

Rancang Bangun Sistem Kontrol Kelembaban Udara *Greenhouse* Menggunakan PWM (*Pulse Width Modulation*)

*The Development of Greenhouse Air Humidity Control System Using PWM (*Pulse Width Modulation*)*

Wily Goldramijaya, Ni Nyoman Sulastri*, I Made Anom S. Wijaya, I Putu Gede Budisanjaya

Program Studi Teknik Pertanian dan Biosistem, Fakultas Teknologi Pertanian, Universitas Udayana, Badung, Bali, Indonesia

*Email: Sulastri@unud.ac.id

Abstrak

Sistem kontrol banyak diterapkan pada bidang pertanian seperti penggunaan mikrokontroler dalam mengontrol dan memantau kondisi kelembaban udara dalam *greenhouse*. Ketepatan pengkondisian kelembaban udara merupakan hal penting bagi pertumbuhan dan perkembangan tanaman seperti *microgreen*. Kelembaban media tanam secara tidak langsung mempengaruhi kelembaban udara melalui penguapan, sehingga diperlukan sistem yang dapat mengontrol dan memantau kelembaban udara sekaligus kelembaban media tanam. Penelitian ini bertujuan untuk merancang dan membangun sistem kontrol kelembaban udara menggunakan PWM (*Pulse Width Modulation*) dalam *greenhouse* yang dilengkapi sistem *monitoring* dan pemberian irigasi otomatis. Rancang bangun sistem yang dihasilkan pada penelitian ini meliputi penggunaan Arduino UNO ATmega328P sebagai mikrokontroler yang terhubung dengan sensor DHT22 sebagai sensor kelembaban udara, *capacitive soil moisture* sensor sebagai sensor kelembaban media tanam, relay sebagai kontrol pompa *misting* dan irigasi, *driver* sebagai pengendali kipas DC dengan kontrol PWM, kipas DC sebagai pemberi aliran udara dan menurunkan kelembaban udara, dan LCD 16x2 sebagai sistem *monitoring*. Sistem menggunakan *setting point* kelembaban udara untuk *microgreen* yang dimasukkan secara manual melalui *coding*. Sistem yang dirancang menghasilkan miniatur *greenhouse* bertipe *lean-to* yang dilengkapi oleh sensor DHT22 dengan nilai *error* kelembaban udara DHT22(1) 3,09% dan DHT22(2) 3,34%. Sistem berhasil membaca dan menampilkan data, dan memberikan *output* dengan *response time* sistem irigasi dan *misting* sebesar 78mS dan 145mS, dengan *delay* 2 detik. Kendali PWM kipas DC mampu menurunkan dan menjaga kelembaban udara tetap pada rentang optimal bagi pertumbuhan *microgreen*, dengan meningkatkan dan menurunkan nilai RPM kipas DC sesuai dengan nilai *duty cycle* nya.

Kata Kunci: *pulse width modulation, kelembaban udara, microgreen, greenhouse, arduino.*

Abstract

Control systems are widely applied in agriculture, such as using microcontrollers to control humidity in greenhouses. Lack of accuracy in conditioning the humidity in indoor farming affects plant growth and productivity, especially in microgreens. The humidity of the planting media indirectly affects the air humidity through evaporation. Therefore, a system is required to control and monitor air and planting media humidity. The purpose of this study was to design and build an air humidity control system using PWM (*Pulse Width Modulation*) in a greenhouse equipped with a monitoring and automatic irrigation system. The system control employed Arduino UNO ATmega328P as a microcontroller connected to the DHT22 sensor as an air humidity sensor, a capacitive soil moisture sensor as a humidity sensor for the planting media, relay to control misting and irrigation pump. The microcontroller was also connected to a driver as a DC fan control with PWM control, four DC fan as airflow suppliers and dehumidifiers, and LCD 16x2 as a monitoring system. The system used the air humidity setting points for microgreens which were entered manually through coding. The result of this research was a lean-to greenhouse prototype with a DHT22 sensor with air humidity error values of DHT22(1) 3.09% and DHT22(2) 3.34%. The system successfully read and displayed data and provided output with a response time of irrigation and misting systems of 78mS and 145mS, using a delay of 2 seconds. The PWM control of the DC fan was able to reduce and maintain the air humidity within the optimum range for microgreen growth by increasing as well as decreasing the DC fans' RPM, which was based on its duty cycle value.

Keywords: *pulse width modulation, air humidity, microgreen, greenhouse, arduino*

PENDAHULUAN

Sistem kontrol menjadi salah satu bagian dari kehidupan yang banyak ditemukan pada industri, salah satunya pertanian seperti penggunaan mikrokontroler (Indriani et al., 2017). Diperlukan ketelitian dalam proses industri seperti mengontrol kelembaban (Yudaningtyas, 2017). Saat ini penggunaan teknologi sistem kontrol dalam menerapkan otomatisasi masih menggunakan metode kontrol *on-off* saja (Karsid et al., 2018). Dalam mengontrol iklim mikro seperti kelembaban diperlukan *setting point* sebagai batas atau rentang minimum dan maksimum kelembaban yang diinginkan. Kontrol *on-off* memiliki kelemahan yaitu tidak mampu mencapai titik kestabilan karena tidak mampu mencapai titik *setting point*, namun hanya berosilasi di sekitar *setting point* (Hutabarat, 2018). Pada sistem kontrol *on/off* hanya bekerja dengan dua kondisi yaitu posisi *high (on)* dan posisi *low (off)*, tidak akan pernah bekerja di daerah 0% sampai 100% (Megido & Ariyanto, 2016).

Dalam teknologi pengontrolan terdapat metode lain selain kontrol *on-off* yaitu PWM (*Pulse Width Modulation*) (Megido & Ariyanto, 2016). PWM merupakan salah satu teknik modulasi dengan mengubah lebar pulsa atau *duty cycle* dengan nilai amplitudo dan frekuensi yang tetap (Ripai & Wibowo, 2016). Arduino menyediakan fasilitas PWM pada *pin* digitalnya. Pengaplikasian kontrol PWM salah satunya digunakan untuk mengatur kecepatan motor DC. Kontrol PWM mampu bekerja dengan rentang 0% sampai 100%, sehingga banyak variasi kecepatan pada motor DC. Pada penelitian yang dilakukan oleh Arifin dan Fathoni (2014), menggunakan PWM dalam mengontrol kecepatan motor pada robot dapat bergerak dengan baik sesuai instruksi. Selain itu, penelitian yang dilakukan oleh Karsid et al (2018), menggunakan PWM pada mesin penetas telur otomatis dan membandingkan antara penggunaan kontrol PWM dengan kontrol *on-off*, dimana kontrol PWM memberikan hasil yang baik dari segi konsumsi daya dan grafik yang stabil dalam mencapai *setting point* dibandingkan dengan kontrol *on-off*.

Sistem kontrol PWM pada penelitian ini akan diaplikasikan menggunakan *setting point* kelembaban udara *microgreen* di dalam *greenhouse*. *Microgreen* merupakan tanaman yang layak konsumsi dan dipanen diantara fase perkecambahan dan *babygreen* yang berusia sekitar 7 sampai 14 hari setelah semai (As'adiya & Murwani, 2021). Produksi *microgreen* biasanya dilakukan di dalam *greenhouse* yang berteknologi, dikarenakan memerlukan perlindungan dari faktor lingkungan (Anonim, 2017;

Renna et al., 2017). Hal ini dikarenakan, *greenhouse* mampu memberikan faktor lingkungan yang mendekati optimum bagi pertumbuhan tanaman (Alahudin, 2013).

Kelembaban udara merupakan salah satu faktor yang penting bagi pertumbuhan dan perkembangan tanaman. Kelembaban udara yang terlalu rendah akan menyebabkan laju transpirasi yang tinggi, sehingga akan terjadi kehilangan air pada tanaman, dan sebaliknya kelembaban yang tinggi akan mengakibatkan penurunan penyerapan unsur hara (Nurnasari & Djumali, 2016). Selain kelembaban udara, kelembaban media tanam yang juga memiliki pengaruh dalam pertumbuhan tanaman dan secara tidak langsung mempengaruhi kelembaban udara melalui penguapan. Seiring peningkatan kelembaban pada media tanam akan berdampak pada peningkatan persentase pertumbuhan penyakit seperti jamur (Apriast et al., 2015).

Permasalahan penyakit seperti jamur sering sekali terjadi pada tanaman akibat dari kondisi kelembaban yang tinggi, khususnya *microgreen* (Nolan, 2018) karena kerapatan benih yang berdekatan sehingga kelembaban akan terperangkap. Hal tersebut menandakan bahwa *microgreen* sangat sensitif terhadap kelembaban, baik kelembaban udara maupun kelembaban media tanam. Pengendalian kelembaban dalam *greenhouse microgreen* dapat dilakukan dengan pemberian aliran udara, sehingga dapat mencegah jamur tumbuh di *microgreen* (Grooms, 2020). Aliran udara dapat diberikan dengan penerapan kipas yang dikontrol di dalam *greenhouse*.

Dalam penelitian ini akan berfokus pada rancang bangun sistem kontrol kelembaban udara dengan tujuan merancang dan membangun sistem kontrol, serta menguji kinerja alat yang dibangun. Sistem kontrol pada penelitian ini menggunakan PWM sebagai dasar metode kontrol kecepatan pada kipas DC. Kipas berperan sebagai pemberi aliran udara dan menurunkan kelembaban udara pada *greenhouse* yang bertujuan untuk menjaga kelembaban udara tetap pada *setting point* optimal. Penelitian akan menggunakan *setting point microgreen* sebagai penentu rentang kelembaban udara yang optimal. Sistem dilengkapi oleh sistem *misting* sebagai pemberian kelembaban udara dan irigasi secara *on/off* otomatis.

METODE PENELITIAN

Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Laboratorium Pengelolaan Sumber Daya Alam dan Laboratorium Sistem Manajemen Keteknikan Pertanian, Fakultas

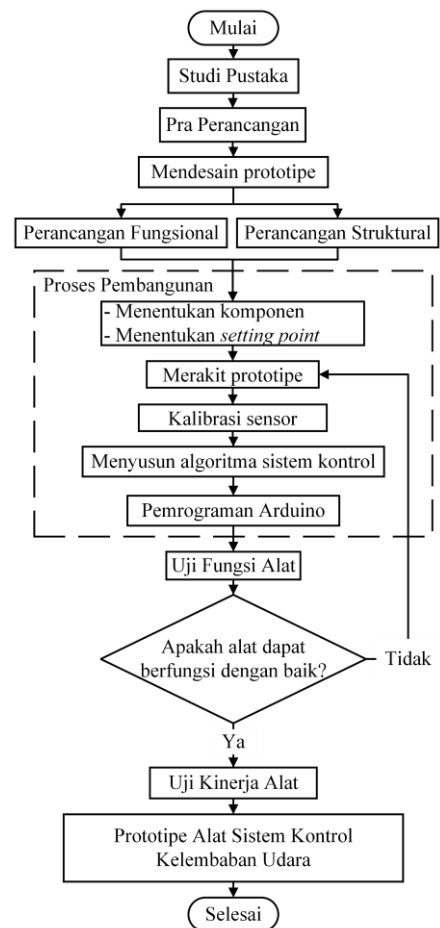
Teknologi Pertanian, Universitas Udayana, Kampus Sudirman. Pelaksanaan penelitian ini dilakukan pada bulan Maret 2022 sampai dengan bulan Mei 2022.

Bahan dan Alat

Bahan yang digunakan pada penelitian ini adalah Arduino UNO ATmega328P, 2 buah sensor DHT22, *Capacitive Soil Moisture Sensor*, relay 2 channel 5V, pompa DC 12V, pompa AC 220V, 4 buah kipas DC 12V, regulator *step down* DC-DC LM2596, 4 buah *driver* motor DC (transistor NPN, resistor, dioda, kapasitor), LCD 16x2, besi *hollow*, plastik UV, *power supply* DC 12V, *breadboard*, *nozzle misting*, *tray*, *cocopeat*, serta beberapa komponen pendukung. Alat yang dibutuhkan, yaitu laptop, multimeter, solder, osiloskop, tachometer, *hygro-thermometer*, *soil survey instrument*, *stopwatch*, perangkat lunak Arduino IDE, Google SketchUp, *fritzing*, serta alat pendukung lainnya.

Diagram Alir Penelitian

Konsep dasar perancangan sistem kontrol kelembaban udara menggunakan PWM akan dilakukan dengan tahap – tahap yang dapat dilihat pada diagram alir penelitian yang disajikan pada Gambar 1. Tahapan awal penelitian dilaksanakan dengan studi pustaka sebagai dasar pengkajian dan pemahaman mengenai PWM, *greenhouse*, *microgreen* dan proses pengendalian kelembaban menggunakan Arduino UNO.

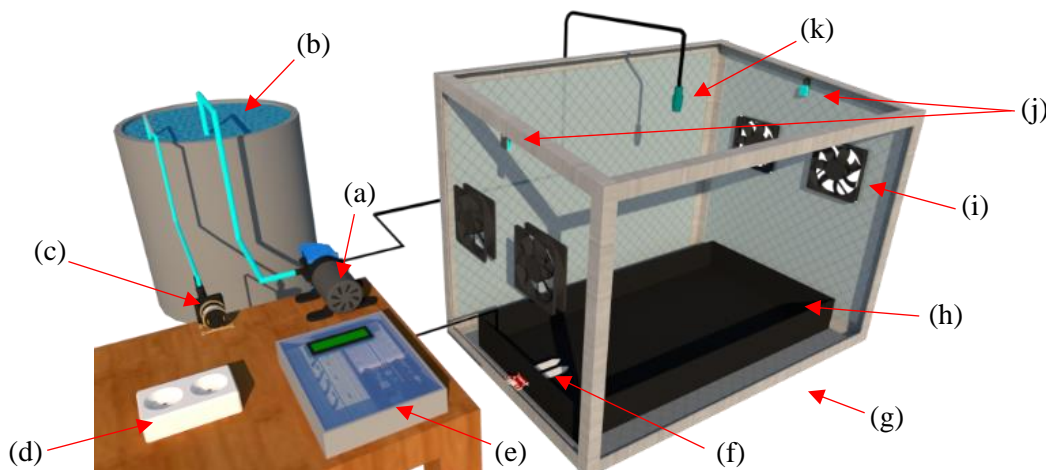


Gambar 1. Diagram Alir Penelitian

Rancangan Sistem Kontrol Kelembaban Udara

Sistem kontrol kelembaban udara yang akan dibangun terdiri dari beberapa perangkat keras yaitu

unit kontrol, unit sensor, aktuator, dan unit pemantauan, serta komponen penunjang lainnya yang dapat dilihat pada Gambar 2.



Gambar 2. Desain Rancangan Prototipe Sistem kontrol Kelembaban Udara

Keterangan: (a) Pompa *Misting*, (b) Sumber Air, (c) Pompa Irigasi, (d) Sumber Listrik, (e) Unit Kontrol, (f) *Capacitive Soil Moisture Sensor*, (g) Struktur Miniatur *Greenhouse*, (h) *Tray*, (i) Kipas DC, (j) Sensor DHT22, dan (k) *Nozzle Misting*.

Sistem kontrol akan diterapkan pada miniatur *greenhouse* sebagai batasan ruang kerja untuk sistem dengan lingkungan luar. Pada unit kontrol

menggunakan Arduino UNO ATmega328P sebagai mikrokontroler yang akan mengatur kerja dari perangkat keras yaitu unit sensor, unit kendali

(aktuator), dan unit pemantauan. Agar mikrokontroler dapat mengatur kerja dari perangkat keras diperlukan perangkat lunak (pemrograman). Unit kontrol akan diprogram menggunakan perangkat lunak Arduino IDE, sehingga dapat bekerja menjalankan pengontrolan kelembaban udara.

Rancangan Fungsional dan Struktural

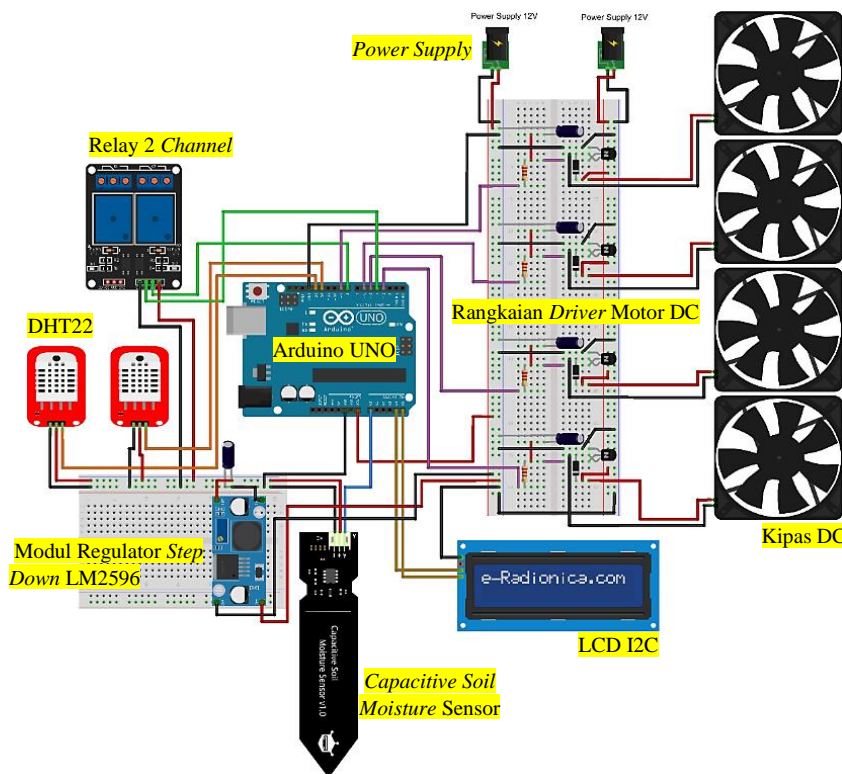
Rancangan Perangkat keras pada penelitian ini menggunakan Arduino UNO ATmega328P sebagai mikrokontroler. Mikrokontroler terhubung dengan 2 buah sensor DHT22 sebagai pengukur kelembaban udara, *capacitive soil moisture* sensor sebagai pengukur kelembaban media tanam yaitu *cocopeat*. Relay 2 channel sebagai pengendali *on/off* aktuator yaitu pompa AC untuk irigasi dan pompa DC untuk *misting*, 4 buah *driver* motor DC yang dikendalikan oleh *pin* PWM Arduino UNO sebagai pengendali kecepatan atau revolusi per menit (rpm) pada 4 buah kipas DC (2 *blower* dan 2 *exhaust*), serta LCD 16x2 bertugas menampilkan data kelembaban udara, kelembaban media tanam, dan nilai *duty cycle* dari kipas DC. Sebagai batasan ruang kerja sistem, sistem kontrol tersebut diterapkan pada miniatur *greenhouse* yang bertipe

lean-to berdimensi panjang 60 cm, lebar 40 cm, tinggi (1) 35 cm, tinggi (2) 63 cm, dan kemiringan atap 35°, yang dibangun menggunakan besi *hollow* 2 cm x 2 cm dan ditutup oleh plastik UV. Dengan dimensi tersebut, miniatur *greenhouse* memiliki kebutuhan aliran udara minimal sebesar 4,15 CFM (*Cubic Feet per Minute*) yang didapatkan dengan persamaan sebagai berikut (Buffington et al., 2019),

$$CFM = \frac{\text{Volume Ruang} \times ACH}{60} \tag{1}$$

- CFM : ft³/menit
- Volume Ruang : Luas Alas x Tinggi (ft³)
- ACH : *Air Change Per Hour*

Bangunan *greenhouse* biasanya membutuhkan pergantian udara (ACH) 1 kali setiap menit atau 60 kali dalam satu jam (Buffington et al., 2019). *Air Change Per Hour* (ACH) merupakan sebuah pengukuran berapa kali volume udara pada suatu ruangan akan dipindahkan, ditambahkan, dihilangkan atau ditukar dengan udara bersih yang baru (Frey, 2020). Nilai CFM minimum tersebut digunakan sebagai dasar penentu spesifikasi penggunaan kipas pada *greenhouse*. Adapun rangkaian perangkat keras sistem dapat dilihat pada Gambar 3.



Gambar 3. Rangkaian Breadboard Perangkat Keras

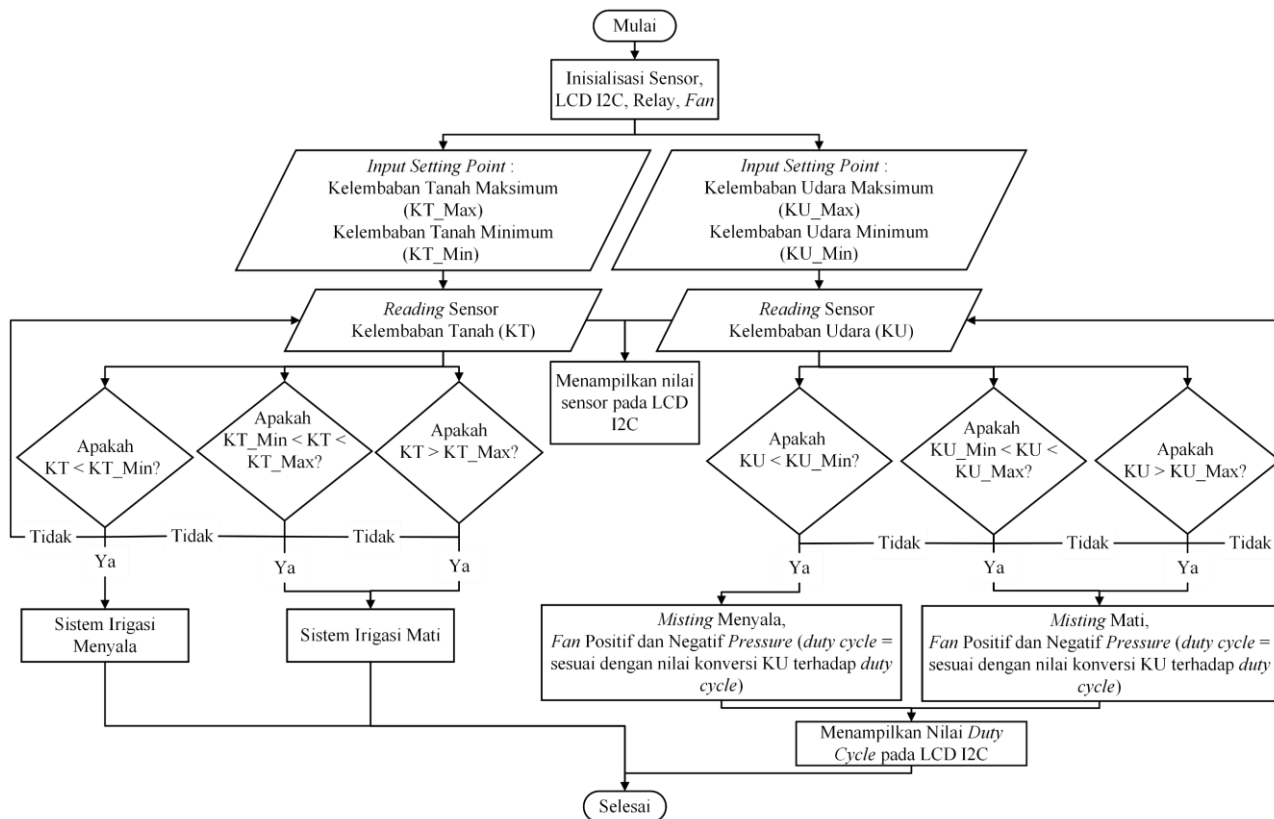
Rancangan Perangkat Lunak Sistem

Proses perancangan sistem pembacaan data sensor, menampilkan data pada LCD, dan mengeksekusi data menjadi *output* sistem diprogram menggunakan

Arduino IDE yang di-*compile* pada mikrokontroler Arduino UNO. Dalam mengeksekusi data menjadi *output* sistem yaitu menyalakan dan mematikan aktuator (pompa *misting* dan irigasi) dibutuhkan nilai

setting point. Rentang optimal nilai kelembaban udara *microgreen* 40% (KU_Min) sampai 80% (KU_Max) akan menjadi *setting point* untuk menyalakan atau mematikan pompa *misting* dan menjadi nilai yang mempengaruhi besaran nilai *duty cycle* pada kipas DC. Rentang nilai kelembaban udara akan dipetakan ulang menjadi nilai *duty cycle*

60 sampai 255 menggunakan fungsi *map()* pada Arduino IDE. Nilai kelembaban media tanam akan menjadi nilai *setting point* dalam menyalakan dan mematikan pompa irigasi secara otomatis. Nilai *setting point* tersebut dimasukkan secara manual melalui *coding*. Adapun diagram alir program atau perangkat lunak dapat dilihat pada Gambar 4.



Gambar 4. Diagram Alir Perangkat Lunak

Uji Fungsi

Dalam pengujian fungsi dilaksanakan pada ruang ber AC (*Air Conditioner*) yang bersuhu terjaga yaitu 25°C sampai 27°C.

1) Validasi Sensor

Validasi pada sensor kelembaban udara dengan membandingkan data kelembaban yang terukur oleh sensor DHT22 dengan higrometer. Perbandingan data akan dilakukan untuk mengetahui nilai *error* sensor DHT22. Sebagai indikator keberhasilan alat, maka persentase *error* sensor DHT22 pada alat yang dirancang diharapkan memiliki nilai *error* ±2-5% RH sesuai dengan *datasheet*. Untuk mengukur nilai *error* sensor menggunakan rumus sebagai berikut:

$$Error = \left| \frac{\text{Nilai Sensor} - \text{Nilai Alat Ukur}}{\text{Nilai Alat Ukur}} \right| \times 100\% \quad [2]$$

2) Pengujian Fungsi PWM, Driver Motor DC, dan Kipas DC

Uji fungsi PWM dilakukan menggunakan osiloskop untuk mengetahui teknik modulasi pulsa yang dihasilkan oleh mikrokontroler bekerja dengan baik.

Uji fungsi pada *driver* motor DC dan kipas DC dilakukan dengan mencari nilai tegangan, arus, daya dan rpm (revolusi per menit). Pengujian tersebut dilakukan menggunakan multimeter dan tachometer pada beberapa titik nilai *duty cycle* (D) yaitu 64 (25%), 128 (50%), 191 (75%), 255 (100%). Tujuan dari pengujian untuk mengetahui apakah nilai tegangan, arus, dan daya yang diukur sesuai dengan nilai perhitungan menggunakan persamaan. Dalam mencari nilai tegangan dan arus, *input* (V_{in} I_{in}) dan *output* (V_{out} I_{out}) menggunakan multimeter diukur pada rangkaian sebelum dan setelah *driver* motor DC (transistor). Nilai hasil pengukuran tersebut nantinya dibandingkan dengan nilai *output* tegangan dan arus, dan mencari nilai daya yang didapatkan dari hasil perhitungan menggunakan persamaan. Berikut konsep PWM dalam mengendalikan keluaran tegangan listrik dapat dirumuskan sebagai berikut (Ginting et al., 2014):

$$V_{out} = D \times V_{in} \quad [3]$$

$$I_{out} = D \times I_{in} \quad [4]$$

$$P = V_{out} \times I \quad [5]$$

Uji Kinerja Alat

Dalam pengujian kinerja dilaksanakan pada ruang ber AC (*Air Conditioner*) yang bersuhu terjaga yaitu 25°C sampai 27°C.

1) Uji Sistem Kendali

Pengujian *misting*, irigasi, dan pemberian aliran udara (kipas dengan PWM) atau keseluruhan sistem dilakukan dengan pengkondisian kelembaban dalam *greenhouse* untuk menciptakan kondisi yang memenuhi *decision* pada diagram alir perangkat lunak (Gambar 4). Selain itu, pengujian ini dilakukan untuk mendapatkan grafik hubungan fungsi *Map()* antara *duty cycle* dengan kenaikan kelembaban dari 40% - 80%. Kemudian didapatkan juga grafik penurunan kelembaban ketika *misting* sudah mati yaitu kelembaban dari 80% dan diukur waktu penurunannya.

2) Response Time Testing

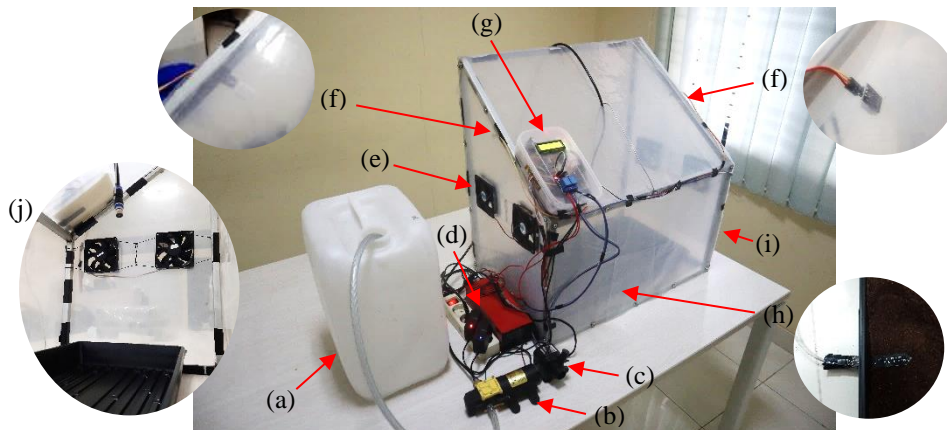
Response Time merupakan waktu yang dibutuhkan pada suatu titik dalam sistem dalam merespon atau menanggapi permintaan dari titik lain dalam sistem tersebut. Pengujian ini bertujuan untuk mengetahui berapa lama waktu yang diperlukan sistem dalam menanggapi kondisi eksternal yang diberikan terhadap sistem. Pengukuran *response time* dilakukan ketika unit sistem (sensor) mengirimkan data untuk diproses ke Arduino UNO sampai proses

tersebut memberikan tanggapan berupa menyalakan atau mematikan pompa irigasi dan *misting*.

HASIL DAN PEMBAHASAN

Prototipe Sistem Kontrol Kelembaban Udara *Greenhouse*

Hasil dari perancangan perangkat keras adalah sistem kontrol kelembaban udara *greenhouse* yang dapat dilihat pada Gambar 5 yang terdiri dari unit kontrol, unit sensor, aktuator, unit pemantauan, dan miniatur *greenhouse* serta komponen penunjang lainnya. Miniatur *greenhouse* yang dibuat bertipe *lean-to* berdimensi panjang 60 cm, lebar 40 cm, tinggi (1) 35 cm, tinggi (2) 63 cm, dan kemiringan atap 35°. Miniatur *greenhouse* dibangun menggunakan besi *hollow* berukuran 2 cm x 2 cm dan ditutup menggunakan plastik UV dengan ketebalan 0,06 mm pada bagian atap dan dinding prototipe tersebut. Miniatur *greenhouse* digunakan untuk meletakkan perangkat keras seperti unit kontrol dan perangkat keras lainnya seperti sensor 2 buah DHT22, *capacitive soil moisture* sensor, 4 buah kipas DC (2 *blower* dan 2 *exhaust*) berspesifikasi 12V 0,3A dengan RPM maksimum 2300 dan nilai CFM 53, *tray*, dan *nozzle misting*, serta memberikan batasan ruang kerja untuk sistem dengan lingkungan luar prototipe.



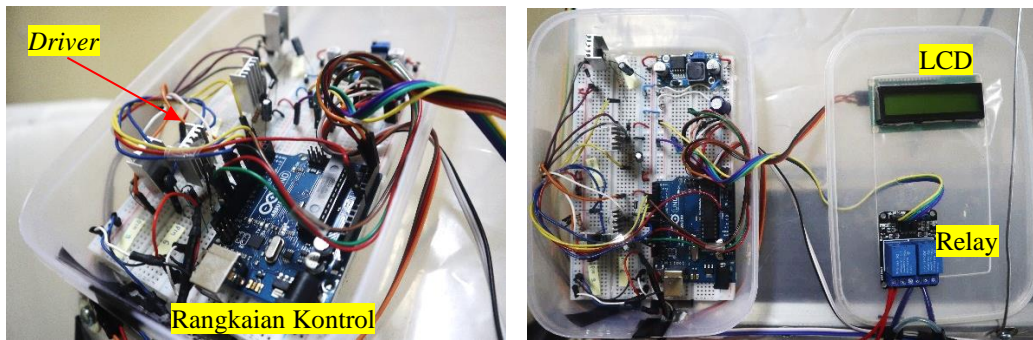
Gambar 5. Prototipe Sistem Kontrol Kelembaban Udara *Greenhouse*

Keterangan: (a) Sumber Air, (b) Pompa *Misting*, (c) Pompa Irigasi, (d) Sumber Listrik, (e) Kipas DC, (f) Sensor DHT22, (g) Unit Kontrol, (h) *Capacitive Soil Moisture* Sensor, (i) Miniatur *Greenhouse*, dan (j) Bagian Dalam.

Hasil Rangkaian Perangkat Keras

Unit kontrol terdiri dari papan Arduino UNO ATmega328P sebagai mikrokontroler yang berfungsi membaca data yang diterima oleh 2 buah sensor kelembaban udara DHT22 melalui *pin* digital D13 dan D12, membaca data *capacitive soil moisture* sensor melalui *pin* analog A0, mengontrol nyala LCD 16x2 yang terhubung dengan *pin* analog A4 dan A5,

mengontrol *on/off* pompa *misting* dan irigasi melalui relay pada *pin* digital D4 dan D8, mengontrol 4 buah *driver* motor DC menggunakan PWM pada *pin* digital PWM D3, D5, D6, dan D9. Arduino terhubung ke seluruh *ground* komponen melalui *pin* GND serta ditenagai oleh adaptor bertegangan 12V melalui *pin* VIN. Adapun perangkat keras unit kontrol dapat dilihat pada Gambar 6.



Gambar 6. Perangkat Keras Unit Kontrol

Hasil Rancangan Perangkat Lunak Sistem

Perangkat lunak sistem diprogram dengan Arduino IDE yang berfungsi untuk memprogram sistem atau Arduino untuk membaca data sensor, mengendalikan relay, dan mengendalikan nilai *duty cycle* kipas DC. Adapun kode pemrogramannya dinyatakan sebagai berikut.

```
#define DHTTYPE DHT22
uint8_t pinDHT1 = 12;
DHT dht1(pinDHT1, DHTTYPE);
uint8_t pinDHT2 = 13;
DHT dht2(pinDHT2, DHTTYPE);
// Soil Sensor
int soilsensor = A0;
//Reading Sensor DHT22
float kelembaban_udara1;
float kelembaban_udara2;
float kelembabanrata;
kelembaban_udara1 = dht1.readHumidity();
kelembaban_udara2 = dht2.readHumidity();
kelembabanrata = ((kelembaban_udara1 +
kelembaban_udara2)/2);
//Reading Sensor tanah
int nilai_sensor;
nilai_sensor = analogRead(soilsensor);

//Setting point kelembaban udara
const float kelembaban_udara_max = 80.00;
const float kelembaban_udara_min = 40.00;

//Set Point kelembaban media tanam
int Nilai_Dryplus = 604;
int Nilai_Dry = 455;
int Nilai_Wetplus = 249;
int Nilai_Wet = 285;

//Irigasi
// kondisi 1 Irigasi Nyala
if(nilai_sensor >= Nilai_Dry){
    digitalWrite(irigasi, LOW);
}
// kondisi 2 Irigasi Mati
if(nilai_sensor <= Nilai_Wet){
    digitalWrite(irigasi, HIGH);
}

//Map for DHT22 Value to PWM Value
float PWMDHT22 = ((kelembaban_udara1 +
kelembaban_udara2)/2);
if(PWMDHT22 > 80.00){
    PWMDHT22 = 80.00;
}
if(PWMDHT22 < 40.00){
    PWMDHT22 = 40.00;
}
int PWMpositive = map(PWMDHT22, 40, 80, 60,
255);
int PWMnegative = map(PWMDHT22, 40, 80, 60,
255);
```

```
// Misting
if(kelembabanrata<kelembaban_udara_min){
    digitalWrite(misting, LOW);
    analogWrite(DcMotor1, PWMpositive);
    analogWrite(DcMotor2, PWMpositive);
    analogWrite(DcMotor3, PWMnegative);
    analogWrite(DcMotor4, PWMnegative);
}

elseif((kelembaban_udara_min<kelembabanrata)&
&(kelembabanrata < kelembaban_udara_max)){
    analogWrite(DcMotor1, PWMpositive);
    analogWrite(DcMotor2, PWMpositive);
    analogWrite(DcMotor3, PWMnegative);
    analogWrite(DcMotor4, PWMnegative);
}

elseif(kelembabanrata>kelembaban_udara_max){
    digitalWrite(misting, HIGH);
    analogWrite(DcMotor1, PWMpositive);
    analogWrite(DcMotor2, PWMpositive);
    analogWrite(DcMotor3, PWMnegative);
    analogWrite(DcMotor4, PWMnegative);
}

// LCD I2C
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
// LCD
lcd.setCursor(0,0);lcd.print("HT:");
lcd.setCursor(7,0);lcd.print("HU:");
lcd.setCursor(10,0);lcd.print(kelembabanrat
a);
lcd.setCursor(15,0);lcd.print("%");
lcd.setCursor(0,1);lcd.print("-PWM:");
lcd.setCursor(8,1);lcd.print("+PWM:");
lcd.setCursor(13,1);lcd.print(PWMpositive);
lcd.setCursor(5,1);lcd.print(PWMnegative);
```

Pengujian fungsi Sensor DHT22

Dalam mencari nilai *error* kelembaban udara digunakan alat ukur *Digital Thermo-Hygrometer* model 303C sebagai nilai pembanding dengan DHT22. Data sampel kelembaban udara diambil setiap 2 detik dalam 10 menit, sehingga dalam 1 menit ada 30 data. Setiap 30 data dalam 1 menit nilai data dibuat rata – rata, sehingga hasil *error* menjadi 10 data dalam 10 menit. Adapun hasil pengukuran nilai *error* dapat dilihat pada Tabel 1. Dari hasil data pengukuran kelembaban udara menggunakan 2 (dua) buah DHT22 dan alat ukur, didapatkan besar rata - rata *error* pengukuran kelembaban udara sensor DHT22(1) adalah 3,3% dan DHT22(2) adalah 4,06%, dengan kisaran kerja dari pada sensor kelembaban udara yaitu 0% RH - 100% RH, dengan kepekaan atau sensitivitas 0,1% RH.

Tabel 1. Hasil Rata - rata Pengujian *Error* Sensor Kelembaban Udara DHT22

No	Menit Ke-	DHT22 (1) (%)	DHT22 (2) (%)	Alat Ukur (%)	Selisih (1) (%)	Selisih (2) (%)	Error (1) (%)	Error (2) (%)
1	1	60,96	60,96	59,00	1,96	1,96	3,33	3,32
2	2	60,94	61,09	59,00	1,94	2,09	3,28	3,54
3	3	60,90	61,03	59,00	1,90	2,03	3,23	3,44
4	4	60,75	61,04	59,00	1,75	2,04	2,97	3,45
5	5	60,76	60,95	59,00	1,76	1,95	2,99	3,31
6	6	60,76	60,86	59,00	1,76	1,86	2,98	3,16
7	7	60,72	60,80	59,00	1,72	1,80	2,91	3,05
8	8	60,72	60,86	59,00	1,72	1,86	2,91	3,15
9	9	60,87	61,04	59,00	1,87	2,04	3,17	3,46
10	10	60,85	61,05	59,00	1,85	2,06	3,14	3,49
Rata - rata					1,82	1,97	3,09	3,34

Pengujian Fungsi Sistem Kendali

Pengujian ini berlangsung pada *pin* digital D9, D6, D5, dan D3 PWM Arduino UNO dan keempat *driver* motor DC yang diberikan beban berupa kipas DC 12V 0,3A. Setiap 2 kipas diberikan daya 12V 1A yang dialirkan melalui transistor NPN BD139 sebagai saklar. Pengukuran menggunakan osiloskop dilakukan dengan menghubungkan *hook probe* pada *pin* digital PWM yang terhubung dengan *driver* atau *pin base* transistor NPN BD139, kemudian kabel *ground* alat pada *ground* mikrokontroler. Pengujian dilakukan pada setiap *duty cycle* 25%, 50%, 75%, dan 100%.

a) Karakteristik *Pulse Wave* pada *Pin* Digital PWM Arduino UNO.

Hasil karakteristik gelombang pulsa yang dihasilkan oleh mikrokontroler, dianalisis berdasarkan bentuk dan spesifikasi gelombang. Bentuk dan spesifikasi gelombang pulsa diukur menggunakan *Digital*

Storage Oscilloscope MDS-9060 60MHz. Gelombang pulsa yang dihasilkan memiliki perbedaan bentuk pada setiap *duty cycle*.

**Gambar 7.** Bentuk Gelombang PWM**Tabel 2.** Hasil Pengujian pada *Output* dan *Input Driver*

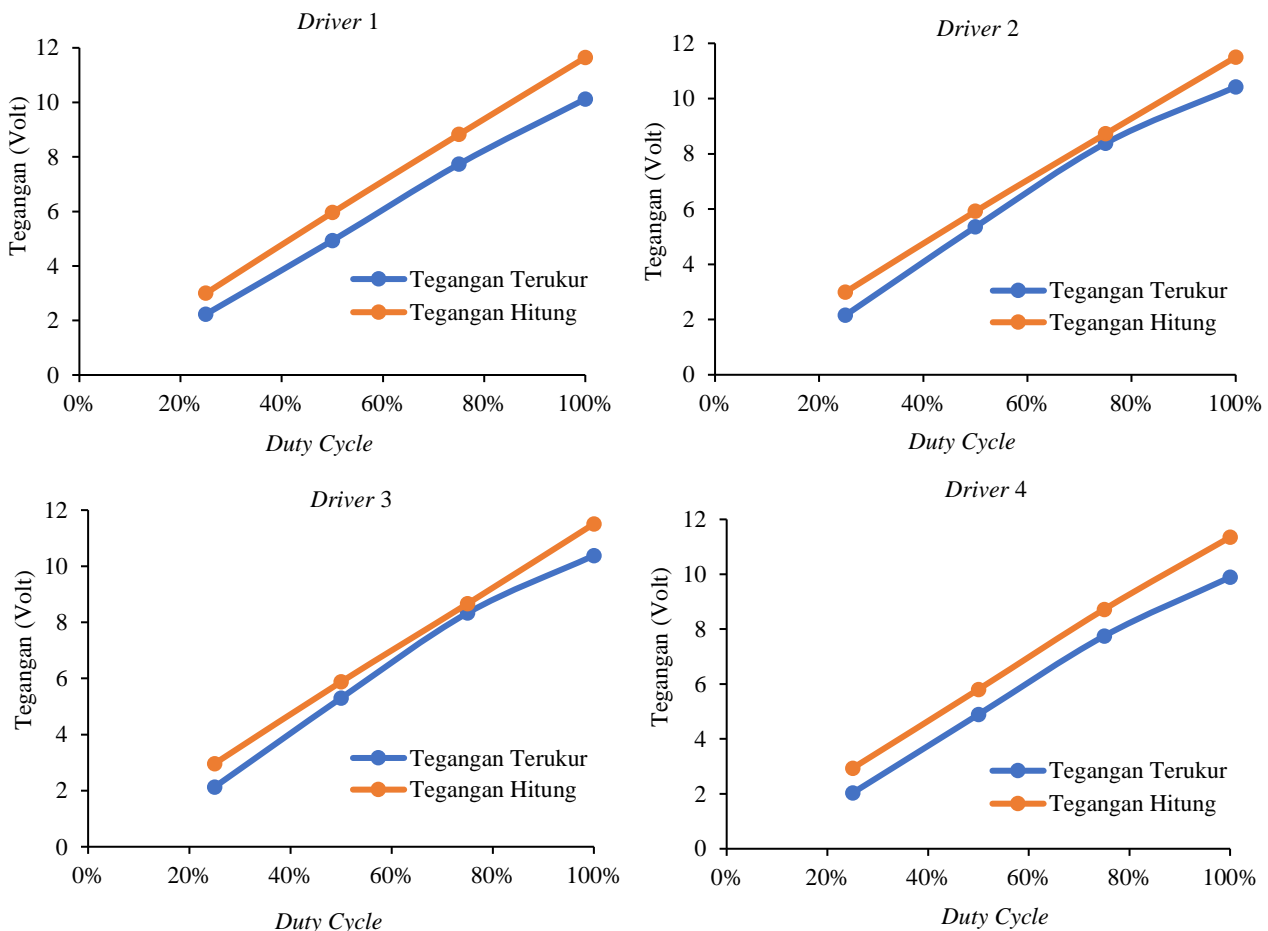
<i>Driver</i>	D	<i>V in</i>	<i>A in</i>	Hasil Pengukuran			Hasil Perhitungan		
				<i>V out</i>	<i>A out</i>	Daya	<i>V out</i>	<i>A in</i>	Daya
(Pin D9)	25%	11,98	0,26	2,23	0,03	0,07	3,00	0,07	0,19
	50%	11,91	0,26	4,93	0,10	0,49	5,96	0,13	0,77
	75%	11,76	0,26	7,73	0,17	1,31	8,82	0,20	1,72
	100%	11,64	0,26	10,12	0,22	2,23	11,64	0,26	3,03
(Pin D6)	25%	11,95	0,26	2,15	0,03	0,06	2,99	0,07	0,19
	50%	11,83	0,26	5,35	0,11	0,59	5,92	0,13	0,77
	75%	11,63	0,26	8,38	0,18	1,51	8,72	0,20	1,70
	100%	11,50	0,26	10,42	0,23	2,40	11,50	0,26	2,99
(Pin D5)	25%	11,85	0,26	2,13	0,03	0,06	2,96	0,07	0,19
	50%	11,75	0,26	5,30	0,11	0,58	5,88	0,13	0,76
	75%	11,55	0,26	8,33	0,19	1,58	8,66	0,20	1,69
	100%	11,51	0,26	10,38	0,24	2,49	11,51	0,26	2,99
(Pin D3)	25%	11,73	0,26	2,03	0,03	0,06	2,93	0,07	0,19
	50%	11,61	0,26	4,89	0,10	0,49	5,81	0,13	0,75
	75%	11,63	0,26	7,75	0,16	1,24	8,72	0,20	1,70
	100%	11,35	0,26	9,89	0,21	2,08	11,35	0,26	2,95

Semakin besar nilai *duty cycle*, maka bentuk dari pulsa *high* atau pulsa positif akan semakin lebar, sementara pulsa *low* atau pulsa negatif akan semakin menyempit yang dapat dilihat pada Gambar 7. Pada *duty cycle* 100% hanya akan membentuk sinyal lurus dengan nilai tegangan yang searah 5V sesuai dengan nilai tegangan pada *pin* digital Arduino. Dari hasil pengukuran menggunakan osiloskop, menunjukkan bahwa semakin besar nilai *duty cycle*, maka tegangan efektif (V_{rms}) yang dihasilkan oleh Arduino UNO pada *pin* digital PWM akan semakin besar. Pada *duty cycle* 100% tidak terdapat periode dan frekuensi, karena lebar pulsa *high* bernilai 100% dan pulsa *low* bernilai 0%, sehingga tidak terbentuk gelombang hanya sebuah sinyal listrik DC yang konstan.

b) Pengukuran dan Perhitungan Tegangan (volt), Arus (ampere), dan Daya (watt)
 Pengukuran tegangan dan arus dilakukan pada keempat *driver* pada komponen dioda yang

merupakan *output driver*. Hasil dari pengukuran dan perhitungan dapat dilihat pada

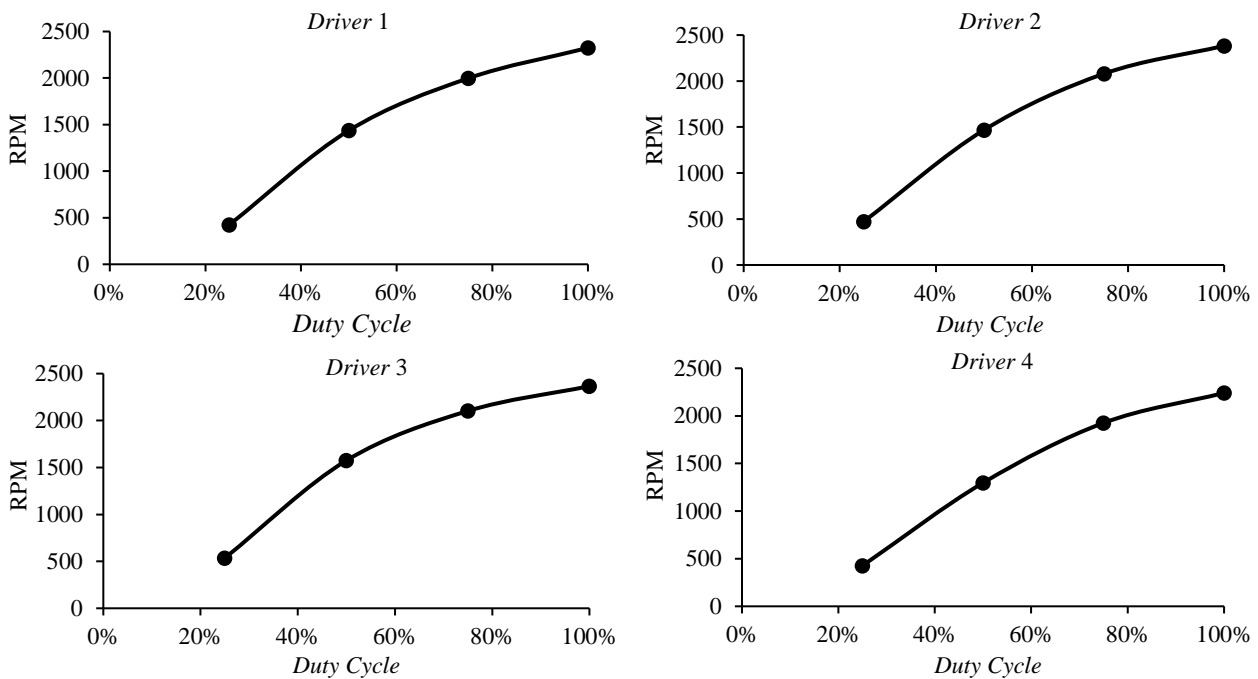
Tabel 2. Keempat pengujian pada *driver* yang berbeda, terdapat perbedaan antara hasil pengukuran dengan perhitungan pada tegangan (*drop voltage*) dan arus pada setiap *duty cycle*. Perbedaan pengukuran dengan perhitungan kemungkinan disebabkan oleh kualitas dari komponen transistor NPN yang digunakan sebagai *driver*. Selain itu, perbedaan pengukuran kemungkinan diakibatkan oleh penggunaan dioda yang dapat menurunkan tegangan sampai 0,7V pada rangkaian *driver* motor menggunakan transistor (Juditova, 2021). Hasil pengukuran dan perhitungan dapat juga dilihat pada Gambar 8. Seiring meningkatnya nilai *duty cycle* 25% sampai 100%, maka nilai tegangan yang dihasilkan akan meningkat. Pernyataan tersebut berlaku juga dengan arus dan daya.



Gambar 8. Grafik Hubungan *Duty Cycle* dengan Tegangan

c) RPM (Revolusi Per Menit)
 Hasil dari pengukuran RPM kipas DC menggunakan tachometer menunjukkan grafik hubungan antara *duty cycle* dengan kecepatan putaran kipas DC (RPM) dapat dilihat pada Gambar 9. Pada Grafik

menunjukkan, seiring meningkatnya *duty cycle* maka kecepatan atau RPM dari kipas DC semakin meningkat, dengan rata – rata RPM maksimum yaitu 2329. Kipas DC tidak dapat berputar ketika *duty cycle* berada di bawah 23,5%.



Gambar 9. Grafik Hubungan *Duty Cycle* dengan RPM

Pengujian ini dapat disimpulkan bahwa fungsi dari rangkaian *driver* motor DC dalam mengontrol kecepatan kipas DC menggunakan kontrol PWM sudah bekerja baik sesuai dengan perancangan

perangkat keras. Berikut Tabel 3 merupakan hasil gabungan dari pengujian PWM, tegangan Arduino ke *driver*, tegangan *driver* ke kipas, dan RPM yang juga menjadi dasar kesimpulan dari pengujian ini.

Tabel 3. Hasil Pengujian Fungsi PWM, *Driver*, dan Kipas DC

<i>Driver</i>	<i>Duty Cycle</i>	Frekuensi	Tegangan Arduino ke Base (Vrms)	Tegangan <i>Emitter</i> ke Kipas DC	RPM
1 (<i>Pin</i> D9)	25%	490,2 Hz	2,48V	2,23V	425
	50%	490,8 Hz	3,52V	4,93V	1437
	75%	490,2 Hz	4,24V	7,73V	1998
	100%	-	4,96V	10,12V	2324
2 (<i>Pin</i> D6)	25%	978,0 Hz	2,48V	2,15V	472,7
	50%	975,6 Hz	3,44V	5,35V	1467
	75%	976,6 Hz	4,40V	8,38V	2080
	100%	-	4,96V	10,42V	2383
3 (<i>Pin</i> D5)	25%	976,6 Hz	2,24V	2,13V	536
	50%	976,6 Hz	3,52V	5,3V	1574
	75%	977,5 Hz	4,40V	8,33V	2102
	100%	-	4,96V	10,38V	2366
4 (<i>Pin</i> D3)	25%	490,2 Hz	2,48V	2,03V	427,1
	50%	490,2 Hz	3,52V	4,89V	1297
	75%	490,2 Hz	4,32V	7,75V	1926
	100%	-	4,96V	9,89V	2241

Adanya perbedaan frekuensi pada *pin* Arduino UNO ATmega328P menyebabkan perbedaan RPM kipas DC pada *duty cycle* yang sama. Rata - rata frekuensi yang dihasilkan oleh *pin* PWM D9 dan D3 adalah 490,3 Hz dan pada *pin* PWM D6 dan D5 adalah 976,82 Hz. Semakin besar nilai frekuensi dan *duty cycle*, maka nilai RPM kipas DC yang dihasilkan akan semakin besar (Sonjaya Suhendar & Koswara, 2020).

d) Hubungan Kelembaban Udara dan Sistem Kendali Kipas DC

Hasil dari implementasi fungsi *map()* pada Arduino IDE pada rentang kelembaban udara 40% sampai 80% dengan rentang *duty cycle* 60 sampai 255 (23,5% - 100%) ditunjukkan dengan sebuah persamaan linier $y = 4,878x - 135,61$ dengan $R^2 = 1$ yang didapatkan dari sebuah grafik. Adapun hasil hubungan antara kelembaban udara dengan *duty cycle* dan RPM pada kipas DC dapat dilihat pada Tabel 4.

Tabel 4. Kelembaban Udara, *Duty Cycle*, dan Kecepatan Kipas DC

No.	Kelembaban Udara	<i>Duty Cycle</i>	Kecepatan Kipas	RPM
1	<40%	<23,5% (<60)	Tidak Bergerak	0
2	40%	23,5% (60)	Sangat Lambat	<465,2
3	41%	25% (64)	Lambat	465,2
4	54%	50% (128)	Sedang	1444
5	67%	75% (191)	Cepat	2027
6	80%	100% (255)	Sangat Cepat	2329
7	>80%	100% (255)	Sangat Cepat	2329

Uji Kinerja Prototipe

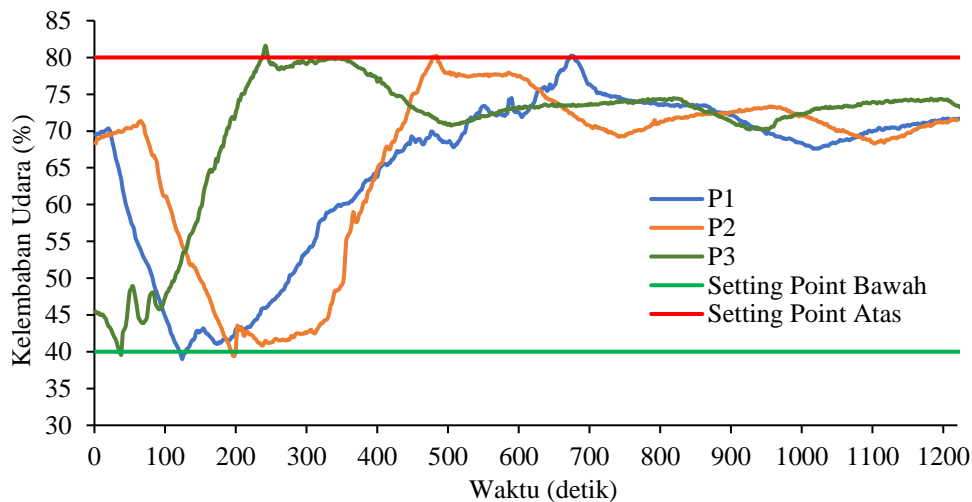
Prototipe sistem kelembaban udara *greenhouse* diuji secara keseluruhan untuk mengetahui apakah pompa *misting* dan irigasi dapat menyala dan mati ketika menyentuh nilai *setting point*. Parameter yang menjadi kinerja yaitu *response time*. Pengujian kinerja dilakukan untuk mendapatkan nilai *response time* sistem yang terdiri dari *response time* kerja *on/off* dan *response time* kerja sensor. Setiap pengujian dilakukan 3 kali pengulangan dengan pengambilan data setiap 2 detik melalui *serial* komunikasi Arduino IDE.

a) Pengujian Prototipe Sistem Kelembaban Udara *Greenhouse*

Dapat dilihat pada Gambar 10 merupakan grafik hasil pengujian prototipe ketika dijalankan. Pada grafik

menunjukkan sistem kerja *on/off* pada sistem *misting* telah bekerja sesuai dengan perancangan. Ketika kelembaban udara dalam miniatur *greenhouse* menyentuh nilai *setting point* bawah, pompa *misting* menyala sehingga kelembaban udara naik sampai 80% hingga pompa *misting* tersebut mati.

Dari grafik pengujian tersebut ditunjukkan juga bahwa kipas DC dapat menurunkan kelembaban udara sampai titik terendah pertama pada P1 (Pengulangan 1) 80,25% sampai 67,60% dengan selisih 12,65% selama 344 detik, P2 (Pengulangan 2) 80,10% sampai 69,40% dengan selisih 10,70% selama 270 detik, dan P3 (Pengulangan 3) 80,15% sampai 70,75% dengan selisih 9,4% selama 268 detik.



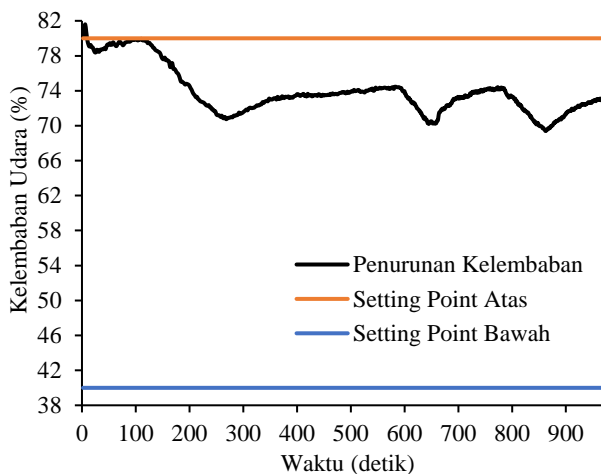
Gambar 10. Grafik Pengujian Prototipe Sistem Kontrol Kelembaban Udara *Greenhouse*

Setelah mengalami penurunan kelembaban udara sampai titik terendah pertama, RPM kipas DC semakin rendah, sehingga kemampuan untuk menurunkan kelembaban udara menurun. Akibatnya kelembaban udara kembali naik yang disebabkan oleh penguapan pada media tanam *cocopeat* dengan kondisi kapasitas lapang. Setelah kelembaban naik, RPM kipas akan berputar lebih cepat sehingga kelembaban kembali turun. Pola penurunan dan kenaikan tersebut akan terus terjadi, namun pada rentang nilai kelembaban optimal bagi pertumbuhan *microgreen* yaitu 40% sampai 80% yang dapat dilihat pada Gambar 11. Dapat disimpulkan sistem kendali

PWM dalam mengontrol kecepatan putaran kipas DC sebagai pemberi aliran udara dan menurunkan kelembaban udara pada *greenhouse*, mampu menurunkan kelembaban udara dari titik maksimum 80%, serta menjaga kelembaban udara tetap pada rentang yang optimal bagi pertumbuhan *microgreen*, dengan menurunkan dan menaikkan nilai RPM dari kipas DC sesuai dengan nilai *duty cycle* nya.

Dapat dilihat pada Gambar 11 terjadi *overshoot* pada detik ke 6 dengan nilai *error* 1,6% diatas *setting point* atas kelembaban udara. Sistem kontrol PWM kipas DC mampu mengembalikan nilai *error* (*overshoot*) pada rentang yang optimal pada detik ke 10 selama

4,174 detik (*response time*). Dimana sistem memiliki *response time* yang baik dalam menanggapi *error* atau kelebihan kelembaban udara setelah pemberian *misting*. Batas maksimal *response time testing* yang dapat diterima yaitu 10 detik, semakin cepat waktu respon maka semakin baik dengan rentang 0,1 detik sampai 1 detik (Ferry, 2019).



Gambar 11. Pola Penurunan Kelembaban Udara

b) Kinerja *Response Time*

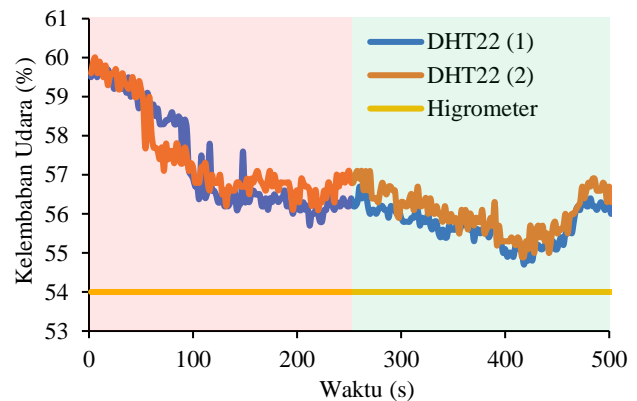
Pompa *misting* dan irigasi otomatis telah bekerja dengan baik, dengan kinerja berupa *response time* atau waktu yang dibutuhkan sistem saat membaca data sensor kelembaban sampai memberikan *output* kendali berupa menyala (5V) dan mematikan (0V) pompa. Hasil pengujian *response time* dapat dilihat pada Tabel 5. Hasil *response time* rata - rata yang dibutuhkan sistem irigasi sebesar 78mS dan sistem *misting* 145mS dengan diberikan rentang waktu (*delay*) pembacaan sensor setiap 2 detik. Debit air rata – rata yang dihasilkan oleh pompa irigasi dan pompa pompa *misting* sebesar 42,67mL/detik dan 222,67mL/detik.

Tabel 5. Hasil Pengujian Kinerja *Response Time*

Kontrol On/Off	Aktuator	Ulangan	Response Time (mS)	Delay (Detik)
Relay 1	Pompa Misting	P1	93	2
		P2	79	2
		P3	62	2
		Rata-rata:	78	2
Relay 2	Pompa Irigasi	P1	137	2
		P2	125	2
		P3	172	2
		Rata-rata:	145	2

Response time yang dibutuhkan oleh sistem dalam beradaptasi menyesuaikan atau menstabilkan pengukuran kelembaban udara untuk mencapai *steady state* dengan kelembaban udara yang terukur menggunakan *Digital Thermo-Hygrometer* model 303C, membutuhkan waktu sekitar 250 detik sampai 300 detik dengan nilai *error* sesuai dengan pengujian

fungsi. *Steady state* merupakan kondisi dimana nilai *output* dari sistem memasuki kondisi yang stabil sesuai dengan nilai *input*, dan kondisi sebelum *steady* disebut dengan *transient state* atau peralihan (Sari, 2013). Adapun grafik *response time* dapat dilihat pada Gambar 12, dimana daerah merah merupakan *transient state* dan daerah hijau merupakan *steady state*.



Gambar 12. Grafik *Response Time* Sensor

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, berhasil dirancang dan dibangun sebuah prototipe sistem kontrol kelembaban udara *greenhouse* menggunakan PWM. Prototipe terapkan pada miniatur *greenhouse* bertipe *lean-to* berdimensi panjang 60 cm, lebar 40 cm, tinggi (1) 35 cm, tinggi (2) 63 cm, dan kemiringan atap 35°, dengan aliran udara minimal sebesar 4,15 CFM. Prototipe dilengkapi sensor DHT22 dengan nilai *error* kelembaban udara DHT22(1) 3,09% dan DHT22(2) 3,34%, dengan kisaran kerja dari pada sensor kelembaban udara yaitu 0% RH - 100% RH, dengan kepekaan atau sensitivitas 0,1% RH. Kendali PWM kipas DC berhasil dibangun, semakin besar nilai tegangan yang diberikan oleh mikrokontroler Arduino UNO menuju *driver* motor DC, maka besar tegangan yang mengalir keluar dari *driver* (*collector – emitter*) menuju kipas DC semakin besar, sehingga mempengaruhi nilai RPM dari kipas DC. Sistem berhasil membaca dan menampilkan data, dan memberikan *output* dengan kinerja *response time on/off* sistem irigasi dan *misting* sebesar 78mS dan 145mS, dengan *delay* 2 detik.

DAFTAR PUSTAKA

- Alahudin, M. (2013). Kondisi Termal Bangunan Greenhouse dan Screen House pada Fakultas Pertanian Universitas Musamus Merauke. *Jurnal Ilmiah Mustek Anim Ha*, 2(1), 16–27.
- Anonim. (2017). Microgreens Production. *Johnny's*

- Selected Seeds*, 1–3. <http://www.johnnyseeds.com/growers-library/micro-greens-production.html>
- Apriast, P., Sudana, I., & Sudarma, I. (2015). Hubungan Sifat Fisika dan Kimia Tanah dengan Persentase Penyakit Layu pada Tanaman Cengkeh (*Syzygium Aromaticum* L.) yang Disebabkan oleh Jamur Akar Putih (*Rigidoporus* Sp.) di Desa Unggahan, Kabupaten Buleleng. *E-Jurnal Agroekoteknologi Tropika (Journal of Tropical Agroecotechnology)*, 4(1), 25–32.
- Arifin, S., & Fathoni, A. (2014). Pemanfaatan Pulse Width Modulation Untuk Mengontrol Motor (Studi Kasus Robot Otomatis Dua Deviana). *Jurnal Ilmiah Teknologi Informasi Asia*, 8(2), 65–80.
- As'adiya, L., & Murwani, I. (2021). Pengaruh Lama Penyinaran Lampu Led Merah, Biru, Kuning Terhadap Pertumbuhan Microgreen Kangkung (*Ipomoea Reptant*). *Folium: Jurnal Ilmu Pertanian*, 5(1), 14. <https://doi.org/10.33474/folium.v5i1.10358>
- Buffington, D E, Bucklin, R. A., Henley, R. W., & Mcconnell, D. B. (2019). Greenhouse Ventilation. *U.S. Department of Agriculture, UF/IFAS Extension Service, University of Florida, IFAS*, 1–4. <https://edis.ifas.ufl.edu/ae030>
- Ferry. (2019). *Response Time Testing*. <https://sis.binus.ac.id/2019/05/13/response-time-testing/>. Diakses 20 Juli 2022.
- Frey, N. (2020). *Air Changes Per Hour: The What, Why, & How Answered*. <https://www.vaniman.com/air-changes-per-hour-the-what-why-how-answered/>. Diakses 22 Februari 2022.
- Ginting, V. br, Perangin-angin, B., & Tamba, T. (2014). Sistem Pengendalian Asap Rokok Multikanal dengan Menggunakan Pwm Berbasis Mikrokontroler Atmega 8. *FISIKA FMIPA USU*, 6, 58–66.
- Grooms, D. (2020). Microgreens: Market Analysis, Growing Methods and Models. *MSc Thesis; California State University, Na*, 1–24.
- Hutabarat, T. M. R. (2018). *Analisis Pengaruh Penambahan Jumlah Gain pada Rancangan Pengendali Adaptif Metode Mrac Diuji pada Sistem Orde Dua*. Universitas Islam Negeri Sultan Syarif Kasim Riau.
- Indriani, A., Witanto, Y., Supriadi, & Hendra. (2017). Sistem Kontrol Kekeruhan dan Temperatur Air Laut Menggunakan Microcontroller Arduino Mega. *Jurnal Teknik Mesin Mercu Buana*, 6(3), 158–163.
- Juditova. (2021). *How to Drive a DC Motor with a BJT Transistor*. <https://www.techzorro.com/en/blog/how-to-drive-a-dc-motor-with-a-bjt-transistor/>. Diakses 7 Juli 2022.
- Karsid, K., Ramadhan, A. W., & Aziz, R. (2018). Perbandingan Kinerja Mesin Penetas Telur Otomatis dengan Menggunakan Kontrol On-Off dan Kontrol PWM. *Matrix: Jurnal Manajemen Teknologi Dan Informatika*, 8(1), 1–5. <https://doi.org/10.31940/matrix.v8i1.733>
- Megido, A., & Ariyanto, E. (2016). Sistem Kontrol Suhu Air Menggunakan Pengendali PID dan Volume Air pada Tangki Pemanas Air Berbasis Arduino Uno. *Gema Teknologi*, 18(4), 21. <https://doi.org/10.14710/gt.v18i4.21912>
- Nolan, D. A. (2018). *Effects of Seed Density and Other Factors on the Yield of Microgreens Grown Hydroponically on Burlap*.
- Nurnasari, E., & Djumali. (2016). Pengaruh Kondisi Ketinggian Tempat Terhadap Produksi dan Mutu Tembakau Temanggung. *Buletin Tanaman Tembakau, Serat & Minyak Industri*, 2(2), 45. <https://doi.org/10.21082/bultas.v2n2.2010.45-59>
- Renna, M., Di Gioia, F., Leoni, B., Mininni, C., & Santamaria, P. (2017). Culinary Assessment of Self-Produced Microgreens as Basic Ingredients in Sweet and Savory Dishes. *Journal of Culinary Science and Technology*, 15(2), 126–142. <https://doi.org/10.1080/15428052.2016.1225534>
- Ripai, A., & Wibowo, A. (2016). Obstacle Avoider Prototype Robot Using After Market Component. *Swabumi, IV(2)*, 129–140.
- Sari, F. Y. (2013). Analisa Steady State Error Sistem Kontrol Linier Invariant Waktu. *Jurnal Matematika UNAND*, 2(3), 91. <https://doi.org/10.25077/jmu.2.3.91-97.2013>
- Sonjaya Suhendar, W., & Koswara, E. (2020). Analisis Pengaruh Duty Cycle dan Frekuensi terhadap Kecepatan Motor Listrik. *Computer Vision: Principles, Algorithms, Applications, Learning: Fifth Edition*, 103, 1–858.
- Yudaningtyas, E. (2017). *Belajar Sistem Kontrol: Soal dan Pembahasan*. UB Express.