

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Bab ini menyajikan hasil dari implementasi dan pengujian sistem *monitoring* dan kontrol suhu ruangan serta kelembaban media tanam pada lahan budidaya jamur tiram berbasis IoT dan web. Penelitian ini mencakup proses integrasi antara NodeMCU dengan sensor DHT22 untuk membaca data suhu dan kelembaban, serta pengendalian pompa air menggunakan modul *relay* secara otomatis.

Data yang dikumpulkan dikirim secara berkala ke server melalui protokol HTTPS dan disimpan dalam *database* PostgreSQL, kemudian ditampilkan dalam *dashboard* berbasis web. Evaluasi dilakukan berdasarkan kestabilan pengiriman data, ketepatan respon kontrol otomatis, dan kesesuaian data yang ditampilkan pada *dashboard*. Hasil implementasi menunjukkan bahwa sistem bekerja dengan baik dan mampu menjalankan fungsinya secara stabil sesuai dengan tujuan yang dirancang dalam penelitian ini.

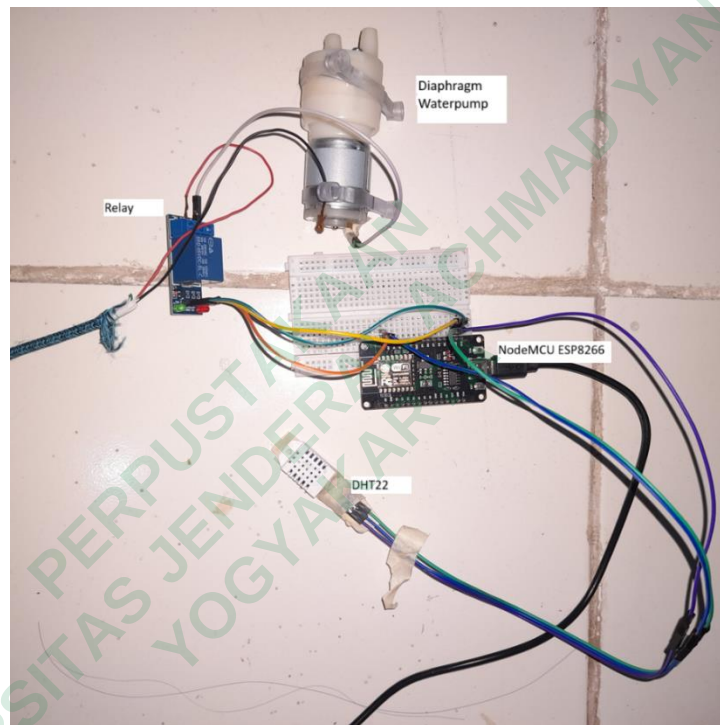
4.2 IMPLEMENTASI *HARDWARE*

Implementasi *hardware* ini mencakup penggunaan sensor DHT22 sebagai alat pemantau suhu dan kelembaban media tanam, serta penggunaan NodeMCU ESP8266 sebagai mikrokontroler utama yang terhubung ke internet melalui *Wi-Fi*. Selain itu, sistem ini juga dilengkapi dengan modul *relay* untuk mengontrol pompa air yang terhubung ke *nozzle* penyiram otomatis. Saat suhu terdeteksi lebih dari 28°C dan kelembaban kurang dari 80%, sistem akan mengaktifkan pompa air secara otomatis, kemudian mematikannya kembali untuk menjaga kondisi lingkungan budidaya jamur tetap ideal.

4.2.1 Implementasi Arsitektur Sistem

Tahap implementasi dilakukan dengan merealisasikan perangkat keras berdasarkan rancangan arsitektur sistem yang telah ditetapkan sebelumnya. Sistem

ini terdiri dari beberapa komponen utama yaitu sensor DHT22, NodeMCU ESP8266, modul *relay*, dan pompa air. Semua komponen tersebut dirangkai dan dipasang di area budidaya jamur tiram secara langsung. Implementasi ini bertujuan agar sistem dapat membaca suhu dan kelembaban secara *real-time*, serta melakukan penyiraman otomatis apabila suhu melebihi ambang batas yang ditentukan. Berikut hasil implementasi dari perangkat keras yang telah dirakit dan siap digunakan ditunjukkan pada gambar 4.1.



Gambar 4.1 Alat Monitoring

4.2.2 Inisialisasi Perangkat

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>
#include <WiFiClientSecure.h>

// Setting WiFi
const char* ssid = "Pandan D";
const char* password = "takonibu354";

// API Endpoint
```

```

const char* apiUrl = "https://api.Gardenision.com/api";

const char* serial_number = "homestaycrak1";

const char* bearer_token =
"29|53ZwWGUVeK99CpwrBhZTe0xKp60fMzwTBsTqdQVD7efab9bf";

// Pin setup
#define DHTPIN D1          // DHT22 di D1 (GPIO5)
#define DHTTYPE DHT22      // Tipe sensor
#define RELAYPIN D0        // Relay di D0 (GPIO16)

DHT dht(DHTPIN, DHTTYPE);

```

Kode di atas berfungsi untuk menghubungkan NodeMCU ke jaringan *WiFi*, mengatur *pin input* dari sensor suhu dan kelembaban dari sensor DHT22, serta *pin output* menuju *relay* sebagai pemicu pompa air.

4.2.3 Pembacaan Data Sensor

Sensor DHT22 digunakan untuk mengambil data suhu dan kelembaban secara berkala. Nilai yang diperoleh disimpan dalam objek JSON dan kemudian dikirimkan ke server:

```

JsonObject temperature = array.add<JsonObject>();
temperature["id"] = 7;
temperature["value"] = dht.readTemperature();

JsonObject humidity = array.add<JsonObject>();
humidity["id"] = 6;
humidity["value"] = dht.readHumidity();

```

4.2.4 Otomatisasi Penyiraman dan Kontrol Pompa

Kontrol pompa air pada sistem ini memiliki dua mode, yaitu otomatis dan manual, yang ditentukan oleh pengaturan dari server. Pada mode manual, pompa akan menyala atau mati sesuai perintah langsung dari dashboard. Sedangkan pada mode otomatis, pompa akan menyala jika suhu lebih dari 28°C atau kelembaban kurang dari 80%, dan akan mati kembali jika suhu turun sampai 28°C atau kurang, atau kelembaban naik hingga 80% atau lebih. Sistem ini memudahkan pengguna

dalam mengatur pompa sesuai kondisi lingkungan. Berikut cuplikan logika kontrolnya:

```
if (controlMode == "manual") {
  if (pumpShouldOn && !pumpOn) {
    digitalWrite(RELAYPIN, LOW); // Pompa ON
  } else if (!pumpShouldOn && pumpOn) {
    digitalWrite(RELAYPIN, HIGH); // Pompa OFF
  }
} else {
  if (!pumpOn && (lastTemp > 28.0 || lastHumidity < 80)) {
    digitalWrite(RELAYPIN, LOW); // Pompa ON
  } else if (pumpOn && (lastTemp <= 28.0 || lastHumidity >= 80)) {
    digitalWrite(RELAYPIN, HIGH); // Pompa OFF
  }
}
```

Jika kondisi suhu memenuhi syarat, *relay* akan aktif dan pompa air akan menyiram media tanam melalui *nozzle*. Sistem ini sangat berguna untuk menjaga kelembaban optimal pada budidaya jamur tiram.

4.2.5 Pengiriman Data ke Server

Data dikirim ke server melalui protokol HTTPS menggunakan metode POST. Proses ini memanfaatkan pustaka ESP8266HTTPClient dan *ArduinoJson*. Pengiriman dilakukan jika terdapat perubahan pada nilai suhu, kelembaban, atau status pompa. Dengan demikian, sistem lebih efisien dalam penggunaan *bandwidth*.

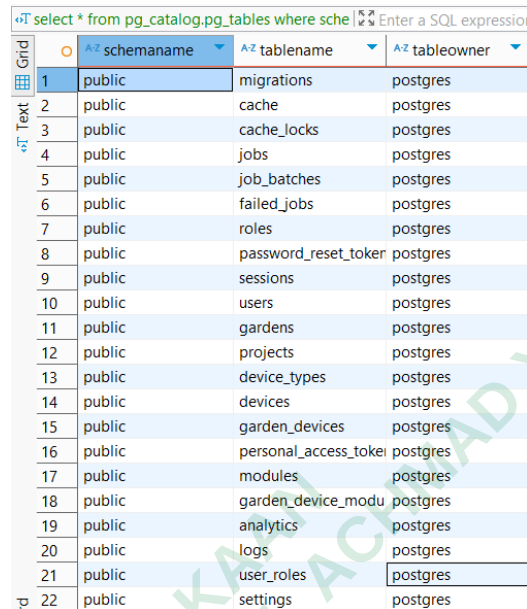
```
void sendData(int id, const char* value) {
  ...
  int httpResponseCode = http.POST(requestBody);
  ...
}
```

4.3 IMPLEMENTASI SOFTWARE

4.3.1 Database

Dalam pengembangan dashboard informasi pada sistem ini, diperlukan penyimpanan dan pengambilan data secara terstruktur melalui sebuah database. Sistem ini menggunakan PostgreSQL sebagai sistem manajemen basis data, yang diakses melalui antarmuka DBeaver. PostgreSQL mendukung pengolahan data

dengan menggunakan bahasa standar SQL (*Structured Query Language*). Gambar database dapat dilihat pada gambar 4.2.



	schemaname	tablename	tableowner
1	public	migrations	postgres
2	public	cache	postgres
3	public	cache_locks	postgres
4	public	jobs	postgres
5	public	job_batches	postgres
6	public	failed_jobs	postgres
7	public	roles	postgres
8	public	password_reset_tokens	postgres
9	public	sessions	postgres
10	public	users	postgres
11	public	gardens	postgres
12	public	projects	postgres
13	public	device_types	postgres
14	public	devices	postgres
15	public	garden_devices	postgres
16	public	personal_access_tokens	postgres
17	public	modules	postgres
18	public	garden_device_modules	postgres
19	public	analytics	postgres
20	public	logs	postgres
21	public	user_roles	postgres
22	public	settings	postgres

Gambar 4.2 Database

4.3.2 Pengembangan API Backend

Backend dalam sistem ini dibangun menggunakan *framework* Laravel yang berbasis bahasa pemrograman PHP. Laravel dipilih karena memiliki struktur kerja yang jelas, mendukung pengolahan data secara efisien, serta mudah diintegrasikan dengan *database* dan antarmuka pengguna. Fungsi utama *backend* ini adalah menerima data suhu, kelembaban, dan status pompa dari perangkat NodeMCU yang dikirim melalui protokol HTTP dalam format JSON. Data tersebut kemudian disimpan ke dalam *database* PostgreSQL untuk keperluan pemantauan dan pencatatan. Berikut merupakan cuplikan kodenya.

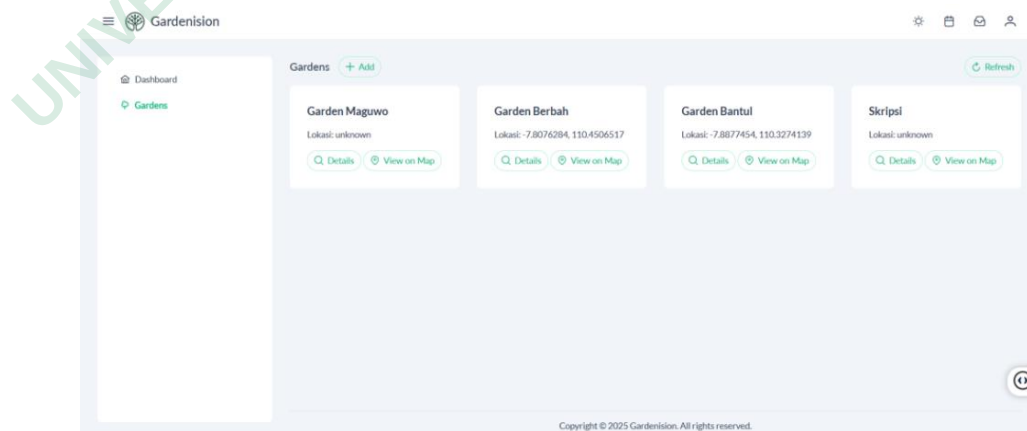
Cuplikan kode di atas menunjukkan bahwa ketika data dari perangkat diterima, sistem akan memperbarui nilai sensor (*unit_value*) dan mencatat informasi tersebut ke dalam log. *Backend* juga memastikan keamanan dengan mewajibkan penggunaan *Bearer Token* untuk setiap permintaan, sehingga hanya perangkat yang sah yang dapat mengakses API. Setelah data berhasil disimpan, sistem akan memicu *event* seperti *AddLog* dan *GardenDeviceModuleUpdated* yang

berfungsi untuk menampilkan pembaruan secara *real-time* di *dashboard* tanpa perlu menyegarkan halaman secara manual.

```
public function store(StoreRequest $request, string
$serial_number, ?Module $module)
{
    ...
    if ($garden_device_module) {
        $where['loggable_type'] =
        GardenDeviceModule::class;
        $where['loggable_id'] = $garden_device_module->id;
        $garden_device_module->unit_value = $request->context['value'];
        $garden_device_module->save();
        event(new
        GardenDeviceModuleUpdated($garden_device_module->toArray(), $garden_device_module->garden_device->garden->user_id));
        ...
    }
    ...
    $log = Log::create($updated_data);
    event(new AddLog($log));
    ...
}
```

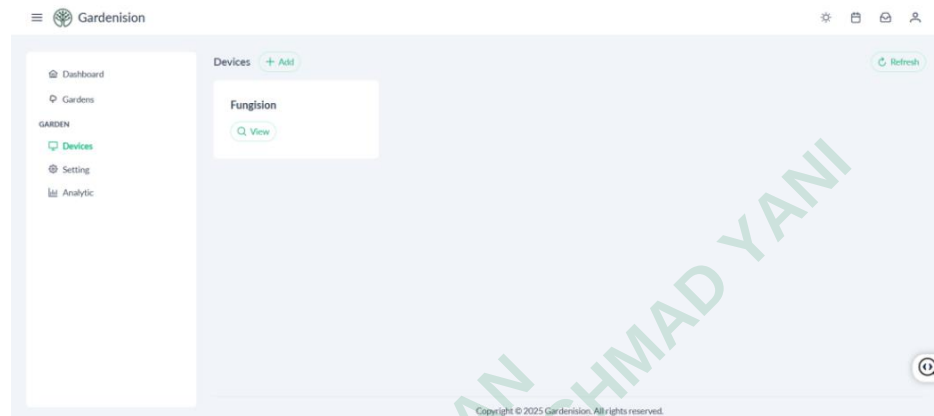
4.3.3 Pengembangan Web

Di dalam web yang telah dibuat terdapat *interface dashboard* yang menyajikan tampilan untuk memantau kondisi lingkungan jamur tiram. Untuk masuk ke halaman ini pengguna harus masuk terlebih dahulu ke *project garden* yang sebelumnya telah dibuat, sebagaimana terlihat pada gambar 4.3.



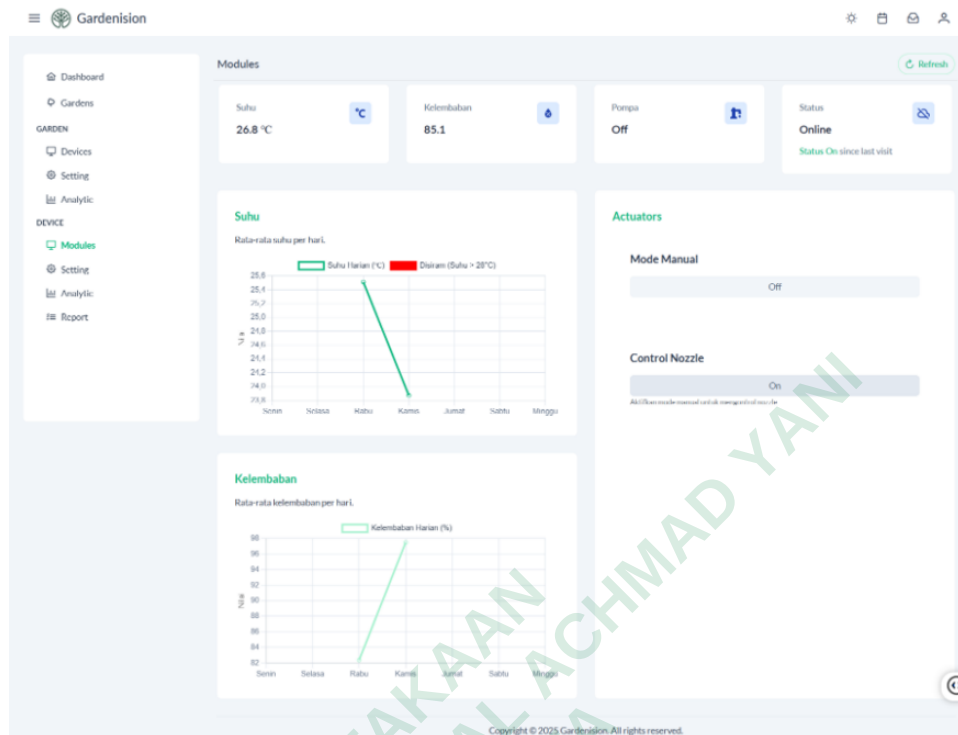
Gambar 4.3 Halaman *Project Garden*

Kemudian setelah masuk ke halaman *project garden*, pengguna bisa masuk ke salah satu *project garden* yang sudah dibuat dan akan menampilkan jenis *device* atau jenis alat yang digunakan pada suatu tanaman, yang ditunjukkan pada gambar 4.4.



Gambar 4.4 Halaman *Device*

Setelah pengguna berhasil masuk ke dalam proyek budidaya jamur tiram dan memilih jenis perangkat yang digunakan, sistem akan menampilkan, utama. Melalui tampilan ini, pengguna dapat melakukan *monitoring* terhadap data suhu dan kelembaban secara *real-time*, serta melakukan kontrol terhadap perangkat seperti pompa air. Adapun antarmuka *dashboard* yang digunakan untuk melakukan *monitoring* dan kontrol ditunjukkan pada gambar 4.5.



Gambar 4.5 Halaman *Dashboard*

4.3.4 Integrasi *Backend* dan *Frontend*

Setelah API dan *dashboard* dikembangkan, tahap integrasi dilakukan agar sistem berjalan secara utuh. *Dashboard* memanfaatkan API untuk mengambil dan menampilkan data sensor secara berkala. Komunikasi antara *frontend* dan *backend* dilakukan menggunakan metode HTTP *request* berbasis JSON. Selain itu, sistem ini juga memungkinkan pengguna mengirim perintah dari *dashboard* ke perangkat IoT untuk melakukan penyiraman secara manual.

4.4 PENGUJIAN KESELURUHAN SISTEM

Pembahasan ini bertujuan untuk menguraikan dan menganalisis hasil implementasi sistem *monitoring* dan kontrol suhu ruangan serta kelembaban media tanam pada lahan budidaya jamur tiram berbasis IoT dan web. Analisis difokuskan pada keterkaitan antara hasil yang diperoleh dengan tujuan penelitian, serta mengidentifikasi keunggulan sistem, kendala yang dihadapi, dan peluang pengembangan di masa mendatang.

4.4.1 Pengujian *Hardware*

Pengujian perangkat keras dilakukan untuk mengevaluasi kinerja sensor suhu dan kelembaban, serta memastikan bahwa NodeMCU mampu mengeksekusi skrip yang telah diprogramkan untuk mengontrol pompa air secara otomatis sesuai kondisi lingkungan. Hasil dari pengujian ini disajikan pada tiga tabel yaitu tabel 4.1 sampai 4.3.

Tabel 4.1 Tabel Pengujian Konektifitas NodeMCU

Pengujian	Langkah Uji	Hasil yang Diharapkan	Status
Koneksi NodeMCU ke jaringan <i>WiFi</i>	Menyalakan NodeMCU dan memantau Serial Monitor	Pesan “ <i>WiFi connected</i> ” muncul pada Serial Monitor dan data suhu, kelembaban serta status pompa terbaca di <i>dashboard</i> web	Berhasil

Tabel 4.2 Tabel Pengujian Sensor DHT22

Kondisi Lingkungan	Suhu	Kelembaban	Hasil yang Diharapkan	Status
Udara ruangan normal	26°C–28°C	80% - 90%	Nilai tampil dan stabil di <i>dashboard</i>	Berhasil
Tingkat suhu udara dipanaskan	32°C	70%	Nilai suhu meningkat	Berhasil
Udara dilembabkan	26.5°C	90%	Nilai kelembaban meningkat	Berhasil

Tabel 4.3 Tabel Pengujian Kontrol Otomatis

Kondisi Simulasi	Suhu	Kelembaban	Aksi Sistem	Hasil yang Diharapkan	Status
Suhu tinggi (>28°C)	34°C	60%	Pompa menyala	Menyiram otomatis	Berhasil
Kelembaban rendah (<80%)	28.5°C	77%	Pompa menyala	Melembabkan otomatis	Berhasil

Kondisi Simulasi	Suhu	Kelembaban	Aksi Sistem	Hasil yang Diharapkan	Status
Suhu & kelembaban ideal	27.3°C	82%	Pompa menyala	Tidak menyiram	Berhasil

4.4.2 Pengujian Software

Pengujian *dashboard* informasi dilakukan untuk memastikan bahwa tampilan web mampu menyajikan data suhu dan kelembaban yang dibutuhkan oleh pengguna, khususnya dalam konteks pemantauan budidaya jamur tiram. Pengujian dilakukan dengan melakukan beberapa simulasi dan observasi terhadap fungsi visualisasi data dan keakuratan informasi yang ditampilkan pada *dashboard*. Rangkuman hasil pengujian dapat dilihat pada Tabel 4.4.

Tabel 4.4 Tabel Pengujian pada Software

Aspek Pengujian	Ekspektasi Sistem	Status
Menampilkan informasi kondisi suhu dan kelembaban.	<i>Dashboard</i> dapat memberikan informasi tentang kondisi suhu dan kelembaban.	Berhasil
Menampilkan status pompa.	Status pompa <i>on/off</i> ditampilkan sesuai kondisi sebenarnya pada sistem fisik.	Berhasil
Visualisasi grafik suhu dan kelembaban.	Sistem menampilkan grafik rata-rata harian untuk suhu dan kelembaban secara informatif.	Berhasil
Kontrol aktuator untuk <i>nozzle sprayer</i> .	Tombol kontrol pada <i>dashboard</i> dapat menghidupkan dan mematikan <i>nozzle</i> secara manual.	Berhasil

4.4.3 Fungsionalitas

Pengujian sistem dilakukan untuk memastikan bahwa perangkat keras dan *dashboard* dapat bekerja secara sinkron dalam menampilkan data. Pengujian dilakukan dari sore hari sampai siang hari, adapun hasil pengujian sistem disajikan pada tabel 4.5 dan tabel 4.6.

Tabel 4.5 Tabel Pengujian *Hardware*

Komponen	Skenario Pengujian	Hasil yang Diharapkan	Status
NodeMCU ESP8266	Diberi catu daya dan dihubungkan ke <i>Wi-Fi</i>	Perangkat menyala dan terkoneksi <i>Wi-Fi</i>	Berhasil
Sensor DHT22	Sensor membaca suhu dan kelembaban	Nilai suhu dan kelembaban tampil di web	Berhasil
Kabel <i>jumper</i>	Koneksi antar komponen	Koneksi stabil, tidak longgar	Berhasil
<i>Relay</i>	Perintah ON/OFF dari <i>dashboard</i>	<i>Relay</i> menyala/mati sesuai perintah	Berhasil
Pompa	<i>Diaktifkan/dimatikan</i> oleh <i>relay</i>	Pompa menyala jika <i>relay</i> ON, mati jika <i>relay</i> OFF	Berhasil

Tabel 4.6 Tabel Pengujian *Software*

Fitur	Skenario Pengujian	Hasil yang Diharapkan	Status
Tampilan sensor	Data suhu & kelembaban ditampilkan secara <i>realtime</i>	Data tampil sesuai nilai sensor	Berhasil
Kontrol pompa	Menekan tombol ON/OFF pada <i>dashboard</i>	Pompa menyala/mati sesuai perintah	Berhasil
Grafik rata-rata harian suhu & kelembaban	Data historis sensor ditampilkan dalam grafik per hari	Grafik menampilkan rata-rata suhu & kelembaban harian dengan benar	Berhasil

4.5 PEMBAHASAN

Sistem *hardware* dan *software* yang dikembangkan dalam penelitian ini telah berhasil diterapkan pada lahan budidaya jamur tiram untuk memantau dan mengontrol suhu serta kelembaban media tanam secara otomatis. Sistem mampu membaca data suhu dan kelembaban melalui sensor DHT22 yang terhubung dengan NodeMCU, serta memberikan respon otomatis dengan mengaktifkan pompa air apabila suhu melebihi 28°C atau kelembaban turun di bawah 80%. Data sensor dikirim ke server melalui koneksi *Wi-Fi* dan disimpan ke dalam *database* PostgreSQL, yang kemudian divisualisasikan melalui *dashboard* web secara *real-time*.

Namun demikian, terdapat beberapa keterbatasan pada implementasi sistem. Pertama, sistem belum dilengkapi dengan fitur notifikasi otomatis ketika terjadi kondisi ekstrem, seperti suhu terlalu tinggi atau kelembaban sangat rendah. Hal ini berpotensi mengurangi kecepatan respon pengguna terhadap situasi kritis. Kedua, selama implementasi ditemukan permasalahan pada kabel penghubung sensor dan *relay* yang mudah longgar, sehingga mengganggu stabilitas sistem. Ketiga, sensor DHT22 yang digunakan belum melalui proses kalibrasi terhadap alat ukur standar, sehingga tingkat akurasi data yang ditampilkan belum dapat dipastikan secara valid. Oleh karena itu, untuk meningkatkan keandalan sistem ke depan, perlu dilakukan penambahan fitur notifikasi, perbaikan pada metode pemasangan kabel, serta pelaksanaan kalibrasi sensor terhadap alat ukur konvensional seperti termometer dan higrometer manual. Selain itu, pengujian lanjutan dalam kondisi lingkungan yang bervariasi juga disarankan guna mengevaluasi ketahanan dan stabilitas sistem secara menyeluruh.