

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation

Install Ubuntu 16.04

In this installation part of Ubuntu OS we have to do this 4 times. One for server load balancing, two for web server, and last for database server. Below are some views during installation:

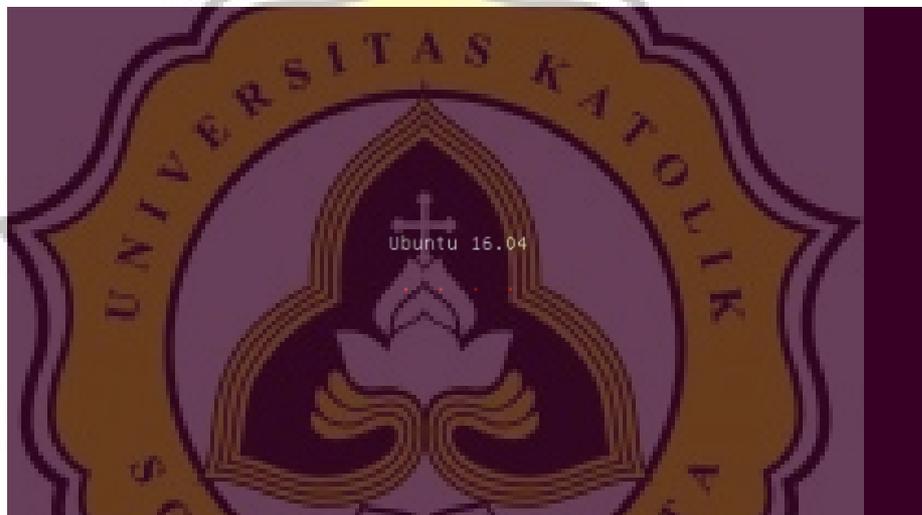


Illustration 5.1: Ubuntu 1

1. This is the first view of instalation.

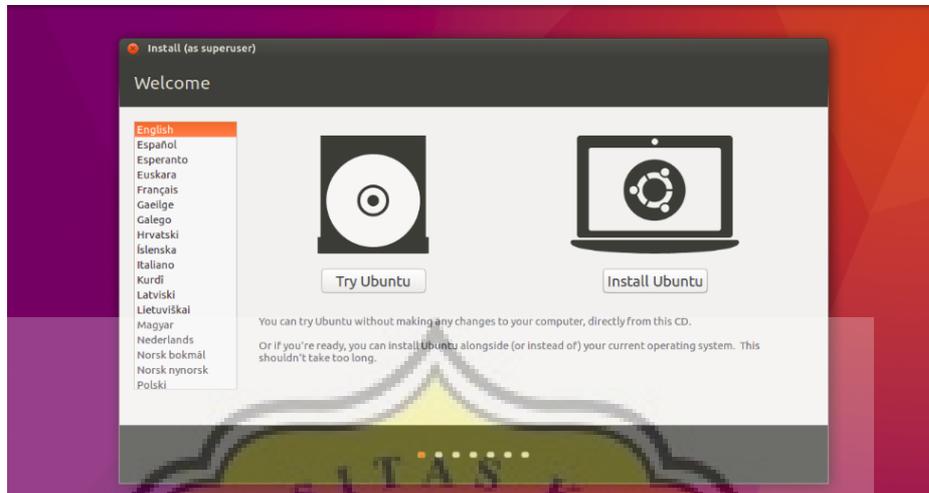


Illustration 5.2: Ubuntu 2

2. In this part choose the language and click “Install Ubuntu” to install the operating system.



Illustration 5.3: Ubuntu 3

3. Select the top part to delete all contents of the hard disk to install ubuntu.

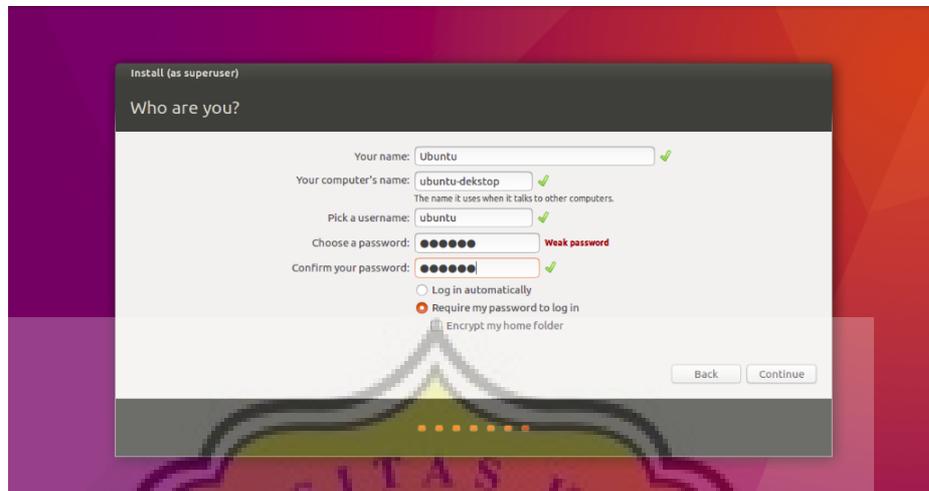


Illustration 5.4: Ubuntu 4

4. Next step are required to fill in the computer name and password.

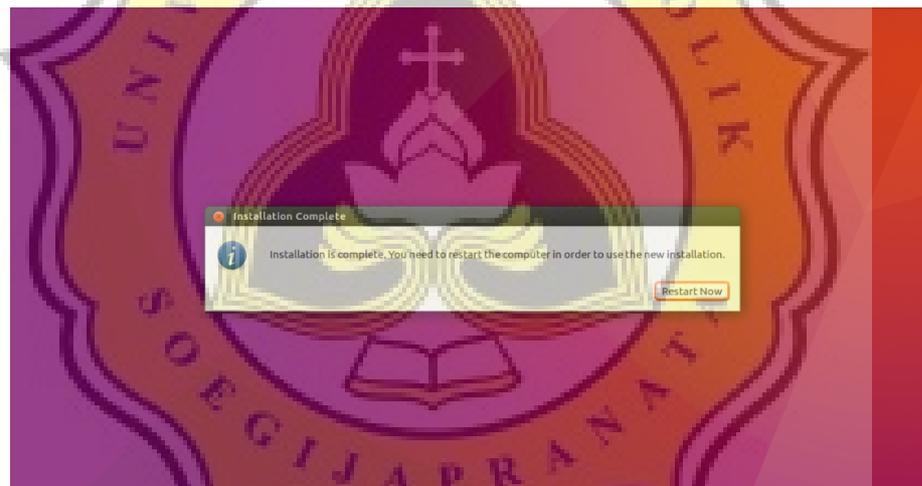


Illustration 5.5: Ubuntu 5

5. When loading for the installation is complete then the display will appear as above, press Restart now to restart the pc and installation is complete.

Configure Web Server 1 & 2

1. The First step is installing the apache2 webserver and and also php module.

```
apt-get install apache2
apt-get install -y php libapache2-mod-php php-mcrypt php-mysql
```

After the installation is complete, we can check apache2 status.

```
service apache2 status
```

2. To be able to display web pages as we want it should create a separate folder in the folder /var/www.

```
mkdir -p /var/www/katalogikom.ac.id/public_html
```

3. The command below is used to change ownership and permissions on the folder. After that we can copy the website file into folder katalog.ac.id.

```
chown -R $ubuntu1:$ubuntu1
/var/www/katalogikom.ac.id/public_html/
chmod -R 755 /var/www
```

4. Next create a new configuration file so that when accessing web server can display web page that we have created. To create a new configuration file can copy the file default with the command,

```
cp /etc/apache2/sites-available/000-default.conf
/etc/apache2/sites-available/katalogikom.ac.id.conf
```

open the copied file.

```
nano katalogikom.ac.id.conf
```

Then replace the contents of the file as shown with:

```
<VirtualHost *:80>
    ServerAdmin admin@katalogikom.ac.id
    ServerName katalogikom.ac.id
    ServerAlias www.katalogikom.ac.id
    DocumentRoot
/var/www/katalogikom.ac.id/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

5. Then enabled the configuration file we have created and the default configuration file disabled.

```
a2ensite katalogikom.ac.id.conf  
a2dissite 000-default.conf
```

6. Lastly restart apache2 so that all the configuration that has been made is running.

```
service apache2 restart
```

Configure DataBase

1. The first step is install mysql, phpmyadmin on the database server, and also apache2. Because to open phpmyadmin need apache2.

```
apt-get install mysql  
apt-get install apache2  
apt-get install phpmyadmin
```

2. If everything is installed then the next step to change the configuration in mysql to be accessed from the web server by opening the configuration file.

```
nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

and replace the contents of the file in the bind address with database server ip address.

```
# Instead of skip-networking the default is now to listen  
only on
```

```
# localhost which is more compatible and is not less  
secure.
```

```
bind-address          = 192.168.227.135
```

Nama pengguna	Host name	Kata Sandi	Hak akses global	Grup pengguna	Pemberi Izin	Tindakan
<input type="checkbox"/> debian-sys-maint	localhost	Ya	ALL PRIVILEGES		Ya	Edit privileges Ekspor
<input type="checkbox"/> mysql.session	localhost	Ya	SUPER		Tidak	Edit privileges Ekspor
<input type="checkbox"/> mysql.sys	localhost	Ya	USAGE		Tidak	Edit privileges Ekspor
<input type="checkbox"/> phpmyadmin	localhost	Ya	USAGE		Tidak	Edit privileges Ekspor
<input checked="" type="checkbox"/> root	%	Ya	ALL PRIVILEGES		Tidak	Edit privileges Ekspor
<input type="checkbox"/> root	localhost	Ya	ALL PRIVILEGES		Ya	Edit privileges Ekspor

Illustration 5.6: user account

3. And the next step creates a user account on mysql with phpmyadmin. Host on user account must be % so that all web server can access database like illustration above.

4. To create a database we can open mysql.

```
mysql -u root -p
```

Then create databases and tables.

```
drop database if exists dbDaftar;
create database dbDaftar;
use dbDaftar;
create table tblmahasiswa(
nim varchar(10) not null primary key,
nama varchar(30) not null,
notelp varchar(20),
email varchar(255),
status varchar(10) not null,
foto varchar(255));
```

5. If it is all done then the database configuration is complete

Configure Load Balancing Server

1. Install nginx.

```
apt-get install nginx
```

2. After nginx is installed then open the default configuration file.

```
nano /etc/nginx/sites-available/default
```

3. Delete all contents in the file and replace with contents.

```
Upstream backend{
    least_conn;
    server 192.168.227.131;
    server 192.168.227.132;
}
server{
    listen 80;
    location / {
        proxy_pass http://backend;
    }
}
```

where in the second line is the load balancing algorithm that is used least connection, line 3 and 4 is ip web server 1 and 2 that we are late to make.

4. Restart nginx to run the configuration we have created.

```
service nginx restart
```

5.2 Testing

In the testing with the jmeter for http requests with 50 number of requests and applied several methods of testing with 1 web server in illustration 5.7, testing when web server down illustration 5.9, load balancing testing illustration 5.8, and testing load balancing 1 server off where in illustration 5.10. In the illustration image in the form of a table, the multiple columns of the description can be used as material to compare with the test results using other methods. Among the time sample columns that indicate the time required to access the website, and the status column that if the green then it can be connected but if oren then can not be connected also at the bottom there is an average time that the client needs to be connected.

From the data presented in the picture shows that if with 1 web server produce average time 9ms, that average time is longer than using the load balancing that produces average time 5ms. While if with 1 web server and server down as in the color column oren status which means can not be connected, but if with load balancing even if 1 of the server down request still can be connected even with the average time is more longer that is 372ms.

In illustration 5.11 and 5.12 also shows when we access the load balancing server ip address from web browser load balancing server can redirects to 2 web servers evidenced by the difference in the title in the browser tab if heading to server 1 reads Katalog1 and if to server 2 it says Katalog2.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
21	11:30:37.548	Thread Group 1-21	HTTP Request	2	🟢	900	2
22	11:30:37.568	Thread Group 1-22	HTTP Request	3	🟢	900	3
23	11:30:37.588	Thread Group 1-23	HTTP Request	3	🟢	900	3
24	11:30:37.613	Thread Group 1-24	HTTP Request	3	🟢	900	3
25	11:30:37.630	Thread Group 1-25	HTTP Request	4	🟢	900	4
26	11:30:37.649	Thread Group 1-26	HTTP Request	3	🟢	900	3
27	11:30:37.671	Thread Group 1-27	HTTP Request	4	🟢	900	4
28	11:30:37.692	Thread Group 1-28	HTTP Request	2	🟢	900	2
29	11:30:37.712	Thread Group 1-29	HTTP Request	3	🟢	900	3
30	11:30:37.733	Thread Group 1-30	HTTP Request	3	🟢	900	3
31	11:30:37.753	Thread Group 1-31	HTTP Request	3	🟢	900	3
32	11:30:37.773	Thread Group 1-32	HTTP Request	4	🟢	900	4
33	11:30:37.794	Thread Group 1-33	HTTP Request	4	🟢	900	4
34	11:30:37.814	Thread Group 1-34	HTTP Request	3	🟢	900	3
35	11:30:37.834	Thread Group 1-35	HTTP Request	2	🟢	900	2
36	11:30:37.854	Thread Group 1-36	HTTP Request	4	🟢	900	4
37	11:30:37.874	Thread Group 1-37	HTTP Request	3	🟢	900	3
38	11:30:37.893	Thread Group 1-38	HTTP Request	4	🟢	900	4
39	11:30:37.914	Thread Group 1-39	HTTP Request	2	🟢	900	2
40	11:30:37.934	Thread Group 1-40	HTTP Request	4	🟢	900	4
41	11:30:37.955	Thread Group 1-41	HTTP Request	3	🟢	900	3
42	11:30:37.974	Thread Group 1-42	HTTP Request	3	🟢	900	3
43	11:30:37.995	Thread Group 1-43	HTTP Request	2	🟢	900	2
44	11:30:38.015	Thread Group 1-44	HTTP Request	4	🟢	900	4
45	11:30:38.035	Thread Group 1-45	HTTP Request	2	🟢	900	2
46	11:30:38.055	Thread Group 1-46	HTTP Request	3	🟢	900	3
47	11:30:38.076	Thread Group 1-47	HTTP Request	2	🟢	900	2
48	11:30:38.098	Thread Group 1-48	HTTP Request	3	🟢	900	3
49	11:30:38.115	Thread Group 1-49	HTTP Request	2	🟢	900	2
50	11:30:38.135	Thread Group 1-50	HTTP Request	3	🟢	900	2

Scroll automatically?
 Child samples?
No of Samples 50
Latest Sample 3
Average 9
Deviation 25

Illustration 5.7: test 1 web server

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
1	11:31:28.214	Thread Group 1-1	HTTP Request	5	🟢	836	5
2	11:31:28.235	Thread Group 1-2	HTTP Request	6	🟢	836	6
3	11:31:28.260	Thread Group 1-3	HTTP Request	6	🟢	867	6
4	11:31:28.281	Thread Group 1-4	HTTP Request	4	🟢	836	4
5	11:31:28.302	Thread Group 1-5	HTTP Request	5	🟢	867	5
6	11:31:28.324	Thread Group 1-6	HTTP Request	7	🟢	836	7
7	11:31:28.349	Thread Group 1-7	HTTP Request	6	🟢	867	6
8	11:31:28.369	Thread Group 1-8	HTTP Request	6	🟢	836	6
9	11:31:28.389	Thread Group 1-9	HTTP Request	7	🟢	867	7
10	11:31:28.410	Thread Group 1-10	HTTP Request	8	🟢	836	8
11	11:31:28.429	Thread Group 1-11	HTTP Request	5	🟢	867	5
12	11:31:28.449	Thread Group 1-12	HTTP Request	6	🟢	836	6
13	11:31:28.471	Thread Group 1-13	HTTP Request	5	🟢	867	5
14	11:31:28.494	Thread Group 1-14	HTTP Request	8	🟢	836	8
15	11:31:28.513	Thread Group 1-15	HTTP Request	6	🟢	867	6
16	11:31:28.534	Thread Group 1-16	HTTP Request	6	🟢	836	6
17	11:31:28.554	Thread Group 1-17	HTTP Request	4	🟢	867	4
18	11:31:28.574	Thread Group 1-18	HTTP Request	6	🟢	836	6
19	11:31:28.596	Thread Group 1-19	HTTP Request	5	🟢	867	5
20	11:31:28.619	Thread Group 1-20	HTTP Request	6	🟢	836	6
21	11:31:28.648	Thread Group 1-21	HTTP Request	6	🟢	867	6
22	11:31:28.669	Thread Group 1-22	HTTP Request	4	🟢	836	4
23	11:31:28.690	Thread Group 1-23	HTTP Request	6	🟢	867	6
24	11:31:28.714	Thread Group 1-24	HTTP Request	5	🟢	836	5
25	11:31:28.730	Thread Group 1-25	HTTP Request	5	🟢	867	5
26	11:31:28.750	Thread Group 1-26	HTTP Request	5	🟢	836	5
27	11:31:28.770	Thread Group 1-27	HTTP Request	6	🟢	867	6
28	11:31:28.790	Thread Group 1-28	HTTP Request	5	🟢	836	5
29	11:31:28.810	Thread Group 1-29	HTTP Request	4	🟢	867	4
30	11:31:28.830	Thread Group 1-30	HTTP Request	6	🟢	836	6

Scroll automatically?
 Child samples?
No of Samples 50
Latest Sample 5
Average 5
Deviation 1

Illustration 5.8: test with load balancing

Thread Group.jmx (/home/nlcz/Documents/Thread Group.jmx) - Apache JMeter (2.11.20151206)

File Edit Search Run Options Help

Test Plan
 Thread Group
 HTTP Cookie Manager
 HTTP Cache Manager
 HTTP Request
 View Results in Table
 WorkBench

View Results in Table
 Name: View Results in Table
 Comments:
 Write results to file / Read from file
 Filename: Browse... Log/Display Only: Errors Successes Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
1	11:32:44.448	Thread Group 1-49	HTTP Request	10400	🚩	1914	0
2	11:32:44.468	Thread Group 1-50	HTTP Request	10380	🚩	1914	0
3	11:32:43.777	Thread Group 1-6	HTTP Request	18211	🚩	1914	0
4	11:32:43.487	Thread Group 1-2	HTTP Request	18501	🚩	1914	0
5	11:32:43.507	Thread Group 1-3	HTTP Request	18482	🚩	1914	0
6	11:32:43.594	Thread Group 1-7	HTTP Request	18395	🚩	1914	0
7	11:32:43.877	Thread Group 1-21	HTTP Request	18112	🚩	1914	0
8	11:32:43.528	Thread Group 1-4	HTTP Request	18461	🚩	1914	0
9	11:32:43.756	Thread Group 1-15	HTTP Request	18234	🚩	1914	0
10	11:32:43.913	Thread Group 1-8	HTTP Request	18377	🚩	1914	0
11	11:32:43.550	Thread Group 1-5	HTTP Request	18440	🚩	1914	0
12	11:32:43.633	Thread Group 1-9	HTTP Request	18358	🚩	1914	0
13	11:32:43.836	Thread Group 1-19	HTTP Request	18155	🚩	1914	0
14	11:32:43.715	Thread Group 1-13	HTTP Request	18276	🚩	1914	0
15	11:32:44.105	Thread Group 1-32	HTTP Request	17866	🚩	1914	0
16	11:32:43.856	Thread Group 1-20	HTTP Request	18135	🚩	1914	0
17	11:32:44.125	Thread Group 1-33	HTTP Request	17866	🚩	1914	0
18	11:32:43.672	Thread Group 1-11	HTTP Request	18318	🚩	1914	0
19	11:32:44.368	Thread Group 1-45	HTTP Request	17624	🚩	1914	0
20	11:32:43.653	Thread Group 1-10	HTTP Request	18339	🚩	1914	0
21	11:32:44.165	Thread Group 1-35	HTTP Request	17829	🚩	1914	0
22	11:32:43.468	Thread Group 1-1	HTTP Request	18526	🚩	1914	0
23	11:32:43.795	Thread Group 1-17	HTTP Request	18199	🚩	1914	0
24	11:32:44.084	Thread Group 1-31	HTTP Request	17906	🚩	1914	0
25	11:32:43.581	Thread Group 1-6	HTTP Request	18407	🚩	1914	0
26	11:32:43.695	Thread Group 1-12	HTTP Request	18299	🚩	1914	0
27	11:32:43.738	Thread Group 1-14	HTTP Request	18259	🚩	1914	0
28	11:32:44.145	Thread Group 1-34	HTTP Request	17850	🚩	1914	0
29	11:32:44.408	Thread Group 1-47	HTTP Request	17587	🚩	1914	0
30	11:32:44.388	Thread Group 1-46	HTTP Request	17607	🚩	1914	0

Scroll automatically? Child samples? No of Samples 50 Latest Sample 18041 Average 17738 Deviation 1524

Illustration 5.9: test 1 web server down

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
21	11:33:55.286	Thread Group 1-22	HTTP Request	3	🟢	867	3
22	11:33:55.307	Thread Group 1-23	HTTP Request	3	🟢	867	4
23	11:33:55.326	Thread Group 1-24	HTTP Request	4	🟢	867	4
24	11:33:55.347	Thread Group 1-25	HTTP Request	4	🟢	867	4
25	11:33:55.370	Thread Group 1-26	HTTP Request	4	🟢	867	4
26	11:33:55.387	Thread Group 1-27	HTTP Request	4	🟢	867	4
27	11:33:55.407	Thread Group 1-28	HTTP Request	4	🟢	867	4
28	11:33:55.427	Thread Group 1-29	HTTP Request	5	🟢	867	5
29	11:33:55.447	Thread Group 1-30	HTTP Request	4	🟢	867	4
30	11:33:55.468	Thread Group 1-31	HTTP Request	6	🟢	867	6
31	11:33:55.488	Thread Group 1-32	HTTP Request	5	🟢	867	5
32	11:33:55.509	Thread Group 1-33	HTTP Request	4	🟢	867	4
33	11:33:55.528	Thread Group 1-34	HTTP Request	6	🟢	867	6
34	11:33:55.549	Thread Group 1-35	HTTP Request	5	🟢	867	5
35	11:33:55.568	Thread Group 1-36	HTTP Request	5	🟢	867	5
36	11:33:55.589	Thread Group 1-37	HTTP Request	5	🟢	867	5
37	11:33:55.609	Thread Group 1-38	HTTP Request	4	🟢	867	4
38	11:33:55.629	Thread Group 1-39	HTTP Request	4	🟢	867	4
39	11:33:55.649	Thread Group 1-40	HTTP Request	5	🟢	867	5
40	11:33:55.669	Thread Group 1-41	HTTP Request	4	🟢	867	4
41	11:33:55.689	Thread Group 1-42	HTTP Request	5	🟢	867	5
42	11:33:55.709	Thread Group 1-43	HTTP Request	4	🟢	867	4
43	11:33:55.729	Thread Group 1-44	HTTP Request	4	🟢	867	4
44	11:33:55.750	Thread Group 1-45	HTTP Request	4	🟢	867	4
45	11:33:55.769	Thread Group 1-46	HTTP Request	8	🟢	867	8
46	11:33:55.790	Thread Group 1-47	HTTP Request	5	🟢	867	5
47	11:33:55.811	Thread Group 1-48	HTTP Request	5	🟢	867	5
48	11:33:55.831	Thread Group 1-49	HTTP Request	4	🟢	867	4
49	11:33:55.854	Thread Group 1-50	HTTP Request	6	🟢	867	6
50	11:33:54.879	Thread Group 1-2	HTTP Request	18371	🟢	867	18371

Scroll automatically? Child samples? No of Samples 50 Latest Sample 18371 Average 372 Deviation 2571

Illustration 5.10: test load balancing 1 web server down



Illustration 5.11: redirect to web server 1



Illustration 5.12: redirect to web server 2