

Dokumentasi Reverse Proxy

Oleh:

Delia Risti N. 5114100033

Tionia Rizkika 5114100053

Aditya Gunawan 5114100167

Luqman Ahmad 5114100187

Daftar Isi

[Pendahuluan 3](#_Toc469985372)

[Latar Belakang 3](#_Toc469985373)

[Rancangan Server 4](#_Toc469985374)

[A. FITUR 4](#_Toc469985375)

[1. Register 4](#_Toc469985376)

[2. Login 4](#_Toc469985377)

[3. Exit 5](#_Toc469985378)

[4. Private Message 5](#_Toc469985379)

[5. Check Ourself 5](#_Toc469985380)

[6. Check Online Friend 5](#_Toc469985381)

[7. Broadcast Message 6](#_Toc469985382)

[8. Checkpast 6](#_Toc469985383)

[9. Check 6](#_Toc469985384)

[10. Logout 6](#_Toc469985385)

[11. Create Group 6](#_Toc469985386)

[12. Join Group 7](#_Toc469985387)

[13. Send Message to Group 7](#_Toc469985388)

[B. SOURCE CODE 8](#_Toc469985389)

[C. SCREENSHOT SERVER 17](#_Toc469985390)

[D. PEMBAGIAN TUGAS 21](#_Toc469985391)

[Rancangan Client dari Server Kelompok lain 22](#_Toc469985392)

[A. FITUR 22](#_Toc469985393)

[1. Sign Up 22](#_Toc469985394)

[2. Login 22](#_Toc469985395)

[3. Chat Personal 23](#_Toc469985396)

[4. Buat Group 23](#_Toc469985397)

[5. Gabung Group 24](#_Toc469985398)

[6. Leave Group 24](#_Toc469985399)

[7. Chat Group 25](#_Toc469985400)

[8. Daftar User yang Aktif 25](#_Toc469985401)

[9. Daftar Group yang Ada 25](#_Toc469985402)

[10. Logout 26](#_Toc469985403)

[B. SOURCE CODE 26](#_Toc469985404)

[C. SCREENSHOT CLIENT 36](#_Toc469985405)

[D. PEMBAGIAN TUGAS 39](#_Toc469985406)

# Pendahuluan

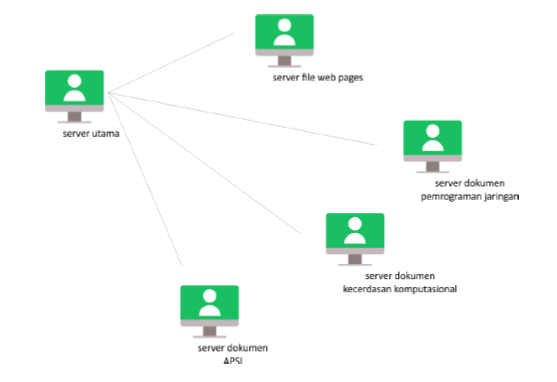
## Latar Belakang

Adanya tugas membuat reverse proxy dalam mata kuliah Pemrograman Jaringan kelas E.

# Rancangan

Simple HTTP Server ini dirancang untuk memenuhi permintaan dokumen-dokumen seputar perkuliahan. Reverse proxy membagi kerja server untuk menangani dokumen sesuai dengan mata kuliah yang telah ditentukan.

## ARSITEKTUR SISTEM



Server yang terlibat dalam arsitektur ini terdiri atas:

1. Server utama yang membagi kerja server-server di bawahnya.
2. Server tier-1
   1. Sebuah server yang menangani jenis request
   2. Tiga buah server untuk menangani permintaan dokumen

Dokumen-dokumen yang dapat ditangani oleh server-server tersebut adalah dokumen mata kuliah yang terdiri atas:

* Dokumen APSI
* Dokumen KK
* Dokumen Progjar

## SOURCE CODE

nama file : http\_server.py

|  |  |
| --- | --- |
|  | import socket  import sys |
|  | import threading |
|  |  |
|  | #inisialisasi |
|  | sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) |
|  |  |
|  | #proses binding |
|  | server\_address = ('localhost', 13006) |
|  | print >>sys.stderr, 'starting up on %s port %s' % server\_address |
|  | sock.bind(server\_address) |
|  |  |
|  | #listening |
|  | sock.listen(1) |
|  |  |
|  |  |
|  | def response\_teks(): |
|  | hasil = "HTTP/1.1 200 OK\r\n" \ |
|  | "Content-Type: text/plain\r\n" \ |
|  | "Content-Length: 7\r\n" \ |
|  | "\r\n" \ |
|  | "PROGJAR" |
|  | return hasil |
|  |  |
|  | def response\_gambar(): |
|  | filegambar = open('gambar.png','r').read() |
|  | panjang = len(filegambar) |
|  | hasil = "HTTP/1.1 200 OK\r\n" \ |
|  | "Content-Type: image/png\r\n" \ |
|  | "Content-Length: {}\r\n" \ |
|  | "\r\n" \ |
|  | "{}" . format(panjang, filegambar) |
|  | return hasil |
|  |  |
|  | def response\_page(url): |
|  | filename = url.split("/") |
|  | filename = filename[1] |
|  | print "filename : "+filename |
|  |  |
|  | filename = "pages/"+filename |
|  |  |
|  | try: |
|  | webfile = open(filename, 'r').read() |
|  | except: |
|  | webfile = open('pages/not\_found.html').read() |
|  |  |
|  | length = len(webfile) |
|  | hasil = "HTTP/1.1 200 OK\r\n" \ |
|  | "Content-Type: text/html;charset=UTF-8" \ |
|  | "Content-Length: {}\r\n" \ |
|  | "\r\n" \ |
|  | "{}" . format(length, webfile) |
|  | return hasil |
|  |  |
|  | def response\_document(url): |
|  | n = len(url) |
|  | filename = url[1:n] |
|  | print "filename : "+filename |
|  |  |
|  | docfile = open(filename, 'r').read() |
|  |  |
|  | length = len(docfile) |
|  | hasil = "HTTP/1.1 200 OK\r\n" \ |
|  | "Content-Type: application/pdf" \ |
|  | "Content-Length: {}\r\n" \ |
|  | "\r\n" \ |
|  | "{}" . format(length, docfile) |
|  | return hasil |
|  |  |
|  | #fungsi melayani client |
|  | def layani\_client(koneksi\_client,alamat\_client): |
|  | try: |
|  | print >>sys.stderr, 'ada koneksi dari ', alamat\_client |
|  | request\_message = '' |
|  | while True: |
|  | data = koneksi\_client.recv(64) |
|  | data = bytes.decode(data) |
|  | request\_message = request\_message+data |
|  | if (request\_message[-4:]=="\r\n\r\n"): |
|  | break |
|  |  |
|  |  |
|  | baris = request\_message.split("\r\n") |
|  | baris\_request = baris[0] |
|  | print "baris request[0] : "+baris\_request |
|  |  |
|  | a,url,c = baris\_request.split(" ") |
|  | print "url : "+url |
|  |  |
|  | ekstensi = url.split(".") |
|  | ekstensi = ekstensi[1] |
|  |  |
|  | respon = "" |
|  | if (ekstensi=='html'): |
|  | respon = response\_page(url) |
|  | else: |
|  | respon = response\_document(url) |
|  | koneksi\_client.send(respon) |
|  | finally: |
|  | # Clean up the connection |
|  | koneksi\_client.close() |
|  |  |
|  |  |
|  | while True: |
|  | # Wait for a connection |
|  | print >>sys.stderr, 'waiting for a connection' |
|  | koneksi\_client, alamat\_client = sock.accept() |
|  | s = threading.Thread(target=layani\_client, args=(koneksi\_client,alamat\_client)) |
|  | s.start() |
|  |  |
|  |  |

## SCREENSHOT