

**Tugas Besar Sistem Paralel dan Terdistribusi  
“Laundry”**



Dipersiapkan oleh  
Kelompok 3 - Maung Bandung :

Shidqi Fadhlurrahman Yusri	1301204256
Fikri Maulana	1301200239
Rangga Eka Permana	1301200307
Adhyasta Naufal Fadhilah	1301200321
Inggit Restu Illahi	1301204446

**Program Studi S1 Informatika – Fakultas Informatika**

**Universitas Telkom**

**Jalan Telekomunikasi Terusan Buah Batu, Bandung**

**Indonesia**

# Pendahuluan

Sistem terdistribusi adalah kumpulan komputer otonom yang dihubungkan oleh jaringan dengan software yang dirancang untuk menghasilkan fasilitas komputerisasi terintegrasi dianggap oleh pengguna sebagai satu sistem komputer tunggal. Salah satu sistem yang kami gunakan pada tugas ini adalah sistem *Publish and Subscribe*, yang mana pada sistem ini terdapat dua jenis client yaitu *Publisher* dan *Subscriber* dan keduanya terhubung pada satu server yang sama yang disebut *Broker*. Kedua client tersebut akan terhubung pada satu *Broker* yang sama dan disanalah kedua client tersebut akan saling berkomunikasi.

Tugas ini bertujuan untuk mengetahui kemampuan kita dalam mengerjakan sebuah *case* yang diberikan kepada kita. Metodologi yang digunakan dalam tugas ini adalah dengan mengerjakan sebuah topik yang dipilih seperti laundry dengan memilih program MQTT.

Pada akhir tugas ini, diharapkan dapat menghubungkan sebuah sistem terdistribusi antara client dan laundry menggunakan bahasa pemrograman python. Kemudian setelah berhasil terhubung, message antara *client* dan laundry dapat terlihat di *console* melalui broker sebagai *server*.

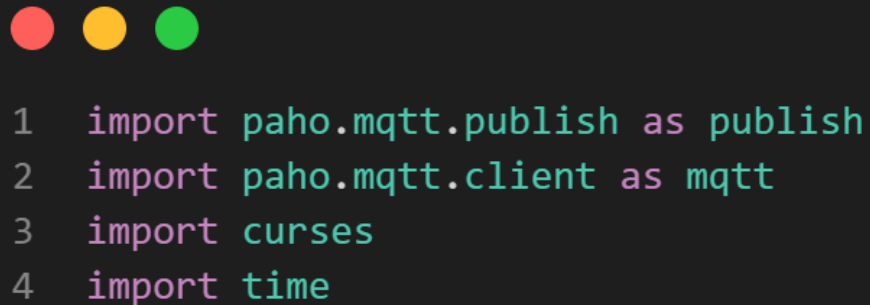
## Rumusan Masalah

Membuat program agar “Laundry Bojongpura” dan “Laundry Soang” dapat mengirim dan menerima pesanan laundry dari banyak client. Kedua laundry tersebut akan mengirimkan informasi ke client kapan penjemputan baju kotor dan kapan perkiraan pengantaran baju hasil laundry. Client harus follow laundry terlebih dahulu agar mendapatkan informasi dari laundry tersebut. Client dapat melihat laundry mana yang memberikan waktu selesai laundry paling cepat, kemudian memilih akan memesan ke laundry yang mana. Setiap laundry hanya mengirimkan kepada client pelanggan saja, dan client hanya dapat menerima informasi dari laundry yang di-follow saja.

1. Bagaimana sistem MQTT bekerja?
2. Bagaimana agar subscriber dapat menerima pesan dari publisher?
3. Seberapa efektifkah penggunaan MQTT?

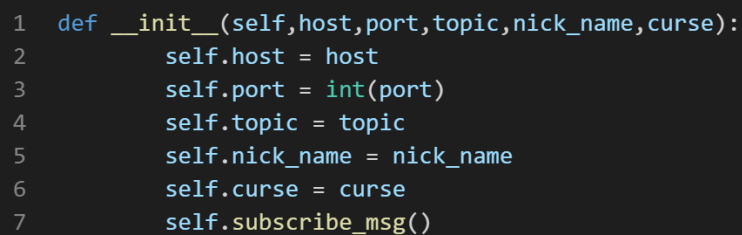
# Pembahasan

1. Disini kami menggunakan 4 package dalam python



```
1 import paho.mqtt.publish as publish
2 import paho.mqtt.client as mqtt
3 import curses
4 import time
```

- a. paho.mqtt.publish = package mqtt untuk mempublish pesan.
  - b. paho.mqtt.client = package mqtt untuk client agar bisa menerima pesan dari publisher pada topic yang sama.
  - c. curses = package untuk membuat bentuk dari aplikasi agar lebih nyaman dilihat.
  - d. time = package yang digunakan untuk memberi keterangan waktu mengirim pesan.
2. class mqtt\_chat
- Ini merupakan class yang mengurus bagian mqtt
- a. def \_\_init\_\_



```
1 def __init__(self,host,port,topic,nick_name,curse):
2     self.host = host
3     self.port = int(port)
4     self.topic = topic
5     self.nick_name = nick_name
6     self.curse = curse
7     self.subscribe_msg()
```

Constructor dari class mqtt\_chat agar variabel yang ada di dalamnya bisa digunakan pada method yang ada setelah constructor.

b. def subscribe\_msg

```
1 def subscribe_msg(self):
2     self.subscriber = mqtt.Client()
3     self.subscriber.on_connect = self.on_connect
4     self.subscriber.on_message = self.on_message
5     self.is_connect = False
6     self.subscriber.connect(self.host,self.port);
7     self.subscriber.loop_start()
8     while self.is_connect == False:
9         pass
```

Method yang digunakan untuk mensubscribe message agar dapat menerima message dari publisher.

c. def send\_msg

```
1 def send_msg(self,msg):
2     waktu = time.localtime()
3     msg = time.strftime("%H:%M:%S", waktu)+" [" +self.nick_name + "] : "+msg
4     publish.single(self.topic,msg, hostname=self.host, port=self.port)
```

Method yang digunakan untuk mengirim pesan kepada subscriber yang berisi waktu pengiriman pesan dan isi pesan.

d. def on\_connect

```
1 def on_connect(self,client, userdata, flags, rc):
2     self.is_connect = True
3     client.subscribe(self.topic);
4     if self.topic == "bojongpura":
5         self.send_msg(self.nick_name+" terhubung dengan Bojongpura Laundry\n")
6     elif self.topic == "soang":
7         self.send_msg(self.nick_name+" terhubung dengan Laundry Soang\n")
```

Method yang digunakan untuk mensubscribe suatu topic yang dipilih, yang mana disini ada 2 topik yaitu laundry soang dan bojongpura laundry.

e. on\_message

```
1 def on_message(self,client,user_data,msg):
2     self.curse.draw_received_msg(msg.payload.decode('utf-8'))
```

Method yang menerima message dan membuat grafik interface menggunakan curse.

### 3. class my\_courses

Class yang mengurus bagian grafik interface.

#### a. def \_\_init\_\_

```
1 def __init__(self):
2     self.stdscr = curses.initscr()
3     self.show_row = self.show_col = 0
4     self.input_col = 0
5     (self.max_row, self.max_col) = self.stdscr.getmaxyx()
6     curses.cbreak()
```

Constructor dari class my\_courses agar variabel yang sudah disediakan pada def \_\_init\_\_ bisa digunakan pada method yang ada setelah constructor.

#### b. def draw\_text

```
1 def draw_text(self, row, col, msg):
2     self.stdscr.addstr(row, col, ' '*(self.max_col-1))
3     self.stdscr.addstr(row, col, msg)
4     self.stdscr.refresh()
```

Method dasar untuk membuat grafik interface.

#### c. def get\_input

```
1 def get_input(self):
2     user_input = ''
3     while True:
4         c = self.stdscr.get_wch()
5         if c == '\n':
6             self.input_col=2
7             break
8         elif c == curses.KEY_BACKSPACE:
9             print('!')
10            pass
11        else:
12            user_input+=str(c)
13            self.input_col+=1
14        return user_input
```

Method untuk mengisi inputan untuk pesan yang ingin dikirimkan dan inputan nama serta laundry yang ingin dipilih.

d. def draw\_start\_window

```
1 def draw_start_window(self):
2     self.stdscr.clear()
3     self.draw_text(1,0,"#"*self.max_col-1))
4     self.draw_text(2,0,"Masukkan keyword dari laundry yang tersedia : ")
5     self.draw_text(3,0,"Pilihan Laundry yang tersedia : ")
6     self.draw_text(4,0,"1. Bojongpura Laundry : bojongpura")
7     self.draw_text(5,0,"2. Laundry Soang : soang")
```

Method untuk membuat grafik interface pada menu utama sebelum masuk kedalam chatroom.

e. def draw\_main\_window\_bojongpura

```
1 def draw_main_window_bojongpura(self):
2     self.stdscr.clear()
3     self.draw_text(1,0,"#"*self.max_col-1))
4     self.draw_text(2,0,"Laundry Bojongpura");
5     self.draw_text(3,0,"Masukkan tanggal pengiriman jika anda ingin informasi selesainya laundry anda!");
6     self.draw_text(4,0,"Anda juga bisa komplain pada aplikasi chat ini");
7     self.draw_text(5,0,"#"*self.max_col-1))
8     row = self.max_row-2
9     self.draw_text(row,0,"-"*self.max_col-1))
10    self.draw_user_input_area()
11    self.show_row = 5
```

Method untuk membuat grafik interface ketika topic bojongpura laundry yang dipilih.

f. def draw\_main\_window\_laundry\_soang

```
1 def draw_main_window_laundry_soang(self):
2     self.stdscr.clear()
3     self.draw_text(1,0,"#"*self.max_col-1))
4     self.draw_text(2,0,"Laundry Soang");
5     self.draw_text(3,0,"Masukkan tanggal pengiriman jika anda ingin informasi selesainya laundry anda!");
6     self.draw_text(4,0,"Anda juga bisa komplain pada aplikasi chat ini!");
7     self.draw_text(5,0,"#"*self.max_col-1))
8     row = self.max_row-2
9     self.draw_text(row,0,"-"*self.max_col-1))
10    self.draw_user_input_area()
11    self.show_row = 5
```

Method untuk membuat grafik interface ketika topic soang yang dipilih.

g. def draw\_user\_input\_area

```
1 def draw_user_input_area(self):
2     self.draw_text(self.max_row-1,0,' '*self.max_col-1))
3     self.draw_text(self.max_row-1,0,">>")
```

Method untuk membuat grafik interface pada bagian inputan user saat ingin mengirim pesan.

h. def draw\_received\_msg

```
1 def draw_received_msg(self,msg):
2     self.show_row+=1
3     if self.show_row == self.max_row-2:
4         self.show_row = 4
5     self.draw_text(self.show_row,0,msg)
6     self.stdscr.move(self.max_row-1,self.input_col)
7     self.stdscr.refresh()
```

Method untuk menampilkan pesan yang masuk.

4. if \_\_name\_\_ == "\_\_main\_\_"

Bagian main atau driver yang mengimplementasikan objek yang sudah dibuat sebelumnya. Diawali dengan membuat objek curse agar bisa digunakan pada proses selanjutnya.

```
1 if __name__ == "__main__":
2     curse = my curses()
3     curse.draw_start_window()
4     curse.draw_text(7, 0, 'Masukkan nama anda : \n')
5     nick_name = curse.get_input()
6     curse.draw_text(9, 0, 'Masukkan nama laundry yang ingin anda ketahui tanggal penjemputan dan pengirimannya!\n')
7     topic = curse.get_input()
8     topic = topic.lower()
9     while True:
10         if topic == "bojongpura":
11             curse.draw_main_window_bojongpura()
12             chat = mqtt_chat("test.mosquitto.org", "1883",str(topic),str(nick_name),curse);
13             while True:
14                 msg = curse.get_input()
15                 curse.draw_user_input_area()
16                 chat.send_msg(msg)
17             elif topic == "soang":
18                 curse.draw_main_window_laundry_soang()
19                 chat = mqtt_chat("test.mosquitto.org", "1883",str(topic),str(nick_name),curse);
20                 while True:
21                     msg = curse.get_input()
22                     curse.draw_user_input_area()
23                     chat.send_msg(msg)
24             else:
25                 curse = my curses()
26                 curse.draw_start_window()
27                 curse.draw_text(7, 0, 'Masukkan nama anda : \n')
28                 nick_name = curse.get_input()
29                 curse.draw_text(9, 0, 'Masukkan nama laundry yang ingin anda ketahui tanggal penjemputan dan pengirimannya!\n')
30                 topic = curse.get_input()
```

a. curse.draw\_start\_window

Menampilkan atau membuat menu utama.

b. curse.draw\_text

Menampilkan string perintah untuk memasukkan nama user.

Menampilkan string perintah untuk memasukkan nama laundry yang dipilih.

c. nick\_name = curse.get\_input()

Meminta input untuk nick\_name

d. topic = curse.get\_input

Meminta input untuk topic

e. topic = topic.lower

Membuat string yang memiliki huruf besar agar menjadi huruf kecil semua.



## Bojongpura.py

```
1 def draw_start_window(self):
2     self.stdscr.clear()
3     self.draw_text(1,0,"#"*(self.max_col-1))
4     self.draw_text(2,0,"MQTT antara client dan laundry")
5     self.draw_text(3,0,"#"*(self.max_col-1))
6     def draw_main_window(self):
7         self.stdscr.clear()
8         self.draw_text(1,0,"#"*(self.max_col-1));
9         self.draw_text(2,0,"Bojongpura Laundry Chatroom");
10        self.draw_text(3,0,"#"*(self.max_col-1))
11        row = self.max_row-2
12        self.draw_text(row,0,"-"*(self.max_col-1))
13        self.draw_user_input_area()
14        self.show_row = 3
```

```
1 if __name__ == "__main__":
2     curse = my curses()
3     curse.draw_start_window()
4     nick_name = "Bojongpura Laundry"
5     topic = "bojongpura"
6     chat = mqtt_chat("test.mosquitto.org", "1883", str(topic), str(nick_name), curse);
7     curse.draw_main_window()
8     while True:
9         msg = curse.get_input()
10        curse.draw_user_input_area()
11        chat.send_msg(msg)
```

## LaundrySoang.py

```
1  def draw_start_window(self):
2      self.stdscr.clear()
3      self.draw_text(1,0,"#"*(self.max_col-1))
4      self.draw_text(2,0,"MQTT antara client dan laundry")
5      self.draw_text(3,0,"#"*(self.max_col-1))
6  def draw_main_window(self):
7      self.stdscr.clear()
8      self.draw_text(1,0,"#"*(self.max_col-1));
9      self.draw_text(2,0,"Laundry Soang Chatroom");
10     self.draw_text(3,0,"#"*(self.max_col-1))
11     row = self.max_row-2
12     self.draw_text(row,0,"-"*(self.max_col-1))
13     self.draw_user_input_area()
14     self.show_row = 3
```

```
1  if __name__ == "__main__":
2      curse = my_curses()
3      curse.draw_start_window()
4      nick_name = "Laundry Soang"
5      topic = "soang"
6      chat = mqtt_chat("test.mosquitto.org", "1883", str(topic), str(nick_name), curse);
7      curse.draw_main_window()
8      while True:
9          msg = curse.get_input()
10         curse.draw_user_input_area()
11         chat.send_msg(msg)
```

## Kesimpulan

Dapat disimpulkan bahwa sistem terdistribusi dapat digunakan untuk menghubungkan client dengan laundry dengan menggunakan program MQTT sebagai protokol komunikasi. Dengan sistem terdistribusi ini, Publisher dapat mengirimkan pesan ke Broker yang kemudian disampaikan kepada Subscriber yang telah terdaftar pada Broker tersebut. Subscriber dapat menerima pesan yang dikirim oleh Publisher melalui Broker. Dengan demikian, Publisher dan Subscriber dapat saling berkomunikasi tanpa harus terhubung langsung. Penggunaan MQTT dapat dikatakan efektif karena protokol ini memiliki overhead yang rendah dan memiliki fitur-fitur yang memudahkan komunikasi antara Publisher dan Subscriber. Setelah terhubung, pesan yang dikirim antara Publisher dan Subscriber dapat ditampilkan di console melalui Broker sebagai server.

# Daftar Pustaka

<https://www.contohapps.com/2020/08/pengenalan-sistem-terdistribusi.html>