Modul 10
Python Pemrograman Jaringan

Topik Bahasan

- Pengenalan protokol HTTP dan FTP

Dalam pemrograman jaringan, **protokol aplikasi** adalah protokol yang berfungsi di lapisan aplikasi dari model OSI dan digunakan untuk berbagai jenis komunikasi antara aplikasi di jaringan. Beberapa protokol aplikasi yang umum digunakan termasuk HTTP, FTP, SMTP, dan banyak lagi.

Untuk memberikan gambaran konkret, mari kita lihat contoh penggunaan beberapa protokol aplikasi di Python:

1. **HTTP (Hypertext Transfer Protocol)**
2. **FTP (File Transfer Protocol)**

Kita akan membuat contoh server dan klien untuk setiap protokol, menjalankannya, dan membahas hasilnya.

## 1. HTTP dengan Flask

Flask adalah framework web Python yang sering digunakan untuk membuat server HTTP.

### a. Membuat Server HTTP dengan Flask

```
python
Copy code
# server_http.py
from flask import Flask, jsonify, request

app = Flask(__name__)

@app.route('/')
def home():
    return "Hello, World!"

@app.route('/api/data', methods=['GET'])
def get_data():
    return jsonify({'message': 'This is some data'})

if __name__ == '__main__':
    app.run(port=5000, debug=True)
```

### Penjelasan Kode:

- **`@app.route('/')`**: Endpoint dasar yang mengembalikan pesan "Hello, World!".

- `@app.route('/api/data', methods=['GET'])`: Endpoint yang mengembalikan data dalam format JSON.

### b. Menjalankan Server HTTP

Jalankan server dengan:

```
bash
Copy code
python server_http.py
```

Output akan menunjukkan bahwa server sedang berjalan:

```
csharp
Copy code
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

### c. Mengakses Server HTTP

Gunakan `requests` untuk mengakses API ini:

```
python
Copy code
# client_http.py
import requests

# Mengakses endpoint dasar
response = requests.get('http://127.0.0.1:5000/')
print("Home Page:", response.text)

# Mengakses endpoint data
response = requests.get('http://127.0.0.1:5000/api/data')
print("API Data:", response.json())
```

### Hasil Running dan Pembahasan:

Jalankan `client_http.py`, dan hasil output akan terlihat seperti ini:

```
css
Copy code
Home Page: Hello, World!
API Data: {'message': 'This is some data'}
```

### Penjelasan Output:

- **Home Page:** Menampilkan teks dari endpoint root.
- **API Data:** Menampilkan data JSON dari endpoint `/api/data`.

## 2. FTP dengan Python

FTP (File Transfer Protocol) digunakan untuk transfer file antara server dan klien.

### a. Membuat Server FTP dengan Python

Gunakan pustaka `pyftpdlib` untuk membuat server FTP. Instal terlebih dahulu:

```
bash
Copy code
pip install pyftpdlib
```

Buat server FTP:

```
python
Copy code
# server_ftp.py
from pyftpdlib.authorizers import DummyAuthorizer
from pyftpdlib.handlers import FTPHandler
from pyftpdlib.servers import FTPServer

# Setup authorizer
authorizer = DummyAuthorizer()
authorizer.add_user("user", "12345", "/path/to/ftp", perm="elradfmwMT")
authorizer.add_anonymous("/path/to/ftp")

# Setup handler
handler = FTPHandler
handler.authorizer = authorizer

# Setup server
server = FTPServer(("127.0.0.1", 21), handler)
server.serve_forever()
```

### Penjelasan Kode:

- **`DummyAuthorizer`**: Digunakan untuk mengatur otentikasi pengguna.
- **`FTPHandler`**: Menangani permintaan FTP.
- **`FTPServer`**: Menyediakan server FTP yang berjalan pada alamat dan port yang ditentukan.

### b. Menjalankan Server FTP

Jalankan server dengan:

```
bash
Copy code
python server_ftp.py
```

### c. Mengakses Server FTP

Gunakan pustaka `ftplib` untuk mengakses server FTP:

```
python
Copy code
# client_ftp.py
from ftplib import FTP

ftp = FTP('127.0.0.1')
ftp.login(user='user', passwd='12345')

# List directory contents
ftp.retrlines('LIST')

# Upload a file
with open('test.txt', 'rb') as file:
    ftp.storbinary('STOR test.txt', file)

# Download a file
with open('downloaded_test.txt', 'wb') as file:
    ftp.retrbinary('RETR test.txt', file.write)

ftp.quit()
```

**Hasil Running dan Pembahasan:**

Jalankan `client_ftp.py`, dan hasil output dari list direktori akan terlihat seperti ini:

```
sql
Copy code
drwxr-xr-x    2 user     user          4096 Aug 26 12:00 .
drwxr-xr-x    2 user     user          4096 Aug 26 12:00 ..
-rw-r--r--    1 user     user            12 Aug 26 12:00 test.txt
```

**Penjelasan Output:**

- **List directory contents:** Menampilkan daftar file dan direktori di server FTP.
- **Upload and Download:** Menunjukkan bahwa file berhasil diunggah dan diunduh dari server FTP.

## Kesimpulan

- **HTTP (Flask):** Flask digunakan untuk membuat server HTTP dan API, dan `requests` digunakan untuk mengonsumsi API tersebut. Ini adalah metode yang umum untuk komunikasi berbasis web.
- **FTP:** `pyftpdlib` digunakan untuk membuat server FTP, dan `ftplib` digunakan untuk berinteraksi dengan server FTP untuk transfer file.