

**E-MONITORING PENDETEKSI CUACA DAN KELEMBABAN
TANAH UNTUK MENINGKATKAN HASIL PANEN PETANI
SAWI BERBASIS INTERNET OF THINGS (IOT)**

SKRIPSI

DISUSUN OLEH

ANGGIE SYAHZILLAH

2109020038



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025**

**E-MONITORING PENDETEKSI CUACA DAN
KELEMBABAN TANAH UNTUK MENINGKATKAN HASIL
PANEN PETANI SAWI BERBASIS INTERNET OF THINGS
(IOT)**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada
Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas
Muhammadiyah Sumatera Utara**

**Anggie Syahzillah
NPM. 2109020038**

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025**

LEMBAR PENGESAHAN

Judul Skripsi : **E-Monitoring Alat Pendeteksi Cuaca dan Kelembaban Tanah Untuk Meningkatkan Hasil Panen Petani Sawi Berbasis Internet Of Things (IoT).**

Nama Mahasiswa : Angie Syahzillah

NPM : 2109020038

Program Studi : Teknologi Informasi

Menyetujui
Komisi Pembimbing

(Dr, Firahmi Rizky, S.Kom., M.Kom.)
NIDN. 0116079201

Ketua Program Studi

Dekan

(Fatma Sari Hutagalung, S.Kom., M.Kom)
NIDN. 0117019301

(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

E-MONITORING PENDETEKSI CUACA DAN KELEMBABAN TANAH UNTUK MENINGKATKAN HASIL PANEN PETANI SAWI BERBASIS INTERNET OF THINGS (IOT)

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 18 April 2025

Yang membuat pernyataan



Anggie Syahzillah

NPM. 2109020038

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Angie Syahzillah
NPM : 2109020038
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**E-MONITORING PENDETEKSI CUACA DAN KELEMBABAN TANAH
UNTUK MENINGKATKAN HASIL PANEN PETANI SAWI BERBASIS
INTERNET OF THINGS (IOT).**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 18 April 2025

Yang membuat pernyataan

Angie Syahzillah

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Angie Syahzillah
Tempat dan Tanggal Lahir : Medan, 08 Mei 2002
Alamat Rumah : Jl. Tertib No. 99, Komplek Panggon Indah,
Lingkungan 11, Rengas Pulau, Medan Marelan
Telepon/Faks/HP : 089604350561
E-mail : Asyahzillah@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD	: SD Negeri 1 Kutapanjang	TAMAT: 2014
SMP	: SMP Negeri 20 Medan	TAMAT: 2017
SMA	: SMA Negeri 16 Medan	TAMAT: 2020

KATA PENGANTAR



Pendahuluan

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, S.Kom., M..Kom. Ketua Program Studi Teknologi Informasi
4. Bapak Mhd. Basri, S.Si., M.Kom. Sekretaris Program Studi Teknologi Informasi
5. Ibu Dr. Firahmi Rizky, M.Kom. selaku pembimbing skripsi yang telah memberikan arahan, saran, dan bimbingan yang penuh kesabaran dan ketulusan.
6. Untuk cinta pertama saya orang paling special dihidup saya walaupun dia tidak bisa melihat perjalanan anak perempuannya yang sedang berjuang agar bisa membahagiakan nya di sana yaitu Ayahanda saya yang sudah tenang di surga Alm. Syahrullah, mungkin saat ini beliau tidak bias melihat saya secara langsung tetapi beliau melihat perjuangan saya secara tidak langsung. Sekarang anak perempuan nya sudah mendapatkan nama tambahan di belakang nama yang sudah kau ciptkan pak sekarang nama anak mu Anggie Syahzillah, S.Kom
7. Ayahanda Akp. Masweliadi yang memberikan dukungan dari segi moral maupun material serta menjadi motivasi penulis untuk menyelesaikan skripsi ini. Terimakasih atas doa-doa yang telah kalian berikan sehingga anakmu selesai dengan titel S.Kom nya.
8. Ibunda tercinta saya, Ibu YuliAnita seorang ibu yang tidak pernah berhenti mendoakan anak perempuan nya sampai ke titik ini, walaupun banyak ke gaduhan antara kita. Terimakasih banyak sudah mendukung anak mu sampai menjadi

sarjana, terimakasih perjuangan mama untuk adek, terimakasih doa doa yang udah mamam berikan untuk adek sampai adek kuat. Gelar baru adek ini adek berikan untuk mama.

9. Abangda yang tersayang Muhammad Ridho Syahrozy, yang telah memberi dukungan dalam segi mentalitas, dorongan dan semangat setiap hari tanpa bosan, walaupun sering marah karena lama kali siap skripsinya, tapi adek buktikan sekarang adek udah selesai. Terimakasih abang yang gak pernah bosen setiap nelfon nanyak gimana dan udah sampai mana skripsi yang udah adek kerjakan, mungkin abang adalah cinta kedua adek pengganti bapak.
10. Teman-teman kesayangan saya Yustika Dasriyanti, Allysha Fadillah Nasution, Tiara Adella, Uswatun Hasanah, dan Tiara Dwi Tasya, Andini vitalaya, dan Natasya Aulia Angkat yang selalu memberikan saya semangat, yang selalu memberikan saya kekuatan untuk bisa bertahan sampai akhir ini. Dan terimakasih telah mendengarkan keluhan dan tangisan saya saat mengerjakan skripsi ini. Kalian selalu membuat saya bangkit di saat saya ingin menyerah.
11. Teman-teman kesayangan ibu kos, Gumay, Vildo, Igun, Azmi, Ardinal yang sudah membantu menghibur saya di saat sedih, yang sudah selalu ada di saat saya butuh, yang selalu memberikan saya semangat, yang selalu membawa saya keluar malam biar tidak stress.
12. Eci, Khairanum, dan Azra, 3 wanita yang selalu ngasih semangat dan dukungan aku walaupun melalui telfon, walaupun kami dipisahkan oleh jarak Eci di Bangka Belitung, Khairanum di Jakarta, dan Azra di Lhoksumawe tapi mereka selalu sigap disaat aku butuh tempat untuk ngeluh, selalu denegrin tangisan ku melalui telfon, terimakasih banyak udah menjadi teman baik aku.
13. Teman seperjuangan saya Hafiz Azman Matondang, yang sudah membantu saya dari awal pengajuan judul, hingga akhir skripsi, tanpa dia mungkin saya tidak bias berani untuk maju seperti sekarang ini.
14. Terimakasih buat Mila dan Lala udah ngasih semangat buat aku, udah nemenin aku disaat pembuatan skripsi ini, selau ada disaat aku sedih dan senang. Uдах selalu menjadi teman cerita disaat saya kesulitan.

15. Dan yang terakhir, terimakasih buat diri sendiri yang sudah bertahan sampai di akhir perjalanan ini, banyak rintangan yang sudah kamu lewatin dan kata kata ingin menyerah yang dulu pernah kamu ucap dari mulutmu, banyak tantangan yang harus kamu selesai kan dan pada akhirnya semua itu selesai. Terimakasih sudah bertahan sampai saat ini tidak sia sia air mata kamu yang selalu keluarn tidak sia sia jam tidur mu hancur. Pada akhirnya hal yang kamu tunggu selama kurang lebih 4 tahun ini selesai. Dan ini merupakan pencapaian terbaik mu gadis kecil Anggie Syahzillah.
16. Seluruh Bapak/Ibu dosen di Jurusan Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammdiya Sumatera Utara atas ilmu dan didikannya selama perkuliahan.
17. Tidak lupa mau ngucapin terimakasih kepada seseorang yang udah membantu saya dalam mengerjakan skripsi mengajarin saya bagaimana membuat UML, membantu saya menyelesaikan BAB 3 saya, mendengarkan amarah, tangisan, ambekan saya, selalu sabar saat mengajari saya, menasehati saya disaat saya udah buntu gak tau mau kearah mana, selalu sigap dimintaiin tolong, jam tidur dia juga jadi ikut rusak gara-gara ngebantuin aku mengerjakan skripsi, terimakasih banyak udah sangat baik kepada aku, terimakasih banyak atas waktu kamu yang udah banyak bantuin aku. Kepada seseorang yang bertitel S.Kom dengan memiliki nomor induk mahasiswa 2012000028, nama mu sudah aku sebut di salah satu ucapan terimakasih ku namun bagi ku kamu itu wajib dapat ucapan terimakasih yang terpisah. Aku sengaja buat ucapan terimakasih untuk kamu paling terakhir karena aku gak mau semua orang tau gimana kebaikan mu.

E-MONITORING PENDETEKSI CUACA DAN KELEMBABAN TANAH UNTUK MENINGKATKAN HASIL PANEN PETANI SAWI BERBASIS INTERNET OF THINGS (IOT)

ABSTRAK

Pertanian sawi merupakan salah satu komoditas hortikultura yang sangat bergantung pada kondisi lingkungan, khususnya cuaca dan kelembaban tanah. Ketidakteraturan iklim dan kesalahan dalam manajemen irigasi seringkali menjadi penyebab menurunnya produktivitas hasil panen. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem e-monitoring berbasis Internet of Things (IoT) yang mampu mendeteksi kondisi cuaca serta kelembaban tanah secara real-time. Sistem ini menggunakan sensor suhu, kelembaban udara, dan kelembaban tanah yang terintegrasi dengan mikrokontroler dan modul Wi-Fi untuk mengirimkan data ke platform monitoring berbasis web atau aplikasi. Hasil pengujian menunjukkan bahwa sistem mampu memberikan informasi yang akurat dan responsif terhadap perubahan lingkungan, sehingga petani dapat mengambil keputusan yang tepat terkait waktu penyiraman dan perlakuan terhadap tanaman. Dengan adanya sistem ini, diharapkan produktivitas pertanian sawi dapat meningkat secara signifikan melalui pengelolaan lahan yang lebih efisien dan berbasis data.

Kata Kunci: IoT, E-Monitoring, Kelembaban Tanah, Cuaca, Sawi,

E-MONITORING WEATHER DETECTION AND SOIL MOISTURE TO IMPROVE THE HARVEST OF MUSTARD FARMERS BASED ON THE INTERNET OF THINGS (IOT)

ABSTRACT

Mustard greens farming is one of the horticultural commodities that is highly dependent on environmental conditions, especially weather and soil moisture. Climate irregularities and errors in irrigation management are often the cause of decreased crop productivity. This study aims to design and develop an Internet of Things (IoT)-based e-monitoring system that is able to detect weather conditions and soil moisture in real-time. This system uses temperature, air humidity, and soil moisture sensors integrated with a microcontroller and Wi-Fi module to send data to a web-based monitoring platform or application. The test results show that the system is able to provide accurate and responsive information to environmental changes, so that farmers can make the right decisions regarding the time of watering and treatment of plants. With this system, it is hoped that the productivity of mustard greens farming can increase significantly through more efficient and data-based land management.

Keywords: IoT, E-Monitoring, Soil Moisture, Weather, Mustard Greens,

DAFTAR ISI

HALAMAN SAMPUL	I
LEMBAR PENGESAHAN	II
PENYATAAN ORISINALITAS.....	III
PERNYATAAN PERSETUJUAN PUBLIKASI.....	IV
RIWAYAT HIDUP	V
KATA PENGANTAR.....	VI
ABSTRAK	IX
ABSTRACT	X
DAFTAR ISI.....	XI
DAFTAR TABEL.....	XIV
DAFTAR GAMBAR	XV
BAB I. PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah	5
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian.....	6
BAB II. GAMBARAN UMUM PERUSAHAAN.....	8
2.1. E-Monitoring	8
2.2. Sistem	9
2.3. Website	9
2.4. Basis Data (DataBase)	11

2.5. Server.....	12
2.6. MySQL (My Structure Query Language).....	12
2.7. UML (Unified Modelling Language)	13
2.8. Soil Moisture	19
2.9. Sensor DHT22	21
2.10. Node MCU ESP 320	22
2.11. Index Kuliats Cuaca dan Kelembaban Tanah.....	23
2.12. Internet Of Things (IoT)	27
BAB III. PELAKSANAAN KERJA PRAKTIK.....	28
3.1. Metode Penelitian	28
3.2. Alur Penelitian	28
3.3. Observasi Penelitian	31
3.4. Tempat Penelitian	32
3.5. Jadwal Penelitian	32
3.6. Jenis Penelitian	33
3.7. Studi Literatur.....	33
3.8. Alat dan Bahan	33
3.9. Rancangan Alat E- Moitoring Sawi.....	34
3.10. Flowchart Sistem dan Flowchart Alat.....	36
3.11. Desain Sistem.....	39
3.12. Desain Database	42
3.13. Rancangan Website	42
BAB IV. HASIL DAN PEMBAHASAN	44
4.1. Hasil Penelitian	44

4.2. Implementasi Perangkat Keras.....	44
4.3. Implementasi Coding IoT.....	46
4.4. Implementasi Server (API).....	47
4.5. Implementasi Website	51
4.6. Implementasi Pengujian	67
BAB V. KESIMPULAN DAN SARAN	68
5.1. Kesimpulan.....	68
5.2. Saran.....	69
DAFTAR PUSTAKA	70
LAMPIRAN	

DAFTAR GAMBAR

	HALAMAN
GAMBAR 2.1. Soil Moisture	21
GAMBAR 2.2 Sensor DHT22	22
GAMBAR 2.3 NodeMCU ESP8266	23
GAMBAR 3.1 Diagram Alur Penelitian	30
GAMBAR 3.2 Tanaman Sawi	31
GAMBAR 3.3 Rangkaian Alat	34
GAMBAR 3.4 Rancangan Alat	35
GAMBAR 3.7 Flowchart Alat	36
GAMBAR 3.8 Flowchart Sistem	38
GAMBAR 3.9 Use Case Diagram	39
GAMBAR 3.10 Class Diagram	40
GAMBAR 3.11 Activity Diagram	41
GAMBAR 3.12 Saquence Diagram	41
GAMBAR 3.13 Rancangan Website	43
GAMBAR 4.1 Hasil Website	64
GAMBAR 4.2 Indikator Sensor Kelembaban Tanah	64
GAMBAR 4.3 Indikator Senosr Suhu Udara	65
GAMBAR 4.4 Grafik Perubahan Sensor Kelembaban Tanah dari Waktu ke Waktu	65
GAMBAR 4.5 Grafik perubahan sensor kelembaban tanah dari waktu ke waktu	66

DAFTAR TABEL

		HALAMAN
TABEL 2.1.	Tabel Simbol Use Case	15
TABEL 2.2.	Tabel Simbol Activity Diagram	17
TABEL 2.3	Tabel Sequence Diagram	19
TABEL 2.4	Tabel Multiplicity Class Diagram	20
TABEL 2.5	Tabel Index Kualitas Cuaca dan Kelembaban Tanah	24
TABEL 2.6	Tabel Kelembaban Tanah	26
TABEL 2.7	Tabel Suhu Udara	27
TABEL 3.1	Tabel Jadwal Penelitian	32
TABEL 3.2	Tabel Rancangan Struktur Tabel Pengguna	42

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi Internet of Things (IoT) merupakan salah satu terobosan inovatif yang mampu menyediakan solusi cerdas di berbagai bidang, termasuk sektor pertanian. Melalui IoT, perangkat pintar dapat terhubung dengan berbagai sensor untuk memantau dan mengirimkan data secara langsung (real-time), sehingga mendukung pengambilan keputusan yang berbasis pada data. Teknologi ini memberikan pendekatan modern dalam mengatasi berbagai permasalahan, seperti dengan memanfaatkan sensor canggih seperti DHT22 untuk mengukur suhu dan kelembapan udara, serta sensor kelembapan tanah untuk mengetahui tingkat kadar air dalam media tanam. Sensor DHT22, yang juga dikenal dengan nama AM2302, digunakan dalam pembuatan sistem penyemprot tanaman otomatis berbasis Arduino Uno adalah sensor yang dapat mengukur suhu dan kelembapan udara di sekitarnya sensor ini lebih akurat dan presisi. (Roihan et al., 2021).

Data yang diperoleh dapat diolah dan ditampilkan dalam platform atau aplikasi berbasis web, Dengan demikian, petani dapat memantau kondisi lingkungan secara langsung dan mengambil langkah yang sesuai, seperti melakukan penyiraman, memberikan pupuk, atau melindungi tanaman dari kondisi cuaca yang ekstrem. Sawi adalah sekelompok tanaman dari genus Brassica yang bagian daun atau bunganya dimanfaatkan sebagai bahan makanan , khususnya sebagai sayuran, baik segar maupun di olah. Sawi mencakup beberapa spesies Brassica yang kadang-kadang mirip satu sama lain. Di Indonesia penyebutan sawi biasanya mengacu pada sawi hijau.

Sawi hijau terus dibudidayakan seiring dengan meningkatnya permintaan pasar. Namun, dalam praktiknya, penanaman sering dilakukan tanpa memperhatikan aspek kualitas, sehingga menghasilkan sayuran dengan mutu yang rendah. Salah satu kendala utama dalam menghasilkan sawi berkualitas tinggi adalah terbatasnya ketersediaan unsur hara dalam media tanam, terutama jika dibudidayakan secara konvensional. Sebagaimana telah diketahui, terdapat berbagai metode dalam budidaya sawi, baik menggunakan media tanah maupun sistem hidroponik. Suhu di suatu wilayah sangat dipengaruhi oleh ketinggian tempat dari permukaan laut (dpl). Umumnya, daerah dengan suhu antara 19°C hingga 21°C berada pada ketinggian 100–1200 mdpl. Semakin tinggi suatu lokasi dari permukaan laut, maka suhu udaranya cenderung semakin rendah.

Di sisi lain, suhu udara merupakan salah satu faktor penting yang memengaruhi pertumbuhan tanaman.. Sawi juga tumbuh optimal didataran tinggi dengan suhu 12°C-25°C dan kelembapan 75-100% lahan pertanian menjadi industri menghambat produksi sawi, meskipun permintaan terus meningkat, sehingga impot sawi juga meningkat, oleh karena itu banyak petani dan masyarakat mulai menanam sawi di perkarangan rumah menggunakan metode hidroponik, tradisional, atau sederhana. Tidak jauh berbeda dengan tanaman selada, Selada dapat tumbuh secara optimal pada kisaran suhu udara antara 25°C hingga 28°C, dengan tingkat kelembapan tanah berkisar antara 65% hingga 78% (Supriyanto, 2020). Hal serupa juga berlaku untuk tanaman stroberi, yang menunjukkan pertumbuhan dan produksi terbaik jika dibudidayakan di daerah dataran tinggi, khususnya pada ketinggian 1.000 hingga 1.500 meter di atas permukaan laut, dengan suhu udara yang sesuai. tidak terlalu panas antara 22-25° C pada siang hari dan 17-20° C pada malam hari, kelembapan udara 80-90%, curah hujan 600-700 mm/tahun, kelembapan tanah 40-60% (Deliana & Periyadi, 2020).

Ada beberapa masalah yang bisa diidentifikasi dari E-Monitoring Pendeteksi Cuaca dan Kelembapan Tanah Meningkatkan Hasil Panen Petani Sawi Berbasis Internet of Things (IoT). Yang dimana Cuaca tidak menentu Petani kesulitan memprediksi perubahan cuaca, seperti hujan tiba-tiba atau suhu ekstrem, yang dapat memengaruhi pertumbuhan sawi. Kelembapan tanah tidak terpantau dengan baik Tanah yang terlalu kering atau terlalu basah dapat merusak tanaman sawi, menurunkan produktivitas dan kualitas panen. Tanah yang terlalu basah dapat menyebabkan pembusukan pada batang Sawi di atas permukaan tanah dan juga Perubahan cuaca yang mendadak dapat menyebabkan stres pada tanaman, meningkatkan kemungkinan Sawi tidak dapat tumbuh sempurna.

Sistem E-Monitoring berbasis IoT tidak hanya memudahkan pemantauan kondisi lingkungan, tetapi juga mengurangi ketergantungan petani pada metode tradisional yang menyita waktu dan tenaga. Dengan kemampuan mengakses data secara cepat dan akurat, petani dapat meningkatkan efisiensi dalam pengelolaan lahan dan memastikan keberhasilan budidaya sawi, meskipun menghadapi tantangan perubahan cuaca. Penelitian ini berfokus pada pengembangan sistem E-Monitoring Deteksi Cuaca dan Kelembapan Tanah berbasis IoT untuk meningkatkan hasil panen sawi. Dengan memanfaatkan teknologi IoT, sistem ini dirancang untuk memberikan solusi praktis dan efektif bagi petani dalam mengelola lahan secara optimal, sekaligus mendukung penerapan teknologi modern di sektor pertanian.

Solusi utama yang dihasilkan dari E-Monitoring Pendeteksi Cuaca dan Kelembapan Tanah Meningkatkan Hasil Panen Petani Sawi Berbasis Internet of Things (IoT) adalah menciptakan sistem berbasis teknologi IoT yang mampu

memberikan data lingkungan secara akurat dan real-time. Sistem ini berfungsi sebagai alat bantu bagi petani untuk mengoptimalkan pengelolaan lahan mereka melalui pemantauan suhu, kelembapan udara, dan kadar air tanah yang memengaruhi pertumbuhan tanaman sawi. Dengan informasi tersebut, petani dapat mengambil keputusan berbasis data yang relevan, sehingga mengurangi ketergantungan pada intuisi atau metode tradisional yang kurang efisien. Solusi yang ditawarkan oleh sistem ini menjawab tantangan utama dalam budidaya sawi, yaitu ketergantungan pada cuaca dan kelembapan tanah. Dengan data yang akurat, efisiensi sumber daya yang meningkat, dan pengelolaan berbasis teknologi modern, sistem ini menjadi alat yang sangat berharga dalam mendukung pertanian berbasis teknologi di era digital. Tidak hanya meningkatkan produktivitas dan kualitas hasil panen, tetapi juga membantu menciptakan pertanian yang lebih ramah lingkungan dan berkelanjutan.

1.2. Rumusan Masalah

1. Bagaimana menerapkan tampilan dan alat yang dapat mempermudah petani dalam memantau dan mengetahui kualitas tanaman sawi selama proses perkembangannya?
2. Bagaimana merancang sebuah sistem berbasis IoT yang menggunakan sensor DHT22 untuk mengukur suhu dan kelembaban guna meningkatkan hasil panen tanaman sawi?
3. Apa saja tantangan dalam implementasi sistem IoT berbasis sensor DHT22 dan sensor kelembapan tanah di sektor pertanian?

1.3. Batasan Masalah

1. Penerapan tampilan dan alat: alat tersebut dibutuhkan untuk mempermudah petani mengetahui kualitas tanaman sawi saat proses perkembangan sawi
2. Perancangan sistem berbasis IoT menggunakan sensor DHT22 dibatasi pada pengukuran suhu dan kelembaban sebagai parameter utama untuk meningkatkan hasil panen tanaman sawi.
3. Sensor DHT22 memiliki keterbatasan akurasi dalam mengukur suhu dan kelembaban udara, terutama dalam kondisi lingkungan ekstrem (terlalu panas, terlalu dingin, atau kelembaban sangat tinggi). Begitu pula sensor kelembaban tanah dapat memberikan hasil kurang akurat jika tanah mengandung banyak logam atau air asin. Penggunaan teknologi Internet of Things (IoT) difokuskan pada pengembangan sistem pemantauan kondisi lingkungan secara real-time untuk mendukung budidaya tanaman sawi.

1.4. Tujuan Penelitian

1. Mengurangi kebutuhan pengamatan manual dengan menyediakan data yang akurat, seperti kelembaban tanah, cuaca, atau parameter lain yang memengaruhi pertumbuhan tanaman sawi.
2. Sebuah sistem berbasis IoT yang menggunakan sensor DHT22 untuk mengukur suhu dan kelembaban guna meningkatkan hasil panen pada tanaman sawi.
3. Merancang dan mengimplementasikan sistem berbasis IoT menggunakan sensor DHT22 untuk mengukur suhu dan kelembaban, sehingga dapat memberikan rekomendasi untuk meningkatkan hasil panen tanaman sawi.

1.5. Manfaat Penelitian

1. Penelitian ini memberikan manfaat untuk petani yang dimana Petani dapat memantau kondisi cuaca, suhu, dan kelembapan tanah secara real-time melalui sistem berbasis IoT, sehingga memungkinkan mereka mengambil tindakan tepat waktu untuk menjaga kesehatan tanaman sawi. Pemanfaatan teknologi IoT membantu memastikan kondisi lingkungan yang optimal bagi pertumbuhan sawi, meningkatkan hasil panen baik dari segi kuantitas maupun kualitas. Informasi yang disediakan oleh sensor memungkinkan petani membuat keputusan berdasarkan data yang valid, bukan hanya berdasarkan intuisi atau pengalaman. Petani dapat memantau kondisi lahan tanpa harus hadir secara langsung, sehingga meningkatkan fleksibilitas mereka dalam mengelola pertanian. Dengan deteksi dini kondisi lingkungan yang kurang ideal, seperti kelembapan tanah yang terlalu rendah atau suhu ekstrem, petani dapat mencegah kerusakan tanaman lebih dini. Penelitian ini membantu petani mengadopsi teknologi pertanian modern, mendukung transformasi menuju sistem pertanian yang lebih berkelanjutan dan ramah lingkungan.
2. Melalui penelitian ini, penulis mendapatkan kesempatan untuk memperdalam pemahaman mengenai teknologi Internet of Things (IoT) dan penerapannya dalam sektor pertanian, khususnya pada budidaya tanaman sawi. Penulis juga dapat meningkatkan keterampilan teknis dalam merancang dan mengembangkan sistem berbasis sensor, seperti DHT22 untuk memantau suhu dan kelembapan, serta sensor kelembapan tanah.
3. Dengan meningkatnya produktivitas pertanian, masyarakat dapat merasakan manfaat berupa ketersediaan pangan yang lebih berkualitas dan terjangkau.

Selain itu, penerapan teknologi IoT di bidang pertanian juga berkontribusi pada keberlanjutan lingkungan dengan mengurangi penggunaan sumber daya secara berlebihan dan meningkatkan efisiensi dalam pengelolaan pertanian.

BAB II

LANDASAN TEORI

2.1. E-Monitoring

E-Monitoring adalah proses pemantauan dan pelaporan yang dilakukan secara elektronik (online), memungkinkan data dikirim dan dipantau secara berkelanjutan guna menilai kualitas serta efektivitas sistem pengendalian, sehingga memastikan bahwa setiap prosedur dijalankan sesuai dengan ketentuan. Salah satu pendekatan dalam arsitektur layanan web yang dapat diterapkan untuk sistem monitoring perencanaan produksi secara terintegrasi di perusahaan adalah metode SOA (Service-Oriented Architecture).

Penerapan Manajemen Monitoring Perencanaan Produksi yang terintegrasi memungkinkan penyampaian informasi secara cepat dan akurat, sehingga perencanaan kebutuhan barang dapat dilakukan secara lebih terkontrol dan efisien. Hal ini juga membantu menekan biaya manajemen serta mendukung perkembangan perusahaan, khususnya dalam aspek pengendalian produksi, yang dapat berjalan seiring dengan pengawasan yang baik dan kemudahan dalam akses informasi (Nurdin et al., 2019).

E-Monitoring memproses pemantauan, pengawasan, dan pengelolaan data secara elektronik menggunakan teknologi informasi dan komunikasi. Sistem ini memungkinkan pengguna untuk mengamati dan menganalisis aktivitas atau kondisi tertentu secara real-time atau periodik, baik melalui perangkat keras (sensor, kamera, dll.) maupun perangkat lunak (platform digital, aplikasi berbasis cloud). E-monitoring sering diterapkan di berbagai bidang, seperti pertanian, kesehatan, pendidikan, dan industri untuk meningkatkan efisiensi, akurasi, dan pengambilan keputusan (Tukino, 2018).

2.2. Sistem

Sistem dapat diartikan sebagai suatu rangkaian elemen yang saling berinteraksi dan terhubung dalam suatu jaringan kerja untuk mencapai tujuan tertentu. Sistem merupakan kesatuan proses yang terdiri dari berbagai komponen yang bekerja secara terpadu guna melaksanakan suatu aktivitas. Dalam konteks yang lebih luas, sistem merupakan gabungan dari berbagai unsur seperti perangkat keras, perangkat lunak, manusia, prosedur, dan data, yang beroperasi dalam suatu lingkungan tertentu untuk menghasilkan keluaran (output) berdasarkan masukan (input) yang diterima.

Elemen-elemen dalam suatu sistem dapat terdiri dari perangkat keras (hardware), perangkat lunak (software), data, manusia, dan prosedur yang saling berinteraksi dalam sebuah lingkungan tertentu. Sistem beroperasi dengan cara menerima masukan (input) dari lingkungan, kemudian mengolahnya melalui proses tertentu untuk menghasilkan keluaran (output) yang bermanfaat sesuai dengan tujuan yang ingin dicapai. Setiap komponen dalam sistem memiliki peran dan fungsi masing-masing yang saling mendukung. Oleh karena itu, jika salah satu elemen tidak berjalan dengan baik, maka kinerja sistem secara keseluruhan dapat terpengaruh atau terganggu. (Saputro et al., 2017).

2.3. Website

Website adalah sebuah lokasi virtual di internet yang terdiri dari satu atau beberapa halaman web yang dapat diakses melalui browser internet. Halaman-halaman ini dapat berisi berbagai jenis konten, seperti teks, gambar, video, dan elemen interaktif lainnya. Tujuan utama dari sebuah website adalah untuk

menyediakan informasi, layanan, atau produk kepada pengunjungnya. Websites dapat memiliki berbagai macam tujuan, termasuk memberikan informasi tentang suatu topik, mempromosikan bisnis atau organisasi, menyediakan platform untuk berbagi konten, atau menyediakan layanan online seperti e-commerce atau perbankan digital.

Tahapan ini menggunakan bahasa pemrograman JavaScript, JavaScript adalah bahasa pemrograman yang relatif sederhana karena tidak digunakan untuk membuat aplikasi applet. Dengan JavaScript, kita dapat dengan mudah menciptakan halaman web yang interaktif. Kode JavaScript biasanya ditulis langsung di dalam file HTML. Dalam penelitian ini juga digunakan aplikasi API, yang merupakan singkatan dari Application Programming Interface. API adalah teknologi yang memudahkan pertukaran data atau informasi antar aplikasi perangkat lunak. API berfungsi sebagai antarmuka virtual yang menghubungkan dua fungsi aplikasi yang bekerja bersama. Selain itu, API juga berisi metode bagi programmer untuk memanfaatkan fitur tertentu pada sebuah komputer. Dengan demikian, API dapat ditemukan dalam berbagai sistem seperti windowing, sistem file, basis data, dan jaringan. (Gerardhy Lainsamputty, 2019).

Di era kemajuan pesat teknologi pengembangan aplikasi Android, masih banyak programmer yang belum menguasai cara pembuatan aplikasi Android. Hal ini terlihat dari banyaknya situs web yang sudah dioptimalkan untuk tampilan layar aplikasi Android, namun belum dilengkapi dengan aplikasi Android resmi. Dalam teknologi pengembangan aplikasi Android terdapat fitur bernama WebView, yaitu teknologi yang memungkinkan halaman web dibuat secara otomatis berdasarkan data dasar yang biasanya disimpan dalam DBMS, lalu diintegrasikan ke dalam aplikasi Android. WebView memungkinkan

pengembang aplikasi untuk dapat menampilkan konten dari laman web secara langsung di dalam aplikasi.

Dalam pengembangan aplikasi Android, WebView menyediakan berbagai API (Application Programming Interface) yang memungkinkan kode aplikasi untuk berinteraksi dengan kode JavaScript di dalam halaman web, mengelola aktivitas web, serta memodifikasi aktivitas tersebut. Dengan fitur-fitur ini, aplikasi dapat berfungsi sebagai "browser" yang disesuaikan khusus untuk aplikasi web tertentu. Saat ini, di pasar Android, sebanyak 86 persen dari 20 aplikasi terpopuler dalam 10 kategori berbeda merupakan aplikasi Android yang menggunakan WebView (Luo et al., 2011). Beberapa keunggulan penggunaan WebView meliputi: pemanfaatan layar yang efisien, dukungan navigasi terintegrasi baik untuk halaman internal maupun eksternal, serta kemampuan yang efektif dalam identifikasi halaman, pengaturan tampilan, dan pengambilan konten halaman. (Putra et al., 2021)

2.4. Basis Data (Database)

Basis data atau database adalah kumpulan data yang tersusun secara terorganisir sehingga memudahkan penyimpanan dan pengelolaan data tersebut. Penggunaan database dapat ditemukan dalam berbagai aspek kehidupan sehari-hari, seperti pada mesin ATM, sistem akademik di universitas atau sekolah, serta sistem informasi penjualan. Salah satu tujuan utama database adalah memberikan pengguna pandangan yang abstrak terhadap data, yaitu dengan menyembunyikan detail teknis mengenai cara data disimpan dan dikelola. Sistem database harus dirancang agar semudah mungkin digunakan. dimengerti karena kebanyakan pengguna sistem database adalah orang-orang yang kurang terlatih di bidang teknologi.

2.5. Server

Server adalah sebuah sistem komputer atau perangkat keras yang dirancang untuk menyediakan layanan, sumber daya, atau data kepada komputer lain, yang dikenal sebagai klien, melalui jaringan. Server berfungsi sebagai pusat pengolahan data dan manajemen sumber daya, memungkinkan berbagai perangkat untuk berkomunikasi dan berbagi informasi secara efisien. Dalam konteks jaringan komputer, server dapat diartikan sebagai entitas yang menerima permintaan dari klien, memproses permintaan tersebut, dan mengirimkan kembali hasilnya.

Server dapat berupa perangkat fisik (hardware) yang berdiri sendiri, seperti server rack atau blade server, atau dapat juga berupa perangkat lunak (software) yang berjalan di atas sistem operasi tertentu. Server sering kali memiliki spesifikasi yang lebih tinggi dibandingkan dengan komputer biasa, termasuk prosesor yang lebih cepat, kapasitas memori yang lebih besar, dan penyimpanan yang lebih luas, untuk menangani beban kerja yang lebih berat dan banyak permintaan secara bersamaan.

2.6. MySQL (My Structure Query Language)

MySQL adalah sebuah program pembuat basis data yang bersifat Open Source, yang berarti dapat digunakan oleh siapa saja dan berjalan di berbagai platform seperti Windows maupun Linux. MySQL juga merupakan perangkat lunak sistem manajemen basis data SQL yang mendukung jaringan, sehingga cocok digunakan untuk aplikasi multi pengguna. MySQL sering disebut sebagai sistem manajemen basis data relasional, di mana data disimpan dalam tabel-tabel terpisah yang saling terhubung melalui relasi tertentu, memungkinkan pengambilan data gabungan dari beberapa tabel dalam satu permintaan. Untuk administrasi database, seperti membuat database atau tabel, dapat digunakan aplikasi berbasis web seperti PHPMysqlAdmin yang biasanya dijalankan melalui paket aplikasi XAMPP.

(Anwar et al., 2016).

2.7. UML (Unified Modelling Language)

Unified Modeling Language (UML) adalah sekumpulan konvensi pemodelan yang digunakan untuk mendefinisikan atau menggambarkan suatu sistem perangkat lunak yang berhubungan dengan objek. UML merupakan alat bantu yang sangat efektif dalam pengembangan sistem berorientasi objek karena menyediakan bahasa pemodelan visual yang memungkinkan para pengembang untuk membuat cetak biru (blueprint) dari konsep mereka dalam format yang standar. UML berperan sebagai penghubung dalam komunikasi berbagai aspek sistem dengan menggunakan berbagai elemen grafis yang dapat dikombinasikan sesuai kebutuhan.

Unified Modeling Language (UML) biasa digunakan untuk :

- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan use case dan actor.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan interaction diagrams.
- c. Menggambarkan representasi struktur static sebuah sistem dalam bentuk class diagrams.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan state transition diagrams.
- e. Menyatakan arsitektur implementasi fisik menggunakan component and development.
- f. Menyampaikan atau memperluas fungsionalitas dengan stereotypes.

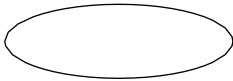
UML merupakan bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, merinci, dan membangun perangkat lunak. Saat ini, UML banyak digunakan dalam industri sebagai standar bahasa pemodelan umum dalam pengembangan perangkat dan sistem. Dalam perancangan berorientasi objek berbasis UML, terdapat empat alat bantu utama yang digunakan, yaitu: Use Case Diagram, Activity Diagram, Sequence Diagram dan Class Diagram.

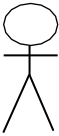
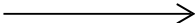
(Alfina & Harahap, n.d.)

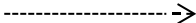
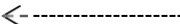
1. Use Case Diagram

Use Case Diagram adalah Alat pemodelan untuk menciptakan perilaku sistem informasi yang sedang dikembangkan adalah diagram use case. Use case menggambarkan bagaimana satu atau beberapa aktor berinteraksi dengan sistem informasi. Dengan kata lain, use case membantu menentukan fungsionalitas sistem informasi mana yang tersedia dan siapa yang berwenang untuk menggunakannya. Simbol-simbol berikut digunakan dalam diagram use case:

Tabel 2.1. Simbol Use Case

Gambar	Keterangan
	Use case yang sering kali dimulai dengan kata kerja, menjelaskan fungsionalitas yang ditawarkan oleh sistem sebagai unit yang berkomunikasi satu sama lain dan dengan aktor

	Aktor adalah representasi manusia atau sistem lain yang memulai operasi sistem target. Menentukan tanggung jawab peran dan pembagian kerja dalam sistem target diperlukan untuk mengidentifikasi aktor. Seseorang atau sistem dapat memainkan beberapa peran. Perlu diingat bahwa ketika aktor terlibat dengan use case, mereka tidak memiliki kendali atas kasus tersebut.
	Alih-alih menunjukkan aliran data, hubungan antara aktor dan kasus penggunaan—ditunjukkan oleh garis tanpa panah—menunjukkan siapa atau apa yang secara langsung meminta interaksi.
	Kasus penggunaan dan aktor dikaitkan dengan panah terbuka yang menunjukkan saat aktor terlibat dalam interaksi sistem pasif.

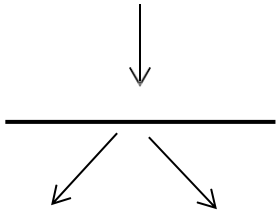
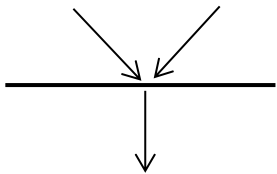
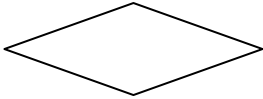
	<i>Include</i> , dipanggil oleh kasus penggunaan lain (misalnya, memanggil fungsi program) atau dalam kasus penggunaan lain (diperlukan).
	<i>Extend</i> , merupakan kelanjutan dari kasus penggunaan yang berbeda asalkan kondisi atau prasyarat tertentu terpenuhi.

2. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas menjelaskan alur kerja atau aktivitas suatu sistem atau proses bisnis. Simbol-simbol berikut digunakan dalam diagram aktivitas:

Tabel 2.2. Simbol Activity Diagram

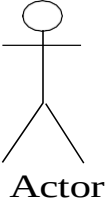
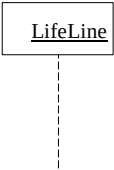
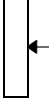
Gambar	Keterangan
	Aktivitas dimulai di titik awal, yang berada di sudut kiri atas.
	Titik akhir, akhir aktivitas.
	Aktivitas, yang mencirikan prosedur atau aktivitas bisnis

	Garpu digunakan untuk menunjukkan aktivitas paralel atau untuk menggabungkan dua aktivitas paralel menjadi satu aktivitas tunggal.
	Dekomposisi ditunjukkan dengan penggunaan join (gabungan) atau rake.
	Titik Keputusan menunjukkan pilihan untuk membuat keputusan, seperti benar atau salah.
	Swimlane adalah divisi diagram aktivitas yang menggambarkan siapa yang bertanggung jawab untuk apa.

3. Sequence Diagram

Diagram Urutan menggambarkan perilaku objek dalam suatu kasus penggunaan dengan menggambarkan masa pakai objek dan pesan yang dikirim dan diterima antar objek. Simbol yang digunakan dalam Diagram Urutan adalah:

Tabel 2.3. Sequence Diagram

Gambar	Keterangan
	interaksi antara sistem informasi dan sistem, orang, atau proses lainnya.
	menggambarkan satu entitas dalam diagram sekuens, dilambangkan dengan kotak. Ada label, stereotip, atau contoh untuk ini.
	Pesan, simbol mengirim pesan antar kelas.
	Rekursif mengacu pada komunikasi dengan diri sendiri.
	
	Ada aktivasi sepanjang garis hidup, yaitu garis titik yang menempel pada objek.

4. Class Diagram (Diagram Kelas)

Diagram kelas adalah hubungan antara kelas dan deskripsi menyeluruh dari setiap kelas dalam model desain sistem, yang juga menggambarkan tugas dan peraturan entitas yang membentuk perilaku sistem. Diagram kelas juga menampilkan batasan yang berkaitan dengan objek yang terhubung serta karakteristik dan fungsi kelas. Kelas, Hubungan, Asosiasi, Generalisasi dan Agregasi, Atribut, Operasi/Metode, dan Visibilitas semuanya umumnya disertakan dalam diagram kelas.

Tabel 2.4. Multiplicity Class Diagram

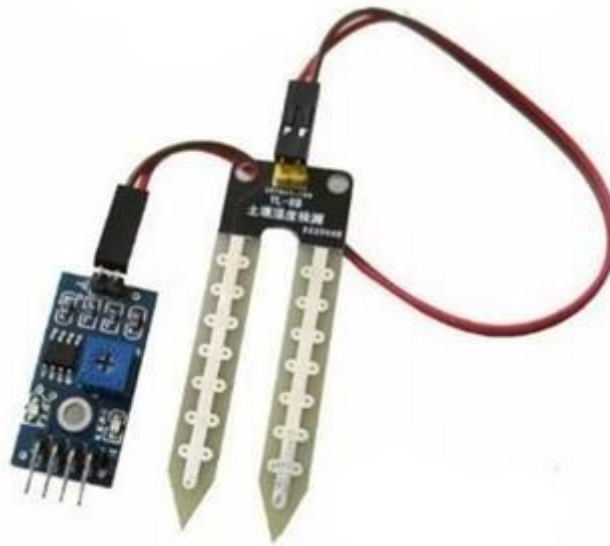
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Mungkin tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Mungkin tidak ada, maksimal 1
n..n	Batas antara. Contoh 2..4 memiliki arti minimum 2 maksimum 4

2.8. Soil Moisture

Sensor kelembaban tanah FC-28 mampu mendeteksi kelembaban tanah. Meskipun sederhana, sensor ini sangat cocok untuk memantau taman kota, atau tingkat air pada tanaman pekarangan. Sensor ini terdiri dua probe untuk melewati arus melalui tanah, kemudian membaca resistansinya untuk mendapatkan nilai tingkat kelembaban. Semakin banyak air membuat tanah lebih

Tanah kering membuatnya sangat sulit mengalirkan listrik (resistansi tinggi), sedangkan tanah basah membuatnya mudah (resistansi rendah).

Sensor ini sangat berguna untuk memantau kelembapan tanah atau mengingatkan tanaman akan kadar airnya. Sensor kelembapan tanah FC-28 memiliki rentang nilai ADC 1024 bit, mulai dari 0 hingga 1023 bit, tegangan keluaran 0 hingga 4,2 V, arus 35 mA, dan spesifikasi tegangan masukan 3,3 V atau 5 V. Cara kerja sensor kelembapan alat ini adalah dengan menanam sensor kelembapan di tanah. Sensor ini mengukur jumlah kelembapan yang ada. Parameter digunakan untuk mengatur kelembapan. (Verdi & Sarwoko, n.d.)



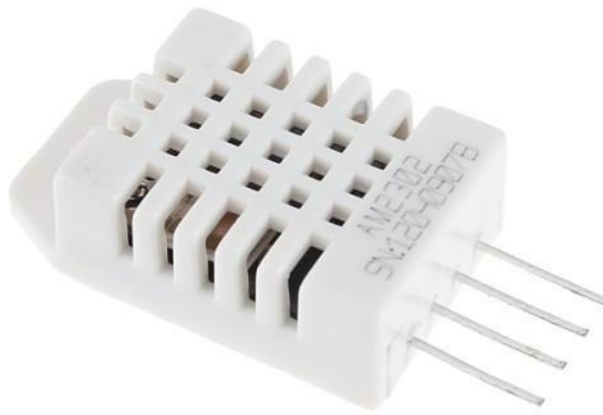
Gambar 2.1. Soil Moisture

2.9. Sensor DHT22

Sensor DHT22 adalah sensor digital kelembaban dan suhu relatif. Sensor DHT22 menggunakan kapasitor dan termistor untuk mengukur udara disekitarnya

dan sinyal keluaran pin data. Berdasarkan respons proses pengumpulan data yang cepat, ukurannya yang kecil, dan biaya yang cukup rendah dibandingkan dengan instrumen termohigrometer, DHT22 dikatakan memiliki kualitas pembacaan yang baik.

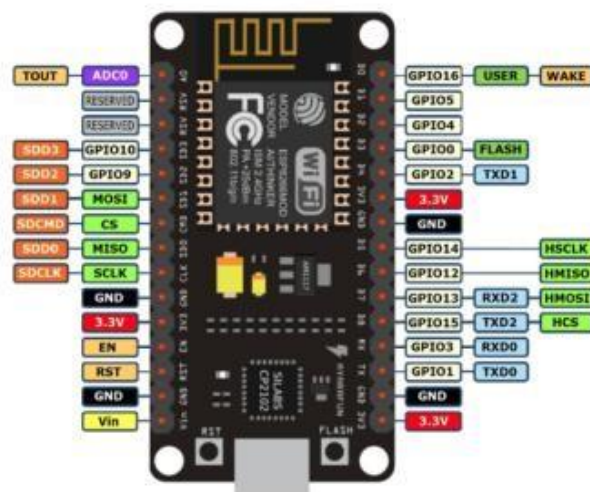
Suhu dan kelembapan adalah dua karakteristik lingkungan yang dapat dipantau secara bersamaan oleh sensor DHT22. Mirip dengan DHT11, DHT22 (kadang-kadang disebut AM2302) adalah sensor suhu dan kelembapan, tetapi menawarkan manfaat seperti MCU 8-bit mengonversi dan menghitung output, yang sudah berupa sinyal digital. DHT-22 menghasilkan data pengukuran yang lebih tepat dan presisi daripada DHT11. Dapat ditempatkan di mana saja karena memiliki rentang pengukuran suhu dan kelembapan yang lebih luas dan dapat mengirim sinyal output melalui kabel yang panjang (hingga 20m).



Gambar 2.2. Sensor DHT22

2.6. NodeMCU ESP8266

Berdasarkan chip ESP8266, NodeMCU adalah papan listrik yang dapat beroperasi sebagai mikrokontroler dan membuat koneksi WiFi. NodeMCU dapat dikembangkan menjadi aplikasi pemantauan atau pengendalian untuk proyek Internet of Things karena memiliki beberapa pin I/O. Arduino IDE dapat digunakan untuk memprogram NodeMCU ESP8266 dengan kompilasi Arduino-nya. Bentuk fisik NodeMCU ESP 8266 mencakup port USB (micro USB) untuk memfasilitasi pemrograman. NodeMCU ESP8266 adalah modul turunan dari modul platform IoT (Internet of Things) tipe keluarga ESP8266 ESP-12. Modul ini hampir identik dengan platform modul Arduino dalam hal fungsionalitas; satu-satunya perbedaan adalah modul ini dirancang khusus untuk "Terhubung ke Internet". (Nurul Hidayati Lusita Dewi, n.d.)



Gambar 2.3. NodeMCU ESP8266.

2.7. Index Kualitas Cuaca Dan Kelembapan Tanah

Indikator cuaca dan kelembapan tanah digunakan untuk menilai dan melacak faktor lingkungan yang memengaruhi sawi hijau dan pertumbuhan tanaman lainnya. Banyak faktor terkait cuaca dan kelembapan tanah yang memengaruhi kesehatan dan hasil tanaman disertakan dalam indeks ini. Tidak adanya nutrisi dalam media tanam selama budidaya konvensional merupakan salah satu masalah yang membatasi produksi berkualitas tinggi. Prasyarat penanaman yang penting meliputi tanah gembur, banyak mengandung zat organik. Seperti yang kita ketahui bahwa sawi tumbuh optimal di daratan tinggi dengan suhu 12- 22°C dan kelembapan 75-100% (Saputro et al., 2017) .Berikut adalah penjelasan lebih lanjut tentang indeks ini :

Tabel 2.5. Tabel Index Kualitas Cuaca Dan Kelembapan Tanah

Faktor	Rentang Optimal	Dampak Pada Hasil Panen Sawi
Kualitas Cuaca		
A.Suhu	20°C - 25°C	-Mendukung pertumbuhan optimal dan pembungaan. - Suhu >30°C dapat mengurangi pembentukan biji dan hasil panen.
B. Kelembapan Udara	50% -70%	- Mengurangi penguapan air dari tanah dan tanaman.

		- Kelembapan < 30% dapat menyebabkan stres pada tanaman.
C. Curah Hujan	300 mm – 500 mm/tahun	- Menjaga ketersediaan air dalam tanah. - Curah hujan <200 mm dapat menyebabkan kekeringan dan mengurangi hasil.
D. Paparan Sinar Matahari	6-8 jam/Hari	- Penting untuk fotosintesis. - Paparan berlebihan dapat menyebabkan stres panas dan mengurangi hasil.
Kelembapan Tanah		
A. Kelembapan Tanah	60% - 80%	- Mendukung pertumbuhan akar dan penyerapan nutrisi. - Kelembapan < 40% dapat menyebabkan stres dan mengurangi hasil.
B. Jenis Tanah	-	-Jenis tanah mempengaruhi kemampuan menahan kelembapan.

		- Tanah berpasir mengering lebih cepat, tanah liat menahan kelembapan lebih lama.
C. Pengelolaan Kelembapan	-	- Penggunaan mulsa dan irigasi efisien membantu menjaga kelembapan. - Memantau kelembapan tanah secara berkala penting untuk pengambilan keputusan irigasi.

Tabel 2.6. Kelembaban Tanah

Kategori	Rentang Optimal	Dampak Pada Hasil Panen Sawi
Kelembapan Tanah		
Baik	60% - 90%	Kelembaban dalam rentang ini mendukung pertumbuhan akar yang baik dan penyerapan nutrisi.
Buruk	< 40% - > 90%	Kelembaban terlalu rendah dapat menyebabkan stres pada tanaman, sedangkan

		kelembaban terlalu tinggi dapat menyebabkan pembusukan akar.
--	--	--

Tabel 2.7. Suhu Udara

Kategori	Rentang Optimal	Dampak Pada Hasil Panen Sawi
Suhu Udara		
Baik	15,6°C - 21,1°C	Suhu dalam rentang ini mendukung fotosintesis dan pertumbuhan optimal.
Buruk	< 10°C atau > 30°C	Suhu terlalu rendah dapat memperlambat pertumbuhan, sedangkan suhu terlalu tinggi dapat menyebabkan layu dan penurunan kualitas daun.

2.8. Internet Of Things (IoT)

Internet of Things (IoT) adalah teknologi yang sangat populer akhir-akhir ini di seluruh dunia, yang membantu menghubungkan objek-objek seperti sensor, kendaraan, peralatan rumah sakit, industri, dan peralatan rumah tangga melalui internet. Perangkat pintar yang ringkas merupakan bagian penting dari IoT. Perangkat ini bervariasi dalam hal penggunaan, ukuran, sumber daya komputasi, dan memori. (Najib et al., 2020).

Internet of Things (IoT) adalah arsitektur koneksi jaringan global yang menggunakan pengumpulan data dan teknologi komunikasi untuk menghubungkan benda fisik dan virtual, menurut Koordinator dan mendukung tindakan untuk aktivitas dan standarisasi terkait RFID global. Jaringan yang ada dan pertumbuhan Internet membentuk infrastruktur IoT. Selain memiliki tingkat otonomi penangkapan data, transfer peristiwa, konektivitas jaringan, dan interoperabilitas yang tinggi, IoT menyediakan identifikasi objek, identifikasi sensor, dan kemampuan koneksi yang berfungsi sebagai dasar untuk pembuatan layanan dan aplikasi kolaboratif yang beroperasi secara independen. (Setiadi, 2018).

BAB III

METODOLOGI PENELITIAN

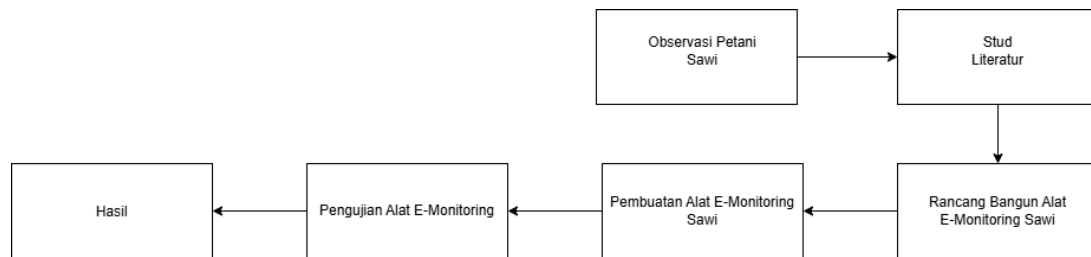
3.1. Metode Penelitian

Penelitian yang difokuskan pada pengembangan dan penelitian Mencari pola dan urutan pertumbuhan lintas waktu merupakan fokus dari karya ini. Perbaikan atau kemajuan yang dilakukan oleh orang-orang seperti penulis saat ini merupakan subjek dari penelitian ini. Menentukan evolusi model Internet of Things dan situs web yang sedang dikembangkan merupakan tujuan dari penelitian ini.

Peneliti dalam penelitian ini menggunakan metode pendekatan jenis penelitian R&D (Research and Development), yang juga dikenal sebagai penelitian pengembangan, dari beberapa upaya jenis penelitian yang biasa digunakan. Metodologi penelitian yang digunakan adalah memproduksi barang-barang tertentu dan menguji kemanjurannya, serta pengembangan produk yang dilakukan peneliti sendiri.

3.2. Alur Penelitian

Penelitian perancangan E-monitoring pendeteksi cuaca dan kelembaban tanah untuk meningkatkan hasil panen petani sawi berbasis iot menggunakan NodeMCU ESP8266 ini meliputi beberapa tahapan pelaksanaan yang akan ditunjukkan pada metode diagram dibawah ini :



Gambar 3.1 Diagram Alur Penelitian

Pada diagram diatas dimulai dengan observasi petani sawi kemudian melakukan studi literatur yang bertujuan untuk mempelajari teori penting yang mendukung desain dan perencanaan alat E-monitoring sawi yang dimana mempelajari sistem kerja dari E-Monitoring pendeteksi Cuaca dan Kelembaban tanah untuk meningkat hasil panen Petani Sawi. Kemudian pembuatan alat E-Monitoring pendeteksi Cuaca dan Kelembaban tanah dan dilanjutkan dengan pengujian alat E-Monitoring pendeteksi Cuaca dan Kelembaban tanah dari website hingga alat. Langkah selanjutnya yang akan dilakukan setelah pembuatan yaitu memasuki proses pengujian yang mencakup website dan alat yang dimana Ketika memasuki proses pengujian maka akan mendapatkan data hasil pengujian. Setelah mendapatkan data hasil pengujian maka akan dilakukan hasil dan pembahasan dari data peoses pengujian tersebut. Ketika hasil dan pembahasan sudah dilakukan maka penulis akan membuat kesimpulan dari penelitian dan prosedur penelitian sudah selesai.

3.3. Observasi Penelitian

Penulis melakukan pengamatan di perkebunan sawi yang berada pada lingkungan perkomplekan daerah marelان dan menemukan masalah terkait kegagalan panen pada tanaman sawi. Masalah yang dihadapi adalah kesalahan yang sering terjadi saat cuaca yang ekstrem mengakibatkan gagal nya pertumbuhan tanaman sawi.



Gambar 3.2 Tanaman Sawi

Dalam konteks Salah satu penyebab utama adalah perubahan suhu yang drastis, di mana suhu yang sangat tinggi atau rendah dapat mempengaruhi fotosintesis dan perkembangan tanaman. Ketika tanaman mengalami stres akibat suhu yang tidak sesuai, hal ini dapat menyebabkan layu atau bahkan kematian. Selain itu, curah hujan yang tidak menentu juga berkontribusi pada masalah ini hujan yang berlebihan dapat menyebabkan genangan air yang merusak akar tanaman, sementara kekurangan air dapat menghambat pertumbuhan dan menyebabkan tanaman kering.

3.4. Tempat Penelitian

Penelitian ini dilakukan pada sebuah Perkebunan Sawi yang bertempat
dilingkungan perkomplekan rumah di Komplek Panggon Indah Kec. Medan
Marelan

3.5. Jadwal Penelitian

Waktu pelaksanaan Tabel berikut menunjukkan periode enam bulan dari Desember 2024 hingga Mei 2025 di mana penelitian ini dilakukan. Tabel 3.1.

Jadwal Penelitian

Kegiatan	Waktu Kegiatan																							
	Des 2024				Jan 2025				Feb 2025				Mar 2025				April 2025				Mei 2025			
	Minggu Ke				Minggu Ke				Minggu Ke				Minggu Ke				Minggu Ke				Minggu Ke			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Pengajuan Judul																								
Penyusunan Proposal Bab I – Bab III																								
Perakitan Alat																								
Pengujian Sistem																								
Menentukan Hasil																								
Penulisan Bab IV – Bab V																								
Pengumpulan Skripsi																								

3.6. Jenis Penelitian

Jenis penelitian yang dilakukan adalah dengan merancang alat serta menguji pada E- Monitoring Pendeteksi Cuaca dan Kelembaban Tanah untuk Meningkatkan Hasil Panen Petani Sawi yang mana menjadi sarana informasi bagi user dengan menggunakan NodeMCU ESP8266 yang terkonfigurasi melalui Website dengan menggunakan teknologi Internet of Things (IoT). Penelitian ini diharapkan akan menghasilkan data yang menunjukkan Hasil Panen Petani Sawi yang lebih meningkat.

3.7. Studi Literatur

Studi literatur ini menunjukkan bahwa penerapan e-monitoring berbasis IoT untuk mendeteksi cuaca dan kelembaban tanah memiliki potensi besar untuk meningkatkan hasil panen petani sawi. Dengan memberikan data akurat dan real time, sistem ini dapat membantu petani dalam pengelolaan lahan mereka. Namun, tantangan dalam akses teknologi, biaya, dan keterampilan pengguna perlu diatasi untuk memastikan keberhasilan implementasi.

3.8. Alat dan Bahan Penelitian

Adapun alat yang digunakan dalam E-Monitoring Pendeteksi Cuaca dan Kelembaban Tanah untuk Meningkatkan Hasil Panen Petani Sawi Berbasis website:

Perangkat Keras (Hardware)

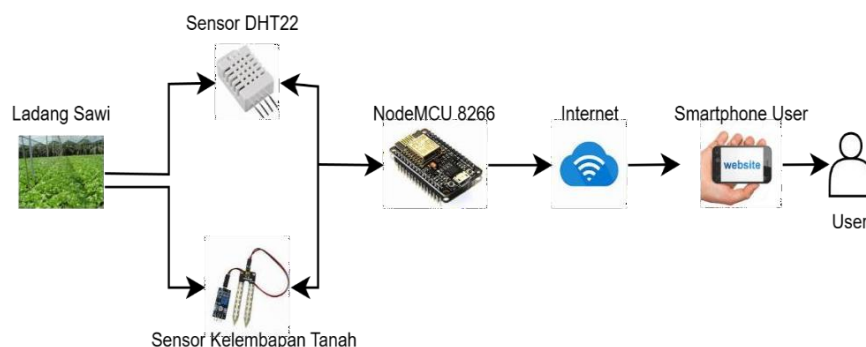
1. Laptop
2. Smartphone

Perangkat Lunak (Software)

1. Visual Studio Code
2. Arduino Uno
3. GraphQL

3.9. Rancangan alat E-Monitoring Sawi

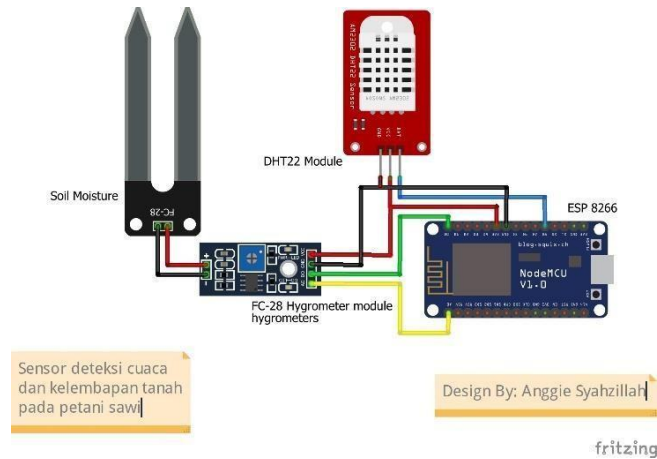
Adapun rangkaian alat dari pada E-Monitoring Pendeteksi cuaca dan kelembapan tanah untuk menunjukkan rangkaian proses kerja yang dilakukan oleh E-Monitoring Pendeteksi cuaca dan kelembapan tanah.



Gambar 3.3 Rangkaian Alat

Gambar diatas menjelaskan tentang rangkaian alat E-Monitoring pendeteksi cuaca dan kelembapan tanah. Yang Dimana sensor cuaca dan kelembaban lalu mengirim data NodeMCU ESP8266 yang terhubung ke internet setelah itu di kirim ke website yang terdapat pada smartphone user.

Kemudian rancangan alat E-Monitoring Pendeteksi cuaca dan kelembaban tanah.



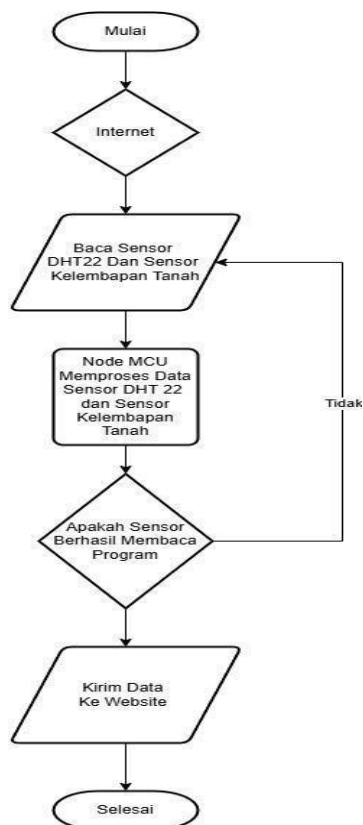
Gambar 3.4 Rancang alat

Rancangan ini bertujuan untuk menciptakan perangkat sensor yang membantu petani sawi memaksimalkan panen mereka dengan mendeteksi kelembapan tanah dan cuaca. Sistem ini menggunakan dua sensor: sensor DHT22 yang terhubung ke ESP8266 melalui pin digital D2 dan sensor kelembapan tanah FC-28 yang terpasang pada ESP8266 melalui pin analog A0. Setelah memproses data dari kedua sensor, ESP8266 menggunakan jaringan wifi untuk mendistribusikannya ke perangkat lain. Dengan penggunaan data ini, petani dapat meningkatkan hasil panen secara lebih efektif dengan melacak kondisi cuaca dan tanah secara real time.

3.10. Flowchart Sistem dan Flowchart Alat

Flowchart menggambarkan alur kerja dari E-Monitoring pendeteksi cuaca dan kelembaban tanah untuk meningkatkan hasil panen petani sawi sesuai dengan data yang terdapat pada website. Dapat dilihat padagambar berikut ini:

Flowchart Alat



Gambar 3.7 Flowchart alat

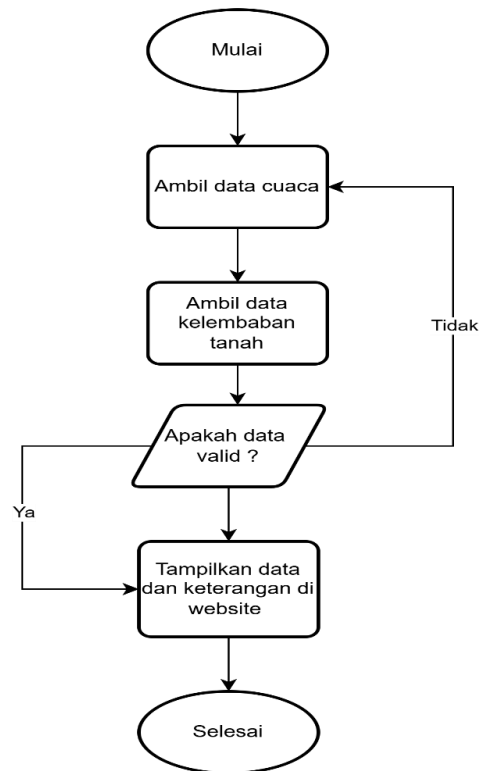
Berikut penjelasan dari gambar Flowchart Alat diatas :

1. Mulai: Flowchart ini dimulai dengan simbol oval yang menunjukkan titik awal program.
2. Internet: Program akan terhubung ke internet, yang diwakilkan oleh simbol diamond.

3. Baca Sensor DHT22 Dan Sensor Kelembapan Tanah: Program akan membaca data dari sensor DHT22 dan sensor kelembapan tanah.
4. Node MCU Memproses Data Sensor DHT 22 dan Sensor Kelembapan Tanah: Node MCU akan memproses data yang diterima dari sensor DHT22 dan sensor kelembapan tanah.
5. Apakah Sensor Berhasil Membaca Program: Program akan mengecek apakah sensor berhasil membaca data, diwakilkan oleh simbol diamond.
6. Tunggu: Jika sensor gagal membaca data, program akan menunggu hingga sensor berhasil membaca data.
7. Kirim Data Ke Website: Jika sensor berhasil membaca data, program akan mengirimkan data ke website, diwakilkan oleh simbol paralelogram.
8. Selesai: Program selesai, diwakilkan oleh simbol oval.

Setelah penjelasan Flowchart Alat diatas, kemudain Berikut gambar dan penjelasan dari Flowchart Sistem

Flowchart Sistem



Gambar 3.8. Flowchart Sistem

Berikut Penjelasan dari Flowchart Sistem diatas:

1. Mulai: Proses dimulai dari titik ini.
2. Ambil data cuaca: Sistem mengambil informasi data cuaca terkini, kemungkinan dari API atau perangkat sensor cuaca.
3. Ambil data kelembaban tanah: Setelah mendapatkan data cuaca, sistem mengambil data kelembaban tanah, misalnya melalui sensor atau sumber data lain.
4. Apakah data valid: Sistem memeriksa validitas data yang telah diambil (baik data cuaca maupun kelembaban tanah). Validasi dapat mencakup

pengecekan kelengkapan, akurasi, atau format data. Jika data tidak valid, proses dihentikan atau data tidak digunakan lebih lanjut. dan Jika data valid, proses dilanjutkan.

5. Tampilkan data dan keterangan di website: Jika data dinyatakan valid, sistem menampilkan data cuaca dan kelembaban tanah beserta keterangannya pada website.
6. Selesai:Proses selesai.

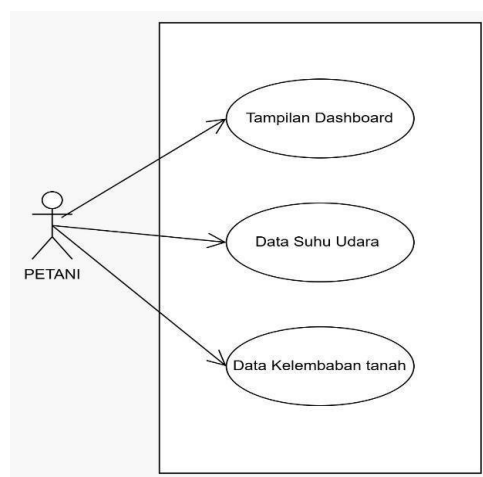
3.11. Desain Sistem

Perancangan ini akan memberikan penjelasan mengenai rancangan aplikasi pendeteksi cuaca dan kelembaban tanah untuk hasil panen petani sawi.

3.11.1. Use Case Diagram

Diagram use case menggambarkan bagaimana aktor dan sistem berinteraksi.

Gambar 3.9 di bawah ini menunjukkan proses sistem yang akan dirancang.:

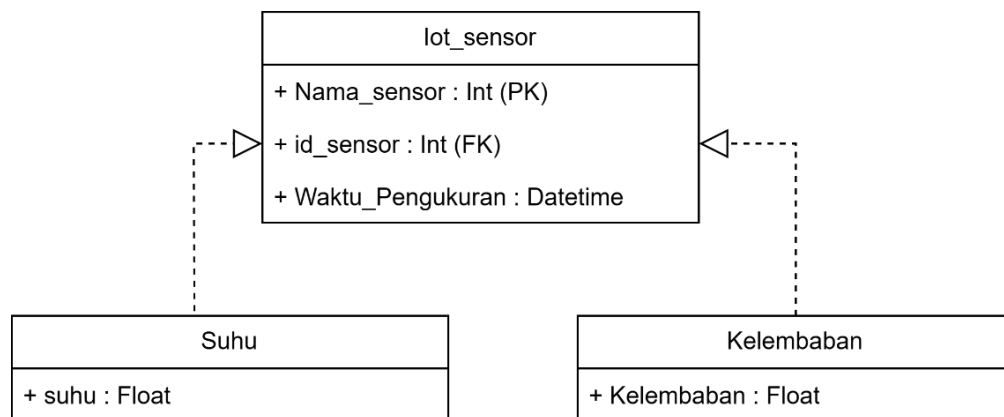


Gambar 3.9 Use case diagram

Use case ini menggambarkan sistem informasi pertanian yang digunakan oleh seorang petani untuk mengakses data yang dibutuhkan dalam kegiatan bertani. Petani dapat mengakses Tampilan Dashboard yang berisi informasi penting seperti Data Suhu Udara dan Data Kelembaban Tanah. Informasi ini dapat membantu petani membuat pilihan terbaik untuk meningkatkan hasil panen.

3.11.2. Class Diagram

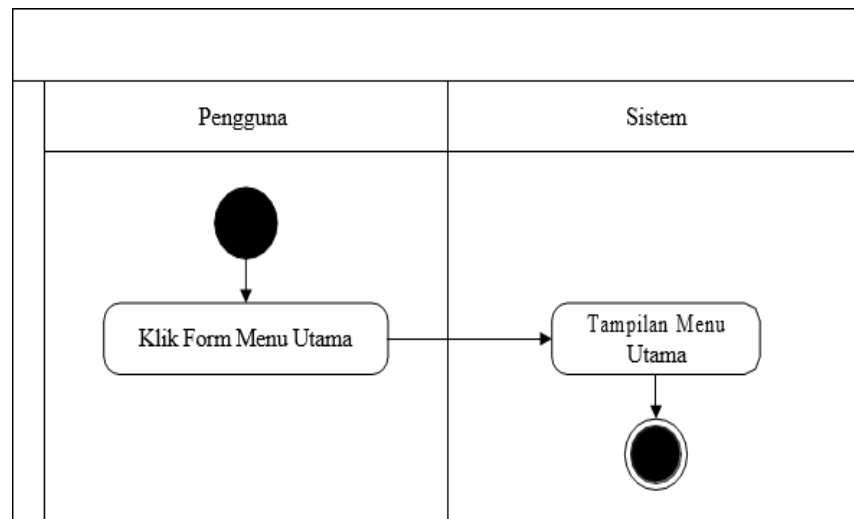
Class Diagram sistem pendeteksi cuaca dan kelembaban tanah dapat dilihat pada gambar berikut ini:



Gambar 3.10. Class Diagram

3.11.3. Activity Diagram

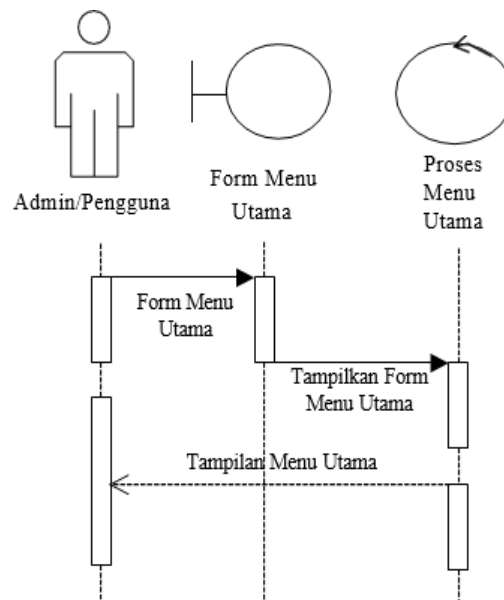
Gambar berikut menampilkan diagram aktivitas sistem deteksi cuaca dan kelembaban tanah:



Gambar 3.11 Activity diagram

3.11.4. Sequence Diagram

Gambar berikut menunjukkan diagram rangkaian sistem deteksi cuaca dan kelembapan tanah :



Gambar 3.12 Sequence Diagram

3.12. Desain Database

Fungsi dari struktur basis data adalah menentukan file basis data yang digunakan, termasuk kolom, tipe data, dan ukuran data. Basis data MySQL digunakan dalam pembangunan sistem ini.

Nama Database: db_iotsensor

Nama Tabel : users

Primary Key : Id_sensor

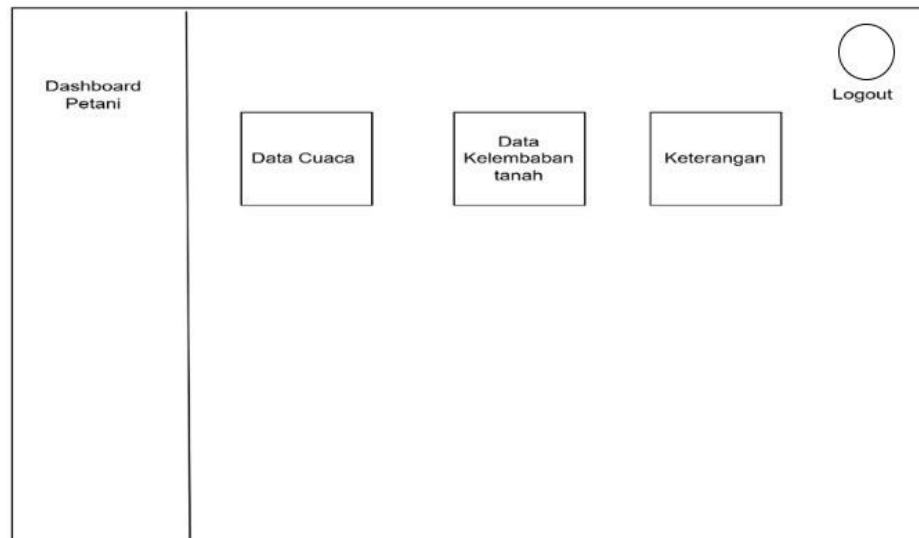
Foreign Key: Nama_sensor

Tabel 3.2. Rancangan Struktur Tabel Pengguna

<i>Field</i>	<i>Data Type</i>	<i>Length</i>	<i>Constraint</i>
Id_sensor	<i>Int</i>	11	<i>Primary key</i>
Nama_sensor	<i>Int</i>	50	<i>Foreign key</i>
Suhu	<i>Float</i>	50	
Kelembapan	<i>Float</i>	50	
Waktu_pengukuran	<i>Date time</i>	50	

3.12. Rancangan Website

Adapun tampilan yang direncanakan pada sistem E-Monitoring pendeteksi cuaca dan kelembaban tanah berbasis website ini ialah yang pertama design perangkat lunak. Ketika alat sudah terdesign maka dapat dilanjutkan proses lanjutan pada website, Dimulai daripada halaman dashboard.



Gambar 3.13 Rancangan Website

Dimana gambar diatas merupakan sebuah halaman Dashboard Website yang akan digunakan oleh petani sawi. Yang dimana sebelah kiri merupakan bagian tampilan Dhasboard, kemudian di bagian (Sari et al., 2023) Tiga kolom data cuaca, data kelembapan tanah, dan deskripsi terletak di bagian tengah. Tombol Log Out situs web adalah yang terakhir di pojok kanan atas.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Penelitian ini bertujuan untuk merancang dan mengimplementasikan system pendeteksi cuaca dan kelembaban tanah untuk meningkatkan hasil panen petani sawi berbasis internet of things. Sistem ini diharapkan dapat mengurangi kebutuhan pengamatan manual dengan menyediakan data yang akurat, seperti kelembaban tanah, cuaca, atau parameter lain yang memengaruhi pertumbuhan tanaman sawi.

4.2. Implementasi Perangkat Keras



Gambar 4. 1 Tanaman Sawi Normal

Alat IoT digunakan untuk mengenali tanaman sawi dalam rangkaian pada gambar 4.1, yang merupakan tanaman sawi biasa dengan suhu 21°C dan kadar air tanah 65%. Halaman web pada gambar 4.2 kemudian akan digunakan untuk mengirimkan hasilnya..



Gambar 4. 2 Website E-Monitoring Sawi Normal

Data dari kebun sawi dengan suhu dan kelembapan tanah normal dijelaskan pada Gambar 4.2. Seperti biasa, situs web menampilkan angka-angka yang muncul saat perangkat IoT diletakkan di tanah. Grafik untuk menentukan jumlah keseluruhan data terkini juga tersedia.



Gambar 4.3 Tanaman sawi Buruk

Rangkain pada gambar 4.3 merupakan tanaman sawi buruk dengan suhu mencapai 40°C dan memiliki kelembaban Tanah mencapai 30%, yang dimana sawi tersebut sudah terdeteksi menggunakan alat IoT. Terlihat jelas kondisi sawi terlihat banyak kerusakan yang ada pada tanaman sawi seperti, daun yang menguning, batang sawi mengecil. Terlihat jelas bahwasannya suhu dan kelembaban tanah sangat mempengaruhi pertumbuhan tanaman sawi. Sama seperti tanaman sawi normal yang dimana hasil datanya akan dikirim melalui website pada gambra 4.4.



Gambar 4.4 Website E-Monitoring Tanaman Sawi Buruk

Data dari kebun sawi, yang menunjukkan kelembaban dan suhu tanah yang buruk, dijelaskan pada Gambar 4.4. Jumlah yang muncul saat gawai IoT diletakkan di tanah yang buruk dapat dilihat dengan mudah di internet. Grafik untuk menentukan jumlah data terbaru juga tersedia.

Kita dapat memperoleh pemahaman yang jelas tentang pengoperasian sistem deteksi cuaca dan kelembaban tanah berbasis Internet of Things ini dengan memahami peran setiap komponen dan bagaimana mereka saling terkait. Semua perangkat input dan output dikoordinasikan oleh modul ESP8266, yang berfungsi sebagai pengontrol pusat, yang menjamin bahwa sistem dapat berfungsi secara otomatis dan membuat koneksi jaringan.

4.3. Implementasi Coding IOT

Implementasi coding untuk sistem pendeteksi cuaca dan kelembaban tanah IOT melibatkan beberapa komponen perangkat keras dan perangkat lunak yang dikendalikan oleh mikrokontroler ESP8266. Berikut adalah penjelasan rinci mengenai implementasi coding yang digunakan dalam sistem ini.

Kode ini diimplementasikan menggunakan Arduino IDE, sebuah platform pengembangan yang memungkinkan penulisan, kompilasi, dan pengunggahan kode ke mikrokontroler ESP8266 dengan mudah. Arduino IDE menyediakan antarmuka pengguna yang intuitif dan berbagai perpustakaan yang memudahkan pengembangan sistem berbasis

mikrokontroler.

4.4. Implementasi Server (API)

Implementasi coding untuk API sistem pendeteksi cuaca dan kelembaban tanah IOT melibatkan beberapa library perangkat lunak yang berfungsi sebagai web server. Berikut adalah penjelasan rinci mengenai implementasi API yang digunakan dalam sistem ini.

Pertama, perpustakaan Express JS yang untuk membuat web server sebagai tempat pengelolaan data yang dikirim oleh microcontroller. Perpustakaan Web Socket digunakan untuk mengirim data secara real time kepada client apabila data diserver ada perubahan menggunakan broadcast.(Sari et al., 2024)

Kode ini diimplementasikan menggunakan IDE Visual Studio Code, sebuah platform text editor yang memungkinkan penulisan kode dengan mudah. IDE Visual Studio Code menyediakan antarmuka pengguna yang intuitif dan berbagai extension yang memudahkan pengembangan sistem berbasis API.(2021)

Berikut adalah kode lengkap untuk sistem ini:

main.js

```
const http = require("http");
const express = require("express");
const dotenv = require("dotenv");
const cors = require("cors");
const rateLimit = require("express-rate-limit");
const bodyParser = require("body-parser");
const controller = require("./controller");
const { PORT } = require("./constant");
const { ioConfig, corsConfig, rateLimiterConfig } = require("./utilities");

dotenv.config();

const app = express();
const server = http.createServer(app);

app.set("trust proxy", ["loopback", "linklocal", "uniquelocal"]);
app.use(express.static("public"));
app.use(cors(corsConfig));
app.use(rateLimit(rateLimiterConfig));
app.use(express.json());
app.use(express.urlencoded({ extended: true }));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

app.post("/login", controller.login);
app.get("/sensor", controller.sensor);
app.post("/sensor", async (request, response) => await controller.sensorTambah(request, response, io));
```

```
server.listen(PORT, async () => {
  console.log(`Server running on: http://localhost:${PORT}`);
});
```

Selanjutnya, coding untuk implementasi web socket pada server agar memungkinkan server melakukan broadcast ke client. Apabila ada data diserver yang beubah, client akan secara otomatis diberitau dan memungkinkan client dapat mengubah data yang ditampilkan tanpa memuat ulang halaman.

main.js

```
const { Server } = require("socket.io");
const websocket = require("./websocket");
const io = new Server(server, ioConfig);
websocket(io);
```

websocket.js

```
const websocket = (io) => {
  io.on("connection", (socket) => {
    console.log("Client connected:", socket.id);

    socket.on("sensorKelembapan", (message) => {
      console.log("Received sensor kelembapan data:", message);
      io.emit("sensor", message);
    });

    socket.on("sensorSuhu", (message) => {
      console.log("Received sensor suhu data:", message);
      io.emit("sensor", message);
    });

    socket.on("disconnect", () => {
      console.log("Client disconnected:", socket.id);
    });
  });
};

module.exports = websocket;
```

Selanjutnya, coding untuk implementasi logic pada server.

```
const bcrypt = require("bcrypt");
const jwt = require("jsonwebtoken");
const schema = require("./schema");
const prisma = require("./prisma");
```

```

const { JWT_SECRET, TOKEN_EXPIRED } = require("../constant");
const { message } = require("../utilities");

const controller = {
  login: async (request, response) => {
    try {
      const validation = schema.login.safeParse(request.body);
      if (!validation.success) return message(response, 400, false, "Gagal validasi request", validation.error.format());
      const { username, password } = validation.data;
      await prisma.$connect();
      const user = await prisma.user.findUnique({
        where: { username },
        select: {
          id: true,
          username: true,
          password: true,
        },
      });
      if (!user) return message(response, 200, false, "Username dan password salah", null);
      const isPasswordMatch = await bcrypt.compare(password, user.password);
      if (!isPasswordMatch) return message(response, 200, false, "Username dan password salah", null);
      const token = jwt.sign({ userId: user.id }, JWT_SECRET, { expiresIn: TOKEN_EXPIRED });
      return message(response, 200, true, "Berhasil login", { token });
    } catch (error) {
      await prisma.$disconnect();
      return message(response, 500, false, error.message, null);
    }
  },
  sensorTambah: async (request, response, io) => {
    try {
      const validation = schema.sensorTambah.safeParse(request.body);
      if (!validation.success) return message(response, 200, false, "Gagal validasi request", validation.error.format());
      const { type, value } = validation.data;
      await prisma.$connect();
      const createdSensor = await prisma.sensor.create({
        data: { type, value },
      });
      if (!createdSensor) return message(response, 200, true, "Gagal menambah data sensor", null);
      if (io) {
        const listData = await prisma.sensor.findMany({
          take: 50,
          orderBy: { updatedAt: "asc" },
          select: { type: true, value: true, createdAt: true },
          where: {
            ...(type && { type }),
          },
        });
      }
    }
  },
};

```



```

    },
  });

    io.emit(createdSensor.type === "KELEMBAPAN_TANAH" ?
"sensorKelembapan" : "sensorSuhu", createdSensor);
    io.emit(createdSensor.type === "KELEMBAPAN_TANAH" ?
"listSensorKelembapan" : "listSensorSuhu", listData);
  }
  await prisma.$disconnect();
  return message(response, 200, true, "Berhasil menambah data sensor", null);
} catch (error) {
  await prisma.$disconnect();
  return message(response, 500, false, error.message, null);
}
},
sensor: async (request, response) => {
  try {
    const validation = schema.sensor.safeParse(request.body);
    if (!validation.success) return message(response, 200, false, "Gagal validasi
request", validation.error.format());
    const { type } = validation.data;
    let data = [];
    if (type) {
      data = await prisma.sensor.findMany({
        take: 50,
        orderBy: { updatedAt: "asc" },
        select: { type: true, value: true, createdAt: true },
        where: { type },
      });
    } else {
      const [kelembapan, suhu] = await Promise.all([
        prisma.sensor.findMany({
          take: 50,
          orderBy: { updatedAt: "asc" },
          select: { type: true, value: true, createdAt: true },
          where: { type: "KELEMBAPAN_TANAH" },
        }),
        prisma.sensor.findMany({
          take: 50,
          orderBy: { updatedAt: "asc" },
          select: { type: true, value: true, createdAt: true },
          where: { type: "SUHU_UDARA" },
        }),
      ]);
      data = [...kelembapan, ...suhu];
    }

    return message(response, 200, true, "Berhasil mengambil data
sensor", !data.length ? [] : data);
  } catch (error) {
    await prisma.$disconnect();
    return message(response, 500, false, error.message, null);
  }
}

```

```

    },
  };

module.exports = controller;

```

Implementasi ini memastikan bahwa sistem dapat mengirim, menerima, dan menulis data dari atau ke pengguna.

4.5. Implementasi Website

Implementasi coding untuk Website sistem pendeteksi cuaca dan kelembaban tanah IOT melibatkan beberapa library perangkat lunak yang berfungsi sebagai struktur website, component, dan subscribe websocket. Berikut adalah penjelasan rinci mengenai implementasi Website yang digunakan dalam sistem ini.

Pertama, perpustakaan React JS yang sebagai kerangka kerja yang digunakan untuk membuat website. Perpustakaan TailwindCSS dan ShadcnUI berfungsi untuk styling stuktur website. Perpustakaan Web Socket digunakan untuk menerima data secara real time dari server apabila data diserver ada perubahan.

Berikut adalah kode lengkap untuk sistem ini:

root.tsx

```

export function Layout({ children }: { children: React.ReactNode }) {
  return (
    <html lang="en">
      <head>
        <meta charSet="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <Meta />
        <Links />
      </head>
      <body>
        {children}
        <ScrollRestoration />
        <Scripts />
      </body>
    </html>
  );
}

```

login.tsx

```

import Cookies from "js-cookie";
import { useState, useEffect } from "react";
import { useNavigate } from "react-router";
import { BACKEND_URL, TOKEN_EXPIRED } from "~/constant";
import type { Route } from "./types/home";
import { cn } from "~/lib/utls";
import { Input } from "~/components/ui/input";
import { Label } from "~/components/ui/label";
import { Button } from "~/components/ui/button";
import { Card, CardContent, CardDescription, CardHeader, CardTitle } from
~/components/ui/card";

export function meta({}: Route.MetaArgs) {
  return [{ title: "Login | Monitoring Sawi Menggunakan Internet Of Things" },
    { name: "Sawiku", content: "Login" }];
}

export default function page() {
  const navigate = useNavigate();

  const [isLoading, setIsLoading] = useState(false);
  const [error, setError] = useState("");
  const [data, setData] = useState({
    username: "",
    password: "",
  });

  useEffect(() => {
    if (typeof window !== "undefined") {
      const accessToken = Cookies.get("token");
      if (accessToken) {
        navigate("/dashboard");
      }
    }
  }, [navigate]);

  const handleChange = (event: React.ChangeEvent<HTMLInputElement>) => {
    setData({ ...data, [event.target.name]: event.target.value });
  };

  const handlerLogin = async (event: React.FormEvent) => {
    event.preventDefault();
    setIsLoading(true);
    setError("");
    try {
      const response = await fetch(`${BACKEND_URL}/login`, {
        method: "POST",

```

```
headers: {  
    "Content-Type": "application/json",  
},  
body: JSON.stringify(data),  
});  
if (!response.ok) throw new Error("Unauthorized");
```

```

const result = await response.json();
if (!result.success) {
  setError(result.message);
  return;
}
const token = result.data.token;
Cookies.set("token", token, { expires: TOKEN_EXPIRED });
navigate("/dashboard");
setError("");
} catch (err: any) {
  setError(err.message);
} finally {
  setIsLoading(false);
}
};

return (
  <div className="flex min-h-svh w-full items-center justify-center p-6 md:p-10 bg-slate-50">
    <div className="w-full max-w-sm">
      <div className={cn("flex flex-col gap-6")}>
        <Card>
          <CardHeader>
            <CardTitle className="text-2xl">Masuk</CardTitle>
            <CardDescription>Masukkan email dan password di bawah ini untuk masuk ke akun Anda</CardDescription>
            {Boolean(error) && <p className="text-sm mt-4 bg-red-200 font-medium text-red-600 p-4 rounded">{error}</p>}
          </CardHeader>
          <CardContent>
            <form
              method="post"
              onSubmit={handlerLogin}
            >
              <div className="flex flex-col gap-6">
                <div className="grid gap-2">
                  <Label htmlFor="username">Username</Label>
                  <Input
                    type="text"
                    id="username"
                    name="username"
                    // placeholder="....."
                    onChange={handleChange}
                    disabled={isLoading}
                    required
                  />
                </div>
                <div className="grid gap-2">

```

```

        <Label htmlFor="password">Password</Label>
        <Input
          id="password"
          type="password"
          name="password"
          onChange={handleChange}
          // placeholder="....."
          disabled={isLoading}
          required
        />
      </div>
      <Button
        type="submit"
        className="w-full bg-blue-600 hover:cursor-pointer
hover:bg-blue-700"
        disabled={isLoading}
      >
        Masuk
      </Button>
    </div>
  </form>
</CardContent>
</Card>
</div>
</div>
</div>
);
}

```

dashboard.tsx

```

import * as React from "react";
import Cookies from "js-cookie";
import { useState, useEffect } from "react";
import { useNavigate } from "react-router";
import { BACKEND_URL } from "~/constant";
import type { Route } from "./types/home";
import { io } from "socket.io-client";
import { Area, AreaChart, CartesianGrid, XAxis } from "recharts";
import { formatTanggal } from "~/lib/utlis";
import { Button } from "~/components/ui/button";
import { Breadcrumb, BreadcrumbItem, BreadcrumbLink, BreadcrumbList } from
"~/components/ui/breadcrumb";
import { type ChartConfig, ChartContainer, ChartTooltip, ChartTooltipContent }
from "~/components/ui/chart";
import { Card, CardContent, CardDescription, CardHeader, CardTitle,
CardFooter } from "~/components/ui/card";

```

```

import { Table, TableBody, TableCell, TableFooter, TableHead, TableHeader,
TableRow } from "~/components/ui/table";

export function meta({}: Route.MetaArgs) {
  return [{ title: "Dashboard | Monitoring Sawi Menggunakan Internet Of
Things" }, { name: "Sawiku", content: "Dashboard" }];
}

interface DataSensorType {
  type: "KELEMBAPAN_TANAH" | "SUHU_UDARA";
  value: number;
  createdAt: Date;
}

const socket = io(BACKEND_URL, {
  transports: ["websocket"],
});

const chartConfigKelembapanTanah = {
  kelembapanTanah: {
    label: "Kelembapan Tanah",
  },
  value: {
    label: "Value",
    color: "hsl(var(--chart-1))",
  },
} satisfies ChartConfig;

const chartConfigSuhuUdata = {
  suhuUdara: {
    label: "SuhurUdara",
  },
  value: {
    label: "Value",
    color: "hsl(var(--chart-2))",
  },
} satisfies ChartConfig;

export default function Page() {
  const navigate = useNavigate();

  const [isLoading, setIsLoading] = useState(false);
  const [error, setError] = useState("");
  const [data, setData] = useState<{
    KELEMBAPAN_TANAH: DataSensorType[];
    SUHU_UDARA: DataSensorType[];
  }>({
    KELEMBAPAN_TANAH: [],

```

```

    SUHU_UDARA: [],
  });
  const [sensorNow, setSensorNow] = useState<{
    KELEMBAPAN_TANAH: DataSensorType | null;
    SUHU_UDARA: DataSensorType | null;
  }>({
    KELEMBAPAN_TANAH: null,
    SUHU_UDARA: null,
  });
  const [listDataSensor, setListDataSensor] = useState<{
    KELEMBAPAN_TANAH: DataSensorType[];
    SUHU_UDARA: DataSensorType[];
  }>({
    KELEMBAPAN_TANAH: [],
    SUHU_UDARA: [],
  });

  useEffect(() => {
    const fetchData = async () => {
      setIsLoading(true);
      if (typeof window === "undefined") return;
      const accessToken = Cookies.get("token");
      if (!accessToken) {
        navigate("/login");
        setIsLoading(false);
        return;
      }
      try {
        const response = await fetch(`${BACKEND_URL}/sensor`, {
          method: "GET",
          headers: {
            "Content-Type": "application/json",
          },
        });
        if (!response.ok) return;
        const result = await response.json();
        const data = result.data;
        const dataKelembapan = data.filter((item: DataSensorType) => item.type === "KELEMBAPAN_TANAH");
        const dataSuhu = data.filter((item: DataSensorType) => item.type === "SUHU_UDARA");
        setData({ KELEMBAPAN_TANAH: dataKelembapan ?? [], SUHU_UDARA: dataSuhu ?? [] });
        setListDataSensor({ KELEMBAPAN_TANAH: dataKelembapan ?? [], SUHU_UDARA: dataSuhu ?? [] });
      } catch (error) {
        console.error("Error fetching data:", error);
        Cookies.remove("token");
      }
    };
    fetchData();
  });

```



```

        navigate("/login");
    } finally {
        setIsLoading(false);
    }
};
fetchData();

if (socket) {
    socket.on("sensorKelembapan", (data) => {
        setSensorNow((prev) => ({ ...prev, KELEMBAPAN_TANAH: data }));
    });

    socket.on("sensorSuhu", (data) => {
        setSensorNow((prev) => ({ ...prev, SUHU_UDARA: data }));
    });

    socket.on("listSensorKelembapan", (data) => {
        setListDataSensor((prev) => ({ ...prev, KELEMBAPAN_TANAH:
data }));
    });

    socket.on("listSensorSuhu", (data) => {
        setListDataSensor((prev) => ({ ...prev, SUHU_UDARA: data }));
    });
}

return () => {
    if (socket) {
        socket.off("sensorKelembapan");
        socket.off("sensorSuhu");
        socket.off("listSensorKelembapan");
        socket.off("listSensorSuhu");
    }
};
}, [navigate]);

const handlerLogout = async (event: React.FormEvent) => {
    event.preventDefault();
    setIsLoading(true);
    setError("");
    try {
        Cookies.remove("token");
        navigate("/login");
    } catch (err: any) {
        setError(err.message);
    } finally {
        setIsLoading(false);
    }
}

```

```

};

if (isLoading) return <p>Loading...</p>;

return (
  <div className="bg-slate-50 h-full pb-16">
    <header className="flex sticky top-0 z-50 w-full items-center border-b bg-white">
      <div className="mx-auto w-[1200px] flex px-4 xl:px-0 h-12 items-center justify-between">
        <Breadcrumb className="block">
          <BreadcrumbList>
            <BreadcrumbItem>
              <BreadcrumbLink
                href="/dashboard"
                className="font-semibold"
              >
                Dashboard Pendeteksi Tanaman Sawi
              </BreadcrumbLink>
            </BreadcrumbItem>
          </BreadcrumbList>
        </Breadcrumb>
        <div className="flex gap-4 items-center">
          <Button
            onClick={handlerLogout}
            variant="destructive"
            className="hover:cursor-pointer"
          >
            Keluar
          </Button>
        </div>
      </div>
    </header>

    <div className="xl:w-[1200px] mx-auto mt-4 grid gap-4 px-4 xl:px-0">
      <div>
        <h1 className="font-semibold text-2xl ">Dashboard Operator</h1>
        <p>Pantau dan analisa data sensor internet of things yang ada
dilapangan.</p>
      </div>

      <div className="grid grid-cols-1 lg:grid-cols-2 gap-4">
        <Card className="@container/card">
          <CardHeader className="relative">
            <CardDescription>Sensor Kelembapan Tanah</CardDescription>
            <CardTitle className="@[250px]/card:text-3xl text-2xl font-semibold tabular-nums">{sensorNow?.KELEMBAPAN_TANAH?.value ??

```

```

data?.KELEMBAPAN_TANAH?.[data.KELEMBAPAN_TANAH.length
1]?.value ?? "0"} %</CardTitle>
</CardHeader>
<CardFooter className="flex-col items-start gap-1 text-sm">
  <div className="line-clamp-1 flex gap-2 font-medium">Terakhir
diupdate:</div>
<div className="text-muted-
foreground">{sensorNow?.KELEMBAPAN_TANAH?.createdAt
formatTanggal(sensorNow?.KELEMBAPAN_TANAH?.createdAt)
data?.KELEMBAPAN_TANAH?.[data.KELEMBAPAN_TANAH.length
1]?.createdAt
formatTanggal(data?.KELEMBAPAN_TANAH?.[data.KELEMBAPAN_TANA
H.length - 1]?.createdAt) : "-"}</div>
</CardFooter>
</Card>
<Card className="@container/card">
  <CardHeader className="relative">
    <CardDescription>Sensor Suhu Udara</CardDescription>
    <CardTitle className="@[250px]/card:text-3xl text-2xl font-
semibold tabular-nums">{sensorNow?.SUHU_UDARA?.value
data?.SUHU_UDARA?.[data.SUHU_UDARA.length - 1]?.value
"0"} °C</CardTitle>
  </CardHeader>
  <CardFooter className="flex-col items-start gap-1 text-sm">
    <div className="line-clamp-1 flex gap-2 font-medium">Terakhir
diupdate:</div>
    <div className="text-muted-
foreground">{sensorNow?.SUHU_UDARA?.createdAt
formatTanggal(sensorNow?.SUHU_UDARA?.createdAt)
data?.SUHU_UDARA?.[data.SUHU_UDARA.length - 1]?.createdAt
formatTanggal(data?.SUHU_UDARA?.[data.SUHU_UDARA.length
1]?.createdAt) : "-"}</div>
  </CardFooter>
</Card>
</div>

<Card className="@container/card">
  <CardHeader className="relative">
    <CardTitle>List Data Suhu Kelembapan Tanah</CardTitle>
    <CardDescription>
      <span className="@[540px]/card:block">Total 50 data
terakhir</span>
    </CardDescription>
  </CardHeader>
  <CardContent className="px-2 pt-4 sm:px-6 sm:pt-6">
    <ChartContainer
      config={chartConfigKelembapanTanah}
      className="aspect-auto h-[250px] w-full"

```

```

>
<AreaChart data={listDataSensor.KELEMBAPAN_TANAH}>
  <defs>
    <linearGradient
      id="fillDesktop"
      x1="0"
      y1="0"
      x2="0"
      y2="1"
    >
      <stop
        offset="5%"
        stopColor="var(--primary)"
        stopOpacity={1.0}
      />
      <stop
        offset="95%"
        stopColor="var(--primary)"
        stopOpacity={0.1}
      />
    </linearGradient>
    <linearGradient
      id="fillMobile"
      x1="0"
      y1="0"
      x2="0"
      y2="1"
    >
      <stop
        offset="5%"
        stopColor="var(--primary)"
        stopOpacity={0.8}
      />
      <stop
        offset="95%"
        stopColor="var(--primary)"
        stopOpacity={0.1}
      />
    </linearGradient>
  </defs>
  <CartesianGrid vertical={false} />
  <XAxis
    dataKey="createdAt"
    tickLine={false}
    axisLine={false}
    tickMargin={8}
    minTickGap={32}
    tickFormatter={(value) => {

```

```

        const date = new Date(value);
        return date.toLocaleDateString("en-US", {
            month: "short",
            day: "numeric",
        });
    });
}
/>
<ChartTooltip
  cursor={false}
  content={
    <ChartTooltipContent
      labelFormatter={(value) => {
        return new Date(value).toLocaleDateString("en-US",
{
            month: "short",
            day: "numeric",
        });
      }}
      indicator="dot"
    />
  }
/>
<Area
  dataKey="value"
  type="natural"
  fill="var(--primary)"
  stroke="var(--primary)"
  stackId="a"
/>
</AreaChart>
</ChartContainer>
</CardContent>
</Card>

<Card className="@container/card">
  <CardHeader className="relative">
    <CardTitle>List Data Sensor Suhu Udara</CardTitle>
    <CardDescription>
      <span className="@[540px]/card:block">Total 50 data
terakhir</span>
    </CardDescription>
  </CardHeader>
  <CardContent className="px-2 pt-4 sm:px-6 sm:pt-6">
    <ChartContainer
      config={chartConfigSuhuUdata}
      className="aspect-auto h-[250px] w-full"
    >
      <AreaChart data={listDataSensor.SUHU_UDARA}>

```

```

<defs>
  <linearGradient
    id="fillDesktop"
    x1="0"
    y1="0"
    x2="0"
    y2="1"
  >
    <stop
      offset="5%"
      stopColor="var(--primary)"
      stopOpacity={1.0}
    />
    <stop
      offset="95%"
      stopColor="var(--primary)"
      stopOpacity={0.1}
    />
  </linearGradient>
  <linearGradient
    id="fillMobile"
    x1="0"
    y1="0"
    x2="0"
    y2="1"
  >
    <stop
      offset="5%"
      stopColor="var(--primary)"
      stopOpacity={0.8}
    />
    <stop
      offset="95%"
      stopColor="var(--primary)"
      stopOpacity={0.1}
    />
  </linearGradient>
</defs>
<CartesianGrid vertical={false} />
<XAxis
  dataKey="createdAt"
  tickLine={false}
  axisLine={false}
  tickMargin={8}
  minTickGap={32}
  tickFormatter={(value) => {
    const date = new Date(value);
    return date.toLocaleDateString("en-US", {

```

```

        month: "short",
        day: "numeric",
    });
    }}
</>
<ChartTooltip
  cursor={false}
  content={
    <ChartTooltipContent
      labelFormatter={(value) => {
        return new Date(value).toLocaleDateString("en-US",
{
        month: "short",
        day: "numeric",
      });
    }}
    indicator="dot"
  />
  }
</>
<Area
  dataKey="value"
  type="natural"
  stackId="a"
</>
</AreaChart>
</ChartContainer>
</CardContent>
</Card>

<Card>
  <CardHeader className="relative">
    <CardTitle>Tabel Data Sensor</CardTitle>
    <CardDescription>
      <span className="@[540px]/card:block">Total 50 data
terakhir</span>
    </CardDescription>
  </CardHeader>
  <CardContent className="px-2 pt-4 sm:px-6 sm:pt-6">
    <Table className="">
      <TableHeader>
        <TableRow>
          <TableHead className="w-1/3">Sensor</TableHead>
          <TableHead className="w-1/3">Nilai</TableHead>
          <TableHead className="w-1/3">Tanggal
Pengambilan</TableHead>
        </TableRow>
      </TableHeader>

```

```

        <TableBody>
            {[...data.KELEMBAPAN_TANAH, ...data.SUHU_UDARA]
                .sort((a, b) => new Date(b.createdAt).getTime() - new
Date(a.createdAt).getTime())
                .map((item, index) => (
                    <TableRow key={index}>
                        <TableCell className="font-medium">{item.type ===
"KELEMBAPAN_TANAH" ? "Kelembapan Tanah" : "Suhu Udara"}</TableCell>
                        <TableCell>
                            {item.value} {item.type ===
"KELEMBAPAN_TANAH" ? "%" : "°C"}
                        </TableCell>

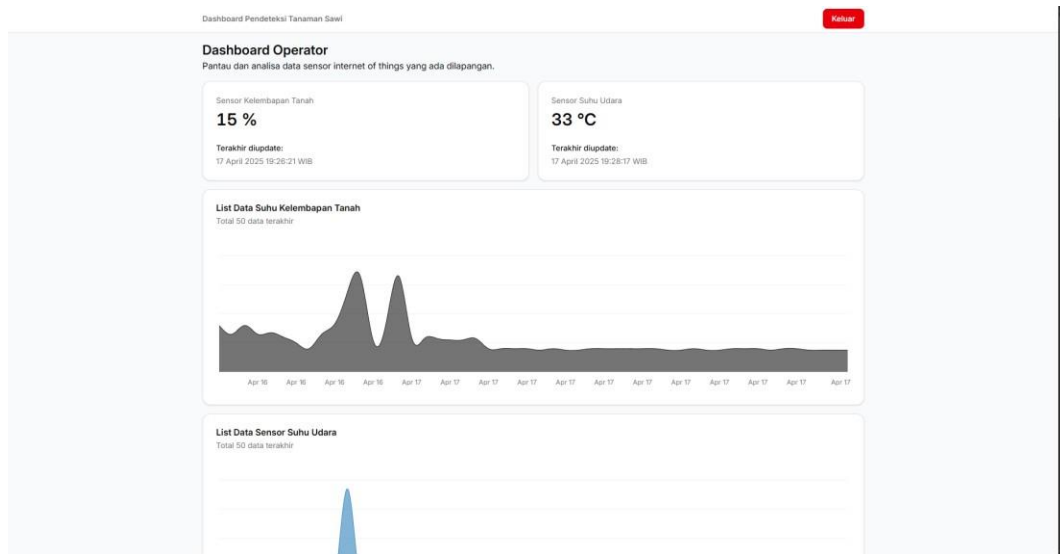
                        <TableCell>{formatTanggal(item.createdAt)}</Table
Cell>

                    </TableRow>
                ))}
        </TableBody>
        <TableFooter>
            <TableRow>
                <TableCell colspan={2}>Total Data</TableCell>
                <TableCell className="text-
right">{[...data.KELEMBAPAN_TANAH, ...data.SUHU_UDARA].length}</Ta
bleCell>

            </TableRow>
        </TableFooter>
    </Table>
</CardContent>
</Card>
</div>
</div>
);
}

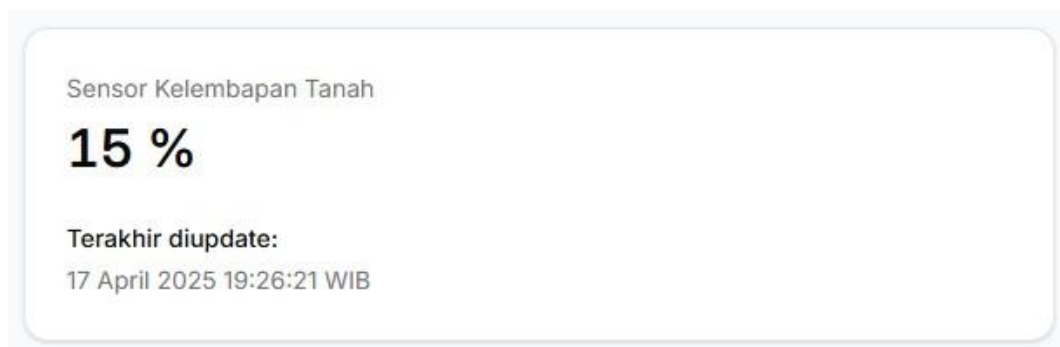
```


Hasil website:



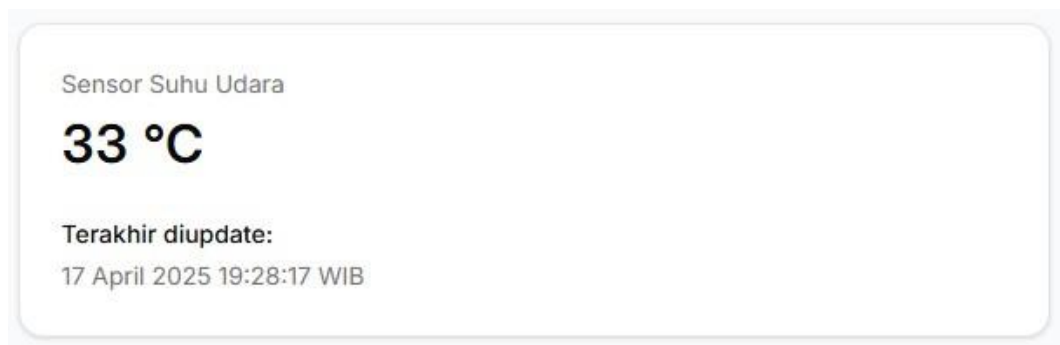
Gambar 4.1. Hasil Website

Indikator Sensor Kelembaban Tanah



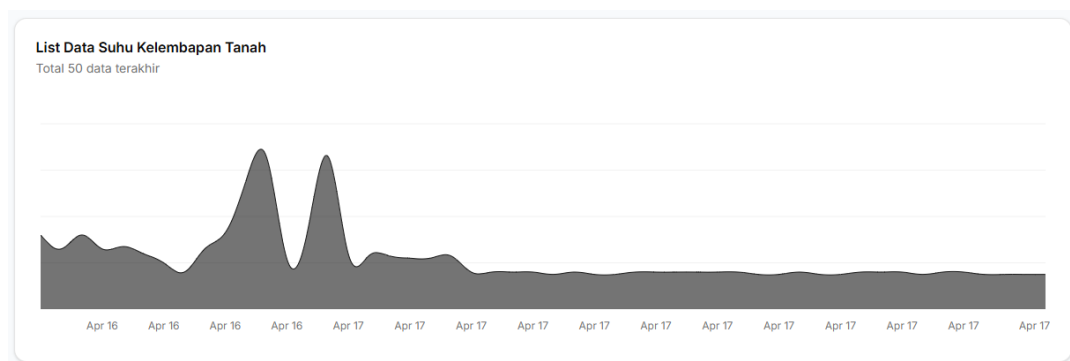
Gambar 4.2. Indikator Sensor Kelembaban Tanah

Indikator Sensor Suhu Udara:



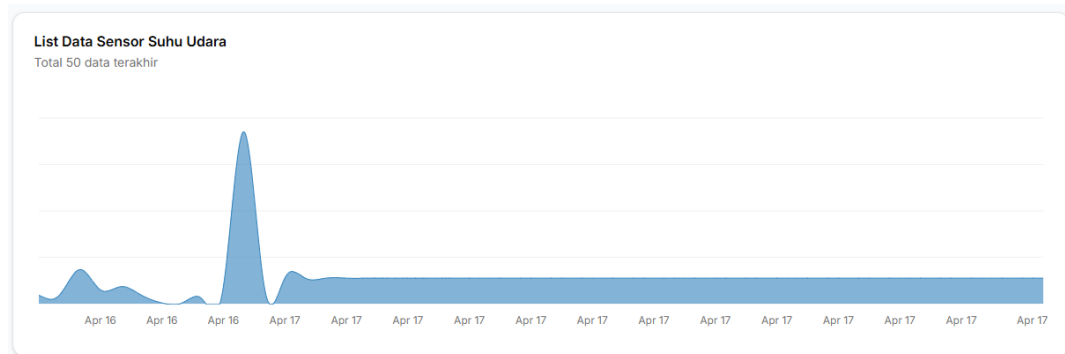
Gambar 4.3. Indikator Senosr Suhu Udara

Grafik perubahan sensor kelembaban tanah dari waktu ke waktu:



Gambar 4.4. Grafik Perubahan Sensor Kelembaban Tanah Dari Waktu ke Waktu

Grafik perubahan sensor kelembaban tanah dari waktu ke waktu:



Gambar 4.5. Grafik perubahan Sensor Kelembaban Tanah Dari waktu ke waktu

4.6. Implementasi Pengujian

Untuk memastikan sistem deteksi cuaca dan kelembapan tanah IOT berfungsi sebagaimana mestinya, pengujian implementasi dilakukan. Untuk mengurangi kemungkinan kesalahan atau penyimpangan yang tidak diinginkan, pengujian ini penting untuk menilai kesesuaian antara strategi awal dan hasil yang diperoleh.

Pengujian dilakukan dalam beberapa tahap, yaitu pengujian perangkat keras, perangkat lunak, dan integrasi sistem. Setiap tahap pengujian memiliki fokus tertentu untuk memastikan semua komponen dan fungsi sistem bekerja dengan optimal.

1. **Pengujian Perangkat Keras:** Pengujian dilakukan untuk memastikan bahwa setiap perangkat keras, termasuk sensor suhu udara, sensor kelembapan tanah, dan mikrokontroler, beroperasi dengan benar dan sesuai dengan spesifikasi. Baik sensor kelembapan tanah maupun sensor suhu udara berhasil menangkap dan merekam kelembapan tanah dan suhu udara, masing-masing dengan tingkat keberhasilan 100%.

2. Pengujian Perangkat Lunak: Sistem diuji untuk memastikan bahwa koneksi WiFi dan integrasi dengan server berjalan lancar. Kecepatan Notifikasi Sistem dapat mengirimkan data ke server dalam waktu 1-2 detik.
3. Pengujian Integrasi Sistem: Sistem diuji secara keseluruhan untuk memastikan bahwa semua komponen dapat berfungsi secara terintegrasi. Verifikasi bahwa sistem dapat merekam/mengambil data dengan akurasi tinggi. Setiap skenario penggunaan diuji secara menyeluruh.

Sistem ini beroperasi sebagaimana mestinya, menurut hasil pengujian. Diharapkan bahwa sistem pendeteksi cuaca dan kelembapan tanah IoT ini dapat berfungsi secara merata dan sesuai dengan tujuan yang ditetapkan melalui pengujian yang menyeluruh dan terorganisasi. Oleh karena itu, pengujian ini sangat penting untuk menjamin bahwa sistem tidak hanya dapat diandalkan oleh pengguna dalam skenario aktual, tetapi juga efektif dan efisien.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berikut ini adalah beberapa kesimpulan penting yang dapat diambil dari penelitian tentang "Sistem Deteksi Cuaca dan Kelembaban Tanah untuk Meningkatkan Hasil Panen Petani Sawi Hijau Berbasis Internet of Things (IoT)":

1. Sistem pendeteksi berhasil mengambil/merekam data dari sensor-sensor yang tersedia dengan tingkat akurasi tinggi. Penggunaan sensor-sensor memberikan tingkat keberhasilan otentikasi sebesar 100%.
2. Sistem mampu mengirimkan data ke server dalam waktu 1-2 detik dan Website sebagai client mendapatkan data baru dari server secara realtime. Ini menunjukkan bahwa sistem memberikan informasi secara cepat dan efisien kepada pengguna.
3. Komponen perangkat keras seperti microcontroller, sensor kelembaban tanah, dan sensor suhu udara berfungsi dengan baik sesuai spesifikasi. Sensor-sensor yang dapat mengambil/merekam data dengan akurasi tinggi.

5.2. Saran

1. Sistem Pendukung Keputusan (SPK) atau teknik pembelajaran mesin dapat digunakan untuk mengembangkan sistem pengambilan keputusan, yang memungkinkan pengguna memperoleh informasi berharga untuk dijadikan dasar pengambilan keputusan selanjutnya.
2. Dapat bermanfaat untuk diintegrasikan dengan instrumen pertanian lainnya, seperti sistem irigasi otomatis berbasis Internet of Things (IoT). Saat kadar air tanah mendekati nol (kondisi tanah kering), perangkat ini dapat membantu pelanggan menyiram tanaman secara otomatis.

Daftar Pustaka

- Alfina, O., & Harahap, F. (n.d.). *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi PEMODELAN UML SISTEM PENDUKUNG KEPUTUSAN DALAM PENENTUAN KELAS SISWA SISWA TUNAGRAHITA* 1. <https://doi.org/10.46880/jmika.Vol3No2.pp143-150>
- Deliana, R., & Periyadi. (2020). *Sistem Monitoring Pengukur PH dan Kelembaban Tanah Untuk Tanaman Strawberry dengan Sensor DHT22 serta Penyiram Otomatis Berbasis IoT*.
- Gerardhy Lainsamputty, B. (2019). Perancangan Notifikasi Pesan Disposisi Pada Sistem Pengelolaan Surat Dinas Menggunakan Layanan API Media Social (Studi Kasus: Dinas Pertanian Kota Salatiga). *AITI: Jurnal Teknologi Informasi*, 16(Agustus), 165–186.
- Al-Khowarizmi, A.-K., & Hutagalung Sari Fatma. (2021). Implementasi E-Monitoring Aktivitas Siswa Pada SMKN 5 dan SMKS 2 Medan Putri Berbasis Web. *Ihsan: Jurnal Pengabdian Masyarakat*. <https://doi.org/10.30596/ihsan.v3i2.7571>
- Luo, T., Hao, H., Du, W., Wang, Y., & Yin, H. (2011). Attacks on WebView in the Android system. *ACM International Conference Proceeding Series*, 343–352. <https://doi.org/10.1145/2076732.2076781>
- Nurdin, M., Muslih, M., Rizki, N., Maula Utami, E., Al-ayyubi, S., Teknik Informatika, J., Studi Sistem Informasi, P., Dwiwarna, S., Nusa Putra, U., Palabuhanratu Km, J., Sukaharja, D., Warungkiara, K., Sukabumi, K., & Raya Cibolang Kaler No, J. (2019). PERENCANAAN PRODUKSI E-MONITORING DENGAN MODEL ARSITEKTUR TERINTEGRASI. In *Jurnal Rekayasa Teknologi Nusa Putra* (Vol. 5, Issue 2).
- Nurul Hidayati Lusita Dewi, M. F. R. S. Z. (n.d.). *PROTOTYPE SMART HOME DENGAN MODUL NODEMCU ESP8266 BERBASIS INTERNET OF THINGS (IOT)*.
- Putra, R. B. D., Budi, E. S., & Kadafi, A. R. (2021). Perancangan WebView Template Pada Android Studio Arctic Fox Studi Kasus: Dagangrumah.com. *JURIKOM (Jurnal Riset Komputer)*, 8(6), 374. <https://doi.org/10.30865/jurikom.v8i6.3685>
- Roihan, A., Mardiansyah, A., Pratama, A., & Pangestu, A. A. (2021). SIMULASI PENDETEKSI KELEMBABAN PADA TANAH MENGGUNAKAN SENSOR DHT22 DENGAN PROTEUS. *METHODIKA: Jurnal Teknik Informatika Dan Sistem Informasi*, 7(1), 25–30. <https://doi.org/10.46880/mtk.v7i1.260>
- Saputro, I. A., Suseno, J. E., Catur, D., & Widodo, E. (2017). Rancang bangun sistem pengaturan kelembaban tanah secara real time menggunakan mikrokontroler dan diakses di web. In *Youngster Physics Journal* (Vol. 6, Issue 1).
- Sari, I. P., Al-Khowarizmi, A.-K., Apdilah, D., Manurung, A. A., & Basri, M. (2023). Perancangan Sistem Pengaturan Suhu Ruangan Otomatis Berbasis Hardware Mikrokontroler Berbasis AVR. *Sudo Jurnal Teknik Informatika*, 2(3), 131–142. <https://doi.org/10.56211/sudo.v2i3.327>
- Sari, I. P., Novita, A., Al-Khowarizmi, A.-K., Ramadhani, F., & Satria, A. (2024). Pemanfaatan Internet of Things (IoT) pada Bidang Pertanian Menggunakan Arduino UnoR3. *Blend Sains Jurnal Teknik*, 2(4), 337–343. <https://doi.org/10.56211/blendsains.v2i4.505>
- Supriyanto, T. (2020). *SISTEM PENGENDALI SUHU DAN KELEMBAPAN TANAH BILIK TANAMAN SELADA BERBASIS IOT MENGGUNAKAN APLIKASI WHATSAPP*.

- Tukino. (2018). Perancangan Sistem Informasi Pelaporan Gangguan Dan Restitusi Pelanggan Internet Corporate Berbasis Web (Studi Kasus Di PT. Indosat Mega Media West Regional). *Jurnal Ilmiah Informatika (JIF)*.
<http://ejournal.upbatam.ac.id/index.php/jif>
- Verdi, V. V., & Sarwoko, I. M. (n.d.). *MENGGUNAKAN SMS GATEWAY BERBASIS ARDUINO DESIGN AND IMPLEMENTATION OF SOIL MOISTURE MEASUREMENT SYSTEM USING SMS GATEWAY BASED ON ARDUINO*.