BAB 3

METODOLOGI PENELITIAN

3.1 Perangkat Yang Digunakan

Perangkat yang digunakan dalam penelitian ini meliputi perangkat keras (hardware) dan perangkat lunak (Sofware).

3.1.1 Perangkat Keras (Hardware)

Perangkat keras yang digunakan pada penelitian ini menggunakan 1 laptop sebagai *remote server* pada *cloud* AWS dengan spesifikasi terdapat pada Tabel 3. 1.

	OS	Windows 10
Laptop	Processor	AMD A9-9420 RADEON R5
	RAM	4 GB

Tabel 3. 1 Spesifikasi Perangkat Keras

3.1.2 Perangkat Lunak (Software)

Perangkat lunak dalam penelitian ini menggunakan perangkat *virtual* dan beberapa aplikasi.

3.1.2.1 Perangkat Virtual

Pada penelitian ini terdapat beberapa perangkat virtual yang dibangun pada *cloud Amazon Web Service* (AWS) yaitu satu sebagai *load balancer* dan tiga *server* LMS. Sepesifikasi perangkat lunak tercantum pada Tabel 3.2

Tabel 3.	25	Spesifikasi	Perangkat	Virtual
I aber 5.		pesimasi	I CI anghat	v II tuai

Nama Perangkat Virtual	Spesifikasi			
	OS	Ubuntu Server 18.04		
Load Balancer	RAM	2 GB		
	Harddisk	20 GB		
	Alamat IP	172.31.6.228		
	OS	Ubuntu Server 22.04		
LMS	RAM	16 GB		
	Harddisk	50 GB		
	Alamat IP	172.31.100.123		

Nama Perangkat Virtual	Spesifikasi		
	OS	Ubuntu Server 22.04	
LMS	RAM	16 GB	
	Harddisk	50 GB	
	Alamat IP	171.31.100.208	
	OS	Ubuntu Server 22.04	
IMS	RAM	16 GB	
LIVIS	Harddisk	50 GB	
	Alamat IP	172.31.100.251	

3.1.2.2 Software Tools

Perangkat lunak sebagai *tools* dan aplikasi yang digunakan pada penelitian ini ditunjukkan pada Tabel 3.3

No	Software	Versi	Fungsi
1	Cloud AWS		Resource
2	Docker	19.03	Tool container
3	Moodle	3.11	LMS
4	Mariadb	10.9.1	Database Server
5	Fortigate Firewall	7.2.3	Load Balancing
6	Apache benchmark	2.6	<i>Tools</i> Pengujian <i>Load</i> <i>balancing</i> LMS
7	Wireshark	3.2.4	Pengukuran QoS

 Tabel 3. 3 Software dan Tools

3.2 Alur Penelitian

Alur penelitian yang dilakukan pada penelitian ini akan digambarkan pada Gambar 3.1. Dalam alur penelitian ini, akan tergambar langkah-langkah sistematis yang diikuti selama proses penelitian. Dimulai dengan menentukan topologi jaringan, selanjutnya membangun infrastruktur *cloud computing* pada *provider Amazon Web Service* (AWS). Tahap berikutnya membuat instance untuk ketiga LMS. Berikutnya dengan membuat system *load balancing* yang berfungsi membagi beban kepada *server* LMS dan setela berhasil, maka dilakukan pengambilan data menggunakan parameter seperti *throughput, delay, jitter, packet loss,* dan Cpu *usage*. Setelah selesai, data diberikan analisis dan kesimpulan.



Gambar 3. 1 Diagram alur penelitian

Alur penelitian yang akan dilakukan digambarkan pada Gambar 3. 1. Hal pertama yang dilakukan membuat topologi jaringan sebelum membuat simulasi, yang bertujuan sebagai gambaran dalam melakukan implementasi *docker container* pada LMS berbasis *cloud computing*. Selanjutnya menerapkan

infrastruktur *cloud computing* dengan menggunakan *Virtual Private Cloud* (VPC) yang bertujuan untuk membuat dua jaringan secara terpisah. Setelah infrastruktur *cloud* selesai dikerjakan, hal selanjutnya yang perlu dilakukan yaitu membuat *instance* sebanyak 2 *instance* dengan satu sebagai *load balancer* dan *instance* kedua untuk meng*install* tiga 3 *server* LMS yang dibedakan *port*nya. Selanjutnya konfigurasi *load balancing*, apabila berhasil maka selanjutnya melakukan pengujian berdasarkan parameter berupa *throughput, packet loss, delay, jitter* dan CPU *usage*. Setelah data performansi dikumpulkam, maka selanjutnya menganalisisis hasil pengujian yang dilakukan untuk mendapatkan hasil performansi pada LMS menggunakan *docker container* dengan *load balancing* sebagai pembagi *traffic*.

3.3 Topologi Jaringan

Sebelum melakukan penelitian ini, diperlukan topologi jaringan sebagai gambaran implementasi docker container pada LMS dan load balancing sebagai pembagi traffic digambarkan pada Gambar 3.2. langkah awal yang diperlukan adalah menyusun topologi jaringan sebagai landasan implementasi docker container pada Learning Management System (LMS) serta konsep load balancing yang bertindak sebagai alat pembagi lalu lintas. Dalam konteks topologi ini, terdapat beberapa komponen yang saling berinteraksi. Pertama, terdapat satu entitas sebagai *client* yang akan digunakan untuk melakukan uji performansi. Selanjutnya, ada Virtual Private Cloud (VPC) yang memiliki dua jaringan terpisah dan memungkinkan akses internet ke kedua instance yang ada dalam jaringan tersebut. Komponen pertama dari instance adalah fortigate firewall yang berperan ganda sebagai load balancing dan router. Sebagai load balancer, fortigate firewall bertanggung jawab untuk meratakan beban trafik agar distribusi lalu lintas ke *instance* LMS berjalan seimbang. Selain itu, *fortigate* juga berfungsi sebagai router yang menghubungkan semua jaringan pada *instance*, memastikan koneksi yang tepat dan aman.Komponen kedua adalah instance LMS yang berperan sebagai pusat Learning Management System. Dengan begitu, topologi ini menggambarkan infrastruktur yang perlu diimplementasikan untuk menjalankan penelitian.



Gambar 3. 2 Topologi jaringan

3.4 Skenario Pengujian

3.4.1 Implementasi jaringan

Implementasi jaringan dalam penelitian ini menggunakan tiga server LMS sebagai layanan *e-learning*, satu *client* sebagai simulasi melakukan pengujian jaringan. *Load balancing* pada server akan membagikan beban kerja pada setiap server LMS. VPC berfungsi untuk membuat dua jaringan secara terpisah sekaligus memberikan akses internet untuk kedua *instance* yaitu *fortigate firewall* dan server LMS. *Client* akan digunakan untuk melakukan pengujian performansi dengan parameter berupa *throughput, packet loss, delay, jitter, dan CPU Usage* menggunakan *apache benchmark* dan *wireshark* untuk menguji server LMS.

3.4.2 Implementasi Load Balancing Pada AWS

Implementasi *load balancing* pada penelitian ini menggunakan *software fortigate firewall* dengan membagikan beban pada tiga *server* LMS. Sebelum mengimplementasikan *load balancing*, diperlukan konfigurasi *Virtual Private*

Cloud (VPC) sebagai infrastruktur jaringan yang akan memberikan akses internet pada *fortigate firewall* dan *server* LMS.

3.4.2.1. Konfigurasi Virtual Private Cloud (VPC)

Virtual Private Cloud (VPC) merupakan salah satu jenis layanan Infrastructure as a Service (IaaS) melalui platform cloud AWS sebagai infrastructure network. VPC memiliki fungsi untuk membuat dua jaringan secara terpisah. Jaringan tersebut berupa Wide Area Network (WAN) dan Local Area Network (LAN). Jaringan WAN dihubungkan ke fortigate firewall yang berfungsi untuk menghubungkan jaringan lokal dengan internet. Dengan menggunakan jaringan WAN, fortigate firewall dapat memonitor dan mengelola lalu lintas data yang masuk dan keluar dari jaringan lokal. Sedangkan untuk jaringan LAN hanya dihubungkan ke server. Selain itu, VPC juga menggunakan internet gateway agar semua instance yang dibuat memiliki akses internet. Pada VPC juga menggunakan dua route berupa public route dan private route. Public route dihubungkan ke internet gateway agar semua routing diarahkan ke-default route (internet), sedangkan private route hanya digunakan untuk server yang berada disubnet LAN. Tujuan dibuatkan VPC adalah memberikan akses internet untuk kedua instance. Instance pertama ialah fortigate firewall berfungsi sebagai load balancing sekaligus sebagai router yang akan menghubungkan pada semua jaringan. Intsance kedua berfungsi sebagai LMS yang di dalamnya sudah terinstall moodle dan database server.

3.4.2.2. Konfigurasi Fortigate Firewall

Konfigurasi *fortigate firewall* dilakukan pada satu *instance* EC2 AWS yang telah dipersiapkan. *Fortigate firewall* merupakan salah satu jenis layanan *Software as a Service* (SaaS) yang ditawarkan oleh *platform cloud* AWS. Konfigurasi awal yaitu mencari *fortigate firewall* pada *menu marketplace* AWS. Setelah itu memilih opsi dan versi untuk kebutuhan *fortigate firewall* dengan membuat *instance* sebagai wadah instalasi dan dihubungkan dengan VPC serta menggunakan jaringan WAN agar memiliki akses internet. Berikut Gambar 3. 3 mengenai instalasi *fortigate firewall*.

roduct Detail Subscribe	Configure Launch
aunch this s	oftware
ongratulations! An in	istance of this software is successfully deployed on EC2!
MI ID: ami-0a423117r2f	585-41 (View Laurch Configuration Details)
ou can view this instance sage fees apply when the	on EC2 Console. You can also view all instances on Your Software. Software and AWS hourly instance is running and will appear on your monthly bill.
r can launch this configur	ation again below or go to the configuration page to start a new one.
i can launch this configur Configuration details	ation again below or go to the <mark>configuration page</mark> to start a new one.
i can launch this configur Configuration details Fulfillment option	ation again below or go to the configuration page to start a new one. i 64-bit (x86) Amazon Machine Image (AMI)
i can launch this configur Configuration details Fulfillment option	ation again below or go to the configuration page to start a new one. 64-bit (x86) Amazon Machine Image (AMI) Fortinet FortiGate Next-Generation Firewall runnetsg on cflukinge
i can launch this configur Configuration details Fulfillment option Software version	ation again below or go to the configuration page to start a new one. 64-bit (x86) Amazon Machine Image (AMI) Fortinet FortiGate Next-Generation Firewall names on clickinge 7.2.3
i can launch this configur Configuration details Fulfillment option Software version Region	ation again below or go to the configuration page to start a new one. 64-bit (x86) Amazon Machine Image (AMI) Fortinet FortiGate Next-Generation Firewall running on cEckinge 7.2.3 Asia Pacific (Jakarta)

Gambar 3. 3 Instalasi Fortigate Firewall

Selanjutnya, pada instance *fortigate firewall* membuat *allocate elastic* IPs yang bertujuan agar *fortigate firewall* dapat IP *public* secara *dhcp*. Kemudian dibagian *network interface* di-*attach* ke *fortigate firewall* agar dapat *mapping* antara *port fortigate firewall* dengan *subnet* yang digunakan. Berikut ditunjukkan pada Gambar 3.4 mengenai *network interface* pada *instance fortigate firewall*

Netwo	ork interfaces (1	/2) 1nfo								
Q. Set	arch									
	Description	Ψ	Instance ID	v	Status	Ψ.	Public IPv4 address	v	Primary private IPv4 address	v
rface	8 - 2		i-01b84660b56011498		⊘ In-use		108.137.148.200		10.0.1.132	
rface	lan-interface		i-01b84660b56011498		⊘ In-dise		-		10.0.2.155	
N/N/										

Gambar 3. 4 Network interface pada instance Fortigate Firewall

Selanjutnya *login* pada *fortigate firewall* menggunakan IP *public* yang didapatkan dari IP NATnya dari IP WAN.

+ + 0 A Minute 1					
Forthan-WAS4-ANS	- = q				
Derfound Network Interferen	TO FortiCate VM64.AW	5 1 8 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3			
DNS JOWN	+ CompNew+	es # Denie # letapole mechan	forch.		Q
SEFWAN	Name #	Type #	Manuers #	(PNetmasii: 2	Administrative Access #
State Readers	E > 812.5ad Appropriate	0			
- KIP	B+ fortline	B• 002.3et Aggregate		Deskamd to FortNwitch	MNG Second Address Compaction
0.01	🔁 🛤 Physical infortance 🔮				
Ruth Olives	E LAN (port25	🚊 Preskal litterfært		8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	PresC HITPS SEL
Disponition					Superl'Test:
 Antipic Copieta Security Publics Unit & Authentication 	WWW(part1)	📕 Physical Interface		300,1,130,255,255,255,0	MING GETTER SELF GETER

Gambar 3. 5 Interface Fortigate Firewall

Pada Gambar 3. 5 akan menampilkan dari *interface fortigate firewall* yang sudah login menggunakan IP *public*.

3.4.2.3. Konfigurasi Moodle LMS

Konfigurasi *Moodle* dilakukan pada satu *instance* EC2 AWS yang telah dipersiapkan sebagai *server* LMS. Konfigurasi awal yaitu melakukan instalasi *docker* sebagai *tool container* untuk wadah *moodle* dan *database server*. Selanjutnya, *pull image moodle* dan *pull image mariadb* pada *container* yang dihubungkan oleh *docker network*. Setelah berhasil melakukan instalasi *Moodle*, langkah selanjutnya yaitu melakukan *setting* pada *instance moodle* dengan *instance fortigate firewall* agar saling terhubung. Berikut dibawah Gambar 3.5. instalasi tiga *moodle* dan satu *database server*.

	and A alar				- 0 x
rest#ip-18-8-1	-64:/home/uburtu# docker	1.95			
CONTAINER 1D	INTE	CORRAND	CREATED	STATUS	PORTS
d8df2585a478 3443->8443/tcp	bitnami/moodle:latest moodle3	*/opt/bitmami/script_*	About a minute ago	Up About a minute	0.0.0.0.0.0.0-0000/tcp, :::03-0000/tcp, 0.0.0.0-3443-4043/tcp, :::
7x1138342659 2443->8443/tcp	bitrumi/mosfie:latest moofie2	*/spt/bitessi/script_*	About a simils age	Up About a winute	0.0.0.0.02>>0000/tcp, :::02>0000/tcp, 0.0.0.02>0000/tcp, :::
4bekc17a8864 1443-+8443/tcp	bitnami/mocdle:latest mocdle1	*/ept/bitrael/script.*	5 aboutes ago	Up 5 admites	4.4.8.8:31-+8880/tcp, :::01-+8089/tcp, 0.8.8.4:1443-+8443/tcp, :::
88968cd9F355	mariado moofiedo -66: /home/uburtad	'decker-entrypoint.s.*	ó númetes ago	Up 6 minutes	3380/tcp
anon terrai	A TRADE A STATE AND				

Gambar 3. 6 Instalasi tiga *Moodle*

3.4.2.4. Konfigurasi dan Ujicoba Akses Load Balancing

Load balancing dikonfigurasi pada satu instance EC2 AWS yang dipersiapkan sebagai load balancer. Load balancing memakai fortigate firewall

yang telah diinstalasi pada *instance* EC2 AWS. Konfigurasi awal yang dilakukan dengan membuat ssh-moodle pada *virtual* IP yang bertujuan agar *server moodle* (yang tidak punya IP *public*) dapat diakses dari *public* melalui *fortigate firewall*. Selanjutnya membuat acces-to-moodle pada *firewall policy* yang betujuan untuk mengatur lalu lintas dari jaringan WAN (internet) diperbolehkan mecapai jaringan LAN yang sudah terhubung dengan *server moodle*. Langkat berikutnya dengan membuat acces-inet *pada firewall policy* yang bertujuan supaya *server moodle* mempunyai akses internet. Berikut dibawah Gambar 3. 7 konfigurasi *fortigate firewall* untuk mengakses server LMS

≣ Q.						
+Create New 🖋 Edit	B Delete	olicy Lookup Search			Q	
Name	Source	Destination	Schedule	Service	Action	NAT
🖃 🔤 LAN (port2)	N (port1) 📵					
ACCESS-INET	all all	🔳 all	Co always	ALL ALL	✓ ACCEPT	CENabled
□ WAN (port1) → ■ LAI	N (port?) 🕚					
ACCESS-TO-MOODLE	🖾 all	🚳 SSH-MOODLE	Co always	D ALL	ACCEPT	O Disabled
Implicit 1						

Gambar 3. 7 Konfigurasi Fortigate Firewall untuk akses server LMS

Kemudian sebelum ke konfigurasi *load balancing* dilanjutkan dengan membuat tiga *virtual* IP uktuk mengakses *server* LMS yang dibedakan *port*-nya. Berikut dibawah Gambar 3. 8 Konfigurasi tiga *virtual* IP untuk mengakses server LMS.

≡ q.					
+Could Note - /	Lift & Chron B Linker Search		Q.		
Name #	Details #	interfaces #	Services 0	Hef. 9	Hit Court #
🗄 IP+4 Victual IP 💿					
25H MOGDLE	$172.31.6.228 \rightarrow \odot 0.0.31.000.523 (009, 2223 \rightarrow 22)$	🛎 purti		(1)	210
te woodw1	$172.31.8.228 \rightarrow 172.31100.123(90\%)(11 \rightarrow 0.0)$	🗮 porti		1	2
S mode2	$172.71 \pm 228 \rightarrow 172.71100.127(309.02 \rightarrow 0.0)$	🗶 purti		3	17
Cuboon @	172.31.#.228 → 172.31.100.123 (VCP: 83 → 83)	m ports		1	0

Gambar 3. 8 Konfigurasi tiga virtual IP untuk server LMS

Setelah itu, uji coba akses *server* LMS menggunakan IP *public* dengan menambahkan port untuk membedakan *server* mana yang sedang diakses. Berikut dibawah Gambar 3. 9 uji coba akses *server* LMS 1 menggunakan IP *public*.

+ + C & 12517792	1
LMS-moodle 1 Implicitien -	Thu are not bygged in (log in)
moodle	
Welcome to LMS-moodle.my.id The LMS-Moodle allows you to take leading on the reactive to a new level, with velocative index to the treatment and ends.	Login 'Userune
 Easily ones: upload, dig, manage, publish and deliver high-quality wideo to any device. Post-video assignments, and incorporate videos into texts, annuareaments, blogs, and discussions, improving engagement, creativity, sense of community, and incorporate videos into texts. 	Passanni
	 Bernynter unersame Logan i
Sealth courses 🖉 🔍	Newgeine • Newgeine

Gambar 3. 9 Uji coba akses server LMS 1 menggunakan IP public

Langkah berikutnya untuk mengaktifkan *load balancing* yaitu dengan cara pilih menu *feature visibility* yang ada di *system fortigate firewall* dan aktifkan *load balance*. Berikut dibawah Gambar 3. 10 untuk mengaktifkan *fitur load balance*.

B FGTAWST5H75F5_08	5	≣ 0,					
Dashboard	>	Feature Visibility					
Network	2	Core Features		Security Features		Additional Features	
Policy & Objects Security Profiles	;	C Advanced Routing	٥	C Application Control	D	Allow Unnamed Policies	0
므 VPN	•	O IPv6	0	D Email Filter	0	Certificates	0
User & Authentication	2	C VPN	0	C Endpoint Control	0	DNS Database	0
System	1 ~	C Switch Controller	0	C AntiVirus	0	C DoS Policy	0
Administrators		WIFI Controller	0	C DNS Filter	0	Advanced Endpoint Control	0
Admin Profiles Fabric Management				C Explicit Proxy	0	Advanced Wireless Features	0
Settings				C File Filter	0	Application Detection-Based SD-	0
HA SNMP				C Intrusion Prevention	D	Con Email Collection	-
Replacement Messages				C Video Filter	0		-
FortiGuard	0			Web Application Firewall	0	C ICAP	
Feature Visibility	- 12			Web Filter	0	C Implicit Firewall Policies	
Security Fabric	,			Taro Trust Natural Access	-	E Load Batance	0
ビ Log & Report	>			2ero nos network Access		Decal In Policy	0
						D Local Out Routing	0
						D Local Reports	0
						Multicast Policy	0
						Multiple Interface Policies	0
						Operational Technology (OT)	0
						Policy Advanced Options	0
						Policy Disclaimer	n
FEIRTIDET	v7.2.3					Apply	

Gambar 3. 10 Mengaktifkan fitur *load balance*

Setelah mengaktifkan *load balance*, selanjutnya membuat *virtual server* untuk moodle-http dan moodle-https. Pada fitur *virtual server* berfungsi untuk mengatur load balancing dengan *service* berupa http dan https pada *server* LMS dan juga menentukan algoritma yang akan digunakan yaitu algoritma *least connection*. Berikut dibawah Gambar 3. 11 membuat *virtual server* untuk *moodle-https*.

+CreaterNew /	int - Michael	B Deven Search		Q		
Name 9	Type 0	Virtual Server #P 2	Load Balancing Method #	Hearth Check #	Reaf Servera #	Interface 2
🖸 IPu4 Virtual Server	0					
🛛 moatle-http Ö	HELTS.	17231422890	Loat Stolan		編 172.31.100.123 編 172.31.100.208 編 172.31.100.208 編 172.31.100.251	im porti
😨 moodfe-fictun	HTTPS	172.31.6.228.443	Least Session		# 172.31.100.123 # 172.31.100.208 # 172.31.100.251	m pott

Gambar 3. 11 Virtual server untuk moodle-http dan moodle-https

Lalu, membuat moodle-load balance pada *firewall policy* yang bertujuan untuk mengelola server LMS agar dapat diakses melalui *http* atau *https* dan sekaligus mengaktifkan *load balancing*. Berikut dibawah Gambar 3. 12 untuk membuat *moodle-load balance*.

≡ Q							
+Create New / Edit	B Delete	Policy Lookup Search			Q		
Name	Source	Destination	Schedule	Service	Action	NAT	Security Profiles
🖃 📓 port1 📓 port2 📀			No.		1		**
ACCESS-TO-MOODLE	🔲 ali	SSH-MOODLE	lo always	O ALL	ACCEPT	O Disabled	no-inspection
moodle-load balance	moodle-load balance		Co always	@ ALL	✓ ACCEPT	O Disabled	no-inspection
🖃 🖩 port2 🎬 port1 🚯		_					
ACCESS-INET	📮 all	🖀 all	Co always	I ALL	✓ ACCEPT	C Enabled	no-inspection
🕀 Implicit 🜖							

Gambar 3. 12 Moodle-load balance

Setelah itu, untuk mengetahui *server* LMS yang sedang aktif dapat diketahui dengan *klik* pada *menu dasboard fortigate firewall* dan klik *load balance monitor*. Berikut dibawah Gambar 3. 13 untuk konfigurasi *load balancing* dengan *fortigate firewall*.



Gambar 3. 13 Konfigurasi load balancing dengan Fortigate Firewall

Sistem *load balancing* perlu diuji akses untuk melihat apakah sistem berhasil. Uji akses dilakukan dengan memakai alamat IP *public* dari *fortigate firewall* pada *browser*. Seperti pada Gambar 3.14, ketika mengakses LMS menggunakan alamat IP *public* pada *browser*.



Gambar 3. 14 Tampilan akses LMS

Selanjutnya agar dalam mengakses *server* LMS lebih mudah yang awalnya menggunakan IP *public* dapat dirubah *domain*nya menjadi lms-moodle.my.id. Berikut gambar 3.15 tampilan LMS yang sudah dirubah *domain*nya.

The second secon	
LMS-moodle biglith.int *	You are not logged in due and
toodle	
Welcome to LMS-moodle, my.id The UK5-Mode allows you to take teaching and tearing to a new level, with interactive viteo tools for teachers and statems.	Login Usernanie
Easily create, lipitival, clip, manage, publish and relover regin-guality values to any device. Post video aniignments, and incorporate videos into tests, announcements, blogs, and discussions, improving orgagement, creativity, write of sommarily, well loarning results.	Fatimore
	Remember Members Ing in
Einricht nitartant	fan gaawafi
	C = Home

Gambar 3. 15 Tampilan akses LMS setelah berubah domain

3.4.3 Pengujian Performansi LMS

Pengujian performansi dengan parameter *Quality of Service* (QoS) *throughput, delay, packet loss, jitter* dan parameter luaran yaitu CPU *usage* bertujuan untuk melihat kualitas jaringan pada saat permintaan dikirimkan ke *server* LMS. Pengujian dilakukan sebanyak 30 kali, yang selanjutnya akan diambil hasil rata-rata pada setiap parameter. Pengujian pemberian beban dilakukan dengan bantuan *software apache benchmark* 1000, 3000, 5000, 7000 dan 10000 dengan 10 *request rate* per detik yang kemudian hasil datanya di*capture* pada *wireshark* yang ada pada pc *client*. Jumlah permintaan, *request* per detik dan banyaknya pengujian yang akan dilakukan dilampirkan pada Tabel 3.4.

No	Jumlah permintaan	Request per detik	Banyaknya Pengujian
1	1000	10	30
2	3000	10	30
3	5000	10	30
4	7000	10	30
5	10000	10	30

Tabel 3. 4 Skenario jumlah beban *request* dan banyaknya pengujian

3.4.4 Proses Pengujian

Pengujian LMS dilakukan dengan mengirimkan *request* menggunakan *apache benchmark*. Pengujian dilakukan dengan menngunakan *script* "bpy" yang bertujuan untuk menjalankan *apache bencmark* sebagai *traffic generator dan* untuk menjalankan *wireshark* sebagai *packet capture* secara bersamaan. Pada perintah "ab" digunakan untuk menjalankan uji kinerja atau pengujian beban pada *server* LMS, sedangkan perintah "-n" digunakan untuk mengatur jumlah total permintaan yang akan dikirimkan selama pengujian dan perintah "-c" untuk mengatur jumlah koneksi *concurent* yang akan dibuat secara bersamaan. Pada perintah "tshark" digunakan untuk menganalisis jaringan yang sedang berjalan pada *wireshrak*, " -i any" menginstruksikan *tshark* untuk menangkap lalu lintas dari semua antarmuka jaringan yang ada, dan "-w filename" mengarahkan *tshark* untuk menyimpan data lalu lintas yang ditangkap ke dalam *file* dengan nama yang diberikan (filename).

```
"p1 = subprocess.Popen("ab -n 1000 -c 10 https://lms-moodle.my.id/",
shell=True)"
"p2 = os.system("tshark -i any -w " + filename)"
```

Penggunaan *wireshark* untuk mendapatkan hasil pengujian berupa parameter *throughput, latency, jitter*, dan *packet loss*. Diagram blok alur pengujian akan digambarkan pada Gambar 3.16.



Gambar 3. 16 Diagram blok alur pengujian

Selanjutnya melakukan pengujian dengan *Apache Benchmark* yang akan mengirimkan 1000, 3000, 5000, 7000, dan 10000 permintaan dengan 10 *request rate* per detik. Sebagai contoh pengujian menggunakan *Apache Benchmark* dengan jumlah 1000 permintaan akan ditampilkan pada Gambar 3.17.

```
×
 ( root@LAPTOP-QV1U26UM: /t ×
                               +
                                   -
12258 Completed 500 requests
15082 Completed 600 requests
17933 Completed 700 requests
20220 Completed 800 requests
22914 Completed 900 requests
25527 Completed 1000 requests
Finished 1000 requests
Server Software:
                              Apache
Server Hostname:
                              lms-moodle.my.id
Server Port:
                              443
                              LLSv1.2,ECDHE-RSA-AES256-GCM-SHA384,2048,256
ECDH P-521 521 bits
SSL/TLS Protocol:
Server Temp Key:
TLS Server Name:
                              lms-moodle.my.id
Document Path:
                              25594 bytes
Document Length:
Concurrency Level:
                              10
Time taken for tests:
                              24.457 seconds
Complete requests:
                              1000
Failed requests:
                              θ
Total transferred:
                              26196000 bytes
                              25594000 bytes
40.89 [#/sec] (mean)
244.573 [ms] (mean)
24.457 [ms] (mean, across all concurrent requests)
1045.99 [Kbytes/sec] received
HTML transferred:
Requests per second:
Time per request:
Time per request:
Transfer rate:
Connection Times (ms)
                       mean[+/-sd] median
                                                 max
1230
                 min
                       126 108.8
Connect:
                  66
                                        105
                      117 176.6
81 163.6
Processing:
                  68
                                         98
                                                 5083
Waiting:
                                         68
                                                 5063
                  47
                      242 209.3
                                        206
                                                 5178
Total:
                 142
Percentage of the requests served within a certain time (ms)
50% 206
66% 226
75% 242
80% 253
90% 296
  95%
           358
  98%
           820
  99%
          1167
 100%
          5178 (longest request)
26321
```

Gambar 3. 17 Contoh pengujian apache benchmark 1000 permintaan

Selanjutnya menggunakan script "qospy" yang di dalamnya memuat perintah untuk menghasilkan nilai QoS yang di-*capture* menggunakan *wireshark* pada *server* LMS. Proses ini dilakukan untuk melihat pembagian beban ke setiap *server* ketika *request* dikirimkan. Contoh hasil pengujian nilai QoS pada saat 1000 permintaan akan ditampilkan pada Gambar 3.18.



Gambar 3. 18 Contoh hasil QoS pada 1000 permintaan

Selanjutnya dengan melakukan pemantauan di setiap *server* dengan menggunakan perintah "sar –u 2". Pemantauan ini dilakukan untuk melihat pembagian beban ke setiap *server* ketika permintaan dikirimkan. Contoh pemantauan CPU pada saat 1000 permintaan akan ditampilkan pada Gambar 3.19.

08:31:00 AM	CPU	Suser	Snice	%system	%iowait	%steal	%idle	
08:31:02 AM	s 11	2 98		1.68	8.31	8 93	94-110	
08:31:04 AM	all	21.95		18 77	1.18	23.78	39.32	
08:31:06 AM	811			16 97	1.70	28 37		
08:31:08 AM	all_	29.79		16 59		23-77	28.15	
08:31:10 AM	a11	28.91		17.83	1.91	25.90	26.25	
08:31:12 AM	all	28 36		15.98	2.15	26 91	26,67	
08:31:14 AM	all	31.02		16.70	2.87	22.97	27.24	
08:31:16 AM ^C	a11			2.85		2.91	89.71	
Average:	all	22.49	0,00	12.71	1.43	19.06	44.31	

Gambar 3. 19 Contoh pemantauan CPU pada saat 1000 permintaan