ENHANCING NETWORK PERFORMANCE LOAD BALANCING IN CYBER CAFE NETWORKS WITH DIJKSTRA ALGORITHM ON MIKROTIK

Andi Prayogi[*1], Phak Len Al Eh Kan[2], Muhammad Akbar Syabana Pane[3], Rahmad Dian[4], Ratu Mutiara Siregar[5], Hasanal Fachri Satia Simbolon[6]

[1,3,4,5,6]Department Information Systems and Technology, Institut Teknologi Sawit Indonesia, Indonesia
[2]Faculty of Electronic Engineering & Technology, Universiti Malaysia Perlis, Malaysia
Email: [1]andiprayogi@itsi.ac.id

Abstract

The internet has become a fundamental necessity in various activities today. Stream Cyber Cafe, as an internet service provider, faces the challenge of maintaining network quality so that users can comfortably engage in activities such as gaming, streaming services, and social media. Load balancing, utilizing multiple internet sources from various ISPs, becomes a solution to enhance network efficiency and responsiveness. This research focuses on implementing the Dijkstra algorithm on MikroTik devices to determine the shortest path based on DNS server latency from various internet service providers (ISP). The main steps include configuring MikroTik devices, analyzing latency connections to DNS servers, and employing the Dijkstra algorithm. The Dijkstra algorithm, utilizing a Greedy approach, considers the minimum weight from the starting node to other nodes. Testing using the "PING" command provides information on the number of hops or steps required to reach each DNS server. Dijkstra adapts the shortest path based on latency, yielding optimal load balancing efficiency. Configuring MikroTik features, such as Firewall Mangle, Routing Table, and Routing Gateway, supports the functionality of the Dijkstra algorithm. Test results show that each ISP has a different shortest path to DNS servers, and the Dijkstra algorithm can determine the shortest path by considering time or latency factors. Although the author acknowledges some technical challenges during implementation, the proposed solution successfully overcomes these challenges. Thus, the use of the Dijkstra algorithm on MikroTik proves its effectiveness in enhancing the performance and reliability of the network in the Stream Cyber Cafe environment.

Keywords: *Dijkstra Algorithm, DNS, Mikrotik.*

1. INTRODUCTION

The internet, as an information medium, serves as an effective and efficient means for the exchange and dissemination of information without being hindered by distance, time differences, and other factors [1]. The quality of information exchange and dissemination relies on the condition of network traffic. A well-functioning network is characterized by data traffic that swiftly and accurately exchanges information [2]. However, it should be noted that in internet activities, data traffic involved in gaming, streaming services, and social media activities follows different paths [3]. Internet activities necessitate an internet source from an Internet Service Provider (ISP) to carry out these activities. Each ISP has its own advantages in determining its internet routes [4].

A common element used in the internet is the use of DNS (Domain Name Server). DNS serves as the transportation used by internet sources (ISPs) to deliver requested packets and retrieve packets provided by the destination [5]. Numerous DNS services are freely available on the internet. The determination of a good DNS is typically made by the

Internet Service Provider (ISP) [6]. Therefore, it is crucial to conduct a connectivity test to ascertain the most optimal DNS for internet source utilization [7].

Activities and engagements in this era of modernization heavily rely on the internet to function [8]. The internet has permeated various fields to assist human activities and endeavors [9]. Many capitalize on this by establishing businesses that leverage internet networks [10]. Stream Cyber Cafe is one such venture that offers users computer rental services for gaming, streaming, social media, and more. However, its usage often encounters challenges such as delays in accessing social media and disruptions during gaming. This can be attributed to insufficient network capacity, leading to uncomfortable disturbances [11]. The simultaneous use of gaming, streaming services, and social media creates congestion in data traffic, coupled with a lack of multitasking in determining the appropriate data pathways [12].

In internet network connections, to access a service such as playing games and social media, a specific entry point called a communication port is utilized. The communication port can be seen as an entrance or "address" used by computers or devices in the network to communicate with each other. Each

entry point has a unique identifier known as a port number [13]. Similar to an address on a street that helps identify a home's location, ports aid in identifying the destination or source of data transmitted through the network. For example, if a computer wants to access social media and play games, the network service will direct it to port 443 for social media and port 37170 for gaming. This is akin to a building's door numbering system, where specific doors are designated for entry and exit. These ports have specific numbers assigned for various types of services and applications, ensuring smooth communication. Additionally, there is a specific range of ports designated for general or specific use. For instance, ports in the range of 49152 to 65535 are often referred to as "Dynamic" or "Private" ports and can be used by dynamically generated applications. Therefore, it is essential to perform checks on internet service providers to assess data traffic conditions related to the offered services.

Load Balancing refers to the state of equilibrium in a network. A commonly used tool for distributing the management of data traffic and internet speed is Mikrotik [14]. The functions on Mikrotik devices can combine two internet sources from Internet Service Providers (ISPs) into one for collaborative operation [15].. The Load Balancing method, even with the combination of two internet sources, is not sufficient to enhance the network quality at Stream Cyber Cafe. There is a need for the management of data traffic paths based on the closest routes for access to user activities such as gaming, streaming services, social media, and others [16].

In data traffic, a Domain Name Server (DNS) serves as a transportation tool to reach the user's destination [17]. Numerous DNS services are available for free on the internet. It's crucial to consider the selection of DNS based on the internet source from the Internet Service Provider (ISP) being used. Each ISP has its own pathways for utilizing DNS [18]. Therefore, it is necessary to test the available DNS on the internet to determine a reliable path for internet users. The Dijkstra Algorithm is well-suited to the needs of Stream Cyber Cafe in determining the shortest path in a network when the primary route encounters issues [19]. Despite the existence of many similar algorithms, the Dijkstra Algorithm has proven to be effective in various routing contexts, particularly in the multitasking of data traffic.

2. RESEARCH METHOD

This research method employs a mixed-method approach involving experimental and case study approaches. The quantitative understanding of algorithm performance will be conducted through experimental methods, while the case study will complement qualitative understanding through field observations. The results obtained using the mixed-method approach can cross-check findings from both

qualitative and quantitative methods. The expectation is that by confirming results from two different methodologies, researchers and stakeholders can use the research findings as a basis for decision-making and other actions, ultimately eliminating ambiguity [20]. Figure 1 provides an explanation of the research flow utilized.
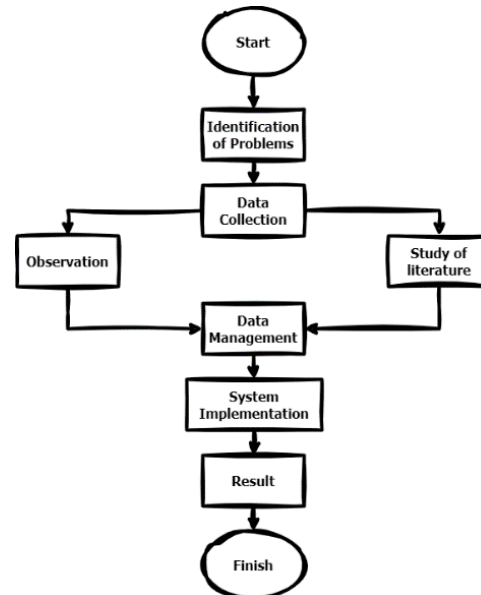

Figure 1. The research flow uses mixed methods

Load balancing is a technique used in computer networks to evenly distribute traffic or workload among multiple resources or paths, thereby preventing a specific resource or path from becoming a bottleneck or overloaded. The primary goal of load balancing is to enhance the performance, availability, and resilience of a system or network. By distributing the workload evenly, load balancing helps ensure that there is no single point of failure or excessive workload.

Algoritma Dijkstra is a shortest path search algorithm used in positively weighted graphs, such as network graphs. Below are the steps and formulas underlying the Dijkstra algorithm:

a) Initialization:
   Set the distance from the source to itself to 0.
b) Choose the Node with the Shortest Distance:
   Choose the node with the shortest distance from the source that has not been processed.
c) Relaxation:
   - Update the distance from the source to neighboring nodes if going through the currently processed node provides a shorter distance.
   - If a change occurs, update the distance and record the shortest path.
d) Mark Node as Processed:
   Mark the node as processed so that it won't be considered again.
e) Repeat:
   Repeat steps 2-4 until all nodes have been processed.

Formulas for Dijkstra's Algorithm:

a) Distance (d[v]): The distance from the source to a specific node, represented by d[v].
   - Initially, set each d[v] to infinity.
   - Set d[s] to 0, where s is the source node.

b) Weight or Distance between Two Nodes w(u,v): The weight of the edge connecting two nodes, denoted by w(u,v), which represents the weight of the edge connecting nodes u and v.

c) Relaxation Formula:
   If d[u] + w(u,v) < d[v], then update d[v] to d[u] + w(u,v).

d) Shortest Path:
   The shortest path from source s to node v can be reconstructed by following the recorded shortest path during the program execution.
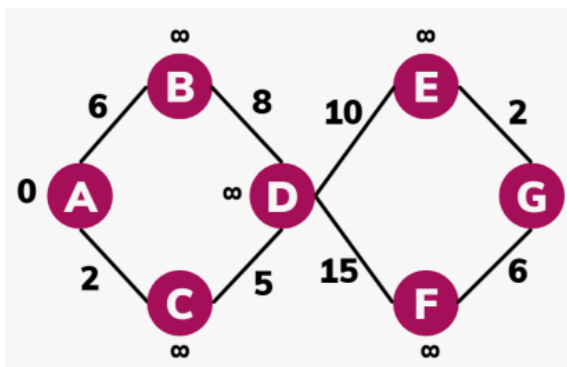

Figure 2. Example of Dijkstra's Algorithm Formula

Dijkstra's algorithm can be applied in various contexts, especially in computer networks and route optimization. In Figure 2, you can see the terminology of the Dijkstra Algorithm formula.
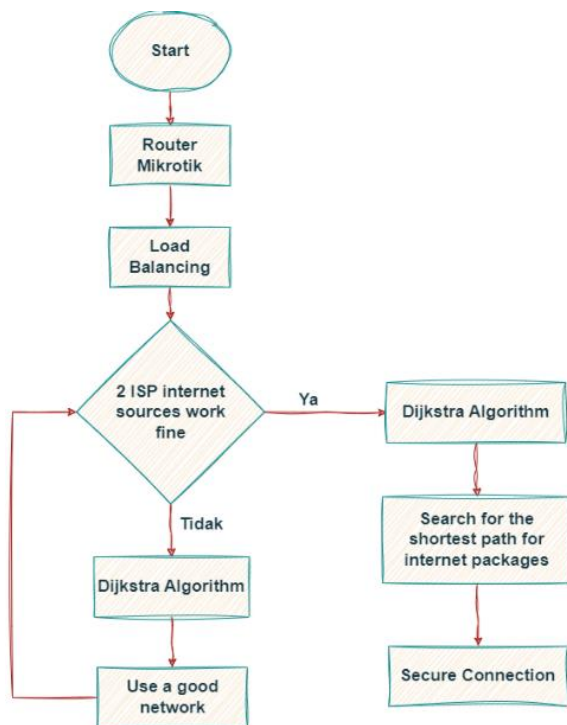

Figure. 3. Stages of Dijkstra Algorithm Research with Load Balancing

The flow depicted in Figure 3 utilizes Mikrotik Router devices due to their numerous features and instructions required in the Dijkstra Algorithm. Mikrotik devices excel in bandwidth management sourced from Internet Service Providers (ISPs) [21]. The role of the Dijkstra Algorithm in this Load Balancing method is to find the shortest path in network data traffic. The Dijkstra Algorithm adopts a Greedy approach by considering each minimum weight traversed and storing it in the corresponding set. The application of this algorithm, known as the Dijkstra algorithm, is often chosen to find the optimal path in the context of network routing. The Dijkstra algorithm's process involves calculating all vertices or points in the network to achieve satisfactory results [19]. The implementation of the Dijkstra algorithm demonstrates more efficient execution performance compared to other algorithms like the Ant Colony algorithm. Therefore, the Dijkstra algorithm is often preferred in the context of searching for the optimal path. The steps of the Dijkstra algorithm can be divided into the following stages [22]:

a) Identify the starting point, which will serve as the initial node or starting node. Next, set the distance weights from the starting node to each nearest node one by one. In this development process, the Dijkstra algorithm will iteratively search for the smallest weight from one node to another and to the next node.

b) Set the weight (distance) from each node to every other node. Initialize the value as 0 for the first node and as infinity for the other nodes.

c) Determine the status of all nodes, whether they have been traversed or not, and set the starting node as the "departure node."

d) At the departure node, calculate the distance from the departure node to every other untraversed node. Check if this distance is smaller than the previously calculated distance. If yes, update the previous distance data with the shorter distance.

e) After all calculations and assessments of distances to other nearby nodes are completed, mark the traversed nodes as "Traversed Nodes." Traversed nodes will not be revisited, and the stored distance is the one with the smallest weight.

f) Reset the status to "Nodes Not Traversed," with the smallest distance weight (from the departure node) as the "Departure Node" for the next departure. Repeat these steps to obtain the optimal path.

Therefore, this process enables the Dijkstra algorithm to systematically search and identify the shortest path from one node to another in a graph.

Internet packets are units of data that flow through data traffic via communication ports. These communication ports are categorized into TCP/IP, UDP, ICMP, and others. The selection of communication ports is based on the 7-layer protocol.

All internet data follows these 7 layers of protocol to facilitate data communication both on the internet and locally [23]
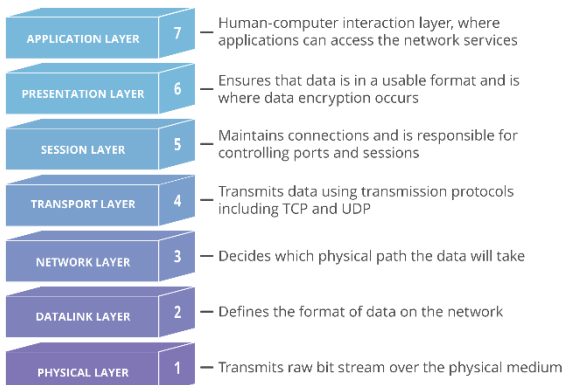

Figure 4. 7 OSI Layer Protocol

In general, data traffic passes through the 7-layer protocol. Figure 4 illustrates the functions and components of the 7-layer protocol [24]. Each layer has communication ports that operate within the framework of the internet gateway.

3. RESULTS AND DISCUSSION

The initial steps of this research involve implementing the Dijkstra Algorithm on Mikrotik devices with the aim of enhancing load balancing efficiency in the Stream Cyber Cafe network. This process includes configuring the Mikrotik device and utilizing features that support the logic of the Dijkstra Algorithm. The Mikrotik device used is RB750GR3 with RouterOS Version 7. Different internet sources from various Internet Service Providers are utilized. Each internet service provider is analyzed for the proximity of latency when connecting to Domain Name Server sources. The selected fastest Domain Name Servers currently used are as follows [25]:

Table 1. Fastest DNS List

Name	DNS Web Address	Internet Address (IP) DNS
Cloudflare	www.cloudflare.com	DNS: 1.1.1.1
		DNS: 1.0.0.1
Google Public DNS	www.google.com	DNS: 8.8.8.8
		DNS: 8.8.4.4
Quad9 DNS	www.quad9.net	DNS: 9.9.9.9
		DNS: 149.112.112.112
Open DNS	www.opendns.com	DNS: 208.67.222.222
		DNS: 208.67.220.220
Safe DNS	www. safedns.com	DNS: 195.46.39.39
		DNS: 195.46.39.40
DNS Filter	www.dnsfilter.com	DNS: 103.247.36.36
		DNS: 103.247.37.37
Next DNS	www.nextdns.io	DNS: 45.90.28.231
		DNS: 45.90.30.231
Comodo Secure DNS	www.comodo.com	DNS: 8.26.56.26
		DNS: 8.20.247.20

After obtaining the list of the fastest DNS servers from Table 1, the next step involves testing

each internet source to conduct a connection test with the aforementioned DNS servers. The internet sources in this study utilize products from Internet Service Providers ISN and Bizznet. The following image illustrates the experimental connections to determine which DNS is closer to the two types of Internet Service Provider (ISP) products.
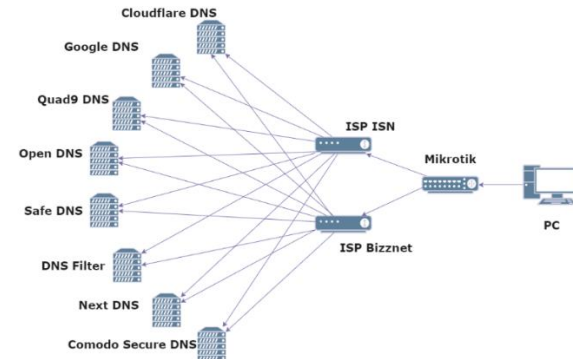

Figure 5. ISP connection topology with DNS List

Figure 5 illustrates the network topology of the service providers connecting to each of the fastest Domain Name Server (DNS) in the list. Subsequently, network testing is conducted using the "PING" command on each Internet Service Provider (ISP) route towards the aforementioned fastest DNS servers.

Table 2. Network Connection Test Results Against Internet Service Provider

Name	Internet Address (IP) DNS	ISP Name	
		ISN	Biznet
Cloudflare	DNS: 1.1.1.1	12ms	24 ms
	DNS: 1.0.0.1	15 ms	20 ms
Google Public DNS	DNS: 8.8.8.8	20 ms	25 ms
	DNS: 8.8.4.4	22 ms	26 ms
Quad9 DNS	DNS: 9.9.9.9	25 ms	18 ms
	DNS: 149.112.112.112	26 ms	20 ms
Open DNS	DNS: 208.67.222.222	23 ms	20 ms
	DNS: 208.67.220.220	25 ms	21 ms
Safe DNS	DNS: 195.46.39.39	26 ms	28 ms
	DNS: 195.46.39.40	28 ms	25 ms
DNS Filter	DNS: 103.247.36.36	18 ms	22 ms
	DNS: 103.247.37.37	20 ms	28 ms
Next DNS	DNS: 45.90.28.231	26 ms	22 ms
	DNS: 45.90.30.231	28 ms	24 ms
Comodo Secure DNS	DNS: 8.26.56.26	28 ms	30 ms
	DNS: 8.20.247.20	26 ms	32 ms

Table 2 explains the distance conditions of each DNS to the used internet source. Before applying the Dijkstra's Algorithm, the Internet Service Provider randomly selects the path it uses and also randomly chooses the time to select the path, causing discomfort such as packet loss and network disruptions for its users. As a result, the internet connection distance in accessing services takes almost 100 ms.
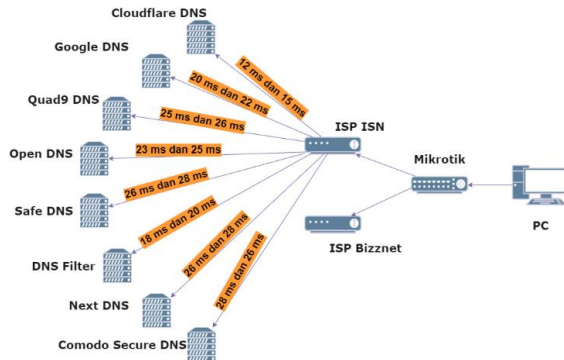
Figure 6. Product Internet Service Provider ISN Network Connection Trial Results with DNS List
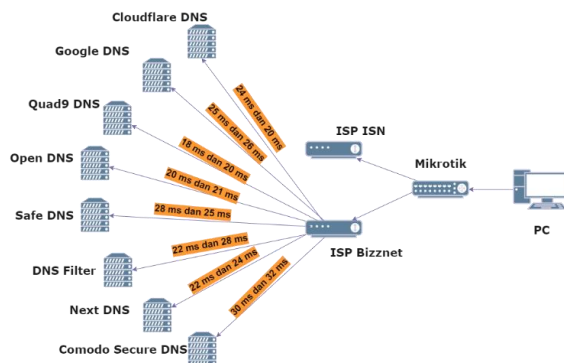

Figure 7. Product Internet Service Provider Bizznet Network Connection Trial Results with DNS List

From the network testing in Table 2 above, we can determine that for the Internet Service Provider ISN, a good DNS network connection is achieved with Cloudflare DNS, as shown in Figure 6. Meanwhile, for the Internet Service Provider Bizznet, a good DNS network connection is obtained from Quad9 DNS, as shown in Figure 7. Here is an overview before using the Dijkstra's Algorithm when accessing the gaming service with a game called Valorant.

Figure 8. Network Connection Results Before Using Dijkstra's Algorithm for Gaming Service to Valorant Server

In Figure 8, with a ping test result of 104 ms for the Valorant gaming network service to the Asia-Pacific server, it will undoubtedly lead to discomfort such as lag during gameplay. This becomes a reason for users to switch to another location for a more comfortable gaming experience. Therefore, there is a need for a network rule to determine the shortest distance for network traffic, making it more optimal.

Next, the implementation of features on Mikrotik to achieve the functionality of the Dijkstra Algorithm. The following are the command instructions on the Mikrotik device to implement features that support this algorithm. The Mikrotik features used include functions from the Firewall Mangle, Routing Table, and Routing Gateway features. Prior to this, the configuration on Mikrotik is set up as usual to ensure that the Internet Service Providers are recognized and connected to the Mikrotik device.

a) Command the Mangle Firewall Feature.

Table 3. List of Command Firewall Mangle on Mikrotik

```
/ip firewall mangle
    add action=accept chain=prerouting comment=Accept
    dst-address-list=internet
    add action=accept chain=prerouting src-address-
    list=internet
    add action=mark-connection chain=input comment=input
    in-interface=pppoe-isn new-connection-mark=line1
    passthrough=yes
    add action=mark-connection chain=input in-
    interface=ether2 new-connection-mark=line2
    passthrough=yes
    add action=mark-connection chain=prerouting
    comment=pcc dst-address-type=!local in-interface=client
    new-connection-mark=line1 passthrough=yes per-
    connection-classifier=both-addresses-and-ports:4/0
    add action=mark-connection chain=prerouting dst-
    address-type=!local in-interface=client new-connection-
    mark=line1 passthrough=yes per-connection-
    classifier=both-addresses-and-ports:4/1
    add action=mark-connection chain=prerouting dst-
    address-type=!local in-interface=client new-connection-
    mark=line2 passthrough=yes per-connection-
    classifier=both-addresses-and-ports:4/2
    add action=mark-connection chain=prerouting dst-
    address-type=!local in-interface=client new-connection-
    mark=line2 passthrough=yes per-connection-
    classifier=both-addresses-and-ports:4/3
    add action=mark-routing chain=prerouting
    comment="mark routing" connection-mark=line1 in-
    interface=client new-routing-mark=line1
    passthrough=yes
    add action=mark-routing chain=prerouting connection-
    mark=line2 in-interface=client new-routing-mark=line2
    passthrough=yes
    add action=mark-routing chain=output comment=output
    connection-mark=line1 new-routing-mark=line1
    passthrough=yes
    add action=mark-routing chain=output connection-
    mark=line2 new-routing-mark=line2 passthrough=yes
```

b) Command the Routing Table Feature

Table 4. List of Command Routing Table on Mikrotik

```
/routing table
    add disabled=no fib name=line1
    add disabled=no fib name=line2
```

c) Command the Routing Gateway Feature

Table 5. List of Command Routing Gateay on Mikrotik

```
/ip route
    add check-gateway=ping disabled=no distance=1 dst-
    address=0.0.0.0/0 gateway=9.9.9.9 routing-table=main
    scope=30 suppress-hw-offload=no target-scope=11
    add check-gateway=ping disabled=no distance=1 dst-
    address=0.0.0.0/0 gateway=1.1.1.1 routing-table=main
    scope=30 suppress-hw-offload=no target-scope=11
```

```
add check-gateway=ping disabled=no distance=1 dst-
address=0.0.0.0/0 gateway=103.109.125.255 pref-src=""
routing-table=line1 scope=30 suppress-hw-offload=no
target-scope=10
add check-gateway=ping disabled=no distance=2 dst-
address=0.0.0.0/0 gateway=192.168.18.1 pref-src=""
routing-table=line2 scope=30 suppress-hw-offload=no
target-scope=10
add disabled=no distance=1 dst-address=1.1.1.1/32
gateway=103.109.125.255 pref-src="" routing-
table=main scope=10 suppress-hw-offload=no target-
scope=10
add disabled=no distance=1 dst-address=9.9.9.9/32
gateway=192.168.18.1 pref-src="" routing-table=line1
scope=10 suppress-hw-offload=no target-scope=10
```
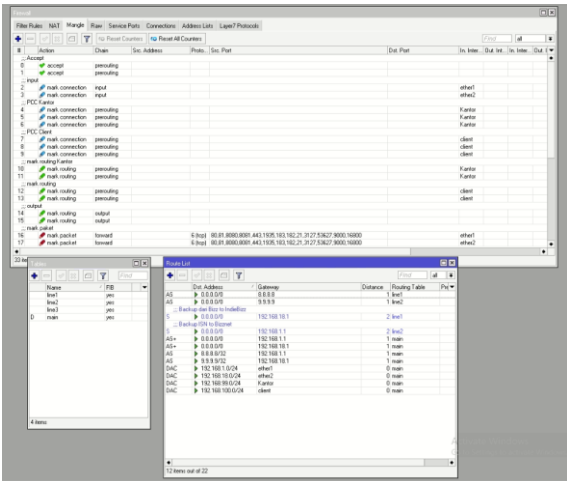

Figure 9. Results Description of Mikrotik Functions

The Firewall Mangle feature in Table 3 is used to indicate the conditions for the rules set by the instructions of the Dijkstra Algorithm. The Routing Table feature in Table 4 is utilized to mark and connect the internet sources from the Internet Service Providers so that they can be redirected to the specified paths. The Routing Gateway feature in Table 5 acts as the gateway terminal for the specified Domain Name Servers on each internet source. The representation in Figure 9 illustrates the results of the command instructions on the Mikrotik device when executed. Testing the connection from the Mikrotik device to the list of Domain Name Servers (DNS) is done through the Windows Command Prompt (CMD) using the "tracert" command [26]. The results displayed in the Command Prompt (CMD) tool indicate the number of hops taken through the specified list of Domain Name Servers to each internet source from the Internet Service Provider (ISP). The screen view in Figure 10 shows the user interface after implementing features that fulfill the workings of the Dijkstra Algorithm.

The Dijkstra Algorithm is capable of addressing network issues that encounter disruptions in the Load Balancing method due to path disturbances. Its greedy nature serves as a solution for finding short-path decisions, mitigating disruptions on specific paths from lasting too long. In Figure 11, the flow in the Stream Cyber Cafe network utilizing the Dijkstra

Algorithm for multitasking in determining the shortest paths on Mikrotik is depicted.
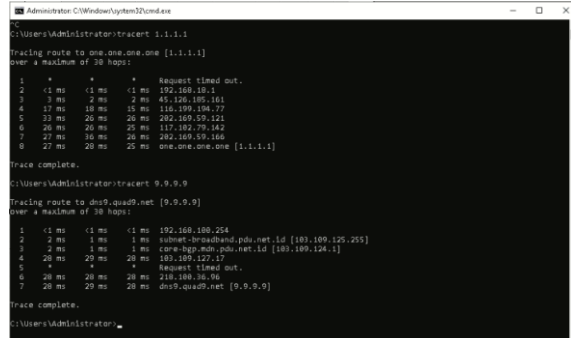

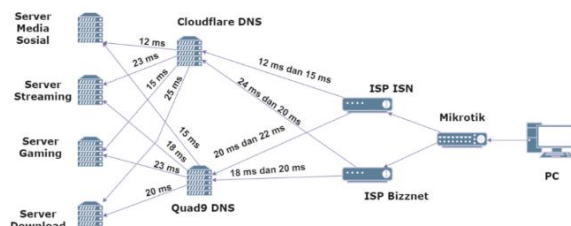Figure 10. Results of Using the Dijkstra Algorithm on Mikrotik

Figure 11. Mikrotik Dijsktra Algorithm Flow

In Figure 11, it can be observed that by using the Dijkstra's Algorithm, the internet connection path can be automatically traversed through the shortest route. For example, if a user wants to engage in gaming services with the destination being a gaming server, the flow is through the ISP ISN, then enters the DNS using the shortest time of 12 ms and 15 ms. Next, from the DNS to the gaming server service takes 15 ms, resulting in a total time of approximately 27 ms required to perform gaming services. This is a significant improvement from the previous 100 ms. The Dijkstra's Algorithm successfully enhances the efficiency of network traffic. The topology can be seen in Figure 12.
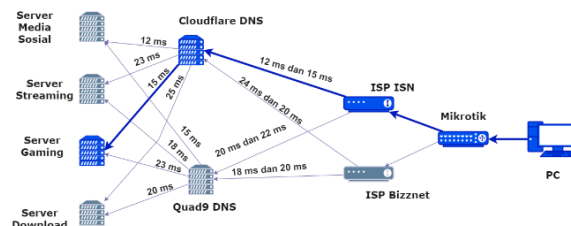

Figure 12. Shortest Path Dijkstra Algorithm Gaming Server Destination

Experimenting with the results of the Dijkstra's Algorithm for the Valorant server, which has been previously tested for network connectivity in Figure 8. Next, conducting an experiment with the Valorant Game Server using the Dijkstra's Algorithm.

Figure 13 shows that the Dijkstra's Algorithm selects the Valorant Game Server point with the shortest path, ensuring users are comfortable with a connectivity route that is close to the gaming server.

Figure 13. Results of Network Traffic Experiment to Valorant
Game Server with Dijkstra's Algorithm

In a study by Ika and Pristi titled "Implementasi Algoritma Dijkstra untuk Menentukan Rute Terpendek Menuju Pelayanan Kesehatan" [27] .it was concluded that the Dijkstra's Algorithm is capable of determining the shortest path based on the smallest weight between points gradually. The same principle is applied in Mikrotik Routing networks. Internet traffic can have its path determined through DNS to find the shortest route from one server point to another traffic data server, making the network services at Stream Cyber Cafe more optimal in selecting a secure connection path.

4.   DISCUSSION

Examining the execution of the Dijkstra algorithm on MikroTik for determining the shortest path, this research focuses on the use of DNS as a determining parameter. The first step involves identifying the starting point, namely the first node or initial node, which in this case is associated with the source DNS server. Configuring MikroTik, specifically the RB750GR3 device with RouterOS Version 7, is the initial step in mapping the network topology.

Next, the weight of the distance from the initial node to each nearest node (DNS server) is set one by one. This process includes analyzing the latency of connections to each DNS server from various Internet Service Providers (ISPs), such as ISN and Bizznet. Since the basic concept of the Dijkstra algorithm is to find the shortest path, the time required to reach each DNS server from two different ISPs becomes a determining factor. This result can be seen in Table 2, which shows the latency distance values in milliseconds (ms) between Internet Service Providers (ISPs) and DNS.

At this stage, the Dijkstra algorithm applies a Greedy approach, considering each minimum weight passed and storing it in the appropriate set. As seen in Figure 8, the testing is performed by accessing a gaming server service, namely Game Valorant, before applying the Dijkstra Algorithm, and in Figure 13, with the Dijkstra Algorithm applied, the configuration implemented on the MikroTik device will seek the shortest path taken by network traffic. The use of the "PING" command from each internet service provider to the fastest DNS list provides

information about the number of hops or steps needed to reach each DNS server.

The test results in Figure 11 show that each ISP has a different shortest path to the DNS server, and the time required to reach these DNS servers varies. In Figure 12, the implementation of the Dijkstra algorithm on MikroTik illustrates its ability to determine the shortest path by considering the time or latency factor.

The next configuration process involves the implementation of MikroTik features, such as Firewall Mangle, Routing Table, and Routing Gateway, as seen in Tables 3, 4, and 5, to support the functions of the Dijkstra algorithm. In Figure 10, further testing using the "tracert" command from MikroTik to the fastest DNS list visualizes the path taken through each internet service provider.

Overall, the use of the Dijkstra algorithm on MikroTik in the context of determining the shortest path using DNS servers from various types of internet service providers provides an insight into the effectiveness of this algorithm in handling network dynamics and time variability. This implementation opens up the potential for improving load balancing efficiency in internet cafes by leveraging the adaptive and responsive approach of the Dijkstra algorithm.

5.   CONCLUSION

The use of internet networks is crucial in the activities of modern users. Besides merely engaging in activities, there are often disruptions in its utilization. It is essential to note that in internet usage, activities such as gaming, streaming services, and social media are a set of processes that run conclusively. The smooth transmission of these activities requires adherence to the 7-layer protocol. Additionally, the use of Domain Name Servers (DNS) as a means of transporting internet packets also needs attention. This is grounded in the selection of internet source products from Internet Service Providers. The improvement in load balancing network performance in internet cafes using the Dijkstra Algorithm on MikroTik devices has yielded significant positive impacts. By cutting the distance of network traffic which initially took 100ms to the service destination, now with the Dijkstra Algorithm it only takes around 27ms to the user's service destination. Performance evaluations of load balancing indicate enhanced optimal load distribution, improved resource management efficiency, and reduced downtime risks. The reliability of services in the Stream Cyber Cafe network has increased, providing users with a more responsive and consistent streaming experience. Despite facing challenges during the implementation phase, the applied solutions successfully addressed these constraints. The results of this research provide a robust foundation for further understanding the practical application of the Dijkstra Algorithm in the context of load balancing. It also sets the stage for

advanced research and the development of more sophisticated solutions in the future. Thus, the implementation of the Dijkstra algorithm on MikroTik emerges as an effective approach to enhance the performance and reliability of networks in dynamic Stream Cyber Cafe environments. Furthermore, the distribution of packets based on cache obtained from web browsers needs attention to assist and influence the sustained performance load of the internet.

DAFTAR PUSTAKA

[1]   F. Al Faribi, M. Ulfa Batubara, and U. Dharmawangsa, "Penggunaan Internet Sebagai Media Komunikasi Dalam Meningkatkan Pelayanan Hukum Online Di Kantor Kejaksaan Negeri Bangka Selatan," *Jurnalsocialopinion*, vol. 7, pp. 1–8, 2022, [Online]. Available: https://jurnal.darmaagung.ac.id/index.php/socialopinion/article/view/1479/1316

[2]   M. Hasbi and N. R. Saputra, "Analisis Quality of Service (Qos) Jaringan Internet Kantor Pusat King Bukopin Dengan Menggunakan Wireshark," *Univ. Muhammadiyah Jakarta*, vol. 12, no. 1, pp. 1–7, 2021, [Online]. Available: https://jurnal.umj.ac.id/index.php/just-it/article/view/13596/7236

[3]   G. Adhiyasa, P. Nugraha, I. M. Agus, D. Suarjaya, I. P. Agus, and E. Pratama, "Analisis Tren Lalu-lintas Data Jaringan Menggunakan Teknologi Big Data (Studi Kasus: UNIVERSITAS MAHADEWA INDONESIA)," *JITTER-Jurnal Ilm. Teknol. dan Komput.*, vol. 3, no. 2, 2022.

[4]   Y. Ilham and I. M. B. Dirgantara, "Analisis Pengaruh Kualitas Jaringan, Kualitas Layanan, Kualitas Informasi, Keamanan dan Privasi pada Penyedia Layanan Internet terhadap Kepuasan Pelanggan dan Dampak pada Niat Pembelian Ulang," *Diponegoro Journal of Management*, vol. 9, no. 4. pp. 1–7, 2020. [Online]. Available: https://ejournal3.undip.ac.id/index.php/djom/article/view/31080

[5]   Dwi Nurul Huda and M. T. Margianto, "Sistem Pendukung Keputusan Pada Aplikasi Penyedia Layanan Internet Terbaik Menggunakan Metode Multi Factor Evaluation Process," *J. Bangkit Indones.*, vol. 11, no. 1, pp. 30–39, 2022, doi: 10.52771/bangkitindonesia.v11i1.208.

[6]   D. Kartika and Y. Mardiana, "Dns Server And Web Server Simulation With Debian Operating System On Local Area Network Simulasi Dns Server Dan Web Server Dengan Sistem Operasi Debian Pada Jaringan Local Area Network," *J. Media*

Comput. Sci., vol. 2, no. 1, pp. 83–92, 2023.

[7]   A. Assagaf *et al.*, "Pengujian Koneksi Jaringan Wireless Poin To Poin Untuk Mengimplementasikan Opensid di Desa Tawabi Kabupaten Halmahera Selatan," *J. Pengabdi. Masy.*, vol. 2, pp. 10–15, 2020.

[8]   A. Muis and S. Pitra, "Peranan Internet Sebagai Sumber Belajar Dalam Meningkatkan Minat Belajar Pendidikan Agama Islam Peserta Didik Kelas XI Di Sma Muhammadiyah Parepare," *J. Al-Ibrah*, vol. 10, no. 1, pp. 189–222, 2021.

[9]   R. Setiawan, "KETERLEKATAN INTERNET DALAM AKTIVITAS KESEHARIAN DAN PENDIDIKAN GENERASI MILENIAL," *J. Sosioteknologi ITB*, vol. Volume 20, no. April, pp. 66–79, 2021.

[10]   S. Prahara and I. Ali, "Optimalisasi Jaringan Internet Dengan Optimalisasi Load Balancing Menggunakan Parameter Qos (Studi Kasus : SMK Bina Warga Lemahabang)," *JATI - J. Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 211–217, 2023.

[11]   R. Duyo, "Analisis Penyebab Gangguan Jaringan pada Distribusi Listrik Menggunakan Metode Fault Tree Analysis," *J. Tek. Elektro UNISMUH*, vol. 12, no. 2, pp. 1–12, 2020, [Online]. Available: https://journal.unismuh.ac.id/index.php/vertex/article/view/4017

[12]   M. Fajri, R. Munadi, and T. Y. Arif, "Manajemen Bandwidth Pada Jaringan Lokal Menggunakan Sistem Operasi Vyos," *J. Komputer, Inf. Teknol. dan Elektro*, vol. 5, no. 2, pp. 1–5, 2020, doi: 10.24815/kitektro.v5i2.15542.

[13]   W. Farhan Fatoni, A. Hidayat, P. Studi Ilmu Komputer, and F. Ilmu Komputer, "Implementasi Sistem Keamanan Jaringan Komputer Dengan Metode Port Knocking Pada Lkp Surya Komputer," *J. Mhs. Ilmu Komput.*, vol. 03, no. 01, 2022.

[14]   Y. Yuliadi, R. Rodianto, M. Julkarnain, E. S. Susanto, N. S. Bahtiar, and F. D. Ikram, "Pembuatan Jaringan Komputer Berbasis Mikrotik," *J. Pengabdi. Rekayasa Sist.*, vol. 1, no. 2, pp. 21–24, 2023.

[15]   N. Nurmiati, L. Surimi, and S. Subardin, "Analisis Kinerja Load Balancing Terhadap Jaringan Internet Menggunakan Metode Equal Cost Multi Path (ECMP)," *Digit. Transform. Technol.*, vol. 2, no. 2, pp. 52–62, 2022, doi: 10.47709/digitech.v2i2.1779.

[16]   S. Ditama, "Analisis Kemacetan Lalu Lintas Data & Broadcast Domain Pada Jaringan Komputerdi Lingkungan Stkip Pgri Pacitan," *J. Penelit. Pendidik.*, vol. 11, no. 2, pp. 1615–

1666, 2019.

[17] Z. M. Subekti, H. Setiawan, S. Satria, W. M. Wijaya, A. Hafiz, and W. Warsudi, "Perancangan Infrastruktur Domain Name Server Lokal Menggunakan Ubuntu Server 16.04 Pada Pt. Xyz," *J. Inf. dan Komput.*, vol. 8, no. 2, p. 6, 2020.

[18] I. Mayasari, "Analisis Performansi Layanan Internet Dengan Menggunakan Teknologi Broadband Over Powerline (Studi Kasus : Icon + Bandung) Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)," *J. Ilm. Komput. dan Inform.*, 2020.

[19] S. Sunardi, A. Yudhana, and A. A. Kadim, "Implementasi Algoritma Dijkstra Untuk Analisis Rute Transportasi Umum Transjogja Berbasis Android," *J. Sist. Inf. Bisnis*, vol. 9, no. 1, p. 32, 2019, doi: 10.21456/vol9iss1pp32-38.

[20] J. H. Yam, "Refleksi Penelitian Metode Campuran (Mixed Method)," *J. Emp.*, vol. 2, no. 2, pp. 126–134, 2022, doi: 10.33592/empire.v2i2.3310.

[21] H. Heromadhani, S. Sudarmaji, and A. Hidayat, "Pengembangan Jaringan Komputer Berbasis Mikrotik Pada Smp Negeri 8 Metro," *J. Mhs. Ilmu Komput.*, vol. 2, no. 2, pp. 212–219, 2021, doi: 10.24127/ilmukomputer.v2i2.1671.

[22] A. Cantona, F. Fauziah, and W. Winarsih, "Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta," *J. Teknol. dan Manaj. Inform.*, vol. 6, no. 1, pp. 27–34, 2020, doi: 10.26905/jtmi.v6i1.3837.

[23] I. F. Anshori, "Implementasi Socket Tcp/Ip Untuk Mengirim Dan Memasukan File Text Kedalam Database," *Responsif*, vol. Vol 1 No 1, no. 1, pp. 1–5, 2019.

[24] "What is layer 7? | How layer 7 of the Internet works | Cloudflare." Accessed: Dec. 14, 2023. [Online]. Available: https://www.cloudflare.com/learning/ddos/what-is-layer-7/

[25] "Rekomendasi DNS Tercepat & Terupdate tahun 2023 – Pusat Teknologi Informasi dan Pangkalan Data." Accessed: Dec. 14, 2023. [Online]. Available: https://ptipd.syekhnurjati.ac.id/rekomendasi-dns-tercepat-terupdate-tahun-2023/

[26] J. Meybie, J. Triyono, and U. Lestari, "Analisis Kualitas Jaringan Wlan Berdasarkan Jarak," *J. JARKOM*, vol. 7, no. 2, pp. 140–149, 2019.

[27] I. Arthalia Wulandari and P. Sukmasetyan, "Implementasi Algoritma Dijkstra untuk Menentukan Rute Terpendek Menuju Pelayanan Kesehatan," *J. Ilm. Sist. Inf.*, vol. 1, no. 1, pp. 30–37, 2022, doi: 10.24127/jisi.v1i1.1953.