



**SISTEM OTOMATISASI PERANGKAT *SMART HOME* BERDASARKAN  
PRAKIRAAN CUACA MENGGUNAKAN NODE.JS**

**TUGAS AKHIR**



UNIVERSITAS  
**Dinamika**

**Oleh:**

**MUHAMMAD AZHAR ALI**

**21410200020**

---

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

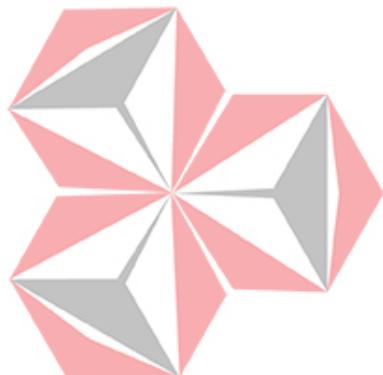
**UNIVERSITAS DINAMIKA**

**2025**

**SISTEM OTOMATISASI PERANGKAT *SMART HOME* BERDASARKAN  
PRAKIRAAN CUACA MENGGUNAKAN NODE.JS**

**TUGAS AKHIR**

**Diajukan sebagai syarat untuk menyelesaikan  
Program Sarjana**



Oleh:  
Nama : Muhammad Azhar Ali  
NIM : 21410200020  
Program : S1 (Strata Satu)  
Jurusan : Teknik Komputer

**UNIVERSITAS**  
**Dinamika**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
UNIVERSITAS DINAMIKA  
2025**

## TUGAS AKHIR

### SISTEM OTOMATISASI PERANGKAT SMART HOME BERDASARKAN PRAKIRAAN CUACA MENGGUNAKAN NODE.JS

Dipersiapkan dan disusun oleh:

**Muhammad Azhar Ali**

**NIM : 21410200020**

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahasan

Pada: 30 Juli 2025

#### Susunan Dewan Pembahasan

##### Pembimbing:

I Harianto, S.Kom., M.Eng.

NIDN. 0722087701

cn=Harianto Harianto,  
o=Universitas Dinamika,  
ou=Prodi S1 Teknik Komputer,  
email=hari@dinamika.ac.id,  
c=ID  
2025.08.07 16:50:28 +07'00'

II Weny Indah Kusumawati, S.Kom., M.MT.

NIDN. 0721047201

cn=Weny Indah Kusumawati,  
o=Undika, ou=Prodi S1 TK - FTI,  
email=weny@dinamika.ac.id, c=ID  
2025.08.06 17:56:14 +07'00'

##### Pembahasan:

I Pauladie Susanto, S.Kom., M.T.

NIDN. 0729047501

cn=Pauladie Susanto, o=Universitas  
Dinamika, ou=PS S1 Teknik Komputer,  
email=pauladie@dinamika.ac.id, c=ID  
2025.08.08 12:22:36 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar Sarjana

Digitally signed by

Julianto

Date: 2025.08.12

18:50:10 +07'00'

**Julianto Lemantara, S.Kom., M.Eng.**

**NIDN. 0722108601**

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

**PERNYATAAN  
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, Saya :

Nama : Muhammad Azhar Ali  
NIM : 21410200020  
Program Studi : S1 Teknik Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Laporan Tugas Akhir  
Judul Karya : **SISTEM OTOMATISASI PERANGKAT SMART HOME BERDASARKAN PRAKIRAAN CUACA MENGGUNAKAN NODE.JS**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikan surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 30 Juni 2025



Muhammad Azhar Ali  
NIM : 21410200020

## ABSTRAK

Seiring meningkatnya kebutuhan efisiensi, pengelolaan perangkat rumah dan taman secara manual seringkali tidak efektif dan boros sumber daya. Sistem otomatisasi yang ada saat ini terkadang kurang terintegrasi dengan faktor eksternal seperti kondisi cuaca dan rentan gagal saat koneksi internet terputus. Penelitian ini bertujuan untuk mengembangkan "SilverLink," sebuah ekosistem Internet of Things (IoT) terintegrasi untuk otomatisasi penyiraman cerdas. Sistem ini dibangun menggunakan tiga komponen utama: controller berbasis ESP32, backend server dengan Node.js, dan aplikasi mobile. Proses pengembangan sistem ini meliputi perancangan perangkat keras, pemrograman firmware untuk akuisisi data sensor, dan pengembangan backend yang mengintegrasikan data kelembapan tanah dengan prakiraan cuaca dari API BMKG. Pengujian akurasi sensor menunjukkan nilai error sebesar 7.96% untuk sensor kelembapan tanah, 21.02% untuk sensor kelembapan udara, dan 6.43% untuk sensor suhu udara, dimana keseluruhan hasil berada di luar batas toleransi yang diharapkan sehingga memerlukan kalibrasi lebih lanjut. Meskipun demikian, hasil dari proyek ini adalah sebuah sistem terpadu yang mampu melakukan penyiraman tanaman secara cerdas dan efisien dengan mempertimbangkan prediksi hujan. Sistem menunjukkan keandalan yang tinggi berkat fitur mode offline pada controller dan menyediakan antarmuka monitoring serta kontrol terpusat yang intuitif melalui aplikasi mobile, yang secara efektif meningkatkan efisiensi penggunaan sumber daya air dan energi.

**Kata Kunci :** *Internet of Things, Smart Home, Sistem Penyiraman Otomatis.*



## KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul "Otomatisasi Perangkat Smart Home Berdasarkan Prakiraan Cuaca Menggunakan Node.JS". Selama proses penyusunan Laporan Tugas Akhir ini, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya. Ucapan terima kasih ini ditujukan kepada:

1. Allah SWT, karena dengan Rahmat-Nya penulis dapat menyelesaikan Laporan Tugas Akhir ini.
2. Orang tua dan seluruh keluarga yang sudah memberikan dukungan dan motivasi selama proses penyelesaian Laporan Tugas Akhir ini.
3. Bapak Julianto Lemantara, S.Kom., M.Eng., Selaku Dekan Fakultas Teknologi dan Informatika Universitas Dinamika.
4. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik komputer sekaligus dosen pembahas, atas masukan dan saran, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini.
5. Bapak Harianto, S.Kom., M.Eng., selaku Dosen Pembimbing 1 yang telah memberikan masukan, wawasan, dan arahan bagi penulis selama penggerjaan dan pembuatan Laporan Tugas Akhir ini.
6. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku Dosen Pembimbing 2 yang telah memberikan bimbingan, dukungan, dan motivasi bagi penulis dalam penggerjaan dan pembuatan Laporan Tugas Akhir ini.
7. Seluruh rekan-rekan S1 Teknik komputer Angkatan 2021, yang telah membantu memberikan dukungan dan semangat.
8. Seluruh pihak baik nyata maupun fiksi yang tidak dapat penulis sebutkan satu per satu, yang telah memberikan dukungan, motivasi, serta bantuan, baik secara langsung maupun tidak langsung dalam penyelesaian Laporan Tugas Akhir ini.

Penulis sangat menyadari bahwa masih terdapat kekurangan pada penulisan laporan ini. Oleh karena itu, penulis mengharapkan kritik dan saran yang

membangun agar laporan ini menjadi lebih baik. Semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembaca.

Surabaya, 30 Juli 2025

Penulis



## DAFTAR ISI

	Halaman
ABSTRAK .....	v
KATA PENGANTAR.....	vi
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Manfaat .....	2
BAB II LANDASAN TEORI .....	3
2.1 Smart Home .....	3
2.2 <i>Internet of Things</i> (IoT) .....	3
2.3 Node.js .....	4
2.4 Capacitor.js .....	4
2.5 Ionic Framework.....	5
2.6 PostgreSQL.....	5
2.7 WebSocket .....	5
2.8 LittleFS .....	6
2.9 API .....	6
2.10ESP32.....	7
2.11DHT11 .....	7
2.12Capacitive Soil Moisture Sensor.....	8
2.13API Cuaca BMKG .....	9
BAB III METODOLOGI PENELITIAN.....	10
3.1 Studi Literatur.....	10
3.2 Mempersiapkan Alat dan Bahan .....	10
3.2.1 Alat .....	10
3.2.2 Bahan.....	11



**UNIVERSITAS**  
**Dinamika**

3.3 Perancangan Sistem .....	11
3.3.1 Arsitektur Keseluruhan Sistem.....	11
3.3.2 Perancangan Perangkat Keras .....	17
3.3.3 Perancangan Perangkat Lunak .....	20
BAB IV HASIL DAN PEMBAHASAN .....	35
4.1 Hasil Implementasi Sistem .....	35
4.1.1 Hasil Implementasi Hardware .....	35
4.1.2 Hasil Implementasi Software .....	37
4.2 Pengujian Fungsionalitas Sistem .....	38
4.2.1 Pengujian Suhu.....	38
4.2.2 Pengujian Kelembapan Udara .....	39
4.2.3 Pengujian Kelembapan Tanah .....	40
4.2.4. Pengujian Penyiraman Otomatis .....	41
BAB V PENUTUP.....	42
DAFTAR PUSTAKA .....	44
LAMPIRAN .....	45



UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR



	Halaman
Gambar 2.1 ESP32 .....	7
Gambar 2.2 Sensor DHT 11 .....	8
Gambar 2.3 Capacitive Soil Moisture Sensor .....	9
Gambar 3.1 Diagram blok sistem.....	17
Gambar 3.2 Skematik perangkat mode penyiram otomatis .....	18
Gambar 3.3 Skematik perangkat mode kendali saklar.....	19
Gambar 3.4 Skematik perangkat mode pengukuran suhu dan kelembapan.....	20
Gambar 3.5 Flowchart Loop ESP32 .....	21
Gambar 3.6 Flowchart Setup Mode .....	22
Gambar 3.7 Flowchart WebSocket Loop .....	23
Gambar 3.8 Flowchart Mode Offline .....	24
Gambar 3.9 Flowchart Send Sensor Data .....	25
Gambar 3.10 Diagram Arsitektur Backend .....	26
Gambar 3.11 Flowchart Start Server.....	27
Gambar 3.12 Flowchart penyiraman otomatis .....	28
Gambar 3.13 Struktur Database .....	29
Gambar 3.14 Halaman Dashbord.....	32
Gambar 3.15 Halaman pengaturan penyiraman.....	33
Gambar 3.16 Halaman monitoring sensor .....	34
Gambar 4.1 Perangkat penyiram otomatis .....	35
Gambar 4.2 Perangkat mode kendali saklar.....	36
Gambar 4.3 Perangkat pengukur suhu dan kelembapan .....	36

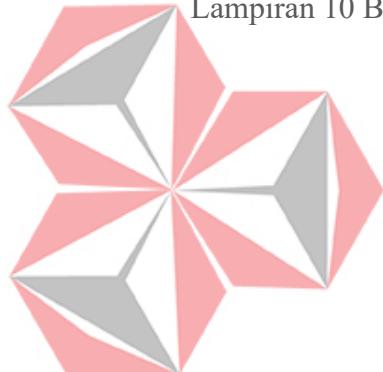
## **DAFTAR TABEL**

	Halaman
Tabel 4.1 Hasil uji suhu.....	38
Tabel 4.2 Hasil uji kelembapan udara .....	39
Tabel 4.3 Hasil uji kelembapan tanah .....	40
Tabel 4.4 Hasil uji sistem penyiraman otomatis .....	41



## **DAFTAR LAMPIRAN**

	Halaman
Lampiran 1 Flowchart HTTP API Handler .....	45
Lampiran 2 Flowchart Websocket Handle .....	46
Lampiran 3 Sistem Otentikasi User .....	47
Lampiran 4 Kode Program Penyiraman Otomatis .....	48
Lampiran 5 Kode Program ESP32.....	65
Lampiran 6 Kode Program Dashboard Aplikasi .....	99
Lampiran 7 Kode Program Keseluruhan Sistem.....	109
Lampiran 8 Bukti Originalitas .....	110
Lampiran 9 Kartu Bimbingan .....	112
Lampiran 10 Biodata Penulis .....	113



UNIVERSITAS  
**Dinamika**

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Seiring dengan perkembangan teknologi dan meningkatnya kebutuhan akan efisiensi dalam kehidupan sehari-hari, sistem Smart Home semakin banyak digunakan. Sistem otomatisasi ini bertujuan untuk meningkatkan kenyamanan, efisiensi, serta keamanan bagi penggunanya. Salah satu aspek penting dalam sistem ini adalah pengelolaan perangkat rumah tangga secara otomatis, terutama dalam menghadapi perubahan lingkungan, seperti kondisi cuaca.

Salah satu faktor tumbuh dan berkembangnya tanaman yaitu dengan proses penyiraman. Penyiraman dapat menjaga serta merawat tanaman agar tumbuh dengan subur. Kebutuhan air yang cukup sangat penting pada tanaman, sehingga perlu dilakukan monitoring dalam proses penyiraman untuk menjaga agar penyiraman berjalan optimal. Ada beberapa hal yang harus diperhatikan dalam melakukan monitoring penyiraman tanaman, diantaranya adalah kelembapan tanah (Nabil Azzaky, 2020).

Selain itu, kenyamanan dalam ruangan juga merupakan salah satu faktor yang diperhatikan dalam Smart Home. Pemantauan suhu ruangan secara real-time memungkinkan pengguna untuk mengetahui kondisi lingkungan dalam rumah. Dengan informasi ini, pengguna dapat mengontrol perangkat seperti kipas angin dan lampu secara lebih efisien.

Proyek ini bertujuan untuk membangun sebuah sistem otomatisasi perangkat Smart Home berbasis Node.js dengan komunikasi WebSocket antara server dan Perangkat IoT, serta API untuk komunikasi dengan aplikasi berbasis Expo. Dengan sistem ini, pengguna dapat dengan mudah mengontrol perangkat rumah tangga melalui aplikasi Android, baik secara otomatis maupun manual, sehingga meningkatkan efisiensi penggunaan energi dan memberikan kemudahan dalam mengelola perangkat.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka disampaikan rumusan masalah sebagai berikut:

1. Bagaimana sistem dapat menentukan kapan tanaman membutuhkan penyiraman?
2. Bagaimana sistem dapat memantau suhu ruangan secara real-time?

## 1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah, maka terdapat beberapa Batasan masalah sebagai berikut:

1. Data dari pengukuran suhu dan kelembapan ruangan hanya akan ditampilkan di aplikasi tanpa adanya kontrol otomatis terhadap kipas atau AC.
2. Perangkat yang dapat dikontrol melalui aplikasi hanya sebatas alat yang mempunyai state on/off.
3. Saat koneksi internet tidak tersedia, sistem hanya menggunakan pengukuran kelembapan tanah untuk otomatisasi penyiraman.

## 1.4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut

1. Mengembangkan sistem yang dapat mengontrol penyiraman tanaman secara otomatis berdasarkan kelembapan tanah dan prediksi cuaca.
2. Merancang aplikasi yang dapat menampilkan suhu dan mengontrol peralatan di rumah.

## 1.5 Manfaat

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Meningkatkan Efisiensi dan Kemudahan dalam mengelola Rumah dan Taman sehingga dapat menghemat energi.
2. Pengguna dapat memantau suhu ruangan dan kondisi kelembapan tanah secara langsung dari aplikasi sehingga membantu dalam pengambilan keputusan.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Smart Home**

Smart Home atau rumah pintar adalah salah satu dari IoT project yang dapat memberikan manfaat bagi penggunanya untuk mengontrol rumah dari jarak jauh menggunakan sistem cloud. Sistem cloud ini memungkinkan untuk mengakses (memantau atau mengendalikan) perangkat rumah kapan saja dan di mana saja. Penerapan IoT pada sistem rumah cerdas dapat dimanfaatkan untuk pengendalian dan pemantauan perangkat elektronik (Aryani Rombekila, 2022).

Selain itu sistem Smart Home dapat digunakan sebagai sarana untuk menghemat sumber daya dengan melakukan pengoptimalan pada penggunaan listrik dan air berdasarkan kondisi lingkungan. Pada proyek ini, sensor kelembapan digunakan untuk mendeteksi kelembapan pada tanah, sehingga penyiraman hanya dilakukan saat dibutuhkan. Hal ini dapat membantu menghemat air tetapi tetap menjaga kebutuhan air tumbuhan terpenuhi.

#### **2.2 *Internet of Things* (IoT)**

*Internet of Things* atau IoT mengacu pada jaringan perangkat fisik yang berfungsi untuk mengumpulkan berbagai data dan terhubung melalui internet. IoT memungkinkan berbagai sensor dan aktuator untuk berkomunikasi tanpa adanya campur tangan dari manusia.

Teknologi ini banyak diterapkan dalam berbagai bidang, seperti otomasi rumah, Kesehatan, industry, dan pertanian. Sistem IoT meliputi sensor yang mengumpulkan data, konektivitas jaringan yang mengumpulkan data, dan sistem pemrosesan yang berfungsi untuk menganalisis data yang diperoleh dan mengambil tindakan berdasarkan data tersebut. Dalam proyek ini sistem yang dibangun meliputi monitoring suhu ruang dan otomatisasi penyiraman tanaman berdasarkan, kelembapan tanah dan prakiraan cuaca, juga pengontrol nyala mati lampu menggunakan aplikasi mobile.

### 2.3 Node.js

Node.js adalah lingkungan runtime JavaScript yang berbasis pada mesin V8 milik Google Chrome. Node.js memungkinkan pengguna menjalankan kode JavaScript di sisi server. Node.js banyak digunakan dalam pengembangan aplikasi berbasis web, API, serta proyek berbasis IoT.

Keunggulan utama dari Node.js adalah memungkinkannya untuk menjalankan pemrosesan secara asinkron karena menggunakan arsitektur event-driven dan non-blocking. Hal ini membuat Node.js dapat menangani banyak koneksi secara simultan secara efisien dengan penggunaan sumber daya rendah, hal ini dapat mengurangi beban kinerja server. Ekosistem dalam Node.js juga sangat kaya dengan adanya NPM atau Node Package Manager yang menyediakan banyak modul siap pakai, hal ini memungkinkan pengguna untuk menambahkan fitur atau fungsi di program tanpa harus membuatnya lagi dari awal.

### 2.4 Capacitor.js

Capacitor.js adalah sebuah *cross-platform native runtime* yang dikembangkan oleh Ionic, *framework* ini dirancang untuk memungkinkan pengembangan aplikasi web modern yang dapat dijalankan sebagai aplikasi *mobile*, *native*, *desktop*, ataupun web. Capacitor bertindak sebagai jembatan antara teknologi web seperti penggunaan HTML, Javascript, dan CSS dengan fitur native pada perangkat seperti penggunaan kamera, file sistem, lokasi, notifikasi, dan lain sebagainya.

Capacitor menyediakan arsitektur yang lebih modern dibanding pendahulunya dengan pendekatan *native-first*. Ia mendukung penggunaan API modern berbasis *promise*, serta memiliki struktur proyek yang memisahkan kode native dan web secara lebih terstruktur.

Keunggulan dari Capacitor antara lain adalah dukungan terhadap JavaScripts murni, dimana tidak memaksa penggunaan framework khusus seperti React, Angular, Vue, dan lainnya sehingga lebih fleksibel. Capacitor juga memberikan akses ke native program yang dapat dimodifikasi langsung dengan Android Studio atau Xcode.

## 2.5 Ionic Framework

Ionic adalah framework *open-source* untuk membangun antarmuka pengguna aplikasi berbasis web yang dapat berjalan secara *cross-platform*, terutama untuk aplikasi mobile. Ionic dibangun menggunakan teknologi web standar seperti HTML, CSS, dan JavaScript yang sering dikombinasikan dengan framework *front-end* lain seperti React dan Vue.

Ionic menyediakan banyak koleksi komponen antarmuka yang responsif seperti tombol, *tab*, *modal*, *input*, dan lain sebagainya yang membuat aplikasi berbasis web dapat terlihat seperti aplikasi mobile native. Keunggulan dari Ionic antara lain adalah *Toolkit* antarmuka yang lengkap dan modern, Integrasi *seamless* dengan Capacitor untuk akses fitur native, juga dukungan komunitas yang besar juga dokumentasi yang baik sehingga memudahkan dalam pengembangan aplikasi.

## 2.6 PostgreSQL

PostgreSQL merupakan sebuah sistem manajemen basis data relasional (RDBMS) dengan lisensi open-source yang pengembangannya telah dimulai sejak 1986. Sistem ini dirancang untuk memiliki kepatuhan tinggi terhadap standar SQL dan dilengkapi dengan berbagai fitur lanjutan, seperti kemampuan mendefinisikan tipe data sendiri (custom data types), fungsi agregat, serta stored procedures.

Berkat reputasi stabilitas dan keandalannya, PostgreSQL telah menjadi pilihan utama untuk beragam skala aplikasi, mulai dari proyek kecil hingga implementasi skala besar. Selain itu, PostgreSQL mengadopsi mekanisme Multi-Version Concurrency Control (MVCC) untuk manajemen konkurensi, yang memungkinkan proses baca dan tulis data oleh banyak pengguna dapat berjalan secara simultan tanpa menimbulkan konflik. Fitur ini krusial bagi aplikasi yang menuntut tingkat konsistensi data yang tinggi serta kinerja transaksi yang dapat diandalkan.

## 2.7 WebSocket

WebSocket adalah protokol komunikasi dua arah antara klien dan server melalui satu saluran TCP yang persisten. Tidak seperti HTTP yang berbasis request-

response, WebSocket melakukan komunikasi secara real-time tanpa perlu mengirim permintaan dari sisi klien.

Keunggulan dari WebSocket adalah latensi rendah dan efisien dalam komunikasi data. Dengan WebSocket, data dapat dikirim dan diterima secara langsung tanpa menunggu respon dari server. WebSocket banyak digunakan untuk aplikasi yang membutuhkan pengiriman data secara real-time seperti chat, game online, dan IoT. Dengan kemampuan tersebut, perangkat-perangkat IoT dapat berkomunikasi dengan lebih efisien, mempercepat proses pengiriman data, dan mengurangi beban server.

## 2.8 LittleFS

LittleFS adalah sebuah *filesystem* yang dirancang khusus untuk penggunaan pada mikrokontroller seperti Arduino atau ESP32. LittleFS dikembangkan oleh ARM dan dioptimalkan untuk beroperasi pada perangkat dengan ROM dan RAM yang terbatas.



Salah satu keunggulannya yaitu desain yang *fail-safe*, dimana sistem ini sangat tangguh saat dihadapkan pada kegagalan daya yang tiba-tiba. LittleFS memastikan sistem file tidak korup dan kembali ke kondisi awal sebelum kehilangan daya. Hal ini membuat LittleFS sangat cocok digunakan untuk aplikasi IoT yang dalam proyek ini digunakan untuk menyimpan konfigurasi dan status aktuator.

## 2.9 API

API atau Application Programming Interface merupakan serangkaian protokol atau alat yang memungkinkan dua perangkat lunak atau aplikasi saling berinteraksi dan berkomunikasi. API dalam kata lain menjadi perantara antara aplikasi untuk mengintegrasikan sebuah fitur tanpa harus membuat layanan baru dari nol.

API juga memberikan fleksibilitas kepada pengembang untuk menciptakan antarmuka aplikasi yang lebih sederhana. Sebab tidak perlu membuat kode baru untuk menambahkan program pada proyek yang kembangkan sehingga lebih efisien.

## 2.10 ESP32

ESP32 merupakan sebuah mikrokontroler yang dikenalkan oleh Espressif System dan merupakan penerus dari mikrokontroler ESP8266. Berbeda dengan Arduino dan ESP8266, ESP32 sudah terdapat Wi-Fi dan Bluetooth di dalamnya, yang sangat mempermudah pembuatan sistem IoT yang memerlukan koneksi nirkabel.



Gambar 2.1 ESP32

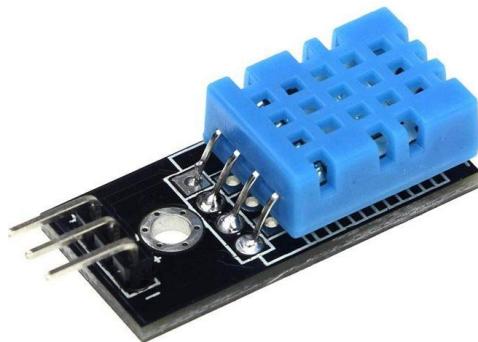
(Sumber: <https://ozami.co.id/mengenal-esp32-mikrokontroler-iot/>)

ESP32 merupakan chip yang cukup lengkap, salah satu keunggulan utama ESP32 adalah kemampuannya untuk mendukung berbagai protokol komunikasi, termasuk Wi-Fi dan Bluetooth, yang memungkinkan perangkat untuk terhubung ke internet atau berkomunikasi dengan perangkat lain secara nirkabel. Selain itu, ESP32 juga dilengkapi dengan antarmuka GPIO (General Purpose Input/Output) yang fleksibel, SPI, I2C, UART, dan ADC (Analog-to-Digital Converter), yang memungkinkannya untuk berinteraksi dengan berbagai sensor, aktuator, dan komponen elektronik lainnya.

## 2.11 DHT11

DHT11 adalah sensor suhu dan kelembapan digital yang menggunakan teknologi penginderaan kapasitif untuk mengukur kelembapan dan termistor untuk mengukur suhu, DHT11 banyak digunakan dalam berbagai aplikasi elektronik dan Internet of Things (IoT). Sensor ini mampu mengukur suhu dengan rentang 0°C

hingga 50°C dengan akurasi  $\pm 2^\circ\text{C}$ , serta kelembapan relatif dalam rentang 20% hingga 90% dengan akurasi  $\pm 5\%$ .



Gambar 2.2 Sensor DHT 11

(Sumber: <https://iotkece.com/cara-mudah-mengakses-sensor-dht-11-dengan-arduino/>)

Salah satu keunggulan DHT11 adalah kemudahan dalam penggunaannya.

Sensor ini hanya memerlukan tiga pin untuk beroperasi, yaitu pin VCC untuk daya, GND atau ground, dan DATA untuk komunikasi. Sensor ini juga tidak memerlukan komponen tambahan seperti ADC atau Analog-to-Digital-Converter untuk dapat digunakan. Meskipun memiliki keterbatasan dalam hal kecepatan pembacaan dan akurasi data, DHT11 tetap menjadi pilihan yang hemat dan mudah diimplementasikan untuk proyek-proyek sederhana.

## 2.12 Capacitive Soil Moisture Sensor

Capacitive Soil Moisture Sensor adalah jenis sensor yang digunakan untuk mengukur kadar air dalam tanah dengan prinsip kapasitansi. Berbeda dengan sensor kelembapan tanah resistif, sensor kapasitif bekerja dengan mendeteksi perubahan kapasitansi yang terjadi akibat variasi kadar air dalam tanah.



Gambar 2.3 Capacitive Soil Moisture Sensor

(Sumber: <https://how2electronics.com/interface-capacitive-soil-moisture-sensor-arduino/>)

### 2.13 API Cuaca BMKG

API Cuaca BMKG adalah platform yang disediakan oleh pemerintah Indonesia untuk memberikan akses terbuka terhadap data prakiraan cuaca yang akurat. Layanan ini bertujuan untuk memudahkan masyarakat, peneliti, developer, dan berbagai pihak yang membutuhkan data cuaca terkini dan akurat. Dengan menggunakan website atau API ini, pengguna dapat mengakses berbagai informasi seperti prakiraan cuaca harian, peringatan dini cuaca ekstrem, data iklim, dan informasi gempa bumi.

Data prakiraan cuaca dapat diakses dengan cara memanggil *endpoint* API dengan memasukkan parameter lokasi dengan akurasi tingkat desa. Misalnya ingin diambil prakiraan untuk wilayah desa Sambopinggir, maka pertama pengguna harus melihat data wilayah dari situs kodewilayah.id dan didapatkan kode 35.24.24.2015 yang merupakan kode dari desa Sambopinggir. Dan Ketika API dipanggil, maka API merespon dengan mengirimkan data prakiraan cuaca hari ini dan beberapa hari kedepan untuk wilayah desa Sambopinggir

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Studi Literatur**

Pada tahap ini, dilakukan pengumpulan dan peninjauan dari berbagai sumber informasi terkait topik dan judul penelitian yang dilakukan. Pencarian informasi dan literatur terkini, terutama informasi terkait topik penelitian sejenis dalam lima tahun terakhir. Hal ini dilakukan untuk mendapatkan wawasan yang mendalam tentang dasar teori dan variabel yang menjadi fokus penelitian. Hal ini bertujuan untuk mendapatkan metode yang tepat untuk penelitian ini.

#### **3.2 Mempersiapkan Alat dan Bahan**

##### **3.2.1 Alat**

Peralatan yang disiapkan adalah sebagai berikut:

- a. Laptop: Digunakan untuk memprogram ESP32, pengembangan aplikasi, pengujian sistem, dan lain sebagainya.
- b. Ponsel Android: Digunakan untuk memasang aplikasi yang sudah dikemas dalam format APK dan menguji fungsionalitas dan *debugging* aplikasi.
- c. Arduino IDE: Perangkat lunak untuk menulis program mikrokontroller
- d. Server: Digunakan untuk menjalankan backend aplikasi, penyimpanan data sensor, menjalankan otomatisasi, dan pemrosesan data.
- e. IP Publik: Digunakan agar mikrokontroller dan aplikasi dapat mengakses server tanpa terbatas pada jaringan lokal.
- f. Visual Studio Code: Adalah perangkat lunak penyunting kode yang digunakan untuk membuat kode program.
- g. Android Studio: Digunakan untuk mengemas aplikasi dalam format APK yang dapat dipasang pada perangkat android.
- h. Node.js: berguna untuk menjalankan kode JavaScript di sisi server untuk membangun aplikasi backend yang cepat dan efisien.

### 3.2.2 Bahan

Bahan-bahan yang disiapkan adalah sebagai berikut:

- a. ESP32: Berfungsi sebagai mikrokontroller yang mengontrol aktuator dan mengirim data sensor ke server.
- b. DHT11: Sensor ini digunakan untuk mengambil data suhu dan kelembapan pada ruangan.
- c. Capacitive Soil Moisture Sensor: Digunakan untuk mengambil data kelembapan tanah.
- d. Relay: Digunakan untuk saklar elektronik yang dikendalikan oleh mikrokontroller.
- e. Pompa Air 12V: Digunakan untuk memompa air saat diperlukan penyiraman, ini dikontrol otomatis oleh sistem atau manual dari aplikasi.
- f. Lampu: ini digunakan sebagai penerangan ruangan yang dikontrol melalui aplikasi.
- g. Adaptor: digunakan untuk suplai daya pada perangkat penyiram otomatis yang terdiri dari ESP32, Relay, dan Pompa.
- h. DC Step Down: untuk memberi daya yang stabil dan sesuai pada perangkat ESP32 dan pompa.
- i. API BMKG: Digunakan untuk mendapatkan data prakiraan cuaca yang akurat.



## 3.3 Perancangan Sistem

### 3.3.1 Arsitektur Keseluruhan Sistem

Sistem ini terdiri dari 3 komponen utama diantaranya:

1. SilverLink-Controller

SilverLink-Controller merupakan unit perantara (*bridge*) antara perangkat keras sensor/aktuator dengan server SilverLink. Dibangun di atas platform ESP32, unit ini berfungsi sebagai gateway komunikasi yang menghubungkan sensor fisik di lapangan dengan sistem kontrol pusat melalui koneksi WiFi dan protokol WebSocket.

1. Peran Utama sebagai Interface Hardware

- A. Controller membaca data dari berbagai sensor seperti DHT11 (suhu/kelembaban udara) dan sensor kelembaban tanah, kemudian

mengirimkan data tersebut ke server setiap 5 detik saat online. Pembacaan dilakukan dengan kalibrasi yang tepat untuk memastikan akurasi data.

- B. Menerima perintah kontrol dari server dan mengeksekusinya pada aktuator fisik seperti pompa, kipas, atau relay. Setiap perintah divalidasi dan dieksekusi.
- C. Menjalankan koneksi real-time dengan server silverlink.eula.my.id:5050 menggunakan protokol WebSocket untuk pertukaran data dan perintah dengan latency minimal. Sistem heartbeat memastikan koneksi tetap stabil.
- D. Mode Offline Emergency Saat koneksi terputus. controller dapat beroperasi mandiri untuk fungsi kritis seperti penyiraman otomatis berdasarkan threshold kelembaban tanah yang telah dikonfigurasi. Mode ini memastikan tanaman tetap terlindungi meskipun server tidak dapat diakses.

## 2. Sistem Mode Operasi Adaptif

Controller secara otomatis beralih antara tiga mode operasi berdasarkan status koneksi:

- Online Mode: Operasi normal dengan koneksi penuh ke WiFi dan server
- Offline WiFi Mode: Operasi emergency selama 30 menit saat WiFi terputus.
- Offline Server Mode: Operasi emergency selama 6 jam saat server tidak dapat diakses

## 3. Konfigurasi Dinamis

Setiap unit menyimpan dua jenis konfigurasi: konfigurasi dasar (WiFi dan token) dalam config.json, dan konfigurasi runtime sensor/aktuator dalam config\_runtime.json yang dapat diperbarui dari server secara real-time tanpa restart.

## 4. Setup Interface

Untuk kemudahan deployment, controller menyediakan access point SilverLink-Setup dengan web interface di 192.168.4.1 ketika belum

dikonfigurasi, serta factory reset melalui tombol fisik untuk maintenance lapangan.

## 2. SilverLink-Core

SilverLink-Core merupakan backend server berbasis Node.js yang berfungsi sebagai pusat kendali dan pemrosesan data dari seluruh ekosistem IoT smart farming. Dibangun dengan arsitektur modular dan scalable, sistem ini mengelola multiple controller ESP32, mengintegrasikan data cuaca real-time, dan menjalankan algoritma otomatisasi penyiraman cerdas dengan dukungan timezone yang tepat untuk wilayah Indonesia.

### 1. Peran Utama sebagai Sistem Manajemen Terpusat

- REST API Server menyediakan 10 endpoint utama untuk integrasi dengan aplikasi mobile di port 3000. Endpoint dapat dilihat di tabel 3.1 sebagai berikut:

Tabel 3.1 Endpoint REST API

Fungsi	Base Endpoint	Deskripsi
Otentikasi	/api/auth	Sistem autentikasi dan manajemen user
Kontroller	/api/controllers	Manajemen perangkat ESP32 controller
Sensor	/api/sensors	Konfigurasi dan monitoring sensor
Aktuator	/api/actuators	Kontrol aktuator dan manual watering
Penjadwalan	/api/schedules	Penjadwalan penyiraman otomatis
Log	/api/logs	Sistem logging dan analytics
Cuaca (BMKG)	/api/bmkg	Integrasi API cuaca Indonesia
Pin Perangkat	/api/pins	Manajemen pin ESP32
Timezone	/api/timezone	Utilitas timezone Indonesia
Kesehatan Sistem	/health, /api/status	Monitoring kesehatan sistem

- WebSocket Real-time Communication Menjalankan protokol WebSocket dengan path /ws di port 5050 untuk komunikasi real-time dengan controller ESP32. Sistem ini menangani otentikasi dengan validasi token, *sensor data processing* dengan *bulk insert* dan *timezone awareness*, kontrol aktuator dengan konfirmasi status, *heartbeat monitoring* setiap 60 detik dengan ping/pong timeout 45 detik, serta

*runtime configuration sync* yang dapat diperbarui tanpa restart controller.

- c. Database Management System Menggunakan PostgreSQL dengan Knex.js ORM untuk mengelola 8 tabel utama: users (multi-user authentication), controllers (device registry dengan timezone support), sensors/actuators (hardware configuration), watering\_schedules (cron-like scheduling), sensor\_logs (time-series data), watering\_logs (execution history), dan controller\_logs (system events).
  - d. Integrasi real-time dengan API BMKG ([api.bmkg.go.id](http://api.bmkg.go.id)) untuk prakiraan cuaca Indonesia. Algoritma prediksi cuaca menganalisis *confidence level, precipitation amount, weather description keywords, dan humidity patterns* untuk menghasilkan rekomendasi penyiraman.
- Mendukung timezone *auto-sync*:

- o WIB (Asia/Jakarta)
- o WITA (Asia/Makassar)
- o WIT (Asia/Jayapura).

## 2. Sistem Penyiraman Otomatis

- a. Service berjalan setiap menit untuk memantau semua controller dengan auto watering enabled. Sistem memeriksa jadwal yang match dengan waktu saat ini dalam timezone masing-masing controller, memvalidasi *hardware availability* (soil sensors dan pump actuators), dan melakukan *execution tracking* untuk mencegah duplikasi.
- b. Setiap eksekusi penyiraman melalui 5 tahap validasi:
  - o *controller eligibility check* (online, auto watering enabled, WebSocket connected)
  - o *schedule matching* (timezone-aware time comparison)
  - o *hardware validation* (soil sensors dengan threshold, pump actuators available)
  - o *soil moisture analysis* (rata-rata 30 pembacaan terakhir vs threshold)
  - o *weather forecast analysis* (rain confidence >50% atau precipitation >0.5mm dalam 3 jam ke depan).

### 3. Keamanan dan Performa

- a. Multi-layer Authentication menggunakan JWT dengan access token (7 hari) dan refresh token (30 hari), *bcrypt password hashing* dengan salt 10 rounds, rate limiting 1000 requests per 15 menit per IP, dan CORS configuration untuk mobile apps (Capacitor, Ionic, React Native).
- b. Penanganan Shutdown yang baik dengan *WebSocket cleanup, uncaught exception logging* tanpa crash di production, database connection pooling dengan *retry mechanism*, dan *weather service fallback* ketika BMKG API tidak tersedia.

### 4. Penerapan dan Konfigurasi

- a. Dukungan PM2 untuk *production deployment* dengan *cluster mode, automatic restart on crash, log rotation, dan memory monitoring*.
- b. Modular route system yang mudah dikembangkan, *service layer separation* untuk *business logic, configurable timeouts dan limits, dan database indexing* untuk *optimal query performance*. Sistem mendukung *horizontal scaling* dengan *shared database*.

### 3. SilverLink: Connect

SilverLink: Connect merupakan aplikasi mobile berbasis Capacitor.js dengan React Framework yang berfungsi sebagai interface utama untuk manajemen dan monitoring sistem IoT SilverLink. Aplikasi ini menyediakan kontrol penuh terhadap controller, sensor, dan aktuator melalui komunikasi API dengan server, serta memungkinkan pengaturan penyiraman otomatis dan visualisasi data real-time.

#### 1. Peran Utama sebagai Interface Management

- a. Aplikasi menyediakan dashboard terpusat untuk manajemen multiple controller dengan 4 mode operasi (watering, remote\_switch, temperature\_monitoring, full\_custom). Setiap mode memiliki konfigurasi preset yang sesuai dengan kebutuhan spesifik. Khusus mode full\_custom, user dapat memilih sendiri pin yang digunakan dan dapat mengaktifkan sistem penyiram saat diperlukan.
- b. Melakukan komunikasi API dengan server SilverLink melalui HTTP untuk operasi CRUD (Create, Read, Update, Delete) data controller,

sensor, dan aktuator. Sistem auto-refresh memastikan data selalu terupdate setiap 30 detik untuk status controller dan 15 detik untuk data sensor.

- c. Menyediakan interface manual override untuk kontrol aktuator secara langsung dengan fitur *duration control* untuk mencegah operasi yang tidak aman.

## 2. Sistem Mode Controller Adaptif

Aplikasi mendukung empat mode controller yang dapat dipilih saat setup:

- Watering Mode: Auto-setup sensor kelembaban tanah (pin 34) dan pompa air (pin 17)
- Remote Switch Mode: Auto-setup 4 relay pada pin 16, 17, 18, 19 untuk kontrol perangkat jarak jauh
- Temperature Monitoring Mode: Auto-setup sensor DHT11 (pin 16) untuk monitoring suhu dan kelembaban
- Full Custom Mode: Konfigurasi manual sensor dan aktuator sesuai kebutuhan spesifik

## 3. Interface Monitoring dan Analytics

- a. Visualisasi data sensor dalam bentuk grafik real-time dengan Chart.js, menampilkan tren historis selama 1 hari, 1 minggu, atau 1 bulan.
- b. *Weather Page* menampilkan hasil prakiraan cuaca 3 jam kedepan.

## 4. Manajemen User Experience

- a. Authentication system dengan JWT token dan auto-login capability untuk kemudahan akses berulang. Sistem refresh token memastikan sesi tetap aktif selama penggunaan normal.
- b. Responsive design yang optimal untuk perangkat mobile dengan dark theme yang terinspirasi dari karakter Silver Wolf, menggunakan color scheme ungu-silver yang konsisten.
- c. Backend connectivity monitoring dengan automatic retry mechanism dan offline state indication ketika server tidak dapat diakses.

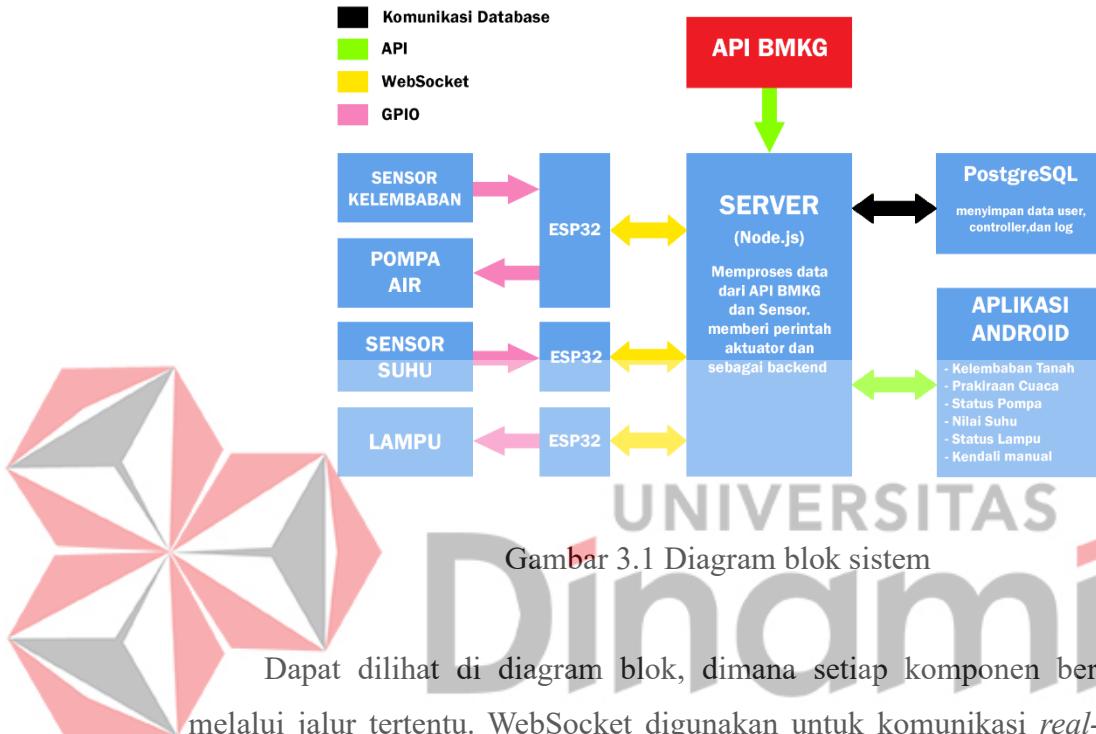
## 5. Setup dan Deployment Workflow

Aplikasi menyediakan panduan setup untuk :

- Pemilihan mode controller dengan preview konfigurasi hardware

- BMKG location search dan selection untuk *weather integration*
- Pembuatan token dan mekanisme salin untuk konfigurasi ESP32
- Validasi setup untuk memastikan controller terhubung dengan benar

Arsitektur sistem ini dirancang dengan prinsip scalability, reliability, dan maintainability. Dimana setiap komponen berkomunikasi melalui jalur yang disusun rapi agar memudahkan dalam pengoperasian.



Dapat dilihat di diagram blok, dimana setiap komponen berkomunikasi melalui jalur tertentu. WebSocket digunakan untuk komunikasi *real-time* antara controller dan *server*, sementara REST API digunakan untuk komunikasi antara aplikasi mobile dan server. Sistem juga mengintegrasikan API BMKG untuk data cuaca yang digunakan dalam pembuatan keputusan untuk penyiraman otomatis bersamaan dengan data kelembapan tanah.

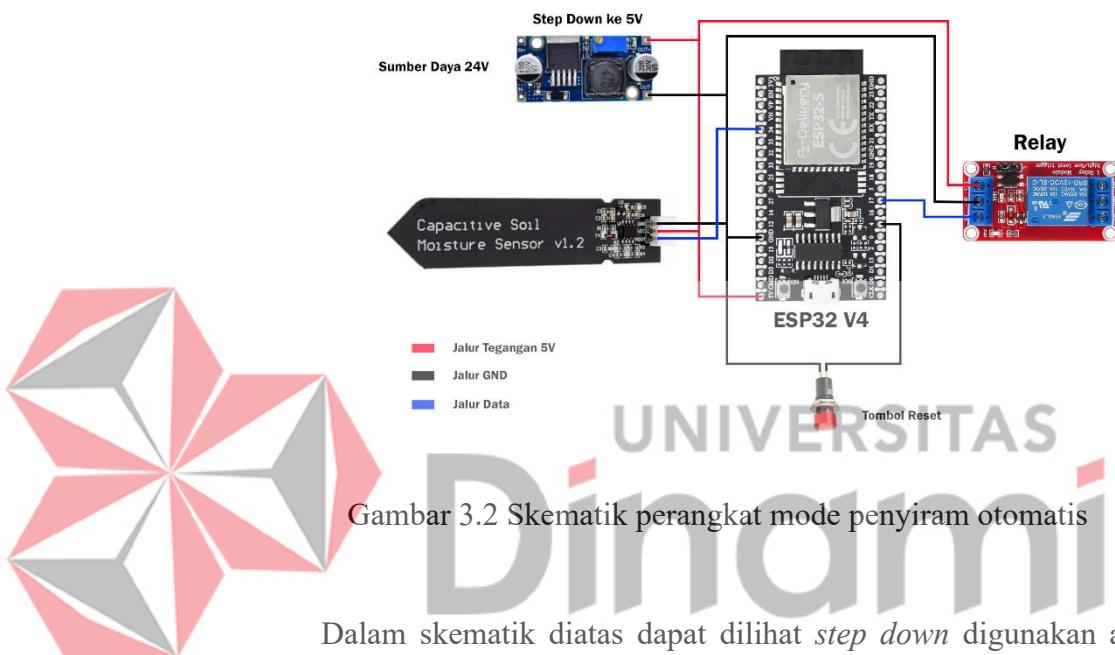
### 3.3.2 Perancangan Perangkat Keras

Perangkat Keras dalam sistem ini menggunakan ESP32 sebagai controller utama dengan 2 jenis sensor yang dapat dipasang yaitu DHT11 untuk mengukur suhu dan kelembapan lingkungan, juga sensor Soil Moisture untuk mengukur kelembapan tanah. Untuk aktuator tidak ada Batasan penggunaan dimana semua perangkat yang mempunyai state on/off dapat dikendalikan oleh sistem ini. User dapat membuat perangkat keras custom yang dapat disesuaikan dengan kebutuhan

semisal dalam 1 controller dipasang 1 relay dan 1 DHT11. Dalam proyek ini, penulis menggunakan 3 jenis hardware yang sudah disesuaikan sebagai berikut.

a. Perangkat mode Penyiram Otomatis

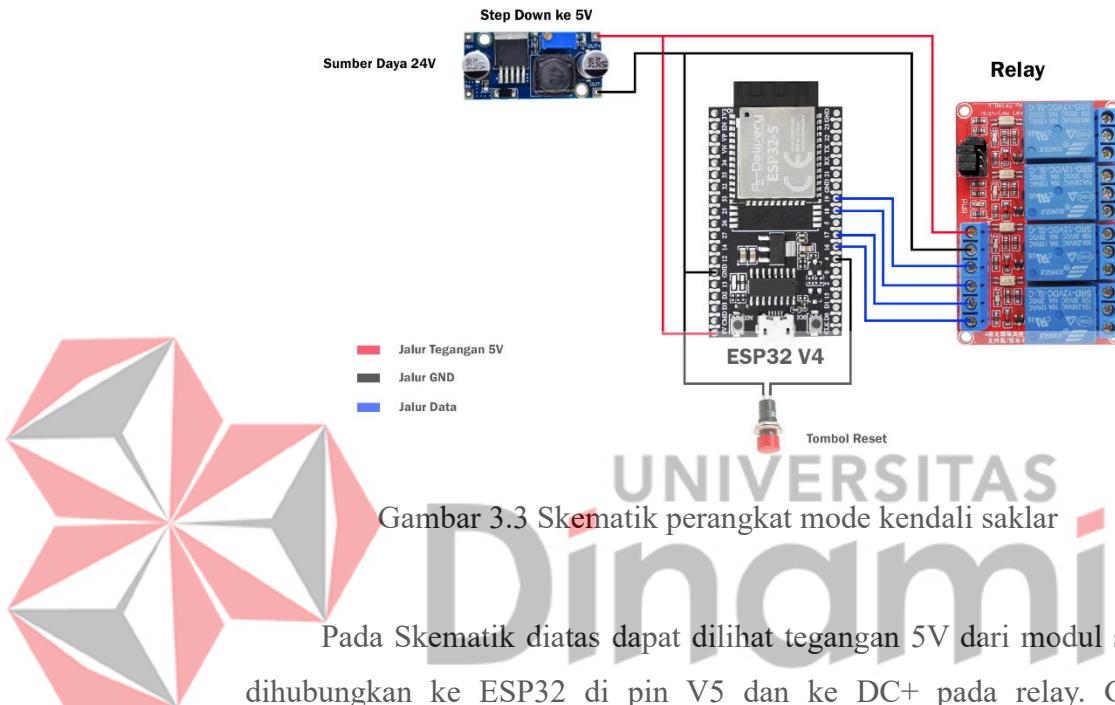
Mode ini dirancang sebagai solusi sistem irigasi pintar untuk aplikasi pertanian atau taman. Sistem ini mengintegrasikan *Capacitive Soil Moisture Sensor* dengan Pompa air yang dikontrol berdasarkan tingkat kelembapan tanah dan hasil prediksi cuaca.



Dalam skematik diatas dapat dilihat *step down* digunakan agar dapat menyediakan daya listrik yang stabil untuk menghidupkan ESP32, Sensor Kelembapan, dan Relay. *Step down* menurunkan tegangan menjadi 5V sesuai dengan spesifikasi komponen, kemudian arus VCC dialirkan menuju ESP32 di pin V5, pin VCC sensor, dan DC+ pada Relay. Pin GND juga dihubungkan sesuai label pada komponen. Pin output sensor kemudian dihubungkan ke Pin 34 pada ESP32 yang merupakan pin input yang dapat digunakan untuk membaca nilai ADC dari sensor. Sedangkan input relay dihubungkan ke Pin 17 pada ESP32 yang diatur sebagai pin aktuator atau output. Kemudian saklar pompa dapat disambung ke pin COM dan NO (*Normally Open*) pada Relay. Pin 4 digunakan untuk reset perangkat ke mode setup dengan cara dihubungkan ke GND atau menekan tombol reset saat ESP32 dinyalakan.

b. Perangkat mode Kendali Saklar

Mode ini dirancangan untuk aplikasi *Home Automation* yang memungkinkan kontrol perangkat listrik semisal lampu dan kipas dari jarak jauh menggunakan aplikasi mobile. Sistem ini dapat mengontrol 4 perangkat secara independen per 1 modul kontroler, jadi user dapat mengendalikan 8 perangkat jika mempunyai 2 modul ini.



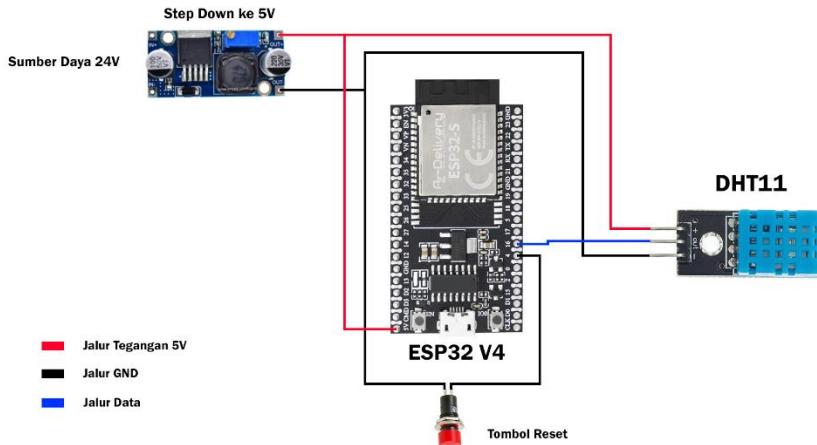
Gambar 3.3 Skematik perangkat mode kendali saklar

Pada Skematik diatas dapat dilihat tegangan 5V dari modul step down dihubungkan ke ESP32 di pin V5 dan ke DC+ pada relay. GND juga dihubungkan ke pin GND pada ESP32 dan pin DC- pada relay. Kemudian pin 16, 17, 18,dan 19 pada ESP32 dihubungkan masing-masing ke 1 channel pada relay 4 channel, pin tersebut semuanya merupakan pin mode output untuk mengontrol relay. Kemudian pin COM dan NO (*Normally Open*) pada relay dapat dihubungkan dengan perangkat elektronik lain yang dikontrol. Pin 4 digunakan untuk reset perangkat ke mode setup dengan cara dihubungkan ke GND atau menekan tombol reset saat ESP32 dinyalakan.

c. Perangkat mode Pengukur Suhu dan Kelembapan

Mode ini dirancang sebagai sensor pengukur suhu dan kelembapan yang memantau keadaan lingkungan dengan konsumsi daya minimal dan akurasi data tinggi baik di *indoor* maupun *outdoor*. Perangkat ini memungkinkan pengguna memantau suhu dan kelembapan lingkungan dengan sensor DHT11

yang dapat mengukur suhu di batas 0-50°C dengan akurasi  $\pm 2^\circ\text{C}$  dan kelembapan di batas 20-90% RH dengan akurasi  $\pm 5\%$ .



Gambar 3.4 Skematic perangkat mode pengukuran suhu dan kelembapan

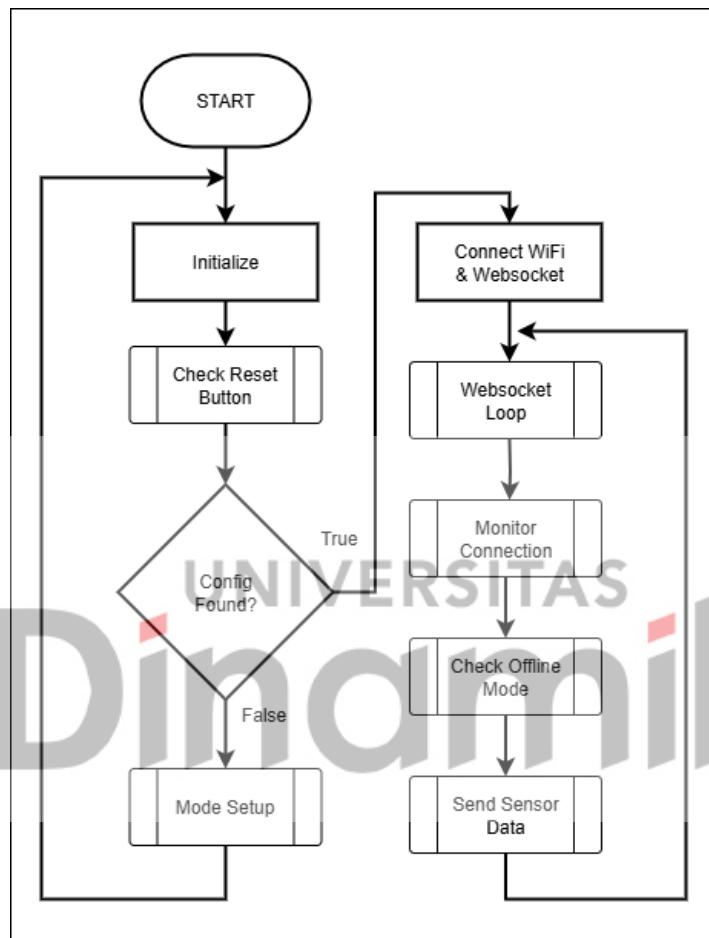
Pada skematic diatas dapat dilihat tegangan 5V dari modul step down dihubungkan ke ESP32 di pin V5 dan ke + pada DHT11. GND juga dihubungkan ke pin GND pada ESP32 dan pin - pada DHT11. Kemudian pin 16 pada ESP32 dihubungkan ke pin out pada DHT1, pin 16 merupakan pin *multi mode* yang dibutuhkan untuk membaca sensor DHT11 yang membutuhkan *read* dan *write* data. Pin 4 digunakan untuk reset perangkat ke mode setup dengan cara dihubungkan ke GND atau menekan tombol reset saat ESP32 dinyalakan.

### 3.3.3 Perancangan Perangkat Lunak

#### a. Firmware ESP32 (SilverLink-Controller)

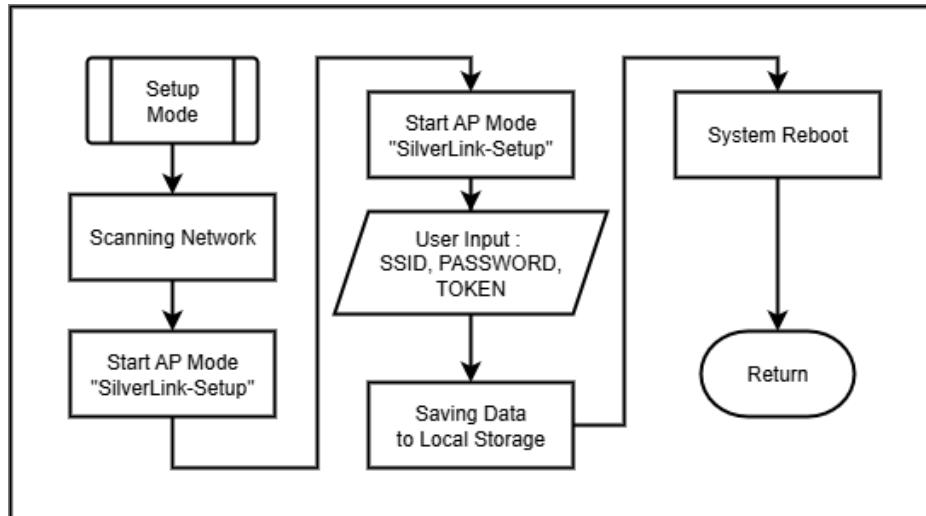
Firmware ESP32 ini dirancang dengan arsitektur modular yang memungkinkan banyak sensor dan aktuator yang berbeda berjalan dalam basis kode yang sama. Arsitektur ini menggunakan Token unik yang didapat dari server untuk identifikasi setiap kontroler, hal ini memungkinkan kontroler menjalankan berbagai fungsi sekaligus hanya dengan satu program yang sama dengan cara ESP32 meminta server untuk mengirim data alokasi pin yang digunakan oleh user untuk sensor dan aktuator. Dengan arsitektur seperti ini, kontroler

memungkinkan untuk mengganti fungsi pin tanpa harus flash firmware dan fungsi pin dapat disesuaikan langsung pada aplikasi mobile oleh user. Tombol reset juga dipasang pada pin 4 untuk menghapus konfigurasi WiFi, token, dan alokasi pin.



Gambar 3.5 Flowchart Loop ESP32

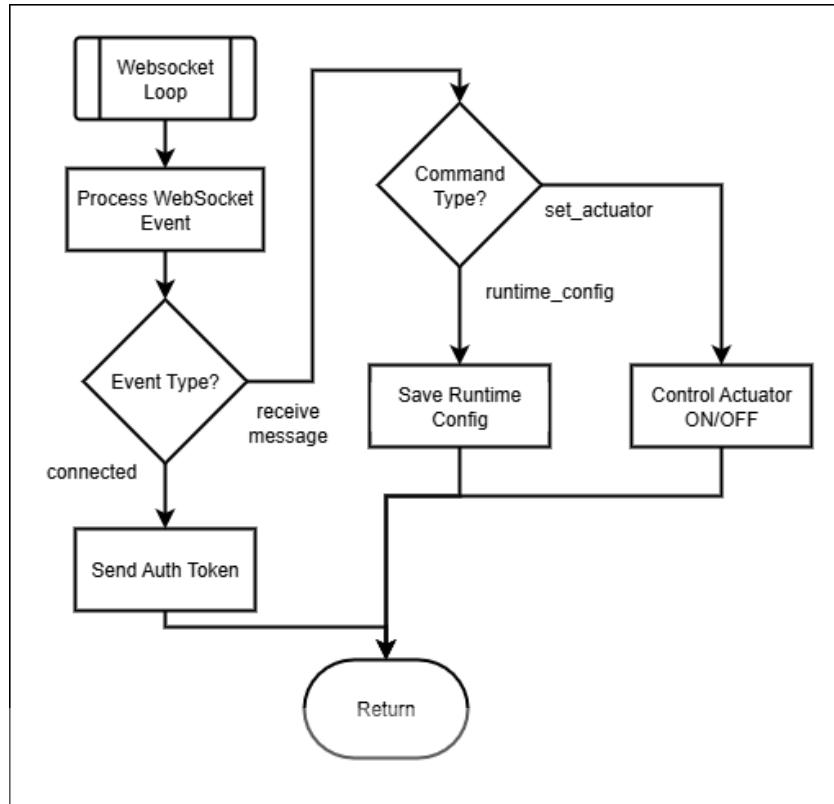
Pada Flowchart pada Gambar 3.5 dapat dilihat saat ESP32 dinyalakan, maka program akan langsung memuat file konfigurasi, jika konfigurasi tidak ditemukan, maka langsung masuk ke Mode Setup. Saat di mode Setup, ESP32 menerima data dari User berupa SSID, Password, dan Token. Setelah data dimasukkan, ESP32 melalukan reboot sistem.



Gambar 3.6 Flowchart Setup Mode

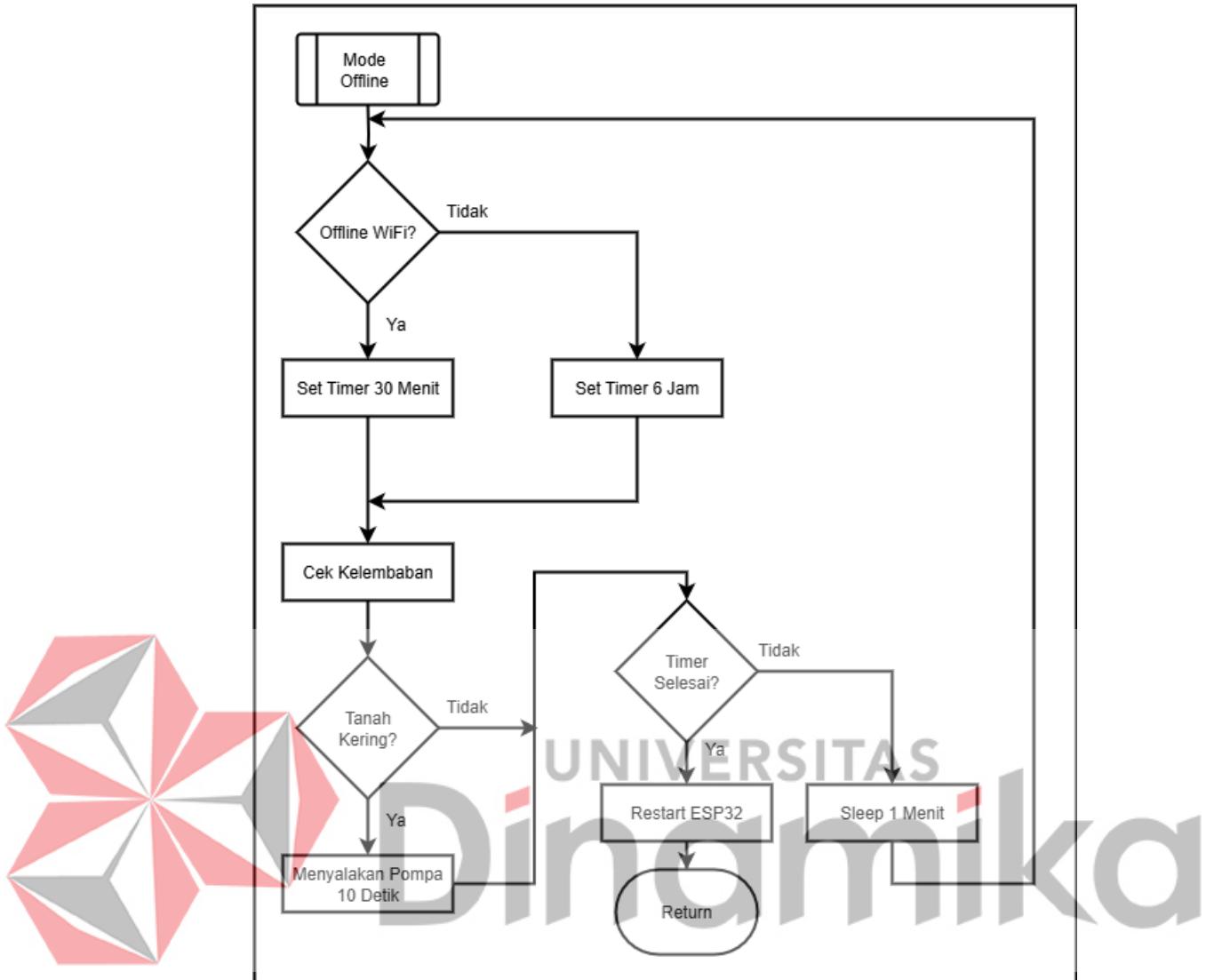
Saat setup Mode, ESP32 scan semua hotspot WiFi kemudian mencatat semua SSID yang terdeteksi dalam list yang ditampilkan di Dropdown pada Web sehingga dapat dengan mudah dipilih oleh user. Setelah semua SSID WiFi berhasil dideteksi, ESP32 memancarkan hotspot WiFi dengan SSID “SilverLink-Setup” dan mengaktifkan Web Server untuk menampilkan web untuk user memasukkan data WiFi dan Token. Saat user klik tombol save, maka data WiFi dan Token disimpan pada file json di penyimpanan lokal ESP32 dan diload setiap ESP32 dinyalakan. Flowchart Setup Mode dapat dilihat di Gambar 3.6.

Saat config berhasil ditemukan, maka ESP32 mencoba terkoneksi ke jaringan WiFi dan ke server melalui WebSocket menggunakan data yang sebelumnya sudah di masukkan oleh user. Jika berhasil terkoneksi, maka ESP32 lanjut ke Loop utama.



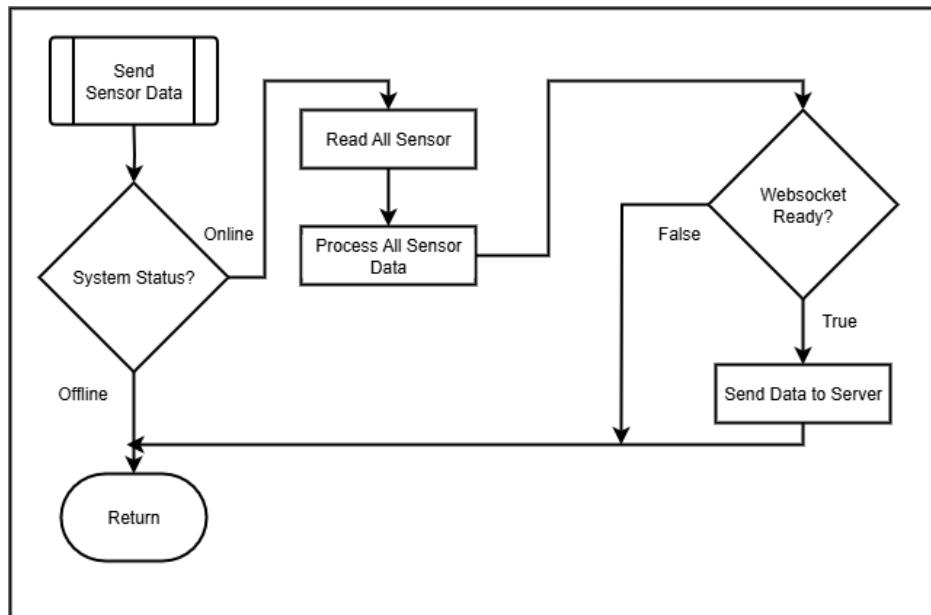
Gambar 3.7 Flowchart WebSocket Loop

Pada Loop Websocket, pertama-tama menerima Event, kemudian di cek tipe event yang diterima. Jika event connected, maka dikirim token autentikasi ke server. Jika event adalah pesan diterima, maka dicek command yang diterima, jika runtime\_config, maka ESP32 menyimpan konfigurasi runtime yang dikirim oleh server. Jika yang diterima set\_actuator, maka ESP32 update status dari aktuator sesuai yang diperintah oleh server. Flowchart fungsi WebSocket Loop dapat dilihat pada Gambar 3.7.



Gambar 3.8 Flowchart Mode Offline

Saat hasil cek koneksi dinyatakan tidak terhubung selama beberapa kali percobaan, sistem masuk ke Mode Offline. Di mode ini pertama di cek apakah offline WiFi, jika iya, maka di set timer di 30 menit, jika tidak, maka di set timer di 6 jam. Selanjutnya dilanjutkan dengan cek kelembaban tanah, selanjutnya dicek apakah tanah kering dengan cara dibandingkan dengan data threshold di file runtime yang diperoleh saat masih terhubung di server. Jika kering, maka pompa dinyalakan selama 10 detik, dan jika tidak, maka lanjut pengecekan timer. Jika timer selesai, maka ESP32 restart. Jika tidak, maka ditunggu selama 5 menit sebelum cek kelembaban kembali. Flowchart Mode Offline dapat dilihat di Gambar 3.8.



Gambar 3.9 Flowchart Send Sensor Data

Selanjutnya adalah fungsi Send Sensor Data. Fungsi ini digunakan untuk mengirim data hasil deteksi sensor ke server melalui WebSocket. Yang pertama dilakukan pengecekan status sistem, jika offline, maka langsung dilewati. Jika Online, maka ESP32 membaca semua sensor yang ada, lanjut memproses data sensor seperti mapping ADC pada soil moisture. Setelah data selesai diproses, dicek lagi status websocket apakah siap digunakan, jika siap, maka data dikirim ke server dengan format JSON dan program kembali ke loop awal.

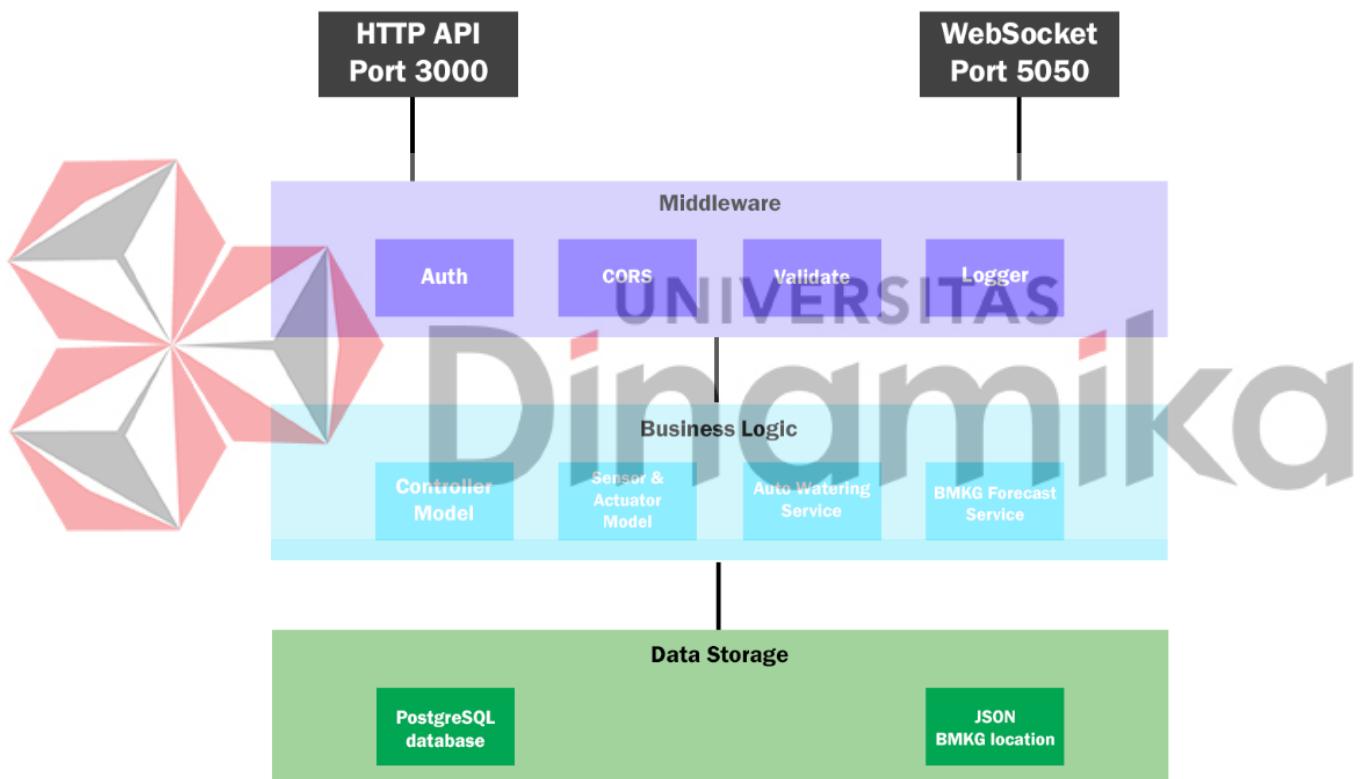
Tabel 3.2 Library ESP32

Library	Fungsi
WiFi.h	Menghubungkan ESP32 ke jaringan WiFi, baik sebagai Station maupun Access Point.
ESPAsyncWebServer.h	Membuat antarmuka web untuk mode setup
LittleFS.h	Untuk menyimpan konfigurasi runtime di internal
ArduinoJson.h	Parsing dan pembuatan JSON pada sistem kirim dan terima data dari server.
WebSocketsClient.h	Menghubungkan ESP32 ke server WebSocket untuk komunikasi real-time dua arah.
DHT.h	Membaca data suhu dan kelembaban dari sensor DHT11/DHT22

b. Program Backend (SilverLink-Core)

Program ini dirancang diatas runtime Node.js untuk mengakomodasi otomatisasi penyiraman, integrasi API eksternal, dan penyimpanan data Riwayat pengukuran, dan pengambilan keputusan. Program ini menggunakan Node.js sebagai runtime untuk menjalankan sistemnya. Program ini membutuhkan IP publik agar kontroller dan aplikasi mobile dapat terhubung dengan kebutuhan 2 port yaitu 3000 untuk http API dan 5050 untuk WebSocket.

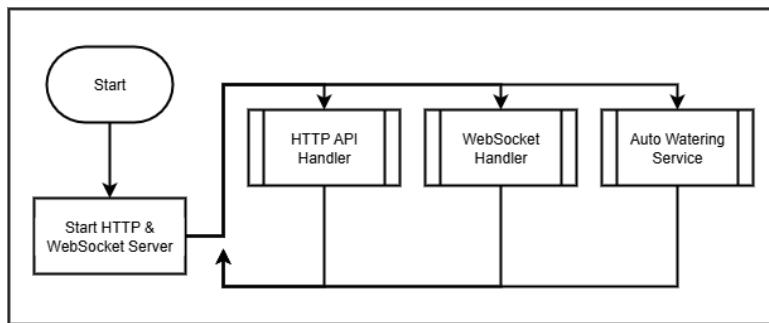
A. Arsitektur Sistem



Gambar 3.10 Diagram Arsitektur Backend

Sistem ini mengimplementasikan arsitektur berlapis yang terdiri dari empat layer utama yaitu Presentation Layer, Middleware Layer, Business Logic, dan Data Layer. Sistem ini menggunakan 2 server terpisah yaitu HTTP Server di port 3000 untuk komunikasi dengan aplikasi mobile dan

WebSocket server di port 5050 untuk berkomunikasi dengan perangkat ESP32.

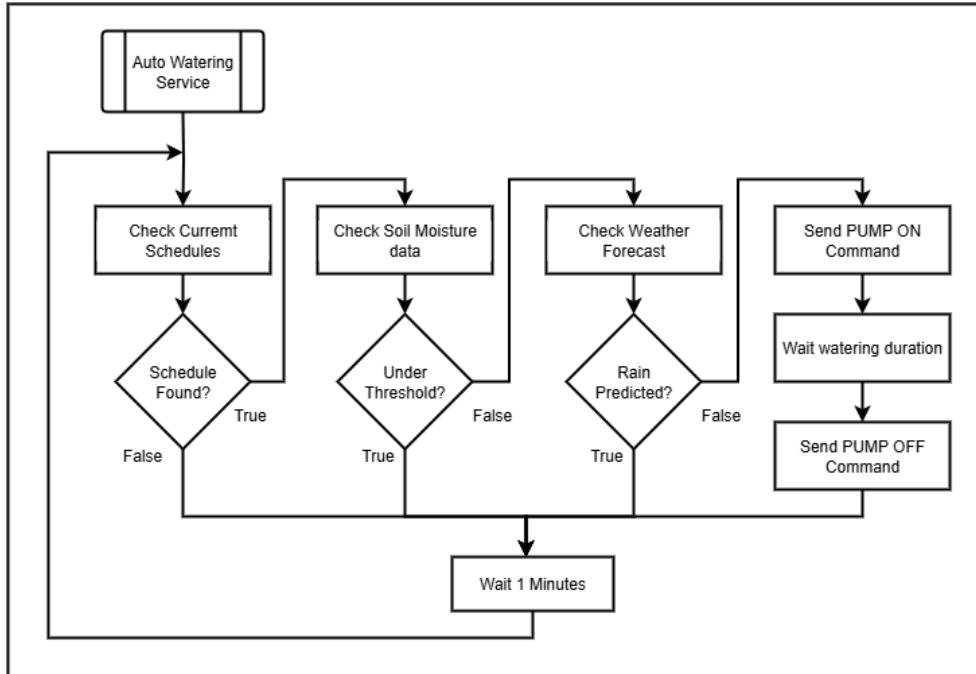


Gambar 3.11 Flowchart Start Server

Flowchart pada Gambar 3.11 menunjukkan saat program SilverLink-Core dijalankan, maka dimulai dengan mengaktifkan server HTTP di Port 3000 dan server WebSocket di port 5050. Setelah kedua server aktif, sistem menjalankan fungsi HTTP API Handler, WebSocket Handler dan Cron Job Auto Watering secara bersamaan.

HTTP API Handler menangani semua operasi CRUD (Create, Read, Update, Delete) melalui endpoint REST API. Komponen utama meliputi: Sistem otentikasi berbasis JWT (*Json Web Token*) untuk keamanan dan identifikasi user, manajemen kontroler, konfigurasi sensor dan aktuator, sistem penjadwalan otomatis, dan sistem logging dan analytics untuk pelacakan aktivitas sistem. Flowchart HTTP API Handler dapat dilihat di Lampiran 1.

WebSocket server berfungsi sebagai layer komunikasi yang menangani koneksi dengan perangkat ESP32. Server ini menangani penerimaan data sensor, kontrol aktuator realtime, dan update runtime untuk mengatur pin dan status aktuator. Flowchart Websocket Handle dapat dilihat di Lampiran 2.



Gambar 3.12 Flowchart penyiraman otomatis



*Auto Watering Service* merupakan fungsi utama yang mengimplementasikan algoritma penyiraman otomatis yang didasarkan pada hasil pengukuran kelembapan tanah dan hasil prediksi cuaca dari BMKG. Fungsi ini bekerja setiap 1 menit dengan cara melakukan pengecekan jadwal yang terdaftar pada database, jika ada jadwal penyiraman, maka system cek 30 data terakhir yang dikirim oleh ESP32 dan menjadikannya rata-rata. Jika hasil rata-rata ini berada di bawah threshold, maka sistem lanjut melakukan pengecekan prediksi cuaca selama 3 jam kedepan, jika tidak hujan, maka penyiraman dilakukan. Flowchart dapat dilihat di Gambar 3.12.

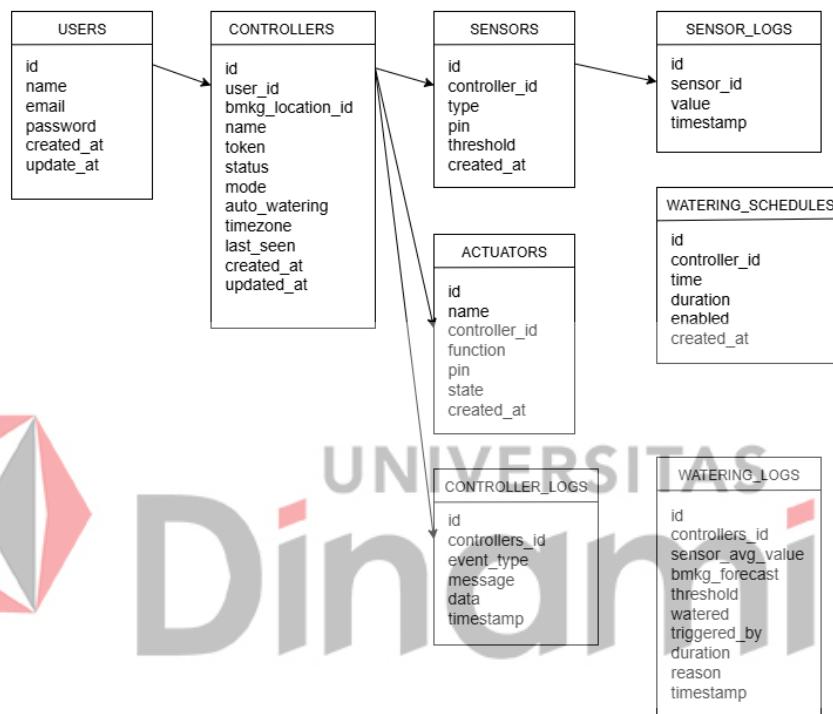
Data layer menggunakan PostgreSQL untuk database utama dengan Knex.js digunakan untuk ORM (*Object Relational Mapping*) untuk *query building*. Sistem ini juga menggunakan sistem penyimpanan data berbasis JSON untuk data yang bersifat statis seperti data lokasi daerah di indonesia untuk query API BMKG.

Komunikasi sistem ini bekerja dengan cara aplikasi mobile berkomunikasi dengan backend melalui REST API untuk operasi CRUD, backend berkomunikasi dengan ESP32 melalui WebSocket untuk kontrol

realtime dan monitoring, dan *Auto Watering Service* terintegrasi dengan BMKG API untuk mendapatkan data prediksi cuaca. ESP32 mengirimkan data sensor secara berkala ke backend, yang kemudian diproses dan disimpan dalam database untuk analytics dan pengambilan keputusan dalam sistem penyiraman otomatis.

## B. Struktur Database

Sistem ini menggunakan database PostgreSQL dengan 8 tabel utama.



Gambar 3.13 Struktur Database

1. Tabel users berfungsi menyimpan data pengguna dengan struktur: id digunakan sebagai *primary-key auto increment*, nama untuk nama pengguna, email sebagai *unique identifier* untuk login, password yang dihash dengan bcrypt, timestamp created\_at dan update\_at untuk logging.
2. Tabel controllers berisi konfigurasi untuk setiap perangkat ESP32 dengan id sebagai *primary-key*, user\_id sebagai *foreign key* ke tabel users untuk penandaan kepemilikan kontroler, bmkg\_location\_id untuk menyimpan data lokasi untuk query API BMKG, name untuk identifikasi kontroler, token sebagai unique identifier untuk otentikasi

WebSocket, status untuk penanda kontroler sedang offline atau online, mode untuk menentukan mode operasi controller (watering, remote\_switch, temp\_mon, dan full\_custom), auto\_watering\_enabled untuk menentukan apakah penyiram otomatis diaktifkan, timezone untuk pengaturan zona waktu, dan last\_seen untuk data kapan kontroler terakhir online.

3. Tabel sensors menyimpan konfigurasi sensor yang terpasang pada kontroler dengan struktur : id sebagai *primary key*, controller\_id sebagai *foreign key*, type untuk jenis sensor (dht\_temperature, dht\_humidity, soil\_moisture), pin untuk pin ESP32 yang digunakan, threshold khusus untuk soil moisture sensor sebagai batas minimal kelembapan tanah, dan timestamp untuk logging. Tabel ini memiliki relasi *one-to-many* dengan tabel sensor\_logs untuk menyimpan data pembacaan.
4. Tabel actuators berfungsi mencatat konfigurasi aktuator dengan id sebagai *primary key*, controller\_id sebagai *foreign key*, function untuk jenis aktuator (pump, lamp, fan, relay, dll), pin untuk pin ESP32, state untuk status ON/OFF, dan timestamp untuk logging. Setiap kontroler dapat memiliki beberapa actuators dengan pin yang berbeda.
5. Tabel sensor\_logs berfungsi menyimpan data hasil pembacaan sensor dengan id sebagai *primary key*, sensor\_id sebagai *foreign key* ke tabel sensors, value untuk nilai pembacaan, timestamp untuk waktu pembacaan dalam timezone controller, dan created\_at untuk waktu log dibuat.
6. Tabel watering\_schedules Berisi jadwal penyiraman otomatis dengan id sebagai *primary key*, controller\_id sebagai *foreign key*, time dalam format HH:MM untuk waktu eksekusi penyiraman, duration dalam detik untuk lama penyiraman, enabled boolean untuk status aktif/nonaktif, dan timestamp untuk tanda waktu jadwal dibuat. Scheduler menggunakan tabel ini untuk menentukan kapan harus melakukan pengecekan kondisi tanah dan cuaca untuk penyiraman.

7. Tabel watering\_logs Menyimpan catatan aktivitas penyiraman dengan id sebagai *primary key*, controller\_id sebagai *foreign key*, sensor\_avg\_value untuk rata-rata kelembapan tanah yang diambil dari 30 data terakhir, threshold untuk batas kelembapan yang digunakan, bmkg\_forecast dalam format JSON untuk data cuaca, watered boolean untuk status eksekusi, triggered\_by untuk sumber trigger (schedule/manual/auto), duration untuk lama penyiraman, reason untuk alasan keputusan, timestamp untuk waktu kejadian, dan created\_at untuk waktu kapan log dibuat.
8. Tabel controller\_logs Mencatat seluruh aktivitas controller dengan id sebagai *primary key*, controller\_id sebagai *foreign key*, event\_type untuk kategorisasi event (online, offline, error, config\_update, manual\_control), message untuk deskripsi event, data dalam format JSON untuk informasi tambahan, timestamp untuk waktu kejadian, dan created\_at untuk data kapan log ini dibuat.

d. Aplikasi Mobile (SilverLink: Connect)

Aplikasi ini mobile ini dibuat menggunakan Capacitor.js dengan Ionic Library untuk memberikan pengalaman native pada platform android. Aplikasi mengimplementasikan *adaptive UI* yang secara otomatis menyesuaikan interface berdasarkan jenis controller yang terhubung. Aplikasi juga berfungsi sebagai alat monitoring sensor, kontrol aktuator, dan pengaturan penyiraman.

#### A. Framework dan Library yang digunakan

Tabel 3.3 Framework dan Library yang digunakan pada SilverLink: Connect

Kategori	Library/Framework	Versi	Fungsi
Framework Utama	React.js	18.x	Komponen UI & Manajemen State
Platform Mobile	Capacitor.js	5.x	Integrasi Native Android
Framework UI	Material-IO	5.x	Library Komponen & Tema
HTTP Client	Axios	1.x	Komunikasi REST API
Grafik	Chart.js	4.x	Visualisasi Data
Penanganan Tanggal	Date-fns	2.x	Format & Manipulasi Tanggal
Penanganan Form	React Hook Form	7.x	Validasi & Manajemen Form
Penyimpanan	Capacitor Preferences	5.x	Persistensi Data Lokal

Kategori	Library/Framework	Versi	Fungsi
Notifikasi	React Toastify	9.x	Feedback & Peringatan User
Routing	React Router	6.x	Navigasi & Manajemen Route
Manajemen State	React Context API	Bawaan	Global State Management
Utilitas	Lodash	4.x	Fungsi Utilitas JavaScript
Validasi	Joi/Yup	17.x	Skema Validasi Data

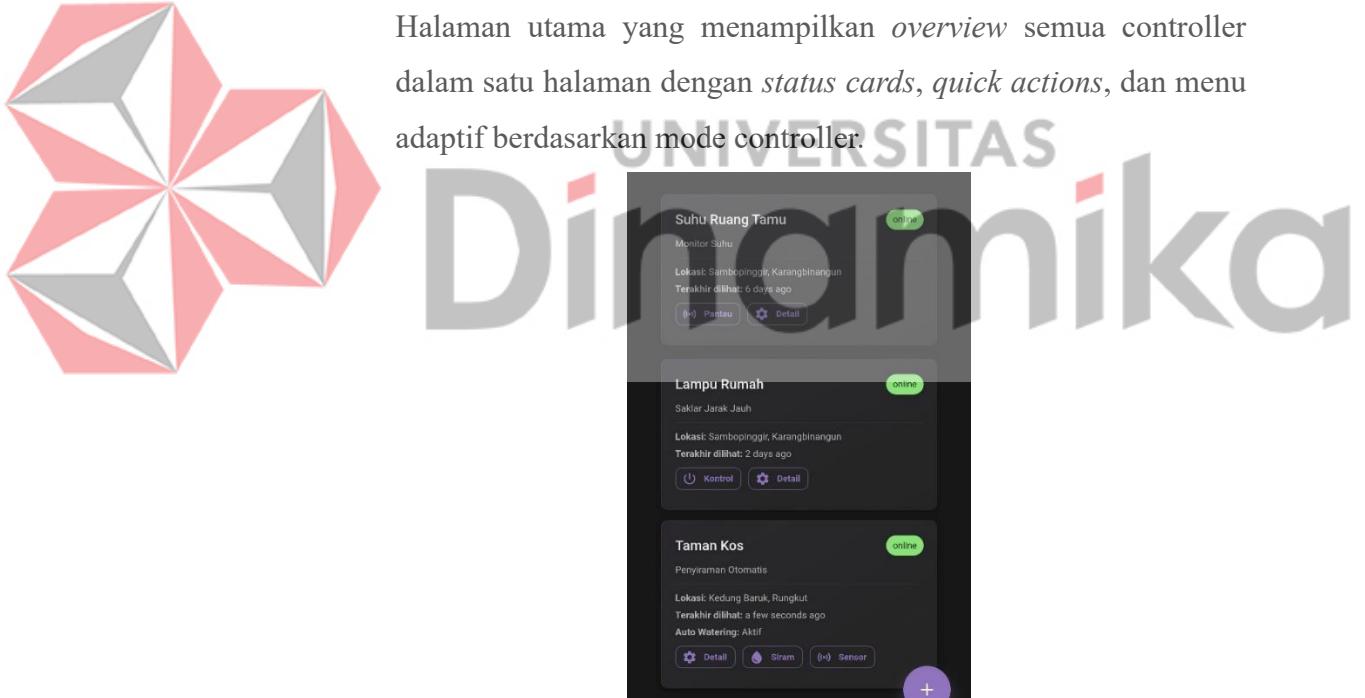
## B. Komponen Utama

### 1. Sistem Otentikasi User

Sistem ini menangani login user dengan JWT *token management*, *session persistence* menggunakan Capacitor Preferences, dan *automatic token refresh* untuk *seamless user experience*. Flowchart dapat dilihat di Lampiran 3.

### 2. Dashboard

Halaman utama yang menampilkan *overview* semua controller dalam satu halaman dengan *status cards*, *quick actions*, dan menu adaptif berdasarkan mode controller.

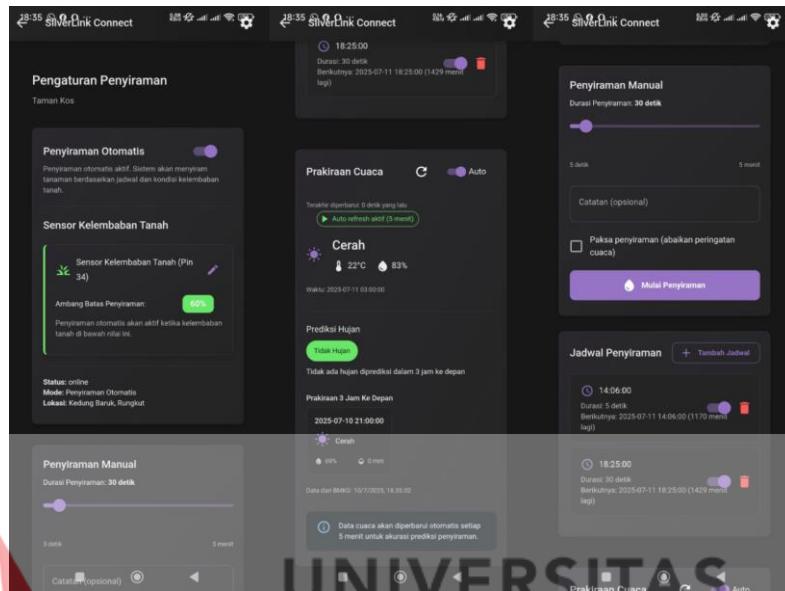


Gambar 3.14 Halaman Dashbord

### 3. Pengaturan Penyiraman

Interface khusus untuk controller mode watering dengan soil moisture monitoring, manual watering controls, weather integration, dan scheduling system.

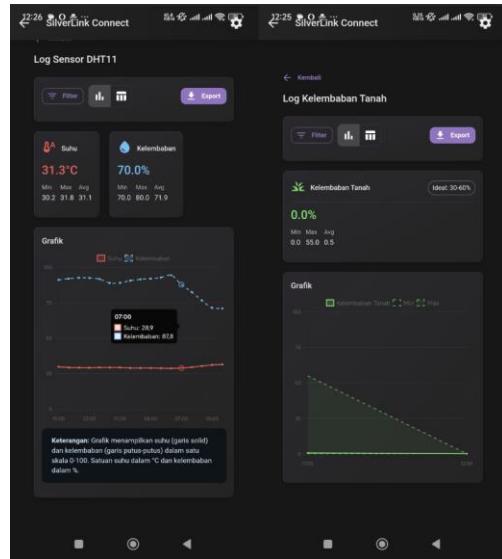
Halaman ini Terdiri dari fungsi penyiraman manual, Penjadwalan penyiraman, hasil prediksi cuaca dari BMKG, edit threshold kelembapan, menu untuk monitoring sensor, dan saklar untuk mematikan fungsi penyiraman otomatis. Khusus penyiraman manual user dapat menentukan durasi penyiraman.



Gambar 3.15 Halaman pengaturan penyiraman

#### 4. Halaman Monitoring Sensor

Halaman ini terdiri dari 2 halaman yaitu khusus untuk DHT11 dan Soil Moisture. Data Riwayat sensor akan ditampilkan dalam 2 bentuk berbeda, yaitu bentuk grafik dan log. Grafik dapat menampilkan data per jam atau per hari sesuai filter, dan log dapat melihat 50 data terakhir dari sensor.



Gambar 3.16 Halaman monitoring sensor



## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil Implementasi Sistem

##### 4.1.1 Hasil Implementasi Hardware

Implementasi hardware SilverLink berhasil direalisasikan dalam tiga prototype yang berbeda sesuai dengan kebutuhan spesifik pengguna. Setiap prototype telah dirakit dan diuji secara individual untuk memastikan fungsionalitas yang optimal. Setiap mode prototype tetap dipasang firmware yang sama, yang membedakan hanya sensor dan aktuator yang dipasang disesuaikan dengan kebutuhan pengguna.

###### a. Mode Penyiram Otomatis

Prototype watering berhasil dirakit dengan konfigurasi ESP32 sebagai controller utama, *capacitive soil moisture sensor* pada pin analog 34, relay 1 channel pada pin digital 17, *power supply* 24V, dan dual step-down (12V untuk pompa air, 5V untuk ESP32). Sistem dilengkapi dengan box tanah air untuk penggunaan outdoor dan LED indicator untuk status operasi.



Gambar 4.1 Perangkat penyiram otomatis

###### b. Mode Kendali Saklar

Prototype remote switch berhasil dirakit dengan konfigurasi ESP32, relay 4 channel pada pin 16, 17, 18, 19, dan power supply 24V dengan *step-down*

converter ke 5V. Sistem dilengkapi dengan terminal block AC untuk koneksi aman perangkat 220V dan housing dengan proper.



Gambar 4.2 Perangkat mode kendali saklar

c. Mode Pengukur Suhu dan Kelembapan

Prototype temperature monitoring berhasil dirakit dengan konfigurasi ESP32, sensor DHT11, dan power supply 24V, *step-down* 5V. Desain compact dengan penempatan sensor di luar box untuk akurasi pengukuran.



Gambar 4.3 Perangkat pengukur suhu dan kelembapan

#### 4.1.2 Hasil Implementasi Software

Implementasi software SilverLink telah berhasil dikembangkan dalam tiga layer utama: firmware ESP32, backend server, dan aplikasi mobile.

- a. Firmware ESP32 berhasil dikembangkan dengan arsitektur modular sehingga dapat digunakan untuk semua jenis mode operasi dalam satu basis kode yang sama. Setelah flashing firmware pada ESP32, dilanjutkan flash menggunakan LittleFS untuk membuat filesystem tempat menyimpan konfigurasi dan data lain pada ESP32. Implementasi menggunakan ESP32-V4, ini dipilih karena mempunyai PIN V5 sehingga dapat menggunakan power selain USB.
- b. Backend server berhasil dikembangkan dengan basis Node.js, database PostgreSQL, WebSocket server pada port 5050, dan REST API pada port 3000. Sistem dapat menangani banyak pengguna dan kontroler dengan jenis mode berbeda secara simultan. Pada implementasi ini, Server dipasang di perangkat PC dengan sistem operasi Windows 10 dengan spesifikasi AMD Ryzen 5 2400G, 24GB RAM, SSD. Database PostgreSQL dipasang di perangkat PC dengan sistem manajemen menggunakan Proxmox dengan alokasi sistem container untuk database PostgreSQL adalah 1 Core, 1GB RAM. Kedua mesin yaitu server dan database berada dalam segmen jaringan lokal yang sama 192.168.50.0/24 untuk kemudahan dan keamanan komunikasi antara database dan server. Dalam pengetesan ini, IP Publik digunakan untuk akses ke server dengan melakukan forwarding pada port 5050 dan 3000 di protokol TCP pada router, penulis juga menggunakan nama domain silverlink.eula.my.id untuk kemudahan dalam mengingat Alamat server.
- c. Aplikasi Mobile (SilverLink: Connect) berhasil dikembangkan menggunakan Capacitor.js dan React sebagai framework utama. Aplikasi dapat mengakomodasi setiap mode kontroler dengan memberi interface berbeda untuk setiap mode controller. Dalam pengetesan aplikasi, penulis menggunakan Smartphone Pribadi Huawei Nova 5T dengan Spesifikasi Kirin 980 Chipset, 8GB RAM, Android 10, dengan koneksi internet WiFi dan Seluler.

## 4.2 Pengujian Fungsionalitas Sistem

### 4.2.1 Pengujian Suhu

Pengujian suhu dilakukan untuk menentukan tingkat akurasi pengukuran dengan cara membandingkannya dengan sensor suhu yang siap pakai. Hasil pengujian akurasi suhu dapat dilihat pada tabel dibawah.

Tabel 4.1 Hasil uji suhu

No	Waktu Pengujian	Suhu Referensi (°C)	Suhu Sensor (°C)	Error (%)
1	11:00	26.8	28.5	6.33
2	11:05	26.8	28.5	6.33
3	11:10	26.8	28.5	6.33
4	11:15	26.8	28.5	6.33
5	11:20	26.8	28.5	6.33
6	11:25	26.9	28.5	5.92
7	11:30	26.9	28.5	5.92
8	11:35	26.9	28.5	5.92
9	11:40	26.9	28.5	5.92
10	11:45	26.9	28.7	6.67
11	11:50	27.0	28.9	7.04
12	11:55	27.0	28.9	7.04
13	12:00	27.0	28.9	7.04
14	12:05	27.0	28.9	7.04
15	12:10	27.0	28.9	7.04
16	12:15	27.0	28.9	7.04
Rata-Rata				6.43

Data hasil pengujian di atas menunjukkan bahwa pengujian dilakukan dengan menghitung error dari data yang diukur dan data referensi dari sensor yang sudah ada. Nilai error dapat dihitung dengan rumus:

$$\text{Error (\%)} = \left( \frac{|Suhu\ Sensor - Suhu\ Referensi|}{Suhu\ Referensi} \right) \times 100 \quad (1)$$

Dan dari semua pengukuran yang dilakukan, didapat rata-rata error sebesar 6.43% dengan total sampel sebanyak 10 data. Dengan ini dapat disimpulkan bahwa sensor ini belum terlalu akurat dalam mengukur suhu ruangan.

#### 4.2.2 Pengujian Kelembapan Udara

Pengujian Kelembapan Udara dilakukan untuk menentukan tingkat akurasi pengukuran dengan cara membandingkannya dengan sensor kelembapan udara yang siap pakai. Hasil pengujian akurasi kelembaban udara dapat dilihat pada tabel dibawah.

Tabel 4.2 Hasil uji kelembapan udara

No	Waktu Pengujian	Kelembapan Referensi	Kelembapan Sensor	Error (%)
		(%)	(%)	
1	11:00	63	78	23.81
2	11:05	63	77	22.22
3	11:10	63	77	22.22
4	11:15	63	77	22.22
5	11:20	63	77	22.22
6	11:25	63	77	22.22
7	11:30	63	77	22.22
8	11:35	63	77	22.22
9	11:40	64	79	23.44
10	11:45	64	80	25.00
11	11:50	64	79	23.44
12	11:55	64	79	23.44
13	12:00	64	79	23.44
14	12:05	64	78	18.75
15	12:10	64	77	10.94
16	12:15	63	76	19.05
Rata-Rata				21.02

Data hasil pengujian di atas menunjukkan bahwa pengujian dilakukan dengan menghitung error dari data yang diukur dan data referensi dari sensor yang sudah ada. Nilai error dapat dihitung dengan rumus:

$$\text{Error (\%)} = \left( \frac{|Kelembapan Sensor - Kelembapan Referensi|}{Kelembapan Referensi} \right) \times 100$$

Dan dari semua pengukuran yang dilakukan, didapat rata-rata error sebesar 21.02% dengan total sampel sebanyak 10 data. Dengan ini dapat disimpulkan bahwa sensor ini belum terlalu akurat dalam mengukur Kelembaban Udara.

### 4.2.3 Pengujian Kelembapan Tanah

Pengujian kelembapan tanah dilakukan untuk menentukan tingkat akurasi pengukuran dengan cara membandingkannya dengan sensor kelembapan tanah yang siap pakai. Hasil pengujian akurasi kelembapan udara dapat dilihat pada tabel dibawah.

Tabel 4.3 Hasil uji kelembapan tanah

No	Waktu Pengujian	Kelembapan Referensi	Kelembapan Sensor	Error (%)
		(%)	(%)	
1	11:00	94	78	17.02
2	11:05	86	80	6.98
3	11:10	80	77	3.75
4	11:15	78	79	1.28
5	11:20	76	77	1.32
6	11:25	74	77	4.05
7	11:30	73	76	4.11
8	11:35	72	78	8.33
9	11:40	71	77	8.45
10	11:45	70	77	10.00
11	11:50	70	76	8.57
12	11:55	70	76	8.57
13	12:00	68	76	11.76
14	12:05	69	75	8.70
15	12:10	67	75	11.94
16	12:15	66	75	13.64
Rata-Rata				7.96

Data hasil pengujian di atas menunjukkan bahwa pengujian dilakukan dengan menghitung error dari data yang diukur dan data referensi dari sensor yang sudah ada. Nilai error dapat dihitung dengan rumus:

$$\text{Error (\%)} = \left( \frac{|Kelembapan Sensor - Kelembapan Referensi|}{Kelembapan Referensi} \right) \times 100$$

Dan dari semua pengukuran yang dilakukan, didapat rata-rata error sebesar 7.96 % dengan total sampel sebanyak 5 data. Dengan ini dapat disimpulkan bahwa sensor ini belum terlalu akurat dalam mengukur kelembapan tanah.

#### 4.1.4. Pengujian Penyiraman Otomatis

Pengujian sistem penyiraman otomatis bertujuan untuk mengevaluasi akurasi dan efektivitas sistem pengambilan keputusan berdasarkan kelembapan tanah dan integrasi data cuaca BMKG. Data diambil dari log penyiraman pada sistem yang sudah disiapkan untuk pengujian.

Prosedur pengujian:

1. Setup perangkat dengan menancapkan sensor ke tanah.
2. Konfigurasi threshold kelembapan disetel diatas nilai kelembapan untuk trigger penyiraman.
3. Set waktu penyiraman di beberapa menit kedepan.
4. Menunggu sampai waktu penyiraman tiba.
5. Mengulangi dengan kondisi tanah lebih lembap.

Hasil pengujian dapat dilihat di tabel 4.4 sebagai berikut.

Tabel 4.4 Hasil uji sistem penyiraman otomatis

No	Waktu Pengujian	Nilai Kelembapan	threshold	Prakiraan Cuaca	Status Penyiraman	Status Validasi
1	11:30	66	75	Cerah	True	Valid
2	11:45	62	75	Cerah	True	Valid
3	12:00	56	75	Cerah	True	Valid
4	12:15	56	75	Cerah	True	Valid
5	13:00	59	50	Cerah	false	Valid

Data hasil pengujian di atas menunjukkan, semua pengujian yang dilakukan valid. Semua hasil uji penyiraman berjalan sesuai logika yang ditanam di program penyiraman otomatis, dimana jika tanah kering dan tidak hujan, maka disiram.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan hasil implementasi dan pengujian sistem SilverLink, dapat disimpulkan sebagai berikut:

1. Sistem SilverLink mampu menentukan kapan tanaman perlu disiram secara otomatis menggunakan sensor kelembapan tanah (capacitive soil moisture sensor) yang membaca nilai kelembapan dan membandingkannya dengan ambang batas yang ditentukan. Selain itu, sistem juga memperhitungkan data prakiraan cuaca untuk menghindari penyiraman saat turun hujan. Logika keputusan ini berhasil diterapkan dengan tingkat keberhasilan 100% pada skenario uji yang dilakukan.
2. Sistem SilverLink dapat memantau suhu ruangan secara real-time menggunakan sensor DHT11 yang mengirimkan data secara terus-menerus ke backend server. Data ini kemudian ditampilkan secara langsung di aplikasi pengguna. Namun, dari hasil pengujian, sensor DHT11 menunjukkan rata-rata error sebesar 6.43%, sehingga sistem masih dapat digunakan untuk monitoring suhu kasar, tetapi tidak ideal untuk aplikasi yang membutuhkan akurasi tinggi.

#### **5.2. Saran**

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, berikut adalah saran untuk pengembangan dan perbaikan sistem SilverLink:

1. Berdasarkan pengujian yang menunjukkan error rata-rata sebesar 7.96% pada sensor kelembapan tanah, disarankan untuk menerapkan metode kalibrasi multi-titik (multi-point calibration) guna meningkatkan akurasi pembacaan dan keandalan sistem penyiraman otomatis.
2. Untuk meningkatkan ketahanan dan akurasi sistem dalam jangka panjang, pengembangan algoritma machine learning dapat digunakan untuk melakukan kompensasi terhadap error sensor secara adaptif, serta untuk predictive maintenance dengan memantau degradasi performa sensor seiring waktu.

3. Disarankan untuk mengembangkan desain housing perangkat yang lebih profesional dan tahan terhadap kondisi lingkungan, dengan mempertimbangkan penggunaan material yang sesuai dengan standar IP rating (Ingress Protection) untuk aplikasi indoor maupun outdoor, serta memperhatikan standar keselamatan listrik agar sistem dapat digunakan secara aman dan andal.



## DAFTAR PUSTAKA

- Achmad Ridho'I, Kukuh Setyadjit, Bayu Era Yordhan. (2023). Sistem Monitoring Suhu Dan Kelembapan Pada Budidaya Jamur Tiram Menggunakan ESP32. Jurnal FORTECH. Vol. 4. No. 1. DOI : 10.56795/fortech.v4i1.4103
- Arfend Atma Maulana Khalifa, Kiki Prawiroredjo. (2022). MODEL SISTEM PENGENDALIAN SUHU DAN KELEMBAPAN RUANGAN PRODUKSI IBAT BERBASIS NODEMCU ESP32. Jurnal Teknik Elektro, Teknologi Informasi dan Komputer. Vol. 6, No. 1. DOI : 10.31961/eltikom.v6i1.415
- Aryani Rombekila, Bernardo Luoukelay Entamoing. (2022). Prototype Smart Home Berbasis IoT Dengan Handphone Android Menggunakan NODEMCU ESP32. Jurnal Teknik AMATA Vol. 03 No. 1, DOI : 10.55334/jtam.v3i1.275
- E B Agustina, A T Dewi. (2023). Perancangan Alat Penyiram Tanaman Otomatis Berbasis Mikrokontroler dan *Internet of Things* (IoT). Jurnal Lontar Physics Today. Vol 2, No 3. DOI : 10.26877/lpt.v2i3.18031
- Ihsanul Fadillah Amin, Dewi Laksmiati. (2024). Perancangan Jemuran Otomatis Berbasis IoT Menggunakan ESP32 Dan API. Jurnal Ilmiah GIGA. Vol 27 (1). DOI : 10.47313/jig.v27i1.3710
- Nabil Azzaky, Anang Widiantoro. (2020). Alat Penyiram Tanaman Otomatis Berbasis Arduino menggunakan *Internet Of Things* (IOT). J-Eltrik, Vol. 2, No. 3. DOI : 10.30649/j-eltrik.v2i2.48
- Nimas Raya Ahsy, Adhitya Bhawiyuga, Dany Primanita Kartikasari. (2019). Implementasi Sistem Kontrol dan Monitoring Smart Home Menggunakan Integrasi Protokol Websocket dan MQTT. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 3, No. 4.