

Laporan Praktikum Teknologi IOT

Lampu Otomatis menggunakan Sensor Suara



Kelompok 11 :

13321005_Frengky Manurung

13321017_Nova Sirait

**INSTITUT TEKNOLOGI DEL
FAKULTAS VOKASI**

A. Pendahuluan

Pada proyek ini kami membuat sistem lampu otomatis menggunakan sensor Suara. Dalam implementasinya, kami menggunakan Raspberry Pi sebagai perangkat mikrokontroler dan led untuk mengetahui apakah led dapat merespon output sinyal dari mikrokontroler. Sinyal output tersebut akan menentukan apakah lampu akan menyala atau mati. Sensor Suara akan bekerja dengan nilai yang dimana nilai tersebut akan berubah-ubah yang tergantung pada intensitas suara yang diterima oleh sensor.

B. Hasil dan Implementasi

1. Komponen

Ada beberapa komponen yang diperlukan untuk membangun sistem “Lampu Otomatis”, sebagai berikut:

- Breadboard
- Raspberry pi 4
- Sensor Suara
- Led
- Kabel Jumper

2. Code program

Berikut adalah kode program yang mengintegrasikan Raspberry Pi dengan sensor suara, LED, dan broker MQTT. Code ini dirancang untuk menggunakan sensor suara guna mendeteksi intensitas suara, mengatur LED sesuai dengan nilai sensor, serta berkomunikasi dengan broker MQTT untuk mentransfer data dan menerima instruksi pengendalian.

```
import time
import json
import paho.mqtt.client as mqtt
import RPi.GPIO as GPIO

broker = '192.168.43.44' # Ganti dengan alamat IP broker MQTT Anda
port = 1883
topic_publish = "sensor/"
client_id = 'python-mqtt'
username = 'test'
password = 'test'

GPIO.setmode(GPIO.BCM)
SOUND_PIN = 17 # Pin sensor suara (mikrofon)
LAMP_PIN = 27 # Pin untuk lampu

def connect_mqtt():
    def on_connect(client, userdata, flags, rc):
        if rc == 0:
            print("Terhubung ke Broker MQTT!")
        else:
            print("Gagal terhubung, kode return %d\n" % rc)
```

```

client = mqtt.Client(client_id)
client.username_pw_set(username, password)
client.on_connect = on_connect
client.connect(broker, port)
return client

def read_sound_sensor():
    GPIO.setup(SOUND_PIN, GPIO.IN)
    sound_value = GPIO.input(SOUND_PIN)
    if sound_value == GPIO.LOW: # Jika suara terdeteksi
        return "Suara Terdeteksi"
    else: # Jika suara tidak terdeteksi
        return "Suara Tidak Terdeteksi"

def publish_sound_data(client):
    GPIO.setup(LAMP_PIN, GPIO.OUT) # Set up pin lampu sebagai output

    while True:
        sound_value = read_sound_sensor()

        sound_message = {
            "Sound": sound_value
        }
        sound_msg = json.dumps(sound_message)
        client.publish(topic_publish, sound_msg)
        print(f"Published Sound Data - Value: {sound_value}")

        # Menyalakan lampu jika suara terdeteksi
        if sound_value == "Suara Terdeteksi":
            GPIO.output(LAMP_PIN, GPIO.HIGH) # Menyalakan lampu
            print("Lampu dinyalakan!")
        else:
            GPIO.output(LAMP_PIN, GPIO.LOW) # Mematikan lampu
            print("Lampu dimatikan!")

        time.sleep(5) # Menunggu 5 detik sebelum membaca ulang sensor

def on_message(client, userdata, msg):
    # Fungsi untuk menangani pesan MQTT
    # (tanpa perubahan karena tidak berkaitan dengan sensor suara)
    pass # Ganti 'pass' dengan logika penanganan pesan jika diperlukan

def run():
    client = connect_mqtt()
    client.on_message = on_message # Set fungsi on_message untuk handle pesan
    client.loop_start()

    publish_sound_data(client) # Publish data sensor suara

    try:
        while True:
            # Melakukan apapun yang diperlukan selama program berjalan
            # Implementasi sensor suara di sini (jika diperlukan)
            time.sleep(1)
    except KeyboardInterrupt:

```

```
print("Keyboard interrupt detected. Cleaning up GPIO.")
GPIO.cleanup()
```

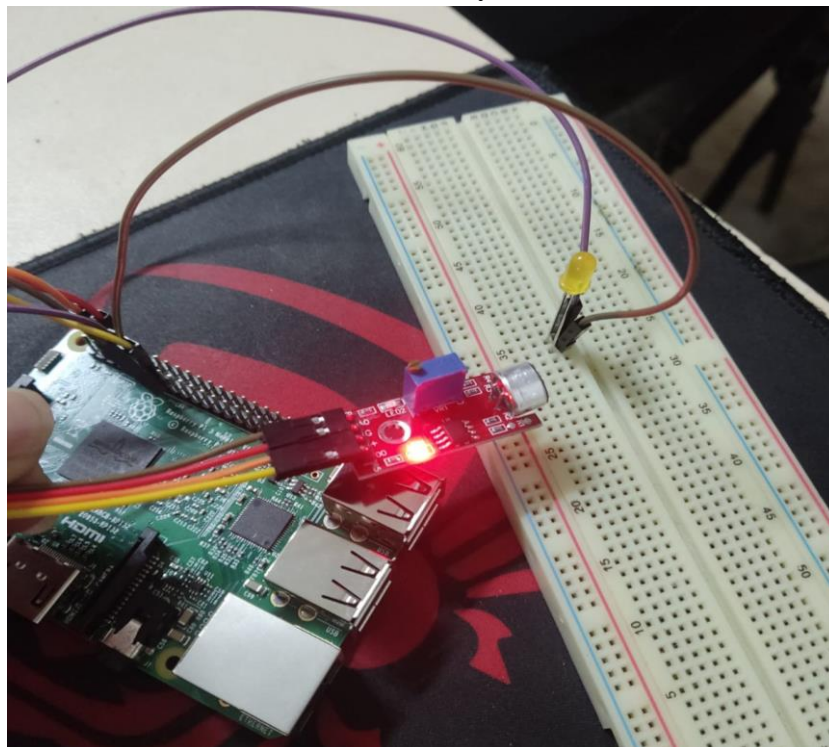
```
# Bagian utama kode
if name == 'main':
    run()
```

Memastikan untuk menghubungkan sensor suara pada pin GPIO yang sesuai dan sesuaikan konfigurasi MQTT sesuai dengan broker yang akan Anda gunakan. Program ini akan membaca nilai dari sensor suara secara periodik, mengaktifkan atau menonaktifkan LED berdasarkan nilai sensor, dan mengirim status LED ke broker MQTT.

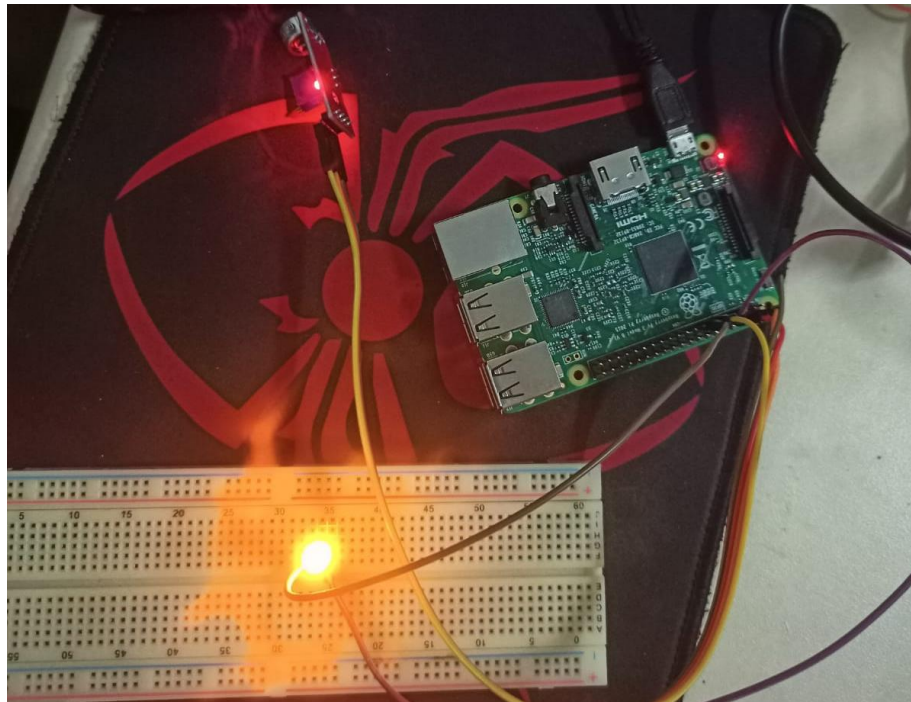
3. Rangkaian

Berikut adalah rangkaian dari proyek kami yaitu lampu otomatis menggunakan sensor Suara pada *Smart Home*.

Jika cahaya terdeteksi maka led tidak akan menyala

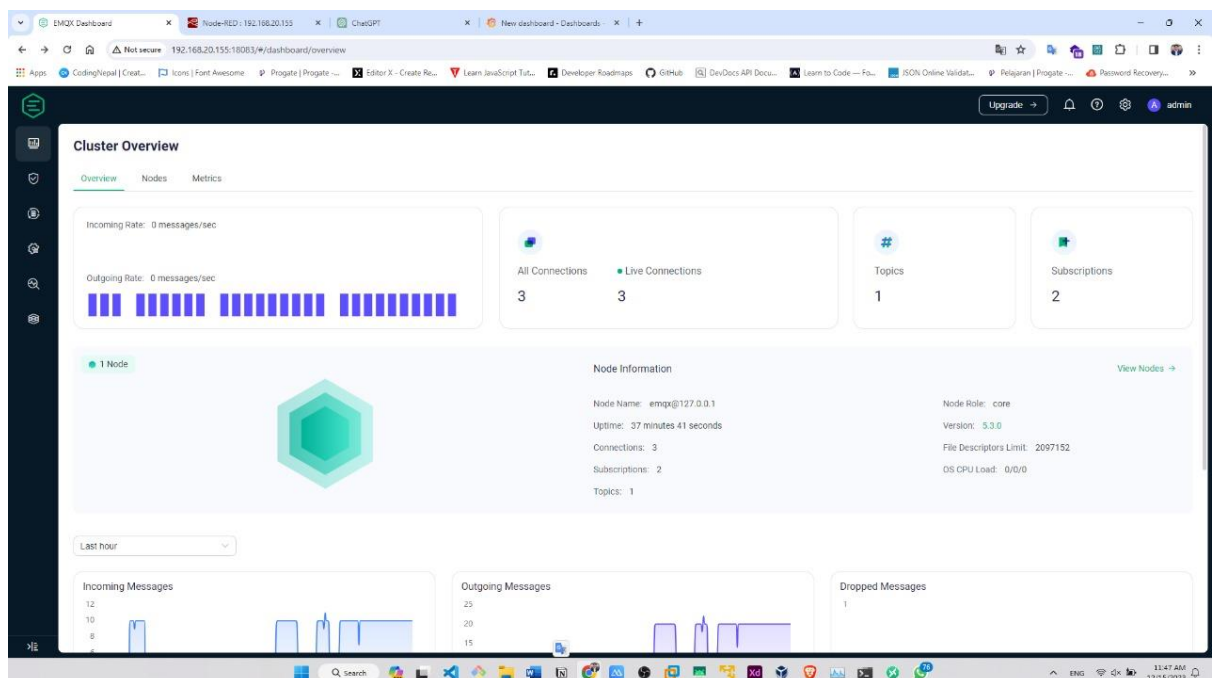


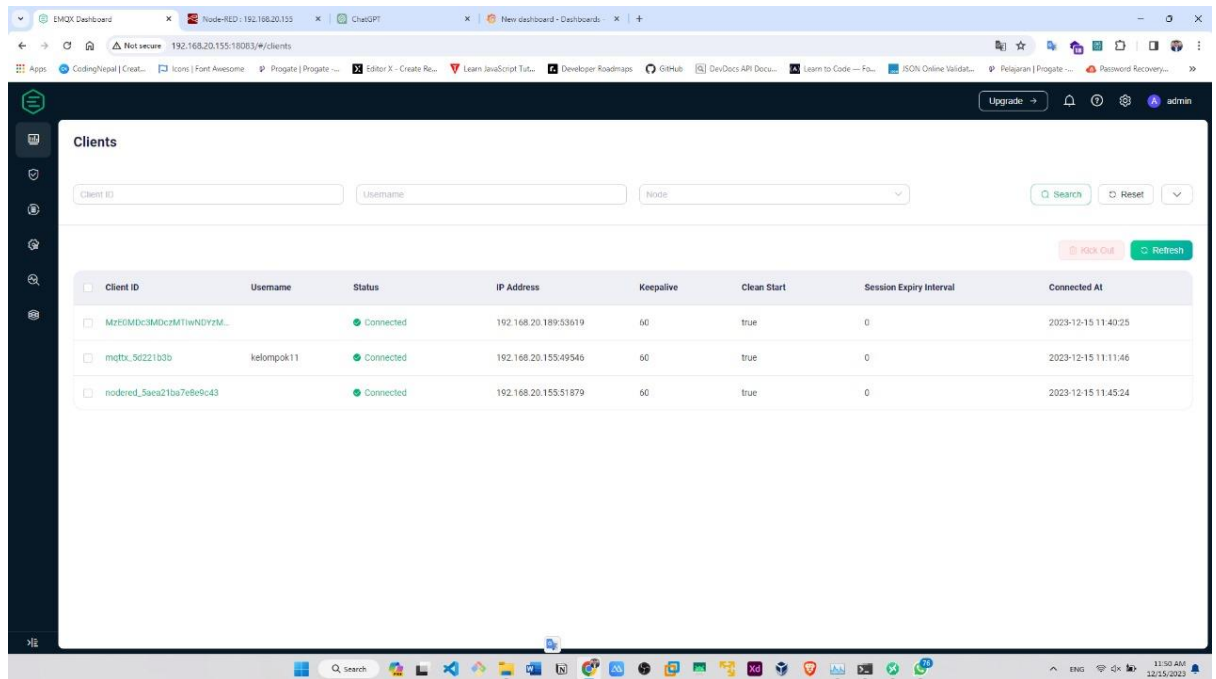
Jika cahaya tidak terdeteksi maka led akan menyala



4. Komunikasi Terhadap EMQX

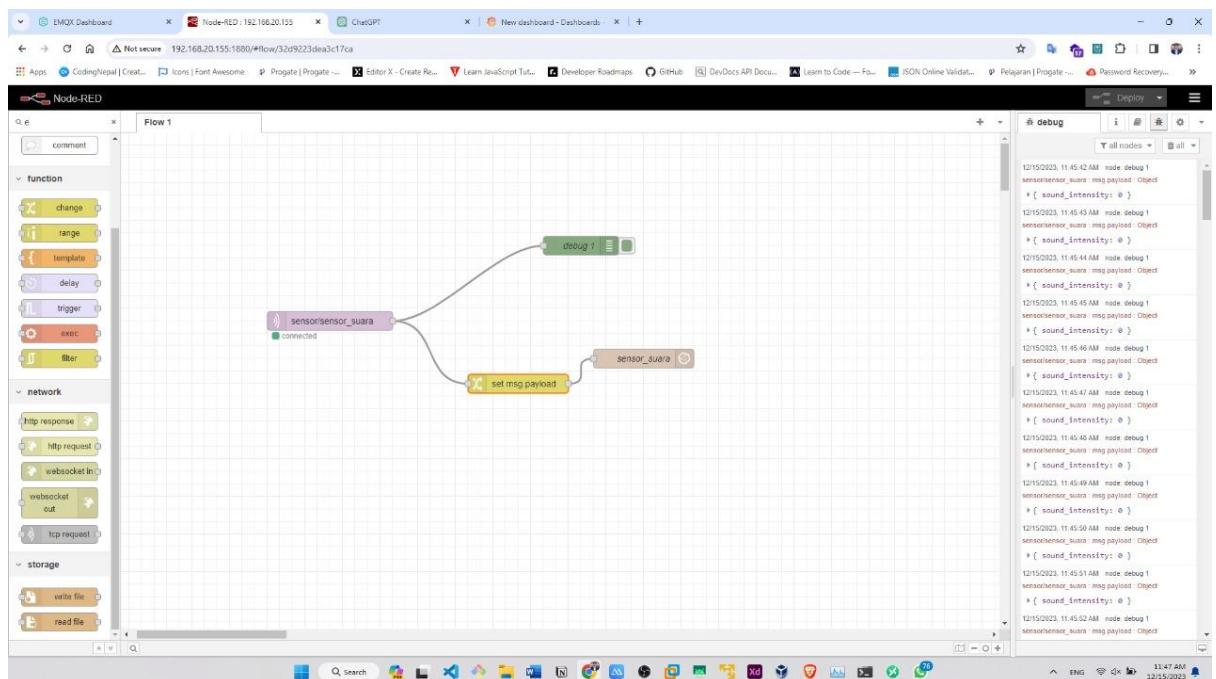
MQTT atau Message Queuing Telemetry Transport adalah suatu protokol komunikasi untuk mengirimkan data di antara perangkat. Dalam proyek otomatisasi lampu dengan memanfaatkan sensor suara, MQTT dapat dipakai untuk menghubungkan sensor suara, Raspberry Pi, dan mungkin perangkat lain yang turut serta dalam proyek tersebut. Apabila ada perubahan dalam tingkat kebisingan atau intensitas suara, sensor suara akan menghasilkan data dalam bentuk analog atau digital yang mencerminkan perubahan tersebut.





5. Node – RED

Node-RED merupakan alat pemrograman visual yang dirancang untuk otomatisasi serta integrasi perangkat dengan bantuan antarmuka grafis yang sederhana untuk mengonfigurasi alur kerja. Pada proyek lampu otomatis dengan menggunakan sensor suara, Node-RED dapat digunakan untuk membuat alur kerja yang menggabungkan sensor suara, Raspberry Pi, dan kontrol output dengan mudah.



6. InfluxDB

InfluxDB dapat digunakan sebagai basis data (database) untuk menyimpan dan menganalisis data dari sensor suara pada proyek lampu otomatis. InfluxDB biasanya digunakan untuk menyimpan data deret waktu (time series data), yang sesuai dengan karakteristik data yang dihasilkan oleh sensor suara yang mengukur intensitas suara seiring waktu.

Jika 1 maka dia terdeteksi, jika 0 maka dia tidak terdeteksi

```
Administrator: C:\Window x + -
1702615757089836100 0
1702615758091214700 0
1702615759224714300 0
1702615760185239600 1
1702615761149441800 0
1702615762123373000 0
1702615763107395800 0
1702615764104757700 0
1702615765098004200 0
1702615766127515800 0
1702615767204347400 0
1702615768174778100 0
1702615769104840400 1
1702615770109717200 1
1702615771116923800 0
1702615772120788600 0
1702615773165313200 0
1702615774213147200 1
1702615775127604100 1
1702615776145008000 0
1702615777124539300 0
1702615778126757600 1
1702615779129807300 0
1702615780192625500 0
1702615781232872800 0
1702615782195884900 0
1702615783144408700 0
1702615784154606200 0
1702615785149021200 0
1702615786149398300 0
>
```

7. Grafana

Grafana adalah alat visualisasi data yang dapat diintegrasikan dengan basis data seperti InfluxDB untuk menyajikan data secara interaktif dan informatif.

Jika hijau artinya terdeteksi, jika merah artinya tidak terdeteksi

