

Tugas 4 Pemrograman Jaringan A

- Pada file server protocol (prograr4a), tambahkan kemampuan
 - **Upload file**
 - Pertama, buat fungsi baru pada “file_client_cli.py” Bernama “remote_upload” untuk membuat string command yang akan menjalankan fungsi pada “file_interface.py” dan menampung data pada “file_protocol.py”.

```
def remote_upload(filename=""):
    fp = open(filename, 'rb')
    isifile = base64.b64encode(fp.read()).decode()
    command_str = f"UPLOAD {filename} {isifile}"
    hasil = send_command(command_str)
    fp.close()
    if hasil['status'] == 'OK':
        print(hasil['data'])
        return True
    else:
        print(hasil)
        return False
```

Setelah membuka file yang diinput pada “fp” dan mengambil sekaligus meng-encode isi file menggunakan library “base64” pada “isifile”, kedua variable tersebut akan digabungkan dan dimasukkan ke dalam string “command_str” yang diawali dengan kata “UPLOAD”. Kata “UPLOAD” inilah yang akan menjalankan fungsi “upload” pada “file_interface.py” setelah data diinput ke dalam array “params” melewati “file_protocol.py”.

- Buat fungsi baru pada “file_interface.py” di dalam class “FileInterface” dengan code di bawah ini:

```
def upload(self, params=[]):
    try:
        filename = params[0]
        content = params[1]
        decoded_content = base64.b64decode(content).decode('utf-8')
        with open(filename, 'w') as file:
            file.write(decoded_content)
        return dict(status='OK', data=f"{filename} berhasil di-upload")
    except Exception as e:
        return dict(status='ERROR', data=str(e))
```

Fungsi ini mengikuti format cukup mirip dengan fungsi ‘get’. Namun karena array “params” ada 2 (param[0] nama file dan param[1] isi file yang ter-encode), maka dibuat kode untuk men-decode isi file pada “decoded_content” dan setelah itu dimasukkan ke dalam file yang sudah dimasukkan ke dalam folder “files” dengan write.

- Hapus file

- Pertama, buat fungsi baru pada “file_client_cli.py” Bernama “remote_delete” untuk membuat string command yang akan menjalankan fungsi pada “file_interface.py” dan menampung data pada “file_protocol.py”.

```
def remote_delete(filename=""):
    command_str = f"DELETE {filename}"
    hasil = send_command(command_str)
    if hasil['status'] == 'OK':
        print(hasil['data'])
        return True
    else:
        print("Gagal")
        return False
```

Karena hanya diminta untuk menghapus file, maka nama file bisa langsung ditampung ke dalam command string yang diawali dengan kata “DELETE” yang akan menjalankan fungsi “delete” pada “file_interface.py” setelah data diinput ke dalam array “params” melewati “file_protocol.py”.

- Buat fungsi baru pada “file_interface.py” di dalam class “FileInterface” dengan code di bawah ini:

```
def delete(self, params=[]):
    try:
        filename = params[0]
        os.remove(filename)
        return dict(status='OK', data=f"{filename} berhasil dihapus")
    except Exception as e:
        return dict(status='ERROR', data=str(e))
```

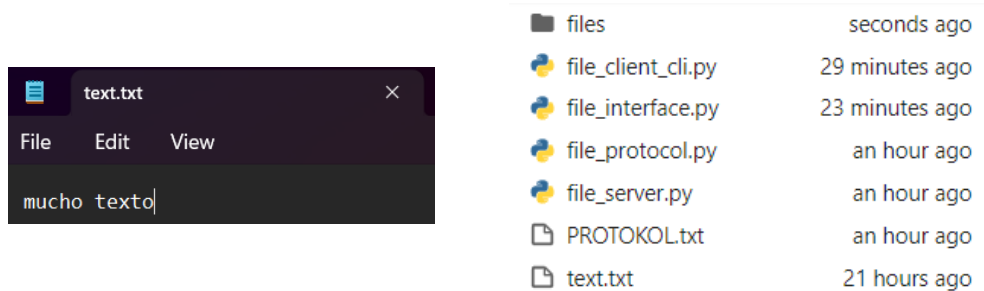
Setelah mengambil data params pada “file_protocol.py” (lebih tepatnya params[0] atau nama file), jalankan ‘os.remove(filename)’ untuk langsung menghapus file pada folder “files”.

- Update-lah spesifikasi protokol yang ada pada contoh dengan kemampuan yang baru ditambahkan tersebut, berikan penjelasan tambahan dalam satu paragraf
 - Berikut isi “PROTOKOL.txt” yang sudah di-update:

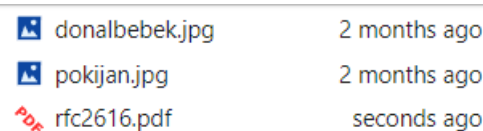
```
UPLOAD
* TUJUAN: untuk mengupload file ke dalam folder 'files'
* PARAMETER:
  - PARAMETER1 : nama file
  - PARAMETER2 : isi file
* RESULT:
- BERHASIL:
  - status: OK
  - data_namafile : nama file yang diminta
  - data : {filename} berhasil di-upload
- GAGAL:
  - status: ERROR
  - data: pesan kesalahan

DELETE
* TUJUAN: untuk menghapus file dari folder 'files'
* PARAMETER:
  - PARAMETER1 : nama file
* RESULT:
- BERHASIL:
  - status: OK
  - data_namafile : nama file yang diminta
  - data : {filename} berhasil dihapus
- GAGAL:
  - status: ERROR
  - data: pesan kesalahan
```

- Buatlah client implementation dari operasi tambahan tersebut. Jalankan operasi client server untuk kemampuan tersebut, berikanlah screenshot seperlunya, dan penjelasan dalam paragraf
 - Sebelum mengimplementasikan lewat client, buat text file baru pada directory utama untuk mencoba fungsi upload. Berikut isi text file dan tampilan awal directory utama:



- Berikut tampilan isi folder “files”



- Lalu jalankan fungsi-fungsi yang telah dibuat ke dalam fungsi main pada file “file_client_cli.py”.

```
if __name__ == '__main__':
    server_address=('0.0.0.0',6666)
    remote_get('donalbebek.jpg')
    remote_upload('text.txt')
    remote_delete('rfc2616.pdf')
    remote_list()
```

- Setelah melakukan step di atas, kita jalankan “file_server.py” pada terminal pertama.

```
(base) jovyan@62759db46c0b:~/work/progjar/progjar4a$ python3 file_server.py
WARNING:root:server berjalan di ip address ('0.0.0.0', 6666)
```

- Setelah terhubung pada port tersebut, jalankan “file_client_cli.py” pada terminal yang kedua.

```
(base) jovyan@62759db46c0b:~/work/progjar/progjar4a$ python3 file_client_cli.py
WARNING:root:connecting to ('0.0.0.0', 6666)
WARNING:root:sending message
WARNING:root:data received from server:
WARNING:root:connecting to ('0.0.0.0', 6666)
WARNING:root:sending message
WARNING:root:data received from server:
text.txt berhasil di-upload
WARNING:root:connecting to ('0.0.0.0', 6666)
WARNING:root:sending message
WARNING:root:data received from server:
rfc2616.pdf berhasil dihapus
WARNING:root:connecting to ('0.0.0.0', 6666)
WARNING:root:sending message
WARNING:root:data received from server:
daftar file :
- donalbebek.jpg
- pokijan.jpg
- text.txt
```

- Berikut output pada server yang muncul setelah menjalankan file di atas:

```
WARNING:root:connection from ('127.0.0.1', 54486)
WARNING:root:string diproses: GET donalbebek.jpg
WARNING:root:memproses request: get
WARNING:root:connection from ('127.0.0.1', 54488)
WARNING:root:string diproses: UPLOAD text.txt bXVjaG8gdGV4dG8=
WARNING:root:memproses request: upload
WARNING:root:connection from ('127.0.0.1', 54490)
WARNING:root:string diproses: DELETE rfc2616.pdf
WARNING:root:memproses request: delete
WARNING:root:connection from ('127.0.0.1', 54492)
WARNING:root:string diproses: LIST
WARNING:root:memproses request: list
```

- Dapat dilihat semua operasi berhasil dijalankan, berikut hasil dari operasinya masing-masing:




- **Upload**

- Berikut indicator operasi berhasil dijalankan:

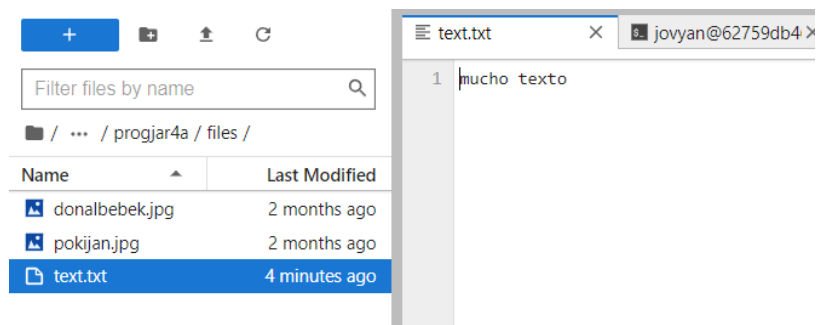
```
WARNING:root:data received from server:
text.txt berhasil di-upload
```

- Berikut tampilan file yang masuk ke dalam folder “files”:

📁 / ... / progjar4a / files /

Name	Last Modified
 donalbebek.jpg	2 months ago
 pokijan.jpg	2 months ago
 text.txt	3 minutes ago

- Jika dibuka, isi file berhasil di decode menggunakan UTF-8



- **Delete**

- Berikut indicator operasi berhasil dijalankan:

```
WARNING:root:data received from server:  
rfc2616.pdf berhasil dihapus
```

- Berikut tampilan directory utama setelah file tersebut dihapus:

📁 / ... / progjar / progjar4a /

Name	Last Modified
📁 files	7 minutes ago
🖼️ donalbebek.jpg	7 minutes ago
🔗 file_client_cli.py	13 minutes ago
🔗 file_interface.py	18 minutes ago
🔗 file_protocol.py	an hour ago
🔗 file_server.py	an hour ago
📄 PROTOKOL.txt	18 minutes ago
📄 text.txt	21 hours ago

- Link menuju ke repository di github (satu repository)

Link Repository: https://github.com/Ichlas02/Tugas4_Progjar_A