NAMA : FIDELIA PING
NIM : 245410012
KELAS : INFORMATIKA 1

MODUL 9
IMPLEMENTASI PUB/SUB (BROADCASTING)

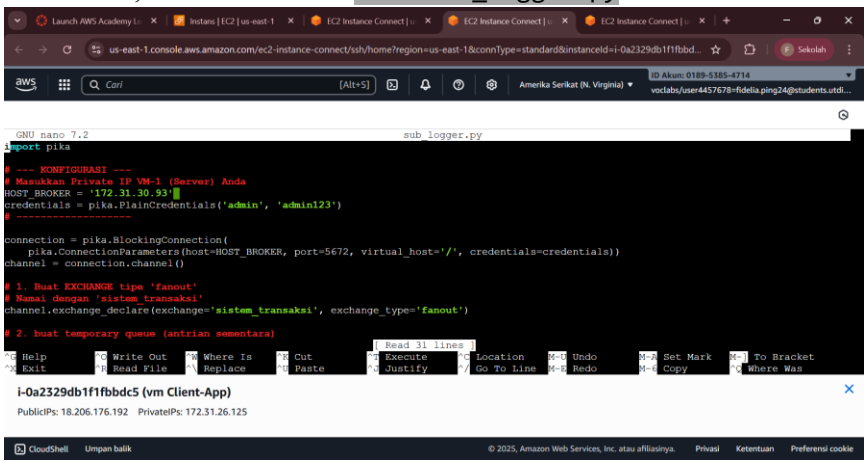
DASAR TEORI

Publish/Subscribe (Pub/Sub) adalah sebuah messaging pattern di mana pengirim pesan (publisher) tidak mengirim pesan langsung kepada penerima tertentu, tetapi mengirimkannya ke sebuah channel, topic, atau exchange. Penerima pesan (subscriber) kemudian menerima pesan dari channel/topic tersebut sesuai dengan langganannya.
Pada model ini, publisher dan subscriber tidak saling mengetahui satu sama lain, sehingga komunikasi menjadi lebih fleksibel, terdesentralisasi, dan skalabel.

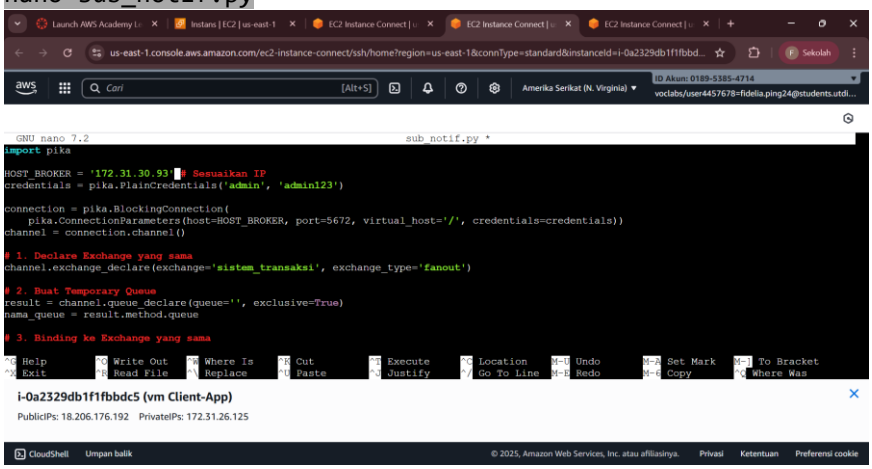
PRAKTIK

BAGIAN 2: MEMBUAT SUBSCRIBER (PENERIMA)

Langkah 1: Buat Script Logger (sub_logger.py)
Di **VM-2**, buat file baru: `nano sub_logger.py`



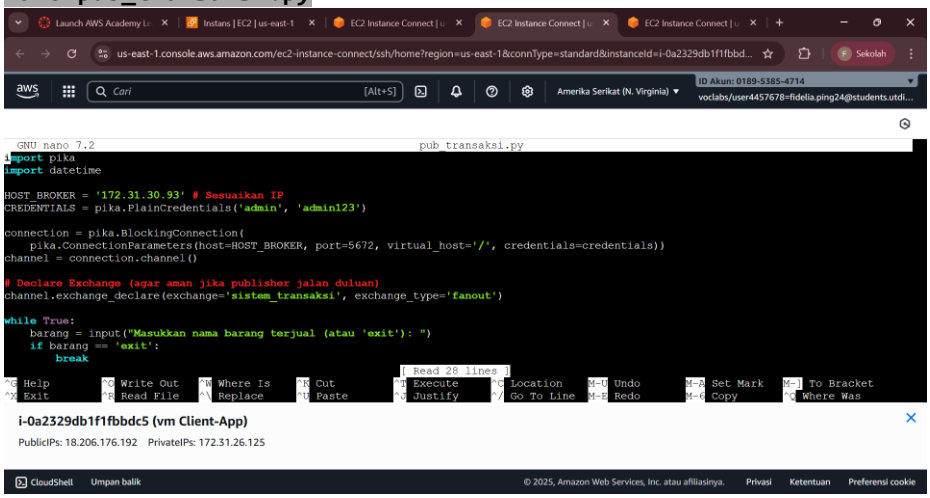
Langkah 2: Buat Script Notifikasi (sub_notif.py)
`nano sub_notif.py`



BAGIAN 3: MEMBUAT PUBLISHER (PENGIRIM)

Langkah 3: Buat Script Transaksi (pub_transaksi.py)

```
nano pub_transaksi.py
```



BAGIAN 4: UJI COBA

1. Terminal 1 (Logger):
   Jalankan: python3 sub_logger.py
   (Akan muncul: Waiting...)

```
ubuntu@ip-172-31-26-125:~$ python3 sub_notif.py
 [*] SISTEM NOTIFIKASI siap. Menunggu alert...
```

i-0a2329db1f1fbbdc5 (vm Client-App)

PublicIPs: 18.206.176.192 PrivateIPs: 172.31.26.125

2. Terminal 2 (Notif):
   Jalankan: python3 sub_notif.py
   (Akan muncul: Waiting...)

```
Last login: Thu Dec  4 08:16:04 2025 from 18.206.107.29
ubuntu@ip-172-31-26-125:~$ python3 sub_logger.py
 [*] SISTEM LOGGER siap. Menunggu data transaksi...
```

i-0a2329db1f1fbbdc5 (vm Client-App)

PublicIPs: 18.206.176.192 PrivateIPs: 172.31.26.125

3. Terminal 3 (Publisher):
   Jalankan: python3 pub_transaksi.py
   - Ketik: Laptop Gaming -> Enter.
   - Ketik: Mouse Wireless -> Enter.

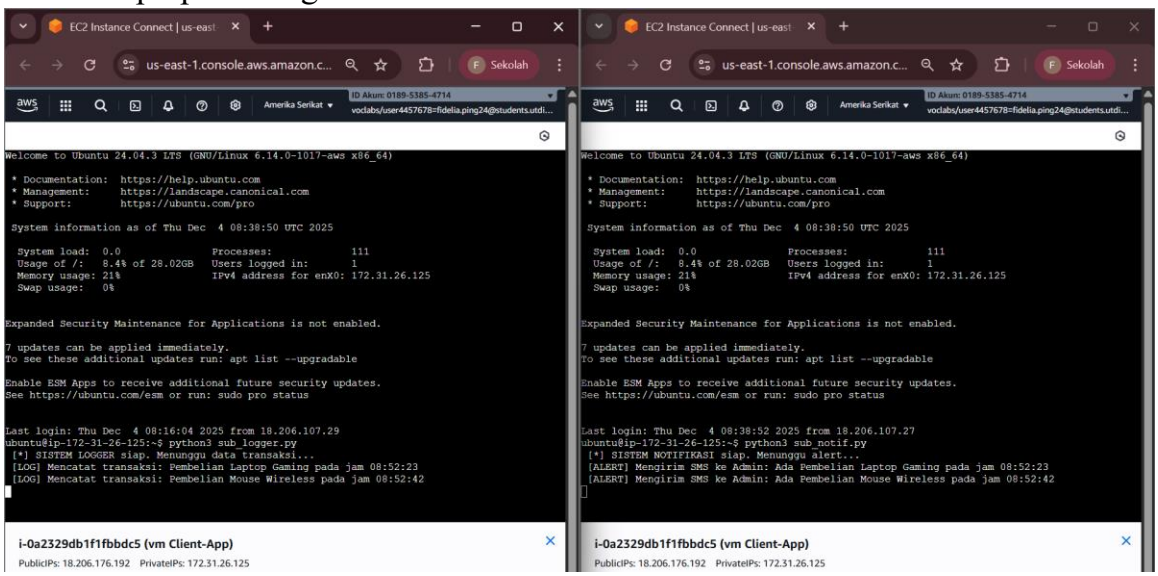
```
ubuntu@ip-172-31-26-125:~$ python3 pub_transaksi.py
Masukkan nama barang terjual (atau 'exit'): Laptop Gaming
 [x] Data terkirim ke sentral: 'Laptop Gaming'
Masukkan nama barang terjual (atau 'exit'): Mouse Wireless
 [x] Data terkirim ke sentral: 'Mouse Wireless'
Masukkan nama barang terjual (atau 'exit'):
```

i-0a2329db1f1fbbdc5 (vm Client-App)

PublicIPs: 18.206.176.192 PrivateIPs: 172.31.26.125

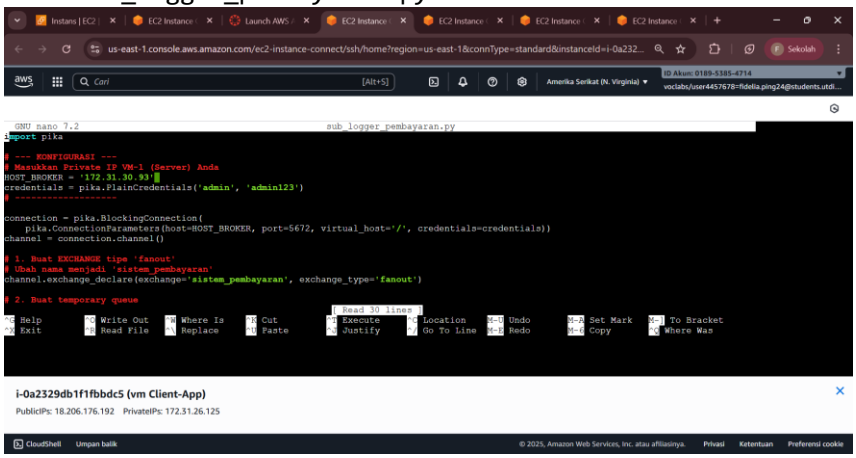
Lihat Terminal 1 dan Terminal 2 secara bersamaan.

- Terminal 1 akan muncul: [LOG] Mencatat transaksi: Pembelian Laptop Gaming...
- Terminal 2 akan muncul: [ALERT] Mengirim SMS ke Admin: Ada Pembelian Laptop Gaming...

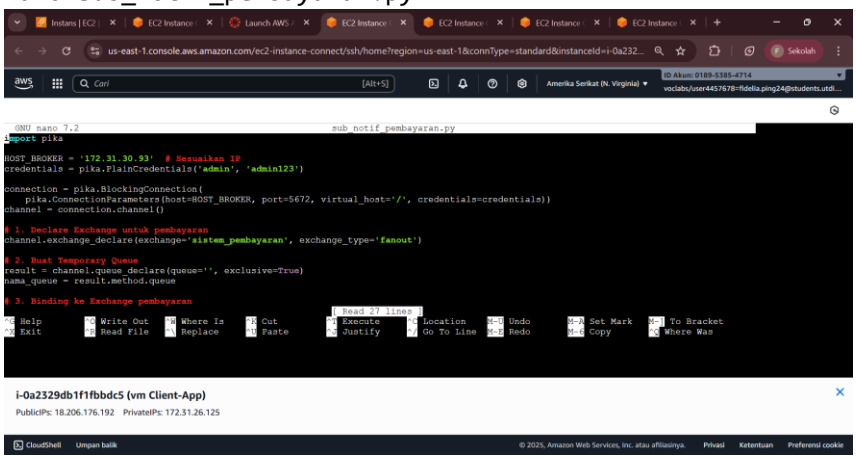


TUGAS

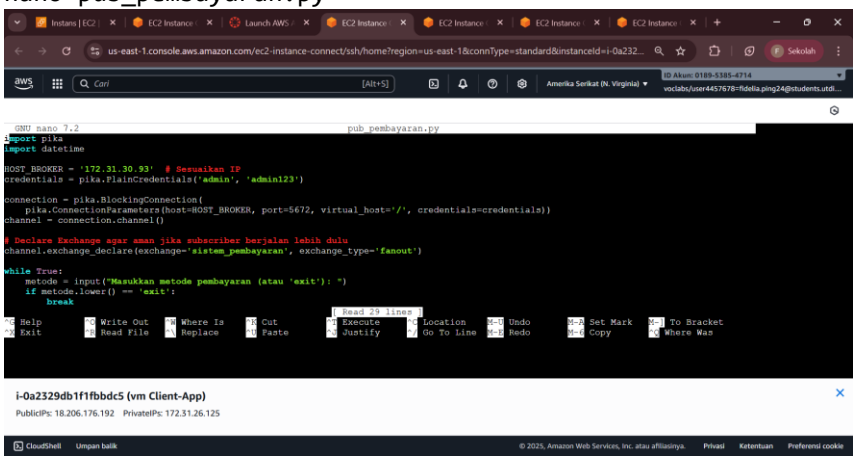
Nano sub_logger_pembayaran.py



nano sub_notif_pembayaran.py



nano pub_pembayaran.py



python3 sub_logger_pembayaran.py

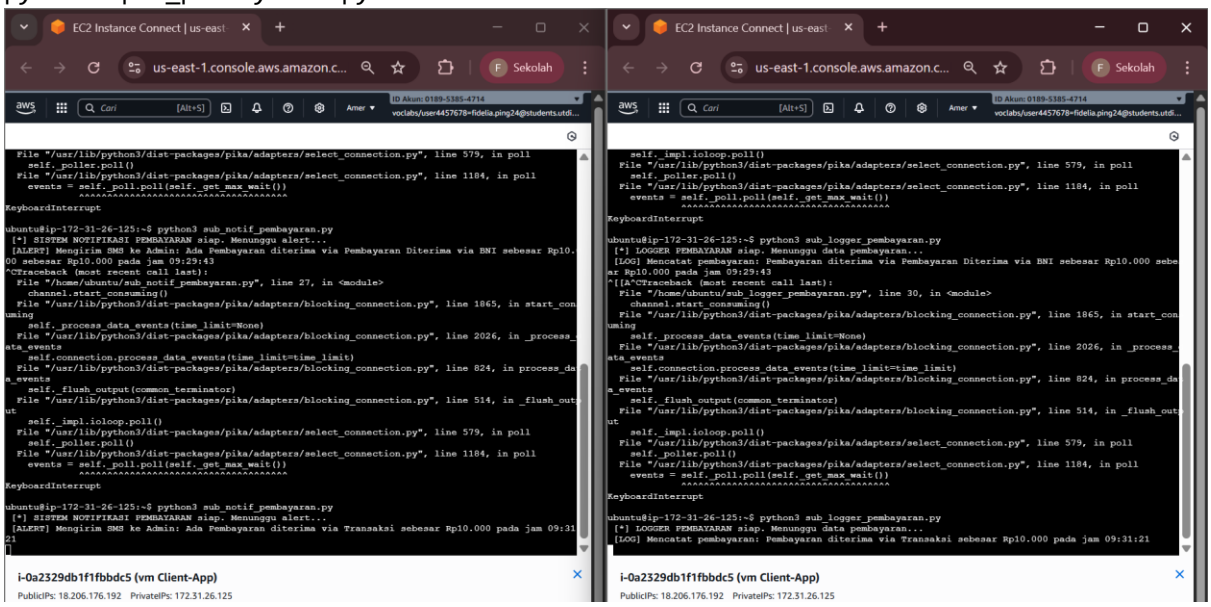
```
ubuntu@ip-172-31-26-125:~$ python3 pub_pembayaran.py
Masukkan metode pembayaran (atau 'exit'): Transaksi
Masukkan nominal pembayaran: Rp10.000
 [x] Data pembayaran terkirim: 'Pembayaran diterima via Transaksi sebesar Rp10.000 pada jam 09:31:21'
Masukkan metode pembayaran (atau 'exit'):
```

i-0a2329db1f1fbbdc5 (vm Client-App)

PublicIPs: 18.206.176.192 PrivateIPs: 172.31.26.125

python3 sub_notif_pembayaran.py

python3 pub_pembayaran.py



KESIMPULAN

Saya membuat tiga komponen utama, yaitu publisher, logger, dan notifikasi. Publisher bertugas mengirimkan pesan terkait transaksi atau pembayaran, sedangkan subscriber logger mencatat pesan tersebut, dan subscriber notifikasi menampilkan peringatan seolah mengirimkan SMS atau notifikasi kepada admin. Dari hasil uji coba, saya melihat bahwa setiap pesan yang saya kirimkan melalui publisher dapat langsung diterima oleh kedua subscriber tanpa adanya konflik atau keterikatan langsung antar komponen.