|  |
| --- |
| **CAK3EAB3 KOMPUTASI AWAN DAN TERDISTRIBUSI**  **Tugas-4**  **KOMUNIKASI PADA SISTEM TERDISTIBUSI**  **Kelompok: xxx**  **KELAS IF-46-10**  **Dosen: ISB** |
| http://adam.staff.telkomuniversity.ac.id/wp-content/uploads/sites/10/2013/12/3.-Logo-Telkom-University-Konfigurasi-Memusat.png  **PROGRAM STUDI INFORMATIKA**  **FAKULTAS INFORMATIKA**  **UNIVERSITAS TELKOM**  **BANDUNG**  **2024** |

**Ketentuan:**

1. Dikerjakan secara kelompok.
2. Dikerjakan langsung pada lembar soal.
3. Untuk soal perhitungan, maka cara atau langkah pengerjaan **wajib** dituliskan.
4. Setiap anggota kelompok wajib mengerjakan soal (ikut berkontribusi).
5. Anggota kelompok yang tidak ikut mengerjakan maka nilainya nol, nama mahasiswa tersebut tidak perlu ditulis di tabel.
6. Unggah jawaban ke LMS diwakili oleh salah satu anggota kelompok.
7. Lengkapi tabel pernyataan di bawah ini sebagai syarat nilai diinput.

**PERNYATAAN**

**Saya menyatakan bahwa saya:**

**(a) benar-benar ikut mengerjakan soal**

**(b) telah menjelaskan jawaban saya ke teman anggota kelompok**

**(c) telah memahami penjelasan dari teman anggota kelompok**

**(d) tidak mencontek jawaban dari kelompok lain**

**(e) tidak memberikan jawaban PR ini ke kelompok lain**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **NIM** | **Kontribusi Tugas** | **Tanda Tangan** |
| **Pradipa Rafi Anggaraksa** | **1301223488** |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TUGAS 4**

1. **Buatlah program sederhana menggunakan RPC!**
2. **Buatlah program sederhana menggunakan MQTT!**

**Jawaban**

1. Rpc\_Server.py

import rpc

class RPCServer(rpc.RPCServer):

def \_\_init\_\_(self):

super(RPCServer, self).\_\_init\_\_()

def add(self, a, b):

return a + b

def subtract(self, a, b):

return a - b

server = RPCServer()

server.register\_instance()

server.serve\_forever()

Rpc\_client.py

import rpc

# Create an RPC client

client = rpc.RPCClient()

# Connect to the RPC server

client.connect('localhost', 8080)

# Call the RPC methods

result\_add = client.add(2, 3)

result\_subtract = client.subtract(5, 2)

print(f"Result of addition: {result\_add}")

print(f"Result of subtraction: {result\_subtract}")

1. Program Publisher

import paho.mqtt.client as mqtt

broker\_url = "localhost"

broker\_port = 1883

topik = "test/mqtt"

client = mqtt.Client()

client.connect(broker\_url, broker\_port)

client.publish(topik, "Halo, MQTT!")

client.disconnect()

Publisher Subscriber

import paho.mqtt.client as mqtt

# Konfigurasi MQTT

broker\_url = "localhost"

broker\_port = 1883

topik = "test/mqtt"

# Buat klien MQTT

client = mqtt.Client()

# Koneksi ke broker

client.connect(broker\_url, broker\_port)

# Fungsi callback untuk menerima pesan

def on\_message(client, userdata, message):

print("Pesan diterima:", message.payload.decode())

# Berlangganan ke topik

client.subscribe(topik)

# Jalankan loop untuk menerima pesan

client.loop\_forever()