

## BAB 3

### METODE PENELITIAN

#### 3.1 PERANGKAT YANG DIGUNAKAN

Untuk mendukung penelitian, diperlukan penggunaan perangkat *hardware* dan *software*.

##### 3.1.1 Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan untuk merealisasikan sistem yang akan dibangun adalah sebuah laptop. Tabel 3.1 menunjukkan spesifikasi perangkat keras yang digunakan.

**Tabel 3.1 Spesifikasi Perangkat Keras**

OS	Windows 11
<i>Processor</i>	12th Gen Intel(R) Core(TM) i5-1235U 1.30 GHz
RAM	8 GB
<i>Hard disk</i>	500 GB

##### 3.1.2 Perangkat Lunak (*Software*)

Tabel 3.2 menunjukkan perangkat lunak yang digunakan sebagai alat dan aplikasi untuk membantu dalam implementasi sistem yang akan dibangun.

**Tabel 3.2 *Software Tool* dan Aplikasi**

No	Nama Software	Versi	Lokasi Penggunaan	Fungsi
1	<i>Amazon Elastic Compute Cloud (EC2)</i>	-	<i>AWS Console</i>	<i>Virtualisasi server</i>
2	MariaDB	10.6.11	<i>AWS Console</i>	<i>Database server</i>
3	<i>Nginx</i>	1.22.1	<i>AWS Console</i>	<i>Web server 1</i>
4	<i>Apache (HTTPD)</i>	2.4.55	<i>AWS Console</i>	<i>Web server 2</i>
5	PHP	7.4.33	<i>AWS Console</i>	Bahasa pemrograman
7	<i>Apache Jmeter</i>	5.5	<i>Local laptop</i>	Pengukuran QoS
8	<i>Wireshark</i>	4.0.5	<i>Local laptop</i>	Pengukuran QoS

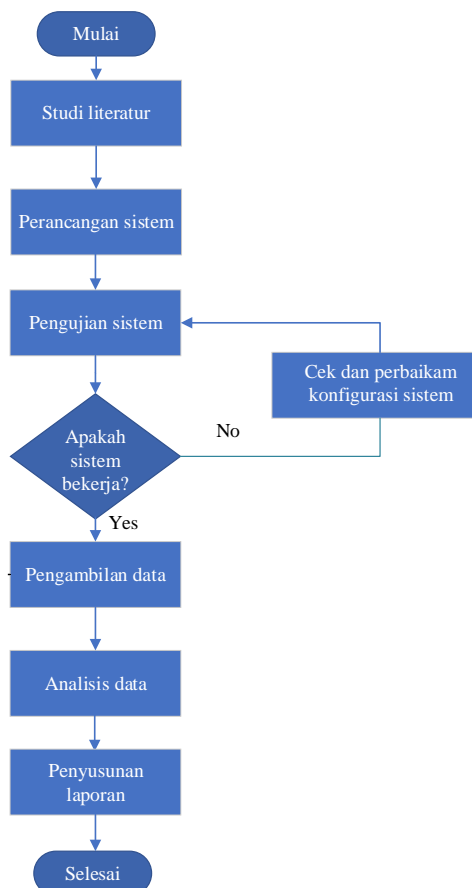
Tabel 3.3 menunjukkan spesifikasi yang digunakan pada *server Amazon Elastic Compute Cloud (EC2)*.

**Tabel 3.3 Spesifikasi Server EC2**

OS	Amazon Linux 2 AMI (HVM)
Processor	2 vCPU @ 2.5Ghz
RAM	1 GiB
Storage	8 GB

### 3.2 ALUR PENELITIAN

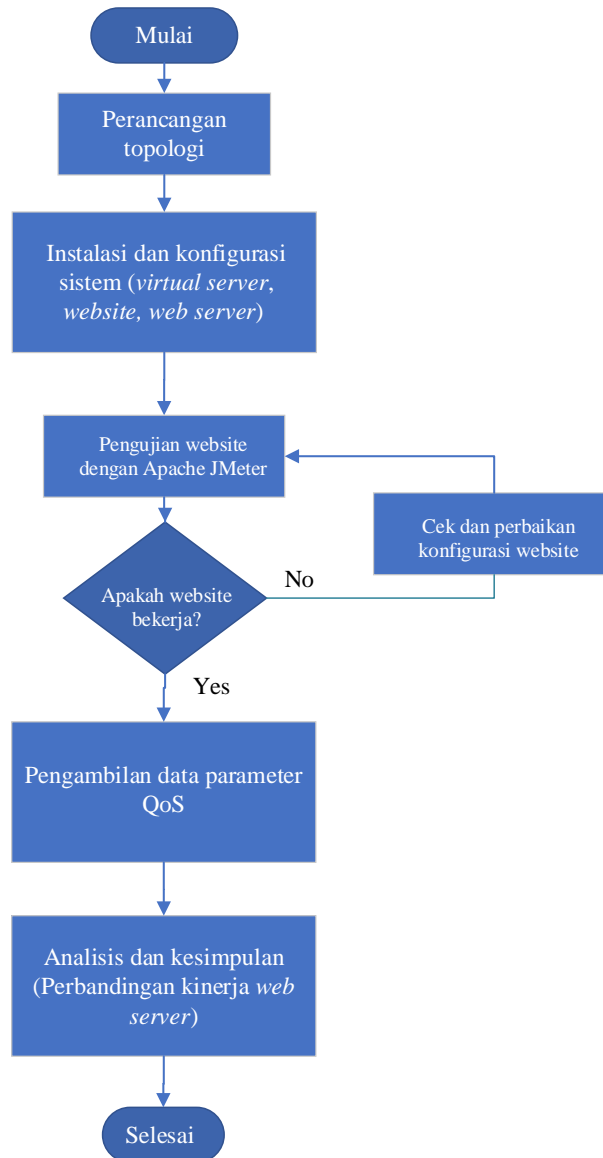
Diagram alur penelitian menunjukkan proses penyusunan penelitian. Gambar 3.1 menunjukkan diagram alur yang akan menjelaskan struktur penelitian.



**Gambar 3.1 Diagram Alur Penelitian**

### 3.3 ALUR PENGUJIAN

Semua langkah yang diambil untuk membuat sistem pengujian untuk penelitian dijelaskan dalam diagram alur pengujian. Diagram alur akan menjelaskan proses pengujian, seperti yang ditunjukkan pada gambar 3.2.



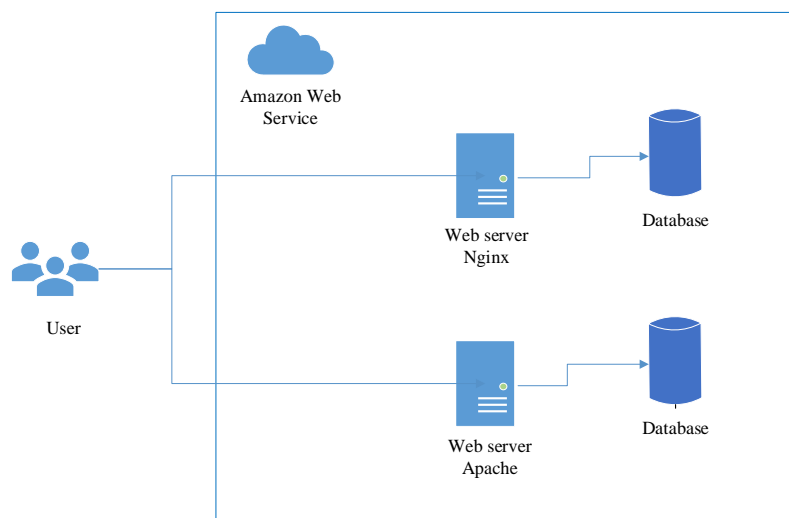
**Gambar 3.2 Diagram Alur Pengujian**

Gambar 3.2 menunjukkan *flowchart* yang menggambarkan proses pembuatan sistem pada penelitian untuk mencapai hasil penelitian. Dimulai dengan perancangan topologi jaringan, yang digunakan sebagai dasar untuk implementasi *website* menggunakan *web server Nginx* dan *Apache*. Setelah

perancangan topologi jaringan selesai, beberapa *software* diinstal dan dikonfigurasi, seperti menyiapkan *virtual server* pada AWS dengan EC2, mengkonfigurasi *web server*, dan memulai implementasi *website*. Setelah langkah-langkah tersebut selesai, pengujian sistem yang sudah dikonfigurasi akan dilakukan. *Apache Jmeter* adalah alat untuk menguji sistem. Jika pengujian berhasil, data akan diambil dari parameter Qos penelitian, yaitu *throughput, packet loss, delay, jitter, dan response time*. Jika tidak, pengujian akan diulang. Setelah semua langkah dilakukan, langkah selanjutnya adalah menganalisis parameter dari data yang telah diperoleh untuk membuat kesimpulan tentang penelitian dan tujuan penelitian.

### 3.4 TOPOLOGI JARINGAN

Topologi jaringan yang digunakan pada penelitian ini terdiri dari 1 *client* yang digunakan untuk *remote* serta melakukan simulasi pengujian data. 2 *server virtual* menggunakan EC2 di *Amazon Web Services* dengan sistem operasi *Amazon Linux 2* dan 2 RDS atau *database* menggunakan *mariadb*. 1 server digunakan untuk membangun *website* dengan web server *Nginx* serta 1 *database server* untuk menyimpan data serta file yang dimasukkan pada web server *Nginx*, lalu 1 server lainnya digunakan untuk membangun *website* dengan web server *Apache* dan juga pendukung 1 *database* untuk menyimpan file yang dimasukkan pada web server *Apache* seperti yang ditunjukkan pada gambar 3.3.



**Gambar 3.3 Topologi Jaringan**

### 3.5 SKENARIO PENGUJIAN

Berikut ini adalah skenario pengujian yang akan dilakukan pada penelitian:

#### 3.5.1 Membuat Skenario Jaringan

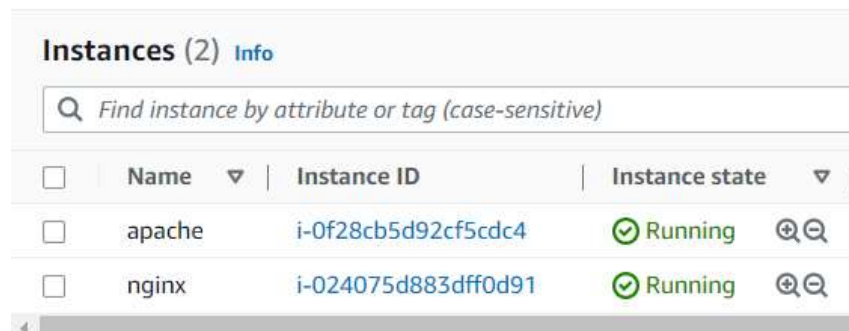
Berdasarkan gambar 3.3 ada 2 server *virtual* yang akan di-*deploy* di *Amazon Web Services*. 2 server tersebut akan di *deploy* dengan *amazon EC2* (*Amazon Elastic Compute Cloud*), pada server akan di konfigurasi *web server Apache* dan *Nginx* untuk kebutuhan implementasi *website e-learning* untuk di test kinerjanya. *Database server* digunakan untuk penyimpanan file atau data yang terhubung dengan *websitenya*. *Client* berfungsi menjalankan aplikasi *tools Apache Jmeter* untuk mengirimkan *request* ke *website e-learning* serta melakukan *capture* hasil datanya menggunakan *wireshark* yaitu dengan parameter QoS yang diuji seperti *throughput*, *packet loss*, *delay*, *jitter*, dan *response time*.

#### 3.5.2 Uji Coba Website

Sebelum dilakukan uji coba untuk mengetahui perbandingan kinerja *web server* pada *website Amazon Web Services*, peneliti terlebih dahulu melakukan konfigurasi *virtual server* dan *web server* untuk pendukung implementasi *website e-learning*.

##### A. Konfigurasi *virtual server* pada *Amazon Web Services*

*Virtual server* dibuat dengan *amazon EC2* pada *Amazon Web Services*, peneliti menyediakan 2 server dengan os *Amazon linux*. 2 server ini digunakan untuk konfigurasi *web server Nginx* dan *Apache* yang akan diimplementasikan ke *website e-learning*. Pada gambar 3.4 merupakan contoh 2 server yang sudah di *create*.



The screenshot shows the Amazon EC2 console 'Instances (2)' page. It features a search bar and a table with columns for Name, Instance ID, and Instance state. Two instances are listed: 'apache' with ID 'i-0f28cb5d92cf5cdc4' and 'nginx' with ID 'i-024075d883dff0d91', both in a 'Running' state.

<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾
<input type="checkbox"/>	apache	i-0f28cb5d92cf5cdc4	Running
<input type="checkbox"/>	nginx	i-024075d883dff0d91	Running

Gambar 3.4 Contoh Amazon EC2

## B. Konfigurasi *web server Nginx* dan *Apache*

*Web server Nginx* dan *Apache* akan diinstall pada *virtual server* yang sudah dibuat. Perintah yang digunakan untuk menginstall *web server Nginx* pada os *Amazon linux* sebagai berikut:

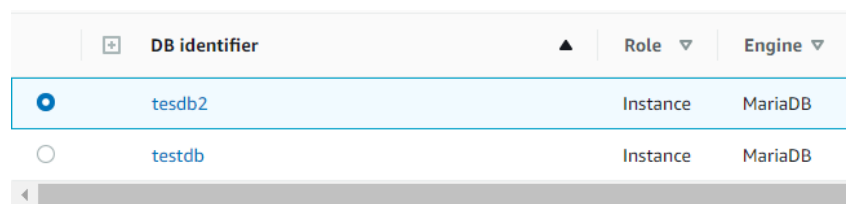
```
amazon-linux-extras install Nginx1
```

Sedangkan, perintah yang digunakan untuk menginstall *web server Apache* pada os *Amazon linux* adalah sebagai berikut:

```
yum install httpd
```

## C. Konfigurasi *database server*

*Database server* pada penelitian ini menggunakan 2 *virtual server* yang terpisah dengan *server* yang digunakan untuk *web server*, pada AWS ada *tools* untuk *create server database* yaitu *Amazon Relational Database Service (RDS)* jadi untuk *database server*nya nanti akan di *create* menggunakan *tools RDS*. Tujuan *server database* dipisah karena agar *database* hanya digunakan untuk menyimpan data-data yang diinput oleh *client* pada *web server*. *Database* yang digunakan pada penelitian ini adalah *MariaDB* yang dimana *database* ini adalah salah satu bagian dari *database MySQL*. Keunggulan *database MariaDB* adalah dapat menyimpan *file content* dan beberapa data-data hanya dalam satu *database server*. Pada gambar 3.5 merupakan contoh 2 *server database* yang sudah di *create* di AWS.



DB identifier	Role	Engine
tesdb2	Instance	MariaDB
testdb	Instance	MariaDB

**Gambar 3. 5 Contoh RDS**

Untuk *login* ke *database* yang sudah di *create* bisa dilakukan dari *web server*nya, yaitu dengan perintah berikut:

```
yum install mariadb  
mysql host=testdb.cfygzm8w5yg6.apsoutheast1.rds.  
amazonaws.com --user=root --password=admin1
```

#### D. Konfigurasi *website*

Untuk konfigurasi *website* dilakukan dengan melakukan *get application source* nya terlebih dahulu dari *github* dengan perintah berikut:

```
yum install git-core -y
git clone https://github.com/fulqiysc/elearning\_test.git
```

Setelah itu dilakukan penyesuaian koneksi *database* dengan perintah berikut:

```
CREATE DATABASE learning;
use learning;
source learning.sql;
Show tables;
```

Kemudian setelah *get application source* dan penyesuaian koneksi *database* dilakukan, maka konfigurasi dilanjutkan dengan menyesuaikan file *.conf* dari *web server Nginx* dan *Apache* nya.

#### E. Pengujian QOS *website* dengan *web server Nginx* dan *Apache* berdasarkan *request time*

Pengujian ini bertujuan untuk mengetahui perbandingan kinerja *website* yang menggunakan *web server Nginx* dan *Apache* dengan jumlah *request* yang berbeda-beda. Peneliti akan melakukan pengujian sebanyak 3 skenario rentang waktu dengan 8 variasi jumlah koneksi, pada masing-masing skenario dilakukan pengujian sebanyak 10 kali. Hal ini bertujuan untuk mendapatkan data QOS dari parameter yang telah ditentukan. Pengujian dilakukan dengan *software Apache Jmeter* untuk melakukan *request* dan *output* akan dihitung dengan *wireshark*.

##### 1. Skenario pertama

Skenario pertama pada pengujian *website* menggunakan *Apache Jmeter* dengan rentang waktu 5 detik yang dilakukan 8 kali pengujian pada jumlah koneksi yang berbeda. Masing-masing pengujian dilakukan 10 kali pengambilan data. Untuk detail dari skenario pertama ditunjukkan pada tabel 3.4.

**Tabel 3.4 Detail Skenario Pertama**

No	Jumlah Koneksi	Rentang Waktu	Jumlah pengujian	Parameter yang diuji
1	200	5 detik	10	<i>Throughput</i> <i>Delay</i> <i>Jitter</i> <i>Packet loss</i> <i>Response time</i>
2	400	5 detik	10	
3	600	5 detik	10	
4	800	5 detik	10	
5	1000	5 detik	10	
6	1200	5 detik	10	
7	1400	5 detik	10	
8	1600	5 detik	10	

2. Skenario kedua

Skenario kedua pada pengujian *website* menggunakan *Apache Jmeter* dengan rentang waktu 15 detik yang dilakukan 8 kali pengujian pada jumlah koneksi yang berbeda. Masing-masing pengujian dilakukan 10 kali pengambilan data. Untuk detail dari skenario pertama ditunjukkan pada tabel 3.5.

**Tabel 3.5 Detail Skenario Kedua**

No	Jumlah Koneksi	Rentang Waktu	Jumlah pengujian	Parameter yang diuji
1	200	15 detik	10	<i>Throughput</i> <i>Delay</i> <i>Jitter</i> <i>Packet loss</i> <i>Response time</i>
2	400	15 detik	10	
3	600	15 detik	10	
4	800	15 detik	10	
5	1000	15 detik	10	
6	1200	15 detik	10	
7	1400	15 detik	10	
8	1600	15 detik	10	

3. Skenario ketiga

Skenario ketiga pada pengujian *website* menggunakan *Apache Jmeter* dengan rentang waktu 30 detik yang dilakukan 8 kali pengujian pada jumlah koneksi yang berbeda. Masing-masing



pengujian dilakukan 10 kali pengambilan data. Untuk detail dari skenario pertama ditunjukkan pada tabel 3.6.

**Tabel 3.6 Detail Skenario Ketiga**

No	Jumlah Koneksi	Rentang Waktu	Jumlah pengujian	Parameter yang diuji
1	200	30 detik	10	<i>Throughput</i> <i>Delay</i> <i>Jitter</i> <i>Packet loss</i> <i>Response time</i>
2	400	30 detik	10	
3	600	30 detik	10	
4	800	30 detik	10	
5	1000	30 detik	10	
6	1200	30 detik	10	
7	1400	30 detik	10	
8	1600	30 detik	10	