

LAPORAN TUGAS BESAR BANDUNG WEATHER NEWS

Diajukan Untuk Memenuhi Tugas Mata Sistem Paralel dan Terdistribusi

Dosen Pengampu: Ikke Dian Oktaviani



Disusun Oleh :

Ahmad Julius Tarigan	(1301190345)
Fadlurahman Akbar Nasution	(1301194258)
Iqbal Saviola	(1301194288)
Kasyfi Zulkaisi Aufar	(1301194397)

**PROGRAM STUDI INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
2021/2022**

1. PENDAHULUAN

A. Penjelasan singkat tentang tugas besar yang dibuat

Pada tugas besar sistem paralel dan terdistribusi kami mengambil topik **“Bandung Weather News”**, dimana kami membuat program mengenai simulasi pengukuran suhu dari 3 sensor suhu pada kota Bandung, yang nantinya akan mengirimkan suhu ke server dalam rentang waktu setiap 10 detik. Subscriber menghitung dan menampilkan rata-rata suhu setiap kali menerima informasi. Rata-rata suhu yang akan dihitung akan bergantung dari jumlah sensor yang terdeteksi, jika data yang masuk masih dari 2 sensor maka pembagiannya 2, sedangkan jika data yang masuk 3 sensor maka pembagiannya menjadi 3.

B. Peran Anggota

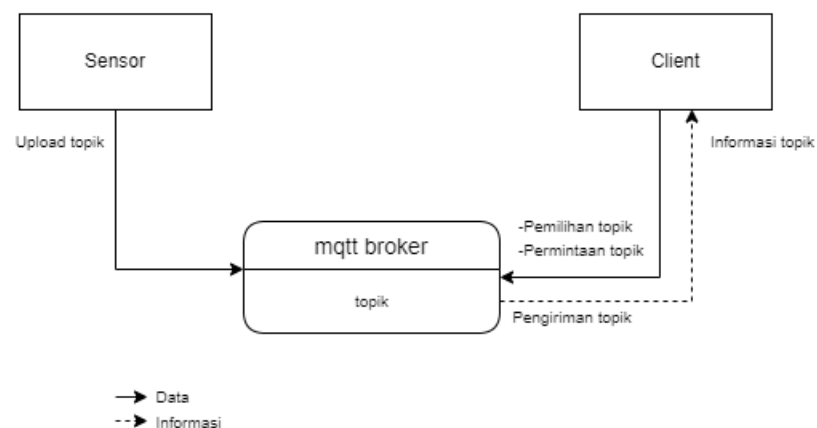
- Ahmad Julius Tarigan :
 - Mengerjakan Laporan
 - Mengerjakan Source Code Publisher dan Subscriber
- Fadlurahman Akbar Nasution :
 - Mengerjakan Laporan
 - Membuat Arsitektur Sistem dan Jaringan
- Iqbal Saviola:
 - Mengerjakan Laporan
 - Membuat Alur Proses Aplikasi
- Kasyfi Zulkaisi Aufar:
 - Mengerjakan Laporan
 - Membuat Model Sistem

2. ANALISIS

A. Alasan Pemilihan Solusi

Kami metode publish-subscribe sebagai solusi dari topik “Bandung Weather News”. Publish-Subscribe merupakan sistem yang tepat untuk topik “Bandung Weather News”, dimana deskripsi untuk topik tersebut melibatkan server (subscriber) dan sensor (Publisher). Publish-Subscribe merupakan sebuah sistem dimana publisher menggunakan service event to event dan subscriber bergabung dalam jaringan pada sebagian event melewati proses subscription. dengan menggunakan metode publish-subscribe sensor mengirimkan pesan berupa suhu kepada server setiap 10 detik. dan server akan menampilkan rata-rata suhu dari setiap informasi pada setiap sensor.

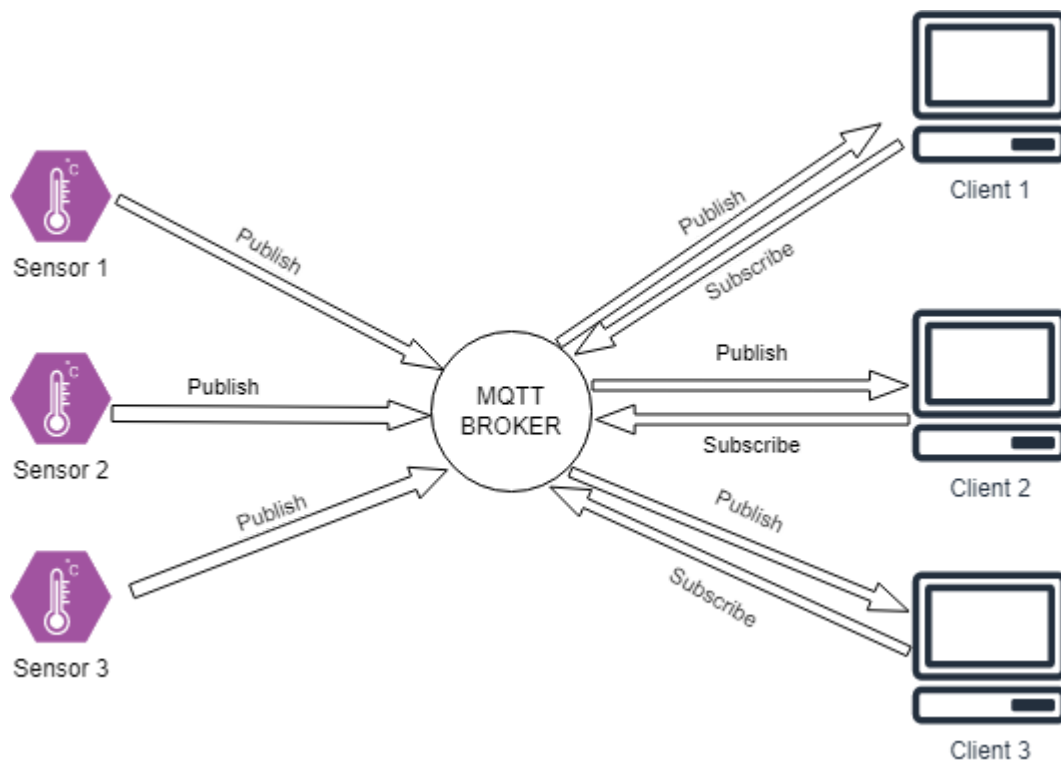
B. Model Sistem



Gambar diatas merupakan model sistem mengenai pengontrolan suhu, dimana kami menggunakan sensor dan client, dan juga kami menggunakan MQTT broker yang berfungsi untuk mengatur data yang dikirimkan pada server dan yang akan diminta oleh client, sebagai gambaran umum sistem Bandung Weather News ini akan dilakukan pengiriman topik oleh sensor berupa data kepada MQTT broker, pada MQTT broker akan menyimpan informasi topik atau data yang telah dikirimkan sensor. informasi ini diantaranya suhu, kelembaban, dan lain-lain. Kemudian client akan melakukan permintaan topik dan pemilihan topik yang sudah berada pada mqtt broker, lalu mqtt akan mengirimkan sebuah informasi berisikan topik yang sudah dipilih client.

3. PERANCANGAN

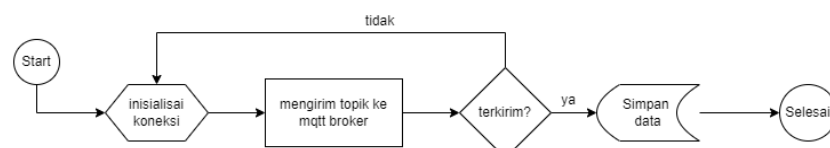
A. Arsitektur Sistem dan Jaringan



Disini sensor akan melakukan publish yang akan mengirimkan topik datanya kepada subscriber melalui MQTT Broker yang menjadi penghubung antara publisher dan subscriber sebagai penanggung jawab terkirimnya semua pesan termasuk jalur distribusinya. Client akan melakukan subscribe kepada topik data yang diinginkan dan broker akan mengirimkan message/data yang sesuai dengan topik yang diinginkan oleh Client.

B. Alur Proses Aplikasi

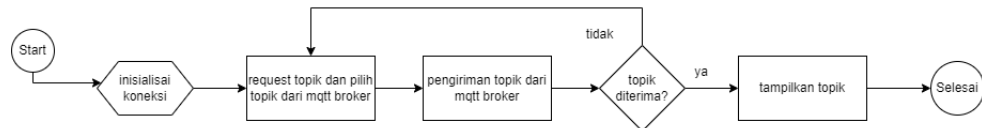
- Sensor



Alur pada Sensor dimulai dengan inisialisasi koneksi, berupa menjalankan sistem, kemudian akan melakukan proses berupa mengirimkan topik (data)

kepada mqtt broker, lalu akan di cek apakah data terkirim atau tidak, jika terkirim, maka data akan tersimpan pada mqtt broker, jika tidak terkirim maka proses akan diulang.

- Client



Seperti alur proses pada client, dimulai dengan inisialisasi koneksi dengan menjalankan sistem, kemudian client akan meminta topik dan memilih topik yang sudah tersimpan pada mqtt server, dilanjutkan dengan proses pengiriman yang dilakukan mqtt server, dan akan melakukan seleksi apakah topik sudah diterima client, jika sudah, akan ditampilkan berupa informasi berisikan topik yang telah diminta pada client.

3. IMPLEMENTASI

A. Source Code

a. Publisher 1 (Sensor 1)

```
###Sensor 1

#import paho.mqtt.client
import paho.mqtt.client as mqtt
#import random to generate random value
from random import uniform
#import time to sleep/delay
import time

#set public mqtt broker
mqttBroker = 'mqtt.eclipseprojects.io'
#set client name
sensor1 = mqtt.Client("Sensor 1")
#connect to public broker
sensor1.connect(mqttBroker)
#always run the sensor
sensor1.loop_start()
while True:
    #generate random number as a temperature
    randNumber = uniform(25.0, 30.0)
    #publish a topic to broker
    sensor1.publish("Sensor 1", round(randNumber,2))
    #print output
    print("Suhu: " + str(round(randNumber,2))+ " derajat celcius")
    #delay 10 second
    time.sleep(10)
```

b. Publisher 2 (Sensor 2)

```
####Sensor 2

#import paho.mqtt.client
import paho.mqtt.client as mqtt
#import random to generate random value
from random import uniform
#import time to sleep/delay
import time

#set public mqtt broker
mqttBroker = 'mqtt.eclipseprojects.io'
#set client name
sensor2 = mqtt.Client("Sensor 2")
#connect to public broker
sensor2.connect(mqttBroker)
#always run the sensor
sensor2.loop_start()
while True:
    #generate random number as a temperature
    randNumber = uniform(20.0, 25.0)
    #publish a topic to broker
    sensor2.publish("Sensor 2", round(randNumber,2))
    #print output
    print("Suhu: "+ str(round(randNumber,2))+ " derajat celcius")
    #delay 10 second
    time.sleep(10)
```

c. Publisher 3 (Sensor 3)

```
####Sensor 3

#import paho.mqtt.client
import paho.mqtt.client as mqtt
#import random to generate random value
from random import uniform
#import time to sleep/delay
import time

#set public mqtt broker
mqttBroker = 'mqtt.eclipseprojects.io'
#set client name
sensor3 = mqtt.Client("Sensor 3")
#connect to public broker
sensor3.connect(mqttBroker)
#always run the sensor
sensor3.loop_start()
while True:
    #generate random number as a temperature
    randNumber = uniform(22.0, 27.0)
    #publish a topic to broker
    sensor3.publish("Sensor 3", round(randNumber,2))
    #print output
    print("Suhu: "+ str(round(randNumber,2))+ " derajat celcius")
    #delay 10 second
    time.sleep(10)
```

d. Subscriber (Client)

```
###Client

#import paho.mqtt.client
import paho.mqtt.client as mqtt
#import time to sleep/delay
import time

#set empty list to save current temperature
val=[]
#on_message function
def on_message(client, userdata, message):
    #get the message and cast to float64
    msg = float(message.payload.decode("utf-8"))
    #print output
    print("Received Message: ", round(msg,2), " From ", message.topic)
    #append message(temperature) to list
    val.append(msg)
    #set list empty when have 3 value
    if len(val)>3:
        del val[:]
        val.append(msg)
    #get the average by divided with number of message on list
    avg = round(sum(val)/len(val),2)
    #output the list
    print(val)
    #output the average
    print("Average:", avg)

#set public mqtt broker
mqttBroker = 'mqtt.eclipseprojects.io'
#set client name
client = mqtt.Client("Client1")
#set function on_message
client.on_message = on_message
#connect to public broker
client.connect(mqttBroker)

#always run the client
client.loop_start()
while True:
    #delay 1 second
    time.sleep(1)
    #subscribe to sensor 1, sensor 2, sensor 3 with QOS=2
    client.subscribe([("Sensor 1", 2), ("Sensor 2", 2), ("Sensor 3", 2)])
```

B. Ouput

Sensor 1:

```
Suhu: 27.81 derajat celcius  
Suhu: 25.4 derajat celcius  
Suhu: 25.14 derajat celcius  
Suhu: 26.28 derajat celcius
```

Sensor 2:

```
Suhu: 21.15 derajat celcius  
Suhu: 22.05 derajat celcius  
Suhu: 21.08 derajat celcius  
Suhu: 22.2 derajat celcius
```

Sensor 3:

```
Suhu: 23.16 derajat celcius  
Suhu: 24.03 derajat celcius  
Suhu: 26.64 derajat celcius  
Suhu: 22.35 derajat celcius  
Suhu: 25.18 derajat celcius
```

Client:

```
Received Message: 25.4 From Sensor 1  
[25.4]  
Average: 25.4  
Received Message: 22.05 From Sensor 2  
[25.4, 22.05]  
Average: 23.73  
Received Message: 24.03 From Sensor 3  
[25.4, 22.05, 24.03]  
Average: 23.83  
Received Message: 25.14 From Sensor 1  
[25.14]  
Average: 25.14  
Received Message: 21.08 From Sensor 2  
[25.14, 21.08]  
Average: 23.11  
Received Message: 26.64 From Sensor 3  
[25.14, 21.08, 26.64]  
Average: 24.29  
Received Message: 26.28 From Sensor 1  
[26.28]  
Average: 26.28  
Received Message: 22.2 From Sensor 2  
[26.28, 22.2]  
Average: 24.24  
Received Message: 22.35 From Sensor 3  
[26.28, 22.2, 22.35]  
Average: 23.61  
Received Message: 25.18 From Sensor 3  
[25.18]  
Average: 25.18
```


C. Keterbatasan/permasalahan

Karena kami menggunakan public broker sehingga terkadang koneksi antara publisher dan subscriber terkendala. Solusi dari masalah ini adalah dengan membuat broker sendiri dengan cara membayar atau membuat server sendiri. Selain itu kami tidak menggunakan sensor asli melainkan hanya melakukan generate random number yang dianggap sebagai derajat suhu.

D. Link Penting

Video Demo :

https://drive.google.com/file/d/1YMQ6Nt_rNMcv6rU41Gif_O1twWQ2HTvg/view?usp=sharing

Repository: <https://github.com/Ahmad20/Publish-Subscribe>