

ANALISIS PERBANDINGAN KECEPATAN DOWNLOAD FILE PADA BERBAGAI PROTOKOL JARINGAN

Wagito

STMIK Akakom, Yogyakarta
e-mail: wagito@akakom.ac.id

ABSTRAK

Download file memerlukan koneksi terus-menerus yang berpotensi menghabiskan bandwidth jaringan. Hal ini dapat mengganggu aktivitas Internet lain. Oleh karena itu perlu ditentukan cara paling baik dalam melakukan download file. Pemilihan protokol jaringan paling cepat dalam melakukan download file menjadi sesuatu yang penting. Penelitian bertujuan untuk membandingkan kecepatan download file menggunakan bermacam-macam protokol jaringan. Download file dapat dilakukan menggunakan bermacam-macam protokol jaringan. Protokol jaringan yang biasa dipakai untuk download file adalah HTTP, HTTPS, FTP, SFTP, SCP dan TFTP. Masing-masing protokol jaringan punya perilaku yang berbeda dalam melakukan proses download file. Suatu protokol perlu proses login sebelum download, Protokol lain memerlukan proses enkripsi pada saat proses download. Secara keseluruhan, perilaku tersebut akan mempengaruhi kecepatan dalam melakukan proses download file. Protokol TFTP menunjukkan hasil kecepatan paling tinggi. Untuk protokol yang umum dipakai pada jaringan Internet, protokol SCP menunjukkan kecepatan download file paling tinggi.

Kata kunci: kecepatan, download, protokol

ABSTRACT

Files download requires a continuous connection potentially spend network bandwidth. It can interfere with other Internet activity. Therefore, it is necessary to determine the best way to download files. Selection of the most rapid network protocol to download files become something important. The study aims to compare the speed of files download using a variety of network protocols. Files download can be done using a variety of network protocols. Network protocols used for files downloads are HTTP, HTTPS, FTP, SFTP, SCP, and TFTP. Each network protocol has a different behavior in the process of files download. A protocol needs to process log in before downloading, other protocols need encryption process during the download process. Overall, the behavior will affect the speed in the process of files downloading. TFTP protocol indicates the highest speed. For common protocol used on the Internet network, SCP protocol shows the highest speed of the file download.

Keywords: speed, download, protocol

I. PENDAHULUAN

Jaringan Internet sudah menjadi kebutuhan kebanyakan orang. Hampir seluruh sektor kehidupan tidak terlepas dari peran jaringan Internet. Orang menggunakan jaringan Internet untuk berbagai keperluan mulai dari mencari informasi, menyebarkan informasi, mengirim e-mail, menerima e-mail, berbincang secara online, melakukan *download file*, melakukan *upload file*, dan sebagainya. Bahkan keperluan melihat acara televisi, mendengarkan siaran radio, mendengarkan alunan musik, menggunakan telepon serta melihat video juga dapat dilakukan melalui jaringan Internet. Internet sudah menguasai banyak perilaku kehidupan manusia.

Masing-masing aktivitas dalam berhubungan dengan jaringan Internet memerlukan protokol jaringan. Untuk mencari informasi biasanya dilakukan dengan aktivitas *browsing*. Menurut kamus istilah komputer dan informatika [2], *browsing* adalah aktivitas menjelajah beberapa situs di Internet. Aktivitas *browsing* menjadi aktivitas yang paling banyak dilakukan pada saat berhubungan dengan jaringan Internet. Aktivitas *browsing* umumnya menggunakan protokol jaringan HTTP dan HTTPS. Protokol adalah sekelompok fungsi yang saling terkait yang diperlukan untuk melakukan fungsi komunikasi. Protokol suite dilaksanakan oleh *host* dan perangkat jaringan baik perangkat lunak, perangkat keras atau keduanya [4]. Protokol HTTP umumnya menggunakan *port* nomor 80 sedangkan protokol HTTPS umumnya menggunakan *port* nomor 443. Informasi yang ditampilkan umumnya berbentuk dokumen teks HTML.

Aktivitas yang umumnya menyertai *browsing* adalah *download file*. Menurut kamus istilah komputer dan informatika [2], *download* adalah mengambil *file* atau melakukan transfer *file* dari satu

komputer ke komputer lainnya. *Download* dilakukan untuk mengambil *file* dari suatu situs penyedia untuk disimpan pada komputer. Aktivitas *download* biasa dipakai untuk mengambil *file* musik, video, buku, perangkat lunak dan sebagainya. *file* musik yang biasa *download* misalnya *file* mp3. *file* video yang biasa *download* misalnya mp4. *file* buku yang biasa *download* umumnya berbentuk pdf. *file* perangkat lunak biasa *download* dalam bentuk exe. Selain itu, *file* juga dapat diwujudkan dalam bentuk *file* terkompresi zip.

Aktivitas *download* dapat dilakukan menggunakan beberapa protokol misalnya HTTP, HTTPS, FTP, SCP, TFTP dan sebagainya. Masing-masing punya perilaku yang berbeda-beda dalam melakukan proses *download*. Protokol HTTP dan HTTPS biasanya dapat digunakan untuk *download file* tanpa melakukan *login*. Protokol FTP biasanya digunakan untuk proses *download file* melalui proses *login*, walaupun bisa diatur tanpa melakukan proses *login*. Protokol SCP biasa digunakan untuk proses *download file* dengan melibatkan faktor keamanan data yaitu enkripsi dan dekripsi.

Versi pertama HTTP, disebut sebagai HTTP/0.9, adalah sebuah protokol sederhana untuk transfer data mentah di Internet. HTTP/1.0, seperti yang didefinisikan oleh RFC 1945 [5]. Protokol HTTP yang sekarang banyak dipakai adalah versi 1.1. Protokol ini mencakup kebutuhan yang lebih ketat dibanding HTTP/1.0 untuk menjaga kelayakan implementasi masa depan [9]. HTTP [RFC2616] pada awalnya digunakan secara luas di Internet. Namun, peningkatan penggunaan HTTP untuk aplikasi yang sensitif diperlukan langkah-langkah keamanan. SSL, dan penggantinya TLS [RFC2246] dirancang untuk memberikan keamanan *channel-oriented*. [12].

Protokol SSH pada mulanya digunakan untuk *login* dalam suatu komputer dari jauh dan mengeksekusi perintah pada komputer tersebut. Dalam perkembangannya, protokol SSH dapat digunakan untuk *download*. Program ssh mirip dengan telnet, tetapi punya pengendalian terhadap keamanan. Program ssh menyediakan komunikasi aman terenkripsi antara dua host yang tidak saling kenal melalui jaringan yang umumnya tidak aman. Penerapan program sshd mendukung protokol SSH versi 1 dan versi 2 secara serentak. Protokol SSH diatur pada RFC 4250, 4251, 4252, 4253 dan 4254 [10][18][19][20][21]

FTP adalah protokol yang bertujuan untuk mempromosikan berbagi *file*, untuk mendorong penggunaan tak langsung atau implisit (melalui program) dari komputer *remote*, untuk melindungi pengguna dari variasi dalam sistem penyimpanan *file* antar host, dan untuk melakukan transfer data andal dan efisien [11]. Meski pun dapat digunakan secara langsung oleh pengguna pada terminal, FTP dirancang terutama untuk digunakan oleh program.

TFTP adalah protokol yang sangat sederhana yang digunakan untuk melakukan transfer *file*. Setiap paket *non-terminal* diakui secara terpisah. Protokol ini awalnya dirancang oleh Noel Chiappa, dan dirancang ulang oleh dia, Bob Baldwin dan Dave Clark, dengan komentar dari Steve Szymanski. Revisi dilakukan saat dokumentasi termasuk modifikasi yang berasal dari diskusi dengan dan saran dari Larry Allen, Noel Chiappa, Dave Clark, Geoff Cooper, Mike Greenwald, Liza Martin, David Reed, Craig Milo Rogers (USC-ISI) dan Kathy Yellick. Pengakuan dan transmisi skema terinspirasi oleh TCP, dan mekanisme error disarankan oleh pesan EFTP batalkan PARC [13].

Seringkali suatu situs Internet menyediakan beberapa cara (menggunakan beberapa protokol) untuk menyediakan fasilitas *download* bagi para penggunanya. Misalnya saja situs www.kernel.org (2014) yang menyediakan perangkat lunak kernel sistem operasi Linux. Situs menyediakan protokol HTTP, FTP dan RSYNC untuk melakukan *download*. Contoh lain adalah situs kambing.ui.ac.id (2014) milik Universitas Indonesia. Situs menyediakan protokol HTTP dan FTP untuk *download file*. Pada situs lain, *download* disediakan dengan variasi protokol jaringan yang lain. Secara umum, situs penyedia *file* biasanya menyediakan beberapa protokol untuk melakukan proses *download file*.

Jika tersedia beberapa protokol jaringan untuk proses *download file*, maka pengguna dapat menggunakan salah satu. Pemilihan protokol yang dipakai untuk *download file* umumnya berdasarkan kecepatan *download*. Pertimbangan kecepatan bisa dipakai apabila jaringan Internet tidak melakukan blok pada protokol jaringan tertentu. Karena alasan keamanan, suatu sistem tidak mengizinkan akses ke jaringan Internet menggunakan suatu protokol tertentu.

Proses *download file* memerlukan koneksi terus-menerus (*continuous connection*). Koneksi terus-menerus punya potensi untuk menghabiskan *bandwidth* jaringan. Dengan demikian proses *download* dapat mengganggu aktivitas Internet lain seperti *browsing*. Oleh karena itu perlu ditentukan pemilihan protokol yang paling cepat dalam melakukan *download*.

Pada penelitian dicoba untuk membandingkan kecepatan *download file* menggunakan bermacam-macam protokol jaringan yang umum dipakai untuk proses *download file* yaitu HTTP, HTTPS, FTP, SCP dan TFTP. Setelah dibandingkan, tentunya akan diketahui protokol jaringan yang paling cepat

dalam melakukan *download file*. Untuk melakukan proses *download file*, digunakan perangkat lunak bantu Curl yang mendukung beberapa protokol jaringan. Untuk mengamati proses *download file*, disusun algoritme untuk mengamati kecepatan *download file* menggunakan berbagai protokol.

Pengamatan terhadap pengaruh penggunaan protokol jaringan terhadap kecepatan *download* harus dilakukan tanpa pembatasan *bandwidth* jaringan. Pengamatan kecepatan *download* melalui jaringan Internet, tidak dapat menampilkan hasil yang benar, karena adanya pembatasan *bandwidth*. Dengan demikian, perlu disusun suatu sistem jaringan yang dilengkapi dengan *server* yang mendukung bermacam-macam protokol jaringan.

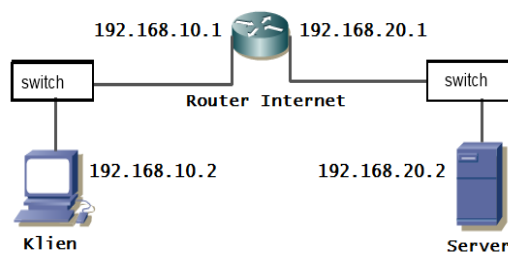
Tujuan yang ingin dicapai dari hasil penelitian yang dilakukan adalah menyusun algoritme untuk mengamati kecepatan *download* menggunakan berbagai protokol, membandingkan kecepatan *download* menggunakan bermacam-macam protokol jaringan dan mengetahui protokol jaringan yang paling cepat dalam proses *download*. Manfaat penelitian adalah dapat mengetahui protokol jaringan yang paling cepat dalam melakukan *download file*. Pengetahuan ini sangat bermanfaat untuk menentukan pemilihan protokol paling baik untuk proses *download* dari beberapa situs Internet. Pengetahuan ini juga sangat bermanfaat untuk menentukan pilihan protokol jaringan pada pengelolaan sistem *server* penyedia layanan *file*.

II. METODE PENELITIAN

Konfigurasi jaringan yang diperlukan dalam penelitian melibatkan satu *server*, satu klien, satu *router* dan sistem pengkabelan. Pada penelitian, sistem pengkabelan diabaikan karena komputer *server* dan *router* dijalankan pada VirtualBox. *Server* dan *router* dijalankan sebagai komputer virtual, sedangkan klien berupa komputer riil.

Pengaturan *device network* pada komputer *server* menggunakan Internal Network dan dihubungkan pada *switch* pertama. Pengaturan *device network* pertama pada *router* menggunakan Internal Network dan dihubungkan pada *switch* pertama, sedangkan *device network* kedua menggunakan Host-only Adapter. Pengaturan ini dan alamat IP sesuai dengan diagram jaringan yang ditunjukkan pada Gambar 1.

Antarmuka jaringan pada *server* diberi alamat IP 192.168.20.2/24. Klien berlaku sebagai tempat untuk mengukur kecepatan *download file*. Antarmuka pada klien diberi alamat IP 192.168.10.2/24. *Router* berlaku sebagai simulasi jaringan Internet. Supaya tidak terjadi proses translasi alamat jaringan, pada *router* tidak diberlakukan sebagai Firewall. Hal demikian dimaksudkan untuk meminimalkan pengaruh selain faktor tipe protokol. Pada *router* hanya terjadi proses routing paket. Antarmuka pada *router* yang berada satu jaringan dengan klien diberi alamat IP 192.168.10.1/24, sedangkan antarmuka pada *router* yang berada satu jaringan dengan komputer *server* diberi alamat IP 192.168.20.1/24.



Gambar 1. Rancangan Jaringan

Komputer *server* disiapkan untuk melayani permintaan *download* dari komputer klien. Komputer *server* diatur supaya berada pada jaringan yang sama dengan komputer klien. Hal demikian untuk mengurangi pengaruh translasi alamat jaringan pada proses *download*. Dengan demikian, diharapkan proses *download* hanya dipengaruhi oleh perbedaan tipe protokol yang dipakai. Komputer *server* menggunakan sistem operasi Linux *distro* Mandriva 2010. Komputer *Server* diberi alamat IP 192.168.20.2/24 dan alamat IP Gateway 192.168.20.1/24.

Server adalah komputer yang dirancang untuk memproses permintaan dan mengirimkan data ke komputer lain melalui jaringan lokal atau Internet. *Server* jaringan biasanya dikonfigurasi dengan pemroses, memori dan kapasitas penyimpanan tambahan untuk menangani beban permintaan klien [3]. Supaya kerja *server* lebih ringan, mode kerja yang diterapkan pada *server* adalah mode teks. Seluruh layanan bekerja pada mode teks dan bekerja secara latar. Layanan yang disiapkan pada *server*

adalah layanan *download*. Komputer *server* melayani *download file* menggunakan protokol HTTP, HTTPS, FTP, SFTP, SCP dan TFTP.

Layanan protokol HTTP ditangani oleh perangkat lunak Apache. Versi Apache yang digunakan adalah 2.2.4. Program Apache merupakan standar program untuk *server* HTTP pada sistem operasi Linux. Program Apache berkembang sedemikian, sehingga sampai versi terakhir mendukung banyak hal yang diperlukan koneksi HTTP. Informasi server Apache dapat dilihat pada dokumentasi [14].

Layanan protokol HTTPS ditangani oleh perangkat lunak Nginx. Versi Nginx yang digunakan adalah 0.8.17. Nginx memiliki satu proses master dan beberapa proses pekerja. Tujuan utama dari proses master adalah membaca dan mengevaluasi konfigurasi, serta memelihara proses pekerja. Proses pekerja yang sebenarnya melakukan proses terhadap permintaan. Nginx mempekerjakan mekanisme model *event-based* dan *OS-dependent* efisien mendistribusikan permintaan antara proses pekerja. Jumlah pekerja proses ditentukan dalam *file* konfigurasi dan dapat tetap untuk konfigurasi tertentu atau secara otomatis disesuaikan jumlah *core* CPU yang tersedia [1]

Layanan protokol FTP ditangani oleh perangkat lunak ProFTPD [15]. Versi Proftpd yang digunakan adalah 1.3.3. *Server* FTP adalah program yang dapat melayani permintaan koneksi melalui protokol FTP. Klien FTP adalah program yang dapat digunakan untuk melakukan koneksi melalui protokol FTP.

Layanan protokol SCP ditangani *server* SSH. Perangkat lunak *server* SSH yang dipakai adalah OpenSSH. Versi OpenSSH yang dipakai adalah 5.3. Program sshd (SSH Daemon) adalah *daemon* untuk program *server* ssh (*secure shell*). Program sshd bekerja pada *port* 22. Program ssh digunakan sebagai klien untuk *login* dalam suatu komputer dari jauh dan mengeksekusi perintah pada komputer tersebut.

Layanan protokol TFTP ditangani oleh Tftpd. Versi Tftpd yang dipakai adalah 5.0.3. Untuk menjalankan *server* Tftpd digunakan *server* Xinetd. Protokol TFTP banyak digunakan untuk mendukung *booting* perangkat *remote diskless*. Penggunaan layanan TFTP tidak memerlukan *account* atau *password* pada sistem *server*.

Router Internet digunakan sebagai *router* penghubung dari jaringan *server* dengan jaringan klien. *Router* Internet sepenuhnya difungsikan hanya sebagai *router*, tidak difungsikan sebagai Firewall. Hal demikian dimaksudkan untuk menghilangkan pengaruh translasi alamat jaringan dalam penelitian ini.

Pada klien dipasang perangkat lunak curl. Perangkat lunak ini digunakan untuk mentransfer data dari atau ke *server*, menggunakan salah satu protokol yang didukung (HTTP, HTTPS, FTP, FTPS, SCP, SFTP, TFTP, DICT, TELNET, LDAP atau FILE). Perintah dirancang untuk bekerja tanpa interaksi pengguna. Curl menawarkan fitur-fitur yang berguna seperti dukungan proxy, autentikasi, *upload* FTP, HTTP post, koneksi SSL, cookies, resume transfer *file* dan sebagainya. Curl didukung oleh libcurl untuk semua fitur transfer terkait [7].

Antarmuka jaringan yang menghadap jaringan klien yaitu ether1 diberi alamat IP 192.168.10.1/24. Alamat IP ini merupakan alamat IP Gateway komputer klien. Antarmuka jaringan yang menghadap jaringan *server* yaitu ether2 diberi alamat IP 192.168.20.1/24. Alamat IP ini menjadi alamat Gateway komputer *server*.

Pada penelitian digunakan beberapa *file* dengan ukuran dan tipe yang berbeda. Uji coba transfer data dilakukan pada dua jenis *file* yaitu *file* teks dan *file* gz. *File* teks mewakili transfer data untuk *file* yang belum mengalami kompresi, sedangkan *file* gz mewakili transfer data yang sudah mengalami kompresi. Ukuran *file* dipilih yang cukup besar. Sedemikian, sehingga kecepatan transfer data dapat diamati lebih baik.

Pada percobaan *download file*, dicoba dilakukan *download* terhadap *file* teks (belum mengalami kompresi) dan *file* gz (sudah mengalami kompresi). Masing-masing sebanyak lima *file* dengan ukuran yang berbeda. Tiap-tiap *file* dilakukan pengambilan data sebanyak lima puluh kali. Pencuplikan data dilakukan pada berbagai tipe protokol, tipe *file* dan ukuran *file*. Hal demikian akan cukup merepotkan apabila pencuplikan dilakukan manual satu demi satu, Untuk memudahkan pada pencuplikan data kecepatan *download file*, dibuat skrip Bash sebagai berikut.

```
aprotocol="http https ftp scp sftp\  
tftp"  
afile="file0.txt file1.txt\  
file2.txt file3.txt\  
file4.txt file1.gz file2.gz\  
file3.gz file4.gz file5.gz"  
  
server="192.168.20.2"  
user="xxx"
```

```

pass="yyy"
tmpfile="tmp.txt"

#variasi protocol jaringan
for protocol in `echo $aprotocol`
do
  dev="data_download-$protocol.csv"
  #dev="/dev/stdout"

  echo `date` > $dev

  sleep 40

#variasi tipe file
for file in `echo $afile`
do
  echo "$protocol;$file;"
  echo -n "$protocol;$file;"
  >>$dev

#jumlah eksperimen
for exp in {1..10}
do
  #hapus file
  rm -f *.txt >/dev/null
  rm -f *.gz >/dev/null

#download file & membaca rate
if [ $protocol='http' ]
then
  curl -o $file \
  http://$server/file/$file \
  2> $tmpfile
elif [ $protocol='https' ]
then
  curl -o $file \
  http://$server:443/file/\
  $file 2> $tmpfile
elif [ $protocol='ftp' ]
then
  curl -o $file \
  ftp://$user:\
  $pass@$server/file/$file\
  2> $tmpfile
elif [ $protocol='scp' ]
then
  curl scp://$user:\
  $pass@$server/home/\
  $user/file/$file 2> $tmpfile
elif [ $protocol='sftp' ]
then
  curl sftp://$user:\
  $pass@$server/home/\
  $user/file/$file 2>\
  $tmpfile
elif [ $protocol='tftp' ]
then
  curl -o $file tftp://\
  $server/$file 2> $tmpfile
fi
dos2unix $tmpfile
rate=`tail -1 $tmpfile\
| awk '{print $7}'`
echo -n "$rate;" >>$dev
done
echo >>$dev
done
done
done

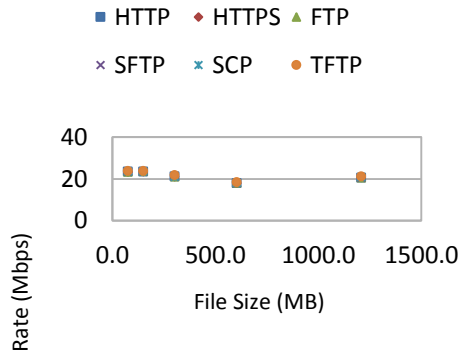
```

Skrip program yang ditampilkan sedikit dimodifikasi dari program sesungguhnya untuk menyembunyikan *username* dan *password*. Hal demikian untuk menjaga keamanan server yang dipakai untuk penelitian. *Username* yang dituliskan dalam program adalah xxx, sedangkan *password* yang dituliskan adalah yyy.

III. HASIL DAN PEMAHASAN

Pengaruh perbedaan protokol jaringan pada kecepatan *download file* ditunjukkan pada Gambar 2.

Proses *download file* dilakukan menggunakan protokol HTTP, HTTPS, FTP, SFTP, SCP dan TFTP.



Gambar 2. Pengaruh Protokol Pada Download File

Pengaruh protokol jaringan pada proses *download file* menunjukkan bahwa tidak ada perbedaan secara signifikan antar protokol yang diuji. Namun demikian dapat dipetakan untuk menentukan kecenderungan pengaruh protokol jaringan yang lebih cepat. Pengaruh protokol jaringan pada kecepatan *download file* dapat ditunjukkan pada Tabel 1. Kolom pertama menunjukkan variasi ukuran *file*, sedangkan baris pertama menunjukkan variasi protokol jaringan yang digunakan untuk uji coba.

TABEL I.
PEMETAAN PENGARUH PROTOKOL JARINGAN KECEPATAN

~~| File Size (MB) | HTTP (Mbps) | HTTPS (Mbps) | FTP (Mbps) | SFTP (Mbps) | SCP (Mbps) | TFTP (Mbps) |
|----------------|-------------|--------------|------------|-------------|------------|-------------|
| ~100 | ~22 | ~22 | ~22 | ~22 | ~22 | ~22 |
| ~250 | ~20 | ~20 | ~20 | ~20 | ~20 | ~20 |
| ~500 | ~18 | ~18 | ~18 | ~18 | ~18 | ~18 |
| ~1200 | ~20 | ~20 | ~20 | ~20 | ~20 | ~20 |~~

Tabel 1 menunjukkan bahwa protokol jaringan yang paling cepat dalam proses *download file* adalah TFTP. Protokol TFTP menggunakan datagram dalam koneksi jaringan. Protokol datagram lebih ringan dalam koneksi jaringan karena tidak memerlukan adanya gandengan. Namun protokol TFTP umumnya dipakai untuk protokol boot yaitu proses booting melalui jaringan dan jarang digunakan untuk *download file* lewat jaringan Internet.

Apabila protokol TFTP ditiadakan dalam uji kecepatan *download file*, maka hasilnya ditunjukkan pada Tabel 2.

TABEL II.
UJI COBA MENGABAIKAN PROTOKOL TFTP

~~| File Size (MB) | HTTP (Mbps) | HTTPS (Mbps) | FTP (Mbps) | SFTP (Mbps) | SCP (Mbps) | TFTP (Mbps) |
|----------------|-------------|--------------|------------|-------------|------------|-------------|
| ~100 | ~22 | ~22 | ~22 | ~22 | ~22 | ~22 |
| ~250 | ~20 | ~20 | ~20 | ~20 | ~20 | ~20 |
| ~500 | ~18 | ~18 | ~18 | ~18 | ~18 | ~18 |
| ~1200 | ~20 | ~20 | ~20 | ~20 | ~20 | ~20 |~~

Tabel 2 menunjukkan bahwa kecepatan *download file* yang paling besar adalah cenderung pada penggunaan protokol SCP. Protokol ini dilayani oleh *server* SSH. Protokol SCP adalah jenis protokol yang dilengkapi fitur keamanan.

Dengan demikian, antar protokol jaringan yang diuji, tidak terlihat perbedaan yang signifikan dalam proses *download file*. Namun demikian, protokol yang cenderung lebih cepat dalam proses *download file* adalah TFTP. Jika percobaan hanya mempertimbangkan protokol yang umum dipakai untuk jaringan Internet, maka protokol SCP cenderung memberikan nilai kecepatan yang paling besar.

IV. SIMPULAN

Penelitian berhasil disusun metode untuk mengamati kecepatan *download file* menggunakan berbagai variasi protokol jaringan. Pengaruh protokol jaringan pada proses *download file* menunjukkan bahwa protokol TFTP menunjukkan hasil yang paling cepat. Pada jenis protokol yang

umum dipakai pada jaringan Internet, protokol SCP menunjukkan kecepatan paling tinggi.

REFERENSI

- [1] -----, 2015, *Nginx Documentation*, <http://www.nginx.org/docs>
- [2] Andino Maselena, 2003, *Kamus Istilah Komputer dan Informatika*, IlmuKomputer.Com
- [3] Bradley M., 2015, *Protocol (network)*, <http://compnetworking.about.com/od/networkprotocols/g/protocols.htm>
- [4] Cisco CCNA, 2014, *Introduction To Networks, CCNA 1 Course, Cisco Networking Academy*, <http://www.netacad.com>
- [5] Berners-Lee, T., et al, 1996, *Hypertext Transfer Protocol – HTTP/1.0*, The Internet Society, ISI, <http://tools.ietf.org/html/rfc1945>
- [6] Brian Masney, 2008, *gFTP GNU File Transfer Protocol*, <http://gftp.seul.org>
- [7] Daniel Stenberg, 2015, *Curl Documentation Overview*, <http://curl.haxx.se/docs/>
- [8] Dierks, T., et al, 1999, *The TLS Protocol Version 1.0*, The Internet Society, ISI, <http://tools.ietf.org/html/rfc2246>
- [9] Fielding, R., et al., 1999, *RFC 2616, Hypertext Transfer Protocol – HTTP/1.1*, The Internet Society, W3C/MIT, <http://tools.ietf.org/html/rfc2616>
- [10] Lehtinen, S., et al, 2006, *The Secure Shell (SSH) Protocol Assigned Numbers*, The Internet Society, ISI, <http://tools.ietf.org/html/rfc4250>
- [11] Postel, J., Reynolds, J., 1985, *RFC 959, File Transfer Protocol (FTP)*, The Internet Society, ISI, <http://tools.ietf.org/html/rfc959>
- [12] Rescorla, E., 2000, *RFC 2818, HTTP (Hypertext Transfer Protocol) Over TLS*, The Internet Society, RTFM, Inc., <http://tools.ietf.org/html/rfc2818>
- [13] Sollins, K., 1992, *RFC 1350, The TFTP Protocol (Revision 2)*, The Internet Society, MIT, <http://tools.ietf.org/html/rfc1350>
- [14] The Apache Software Foundation, 2015, *Apache HTTP Server Documentation*, <https://httpd.apache.org/docs/>
- [15] The ProFTPD Project, 2015, *ProFTPD Highly configurable GPL-licensed FTP server software*. <http://www.proftpd.org/docs/>
- [16] Wagito, 2012, *Implementasi VPN PPTP Untuk Integrasi Jaringan*, Puslitbang PPM, STMIK AKAKOM
- [17] Wagito, 2013, *Analisis Kecepatan transfer Data Point-To-Point Tunneling Protocol*, Puslitbang PPM, STMIK AKAKOM
- [18] Ylonen, T., et al, 2006, *The Secure Shell (SSH) Connection Protocol*, The Internet Society, ISI, <http://tools.ietf.org/html/rfc4254>
- [19] Ylonen, T., et al, 2006, *The Secure Shell (SSH) Authentication Protocol*, The Internet Society, ISI, <http://tools.ietf.org/html/rfc4252>
- [20] Ylonen, T., et al, 2006, *The Secure Shell (SSH) Protocol Architecture*, The Internet Society, ISI, <http://tools.ietf.org/html/rfc4251>
- [21] Ylonen, T., et al, 2006, *The Secure Shell (SSH) Transport Layer Protocol*, The Internet Society, ISI, <http://tools.ietf.org/html/rfc4253>