



Institut Teknologi Del

10S2202 Jaringan Komputer
Semester Genap 2021/2022

Praktikum Week 12 Socket Programming II

Membangun web server sederhana menggunakan socket programming dengan java.

Sebelum mengimplementasikan aplikasi web server dengan menggunakan java, terlebih dahulu kita mengerti konsep dasar bagaimana cara kerja HTTP dan bagaimana hubungannya dengan socket programming.

Seperti yang telah dibahas sebelumnya, ketika sebuah browser memberikan request HTTP ke sebuah web server, maka yang diberikan adalah request berupa clear text melalui socket TCP. Dalam praktikum ini, kita hanya fokus pada baris pertama setiap request HTTP.

```
GET <PATH> HTTP/1.1
```

Contoh:

```
GET /gambar.jpg HTTP/1.1
```

Jika dihubungkan dengan socket programming, maka yang kita lakukan adalah:

1. Membuat sebuah socket server
2. Membaca baris pertama dari data yang dikirimkan oleh client, yang pastinya sesuai dengan format diatas.
3. Membaca method request (kita hanya handle method GET)
4. Membaca path file yang direquest oleh client (kita hanya handle request file jpg dan gif)
5. Membaca file tersebut dari file system dan memberikannya ke client melalui socket TCP.
6. Menutup koneksi TCP (HTTP).

Berikut ini adalah kode program untuk melakukan hal diatas (WebServer.java).

```
import java.io.*;
import java.net.*;
import java.util.*;
class WebServer{
    public static void main(String argv[]) throws Exception {
        String requestMessageLine;
        String fileName;
```

```

        ServerSocket listenSocket = new ServerSocket(6789);
        Socket connectionSocket = listenSocket.accept();
        BufferedReader inFromClient = new BufferedReader(new
InputStreamReader(connectionSocket.getInputStream()));
        DataOutputStream outToClient = new
DataOutputStream(connectionSocket.getOutputStream());
        requestMessageLine = inFromClient.readLine();
        StringTokenizer tokenizedLine = new
StringTokenizer(requestMessageLine);

        if (tokenizedLine.nextToken().equals("GET")){
            fileName = tokenizedLine.nextToken();
            if (fileName.startsWith("/") == true )
                fileName = fileName.substring(1);

            File file = new File(fileName);
            int numBytes = (int) file.length();
            FileInputStream inFile = new FileInputStream
(fileName);

            byte[] fileInBytes = new byte[numBytes];
            inFile.read(fileInBytes);
            outToClient.writeBytes("HTTP/1.0 200 Document
Follows\r\n");

            if (fileName.endsWith(".jpg"))
                outToClient.writeBytes("Content-Type:
image/jpeg\r\n");
            if (fileName.endsWith(".gif"))
                outToClient.writeBytes("Content-Type:
image/gif\r\n");
            outToClient.writeBytes("Content-Length: " + numBytes
+ "\r\n");
            outToClient.writeBytes("\r\n");
            outToClient.write(fileInBytes, 0, numBytes);

            connectionSocket.close();
        }else
            System.out.println("Bad Request Message");
    }
}

```

Untuk melakukan percobaan program diatas, lakukan langkah berikut:

1. Compile program tersebut dengan menggunakan perintah #javac WebServer.java
2. Jalankan aplikasi yang telah di-compile #java WebServer
3. Buatlah sebuah file jpg pada direktori yang sama dengan aplikasi anda (cth. gambar.jpg).
4. Bukalah browser anda dan akses file gambar pada web server anda (cth: <http://127.0.0.1:6789/gambar.jpg>) Akses HTTP menggunakan port 6789
5. Browser anda akan menampilkan gambar yang ada pada webserver dan aplikasi server akan berhenti.

Seperti yang anda coba, web server tersebut akan berhenti setiap kali selesai melayani sebuah request.

Tugas:

Dengan menggunakan teknik pada praktikum I,

1. Modifikasilah program tersebut sehingga aplikasi bisa melayani beberapa request yang sequential (tidak berhenti setelah melayani sebuah request)
2. Modifikasilah program tersebut dengan menambahkan fungsi thread sehingga bisa melayani beberapa request secara paralel
3. Modifikasilah program tersebut sehingga bisa melayani request terhadap file .html, .css dan .js.
4. Buatlah contoh dimana sebuah file html (berserta css, dan javascript) bisa diakses oleh client menggunakan browser.

Selamat Mengerjakan.