

Implementation of Samba Server Using OpenVPN Based on Single Board Computer (SBC) for Private Cloud Storage

Dwi Bayu Putra Pamungkas^{1*}, Isnawaty^{2*}, L.M. Fid Aksara^{3*}

* Program Studi Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo
Jalan H.E.A. Mokodompit, Kota Kendari, 93232, Sulawesi Tenggara, Indonesia
dwibayu.putrapamungkas99@gmail.com¹

Article Info

Article history:

Received 2024-08-15

Revised 2024-08-24

Accepted 2024-08-27

Keyword:

Raspberry Pi,
Samba,
OpenVPN,
Server,
Cloud Storage.

ABSTRACT

In the current digital era, people's need for data storage media that is practical and can be accessed at any time is increasing. This research aims to design and implement a practical cloud-based data storage system using a Raspberry Pi 4 Model B device using the Samba and OpenVPN applications. The system focuses on storing users' data (private cloud), which allows users to directly access files and data via a storage server. The method used in this research includes a literature review to support system development. Testing was carried out to evaluate the security of the system being built by comparing access to private cloud server services before and after using the OpenVPN application. Test results show that using the OpenVPN application increases the security of data exchange, with good encryption in communications between client and server. The resulting system runs according to the initial design and can function as a secure private cloud system. This research can contribute to the development of efficient and secure data storage solutions, as well as show the potential for using the Raspberry Pi as an energy and cost-saving personal cloud server device.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Pada era modern saat ini khususnya pada bidang teknologi informasi, media penyimpanan data merupakan sebuah bagian yang sangat diperlukan, karena saat ini data ataupun informasi yang dikirim serta diterima semakin banyak seiring dengan berkembangnya berbagai macam perangkat telekomunikasi. Semakin banyak data dan informasi, maka semakin besar juga tempat penyimpanan data yang dibutuhkan. Pengguna menginginkan sebuah media penyimpanan data yang praktis dan bisa diakses langsung kapan saja tanpa perlu membawa sebuah perangkat penyimpanan.

Dari pernyataan tersebut pengguna bisa memanfaatkan teknologi penyimpanan *cloud computing*. *Cloud computing* tercipta karena timbulnya permasalahan pada kegiatan komputasi yang berhubungan dengan keterbatasan sumber daya penyimpanan pada sebuah komputer tunggal [1]. *Cloud computing* merupakan suatu layanan teknologi informasi yang berjalan pada jaringan internet [2]. Media penyimpanan

cloud menjadi sebuah solusi keterbatasan *resource* media penyimpanan fisik yang membuat pengguna lebih fleksibel dalam mengakses data, sehingga pengguna lebih memilih penyedia jasa penyimpanan data berbasis *cloud*. Namun, pihak penyedia jasa memberikan batasan kapasitas penyimpanan kepada pengguna untuk menyimpan data mereka pada *server* yang mereka miliki apalagi layanan *cloud* yang dipilih pengguna bersifat gratis [3].

Terdapat beberapa jenis teknologi *cloud* yang umum digunakan, yaitu *public cloud* dan *private cloud*. Untuk teknologi *public cloud* memiliki sifat yang *multitenancy*, karena dikhususkan untuk melayani banyak pengguna, bukan hanya satu pelanggan, sedangkan *private cloud* dikhususkan bekerja pada infrastruktur pribadi dan menyediakan penggunaanya dengan sumber daya komputasi yang dinamis serta hanya dapat diakses oleh pengguna yang berwenang. Teknologi *cloud* dihadirkan menjadi upaya untuk memungkinkan pengguna dapat mengakses data, sumber daya, dan perangkat dari mana saja melalui jaringan internet.

Penelitian sebelumnya mengatakan bahwa untuk membangun sebuah *server* penyimpanan *cloud* diperlukan biaya yang tidak sedikit dan juga konsumsi listrik yang besar sehingga banyak pengguna layanan penyimpanan data digital *online* beralih ke *server privat cloud* mereka sendiri. Dengan menggunakan perangkat Raspberry Pi dan aplikasi *Owncloud* pengguna dapat membuat *server privat cloud* mereka sendiri dengan daya yang rendah, biaya yang rendah dan fleksibilitas dalam pemilihan kapasitas penyimpanan [3].

Selain itu kekhawatiran tentang kerentanan keamanan data dari layanan penyimpanan *cloud*. Data yang disimpan di *cloud* dapat diakses oleh penyedia layanan, yang menimbulkan risiko pelanggaran data dan kehilangan privasi. Dalam penelitiannya mendapatkan solusi penyimpanan *cloud* yang lebih aman, di mana pengguna memiliki kontrol penuh atas data mereka dengan memanfaatkan perangkat Raspberry Pi dan aplikasi *Nextcloud* sebagai *server cloud* [4].

Berdasarkan permasalahan yang ada dan beberapa penelitian terdahulu, pada penelitian ini akan kembali membahas mengenai pengimplementasian media penyimpanan *cloud* dengan menggunakan sebuah perangkat Raspberry Pi yang akan digunakan sebagai *server* utamanya. Perkembangan layanan *cloud* dengan menerapkan sistem berlangganan sebuah tempat penyimpanan data berharga tinggi dapat diminimalisir secara signifikan dengan memanfaatkan *resource* kecil, dalam hal ini menggunakan perangkat Raspberry Pi [5]. Peneliti menggunakan perangkat Raspberry Pi 4 Model B sebagai perangkat *server* dan sistem operasi yang digunakan adalah Raspberry Pi OS. Raspberry Pi merupakan salah satu *resource* CPU yang sangat *powerfull* untuk membangun sebuah *server* karena dukungan sistem operasi GNU Linux [5].

Jenis sistem *cloud* yang akan dibangun pada penelitian ini adalah sistem *private cloud*. *Private cloud* diimplementasikan untuk memenuhi kebutuhan tertentu secara khusus [6]. Yang berarti pada penelitian ini, peneliti hanya memfokuskan untuk penggunaan dan penyimpanan data pribadi pengguna saja dan memungkinkan pengguna dapat mengakses *file* dan data mereka secara langsung pada *server* penyimpanan.

Namun, untuk membuat sistem *private cloud*, peneliti memanfaatkan layanan yang berbeda dari penelitian-penelitian yang sudah ada yaitu layanan *open source Samba* yang berfungsi sebagai program untuk *sharing file* antara pengguna dan *server*. Kemudian ditambahkan dengan protokol keamanan pada *server private cloud* tersebut, dalam hal ini peneliti menggunakan layanan *Virtual Private Network* (VPN) yaitu layanan *open source OpenVPN* yang dimanfaatkan sebagai protokol keamanan pada *server*, karena menggunakan enkripsi kriptografi *Secure Sockets Layer/Transport Layer Security* (SSL/TLS) untuk mengamankan komunikasi *online* dan pertukaran data dengan menggunakan kunci, *username*, *password*, serta sertifikat otentikasi.

II. METODE PENELITIAN

A. Analisis Kebutuhan Alat dan Bahan

Dalam penelitian ini, peneliti menggunakan perangkat Raspberry Pi 4 Model B sebagai perangkat *server* utama yang menjalankan sistem operasi dan aplikasi yang diperlukan untuk penyimpanan *cloud*, dengan dukungan *harddisk eksternal* sebagai media penyimpanan untuk menyimpan data dan *file* pengguna. Untuk menyediakan koneksi internet yang memungkinkan pengguna mengakses *server* dari jarak jauh, dibutuhkan perangkat *access point*. Sistem operasi Raspberry Pi OS diinstal pada Raspberry Pi untuk menjalankan aplikasi *Samba*, yang berfungsi untuk melakukan *sharing file* antara *server* dan *client*, dan memungkinkan akses data dari berbagai perangkat. Keamanan komunikasi antara *klien* dan *server* tetap terjaga dengan menggunakan aplikasi *OpenVPN* yang mengenkripsi data.

Untuk menganalisis dan menangkap paket data yang dikirim dan diterima, serta untuk menguji keamanan sistem sebelum dan setelah penggunaan aplikasi *OpenVPN*, digunakan aplikasi *Wireshark*. Seluruh instalasi dan konfigurasi aplikasi *Samba* dan *OpenVPN* pada Raspberry Pi dilakukan melalui *PuTTY* yang merupakan aplikasi emulator terminal sebagai tools pengembangan utama.

B. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan adalah kajian pustaka atau studi literatur. Metode pengumpulan data ini menjadi pendukung penelitian dan dijadikan sebagai referensi, mulai dari buku, jurnal, skripsi maupun sumber bacaan *softcopy* yang diperoleh dari internet dan juga sumber-sumber lain. Metode ini dilaksanakan dengan melakukan studi kepustakaan yang relevan untuk mencari sumber pelengkap yang berhubungan, sehingga bisa diimplementasikan pada sistem yang akan dibangun. Beberapa studi kepustakaan yang berkaitan dengan penelitian ini adalah sebagai berikut:

- 1) *Jaringan Komputer*: Setiap perangkat yang terhubung dalam jaringan komputer dapat meminta layanan dan mengirim layanan [7]. Pihak yang meminta atau menerima layanan disebut *client* dan yang memberikan atau mengirim layanan yang diminta oleh *client* disebut sebagai *server*. Model sistem seperti ini disebut dengan sistem *client-server* dan banyak digunakan pada seluruh jaringan komputer, misalnya pada *cloud computing* [8].
- 2) *Cloud Storage*: *Cloud storage* menerapkan metode kapabilitas yang terkait tentang teknologi informasi yang disajikan sebagai suatu layanan sehingga membuat pengguna dapat mengaksesnya menggunakan internet [9].
- 3) *Samba*: *Samba* menggunakan protokol *Service Message Block* (SMB) yang digunakan untuk melakukan *sharing file* dan data dari berbagai perangkat agar dapat digunakan secara bersama-sama oleh banyak pengguna dan bisa dijalankan pada *Windows* [10]. Program *Samba*

sangat kompatibel pada sistem *Linux* dan juga bisa digunakan pada sistem *Windows*. Dengan menggunakan *Samba* semua fitur bisa didapatkan dengan lebih murah dan juga hemat [11].

4) *OpenVPN*: *OpenVPN* dapat membuat koneksi *Point-to-Point Tunneling Protocol* (PPTP) yang telah terenkripsi dengan baik, karena menggunakan kunci *private*, sertifikat autentikasi, *username* dan juga *password* untuk membangun koneksinya. Untuk proses enkripsi *OpenVPN* menggunakan *Secure Sockets Layer/Transport Layer Security* (SSL/TLS) yang dimana pembuatan sertifikat SSL-nya dilakukan oleh *OpenSSL* yang telah disediakan oleh *Linux* [12]. Dengan menggunakan *OpenVPN* maka pertukaran data melalui jaringan publik akan terjamin keamanannya dikarenakan adanya sistem enkripsi data dan juga menggunakan teknologi *tunneling* antara *VPN client* dengan *VPN server*. Aplikasi *OpenVPN* tidak berbasis browser, tetapi aplikasi ini perlu dipasang pada komputer *server* maupun komputer *client* [13].

5) *Raspberry Pi*: *Raspberry Pi* memiliki perangkat lunak bawaan yang memungkinkan penggunaanya untuk memprogram dan mendesain animasi, *game*, video hingga mengembangkan *script* atau kode program [14].

C. Metode Pengembangan Sistem

Metode yang digunakan untuk pengembangan sistem adalah metode *Rational Unified Process* (RUP). Metode RUP menyediakan pendekatan untuk menentukan tugas dan tanggung jawab dalam pengembangan suatu organisasi, dengan tujuan untuk memastikan produksi kualitas tinggi, *software* memenuhi dengan kebutuhan pengguna sesuai dengan jadwal dan biaya yang telah ditentukan [15]. Dalam metode ini, terdapat 4 tahap pengembangan, yaitu:

1) *Inception*: Pada tahap ini dilakukan proses pengidentifikasian sistem dengan menganalisis kebutuhan akan sistem yang akan dibuat, melakukan kajian terhadap penelitian yang terkait dengan penyimpanan *private cloud* [16-19] menggunakan *Samba* dan *OpenVPN* yang berbasis pada perangkat *Raspberry Pi*.

2) *Elaboration*: Pada tahap ini dilakukan proses analisis masalah dengan menggambarkan rancangan alur sistem yang akan dibuat menggunakan blok diagram sistem seperti pada Gambar 1.

Pada Gambar 1 menjelaskan tentang blok diagram perancangan alur sistem *private cloud server* yang akan dibangun. Perangkat *Raspberry Pi* berperan sebagai komputer *server* yang akan memberikan layanan penyimpanan data kepada pengguna. Didalam perangkat *Raspberry Pi* terpasang aplikasi *Samba* yang berperan sebagai layanan *file sharing* dan aplikasi *OpenVPN* diterapkan sebagai protokol keamanan data pada *server*. Untuk bisa saling terhubung antara *server* dan pengguna, dibutuhkannya koneksi internet yang disediakan oleh ISP dan disalurkan melalui perangkat *access point*.

3) *Construction*: Pada tahap ini, peneliti membangun semua perancangan yang sudah direncanakan, mulai dari melakukan perakitan alat *Raspberry Pi 4 Model B* dengan menambahkan komponen *harddisk external*. Kemudian, melakukan konfigurasi dan pengkodean (*coding*) pada perangkat *Raspberry Pi* untuk membuat sistem *private cloud* menggunakan program *Samba* dan dikombinasikan dengan *OpenVPN* sebagai protokol keamanan pada *server*.

4) *Transition*: Pada tahap ini difokuskan untuk melakukan proses *testing*, analisis kinerja, dan pengujian terhadap sistem yang telah dibuat serta memperbaiki masalah-masalah yang muncul pada saat pembuatan dan setelah dilakukannya pengujian sistem.

III. HASIL DAN PEMBAHASAN

A. Implementasi Perangkat Keras

Perangkat *Raspberry Pi* berfungsi sebagai komputer *server* yang di dalamnya terpasang program *Samba* dan *OpenVPN*. Program tersebut dirangkai sehingga menjadi sebuah sistem *private cloud server* yang bisa menjembatani semua aktifitas pengguna dalam menyimpan atau mengakses file dan data pribadi mereka. Perangkat *harddisk* digunakan sebagai tempat penyimpanan data yang dihubungkan langsung pada perangkat *Raspberry Pi 4 Model B*. Perangkat ini membutuhkan akses jaringan internet dari sebuah penyedia layanan internet yang disalurkan melalui *access point* agar penggunaanya bisa mengakses *server* dari jarak jauh. Untuk hasil dari implementasi perangkat keras dapat di tampilkan pada Gambar 2.

B. Implementasi Perangkat Lunak

1) *Instalasi dan Konfigurasi Aplikasi Samba*: Langkah pertama yang dilakukan adalah *install Samba* pada perangkat *Raspberry Pi*. *Samba* dalam penelitian ini merupakan program yang digunakan untuk menjembatani hubungan antara *client* dan *server*. Dengan menggunakan *samba*, *client* dapat mengakses penyimpanan dari tempat lain menggunakan jaringan internet. *Install Samba* dapat dilakukan melalui terminal dengan cara mengetikkan perintah sebagai berikut:

```
# sudo apt install samba samba-common-bin
```

Setelah *Samba* selesai terpasang, langkah selanjutnya melakukan konfigurasi dengan mengetikkan perintah sebagai berikut:

```
# sudo nano /etc/samba/smb.conf
```

Tahap selanjutnya menambahkan bagian yang baru untuk *file* konfigurasi *Samba* sesuai dengan kebutuhan penelitian. Dalam penelitian ini, peneliti akan membuat 4 folder *sharing* yang memiliki fungsinya masing-masing. Caranya yaitu *scroll bar* ke bagian paling bawah dari *file* konfigurasi dan mengetikkan perintah sebagai berikut:

```
[File_Skripsi1]
path = /home/pi/BayuHDD/skripsi1
writeable = yes
browseable = yes
create mask = 0777
directory mask = 0777
valid users = skripsi1
public = yes
[File_Skripsi2]
path = /home/pi/BayuHDD/
writeable = yes
browseable = yes
create mask = 0777
directory mask = 0777
valid users = skripsi2
public = yes
[File_Skripsi3]
path = /home/pi/BayuHDD/
writeable = yes
browseable = yes
create mask = 0777
directory mask = 0777
guest ok = yes
public = yes
[File_Skripsi4]
path = /home/pi/BayuHDD/skripsi4
writeable = yes
browseable = yes
create mask = 0777
directory mask = 0777
valid users = skripsi4
public = yes
```

Sebagai keterangan “File Skripsi” adalah nama *file sharing Samba*, “*path*” adalah tempat direktori penyimpanan file yang akan di *share*, “*writeable = yes*” berarti *file sharing* dapat di kelola (menambah, mengedit, menghapus) oleh pengguna, “*browseable = yes*” berarti *file sharing* dapat dilihat oleh pengguna saat mengakses *server*, “*create mask = 0777*” berarti mengatur *permission default* untuk *file* baru, “*directory mask = 0777*” berarti mengatur *permission default* untuk *file* direktori baru. Angka *0777* berarti semua pengguna memiliki izin penuh untuk membaca, menulis, dan mengeksekusi *file*. “*Valid users*” berarti hanya pengguna yang disebutkan yang diizinkan untuk memiliki akses, “*guest = ok*” berarti pengguna tanpa akun (*guest*) dapat mengakses *file sharing* tanpa perlu memasukkan *username* dan *password*, “*public = yes*” berarti *file sharing* dapat diakses oleh pengguna yang valid yang akan diizinkan untuk mengakses sesuai dengan *valid users*.

Kemudian, langkah selanjutnya yaitu tekan **Ctrl+X** untuk keluar dan tekan **Y** untuk menyimpan perubahan *file* konfigurasi yang sudah ditambahkan. Lalu, menambahkan pengguna yang dapat mengakses *Samba* dengan membuat *username* dan *password* dengan mengetikkan perintah sebagai berikut:

```
# sudo useradd -M Skripsi1
# sudo smbpasswd -a skripsi1
```

Instalasi dan konfigurasi *Samba* telah berhasil dibangun, selanjutnya lakukan *restart* perangkat Raspberry Pi.

2) *Instalasi dan Konfigurasi OpenVPN*: Hal berikutnya yang dilakukan adalah *install OpenVPN server* pada perangkat Raspberry Pi. *Install OpenVPN server* dapat dilakukan melalui terminal dengan cara mengetikkan perintah sebagai berikut:

```
# curl-L https://install.pivpn.io | bash
```

Konfigurasi *OpenVPN* dilakukan secara bersamaan saat instalasi sedang berjalan. Konfigurasi yang diatur seperti penggunaan *IP address* statis untuk kebutuhan *server*, mengatur *user default*, mengatur protokol *VPN server*, menentukan layanan penyedia *DNS*, dan mengatur beberapa fitur tambahan sesuai kebutuhan *server*. Setelah instalasi *OpenVPN* sudah selesai, tambahkan profil baru untuk pengguna. Profil ini nantinya akan digunakan oleh pengguna untuk terhubung pada *OpenVPN*. Adapun cara menambahkan profil pengguna dapat dilakukan dengan cara mengetikkan perintah sebagai berikut:

```
# pivpn -a
```

Kemudian otomatis akan muncul *form input* untuk menambahkan nama pengguna dan *password* pengguna. Jika sudah selesai, profil *VPN client* tersebut akan otomatis masuk ke dalam *directory* “*ovpns*” pada penyimpanan *home* perangkat Raspberry Pi seperti yang di tampilkan pada Gambar 3.

3) *Instalasi Dan Konfigurasi OpenVPN Connect Untuk Client*: Langkah selanjutnya yang dilakukan adalah *install* aplikasi *OpenVPN Connect* pada perangkat *client*. Aplikasi ini yang akan digunakan oleh *client* atau pengguna untuk bisa terhubung pada *server cloud* menggunakan profil *VPN client* yang sudah dibuat sebelumnya seperti yang ada pada Gambar 3.

Pengguna bisa mengunduh aplikasi *OpenVPN Connect* pada *website* resmi *OpenVPN* dan lakukan instalasi sampai selesai. Setelah itu, jalankan aplikasi tersebut dan coba untuk terkoneksi pada *VPN server* menggunakan profil *VPN client*. Masukkan *password* yang sesuai dengan profil *VPN client* yang telah dibuat. Jika *client* berhasil terhubung pada *VPN server*, maka akan tampil *IP private*, nama *server*, dan *IP public server* seperti pada Gambar 4.

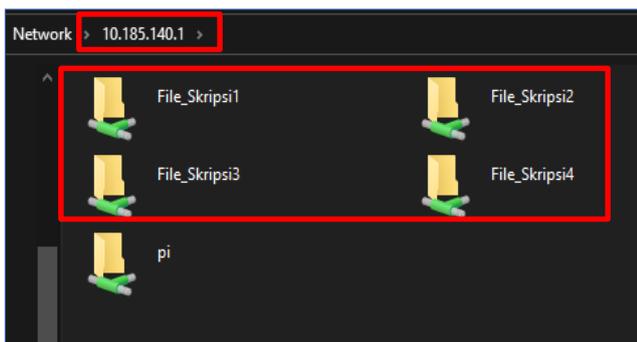


Gambar 4. *IP private server*

Untuk bisa mengakses *file directory Samba*, *client* hanya bisa terhubung menggunakan IP *private* saja yaitu 10.185.140.1. Setelah mendapatkan IP *private server*, langkah selanjutnya yang dilakukan adalah pembatasan *client*. Pembatasan *client* bertujuan agar hanya pengguna tertentu saja yang dapat mengakses *server cloud*. Jadi, hanya pengguna yang memiliki profil VPN *client* saja yang mendapatkan akses untuk terhubung ke VPN *server* dan *Samba*. Adapun cara membatasi *client* yaitu dengan masuk kembali pada bagian *file konfigurasi Samba*, kemudian pada bagian awal konfigurasi ketikkan perintah berikut:

```
[global]
# hosts allow 10.185.140.
```

Lalu tekan Ctrl+X untuk keluar dan tekan Y untuk menyimpan perubahan data konfigurasi yang sudah ditambahkan. Dan langkah terakhir yaitu mencoba untuk masuk pada *file sharing Samba* dengan cara tekan Windows+R, lalu ketik IP *private server* pada jendela *run* yaitu 10.185.140.1 lalu tekan *enter*. Kemudian memasukkan *username* dan *password Samba* yang sudah dibuat sebelumnya. Jika *username* dan *password* yang dimasukkan sudah benar, maka pengguna telah berhasil masuk kedalam *file sharing Samba* seperti yang ditampilkan pada Gambar 5 dan sudah bisa melakukan *sharing* antara *client* dan *server*.



Gambar 5. File sharing samba

C. Pengujian Sistem

Untuk membuktikan kegunaan dari sistem *private cloud server* yang telah dibangun, maka akan dilakukan beberapa tahapan pengujian, antara lain:

1) *Pengujian Kinerja Sistem*: Dilakukan untuk mengetahui apakah sistem yang dibangun sudah berjalan dengan baik sesuai dengan rancangan. Hasil pengujian sistem dapat dilihat pada Tabel I yang menampilkan keterangan sesuai dengan fungsi yang dijalankan apakah berhasil atau tidak.

TABEL I
HASIL PENGUJIAN KINERJA SISTEM

Jenis Pengujian	Berhasil	Gagal
Login menggunakan profil <i>OpenVPN</i>	Ya	-
Terkoneksi dengan <i>OpenVPN server</i>	Ya	-
Login <i>Samba</i>	Ya	-
Upload file	Ya	-
Download file	Ya	-
Sharing file antara <i>client</i>	Ya	-
Membuat file baru	Ya	-
Menghapus file	Ya	-
Mengedit file	Ya	-

Dari hasil yang disajikan pada Tabel I, dapat disimpulkan bahwa sistem *private cloud* yang dibangun sudah sesuai dengan rancangan dan bisa berfungsi sebagai sistem *private cloud server*.

2) *Pengujian Keamanan Terhadap OpenVPN*: Pada pengujian ini akan dilakukan dua tahap pengujian, yaitu pengujian akses layanan *private cloud server* sebelum menggunakan *OpenVPN* dan pengujian akses layanan *private cloud server* setelah menggunakan *OpenVPN*. Pada pengujian pertama, penguji mengakses sistem *private cloud server* dengan IP lokal 192.168.1.6 perangkat Raspberry Pi tanpa menggunakan aplikasi *OpenVPN* dan mencoba melakukan *download file* dengan nama "Dokumen Skripsi Bayu.docx". Kemudian proses *download* tersebut akan direkam dari awal hingga selesai menggunakan aplikasi *Wireshark* untuk menguji keamanannya seperti yang ditampilkan pada Gambar 6.

Dari hasil *capture* menggunakan aplikasi *Wireshark* seperti pada Gambar 6, nama *file* dan isi *file* dapat terlihat, serta IP *client* yang melakukan *download* juga terlihat yaitu 192.168.1.4, lalu IP *server* 192.168.1.6 dan menggunakan protokol *Samba*. *File* tersebut belum terenkripsi karena belum menggunakan VPN.

Selanjutnya pada pengujian kedua, penguji mengakses kembali sistem *private cloud server* Raspberry Pi dengan IP yang bisa diakses dari jaringan publik yaitu 10.185.140.1 dan sudah menggunakan *OpenVPN* sebagai protokol keamanan pada sistem *private cloud server*. Kemudian penguji mencoba melakukan *download file* dengan nama "Dokumen Skripsi Bayu.docx". Kemudian proses *download* tersebut akan direkam dari awal hingga selesai menggunakan aplikasi *Wireshark* untuk menguji keamanannya seperti yang ditampilkan pada Gambar 7.

Dari hasil *capture* menggunakan aplikasi *Wireshark* seperti pada Gambar 7, nama *file*, isi *file*, serta IP *server* tidak terbaca pada aplikasi *Wireshark* dan terbukti *file* tersebut berhasil terenkripsi dengan baik setelah menggunakan *OpenVPN*.

Dari pengujian yang telah dilakukan, dapat disimpulkan bahwa dengan menggunakan aplikasi *OpenVPN* sangat berguna jika diterapkan sebagai protokol keamanan *file* dan data pada sistem *private cloud server* berbasis perangkat Raspberry Pi. Karena *OpenVPN* menggunakan *Secure*

Sockets Layer (SSL) sebagai pengamanan dalam melakukan enkripsi data, inilah sebabnya *OpenVPN* membutuhkan *Certificate Authority* (CA).

3) *Pengujian Transfer Data Dengan Layanan Public Cloud*: Pengujian transfer data dengan layanan *public cloud* dilakukan untuk mengetahui kemampuan sistem *private cloud server* berbasis perangkat *Raspberry Pi* ini dengan layanan *public cloud* yang sudah ada seperti *Google Drive* dan *Dropbox*. Kemudian, yang akan diukur pada pengujian ini adalah kecepatan maksimal transfer data saat *upload* dan *download* file. Selanjutnya hasil dari pengujian tersebut akan disajikan pada Tabel II.

TABEL II
HASIL PENGUJIAN TRANSFER DATA DENGAN LAYANAN *PUBLIC CLOUD*

Jenis Layanan Cloud	Ukuran File	Kecepatan Maksimal Transfer Data dan Estimasi Waktu Transfer Data	
		Upload	Download
<i>Private Cloud</i> Berbasis Perangkat <i>Raspberry Pi</i>	50 MB	2,28 MBps (1 menit 50 detik)	1,93 MBps (1 menit 16 detik)
<i>Google Drive</i>	50 MB	0,96 MBps (1 menit 17 detik)	2,81 MBps (38 detik)
<i>Dropbox</i>	50 MB	0,97 MBps (1 menit 5 detik)	2,71 MBps (30 detik)

Dari hasil yang disajikan pada Tabel II, dapat dilihat hasil pengujian transfer data saat proses *upload* dan *download* data pada layanan *private cloud server* berbasis perangkat *Raspberry Pi*, *Google Drive*, dan *Dropbox*. Pada layanan *Google Drive* maupun *Dropbox* kecepatan maksimal *upload* nya hampir sama yaitu dibawah 1 MBps dan estimasi waktunya juga hampir mirip, kemudian untuk kecepatan maksimal *download* nya juga hampir sama yaitu dibawah 3 MBps dan estimasi waktunya juga hampir mirip. Lalu pada layanan *private cloud server* berbasis perangkat *Raspberry Pi* kecepatan maksimal *upload* nya lebih besar dibanding kedua layanan lainnya yaitu diatas 2 MBps, namun estimasi waktu *upload* nya lebih lama dikarenakan proses transfer data yang kurang stabil. Kemudian kecepatan maksimal *download* nya lebih kecil dibanding kedua layanan lainnya yaitu dibawah 2 MBps, maka otomatis estimasi waktu *download* nya juga nya lebih lama dibanding kedua layanan lainnya.

Sehingga dapat disimpulkan layanan *Google Drive* dan *Dropbox* sedikit lebih unggul dalam hal transfer data dibandingkan dengan layanan *cloud server* *Raspberry Pi*. Tetapi bukan berarti layanan *cloud server* *Raspberry Pi* lebih buruk. Proses *upload* maupun *download* pada setiap layanan *cloud* tersebut tidak terlalu berbeda jauh karena untuk proses *upload* dan *download* juga masih sangat dipengaruhi oleh

bandwidth dari sisi *client* untuk terhubung ke masing-masing layanan *cloud server*.

D. Pemeliharaan dan Pengawasan Sistem

Pemeliharaan dan pengawasan sistem *private cloud server* berbasis perangkat *Raspberry Pi* sangatlah penting untuk dilakukan. Seiring bertambahnya data, penting untuk memantau kapasitas penyimpanan yang tersedia. Pemeliharaan bertujuan untuk memastikan bahwa sistem dapat menambah kapasitas atau mengelola penyimpanan dengan efisien untuk menghindari kehabisan ruang penyimpanan. Layanan *cloud* yang dibangun hanya mengandalkan 1 buah perangkat *harddisk* saja yang sangat berbahaya karena apabila *harddisk* tersebut bermasalah atau rusak, data tidak akan dapat diakses kembali.

Solusi untuk mengatasi masalah tersebut adalah admin hanya perlu melakukan pengecekan dan melakukan *backup* data secara rutin ke perangkat penyimpanan eksternal tambahan untuk memastikan data tetap aman jika terjadi kerusakan pada *harddisk* utama. Kemudian, sebagai alternatif lain untuk menghindari masalah tersebut, admin juga perlu menambahkan perangkat *harddisk* baru dan menambahkan konfigurasi RAID (*Redundant Array of Independent Disks*) dengan tujuan untuk meningkatkan keandalan penyimpanan data. RAID memungkinkan data dapat disimpan dan dibagi pada beberapa *harddisk*, sehingga jika terjadi kegagalan pada salah satu *harddisk* maka data masih dapat diakses dari *harddisk* lainnya.

E. Perbandingan Total Cost of Ownership (TCO)

Perbandingan *Total Cost of Ownership* (TCO) antara mengimplementasikan sistem *private cloud server* berbasis perangkat *Raspberry Pi* dan berlangganan layanan *public cloud* melibatkan beberapa komponen biaya.

1) *Sistem Private Cloud Server Berbasis Perangkat Raspberry Pi*: Untuk membangun sebuah sistem *private cloud server* berbasis perangkat *Raspberry Pi* diperlukan beberapa perangkat utama dan pendukung yang dapat dilihat pada Tabel III.

TABEL III
PERANGKAT UNTUK MEMBANGUN SISTEM *PRIVATE CLOUD SERVER* BERBASIS PERANGKAT *RASPBERRY PI*

No.	Nama Perangkat	Harga (Rp)
1	<i>Raspberry Pi</i> 4 Model B	900.000
2	<i>Harddisk</i> <i>Seagate</i> 1 TB	900.000
3	USB Hub	60.000
4	Power Adaptor <i>Raspberry Pi</i>	35.000
5	MicroSD 32 GB	75.000
6	<i>Raspberry Pi</i> Case	30.000
Total (Rp)		2.000.000

Selain beberapa perangkat yang sudah disebutkan pada Tabel III, perlu di perhatikan juga terkait biaya konsumsi

listrik dan biaya internet. Oleh karena itu dapat dijabarkan sebagai berikut:

- Perangkat Raspberry Pi 4 Model B mengkonsumsi sekitar 3W daya.
- Maka dapat dihitung biaya listrik per bulan $3W \times 24 \text{ jam} \times 30 \text{ hari} = 2.16 \text{ kWh}$ per bulan.
- Harga tarif listrik Rp 1.500 per kWh, maka $2.16 \text{ kWh} \times \text{Rp } 1.500 = \text{Rp } 3.240$ per bulan.
- Biaya konsumsi listrik per bulan $\text{Rp } 3.240 \times 12 \text{ bulan} = \text{Rp } 38.880$ per tahun.
- Biaya internet per bulan yaitu $\text{Rp } 300.000 \times 12 \text{ bulan} = \text{Rp } 3.600.000$ per tahun.

Sehingga, total biaya yang perlu dikeluarkan untuk membangun sebuah sistem *private cloud server* berbasis perangkat Raspberry Pi dengan kapasitas penyimpanan 1 TB yaitu $\text{Rp } 2.000.000 + \text{Rp } 38.880 + \text{Rp } 3.600.000 = \text{Rp } 5.638.880$.

2) *Berlangganan Layanan Public Cloud Google Drive Dan Dropbox*: Untuk layanan Google Drive dengan kapasitas penyimpanan 2 TB mematok harga sebesar Rp 1.950.00 per tahun atau Rp 975.000 per tahun untuk kapasitas penyimpanan 1 TB. Kemudian untuk layanan Dropbox dengan kapasitas penyimpanan 2 TB mematok harga sebesar 2.300.000 per tahun atau Rp 1.150.000 per tahun untuk kapasitas penyimpanan 1 TB. Saat menggunakan layanan Google Drive dan Dropbox, biaya konsumsi listrik dan internet tetap perlu diperhitungkan dikarenakan perangkat komputer harus mengakses layanan tersebut untuk menyimpan file dan data. Sehingga dapat dijabarkan sebagai berikut:

- Perangkat komputer rata-rata mengkonsumsi sekitar 150W daya.
- Misalkan waktu penggunaan layanan *cloud* adalah 8 jam per hari, maka konsumsi energi per hari = $150W \times 8 \text{ jam} \times 30 \text{ hari} = 36 \text{ kWh}$ per bulan.
- Harga tarif listrik Rp 1.500 per kWh, maka $36 \text{ kWh} \times \text{Rp } 1.500 = \text{Rp } 54.000$ per bulan.
- Biaya konsumsi listrik per bulan $\text{Rp } 54.000 \times 12 \text{ bulan} = \text{Rp } 648.000$ per tahun.
- Biaya internet per bulan yaitu $\text{Rp } 300.000 \times 12 \text{ bulan} = \text{Rp } 3.600.000$ per tahun.

Sehingga, total biaya keseluruhan yang perlu dikeluarkan untuk berlangganan layanan Google Drive dengan kapasitas penyimpanan 1 TB yaitu $\text{Rp } 975.000 + \text{Rp } 648.000 + \text{Rp } 3.600.000 = \text{Rp } 5.223.000$ per tahun. Sedangkan, untuk berlangganan layanan Dropbox dengan kapasitas penyimpanan 1 TB yaitu $\text{Rp } 1.150.000 + \text{Rp } 648.000 + \text{Rp } 3.600.000 = \text{Rp } 5.398.000$ per tahun.

3) *Total Cost of Ownership (TCO) Per Tahun*: Berdasarkan perhitungan biaya yang telah di jabarkan pada poin A dan poin B, maka hasil perbandingan biaya dapat dilihat pada Tabel IV berikut.

TABEL IV
HASIL PERBANDINGAN *TOTAL COST OF OWNERSHIP* PER TAHUN

Jenis Layanan	Kapasitas Penyimpanan	Total Cost of Ownership (Per Tahun)
Private Cloud Berbasis Perangkat Raspberry Pi	1 TB	Rp 5.638.880
Google Drive	1 TB	Rp. 5.223.000
Dropbox	1 TB	Rp 5.398.000

Dari hasil yang sajikan pada Tabel IV, dapat disimpulkan bahwa sistem *private cloud server* berbasis perangkat Raspberry Pi membutuhkan *Total Cost of Ownership (TCO)* yang lebih mahal per tahunnya di bandingkan dengan layanan Google Drive dan Dropbox. Namun, perlu di ingat bahwa untuk layanan Google Drive dan Dropbox perlu mengeluarkan biaya tersebut setiap tahunnya (berlangganan) jika ingin menggunakan layanannya. Sedangkan untuk sistem *private cloud server* berbasis perangkat Raspberry Pi hanya perlu mengeluarkan biaya di awal pembuatan sistemnya saja tanpa perlu berlangganan setiap bulan ataupun setiap tahun.

Adapun biaya tambahan yang akan di keluarkan jika diperlukan yaitu biaya pemeliharaan pada perangkat keras dan perangkat lunak, seperti penggantian komponen perangkat, pembaruan sistem operasi, pembaruan konfigurasi VPN, penambahan koneksi internet, menerapkan alat monitoring tambahan, serta biaya tambahan jika terjadinya kegagalan sistem (*downtime*).

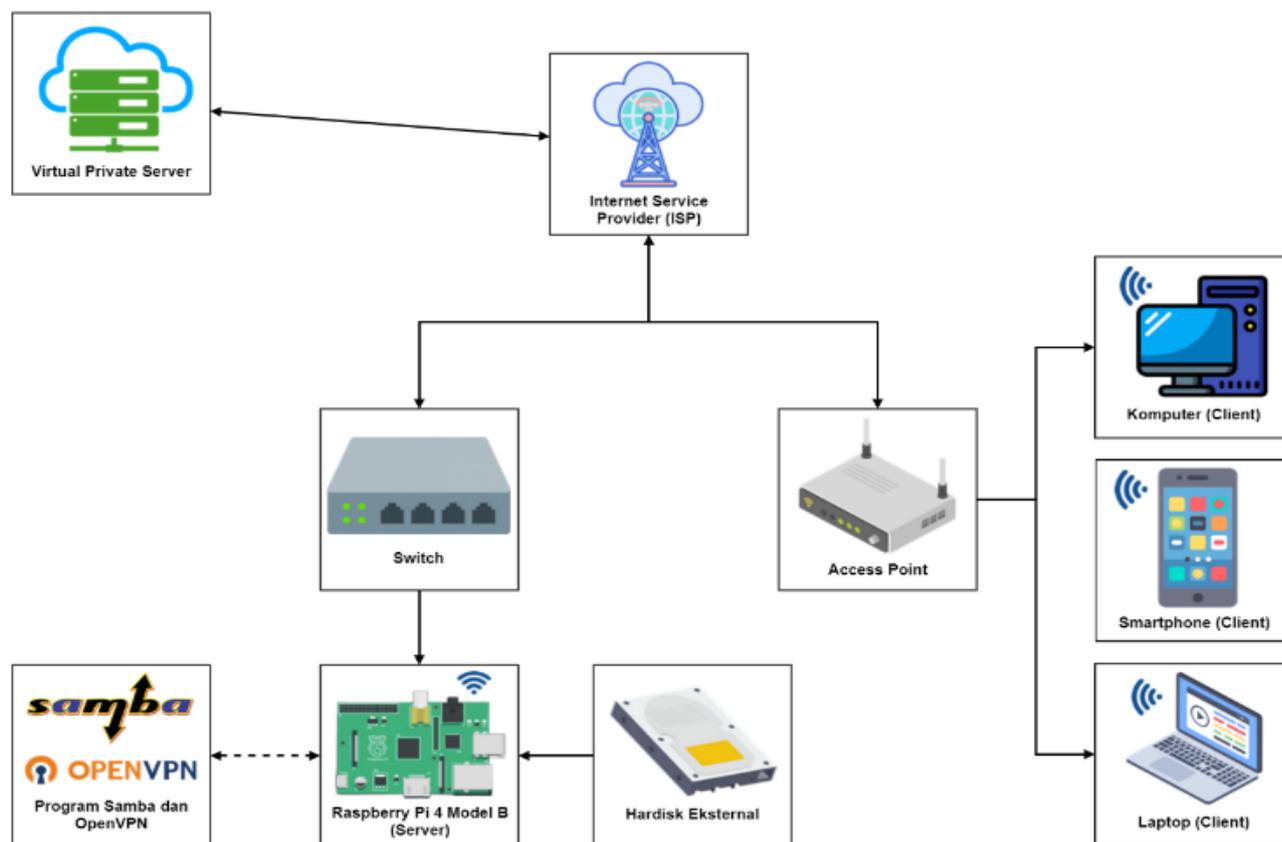
IV. KESIMPULAN

Kesimpulan yang didapatkan pada penelitian ini adalah pengimplementasian sistem *private cloud server* berbasis perangkat Raspberry Pi 4 Model B berhasil dibangun dengan memanfaatkan aplikasi *Samba* dan aplikasi *OpenVPN*. Berdasarkan pengujian kinerja sistem, didapatkan hasil bahwa sistem yang dibangun telah berjalan sesuai dengan rancangan awal dan bisa digunakan sebagai sistem *private cloud*. Dengan penggunaan aplikasi *OpenVPN* membuat sistem *private cloud* menjadi lebih aman dikarenakan pertukaran data yang masuk dan keluar antara *client* dan sistem *cloud* akan terenkripsi dengan baik.

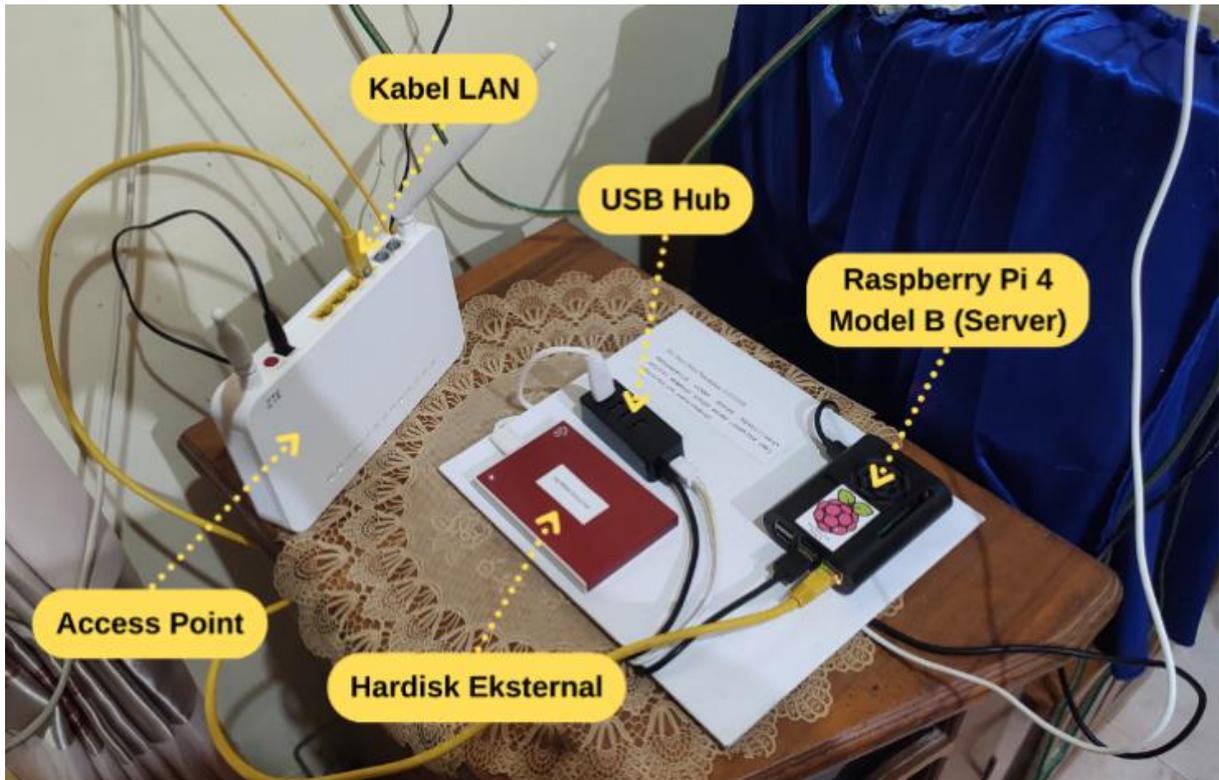
DAFTAR PUSTAKA

- [1] K. Suharta. (2018). Rancang Bangun Personal Cloud Storage Berbasis Raspberry Pi. [Online]. Available: <http://eprints.uty.ac.id/id/eprint/2387>.
- [2] L. F. Aksara, S. N. Rifai, and Sutardi, "Perancangan dan implementasi private cloud storage studi kasus: jurusan teknik informatika universitas halu oleo," *semanTIK*, vol. 4, no. 1, pp. 143–150, 2018.
- [3] M. A. S. Arifin, "Privat cloud storage server rendah energi menggunakan raspberry pi sebagai media penyimpanan online pribadi," vol. 3, no. 2, 2017.
- [4] S. Prasath Kumar, P. Rayavel, N. Anbarasi, B. Renukadevi, and D. Maalini, "Raspberry pi based secured cloud data," *J. Phys. Conf. Ser.*, vol. 1964, no. 4, 2021.
- [5] S. Aisa, "Implementation raspberry pi using private cloud for accessing personal data," *J. Penelit. Pos dan Inform.*, vol. 6, no. 2, p. 137, 2016.

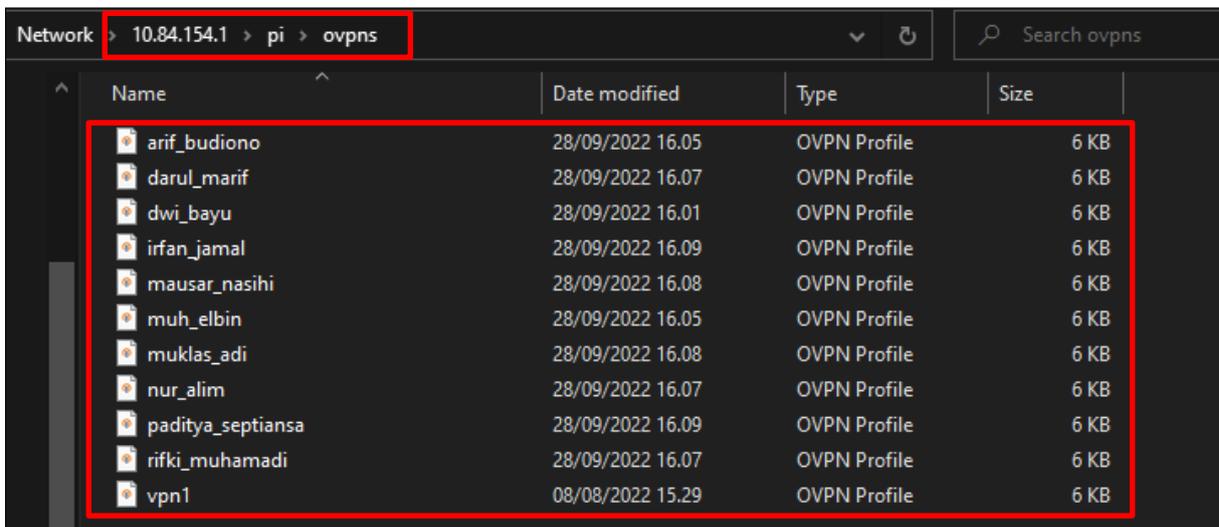
- [6] W. Setiawan, N. Fajriyah, and T. Duha, "Analisa layanan cloud computing di era digital," *J. Inform.*, vol. 1, no. 1, pp. 32–39, 2022.
- [7] F. Roma Doni, "Jaringan komputer dengan router mikrotik," *Tek. Komput. AMIK BSI*, vol. 1, p. 88, 2016.
- [8] M. Redi and R. Mohamad, "Aplikasi penggajian karyawan berbasis client-server pada pt. radio nasional buana suara," *Ekp*, vol. 13, no. 3, pp. 1576–1580, 2017.
- [9] I. Santiko and R. Rosidi, "Pemanfaatan private cloud storage sebagai media penyimpanan data e-learning pada lembaga pendidikan," *J. Tek. Inform.*, vol. 10, no. 2, pp. 137–146, 2018.
- [10] R. Astuti, I. Ruslianto, and Suhardi, "Rancang bangun network attached storage pada raspberry pi 3 model b berbasis website," *Coding J. Komput. dan Apl.*, vol. 08, no. 01, pp. 185–196, 2020.
- [11] H. W. Sulistyono and H. Oktavianto, "Perancangan dan implementasi file sharing," *J. Apl. Sist. Inf. Dan Elektron.*, vol. 2, no. 1, pp. 24–30, 2020.
- [12] M. Badrul, "Open vpn-access server dengan enkripsi ssl/ti open ssl," *INFORMATICS Educ. Prof. J. Informatics*, vol. 1, no. 1, pp. 1–12, 2016.
- [13] A. Skendzic and B. Kovacic, "Open source system openvpn in a function of virtual private network," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 200, no. 1, 2017.
- [14] C. Wai Zhao, J. Jegatheesan, and S. Chee Loon, "Exploring iot application using raspberry pi," *Int. J. Comput. Networks Appl.*, vol. 2, no. 1, pp. 27–34, 2015.
- [15] F. Mubarak, H. Harliana, and I. Hadijah, "Perbandingan antara metode rup dan prototype dalam aplikasi penerimaan siswa baru berbasis web," *Creat. Inf. Technol. J.*, vol. 2, no. 2, p. 114, 2015.
- [16] D. E. Kurniawan, H. Arif, N. Nelmiawati, A. H. Tohari, and M. Fani, "Implementation and analysis ipsec-vpn on cisco asa firewall using gns3 network simulator," in *Journal of Physics: Conference Series*, IOP Publishing, 2019, p. 012031.
- [17] D. E. Kurniawan, I. Ahmad, M. R. Ridho, F. Hidayat, and A. A. Js, "Analysis of performance comparison between Software-Based iSCSI SAN and Hardware-Based iSCSI SAN," *J. Phys.: Conf. Ser.*, vol. 1351, no. 1, p. 012009, Nov. 2019, doi: 10.1088/1742-6596/1351/1/012009.
- [18] D. E. Kurniawan, M. Iqbal, J. Friadi, R. I. Borman, and R. Rinaldi, "Smart monitoring temperature and humidity of the room server using raspberry pi and whatsapp notifications," in *Journal of Physics: Conference Series*, IOP Publishing, 2019, p. 012006.
- [19] N. Nelmiawati, N. C. Kushardianto, A. H. Tohari, Y. P. Hasibuan, and D. E. Kurniawan, "Rancang Bangun Lab Komputer Virtual Berbasis Cloud Computing Menggunakan Openstack Pada Jaringan Terpusat," *Journal of Applied Informatics and Computing*, vol. 2, no. 1, Art. no. 1, Jul. 2018, doi: 10.30871/jaic.v2i1.821..



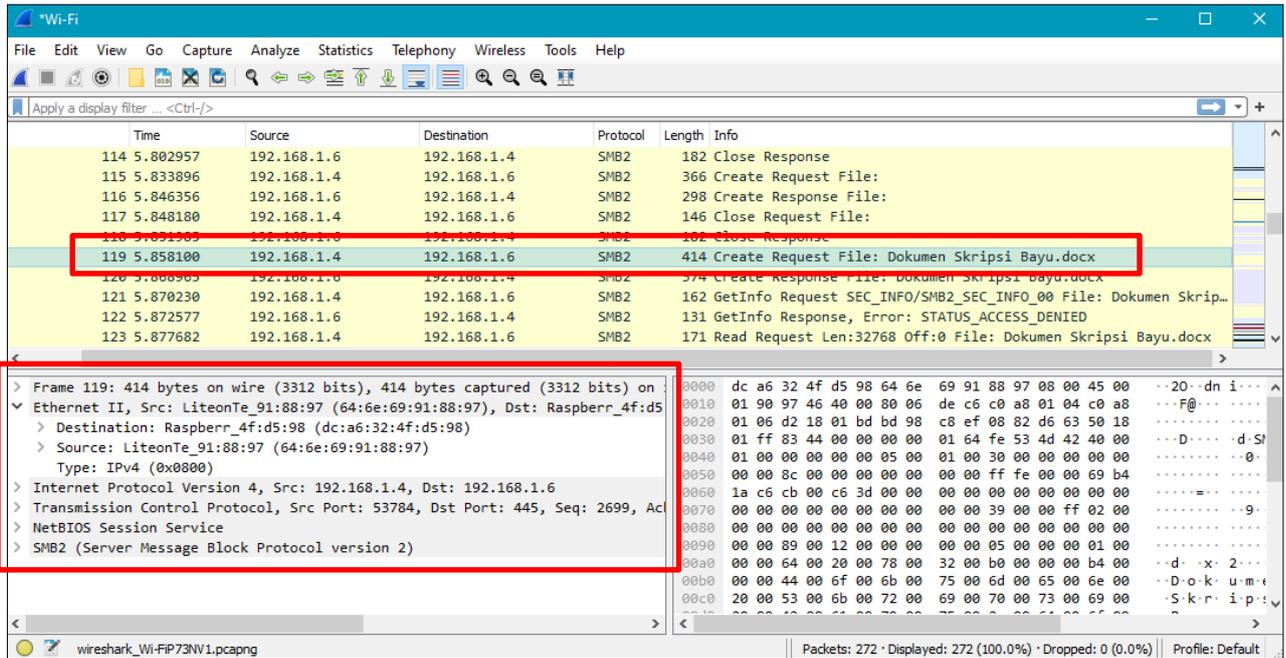
Gambar 1. Blok diagram sistem



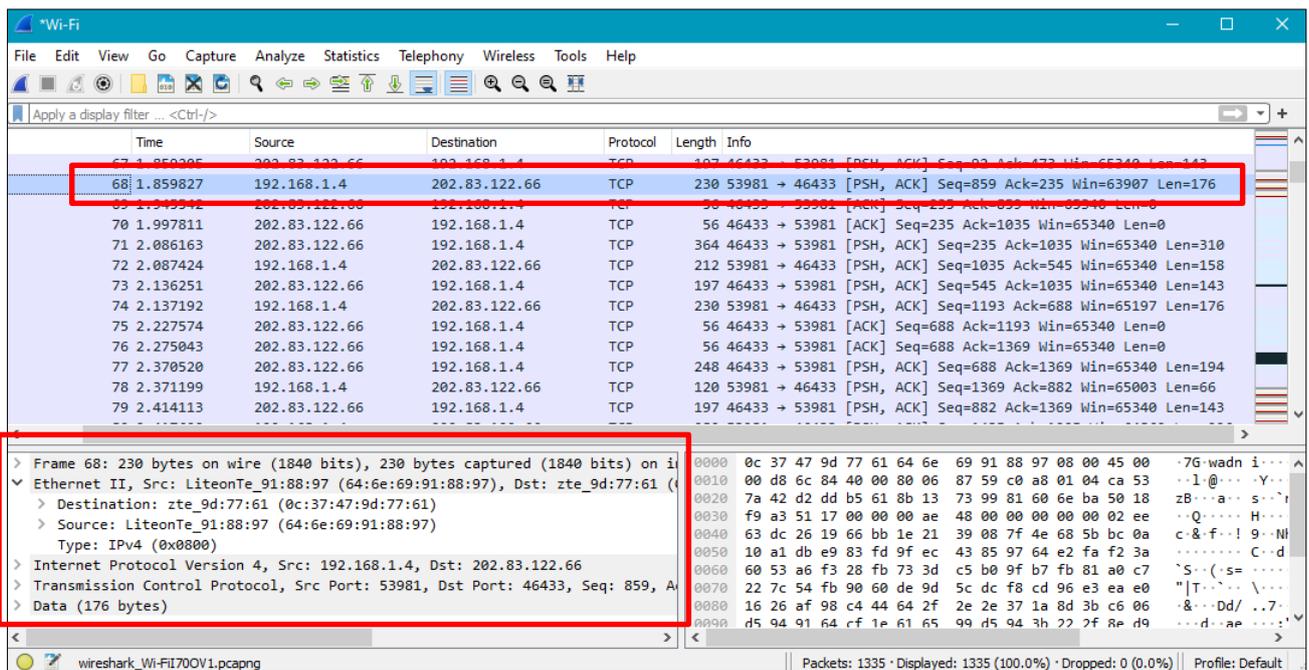
Gambar 2. Tampilan implementasi perangkat keras



Gambar 3. Profil VPN client



Gambar 6. Hasil capture wireshark sebelum menggunakan OpenVPN



Gambar 7. Hasil capture wireshark sesudah menggunakan OpenVPN