

## TUGAS 4 PROGRAM JARINGAN

1. Pada file server protocol (progjar4a), tambahkan kemampuan
  - a) Upload file
    - i. Content file yang diupload harus diencode dulu dengan format base64
  - b) Hapus file
2. Update-lah spesifikasi protokol yang ada pada contoh dengan kemampuan yang baru ditambahkan tersebut, berikan penjelasan tambahan dalam satu paragraf
3. Buatlah client implementation dari operasi tambahan tersebut. Jalankan operasi client server untuk kemampuan tersebut, berikanlah screenshot seperlunya, dan penjelasan dalam paragraf

### Solusi:

Akan akan ditambahkan kemampuan untuk upload file (POST) dan hapus file (DELETE).

### POST (Upload File)

Terdapat dua parameter dalam upload file, parameter 1 berisi nama file dan parameter kedua berisi isi file. Bila berhasil, hasil keluarannya berupa status berhasil dan list file yang ada pada server. Jika error, maka akan mengeluarkan status error dan data pesan kesalahannya.

### Server

Pada server, kita akan menambahkan fungsi post() untuk menambahkan file pada server. Isi file yang diterima dalam bentuk biner, sehingga harus didecode dengan format base64. Kemudian dilakukan penulisan file dengan menggunakan isi file yang telah diterima.

```
def post(self, params=[]):
    try:
        filename = params[0]
        isifile = base64.b64decode(params[1])
        if filename == '' or isifile == '':
            return None
        with open(filename, 'wb+') as fp:
            fp.write(isifile)
        return self.list()
    except Exception as e:
        return dict(status='ERROR', data=str(e))
```

Berikut ketika server menerima request POST dari client untuk upload file test.txt

```
(base) jovyan@99f84a575a90:~/work/progjar/latihan-tugas/tugas-4$ python3 file_server.py
WARNING:root:server berjalan di ip address ('0.0.0.0', 6666)
WARNING:root:connection from ('172.16.16.101', 57816)
WARNING:root:string diproses: POST test.txt SGFueWEgZmlsZSB0ZXN0IGJpYXNh
```

```
nilai c adalah ['post', 'test.txt', 'SGFueWEgZmlsZSB0ZXN0IGJpYXNh']
WARNING:root:[FILEPROTOKOL] memproses request: post
```

## Client

Pada client, akan ditambahkan fungsi `remote_post()`. Dengan menerima input nama file, client akan membaca file sesuai dengan input dalam mode biner, kemudian diencode dalam bentuk base64 serta di decode. Setelah itu, client akan mengirimkan nama file serta isi file tersebut ke dalam server sebagai parameter.

```
def remote_post(filename=""):
    with open(filename, 'rb') as fp:
        isifile = base64.b64encode(fp.read()).decode()
        command_str=f"POST {filename} {isifile}\r\n\r\n"
        # print('hasil command adalah', command_str) # untuk debugging
        hasil = send_command(command_str)
        if (hasil["status"] == 'OK'):
            print("daftar file : ")
            for nmfile in hasil['data']:
                print(f"- {nmfile}")
            return True
        else:
            print("Gagal")
            return False
```

Berikut ketika client menjalankan request POST dengan upload file test.txt

```
(base) jovyan@99f84a575a90:~/work/progjar/latihan-tugas/tugas-4$ python3 file_client_cli.py
WARNING:root:connecting to ('172.16.16.101', 6666)
WARNING:root:sending message
WARNING:root:data received from server:
daftar file :
- donalbebek.jpg
- pokijan.jpg
- rfc2616.pdf
- test.txt
```

## DELETE (Hapus file)

Untuk menghapus file, terdapat 1 parameter, yaitu berisi nama file. Hasil keluarannya jika proses berhasil dilakukan adalah status berhasil dan list file setelah file yang dimaksud dihapus. Jika gagal, maka akan mengeluarkan status error dan data pesan kesalahannya.

## Server

Akan dilakukan penghapusan file berdasarkan namafile yang telah diterima dari client dengan menggunakan `os.remove()`

```
def post(self, params=[]):
    try:
        filename = params[0]
        isifile = base64.b64decode(params[1])
        if filename == '' or isifile == '':
            return None
        with open(filename, 'wb+') as fp:
            fp.write(isifile)
        return self.list()
    except Exception as e:
        return dict(status='ERROR', data=str(e))
```

Berikut ketika server menerima request DELETE dari client untuk hapus file test.txt

```
WARNING:root:connection from ('172.16.16.101', 37434)
WARNING:root:string diproses: DELETE test.txt

nilai c adalah ['delete', 'test.txt']
WARNING:root:[FILEPROTOKOL] memproses request: delete
□
```

## Client

Command DELETE beserta parameter berupa nama file yang diinputkan untuk dihapus akan dikirim ke server, kemudian client akan menerima balasan dari server.

```
def remote_delete(filename=""):
    command_str = f"DELETE {filename}\r\n\r\n"
    hasil = send_command(command_str)
    if (hasil["status"] == 'OK'):
        print("daftar file : ")
        for nmfile in hasil['data']:
            print(f"- {nmfile}")
        return True
    else:
        print("Gagal")
        return False
```

Berikut ketika client menjalankan request DELETE untuk hapus file test.txt

```
WARNING:root:connecting to ('172.16.16.101', 6666)
WARNING:root:sending message
WARNING:root:data received from server:
daftar file :
- donalbebek.jpg
- pokijan.jpg
```

Link Github: <https://github.com/silabanjames/tugas-program-jaringan/tree/main/tugas4>