

BAB III METODOLOGI PENELITIAN

3.1 Subyek dan Obyek Penelitian

Subjek pada penelitian ini adalah pengguna layanan *Learning management system moodle*, dan objek dalam penyelesaian penelitian ini adalah melakukan implementasi dan menganalisis metode *load balancing* dengan menggunakan *nginx* yang diterapkan pada teknologi *container docker*.

3.2 Alat dan Bahan Penelitian

3.2.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini adalah satu laptop dengan spesifikasi sebagai berikut :

Tabel 3. 1 Spesifikasi perangkat keras

No	Perangkat	Jumlah	Kegunaan
1	Laptop (AMD Ryzen 5 4500U with Radeon Graphics 2.38 GHz, 8 GB RAM , 500 GB SSD, 6 CPUs	1	Host OS

3.2.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan pada penelitian ini adalah virtual box sebagai *hyper visor* versi 2 yang digunakan untuk alat virtualisasi sistem operasi, pada penelitian ini terdapat 5 *server* dimana satu *server* sebagai *load balancer*, dua *server* sebagai *server learning management system*, satu *server* sebagai *database server* dan satu *server* terakhir sebagai *client*. Spesifikasi *server* yang akan digunakan adalah sebagai berikut :

Tabel 3. 2 Spesifikasi perangkat lunak

No	Kegunaan	Keterangan	Ubuntu server 20.04.2
1	Loadbalancer	RAM	1 GB
		Harddisk	25 GB
		Alamat IP	192.168.56.110
2	<i>Server LMS 1</i>	RAM	2 GB
		Harddisk	50 GB
		Alamat IP	192.168.56.11
3	<i>Server LMS 2</i>	RAM	2 GB
		Harddisk	50 GB
		Alamat IP	192.168.56.10
4	<i>Database server</i>	RAM	1 GB
		Harddisk	20 GB
		Alamat IP	192.168.56.21
5	<i>Client</i>	RAM	1 GB
		Harddisk	10 GB
		Alamat IP	192.168.56.22

3.2.2 Perangkat Lunak (*Software*)

Untuk menjalankan layanan *LMS moodle* dan melakukan pengukuran hasil performa diperlukan beberapa aplikasi, berikut aplikasi/*software* yang digunakan :

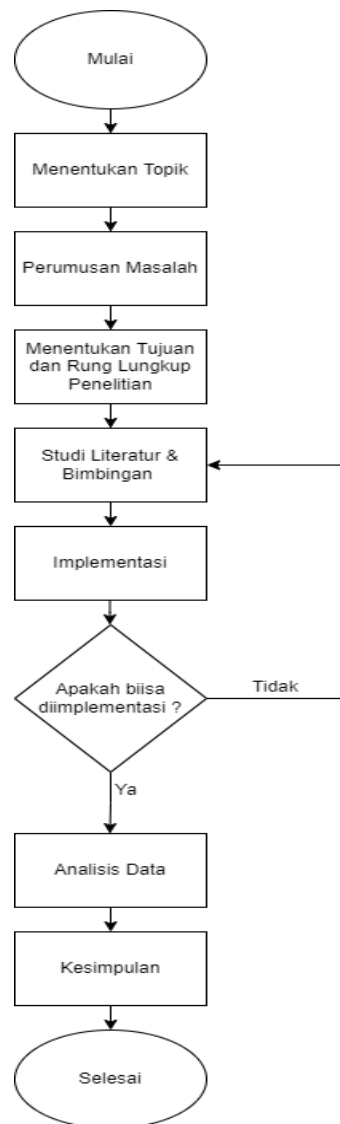
Tabel 3. 3 Spesifikasi perangkat lunak (Aplikasi)

No	Nama Aplikasi	Versi	Kegunaan
1	Virtual box	7.0	Virtualisasi
2	Docker container	20.10.21	<i>Container</i>
3	Nginx	1.23.2	<i>Web server</i> dan <i>load balancer</i>
4.	Mariadb	10.10.2	<i>Database server</i>

No	Nama Aplikasi	Versi	Kegunaan
4	Moodle	4.1	<i>Learning management system</i>
5	HTTPerf	2.3	Tools pengujian

3.3 Diagram Alir Penelitian

Penelitian ini memiliki beberapa tahapan agar mendapatkan sistem *load balancing* yang baik dan optimal.



Gambar 3. 1 Diagram alir penelitian

Gambar 3.1 menggambarkan alur dari penelitian yang dilakukan agar hasil yang didapatkan menjadi maksimal.

3.3.1 Menentukan topik

Tahap pertama yang dilakukan adalah menentukan topik penelitian berdasarkan permasalahan yang ditemukan.

3.3.2 Merumuskan masalah

Tahap kedua yang dilakukan adalah merumuskan masalah yang terdapat pada penelitian yang akan dilakukan, dengan menganalisis permasalahan pada topik yang dipilih, dan didapatkan bahwa *server learning management system* perlu diterapkan *metode load balancing* untuk mencegah terjadinya *down* pada *server*.

3.3.3 Menentukan tujuan dan ruang lingkup

Tahap ketiga adalah menentukan tujuan dan ruang lingkup penelitian, agar penelitian yang dilakukan mempunyai arah yang jelas. Tujuan pada penelitian ini adalah untuk mengetahui apakah *lms moodle* dapat diterapkan pada *docker container*, kemudian untuk mengetahui apakah *load balancing* pada *docker container* dapat mencegah *server down*, lalu melihat performa dari *metode load balancing pada container docker* dan untuk mengetahui bagaimana performa penggunaan algoritma *round robin* dan *least connection* pada *metode load balancing* dalam *container docker*.

3.3.4 Studi literatur

Tahap keempat adalah studi literatur dengan *me-review* dan menganalisis penelitian-penelitian terdahulu yang terkait dengan penelitian yang akan dilakukan, studi literatur yang dilakukan berasal dari sumber jurnal, skripsi atau dokumentasi resmi.

3.3.5 Implementasi dan pengujian projek

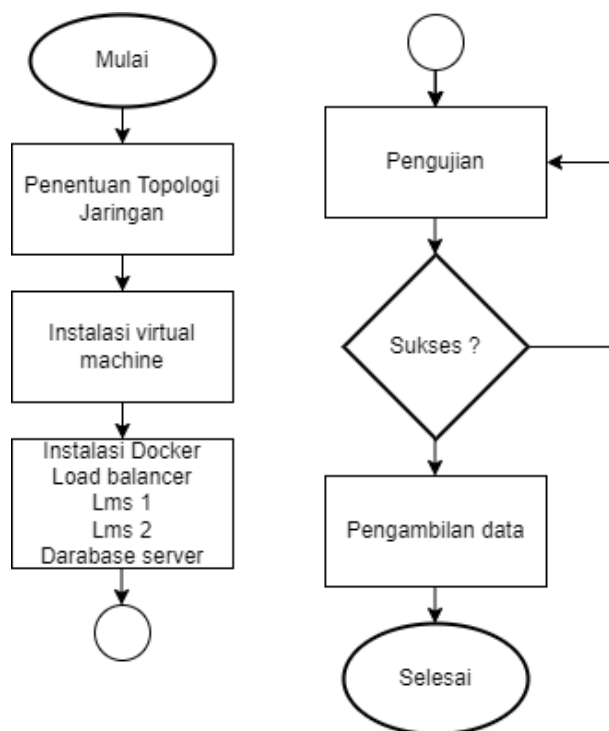
Tahapan ke lima adalah mengimplemtasikan *metode load balancing* terhadap *server learning management system moodle* yang akan diterapkan pada *docker container* menggunakan algoritma *round-robin* dan *least-connection*. Implementasi pada penelitian ini dilakukan dengan

beberapa tahap, tahap mulai dengan menentukan topologi jaringan, membuat *virtual machine*, menginstall docker, *deployment learning management system* dan *load balancing* lalu menguji sistem dengan 3 skema pengujian yaitu dengan 1000, 2000 dan 4000 total *request* dengan 100 *request* perdetik, pengujian dilakukan sebanyak 20 kali agar mendapatkan hasil yang maksimal.

Tabel 3. 4 Skema pengujian HTTPerf

No	Total <i>request</i>	Jumlah <i>request</i> perdetik	Jumlah pengujian
1	1000	100	20
2	2000		
3	4000		

Darri hasil pengujian, data akan rekap / dikumpulkan untuk selanjutnya dilakukan analisis data. Berikut merupakan alur dari implementasi sistem *load balancing learning management system moodle*.



Gambar 3. 2 Diagram alir implementasi

3.3.6 Analisis data

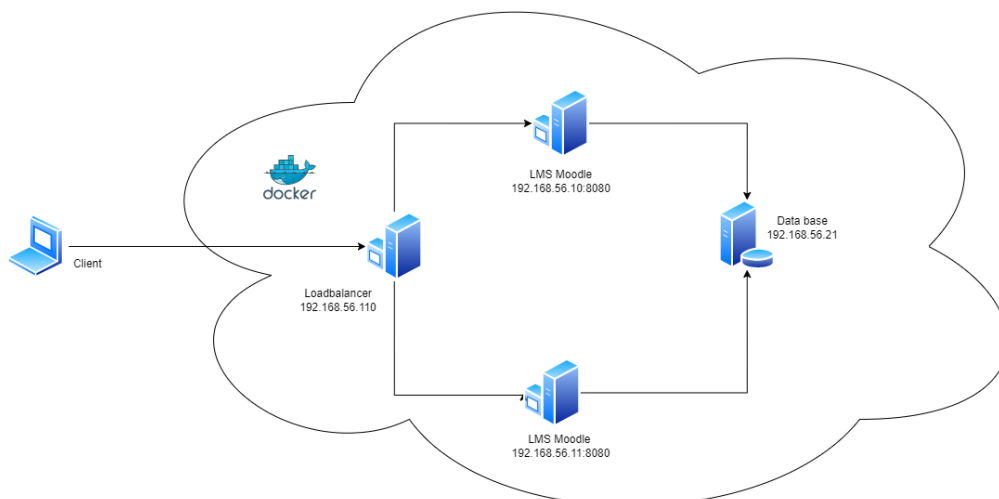
Tahap Menganalisis data dari penerapan penelitian yang dilakukan, untuk mengukur dan mengetahui performa dari metode yang telah ditepkan. Parameter yang akan di analisis adalah *throughput* , *response time*, *CPU utilization* dan *total error*.

3.3.7 Menentukan kesimpulan

Tahap terakhir adalah menentukan kesimpulan dari hasil analisis data yang telah dilakukan, sehingga didapatkan kesimpulan bagaimana performa dari penerapan metode *load balancing learning management system moodle* pada *docker container* menggunakan algoritma *round-robin* dan *least-connection*.

3.4 Rancangan Topologi Jaringan

Penelitian ini menggunakan satu client untuk simulasi pengujian tampilan *learning management system moodle*, kemudian menggunakan tiga *server*, dimana satu *server* sebagai *system load balancer* dan dua *server* lainnya sebagai *sever learning management system moodle* yang sebelumnya masing masing *server* sudah di install *container docker*.



Gambar 3. 3 Rancangan topologi jaringan

3.4.1 Instalasi dan konfigurasi docker

Instalasi docker menggunakan docker ce (*community edition*) yang digunakan sebagai wadah dari tiap tiap *service* yang akan dijalankan pada penelitian ini. Berikut merupakan tahapan dari instalasi docker *container*.

3.4.1.1 Install prerequisite packages

Langkah pertama adalah dengan mengkonfiguasi *prerequisite packages* untuk mengizinkan sistem operasi mengakses repositori docker melalui HTTPS.

```
server1@server1:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcurl4 python3-software-properties
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  ca-certificates curl libcurl4 python3-software-properties software-properties-common
5 upgraded, 1 newly installed, 0 to remove and 229 not upgraded.
```

3.4.1.2 Menambahkan GPG key

Menambahkan GPG key untuk memverifikasi keaslian perangkat lunak yang digunakan. lalu menambahkan *repository* docker ke sumber *apt* yang digunakan pada sistem.

```
server1@server1:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
server1@server1:~$
server1@server1:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Get:2 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Hit:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease
```

3.4.1.3 Verifikasi sumber instalasi

Verifikasi sumber instalasi untuk memastikan bahwa sumber instalasi docker menggunakan *repository* docker, bukan menggunakan *repository* os yang digunakan.

```
server1@server1:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:20.10.21~3-0~ubuntu-focal
  Version table:
   5:20.10.21~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.20~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.19~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
```

3.4.1.4 Langkah terakhir adalah melakukan instalasi docker ce.

```
server1@server1:~$ sudo apt install docker-ce -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin libseccomp2 pigz
  slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-scan-plugin pigz
  slirp4netns
The following packages will be upgraded:
  libseccomp2
1 upgraded, 7 newly installed, 0 to remove and 228 not upgraded.
Need to get 102 MB of archives.
After this operation, 383 MB of additional disk space will be used.
```

3.4.2 Instalasi dan konfigurasi database

Database server yang digunakan pada penelitian ini adalah mariadb-server, dengan beberapa environment variable yang digunakan yaitu nama *user*, *password*, nama database dan *network* dari container mariadb.

```
server2@server2:~$ docker run -d --name mariadb \
> --env ALLOW_EMPTY_PASSWORD=yes \
> --env MARIADB_USER=bn_moodle \
> --env MARIADB_PASSWORD=bitnami \
> --env MARIADB_DATABASE=bitnami_moodle \
> --network=host \
> -h 127.0.0.1 \
> bitnami/mariadb:latest
```

Gambar 3. 4 Instalasi *Data base Server* pada Docker

3.4.3 Instalasi dan konfigurasi *LMS Moodle*

Penelitian ini menggunakan dua *server* sebagai *server learning management system* yang masing masing dikelola dalam *container docker*. Konfigurasi dilakukan dengan menentukan nama, *port* dan *enviromen variable* dari lms yaitu *username*, *password*, nama *database*, ip host dari *server database* dan nama *images lms moodle* yang digunakan.

```
server2@server2:~$ docker run -d --name moodle \
> -p 8080:8080 -p 8443:8443 \
> --env ALLOW_EMPTY_PASSWORD=yes \
> --env MOODLE_DATABASE_USER=bn_moodle \
> --env MOODLE_DATABASE_PASSWORD=bitnami \
> --env MOODLE_DATABASE_NAME=bitnami_moodle \
> --env MOODLE_DATABASE_HOST=192.168.56.21 \
> bitnami/moodle:latest
```

Gambar 3. 5 Instalasi database lms moodle

3.4.4 Konfigurasi Loadbalancer round robin

Konfigurasi loadbalancing dilakukan didalam container nginx pada direktori */etc/nginx/conf.d/default.conf* dengan memasukan ip dari masing masing server *lms moodle*.

```
GNU nano 5.4
upstream backend {
server 192.168.56.11:8080; #backend1
server 192.168.56.10:8080; #backend2
}
```

Gambar 3. 6 Konfigurasi *load balancing round robin*

Konfigurasi dengan algoritma *least connection* dilakukan di direktori dan file yang sama dengan konfigurasi algoritma *round robin*, namu pada bagian isinya terdapat sedikit perbedaan, pada algoritma *least connection* perlu ditambahkan script “*least coon*” dibawah script “*upstream beckend*”

```

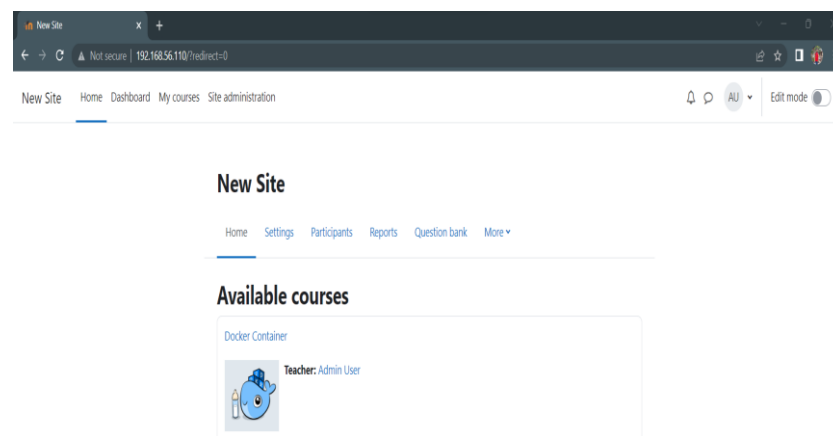
GNU nano 5.4
upstream backend {
    least_conn;
    server 192.168.56.11:8080; #backend1
    server 192.168.56.10:8080; #backend2
}

```

Gambar 3. 7 Konfigurasi *load balancing least connection*

3.4.5 Pengujian instalasi LMS Moodle

Pengujian *load balancer* dilakukan dengan mengakses *ip server load balancer* melalui *pc client* menggunakan *web browser*. Hasil dari akses *ip load balancer* akan langsung mengarahkan pada *server lms moodle* dan menampilkan tampilan *lms moodle*. Tampilan pada gambar 3.5 merupakan tampilan setelah *login* dan didalamnya sudah terdapat *course* yang sudah coba dibuat.

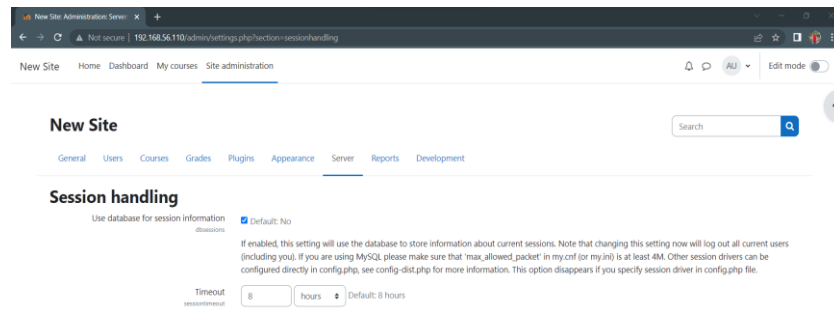


Gambar 3. 8 Uji tampilan LMS moodle

3.4.5 pengaturan *session*

Session merupakan metode yang digunakan untuk mempertahankan informasi pada *web* dan menyimpan data pada server. *Session* dapat digunakan untuk mempertahankan *user login* agar saat user membuka halaman lain, *user* tersebut tidak diharuskan melakukan *login ulang*. *Session* yang diatur pada penelitian ini ada dua, yang pertama *session database* dan

yang kedua adalah *session file driver*. Pengaturan *session database* dilakukan pada tampilan *web learning management system moodle* yang terletak pada *Site administration – server*



Gambar 3. 9 Konfigurasi *session database lms moodle*

Konfigurasi *session file driver* terletak pada *container moodle* di direktori `/opt/bitnami/moodle/` pada file `config.php`. konfigurasi dilakukan dengan menambahkan script berikut :

```
$CFG->session_handler_class = '\core\session\file';
$CFG->session_file_save_path = $CFG->dataroot.'/sessions';
```

Gambar 3. 10 Konfigurasi *session file driver lms moodle*