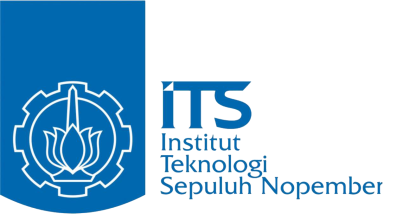
****

**LAPORAN TUGAS REVERSE PROXY**

**PEMROGRAMAN JARINGAN E**

**OLEH :**

**KELOMPOK 5**

**RAHMA DINI MAGHFIROTUL L. (5114100027)**

**I GEDE PUTU NOBBY A. P. (5114100048)**

**NURHAMIDAH TYAS P. (5114100057)**

**ADENUAR PURNOMO (5114100079)**

**IRZAL AHMAD S. (5114100123)**

1. DESKRIPSI

EMPITRI merupakan aplikasi web yang dapat digunakan untuk searching, download, streaming musik.

1. FITUR
   1. MEMBERSHIP

Untuk menggunakan aplikasi ini user harus memiliki akun terlebih dahulu yaitu dengan cara register dan login dengan akun yang telah dibuat

Register: bagi user yang belum memiliki akun di empitri

Login: bagi user yang telah memiliki akun di empitri (sudah melakukan registrasi)

* 1. SEARCHING LAGU

Fitur ini memungkinkan user untuk mencari lagu berdasarkan Artis, Judul Lagu, Genre, dan Album.

* 1. DOWNLOAD LAGU

Fitur yang memungkinkan user untuk mengunduh lagu

* 1. STREAMING LAGU

Fitur yang memungkinkan user untuk mendengarkan lagu tanpa harus mengunduh

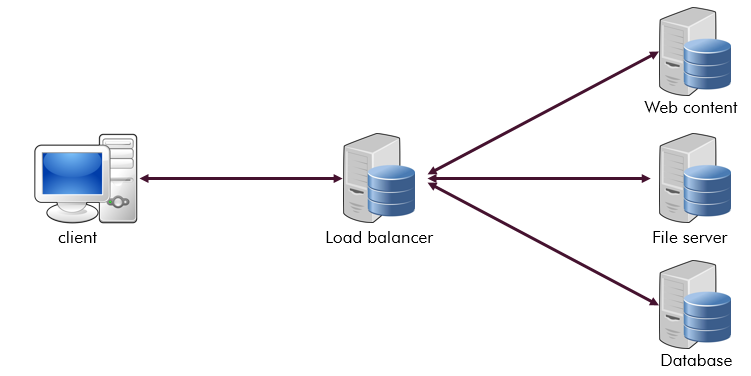
1. PENJELASAN REVERSE PROXY

reverse proxy berada di depan web server. atau bahasa kasarnya cache pada web.

seperti contoh kita mengakses google.com, situs ini adalah web server yang paling sering dikunjungi, untuk perhitungan kasar bahwa setiap request google sebesar 4 kb, dan dimasukan kedalam bandwith. seperti yang kita tau bandwith menuju server google juga ada batasnya. jika kita mengakses google bersama2 dgn 1 jaringan,maka akan ada trafic yang sangat besar/overload. Maka dari itu diciptakanya reverse proxy, sehingga yang mengakses google cukup serve proxy. sedangkan client akan ditangani cache reverse proxy. untuk akses yang lainya akan ditangani oleh load balance/forward proxy. dengan konsep seperti ini maka overload pada bandwith lbh efisien.

kegunaan ke dua adalah masalah keamanan. sbenarnya keamanan ini ditanggung oleh proxy. namun reverse proxy juga mengambil peran, dikarenakan kita/user untuk mengakses ke jaringan luar contoh google.com maka , alurnya kita request kepada server proxy,kemudian diteruskan kepada server google. pada server google akan menerima request alamat dari server proxy yang kita gunakan, bukan ip yang kita gunakan. maka dari itu jaringan luar akan kesulitan untuk memasuki ip client/kita. karena diatasnamakan/ditanggulangi oleh proxyserver.

1. ARSITEKTUR REVERSE PROXY



1. PEMBAGIAN KERJA

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama** | **NRP** | **Fitur yang dibuat** |
| 1 | Rahma Dini M. Laily | 5114100027 | Register |
| 2 | I Gede Putu Nobby A.P. | 5114100048 | Streaming |
| 3 | Nurhamidah Tyas P. | 5114100057 | Login |
| 4 | Adenuar Purnomo | 5114100079 | Download |
| 5 | Irzal Ahmad S. | 5114100123 | Searching |

1. SOURCE CODE

Didalam folder mp3server:

\_maincontrol.py

**import** os**.**path

**import** \_httpresponse

ext\_audio**=[**'mp3'**]**

**def** index**(**url**):**

f**=**open**(**'html/index.html'**).**read**()**

**return** \_httpresponse**.**resOKHTML**(**f**)**

**def** resource**(**url**):**

url**=**url**[**1**:]**

**print** url

tipe**=**url**.**split**(**'/'**)[**0**]**

**if** os**.**path**.**isfile**(**'html/'**+**url**):**

f**=**open**(**'html/'**+**url**).**read**()**

**if** tipe **in** ext\_audio**:**

**return** \_httpresponse**.**resOKMp3**(**f**)**

**else:**

**return** \_httpresponse**.**resOKFile**(**f**,**tipe**)**

**else:**

**return** \_httpresponse**.**res404**()**

\_url.py

**import** \_maincontrol

**import** \_httpresponse

pattern**={}**

#pattern['/']=\_maincontrol.index

pattern**[**'res'**]=**\_maincontrol**.**resource

pattern**[**'404'**]=**\_httpresponse**.**res404**()**

http.py

**import** socket

**import** sys

**import** threading

**import** \_url

#inisialisasi

sock **=** socket**.**socket**(**socket**.**AF\_INET**,** socket**.**SOCK\_STREAM**)**

#proses binding

server\_address **=** **(**'localhost'**,** 13001**)**

**print** **>>**sys**.**stderr**,** 'starting up on %s port %s' **%** server\_address

sock**.**bind**(**server\_address**)**

#listening

sock**.**listen**(**1**)**

**def** layani\_client**(**koneksi**,**alamat\_client**):**

**try:**

request\_message **=** ''

**while** **True:**

data **=** koneksi**.**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

a**,**url**,**c **=** baris\_request**.**split**(**" "**)**

**print** a

**print** url

**print** c

**if** url **in** \_url**.**pattern**:**

koneksi**.**send**(**\_url**.**pattern**[**url**](**url**))**

**else:**

koneksi**.**send**(**\_url**.**pattern**[**'res'**](**url**))**

#koneksi\_client.send(respon)

**finally:**

# Clean up the connection

koneksi**.**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**()**

\_httpresponse.py

**import** sys

**import** socket

**def** resOKHTML**(**data**):**

panjang**=**len**(**data**)**

hasil **=** "HTTP/1.1 200 OK\r\n" \

"Content-Type: text/html\r\n" \

"Content-Length: {}\r\n" \

"\r\n" \

"{}"**.**format**(**panjang**,**data**)**

**return** hasil

**def** res404**():**

data**=**open**(**'html/404.html'**).**read**()**

panjang**=**len**(**data**)**

hasil **=** "HTTP/1.1 404 Not Found\r\n" \

"Content-Type: text/html\r\n" \

"Content-Length: {}\r\n" \

"\r\n" \

"{}"**.**format**(**panjang**,**data**)**

**return** hasil

**def** resOKFile**(**data**,**tipe**):**

panjang**=**len**(**data**)**

hasil **=** "HTTP/1.1 200 OK\r\n" \

"Content-Type: text/{}\r\n" \

"Content-Length: {}\r\n" \

"\r\n" \

"{}"**.**format**(**tipe**,**panjang**,**data**)**

**return** hasil

**def** resOKGambar**(**data**,**tipe**):**

panjang**=**len**(**data**)**

hasil **=** "HTTP/1.1 200 OK\r\n" \

"Content-Type: image/{}\r\n" \

"Content-Length: {}\r\n" \

"\r\n" \

"{}"**.**format**(**tipe**,**panjang**,**data**)**

return hasil

def resOKMp3(data):

panjang=len(data)

hasil = "HTTP/1.1 200 OK\r\n" \

"Content-Type: application/misc\r\n" \

"Content-Length: {}\r\n" \

"\r\n" \

"{}".format(panjang,data)

return hasil

proxy\_server.py

**import** socket

**import** sys

**import** threading

sock **=** socket**.**socket**(**socket**.**AF\_INET**,** socket**.**SOCK\_STREAM**)**

server\_address **=** **(**'localhost'**,** 8080**)**

sock**.**bind**(**server\_address**)**

sock**.**listen**(**1**)**

**def** http\_get**(**message\_yang\_diteruskan**,**target**):**

client\_socket **=** socket**.**socket**(**socket**.**AF\_INET**,** socket**.**SOCK\_STREAM**)**

server\_address **=** target

client\_socket**.**connect**(**server\_address**)**

**try:**

message **=** message\_yang\_diteruskan**+**"\r\n\r\n"

client\_socket**.**sendall**(**message**)**

data\_respon **=** ""

data\_dari\_server **=** client\_socket**.**recv**(**32**)**

data\_respon **=** data\_respon**+**data\_dari\_server

**while** data\_dari\_server**:**

data\_dari\_server **=** client\_socket**.**recv**(**32**)**

data\_respon **=** data\_respon**+**data\_dari\_server

**finally:**

client\_socket**.**close**()**

**return** data\_respon

**def** layani\_client**(**koneksi\_client**,**alamat\_client**):**

**try:**

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]

#baris\_host = baris[1]

a,url,c = baris\_request.split(" ")

url=url[1:]

ext=url.split('/')[0]

target=''

print url.split('/')

if ext=='mp3':

target=('localhost',13001)

else:

target=('localhost',13000)

koneksi\_keluar = http\_get(request\_message,target)

respon = koneksi\_keluar

#print respon

koneksi\_client.send(respon)

finally:

koneksi\_client.close()

while True:

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()

Didalam folder httpserver:

http.py

**import** socket

**import** sys

**import** threading

**import** \_url

#inisialisasi

sock **=** socket**.**socket**(**socket**.**AF\_INET**,** socket**.**SOCK\_STREAM**)**

#proses binding

server\_address **=** **(**'localhost'**,** 13000**)**

**print** **>>**sys**.**stderr**,** 'starting up on %s port %s' **%** server\_address

sock**.**bind**(**server\_address**)**

#listening

sock**.**listen**(**1**)**

**def** layani\_client**(**koneksi**,**alamat\_client**):**

**try:**

request\_message **=** ''

**while** **True:**

data **=** koneksi**.**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

a**,**url**,**c **=** baris\_request**.**split**(**" "**)**

**print** a

**print** url

**print** c

**if** url **in** \_url**.**pattern**:**

koneksi**.**send**(**\_url**.**pattern**[**url**](**url**))**

**else:**

koneksi**.**send**(**\_url**.**pattern**[**'res'**](**url**))**

#koneksi\_client.send(respon)

finally:

# Clean up the connection

koneksi.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()

\_httpresponse.py

**import** sys

**import** socket

**def** resOKHTML**(**data**):**

panjang**=**len**(**data**)**

hasil **=** "HTTP/1.1 200 OK\r\n" \

"Content-Type: text/html\r\n" \

"Content-Length: {}\r\n" \

"\r\n" \

"{}"**.**format**(**panjang**,**data**)**

**return** hasil

**def** res404**():**

data**=**open**(**'html/404.html'**).**read**()**

panjang**=**len**(**data**)**

hasil **=** "HTTP/1.1 404 Not Found\r\n" \

"Content-Type: text/html\r\n" \

"Content-Length: {}\r\n" \

"\r\n" \

"{}"**.**format**(**panjang**,**data**)**

**return** hasil

**def** resOKFile**(**data**,**tipe**):**

panjang**=**len**(**data**)**

hasil **=** "HTTP/1.1 200 OK\r\n" \

"Content-Type: text/{}\r\n" \

"Content-Length: {}\r\n" \

"\r\n" \

"{}"**.**format**(**tipe**,**panjang**,**data**)**

**return** hasil

**def** resOKGambar**(**data**,**tipe**):**

panjang**=**len**(**data**)**

hasil **=** "HTTP/1.1 200 OK\r\n" \

"Content-Type: image/{}\r\n" \

"Content-Length: {}\r\n" \

"\r\n" \

"{}"**.**format**(**tipe**,**panjang**,**data**)**

return hasil

\_maincontrol.py:

**import** os**.**path

**import** \_httpresponse

ext\_gambar**=[**'jpg'**,**'png'**]**

**def** index**(**url**):**

f**=**open**(**'html/index.html'**).**read**()**

**return** \_httpresponse**.**resOKHTML**(**f**)**

**def** resource**(**url**):**

url**=**url**[**1**:]**

**print** url

tipe**=**url**.**split**(**'/'**)[**0**]**

**if** os**.**path**.**isfile**(**'html/'**+**url**):**

f**=**open**(**'html/'**+**url**).**read**()**

**if** tipe **in** ext\_gambar**:**

**return** \_httpresponse**.**resOKGambar**(**f**,**tipe**)**

**else:**

**return** \_httpresponse**.**resOKFile**(**f**,**tipe**)**

**else:**

**return** \_httpresponse**.**res404**()**

url.py:

**import** \_maincontrol

**import** \_httpresponse

pattern**={}**

pattern**[**'/'**]=**\_maincontrol**.**index

pattern**[**'res'**]=**\_maincontrol**.**resource

pattern**[**'404'**]=**\_httpresponse**.**res404**()**

1. Screenshoot

