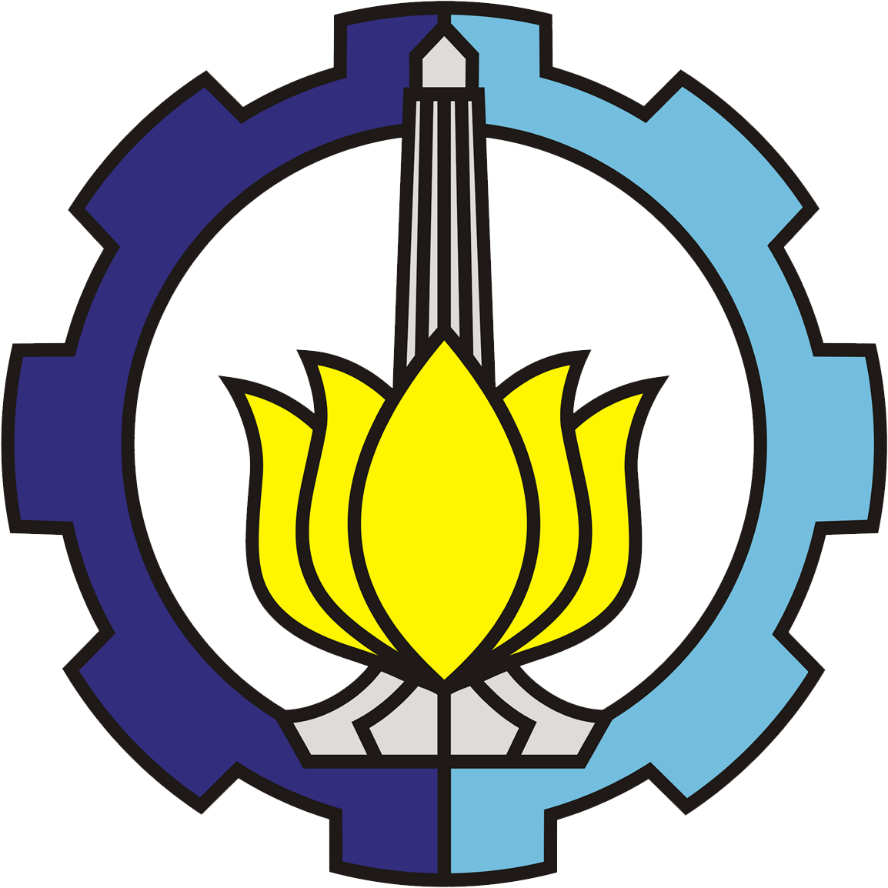
**LAPORAN TUGAS**

**LOAD BALANCER PADA WEB SERVER**



Oleh

**Kelompok 4 Pemrograman Jaringan E**

MUHAMMAD FAISHAL ILHAM (5114100076)

FARHAN RAMADHANA (5114100078) WILDAN LUTFI (5114100080)

KHARISMA MONIKA (5114100092)ANTONIUS KEVIN (5114100132)

A. LATAR BELAKANG

Salah satu mata kuliah semester 5 adalah Pemrograman Jaringan. Pada mata kuliah ini, penulis mendapatkan tugas untuk membuat load balancer. Load balancer ini berfungsi untuk mengatur network traffic yang menuju pada server.

B. RANCANGAN LOAD BALANCER

1. DESKRIPSI

Load balancing merupakan suatu teknik untuk mendistribusikan beban kerja pada dua atau bahkan lebih suatu koneksi jaringan secara seimbang agar pekerjaan dapat berjalan optimal dan tidak overload (kelebihan) beban pada salah satu jalur koneksi.Selain itu fungsi load balancing adalah untuk mengarahkan client pada server yang lain, disaat server yang satu sedang mengalami gangguan

1. SPESIFIKASI LOAD BALANCER

Load balancing yang kami rancang memiliki 5 server untuk menangani data terkait format video. Nantinya pembagian beban tersebut akan didistribusikan kepada 5 server tersebut. Setiap server memiliki spesifikasi yang sama untuk menangani request dari client terkait pemutaran video.

1. ALGORITMA YANG DIGUNAKAN PADA LOAD BALANCER

Karena setiap server yang ada memiliki spesifikasi yang sama maka kami memutuskan untuk menggunakan algoritma round robin. Kami memilih algoritma ini karena algoritma ini cocok digunakan jika suatu load balancer memiliki beberapa server dengan kemampuan proses yang sama. algoritma ini membagi beban secara berurutan ke tiga server tersebut.

1. TIPE LOAD BALANCER

Tipe load balancing yang kami gunakan yaitu merupakan software load balancing. Maksud dari tipe ini yaitu suatu load balancing berjalan di sebuah PC/server. Kami menggunakan software load balancing karena pengoperasiannya lebih mudah terutama ketika ada fitur yang ingin ditambahkan maka tidak perlu mengganti keseluruhan perangkat load balancing

1. TYPE DATA YANG DITANGANI PADA WEB SERVER

Seperti yang dijelaskan sebelumnya, tipe data yang ditangani oleh load balancing kami berupa format file video. Tipe-tipe data tersebut diantaranya seperti 3gp,mp4,flv,mkv, dan beberapa tipe data file video lainnya.

1. RETURN CODE WEB SERVER

Return code yang kami gunakan mengacu pada return code HTTP. Berikut beberapa return code yang kami gunakan yaitu :200 = jika file video ada 404 = jika file tidak ditemukan301 = file dipindah permanen

1. PEMBAGIAN KERJA TIM

|  |  |  |
| --- | --- | --- |
| **NRP** | **NAMA** | **TUGAS YANG DISELESAIKAN** |
| 5114100076 | MUHAMMAD FAISHAL ILHAM | Membuat fungsi untuk menghandle ketika request klien terhadap halaman utama |
| Membuat fungsi untuk menghandle ketika klien merequest halaman video 3gp |
| Membuat file index.html sbgai halaman depan |
| 5114100078 | FARHAN RAMADHANA | Membuat fungsi untuk menangani request klien berupa halaman video flv |
| Membuat penghandlean request ketika klien meminta video flv |
| 5114100080 | WILDAN LUTFI | membuat algo round robin utk melayani request yg datang dari client utk dibagi2 ke setiap server yang ada |
| Membuat fungsi untuk menangani request client yg datang ke load balancer sebelum diteruskan ke server dan juga meneruskan respon server ke client melalui load balancer juga. |
| 5114100092 | KHARISMA MONIKA | Handler ketika salah mengetikan path |
| HTML wrong page |
| 5114100132 | ANTONIUS KEVIN | Membuat fungsi untuk menerima request client dari load balancer dan juga untuk memberikan respon balik ke client melalui load balancer |
| Membuat fungsi ketika ada request klien berupa video mp4 |
| Membuat rancangan awal fungsi untuk penghandlean request. |

1. SOURCE CODE LOAD BALANCER

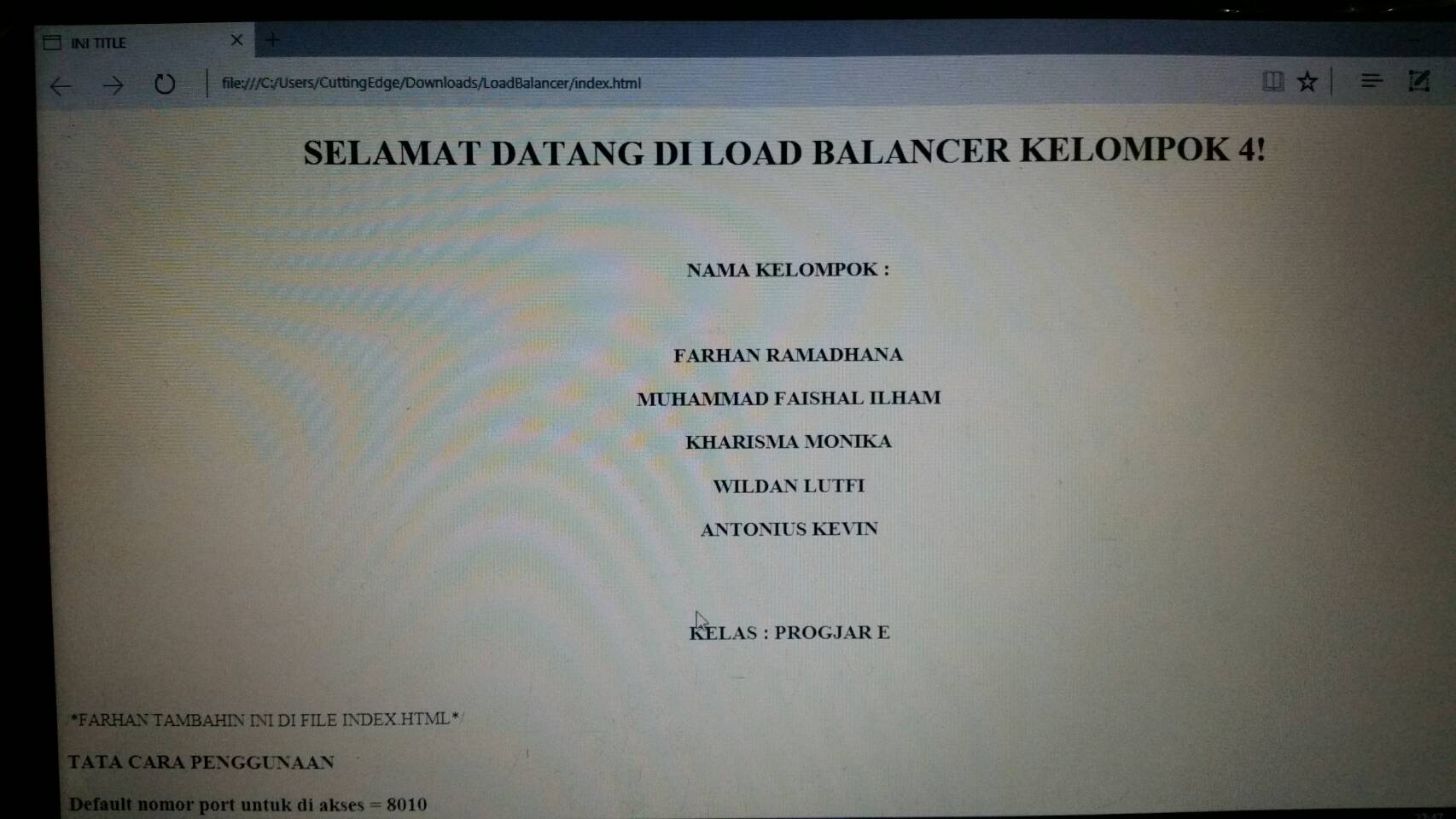
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | import socket, sys, threading | |  |  | |  | #inisialisasi | |  | sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) | |  |  | |  | #proses binding | |  | server\_address = ('localhost', 8010) | |  | print >>sys.stderr, 'starting up on %s port %s' % server\_address | |  | sock.bind(server\_address) | |  |  | |  | #listening | |  | sock.listen(1) | |  |  | |  | counter = 1 | |  |  | |  | def forwardedServer(messageToForward, link, port): | |  | forwardSocket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) | |  | linkAndPort = (link, port) | |  | forwardSocket.connect(linkAndPort) | |  | responData = "" | |  | try: | |  | #sending data | |  | message = messageToForward | |  | forwardSocket.sendall(message) | |  |  | |  | #read data | |  | while True: | |  | dataServer = forwardSocket.recv(64) | |  | if dataServer: | |  | responData += dataServer | |  | else: | |  | break; | |  | print dataserver | |  | finally: | |  | return responData | |  | print "Closing socket" | |  | client\_socket.close() | |  | #return responData | |  |  | |  | def forwardTo(conn, address, serverLink, serverPort): | |  | try: | |  | print >>sys.stderr, 'connection from ', address | |  | requestMessage = "" | |  | while True: | |  | data = conn.recv(64) | |  | data = bytes.decode(data) | |  | requestMessage += data | |  | if(requestMessage[-4:] == "\r\n\r\n"): | |  | break | |  |  | |  | #forwarding the message | |  |  | |  | backwardMessage = forwardedServer(requestMessage,serverLink, serverPort) | |  | conn.send(backwardMessage) | |  | finally: | |  | # Clean up the connection | |  | conn.close() | |  |  | |  | while True: | |  | #waiting for a connection | |  | print >>sys.stderr, 'waiting for a connection' | |  | clientConnection, clientIP = sock.accept() | |  | if (counter == 1): | |  | a = threading.Thread(target=forwardTo, args=(clientConnection,clientIP,'localhost',8011)) | |  | a.start() | |  | counter+=1 | |  | elif (counter == 2): | |  | b = threading.Thread(target=forwardTo, args=(clientConnection,clientIP,'localhost',8012)) | |  | b.start() | |  | counter+=1 | |  | elif (counter == 3): | |  | c = threading.Thread(target=forwardTo, args=(clientConnection,clientIP,'localhost',8013)) | |  | c.start() | |  | counter+=1 | |  | elif (counter == 4): | |  | d = threading.Thread(target=forwardTo, args=(clientConnection,clientIP,'localhost',8014)) | |  | d.start() | |  | counter+=1 | |  | elif (counter == 5): | |  | e = threading.Thread(target=forwardTo, args=(clientConnection,clientIP,'localhost',8015)) | |  | e.start() | |  | counter+=1 | |  | if(counter > 5): | |  | counter = 1 | |

1. SOURCE CODE WEB SERVER

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | |  | | import socket | |  | import sys | |  | import threading | |  |  | |  | #inisialisasi | |  | sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) | |  |  | |  | #proses binding | |  | server\_address = ('localhost', 8011) | |  | print >>sys.stderr, 'starting up on %s port %s' % server\_address | |  | sock.bind(server\_address) | |  |  | |  | #listening | |  | sock.listen(1) | |  |  | |  | def response\_hal\_depan(): | |  | filedepan = open('index.html','r').read() | |  | panjang = len(filedepan) | |  |  | |  | hasil = "HTTP/1.1 200 OK\r\n" \ | |  | "Content-Type: text/html; charset=utf-8\r\n" \ | |  | "Content-Length: {}\r\n" \ | |  | "\r\n" \ | |  | "{}" . format(panjang, filedepan) | |  | return hasil | |  |  | |  | def response\_video\_mp4(): | |  | filevideo = open('vidmp4','r').read() | |  | panjang = len(filevideo) | |  | hasil = "HTTP/1.1 200 OK\r\n" \ | |  | "Content-Type: video/mp4\r\n" \ | |  | "Content-Length: {}\r\n" \ | |  | "\r\n" \ | |  | "{}" . format(panjang, filevideo) | |  | return hasil | |  |  | |  | def response\_video\_flv(): | |  | filevideo = open('vidflv','r').read() | |  | panjang = len(filevideo) | |  | hasil = "HTTP/1.1 200 OK\r\n" \ | |  | "Content-Type: video/x-flv\r\n" \ | |  | "Content-Length: {}\r\n" \ | |  | "\r\n" \ | |  | "{}" . format(panjang, filevideo) | |  | return hasil | |  |  | |  | def response\_video\_3gp(): | |  | filevideo = open('vid3gp','r').read() | |  | panjang = len(filevideo) | |  | hasil = "HTTP/1.1 200 OK\r\n" \ | |  | "Content-Type: video/3gp\r\n" \ | |  | "Content-Length: {}\r\n" \ | |  | "\r\n" \ | |  | "{}" . format(panjang, filevideo) | |  | return hasil | |  |  | |  | def response\_icon(): | |  | filegambar = open('myicon.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\_redirect(): | |  | hasil = "HTTP/1.1 301 Moved Permanently\r\n" \ | |  | "Location: {}\r\n" \ | |  | "\r\n" . format('http://www.its.ac.id') | |  | 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 | |  |  | |  | a,url,c = baris\_request.split(" ") | |  |  | |  |  | |  | if (url=='/front'): | |  | respon = response\_hal\_depan() | |  | elif (url=='/front/videomp4'): | |  | respon = response\_video\_mp4() | |  | elif (url=='/front/videoflv'): | |  | respon = response\_video\_flv() | |  | elif (url=='/front/video3gp'): | |  | respon = response\_video\_3gp() | |  | 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() | |  |  | |

1. TAMPILAN

10.1 Tampilan Halaman Awal



11. HASIL UJI COBA

Web server ini berhasil berjalan dengan baik di browser mozilla dan chrome.