

**LAPORAN PROYEK AKHIR MATA KULIAH
MIKROPROSESOR**

**PENYIRAMAN TANAMAN OTOMATIS BERBASIS
MIKROKONTROLER**






DOSEN: Dr.Satria Gunawan Zain S.Pd,MT.

Disusun oleh:

| | |
|------------------------------------|-----------------------|
| A. Ahmad Nadil | (230210501050) |
| Rezky Nur Intan | (230210500020) |
| Melinda Keizyah Mandaka | (230210500021) |
| Adniel Pascal Ancelo Jonas | (230210501072) |
| Yudhistira Wikrama Desember | (230210501066) |

**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
FAKULTAS TEKNIK
UNIVERSITAS NEGERI MAKASSAR
2024**

LEMBAR KONTRIBUSI TIM

| Nama | NIM | FOTO | PERAN | KONTRIBUSI | TTD |
|----------------------------|--------------|-------------------------------------------------------------------------------------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Melinda Keizyah Mandaka | 230210500021 |  | Project Manager | <ul style="list-style-type: none"> - Bertanggung jawab atas semua proses pembuatan proyek ini. - Membuat tahapan rancangan awal proyek Sistem Penyiraman | |
| Rezky Nur Intan | 230210500020 |  | Hardware Engineer | <ul style="list-style-type: none"> - Menghubungkan Sensor ke Arduino - Menghubungkan ESP32 ke Arduino | |
| A. Ahmad Nadil | 230210501050 |  | Software Engineer | <ul style="list-style-type: none"> - Menghubungkan Arduino ke ESP32 (I2C) - Membuat Logika dari Rangkaian - Mengirim data ke Database | |
| Adniel Pascal Ancelo Jonas | 230210501072 |  | Tester | <ul style="list-style-type: none"> - Menguji Rangkaian tiap dari Komponen | |

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan laporan praktikum yang berjudul "**Penyiraman Tanaman Otomatis Berbasis Mikrokontroler**" ini dengan baik. Laporan ini disusun sebagai bagian dari kegiatan praktikum pada Program Studi Teknik Komputer, Fakultas Teknik, Universitas Negeri Makassar.

Kami menyampaikan rasa terima kasih yang sebesar-besarnya kepada berbagai pihak yang telah memberikan dukungan selama penyusunan laporan ini. Ucapan terima kasih khusus kami tujukan kepada **Dr. Satria Gunawan Zain, S.Pd., M.T.**, selaku dosen pembimbing, atas arahan, bimbingan, dan motivasi yang sangat berharga. Tak lupa, kami juga berterima kasih kepada rekan-rekan satu tim atas kerja sama yang solid serta semua pihak yang turut memberikan bantuan, baik secara langsung maupun tidak langsung.

Laporan ini bertujuan untuk menjelaskan proses perancangan dan implementasi sistem penyiraman tanaman otomatis berbasis mikrokontroler, sekaligus mendokumentasikan hasil praktikum. Harapannya, laporan ini dapat menjadi referensi yang bermanfaat bagi pembaca yang ingin memahami atau mengembangkan sistem serupa.

Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan demi penyempurnaan di masa depan. Semoga laporan ini dapat memberikan manfaat bagi semua pihak, khususnya dalam bidang pengembangan teknologi berbasis mikrokontroler.

Makassar, 12 Desember 2024

Penulis

ABSTRAK

Sistem penyiraman tanaman otomatis berbasis mikrokontroler dikembangkan untuk mengatasi kebutuhan efisiensi dalam pengelolaan irigasi tanaman. Proyek ini dirancang untuk mendeteksi kelembapan tanah, kelembapan udara, dan intensitas cahaya menggunakan sensor, serta mengendalikan pompa air secara otomatis berdasarkan data yang diperoleh. Tujuannya adalah untuk menciptakan sistem irigasi yang efisien dan terintegrasi dengan platform IoT melalui pengiriman data ke server.

Metode yang digunakan melibatkan penggabungan sensor (DHT11, LDR, dan sensor kelembapan tanah), kontrol pompa menggunakan relay, dan komunikasi antar perangkat melalui protokol I2C. Data lingkungan yang diperoleh dari Arduino dikirim ke ESP32, yang kemudian meneruskannya ke database melalui koneksi Wi-Fi. Pengujian dilakukan menggunakan metode **Blackbox Testing** untuk memastikan fungsionalitas sistem berjalan sesuai spesifikasi.

Hasil menunjukkan bahwa sistem berhasil mendeteksi kondisi lingkungan dengan akurasi tinggi dan mengaktifkan pompa air sesuai dengan parameter yang ditentukan. Data yang dikirimkan ke server melalui ESP32 dapat disimpan dengan baik, dan semua modul berfungsi secara terintegrasi.

Kesimpulannya, proyek ini berhasil mengotomasi sistem penyiraman dengan efisiensi tinggi. Pengembangan lebih lanjut disarankan, seperti penambahan antarmuka pengguna dan pengoptimalan sensor, untuk meningkatkan keandalan dan fleksibilitas sistem di berbagai kondisi lapangan.

DAFTAR ISI

| | |
|-----------------------------------|----|
| LEMBAR KONTRIBUSI TIM | 2 |
| KATA PENGANTAR | 3 |
| DAFTAR ISI..... | 5 |
| BAB I PENDAHULUAN | 6 |
| A. Latar Belakang | 6 |
| B. Rumusan Masalah | 6 |
| C. Tujuan Proyek | 7 |
| D. Manfaat Proyek | 7 |
| E. Batasan Masalah..... | 7 |
| BAB II TINJAUAN PUSTAKA..... | 8 |
| A. Landasan Teori..... | 8 |
| B. Penelitian Terdahulu..... | 9 |
| C. Kerangka Teoritis | 10 |
| BAB III METODELOGI | 12 |
| A. Desain Sistem | 12 |
| B. Alat dan Bahan | 13 |
| C. Langkah Kerja | 13 |
| D. Pengujian Sistem | 16 |
| BAB IV HASIL DAN PEMBAHASAN | 17 |
| A. Hasil Implementasi..... | 17 |
| B. Analisis Hasil | 20 |
| BAB V KESIMPULAN DAN SARAN..... | 21 |
| A. Kesimpulan | 21 |
| B. Saran | 21 |
| DAFTAR PUSTAKA | 22 |
| LAMPIRAN..... | 23 |

BAB I PENDAHULUAN

A. Latar Belakang

Dalam beberapa dekade terakhir, pertanian telah mengalami perubahan signifikan seiring dengan perkembangan teknologi. Salah satu tantangan utama yang dihadapi oleh petani adalah pengelolaan sumber daya air yang efisien. Dengan meningkatnya populasi global, kebutuhan akan pangan juga meningkat, sehingga memerlukan metode pertanian yang lebih produktif dan berkelanjutan. Menurut FAO (Food and Agriculture Organization), sekitar 70% dari total penggunaan air di dunia digunakan untuk irigasi pertanian, dan pengelolaan air yang buruk dapat menyebabkan pemborosan sumber daya serta mengurangi hasil panen (FAO, 2020).

Penyiraman tanaman yang tidak teratur sering kali menjadi penyebab utama stres pada tanaman, yang berujung pada penurunan kualitas dan kuantitas hasil panen. Dalam konteks ini, teknologi otomasi menjadi solusi yang menjanjikan untuk meningkatkan efisiensi penyiraman tanaman. Dengan menggunakan sistem penyiraman otomatis, petani dapat memantau kelembapan tanah secara real-time dan melakukan penyiraman sesuai kebutuhan tanaman. Hal ini tidak hanya menghemat air tetapi juga mengurangi waktu dan tenaga yang diperlukan untuk merawat tanaman.

Mikrokontroler telah menjadi komponen kunci dalam pengembangan sistem otomasi pertanian. Dengan kemampuannya untuk mengontrol berbagai sensor dan aktuator, mikrokontroler dapat digunakan untuk merancang sistem penyiraman otomatis yang responsif dan efisien. Selain itu, penggunaan mikrokontroler dalam pertanian pintar memungkinkan integrasi dengan teknologi Internet of Things (IoT), di mana data dapat dikumpulkan dan dianalisis untuk meningkatkan praktik pertanian (Gonzalez et al., 2018).

Proyek ini bertujuan untuk mengembangkan sistem penyiraman tanaman otomatis berbasis mikrokontroler yang dapat memantau kelembapan tanah dan melakukan penyiraman secara otomatis. Dengan demikian, diharapkan proyek ini dapat memberikan kontribusi positif terhadap efisiensi penggunaan air dalam pertanian serta meningkatkan hasil panen.

B. Rumusan Masalah

1. Bagaimana merancang sistem penyiraman tanaman otomatis berbasis mikrokontroler yang dapat bekerja secara efisien dan andal?
2. Bagaimana memanfaatkan sensor kelembapan tanah untuk mendeteksi kebutuhan air tanaman secara real-time?
3. Bagaimana mengintegrasikan perangkat keras dan perangkat lunak untuk menghasilkan sistem yang terukur dan responsif terhadap kondisi lingkungan?
4. Bagaimana mengevaluasi kinerja sistem penyiraman otomatis berdasarkan parameter yang telah ditentukan, seperti tingkat kelembapan tanah yang optimal?

C. Tujuan Proyek

1. Membantu mahasiswa mengenali komponen utama dalam sistem penyiraman otomatis berbasis mikrokontroler.
2. Melatih mahasiswa memprogram mikrokontroler untuk mengontrol proses penyiraman tanaman berdasarkan data kelembaban tanah.
3. Mengintegrasikan sensor kelembaban tanah untuk mendeteksi kebutuhan air tanaman secara otomatis.
4. Menghubungkan perangkat keras (sensor, mikrokontroler, dan aktuator) dengan perangkat lunak untuk membangun sistem yang efisien dan andal.
5. Mengevaluasi kinerja sistem penyiraman otomatis berdasarkan parameter yang telah ditentukan, seperti tingkat kelembaban optimal dan efisiensi penggunaan air.

D. Manfaat Proyek

1. Pengembangan Pemahaman Teknologi: Membantu mahasiswa dan pengguna memahami komponen utama dalam sistem penyiraman otomatis berbasis mikrokontroler, yang dapat menjadi dasar untuk pengembangan lebih lanjut.
2. Penguasaan Pemrograman Mikrokontroler: Meningkatkan kemampuan teknis mahasiswa dalam memprogram mikrokontroler, khususnya dalam bidang otomasi pertanian.
3. Peningkatan Efisiensi Pertanian: Menggunakan sensor kelembaban tanah untuk mendeteksi kebutuhan air, sehingga penyiraman dapat dilakukan secara tepat dan efisien.
4. Inovasi Integrasi Sistem: Mendorong inovasi melalui integrasi perangkat keras dan perangkat lunak, menghasilkan sistem yang responsif dan hemat energi.
5. Evaluasi Sistem Berbasis Data: Memberikan wawasan tentang cara mengevaluasi kinerja sistem otomatis, yang dapat dijadikan panduan untuk mengoptimalkan teknologi otomasi serupa di masa depan.

E. Batasan Masalah

1. Sistem dirancang untuk skala kecil, seperti kebun rumah tangga atau taman.
2. Mikrokontroler yang digunakan adalah Arduino, karena fleksibilitas dan kemudahannya dalam pemrograman.
3. Sistem hanya menggunakan sensor kelembaban tanah untuk mendeteksi tingkat kelembaban media tanam sebagai parameter utama.
4. Sistem tidak mempertimbangkan faktor lingkungan eksternal lain seperti curah hujan atau suhu udara.
5. Pengujian dilakukan dalam lingkungan terkendali untuk mengevaluasi performa sistem secara spesifik terhadap parameter yang telah ditentukan.

BAB II TINJAUAN PUSTAKA

A. Landasan Teori

1. Mikrokontroler

Mikrokontroler adalah sistem komputer mini yang dirancang untuk mengontrol perangkat lain, dan sering digunakan dalam aplikasi otomasi. Mikrokontroler terdiri dari unit pemrosesan pusat (CPU), memori (RAM dan ROM), serta perangkat input/output (I/O) dalam satu chip. Dalam konteks penyiraman tanaman otomatis, mikrokontroler seperti Arduino menjadi populer karena kemudahan penggunaannya dan fleksibilitasnya dalam pemrograman. Arduino dapat diprogram menggunakan bahasa pemrograman C/C++, yang memungkinkan pengembang untuk menulis kode untuk membaca data dari sensor dan mengendalikan aktuator seperti pompa air (Banerjee, Roy, & Dasgupta, 2020). Mikrokontroler juga dilengkapi dengan berbagai pin I/O yang dapat digunakan untuk menghubungkan sensor, aktuator, dan perangkat lain. Dengan kemampuan ini, mikrokontroler dapat digunakan untuk merancang sistem penyiraman otomatis yang responsif terhadap kondisi tanah dan kebutuhan tanaman.

2. Sensor Kelembapan Tanah

Sensor kelembapan tanah berfungsi untuk mengukur kadar air dalam tanah, yang merupakan faktor kritis dalam penyiraman tanaman. Ada dua jenis utama sensor kelembapan tanah: sensor resistif dan sensor kapasitif. Sensor Resistif mengukur kelembapan tanah berdasarkan resistivitas listrik. Ketika tanah kering, resistivitas tinggi; sebaliknya, ketika tanah lembab, resistivitas rendah. Sensor ini cenderung lebih murah tetapi memiliki umur pakai yang lebih pendek karena korosi pada elektroda (Kumar & Singh, 2019). Sensor Kapasitif mengukur kelembapan dengan cara mengukur perubahan kapasitansi antara dua elektroda yang terbenam di dalam tanah. Sensor ini lebih tahan lama dan tidak terpengaruh oleh korosi, sehingga lebih cocok untuk penggunaan jangka panjang (Kumar & Singh, 2019). Data dari sensor kelembapan tanah sangat penting untuk menentukan kapan penyiraman harus dilakukan, sehingga membantu dalam pengelolaan air yang efisien.

3. Sistem Otomasi Pertanian

Sistem otomasi pertanian mengacu pada penggunaan teknologi untuk meningkatkan efisiensi dan produktivitas pertanian. Dengan memanfaatkan mikrokontroler dan sensor, sistem ini memungkinkan pemantauan dan pengendalian proses pertanian secara otomatis. Implementasi sistem otomasi dapat mencakup berbagai aspek pertanian seperti penyiraman otomatis, pengendalian suhu, pencahayaan, dan nutrisi tanaman. Dengan menggunakan teknologi ini, petani dapat mengurangi ketergantungan pada tenaga kerja manual serta meningkatkan hasil panen melalui pengelolaan sumber daya yang lebih baik (Gonzalez, Garcia, & Rodriguez, 2018). Sistem otomasi memungkinkan petani

untuk mengontrol berbagai parameter secara real-time, meningkatkan efisiensi dalam penggunaan sumber daya dan hasil panen.

4. Internet of Things (IoT) dalam Pertanian

Internet of Things (IoT) adalah konsep di mana perangkat fisik terhubung ke internet dan dapat saling berkomunikasi serta bertukar data. Dalam konteks pertanian, IoT memungkinkan petani untuk memantau kondisi tanaman dan lingkungan secara real-time melalui aplikasi mobile atau web. Dengan integrasi IoT dalam sistem penyiraman otomatis, data dari sensor kelembapan tanah dapat dikirim ke cloud untuk analisis lebih lanjut. Petani dapat menerima notifikasi tentang kondisi kelembapan tanah dan melakukan tindakan yang diperlukan dari jarak jauh (Kumar, Singh, & Gupta, 2020). Hal ini tidak hanya meningkatkan efisiensi tetapi juga memberikan wawasan berharga bagi petani dalam pengambilan keputusan. Penerapan IoT dalam pertanian juga memberikan efisiensi dalam penggunaan air dan memperbaiki pengelolaan tanaman. Sistem yang menghubungkan berbagai perangkat ini memungkinkan analisis data yang lebih mendalam dan pemantauan yang lebih akurat, membantu petani menghemat biaya dan memaksimalkan hasil pertanian.

5. Manfaat Penyiraman Otomatis

Penyiraman otomatis menawarkan berbagai manfaat signifikan bagi praktik pertanian modern. Salah satu manfaat utama adalah efisiensi penggunaan air, di mana sistem ini memastikan bahwa tanaman hanya disiram saat diperlukan berdasarkan data kelembapan tanah yang akurat. Hal ini tidak hanya mengurangi pemborosan air, tetapi juga membantu menjaga keseimbangan kelembapan yang optimal untuk pertumbuhan tanaman. Selain itu, penyiraman otomatis dapat mengurangi stres pada tanaman dengan memberikan pasokan air yang tepat waktu, sehingga meningkatkan kesehatan dan produktivitas tanaman. Dengan mengotomatiskan proses penyiraman, petani juga dapat menghemat waktu dan tenaga, memungkinkan mereka untuk fokus pada aspek lain dari pengelolaan pertanian. Di samping itu, sistem ini mendukung pengambilan keputusan yang lebih baik melalui pemantauan real-time, memberikan petani wawasan yang lebih dalam mengenai kondisi tanaman dan kebutuhan air mereka. Dengan demikian, penerapan sistem penyiraman otomatis tidak hanya meningkatkan efisiensi dan hasil panen, tetapi juga berkontribusi pada praktik pertanian yang lebih berkelanjutan dan ramah lingkungan.

B. Penelitian Terdahulu

Penelitian terkait sistem penyiraman otomatis berbasis mikrokontroler telah dilakukan oleh berbagai peneliti untuk mengeksplorasi efisiensi dan efektivitas teknologi dalam pertanian. Salah satu penelitian penting adalah yang dilakukan oleh Banerjee et al. (2020), yang mengembangkan sistem irigasi pintar menggunakan Arduino. Sistem ini dirancang untuk secara otomatis mengatur penyiraman tanaman berdasarkan data kelembapan tanah yang diperoleh dari sensor. Hasil penelitian menunjukkan bahwa dengan menggunakan sensor kelembapan tanah, efisiensi penggunaan air dapat meningkat hingga 30%, dan

hasil panen juga mengalami peningkatan signifikan. Penelitian ini menekankan pentingnya pemantauan kelembapan tanah secara real-time untuk mengoptimalkan proses penyiraman.

Hussain et al. (2019) melakukan penelitian yang berfokus pada otomasi pertanian dengan menggunakan mikrokontroler dan sensor untuk memantau kondisi tanaman. Dalam studi ini, mereka mengembangkan sistem yang tidak hanya melakukan penyiraman otomatis tetapi juga memantau parameter lingkungan lainnya, seperti suhu dan kelembapan udara. Hasil dari penelitian ini menunjukkan bahwa sistem otomasi dapat mengurangi ketergantungan pada tenaga kerja manual, serta meningkatkan produktivitas pertanian secara keseluruhan.

Gonzalez et al. (2018) melakukan kajian mengenai aplikasi Internet of Things (IoT) dalam pertanian, di mana mereka mengintegrasikan perangkat sensor dengan platform cloud untuk memantau kondisi tanaman secara real-time. Penelitian ini menunjukkan bahwa dengan menggunakan IoT, petani dapat menerima notifikasi tentang kondisi kelembapan tanah dan melakukan tindakan yang diperlukan dari jarak jauh, sehingga meningkatkan efisiensi operasional dan pengelolaan sumber daya air.

Dalam konteks sensor kelembapan tanah, Kumar & Singh (2019) memberikan tinjauan mendalam mengenai berbagai jenis sensor yang digunakan dalam pertanian. Mereka membandingkan sensor resistif dan kapasitif, serta membahas kelebihan dan kekurangan masing-masing. Penelitian ini memberikan wawasan berharga tentang bagaimana memilih sensor yang tepat untuk aplikasi tertentu dalam sistem penyiraman otomatis.

C. Kerangka Teoritis

Kerangka teoritis dari proyek penyiraman tanaman otomatis berbasis mikrokontroler ini mencakup beberapa konsep dasar yang saling terkait, yang memberikan landasan bagi pengembangan sistem:

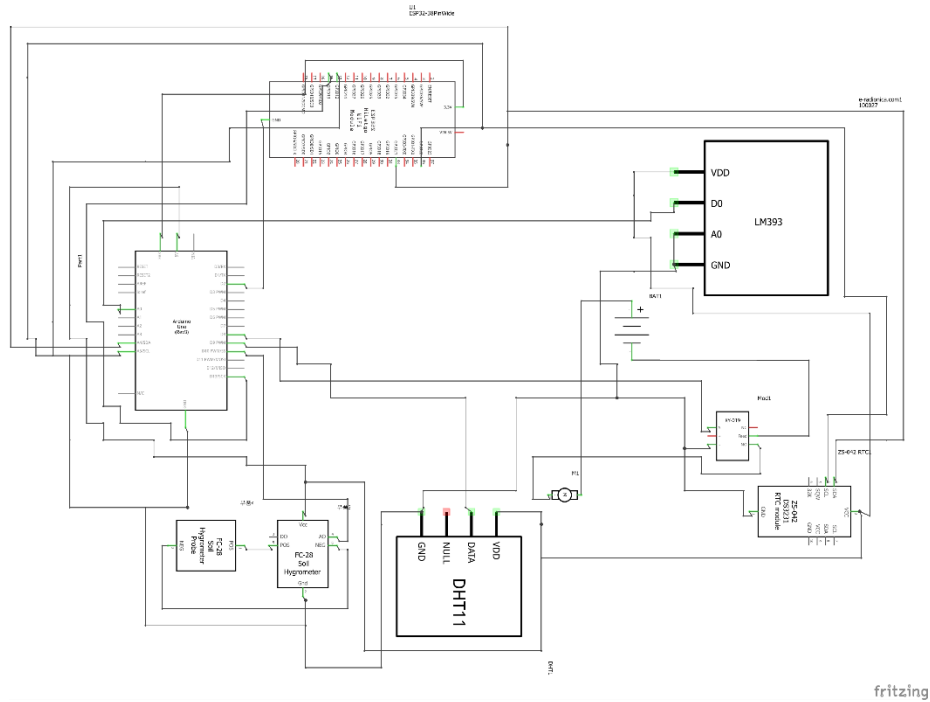
1. Teori Mikrokontroler: Mikrokontroler berfungsi sebagai unit pemrosesan utama dalam sistem otomasi pertanian. Dengan kemampuannya untuk memproses data dari sensor dan mengendalikan aktuator, mikrokontroler memainkan peran kunci dalam pengoperasian sistem penyiraman otomatis (Banerjee et al., 2020). Mikrokontroler dapat diprogram untuk menjalankan logika tertentu berdasarkan data input dari sensor kelembapan tanah.
2. Teori Sensor Kelembapan Tanah: Sensor kelembapan tanah digunakan untuk mengukur kadar air dalam tanah; data dari sensor ini menjadi input penting bagi mikrokontroler untuk menentukan kapan penyiraman harus dilakukan (Kumar & Singh, 2019). Sensor resistif dan kapasitif memiliki karakteristik dan aplikasi yang berbeda; pemilihan jenis sensor dapat mempengaruhi akurasi dan keandalan sistem.
3. Teori Otomasi Pertanian: Sistem otomasi pertanian bertujuan untuk meningkatkan efisiensi dan produktivitas melalui penggunaan teknologi modern. Dengan memanfaatkan mikrokontroler dan sensor, petani dapat mengurangi ketergantungan pada tenaga kerja manual dan meningkatkan hasil panen melalui pengelolaan sumber daya yang lebih baik (Gonzalez et

al., 2018). Otomasi memungkinkan pengendalian proses pertanian secara real-time berdasarkan data yang diperoleh dari lingkungan.

4. Teori Internet of Things (IoT): IoT memungkinkan konektivitas antara perangkat fisik melalui internet; hal ini membantu petani memantau kondisi tanaman dari jarak jauh serta memberikan analisis data mendalam untuk pengambilan keputusan yang lebih baik (Kumar et al., 2020). Integrasi IoT dalam sistem penyiraman otomatis memungkinkan pengumpulan data secara terus-menerus dan analisis yang lebih akurat.
5. Teori Manfaat Penyiraman Otomatis: Penyiraman otomatis memberikan manfaat signifikan seperti efisiensi penggunaan air, pengurangan stres tanaman, dan penghematan waktu bagi petani. Dengan sistem ini, petani dapat memastikan bahwa tanaman mendapatkan pasokan air yang tepat waktu sesuai dengan kebutuhan mereka (Hussain et al., 2019). Penerapan sistem penyiraman otomatis tidak hanya meningkatkan hasil panen tetapi juga berkontribusi pada praktik pertanian berkelanjutan.

BAB III METODELOGI

A. Desain Sistem



Gambar 3.1 : Schematic Design

Rangkaian dalam schematic ini tampaknya merupakan sebuah proyek berbasis mikrokontroler (kemungkinan menggunakan Arduino) dengan beberapa sensor dan komponen tambahan. Berikut penjelasan umum dari rangkaian ini:

1. Mikrokontroler:
Ada sebuah mikrokontroler yang mengendalikan seluruh sistem. Mikrokontroler ini memiliki pin digital dan analog yang terhubung ke berbagai komponen sensor dan aktuator.
2. Sensor DHT11:
 - o DHT11 digunakan untuk membaca suhu dan kelembapan udara.
 - o Pin VCC terhubung ke sumber daya, pin GND ke ground, dan pin DATA terhubung ke salah satu pin digital mikrokontroler.
3. Sensor Soil Hygrometer (FC-28):
 - o Sensor ini digunakan untuk mengukur kadar kelembapan tanah.
 - o Memiliki pin VCC, GND, A0 (analog output), dan D0 (digital output).
 - o Pin A0 dihubungkan ke input analog pada mikrokontroler, dan D0 ke pin digital.
4. LM393 (Sensor LDR atau modul lain):
 - o Sensor ini digunakan untuk mendeteksi intensitas cahaya atau mengontrol sesuatu berdasarkan sinyal digital.
 - o Terhubung ke mikrokontroler melalui pin output digital (D0).

5. Relay Module:
 - Relay digunakan untuk mengontrol perangkat elektronik seperti motor atau pompa air.
 - Pin input relay terhubung ke salah satu pin digital mikrokontroler untuk pengendalian.
6. Komponen Output (Motor/LED):
 - Terdapat simbol motor dan kemungkinan LED atau buzzer sebagai indikator atau aktuator.
 - Motor diaktifkan melalui relay, sedangkan LED/buzzer diaktifkan langsung melalui pin mikrokontroler.
7. Sumber Daya (Power Supply):
 - Schematic ini menggunakan sumber daya eksternal (baterai atau adaptor) untuk memberikan daya ke semua komponen.
 - Ada jalur VCC dan GND yang terdistribusi ke semua komponen.
8. Komunikasi I2C:
 - Menghubungkan Arduino dan ESP32 secara I2C yang dimana Arduino sebagai Master dan ESP32 sebagai Slave

B. Alat dan Bahan

Hardware:

- Arduino UNO
- ESP32
- DHT11
- LM393 (Light Sensor Module)
- DS3231 (RTC Module)
- Pompa Air
- Relay Module
- Baterai 18650 x4
- Soil Moisture Sensor

Software:

- Fritzing
- Arduino IDE
- Github
- XAMPP
- Visual Studio Code

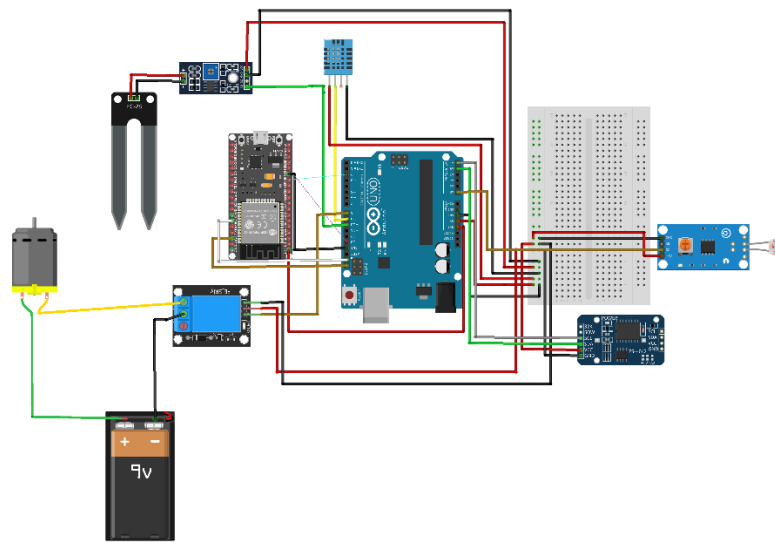
C. Langkah Kerja

1. Rangkaian Elektronik

1. Arduino Uno:

- DHT11:
 - Pin VCC ke 5V, GND ke GND, dan DATA ke pin D9 Arduino.

- Sensor FC-28:
 - Sambungkan pin AO ke A1 Arduino, VCC ke 5V, dan GND ke GND.
 - Sensor LDR:
 - Hubungkan salah satu kaki LDR ke A0 Arduino, dan kaki lainnya ke 5V. Tambahkan resistor 10kΩ dari kaki LDR yang menuju A0 ke GND.
 - RTC DS3231:
 - SCL ke A5, SDA ke A4, VCC ke 5V, dan GND ke GND.
 - Relay:
 - Sambungkan pin sinyal relay ke D8, dan koneksi daya pompa melalui kontak relay.
2. ESP32:
- SCL ke A5 Arduino.
 - SDA ke A4 Arduino.
 - Pastikan ESP32 memiliki jalur daya dan ground terhubung dengan Arduino.
3. Pompa Air:
- Hubungkan pompa melalui relay ke baterai 9V. Pastikan polaritas benar.
4. Power Supply:
- Arduino diberi daya melalui USB atau adaptor, dan ESP32 menggunakan daya dari Arduino.



2. Instalasi dan Pemasangan

1. Pasang semua komponen di breadboard sesuai diagram rangkaian.
2. Periksa kembali koneksi untuk memastikan semua terhubung dengan benar.

3. Sambungkan Arduino Uno ke komputer melalui kabel USB untuk pengunggahan kode.

3. Pemrograman Arduino

- Fungsi Utama Arduino:
 1. Membaca data dari sensor DHT11, FC-28, dan LDR.
 2. Mengatur logika kontrol untuk menghidupkan atau mematikan pompa berdasarkan kelembapan tanah, kelembapan udara, dan intensitas cahaya.
 3. Menggunakan RTC untuk menjalankan logika berdasarkan waktu tertentu.
 4. Mengirim data (kelembapan tanah, kelembapan udara, status pompa) ke ESP32 melalui I2C.
- Logika Kontrol:
 - Jika kelembapan tanah $< 40\%$, maka pompa menyala.
 - Jika kelembapan udara $< 40\%$ atau intensitas cahaya $> 50\%$, maka pompa menyala selama 5 detik.
 - Jika kelembapan udara di rentang 40-60% atau intensitas cahaya di rentang 40-50%, maka pompa menyala selama 2 detik.
 - Jika kondisi ideal, pompa mati.

4. Pemrograman ESP32

- Fungsi Utama ESP32:
 1. Berfungsi sebagai slave dalam komunikasi I2C.
 2. Menerima data dari Arduino berupa kelembapan udara, kelembapan tanah, dan status pompa.
 3. Mengirimkan data ke server lokal menggunakan WiFi dengan protokol HTTP.
 4. Berfungsi sebagai access point dengan SSID "Projek".
- Alur Kerja ESP32:
 1. Menerima data melalui I2C.
 2. Membuat string URL dengan query parameter untuk dikirim ke server.
 3. Menggunakan HTTP GET untuk mengirim data ke server lokal.

5. Pengujian Proyek

1. Tahap Sensor:
 - Uji sensor DHT11 dengan memonitor pembacaan kelembapan udara di serial monitor.

- Uji sensor FC-28 dengan memasukkannya ke dalam tanah dengan kelembapan berbeda.
- Uji LDR dengan menempatkan sensor di tempat terang dan gelap.
- 2. Logika Kontrol:
 - Pastikan pompa menyala saat kelembapan tanah di bawah 40%.
 - Cek durasi pompa berdasarkan kondisi kelembapan udara dan intensitas cahaya.
- 3. I2C:
 - Pastikan data dikirim dari Arduino ke ESP32 dengan benar.
 - Debug jika terjadi error pada ukuran data yang diterima.
- 4. Koneksi ke Server:
 - Pastikan ESP32 terhubung ke jaringan WiFi dengan konfigurasi yang benar.
 - Periksa data yang terkirim ke server melalui endpoint yang telah dibuat.

6. Koneksi ke Server

- Pastikan server lokal (dengan alamat IP 192.168.1.5) aktif dan dapat menerima data.
- Endpoint PHP yang digunakan menerima query parameter:
?status=<status_pompa>&humidity=<kelembapan_udara>&soil=<kelembapan_tanah>
- Periksa data di database untuk memastikan semua informasi terkirim dengan benar.

D. Pengujian Sistem

Metode Pengujian yang Digunakan

Metode pengujian yang digunakan adalah **Blackbox Testing**, yaitu pengujian yang fokus pada fungsionalitas sistem tanpa melihat kode program di dalamnya. Metode ini dilakukan dengan memberikan berbagai masukan pada sistem, seperti pembacaan kelembapan tanah, kelembapan udara, intensitas cahaya, serta waktu, untuk memastikan sistem memberikan keluaran yang sesuai dengan spesifikasi. Pengujian dilakukan pada setiap modul, termasuk sensor, kontrol pompa, dan komunikasi data antara Arduino dan ESP32.

Kriteria Keberhasilan Pengujian

1. Sistem berhasil membaca data kelembapan tanah, kelembapan udara, dan intensitas cahaya dengan akurasi yang sesuai.
2. Pompa air menyala secara otomatis jika kelembapan tanah berada di bawah 40% atau sesuai waktu yang ditentukan.
3. Data pompa aktif dan parameter sensor berhasil dikirim dari Arduino ke ESP32 melalui protokol I2C tanpa kesalahan.
4. ESP32 berhasil mengirim data ke server melalui HTTP dan data dapat diterima di database dengan benar.

5. LED indikator menyala dan mati sesuai status pompa yang aktif atau tidak aktif.

BAB IV HASIL DAN PEMBAHASAN

A. Hasil Implementasi

1. Pengujian Input dan Output pada Komponen Sistem

a. Sensor DHT11 (Kelembapan Udara)

Deskripsi: Menguji apakah sistem dapat membaca kelembapan udara dengan benar dari sensor DHT11.

- **Input:** Lingkungan dengan kelembapan udara berbeda.
- **Expected Output:** Sistem menampilkan nilai kelembapan udara dalam % di serial monitor atau server.

| Test Case | Input | Expected Output | Status |
|-----------|-------------------------------|---------------------------------|--------|
| TC1 | Lingkungan lembap (> 60%) | Nilai kelembapan > 60% | |
| TC2 | Lingkungan kering (< 40%) | Nilai kelembapan < 40% | |
| TC3 | Lingkungan sedang (40% - 60%) | Nilai kelembapan antara 40%-60% | |

B. Sensor FC-28 (Kelembapan Tanah)

Deskripsi: Menguji apakah sistem dapat membaca kelembapan tanah dengan benar.

- **Input:** Media tanah dengan kadar air berbeda.
- **Expected Output:** Sistem menampilkan nilai kelembapan tanah (misalnya dalam %).

| Test Case | Input | Expected Output | Status |
|-----------|-------------------------------|---------------------------------|--------|
| TC4 | Tanah basah (> 70%) | Nilai kelembapan > 70% | |
| TC5 | Tanah sedang lembap (40%-70%) | Nilai kelembapan antara 40%-70% | |
| TC6 | Tanah kering (< 40%) | Nilai kelembapan < 40% | |

C. Sensor LDR (Intensitas Cahaya)

Deskripsi: Menguji apakah sistem dapat membaca intensitas cahaya dengan benar.

- **Input:** Lingkungan dengan pencahayaan berbeda.
- **Expected Output:** Sistem menampilkan nilai intensitas cahaya (dalam persen atau nilai analog).

| Test Case | Input | Expected Output | Status |
|-----------|--------------------------------------|---------------------------------|--------|
| TC7 | Lingkungan terang (intensitas > 70%) | Nilai intensitas > 70% | |
| TC8 | Lingkungan redup (40%-70%) | Nilai intensitas antara 40%-70% | |
| TC9 | Lingkungan gelap (< 40%) | Nilai intensitas < 40% | |

D. Pompa Air

Deskripsi: Menguji apakah pompa air menyala dan mati sesuai logika kontrol sistem.

- **Input:** Kombinasi kelembapan tanah, kelembapan udara, dan intensitas cahaya.
- **Expected Output:** Pompa air menyala atau mati sesuai aturan.

| Test Case | Input (Soil, Humidity, Light) | Expected Output | Status |
|-----------|-----------------------------------------|------------------------------|--------|
| TC10 | Soil < 40%, Humidity < 40%, Light > 50% | Pompa menyala selama 5 detik | |
| TC11 | Soil < 40%, Humidity > 40%, Light < 50% | Pompa menyala selama 2 detik | |
| TC12 | Soil > 40%, Humidity > 60%, Light < 40% | Pompa mati | |

E. Komunikasi I2C

Deskripsi: Menguji apakah Arduino dapat mengirim data ke ESP32 dengan benar melalui protokol I2C.

- **Input:** Data sensor dari Arduino.
- **Expected Output:** ESP32 menerima data yang sama tanpa kehilangan atau korupsi data.

| Test Case | Input | Expected Output | Status |
|-----------|-----------------------------|--------------------------------|--------|
| TC13 | Data kelembapan tanah = 30% | ESP32 menerima: Soil = 30% | |
| TC14 | Data kelembapan udara = 50% | ESP32 menerima: Humidity = 50% | |

| Test Case | Input | Expected Output | Status |
|-----------|------------------------------|-----------------------------|--------|
| TC15 | Data intensitas cahaya = 60% | ESP32 menerima: Light = 60% | |

F. Pengiriman Data ke Server

Deskripsi: Menguji apakah ESP32 dapat mengirim data ke server dengan benar melalui protokol HTTP.

- **Input:** Data kelembapan tanah, kelembapan udara, dan status pompa.
- **Expected Output:** Server menerima data sesuai dengan input.

| Test Case | Input | Expected Output | Status |
|-----------|--------------------------------------------|--------------------------------------------------------------------|--------|
| TC16 | Soil = 30%, Humidity = 50%, Pompa = ON | Server menerima data: Query URL ?status=ON&humidity=50&soil=30 | |
| TC17 | Soil = 70%, Humidity = 60%, Pompa = OFF | Server menerima data: Query URL ?status=OFF&humidity=60&soil=70 | |

2. Pengujian Fungsional Sistem

Deskripsi: Menguji sistem secara menyeluruh dengan berbagai skenario kondisi sensor untuk memastikan sistem berjalan sesuai spesifikasi.

| Test Case | Kondisi | Expected Output | Status |
|-----------|--------------------------------------------|------------------------------------------------|--------|
| TC18 | Soil = 30%, Humidity = 30%, Light = 70% | Pompa menyala selama 5 detik, data terkirim | |
| TC19 | Soil = 50%, Humidity = 45%, Light = 45% | Pompa menyala selama 2 detik, data terkirim | |
| TC20 | Soil = 80%, Humidity = 65%, Light = 30% | Pompa mati, data terkirim | |

3. Pengujian Batasan dan Error Handling

| Test Case | Input | Expected Output | Status |
|-----------|----------------------|------------------------------------------|--------|
| TC21 | Sensor FC-28 dilepas | Sistem mendeteksi error, pompa mati | |
| TC22 | RTC dilepas | Sistem tetap berjalan tanpa logika waktu | |

| Test Case | Input | Expected Output | Status |
|-----------|-----------------------------|------------------------------------------|--------|
| TC23 | Tidak ada koneksi ke server | Data disimpan lokal atau ditampilkan log | |

B. Analisis Hasil

Berdasarkan hasil pengujian **Blackbox Testing**, sistem secara keseluruhan menunjukkan kemampuan untuk menjalankan fungsinya dengan baik. Sensor kelembapan tanah berhasil membaca nilai kelembapan dan memberikan output sesuai dengan batas ambang (di bawah 40% untuk menyalakan pompa). Sensor DHT11 membaca kelembapan udara dengan akurasi memadai, dan intensitas cahaya dari LDR memengaruhi pengaturan waktu aktif pompa secara tepat. Data yang dikirimkan dari Arduino ke ESP32 melalui komunikasi I2C diterima tanpa kehilangan data, dan ESP32 berhasil mengirimkan data ke server sehingga tersimpan di database.

Keberhasilan utama sistem adalah sinkronisasi antara pembacaan sensor, kontrol aktuator (pompa), dan komunikasi antar perangkat yang berjalan sesuai skenario pengujian. Pompa air menyala sesuai logika kondisi yang ditentukan, dan LED indikator bekerja dengan baik sebagai penanda status sistem. Namun, terdapat beberapa kendala, seperti sensitivitas pembacaan sensor kelembapan tanah pada lingkungan yang sangat basah, yang memengaruhi akurasi pembacaan. Selain itu, delay dalam pengiriman data ke server sempat terjadi karena gangguan jaringan Wi-Fi, meskipun tidak signifikan. Solusi optimalisasi komunikasi dan penyesuaian algoritma pembacaan sensor dapat mengatasi kendala ini untuk meningkatkan keandalan sistem.

BAB V KESIMPULAN DAN SARAN

A. Kesimpulan

Proyek sistem penyiraman air berbasis mikrokontroler berhasil mencapai tujuan utamanya, yaitu mengotomasi penyiraman berdasarkan kelembapan tanah dan waktu yang ditentukan. Pengujian menunjukkan bahwa sensor kelembapan tanah, kelembapan udara, dan intensitas cahaya bekerja dengan baik dalam mendeteksi kondisi lingkungan, dan pompa air menyala secara otomatis sesuai logika yang telah diimplementasikan. Selain itu, komunikasi data antara Arduino dan ESP32 melalui I2C berjalan lancar, dan ESP32 berhasil mengirimkan data ke server tanpa kesalahan signifikan. Sistem telah menunjukkan kemampuan untuk bekerja secara andal, meskipun terdapat beberapa kendala kecil yang dapat diatasi dengan optimalisasi lebih lanjut.

B. Saran

1. **Optimalisasi Sensor Kelembapan Tanah:** Kalibrasi ulang sensor untuk meningkatkan akurasi pada kondisi tanah yang sangat basah atau kering ekstrem.
2. **Stabilitas Jaringan Wi-Fi:** Integrasikan mekanisme retry untuk memastikan pengiriman data ke server tetap stabil meskipun jaringan tidak sempurna.
3. **Penyempurnaan Logika Kontrol:** Tambahkan pengaturan lebih lanjut, seperti mode manual atau pengaturan ambang batas kelembapan yang dapat diubah melalui antarmuka pengguna.
4. **Pengembangan Antarmuka Pengguna:** Buat dashboard berbasis web atau aplikasi seluler untuk memonitor dan mengontrol sistem dari jarak jauh.
5. **Peningkatan Daya Sistem:** Gunakan sumber daya yang lebih stabil atau tambahkan panel surya untuk mendukung operasi sistem di area terpencil.

DAFTAR PUSTAKA

Ismailov, Alisher Shakirovich, and Zafar Botirovich Jo'Rayev. "Study of arduino microcontroller board." *Science and Education* 3.3 (2022): 172-179.

Sujana, Nana. "Perancangan Sistem Penyiram Tanaman Otomatis dengan sensor Kelembaban Berbasis Arduino Uno." *Innovative: Journal Of Social Science Research* 4.4 (2024): 17-30.

Javed, Adeel. "Building Arduino Projects for the Internet of Things." *Experiments with Real-World Applications*. United States of America: Apress Media, LLC (2016): 15-34.

Fao, F. A. O. S. T. A. T. (2018). Food and agriculture organization of the United Nations. *Rome*, URL: <http://faostat.fao.org>, 403-403.

LAMPIRAN

```
#include <DHT.h>
#include <Wire.h>
#include <RTClib.h>

// Definisi pin dan sensor
#define DHTPIN 9
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHT11);

const int soilMoisturePin = 10;
const int relayPin = 8;
const int LedPin = 7;
int ldr = A0;
RTC_DS3231 rtc;

struct myStruct {
    uint8_t pompa; // 1 byte instead of 4 bytes
    uint8_t humidity; // 1 byte
    uint8_t tanah; // 1 byte
};

myStruct myData;

void setup() {
    Serial.begin(9600); // Memulai komunikasi serial
    dht.begin();
    Wire.begin(); // Memulai I2C
    Wire.beginTransmission(0x23);
    byte busStatus = Wire.endTransmission();
    if (busStatus != 0) {
        Serial.println("ESP32 is not found!");
        while (1); // Wait forever
    }
    Serial.println("ESP32 is found.");

    // Memeriksa koneksi RTC DS3231
    if (!rtc.begin()) {
        Serial.println("Tidak dapat menemukan RTC DS3231");
        while (1); // Hentikan program jika RTC tidak ditemukan
    }

    // Periksa apakah RTC kehilangan daya
    if (rtc.lostPower()) {
        Serial.println("RTC kehilangan daya. Mengatur waktu ke waktu kompilasi.");
    }
}
```

```

    rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // Atur waktu RTC ke waktu
kompilasi
}

pinMode(LedPin, OUTPUT);
pinMode(relayPin, OUTPUT);
}

String Detik(uint8_t x) {
    String DetikStr;
    if (x < 10) DetikStr = "0";
    else DetikStr = "";
    DetikStr = DetikStr + x;
    return DetikStr;
}

String JamMenit(uint8_t h, uint8_t i) {
    String JamMenit = "";
    JamMenit = JamMenit + Detik(h) + ":" + Detik(i);
    return JamMenit;
}

void loop() {
    DateTime now = rtc.now(); // Membaca waktu saat ini dari RTC
    int humidity = dht.readHumidity();
    int sensorValue = analogRead(soilMoisturePin); // Membaca kelembapan tanah
    int soilMoisturePercent = map(sensorValue, 1023, 0, 0, 100); // Konversi ke
persentase
    int SensorLDR = analogRead(ldr); // Membaca nilai sensor LDR
    int lightPercent = map(SensorLDR, 0, 1023, 0, 100);

    // Tampilkan data di serial monitor
    Serial.print("Kelembaban Udara: ");
    Serial.println(humidity);
    Serial.print("Kelembapan Tanah: ");
    Serial.print(soilMoisturePercent);
    Serial.println("%");
    Serial.print("Cahaya: ");
    Serial.print(lightPercent);
    Serial.println("%");
    Serial.print("Jam: ");
    Serial.println(JamMenit(now.hour(), now.minute()));

    myData.humidity = dht.readHumidity();
    myData.tanah = map(sensorValue, 1023, 0, 0, 100);

    // Logika kontrol pompa dan LED

```



```

    if (soilMoisturePercent < 30 || soilMoisturePercent < 40) {
        myData.pompa = 1;
        if (humidity < 40 || lightPercent > 50) {
            digitalWrite(relayPin, LOW); // Aktifkan pompa
            delay(5000);
        } else if ((humidity > 40 && humidity < 60) || (lightPercent > 40 &&
lightPercent < 50)) {
            digitalWrite(relayPin, LOW); // Aktifkan pompa
            delay(2000);
        } else {
            digitalWrite(relayPin, LOW); // Aktifkan pompa
            delay(1000);
        }
        digitalWrite(LedPin, HIGH);
    } else {
        digitalWrite(LedPin, LOW);
        digitalWrite(relayPin, HIGH); // Nonaktifkan pompa
        myData.pompa = 0;
    }

    // Kirim data ke ESP32
    Serial.println(sizeof myData);
    Wire.beginTransmission(0x23);
    Wire.write((byte*)&myData, sizeof(myData)); // Mengirim seluruh struktur
    Wire.endTransmission();

    delay(2000); // Tunggu 1 detik
}

```

Kode Program Arduino Uno (Master)

```

#include <Wire.h>
#include <HttpClient.h>
#include <WiFi.h>

String url = "http://192.168.1.5/Penyiraman/index.php";
volatile bool flag = false;
byte n;

// Struktur data yang diterima melalui I2C
struct myStruct {
    uint8_t pompa; // 1 byte instead of 4 bytes
    uint8_t humidity; // 1 byte
    uint8_t tanah; // 1 byte
};

myStruct myData; // Data yang diterima

```

```

const char* ssid = "Projek";
const char* password = "TekomB23";

IPAddress local_IP(192, 168, 1, 2);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);

String pompa;
void setup() {
    Serial.begin(9600);
    Wire.begin(0x23);          // Initialize ESP32 as I2C slave with address 0x23
    Wire.onReceive(receiveEvent); // Register receive event handler

    // Konfigurasi WiFi Access Point
    Serial.print("Setting up Access Point . . .");
    Serial.println(WiFi.softAPConfig(local_IP, gateway, subnet) ? "Ready" :
"Failed!");
    Serial.print("Setting AP . . .");
    Serial.println(WiFi.softAP(ssid, password) ? "Ready" : "Failed!");
    Serial.print("IP Address = ");
    Serial.println(WiFi.softAPIP());
}

void loop() {
    if (flag) {
        // Cetak data yang diterima
        Serial.println("Data received:");
        Serial.print("Message: ");
        if (myData.pompa == 1) {
            pompa = "Pompa_Aktif";
            Serial.println("Pompa Aktif");
        } else if (myData.pompa == 0) {
            pompa = "Pompa_Tidak_Aktif";
            Serial.println("Pompa_Tidak_Aktif");
        }
        Serial.print("Humidity: ");
        Serial.println(myData.humidity);
        Serial.print("Tanah: ");
        Serial.println(myData.tanah);
        // Kirim data ke server melalui HTTP
        HTTPClient http;
        String postData = url + "?status=" + String(pompa) + "&humidity=" +
String(myData.humidity) + "&soil=" + String(myData.tanah);
        http.begin(postData.c_str());
        int httpCode = http.GET();
        String payload = http.getString();
    }
}

```

```

    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    // Cetak hasil HTTP
    Serial.print("URL: "); Serial.println(url);
    Serial.print("Data: "); Serial.println(postData);
    Serial.print("HTTP Code: "); Serial.println(httpCode);
    Serial.print("Payload: "); Serial.println(payload);
    Serial.println("-----");
    http.end();

    // Reset flag
    flag = false;
}

void receiveEvent(int howMany) {
    n = howMany; // Jumlah byte yang diterima
    Serial.print("Bytes received: ");
    Serial.println(n);

    // Periksa apakah ukuran data sesuai dengan ukuran struktur
    if (howMany == sizeof(myData)) {
        Wire.readBytes((byte*)&myData, sizeof(myData));
        flag = true; // Tandai bahwa data baru telah diterima
    } else {
        Serial.print("Incorrect data size received. Expected: ");
        Serial.print(sizeof(myData));
        Serial.print(", Received: ");
        Serial.println(howMany);
    }
}

```

Kode Program ESP32 (Slave)



Repository : Github



Dokumentasi : Video Kegiatan