitungan *System Usability Testing* (SUS). Kuesioner yang diberikan kepada responden dapat dilihat pada tabel 3. 3.

Tabel 3. 3 Kuesioner *System Usability Scale* (SUS)

No.	Pertanyaan	1	2	3	4	5
1	Saya akan senang untuk menggunakan sistem ini secara sering.					
2	Saya merasa sistem ini rumit untuk digunakan.					
3	Saya merasa sistem ini mudah digunakan.					
4	Saya membutuhkan bantuan dari orang lain dalam Menggunakan sistem ini.					
5	Saya merasa fitur-fitur sistem ini diintegrasikan dengan baik.					
6	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.					
7	Saya merasa orang lain akan memahami sistem ini dengan cepat.					
8	Saya merasa sistem ini membingungkan.					
9	Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.					
10	Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.					

Skala:

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju

3.10 Rumus dan Teori

Berikut merupakan rumus dan teori yang terdapat pada penelitian ini.

3.10.1 *System Usability Scale* (SUS)

Menurut Horsky et al. (2010), pembuatan kuesioner *System Usability Scale* dilakukan dengan membuat 10 butir pertanyaan, dengan angka ganjil yang berkonotasi positif (kepuasan pengguna) dan angka genap yang berkonotasi negatif (ketidakpuasan pengguna). Jawaban yang diberikan oleh pengguna berjangka dari skala 1 (Sangat tidak setuju) hingga 5 (Sangat setuju). Setelah mendapatkan jawaban dari kuesioner tersebut, selanjutnya dilakukan perhitungan skor. Berikut merupakan rumus dari perhitungan tersebut.

$$((P1 - 1) + (P3 - 1) + (P5 - 1) + (P7 - 1) + (P9 - 1) + (5 - P2) + (5 - P4) + (5 - P6) + (5 - P8) + (5 - P10)) \times 2,5 \dots (1)$$

Keterangan:

P1 – P10 = Jawaban pertanyaan 1 hingga 10.

Selanjutnya, mencari nilai rata-rata dari skor SUS dari setiap responden dengan cara menjumlahkan semua skor dan membaginya dengan jumlah responden. Rumus untuk mendapatkan rata-rata skor SUS adalah sebagai berikut.

$$\bar{x} = \frac{\sum x}{n} \dots (2)$$

Di mana \bar{x} merupakan nilai rata-rata skor, $\sum x$ adalah jumlah dari semua skor, dan n adalah jumlah responden. Secara garis besar, jika hasil yang didapatkan lebih dari nilai 68, maka sistem yang diuji dapat dikatakan memiliki tingkat *usability* diatas rata-rata. Hasil dari penilaian tersebut akan digunakan untuk menilai hasil kuesioner *System Usability Scale* (SUS) pada *prototype* yang telah dibangun untuk mengukur tingkat *usability*.

BAB IV

PEMBANGUNAN SISTEM

4.1 Analisis kebutuhan

Analisis kebutuhan dilakukan dengan melakukan wawancara dengan pengelola kebun organik Nara Kupu Yogyakarta, observasi langsung, dan studi literatur. Tahapan ini dilakukan untuk mengkaji seluruh kebutuhan yang akan diimplementasikan pada sistem yang dibangun, agar sesuai dengan kebutuhan pengguna. Berikut merupakan poin-poin penting yang didapatkan pada analisis kebutuhan.

4.1.1 Kebutuhan Pengguna

Setelah peneliti mengumpulkan data dari wawancara, observasi langsung, dan studi literatur mengenai kebutuhan untuk melakukan proses perawatan kebun organik, berikut merupakan kebutuhan pengguna pada kebun organik Nara Kupu Yogyakarta pada tabel 4.1.

Tabel 4. 1 Kebutuhan Pengguna

Kondisi saat ini	Masalah yang ditimbulkan	Solusi
	Keterbatasan tenaga kerja untuk melakukan perawatan	Mengimplementasikan sistem <i>monitoring</i> dan <i>controlling</i> berbasis <i>Internet of Things</i> (IoT) yang melakukan proses perawatan kebun organik secara automasi yang berdasarkan dari penjadwalan waktu dan pembacaan sensor untuk mengurangi beban kerja pengelola kebun.
Penggunaan metode konvensional dalam proses perawatan.	Penurunan intensitas perawatan	Menggunakan sistem <i>monitoring</i> dan <i>controlling</i> kebun organik berbasis <i>Internet of Things</i> (IoT) untuk melakukan automasi perawatan kebun secara konsisten, akurat, dan tepat waktu dengan menggunakan penjadwalan waktu, dan pembacaan sensor demi menjaga intensitas perawatan kebun organik.
	Kurangnya informasi mengenai kondisi lingkungan kebun organik.	Mengimplementasikan sensor suhu tanah dan kelembaban tanah untuk memantau kondisi lingkungan kebun organik dan

Kondisi saat ini	Masalah yang ditimbulkan	Solusi
		menggunakan aplikasi <i>Blynk</i> untuk memantau data pembacaan sensor-sensor tersebut.

4.1.2 Kebutuhan Dasar Sistem

Pada tahapan ini, peneliti mengumpulkan informasi mengenai kebutuhan dasar proses perawatan kebun organik yang dibutuhkan dalam merancang sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT).

4.1.1.1 Penetapan Batas Suhu dan Kelembaban Tanah

Dalam melakukan proses perawatan kebun organik, suhu dan kelembaban tanah merupakan dua variabel utama dalam produktivitas pertumbuhan tanaman. Maka, diperlukan fitur automasi penyiraman air untuk memastikan bahwa suhu dan kelembaban tanah selalu optimal bagi pertumbuhan tanaman. Berikut merupakan batas suhu dan kelembaban tanah yang diterapkan pada sistem ini pada tabel 4.2.

Tabel 4. 2 Pembatasan Suhu dan Kelembaban Tanah

Variabel	Penetapan Batas
Suhu	31°C
Kelembaban	50%

Menurut Asih (2018), melalui metode *fuzzification* suhu dan kelembaban tanah, didapatkan bahwa batas suhu tanah yang optimal untuk pertumbuhan tanaman adalah 31°C. Sedangkan untuk kelembaban tanah, batas yang optimal untuk pertumbuhan tanaman adalah 50%.

4.1.1.2 Penjadwalan Waktu Penyiraman Air

Waktu penyiraman air pada tanaman sangat berpengaruh pada produktivitas pertumbuhan tanaman. Oleh karena itu, dibutuhkan fitur automasi penyiraman air pada sistem *monitoring* dan *controlling* kebun organik. Berikut merupakan penjadwalan waktu penyiraman air pada sistem ini pada tabel 4.3.

Tabel 4. 3 Penjadwalan Waktu Penyiraman Air

Frekuensi Penyiraman	Waktu Penyiraman
Pertama	Pukul 06.00
kedua	Pukul 18.00

Menurut Seran et al. (2022), penyiraman rutin sehari dua kali pada pagi dan sore hari menyebabkan pertumbuhan tanaman yang optimal. Jumlah air yang dibutuhkan

pertanaman adalah 155 ml dalam sekali penyiraman. Penyiraman dilakukan saat pagi hari (saat suhu masih dingin) untuk menyiapkan tanaman dari kenaikan suhu yang akan terjadi pada siang hari. Sedangkan penyiraman pada sore hari dilakukan agar tanaman dapat menyimpan air tersebut hingga penyiraman berikutnya. Peneliti tidak melakukan penyiraman pada siang hari karena air tersebut akan ter-evaporasi, sehingga kuantitas air yang didapatkan oleh tanaman tidak akan optimal.

4.1.1.3 Penjadwalan Waktu Penyiraman Pupuk Cair

Pupuk cair merupakan elemen yang penting dalam melakukan proses perawatan tanaman karena berisi berbagai nutrisi penting untuk perawatan tanaman. Oleh karena itu, dibutuhkan fitur automasi penyiraman pupuk cair pada sistem *monitoring* dan *controlling* kebun organik. Menurut Gumelar & Yunus (2021) perlakuan frekuensi penyiraman pupuk organik cair sebanyak 5 ml/tanaman pada dua minggu sekali memberikan efek yang paling optimal dalam pertumbuhan tanaman.

4.1.1.4 Penjadwalan Waktu Penyiraman Pestisida

Proses perawatan tanaman akan lebih optimal jika memberikan pestisida agar tanaman terhindar dari hama. Frekuensi dan jumlah penyiraman yang dilakukan pun juga berpengaruh besar pada proses perawatan tanaman, jika frekuensi dan jumlah penyiraman terlalu sedikit dapat menyebabkan masih timbulnya hama, sedangkan jika frekuensi dan jumlah penyiraman terlalu banyak dapat menyebabkan tanaman menjadi layu/mati. Oleh karena itu, menurut Nboyine et al. (2022) jumlah dan frekuensi penyiraman yang memberikan hasil paling optimal adalah 10ml/tanaman pada 1 × seminggu.

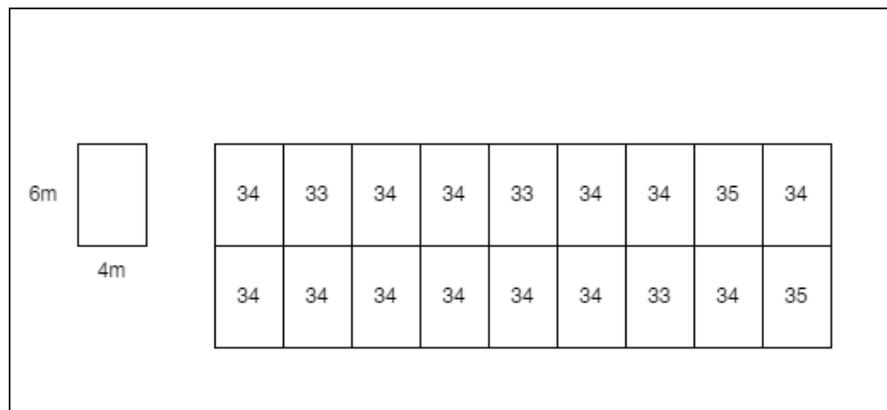
4.1.3 Pengambilan Data Lingkungan Kerja Fisik (LKF) Kebun Organik

Pengambilan data Lingkungan Kerja Fisik (LKF) dilakukan pada kebun organik Nara Kupu Yogyakarta dengan luas kebun sebesar 12 × 36 m (432 m²). Peneliti mengambil data mengenai suhu dan kelembaban tanah dengan menanam sensor-sensor di tanah kebun organik tersebut. Pengambilan data dilakukan pada 18 titik, masing-masing titik pengambilan memiliki dimensi 6 × 4 m (24 m²). Pengambilan data LKF dilakukan untuk mengetahui kondisi tanah kebun organik sebelum diterapkannya sistem *monitoring* dan *controlling*, serta untuk merencanakan tata letak sensor-sensor pada sistem ini. Pengambilan data dilakukan pada pukul 10.00 hingga 15.00 selama satu minggu. Peneliti memilih melakukan pengambilan data pada rentang waktu tersebut karena merupakan

jam-jam dimana sinar matahari paling terik pada daerah kebun organik Nara Kupu Yogyakarta. Berikut merupakan hasil dari pengambilan data LKF.

4.1.3.1 Pengambilan Data Suhu Tanah

Didapatkan bahwa suhu tanah paling tinggi terjadi pada pukul 14.00 siang. Berikut merupakan hasil pengambilan data tersebut pada gambar 4.1.

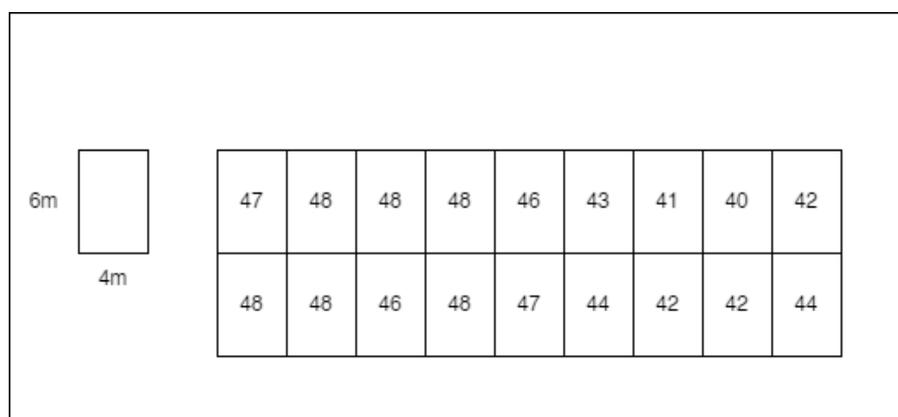


Gambar 4. 1 Hasil Pengambilan Data LKF Suhu Tanah

Berdasarkan dari hasil yang didapatkan, rentan nilai suhu yang didapatkan adalah sekitar 33 hingga 35°C. Dapat disimpulkan bahwa tidak terjadi banyak volatilitas dalam pembacaan suhu tanah, hal tersebut akan dijadikan sebagai acuan dalam proyeksi tata letak alat.

4.1.3.2 Pengambilan Data Kelembaban Tanah

Didapatkan bahwa kelembaban tanah paling rendah terjadi pada pukul 14.00 siang. Berikut merupakan hasil pengambilan data tersebut pada gambar 4.2.



Gambar 4. 2 Hasil Pengambilan Data LKF Kelembaban Tanah

Berdasarkan dari hasil yang didapatkan, rentan nilai kelembaban yang didapatkan adalah sekitar 40 hingga 48°C. Dibandingkan dengan hasil LKF suhu tanah, didapatkan bahwa terjadi banyak volatilitas dalam pembacaan kelembaban tanah, hal tersebut akan dijadikan sebagai acuan dalam proyeksi tata letak alat.

4.1.4 Kebutuhan *Software*

Berikut daftar kebutuhan *software* dari pengembangan sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) di Kebun Nara Kupu Yogyakarta:

Tabel 4. 4 Kebutuhan *Software*

No.	<i>Software</i>	Deskripsi
1.	Arduino IDE	Program Arduino IDE untuk mengendalikan <i>microcontroller</i> dan berkomunikasi dengan sensor dan perangkat lainnya.
2.	Aplikasi <i>Blynk</i>	Aplikasi yang terhubung dengan sistem untuk memantau dan mengontrol kebun organik melalui <i>smartphone</i> atau tablet.
3.	<i>Web Server Blynk</i>	Server untuk menerima dan mengirim data dari dan ke sistem secara online.

4.1.5 Kebutuhan *Hardware*

Berikut daftar kebutuhan *hardware* dari pengembangan sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) di Kebun Nara Kupu Yogyakarta:

Tabel 4. 5 Kebutuhan *Hardware*

<i>Hardware</i>	Deskripsi	Jumlah
<i>Microcontroller</i> ESP32	Mengendalikan dan menghubungkan sensor dan perangkat lainnya.	1
Sensor Kelembaban Tanah	Sensor untuk mengukur kelembaban tanah dan memberikan informasi tentang kebutuhan penyiraman.	1
Sensor Suhu Tanah DS18B20	Sensor untuk mengukur suhu tanah dan	1
Sensor <i>Ultrasonic</i>	Sensor <i>Ultrasonic</i> untuk mendeteksi ketinggian air dalam bak air atau reservoir.	3
Pompa Air	3 buah pompa untuk memberikan pasokan air, pupuk, dan pestisida pada penyiraman tanaman.	3
LCD	Layar LCD untuk menampilkan informasi seperti kelembaban tanah, suhu tanah, dan status sistem.	1
<i>Smartphone</i>	Perangkat untuk memantau kondisi kebun secara real-time melalui aplikasi yang terhubung dengan sistem.	1

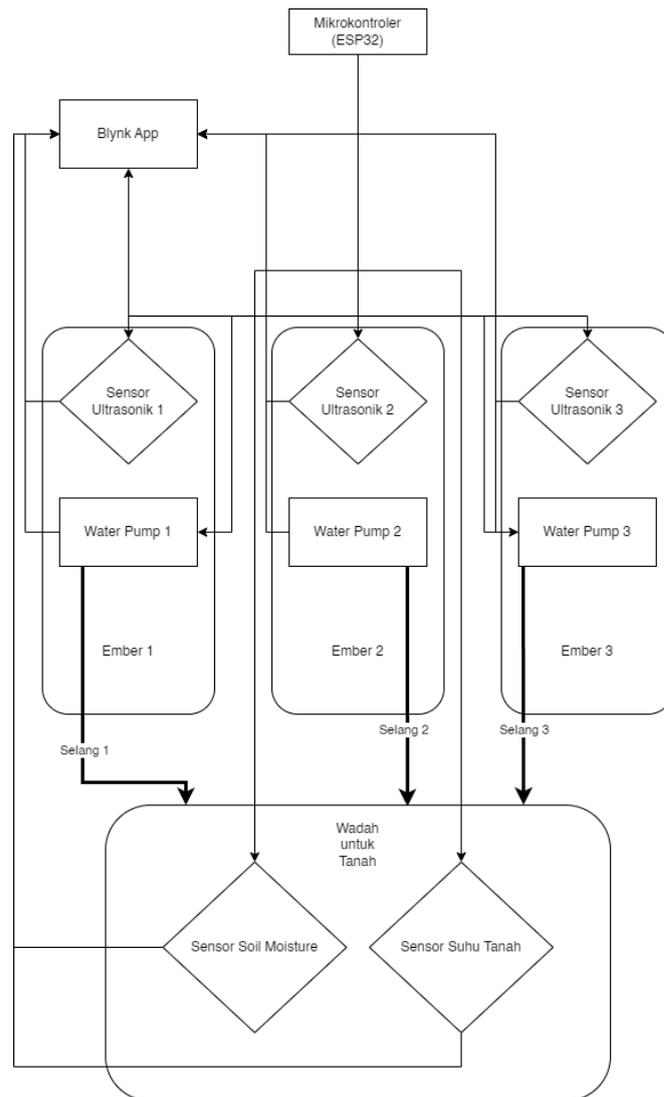
<i>Hardware</i>	Deskripsi	Jumlah
<i>Adapter 12V</i>	Sebagai <i>adapter</i> dari stopkontak/sumber daya untuk 3 pompa.	1
<i>Adapter 5V</i>	Sebagai <i>adapter</i> dari stopkontak/sumber daya untuk Modul ESP32 dan sensor-sensor lainnya.	1
Kabel Jumper	Sebagai penghubung antara sensor dan komponen lainnya pada sistem ini.	16
<i>Resistor</i>	Membatasi/memperkecil aliran arus pada pompa.	1
<i>PCB Board</i>	Penghubung antara ESP32 dengan komponen lainnya pada sistem ini	1
<i>Relay 4 Channel</i>	Penghubung antara ESP32 dengan 3 Pompa air karena membutuhkan daya tambahan.	1
PCF8574 i2c	Sebagai perpanjangan dari <i>input</i> dan <i>output</i> untuk bus i2C.	1
Selang Air	Digunakan untuk menyalurkan air, pupuk cair, dan pestisida dari bak penyimpanan menuju ke tanah.	6 Meter
Bak Penampung	Digunakan sebagai penampung untuk air, pupuk cair, dan pestisida.	3

4.2 Perancangan *Prototype*

Pada tahapan ini, dilakukan perancangan sistem *prototype monitoring* dan *controlling* kebun organik yang berisi mengenai perancangan alur sistem, fitur, *input* dan *output*.

4.3.1 Perancangan Alur *Hardware* pada *Prototype*

Tahapan ini dilakukan untuk memberikan gambaran awal mengenai alur *Hardware* pada *Prototype* sebelum melakukan pembangunan. Berikut merupakan perancangan Alur *Hardware* pada gambar 4.3.



Gambar 4. 3 *flowchart* Perancangan Sistem

4.3.2 Perancangan Fitur *Prototyping*

Setelah peneliti mengumpulkan data dari wawancara, observasi langsung, dan studi literatur mengenai kebutuhan fitur untuk melakukan proses perawatan kebun organik, berikut merupakan fitur yang dipilih untuk diterapkan pada *prototype monitoring* dan *controlling* kebun organik yang terdapat pada *flowchart* diatas.

1. Penyiraman Air Rutin

Proses penyiraman air dilakukan setiap hari dengan penjadwalan 2 kali sehari (jam 6 pagi dan sore), Jumlah air yang akan diberikan pada setiap penyiraman adalah sekitar 155 ml/tanaman.

2. Penyiraman Air ketika Kelembaban Kurang dari Batas

proses penyiraman air juga akan dilakukan jika sensor mendeteksi bahwa kelembaban tanah kurang dari nilai yang sudah ditentukan (50%), penyiraman akan dilakukan hingga kelembaban telah memenuhi batas. Proses penyiraman air akan selalu disertai dengan pengiriman notifikasi ke *Smartphone* pengguna melalui aplikasi *Blynk*.

3. Penyiraman Air ketika Suhu melebihi Batas

proses penyiraman air juga akan dilakukan jika sensor mendeteksi bahwa suhu tanah melebihi dari nilai yang sudah ditentukan (31°C), penyiraman akan dilakukan hingga suhu telah memenuhi batas. Proses penyiraman air akan selalu disertai dengan pengiriman notifikasi ke *Smartphone* pengguna melalui aplikasi *Blynk*.

4. Penyiraman Pupuk Cair

Proses penyiraman pupuk cair dilakukan setiap 2 minggu sekali dalam sebulan, jumlah pupuk cair yang akan diberikan adalah 5 ml/tanaman. Proses penyiraman pupuk cair akan selalu disertai dengan pengiriman notifikasi ke *Smartphone* pengguna melalui aplikasi *Blynk*.

5. Penyiraman Pestisida

Proses penyiraman pestisida dilakukan setiap minggu sekali, Jumlah pestisida yang akan diberikan adalah 10ml/tanaman. Proses penyiraman pestisida akan selalu disertai dengan pengiriman notifikasi ke *Smartphone* pengguna melalui aplikasi *Blynk*.

6. Pemantauan Bak Air

Pemantauan bak air dilakukan untuk memudahkan pengguna untuk melakukan proses pengecekan level dan isi ulang air yang digunakan untuk melakukan penyiraman air pada tanaman. Jika jumlah air pada bak sudah hampir habis (± 5 cm dari dasar), maka sistem akan mengirimkan notifikasi ke *Smartphone* pengguna melalui aplikasi *Blynk* untuk mengingatkan pengguna untuk melakukan pengisian ulang.

7. Pemantauan Bak Pupuk Cair

Pemantauan bak pupuk cair dilakukan untuk memudahkan pengguna untuk melakukan proses pengecekan level dan isi ulang pupuk cair yang digunakan untuk melakukan penyiraman pupuk cair pada tanaman. Jika jumlah pupuk cair pada bak sudah hampir habis (± 5 cm dari dasar), maka sistem akan mengirimkan

notifikasi ke *Smartphone* pengguna melalui aplikasi *Blynk* untuk mengingatkan pengguna untuk melakukan pengisian ulang.

8. Pemantauan Bak Pestisida

Pemantauan bak pestisida dilakukan untuk memudahkan pengguna untuk melakukan proses pengecekan level dan isi ulang pestisida yang digunakan untuk melakukan penyiraman pestisida pada tanaman. Jika jumlah pestisida pada bak sudah hampir habis (± 5 cm dari dasar), maka sistem akan mengirimkan notifikasi ke *Smartphone* pengguna melalui aplikasi *Blynk* untuk mengingatkan pengguna untuk melakukan pengisian ulang.

4.2.3 Perancangan *Input*

Didapatkan perancangan *input* dari penerapan *Internet of Things* pada sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things (IoT)* di Kebun Nara Kupu Yogyakarta:

1. Sensor suhu tanah.
2. Sensor kelembaban tanah.
3. Sensor *Ultrasonic*.
4. Perangkat IoT.
5. Sumber daya energi.
6. Sistem pemantauan dan kontrol.

4.2.4 Perancangan *Output*

Didapatkan perancangan *output* dari pengembangan sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things (IoT)* di Kebun Nara Kupu Yogyakarta:

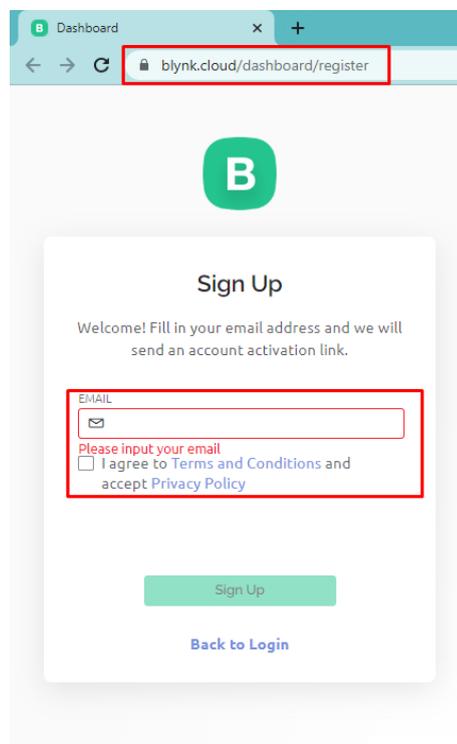
1. Data suhu tanah.
2. Data kelembaban tanah.
3. Data hasil pengukuran sensor *Ultrasonic*.
4. Integrasi data sensor ke dalam sistem pemantauan dan kontrol.
5. Akses data yang diperoleh melalui perangkat IoT.
6. Informasi pengambilan keputusan berdasarkan data yang terkumpul.

4.3 Bangun Prototyping

Pada tahap ini, peneliti melakukan pembangunan *hardware & software*. Berikut merupakan proses pembangunan sistem *prototype monitoring* dan *controlling* kebun organik pada penelitian ini.

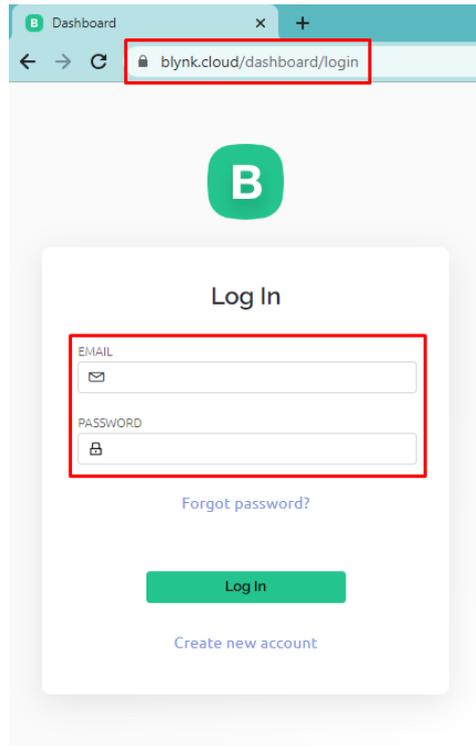
4.3.1 Pengaturan Awal Blynk Cloud

Langkah Pertama, membuka browser web dan mengunjungi situs *web Blynk* di <https://Blynk.io/>. Pada halaman beranda, klik tombol "Sign Up" atau "Daftar" untuk membuat akun baru. Lalu mengisi formulir pendaftaran dengan informasi yang diminta, seperti pada gambar 4.4.

The image shows a web browser window with the address bar displaying "blynk.cloud/dashboard/register". The page features the Blynk logo (a green circle with a white 'B') at the top. Below the logo is a "Sign Up" heading followed by the text "Welcome! Fill in your email address and we will send an account activation link." There is a red rectangular box highlighting the registration form, which includes an "EMAIL" label, an email input field with a red border, a "Please input your email" error message, and a checkbox labeled "I agree to Terms and Conditions and accept Privacy Policy". At the bottom of the form is a green "Sign Up" button and a blue "Back to Login" link.

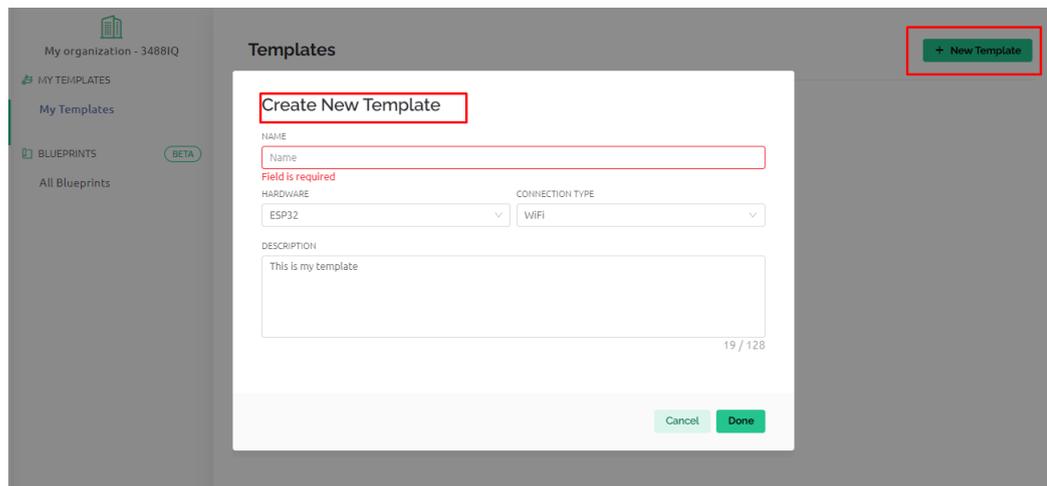
Gambar 4. 4 Form Pendaftaran *Blynk*

Setelah akun telah aktif, kembali ke situs web *Blynk*. Lalu ke halaman "Login" dan memasukkan alamat *E-mail* serta kata sandi yang telah didaftarkan sebelumnya. Halaman *login* dapat dilihat pada gambar 4.5.



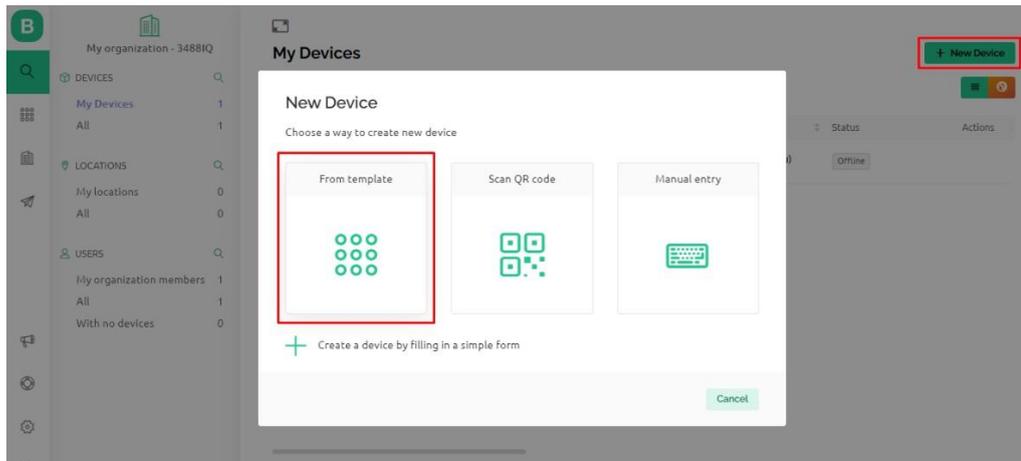
Gambar 4. 5 Form Pendaftaran *Blynk*

Setelah berhasil masuk, klik menu “*Templates*”, lalu klik tombol “*Create New Template*”, lengkapi *form* seperti nama proyek dan jenis perangkat yang akan digunakan (seperti Arduino, Raspberry Pi, atau ESP8266) detail form dapat dilihat pada gambar 4.6.



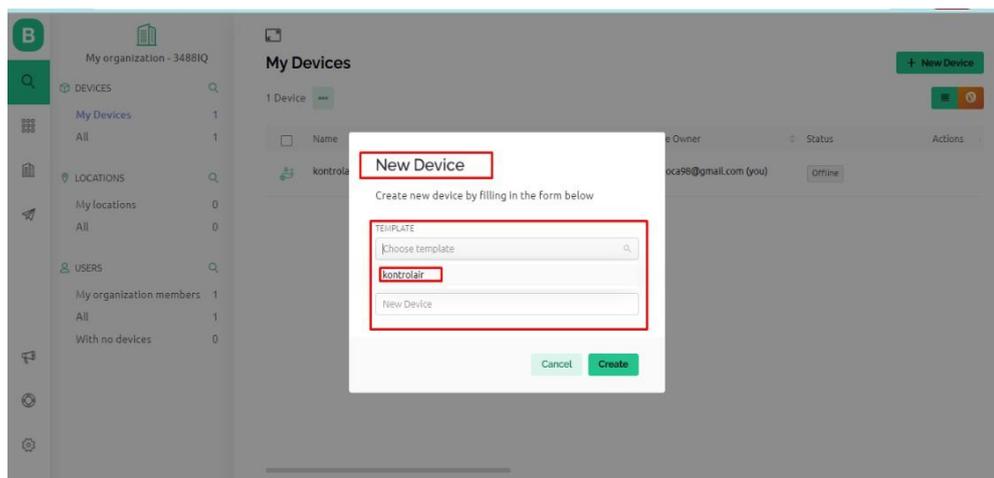
Gambar 4. 6 Form Create New Templates

Selanjutnya setelah selesai membuat *templates*, Klik menu icon “*Search*” di sidebar *Blynk* lalu klik tombol “*Create New Project*” atau “*Buat Proyek Baru*”, selanjutnya klik opsi “*Form template*” seperti pada gambar 4.7.



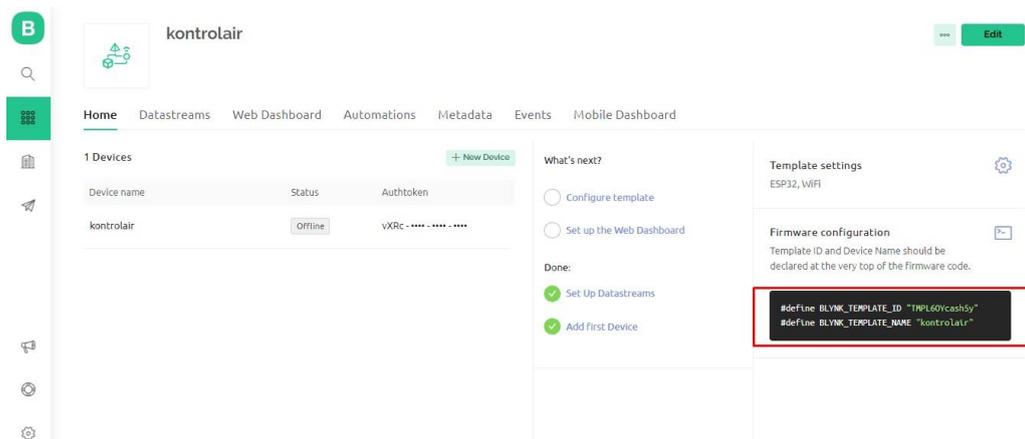
Gambar 4. 7 Pemilihan *New Device*

Lalu pilih nama templates yang telah kita isikan pada form sebelumnya di gambar 4.8.



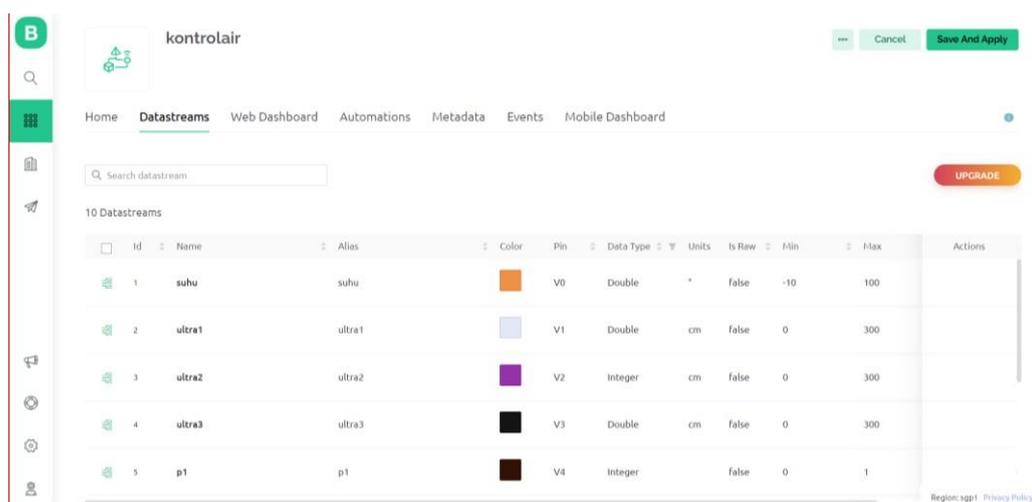
Gambar 4. 8 Tambahkan *New Device*

Setelah berhasil membuat *Project* baru kembali pada menu “*Templates*” untuk mendapatkan *template_id* dan *template_name* yang akan digunakan untuk proses pengkodean pada perangkat lunak Arduino IDE.



Gambar 4. 9 Mengatur *Template*

Selanjutnya peneliti mendefinisikan berbagai variabel yang ada pada sistem dengan membuat *Datastream*. Hal ini dibutuhkan untuk melakukan *monitoring* dan *controlling* kebun dengan aplikasi *Blynk*.



Gambar 4. 10 Membuat *Datastream*

Pada pembuatan *Datastream*, Peneliti memberikan nama pada masing-masing variabel, memberikan warna, mengkoneksikan ke pin virtual, mendefinisikan jenis data, satuan, dan nilai pembacaan minimal dan maksimalnya.

4.3.2 Pembangunan *Hardware*

Pada tahapan ini, peneliti melakukan pembangunan *hardware* pada *Prototyping*. Pembangunan *Hardware* dilakukan dengan merancang tata letak serta mencari alat-alat yang dibutuhkan, setelah itu peneliti melakukan perakitan. Peneliti tidak membuat/mendesain alat yang dibutuhkan, melainkan membeli dan memodifikasi alat-alat agar sesuai dengan penelitian ini.

4.3.2.1 Housing

Pada bagian ini, peneliti membuat *Housing* untuk menghubungkan semua komponen-komponen yang ada di sistem ini menuju ke *microcontroller* ESP32. Komponen yang ada di dalam *Housing* sendiri antara lain adalah *Microcontroller* ESP32, *Relay 4 Channel*, LCD, PCF8574 i2c, *PCB Board*, Resistor, dan *On/Off* Switch. Berikut merupakan *Housing* tersebut pada gambar 4.11 dan 4.12.



Gambar 4. 11 Tampak Tertutup *Housing* Modul Utama

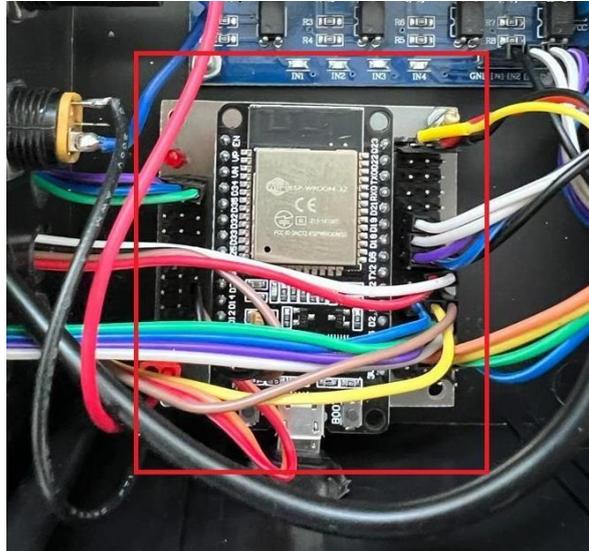


Gambar 4. 12 Tampak Terbuka *Housing* Modul Utama

4.3.2.2 *Microcontroller* ESP32

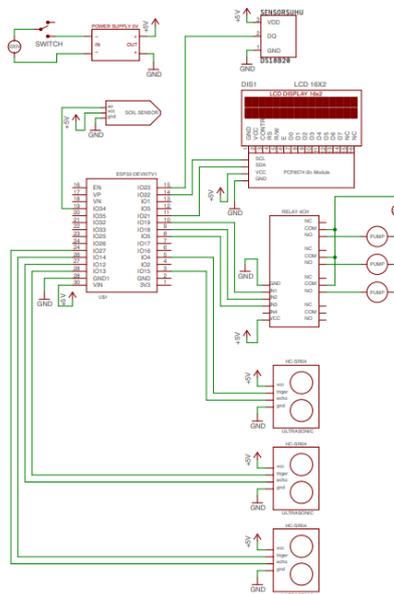
Microcontroller ESP32 merupakan komponen utama pada penelitian ini yang digunakan sebagai otak dari sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of*

Things (IoT). Berikut merupakan *microcontroller* ESP32 yang digunakan pada penelitian ini pada gambar 4.13.



Gambar 4.13 *Microcontroller* ESP32

Untuk menjalankan sistem *monitoring* dan *controlling* kebun organik, *Microcontroller* ESP32 dihubungkan ke beberapa komponen. Berikut merupakan rangkaiannya pada gambar 4. 14.



Gambar 4. 14 Rangkaian ESP32

Selanjutnya, berikut merupakan penjelasan mengenai rangkaian komponen yang terhubung langsung dengan *microcontroller* ESP32 pada tabel 4. 6.

Tabel 4. 6 Rangkaian ESP32

Komponen	Deskripsi
Sensor Kelembaban Tanah	Dihubungkan ke pin GPIO 34
Sensor Suhu Tanah DS18B20	Dihubungkan ke pin GPIO 23
Sensor <i>Ultrasonic</i> (Air)	Dihubungkan ke pin GPIO 4 (<i>Trig</i>) dan GPIO 15 (<i>Echo</i>)
Sensor <i>Ultrasonic</i> (Pupuk Cair)	Dihubungkan ke pin GPIO 13 (<i>Trig</i>) dan GPIO 12 (<i>Echo</i>)
Sensor <i>Ultrasonic</i> (Pestisida)	Dihubungkan ke pin GPIO 14 (<i>Trig</i>) dan GPIO 27 (<i>Echo</i>)
PCF8574 i2c	Dihubungkan ke pin GPIO 22 (SCL) dan GPIO 21 (SDA)
LCD	Terhubung melalui PCF8574 i2c
<i>Relay 4 Channel</i>	Dihubungkan ke pin GPIO 19 (IN1), GPIO 18 (IN2), dan GPIO 5 (IN3).
<i>Adapter 12V</i>	Terhubung melalui COM (<i>Common</i>) <i>Relay 4 Channel</i>
Pompa Air	Terhubung melalui NO (<i>Normally Open</i>) pada <i>Relay 4 Channel</i>
<i>Adapter 5V</i>	Dihubungkan ke VIN dan <i>On/Off Switch</i>
<i>On/Off Switch</i>	Dihubungkan ke VIN dan <i>Adapter 5V</i>

4.3.2.3 LCD

LCD digunakan untuk menampilkan berbagai informasi yang ada pada sistem. Berikut Berikut merupakan tampilan LCD pada gambar 4.15, 4.16, dan 4.17.



Gambar 4. 15 Tampilan pertama LCD saat sistem dinyalakan

Saat sistem pertama kali menyala LCD akan menampilkan nama *Wi-Fi* (SSID) dan *Password* yang terkoneksi.



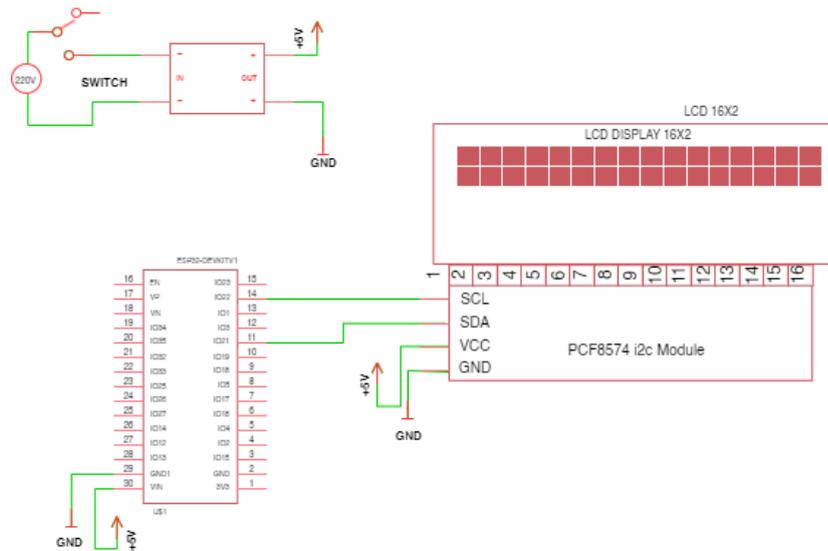
Gambar 4. 16 Tampilan LCD pertama

Setelah sistem terkoneksi dengan internet, selanjutnya LCD akan menampilkan informasi mengenai suhu tanah saat ini, dan ketinggian cairan (Air, Pupuk cair, dan Pestisida) pada masing-masing bak penampungannya yang ditampilkan dengan huruf A, B, dan C secara urut.



Gambar 4. 17 Tampilan LCD kedua

Setelah itu, LCD juga akan menampilkan waktu dan tanggal saat ini, kondisi pompa (*On/Off*) yang ditulis dalam bentuk angka 1 dan 0 (Jika 1 nyala dan jika 0 mati), dan kelembaban tanah. Selain itu, berikut merupakan rangkaian LCD di sistem ini pada gambar 4. 18.



Gambar 4. 18 Rangkaian LCD pada sistem

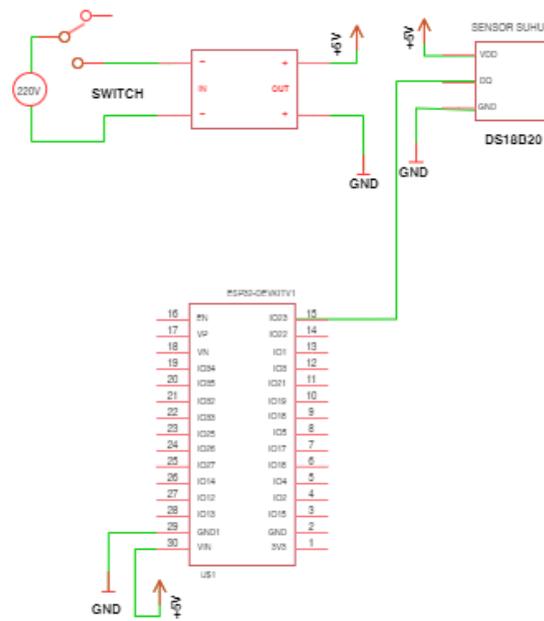
4.3.2.4 Sensor Suhu Tanah

Sensor suhu tanah pada sistem ini digunakan untuk mengetahui suhu tanah secara *real-time*. Berikut merupakan sensor suhu tanah pada gambar 4. 19.



Gambar 4. 19 Sensor suhu tanah

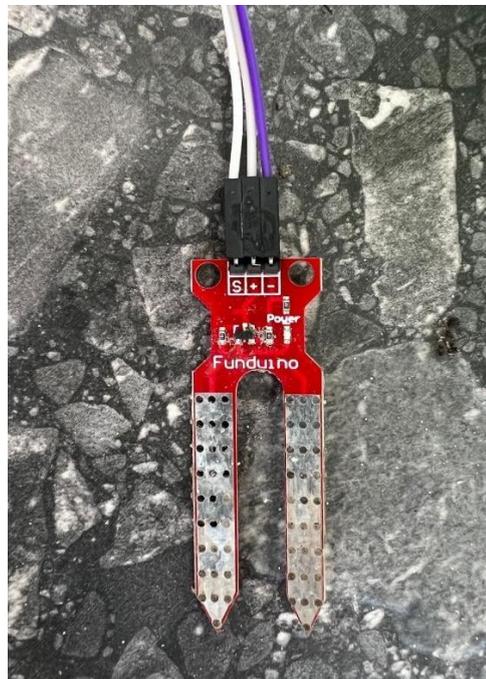
Selain itu, Berikut merupakan rangkaian sensor suhu tanah pada sistem *monitoring* dan *controlling* kebun organik.



Gambar 4. 20 Rangkaian sensor suhu tanah pada sistem

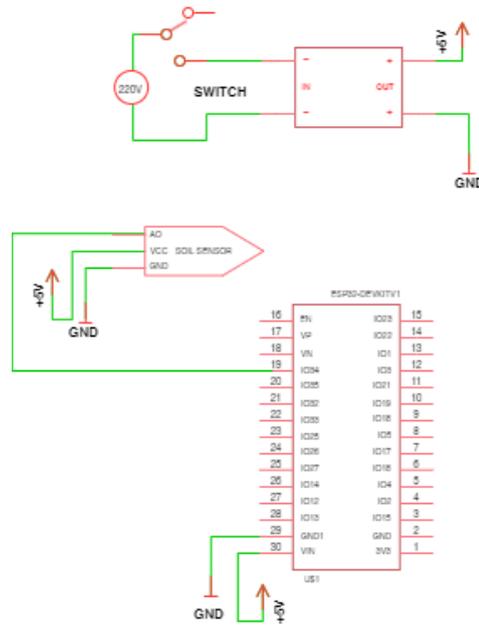
4.3.2.5 Sensor Kelembaban Tanah

Sensor kelembaban tanah pada sistem ini digunakan untuk mengetahui kelembaban tanah secara *real-time*. Berikut merupakan sensor kelembaban tanah pada gambar 4.21.



Gambar 4. 21 Sensor kelembaban tanah

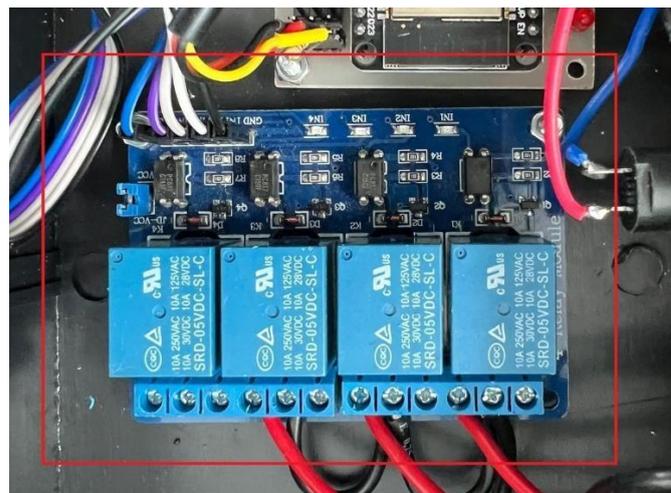
Selain itu, Berikut merupakan rangkaian sensor kelembaban tanah pada sistem *monitoring* dan *controlling* kebun organik.



Gambar 4. 22 Rangkaian sensor kelembaban tanah pada sistem

4.3.2.6 Relay 4 Channel dan Pompa Air

Relay 4 Channel pada penelitian ini digunakan untuk menghubungkan antara ESP32 dengan pompa air, dikarenakan ESP32 tidak mampu memberi daya kepada 3 pompa air, maka dibutuhkan sumber daya lainnya. Sedangkan pompa air pada penelitian ini digunakan untuk melakukan penyiraman air, pemupukan dan penyiraman pestisida. Berikut merupakan *Relay 4 Channel* dan Pompa Air pada gambar 4.23 dan 4.24.

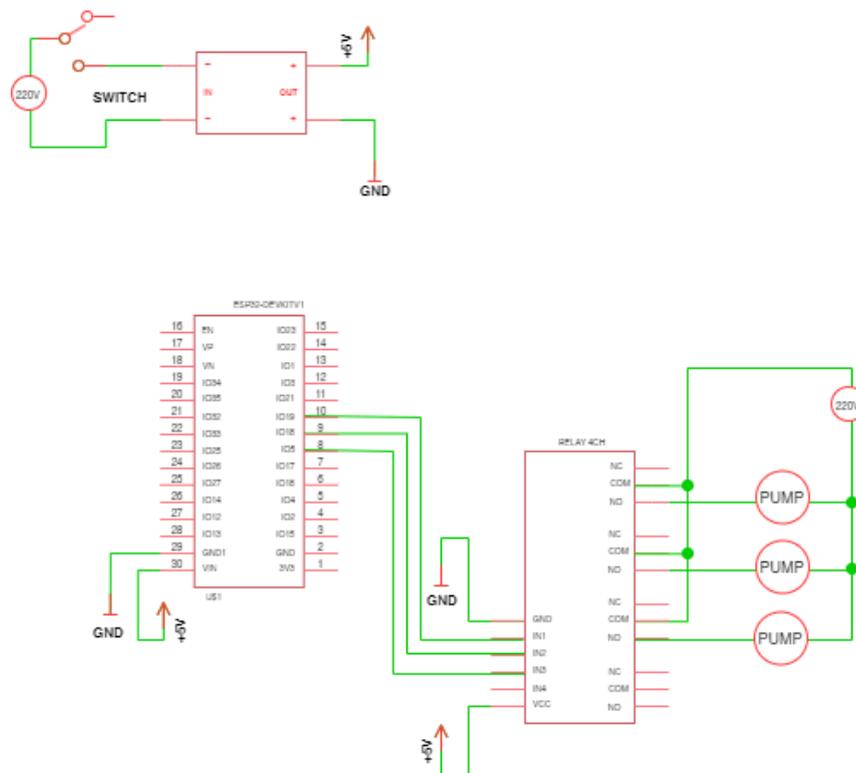


Gambar 4. 23 Relay 4 Channel



Gambar 4. 24 Pompa Air

Selain itu, Berikut merupakan rangkaian dari *Relay 4 Channel* dan pompa air pada sistem *monitoring* dan *controlling* kebun organik pada gambar 4. 24.



Gambar 4. 25 Rangkaian *Relay 4 Channel* dan Pompa air pada sistem

Dapat dilihat bahwa terdapat tambahan daya yang terhubung dengan *Relay 4 Channel* melalui *adapter 12V*. Berikut merupakan *adapter 12V* pada gambar 4.26.



Gambar 4. 26 Adapter 12V

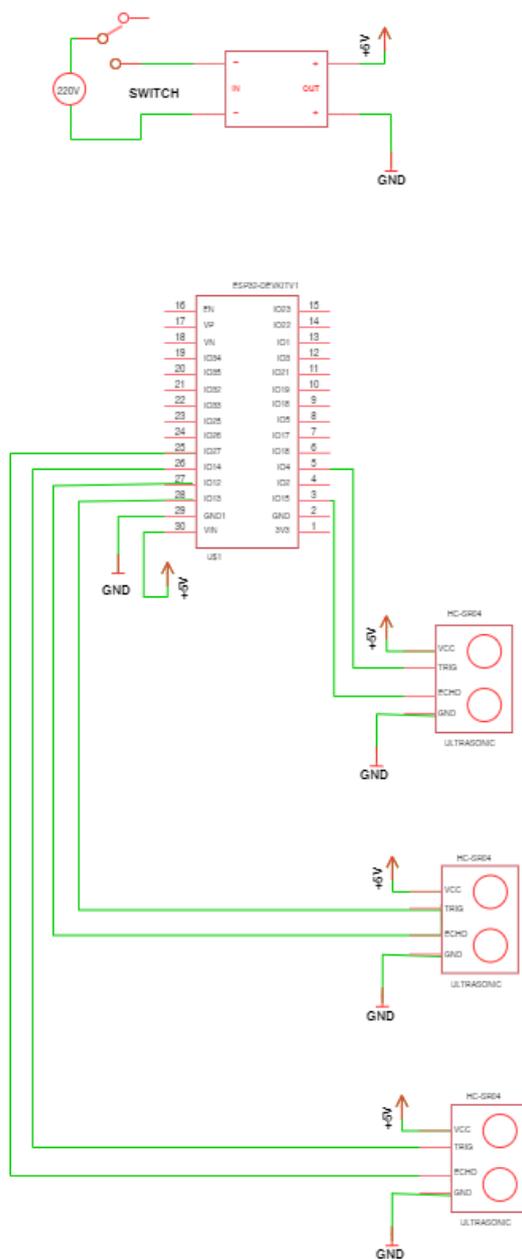
4.3.2.7 Sensor Ultrasonic

Sensor *Ultrasonic* pada sistem *monitoring* dan *controlling* digunakan untuk mengetahui level cairan (Air, Pupuk cair, dan Pestisida) pada bak penampungan. Jumlah sensor *Ultrasonic* yang digunakan adalah 3, masing-masing untuk bak penampungan yang berisi 3 cairan tersebut yang digunakan untuk melakukan proses perawatan tanaman. Berikut merupakan sensor *Ultrasonic* yang digunakan pada gambar 4. 27.



Gambar 4. 27 Sensor Ultrasonic

Selanjutnya, berikut merupakan rangkaian sensor *Ultrasonic* pada sistem *monitoring* dan *controlling* pada gambar 4. 28.



Gambar 4. 28 Rangkaian sensor *Ultrasonic* pada sistem *monitoring* dan *controlling* kebun organik

4.3.2.8 Bak Penampung

Bak Penampung digunakan untuk menampung air, pupuk cair, dan pestisida yang dibutuhkan oleh sistem. Bak ini memiliki volume 5.67L, namun dikarenakan dengan menggunakan sensor *Ultrasonic* bak tidak boleh diisi penuh (titik buta sensor

±5) sehingga jumlah volume cairan yang bisa diisi pada setiap bak adalah 4.25L. Berikut merupakan gambar dari bak penampungan tersebut:



Gambar 4. 29 Bak Penampung

4.3.2.9 Adapter 5V

Adapter 5V pada sistem ini digunakan sebagai penghubung antara ESP32 dan komponen-komponen pada sistem ke sumber daya. Berikut merupakan *Adapter 5V* pada gambar 4. 30.



Gambar 4. 30 Adapter 5V

4.3.3 Pembangunan Software

Pada tahap ini, peneliti melakukan pembangunan *Software* dengan melakukan pengkodean (*Coding*) pada aplikasi Arduino IDE dan *Blynk*. Setelah itu, disesuaikan kembali dengan *Hardware* yang dibangun.

4.3.3.1 Konfigurasi Awal

Pada tahap ini peneliti melakukan konfigurasi awal pada aplikasi Arduino IDE untuk mendefinisikan komponen *hardware* dan *library* yang digunakan pada sistem ini. Berikut merupakan *Pseudocode*-nya pada tabel 4.7.

Tabel 4. 7 *Pseudocode* Konfigurasi Awal

Pengkodean	Penjelasan
#define pump1 19	Mendefinisikan konstanta pump1 yang merupakan pompa 1 yang terhubung pada pin GPIO nomor 19.
#define pump2 18	Mendefinisikan konstanta pump2 yang merupakan pompa 2 yang terhubung pada pin GPIO nomor 18.
#define pump3 5	Mendefinisikan konstanta pump3 yang merupakan pompa 3 yang terhubung pada pin GPIO nomor 5.
#define pinled 2	Mendefinisikan konstanta pinled yang merupakan LED yang terhubung pada pin GPIO nomor 2.
#define soil 34	Mendefinisikan konstanta soil yang merupakan sensor kelembaban tanah yang terhubung pada pin GPIO nomor 34.
#define pinsuhu 23	Mendefinisikan konstanta pinsuhu yang merupakan sensor suhu tanah yang terhubung pada pin GPIO nomor 23.
LiquidCrystal_I2C lcd(0x27, 16, 2);	Mendefinisikan LCD yang digunakan pada penelitian ini, yang memiliki ukuran 16 × 2.
#include <OneWire.h> #include <DallasTemperature.h>	Memanggil <i>library</i> <i>OneWire</i> dan <i>DallasTemperature</i> yang akan digunakan untuk sensor suhu tanah.
OneWire configsensor(pinsuhu); DallasTemperature	Membuat objek configsensor dengan menggunakan konstruktor <i>OneWire</i> dan menyediakan pin data yang telah didefinisikan sebelumnya.
sensor(&configsensor);	Membuat objek sensor dengan menggunakan konstruktor <i>DallasTemperature</i> dan menginisialisasinya dengan objek configsensor. Ini dilakukan agar objek sensor dapat berkomunikasi dengan sensor suhu berbasis <i>OneWire</i> yang terhubung ke pin data.
#include <BlynkSimpleEsp32.h> // library <i>Blynk</i>	Memanggil <i>library</i> <i>Blynk</i> .
#define BLYNK_TEMPLATE_ID "TMPL6OYcash5y"	Mendefinisikan ID template <i>Blynk</i> .

Pengkodean	Penjelasan
#define <i>BLYNK_TEMPLATE_NAME</i> "kontrolair"	Mendefinisikan nama template <i>Blynk</i> .
#define <i>BLYNK_AUTH_TOKEN</i> "vXRcQGCOIBvixS23m5oGGyaBrlWsb5V2"	Mendefinisikan token otorisasi <i>Blynk</i> yang diperlukan untuk menghubungkan perangkat ke server <i>Blynk</i> .
#define <i>BLYNK_PRINT</i> Serial <i>BlynkTimer</i> timers;	Mengarahkan output <i>Blynk</i> ke Serial Monitor untuk tampilan informasi <i>Blynk</i> . Membuat objek <i>timers</i> dari <i>BlynkTimer</i> yang digunakan untuk menjalankan tugas-tugas secara berkala.
char auth[] = <i>BLYNK_AUTH_TOKEN</i> ;	Mendefinisikan variabel auth sebagai array karakter (char array) yang akan menyimpan token otentikasi <i>Blynk</i> .
char ssid[] = "Nama <i>Wifi</i> yang digunakan";	Mendefinisikan variabel ssid sebagai nama jaringan <i>Wi-Fi</i> yang akan digunakan.
char pass[] = " <i>Password Wifi</i> ";	Mendefinisikan variabel pass sebagai kata sandi jaringan <i>Wi-Fi</i> yang akan digunakan.
#include <NTPClient.h>	Memanggil <i>Library</i> NTP yang digunakan untuk melakukan penjadwalan waktu melalui <i>software</i> .
const long utcOffsetInSeconds = 25200;	Mengatur offset waktu ke GMT+7 (Waktu Indonesia Barat).
WiFiUDP ntpUDP;	Membuat objek ntpUDP dari WiFiUDP yang digunakan untuk koneksi UDP dengan server NTP.
NTPClient timeClient(ntpUDP, "id.pool.ntp.org", utcOffsetInSeconds);	Membuat objek timeClient dari NTPClient yang digunakan untuk mendapatkan waktu dari server NTP.

4.3.3.2 Mendefinisikan Variabel

Pada tahap ini, peneliti mendefinisikan variabel yang digunakan pada sistem ini, berikut merupakan *Pseudocode*-nya pada tabel 4.8.

Tabel 4. 8 *Pseudocode* mendefinisikan variabel

Pengkodean	Penjelasan
int monthDay;	Variabel untuk menyimpan hari bulan saat ini.
int currentMonth;	Variabel untuk menyimpan bulan saat ini.
int currentYear;	Variabel untuk menyimpan tahun saat ini.
String tanggal,waktu;	Variabel string untuk menyimpan tanggal dan waktu saat ini.
int jam;	Variabel untuk menyimpan jam saat ini.
int count1=0,count2=0;	Variabel untuk pengaturan tampilan LCD.

Pengkodean	Penjelasan
float ultra1,ultra2,ultra3;	Variabel untuk menyimpan nilai jarak yang diukur oleh sensor <i>Ultrasonic</i> .
float tinggi1,tinggi2,tinggi3;	Variabel untuk menyimpan nilai tinggi air di dalam tangki.
int tpump1=0,tpump2=0,tpump3=0;	Variabel untuk menghitung waktu pompa aktif.
float suhu;	Variabel untuk menyimpan nilai suhu yang diukur oleh sensor suhu.
int kelembaban;	Variabel untuk menyimpan nilai kelembaban tanah yang diukur.
int lock1,lock2,lock3;	Variabel pengunci untuk menghindari pengiriman notifikasi berulang kali.
int lock4,lock5;	Variabel pengunci untuk menghindari pengiriman notifikasi berulang kali.
int lock6,lock7,lock8;	Variabel pengunci untuk menghindari pengiriman notifikasi berulang kali.
int pin_trig[]={4,13,14,};	Array yang berisi pin trigger untuk sensor <i>Ultrasonic</i> .
int pin_echo[]={15,12,27,};	Array yang berisi pin echo untuk sensor <i>Ultrasonic</i> .

4.3.3.3 Pengaturan Sistem

Pada tahap ini, peneliti melakukan pengaturan sistem dengan menentukan variabel-variabel pada sistem, seperti pengaturan batas suhu, waktu dan tanggal penyiraman, dll. Berikut merupakan *Pseudocode* pengaturan sistem pada tabel 4.9.

Tabel 4. 9 *Pseudocode* pengaturan sistem

Pengkodean	Penjelasan
float setultra1 = 20.0;	Nilai tinggi air yang diatur untuk tangki 1.
float setultra2 = 20.0;	Nilai tinggi air yang diatur untuk tangki 2.
float setultra3 = 20.0;	Nilai tinggi air yang diatur untuk tangki 3.
float setsuhu = 31.0;	Nilai suhu yang diatur.
int setkelembaban = 50;	Nilai kelembaban tanah yang diatur.
int air1 = 6;	Jam penyiraman air pagi.
int air2 = 18;	Jam penyiraman air sore.
int jampupuk = 7;	Jam penyiraman pupuk cair
Int jamhama = 8;	Jam penyiraman pestisida
int pupukcair1 = 1;	Tanggal pemberian pupuk cair minggu ke-1.
int pupukcair2 = 15;	Tanggal pemberian pupuk cair minggu ke-3.
int hama1 = 1;	Tanggal penyemprotan pestisida minggu ke-1.
int hama2 = 8;	Tanggal penyemprotan pestisida minggu ke-2.
int hama3 = 15;	Tanggal penyemprotan pestisida minggu ke-3.
int hama4 = 22;	Tanggal penyemprotan pestisida minggu ke-4.
int waktupump1 = 60;	Durasi penyiraman pompa 1 dalam detik.
int waktupump2 = 60;	Durasi penyiraman pompa 2 dalam detik.

Pengkodean	Penjelasan
<pre>int waktupump3 = 60; int habis1 = 5; int habis2 = 5; int habis3 = 5;</pre>	<p>Durasi penyiraman pompa 3 dalam detik. Batas tinggi air pada tangki 1 yang menunjukkan bahwa air telah habis. Batas tinggi air pada tangki 2 yang menunjukkan bahwa air telah habis. Batas tinggi air pada tangki 3 yang menunjukkan bahwa air telah habis.</p>

4.3.3.4 Aplikasi Blynk

Aplikasi *Blynk* pada penelitian ini digunakan sebagai alat untuk melakukan *monitoring* dan *controlling prototype* yang sudah dibangun. Berikut merupakan *Pseudocode*-nya pada tabel 4. 10.

Tabel 4. 10 *Pseudocode* aplikasi *Blynk*

Pengkodean	Penjelasan
<pre>BLYNK_WRITE(V4){ int pinValue = param.asInt(); if(pinValue==1) digitalWrite(pump1, 0); if(pinValue==0) digitalWrite(pump1, 1); Serial.print("pump1:"); Serial.println(pinValue); } BLYNK_WRITE(V5){ int pinValue = param.asInt(); if(pinValue==1) digitalWrite(pump2, 0); if(pinValue==0) digitalWrite(pump2, 1); Serial.print("pump2:"); Serial.println(pinValue); } BLYNK_WRITE(V6){ int pinValue = param.asInt(); if(pinValue==1) digitalWrite(pump3, 0); //on if(pinValue==0) digitalWrite(pump3, 1); //off Serial.print("pump3:"); Serial.println(pinValue); }</pre>	<p>Fungsi yang dieksekusi ketika widget <i>Blynk</i> dengan pin virtual V4 diubah nilainya. Berfungsi untuk mengontrol pompa 1 berdasarkan input dari <i>Blynk</i>.</p> <p>Fungsi yang dieksekusi ketika widget <i>Blynk</i> dengan pin virtual V5 diubah nilainya. Berfungsi untuk mengontrol pompa 2 berdasarkan input dari <i>Blynk</i>.</p> <p>Fungsi yang dieksekusi ketika widget <i>Blynk</i> dengan pin virtual V6 diubah nilainya. Berfungsi untuk mengontrol pompa 3 berdasarkan input dari <i>Blynk</i>.</p>

4.3.3.5 Pengaturan Waktu

Penjadwalan waktu pada penelitian ini digunakan untuk menentukan waktu penyiraman air, pupuk, dan pestisida. Penjadwalan waktu hanya dilakukan melalui *software* dan tidak menggunakan *hardware*. Berikut merupakan *Pseudocode* penjadwalan waktu pada tabel 4. 11.

Tabel 4. 11 *Pseudocode* penjadwalan waktu

Pengkodean	Penjelasan
void get_time(){...}	Fungsi untuk mendapatkan waktu saat ini dari server NTP dan menyimpannya dalam variabel tanggal, waktu, dan jam.
timeClient.update();	Memperbarui waktu pada objek timeClient.
unsigned long epochTime =	Mendapatkan waktu dalam bentuk epoch (jumlah detik sejak 1 Januari 1970)
timeClient.getEpochTime();	menggunakan timeClient.getEpochTime() dan menyimpannya dalam variabel epochTime.
struct tm *ptm = gmtime ((time_t	Mengonversi epoch time menjadi struktur waktu dengan gmtime() dan menyimpannya dalam pointer ptm.
*)&epochTime);	
monthDay = ptm->tm_mday;	Mendapatkan tanggal bulan saat ini dari struktur waktu dengan ptm->tm_mday dan menyimpannya dalam variabel monthDay.
currentMonth = ptm->tm_mon+1;	Mendapatkan bulan saat ini dari struktur waktu dengan ptm->tm_mon+1 (penambahan 1 karena indeks bulan dimulai dari 0) dan menyimpannya dalam variabel currentMonth.
currentYear = ptm->tm_year+1900;	Mendapatkan tahun saat ini dari struktur waktu dengan ptm->tm_year+1900 (penambahan 1900 karena tm_year menyimpan tahun sejak 1900) dan menyimpannya dalam variabel currentYear.
tanggal = String(monthDay);	Mengubah nilai monthDay menjadi string dan menyimpannya dalam variabel tanggal.
tanggal += "-";	Menggabungkan tanda "-" ke string tanggal untuk memisahkan tanggal dan bulan.
tanggal += String(currentMonth);	Menggabungkan nilai bulan (sebagai string) ke dalam string tanggal.
tanggal += "-";	Menggabungkan tanda "-" ke string tanggal untuk memisahkan bulan dan tahun.
tanggal += String(currentYear);	Menggabungkan nilai tahun (sebagai string) ke dalam string tanggal.
jam = timeClient.getHours();	Mendapatkan jam saat ini dari timeClient.getHours() dan menyimpannya dalam variabel jam.
waktu = timeClient.getFormattedTime();	Mendapatkan waktu saat ini dalam format yang diformat dengan timeClient.getFormattedTime() dan menyimpannya dalam variabel waktu.
Serial.println(tanggal);	Mencetak tanggal menggunakan Serial.println() untuk output di monitor serial.
Serial.println(waktu);	Mencetak waktu menggunakan Serial.println() untuk output di monitor serial.
Serial.println(jam);	Mencetak jam menggunakan Serial.println() untuk output di monitor serial.

4.3.3.6 Pengaturan LCD

LCD pada penelitian ini digunakan untuk menampilkan berbagai data yang ada pada sistem, Jenis LCD yang digunakan adalah *Liquid Crystal Display* dengan ukuran 16×2 . Berikut merupakan *Pseudocode*-nya pada tabel 4. 12.

Tabel 4. 12 *Pseudocode* LCD

Pengkodean	Penjelasan
void display_lcd() count1++; if (count1 >=5){count1=0;count2++;}	Mendefinisikan fungsi display_lcd(). Menaikkan nilai variabel count1 sebanyak 1. Jika nilai variabel count1 mencapai 5 atau lebih, maka count1 diatur kembali menjadi 0 dan count2 ditambah 1.
if(count2>1)count2=0;	Jika nilai variabel count2 lebih dari 1, maka count2 diatur kembali menjadi 0.
lcd.clear();	Menghapus semua konten yang ditampilkan di LCD.
if(count2==0){ ... }	Jika nilai variabel count2 adalah 0, maka jalankan blok kode berikutnya.
lcd.setCursor(0,0);	Mengatur posisi kursor LCD ke baris 0, kolom 0.
lcd.print(waktu);	Menampilkan nilai dari variabel waktu pada LCD.
lcd.setCursor(11,0);	Mengatur posisi kursor LCD ke baris 0, kolom 11.
lcd.print(digitalRead(pump1));	Menampilkan nilai bacaan digital dari pin pump1 pada LCD.
lcd.print(" ");	Menampilkan spasi pada LCD.
lcd.print(digitalRead(pump2));	Menampilkan nilai bacaan digital dari pin pump2 pada LCD.
lcd.print(" ");	Menampilkan spasi pada LCD.
lcd.print(digitalRead(pump3));	Menampilkan nilai bacaan digital dari pin pump3 pada LCD.
lcd.setCursor(0,1);	Mengatur posisi kursor LCD ke baris 1, kolom 0.
lcd.print(tanggal);	Menampilkan nilai dari variabel tanggal pada LCD.
lcd.setCursor(11,1);	Mengatur posisi kursor LCD ke baris 1, kolom 11.
lcd.print("K:");	Menampilkan teks "K:" pada LCD.
lcd.print(kelembaban);	Menampilkan nilai dari variabel kelembaban pada LCD.
if(count2==1){ ... }	Jika nilai variabel count2 adalah 1, maka jalankan blok kode berikutnya.
lcd.setCursor(0,0);	Mengatur posisi kursor LCD ke baris 0, kolom 0.
lcd.print("S:");	Menampilkan teks "S:" pada LCD.

Pengkodean	Penjelasan
<code>lcd.print(suhu,1);</code>	Menampilkan nilai dari variabel suhu dengan 1 digit desimal pada LCD.
<code>lcd.setCursor(9,0);</code>	Mengatur posisi kursor LCD ke baris 0, kolom 9.
<code>lcd.print("A:");</code>	Menampilkan teks "A:" pada LCD.
<code>lcd.print(tinggi1,1);</code>	Menampilkan nilai dari variabel tinggi1 dengan 1 digit desimal pada LCD.
<code>lcd.setCursor(0,1);</code>	Mengatur posisi kursor LCD ke baris 1, kolom 0.
<code>lcd.print("B:");</code>	Menampilkan teks "B:" pada LCD.
<code>lcd.print(tinggi2,1);</code>	Menampilkan nilai dari variabel tinggi2 dengan 1 digit desimal pada LCD.
<code>lcd.setCursor(9,1);</code>	Mengatur posisi kursor LCD ke baris 1, kolom 9.
<code>lcd.print("C:");</code>	Menampilkan teks "C:" pada LCD.
<code>lcd.print(tinggi3,1);</code>	Menampilkan nilai dari variabel tinggi3 dengan 1 digit desimal pada LCD.
<code>lcd.begin();</code>	Menginisialisasi objek LCD.
<code>lcd.backlight();</code>	Menghidupkan backlight LCD.
<code>lcd.clear();</code>	Menghapus semua konten yang ditampilkan di LCD.
<code>lcd.setCursor(0,0);</code>	Mengatur posisi kursor LCD ke baris 0, kolom 0.
<code>lcd.print("ssid:");</code>	Menampilkan teks "ssid:" pada LCD.
<code>lcd.print(ssid);</code>	Menampilkan nilai dari variabel ssid pada LCD.
<code>lcd.setCursor(0,1);</code>	Mengatur posisi kursor LCD ke baris 1, kolom 0.
<code>lcd.print("pass:");</code>	Menampilkan teks "pass:" pada LCD.
<code>lcd.print(pass);</code>	Menampilkan nilai dari variabel pass pada LCD.

4.3.3.7 Pengaturan Sensor Suhu Tanah

Sensor suhu tanah pada penelitian ini digunakan untuk mengetahui suhu tanah, yang merupakan salah satu variabel pada automasi sistem ini. Berikut merupakan *Pseudocode*-nya pada tabel 4. 13.

Tabel 4. 13 *Pseudocode* Sensor suhu tanah

Pengkodean	Penjelasan
<code>sensor.requestTemperatures();</code>	Meminta sensor suhu untuk membaca nilai suhu saat ini dengan memanggil fungsi <code>requestTemperatures()</code> pada objek sensor.
<code>suhu = sensor.getTempCByIndex(0);</code>	Membaca suhu saat ini dari sensor dan menyimpannya di variabel suhu.

4.3.3.8 Pengaturan Sensor Kelembaban Tanah

Sensor kelembaban tanah pada penelitian ini digunakan untuk mengetahui kelembaban tanah, yang merupakan salah satu variabel pada automasi sistem ini. Berikut merupakan *pseudocode*-nya pada tabel 4. 14.

Tabel 4. 14 *Pseudocode* sensor kelembaban tanah

Pengkodean	Penjelasan
kelembaban = map(analogRead(soil),0,4096,0,100);	Membaca nilai dari sensor kelembaban tanah yang terhubung ke pin soil, dan memetakan nilainya ke dalam rentang 0-100 persen.

4.3.3.9 Pengaturan Sensor *Ultrasonic*

Sensor *Ultrasonic* pada penelitian ini digunakan untuk mengetahui ketinggian/level air, pupuk organik, dan pestisida organik didalam bak. Berikut merupakan *pseudocode*-nya pada tabel 4. 15.

Tabel 4. 15 *Pseudocode* sensor *Ultrasonic*

Pengkodean	Penjelasan
float read_srf(int i)	Fungsi untuk membaca jarak dari sensor <i>Ultrasonic</i> dengan parameter i sebagai indeks pin.
unsigned int duration;	Mendeklarasikan variabel duration sebagai tipe data unsigned int untuk menyimpan durasi pulsa <i>Ultrasonic</i> yang akan diukur.
float distance;	Mendeklarasikan variabel distance sebagai tipe data float untuk menyimpan jarak yang akan dihitung.
digitalWrite(pin_trig[i], LOW);	Mengirim sinyal LOW ke pin trigger (pin_trig) yang sesuai dengan nilai i. Ini mematikan trigger sinyal <i>Ultrasonic</i> .
delayMicroseconds(5);	Menunda eksekusi selama 5 mikrodetik (us) untuk memberikan jeda sebelum mengaktifkan kembali trigger sinyal <i>Ultrasonic</i> .
digitalWrite(pin_trig[i], HIGH);	Mengirim sinyal HIGH ke pin trigger (pin_trig) yang sesuai dengan nilai i. Ini mengaktifkan kembali trigger sinyal <i>Ultrasonic</i> .
delayMicroseconds(10);	Menunda eksekusi selama 10 mikrodetik (us) untuk memberikan jeda sebelum mematikan kembali trigger sinyal <i>Ultrasonic</i> .
digitalWrite(pin_trig[i], LOW);	Mengirim sinyal LOW ke pin trigger (pin_trig) yang sesuai dengan nilai i. Ini mematikan kembali trigger sinyal <i>Ultrasonic</i> .
duration = pulseIn(pin_echo[i], HIGH);	Mengukur durasi pulsa yang diterima pada pin echo (pin_echo) yang sesuai dengan nilai i dan menyimpannya ke dalam variabel duration.

Pengkodean	Penjelasan
distance = duration * 0.034 / 2;	Durasi pulsa ini akan digunakan untuk menghitung jarak.
return distance;	Menghitung jarak berdasarkan durasi pulsa yang diperoleh. Menggunakan faktor konversi 0,034 cm/us (kecepatan suara dalam udara) dan membagi hasilnya dengan 2 karena perjalanan bolak-balik sinyal <i>Ultrasonic</i> . Mengembalikan nilai jarak yang telah dihitung dari fungsi sebagai hasil pengukuran.

4.3.3.10 Fungsi utama

Pada tahapan ini peneliti melakukan pengkodean untuk fungsi-fungsi utama pada sistem ini, seperti penyiraman air, pupuk, dan pestisida yang dijadwal, penyiraman air ketika suhu tinggi dan kelembaban rendah, dll. Berikut merupakan *pseudocodenya* pada tabel 4.16.

Tabel 4.16 *Pseudocode* fungsi utama

Pengkodean	Penjelasan
void tasktimer(){...}	merupakan fungsi utama yang akan dieksekusi secara berulang oleh <i>BlynkTimer</i> .
digitalWrite(pinled, digitalRead(pinled)^1);	Memflip status LED dengan menggunakan operasi bitwise XOR (XOR) pada nilai yang saat ini dibaca dari pin pinled.
get_time();	Memanggil fungsi <code>get_time()</code> untuk mendapatkan waktu dan tanggal saat ini menggunakan objek <code>timeClient</code> .
kelembaban = map(analogRead(soil), 0, 4096, 0, 100);	Membaca nilai analog dari sensor kelembaban tanah yang terhubung ke pin soil, dan memetakan nilainya ke dalam rentang 0-100 menggunakan fungsi <code>map()</code> .
ultra1 = read_srf(0);	Membaca jarak yang terukur dari sensor <i>Ultrasonic</i> pada bak 1 dengan memanggil fungsi <code>read_srf(0)</code> .
ultra2 = read_srf(1);	Membaca jarak yang terukur dari sensor <i>Ultrasonic</i> pada bak 2 dengan memanggil fungsi <code>read_srf(1)</code> .
ultra3 = read_srf(2);	Membaca jarak yang terukur dari sensor <i>Ultrasonic</i> pada bak 3 dengan memanggil fungsi <code>read_srf(2)</code> .
tinggi1 = setultra1 - ultra1;	Menghitung tinggi air pada bak 1 dengan mengurangi jarak yang terukur dari tinggi yang telah ditetapkan (<code>setultra1</code>).
tinggi2 = setultra2 - ultra2;	Menghitung tinggi air pada bak 2 dengan mengurangi jarak yang terukur dari tinggi yang telah ditetapkan (<code>setultra2</code>).

Pengkodean	Penjelasan
tinggi3 = setultra3 - ultra3;	Menghitung tinggi air pada bak 3 dengan mengurangi jarak yang terukur dari tinggi yang telah ditetapkan (setultra3).
if (tinggi1 < 0) tinggi1 = 0;	Memastikan bahwa tinggi air pada bak 1 tidak menjadi negatif dengan mengaturnya menjadi 0 jika kurang dari 0.
if (tinggi2 < 0) tinggi2 = 0;	Memastikan bahwa tinggi air pada bak 2 tidak menjadi negatif dengan mengaturnya menjadi 0 jika kurang dari 0.
if (tinggi3 < 0) tinggi3 = 0;	Memastikan bahwa tinggi air pada bak 3 tidak menjadi negatif dengan mengaturnya menjadi 0 jika kurang dari 0.
sensor.requestTemperatures();	Meminta sensor suhu untuk membaca nilai suhu saat ini dengan memanggil fungsi requestTemperatures() pada objek sensor.
suhu = sensor.getTempCByIndex(0);	Membaca suhu saat ini dari sensor dan menyimpannya di variabel suhu.
<pre> if(suhu > setsuhu){ if(lock1!=1){ lock1=1; digitalWrite(pump1, 0); Blynk.logEvent("notif","Suhu melebihi batas, sistem melakukan penyiraman otomatis"); } } </pre>	Memeriksa apakah suhu saat ini melebihi suhu yang telah ditetapkan (setsuhu). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i> dan melakukan penyiraman air.
<pre> if(suhu < setsuhu){ if(lock1!=0){ lock1=0; digitalWrite(pump1, 1); } } </pre>	Memeriksa apakah suhu saat ini kurang dari suhu yang telah ditetapkan (setsuhu). Jika ya, maka sistem mematikan pompa (memberhentikan penyiraman).
<pre> if(kelembaban < setkelembaban){ if(lock2!=1){ lock2=1; digitalWrite(pump1, 0); Blynk.logEvent("notif","Kelembaban kurang dari batas, sistem melakukan penyiraman otomatis"); } } </pre>	Memeriksa apakah kelembaban tanah saat ini kurang dari kelembaban yang telah ditetapkan (setkelembaban). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i> dan melakukan penyiraman air.
<pre> if(kelembaban > setkelembaban){ if(lock2!=0){ lock2=0; digitalWrite(pump1, 1); } } </pre>	Memeriksa apakah kelembaban tanah saat ini melebihi kelembaban yang telah ditetapkan (setkelembaban). Jika ya, maka sistem mematikan pompa (memberhentikan penyiraman).

Pengkodean	Penjelasan
<pre> if(jam == air1){ if(lock3!=1){ lock3=1; tpump1=0; digitalWrite(pump1, 0); Blynk.logEvent("notif", "Penyiraman otomatis pagi"); } if(tpump1>=waktupump1) { if(lock3!=2){ lock3=2; digitalWrite(pump1, 1); } } else {tpump1++;} if(jam == air2){ if(lock3!=3){ lock3=3; tpump1=0; digitalWrite(pump1, 0); Blynk.logEvent("notif", "Penyiraman otomatis sore"); } if(tpump1>=waktupump1) { if(lock3!=4){ lock3=4; digitalWrite(pump1, 1); } } else {tpump1++;} } </pre>	<p>Memeriksa apakah jam saat ini sama dengan waktu penyiraman air pagi (air1). Jika ya, maka sistem melakukan penyiraman selama waktu yang sudah ditentukan dan mengirimkan notif ke aplikasi <i>Blynk</i>.</p> <p>Memeriksa apakah jam saat ini sama dengan waktu penyiraman air sore (air2). Jika ya, maka sistem melakukan penyiraman selama waktu yang sudah ditentukan dan mengirimkan notif ke aplikasi <i>Blynk</i>.</p>
<pre> if(monthDay == pupukcair1 && jam==jampupuk){ if(lock4!=1){ lock4=1; tpump2=0; digitalWrite(pump2, 0); Blynk.logEvent("notif", "Penyiraman pupuk cair minggu ke 1"); } if(tpump2>=waktupump2) { if(lock4!=2){ lock4=2; </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyiraman pupuk cair minggu ke 1 dan waktu penyiramannya. Jika ya, maka sistem melakukan penyiraman pupuk cair selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>

Pengkodean	Penjelasan
<pre>digitalWrite(pump2, 1); } } else {tpump2++;} } if(monthDay == pupukcair2 && jam==jampupuk){ if(lock4!=3){ lock4=3; tpump2=0; digitalWrite(pump2, 0); Blynk.logEvent("notif", "Penyiraman pupuk cair minggu ke 3"); } if(tpump2>=waktupump2) { if(lock4!=4){ lock4=4; digitalWrite(pump2, 1); } } else {tpump2++;} } if(monthDay == hama1 && jam==jamhama){ if(lock5!=1){ lock5=1; tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif", "Penyemprotan pestisida minggu ke 1"); } if(tpump3>=waktupump3) { if(lock5!=2){ lock5=2; digitalWrite(pump3, 1); } } else {tpump3++;} } if(monthDay == hama2 && jam==jamhama){ if(lock5!=3){</pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyiraman pupuk cair minggu ke 3 dan waktu penyiramannya. Jika ya, maka sistem melakukan penyiraman pupuk cair selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> } } else {tpump3++;} } if(monthDay == hama2 && jam==jamhama){ if(lock5!=3){</pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 1 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> } } else {tpump3++;} } if(monthDay == hama2 && jam==jamhama){ if(lock5!=3){</pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 2 dan waktu penyiraman yang</p>

Pengkodean	Penjelasan
<pre>lock5=3; tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif", "Penyemprotan pestisida minggu ke 2"); }</pre>	<p>sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre>if(tpump3>=waktupump3) { if(lock5!=4){ lock5=4; digitalWrite(pump3, 1); } } else {tpump3++;}</pre>	
<pre>if(monthDay == hama3 && jam==jamhama){ if(lock5!=5){ lock5=5; tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif", "Penyemprotan pestisida minggu ke 3"); }</pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 3 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre>if(tpump3>=waktupump3) { if(lock5!=6){ lock5=6; digitalWrite(pump3, 1); } } else {tpump3++;}</pre>	
<pre>if(monthDay == hama4 && jam==jamhama){ if(lock5!=7){ lock5=7; tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif", "Penyemprotan pestisida minggu ke 4"); }</pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 4 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>

Pengkodean	Penjelasan
<pre> if(tpump3>=waktupump3) { if(lock5!=8){ lock5=8; digitalWrite(pump3, 1); } } else {tpump3++;} } </pre>	
<pre> if(tinggi1 < habis1){ if(lock6!=1){ lock6=1; Blynk.logEvent("notif","Air habis, isi bak segera!"); } } </pre>	<p>Memeriksa apakah tinggi air pada bak 1 lebih rendah dari batas habis (habis1). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tinggi1 > habis1){ lock6=0; } if(tinggi2 < habis2){ if(lock7!=1){ lock7=1; Blynk.logEvent("notif","Pupuk habis, isi bak segera!"); } } </pre>	<p>Memeriksa apakah tinggi air pada bak 1 lebih tinggi dari batas habis (habis1). Jika ya, maka sistem memberhentikan notifikasi. Memeriksa apakah tinggi air pada bak 2 lebih rendah dari batas habis (habis2). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tinggi2 > habis2){ lock7=0; } if(tinggi3 < habis3){ if(lock8!=1){ lock8=1; Blynk.logEvent("notif","Pestisida habis, isi bak segera!"); } } </pre>	<p>Memeriksa apakah tinggi air pada bak 2 lebih tinggi dari batas habis (habis2). Jika ya, maka sistem memberhentikan notifikasi. Memeriksa apakah tinggi air pada bak 3 lebih rendah dari batas habis (habis3). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tinggi3 > habis3){ lock8=0; } Blynk.virtualWrite(V0, suhu); Blynk.virtualWrite(V1, tinggi1); Blynk.virtualWrite(V2, tinggi2); Blynk.virtualWrite(V3, tinggi3); </pre>	<p>Memeriksa apakah tinggi air pada bak 3 lebih tinggi dari batas habis (habis3). Jika ya, maka sistem memberhentikan notifikasi. Mengirim nilai suhu ke pin virtual V0 pada aplikasi <i>Blynk</i>. Mengirim nilai tinggi air pada bak 1 ke pin virtual V1 pada aplikasi <i>Blynk</i>. Mengirim nilai tinggi air pada bak 2 ke pin virtual V2 pada aplikasi <i>Blynk</i>. Mengirim nilai tinggi air pada bak 3 ke pin virtual V3 pada aplikasi <i>Blynk</i>.</p>

Pengkodean	Penjelasan
<i>Blynk.virtualWrite</i> (V7, kelembaban);	Mengirim nilai kelembaban tanah ke pin virtual V7 pada aplikasi <i>Blynk</i> .
Serial.print(" S: ");	Mencetak teks " S: " di Serial Monitor.
Serial.print(suhu);	Mencetak nilai suhu saat ini di Serial Monitor.
Serial.print(" A: ");	Mencetak teks " A: " di Serial Monitor.
Serial.print(ultra1);	Mencetak nilai jarak dari sensor <i>Ultrasonic</i> pada bak 1 di Serial Monitor.
Serial.print(" B: ");	Mencetak teks " B: " di Serial Monitor.
Serial.print(ultra2);	Mencetak nilai jarak dari sensor <i>Ultrasonic</i> pada bak 2 di Serial Monitor.
Serial.print(" C: ");	Mencetak teks " C: " di Serial Monitor.
Serial.print(ultra3);	Mencetak nilai jarak dari sensor <i>Ultrasonic</i> pada bak 3 di Serial Monitor.
Serial.print(" K: ");	Mencetak teks " K: " di Serial Monitor.
Serial.print(kelembaban);	Mencetak nilai kelembaban tanah di Serial Monitor.
String text = "";	Membuat variabel text dengan tipe data String dan menginisialisasinya dengan nilai kosong.
text = " pump1:" + String(digitalRead(pump1)) + " pump2:" + String(digitalRead(pump2)) + " pump3:" + String(digitalRead(pump3)); <i>Blynk.virtualWrite</i> (V8, text);	Menggabungkan teks dan status pompa air pada variabel text.
display_lcd();	Mengirim nilai dari variabel text ke pin virtual V8 pada aplikasi <i>Blynk</i> .
Serial.println();	Memanggil fungsi <i>display_lcd()</i> untuk menampilkan informasi di layar LCD.
void setup(){...}	Mencetak baris kosong di Serial Monitor.
Serial.begin(9600);	Melakukan inisialisasi/konfigurasi input dan output saat ESP32 pertama kali menyala.
Serial.println(" starting "); pinMode(pinled, OUTPUT); pinMode(pump1, OUTPUT); pinMode(pump2, OUTPUT); pinMode(pump3, OUTPUT); for(int i=0; i<3; i++) { ... }	Memulai komunikasi serial dengan baud rate 9600.
digitalWrite(pump1, 1);	Mencetak teks " starting " di Serial Monitor.
digitalWrite(pump2, 1);	Mengatur pin pinled sebagai output.
digitalWrite(pump3, 1);	Mengatur pin pump1 sebagai output.
	Mengatur pin pump2 sebagai output.
	Mengatur pin pump3 sebagai output.
	Melakukan iterasi tiga kali untuk mengatur pin pin_trig dan pin_echo sebagai output dan input, sesuai dengan konfigurasi sensor <i>Ultrasonic</i> .
	Mematikan pompa air pada pump1 dengan mengatur pin pump1 menjadi HIGH.
	Mematikan pompa air pada pump2 dengan mengatur pin pump2 menjadi HIGH.
	Mematikan pompa air pada pump3 dengan mengatur pin pump3 menjadi HIGH.

Pengkodean	Penjelasan
<code>lcd.begin();</code>	Menginisialisasi layar LCD.
<code>lcd.backlight();</code>	Mengaktifkan backlight pada layar LCD.
<code>lcd.clear();</code>	Menghapus konten yang ditampilkan di layar LCD.
<code>lcd.setCursor(0, 0);</code>	Menetapkan posisi kursor LCD ke baris 0, kolom 0.
<code>lcd.print("ssid:");</code>	Menampilkan teks "ssid:" pada LCD.
<code>lcd.print(ssid);</code>	Menampilkan nilai dari variabel ssid pada LCD.
<code>lcd.setCursor(0, 1);</code>	Menetapkan posisi kursor LCD ke baris 1, kolom 0.
<code>lcd.print("pass:");</code>	Menampilkan teks "pass:" pada LCD.
<code>lcd.print(pass);</code>	Menampilkan nilai dari variabel pass pada LCD.
<code>Blynk.begin(auth, ssid, pass);</code>	Memulai koneksi dengan server <i>Blynk</i> menggunakan autentikasi dan SSID serta password WiFi yang telah diberikan.
<code>timers.setInterval(1000L, tasktimer);</code>	Menetapkan interval 1000ms (1 detik) untuk menjalankan fungsi <code>tasktimer()</code> menggunakan timer.
<code>Serial.println(" ready ");</code>	Mencetak teks " ready " di Serial Monitor.
<code>timeClient.begin();</code>	Memulai waktu klien NTP.
<code>timers.run();</code>	Menjalankan timer untuk fungsi yang telah diatur menggunakan <code>timers.setInterval()</code> .
<code>Blynk.run();</code>	Menjalankan koneksi <i>Blynk</i> .

4.4 Evaluasi Pengguna

Pada tahapan ini peneliti memberikan *Prototyping* yang sudah dibangun ke pengguna untuk melakukan pengujian, jika terdapat perbaikan maka peneliti akan menyesuaikan kembali *Prototyping* dengan keinginan pengguna dan melakukan perbaikan sistem. Evaluasi pengguna dilakukan dengan menggunakan metode *Black Box Testing*. Berikut merupakan tahapan evaluasi pengguna dengan para pengelola Kebun Nara Kupu Yogyakarta.



Gambar 4. 31 Wawancara dengan Pengelola Kebun Organik Nara Kupu Yogyakarta



Gambar 4. 32 Peneliti melakukan Presentasi *Prototype* yang telah dibuat.



Gambar 4. 33 Peneliti Melakukan Presentasi Alat kepada Pengelola Kebun Organik Nara Kupu Yogyakarta

4.4.1 Black Box Testing

Black Box Testing dilakukan untuk melakukan evaluasi pengguna. Pengujian *Black Box* berisi mengenai fitur sistem IoT kebun organik, serta deskripsi fitur-fitur yang diuji dan notifikasi yang dihasilkan dari masing-masing fitur tersebut. Evaluasi dan analisis hasil pengujian ini akan menjadi dasar untuk menilai sejauh mana sistem IoT kebun organik dapat memenuhi kebutuhan dan ekspektasi pengguna. Berikut merupakan *Black Box Testing* yang disajikan dalam bentuk tabel.

Tabel 4. 17 *Black Box Testing*

No.	Skenario	Pengujian	Hasil yang Diharapkan	Kesimpulan
1	Penyiraman Air	1. Penjadwalan penyiraman: Memeriksa waktu penyiraman air	Berhasil: Proses penyiraman air dilakukan pada jam 6 pagi dan 6 sore setiap hari.	<input checked="" type="checkbox"/> Valid <input type="checkbox"/> Tidak Valid
		2. Sensor kelembaban tanah: Menyiram saat kelembaban < 50%	Berhasil: Kelembaban tanah rendah, proses penyiraman air	<input checked="" type="checkbox"/> Valid <input type="checkbox"/> Tidak Valid
		1. Sensor suhu tanah: Menyiram saat suhu < 31°C	Berhasil: Suhu tanah rendah, proses penyiraman air	<input checked="" type="checkbox"/> Valid <input type="checkbox"/> Tidak Valid
		2. Pengiriman notifikasi: Memeriksa pengiriman notifikasi	Berhasil: Pengiriman notifikasi berhasil dilakukan	<input checked="" type="checkbox"/> Valid <input type="checkbox"/> Tidak Valid
2	Penyiraman Pupuk Cair	1. Penjadwalan penyiraman: Memeriksa proses penyiraman setiap 2 minggu	Berhasil: Proses penyiraman pupuk cair berhasil dilakukan	<input checked="" type="checkbox"/> Valid <input type="checkbox"/> Tidak Valid
		2. Pengiriman notifikasi: Memeriksa pengiriman notifikasi	Berhasil: pengiriman notifikasi berhasil dilakukan	<input checked="" type="checkbox"/> Valid <input type="checkbox"/> Tidak Valid

No.	Skenario	Pengujian	Hasil yang Diharapkan	Kesimpulan
3	Penyiraman Pesticida	1. Penjadwalan penyiraman: Memeriksa proses penyiraman setiap minggu	Berhasil: Proses penyiraman pestisida berhasil dilakukan	[✓] Valid [] Tidak Valid
		2. Pengiriman notifikasi: Memeriksa pengiriman notifikasi	Berhasil: Pengiriman notifikasi berhasil dilakukan	[✓] Valid [] Tidak Valid
4	Pemantauan Bak Air	1. Pemantauan level air: Memeriksa pemantauan level air	Berhasil: Pemantauan level air berhasil dilakukan	[✓] Valid [] Tidak Valid
		2. Pengiriman notifikasi: Memeriksa pengiriman notifikasi	Berhasil: Pengiriman notifikasi berhasil dilakukan	[✓] Valid [] Tidak Valid
5	Pemantauan Bak Pupuk Cair	1. Pemantauan level pupuk cair: Memeriksa pemantauan level pupuk cair	Berhasil: Pemantauan level pupuk cair berhasil dilakukan	[✓] Valid [] Tidak Valid
		2. Pengiriman notifikasi: Memeriksa pengiriman notifikasi	Berhasil: Pengiriman notifikasi berhasil dilakukan	[✓] Valid [] Tidak Valid
6	Pemantauan Bak Pesticida	1. Pemantauan level pestisida: Memeriksa pemantauan level pestisida	Berhasil: Pemantauan level air berhasil dilakukan	[✓] Valid [] Tidak Valid
		2. Pengiriman notifikasi: Memeriksa pengiriman notifikasi	Berhasil: Pengiriman notifikasi berhasil dilakukan	[✓] Valid [] Tidak Valid

Setelah melakukan evaluasi pengguna dengan metode *Black Box Testing* dengan pengelola kebun organik Nara Kupu Yogyakarta, didapatkan bahwa sistem yang dibangun sudah berjalan sesuai dengan kebutuhan pengguna. Namun, pengguna meminta peneliti untuk merubah jam penyiraman pupuk cair dan pestisida yang masing-masing dilakukan pada jam 6 pagi. Menurut pengguna jika penyiraman pupuk cair dan pestisida dilakukan pada jam 6 pagi atau sama dengan penyiraman air, maka kadar pupuk cair dan pestisida yang diberikan ke tanaman akan berkurang karena tercampur dengan air. Pengguna juga menyampaikan bahwa penyiraman pupuk cair dan pestisida dengan jarak waktu 1 jam dari penyiraman air sudah cukup. Maka dari itu, peneliti melakukan perbaikan pada jam penyiraman pupuk cair dan pestisida. Berikut merupakan *pseudocode* dari perbaikan tersebut pada tabel 4. 18 dan 4. 19.

Tabel 4. 18 Jam penyiraman pupuk cair dan pestisida sebelum dilakukan evaluasi pengguna dan perbaikan.

Pengkodean	Penjelasan
int jampupuk = 6;	Jam penyiraman pupuk cair pada jam 6 pagi.
int jamhama = 6;	Jam penyiraman pestisida pada jam 6 pagi.

Tabel 4. 19 Jam penyiraman pupuk cair dan pestisida sesudah dilakukan evaluasi pengguna dan perbaikan.

Pengkodean	Penjelasan
int jampupuk = 7;	Jam penyiraman pupuk cair pada jam 7 pagi.
int jamhama = 8;	Jam penyiraman pestisida pada jam 8 pagi.

Maka dari itu, penyiraman pupuk cair yang dilakukan pada setiap 2 minggu sekali dilakukan pada jam 7 pagi, dengan jarak 1 jam dengan penyiraman air rutin yang dilakukan pada jam 6 pagi. Sedangkan penyiraman pestisida yang dilakukan pada setiap minggu sekali dilakukan pada jam 8 pagi, dengan jarak 1 jam dari penyiraman pupuk cair, dan 2 jam dari penyiraman air rutin agar kadar penyiramannya tidak tercampur.

4.5 Implementasi Produk

Implementasi produk pada penelitian ini berisi mengenai penggabungan antara pembangunan *Prototyping* secara *software* dengan *hardware* setelah dilakukannya evaluasi pengguna dan perbaikan sistem, sehingga fitur-fitur utama dari *Prototyping* yang sudah dibangun dapat diuji dan implementasikan secara nyata.

4.5.1 Pengaturan Tampilan *Blynk* pada *Smartphone*

Pada tahapan ini peneliti melakukan pengaturan tampilan *Blynk* pada *Smartphone* dengan tujuan bisa melakukan *monitoring* dan *controlling* kebun organik melalui *smartphone* pengguna. Berikut merupakan tahapannya.

Membuka menu pengaturan (*Icon* kunci inggris) pada aplikasi *Blynk* di *smartphone*.



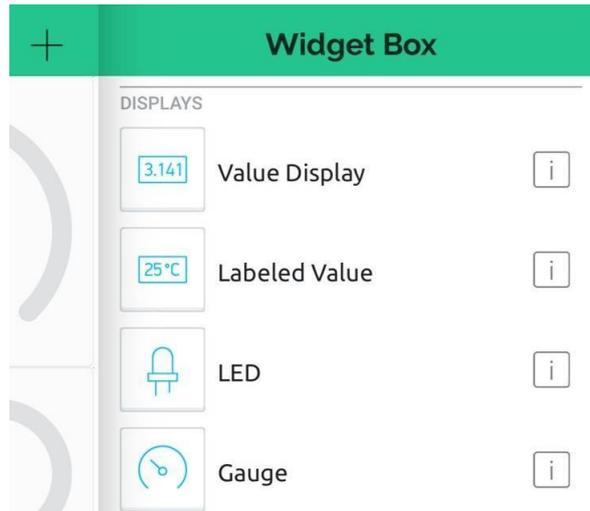
Gambar 4. 34 Membuka menu pengaturan

Kemudian membuka menu *Widget Box* dengan meng-klik *icon* tambah.



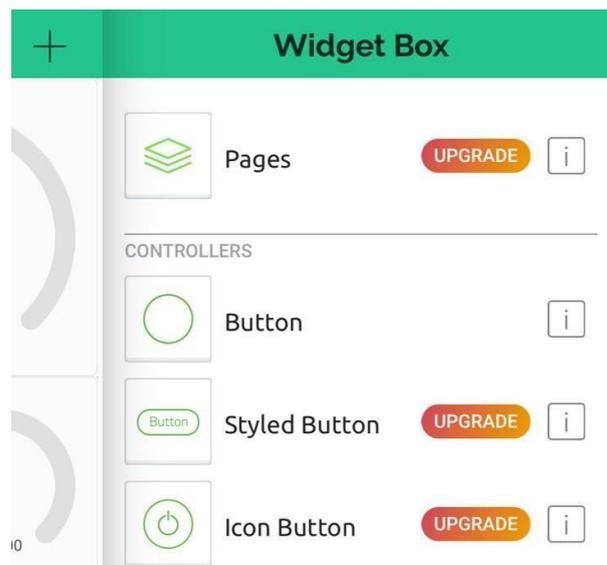
Gambar 4. 35 membuka menu *Widget Box*

Selanjutnya peneliti menambahkan 5 *widget Gauge*, untuk menampilkan pembacaan sensor *Ultrasonic*, kelembaban tanah dan suhu tanah.



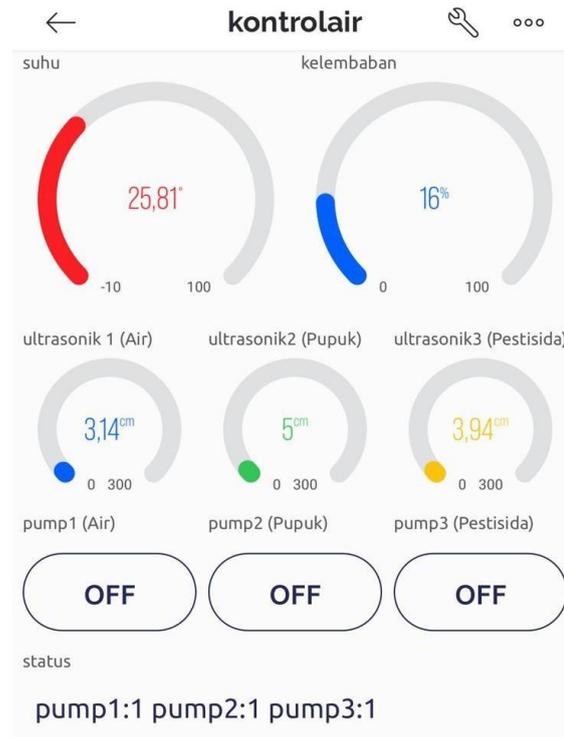
Gambar 4. 36 Menambahkan *Widget Gauge*

Setelah itu peneliti menambahkan 3 *widget Button* yang digunakan sebagai tombol *On/Off* dari 3 pompa air.



Gambar 4. 37 Menambahkan *Widget Button*

Setelah itu didapatkan tampilan sistem *monitoring* dan *controlling* kebun organik pada aplikasi *Blynk* di *Smartphone*. Berikut merupakan tampak dari tampilan tersebut.



Gambar 4. 38 Tampilan Aplikasi *Blynk* di *Smartphone*

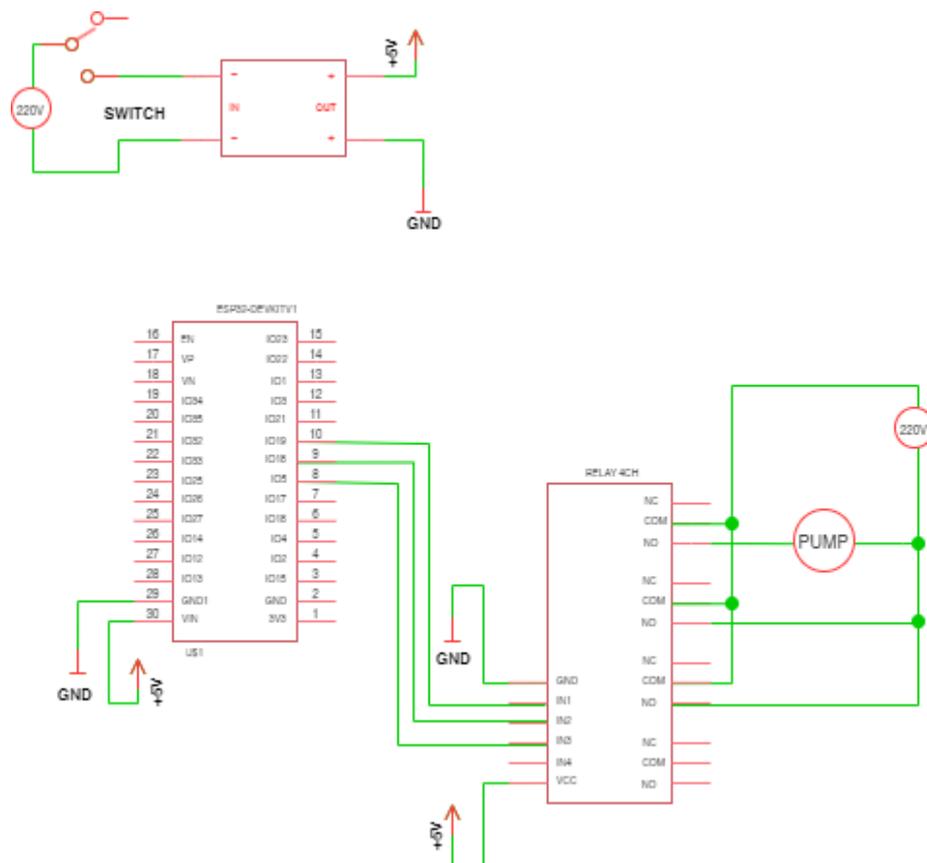
4.5.2 Penyiraman Air Rutin

Fitur ini memungkinkan sistem untuk melakukan automasi penyiraman air secara rutin pada tanaman, penjadwalan penyiraman air dilakukan 2 kali sehari pada pukul 6 pagi dan sore. Untuk melakukan fitur ini, diperlukan beberapa *hardware* Berikut merupakan daftar *hardware* yang digunakan pada fitur ini pada tabel 4. 20.

Tabel 4. 20 Daftar *hardware* fitur penyiraman air rutin.

Nama Hardware
ESP32
Pompa air
Bak penampungan
Selang air

Selanjutnya, berikut merupakan rangkaian *hardware* pada fitur ini yang dibuat dalam bentuk *flowchart*.



Gambar 4. 39 Rangkaian *Hardware* dalam bentuk *flowchart*.

Setelah melakukan rangkaian *hardware*, peneliti melakukan pengkodean untuk fitur tersebut. Berikut merupakan *pseudocode* untuk fitur penyiraman air secara rutin pada tabel 4. 21.

Tabel 4. 21 *Pseudocode* fitur penyiraman rutin.

Pengkodean	Penjelasan
<pre> if(jam == air1){ if(lock3!=1){ lock3=1; tpump1=0; digitalWrite(pump1, 0); Blynk.logEvent("notif", "Penyiraman otomatis pagi"); } if(tpump1>=waktupump1) { if(lock3!=2){ lock3=2; digitalWrite(pump1, 1); } </pre>	<p>Memeriksa apakah jam saat ini sama dengan waktu penyiraman air pagi (air1). Jika ya, maka sistem melakukan penyiraman selama waktu yang sudah ditentukan dan mengirimkan notif ke aplikasi <i>Blynk</i>.</p>

Pengkodean	Penjelasan
<pre> } else {tpump1++;} if(jam == air2){ if(lock3!=3){ lock3=3; tpump1=0; digitalWrite(pump1, 0); Blynk.logEvent("notif","Penyiraman otomatis sore"); } if(tpump1>=waktupump1) { if(lock3!=4){ lock3=4; digitalWrite(pump1, 1); } } else {tpump1++;} } </pre>	<p>Memeriksa apakah jam saat ini sama dengan waktu penyiraman air sore (air2). Jika ya, maka sistem melakukan penyiraman selama waktu yang sudah ditentukan dan mengirimkan notif ke aplikasi <i>Blynk</i>.</p>

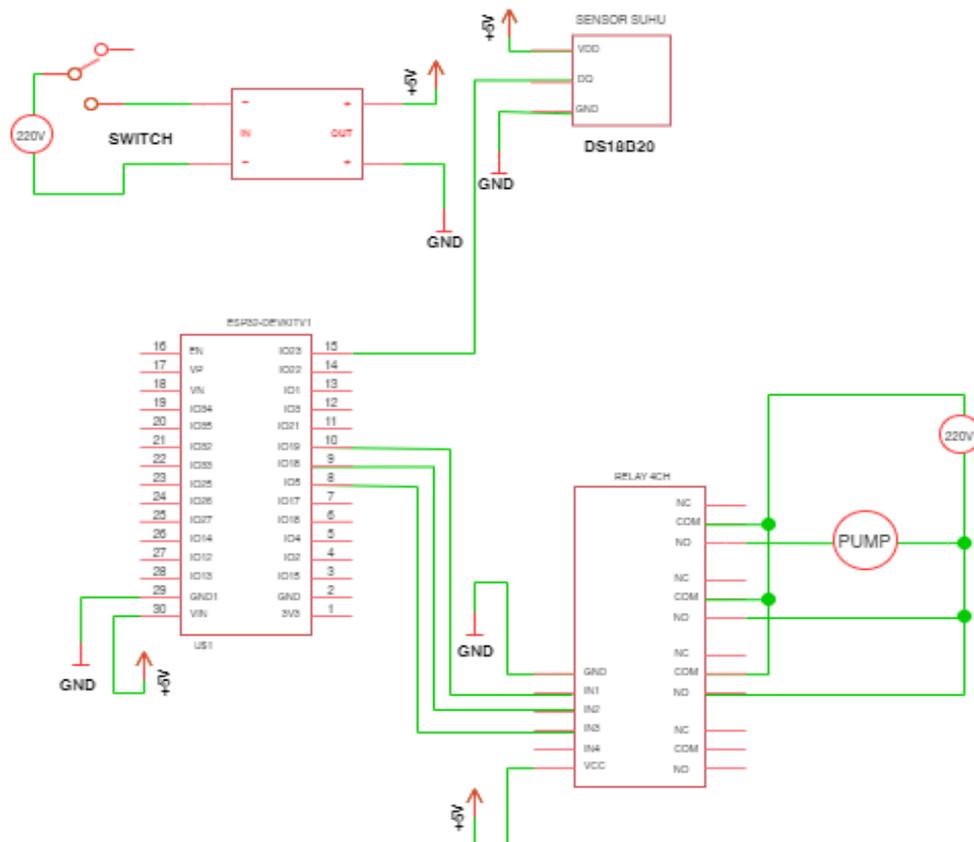
4.5.3 Penyiraman Air ketika Suhu Melebihi Batas

Proses penyiraman air juga akan dilakukan jika sensor mendeteksi bahwa suhu tanah melebihi nilai yang sudah ditentukan (31°C). Setelah proses penyiraman air selesai, sistem akan mengirimkan notifikasi ke *smartphone* pengguna melalui aplikasi *Blynk*. Untuk melakukan fitur ini, diperlukan beberapa *hardware* Berikut merupakan daftar *hardware* yang digunakan pada fitur ini pada tabel 4. 22.

Tabel 4. 22 Daftar *hardware* fitur penyiraman ketika suhu melebihi batas.

Nama <i>Hardware</i>
ESP32
Pompa air
Bak penampungan
Selang air
Sensor suhu tanah

Selanjutnya, berikut merupakan rangkaian *hardware* pada fitur ini yang dibuat dalam bentuk *flowchart*.



Gambar 4. 40 Rangkaian *Hardware* dalam bentuk *flowchart*.

Setelah melakukan rangkaian *hardware*, peneliti melakukan pengkodean untuk fitur tersebut. Berikut merupakan *pseudocode* untuk fitur penyiraman air ketika suhu melebihi batas pada tabel 4. 23.

Tabel 4. 23 *Pseudocode* penyiraman air ketika suhu melebihi batas.

Pengkodean	Penjelasan
<pre> if(suhu > setsuhu){ if(lock1!=1){ lock1=1; digitalWrite(pump1, 0); Blynk.logEvent("notif","Suhu melebihi batas, sistem melakukan penyiraman otomatis"); } } </pre>	<p>Memeriksa apakah suhu saat ini melebihi suhu yang telah ditetapkan (setsuhu). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i> dan melakukan penyiraman air.</p>
<pre> if(suhu < setsuhu){ if(lock1!=0){ lock1=0; digitalWrite(pump1, 1); } } </pre>	<p>Memeriksa apakah suhu saat ini kurang dari suhu yang telah ditetapkan (setsuhu). Jika ya, maka sistem mematikan pompa (memberhentikan penyiraman).</p>

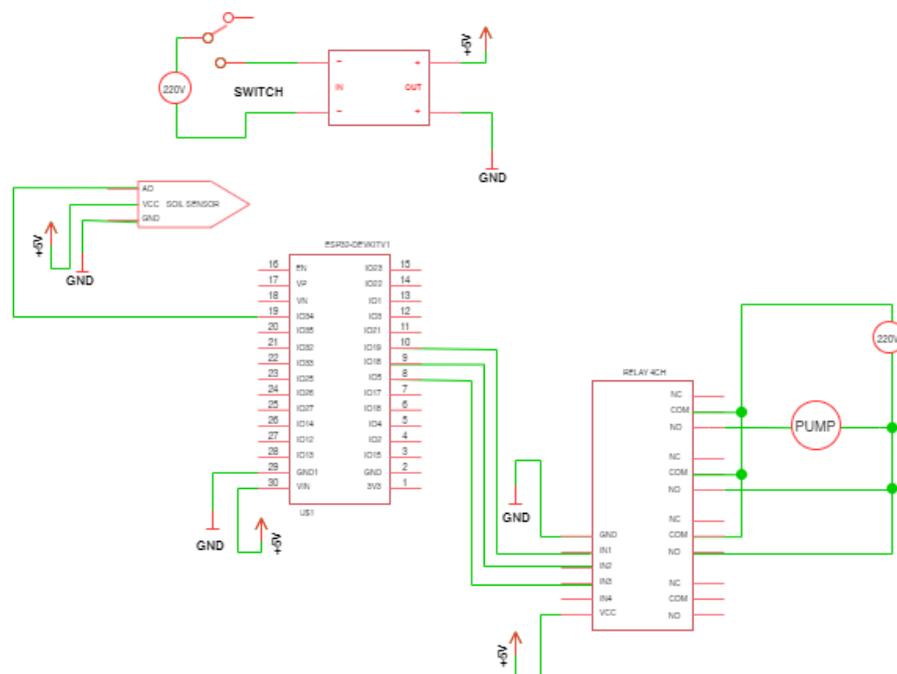
4.5.4 Penyiraman Air ketika Kelembaban kurang dari Batas

Proses penyiraman air juga akan dilakukan jika sensor mendeteksi bahwa kelembaban tanah kurang dari nilai yang sudah ditentukan (50%). Setelah proses penyiraman air selesai, sistem akan mengirimkan notifikasi ke *smartphone* pengguna melalui aplikasi *Blynk*. Untuk melakukan fitur ini, diperlukan beberapa *hardware*. Berikut merupakan daftar *hardware* yang digunakan pada fitur ini pada tabel 4. 24.

Tabel 4. 24 Daftar *hardware* fitur penyiraman air ketika kelembaban kurang dari batas.

Nama Hardware
Pompa air
Bak penampungan
Selang air
Sensor kelembaban tanah

Selanjutnya, berikut merupakan rangkaian *hardware* pada fitur ini yang dibuat dalam bentuk *flowchart*.



Gambar 4. 41 Rangkaian *Hardware* dalam bentuk *flowchart*.

Setelah melakukan rangkaian *hardware*, peneliti melakukan pengkodean untuk fitur tersebut. Berikut merupakan *pseudocode* untuk fitur penyiraman air ketika suhu melebihi batas pada tabel 4. 25.

Tabel 4. 25 *Pseudocode* penyiraman air ketika kelembaban kurang dari batas.

Pengkodean	Penjelasan
<pre> if(kelembaban < setkelembaban){ if(lock2!=1){ lock2=1; digitalWrite(pump1, 0); Blynk.logEvent("notif","Kelembaban kurang dari batas, sistem melakukan penyiraman otomatis"); } } </pre>	<p>Memeriksa apakah kelembaban tanah saat ini kurang dari kelembaban yang telah ditetapkan (setkelembaban). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i> dan melakukan penyiraman air.</p>
<pre> if(kelembaban > setkelembaban){ if(lock2!=0){ lock2=0; digitalWrite(pump1, 1); } } </pre>	<p>Memeriksa apakah kelembaban tanah saat ini melebihi kelembaban yang telah ditetapkan (setkelembaban). Jika ya, maka sistem mematikan pompa (memberhentikan penyiraman).</p>

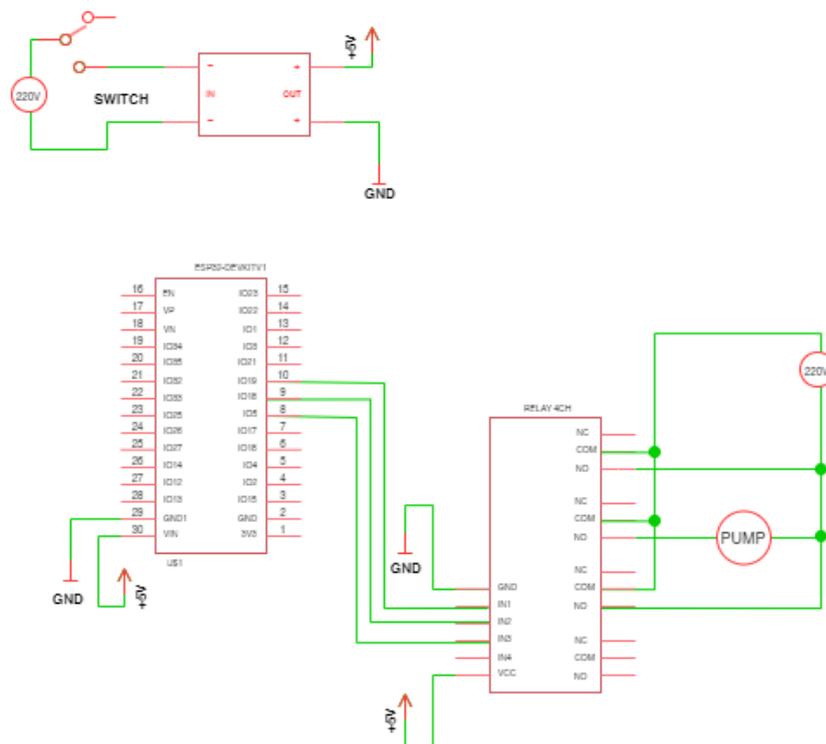
4.5.5 Penyiraman Pupuk Cair

Fitur ini memungkinkan sistem untuk melakukan automasi penyiraman pupuk cair pada tanaman. Penjadwalan penyiraman pupuk cair dilakukan setiap 2 minggu sekali dalam sebulan. Setelah proses penyiraman pupuk cair selesai, sistem akan mengirimkan notifikasi ke *smartphone* pengguna melalui aplikasi *Blynk*. Untuk melakukan fitur ini, diperlukan beberapa *hardware* Berikut merupakan daftar *hardware* yang digunakan pada fitur ini pada tabel 4. 26.

Tabel 4. 26 Daftar *hardware* fitur penyiraman pupuk cair.

Nama <i>Hardware</i>
Pompa air
Bak penampungan
Selang air
ESP32

Selanjutnya, berikut merupakan rangkaian *hardware* pada fitur ini yang dibuat dalam bentuk *flowchart*.



Gambar 4. 42 Rangkaian *Hardware* dalam bentuk *flowchart*.

Setelah melakukan rangkaian *hardware*, peneliti melakukan pengkodean untuk fitur tersebut. Berikut merupakan *pseudocode* untuk fitur penyiraman pupuk cair pada tabel 4. 27.

Tabel 4. 27 *Pseudocode* penyiraman pupuk cair.

Pengkodean	Penjelasan
<pre> if(monthDay == pupukcair1 && jam==jampupuk){ if(lock4!=1){ lock4=1; tpump2=0; digitalWrite(pump2, 0); Blynk.logEvent("notif", "Penyiraman pupuk cair minggu ke 1"); } if(tpump2>=waktupump2) { if(lock4!=2){ lock4=2; digitalWrite(pump2, 1); } } else {tpump2++;} </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyiraman pupuk cair minggu ke 1 dan waktu penyiramannya. Jika ya, maka sistem melakukan penyiraman pupuk cair selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>

Pengkodean	Penjelasan
<pre> } if(monthDay == pupukcair2 && jam==jampupuk){ if(lock4!=3){ lock4=3; tpump2=0; digitalWrite(pump2, 0); Blynk.logEvent("notif","Penyiraman pupuk cair minggu ke 3"); } if(tpump2>=waktupump2) { if(lock4!=4){ lock4=4; digitalWrite(pump2, 1); } } else {tpump2++;} } </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyiraman pupuk cair minggu ke 3 dan waktu penyiramannya. Jika ya, maka sistem melakukan penyiraman pupuk cair selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>

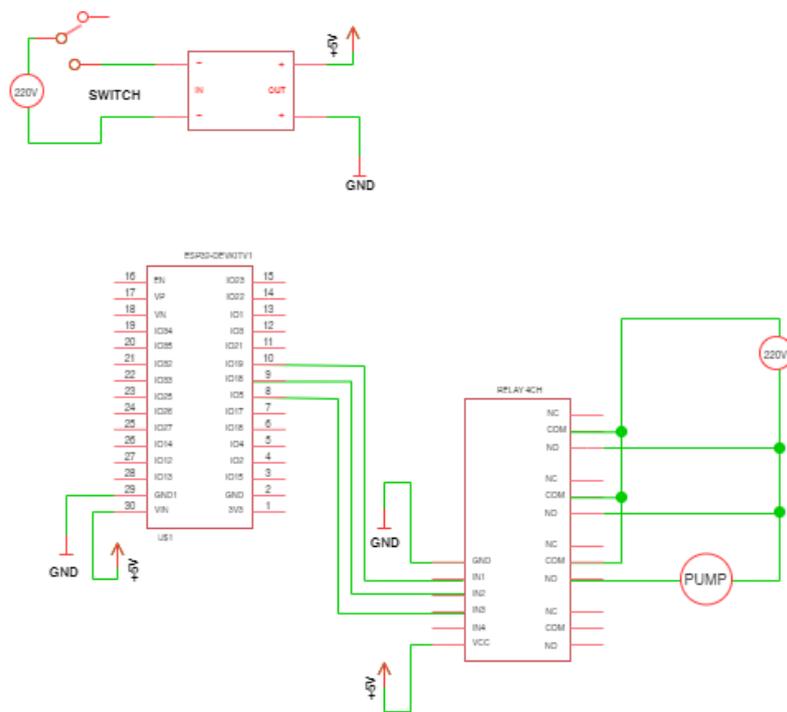
4.5.6 Penyiraman Pestisida

Fitur ini memungkinkan sistem untuk melakukan automasi penyiraman pestisida pada tanaman. Penjadwalan penyiraman pestisida dilakukan setiap minggu sekali. Setelah proses penyiraman pestisida selesai, sistem akan mengirimkan notifikasi ke *smartphone* pengguna melalui aplikasi *Blynk*. Untuk melakukan fitur ini, diperlukan beberapa *hardware* Berikut merupakan daftar *hardware* yang digunakan pada fitur ini pada tabel 4. 28.

Tabel 4. 28 Daftar *hardware* fitur penyiraman pestisida.

Nama Hardware
Pompa air
Bak penampungan
Selang air
ESP32

Selanjutnya, berikut merupakan rangkaian *hardware* pada fitur ini yang dibuat dalam bentuk *flowchart*.



Gambar 4. 43 Rangkaian *Hardware* dalam bentuk *flowchart*.

Setelah melakukan rangkaian *hardware*, peneliti melakukan pengkodean untuk fitur tersebut. Berikut merupakan *pseudocode* untuk fitur penyiraman pestisida pada tabel 4. 29.

Tabel 4. 29 *Pseudocode* penyiraman pestisida.

Pengkodean	Penjelasan
<pre> if(monthDay == hamal && jam==jamhama){ if(lock5!=1){ lock5=1; tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif","Penyemprotan pestisida minggu ke 1"); } if(tpump3>=waktupump3) { if(lock5!=2){ lock5=2; digitalWrite(pump3, 1); } } else {tpump3++;} </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 1 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>

Pengkodean	Penjelasan
<pre> } if(monthDay == hama2 && jam==jamhama){ if(lock5!=3){ lock5=3; tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif", "Penyemprotan pestisida minggu ke 2"); } if(tpump3>=waktupump3) { if(lock5!=4){ lock5=4; digitalWrite(pump3, 1); } } else {tpump3++;} } </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 2 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(monthDay == hama3 && jam==jamhama){ if(lock5!=5){ lock5=5; tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif", "Penyemprotan pestisida minggu ke 3"); } if(tpump3>=waktupump3) { if(lock5!=6){ lock5=6; digitalWrite(pump3, 1); } } else {tpump3++;} } </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 3 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(monthDay == hama4 && jam==jamhama){ if(lock5!=7){ lock5=7; tpump3=0; digitalWrite(pump3, 0); </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 4 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>

Pengkodean	Penjelasan
<pre> Blynk.logEvent("notif", "Penyemprotan pestisida minggu ke 4"); } if(tpump3>=waktupump3) { if(lock5!=8){ lock5=8; digitalWrite(pump3, 1); } } else {tpump3++;} } </pre>	

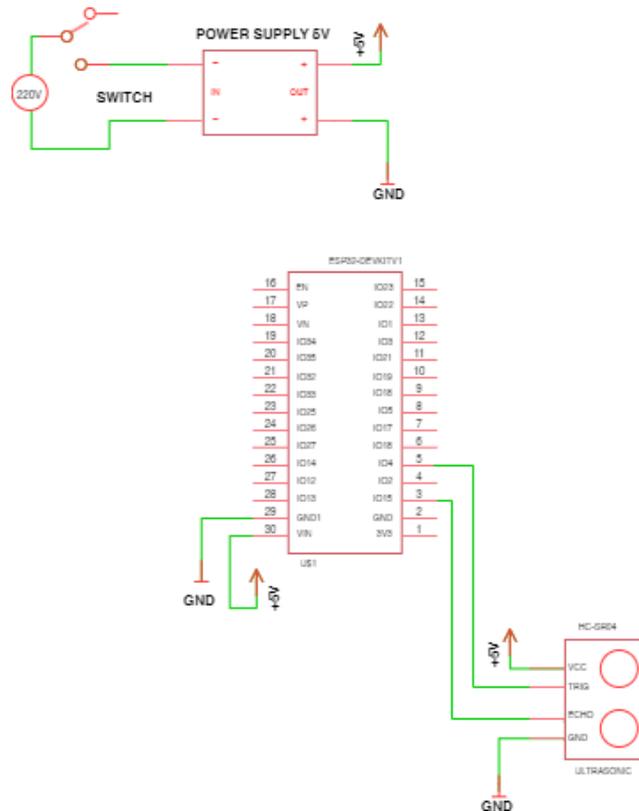
4.5.7 Pemantauan Bak Air

Fitur ini memungkinkan sistem untuk memantau level air pada bak yang digunakan untuk penyiraman air. Jika jumlah air dalam bak sudah hampir habis (sekitar ± 5 cm dari dasar), sistem akan mengirimkan notifikasi ke *smartphone* pengguna melalui aplikasi *Blynk* untuk mengingatkan pengguna untuk melakukan pengisian ulang. Untuk melakukan fitur ini, diperlukan beberapa *hardware* Berikut merupakan daftar *hardware* yang digunakan pada fitur ini pada tabel 4. 30.

Tabel 4. 30 Daftar *hardware* fitur pemantauan bak air.

Nama <i>Hardware</i>
Bak penampungan
Selang air
Sensor <i>Ultrasonic</i>
ESP32

Selanjutnya, berikut merupakan rangkaian *hardware* pada fitur ini yang dibuat dalam bentuk *flowchart*.



Gambar 4. 44 Rangkaian *Hardware* dalam bentuk *flowchart*.

Setelah melakukan rangkaian *hardware*, peneliti melakukan pengkodean untuk fitur tersebut. Berikut merupakan *pseudocode* untuk fitur pemantauan bak air pada tabel 4. 31.

Tabel 4. 31 *Pseudocode* pemantauan bak air.

Pengkodean	Penjelasan
<pre> if(tinggi1 < habis1){ if(lock6!=1){ lock6=1; Blynk.logEvent("notif","Air habis, isi bak segera!"); } } </pre>	<p>Memeriksa apakah tinggi air pada bak 1 lebih rendah dari batas habis (habis1). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tinggi1 > habis1){ lock6=0; } </pre>	<p>Memeriksa apakah tinggi air pada bak 1 lebih tinggi dari batas habis (habis1). Jika ya, maka sistem memberhentikan notifikasi.</p>

4.5.8 Pemantauan Bak Pupuk Cair

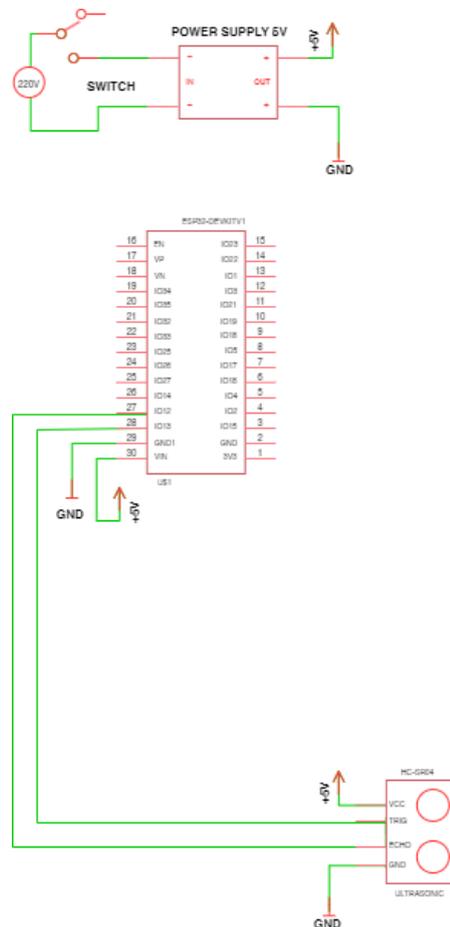
Fitur ini memungkinkan sistem untuk memantau level pupuk cair pada bak yang digunakan untuk penyiraman pupuk cair. Jika jumlah pupuk cair dalam bak sudah hampir habis (sekitar ± 5 cm dari dasar), sistem akan mengirimkan notifikasi ke *smartphone*

pengguna melalui aplikasi *Blynk* untuk mengingatkan pengguna untuk melakukan pengisian ulang. Untuk melakukan fitur ini, diperlukan beberapa *hardware*. Berikut merupakan daftar *hardware* yang digunakan pada fitur ini pada tabel 4. 32.

Tabel 4. 32 Daftar *hardware* fitur pemantauan bak pupuk cair.

Nama <i>Hardware</i>
Bak penampungan
Selang air
Sensor <i>Ultrasonic</i>
ESP32

Selanjutnya, berikut merupakan rangkaian *hardware* pada fitur ini yang dibuat dalam bentuk *flowchart*.



Gambar 4. 45 Rangkaian *Hardware* dalam bentuk *flowchart*.

Setelah melakukan rangkaian *hardware*, peneliti melakukan pengkodean untuk fitur tersebut. Berikut merupakan *pseudocode* untuk fitur pemantauan bak pupuk cair pada tabel 4. 33.

Tabel 4. 33 *Pseudocode* pemantauan bak pupuk cair.

Pengkodean	Penjelasan
<pre> if(tinggi2 < habis2){ if(lock7!=1){ lock7=1; Blynk.logEvent("notif","Pupuk habis, isi bak segera!"); } } </pre>	Memeriksa apakah tinggi air pada bak 2 lebih rendah dari batas habis (habis2). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i> .
<pre> if(tinggi2 > habis2){ lock7=0; } </pre>	Memeriksa apakah tinggi air pada bak 2 lebih tinggi dari batas habis (habis2). Jika ya, maka sistem memberhentikan notifikasi.

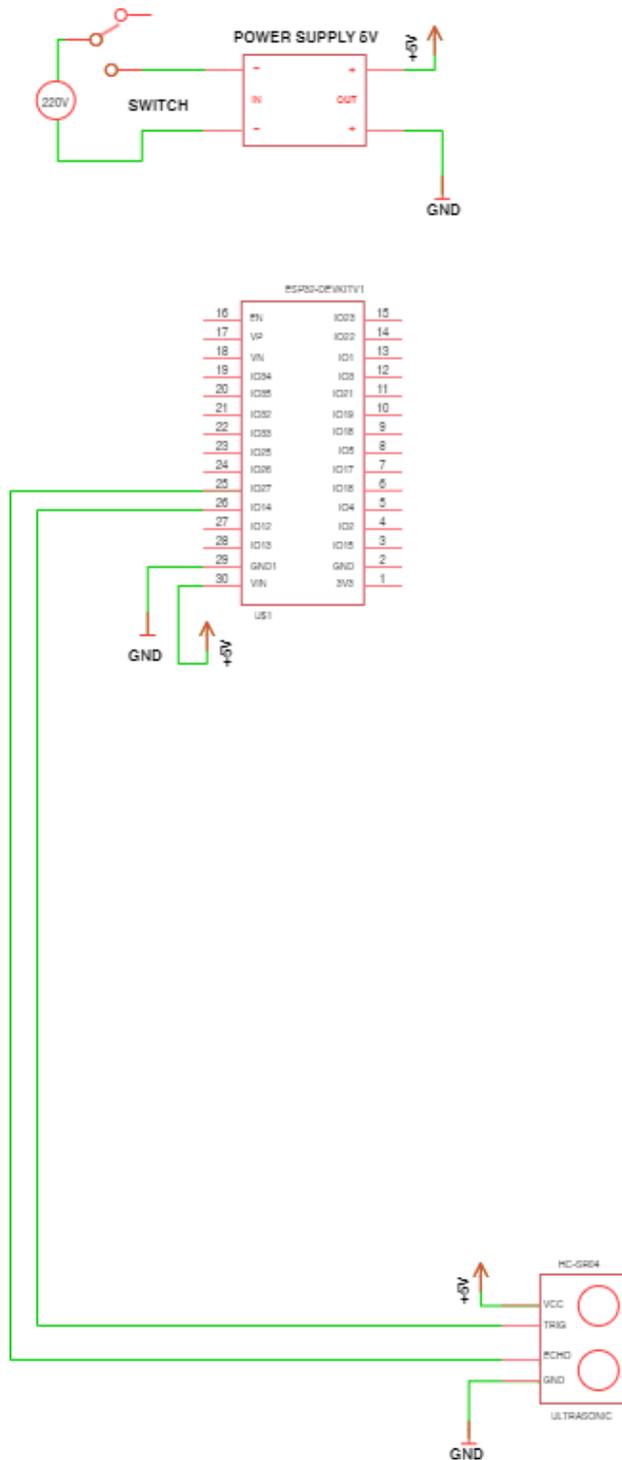
4.5.9 Pemantauan Bak Pestisida

Fitur ini memungkinkan sistem untuk memantau level pestisida pada bak yang digunakan untuk penyiraman pestisida. Jika jumlah pestisida dalam bak sudah hampir habis (sekitar ± 5 cm dari dasar), sistem akan mengirimkan notifikasi ke *smartphone* pengguna melalui aplikasi *Blynk* untuk mengingatkan pengguna untuk melakukan pengisian ulang. Untuk melakukan fitur ini, diperlukan beberapa *hardware*. Berikut merupakan daftar *hardware* yang digunakan pada fitur ini pada tabel 4. 33.

Tabel 4. 34 Daftar *hardware* fitur pemantauan bak pestisida.

Nama Hardware
Bak penampungan
Selang air
Sensor <i>Ultrasonic</i>
ESP32

Selanjutnya, berikut merupakan rangkaian *hardware* pada fitur ini yang dibuat dalam bentuk *flowchart*.



Gambar 4. 46 Rangkaian *Hardware* dalam bentuk *flowchart*.

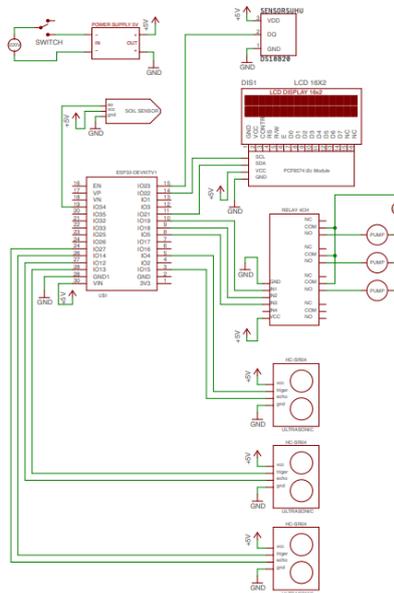
Setelah melakukan rangkaian *hardware*, peneliti melakukan pengkodean untuk fitur tersebut. Berikut merupakan *pseudocode* untuk fitur pemantauan bak pestisida pada tabel 4. 35.

Tabel 4. 35 *Pseudocode* pemantauan bak pestisida.

Pengkodean	Penjelasan
<pre> if(tinggi3 < habis3){ if(lock8!=1){ lock8=1; Blynk.logEvent("notif","Pestisida habis, isi bak segera!"); } } if(tinggi3 > habis3){ lock8=0; } </pre>	<p>Memeriksa apakah tinggi air pada bak 3 lebih rendah dari batas habis (habis3). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i>.</p> <p>Memeriksa apakah tinggi air pada bak 3 lebih tinggi dari batas habis (habis3). Jika ya, maka sistem memberhentikan notifikasi.</p>

4.5.10 Rangkaian Sistem Keseluruhan

Rangkaian sistem keseluruhan dibuat untuk memudahkan peneliti dalam mengembangkan dan membangun rangkaian elektronik pada sistem yang telah dibuat. Berikut merupakan gambar dari rangkaian keseluruhan sistem *monitoring* dan *controlling* kebun organik.



Gambar 4. 47 Rangkaian Sistem Keseluruhan

Selanjutnya, berikut merupakan penjelasan mengenai rangkaian keseluruhan pada tabel 4. 36.

Tabel 4. 36 Rangkaian Keseluruhan

Komponen	Deskripsi
Sensor Kelembaban Tanah	Dihubungkan ke pin GPIO 34

Komponen	Deskripsi
Sensor Suhu Tanah DS18B20	Dihubungkan ke pin GPIO 23
Sensor <i>Ultrasonic</i> (Air)	Dihubungkan ke pin GPIO 4 (<i>Trig</i>) dan GPIO 15 (<i>Echo</i>)
Sensor <i>Ultrasonic</i> (Pupuk Cair)	Dihubungkan ke pin GPIO 13 (<i>Trig</i>) dan GPIO 12 (<i>Echo</i>)
Sensor <i>Ultrasonic</i> (Pestisida)	Dihubungkan ke pin GPIO 14 (<i>Trig</i>) dan GPIO 27 (<i>Echo</i>)
PCF8574 i2c	Dihubungkan ke pin GPIO 22 (SCL) dan GPIO 21 (SDA)
LCD	Terhubung melalui PCF8574 i2c
<i>Relay 4 Channel</i>	Dihubungkan ke pin GPIO 19 (IN1), GPIO 18 (IN2), dan GPIO 5 (IN3).
<i>Adapter 12V</i>	Terhubung melalui COM (<i>Common</i>) <i>Relay 4 Channel</i>
Pompa Air	Terhubung melalui NO (<i>Normally Open</i>) pada <i>Relay 4 Channel</i>
<i>Adapter 5V</i>	Dihubungkan ke VIN dan <i>On/Off Switch</i>
<i>On/Off Switch</i>	Dihubungkan ke VIN dan <i>Adapter 5V</i>

4.5.11 Pengkodean Keseluruhan

Berikut merupakan *pseudocode* pengkodean secara keseluruhan yang dilakukan pada *software* Arduino IDE pada tabel 4.37.

Tabel 4. 37 Pengkodean Keseluruhan

Pengkodean	Penjelasan
#define pump1 19	Mendefinisikan konstanta pump1 yang merupakan pompa 1 yang terhubung pada pin GPIO nomor 19.
#define pump2 18	Mendefinisikan konstanta pump2 yang merupakan pompa 2 yang terhubung pada pin GPIO nomor 18.
#define pump3 5	Mendefinisikan konstanta pump3 yang merupakan pompa 3 yang terhubung pada pin GPIO nomor 5.
#define pinled 2	Mendefinisikan konstanta pinled yang merupakan LED yang terhubung pada pin GPIO nomor 2.

Pengkodean	Penjelasan
#define soil 34	Mendefinisikan konstanta soil yang merupakan sensor kelembaban tanah yang terhubung pada pin GPIO nomor 34.
#define pinsuhu 23	Mendefinisikan konstanta pinsuhu yang merupakan sensor suhu tanah yang terhubung pada pin GPIO nomor 23.
LiquidCrystal_I2C lcd(0x27, 16, 2);	Mendefinisikan LCD yang digunakan pada penelitian ini, yang memiliki ukuran 16 × 2.
#include <OneWire.h> #include <DallasTemperature.h>	Memanggil <i>library</i> <i>OneWire</i> dan <i>DallasTemperature</i> yang akan digunakan untuk sensor suhu tanah.
OneWire configsensor(pinsuhu); DallasTemperature	Membuat objek configsensor dengan menggunakan konstruktor <i>OneWire</i> dan menyediakan pin data yang telah didefinisikan sebelumnya.
sensor(&configsensor);	Membuat objek sensor dengan menggunakan konstruktor <i>DallasTemperature</i> dan menginisialisasinya dengan objek configsensor. Ini dilakukan agar objek sensor dapat berkomunikasi dengan sensor suhu berbasis <i>OneWire</i> yang terhubung ke pin data.
#include <BlynkSimpleEsp32.h> // library <i>Blynk</i>	Memanggil <i>library Blynk</i> .
#define BLYNK_TEMPLATE_ID "TMPL6OYcash5y"	Mendefinisikan ID template <i>Blynk</i> .
#define BLYNK_TEMPLATE_NAME "kontrolair"	Mendefinisikan nama template <i>Blynk</i> .
#define BLYNK_AUTH_TOKEN "vXRcQGCOlBvixS23m5oGGyaBrlWsb5V2"	Mendefinisikan token otorisasi <i>Blynk</i> yang diperlukan untuk menghubungkan perangkat ke server <i>Blynk</i> .
#define BLYNK_PRINT Serial	Mengarahkan output <i>Blynk</i> ke Serial Monitor untuk tampilan informasi <i>Blynk</i> .
<i>BlynkTimer</i> timers;	Membuat objek <i>timers</i> dari <i>BlynkTimer</i> yang digunakan untuk menjalankan tugas-tugas secara berkala.
char auth[] = BLYNK_AUTH_TOKEN;	Mendefinisikan variabel <i>auth</i> sebagai array karakter (char array) yang akan menyimpan token otentikasi <i>Blynk</i> .
char ssid[] = "Nama Wifi yang digunakan";	Mendefinisikan variabel <i>ssid</i> sebagai nama jaringan <i>Wi-Fi</i> yang akan digunakan.
char pass[] = "Password Wifi";	Mendefinisikan variabel <i>pass</i> sebagai kata sandi jaringan <i>Wi-Fi</i> yang akan digunakan.

Pengkodean	Penjelasan
#include <NTPClient.h>	Memanggil <i>Library</i> NTP yang digunakan untuk melakukan penjadwalan waktu melalui <i>software</i> .
const long utcOffsetInSeconds = 25200;	Mengatur offset waktu ke GMT+7 (Waktu Indonesia Barat).
WiFiUDP ntpUDP;	Membuat objek ntpUDP dari WiFiUDP yang digunakan untuk koneksi UDP dengan server NTP.
char pass[] = "Password Wifi";	Mendefinisikan variabel pass sebagai kata sandi jaringan <i>Wi-Fi</i> yang akan digunakan.
#include <NTPClient.h>	Memanggil <i>Library</i> NTP yang digunakan untuk melakukan penjadwalan waktu melalui <i>software</i> .
const long utcOffsetInSeconds = 25200;	Mengatur offset waktu ke GMT+7 (Waktu Indonesia Barat).
WiFiUDP ntpUDP;	Membuat objek ntpUDP dari WiFiUDP yang digunakan untuk koneksi UDP dengan server NTP.
NTPClient timeClient(ntpUDP, "id.pool.ntp.org", utcOffsetInSeconds);	Membuat objek timeClient dari NTPClient yang digunakan untuk mendapatkan waktu dari server NTP.
int monthDay;	Variabel untuk menyimpan hari bulan saat ini.
int currentMonth;	Variabel untuk menyimpan bulan saat ini.
int currentYear;	Variabel untuk menyimpan tahun saat ini.
String tanggal,waktu;	Variabel string untuk menyimpan tanggal dan waktu saat ini.
int jam;	Variabel untuk menyimpan jam saat ini.
int count1=0,count2=0;	Variabel untuk pengaturan tampilan LCD.
float ultra1,ultra2,ultra3;	Variabel untuk menyimpan nilai jarak yang diukur oleh sensor <i>Ultrasonic</i> .
float tinggi1,tinggi2,tinggi3;	Variabel untuk menyimpan nilai tinggi air di dalam tangki.
int tpump1=0,tpump2=0,tpump3=0;	Variabel untuk menghitung waktu pompa aktif.
float suhu;	Variabel untuk menyimpan nilai suhu yang diukur oleh sensor suhu.
int kelembaban;	Variabel untuk menyimpan nilai kelembaban tanah yang diukur.
int lock1,lock2,lock3;	Variabel pengunci untuk menghindari pengiriman notifikasi berulang kali.
int lock4,lock5;	Variabel pengunci untuk menghindari pengiriman notifikasi berulang kali.
int lock6,lock7,lock8;	Variabel pengunci untuk menghindari pengiriman notifikasi berulang kali.

Pengkodean	Penjelasan
int pin_trig[]={4,13,14,};	Array yang berisi pin trigger untuk sensor <i>Ultrasonic</i> .
int pin_echo[]={15,12,27,};	Array yang berisi pin echo untuk sensor <i>Ultrasonic</i> .
float setultra1 = 20.0;	Nilai tinggi air yang diatur untuk tangki 1.
float setultra2 = 20.0;	Nilai tinggi air yang diatur untuk tangki 2.
float setultra3 = 20.0;	Nilai tinggi air yang diatur untuk tangki 3.
float setsuhu = 31.0;	Nilai suhu yang diatur.
int setkelembaban = 50;	Nilai kelembaban tanah yang diatur.
int air1 = 6;	Jam penyiraman air pagi.
int air2 = 18;	Jam penyiraman air sore.
int jampupuk = 7;	Jam penyiraman pupuk cair
int jamhama = 8;	Jam penyiraman pestisida
int pupukcair1 = 1;	Tanggal pemberian pupuk cair minggu ke-1.
int pupukcair2 = 15;	Tanggal pemberian pupuk cair minggu ke-3.
int hama1 = 1;	Tanggal penyemprotan pestisida minggu ke-1.
int hama2 = 8;	Tanggal penyemprotan pestisida minggu ke-2.
int hama3 = 15;	Tanggal penyemprotan pestisida minggu ke-3.
int hama4 = 22;	Tanggal penyemprotan pestisida minggu ke-4.
int waktupump1 = 60;	Durasi penyiraman pompa 1 dalam detik.
int waktupump2 = 60;	Durasi penyiraman pompa 2 dalam detik.
int waktupump3 = 60;	Durasi penyiraman pompa 3 dalam detik.
int hama4 = 22;	Tanggal penyemprotan pestisida minggu ke-4.
int habis1 = 5;	Batas tinggi air pada tangki 1 yang menunjukkan bahwa air telah habis.
int habis2 = 5;	Batas tinggi air pada tangki 2 yang menunjukkan bahwa air telah habis.
int habis3 = 5;	Batas tinggi air pada tangki 3 yang menunjukkan bahwa air telah habis.
<pre> BLYNK_WRITE(V4){ int pinValue = param.asInt(); if(pinValue==1) digitalWrite(pump1, 0); if(pinValue==0) digitalWrite(pump1, 1); Serial.print("pump1:"); Serial.println(pinValue); } </pre>	Fungsi yang dieksekusi ketika widget <i>Blynk</i> dengan pin virtual V4 diubah nilainya. Berfungsi untuk mengontrol pompa 1 berdasarkan input dari <i>Blynk</i> .

Pengkodean	Penjelasan
<pre> BLYNK_WRITE(V5){ int pinValue = param.asInt(); if(pinValue==1) digitalWrite(pump2, 0); if(pinValue==0) digitalWrite(pump2, 1); Serial.print("pump2:"); Serial.println(pinValue); } BLYNK_WRITE(V6){ int pinValue = param.asInt(); if(pinValue==1) digitalWrite(pump3, 0); //on if(pinValue==0) digitalWrite(pump3, 1); //off Serial.print("pump3:"); Serial.println(pinValue); } </pre>	<p>Fungsi yang dieksekusi ketika widget <i>Blynk</i> dengan pin virtual V5 diubah nilainya. Berfungsi untuk mengontrol pompa 2 berdasarkan input dari <i>Blynk</i>.</p> <p>Fungsi yang dieksekusi ketika widget <i>Blynk</i> dengan pin virtual V6 diubah nilainya. Berfungsi untuk mengontrol pompa 3 berdasarkan input dari <i>Blynk</i>.</p>
<pre> void display_lcd() count1++; if (count1 >=5){count1=0;count2++;} if(count2>1)count2=0; lcd.clear(); if(count2==0){ ... } lcd.setCursor(0,0); lcd.print(waktu); lcd.setCursor(11,0); lcd.print(digitalRead(pump1)); lcd.print(" "); lcd.print(digitalRead(pump2)); lcd.print(" "); lcd.print(digitalRead(pump3)); lcd.setCursor(0,1); </pre>	<p>Mendefinisikan fungsi <code>display_lcd()</code>. Menaikkan nilai variabel <code>count1</code> sebanyak 1. Jika nilai variabel <code>count1</code> mencapai 5 atau lebih, maka <code>count1</code> diatur kembali menjadi 0 dan <code>count2</code> ditambah 1. Jika nilai variabel <code>count2</code> lebih dari 1, maka <code>count2</code> diatur kembali menjadi 0. Menghapus semua konten yang ditampilkan di LCD. Jika nilai variabel <code>count2</code> adalah 0, maka jalankan blok kode berikutnya. Mengatur posisi kursor LCD ke baris 0, kolom 0. Menampilkan nilai dari variabel <code>waktu</code> pada LCD. Mengatur posisi kursor LCD ke baris 0, kolom 11. Menampilkan nilai bacaan digital dari pin <code>pump1</code> pada LCD. Menampilkan spasi pada LCD. Menampilkan nilai bacaan digital dari pin <code>pump2</code> pada LCD. Menampilkan spasi pada LCD. Menampilkan nilai bacaan digital dari pin <code>pump3</code> pada LCD. Mengatur posisi kursor LCD ke baris 1, kolom 0.</p>

Pengkodean	Penjelasan
lcd.print(tanggal);	Menampilkan nilai dari variabel tanggal pada LCD.
lcd.setCursor(11,1);	Mengatur posisi kursor LCD ke baris 1, kolom 11.
lcd.print("K:"); lcd.print(kelembaban);	Menampilkan teks "K:" pada LCD. Menampilkan nilai dari variabel kelembaban pada LCD.
if(count2==1){ ... }	Jika nilai variabel count2 adalah 1, maka jalankan blok kode berikutnya.
lcd.setCursor(0,0);	Mengatur posisi kursor LCD ke baris 0, kolom 0.
lcd.print("S:"); lcd.print(suhu,1);	Menampilkan teks "S:" pada LCD. Menampilkan nilai dari variabel suhu dengan 1 digit desimal pada LCD.
lcd.setCursor(9,0);	Mengatur posisi kursor LCD ke baris 0, kolom 9.
lcd.print("A:"); lcd.print(tinggi1,1);	Menampilkan teks "A:" pada LCD. Menampilkan nilai dari variabel tinggi1 dengan 1 digit desimal pada LCD.
lcd.setCursor(0,1);	Mengatur posisi kursor LCD ke baris 1, kolom 0.
lcd.print("B:"); lcd.print(tinggi2,1);	Menampilkan teks "B:" pada LCD. Menampilkan nilai dari variabel tinggi2 dengan 1 digit desimal pada LCD.
lcd.setCursor(9,1);	Mengatur posisi kursor LCD ke baris 1, kolom 9.
lcd.print("C:"); lcd.print(tinggi3,1);	Menampilkan teks "C:" pada LCD. Menampilkan nilai dari variabel tinggi3 dengan 1 digit desimal pada LCD.
float read_srf(int i)	Fungsi untuk membaca jarak dari sensor <i>Ultrasonic</i> dengan parameter i sebagai indeks pin.
unsigned int duration;	Mendeklarasikan variabel duration sebagai tipe data unsigned int untuk menyimpan durasi pulsa <i>Ultrasonic</i> yang akan diukur.
float distance;	Mendeklarasikan variabel distance sebagai tipe data float untuk menyimpan jarak yang akan dihitung.
digitalWrite(pin_trig[i], LOW);	Mengirim sinyal LOW ke pin trigger (pin_trig) yang sesuai dengan nilai i. Ini mematikan trigger sinyal <i>Ultrasonic</i> .
delayMicroseconds(5);	Menunda eksekusi selama 5 mikrodetik (us) untuk memberikan jeda sebelum mengaktifkan kembali trigger sinyal <i>Ultrasonic</i> .
digitalWrite(pin_trig[i], HIGH);	Mengirim sinyal HIGH ke pin trigger (pin_trig) yang sesuai dengan nilai i. Ini

Pengkodean	Penjelasan
delayMicroseconds(10);	mengaktifkan kembali trigger sinyal <i>Ultrasonic</i> . Menunda eksekusi selama 10 mikrodetik (us) untuk memberikan jeda sebelum mematikan kembali trigger sinyal <i>Ultrasonic</i> .
digitalWrite(pin_trig[i], LOW);	Mengirim sinyal LOW ke pin trigger (pin_trig) yang sesuai dengan nilai i. Ini mematikan kembali trigger sinyal <i>Ultrasonic</i> .
duration = pulseIn(pin_echo[i], HIGH);	Mengukur durasi pulsa yang diterima pada pin echo (pin_echo) yang sesuai dengan nilai i dan menyimpannya ke dalam variabel duration. Durasi pulsa ini akan digunakan untuk menghitung jarak.
distance = duration * 0.034 / 2;	Menghitung jarak berdasarkan durasi pulsa yang diperoleh. Menggunakan faktor konversi 0,034 cm/us (kecepatan suara dalam udara) dan membagi hasilnya dengan 2 karena perjalanan bolak-balik sinyal <i>Ultrasonic</i> .
return distance;	Mengembalikan nilai jarak yang telah dihitung dari fungsi sebagai hasil pengukuran.
void tasktimer(){...}	merupakan fungsi utama yang akan dieksekusi secara berulang oleh <i>BlynkTimer</i> .
digitalWrite(pinled, digitalRead(pinled)^1);	Memflip status LED dengan menggunakan operasi bitwise XOR (XOR) pada nilai yang saat ini dibaca dari pin pinled.
get_time();	Memanggil fungsi get_time() untuk mendapatkan waktu dan tanggal saat ini menggunakan objek timeClient.
kelembaban = map(analogRead(soil), 0, 4096, 0, 100);	Membaca nilai analog dari sensor kelembaban tanah yang terhubung ke pin soil, dan memetakan nilainya ke dalam rentang 0-100 menggunakan fungsi map().
ultra1 = read_srf(0);	Membaca jarak yang terukur dari sensor <i>Ultrasonic</i> pada bak 1 dengan memanggil fungsi read_srf(0).
ultra2 = read_srf(1);	Membaca jarak yang terukur dari sensor <i>Ultrasonic</i> pada bak 2 dengan memanggil fungsi read_srf(1).
ultra3 = read_srf(2);	Membaca jarak yang terukur dari sensor <i>Ultrasonic</i> pada bak 3 dengan memanggil fungsi read_srf(2).

Pengkodean	Penjelasan
tinggi1 = setultra1 - ultra1;	Menghitung tinggi air pada bak 1 dengan mengurangi jarak yang terukur dari tinggi yang telah ditetapkan (setultra1).
tinggi2 = setultra2 - ultra2;	Menghitung tinggi air pada bak 2 dengan mengurangi jarak yang terukur dari tinggi yang telah ditetapkan (setultra2).
tinggi3 = setultra3 - ultra3;	Menghitung tinggi air pada bak 3 dengan mengurangi jarak yang terukur dari tinggi yang telah ditetapkan (setultra3).
if (tinggi1 < 0) tinggi1 = 0;	Memastikan bahwa tinggi air pada bak 1 tidak menjadi negatif dengan mengaturnya menjadi 0 jika kurang dari 0.
if (tinggi2 < 0) tinggi2 = 0;	Memastikan bahwa tinggi air pada bak 2 tidak menjadi negatif dengan mengaturnya menjadi 0 jika kurang dari 0.
if (tinggi3 < 0) tinggi3 = 0;	Memastikan bahwa tinggi air pada bak 3 tidak menjadi negatif dengan mengaturnya menjadi 0 jika kurang dari 0.
sensor.requestTemperatures();	Meminta sensor suhu untuk membaca nilai suhu saat ini dengan memanggil fungsi requestTemperatures() pada objek sensor.
suhu = sensor.getTempCByIndex(0);	Membaca suhu saat ini dari sensor dan menyimpannya di variabel suhu.
<pre> if(suhu > setsuhu){ if(lock1!=1){ lock1=1; digitalWrite(pump1, 0); Blynk.logEvent("notif","Suhu melebihi batas, sistem melakukan penyiraman otomatis"); } } </pre>	<p>Memeriksa apakah suhu saat ini melebihi suhu yang telah ditetapkan (setsuhu). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i> dan melakukan penyiraman air.</p>
<pre> if(suhu < setsuhu){ if(lock1!=0){ lock1=0; digitalWrite(pump1, 1); } } </pre>	<p>Memeriksa apakah suhu saat ini kurang dari suhu yang telah ditetapkan (setsuhu). Jika ya, maka sistem mematikan pompa (memberhentikan penyiraman).</p>
<pre> if(kelembaban < setkelembaban){ if(lock2!=1){ lock2=1; digitalWrite(pump1, 0); </pre>	<p>Memeriksa apakah kelembaban tanah saat ini kurang dari kelembaban yang telah ditetapkan (setkelembaban). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i> dan melakukan penyiraman air.</p>
<pre> Blynk.logEvent("notif","Kelembaban kurang dari batas, sistem melakukan penyiraman otomatis"); } </pre>	

Pengkodean	Penjelasan
<pre> } if(kelembaban > setkelembaban){ if(lock2!=0){ lock2=0; digitalWrite(pump1, 1); } } </pre>	<p>Memeriksa apakah kelembaban tanah saat ini melebihi kelembaban yang telah ditetapkan (setkelembaban). Jika ya, maka sistem mematikan pompa (memberhentikan penyiraman).</p>
<pre> if(jam == air1){ if(lock3!=1){ lock3=1; tpump1=0; digitalWrite(pump1, 0); Blynk.logEvent("notif","Penyiraman otomatis pagi"); } </pre>	<p>Memeriksa apakah jam saat ini sama dengan waktu penyiraman air pagi (air1). Jika ya, maka sistem melakukan penyiraman selama waktu yang sudah ditentukan dan mengirimkan notif ke aplikasi <i>Blynk</i>.</p>
<pre> if(tpump1>=waktupump1) { if(lock3!=2){ lock3=2; digitalWrite(pump1, 1); } } </pre>	
<pre> else { tpump1++;} if(jam == air2){ if(lock3!=3){ lock3=3; tpump1=0; digitalWrite(pump1, 0); Blynk.logEvent("notif","Penyiraman otomatis sore"); } </pre>	<p>Memeriksa apakah jam saat ini sama dengan waktu penyiraman air sore (air2). Jika ya, maka sistem melakukan penyiraman selama waktu yang sudah ditentukan dan mengirimkan notif ke aplikasi <i>Blynk</i>.</p>
<pre> if(tpump1>=waktupump1) { if(lock3!=4){ lock3=4; digitalWrite(pump1, 1); } } else { tpump1++;} } </pre>	
<pre> if(monthDay == pupukcair1 && jam==jampupuk){ if(lock4!=1){ lock4=1; tpump2=0; digitalWrite(pump2, 0); </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyiraman pupuk cair minggu ke 1 dan waktu penyiramannya. Jika ya, maka sistem melakukan penyiraman pupuk cair selama waktu</p>

Pengkodean	Penjelasan
<pre> Blynk.logEvent("notif","Penyiraman pupuk cair minggu ke 1"); } if(tpump2>=waktupump2) { if(lock4!=2){ lock4=2; digitalWrite(pump2, 1); } } else {tpump2++;} } </pre>	<p>yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(monthDay == pupukcair2 && jam==jampupuk){ if(lock4!=3){ lock4=3; tpump2=0; digitalWrite(pump2, 0); Blynk.logEvent("notif","Penyiraman pupuk cair minggu ke 3"); } } </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyiraman pupuk cair minggu ke 3 dan waktu penyiramannya. Jika ya, maka sistem melakukan penyiraman pupuk cair selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tpump2>=waktupump2) { if(lock4!=4){ lock4=4; digitalWrite(pump2, 1); } } else {tpump2++;} } </pre>	
<pre> if(monthDay == hama1 && jam==jamhama){ if(lock5!=1){ lock5=1; tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif","Penyemprotan pestisida minggu ke 1"); } } </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 1 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tpump3>=waktupump3) { if(lock5!=2){ lock5=2; digitalWrite(pump3, 1); } } </pre>	

Pengkodean	Penjelasan
<pre> } } else {tpump3++;} } if(monthDay == hama2 && jam==jamhama){ if(lock5!=3){ lock5=3; tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif", "Penyemprotan pestisida minggu ke 2"); } </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 2 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tpump3>=waktupump3) { if(lock5!=4){ lock5=4; digitalWrite(pump3, 1); } } else {tpump3++;} } if(monthDay == hama3 && jam==jamhama){ if(lock5!=5){ lock5=5; tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif", "Penyemprotan pestisida minggu ke 3"); } </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 3 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tpump3>=waktupump3) { if(lock5!=6){ lock5=6; digitalWrite(pump3, 1); } } else {tpump3++;} } if(monthDay == hama4 && jam==jamhama){ if(lock5!=7){ lock5=7; </pre>	<p>Memeriksa apakah tanggal saat ini sama dengan tanggal penyemprotan pestisida minggu ke 4 dan waktu penyiraman yang sudah ditentukan. Jika ya, maka sistem</p>

Pengkodean	Penjelasan
<pre> tpump3=0; digitalWrite(pump3, 0); Blynk.logEvent("notif","Penyemprotan pestisida minggu ke 4"); } </pre>	<p>melakukan penyiraman pestisida selama waktu yang telah ditentukan serta mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tpump3>=waktupump3) { if(lock5!=8){ lock5=8; digitalWrite(pump3, 1); } } else {tpump3++;} } </pre>	
<pre> if(tinggi1 < habis1){ if(lock6!=1){ lock6=1; Blynk.logEvent("notif","Air habis, isi bak segera!"); } } </pre>	<p>Memeriksa apakah tinggi air pada bak 1 lebih rendah dari batas habis (habis1). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tinggi1 > habis1){ lock6=0; } </pre>	<p>Memeriksa apakah tinggi air pada bak 1 lebih tinggi dari batas habis (habis1). Jika ya, maka sistem memberhentikan notifikasi.</p>
<pre> if(tinggi2 < habis2){ if(lock7!=1){ lock7=1; Blynk.logEvent("notif","Pupuk habis, isi bak segera!"); } } </pre>	<p>Memeriksa apakah tinggi air pada bak 2 lebih rendah dari batas habis (habis2). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tinggi2 > habis2){ lock7=0; } </pre>	<p>Memeriksa apakah tinggi air pada bak 2 lebih tinggi dari batas habis (habis2). Jika ya, maka sistem memberhentikan notifikasi.</p>
<pre> if(tinggi3 < habis3){ if(lock8!=1){ lock8=1; Blynk.logEvent("notif","Pestisida habis, isi bak segera!"); } } </pre>	<p>Memeriksa apakah tinggi air pada bak 3 lebih rendah dari batas habis (habis3). Jika ya, maka sistem mengirimkan notifikasi ke <i>Blynk</i>.</p>
<pre> if(tinggi3 > habis3){ lock8=0; } </pre>	<p>Memeriksa apakah tinggi air pada bak 3 lebih tinggi dari batas habis (habis3). Jika</p>

Pengkodean	Penjelasan
<i>Blynk.virtualWrite</i> (V0, suhu);	ya, maka sistem memberhentikan notifikasi.
<i>Blynk.virtualWrite</i> (V1, tinggi1);	Mengirim nilai suhu ke pin virtual V0 pada aplikasi <i>Blynk</i> .
<i>Blynk.virtualWrite</i> (V2, tinggi2);	Mengirim nilai tinggi air pada bak 1 ke pin virtual V1 pada aplikasi <i>Blynk</i> .
<i>Blynk.virtualWrite</i> (V3, tinggi3);	Mengirim nilai tinggi air pada bak 2 ke pin virtual V2 pada aplikasi <i>Blynk</i> .
<i>Blynk.virtualWrite</i> (V7, kelembaban);	Mengirim nilai tinggi air pada bak 3 ke pin virtual V3 pada aplikasi <i>Blynk</i> .
<i>Serial.print</i> (" S: ");	Mengirim nilai kelembaban tanah ke pin virtual V7 pada aplikasi <i>Blynk</i> .
<i>Serial.print</i> (suhu);	Mencetak teks " S: " di Serial Monitor.
<i>Serial.print</i> (" A: ");	Mencetak nilai suhu saat ini di Serial Monitor.
<i>Serial.print</i> (ultra1);	Mencetak teks " A: " di Serial Monitor.
<i>Serial.print</i> (" B: ");	Mencetak nilai jarak dari sensor <i>Ultrasonic</i> pada bak 1 di Serial Monitor.
<i>Serial.print</i> (ultra2);	Mencetak teks " B: " di Serial Monitor.
<i>Serial.print</i> (" C: ");	Mencetak nilai jarak dari sensor <i>Ultrasonic</i> pada bak 2 di Serial Monitor.
<i>Serial.print</i> (ultra3);	Mencetak teks " C: " di Serial Monitor.
<i>Serial.print</i> (" K: ");	Mencetak nilai jarak dari sensor <i>Ultrasonic</i> pada bak 3 di Serial Monitor.
<i>Serial.print</i> (kelembaban);	Mencetak teks " K: " di Serial Monitor.
String text = "";	Mencetak nilai kelembaban tanah di Serial Monitor.
	Membuat variabel text dengan tipe data String dan menginisialisasinya dengan nilai kosong.
text = " pump1:" +	Menggabungkan teks dan status pompa air pada variabel text.
String(digitalRead(pump1)) + " pump2:"	
+ String(digitalRead(pump2)) + "	
pump3:" + String(digitalRead(pump3));	
<i>Blynk.virtualWrite</i> (V8, text);	Mengirim nilai dari variabel text ke pin virtual V8 pada aplikasi <i>Blynk</i> .
display_lcd();	Memanggil fungsi <i>display_lcd()</i> untuk menampilkan informasi di layar LCD.
<i>Serial.println</i> ();	Mencetak baris kosong di Serial Monitor.
void setup(){...}	Melakukan inisialisasi/konfigurasi input dan output saat ESP32 pertama kali menyala.
<i>Serial.begin</i> (9600);	Memulai komunikasi serial dengan baud rate 9600.
<i>Serial.println</i> (" starting ");	Mencetak teks " starting " di Serial Monitor.
<i>pinMode</i> (pinled, OUTPUT);	Mengatur pin pinled sebagai output.
<i>pinMode</i> (pump1, OUTPUT);	Mengatur pin pump1 sebagai output.

Pengkodean	Penjelasan
<pre>pinMode(pump2, OUTPUT); pinMode(pump3, OUTPUT); for(int i=0; i<3; i++) { ... }</pre>	<p>Mengatur pin pump2 sebagai output. Mengatur pin pump3 sebagai output. Melakukan iterasi tiga kali untuk mengatur pin pin_trig dan pin_echo sebagai output dan input, sesuai dengan konfigurasi sensor <i>Ultrasonic</i>.</p>
<pre>digitalWrite(pump1, 1);</pre>	<p>Mematikan pompa air pada pump1 dengan mengatur pin pump1 menjadi HIGH.</p>
<pre>digitalWrite(pump2, 1);</pre>	<p>Mematikan pompa air pada pump2 dengan mengatur pin pump2 menjadi HIGH.</p>
<pre>digitalWrite(pump3, 1);</pre>	<p>Mematikan pompa air pada pump3 dengan mengatur pin pump3 menjadi HIGH.</p>
<pre>lcd.begin(); lcd.backlight(); lcd.clear();</pre>	<p>Menginisialisasi layar LCD. Mengaktifkan backlight pada layar LCD. Menghapus konten yang ditampilkan di layar LCD.</p>
<pre>lcd.setCursor(0, 0);</pre>	<p>Menetapkan posisi kursor LCD ke baris 0, kolom 0.</p>
<pre>lcd.print("ssid:"); lcd.print(ssid);</pre>	<p>Menampilkan teks "ssid:" pada LCD. Menampilkan nilai dari variabel ssid pada LCD.</p>
<pre>lcd.setCursor(0, 1);</pre>	<p>Menetapkan posisi kursor LCD ke baris 1, kolom 0.</p>
<pre>lcd.print("pass:"); lcd.print(pass);</pre>	<p>Menampilkan teks "pass:" pada LCD. Menampilkan nilai dari variabel pass pada LCD.</p>
<pre><i>Blynk</i>.begin(auth, ssid, pass);</pre>	<p>Memulai koneksi dengan server <i>Blynk</i> menggunakan autentikasi dan SSID serta password WiFi yang telah diberikan.</p>
<pre>timers.setInterval(1000L, tasktimer);</pre>	<p>Menetapkan interval 1000ms (1 detik) untuk menjalankan fungsi tasktimer() menggunakan timer.</p>
<pre>Serial.println(" ready "); timeClient.begin(); timers.run();</pre>	<p>Mencetak teks " ready " di Serial Monitor. Memulai waktu klien NTP. Menjalankan timer untuk fungsi yang telah diatur menggunakan timers.setInterval().</p>
<pre><i>Blynk</i>.run();</pre>	<p>Menjalankan koneksi <i>Blynk</i>.</p>

4.5.11 Pembangunan Tata Letak Sistem

4.5.11.1 Desain Bangun Ruang *Prototype*

Desain bangun ruang digunakan untuk merencanakan dan memberikan gambaran mengenai tata letak sistem yang dibangun. Sistem yang dibuat pada penelitian ini merupakan *prototype* skala kecil dengan skala 1:60. Pada penelitian ini, pembuatan desain 3D bangun ruang *prototype* sistem *monitoring* dan *controlling* kebun organik dilakukan menggunakan *software Autodesk Fusion 360*. Berikut merupakan hasil render dari desain 3D *prototype* pada gambar 4.48.

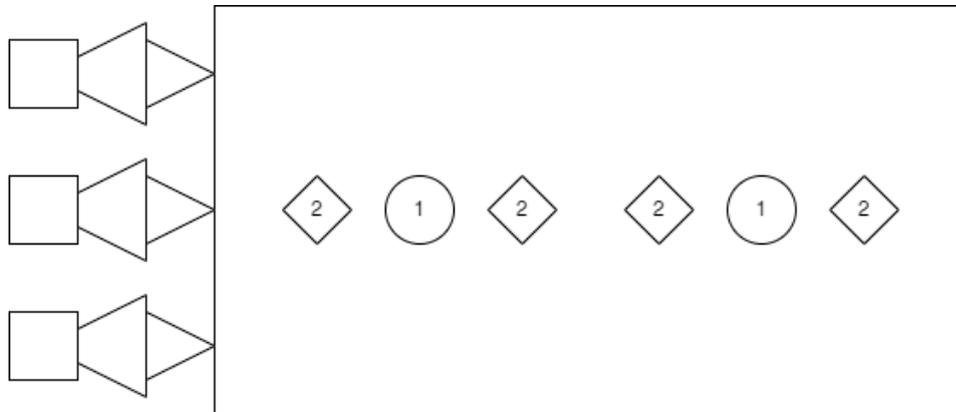


Gambar 4. 48 Desain 3D *Prototype*

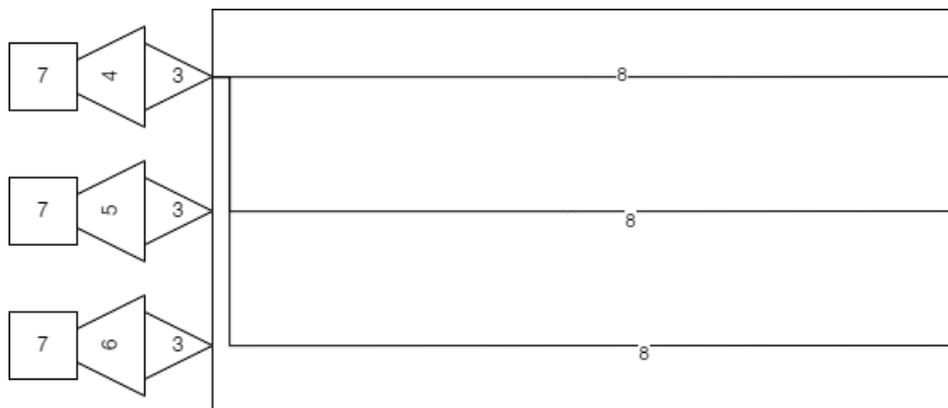
4.5.11.2 Gambar Teknik Bangun Ruang *Prototype*

Pada penelitian ini dilakukan pembuatan gambar teknik *prototype* sistem *monitoring* dan *controlling* yang berfungsi sebagai acuan dalam proses pembangunan *prototype* secara nyata. Berikut merupakan gambar teknik *prototype* dengan skala 1:5 menggunakan satuan mm (millimeter) pada gambar 4.49.

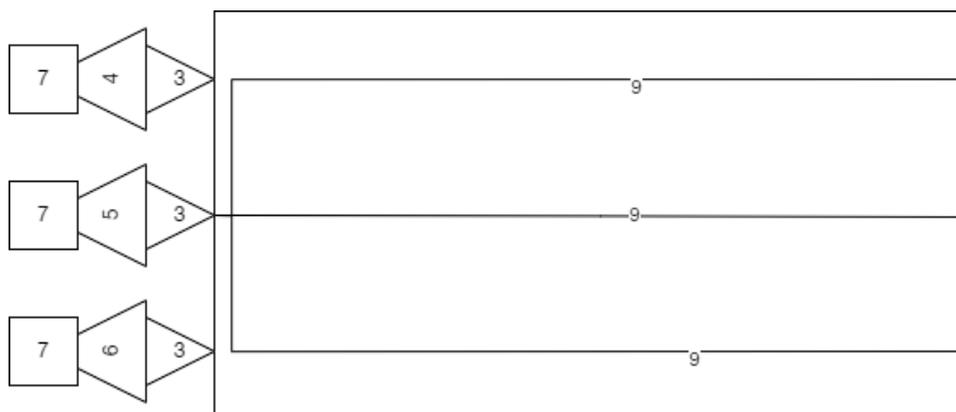
Setelah peneliti melakukan analisis Lingkungan Kerja Fisik, berikut merupakan proyeksi peletakkan sistem pada gambar 4.51, 4.52, 4.53, dan 4.54.



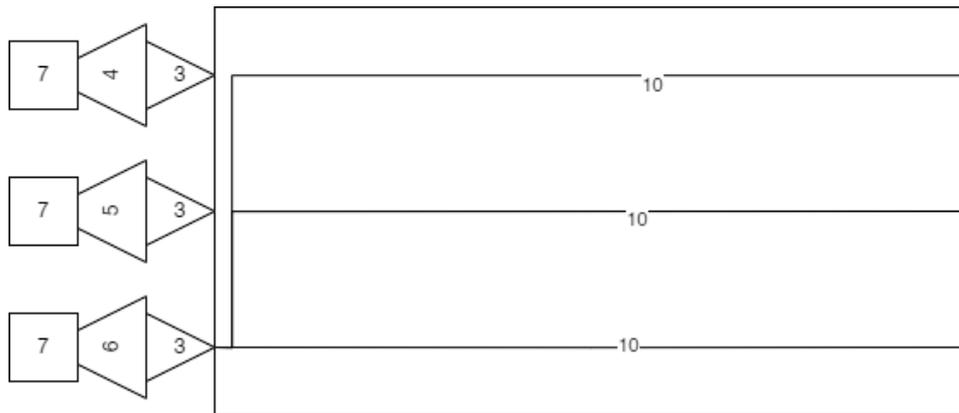
Gambar 4. 51 Proyeksi Pertama



Gambar 4. 52 Proyeksi Kedua



Gambar 4. 53 Proyeksi Ketiga



Gambar 4. 54 Proyeksi Keempat

Selanjutnya, berikut merupakan keterangan proyeksi peletakkan sistem yang disajikan pada tabel 4.38

Tabel 4. 38 Keterangan Proyeksi Peletakkan Sistem

Nomor	Nama Komponen
1	Sensor Suhu Tanah
2	Sensor Kelembaban Tanah
3	Pompa Air
4	Bak Penampungan Air
5	Bak Penampungan Pupuk Cair
6	Bak Penampungan Pestisida
7	Sensor <i>Ultrasonic</i>
8	Saluran/Selang Air
9	Saluran/Selang Pupuk Cair

Nomor	Nama Komponen
10	Saluran/Selang Pesticida

4.6 Usability Testing

Dilakukan *Usability Testing* pada sistem yang telah dibangun dengan pendekatan *System Usability Scale (SUS)*. *Usability Testing* dilakukan untuk mengukur tingkat *usability* dari sistem yang telah dibangun. Pengujian dilakukan dengan memberikan kuesioner kepada para pengelola kebun Nara Kupu Yogyakarta Berikut merupakan gambar dari kegiatan pengambilan data untuk *Usability Testing* pada kebun organik Nara Kupu Yogyakarta.



Gambar 4. 55 Pengisian Kuesioner *System Usability Scale (SUS)* oleh pengelola kebun Nara Kupu Yogyakarta

Setelah melakukan pengisian kuesioner oleh pengelola kebun organik Nara Kupu Yogyakarta, Berikut merupakan hasil yang didapat.

Tabel 4. 39 Hasil Kuesioner Responden Pertama

No.	Pertanyaan	1	2	3	4	5
1	Saya akan senang untuk menggunakan sistem ini secara sering.					√
2	Saya merasa sistem ini rumit untuk digunakan.		√			
3	Saya merasa sistem ini mudah digunakan.					√
4	Saya membutuhkan bantuan dari orang lain dalam Menggunakan sistem ini.		√			

No.	Pertanyaan	1	2	3	4	5
5	Saya merasa fitur-fitur sistem ini diintegrasikan dengan baik.				√	
6	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.			√		
7	Saya merasa orang lain akan memahami sistem ini dengan cepat.				√	
8	Saya merasa sistem ini membingungkan.		√			
9	Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.			√		
10	Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.		√			

Skala:

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju

Selanjutnya, berikut merupakan hasil kuesioner responden kedua.

Tabel 4. 40 Hasil Kuesioner Responden kedua

No.	Pertanyaan	1	2	3	4	5
1	Saya akan senang untuk menggunakan sistem ini secara sering.					√
2	Saya merasa sistem ini rumit untuk digunakan.			√		
3	Saya merasa sistem ini mudah digunakan.				√	
4	Saya membutuhkan bantuan dari orang lain dalam Menggunakan sistem ini.		√			
5	Saya merasa fitur-fitur sistem ini diintegrasikan dengan baik.				√	
6	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.			√		
7	Saya merasa orang lain akan memahami sistem ini dengan cepat.					√
8	Saya merasa sistem ini membingungkan.		√			
9	Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.			√		
10	Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.		√			

Skala:

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju

Selanjutnya, berikut merupakan hasil kuesioner responden ketiga.

Tabel 4. 41 Hasil Kuesioner Responden Ketiga

No.	Pertanyaan	1	2	3	4	5
1	Saya akan senang untuk menggunakan sistem ini secara sering.			√		
2	Saya merasa sistem ini rumit untuk digunakan.	√				
3	Saya merasa sistem ini mudah digunakan.				√	
4	Saya membutuhkan bantuan dari orang lain dalam Menggunakan sistem ini.		√			
5	Saya merasa fitur-fitur sistem ini diintegrasikan dengan baik.				√	
6	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.	√				
7	Saya merasa orang lain akan memahami sistem ini dengan cepat.				√	
8	Saya merasa sistem ini membingungkan.	√				
9	Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.				√	
10	Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.				√	

Skala:

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju

Selanjutnya, berikut merupakan hasil kuesioner responden keempat.

Tabel 4. 42 Hasil Kuesioner Responden keempat

No.	Pertanyaan	1	2	3	4	5
1	Saya akan senang untuk menggunakan sistem ini secara sering.				√	
2	Saya merasa sistem ini rumit untuk digunakan.		√			
3	Saya merasa sistem ini mudah digunakan.			√		

No.	Pertanyaan	1	2	3	4	5
4	Saya membutuhkan bantuan dari orang lain dalam Menggunakan sistem ini.	√				
5	Saya merasa fitur-fitur sistem ini diintegrasikan dengan baik.					√
6	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.		√			
7	Saya merasa orang lain akan memahami sistem ini dengan cepat.					√
8	Saya merasa sistem ini membingungkan.		√			
9	Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.			√		
10	Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.	√				

Skala:

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju

Dari hasil kuesioner-kuesioner tersebut, didapatkan hasil skor *Usability Testing* yang dihitung menggunakan rumus *System Usability Scale* (SUS). Berikut merupakan hasilnya.

1. Hasil skor responden pertama:

$$((4 - 1) + (4 - 1) + (4 - 1) + (4 - 1) + (3 - 1) + (5 - 2) + (5 - 2) + (5 - 3) + (5 - 2) + (5 - 2)) \times 2,5 = 70 \dots (3)$$

2. Hasil skor responden kedua:

$$((5 - 1) + (4 - 1) + (4 - 1) + (5 - 1) + (3 - 1) + (5 - 3) + (5 - 2) + (5 - 3) + (5 - 2) + (5 - 2)) \times 2,5 = 72,5 \dots (4)$$

3. Hasil skor responden ketiga:

$$((3 - 1) + (4 - 1) + (4 - 1) + (4 - 1) + (4 - 1) + (5 - 1) + (5 - 2) + (5 - 1) + (5 - 1) + (5 - 4)) \times 2,5 = 75 \dots (5)$$

4. Hasil skor responden keempat:

$$((4 - 1) + (3 - 1) + (5 - 1) + (5 - 1) + (3 - 1) + (5 - 2) + (5 - 1) + (5 - 2) + (5 - 2) + (5 - 1)) \times 2,5 = 80 \dots (6)$$

Selanjutnya, dilakukan perhitungan rata-rata skor *System Usability Scale* (SUS). Berikut merupakan hasilnya.

$$\tilde{x} = \frac{70 + 72,5 + 75 + 80}{4} = 74,375 \dots (7)$$

Maka, hasil skor *Usability Testing* dengan *System Usability Scale* (SUS) yang didapatkan adalah 74,375.

BAB V

PENGUJIAN SISTEM DAN PEMBAHASAN

5.1 Pengujian sistem

Pengujian sistem bertujuan untuk mengevaluasi sistem *prototype* yang sudah dibangun serta untuk memastikan bahwa sistem dapat berjalan dengan sesuai di kondisi nyata. Berikut merupakan tahapan pengujian sistem pada penelitian ini.

5.1.1 *Usability Testing* dengan *System Usability Scale* (SUS)

Pengujian *Usability Testing* digunakan oleh peneliti untuk mengukur tingkat *usability* dari sistem yang dibangun, serta membuktikan sistem yang dibangun memiliki tingkat *usability* yang tinggi dan efisien. *Usability Testing* dilakukan dengan menggunakan pendekatan *System Usability Scale* (SUS), pembahasan mengenai hasil *System Usability Scale* (SUS) dimulai dengan analisis rata-rata setiap jawaban responden pada kuesioner yang telah diberikan. Berikut merupakan analisis tersebut yang disajikan dalam bentuk tabel.

Tabel 5. 1 Analisis rata-rata jawaban responden

Pertanyaan	Rata-rata Jawaban	Analisis
Pertanyaan 1: Saya senang menggunakan sistem ini.	4.0	Mayoritas responden menunjukkan ketertarikan untuk menggunakan sistem ini secara rutin, ini mengindikasikan bahwa sistem menarik minat pengguna untuk berinteraksi secara berulang dengan sistem yang telah dibangun.
Pertanyaan 2: Saya merasa sistem ini rumit untuk digunakan.	2.0	Responden merasa bahwa sistem ini tidak terlalu rumit, mengindikasikan bahwa antarmuka sistem (<i>User Interface</i>) telah dirancang dengan baik dan pengguna tidak mengalami kesulitan dalam berinteraksi dengan sistem.
Pertanyaan 3: Saya merasa sistem ini mudah untuk digunakan.	3.75	Responden merasa bahwa sistem ini cenderung mudah untuk digunakan, namun masih terdapat keraguan yang berarti bahwa ada ruang untuk peningkatan dalam desain pengalaman pengguna (<i>User Experience</i>).
Pertanyaan 4: Saya membutuhkan bantuan dari orang lain dalam menggunakan sistem ini.	1.75	Rata-rata rendah menunjukkan bahwa mayoritas responden merasa mampu untuk menggunakan sistem tanpa bantuan eksternal. Hal ini menandakan bahwa sistem telah dirancang dengan baik untuk kemandirian pengguna.

Pertanyaan	Rata-rata Jawaban	Analisis
Pertanyaan 5: Saya merasa fitur-fitur sistem ini diintegrasikan dengan baik	4.25	Responden merasa fitur-fitur pada sistem telah diintegrasikan dengan baik, ini menunjukkan bahwa sistem mampu memberikan pengalaman yang konsisten bagi pengguna.
Pertanyaan 6: Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.	2.25	Responden merasa ada sedikit ketidak konsistenan pada sistem. Hal ini dapat menunjukkan untuk melakukan perluasan konsistensi dalam desain sistem untuk mengurangi rasa kebingungan.
Pertanyaan 7: Saya merasa orang lain akan memahami sistem ini dengan cepat.	4.5	Mayoritas responden percaya bahwa sistem ini mudah dipahami oleh pengguna lain, Ini menunjukkan bahwa sistem dapat diterima oleh berbagai pengguna.
Pertanyaan 8: Saya merasa sistem ini membingungkan.	1.75	Rata-rata rendah mengindikasikan bahwa mayoritas responden merasa sistem ini tidak membingungkan. Ini menandakan kegunaan sistem memberikan petunjuk yang jelas kepada pengguna.
Pertanyaan 9: Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.	3.25	Rata-rata menunjukkan bahwa responden merasa cukup percaya diri dalam menggunakan sistem. Namun, terdapat ruang untuk meningkatkan rasa percaya diri pengguna.
Pertanyaan 10: Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.	2.25	Responden merasa bahwa mereka tidak perlu banyak mempelajari sebelum menggunakan sistem. Ini menunjukkan bahwa sistem ini cukup mudah dipahami dan diakses oleh pengguna.

Selanjutnya, didapatkan beberapa pertanyaan dengan skor terbaik dan terburuk pada pertanyaan bernomor ganjil dan genap. Berikut merupakan analisis lanjutan mengenai pertanyaan-pertanyaan tersebut yang disajikan dalam bentuk tabel.

Tabel 5. 2 Pertanyaan dengan skor terbaik

Pertanyaan dengan Skor Terbaik	Rata-rata Jawaban	Analisis
Pertanyaan 4: Saya membutuhkan bantuan dari orang lain dalam menggunakan sistem ini.	1.75	Analisis skor rendah menunjukkan bahwa pengguna merasa cukup mampu menggunakan sistem ini secara mandiri tanpa bantuan dari orang lain. Ini mengindikasikan bahwa sistem memiliki <i>user interface</i> yang jelas dan <i>user friendly</i> .
Pertanyaan 7: Saya merasa orang lain akan memahami sistem ini dengan cepat.	4.5	Analisis skor tinggi mengindikasikan bahwa mayoritas pengguna merasa bahwa sistem ini mudah dimengerti oleh pengguna lain. Hal ini menunjukkan bahwa <i>user</i>

Pertanyaan dengan Skor Terbaik	Rata-rata Jawaban	Analisis
		<i>interface</i> dari sistem yang dibangun dibuat secara jelas dan sederhana sehingga mudah dipahami oleh pengguna lain.

Tabel 5. 3 Pertanyaan dengan skor terburuk

Pertanyaan dengan Skor Terburuk	Rata-rata Jawaban	Analisis
Pertanyaan 6: Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.	2.25	Hasil skor mencerminkan persepsi beberapa pengguna terkait konsistensi sistem. Kemungkinan ada beberapa hal dari sistem yang dapat mengakibatkan kebingungan. Penting untuk mengidentifikasi area yang tidak konsisten dan melakukan perbaikan kedepannya untuk meningkatkan pengalaman pengguna (<i>User experience</i>).
Pertanyaan 9: Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.	3.25	Skor ini menunjukkan bahwa Sebagian pengguna merasa kurang yakin dalam berinteraksi dengan sistem. Ini bisa berkaitan dengan <i>User Interface</i> yang memerlukan peningkatan panduan atau instruksi yang lebih jelas agar pengguna merasa lebih percaya diri dalam menggunakan sistem .

Selain itu, hasil *Usability Testing* yang didapatkan dari perhitungan *System Usability Scale* (SUS) adalah skor akhir dengan nilai sebesar 74,375. Menurut Brooke (2020), jika sistem yang diuji mendapatkan hasil skor *System Usability Scale* (SUS) lebih dari 68 maka sistem yang diuji dapat dikatakan memiliki tingkat *usability* yang baik. Maka dari itu, dapat disimpulkan bahwa sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) yang telah dibangun sudah berfungsi secara efisien dan memiliki tingkat *usability* yang baik.

5.2 Pembahasan Lingkungan Kerja Fisik (LKF)

Pada penelitian ini, metode lingkungan kerja fisik digunakan sebagai alat untuk membantu peneliti dalam membuat proyeksi peletakkan alat dan rekayasa perbandingan hasil panen jika sistem yang dibangun diterapkan secara nyata pada kebun organik Nara Kupu Yogyakarta.

5.2.1 Perbandingan Skala *Prototype*

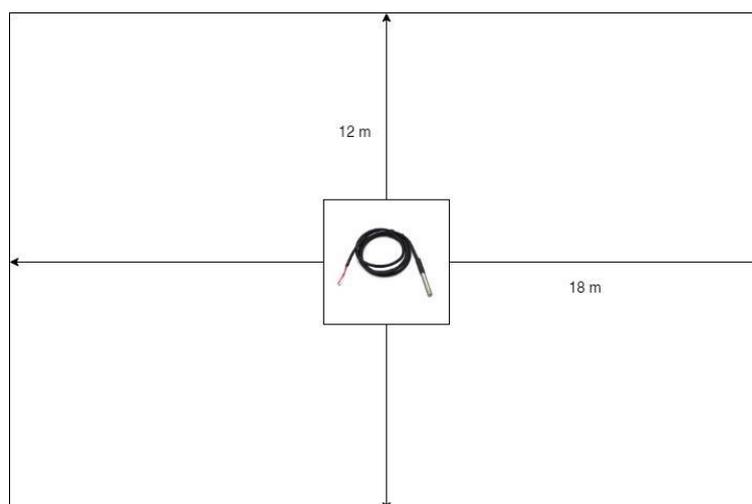
Sistem yang dibuat merupakan *prototype* skala kecil dari kebun organik Nara Kupu Yogyakarta. Sistem ini dibuat dengan skala 1:60, Berikut merupakan rekayasa kebutuhan sistem dan perbandingannya dengan *prototype* yang telah dibangun.

Tabel 5. 4 Perbandingan Skala *Prototype*

Kategori	Sistem yang dibangun	Sistem secara nyata
Luas Area	0.2214m ²	432m ²
Jumlah Sensor Kelembaban Tanah	1	4
Jumlah Sensor Suhu Tanah	1	2
Jumlah Sensor <i>Ultrasonic</i>	3	3
Volume Penampungan Air, Pupuk Cair, dan Pestisida	4.25L (Untuk 1 Tanaman)	240L (Untuk seluruh kebun)
Waktu Habis Penampungan Air, Pupuk Cair, dan Pestisida	±1 Bulan	±1 Bulan

5.2.2 Proyeksi Peletakkan Sensor

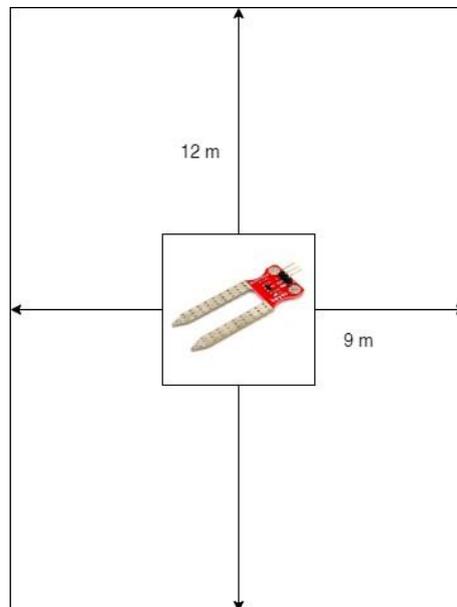
Setelah melakukan pengambilan data suhu dan kelembaban tanah pada kebun Organik Nara Kupu Yogyakarta sebanyak 18 titik, peneliti membuat proyeksi peletakkan sensor suhu dan kelembaban tanah tersebut. Berikut merupakan proyeksi peletakkan sensor suhu dan kelembaban tanah pada gambar 5.1.



Gambar 5. 1 Proyeksi Peletakkan Sensor Suhu Tanah

Pembuatan proyeksi diatas didasari oleh Zhao et al. (2019) yang menyatakan bahwa sensor suhu tanah DS18B20 memiliki jangkauan pembacaan optimal sejauh 20 m dengan

akurasi pembacaan sekitar $\pm 0.5^{\circ}\text{C}$. Selanjutnya, berikut merupakan proyeksi peletakan sensor kelembaban tanah pada gambar 5.2.



Gambar 5. 2 Proyeksi Peletakan Sensor Kelembaban Tanah

Pembuatan proyeksi diatas didasari oleh Kuppler et al. (2021) yang menyatakan bahwa sensor kelembaban tanah ME110 memiliki jangkauan pembacaan optimal sejauh 15 m, namun hal tersebut juga dapat dipengaruhi oleh beberapa variabel lainnya seperti jenis tanah yang digunakan, arah sumber air, dll.

Maka dari itu, berdasarkan dari proyeksi yang telah dibuat, jumlah sensor suhu tanah yang dibutuhkan untuk kebun organik Nara Kupu Yogyakarta dengan ukuran tanah sebesar $12 \times 36\text{m}$ (432m^2) berjumlah 2 dengan masing-masing diletakkan pada petak tanah dengan dimensi $12 \times 18\text{m}$ (218m^2) dan sensor kelembaban tanah yang dibutuhkan berjumlah 4 dengan masing-masing diletakkan pada petak tanah dengan dimensi $12 \times 9\text{m}$ (108m^2). Dengan peletakan tersebut, hasil pembacaan sensor suhu dan kelembaban tanah akan didapatkan secara optimal.

5.2.3 Rekayasa Perbandingan Hasil Panen

Berdasarkan dari pengambilan data Lingkungan Kerja Fisik (LKF), rentang suhu tanah yang didapatkan pada kebun organik adalah $33 - 35^{\circ}\text{C}$ serta kelembaban tanah yang didapatkan adalah $40 - 48\%$. Sedangkan, nilai suhu tanah yang optimal untuk proses pertumbuhan tanaman adalah sekitar 29 hingga 31°C dan untuk kelembaban tanah dibawah 50% Asih (2018). Selain itu, dari hasil wawancara dengan pengelola kebun

organik Nara Kupu Yogyakarta, didapatkan bahwa proses pengelolaan kebun masih tidak konsisten. Kegiatan penyiraman air, pemupukan dan pemberian pestisida masih sering telat/tidak dilakukan karena pengelola kebun mengalami kesibukan. Alhasil, terjadi 15 – 30% kegagalan pada setiap kali panen. Menurut Lysenko et al. (2023), pengelolaan kebun yang tidak konsisten serta kondisi lingkungan kebun yang tidak optimal dapat berkontribusi pada angka kegagalan panen tersebut.

Maka dari itu, peneliti membuat rekayasa perbandingan hasil panen pada kebun organik Nara Kupu Yogyakarta jika diterapkannya sistem *monitoring* dan *controlling* kebun berbasis *Internet of Things* (IoT) secara nyata yang didasarkan oleh penelitian-penelitian terdahulu yang mengangkat topik serupa. Menurut Quy et al. (2022), penerapan *Internet of Things* (IoT) pada bidang agrikultur dapat berkontribusi pada pengurangan tenaga dan biaya untuk mengelola perkebunan serta meningkatkan efisiensi dan produktivitas tanah.

Hal tersebut terealisasi pada penelitian “Rancang Bangun Mesin Pemberi Pupuk Cair Otomatis Hemat Daya Berbasis Iot untuk Budidaya Tanaman Organik” oleh Shofa et al. (2021). Setelah menerapkan alat pemberian pupuk otomatis berbasis *Internet of Things* (IoT) pada kebun, terjadi kenaikan hasil panen tanaman sebesar 20% (dari 70% menjadi 90%). Kenaikan hasil panen tersebut menyebabkan pengurangan pada persentase gagal panen, yang awalnya 30% menjadi 10%. Selain itu, pada penelitian “*An Implementation of Smart Agriculture for Optimizing Growth Using Sonic Bloom and IoT Integrated*” oleh Galina et al. (2022), didapatkan bahwa setelah menerapkan *Internet of Things* (IoT) pada kebun cabai terjadi kenaikan hasil panen sebesar 31.25%. Menurut penulis, variabel-variabel yang mempengaruhi kenaikan persentase hasil panen tersebut merupakan kelembaban dan suhu tanah yang selalu terjaga agar tanaman tumbuh secara optimal.

Setelah mengkaji berbagai penelitian terdahulu, peneliti membuat rekayasa hasil panen jika sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) diterapkan secara nyata. Dampak dari penerapan sistem *IoT* pada kebun organik adalah seluruh kegiatan pengelolaan kebun dapat dilakukan secara automasi sehingga proses pengelolaan kebun menjadi konsisten. Serta, sistem dapat melakukan tindakan secara *real-time* dan akurat dalam menjaga kondisi lingkungan kebun organik agar tetap optimal untuk pertumbuhan tanaman. Berikut merupakan tabel analisis dari perbedaan

berbagai aspek pada pengelolaan kebun secara konvensional dengan menerapkan sistem *Internet of Things* (IoT).

Tabel 5. 5 Perbedaan antara pengelolaan kebun secara konvensional dengan penerapan sistem IoT

No	Aspek	Konvensional	Sistem IoT
1	Proses pengelolaan kebun (Penyiraman air, dan pemberian pupuk, pemberian pestisida)	Terjadwal namun tidak konsisten karena dilakukan secara manual	Terjadwal (Dilakukan secara automasi)
2	Pemantauan kondisi kebun (Suhu dan kelembaban tanah)	Tidak konsisten karena kesibukan pengelola kebun dan tidak akurat karena kondisi kebun tergantung dari persepsi subjektif dari pengawas	Konsisten dan akurat karena menggunakan sensor yang memberikan data kuantitatif
3	Dosis nutrisi (Air, Pupuk cair, dan Pestisida)	Tidak terukur (Subjektif karena tergantung dari persepsi pemberi nutrisi)	Terukur karena jumlah pemberian nutrisi telah ditetapkan

Maka dari itu, hal-hal tersebut diharapkan dapat meningkatkan jumlah panen yang dihasilkan dengan kenaikan persentase sebesar 20 – 30% serta dapat mengurangi atau bahkan mengeliminasi kegagalan panen yang terjadi pada kebun organik Nara Kupu Yogyakarta.

BAB VI

KESIMPULAN

6.1 Kesimpulan

Berdasarkan dari penelitian yang dilakukan mengenai rancang bangun sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) menggunakan metode *Prototyping* dan *Usability Testing*, diperoleh kesimpulan sebagai berikut:

1. Sistem *monitoring* dan *controlling* berbasis *Internet of Things* (IoT) dengan metode berhasil dirancang dan dibangun dengan menggunakan metode *Prototyping* dan *Usability Testing* dengan menggabungkan berbagai komponen seperti *Microcontroller* ESP32, sensor kelembaban tanah, sensor suhu tanah, sensor *Ultrasonic*, pompa air, *Relay 4 Channel*, dan LCD. Sistem ini memungkinkan pemantauan kondisi kebun organik secara real-time dan akurat serta mengontrol pemberian nutrisi seperti air, pupuk cair, dan pestisida secara automasi.
2. Implementasi sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) memungkinkan proses pengelolaan kebun menjadi efisien, akurat dan *real-time*. Efisiensi tersebut dibuktikan dari hasil *Usability Testing* pada sistem yang bangun mendapatkan nilai *usability* yang baik yaitu 74,375, sedangkan akurasi dibuktikan oleh pembacaan dari berbagai sensor pada sistem ini yang memberikan data berbentuk kuantitatif, serta *real-time* dibuktikan oleh proses *monitoring* dan *controlling* yang dilakukan secara instan melalui konektivitas internet.
3. Kombinasi dari berbagai komponen pada sistem *monitoring* dan *controlling* kebun organik berbasis *Internet of Things* (IoT) memungkinkan sistem untuk mengirimkan notifikasi pada aplikasi *Blynk* secara akurat dan *real-time* melalui konektivitas internet. Akurasi dan *real-time* tersebut telah dibuktikan dari hasil pengujian sistem menggunakan pendekatan *Black box* yang memberikan hasil yang valid.

6.2 Saran

Berdasarkan kesimpulan yang diperoleh, terdapat beberapa saran yang dapat diberikan untuk penelitian selanjutnya:

1. Penerapan sensor untuk mendeteksi jumlah hama pada kebun, sehingga pemberian pestisida dapat disesuaikan dengan parameter tersebut.

2. Pengembangan operasi/fitur untuk menggemburkan tanah secara otomatis, sehingga tidak perlu dilakukan secara manual oleh pengelola kebun.
3. Mengkaji kembali arah (Posisi atas/bawah) dan jenis alat yang digunakan (*Sprinkler, Nozzle*, dll) pada pemberian nutrisi untuk mengoptimal pertumbuhan tanaman.

DAFTAR PUSTAKA

- Aarathi, R. (2023). Soil Property Analysis Using IoT : A Case Study Implementation in Backyard Gardening Smart Soil Property Analysis Using IoT : A Case Study. *Procedia Computer Science*, 218(22), 2842–2851. Elsevier B.V. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1877050923002557?via%3Dihub>
- Abdulsalam, K. A., Adebisi, J. A., & Adetona, I. O. (2023). A Microcontroller-Based Combinational Lock System. *World Scientific News*, 181(4), 32–52.
- Abraham, A. (2021). *AI, Edge and IoT-Based Smart Agriculture*. (S. D. Biswaranjan Acharya, Joel J.P.C. Rodrigues, Subhendu Kumar Pani, Ed.). Auburn: Elsevier Science.
- Adani, F., & Salsabil, S. (2019). Internet of Things: Sejarah Teknologi dan Penerapannya. *Cyber Resilience of Systems and Networks*, 14(2), 1–150. Retrieved from http://link.springer.com/10.1007/978-3-319-77492-3_16
- Affandi, P. (2018). *Manajemen Sumber Daya Manusia: Teori, Konsep dan Indikator* (1st ed.). Pekanbaru: Zanafa Publishing.
- Ahmed, K. R. (2023). *Image Processing Applications* (3rd ed.). Noida: CIIR Research Publications.
- Al-Obaidi, M. A. M., Radhi, M. A. H., Ibrahim, R. S., & Sutikno, T. (2020). Technique Smart Control Soil Moisture System to Watering Plant Based on IoT with Arduino Uno. *Bulletin of Electrical Engineering and Informatics*, 9(5), 2038–2044.
- Ali, A. H., Chisab, R. F., & Mnati, M. J. (2019). A Smart Monitoring and Controlling for Agricultural Pumps Using LoRa IOT Technology. *Indonesian Journal of Electrical Engineering and Computer Science*, 13(1), 286–292.
- Ardiansah, I., Bafdal, N., Suryadi, E., & Bono, A. (2020). Greenhouse Monitoring and Automation Using Arduino: A Review on Precision Farming and Internet of Things (IoT). *International Journal on Advanced Science, Engineering and Information Technology*, 10(2), 703–709.
- Asih, M. (2018). Sistem Pendukung Keputusan Fuzzy Mamdani pada Alat Penyiraman Tanaman Otomatis. *Jurnal Sistem Informasi*, 2(1), 41–52.
- Bagus, M., Huda, R., & Kurniawan, W. D. (2022). Analisa Sistem Pengendalian Temperatur Menggunakan Sensor DS18B20 Berbasis Mikrokontroler Arduino. *Jurnal Rekayasa Mesin*, 7(2), 18–23. Retrieved from <https://ejournal.unesa.ac.id/index.php/jurnal-rekayasa-mesin/article/view/47897>
- Brooke, J. (2020). SUS: A Quick and Dirty Usability Scale. *Usability Evaluation In*

- Industry*, 8(5), 207–212. Retrieved from <https://www.researchgate.net/publication/228593520>
- Brous, P., Janssen, M., & Herder, P. (2020). The Dual Effects of The Internet of Things (IoT): A Systematic Review of The Benefits and Risks of IoT Adoption by Organizations. *International Journal of Information Management*, 51(4), 1019–1052.
- Budihartono, E., & Rakhman, A. (2022). Monitoring Suhu dan Kelembaban Tanah pada Budidaya Porang Berbasis Arduino. *Smart Comp: Jurnalnya Orang Pintar Komputer*, 11(1), 9–13.
- Cahyani, M. P. (2023). IoT Dalam Smart Farming 4.0 untuk Upaya Tingkatkan Efisiensi Agribisnis. *Teknois : Jurnal Ilmiah Teknologi Informasi dan Sains*, 3(2), 154–190.
- Darmawan, I. W. B., Kumara, I. N. S., & Khrisne, D. C. (2022). Smart Garden Sebagai Implementasi Sistem Kontrol Dan Monitoring Tanaman Berbasis Teknologi Cerdas. *Jurnal SPEKTRUM*, 8(4), 161.
- Deworo, D., Dharmawati, N. D., Pinandito, K., & Putri, A. G. (2023). Rancang Bangun Alat Penyiram Otomatis Menggunakan Sensor Kelembaban Tanah Berbasis Mikrokontroler pada Tanaman Kopi Design of an Automatic Sprinkler Using a Microcontroller-Based Soil Moisture Sensor in Coffee Plants Abstrak. *Agrotechno*, 8(1), 28–35.
- Dumas, J. S., & Redish, J. C. (1999). *A Practical Guide to Usability Testing* (Revised Ed.). Portland, Oregon: Intellect Books.
- Ervan Septa, I., Angga, R., & Irwan, P. (2022). Rancang Bangun Sistem Catu Daya Pada Smart Agriculture Rover Berbasis Internet of Things. *eProceedings of Engineering*, 9(5), 2379–2383.
- Francisco, C., Gomes, S., Santos, M., Francisco, C., & Gomes, S. (2022). Creation and Implementation of an IoT-based Thermometer Prototype for a Food Organization : Case Study. *Procedia Computer Science*, 199(2), 710–717. Elsevier B.V. Retrieved from <https://doi.org/10.1016/j.procs.2022.01.088>
- Gaitan, N. C. (2021). A Long-distance Communication Architecture for Medical Devices Based on Lorawan Protocol. *Electronics*, 10(8), 910–940.
- Galina, M., Safitri, C., Bukhori, I., Silitonga, A., & Suhartomo, A. (2022). An Implementation of Smart Agriculture for Optimizing Growth Using Sonic Bloom and IoT Integrated. *Jurnal Infotel*, 14(1), 65–74.
- Groover, M. (2022). *Introduction to Automation*. (M. Horison, H. Stark, & S. Rodrigez, Eds.) *Construction and Building Automation* (4th ed.). Bethlehem.
- Gumelar, A., & Yunus, K. (2021). Pengaruh Frekuensi Pemberian Pupuk Organik Cair dan Takaran Biochar terhadap Pertumbuhan dan Hasil Sawi Pakcoy (Brassica rappa

- L.). *Savana Cendana*, 6(1), 4–7.
- Hafizh, T., Permana, A., & Riza, T. (2018). Perancangan dan Implementasi Pompa Air Bertenaga Surya di Perumahan Permata Buah Batu. *E-Proceeding of Applied Science*, 4(3), 2633–2642.
- Hardani, Andriani, H., Ustiawaty, J., & Utami, E. (2020). *Metode Penelitian Kualitatif & Kuantitatif*. (H. Abadi, Ed.) (1st ed.). Yogyakarta: CV. Pustaka Ilmu.
- Harini, N., & Lavanya, P. (2022). High Security Locking System Using Arduino. *Journal of Interdisciplinary Cycle Research*, 25(2), 1065–1081. Retrieved from <https://www.researchgate.net/publication/361757496%0AHIGH-SECURITY>
- Horsky, J., McColgan, K., Pang, J. E., Melnikas, A. J., Linder, J. A., Schnipper, J. L., & Middleton, B. (2010). Complementary Methods of System Usability Evaluation: Surveys and Observations During Software Design and Development Cycles. *Journal of Biomedical Informatics*, 43(5), 782–790. Elsevier Inc. Retrieved from <http://dx.doi.org/10.1016/j.jbi.2010.05.010>
- Husmaini, U., & Pranomo. (2001). *Metodologi penelitian sosial / Husaini Usman, Purnomo Setiady Akbar*. (A. Prakoso, Ed.) (1st ed.). Jakarta: Bumi Aksara.
- Ishak, A., Ginting, R., Suwandira, B., & Fauzi Malik, A. (2020). Integration of Kano Model and Quality Function Deployment (QFD) to Improve Product Quality: A Literature Review. *IOP Conference Series: Materials Science and Engineering*, 1003(1), 110–120.
- Kant, R. (2023). Smart Agricultural Technology Experimental Performance of Smart IoT-enabled Drip Irrigation System Using and Controlled Through Web-based Applications. *Smart Agricultural Technology*, 4(3), 1019–1039. Elsevier B.V. Retrieved from <https://doi.org/10.1016/j.atech.2023.100215>
- Kiani, F., & Seyyedabbasi, A. (2018). Wireless Sensor Network and Internet of Things in Precision Agriculture. *International Journal of Advanced Computer Science and Applications*, 9(6), 99–103.
- Krug, S. (2000). *Don't Make Me Think! A Common Sense Approach to Web Usability* (1st ed.). San Francisco, California: New Riders Press.
- Kumar, A. (2021). *Internet of Things and Machine Learning in Agriculture*. (V. J. Jyotir Moy Chatterjee, Pramod Singh Rathore, Ed.) (1st ed.). Berlin: De Gruyter.
- Kuppler, J., Wieland, J., Junker, R. R., & Ayasse, M. (2021). Drought-induced Reduction in Flower Size and Abundance Correlates with Reduced Flower Visits by Bumble Bees. *AoB Plants*, 13(1), 1–8.
- Lewis, J. (2018). Measuring Perceived Usability: The CSUQ, SUS, and UMUX. *International Journal of Human-Computer Interaction*, 34(12), 1148–1156. Retrieved from <https://app.dimensions.ai/details/publication/pub.1100277191>

- Lysenko, E. A., Kozuleva, M. A., Klaus, A. A., Pshybytko, N. L., & Kusnetsov, V. V. (2023). Lower Air Humidity Reduced Both The Plant Growth and Activities of Photosystems I and II Under Prolonged Heat Stress. *Plant Physiology and Biochemistry*, 194(11), 246–262. Elsevier Masson SAS. Retrieved from <https://doi.org/10.1016/j.plaphy.2022.11.016>
- Mahmud, M. S. A., Buyamin, S., Mokji, M. M., & Abidin, M. S. Z. (2018). Internet of Things Based Smart Environmental Monitoring for Mushroom Cultivation. *Indonesian Journal of Electrical Engineering and Computer Science*, 10(3), 847–852.
- Manik, V., Primasari, C., Wibisono, Y., Bagas, A., & Irianto, P. (2021). Investigasi Usability pada Aplikasi Mobile Pembiayaan Mobil di Indonesia. *Jurnal Sains dan Informatika*, 7(1), 1–10.
- Mardianah, M., Setiyowati, T., & Ernawati, E. (2022). Minat dan Perilaku Petani dalam Penerapan Pertanian Organik di Tidore Maluku Utara. *Jurnal Ilmiah Inovasi*, 22(2), 206–214.
- Meisak, D., Hendri, & Agustini, S. R. (2022). Penerapan Metode Prototype Pada Perancangan Sistem Informasi Penjualan Mediatama Solusindo Jambi. *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, 1(4), 1–11.
- Miftahul Walid, Hoiriyah, H., & Fikri, A. (2022). Pengembangan Sistem Irigasi Berbasis Internet of Things (IoT). *Jurnal Mnemonic*, 5(1), 31–38.
- Mishra, B. K. (2021). *Next Generation of Internet of Things*. (R. K. Prasant Kumar Pattnaik, Ed.) (1st ed.). Singapore: Springer Nature Singapore.
- Nayagam, M. G., Vijayalakshmi, B., Somasundaram, K., Mukunthan, M. A., Yogaraja, C. A., & Partheeban, P. (2023). Control of Pests and Diseases in Plants Using IOT Technology. *Measurement: Sensors*, 26(12), 1007–1014. Elsevier Ltd. Retrieved from www.sciencedirect.com/journal/measurement-sensors Control
- Nboyine, J. A., Asamani, E., Agboyi, L. K., Yahaya, I., Kusi, F., Adazebra, G., & Badii, B. K. (2022). Assessment of The Optimal Frequency of Insecticide Sprays Required to Manage Fall Armyworm (*Spodoptera frugiperda* J.E Smith) in Maize (*Zea mays* L.) in Northern Ghana. *CABI Agriculture and Bioscience*, 3(1), 1–11. BioMed Central. Retrieved from <https://doi.org/10.1186/s43170-021-00070-7>
- Nugroho, P. A. (2022). Sistem Pengaman Sepeda Motor Menggunakan E-Ktp Berbasis Arduino Uno. *Jeis: Jurnal Elektro Dan Informatika Swadharma*, 2(2), 9–16.
- Onwuka, B. (2019). Effects of Soil Temperature on Some Soil Properties and Plant Growth. *Advances in Plants & Agriculture Research*, 8(1), 89–93.
- Pal, D., & Vanijja, V. (2020). Perceived Usability Evaluation of Microsoft Teams as an Online Learning Platform During COVID-19 Using System Usability Scale and Technology Acceptance Model in India. *Children and Youth Services Review*,

- 119(10), 136–148. Retrieved from www.elsevier.com/locate/chilyouth
- Pricillia, T., & Zulfachmi. (2021). Survey Paper : Perbandingan Metode Pengembangan Perangkat Lunak. *Bangkit Indonesia*, 10(1), 6–12.
- Quy, V. K., Hau, N. Van, Anh, D. Van, Quy, N. M., Ban, N. T., Lanza, S., Randazzo, G., et al. (2022). IoT-Enabled Smart Agriculture: Architecture, Applications, and Challenges. *Applied Sciences (Switzerland)*, 12(7), 256–275.
- Ramakant, V., Khatri, N., Kumar, S., Shaddad, A., Abdul-qawy, H., & Kumar, A. (2023). A Systematic Review of IoT Technologies and Their Constituents for Smart and Sustainable Agriculture Applications. *Scientific African*, 19(7), 1–14. Elsevier B.V. Retrieved from <https://doi.org/10.1016/j.sciaf.2023.e01577>
- Ramirez, J., Manuel, L., & Fernandez, E. (2020). Monitoring of Temperature in Retail Refrigerated and Software. *Sensors*, 20(864), 2–18.
- Ratna, S., & Ibrahim, W. (2021). Perancangan dan Pengujian Alat untuk Monitoring Kelembaban Tanah dan Pemberian Pupuk Cair pada Tanaman Cabai berbasis Internet of Things. *Technologia*, 12(4), 286–291.
- Rehman, A., Saba, T., Kashif, M., Fati, S. M., Bahaj, S. A., & Chaudhry, H. (2022). A Revisit of Internet of Things Technologies for Monitoring and Control Strategies in Smart Agriculture. *Agronomy*, 12(1), 1–21.
- Sandi, G. H., & Fatma, Y. (2023). Pemanfaatan Teknologi Internet of Things (Iot) Pada Bidang Pertanian. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), 1–5.
- Santoso, G., Hani, S., & Putra, U. D. (2022). Monitoring kualitas tanah lahan pertanian Desa Sidorejo menggunakan sensor pH tanah dan Internet of Things. *Jurnal Nusantara Mengabdi*, 2(1), 1–10.
- Santoso, I. H., & Irawan, A. I. (2022). Analisis Perbandingan Kinerja Sensor Jarak HC-SR04 dan GP2Y0A21YK Dengan Menggunakan Thingspeak dan Wireshark. *Jurnal Rekayasa Elektrika*, 18(1), 43–52.
- Sedarmayanti. (2019). *Perencanaan Dan Pengembangan Sumber Daya Manusia Untuk Meningkatkan Kompetensi, Kinerja, Dan Produktivitas Kerja* (1st ed.). Bandung: Refika Aditama.
- Seran, M. O. ., Tuas, M. ., Pareira, M. ., & Gumelar, A. . (2022). Pengaruh Komposisi Media Tanam dan Frekuensi Penyiraman terhadap Pertumbuhan dan Produksi Tanaman Selada (*Lactuca sativa* L.). *Savana Cendana*, 7(3), 45–47.
- Setiawan, M. H., Ma'arif, A., Marhoon, H. M., Sharkawy, A., & Dahlan, A. (2023). Distance Estimation on Ultrasonic Sensor Using Kalman Filter. *Buletin Ilmiah Sarjana Teknik Elektro*, 5(2), 210–217.
- Shofa, D., Tavana Dewi, D., Muhammad Faris Ihda Fuad Baharudin, I., & Mitasari, H.

- (2021). Rancang Bangun Mesin Pemberi Pupuk Cair Otomatis Hemat Daya Berbasis Iot untuk Budaya Tanaman Organik. *Jurnal Rekayasa Mesin*, 16(1), 109–115. Retrieved from <https://jurnal.polines.ac.id/index.php/rekayasa>
- Siswidiyanto, Munif, A., Wijayanti, D., & Haryadi, E. (2020). Sistem Informasi Penyewaan Rumah Kontrakan Berbasis Web Dengan Menggunakan Metode Prototype. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, 15(1), 18–25.
- Subashini, M. M., Das, S., Heble, S., Raj, U., & Karthik, R. (2018). Internet of things Based Wireless Plant Sensor for Smart Farming. *Indonesian Journal of Electrical Engineering and Computer Science*, 10(2), 456–468.
- Sugiyono. (2018). *Populasi Penelitian Kuantitatif, Kualitatif, dan R&D*. (B. Mahmud, Ed.) (1st ed.). Bandung: Alfabeta, CV.
- Supriyanto, D., Prayitno, M., & Affandi, L. (2022). Smart Greenhouse Coffee Dryer with Fuzzy Algorithm on Internet of Things Platform. *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, 7(1), 1–8.
- Supriyono, S. (2020). Software Testing with The Approach of Blackbox Testing on The Academic Information System. *IJISTECH (International Journal of Information System and Technology)*, 3(2), 227–233. Retrieved from <https://ijistech.org/ijistech/index.php/ijistech/article/view/54>
- Syahputra, R. (2023). Alat Pendeteksi Level Air Otomatis Pada Tangki Air Wudhu Masjid Ulil Albab UNSRAT Berbasis Mikrokontroler. *Informatika*, 11(1), 88–100.
- Tang, X., Gao, F., Ji, Y., & Wang, S. (2021). Research on Consistency of IoT Data Communication Based on Eigenvector Vector. *Journal of Physics: Conference Series*, 1748(3), 1–6.
- Thombare, S., Chambhare, S., Gorte, P., & Pinneti, T. (2023). Greenhouse Monitoring and Controlling System Using IoT. *International Research Journal of Modernization in Engineering Technology and Science*, 5(5), 6558–6564.
- Tudi, M., Ruan, H. D., Wang, L., Lyu, J., Sadler, R., Connell, D., Chu, C., et al. (2021). Agriculture Development, Pesticide Application and Its Impact on the Environment. *International Journal of Environmental Research and Public Health*, 18(3), 1–24.
- Visconti, P., Giannoccaro, N. I., de Fazio, R., Strazzella, S., & Cafagna, D. (2020). IoT-oriented Software Platform Applied to Sensors-based Farming Facility with Smartphone Farmer App. *Bulletin of Electrical Engineering and Informatics*, 9(3), 1095–1105.
- Wagyana, A. (2019). Prototipe Modul Praktik untuk Pengembangan Aplikasi Internet of Things (IoT). *Jurnal Ilmiah Setrum*, 8(2), 238.
- Wang, B., Liu, X., & Zhang, Y. (2022). Internet of Things. *Internet of Things and BDS*

- Application* (pp. 71–127). Singapore: Springer Singapore. Retrieved from https://doi.org/10.1007/978-981-16-9194-2_2
- Wicaksono, G., Suryani, E., & Hendrawan, R. (2021). Increasing Productivity of Rice Plants Based on IoT (Internet of Things) to Realize Smart Agriculture Using System Thinking Approach. *Procedia Computer Science*, 197(3), 607–616. Elsevier B.V. Retrieved from <https://doi.org/10.1016/j.procs.2021.12.179>
- Widjajanto, D. W. (2021). Penerapan Pertanian Organik dalam Mendukung Ketahanan Pangan pada Masa dan Pasca Pandemi COVID-19. In D. Ahmad Ni'matullah Al-Baari (Ed.), *Ketahanan Pangan dari Aspek Kesehatan, Pertanian, dan Sosial di Masa Pandemi* (1st ed., p. 50=59). Semarang: Indonesian Food Technologists.
- Xu, J., Gu, B., & Tian, G. (2022). Review of Agricultural IoT Technology. *Artificial Intelligence in Agriculture*, 6(2), 10–22. The Authors. Retrieved from <https://doi.org/10.1016/j.aiia.2022.01.001>
- Yadav, L. N. (2023). Study on Internet of Things Based Applications. *IJARCE (International Journal of Advanced Research in Computer and Communication Engineering)*, 11(5), 803=806.
- Zahwa, D. A. (2023). Pengaruh Lingkungan Kerja Fisik, Karakteristik Individu, dan Motivasi Ekstrinsik Terhadap Kinerja Pegawai pada Kementerian Pertahanan RI Jakarta Pusat. *Journal of Physics A: Mathematical and Theoretical*, 44(8), 149–200.
- Zhao, X., Li, W., Zhou, L., Song, G., Ba, Q., & Ou, J. (2019). Active Thermometry Based DS18B20 Temperature Sensor Network for Offshore Pipeline Scour Monitoring Using K -Means Clustering Algorithm. *International Journal of Distributed Sensor Networks*, 20(13), 1–11.

LAMPIRAN

A-Surat Izin Permohonan Tugas Akhir



FAKULTAS
TEKNOLOGI INDUSTRI

Gerbang KH. Mas Mansur
Kampus Terpadu Universitas Islam Indonesia
Jl. Kalirejo km 14,5 Yogyakarta 55584
T. (0274) 898444 ext. 4110, 4100
F. (0274) 895007
E. ft@uii.ac.id
W. ft.uii.ac.id

Nomor : 148/Penelitian TA/Sek.Prodi.S1/20/TI/VIII/2022
Lampiran : -
Hal : Permohonan ijin penelitian tugas akhir

Kepada Yth.
Bapak/Ibu Pimpinan

Assalamu'alaikum wr. wb

Berkaitan dengan kegiatan penelitian mahasiswa Prodi Teknik Industri Fakultas Teknologi Industri Universitas Islam Indonesia yaitu

No	Nama Mahasiswa	NIM	Penelitian
1.	Irfan Rasyid Jatmiko	19522048	RANCANG BANGUN SISTEM MONITORING DAN CONTROLLING KEBUN ORGANIK BERBASIS INTERNET OF THINGS DENGAN METODE PROTOTYPING DAN USABILITY TESTING (STUDI KASUS: NARA KUPU YOGYAKARTA)

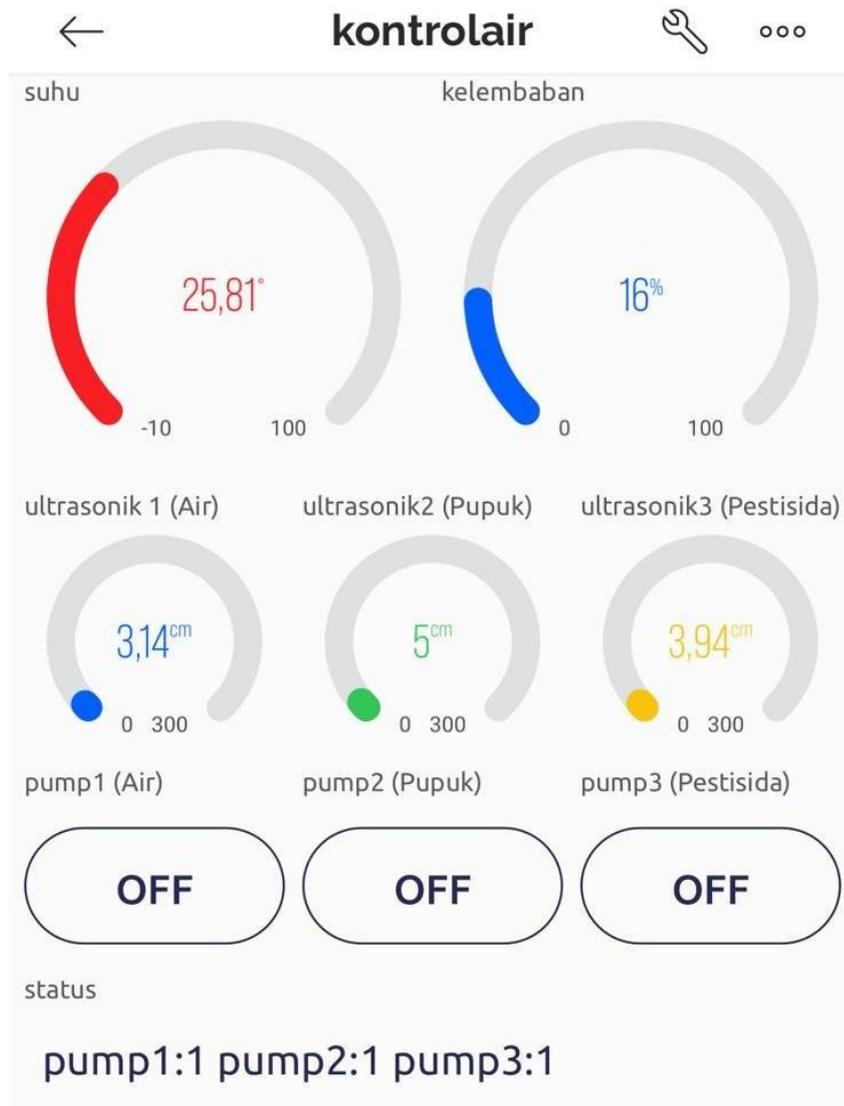
Maka bersama ini kami memohon kepada Bapak/Ibu untuk memberikan ijin kepada Mahasiswa tersebut untuk melakukan penelitian di instansi yang Bapak/Ibu pimpin.

Demikianlah surat permohonan ini kami sampaikan, atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb



B-Hasil Dashboard *Blynk* Cloud IoT



C-Hasil Perancangan *Prototype*



D-Hasil Kuesioner System Usability Testing

Nama : Rapi Rahma A

Usia : 26 tahun

Pekerjaan : staff kebun

No.	Pertanyaan	1	2	3	4	5
1	Saya akan senang untuk menggunakan sistem ini secara sering.				✓	
2	Saya merasa sistem ini rumit untuk digunakan.		✓			
3	Saya merasa sistem ini mudah digunakan.				✓	
4	Saya membutuhkan bantuan dari orang lain dalam Menggunakan sistem ini.		✓			
5	Saya merasa fitur-fitur sistem ini diintegrasikan dengan baik.				✓	
6	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.			✓		
7	Saya merasa orang lain akan memahami sistem ini dengan cepat.				✓	
8	Saya merasa sistem ini membingungkan.		✓			
9	Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.				✓	
10	Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.		✓			

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju

Rapi

Nama : ROHMAT
 Usia : 28
 Pekerjaan : Staf KEBUN

No.	Pertanyaan	1	2	3	4	5
1	Saya akan senang untuk menggunakan sistem ini secara sering.					✓
2	Saya merasa sistem ini rumit untuk digunakan.			✓		
3	Saya merasa sistem ini mudah digunakan.				✓	
4	Saya membutuhkan bantuan dari orang lain dalam Menggunakan sistem ini.	✓				
5	Saya merasa fitur-fitur sistem ini diintegrasikan dengan baik.				✓	
6	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.			✓		
7	Saya merasa orang lain akan memahami sistem ini dengan cepat.					✓
8	Saya merasa sistem ini membingungkan.	✓				
9	Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.				✓	
10	Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.	✓				

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju



Nama : *Ganawan*
 Usia : *19*
 Pekerjaan : *guru kelas*

No.	Pertanyaan	1	2	3	4	5
1	Saya akan senang untuk menggunakan sistem ini secara sering.			✓		
2	Saya merasa sistem ini rumit untuk digunakan.	✓				
3	Saya merasa sistem ini mudah digunakan.				✓	
4	Saya membutuhkan bantuan dari orang lain dalam Menggunakan sistem ini.		✓			
5	Saya merasa fitur-fitur sistem ini diintegrasikan dengan baik.				✓	
6	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.	✓				
7	Saya merasa orang lain akan memahami sistem ini dengan cepat.				✓	
8	Saya merasa sistem ini membingungkan.	✓				
9	Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.				✓	
10	Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.				✓	

- 1 = Sangat Tidak Setuju
 2 = Tidak Setuju
 3 = Netral
 4 = Setuju
 5 = Sangat Setuju

Ganawan

Nama : Toguh
 Usia : 38
 Pekerjaan : Staf Kebu

No.	Pertanyaan	1	2	3	4	5
1	Saya akan senang untuk menggunakan sistem ini secara sering.				✓	
2	Saya merasa sistem ini rumit untuk digunakan.		✓			
3	Saya merasa sistem ini mudah digunakan.			✓		
4	Saya membutuhkan bantuan dari orang lain dalam Menggunakan sistem ini.	✓				
5	Saya merasa fitur-fitur sistem ini diintegrasikan dengan baik.					✓
6	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.		✓			
7	Saya merasa orang lain akan memahami sistem ini dengan cepat.					✓
8	Saya merasa sistem ini membingungkan.		✓			
9	Saya merasa yakin (tidak ragu) dalam menggunakan sistem ini.			✓		
10	Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.	✓				

- 1 = Sangat Tidak Setuju
 2 = Tidak Setuju
 3 = Netral
 4 = Setuju
 5 = Sangat Setuju



E-Prentasi Sistem untuk Evaluasi Pengguna





F-Pengkodean Arduino IDE

```
1 #define pump1 19
2 #define pump2 18
3 #define pump3 5
4 #define pinled 2
5 #define soil 34
6
7 #include <OneWire.h>
8 #include <DallasTemperature.h>
9
10 #define pinsuhu 23
11 OneWire configsensor(pinsuhu);
12 DallasTemperature sensor(&configsensor);
13
14 // token blynk
15 #define BLYNK_TEMPLATE_ID "TMPL60Ycash5y"
16 #define BLYNK_TEMPLATE_NAME "kontrolair"
17 #define BLYNK_AUTH_TOKEN "vXRcQGC0lBvixS23m5oGGyaBrlWsb5V2"
18
19 #define BLYNK_PRINT Serial // tampilkan informasi blynk di serial
20 #include <BlynkSimpleEsp32.h> // library blynk
21
22 // input wifi dan password
23 char auth[] = BLYNK_AUTH_TOKEN;
24 char ssid[] = "Bimonet"; // type your wifi name
25 char pass[] = "Elfatih1"; // type your wifi password
26 BlynkTimer timers;
27
28 #include <NTPClient.h>
29 const long utcOffsetInSeconds = 25200;
30 // Define NTP Client to get time
31 WiFiUDP ntpUDP;
32 NTPClient timeClient(ntpUDP, "id.pool.ntp.org", utcOffsetInSeconds);
33
```

```
34 int monthDay;
35 int currentMonth;
36 int currentYear;
37 String tanggal,waktu;
38 int jam;
39 int count1=0,count2=0; //LCD
40 float ultra1,ultra2,ultra3;
41 float tinggi1,tinggi2,tinggi3;
42 int tpump1=0,tpump2=0,tpump3=0;
43 float suhu;
44 int kelembaban;
45 int lock1,lock2,lock3;
46 int lock4,lock5;
47 int lock6,lock7,lock8;
48
49 int pin_trig[]={4,13,14,};
50 int pin_echo[]={15,12,27,};
51
52 // setting tinggi sensor (cm)
53 float setultra1 = 20.0;
54 float setultra2 = 20.0;
55 float setultra3 = 20.0;
56
57 // setting suhu (Celcius) dan kelembaban (%)
58 float setsuhu = 31.0;
59 int setkelembaban = 50;
60
61 // setting jam penyiraman air
62 int air1 = 6;
63 int air2 = 18;
```

```
65 //setting jam penyiraman pupuk cair
66 int jampupuk = 7;
67
68 //setting jam penyiraman pestisida
69 int jamhama = 8;
70
71 // setting tanggal pupuk cair
72 int pupukcair1 = 1;
73 int pupukcair2 = 15;
74
75 // setting tanggal semprot hama cair
76 int hama1 = 1;
77 int hama2 = 8;
78 int hama3 = 15;
79 int hama4 = 22;
80
81 // setting waktu pompa on (detik)
82 int waktupump1 = 60;
83 int waktupump2 = 60;
84 int waktupump3 = 60;
85
86 // setting kondisi habis penampungan (cm)
87 int habis1 = 5;
88 int habis2 = 5;
89 int habis3 = 5;
90
91 // lcd library
92 #include <LiquidCrystal_I2C.h>
93 LiquidCrystal_I2C lcd(0x27, 16, 2);
94
```

```
95 // membaca input dari blynk
96 BLYNK_WRITE(V4){
97   int pinValue = param.asInt();
98   if(pinValue==1) digitalWrite(pump1, 0); //on
99   if(pinValue==0) digitalWrite(pump1, 1); //off
100   Serial.print("pump1:");
101   Serial.println(pinValue);
102
103 }
104
105 // membaca input dari blynk
106 BLYNK_WRITE(V5){
107   int pinValue = param.asInt();
108   if(pinValue==1) digitalWrite(pump2, 0); //on
109   if(pinValue==0) digitalWrite(pump2, 1); //off
110   Serial.print("pump2:");
111   Serial.println(pinValue);
112
113 }
114
115 // membaca input dari blynk
116 BLYNK_WRITE(V6){
117   int pinValue = param.asInt();
118   if(pinValue==1) digitalWrite(pump3, 0); //on
119   if(pinValue==0) digitalWrite(pump3, 1); //off
120   Serial.print("pump3:");
121   Serial.println(pinValue);
122
123 }
124
```

```
129 float read_srf(int i){
130     unsigned int duration;
131     float distance;
132     // trigger off
133     digitalWrite(pin_trig[i], LOW);
134     delayMicroseconds(5);
135     // trigger on
136     digitalWrite(pin_trig[i], HIGH);
137     delayMicroseconds(10);
138     // trigger off
139     digitalWrite(pin_trig[i], LOW);
140     // baca pulsa
141     duration = pulseIn(pin_echo[i], HIGH);
142     // konversi ke cm
143     distance= duration*0.034/2;
144     return distance;
145 }
146
147 void get_time(){
148     timeClient.update();
149     unsigned long epochTime = timeClient.getEpochTime();
150     struct tm *ptm = gmtime ((time_t *)&epochTime);
151     monthDay = ptm->tm_mday;
152     currentMonth = ptm->tm_mon+1;
153     currentYear = ptm->tm_year+1900;
154     tanggal=String(monthDay);
155     tanggal+="-";
156     tanggal+=String(currentMonth);
157     tanggal+="-";
158     tanggal+=String(currentYear);
159     jam = timeClient.getHours();
160     waktu = timeClient.getFormattedTime();
161     Serial.println(tanggal);
```

```
162 Serial.println(waktu);
163 Serial.println(jam);
164 }
165
166 void display_lcd(){
167     count1++;
168     if( count1 >=5 ){count1=0;count2++;}
169     if(count2>1)count2=0;
170     lcd.clear();
171     if(count2==0){
172
173         lcd.setCursor(0,0);
174         lcd.print(waktu);
175         lcd.setCursor(11,0);
176
177         lcd.print(digitalRead(pump1));
178         lcd.print(" ");
179         lcd.print(digitalRead(pump2));
180         lcd.print(" ");
181         lcd.print(digitalRead(pump3));
182
183         lcd.setCursor(0,1);
184         lcd.print(tanggal);
185         lcd.setCursor(11,1);
186         lcd.print("K:");
187         lcd.print(kelembaban);
188
189     }
190     if(count2==1){
191
192         lcd.setCursor(0,0);
193         lcd.print("S:");
194         lcd.print(suhu,1);
```

```
195     lcd.setCursor(9,0);
196     lcd.print("A:");
197     lcd.print(tinggi1,1);
198
199     lcd.setCursor(0,1);
200     lcd.print("B:");
201     lcd.print(tinggi2,1);
202     lcd.setCursor(9,1);
203     lcd.print("C:");
204     lcd.print(tinggi3,1);
205
206 }
207
208 }
209
210 // update ke blynk dan tampilan blynk
211 void tasktimer(){
212     digitalWrite(pinled,digitalRead(pinled)^1);
213
214     get_time();
215
216
217
218     kelembaban = map(analogRead(soil),0,4096,0,100);
219     ultra1 = read_srf(0);
220     ultra2 = read_srf(1);
221     ultra3 = read_srf(2);
222
223     tinggi1 = setultra1 - ultra1;
224     tinggi2 = setultra2 - ultra2;
225     tinggi3 = setultra3 - ultra3;
226
227     if(tinggi1<0)tinggi1=0;
```

```

228     if(tinggi2<0)tinggi2=0;
229     if(tinggi3<0)tinggi3=0;
230
231     sensor.requestTemperatures();
232     suhu = sensor.getTempCByIndex(0);
233
234     if(suhu > setsuhu){ //jika suhu diatas suhu yang telah di tentukan maka akan ada mengirim notifikasi ke blynk
235         if(lock1!=1){ // artinya jika 1 maka notifikasi
236             lock1=1;
237             digitalWrite(pump1, 0); //on
238             Blynk.logEvent("notif","Suhu melebihi batas, sistem melakukan penyiraman otomatis");
239         }
240     }
241     if(suhu < setsuhu){
242         if(lock1!=0){
243             lock1=0;
244             digitalWrite(pump1, 1); //off
245         }
246     }
247
248     if(kelembaban < setkelembaban){
249         if(lock2!=1){
250             lock2=1;
251             digitalWrite(pump1, 0); //on
252             Blynk.logEvent("notif","Kelembaban kurang dari batas, sistem melakukan penyiraman otomatis");
253         }
254     }
255
256     if(kelembaban > setkelembaban){
257         if(lock2!=0){
258             lock2=0;
259             digitalWrite(pump1, 1); //off
260         }

```

```

261     }
262
263     if(jam == air1){
264         if(lock3!=1){
265             lock3=1;
266             tpump1=0;
267             digitalWrite(pump1, 0); //on
268             Blynk.logEvent("notif","Penyiraman otomatis pagi");
269         }
270
271         if(tpump1>=waktupump1) {
272             if(lock3!=2){
273                 lock3=2;
274                 digitalWrite(pump1, 1);
275             }
276             } //off
277         else {tpump1++;}
278     }
279
280
281     if(jam == air2){
282         if(lock3!=3){
283             lock3=3;
284             tpump1=0;
285             digitalWrite(pump1, 0); //on
286             Blynk.logEvent("notif","Penyiraman otomatis sore");
287         }
288
289         if(tpump1>=waktupump1) {
290             if(lock3!=4){
291                 lock3=4;
292                 digitalWrite(pump1, 1);
293             }

```

```
294     }
295     } //off
296     else {tpump1++;}
297 }
298
299 if(monthDay == pupukcair1 && jam==jampupuk){
300     if(lock4!=1){
301         lock4=1;
302         tpump2=0;
303         digitalWrite(pump2, 0); //on
304         Blynk.logEvent("notif","Penyiraman pupuk cair minggu ke 1");
305     }
306
307     if(tpump2>=waktupump2) {
308         if(lock4!=2){
309             lock4=2;
310             digitalWrite(pump2, 1);
311         }
312     }
313     } //off
314     else {tpump2++;}
315 }
316
317 if(monthDay == pupukcair2 && jam==jampupuk){
318     if(lock4!=3){
319         lock4=3;
320         tpump2=0;
321         digitalWrite(pump2, 0); //on
322         Blynk.logEvent("notif","Penyiraman pupuk cair minggu ke 3");
323     }
324
325     if(tpump2>=waktupump2) {
326         if(lock4!=4){
```

```
327     lock4=4;
328     digitalWrite(pump2, 1);
329
330     }
331     } //off
332     else {tpump2++;}
333 }
334
335 if(monthDay == hama1 && jam==jamhama){
336     if(lock5!=1){
337         lock5=1;
338         tpump3=0;
339         digitalWrite(pump3, 0); //on
340         Blynk.logEvent("notif","Penyemprotan pestisida minggu ke 1");
341     }
342
343     if(tpump3>=waktupump3) {
344         if(lock5!=2){
345             lock5=2;
346             digitalWrite(pump3, 1);
347
348             }
349             } //off
350     else {tpump3++;}
351 }
352
353 if(monthDay == hama2 && jam==jamhama){
354     if(lock5!=3){
355         lock5=3;
356         tpump3=0;
357         digitalWrite(pump3, 0); //on
358         Blynk.logEvent("notif","Penyemprotan pestisida minggu ke 2");
359     }
```

```
361     if(tpump3>=waktupump3) {
362         if(lock5!=4){
363             lock5=4;
364             digitalWrite(pump3, 1);
365         }
366     } //off
367     else {tpump3++;}
368 }
369
370
371 if(monthDay == hama3 && jam==jamhama){
372     if(lock5!=5){
373         lock5=5;
374         tpump3=0;
375         digitalWrite(pump3, 0); //on
376         Blynk.logEvent("notif","Penyemprotan pestisida minggu ke 3");
377     }
378
379     if(tpump3>=waktupump3) {
380         if(lock5!=6){
381             lock5=6;
382             digitalWrite(pump3, 1);
383         }
384     } //off
385     else {tpump3++;}
386 }
387
388
389 if(monthDay == hama4 && jam==jamhama){
390     if(lock5!=7){
391         lock5=7;
392         tpump3=0;
393         digitalWrite(pump3, 0); //on
```

```

394     Blynk.logEvent("notif","Penyemprotan pestisida minggu ke 4");
395   }
396
397   if(tpump3>=waktupump3) {
398     if(lock5!=8){
399       lock5=8;
400       digitalWrite(pump3, 1);
401     }
402   } //off
403   else {tpump3++;}
404 }
405
406
407 if(tinggi1 < habis1){
408   if(lock6!=1){
409     lock6=1;
410     Blynk.logEvent("notif","Air habis, isi bak segera!");
411   }
412 }
413 if(tinggi1 > habis1){
414   lock6=0;
415 }
416
417 if(tinggi2 < habis2){
418   if(lock7!=1){
419     lock7=1;
420     Blynk.logEvent("notif","Pupuk habis, isi bak segera!");
421   }
422 }
423 if(tinggi2 > habis2){
424   lock7=0;
425 }
426

```

```

427 if(tinggi3 < habis3){
428   if(lock8!=1){
429     lock8=1;
430     Blynk.logEvent("notif","Pestisida habis, isi bak segera!");
431   }
432 }
433 if(tinggi3 > habis3){
434   lock8=0;
435 }
436
437 // menampilkan di blynk
438 Blynk.virtualWrite(V0,suhu);
439 Blynk.virtualWrite(V1,tinggi1);
440 Blynk.virtualWrite(V2,tinggi2);
441 Blynk.virtualWrite(V3,tinggi3);
442 Blynk.virtualWrite(V7,kelembaban);
443
444
445 Serial.print(" S:");
446 Serial.print(suhu);
447 Serial.print(" A:");
448 Serial.print(ultra1);
449 Serial.print(" B:");
450 Serial.print(ultra2);
451 Serial.print(" C:");
452 Serial.print(ultra3);
453 Serial.print(" K:");
454 Serial.print(kelembaban);
455
456 String text = "";
457 text = " pump1:" + String(digitalRead(pump1)) + " pump2:" + String(digitalRead(pump2)) + " pump3:" + String(digitalRead(pump3));
458 Blynk.virtualWrite(V8,text);
459 Serial.print(text);

```

```
461     Serial.println();
462     display_lcd();
463
464 }
465
466 // fungsi setup hanya di run sekali saat esp pertama on
467 // gunanya untuk inisialisasi / konfigurari input output
468 void setup(){
469     Serial.begin(9600);
470     Serial.println(" starting ");
471
472     pinMode(pinled, OUTPUT);
473     pinMode(pump1, OUTPUT);
474     pinMode(pump2, OUTPUT);
475     pinMode(pump3, OUTPUT);
476     for(int i=0; i<3; i++){
477         pinMode(pin_trig[i], OUTPUT);
478         pinMode(pin_echo[i], INPUT);
479     }
480
481     digitalWrite(pump1, 1); //off
482     digitalWrite(pump2, 1); //off
483     digitalWrite(pump3, 1); //off
484
485
486
487 lcd.begin();
488 lcd.backlight();
489 // tampilan awal lcd
490 lcd.clear();
491 lcd.setCursor(0,0);
492 lcd.print("ssid:");
493 lcd.print(ssid);
```

```
493 lcd.print(ssid);
494 lcd.setCursor(0,1);
495 lcd.print("pass:");
496 lcd.print(pass);
497
498
499 // inisialisasi blynk
500 Blynk.begin(auth, ssid, pass);
501 timers.setInterval(1000L, tasktimer);
502
503 Serial.println(" ready ");
504 timeClient.begin();
505 sensor.begin();
506 }
507
508 String in="";
509 void loop()
510 {
511
512     Blynk.run(); // jalankan blynk
513     timers.run(); // jalankan timer
514
515
516
517 }
```

