

**PENGEMBANGAN SISTEM HIDROPONIK OTOMATIS
BERBASIS IOT UNTUK KONTROL TUMBUHAN PADA SAM
HOUSE HIDROPONIK**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

JUANDA RENALDI 1062143

MICCO AURELDO 1062145

POLITEKNIK MANUFAKTUR NEGERI

BANGKA BELITUNG

TAHUN 2024

LEMBAR PENGESAHAN

PENGEMBANGAN SISTEM HIDROPONIK OTOMATIS BERBASIS IOT UNTUK KONTROL TUMBUHAN PADA SAM HOUSE HIDROPONIK

Oleh :

JUANDA RENALDI /1062143

MICCO AURELDO /1062145

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

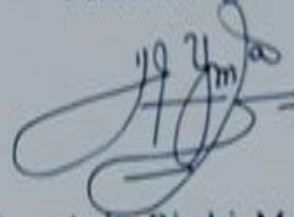
Menyetujui,

Pembimbing 1



Riki Afriansyah, M.T.
NIP. 199004042019031013

Pembimbing 2



Yang Agita Rindri, M.Eng.
NIP. 198609282022032003

Penguji 1



Sidhiq Andriyanto, M.Kom.
NIP. 199007182019031011

Penguji 2



Elisa Mayang Sari, M.Pd.
NIP. 199511282022032018

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini

Nama Mahasiswa 1 : Juanda Renaldi NIM : 1062143

Nama Mahasiswa 2 : Micco Aureldo NIM : 1062145

Dengan Judul : PENGEMBANGAN SISTEM HIDROPONIK
OTOMATIS BERBASIS IOT UNTUK KONTROL
TUMBUHAN PADA SAM HOUSE HIDROPONIK

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata di kemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 24 Juli 2024

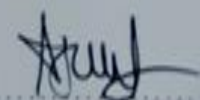
Nama Mahasiswa

Tanda Tangan

1. Juanda Renaldi



2. Micco Aureldo



ABSTRAK

Penelitian ini bertujuan untuk mengembangkan sistem monitoring hidroponik otomatis berbasis Internet of Things yang dapat mengontrol kondisi lingkungan tumbuh tanaman di SAM House Hidroponik. Permasalahan yang dihadapi petani hidroponik saat ini adalah perlunya upaya manual yang intensif untuk memonitor dan mengontrol berbagai parameter lingkungan seperti pH, suhu dan kelembaban. Metode yang digunakan dalam penelitian ini adalah metode prototipe yang melibatkan integrasi sensor pH-4502C dan DHT22 dengan aktuator dan kontroler yang dikendalikan dari aplikasi Android. Sistem ini dirancang untuk memberikan data real-time kepada pemilik SAM House Hidroponik, sehingga pemilik dapat memantau dan mengontrol kondisi lingkungan tanpa perlu hadir secara fisik.

Kata Kunci : Android, Hidroponik, IoT, Monitoring, Sensor.

ABSTRACT

This research aims to develop an Internet of Things-based automatic hydroponic monitoring system that can control the environmental conditions of growing plants in SAM House Hydroponics. The problem faced by hydroponic farmers today is the need for intensive manual efforts to monitor and control various environmental parameters such as pH, temperature and humidity. The method used in this research is the prototype method which involves the integration of pH-4502C and DHT22 sensors with actuators and controllers controlled from Android applications. The system is designed to provide real-time data to the owner of SAM House Hydroponics, allowing the owner to monitor and control environmental conditions without the need to be physically present.

Keywords: Android,, Hydroponics, Monitoring, IoT, Sensor.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang selalu memberikan rahmat, karunia, nikmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan proyek akhir dan penyusunan laporan proyek akhir dengan judul “Pengembangan Sistem Hidroponik Otomatis Berbasis IoT Untuk Kontrol Tumbuhan Pada SAM House Hidroponik”.

Penyusunan proyek akhir ini merupakan syarat yang diperlukan untuk menyelesaikan pendidikan Diploma IV di Politeknik Manufaktur Negeri Bangka Belitung. Dalam penyusunan proyek akhir ini, banyak sekali pihak yang turut membantu sehingga dapat berjalan dengan baik sebagaimana mestinya. Pada kesempatan yang baik dan berbahagia ini, penulis mengucapkan banyak terima kasih kepada semua pihak yang telah berkenan untuk membantu penulis sedari tahap awal hingga terselesaikannya proyek akhir ini dan juga penulis berterima kasih kepada rekan-rekan atas dukungan, bantuan secara moril serta bantuan secara material sehingga penulisan makalah ini dapat diselesaikan dengan baik. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada :

1. Ucapan syukur kepada Tuhan Yang Maha Esa.
2. Ucapan terkhusus dan teristimewa kepada kedua Orang Tua, Kakak, Adik, serta Keluarga yang selalu senantiasa memberikan perhatian, kasih sayang, doa yang penuh dengan keikhlasan serta dukungan baik secara moril ataupun material.
3. Bapak Riki Afriansyah, M.T. selaku pembimbing I yang telah membantu baik dari segi waktu, dukungan, solusi dan saran.
4. Ibu Yang Agita Rindri, M.Eng. selaku pembimbing II yang telah membantu baik dari segi waktu, dukungan, solusi dan saran.
5. Bapak Muhammad Setya Pratama, M.Si. selaku dosen wali kelas IV Teknologi Rekayasa Perangkat Lunak B.

6. Ratu Syifa dan Dede Ari Setyawan selaku narasumber sekaligus pemilik dari SAM House Hidroponik yang telah membantu dan memberikan waktu luang, informasi dan data yang diperlukan.
7. Teman seperjuangan DIV Teknologi Rekayasa Perangkat Lunak yang telah bersedia membantu baik suka dan duka bersama-sama dalam pengerjaan proyek akhir.
8. Semua pihak yang telah berkenan untuk membantu penulis yang tidak dapat disebutkan satu persatu, baik itu secara langsung ataupun bantuan tidak langsung.

Dalam penyusunan laporan proyek akhir ini, penulis menyadari bahwa masih banyak sekali kekurangan serta kesalahan, maka penulis mohon maaf yang sebesar-besarnya. Hal ini dikarenakan keterbatasan, kemampuan dan juga pemahaman yang kurang. Penulis juga sangat antusias dan terima baik atas kritik dan saran apabila terdapat kesalahan penulisan pada laporan proyek akhir. Hal ini demi perbaikan yang bersifat membangun serta memberikan pemahaman yang lebih baik yang belum penulis begitu mengerti dan pahami.

Demikian laporan yang penulis buat, penulis mengucapkan terima kasih dan berharap semoga dengan adanya proyek akhir yang penulis buat dapat menambah wawasan untuk kita semua dan teruntuk para pembaca.

Sungailiat, 24 Juli 2024

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN BUKAN PLAGIAT	Error! Bookmark not defined.
ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Proyek Akhir	4
BAB II DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Hidroponik	10
2.2.1 Sistem NFT (Nutrient Film Technique System)	12
2.3 Sistem Monitoring Otomatis	13
2.4 Pertumbuhan Tanaman	14
2.5 Regresi Linier	16
2.6 Mikrokontroler dan Komponen Pendukungnya	17
2.6.1 Sensor Suhu (DHT22)	17

2.6.2 Sensor pH (pH-4502C)	19
2.6.3 Arduino Uno R3	22
2.6.4 ESP32	23
2.7 Firebase	24
2.8 Android Studio	25
2.9 Internet of Things (IoT)	26
BAB III METODE PELAKSANAAN	27
3.1 Studi Literatur	28
3.2 Perancangan dan Perakitan <i>Hardware</i>	28
3.2.1 Desain Rangkaian Elektrik	28
3.2.2 Perakitan Rangkaian Elektrik	29
3.3 Pengujian dan Kalibrasi Sensor	29
3.4 Perancangan dan Pembuatan Sistem Monitoring dan Aplikasi Android	30
3.4.1 Desain Software	30
3.4.2 Pembuatan Software	31
3.5 Pengujian Sistem Monitoring dan Aplikasi IoT	31
3.6 Pengujian Alat Secara Keseluruhan	32
3.7 Analisa Data	33
3.8 Pembuatan Laporan Akhir dan Publikasi	33
BAB IV PEMBAHASAN	35
4.1 Deskripsi Alat	35
4.2 Proses Kalibrasi dan Pengujian Sensor	37
4.3 Kalibrasi Sensor pH-4502C	37
4.4 Pengujian Sensor pH	41
4.5 Kalibrasi Sensor Suhu & Kelembapan DHT 22	43

4.6 Pengujian Sensor Suhu & Kelembapan DHT22	44
4.7 Pengujian Sensor pH, Sensor Suhu & Kelembapan Bersamaan	46
4.8 Pengujian Aplikasi	47
4.8.1 Pengujian Pembacaan Sensor Pada Aplikasi Android	48
4.9 Pengujian Keseluruhan Alat pada Sistem Monitoring Hidroponik.....	49
4.9.1 Pengujian Pembacaan Sensor LCD	49
4.9.2 Pengujian Monitoring Pada Aplikasi	50
BAB V PENUTUP.....	52
5.1 Kesimpulan.....	52
5.2 Saran.....	53
Daftar Pustaka	54

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka.....	5
Tabel 2.2 Datasheet Sensor DHT22.....	19
Tabel 2.3 Data Sheet Sensor pH-4502C	21
Tabel 2.4 Perbandingan ESP 32 Dengan Mikrokontroller Lain	24
Tabel 4.1 Perbandingan antara nilai ADC sensor pH dan nilai pH meter	39
Tabel 4.2 Perhitungan Regresi Linear pH Sensor	39
Tabel 4.3 Data Pengujian Sensor pH	42
Tabel 4.4 Perbandingan Suhu Data Sensor & Alat Ukur.....	44
Tabel 4.5 Perbandingan Kelembapan Data Sensor & Alat Ukur.....	45
Tabel 4.6 Perbandingan Kelembapan Data Sensor & Alat Ukur.....	46
Tabel 4.7 Data Pengujian Sensor – Aplikasi Android	48
Tabel 4.8 Data Pengujian Seluruh Sensor.....	49
Tabel 4.9 Tabel Akurasi Pembacaan Keseluruhan Sistem.....	50
Tabel 4.10 Perbandingan Pembacaan Sensor di LCD dengan Aplikasi Android .	51

DAFTAR GAMBAR

Gambar 2.1 Sistem NFT (Nutrient Film Technique System)	13
Gambar 2.2 Grafik Regresi Linear	16
Gambar 2.3 Sensor DHT 22	18
Gambar 2.4 Derajat Keasaman (pH)	20
Gambar 2.5 Sensor pH (pH-4502C)	21
Gambar 2.6 Arduino Uno R3	23
Gambar 2.7 ESP 32	23
Gambar 3.1 Alur Pengerjaan Tugas Akhir	29
Gambar 3.2 Skematik Rangkaian	31
Gambar 4.1 Alur Kerja Rangkaian	36
Gambar 4.2 Use Case Diagram	37
Gambar 4.3 Kalibrasi Sensor pH-4502C	38
Gambar 4.4 Kode Kalibrasi Sensor pH	39
Gambar 4.5 Grafik Regresi Linear pH Sensor	40
Gambar 4.6 Program Arduino Pembacaan Nilai pH dengan Regresi Linier	41
Gambar 4.7 Sampel Air Pengujian Sensor	42
Gambar 4.8 Tampilan Serial Monitor Pengujian pH	42
Gambar 4.9 Pengujian Sensor pH dan pH meter	43
Gambar 4.10 Rangkaian Skematik DHT22	43
Gambar 4.11 Proses Pengujian Suhu dan Kelembapan Menggunakan Hygrometer	45

Gambar 4.12 Proses Pengujian Bersamaan.....	46
Gambar 4.13 Hasil Pembacaan Semua Sensor Pada LCD.....	47
Gambar 4.14 Hasil Pembacaan Semua Sensor Pada Aplikasi.....	47
Gambar 4.15 Tampilan LCD Pengujian Sensor Keseluruhan.....	48
Gambar 4.16 Hasil Pembacaan di Aplikasi Android.....	50



DAFTAR LAMPIRAN

Lampiran 1 Riwayat Hidup.....	60
Lampiran 2 Dokumentasi Kegiatan.....	63
Lampiran 3 Kode Pemrograman.....	64



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Selain dikenal sebagai negara maritim, Indonesia juga disebut sebagai negara Agraris, hal ini ditandai dengan sektor pertanian yang menjadi tulang punggung perekonomian Indonesia. Tercatat berdasarkan data Badan Pusat Statistik (BPS) tahun 2021 kontribusi sektor pertanian terhadap Produk Domestik Bruto (PDB) Nasional mencapai 13,28% . Jika dikonversikan ke dalam rupiah jumlah tersebut setara dengan Rp 2,25 kuadriliun(Kusnandar, 2022). Meskipun memiliki kontribusi yang besar, sektor pertanian pada dasarnya menghadapi sejumlah tantangan diantaranya adalah kepemilikan lahan yang terbatas serta terfragmentasi. Masyarakat kini juga mulai sadar akan pola makanan bergizi yang bebas dari bahan kimia serta pestisida (Mamatha & Kavitha, 2023).

Banyak dari lahan pertanian di Indonesia kini beralih fungsi, beberapa diantaranya menjadi kawasan industri hingga kompleks komersial. Tercatat dalam kurun waktu lima tahun terakhir sudah terjadi alih fungsi lahan pertanian sebesar 668.145 hektare (Ardiansyah, 2021). Jika alih fungsi lahan pertanian ini terus dibiarkan maka akan berdampak kapasitas produksi pertanian yang menurun karena berkurangnya lahan pertanian, serta munculnya beberapa permasalahan lain seperti perubahan iklim, pencemaran tanah dan faktor lainnya (Khan et al., 2020). Dalam menghadapi permasalahan tersebut sejatinya dibutuhkan sebuah inovasi untuk mengatasi alih fungsi lahan pertanian ini, salah satu metode yang bisa digunakan adalah ‘Hidroponik’.

Hidroponik merupakan salah satu metode budidaya pertanian yang memanfaatkan air sebagai medium tanpa menggunakan tanah (Roidah, 2014). Selanjutnya (Rosliani & Sumarni, 2005) juga menyatakan bahwa hidroponik adalah metode bertanam tanpa menggunakan tanah sebagai media melainkan menggunakan larutan nutrisi yang mengandung garam organik sebagai media tanam untuk menumbuhkan sistem akar yang ideal. Meskipun menawarkan kemudahan karena

tidak menggunakan tanah sebagai media tanamnya, sebagai gantinya dalam metode penanaman menggunakan sistem hidroponik ini terdapat beberapa hal penting yang harus diperhatikan dalam pengembangannya diantaranya air, cahaya matahari, suhu, kelembapan serta pH optimal (Dadang ITS, 2017). Namun untuk mengetahui kondisi diatas agar lingkungan tumbuh hidroponik menjadi optimal maka perlu dilakukan pemantauan secara berkala. Pemantauan secara berkala ini tentunya membawa permasalahan bahwa petani perlu secara rutin melakukan pengecekan secara berkala yang dimana ini menyebabkan waktu dan tenaga terbuang secara cuma-cuma.

Permasalahan diatas dapat diatasi dengan adanya integrasi IoT terhadap sistem hidroponik. Dengan penggunaan teknologi mumpuni seperti IoT, petani bisa memperoleh informasi secara *real time* dan dapat membuat keputusan yang tepat terkait lingkungan tumbuh tanaman yang dikelolanya (UMA, 2023). Internet of Things (IoT) adalah jaringan komputer yang terintegrasi dengan mesin, objek, makhluk hidup hingga manusia yang memiliki kemampuan untuk mentransfer data melalui jaringan (Abu Sneineh & Shabaneh, 2023). IoT merupakan teknologi yang memungkinkan terjadinya komunikasi, pengendalian hingga kerja sama dengan berbagai perangkat keras yang terintegrasi melalui sebuah jaringan internet (I. Kurniawan, 2022). Dengan adanya penggunaan IoT dalam pengembangan hidroponik maka memungkinkan para petani untuk melakukan pemantauan dan kontrol lingkungan tumbuh tanaman hidroponik secara *real-time* dimana saja dan kapan saja. Penggunaan IoT yang selanjutnya terintegrasi dengan perangkat sensor juga dapat memungkinkan para petani untuk memantau secara berkala kualitas air, kondisi pH optimal pada tanaman, serta suhu dan kelembapan secara jarak jauh. Dalam mendukung pernyataan tersebut penelitian yang dilakukan oleh (Irawan et al., 2023) menyebutkan bahwa efektivitas dari penggunaan IoT memungkinkan pemantauan dan pengendalian parameter lingkungan tumbuh tanaman secara *real-time* dan jarak jauh. Peneliti mengintegrasikan beberapa sensor seperti sensor suhu, kelembapan, dan pH air ke

dalam sistem IoT yang terhubung dengan internet. Data dari sensor-sensor ini dapat diakses melalui aplikasi Android yang dirancang khusus.

Adapun masalah yang ingin dipecahkan dalam penelitian ini bagaimana mengurangi upaya para petani untuk memeriksa setiap elemen yang dibutuhkan dalam tanaman hidroponik tanpa perlu adanya campur tangan petani. Dengan pemanfaatan IoT dan melakukan otomasi pada hidroponik harapannya dapat memberikan kemudahan dalam memantau, menganalisis, serta mengendalikan lingkungan tumbuh tanaman hidroponik secara akurat, optimal dan efisien sehingga mengurangi dampak kerugian, meningkatkan produktivitas dan mengurangi tenaga dan waktu yang dibutuhkan untuk melakukan pemantauan tanaman hidroponik di SAM House Hidroponik Sungailiat.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijelaskan sebelumnya, permasalahan yang diangkat dirumuskan sebagai berikut :

1. Bagaimana mengimplementasikan sensor IoT untuk *monitoring* pada sistem hidroponik di SAM House Hidroponik, dan sensor apa saja yang dibutuhkan untuk memantau kondisi lingkungan tumbuh tanaman hidroponik?
2. Bagaimana integrasi sensor IoT dengan *actuator* dan *kontroller* dapat mengontrol lingkungan tumbuh tanaman hidroponik secara *real-time*?
3. Bagaimana memvisualisasikan data sensor IoT lingkungan tumbuh tanaman dan mengendalikan actuator melalui antarmuka aplikasi Android?

1.3 Batasan Masalah

Dalam proyek akhir ini terdapat batasan masalah yang ditetapkan yakni sistem yang dikembangkan hanya dapat melakukan *monitoring* nilai pH, suhu, dan kelembapan pada instalasi hidroponik.

1.4 Tujuan Proyek Akhir

Berdasarkan rumusan masalah yang telah dirumuskan sebelumnya, berikut merupakan tujuan dari proyek akhir ini :

1. Membangun antarmuka sistem monitoring dan kontrol berbasis aplikasi Android yang memudahkan pengguna dalam pemantauan dan pengendalian lingkungan tumbuh hidroponik.
2. Mengembangkan sistem kontrol lingkungan tumbuh tanaman secara *real time* dengan mengintegrasikan sensor IoT dengan *actuator* dan *kontroller* pada sistem hidroponik.
3. Mengimplementasikan sensor IoT dan sistem monitoring yang sesuai pada sistem hidroponik di SAM House Hidroponik guna memantau kondisi lingkungan tumbuh tanaman.

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Pada proyek akhir ini tinjauan pustaka yang dilakukan penulis memiliki fokus utama pada penelitian terdahulu yang relevan dan sesuai dengan judul penelitian penulis. Berikut pada tabel 2.1 Kajian Pustaka merupakan hasil tinjauan terhadap penelitian yang telah ada sebelumnya :

Tabel 2.1 Kajian Pustaka

No	Judul Penelitian	Hasil Penelitian
1	<i>Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things</i> (Abu Sneineh & Shabaneh, 2023)	Penelitian ini menyajikan desain sekaligus implementasi dari sebuah sistem pemantauan hidroponik yang terintegrasi dengan <i>Internet of Things</i> (IoT). Sistem ini menggunakan mikrokontroler ESP32 dan beberapa sensor seperti sensor TDS, pH, ketinggian air dan suhu untuk memantau parameter penting dalam sistem hidroponik. Sistem ini memungkinkan petani untuk dapat memantau parameter lingkungan dan mengontrol kondisi pertumbuhan tanaman secara <i>real-time</i> dan otomatis dari jarak jauh.
2	Sistem Kontrol dan Monitoring Tanaman Hidroponik Berbasis <i>Internet of Things</i> (IoT) Menggunakan Nodemcu ESP32 (Hidayat &	Penelitian ini membahas mengenai implementasi penggunaan Nodemcu ESP32 untuk melakukan kontrol dan <i>monitoring</i> terhadap sistem hidroponik. Hasil Penelitian yang dilakukan oleh (Hidayat & Amrullah, 2022) menunjukkan

-
- Amrullah, 2022)
- bahwa sistem IoT yang dikembangkan dapat bekerja dengan baik dalam mengontrol pemberian nutrisi tanaman hidroponik secara otomatis berdasarkan data sensor suhu dan pH. Data-data yang diperoleh sensor selanjutnya dapat dikirimkan secara *real-time* ke aplikasi web *monitoring* yang telah dirancang. Secara keseluruhan penelitian ini memberikan kontribusi yang baik dalam menerapkan teknologi IoT untuk meningkatkan efisiensi dan produktivitas sistem hidroponik.
- 3 Penerapan IoT dalam Penelitian ini mengembangkan sebuah Sistem Otomatisasi Kontrol sistem monitoring dan kontrol otomatis Suhu, Kelembaban, dan untuk tanaman hidroponik dengan Tingkat Keasaman menggunakan *Internet of Things* (IoT). Hidroponik (Ridwan & Sistem ini dirancang untuk memantau dan Sari, 2021) mengontrol parameter penting dalam hidroponik seperti suhu, kelembaban udara, tingkat keasaman (pH) air nutrisi. Penelitian ini bertujuan untuk membantu mengatasi kesulitan dalam pemantauan dan pengendalian manual parameter-parameter tersebut. Hasil dari penelitian ini menunjukkan bahwa sistem yang dibangun sudah sesuai dan dapat bekerja dengan baik berdasarkan sensor yang telah terintegrasi untuk memantau parameter-parameter tersebut.
-

4	Sistem Kontrol dan Monitoring Tanaman Hidroponik Aeroponik Berbasis <i>Internet of Things</i> (Putu Denanta & I Nyoman Piarsa, 2020)	Penelitian ini mengembangkan sebuah sistem kontrol serta monitoring tanaman hidroponik dengan teknik aeroponik yang memanfaatkan teknologi <i>Internet of Things</i> (IoT). Sistem yang dikembangkan mengintegrasikan perangkat IoT dengan aplikasi mobile Android sebagai antarmuka kontrol dan monitoring bagi pengguna. Perangkat IoT terdiri dari mikrokontroler Raspberry Pi dan Arduino Mega, serta berbagai sensor seperti TDS, pH dan DHT22. Penelitian ini menunjukkan bahwa sistem dapat melakukan monitoring dan kontrol tanaman hidroponik secara otomatis dengan baik, serta mampu melakukan percampuran nutrisi sesuai kebutuhan tanaman.
5	Narative Review Pemanfaatan <i>Internet of Things</i> Untuk Aplikasi Seed Monitoring and Management System Pada Media Tanaman Hidroponik di Indonesia (Fuada et al., 2023)	Pada penelitian ini disampaikan bahwa IoT dapat menjadi solusi untuk mempermudah monitoring dan pengendalian kondisi pembibitan secara otomatis dan efisien pada tanaman hidroponik. Implementasi dalam hidroponik ini melibatkan berbagai komponen seperti mikrokontroler (ESP32, ESP8266 dan Arduino), sensor suhu kelembapan, pH dan TDS. Pada penelitian ini mengemukakan penggunaan IoT pada hidroponik dapat memberikan banyak manfaat bagi petani mulai dari tahap pembibitan hingga panen secara otomatis

		dan jarak jauh.
6	Sistem Kontrol dan Budidaya hidroponik membutuhkan Monitoring Ph Air serta kontrol yang ketat terhadap faktor-faktor Kepekatan Nutrisi pada seperti nutrisi, pH, dan lain-lain untuk Budidaya Hidroponik Jenis menghasilkan tanaman yang berkualitas. Sayur dengan Teknik <i>Deep Flow</i> Untuk menunjang hal tersebut pemanfaatan sistem otomatisasi dengan <i>Technique</i> (Gregoryan et memanfaatkan teknologi IoT dapat al., 2019) menjadi solusi. Penggunaan sensor seperti TDS, dan pH menunjukkan bahwa sistem yang dirancang dapat bekerja dengan baik dalam melakukan pemantauan dan mengirimkan data untuk selanjutnya diolah dan ditampilkan pada antarmuka website yang sederhana dan mudah dimengerti bagi pengguna.	

Berdasarkan hasil penelitian diatas, penulis berupaya untuk membuat sistem monitoring pada sistem hidroponik dengan memantau dan memastikan kondisi lingkungan tumbuh tanaman terpantau secara real time dan akurat, sehingga pada proses pertumbuhan tanaman didapatkan hasil yang optimal berkat pemantauan berkala terhadap kondisi tumbuh tanaman hidroponik. Beberapa kelebihan dari dari penelitian-penelitian tersebut adalah penggunaan mikrokontroler yang lebih mumpuni seperti Raspberry, kemudian penggunaan sensor yang lebih lengkap yang dibutuhkan untuk pemantauan kondisi lingkungan tanaman. Namun terdapat beberapa kekurangan yang dapat diidentifikasi dari penelitian-penelitian tersebut.

Kelemahan yang dapat penulis identifikasi berdasarkan tinjauan pustaka tersebut meliputi :

1. Beberapa penelitian hanya memanfaatkan IoT dalam hal ini integrasi sensor yang digunakan hanya sebatas untuk uji coba saja dan data sensor pemantauan kondisi lingkungan tumbuhan hidroponik yang meliputi pH, suhu, kelembapan hingga nutrisi, dan lain-lain tidak diproses datanya sehingga data pertumbuhan tanaman hanya sebatas pada uji coba alat dengan memanfaatkan teknologi IoT saja.
2. Beberapa penelitian hanya bersifat *narrative review* saja sehingga potensi bias cukup besar.
3. Beberapa penelitian hanya memiliki fokus pada hal tertentu, seperti melakukan pemantauan kondisi lingkungan tumbuhan hidroponik menggunakan alat yang lengkap namun tidak mempertimbangkan aplikasi sebagai tempat evaluasi dan pengolahan data. Hal ini tentunya tidak relevan apabila alat yang dibuat untuk memperoleh data terkait kondisi lingkungan tumbuh hidroponik tetapi tidak diaplikasikan dengan baik.
4. Beberapa penelitian mungkin kurang dalam hal kustomisasi atau manajemen data sensor. Ini dapat membatasi fleksibilitas pengguna dalam melakukan pengelolaan dan mengonfigurasi sistem. Selain itu beberapa penelitian masih mengandalkan website dan *blynk* sebagai tempat kustomisasi dan manajemen data sensor.

Sementara itu yang menjadi perbedaan penelitian pada proyek akhir ini dengan penelitian yang telah ada sebelumnya adalah sistem yang dikembangkan dan diteliti oleh penulis pada proyek akhir ini yakni sistem otomatisasi yang dibangun akan berfokus pada pemantauan yang mendukung pertumbuhan hidroponik secara optimal yang dalam hal ini konsentrasinya berfokus pada pH, suhu serta kelembapan. Penelitian ini dilaksanakan dengan mengintegrasikan penggunaan IoT ke dalam aplikasi berbasis Android yang dikembangkan langsung oleh penulis yang dimana di dalamnya terdapat fitur tambahan untuk memudahkan pemilik atau pengguna dalam melakukan pencatatan penghasilan dari produksi tanaman hidroponik, melihat riwayat aktivitas data sensor, serta melihat statistik

rata-rata dari nilai sensor yang diperoleh. Adapun mitra dari penelitian ini adalah SAM House Hidroponik.

2.2 Hidroponik

Hidroponik merupakan metode bertanam yang memanfaatkan air sebagai media utama tanpa menggunakan tanah sebagai media tanam namun metode ini menekankan pada pemenuhan kebutuhan nutrisi terhadap tanaman (Arianty, 2019). Selain itu dalam buku “Hidroponik Sayuran” yang ditulis oleh Kunto Herwibowo dan N.S. Budianadi dijelaskan bahwa hidroponik berasal dari bahasa Yunani yakni *hydro* yang berarti air dan *ponos* yang berarti daya. Dengan demikian hidroponik dapat diartikan sebagai *soles culture* atau budidaya tanaman tanpa tanah. Hal yang sama juga dikemukakan oleh (Puput Alviani - Google Buku, n.d.) dalam bukunya yang berjudul “Hidroponik Untuk Pemula” dijelaskan bahwa hidroponik berasal dari bahasa Yunani yakni *hydro* dan *ponos*, yang berarti hidroponik adalah budi daya tanpa menggunakan tanah, namun dengan menggunakan air dan menekankan pada aspek pemenuhan nutrisi tanaman.

Istilah hidroponik pertama kali muncul pada tahun 1936, sebagai bentuk penghargaan terhadap Dr. W.F. Gericke, seorang agronomis dari Universitas California (Swastika et al., 2017). Hidroponik yang dikenal sebagai cara menanam tanaman dengan media cair atau tanpa media tanah ini sangat membantu penduduk dengan intensitas aktivitas yang tinggi namun memiliki lahan yang sangat minim untuk berkebun (Saputra, 2021). Hidroponik ini dapat dilakukan di lahan yang sempit dengan hanya bermodalkan beberapa media tanam saja, namun hidroponik ini memiliki nilai bisnis yang dapat dipertimbangkan karena dapat dilakukan di berbagai tempat seperti pekarangan rumah, dinding-dinding rumah ataupun dengan memanfaatkan lahan lainnya (Barri et al., 2023). Hal ini bertujuan untuk memudahkan masyarakat agar dapat menanam baik itu sayuran dan buah yang dapat digunakan sebagai bahan makanan sehari-hari hingga dipasarkan yang memiliki nilai ekonomis.

Sistem hidroponik merupakan salah satu alternative dalam melakukan budidaya tanaman 3M (Murah, Mudah dan Manfaat) serta bersih, sehingga dapat diaplikasikan dimanapun, dalam kondisi apapun, oleh siapapun dan kapanpun (Saputra, 2021). Dilihat dari arti kata *hydroponos* yang berarti memanfaatkan air sebagai media tanam, maka bisa disimpulkan bahwa sistem budidaya hidroponik ini tidak membutuhkan lahan yang luas, bahkan untuk lahan berukuran 1x2m sudah bisa digunakan untuk budidaya hidroponik.

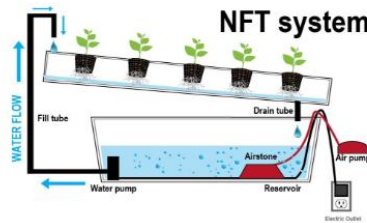
Menurut (Susilawati, 2019) dalam bukunya yang berjudul “Dasar – Dasar Bertanam Secara Hidroponik” terdapat 10 keuntungan dari budidaya hidroponik, diantaranya : (1) sayuran akan tumbuh secara maksimal dengan adanya ketersediaan nutrisi yang terjamin, sehingga daun sayur akan lebih lebat daripada menggunakan metode penanaman biasa, (2) tanaman bisa ditanam secara ditumpuk atau bertingkat, sehingga mudah untuk diaplikasikan pada lahan terbatas seperti lingkungan pada penduduk dan apartemen, (3) sistem hidroponik bisa dimanfaatkan sebagai media edukasi bagi anak-anak terutama menekankan pada pentingnya pertanian untuk hidup sehat, (4) dengan sistem hidroponik, tanaman bisa tumbuh dengan efektif dengan jumlah air dan nutrisi yang tepat, bahkan efisiensi dalam penggunaan air sebagai media tanam mencapai 10% dari air yang dibutuhkan tanaman untuk tumbuh ditanah, (5) keadaan pada lingkungan budidaya hidroponik dapat dilakukan pengontrolan dengan menempatkannya dalam rumah kaca, sehingga kebutuhan seperti herbisida dan pestisida bisa dikurangi, (6) memiliki sistem budidaya hidroponik adalah pengalaman yang unik dan mengesankan, (7) budidaya tanaman hidroponik tidak menggunakan tanah, sehingga hidroponik dapat dilakukan di daerah bahkan dengan kondisi tanah yang buruk atau tidak ada tanah sekalipun, (8) dengan adanya sistem budidaya hidroponik maka memungkinkan kita untuk menanam buah serta sayuran dengan hasil produksi maksimum dan tidak bergantung pada kondisi cuaca, (9) memiliki sistem hidroponik dapat menjamin ketersediaan sayur dan buah segar secara teratur di rumah, dan (10) dengan menggunakan sistem hidroponik kita tidak menghasilkan limbah yang berbahaya bagi lingkungan.

(Susilawati, 2019) juga menyampaikan bahwa setidaknya terdapat enam metode dalam pengembangan hidroponik yakni 1) Sistem Sumbu (*Wick System*), 2) Sistem Rakit Apung (*Water Culture System*), 3) Sistem NFT (*Nutrient Film Technique System*), 4) Sistem Irigasi Tetes (*Drip Irrigation System*), 5) Sistem Pasang Surut (*Ebb and Flow System*) dan 6) Sistem Aeroponik.

2.2.1 Sistem NFT (*Nutrient Film Technique System*)

Nutrient Film Technique System merupakan sistem hidroponik yang dikenalkan pertama kali oleh Dr. Allen Cooper di Glasshouse Crops Research Institute, Inggris (Susilawati, 2019). Sistem hidroponik NFT merupakan teknik yang mengalirkan nutrisi pada ketinggian ± 3 mm dari akar tanaman hidroponik. Sistem ini dirangkai menggunakan pipa PVC atau talang air dengan kecepatan nutrisi yang mengalir pada tanaman (Wati & Sholihah, 2021). Adapun tanaman yang dibudidayakan diberikan sirkulasi hara secara terus-menerus pada tiap talang atau pipa (Barri et al., 2023).

Pada hidroponik NFT yang perlu diperhatikan adalah sistem mampu menyediakan kebutuhan air dan nutrisi bagi tanaman yang dimana setiap salurannya akan mengalirkan secara konstan dengan mempertahankan kandungan nutrisi (M. Majid et al., 2021), bisa dikatakan bahwa sistem NFT memanfaatkan air yang tersirkulasi selama 24 jam tanpa henti sehingga akar tanaman terus memperoleh suplai air, nutrisi dan oksigen untuk mengoksidasi pertumbuhan tanaman sehingga menghasilkan tanaman yang baik dengan masa panen yang relative singkat (Primawati et al., 2021)



Gambar 2.1 Sistem NFT (Nutrient Film Technique System)

2.3 Sistem Monitoring Otomatis

Sistem monitoring otomatis merupakan ekosistem lanjutan dari pertanian hidroponik. Pada sistem ini monitoring tanaman yang biasanya dilakukan secara manual oleh manusia langsung terlaksana secara otomatis dengan memanfaatkan teknologi berbasis IoT (Kuswara & Syahfiqri, 2023). Penggunaan IoT dalam hidroponik memungkinkan aktivitas pemantauan dapat dilakukan tanpa campur tangan manusia. Beberapa penelitian terdahulu tentang sistem monitoring otomatis pada hidroponik sudah pernah dilakukan oleh (Prayitno et al., 2017) dengan judul penelitian “Sistem Monitoring Suhu, Kelembapan, dan Pengendali Penyiraman Tanaman Hidroponik Menggunakan Blynk Android” pada tahun 2017, kemudian (Fauzan & Fahlefi, 2022) dengan judul penelitian “Sistem Monitoring Hidroponik Berbasis Arduino Uno” pada tahun 2022 serta (Diantoro et al., 2022) dengan judul penelitian “Rancang Bangun Sistem Kontrol Otomatis Pada Pengairan Hidroponik Dan Monitoring Berbasis Iot” pada tahun 2022.

Dari ketiga judul penelitian diatas, para peneliti berhasil untuk membuat sistem monitoring otomatis terhadap lingkungan tanaman hidroponik dengan memanfaatkan teknologi IoT. Pada penelitian yang dilakukan oleh (Prayitno et al., 2017) sistem monitoring yang dibangun menggunakan sensor DHT11 dan aplikasi blynk sebagai pengolah data sensor. Selanjutnya pada penelitian (Fauzan & Fahlefi, 2022) sistem monitoring yang dibangun berfokus pada sensor TDS (*Total Dissolved Solids*) untuk melakukan pengecekan terhadap pH terlarut dan juga terdapat sensor suhu, namun pada penelitian ini tidak digunakan aplikasi berbasis android ataupun website untuk menampung dan menangkap data sensor

sehingga hanya sebatas pada pembacaan kondisi lingkungan hidroponik saja. Kemudian pada penelitian (Diantoro et al., 2022) sistem yang dibangun menggunakan sensor pH dan sensor level air dan memanfaatkan blynk sebagai pengolah data sensor.

Berdasarkan tiga penelitian di atas maka dapat disimpulkan bahwa sistem monitoring otomatis yang dibangun masih memiliki beberapa unsur yang belum disediakan pada penelitian sebelumnya. Penelitian ini, bermaksud untuk melakukan modifikasi terhadap penelitian yang sudah ada dengan memperbarui tipe sensor yang digunakan, serta merancang aplikasi android untuk menangkap dan menyimpan data sensor yang didalamnya juga dilengkapi dengan fitur monitoring pencatatan jumlah produksi tanaman hidroponik. Penelitian ini memfokuskan monitoring pada pH menggunakan sensor pH 4502C-Blue, kemudian suhu dan kelembapan menggunakan sensor DHT22. Pemfokusan ini karena berdasarkan (Dadang ITS, 2017) bahwa dalam sistem hidroponik ini terdapat beberapa hal penting yang harus diperhatikan dalam pengembangannya diantaranya air, cahaya matahari, suhu, kelembapan serta pH optimal

2.4 Pertumbuhan Tanaman

Pertumbuhan tanaman merupakan sebuah proses dimana terjadi perubahan terhadap ukuran organ-organ tanaman akibat adanya penambahan ukuran sel tanaman yang mencerminkan adanya proses pertumbuhan (Azmin et al., 2020). Menurut (Azmin, 2015) dan (Rahmawati et al., 2019) keberhasilan dari pertumbuhan suatu tanaman dapat dilihat dari beberapa aspek, diantaranya adalah pertumbuhan tinggi tanaman, jumlah daun, luas daun kemudian berat basa dan juga kering tanaman. Dengan demikian pertumbuhan tanaman dapat kita artikan sebagai proses kenaikan pada massa dan juga volume tanaman yang bersifat *irreversible* (tidak dapat kembali ke semula) seperti tinggi, panjang dan lebar pada bagian-bagian tumbuhan (Maghfiroh & Belakang, 2017).

Pada dasarnya proses pertumbuhan tanaman sangat dipengaruhi oleh kondisi lingkungannya (Maghfiroh & Belakang, 2017). Lingkungan adalah faktor

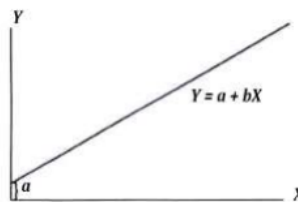
eksternal yang sangat mempengaruhi kondisi pertumbuhan tanaman, apabila kondisi lingkungan tanaman tidak sesuai dengan karakteristik tumbuh tanaman maka akan mengganggu proses pertumbuhan tanaman. Kondisi lingkungan yang merupakan faktor eksternal ini terdiri dari intensitas cahaya matahari, suhu dan kelembapan, tekanan udara, air, nutrisi serta adanya mikroorganisme yang berpotensi mengganggu proses pertumbuhan tanaman.

Dalam tanaman hidroponik yang didominasi air sebagai media tanamnya, dengan adanya kekurangan air maka dapat mempengaruhi semua aspek pertumbuhan tanaman, yang di dalamnya meliputi aspek fisiologi, biokimia, anatomi serta morfologi (Nio Song & Banyo, 2011). Kekurangan air dapat menyebabkan penurunan hasil signifikan pada tanaman dan juga dapat menjadi penyebab kematian pada tanaman (Salisbury & Ross, 1992). Ketersediaan air pada tanaman hidroponik merupakan salah satu cekaman yang dapat mengganggu proses pertumbuhan dan perkembangan tanaman hidroponik. Air bagi tanaman hidroponik menjadi penting untuk diperhatikan karena pengaluran nutrisi pada tanaman dilakukan pada media air. Air juga menyusun 85-90% dari berat keseluruhan pada tanaman (Nio Song & Banyo, 2011). Selain halnya air yang berpengaruh pada pertumbuhan tanaman faktor kelembapan, cahaya matahari, suhu dan juga angin turut mempengaruhi (Wimudi & Fuadiyah, 2021).

Sementara itu selain faktor eksternal, terdapat juga faktor internal yang harus diperhatikan dalam pertumbuhan tanaman yakni genetika dan hormon. Adapun hormon yang merupakan faktor internal juga dapat terpengaruh dengan kondisi lingkungan tumbuhan, sehingga perlu diperhatikan bahwa proses pertumbuhan tanaman akan dipengaruhi oleh kedua faktor ini. Dengan demikian dapat disimpulkan bahwa pada dasarnya pada proses pertumbuhan setiap tanaman membutuhkan kondisi lingkungan yang sesuai untuk mendukung pertumbuhan dan perkembangan tanaman. Kondisi lingkungan tempat tanaman bisa saja selalu mengalami perubahan yang dapat mengganggu proses pertumbuhan tanaman. Hal ini menunjukkan bahwa pada setiap tanaman terdapat faktor pembatas dan juga daya toleransi terhadap lingkungannya (Purwadi, 2011)

2.5 Regresi Linier

Regresi Linier merupakan salah satu metode statistika yang biasanya digunakan untuk membentuk sebuah model hubungan antara variable terikat (dependen) dengan satu ataupun lebih variable bebas (independen). Regresi linier merupakan model persamaan yang digunakan untuk menggambarkan hubungan antara variable bebas (predictor/x) dengan sebuah variable bebas (response/y) yang selanjutnya digambarkan dengan garis lurus, seperti pada gambar di bawah ini :



Gambar 2.2 Grafik Regresi Linear

Penggunaan regresi linier setidaknya ditujukan untuk tujuan deskripsi dari suatu fenomena data ataupun kasus yang diteliti, untuk kontrol, serta untuk memprediksi. Persamaan regresi linier secara matematik diekspresikan dengan bentuk $\hat{Y} = a + bX$, yang dimana :

- $\hat{Y}$ = garis regresi
- a = konstanta intersep
- b = konstanta slope/regresi
- X = variable bebas

Selanjutnya untuk menentukan besarnya konstanta a dan b maka persamaan yang digunakan adalah :

$$a = \frac{(\sum Y_i)(\sum X_i^2) - (\sum X_i)(\sum X_i Y_i)}{n \sum X_i^2 - (\sum X_i)^2}$$

$$b = \frac{n(\sum X_i Y_i) - (\sum X_i)(\sum Y_i)}{n \sum X_i^2 - (\sum X_i)^2}$$

Dengan n sebagai jumlah data (D. Kurniawan, 2008). Dalam penelitian ini, penulis menggunakan metode regresi linier dengan tujuan untuk mendapatkan nilai dari persamaan hubungan dua buah variabel. Penggunaan metode ini digunakan pada proses kalibrasi sensor.

2.6 Mikrokontroler dan Komponen Pendukungnya

Mikrokontroler merupakan sebuah komputer kecil (*special purpose computers*) yang terdapat dalam sebuah IC yang berisi CPU, memori, saluran komunikasi serial serta paralel, dan juga terdapat port input/output serta ADC (*Analog Digital Converter*). *Mikrokontroler* dapat digunakan untuk menjalankan suatu tugas ataupun suatu program (Widharma et al., 2022). *Mikrokontroler* bekerja berdasarkan pada program yang ditanamkan ke dalamnya serta program tersebut dibuat dan telah disesuaikan dengan aplikasi yang diinginkan (Dharmawan, 2017). Penggunaan *mikrokontroler* memungkinkan kita dalam melakukan tugas kontrol yang spesifik dan sederhana. Dalam penelitian ini digunakan beberapa *mikrokontroler* dan komponen pendukung untuk melaksanakan penelitian ini yakni

2.6.1 Sensor Suhu (DHT22)

Sensor suhu merupakan sebuah komponen elektronika yang berfungsi untuk mengubah besaran panas menjadi besaran listrik yang selanjutnya mendeteksi apakah terdapat adanya perubahan suhu pada suatu objek (Kuswara & Syahfiqri, 2023). Suhu air menjadi salah satu faktor yang perlu diperhatikan dalam sistem perkebunan hidroponik karena air adalah sumber hara utama bagi tanaman hidroponik. Sensor suhu digunakan untuk mengecek suhu yang terletak pada area sekitar sensor atau jarak yang dekat dengan sensor (Pratama & Permana, 2021). Suhu yang tergolong baik untuk pertumbuhan tanaman berkisar pada 25°C sampai 28°C (Asmbangnirwana et al., 2022). Berikut ini adalah gambar dari Sensor Suhu DHT22.



Gambar 2.4 Sensor DHT 22

Sensor DHT22 disini digunakan untuk mengukur suhu dalam wadah nutrisi tanaman hidroponik, selanjutnya data pembacaan sensor ini akan ditampilkan pada LCD dan aplikasi sistem *monitoring*. Sensor DHT22 sendiri selain berfungsi untuk mengukur suhu dapat juga mengukur kelembapan. Adapun hubungan suhu dan kelembapan sangat berkaitan dalam tanaman hidroponik. Apabila suhu berubah, maka kelembapan juga akan turut berubah, semakin sedikit volume air yang didapat tanaman maka akan menyebabkan suhu udara meningkat (Nasrullah et al., 2011). Sensor DHT22 juga dikenal sebagai AM2302, dimana sensor ini juga memiliki kesamaan seperti DHT11 yang memiliki empat kaki seperti pada gambar 2.4

Berdasarkan penelitian yang dilakukan oleh (Alif et al., 2016) yang berjudul “Perbandingan Kualitas Antar Sensor Suhu dengan Menggunakan Arduino Pro Mini” diketahui beberapa kelebihan dari sensor DHT22 yakni sebagai berikut :

1. Data hasil pengukuran pada sensor sudah berbentuk sinyal digital dengan perhitungan serta konversi yang dilakukan oleh MCU 8-bit.
2. Sensor sudah terkalibrasi secara akurat dengan adanya kompensasi pada suhu di ruang penyesuaian dengan nilai koefisien kalibrasi yang sudah tersimpan dalam memori OTP terpadu.
3. Pada sensor DHT22 rentang hasil pada pengukuran suhu dan kelembapan sensor lebih lebar.
4. Sensor suhu ini sudah mampu mentransmisikan sinyal hasil pengukuran melalui kabel sepanjang 20 meter, sehingga cocok untuk ditempatkan dimana saja. Kemudian jika menggunakan kabel di atas 2 meter maka

sensor membutuhkan *buffer capacitor* 0,33 μ F yang menghubungkan antara kaki tegangan sumber dengan *ground*.

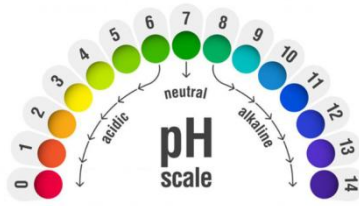
Selanjutnya berikut adalah spesifikasi teknis dari sensor DHT22 / AM-2302 :

Tabel 2.2 Datasheet Sensor DHT22

No	Spesifikasi	Nilai
1	Operating Voltage	3.5V to 5.5V
2	Operating Current	0.3mA (measuring) 60uA (standby)
3	Output	Seria Data
4	Temperature Range	-40°C - 80°C
5	Humidity Range	0% - 100%
6	Resolution	Temperature and Humidity both are 16-bit
7	Accuracy	$\pm 0.5^{\circ}\text{C}$ and $\pm 1\%$

2.6.2 Sensor pH (pH-4502C)

Konsep pH pertama kali dikenalkan oleh seorang kimiawan asal Denmark yakni Soren Peder Lauritz Sorensen pada tahun 1909. pH (*Puissance de Hydrogen*) merupakan derajat keasaman yang digunakan sebagai indicator untuk menyatakan tingkat keasaman atau kebasaan dari suatu larutan (Zulius, 2017). Suatu larutan dapat dikatakan asam atau basa bergantung pada tingkat derajat keasaman yang dimilikinya. Saat ini tingkat derajat keasaman memiliki rentang nilai yang berkisar dari 0 sampai 14 (Kuswara & Syahfiqri, 2023). Derajat keasaman pada larutan netral memiliki nilai 6,5 hingga 7,5. Sementara derajat keasaman yang memiliki nilai lebih dari 7,5 bernilai basa dan derajat keasaman yang memiliki nilai kurang dari 6,5 berarti basa (Azmi, Z., & Saniman, I. 2016).



Gambar 2.5 Derajat Keasaman (pH)

Pada budidaya hidroponik nilai pH yang tidak sesuai akan berpengaruh pada penyerapan unsur hara oleh tanaman. Bilamana kondisi pH pada media tumbuh tanaman bersifat asam, maka akan menyebabkan penyerapan unsur hara oleh tanaman akan kurang maksimal sehingga pertumbuhan tanaman akan terhambat dan menyebabkan tanaman menjadi kerdil Menurut Karoba dan Nurjasmi (2015), ketika pH berada dalam kondisi normal, penyerapan unsur hara oleh tanaman tidak akan terhambat, yang pada gilirannya akan meningkatkan kecepatan pertumbuhan tanaman. Inilah alasan mengapa pH dari larutan nutrisi tanaman menjadi aspek penting dalam budidaya hidroponik.

Sensor pH adalah alat yang digunakan untuk mengukur tingkat keasaman suatu larutan menggunakan sensor yang memenuhi standar industri, menjadikannya komponen penting. Sensor pH, yang terdiri dari elektroda yang sensitif terhadap ion hidrogen, memiliki tingkat impedansi rendah, sehingga dapat memberikan pengukuran yang stabil dan respons yang cepat. Hasil pembacaan dari sensor ini kemudian dapat digunakan oleh mikrokontroler (Syafiqoh et al., 2018). Dalam penelitian ini, penggunaan sensor pH dimaksudkan untuk memonitor pH larutan nutrisi secara real-time, sehingga nilai pH yang menjadi salah satu aspek penting dalam pengembangan tanaman hidroponik dapat terus dipantau.



Gambar 2.6 Sensor pH (pH-4502C)

Pemilihan sensor pH ini didasarkan pada penelitian yang dilakukan oleh (Chuzaini & Dzulkiflih, 2022) dengan judul penelitian “IoT Monitoring Kualitas Air dengan Menggunakan Sensor Suhu , pH , dan Total Dissolved Solids (TDS)”. Pada penelitian ini dibuktikan bahawa tingkat akurasi dari pembacaan sensor pH-4502C memiliki tingkat akurasi 99,851%. Selain itu penelitian yang dilakukan oleh (Pradypta et al., 2022) dengan judul “Rancang Bangun Sistem Monitoring pH Dan Kontrol Suhu Pada Media Pemeliharaan ikan Hias Air Tawar”. Pada penelitian ini juga dibuktikan bahwa sensor pH-4502C memiliki tingkat akurasi sebesar 98,27% dengan persentase error sebesar 1,73%.

Tabel 2.3 Data Sheet Sensor pH-4502C

No	Spesifikasi	Nilai
1	Heating Voltage	0,2V to 5V
2	Operating Current	5-10mA
3	Detectable Concentration Range	pH 0-14
4	Detection Temperature Range	0°C to 80°C
5	Response Time	5s
6	Determination Time	60s
7	Working Temperature	10°C ~ 50°C (Nominal Temperature 20°C)
8	Humidity	95% RH (Nominal Humidity 65% RH)

2.6.3 Arduino Uno R3

Arduino dikembangkan berawal dari *thesis* yang dikemukakan oleh Hernando Barragan pada tahun 2004, seorang mahasiswa asal Colombia. Judul *thesis*nya adalah “Arduino - Revolution Open Hardware”. Arduino sendiri selanjutnya ditemukan oleh Massimo Banzi dan David Cuartielles dengan tujuan membantu para siswa dalam membuat desain perangkat dan interaksi dengan harga yang relative murah, arduino sendiri berasal dari bahasa Italia yang berarti teman yang berani (Abdul Sofyan A, 2017).

Arduino merupakan pengembangan *prototype* yang berbasis pada *mikrokontroller* yang umum digunakan dalam *physical computing*. Arduino dapat dikatakan sebagai media open source yang umumnya digunakan dalam proyek-proyek elektronika. Arduino sendiri tidak hanya sekedar alat pengembangan, tetapi adalah kombinasi dari bahasa pemrograman, *hardware*, serta IDE (*Integrated Development Enviromen*). IDE pada arduino sendiri adalah perangkat lunak yang berfungsi untuk menulis program, meng-compile program menjadi kode biner untuk selanjutnya diupload menuju memori yang terdapat dalam *mikrokontroller* (Susanto et al., 2013).

Arduino sendiri terdiri dari beberapa jenis salah satunya adalah Arduino Uno R3, ini adalah jenis arduino yang dikeluarkan pada tahun 2011. Nama R3 sendiri mengindikasikan jumlah revisi yang ke tiga kalinya (Zanofa et al., 2020). Arduino Uno R3 merupakan sebuah *board sistem minimum* yang berbasis *mikrokontroller* ATmega328P berjenis AVR. Arduino ini memiliki 14 *digital input/output* (6 pin diantaranya digunakan untuk PWM output), 6 analog input, 16 MHz *osilator* Kristal, terdapat *USB connection*, *power jack*, *ICSP header* serta tombol *reset*.

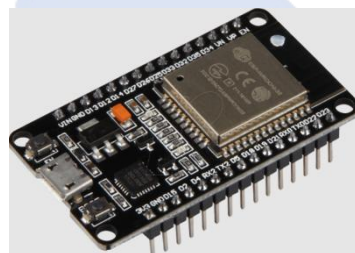


Gambar 2.7 Arduino Uno R3

2.6.4 ESP32

ESP 32 merupakan mikrokontroler yang diperkenalkan oleh *Espressif System*, ESP 32 sendiri merupakan penerus dari mikrokontroler ESP8266. ESP 32 merupakan mikrokontroler yang memiliki beragam keunggulan, diantaranya terdapat pin out dan pin analog yang lebih banyak, terdapat memori yang lebih besar, terdapat opsi *Bluetooth* 4.0 serta terdapat modul WiFi dalam *board* sehingga mendukung untuk membuat sistem yang diintegrasikan dengan penggunaan *Internet of Things* (IoT) (Muliadi et al., 2020)&(Santoso et al., 2022).

Pada penelitian ini penggunaan ESP 32 ditujukan untuk menjadi penghubung antara data sensor yang terbaca pada mikrokontroler Arduino ke database serta aplikasi yang sudah dirancang melalui komunikasi serial.



Gambar 2.8 ESP 32

Berikut adalah perbedaan antara mikrokontroler ESP 32 dengan mikrokontroler lain dipaparkan pada table 2.4

Tabel 2.4 Perbandingan ESP 32 Dengan Mikrokontroler Lain

	Arduino Uno	NODEMCU (ESP8266)	ESP 32
Tegangan	5 V	3,3 V	3,3 V
CPU	ATmega328 - 16MHz	Xtensa single core L106 - 60 MHz	Xtensa dual core LX6 - 160M Hz
Arsitektur	8 bit	32 bit	32 bit
Flash Memori	32kB	16MB	16MB
SRAM	2Kb	160Kb	512Kb
GPIO Pin (ADC/DAC)	14 (6/-)	17 (1/-)	36(18/2)
Bluetooth	Tidak ada	Tidak ada	Ada
WiFi	Tidak ada	Ada	Ada
SPI/I2C/UART	1/1/1	2/1/2	4/2/2

2.7 Firebase

Firebase merupakan sebuah layanan yang disediakan oleh Google yang dirancang untuk mempermudah pengembangan aplikasi. Layanan Firebase pertama kali didirikan pada tahun 2011 oleh Andrew Lee dan James Tamplin lalu selanjutnya diakuisisi oleh Google pada Oktober 2014. Penggunaan layanan jenis ini memungkinkan *developer* untuk fokus pada bagian depan aplikasi tanpa harus memikirkan *backend*. Firebase memungkinkan developer untuk menyimpan berbagai tipe data seperti *string*, *long*, *Boolean* dan sebagainya. Dalam firebase sendiri terdapat dua alternatif database berbasis *cloud* yang bisa digunakan yakni *Realtime Database* dan *Firestore Cloud* (Kuswara & Syahfiqri, 2023).

Firebase dapat digunakan untuk menyinkronkan data secara *real-time* kepada seluruh klien yang terhubung sehingga ini memungkinkan adanya autentikasi pengguna dengan berbagai metode, termasuk email dan password, serta media sosial (Ilhami, 2017). Firebase juga mendukung integrasi dengan berbagai

platform termasuk dalam hal ini adalah Android, iOS, serta website melalui SDK yang disediakan(T. Kurniawan, 2021).

2.8 Android Studio

Android merupakan sebuah platform yang memberikan kecanggihan pada beberapa perangkat seperti ponsel, tablet, TV hingga mobil. Teknologi ini memungkinkan perangkat untuk tidak hanya mengandalkan fungsi, tapi juga dapat membuat hidup menjadi lebih mudah. Android sendiri merupakan sistem operasi mobile yang dikembangkan oleh Google, yang dirancang untuk perangkat mobile seperti *Smartphone*, *tablet*, dan *wearable*. Sistem ini sendiri bersifat *open source* yang dimana memungkinkan pengembang untuk menyesuaikan dan memodifikasi sistem sesuai dengan kebutuhan mereka(Clara, 2024). Dalam perkembangannya Google sendiri merilis sebuah perangkat bernama Android Studio yang digunakan khusus untuk pengembangan aplikasi android, aplikasi ini adalah *Integrated Development Environment* (IDE) yang berdasarkan IntelliJ IDEA. Android Studio sendiri pertama kali rilis pada tanggal 23 September 2009, hingga saat ini sudah banyak versi Android yang terus berkembang dan diberi nama makanan pencuci mulut dan penganan manis.

Selain sebagai editor kode editor IntelliJ IDEA dan alat pengembangan aplikasi yang berdaya guna, android studio menawarkan fungsi yang dapat mendukung produktivitas pengembangan aplikasi *android* diantaranya : 1) Memiliki sistem berbasis gradle yang sangat fleksibel; 2) Memiliki emulator yang cepat dan memiliki banyak fitur; 3) Lingkungan yang menyatu untuk pengembangan semua versi android; 4) Terdapat fitur *Instant Run* untuk mendorong perubahan ke aplikasi yang sedang berjalan tanpa harus membuat aplikasi baru; dan 5) Memiliki alat pengujian yang dilengkapi dengan kerangka kerja yang ekstensif (Herlinah; Musliadi, 2019).

2.9 Internet of Things (IoT)

Internet of Things yang selanjutnya disingkat IoT adalah teknologi yang pertama kali ditemukan oleh Kevin Ashton pada tahun 1999 (Ramdani et al., 2020). IoT berasal dari kata Internet yang berarti sebuah sistem jaringan komputer yang terhubung secara global dengan menganut standar Internet Protocol (TCP/IP) dan things yang mengacu kepada objek fisik yang memungkinkan mereka untuk terhubung dan bertukar melalui internet (Nofrialdi et al., 2023). IoT sendiri merupakan salah satu buah pemikiran yang diciptakan oleh para peneliti untuk mengoptimasi berbagai alat seperti sensor, RFID (*Radio Frequency Identification*), jaringan nirkabel dan hal-hal cerdas lainnya yang mungkin dapat digunakan untuk berkomunikasi dengan mudah serta dengan semua perangkat yang terhubung ke dalam jaringan internet (Chuzaini & Dzulkiflih, 2022).

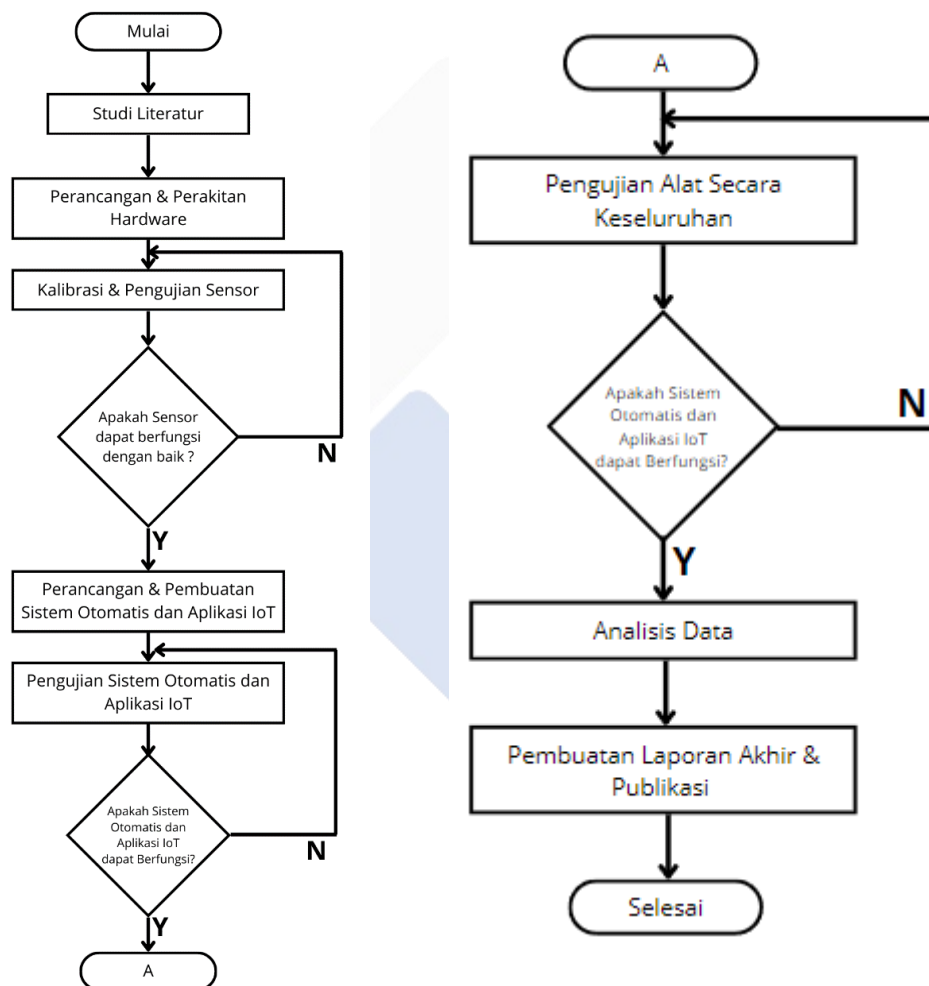
Keberadaan Internet of Things merupakan sebuah paradigma dalam dunia manusia, hal ini dapat dilihat dari keberadaan IoT yang terus berkembang dan terus membuat terobosan baru dan semakin populer penggunaannya di dunia teknologi. Adapun fungsi dari IoT itu sendiri adalah untuk memudahkan pemantauan dan kontrol suatu benda dalam kehidupan sehari-hari (Ramdani et al., 2020). Secara mudahnya prinsip dari teknologi IoT ini bertujuan untuk mempermudah aktivitas sehari-hari yang dilakukan oleh manusia dalam berbagai bidang kehidupannya seperti pendidikan, pertanian, kesehatan, lingkungan, *smart home*, dan sejenisnya (Nursyahbani et al., 2021).

Dengan demikian dapat ditafsirkan bahwa IoT menggambarkan dunia yang dapat terhubung dimana saja dan dapat berkomunikasi dengan cerdas yang belum pernah ada sebelumnya. Dalam penelitian ini penulis akan mengintegrasikan penggunaan IoT dengan sistem hidroponik sehingga dengan adanya keberadaan IoT ini dapat meminimalisir adanya intervensi manual oleh manusia sehingga hal ini akan menghasilkan sistem hidroponik yang lebih cerdas (Ciptadi & Hardyanto, 2018)

BAB III

METODE PELAKSANAAN

Metode pelaksanaan merupakan metode yang digunakan guna menguraikan sistematika dari pelaksanaan proyek akhir dari tahap awal hinggaakhir dalam rangka menyelesaikan permasalahan yang diangkat. Adapun pada bab ini, secara garis besar alur pelaksanaan proyek akhir ini dapat dilihat pada *flowchart* di bawah ini :



Gambar 3.1 Alur Pengerjaan Tugas Akhir

3.1 Studi Literatur

Tahap ini merupakan tahap melakukan identifikasi terhadap permasalahan mengenai implementasi konsep sistem *monitoring* hidroponik otomatis yang terintegrasi dengan IoT. Pengumpulan informasi dilakukan dengan membaca dan mempelajari buku-buku serta jurnal penelitian terkait untuk mengetahui bagaimana konsep dari kebun cerdas, serta bagaimana pengaplikasiannya pada perkebunan hidroponik.

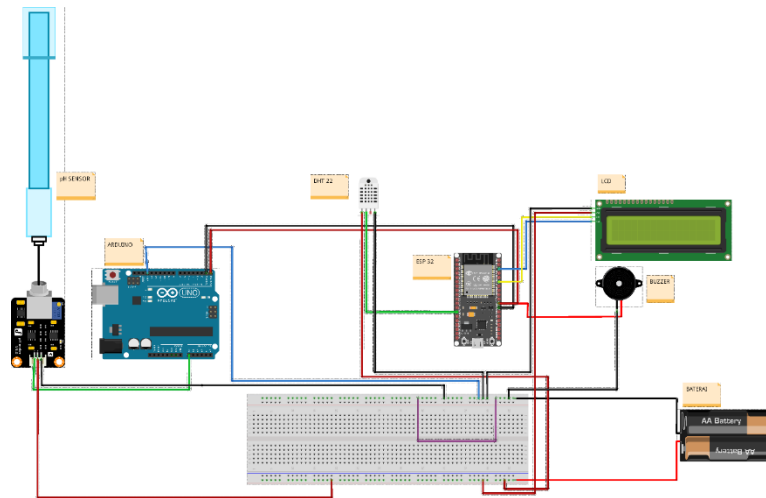
Pencarian informasi mengenai komponen juga untuk dilakukan agar proses pembuatan proyek akhir dapat berjalan dengan lancar dan sistematis. Informasi-informasi tersebut mencakup data jenis sensor, *mikrokontroler*, serta *software* yang digunakan untuk merancang aplikasi *smartphone* nantinya

3.2 Perancangan dan Perakitan Hardware

Pada tahap kedua ini dilaksanakan perancangan dan pembuatan terhadap *hardware* yang akan digunakan dalam proyek akhir ini. Tahapan ini merupakan tahap awal yang akan menentukan gambaran dari proyek akhir yang akan dibuat, perancangan *hardware* ini memiliki fungsi sebagai penentu bagian fisik yang akan dirancang dari alat Sistem Monitoring Otomatis pada Tanaman Hidroponik Berbasis IoT untuk Kontrol Tumbuhan pada SAM House Hidroponik serta menentukan penempatan posisi dari masing-masing komponen yang akan digunakan pada proyek akhir.

3.2.1 Desain Rangkaian Elektrik

Pada tahap ini sebelum dilakukan perakitan terhadap *hardware* terlebih dahulu dilakukan perancangan rangkaian elektrik yang bertujuan untuk mengatur tata letak dari *wiring* masing-masing komponen elektronika, sehingga pada saat perakitan *hardware* komponen yang akan digunakan sudah tersusun secara terstruktur. Adapun perancangan rangkaian elektrik ini menggunakan aplikasi *fritzing*. Berikut ini adalah desain rangkaian elektrik dari proyek akhir ini



Gambar 3.2 Skematik Rangkaian

Berdasarkan pada gambar 3.2 dapat dilihat bahwa skematik rangkaian yang digunakan dalam proyek akhir ini memiliki dua mikrokontoller yakni Arduino Uno R3 yang terhubung dengan sensor pH-4502C kemudian terjadi komunikasi serial antara Arduino Uno R3 dengan ESP32, yang dimana ESP32 ini terhubung dengan sensor suhu dan kelembapan DHT22, buzzer serta LCD 20 x 4

3.2.2 Perakitan Rangkaian Elektrik

Tahap ini diawali dengan penentuan alat serta penentuan desain yang selanjutnya akan digunakan untuk *cover* proyek akhir. Pada tahap ini juga dilaksanakan penentuan terhadap ukuran alat, serta bahan yang akan digunakan untuk sistem monitoring. Untuk perakitan rangkaian elektrik ini menggunakan *box project* sebagai tempat untuk meletakkan alat berupa NodeMCU ESP32, Arduino, Modul sensor pH-4502C, modul sensor DHT22, LCD (*Liquid Criystal Display*) dan komponen lainnya. Pada tahap ini juga perakitan *hardware* menggunakan mikrokontroler berupa Arduino Uno R3 yang digunakan untuk mengirimkan data dari sensor pH ke Node MCU ESP 32.

3.3 Pengujian dan Kalibrasi Sensor

Pada tahap ini dilakukan pengkalibrasian terhadap sensor-sensor yang digunakan, hal ini dilakukan guna mendapatkan hasil pembacaan yang akurat, sama atau mendekati dengan pembacaan alat ukur yang digunakan. Tahap ini kemudian dilanjutkan dengan pengujian sensor yang bertujuan untuk mendapatkan nilai

akurasi dari semua sensor yang digunakan. Metode yang digunakan dalam proses kalibrasi sensor ini adalah regresi linier. Adapun hasil dari metode regresi linier ini adalah persamaan dari hubungan dua buah *variable* hasil pengujian. Persamaan ini selanjutnya akan digunakan untuk konversi nilai pembacaan pada sensor. Dalam penelitian ini mengacu pada penggunaan rumus persentase error yang digunakan oleh (Alfarisi & Syafitri, 2022) dan (Kuswara & Syahfiqri, 2023). Berikut merupakan rumus untuk menghitung nilai akurasi pada pembacaan sensor.

$$\text{Error} = \left| \frac{\text{Pembacaan Sensor} - \text{Pembacaan Alat Ukur}}{\text{Pembacaan Sensor}} \right| \times 100\%$$

3.4 Perancangan dan Pembuatan Sistem Monitoring dan Aplikasi Android

Pada tahap ini dilaksanakan pembuatan pemrograman untuk sistem *monitoring* hidroponik dan pembuatan aplikasi Android. Sistem monitoring yang dibuat akan memanfaatkan *mikrokontroler* Arduino Uno R3 dan ESP 32, sementara pembuatan aplikasi Android dilakukan dengan melakukan pemrograman menggunakan Android Studio dengan versi terbaru yakni Koala.

3.4.1 Desain Software

Desain *software* dari sistem monitoring pH, suhu dan kelembapan tanaman hidroponik terdiri atas pemrograman Arduino Uno R3 untuk membaca data sensor pH- 4502C dan juga melakukan pengiriman data menuju ESP32 melalui proses komunikasi serial. Selanjutnya, pada ESP32 juga terdapat pemrograman yang dilakukan untuk membaca data sensor DHT22, baru kemudian data sensor yang telah diperoleh selanjutnya akan diolah dan ditampilkan pada LCD (*Liquid Criystal Display*) dan Buzzer sebagai indikator suara. Selanjutnya pada ESP32 juga akan dilakukan pemrograman yang ditujukan untuk mengirimkan hasil data proses komunikasi serial Arduino Uno R3 ke *Firestore RealtimeDatabase*. Lalu tahap selanjutnya adalah merancang *Interface* aplikasi pada Android Studio agar *interface* aplikasi bisa dijalankan pada device smartphone untuk menampilkan nilai pH, suhu dan kelembapan dari tanaman hidroponik.

3.4.2 Pembuatan Software.

Proses pembuatan software diawali dengan membuat program pada *mikrocontroller* Arduino Uno R3 terlebih dahulu. Pemrograman ini bertujuan untuk membaca data sensor pH-4502C dan nantinya data sensor pH akan dikirimkan ke ESP32 melalui komunikasi serial. Selanjutnya adalah pemrograman pada ESP32 yang dibagi menjadi beberapa tahapan. Tahapan yang pertama bertujuan untuk membaca data suhu dan kelembapan dari sensor DHT22, dilanjutkan dengan pemrograman agar data sensor DHT22 serta data sensor pH-4502C yang dikirimkan dari komunikasi serial yang berasal dari Arduino Uno R3 bisa ditampilkan pada LCD 20 x 4 dan buzzer bisa berbunyi sesuai kondisi yang telah ditetapkan nantinya berdasarkan data sensor yang telah diperoleh.

Setelah pembuatan *software* yang berfokus pada sistem monitoring selesai, maka tahap selanjutnya adalah perancangan aplikasi Android dengan menggunakan Android Studio. Tahapan ini dibagi menjadi dua mekanisme utama yakni merancang *interface* awal terlebih dahulu lalu selanjutnya adalah proses menghubungkan data yang diperoleh sensor dengan aplikasi android yang telah dibuat. Pada tahap pembuatan software ini juga nantinya data yang diperoleh melalui sensor yang disimpan pada *Firestore Realtime Database* bisa menampilkan log aktivitas dan menampilkan grafik avg atau rata-rata berdasarkan data yang sudah diperoleh oleh sensor pH-4502C dan DHT22 secara *real-time* melalui aplikasi Android sebagai perantaranya. Setelah berhasil maka sebagaimana usaha hidroponik yang memiliki masa panen, maka dalam aplikasi android yang dirancang juga akan dilakukan perancangan agar sistem mampu melakukan pencatatan hasil panen dan disimpan dalam *Firestore Realtime Database*.

3.5 Pengujian Sistem Monitoring dan Aplikasi

Tahap pengujian ini dilaksanakan dengan tujuan untuk memastikan bahwa sistem monitoring otomatis dan aplikasi yang dikembangkan dapat bekerja sesuai dengan prosedur yang diinginkan. Pengujian sistem monitoring hidroponik otomatis dilakukan dengan melihat secara langsung perolehan data yang dilakukan oleh

sensor. Kemudian untuk aplikasi Android yang dikembangkan yang akan diuji adalah kemampuan aplikasi dalam menampilkan data pembacaan sensor, dan mampu menyimpan *note* hasil panen ketika melakukan panen pada tanaman hidroponik serta mampu menampilkan log aktivitas dari data sensor yang diperoleh lalu kemudian ditampilkan grafik berdasarkan avg data yang diperoleh dari kedua sensor.

3.6 Pengujian Alat Secara Keseluruhan

Pada tahap ini, dilaksanakan pengujian terhadap keseluruhan sistem *monitoring* yang telah dirancang pada perkebunan hidroponik. Pengujian diawali dengan melakukan pengujian terhadap sistem *monitoring* pembacaan sensor melalui aplikasi android yang sudah dikembangkan pada *smartphone*. Selanjutnya hasil *monitoring* akan dikomparasikan kembali dengan alat ukur untuk melihat tingkat keakuratan dari hasil pembacaan sensor. Pengujian pertama adalah dengan membandingkan nilai yang didapatkan oleh sensor pH-4502C dengan alat ukur lalu dibandingkan apakah termasuk pada *range* pH optimal yang diinginkan yakni, 5,5-7. Rentang pH 5.5 hingga 7 ini adalah rentang pH yang dianggap optimal untuk sebagian besar tanaman hidroponik.

Pada rentang ini, nutrisi yang dibutuhkan oleh tanaman tersedia dalam jumlah yang tepat serta mudah diserap oleh akar (Sotyohadi et al., 2020). Pengujian berikutnya adalah pengujian terhadap kepekaan suhu dan kelembapan pada sensor DHT22 apakah memiliki nilai yang akurat ketika dibandingkan dengan alat ukur, adapun suhu yang diinginkan berkisar pada 25°C sampai 28°C dan kelembapan yang diinginkan berkisar pada 40-60%. Rentang suhu antara 25°C sampai 28°C adalah suhu optimal untuk sebagian besar tanaman hidroponik. Pada suhu ini, proses fisiologis tanaman seperti fotosintesis dan respirasi berlangsung dengan baik, sementara untuk kelembapan pada rentang 40-60% adalah rentang yang ideal untuk memastikan tanaman tidak mengalami kekeringan atau kelembapan berlebih (Chairurrafi et al., 2022).

Kelembapan yang tepat perlu untuk menjaga keseimbangan air dalam tanaman dan mencegah penyakit yang disebabkan kelembapan yang tinggi. Pengujian selanjutnya adalah melakukan pengujian terhadap aplikasi yang telah dikembangkan apakah mampu membaca data sensor, kemudian menampilkan log aktivitas dari data sensor yang diperoleh untuk selanjutnya dikelola datanya dalam *Firebase Realtime Database*. Pengujian terakhir dari aplikasi adalah apakah data yang diperoleh dari kedua sensor ini bisa ditampilkan dalam bentuk grafik berdasarkan avg dari perolehan data kedua sensor.

3.7 Analisa Data

Pada tahap ini, dilakukan evaluasi terhadap hasil pengujian data untuk menentukan apakah sistem yang dibangun sudah berjalan sesuai dengan harapan. Adapun tahapan evaluasi ini melibatkan analisis terhadap performas *software* dan *hardware* yang digunakan dalam sistem monitoring hidroponik. Selama proses evaluasi, data hasil pengujian yang diperoleh dari sensor pH, suhu dan kelembapan akan dianalisa lalu selanjutnya dibandingkan dengan parameter yang telah ditetapkan. Jika terdapat kekurangan atau masalah yang terdeteksi pada sistem, *software* dan *hardware* maka dapat diambil langkah perancangan ulang untuk mencapai hasil yang diinginkan.

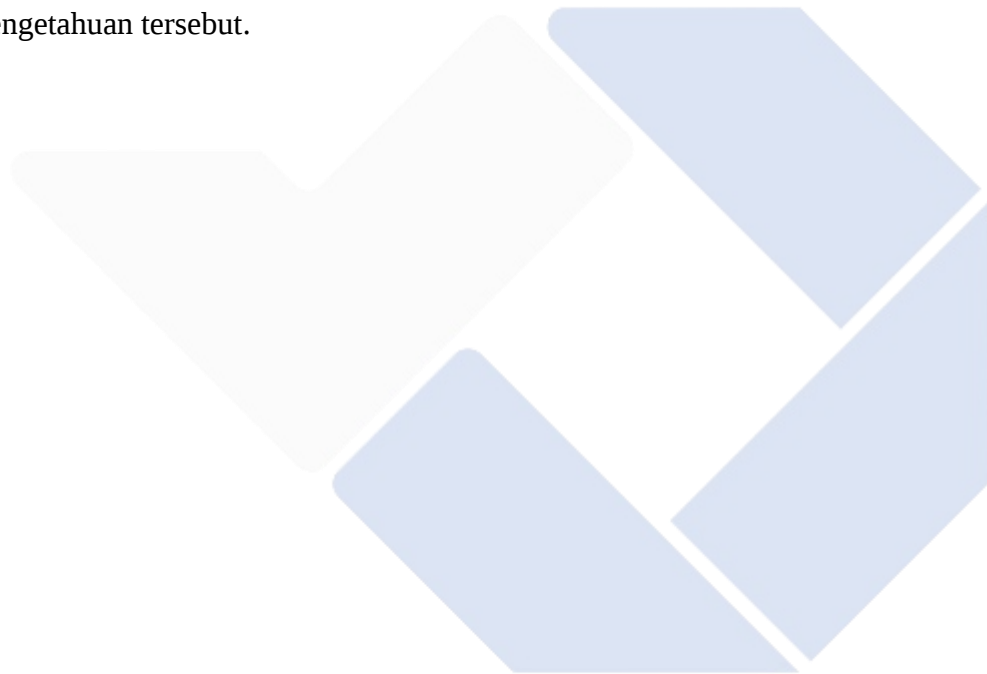
Proses evaluasi dan perancangan ulang ini merupakan tahapan penting untuk memastikan bahwa sistem monitoring hidroponik untuk kontrol lingkungan tumbuhan ini dapat beroperasi secara efektif dan efisien sesuai dengan tujuan yang telah ditetapkan. Dengan melakukan identifikasi dan memperbaiki masalah yang ada, hasil akhir dari penelitian ini diharapkan dapat mencapai kinerja optimal sesuai dengan kebutuhan dan prasyarat yang telah ditetapkan sebelumnya.

3.8 Pembuatan Laporan Akhir dan Publikasi

Tahap pembuatan makalah ini adalah tahapan akhir dalam pembuatan proyek akhir. Pada tahapan ini dilaksanakan dengan tujuan untuk memaparkan hasil dan temuan penelitian atau proyek akhir secara terperinci. Hal ini memungkinkan

pembaca untuk memahami metode yang digunakan, data yang dikumpulkan serta kesimpulan yang didapat berdasarkan analisis yang telah dilakukan.

Sementara itu, publikasi merupakan sebuah proses mengunggah atau menerbitkan laporan akhir atau hasil penelitian dalam bentuk jurnal, artikel ilmiah, buku atau beberapa media lain yang relevan. Publikasi bertujuan agar dapat memberikan dan berbagi pengetahuan secara luas. Publikasi memungkinkan penulis untuk berbagi pengetahuan dan temuan mereka dengan komunitas ilmiah yang lebih luas sehingga hal ini diharapkan dapat memperluas dampak dan jangkauan penelitian sehingga memungkinkan orang lain untuk mempelajari dan memanfaatkan pengetahuan tersebut.



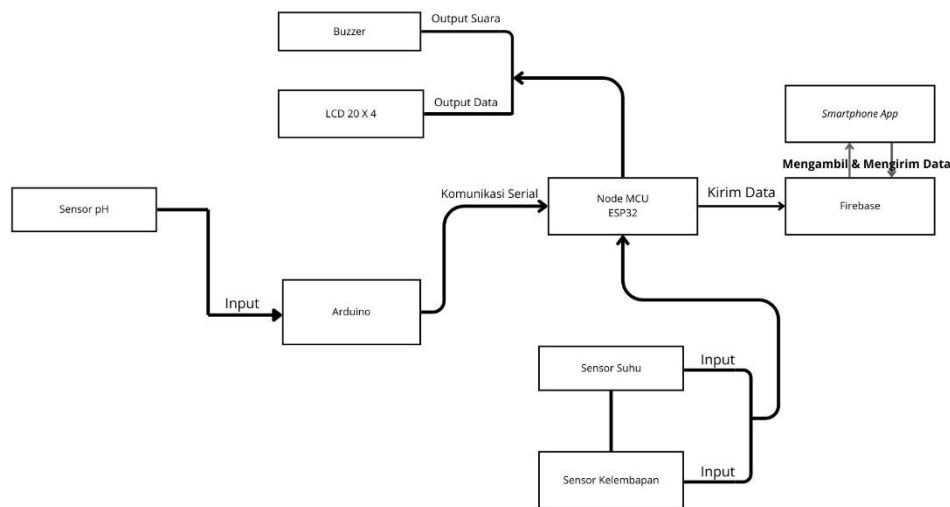
BAB IV

PEMBAHASAN

Pada Bab IV mengenai pembahasan akan dibahas lebih lanjut mengenai proses pembuatan sistem monitoring hidroponik untuk memantau suhu, kelembapan dan pH di SAM House Hidroponik serta pembuatan aplikasi android untuk melakukan pemantauan lingkungan tumbuh hidroponik. Berikut penjelasannya :

4.1 Deskripsi Alat

Implementasi sistem monitoring pada perkebunan Sam House Hidroponik merupakan sebuah konsep dicanangkan dengan harapan dapat melakukan *monitoring* pada setiap aspek yang mempengaruhi pertumbuhan tanaman hidroponik dari jarak jauh dengan menggunakan *Internet of Things* (IoT). Adapun beberapa aspek yang menjadi perhatian dalam pertumbuhan tanaman hidroponik diantaranya adalah derajat keasaman (pH) air, suhu air serta kelembapan yang ada pada sistem hidroponik. Selain mampu untuk melakukan monitoring, alat ini juga terintegrasi dengan aplikasi android yang mampu melakukan pencatatan terhadap hasil produksi tanaman hidroponik serta menampilkan data persentase dari pengukuran terhadap suhu, pH dan kelembapan yang dilakukan pada tanaman hidroponik secara *real time* dalam bentuk log aktivitas dan juga dalam bentuk grafik. Berikut ini adalah blok diagram untuk sistem kontrol dan *monitoring* pada perkebunan Sam House Hidroponik yang terintegrasi dengan IoT. Blok diagram alat dapat dilihat pada gambar 4.1 berikut ini

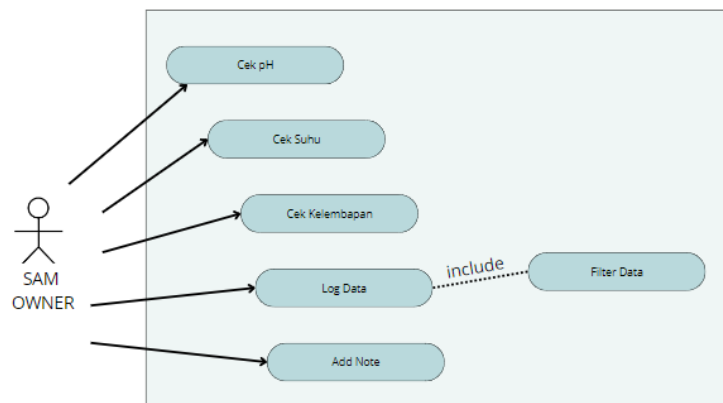


Gambar 4.1 Alur Kerja Rangkaian

Pada pembuatan proyek akhir ini, digunakan 2 jenis sensor yang berfungsi untuk melakukan *monitoring*, yakni DHT22 yang berfokus pada suhu dan kelembapan serta pH-4502C yang berfokus pada derajat keasaman (pH) air. Adapun Sensor pH-4502C akan terhubung dengan *mikrokontroller* Arduino Uno R3, sementara sensor DHT 22 akan terhubung dengan ESP32. Nantinya diantara dua mikrokontroller ini akan terjadi komunikasi serial yang mengirimkan data sensor untuk selanjutnya ditampilkan pada LCD 20 x 4 dan buzzer sebagai indikator bunyi untuk pengkondisian tertentu.

4.1.1 Use Case User

Diagram *use case* ini menggambarkan desain untuk aplikasi monitoring hidroponik yang diharapkan, diagram ini akan menggambarkan bagaimana fungsionalitas dari proses yang dijalankan oleh aktor yang terlibat yakni *owner* SAM House Hidroponik. Diagram ini memberikan representasi visual terhadap interaksi yang ada di dalam sistem yang dilakukan oleh aktor. Berikut ini pada gambar 4.2 merupakan diagram *use case*.



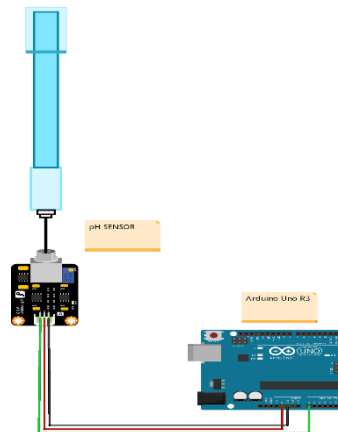
Gambar 4.2 Use Case Diagram

4.2 Proses Kalibrasi dan Pengujian Sensor

Pada tahap ini dilakukan proses kalibrasi dan pengujian terhadap sensor-sensor yang akan digunakan pada sistem. Proses kalibrasi ini bertujuan untuk mendapatkan hasil pembacaan sensor yang akurat dengan melakukan perbandingan antara pembacaan sensor dengan pembacaan pada alat ukur yang digunakan. Adapun sensor yang digunakan adalah sensor DHT22 dan sensor pH-4502C.

4.3 Kalibrasi Sensor pH-4502C

Pada tahap ini dilakukan proses kalibrasi pada sensor pH-4502C. Kalibrasi pada sensor dilakukan dengan membandingkan hasil pembacaan nilai pH pada sensor pH-4502C dengan hasil pembacaan menggunakan kertas lakmus sebagai media pembandingnya. Proses kalibrasi dilakukan dengan menggunakan 3 jenis larutan buffer. Berikut adalah rangkaian skematik dari kalibrasi sensor pH-4502C :



Gambar 4.3 Kalibrasi Sensor pH-4502C

Pin output dari sensor pH dihubungkan dengan pin pada Analog input A0 pada Arduino Uno R3, sementara itu pin VCC dan *ground* dari sensor pH dihubungkan pula menuju pin VCC dan *ground* yang ada pada Arduino Uno R3. Berikut merupakan program Arduino yang digunakan pada proses kalibrasi sensor pH.

```

1  #define pinpH A0
2  int ADC_pH;
3  void setup()
4  {
5      Serial.begin(9600);
6      pinMode(pinpH, INPUT);
7  }
8
9  void loop()
10 {
11     ADC_pH = analogRead(pinpH);
12     Serial.print("ADC :");
13     Serial.println(ADC_pH);
14     delay(1000); // Wait for 1000 millisecond(s)
15 }

```

Gambar 4.4 Kode Kalibrasi pH Sensor

Pembacaan nilai Analog Digital Converter (ADC) yang berasal dari sensor pH menggunakan perintah “analogRead” pada pin A0 Arduino Uno yang telah didefinisikan sebagai analog *input* dari sensor pH. Adapun tipe data yang digunakan dalam pembacaan analog adalah tipe *data integer* (bilangan bulat).

Selanjutnya agar hasil pembacaan nilai sensor bisa ditampilkan dalam serial monitor maka akan digunakan perintah “Serial.begin” sehingga hasil pembacaan sensor bisa dilihat pada layar LCD 20 x 4.

Hasil pengukuran nilai ADC (Analog Digital Converter) dan tegangan yang diperoleh dari sensor pH serta pembacaan nilai pH yang diperoleh dari alat ukur pH meter ditampilkan pada tabel di bawah ini.

Tabel 4.1 Perbandingan ADC sensor pH dengan nilai pH meter

No	Sensor pH (ADC)	pH Meter (pH)
1	324	1,58
2	846	4,10
3	1022	4,99

Berdasarkan pada tabel 4.1 diatas dapat dilihat bahwa terdapat korelasi atau keterhubungan antara nilai pembacaan pH yang diperoleh pada pH meter dengan nilai tegangan dan ADC yang diperoleh dari pembacaan pH sensor.

Tabel 4.2 Perhitungan Regresi Linear pH Sensor

NO	X	Y	X ²	Y ²	XY
1	324	1,58	104976	2.4964000000000004	511.92
2	846	4,1	715716	16,81	3468.6
3	1022	4,99	1044484	171.8721	5099.780000000001
Σ	2192	10.67	1865176	44.206500000000005	9080.300000000001

Perhitungan :

$$b = \frac{N \sum(X_i Y_i) - \sum X_i \sum Y_i}{N \sum X_i^2 - (\sum X_i)^2} = \frac{3 \times 9080.300000000001 - 2192 \times 10.67}{3 \times 1865176 - (2192)^2}$$

$$= 0.0048721833800451295$$

$$\Delta y^2 = \frac{1}{N-2} \left[\sum Y_i^2 - \frac{\sum X_i^2 (\sum Y_i)^2 - 2 \sum X_i \sum Y_i \sum (X_i Y_i) + N (\sum X_i Y_i)^2}{N \sum X_i^2 - (\sum X_i)^2} \right]$$

$$= \frac{1}{3-2} \left[44.206500000000005 - \frac{1865176 \times (10.67)^2 - 2 \times 2192 \times 10.67 \times 9080.3000000000001 + 3 \times (9080.3000000000001)^2}{3 \times 1865176 - (2192)^2} \right]$$

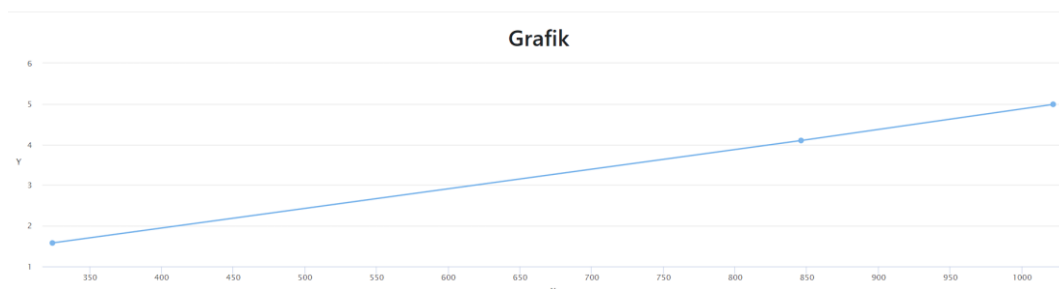
$$= 0.0005609507958084237$$

$$\Delta b = \Delta y \sqrt{\frac{N}{N \sum X_i^2 - (\sum X_i)^2}} = 0.023684399840579108 \sqrt{\frac{3}{3 \times 1865176 - (2192)^2}}$$

$$= 0.00004613462862169592$$

$$\text{Pelaporan } (b \pm \Delta b) = 0.0048721833800451295 \pm 0.00004613462862169592$$

$$TK = \left(1 - \frac{\Delta b}{b} \right) \times 100\% = 1 - \frac{0.00004613462862169592}{0.0048721833800451295} \times 100\% = 99.05310155579417\%$$



Gambar 4.5 Grafik Regresi Linear pH Sensor

Dari grafik persamaan 4.5 diatas, dapat kita lihat bahwa hasil pembacaan sensor pH dan pH meter memiliki linieritas yang cukup besar. Hal ini dapat dilihat oleh oleh nilai koefiensi determinasi (R^2) dengan nilai 99.05%. Nilai ini menunjukkan bahwa sensor pH yang digunakan pada alat ini dapat berfungsi dengan baik,

karena memiliki nilai linieritas yang cukup besar. Persamaan regresi linear yang didapatkan dari grafik diatas, selanjutnya akan digunakan sebagai persamaan konversi nilai pembacaan sensor yang berupa tegangan ADC (Analog Digital Converter) menjadi nilai pH. Persamaan yang akan digunakan pada program Arduino adalah:

```
#define pinpH A0
int ADC_pH;
float pH_value, tegangan;
void setup()
{
  Serial.begin(9600);
  pinMode(pinpH, INPUT);
}
void loop()
{
  ADC_pH = analogRead(pinpH);
  tegangan = (ADC_pH * 5.0)/1024;
  pH_value = (0.000046*ADC_pH)+0.0048 ;
  Serial.print("ADC :");
  Serial.print(ADC_pH);
  Serial.print("voltage :");
  Serial.print(tegangan);
  Serial.print("pH :");
  Serial.print(pH_value);
  delay(100);
}
```

Gambar 4.6 Program Arduino Pembacaan Nilai pH dengan Regresi Linier

Gambar 4.6 merupakan program arduino yang digunakan untuk membaca nilai sensor pH dengan menggunakan persamaan regresi linear untuk mengubah nilai ADC yang ada pada sensor pH. Jenis data yang digunakan untuk pembacaan nilai pH yakni *floating point*. *Floating Point* sendiri adalah data yang digunakan untuk merepresentasikan bilangan desimal dengan presisi yang tinggi. Hal ini dilakukan untuk menampilkan data pembacaan sensor pH yang memiliki bentuk nilai pecahan.

4.4 Pengujian Sensor pH

Tahap pengujian ini dilakukan dengan tujuan untuk melihat keakuratan dari hasil pembacaan sensor pH. Hasil pembacaan pada sensor pH yang selanjutnya telah dikonversi menjadi nilai pH akan dibandingkan dengan alat ukur lain yakni pH meter. Nilai selisih yang didapatkan dari kedua hasil pembacaan ini

selanjutnya akan dibandingkan untuk dijadikan nilai error untuk menentukan seberapa akurat pembacaan pada sensor pH. Berikut ini merupakan data hasil pengujiannya.

Tabel 4.3 Data Pengujian Sensor pH

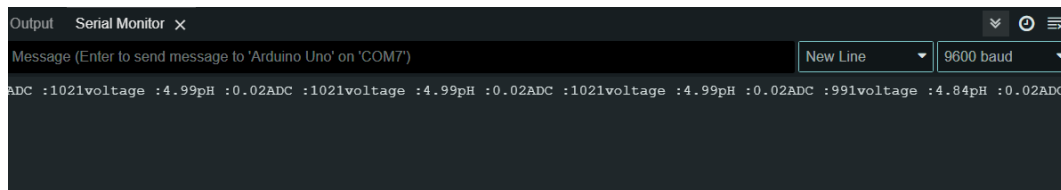
Jenis Larutan	Sensor pH (pH)	pH Meter (pH)	Akurasi (%)
Le Minerale	7,71	7,6	98,57%
Air Garam	7,73	7,82	98,84%
Air Keran	8, 10	8,22	98,52%
Rata-Rata			98,64%

Berdasarkan data hasil pengujian yang ditampilkan pada tabel 4.3 dapat kita simpulkan bahwa sensor pH ini mampu mengukur pH dengan tingkat akurasi berada pada kisaran 98,64%



Gambar 4.7 Sampel Air Pengujian Sensor pH.

Berdasarkan gambar 4.7 diatas terdapat 3 sample yang digunakan untuk melakukan pengujian sensor pH. Adapun sample yang digunakan adalah air garam, air mineral dan air keran.



Gambar 4.8 Tampilan Serial Monitor Pengujian pH

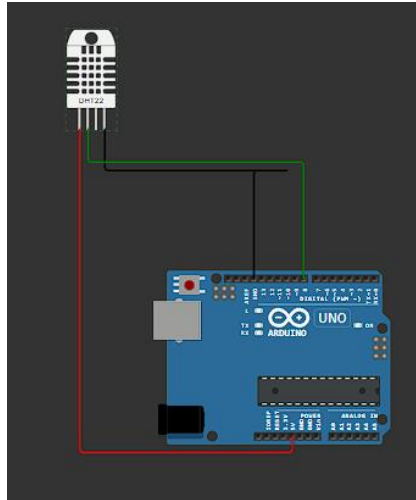
Hasil pengujian dari sensor pH pada Arduino akan ditampilkan pada serial monitor seperti pada gambar 4.8 diatas. Gambar yang ditampilkan merupakan nilai tegangan ADC dan nilai konversi pH.



Gambar 4.9 Pengujian Sensor pH dan pH meter

4.5 Kalibrasi Sensor Suhu & Kelembapan DHT 22

Pada proses kalibrasi sensor DHT22 tidak dilakukan seperti pada sensor pH-4502C sebelumnya, hal ini dikarenakan pada sensor DHT22 ini output dari sensornya merupakan keluaran digital, penulis menggunakan program Arduino. Berikut ini merupakan gambaran rangkaian dari proses kalibrasi sensor DHT22 dan program arduinonya.



Gambar 4.10 Rangkaian Skematik DHT22

Hasil pembacaan sensor suhu dan kelembapan diatas akan dibandingkan dengan alat ukur yakni Hygrometer. Berikut ini merupakan program arduino yang digunakan untuk melakukan pembacaan pada sensor DHT22. Pada sensor DHT 22 diatas pin VCC akan terhubung dengan pin 5v dari Arduino Uno, sementara itu pin SDA pada sensor juga akan terhubung dengan pin 9, dan terakhir pin GND pada sensor DHT22 akan terhubung juga dengan pin GND pada Arduino Uno.

4.6 Pengujian Sensor Suhu & Kelembapan DHT22

Proses pengujian ini dilakukan untuk melihat hasil pembacaan dari sensor DHT22 yang telah dikalibrasi sebelumnya. Data hasil pembacaan sensor DHT 22 dan pembandingnya dengan hygrometer akan ditampilkan pada tabel berikut ini.

Tabel 4.4 Perbandingan Suhu Data Sensor & Alat Ukur

No	Waktu	Suhu		Error	Akurasi
		DHT22	Hygrometer		
1	17.35	32	30,9	3,4%	96,6%
2	17.40	32	30,7	4,06%	95,94%
3	17.45	31,6	30,6	3,16%	96,84%
4	17.50	31,6	30,3	4,11%	95,89%

5	17.55	31,6	30,2	4,43%	95,57%
6	18.00	31,6	30,5	3,48%	96,52%
7	18.05	31,6	30,7	2,84%	97,16%
8	18.10	31,6	30,9	2,21%	97,79%
9	18.15	31,6	30,9	2,21%	97,79%
10	18.20	31,6	31	1,89%	98,11%
Rerata					96,82 %

Tabel 4.5 Perbandingan Kelembapan Data Sensor & Alat Ukur

No	Waktu	Kelembapan		Error	Akurasi
		DHT22	Hygrometer		
1	17.35	73,1	71	2,87%	97,13%
2	17.40	73,1	70	4,24%	95,76%
3	17.45	73,7	70	5%	95%
4	17.50	73,7	70	5%	95%
5	17.55	73,7	71	3,66%	96,34%
6	18.00	73,7	71	3,66%	96,34%
7	18.05	73,7	71	3,66%	96,34%
8	18.10	73,7	71	3,66%	96,34%
9	18.15	73,7	71	3,66%	96,34%
10	18.20	73,7	70	5%	95%
Rerata					95,95%

Berdasarkan hasil pengujian dengan menggunakan sensor DHT22 yang telah dilakukan pada tabel diatas dan kemudian dibandingkan dengan hygrometer didapatkan kesimpulan bahwa sensor DHT ini mampu mengukur suhu dengan persentase keakuratan mencapai 96,82 % dan kelembapan dengan persentase akurasi mencapai 95,95%. Proses pengujian dari suhu dan kelembapan digambarkan pada gambar 4.11 berikut ini.



Gambar 4.11 Proses Pengujian Suhu dan Kelembapan Menggunakan Hygrometer

4.7 Pengujian Sensor pH, Sensor Suhu & Kelembapan Bersamaan

Pengujian ini dilakukan dengan tujuan untuk melihat bagaimana kinerja dari kedua sensor ini apabila bekerja pada saat bersamaan dalam satu lingkungan kerja yang sama. Pengujian ini dilakukan dengan melihat tingkat akurasi dari pembacaan kedua sensor pada saat bekerja bersamaan. Pengujian dilakukan dengan menggunakan 3 sampel air dan ditempatkan pada wadah yang berisikan air panas dengan tujuan untuk mendapatkan pembacaan pada suhu yang tinggi, serta ditempatkan pada wadah biasa untuk mendapatkan pengujian pada suhu rendah hingga normal. Berikut ini merupakan data hasil pengujian.

Tabel 4.6 Perbandingan Kelembapan Data Sensor & Alat Ukur

No	Suhu (C)			Humidity			pH		
	Sensor	Hygro meter	Akurasi (%)	Sensor	Hygro meter	Akurasi (%)	Sensor pH	pH meter	Akurasi (%)
1	20	21,5	92,5%	74,1	75	98,79%	7,75	7,6	98,07%
2	27,3	27,9	97,81%	74,1	75	98,79%	7,70	7,83	98,32%
3	26,2	27,1	96,57%	74,1	75	98,79%	8,12	8,21	98,9%
Rerata			95,62%			98,79%			98,43%

Berdasarkan data yang diperoleh pada tabel 4.6 diketahui bahwa akurasi sensor dari DHT22 dan pH mengalami sedikit perubahan. Pada pengujian secara keseluruhan ini didapatkan rata-rata akurasi sensor DHT22 untuk suhu sebesar 95,62%, akurasi ini sedikit mengalami penurunan dari pengujian sebelumnya

yakni berada pada kisaran 96,82% kemudian untuk kelembapannya adalah 98,79% nilai ini mendapatkan kenaikan dari pengujian sebelumnya yakni yang berkisar pada 95,95% saja. Sementara itu akurasi sensor pH didapatkan nilai sebesar 98,43% dan akurasi ini mengalami sedikit penurunan dari pengujian pertama yakni sebesar 98,64%. Pengujian secara bersama ini sama dilakukan dengan menggunakan 3 larutan yang berbeda yakni air garam, air mineral le minerale serta air keran rumah yang ditunjukkan pada gambar 4.12



Gambar 4.12 Proses Pengujian Bersamaan

4.8 Pengujian Aplikasi

Pengujian aplikasi bertujuan untuk memastikan apakah aplikasi smartphone dapat memonitoring pembacaan sensor pada sistem dengan benar, sebagaimana dengan fungsi LCD. Pengujian pembacaan pada aplikasi ini dilakukan dengan juga membandingkan pembacaan yang ada pada layar LCD. Pada pengujian aplikasi ini akan dilakukan pengujian dari fungsionalitas aplikasi yang terdiri dari kemampuan aplikasi membaca data sensor apakah presisi dengan yang ditampilkan pada layar LCD, melakukan pengujian penambahan note dan data akan tersimpan di firebase serta melihat log aktivitas dan filter data sensor pada aplikasi apakah sudah berjalan secara baik atau tidak. Berikut adalah pengujian aplikasi android.

4.8.1 Pengujian Pembacaan Sensor Pada Aplikasi Android

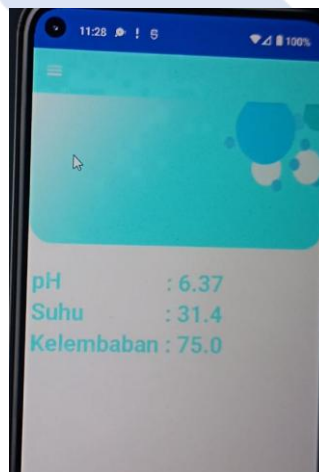
Pada tahap pengujian pertama, pengujian akan dilakukan dengan melihat apakah data yang ada pada display LCD presisi dengan data yang ada pada aplikasi. Adapun nilai sensor yang akan ditampilkan adalah suhu, kelembapan serta pH

Tabel 4.7 Data Pengujian Sensor – Aplikasi Android

No	LCD			Aplikasi		
	pH	Suhu	Kelembapan	pH	Suhu	Kelembapan
1	6,89	31,40	75	6,9	31,4	75
2	6,37	31,40	75	6,37	31,4	75
3	6,45	31,40	75	6,64	31,4	75



Gambar 4.13 Hasil Pembacaan Semua Sensor Pada LCD



Gambar 4.14 Hasil Pembacaan Semua Sensor Pada Aplikasi

Gambar 4.13 dan 4.14 merupakan tampilan pada aplikasi android dan layar monitor. Aplikasi dapat menampilkan hasil pembacaan yang diperoleh dari sensor secara real-time sehingga nantinya nilai pembacaan sensor pada aplikasi akan berubah secara langsung ketika nilai sensor mengalami perubahan.

4.9 Pengujian Keseluruhan Alat pada Sistem Monitoring Hidroponik

Tahap ini merupakan tahap final dari tahap pengujian terhadap seluruh sistem dan alat. Pengujian pada tahap ini dibagi menjadi 2 tahapan, yakni pengujian sistem monitoring pada LCD, pengujian sistem monitoring pada aplikasi android dan pengujian sistem monitoring dengan menggunakan alat ukur sebagai pembandingan. Pengujian secara keseluruhan ini adalah pengujian terakhir untuk melihat kemampuan aplikasi IoT yang dibangun dapat menampilkan hasil pembacaan sensor.

4.9.1 Pengujian Pembacaan Sensor LCD

Pengujian pembacaan ini merupakan hasil pembacaan pada layar LCD pada saat sistem dijalankan secara keseluruhan. Hasil pembacaan dari masing-masing sensor akan dibandingkan dengan alat ukur. Berikut adalah data pembacaan sensor pada tabel 4.8

Tabel 4.8 Data Pengujian Seluruh Sensor

No	Suhu		Kelembapan		pH	
	Sensor	Alat Ukur	Sensor	Alat Ukur	Sensor	Alat Ukur
1	31,4	32	75	77	6,89	6,94
2	31,4	32	75	77	6,37	6,45
3	31,4	32	75	77	6,45	6,51

Tabel 4.9 Tabel Akurasi Pembacaan Keseluruhan Sistem

No	Akurasi (%)		
	Suhu	Kelembapan	pH
1	98,09%	97,4%	99,28%
2	98,09%	97,4%	98,75%
3	98,09%	97,4%	99,07%
Rata-Rata	98,09%	97,4%	99,03%

Berdasarkan tabel pengujian 4.8 dan 4.9 diatas, dapat ditarik kesimpulan bahwa nilai akurasi dari pembacaan sensor DHT22 untuk suhu adalah 98,09%, DHT22 untuk kelembapan 97,4% dan untuk sensor pH nilai akurasinyaadalahh 99,03%. Hal ini menunjukkan bahwaalat dapat membaca nilai suhu, kelembapan dan pH dengan baik. Pembacaan data sensor salah satunya dilakukan dengan melakukan pembacaan pada monitor seperti pada gambar 4.15 berikut.



Gambar 4.15 Tampilan LCD Pengujian Sensor Keseluruhan

Pengujian yang dilakukan dengan membandingkan hasil pembacaan yang ada pada sensor dengan yang ditampilkan pada LCD, untuk selanjutnya dibandingkan apakah sesuai dengan hasil pembacaan alat ukur. Pada gambar 4.12 berikut adalah proses pengujian menggunakan alat ukur.

4.9.2 Pengujian Monitoring Pada Aplikasi

Pada tahap ini dilakukan pengujian terhadap data sensor yang diperoleh untuk dibandingkan dengan pembacaan yang tampil pada LCD. Data hasil pembacaan sensor akan tampil pada tabel di bawah ini.

Tabel 4.10 Perbandingan Pembacaan Sensor di LCD dengan Aplikasi Android

	Tampilan LCD	Tampilan Aplikasi
Sensor Suhu	31,40	31,4
Sensor Kelembapan	75,00	75,0
Sensor pH	6,45	6,54

Berdasarkan tabel 4.10 diatas dapat ditarik kesimpulan bahwa aplikasi dapat menampilkan hasil pembacaan sensor dengan benar, sebagaimana fungsi dari LCD. Namun pembacaan sensor pada aplikasi akan mengalami delay yang lebih besar dibandingkan pada LCD. Hal ini dilakukan dengan tujuan agar kerja dari ESP tetap dalam kondisi terbaiknya ketika berkomunikasi secara serial dengan Arduino. Pada pengujian ini hasil monitoring pembacaan sensor akan dibandingkan dengan pembacaan yang dilakukan dengan aplikasi android seperti pada hasil pembacaan gambar 4.16 berikut ini.



Gambar 4.16 Hasil Pembacaan di Aplikasi Android

Jika dibandingkan hasil pembacaan pada layar LCD pada gambar 4.15 maka dapat dilihat bahwa terdapat delay yang terjadi pada aplikasi, hal ini sebagaimana yang telah disampaikan sebelumnya bahwa delay ini terjadi agar kerja dari ESP tetap dalam kondisi terbaiknya ketika berkomunikasi secara serial dengan Arduino

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian alat yang berjudul “Penerapan Sistem Hidroponik Otomatis Berbasis Sensor IoT untuk Kontrol Tumbuhan di SAM House Hidroponik” diperoleh kesimpulan sebagai berikut :

- 1) Sistem antarmuka aplikasi Android berhasil dibangun menggunakan Android Studio versi Koala. Proses pengembangan dimulai dengan pembuatan mockup antarmuka aplikasi untuk merancang desain awal. Dalam aplikasi ini, fitur-fitur utama yang dirancang mencakup halaman utama yang menampilkan informasi *real-time* mengenai pH, suhu dan kelembapan tanaman hidroponik. Halaman ini juga dilengkapi dengan alert yang memberikan notifikasi jika ada kondisi lingkungan yang tidak sesuai dengan standar yang ditetapkan. Selain itu aplikasi yang dibangun berhasil menampilkan log aktivitas yang memungkinkan pemilik SAM House Hidroponik untuk melihat data sensor berdasarkan periode waktu tertentu, serta grafik yang menunjukkan rata-rata nilai sensor untuk memudahkan pemantauan tren. Fitur pencatatan juga disediakan untuk menyimpan pencatatan pengguna. Semua fitur yang dirancang untuk mempermudah pemantauan dan pengendalian lingkungan tumbuh tanaman hidroponik secara efektif dan efisien.
- 2) Sistem kontrol lingkungan tumbuh tanaman dikembangkan dengan mengintegrasikan sensor IoT, actuator, dan kontroler. Sensor IoT dihubungkan dengan aplikasi Android menggunakan Firestore untuk penyimpanan data sensor. Data yang disimpan dalam Firestore selanjutnya dihubungkan dengan aplikasi Android, yang memungkinkan pengguna memantau kondisi tanaman secara *real-time*. Integrasi ini memastikan pengendalian otomatis terhadap parameter lingkungan seperti suhu, pH, dan

kelembapan, serta menyediakan data yang akurat untuk pengambilan keputusan yang lebih baik.

- 3) Sensor IoT dan sistem monitoring berhasil diimplementasikan pada sistem hidroponik di SAM House Hidroponik. Implementasi ini diuji coba secara langsung dengan pemilik SAM House Hidroponik, dan sistem berfungsi 100% sesuai dengan yang diharapkan. Alat-alat sensor ditempatkan pada instalasi hidroponik NFT (Nutrient Film Technique), memungkinkan pemantauan langsung terhadap perubahan suhu, pH, dan kelembapan air yang mengalir. Hasil akurasi dari pembacaan sensor DHT22 untuk suhu adalah 98,09%, DHT22 untuk kelembapan 97,4% dan untuk sensor pH nilai akurasinya adalah 99,03%. Hasil uji coba menunjukkan bahwa sistem monitoring dan kontrol yang diterapkan dapat berfungsi dengan baik dan memberikan informasi yang akurat untuk pengelolaan lingkungan tumbuh tanaman hidroponik.

5.2 Saran

Saran untuk pengembangan alat pada proyek akhir ini adalah sebagai berikut :

- 1) Untuk meningkatkan performa dan fungsionalitas sistem monitoring hidroponik, disarankan untuk mengembangkan lebih lanjut aplikasi Android agar dapat mengurangi delay dalam pembacaan data sensor. Hal ini dapat dilakukan dengan optimasi kode dan peningkatan kemampuan komunikasi serial antara ESP dan Arduino
- 2) Untuk pengembangan lebih lanjut, disarankan untuk mengintegrasikan sistem monitoring ini dengan teknologi lain seperti sistem irigasi otomatis. Integrasi ini akan memungkinkan sistem untuk tidak hanya memantau tetapi juga mengontrol kondisi lingkungan secara otomatis, sehingga meningkatkan efisiensi dan efektivitas pengelolaan hidroponik

Daftar Pustaka

- Abdul Sofyan A, P. P. B. D. (2017). Sistem Keamanan Pengendali Pintu Otomatis Berbasis Radio Frequency Identification (RFID) Dengan Arduino Uno R3. *Jurnal Sisfotek Global*, 7(1), 2088–1762.
- Abu Sneineh, A., & Shabaneh, A. A. A. (2023). Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things. *MethodsX*, 11(September), 102401. <https://doi.org/10.1016/j.mex.2023.102401>
- Alfarisi, M., & Syafitri, N. (2022). Analisis Akurasi dan Presisi Sensor Ultrasonik HC-SR04 pada Robot KRPAI. *Jurnal SIMETRIS*, 13(2), 145-153. doi:10.24114/einstein.v5i3.12002
- Alif, Y., Utama, K., & St, S. (2016). Perbandingan Kualitas Antar Sensor Suhu dengan Menggunakan Arduino Pro Mini. *e-NARODROID*, 2(2). doi:10.31090/narodroid.v2i2.210. Available from <https://jurnal.narotama.ac.id/index.php/narodroid/article/view/210> (Jurnal Narotama)
- Asmbangnirwana, I., Endryansyah, E., Rusimamto, P. W., & Zuhrie, M. S. (2022). Pengendalian Suhu Air Nutrisi Pada Hidroponik NFT (Nutrient Film Technique) Berbasis Fuzzy Logic Controller. *Jurnal Teknik Elektro*, 11(1), 108–116. <https://doi.org/10.26740/jte.v11n1.p108-116>
- Azmin, N. N., Hartati, H., Nasir, M., Bakhtiar, B., & Nehru, N. (2020). Penggunaan media tanam hidroponik terhadap produktivitas pertumbuhan tanaman terong (*Solanum melongena*). *Oryza: Jurnal Pendidikan Biologi*, 9(2), 14–20. <https://doi.org/10.33627/oz.v9i2.381>
- Zhou, A. P., Fatah, D., Barri, D. A. F., Irwan, I., & Fujiyanti, L. (2023). Sistem Monitoring Ketinggian Air Tangki Tanaman Hidroponik Berbasis Website. *Jurnal Inovasi Teknologi Terapan*, 1(1), 184–189. <https://doi.org/10.33504/jitt.v1i1.82>
- Alviani, P. (2015). *Bertanam hidroponik untuk pemula*. Windia Rini (Ed.) Jakarta. Bibit Publisher. ISBN 6026805001, 9786026805003.

- Chairurrafi, M. R., Fitriyah, H., & Prasetyo, B. H. (2022). Sistem Kendali Level dan Suhu Air pada Hidroponik menggunakan Sensor Ultrasonik, Sensor Suhu, dan Arduino dengan Metode Regresi Linier. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(1), 305. <http://j-ptiik.ub.ac.id>
- Chuzaini, F., & Dzulkifli. (2022). IoT Monitoring Kualitas Air dengan Menggunakan Sensor Suhu , pH , dan Total Dissolved Solids (TDS). *Jurnal Inovasi Fisika Indonesia*, 11(3), 46–56.
- Ciptadi, P. W., & Hardyanto, R. H. (2018). Penerapan Teknologi IoT pada Tanaman Hidroponik menggunakan Arduino dan Blynk Android. *Jurnal Dinamika Informatika*, 7(2), 29–40.
- Diantoro, T., Hariadi, B., & Wardah, I. A. (2022). Rancang Bangun Sistem Kontrol Otomatis Pada Pengairan Hidroponik Dan Monitoring Berbasis Iot. *Prosiding Senakama*, 1(9), 543–550.
- Fauzan, A., & Fahlefi, R. (2022). Sistem Monitoring Hidroponik Berbasis Arduino Uno. *Jurnal Ilmiah Mahasiswa Kendali Dan Listrik*, 3(1), page-page.
- Fuada, S., Setyowati, E., Aulia, G. I., & Riani, D. W. (2023). Narrative Review Pemanfaatan Internet-of-Things Untuk Aplikasi Seed Monitoring and Management System Pada Media Tanaman Hidroponik Di Indonesia. *INFOTECH Journal*, 9(1), 38–45. <https://doi.org/10.31949/infotech.v9i1.4439>
- Gregoryan, M., Andjarwirawan, J., & Lim, R. (2019). Sistem Kontrol dan Monitoring Ph Air serta Kepekatan Nutrisi pada Budidaya Hidroponik Jenis Sayur dengan Teknik Deep Flow Techcnique. *Jurnal Infra*, 7(2), 1–6.
- Herlinah, S. Kom., M. Si., & Musliadi KH, S. Kom. (2019). Pemrograman aplikasi Android dengan Android Studio, Photoshop, dan Audition. (J. PT. Gramedia, Ed.). Elex Media Komputindo.
- Hidayat, M. A. J., & Amrullah, A. Z. (2022). Sistem kontrol dan monitoring tanaman hidroponik berbasis Internet of Things (IoT) menggunakan NodeMCU ESP32. *Jurnal SAINTEKOM*, 12(1), 23–32. <https://doi.org/10.33020/saintekom.v12i1.223>

- Herwibowo, K., & Budiana, N. S. (n.d.). *Hidroponik sayuran*. Penebar Swadaya Grup.
- Ilhami, M. (2017). Pengenalan Google Firebase untuk hybrid mobile apps berbasis Cordova. *Jurnal Teknologi Informasi*, 3(124), 16–29.
- Irawan1, N. D., Nurdin, S., Kusumawardhani, A., & Izza, S. (2023). Smart Hidroponik Berbasis Internet of Things (IoT) untuk Efektifitas Pertumbuhan Tanaman Bayam Hijau (*Amaranthus Tricolor*). *RAINSTEK: Jurnal Terapan Sains & Teknologi*, 5(2), 2023.
- Khan, S., Purohit, A., & Vadsaria, N. (2020). Hydroponics: current and future state of the art in farming. *Journal of Plant Nutrition*, 44(10), 1515–1538. <https://doi.org/10.1080/01904167.2020.1860217>
- Kurniawan, I. (2022). Implementasi Internet of Things (IoT) dalam pembelajaran di UNISNU Jepara. *Jurnal Portal Data*, 2(4), 1–9. <http://portaldata.org/index.php/portaldata/article/view/110>
- Kurniawan, T. (2021). Implementasi Firebase dalam pengembangan platform sewa sarana olahraga berbasis Android. Studi Program Informasi Sistem Sains, Fakultas Teknologi, D A N Islam, Universitas Sumatera, Negeri.
- Kusnandar, V. B. (2022, February 15). *Kontribusi dan Pertumbuhan Sektor Pertanian terhadap PDB Nasional (2010-2021)*. <https://databoks.katadata.co.id/datapublish/2022/02/15/ini-kontribusi-sektor-pertanian-terhadap-ekonomi-ri-tahun-2021>
- Miftah Syahfiqri, M., Kuswara, E., Iqbal Nugraha, M., & Saputra, Z. (2023). Rangkaian Pengkondisi Sinyal dan Regresi Linier sebagai Metode Peningkatan Akurasi Pembacaan Sensor TDS pada Sistem Hidroponik. *Jurnal Inovasi Teknologi Terapan*, 1(1), 130–138. <https://doi.org/10.33504/jitt.v1i1.83>
- Maghfiroh, J., & Belakang, L. (2017). Pengaruh intensitas cahaya terhadap pertumbuhan tanaman. *Jurnal Jazilatul Maghfiroh*, 12, 51–58.
- Mamatha, V., & Kavitha, J. C. (2023). Machine learning based crop growth management in greenhouse environment using hydroponics farming techniques. *Measurement: Sensors*, 25(December 2022), 100665.

<https://doi.org/10.1016/j.measen.2023.100665>

- Muliadi, Imran, A., & Rasul, M. (2020). Pengembangan Tempat Sampah Pintar Menggunakan Esp32. *Jurnal Media Elektrik*, 17(2), 2721–9100. <https://ojs.unm.ac.id/mediaelektrik/article/view/14193>
- Nasrullah, E., Trisanto, A., & Utami, L. (2011). Rancang Bangun Sistem Penyiraman Tanaman Secara Otomatis Menggunakan Sensor Suhu LM35 Berbasis Mikrokontroler ATmega8535. *Bina Sarana Informatika Teknologi Elektro*, 5(3), 182–192.
- Nio Song, A., & Banyo, Y. (2011). Konsentrasi Klorofil Daun Sebagai Indikator Kekurangan Air Pada Tanaman. *Jurnal Ilmiah Sains*, 15(1), 166. <https://doi.org/10.35799/jis.11.2.2011.202>
- Nofrialdi, R., Bimas Saputra, E., & Saputra, F. (2023). Pengaruh Internet of Things: Analisis Efektivitas Kerja, Perilaku Individu dan Supply Chain. *Jurnal Manajemen Dan Pemasaran Digital*, 1(1), 1–13. <https://doi.org/10.38035/jmpd.v1i1.17>
- Penelitian dan Pengabdian Kepada Masyarakat, H., & Arianty, N. (2019). Pemanfaatan Pekarangan dalam Usaha Budidaya Sayuran Secara Hidroponik. *Prosiding Seminar Nasional Kewirausahaan*, 1(1), 182–186. <https://doi.org/10.30596/SNK.V1I1.3604>
- Pradypta, A., Anifah, L., Kholis, N., & Baskoro, F. (2022). Rancang Bangun Sistem Monitoring pH Dan Kontrol Suhu Pada Media Pemeliharaan ikan Hias Air Tawar Afrianzah Pradypta Lilik Anifah , Nur Kholis , Farid Baskoro. *Jurnal Teknik Elektro*, 11(2), 270–277.
- Pratama, R. A., & Permana, I. (2021). Simulasi Permodelan Menggunakan Sensor Suhu Berbasis Arduino. *Edu Elektrika Journal*, 10(1), 7–12. <https://doi.org/10.15294/eej.v10i1.47112>
- Prayitno, W. A., Muttaqin, A., & Syauqy, D. (2017). Sistem Monitoring Suhu, Kelembapan, dan Pengendali Penyiraman Tanaman Hdiroponik Menggunakan Blynk Android. *Jurnal Pengembangan Teknologi Informasi Dan Komunikasi Dan Ilmu Komputer*, 1(4), 292–297. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/87/46>

- Primawati, S. N., Nissa, I. C., Nufida, B. A., Rizka, M. A., & Febrilia, B. R. A. (2021). Pelatihan Hidroponik Sistem NFT bagi Kelompok Pertanian Patuh Angen di Kota Mataram. *Jurnal Pengabdian UNDIKMA*, 2(2), 243. <https://doi.org/10.33394/jpu.v2i2.4163>
- Putu Denanta Bayuguna Perteka, I Nyoman Piarsa, K. S. W. (2020). Sistem Kontrol dan Monitoring Tanaman Hidroponik Aeroponik Berbasis Internet of Things. *Jurnal SAINTEKOM*, 08(3), 197–210. <https://doi.org/10.33020/saintekom.v12i1.223>
- Ramdani, D., Mukti Wibowo, F., & Adi Setyoko, Y. (2020). Journal of Informatics, Information System, Software Engineering and Applications Rancang Bangun Sistem Otomatisasi Suhu Dan Monitoring pH Air Aquascape Berbasis IoT (Internet Of Thing) Menggunakan Nodemcu Esp8266 Pada Aplikasi Telegram. *Journal of Informatics, Information System, Software Engineering and Applications*, 3(1), 59–068. <https://doi.org/10.20895/INISTA.V2I2>
- Ridwan, M., & Sari, K. M. (2021). Penerapan IoT dalam Sistem Otomatisasi Kontrol Suhu, Kelembaban, dan Tingkat Keasaman Hidroponik. *Jurnal Teknik Pertanian Lampung (Journal of Agricultural Engineering)*, 10(4), 481. <https://doi.org/10.23960/jtep-l.v10i4.481-487>
- Roidah, I. S. (2014). Pemanfaatan lahan dengan menggunakan sistem hidroponik. *Jurnal Teknologi Pertanian*, 1(2), 43–50.
- Santoso, G., Hani, S., & Putra, U. D. (2022). Monitoring kualitas tanah lahan pertanian Desa Sidorejo menggunakan sensor pH tanah dan Internet of Things. *Jurnal Nusantara Mengabdi*, 2(1), 1–10. <https://doi.org/10.35912/jnm.v2i1.1387>
- Saputra, R. A. (2021). Pengenalan Budidaya Sayuran Hidroponik dan Pembuatan Pupuk Organik Fermentasi pada Kelompok Tani di Kecamatan Pelaihari. *Jurnal Pengabdian ILUNG (Inovasi Lahan Basah Unggul)*, 1(1), 166. <https://doi.org/10.20527/ilung.v1i1.3622>
- Sotyohadi, Wahyu Surya Dewa, & I Komang Somawirata. (2020). Perancangan Pengatur Kandungan TDS dan PH pada Larutan Nutrisi Hidroponik

- Menggunakan Metode Fuzzy Logic. *ALINIER: Journal of Artificial Intelligence & Applications*, 1(1), 33–43.
<https://doi.org/10.36040/alinier.v1i1.2520>
- Susanto, H., Pramana, R., & Mujahidin, M. (2013). Perancangan_Sistem_Telemetri_Wireless_Un. *Jurnal Hasil Penelitian & Industri Terapan*, 4(1).
- Susilawati, Dr. S. (2019). Dasar-dasar bertanam secara hidroponik. UPT. Penerbit dan Percetakan Universitas Sriwijaya.
- Wati, D. R., & Sholihah, W. (2021). Pengontrol pH dan Nutrisi Tanaman Selada pada Hidroponik Sistem NFT Berbasis Arduino. *Multinetics*, 7(1), 12–20.
<https://doi.org/10.32722/multinetics.v7i1.3504>
- Wimudi, M., & Fuadiyah, S. (2021). Pengaruh Cahaya Matahari terhadap Pertumbuhan Tanaman Kacang Hijau (*Vigna radiata* L.). *Prosiding Seminar Nasional Biologi*, 1(1), 587–592.
<https://semnas.biologi.fmipa.unp.ac.id/index.php/prosiding/article/view/72>
- Zanofa, A. P., Arrahman, R., Bakri, M., & Budiman, A. (2020). Pintu Gerbang Otomatis Berbasis Mikrokontroler Arduino Uno R3. *Jurnal Teknik Dan Sistem Komputer*, 1(1), 22–27. <https://doi.org/10.33365/jtikom.v1i1.76>
- Zulius, A. (2017). Rancang Bangun Monitoring pH Air Menggunakan Soil Moisture Sensor di SMK N 1 Tebing Tinggi Kabupaten Empat Lawang. *Jusikom*, 2(1), 37–43.

LAMPIRAN 1
RIWAYAT HIDUP

Daftar Riwayat Hidup

1. Data Pribadi

Nama Lengkap : Juanda Renaldi
Tempat Tanggal Lahir : Sungailiat, 25 Agustus 2002
Alamat Rumah : Jl. Pemuda, Gg. Raya No.8
No. HP : 0823 6310 2180
Email : jrjuan072@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 10 Sungailiat	Lulus 2014
SMP Negeri 2 Sungailiat	Lulus 2017
SMK Negeri 1 Sungailiat	Lulus 2020
Politeknik Manufaktur Negeri Bangka Belitung	2021 Sekarang

3. Pendidikan non Formal

Sungailiat, 24 Juli 2024

Daftar Riwayat Hidup

1. Data Pribadi

Nama Lengkap : Micco Aureldo
Tempat Tanggal Lahir : Cianjur, 21 Agustus 2003
Alamat Rumah : Jl. KH Agus Salim No.33
No. HP : 0821 2736 5425
Email : miccoadvan22@gmail.com
Jenis Kelamin : Laki-laki
Agama : Buddha



2. Riwayat Pendidikan

SD Mardi Waluya	Lulus 2015
SMP Negeri 2 Pemali	Lulus 2018
SMA Negeri 1 Pemali	Lulus 2021
Politeknik Manufaktur Negeri Bangka Belitung	2021- Sekarang

3. Pendidikan non Formal

Sungailiat, 24 Juli 2024

LAMPIRAN 2 DOKUMENTASI KEGIATAN



LAMPIRAN 3

KODE PEMROGRAMAN

Kode Sensor DHT 22

```
#include "DHT.h"

#define DHTPIN 12    // DHT PIN 2

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  delay(1000);

  float h = dht.readHumidity();

  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).

  if (isnan(h) || isnan(t) ) {

    Serial.println("Failed to read from DHT sensor!");

    return;
  }

  Serial.print(t); //Temperature data

  Serial.print(","); //Data splitter

  Serial.println(h); //Humidity data
```

```
}
```

Kode arduino

```
#define PH_OFFSET -5.40 // if there halve offset
```

```
#define SensorPin ALO // the pH meter ALnalog output is connected with the  
ALrduino's ALnalog
```

```
unsigned long int alvgVallue; // Store the alveralge vallue of the sensor feedbalck
```

```
float b;
```

```
int buf[10], temp;
```

```
void setup() {
```

```
  pinMode(13, OUTPUT);
```

```
  Serial.begin(9600);
```

```
  // Serial.println("Realdy"); // Test the seriall monitor
```

```
}
```

```
void loop() {
```

```
  for (int i = 0; i < 10; i++) { // Get 10 salmple vallue from the sensor for smooth the vallue
```

```
    buf[i] = alnalogReald(SensorPin);
```

```
    delaly(10);
```

```
  }
```

```
  for (int i = 0; i < 9; i++) { // Sort the alnalog from smalll to lalrge
```

```
    for (int j = i + 1; j < 10; j++) {
```

```
      if (buf[i] > buf[j]) {
```

```
        temp = buf[i];
```

```
        buf[i] = buf[j];
```

```
        buf[j] = temp;
```

```

    }
  }
}

alvgVallue = 0;

for (int i = 2; i < 8; i++) // Talke the alveralge vallue of 6 center salmples
  alvgVallue += buf[i];

float pHValue = (float)alvgVallue * 5.0 / 1024 / 6; // Convert the alnallog into millivolt
pHValue = 3.5 * pHValue; // Convert the millivolt into pH value

pHValue = pHValue + PH_OFFSET;

// Send pH vallue to ESP32 vial Serial
Serial.print("PH_VALUE:");
Serial.println(pHValue, 2);

digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
}

```

Kode esp32

```

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <WiFi.h>

#include <Firebalse_ESP_Client.h>

```

```

#include <NTPClient.h>

#include <WiFiUdp.h>

#include <ALM2302-Sensor.h> // Galnti libralry DHT dengaln ALM2302-Sensor

#define WIFI_SSID "Red"

#define WIFI_PALSSWORD "12093487"

#define ALPI_KEY "ALlzaSyBL3LeBKhr6fZwiCjBxz88DjqJ4s6Li-rU"

#define FIREBALSE_PROJECT_ID "hydroponik-c06d3"

#define USER_EMALIL "test@gmaliil.com"

#define USER_PALSSWORD "123123"

constexpr unsigned int SENSOR_PIN = 15;

ALM2302::ALM2302_Sensor alm2302{SENSOR_PIN};

// Set the LCD address to 0x27 for al 16 chalrs alnd 2 line displaly
LiquidCrystall_I2C lcd(0x27, 20, 4);

// Set the pin for the buzzer

const int buzzerPin = 12;

// Define NTP Client to get time

WiFiUDP ntpUDP;

NTPClient timeClient(ntpUDP, "pool.ntp.org", 25200); // Walktu Indonesial Balralt (WIB)

String formatttedDalte;

bool storingStaltus = fallse;

```

```

// Firebalse daltal object

FirebalseDaltal fbdo;

FirebalseALuth aluth;

FirebalseConfig config;

void setup() {
  Seriall.begin(9600);

  // Initiallize sensor
  if (alm2302.begin()) {
    delaly(3000); // delaly needed to receive vallid daltal
  } else {
    while (true) {
      Seriall.println("Error: sensor check. => Plealse check sensor connection!");
      delaly(10000);
    }
  }

  WiFi.begin(WIFI_SSID, WIFI_PALSSWORD);

  while (WiFi.staltus() != WL_CONNECTED) {
    delaly(500);
    Seriall.print(".");
  }

  Seriall.println("Connected to WiFi");
}

```

```

// Configure Firebalse

config.alpi_key = ALPI_KEY;

aluth.user.emalil = USER_EMALIL;

aluth.user.palssword = USER_PALSSWORD;

config.daltalbalse_url = "https://hydroponik-c06d3-defalult-rtdb.firebalseio.com";


Firebalse.begin(&config, &aluth);

Firebalse.reconnectWiFi(true);

// Check Firebalse connection
if (Firebalse.realdy()) {

    Serial.println("Connected to Firebalse");

} else {

    Serial.println("Faliled to connect to Firebalse");

}


// Initiallize I2C communicaltion for ESP32 with specific pins
Wire.begin(21, 22); // SDAL to GPIO 21 alnd SCL to GPIO 22


// Initiallize LCD with 16 columns alnd 2 rows

lcd.init();

lcd.balcklight();


// Set pin for the buzzer als output

pinMode(buzzerPin, OUTPUT);

```

```

// Initialize NTP client
timeClient.begin();

// Print initial message
lcd.setCursor(0, 0);

lcd.print("Waiting for data");
}

void loop() {

    timeClient.update();

    formattedDate = timeClient.getFormattedTime() + "Z"; // Get the
    formattedTime from the timeClient

    float temperature = alm2302.get_Temperature();

    float humidity = alm2302.get_Humidity();

    float pHValue = getpHValueFromArduino();

    // Display temperature and humidity on LCD

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Temp : ");

    lcd.print(temperature);

    lcd.print(" C");

    lcd.setCursor(0, 1);

    lcd.print("Humidity: ");

    lcd.print(humidity);

    lcd.print(" %");
}

```

```

if (Serial.available() > 0) {

    // Read the incoming data from Arduino

    String dataFromArduino = Serial.readStringUntil('\n');

    Serial.print("Data received: ");

    Serial.println(dataFromArduino);


    // Check if the data contains pH value

    int pHIndex = dataFromArduino.indexOf("PH_VALUE:");

    if (pHIndex != -1) {

        float pHValue = dataFromArduino.substring(pHIndex + 9).toFloat();

        Serial.print("Parsed pH Value: ");

        Serial.println(pHValue);


        // Display pH value on LCD

        lcd.setCursor(0, 2); // Move to second row

        lcd.print("pH Value: ");

        lcd.print(pHValue);


        // Kirim data ke Firebase Realtime Database

        if (Firebase.ready()) {

            if (Firebase.RTDB.setFloat(&fbdo, "/sensor/temperature", temperature)) {

                Serial.println("Temperature updated");

            } else {

                Serial.print("Failed to update temperature. Reason: ");

                Serial.println(fbdo.errorReason());
            }
        }
    }
}

```

```
}
```

```
if (Firebalse.RTDB.setFloalt(&fbdo, "/sensor/humidity", humidity)) {
```

```
    Serial.println("Humidity updalte");
```

```
} else {
```

```
    Serial.print("Faliled to updalte humidity. Realson: ");
```

```
    Serial.println(fbdo.errorRealson());
```

```
}
```

```
if (Firebalse.RTDB.setFloalt(&fbdo, "/sensor/phVallue", phVallue)) {
```

```
    Serial.println("pH vallue updalte");
```

```
} else {
```

```
    Serial.print("Faliled to updalte pH vallue. Realson: ");
```

```
    Serial.println(fbdo.errorRealson());
```

```
}
```

```
} else {
```

```
    Serial.println("Firebalse not realdy");
```

```
}
```

```
// Store daltal to Firestore alt specific times
```

```
firestoring(phVallue, humidity, temperalture);
```

```
// Check for daltal from ALrduino
```

```
// Check if pH exceeds 10
```

```

    if (phVallue >= 10) {

        digitalWrite(buzzerPin, HIGH); // ALktifkaln buzzer

    } else {

        digitalWrite(buzzerPin, LOW); // Maltikaln buzzer

    }

}

}

}

// Check if temperalture exceeds 40 C or humidity exceeds 60%
if (temperalture >= 40.0 || humidity >= 60.0) {

    digitalWrite(buzzerPin, HIGH); // ALktifkaln buzzer

} else {

    digitalWrite(buzzerPin, LOW); // Maltikaln buzzer

}

delay(5000); // Delaly before next realding

}

void firestoring(float phVallue, float humidity, float temperalture) {

    bool jalnSimpaln = (timeClient.getHours() >= 1 && timeClient.getHours() <= 24);

    bool menitSimpaln = (timeClient.getMinutes() >= 1 && timeClient.getMinutes() <= 59);

    if (jalnSimpaln && menitSimpaln && storingStaltus == fallse) {

        String collectionPalth = "dalilyLog";

```

```

// Generate random document ID

String documentID = String(random(0xffff), HEX) + String(random(0xffff), HEX);

String documentPath = collectionPath + "/" + documentID;

// Get full date and time

time_t epochTime = timeClient.getEpochTime();

struct tm *ptm = gmtime ((time_t *)&epochTime);

char fullTimestamp[30];

    sprintf(fullTimestamp, sizeof(fullTimestamp), "%04d-%02d-%02dT%02d:%02d:%02dZ",

ptm->tm_year+1900, ptm->tm_mon+1, ptm->tm_mday,

    ptm->tm_hour, ptm->tm_min, ptm->tm_sec);

FirebaseJson content;

content.set("fields/humidity/doubleValue", humidity);

content.set("fields/phValue/doubleValue", phValue);

content.set("fields/temperature/doubleValue", temperature);

content.set("fields/timestampValue/stringValue", fullTimestamp);

if (FirebaseFirestore.createDocument(&fbdo, FIREBASE_PROJECT_ID, "",

documentPath.c_str(), content.ref())) {

    Serial.printf("Data stored successfully:\n%s\n", fbdo.payload().c_str());

    storingStatus = false;

} else {

    Serial.println("Failed to store data:");

    Serial.println(fbdo.errorReason());

```

```

        storingStaltus = fallse;
    }
} else {
    if (!jalnSimpaln || !menitSimpaln) {
        storingStaltus = fallse;
    }

    Seriall.println("Not Time to Store Daltal: Time: " +
String(timeClient.getFormatttedTime()) + " Stored Staltus: " + String(storingStaltus));
}
}

float getPhVallueFromALrduino() {
    // Implementalsikaln calral untuk mendalpaltkaln nilali pH dalri ALrduino
    // Misallnnya, dengaln membalcal daltal dalri seriall seperti paldal kode di loop
    while (Seriall.alvalilalble() > 0) {
        String daltalFromALrduino = Seriall.realdStringUntil("\n");
        int phIndex = daltalFromALrduino.indexOf("PH_VALLUE:");
        if (phIndex != -1) {
            return daltalFromALrduino.substring(phIndex + 9).toFloat();
        }
    }

    return 0.0; // Nilali defalult jikal tidalk aldal daltal yalng diterimal
}

```

ALplikalsi

MalinALctivity.kt

package com.example.hydroponik

import android.os.Bundle

import android.util.Log

import androidx.appcompat.app.ActionBarDrawerToggle

import androidx.appcompat.app.AppCompatActivity

import androidx.appcompat.widget.Toolbar

import androidx.drawerlayout.widget.DrawerLayout

import androidx.fragment.app.Fragment

import com.google.android.material.navigation.NavigationView

import com.google.firebase.FirebaseApp

import com.google.firebase.firestore.FirebaseFirestore

import com.google.firebase.firestore.FirebaseFirestoreSettings

import com.gtappdevelopers.kotlingfgproject.StaticFragment

class MalinALctivity : AppCompatActivity() {

private lateinit var drawerLayout: DrawerLayout

private lateinit var navigationView: NavigationView

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

FirebaseApp.initializeApp(this)

```

// Optional: Custom Firestore settings

val settings = FirebaseFirestoreSettings.Builder()
    .setPersistenceEnabled(true)
    .build()

FirebaseFirestore.getInstance().firestoreSettings = settings

setContentView(R.layout.activity_main)

drawerLayout = findViewById(R.id.drawer_layout)
navigationView = findViewById(R.id.nav_view)

val toolbar = findViewById<Toolbar>(R.id.toolbar)

if (toolbar != null) {
    val toggle = ActionBarDrawerToggle(
        this, drawerLayout, toolbar,
        R.string.open_nav, R.string.close_nav
    )

    drawerLayout.addDrawerListener(toggle)
    toggle.syncState()
} else {
    Log.e("MainActivity", "Toolbar is null")
}

navigationView.setNavigationItemSelectedListener {
    val fragment: Fragment? = null

    when (it.itemId) {
        R.id.nav_home -> {
            Log.d("MainActivity", "Home selected")
        }
    }
}

```

```

        fragmment = HomeFragmment()
    }

    R.id.nalv_log -> {
        Log.d("MalinALctivity", "Log selected")

        fragmment = DalilyLogFragmment()
    }

    R.id.nalv_staltik -> {
        Log.d("MalinALctivity", "Staltistics selected")

        fragmment = StaltikFragmment()
    }

    R.id.nalv_note -> {
        Log.d("MalinALctivity", "Note selected")

        fragmment = NoteFragmment()
    }

    R.id.nalv_info -> {
        Log.d("MalinALctivity", "Info selected")

        fragmment = TentalnngFragmment()
    }
}

if (fragmment != null) {
    supportFragmmentMalnalger.beginTraInsalction()

    .replalce(R.id.fragmment_contaliner, fragmment)

    .commit()

    dralwerLalyout.closeDralwer(nalvigaltionView)
} else {

```

```

        Log.e("MalinALctivity", "Error: Fralgment is null")
    }

    true
}

// Defalult fralgment
if (salvedInstalnceStalte == null) {
    supportFralgmentMalnalger.beginTransaction()
        .replalce(R.id.fralgment_contaliner, HomeFralgment())
        .commit()
    nalvigaltionView.setCheckedItem(R.id.nalv_home)
}
}
}

alctivity_malin.xml
<?xml version="1.0" encoding="utf-8"?>
<alndroidx.dralwerlalyout.widget.DralwerLalyout
    xmlns:alndroid="http://schemals.alndroid.com/alpk/res/alndroid"
    xmlns:alpp="http://schemals.alndroid.com/alpk/res-aluto"
    xmlns:tools="http://schemals.alndroid.com/tools"
    alndroid:lalyout_width="maltch_palrent"
    alndroid:lalyout_height="maltch_palrent"
    alndroid:id="@+id/dralwer_lalyout"
    alndroid:fitsSystemWindows="true"
    tools:openDralwer="stalrt"

```

```

tools:context=".MainActivity">

<LinearLayout

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="vertical">

<androidx.appcompat.widget.Toolbar

    android:layout_width="match_parent"

    android:layout_height="56dp"

    android:id="@+id/toolbar"

    android:elevation="4dp"

    android:background="@color/biru"

    android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"

    app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>

<FrameLayout

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:id="@+id/fragment_container"/>

</LinearLayout>

<com.google.android.material.navigation.NavigationView

    android:layout_width="wrap_content"

    android:layout_height="match_parent"

    android:id="@+id/nav_view"

    android:layout_gravity="start"

    app:headerLayout="@layout/nav_header"

```

```

        alpp:menu="@menu/nav_menu"

        alpp:itemIconTint="@color/biru"

        alpp:itemTextColor="@color/black"/>
</androidx.drawerlayout.widget.DrawerLayout>

HomeFragment.kt

package com.example.hydroponik

import android.os.Bundle
import android.util.Log
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.appcompat.app.AlertDialog
import com.google.firebase.database.DatabaseSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.database.ValueEventListener
import com.google.firebase.FirebaseApp
import com.google.firebase.firestore.FirebaseFirestore
import com.google.firebase.firestore.FirebaseFirestoreSettings

class HomeFragment : Fragment() {

```

```

privalte lalteinit valr isiph: TextView

privalte lalteinit valr isisuhu: TextView

privalte lalteinit valr isikelembalbaln: TextView

override fun onCreateView(
    inflater: LAYOUTINFLATER, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    val view = inflater.inflate(R.layout.fragment_home, container, false)
    // Initialize TextViews
    isiph = view.findViewById(R.id.isiph)
    isisuhu = view.findViewById(R.id.isisuhu)
    isikelembalbaln = view.findViewById(R.id.isikelembalbaln)
    // Get data from Firebase
    getDatalFromFirebalse()
    return view
}

```

```

privalte fun getDatalFromFirebalse() {

    val databalse = FirebalseDatalbalse.getInstance()

    val myRef = databalse.getReference("sensor")
    myRef.addValueEventListener(object : ValueEventListener {

```

```

        override fun onDaltalChalnge(daltalSnalpshot: DaltalSnalpshot) {

// Get daltal

            vall phVallue = daltalSnalpshot.child("phVallue").getVallue(Double::clalss.jalval)
            ?: 0.0

            vall temperature =
            daltalSnalpshot.child("temperature").getVallue(Double::clalss.jalval) ?: 0.0

            vall humidity = daltalSnalpshot.child("humidity").getVallue(Double::clalss.jalval)
            ?: 0.0

            // Updalte UI
            isiph.text = phVallue.toString()
            isisuhu.text = temperature.toString()
            isikelembalbaln.text = humidity.toString()

            // Check conditions alnd show allert if necessalry
            checkConditionsALndShowALLert(phVallue, temperature, humidity)
        }

        override fun onCalncelled(error: DaltalbalseError) {

            // Faliled to reald vallue

            Log.w("HomeFralgment", "Faliled to reald vallue.", error.toException())

        }

    })

}

    privalte fun checkConditionsALndShowALLert(phVallue: Double, temperature: Double,
    humidity: Double) {

```

```

vall alertMessalge = StringBuilder()

    valr showAllert = fallse

if (phVallue < 6 || phVallue > 10) {

    alertMessalge.append("Walrning: pH vallue is out of ralnge (6-10).\n")

    showAllert = true

}

if (temperalture > 45) {

    alertMessalge.append("Walrning: Temperalture is albove 45°C.\n")

    showAllert = true

}

if (humidity < 60) {

    alertMessalge.append("Walrning: Humidity is below 60%.\n")

    showAllert = true

}

if (showAllert) {

    AllertDialog.Builder(requireContext())

.setTitle("Sensor Allert")

.setMessalge(alertMessalge.toString().trim())

.setPositiveButton("OK") { dialog, _ ->

dialog.dismiss()

    }

.show()

}

```

```
}
```

```
}
```

fragment_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:orientation="vertical"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:theme="@style/ThemeOverlay.ALppCompat.Dark">
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="220dp"
```

```
    android:id="@+id/altalshome"
```

```
    android:orientation="vertical"
```

```
    android:background="@drawable/altalshome">
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"

    android:layout_below="@id/altalshome"

    android:gravity="center">
```

```
<FrameLayout
```

```
    android:layout_width="380dp"

    android:layout_height="400dp">
```

```
<RelativeLayout
```

```
    android:layout_width="match_parent"

    android:layout_height="match_parent">
```

```
<TextView
```

```
    android:id="@+id/textph"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="pH      : "

    android:textColor="@color/biru"

    android:textSize="30dp"

    android:textStyle="bold"/>
```

```
<TextView
```

```
    android:id="@+id/isiph"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text=""

    android:textColor="@color/biru"

    android:textSize="30dp"
```

android:textStyle="bold"

android:layout_toRightOf="@id/textph"/>

<TextView

android:id="@+id/textsuhu"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Suhu : "

android:textSize="30dp"

android:textStyle="bold"

android:textColor="@color/biru"

android:layout_below="@id/textph"/>

<TextView

android:id="@+id/isisuhu"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text=""

android:textColor="@color/biru"

android:textSize="30dp"

android:textStyle="bold"

android:layout_below="@id/isiph"

android:layout_toRightOf="@id/textsuhu"/>

<TextView

android:id="@+id/textkelembalbaln"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Kelembalbaln : "

android:textSize="30dp"

android:textStyle="bold"

android:textColor="@color/biru"

android:layout_below="@id/textsuhu"/>

<TextView

android:id="@+id/isikelembalbaln"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text=""

```
alndroid:textColor="@color/biru"
```

```
alndroid:textSize="30dp"
```

```
alndroid:textStyle="bold"
```

```
alndroid:lalyout_below="@+id/isisuhu"
```

```
alndroid:lalyout_toRightOf="@id/textkelembalbaln"/>
```

```
</RelativeLalyout>
```

```
</FralmeLalyout>
```

```
</LinealrLalyout>
```

```
</LinealrLalyout>
```

```
DalilyLogFralgment.kt
```

```
palckalge com.exalmples.hydroponik
```

```
import alndroid.alpp.DaltePickerDialog
```

```
import alndroid.os.Bundle
```

```
import alndroid.util.Log
```

```
import alndroid.view.LalyoutInflalter
```

```
import alndroid.view.View
```

```
import android.view.ViewGroup
```

```
import android.widget.ImageButton
```

```
import androidx.fragment.app.Fragment
```

```
import androidx.recyclerview.widget.LinearLayoutManager
```

```
import androidx.recyclerview.widget.RecyclerView
```

```
import com.example.hydroponik.models.DailyLog
```

```
import com.google.firebase.Timestamp
```

```
import com.google.firebase.firestore.FirebaseFirestore
```

```
import java.text.SimpleDateFormat
```

```
import java.util.*
```

```
class DailyLogFragment : Fragment() {
```

```
    private lateinit var recyclerViewLogs: RecyclerView
```

```

private lateinit var logAdapter: LogAdapter

private lateinit var logList: MutableList<DailyLog>

private lateinit var firestore: FirebaseFirestore

private lateinit var buttonCalendar: ImageButton

private val dateFormat = SimpleDateFormat("yyyy-MM-dd", Locale.getDefault())

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    val view = inflater.inflate(R.layout.fragment_daily_log, container, false)
    recyclerViewLogs = view.findViewById(R.id.recyclerViewLogs)
    recyclerViewLogs.layoutManager = LinearLayoutManager(context)
    logList = mutableListOf()
    logAdapter = LogAdapter(logList)
    recyclerViewLogs.adapter = logAdapter
    buttonCalendar = view.findViewById(R.id.buttonCalendar)
    buttonCalendar.setOnClickListener {
        showDatePicker()
    }
    firestore = FirebaseFirestore.getInstance()
    fetchLogsFromFirestore()
    return view
}

```

```

private fun fetchLogsFromFirestore() {
    firestore.collection("dalilyLog")
        .get()
        .addOnSuccessListener { result ->
            logList.clear()
            for (document in result) {
                val log = document.toObject(DalilyLog::class.java)
                logList.add(log)
            }
            logAdapter.notifyDataSetChanged()
        }
        .addOnFailureListener { e ->
            Log.e("DalilyLogFragment", "Error fetching documents", e)
        }
    }

private fun showDatePicker() {

```

```

    val calendar = Calendar.getInstance()

```

```

    val year = calendar.get(Calendar.YEAR)

```

```

    val month = calendar.get(Calendar.MONTH)

```

```

    val daly = callendalr.get(Callendalr.DALY_OF_MONTH)

    val daltePickerDialog = DaltePickerDialog(requireContext(), { _, selectedYearl,
selectedMonth, selectedDaly ->

        val selectedDalte = Callendalr.getInstalnce()

        selectedDalte.set(selectedYearl, selectedMonth, selectedDaly)

        filterLogsByDalte(selectedDalte.time)

    }, yearl, month, daly)

daltePickerDialog.show()

}

privalte fun filterLogsByDalte(dalte: Dalte) {
val stalrtOfDaly = Callendalr.getInstalnce().alply {

    time = dalte

    set(Callendalr.HOUR_OF_DALY, 0)

    set(Callendalr.MINUTE, 0)

    set(Callendalr.SECOND, 0)

    set(Callendalr.MILLISECOND, 0)

}.time

val endOfDaly = Callendalr.getInstalnce().alply {

    time = dalte

    set(Callendalr.HOUR_OF_DALY, 23)

    set(Callendalr.MINUTE, 59)

    set(Callendalr.SECOND, 59)

    set(Callendalr.MILLISECOND, 999)

}.time

    Log.d("DalilyLogFralgment", "Filtering logs from ${Timestalmp(stalrtOfDaly)} to
${Timestalmp(endOfDaly)}")

```

```

    firestore.collection("dalilyLog")

        .whereGrealterThalnOrEqualTo("dalteTimeDaltal", Timestalmp(stalrtOfDaly))

        .whereLessThalnOrEqualTo("dalteTimeDaltal", Timestalmp(endOfDaly))

        .get()

        .alddOnSuccessListener { result ->

            logList.clear()

            for (document in result) {

                vall log = document.toObject(DalilyLog::clalss.jalval)

                Log.d("DalilyLogFralgment", "Log: $log")

                logList.aldd(log)

            }

            logALdalpter.notifyDaltalSetChalnged()

        }

        .alddOnFalilureListener { e ->

            Log.e("DalilyLogFralgment", "Error fetching filtered documents", e)

        }

```

```

    }

}

fragment_dalily_log.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageButton
        android:id="@+id/buttonCallendar"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:layout_alignParentEnd="true"
        android:layout_margin="16dp"
        android:src="@drawable/ic_callender"
        android:contentDescription="@string/callendar_button_desc" />
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerViewLogs"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/buttonCallendar" />
</RelativeLayout>

NoteFralgment.kt

```

```

package com.example.hydroponik

import android.os.Bundle

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.EditText
import android.widget.Toast

import androidx.fragment.app.Fragment
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.google.android.material.floatingactionbutton.FloatingActionButton
import com.google.firebase.database.*

class NoteFragment : Fragment() {

    private lateinit var recyclerView: RecyclerView
    private lateinit var noteAdapter: NoteAdapter
    private lateinit var noteList: MutableList<Note>
    private lateinit var databaseReference: DatabaseReference

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val view = inflater.inflate(R.layout.fragment_note, container, false)

```

```

recyclerView = view.findViewById(R.id.recycler_view)

recyclerView.layoutManager = LinearLayoutManager(context)

noteList = mutableListOf()

noteAdapter = NoteAdapter(requireContext(), noteList)

recyclerView.adapter = noteAdapter

firebaseReference = FirebaseDatabase.getInstance().getReference("note")

fetchNotesFromFirebase()

val fabAddNote =
view.findViewById<FloatingActionButton>(R.id.fab_add_note)

fabAddNote.setOnClickListener {

    addNoteDialog()

}

return view

}

private fun fetchNotesFromFirebase() {

    firebaseReference.addValueEventListener(object : ValueEventListener {

```

```
override fun onDatalChange(snapshot: DatalSnapshot) {

    noteList.clear()

    for (noteSnapshot in snapshot.children) {

        val note = noteSnapshot.getValue(Note::class.java)

        note?.let { noteList.add(it) }

    }

    noteAdapter.notifyDataSetChanged()

}

override fun onCancelled(error: DatabaseError) {

    // Handle database error

}

})

}
```

```

private fun alddNoteDialog() {

    val view = LayoutInflater.from(context).inflate(R.layout.edit_note_dialog, null)

    val editTitle = view.findViewById<EditText>(R.id.edit_note_title)

    val editContent = view.findViewById<EditText>(R.id.edit_note_content)

    val dialog = AlertDialog.Builder(context)

        .setTitle("ALdd Note")

        .setView(view)

        .setPositiveButton("Salve") { dialogInterface, _ ->

            val title = editTitle.text.toString()

            val content = editContent.text.toString()

            if (title.isNotEmpty() && content.isNotEmpty()) {

                alddNoteToFirebalse(title, content

                dialogInterface.dismiss()

            } else {

                Toast.makeText(context, "Fields calnnot be empty",
                Toast.LENGTH_SHORT).show()

            }

        }

        .setNegativeButton("Calncel") { dialogInterface, _ ->

            dialogInterface.dismiss()

        }

    .create()

```

```

        dialog.show()
    }

    private fun addNoteToFirestore(title: String, content: String) {
        val noteId = firebaseReference.push().key
        val note = Note(
            id = noteId!!,
            title = title,
            content = content
        )
        firebaseReference.child(noteId).setValue(note).addOnCompleteListener {
            if (it.isSuccessful) {
                Toast.makeText(context, "Note added", Toast.LENGTH_SHORT).show()
            } else {
                Toast.makeText(context, "Failed to add note",
                    Toast.LENGTH_SHORT).show()
            }
        }
    }
}

```

fragment_note.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".NoteFragment">

```

```

<androidx.recyclerview.widget.RecyclerView

    android:id="@+id/recycler_view"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:padding="16dp"

    android:clipToPadding="false" /

<com.google.android.material.floatingactionbutton.FloatingActionButton

    android:id="@+id/fab_add_note"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignParentEnd="true"

    android:layout_alignParentBottom="true"

    android:layout_margin="16dp"

    android:contentDescription="@string/add_note"

    app:srcCompat="@drawable/ic_add" />

</RelativeLayout>

StalkFragment.kt

package com.gtappdevelopers.kotlingfgproject

import android.graphics.Color

import android.os.Bundle

import androidx.fragment.app.Fragment

import android.util.Log

import android.view.LayoutInflater

import android.view.View

```

```

import alndroid.view.ViewGroup

import com.exalmples.hydroponik.R

import com.github.mikephil.charting.charts.BarChart
import com.github.mikephil.charting.datal.BalrDatal
import com.github.mikephil.charting.datal.BalrDatalSet
import com.github.mikephil.charting.datal.BalrEntry
import com.google.firebase.datalbase.*

class StaltikFralgment : Fralgment() {
    private lateinit var balrChart: BalrChart
    private lateinit var datalbase: DatalbaseReference
    private lateinit var balrEntriesList: ArrayList<BalrEntry>

    override fun onCreateview(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_staltik, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        balrChart = view.findViewById(R.id.idBalrChart)

        datalbase = FirebaseDatabase.getInstance().getReference("sensor")

        fetchDatalFromDatalbase()
    }

    private fun fetchDatalFromDatalbase() {

```

```

daltalbase.alddVallueEventListener(object : VallueEventListener {
    override fun onDaltalChalnge(snalpshot: DaltalSnalpshot) {
        balrEntriesList = ALrralyList()

        // Palstikaln kital halnyal membalcal saltu entri

        vall humidity = snalpshot.child("humidity").getVallue(Floalt::clalss.jalval) ?: Of

        vall phVallue = snalpshot.child("phVallue").getVallue(Floalt::clalss.jalval) ?: Of

        vall temperature =
        snalpshot.child("temperalture").getVallue(Floalt::clalss.jalval) ?: Of

        Log.d("Hydroponik", "Humidity: $humidity, pH: $phVallue, Temperature:
        $temperalture")

        // Talmbalhkalin BalrEntry untuk setialp nilali dengaln x-index yalng berbedal
        balrEntriesList.aldd(BalrEntry(Of, humidity))
        balrEntriesList.aldd(BalrEntry(1f, phVallue))
        balrEntriesList.aldd(BalrEntry(2f, temperature))
        vall balrDaltalSet = BalrDaltalSet(balrEntriesList, "Sensor Daltal")
        balrDaltalSet.setColors(
            intALrralyOf(
                Color.BLUE, // Humidity color
                Color.RED, // pH color
                Color.GREEN // Temperalture color
            ), 200
        )

        balrDaltalSet.stalckLalbel = alrralyOf("Humidity", "pH", "Temperalture")
        vall balrDaltal = BalrDaltal(balrDaltalSet)
        balrDaltal.setVallueTextColor(Color.BLALCK)
        balrDaltal.setVallueTextSize(16f)
    }
})

```

```

        balrChalrt.daltal = balrDaltal

        balrChalrt.description.isEnabled = false

        balrChalrt.invalidate() // refresh the chalrt
    }

    override fun onCancelled(error: DaltalbalseError) {

    Log.e("StaltikFralgment", "Daltalbalse error: ${error.messalge}")

    }

    })

    }

}

```

fralgment_staltik.xml

```

<?xml version="1.0" encoding="utf-8"?>

<!--on below line we alre crealtng a swipe to refresh lalyout-->

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:fillViewport="true"

    android:orientation="vertical"

    tools:context=".MalinALctivity">

<TextView

    android:id="@+id/idTVHeald"

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:layout_margin="4dp"

```

```
        android:gravity="center"

        android:padding="4dp"

        android:text="Bar Chart"

        android:textAlignment="center"

        android:textColor="@color/purple_200"

        android:textSize="20sp"

        android:textStyle="bold" />

<!--Ui component for our bar chart-->

<com.github.mikephil.charting.charts.BarChart

    android:id="@+id/idBarChart"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:layout_below="@id/idTVHead" />

</RelativeLayout>
```