

BAB II

DASAR TEORI

2.1 KAJIAN PUSTAKA

Pada Tabel 2.1 menunjukkan keterkaitan antara penelitian sebelumnya dengan penelitian penulis. Pada penelitian Sumbayak,dkk [5] mengenai algoritma *weighted least connection* pada jaringan SDN(*Software Defined Network*) yang diujikan pada server heterogen. Penelitian mengungkapkan bahwa algoritma *weighted least connection* membagi beban server berdasarkan jumlah koneksi dan nilai *weight* yang dimiliki oleh setiap server. Hasil dari penelitian menunjukkan kinerja nilai *response time* dan *CPU utilization weighted least connection* lebih baik dibandingkan dengan algoritma *least connection*. Hal ini menunjukkan algoritma *load balance* yang tidak memedulikan kondisi dan kapasitas server tidak cocok untuk server dengan spesifikasi yang beragam.

Penelitian Oktariyadi, dkk [6] mengenai analisis *load* balancing pada algoritma *weighted round robin* dan *round robin*. Penelitian menggunakan *weight* 1:2 dengan parameter pengujian *response time*. Pengukuran penelitian dilakukan pada sisi server untuk mengetahui nilai *response time* pada masing-masing server menggunakan *software* apache benchmark.

Pada penelitian Arifwidodo, dkk [7] yang membahas mengenai algoritma *Weighted Round Robin* pada Proxmox VE. Pengujian dilakukan menggunakan 2 server dengan skenario *weighted lound robin load sharing*, yaitu WRR 2:1, WRR 3:1, WRR 4:1, WRR 5:1. Parameter yang diujikan berupa *Throughput*, *delay*, *jitter*, *packet loss* dan *CPU Utilization* menggunakan *software* apache.

Pada penelitian Poteete.[8], membahas mengenai *cluster* penyimpanan berkelanjutan yang didistribusikan secara seimbang. Dalam penelitiannya perangkat *FortiGate* digunakan sebagai perangkat *load balance* dengan memakai algoritma *round robin*. Parameter yang diuji berupa *high-avaibility* dan rata-rata mbps.

Tabel 2. 1 keterkaitan antara penelitian sebelumnya dan penelitian penulis

Penelitian Oleh	<i>Load Balance</i>	Algoritma		Perangkat <i>Load Balance</i>		<i>Web server</i>	Parameter Pengujian				
		<i>Weighted</i>	Non- <i>Weighted</i>	<i>FortiGate</i>	Sever/perangkat lain		<i>Throughput</i>	<i>Delay</i>	<i>Jitter</i>	<i>Packet loss</i>	Cpu <i>Utilization</i>
Sumbayak, dkk	✓	✓	✓		✓	✓					✓
Oktariyadi, dkk	✓	✓	✓		✓	✓					
Arifwidodo, dkk	✓	✓			✓	✓	✓	✓	✓	✓	✓
Poteete	✓		✓	✓							
Cornelius Aji	✓	✓		✓		✓	✓	✓		✓	✓

2.2 Jaringan Komputer

Jaringan komputer merupakan hasil gabungan antara teknologi komputer dan teknologi komunikasi yang memungkinkan perangkat komputer untuk saling terhubung satu sama lain dan bertukar data melalui media komunikasi yang berbeda, seperti kabel, gelombang radio, atau media nirkabel lainnya.[9]. Jika dua perangkat komputer dapat bertukar informasi melalui jaringan komputer yang menghubungkan keduanya, maka kedua perangkat tersebut dapat dikatakan saling terkoneksi..

Keuntungan atau manfaat dalam membangun jaringan komputer, yaitu:[10]

a. Berbagi Sumber (*Sharing Resource*)

Setiap orang di jaringan komputer yang terhubung harus dapat menggunakan sumber daya kapan saja dan dari mana saja.

b. Media Komunikasi

Menggunakan jaringan komputer sebagai saluran komunikasi, pengguna dapat mengadakan telekonferensi atau mengirim pesan (baik teks maupun suara) satu sama lain untuk berbagi informasi.

c. Integrasi Data

Jaringan komputer mengurangi ketergantungan pada pusat komputer karena setiap prosedur pertukaran data dapat didistribusikan melalui beberapa perangkat daripada hanya bergantung pada satu komputer.

2.2.1 Perangkat Jaringan

Untuk menghubungkan dua komputer atau lebih pada jaringan komputer dibutuhkan sebuah perangkat keras jaringan komputer agar dapat saling terhubung. Beberapa jenis perangkat jaringan di antaranya:[11]

1. *Network Interface Cards* (NIC) atau Kartu Jaringan.

Kartu Jaringan (NIC) berperan penting dalam menghubungkan komputer ke media transmisi. Biasanya, kartu jaringan merupakan perangkat internal yang dipasang pada slot ekspansi di dalam komputer untuk memudahkan proses koneksi. Dengan menggunakan kartu jaringan, komputer dapat terhubung dengan jaringan komputer dan memungkinkan pertukaran data dan informasi dengan perangkat lainnya dalam jaringan.

2. Media (kabel, gelombang radio)

Empat jenis kabel jaringan yang umum digunakan saat ini yaitu :

a. Kabel *Coaxial*

Kabel tersebut terdiri dari dua kabel yang dilapisi oleh dua lapisan isolasi. Lapisan isolasi pertama terdiri dari serabut konduktor, sedangkan lapisan isolasi kedua dilindungi oleh bahan plastik yang berfungsi sebagai pelindung terakhir untuk mencegah goresan pada kabel.

b. Kabel *Unshielded Twisted Pair* (UTP)

Untuk memberikan perlindungan terhadap gangguan listrik, kabel *twisted pair* terbuat dari dua kabel yang dijalin bersama dan dipilin enam kali per inci. Hal ini dilakukan dengan tujuan untuk menjaga impedansi atau tahanan listrik agar konsisten, sehingga dapat mengurangi kemungkinan terjadinya gangguan dan kerusakan pada sinyal data yang ditransmisikan melalui kabel tersebut.

c. Kabel *Shielded Twisted Pair* (STP)

Mirip dengan kabel UTP, kabel STP memiliki kabel yang lebih besar dan lapisan pelindung isolasi tambahan di atasnya. Hal ini bertujuan untuk mencegah interferensi atau gangguan yang dapat terjadi pada kabel, sehingga kualitas sinyal data yang ditransmisikan melalui kabel tersebut dapat tetap terjaga.

d. Kabel Serat Optik (*Fiber Optik*)

Data ditransmisikan menggunakan kabel serat optik menggunakan pulsa cahaya sebagai media pembawa. Kabel serat optik memiliki kecepatan transmisi data yang paling tinggi dan lebih dapat diandalkan, karena interferensi listrik yang dapat menyebabkan kehilangan data jarang terjadi pada jenis kabel ini.

3. Hub/Konsentrator



Gambar 2. 1 Hub

Hub adalah perangkat yang mengumpulkan dan menghubungkan kabel jaringan dari semua perangkat jaringan. Perangkat Hub dapat dilihat pada Gambar 2.1. Dengan adanya hub, setiap perangkat dapat terhubung satu sama lain dan berkomunikasi dalam jaringan. Namun, hub mengirimkan paket data ke semua perangkat yang terhubung ke dalamnya (*broadcast*), sehingga dapat menimbulkan masalah jika terjadi *collision* atau interferensi pada jaringan [11].

4. Switch



Gambar 2. 2 Switch

Switch adalah perangkat yang berfungsi untuk menghubungkan beberapa jaringan dan meneruskan paket data berdasarkan alamat MAC. Perangkat switch dapat dilihat pada Gambar 2.2. Berbeda dengan bridge, switch memiliki beberapa port dan memiliki *collision* domain serta broadcast domain yang terpisah. Selain itu, switch juga dapat mengatur lalu lintas paket data yang melalui jaringannya [11].

5. Repeaters



Gambar 2. 3 Repeater

Contoh perangkat repeater dapat dilihat pada Gambar 2.3. Pendekatan paling sederhana adalah dengan menggunakan *kabel unshielded twisted pair* dengan panjang maksimal 100 meter untuk meningkatkan sinyal pada LAN. Repeater kemudian dapat digunakan di jaringan tersebut [11].

6. Bridge



Gambar 2. 4 Bridge

Bridge adalah sebuah perangkat jaringan yang berfungsi membagi satu jaringan menjadi dua jaringan untuk mencapai efisiensi jaringan yang lebih baik. Perangkat bridge dapat dilihat pada Gambar 2.4. Perangkat ini sering digunakan ketika pertumbuhan jaringan terjadi dengan cepat. Bridge memungkinkan pengaturan jaringan menjadi lebih optimal dan teratur [11].

7. Router

Router adalah jenis peralatan jaringan yang dapat mentransfer data antar jaringan. Perangkat router dapat dilihat pada gambar 2.5. Meskipun mirip dengan Bridge, Router lebih pintar karena dapat memilih rute optimal untuk mentransfer pesan dengan memperhitungkan alamat sumber dan tujuan [11].



Gambar 2. 5 Router

8. Perangkat Firewall

Firewall berfungsi sebagai sebuah penghalang yang memisahkan jaringan lokal dan jaringan publik agar setiap data yang masuk bisa diidentifikasi dan disaring untuk mencegah bahaya dan ancaman yang datang dari jaringan publik. Dengan adanya firewall, keamanan jaringan lokal bisa ditingkatkan dan mengurangi risiko serangan dari luar.[12]

9. Printer Dan Peripheral Lain

Salah satu alasan utama untuk menggunakan jaringan adalah untuk memudahkan penggunaan printer. Dengan menggunakan satu printer bersama-sama pada jaringan, pengguna tidak perlu memiliki printer sendiri dan dapat lebih ekonomis. Printer dapat dihubungkan langsung ke workstation atau server dan dapat digunakan secara bersama-sama pada jaringan.

2.2.2 Topologi Jaringan

Topologi menggambarkan protokol atau prosedur untuk menghubungkan komputer atau *node* secara fisik serta susunan hubungan antara bagian-bagian yang berinteraksi melalui perangkat keras atau media jaringan. Tiga topologi, terutama Topologi *Bus*, *Star*, dan *Ring*, sering digunakan dalam jaringan area lokal dasar. Struktur jaringan berkembang dari waktu ke waktu dan menjadi lebih akrab dengan topologi pohon (*tree*), *extended star*, dan *mesh* (tak beraturan)[13].

2.2.2.1 Topologi Jaringan Bus

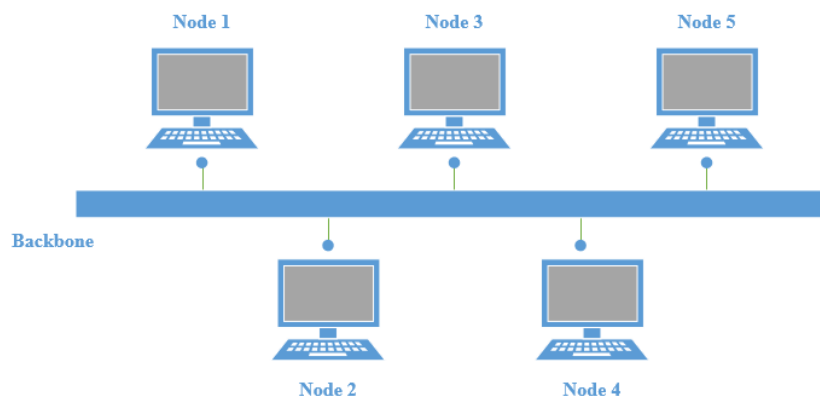
Dalam topologi jaringan *bus*, satu kabel digunakan untuk menghubungkan semua perangkat di internet. Kabel ini sering disebut sebagai jaringan *backbone*. Ketika komunikasi terjadi antar *node*, perangkat yang mengirimkan pesan

menyiarkan ke semua *node* di jaringan, tetapi hanya penerima yang diinginkan yang mencerna pesan tersebut. Topologi jaringan bus dapat dilihat pada Gambar 2.6. Keuntungan dari Topologi Fisik jenis ini antara lain:[14]

- a. kemudahan instalasi dan minimalkan kabel yang dibutuhkan.
- b. kegagalan sebuah *node* yang terhubung ke jaringan tidak berpengaruh pada *node* lain yang terhubung ke jaringan.
- c. Juga pesan dari satu *node* dapat dilihat secara bersamaan oleh semua *node* lain di jaringan.

Kerugian dari topologi antara lain :

- a. Batas kinerja pada jumlah *node* jaringan,
- b. Semua jaringan akan terputus jika kabel *backbone* gagal.



Gambar 2. 6 Topologi *BUS*

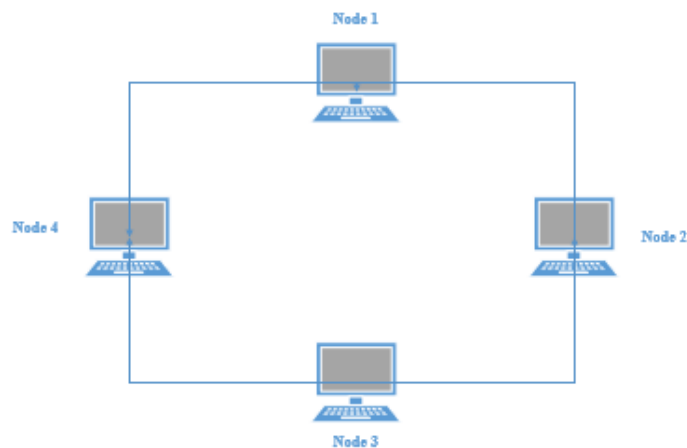
2.2.2.2 Topologi Ring

Topologi jaringan *ring* adalah jenis topologi jaringan di mana setiap *node* terhubung ke dua *node* lainnya dalam jaringan, membentuk cincin dengan *node* pertama dan terakhir terhubung. Pesan dalam jaringan ini mengalir dari *node* pengirim ke *node* penerima melalui *node* perantara, yang berfungsi sebagai pengulang aktif. Beberapa jenis topologi jaringan *ring* memiliki arah aliran pesan yang sama, sementara yang lain memungkinkan arah aliran pesan ke kedua arah dengan bantuan dua kabel. Dalam beberapa kasus, perangkat pemblokiran diperlukan untuk mencegah penyerbuan paket yang dapat mengganggu koneksi jaringan. Topologi jaringan *ring* dapat dilihat pada Gambar 2.7.

Kelemahan dari topologi jaringan *ring* meliputi:[14]

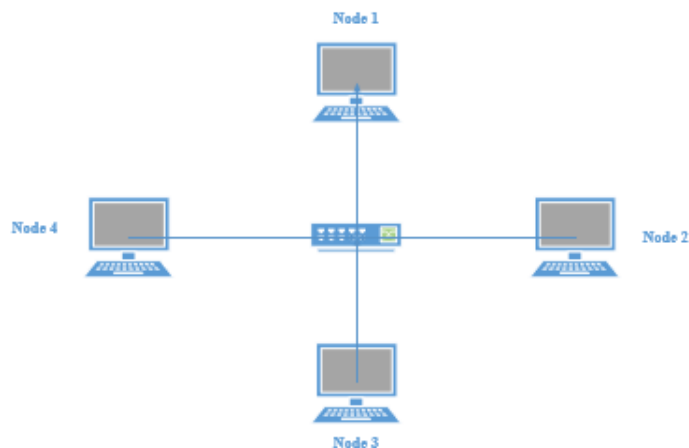
- a. waktu transmisi yang relatif lama antara *node* dalam cincin
- b. serta dampak yang lebih luas pada komunikasi jaringan secara keseluruhan ketika terjadi kegagalan pemasangan kabel antara dua *node*

Namun, topologi jaringan *ring* memiliki keunggulan dalam hal keandalan yang cukup besar, karena sifat ganda dari pemasangan kabel antar *node* memungkinkan lalu lintas dialihkan ke jalur alternatif untuk memperbaiki gangguan koneksi antara dua *node* dalam jaringan.



Gambar 2. 7 Topologi *Ring*

2.2.2.3 Topologi Jaringan Star



Gambar 2. 8 Topologi *Star*

Topologi *star* atau Bintang adalah suatu jenis topologi jaringan yang memiliki tampilan seperti sebuah bintang, di mana terdapat sebuah alat concentrator (hub, switch) sebagai pusat yang terhubung ke setiap *node* atau komputer di

sekitarnya. Dalam topologi ini, semua komputer terhubung secara langsung ke alat concentrator di tengah jaringan. Topologi jaringan start dapat dilihat pada Gambar 2.8. Topologi *star* memiliki beberapa keunggulan, antara lain:

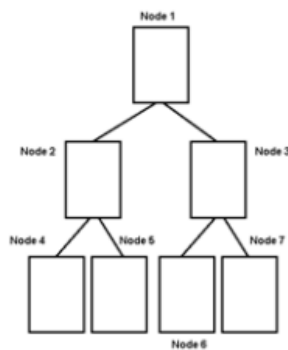
- a. kontrol akses terpusat
- b. kemudahan dalam menambahkan komputer atau terminal
- c. kemudahan dalam mendeteksi kesalahan dan kerusakan pada jaringan
- d. tahan terhadap arus lalu lintas jaringan yang sibuk dan tingkat keamanan yang cukup tinggi.

Namun, topologi *star* juga memiliki beberapa kekurangan, di antaranya:

- a. jika *node* tengah atau pusat mengalami gangguan atau kerusakan, maka seluruh rangkaian jaringan akan berhenti.
- b. pemakaian kabel jaringan juga sangat banyak
- c. jaringan sangat tergantung pada terminal pusat
- d. biaya pengadaan jaringan cenderung lebih mahal dibandingkan dengan topologi *bus* atau *ring*[15].

2.2.2.4 Topologi Jaringan Pohon

Topologi jaringan pohon dibangun dengan membuat kumpulan Topologi jaringan bintang yang bersubordinasi pada sebuah *node* pusat, atau dengan menghubungkan beberapa topologi jaringan bintang langsung melalui sebuah *bus*, sehingga mendistribusikan fungsionalitas *node* pusat di antara beberapa *node* level atas topologi jaringan bintang. Dalam topologi jaringan pohon sederhana, tidak ada *node* bawahan topologi jaringan bintang yang terhubung ke *bus*. Pesan dalam topologi jaringan pohon dapat disiarkan dari *node* pusat ke semua Topologi Jaringan Bintang yang saling terhubung, atau ditujukan untuk memilih Topologi Jaringan Bintang. Topologi jaringan pohon dapat dilihat pada Gambar 2.9.



Gambar 2. 9 Topologi pohon

keunggulan utama dari topologi jaringan pohon adalah mudahnya jaringan diperluas. Pembaruan bisa semudah menghubungkan topologi jaringan bintang tambahan ke *bus*. Selain itu, seperti topologi jaringan bintang, konfigurasi ini memiliki kelebihan bahwa kegagalan kabel terlokalisasi. Namun, jika *node* level atas topologi jaringan bintang gagal atau kabel ke *node* tersebut gagal, seluruh bagian jaringan akan kehilangan komunikasi, tidak seperti pada topologi jaringan bintang murni yang hanya kehilangan satu *node* bawahan saja[14].

2.2.3 Protokol Jaringan

Protokol jaringan komputer adalah seperangkat pedoman yang berlaku dalam jaringan komputer dan pihak pengirim dan penerima harus mengikuti agar mereka dapat bertukar informasi dan berinteraksi saat menggunakan sistem operasi yang berbeda[16]. Pada tahun 1980-an, Organisasi Standar Internasional (ISO) memperkenalkan suatu model referensi standar yang menggambarkan cara kerja sejumlah protokol SNA. Model referensi ini dikenal sebagai *Open System Interconnection* (OSI).

Model OSI memiliki tujuh lapisan. Proses untuk komunikasi data secara eksplisit berada di bawah kendali setiap lapisan. Melalui seperangkat protokol dan standar, setiap lapisan harus dapat berkomunikasi langsung dengan lapisan di atas dan di bawahnya.[11]. Urutan Lapisan OSI dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Lapisan OSI

Lapisan Ke-	Nama Lapisan	Keterangan
7	<i>Application Layer</i>	Lapisan ini mengontrol bagaimana program dapat mengakses jaringan, membuat pesan kesalahan, dan berfungsi sebagai antarmuka antara aplikasi dan fungsi jaringan. Pada lapisan ini, terdapat protokol seperti HTTP, FTP, SMTP, dan NFS.
6	<i>Presentation Layer</i>	Tujuan dari lapisan ini adalah untuk mengubah data yang ingin dikirimkan oleh aplikasi ke dalam format yang dapat ditransmisikan melalui jaringan. Protokol pada layer ini seperti beralih perangkat lunak, seperti layanan Workstation (di Windows NT), dan Network Shell (seperti <i>Virtual Network Computing</i> (VNC) atau <i>Remote Desktop Protocol</i> (RDP)), adalah beberapa protokol yang ada di tier ini.
5	<i>Sesion Layer</i>	Lapisan ini bertugas mencari tahu bagaimana koneksi dapat dibuat, dipertahankan, atau diakhiri. Selain itu, resolusi nama dilakukan oleh lapisan ini..
4	<i>Transport Layer</i>	Agar dapat dipasang kembali di sisi tujuan setelah diterima, lapisan ini membagi data menjadi paket-paket data dan memberikan nomor seri ke paket-paket tersebut. Selain itu, lapisan ini membuat pengakuan untuk transmisi yang berhasil dan mentransmisikan ulang paket yang hilang dalam rute. TCP (<i>Transmission Control Protocol</i>) dan UDP (<i>User Data Protocol</i>) adalah protokol pada tingkat ini.
3	<i>Network Layer</i>	Lapisan ini menetapkan header untuk paket data, memberikan alamat IP ke perangkat jaringan, dan menggunakan router dan switch layer-3 untuk mengirimkan paket data melalui jaringan.

Lapisan Ke-	Nama Lapisan	Keterangan
2	<i>Data Link</i>	Lapisan ini memutuskan bagaimana mengatur bit data ke dalam bentuk bingkai dan bertugas menjaga konvensi alamat perangkat keras seperti Alamat Kontrol Akses Media (Alamat MAC), kontrol aliran data, dan koreksi kesalahan. Selain itu, spesifikasi IEEE 802, yang mengatur fungsi perangkat seperti bridge, repeater, dan switch layer 2, membagi level ini menjadi dua bagian: <i>Media Access Control</i> (MAC) dan <i>Logical Link Control</i> (LLC).
1	<i>Physical Layer</i>	Media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan (seperti Ethernet atau <i>Token Ring</i>), topologi jaringan, dan pemasangan kabel semuanya ditentukan pada lapisan ini. Level ini juga menentukan bagaimana <i>Network Interface Card</i> (NIC) dapat berkomunikasi dengan radio atau kabel.

2.3 Load Balancing

Load balancing adalah proses pendistribusian lalu lintas jaringan yang masuk dengan efisien melalui sekelompok server atau perangkat jaringan. [17]. Dengan menggunakan load balancing, akses ke sumber daya dalam jaringan dapat didistribusikan secara merata ke beberapa host atau server lainnya sehingga tidak terpusat pada satu titik saja. Hal ini dapat membantu menjaga stabilitas kinerja layanan secara keseluruhan. Secara umum, algoritma load balancing server dapat dibagi menjadi dua jenis, yaitu algoritma berbobot (*weighted*) dan algoritma tanpa bobot (*non-weighted*).[18]

a. *weighted* algoritma

Untuk memilih server mana yang akan memproses permintaan berikut, *weighted* algoritma menggunakan bobot (atau preferensi). Server dengan preferensi yang lebih tinggi akan memiliki lebih banyak lalu lintas daripada

server dengan preferensi yang lebih rendah. Persentase lalu lintas yang dikirim ke setiap server sekitar sama dengan preferensi server tersebut dibagi dengan preferensi total dari semua server dalam kelompok tersebut.

b. *non-weighted* algoritma

Non-weighted algoritma mengasumsikan bahwa kapasitas semua server dalam grup menjadi setara. Meskipun algoritma tanpa bobot biasanya lebih cepat daripada algoritme berbobot, beberapa algoritme tanpa bobot, seperti algoritme hash, dapat mengirimkan lebih banyak lalu lintas ke beberapa server. Jika ada server dengan kapasitas berbeda dalam grup, pemrosesan tidak dapat mengoptimalkan kapasitas semua server.

Manfaat *load* balancing antara lain:[19]

a. *Flexibility*

Mmemberikan fleksibilitas untuk menambah dan menghapus server sesuai permintaan.

b. *Scalability*

Penyeimbangan beban memiliki kemampuan untuk menambahkan server fisik atau virtual untuk mengakomodasi permintaan tanpa menyebabkan gangguan layanan. Saat server baru online, penyeimbang beban mengenalinya dan memasukkannya dengan mulus ke dalam proses

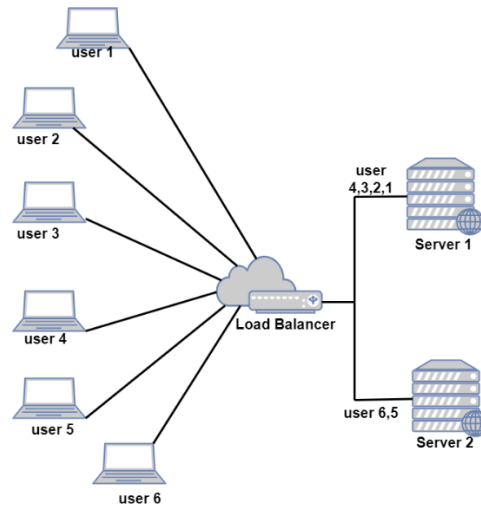
c. *Redundancy*

Dalam mendistribusikan lalu lintas melalui sekelompok server, penyeimbangan beban menyediakan redundansi bawaan.

2.3.1 **Weighted Round Robin**

Algoritma *Weighted Round Robin* mirip dengan algoritma *round robin*, bersifat statis. algoritma *Weighted Round Robin* menggunakan CPU yang sedikit. *Request* akan diarahkan ke masing-masing server yang dibuat berdasarkan bobot yang telah ditentukan oleh administrator. Semakin tinggi bobot, semakin tinggi persentase permintaan pengunjung yang diterima server, seperti pada Gambar 2.10. Misalnya, server1 diberi bobot 4 (user 1,2,3,4) dan server 2 diberi bobot 2 (user 6,5). Penyeimbang beban meneruskan empat permintaan ke server 1 permintaan dikirim ke server 2. Jika beberapa mesin virtual memiliki lebih banyak sumber daya

dalam CPU, Ram, GPU, dan spesifikasi lainnya, maka algoritma *weighted round robin* cocok untuk digunakan. Dengan bantuan *weighted round robin*, admin dapat dengan mudah mendistribusikan lalu lintas web pada server yang memiliki sumber daya tinggi dengan menentukan bobotnya [20].



Gambar 2. 10 *Weighted Round Robin*

2.4 Web Server

Di jaringan komputer, server adalah mesin yang melakukan layanan tertentu. Server menjalankan sistem operasi yang dirancang khusus untuk digunakan di jaringan, sering dikenal sebagai sistem operasi jaringan, dan memiliki spesifikasi perangkat keras yang memadai, termasuk CPU dan RAM yang memadai. Selain itu, server menjalankan perangkat lunak manajemen yang mengontrol akses pengguna ke sumber daya jaringan seperti *file* atau printer dan memberikan akses layanan pengguna *workstation*[21]. Jadi secara umum, server digunakan untuk membangun aplikasi yang mengadopsi arsitektur *client-server* dengan menggunakan sistem operasi yang sudah terpasang.

Komponen terpenting dari *world wide web* (www) adalah server web, yang merupakan bagian dari perangkat lunak komputer. Program ini menunggu permintaan dari pengguna browser. Saat permintaan diterima dari browser, server web memprosesnya dan kemudian mengirimkan kembali data yang diproses ke browser. Dalam prosedur ini, web server berkomunikasi dengan klien melalui protokol HTTP (*hypertext transfer protocol*)[22]. HTTP menggunakan *Transmission Control Protocol* (TCP) untuk mengirimkan data antara klien HTTP dan server HTTP. Ketika klien HTTP ingin mengirim permintaan HTTP, pertama-

tama harus membuat koneksi TCP ke server HTTP[23]. Beberapa web server yang umum digunakan antara lain Apache, Microsoft Windows Server 2008 IIS (*Internet Information Services*), NginX, Zeus, IIS, dan Sun Java System.

2.5 Apache Benchmark

Sebuah program bernama apache benchmark dibuat oleh Apache Foundation untuk mengevaluasi kinerja server web, khususnya server web apache. Dalam melakukan pengujian, apache benchmark mengirimkan permintaan ke server web dengan parameter yang telah ditentukan.. Beberapa parameter penting yang dapat digunakan sebagai acuan dalam pengujian benchmarking adalah jumlah permintaan dan jumlah penggunaan bersamaan (*concurrency*)[24].

2.6 Eve-NG

EVE-NG adalah singkatan dari Emulated Virtual Environment - Next Generation, dan berisi alat untuk membangun jaringan virtual dan menghubungkannya ke perangkat virtual atau nyata lainnya. Itu membuatnya lebih mudah untuk menggunakan, menggunakan kembali, mengelola, menghubungkan, mendistribusikan, dan berbagi topologi, pekerjaan, ide, konsep, dan lab. Pengguna dapat memanfaatkan EVE-NG untuk meminimalkan biaya dan waktu yang diperlukan untuk menyiapkan apa yang mereka butuhkan, atau bahkan menjalankan aktivitas yang sebelumnya dianggap mustahil dengan cara yang sederhana. EVE-NG pada dasarnya adalah platform jaringan virtual yang memungkinkan pengguna untuk mengembangkan dan menguji skenario jaringan yang rumit tanpa memerlukan perangkat keras aktual yang mahal. Ini bermanfaat bagi insinyur jaringan, arsitek, pelatih, dan peneliti yang membutuhkan desain jaringan, konfigurasi, dan alat pengujian[25].

2.7 VMware Workstation

VMware Workstation adalah hypervisor yang dapat digunakan untuk menyebar mesin virtual seperti desktop dan server pada satu mesin fisik[26].

2.8 Wireshark

Wireshark adalah alat analisis paket jaringan. Penganalisis paket jaringan akan berusaha menangkap paket jaringan dan akan melakukan segala upaya untuk membuat data paket mudah dibaca. Jaringan LAN nirkabel adalah salah satu dari banyak jenis media jaringan tempat Wireshark dapat mengumpulkan lalu lintasnya. Jenis media yang didukung bervariasi tergantung pada sejumlah faktor, termasuk sistem operasi yang digunakan[27].

2.9 Ubuntu Server

Ubuntu merupakan distro Linux Debian yang populer, bahkan pada versi desktopnya banyak digunakan oleh banyak orang karena antarmuka pengguna (UI) yang hampir mirip dengan Windows. Untuk penggunaan sebagai server, Ubuntu dapat dijadikan sebagai opsi karena distro ini mudah digunakan, stabil, aman, cepat, serta memiliki perintah yang cukup mudah untuk dipelajari terutama bagi pemula. Banyak penjual VPS juga menyediakan Ubuntu sebagai salah satu pilihan sistem operasi yang dapat dipilih untuk diinstal pada server mereka[28][29]. Minimal spesifikasi untuk instalasi ubuntu server dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Recommended Minimum Requirements

Install Type	Install method	CPU	RAM	Hard Drive	
				Base Sysytem	All Task Installed
Server (Standard)	debian-installer	1 Ghz	512 Mb	1,5 Gb	2,5 Gb
	live-server	1 Ghz	1 Gb	1,5 Gb	n/a
Server (Minimal)	debian-installer	300 Mhz	384 Mb	1,5 Gb	2,5 Gb

2.10 FortiGate

Fortigate adalah salah satu sistem keamanan yang dikeluarkan oleh perusahaan ternama yaitu, fortinet. Perusahaan ini juga menyediakan suatu layanan simulator untuk melakukan simulasi tanpa batas waktu dengan evaluation license (untuk versi 7.2.1 ke atas). Dengan memakai simulator yang sudah ada sebelumnya, dapat

memberikan akses penuh kecuali batas pengaturan firewall yang hanya ditentukan untuk batas tertentu[30].

Firewall *Fortigate* didasarkan pada kernel Linux yang berjalan di FortiOS (eksklusif). Ini memberikan kinerja tinggi untuk hampir semua jenis lalu lintas termasuk lalu lintas terenkripsi, yang menjadi populer baru-baru ini waktu. *Throughput Fortigate* berkisar dari 17Gbps hingga 1Tbps. *Fortigate* menggunakan dua arsitektur komponen yang diaktifkan dengan pemrosesan jalur paralel, untuk mengoptimalkan kinerja *throughput*. Ini membuat paket data sistem menjadi independen. Kedua komponen tersebut adalah Content Processor (CP) dan Network Processor (NP). Seri CP beroperasi di luar bidang arus lalu lintas dan menyediakan fasilitas kriptografi cepat yang membantu memeriksa konten lalu lintas. Seri NP bekerja dengan sistem operasi khusus (FortiOS) untuk memberikan kinerja kelas atas untuk protokol internet yang berbeda seperti IPV4 dan IPV6. *Fortigate* juga menangani lalu lintas multicast dengan latensi paling sedikit (sekitar tiga mikrodetik)[31].

Fortinet *FortiGate* adalah solusi terhadap sebagian besar ancaman eksternal ke organisasi. Firewall *FortiGate* menjalankan fungsi di *Layer 3* (jaringan), 4 (transportasi), dan 7 (aplikasi). Pada *Layer 3*, *FortiGate* beradadi antara dua jaringan yang saling terhubung. Saat data ditransmisikan dari satu jaringan ke jaringan lainnya, *FortiGate* melakukan inspeksi paket mendalam, memeriksa apakah koneksi sedang digunakan untuk mengirim ancaman dunia maya. Pada *Layer 4*, *FortiGate* diposisikan di antara dua perangkat, memeriksa setiap segmen data sebelum mencapai perangkat penerima. Dengan cara ini, *FortiGate* mencegah satu perangkat digunakan untuk menginfeksi perangkat lain. *FortiGate* juga berinteraksi dengan *Layer 7* karena ia memiliki kemampuan untuk memeriksa perilaku aplikasi, termasuk bagaimana ia menggunakan sumber daya datanya. Pada *Layer 7*, *FortiGate* dapat mendeteksi aktivitas mencurigakan di dalam aplikasi dan bertindak sesuai dengan itu[32].

Fortigate mendukung *load balance* server dengan membentuk virtual server. Penyeimbangan beban server FortiOS berisi semua fitur solusi penyeimbangan beban server. Anda dapat menyeimbangkan lalu 22 lintas di

beberapa server backend berdasarkan beberapa metode *load balance* antara lain[33].

a. *Static*

Beban lalu lintas secara statis tersebar merata di semua real server. Sesi tidak ditetapkan sesuai dengan seberapa sibuk masing-masing server sebenarnya. Metode penyeimbangan beban ini memberikan beberapa kegigihan karena semua sesi dari alamat sumber yang sama selalu menuju ke real server yang sama. Karena distribusinya tidak bernegara, jadi jika real server ditambahkan, dihapus, atau naik atau turun, distribusinya berubah dan persistensi mungkin hilang.

b. *Round Robin*

Mengarahkan permintaan baru ke real server berikutnya. Metode ini memperlakukan semua real server sebagai sama terlepas dari waktu respons atau jumlah koneksi.

c. *Weighted*

Pembagian beban dimana real server dengan nilai bobot yang lebih tinggi menerima persentase koneksi yang lebih besar berdasarkan kesehatan dan kinerja server termasuk waktu pulang pergi dan jumlah koneksi.

d. *Least Session*

Mengarahkan permintaan ke real server yang memiliki jumlah koneksi saat ini paling sedikit. Metode ini bekerja paling baik di lingkungan di mana server sebenarnya atau peralatan lain yang Anda load balancing semuanya memiliki kemampuan yang serupa. Metode penyeimbangan beban ini menggunakan tabel sesi *Fortigate* untuk melacak jumlah sesi yang sedang diproses oleh setiap real server. Unit *Fortigate* tidak dapat mendeteksi jumlah sesi yang sebenarnya sedang diproses oleh real server.

e. *Least RTT*

Mengarahkan sesi ke server sebenarnya dengan waktu perjalanan pulang pergi terendah. Waktu perjalanan pulang pergi ditentukan oleh monitor pemeriksaan kesehatan ping. Standarnya adalah 0 jika tidak ada 23 monitor pemeriksaan kesehatan ping yang ditambahkan ke server virtual.

f. *First Alive*

Mengarahkan sesi ke real server yang pertama. Jadwal penyeimbangan beban ini memberikan perlindungan *failover* real server dengan mengirimkan semua sesi ke real server yang pertama. Jika real server gagal, semua sesi dikirim ke real server berikutnya. Sesi tidak didistribusikan ke semua real server sehingga semua sesi diproses oleh real server pertama saja. HTTP Host Memuat keseimbangan koneksi host HTTP di beberapa real server menggunakan header HTTP host untuk memandu koneksi ke real server yang benar.

2.11 Parameter Pengujian

2.9.1 Throughput

Throughput adalah kecepatan data pada jaringan. Hal ini merujuk pada jumlah data yang diangkut oleh lalu lintas jaringan. Pengukuran *throughput* dilakukan dengan menghitung jumlah *byte* yang dalam jangka waktu tertentu. Lamanya jangka waktu pengukuran dapat mempengaruhi hasil pengamatan perilaku jaringan. Jika jangka waktunya terlalu panjang, maka gambaran perilaku jaringan dapat menjadi kabur, sedangkan jika jangka waktunya terlalu pendek, maka data yang dihasilkan terlalu banyak dan dapat menyebabkan perubahan pada gambaran perilaku jaringan yang sebenarnya.[34]. Kategori kualitas *throughput* dapat dilihat pada Tabel 2.4 dan persamaan *throughput* dapat dilihat pada persamaan 2.1 [35].

$$Throughput = \frac{\text{jumlah data yang dikirim}}{\text{waktu pengiriman data}} \quad (2.1)$$

Tabel 2. 4 Kualitas *Throughput*

Kategori	<i>Throughput</i>	Indeks
Sangat memuaskan	>2,1Mbps	4
memuaskan	1200kbps – 2,1 Mbps	3
Kurang memuaskan	700 kbps – 1200 kbps	2
Tidak memuaskan	< 700 kbps	1

2.9.2 Delay

Delay merujuk pada waktu yang dibutuhkan oleh data atau informasi untuk mencapai tujuannya dari sumbernya. Faktor-faktor yang memengaruhi *delay*

meliputi jarak, media fisik yang digunakan, kepadatan jaringan, dan waktu yang dibutuhkan untuk memproses data. Oleh karena itu, *delay* dapat bervariasi tergantung pada situasi dan kondisi yang terkait [36]. Kategori *delay* dapat dilihat pada Tabel 2.5 dan persamaan *delay* dapat dilihat pada persamaan 2.2 [35].

$$Delay\ rata - rata = \frac{total\ delay}{total\ packet\ yang\ diterima} \quad (2.2)$$

Tabel 2. 5 Kategori *delay*

Kategori	<i>Delay</i>	Indeks
Sangat memuaskan	<150 ms	4
memuaskan	150 s/d 300 ms	3
kurang memuaskan	300 s/d 450 ms	2
tidak memuaskan	> 450 ms	1

2.9.3 Packet loss

parameter yang digunakan untuk menggambarkan situasi jaringan di mana paket data hilang akibat tabrakan atau kemacetan [36]. Kategori kualitas *packet loss* dapat dilihat pada Tabel 2.6 dan persamaan *packet loss* dapat dilihat pada persamaan 2.3 [35].

$$packet\ loss = \frac{packet\ data\ yang\ dikirim - packet\ data\ yang\ diterima}{packet\ data\ yang\ dikirim} \times 100\ \% \quad (2.3)$$

Tabel 2. 6 Level kategori paket *loss*

Kategori	Paket <i>loss</i> (%)
Sangat memuaskan	Zero
memuaskan	3
Kurang memuaskan	15
Tidak memuaskan	25

2.9.4 CPU Utilization

CPU Utilization adalah persentase konsumsi CPU saat menangani beban tertentu. Ketika jumlah permintaan pengguna meningkat, penggunaan CPU juga meningkat[37].