

Design and Build Automatic Fan Control Based Internet of Things

Refli Septian^{1*}, Achmad Fauzi², Milli Alfhi Syari³

^{1,2,3} STMIK Kaputama
refliseptian1792@gmail.com^{1*}, fauzyrivai88@gmail.com², milli.fhisya@gmail.com³

Abstract

This research aims to create and develop an automatic fan control system based on the Internet of Things (IoT) that can regulate the room temperature well. The NodeMCU ESP8266 serve as a microcontroller for this system. The DHT22 sensor detects temperature, the L298N Driver Motor controls the speed of the 12V DC fan, and the 16x2 LCD displays temperature information in realtime. When the room temperature exceeds the threshold, the fan will automatically turn on and off when the temperature returns to normal. The Blynk app also allows users to monitor and control fans through smartphones. It allows users to manually control the fan and monitor it remotely. The results of the study show that this system works well and improves user comfort by managing the room temperature. The system also offers high flexibility with IoT integration, allowing for further development for wider applications in home automation and energy management.

Keywords: *Internet of Things (IoT), Automatic Fan Control, NodeMCU ESP8266, DHT22 Sensor, Blynk Application, Temperature Setting*

1. Introduction

An ever-evolving digitalization system, the Internet of Things has become a driving force that is changing the way we control and automatically monitor our environment. Temperature regulation is one of the areas that has been severely affected by this development. By using smart sensors connected to the IoT network, we can design intelligent and adaptive solutions to solve temperature problems in a hot and stuffy place with IoT-based automatic fans. With this technology, temperature data can be collected and analyzed directly, which allows for effective fan setup.

This solution can improve air quality and efficiency in the use of electrical energy. This is because IoT-based automatic fan control systems can not only keep the temperature within a comfortable range, but can also offer intelligent solutions for overall room environment management that can be remotely controlled through Internet of Things technology.

2. Literature Review

2.1. Design

Design is the initial stage of the process of creating images and sketches that have never been made before and then managed into the desired image or sketch. Before the manufacturing stage, design is an important step in planning and designing thoroughly. In information technology, this includes designing software systems and regulating how system components interact with each other. In other words, the design ensures that all elements have been thoroughly planned before the implementation phase begins.

2.2. System Control

A control system is a collection of methods and techniques learned from human work habits that require quality observation of what they have done so that it has characteristics that are in line with what was expected at first.

2.3. 12Volt DC Fan

A fan is a tool used to create airflow by rotating blades or fans. It is usually used to cool a room or maintain air circulation so that the air becomes fresher and more comfortable. Fans use electric motors to convert electrical energy into motion energy. In this electric motor, the moving iron coil is in the stationary part, and a pair of U magnets are in the stationary part. When electricity flows on the wire winding, the iron coil becomes a magnet.

2.4. Internet of Things (IOT)

The Internet of Things (IOT) is a system in which objects and people are given exclusive identities and have the ability to move data over a network without the need for bidirectionality between humans and humans, i.e. source-to-destination or human-to-computer interaction.

2.5. NodeMCU Esp8266

NodeMCU ESP8266 is an open-source hardware development platform that aims to simplify the process of developing Internet of Things applications. NodeMCU has a WiFi module that allows devices connected to it to communicate over the internet network. NodeMCU is a processing-based firmware and is a development of ESP8266. NodeMCU uses the same logic and programming structure of the processing language as the C language, only the syntax is different.

2.6. Blynk

Blynk is an Internet of Things Cloud platform application for iOS and Android that allows the control of the Raspberry Pi module, NodeMCu ESP8266, and other modules via the internet. Since this app has many features designed to make it easy for its users, it is very easy to use for those who are just starting to use it. The app gives users the ability to control anything from wherever they are as long as they are connected to the internet.

2.7. Arduino IDE

Arduino IDE is a software specifically designed to help users create electronic projects by using Arduino microcontroller boards. The software uses the Wiring programming language, which is a derivative of the C/C++ language, to write, edit, and organize program code. The Arduino IDE has many useful features, such as a code editor, board settings to choose the type of motherboard, and settings to choose the desired type of motherboard. The Arduino IDE is open-source software that allows developers and hobbyists to create and develop a variety of electronic projects, from simple to more complex, while exploring the world of electronics and the Internet of Things (IoT).

2.8. DHT22

The DHT22 is an inexpensive digital sensor that measures temperature and humidity by using a capacitor-based humidity sensor and temperature sensor components to measure the temperature of the surrounding air. These sensors are highly accurate in a wide range of humidity and temperature levels, and their output is a digital signal that must be sent to the appropriate data pins.

2.9. LCD

LCD is a display medium that can display a lot of good characters. The 16x2 LCD screen has the ability to display 32 characters, with 16 characters in the top row and 16 characters in the bottom row. I2C LCD materials are transferred in parallel, both information and control lines, with I2C/IIC (Inter Integrated Circuit) or TWI (Two Wire Interface) protocols.

2.10. Motor Driver L298N

A motor driver is a motor control module that is commonly used in various electronic projects involving DC motor control. The L298N driver motor is based on the L298 dual H-bridge IC and functions to regulate the direction and speed of the DC motor. This is necessary because DC motors typically require more than 250 mA of current, which some cannot provide IC, like the ATmega family.

2.11. LM2596s Regulator

LM2596S is a DC-DC step-down voltage regulator module or buck converter that is used to convert a higher DC input voltage into a lower, stable DC output voltage. This module is based on the LM2596 regulator chip, which is a switching regulator, so it is highly efficient compared to conventional linear regulators. Due to the switching technology, these modules are highly efficient and generate little heat.

3. Analysis and Design

3.1. System Design

A research method is a specific way to collect and analyze data to determine an answer to a research question or hypothesis. During the research process, these concrete actions are taken to obtain the information needed. This study uses the prototype method, which combines ideas from the system development method to create a prototype of the research system that will show the results of the research.

3.2. Block Diagram Network

The block diagram of the designed system, as shown in the Figure 1:

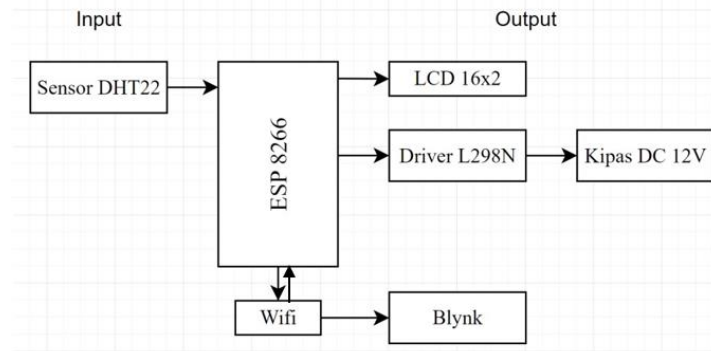


Fig.1: Block Diagram Network

Here is a brief explanation of the diagram block series:

1. The temperature sensor functions for data input connected to the MCU Node to be managed.
2. LCD functions to display the temperature data that has been read.
3. Then ESP8266 connect to the L298N Driver which will set the speed of the 12V DC Fan with the already managed temperature data.
4. The data that has been generated is then sent ESP8266 through the internet network to the Blynk Application which will later become an interface for users.
5. In the Blynk application, you can adjust the fan speed and view the read temperature data.

3.3. Overall Tool Set

The following figure is the schematic of the automatic fan control circuit :

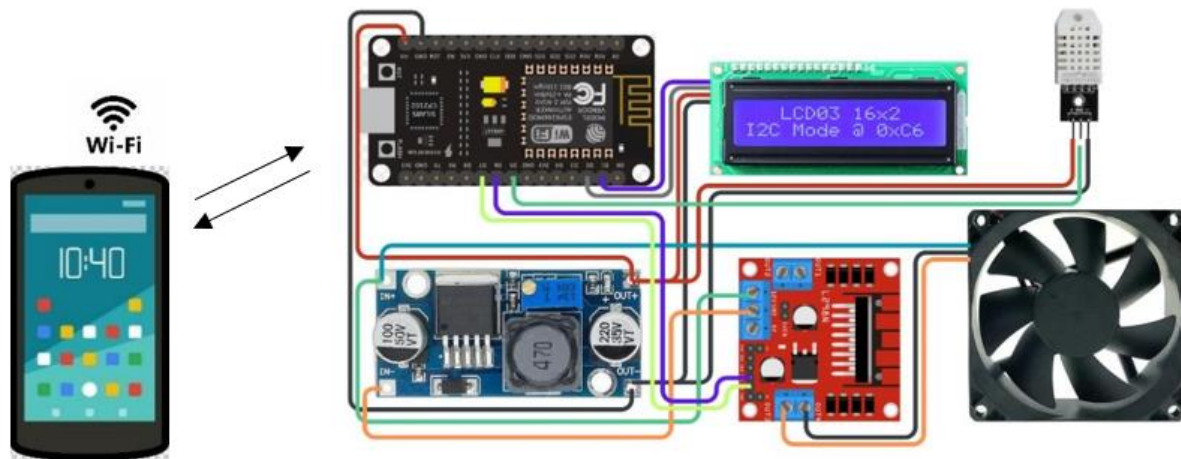


Fig.2: Overall Tool Set

The following is an explanation from Figure III.2 above:

1. The LM2596s regulator functions to distribute power to other components. Such as Fans and Motor Drivers that require 12Volt Power comes from the direct adapter, and other devices require 5V Power from OUT+.
2. NodeMCU ESP8266 has pins connected to various tools: Pin D1 is connected to the SCL Pin on the LCD, Pin D2 is connected to the SDA Pin on the LCD, Pin D5 is connected to the DHT22 Sensor, and Pin D6 and Pin D7 are connected to In3 and In4 on the Motor Driver.
3. DHT22 Temperature Sensor i.e. VCC and GND Pins are connected to positive and negative on OUT+ and OUT+ Regulators, Data pins are connected to D5 NodeMCU pins for temperature data input.
4. The 16x2 LCD in the above scheme is the positive and negative power pins connected to the positive and negative on the OUT+ and OUT+ Regulators, and the SCA and SCL pins are connected to the NodeMCU as a communication series.
5. The 12V pin on the Motor Driver is connected to IN+ on the Regulator, the GND pin is connected to IN- on the Regulator, OUT3 and OUT4 are connected to the DC Fan The 12V DC Fan has 4 wires namely the VCC Pin is connected to IN+ on the Regulator, the GND is connected to OUT4 on the Motor Driver, the PWM is connected to OUT3 on the Motor Driver, and the Tachometer on the fan is not connected anywhere.

3.4. System Flowchart

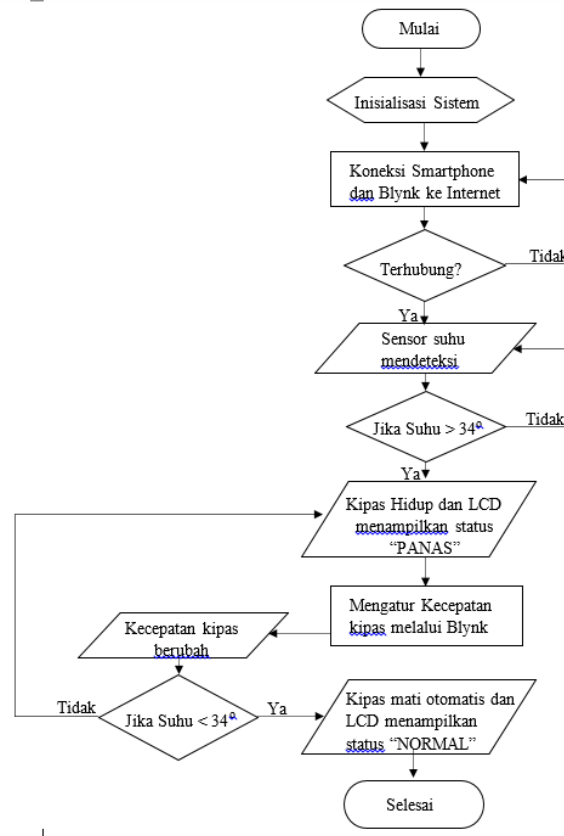


Fig. 3: Flowchart System Control

Description:

1. Start the testing process.
2. Initial initialization i.e. preparation of the device in ready-to-use mode.
3. Connect your Smartphone or computer to the internet network and ESP8266 to the Wi-Fi network that has been included in the program's coding.
4. Next, connect to the internet. If Yes, then to the next process, if No then back to connecting to the internet
5. When the device is connected to the internet, the next process, namely the temperature sensor, works to detect the room temperature.
6. Next is the branching process, if the Temperature is greater than 34o then the next process will be carried out, and if not then the sensor again detects the room temperature.
7. If the previous process is correct, then the fan will turn on and the LCD shows "STATUS = HOT".
8. If the fan is already on, then you can adjust the speed of the fan using the Blynk software on the smartphone, the data will be communicated through the Blynk Server.
9. The data that has been communicated from Blynk will be received by NodeMCU and change the speed of the fan.
10. Then there is an option if the temperature is less than 34o then the fan will turn off and the LCD shows "STATUS=NORMAL".
11. Done.

4. Discussion and Implementation

4.1. Discussion

In this chapter, we describe and explain the results of the research by conduct tests. The tests that will be carried out are software and hardware testing.

4.2. Software Testing

This experiment starts with the following steps to test the program:

1. Open the Arduino IDE software with the following display:

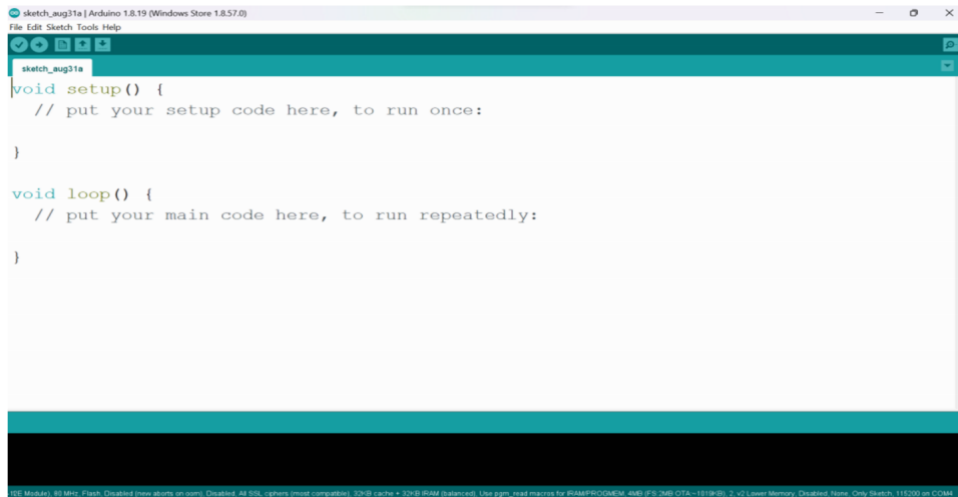


Fig. 4: Arduino IDE Initial Look

2. Enter the program as below which is intended for Blynk and Wi-Fi Connection and initialization of components:

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"

#define DHTPIN D5
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27,16,2);

char ssid[] = "FAN-IOT";
char pass[] = "FAN123456";
const int pinFan = D7; const
int pinPwm = D6;
```

Fig. 5: Blynk Connection Program

3. Then below is the program to set the Speed and Modes on Fans set via Blynk:

```
int temp, humid;
unsigned long last;
String mode, status; int
dt, pwm;
int setSuhu = 34; //derajat celcius

BLYNK_WRITE(V2){
  int val = param.asInt();
  dt = val;
  pwm = map(dt, 0, 100, 0, 255);
}

BLYNK_WRITE(V0){
  int val = param.asInt();
  if(val == 1){
```

Fig. 6: Program to Set the Speed and Modes on the Fan

4. Then is the program initialize all components and the 16x2 LCD displays the Wi-Fi connection status:

```
void setup() {
  Serial.begin(9600);
  lcd.begin();
  lcd.backlight();
  Serial.begin(9600);
  lcd.setCursor(0,0);
  lcd.print("WIFI CONNECTING");
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  lcd.clear(); lcd.setCursor(0,0);
  lcd.print("WIFI CONNECTED");
  delay(1000);
  lcd.clear();
  pinMode(pinFan,OUTPUT);
  pinMode(pinPwm,OUTPUT);
  digitalWrite(pinFan,HIGH);
  dht.begin();
  delay(1000);
}
```

Fig. 7: Program initializes all components and the 16x2 LCD displays the Wi-Fi connection status

5. Next is the program to work in two modes: automatic based on the measured temperature, or manual with control from the user via the Blynk app:

```
void loop() { Blynk.run();
  if(millis() - last > 1000){
    cekSuhu();
    if(temp < setSuhu)status = "NORMAL";
    if(temp > setSuhu)status = "PANAS ";
    tampil();
    Blynk.virtualWrite(V1,temp);
    last = millis();
  }

  if(mode == "AUTO"){
  if(temp < setSuhu){
  digitalWrite(pinFan,HIGH);
  analogWrite(pinPwm,0);
  }
  if(temp > setSuhu){
  if(pwm <= 0)pwm = 20;
  digitalWrite(pinFan,LOW);
  analogWrite(pinPwm,pwm);
  }
  }else{ if(pwm == 0){
  digitalWrite(pinFan,HIGH);
  analogWrite(pinPwm,0);
  }else{
  digitalWrite(pinFan,LOW);
  analogWrite(pinPwm,pwm);
  }
  }
  delay(10);
}
```

Fig. 8: Programs to Work in Two Modes

6. Then the program to display the Temperature Status on the LCD and read the temperature and humidity data from the DHT22 sensor:

```
lcd.print("C ");

  lcd.setCursor(0,1);
  lcd.print("STATUS: ");
  lcd.print(status);
}

void cekSuhu(){ humid =
dht.readHumidity(); temp =
dht.readTemperature(); if
(isnan(humid) || isnan(temp)) {
  Serial.println(F("Failed to read from DHT sensor!"));
return;
}

  Serial.print(F("Humidity: "));
  Serial.print(humid);
  Serial.print(F("% Temperature: "));
  Serial.print(temp);
  Serial.println(F("°C")); }
```

Fig. 9: Program to display the Temperature Status on the LCD

4.3. Hardware Testing

After all the programs have been successfully typed, here is the hardware design that has been created :

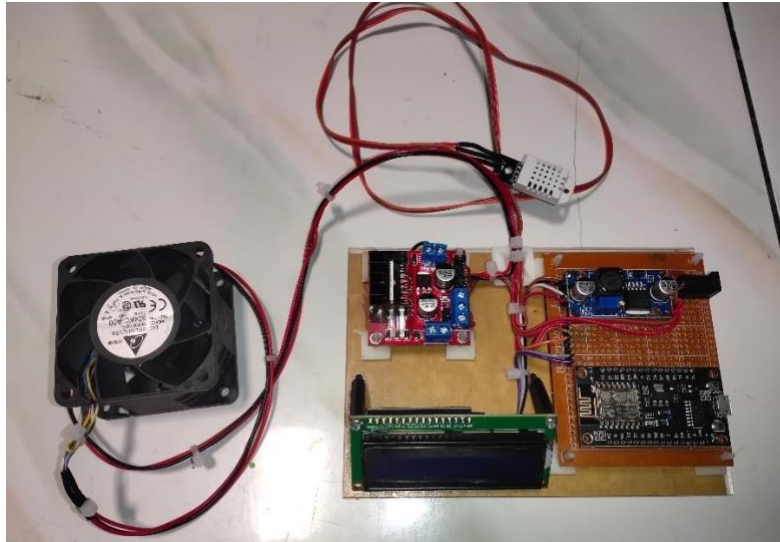


Fig. 10: Automatic Fan Circuit Drawing by Temperature

4.4. Blynk Testing

In the Blynk Software, create a template as shown in Figure below

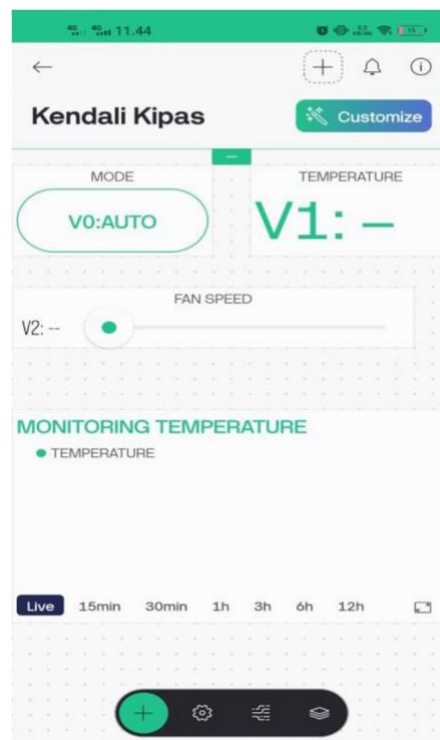


Fig. 11: Blynk Display

Once the program is fully typed, the next step is to insert the program code into the network. This is done by clicking on the Menu Bar on the Arduino IDE, then clicking "Upload" with the note that the Board and Ports must be appropriate. Once this process is complete, wait a few moments for the program to automatically save to the NodeMCU ESP8266.

4.5. Overall Tool Testing Implementation

This test is carried out to measure the performance of the parts that will be discussed in this thesis. The output of this test is the Blynk app, which automatically controls the fan based on the temperature displayed in realtime. With a smartphone connected to the internet, the microcontroller can receive and execute commands sent by Blynk to control the fan.

This experiment was carried out by regulating the temperature and ensuring that the fan could be turned on or off automatically according to the measured temperature. In addition, the test also monitors the temperature on the LCD in real-time when indoors and outdoors with different temperatures. After all the circuits are designed, here are the results of the Internet of Things-based automatic fan control test:

1. The test results below are the test results that are in the room at 12.30 WIB. It can be seen that the Temperature is at 31°C and the Status is "NORMAL" and the Fan is Off.

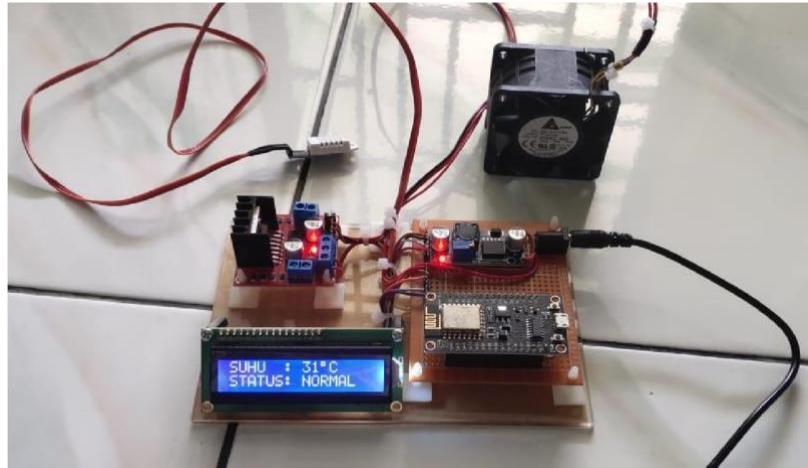


Fig. 12: Test Results at 31°C

2. The results below are tests that were outdoors at 12.40 WIB. It can be seen that the Temperature is at 37°C and the Status is "HOT" and the Fan is on.

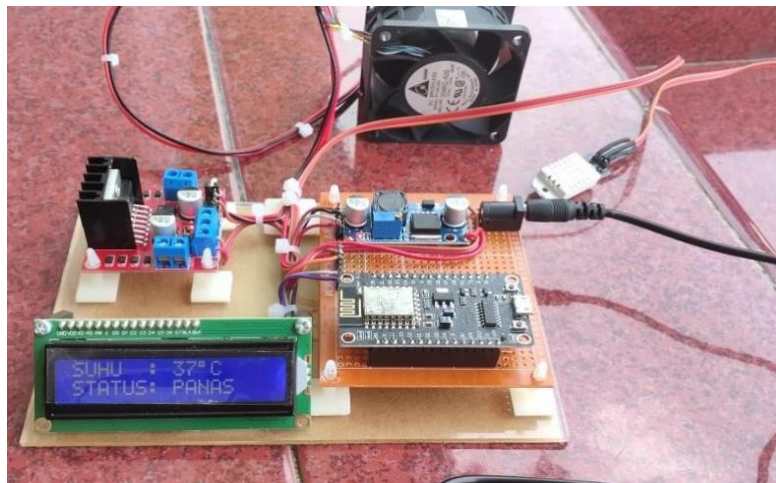


Fig. 13: Test Results at 37°C

3. In the Blynk application you can adjust the speed of the fan. When in "AUTO" Mode, the fan turns on based on the detected temperature sensor limits, while when in "MANUAL" Mode, the fan turns on based on the set speed.



Fig. 14: Blynk in AUTO Mode



Fig. 15: Blynk in MANUAL Mode

5. Conclusion

After carrying out the design and manufacturing stage of the system which is then followed by the testing and analysis stage, the following conclusions can be drawn:

1. In designing an automatic fan control system based on the Internet of Things (IoT). The main hardware used includes a NodeMCU ESP8266 as a microcontroller, a DHT22 sensor to detect temperature, and a DC Fan controlled via the L298N Motor Driver. Furthermore, programming was carried out to integrate the hardware with the Blynk application, which allows remote control and monitoring of fan temperature and speed in real-time. The system is capable of turning the fan on and off automatically according to the detected temperature conditions.
2. Implementation of IoT-based automatic fan speed control system, users can perform manual control through the Blynk application which is able to adjust the fan speed through a percentage of 0-100, providing flexibility in the operation of the tool. This implementation has succeeded in producing a responsive and efficient system in managing temperature automatically.
3. Although it succeeded in achieving the main goal, this research still has limitations, such as still using a 12V DC fan that is not able to manage the temperature in the room.

Reference

- [1] Nurhayati, Ana Naela, Ahmat Josi, and Nur Aini Hutagalung. 2018. "Sales." *Journal of Technology and Information* 7(2): 13–23.
- [2] Friadi, Roby, and Junadhi Junadhi. 2019. "Light Intensity, Temperature and Air Humidity Control System in Raspberry PI-Based Greenhouses." *Journal of Technopreneurship and Information System (JTIS)* 2(1): 30–37.
- [3] Nadiansyah, Ricza Rahmad. 2018. "Sistem Pengendali Kipas Angin Berbasis Nodemcu Esp8226." *Universitas Teknologi Digital Indonesia*: 3–10. <http://eprints.akakom.ac.id/id/eprint/8173>.
- [4] Nurul Hidayati Lusita Dewi, Mimin F. Rohmah, Soffa Zahara Students. 2019. "Prototype Smart Home with Nodemcu ESP8266 Module Based on Internet of Things (IoT)." *Information Technology*: 3–3.
- [5] Sima, Brema Arisma, Relita Buatun, and Marto Sihombing. 2024. "IOT Based Automatic Light Control System Using MQTT Protocol." 3(1): 1–9.
- [6] Abdulrazzak, Ibtihaj Abdulwahhab et al. 2018. "Humidity and Temperature Monitoring." *International Journal of Engineering & Technology* 7(4): 5174–77.
- [7] Khairul, H, R Buatun, and ... 2024. "Use of Internet of Things (IOT) on Fish Feeding Tools with Nodemcu." ... *Journal of Informatics* ... 3(1): 10–17.
- [8] Zanofo, Arief Pratama, Ristiandika Arrahman, Muhammad Bakri, and Arief Budiman. 2020. "Arduino Uno R3 Microcontroller-Based Automatic Gateway." *Journal of Computer Engineering and Systems* 1(1): 22–27.
- [9] PELAWI, GOESTI MESKANA PELAWI, ACHMAD FAUZI, and MILLI ALFHI SYARI. 2023. "Design Of Electric Switching Systems For Electronic Equipment In Home Based Internet of Things (IoT)." *Journal of Artificial Intelligence and Engineering Applications (JAIEA)* 3(1): 368–72.