**LAPORAN TUGAS REMOTE INVOCATION**

*Dibuat untuk memenuhi Tugas Mata Kuliah Sistem Terdistribusi dan Paralel*



Disusun Oleh:

|  |  |
| --- | --- |
| 1301160369 | Faishol Muzaky |
| 1301160609 | Ghalih Mandaveqia |
| 1301164196 | Nabila Kamil |
| 1301160439 | Tubagus Rifky |

IF – 40 – 09

**S1 Informatika**

**Fakultas Informatika**

**Universitas Telkom**

**Bandung**

**2019**

**Web Server**

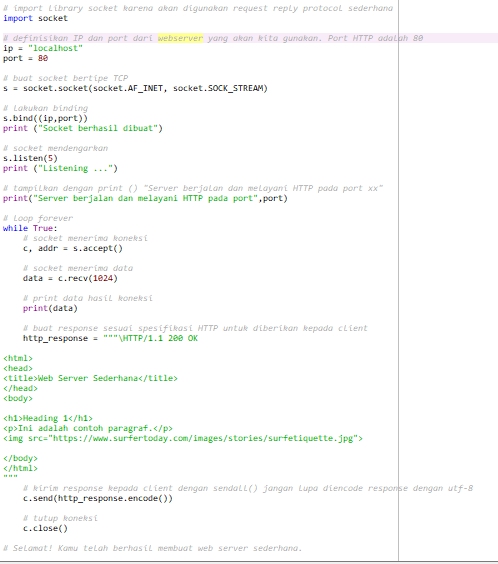
Cara kerja :

Browser meminta data web page kepada server, maka instruksi permintaan data oleh browser tersebut akan dikemas di dalam TCP yang merupakan protokol transport dan dikirim ke alamat yang dalam hal ini merupakan protokol berikutnya yaitu HTTP atau HTTPS.

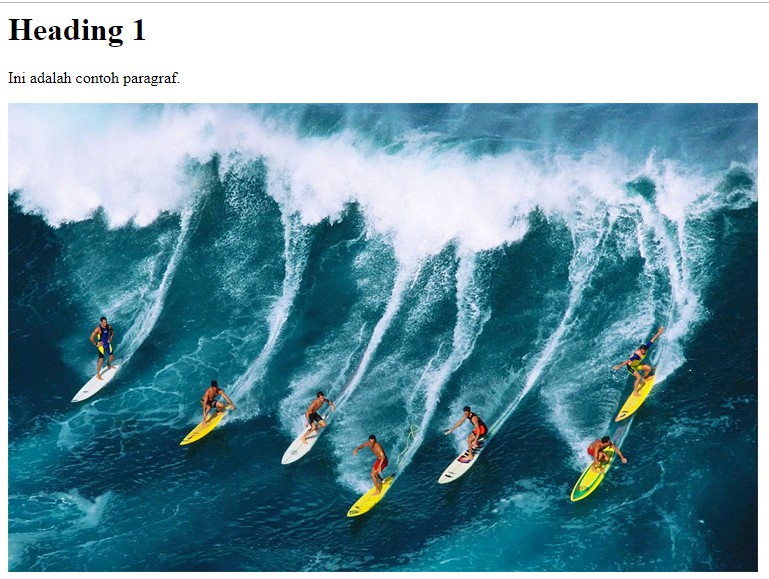
Data yang diminta dari browser ke web server disebut dengan HTTP request. Jika ditemukan, data tersebut akan dikemas oleh web server dalam TCP dan dikirim kembali ke browser untuk ditampilkan.

Data yang dikirim dari server ke browser disebut dengan HTTP response. Jika data yang diminta oleh browser tersebut tidak ditemukan oleh web server, maka web server akan menolak permintaan tersebut dan browser akan menampilkan notifikasi Page Not Found atau Error 404.

Source code :



Output :



**Voting**

Cara kerja :

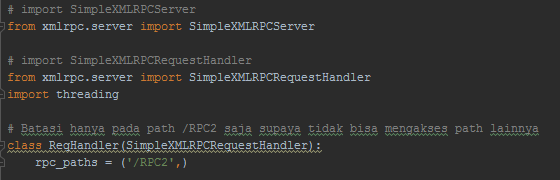
Remote Procedure Call (RPC) adalah sebuah metode yang memungkinkan kita untuk mengakses sebuah prosedur yang berada di komputer lain. Pada kasus voting ini, untuk dapat melakukan voting pertama sebuah server harus menyediakan layanan remote procedure. Hal dilakukan adalah sebuah server membuat socket yang berisikan ip dan port yang akan digunakan client untuk dapat terhubung dengan server, lalu menunggu client yang meminta prosedur yang disediakan oleh server.

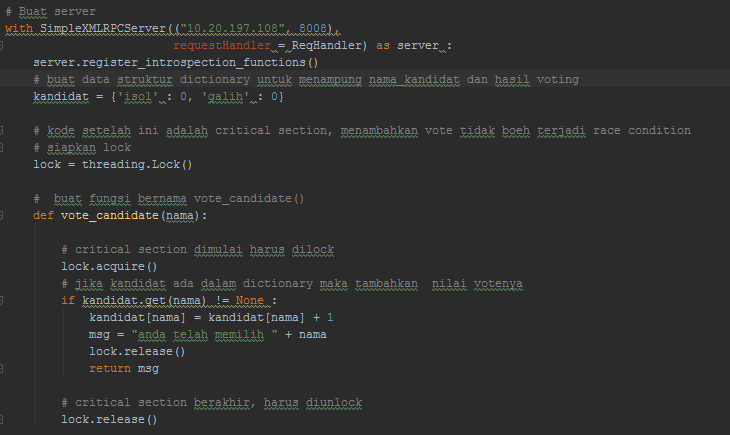
Source code :

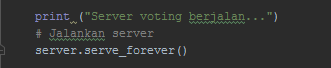
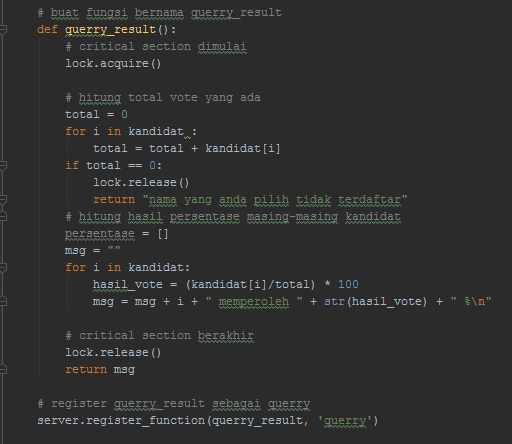
Client



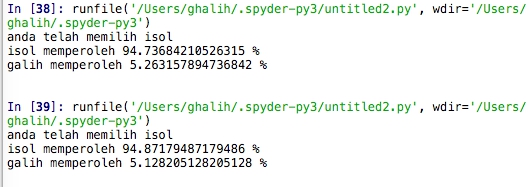
Server







Hasil



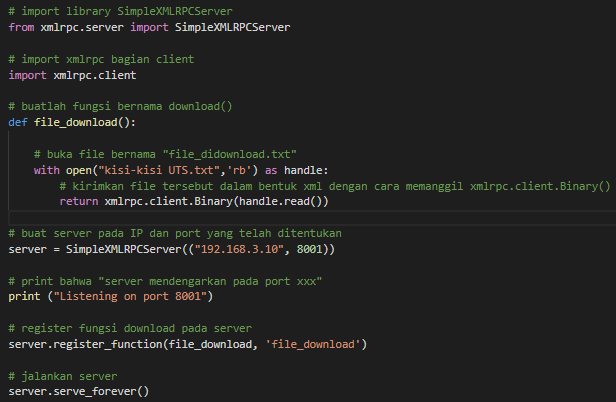
**Download**

Cara kerja :

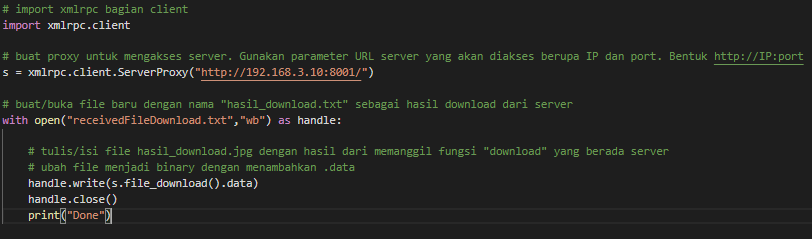
Pada kasus file transfer (download) ini, pertama sebuah server harus menyediakan fungsi atau layanan file transfer (download). Lalu server membuat socket yang berisikan ip dan port yang akan digunakan client untuk dapat terhubung dengan server, lalu server menunggu client meminta fungsi atau layanan file transfer (download) yang disediakan oleh server. Ketika file berhasil di akses (client) maka file tersebut akan ditulis ulang kedalam receivedFileDownload.txt.

Source code :

* Server

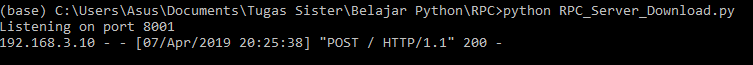


* Client



Output :

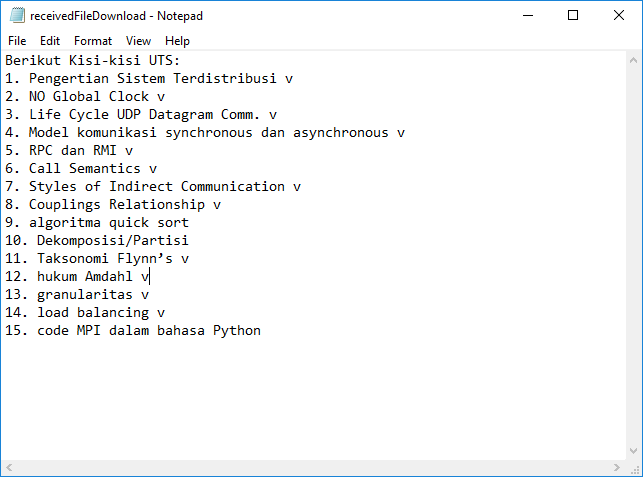
Server



Client



Isi file receivedFileDownload.txt



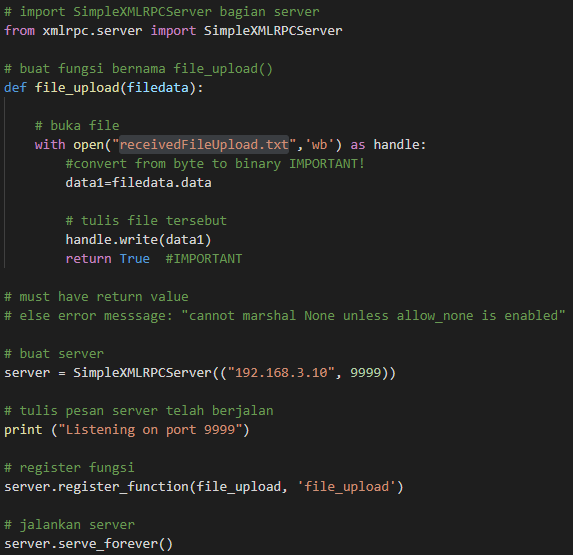
**Upload**

Cara kerja :

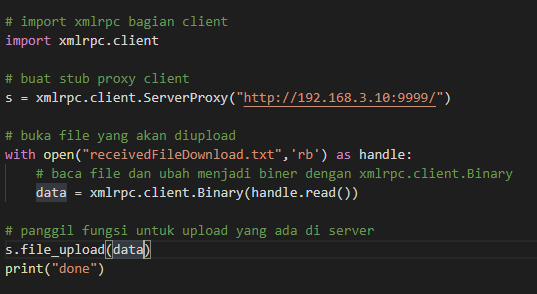
Pada kasus file transfer (upload) ini, pertama sebuah server harus menyediakan fungsi atau layanan file transfer (upload). Lalu server membuat socket yang berisikan ip dan port yang akan digunakan client untuk dapat berkomunikasi kepada server, lalu server menunggu client meminta fungsi atau layanan file transfer (upload) yang disediakan oleh server. Ketika client telah menentukan data yang akan dikirim (receivedFileDownload.txt) ke server maka, client akan memanggil fungsi atau layanan yang disediakan oleh server untuk mengirimkan file tersebut ke server dan server akan menulis ulang file yang dikirim ke dalam file yang bernama receivedFileUpload.txt.

Souce code :

* Server



* Client



Output

* Server



* Client



Isi file receivedFileUpload.txt

