NABILAH ATIKA RAHMA
5025221005 - PROGJAR C
https://github.com/ranabel/network-programming-5025221005

Tugas 4 Tugas 4

Http server 1

- Dengan berbasis pada program http server pada https://github.com/rm77/progjar/tree/master/progjar5
    - Dengan method yang ada pada spesifikasi HTTP server, tambahkan kemampuan http server untuk
        - Melihat daftar file pada satu direktori
        - Mengupload sebuah file
        - Menghapus file
    - Jalankan http server dengan mode
        - Thread pool
        - Process pool
    - Buatlah client implementation dari operasi tambahan tersebut. Jalankan operasi client server untuk kemampuan tersebut, berikanlah screenshot seperlunya, dan penjelasan dalam paragraf

Instruksi submission

- Semua dimasukkan ke satu file PDF
- Dalam satu file PDF berisi
    - Link menuju ke repository di github
    - Penjelasan tentang modifikasi yang dilakukan baik pada client ataupun pada server (minimal 2 paragraf)
    - Capture operasi client server untuk operasi upload dan hapus
    - Deskripsi dan penjelasan minimal 1 paragraf untuk capture screenshot

GITHUB REPO :

NABILAH ATIKA RAHMA - 5025221005

https://github.com/ranabel/network-programming-5025221005

NABILAH ATIKA RAHMA
5025221005 - PROGJAR C
https://github.com/ranabel/network-programming-5025221005

Modifikasi pada Server

- **Modifikasi pada File http.py**

  Saya menambahkan beberapa fitur utama pada HTTP server, yaitu untuk menampilkan daftar file dalam direktori, mengunggah file, dan menghapus file. Fitur unggah menggunakan metode POST, sedangkan penghapusan file menggunakan metode DELETE. Untuk tampilan daftar atau list file, server akan memberi nama file. Saya juga menambahkan validasi path untuk mencegah akses ke luar folder utama (directory traversal), dan membatasi ukuran maksimum file yang bisa diunggah. Server juga dilengkapi dengan logging yang komprehensif yang menampilkan aktivitas real time di console, ,sehingga administrator dapat memantau semua operasi yang dilakukan client.

- **Modifikasi pada Thread Pool dan Process Pool**

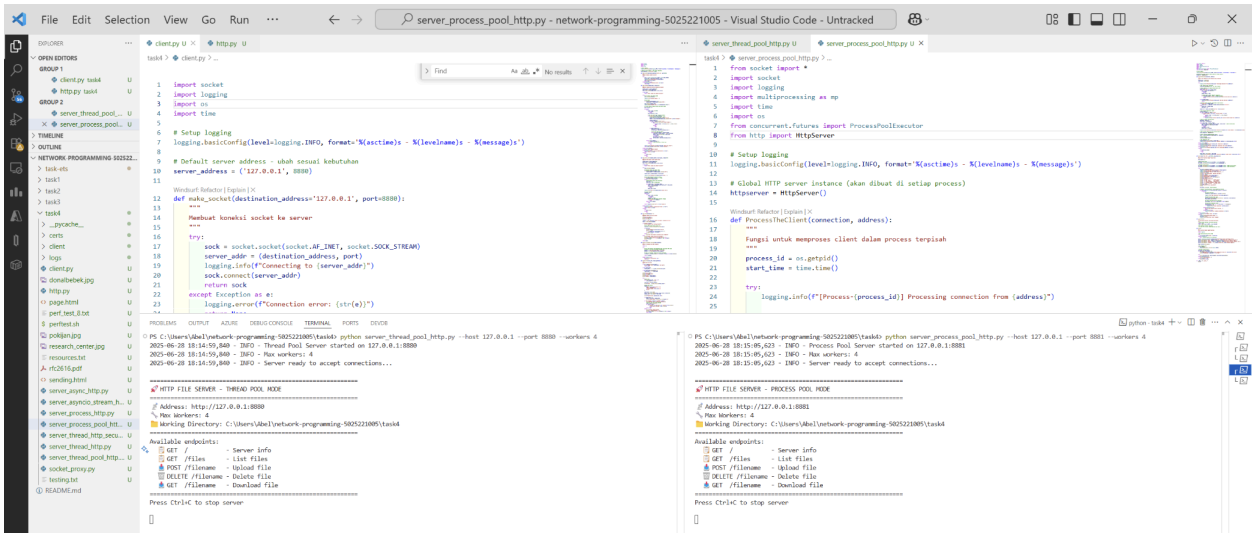
  Saya menjalankan server dalam dua mode yaitu thread pool dan process pool. Untuk thread pool, saya menggunakan ThreadPoolExecutor dengan maksimal 20 thread, di mana setiap permintaan (request) ditangani oleh thread yang berbeda. Server dilengkapi dengan timeout handling, buffer optimization, dan connection management yang lebih baik. Masing-masing thread juga memiliki ID yang dicatat dalam log agar lebih mudah dilacak. Keduanya saya berikan penambahan log untuk pemantauan activity yang cukup kompleks agar mudah untuk dipahami. Error handling juga diperbaiki untuk menangani various network exceptions seperti connection reset dan timeout.

Modifikasi pada Client

Client saya modifikasi agar mendukung operasi menampilkan / listing file, mengunggah / upload, mengunduh / download, dan menghapus / delete file. Program client bisa dijalankan dalam mode interaktif dengan menu pilihan, atau

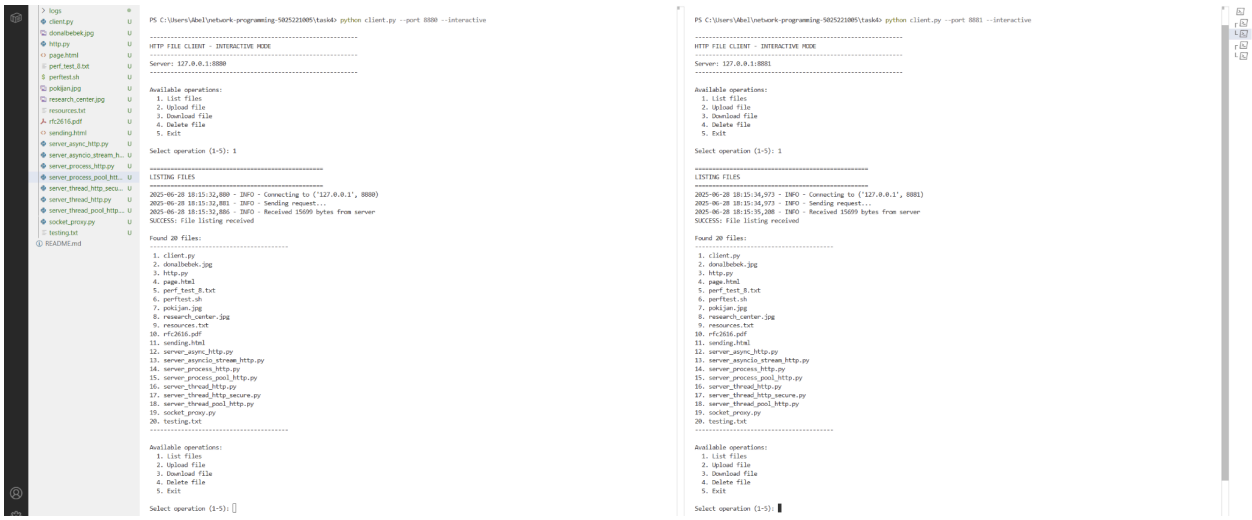
langsung lewat command line dengan argumen tertentu, hal tesebut juga sudah saya setup untuk lebih memantau dan meminimalkan ketidakmenauan bug. Operasi upload dan download menggunakan metode binary data handling, proper dengan progress tracking dan speed calculation. Client dapat menangani file dengan berbagai ukuran dan memberikan feedback real-time tentang progress upload. Saya juga menambahkan pengecekan kesalahan, seperti saat koneksi gagal atau file tidak ditemukan. Informasi aktivitas client akan tercatat di log untuk membantu proses debugging. Semua fitur dibuat sesederhana mungkin agar mudah digunakan. Semua operasi dibuat sejelas mungkin agar mudah digunakan siapa saja.

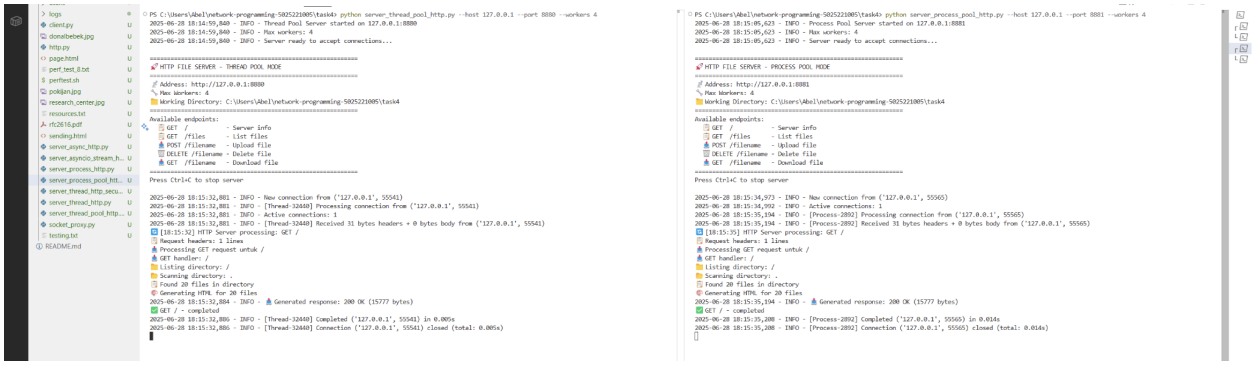
Berikut SS pada saat awal server open hingga client dan percobaan list, upload, download, delete file untuk thread dan process pool.



Berikut gambar ini menunjukkan server HTTP yang berjalan dalam mode thread pool pada port 8880, dan process pool pada port 8881 dengan tampilan awal yang menampilkan informasi server dan endpoint yang tersedia. Server log nantinya akan menampilkan aktivitas real-time termasuk thread ID dan processing time untuk setiap request. Disini juga terdapat address, max worker, dan working directory yang digunakan pada tugas ini.

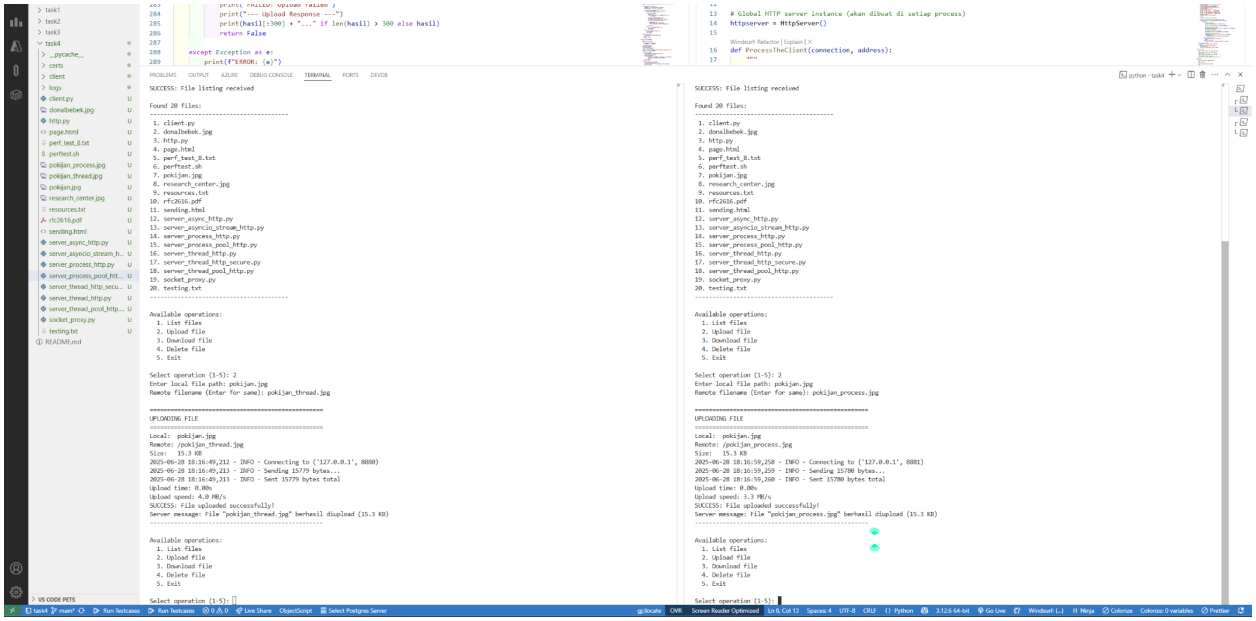
NABILAH ATIKA RAHMA
5025221005 - PROGJAR C
https://github.com/ranabel/network-programming-5025221005

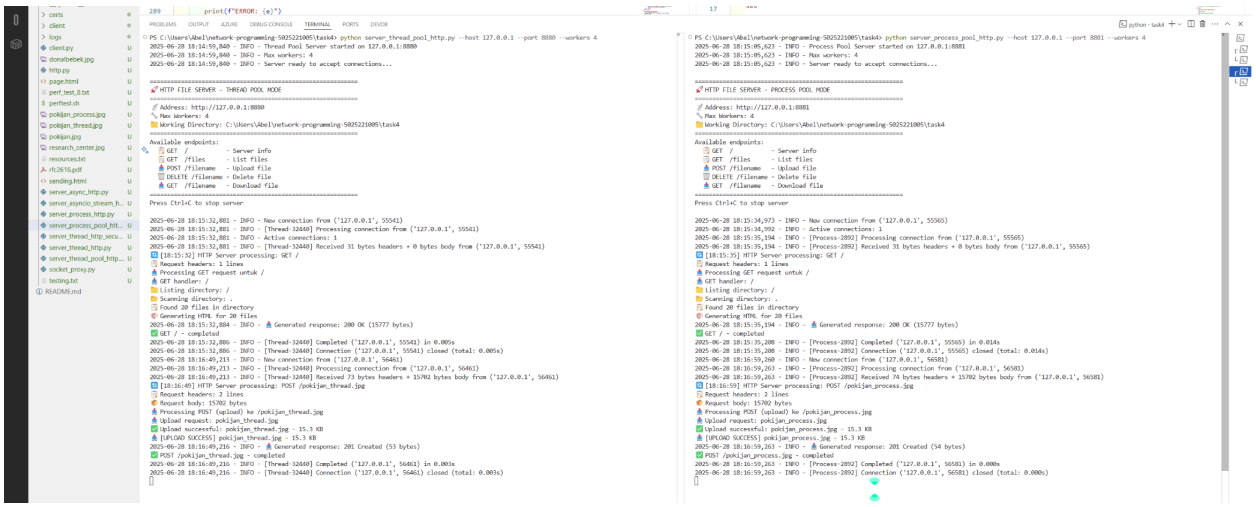




Pada gambar diatas ditunjukkan sisi client dan dibawah ditunjukkan di sisi server. Pada awalnya kita diberi pilihan ingin melakukan apa, list, upload, download, delete, atau exit dan terlihat bahwasannya jika kita memilih list, maka function akan mencari dan akan ditampilkan listing files yang ada dalam directory saat ini (pada directory tersebut terdapat 20 files mulai dari client.py sampai testing.txt), disini juga dapat kita lihat kita sedang di port dan address mana.

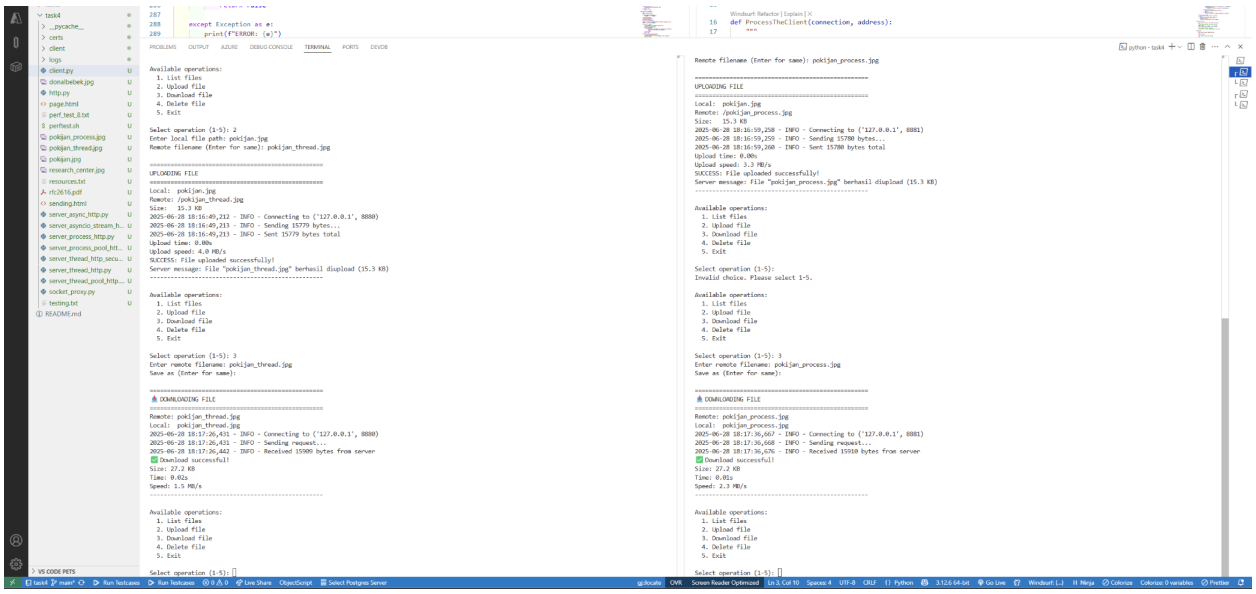
NABILAH ATIKA RAHMA
5025221005 - PROGJAR C
https://github.com/ranabel/network-programming-5025221005

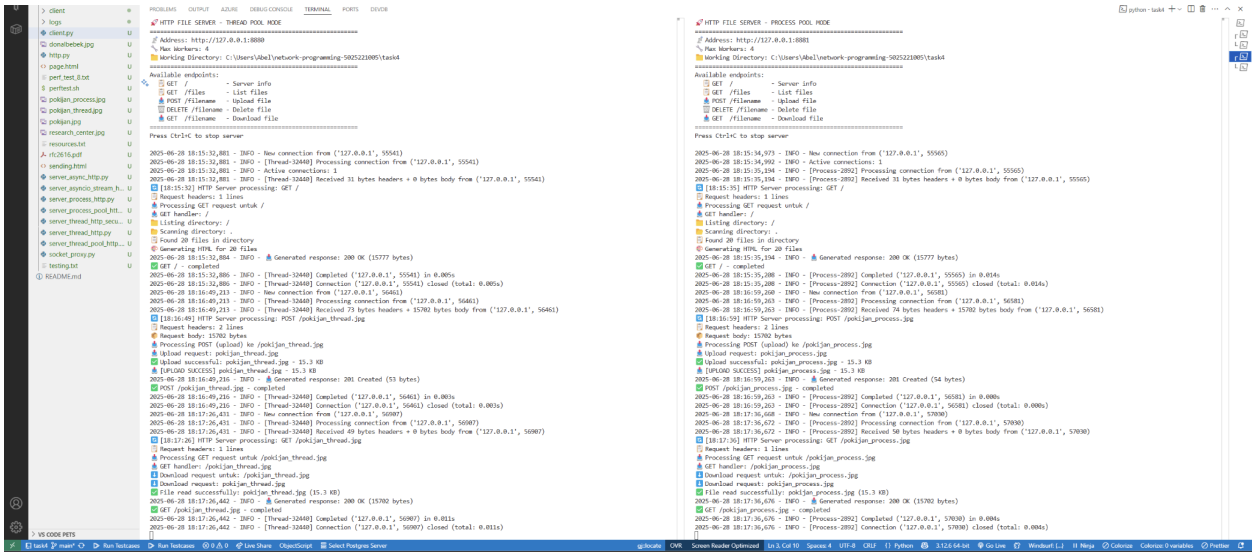




Pada gambar diatas ditunjukkan sisi client dan dibawah ditunjukkan di sisi server. Sedangkan pada gambar upload menunjukkan client berhasil mengupload file ke server menggunakan metode POST ditampilkan lengkap dengan ukuran file, kecepatan transfer, dan durasi. Server mencatat file yang diterima, ukurannya, dan lokasi penyimpanan. Client menerima respons sukses dari server.

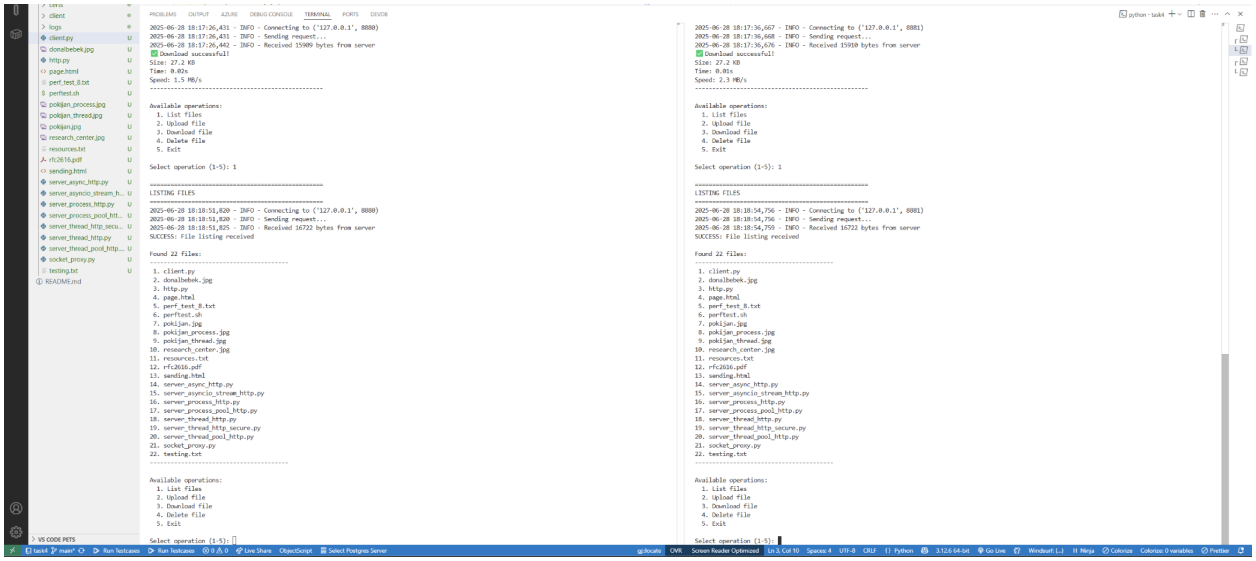
NABILAH ATIKA RAHMA
5025221005 - PROGJAR C
https://github.com/ranabel/network-programming-5025221005

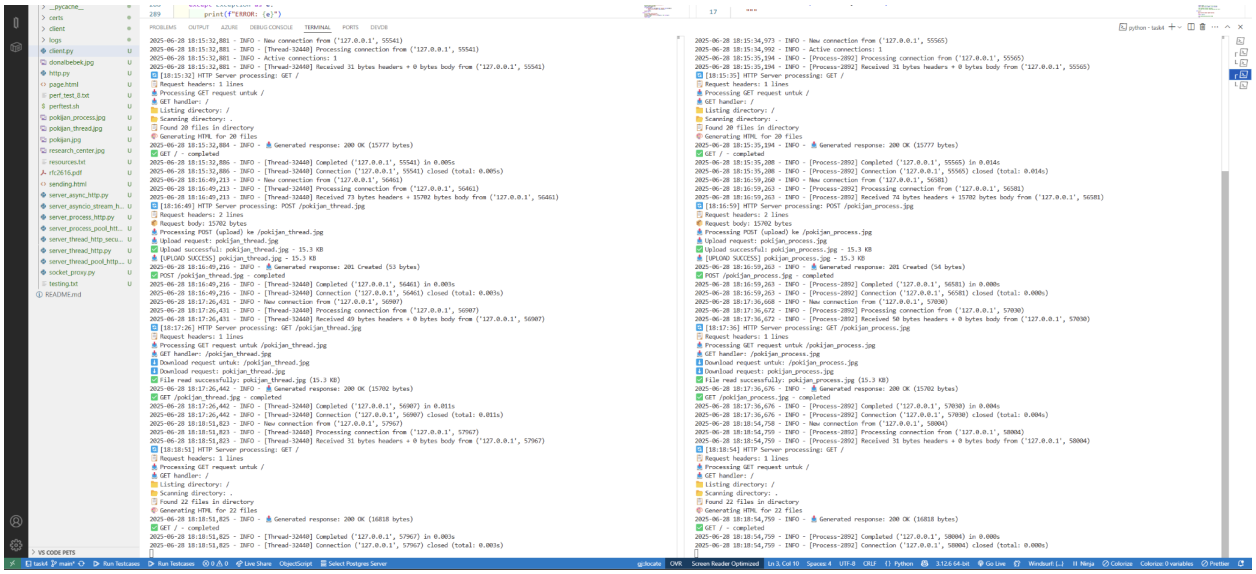




Pada gambar diatas ditunjukkan sisi client dan dibawah ditunjukkan di sisi server. Tidak jauh berbeda seperti upload yang menunjukkan client berhasil mengupload file ke server menggunakan metode POST ditampilkan lengkap dengan ukuran file, kecepatan transfer, dan durasi. Server mencatat file yang diterima, ukurannya, dan lokasi penyimpanan. Client menerima respons sukses dari server. Begitupun juga sama dengan download.

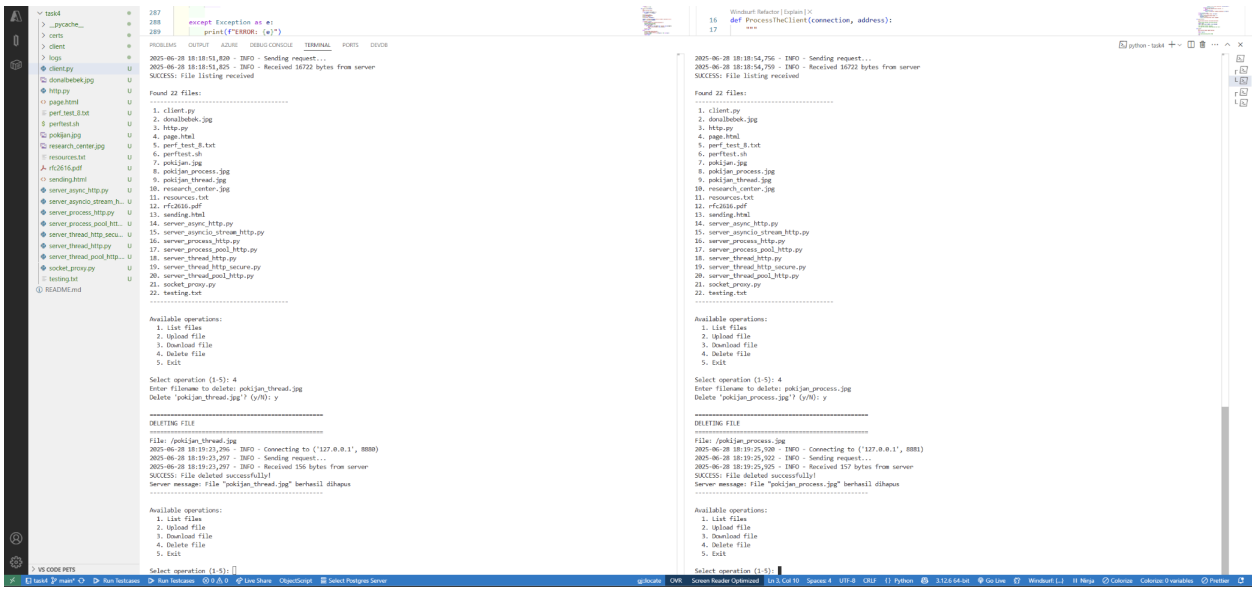
NABILAH ATIKA RAHMA
5025221005 - PROGJAR C
https://github.com/ranabel/network-programming-5025221005

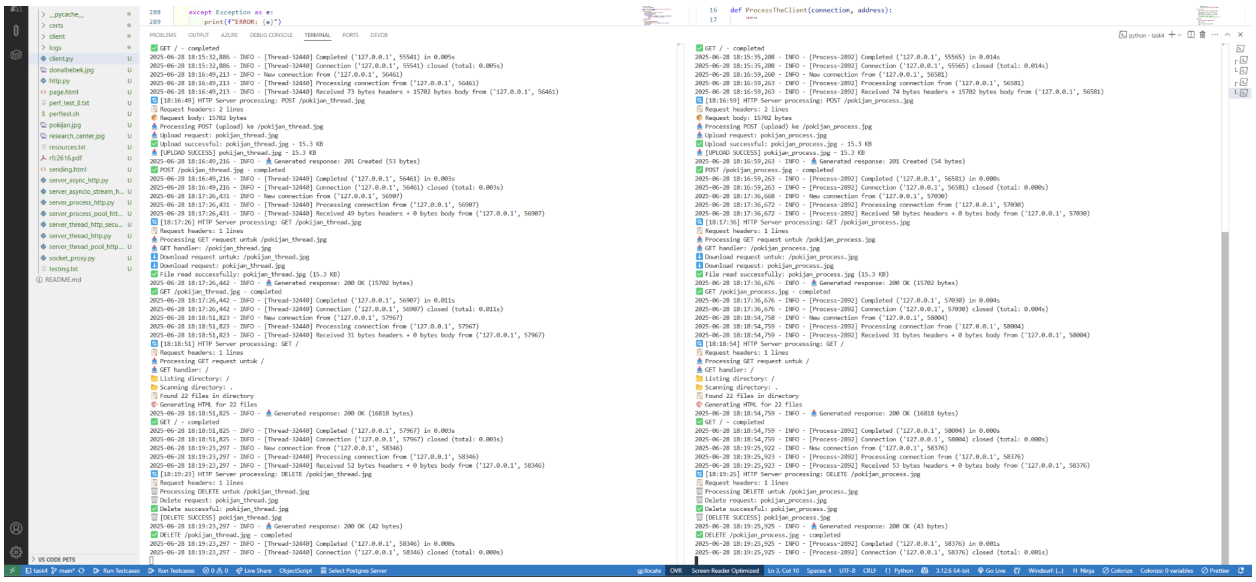




Pada gambar diatas ditunjukkan sisi client dan dibawah ditunjukkan di sisi server. Gambar diatas tidak jauh berbeda dengan listing diawal, namun ddisini saya memberikannya untuk membedakan setelah dan sebelum proses upload download, sehingga disini menambah 2 file. Function akan mencari dan akan ditampilkan listing files yang ada dalam directory saat ini (pada directory tersebut terdapat 22 files mulai dari client.py sampai testing.txt), disini juga dapat kita lihat kita sedang di port dan address mana.

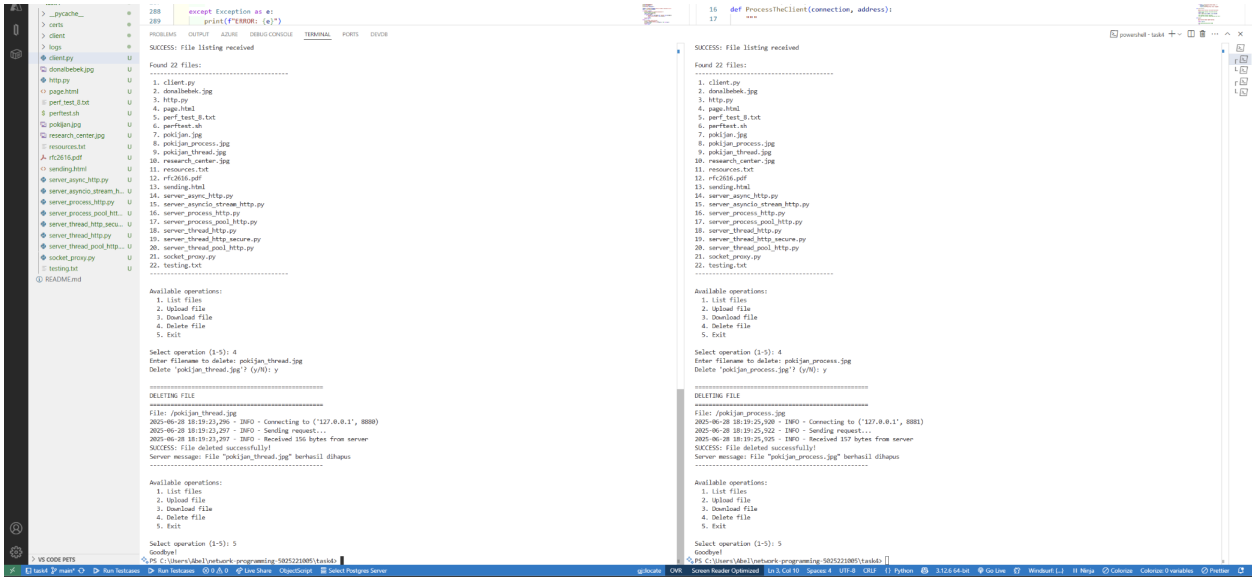
NABILAH ATIKA RAHMA
5025221005 - PROGJAR C
https://github.com/ranabel/network-programming-5025221005

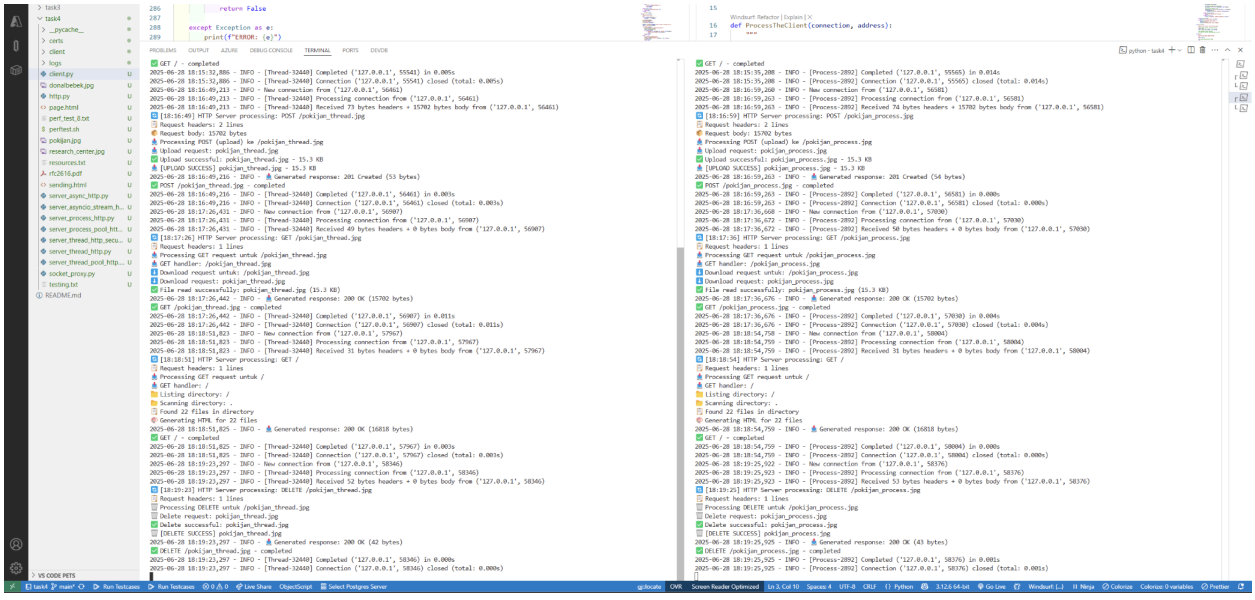




Pada gambar diatas ditunjukkan sisi client dan dibawah ditunjukkan di sisi server. Pada gambar diatas menunjukkan client menghapus file menggunakan metode DELETE. Sebelum menghapus, client meminta konfirmasi. Server memeriksa keberadaan file, lalu menghapusnya. Respons dari server menyatakan file berhasil dihapus, dan log mencatat waktu serta nama file yang dihapus.

NABILAH ATIKA RAHMA
5025221005 - PROGJAR C
https://github.com/ranabel/network-programming-5025221005





Pada gambar diatas ditunjukkan sisi client dan dibawah ditunjukkan di sisi server. Disini karena semua sudah dicoba maka ada menu nomer 5 yaitu exit. Jika client memilih exit maka akan say goodbye dan server akan close connection sebelumnya.

NABILAH ATIKA RAHMA
5025221005 - PROGJAR C
https://github.com/ranabel/network-programming-5025221005

Server HTTP ini juga memiliki beberapa fitur tambahan untuk meningkatkan kenyamanan dan keamanan penggunaan. Beberapa di antaranya adalah path validation, file size limits, dan input sanitization untuk mencegah penyalahgunaan. Server juga mencatat semua aktivitas melalui log, sehingga mudah dipantau. Dari sisi performa, saya menyesuaikan buffer dan koneksi agar server bisa menangani banyak request secara bersamaan. Penanganan error juga diperhatikan, misalnya untuk masalah jaringan, file tidak ditemukan, atau request yang tidak valid. Untuk client, saya menyediakan beberapa mode seperti interaktif, demo, dan command-line agar lebih fleksibel digunakan dan mudah diuji.