

Implementasi IoT untuk Pemantauan Suhu Menggunakan Sensor DHT, Mikrokontroler, dan Protokol MQTT

Ade Hikmat Pauji Ridwan¹, Fahmi Agung Tajul Abidin²

^{1,2}Teknik Informatika, Universitas Teknologi Bandung

¹adehikmatfr@gmail.com, ² fata.wasd@gmail.com

Abstrak

Penelitian ini bertujuan untuk mengembangkan dan mensimulasikan sistem *IoT* untuk deteksi suhu. Sistem ini mencakup penggunaan sensor suhu, mikrokontroler, komunikasi melalui protokol *MQTT*, dan penyimpanan data di *server*. Kami juga membuat diagram jaringan untuk menggambarkan komunikasi antar komponen dalam sistem. Implementasi dilakukan menggunakan *platform* Arduino dan *Python* untuk simulasi serta visualisasi. Hasil penelitian menunjukkan bahwa sistem *IoT* dapat mengukur, mengirim, dan menampilkan data suhu secara efektif dan *real-time*.

Kata kunci: *Internet of Things (IoT)*, *Monitoring*, Pendeteksi Suhu, DHT Sensor

Abstract

This study aims to develop and simulate an IoT system for temperature detection. The system includes the use of temperature sensors, microcontrollers, communication via the MQTT protocol, and data storage on a server. We also created a network diagram to illustrate the communication between components in the system. The implementation was carried out using the Arduino and Python platforms for simulation and visualization. The results show that the IoT system can effectively measure, transmit, and display temperature data in real-time.

Keywords: Internet of Things (IoT), Monitoring, Temperature Detection, DHT Sensor

PENDAHULUAN

Internet of Things (IoT) telah menjadi teknologi yang sangat penting dalam berbagai aplikasi, termasuk pemantauan lingkungan. Deteksi suhu merupakan salah satu aplikasi penting dari IoT, di mana data suhu dari berbagai lokasi dapat dikumpulkan, dianalisis, dan digunakan untuk berbagai tujuan seperti pengendalian lingkungan, pemantauan kesehatan, dan manajemen energi. Penelitian ini bertujuan untuk merancang dan mensimulasikan sistem IoT yang dapat mendeteksi suhu, mengirim data ke server, dan menyimpan data untuk analisis lebih lanjut.

TUJUAN PENELITIAN

Penelitian ini memiliki beberapa tujuan:

1. Merancang sistem *IoT* untuk mendeteksi suhu.
2. Mensimulasikan pengumpulan dan pengiriman suhu dari sensor ke *server*.
3. Membuat diagram jaringan untuk menggambarkan komunikasi antar komponen dalam sistem.
4. Menganalisis kinerja sistem dalam hal akurasi pengukuran suhu dan efisiensi pengiriman data.

METODOLOGI

Penelitian ini dilakukan melalui beberapa tahapan:

1. Pengaturan *Hardware*
Menggunakan sensor suhu (*DHT11*), mikrokontroler (Arduino), dan modul Wifi.
2. Pengumpulan Data
Sensor suhu mengukur suhu dan mengirimkan data ke mikrokontroler.
3. Pengiriman Data
Mikrokontroler mengirimkan suhu ke server menggunakan protokol *MQTT*.
4. Penyimpanan Data
Server menyimpan data di *database* untuk analisis lebih lanjut.
5. Visualisasi Data
Data yang dikumpulkan divisualisasikan menggunakan aplikasi web.
6. Diagram Jaringan
Menggunakan pustaka *networkx* dan *matplotlib* untuk membuat diagram komunikasi antar komponen

IMPLEMENTASI SISTEM

Hardware

Implementasi yang dilakukan dengan menghubungkan sensor *DHT11* ke Arduino, yang kemudian mengirimkan data suhu melalui modul wifi ke *server*. Berikut adalah komponen-komponen yang digunakan:

- Sensor Suhu *DHT11*
Mengukur suhu lingkungan.
- Mikrokontroler Arduino
Mengumpulkan data dari sensor dan mengirimkannya ke *server*.
- Modul Wifi *ESP8266*
Menghubungkan arduino ke internet untuk mengirim data ke *server*.

Software

Kode yang digunakan pada Arduino untuk mengumpulkan dan mengirim data suhu ke server adalah sebagai berikut:

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

#define DHTPIN 2
#define DHTTYPE DHT11
DHT dhtSensor(DHTPIN, DHTTYPE);

const char* wifiSSID = "your_SSID";
const char* wifiPassword = "your_PASSWORD";
const char* mqttServer = "broker.mqtt-dashboard.com";

WiFiClient wifiClient;
PubSubClient mqttClient(wifiClient);

void setup() {
  Serial.begin(115200);
  connectToWiFi();
  mqttClient.setServer(mqttServer, 1883);
  dhtSensor.begin();
}

void connectToWiFi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(wifiSSID);

  WiFi.begin(wifiSSID, wifiPassword);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
Serial.println();
Serial.println("WiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
}

void loop() {
  if (!mqttClient.connected()) {
    reconnectToMQTT();
  }
  mqttClient.loop();

  float humidity = dhtSensor.readHumidity();
  float temperature = dhtSensor.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  String dataPayload = "Temperature: " +
    String(temperature) + " *C, Humidity: " + String(humidity)
    + " %";
  Serial.print("Publishing message: ");
  Serial.println(dataPayload);

  mqttClient.publish("sensor/temperature",
    dataPayload.c_str());
  delay(2000);
}

void reconnectToMQTT() {
  while (!mqttClient.connected()) {
    Serial.print("Attempting MQTT connection...");
    if (mqttClient.connect("ArduinoClient")) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.print(mqttClient.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}
```

Server dan Penyimpanan Data

Server menerima data melalui protokol *MQTT* dan menyimpan di *database*. Berikut adalah contoh kode *server* menggunakan Flask dan MongoDB:

```
from flask import Flask, request, jsonify
from pymongo import MongoClient
```

```
import datetime

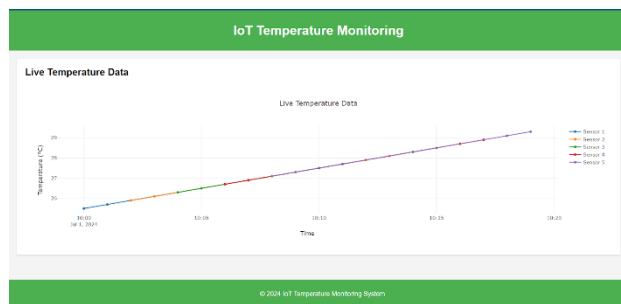
app = Flask(__name__)
mongo_client = MongoClient('localhost', 27017)
database = mongo_client['temperature_database']
temp_collection = database['temp_data']

@app.route('/temperature', methods=['POST'])
def handle_temperature_post():
    temperature_data = request.json
    temperature_data['timestamp'] =
    datetime.datetime.now()
    temp_collection.insert_one(temperature_data)
    return jsonify({"status": "success"}), 200

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

Visualisasi Data

Data yang dikumpulkan disimpan dalam *database* dan divisualisasikan menggunakan aplikasi web yang dibuat dengan Dash. Berikut adalah contoh kode untuk aplikasi web:



```
    dcc.Interval(id='update-interval', interval=1*1000) #
Interval set to 1 second
])

# Define callback to update graph
@app.callback(Output('live-temperature-graph', 'figure'),
              [Input('update-interval', 'n_intervals')])
def update_live_graph(n_intervals):
    # Fetch latest 20 temperature records from MongoDB
    recent_data = list(temp_collection.find().sort('_id', -
1).limit(20))
    recent_data.reverse() # Reverse the list to have the
oldest data first

    # Extract temperature and timestamp data
    temperature_values = [record['temperature'] for record
in recent_data]
    timestamp_values = [record['timestamp'] for record in
recent_data]

    # Create the figure for the graph
    graph_figure = {
        'data': [{ 'x': timestamp_values, 'y':
temperature_values, 'type': 'line', 'name': 'Temperature' }],
        'layout': {
            'title': 'Live Temperature Data'
        }
    }
    return graph_figure

# Run the Dash app
if __name__ == '__main__':
    app.run_server(debug=True)
```

```
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
from pymongo import MongoClient
import pandas as pd

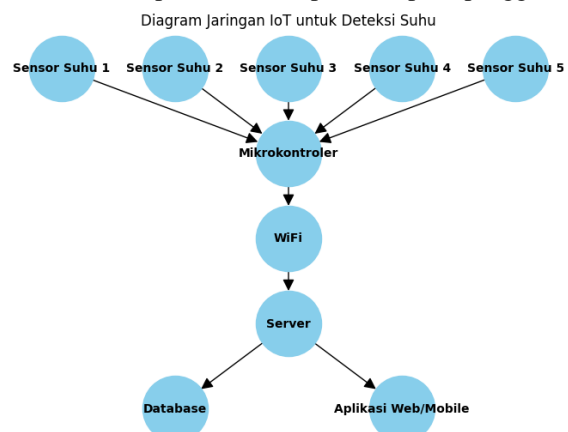
# Initialize the Dash app
app = dash.Dash(__name__)

# Set up MongoDB connection
mongo_client = MongoClient('localhost', 27017)
database = mongo_client['temperature_database']
temp_collection = database['temp_data']

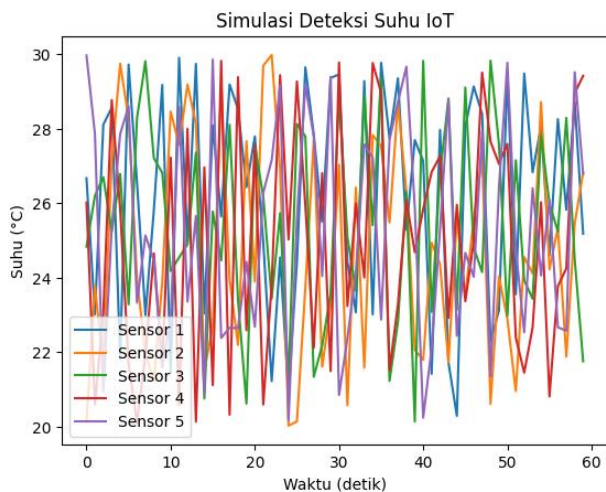
# Define the layout of the Dash app
app.layout = html.Div([
    dcc.Graph(id='live-temperature-graph', animate=True),
```

Diagram Jaringan

Diagram jaringan dibuat untuk menggambarkan komunikasi antara sensor suhu, mikrokontroler, Wifi, server, database, dan aplikasi web/mobile. Diagram ini menunjukkan bagaimana data suhu sudah dikumpulkan, dikirim, disimpan, dan ditampilkan kepada pengguna.



Simulasi Deteksi Suhu



PENUTUP

Simpulan

Penelitian ini berhasil merancang dan mensimulasikan sistem *IoT* untuk deteksi suhu. Sistem ini menggunakan sensor *DHT11* untuk mengukur suhu, Arduino untuk mengirim data ke server, dan server untuk menyimpan serta menampilkan data suhu melalui aplikasi web. Diagram jaringan yang dibuat memberikan gambaran yang jelas tentang komunikasi antar komponen dalam sistem. Implementasi ini menunjukkan potensi besar *IoT* dalam pemantauan lingkungan dan aplikasi lainnya.

Saran

Penelitian lebih lanjut diperlukan untuk mengembangkan desain sensor yang lebih baik sehingga lebih layak digunakan oleh pengguna.

DAFTAR PUSTAKA

- A. Bahga and V.K. Madiseti, "*Internet of Things: A Hands-On Approach*," VPT, 2014.
- M. Collina, G. E. Corazza, and A. Vanelli-Coralli, "*Introducing the QEST broker: Scaling the IoT by bridging MQTT and REST*," IEEE World Forum on Internet of Things (WF-IoT), 2014.
- C. Pfister, "*Getting Started with the Internet of Things: Connecting Sensors and Microcontrollers to the Cloud*," Maker Media, 2011.
- P. Raj and A. C. Raman, "*The Internet of Things: Enabling Technologies, Platforms, and Use Cases*," CRC Press, 2017.
- R. Buyya and A. V. Dastjerdi, "*Internet of Things: Principles and Paradigms*," Morgan Kaufmann, 2016.