Journal of Information System Research (JOSH)
Volume 5, No. 1, Oktober 2023, pp 252−260
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i1.4386

Implementasi Metode Load Balancing Untuk Optimalisasi Performa Server Pada Jaringan Internet

Hamid Wijaya[1,*], Iim Abdurrohim[2], Jentot Tugiyono[2], Rhaishudin Jafar Rumandan[3]

[1]Program Studi Ilmu Komputer, Universitas Sembilanbelas November Kolaka, Kolaka
Jl. Pemuda No.339, Tahoa, Kec. Kolaka, Kabupaten Kolaka, Sulawesi Tenggara, Indonesia
[2]Program Studi Teknik Informatika, Universitas Kebangsaan Republik Indonesia, Bandung
Jl. Terusan Halimun No.37, Lkr. Sel., Kec. Lengkong, Kota Bandung, Jawa Barat, Indonesia
[3]Program Studi Manajemen Pendidikan Islam, Institut Agama Islam Negeri Ambon, Ambon
Jl. Dr. H. Tarmizi Taher, Jalan Kebun Cengkeh, Batu Merah, Kec. Sirimau, Kota Ambon, Maluku
Email: [1,*]hamidwijaya35@gmail.com, [2]iimabdurrohim@ukri.ac.id, [3]jentot.uk@gmail.com,[4]jafarrumadan@gmail.com
Email Penulis Korespondensi: hamidwijaya35@gmail.com
Submitted: **09/10/2023;** Accepted: **28/10/2023;** Published: **29/10/2023**

Abstrak–Semakin meningkatnya jumlah pengguna internet dan masing-masing pengguna memiliki beragam kebutuhan berakibat pada lalu lintas internet yang sangat tinggi dan berfluktuasi yang berdampak pada peningkatan beban kerja pada server serta infrastruktur jaringan. Banyaknya beban kerja yang dialami oleh server maka diperlukan pendekatan yang dapat mendistribusikan beban kerja secara merata di antara beberapa sumber daya atau yang dikenal dengan teknik load balancing. Sehingga, tujuan utama dari penelitian ini yaitu untuk mengoptimalkan performa jaringan internet dengan mengimplementasikan teknik load balancing dan mengukur peningkatan kinerja yang dapat dicapai. Metode load balancing digunakan untuk menghindari ketidakseimbangan yang dapat menyebabkan satu sumber daya terlalu terbebani sementara yang lain mungkin tidak digunakan secara optimal. Untuk pengujian dilakukan koneksi sebanyak 1000 kali dengan 100 permintaan/detik, kemudian ditingkatkan bertahap sampai dengan koneksi sebanyak 11000 kali dengan 1100 permintaan/detik. Hasil pengujian menunjukkan bahwa pada tingkat koneksi rendah hingga sedang (1000/100 hingga 6000/600), server tanpa load balancing cenderung memberikan waktu respon yang lebih baik. Namun, saat tingkat koneksi meningkat (7000/700 hingga 11000/1100), server dengan load balancing menunjukkan keunggulannya dengan memiliki waktu respon yang lebih baik dibandingkan server tanpa teknik tersebut. Oleh karena itu, penelitian ini mengemukakan bahwa penerapan strategi load balancing sangat penting dalam menjaga performa optimal dan stabilitas sistem di bawah kondisi beban kerja tinggi.

Kata Kunci: Jaringan Internet; Load Balancing; Optimalisasi Performa; Server; Waktu Respons

Abstract–The increasing number of internet users and the fact that each user has various needs result in very high and fluctuating internet traffic, which has an impact on the increasing workload on servers and network infrastructure. Due to the large workload experienced by the server, an approach is needed that can distribute the workload evenly among several resources, or what is known as a load balancing technique. Thus, the main objective of this research is to optimize internet network performance by implementing load balancing techniques and measuring the performance improvements that can be achieved. Load balancing methods are used to avoid imbalances that can cause one resource to be overburdened while others may not be used optimally. For testing, connections were made 1000 times with 100 requests/second, then increased gradually up to 11,000 connections with 1100 requests/second. Test results show that at low to medium connection levels (1000/100 to 6000/600), servers without load balancing tend to provide better response times. However, as connection rates increase (from 7000/700 to 11000/1100), servers with load balancing show superiority by having better response times than servers without the technique. Therefore, this research suggests that implementing a load balancing strategy is very important for maintaining optimal performance and system stability under high workload conditions.

Keywords: Internet Network; Load Balancing; Performance Optimization; Servers; Response Time

1. PENDAHULUAN

Jaringan komputer modern telah berkembang menjadi infrastruktur yang sangat kompleks dan kuat, dengan banyak server yang mendukung aplikasi dan layanan beragam. Perkembangan internet yang cepat di era saat ini berkontribusi terhadap peningkatan jumlah pengguna yang online dan menggunakan beragam layanan digital [1]. Perkembangan ini memicu munculnya teknologi terbaru, yaitu cloud computing. Cloud computing merupakan kumpulan sumber daya komputasi dan jaringan yang menawarkan model penagihan berbasis manajemen penyimpanan dan dukungan untuk berbagai aplikasi virtual [2]. Teknologi ini memberikan fleksibilitas dalam menyesuaikan ketersediaan sesuai kebutuhan, dengan mempertimbangkan faktor ekonomi dan lainnya [3]. Jumlah pengguna internet terus meningkat, dan masing-masing pengguna memiliki beragam kebutuhan seperti mengakses situs web, mengirim email, streaming video, bermain game online, dan banyak lagi [4]. Ini menghasilkan lalu lintas internet yang sangat tinggi dan berfluktuasi sehingga menyebabkan peningkatan beban kerja pada server serta infrastruktur jaringan [5]. Banyaknya beban kerja yang dialami oleh server maka diperlukan pendekatan yang dapat mendistribusikan beban kerja (workload) secara merata di antara beberapa sumber daya atau yang dikenal dengan teknik load balancing. Prinsip dasar dari load balancing adalah memastikan bahwa setiap sumber daya yang tersedia digunakan secara optimal, menghindari ketidakseimbangan yang dapat menyebabkan overloading pada satu sumber daya sementara yang lain mungkin berlebihan kapasitas [6]. Teknik load balancing berfokus untuk menghindari ketidakseimbangan yang dapat menyebabkan satu sumber daya terlalu terbebani sementara yang lain mungkin tidak digunakan secara optimal [7]–[9]. Teknik load balancing memiliki banyak fungsi selain

Journal of Information System Research (JOSH)
Volume 5, No. 1, Oktober 2023, pp 252−260
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i1.4386

sebagai pengatur lalu lintas jaringan juga bisa digunakan untuk menilai kondisi aplikasi dan konten pada setiap server sehingga dapat meningkatkan pelayanan serta menyederhanakan manajemen [10], [11]. Kondisi workload pada server dapat terjadi ketika jumlah klien yang mengakses layanan melebihi kapasitas maksimal dari satu server hingga menyebabkan pemutusan layanan, sehingga pembagian beban kerja dengan teknik load balancing dapat menjadi solusi [12]. Load balancing juga memfasilitasi pemeliharaan terencana, artinya ketika satu server perlu diperbarui atau diperbaiki, load balancer dapat mengalihkan lalu lintas ke server lainnya, sehingga tidak ada layanan yang terganggu [13]. Selain itu, load balancing memungkinkan organisasi untuk menambahkan server baru secara dinamis ketika diperlukan. Ini memungkinkan skalabilitas, sehingga server dapat menangani peningkatan lalu lintas tanpa mengorbankan kinerja.

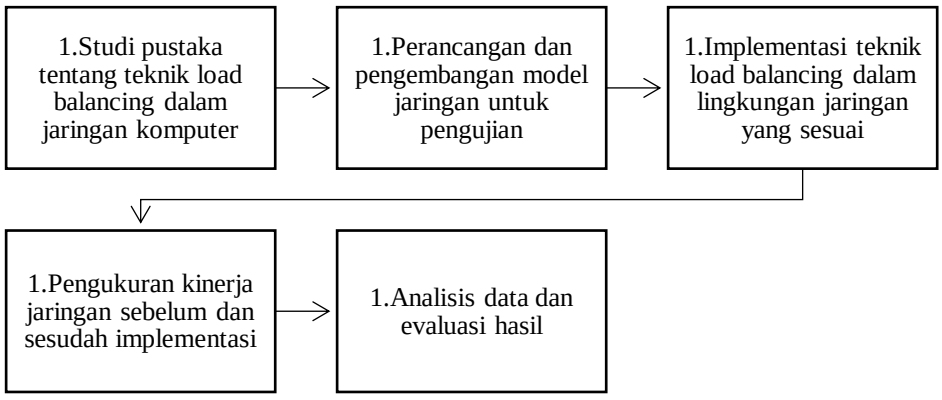
Terdapat beberapa penelitian yang menggunakan pendekatan load balancing untuk membagi beban pada server agar dapat melakukan pelayanan lebih optimal. Penelitian pertama, terkait tentang penggunaan teknik load balancing pada server yang digunakan untuk peningkatan kinerja website e-learning [14]. Pada penelitian tersebut, teknik load balancing memperoleh nilai response time 36,4 ms dengan uji sebanyak 500 koneksi dengan 10 permintaan/detik. Penelitian selanjutnya mengenai penerapan pendekatan load balancing pada web server yang menunjukkan bahwa pendekatan tersebut dapat mengatasi error request [15]. Pada penelitian ini pengujian response time langsung menggunakan 10.000 request. Terdapat penelitian mengenai penerapan metode load balancing untuk mengoptimalkan jaringan internet pada jaringan dengan lalu lintas yang tinggi [16]. Pada penelitian ini memperlihatkan bahwa metode yang diterapkan mampu menghasilkan delay sebesar 21 dan 24 ms. Penelitian berikutnya tentang penerapan pendekatan load balancing pada Mikrotik pada jaringan internet [17]. Hasil penelitian ini menunjukkan bahwasanya teknik load balancing per-connection classifier mendapatkan hasil yang optimal saat menggunakan bandwidth dari Internet Service Provider (ISP).

Perbedaan penelitian ini yaitu pada fokus pada penerapan metode load balancing untuk menguji performa sistem load balancing dan server tunggal dalam berbagai tingkat koneksi. Hal ini dilakukan karena berdasarkan dari ulasan penelitian sebelumnya, bahwa pada penerapan load balancing diperlukan pengujian untuk mengetahui waktu respons terbaik bagi pengguna. Namun, pengujian terhadap waktu respons ini dibutuhkan dengan berbagai tingkat koneksi dan perlu dibandingkan hasil pengujian arsitektur tanpa menggunakan load balancing dan yang menggunakan load balancing agar peningkatan kinerjanya dapat terlihat. Maka, Tujuan penelitian ini adalah untuk mengoptimalkan performa jaringan komputer dengan mengimplementasikan teknik load balancing dan mengukur peningkatan kinerja yang dapat dicapai. Untuk pengujian dilakukan koneksi sebanyak 1000 kali dengan 100 permintaan/detik, kemudian ditingkatkan bertahap sampai dengan koneksi sebanyak 11000 kali dengan 1100 permintaan/detik. Hal ini digunakan untuk melihat peningkatan kinerja server dengan arsitektur metode load balancing agar beban kerja akan tersebar merata di antara semua server dan menghasilkan waktu respons yang optimal.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian digunakan untuk menyusun penelitian agar dapat dilakukan secara sistematis, terdokumentasi, dan mendalam [18]. Penelitian ini melibatkan serangkaian tahap yang berurutan untuk mencapai hasil yang dapat diandalkan. Gambar 1 menunjukkan berbagai fase yang dijalani dalam penelitian ini.



Gambar 1. Alur Penelitian

Pada Gambar 1, menunjukkan alur penelitian dengan dimulai dengan tahap pertama adalah studi pustaka yang mendalam mengenai teknik load balancing dalam konteks jaringan komputer. Ini merupakan langkah awal yang penting untuk memahami dasar-dasar konsep dan aplikasi teknik tersebut. Langkah selanjutnya adalah perancangan dan pengembangan model jaringan yang akan digunakan untuk pengujian. Model ini dirancang dengan cermat agar mencerminkan situasi yang relevan dalam penggunaan sehari-hari. Setelah model jaringan selesai dirancang, tahap ketiga melibatkan implementasi teknik load balancing dalam lingkungan jaringan yang

Journal of Information System Research (JOSH)
Volume 5, No. 1, Oktober 2023, pp 252−260
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i1.4386

sesuai dengan kondisi yang telah ditetapkan. Implementasi ini dilakukan dengan cermat sesuai dengan rencana yang telah disusun sebelumnya. Untuk mengevaluasi dampak dari implementasi tersebut, tahap keempat adalah pengukuran kinerja jaringan sebelum dan sesudah penerapan teknik load balancing. Hal ini bertujuan untuk mengidentifikasi perubahan dalam kinerja jaringan setelah implementasi. Data yang diperoleh dari pengukuran tersebut kemudian dianalisis secara mendalam dalam tahap kelima. Hasil analisis ini dievaluasi untuk mengidentifikasi perubahan yang signifikan dalam kinerja jaringan setelah implementasi teknik load balancing.

Penelitian ini melibatkan pengujian metode load balancing untuk mendistribusikan sejumlah besar koneksi masuk ke setiap server web. Pengujian dilakukan dengan membandingkan kondisi sebelum dan sesudah penerapan metode load balancing untuk mengukur peningkatan kinerja. Hasil pengujian kami dibandingkan dengan metrik kinerja jaringan yang ada yaitu waktu respon. Pada tahap awal, studi literatur menjadi langkah pertama yang dilakukan. Setelah itu, proses pembuatan server web diinisiasi. Selanjutnya, server penyeimbang beban atau load balancer dibuat. Dalam pengukuran kinerjanya, jumlah koneksi ke server web ditingkatkan secara bertahap. Koneksi tersebut dialihkan baik ke satu server tunggal maupun ke grup dari beberapa server web yang telah terintegrasi dengan load balancer. Langkah berikutnya merupakan evaluasi terhadap teknik load balancing yang telah dibuat dengan menggunakan berbagai skenario pengujian. Hasil-hasil yang diperoleh dari pengujian tersebut kemudian dievaluasi untuk menentukan sejauh mana peran dari load balancing dalam meningkatkan kinerja sistem.

2.2 Metode Load Balancing

Masifnya perkembangan website dengan berbagai manfaat dan kegunaannya, maka penting untuk menyiapkan arsitektur server yang sesuai dengan kebutuhan [19]. Meningkatnya permintaan layanan online dan aplikasi berbasis web, server yang berada di belakang layar seringkali menghadapi tekanan besar. Teknik load balancing memecahkan tantangan ini dengan mendistribusikan beban kerja secara merata di antara sejumlah server yang tersedia [20]. Load balancing adalah teknik yang digunakan untuk memastikan bahwa beban kerja atau permintaan dari pengguna atau perangkat didistribusikan secara merata di antara beberapa sumber daya yang tersedia, seperti server, komputer, atau node jaringan [21]. Dengan load balancing, setiap sumber daya dapat digunakan secara optimal, meningkatkan ketersediaan layanan, menghindari overloading, dan memberikan pengalaman pengguna yang lebih baik [22]. Load balancing merujuk pada bagaimana membagi atau mendistribusikan beban kerja di antara sumber daya yang tersedia [17]. Berikut beberapa teknik yang digunakan pada pendekatan load balancing:

1) Round Robin
   Dalam teknik round robin, setiap permintaan berikutnya akan diberikan kepada server berikutnya dalam daftar server yang tersedia. Sehingga untuk menentukan server yang akan melayani permintaan dapat dihitung dengan persamaan (1).

   $$\text{Server x} = (\text{Permintaan Sebelumnya} + 1)\% \text{ jumlah server} \quad (1)$$

   dimana Server x merupakan server yang akan melayani permintaan.
2) Least Connections
   Pada teknik least connections, permintaan akan diberikan kepada server yang memiliki jumlah koneksi yang paling sedikit saat permintaan diterima. Sehingga untuk menentukan server yang akan melayani permintaan dapat dihitung dengan persamaan (2).

   $$\text{Server x} = \text{Server y} \quad (2)$$

   dimana Server x merupakan server yang akan melayani permintaan, sedangkan Server y merupakan server dengan jumlah koneksi terendah.
3) Least Response Time
   Pada teknik least response time, permintaan dialihkan ke server yang memiliki waktu respons tercepat saat permintaan diterima. Sehingga untuk menentukan server yang akan melayani permintaan dapat dihitung dengan persamaan (3).

   $$\text{Server x} = \text{Server z} \quad (3)$$

   dimana $Server\ x$ merupakan server yang akan melayani permintaan, sedangkan $Server\ z$ server dengan waktu respons tercepat

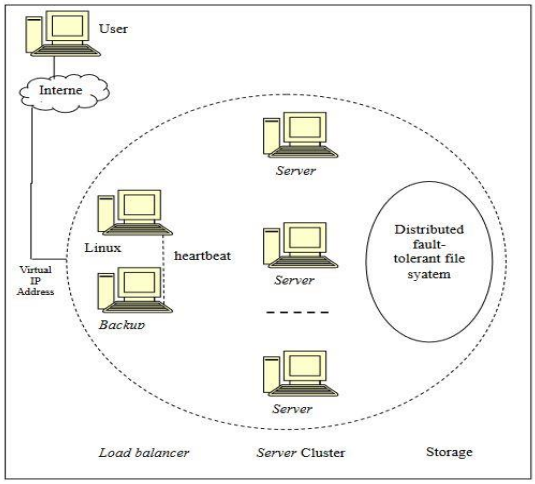
2.3 Skenario Pengujian

Skenario pengujian ini dirancang untuk menguji performa sistem load balancing dan server tunggal dalam berbagai tingkat koneksi. Untuk pengujian pertama dilakukan koneksi sebanyak 1000 kali dengan 100 permintaan/detik, kemudian ditingkatkan bertahap sampai dengan koneksi sebanyak 11000 kali dengan 1100 permintaan/detik. Tujuan utama dari penggunaan load balancing dalam skenario ini adalah untuk memastikan waktu respons terbaik bagi pengguna. Load balancer akan secara cerdas mendistribusikan permintaan dari pengguna ke server-server tersebut sesuai dengan kapasitas koneksi masing-masing server. Dengan demikian, beban kerja akan tersebar merata di antara semua server dan menghasilkan waktu respons yang optimal.

Journal of Information System Research (JOSH)
Volume 5, No. 1, Oktober 2023, pp 252−260
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i1.4386

2.4 Rancangan Sistem Server

Arsitektur server load balancing yang digunakan dalam penelitian ini terdiri dari tiga bagian, seperti yang diilustrasikan pada Gambar 2. Bagian-bagian tersebut mencakup server penyeimbang beban, satu set server aplikasi, dan server database.



Gambar 2. Arsitektur Server Load Balancer

Pada Gambar 2, komponen pertama adalah server penyeimbang beban, yang bertanggung jawab untuk mendistribusikan beban koneksi ke setiap server web. Server penyeimbang beban ini memiliki alamat IP tertentu yang memungkinkannya untuk terhubung dengan setiap server aplikasi dan komputer klien. Komponen kedua terdiri dari serangkaian server web. Dalam konteks penelitian ini, kami menggunakan dua unit sebagai server web. Masing-masing dari dua unit server ini menjalankan program aplikasi web yang dapat diakses oleh klien. Komponen ketiga adalah penyimpanan data atau basis data, yang dalam penelitian ini ditempatkan langsung di setiap server web bukan secara individu. Pengujian untuk sistem ini dilakukan dalam lingkungan mesin virtual dengan spesifikasi seperti yang disajikan pada Tabel 1.

Tabel 1. Spesifikasi mesin virtual

Komponen	CPU	RAM	SSD	Sistem Operasi
Komputer Utama	AMD Ryzen 5 (8 CPU) - 2,0 GHz	8 GB	250 GB	Windows 10
Mesin Virtual Load Balancer	AMD Ryzen 5 (1 CPU) - 2,0 GHz	1 GB	20 GB	Ubuntu Server
Mesin Virtual Server 1	AMD Ryzen 5 (1 CPU) - 2,0 GHz	1 GB	20 GB	Ubuntu Server
Mesin Virtual Server 2	AMD Ryzen (1 CPU) - 2,0 GHz	1 GB	20 GB	Ubuntu Server
Mesin Virtual Klien	AMD Ryzen 5 (1 CPU) - 2,0 GHz	1 GB	20 GB	Ubuntu Server

Tabel 1 menyajikan spesifikasi komputer utama dan mesin virtual yang digunakan dalam pengujian sistem Anda, termasuk informasi tentang CPU, RAM, SSD, dan sistem operasi yang diinstal pada masing-masing komponen.

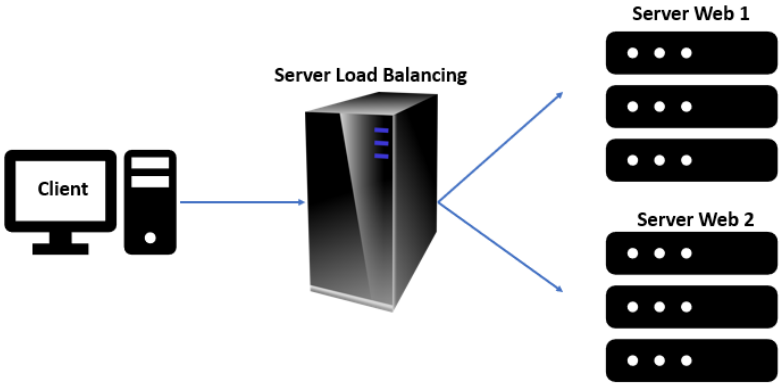
3. HASIL DAN PEMBAHASAN

Pengujian berbagai tingkat koneksi pada server load balancing dilakukan untuk mengevaluasi kinerja sistem server dalam mengelola sejumlah besar permintaan yang datang dari klien dalam periode waktu tertentu. Parameter utama yang diukur dalam proses pengujian ini adalah waktu respons. Pengukuran ini dilakukan sebelum dan setelah berhasil mengimplementasikan sistem penyeimbangan beban. Untuk mendirikan sistem penyeimbangan beban, mesin virtual pada komputer utama digunakan sebagai dasar infrastruktur server. Dalam studi ini, dua model arsitektur server dianalisis. Model pertama adalah arsitektur server tunggal, yaitu skema untuk sistem yang tidak menggunakan teknik penyeimbangan beban. Arsitektu pertama divisualisasikan arsitekturnya pada Gambar 3.

Gambar 3. Model Arsitektur Server Tunggal

Journal of Information System Research (JOSH)
Volume 5, No. 1, Oktober 2023, pp 252−260
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i1.4386

Pada Gambar 3, arsitektur tersebut memperlihatkan bahwa hanya ada satu server yang bertugas menangani semua permintaan dari pengguna. Sehingga, ketika jumlah permintaan meningkat secara signifikan, semua permintaan tersebut dialihkan ke satu server tersebut saja. Arsitektur kedua melibatkan implementasi serangkaian server yang didedikasikan untuk memproses permintaan masuk secara bersama-sama. Arsitektur ini menggunakan konsep mekanisme penyeimbangan beban dan terdiri dari sebuah server penyeimbang beban serta dua web server. Seperti yang diketahui bahwa konsep dari load balancing yaitu untuk membagi beban kerja atau permintaan dari pengguna atau perangkat didistribusikan secara merata di antara beberapa sumber daya yang tersedia. Berbeda dengan model pertama, dalam arsitektur ini setiap permintaan masuk dari klien diproses oleh dua atau lebih server secara simultan. Model arsitektur load balancing pada kasus ini divisualisasikan pada Gambar 4.



Gambar 4. Model Arsitektur Load Balancing

Pada Gambar 4 menampilkan desain sistem server yang terdiri dari server penyeimbang beban, Server web 1, Server web 2, dan klien. Semua komponen ini saling berinteraksi dalam satu jaringan melalui switch. Dalam susunan ini, klien memiliki kemampuan untuk mengakses layanan melalui jaringan dan mendapatkan layanan dari Server web 1 dan Server web 2. Alamat IP yang telah ditetapkan pada setiap perangkat dalam jaringan tersebut seperti yang tersaji pada Tabel 2.

Tabel 2. Nama Komponen dan Alamat IP

Komponen	Alamat IP
Server Load balancing	192.168.109.184
Web Server 1	192.168.109.179
Web Server 2	192.168.109.178
Client	192.168.109.182

Tabel 2 menyajikan informasi mengenai alamat IP yang telah diinstal pada setiap perangkat dalam jaringan. Dalam tabel tersebut, terlihat bahwa server penyeimbang beban memiliki alamat IP 192.168.109.182. Server penyeimbang beban ini berfungsi untuk mendistribusikan permintaan layanan yang datang dari klien ke masing-masing server web. Saat klien mencoba mengakses layanan melalui server web, mereka secara otomatis dialihkan ke server penyeimbang beban. Server ini kemudian meneruskan permintaan tersebut ke salah satu dari server web yang ada di belakangnya. Dalam skenario ini, klien tidak perlu mengetahui mana dari server web yang akan memproses permintaannya karena tugas tersebut ditangani oleh server penyeimbang beban. Server ini meneruskan permintaan layanan masuk ke salah satu dari server web sesuai dengan tabel alokasi yang telah ditentukan.

Penelitian ini mencakup pengujian untuk mengamati nilai response time sebagai indikator kinerja layanan ketika teknik load balancing diterapkan pada sistem server. Namun, pengujian terhadap waktu respons ini dibutuhkan dengan berbagai tingkat koneksi dan perlu dibandingkan hasil pengujian arsitektur tanpa menggunakan load balancing dan yang menggunakan load balancing agar peningkatan kinerjanya dapat terlihat. Maka, perbedaan penelitian ini yaitu pada penelitian ini fokus pada penerapan metode load balancing untuk menguji performa sistem load balancing dan server tunggal dalam berbagai tingkat koneksi. Selama proses pengujian, klien menggunakan perangkat lunak benchmarking httperf untuk memulai serentak sejumlah koneksi dan mendapatkan variabel waktu respons. Pengujian implementasi load balancing dengan rangkaian koneksi dari client ke web server melalui software benchmarking httperf. Permintaan-permintaan dilakukan secara bertahap, mulai dari koneksi sebanyak 1000 kali dengan 100 permintaan/detik, kemudian ditingkatkan secara bertahap sampai dengan koneksi sebanyak 11000 kali dengan 1100 permintaan/detik. Hal ini digunakan untuk melihat peningkatan kinerja server dengan arsitektur metode load balancing agar beban kerja akan tersebar merata di antara semua server dan menghasilkan waktu respons yang optimal. Pada setiap skenario pengujian, hasil dapat berbeda-beda dikarenakan durasi serta antrian proses tiap-tiap skenario yang unik dan beragam, hal ini mempengaruhi nilai waktu respons, dimana semakin banyak jumlah permintaan kepada web server maka waktu respons juga akan semakin meningkat. Berdasarkan nilai response time yang didapatkan dalam pengujian tersebut, kita dapat mengevaluasi performa

Journal of Information System Research (JOSH)
Volume 5, No. 1, Oktober 2023, pp 252−260
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i1.4386

aplikasi dalam konteks kedua desain arsitektur sistem. Untuk hasil pengujian waktu respons server dengan menerapkan load balancing tersaji pada Tabel 3.

Tabel 3. Hasil Pengujian Waktu Respons (ms) Server Menggunakan Load Balancing

No	Waktu Respons (ms) Server dengan Load balancing						
	Tingkat Koneksi	Pengujian ke- 1	Pengujian ke-2	Pengujian ke-3	Pengujian ke- 4	Pengujian ke-5	Hasil Rata-rata Pengujian
1	1000/100	2.6	2.4	2.5	2.5	2.4	2.5
2	2000/200	2.3	2.3	2.3	2.3	2.3	2.3
3	3000/300	2.9	2.9	3	2.9	3	2.94
4	4000/400	2.8	2.9	2.8	2.9	2.8	2.84
5	5000/500	3	2.8	2.9	2.9	2.8	2.88
6	6000/600	3.3	3.1	3.3	3.1	4.5	3.46
7	7000/700	2.2	3.6	3.7	4.3	3.6	3.48
8	8000/800	5.7	11	12.4	5.8	8.7	8.72
9	9000/900	2.1	12.6	11.5	9.7	11.9	9.56
10	10000/1000	28	24.8	19.1	25.5	48.5	29.18
11	11000/1100	70.7	75.6	70	97.2	86.8	80.06

Tabel 3 menunjukkan hasil pengujian waktu respons (dalam milidetik) dari server yang menerapkan teknik load balancing pada berbagai tingkat koneksi. Setiap baris dalam tabel mencerminkan satu tingkat koneksi, dan setiap tingkat koneksi diuji sebanyak lima kali. Baris pertama, misalnya, merujuk pada server dengan tingkat koneksi 1000/100. Hasil dari lima uji coba untuk server ini berkisar antara 2.4 ms hingga 2.6 ms, dengan rata-rata waktu respons adalah 2.5 ms. Baris kedua menunjukkan data untuk server dengan tingkat koneksi 2000/200. Waktu respons untuk semua lima uji coba konstan di 2.3 ms. Baris ketiga hingga kelima menunjukkan data untuk server dengan tingkat koneksi 3000/300, 4000/400, dan 5000/500 masing-masing. Rata-rata waktu respons berkisar antara 2.84 ms hingga 2.94 ms. Mulai dari baris keenam hingga sebelas memperlihatkan peningkatan signifikan dalam rata-rata waktu respons saat meningkatkan tingkat koneksi dari 6000/600 menjadi sebesar 11000/1100. Rata-ratanya meningkat secara dramatis dari sekitar 3ms sampai ke angka 80ms. Secara keseluruhan, tabel ini menggambarkan bagaimana peningkatan dalam kapasitas jaringan atau bandwith (tingkat koneksi) dapat berdampak pada peningkatan waktu respons saat menggunakan teknik load balancing. Untuk hasil pengunjian waktu respons server tan menggunakan load balancing tersaji pada Tabel 4.

Tabel 4. Hasil Pengujian Waktu Respons (ms) Server Tanpa Load Balancing

No	Waktu Respons (ms) Server tanpa Load balancing						
	Tingkat Koneksi	Pengujian ke-1	Pengujian ke-2	Pengujian ke-3	Pengujian ke- 4	Pengujian ke-5	Hasil Rata-rata Pengujian
1	1000/100	1.2	1.1	1.2	1.2	1.1	1.16
2	2000/200	1.2	1.2	1.2	1.1	1.1	1.175
3	3000/300	1.9	1.9	1.9	2	2	1.925
4	4000/400	1.7	1.7	1.7	1.7	1.7	1.7
5	5000/500	1.6	1.6	1.6	1.6	1.3	1.54
6	6000/600	1.3	1.2	1.7	1.9	1.7	1.56
7	7000/700	2.1	7.2	2	1.8	2.2	3.06
8	8000/800	7.5	3.2	4.1	5.5	4	4.86
9	9000/900	9.4	8.3	10.6	6.6	4.7	7.92
10	10000/1000	63	107.5	34	91.5	29.9	65.18
11	11000/1100	215.8	160.4	94.1	159.4	187.9	163.52

Tabel 4 menunjukkan hasil pengujian waktu respons (dalam milidetik) dari server yang tidak menerapkan teknik load balancing pada berbagai tingkat koneksi. Setiap baris dalam tabel mencerminkan satu tingkat koneksi, dan setiap tingkat koneksi diuji sebanyak lima kali. Baris pertama, misalnya, merujuk pada server dengan tingkat koneksi 1000/100. Hasil dari lima uji coba untuk server ini berkisar antara 1.1 ms hingga 1.2 ms, dengan rata-rata waktu respons adalah 1.16 ms. Baris kedua menunjukkan data untuk server dengan tingkat koneksi 2000/200. Waktu respons untuk semua lima uji coba berkisar antara 1.1 ms hingga 1.2 ms, dengan rata-rata waktu respons adalah 1.175 ms. Baris ketiga hingga kelima menunjukkan data untuk server dengan tingkat koneksi 3000/300, 4000/400, dan 5000/500 masing-masing. Rata-rata waktu respons berkisar antara 1.54 ms hingga 1.925 ms. Mulai dari baris keenam hingga sebelas memperlihatkan peningkatan dalam rata-rata waktu respons saat meningkatkan tingkat koneksi dari 6000/600 menjadi sebesar 11000/11000. Rata-ratanya meningkat secara signifikan dari sekitar

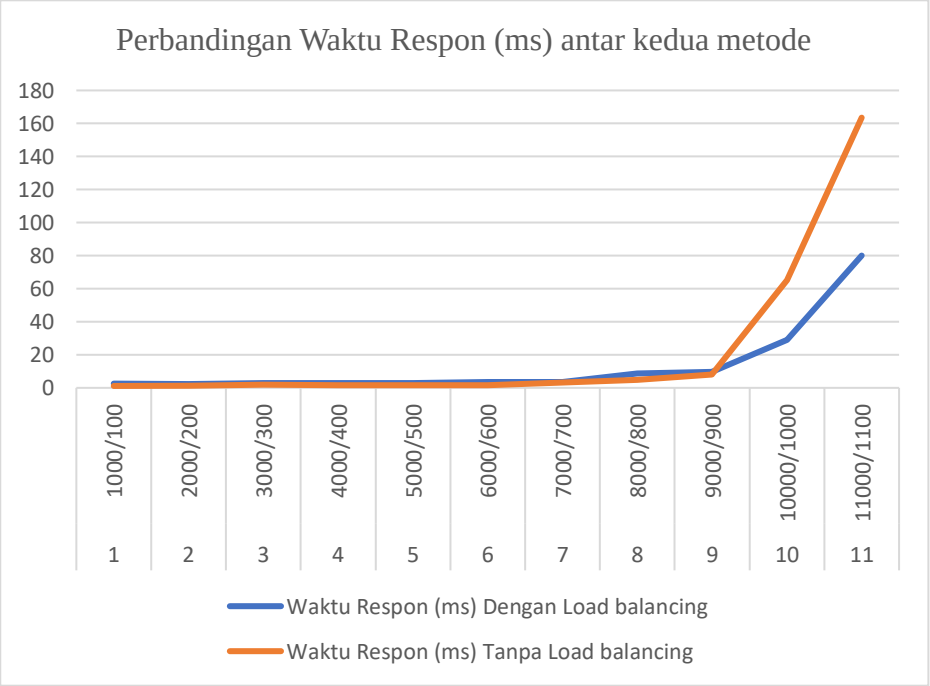
Journal of Information System Research (JOSH)
Volume 5, No. 1, Oktober 2023, pp 252−260
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i1.4386

3 ms sampai ke angka 163 ms. Hasil rata-rata untuk hasil pengujian arsitektur menggunakan load balancing dan tanpa load balancing disajikan pada Tabel 5.

Tabel 5. Hasil Rata-Rata Pengujian Waktu Respons (ms) Server Untuk Kedua Arsitektur

No	Tingkat Koneksi	Rata-rata Waktu Respons (ms)	
		Dengan Load Balancing	Tanpa Load Balancing
1	1000/100	2.5	1.16
2	2000/200	2.3	1.175
3	3000/300	2.94	1.925
4	4000/400	2.84	1.7
5	5000/500	2.88	1.54
6	6000/600	3.46	1.56
7	7000/700	3.48	3.06
8	8000/800	8.72	4.86
9	9000/900	9.56	7.92
10	10000/1000	29.18	65.18
11	11000/1100	80.06	163.52

Pada Tabel 5 menunjukkan perbandingan waktu respons rata-rata (dalam milidetik) antara server yang menerapkan teknik load balancing dan server yang tidak menggunakan teknik load balancing pada berbagai tingkat koneksi. Setiap baris dalam tabel mencerminkan satu tingkat koneksi. Misalnya, pada baris pertama untuk tingkat koneksi 1000/100, waktu respons rata-rata untuk server dengan load balancing adalah 2.5 ms, sedangkan untuk server tanpa load balancing adalah 1.16 ms. Pada baris kedua hingga keenam, kita dapat melihat bahwa waktu respons rata-rata dari server dengan load balancing lebih tinggi dibandingkan dengan server tanpa load balancing. Namun, mulai dari baris ketujuh hingga sebelas, saat kapasitas jaringan atau bandwidth meningkat (dari 7000/700 menjadi sebesar 11000/11000), terlihat bahwa server dengan teknik load balancing memiliki waktu respons rata-rata yang lebih rendah dibandingkan dengan server tanpa teknik tersebut. Hal ini menunjukkan bahwa dalam kondisi beban tinggi atau kapasitas jaringan besar, penerapan teknik load balancing dapat membantu dalam mengoptimalkan dan memperbaiki performa sistem secara keseluruhan serta memberikan pengalaman pengguna yang lebih baik. Berdasarkan Tabel 5, kemudian disusun dalam bentuk grafik untuk memudahkan melihat perbandingan dari hasil pengujian terhadap dua arsitektur yang tersaji pada Gambar 5.



Gambar 5. Hasil Nilai Waktu Respons Antara Metode Server Tunggal dan Load Balancing

Pada Gambar 5, menampilkan grafik perbandingan antara metode load balancing dengan server tunggal. Terlihat bahwa server dengan teknik load balancing memiliki waktu respons rata-rata yang lebih rendah dibandingkan dengan server tanpa teknik tersebut. Selain itu, implementasi load balancing mampu memproses semua permintaan masuk secara efisien dan membantu mencapai nilai Quality of Service (QoS) yang baik. Hasil pengujian tersebut tersaji pada Gambar 6.

Journal of Information System Research (JOSH)
Volume 5, No. 1, Oktober 2023, pp 252−260
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i1.4386

```
loadbalancing@httperf:~$ httperf --server=192.168.109.184 --uri=/index.html --num-conns=11000 --rate
=1100
httperf --client=0/1 --server=192.168.109.184 --port=80 --uri=/index.html --rate=1100 --send-buffer=
4096 --recv-buffer=16384 --num-conns=11000 --num-calls=1
httperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 33

Total: connections 11000 requests 11000 replies 11000 test-duration 10.091 s

Connection rate: 1090.1 conn/s (0.9 ms/conn, <=189 concurrent connections)
Connection time [ms]: min 2.1 avg 73.7 max 328.3 median 73.5 stddev 45.6
Connection time [ms]: connect 1.4
Connection length [replies/conn]: 1.000

Request rate: 1090.1 req/s (0.9 ms/req)
Request size [B]: 78.0

Reply rate [replies/s]: min 1081.9 avg 1083.0 max 1084.0 stddev 1.4 (2 samples)
Reply time [ms]: response 70.7 transfer 1.7
Reply size [B]: header 254.0 content 11523.0 footer 0.0 (total 11777.0)
Reply status: 1xx=0 2xx=11000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 0.16 system 9.64 (user 1.6% system 95.5% total 97.1%)
Net I/O: 12620.0 KB/s (103.4*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
```

Gambar 6. Salah Satu Contoh Hasil Pengujian Koneksi 11000/1100 Menggunakan Load Balancing

Pada Gambar 6, terlihat bahwa untuk pengujian koneksi 11000/1100 menggunakan load balancing dapat melakukan proses selutu permintaan. Untuk arsitektur yang tanpa load balancing pada pengujian dengan beban sebanyak 11.000 request pada tingkat koneksi sebesar1100 koneks/detik, menunnjukkan bahwa server tidak mampu memproses semua request yang diterima. Hal tersebut ditandai dengan terdeteksinya kesalahan yang tercatat selama proses pengujian. Hasil pengujian tersebut disajikan pada Gambar 7.

```
loadbalancing@httperf:~$ httperf --server=192.168.109.178 --uri=/index.html --num-conns=11000 --rate
=1100
httperf --client=0/1 --server=192.168.109.178 --port=80 --uri=/index.html --rate=1100 --send-buffer=
4096 --recv-buffer=16384 --num-conns=11000 --num-calls=1
httperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 15

Total: connections 10316 requests 10316 replies 10316 test-duration 17.720 s

Connection rate: 582.2 conn/s (1.7 ms/conn, <=1022 concurrent connections)
Connection time [ms]: min 6.0 avg 786.8 max 15040.4 median 150.5 stddev 1647.8
Connection time [ms]: connect 569.2
Connection length [replies/conn]: 1.000

Request rate: 582.2 req/s (1.7 ms/req)
Request size [B]: 78.0

Reply rate [replies/s]: min 155.0 avg 686.1 max 982.3 stddev 460.9 (3 samples)
Reply time [ms]: response 215.8 transfer 1.8
Reply size [B]: header 254.0 content 11523.0 footer 0.0 (total 11777.0)
Reply status: 1xx=0 2xx=10316 3xx=0 4xx=0 5xx=0

CPU time [s]: user 0.34 system 16.98 (user 1.9% system 95.8% total 97.7%)
Net I/O: 6739.8 KB/s (55.2*10^6 bps)

Errors: total 684 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 684 addrunavail 0 ftab-full 0 other 0
```

Gambar 7. Salah Satu Contoh Error Dari Hasil Pengujian Koneksi 11000/1100 Tanpa Load Balancing

Pada Gambar 7 memperlihatkan fenomena lain saat melakukan pengujian dengan beban sebanyak 11.000 request pada tingkat koneksi sebesar1100 koneks/detik pada arsitektur tanpa load balancing. Server tidak mampu memproses semua request yang diterima. Hal tersebut ditandai dengan terdeteksinya kesalahan yang tercatat selama proses pengujian.

4. KESIMPULAN

Penelitian ini berhasil mengoptimalkan performa jaringan komputer melalui penerapan teknik load balancing. Hasil penelitian menunjukkan waktu respons yang relatif lebih kecil dibandingkan tanpa load balancing. Teknik load balancing sangat efektif untuk kondisi beban kerja tinggi dan dapat membantu dalam menjaga stabilitas sistem serta memberikan performa optimal. Namun, untuk beban kerja rendah atau kapasitas jaringan kecil, overhead dari proses load balancing mungkin akan membuatnya kurang efisien dibandingkan dengan sistem tanpa load balancing. Hasil ini menunjukkan bahwa penyeimbangan beban server memungkinkan distribusi beban koneksi diterima secara merata di antara sejumlah server web. Hal ini memiliki dampak signifikan terhadap penurunan response time saat server memproses sebuah request. Sebagai rekomendasi untuk penelitian di masa mendatang, perlu dipertimbangkan penambahan jumlah server penyeimbang beban sebagai cadangan. Langkah ini bertujuan untuk mendukung model toleransi kesalahan dan meningkatkan ketersediaan layanan dari sistem server. Meskipun

Journal of Information System Research (JOSH)
Volume 5, No. 1, Oktober 2023, pp 252−260
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i1.4386

demikian, penelitian ini memiliki keterbatasan dalam hal lingkup dan implementasi yang dapat diperbaiki dalam penelitian selanjutnya. Dari pengujian yang telah dilakukan dapat diamati bahwa untuk tingkat koneksi rendah hingga sedang (1000/100 hingga 6000/600), server tanpa load balancing cenderung memiliki waktu respons yang lebih baik dibandingkan dengan server dengan load balancing. Hal ini mungkin disebabkan oleh overhead tambahan yang diperlukan oleh proses load balancing itu sendiri. Namun, ketika tingkat koneksi meningkat (7000/700 hingga 11000/1100), terlihat bahwa server dengan load balancing dapat memberikan waktu respons yang lebih baik dibandingkan server tanpa load balancing. Ini menunjukkan bahwa dalam skenario beban tinggi atau kapasitas jaringan besar, penerapan teknik load balancing dapat membantu mengoptimalkan performa sistem dan memperbaiki pengalaman pengguna.

REFERENCES

[1] E. Coll, The Internet and Cloud Computing. Las Vegas: Teracom Training Institute, 2023.
[2] D. Comer, The Cloud Computing Book: The Future of Computing Explained. Florida: CRC Press, 2021.
[3] Y. Afrianto and A. H. Hendrawan, “Implementasi Data Center Untuk Penempatan Host Server Berbasis Private Cloud Computing,” Krea-Tif, vol. 7, no. 1, pp. 50–59, 2019, doi: 10.32832/kreatif.v7i1.2031.
[4] A. R. Sofyan and S. D. Y. Kusuma, “Implementasi Load Balancing Web Server menggunakan Haproxy pada Virtual Server Direktorat SMK Kemendikbudristek,” J. Pendidik. Tambusai, vol. 6, no. 2, pp. 9669–9682, 2022.
[5] B. P. Mulla, C. R. Krishna, and R. K. Tickoo, Load Balancing Algorithm for Efficient VM Allocation in Heterogeneous Cloud. New York: SSRN, 2020.
[6] S. D. Riskiono and D. Pasha, “Analisis Perbandingan Server Load Balancing dengan Haproxy & Nginx dalam Mendukung Kinerja Server E- Learning,” J. Telekomun. dan Komput., vol. 10, no. 3, pp. 135–144, 2020, doi: 10.22441/incomtech.v10i3.8751.
[7] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, “Load balancing techniques in cloud computing environment: A review,” J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 7, pp. 3910–3933, 2022, doi: https://doi.org/10.1016/j.jksuci.2021.02.007.
[8] S. Mohanty, Evolutionary Approaches for Load Balancing in Cloud Computing. Lagos: Bibi Incorporated, 2022.
[9] S. D. Riskiono and D. Darwis, “Peran Load Balancing Dalam Meningkatkan Kinerja Web Server Di Lingkungan Cloud,” Krea-TIF, vol. 8, no. 2, p. 1, 2020, doi: 10.32832/kreatif.v8i2.3503.
[10] P. Kumar and R. Kumar, “Issues and Challenges of Load Balancing Techniques in Cloud Computing: A Survey,” ACM Comput. Surv., vol. 51, no. 6, Feb. 2019, doi: 10.1145/3281010.
[11] I. R. Management Association, Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing. Pennsylvania: IGI Global, 2021.
[12] D. K. Hakim, D. Y. Yulianto, and A. Fauzan, “Pengujian Algoritma Load Balancing pada Web Server Menggunakan NGINX,” J. Ris. Sains dan Teknol., vol. 3, no. 2, pp. 85–92, 2019, doi: 10.30595/jrst.v3i2.5165.
[13] T. S. Hendrana and I. M. Suartana, “Penerapan Container Load Balancing untuk Manajemen Trafik pada Learning Management System,” JINACS J. Informatics Comput. Sci., vol. 04, no. 02, pp. 169–182, 2022.
[14] S. D. Riskiono and D. Pasha, “Analisis Metode Load Balancing Dalam Meningkatkan Kinerja Website E-Learning,” J. Teknoinfo, vol. 14, no. 1, pp. 22–26, 2020, doi: 10.33365/jti.v14i1.466.
[15] H. S. Harefa, J. Triyono, and S. Raharjo, “Implementasi Load Balancing Web Server Untuk Optimalisasi Kinerja Web Server Dan Database,” J. Jarkom, vol. 09, no. 01, pp. 10–20, 2021.
[16] T. Octavriana, K. Joni, and A. F. Ibadillah, “Optimalisasi Jaringan Internet Dengan Load Balancing Pada High Traffic Network,” J. Tek. Inform., vol. 14, no. 1, pp. 28–39, 2021, doi: 10.15408/jti.v14i1.15018.
[17] B. G. Malau, “Implementasi Load Balancing Mikrotik Jaringan Internet Di Pardamean Sibisa, Ajibata, Toba Samosir, Sumatra Utara,” JCS-TECH J. Comput. Sci. Technol., vol. 2, no. 1, pp. 20–29, 2022, doi: 10.54840/jcstech.v2i1.23.
[18] R. D. Gunawan, R. Napianto, R. I. Borman, and I. Hanifah, “Penerapan Pengembangan Sistem Extreme Programming Pada Aplikasi Pencarian Dokter Spesialis di Bandar lampung Berbasis Android,” J. Format, vol. 8, no. 2, pp. 148–157, 2019.
[19] I. Ahmad, E. Suwarni, R. I. Borman, A. Asmawati, F. Rossi, and Y. Jusman, “Implementation of RESTful API Web Services Architecture in Takeaway Application Development,” in International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS), 2022, pp. 132–137. doi: 10.1109/ICE3IS54102.2021.9649679.
[20] Z. Bustomi, M. Syahiruddin, M. I. Afandi, and K. F. H. Holle, “Load Balancing Web Server Menggunakan Nginx pada Lingkungan Virtual,” J. Inform. J. Pengemb. IT, vol. 5, no. 1, pp. 32–36, 2019.
[21] F. K. Saiputra and A. R. Mukti, “Implementasi Load Balancing Menggunakan Metode PCC (Per Connection Classifier) Pada Yayasan Bina Jaya Palembang,” J. Jupiter, vol. 15, no. 1, pp. 489–499, 2023.
[22] A. Panwar, A. Singh, A. Dixit, and G. Parashar, “Cloud Computing and Load Balancing: A Review,” in International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), 2022, pp. 334–343. doi: 10.1109/CISES54857.2022.9844367.