

Virtual Private Network (VPN) Secure Socket Tunneling Protocol (SSTP) Menggunakan Raspberry Pi

Sugiyatno ^{1,*}, Prima Dina Atika ¹

¹ Teknik Informatika; Universitas Bhayangkara Jakarta Raya; Jl. Raya Perjuangan, Marga Mulya, Bekasi Utara, Marga Mulya, Bekasi Utara, Kota Bks, Jawa Barat 17121 (021) 88955882; e-mail: sugiyatno@dsn.ubharajaya.ac.id, primadina@dsn.ubharajaya.ac.id

* Korespondensi: e-mail: sugiyatno@dsn.ubharajaya.ac.id

Diterima: 16 April 2018 ; Review: 25 April 2018 ; Disetujui: 4 Mei 2018

Cara sitasi: Sugiyatno, Atika PD. 2018. *Virtual Private Network (VPN) Secure Socket Tunneling Protocol (SSTP) Menggunakan Raspberry Pi*. Information System For Educators and Professionals. 2 (2): 155-166.

Abstrak: Dalam implementasinya jaringan VPN diperlukan sebuah server sebagai penghubung dan manajemen user. Perangkat dengan spesifikasi mumpuni biasa digunakan untuk merancang VPN server pada jaringan *enterprise* yang berskala besar. Namun, penggunaan perangkat tersebut untuk jaringan skala kecil kurang efisien. Oleh karena itu, digunakan perangkat Raspberry Pi sebagai alternatif dalam membangun VPN server, dimana protokol yang digunakan adalah VPN SSTP. Pada penelitian ini, dirancang suatu koneksi VPN SSTP dengan server Raspberry Pi dan client PC dengan sistem operasi Ubuntu. Dan juga, dibuat suatu rancangan VPN PPTP sebagai perbandingan terhadap VPN SSTP. Kemudian dilakukan pengujian terhadap performa dan keamanannya. Pada pengujian performa, parameter yang diuji adalah *packet loss*, *round trip time* dan *SFTP file transfer*. Dan pada pengujian keamanan, parameter yang diuji adalah *sniffing*. Hasil dari pengujian performa, menunjukkan bahwa VPN SSTP sedikit lebih baik daripada VPN PPTP, terutama pada pengujian *packet loss* dan *round trip time*. Tetapi pada pengujian *SFTP file transfer*, VPN PPTP lebih baik daripada VPN SSTP. Sedangkan hasil dari pengujian keamanan, menunjukkan bahwa VPN SSTP aman terhadap serangan *sniffing*, hal itu ditunjukkan pada hasil yang didapat dimana *username* dan *password* yang digunakan untuk *login* tidak dapat diketahui oleh *attacker*, sedangkan pada VPN PPTP *username*-nya dapat diketahui, tetapi *password* tidak dapat terbaca karena terenkripsi MS-CHAPv2.

Kata kunci: Raspberry Pi, SSTP, Ubuntu, VPN

Abstract: *A In the implementation of VPN network required a server as a liaison and user management. A device with a highly qualified specification is used to design VPN servers on large enterprise networks. However, the use of such devices for small-scale networks is less efficient. Therefore, Raspberry Pi device used as an alternative in building a VPN server, where the protocol used is the VPN SSTP. In this study, designed an SSTP VPN connection with Raspberry Pi server and client PC with Ubuntu operating system. Also, a VPTP VPTP design is made in comparison to the SSTP VPN. Then tested the performance and safety. In performance testing, the parameters tested were packet loss, round trip time and SFTP file transfer. And on security testing, the parameter tested is sniffing. The result of performance testing shows that the SSTP VPN is slightly better than PPTP VPN, especially in packet loss and round trip time testing. But on SFTP file transfer testing, PPTP VPN is better than SSTP VPN. While the result of security testing shows that the SSTP VPN is safe against sniffing attacks, it is shown in the results obtained where the username and password used to login can not be known by the attacker, while the VPN PPTP username can be known, but the password can not be readable because of MS-CHAPv2 encrypted.*

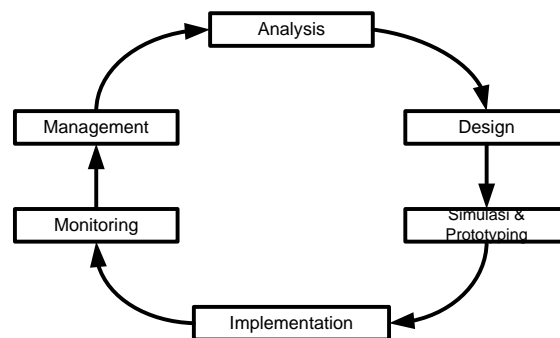
Keywords: Raspberry Pi, SSTP, Ubuntu, VPN

1. Pendahuluan

Pemanfaatan internet untuk transmisi data harus diperhatikan tingkat keamanannya, salah satu upaya dilakukan dengan *Virtual Private Network (VPN)* [Badrul, 2016]. *VPN* merupakan teknologi jaringan private yang berada diatas jaringan public (internet). Perusahaan yang telah mengimplementasi teknologi *VPN*, dapat memberi akses kepada setiap orang yang berada di semua cabang seperti halnya ketika menggunakan jaringan lokal. Hal ini dilakukan agar hanya orang yang mempunyai akses lokal saja yang berhak mengakses system *VPN*, sehingga sistem keamanan perusahaan dapat terjaga [Yang, 2011]. Untuk proses enkapsulasi trafik pada protocol *HTTPS* difasilitasi oleh *SSTP*. Penggunaan *PPP* memungkinkan dukungan untuk metode autentikasi yang kuat dan handal seperti *EAP-TLS*. Penggunaan *HTTPS* berarti trafik akan mengalir melalui *TCP* port 443, *port* yang umum digunakan untuk akses web [Sirisukha, 2003]. Untuk membangun sebuah jaringan *VPN* diperlukan sebuah *server* sebagai penghubung dan manajemen *user*. Oleh karena itu, digunakan perangkat *Raspberry Pi* sebagai alternatif dalam membangun *VPN server* [Dinata, 2017]. Pada penelitian ini, penulis akan merancang suatu koneksi *VPN SSTP* dengan *Raspberry Pi* sebagai *VPN server*, dan *client* menggunakan *PC* atau *laptop* dengan sistem operasi *Ubuntu Desktop*. Dan juga akan dilakukan perbandingan dengan protokol *VPN* yang lain, dalam hal ini adalah protokol *PPTP* [Oktivasari and Utomo, 2016].

2. Metode Penelitian

Penelitian tentang perancangan *VPN SSTP* ini menggunakan metode penelitian *Network Development Life Cycle (NDLC)*. Metodologi analisis dan desain jaringan adalah pendekatan praktis, langkah demi langkah untuk analisis dan desain jaringan [ORACLE et al., 2011].



Sumber: Oppenheimer (2011)

Gambar 1. Network Development Life Cycle

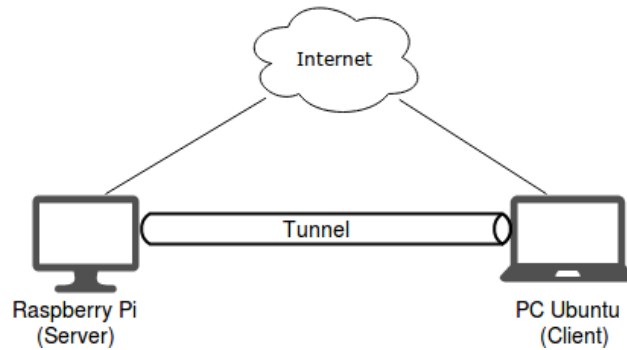
Sesuai dengan Gambar 1, berikut penjelasan dari *NDLC*: **1) Analysis**, mengidentifikasi yang dibutuhkan pengguna jaringan seperti lingkup bisnis, aplikasi, dan data sebelum jaringan kegiatan desain dalam hal ini menganalisis kebutuhan peralatan untuk implementasi *VPN*, seperti komputer dan server (*Raspberry Pi*). **2) Design**, Suatu jaringan harus dirancang untuk memberikan solusi dan kinerja menangani yang di perlukan pengguna dengan software simulasi dan penentuan topologi fisik dan topologi logic. **3) Simulasi & Prototyping**, membuat simulasi dari hasil analisis dan desain. **4) Implementation**, sebelum di sebarluaskan diperlukan tahapan menguji coba hasil simulasi untuk memantau kinerja, memperbaiki masalah, dan mendapatkan pengalaman dengan melakukan tahapan seperti : *VPN SSTP* antara *server Raspberry Pi* dengan *client* sistem operasi *Ubuntu* berhasil dibuat, (b) Hasil pengujian performa pada *packet loss*, *round trip time* dan *SFTP file transfer* pada *VPN SSTP* menunjukkan hasil yang kurang baik, karena menggunakan enkripsi yang lebih dibandingkan *VPN PPTP*, (c) Hasil pengujian keamanan pada *sniffing* menunjukkan hasil bahwa *VPN SSTP* lebih aman daripada *VPN PPTP*, karena *VPN SSTP* menggunakan *certificate SSL* sebagai autentikasi tambahan. **5) Monitoring**, memantau hasil implementasi dengan memantau kinerja dan masalah yang muncul dengan melakukan (a) Variabel terikat (*dependent variable*), variabel yang diukur sebagai akibat adanya manipulasi pada variabel bebas. Variabel terikat penelitian ini adalah performa dan keamanan pada *VPN SSTP* dan *PPTP*. (b) Variabel bebas (*independent variable*), Variabel yang dimanipulasi secara sistematis. Variabel bebas penelitian ini adalah parameter

performa dan keamanan yang menguji pada VPN yang dirancang. **6) Management**, mengatur agar sistem yang sudah berjalan dapat bekerja lebih baik.

Desain penelitian yang digunakan dalam penelitian ini adalah *static group comparison design* yang merupakan bentuk desain penelitian eksperimen semu. Pada desain ini menggunakan dua kelompok, yaitu kelompok VPN SSTP dan kelompok VPN PPTP. Kedua kelompok tersebut akan diberikan *posttest* berupa parameter performa dan keamanan untuk menguji pada VPN yang dirancang, dan kemudian dibandingkan untuk mengetahui VPN mana yang lebih unggul dalam performa dan keamanannya [Brenton et al , 2016].

2.1. Perancangan Topologi Jaringan

Topologi jaringan yang akan dibangun adalah sebagai berikut:

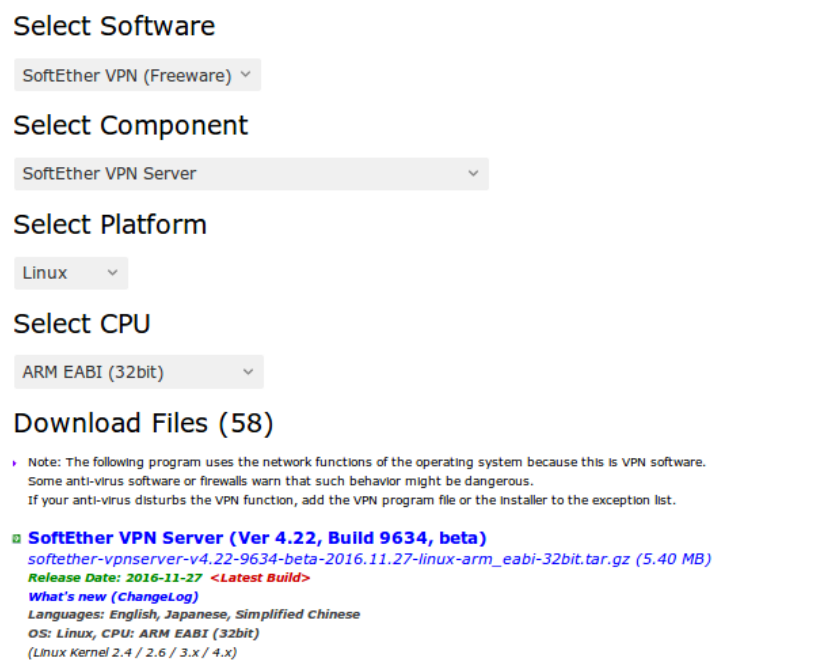


Sumber: Sirisukha (2003)

Gambar 2. Topologi Jaringan

2.2. Perancangan VPN SSTP Server

Perancangan konfigurasi pada server adalah dengan menggunakan *software Softether VPN Server* [Ocean, 2012].



Sumber: Ocean (2012)

Gambar 3. Softether VPN Server Download

Setelah didownload, masuk ke *directory Downloads* dan lakukan *extract file* dengan *command tar*.

```
root@raspin:~# cd /home/pi/Downloads/
root@raspin:/home/pi/Downloads# tar zxvf softether-vpnserver-v4.22-9634-beta-2016.11.27-linux-arm_eabi 32bit.tar.gz
```

```
root@raspin:/home/pi/Downloads# cd vpnserver/
root@raspin:/home/pi/Downloads/vpnserver# make
```

Setelah melakukan perintah *make*, ketik 1 sebanyak tiga kali untuk *read and accept License Agreement*. Lalu, pindahkan **vpnserver** ke *directory /usr/local/* dan ubah *permission* dari semua file dengan *permission 600 (read, write)*, sedangkan untuk file *vpnserver* dan *vpncmd* dengan *permission 700 (read, write, execute)*.

```
root@raspin:/home/pi/Downloads# cd /usr/local/vpnserver/
root@raspin:/usr/local/vpnserver# chmod 600 *
root@raspin:/usr/local/vpnserver# chmod 700 vpnserver
root@raspin:/usr/local/vpnserver# chmod 700 vpncmd
```

Lalu, *start service VPN client* dengan perintah **./vpnserver start** dan untuk melakukan *configure VPN server* menggunakan **./vpncmd** kemudian pilih **1** untuk *management VPN server*. Tekan Enter untuk masuk ke *command line VPN server*.

```
root@raspin:/usr/local/vpnserver# ./vpnserv start
The SoftEther VPN Server service has been started.
root@raspin:/usr/local/vpnserver# ./vpncmd
1. Management of VPN Server or VPN Bridge
2. Management of VPN Client
3. Use of VPN Tools (certificate creation and Network Traffic Speed Test Tool)
Select 1, 2 or 3: 1
VPN Server>
```

Setelah itu tambahkan *password* untuk *server* dengan perintah **ServerPasswordSet**.

```
VPN Server>ServerPasswordSet
ServerPasswordSet command - Set VPN Server Administrator Password
Please enter the password. To cancel press the Ctrl+D key.
Password: *****
Confirm input: *****
The command completed successfully.
```

Softether menggunakan *hub* untuk membedakan file konfigurasi. Untuk membuat *hub* adalah menggunakan perintah **HubCreate** dan juga masukkan *password* untuk *hub*. *Hub* yang dibuat adalah dengan nama **myHub**. Setelah dibuat, kemudian pilih *hub* tersebut untuk melakukan konfigurasi.

```
VPN Server>HubCreate myHub
HubCreate command - Create New Virtual Hub
Please enter the password. To cancel press the Ctrl+D key.
Password: ****
Confirm input: ****
The command completed successfully.
```

```
VPN Server>Hub myHub
Hub command - Select Virtual Hub to Manage
The Virtual Hub "myHub" has been selected.
```

```
The command completed successfully.
VPN Server/myHub>
```

Setelah *hub* dipilih, selanjutnya aktifkan *SecureNAT*. *SecureNAT* memudahkan akses dari *hub* ke jaringan lokal. Jika *SecureNAT* tidak diaktifkan, maka memerlukan *DHCP server* terpisah dengan konfigurasi untuk mengakses jaringan lokal. Untuk mengaktifkan *SecureNAT*, jalankan perintah **SecureNatEnable**.

```
VPN Server/myHub>SecureNatEnable
SecureNatEnable command - Enable the Virtual NAT and DHCP Server
Function (SecureNat Function)
The command completed successfully.
```

Selanjutnya adalah menambahkan *user* dengan perintah **UserCreate** dan tambahkan *password* pada *user* yang telah dibuat dengan perintah **UserPasswordCreate**.

```
VPN Server/myHub>UserCreate ipin
UserCreate command - Create User
Assigned Group Name:
User Full Name: sugiyatno
User Description: Test User
The command completed successfully.
```

```
VPN Server/myHub>UserPasswordSet ipin
UserPasswordSet command - Set Password Authentication for User Auth Type and Set
Password
Please enter the password. To cancel press the Ctrl+D key.
Password: ****
Confirm input: ****
The command completed successfully.
```

Setelah itu adalah membuat *certificate* SSL dengan perintah **ServerCertRegenerate CN** (*Common Name*). Untuk *common name* diisi dengan IP *public* pada Raspberry dan IP tersebut juga digunakan sebagai *gateway* untuk koneksi dari *client* ke *server*.

```
VPN Server/myHub>ServerCertRegenerate 192.168.43.21
ServerCertRegenerate command - Generate New Self-Signed Certificate with Specified CN
(Common Name) and Register on VPN Server
The command completed successfully.
```

Setelah *certificate* berhasil dibuat, selanjutnya simpan *certificate* tersebut ke dalam sebuah file agar dapat digunakan oleh *client* dengan perintah **ServerCertGet**.

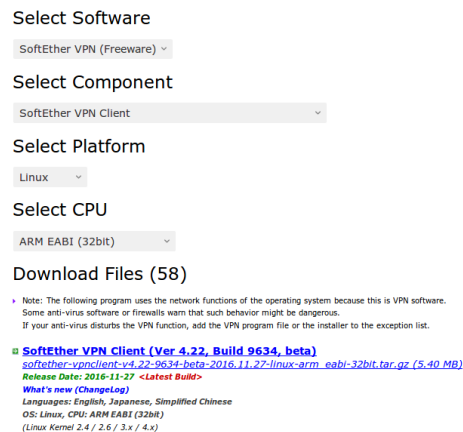
```
VPN Server/myHub>ServerCertGet ~/cert.cer
ServerCertGet command - Get SSL Certificate of VPN Server
The command completed successfully.
```

Setelah semuanya berhasil dibuat, selanjutnya adalah mengaktifkan *service* dari SSTP *server* dengan perintah **SstpEnable**.

```
VPN Server/myHub>SstpEnable yes
The command completed successfully.
```

2.3. Perancangan VPN SSTP Client

Setelah SSTP *server* telah dibuat, maka dilanjutkan konfigurasi pada sisi *client* agar dapat terhubung dengan *server* [Ocean, 2012]. Untuk membuat koneksi *client* ke *server* VPN SSTP adalah dengan menggunakan *software* **SoftEther VPN Client**.



Sumber : Ocean (2012)

Gambar 4. Softether VPN Client Download

Setelah didownload, masuk ke *directory* *Downloads* dan lakukan *extract file* dengan *command* **tar**.

```
root@ipin:/home/ipin/Downloads# tar zxvf softether-vpnclient-v4.22-
9634-beta-2016.11.27-linux-arm_eabi-32bit.tar.gz
root@ipin:/home/ipin/Downloads# cd vpnclient/
root@ipin:/home/ipin/Downloads/vpnclient# make
```

Setelah melakukan perintah *make*, ketik 1 sebanyak tiga kali untuk *read and accept License Agreement*. Lalu, pindahkan *vpnclient* ke *directory* lain dan ubah *permission* dari semua

file dengan permission 600 (*read, write*), sedangkan untuk file *vpnclient* dan *vpncmd* dengan permission 700 (*read, write, execute*).

```
root@ipin:/home/ipin/Downloads# cd /usr/local/vpnclient/
root@ipin:/usr/local/vpnclient# chmod 600 *
root@ipin:/usr/local/vpnclient# chmod 700 vpnclient
root@ipin:/usr/local/vpnclient# chmod 700 vpncmd
```

Lalu, *start service VPN client* dengan perintah ***.vpnclient start*** dan untuk melakukan *configure VPN client* menggunakan ***.vpnsmd*** kemudian pilih 2 untuk *management VPN client*. Tekan Enter untuk masuk ke *command line VPN client*.

```
root@raspin:/usr/local/vpnclient# ./vpnclient start
root@raspin:/usr/local/vpnclient# ./vpncmd
By using vpncmd program, the following can be achieved.
1. Management of VPN Server or VPN Bridge
2. Management of VPN Client
3. Use of VPN Tools (certificate creation and Network Traffic Speed Test Tool)

Select 1, 2 or 3: 2
Hostname of IP Address of Destination: [Enter]
Connected to VPN Client "localhost".
```

SoftEther menggunakan virtual adapter untuk membuat koneksi ke server VPN, gunakan perintah ***NicCreate*** untuk membuat virtual adaptor dengan nama ***myAdapter***.

```
VPN Client>NicCreate myAdapter
NicCreate command - Create New Virtual Network Adapter
The command completed successfully.
```

Selanjutnya adalah membuat koneksi VPN dengan perintah ***AccountCreate***. Koneksi yang dibuat adalah ***mySSTP***. Isikan IP *server* dan *port*, *virtual hub*, *username* dan virtual adaptor yang digunakan.

```
VPN Client>AccountCreate mySSTP
AccountCreate command - Create New VPN Connection Setting
Destination VPN Server Host Name and Port Number: 192.168.43.21:443

Destination Virtual Hub Name: myHub
Connecting User Name: sugiyatno
Used Virtual Network Adapter Name: myAdapter
The command completed successfully.
```

Koneksi VPN telah berhasil dibuat dan selanjutnya konfigurasi *password* untuk autentikasi *user* dengan perintah ***AccountPasswordSet***. Masukkan *password* sebanyak dua kali dan untuk autentikasi *password* isikan dengan *standard*.

```
VPN Client>AccountPasswordSet mySSTP
AccountPasswordSet command - Set User Authentication Type of VPN
Connection Setting to Password Authentication
Please enter the password. To cancel press the Ctrl+D key.

Password: *****
Confirm input: *****
Specify standard or radius: standard
The command completed successfully.
```

Selanjutnya tambahkan SSL *certificate* dengan menggunakan perintah ***CertAdd***. Sebelum ditambahkan, pindahkan terlebih dahulu *certificate* SSL ke *directory vpnclient* agar mudah untuk ditambahkan.

```
VPN Client>CertAdd cert.cer
CertAdd command - Add Trusted CA Certificate
The command completed successfully.
```

Setelah *certificate* berhasil ditambahkan, lakukan perintah ***AccountServerCertEnable*** untuk mengaktifkan *certificate verification* koneksi VPN SSTP dan pilih nama koneksi VPN yang telah dibuat sebelumnya yaitu ***mySSTP***.

```
VPN Client>AccountServerCertEnable
AccountServerCertEnable command - Enable VPN Connection Setting
Server Certificate Verification Option
Name of VPN Connection Setting: mySSTP
The command completed successfully.
```

Setelah semua berhasil dibuat, maka yang terakhir adalah melakukan koneksi VPN dengan perintah **AccountConnect**.

```
VPN Client>AccountConnect mySSTP
AccountConnect command - Start Connection to VPN Server using VPN Connection
Setting
The command completed successfully.
```

2.4. Perancangan VPNPPTP

Untuk pertama adalah melakukan instalasi *package* PPTP server dengan perintah [Domoticz, 2015]:

```
root@raspin:~# apt install pptpd
root@raspin:~# nano /etc/pptpd.conf
localip 192.168.20.1
remoteip 191.168.20.10-100
root@raspin:~# nano /etc/ppp/pptpd-options
```

Pada file pptpd-options, cari dan hilangkan tanda pagar “#” pada baris ms-dns dan tambahkan DNS Google atau DNS lainnya.

```
ms-dns 8.8.8.8
ms-dns 8.8.4.4
```

Setelah menambahkan DNS, selanjutnya adalah menambahkan *user* untuk autentikasi VPN PPTP pada file chap-secrets.

```
root@raspin:~# nano /etc/ppp/chap-secrets
upin * upinipin *
root@raspin:~# nano /etc/sysctl.conf
root@raspin:~# sysctl -p
root@raspin:~# service pptpd restart
```

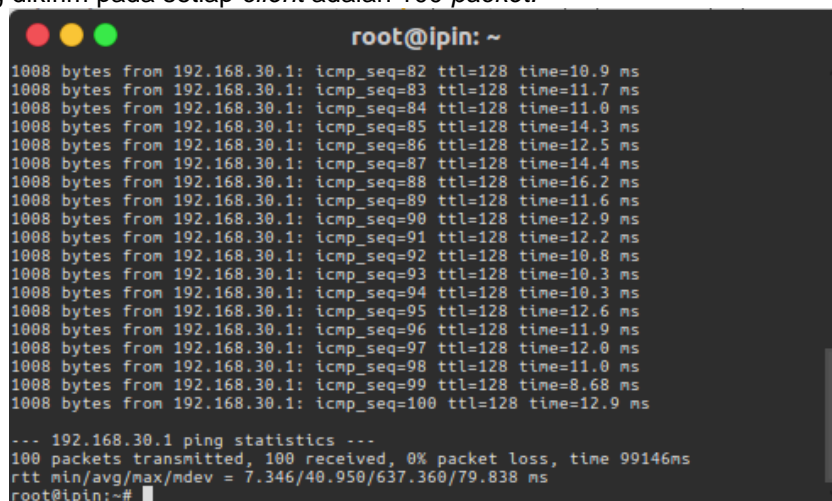
Parameter yang akan diuji pada pengujian performa adalah *packet loss*, *round trip time* dan *SFTP file transfer*.

3. Hasil dan Pembahasan

Pada pengujian akan menjelaskan konektivitas VPN, yang dilakukan menggunakan jaringan internet *provider* 3 (Tri) dengan *bandwidth* sesuai yang disediakan oleh *provider* tersebut. Sehingga kecepatannya tergantung kondisi cuaca dan tempat. Pengujian pada koneksi VPN SSTP dan PPTP dilakukan secara bergantian

1. Packet Loss

Pada pengujian ini bertujuan untuk mengetahui paket yang terkirim, paket yang diterima dan rata-rata *packet loss* yang melalui tunnel VPN. Pengujian dilakukan dengan cara *ping* melalui *Terminal*. Hal ini dilakukan pada kedua VPN SSTP dan PPTP dengan destinasi IP pada masing-masing VPN server. Di setiap VPN, pengujian dilakukan sebanyak 1 kali dengan jumlah *packet* yang dikirim pada setiap *client* adalah 100 *packet*.



```
root@ipin: ~
1008 bytes from 192.168.30.1: icmp_seq=82 ttl=128 time=10.9 ms
1008 bytes from 192.168.30.1: icmp_seq=83 ttl=128 time=11.7 ms
1008 bytes from 192.168.30.1: icmp_seq=84 ttl=128 time=11.0 ms
1008 bytes from 192.168.30.1: icmp_seq=85 ttl=128 time=14.3 ms
1008 bytes from 192.168.30.1: icmp_seq=86 ttl=128 time=12.5 ms
1008 bytes from 192.168.30.1: icmp_seq=87 ttl=128 time=14.4 ms
1008 bytes from 192.168.30.1: icmp_seq=88 ttl=128 time=16.2 ms
1008 bytes from 192.168.30.1: icmp_seq=89 ttl=128 time=11.6 ms
1008 bytes from 192.168.30.1: icmp_seq=90 ttl=128 time=12.9 ms
1008 bytes from 192.168.30.1: icmp_seq=91 ttl=128 time=12.2 ms
1008 bytes from 192.168.30.1: icmp_seq=92 ttl=128 time=10.8 ms
1008 bytes from 192.168.30.1: icmp_seq=93 ttl=128 time=10.3 ms
1008 bytes from 192.168.30.1: icmp_seq=94 ttl=128 time=10.3 ms
1008 bytes from 192.168.30.1: icmp_seq=95 ttl=128 time=12.6 ms
1008 bytes from 192.168.30.1: icmp_seq=96 ttl=128 time=11.9 ms
1008 bytes from 192.168.30.1: icmp_seq=97 ttl=128 time=12.0 ms
1008 bytes from 192.168.30.1: icmp_seq=98 ttl=128 time=11.0 ms
1008 bytes from 192.168.30.1: icmp_seq=99 ttl=128 time=8.68 ms
1008 bytes from 192.168.30.1: icmp_seq=100 ttl=128 time=12.9 ms
--- 192.168.30.1 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99146ms
rtt min/avg/max/mdev = 7.346/40.950/637.360/79.838 ms
root@ipin:~#
```

Sumber : Hasil Penelitian (2017)

Gambar 5. Pengujian *Packet Loss*

Tabel 1 adalah hasil pengujian dari *packet loss* pada VPN SSTP:

Tabel 1. Hasil *Packet Loss* VPN SSTP

| <i>Client</i> | Paket Terkirim | Paket Diterima | <i>Packet Loss (%)</i> |
|---------------|----------------|----------------|------------------------|
| 1 | 100 | 100 | 0 |
| 2 | 100 | 100 | 0 |
| 3 | 100 | 100 | 0 |
| 4 | 100 | 100 | 0 |
| 5 | 100 | 100 | 0 |

Sumber : Hasil Penelitian (2017)

Berdasarkan gambar 5 peroleh data seperti tabel 1 hasil pengujian *packet loss* yang dilakukan pada VPN SSTP, *client* yang melakukan *ping* terhadap VPN SSTP server mampu menyelesaikan *ping* tanpa ada *packet* yang hilang atau *loss*. Tabel 2 adalah hasil pengujian dari *packet loss* pada VPN PPTP:

Tabel 2. Hasil *Packet Loss* VPN PPTP

| <i>Client</i> | Paket Terkirim | Paket Diterima | <i>Packet Loss (%)</i> |
|---------------|----------------|----------------|------------------------|
| 1 | 100 | 96 | 4 |
| 2 | 100 | 100 | 0 |
| 3 | 100 | 99 | 1 |
| 4 | 100 | 98 | 2 |
| 5 | 100 | 100 | 0 |

Sumber : Hasil Penelitian (2017)

Berdasar tabel 2 hasil pengujian *packet loss* yang dilakukan pada VPN PPTP, *client* yang melakukan *ping* terhadap VPN PPTP server terdapat 3 *client* yang tidak mampu menyelesaikan *ping* karena adanya *packet* yang hilang atau *loss*. Berdasarkan kedua pengujian *packet loss*, VPN SSTP menunjukkan hasil yang lebih baik dibandingkan VPN PPTP, karena pada VPN SSTP tidak adanya *packet* yang hilang atau *loss*.

2. Round Trip Time

Pengujian ini bertujuan untuk mengetahui minimum, maksimum dan rata-rata waktu *round trip* pada *tunnel* VPN. Pengujian ini sebenarnya dilakukan bersamaan dengan *packet loss*, karena *round trip time* terdapat pada hasil *ping*. *Round trip time* merupakan paket *ping* yang melewati komputer *user*, sehingga IP *gateway* memberi respon balik ke komputer *user*. Pada pengujian ini, jumlah *packet* yang dikirim pada setiap *client* adalah sama dengan yang digunakan pada pengujian *packet loss* yaitu 100 *packet*.

```

root@ipin: ~
1008 bytes from 192.168.30.1: icmp_seq=82 ttl=128 time=10.9 ms
1008 bytes from 192.168.30.1: icmp_seq=83 ttl=128 time=11.7 ms
1008 bytes from 192.168.30.1: icmp_seq=84 ttl=128 time=11.0 ms
1008 bytes from 192.168.30.1: icmp_seq=85 ttl=128 time=14.3 ms
1008 bytes from 192.168.30.1: icmp_seq=86 ttl=128 time=12.5 ms
1008 bytes from 192.168.30.1: icmp_seq=87 ttl=128 time=14.4 ms
1008 bytes from 192.168.30.1: icmp_seq=88 ttl=128 time=16.2 ms
1008 bytes from 192.168.30.1: icmp_seq=89 ttl=128 time=11.6 ms
1008 bytes from 192.168.30.1: icmp_seq=90 ttl=128 time=12.9 ms
1008 bytes from 192.168.30.1: icmp_seq=91 ttl=128 time=12.2 ms
1008 bytes from 192.168.30.1: icmp_seq=92 ttl=128 time=10.8 ms
1008 bytes from 192.168.30.1: icmp_seq=93 ttl=128 time=10.3 ms
1008 bytes from 192.168.30.1: icmp_seq=94 ttl=128 time=10.3 ms
1008 bytes from 192.168.30.1: icmp_seq=95 ttl=128 time=12.6 ms
1008 bytes from 192.168.30.1: icmp_seq=96 ttl=128 time=11.9 ms
1008 bytes from 192.168.30.1: icmp_seq=97 ttl=128 time=12.0 ms
1008 bytes from 192.168.30.1: icmp_seq=98 ttl=128 time=11.0 ms
1008 bytes from 192.168.30.1: icmp_seq=99 ttl=128 time=8.68 ms
1008 bytes from 192.168.30.1: icmp_seq=100 ttl=128 time=12.9 ms

--- 192.168.30.1 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99146ms
rtt min/avg/max/mdev = 7.346/40.950/637.360/79.838 ms
root@ipin:~#

```

Sumber : Hasil Penelitian (2017)

Gambar 6. Pengujian *Round Trip Time*

Tabel 3 adalah hasil pengujian dari *round trip time* pada VPN SSTP:

Tabel 3. Hasil *Round Trip Time* VPN SSTP

| Client | Min. (ms) | Max. (ms) | Rata-rata(ms) |
|--------|-----------|-----------|---------------|
| 1 | 9,474 | 775,887 | 139,145 |
| 2 | 6,940 | 690,038 | 136,081 |
| 3 | 6,729 | 908,255 | 144,784 |
| 4 | 8,338 | 1154,305 | 204,539 |
| 5 | 6,599 | 1501,793 | 172,122 |

Sumber : Hasil Penelitian (2017)

Berdasarkan hasil pengujian *round trip time* pada VPN SSTP, menunjukkan bahwa waktu yang dimiliki pada setiap *client* tidaklah sama, baik dalam waktu minimum, maksimum, maupun rata-ratanya. Tabel 4 adalah hasil pengujian dari *round trip time* pada VPN PPTP:

Tabel 4. Hasil *Round Trip Time* VPN PPTP

| Client | Min. (ms) | Max. (ms) | Rata-rata(ms) |
|--------|-----------|-----------|---------------|
| 1 | 13.839 | 4347.726 | 560.845 |
| 2 | 14,057 | 8178,808 | 953,870 |
| 3 | 16,397 | 5136,419 | 1137,129 |
| 4 | 7,588 | 4209,528 | 692,039 |
| 5 | 10,543 | 7035,037 | 1576,147 |

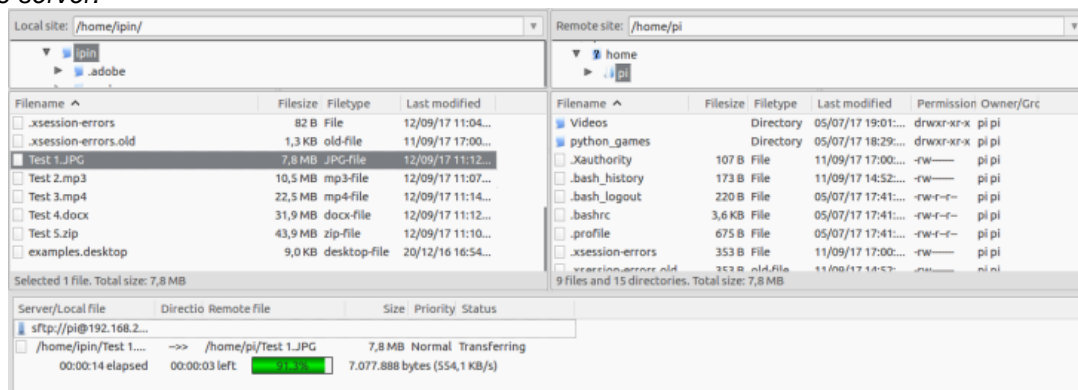
Sumber : Hasil Penelitian (2017)

Berdasarkan tabel 4 hasil pengujian *round trip time* yang dilakukan pada VPN PPTP, menunjukkan bahwa waktu yang dimiliki pada setiap *client* tidaklah sama, baik dalam waktu minimum, maksimum, maupun rata-ratanya.

Berdasarkan kedua pengujian *round trip time*, waktu paling minimum adalah *client* ke-5 pada VPN SSTP, ini menunjukkan VPN SSTP dapat berjalan sedikit lebih cepat dari VPN PPTP. Waktu paling maksimum adalah *client* ke-2 pada VPN PPTP, ini menunjukkan VPN PPTP dapat berjalan sedikit lebih lama dari VPN SSTP. Sedangkan untuk rata-rata, dari semua percobaan *client*, hasil dari VPN SSTP menunjukkan waktu yang lebih cepat daripada VPN PPTP.

3. SFTP File Transfer

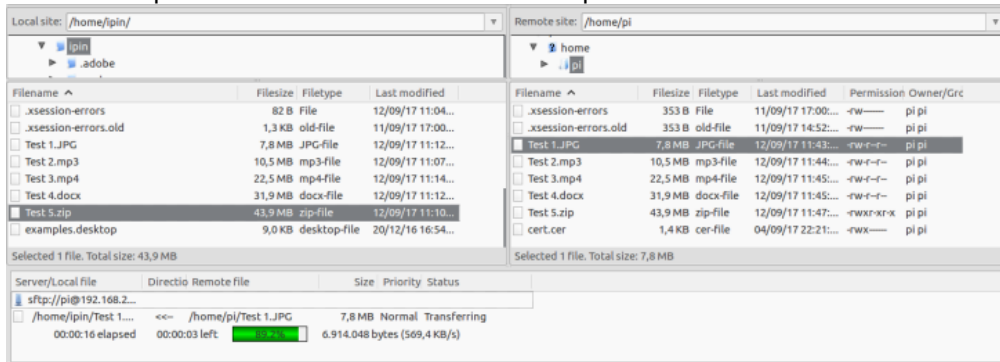
Pengujian ini bertujuan untuk mengetahui waktu yang dibutuhkan untuk transfer file menggunakan SFTP melalui *tunnel* VPN, baik itu *download* maupun *upload* dan dilakukan pada setiap koneksi VPN. Dalam pengujiannya menggunakan beberapa file dengan ekstensi dan ukuran yang berbeda, yaitu file gambar (.jpg) berukuran 7.8MB, audio (.mp3) berukuran 10.5MB, video (.mp4) berukuran 22.5MB, *document* (.docx) berukuran 31.9MB dan *compress file* (.zip) berukuran 43.9MB. File tersebut akan diuji menggunakan *software FileZilla File Transfer* pada komputer *client*. Proses pengujian menggunakan 1 *client* dan dilakukan sebanyak 3 kali yang kemudian akan diambil rata-rata dari ketiga percobaan tersebut sebagai pembandingan antara VPN SSTP dan PPTP. Gambar 7 adalah proses *upload* dari komputer *client* ke *server*.



Sumber : Hasil Penelitian (2017)

Gambar 7. Proses *Upload* melalui SFTP

Gambar 8 adalah proses *download* dari server ke komputer *client*.



Sumber : Hasil Penelitian (2017)

Gambar 8. Proses *Download* melalui SFTP

Tabel 5 adalah hasil *upload* pada VPN SSTP dan VPN PPTP:

Tabel 5. Upload Rate pada VPN SSTP

| Percobaan | Speed (KB/s) | | | | |
|-----------|--------------|-------|-------|----------|---------------|
| | Gambar | Audio | Video | Document | Compress File |
| 1 | 611,2 | 509,6 | 749 | 986,1 | 613,3 |
| 2 | 431,9 | 660,4 | 492,2 | 573,7 | 638,8 |
| 3 | 572,3 | 559,8 | 460,1 | 582,7 | 1.100 |
| Rata-rata | 538,5 | 576,6 | 567,1 | 714,2 | 784 |

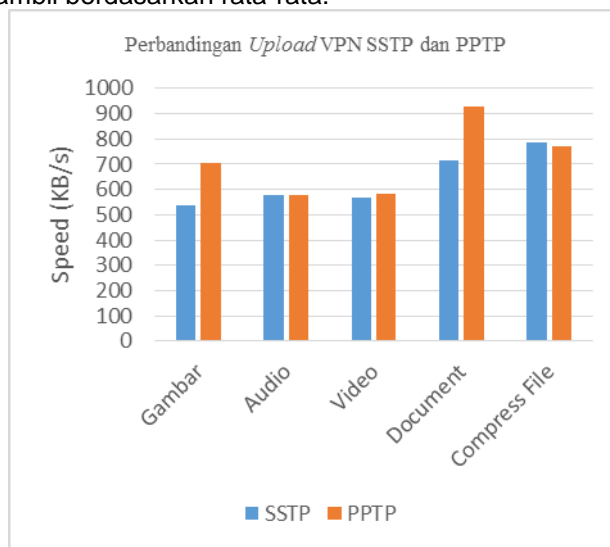
Sumber : Hasil Penelitian (2017)

Tabel 6. Upload Rate pada VPN PPTP

| Percobaan | Speed (KB/s) | | | | |
|-----------|--------------|-------|-------|----------|---------------|
| | Gambar | Audio | Video | Document | Compress File |
| 1 | 554,1 | 594,7 | 542,1 | 888,3 | 743,9 |
| 2 | 819,5 | 564,5 | 451,2 | 1.200 | 949 |
| 3 | 736,8 | 573,2 | 762 | 688,6 | 612,9 |
| Rata-rata | 703,5 | 577,5 | 585,1 | 925,6 | 768,6 |

Sumber : Hasil Penelitian (2017)

Gambar. 9 adalah hasil perbandingan *upload rate* antara VPN SSTP dan PPTP dalam bentuk grafik yang diambil berdasarkan rata-rata:



Sumber : Hasil Penelitian (2017)

Gambar 9. Grafik Perbandingan *Upload Rate* VPN SSTP dan PPTP

Berdasarkan gambar 9, dapat dilihat bahwa *upload rate* VPN PPTP lebih cepat dibanding VPN SSTP. Kecuali pada *compress file* milik VPN SSTP lebih cepat dibanding VPN PPTP. Tabel 7 adalah hasil *download* pada VPN SSTP dan VPN PPTP:

Tabel 7. Download Rate pada VPN SSTP

| Percobaan | Speed (KB/s) | | | | |
|-----------|--------------|-------|-------|----------|---------------|
| | Gambar | Audio | Video | Document | Compress File |
| 1 | 672,7 | 455,8 | 383,7 | 363,6 | 392,5 |
| 2 | 445 | 430,8 | 410 | 429,9 | 403,8 |
| 3 | 443,7 | 397,6 | 471,3 | 423,6 | 430 |
| Rata-rata | 520,5 | 428,1 | 421,7 | 405,7 | 408,8 |

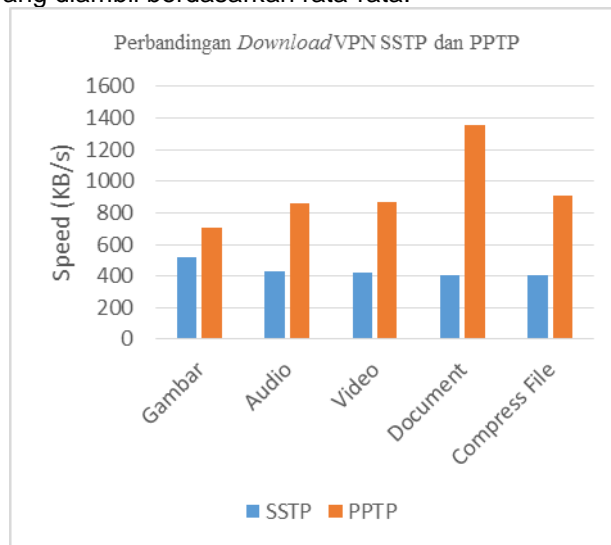
Sumber Hasil Penelitian (2017)

Tabel 8. Download Rate pada VPN PPTP

| Percobaan | Speed (KB/s) | | | | |
|-----------|--------------|-------|-------|----------|---------------|
| | Gambar | Audio | Video | Document | Compress File |
| 1 | 569,4 | 810,2 | 1.200 | 1.400 | 710,7 |
| 2 | 910 | 849,1 | 711,7 | 954,2 | 1.500 |
| 3 | 640,2 | 921,4 | 700,2 | 1.700 | 525,1 |
| Rata-rata | 706,5 | 860,2 | 870,6 | 1.351,4 | 911,9 |

Sumber : Hasil Penelitian (2017)

Gambar 10 adalah hasil perbandingan *download rate* antara VPN SSTP dan PPTP dalam bentuk grafik yang diambil berdasarkan rata-rata:



Sumber : Hasil Penelitian (2017)

Gambar 10. Grafik Perbandingan *Download Rate* VPN SSTP dan PPTP

Berdasarkan gambar 10, dapat dilihat bahwa *download rate* VPN PPTP lebih cepat daripada VPN SSTP pada semua jenis *file* yang diuji.

4. Kesimpulan

Berdasarkan hasil penelitian dari beberapa pengujian yang telah dilakukan, maka dapat diambil beberapa kesimpulan bahwa (1) VPN SSTP dengan Raspberry Pi sebagai *server* dan PC dengan sistem operasi Ubuntu sebagai *client* telah berhasil dibuat, sehingga *client* dapat terkoneksi dengan VPN *server*, (2) Hasil pengujian performa VPN SSTP menunjukkan bahwa performa dari VPN SSTP cukup baik, terutama pada pengujian *packet loss* dan *round trip time*. Sedangkan hasil pengujian keamanan VPN SSTP menunjukkan bahwa VPN SSTP aman terhadap serangan *sniffing*, hal itu ditunjukkan pada hasil yang didapat dimana *username* dan

password yang digunakan untuk *login* tidak dapat diketahui oleh *attacker* dan (3) Hasil perbandingan antara VPN SSTP dan PPTP, pada pengujian performa, VPN PPTP lebih cepat dalam pengujian SFTP *file transfer*, baik *upload* maupun *download*. Sedangkan pada pengujian keamanan, *username* dari VPN PPTP dapat diketahui oleh *attacker*, tetapi *password*-nya tidak dapat terbaca karena terenkripsi oleh MS-CHAPv2.

Referensi

- Badrul M. 2016. Open VPN-Access Server Dengan Enkripsi SSL / TI Open SSL. *Informatics For Educator And Professional*. 1(1): 1–12.
- Brenton, Chris dan Hunt C. 2016. *Network Security 2nd edition*. Jakarta: Elex Media Computindo.
- Dinata A. 2017. *Physical Computing dengan Raspberry Pi*. Jakarta.: Elex Media Komputindo.
- Domoticz. 2015. Installing a PPTP-VPN Server on a Raspberry Pi. Mediowiki. https://www.domoticz.com/wiki/Installing_a_PPTP-VPN_server_on_a_Raspberry_Pi. Diakses pada tanggal 17 April 2018.
- Ocean D. 2012. How to Setup a Multi-Protocol VPN Server Using SoftEther. DigitalOcean. <https://www.digitalocean.com/community/tutorials?q=vpn>. Diakses pada tanggal 17 April 2018.
- Oktivasari P, Utomo AB. 2016. Analisa Virtual Private Network Open VPN dan Point to Point Tunneling Protocol. *Jurnal Penelitian Komunikasi dan Opini Publik*. 20(2): 185–202.
- Oppenheimer P. 2011. *Top-Down Network Design*. Indianapolis: Cisco Press.
- Sirisukha S. 2003. The Advantages A Virtual Private Network For Computer Security. *Proceedings of the 16th Annual NACCQ*. Palmerston North New Zeland. 397–402
- Yang Y. 2011. *Virtual Private Network Management*. Bachelor ' s Thesis. University of Technology Sydney.