

Bidang: Teknik Elektro, Listrik, dan otomasi

Topik: Teknologi Rekayasa Otomasi

## Penerapan Node-Red dan Wokwi Berbasis Cloud untuk Pembelajaran Sistem Manufaktur Diskrit pada Pengenalan Internet of Things (IoT) dan Protokol MQTT

Nurhayati Jabir<sup>1</sup>, Taufik Muchtar<sup>2</sup>, Mutmainnah<sup>3</sup>, Sukriyah Buwarda<sup>4</sup>, Lutfi<sup>5</sup>  
<sup>1,2,3,4,5</sup> Politeknik ATI Makassar

nurhayati.djabir@atim.ac.id<sup>1</sup>, taufik@atim.ac.id<sup>2</sup>, mutmainnah@atim.ac.id<sup>3</sup>,  
sukriyah.buwarda@atim.ac.id<sup>4</sup>, lutfi@atim.ac.id<sup>5</sup>

### ABSTRAK

Penelitian ini mengeksplorasi potensi penggunaan Node-RED dan Wokwi dalam pembelajaran sistem manufaktur diskrit, dengan fokus pada penerapan Internet of Things (IoT) dan protokol MQTT. Konsep IoT yang didorong oleh kemajuan teknologi informasi memungkinkan interkoneksi perangkat fisik untuk pengumpulan data secara real-time, meningkatkan efisiensi dan kualitas produk dalam industri. Kombinasi Node-RED dan Wokwi memberikan lingkungan belajar interaktif yang memudahkan mahasiswa dalam membangun prototipe sistem IoT. Hasil simulasi menunjukkan bahwa mekanisme komunikasi MQTT berfungsi dengan baik, meskipun terdapat kendala komunikasi yang diakibatkan oleh penggunaan virtual WiFi access point. Penelitian ini menegaskan bahwa pendekatan berbasis simulasi dapat menjadi alat yang efektif dalam mempersiapkan mahasiswa untuk menghadapi tantangan di era industri 4.0.

**Kata kunci:** Internet of Things (IoT), MQTT, Node-RED, Wokwi, Sistem Manufaktur Diskrit.

### ABSTRACT

*This research explores the potential use of Node-RED and Wokwi in learning discrete manufacturing systems, focusing on the application of the Internet of Things (IoT) and the MQTT protocol. The IoT concept, driven by advancements in information technology, enables the interconnection of physical devices for real-time data collection, enhancing efficiency and product quality in industry. The combination of Node-RED and Wokwi provides an interactive learning environment that facilitates students in building IoT system prototypes. Simulation results indicate that the MQTT communication mechanism functions effectively, although communication issues arise due to the use of a virtual WiFi access point. This study emphasizes that a simulation-based approach can be an effective tool in preparing students to face challenges in the era of Industry 4.0.*

**Keywords:** Internet of Things (IoT), MQTT, Node-RED, Wokwi, Discrete Manufacturing Systems.

### PENDAHULUAN

Di Politeknik ATI Makassar, transformasi industri 4.0 dalam konteks manufaktur diskrit dipelajari oleh mahasiswa melalui enam materi utama. Materi-materi tersebut terdiri dari pengenalan transformasi industri 4.0, siklus manufaktur industri 4.0, proses bisnis industri 4.0, teknologi pendukung industri 4.0, strategi implementasi transformasi industri 4.0, dan proyek solusi transformasi industri 4.0. Salah satu teknologi utama dalam industri 4.0 adalah Internet of Things (IoT). Konsep Internet of Things (IoT) telah didorong oleh perkembangan pesat teknologi informasi dan komunikasi. Interkoneksi berbagai perangkat fisik telah dimungkinkan oleh IoT, sehingga memungkinkan pengumpulan dan pertukaran data secara real-time. Dalam konteks industri manufaktur, potensi besar untuk meningkatkan efisiensi, produktivitas, dan kualitas produk telah ditawarkan oleh penerapan IoT. [1]

Platform Node-RED dan Wokwi, yang semakin populer dalam pengembangan sistem IoT, telah digunakan dalam penelitian ini. Antarmuka pemrograman visual yang intuitif dari Node-RED telah memungkinkan pengguna untuk menghubungkan berbagai perangkat dan layanan tanpa perlu menulis kode yang kompleks. Sementara itu, rangkaian

elektronik telah dapat dirancang dan diuji secara virtual melalui simulator elektronik online yang disediakan oleh Wokwi. Lingkungan pengembangan yang fleksibel dan interaktif telah dapat diberikan kepada mahasiswa melalui kombinasi Node-RED dan Wokwi dalam konteks pembelajaran. Prototipe sistem IoT telah dapat dibangun dengan mudah oleh mahasiswa, berbagai skenario telah dapat diuji, dan pemahaman tentang cara kerja protokol komunikasi MQTT telah dapat ditingkatkan. [2] [3]

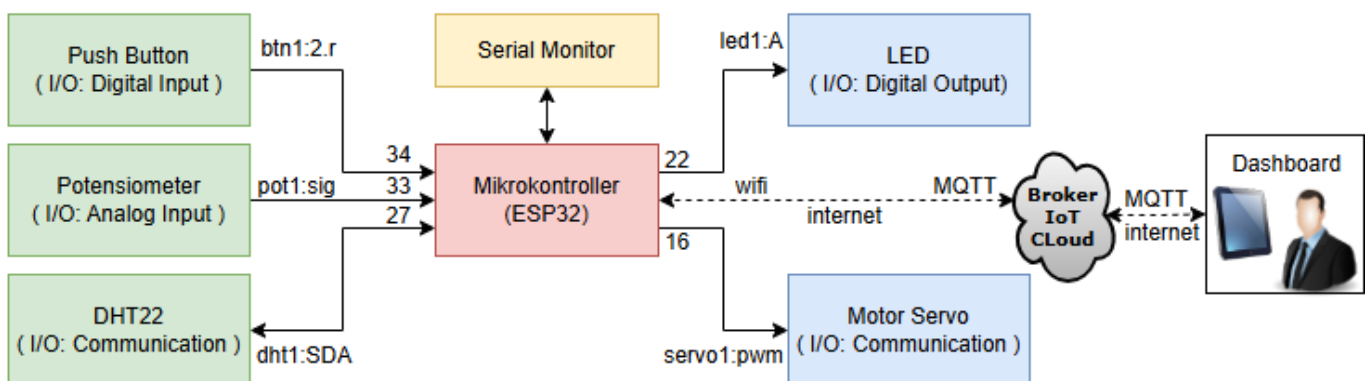
Protokol Message Queuing Telemetry Transport (MQTT), yang telah menjadi standar de facto untuk komunikasi machine to machine (M2M) dalam lingkungan IoT, telah dimanfaatkan dalam penelitian ini. Mekanisme publish-subscribe yang ringan dan efisien dari MQTT telah memungkinkan komunikasi real-time dengan bandwidth yang terbatas dalam aplikasi IoT. Dalam sistem manufaktur diskrit, berbagai sensor, aktuator, dan perangkat cerdas telah dapat dihubungkan melalui protokol MQTT. Pengumpulan dan analisis data produksi secara real-time telah dimungkinkan dengan penggunaan MQTT, sehingga pengambilan keputusan yang lebih cepat dan akurat telah dapat dilakukan. [4] [5] [6] [7]

Penelitian ini bertujuan untuk mengeksplorasi potensi penggunaan Node-RED dan Wokwi dalam pembelajaran sistem manufaktur diskrit. Modul pembelajaran yang terstruktur telah dikembangkan dengan harapan dapat meningkatkan pemahaman mahasiswa tentang konsep IoT dan protokol komunikasi MQTT serta mempersiapkan mereka untuk menghadapi tantangan di dunia industri yang semakin terdigitalisasi.

### METODE PENELITIAN

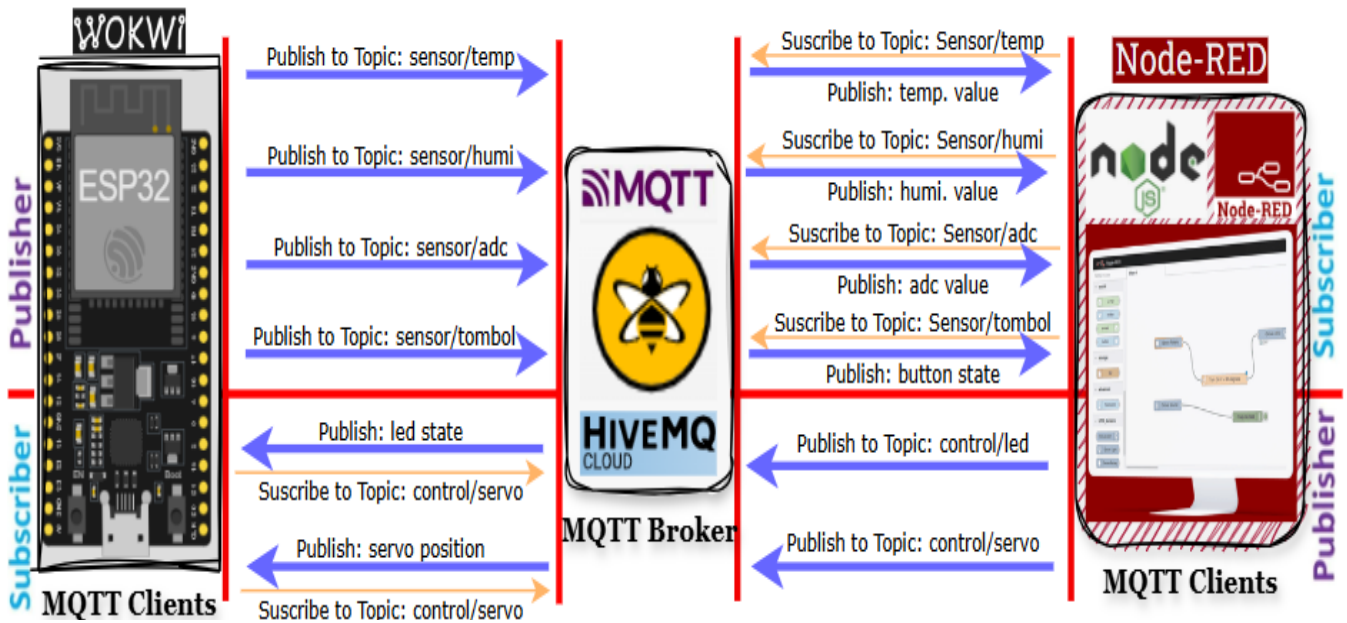
Metode penelitian yang digunakan adalah metode eksperimen, yang melibatkan observasi terhadap variabel-variabel objek yang diteliti. Dalam metode eksperimen, kondisi tertentu dikendalikan untuk memungkinkan pengendalian satu atau beberapa variabel. Pada penelitian ini digunakan sebuah komputer dengan processor Intel(R) Core(TM) i7-8700 CPU 3.20GHz, RAM 16 GB dan sistem operasi windows 11 dengan sistem 64 bit. Aplikasi yang digunakan yakni aplikasi Wokwi pada link <https://wokwi.com/projects/new/esp32> serta Node-red pada link <http://127.0.0.1:1880> untuk pembuatan node flow diagram dan pada link <http://127.0.0.1:1880/ui> untuk HMI Dashboardnya. Untuk pengujian komunikasi protokolnya digunakan aplikasi cloud MQTT Broker Client dengan link <https://www.hivemq.com/demos/websocket-client/> dengan topic publish-subscribe sensor/temp, sensor/humi, sensor/adc, sensor/tombol, control/led, control/servo. [8] [9] [10]

Penelitian ini melalui beberapa tahapan metodologi yakni Define - Explore - Design - Test - Evaluate. Pada tahap **Define**, ditetapkan tujuan percobaan, yaitu mahasiswa dikenalkan dengan definisi dan arsitektur IoT menggunakan protokol MQTT serta penggunaan aplikasi Cloud untuk belajar IoT melalui Wokwi dan Node-Red. Pada tahap **Explore**, analisa kebutuhan I/O dilakukan untuk mengidentifikasi sensor dan aktuator yang dapat mewakili beberapa tipe I/O, yaitu Digital Input (DI), Digital Output (DO), Analog Input (AI), dan Communication (Comm), meskipun dalam kasus ini hanya tipe DI, DO, AI, dan Comm yang digunakan.



Gambar 1. Diagram Blok Sistem hasil analisa kebutuhan I/O

Selanjutnya, pada tahap **Design**, dibuat wiring diagram dan kode program dengan menggunakan aplikasi Wokwi, serta workflow dan desain HMI dashboard dengan aplikasi Node-Red. Pada tahap **Test**, simulasi sistem dilakukan antara simulator Wokwi dan dashboard Node-Red melalui browser di smartphone, di mana juga dipelajari konsep publisher dan subscriber pada protokol MQTT dengan menggunakan Hive broker client. [11]

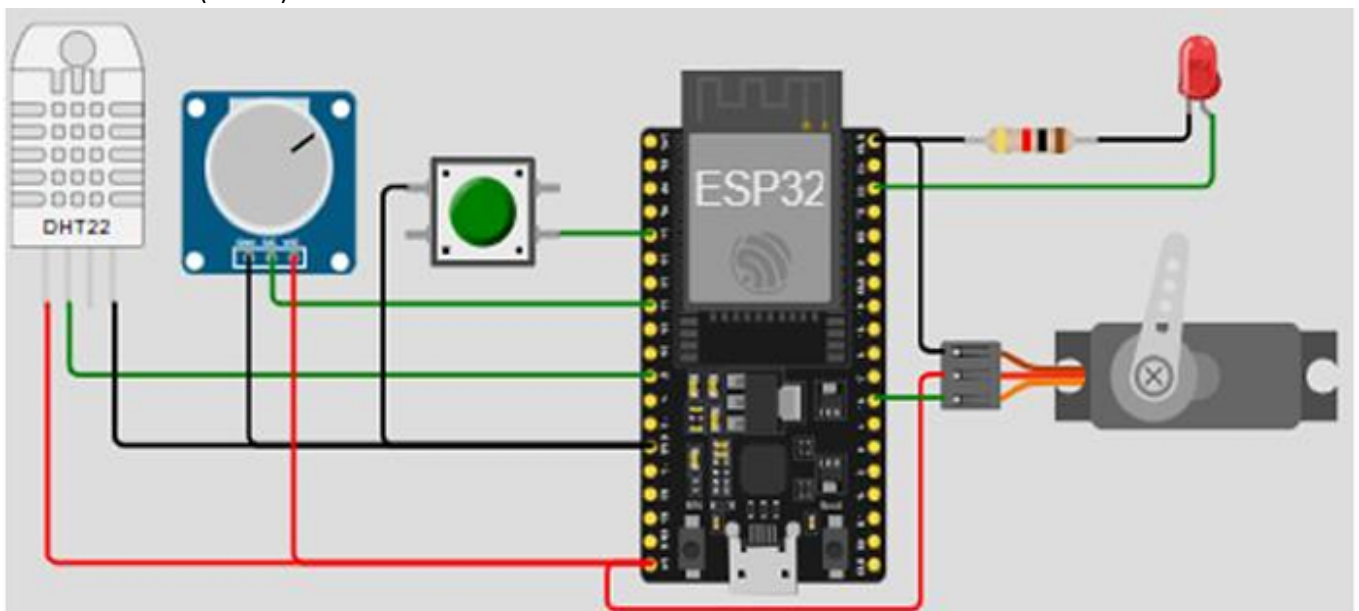


**Gambar 2.** Model Publish/Subscribe untuk protokol MQTT yang akan diuji

Akhirnya, pada tahap **Evaluate**, diskusi antara dosen dan mahasiswa dilaksanakan untuk memberikan kesimpulan dan ide saran perbaikan.

### HASIL DAN PEMBAHASAN

Pada tab simulasi di Wokwi, ESP32 digunakan sebagai mikrokontroler sistem-on-chip berbiaya rendah dan berdaya rendah dengan Wi-Fi terintegrasi serta Bluetooth dual-mode. Dalam rangkaian ini, sebuah push button dipakai sebagai perangkat digital input (DI), potensiometer sebagai perangkat analog input (AI), sensor suhu dan kelembaban DHT22 sebagai perangkat komunikasi one wire (Comm), LED indikator sebagai perangkat digital output (DO), dan motor servo sebagai perangkat komunikasi PWM (Comm).



**Gambar 3.** Wiring Diagram sistem yang telah didesain pada tab simulation di wokwi

Pada pembuatan kode program untuk tab sketch.ino di wokwi simulator, akan dimasukkan beberapa file header melalui tab library manager seperti WiFi.h, PubSubClient.h, DHT.h, dan ESP32Servo.h.

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <ESP32Servo.h>
#define DHT22_PIN 27
```

```
void callback(char* topic, byte* message, unsigned int length) {
  Serial.print("Pesan tiba di topik: ");
  Serial.print(topic);
  Serial.print(". Pesan: ");
  String messageTemp;
```

```

#define SERVO_PIN 16
int ledPin = 22, pb = 34, pot = 33;
const char* ssid = "Wokwi-GUEST";
const char* password = "";
const char* mqtt_server = "mqtt-dashboard.com";
const int mqtt_port = 1883;
const char* mqtt_user = "", mqtt_password = "";
long lastMsg = 0;
Servo servoMotor;
WiFiClient espClient;
PubSubClient client(espClient);
DHT dht22(DHT22_PIN, DHT22);

void setup() {
  setupWiFi();
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
  Serial.begin(115200);
  dht22.begin();
  pinMode(pb, INPUT_PULLUP);
  pinMode(ledPin, OUTPUT);
  servoMotor.attach(SERVO_PIN);
}

void setupWiFi() {
  delay(10);
  Serial.println();
  Serial.print("Menghubungkan ke ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println(""); Serial.println("Terhubung ke WiFi");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Menghubungkan ke MQTT...");
    if (client.connect("ESP32Client", mqtt_user,
mqtt_password)) {
      Serial.println("Terhubung!");
      client.subscribe("control/led");
      client.subscribe("control/servo");
    } else {
      Serial.print("Gagal, rc="); Serial.print(client.state());
      Serial.println(" Coba lagi dalam 5 detik");
      delay(5000);
    }
  }
}

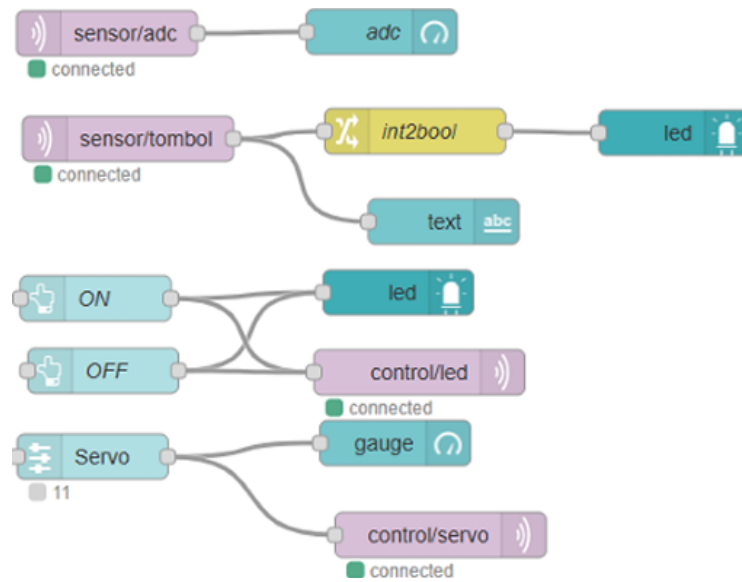
for (int i = 0; i < length; i++) {
  Serial.print((char)message[i]);
  messageTemp += (char)message[i];
}
Serial.println();
if (String(topic) == "control/led") {
  Serial.print("Mengubah output ke ");
  if (messageTemp == "true") {
    Serial.println("true"); digitalWrite(ledPin, HIGH);
  }
  else if (messageTemp == "false") {
    Serial.println("false"); digitalWrite(ledPin, LOW);
  }
}
if (String(topic) == "control/servo") {
  int pos = messageTemp.toInt();
  servoMotor.write(pos);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  long now = millis();
  if (now - lastMsg > 5000) {
    lastMsg = now;
    float humi = dht22.readHumidity();
    float temp = dht22.readTemperature();
    int tombol = digitalRead(pb);
    int adc = analogRead(pot);
    char msg_humi[50], msg_temp[50], msg_tombol[50],
msg_adc[50];
    snprintf(msg_humi, 50, "%.2f", humi);
    snprintf(msg_temp, 50, "%.2f", temp);
    snprintf(msg_tombol, 50, "%i", tombol);
    snprintf(msg_adc, 50, "%i", adc);
    if (isnan(temp) || isnan(humi)) {
      Serial.println("Failed to read from DHT22 sensor!");
    } else {
      Serial.print("Tombol: "); Serial.print(tombol);
      Serial.print(" ADC: "); Serial.print(adc);
      Serial.print(" temp: "); Serial.print(temp);
      Serial.print(" humi: "); Serial.println(humi);
      client.publish("sensor/tombol", msg_tombol);
      client.publish("sensor/adc", msg_adc);
      client.publish("sensor/temp", msg_temp);
      client.publish("sensor/humi", msg_humi);
    }
  }
}

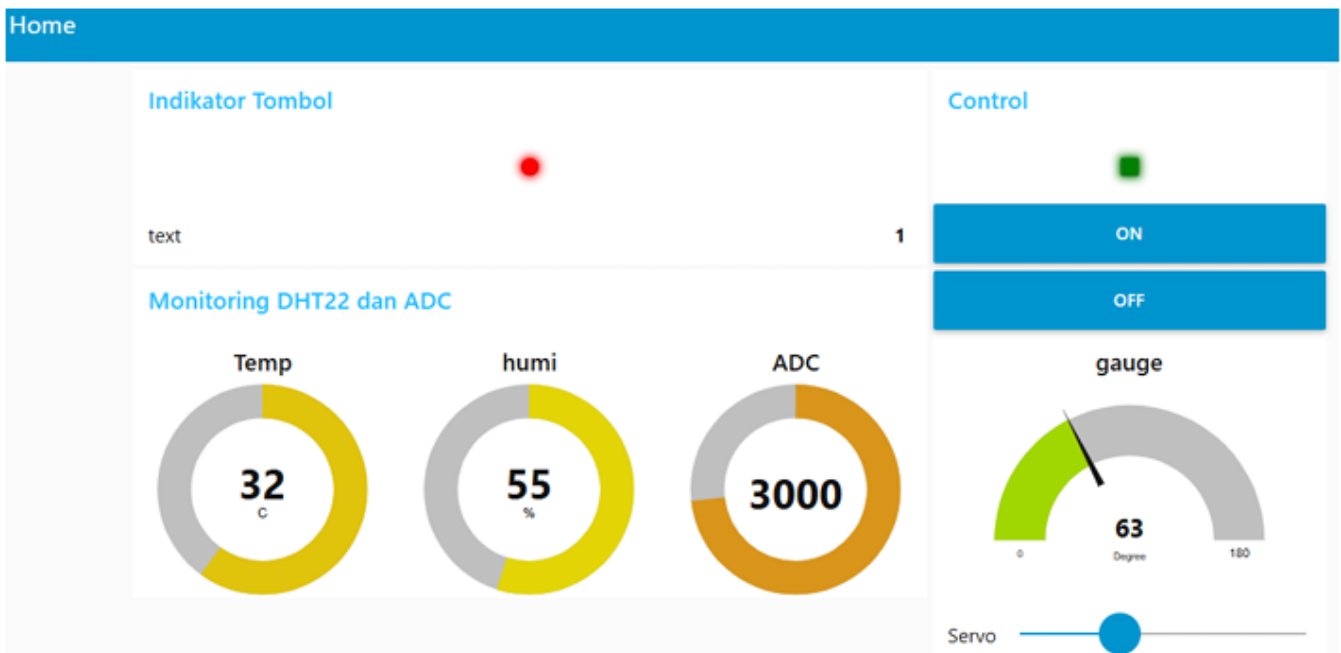
```

Pada pembuatan HMI dashboard berbasis Node-RED, node flow diagram telah dibuat dan ditampilkan pada gambar 4 di bawah. Dalam node flow tersebut, digunakan empat node MQTT in dan dua node MQTT out dengan konfigurasi server: mqtt-dashboard.com:1883 serta topik Publish/Subscribe sesuai gambar 2 sebelumnya. Terdapat empat node gauge dengan tiga tipe donut dan satu tipe gauge, dua node button, satu node slider, satu node text dan dua node LED. Selain itu,

satu node change juga disertakan untuk mengubah tipe data string menjadi boolean. Hasil dashboard dapat dilihat pada gambar 5 di bawah.



**Gambar 4.** Tampilan hasil node flow Diagram yang telah di desain menggunakan aplikasi node-red



**Gambar 5.** Tampilan HMI Dashboard sebagai antarmuka pengguna

### Pembahasan

Hasil desain telah diimplementasikan melalui simulasi Wokwi dan dashboard Node-RED, lalu diuji dengan menggunakan MQTT broker client sebagai perantara komunikasi publish-subscribe. Pengujian antara Node-RED dan MQTT broker client menunjukkan hasil yang memuaskan dengan waktu respons rata-rata kurang dari satu detik. Namun, ketika diuji antara simulasi Wokwi dan MQTT broker client, sering terjadi kegagalan pengiriman data melalui MQTT sehingga waktu respons menjadi lebih lama, yaitu rata-rata lima hingga sepuluh detik. Kegagalan ini diduga disebabkan oleh penggunaan virtual WiFi access point "Wokwi-GUEST". Simulasi Wokwi tanpa menggunakan virtual WiFi access point tetap memberikan hasil yang baik dengan waktu respons kurang dari satu detik. Implementasi langsung kode program ke perangkat keras ESP32 juga menghasilkan waktu respons yang sangat baik, yaitu kurang dari satu detik.

### KESIMPULAN

Penelitian ini berhasil menunjukkan bahwa kombinasi Node-RED dan Wokwi dapat menjadi sebuah lingkungan pembelajaran yang efektif untuk memperkenalkan mahasiswa pada konsep Internet of Things (IoT) dan protokol MQTT dalam konteks manufaktur diskrit. Melalui simulasi dan percobaan yang terstruktur, mahasiswa dapat membangun

prototipe sistem IoT sederhana, memahami mekanisme komunikasi MQTT, dan mengembangkan keterampilan dalam merancang antarmuka pengguna. Hasil penelitian juga menunjukkan bahwa kendala dalam komunikasi MQTT dapat terjadi akibat penggunaan virtual WiFi access point pada simulator Wokwi. Meskipun demikian, penelitian ini memberikan bukti kuat bahwa pendekatan berbasis simulasi ini dapat menjadi alat yang berharga dalam mempersiapkan mahasiswa untuk menghadapi tantangan di era industri 4.0.

#### **DAFTAR PUSTAKA**

- [1] Mujiyono, M. K. Simarmata, Mustofa, M. Agus, L. Heriyanto, M. T. Siregar and Lutfi, TRANSFORMASI INDUSTRI 4.0 MANUFAKTUR DISKRIT, Jakarta Selatan: Pusat Pengembangan Pendidikan Vokasi Industri Badan Pengembangan Sumber Daya Manusia Industri Kementerian Perindustrian Republik Indonesia, 2021.
- [2] CodeMagic LTD., 18 Oktober 2024. [Online]. Available: [https://docs.wokwi.com/?utm\\_source=wokwi](https://docs.wokwi.com/?utm_source=wokwi).
- [3] OpenJS Foundation, [Online]. Available: <https://nodered.org/docs/user-guide/>. [Accessed 25 Oktober 2024].
- [4] CavliWireless, CavliWireless, 8 Juli 2024. [Online]. Available: <https://www.cavliwireless.com/blog/nerdiest-of-things/what-is-the-mqtt-protocol.html>.
- [5] Amazon Web Services, 23 Oktober 2024. [Online]. Available: <https://aws.amazon.com/id/what-is/mqtt/>.
- [6] B. S. Dgp, 6 April 2024. [Online]. Available: <https://medium.com/@bayusamodradgp/belajar-esp32-mqtt-37be6a20618b>.
- [7] D. Tao, 5 Agustus 2024. [Online]. Available: <https://www.emqx.com/en/blog/esp32-connects-to-the-free-public-mqtt-broker>.
- [8] Circuits4you.com, 9 Januari 2019. [Online]. Available: <https://circuits4you.com/2019/01/09/nodemcu-esp8266-mqtt-led-on-off-example/>.
- [9] electronicwings, 25 Maret 2020. [Online]. Available: <https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/>.
- [10] electronicwings, 14 Januari 2024. [Online]. Available: <https://www.electronicwings.com/esp32/esp32-mqtt-client>.
- [11] EMQ Technologies, 15 Februari 2023. [Online]. Available: <https://dev.to/emqx/esp32-connects-to-the-free-public-mqtt-broker-386k>.