

SIMULASI LOAD BALANCING SERVER BLOG UMY MENGGUNAKAN NGINX WEB SERVER

Skripsi

Untuk memenuhi sebagian persyaratan

Mencapai derajat Sarjana S-1



Diajukan oleh

Nurbawono Auliajati

20130140076

Kepada

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH YOGYAKARTA

2018

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam daftar perpustakaan.

Yogyakarta, 28 Desember 2017



Nurbawono Auliajati

PRAKATA

Puji Syukur kehadiran Allah SWT yang telah melimpahkan rahmat beserta karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Load Balancing Server Blog Menggunakan Nginx Web Server” yang dilaksanakan di Universitas Muhammadiyah Yogyakarta (UMY). Penulis menyadari baik dalam tulisan maupun prakteknya, skripsi ini jauh dari kata sempurna, maka dari itu penulis sangat mengharapkan kritik, saran, serta kontribusi pembaca sekalian. Skripsi ini juga diperuntukan untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Teknik (S.T) pada Program Studi S1 Teknik Informatika Universitas Muhammadiyah Yogyakarta.

Penulis mengucapkan rasa terimakasih yang sebesar-besarnya atas semua bantuan yang telah diberikan, baik secara langsung maupun tidak langsung selama penyusunan tugas akhir ini hingga selesai. Secara khusus rasa terima kasih tersebut penulis sampaikan kepada:

1. Bapak Ir. Eko Prasetyo, M.Eng. dan Bapak Reza Giga Isnanda, S.T., M.Sc. selaku dosen pembimbing satu dan dua yang telah banyak memberikan arahan, masukan, serta bimbingan selama pengerjaan skripsi ini, sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
2. Bapak Chayadi Oktomy N S, S.T., M.Eng., ITILF. Selaku kepala prodi teknologi informasi UMY sekaligus dosen penguji pendadaran yang telah memudahkan penulis dalam menyelesaikan skripsi ini.

Bapakku Akbar Sri Gutomo, Ibuku Noples Suharti yang telah membesarkanku dengan penuh kasih sayang serta memberikanku semangat

dan motifasi berharga dalam mengarungi hidup diperantauan. Juga kedua adikku Wahyu Budianto dan Damar Jati Abdullah yang kurindukan.

3. Rekan-rekan kerja di Biro Sistem Informasi Universitas Muhammadiyah Yogyakarta yang memberikan dorongan dan semangat untuk menyelesaikan studi S1 penulis.
4. Teman-teman seperjuangan teknologi informasi UMY angkatan 2013, khususnya TI UMY kelas B yang penulis sayangi, terima kasih telah memberikan warna didalam perkuliahan sehari-hari.

Terakhir penulis berharap, semoga tugas akhir ini dapat memberikan hal yang bermanfaat dan menambah wawasan bagi pembaca dan khususnya bagi penulis juga.

Yogyakarta, November 2017

Penulis,

Nurbawono Auliajati

DAFTAR ISI

HALAMAN PENGESAHAN I.....	ii
HALAMAN PENGESAHAN II.....	iii
PRAKATA.....	iv
PERNYATAAN	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiv
INTISARI	xv
ABSTRAK.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	6
2.1 Tinjauan Pustaka	6
2.2 Landasan Teori	10
2.2.1 Blog Universitas Muhammadiyah Yogyakarta	10
2.2.2 Load Balance	11
2.2.3 Nginx	12
2.2.4 FastCGI Proxy	12
2.2.5 Least connected load balancing.....	17

2.2.6 Virtualization.....	18
2.2.7 Proxmox	19
2.2.8 Server.....	20
2.2.9 Server Hardening.....	21
2.2.10 Ubuntu	22
2.2.11 Nano	22
2.2.12 PHP 7.0.....	23
2.2.13 PHP-FPM	23
2.2.14 Wordpress.....	23
2.2.15 Putty SSH Client	24
2.2.16 MariaDB Database	24
BAB III METODOLOGI PENELITIAN	25
3.1 Metode Penelitian.....	25
3.1.1 Persiapan	26
3.1.2 Perencanaan.....	26
3.1.3 Desain.....	26
3.1.4 Implementasi	27
3.1.5 Operasi.....	27
3.1.6 Optimasi	28
3.2 Peralatan Pendukung	28
3.2.1 Software Pendukung.....	28
3.2.2 Hardware Pendukung	29
BAB IV HASIL DAN PEMBAHASAN	32

4.1 Persiapan	32
4.1.1 Informasi Kebutuhan Pengguna.....	32
4.1.2 Spesifikasi serta konfigurasi Nginx server yang dibutuhkan.....	32
4.1.3 Evaluasi dampak terhadap server Load Balance yang ada	33
4.1.4 Merancang Desain Topologi Server	34
4.2 Perencanaan	34
4.2.1 Observasi Server	34
4.3 Desain Topologi.....	35
4.3.1 Desain Topologi Fisik.....	35
4.3.2 Desain Topologi Logikal	36
4.3.3 Implementasi Dan Konfigurasi Nginx Load Balance Server	38
4.3.4 Instalasi MariaDB	39
4.3.5 Implementasi Dan Konfigurasi Nginx Load Balance Backend Server	40
4.3.6 Implementasi Dan Konfigurasi PHP 7.0 dan PHP 7.0 Fpm	42
4.3.7 Konfigurasi MySQL Remote Database	42
4.3.8 Implementasi Dan Konfigurasi Wordpress.....	44
4.4 Passing a Request to a Proxied Server Menggunakan Module Proxy_Pass	48
4.4.1 Penjelasan Cara Kerja Konfigurasi Load Balance.....	48
4.4.2 Mengoptimalkan Performa Nginx	49
4.5 TCP Load Balancing.....	51
4.5.1 Pencegahan DDOS Attack.....	52
4.5.2 Worker Process And Worker Connection.....	53
4.6 Nginx Gzip Compression Module	54
4.6.1 Nginx Http Gzip Module	54
4.7 Optimasi Dan Mini Security	56
4.7.1 HTTP Secure Header	56

4.8 Pengujian.....	58
4.8.1 Simulasi Load Balance	58
4.8.2 DOS/DDOS Attack Simulasi.....	59
4.8.3 Mencegah DOS/DDOS Attack dengan Limit Connection	61
BAB V KESIMPULAN DAN SARAN	63
5.1 Kesimpulan	63
5.2 Saran	65
DAFTAR PUSTAKA	66

DAFTAR TABEL

Tabel 2.1 Nginx dan Apache Jumlah Proses berdasarkan Waktu.....	6
Tabel 3.1 Software dan kegunaannya	29
Tabel 3.2 Spesifikasi perangkat keras server load balance	30
Table 3.3 Spesifikasi perangkat keras server backend.....	30
Table 3.4 Spesifikasi Laptop.....	31

DAFTAR GAMBAR

Gambar 2.1	Gartner Magic quadrant Application Delivery Controller	8
Gambar 2.2	Tampilan halaman depan blog UMY 10	10
Gambar 2.3	Login page blog UMY.....	11
Gambar 2.4	Struktur topologi dan mekanisme load balance.....	12
Gambar 2.5	Nginx php FastCGI.....	13
Gambar 2.6	Nginx fastcgi_intercept_error.....	15
Gambar 2.7	Least connected load balance	18
Gambar 2.8	Client- Server Communication.....	20
Gambar 2.9	IBM microserver	21
Gambar 3.1	PPDIOO Life Cycle.....	26
Gambar 4.1	Desain Topologi Fisik	34
Gambar 4.2	Desain Topologi Fisik	35
Gambar 4.2	Desain Topologi Logical	34
Gambar 4.3	Desain Topologi Logikal.....	36
Gambar 4.4	Pengaturan Load Balance Least Connected	37
Gambar 4.5	Langkah Kerja Implementasi Load Balance	39
Gambar 4.6	Pengaturan Load Balance	41
Gambar 4.7	Membuat Database pada MySQL	43

Gambar 4.8 Menampilkan Keseluruhan Database	43
Gambar 4.9 Wordpress Remot Database Konfigurasi	44
Gambar 4.10 Remote Database Konfigurasi	44
Gambar 4.11 Nginx konfigurasi 1	47
Gambar 4.12 Nginx konfigurasi 2	47
Gambar 4.13 Pengaturan proxy_pass berdasarkan file path	48
Gambar 4.14 Server Block	49
Gambar 4.15 Nginx http log module.....	50
Gambar 4.16 Nginx Gzip Compression Module.....	50
Gambar 4.17 Nginx client body buffer size	51
Gambar 4.18 Nginx TCP loadbalance.....	51
Gambar 4.19 Nginx Ddos Proteksi.....	52
Gambar 4.20 Nginx Worker Process and Worker Connection	53
Gambar 4.21 Jumlah CPU Core	53
Gambar 4.22 Nginx Gzip Compression	55
Gambar 4.23 Nginx Gzip Static	55
Gambar 4.24 Nginx HTTP secure Header	56
Gambar 4.25 Backend Server 1	58
Gambar 4.26 Backend Server 2.....	58
Gambar 4.27 Backend Server 3.....	59
Gambar 4.28 Pengaturan jumlah user/request DDOS	60

Gambar 4.29 HTTP Request DOS	61
Gambar 4.30 Nginx Connection Request Result.....	61
Gambar 4.31 Nginx Limit Connection and Request	62
Gambar 5.1 Unsecure Web Header	64
Gambar 5.2 Database Spam	64

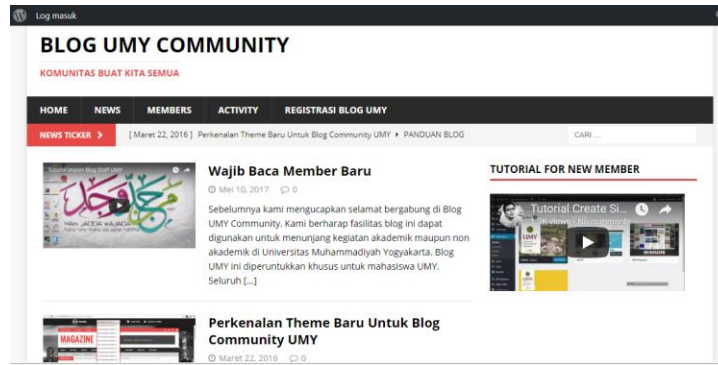
DAFTAR LAMPIRAN

Tabel 2.1 Nginx dan Apache Jumlah Proses berdasarkan Waktu [2]

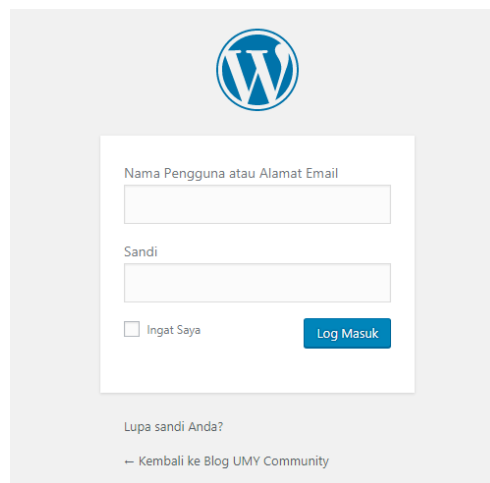
<i>No. of Requests</i>	<i>Total Processing Time (sec)</i>	
	<i>Apache Load Balancer</i>	<i>Nginx Load Balancer</i>
100	14.7	8.39
400	22.8	18.05
900	52.2	36.6
1600	177.2	59.5
2500	192.69	124.31
6400	292.52	186.78
10000	538.55	346.09



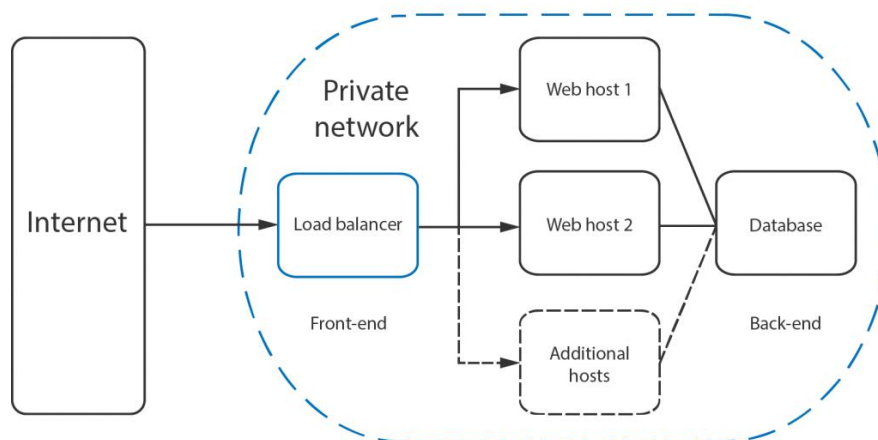
Gambar 2.1 Gartner *Magic quadrant Application Delivery Controller*. Diadopsi dari Gartner
 Quadrant



Gambar 2.2 Tampilan halaman depan blog UMY



Gambar 2.3 Login page blog UMY



Gambar 2.4 Struktur topologi dan mekanisme *load balance*

```

location ~ /\.php$ {
#   root          html;
  fastcgi_pass   unix:/var/run/php/php7.0-fpm.sock;
  fastcgi_split_path_info ^(.+\.(php|/.+))$;
#   fastcgi_pass   127.0.0.1:9000;
  fastcgi_index  index.php;
  fastcgi_param  SCRIPT_FILENAME $document_root$fastcgi_script_name;
#   fastcgi_param  SCRIPT_FILENAME /scripts$fastcgi_script_name;
  include       fastcgi_params;
  fastcgi_intercept_errors on;
  fastcgi_ignore_client_abort off;
  fastcgi_connect_timeout 60;
  fastcgi_send_timeout 180;
  fastcgi_read_timeout 180;
  fastcgi_buffer_size 128k;
  fastcgi_buffers 4 256k;
  fastcgi_busy_buffers_size 256k;
  fastcgi_temp_file_write_size 256k;
}

```

Gambar 2.5 Nginx php FastCGI

```

error_page 404 /errors/404.html;

location / {
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_intercept_errors on;
    ...
}

location /errors/ {
    # static
}

```

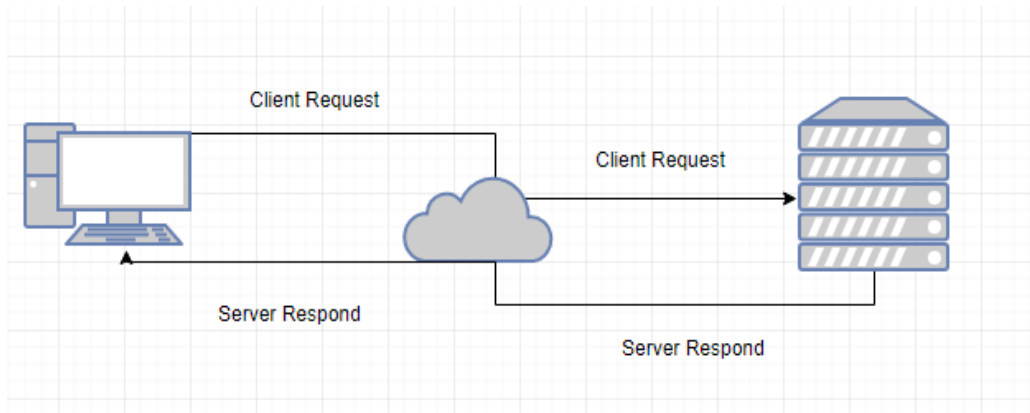
Gambar 2.6 Nginx fastcgi_intercept_error

```

upstream myapp1 {
    least_conn;
    server srv1.example.com;
    server srv2.example.com;
    server srv3.example.com;
}

```

Gambar 2.7 Least connected load balance



Gambar 2.8 Client- *Server* Communication



Gambar 2.9 IBM *microserver*



Gambar 3.1 PPDIIO Life Cycle

Tabel 3.1 Software dan kegunaannya

No	Software	Fungsi
1	Proxmox	Server Virtualisasi
2	Ubuntu 14.04 LTS	Operating System load balance dan backend server
3	Ubuntu 16.04 LTS	Operating System Laptop penulis sebagai <i>client</i>
4	Nginx	Load balancer dan web server
5	Nano	<i>Editor text</i> berbasis <i>command line</i> pada terminal GNU/Linux, digunakan untuk melakukan <i>editing text</i> pada konfigurasi Nginx dan lainnya
6	Php 7.0	Bahasa pemrograman Wordpress (backend server) serta server scripting pada Nginx
7	Wordpress	CMS pada backend server
8	MariaDB	Database Wordpress
9	Putty ssh client	Alat atau <i>tool</i> yang digunakan sebagai penghubung (<i>remoting</i>) penulis dengan <i>server-server</i> yang akan dikonfigurasi dalam penelitian ini

Tabel 3.2 Spesifikasi perangkat keras *server load balance*

Manufacture	Lenovo
Model	3250 M5
CPU Cores	6 x CPU E5-2620 0 @ 2.00GHz
Processor Type	Intel(R) Xeon(R)
Memory	8GB

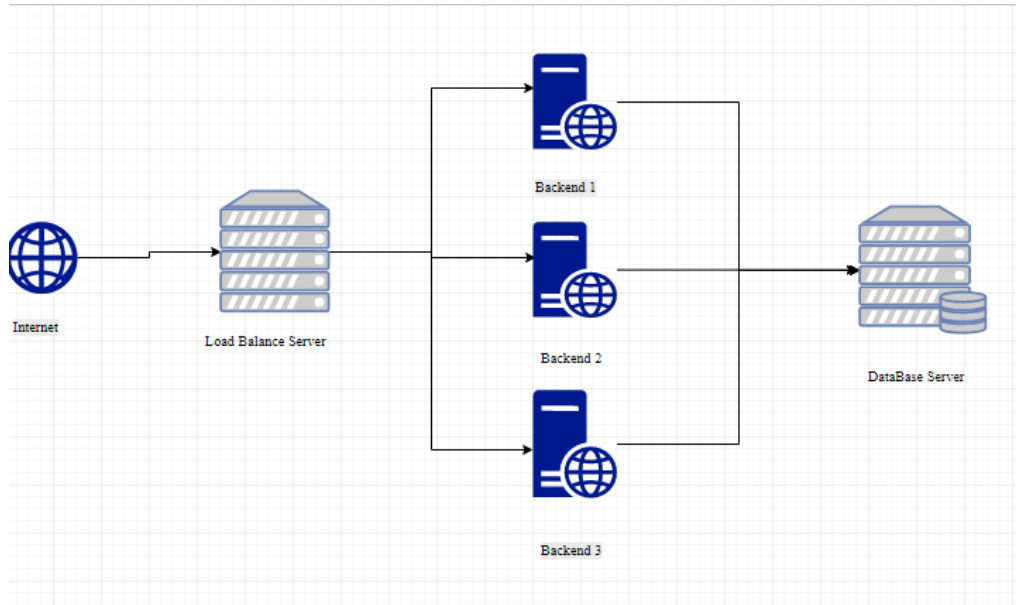
HardDisk	2 TB Configured Raid 1
----------	------------------------

Tabel 3.3 Spesifikasi perangkat keras *server backend*

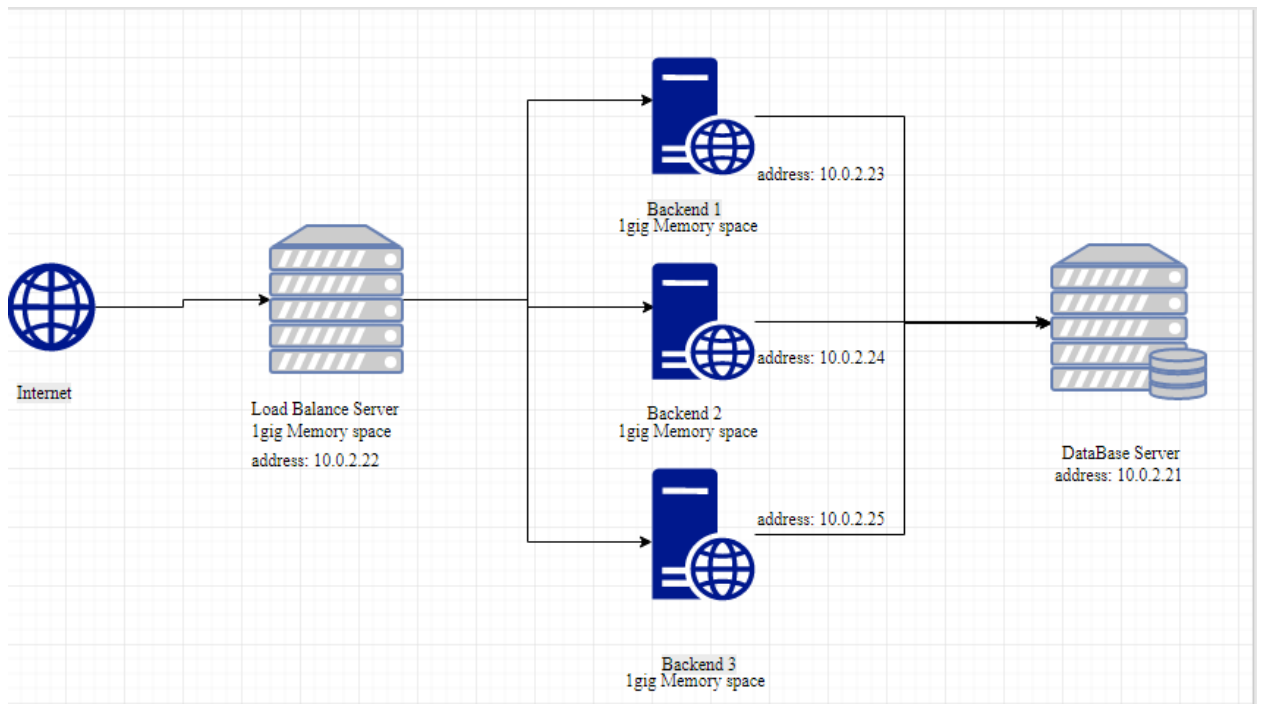
Manufacture	Lenovo
Model	3650 M4
CPU Cores	8 x CPU E3-1231 v3 @ 3.40GHz
Processor Type	Intel(R) Xeon(R)
Memory	32GB
HardDisk	2 TB Configured Raid 1

Tabel 3.4 Spesifikasi Laptop

Manufacture	Asus
Model	A46C
CPU Cores	i5 x 3517u CPU NVIDIA GEFORCE 740M 2GB @ 2.4GHz
Processor Type	Intel(R) HD Graphics
Memory	4GB
HardDisk	500GB



Gambar 4.1 Desain Topologi Fisik



Gambar 4.3 Desain Topologi Logikal

```

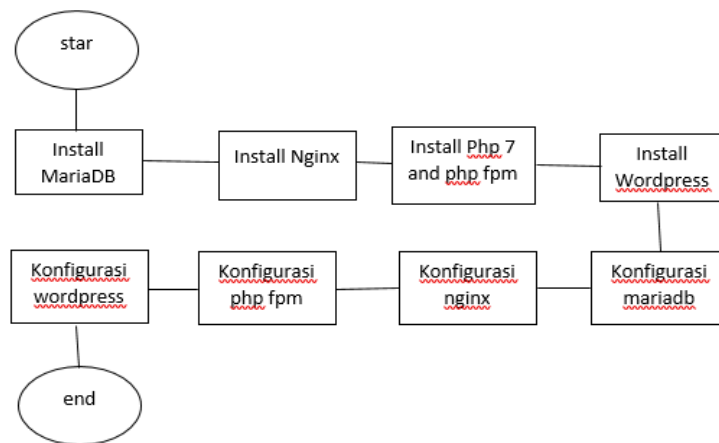
upstream test {
    least_conn;
    server 10.0.2.23;
    server 10.0.2.24;
    server 10.0.2.25;
}

server {
    listen 80;
    server_name loadbtest.ummy.ac.id;

    location / {
        proxy_pass http://test;
    }
}

```

Gambar 4.4 Pengaturan *Load Balance Least Connected*



Gambar 4.5 Langkah Kerja Implementasi *Load Balance*

```

upstream test {
    least_conn;
    server 10.0.2.23;
    server 10.0.2.24;
    server 10.0.2.25;
}

```

Gambar 4.6 Pengaturan *Load Balance*

```
CREATE DATABASE wordpress;
```

Gambar 4.7 Membuat Database pada MySQL

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| dbBlogUMY2 |
| mysql |
| performance_schema |
| wepe |
| wepe23 |
| wepe25 |
+-----+
7 rows in set (0.03 sec)
```

Gambar 4.8 Menampilkan Keseluruhan Database

```

// ** MySQL settings - You can get
/** The name of the database for W
define('DB_NAME', 'wepe');

/** MySQL database username */
define('DB_USER', ' ');

/** MySQL database password */
define('DB_PASSWORD', ' ');

/** MySQL hostname */
define('DB_HOST', '10.0.2.21');

* @link https://codex.wordpress
*
* @package WordPress
*/

// ** MySQL settings - You can get
/** The name of the database for W
define('DB_NAME', 'wepe25');

/** MySQL database username */
define('DB_USER', ' ');

/** MySQL database password */
define('DB_PASSWORD', ' ');

/** MySQL hostname */
define('DB_HOST', '10.0.2.21');

```

Gambar 4.9 Wordpress Remot Database Konfigurasi

```
#skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
```

Gambar 4.10 Remote Database Konfigurasi

```

listen      80;
server_name 10.0.2.25;
root /usr/share/nginx/wp/wordpress;
index index.php;
limit_conn backend3_limit_conn 1;
limit_req zone=mybackend3 burst=10 nodelay;

port_in_redirect off;
server_tokens off;
autoindex off;

client_max_body_size 15m;
client_body_buffer_size 128k;

#charset koi8-r;
access_log /var/log/nginx/testaccess.log main;
error_log /var/log/nginx/testerror.log;

location / {
    limit_conn backend3_limit_conn 1;
    limit_req zone=mybackend3 burst=10 nodelay;
    try_files $uri $uri/ /index.php?$args;
    autoindex off;
}

```

Gambar 4.11 Nginx konfigurasi 1

```

# proxy the PHP scripts to Apache listening on 127.0.0.1:80
#
#location ~ /\.php$ {
#    proxy_pass http://127.0.0.1;
#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
#
location ~ /\.php$ {
#    root          html;
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
    fastcgi_split_path_info ^(.+\.(php|php5))(/.+)$;
    fastcgi_intercept_errors on;
    fastcgi_ignore_client_abort off;
    fastcgi_connect_timeout 60;
    fastcgi_send_timeout 180;
    fastcgi_read_timeout 180;
    fastcgi_buffer_size 128k;
    fastcgi_buffers 4 256k;
    fastcgi_busy_buffers_size 256k;
    fastcgi_temp_file_write_size 256k;
}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
location ~ /\.ht {
    deny all;
}

```

Gambar 4.12 Nginx konfigurasi 2

```

location /some/path/ {
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_pass http://localhost:8000;
}

```

Gambar 4.13 Pengaturan *proxy_pass* berdasarkan *file path*

```

server {
    listen      80;
    server_name example.org www.example.org;
    ...
}

server {
    listen      80;
    server_name *.example.org;
    ...
}

```

Gambar 4.14 *Server Block*

```

log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for" '
                '"$host" sn="$server_name" '
                'rt=$request_time '
                'ua="$upstream_addr" us="$upstream_status" '
                'ut="$upstream_response_time" ul="$upstream_response_length" '
                'cs=$upstream_cache_status' ;

access_log /var/log/nginx/access.log main;

```

Gambar 4.15 Nginx *http log module*

```
gzip on;
gzip_comp_level 2;
gzip_static on;
gzip_disable "msie6";
gzip_vary on;
gzip_proxied any;
gzip_min_length 1000;
gzip_buffers 16 8k;
gzip_http_version 1.1;
gzip_types text/css text/javascript text/xml text/plain text/x-component
application/javascript application/x-javascript application/json
application/xml application/rss+xml font/truetype application/x-font-ttf
font/opentype application/vnd.ms-fontobject image/svg+xml;
```

Gambar 4.16 Nginx *Gzip Compression Module*

```
client_max_body_size 100k;
client_body_timeout 10;
client_header_timeout 10;
client_body_buffer_size 128k;
client_header_buffer_size 1k;
large_client_header_buffers 2 4k;
```

Gambar 4.17 Nginx client body buffer size

```
upstream test {
    least_conn;
    server 10.0.2.23;
    server 10.0.2.24;
    server 10.0.2.25;
}
```

Gambar 4.18 Nginx *TCP Load Balance*

```
limit_conn_zone $binary_remote_addr zone=backendl_limit_conn:5m;
limit_req_zone $binary_remote_addr zone=mybackendl:10m rate=5r/s;
```

```
location / {
    limit_conn backendl_limit_conn 1;
    limit_req zone=mybackendl burst=10 nodelay;
    try_files $uri $uri/ /index.php?$args;
}
```


Gambar 4.19 Nginx Ddos Proteksi

```
worker_processes 1;

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}
```

Gambar 4.20 Nginx Worker Process and Worker Connection

```
root@cluster03:/etc/nginx# lscpu | egrep '^Thread|^Core|^Socket|^CPU\'
CPU(s): 1
Thread(s) per core: 1
Core(s) per socket: 1
Socket(s): 1
```

Gambar 4.21 Jumlah CPU Core

```
gzip on;
gzip_comp_level 2;
gzip_static on;
gzip_disable "msie6";
gzip_vary on;
gzip_proxied any;
gzip_min_length 1000;
gzip_buffers 16 8k;
gzip_http_version 1.1;
gzip_types text/css text/javascript text/xml text/plain text/x-component
application/javascript application/x-javascript application/json
application/xml application/rss+xml font/truetype application/x-font-ttf
font/opentype application/vnd.ms-fontobject image/svg+xml;
```

Gambar 4.22 Nginx Gzip Compression

```
Sep 29 13:45 access.log.10.gz
Sep 26 12:54 access.log.11.gz
Sep 25 13:32 access.log.12.gz
Sep 23 15:17 access.log.13.gz
Sep 22 17:57 access.log.14.gz
Okt 24 13:08 access.log.2.gz
Okt 23 09:35 access.log.3.gz
Okt 19 15:24 access.log.4.gz
Okt 14 14:37 access.log.5.gz
Okt 11 09:29 access.log.6.gz
Okt 6 11:17 access.log.7.gz
Okt 4 15:25 access.log.8.gz
Sep 30 13:30 access.log.9.gz
```

Gambar 4.23 Nginx Gzip Static

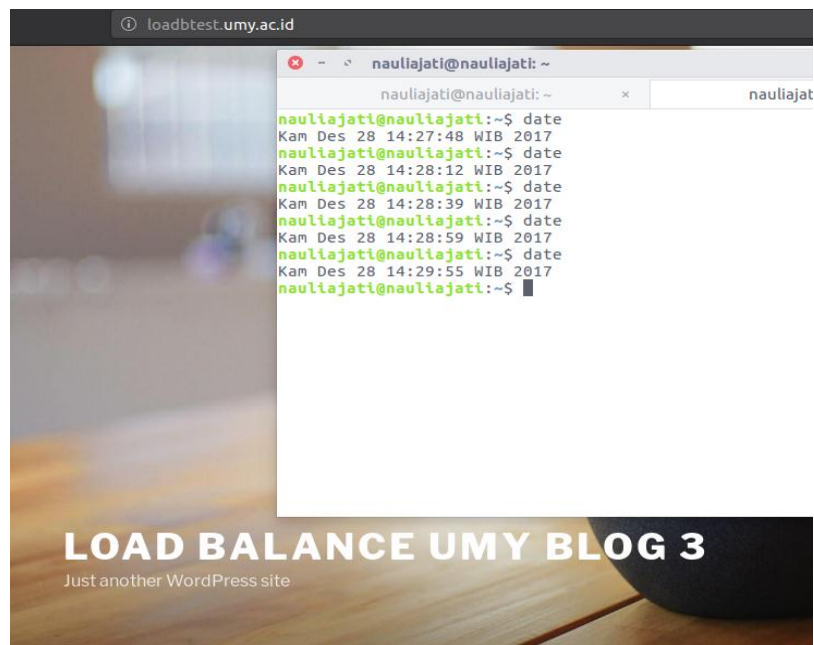
```
# X-Frame-Options is to prevent from clickJacking attack
add_header X-Frame-Options SAMEORIGIN;

# disable content-type sniffing on some browsers.
add_header X-Content-Type-Options nosniff;

# This header enables the Cross-site scripting (XSS) filter
add_header X-XSS-Protection "1; mode=block";

# This will enforce HTTP browsing into HTTPS and avoid ssl stripping attack
add_header Strict-Transport-Security "max-age=31536000; includeSubdomains;";
```

Gambar 4.24 Nginx HTTP secure Header



The image shows a terminal window with the following output:

```
naulijati@naulijati:~$ date
Kam Des 28 14:27:48 WIB 2017
naulijati@naulijati:~$ date
Kam Des 28 14:28:12 WIB 2017
naulijati@naulijati:~$ date
Kam Des 28 14:28:39 WIB 2017
naulijati@naulijati:~$ date
Kam Des 28 14:28:59 WIB 2017
naulijati@naulijati:~$ date
Kam Des 28 14:29:55 WIB 2017
naulijati@naulijati:~$
```

The background of the terminal window is a blurred screenshot of a website. At the bottom of the image, there is a banner with the text:

LOAD BALANCE UMY BLOG 3
Just another WordPress site

Gambar 4.27 Backend Server 3

Thread Properties	
Number of Threads (users):	15
Ramp-Up Period (in seconds):	3
Loop Count:	<input checked="" type="checkbox"/> Forever

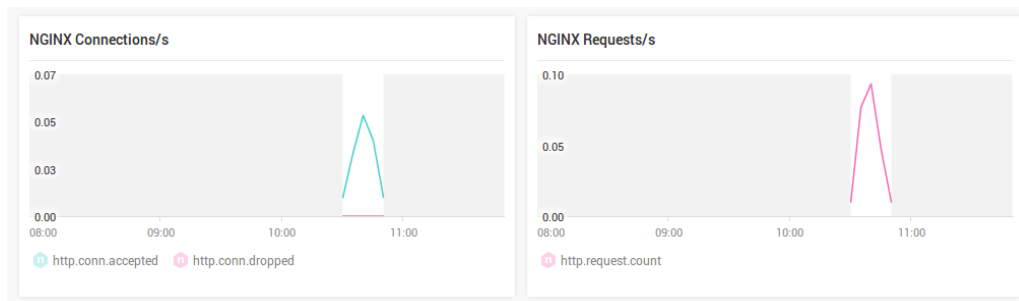
Gambar 4.28 Pengaturan jumlah user/request DDOS

The screenshot shows the JMeter interface with a list of 20 'HTTP Request' samplers on the left, each marked with a red 'X'. The right pane displays the 'Sampler result' for one of these requests. The response details are as follows:

- Thread Name: Thread Group 1-6
- Sample Start: 2017-12-21 13:53:34 WIB
- Load time: 5
- Connect Time: 0
- Latency: 5
- Size in bytes: 333
- Sent bytes: 113
- Headers size in bytes: 171
- Body size in bytes: 162
- Sample Count: 1
- Error Count: 1
- Data type ("text"|"bin"|"*"): text
- Response code: 403
- Response message: Forbidden
- Response headers: HTTP/1.1 403 Forbidden
- Server: nginx
- Date: Thu, 21 Dec 2017 06:56:19 GMT
- Content-Type: text/html
- Content-Length: 162
- Connection: keep-alive
- Keep-Alive: timeout=5

At the bottom, the 'HTTPSampleResult fields' are listed: ContentType: text/html, DataEncoding: null. The 'Raw' and 'Parsed' tabs are visible at the bottom.

Gambar 4.29 HTTP Request DOS



Gambar 4.30 Nginx Connection Request Result

```
#charset koi8-r;  
access_log /var/log/nginx/access.log main;  
error_log /var/log/nginx/error.log;  
location / {  
    limit_conn backend1_limit_conn 1;  
    limit_req zone=mybackend1 burst=10 nodelay;  
    limit_req_status 403;  
    try_files $uri $uri/ /index.php?$args;  
}
```

Gambar 4.31 *Nginx Limit Connection and Request*