

**Internet of Things**

**Smart Home**

**Lampu Otomatis Menggunakan Sensor  
LDR**



**13321036      Brian Daniel Napitupulu**

**D3 Teknologi Komputer**

**Institut Teknologi Del  
Fakultas Vokasi**

# **SMART HOME**

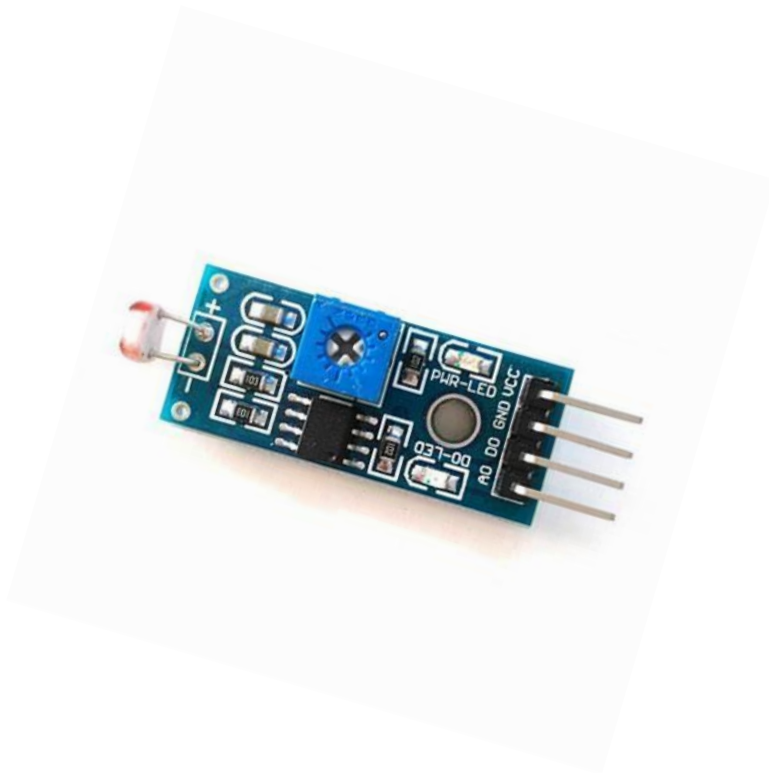
Smart Home adalah teknologi yang meningkatkan kecerdasan dan otomatisasi untuk mengelola berbagai sistem dan perangkat di dalam rumah. Melalui perangkat seluler atau perangkat jaringan, pengguna dapat mengontrol fungsi seperti suhu, pencahayaan, akses keamanan, dan home theater dari jarak jauh. Tujuan sistem ini adalah untuk membuat kehidupan rumah lebih mudah dan nyaman dan membantu mereka menghemat biaya.

Sistem smart home yang telah dibangun dapat diakses melalui satu titik, seperti smartphone, tablet, laptop, kamera, televisi, atau alat lainnya yang dapat melakukan kontrol jarak jauh. Pengguna harus menginstal sistem smart home di perangkat yang akan digunakan untuk melakukan kontrol jarak jauh. Sistem smart home menggunakan artificial intelligence untuk mengetahui jadwal pemilik rumah dan menyesuaikannya sesuai kebutuhan. Pengguna dapat membuat jadwal dan waktu agar perubahan tertentu dilakukan pada sistem sehingga tetap up to date.

## **LAMPU OTOMATIS MENGGUNAKAN SENSOR LDR**

Sensor LDR (Light Dependent Resistant) adalah suatu jenis resistor yang nilai resistansinya berubah-ubah karena adanya intensitas cahaya yang diserap. Karakteristik sensor LDR adalah LDR akan berubah resistansinya/tahanannya ketika terjadi perubahan cahaya yang dideteksinya. LDR terbuat dari Cadmium Sulfrida yaitu bahan semi konduktor yang resistansinya berubah-ubah menurut banyaknya cahaya (sinar) yang mengenainya.

Pada saat gelap atau cahaya redup, bahan dari cakram LDR menghasilkan electron bebas dengan jumlah yang relative kecil. Sehingga hanya sedikit electron untuk mengangkut muatan elektrik. Yang berarti pada saat cahaya redup LDR menjadi pengantar arus kurang baik, atau bisa disebut juga LDR memiliki resistansi yang besar pada saat gelap atau cahaya redup. Pada saat cahaya terang, ada lebih banyak eletron untuk mengangkut muatan elektrik. Artinya pada saat cahaya terang LDR menjadi kondukt atau bisa disebut juga LDR memiliki resistansi yang kecil pada saat cahaya terang.



## CARA KERJA

Hal pertama yang harus dilakukan adalah dengan merangkai alat dan memastikan alat sudah dapat bekerja. Kemudian menjalankan EMQX, NODE-RED, InfluxDB, Grafana, MQTTX.

## EMQX

```
Select C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

D:\Apk\EMQX\bin>emqx start

EMQX_NODE__DB_ROLE [node.role]: core
EMQX_NODE__DB_BACKEND [node.db_backend]: mnesia

D:\Apk\EMQX>
```

# NODE-RED

```
node-red
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>NODE-RED
18 Dec 21:33:54 - [info]

Welcome to Node-RED
=====

18 Dec 21:33:54 - [info] Node-RED version: v3.1.0
18 Dec 21:33:54 - [info] Node.js version: v18.18.0
18 Dec 21:33:54 - [info] Windows_NT 10.0.19045 x64 LE
18 Dec 21:33:59 - [info] Loading palette nodes
18 Dec 21:34:12 - [info] Dashboard version 3.6.0 started at /ui
18 Dec 21:34:14 - [info] Settings file : C:\Users\LENOVO\.node-red\settings.js
18 Dec 21:34:14 - [info] Context store : 'default' [module=memory]
18 Dec 21:34:14 - [info] User directory : \Users\LENOVO\.node-red
18 Dec 21:34:14 - [warn] Projects disabled : editorTheme.projects.enabled=false
18 Dec 21:34:14 - [info] Flows file : \Users\LENOVO\.node-red\flows.json
18 Dec 21:34:14 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
```

# InfluxDb

```
C:\Windows\System32\cmd.exe - influxd
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

D:\Apk\InfluxDB\influxdb-1.7.11>influxd

88888888      .d888 888      88888888b. 8888888b.
888      d88P" 888      888  "Y88b 888  "88b
888      888  888      888  888 888  .88P
888 888888b. 8888888 888 888 888 888 888 88888888K.
888 888 "88b 888 888 888 888 Y8dd8P' 888 888 888 "Y88b
888 888 888 888 888 888 888 X88K 888 888 888 888
888 888 888 888 888 Y88b 888 .d8""8b. 888 .d88P 888 d88P
88888888 888 888 888 888 "Y88888 888 888 8888888P" 88888888P"

2023-12-18T14:33:11.107436Z info InfluxDB starting {"log_id": "0mBLMxl0000", "version": "1.7.11", "branch": "1.7.11", "commit": "f11ad4780c8a61108108a18b141c1d067d920a80"}
2023-12-18T14:33:11.107436Z info Go runtime {"log_id": "0mBLMxl0000", "version": "go1.13.8", "maxprocs": 12}
2023-12-18T14:33:11.225459Z info Using data dir {"log_id": "0mBLMxl0000", "service": "store", "path": "C:\\Users\\LENOVO\\.influxdb\\data"}
2023-12-18T14:33:11.226459Z info Compaction settings {"log_id": "0mBLMxl0000", "service": "store", "max_concurrent_compactions": 6, "throughput_bytes_per_second": 50331648, "throughput_bytes_per_second_burst": 50331648}
2023-12-18T14:33:11.227460Z info Open store (start) {"log_id": "0mBLMxl0000", "service": "store", "trace_id": "0mBLMxl000", "op_name": "tsdb_open", "op_event": "start"}
2023-12-18T14:33:11.298478Z info Reading file {"log_id": "0mBLMxl0000", "engine": "tsml", "service": "cachelober", "path": "C:\\Users\\LENOVO\\.influxdb\\wal\\_internal\\monitor\\9\\_00001.wal", "size": 3074067}
2023-12-18T14:33:11.309479Z info Opened file {"log_id": "0mBLMxl0000", "engine": "tsml", "service": "filestore", "path": "C:\\Users\\LENOVO\\.influxdb\\data\\_internal\\monitor\\7\\000000001-000000001.tsm", "id": 0, "duration": "2.999ms"}
2023-12-18T14:33:11.309479Z info Opened file {"log_id": "0mBLMxl0000", "engine": "tsml", "service": "filestore"}
```

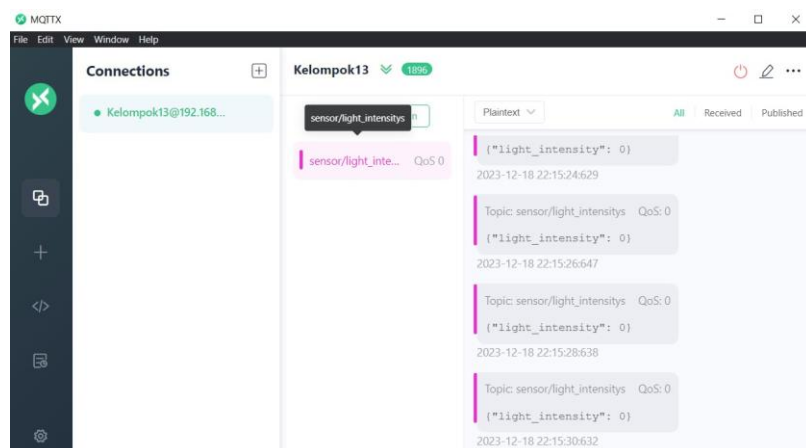
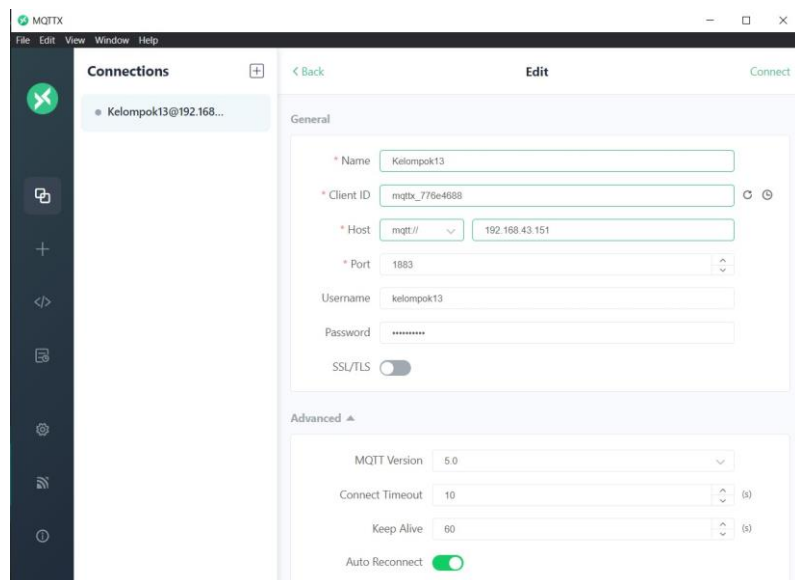
# Grafana

```
C:\Windows\System32\cmd.exe - grafana-server.exe
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

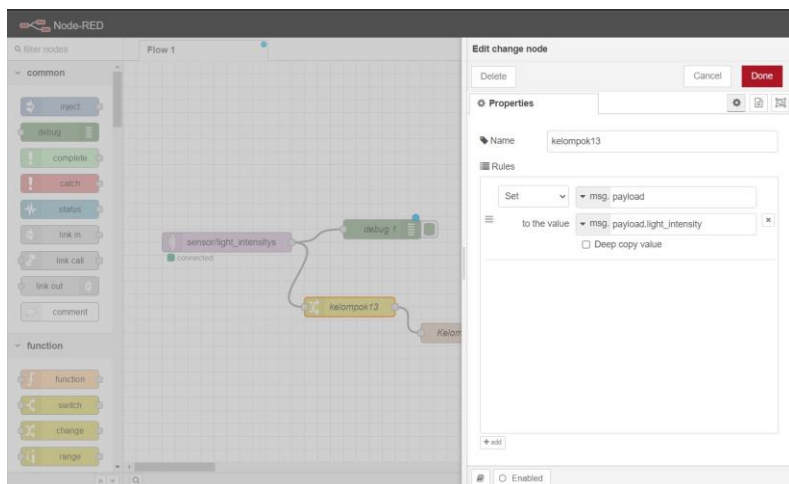
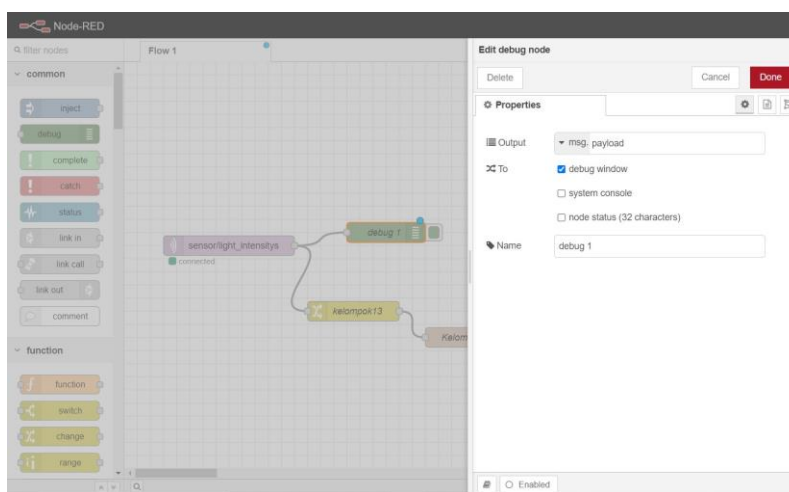
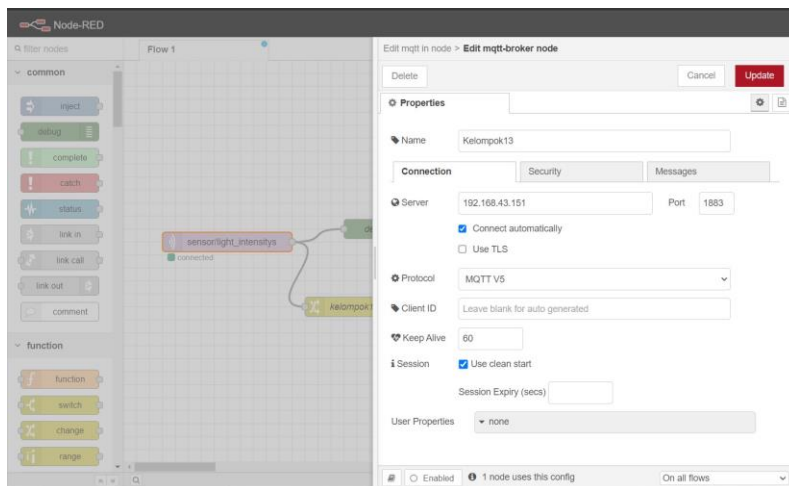
D:\Apk\Grafana\grafana-10.1.2\bin>grafana-server.exe
INFO [12-18|21:34:13] Starting Grafana
INFO [12-18|21:34:13] Target
INFO [12-18|21:34:13] Path Home
INFO [12-18|21:34:13] Path Data
INFO [12-18|21:34:13] Path Logs
INFO [12-18|21:34:13] Path Plugins
INFO [12-18|21:34:13] Path Provisioning
INFO [12-18|21:34:13] App mode production
INFO [12-18|21:34:13] Connecting to DB
WARN [12-18|21:34:13] SQLite database file has broader permissions than it should
INFO [12-18|21:34:13] Starting DB migrations
INFO [12-18|21:34:13] migrations completed
INFO [12-18|21:34:13] Validated license token
WARN [12-18|21:34:13] Envelope encryption state
WARN [12-18|21:34:14] Plugin missing module.js
INFO [12-18|21:34:14] Plugin registered

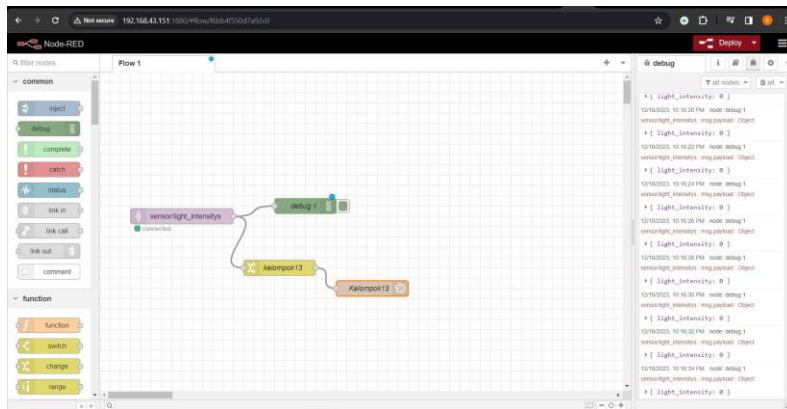
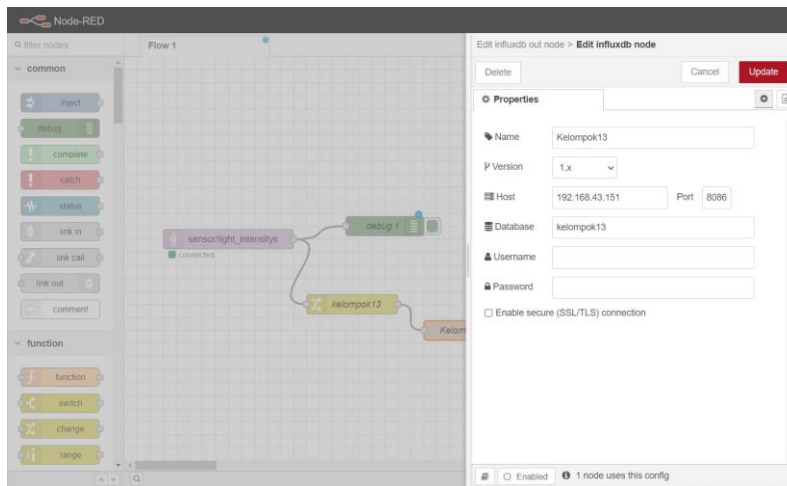
logger=settings version=10.1.2 commit=2b2d40ba32 branch=H
logger=settings file=D:\Apk\Grafana\grafana-10.1.2\con
logger=settings target=[all]
logger=settings path=D:\Apk\Grafana\grafana-10.1.2
logger=settings path=D:\Apk\Grafana\grafana-10.1.2\da
logger=settings path=D:\Apk\Grafana\grafana-10.1.2\da
logger=settings path=D:\Apk\Grafana\grafana-10.1.2\da
logger=settings path=D:\Apk\Grafana\grafana-10.1.2\co
logger=settings
logger=sqlstore dbtype=sqlite3
logger=sqlstore path=D:\Apk\Grafana\
logger=migrator
logger=migrator performed=0 skipped=584 duration=0s
logger=licensing appURL=http://localhost:3000/ source=dis
logger=secrets enabled=true current provider=secretKey.v1
logger=plugin.loader pluginID=input warning="Missing modu
logger=plugin.loader pluginID=input
```

# MQTTX

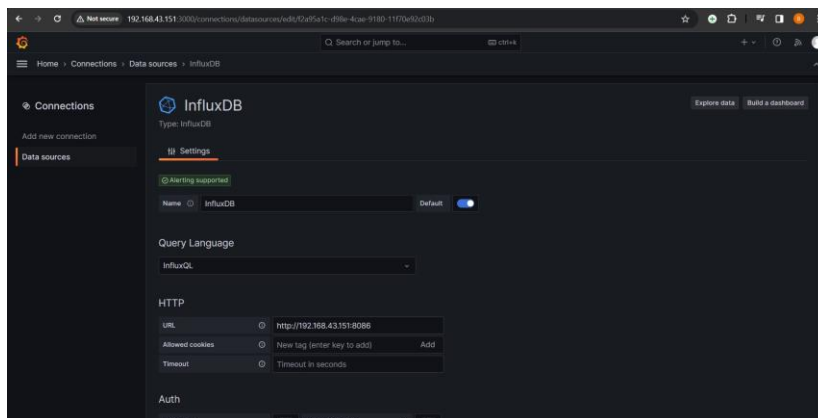


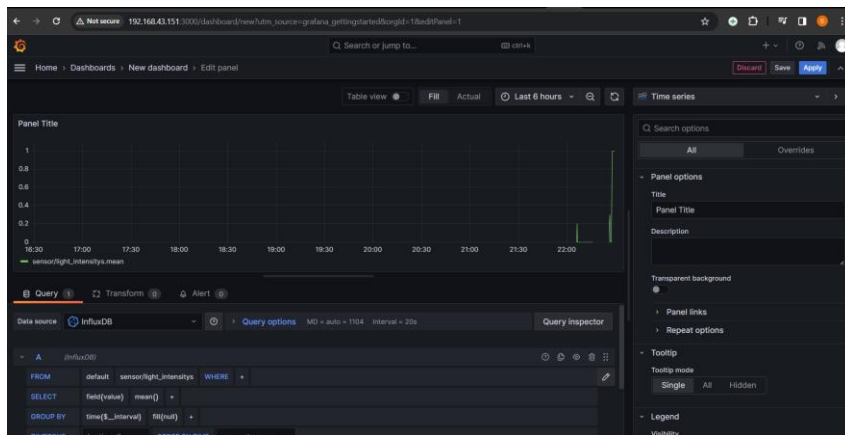
## Melakukan setting pada Node-Red





## Menghubungkan Grafana ke InfluxDb





## Menginstall mosquito

```
kelompok13@raspberrypi:~$ sudo apt install mosquito && mosquito-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gtk2-engines-clearlookspix libcamera-apps libpisp0.0.1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libdlt2 libmosquitto1
The following NEW packages will be installed:
  libdlt2 libmosquitto1 mosquito
0 upgraded, 3 newly installed, 0 to remove and 5 not upgraded.
Need to get 524 kB of archives.
After this operation, 1,573 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bookworm/main arm64 libdlt2 arm64 2.18.8-6 [54.4 kB]
Get:2 http://deb.debian.org/debian-security bookworm-security/main arm64 libmosquitto1 arm64 2.0.11-1.2+deb12u1 [85.9 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security/main arm64 mosquito arm64 2.0.11-1.2+deb12u1 [384 kB]
Fetched 524 kB in 2s (300 kB/s)
Selecting previously unselected package libdlt2:arm64.
(Reading database ... 174167 files and directories currently installed.)
Preparing to unpack .../libdlt2_2.18.8-6_arm64.deb ...
Unpacking libdlt2:arm64 (2.18.8-6) ...
Selecting previously unselected package libmosquitto1:arm64.
Preparing to unpack .../libmosquitto1_2.0.11-1.2+deb12u1_arm64.deb ...
Unpacking libmosquitto1:arm64 (2.0.11-1.2+deb12u1) ...
Selecting previously unselected package mosquito.
Preparing to unpack .../mosquitto_2.0.11-1.2+deb12u1_arm64.deb ...
Unpacking mosquito (2.0.11-1.2+deb12u1) ...
Setting up libmosquitto1:arm64 (2.0.11-1.2+deb12u1) ...
Setting up libdlt2:arm64 (2.18.8-6) ...
Setting up mosquito (2.0.11-1.2+deb12u1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mosquitto.service → /lib/systemd/system/mosquitto.service.
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9+pt2+deb12u3) ...
-bash: mosquito-client: command not found
kelompok13@raspberrypi:~$ sudo apt install mosquito-client
```



# CODE PROGRAM

```
kelompok13@raspberrypi: ~/Scode
GNU nano 7.2                                kelompok_13.py
import RPi.GPIO as GPIO
import paho.mqtt.client as mqtt
import time
import json

# Konfigurasi pin GPIO untuk sensor LDR
ldr_pin = 4 # Sesuaikan dengan pin GPIO yang digunakan
led_pin = 17

# Konfigurasi broker MQTT
mqtt_broker = "192.168.43.151" # Ganti dengan alamat broker MQTT yang Anda gunakan
mqtt_port = 1883
mqtt_topic = "sensor/light_intensitys"

# Fungsi untuk mengirim data ke MQTT dalam format JSON
def send_mqtt_data(client, light_intensity):
    payload = {"light_intensity": light_intensity}
    json_payload = json.dumps(payload) # Convert to JSON format
    client.publish(mqtt_topic, json_payload)

# Fungsi callback ketika terhubung ke broker MQTT
def on_connect(client, userdata, flags, rc):
    print("Terhubung ke broker MQTT dengan kode:", rc)

# Inisialisasi client MQTT
client = mqtt.Client()
client.on_connect = on_connect

# Konfigurasi mode GPIO dan resistansi pull-down untuk sensor LDR
GPIO.setmode(GPIO.BCM)
GPIO.setup(ldr_pin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(led_pin, GPIO.OUT)

# Koneksi ke broker MQTT
client.connect(mqtt_broker, mqtt_port, 60)

# Loop utama
try:
    while True:
        # Baca data dari sensor LDR
        light_intensity = GPIO.input(ldr_pin)

        # Jika data valid, kirim ke MQTT dalam format JSON
        print(f"Intensitas Cahaya: {light_intensity}")
        send_mqtt_data(client, light_intensity)

        # Lampu

        # Lampu
        if light_intensity == 1:
            GPIO.output(led_pin, GPIO.HIGH)
        else:
            GPIO.output(led_pin, GPIO.LOW)

        # Tunggu selama 10 detik sebelum membaca sensor lagi
        time.sleep(2)

except KeyboardInterrupt:
    print("Program dihentikan oleh pengguna.")
finally:
    # Tutup koneksi ke broker MQTT
    client.disconnect()
    # Matikan mode GPIO
    GPIO.cleanup()
```

[illegible]