

## **BAB I PENDAHULUAN**

### **1.1 Latar Belakang**

Salah satu masalah yang sering dihadapi oleh perusahaan adalah kebutuhan akan penyimpanan data yang tinggi. Hal ini dapat terjadi karena adanya pertumbuhan bisnis yang cepat, peningkatan jumlah data yang harus disimpan, dan kebutuhan akan keamanan data yang lebih tinggi.

Untuk mengatasi masalah ini, banyak perusahaan yang menggunakan penyimpanan data untuk menyimpan data-data mereka. Namun, solusi ini seringkali menimbulkan masalah baru, seperti biaya yang tinggi, kompleksitas pengelolaan, dan keamanan yang kurang terjamin.

Untuk mengatasi masalah ini, banyak perusahaan yang mulai menggunakan alternatif penyimpanan data. Alternatif tersebut merupakan solusi penyimpanan data yang memungkinkan perusahaan untuk mengelola data mereka sendiri, sehingga biaya yang dikeluarkan lebih efisien dan pengelolaan data lebih mudah.

Namun, ada masalah yang sering dihadapi dalam implementasi, yaitu bagaimana cara mengakses data yang disimpan dari lokasi yang berbeda. Solusi yang sering digunakan adalah dengan menggunakan teknologi *port forwarding* dan *dynamic domain name service* (DDNS)[1].

Teknologi *port forwarding* memungkinkan komputer di lokasi yang berbeda untuk terhubung dengan menggunakan *port* tertentu[1]. Sedangkan teknologi DDNS memungkinkan perusahaan untuk membuat alamat domain yang dinamis, sehingga komputer di lokasi yang berbeda dapat dengan mudah mengakses melalui alamat domain tersebut[1].

Sebelumnya, saya pernah ikut terlibat dalam proses implementasi NAS ketika saya magang di sebuah perusahaan yang akan disebut sebagai PT XYZ. Di PT XYZ, telah diimplementasikan NAS dengan perangkat keras dan perangkat lunak yang kurang lebih sama yang dibutuhkan untuk

menyimpan data-data kantor seperti Microsoft Office, hasil scan gambar, dan *file* PDF dengan ukuran *file* kurang dari 15MB yang hanya bisa diakses menggunakan jaringan di kantor. Karena keadaan itu, implementasi diinisiasi untuk membuat prototipe yang sama namun dengan penambahan teknologi *port forwarding* dan DDNS agar bisa diakses di mana pun dan kapan pun, sehingga memberikan kemudahan lebih. Oleh karena itu, diasumsikan dengan berbagai ukuran *file* yang akan diuji, yaitu 10MB, 50MB, dan 100MB.

Namun, masih terdapat beberapa masalah dalam implementasi teknologi *port forwarding* dan DDNS, seperti konfigurasi yang rumit, risiko keamanan yang tinggi, dan keterbatasan dalam jumlah koneksi yang dapat diterima[1]. Oleh karena itu, perlu adanya penelitian untuk menganalisis dan menyelesaikan masalah-masalah tersebut dalam implementasi *network attached storage* (NAS) dengan teknologi *port forwarding* dan *dynamic domain name service* (DDNS) sebagai alternatif penyimpanan data[1].

Metode incremental pada implementasi "*NETWORK ATTACHED STORAGE* (NAS) DENGAN TEKNOLOGI *PORT FORWARDING* DAN *DYNAMIC DOMAIN NAME SERVICE* (DDNS) SEBAGAI ALTERNATIF PENYIMPANAN DATA DI PT.XYZ" memberikan sejumlah kelebihan. Pertama, pendekatan ini memungkinkan peneliti untuk secara bertahap membangun fungsionalitas sistem, mempermudah pemantauan perkembangan, serta memberikan fleksibilitas dalam menyesuaikan dengan perubahan kebutuhan atau persyaratan proyek. Kedua, metode incremental dapat meminimalkan risiko kesalahan karena setiap tambahan fitur diuji dan di evaluasi secara terpisah, memungkinkan deteksi dini dan perbaikan kesalahan. Selain itu, kelebihan lainnya adalah kemampuan untuk memvalidasi setiap tahapan secara mandiri, memastikan bahwa setiap peningkatan memberikan nilai tambah yang signifikan[2].

## 1.2 Rumusan Masalah

1. Apa saja tahapan implementasi NAS dengan teknologi *port forwarding* dan DDNS sebagai solusi alternatif penyimpanan data?
2. Bagaimana kinerja jaringan dalam hal kecepatan transfer data serta fungsionalitas NAS yang menggunakan teknologi *port forwarding* dan DDNS?

## 1.3 Batasan Masalah

Masalah yang dibatasi dalam skripsi ini adalah tentang implementasi *network attached storage* (NAS) yang menggunakan teknologi *port forwarding* dan *dynamic domain name service* (DDNS) sebagai alternatif penyimpanan data. Masalah yang akan dibahas dalam skripsi ini meliputi:

1. Dalam implementasi *network attached storage* (NAS) ini menggunakan teknologi *port forwarding* dan *dynamic domain name service* (DDNS).
2. Untuk aspek fungsionalitas, dalam pengujian penulis melakukan uji akses *web gui*, *port*, *domain*, *login* dan *lupa password*.
3. Dalam pengujian performa jaringan dilakukan uji kecepatan transfer data dengan tiga jenis ukuran *file* yaitu 10MB, 50MB dan 100MB.
4. Penelitian ini hanya akan membahas implementasi *Network Attached Storage* (NAS) dengan teknologi *port forwarding* dan *dynamic domain name service* (DDNS), tanpa mencakup jenis *storage* atau teknologi lainnya.
5. Implementasi ini menggunakan *OpenMediaVault*, *Nextcloud*, *DuckDNS*, *Docker*, dan *Raspbian*. Detail teknis yang tidak terkait langsung dengan fungsi utama NAS.
6. Implementasi ini dijalankan dengan lingkungan jaringan provider GSM yaitu MyOrbit by Telkomsel dengan kecepatan Up To 15Mbps.

## 1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengetahui keberhasilan implementasi *network attached storage* (NAS) menggunakan teknologi *port forwarding* dan *dynamic domain name service* (DDNS) sebagai alternatif penyimpanan data dalam meningkatkan keamanan dan kemudahan akses data.

## 1.5 Manfaat Penelitian

Penelitian ini dapat memberikan manfaat sebagai berikut:

1. Menyediakan solusi alternatif untuk mengelola penyimpanan data dengan teknologi *Network Attached Storage* (NAS) yang dapat meningkatkan efisiensi dan fleksibilitas penyimpanan data.
2. Menggunakan teknologi *Port Forwarding* dan *Dynamic Domain Name Service* (DDNS) dapat aksesibilitas data pada sistem *Network Attached Storage* (NAS).
3. Dapat memberikan referensi bagi pengguna yang ingin mengimplementasikan layanan *Network Attached Storage* (NAS) untuk kebutuhan penyimpanan data.
4. Dapat menjadi acuan bagi perusahaan atau organisasi yang ingin mengoptimalkan pengelolaan penyimpanan data dengan menggunakan teknologi *Network Attached Storage* (NAS).
5. Dapat memberikan masukan bagi pengembangan teknologi *Network Attached Storage* (NAS) dan aplikasinya dalam penyimpanan data.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terdahulu

Sebagai bahan acuan pada umumnya, sebelum melakukan penelitian membutuhkan referensi hasil penelitian yang serupa untuk mendukung hasil penelitian yang sedang dikerjakan, berikut diantaranya:

*Tabel 2 1 Penelitian Terdahulu*

Ref	Judul	Masalah	Metode	Hasil
1.	I. Santiko and R. Rosidi, “PEMANFAATAN <i>PRIVATE CLOUD STORAGE</i> SEBAGAI MEDIA PENYIMPANAN DATA <i>E-LEARNING</i> PADA LEMBAGA PENDIDIKAN,” <i>JURNAL TEKNIK INFORMATIKA</i> , vol. 10, no. 2, pp. 137–146, Jan. 2018, doi: 10.15408/jti. v10i2.6992.	keterbatasan dan biaya tinggi dalam penyimpanan data untuk <i>e-learning</i> dalam lembaga pendidikan, yang kemudian menghambat kinerja dan fungsi <i>e-learning</i> tersebut.	<i>incremental</i>	Perancangan solusi <i>Cloud Storage</i> yang disesuaikan dengan kondisi dan kebutuhan lembaga pendidikan. Beberapa faktor yang diuji dalam penelitian ini meliputi akurasi, kecepatan akses, kapasitas, keamanan, dan efektivitas

				<p>biaya.</p> <p>Penelitian ini bertujuan untuk mengatasi kendala yang dihadapi lembaga pendidikan terkait dengan biaya tinggi yang dikeluarkan untuk penyimpanan data <i>e-learning</i> melalui penyedia layanan <i>cloud</i> eksternal.</p>
2.	<p>M. Putra and C. Sitasi, “Implementasi <i>Network Attached Storage</i> (NAS) Menggunakan <i>Synology Disk Station Manager</i> (DSM 5.2) Untuk Optimalisasi Data <i>Sharing Center</i>,” <i>Jurnal Teknik Komputer</i>, vol. IV, no. 2, pp. 39–47, 2018, doi: 10.31294/jtk. v4i2.3508.</p>	<p>memerlukan biaya lisensi dan sumber daya <i>hardware</i> yang tinggi untuk infrastruktur <i>file server</i> yang andal</p>	<p>Model Pengembangan Sistem, Metode Pengumpulan Data</p>	<p><i>File sharing</i> dan <i>file server</i> adalah solusi dari masalah yang ada dimana banyak pengguna dapat berbagi <i>file</i> dan</p>

		untuk menyimpan data secara sentralistik.		sumber daya penyimpanan dengan konsep <i>client-server</i> yang dijalankan melalui jaringan serta memungkinkan pengguna untuk berbagi informasi melalui jaringan tanpa harus mengirim <i>file</i> secara fisik dengan disket atau perangkat penyimpanan eksternal lainnya.
3.	R. A. Putra, A. Handayanto, and S. Wibowo, "IMPLEMENTASI <i>PRIVATE CLOUD STORAGE</i> MENGGUNAKAN RASPBERRY Pi," 2021. [Online]. Available:	Biaya yang Tinggi dan Kendala Konfigurasi pada Layanan Penyimpanan	<i>Research and Development</i> (R&D).	Hasil penelitian menunjukkan implementasi efektif <i>Cloud Storage</i> mengguna

	<a href="http://journals.usm.ac.id/index.php/jprt/index">http://journals.usm.ac.id/index.php/jprt/index</a>	Awan Umum, penyedia layanan penyimpanan awan sering kali menetapkan harga yang tinggi dan sering kali tidak memperbolehkan pengguna untuk menyesuaikan konfigurasi sesuai preferensi mereka.		n Raspberry Pi dengan <i>Nextcloud</i> sebagai antarmuka, memungkinkan akses <i>file</i> melalui <i>web browser</i> . Meski terbatas oleh spesifikasi RAM 1GB, sistem dapat menjalankan fungsi utama penyimpanan awan seperti <i>download</i> , <i>upload</i> , dan berbagi <i>file</i> .
4.	L. Supriadi Kalaena and W. Bagye, "IMPLEMENTASI <i>NETWORK ATTACHED STORAGE</i> (NAS) MENGGUNAKAN <i>FREENAS</i> PADA STMIK LOMBOK," <i>Jurnal Manajemen Informatika &amp; Sistem Informasi</i> ), vol. 1, [Online]. Available: <a href="http://e-journal.stmiklombok.ac.id/ind">http://e-journal.stmiklombok.ac.id/ind</a>	media penyimpanan <i>file</i> yang terdistribusi pada komputer masing-masing menyebabkan risiko kehilangan	NDLC ( <i>Network Development Life Cycle</i> )	Dengan sistem operasi <i>FreeNAS</i> yang mampu menangani tugas penyimpanan dan pengaksesan <i>file</i> dengan



	<a href="#">ex.php/misi</a>	data akibat virus atau kerusakan perangkat keras		baik memudahkan pengguna dalam mengkonfigurasi sistem, dan penggunaannya tidak memerlukan biaya besar.
5.	E. H. Orsi, "Implementasi Cloud Storage (Studi Kasus SMK Mojosongo)," <i>Jurnal Teknik Elektro</i> , vol. 19, no. 01, 2019.	penggunaan media penyimpanan fisik ( <i>hardisk</i> dan <i>flashdisk</i> ) di SMK Negeri 1 Mojosongo Boyolali. Keterbatasan efisiensi dan aksesibilitas berbagi data perlu solusi penyimpanan terpusat dan fleksibel.	metode analisis kebutuhan, implementasi, dan pengujian.	Hasil penelitian menyoroti kinerja layanan <i>cloud storage</i> yang diimplementasikan, dengan rata-rata kecepatan proses pengunggahan sekitar 139 Kbps, pengunduhan sekitar 655 Kbps, dan sinkronisasi desktop sekitar 274

				<p>Kbps (pengunggah an) dan 1.12 Mbps (pengunduha n). Pengujian pada 3 PC yang mengakses layanan <i>seafile</i> secara bersamaan memvalidasi efisiensi solusi <i>cloud storage</i> dalam memenuhi kebutuhan penyimpanan dan akses data di SMK Negeri 1 Mojosongo Boyolali.</p>
--	--	--	--	--

## 2.2 Network attached storage (NAS)

*Network Attached Storage* (NAS) adalah sebuah perangkat penyimpanan data yang terhubung ke jaringan dan memungkinkan pengguna untuk menyimpan dan mengakses data dari berbagai perangkat melalui jaringan tersebut. NAS berfungsi sebagai *server* penyimpanan yang dapat diakses oleh beberapa pengguna secara bersamaan, sehingga memudahkan dalam berbagi dan mengelola data secara terpusat[3]. NAS memiliki beberapa fungsi dan manfaat utama, antara lain:

1. Penyimpanan Terpusat: NAS memungkinkan penyimpanan data secara terpusat, sehingga memudahkan pengelolaan dan *backup* data. Hal ini sangat penting bagi organisasi yang memiliki banyak data yang perlu diakses oleh berbagai departemen atau individu[4].
2. Akses Data Cepat: NAS menggunakan protokol jaringan seperti TCP/IP untuk mengakses data, yang memungkinkan transfer data yang cepat dan efisien. Kecepatan akses ini sangat penting untuk aplikasi yang membutuhkan waktu respon cepat[5].
3. Biaya Efektif: NAS merupakan solusi penyimpanan yang lebih murah dibandingkan dengan *Storage Area Network* (SAN). NAS dapat menggunakan perangkat keras dan perangkat lunak *open-source*, yang mengurangi biaya lisensi dan pemeliharaan[6].
4. Kemudahan Konfigurasi dan Pemeliharaan: NAS relatif mudah untuk diatur dan dikelola. Banyak perangkat NAS yang dilengkapi dengan antarmuka pengguna berbasis web yang intuitif, sehingga memudahkan pengguna dalam melakukan konfigurasi dan pemeliharaan[3].

Arsitektur NAS terdiri dari beberapa komponen utama, yaitu:

1. Server NAS: Server NAS adalah perangkat keras yang berfungsi sebagai tempat penyimpanan data. Server ini biasanya dilengkapi dengan beberapa *hard drive* yang di konfigurasi dalam berbagai mode RAID untuk meningkatkan keandalan dan kinerja[4].
2. Sistem Operasi NAS: Sistem operasi NAS adalah perangkat lunak yang

mengelola fungsi penyimpanan dan akses data. Beberapa sistem operasi NAS yang populer adalah *FreeNAS*, *NAS4Free*, dan *Synology Disk Station Manager (DSM)*[5].

3. Protokol Jaringan: NAS menggunakan berbagai protokol jaringan untuk mengakses data, seperti *Network File System (NFS)* untuk sistem operasi UNIX, *Common Internet File System (CIFS)* untuk Windows, dan *File Transfer Protocol (FTP)* untuk transfer file[6].

Implementasi NAS dapat dilakukan dengan menggunakan perangkat keras khusus atau dengan mengubah komputer biasa menjadi *server* NAS menggunakan perangkat lunak tertentu. Berikut adalah langkah-langkah umum dalam mengimplementasikan NAS:

1. Instalasi Sistem Operasi NAS: Instalasi sistem operasi NAS pada perangkat keras yang akan digunakan. Misalnya, *FreeNAS* atau *NAS4Free* dapat diinstal pada komputer untuk mengubahnya menjadi *server* NAS[3].
2. Konfigurasi Jaringan: Meng konfigurasi alamat IP dan pengaturan jaringan lainnya agar *server* NAS dapat diakses melalui jaringan lokal[4].
3. Penambahan Disk dan Pengaturan RAID: Menambahkan hard *drive* ke *server* NAS dan meng konfigurasi mode RAID untuk meningkatkan keandalan dan kinerja penyimpanan[5].
4. Pengaturan Layanan dan Pengguna: Mengaktifkan layanan seperti NFS, CIFS, atau FTP dan membuat akun pengguna untuk mengatur hak akses ke data yang disimpan[6].

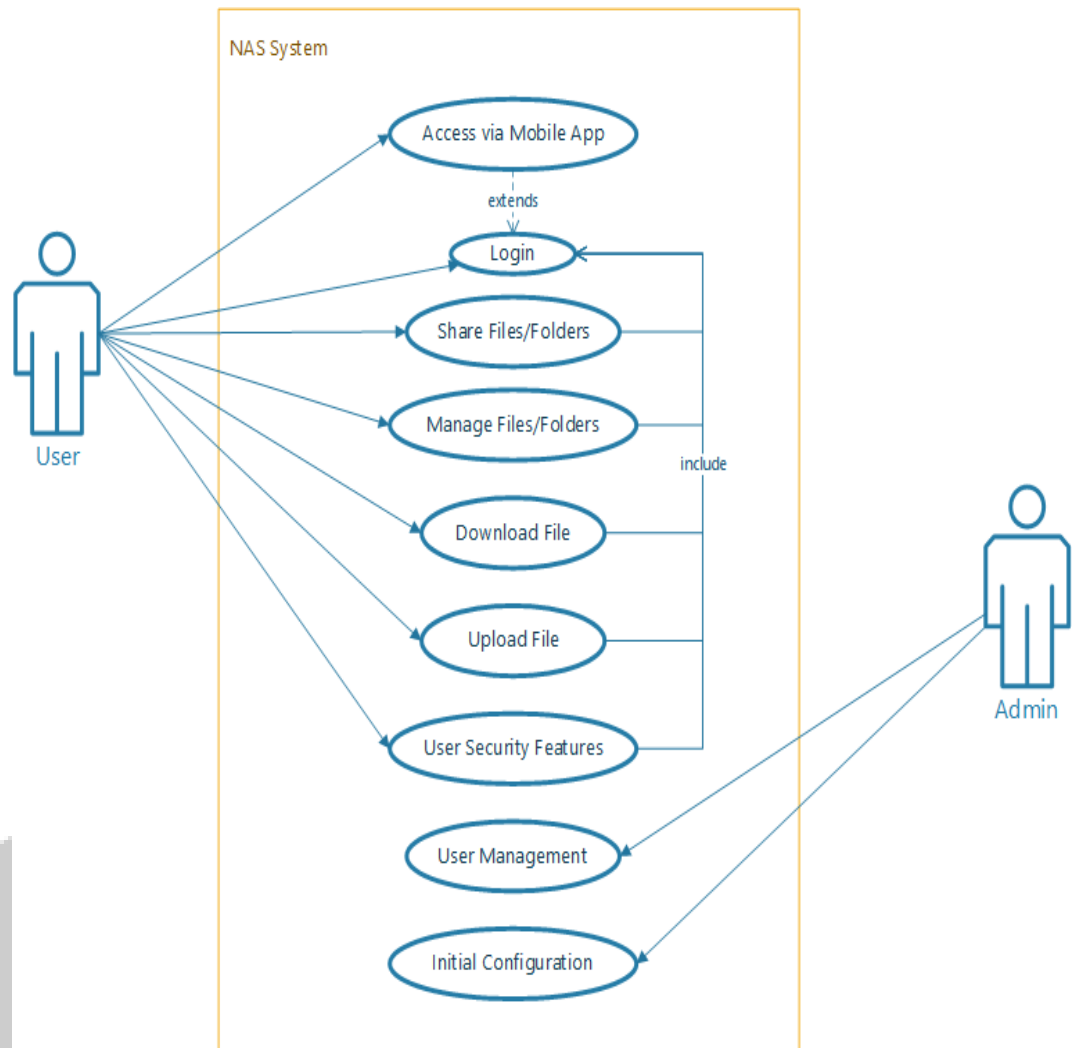
Beberapa studi kasus menunjukkan keberhasilan implementasi NAS dalam berbagai lingkungan, seperti:

1. STMIK Lombok: Implementasi NAS menggunakan *FreeNAS* di STMIK Lombok menunjukkan peningkatan kecepatan akses data dan efisiensi biaya. NAS digunakan untuk menyimpan data akademik dan praktikum mahasiswa, yang sebelumnya disimpan secara terpisah di komputer masing-masing[3].

2. PT. Prisma Dinamika Indonesia: Implementasi NAS menggunakan *Synology Disk Station Manager* (DSM) di PT. Prisma Dinamika Indonesia memudahkan dalam melakukan *sharing file* dan meningkatkan efisiensi pengelolaan data proyek, keuangan, dan penjualan[6].
3. STMIK Bina Patria Magelang: Implementasi NAS menggunakan NAS4Free di STMIK Bina Patria Magelang digunakan untuk *backup file* penting dan privasi, dengan hasil pengujian menunjukkan kecepatan upload rata-rata 3,82 Mbps dan download rata-rata 19,1 Mbps[4].

Network Attached Storage (NAS) adalah solusi penyimpanan data yang efisien dan efektif untuk berbagai kebutuhan organisasi. Dengan keunggulan dalam hal penyimpanan terpusat, akses data cepat, biaya efektif, dan kemudahan konfigurasi, NAS menjadi pilihan yang tepat untuk mengelola dan berbagi data secara terpusat. Implementasi NAS yang tepat dapat meningkatkan kinerja dan efisiensi operasional organisasi.

## Use Case Diagram dan Penjelasan Alur Penggunaan NAS



Gambar 2.1 Use case Diagram Alur Penggunaan NAS

Use case diagram ini menggambarkan interaksi antara aktor (*User* dan *Admin*) dengan sistem NAS. Berikut adalah penjelasan singkat dari tiap komponen dalam diagram tersebut:

1. Aktor:

- *User*: Pengguna biasa yang mengakses dan mengelola data di NAS.
- *Admin*: Pengelola sistem yang melakukan konfigurasi dan manajemen pengguna.

2. Use Cases untuk *User*:

- *Login*: *User* memasukkan kredensial untuk masuk ke sistem NAS.
- *Upload File*: *User* mengunggah file ke NAS setelah login.
- *Download File*: *User* mengunduh file dari NAS setelah login.

- *Manage Files/Folders*: User mengelola *file* dan folder di NAS setelah *login*.
- *Share Files/Folders*: User membagikan *file* atau folder dengan pengguna lain setelah *login*.
- *Use Security Features*: User mengaktifkan fitur keamanan seperti otentikasi dua faktor setelah *login*.
- *Access via Mobile App*: User mengakses NAS melalui aplikasi *mobile*, yang merupakan ekstensi dari *login*.

### 3. *Use Cases* untuk Admin:

- *Initial Configuration*: Admin melakukan konfigurasi awal sistem NAS, termasuk pengaturan *port forwarding* dan DDNS.
- *User Management*: Admin mengelola akun pengguna, termasuk pembuatan dan penghapusan akun.

## 2.3 *Port forwarding*

*Port forwarding* juga dikenal sebagai port mapping, adalah teknik di jaringan komputer di mana router mengarahkan lalu lintas pada port tertentu ke perangkat spesifik dalam jaringan lokal. Teknik ini mengubah alamat tujuan dan/atau *port* dari paket data untuk mencapai host dalam jaringan pribadi yang disembunyikan, seperti *Local Area Network* (LAN)[1].

*Port forwarding* memungkinkan perangkat eksternal untuk terhubung dengan layanan yang disediakan oleh perangkat dalam LAN. Contohnya, ini dapat digunakan untuk memungkinkan akses ke layanan tertentu dari internet ke perangkat dalam jaringan lokal[1].

Ketika router menerima permintaan dari perangkat di dalam jaringan, router mengambil permintaan tersebut dan meneruskan data yang diperlukan dari internet. Port digunakan untuk mengarahkan router ke komputer yang melakukan permintaan. Setiap koneksi jaringan memanfaatkan port untuk memastikan data sampai ke komputer internal yang tepat[1].

*Port forwarding* memungkinkan koneksi tanpa inisiasi dari *firewall*

NAT pada port tertentu, memungkinkan perangkat eksternal untuk memulai koneksi dan mengakses layanan di perangkat lokal. Pengaturan ini juga digunakan dalam pengaturan VPN untuk memungkinkan akses terkontrol ke jaringan *internal*[1].

Untuk mengimplementasikan port forwarding, pengguna perlu mengakses router melalui alamat IP lokalnya (sering kali 192.168.1.1) dan mengkonfigurasi pengaturan *port forwarding* melalui antarmuka *web* router. Pengguna dapat menentukan layanan khusus dan membuka *port* yang diperlukan untuk mengarahkan koneksi ke perangkat dalam jaringan[7].

Keamanan dalam pengaturan *port forwarding* penting, karena membuka port di firewall NAT meningkatkan risiko terhadap serangan dari luar. Dianjurkan untuk hanya membuka *port* yang benar-benar diperlukan dan memastikan bahwa konfigurasi *firewall* dan langkah-langkah keamanan lainnya telah diatur dengan benar[5].

#### **2.4 Dynamic domain name service (DDNS)**

*Dynamic domain name service* (DDNS) adalah teknik yang digunakan untuk memperbarui server nama di *Domain Name System* (DNS) secara otomatis dengan konfigurasi DDNS aktif dari nama host, alamat, atau informasi lainnya yang telah dikonfigurasi. Teknik ini sangat berguna untuk mempertahankan nama domain tetap untuk alamat IP dinamis, sehingga memungkinkan akses tanpa gangguan ke layanan tanpa perlu memperbarui catatan DNS secara manual setiap kali alamat IP berubah.

DDNS mengatasi tantangan yang ditimbulkan oleh alamat IP dinamis, yang dapat berubah dengan sering. Sistem DNS tradisional bersifat statis dan tidak mendukung perubahan alamat IP yang terjadi dengan beberapa Penyedia Layanan Internet (ISP). DDNS memecahkan masalah ini dengan memungkinkan nama domain diberikan kepada alamat IP dinamis. Dengan cara ini, meskipun alamat IP berubah, nama domain tetap sama, memastikan akses terus-menerus ke perangkat atau layanan yang terkait



dengan nama domain tersebut[1].

Salah satu aplikasi umum dari DDNS adalah dalam solusi akses jarak jauh. Misalnya, pengaturan remote desktop, kamera keamanan, atau perangkat IoT lainnya dapat disederhanakan dengan DDNS, karena memungkinkan pengguna untuk menghubungkan perangkat mereka menggunakan nama domain yang konsisten daripada melacak perubahan alamat IP. Ini sangat menguntungkan untuk usaha kecil dan jaringan rumah yang tidak memiliki alamat IP statis.

Layanan DDNS seperti No-IP dan DynDNS menawarkan *platform* di mana pengguna dapat membuat akun DNS dinamis untuk mengelola nama domain mereka secara efektif. Layanan ini menyediakan alat untuk memetakan nama domain ke alamat IP dinamis, memungkinkan pengguna untuk mengelola konfigurasi jaringan mereka dengan mudah dan efisien[1], [7].

Implementasi DDNS melibatkan konfigurasi perangkat atau sistem jaringan untuk berkomunikasi dengan penyedia DDNS. Misalnya, dalam pengaturan *Digital Video Recorder (DVR)*, DDNS dapat dikonfigurasi untuk memperbarui catatan DNS secara otomatis setiap kali alamat IP eksternal berubah. Ini memastikan DVR tetap dapat diakses melalui nama domain yang konsisten, terlepas dari perubahan alamat IP yang ditetapkan oleh ISP. Proses pengaturannya biasanya melibatkan mengaktifkan DDNS di pengaturan jaringan dan memilih penyedia DDNS[7].

Pertimbangkan penggunaan DDNS dalam pengaturan *server OpenVPN*. Karena alamat IP publik yang disediakan oleh ISP dapat berubah, menggunakan DDNS sangat penting untuk mempertahankan akses konsisten ke *server*. Dalam skenario ini, layanan seperti NO-IP dapat digunakan untuk memetakan nama host ke alamat IP, memungkinkan pengguna untuk menghubungkan ke server VPN menggunakan nama host alih-alih alamat IP. Ini menyederhanakan proses koneksi dan memastikan bahwa pengguna selalu dapat mengakses server tanpa perlu mengetahui alamat IP saat ini[1]

## 2.5 Model *Incremental*

Model Incremental adalah salah satu model proses pengembangan perangkat lunak yang menggabungkan elemen dari model sekuensial linear (waterfall) dengan filosofi iteratif dari model prototyping. Dalam model ini, perangkat lunak dikembangkan secara bertahap melalui serangkaian iterasi yang disebut "increment"[2]. Setiap increment menghasilkan bagian dari perangkat lunak yang dapat digunakan, yang kemudian dievaluasi dan digunakan sebagai dasar untuk increment berikutnya.

### Tahapan dalam Model Incremental

1. Analisis Kebutuhan: Pada tahap ini, kebutuhan perangkat lunak dikumpulkan dan dianalisis untuk menentukan fitur dan fungsi yang akan disertakan dalam increment pertama.
2. Desain: Berdasarkan hasil analisis kebutuhan, desain perangkat lunak dibuat untuk increment pertama. Desain ini mencakup struktur data, arsitektur perangkat lunak, antarmuka, dan detail prosedural.
3. Implementasi: Desain yang telah dibuat kemudian diimplementasikan menjadi kode program. Increment pertama dari perangkat lunak dibuat dan diuji.
4. Pengujian: Increment pertama diuji untuk memastikan bahwa ia memenuhi kebutuhan yang telah ditentukan. Pengujian ini mencakup pengujian unit, pengujian integrasi, dan pengujian sistem.
5. Evaluasi dan Perencanaan *Increment* Berikutnya: Setelah increment pertama diuji dan disetujui, evaluasi dilakukan untuk menentukan perubahan atau penambahan yang diperlukan untuk increment berikutnya. Proses ini diulang untuk setiap increment hingga perangkat lunak selesai.

### Keuntungan Model Incremental

1. Manajemen Risiko yang Lebih Baik: Dengan mengembangkan perangkat lunak secara bertahap, risiko dapat diidentifikasi dan dikelola

lebih awal dalam proses pengembangan.

2. Umpan Balik Pengguna yang Cepat: Setiap increment memberikan kesempatan untuk mendapatkan umpan balik dari pengguna, yang dapat digunakan untuk memperbaiki dan menyesuaikan perangkat lunak sebelum increment berikutnya.
3. Penggunaan Sumber Daya yang Efisien: Increment awal dapat dikembangkan dengan tim yang lebih kecil, dan tim dapat ditingkatkan seiring dengan bertambahnya kompleksitas perangkat lunak.

#### Kekurangan Model Incremental

1. Memerlukan Perencanaan yang Matang: Setiap increment harus direncanakan dengan hati-hati untuk memastikan bahwa setiap bagian perangkat lunak dapat berfungsi dengan baik secara mandiri.
2. Ketergantungan pada Increment Sebelumnya: Kesalahan atau kekurangan dalam increment sebelumnya dapat mempengaruhi increment berikutnya, sehingga memerlukan perhatian ekstra dalam pengujian dan evaluasi.

Contoh penerapan model incremental dapat ditemukan dalam pengembangan perangkat lunak pengolah kata. Increment pertama mungkin hanya mencakup fungsi dasar seperti manajemen file dan pengeditan teks. Increment berikutnya dapat menambahkan fitur-fitur seperti pemeriksaan ejaan dan tata bahasa, serta kemampuan tata letak halaman yang lebih canggih[2].

## 2.6 Raspberry Pi OS

Raspberry Pi OS, sebelumnya dikenal sebagai Raspbian, merupakan sistem operasi yang dirancang khusus untuk Raspberry Pi, sebuah komputer mini yang sangat populer dalam berbagai proyek DIY dan pengembangan perangkat. Pengembangan Raspberry Pi OS dilakukan oleh Raspberry Pi Foundation, dan sistem operasi ini memiliki karakteristik yang membuatnya

menjadi pilihan utama dalam pengembangan berbasis Raspberry Pi[8].

Raspberry Pi OS memiliki beberapa keunggulan yang mencolok. Salah satunya adalah penyesuaian yang kuat dengan Raspberry Pi, yang memastikan bahwa sistem operasi ini berjalan optimal pada perangkat tersebut. Hal ini sangat penting mengingat keterbatasan perangkat keras Raspberry Pi, dan optimasi ini memungkinkan kinerja yang maksimal[9].

Selain itu, Raspberry Pi OS adalah perangkat lunak *open-source*, yang berarti bahwa pengguna dapat mengunduh, menginstal, dan menggunakan sistem ini tanpa biaya. Faktor ini sangat mendukung komunitas pengembang dan DIY, yang selalu mencari solusi yang ekonomis[8].

Kelebihan lain dari Raspberry Pi OS adalah efisiensi dalam penggunaan sumber daya. Dengan konsumsi daya yang rendah, sistem ini mampu berjalan pada perangkat keras dengan kemampuan komputasi yang terbatas. Hal ini membuatnya sangat cocok untuk proyek-proyek yang memerlukan perangkat keras yang hemat daya[10].

Raspberry Pi OS juga menawarkan beragam aplikasi yang dapat digunakan untuk berbagai jenis proyek. Ini mencakup aplikasi untuk pemrograman, *media center*, *Internet of Things* (IoT), dan banyak lagi. Dengan berbagai aplikasi yang tersedia, Raspberry Pi OS mendukung fleksibilitas dalam menjalankan berbagai jenis proyek dengan Raspberry Pi[8].

Dalam konteks penelitian ini, Raspberry Pi OS dapat digunakan sebagai dasar yang relevan. Dalam berbagai proyek pendidikan dan pengembangan, Raspberry Pi OS digunakan untuk mencapai berbagai tujuan, seperti membuat *media center* dengan Kodi atau OSMC, mengonversi Raspberry Pi menjadi server rumah untuk berbagi *file* dan mencetak, serta menciptakan lingkungan pemrograman yang mendukung bahasa pemrograman seperti Python atau Scratch. Selain itu, Raspberry Pi OS sering digunakan dalam proyek-proyek *Internet of Things* (IoT) untuk mengendalikan perangkat rumah pintar atau memantau kondisi seperti suhu dan kelembaban[8], [9].

Tidak hanya memiliki sejumlah kelebihan, Raspberry Pi OS juga mendapat dukungan aktif dari komunitas pengguna yang luas. Ini berarti bahwa sumber daya seperti tutorial, dukungan *online*, dan diskusi komunitas sangat mudah diakses, yang dapat memudahkan pengguna dalam mengimplementasikan dan memanfaatkan Raspberry Pi OS dalam berbagai proyek berbasis Raspberry Pi.

## 2.7 Openmediavault

Openmediavault (OMV) adalah perangkat lunak manajemen penyimpanan berbasis *open-source* yang dikembangkan khusus untuk mengelola penyimpanan berbasis jaringan (NAS) dan server berbasis Linux. Dalam konteks landasan teori untuk skripsi ini, OMV menjadi sangat relevan karena merupakan salah satu solusi yang dapat digunakan untuk mengatasi masalah penyimpanan data yang kompleks dan memerlukan manajemen yang cermat[11].

Salah satu keunggulan utama dari OMV adalah sifatnya yang *open source*, yang berarti bahwa perangkat lunak ini tersedia secara gratis dan dapat diakses oleh siapa saja tanpa perlu membayar biaya lisensi. Hal ini menjadi penting terutama dalam konteks lembaga pendidikan yang sering kali memiliki anggaran yang terbatas. Pengguna dapat mengunduh dan menginstal OMV tanpa mengeluarkan biaya tambahan, yang menjadikan nya solusi yang ekonomis[12].

OMV juga menonjol berkat antarmuka *web* yang *user-friendly*. Antarmuka ini dirancang untuk menjadi intuitif dan mudah digunakan, bahkan oleh mereka yang tidak memiliki pengetahuan teknis yang mendalam. Dengan demikian, pengguna dapat dengan mudah mengatur dan mengkonfigurasi penyimpanan serta mengelola server tanpa perlu keahlian khusus[12].

Selain itu, OMV menawarkan berbagai fitur yang kuat untuk manajemen penyimpanan. Ini termasuk dukungan untuk berbagai sistem *file*

dan teknologi penyimpanan seperti RAID, yang memungkinkan pengguna untuk mengatur tingkat keamanan dan kinerja penyimpanan sesuai dengan kebutuhan mereka. Kemampuan ini menjadi sangat penting dalam lembaga pendidikan di mana data *e-learning* dan informasi penting harus diatur dengan baik dan diamankan[11].

Contoh penggunaan OMV dapat mencakup implementasi di lembaga pendidikan. Lembaga pendidikan sering kali menghadapi tantangan biaya tinggi terkait dengan penyimpanan data *e-learning* melalui penyedia layanan *cloud* eksternal. Dalam konteks ini, OMV dapat digunakan untuk mengatasi kendala ini dengan memberikan kontrol penuh kepada lembaga pendidikan atas penyimpanan data mereka sendiri[11]. Dengan dukungan untuk manajemen akses yang kuat, *backup* otomatis, dan fleksibilitas dalam konfigurasi, OMV memungkinkan lembaga pendidikan untuk mengatur penyimpanan data *e-learning* mereka dengan efektif dan ekonomis.

Selain itu, OMV juga dapat digunakan oleh individu atau pengembang perangkat lunak yang ingin membuat server rumahan. Dalam kasus ini, OMV memungkinkan mereka untuk mengatur server yang berfungsi sebagai pusat media untuk *streaming* konten ke perangkat lain dalam jaringan, seperti *Smart TV* atau perangkat *mobile*. Fitur-fitur tambahan yang dapat diintegrasikan melalui *plug-in* dan ekstensi membuat OMV menjadi pilihan yang sangat fleksibel untuk berbagai keperluan pengguna individu[13].

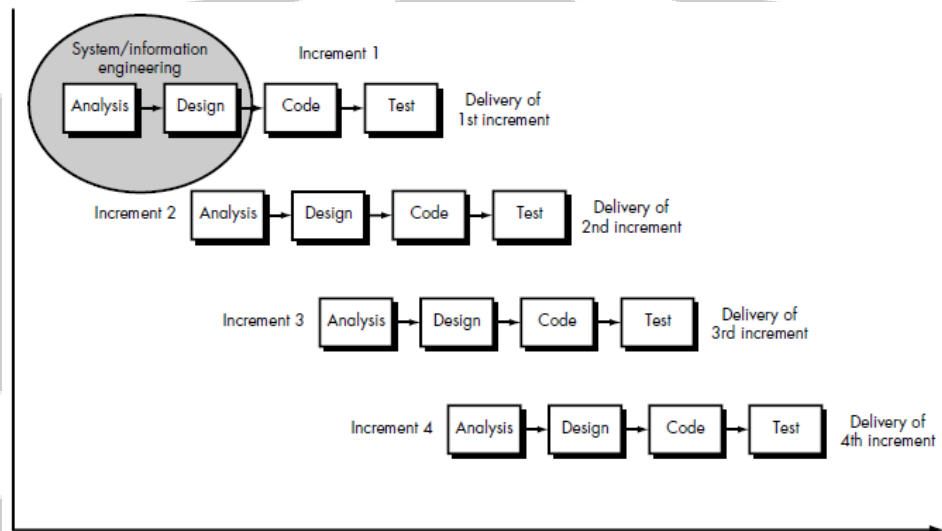
Dengan demikian, *openmediavault (OMV)* adalah solusi yang sangat relevan dalam konteks penyimpanan data, terutama dalam lembaga pendidikan dan penggunaan rumahan. Kelebihan OMV dalam hal *open source*, antarmuka *web* yang *user-friendly*, dan fitur-fitur manajemen penyimpanan yang kuat menjadikannya solusi yang layak untuk dipertimbangkan dalam penelitian ini[14].

## BAB III

### METODE PENELITIAN

#### 3.1 Desain Penelitian

Metode Incremental terapan untuk memastikan peningkatan bertahap pada fungsionalitas sistem. Metode ini memberikan keleluasaan untuk mengatasi kendala pengembangan yang tidak memungkinkan proses sepenuhnya linear[2]. Dengan fokus pada peningkatan aksesibilitas, keamanan, dan efisiensi penyimpanan data, penggunaan teknologi *port forwarding* dan DDNS dapat dioptimalkan secara bertahap. Keunggulan metode incremental juga terletak pada kemampuannya mengurangi risiko kesalahan konfigurasi yang kompleks melalui pengujian dan evaluasi terpisah setiap tambahan fitur[2].



Gambar 3 1 Siklus incremental model

#### 3.2 Perencanaan

Tentukan tujuan dan sasaran implementasi *Network Attached Storage* (NAS), identifikasi sumber daya yang dibutuhkan, dan buat rencana proyek yang mencakup penggunaan *port forwarding* dan teknologi DDNS. Berikut

identifikasi perangkat yang dibutuhkan:

*Tabel 3 1 Perencanaan Perangkat*

No	Hardware	Satuan
1	PC Client	1
2	Raspberry Pi 4 Model B	1
3	Router Huawei Model B311B	1
4	HDD Seagate 1TB	1

Dalam implementasi ini openmediavault digunakan untuk melakukan konfigurasi pada NAS dan untuk menyediakan layanan agar bisa diakses dari internet maka dibutuhkan beberapa perangkat lunak pendukung yang dipasang pada *openmediavault*, yaitu:

*Tabel 3 2 perangkat lunak*

Aplikasi Pendukung	Lokasi Pemasangan	Fungsi
Docker	openmediavault	Untuk menjalankan aplikasi pendukung agar NAS dapat diakses dari internet tanpa menambah beban server.
Portainer	docker	Dengan web GUI memudahkan untuk mengelola stack atau container yang dibutuhkan dalam implementasi.
Nextcloud	docker	Digunakan template aplikasi nextcloud sebagai user interface NAS
Mariadb	docker	Digunakan untuk menyimpan database konfigurasi dan user nextcloud
SWAG	docker	Untuk menjembatani aplikasi nextcloud ke duckdns agar bisa diakses dari internet

Adapun penyedia layanan jaringan pada implementasi ini menggunakan MyOrbit by Telkomsel yaitu salah satu provider GSM yang mana ada beberapa hal yang tidak bisa dikontrol karena ketentuan dari penyedia layanan walaupun dengan klaim *Up To 15Mbps*, namun tentu yang didapatkan bisa lebih atau bahkan kurang dari klaim penyedia layanan.

### 3.3 Analisis

Menganalisis infrastruktur yang ada, mengidentifikasi kesenjangan, dan menentukan persyaratan untuk implementasi *Network Attached Storage* (NAS) dengan teknologi *port forwarding* dan DDNS. Berikut adalah hasil analisis kebutuhan perangkat lunak dan perangkat keras yang digunakan



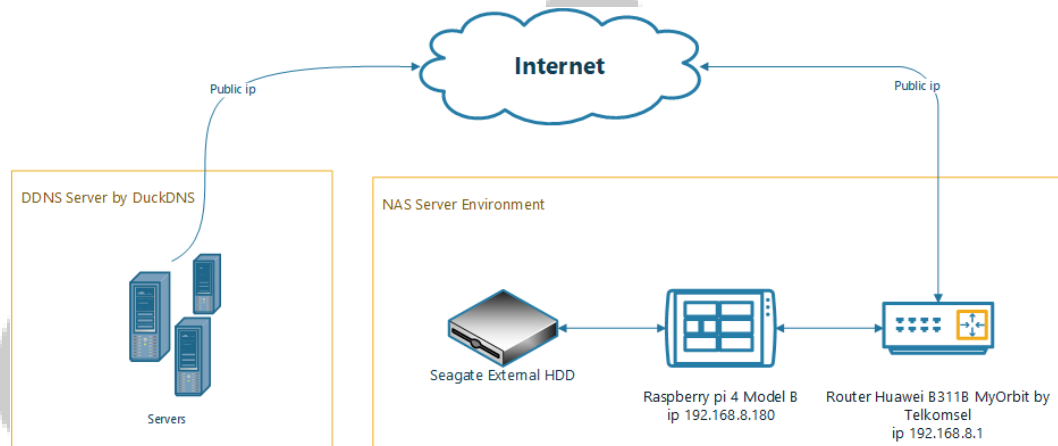
dalam implementasi kali ini:

*Tabel 3 3 Analisis Hardware dan Sistem Operasi*

No.	Hardware	Sistem Operasi
1	PC Client	Windows 11 Pro
2	Raspberry Pi 4 Model B	Raspberry Pi OS Lite Version
3	Router Huawei B311B	Huawei WEBUI 11.0.2.1
4	HDD Seagate 1TB	-

### 3.4 Desain

Desain topologi *Network Attached Storage* (NAS), termasuk komponen perangkat keras dan perangkat lunak, topologi jaringan, dan langkah-langkah keamanan, dengan integrasi teknologi *port forwarding* dan DDNS.



*Gambar 3 2 Desain Topologi Jaringan*

### 3.5 Implementasi

Menginstal dan meng konfigurasi komponen perangkat keras dan perangkat lunak, termasuk perangkat NAS, perangkat penyimpanan, dan perangkat lunak virtualisasi, dengan menggunakan *port forwarding* dan teknologi DDNS.

### 3.6 Pengujian

Menguji implementasi NAS untuk memastikan bahwa implementasi tersebut memenuhi persyaratan dan fungsi seperti yang diharapkan, termasuk penggunaan *port forwarding* dan teknologi DDNS serta keamanan. Penyedia layanan jaringan yang digunakan dalam implementasi ini adalah MyOrbit by Telkomsel, salah satu provider GSM. Meskipun layanan ini mengklaim kecepatan hingga 15Mbps dengan perbandingan 1:1, kecepatan yang sebenarnya dapat bervariasi, bisa lebih tinggi atau lebih rendah dari klaim tersebut, tergantung pada ketentuan dan kondisi dari penyedia layanan yang tidak sepenuhnya dapat dikontrol yang dapat mempengaruhi hasil uji.

Adapun parameter pengujian sebagai berikut:

1. Fungsionalitas, dengan melakukan pengujian akses:
  - Raspberry Pi; akses ssh, web *gui* openmediavault dan portainer
  - *Port Forwarding*, akses *port* yang telah diteruskan oleh *router*
  - *Dynamic Domain Name Services*, akses menyambungkan swag dengan duckdns agar *Nextcloud* dapat diakses dengan domain
  - *Nextcloud*, login dengan akses kredensial yang valid, tidak valid serta fitur lupa *password*
2. Performa Jaringan, dengan melakukan pengujian transfer data:
  - *Upload* dengan tiga ukuran *file* yaitu 10MB, 50MB dan 100MB
  - *Download* dengan tiga ukuran *file* yaitu 10MB, 50MB dan 100MB

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi *Network Attached Storage (NAS)*

##### 4.1.1 Analisis Kebutuhan

Dalam penelitian ini telah diketahui bahwa kebutuhan yang perlu diperhatikan untuk melakukan implementasi ini adalah sebagai berikut:

1. Kebutuhan Teknis

Dalam memperhatikan kebutuhan ini, perlu disadari bahwa implementasi ini diharapkan menjadi solusi alternatif penyimpanan yang dapat diakses baik secara lokal maupun melalui internet. Oleh karena itu, penggunaan *port forwarding* diperlukan untuk memungkinkan akses ke NAS dari luar jaringan lokal melalui internet. Selain itu, penggunaan DDNS juga diperlukan untuk menyediakan alamat *domain* yang mudah diingat dan stabil, meskipun alamat IP eksternal berubah.

2. Kebutuhan Pengguna

Dalam konteks kebutuhan penggunaan, implementasi ini bertujuan untuk memberikan kemudahan akses dan penggunaan data, termasuk kemampuan untuk mengakses nya melalui perangkat seluler dan komputer pribadi. Akses ini dapat dilakukan melalui aplikasi yang disediakan atau langsung dengan mengunjungi domain yang telah tersedia.

##### 4.1.2 Pemilihan Perangkat

Dalam konteks pemilihan perangkat, ada dua klasifikasi yang harus dipertimbangkan, yakni perangkat keras dan perangkat lunak. Di bawah ini adalah penjabaran detail mengenai setiap kategori perangkat beserta alasan mengapa masing-masing dipilih:

1. Perangkat Keras :

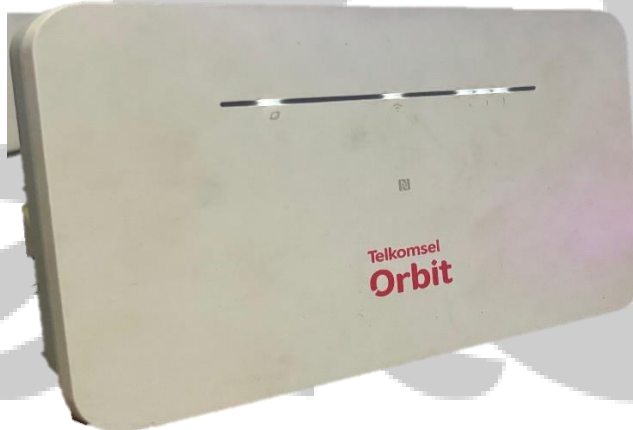
a) Raspberry Pi 4 Model B



*Gambar 4 1 Raspberry Pi 4 Model B*

Server NAS : dengan konsumsi sumber daya listrik yang minim menjadi sebuah keuntungan namun hanya pada skala kantor kecil.

b) Router Huawei B311B



*Gambar 4 2 Router Huawei B311B*

Sumber internet NAS : selain menjadi sumber internet NAS, router ini juga memiliki fitur teknologi *port forwarding* yang dibutuhkan dalam implementasi ini.

c) HDD Seagate 1TB

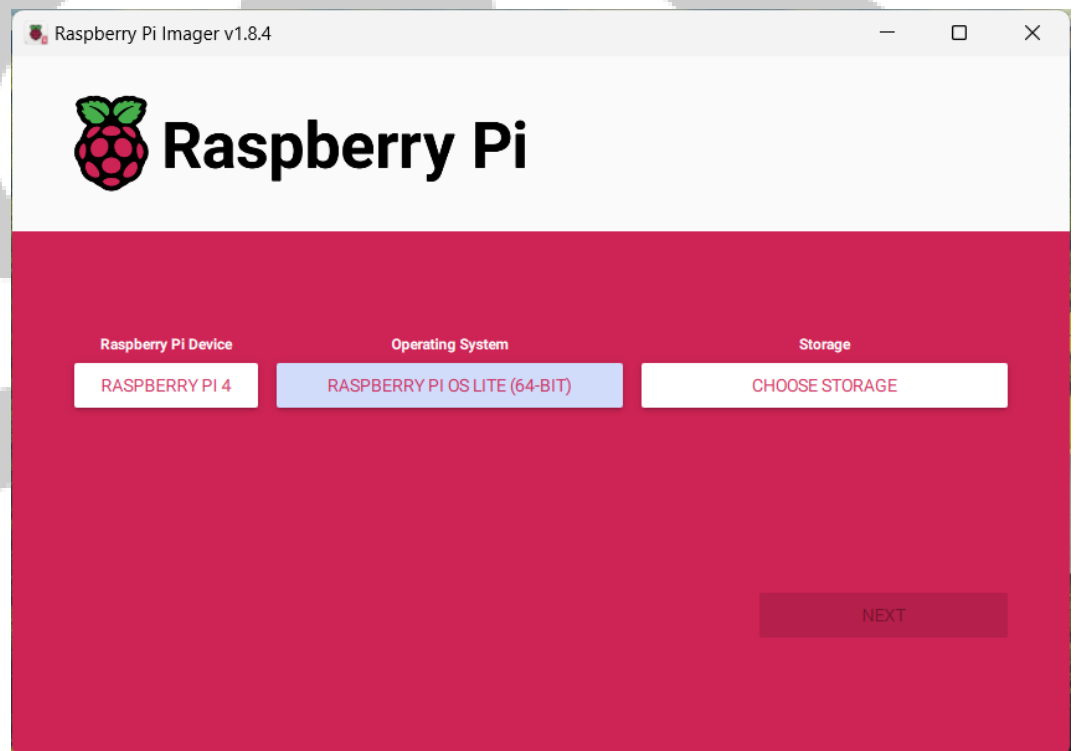


*Gambar 4 3 HDD Seagate 1TB*

Media Penyimpanan NAS : agar mampu menunjang kebutuhan penyimpanan dari pengguna.

## 2. Perangkat Lunak:

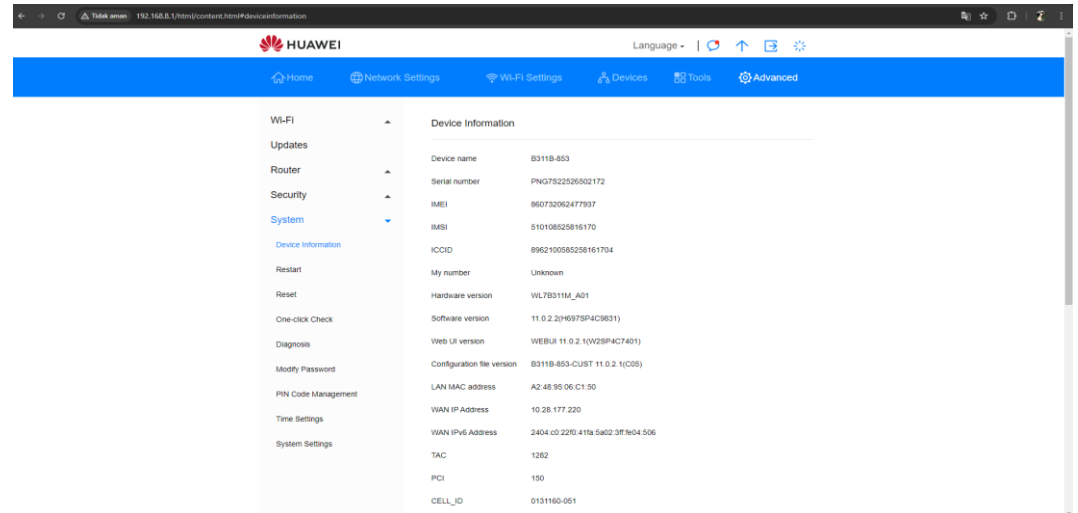
a. Raspberry Pi OS Lite Version



*Gambar 4 4 Software Raspberry Pi OS Lite Version*

OS NAS : dipilih agar perangkat keras dapat menghasilkan kinerja yang optimal, selain itu os ini juga dibutuhkan agar mampu memasang *openmediavault* yang merupakan salah satu aplikasi yang dibutuhkan ketika ingin melakukan konfigurasi pada NAS.

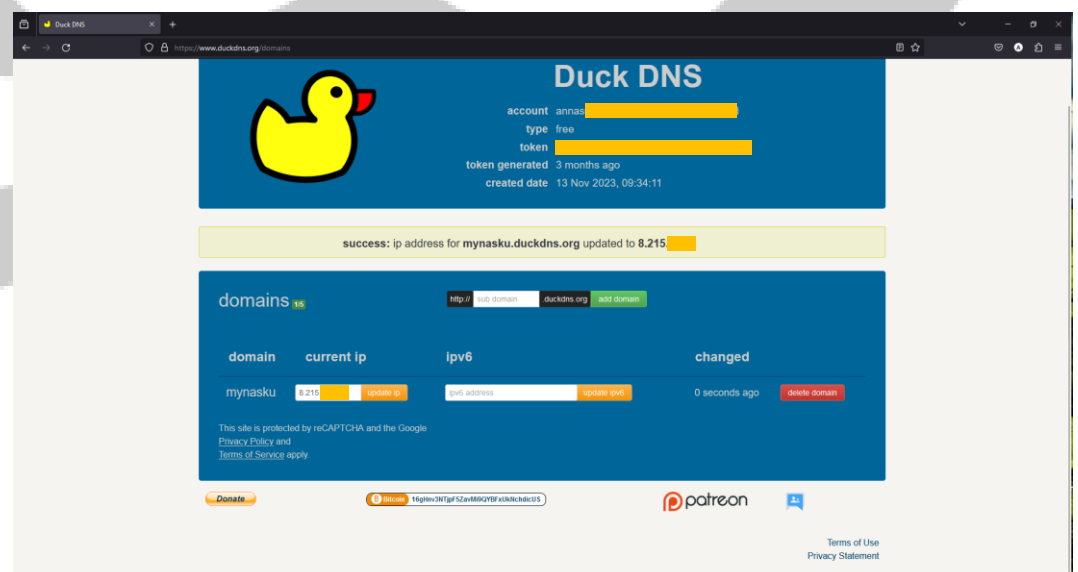
b. Router Huawei B311B



Gambar 4 5 Software Router Huawei B311B

*Port forwarding* : dikarenakan fitur ini sudah terdapat pada router maka konfigurasi bisa langsung dijalankan pada router dengan mengakses web gui.

c. Duckdns



Gambar 4 6 Software Duckdns

DDNS : karena kemudahan yang ditawarkan pengguna dengan mudah dapat melakukan konfigurasi ddns yang dibutuhkan dengan cepat dan mudah, selain itu layanan ddns pada duckdns tidak dipungut biaya atau dengan kata lain “gratis”.

#### 4.1.3 Konfigurasi Raspberry Pi

Dalam implementasi ini, berikut adalah konfigurasi Raspberry Pi yang digunakan untuk dijadikan sebuah server NAS :

1. Langkah pertama perlu memastikan bahwa perangkat telah berjalan dengan menggunakan OS dari Raspberry Pi OS Lite version dikarenakan syarat agar openmediavault dapat dipasang pada perangkat adalah kecocokan dengan OS tersebut.
2. Kemudian lakukan instalasi openmediavault dengan menjalankan perintah berikut pada terminal :










*Tabel 4 1Source Code Installation Openmediavault*

```
sudo apt update
sudo apt upgrade
wget -O -
https://raw.githubusercontent.com/OpenMediaVault-
Plugin-Developers/installScript/master/install |
sudo bash
sudo reboot
```

3. Setelah melakukan instalasi silakan jalankan perintah *hostname -I* untuk mengetahui alamat ip *openmediavault* agar dapat mengakses web gui dengan kredensial default yaitu dengan user: *admin* dan *password: openmediavault*.
4. Pada web gui *openmediavault* pasang plugin docker yang bertujuan untuk digunakan *Nextcloud* beserta aplikasi pendukung nya
5. Lanjutkan konfigurasi *port forwarding* dan *ddns*

#### 4.1.4 Konfigurasi Port Forwarding

1. Kunjungi web gui pada router kemudian *login*
2. Setelah *login*, pada umumnya menu konfigurasi *port forwarding* berada pada pilihan *security* menyesuaikan dengan router yang digunakan
3. Lalu pastikan *internal port* dan *external* port sudah benar dan ip yang dituju tidak ada kekeliruan
4. Lanjutkan konfigurasi ddns. Berikut konfigurasi *port forwarding* yang telah diterapkan pada implementasi ini:

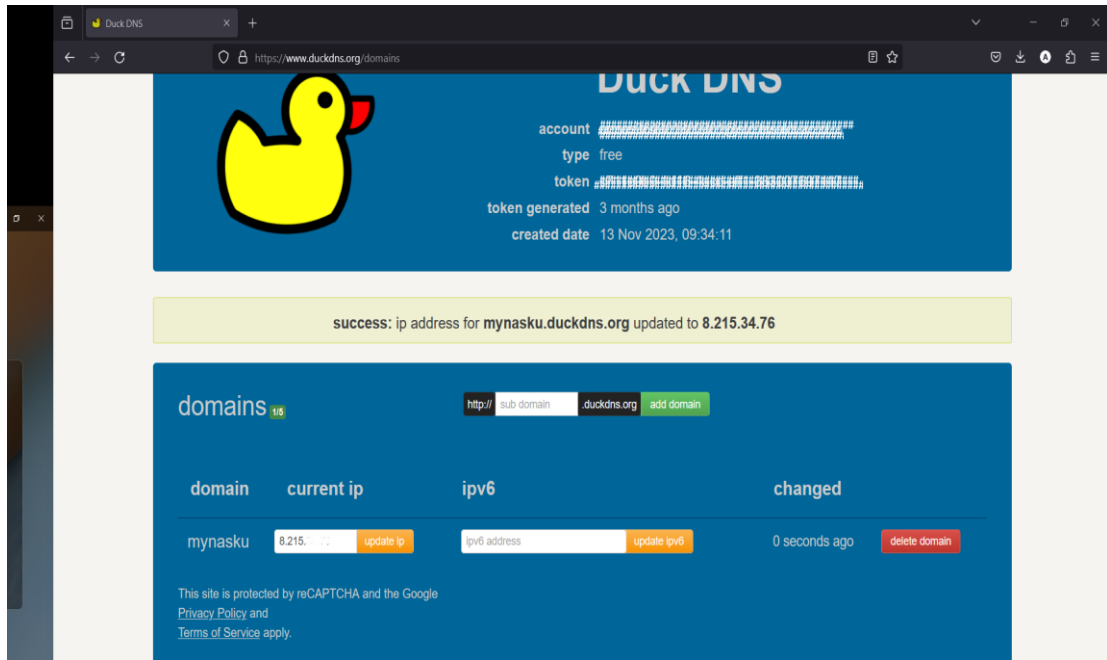
Name	Remote host	WAN Port	LAN IP Address	LAN Port	Protocol	Status	Options
http		80-80	192.168.8.15 0	81-81	TCP/UDP		 
https		443-443	192.168.8.15 0	443-443	TCP/UDP		 
SSH		2222-2222	192.168.8.15 0	22-22	TCP/UDP		 

Gambar 4 7Konfigurasi Port Forwarding

#### 4.1.5 Konfigurasi Dynamic Domain Name Services (DDNS)

1. Silakan kunjungi situs duckdns kemudian *login/register*
2. Setelah itu silakan lanjutkan konfigurasi ddns dengan mendaftarkan alamat domain yang diinginkan kemudian pastikan ip publik yang digunakan server NAS sudah terinput dalam konfigurasi pendaftaran alamat domain yang diinginkan. Berikut adalah konfigurasi duckdns pada implementasi ini:





Gambar 4 8 konfigurasi duckdns

3. Kemudian lanjut kan kembali konfigurasi ddns yang ada pada aplikasi pendukung *Nextcloud* yaitu *swag*, tambahkan detail-detail khusus sesuai dengan konfigurasi *duckdns*

#### 4.1.6 Pengujian

1. Pengujian fungsionalitas Raspberry Pi dengan mengakses ssh, web gui *openmediavault* dan *portainer*. Berikut adalah tahapan uji fungsionalitas untuk SSH, *OpenMediaVault*, dan Portainer pada Raspberry Pi:

##### A. Uji Fungsionalitas Raspberry Pi

Tujuan: Memastikan Raspberry Pi sudah berfungsi dengan baik.

Langkah-langkah:

- a. Dari komputer lain, buka terminal atau aplikasi SSH *client* (misalnya, PuTTY) dan ketik perintah berikut untuk menghubungkan ke Raspberry Pi:

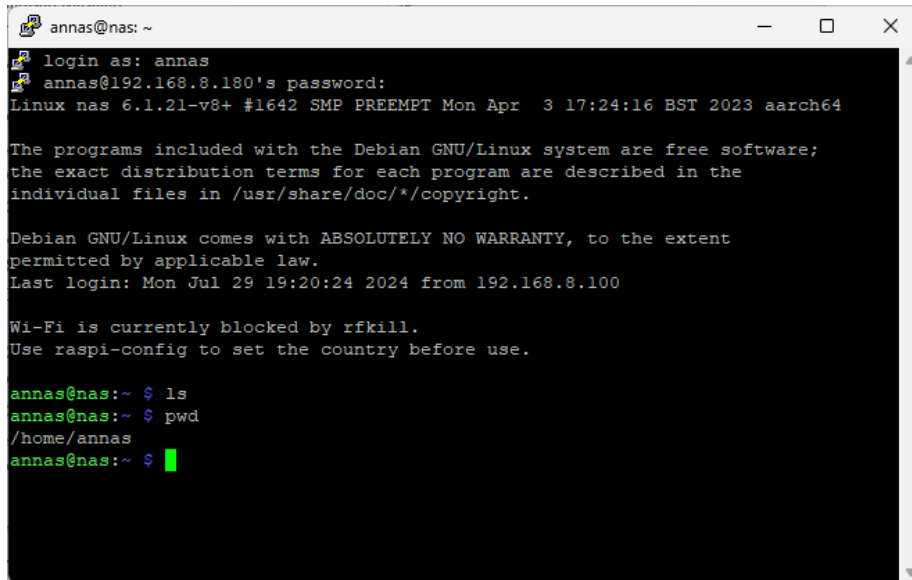
*Tabel 4 2 perintah berikut untuk menghubungkan ke Raspberry Pi*

(ganti `pi@raspberrypi` dengan username dan hostname atau alamat IP Raspberry Pi serta masukkan *password* jika diminta)  
`ssh pi@raspberrypi`

- b. Setelah berhasil login, jalankan beberapa perintah dasar untuk memastikan telah memiliki akses penuh, misalnya:

*Tabel 4 3 perintah dasar untuk memastikan telah memiliki akses penuh*

`ls`  
`pwd`



```
annas@nas: ~  
login as: annas  
annas@192.168.8.180's password:  
Linux nas 6.1.21-v8+ #1642 SMP PREEMPT Mon Apr 3 17:24:16 BST 2023 aarch64  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Mon Jul 29 19:20:24 2024 from 192.168.8.100  
Wi-Fi is currently blocked by rfkill.  
Use raspi-config to set the country before use.  
annas@nas:~ $ ls  
annas@nas:~ $ pwd  
/home/annas  
annas@nas:~ $
```

Gambar 4 9 Pengujian akses SSH

Hasil yang Diharapkan: Dapat terhubung ke Raspberry Pi melalui SSH dan menjalankan perintah dengan sukses.

## B. Uji Fungsionalitas *OpenMediaVault*

Tujuan: Memastikan *OpenMediaVault* berfungsi dengan baik dan dapat diakses melalui web interface.

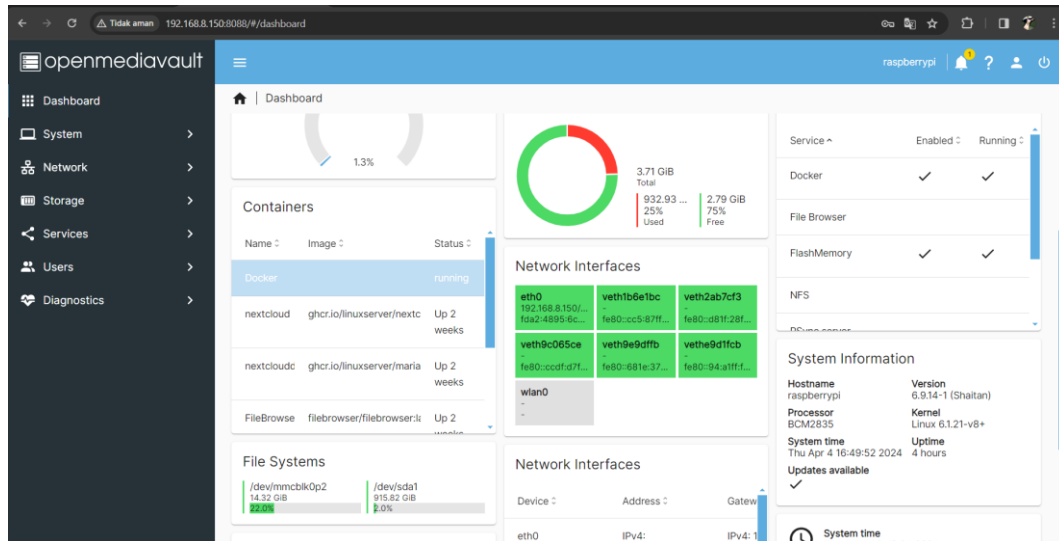
Langkah-langkah:

- a. Dari browser masukkan alamat IP Raspberry Pi diikuti dengan port 80 (default port), misalnya:

Tabel 4 4 memasukkan alamat IP Raspberry Pi (default port)

<a href="http://&lt;Raspberry Pi IP Address&gt;">http://&lt;Raspberry Pi IP Address&gt;</a>
---

- b. Login dengan kredensial default (*username: admin, password: openmediavault*).
- c. Verifikasi fungsi dasar jika telah berhasil *login*, seperti melihat status sistem pada halaman *dashboard*



Gambar 4 10 Dashboard Openmediavault Web GUI

Hasil yang Diharapkan: Dapat mengakses dan login ke web *interface* OpenMediaVault.

### C. Uji Fungsionalitas *Portainer*

Tujuan: Memastikan Portainer berfungsi dengan baik dan dapat diakses melalui web *interface*.

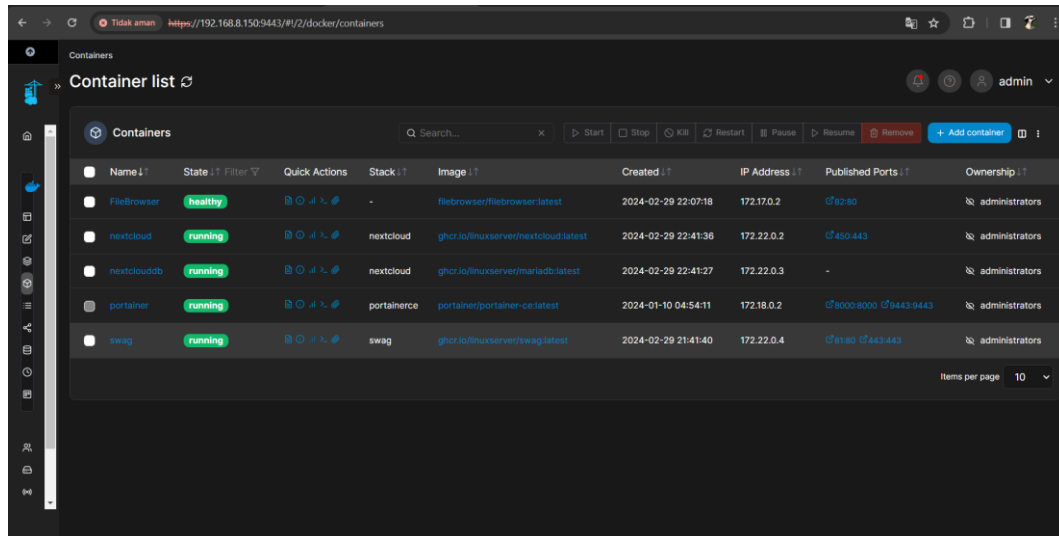
Langkah-langkah:

- Dari browser, masukkan alamat IP Raspberry Pi diikuti dengan port 9000 (default port), misalnya:

Tabel 4 5 perintah untuk memasukkan alamat IP Raspberry Pi

<http://<Raspberry Pi IP Address>:9000>

- Login dengan kredensial yang telah dibuat.
- Verifikasi fungsi dasar dengan melihat daftar *container* yang berjalan atau membuat dan menjalankan *container* baru.



Gambar 4 11 Daftar Container Portainer WebGUI

Hasil yang Diharapkan: Dapat mengakses dan login ke web interface Portainer, serta melihat dan mengelola container dengan sukses.

2. Pengujian fungsionalitas *port forwarding* dengan melakukan akses *port ssh* yang telah diteruskan oleh *router*.

#### A. Uji Fungsionalitas *Port Forwarding*

Tujuan: Memastikan bahwa *port forwarding* pada router berfungsi dengan baik dengan melakukan akses *port ssh* yang telah diteruskan oleh *router* dari jaringan eksternal.

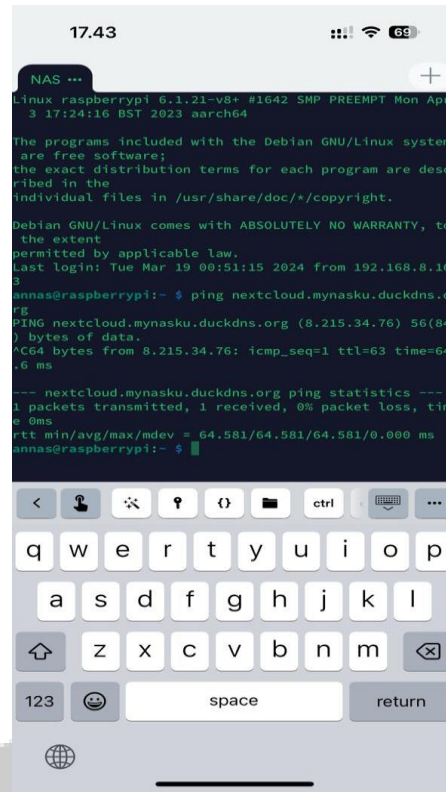
Langkah-langkah:

- a. Cari alamat IP *public* dengan situs seperti WhatIsMyIP.
- b. Lakukan perintah akses SSH menggunakan jaringan yang berbeda (misalnya, menggunakan data seluler atau Wi-Fi pada *smartphone*).
- c. Ketik perintah berikut untuk menghubungkan ke Raspberry Pi melalui alamat IP publik dan *port* yang telah diteruskan (ganti `<Public_IP>` dan `<External_Port>` dengan nilai yang sesuai):

Tabel 4 6 perintah untuk menghubungkan Raspberry Pi

```
ssh -p <External_Port> pi@<Public_IP>
```

d. Verifikasi dengan menjalankan perintah dasar.



*Gambar 4 12 Pengujian SSH via Smartphone*

Hasil yang Diharapkan: Dapat terhubung ke Raspberry Pi melalui SSH menggunakan alamat IP publik dan *port* yang telah diteruskan, serta menjalankan perintah dengan sukses.

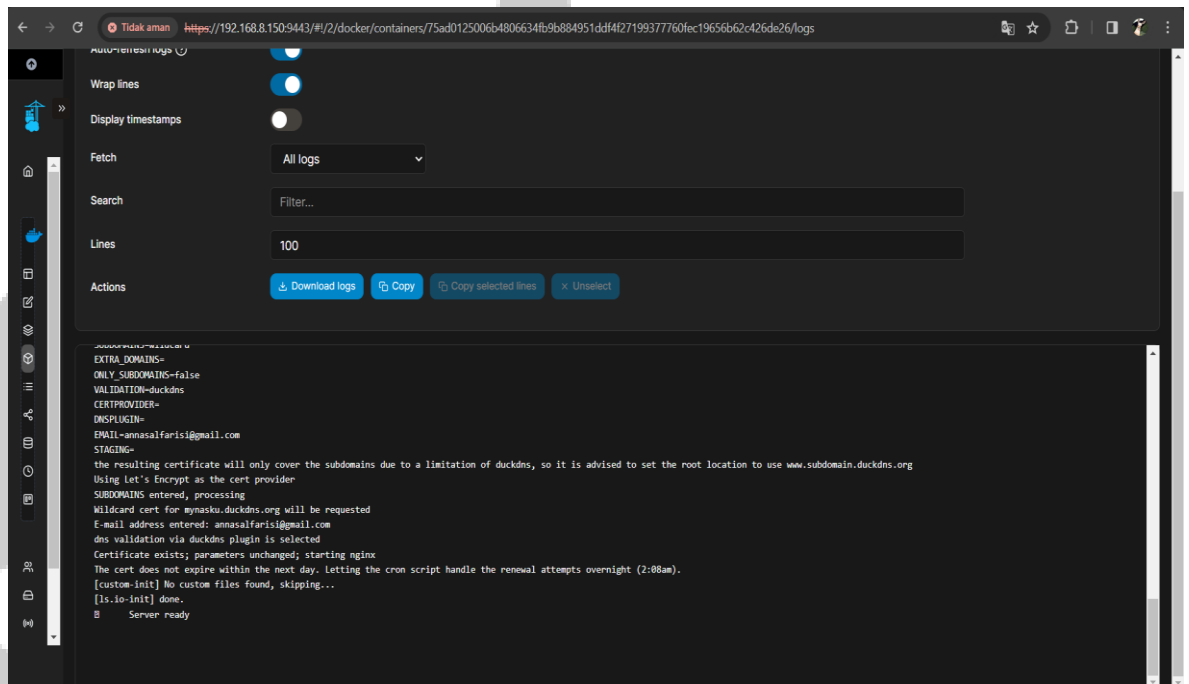
3. Pengujian fungsionalitas DDNS dengan menyambungkan swag dengan duckdns agar *Nextcloud* dapat diakses dengan domain.

#### A. Uji Fungsionalitas DDNS

Tujuan: Memastikan bahwa konfigurasi DDNS dengan DuckDNS dan SWAG berfungsi dengan baik sehingga *Nextcloud* dapat diakses melalui domain.

Langkah-langkah:

- a. Akses ke *Nextcloud* langsung dengan memasukkan domain yang telah dibuat di DuckDNS (<https://nextcloud.mynasku.duckdns.org>).
- b. Verifikasi dengan login ke *Nextcloud* atau melihat logs dari *container* SWAG.



Gambar 4 13 Pengujian konfigurasi DDNS

Hasil yang Diharapkan: Dapat mengakses *Nextcloud* melalui domain DuckDNS dan menggunakan semua fitur *Nextcloud* dengan lancar.

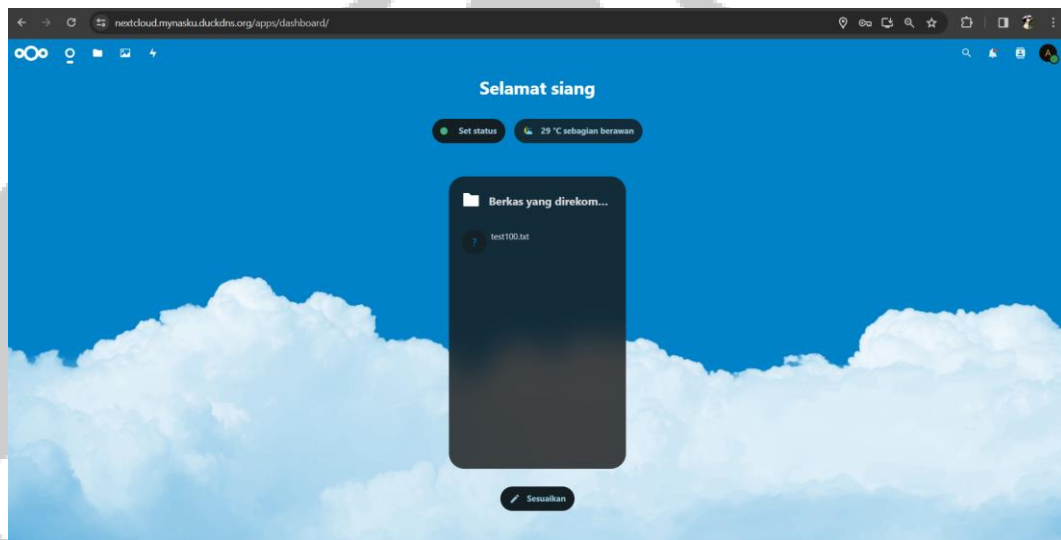
4. Proses pengujian fungsionalitas dilakukan secara manual dengan tujuan untuk menguji akses ke halaman login dari aplikasi *Nextcloud*. Pengujian dilakukan dengan tiga skenario, yakni menggunakan kredensial yang valid dan tidak valid dan fitur lupa *password* juga diuji guna memastikan fungsionalitas nya sesuai dengan yang diharapkan.

#### A. Uji Fungsionalitas Login dengan Kredensial Valid

Tujuan: Memastikan pengguna dapat login dengan menggunakan kredensial yang benar.

Langkah-langkah:

- a. Buka browser dan akses *Nextcloud* menggunakan URL yang sudah dikonfigurasi (<https://nextcloud.mynasku.duckdns.org>).
- b. Verifikasi akses dengan memasukkan kredensial yang valid untuk *login* *Nextcloud* dan pastikan diarahkan ke *dashboard*.



Gambar 4 14 Halaman Welcome Page Nextcloud

Hasil yang Diharapkan: Pengguna dapat *login* dengan sukses dan diarahkan ke halaman utama *Nextcloud*.

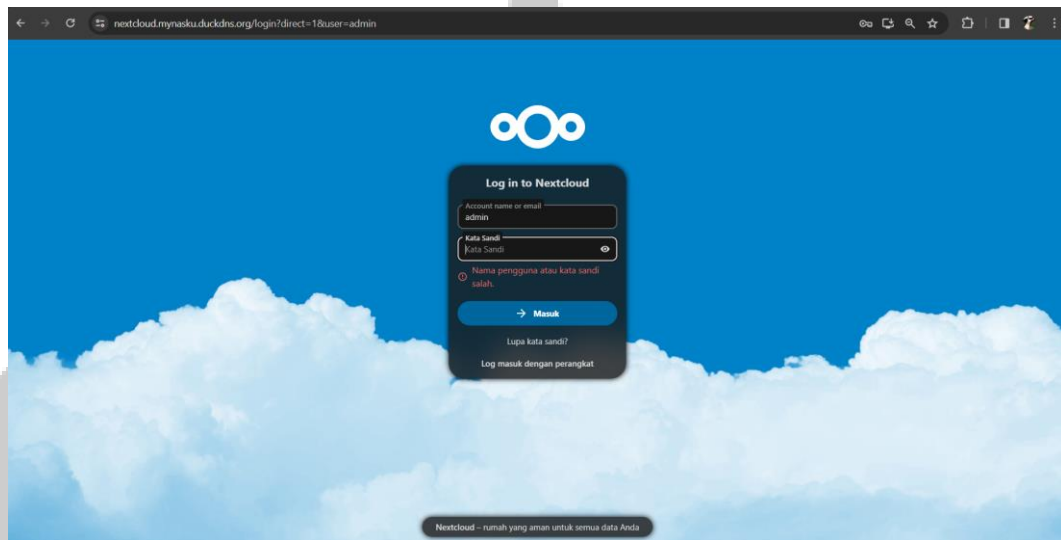


## B. Uji Fungsionalitas *Login* dengan Kredensial Tidak Valid

Tujuan: Memastikan pengguna dapat login dengan menggunakan kredensial yang benar.

Langkah-langkah:

- Buka *browser* dan akses *Nextcloud* menggunakan URL yang sudah dikonfigurasi (<https://nextcloud.mynasku.duckdns.org>).
- Verifikasi respon dengan memasukkan kredensial yang tidak valid untuk *login Nextcloud* dan pastikan sistem menampilkan pesan kesalahan yang sesuai, seperti "*Username atau password salah*".



Gambar 4 15 Halaman *Login Nextcloud* dengan pesan kesalahan.

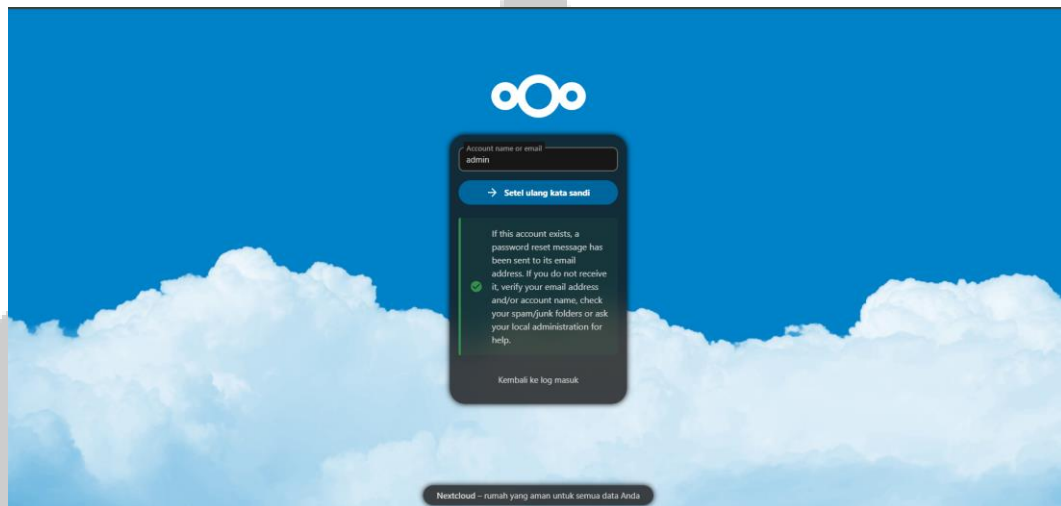
Hasil yang Diharapkan: Pengguna tidak dapat *login* dan sistem menampilkan pesan kesalahan yang sesuai.

### C. Uji Fungsionalitas *Login* dengan Kredensial Valid

Tujuan: Memastikan pengguna dapat login dengan menggunakan kredensial yang benar.

Langkah-langkah:

- Buka browser dan akses *Nextcloud* menggunakan URL yang sudah dikonfigurasi (<https://nextcloud.mynasku.duckdns.org>).
- Klik “Lupa Password” dan masukkan email yang terkait dengan akun *Nextcloud* lalu klik “Reset Password”.
- Verifikasi email yang telah dikirim oleh *Nextcloud* untuk *reset password* dan klik tautan untuk mengatur ulang *password*.



*Gambar 4 16 Halaman Forget Password*

Hasil yang Diharapkan: Pengguna menerima *email* untuk *reset password*, dapat mengatur ulang *password*, dan dapat *login* dengan *password* baru.

#### D. Uji Transfer Data *Nextcloud*

Tujuan: Menguji transfer data pada *Nextcloud* dengan ukuran file 10MB, 50MB, dan 100MB sebanyak 10 kali percobaan menggunakan *script* Python untuk mencatat durasi dan kecepatan transfer.

Langkah-langkah:

- Siapkan *file* berukuran 10MB, 50MB dan 100MB.
- Jalankan *script monitoring* dengan memasukkan perintah (python net\_speed\_monitor.py) dan tekan r untuk *resume* p untuk *pause* q untuk *quit*, berikut *source code* -nya:

Tabel 4 7 *script monitoring*

```
import time
import psutil
import threading
import keyboard

# Global flags to control the monitoring loop
running = True
paused = False

def get_net_info(interface="Wi-Fi"):
    """
    Retrieves network interface statistics for a specific interface
    (optional).

    Args:
        interface (str, optional): Name of the network interface to
        monitor. Defaults to "Ethernet".

    Returns:
        tuple: A tuple containing interface name (str), upload speed
        (Mbps), and download speed (Mbps),
        or None if the specified interface is not found.
    """
    try:
        net_io = psutil.net_io_counters(pernic=True)
        if interface in net_io:
```

```

        net_stats = net_io[interface]
        # Initial measurements
        net_in_1 = net_stats.bytes_recv
        net_out_1 = net_stats.bytes_sent

        time.sleep(1) # Pause for 1 second

        net_io = psutil.net_io_counters(pernic=True)
        net_stats = net_io[interface]
        # Final measurements
        net_in_2 = net_stats.bytes_recv
        net_out_2 = net_stats.bytes_sent

        # Calculate upload and download speeds (in Mbps)
        upload_speed = (net_out_2 - net_out_1) * 8 / (1024 * 1024)
        download_speed = (net_in_2 - net_in_1) * 8 / (1024 * 1024)

        return interface, upload_speed, download_speed
    else:
        print(f"Error: Interface '{interface}' not found.")
        return None, None, None
    except (psutil.NoSuchProcess, psutil.AccessDenied) as e:
        print(f"Error retrieving network data: {e}")
        return None, None, None

def main(sleep_interval=0):
    """
    Main function to monitor and log network speeds.

    Args:
        sleep_interval (int, optional): The number of seconds to wait
        between measurements. Defaults to 1.
    """
    global running, paused

    # Open file for writing data (append mode)
    with open("net_speed_data.csv", "a") as f:
        # Write header row (if file is empty)
        if f.tell() == 0:
            f.write("Timestamp,Interface,Upload Speed (Mbps),Download Speed (Mbps)\n")

    while running:

```

```

        if not paused:
            # Get current time
            current_time = time.strftime("%Y-%m-%d %H:%M:%S")

            # Get interface name, upload and download speeds
            interface, upload_speed, download_speed = get_net_info()

            if interface is not None:
                # Write data to the file

f.write(f"{current_time},{interface},{upload_speed:.2f},{download_speed
:.2f}\n")

                # Print data to the console with interface name
                print(f"[{current_time}]  Interface:  {interface},
Upload: {upload_speed:.2f} Mbps, Download: {download_speed:.2f} Mbps")

            # Wait for the specified sleep interval before the next
measurement
            time.sleep(sleep_interval)

def input_handler():
    global paused, running
    while running:
        if keyboard.is_pressed('p'):
            paused = True
            print("Paused monitoring.")
            while keyboard.is_pressed('p'):
                time.sleep(0.1)  # Wait for the key to be released to
avoid multiple triggers
        elif keyboard.is_pressed('r'):
            paused = False
            print("Resumed monitoring.")
            while keyboard.is_pressed('r'):
                time.sleep(0.1)
        elif keyboard.is_pressed('q'):
            running = False
            print("Exiting monitoring.")
            break
        time.sleep(0.1)  # Avoid high CPU usage

if __name__ == "__main__":
    # Start a separate thread to handle user input

```

```
input_thread = threading.Thread(target=input_handler)
input_thread.start()

# Run the main monitoring function
main()

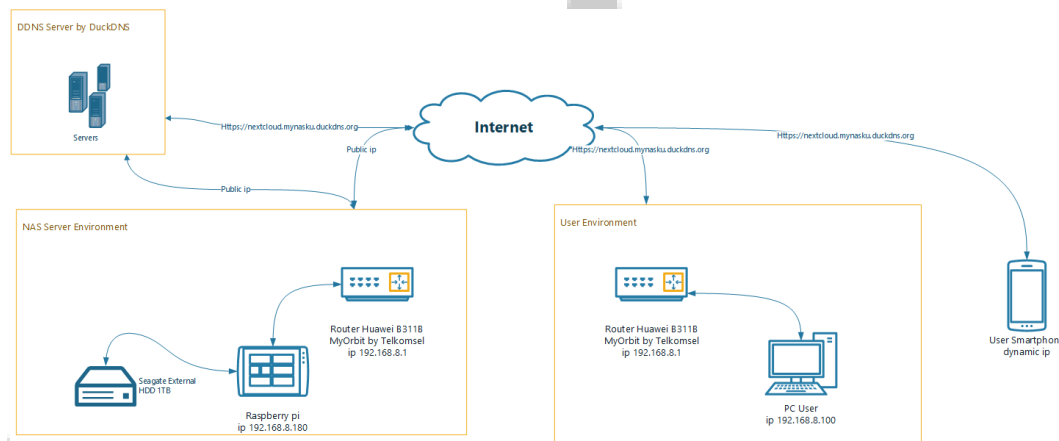
# Ensure all threads finish before exiting
input_thread.join()
```

- c. Unggah *file* masing-masing sebanyak 10 kali dimulai dari 10MB, 50MB dan 100MB secara bergantian dan catat waktu mulai dan waktu selesai dari tiap percobaan.
- d. Unduh *file* masing-masing sebanyak 10 kali dimulai dari 10MB, 50MB dan 100MB secara bergantian dan catat waktu mulai dan waktu selesai dari tiap percobaan.
- e. Hentikan *script monitoring* dengan menekan tombol q untuk menghentikannya.
- f. Data akan tersimpan dalam *file* `net_speed_data.csv`.
- g. Verifikasi data csv yang telah didapatkan dengan membandingkan pada hasil yang telah di *print* oleh *script monitoring* di *terminal*.
- h. Olah data yang diperoleh menjadi bentuk tabel setiap proses unggah dan unduh pada masing-masing ukuran *file*.
- i. Lalu sajikan dalam bentuk diagram batang agar lebih mudah visualisasinya.

## 4.2 Analisis Hasil Implementasi

### 4.2.1 Analisis Hasil Uji Kecepatan Transfer Data

Dalam konteks implementasi ini, pengujian dilakukan dengan skenario yang melibatkan aktivitas unggah dan unduh dengan tiga varian ukuran *file*, yakni 10MB, 50MB, dan 100MB. Hal ini bertujuan untuk memperoleh informasi mengenai kecepatan dan durasi yang dibutuhkan ketika pengguna melakukan aktivitas mengunduh dan mengunggah. Selanjutnya, setiap ukuran *file* akan diuji ulang sebanyak 10 kali guna memperoleh nilai kecepatan dan durasi rata-rata yang diperlukan untuk melakukan proses unggah dan unduh.



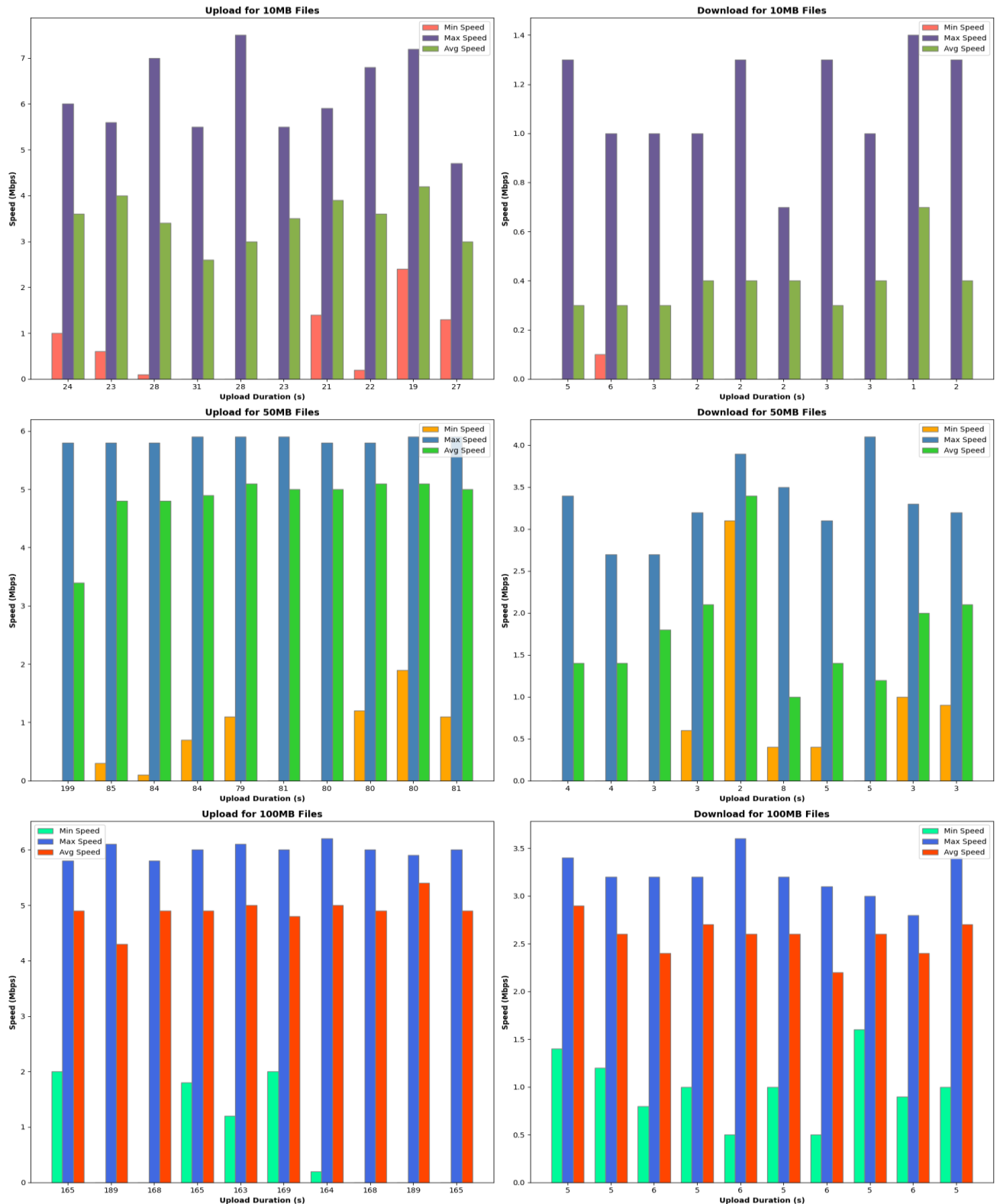
Gambar 4 17 Skema uji transfer data Nextcloud

Berikut adalah tabel dari hasil pengujian yang telah dijalankan:

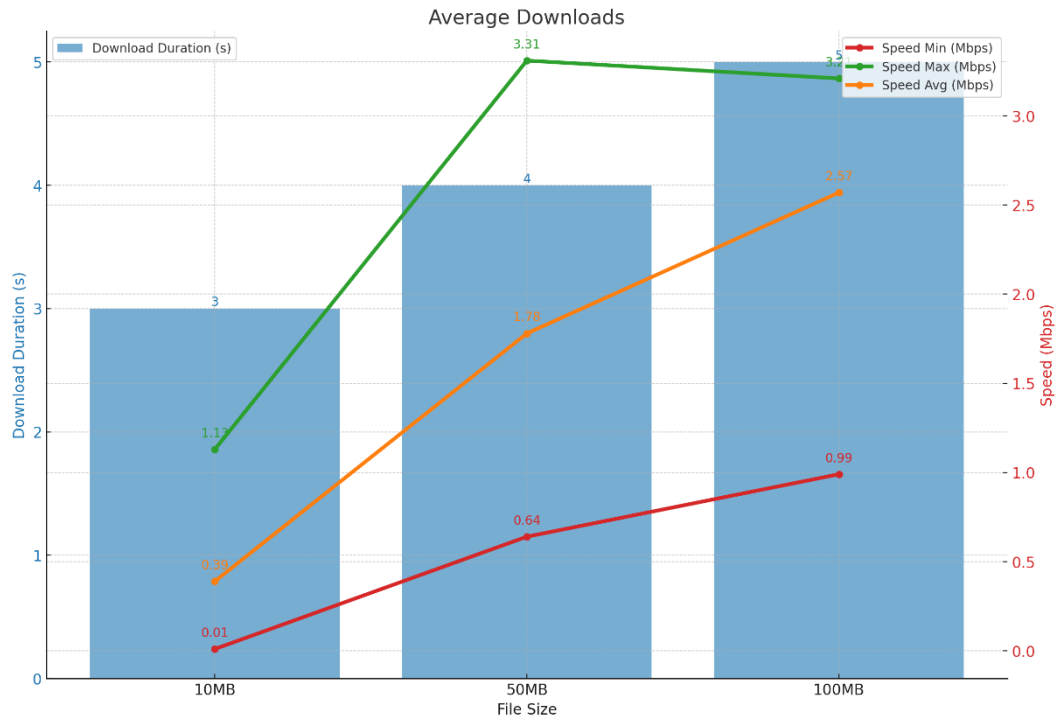
*Tabel 4 8 Hasil Uji Upload dan Download Network Attached Storage*

Upload 10MB					Download 10MB				
File Size	Upload Duration	Speed			File Size	Download Duration	Speed		
		Min	Max	Avg			Min	Max	Avg
10MB	24 s	1,0 Mbps	6,0 Mbps	3,6 Mbps	10MB	5 s	0,00 Mbps	1,30 Mbps	0,30 Mbps
10MB	23 s	0,6 Mbps	5,6 Mbps	4 Mbps	10MB	6 s	0,10 Mbps	1,00 Mbps	0,30 Mbps
10MB	28 s	0,1 Mbps	7,0 Mbps	3,4 Mbps	10MB	3 s	0,00 Mbps	1,00 Mbps	0,30 Mbps
10MB	31 s	0,0 Mbps	5,5 Mbps	2,6 Mbps	10MB	2 s	0,00 Mbps	1,00 Mbps	0,40 Mbps
10MB	28 s	0,0 Mbps	7,5 Mbps	3,0 Mbps	10MB	2 s	0,00 Mbps	1,30 Mbps	0,40 Mbps
10MB	23 s	0,0 Mbps	5,5 Mbps	3,5 Mbps	10MB	2 s	0,00 Mbps	0,70 Mbps	0,40 Mbps
10MB	21 s	1,4 Mbps	5,9 Mbps	3,9 Mbps	10MB	3 s	0,00 Mbps	1,30 Mbps	0,30 Mbps
10MB	22 s	0,2 Mbps	6,8 Mbps	3,6 Mbps	10MB	3 s	0,00 Mbps	1,00 Mbps	0,40 Mbps
10MB	19 s	2,4 Mbps	7,2 Mbps	4,2 Mbps	10MB	1 s	0,00 Mbps	1,40 Mbps	0,70 Mbps
10MB	27 s	1,3 Mbps	4,7 Mbps	3,0 Mbps	10MB	2 s	0,00 Mbps	1,30 Mbps	0,40 Mbps
Upload 50MB					Download 50MB				
File Size	Upload Duration	Speed			File Size	Download Duration	Speed		
		Min	Max	Avg			Min	Max	Avg
50MB	199 s	0,00 Mbps	5,80 Mbps	3,40 Mbps	50MB	4 s	0,00 Mbps	3,40 Mbps	1,40 Mbps
50MB	85 s	0,30 Mbps	5,80 Mbps	4,80 Mbps	50MB	4 s	0,00 Mbps	2,70 Mbps	1,40 Mbps
50MB	84 s	0,10 Mbps	5,80 Mbps	4,80 Mbps	50MB	3 s	0,00 Mbps	2,70 Mbps	1,80 Mbps
50MB	84 s	0,70 Mbps	5,90 Mbps	4,90 Mbps	50MB	3 s	0,60 Mbps	3,20 Mbps	2,10 Mbps
50MB	79 s	1,10 Mbps	5,90 Mbps	5,10 Mbps	50MB	2 s	3,10 Mbps	3,90 Mbps	3,40 Mbps
50MB	81 s	0,00 Mbps	5,90 Mbps	5,00 Mbps	50MB	8 s	0,40 Mbps	3,50 Mbps	1,00 Mbps
50MB	80 s	0,00 Mbps	5,80 Mbps	5,00 Mbps	50MB	5 s	0,40 Mbps	3,10 Mbps	1,40 Mbps
50MB	80 s	1,20 Mbps	5,80 Mbps	5,10 Mbps	50MB	5 s	0,00 Mbps	4,10 Mbps	1,20 Mbps
50MB	80 s	1,90 Mbps	5,90 Mbps	5,10 Mbps	50MB	3 s	1,00 Mbps	3,30 Mbps	2,00 Mbps
50MB	81 s	1,10 Mbps	5,90 Mbps	5,00 Mbps	50MB	3 s	0,90 Mbps	3,20 Mbps	2,10 Mbps
Upload 100MB					Download 100MB				
File Size	Upload Duration	Speed			File Size	Download Duration	Speed		
		Min	Max	Avg			Min	Max	Avg
100MB	165 s	2,00 Mbps	5,80 Mbps	4,90 Mbps	100MB	5 s	1,40 Mbps	3,40 Mbps	2,90 Mbps
100MB	189 s	0,00 Mbps	6,10 Mbps	4,30 Mbps	100MB	5 s	1,20 Mbps	3,20 Mbps	2,60 Mbps
100MB	168 s	0,00 Mbps	5,80 Mbps	4,90 Mbps	100MB	6 s	0,80 Mbps	3,20 Mbps	2,40 Mbps
100MB	165 s	1,80 Mbps	6,00 Mbps	4,90 Mbps	100MB	5 s	1,00 Mbps	3,20 Mbps	2,70 Mbps
100MB	163 s	1,20 Mbps	6,10 Mbps	5,00 Mbps	100MB	6 s	0,50 Mbps	3,60 Mbps	2,60 Mbps
100MB	169 s	2,00 Mbps	6,00 Mbps	4,80 Mbps	100MB	5 s	1,00 Mbps	3,20 Mbps	2,60 Mbps
100MB	164 s	0,20 Mbps	6,20 Mbps	5,00 Mbps	100MB	6 s	0,50 Mbps	3,10 Mbps	2,20 Mbps
100MB	168 s	0,00 Mbps	6,00 Mbps	4,90 Mbps	100MB	5 s	1,60 Mbps	3,00 Mbps	2,60 Mbps
100MB	189 s	0,00 Mbps	5,90 Mbps	5,40 Mbps	100MB	6 s	0,90 Mbps	2,80 Mbps	2,40 Mbps
100MB	165 s	0,00 Mbps	6,00 Mbps	4,90 Mbps	100MB	5 s	1,00 Mbps	3,40 Mbps	2,70 Mbps

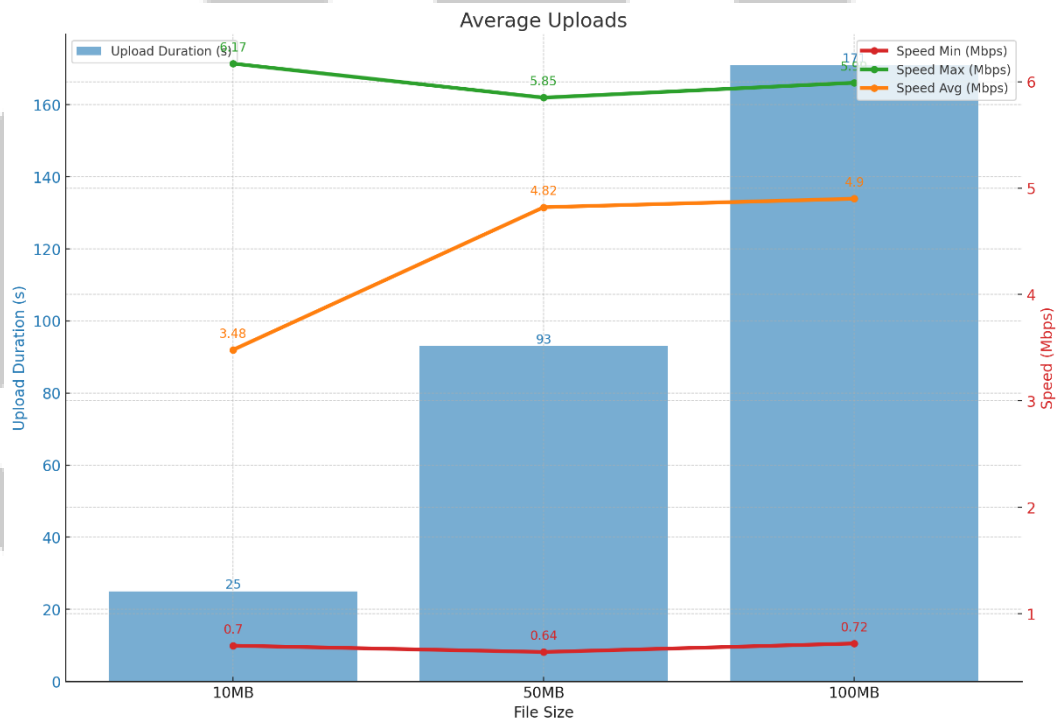




Gambar 4 18 Diagram batang Hasil Uji Upload dan Download Network Attached Storage



Gambar 4 19 Diagram Average Download



Gambar 4 20 Diagram Average Upload

Kedua diagram yaitu Average Download dan Average Upload menunjukkan bahwa durasi unduh dan unggah meningkat seiring dengan

ukuran *file*, namun peningkatan durasi unggah lebih signifikan. Durasi unduh berkisar dari 3 detik untuk *file* 10MB hingga 5 detik untuk *file* 100MB, sementara durasi unggah meningkat dari 25 detik untuk *file* 10MB hingga 160 detik untuk *file* 100MB. Kecepatan unduh rata-rata meningkat dari 1,13 Mbps (10MB) menjadi 2,57 Mbps (100MB), sedangkan kecepatan unggah rata-rata meningkat dari 3,48 Mbps (10MB) menjadi 4,9 Mbps (100MB). Kecepatan maksimum unduh dan unggah masing-masing mencapai 3,31 Mbps dan 5,85 Mbps pada *file* 50MB.

#### 4.2.2 Analisis Fungsionalitas

Berdasarkan hasil analisis fungsionalitas pada sebuah akun pengguna *Nextcloud* dalam implementasi ini, diketahui bahwa pengguna memiliki kemampuan untuk mengaktifkan fitur keamanan, yakni otentikasi tanpa kata sandi yang telah mengikuti standar FIDO2. Selain itu, pengguna juga dapat mengatur perangkat dan sesi yang telah dijalankan, sehingga jika terdapat perangkat yang tidak terdaftar pada salah satu akun pengguna, pengguna dapat langsung mengeluarkannya. Selanjutnya, pengguna juga dapat mengaktifkan otentikasi dua langkah dengan konfigurasi sesuai keinginan. Terakhir, terdapat fitur keamanan lainnya, yaitu enkripsi server-side, yang merupakan proses mengenkripsi data di sisi server sebelum data tersebut disimpan. Hal ini berarti proses enkripsi dilakukan pada server tempat data disimpan, bukan pada perangkat pengguna.

Dengan mengacu pada hasil analisis fungsionalitas, implementasi ini menunjukkan bahwa terdapat berbagai fitur keamanan yang telah diimplementasikan untuk meningkatkan keamanan akun pengguna *Nextcloud*. Dengan adanya fitur-fitur tersebut, keamanan data pengguna dapat lebih terjamin dalam penggunaan *platform Nextcloud*.

*Tabel 4 9 Hasil Uji Fungsionalitas SSH*

Hasil Pengujian Fungsionalitas SSH		
No	Langkah	Hasil
1	Aktifkan SSH	Berhasil
2	Koneksikan ke Raspberry Pi melalui SSH	Berhasil
3	Verifikasi Koneksi	Perintah ls dan pwd berhasil

*Tabel 4 10 Hasil Uji Fungsionalitas OMV*

Hasil Pengujian Fungsionalitas OpenMediaVault (OMV)		
No	Langkah	Hasil
1	Akses Web Interface OMV	Berhasil
2	Login ke OMV	Berhasil
3	Verifikasi Fungsi Dasar	Dashboard, konfigurasi penyimpanan, pengguna baru berhasil diverifikasi
4	Uji Akses Berbagi File	Shared folder berhasil diakses dan diuji

*Tabel 4 11 Hasil Uji Fungsionalitas Portainer*

Hasil Pengujian Fungsionalitas Portainer		
No	Langkah	Hasil
1	Akses Web Interface Portainer	Berhasil
2	Login atau Setup Awal	Berhasil
3	Verifikasi Fungsi Dasar	Daftar container, volume, dan jaringan berhasil diverifikasi

*Tabel 4 12 Hasil Uji Fungsionalitas Port Forwarding Untuk Akses SSH*

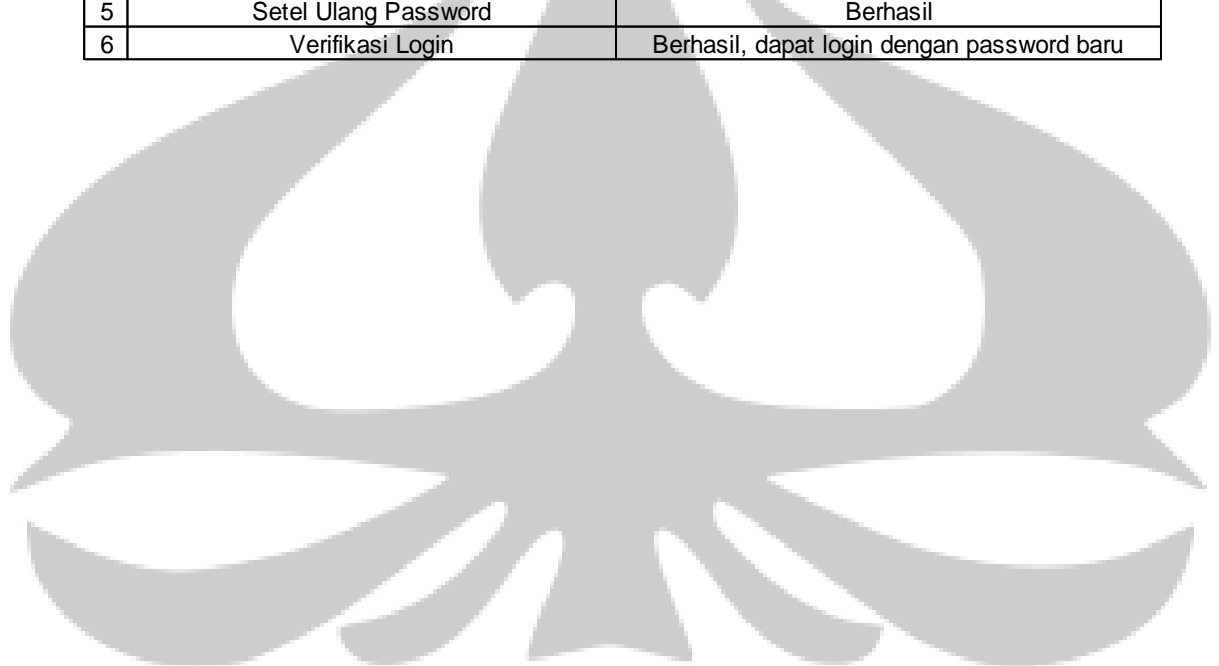
Hasil Pengujian Fungsionalitas Port Forwarding untuk Akses SSH		
No	Langkah	Hasil
1	Konfigurasi Router	Berhasil
2	Verifikasi Konfigurasi Port Forwarding	Berhasil
3	Verifikasi Koneksi dan Fungsionalitas SSH	Login dan perintah ls dan pwd berhasil

*Tabel 4 13 Hasil Uji Fungsionalitas DDNS dengan DuckDNS dan SWAG untuk Akses NextCloud*

Hasil Pengujian Fungsionalitas DDNS dengan DuckDNS dan SWAG untuk Akses Nextcloud		
No	Langkah	Hasil
1	Konfigurasi DuckDNS	Berhasil
2	Konfigurasi SWAG	Berhasil
3	Konfigurasi Port Forwarding pada Router	Berhasil
4	Uji Akses Nextcloud melalui Domain	Berhasil, Nextcloud dapat diakses dengan lancar

*Tabel 4 14 Hasil Uji Fungsionalitas NextCloud, Kredensial Tidak Valid dan fitur Lupa Password*

Hasil Pengujian Nextcloud		
Login dengan Kredensial Valid		
No	Langkah	Hasil
1	Buka Halaman Login	Berhasil
2	Masukkan Kredensial Valid	Berhasil
3	Verifikasi Akses	Berhasil, diarahkan ke dashboard
Login dengan Kredensial Tidak Valid		
No	Langkah	Hasil
1	Buka Halaman Login	Berhasil
2	Masukkan Kredensial Tidak Valid	Berhasil
3	Verifikasi Respons	Berhasil, sistem menampilkan pesan kesalahan
Fitur Lupa Password		
No	Langkah	Hasil
1	Buka Halaman Login	Berhasil
2	Klik "Lupa Password"	Berhasil
3	Masukkan Email	Berhasil
4	Verifikasi Email	Berhasil, email reset password diterima
5	Setel Ulang Password	Berhasil
6	Verifikasi Login	Berhasil, dapat login dengan password baru



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Penelitian tentang implementasi *Network Attached Storage (NAS)* dengan teknologi *Port Forwarding* dan *Dynamic Domain Name Service (DDNS)* di PT. XYZ menghasilkan beberapa kesimpulan penting terkait keberhasilan implementasi dan kinerja sistem:

1. **Keberhasilan Implementasi NAS:** Implementasi NAS dengan teknologi *Port Forwarding* dan DDNS berhasil dilakukan dengan baik. NAS ini menyediakan solusi penyimpanan data yang bisa diakses darimana saja dan kapan saja. Namun dalam proses implementasi masih terdapat beberapa bagian yang tidak dapat dikontrol yaitu trafik dari penyedia layanan dan kebijakan *bandwidth* mereka. Sebelum implementasi ini dieksekusi, penulis sudah terlebih dahulu menghubungi penyedia layanan untuk mendapatkan informasi dan dijawab bahwa kecepatan yang diklaim oleh penyedia layanan adalah *Up to 15Mbps* dengan perbandingan 1:1. Selama proses pengujian implementasi ini berjalan ternyata klaim tersebut tidak sepenuhnya menjadi nyata karena data yang telah didapatkan dari hasil pengujian tidak bisa membuktikan klaim dari penyedia layanan, tetapi hasil pengujian juga telah membuktikan bahwa implementasi ini berhasil dieksekusi dengan NAS mampu diakses dari internet dan mampu mejadi solusi penyimpanan data di PT XYZ...
2. **Fungsionalitas dan Performa:** Pengujian menunjukkan bahwa NAS yang diimplementasikan memiliki fungsionalitas yang baik dan performa transfer data yang memadai. Pengujian meliputi akses web GUI, *port*, *domain*, *login*, dan fitur lupa *password*, yang semuanya berjalan lancar. Uji performa jaringan dengan berbagai ukuran *file* (10MB, 50MB, 100MB) menunjukkan kecepatan

transfer yang memadai. Secara keseluruhan, hasil uji menunjukkan bahwa baik durasi unduh maupun unggah meningkat dengan ukuran *file*. Namun, peningkatan durasi unggah jauh lebih signifikan dibandingkan dengan durasi unduh. Kecepatan unduh dan unggah rata-rata juga meningkat seiring dengan ukuran *file*, dengan kecepatan unggah rata-rata cenderung lebih tinggi dibandingkan dengan kecepatan unduh rata-rata.

## 5.2 Saran

Berdasarkan kesimpulan, ada beberapa saran yang perlu dipertimbangkan untuk meningkatkan implementasi NAS di PT. XYZ. Pertama, penting untuk mengoptimalkan *bandwidth* dan trafik dengan bekerja sama lebih erat dengan penyedia layanan, memastikan *bandwidth* yang dijanjikan terpenuhi, dan mempertimbangkan peningkatan paket layanan jika diperlukan. Selain itu, *monitoring* trafik secara *real-time* dan pengujian performa berkala harus dilakukan untuk memastikan kestabilan dan keandalan NAS, terutama untuk ukuran *file* yang lebih besar. Memberikan pelatihan kepada pengguna tentang cara penggunaan NAS yang efisien dan langkah-langkah yang harus diambil jika terjadi masalah teknis juga sangat penting. Terakhir, dokumentasi lengkap mengenai konfigurasi dan penggunaan NAS harus disusun, serta *feedback* dari pengguna dikumpulkan dan dievaluasi untuk perbaikan berkelanjutan. Dengan mengikuti saran-saran ini, PT. XYZ dapat meningkatkan kinerja dan keandalan sistem NAS serta memastikan penggunaan yang efektif dan efisien..