SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 1, Issue 1, August 2023, Page 10-22
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v1i1.59

Analysis of Distributed File System Replication Using the NDLC Method with Hyper-V Virtual Simulation Machine

Ucu Sumirat[1], Antonius Darma Setiawan[2*], Sinka Wilyanti[3], Rosyid Ridlo Al-Hakim[4]

Department of Electrical Engineering, Faculty of Engineering and Computer Science, Jakarta Global University, Depok, Indonesia

Email: ucu.sumirat@gmail.com[1], a.darma.setiawan@gmail.com[2], sinka@jgu.ac.id[3], rosyidridlo@student.jgu.ac.id[4]

Orcid: https://orcid.org/0009-0002-7204-3547[1], https://orcid.org/0000-0002-6042-8485[2], https://orcid.org/0000-0002-7089-0323[3], https://orcid.org/0000-0003-1502-4745[4]

Abstract. The need to access file sharing easily on an organization's computer network was increased. Users didn't have to worry about the number of file server addresses that can be accessed and are made with only one access address to use file sharing. The availability of data on the network to file storage availability in an organization was also essential. Data would be permanently lost, following a reason including hardware failure, or even accidentally deleted. It was important to ensure that there was a copy of the data. Achieving good data availability requires a system strategy built in the organization's data center. This research used Distributed File System Replication (DFSR) based on active directory domain services with Windows Server. The research method used NDLC (Network Diagram Life Cycle) method. This research was conducted through analysis with the Hyper-V virtual simulation machine. The results of the research with this simulation are that the Distributed File System (DFS) makes it easy for users to access file shares on several file server nodes using only one URL address. DFSR makes it easy for users to clone files automatically on multiple nodes file servers at other locations. DFSR, with its *Share and Publish* features, provide good data availability. If one of the file server nodes experiences an interruption, the file server nodes at another location would be taken over to provide the data. This system makes it easy for administrators to manage file servers.

Keywords: Hyper-V, distributed file system replicated, distributed file, file sharing, high availability.

1 Introduction

Users need easy file share access; if possible, users don't mess around with the number of file server addresses that can be accessed and are made with only one access address to be able to use file sharing. The availability of data on the network to the availability of file storage in an organization is also very important for users. Data can be lost permanently if the computer is lost, following a hardware failure, or even if it is accidentally deleted. For that reason, it is important to ensure that there is a copy of the data [1]. There are several strategies for achieving this, depending on the type and amount of data. Whenever possible, data should not be stored on personal computers. Several centralized options are available, including a client-server document management system.

Data availability is the availability of data on a file server that all users can access and can be obtained at any time when needed. Data availability is intended as a data readiness where the data responds directly when the user requires the data. Data availability is a condition in which the user can access a given resource. Availability of data is often closely related to performance. If the data files on one server are offline, performance is non-existent because no data can be accessed [2].

Achieving a good level of data availability requires a system strategy built in the organization's data center. One of the technologies that can be implemented is Distributed File System Replication (DFSR) based on active directory domain services. This system makes *namespaces* of DFS (Distributed File System) service, i.e., a role service in Windows Server that allows us to group shared folders located on different Servers into one or more logically structured namespaces. Besides, makes it possible to give users a virtual view of shared folders, where one path leads to files located on multiple Servers. DFS replication plays an important role in data availability. DFS replication is a role service in Windows Server that allows us to efficiently replicate folders (including

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 1, Issue 1, August 2023, Page 10-22
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v1i1.59

those referred to by DFS service namespace paths) across multiple Servers and sites. DFS Replication is an efficient multi-master replication engine that we can use to keep folders in sync between Servers across network connections with limited bandwidth [3].

We tried to use the Hyper-V virtual machine as a simulation tool to assist this research method. Hyper-V is Microsoft's hardware virtualization product, which can run versions of computer software known as virtual machines. Each virtual machine (VM) acts like a complete computer, running an operating system and programs [4]. In addition, this study tried to analyze the simulation of a distributed data replication system (DFSR) based on an active directory on a Windows Server to provide solutions to user needs for high availability data on file sharing access.

2 Method

The research method used is qualitative research with the NDLC (Network Diagram Life Cycle) method (Figure 1) to find factual information and identify existing problems. Qualitative research is research that is descriptive and tends to use analysis. The theoretical foundation is used to guide the research focus following the facts. The result is that qualitative research is carried out with a research design in which the findings are not obtained through statistical procedures or calculations [5]. Meanwhile, several studies used the NDLC method to design the physical security system for data centers [6], logical unit number (LUN) simulation on the virtual storage [7], quality of service (QoS) of WLAN network design [8], fiber optic networking infrastructure design [9], computer network simulation [10], [11], and network security infrastructure cabling design [12].

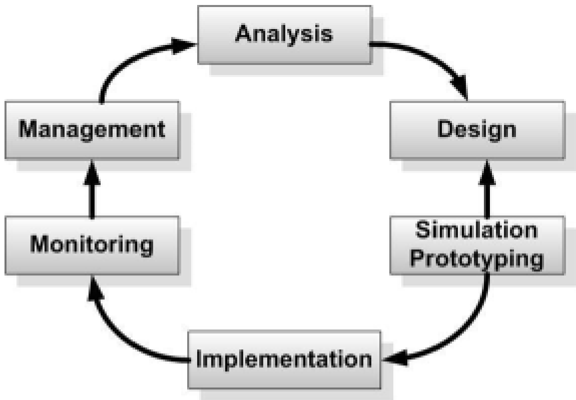

Figure 1. Flowchart of NDLC (Network Diagram Life Cycle) method. Source: Warman & Andrian [13].

2.1 Analysis

Before submitting a production system, we would examine the system built through a simulation by creating a virtual server. In this case, we would describe the hardware requirements for the DFSR simulation as shown in Table 1.

Table 1. Hardware requirements in this study.

Specification	Detail
CPU	8 x Intel Xeon Silver 4114
RAM	32 GB
SSD	250 GB
PSU	750 Watt 80 Plus Platinum
Internet	1 GB/s

The PC requirements would be installed Hyper-V Manager Software Hypervisor type 2 (two), namely one of the virtual machine (VM) software which could be installed for system requirements such as virtual servers and virtual networks. In this case, we would simulate a DFSR (distribution file system replication) replication system using a domain base on Windows Server. Making the simulation requires several virtual servers described in Table 2.

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 1, Issue 1, August 2023, Page 10-22
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v1i1.59

Table 2. Virtual server simulation requirements in this study.

Server Requirement	Operating System	Virtual Machine Specification
VMs Primary Domain Controllers	Windows Server 2022	4 GB RAM, 30 GB HDD, 1 GB/s Internet
VMs File Server 1 (FS01)	Windows Server 2022	4 GB RAM, 30 GB HDD, 1 GB/s Internet
VMs File Server 2 (FS02)	Windows Server 2022	4 GB RAM, 30 GB HDD, 1 GB/s Internet
VMs PC Client (CL01)	Windows 10 Professional	2 GB RAM, 30 GB HDD, 1 GB/s Internet
VMs PC Monitoring	Debian Linux	2 GB RAM, 30 GB HDD, 1 GB/s Internet

2.2 Design

a. Type-2 Hypervisor Simulation Design

Referring to the analysis of the needs of the virtual server that would be built for the simulation design using the Hyper-V Manager virtualization software can be seen in Figure 2.

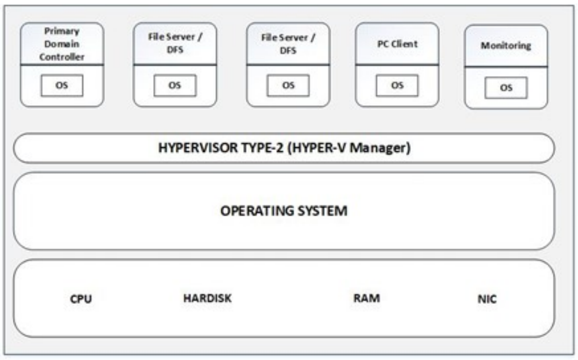

Figure 2. The architecture of a type-2 hypervisor simulation.

b. Distribution File System (DFS) Design

File sharing with a distributed file system (DFS) can be replicated across multiple file servers in different locations to optimize server load and increase the speed of access to shared files. In this case, the user can access the file share on the server closest to them. DFS is intended to simplify access to shared files, as shown in Figure 3.

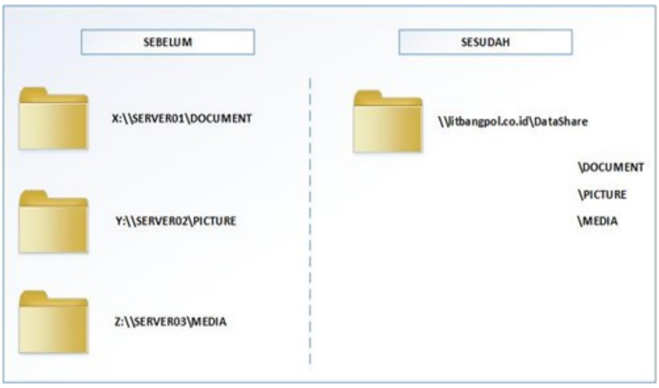

Figure 3. DFS namespace settings.

DFS namespaces is a virtual folder that contains links to shared folders stored on different file servers. DFS namespaces can be standalone or domain based. Standalone DFS namespaces are located on only one server and are not fault tolerant. If the Server root is unavailable, the entire DFS namespace is unavailable. We can use this option if we don't have an active directory domain configured (when using Workgroup). From the analysis of the running system, the existing system has built a domain controller server. So, in this case, simulation research would be use domain-based. Domain-based DFS namespaces store configurations in an active directory. The path to access the root namespace starts with the domain name. We can store domain-based DFS namespaces on multiple servers to increase namespace availability. This approach allows us to provide fault tolerance and load balancing across servers, as shown in Figure 4.

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 1, Issue 1, August 2023, Page 10-22
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v1i1.59

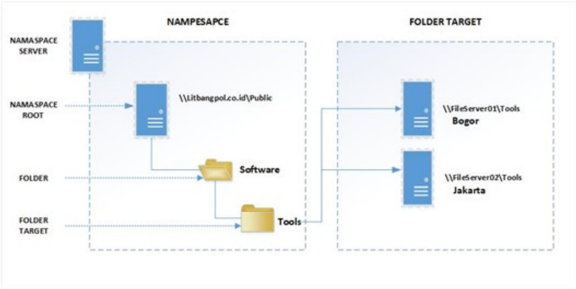

Figure 4. DFS namespace for the domain-based settings.

The following criteria should be set in domain-based DFS namespace requirements addressed:

1) *The server namespace* is the physical server (or VM) holding the DFS namespace. A namespace server can be a regular Server with the DFS role installed or a domain controller.
2) *Folders* is a link in the DFS namespace that points to the target folder containing the content for user access.
3) *A target folder* is a link to a shared file resource located on a specific file server and available through a UNC (Universal Naming Convention) path.

Meanwhile, The DFS tree structure includes the following components addressed as follow:

1) *DFS Root* is the DFS Server on which the DFS service runs.
2) *DFS Link* is a link pointing to the network share used in DFS.
3) *DFS Targets* is the real network portion to which the DFS link belongs.

c. Distribution File System (DFS) Replication Design

DFS replication is a feature used to duplicate existing data by replicating copies of the data to several locations. Physical file shares can be synchronized in two or more locations. An important feature of replication DFS is that file replication is by design only after the file is closed. For this reason, DFS replication is unsuitable for replicating databases because the database has files that are opened during the operation of the database management system. DFS replication supports multi-master replication technology, and any replication group member can modify data that is then replicated. Replication group DFS is a Server group that participates in replicating one or more replication folders. Replicated folders are synchronized between all members of the replication group. The following is a group replication diagram, which can be seen in Figure 5.

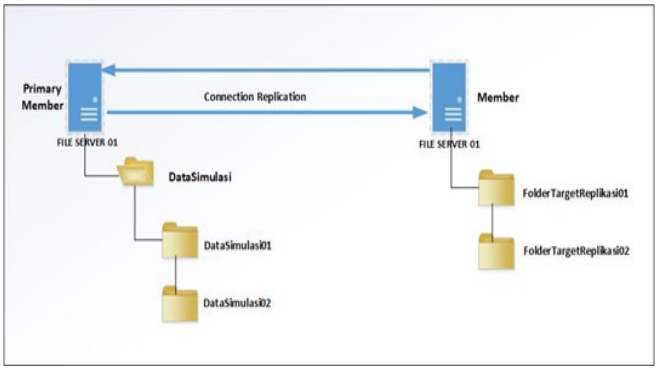

Figure 5. DFS replication group design.

DFS Replication uses a special Remote Differential Compression algorithm that allows DFS to detect changes and copy only changed blocks of files. This approach saves time and reduces replication traffic through the network. It was done asynchronously. There may be a delay between writing changes to the source location and replicating those changes to the target location. There are two main DFS replication topologies, such as:

1) *Hub and Spoke,* i.e., this topology requires at least three replication members: one to act as a hub and the other two to act as spokes. This technique is useful if we have a central data source (hubs) and must replicate this data to multiple locations (radios).
2) *Full Mesh,* i.e., each replication group member replicates data to every group member. Use this technique if at least 10 or fewer members are in the replication group. In this case, the author would use *Full Mesh* topology, selected based on the needs in the field.

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 1, Issue 1, August 2023, Page 10-22
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v1i1.59

2.3 Simulation Implementation

In this study, we build and analyze the simulation engine and need to rebuild the Active Directory Domain Controller System (ADDCS) and build the required environment, such as VM Domain Controller, VM File Server, and VM PC Client. Besides, the simulation process can analyze the DFS Replication system without disturbing the running system and the extent to which this system can be an option as one of the systems that can anticipate data availability. First, set up the PC Workstation to prepare the simulation procedures. Then, after the hardware meets the requirements, should be installed Windows 10 Professional as a host for the type-2 hypervisor. After that, by default on Windows 10 Professional, the virtualization feature is inactive so that the author will install or activate the Hyper-V manager feature for the next stage. Then, create a virtual server environment that would be created on the Hyper-V Manager installed in the previous stage. The VM names are "DC01" as a domain controller server, "FS01" as the first file share server, "FS02" as the second file share server, and "Client01" as a PC that is used to access the file or document sharing.

2.4 Monitoring

To analyze log results from file replication and log test results from high availability file sharing, we create a simple monitoring system script by adding a virtual machine installed with the Linux Debian operating system. It is used to execute the Python programming script and the IO socket library to monitor and analyze replication success and data availability in real time.

2.5 Management

Windows Server active directory logically manages access rights to the DFS replication process. In more detail, the following is the management carried out by Windows Server to run the user creation process and access files:

1) *Management delegation of access rights,* we build a dynamic directory simulation designed to use a domain tree because it uses one domain. The management is divided into several parts, from creating an Organizational Unit (OU), Group Users (GUs), and creating Users. Within the Organizations Unit (OU), a group contains users who would be allowed to access the folders prepared for file sharing. The access permits are described in Table 3. Based on Table 3 shows all permissions or access rights for users in that their permissions are divided into three, such as read/read (RW), write/write (WW) and read (R), write/read (WR), write (W) from the provisions above only administrators have full rights to read and write.

Table 3. Access rights for users used in this study.

User/Group	test01	test02
Dfsadmin	RW	RW
User1	RW	Deny

2) *Management scheduling file replication,* we can manage file replication in the DFS Management Tools, in this study used a 24-hour scheduling replication with a normal day range from Monday to Sunday.

3 Result and Discussion

3.1 System Requirements

In this implementation, we tried to build a file server infrastructure that can handle data stored centrally, allows storage resources to be shared, and provides continuous, fast, and easy access to data. This implementation requires an Active Directory Server (ADS) as a Domain Controller (DC), a server namespace, and a role-distributed file system replicated for high availability of data file sharing for clients or users. The specifications for the physical server requirements for implementing this simulation can be seen in Table 4.

Furthermore, the specifications for implementing server virtual machine requirements are described in Table 5. The software and services required for implementation are described in Table 6.

Table 4. Physical server specifications.

Hardware	Specification	Total Amount
Server (Host Hypervisor)	Workstations, CPUs Intel Xeon®, RAM 32 GB, SSD 250 GB, PSUs 750 Watts 80Plus Platinum, LAN 1 GB/sec.	1

Table 5. Virtual machine (VM) specifications.

No.	Hardware	RAM (GB)	HDD (GB)	CPU Cores
1	DC01	4	30	1
2	FS01 (File Server 1)	4	50	1
3	FS02 (File Server 2)	4	50	1
4	CL01 (PC Client)	4	30	1
5	MON (PC Monitoring)	4	30	1

Table 6. Virtual machine (VM) specifications.

No.	Software	Information
1	Hyper-V Manager	Virtualization server
2	Microsoft Windows Server 2022	Operating system
3	Microsoft Windows 10 Professional	Client operating system
4	Linux Debian	Monitoring test operating system
5	DFS Namespaces	Namespace server
6	DFS Replicated	File replication software
7	GoDaddy DNS	Domain Name Service Software
8	AD DS	Active Directory Domain Service

3.2 System Architecture

System Architecture, an overview of the network topology simulation on implementing the replicated distributed file system (DFSR), can be seen in Figure 6. In Figure 6, to run the system, at least three server VMs are needed, and two test support VM as follows:

1) DC01, a server domain controller, is installed with Windows Server 2022, AD DS, and GoDaddy DNS installation packages.
2) FS01 as server node 1 or File Sharing Primary is installed with Windows Server 2022 with the DFS, DFSR, and Join Domain installation packages.
3) FS02 as server node 2 or File Sharing Secondary is installed with Windows Server 2022 with the DFSR and Join Domain installation packages.
4) CL01 is a file-sharing user computer installed on Windows 10 Professional and the Join Domain.
5) MON as PC monitoring and taking real-time data on test results installed on Linux Debian with the Python installation package and the IO Socket Library.

The following is a description of the IP address of each VM on the architecture above (Table 7). Before the implementation process, ensure the server's operating system has Hyper-V Manager installed, which would be used as a virtualization server, as shown in Figure 7.

Table 7. IP address rules.

No.	IP Address	VMs/Servers	Computer ID
1	172.16.202.125	Domain controller	DC01
2	172.16.202.19	File sharing, DFS, DFSR	FS01
3	172.16.202.20	DFSR file sharing	FS02
4	172.16.202.253	Client PC	CL01
5	172.16.202.191	Monitoring Collect Data PC	MON

Vol 1, Issue 1, August 2023, Page 10-22
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v1i1.59

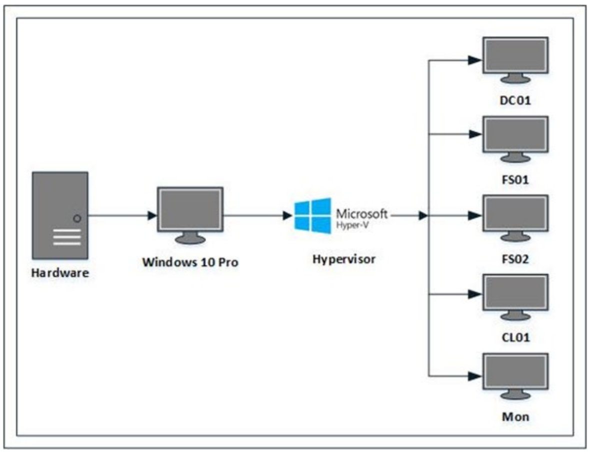

Figure 6. System architecture design.

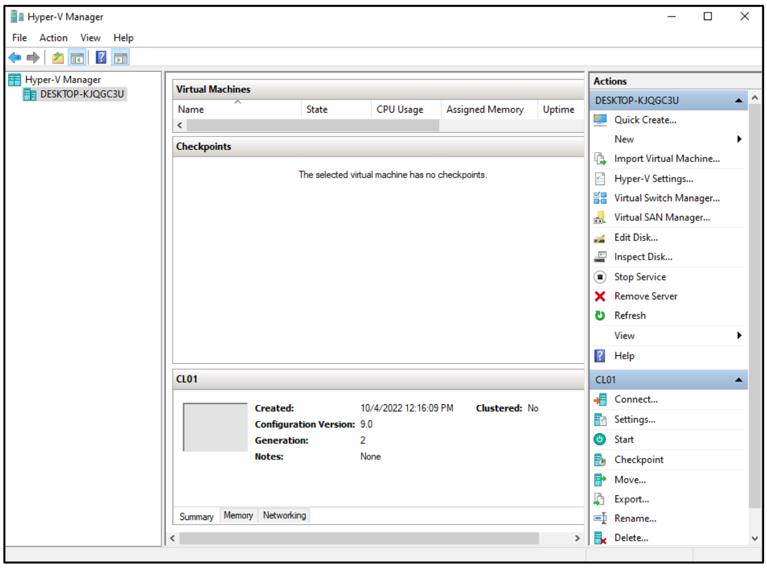

Figure 7. Hyper-V Manager user interface.

The first implementation process is to prepare all supporting servers and VM installed in Hyper-V Manager, as shown in Figure 8.

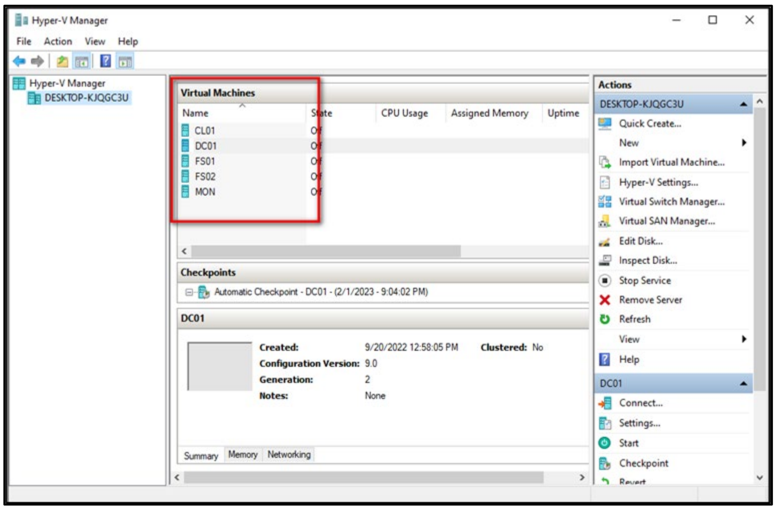

Figure 8. Hyper-V Manager is ready to use.

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 1, Issue 1, August 2023, Page 10-22
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v1i1.59

The second implementation process configures server "DC01" as Primary Domain Controllers (PDCs) and DNS Server by creating a forest as a domain address, as shown in Figure 9.

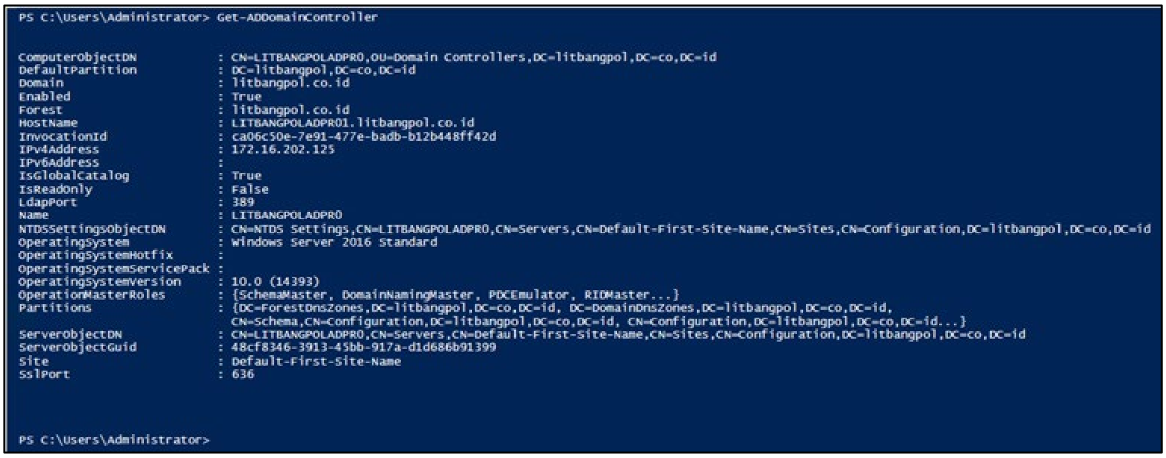

Figure 9. AD DS as domain controller settings.

Furthermore, still on the "DC01" server, configure user management by creating Organizations Units (OUs), Groups, and Users that would be used as file sharing access, as shown in Figure 10.

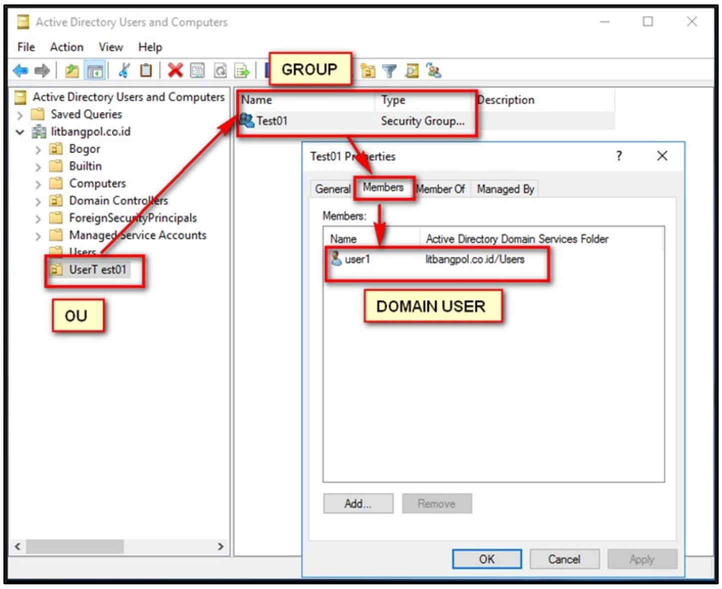

Figure 10. User management configuration.

The third implementation process, namely configuring the file sharing server "FS01" and "FS02", the steps are as follows: (1) join domain to *litbangpol.co.id*, (2) server "FS01" should be installed DFS Manager with add the role of DFS Namespace and DFS Replicated, the result is as shown in Figure 11. In addition, create a new namespace, create a folder, and add a file-sharing target folder. The results can be seen in Figure 12. Then, switch to Server "FS02" for DFS Install Management and add the DFS Replicated role. After being installed, the root namespace would automatically join the DFS management on Server FS01, and the results can be seen in Figure 13. Still in the configuration on the "FS02" server, creating a folder for data replication targets from the "FS01" server. The results can be seen in Figure 14.

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 1, Issue 1, August 2023, Page 10-22
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v1i1.59

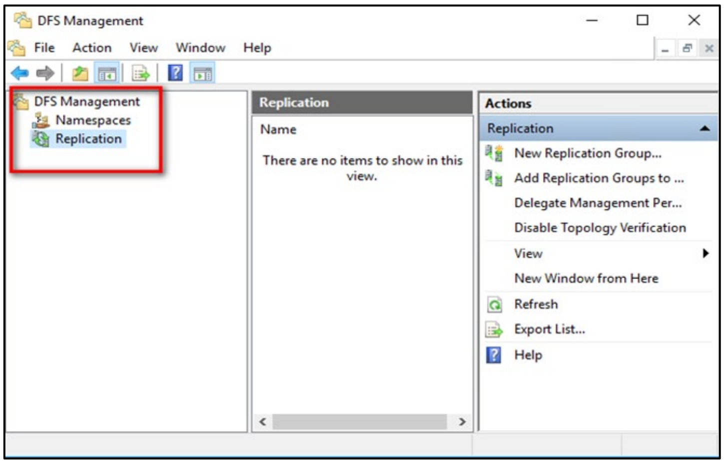

Figure 11. DFS management configuration in FS01.

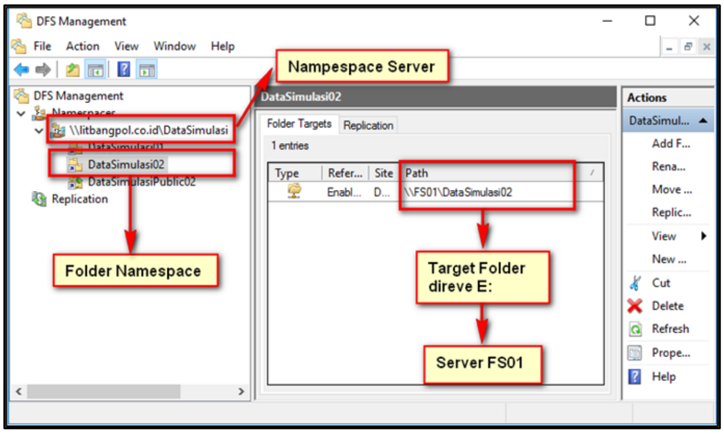

Figure 12. Namespace server configuration.

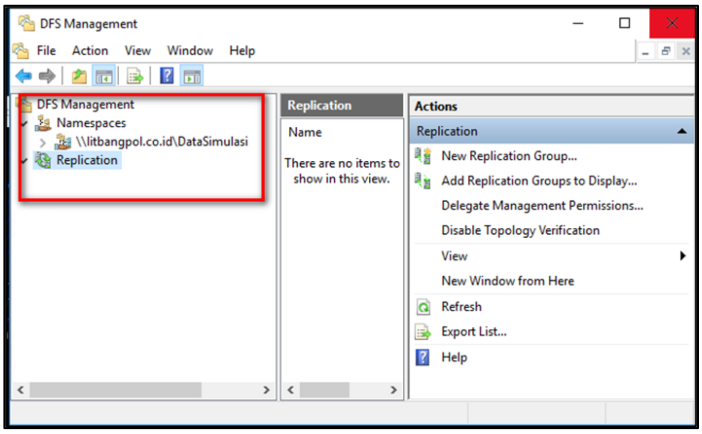

Figure 13. DFS management configuration in FS02.

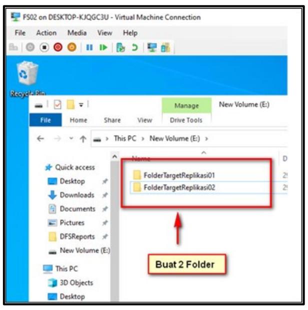

Figure 14. FS02 replication targeted two folders.

Next, after finishing configuring both Server "FS01" and "FS02", we would configure replication and publish the replication folder to the namespace for high availability file sharing purposes. The configuration is done using the DFS Management console on Server "FS01," which would be the primary replication member itself, Server "FS01". The results can be seen in Figure 15. Besides, the results or properties of the DFS Replication configuration can be seen in Table 8.

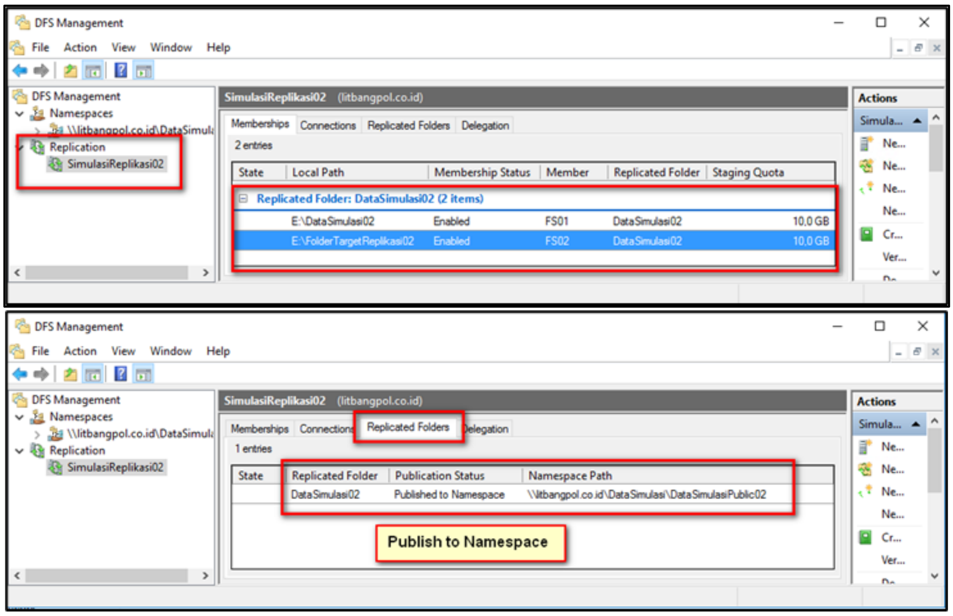

Figure 15. DFS Replication and publish replication configurations.

Table 8. DFS Replication properties.

Members	Local Path	Staging Quotas	Namespace	Public Namespace
FS01	E:\Simulation Data02	10 GB	*\\litbangpol.co.id\Simulation Data*	*\\litbangpol.co.id\DataSimulation\DataSimulationPublic02*
FS02	E:\FolderTargetReplication02	10 GB		

The fourth implementation process configures the virtual machine PC Client "CL01". Here we only need to configure the join domain on *litbangpol.co.id*. The fifth implementation process is to configure the monitoring PC "MON". This PC is installed with a simple application using web (HTML) with Python and embedding the IO Socket library engine to be used as a monitoring tool for the success rate of data testing in real time on user processes when accessing replication and high availability. The results of the monitoring dashboard can be seen in Figure 16.

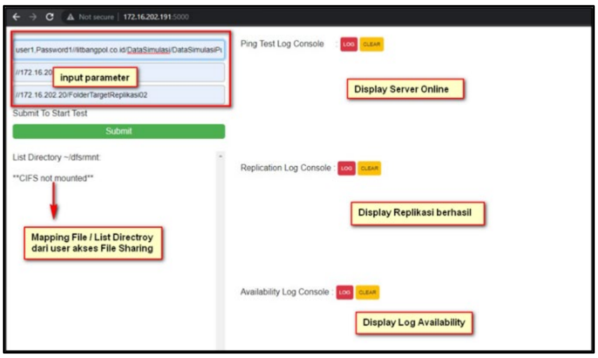

Figure 16. Monitoring dashboard interface.

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 1, Issue 1, August 2023, Page 10-22
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v1i1.59

3.3 System Testing

This system test aims to see the behavior of the DFS replicated system that has been configured previously. Test parameters include:

a. Manual Data Replication Test

Testing begins with logging in to the client's computer using the domain user "User1". This "User1" performs file-sharing access with the DFS namespace address "*\\litbangpol\Simulation Data*" in drive mapping on Windows Explorer. In addition, "User1" creates folders or files with random file sizes. Logically, the file would be replicated by file sharing server FS01 and FS02. Besides, to manually prove whether the User1 file has been replicated on member servers "FS01" and "FS02," it is necessary to log in as an administrator. The result looks like in Figure 17.

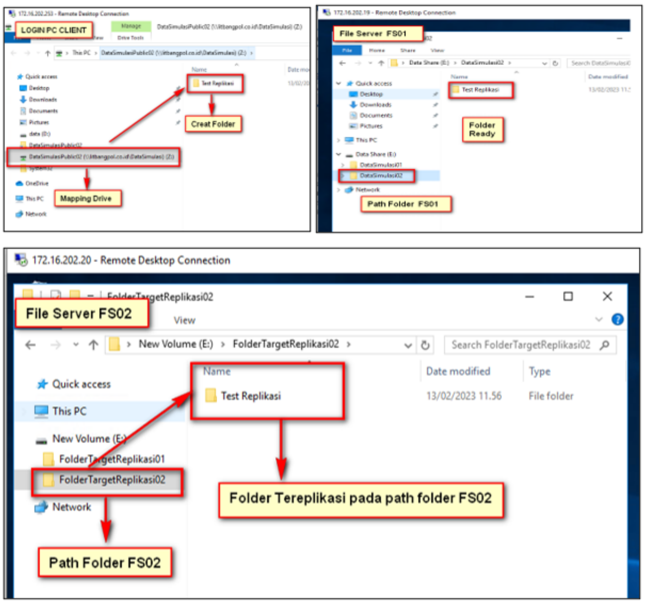

Figure 17. Folder replication test.

b. Availability Test

Testing using a monitoring application, the data that would be retrieved is data on the time and size of files that have been successfully replicated, average data availability success with testing on both member servers carried out online and offline processes. Perform parameter input, fill in as follows: (1) input login with "User1", (2) enter a password, (3) input address File Sharing publish namespace "*//litbangpol.co.id/DataSimulation/DataSimulationPublic*", (4) input server member address "*//FS01/DataSimulation02*", (5) input server member address "*//FS02/FolderTargetReplication02*", (6) then, submit it, (7) create a file would automatically with random size, and (8) run the test as shown in Figure 18, as well as the replication test output data can be seen in Figure 19 as the report table screenshots.

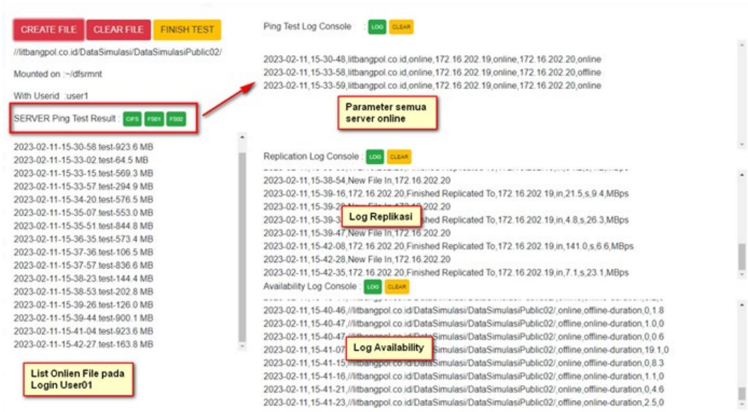

Figure 18. Real time data testing process.

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 1, Issue 1, August 2023, Page 10-22
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v1i1.59

Date	Hour	Creat File	Finish			Replication time		Size	
2/11/2023	15-30-59	New File In	172.16.202.20						
2/11/2023	15-31-26	172.16.202.20	Finished Replicated To	172.16.202.19	in	27	s	33.9	MBps
2/11/2023	15-33-03	New File In	172.16.202.20						
2/11/2023	15-33-07	172.16.202.20	Finished Replicated To	172.16.202.19	in	4.4	s	14.6	MBps
2/11/2023	15-33-17	New File In	172.16.202.20						
2/11/2023	15-33-34	172.16.202.20	Finished Replicated To	172.16.202.19	in	17	s	33.9	MBps
2/11/2023	15-33-58	New File In	172.16.202.20						
2/11/2023	15-34-08	172.16.202.20	Finished Replicated To	172.16.202.19	in	10	s	29.1	MBps
2/11/2023	15-34-21	New File In	172.16.202.20						
2/11/2023	15-34-38	172.16.202.20	Finished Replicated To	172.16.202.19	in	17	s	33.9	MBps
2/11/2023	15-35-08	New File In	172.16.202.20						
2/11/2023	15-35-29	172.16.202.20	Finished Replicated To	172.16.202.19	in	21	s	26.6	MBps
2/11/2023	15-35-52	New File In	172.16.202.20						
2/11/2023	15-36-18	172.16.202.20	Finished Replicated To	172.16.202.19	in	26	s	32.4	MBps
2/11/2023	15-36-36	New File In	172.16.202.20						
2/11/2023	15-37-10	172.16.202.20	Finished Replicated To	172.16.202.19	in	34	s	16.8	MBps

Figure 19. Replication testing is complete.

Scenario		State Time		State		Result
Date	Hour			online	offline	Avrage Availability
2/11/2023	15-31-00	offline	online-duration	12.2	0	92.62295082
2/11/2023	15-31-01	online	offline-duration	0	0.9	
2/11/2023	15-31-09	offline	online-duration	8.6	0	88.37209302
2/11/2023	15-31-10	online	offline-duration	0	1	
2/11/2023	15-33-17	offline	online-duration	126.5	0	99.20948617
2/11/2023	15-33-18	online	offline-duration	0	1	
2/11/2023	15-33-23	offline	online-duration	5.3	0	86.79245283
2/11/2023	15-33-23	online	offline-duration	0	0.7	
2/11/2023	15-33-58	offline	online-duration	34.4	0	97.09302326
2/11/2023	15-33-59	online	offline-duration	0	1	
2/11/2023	15-34-21	offline	online-duration	22.2	0	95.4954955
2/11/2023	15-34-22	online	offline-duration	0	1	
2/11/2023	15-35-09	offline	online-duration	46.6	0	97.63948498
2/11/2023	15-35-10	online	offline-duration	0	1.1	
2/11/2023	15-35-14	offline	online-duration	3.9	0	71.79487179
2/11/2023	15-35-15	online	offline-duration	0	1.1	
2/11/2023	15-35-59	offline	online-duration	43.8	0	98.40182648
2/11/2023	15-35-59	online	offline-duration	0	0.7	
2/11/2023	15-36-36	offline	online-duration	36.8	0	97.01086957
2/11/2023	15-36-37	online	offline-duration	0	1.1	
2/11/2023	15-38-03	offline	online-duration	83.6	0	98.92344498
2/11/2023	15-38-04	online	offline-duration	0	0.9	
2/11/2023	15-38-24	offline	online-duration	20.5	0	96.58536585
2/11/2023	15-38-25	online	offline-duration	0	0.7	
2/11/2023	15-38-57	offline	online-duration	31.8	0	89.62264151
2/11/2023	15-39-00	online	offline-duration	0	3.3	
2/11/2023	15-39-27	offline	online-duration	20.5	0	96.58536585
2/11/2023	15-39-28	online	offline-duration	0	0.7	
2/11/2023	15-39-46	offline	online-duration	17.9	0	86.03351955
2/11/2023	15-39-48	online	offline-duration	0	2.5	

Figure 20. Replication testing is complete.

Based on the report result in Figure 19, the average replication process is 100% successful. In addition, the data availability test results can be seen in Figure 20 as the screenshot of the availability report test table. Figure 20 shows the availability value from the tests carried out with several scenarios. By Abdullah et al. [2], the availability value is theoretically measured by the following equation (1).

$$Availability = \frac{Operation\ time - Down\ time}{Operation\ time} \times 100\% \quad (1)$$

Using the DFS Replicated method can increase the value of server availability, which is quite high. The average availability value is 92.82% from several tests carried out with several scenarios. So it can be concluded that DFS Replicated is quite effective in increasing the availability of file-sharing services.

4 Conclusion

From the results and discussion of the simulation above, it can be concluded that Distributed File Systems (DFS) make it easy for users to access file shares that are located on several file server nodes by using only one name URL/address because it is spread transparently by the DFS namespace. Besides, Distributed File System Replicated (DFSR) has made it easy for users to automatically clone files on several file server nodes at other locations that the administrator has set. In addition, in the test results of the DFSR with its *Share and Publish* features, it has succeeded in providing data availability with an average of 92.82%. Although it would be implemented in the metaverse computation, this study has contributed to the system file sharing and replication virtually.

The suggestions from the research results with this simulation, including this study, did not include aspects of the backup file system which can backup and restore daily, weekly, and monthly data and settings for how long the backup would be used. Backups from third parties, such as from the *Veeam Backup* product, can cover these backup needs because the default Windows Server feature only provides a back-one process. Besides, adding Secondary Domain Controllers (SDCs) and adding file server members would help increase the high availability service score if the Primary Domain Controller (PDC) and file server members experience problems.

Acknowledgments. The author thanks AKBP Mr. Haryono, S.T., as Head of the Police Technology Lab Unit at the National Police Research and Development Center (*Kasubag Lab Teknologi Kepolisan Puslitbang Mabes Polri*), who has helped a lot in obtaining the data needed.

References

[1] M. Yusup, U. Rahardja, and E. Prasetiyani, "Penerapan Arsitektur Three-Tier Terhadap Optimalisasi Keamanan Distributed Database," *CCIT J.*, vol. 3, no. 3, pp. 345–365, 2010, doi: 10.33050/ccit.v3i3.704.

[2] M. T. Abdullah, S. Qidri, W. Nuryadi, and S. R. Widianto, "Failover Cluster Nodes and ISCSI Storage Area Network on Virtualization Windows Server 2016," *J. Online Inform.*, vol. 5, no. 1, pp. 89–96, 2020, doi: 10.15575/join.

[3] Gerend Jason, "Gambaran umum DFS Namespace." 2022.

[4] A. Benjamin, "Ikhtisar Teknologi Hyper-V," 2022. https://docs.microsoft.com/id-ID/windows-server/virtualization/hyper-v/hyper-v-technology-overview

[5] M. R. Fadli, "Memahami desain metode penelitian kualitatif," *HUMANIKA*, vol. 21, no. 1, pp. 33–54, 2021, doi: 10.21831/hum.v21i1.38075.

[6] M. Kamal, R. R. Saedudin, and A. Almaarif, "Perancangan Sistem Keamanan Fisik pada Data Center CV Media Smart Menggunakan Metode NDLC dengan Berdasarkan Standar Tia-942," *eProceedings Eng.*, vol. 6, no. 1, Apr. 2019, doi: 10.34818/EOE.V6I1.8994.

[7] M. H. Prayitno and H. Lubis, "Penerapan Logical Unit Number (LUN) Pada Drobo Virtual Storage Dengan Metode Network Development Life Cycle (NDLC)," *Explor. J. Sist. Inf. dan Telemat. (Telekomunikasi, Multimed. dan Inform.*, vol. 11, no. 1, pp. 45–49, Jun. 2020, doi: 10.36448/JSIT.V11I1.1458.

[8] F. E. Nugroho and Y. Daniarti, "Rancang Bangun QoS (Quality of Service) Jaringan Wireless Local Area Network Menggunakan Metode NDLC (Network Development Life Cycle) di PT Trimitra Kolaborasi Mandiri (3KOM)," *JIKA (Jurnal Inform.*, vol. 5, no. 1, pp. 79–83, Apr. 2021, doi: 10.31000/JIKA.V5I1.3970.

[9] U. A. Ahmad, R. E. Saputra, and P. Y. Pangestu, "Perancangan Infrastruktur Jaringan Komputer Menggunakan Fiber Optik Dengan Metode Network Development Life Cycle (NDLC)," *eProceedings Eng.*, vol. 8, no. 6, Dec. 2021, doi: 10.34818/EOE.V8I6.17035.

[10] Y. Mulyanto and S. B. Prakoso, "Rancang Bangun Jaringan Komputer Menggunakan Sistem Manajemen Omada Controller pada Inspektorat Kabupaten Sumbawa dengan Metode Network Development Life Cycle (NDLC)," *J. Inform. Teknol. dan Sains*, vol. 2, no. 4, pp. 223–233, Nov. 2020, doi: 10.51401/JINTEKS.V2I4.825.

[11] T. Sanjaya and D. Setiyadi, "Network Development Life Cycle (NDLC) Dalam Perancangan Jaringan Komputer Pada Rumah Shalom Mahanaim," *J. Mhs. Bina Insa.*, vol. 4, no. 1, pp. 1–10, Aug. 2019.

[12] K. Rianafirin and M. T. Kurniawan, "Design network security infrastructure cabling using network development life cycle methodology and ISO/IEC 27000 series in Yayasan Kesehatan (Yakes) Telkom Bandung," in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, Mar. 2018, vol. 2018-January, pp. 1–6. doi: 10.1109/CAIPT.2017.8320681.

[13] I. Warman and A. Andrian, "Analisis Kinerja Load Balancing Dua Line Koneksi dengan Metode NTH (Studi Kasus: Laboratorium Teknik Informatika Institut Teknologi Padang)," *J. Teknoif Tek. Inform. Inst. Teknol. Padang*, vol. 5, no. 1, pp. 56–62, Apr. 2017, doi: 10.21063/JTIF.2017.V5.1.56-62.