

LAPORAN PRAKTIKUM PEMROGRAMAN JARINGAN

Mata Kuliah : Bahasa Pemrograman Jaringan



Disusun Oleh :

MUDMAINNAH FUNGKARUA

231401026

Dosen Pengampu : Ucok, S.Kom.,MT

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS INDONESIA TIMUR

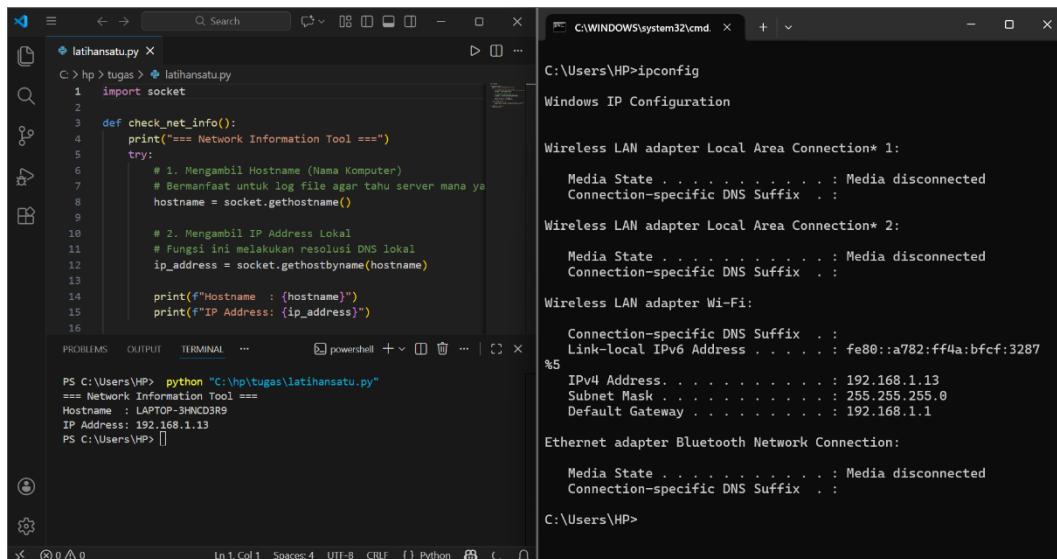
MAKASSAR

2026

BAB I

KONSEP DASAR PEMROGRAMAN JARINGAN

Tujuan praktikum untuk memahami konsep dasar pemrograman jaringan komputer melalui implementasi modul socket pada bahasa pemrograman Python, mengidentifikasi hostname sebagai identitas perangkat dalam jaringan, serta mengetahui alamat IP lokal yang digunakan oleh komputer. Selain itu, praktik ini bertujuan untuk memverifikasi kesesuaian informasi jaringan yang diperoleh melalui pemrograman Python dengan konfigurasi jaringan pada sistem operasi Windows menggunakan perintah ipconfig.



The screenshot shows a terminal window split into two panes. The left pane displays a Python script named 'latihansatu.py' with the following code:

```
latihansatu.py
C > hp > tugas > latihansatu.py
1 import socket
2
3 def check_net_info():
4     print("== Network Information Tool ==")
5     try:
6         # 1. Mengambil Hostname (Nama Komputer)
7         # Bermanfaat untuk log file agar tahu server mana ya
8         hostname = socket.gethostname()
9
10        # 2. Mengambil IP Address Lokal
11        # Fungsi ini melakukan resolusi DNS lokal
12        ip_address = socket.gethostbyname(hostname)
13
14        print(f"Hostname : {hostname}")
15        print(f"IP Address: {ip_address}")
16
```

The right pane shows the output of the 'ipconfig' command in a Windows command prompt:

```
C:\Windows\system32\cmd. x
C:\Users\HP>ipconfig
Windows IP Configuration

Wireless LAN adapter Local Area Connection* 1:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 2:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:
    Connection-specific DNS Suffix . :
    Link-local IPv6 Address . . . . . : fe80::a782:ff4a:bfcf:3287
%5
    IPv4 Address . . . . . : 192.168.1.13
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

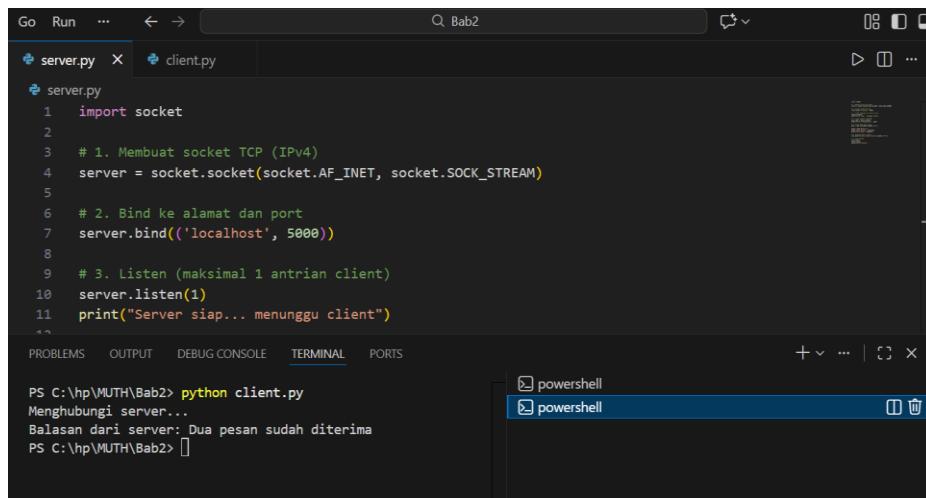
Ethernet adapter Bluetooth Network Connection:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

C:\Users\HP>
```

BAB II

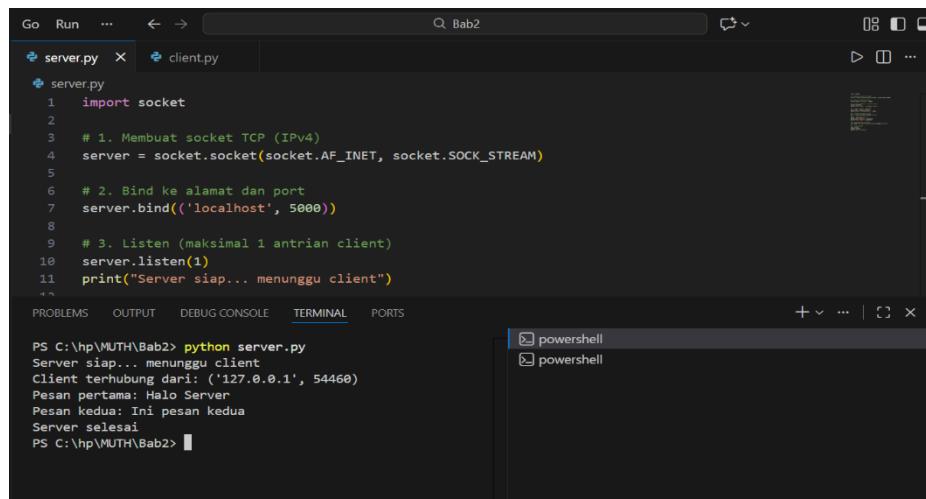
SOCKET API DASAR

Praktikum ini bertujuan untuk mengenalkan konsep dasar komunikasi client-server dengan menggunakan TCP socket di Python. Kegiatan ini memberikan pemahaman tentang alur eksekusi program server dan client yang dijalankan pada dua terminal terpisah, serta menjelaskan bagaimana server menunggu koneksi dari client melalui blocking call pada fungsi accept(). Selain itu, praktikum ini memberikan pengalaman dalam melakukan pertukaran data secara sinkron antara client dan server, termasuk proses pengiriman dan penerimaan pesan menggunakan send() dan recv(). Praktikum juga menekankan pentingnya pengolahan data melalui encoding dan decoding dari string ke bytes dan sebaliknya. Diharapkan, melalui latihan ini, peserta memperoleh pemahaman yang kokoh mengenai mekanisme komunikasi jaringan sederhana sebagai dasar untuk pengembangan aplikasi jaringan yang lebih kompleks.



```
server.py
1 import socket
2
3 # 1. Membuat socket TCP (IPv4)
4 server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5
6 # 2. Bind ke alamat dan port
7 server.bind(('localhost', 5000))
8
9 # 3. Listen (maksimal 1 antrian client)
10 server.listen(1)
11 print("Server siap... menunggu client")
```

```
PS C:\hp\MUTH\Bab2> python client.py
Menghubungi server...
Balasan dari server: Dua pesan sudah diterima
PS C:\hp\MUTH\Bab2>
```



```
server.py
1 import socket
2
3 # 1. Membuat socket TCP (IPv4)
4 server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5
6 # 2. Bind ke alamat dan port
7 server.bind(('localhost', 5000))
8
9 # 3. Listen (maksimal 1 antrian client)
10 server.listen(1)
11 print("Server siap... menunggu client")
```

```
PS C:\hp\MUTH\Bab2> python server.py
Server siap... menunggu client
Client terhubung dari: ('127.0.0.1', 54460)
Pesan pertama: Halo Server
Pesan kedua: Ini pesan kedua
Server selesai
PS C:\hp\MUTH\Bab2>
```

BAB III

PROTOKOL TCP (APLIKASI CHAT)

Praktikum ini bertujuan untuk memahami konsep komunikasi client-server dengan koneksi persistent menggunakan TCP socket di Python. Peserta mempelajari bagaimana server dan client dapat saling bertukar pesan secara berkelanjutan, penerapan loop untuk menjaga koneksi tetap aktif, serta penggunaan kata kunci untuk mengakhiri sesi secara tertib. Praktikum ini juga menekankan pentingnya encoding dan decoding data saat pengiriman pesan.

Peserta berhasil membangun aplikasi chat sederhana di mana server dan client dapat saling bertukar pesan. Diperoleh pemahaman bahwa komunikasi berjalan secara blocking, sehingga percakapan bersifat bergantian (walkie-talkie). Hal ini menunjukkan keterbatasan metode sederhana dan menekankan kebutuhan penggunaan mekanisme concurrency atau threading untuk chat real-time yang dapat berjalan dua arah secara simultan.

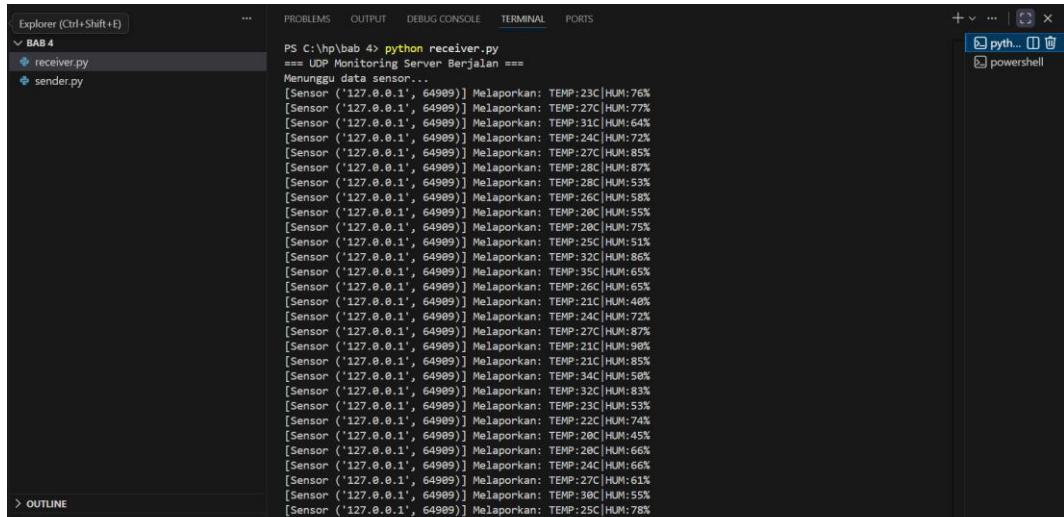
```
EXPLORER ... PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + ... x
LATIHAN BAB3
client.py
server.py
Server berjalan... menunggu client...
Client terhubung: ('127.0.0.1', 51666)
Password diterima: 12345
Client login berhasil
Client > halo selamat siang
Server > oke
Client > apa kabar?
Server > baik.
[]
```

```
EXPLORER ... PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + ... x
LATIHAN BAB3
client.py
server.py
PS C:\hp\latihan bab3> python client.py
Masukkan password:
Password > 12345
Login sukses! Selamat datang.
Client > halo selamat siang
Server > oke
Client > apa kabar?
Server > baik.
Client > []
```

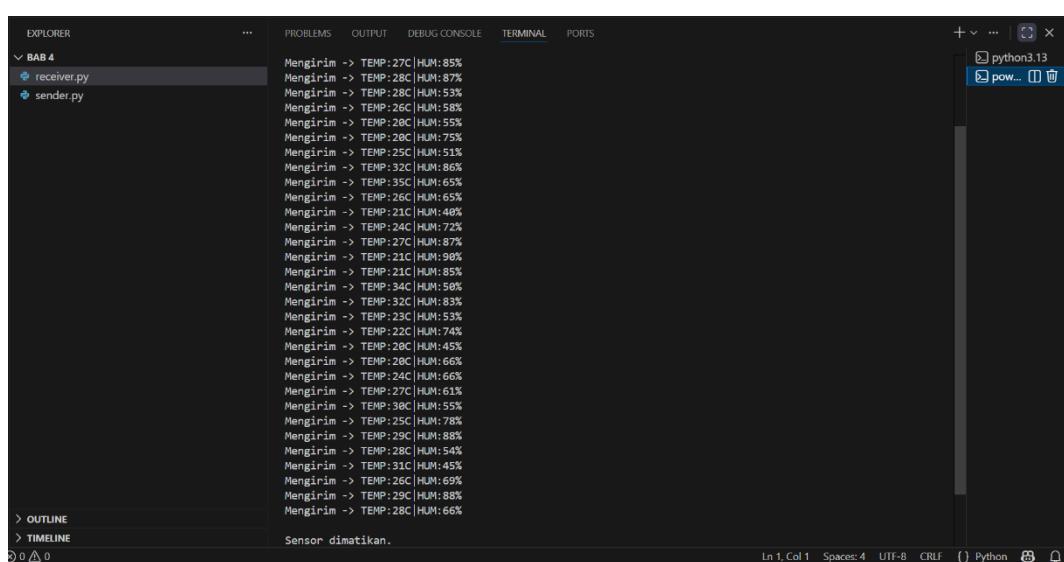
BAB IV

PROTOKOL UDP (STREAMING & BROADCASTING)

Tujuan praktikum ini adalah untuk memahami konsep protokol UDP sebagai protokol connectionless dan low-overhead, serta cara implementasinya menggunakan Python. Praktikum ini bertujuan agar peserta dapat mensimulasikan komunikasi data real-time antara sensor (client) dan server, memahami mekanisme pengiriman paket tanpa koneksi, serta mempraktikkan proses encoding dan decoding data saat pertukaran informasi. Selain itu, praktikum ini memberikan wawasan tentang penggunaan UDP dalam aplikasi industri, seperti IoT, video streaming, dan game online, di mana kecepatan transmisi lebih diutamakan dibandingkan keutuhan data.



```
PS C:\hp\bab 4> python receiver.py
== UDP Monitoring Server Berjalan ==
Menunggu data sensor...
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:23C|HUM:76%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:27C|HUM:77%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:31C|HUM:64%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:24C|HUM:72%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:27C|HUM:85%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:28C|HUM:87%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:28C|HUM:53%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:26C|HUM:58%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:28C|HUM:55%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:28C|HUM:75%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:25C|HUM:51%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:32C|HUM:86%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:35C|HUM:65%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:26C|HUM:65%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:21C|HUM:40%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:24C|HUM:72%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:27C|HUM:87%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:21C|HUM:96%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:21C|HUM:85%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:34C|HUM:50%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:32C|HUM:83%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:23C|HUM:53%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:22C|HUM:74%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:28C|HUM:45%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:28C|HUM:66%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:24C|HUM:66%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:27C|HUM:61%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:30C|HUM:55%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:25C|HUM:78%
[Sensor ('127.0.0.1', 64909)] Melaporkan: TEMP:29C|HUM:88%
```



```
Mengirim --> TEMP:27C HUM:85%
Mengirim --> TEMP:28C HUM:87%
Mengirim --> TEMP:28C HUM:53%
Mengirim --> TEMP:26C HUM:58%
Mengirim --> TEMP:28C HUM:55%
Mengirim --> TEMP:28C HUM:75%
Mengirim --> TEMP:25C HUM:51%
Mengirim --> TEMP:32C HUM:86%
Mengirim --> TEMP:35C HUM:65%
Mengirim --> TEMP:26C HUM:65%
Mengirim --> TEMP:21C HUM:48%
Mengirim --> TEMP:24C HUM:72%
Mengirim --> TEMP:27C HUM:87%
Mengirim --> TEMP:21C HUM:96%
Mengirim --> TEMP:21C HUM:85%
Mengirim --> TEMP:34C HUM:50%
Mengirim --> TEMP:32C HUM:83%
Mengirim --> TEMP:23C HUM:53%
Mengirim --> TEMP:22C HUM:74%
Mengirim --> TEMP:28C HUM:45%
Mengirim --> TEMP:28C HUM:66%
Mengirim --> TEMP:24C HUM:66%
Mengirim --> TEMP:27C HUM:61%
Mengirim --> TEMP:30C HUM:55%
Mengirim --> TEMP:25C HUM:78%
Mengirim --> TEMP:29C HUM:88%
Mengirim --> TEMP:28C HUM:54%
Mengirim --> TEMP:31C HUM:45%
Mengirim --> TEMP:26C HUM:69%
Mengirim --> TEMP:29C HUM:88%
Mengirim --> TEMP:28C HUM:66%
```

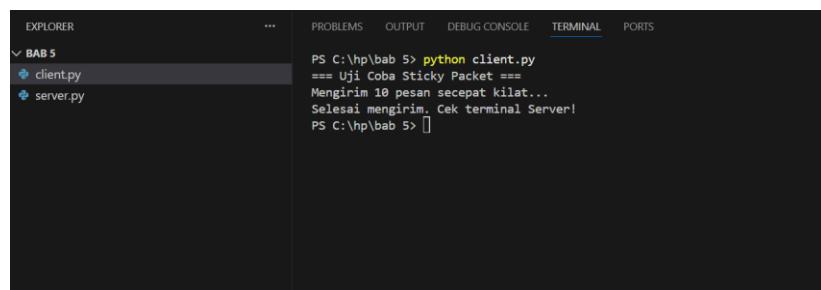
Sensor dimatikan.

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF () Python   

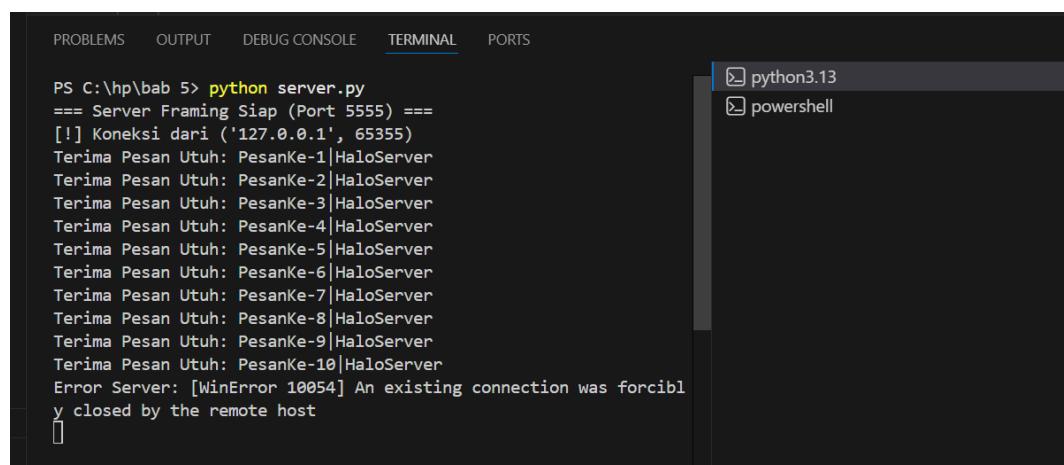
BAB V

ERROR HANDLING & FRAMING DATA

Tujuan praktikum ini adalah untuk memahami fenomena sticky packet pada tcp, di mana pesan yang dikirim secara terpisah dapat diterima sebagai satu kesatuan atau terfragmentasi karena sifat stream-oriented tcp. Praktikum ini bertujuan agar peserta dapat mempelajari dan menerapkan teknik framing untuk menjaga integritas pesan, sehingga penerima dapat mengetahui batas awal dan akhir setiap pesan. Selain itu, peserta mempraktikkan dua strategi framing utama, yaitu delimiter dan length-prefixed, serta menggunakan metode makefile() pada socket python untuk membaca pesan secara utuh per baris. Dengan praktikum ini, peserta diharapkan mampu membangun aplikasi tcp yang lebih robust, menghindari kesalahan interpretasi pesan, dan memahami pentingnya pengelolaan aliran data dalam komunikasi jaringan real-time.



```
PS C:\hp\bab 5> python client.py
== Uji Coba Sticky Packet ==
Mengirim 10 pesan secepat kilat...
Selesai mengirim. Cek terminal Server!
PS C:\hp\bab 5>
```

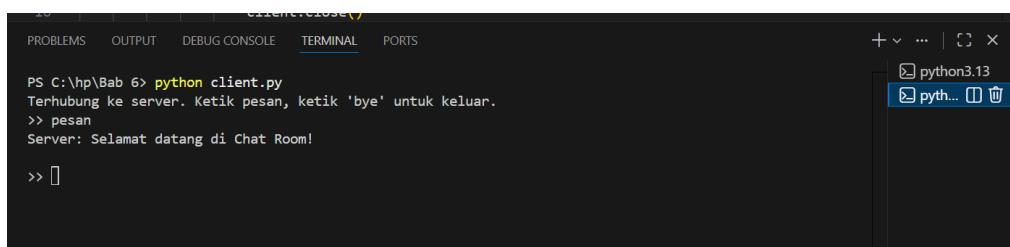


```
PS C:\hp\bab 5> python server.py
== Server Framing Siap (Port 5555) ==
[!] Koneksi dari ('127.0.0.1', 65355)
Terima Pesan Utuh: PesanKe-1|HaloServer
Terima Pesan Utuh: PesanKe-2|HaloServer
Terima Pesan Utuh: PesanKe-3|HaloServer
Terima Pesan Utuh: PesanKe-4|HaloServer
Terima Pesan Utuh: PesanKe-5|HaloServer
Terima Pesan Utuh: PesanKe-6|HaloServer
Terima Pesan Utuh: PesanKe-7|HaloServer
Terima Pesan Utuh: PesanKe-8|HaloServer
Terima Pesan Utuh: PesanKe-9|HaloServer
Terima Pesan Utuh: PesanKe-10|HaloServer
Error Server: [WinError 10054] An existing connection was forcibly closed by the remote host
PS C:\hp\bab 5>
```

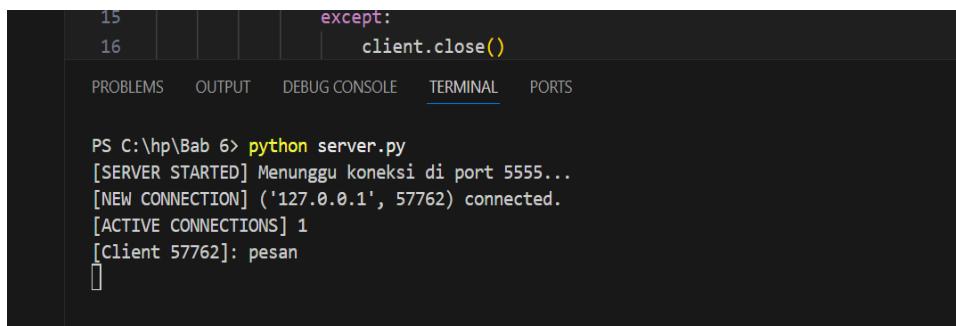
BAB VI

CONCURRENCY PART I – THREADING

Tujuan praktikum ini adalah untuk memahami konsep concurrency pada server menggunakan threading agar dapat melayani banyak client secara bersamaan. Praktikum ini bertujuan agar peserta dapat membandingkan keterbatasan server single-thread (iterative server) dengan server multi-thread, memahami cara membuat worker thread untuk setiap koneksi client, serta belajar menangani shared resource secara aman menggunakan mekanisme locking untuk mencegah race condition. Selain itu, praktikum ini memberikan pengalaman membangun aplikasi chat room multi-client yang dapat melakukan broadcast pesan, sehingga peserta dapat memahami prinsip desain server yang scalable dan mampu melayani komunikasi real-time secara simultan.



```
client.close()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\hp\Bab 6> python client.py
Terhubung ke server. Ketik pesan, ketik 'bye' untuk keluar.
>> pesan
Server: Selamat datang di Chat Room!
>> 
```



```
15 except:
16     client.close()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\hp\Bab 6> python server.py
[SERVER STARTED] Menunggu koneksi di port 5555...
[NEW CONNECTION] ('127.0.0.1', 57762) connected.
[ACTIVE CONNECTIONS] 1
[Client 57762]: pesan

```

BAB VII

SERIALISASI DATA (JSON & PICKLE)

Tujuan praktikum ini adalah untuk memahami konsep serialisasi dan deserialisasi data dalam komunikasi jaringan menggunakan Python. Praktikum ini bertujuan agar peserta dapat mengirim dan menerima data terstruktur (misalnya dictionary atau objek mahasiswa) melalui socket, serta membandingkan dua metode populer, yaitu JSON dan Pickle. Peserta juga diharapkan memahami keterbatasan socket yang hanya mengenal bytes, serta bagaimana mengubah objek kompleks menjadi bentuk yang dapat dikirim (marshalling) dan dikembalikan menjadi objek asli di sisi penerima (unmarshalling). Selain itu, praktikum ini memberikan wawasan tentang keamanan dan interoperabilitas, menunjukkan mengapa JSON lebih aman untuk komunikasi antar-aplikasi dibandingkan Pickle, serta mempraktikkan pembuatan protokol sederhana antara client dan server untuk pertukaran data mahasiswa.

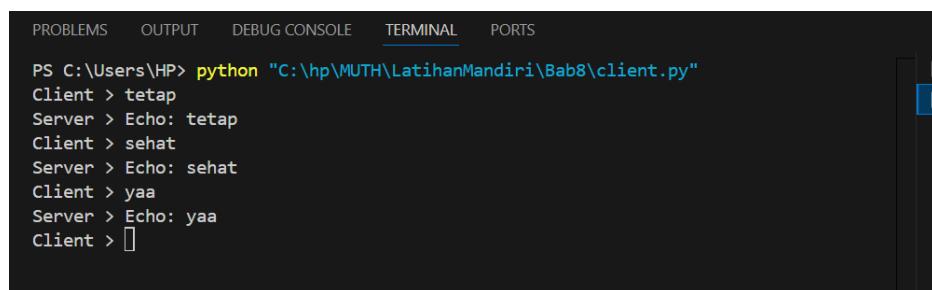
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Mengirim request NIM: 2314010200
Respon Server:
{
    "status": "SUKSES",
    "data": {
        "nama": "Mutma",
        "prodi": "Teknik Informatika",
        "ipk": 3.85
    }
}
-----
Mengirim request NIM: 1302020238
Respon Server:
{
    "status": "SUKSES",
    "data": {
        "nama": "Azizah",
        "prodi": "Sistem Informasi",
        "ipk": 3.9
    }
}
-----
Mengirim request NIM: 1302020242
Respon Server:
{
    "status": "SUKSES",
    "data": {
```

```
PS C:\hp\MUTH\LatihanMandiri\Bab7> python server.py
== Database Server (JSON) Berjalan di Port 6000 ==
[KONEKSI] Dari ('127.0.0.1', 54433)
[KONEKSI] Dari ('127.0.0.1', 54434)
[KONEKSI] Dari ('127.0.0.1', 54435)
```

BAB VIII

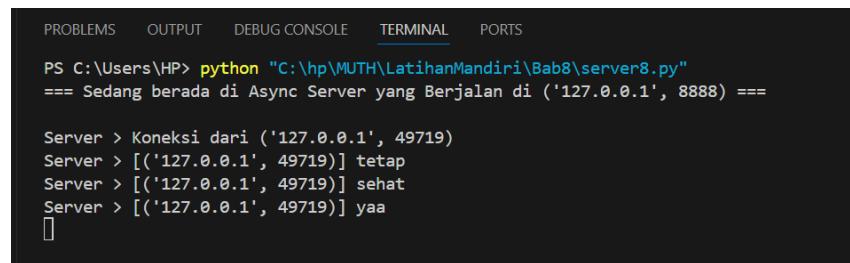
ASYNCHRONOUS I/O (CONCURRENCY PART II)

Tujuan praktikum ini adalah untuk memahami konsep asynchronous I/O menggunakan modul asyncio di Python dalam pembuatan server. Praktikum ini bertujuan agar peserta dapat membandingkan model server berbasis threading dan asynchronous, serta menyadari keterbatasan penggunaan banyak thread untuk jumlah koneksi yang besar (C10k Problem). Peserta juga diharapkan memahami bagaimana event loop, coroutine (async def), dan await bekerja untuk menangani ribuan koneksi secara efisien dalam satu thread, serta belajar menghindari kode blocking yang dapat menghentikan seluruh server. Praktikum ini memberikan pengalaman praktis membangun server high-performance yang mampu melayani banyak client secara simultan dengan penggunaan memori yang minimal.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HP> python "C:\hp\MUTH\LatihanMandiri\Bab8\client.py"
Client > tetap
Server > Echo: tetap
Client > sehat
Server > Echo: sehat
Client > yaa
Server > Echo: yaa
Client > 
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

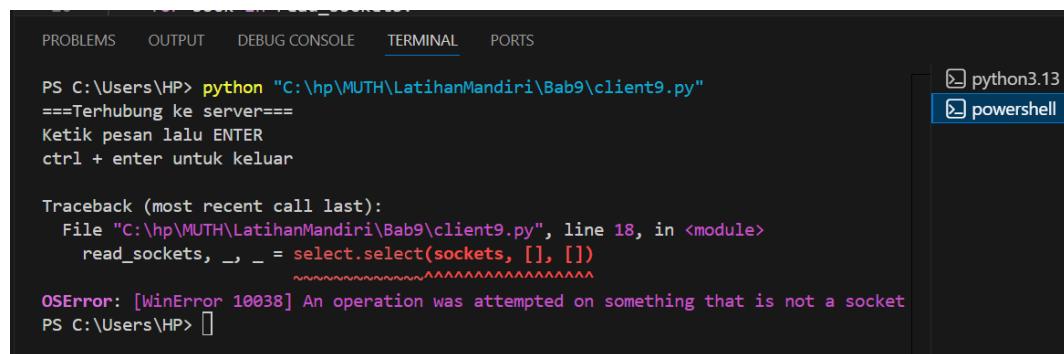
PS C:\Users\HP> python "C:\hp\MUTH\LatihanMandiri\Bab8\server8.py"
==> Sedang berada di Async Server yang Berjalan di ('127.0.0.1', 8888) ==>

Server > Koneksi dari ('127.0.0.1', 49719)
Server > [('127.0.0.1', 49719)] tetap
Server > [('127.0.0.1', 49719)] sehat
Server > [('127.0.0.1', 49719)] yaa
[]
```

BAB IX

I/O MULTIPLEXING (SELECT & POLL)

Praktikum ini bertujuan agar peserta memahami cara kerja I/O Multiplexing di level sistem operasi, khususnya penggunaan fungsi select() untuk memantau banyak socket sekaligus. Peserta diharapkan dapat membedakan konsep Blocking I/O, Non-Blocking I/O, dan I/O Multiplexing, serta mampu membangun server konkuren hemat memori tanpa menggunakan threading. Praktikum ini juga mengajarkan peserta untuk menganalisis batasan skalabilitas select dibanding mekanisme modern seperti epoll pada Linux atau IOCP pada Windows, serta mengembangkan kemampuan pemecahan masalah pada server event-driven.

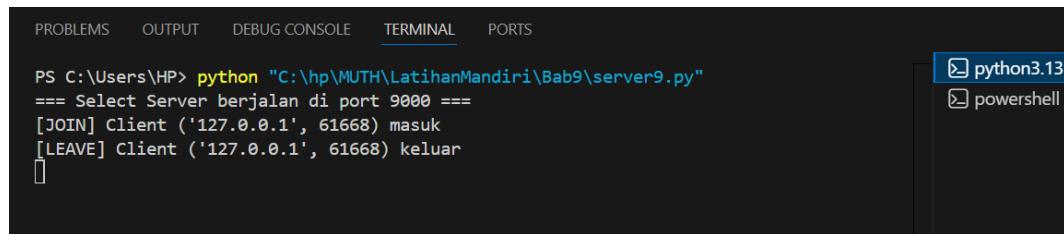


A screenshot of a terminal window in a dark-themed code editor. The tabs at the top are PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected. The output shows:

```
PS C:\Users\HP> python "C:\hp\MUTH\LatihanMandiri\Bab9\client9.py"
==Terhubung ke server==
Ketik pesan lalu ENTER
ctrl + enter untuk keluar

Traceback (most recent call last):
  File "C:\hp\MUTH\LatihanMandiri\Bab9\client9.py", line 18, in <module>
    read_sockets, _, _ = select.select(sockets, [], [])
                           ^^^^^^
OSError: [WinError 10038] An operation was attempted on something that is not a socket
PS C:\Users\HP>
```

The right side of the terminal shows two open ports: python3.13 and powershell.



A screenshot of a terminal window in a dark-themed code editor. The tabs at the top are PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected. The output shows:

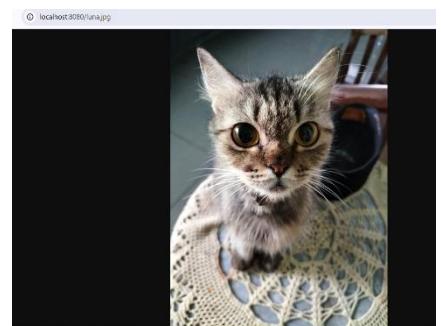
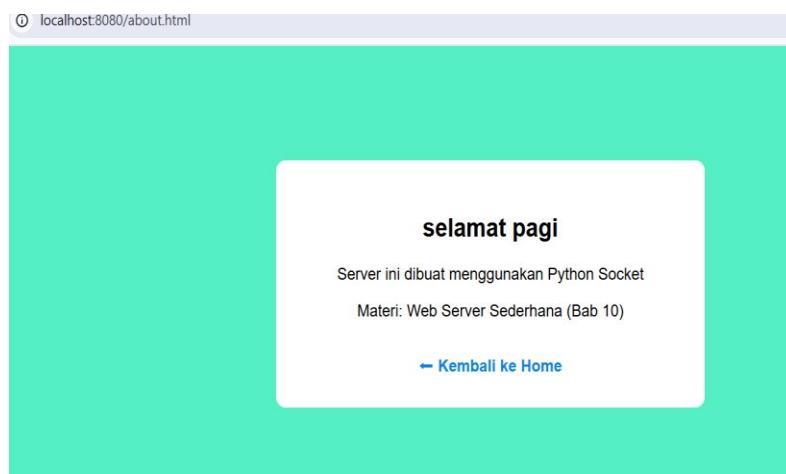
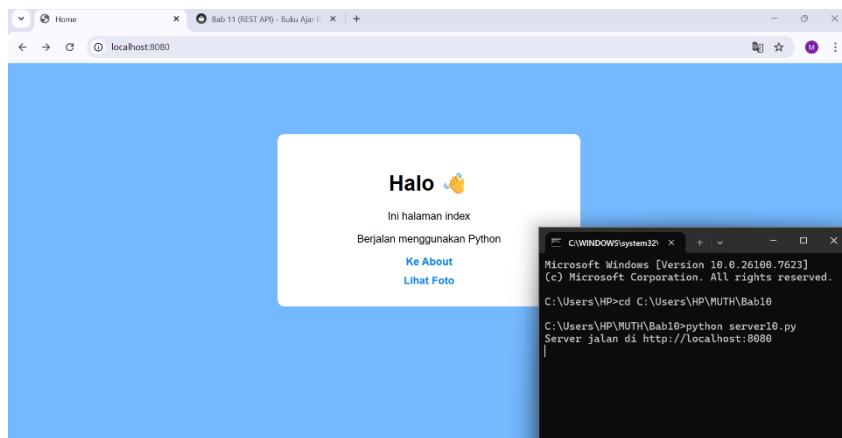
```
PS C:\Users\HP> python "C:\hp\MUTH\LatihanMandiri\Bab9\server9.py"
== Select Server berjalan di port 9000 ==
[JOIN] Client ('127.0.0.1', 61668) masuk
[LEAVE] Client ('127.0.0.1', 61668) keluar
[]
```

The right side of the terminal shows two open ports: python3.13 and powershell.

BAB X

PROTOKOL HTTP & WEB SERVER

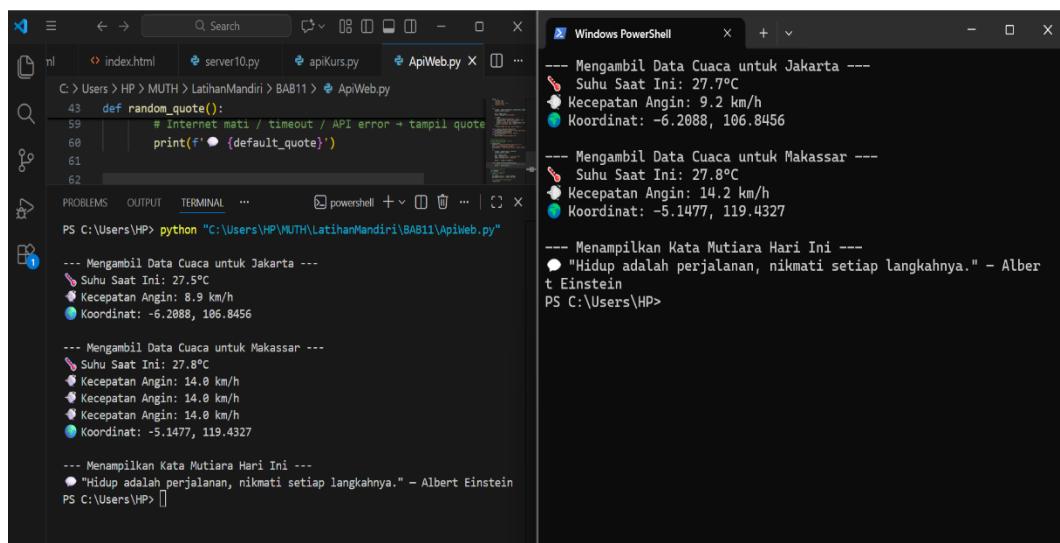
Praktikum ini bertujuan agar peserta memahami cara kerja protokol HTTP secara mendalam, termasuk struktur paket HTTP, metode request, header, body, dan kode status. Peserta diharapkan dapat menjelaskan siklus Request-Response di balik setiap interaksi web, serta membangun Web Server sederhana dari awal menggunakan TCP Socket yang mampu menyajikan file statis. Selain itu, praktikum ini melatih kemampuan peserta untuk mendiagnosa kode status HTTP seperti 200, 404, dan 500, serta memahami implikasinya terhadap pengalaman pengguna saat mengakses website.



BAB XI

REST API & WEB SERVICES

Praktikum ini bertujuan agar peserta dapat memahami cara komputer saling berkomunikasi (machine-to-machine) melalui penggunaan api, khususnya rest api, serta mengaplikasikan protokol http untuk pertukaran data yang terstruktur. Peserta diharapkan mampu membedakan komunikasi antara manusia dengan mesin dan antar mesin, memahami prinsip dasar rest, serta mengetahui bagaimana metode http seperti get, post, put/patch, dan delete merepresentasikan operasi crud. Selain itu, peserta akan belajar menggunakan pustaka python requests untuk mengirim permintaan http dan memproses respon dalam format json. Praktikum ini juga menekankan pembuatan aplikasi sederhana yang menarik data dari api publik, menampilkan di konsol, dan menerapkan error handling yang efektif agar program tetap berjalan meski terjadi masalah jaringan atau respon tidak valid. Peserta pun didorong untuk mengeksplorasi api publik lainnya dan memanfaatkan data real-time dalam pengembangan aplikasi python.



The screenshot shows a terminal window with two panes. The left pane displays a Python script named `ApiWeb.py` which defines a function `random_quote()` to fetch a random quote from an API. The right pane shows the command `python "C:/Users\HP\PycharmProjects\Bab11\ApiWeb.py"` being run in a Windows PowerShell window, and the resulting output: weather data for Jakarta and Makassar, and a random quote by Albert Einstein.

```
--- Mengambil Data Cuaca untuk Jakarta ---
• Suhu Saat Ini: 27.7°C
• Kecepatan Angin: 9.2 km/h
• Koordinat: -6.2088, 106.8456

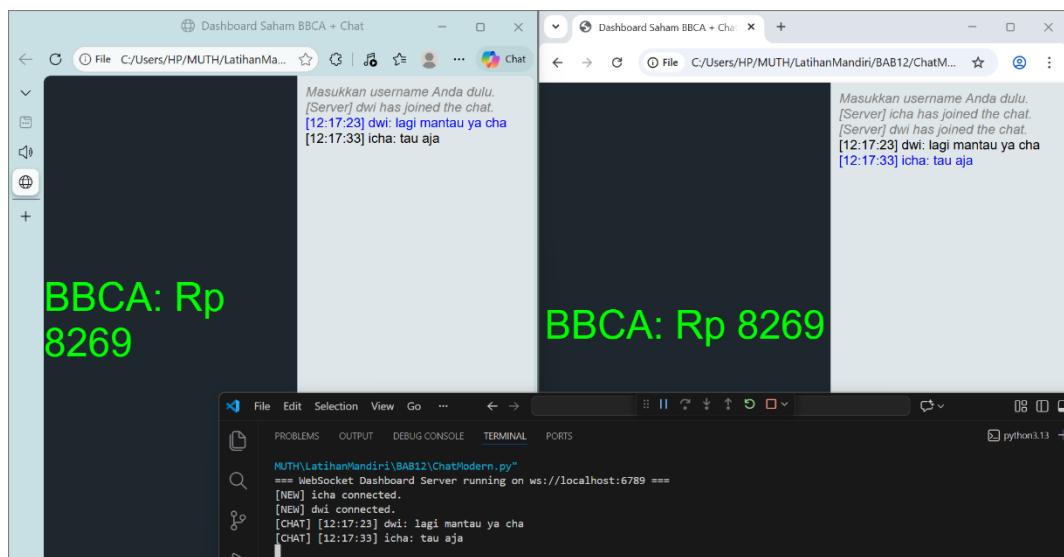
--- Mengambil Data Cuaca untuk Makassar ---
• Suhu Saat Ini: 27.8°C
• Kecepatan Angin: 14.2 km/h
• Koordinat: -5.1477, 119.4327

--- Menampilkan Kata Mutiara Hari Ini ---
• "Hidup adalah perjalanan, nikmati setiap langkahnya." - Albert Einstein
PS C:\Users\HP>
```

BAB XII

REAL-TIME COMMUNICATION (WEBSOCKET)

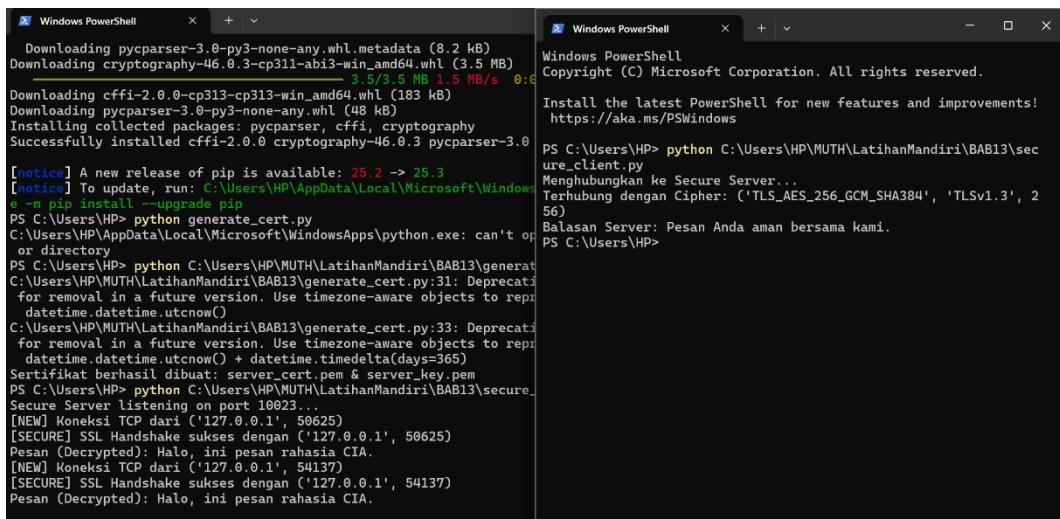
Praktikum ini bertujuan agar peserta memahami konsep komunikasi real-time antar mesin menggunakan protokol websocket, yang memungkinkan pertukaran data dua arah (full duplex) secara efisien dibandingkan http biasa. Peserta diharapkan dapat menjelaskan keterbatasan http untuk aplikasi modern seperti chat, game online, atau dashboard harga saham, dan memahami cara kerja websocket mulai dari handshake hingga pembentukan koneksi persisten. Praktikum ini juga melatih peserta dalam membangun server python berbasis asyncio yang dapat mem-broadcast data secara real-time ke banyak client sekaligus, serta membuat halaman html/javascript sederhana yang menerima dan menampilkan data secara dinamis. Selain itu, peserta akan mempraktikkan mekanisme pengiriman dan penerimaan pesan, manajemen koneksi client, serialisasi data menggunakan json, serta mengeksplorasi pengembangan aplikasi real-time, seperti live ticker saham atau chat room modern. Praktikum ini juga menekankan pentingnya pemahaman broadcasting dan pengelolaan koneksi agar aplikasi tetap responsif dan hemat sumber daya.



BAB XIII

KEAMANAN JARINGAN (NETWORK SECURITY)

Praktikum ini bertujuan untuk memberikan pemahaman mengenai pentingnya keamanan komunikasi jaringan, khususnya risiko pengiriman data dalam bentuk plaintext yang rentan terhadap serangan seperti sniffing dan man-in-the-middle. Melalui praktikum ini, peserta diharapkan mampu memahami konsep dasar kriptografi modern, termasuk perbedaan antara enkripsi simetris dan asimetris, serta peran pasangan kunci publik dan privat dalam menjaga kerahasiaan data. Praktikum ini juga melatih peserta dalam menerapkan SSL/TLS pada aplikasi client-server menggunakan modul ssl di Python untuk mengenkripsi lalu lintas komunikasi. Selain itu, peserta mempraktikkan pembuatan dan penggunaan self-signed certificate sebagai sarana pengamanan koneksi pada lingkungan pengembangan lokal, sehingga dapat membangun aplikasi jaringan yang lebih aman dan terlindungi dari penyadapan.



The image shows two adjacent Windows PowerShell windows. The left window displays the command-line output for installing SSL/TLS dependencies using pip. It shows the download of several packages: pycparser-3.0-py3-none-any.whl.metadata (8.2 kB), cryptography-46.0.3-cp311abi3-win_amd64.whl (3.5 MB), cffi-2.0.0-cp313-cp313-win_amd64.whl (183 kB), and pycparser-3.0-py3-none-any.whl (48 kB). The command `python generate_cert.py` is then run, which generates a self-signed certificate named `server.cert.pem` and `server.key.pem`. The right window shows the server listening on port 10023 and performing an SSL handshake with a client, displaying the message "Balasan Server: Pesan Anda aman bersama kami".

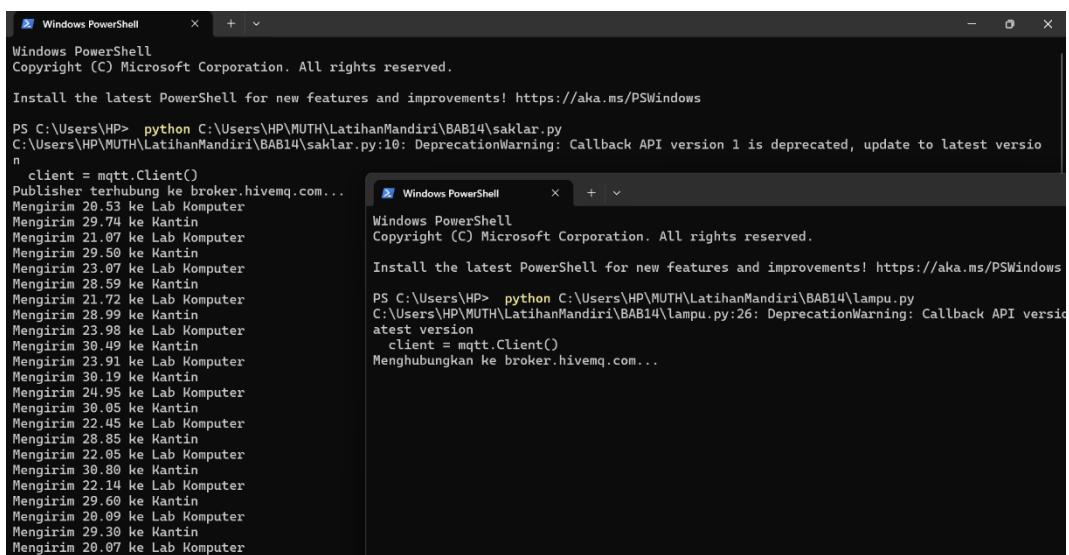
```
Windows PowerShell
Downloaded pycparser-3.0-py3-none-any.whl.metadata (8.2 kB)
Downloaded cryptography-46.0.3-cp311abi3-win_amd64.whl (3.5 MB) 3.5/3.5 MB 1.5 MB/s 0:0
Downloaded cffi-2.0.0-cp313-cp313-win_amd64.whl (183 kB)
Downloaded pycparser-3.0-py3-none-any.whl (48 kB)
Installing collected packages: pycparser, cffi, cryptography
Successfully installed cffi-2.0.0 cryptography-46.0.3 pycparser-3.0
[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: C:\Users\HP\AppData\Local\Microsoft\Windows\PowerShell\6.0\pip install --upgrade pip
PS C:\Users\HP> python generate_cert.py
C:\Users\HP\AppData\Local\Microsoft\WindowsApps\python.exe: can't open file or directory
PS C:\Users\HP> python C:\Users\HP\MUTH\LatihanMandiri\BAB13\generate_cert.py
C:\Users\HP\MUTH\LatihanMandiri\BAB13\generate_cert.py:31: DeprecationWarning: for removal in a future version. Use timezone-aware objects to represent datetime.datetime.utcnow()
C:\Users\HP\MUTH\LatihanMandiri\BAB13\generate_cert.py:33: DeprecationWarning: for removal in a future version. Use timezone-aware objects to represent datetime.datetime.utcnow() + datetime.timedelta(days=365)
Sertifikat berhasil dibuat: server.cert.pem & server.key.pem
PS C:\Users\HP> python C:\Users\HP\MUTH\LatihanMandiri\BAB13\secure_server.py
Secure Server listening on port 10023...
[NEW] Koneksi TCP dari ('127.0.0.1', 50625)
[SECURE] SSL Handshake sukses dengan ('127.0.0.1', 50625)
Pesan (Decrypted): Halo, ini pesan rahasia CIA.
[NEW] Koneksi TCP dari ('127.0.0.1', 54137)
[SECURE] SSL Handshake sukses dengan ('127.0.0.1', 54137)
Pesan (Decrypted): Halo, ini pesan rahasia CIA.
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements!
https://aka.ms/PSWindows
PS C:\Users\HP> python C:\Users\HP\MUTH\LatihanMandiri\BAB13\secure_client.py
Menghubungkan ke Secure Server...
Terhubung dengan Cipher: ('TLS_AES_256_GCM_SHA384', 'TLSv1.3', 256)
Balasan Server: Pesan Anda aman bersama kami.
PS C:\Users\HP>
```

BAB IV

ARSITEKTUR SISTEM TERDISTRIBUSI & IOT (MQTT)

Praktikum ini bertujuan agar peserta memahami perbedaan antara arsitektur request-response seperti http dan publish-subscribe seperti mqtt, khususnya dalam konteks iot dan sistem terdistribusi. Peserta diharapkan dapat menganalisis peran broker sebagai perantara pengiriman pesan, mengimplementasikan komunikasi data sensor menggunakan protokol mqtt dengan pustaka paho-mqtt, serta mengevaluasi keandalan pengiriman data melalui berbagai level quality of service (qos). Praktikum ini juga melatih peserta dalam membuat simulasi smart home atau sistem iot sederhana, termasuk penerapan pengendalian perangkat jarak jauh, sehingga memahami mekanisme pengiriman data real-time yang efisien, hemat bandwidth, dan aman untuk penggunaan skala besar.



The image shows two side-by-side Windows PowerShell windows. The left window is titled 'Windows PowerShell' and shows the command 'python C:\Users\HP\MUTH\LatihanMandiri\BAB14\saklar.py'. It outputs a series of messages starting with 'client = mqtt.Client()' and followed by numerous 'Mengirim' (Sending) messages to various locations like 'Lab Komputer' and 'Kantin' at different times (e.g., 20.53, 29.74, 21.07, etc.). The right window is also titled 'Windows PowerShell' and shows the command 'python C:\Users\HP\MUTH\LatihanMandiri\BAB14\lampa.py'. It follows a similar pattern with 'client = mqtt.Client()' and 'Mengirim' messages to 'Lab Komputer' and 'Kantin' at specific times.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HP> python C:\Users\HP\MUTH\LatihanMandiri\BAB14\saklar.py
C:\Users\HP\MUTH\LatihanMandiri\BAB14\saklar.py:10: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
Publisher terhubung ke broker.hivemq.com...
Mengirim 20.53 ke Lab Komputer
Mengirim 29.74 ke Kantin
Mengirim 21.07 ke Lab Komputer
Mengirim 29.50 ke Kantin
Mengirim 23.07 ke Lab Komputer
Mengirim 28.59 ke Kantin
Mengirim 21.72 ke Lab Komputer
Mengirim 28.99 ke Kantin
Mengirim 23.98 ke Lab Komputer
Mengirim 30.49 ke Kantin
Mengirim 23.91 ke Lab Komputer
Mengirim 30.19 ke Kantin
Mengirim 24.95 ke Lab Komputer
Mengirim 30.05 ke Kantin
Mengirim 22.45 ke Lab Komputer
Mengirim 28.85 ke Kantin
Mengirim 22.05 ke Lab Komputer
Mengirim 30.80 ke Kantin
Mengirim 22.14 ke Lab Komputer
Mengirim 29.60 ke Kantin
Mengirim 20.09 ke Lab Komputer
Mengirim 29.30 ke Kantin
Mengirim 20.07 ke Lab Komputer

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

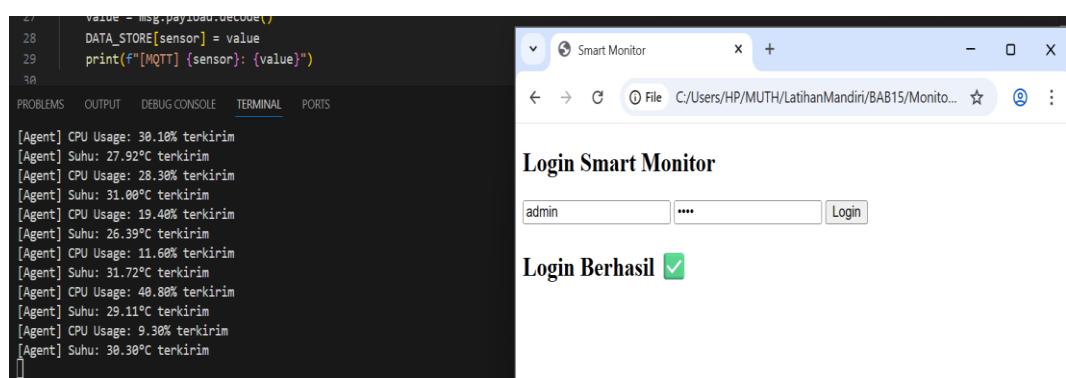
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HP> python C:\Users\HP\MUTH\LatihanMandiri\BAB14\lampa.py
C:\Users\HP\MUTH\LatihanMandiri\BAB14\lampa.py:26: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
Menghubungkan ke broker.hivemq.com...
```

BAB XV

PENUTUP & PROYEK AKHIR (CAPSTONE PROJECT)

Praktikum ini bertujuan agar peserta mampu merancang dan membangun sistem jaringan kompleks yang mengintegrasikan minimal tiga protokol berbeda seperti http, websocket, dan mqtt, serta menciptakan aplikasi full-stack network yang aman, andal, dan scalable sebagai portofolio akhir. Peserta diharapkan dapat mengevaluasi performa dan keamanan sistem yang telah dibuat, termasuk penerapan autentikasi, enkripsi, dan manajemen error, serta mendokumentasikan proyek secara profesional. Praktikum ini juga mendorong peserta untuk menyusun roadmap pengembangan diri di bidang network engineering, backend development, atau iot, sehingga memiliki fondasi kuat untuk melanjutkan karier di dunia jaringan dan aplikasi terdistribusi modern.



The image shows two windows side-by-side. On the left is a terminal window with the following code and logs:

```
27     value = msg.payload.decode()
28     DATA_STORE[sensor] = value
29     print(f"[MQTT] {sensor}: {value}")
30
```

[PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS]

[Agent] CPU Usage: 30.10% terkirim
[Agent] Suhu: 27.92°C terkirim
[Agent] CPU Usage: 28.30% terkirim
[Agent] Suhu: 31.00°C terkirim
[Agent] CPU Usage: 19.40% terkirim
[Agent] Suhu: 26.39°C terkirim
[Agent] CPU Usage: 11.60% terkirim
[Agent] Suhu: 31.72°C terkirim
[Agent] CPU Usage: 40.80% terkirim
[Agent] Suhu: 29.11°C terkirim
[Agent] CPU Usage: 9.30% terkirim
[Agent] Suhu: 30.30°C terkirim

On the right is a browser window titled "Smart Monitor" showing a login interface:

Smart Monitor

Login Smart Monitor

admin Login

Login Berhasil ✓