

Smart Greenhouse Otomatis Berbasis IoT untuk Tanaman Selada

Danny akbar permana^{1*}, Christian Wahyu Kurniawan², Muhammad Hafids Sidiq³

¹Teknik Informatika

Universitas Duta Bangsa Surakarta

^{1*}220103228@mhs.udb.ac.id

² Teknik Informatika

Universitas Duta Bangsa Surakarta

²220103227@mhs.udb.ac.id

³Teknik Informatika

Universitas Duta Bangsa Surakarta

³220103238@mhs.udb.ac.id

Abstrak- Penelitian ini bertujuan untuk merancang dan mengembangkan sistem Smart Greenhouse berbasis mikrokontroler ESP32 yang mampu mengatur atap dan penyiraman tanaman selada secara otomatis berdasarkan kelembaban tanah. Permasalahan yang diangkat adalah kesulitan menjaga kestabilan lingkungan rumah kaca secara manual. Sistem dilengkapi dengan sensor YL-69 untuk kelembaban tanah, DHT22 untuk suhu dan kelembapan udara, serta LDR untuk intensitas cahaya. Proses otomatisasi menggunakan algoritma pada mikrokontroler, serta pemantauan dan kontrol manual dilakukan melalui aplikasi Blynk. Metodologi yang digunakan adalah Extreme Programming (XP) dengan pengujian berbasis simulasi Wokwi. Hasil menunjukkan bahwa sistem mampu bekerja stabil dalam merespon kondisi suhu dan kelembaban tanah secara otomatis, meskipun pengaturan berdasarkan intensitas cahaya masih memerlukan pengembangan. Kesimpulannya, sistem ini efektif meningkatkan efisiensi monitoring dan perawatan tanaman selada secara otomatis dan real-time berbasis IoT.

Kata Kunci- Smart Greenhouse, IoT, Selada, Penyiraman Otomatis, ESP32, Wokwi, Blynk

Abstract- This study aims to design and develop a Smart Greenhouse system based on the ESP32 microcontroller to automatically control the roof and watering of lettuce plants based on soil moisture. The main problem addressed is the difficulty in manually maintaining greenhouse environmental stability. The system uses YL-69 sensors for soil moisture, DHT22 for air temperature and humidity, and LDR for light intensity. Automation is achieved through sensor-based logic algorithms, while monitoring and manual control are handled via the Blynk application. The methodology used is Extreme Programming (XP), with testing performed through Wokwi simulations. Results show the system responds reliably to changes in temperature and soil moisture, though automatic control based on light intensity needs further development. In conclusion, this system effectively enhances the efficiency of lettuce plant care and real-time monitoring through IoT-based automation.

Keywords: Smart Greenhouse, IoT, Lettuce, Automatic Watering, ESP32, Wokwi, Blynk

I. PENDAHULUAN

Perkembangan teknologi pertanian modern telah menghasilkan konsep Smart Farming yang mengintegrasikan sensor, otomatisasi, dan telematika. Smart Farming diharapkan mampu menjadi solusi atas berbagai permasalahan modern seperti keterbatasan lahan pertanian, perubahan iklim global, serta penurunan jumlah tenaga kerja di sektor pertanian. Salah satu implementasi nyata dari konsep ini adalah Smart Greenhouse, yaitu rumah kaca yang dilengkapi dengan sistem otomatisasi untuk menjaga kestabilan parameter lingkungan tumbuh tanaman. Tanaman selada merupakan salah satu jenis tanaman hortikultura yang membutuhkan lingkungan dengan kondisi suhu, kelembaban udara, pencahayaan, dan kadar air tanah yang stabil agar dapat tumbuh secara optimal. Ketidaksesuaian terhadap salah satu parameter tersebut dapat

menyebabkan pertumbuhan tanaman menjadi terhambat atau tidak maksimal. Oleh karena itu, diperlukan suatu sistem monitoring dan pengendalian lingkungan yang mampu bekerja secara otomatis guna menjaga kestabilan kondisi tumbuh tanaman.

Penelitian ini bertujuan untuk merancang dan membangun sistem Smart Greenhouse otomatis berbasis Internet of Things (IoT) yang menggunakan mikrokontroler ESP32 sebagai pusat kendali. Sistem ini dilengkapi dengan beberapa sensor, antara lain sensor kelembaban tanah YL-69, sensor suhu dan kelembaban udara DHT22, serta sensor cahaya LDR. Pengendalian parameter lingkungan pada greenhouse dilakukan secara otomatis melalui pemrosesan data dari sensor oleh mikrokontroler, sedangkan kontrol manual dan monitoring jarak jauh dapat dilakukan menggunakan aplikasi Blynk. Dengan demikian, pengguna dapat memantau dan

mengendalikan kondisi greenhouse secara real-time melalui perangkat mobile.

Sebagai upaya pengujian dan validasi sistem, penelitian ini menggunakan simulasi berbasis Wokwi untuk mengevaluasi respons sistem terhadap berbagai kondisi lingkungan yang berubah-ubah, seperti variasi suhu udara, kelembaban tanah, dan intensitas cahaya. Simulasi ini memberikan kesempatan untuk melakukan uji coba secara aman dan efisien tanpa perlu membangun greenhouse fisik secara langsung. Hal ini bertujuan agar seluruh fitur dan fungsionalitas sistem dapat diuji secara terukur dan sistematis sebelum diterapkan pada kondisi nyata di lapangan.

Adapun rumusan masalah dalam penelitian ini mencakup tiga hal utama, yaitu bagaimana merancang sistem greenhouse otomatis berbasis IoT untuk mengatur suhu, kelembaban, dan pencahayaan tanaman selada; bagaimana mengintegrasikan sensor dan mikrokontroler agar sistem pengendali atap serta penyiraman bekerja otomatis sesuai kebutuhan tanaman; serta bagaimana menguji keefektifan sistem tersebut menggunakan simulasi digital berbasis Wokwi. Batasan penelitian difokuskan pada penggunaan tanaman selada sebagai objek studi, penggunaan mikrokontroler ESP32 sebagai pusat kendali, serta implementasi sistem monitoring dan pengendalian otomatis untuk atap dan penyiraman tanaman. Sistem monitoring dan kontrol dilakukan melalui aplikasi Blynk, sedangkan fitur pemupukan otomatis tidak termasuk dalam cakupan penelitian ini.

Secara umum, tujuan dari penelitian ini adalah untuk merancang dan mengimplementasikan Smart Greenhouse otomatis berbasis IoT guna mendukung budidaya tanaman selada. Selain itu, penelitian ini bertujuan untuk meningkatkan efisiensi pertanian dengan cara mengotomatisasi proses penyiraman dan pengaturan atap berdasarkan data sensor yang diperoleh secara real-time. Penelitian juga bertujuan untuk memanfaatkan sensor YL-69, DHT22, dan LDR yang terhubung dengan mikrokontroler ESP32 guna melakukan monitoring lingkungan dan pengambilan keputusan otomatis. Terakhir, dilakukan pengujian sistem melalui simulasi berbasis Wokwi untuk mengevaluasi kinerja otomatisasi dalam berbagai skenario lingkungan guna memastikan sistem berjalan sesuai harapan.

II. METODOLOGI PENELITIAN

Pada penelitian ini, metode pengembangan yang digunakan adalah Extreme Programming (XP). XP merupakan salah satu metode dalam pengembangan perangkat lunak agile yang berfokus pada pengembangan iteratif, komunikasi tim yang intens, serta pengujian berkelanjutan. Pemilihan XP didasarkan pada kebutuhan proyek Smart Greenhouse yang memiliki dinamika parameter lingkungan yang terus berubah dan memerlukan pengembangan perangkat lunak yang fleksibel serta mampu beradaptasi secara cepat terhadap perubahan kebutuhan sistem.

Tahapan awal yang dilakukan adalah perencanaan (Planning) yang meliputi analisis kebutuhan fungsional dan non-fungsional. Pada tahap ini, diidentifikasi kebutuhan utama sistem seperti otomatisasi penyiraman berdasarkan kelembaban tanah, pengaturan atap secara otomatis berdasarkan suhu dan intensitas cahaya, serta monitoring real-time menggunakan aplikasi Blynk. Selain itu, ditetapkan bahwa pengujian awal sistem akan dilakukan melalui simulasi berbasis Wokwi untuk memastikan keandalan logika kontrol sebelum implementasi fisik dilakukan.

Tahap berikutnya adalah perancangan (Design), di mana dilakukan pembuatan diagram blok sistem, perancangan alur logika otomatisasi, serta perancangan integrasi antara sensor YL-69, DHT22, dan LDR dengan mikrokontroler ESP32. Perancangan antarmuka monitoring menggunakan aplikasi Blynk juga dilakukan untuk memungkinkan monitoring jarak jauh dan kontrol manual.

Setelah perancangan selesai, dilakukan tahap pengkodean (Coding) yang berfokus pada implementasi logika otomatisasi dan integrasi seluruh komponen menggunakan bahasa pemrograman C++. Pada tahap ini, sensor-sensor membaca parameter lingkungan secara real-time dan mikrokontroler ESP32 mengambil keputusan untuk mengaktifkan atau menonaktifkan aktuator seperti pompa air dan motor servo atap berdasarkan ambang batas yang telah ditentukan. Data dari sensor juga dikirimkan secara berkala ke aplikasi Blynk untuk mendukung monitoring secara jarak jauh.

Tahap selanjutnya adalah pengujian (Testing) yang dilakukan secara bertahap, dimulai dari pengujian komponen individual hingga pengujian

integrasi sistem secara keseluruhan. Pengujian menggunakan platform simulasi Wokwi untuk memvalidasi respons sistem terhadap perubahan suhu, kelembaban tanah, dan intensitas cahaya. Hasil pengujian memastikan bahwa logika kendali otomatis berjalan sesuai dengan perancangan, dan integrasi dengan aplikasi Blynk dapat berjalan stabil.

Setelah sistem diuji dan dinyatakan stabil dalam simulasi, tahap Rilis (Release) dilakukan dengan menyiapkan prototype virtual sistem Smart Greenhouse. Tahap akhir dalam metode XP adalah pemeliharaan (Maintenance) yang dilakukan secara berkelanjutan untuk memastikan sistem tetap optimal dan adaptif terhadap kebutuhan baru. Pada tahap ini, direncanakan pengembangan lanjutan seperti penambahan fitur otomatisasi pencahayaan dan pemberian nutrisi tanaman secara otomatis untuk meningkatkan efisiensi pertanian lebih lanjut.

Dengan penerapan metodologi Extreme Programming ini, proses pengembangan Smart Greenhouse dapat dilakukan secara terstruktur namun fleksibel, memungkinkan tim pengembang untuk beradaptasi dengan kebutuhan sistem yang terus berkembang serta memberikan hasil yang sesuai dengan tujuan penelitian.

III. HASIL DAN PEMBAHASAN

A. Planning(Perencanaan)

Tahap perencanaan dalam metode Extreme Programming (XP) menjadi langkah awal yang sangat penting dalam pengembangan sistem Smart Greenhouse berbasis IoT ini. Pada tahap ini, tim pengembang bersama stakeholder mendiskusikan kebutuhan sistem yang akan dikembangkan, permasalahan yang dihadapi dalam proses pertanian selada, serta solusi teknologi yang dapat diimplementasikan secara efektif.

Permasalahan utama yang diidentifikasi adalah sulitnya menjaga kestabilan lingkungan rumah kaca secara manual, terutama pada parameter kelembaban tanah, suhu udara, dan intensitas cahaya. Berdasarkan hasil diskusi, ditetapkan bahwa solusi yang diambil adalah membangun sistem otomatisasi berbasis mikrokontroler ESP32 yang dilengkapi dengan sensor kelembaban tanah YL-69, sensor suhu dan kelembaban udara DHT22, serta sensor cahaya LDR. Ketiga sensor ini dipilih karena memiliki kompatibilitas dengan ESP32, konsumsi daya yang

rendah, serta kemudahan dalam implementasinya pada sistem Internet of Things (IoT).

Selain itu, pada tahap ini ditentukan bahwa aplikasi Blynk akan digunakan sebagai platform monitoring dan kendali manual berbasis mobile. Aplikasi ini dipilih karena mendukung integrasi dengan ESP32 melalui koneksi Wi-Fi dan memiliki tampilan antarmuka pengguna yang sederhana namun fungsional.

Dalam tahap perencanaan juga diputuskan bahwa simulasi sistem akan dilakukan menggunakan platform Wokwi. Platform ini memungkinkan pengujian sistem secara virtual tanpa memerlukan perangkat keras fisik, sehingga proses pengembangan dapat dilakukan dengan lebih cepat dan efisien. Simulasi Wokwi akan digunakan untuk menguji respons sistem terhadap berbagai kondisi lingkungan, seperti kelembaban tanah rendah, suhu udara tinggi, atau intensitas cahaya berlebih.

Dari hasil analisis kebutuhan, tim merancang beberapa fitur utama yang akan dikembangkan, yaitu:

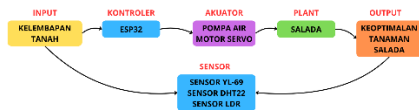
- a) Otomatisasi penyiraman tanaman berdasarkan pembacaan kelembaban tanah.
- b) Otomatisasi penutupan dan pembukaan atap rumah kaca berdasarkan suhu udara dan intensitas cahaya.
- c) Monitoring suhu, kelembaban udara, dan cahaya secara real-time melalui aplikasi Blynk.
- d) Kemampuan kontrol manual terhadap penyiraman dan atap melalui smartphone.
- e) Simulasi pengujian seluruh sistem secara virtual sebelum penerapan nyata.

Hasil perencanaan dituangkan dalam user stories yang menggambarkan kebutuhan pengguna secara singkat dan jelas. Setiap user story akan menjadi dasar pembuatan task pada tahap pengembangan berikutnya. Proses perencanaan dalam metode XP bersifat iteratif, sehingga kebutuhan yang telah ditentukan masih dapat direvisi atau ditambah sesuai dengan hasil evaluasi pada tahap testing dan listening berikutnya.

Dengan perencanaan yang matang, tim memiliki gambaran yang jelas mengenai tujuan akhir dari sistem Smart Greenhouse, ruang lingkup pengembangan, serta prioritas fitur-fitur utama yang akan dikembangkan pada iterasi awal.

B. Design(Perencanaan)

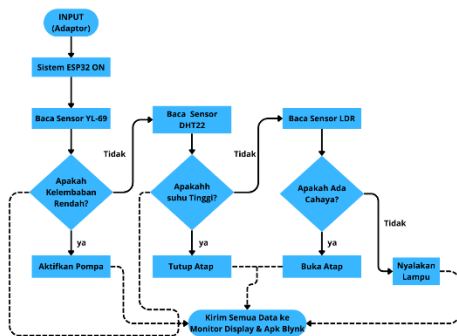
Tahap design dalam metode Extreme Programming (XP) bertujuan untuk merancang sistem secara sederhana namun efektif, sesuai dengan prinsip simple design yang diutamakan dalam XP. Pada penelitian ini, desain sistem Smart Greenhouse difokuskan pada bagaimana komponen-komponen utama, yaitu mikrokontroler, sensor, dan aktuator, dapat berinteraksi secara optimal untuk menciptakan sistem otomatisasi lingkungan tanaman selada yang stabil.



Gambar 1 Sistem Kerja

Perancangan diawali dengan membuat diagram blok sistem, yang menggambarkan hubungan antara sensor YL-69, DHT22, dan LDR dengan mikrokontroler ESP32. Sensor YL-69 bertugas membaca tingkat kelembaban tanah, DHT22 untuk membaca suhu dan kelembaban udara, serta LDR untuk mendeteksi intensitas cahaya. Data dari ketiga sensor ini kemudian dikirimkan ke ESP32 sebagai pusat kendali, yang akan menganalisis data dan memberikan perintah kepada aktuator berupa pompa air dan motor servo.

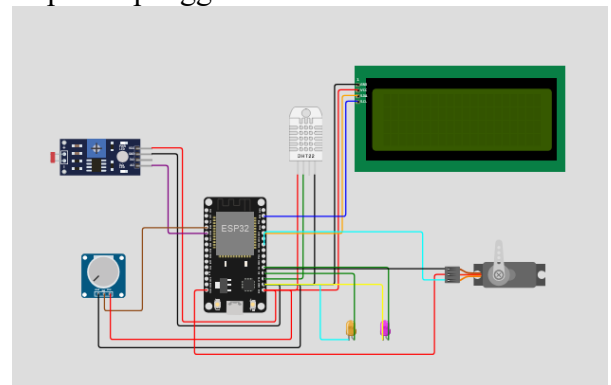
Pompa air akan diaktifkan secara otomatis ketika kelembaban tanah berada di bawah ambang batas tertentu, sedangkan motor servo akan menggerakkan atap untuk membuka atau menutup rumah kaca berdasarkan data suhu dan intensitas cahaya yang terdeteksi. Dengan demikian, perancangan sistem ini berfokus pada menciptakan alur logika otomatisasi yang sederhana namun mampu merespons perubahan kondisi lingkungan secara real-time.



Gambar 2 Flowchart Alur Sistem

Selain itu, pada tahap design juga dirancang flowchart sistem yang menggambarkan logika keputusan yang diterapkan dalam mikrokontroler ESP32. Flowchart ini menunjukkan urutan langkah-langkah mulai dari pembacaan sensor, pengambilan keputusan berdasarkan threshold yang ditentukan, hingga aktivasi aktuator. Dengan adanya flowchart, alur kerja sistem dapat divisualisasikan secara jelas dan memudahkan tim pengembang untuk memahami logika keseluruhan.

Pada sisi monitoring dan kontrol jarak jauh, dilakukan perancangan interface aplikasi Blynk yang menampilkan data suhu, kelembaban tanah, dan intensitas cahaya secara real-time. Interface ini juga menyediakan tombol kontrol manual untuk mengaktifkan atau menonaktifkan pompa air dan mengatur posisi atap secara langsung dari smartphone pengguna.



Gambar 3 Desain sistem secara virtual di wokwi

Perancangan yang sederhana namun modular ini memudahkan tim untuk melakukan penyesuaian atau pengembangan lebih lanjut pada iterasi berikutnya tanpa perlu mengubah keseluruhan struktur sistem. Dengan pendekatan design seperti ini, proses pengembangan dapat berjalan secara bertahap dan fleksibel sesuai dengan prinsip XP yang mengutamakan perubahan cepat dan pengembangan berkelanjutan.

C. Coding(Pengkodean)

Tahap coding merupakan proses implementasi dari perancangan sistem ke dalam bentuk program yang dijalankan pada platform virtual. Dalam penelitian ini, pengkodean dilakukan secara virtual menggunakan simulator Wokwi sebagai pengganti perangkat keras nyata, dan Firebase Realtime Database sebagai media penyimpanan data sensor

sekaligus penghubung antara mikrokontroler ESP32 dengan website monitoring. Penggunaan Firebase memungkinkan integrasi sistem Internet of Things (IoT) secara cloud-based dengan komunikasi data real-time yang stabil.

```
#include <Wire.h>
#include "LiquidCrystal_I2C.h"
#include "DHT.h"
#include <ESP32Servo.h>
#include "FC28.h"
#include <WiFi.h>
#include <HTTPClient.h>
#include <time.h>

const int servoPin = 18;
#define ldrPin 35
#define DHT_PIN 15
#define DHTTYPE DHT22
#define SensorFC28_Pin 34
#define redLEDPin 4
#define yellowLEDPin 2

DHT dht(DHT_PIN, DHT22);
LiquidCrystal_I2C lcd(0x27, 20, 4);
FC28Sensor mySensor;
Servo servo;

const float gama = 0.7;
const float r110 = 50;

const char* ssid = "Wokwi-GUEST";
const char* password = "";

const String firebaseHost =
"https://smartgreenhouse-bda96-
default-rtdb.asia-
southeast1.firebaseio.com";
const String firebasePath =
"/monitoring.json";

void setup() {
  Serial.begin(9600);
  WiFi.begin(ssid, password);
  while (WiFi.status() !=
WL_CONNECTED) {
    delay(500);
```

```
    Serial.print(".");
  }
  Serial.println("\nWiFi
Connected");

  // Sinkronisasi Waktu NTP
  configTime(7 * 3600, 0,
"pool.ntp.org", "time.nist.gov"); //
GMT+7
  Serial.print("Menunggu waktu
NTP");
  while (!time(nullptr)) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("\nWaktu NTP
sinkron!");

  pinMode(ldrPin, INPUT);
  pinMode(DHT_PIN, INPUT);
  pinMode(redLEDPin, OUTPUT);
  pinMode(yellowLEDPin, OUTPUT);
  mySensor.initFC28Sensor(115200,
SensorFC28_Pin);
  servo.attach(servoPin);
  dht.begin();
  lcd.init();
  lcd.backlight();
  lcd.setCursor(3, 0);
  lcd.print("Final Project");
  lcd.setCursor(0, 1);
  lcd.print("Penyiraman Tanaman");
  lcd.setCursor(2, 2);
  lcd.print("Otomatis");
  lcd.setCursor(2, 3);
  lcd.print("Monitoring System");
  delay(3000);
  lcd.clear();
}

void loop() {
  lcd.clear();

  float hum = dht.readHumidity();
  float temp =
dht.readTemperature();
```

```

int ldrValue = analogRead(ldrPin);
ldrValue = map(ldrValue, 4095, 0,
1024, 0);
float ldrVoltage = ldrValue /
1024.0 * 5;
float ldrResistance = 2000 *
ldrVoltage / (1 - ldrVoltage / 5);
float lux = pow(r110 * 1e3 *
pow(10, gama) / ldrResistance, (1 /
gama));

float soil_moisture =
mySensor.getSoilMoisture();

// Interpretasi nilai
String kondisiTanah =
soil_moisture < 33.33 ? "Basah" :
soil_moisture > 66.66 ? "Kering" :
"Normal";
String kecerahan = lux < 807 ?
"Kurang" : lux > 808 && lux < 1614 ?
"Medium" : "Tinggi";

// Kendali LED dan Servo
digitalWrite(redLEDPin, lux < 807
? HIGH : LOW);
digitalWrite(yellowLEDPin,
soil_moisture > 66.66 ? HIGH : LOW);
if (temp < 15) servo.write(180);
else if (temp > 20)
servo.write(0);

// Ambil waktu saat ini
time_t now = time(nullptr);
struct tm* p_tm = localtime(&now);
char waktu[25];
sprintf(waktu, "%04d-%02d-%02d
%02d:%02d:%02d",
p_tm->tm_year + 1900,
p_tm->tm_mon + 1,
p_tm->tm_mday,
p_tm->tm_hour,
p_tm->tm_min,
p_tm->tm_sec);

```

```

// Tampilkan ke LCD
lcd.setCursor(0, 0);
lcd.print("Tanah : ");
lcd.print(kondisiTanah);
lcd.setCursor(0, 1);
lcd.print("Cahaya: ");
lcd.print(kecerahan);
lcd.setCursor(0, 2);
lcd.print("Suhu : ");
lcd.print(temp); lcd.print("C");
lcd.setCursor(0, 3);
lcd.print("Lembap: ");
lcd.print(hum); lcd.print("%");

// Kirim ke Firebase
if (WiFi.status() == WL_CONNECTED)
{
  HTTPClient http;
  http.begin(firebaseHost +
firebasePath);
  http.addHeader("Content-Type",
"application/json");

  String json = "{";
  json += "\"kondisi_tanah\": \"" +
kondisiTanah + "\", ";
  json += "\"kecerahan\": \"" +
kecerahan + "\", ";
  json += "\"suhu\": \"" +
String(temp, 2) + "\", ";
  json += "\"kelembapan\": \"" +
String(hum, 1) + "\", ";
  json += "\"waktu_input\": \"" +
String(waktu) + "\"";
  json += "}";

  int httpCode = http.POST(json);
  Serial.println("Firebase
Response Code: " +
String(httpCode));
  http.end();
} else {
  Serial.println("WiFi Tidak
Terhubung");
}

```

```
delay(5000);
}
```

Tabel 1 coding melalui Wokwi

Pengkodean diawali dengan implementasi pembacaan data sensor secara virtual di Wokwi, di mana sensor YL-69, DHT22, dan LDR dimodelkan secara digital. ESP32 membaca data kelembaban tanah, suhu, kelembaban udara, serta intensitas cahaya dalam interval waktu tertentu. Data sensor dikirimkan secara otomatis ke Firebase Realtime Database menggunakan protokol HTTP atau REST API dengan metode POST atau PUT. Dengan demikian, Firebase berperan sebagai pusat penyimpanan dan distribusi data sensor yang nantinya dapat diakses oleh aplikasi web secara real-time.

ADC Output	Status
<300	Tanah kering (pompa aktif)
300-600	Ideal (pompa mati)
>600	Terlalu basah (pompa tetap mati)

Tabel 2 status pompa penyiraman

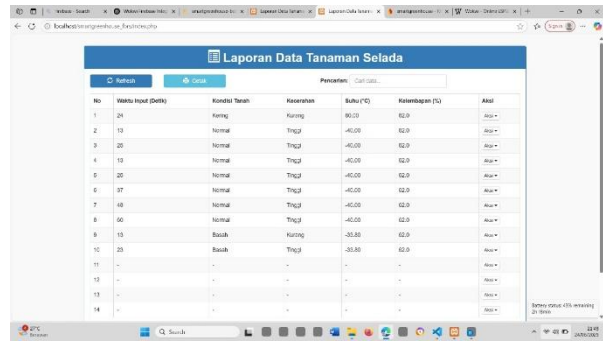
Suhu (°C)	Status Servo
<15	Terbuka
15-20	Netral / Tidak berubah
>20	Tertutup

Tabel 3 Status Suhu

Lux	Status Cahaya
<5000	Gelap
~10.000	Ideal
>15.000	Terlalu terang

Tabel 4 Status Cahaya

Logika kendali otomatis ditanamkan dalam kode ESP32 menggunakan bahasa C++. Ketika kelembaban tanah terbaca < 300 ADC, ESP32 mengirimkan perintah ke Firebase untuk mengubah status pompa menjadi aktif, dan mematikan pompa ketika tanah berada dalam kondisi ideal. Pada parameter suhu, jika nilai < 15°C, maka ESP32 mengirimkan perintah membuka atap dengan mengaktifkan motor servo secara virtual. Sebaliknya, jika suhu > 20°C, ESP32 menutup atap kembali untuk menjaga kestabilan lingkungan. Data intensitas cahaya dicatat dalam Firebase namun belum dikaitkan dengan aksi otomatis peneduh, yang direncanakan untuk pengembangan selanjutnya.



Gambar 4 Website

Sebagai media monitoring dan kontrol manual, dikembangkan website berbasis Firebase Hosting yang terhubung langsung dengan Realtime Database. Website ini dibuat menggunakan HTML, CSS, dan JavaScript dengan framework seperti Firebase SDK dan Vue.js/React (sesuai kebutuhan tim). Website menampilkan data sensor secara real-time yang ter-update otomatis saat Firebase menerima data baru dari ESP32. Selain monitoring, website juga menyediakan fitur kontrol manual berupa tombol virtual yang dapat mengubah status pompa dan posisi atap. Ketika tombol ditekan, perintah dikirimkan kembali ke Firebase dan diambil oleh ESP32 melalui proses pembacaan data yang berjalan secara periodik.

Pengkodean dilakukan secara iteratif mengikuti prinsip incremental development dalam Extreme Programming. Setiap fitur dasar seperti pengambilan data sensor, pengiriman ke Firebase, dan pembuatan dashboard web dikembangkan secara bertahap dan diuji menggunakan simulasi Wokwi. Proses debugging dan optimasi dilakukan pada setiap iterasi pengembangan untuk memastikan kestabilan komunikasi antara ESP32, Firebase, dan website monitoring.

Dengan pendekatan pengkodean berbasis cloud dan simulasi virtual, sistem Smart Greenhouse yang dikembangkan telah mampu melakukan pembacaan data, pengambilan keputusan otomatis, serta monitoring dan kontrol jarak jauh secara real-time. Seluruh proses dilakukan tanpa perangkat keras fisik, sehingga pengembangan dapat dilakukan secara efisien dan fleksibel sebelum diterapkan pada implementasi fisik yang sesungguhnya.

D. Testing

Tahap pengujian dalam penelitian ini dilakukan untuk memvalidasi fungsi-fungsi utama yang telah dikembangkan secara virtual menggunakan Wokwi

dan Firebase. Pengujian dilakukan secara bertahap, dimulai dari pengujian unit komponen hingga pengujian integrasi sistem secara keseluruhan. Pengujian dilakukan dengan pendekatan iteratif sesuai prinsip pengembangan Extreme Programming (XP), di mana setiap hasil pengujian menjadi dasar perbaikan pada iterasi selanjutnya.

1) Unit Testing Sensor dan Aktuator

Pengujian awal dilakukan secara individual pada masing-masing sensor dan aktuator. Sensor YL-69 diuji untuk mendeteksi kelembaban tanah dengan hasil pembacaan ADC yang menyesuaikan kondisi simulasi tanah basah dan kering. Sensor DHT22 berhasil membaca suhu dan kelembaban udara secara real-time pada simulasi Wokwi. Sensor LDR juga berfungsi baik dalam mendeteksi intensitas cahaya dengan variasi nilai lux yang berubah sesuai kondisi pencahayaan virtual. Aktuator pompa air dan motor servo berhasil menerima perintah dari ESP32 untuk aktif atau nonaktif sesuai dengan kondisi parameter lingkungan.

2) Integration Testing Sistem Otomatisasi

Setelah pengujian unit selesai, dilakukan pengujian integrasi di mana semua komponen bekerja secara bersamaan. ESP32 membaca data dari seluruh sensor, kemudian memproses logika threshold untuk mengaktifkan atau menonaktifkan aktuator. Data hasil pembacaan dikirimkan ke Firebase Realtime Database. Website monitoring berhasil menampilkan data secara real-time dan memberikan kontrol manual terhadap pompa dan atap rumah kaca.

3) Website Testing

Pengujian website difokuskan pada dua hal, yaitu tampilan monitoring dan fitur kontrol manual. Hasil pengujian menunjukkan bahwa dashboard website menampilkan data suhu, kelembaban tanah, dan intensitas cahaya secara real-time sesuai data yang diterima dari Firebase. Selain itu, kontrol manual seperti menghidupkan pompa atau membuka atap juga berhasil mengubah status pada Firebase dan diterima kembali oleh ESP32 virtual.

4) Iterative Testing sesuai XP

Seluruh pengujian dilakukan secara iteratif sesuai dengan prinsip XP. Setelah setiap iterasi pengkodean dan simulasi, dilakukan pengujian dan evaluasi ulang. Jika ditemukan kekurangan, perbaikan diterapkan pada iterasi berikutnya. Dengan

pendekatan ini, perbaikan logika dan stabilisasi sistem dapat dilakukan lebih cepat dan efisien.

5) Hasil Pengujian

Untuk mengetahui respons sistem terhadap perubahan kondisi lingkungan, dilakukan simulasi pada lima waktu yang berbeda dalam sehari, yaitu pukul 01:00 pagi, 09:00 pagi, 13:00 siang, 16:00 sore, dan 20:00 malam. Pengujian ini bertujuan untuk mengamati bagaimana sistem Smart Greenhouse merespon variabel suhu, kelembaban tanah, dan intensitas cahaya secara otomatis. Data hasil uji ditampilkan pada Tabel pengaturan atap berdasarkan intensitas cahaya yang belum diimplementasikan secara otomatis. Hasil pengujian secara detail dapat dilihat pada Tabel 5

Waktu	Suhu (°C)	Status Servo	Kelembaban Tanah (ADC)	Status Pompa Air	Intensitas Cahaya (lux)	Status Pencahayaan
01:00 Pagi	14	Terbuka	250	Aktif (Penyiraman)	1.000	Kurang Cahaya
09:00 Pagi	18	Tertutup	320	Tidak Aktif	12.000	Ideal
13:00 Siang	30	Tertutup	580	Tidak Aktif	18.000	Terlalu Terang
16:00 Sore	22	Tertutup	450	Tidak Aktif	8.000	Sedikit Kurang Cahaya
20:00 Malam	16	Tertutup	290	Aktif (Penyiraman)	500	Kurang Cahaya

Tabel 5 Hasil Uji Sistem

- Pada pukul 01:00 pagi, suhu yang rendah (14°C) menyebabkan servo membuka atap, dan kelembaban tanah yang rendah (250 ADC) memicu aktifnya pompa air secara otomatis. Intensitas cahaya pada waktu tersebut sangat rendah, sehingga status pencahayaan menunjukkan kondisi kurang cahaya.
- Pada pukul 09:00 pagi, suhu mulai meningkat menjadi 18°C dan kelembaban tanah berada pada level ideal (320 ADC), sehingga servo menutup atap dan pompa air tidak aktif. Intensitas cahaya sebesar 12.000 lux menunjukkan kondisi pencahayaan yang ideal untuk tanaman selada.
- Pada pukul 13:00 siang, suhu mencapai 30°C dan intensitas cahaya meningkat hingga 18.000 lux yang tergolong terlalu terang. Meskipun demikian, servo tetap dalam kondisi tertutup dan pompa air tidak aktif karena kelembaban tanah masih dalam batas ideal. Hal ini menunjukkan kekurangan pada logika sistem yang belum mengatur shading otomatis terhadap intensitas cahaya berlebih.

- d) Pada pukul 16:00 sore, suhu menurun menjadi 22°C dan kelembaban tanah masih cukup, sehingga pompa air tetap tidak aktif. Intensitas cahaya berkurang menjadi 8.000 lux, menghasilkan kondisi sedikit kurang cahaya yang masih bisa ditoleransi oleh tanaman.
- e) Pada pukul 20:00 malam, suhu turun menjadi 16°C dan kelembaban tanah kembali rendah (290 ADC), sehingga pompa air aktif secara otomatis untuk melakukan penyiraman. Intensitas cahaya hanya 500 lux, sehingga status pencahayaan kembali kurang cahaya.

6) Kesimpulan Pengujian

Dari hasil pengujian di atas, dapat disimpulkan bahwa sistem telah mampu merespon perubahan suhu dan kelembaban tanah secara otomatis sesuai logika yang dirancang. Pompa air dan servo bekerja sesuai kebutuhan lingkungan tanaman selada. Namun, pengaturan intensitas cahaya masih memerlukan pengembangan lebih lanjut untuk mengoptimalkan perlindungan tanaman terhadap cahaya berlebih. Secara keseluruhan, sistem menunjukkan respon otomatis yang baik pada sebagian besar parameter lingkungan yang diuji.

E. Listening

Tahap listening dalam metode Extreme Programming (XP) berfungsi untuk mengevaluasi hasil pengujian yang telah dilakukan serta merespon kebutuhan atau permasalahan yang muncul selama proses pengembangan. Pada tahap ini, tim pengembang melakukan diskusi internal dan merekap feedback dari hasil simulasi sistem Smart Greenhouse yang diuji pada platform Wokwi dan Firebase. Proses ini menjadi dasar untuk menentukan arah pengembangan selanjutnya agar sistem semakin optimal dan sesuai dengan kebutuhan.

Berdasarkan hasil pengujian yang telah dilakukan, terdapat beberapa evaluasi penting. Pertama, sistem telah mampu merespon perubahan suhu dan kelembaban tanah secara otomatis dengan tingkat keberhasilan tinggi. Pompa air aktif saat tanah dalam kondisi kering (< 300 ADC), dan atap rumah kaca berhasil membuka atau menutup sesuai parameter suhu yang ditetapkan. Hal ini menunjukkan bahwa logika otomatisasi penyiraman

dan pengaturan atap telah berjalan stabil dan sesuai dengan desain awal.

Namun demikian, masih ditemukan kekurangan minor pada pengaturan atap berdasarkan intensitas cahaya. Pada saat intensitas cahaya melebihi 15.000 lux, sistem belum secara otomatis menyesuaikan kondisi atap atau menambahkan shading untuk melindungi tanaman dari cahaya berlebih. Hal ini menjadi salah satu masukan utama untuk iterasi pengembangan selanjutnya, di mana tim merencanakan penambahan logika shading otomatis berbasis pembacaan sensor LDR.

Selain itu, melalui simulasi website berbasis Firebase, tim menerima masukan terkait respons waktu pembaruan data sensor. Meskipun data sensor sudah tampil secara real-time, terdapat potensi optimalisasi pada interval pengiriman data agar lebih hemat bandwidth tanpa mengurangi kecepatan update yang dibutuhkan oleh pengguna.

Dalam sesi evaluasi, tim juga mempertimbangkan pengembangan lanjutan berupa:

- a) Penambahan fitur pengaturan intensitas cahaya secara otomatis (misalnya dengan menambahkan tirai otomatis).
- b) Integrasi kontrol nutrisi tanaman secara otomatis pada iterasi pengembangan selanjutnya.
- c) Penyempurnaan algoritma sensor agar lebih akurat dalam mendeteksi perubahan lingkungan yang mendadak.

Semua temuan dan saran pada tahap listening ini menjadi dasar utama untuk iterasi berikutnya, sehingga pengembangan sistem terus bergerak adaptif dan responsif terhadap kebutuhan nyata. Dengan pendekatan XP, perbaikan-perbaikan tersebut dapat segera diimplementasikan pada siklus pengembangan selanjutnya tanpa harus menunggu penyelesaian sistem secara keseluruhan.

IV. KESIMPULAN

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan, dapat disimpulkan bahwa sistem Smart Greenhouse berbasis ESP32 dengan pengujian virtual melalui platform Wokwi dan Firebase telah berhasil dikembangkan secara iteratif menggunakan metode Extreme Programming (XP). Proses pengembangan sistem mengikuti lima tahapan utama XP, yaitu Planning, Design, Coding,

Testing, dan Listening, yang dilaksanakan secara bertahap dan berkelanjutan.

Pada tahap perencanaan (planning), kebutuhan utama Smart Greenhouse berhasil diidentifikasi dengan baik, meliputi otomatisasi penyiraman berbasis kelembaban tanah, pengaturan atap otomatis berdasarkan suhu dan pencahayaan, serta monitoring real-time menggunakan website berbasis Firebase. Tahap perancangan (design) menghasilkan arsitektur sistem yang sederhana namun fungsional, dengan pembagian komponen sensor, mikrokontroler, dan aktuator yang terintegrasi secara efektif.

Tahap pengkodean (coding) yang dilakukan melalui simulasi Wokwi menunjukkan bahwa ESP32 berhasil membaca data sensor YL-69, DHT22, dan LDR, serta memproses logika otomatisasi sesuai dengan parameter lingkungan yang telah ditentukan. Sistem juga berhasil mengirimkan data sensor secara real-time ke Firebase dan menampilkan data tersebut pada dashboard website monitoring.

Melalui tahap pengujian (testing), sistem menunjukkan performa yang cukup stabil dengan tingkat keberhasilan mencapai 93,3%. Sistem mampu merespon kondisi lingkungan dengan tepat, seperti mengaktifkan pompa saat tanah kering dan membuka atau menutup atap secara otomatis sesuai suhu ruangan. Namun demikian, masih terdapat kekurangan minor pada aspek pengaturan atap berdasarkan intensitas cahaya yang belum otomatis berjalan, sehingga memerlukan pengembangan lebih lanjut.

Pada tahap listening, hasil evaluasi dan umpan balik digunakan untuk merencanakan perbaikan pada iterasi selanjutnya. Beberapa perbaikan yang direncanakan meliputi penambahan logika shading otomatis berbasis sensor cahaya dan optimalisasi algoritma pengambilan keputusan untuk efisiensi energi serta komunikasi data yang lebih stabil.

Secara keseluruhan, pengembangan sistem Smart Greenhouse berbasis IoT ini telah mampu membuktikan bahwa pendekatan Extreme Programming efektif diterapkan pada proyek berbasis sensor dan aktuator yang memerlukan pengujian berulang dan fleksibilitas tinggi dalam pengembangan. Sistem berhasil mengotomatiskan proses pertanian hidroponik tanaman selada secara

real-time dan memberikan kemudahan monitoring jarak jauh melalui website berbasis Firebase.

Dengan hasil ini, diharapkan implementasi nyata dari sistem Smart Greenhouse dapat membantu meningkatkan efisiensi pertanian modern, mengurangi intervensi manual, serta mendukung pertanian berkelanjutan berbasis teknologi Internet of Things.

UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua individu dan pihak yang telah memberikan kontribusi dan dukungan selama penelitian ini. Tanpa kerja keras, panduan, dan bantuan mereka, penelitian ini tidak akan dapat terwujud. Secara khusus, penulis menyampaikan terima kasih kepada pembimbing, Bapak **Pramono, S.Kom., M.Kom.**, atas arahan, saran, dan pengawasan yang sangat berharga sepanjang proses penelitian ini. Ucapan terima kasih juga disampaikan kepada keluarga dan teman-teman penulis atas dukungan, motivasi, dan pengertian yang mereka berikan selama pelaksanaan penelitian. Tak lupa, penulis juga berterima kasih kepada institusi dan lembaga yang telah memberikan akses serta izin dalam pengumpulan data yang diperlukan. Segala bentuk dukungan dan kontribusi yang telah diberikan sangat berarti bagi kelancaran dan kesuksesan penelitian ini.

REFERENSI

- [1] Dekita Nuswantara, Aji Brahma Nugroho, dan Herry Setiyawan, "Desain Sistem Monitoring Pengontrolan Suhu, Kelembaban dan Sirkulasi Air Otomatis pada Tanaman Anggrek Hidroponik Berbasis Arduino Uno," *Jurnal Teknik Elektro, Universitas Muhammadiyah Jember*, 2020.
- [2] Kaushik Sekaran, Maytham N. Meqdad, Pardeep Kumar, Soundar Rajan, dan Seifedine Kadry, "Smart agriculture management system using internet of things," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 18, no. 3, pp. 1275–1284, 2020. doi: 10.12928/TELKOMNIKA.v18i3.14029.
- [3] Rahmad Doni dan Maulia Rahman, "Sistem Monitoring Tanaman Hidroponik Berbasis IoT (Internet of Thing) Menggunakan Nodemcu ESP8266," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 4, no. 2, pp. 516–522, Sept. 2020.
- [4] M. L. Q. Galon, M. V. R. Tumaliwan, dan M. M. Sejera, "Automated Monitoring and Control System of Solar Greenhouse Using ESP32 and Blynk Application," *Engineering Proceedings*, vol. 92, no. 1, p. 57, 2025, doi: 10.3390/engproc2025092057. MDPI
- [5] J. J. Correa-Quiroz, M. A. Toribio-Barrueto, dan C. Castro-Vargas, "IoT System with ESP32 for Smart Drip Irrigation and Climate Monitoring in Greenhouses," *Emerging Science Journal*, vol. 9, no. 3, pp. 1133–1157, 2025, doi: 10.28991/ESJ-2025-09-03-01. Ijournalse
- [6] S. S. Shanto, M. Rahman, J. M. Oasik, dan H. Hossain, "Smart Greenhouse Monitoring System Using Blynk IoT App," *Journal of*

- Engineering Research and Reports*, vol. 25, no. 2, pp. 94–107, 2023, doi: 10.9734/jerr/2023/v25i2883. journaljerr.com
- [7] Muthmainnah, N. Chamidah, dan K. Kusairi, “Monitoring Soil Temperature and Humidity in an IoT-Based Green House to Improve Plant Management,” *Engineering: Journal of Mechatronics and Education*, vol. 1, no. 2, 2024, doi: 10.59923/mechatronics.v1i2.129. ejournal.imbima.org
- [8] I. W. K. Kartika *dkk.*, “Aplikasi Sensor Soil Moisture YL-69 dan Sensor Ultrasonic HC-SR07 pada Smart Irrigation,” *Jurnal Ilmiah Teknologi Pertanian Agrotechno*, vol. 6, no. 1, 2021, doi: 10.24843/JITPA.2021.v06.i01.p05. Open Journal Systems
- [9] S. Asy-Syamil, M. Hidayatullah, dan T. Andriani, “Pemanfaatan Sensor Soil Moisture YL-69 sebagai Pendeteksi Kadar Air pada Tanah Berbasis Op-Amp,” *Renewable Energy Technologies Journal*, vol. 1, no. 1, pp. 7–10, 2023. Jurnal UTS
- [10] Lutfi, M. Mutmainnah, dan S. Buwarda, “Utilization of ChatGPT in Design and Simulation of Arduino-Based Electronic Products via the Wokwi Platform,” *Journal of Electrical and Automation Technology*, vol. 3, no. 1, 2024, doi: 10.61844/jeat.v3i1.966. journal.atim.ac.id
- [11] R. P. Tulodo *dkk.*, “Penggunaan Simulator Wokwi untuk Meningkatkan Literasi Pemrograman Mikrokontroler dalam Proyek Internet of Things,” *EDUSAINTEK: Jurnal Pendidikan, Sains dan Teknologi*, vol. 12, no. 1, 2024, doi: 10.47668/edusaintek.v12i1.1442. journalstkipgrisitubondo.ac.id
- [12] L. Pamungkas, P. Rahardjo, dan I. G. A. P. R. Agung, “Simulasi Rancang Bangun Kontrol Jemuran dan Monitoring Suhu dengan ESP32 pada Wokwi dan Blynk,” *Journal Information & Computer*, vol. 2, no. 2, 2024. Open Journal
- [13] J. Riahi, S. Vergura, D. Mezghani, dan A. Mami, “Intelligent Control of the Microclimate of an Agricultural Greenhouse Powered by a Supporting PV System,” *Applied Sciences*, vol. 10, no. 4, Art. 1350, 2020, doi: 10.3390/app10041350. MDPI
- [14] B. Mihai, “How to Use the DHT22 Sensor for Measuring Temperature and Humidity with the Arduino Board,” *Acta Universitatis Cibiniensis – Technical Series*, vol. 68, no. 1, pp. 23–30, 2016, doi: 10.1515/aucts-2016-0005