

**RANCANG BANGUN APLIKASI *MONITORING* SISTEM
HIDROPONIK *EBB AND FLOW* PADA *ROOFTOP*
GREENHOUSE BERBASIS WEB**

SKRIPSI

Disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik (S.T)



Disusun Oleh:

AHMAD AFLAHARRIJAL

3332190015

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SULTAN AGENG TIRTAYASA**

2024

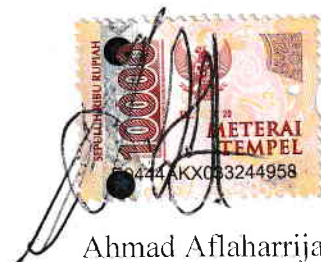
LEMBAR PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya sebagai penulis Skripsi berikut:

Judul : Rancang Bangun Aplikasi Monitoring Sistem
Hidroponik *Ebb and Flow* Pada *Rooftop Greenhouse*
Berbasis Web
Nama Mahasiswa : Ahmad Aflaharrijal
NPM : 3332190015
Fakultas/Jurusan : Teknik/Teknik Elektro

Menyatakan dengan sesungguhnya bahwa Skripsi tersebut di atas adalah benar-benar hasil karya asli saya dan tidak memuat hasil karya orang lain, kecuali dinyatakan melalui rujukan yang benar dan dapat dipertanggungjawabkan. Apabila dikemudian hari ditemukan hal-hal yang menunjukkan bahwa sebagian atau seluruh karya ini bukan karya saya, maka saya bersedia dituntut melalui hukum yang berlaku. Saya juga bersedia menanggung segala akibat hukum yang timbul dari pernyataan yang secara sadar dan sengaja saya nyatakan melalui lembar ini.

Cilegon, 10 Agustus 2024



A handwritten signature in black ink is written over a 1000 Rupiah postage stamp. The stamp is yellow and red, featuring the Garuda Pancasila emblem and the text 'REPUBLIK INDONESIA', '1000', and 'METERAI TEMPEL'. The serial number '444 AKXC33244958' is visible at the bottom of the stamp.

Ahmad Aflaharrijal
NIM. 3332190015

LEMBAR PENGESAHAN

Dengan ini ditetapkan bahwa Skripsi berikut.

Judul : Rancang Bangun Aplikasi Monitoring Sistem
Hidroponik *Ebb and Flow* Pada *Rooftop Greenhouse*
Berbasis Web

Nama Mahasiswa : Ahmad Aflaharrijal

NPM : 3332190015

Fakultas/Jurusan : Teknik/Teknik Elektro

Telah diuji dan dipertahankan pada tanggal 15 Juli 2024 Melalui Sidang Skripsi di Fakultas Teknik, Universitas Sultan Ageng Tirtayasa, Cilegon, Banten dan dinyatakan LULUS / ~~TIDAK LULUS~~.

Dewan Penguji

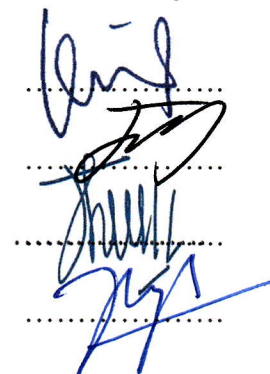
Tanda Tangan

Pembimbing I : Prof. Dr. Ir. Supriyanto, S.T., M.Sc., IPM.

Pembimbing II : Fadil Muhamad, S.T., M.T.

Penguji I : Dr. Eng. Teguh Firmansyah, S.T., M.T.

Penguji II : Dr. Eng. Rocky Alfanz, S.T., M.Sc.



Mengetahui,

Ketua Jurusan Teknik Elektro



(Dr. Eng. Rocky Alfanz, S.T., M.Sc.)
NIP. 198103282010121001

PRAKATA

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan laporan skripsi ini dengan judul “Rancang Bangun Aplikasi Monitoring Sistem Hidroponik *Ebb and Flow* Pada *Rooftop Greenhouse* Berbasis Web”. Penulisan laporan skripsi ini merupakan salah satu syarat untuk dapat menyelesaikan program studi S1 dan untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro, Universitas Sultan Ageng Tirtayasa. Saya menyadari bahwa tanpa adanya bantuan dan bimbingan dari berbagai pihak, sangatlah sulit bagi saya untuk dapat menyelesaikan laporan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Kedua orang tua dan keluarga penulis yang selalu memberikan dukungan dan doa selama penulis menyelesaikan skripsi ini;
2. Bapak Dr. Eng. Rocky Alfanz, S.T., M.Sc., sebagai ketua program studi Teknik Elektro, Fakultas Teknik, Universitas Sultan Ageng Tirtayasa;
3. Bapak Prof. Dr. Ir. Supriyanto, M.Sc., IPM., sebagai dosen pembimbing I skripsi yang telah membimbing serta memberikan arahan dalam penyusunan awal hingga selesainya laporan skripsi kepada penulis;
4. Bapak Fadil Muhamad, S.T., M.T., sebagai dosen pembimbing II skripsi yang telah membimbing serta memberikan arahan dalam penyusunan laporan skripsi hingga selesai kepada penulis;
5. Bapak Masjudin, S.T., M.T., selaku dosen pembimbing akademik;
6. Kepala Laboratorium Agroekoteknologi yang telah mengizinkan penulis untuk melaksanakan kegiatan penelitian skripsi di *greenhouse* agroekoteknologi;
7. Sulthan Esa Maulana dan Mutiara Ramadhani selaku tim *capstone* yang telah membantu selama proses penyusunan laporan skripsi hingga penelitian sampai selesai.

Penulis menyadari bahwa laporan ini masih banyak kekurangan dan masih jauh dari kata sempurna. Oleh karena itu, penulis memerlukan saran dan kritik yang bersifat membangun untuk kedepannya dan semoga laporan ini dapat bermanfaat

bagi para pembaca dan khususnya penulis sendiri sehingga dapat menjadi motivasi untuk mengembangkan kemampuan dalam menulis dan ilmu pengetahuan.

Cilegon, 15 Maret 2024

A handwritten signature in black ink, appearing to read 'Ahmad Aflaharrijal', written in a cursive style.

Ahmad Aflaharrijal

ABSTRAK

Budidaya tanaman di atas atap (*rooftop greenhouse*) semakin diminati sebagai solusi pemanfaatan lahan sempit di perkotaan. Namun, pemantauan kondisi tanaman secara manual seringkali tidak efisien dan memakan waktu. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun aplikasi sistem monitoring pada *rooftop greenhouse* berbasis web. Sistem monitoring ini terdiri dari beberapa komponen, yaitu; Sensor suhu, kelembapan udara, kelembapan tanah, mikrokontroler WEMOS D1R1, serta *Server web*. Data hasil pengukuran sensor dapat ditampilkan secara *real-time* pada aplikasi web dan pengguna dapat memantau kondisi tanaman dengan mudah. Performa dari aplikasi sistem monitoring ini telah diuji menggunakan Google *PageSpeed* dan GTMetrix dan mendapatkan hasil rata-rata performa 88% dengan *grade A*. Sistem ini juga dapat membantu pengguna dalam mengambil keputusan yang tepat untuk menjaga kesehatan tanaman.

Kata kunci: Sistem Monitoring, *Sensor*, Mikrokontroler, *Server Web*, Aplikasi *Web*

ABTRACT

Rooftop greenhouse cultivation has gained increasing popularity as a solution for utilizing limited land space in urban areas. However, manual monitoring of plant conditions is often inefficient and time-consuming. Therefore, this study aims to design and develop a web-based rooftop greenhouse monitoring system application. The monitoring system consists of several components, including temperature, humidity, and soil moisture sensors, a WEMOS D1R1 microcontroller, and a web server. Sensor measurement data can be displayed in real-time on the web application, allowing users to easily monitor plant conditions. The performance of the monitoring system application has been tested using Google PageSpeed and GTMetrix, achieving an average performance score of 88% with an A grade. This system can also assist users in making informed decisions to maintain plant health.

Keywords: Rooftop greenhouse, monitoring system, web application, sensor data, plant health

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERNYATAAN KEASLIAN SKRIPSI	ii
LEMBAR PENGESAHAN	iii
PRAKATA	iv
ABSTRAK	vi
ABTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian.....	3
1.4. Manfaat Penelitian.....	3
1.5. Batasan Masalah.....	4
BAB II TINJAUAN PUSTAKA	5
2.1. Greenhouse	5
2.2. Aklimatisasi.....	6
2.3. Hidroponik.....	6
2.4. Ebb and Flow System.....	8
2.5. IoT pada Sistem <i>Smart Farming</i>	9
2.6. Sistem Komunikasi Data	10
2.7. Sistem Keamanan	11
2.8. Metode <i>Waterfall</i>	12
2.9. Kajian Pustaka	13
BAB III METODE PENELITIAN	17
3.1. Alur Penelitian.....	17
3.2. Survei Lokasi Penelitian.....	19
3.3. Desain Sistem <i>Monitoring</i> Berbasis IoT	20
3.3.1. Pengajuan Solusi Alternatif	21
3.4. Komponen Penelitian	22
3.5. Skematik Alat	23

3.6.	<i>Flowchart</i> Cara Kerja Sistem	25
3.7.	<i>Usecase</i> Diagram Aplikasi Web.....	27
3.8.	Activity Diagram	27
3.9.	Desain Sistem Aplikasi Web	29
BAB IV HASIL DAN PEMBAHASAN		31
4.1.	Hasil Aplikasi AMOR S-FARM	31
4.1.1.	Teknologi <i>Hosting web</i> AMOR S-FARM.....	32
4.1.2.	Fitur Utama	32
4.1.3.	Hasil Antarmuka Pengguna (<i>User Interface Design</i>).....	34
4.2.	Pengujian <i>Database</i>	42
4.3.	Pengujian Komparasi Aplikasi (Waktu Data Tampil)	44
4.4.	Pengujian Performa Aplikasi.....	47
4.5.	Pengujian <i>Usability</i> Aplikasi.....	52
BAB V PENUTUP.....		54
5.1.	Kesimpulan.....	54
5.2.	Saran.....	54
DAFTAR PUSTAKA		
LAMPIRAN A		
LAMPIRAN B		
LAMPIRAN C		

DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi Greenhouse [17]	5
Gambar 2. 2 Jenis-jenis Sistem Hidroponik [23]	7
Gambar 2. 3 Sistem Hidroponik EBB and Flow [20]	8
Gambar 2. 4 Konsep Internet of Things [29]	9
Gambar 2. 5 Arsitektur Sistem IoT [32]	11
Gambar 3.1 Diagram Alir Penelitian	17
Gambar 3. 2 <i>Greenhouse</i> Laboratorium Agroekoteknologi Fakultas Pertanian Untirta	19
Gambar 3.3 Skema Sistem Monitoring Berbasis IoT	20
Gambar 3.4 Skematik Rangkaian Alat <i>Smart</i> Monitoring	24
Gambar 3.5 <i>Flowchart</i> Cara Kerja Sistem	26
Gambar 3.6 Diagram <i>Use Case</i> Aplikasi Sistem Monitoring	27
Gambar 3.7 Activity Diagram User Menggunakan Aplikasi Sistem Monitoring	28
Gambar 3.8 Desain Halaman Beranda Aplikasi Sistem Monitoring	29
Gambar 4.1 Logo AMOR S-FARM	31
Gambar 4.2 Halaman Awal Aplikasi AMOR S-FARM Versi Desktop	34
Gambar 4.3 Hasil <i>Interface</i> Halaman Awal Aplikasi AMOR S-FARM Versi Mobile	35
Gambar 4.4 Hasil <i>Interface</i> Halaman Data <i>Realtime</i> Monitoring	36
Gambar 4.5 Hasil <i>Interface</i> Halaman Menu <i>Sign-up</i>	37
Gambar 4.6 Hasil <i>Interface</i> Halaman Menu <i>Log-in</i> AMOR S-FARM	38
Gambar 4.7 Hasil <i>Interface</i> Halaman <i>Home</i> AMOR S-FARM	39
Gambar 4.8 Hasil <i>Interface</i> Halaman Grafik AMOR S-FARM	40
Gambar 4.9 Hasil <i>Interface</i> Halaman <i>History</i> Data AMOR S-FARM	41
Gambar 4.10 Hasil <i>Interface</i> Konfirmasi Aplikasi AMOR S-FARM	42
Gambar 4.11 Pengujian <i>Database</i> (Tamilan PHP MyAdmin)	42
Gambar 4.12 Pengujian <i>Database</i> (Tamilan Serial Monitor ArduinoIDE)	43
Gambar 4.13 Pengujian Komparasi Waktu Data Tampil (BLYNK)	44
Gambar 4.14 Pengujian Komparasi Waktu Data Tampil (Database AMOR S- FARM)	44

Gambar 4.15 Pengujian Komparasi Waktu Data Tampil (Serial Monitor ArduinoIDE)	45
Gambar 4.16 Hasil Pengujian Performa Dengan Google <i>Pagespeed</i>	47
Gambar 4.17 Hasil Pengujian Performa Dengan GT-metrix	50

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka.....	15
Tabel 3.1 Komponen Alat Sistem <i>Monitoring</i> Berbasis IoT	23
Tabel 4.1 Analisa Waktu Data Tampil Aplikasi	46
Tabel 4.2 Hasil Pengujian Google <i>Pagespeed</i>	48
Tabel 4.3 Hasil pengujian GT Matrix	50
Tabel 4.4 Rekap Nilai Kuesioner	52

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring perkembangan ilmu pengetahuan di bidang pertanian, metode kultur jaringan tanaman mulai banyak digunakan sebagai salah satu metode perbanyakan [1]. Metode perbanyakan bibit tanaman melalui teknik kultur jaringan dapat dijadikan solusi untuk menghasilkan bibit dalam jumlah banyak, waktu yang relatif singkat dan bebas dari penyakit [2]. Pada tahap terakhir teknik kultur jaringan ini, bibit tanam harus melalui tahap aklimatisasi [3].

Tahap aklimatisasi sering disebut juga tahap yang kritis karena kondisi lingkungan di lapangan sangat berbeda dengan lingkungan di dalam botol [4]. Permasalahan pada tahap aklimatisasi berasal dari faktor eksternal yaitu media tanam, kondisi suhu dan kelembaban [5]. Maka dari itu tahapan aklimatisasi ini dilakukan di dalam *greenhouse* dengan tujuan penyesuaian iklim sebelum dipindahkan ke lingkungan luar yang lebih ekstrim.

Greenhouse atau sering disebut rumah kaca secara umum di Indonesia didefinisikan sebagai bangunan konstruksi yang berfungsi untuk rekayasa kondisi lingkungan agar menyesuaikan kondisi yang diinginkan untuk pemeliharaan tanaman [6]. Saat ini pelaku kegiatan pertanian perkotaan atau *urban farming* menggunakan *greenhouse* pada atap gedung (*rooftop greenhouse*) melakukan pertanian. *Rooftop greenhouse* adalah bentuk pertanian terintegrasi bangunan atau pertanian tanpa lahan yang mencakup teknologi baik yang dilindungi (rumah kaca atap) maupun yang tidak dilindungi (kebun atau pertanian atap terbuka) [7].

Kendala dari *greenhouse* di atas gedung sendiri yaitu sulitnya pemantauan parameter *greenhouse* disaat kondisi gedung dalam keadaan terkunci, dengan memanfaatkan teknologi *Internet of Things* (IoT), kendala pemantauan parameter ini dapat diatasi [8]. IoT adalah interkoneksi gadget atau perangkat yang digunakan dalam kehidupan kita sehari-hari; rumah, papan iklan, kendaraan, dll, yang di atasnya tertanam komponen elektronik, GPS, perangkat lunak, dan koneksi internet yang memungkinkan benda-benda ini bertukar data di antara mereka [9].

IoT menggambarkan jaringan objek fisik yang disematkan dengan teknologi yang bertujuan untuk menghubungkan dan bertukar data dengan perangkat dan sistem lain melalui Internet [10]. Penerapan teknologi IoT dalam pertanian cerdas memberikan petani akses terhadap informasi yang lebih akurat dan terkini mengenai kondisi lahan mereka [11]. Konsep IoT ini mengacu pada 3 elemen utama pada arsitektur IoT, yakni; barang fisik yang dilengkapi modul IoT, Perangkat koneksi ke internet, dan *cloud data center* tempat untuk menyimpan aplikasi beserta *database* [12]. Dengan menggunakan *platform cloud* IoT, pengguna/petani dapat memantau dan mengontrol kondisi *greenhouse* dari jarak jauh melalui ponsel mereka melalui aplikasi web/android [13].

Terdapat beberapa penelitian terdahulu yang terkait dengan penelitian ini, seperti penelitian yang dilakukan oleh Armanda Suryaningrat, dkk., tentang “Sistem *Monitoring* Kelembaban Tanaman Cabai Rawit Menggunakan Irigasi Tetes Gravitasi Berbasis *Internet of Things* (IoT)”, penelitian ini menggunakan aplikasi android dan google spreadsheet yang dapat dipantau secara *online* dan *real-time* sebagai sistem monitoring [14]. Kekurangan pada penelitian ini pengguna harus meng-*install* aplikasi terlebih dahulu untuk menggunakannya, dan juga pengguna IOS tidak dapat menggunakan aplikasi sistem monitoring ini.

Selanjutnya penelitian yang dilakukan oleh Fitria Suryatin, dkk., tentang “Sistem Akuisisi Data Suhu dan Kelembaban Tanah Pada Irigasi Tetes Otomatis Berbasis *Internet of Things*” [15]. Serupa dengan penelitian sebelumnya, pada penelitian ini juga hanya menggunakan aplikasi android sebagai sistem monitoring, yang dimana pengguna IOS tidak dapat menggunakan aplikasi tersebut.

Berdasarkan uraian latar belakang dan beberapa metode penelitian di atas, maka penelitian ini bertujuan untuk membuat aplikasi berbasis web dan diimplementasikan untuk sistem *monitoring* pada *rooftop greenhouse*. Sehingga *monitoring* parameter pada *rooftop greenhouse* dapat dilakukan dengan perangkat apapun dan dimanapun secara *real-time*.

1.2. Rumusan Masalah

Dari latar belakang yang telah dikemukakan di atas, penulis dapat merumuskan beberapa masalah yang akan dikerjakan pada penelitian ini, yaitu:

1. Bagaimana cara merancang aplikasi berbasis *web* yang dapat digunakan untuk memantau parameter suhu lingkungan, kelembaban tanah, dan kelembaban udara pada bibit tanaman buah dengan sistem media tanam hidroponik pasang surut atau *ebb and flow* pada *rooftop greenhouse*?
2. Bagaimana cara menguji performa aplikasi berbasis *web* yang dibuat untuk memantau parameter tersebut?
3. Bagaimana kinerja aplikasi berbasis *web* yang dibuat sebagai sistem *monitoring* dari sistem hidroponik *ebb and flow* tanaman buah pada *rooftop greenhouse*?

1.3. Tujuan Penelitian

Atas dasar masalah yang ditulis dalam perumusan masalah, maka dapat ditentukan tujuan penelitian sebagai berikut:

1. Merancang aplikasi berbasis *web* yang digunakan sebagai sistem *monitoring* parameter dari sistem hidroponik *ebb and flow* tanaman buah durian pada *rooftop greenhouse*.
2. Menguji performa dari aplikasi berbasis *web* yang digunakan sebagai sistem *monitoring* parameter dari sistem hidroponik *ebb and flow* tanaman buah durian pada *rooftop greenhouse*.
3. Memantau kinerja dari aplikasi berbasis *web* yang digunakan sebagai sistem *monitoring* parameter dari sistem hidroponik *ebb and flow* tanaman buah durian pada *rooftop greenhouse*.

1.4. Manfaat Penelitian

Adapun manfaat yang dapat diberikan selama penelitian ini adalah sebagai berikut:

1. Bagi penulis, penelitian ini dapat menambah wawasan dan pengalaman secara langsung mengenai cara membuat aplikasi berbasis *web* sebagai sistem *monitoring* parameter yang dapat diimplementasikan pada sistem hidroponik *Ebb and Flow* tanaman buah durian pada *rooftop greenhouse*.
2. Bagi akademisi, penelitian ini diharapkan dapat menjadi acuan untuk penelitian selanjutnya yang berkaitan tentang aplikasi berbasis *web* sebagai

sistem *monitoring* parameter pada bibit tanaman buah durian dengan menggunakan sistem hidroponik pasang surut atau *ebb and flow* .

3. Bagi Masyarakat, penelitian ini diharapkan dapat membantu mempermudah masyarakat dalam memantau kondisi parameter pada sistem hidroponik *ebb and flow* tanaman buah durian pada *rooftop greenhouse* dengan memanfaatkan teknologi internet yang ada.

1.5. Batasan Masalah

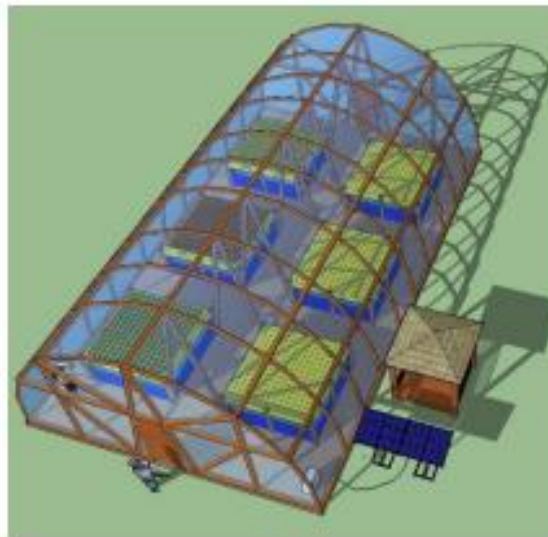
Berdasarkan tujuan penelitian yang telah dipaparkan sebelumnya, maka ruang lingkup penelitian ini dibatasi menjadi sebagai berikut.

1. Aplikasi yang dirancang merupakan aplikasi web.
2. Aplikasi dirancang untuk *monitoring* parameter suhu lingkungan, kelembaban udara, dan kelembaban tanah pada *rooftop greenhouse*.
3. Aplikasi dirancang untuk menampilkan nilai parameter dalam bentuk angka dan grafik.
4. Skripsi ini merupakan bagian *capstone design*, pembahasan pada skripsi ini lebih berfokus pada perancangan serta pembuatan aplikasi berbasis web dan pengujian aplikasi.

BAB II TINJAUAN PUSTAKA

2.1. Greenhouse

Greenhouse atau yang lebih dikenal sebagai rumah kaca di Indonesia adalah bangunan yang dirancang untuk menghindari dan memanipulasi kondisi lingkungan demi menciptakan iklim ideal untuk pertumbuhan tanaman. Dengan menggunakan *greenhouse*, pertumbuhan tanaman menjadi lebih maksimal dan terkontrol dibandingkan dengan budidaya di luar ruangan. Untuk mencapai kondisi ideal tersebut, berbagai aspek iklim harus dikendalikan atau dimanipulasi dengan tepat. [6]. Manipulasi lingkungan terjadi dengan dua cara yaitu menghindari kondisi lingkungan yang tidak diinginkan dan menciptakan kondisi lingkungan yang diinginkan [16]. Gambar *greenhouse* dapat dilihat pada Gambar 2.1 berikut.



Gambar 2. 1 Ilustrasi Greenhouse [17]

Pada Gambar 2.1 dapat dilihat ilustrasi *greenhouse* atau yang biasa dikenal dengan rumah kaca. Membangun rumah kaca tropis sangat mungkin dan menawarkan banyak manfaat untuk tanaman dan budidaya. Produksi dapat dilakukan sepanjang tahun apabila produksi di lapangan terbuka tidak memungkinkan karena sering turun hujan dan angin kencang. Struktur rumah kaca di daerah tropis sering menggunakan sisinya untuk perlindungan dan pengaturan suhu melalui ventilasi alami atau terkontrol dengan sekat yang dapat mengurangi

serangga dan hama. Penggunaan rumah kaca dalam budidaya merupakan salah satu cara untuk menciptakan lingkungan yang mendekati kondisi optimal bagi pertumbuhan tanaman. Rumah kaca pertama kali dikembangkan dan banyak digunakan di iklim subtropis. Penggunaan rumah kaca terutama ditujukan untuk melindungi tanaman terhadap suhu udara yang terlalu rendah di musim dingin [16].

2.2. Aklimatisasi

Aklimatisasi adalah tahap terakhir dalam perbanyakan tanaman menggunakan teknik kultur jaringan. Proses ini melibatkan pemindahan planlet ke media aklimatisasi dengan cahaya rendah dan kelembapan relatif tinggi. Secara bertahap, kelembapan diturunkan dan intensitas cahaya ditingkatkan. Tujuan aklimatisasi adalah untuk menyesuaikan bibit yang baru tumbuh melalui kultur *in vitro* atau kultur jaringan agar dapat beradaptasi dengan lingkungan luar [18].

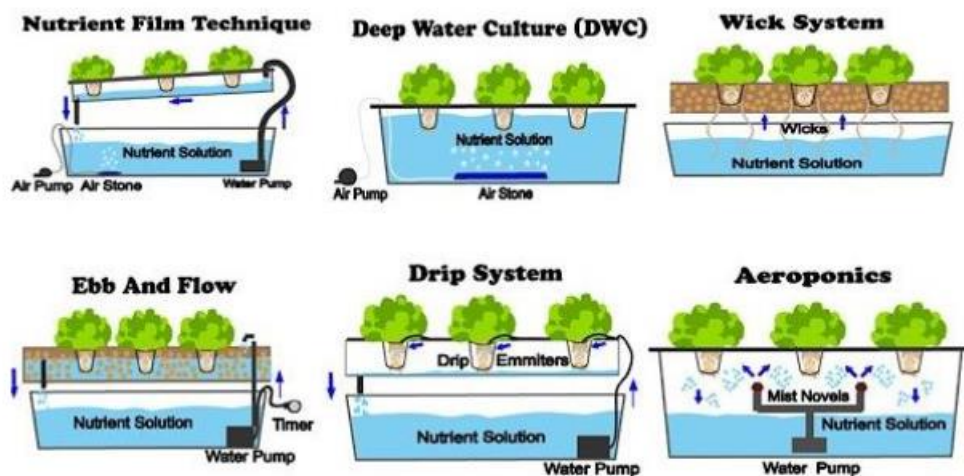
Aklimatisasi adalah fase adaptasi bagi tanaman hasil kultur jaringan yang awalnya tumbuh dalam kondisi terkendali dan kemudian harus beralih ke kondisi lapangan yang tidak terkendali. Selain itu, tanaman harus mengubah pola hidupnya dari *heterotrof* menjadi *autotrof*. Tahap aklimatisasi ini sangat penting dalam proses perbanyakan *in vitro*. Perbedaan besar, terutama dalam kelembapan dan intensitas cahaya antara lingkungan dalam botol dan luar botol, membuat aklimatisasi menjadi tahap yang sangat kritis [19].

2.3. Hidroponik

Konversi lahan pertanian menjadi kawasan industri mengurangi lahan subur untuk bertani. Hidroponik, metode bercocok tanam tanpa tanah yang menggunakan air dan nutrisi, menjadi solusi karena bisa memanfaatkan lahan terbatas [20]. Hidroponik adalah metode budidaya tanaman dalam lingkungan terkendali tanpa menggunakan tanah, di mana nutrisi tanaman diberikan secara terkontrol. Teknik ini dapat dilakukan dengan atau tanpa substrat. Awalnya, hidroponik dikenal sebagai penanaman tanpa tanah, namun seiring perkembangan teknologi, hidroponik digunakan untuk menumbuhkan tanaman dengan mengatur nutrisi sesuai kebutuhannya. Larutan unsur hara atau nutrisi yang berfungsi sebagai sumber air dan mineral sangat penting bagi pertumbuhan dan kualitas tanaman

dalam sistem hidroponik. Namun, jenis dan kadar larutan nutrisi harus diperhatikan dan dikontrol dengan tepat. Ketidakseimbangan nutrisi dapat menyebabkan tanaman menjadi kerdil, daun menguning dan gugur, serta rendahnya luas daun, sehingga tanaman tidak saling menaungi [21].

Metode Hidroponik sendiri memiliki beberapa jenis, diantaranya; aeropoinik, sistem tetes, sistem *Nutrient Film Technique* (NFT), sistem *water culture*, dan sistem pasang surut (*EBB and flow*) [22]. Gambar jenis-jenis sistem hidroponik dapat dilihat pada Gambar 2.2.



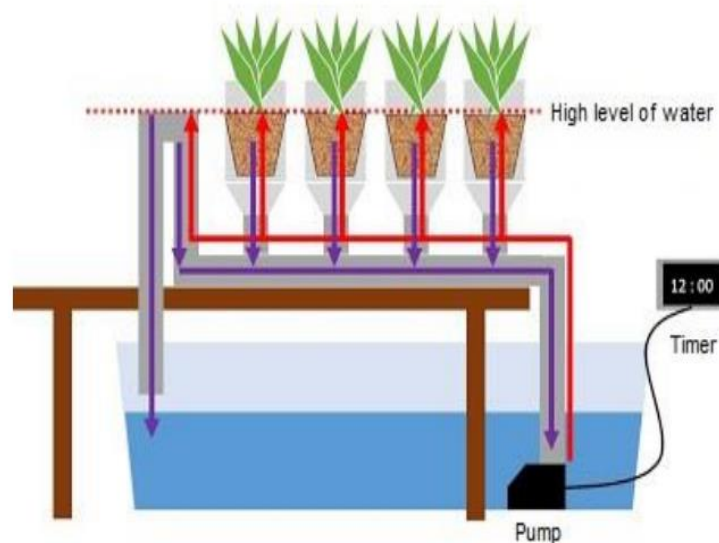
Gambar 2. 2 Jenis-jenis Sistem Hidroponik [23]

Pada Gambar 2.2 dapat dilihat beberapa jenis sistem hidroponik. Dr. William Frederick Gericke, ahli fisiologi tanaman dari Nebrasaka, membuat istilah "hidroponik" pada tahun 1936. Hidroponik adalah proses budidaya tanaman hias yang dapat dikonsumsi dengan memasukkannya ke dalam larutan air atau larutan nutrisi. Salah satu metode budidaya tanaman bersama dengan media tanah adalah hidroponik. Sebagai sumber makanan dan pertumbuhan tanaman, tanaman yang ditanam diberikan nutrisi yang kaya akan unsur-unsur penting dalam bentuk cairan atau larutan. Beberapa sistem memperlakukan akar untuk memberikan suplai makanan dengan memandikan akarnya, tetapi larutan ini disuplai di sekitar akar tanaman. Tanaman yang ditanam melalui hidroponik biasanya lebih sehat daripada menanam di tanah. Banyak keuntungan membudidayakan tanaman melalui hidroponik, salah satunya adalah bahwa tanaman tidak bersentuhan dengan tanah, menjaga kondisi tanaman dari hama dan penyakit yang datang dari atas maupun

dalam tanah, dan juga dapat memberikan nutrisi yang seimbang untuk tanaman [24].

2.4. Ebb and Flow System

Ebb and Flow System atau teknik pasang surut merupakan salah satu teknik hidroponik yang memanfaatkan prinsip pasang surut. Pemberian nutrisi pada akar tanaman dilakukan dengan cara menggenang (pasang dan surut secara bergantian) [25]. Metode *EBB and Flow* (Pasang Surut) merupakan teknik hidroponik yang umum digunakan di mana media pertumbuhan diisi dengan larutan nutrisi, dan larutan yang tidak diserap akan kembali ke bak penampung. Pengaturan waktu pasang surut dapat dilakukan menggunakan timer. Namun, penggunaan timer memiliki kekurangan, seperti konsumsi listrik yang tinggi dan penggunaan nutrisi yang tidak efisien [26]. Gambar sistem hidroponik *ebb and flow* dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Sistem Hidroponik EBB and Flow [20]

Pada Gambar 2.3 Ini adalah representasi sistem hidroponik *Ebb and Flow*. Sistem ini berfungsi dengan cara menyediakan tanaman dengan air, oksigen, dan nutrisi dari wadah penyimpanan, kemudian mengalirkannya ke media yang menopang akar tanaman, biasanya dalam pasangan. Seiring waktu berjalan, air bersama dengan nutrisi akan kembali mengalir ke wadah penyimpanan di bagian bawah sistem. Penggunaan timer memungkinkan pengaturan waktu untuk tinggi

dan rendahnya air sesuai kebutuhan tanaman, sehingga tanaman tidak kebanjiran atau terendam terlalu dalam oleh air. Sistem ini bekerja dalam dua tahap: fase air tinggi, di mana tanaman disiram dengan larutan nutrisi, dan fase air rendah, di mana tanaman tidak menerima nutrisi [27].

2.5. IoT pada Sistem *Smart Farming*

Menurut IEEE (*Institute of Electrical and Electronics Engineers*), *Internet of Things* (IoT) didefinisikan sebagai suatu jaringan di mana setiap objek dilengkapi dengan sensor yang terkoneksi ke internet. IoT, atau sering disingkat sebagai IoT, adalah serangkaian perangkat pintar yang mampu memonitor, berkomunikasi, dan menginterpretasi informasi dari sekitar mereka secara *real-time*. IoT yang dihasilkan mampu mengumpulkan data yang berharga serta mengoptimalkan sistem. IoT menggunakan pemrograman untuk memungkinkan interaksi antar mesin tanpa campur tangan manusia. Interaksi ini terus-menerus memanfaatkan konektivitas internet yang terhubung secara terus-menerus [28]. Konsep *internet of things* dapat dilihat pada Gambar 2.4.



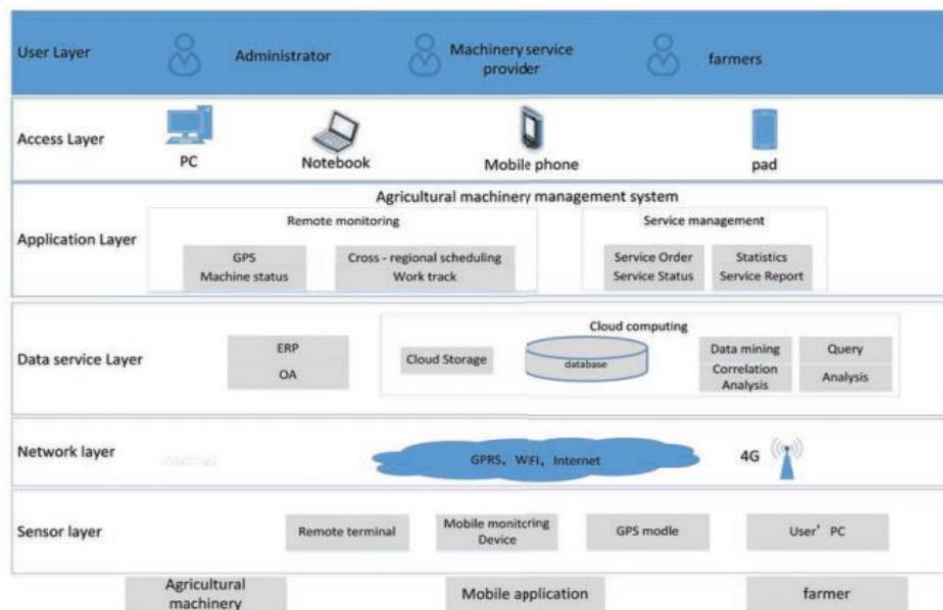
Gambar 2. 4 Konsep Internet of Things [29]

Pada Gambar 2.4 dapat dilihat gambar konsep *internet of things*. Dengan keberadaan teknologi seperti ini, kita dapat mengontrol perangkat-perangkat di sekitar kita, baik yang berada dekat maupun yang jauh. Prinsip kerja IoT adalah dengan menerjemahkan bahasa pemrograman yang telah dimasukkan ke dalam perangkat IoT itu sendiri. Perangkat tersebut umumnya disebut sebagai mikrokontroler. Setelah di-program, mikrokontroler tersebut harus terhubung dengan modul Wi-Fi sebagai akses ke jaringan internet, yang memungkinkan mikrokontroler terkoneksi dengan internet [29].

Penerapan *Internet of Things* (IoT) juga semakin luas di sektor pertanian. Ini memberikan banyak manfaat dalam mencatat atau memantau iklim mikro pertanian, baik itu secara langsung di lapangan (*on site*) maupun dari jarak jauh (*off site*). Konsep teknologi IoT diimplementasikan dengan penggunaan sensor dan aktuator, yang saling berinteraksi dan terhubung ke internet. Hal ini memungkinkan untuk pemantauan dan pengendalian dari jarak jauh. [30].

2.6. Sistem Komunikasi Data

Penggunaan *Internet of Things* (IoT) dalam pertanian melibatkan berbagai komponen, termasuk arsitektur, sensor, pengkodean, dan jaringan. Sistem IoT dalam pertanian umumnya terdiri dari 6 lapisan. Lapisan pertama adalah lapisan sensor yang bertugas untuk mengumpulkan informasi tentang lokasi, kondisi, dan kinerja mesin. Lapisan kedua adalah lapisan jaringan yang bertanggung jawab atas pengiriman informasi dari sensor ke layanan data menggunakan GPRS, Wi-Fi, Intranet, dan sistem komunikasi mobile. Lapisan ketiga adalah lapisan layanan data, di mana informasi yang terkumpul diolah dan disimpan dalam *cloud computing*. Lapisan keempat adalah lapisan aplikasi yang mengatur informasi yang tersimpan dalam layanan data. Lapisan kelima adalah lapisan akses, yang terdiri dari perangkat seperti PC, *notebook*, telepon genggam, dan perangkat sari pintar lainnya yang dapat menjalankan perangkat lunak aplikasi. Terakhir, lapisan *user* berfungsi sebagai pengguna yang mengakses dan menggunakan sistem ini [31]. Gambar arsitektur sistem IoT dapat dilihat pada Gambar 2.5.



Gambar 2. 5 Arsitektur Sistem IoT [32]

2.7. Sistem Keamanan

Sistem keamanan jaringan adalah proses yang bertujuan untuk mengidentifikasi dan mencegah akses pengguna yang tidak sah ke suatu jaringan komputer. Tujuan dari pengamanan sistem jaringan ini adalah untuk mengantisipasi risiko ancaman yang mencakup kerusakan bagian fisik komputer dan pencurian data [32]. Ancaman fisik mengacu pada tindakan yang merusak bagian fisik komputer atau hardware-nya, sedangkan ancaman logis adalah ancaman yang berupa pencurian data atau upaya penyusupan yang bertujuan untuk membobol akun seseorang.

Keamanan jaringan sangat bergantung pada kecepatan sistem jaringan dalam menanggapi gangguan yang terjadi. Untuk memperkuat keamanan jaringan komputer, penerapan sistem pendeteksi serangan dalam jaringan komputer dapat menjadi solusi. *Server*, sebagai elemen penting yang menyimpan *database*, aplikasi, dan layanan vital, memerlukan perlindungan yang kuat baik dari segi infrastruktur maupun aplikasi pendukungnya. Dengan demikian, diharapkan *server* terhindar dari gangguan yang dapat mengganggu kinerjanya sehingga pelayanan kepada klien dapat berfungsi secara optimal.

Keamanan jaringan komputer, sebagai bagian tak terpisahkan dari suatu sistem, memiliki peran yang sangat penting dalam memastikan kebenaran dan

keselamatan data, serta memastikan layanan tetap tersedia bagi pengguna. Serangan terhadap server jaringan komputer bisa terjadi tanpa memandang waktu, baik saat administrator sedang aktif maupun tidak. Oleh karena itu, keberadaan sistem keamanan di dalam server itu sendiri sangatlah krusial, dimana sistem tersebut mampu mendeteksi serangan dengan cepat dan langsung [33].

2.8. Metode *Waterfall*

Metode *Waterfall*, sering digambarkan seperti air terjun yang mengalir dari atas ke bawah, adalah model pengembangan perangkat lunak yang paling tradisional dan mudah dipahami. Metode ini membagi proses pengembangan menjadi beberapa tahapan yang berurutan dan terdefinisi dengan jelas, dengan satu tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya. Berikut ini merupakan tahapan-tahapan dalam Metode *Waterfall*:

1. Analisis Kebutuhan (*Requirement Analysis*)

Pada fase ini, pengembang perangkat lunak harus memahami seluruh informasi terkait kebutuhan perangkat lunak, seperti tujuan penggunaan perangkat lunak yang diinginkan dan batasan-batasan perangkat lunak. Informasi tersebut biasanya diperoleh melalui wawancara, survei, atau diskusi. Kemudian, informasi tersebut dianalisis untuk mendapatkan data yang lengkap mengenai kebutuhan pengguna terhadap perangkat lunak yang akan dikembangkan.

2. Desain (*Design*)

Langkah berikutnya adalah tahap Desain. Desain dilakukan sebelum proses pengkodean dimulai. Hal ini bertujuan untuk memberikan gambaran menyeluruh tentang apa yang perlu dilakukan dan bagaimana tampilan dari sistem yang diinginkan. Dengan demikian, tahap ini membantu dalam menentukan kebutuhan perangkat keras dan sistem, serta merumuskan arsitektur keseluruhan dari sistem yang akan dibuat.

3. Implementasi (*Implementation*)

Pada tahap ini, proses penulisan kode dilakukan. Pembuatan perangkat lunak akan dibagi menjadi modul-modul kecil yang nantinya akan digabungkan pada tahap selanjutnya. Selama tahap ini, pemeriksaan lebih lanjut terhadap

modul yang sudah dibuat juga dilakukan untuk memastikan apakah sudah memenuhi fungsi yang diinginkan atau belum.

4. Integrasi & Pengujian (*Integration & Testing*)

Pada tahap keempat ini, modul-modul yang sudah dibuat sebelumnya akan digabungkan. Selanjutnya, dilakukan pengujian untuk mengevaluasi apakah perangkat lunak sudah sesuai dengan desain yang diinginkan dan untuk mendeteksi kemungkinan kesalahan.

5. Pemeliharaan (*Maintenance*)

Maintenance merupakan tahapan terakhir dalam metode pengembangan *waterfall*. Pada tahapan ini, perangkat lunak yang sudah selesai dikembangkan akan dijalankan atau dioperasikan oleh pengguna [34].

2.9. Kajian Pustaka

Kajian pustaka merupakan sumber informasi hasil penelitian yang telah dilakukan yang berkaitan dengan topik “Rancang Bangun Aplikasi Sistem Monitoring Pada *Rooftop Greenhouse* Berbasis Web”. Berikut beberapa kajian pustaka yang menjadi acuan dalam penelitian yang akan dilakukan. Penelitian pertama membahas mengenai sistem digitalisasi *monitoring* dan *controlling* pada pertanian hidroponik menggunakan IoT MQTT Panel berbasis android. Pembuatan *software* dilakukan menggunakan aplikasi IoT MQTT Panel yang di-*download* melalui *google playstore*. Hasil pengujian aplikasi IoT MQTT Panel tidak terdapat *delay* saat menerima perintah menuju alat penyiram, data yang didapat secara *realtime*, dan sistem *monitoring* dan *controlling* tanaman hidroponik menggunakan aplikasi android berjalan dengan baik [35].

Penelitian kedua membahas mengenai rancang bangun sistem untuk pemantauan kondisi tanah, kontrol penyiraman, dan pemberian pupuk otomatis. Sistem *smart farming* berbasis IoT menggunakan aplikasi android pada halaman *home* dirancang untuk *monitoring* data, halaman aksi untuk *trigger* seperti penyiraman, pemberian pupuk, dan pengisian air tandon, halaman notifikasi untuk pemberitahuan pembaharuan data yang diterima. Hasil penelitian sesuai proses pembuatan sistem *smart farming*, dan aplikasi android bekerja dengan baik dalam *monitoring*, kontrol penyiraman, dan pemberian pupuk [36].

Penelitian ketiga membahas mengenai sistem untuk mengendalikan dan memonitor kondisi *greenhouse* dari jarak jauh. Pengujian *web server* dengan menjalankan SSH raspberry dan aplikasi yang telah dibuat. *Monitoring* dilakukan dengan membuka *browser* pada komputer atau ponsel dengan mengetik IP *address* Raspberry pada URL ditambahkan *port* 8080, setelah situs terbuka maka aplikasi akan menampilkan data sensor. Hasil penelitian bekerja sesuai perancangan, dan sistem *smart greenhouse* dapat mengirimkan data dengan benar ke sistem *interface* halaman situs untuk proses *monitoring* dari jarak jauh selama terhubung dengan *network* yang sama. Kekurangan penelitian ini yaitu akses halaman situs hanya melalui *network* yang sama sehingga belum terpasang IP *Address Public* dan *Dynamic DNS* (DDNS) pada router [6].

Penelitian keempat membahas mengenai penggunaan IoT pada *smart farming* untuk *me-monitoring* tanaman hidroponik. Penelitian ini menggunakan aplikasi *blynk* untuk *monitoring* tanaman. Aplikasi *blynk* menampilkan data *monitoring* keadaan alat dari jarak jauh, *blynk app* dapat di-*download* di *play store* dan terhubung dengan *database* untuk menyimpan data. Hasil penelitian seluruh kerja alat berjalan dengan baik dan aplikasi *blynk* menampilkan hasil *monitoring* dan kondisi relay dengan baik [37].

Penelitian kelima membahas mengenai perancangan sistem pemeliharaan tanaman aeroponik otomatis berbasis web. *Software* yang digunakan dalam penelitian ini yaitu sistem operasi windows 7 (64-bit), php strom, XAMPP for windows, PHP, MySql, Framework CodeIgniter, dan Google chrome. Hasil penelitian ini yaitu aplikasi sistem *monitoring* nutrisi berbasis web menggunakan IoT pada otomatisasi aeroponik, sistem dapat membantu dan mempermudah petani untuk *monitoring* nutrisi tanaman menggunakan *smartphone* dan akses halaman menggunakan web *browser* dengan informasi yang otomatis tersedia pada *dashboard website* dan data tersimpan dalam *database* [38].

Beberapa penelitian terdahulu yang sudah di jelaskan di atas , dapat dilihat juga pada Tabel 2.1 berikut.

Tabel 2.1 Kajian Pustaka

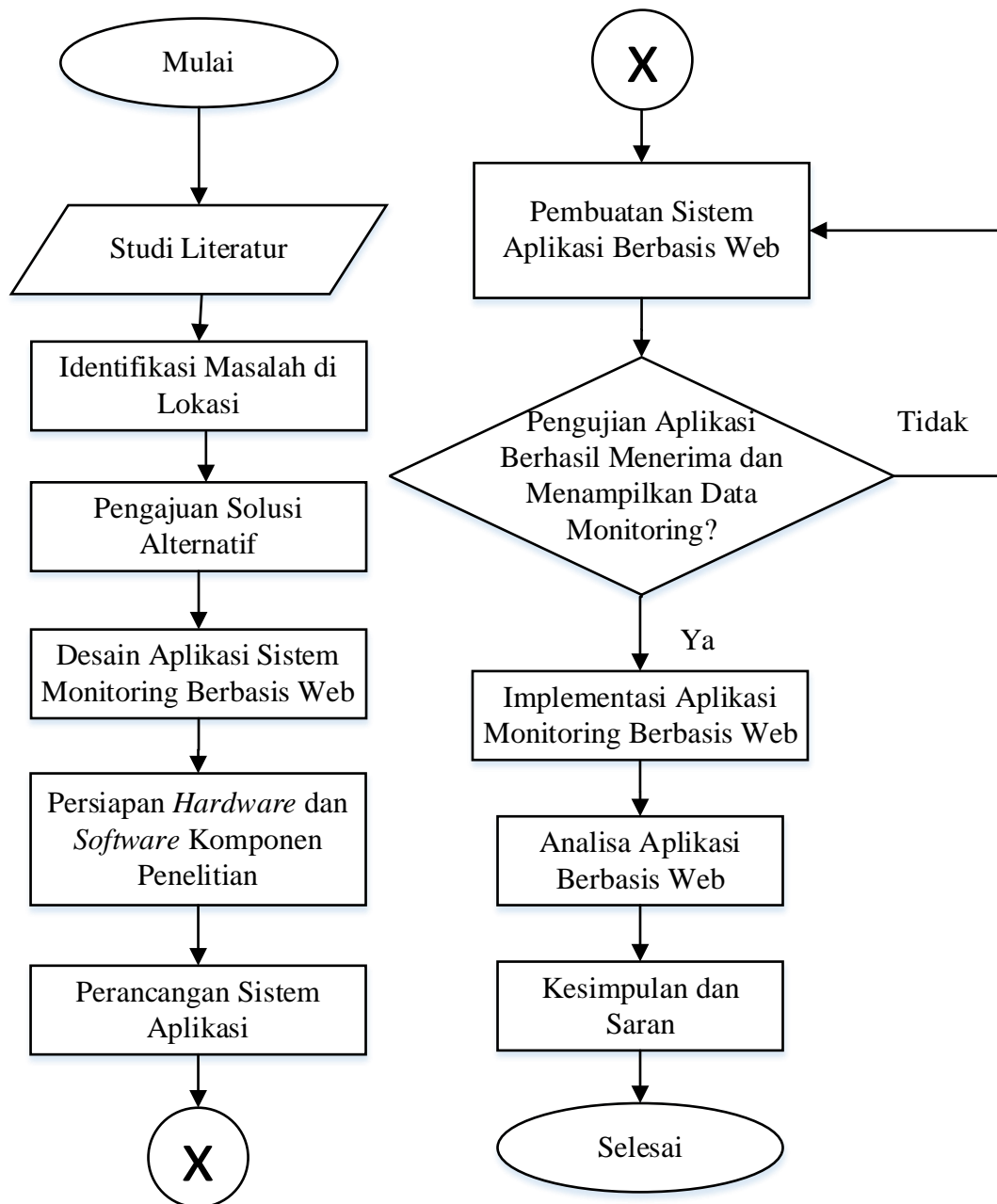
No	Judul	Penulis	Metode	Objek Penelitian	Hasil Penelitian
1	Smart Farming: Sistem Tanaman Hidroponik Terintegrasi IoT MQTT Panel Berbasis Android	I. Z. T. Dewi, M. F. Ulinuha, W. A. Mustofa, A. Kurniawan, and F. A. Rakhmadi	Pengembangan sistem hidroponik terintegrasi IoT MQTT Panel berbasis Android untuk memantau dan mengendalikan tanaman hidroponik secara real-time dan terkoneksi dengan internet.	Sistem hidroponik terintegrasi IoT MQTT Panel berbasis Android	Sistem hidroponik terintegrasi IoT MQTT Panel terbukti efektif dalam memantau dan mengendalikan tanaman hidroponik secara real-time dan terkoneksi dengan internet.
2	Rancang Bangun Sistem Smart Farming Berbasis IoT Studi Kasus Kebun Nyoman Gunitir	I. P. G. E. E. Kurnia and A. A. G. Ekayana	Pengembangan sistem smart farming berbasis IoT untuk memantau dan mengendalikan kondisi kebun Nyoman Gunitir secara real-time dan terkoneksi dengan internet.	Kebun Nyoman Gunitir	Sistem smart farming berbasis IoT terbukti efektif dalam memantau dan mengendalikan kondisi kebun Nyoman Gunitir secara real-time dan terkoneksi dengan internet.
3	Rancang Bangun Smart Greenhouse Berbasis Raspberry Pi dengan Web Framework Flask untuk Pertanian Perkotaan	Y. B. W. Yohanes Bowo Widodo, S. S. Sondang Sibuea, T. S. Tata Sutabri, and I. A. Ibrahim Aziz	Pengembangan sistem smart greenhouse berbasis Raspberry Pi dengan web framework Flask untuk memantau dan mengendalikan kondisi greenhouse secara real-time dan terkoneksi dengan internet.	Sistem smart greenhouse berbasis Raspberry Pi dengan web framework Flask	Sistem smart greenhouse berbasis Raspberry Pi dengan web framework Flask terbukti efektif dalam memantau dan mengendalikan kondisi greenhouse secara real-time dan terkoneksi dengan internet.
4	Implementasi Teknologi IoT pada Smart Farming dalam Memonitoring Tanaman Hidroponik Berbasis Android	R. S. Anwar, N. Agustina, and E. Sutinah	Implementasi teknologi IoT pada sistem smart farming untuk memantau kondisi tanaman hidroponik secara real-time dan terkoneksi dengan internet.	Sistem smart farming berbasis IoT dan aplikasi Android	Sistem smart farming berbasis IoT dan aplikasi Android terbukti efektif dalam memantau kondisi tanaman hidroponik secara real-time dan terkoneksi dengan internet.
5	Perancangan Aplikasi Berbasis Web Pada System	R. Y. Endra, A. Cucus, S. Wulandana,	Pengembangan aplikasi berbasis web pada sistem aeroponik untuk memantau dan	Sistem aeroponik dan	Aplikasi web berbasis framework CodeIgniter terbukti efektif dalam

No	Judul	Penulis	Metode	Objek Penelitian	Hasil Penelitian
	Aeroponik untuk Monitoring Nutrisi Menggunakan Framework CodeIgniter	and M. Aditya	mengendalikan nutrisi tanaman secara real-time dan terkoneksi dengan internet.	aplikasi web	memantau dan mengendalikan nutrisi tanaman aeroponik secara real-time dan terkoneksi dengan internet.

BAB III METODE PENELITIAN

3.1. Alur Penelitian

Pada penelitian yang akan dilakukan terdapat tahapan-tahapan yang ditampilkan dalam bentuk diagram alir. Pada Gambar 3.1 merupakan diagram alir penelitian.



Gambar 3.1 Diagram Alir Penelitian

Pada Gambar 3.1 merupakan gambar diagram alir penelitian tugas akhir yang akan dilakukan dengan judul “Rancang Bangun Aplikasi Monitoring Sistem Hidroponik *Ebb and Flow* Pada *Rooftop Greenhouse* Berbasis Web”. Dapat dilihat pada Gambar 3.1 urutan tahapan yang dilakukan dalam penelitian ini, dimana pada penelitian ini diawali dengan tahapan studi literatur. Studi literatur ini merupakan awal tahapan penelitian karena membantu penulis mendapatkan gambaran terkait penelitian yang akan dilakukan dengan mengacu dari penelitian-penelitian sebelumnya.

Identifikasi masalah membahas permasalahan yang ada di lokasi penelitian. Solusi alternatif untuk masalah hasil identifikasi sebelumnya dan dilanjutkan dengan desain sistem. Persiapan komponen penelitian, dimana komponen yang dimaksud disini meliputi *hardware* maupun *software* yang digunakan sebagai pendukung dalam penelitian ini.

Perancangan sistem aplikasi, dimana pada tahapan ini dilakukan perancangan aplikasi yang akan dibuat. Perancangan sistem aplikasi dirumuskan konsep dan gambaran mengenai aplikasi yang akan di buat berdasarkan kebutuhan pengguna. Pembuatan aplikasi berisi gambaran aplikasi yang sudah dirumuskan, dibangun dan dikembangkan sehingga didapatkan aplikasi yang sesuai dengan kebutuhan pengguna.

Pengujian aplikasi berisi pengujian (*testing*) aplikasi, apakah sudah sesuai dengan kebutuhan yang diinginkan dengan pengguna. Apabila aplikasi sudah sesuai, bisa lanjut ke tahapan selanjutnya dan apabila masih belum sesuai, kembali ke tahap sebelumnya. Implementasi merupakan bentuk aplikasi yang dirancang diterapkan langsung di lapangan, tepatnya pada *greenhouse*. Aplikasi yang sudah dirancang digunakan sebagai media penampil data *monitoring* dari parameter yang digunakan pada *greenhouse*.

Analisis aplikasi merupakan bentuk aplikasi yang sudah diimplementasikan di *greenhouse* dalam rentang waktu tertentu, dianalisis kinerjanya serta kekurangan-kekurangan yang ditemukan selama penerapan aplikasi dalam sistem *monitoring*. Ditarik kesimpulan dari penelitian yang dilakukan serta dirumuskannya saran untuk penelitian serupa dikemudian hari.

3.2. Survei Lokasi Penelitian

Untuk menentukan lokasi penelitian yang sesuai, dilakukan beberapa kali kunjungan pada *greenhouse* Fakultas Pertanian Universitas Sultan Ageng Tirtayasa. Survei yang dilakukan dengan mengunjungi lokasi dan juga wawancara secara langsung dengan pengguna *greenhouse* tersebut guna mengetahui kondisi aktualnya. Dalam survei tersebut juga dilakukan identifikasi masalah serta permohonan perizinan penelitian yang akan dilakukan di lokasi tersebut.

Penelitian ini dilakukan di *Greenhouse* Laboratorium Agroekoteknologi Fakultas Pertanian kampus A (Pakupatan) Universitas Sultan Ageng Tirtayasa. Gambar *greenhouse* dapat dilihat pada Gambar 3.2.

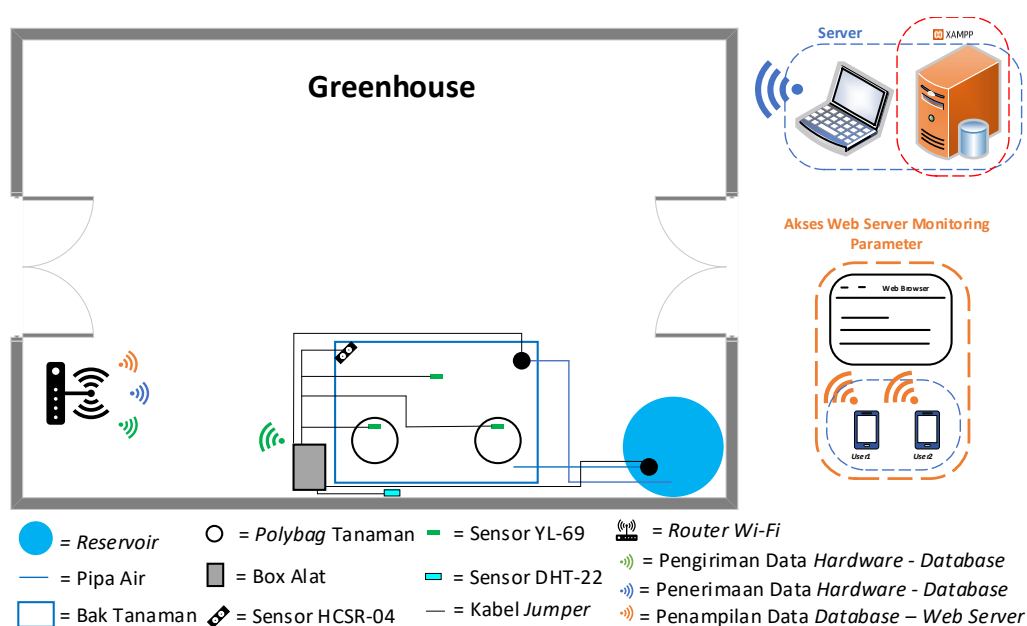


Gambar 3. 2 *Greenhouse* Laboratorium Agroekoteknologi Fakultas Pertanian
Untirta

Dapat dilihat pada gambar 3.2 merupakan gambar *greenhouse* Laboratorium Agroekoteknologi Fakultas Pertanian Untirta. *Greenhouse* ini memiliki dimensi ruangan dengan panjang 9,1m, lebar ruangan 6m dan ketinggian dari dasar ke atap 2,15m. *Greenhouse* ini digunakan sebagai lokasi aklimatisasi bibit tanaman buah yang sudah melalui tahap kultur jaringan.

3.3. Desain Sistem *Monitoring* Berbasis IoT

Sistem *monitoring* berbasis IoT ini tersusun dari beberapa komponen yang terintegrasi dengan koneksi internet. Setiap komponen memiliki peran khusus dalam mengumpulkan dan mentransmisikan data ke platform monitoring. Pada Gambar 3.3, tergambar dengan jelas skema keseluruhan sistem monitoring berbasis IoT ini, memberikan gambaran visual yang memudahkan untuk memahami bagaimana setiap komponen saling terhubung dan berinteraksi dalam pengumpulan data.



Gambar 3.3 Skema Sistem Monitoring Berbasis IoT

Pada Gambar 3.3 dapat dilihat skema sistem *monitoring* parameter berbasis iot. Pada gambar tersebut terdiri dari beberapa komponen penyusun yang saling terintegrasi dengan koneksi internet. Sistem tersebut dapat bekerja dengan optimal apabila koneksi internet terhubung dengan sistem monitoring (mikrokontroler), *server database*, serta *web server*.

Pertama-tama sistem *monitoring* mengukur kondisi di lapangan menggunakan sensor yang terhubung dengan mikrokontroler. Setelah didapatkan hasil pengukuran sensor, mikrokontroler mengirimkan data hasil pengukuran kepada *database* menggunakan koneksi internet. Selanjutnya data pada *database* di-upload pada server, untuk ditampilkan pada website, sehingga saat *user* membutuhkan informasi kondisi di lapangan, cukup dengan mengakses web tersebut.

3.3.1. Pengajuan Solusi Alternatif

Terkait dengan permasalahan yang ada pada sub-bab 3.2.1, maka diajukan solusi untuk menangani permasalahan yang ada pada *greenhouse* tersebut. Terdapat beberapa alternatif solusi yang ditawarkan sebagai berikut:

1. Pemantauan dan pengendalian parameter bibit tanaman secara konvensional, yaitu dilakukan pemantauan secara manual saat pagi dan sore hari. Pemantauan dilakukan secara langsung oleh pengelola *greenhouse* tersebut dengan memantau nilai parameter *greenhouse* dan melihat kondisi bibit tanaman, apabila kondisi tidak sesuai maka dilakukan pengendalian seperti penyiraman pada bibit tanaman dengan kondisi tanah kering. Cara konvensional ini dinilai kurang efektif karena memiliki beberapa kekurangan, diantaranya;
 - a. dibutuhkan tenaga manusia melakukan pemantauan di *greenhouse*
 - b. membutuhkan cukup banyak usaha untuk menjangkau *greenhouse* tersebut
 - c. pemantauan terbatas oleh jam operasional gedung
 - d. tidak dapat dilakukannya pengendalian parameter suhu dan kelembaban udara.
2. Pada penelitian yang dilakukan oleh Ahmad Thoriq, dkk. yaitu dengan merancang sistem monitoring suhu dan kelembaban tanah berbasis IoT. Dalam penelitiannya menggunakan mikrokontroler NodeMCU ESP-32 Dev Kit V1.0, sensor *soil moisture* untuk kelembaban tanah, DHT22 untuk suhu dan kelembaban udara, LCD untuk pemantauan langsung dan aplikasi android untuk memantau parameter suhu dan kelembaban. Penelitian dilakukan di perkebunan ubi cilembu di Desa Cilembu, Jawa Barat dengan menancapkan sensor kelembaban tanah pada tanah dan sensor suhu dan kelembaban udara terletak pada badan alat. Hasil pembacaan sensor disimpan di database dan data pembacaan dapat dilihat melalui aplikasi android. Dengan cara ini sudah cukup efektif mengatasi beberapa masalah sebelumnya, tetapi cara ini dinilai tetap memiliki kekurangan. Dengan cara ini pengendalian dan pemantauan *greenhouse* sudah dilakukan dengan otomatis, tetapi untuk pemantauan

menggunakan aplikasi android dinilai kurang efektif untuk digunakan secara instan, karena pengguna harus mengunduh dan meng-*instal* aplikasi tersebut.

3. Merancang sistem monitoring berbasis IoT untuk suhu lingkungan, kelembaban tanah, dan kelembaban udara untuk bibit tanaman buah di dalam *greenhouse*. Sistem ini menampilkan hasil monitoring melalui *interface* pada aplikasi berbasis website yang dapat dipantau secara jarak jauh karena letak *greenhouse* yang berada di *rooftop* gedung 3 lantai. Saat nilai parameter tidak sesuai dengan kebutuhan ideal suatu tanaman maka sistem kendali akan mengaktifkan aktuator untuk mengendalikan nilai parameter secara otomatis. Perancangan sistem dengan menggunakan sensor yang berfungsi untuk mengukur nilai parameter yang dibutuhkan dalam *greenhouse*. Kemudian data yang dibaca oleh sensor diproses dan diolah oleh mikrokontroler. Aplikasi berbasis website akan menampilkan hasil pembacaan nilai parameter yang terbaca oleh sensor. Dengan adanya penggunaan aplikasi berbasis website baik dosen maupun mahasiswa dapat melakukan monitoring parameter bibit tanaman buah secara jarak jauh tanpa harus melakukan pemantauan langsung ke *greenhouse*.

Dari ketiga alternatif solusi yang diberikan, solusi yang cocok untuk monitoring parameter suhu lingkungan, kelembaban tanah, dan kelembaban udara dalam *greenhouse* yaitu alternatif solusi ketiga. Karena melihat kondisi lokasi *greenhouse* yang berada di *rooftop* gedung 3 lantai dan jam operasional gedung yang terbatas yaitu hanya dijam kerja saja. Sehingga dosen maupun mahasiswa dapat memantau parameter secara *real-time* dari jarak jauh melalui aplikasi berbasis website tanpa harus ke *greenhouse*, dan tanpa harus mengunduh serta meng-*instal* aplikasi tersebut.

3.4. Komponen Penelitian

Pada penelitian ini dibutuhkan beberapa komponen penting yang menunjang proses perancangan dan pengembangan sistem *monitoring* berbasis IoT serta aplikasi sistem monitoring berbasis *web*. Pada Tabel 3.1 dapat dilihat komponen yang akan digunakan yang terdiri dari perangkat lunak (*software*) dan perangkat keras (*hardware*).

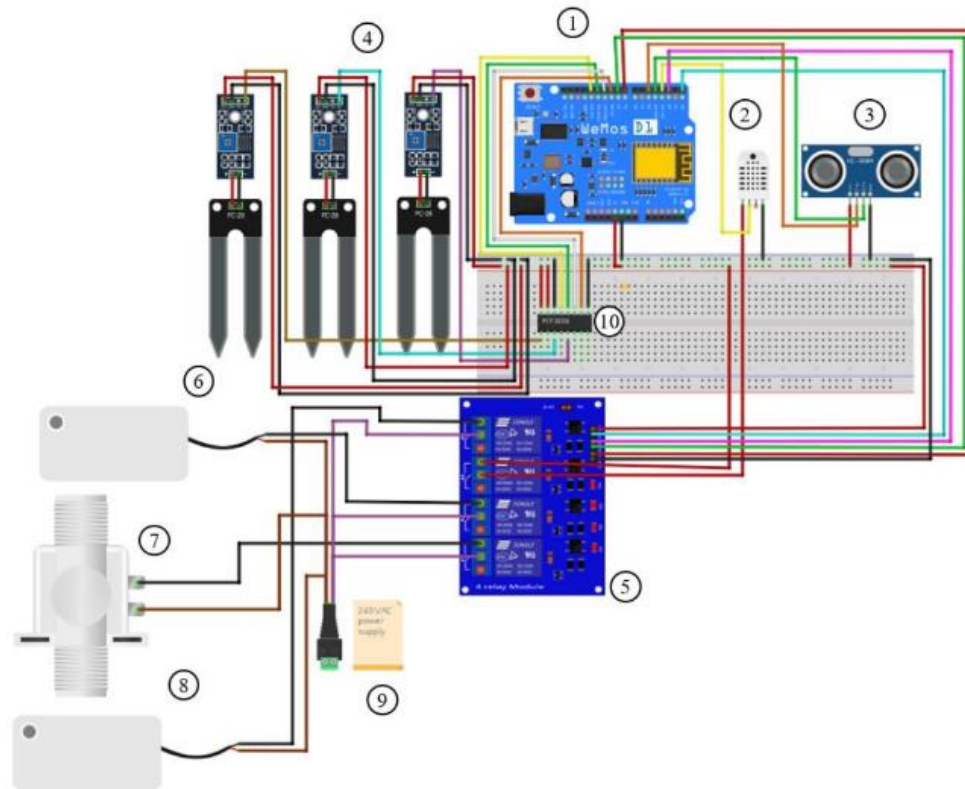
Tabel 3.1 Komponen Alat Sistem *Monitoring* Berbasis IoT

Komponen Pembuatan Alat	
Software	Hardware
Arduino IDE	Arduino Wemos D1 R1
	Sensor DHT22
	Sensor YL-69
	Pompa air dc
	<i>Solenoid valve ac</i>
	Sensor HC-SR04
	<i>Relay</i>
	Pompa <i>misting</i>
	Kabel <i>jumper</i>
	Adapter DC

Dapat dilihat pada Tabel 3.1 merupakan komponen dalam perancangan alat sistem *monitoring* berbasis IoT. Pada Tabel 3.1 komponen yang digunakan dalam perancangan alat sistem *monitoring* ini lebih banyak berupa *hardware*, mulai dari mikrokontroler, sensor, aktuator, dll, sedangkan software yang digunakan hanya ArduinoIDE.

3.5. Skematik Alat

Sub bab ini membahas perancangan *hardware* yang digunakan agar tiap masing-masing komponen bergabung menjadi satu kesatuan sistem yang dapat bekerja dan terkendali. Komponen-komponen mulai dari sensor hingga aktuator dihubungkan melalui kabel-kabel menuju pin mikrokontroler. Dengan mikrokontroler Arduino WEMOS D1R1 diprogram sesuai dengan rancangan dan kebutuhan yang diinginkan. Dalam mendesain skematik rangkaian digunakan *software* Fritzing untuk menampilkan secara dua dimensi bentuk alat dan hubungan masing-masing komponen ke mikrokontroler Arduino WEMOS D1R1. Gambar skematik rangkaian alat dapat dilihat pada Gambar 3.4.



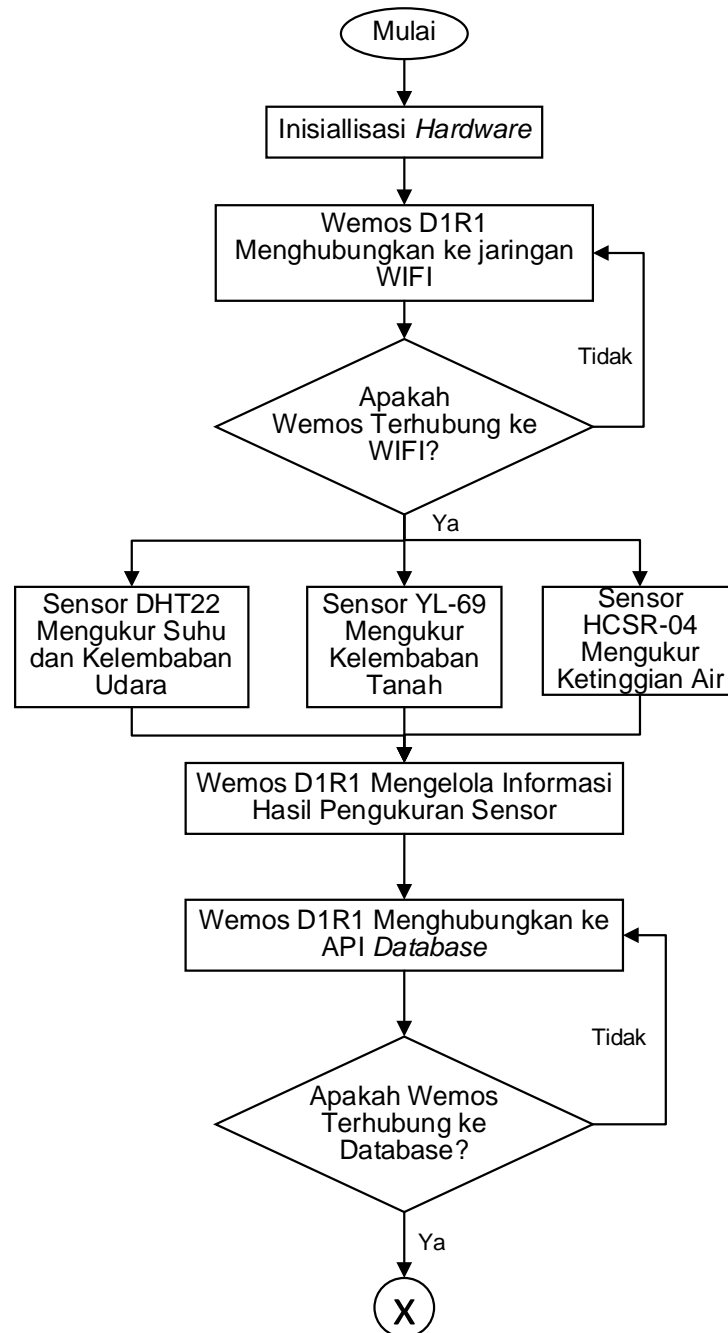
Gambar 3.4 Skematik Rangkaian Alat *Smart Monitoring*

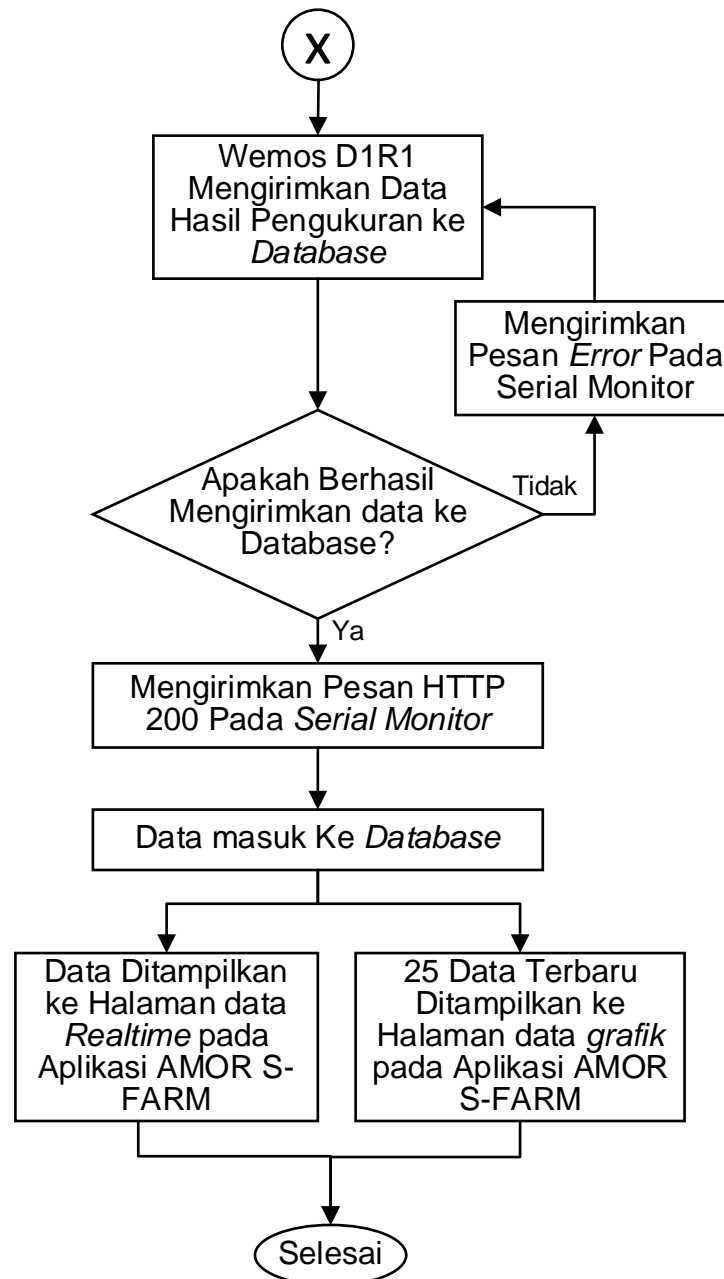
Pada Gambar 3.4 merupakan gambar skematik rangkaian alat *smart monitoring*. Pada Gambar 3.4 dapat dilihat seluruh komponen yang digunakan dihubungkan pada mikrokontroler Arduino WEMOS D1R1. Komponen yang digunakan sebagai berikut:

1. Arduino Wemos D1 R1 (Mikrokontroler)
2. Sensor DHT22 (Sensor suhu dan kelembaban udara)
3. Sensor Ultrasonic HC-SR04 (Sensor jarak)
4. Sensor Soil Moisture YL-69 (Sensor kelembaban tanah)
5. Relay 4 Channel
6. Pompa air (*Water pump*)
7. Katup Selonoid
8. Pompa Kuras (*Drain pump*)
9. Sumber Tegangan
10. MCP3008

3.6. Flowchart Cara Kerja Sistem

Sub bab ini membahas secara detail mengenai *flowchart* cara kerja sistem yang telah dirancang. *Flowchart* ini menggambarkan tahapan-tahapan dalam sistem, serta bagaimana setiap komponen berinteraksi satu sama lain, serta bagaimana setiap elemen sistem saling berhubungan untuk mencapai tujuan akhir. Flowchart cara kerja sistem dapat dilihat pada Gambar 3.5.





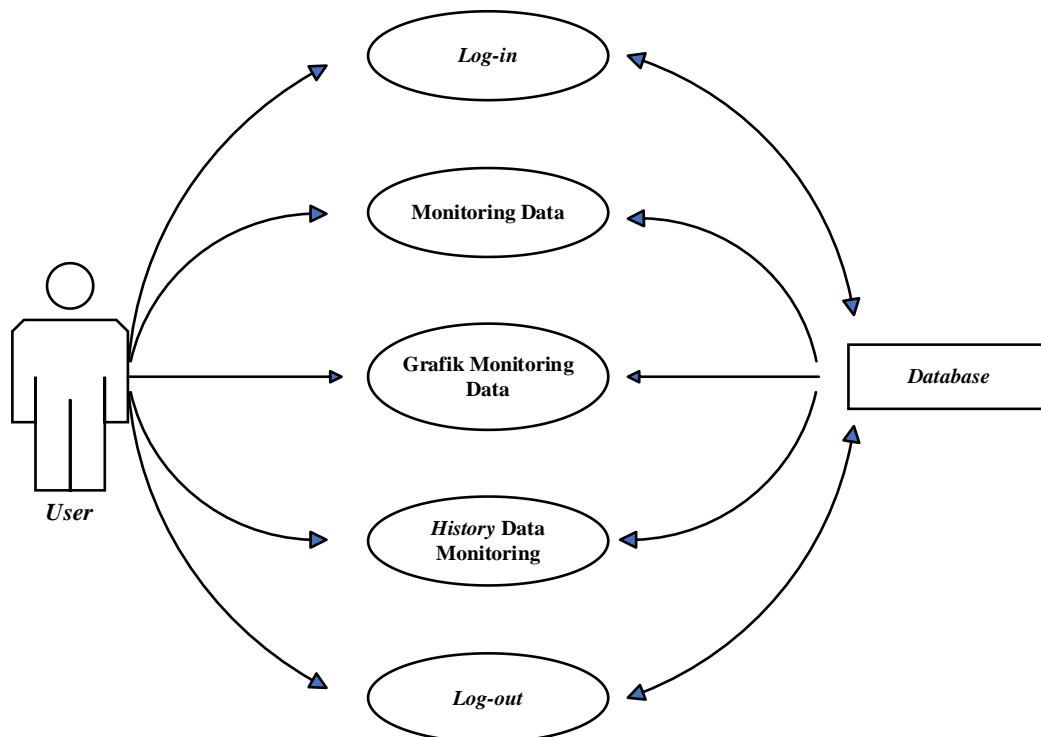
Gambar 3.5 *Flowchart* Cara Kerja Sistem

Pada Gambar 3.5 dapat dilihat gambar *flowchart* cara kerja sistem secara keseluruhan. Pada Gambar 3.5 menjelaskan bagaimana sistem bekerja dari hardware menghubungkan ke jaringan *wifi*, dilanjutkan dengan pengukuran parameter dengan menggunakan sensor. Setelah data berhasil terukur, wemos D1R1 sebagai mikrokontroler mencoba terhubung dengan *database* dan dilanjutkan

dengan proses pengiriman data ke *database*. Setelah data berhasil terkirim ke *database*, data di tampilkan pada halaman aplikasi berbasis web, AMOR S-FARM.

3.7. Usecase Diagram Aplikasi Web

Use case merupakan gambaran skenario dari interaksi antara *user* dengan sistem. Sebuah diagram *use case* menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi. Diagram *usecase* pada aplikasi sistem *monitoring smart farming* dapat di lihat pada Gambar 3.6 .

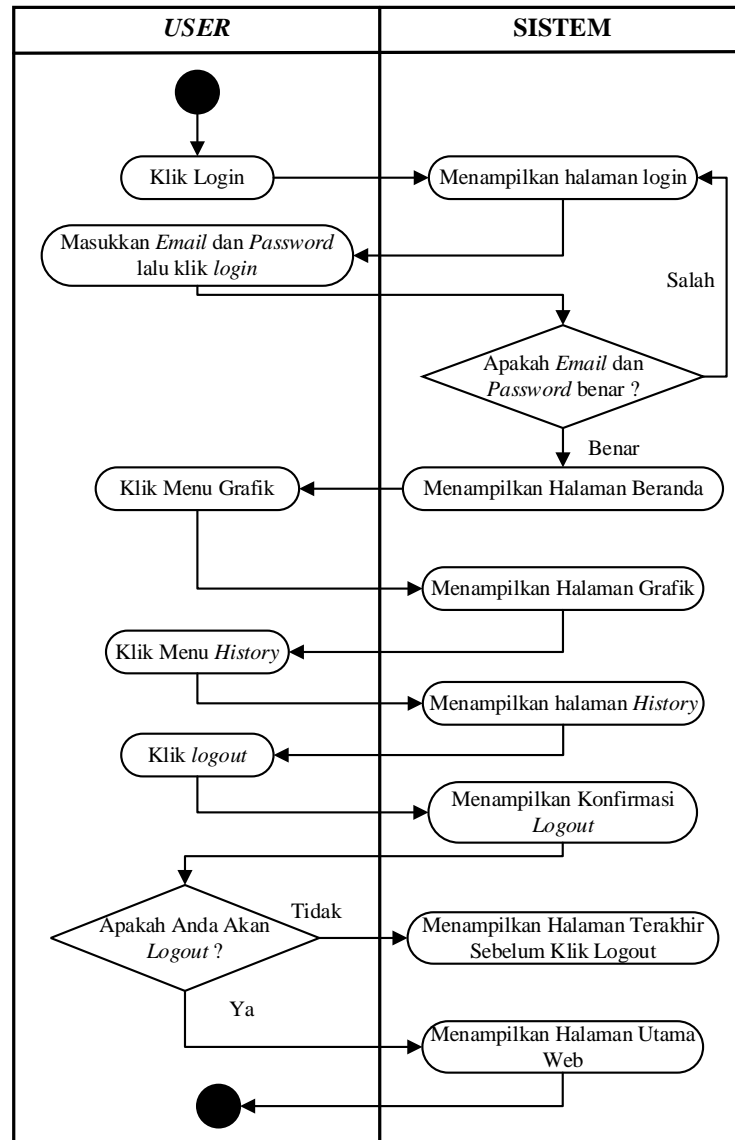


Gambar 3.6 Diagram *Use Case* Aplikasi Sistem Monitoring

Pada Gambar 3.6 merupakan gambar diagram *use case* aplikasi sistem *monitoring smart farming*. Dapat di lihat pada Gambar 3.8 *user* sebagai aktor berinteraksi dengan aplikasi. Kegiatan yang dilakukan aktor pada aplikasi mulai dari *log in* hingga *log out* melibatkan interaksi dengan *database*.

3.8. Activity Diagram

Activity diagram menggambarkan aktifitas-aktifitas yang terjadi dalam sistem dan *user*. Berdasarkan dari narasi *use case*



Gambar 3.7 Activity Diagram User Menggunakan Aplikasi Sistem Monitoring

Pada Gambar 3.7 merupakan gambar *Activity diagram user* menggunakan aplikasi sistem *monitoring*. Pada Gambar 3.7 tergambar seorang *user* melakukan *login* ke sistem, apabila berhasil login, sistem akan menampilkan halaman beranda yang dimana pada halaman ini ditampilkan data *real-time* hasil pengukuran sensor di lapangan. Selanjutnya *user* dapat melihat grafik data hasil pengukuran dengan memilih menu grafik, yang dimana sistem akan beralih pada halaman menu grafik. *User* juga dapat melihat *history* data yang ada dengan memilih menu *history* dan sistem akan mengalihkan ke halaman *history*. Selanjutnya jika *user* akan keluar dari sistem, maka sistem akan menampilkan

menu konfirmasi apakah user ingin keluar dari sistem, apabila user benar ingin keluar, maka sistem akan kembali ke halaman utama web.

3.9.Desain Sistem Aplikasi Web

Berdasarkan masalah yang telah dipaparkan pada bagian sebelumnya, maka dirancang desain aplikasi web untuk menampilkan hasil pembacaan sensor dan kondisi aktuator dalam bentuk angka, grafik dan status teks. Pada Gambar 3.8 merupakan desain halaman beranda pada aplikasi sistem monitoring.



Gambar 3.8 Desain Halaman Beranda Aplikasi Sistem Monitoring

Pada Gambar 3.8 dapat dilihat desain tampilan halaman beranda yang akan diimplementasikan pada aplikasi sistem monitoring pada halaman utama ini terdapat 3 data parameter yang diukur. Pada baris pertama akan ditampilkan nilai suhu lingkungan dan kelembaban udara. Pada baris ke dua akan ditampilkan nilai kelembaban tanah.

Pada aplikasi sistem monitoring ini juga terdapat beberapa menu yang terletak di sebelah kiri halaman. Menu tersebut terdiri dari:

- a. Beranda, pada menu ini sistem menampilkan data *real-time* hasil pengukuran sensor di lapangan.
- b. Grafik, pada menu ini sistem menampilkan grafik data hasil pengukuran beberapa waktu terakhir.

- c. *History*, pada menu ini sistem menampilkan data hasil pengukuran sebelumnya, yang ditampilkan dalam bentuk tabel.
- d. Selanjutnya setelah pengguna selesai menggunakan aplikasi ini, pengguna dapat melakukan *log out*. Menu *log out* berada di sebelah kanan atas halaman web.

BAB IV HASIL DAN PEMBAHASAN

4.1. Hasil Aplikasi AMOR S-FARM

Bagian ini menguraikan hasil rancangan aplikasi yang telah dikembangkan dengan, menggabungkan analisis tentang kebutuhan pengguna yang telah ditetapkan sebelumnya. Aplikasi ini diberi nama AMOR S-FARM yang merupakan singkatan dari "Aplikasi Monitoring Sistem *Smart Farming*", yang menunjukkan fokusnya pada monitoring pada sistem pertanian pintar. Logo aplikasi AMOR S-FARM dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 Logo AMOR S-FARM

Pada Gambar 4.1 dapat dilihat logo dari aplikasi AMOR S-FARM. Rancangan aplikasi ini mencakup berbagai aspek yang penting untuk keberhasilan dan efektivitas aplikasi. Pertama-tama, berbagai fitur yang akan ditawarkan oleh AMOR S-FARM telah ditetapkan, yang meliputi fungsi-fungsi yang penting untuk memantau dan mengelola sistem pertanian secara efisien. Desain antarmuka pengguna juga menjadi fokus utama, dengan tujuan untuk memberikan pengalaman yang intuitif, menarik, dan mudah digunakan bagi pengguna. Penggunaan teknologi yang tepat juga dipertimbangkan dengan cermat, memilih solusi teknis yang sesuai dengan kebutuhan dan tujuan aplikasi. Selain itu, alur kerja aplikasi, yang

menggambarkan urutan langkah-langkah yang harus diikuti oleh pengguna untuk mencapai tujuan mereka. Langkah selanjutnya adalah melanjutkan implementasi berdasarkan rancangan ini, dengan harapan dapat memberikan dampak positif yang signifikan dalam konteks pertanian modern.

4.1.1. Teknologi *Hosting web* AMOR S-FARM

Pada tahap implementasi, aplikasi AMOR S-FARM dikembangkan dengan menggunakan metode pendekatan *waterfall*, sesuai dengan yang dijelaskan dalam sub-bab 3.6. Pendekatan ini dipilih karena memberikan struktur yang jelas dan terstruktur dalam proses pengembangan, memungkinkan untuk mengikuti tahap-tahap yang ditentukan dengan hati-hati mulai dari analisis hingga implementasi.

Setelah selesai dibangun, AMOR S-FARM diluncurkan secara *online*, membuatnya dapat diakses melalui alamat www.amor.my.id. Layanan *hosting* yang digunakan adalah Niagahoster, yang dipilih karena kehandalannya dalam menyediakan layanan *hosting* yang cepat, aman, dan andal. Dengan demikian, pengguna dapat mengakses aplikasi dengan mudah dan nyaman melalui internet.

Untuk penyimpanan data, AMOR S-FARM menggunakan *database SQL* yang disediakan oleh penyedia layanan *hosting*, diakses melalui PHP My Admin. Pendekatan ini memastikan bahwa data pengguna tersimpan dengan aman dan dapat diakses dengan cepat dan efisien oleh aplikasi.

Dengan menggunakan metode pengembangan *waterfall*, layanan *hosting* Niagahoster, dan *database SQL*, AMOR S-FARM telah berhasil diimplementasikan dengan efisien dan efektif. Aplikasi ini siap digunakan untuk membantu petani dan praktisi pertanian dalam memantau dan mengelola sistem pertanian secara lebih cerdas dan efisien.

4.1.2. Fitur Utama

AMOR S-FARM dirancang dengan cermat, memperhatikan kebutuhan pengguna, dan menyediakan fitur-fitur unggulan untuk memberikan pengalaman yang luar biasa dalam pertanian modern. Aplikasi ini menawarkan solusi efektif dengan serangkaian fitur utama yang komprehensif, dirancang untuk memenuhi tuntutan mendalam dalam dunia pertanian saat ini.

1. Kondisi *Realtime*

Salah satu fitur utama AMOR S-FARM adalah kemampuannya untuk menyajikan informasi kondisi lapangan secara *realtime*. Ini memungkinkan petani untuk memantau parameter-parameter penting, seperti suhu, kelembaban, dan keadaan tanah, secara langsung dan akurat. Dengan informasi yang diperbarui secara terus-menerus, petani dapat mengambil keputusan yang tepat waktu untuk menjaga keadaan tanaman dan meningkatkan hasil panen.

2. Data dalam Bentuk Angka dan Grafik

Aplikasi ini menyediakan fleksibilitas untuk menampilkan data dalam berbagai format, baik dalam bentuk angka maupun grafik. Ini memungkinkan petani untuk memahami tren dan pola dengan lebih baik, serta mengidentifikasi anomali atau perubahan yang signifikan dalam kondisi pertanian. Dengan visualisasi data yang mudah dipahami, pengguna dapat dengan cepat menganalisis informasi dan mengambil tindakan yang diperlukan.

3. Data Dapat Diunduh

AMOR S-FARM memungkinkan pengguna untuk mengunduh data secara langsung dari aplikasi. Ini memberikan fleksibilitas kepada petani untuk menyimpan data secara lokal atau membagikannya dengan pihak lain, seperti penasihat pertanian atau instansi yang berwenang. Kemampuan untuk mengakses data di luar aplikasi memperluas penggunaan dan nilai aplikasi bagi pengguna.

4. *User Friendly*

Fitur-fitur utama yang ditawarkan oleh AMOR S-FARM dirancang dengan mempertimbangkan kebutuhan dan keterbatasan pengguna dalam lingkungan pertanian. Antarmuka pengguna yang ramah pengguna memastikan bahwa aplikasi dapat digunakan dengan mudah oleh petani dengan berbagai tingkat keahlian teknologi. Dengan navigasi yang intuitif dan instruksi yang jelas, pengguna dapat dengan cepat memanfaatkan fungsionalitas aplikasi tanpa kesulitan.

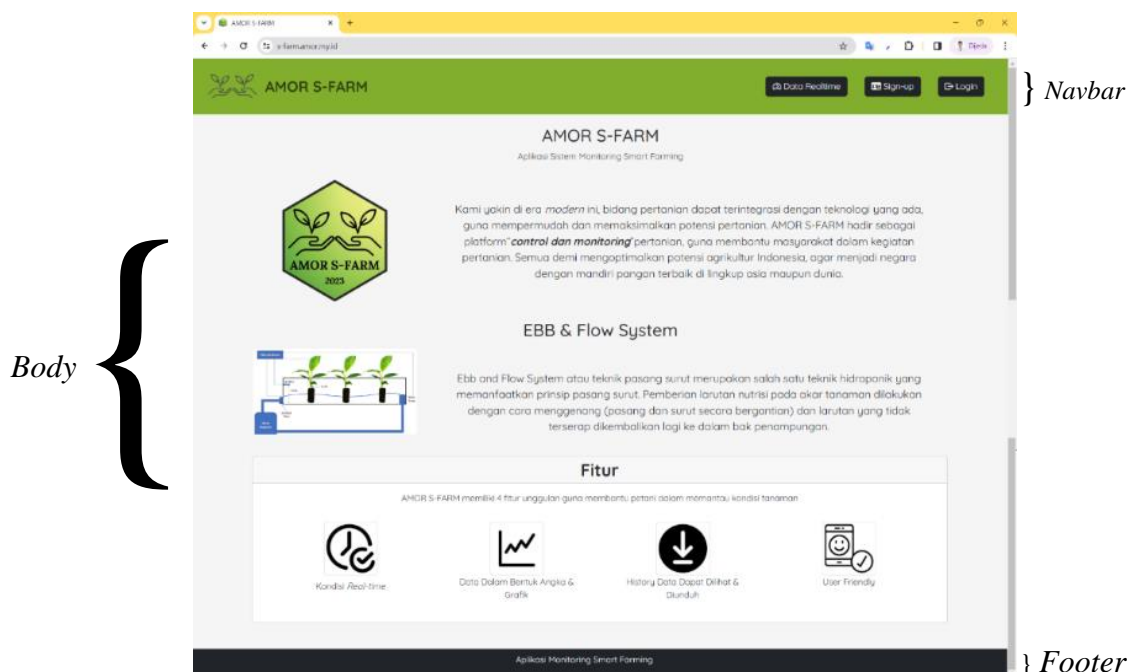
Dengan menyediakan fitur-fitur utama ini, AMOR S-FARM bertujuan untuk meningkatkan produktivitas, efisiensi, dan nilai tambah bagi pengguna, serta

membantu memajukan sektor pertanian menuju keberlanjutan dan keberhasilan yang lebih besar.

4.1.3. Hasil Antarmuka Pengguna (*User Interface Design*)

Dalam proses perancangan aplikasi, desain antarmuka pengguna merupakan aspek yang sangat diperhatikan agar memberikan pengalaman yang memikat dan memuaskan bagi pengguna. Desain antarmuka pengguna merupakan pintu gerbang pertama bagi pengguna untuk berinteraksi dengan aplikasi. Desain antarmuka pengguna pada AMOR S-FARM dirancang untuk dapat beradaptasi dengan berbagai perangkat dan ukuran layar, mulai dari *desktop* hingga perangkat *mobile*.

Tampilan AMOR S-FARM versi *desktop* dapat dilihat pada Gambar 4.2 dan versi *mobile* pada Gambar 4.3. Hal ini memungkinkan pengguna untuk mengakses aplikasi dengan nyaman di mana pun mereka berada, tanpa mengorbankan pengalaman pengguna yang optimal. AMOR S-FARM memastikan aksesibilitas yang mudah dengan menyajikan versi *desktop* dan *mobile*. Pengguna dapat dengan nyaman menggunakan aplikasi di berbagai perangkat, tanpa kehilangan pengalaman pengguna yang optimal. Visualisasi antarmuka mode *desktop* dapat dilihat pada Gambar 4.2, memberikan kemudahan navigasi dan fungsionalitas yang diperlukan bagi pengguna.



Gambar 4.2 Halaman Awal Aplikasi AMOR S-FARM Versi Desktop

Pada Gambar 4.2 merupakan *interface* awal yang akan muncul saat menggunakan *website* dalam bentuk *desktop* pada laptop atau komputer. Dalam desain antarmuka pengguna setiap detail harus diperhatikan, mulai dari pemilihan warna dan tata letak hingga navigasi dan animasi. Selain itu, penyusunan alur navigasi yang intuitif dan logis, memastikan bahwa pengguna dapat dengan mudah menavigasi antara halaman dan fitur aplikasi tanpa kebingungan. Konsistensi desain di seluruh aplikasi ini juga diperhatikan, sehingga pengguna dapat merasakan kesinambungan dan koherensi saat berinteraksi dengan berbagai bagian dari aplikasi. Gambar 4.3 untuk *interface* aplikasi dalam versi mobile.



Gambar 4.3 Hasil *Interface* Halaman Awal Aplikasi AMOR S-FARM Versi Mobile

Pada Gambar 4.3 merupakan *interface* awal yang akan muncul saat membuka aplikasi *website* pada *smartphone*. Dengan melalui desain antarmuka pengguna yang teliti dan berfokus pada pengalaman pengguna, dapat memberikan pengalaman yang memukau dan memuaskan kepada pengguna aplikasi tersebut. Desain antarmuka pengguna yang kuat bukan hanya tentang estetika visual, tetapi juga tentang memahami kebutuhan dan preferensi pengguna, dan itulah yang menjadi fokus utama dalam perancangan *user interface design* pada aplikasi ini.

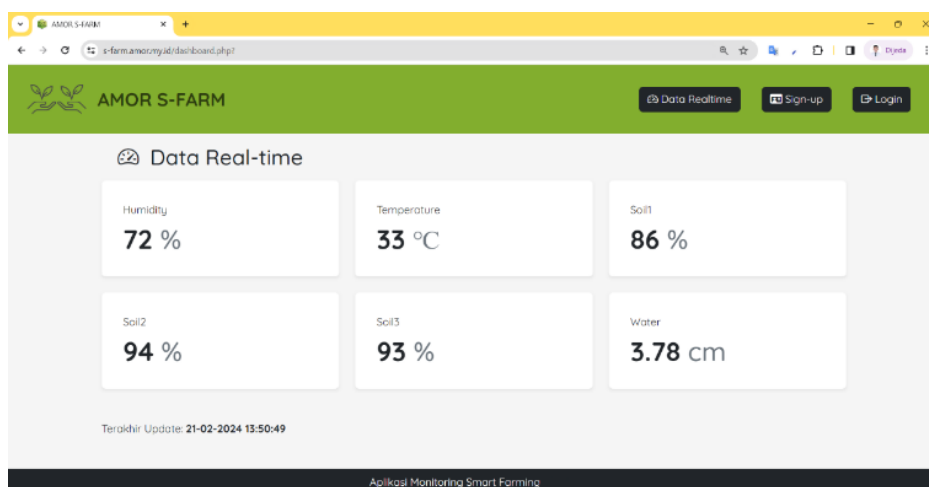
Berdasarkan Gambar 4.2 dapat dilihat Desain *interface* pada AMOR S-FARM dibagi menjadi menjadi 3 bagian, yaitu; *navbar*, *body*, dan *footer*. *Navbar* berisi menu navigasi pada tiap halaman, *body* berisi konten yang disajikan pada suatu halaman, dan *footer* berisi informasi aplikasi web “Aplikasi Moitoring Sistem *Smart Farming* ©2023”. Berikut ini dapat dilihat tampilan desain antarmuka tiap halaman dari aplikasi AMOR S-FARM:

A. Halaman Utama AMOR S-FARM

Halaman utama AMOR S-FARM adalah titik awal yang terbuka ketika pengguna membuka aplikasi, menampilkan desain yang dapat dilihat pada Gambar 4.2. *Navbar* di bagian atas halaman ini menyediakan akses langsung ke fitur-fitur utama, seperti data *realtime*, *sign-up*, dan *login*, memastikan keterjangkauan pengguna terhadap fungsionalitas yang penting. Di bagian *body* halaman, pengguna disajikan dengan penjelasan menyeluruh tentang aplikasi AMOR S-FARM, termasuk sistem *Ebb & Flow System* yang digunakan, serta rangkuman fitur-fitur yang ditawarkan, memberikan pengguna pemahaman yang jelas tentang nilai dan tujuan aplikasi tersebut.

B. Halaman Data *Realtime*

Halaman data *realtime* AMOR S-FARM menjadi jendela yang menghadirkan beberapa nilai parameter dari lingkungan *greenhouse* secara langsung. Tampilan halaman data *realtime* dapat dilihat ada Gambar 4.4, memberikan kemudahan bagi pengguna untuk memantau kondisi lingkungan secara *online*.



Gambar 4.4 Hasil *Interface* Halaman Data *Realtime* Monitoring

Gambar 4.4 merupakan tampilan halaman data *realtime* pada aplikasi AMOR S-FARM. Parameter-parameter yang terpampang mencakup kelembaban udara, suhu lingkungan, dan kelembaban 1, 2, serta 3, bersama dengan ketinggian air. Data-data parameter yang muncul di halaman ini adalah hasil langsung dari pengukuran sensor yang diteruskan ke *database*, memberikan pemantauan yang tepat waktu dan akurat terhadap kondisi lingkungan dalam *greenhouse*.

C. Halaman *Sign-up*

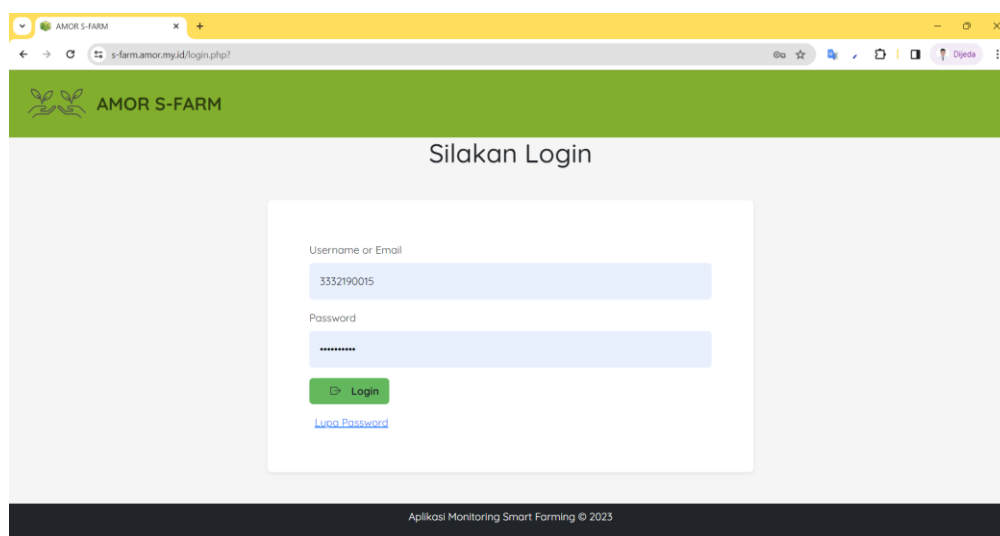
Halaman *sign-up* pada aplikasi AMOR S-FARM adalah tempat di mana pengguna dapat mendaftarkan akun baru. Tampilan halaman menu *sign-up* dapat dilihat pada Gambar 4.5.

Gambar 4.5 Hasil *Interface* Halaman Menu *Sign-up*

Gambar 4.5 merupakan tampilan dari halaman menu *sign-up* pada aplikasi *website* AMOR S-FARM. Bagian navbar pada halaman ini menyajikan logo dan nama aplikasi untuk memberikan identifikasi yang jelas kepada pengguna. Di bagian *body* halaman, pengguna diminta untuk mengisi identitas pribadi yang diperlukan untuk pendaftaran akun, termasuk nama, *username*, alamat email, *password*, dan nomor telepon, memastikan bahwa informasi yang diperlukan untuk membuat akun telah tersedia.

D. Halaman Login

Halaman login dalam aplikasi AMOR S-FARM berfungsi sebagai titik akses bagi pengguna untuk masuk ke dalam akun mereka. Di sini, pengguna diminta untuk memasukkan kredensial mereka sebelum dapat mengakses fitur-fitur aplikasi. Tampilan halaman login dapat dilihat pada Gambar 4.6, memberikan pengguna kemudahan untuk mengakses platform dan mengelola data mereka dengan aman.

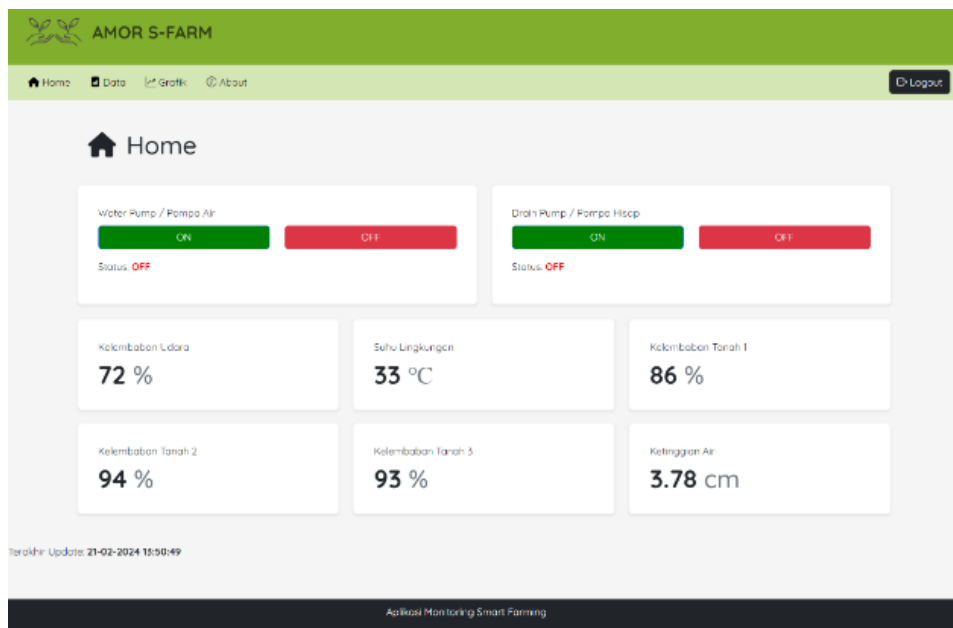


Gambar 4.6 Hasil *Interface* Halaman Menu *Log-in* AMOR S-FARM

Gambar 4.6 merupakan tampilan halaman menu *log-in* dari aplikasi *website* AMOR S-FARM. Bagian navbar pada halaman ini memiliki kesamaan dengan navbar pada halaman sign-up, yang menampilkan logo dan nama aplikasi untuk memberikan konsistensi dalam pengalaman pengguna. Di bagian body halaman, pengguna diminta untuk memasukkan identitas akun mereka, baik berupa *username* atau *email*, serta *password* yang sudah didaftarkan sebelumnya, memungkinkan akses ke berbagai fitur dan data yang tersedia dalam aplikasi.

E. Halaman *Home*

Halaman *home* pada aplikasi AMOR S-FARM menjadi salah satu halaman yang hanya dapat diakses setelah pengguna berhasil login ke akun mereka. Tampilannya dapat ditemukan dalam Gambar 4.7.

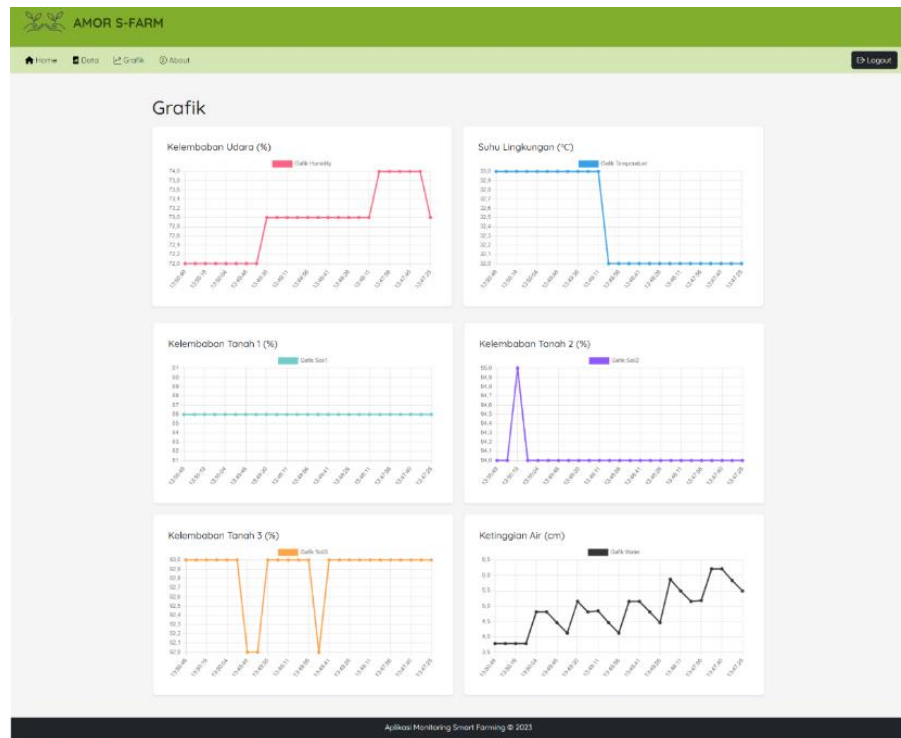


Gambar 4.7 Hasil *Interface* Halaman *Home* AMOR S-FARM

Gambar 4.7 merupakan tampilan halaman *home* dari aplikasi *website* AMOR S-FARM. Di bagian navbar halaman ini, terdapat logo dan nama aplikasi bersama dengan menu navigasi yang mencakup *home*, *data*, *grafik*, *about*, dan *logout*, memberikan akses cepat ke berbagai fitur dan informasi penting aplikasi. Pada bagian *body* halaman, pengguna disajikan dengan fitur *on/off relay* untuk mengontrol perangkat dalam *greenhouse*, serta beberapa nilai parameter lingkungan secara *realtime* seperti yang telah ditampilkan pada halaman data *realtime* sebelumnya, memberikan pemantauan yang terus-menerus terhadap kondisi lingkungan pertanian.

F. Halaman grafik

Halaman grafik dalam aplikasi AMOR S-FARM merupakan fitur yang hanya tersedia bagi pengguna yang telah melakukan *login* ke akun mereka. Di sini, pengguna dapat melihat visualisasi data yang relevan dengan kondisi lingkungan *greenhouse* secara grafis. Tampilan halaman grafik dapat ditemukan pada Gambar 4.8, memberikan pengguna kemampuan untuk memantau dan menganalisis data dengan lebih baik setelah melakukan *login* ke dalam *platform*.



Gambar 4.8 Hasil *Interface* Halaman Grafik AMOR S-FARM

Gambar 4.8 merupakan tampilan halaman yang menyajikan grafik dalam aplikasi AMOR S-FARM. Seperti halaman *home*, bagian *navbar* halaman ini juga memuat logo dan nama aplikasi serta menu navigasi yang serupa, termasuk *home*, *data*, *grafik*, *about*, dan *logout*, memastikan konsistensi dalam navigasi antarmuka pengguna. Di bagian *body* halaman, terdapat visualisasi data parameter lingkungan dalam bentuk grafik. Grafik ini dibentuk dari 25 data terbaru dari setiap parameter, memungkinkan pengguna untuk dengan cepat menganalisis tren dan pola yang berkaitan dengan kondisi lingkungan di dalam *greenhouse*.

G. Halaman History Data

Halaman *history* data dalam aplikasi AMOR S-FARM adalah fitur yang hanya dapat diakses oleh pengguna yang telah berhasil login ke akun mereka. Di sini, pengguna dapat melihat riwayat data sebelumnya yang telah terkumpul dari lingkungan *greenhouse*. Tampilan halaman *history* data dapat ditemukan pada Gambar 4.9, memberikan pengguna akses ke catatan historis untuk analisis dan referensi yang lebih lanjut setelah masuk ke dalam *platform*.

The screenshot displays the 'Data Table' section of the AMOR S-FARM application. The table contains 10 rows of data, each representing a monitoring entry. The columns are: NO #, Tanggal, Waktu, Kelembaban Udara (%), Suhu Lingkungan (C), Kelembaban Tanah 1 (%), Kelembaban Tanah 2 (%), Kelembaban Tanah 3 (%), and Ketinggian Air (cm). The data shows consistent values for ambient temperature (29°C) and air humidity (85%), while soil humidity and water height vary slightly across entries.

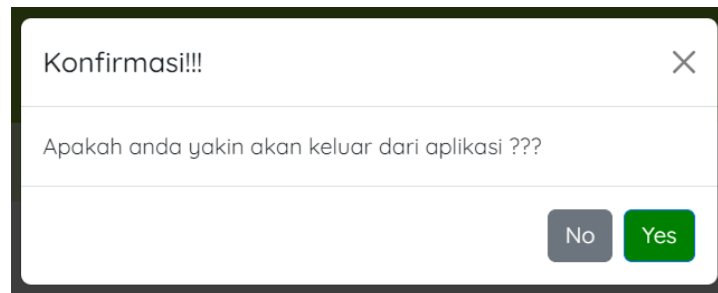
NO #	Tanggal	Waktu	Kelembaban Udara (%)	Suhu Lingkungan (C)	Kelembaban Tanah 1 (%)	Kelembaban Tanah 2 (%)	Kelembaban Tanah 3 (%)	Ketinggian Air (cm)
1	01-02-2024	13:20:33	84	29	13	12	63	15.92
2	01-02-2024	13:20:25	85	29	15	13	75	15.95
3	01-02-2024	13:20:18	85	29	13	12	64	15.92
4	01-02-2024	13:20:10	85	29	12	12	61	15.92
5	01-02-2024	13:20:00	85	29	13	12	62	15.95
6	01-02-2024	13:19:52	85	29	15	12	68	15.95
7	01-02-2024	13:19:45	85	29	15	13	74	15.95
8	01-02-2024	13:19:37	85	29	13	12	62	15.95
9	01-02-2024	13:19:30	85	29	14	12	66	15.95
10	01-02-2024	13:19:22	85	29	12	11	57	15.95

Gambar 4.9 Hasil *Interface* Halaman *History* Data AMOR S-FARM

Gambar 4.9 merupakan tampilan halaman *home* yang ada pada aplikasi website AMOR S-FARM. Sebagaimana halaman *home* dan grafik, bagian navbar halaman ini juga menghadirkan logo dan nama aplikasi, serta menu navigasi yang serupa, termasuk *home*, *data*, *grafik*, *about*, dan *logout*, menjaga konsistensi antarmuka pengguna. Pada bagian *body* halaman, pengguna dapat menemukan tampilan data *history* dalam bentuk tabel yang dapat disortir sesuai keinginan, memungkinkan pengguna untuk menyesuaikan jumlah data yang ingin ditampilkan dalam satu tabel. Selain itu, halaman ini juga dilengkapi dengan fitur unduh, memungkinkan pengguna untuk mengunduh data history tersebut sesuai kebutuhan.

H. Konfirmasi Login

Pada bagian konfirmasi *login* pada aplikasi AMOR S-FARM memberikan pilihan bagi pengguna untuk keluar aplikasi atau masih ingin tetap berada dalam aplikasi *website*, yang dapat dilihat pada Gambar 4.10.



Gambar 4.10 Hasil *Interface* Konfirmasi Aplikasi AMOR S-FARM

Pada Gambar 4.10 merupakan bagian konfirmasi login pada aplikasi AMOR S-FARM, pengguna diberikan pilihan untuk keluar dari aplikasi atau tetap berada di dalamnya. Hal ini memberikan fleksibilitas kepada pengguna untuk mengelola sesi mereka sesuai kebutuhan. Setelah melakukan *login*, pengguna dapat memilih apakah ingin tetap berada dalam aplikasi untuk melanjutkan aktivitasnya atau ingin keluar untuk mengakhiri sesi.

4.2. Pengujian *Database*

Pengujian *database* ini dilakukan untuk menguji fungsionalitas database, pengujian ini dilakukan dengan cara mengirimkan data hasil pengukuran sensor dari WEMOS D1R1. Data tersebut dikirim ke *database phpMyAdmin* menggunakan metode HTTP POST *request*. Poses pengujian pengiriman data dari WEMOS D1R1 ke database dapat dilihat pada Gambar 4.11.

	id_data	tanggal	waktu	humidity	temperature	soil1	soil2	soil3	v
<input type="checkbox"/>	1	24-01-2024	10:21:36	76	30	90	35	84	
<input type="checkbox"/>	2	24-01-2024	10:22:41	76	30	90	35	84	

Gambar 4.11 Pengujian *Database* (Tamilan PHP MyAdmin)

```

Complete_JSON | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Complete_JSON.ino
202 }
Output Serial Monitor x
Message (Enter to send message to 'LOLIN(WeMos) D1 R1' on 'COM5') New Line 9600 baud
10:21:01.564 -> Connecting to server...
10:21:03.002 -> HTTP 500
10:21:03.002 ->
10:21:03.002 -> deserializeJson() failed: EmptyInput
10:21:03.037 -> Kelembaban Tanah 1: 90.00 %
10:21:03.069 -> Kelembaban Tanah 2: 35.00 %
10:21:03.111 -> Kelembaban Tanah 3: 84.00 %
10:21:04.015 -> water: 20.00 cm
10:21:04.016 -> % Temperature: 29.80C
10:21:04.049 -> % Kelembaban: 75.70%
10:21:34.018 -> Connecting to server...
10:21:35.545 -> HTTP 200
10:21:35.545 ->
10:21:35.545 -> {"status":"OK","relay_solenoid":"OFF","relay_drain":"ON"}
10:21:35.624 -> status: OK
10:21:35.624 -> Relay_solenoid: OFF
10:21:35.624 -> Relay_drain: ON
10:22:05.553 -> Kelembaban Tanah 1: 90.00 %
10:22:05.553 -> Kelembaban Tanah 2: 35.00 %
10:22:05.659 -> Kelembaban Tanah 3: 84.00 %
10:22:05.662 -> water: 15.10 cm
10:22:05.674 -> % Temperature: 30.00C
10:22:05.674 -> % Kelembaban: 75.60%
10:22:35.566 -> Connecting to server...
10:22:40.169 -> HTTP 200
10:22:40.169 ->
10:22:40.169 -> {"status":"OK","relay_solenoid":"OFF","relay_drain":"ON"}
10:22:40.369 -> Status: OK
10:22:40.370 -> relay_solenoid: OFF
10:22:40.373 -> Relay_drain: ON
Ln 222, Col 1 LOLIN(WeMos) D1 R1 on COM5

```

Gambar 4.12 Pengujian *Database* (Tampilan Serial Monitor ArduinoIDE)

Berdasarkan Gambar 4.11, dapat diamati dengan jelas bahwa data berhasil dikirim dari perangkat keras ke *database* dan tercatat dalam tabel "*greenhouse*". Kesuksesan proses pengiriman ini ditandai dengan pesan "HTTP 200" yang muncul di serial monitor. Pentingnya pesan ini adalah sebagai indikator bahwa transfer data telah berhasil dilakukan. Selain itu, nilai-nilai yang terukur dari sensor, sebagaimana yang ditampilkan pada serial monitor, juga terbukti identik dengan data yang diterima dan tercatat di dalam *database*.

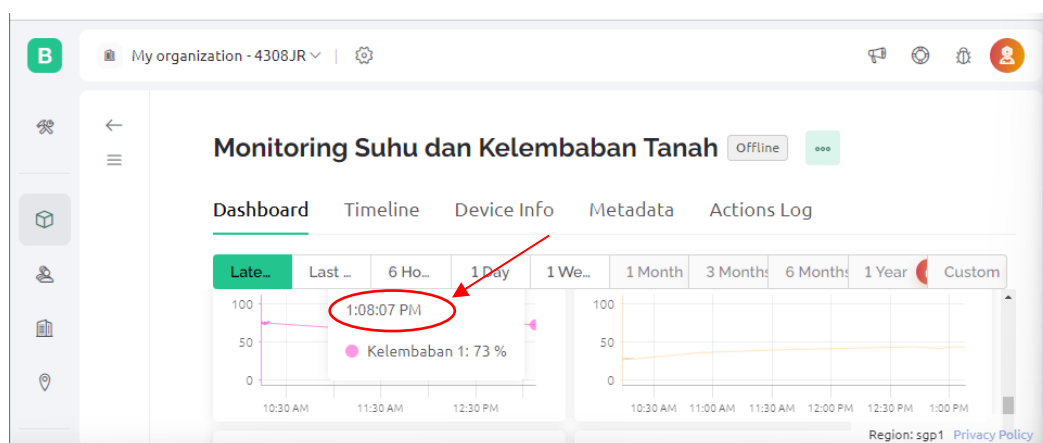
Berdasarkan catatan waktu pada Gambar 4.11, terlihat bahwa proses pengiriman data terjadi dua kali, yaitu pada pukul "10:21:35:545ms WIB" dan "10:22:40:169ms WIB". Meskipun demikian, data yang masuk ke dalam *database*

dicatat pada pukul "10:21:36 WIB" dan "10:22:41 WIB". Hal ini mengindikasikan bahwa proses pengiriman data ke *database* memakan waktu sekitar satu detik, atau dapat dibilang berlangsung dalam rentang waktu yang cukup singkat.

Hasil pengujian secara keseluruhan menunjukkan bahwa proses pengiriman data dari perangkat keras ke *database* berjalan dengan lancar dan stabil. Hal ini memberikan jaminan bahwa sistem ini dapat diandalkan untuk merekam dan menyimpan data sensor secara akurat dan efisien dalam basis data.

4.3. Pengujian Komparasi Aplikasi (Waktu Data Tampil)

Pengujian ini dilakukan untuk dengan performa aplikasi yang dirancang dengan membandingkannya terhadap aplikasi sejenis, yaitu; AMOR S-FARM dan BLYNK. Fokus utama utama pengujian ini adalah untuk mengetahui seberapa cepat data ditampilkan pada serial monitor dibandingkan dengan kedua aplikasi tersebut. Contoh perbandingan waktu tampil data ini dapat dilihat pada Gambar 4.12.



Gambar 4.13 Pengujian Komparasi Waktu Data Tampil (BLYNK)

	id_data	tanggal	waktu	humidity	temperature	soil1	soil2	soil3	water
Ubah Salin Hapus	1	24-01-2024	13:07:25	0	0	73	44	34	1.58
Ubah Salin Hapus	2	24-01-2024	13:08:06	0	0	73	44	34	0.28
Ubah Salin Hapus	3	24-01-2024	13:08:44	0	0	74	44	35	0.31

Gambar 4.14 Pengujian Komparasi Waktu Data Tampil (Database AMOR S-FARM)


```

Complete_JSON | Arduino IDE 2.2.1
Output Serial Monitor X
Message (Enter to send message to 'LOLIN(WeMos) D1 R1' on 'COM5') New Line 9600 baud
13:07:22.693 -> Connecting to server...
13:07:24.199 -> HTTP 200
13:07:24.200 ->
13:07:24.200 -> {"status":"OK","relay_solenoid":"OFF","relay_drain":"OFF"}
13:07:24.264 -> Status: OK
13:07:24.264 -> Relay_solenoid: OFF
13:07:24.264 -> Relay_drain: OFF
13:07:25.042 -> Kelembaban Tanah 1: 73.00 %
13:07:25.042 -> Kelembaban Tanah 2: 44.00 %
13:07:25.073 -> Kelembaban Tanah 3: 34.00 %
13:07:25.111 -> water: 0.28 cm
13:08:01.111 -> Connecting to server...
13:08:05.196 -> HTTP 200
13:08:05.196 ->
13:08:05.196 -> {"status":"OK","relay_solenoid":"OFF","relay_drain":"OFF"}
13:08:05.287 -> Status: OK
13:08:05.287 -> Relay_solenoid: OFF
13:08:05.287 -> Relay_drain: OFF
13:08:06.035 -> Kelembaban Tanah 1: 74.00 %
13:08:06.035 -> Kelembaban Tanah 2: 44.00 %
Ln 226, Col 3 LOLIN(WeMos) D1 R1 on COM5

```

Gambar 4.15 Pengujian Komparasi Waktu Data Tampil (Serial Monitor ArduinoIDE)

Berdasarkan Gambar 4.12, dapat diamati gambar pengujian komparasi waktu tampil data antara serial monitor, aplikasi AMOR S-FARM, dan aplikasi BLYNK. Dari hasil pengujian tersebut, terlihat bahwa waktu data tampil pada serial monitor waktu data tampil adalah 13.08.06.035, pada aplikasi AMOR S-FARM waktu data tampil adalah 13.08.06, sementara pada aplikasi BLYNK waktu data tampil adalah 13.08.07. Analisis lebih lanjut menunjukkan bahwa waktu data tampil pada aplikasi BLYNK mengalami keterlambatan sekitar 1 detik dibandingkan dengan AMOR dan serial monitor.

Selengkapnya dapat dilihat pada Tabel 4.1 hasil pengujian waktu tampil data dari Serial Monitir, AMOR S-FARM, dan BLYNK. Data tersebut memberikan gambaran yang lebih menyeluruh terkait perbedaan waktu tampil antara ketiga sumber tersebut. oleh karena itu, hasil ini penting untuk mengevaluasi kinerja dari setiap aplikasi dan memperoleh pemahaman yang diperlukan untuk pengembangan dan peningkatan aplikasi.

Tabel 4.1 Analisa Waktu Data Tampil Aplikasi

NO	SERIAL MONITOR	AMOR S-FARM		BLYNK	
	Waktu Tampil	Tampil	Selisih	Tampil	Selisih
1	16.02.01	16.02.02	+1s	16.02.04	+3s
2	16.02.08	16.02.09	+2s	16.02.14	+6s
3	16.02.18	16.02.19	+1s	16.02.21	+3s
4	16.02.26	16.02.29	+3s	16.02.33	+7s
5	16.02.36	16.02.37	+1s	16.02.39	+3s
6	16.02.44	16.02.44	+0s	16.02.46	+2s
7	16.02.51	16.02.51	+1s	16.02.54	+3s
8	16.02.58	16.02.59	+1s	16.03.01	+3s
9	16.03.06	16.03.06	+0s	16.03.08	+2s
10	16.03.13	16.03.14	+1s	16.03.16	+3s
11	16.03.21	16.03.22	+1s	16.03.24	+3s
12	16.03.28	16.03.29	+1s	16.03.31	+3s
13	16.03.36	16.03.36	+0s	16.03.38	+2s
14	16.03.43	16.03.44	+1s	16.03.47	+4s
15	16.03.51	16.03.51	+0s	16.03.53	+2s
16	16.03.58	16.03.59	+1s	16.04.02	+4s
17	16.04.06	16.04.07	+1s	16.04.08	+2s
18	16.04.13	16.04.14	+1s	16.04.16	+3s
19	16.04.20	16.04.21	+1s	16.04.24	+4s
20	16.04.28	16.04.29	+1s	16.04.31	+3s
Rata-rata			0.95s	Rata	3.25s

Berdasarkan Tabel 4.1, data hasil pengujian waktu tampil antara serial monitor, AMOR S-FARM dan BLYNK disajikan dengan detail. Pengujian dilakukan sebanyak 20 kali, dengan waktu data tampil pada serial monitor dijadikan sebagai acuan. Hasil pengujian menunjukkan variasi waktu data tampil yang fluktuatif, yang dipengaruhi oleh kondisi koneksi internet yang kurang stabil.

Berdasarkan Tabel 4.1, dapat diamati bahwa selisih waktu data tampil pada AMOR dan BLYNK cukup bervariasi. Pada AMOR S-FARM, selisih waktu data tampil terbesar adalah +3s, sementara yang terkecil adalah +0s yang. Sedangkan pada aplikasi BLYNK, selisih waktu data tampil terbesar mencapai +7s, dengan yang terkecil adalah +2s. Apabila data pada Tabel 4.1 ditampilkan dalam bentuk grafik batang dapat dilihat pada Gambar 4.13.

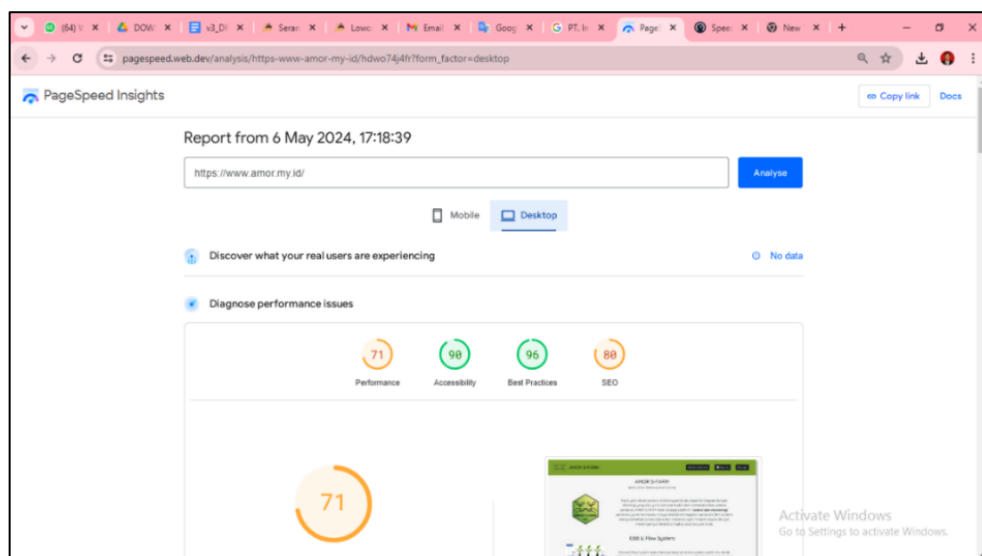
Berdasarkan Tabel 4.1, terlihat bahwa waktu data tampil pada AMOR S-FARM cenderung lebih stabil dan memiliki perbedaan yang lebih kecil dibandingkan dengan aplikasi BLYNK. Analisis lebih lanjut dari 20 kali percobaan

menunjukkan bahwa rata-rata waktu data tampil pada AMOR S-FARM adalah +0,95s, sementara pada BLYNK adalah +3,25s.

Berdasarkan hasil pengujian ini, dapat disimpulkan bahwa aplikasi AMOR S-FARM memberikan pengalaman yang lebih real-time dalam menampilkan data dibandingkan dengan aplikasi BLYNK. Variasi waktu yang lebih kecil dan selisih waktu data tampil yang lebih konsisten, membuktikan keunggulan AMOR S-FARM dalam hal responsivitas dan kestabilan penampilan data. Hal ini mengindikasikan bahwa AMOR S-FARM lebih dapat diandalkan untuk pemantauan data secara langsung dan tepat waktu, yang sangat penting dalam konteks pertanian modern yang memerlukan pemantauan yang akurat dan responsif terhadap kondisi lingkungan.

4.4. Pengujian Performa Aplikasi

Pengujian ini merupakan sebuah pengujian yang di lakukan untuk mengetahui kemampuan/performa aplikasi AMOR S-FARM. Pengujian ini dilakukan dengan menguji aplikasi dengan menggunakan *tools* tambahan seperti *Google Pagespeed* dan *GT-metrix*. Pengujian dilakukan untuk mengukur kecepatan dan efisiensi pemuatan halaman aplikasi, serta mengidentifikasi potensi area yang dapat ditingkatkan. Contoh hasil pengujian menggunakan *google pagespeed* dapat dilihat pada Gambar 4.14.



Gambar 4.16 Hasil Pengujian Performa Dengan *Google Pagespeed*

Pada Gambar 4.14 merupakan hasil pengujian performa halaman utama AMOR S-FARM yang dilakukan menggunakan *tools* google *pagespeed*. Pengujian tersebut dilakukan dengan menggunakan perangkat ASUS TUF FX505DD yang dibekali *processor* Ryzen 5 3550H dengan OS Windows 11. Berdasarkan Gambar 4.14 dapat dilihat hasil pengujian yang didapatkan adalah 71 untuk skor performa dan 90 untuk skor aksesibilitas.

Selanjutnya dilakukan beberapa pengujian dengan *tools* google *pagespeed* menggunakan perangkat dan kondisi yang berbeda guna mengetahui performa aplikasi pada segala kondisi perangkat. Pengujian dilakukan dengan 3 perangkat berbeda, yang dimana masing-masing perangkat memiliki spesifikasi berbeda, serta dengan kondisi jaringan berbeda. Hasil pengujian secara keseluruhan tersebut dapat dilihat pada tabel 4.2.

Tabel 4.2 Hasil Pengujian Google *Pagespeed*

Perangkat	Processor	OS	Percobaan ke-	Koneksi	CPU usage	Performance (%)
ASUS TUF FX505DD	Ryzen 5 3550H	Windows 11	1	Lan Indihome 27,45/9,48 Mbps	23%	71
			2	Lan Indihome 27,45/9,48 Mbps	58%	70
			3	Hotspot Smartfreen 11.64/2.51 Mbps	54%	65
Acer Aspire 14	Intel Core i3-6006U	Windows 10	1	Wi-Fi Indihome 18.55/2.88 Mbps	10%	67
			2	Wi-Fi Indihome 18.55/2.88 Mbps	42%	64
			3	Hotspot HP	18%	65
ASUS x550D	AMD A8-5550M APU	Windows 8.1 pro	1	Wi-Fi Myrepublic 3.56/5.60 Mbps	11%	68
			2	Wi-Fi Myrepublic 3.56/5.60 Mbps	32%	64
			3	Hotspot HP	18%	68

Pada Tabel 4.2 dapat di lihat hasil pengujian performa AMOR S-FARM dengan *tools* Google *Pagespeed* menggunakan perangkat dan kondisi berbeda. Pengujian dilakukan pada tiga perangkat dengan spesifikasi yang berbeda, yaitu ASUS TUF FX505DD (perangkat 1), Acer Aspire 14 (perangkat 2), dan ASUS x550D (perangkat 3). Perangkat 1 dan 2 memiliki prosesor yang lebih *powerfull* (Ryzen 5 3550H dan Intel Core i3-6006U) dibandingkan perangkat 3 (AMD A8-

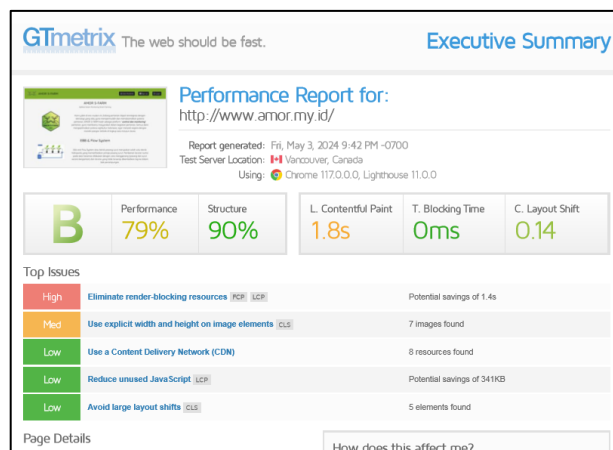
5550M APU). Ketiga perangkat menggunakan sistem operasi Windows, dengan versi yang berbeda.

Seperti yang dijelaskan sebelumnya, pengujian ini dilakukan dengan kondisi dan koneksi berbeda. Rata-rata penggunaan CPU berada di kisaran 10% - 58% pada semua perangkat saat melakukan percobaan pengujian. Sedangkan penggunaan koneksi internet, menggunakan kombinasi koneksi kabel LAN, *Wi-Fi* dan *hotspot* seluler. Pengujian ke-1 & 2 pada perangkat pertama memiliki kecepatan koneksi kabel LAN yang lebih tinggi dibandingkan *Wi-Fi* dan *hotspot* seluler pada pengujian lainnya. Sedangkan pengujian pada perangkat ke-3 menggunakan koneksi *Wi-Fi* Myrepublic dengan kecepatan *upload* yang lebih tinggi dibandingkan kecepatan *download*.

Berdasarkan Tabel 4.2, secara keseluruhan hasil pengujian performa terbesar memiliki skor 71 dengan menggunakan spesifikasi perangkat tertinggi, koneksi tercepat, dan penggunaan CPU terendah pada perangkat tersebut. Sedangkan berdasarkan penggunaan CPU, pada saat penggunaan CPU tertinggi skor performa pada perangkat tersebut mengalami penurunan dan menjadi skor terendah pada perangkat tersebut.

Berdasarkan hasil pengujian tersebut dapat disimpulkan spesifikasi perangkat dan koneksi internet cukup berpengaruh terhadap hasil pengujian. Dimana perangkat yang lebih *powerful* dan koneksi internet yang lebih stabil cenderung menghasilkan hasil pengujian lebih tinggi. Sedangkan penggunaan CPU sendiri berpengaruh terhadap hasil pengujian pada perangkat itu sendiri.

Selanjutnya pengujian performa AMOR S-FARM menggunakan tools Gtmetrix. Pengujian performa dilakukan menggunakan perangkat ASUS X550D yang dibekali processor AMD A8-5550M APU dengan sistem operasi windows 8.1 pro dengan jaringan internet *Wi-Fi* myrepublic. Hasil pengujian meliputi skor dan kelas GTmetrix untuk berbagai metrik, seperti performa, struktur, waktu muat halaman, ukuran halaman, *Largest Contentful Paint (LCP)*, *Cumulative Layout Shift (CLS)*, dan *Total Blocking Time (TBT)*. Contoh hasil pengujian performa AMOR S-FARM menggunakan tools GTmetrix dilihat pada Gambar 4.15.



Gambar 4.17 Hasil Pengujian Performa Dengan GT-metrix

Pada Gambar 4.15 merupakan hasil pengujian performa halaman utama (*index*) AMOR S-FARM dengan GT-metrix. Berdasarkan Gambar 4.15 hasil pengujian yang diperoleh mendapatkan skor nilai B dengan detail nilai *Performance* sebesar 79%, dan *Structure* sebesar 90%. Terdapat juga 5 *top issues* yang berisi 5 permasalahan paling menonjol pada halaman utama (*index*).

Selanjutnya dilakukan pengujian performa menggunakan *tools* GTmetrix pada setiap halaman AMOR S-FARM. Pada aplikasi AMOR S-FARM sendiri terdiri dari 8 halaman, yang berarti dilakukan 8 kali pengujian. Hasil pengujian performa setiap halaman pada AMOR S-FARM dapat dilihat pada Tabel 4.3

Tabel 4.3 Hasil pengujian GT Matrix

No	User interface	Grade	Performance Score	Structure Score	Fuly loaded Time	Largest Contentful Paint	Total Blocking Time	Comulative Layout Shift
1	Halaman Index	B	79%	90%	3.0s	1.8s	0ms	0.137
2	Halaman data realtime	A	92%	92%	30.5s	1.3s	0ms	0.010
3	Halaman sign-up	A	91%	92%	2.4s	1.4s	0ms	0.000
4	Halaman login	A	90%	92%	1.9s	1.4s	0ms	0.000
5	Halaman home	B	86%	90%	31.s	1.7s	0ms	0.004

No	User interface	Grade	Performance Score	Structure Score	Fuly loaded Time	Largest Contentful Paint	Total Blocking Time	Comulative Layout Shift
6	Halaman grafik	A	90%	89%	2.3s	1.4s	0ms	0.001
7	Halaman history data	B	84%	90%	2.8s	1.7s	90ms	0.004
8	Halaman About	A	92%	94%	2.3s	1.4s	0ms	0.008

Pada Tabel 4.2 merupakan hasil pengujian setiap halaman pada AMOR S-FARM menggunakan *tools* Gtmetrix. Berdasarkan hasil pengujian menggunakan GTmetrix, AMOR S-FARM menunjukkan performa UI dan kecepatan yang cukup baik, dengan rata-rata nilai *Grade* A dan B, serta nilai *Performance Score* rata-rata 87%. Website terasa responsif dan cepat dimuat oleh pengguna, dengan waktu *Fully Loaded Time* rata-rata di bawah 3 detik. Nilai LCP dan TBT yang baik menunjukkan bahwa elemen konten terbesar dimuat dengan cepat dan tidak ada kode yang memblokir *main thread browser*. Nilai CLS yang rata-rata di bawah 0.05 menunjukkan pergeseran konten yang tidak terduga saat halaman dimuat sangat minimal.

Namun, terdapat beberapa halaman yang perlu dioptimalkan untuk meningkatkan kecepatan dan stabilitas tampilan, yaitu:

1. Halaman Index yang memiliki nilai *Grade* B dengan *Fully Loaded Time* 3 detik dan CLS 0.137. Hal ini menunjukkan bahwa halaman ini perlu dioptimalkan untuk meningkatkan kecepatan *loading* dan stabilitas tampilan.
2. Halaman data realtime yang memiliki nilai *Grade* A dengan *Fully Loaded Time* 30.5 detik dan LCP 1.3 detik. Hal ini menunjukkan bahwa halaman ini memiliki *Fully Loaded Time* yang cukup lama. Hal ini perlu dioptimalkan untuk meningkatkan kecepatan *loading* halaman.
3. Halaman Home yang memiliki nilai *Grade* B dengan *Fully Loaded Time* 31 detik dan CLS 0.004. Hal ini menunjukkan bahwa halaman ini perlu dioptimalkan untuk meningkatkan kecepatan *loading* dan stabilitas tampilan.

Secara keseluruhan, AMOR S-FARM memiliki potensi untuk meningkatkan performa UI dan kecepatan dengan melakukan optimasi pada beberapa halaman

tersebut. Hal ini dapat dilakukan dengan mengoptimalkan gambar, *minify* dan kompres JavaScript, menunda pemuatan JavaScript, menggunakan CSS yang optimal, dan menggunakan browser *caching*. Dengan melakukan optimasi tersebut, website Anda akan menjadi lebih responsif, cepat dimuat, dan memberikan pengalaman pengguna yang lebih baik.

4.5. Pengujian *Usability* Aplikasi

Pengujian ini merupakan sebuah pengujian yang dilakukan untuk memperoleh penilaian terhadap sebuah aplikasi yang dilakukan dengan mengisi angket atau kuesioner. Sebelum seseorang mengisi angket atau kuesioner, responden harus menggunakan aplikasi yang telah dikembangkan. Setelah menggunakan aplikasi, responden diminta untuk mengisi kuesioner yang disediakan melalui Google Form. Kuesioner ini menggunakan metode skala Likert, di mana responden diminta untuk memberikan penilaian menggunakan skala nilai dari satu hingga lima, yang mencerminkan sikap atau tingkat kepuasan mereka terhadap aplikasi tersebut.

Dalam proses ini, kuesioner telah disebar kepada 30 responden yang merupakan pengguna *greenhouse*. Hasil penilaian dari responden dapat dilihat pada Lampiran B Tabel B.1, yang menggambarkan tingkat kepuasan dari berbagai aspek aplikasi. Berdasarkan hasil penilaian responden, dilakukan rekapitulasi hasil penilaian pada setiap poin pertanyaan. Hasil rekapitulasi hasil penilaian dapat dilihat pada Tabel 4.4.

Tabel 4.4 Rekap Nilai Kuesioner

No.	Pertanyaan	Nilai (Skala 1-5)
<i>Aspek Sistem (Systems)</i>		
1	Apakah tampilan aplikasi mudah dikenali?	4,63
2	Apakah aplikasi mudah dioperasikan?	4,36
3	Apakah informasi yang disediakan oleh aplikasi ini mudah di mengerti ?	4,40
<i>Aspek Pengguna (User)</i>		
4	Apakah menu dan tampilan halaman aplikasi mudah diingat?	4,46
5	Apakah mudah mengakses informasi kondisi yang ditawarkan?	4,36

No.	Pertanyaan	Nilai (Skala 1-5)
6	Apakah tampilan warna pada aplikasi nyaman dilihat dan tidak membosankan?	4,16
Aspek Interaksi (<i>Interaction</i>)		
7	Apakah aplikasi dapat dengan mudah dipelajari ?	4,36
8	Apakah informasi dan fitur yang tersedia sesuai dengan yang dibutuhkan?	4,23
9	Secara keseluruhan apakah penggunaan aplikasi ini memuaskan?	4,30
10	Apakah aplikasi membantu dan memberikan pengaruh terhadap usaha Pertanian?	4,40

Pada Tabel 4.4 mencerminkan hasil keseluruhan dari penilaian kepuasan pengguna berdasarkan aspek-aspek sistem, pengguna, dan interaksi. Berdasarkan data rekapitulasi tersebut, dapat disimpulkan bahwa tingkat kepuasan dari semua aspek yang dievaluasi berada di atas nilai 4. Hal ini mengindikasikan bahwa aplikasi yang telah dibuat berhasil memenuhi harapan pengguna dalam hal kemudahan penggunaan, keefektifan, dan interaksi yang baik.

Hasil rekapitulasi ini memberikan gambaran positif tentang penerimaan aplikasi oleh pengguna. Dengan demikian, dapat disimpulkan bahwa aplikasi tersebut mudah dikenali, mudah diingat, efisien dalam penggunaannya, dan memberikan tingkat kepuasan yang baik oleh responden. Ini menunjukkan bahwa upaya pengembangan aplikasi telah menghasilkan produk yang memenuhi ekspektasi pengguna dan layak untuk dipertahankan atau ditingkatkan lebih lanjut.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan dalam perancangan dan penerapan aplikasi berbasis web ini, dapat disimpulkan sebagai berikut:

1. Aplikasi sistem monitoring yang telah dirancang dan dibangun berhasil mengintegrasikan teknologi web dengan kebutuhan praktis dalam pengelolaan *rooftop greenhouse*. Fitur-fitur yang disediakan dalam aplikasi ini, seperti pemantauan suhu lingkungan, kelembaban udara, dan kelembaban tanah, memberikan pengguna dan pengelola *rooftop greenhouse* kemampuan untuk mengawasi kondisi lingkungan dengan lebih efektif dan efisien.
2. Pengujian dan evaluasi aplikasi menunjukkan bahwa aplikasi ini memiliki keandalan dan performa yang baik dalam menyajikan data monitoring secara *real-time*. Pengguna dapat dengan mudah mengakses informasi yang dibutuhkan untuk pengambilan keputusan yang tepat dalam pengelolaan *rooftop greenhouse*.
3. Respon pengguna terhadap aplikasi sistem monitoring ini cukup positif, dengan pengguna mengapresiasi kemudahan penggunaan antarmuka, kejelasan informasi yang disajikan, dan ketersediaan fitur-fitur yang relevan. Ini menunjukkan bahwa aplikasi ini memenuhi kebutuhan dan harapan pengguna dengan baik.

5.2. Saran

Berdasarkan penelitian yang sudah dilakukan dalam perancangan dan penerapan aplikasi berbasis web ini terdapat beberapa saran untuk peneliti yang akan mengembangkan penelitian ini dikemudian hari, sebagai berikut :

1. Peningkatan Fitur: Menambahkan fitur-fitur tambahan yang dapat meningkatkan fungsionalitas dan kegunaan aplikasi, seperti pemantauan nutrisi tanaman, prediksi cuaca, atau integrasi dengan sistem otomatisasi untuk kontrol lingkungan.

2. Pengembangan Aplikasi Mobile: Mengembangkan versi aplikasi mobile dari sistem monitoring ini untuk meningkatkan aksesibilitas dan kemudahan penggunaan, terutama bagi pengguna yang sering bepergian atau memantau keadaan *greenhouse* dari jarak jauh.
3. Pelatihan Pengguna: Menyelenggarakan pelatihan dan bimbingan untuk pengguna aplikasi, terutama bagi pengelola *rooftop greenhouse* yang belum terbiasa dengan teknologi web. Ini akan membantu memastikan pengguna dapat memanfaatkan fitur-fitur aplikasi secara maksimal.
4. Evaluasi Lanjutan: Melakukan evaluasi lanjutan terhadap aplikasi ini setelah diimplementasikan dalam lingkungan lapangan untuk mengidentifikasi area-area peningkatan dan memastikan kesesuaian aplikasi dengan kebutuhan pengguna yang beragam.

Dengan menerapkan saran-saran tersebut, diharapkan aplikasi sistem monitoring pada *rooftop greenhouse* ini dapat terus berkembang dan memberikan kontribusi yang lebih besar dalam mendukung pertanian berkelanjutan dan pengelolaan lingkungan yang efisien.

DAFTAR PUSTAKA

- [1] F. Faramayuda *et al.*, “Traditional Uses, Phytochemistry, Pharmacology Activity and Plant Tissue Culture of Ficus Carica L.(a Mini Review),” *Riset Informasi Kesehatan*, vol. 11, no. 1, pp. 1–12, 2022.
- [2] R. B. Setiawan *et al.*, “Pengaruh Beberapa Konsentrasi Benzile Amino Purin (Bap) Terhadap Multiplikasi Tunas Nilam (Pogostemon Cablin Benth),” *Jurnal Sains Agro*, vol. 8, no. 1, pp. 83–87, 2023.
- [3] Y. P. B. Ziraluo, “Metode Perbanyak Tanaman Ubi Jalar Ungu (Ipomea Batatas Poiret) Dengan Teknik Kultur Jaringan Atau Stek Planlet,” *Jurnal Inovasi Penelitian*, vol. 2, no. 3, pp. 1037–1046, 2021.
- [4] M. A. Ababil, B. Budiman, and T. K. K. Azmi, “Aklimatisasi Planlet Pisang Cavendish Dengan Beberapa Kombinasi Media Tanam,” *Jurnal Pertanian Presisi (Journal of Precision Agriculture)*, vol. 5, no. 1, pp. 57–70, 2021.
- [5] A. Nora, T. Nova, and B. R. Noveni, “Aklimatisasi Plantlet Pisang Cavendish (Musa Acuminata) Pada Perbedaan Komposisi Media Tanam,” *Gontor AGROTECH Science Journal*, vol. 5, no. 2, pp. 111–126, 2019.
- [6] Y. B. W. Yohanes Bowo Widodo, S. S. Sondang Sibuea, T. S. Tata Sutabri, and I. A. Ibrahim Aziz, “Rancang Bangun Smart Greenhouse Berbasis Raspberry Pi dengan Web Framework Flask untuk Pertanian Perkotaan,” *Jurnal Teknologi Informatika dan Komputer*, vol. 8, no. 2, pp. 237–250, 2022.
- [7] E. Appolloni *et al.*, “The Global Rise of Urban Rooftop Agriculture: A Review of Worldwide Cases,” *J Clean Prod*, vol. 296, p. 126556, 2021.
- [8] D. Sasmoko, “Sistem Monitoring aliran air dan Penyiraman Otomatis Pada Rumah Kaca Berbasis IoT dengan Esp8266 dan Blynk,” *CIRCUIT: Jurnal Ilmiah Pendidikan Teknik Elektro*, vol. 4, no. 1, pp. 1–10, 2020.
- [9] S. C. Ko. Tekouabou, A. Gallais, and S. Agoujil, “DTN Routing Hierarchical Topology for the Internet of Things,” *Procedia Comput Sci*, vol. 170, pp. 490–497, 2020.
- [10] S. Tang, “Performance Modeling and Optimization for a Fog-Based IoT Platform,” *IoT*, vol. 4, no. 2, pp. 183–201, 2023.

- [11] Rocky Alfanz, A. H. A. Aqbal, and Wahyuni Martiningsih, "Smart Farm Agriculture Design by Applying a Solar Power Plant," *JURNAL NASIONAL TEKNIK ELEKTRO*, vol. 12, no. 2, Jul. 2023, doi: 10.25077/jnte.v12n2.1085.2023.
- [12] Y. Efendi, "Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile," *Jurnal Ilmiah Ilmu Komputer*, vol. 4, no. 2, pp. 21–27, Sep. 2018, doi: 10.35329/jiik.v4i2.41.
- [13] R. K. Jain, "Experimental Performance of Smart IoT-Enabled Drip Irrigation System Using and Controlled Through Web-Based Applications," *Smart Agricultural Technology*, vol. 4, p. 100215, 2023.
- [14] A. Suryaningrat, D. Kurnianto, and R. A. Rochmanto, "Sistem Monitoring Kelembaban Tanaman Cabai Rawit menggunakan Irigasi Tetes Gravitasi berbasis Internet Of Things (IoT)," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 10, no. 3, p. 568, 2022.
- [15] F. Suryatini, M. Maimunah, and F. I. Fauzandi, "Sistem Akuisisi Data Suhu Dan Kelembaban Tanah Pada Irigasi Tetes Otomatis Berbasis Internet Of Things," *Prosiding Semnastek*, 2018.
- [16] E. Tando, "Pemanfaatan teknologi greenhouse dan hidroponik sebagai solusi menghadapi perubahan iklim dalam budidaya tanaman hortikultura," *Buana Sains*, vol. 19, no. 1, pp. 91–102, 2019.
- [17] H. Budiyanto, M. Haris, A. B. Setiawan, E. Sonalitha, and M. Iqbal, "Teknologi Greenhouse Hidroponik dengan Tenaga Listrik Mandiri," *Prosiding SEMSINA*, pp. VII–17, 2019.
- [18] R. Suryani and M. N. Sari, "Penggunaan Media Tanam dan Pupuk Organik Cair pada Tahap Aklimatisasi terhadap Pertumbuhan Bibit Anggrek Bulan (*Phalaenopsis amabilis*) Hasil Kultur Jaringan," *Jurnal Agroqua: Media Informasi Agronomi dan Budidaya Perairan*, vol. 17, no. 1, pp. 67–75, 2019.
- [19] R. A. Latif, S. Hasibuan, and S. Mardiana, "Stimulasi Pertumbuhan dan Perkembangan Planlet Anggrek (*Dendrobium* sp) pada Tahap Aklimatisasi

- dengan Pemberian Vitamin B1 dan Atonik,” *Jurnal Ilmiah Pertanian (JIPERTA)*, vol. 2, no. 2, pp. 127–134, 2020.
- [20] W. Martiningsih, R. Wiryadinata, and T. M. Firas, “Pemanfaatan Solar Cell Untuk Penggunaan Hidroponik (Deep Flow Technique) Menggunakan Led Strip Sebagai Pencahayaan Terhadap Pertumbuhan Tanaman Kangkung,” *Setrum : Sistem Kendali-Tenaga-elektronika-telekomunikasi-komputer*, vol. 10, no. 2, pp. 90–104, Nov. 2021, doi: 10.36055/setrum.v10i2.13139.
- [21] A. Romalasari and E. Sobari, “Produksi Selada (*Lactuca sativa* L.) Menggunakan Sistem Hidroponik dengan Perbedaan Sumber Nutrisi,” *Agriprima, Journal of Applied Agricultural Sciences*, vol. 3, no. 1, pp. 36–41, 2019.
- [22] H. Halauddin, S. Supiyati, and S. Suhendra, “Perancangan dan Pemanfaatan Teknologi Hidroponik Vertikal Hidro 40 Hole Bagi Karang Taruna Tri Tunggal di Desa Talang Pauh,” *Dharma Raflesia : Jurnal Ilmiah Pengembangan dan Penerapan IPTEKS*, vol. 16, no. 1, Jun. 2018, doi: 10.33369/dr.v16i1.4825.
- [23] F. Rahutomo, S. Sutrisno, S. Pramono, M. E. Sulistyono, M. H. Ibrahim, and J. Haryono, “Implementasi dan Sosialisasi Smart Farming Hidroponik Berbasis Internet of Thing di Dusun Ngentak, Bulakrejo, Sukoharjo,” *Jurnal Abdi Masyarakat Indonesia*, vol. 2, no. 6, pp. 1961–1970, 2022.
- [24] K. Roberto, *How to hydroponics*. Futuregarden, Inc., 2005.
- [25] A. Fadly and R. W. Astuti, “Rancang Bangun Sistem Hidroponik Ebb and Flow Otomatis Budidaya Tanaman Cabai Pada Kelompok Wanita Tani Fokus Usaha Berbasis Mikrokontroler Arduino,” *Jurnal Karya Informatika (KARTIKA)*, vol. 2, no. 1, pp. 88–94, 2020.
- [26] B. Baihaqi *et al.*, “Pelatihan Budidaya Tanaman Hidroponik untuk Menambah Keterampilan dan Kreativitas Anak Didik Lapas (ANDIKPAS) Kelas II-A Kota Banda Aceh,” *BAKTIMAS: Jurnal Pengabdian pada Masyarakat*, vol. 5, no. 2, pp. 129–139, 2023.
- [27] N. Marbun, F. A. Mayari, F. Fitriah, S. Novani, M. N. Adlini, and K. Khairuna, “Hidroponik dan Aquaponik Sederhana: Solusi Budidaya Sayur

- di Lahan Terbatas Dalam Skala Rumah Tangga,” *Martabe: Jurnal Pengabdian Kepada Masyarakat*, vol. 5, no. 8, pp. 2834–2843, 2022.
- [28] E. Ulfada, N. Nurfiana, and R. D. Handayani, “Perancangan Desain UI/UX Pada Implementasi Sistem Kontrol Smart Farming Berbasis Internet of Things (IoT),” in *Prosiding Seminar Nasional Darmajaya*, 2022, pp. 145–155.
- [29] S. Pamungkas, “Smart Greenhouse System On Paprican Plants Based On Internet of Things,” *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, vol. 7, no. 2, pp. 197–207, 2019.
- [30] T. M. Lubis, “Sistem Monitoring–Pengendalian Pengairan dan Konsultasi Budidaya Pertanian Berbasis Internet of Things (IOT),” *Yogya arta: Universitas Gajah Mada*, 2019.
- [31] D. P. Sari, “Prototype Alat Monitoring Suhu, Kelembaban dan Kecepatan Angin Untuk Smart Farming Menggunakan Komunikasi Lora dengan Daya Listrik Menggunakan Panel Surya,” *KILAT*, vol. 10, no. 2, pp. 370–380, 2021.
- [32] R. Riska and H. Alamsyah, “Penerapan Sistem Keamanan Web Menggunakan Metode Web Application Firewall,” *Jurnal Amplifier: Jurnal Ilmiah Bidang Teknik Elektro Dan Komputer*, vol. 11, no. 1, pp. 37–42, 2021.
- [33] N. Nazwita and S. Ramadhani, “Analisis Sistem Keamanan Web Server Dan Database Server Menggunakan Suricata,” in *Seminar Nasional Teknologi Informasi Komunikasi dan Industri*, 2017, pp. 308–317.
- [34] V. O. Vicky and A. Syaripudin, “Perancangan Sistem Informasi Absensi Pegawai Berbasis Web Dengan Metode Waterfall (Studi Kasus: Kantor Dbpr Tangerang Selatan),” *OKTAL: Jurnal Ilmu Komputer dan Sains*, vol. 1, no. 01, pp. 17–26, 2022.
- [35] I. Z. T. Dewi, M. F. Ulinuha, W. A. Mustofa, A. Kurniawan, and F. A. Rakhmadi, “Smart Farming: Sistem Tanaman Hidroponik Terintegrasi IoT MQTT Panel Berbasis Android,” *Jurnal Keteknikian Pertanian Tropis Dan Biosistem*, vol. 9, no. 1, pp. 71–78, 2021.

- [36] I. P. G. E. E. Kurnia and A. A. G. Ekayana, “Rancang Bangun Sistem Smart Farming Berbasis IoT Studi Kasus Kebun Nyoman Gunitir,” *Jurnal Krisnadana*, vol. 1, no. 3, pp. 37–47, 2022.
- [37] R. S. Anwar, N. Agustina, and E. Sutinah, “Implementasi Teknologi IoT pada Smart Farming dalam Memonitoring Tanaman Hidroponik Berbasis Android,” *Journal of Students ‘Research in Computer Science*, vol. 4, no. 1, pp. 11–22, 2023.
- [38] R. Y. Endra, A. Cucus, S. Wulandana, and M. Aditya, “Perancangan Aplikasi Berbasis Web Pada System Aeroponik untuk Monitoring Nutrisi Menggunakan Framework CodeIgniter,” *Explore: Jurnal Sistem Informasi Dan Telematika*, vol. 11, no. 1, pp. 10–16, 2020.

LAMPIRAN A
DOKUMENTASI KEGIATAN



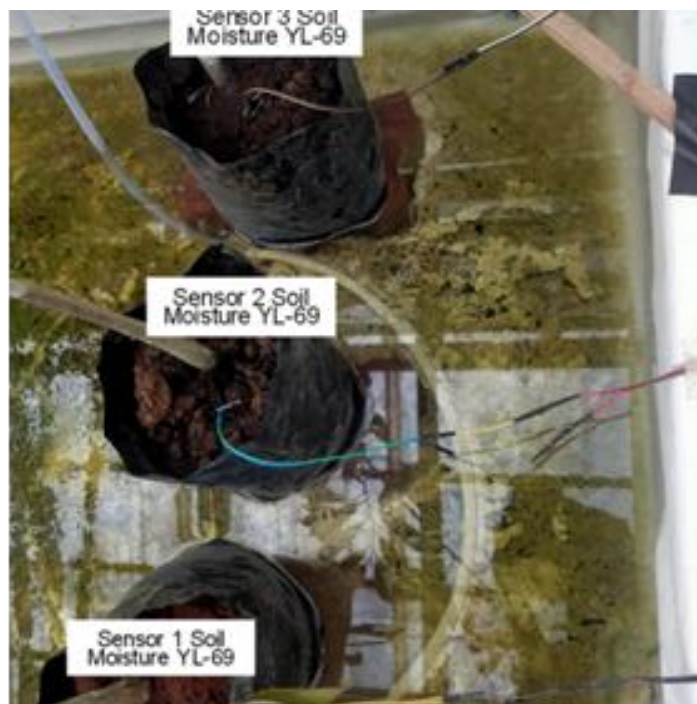
Gambar A.1 Tampak Luar *Greenhouse* Agroekoteknologi



Gambar A.2 Tampak Belakang *Greenhouse* Agroekoteknologi



Gambar A.3 Posisi Penempatan Sensor DHT22 untuk Suhu dan Kelembaban



Gambar A.4 Posisi Penempatan Sensor *Soil Moisture* YL-69



Gambar A.5 Posisi Penempatan Sensor Ultrasonik HC-SR04



Gambar A.10 *Box* Rangkaian Alat



Gambar A.11 Peletakan Alat Penelitian

LAMPIRAN B
KUISIONER

Tabel B.1 Nilai

Pernyataan	STS	TS	N	S	SS
Nilai	1	2	3	4	5

Keterangan :

STS = Sangat Tidak Setuju

TS = Tidak Setuju

N = Netral

S = Setuju

SS = Sangat Setuju

Tabel B.2 Pertanyaan

Kode	Pertanyaan
P1	Apakah tampilan aplikasi mudah dikenali?
P2	Apakah aplikasi mudah dioperasikan?
P3	Apakah informasi yang disediakan oleh aplikasi ini mudah di mengerti ?
P4	Apakah menu dan tampilan halaman aplikasi mudah diingat?
P5	Apakah mudah mengakses informasi kondisi yang ditawarkan?
P6	Apakah tampilan warna pada aplikasi nyaman dilihat dan tidak membosankan?
P7	Apakah aplikasi dapat dengan mudah dipelajari ?
P8	Apakah informasi dan fitur yang tersedia sesuai dengan yang dibutuhkan?
P9	Secara keseluruhan apakah penggunaan aplikasi ini memuaskan?
P10	Apakah aplikasi membantu dan memberikan pengaruh terhadap usaha Pertanian?

Tabel B.3 Skor Responden

Nama	Status responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Ade Kamilatus Soleha	Mahasiswa	5	5	5	5	5	5	5	5	5	5
Figo	Mahasiswa	5	5	5	5	5	3	4	5	5	5

Fachrudin	Mahasiswa	5	5	5	4	4	4	5	4	5	5
Maya Novianti	Mahasiswa	5	5	5	5	5	5	5	5	5	5
Asti	Mahasiswa	5	5	5	5	5	5	5	5	5	5
Anisa	Mahasiswa	5	4	4	4	5	4	3	4	3	4
Nanang Krisdianto	Dosen	5	5	5	4	3	4	4	4	4	4
Mutiara Ramadhani	Mahasiswa	5	4	5	4	5	5	5	5	4	5
Nurdiansyah	Mahasiswa	5	4	4	4	5	4	4	4	4	5
Dimas	Mahasiswa	5	5	5	5	5	5	5	5	5	5
Siti Anisa	Civitas akademika	4	4	3	4	4	4	3	4	3	3
Az-Zahra Farras	Mahasiswa	4	3	4	4	4	4	4	4	4	3
Nabila Khairunnisa	Mahasiswa	5	5	4	5	4	4	5	5	5	5
Esti Isnainiyati	Mahasiswa	5	5	5	5	5	4	5	5	5	5
Ivan	Mahasiswa	5	5	3	5	2	2	3	3	3	4
Yayu Rohmadona	Dosen	5	4	4	4	4	4	4	4	4	4
Novia Oktaviani	Mahasiswa	5	5	5	5	5	5	5	5	5	5
Maya	Mahasiswa	4	4	4	4	4	4	4	4	4	4
Lutfia	Mahasiswa	5	5	5	4	5	5	5	5	5	5
Cindy	Mahasiswa	4	4	5	5	4	4	5	4	4	3
Dzul hafizh amrullah	Mahasiswa	5	3	4	3	5	5	5	4	4	4
Akhdan	Mahasiswa	4	3	3	4	5	3	3	3	4	4
Monica Lidya Putri	Mahasiswa	4	5	5	5	4	5	5	4	5	5
Yusraini Muharni	Dosen	5	5	5	5	5	5	5	5	5	5

Hari Purnomo	Mahasiswa	4	3	4	4	3	3	4	3	3	3
Surya Atmadilaga Ananta	Mahasiswa	4	4	3	5	4	2	5	4	4	5
Safril	Mahasiswa	4	4	5	5	4	5	5	5	4	5
Andrian Azis	Mahasiswa	3	3	4	3	4	3	3	3	4	4
Dyah Lintang Trenggonowati	Dosen	5	5	4	5	4	4	4	4	4	4
Anita Shafarahmah	Mahasiswa	5	5	5	5	5	5	5	5	5	5
Ahmad Adriansyah	Mahasiswa	5	5	5	5	5	5	4	3	5	4

LAMPIRAN C
KODE PROGRAM

1. index.php

```
<?php include('layout/header.php') ?>
<?php include('layout/navbar_home.php') ?>

<div class="container">

    <div class=" d-flex flex-column mb-3 py-1 text-center">
        <?php
            if (isset($_GET['pesan'])) {
                if ($_GET['pesan'] == "username_salah") {
                    echo "<div class='alert alert-danger' role='alert'>Username tidak ditemukan!</div>";
                } else if ($_GET['pesan'] == "password_salah") {
                    echo "<div class='alert alert-danger' role='alert'>Password salah!</div>";
                } else if ($_GET['pesan'] == "logout") {
                    echo "<div class='alert alert-success' role='alert'>Anda telah berhasil logout!</div>";
                } else if ($_GET['pesan'] == "belum_login") {
                    echo "<div class='alert alert-danger' role='alert'>Anda harus login untuk mengakses halaman admin!</div>";
                } else if ($_GET['pesan'] == "signup") {
                    echo "<div class='alert alert-success' role='alert'>Akun anda berhasil disimpan, silahkan login!</div>";
                }
            }
        ?>

        <div>
            <div>
                <h2 class="mt-3">AMOR S-FARM</h2>
                <p>Aplikasi Sistem Monitoring Smart Farming</p>
                <div class="row ">
                    <div class="col-md-3">
                        
                    </div>

                    <div class="col-md-9 d-flex align-items-center ">
                        <p class="text-center p-3 m-4 fs-5">Kami yakin di era <i>modern
                            </i>ini, bidang
                            pertanian dapat
                            terintegrasi dengan teknologi yang ada, guna mempermudah dan memaksimalkan potensi
                            pertanian. AMOR
                            S-FARM hadir sebagai platform<q><strong><i>control dan monitoring</i></strong></q>pertanian,
                            guna
                            membantu masyarakat dalam kegiatan pertanian. Semua demi mengoptimalkan potensi agrikultur
                            Indonesia,
                            agar menjadi negara dengan mandiri pangan terbaik di lingkup asia maupun dunia.
                        </p>
                    </div>
                </div>
            </div>
        </div>

        <div class="pt-3">
            <h2 class="mb-3">EBB & Flow System</h2>
        </div>
    </div>
</div>
```

```

<div class="row ">
  <div class="col-md-3">
    
  </div>

  <div class="col-md-9">
    <p class="text-center p-3 m-4 fs-5">
      Ebb and Flow System atau teknik pasang surut merupakan salah satu teknik hidroponik yang
      memanfaatkan prinsip pasang surut. Pemberian larutan nutrisi pada akar tanaman dilakukan
      dengan
      cara menggenang (pasang dan surut secara bergantian) dan larutan yang tidak terserap
      dikembalikan
      lagi ke
      dalam bak penampungan.
    </p>
  </div>
</div>
</div>
</div>

<div class="card">
  <div class="card-header text-center fs-2 p-0 m-0 .bg-light">
    <strong>Fitur</strong>
  </div>

  <div class="card-body">
    <div class="d-flex justify-content-center">
      <p>AMOR S-FARM memiliki 4 fitur unggulan guna membantu petani dalam memantau kondisi tanaman</p>
    </div>

    <div class="d-flex justify-content-evenly container-fluid mt-2">
      <div class="d-flex flex-column col-sm-3 px-3 text-center">
        <div class="p-2">
          
        </div>
        <div>
          Kondisi <i>Real-time</i>
        </div>
      </div>

      <div class="d-flex flex-column mb-3 col-sm-3 px-3 text-center">
        <div class="p-2">
          
          <div>
            Data Dalam Bentuk Angka & Grafik
          </div>
        </div>
      </div>

      <div class="d-flex flex-column mb-3 col-sm-3 px-3 text-center">

```


2. dashboard.php

```
<?php include('layout/header.php') ?>
<?php include('layout/navbar_home.php') ?>

<div class="container">

    <main>

        <h2 class="m-3"> <i class="bi bi-speedometer2 px-2 "></i> Data Real-time</h2>

        <?php
        if (isset($_GET['pesan'])) {
            if ($_GET['pesan'] == "username_salah") {
                echo "<div class='alert alert-danger' role='alert'>Username tidak ditemukan!</div>";
            } else if ($_GET['pesan'] == "password_salah") {
                echo "<div class='alert alert-danger' role='alert'>Password salah!</div>";
            } else if ($_GET['pesan'] == "logout") {
                echo "<div class='alert alert-success' role='alert'>Anda telah berhasil logout!</div>";
            } else if ($_GET['pesan'] == "belum_login") {
                echo "<div class='alert alert-danger' role='alert'>Anda harus login untuk mengakses halaman admin!</div>";
            } else if ($_GET['pesan'] == "signup") {
                echo "<div class='alert alert-success' role='alert'>Akun anda berhasil disimpan, silahkan login!</div>";
            }
        }
        ?>

        <div class="row row-cols-1 row-cols-md-3 row-cols-2 g-4">

            <div class="col">
                <div class="card h-100 border-0 shadow-sm p-3">
                    <div class="card-body">
                        <p class="card-title">Humidity</p>
                        <h1 class="card-text text-secondary"><strong id='humidity' class="text-dark">00</strong> %</h1>
                    </div>
                </div>
            </div>

            <div class="col">
                <div class="card h-100 border-0 shadow-sm p-3">
                    <div class="card-body">
                        <p class="card-title">Temperature</p>
                        <h1 class="card-text text-secondary"><strong id='temperature' class="text-dark">00</strong> °C
                    </h1>
                    </div>
                </div>
            </div>

            <div class="col">
                <div class="card h-100 border-0 shadow-sm p-3">
                    <div class="card-body">
```

```

        <p class="card-title">Soil1</p>
        <h1 class="card-text text-secondary"><strong id='soil1' class="text-dark">00</strong> %</h1>
    </div>
</div>
</div>

<div class="col">
    <div class="card h-100 border-0 shadow-sm p-3">
        <div class="card-body">
            <p class="card-title">Soil2</p>
            <h1 class="card-text text-secondary"><strong id='soil2' class="text-dark">00</strong> %</h1>
        </div>
    </div>
</div>

<div class="col">
    <div class="card h-100 border-0 shadow-sm p-3">
        <div class="card-body">
            <p class="card-title">Soil3</p>
            <h1 class="card-text text-secondary"><strong id='soil3' class="text-dark">00</strong> %</h1>
        </div>
    </div>
</div>

<div class="col">
    <div class="card h-100 border-0 shadow-sm p-3">
        <div class="card-body">
            <p class="card-title">Water</p>
            <h1 class="card-text text-secondary"><strong id='water' class="text-dark">00</strong> cm</h1>
        </div>
    </div>
</div>

</div>

<p class="mt-5">Terakhir Update: <strong id='tanggal'>2021-07-01 00:00:00</strong></p>
</main>

```

```
</div>
```

```
<script>
```

```

    if (typeof (EventSource) !== "undefined") {
        var source = new EventSource("sse_data.php");
        source.addEventListener('data', function (e) {
            var data = JSON.parse(e.data);
            var tanggal = data.tanggal;
            var waktu = data.waktu;
            var humidity = data.humidity;
            var temperature = data.temperature;
            var soil1 = data.soil1;
            var soil2 = data.soil2;
            var soil3 = data.soil3;

```

```
        var water = data.water;

        document.getElementById("humidity").innerText = humidity;
        document.getElementById("temperature").innerText = temperature;
        document.getElementById("soil1").innerText = soil1;
        document.getElementById("soil2").innerText = soil2;
        document.getElementById("soil3").innerText = soil3;
        document.getElementById("water").innerText = water;
        document.getElementById("tanggal").innerText = tanggal + " " + waktu;

    }, false);

} else {
    document.getElementById("result").innerHTML = "Not Support";
}
</script>

<?php include('layout/footer.php') ?>
```


3. home.php

```
<?php
session start();
include('layout/header.php');
include('layout/navbar.php');

if (!isset($_SESSION['status']) || $_SESSION['status'] !== 'login') {
    header("location: index.php?pesan=belum_login");
    exit();
}

include('config/connection.php');
$query = "SELECT * FROM pengaturan";
$read = mysqli_query($conn, $query);
$row = mysqli_fetch_array($read);
$relay_selenoid = $row['relay_selenoid'];
$relay_drain = $row['relay_drain'];
?>

<div class="container">

    <?php
        if (isset($_GET['pesan'])) {
            if ($_GET['pesan'] == "signup") {
                echo "<div class='alert alert-success' role='alert'>Akun anda berhasil disimpan, Selamat Datang!</div>";
            }
        }
    ?>

    <h1 class="mb-3"><i class="fa-solid fa-house px-3 "></i>Home</h1>

    <div class="row row-cols-1 row-cols-md-2 g-4 py-4">
        <div class="col">
            <div class="card h-100 border-0 shadow-sm p-3">
                <div class="card-body">
                    <p class="card-title">Water Pump / Pompa Air</p>
                    <div class="row">
                        <div class="col">
                            <div class="d-grid gap-2">
                                <a href="api/update.php?relay_selenoid=ON" class="btn btn-primary">ON</a>
                            </div>
                        </div>
                        <div class="col">
                            <div class="d-grid gap-2">
                                <a href="api/update.php?relay_selenoid=OFF" class="btn btn-danger">OFF</a>
                            </div>
                        </div>
                    </div>
                </div>
                <p class="mt-3">Status:
                    <strong>
```

```

        <?php
        if ($relay_selenoid == 'ON') {
            echo "<span class='text-primary'>" . $relay_selenoid . "</span>";
        } else if ($relay_selenoid == 'OFF') {
            echo "<span class='text-danger'>" . $relay_selenoid . "</span>";
        }
        ?>
    </strong>
</p>
</div>
</div>
</div>
</div>
<div class="col">
    <div class="card h-100 border-0 shadow-sm p-3">
        <div class="card-body">
            <p class="card-title">Drain Pump / Pompa Hisap</p>
            <div class="row">
                <div class="col">
                    <div class="d-grid gap-2">
                        <a href="api/update.php?relay_drain=ON" class="btn btn-primary">ON</a>
                    </div>
                </div>
                <div class="col">
                    <div class="d-grid gap-2">
                        <a href="api/update.php?relay_drain=OFF" class="btn btn-danger">OFF</a>
                    </div>
                </div>
            </div>
            <p class="mt-3">Status:
                <strong>
                    <?php
                    if ($relay_drain == 'ON') {
                        echo "<span class='text-primary'>" . $relay_drain . "</span>";
                    } else if ($relay_drain == 'OFF') {
                        echo "<span class='text-danger'>" . $relay_drain . "</span>";
                    }
                    ?>
                </strong>
            </p>
        </div>
    </div>
</div>
</div>
</div>
<div class="row row-cols-1 row-cols-md-3 row-cols-2 g-4 ">
    <div class="col">
        <div class="card h-100 border-0 shadow-sm p-3">
            <div class="card-body">
                <p class="card-title">Kelembaban Udara</p>
                <h1 class="card-text text-secondary"><strong id='humidity' class="text-dark">00</strong> %</h1>
            </div>
        </div>
    </div>

```

```

    </div>
</div>
<div class="col">
  <div class="card h-100 border-0 shadow-sm p-3">
    <div class="card-body">
      <p class="card-title">Suhu Lingkungan</p>
      <h1 class="card-text text-secondary"><strong id='temperature' class="text-dark">00</strong> °C </h1>
    </div>
  </div>
</div>
<div class="col">
  <div class="card h-100 border-0 shadow-sm p-3">
    <div class="card-body">
      <p class="card-title">Kelembaban Tanah 1</p>
      <h1 class="card-text text-secondary"><strong id='soil1' class="text-dark">00</strong> %</h1>
    </div>
  </div>
</div>
<div class="col">
  <div class="card h-100 border-0 shadow-sm p-3">
    <div class="card-body">
      <p class="card-title">Kelembaban Tanah 2</p>
      <h1 class="card-text text-secondary"><strong id='soil2' class="text-dark">00</strong> %</h1>
    </div>
  </div>
</div>
<div class="col">
  <div class="card h-100 border-0 shadow-sm p-3">
    <div class="card-body">
      <p class="card-title">Kelembaban Tanah 3</p>
      <h1 class="card-text text-secondary"><strong id='soil3' class="text-dark">00</strong> %</h1>
    </div>
  </div>
</div>
<div class="col">
  <div class="card h-100 border-0 shadow-sm p-3">
    <div class="card-body">
      <p class="card-title">Ketinggian Air</p>
      <h1 class="card-text text-secondary"><strong id='water' class="text-dark">00</strong> cm</h1>
    </div>
  </div>
</div>
</div>
</div>

```

<p class="mt-5">Terakhir Update: <strong id='tanggal'>2021-07-01 00:00:00</p>

```
</div>
<script>
  if (typeof (EventSource) !== "undefined") {
    var source = new EventSource("sse_data.php");
    source.addEventListener('data', function (e) {
      var data = JSON.parse(e.data);
      var tanggal = data.tanggal;
      var waktu = data.waktu;
      var humidity = data.humidity;
      var temperature = data.temperature;
      var soil1 = data.soil1;
      var soil2 = data.soil2;
      var soil3 = data.soil3;
      var water = data.water;

      document.getElementById("humidity").innerText = humidity;
      document.getElementById("temperature").innerText = temperature;
      document.getElementById("soil1").innerText = soil1;
      document.getElementById("soil2").innerText = soil2;
      document.getElementById("soil3").innerText = soil3;
      document.getElementById("water").innerText = water;
      document.getElementById("tanggal").innerText = tanggal + " " + waktu;

    }, false);

  } else {
    document.getElementById("result").innerHTML = "Not Support";
  }
</script>
<?php include('logout_modal.php') ?>
<?php include('layout/footer.php') ?>
```

4. grafik.php

```
<?php
session start();
if ($_SESSION['status'] != 'login') {
    header("location: index.php?pesan=belum_login");
}
include('config/connection.php');
$query = "SELECT * FROM greenhouse ORDER BY id_data DESC LIMIT 25";
$waktu = mysqli_query($conn, $query);
$humidity = mysqli_query($conn, $query);
$temperature = mysqli_query($conn, $query);
$soil1 = mysqli_query($conn, $query);
$soil2 = mysqli_query($conn, $query);
$soil3 = mysqli_query($conn, $query);
$water = mysqli_query($conn, $query);
?>

<?php include('layout/header.php') ?>
<?php include('layout/navbar.php') ?>

<div class="container">

    <h1 class="mb-3">Grafik</h1>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

    <div class="row row-cols-1 row-cols-md-2 g-4">

        <!-- //-----GRAFIK HUMIDITY----- -->
        <div class="col">
            <div class="card tr-card h-100 border-0 shadow-sm p-3 p-3">
                <div class="card-body">
                    <h5 class="card-title">Kelembaban Udara (%)</h5>
                    <div>
                        <canvas id="humidityChart"></canvas>
                    </div>
                </div>
            </div>
        </div>

        <!-- /-----GRAFIK TEMPERATURE----- -->
        <div class="col">
            <div class="card tr-card h-100 border-0 shadow-sm p-3 p-3">
                <div class="card-body">
                    <h5 class="card-title">Suhu Lingkungan (°C)</h5>
                    <div>
                        <canvas id="temperatureChart"></canvas>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

</div>

<!-- //-----GRAFIK SOIL1----- -->
<div class="col">
  <div class="card tr-card h-100 border-0 shadow-sm p-3 p-3">
    <div class="card-body">
      <h5 class="card-title">Kelembaban Tanah 1 (%)</h5>
      <div>
        <canvas id="soil1Chart"></canvas>
      </div>
    </div>
  </div>
</div>

<!-- /-----GRAFIK SOIL2----- -->
<div class="col">
  <div class="card tr-card h-100 border-0 shadow-sm p-3 p-3">
    <div class="card-body">
      <h5 class="card-title">Kelembaban Tanah 2 (%)</h5>
      <div>
        <canvas id="soil2Chart"></canvas>
      </div>
    </div>
  </div>
</div>

<!-- //-----GRAFIK SOIL3----- -->
<div class="col">
  <div class="card tr-card h-100 border-0 shadow-sm p-3 p-3">
    <div class="card-body">
      <h5 class="card-title">Kelembaban Tanah 3 (%)</h5>
      <div>
        <canvas id="soil3Chart"></canvas>
      </div>
    </div>
  </div>
</div>

<!-- /-----GRAFIK WATER----- -->
<div class="col">
  <div class="card tr-card h-100 border-0 shadow-sm p-3 p-3">
    <div class="card-body">
      <h5 class="card-title">Ketinggian Air (cm)</h5>
      <div>
        <canvas id="waterChart"></canvas>
      </div>
    </div>
  </div>
</div>
</div>

```

```

<script>
  const labels = [
    <?php
      while ($row = mysqli_fetch_array($waktu)) {
        echo "''" . $row['waktu'] . "',";
      }
    ?>
  ]

  // -----GRAFIK HUMIDITY-----
  const dataHumidity = {
    labels: labels,
    datasets: [{
      label: 'Gafik Humidity',
      backgroundColor: 'rgb(255, 99, 132)',
      borderColor: 'rgb(255, 99, 132)',
      data: [
        <?php
          while ($row = mysqli_fetch_array($humidity)) {
            echo $row['humidity'] . ", ";
          }
        ?>
      ],
    }
  ]
};

const configHumidity = {
  type: 'line',
  data: dataHumidity,
  options: {}
};

const humidityChart = new Chart(
  document.getElementById('humidityChart'),
  configHumidity
)

// -----GRAFIK TEMPERATURE-----
const dataTemperature = {
  labels: labels,
  datasets: [{
    label: 'Gafik Temperature',
    backgroundColor: '#36a2eb',
    borderColor: '#36a2eb',
    data: [
      <?php
        while ($row = mysqli_fetch_array($temperature)) {
          echo $row['temperature'] . ", ";
        }
      ?>
    ],
  }
]
};

```

```

const configTemperature = {
  type: 'line',
  data: dataTemperature,
  options: {}
};

const temperatureChart = new Chart(
  document.getElementById('temperatureChart'),
  configTemperature
)

// -----GRAFIK SOIL1-----
const dataSoil1 = {
  labels: labels,
  datasets: [{
    label: 'Gafik Soil1',
    backgroundColor: '#6FCDCD',
    borderColor: '#6FCDCD',
    data: [
      <?php
        while ($row = mysqli_fetch_array($soil1)) {
          echo $row['soil1'] . ",";
        }
      ?>
    ],
  }]
};

const configSoil1 = {
  type: 'line',
  data: dataSoil1,
  options: {}
};

const soil1Chart = new Chart(
  document.getElementById('soil1Chart'),
  configSoil1
)

// -----GRAFIK SOIL2-----
const dataSoil2 = {
  labels: labels,
  datasets: [{
    label: 'Gafik Soil2',
    backgroundColor: '#8F57FF',
    borderColor: '#8F57FF',
    data: [
      <?php
        while ($row = mysqli_fetch_array($soil2)) {
          echo $row['soil2'] . ",";
        }
      ?>
    ]
  }]
};

```



```

    ],
  ]]
};

const configSoil2 = {
  type: 'line',
  data: dataSoil2,
  options: {}
};

const soil2Chart = new Chart(
  document.getElementById('soil2Chart'),
  configSoil2
)

// -----GRAFIK SOIL3-----
const dataSoil3 = {
  labels: labels,
  datasets: [{
    label: 'Gafik Soil3',
    backgroundColor: '#FFA449',
    borderColor: '#FFA449',
    data: [
      <?php
        while ($row = mysqli_fetch_array($soil3)) {
          echo $row['soil3'] . ",";
        }
      ?>
    ],
  }]
};

const configSoil3 = {
  type: 'line',
  data: dataSoil3,
  options: {}
};

const soil3Chart = new Chart(
  document.getElementById('soil3Chart'),
  configSoil3
)

// -----GRAFIK WATER-----
const dataWater = {
  labels: labels,
  datasets: [{
    label: 'Gafik Water',
    backgroundColor: '#333538',
    borderColor: '#333538',
    data: [
      <?php
        while ($row = mysqli_fetch_array($water)) {

```

```
                echo $row['water'] . ",";
            }
            ?>
        ],
    ]];
};

const configWater = {
    type: 'line',
    data: dataWater,
    options: {}
};

const waterChart = new Chart(
    document.getElementById('waterChart'),
    configWater
)

</script>
</div>
<?php include('logout_modal.php') ?>
<?php include('layout/footer.php') ?>
```

5. data.php

```
<?php
session start();
if ($_SESSION['status'] != 'login') {
    header("location: index.php?pesan=belum_login");
}
?>
<?php include('layout/header.php') ?>
<?php include('layout/navbar.php') ?>
<?php
include('config/connection.php');
$query = "SELECT * FROM greenhouse";
$read = mysqli_query($conn, $query);
$jumlahData = mysqli_num_rows($read);
?>

<div class="container">

    <h1 class="text-center mb-3">Data Table</h1>

    <h2 class="text-center mt-4 mb-4"></h2>

    <p>Total Data: <strong>
        <?php echo $jumlahData ?>
    </strong></p>
    <div class="card border-0 shadow-sm p-3">
        <div class="card">

            <div class="card-header">
                <!-- <div class="row">
                    <div class="col col-md-6">Export HTML table data to excel using JavaScript</div>
                    <div class="col col-md-6 text-right">
                        <button onclick="ExportToExcel('xlsx')">Export table to excel</button>
                    </div>
                </div> -->
            </div>
        </div>

        <div class="card-body">
            <table id="example" class="display" style="width:100%">
                <!-- <table id="data_greenhouse" class="table table-striped"> -->
                <thead>
                    <tr>
                        <th scope="col">NO</th>
                        <th scope="col">Tanggal</th>
                        <th scope="col">Waktu</th>
                        <th scope="col">Kelembaban Udara (%)</th>
                        <th scope="col">Suhu Lingkungan (°C)</th>
                        <th scope="col">Kelembaban Tanah 1 (%)</th>
                        <th scope="col">Kelembaban Tanah 2 (%)</th>
                        <th scope="col">Kelembaban Tanah 3 (%)</th>
                        <th scope="col">Ketinggian Air (cm)</th>
                    </tr>
                </thead>
            </table>
        </div>
    </div>
</div>
```



```
        dom: 'Bfrtip',
        buttons: [
            'csv', 'excel', 'pdf', 'print'
        ]
    });
</script>
<?php include('logout_modal.php') ?>
<?php include('layout/footer.php') ?>
```

6. ArduinoIDE

```
#define BLYNK_TEMPLATE_ID "TMPL6PDv30Ano"
#define BLYNK_TEMPLATE_NAME "Monitoring Suhu dan Kelembaban Tanah"
#define BLYNK_AUTH_TOKEN "jTTWblpzPjPmcr18tZwLFyNn3SGz3Sh5"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h> //Library untuk DHT
#include <Adafruit MCP3008.h>
Adafruit_MCP3008 adc;
#include <ArduinoJson.h>
#include <ESP8266HTTPClient.h>
#include <BearSSLHelpers.h>
#include <BearSSLWiFiClient.h>
#define DHTPIN D3 //deklarasi pin D0 untuk output dari DHT22
#define DHTTYPE DHT22 //Tipe DHT22
#define selenoid D8
#define push D7
#define drain D6

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;
WidgetLED led1(V6);
WidgetLED led2(V7);

char auth[] = BLYNK_AUTH_TOKEN; //Auth Token
char ssid[] = "UNTIRTA"; //nama hotspot yang digunakan
char pass[] = "untirtajawara"; //password hotspot yang digunakan
String serverUrl = "https://amor.my.id/api/create.php";
const unsigned long jeda = 500;
unsigned long zero = 0;

float soil1 = 0;
float soil2 = 0;
float soil3 = 0;
const int triggerPin = D5;
const int echoPin = D4;
long duration = 0;
float water = 0;
float temperature = 0;
float humidity = 0;

void setup() {
  Serial.begin(9600);
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(selenoid, OUTPUT); //deklarasi relay sebagai output
  digitalWrite(selenoid, HIGH); //
  // pinMode(push, OUTPUT);
  // digitalWrite(push, HIGH);
  // pinMode(drain, OUTPUT); //deklarasi relay sebagai output
```

```

// digitalWrite(drain, HIGH);

Blynk.begin(auth, ssid, pass);

// Menghubungkan ke jaringan Wi-Fi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println("Connecting to WiFi...");
}
Serial.println("Connected to WiFi");
Serial.print("OK! IP=");
Serial.println(WiFi.localIP());

// Membuat objek WiFiClientSecure tanpa verifikasi sertifikat
BearSSL::WiFiClientSecure client;

// Mengatur objek client ke HTTPClient
HTTPClient http;
http.begin(client, serverUrl);

adc.begin();
dht.begin();
}

void loop() {
//-----SOIL MOISTURE-----
int Soil1 = adc.readADC(1);
int Soil2 = adc.readADC(3);
int Soil3 = adc.readADC(5);
soil1 = map(Soil1, 211, 1023, 100, 0); //ganti x dengan nilai saat sensor1 dicelupkan ke air
soil2 = map(Soil2, 142, 1023, 100, 0); //ganti x dengan nilai saat sensor2 dicelupkan ke air
soil3 = map(Soil3, 434, 1023, 100, 0); //ganti x dengan nilai saat sensor3 dicelupkan ke air

Serial.println("Kelembaban Tanah 1: " + String(soil1) + " %");
Serial.println("Kelembaban Tanah 2: " + String(soil2) + " %");
Serial.println("Kelembaban Tanah 3: " + String(soil3) + " %");

//-----HC-SR04-----
digitalWrite(triggerPin, LOW);
delayMicroseconds(2);
digitalWrite(triggerPin, HIGH);
delayMicroseconds(10);
digitalWrite(triggerPin, LOW);
duration = pulseIn(echoPin, HIGH);
water = duration * 0.0343 / 2; //Perhitungan untuk pembacaan water
Serial.print("water: ");
Serial.print(water);
Serial.println(" cm");

//-----DHT-22-----
float temperature = dht.readTemperature();

```

```

float humidity = dht.readHumidity();
Serial.print("% Temperature: ");
Serial.print(temperature);
Serial.println("C ");
Serial.print("% Kelembaban: ");
Serial.print(humidity);
Serial.println("%");
delay(500);

//-----Sending DATA-----
if (millis() - zero >= jeda) {
  HTTPClient http;

  Serial.println("Connecting to server...");
  if (http.begin(client, serverUrl)) {
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    String httpRequestData = String() + "humidity=" + humidity + "&temperature=" + temperature + "&soil1=" + soil1 + "&soil2=" + soil2 + "&soil3=" + soil3 +
"&water=" + water;

    int httpResponseCode = http.POST(httpRequestData);

    if (httpResponseCode > 0) {
      Serial.print("HTTP ");
      Serial.println(httpResponseCode);
      String payload = http.getString();
      Serial.println();
      Serial.println(payload);

      //----- ARDUINO JSON-----
-----;
      StaticJsonDocument<128> doc;
      DeserializationError error = deserializeJson(doc, payload);
      if (error) {
        Serial.print(F("deserializeJson() failed: "));
        Serial.println(error.f_str());
        return;
      }
      String status = doc["status"]; // "OK"
      String relay_solenoid = doc["relay_solenoid"]; // "ON"
      String relay_drain = doc["relay_drain"]; // "ON"
      Serial.println("Status: " + status);
      Serial.println("Relay_solenoid: " + relay_solenoid);
      Serial.println("Relay_drain: " + relay_drain);

      // -----r--- end json-----
-----

    } else {
      Serial.print("Error code: ");
      Serial.println(httpResponseCode);
      Serial.println(":-(");

```



```

    }

    http.end();
} else {
    Serial.println("Unable to begin HTTP connection.");
}

zero = millis();
}

//-----RELAY-----
if (soil1 < 50 && soil2 < 50 && soil3 < 50 || relay_solenoid == "ON") {
    digitalWrite(solenoid, LOW);
    //    digitalWrite(push, LOW);
}
if (water <= 5 || relay_solenoid == "OFF") {
    digitalWrite(solenoid, HIGH);
    digitalWrite(push, HIGH);
}
if (soil1 > 70 && soil2 > 70 && soil3 > 70 || relay_drain == "ON") {
    digitalWrite(drain, LOW);
}
if (water >= 18 || relay_drain == "OFF") {
    //    digitalWrite(drain, HIGH);
}

Blynk.virtualWrite(V0, soil1);
Blynk.virtualWrite(V1, soil2);
Blynk.virtualWrite(V2, soil3);
Blynk.virtualWrite(V4, temperature);
Blynk.virtualWrite(V3, humidity);
Blynk.virtualWrite(V5, water);

Blynk.run();
timer.run();
}

```