

Maximilian H M Lingga
05111940000092

TUGAS 4 PEMROGRAMAN JARINGAN

KELAS C

- Pada file server protocol (progjar4a), tambahkan kemampuan
 - Upload file
 - Content file yang diupload harus diencode dulu dengan format base64
 - Hapus file
- Update-lah spesifikasi protokol yang ada pada contoh dengan kemampuan yang baru ditambahkan tersebut, berikan penjelasan tambahan dalam satu paragraf
- 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 (satu repository)
 - Implementasi protokol
 - client
- Dokumen Update spesifikasi protokol
- Capture operasi client server untuk operasi upload dan hapus
- Deskripsi dan penjelasan minimal 1 paragraf

Link Github : <https://github.com/maxling92/Tugas-Progjar/tree/main/Tugas%204>

Pada tugas kali ini, akan ditambahkan fungsi untuk mengupload dan menghapus file.

Post (Upload File)

Untuk menjalankan command POST, akan ada beberapa fungsi dan parameter yang diperlukan. Untuk parameter diperlukan 2, yakni nama file dan isi file. Pada Server dan Client ditambahkan beberapa fungsi

Server

Pada server, ditambahkan fungsi post() untuk memposting file ke server. File yang diterima harus didecode terlebih dahulu dalam bentuk base64, lalu fungsi dijalankan dan server akan menunjukkan file yang diterima.

```

30     def post(self, params=[]):
31         try:
32             filename = params[0]
33             isifile = base64.b64decode(params[1])
34             if filename == '' or isifile == '':
35                 return None
36             with open(filename, 'wb+') as fp:
37                 fp.write(isifile)
38             return self.list()
39         except Exception as e:
40             return dict(status='ERROR', data=str(e))

```

Client

Pada client, ditambahkan fungsi `remote_post()`. Fungsi ini akan menerima nama file dan membaca file tersebut. Kemudian file akan di encode dalam base64. Setelah itu, Client akan mengirimkan nama dan isi file sebagai parameter ke server.

```

66 def remote_post(filename=""):
67     with open(filename, 'rb') as fp:
68         isifile = base64.b64encode(fp.read()).decode()
69         command_str=f"POST {filename} {isifile}\r\n\r\n"
70         # print('hasil command adalah', command_str) # untuk debugging
71         hasil = send_command(command_str)
72         if (hasil["status"] == 'OK'):
73             print("daftar file : ")
74             for nmfile in hasil['data']:
75                 print(f"- {nmfile}")
76             return True
77         else:
78             print("Gagal")
79             return False

```

DELETE (Menghapus File)

Untuk menjalankan command DELETE, akan digunakan 1 parameter, yakni nama file. Command ini akan menghapus file dengan nama yang dimaksud.

Server

Pada server, ditambahkan fungsi `delete()`. Pada fungsi ini, terdapat sub-fungsi `os.remove`, yang berfungsi untuk menghapus file. Untuk menjalankannya, nama file target akan diinput. Apabila proses gagal, ditambahkan pesan "error".

```

42     def delete(self, params=[]):
43         try:
44             filename = params[0]
45             os.remove(filename)
46             return self.list()
47         except Exception as e:
48             return dict(status='ERROR',data=str(e))

```

Client

Pada client, ditambahkan fungsi `remote_delete`. Fungsi ini akan menerima input nama file yang akan dihapus, lalu menjalankan string command delete. Command ini akan mengirim parameter ke server dan menerima pesan bahwa file sudah dihapus.

```

81 def remote_delete(filename=""):
82     command_str = f"DELETE {filename}\r\n\r\n"
83     hasil = send_command(command_str)
84     if (hasil["status"] == 'OK'):
85         print("daftar file : ")
86         for nmfile in hasil['data']:
87             print(f"- {nmfile}")
88         return True
89     else:
90         print("Gagal")
91         return False

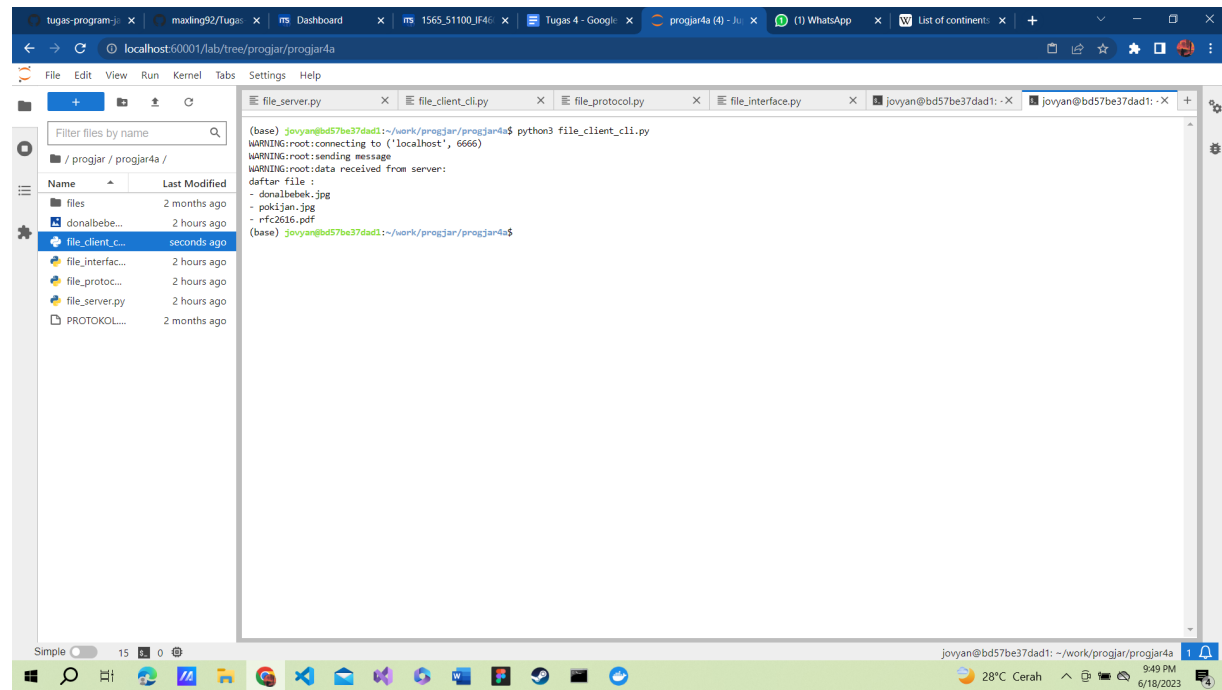
```

Operasi

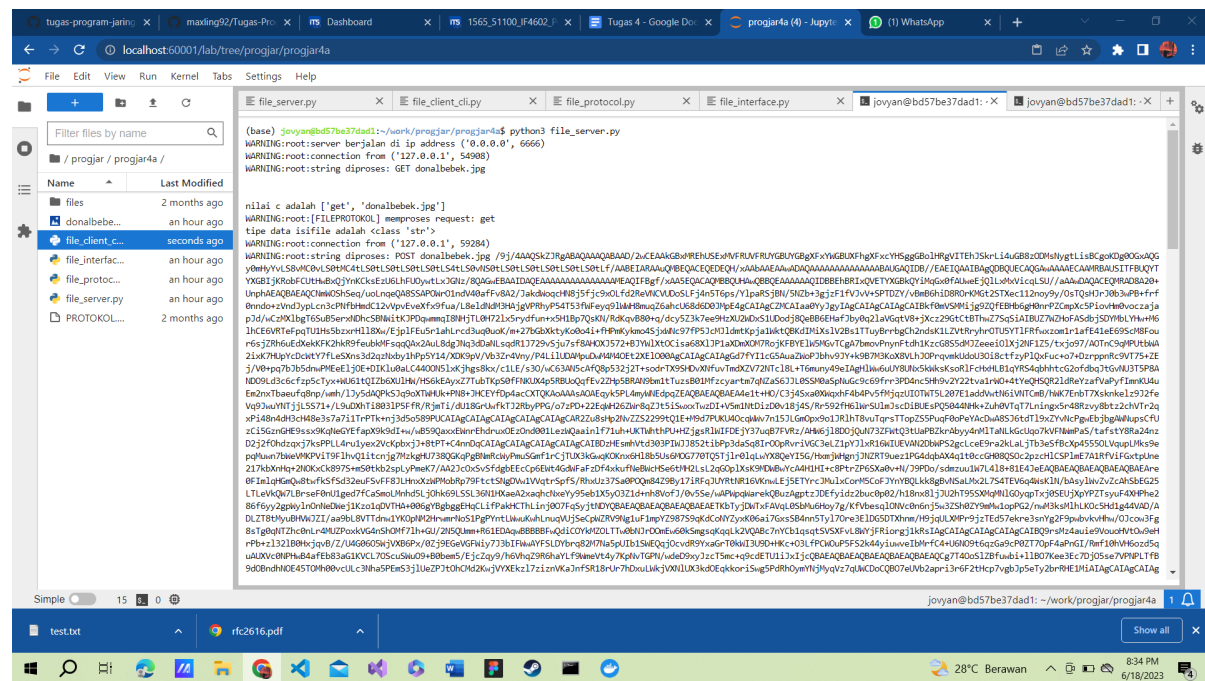
Berikut hasil dari penambahan fungsi **POST** dan **DELETE**.

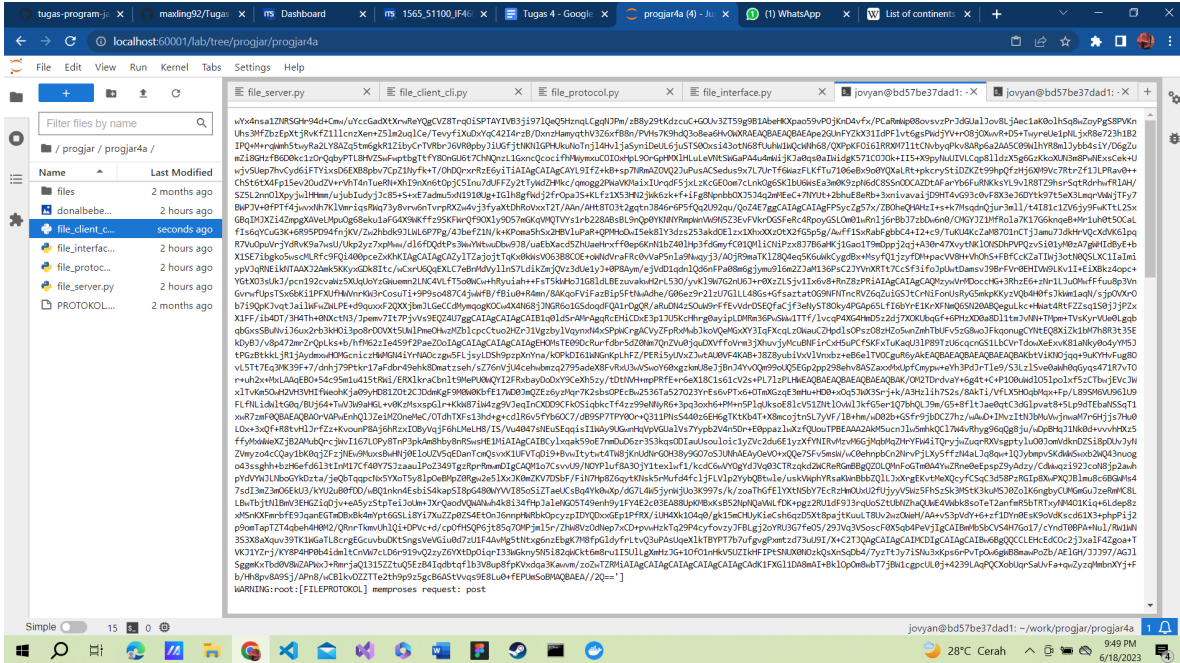
POST

Client



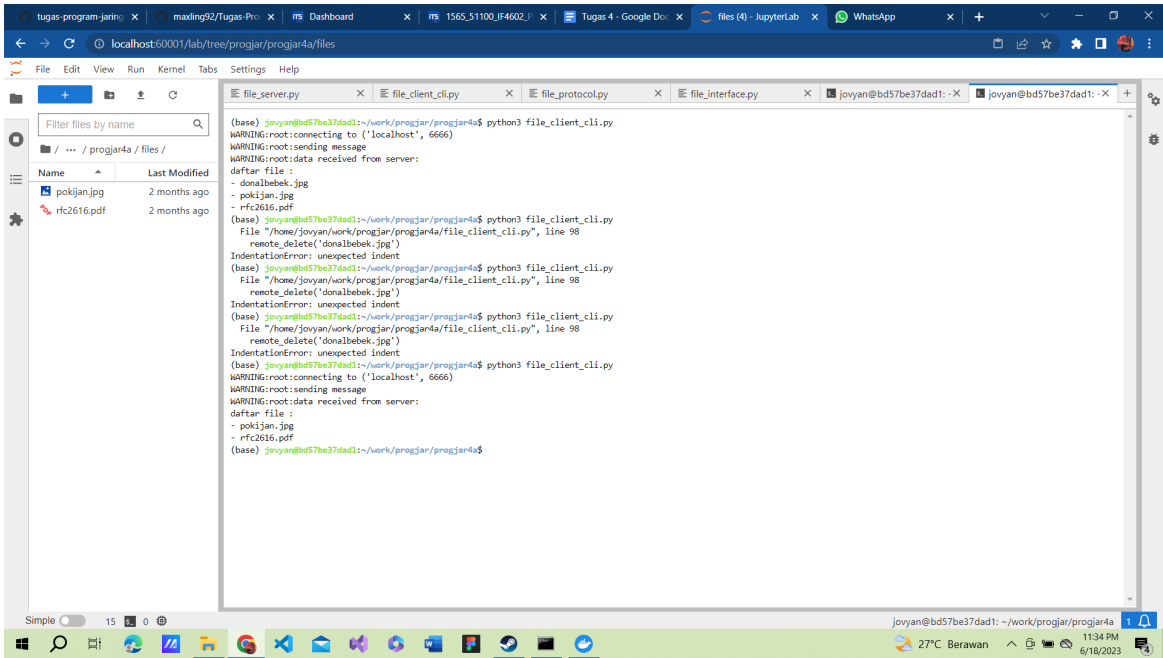
Server





DELETE

Client



Server

