

KONFIGURASI TORRENT SERVER BERBASIS LINUX PADA JARINGAN LOKAL

Henry Aryco^{*)}, Adian Fatchur Rochim, and Yuli Christiyono

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}*email : henryaryco@gmail.com*

Abstrak

Torrent adalah perangkat lunak yang berjalan di protokol bittorrent yang memungkinkan pengguna komputer mendistribusikan data ke pengguna yang lainnya melalui Internet. Protokol BitTorrent dapat digunakan untuk mengurangi dampak server dan jaringan untuk mendistribusikan file besar karena tidak terpusat dalam satu server saja. Perancangan torrent server pada jaringan lokal tidak berbeda dengan torrent server pada jaringan internet seperti Isohunt, Torrenreactor, Demonoid dan torrent server lain. Infrastruktur yang dibutuhkan hanya sebuah server kecil yang berfungsi sebagai tracker untuk menghubungkan antar pengguna yang akan melakukan aktivitas berbagi konten serta antarmuka berbasis web agar pengguna lebih mudah mengunduh file Torrent yang dibagikan pengguna lain. Pengujian dalam penelitian ini pengguna berhasil mengunduh konten melalui torrent pada jaringan lokal. Penggunaan server sebagai tracker tidak terlalu besar karena aplikasi Opentracker memiliki ukuran yang kecil dan pada aktivitas berbagi konten, server tidak bekerja sendirian karena apabila ada pengguna lain yang memiliki konten yang sama akan membantu proses seeding. Dalam pengujian pada jaringan lokal pada MRTG (Multi Router Traffic Grapher) untuk proses seeding, kecepatan maksimal 2,825 MBps sedangkan kecepatan pada saat server mengunduh konten dari pengguna kecepatan maksimal yang didapat 1,525Mbps. Kecepatan seeding pada server maupun pada pengguna yang dihasilkan pada saat pengamatan biasa melalui μ Torrent dan Transmission 4,7 MBps hingga 5,74 MBps sedangkan kecepatan mengunduh konten dari pengguna berkisar antara 400 KBps hingga 3,2MBps.

Kata Kunci : File sharing, Peer-to-Peer, Torrent, Transmission, Tracker

Abstract

Torrent is software that runs on bittorrent protocol that allows computer users to distribute data to other users via the Internet. BitTorrent protocol can be used to reduce the impact of server and network to distribute large files because it is not concentrated in one server only. Designing torrent server on the local network is not different from the torrent server on the Internet such as isoHunt, TorrentReactor, Demonoid and other torrent server. Infrastructure needed only a small server that serves as a tracker to interconnect users who will perform the activity and content sharing web-based interface to let users more easily download Torrent files shared by other users. Users successfully download contents via torrent on a local network in this research. Using a server as an tracker not require a large capacity, because the Opentracker has a small size. While the content sharing activity, the server does not work alone because if there is a user who has the same content will help the seeding process. In testing on a local network, which looks at MRTG (Multi Router Traffic Grapher) for the seeding process, a maximum speed is 2.825 Mbps, while the speed of the server when users download contents from a maximum speed of 1.525 Mbps obtained. Seeding speed on the server and the user generated during normal observation through the μ Torrent and Transmission 5.74 MBps to 4.7 MBps speed while downloading contents from the user range from 400 KBps to 3.2 MBps.

Keywords: File sharing, Peer-to-Peer, Torrent, Transmission, Tracker

1. Pendahuluan

Kemajuan jaringan Internet saat ini memungkinkan adanya penyedia konten yang bebas berbagi pakai, salah satunya melalui torrent. Torrent mendistribusikan data terlebih dalam jumlah yang besar secara luas tanpa harus

menanggung semua beban kinerja perangkat keras serta jaringan karena tidak membutuhkan infrastruktur seperti server dan jalur data yang besar. Pendistribusian konten melalui protokol Torrent, pengguna tidak menjadi server secara penuh karena pada suatu saat komputer pengguna lain juga menjadi server untuk pengguna yang

membutuhkan konten yang sama. Oleh karena itu torrent banyak disukai oleh pengguna internet karena kemudahan dalam berbagi.

Saat ini banyak penyedia torrent yang membagi konten secara bebas seperti Isohunt, Torrentreactor, Demonoid dan masih banyak lagi yang bersifat publik. Penyedia torrent menyediakan file torrent serta menyediakan indeks tentang konten apa saja yang nantinya akan diunduh oleh pengguna melalui halaman website. Penyedia Torrent yang beredar saat ini berjalan pada jaringan internet karena membutuhkan pengguna yang cukup banyak agar bisa melakukan aktivitas berbagi konten. Karena Torrent membutuhkan pengguna yang jumlahnya cukup banyak, torrent dapat diaplikasikan pada jaringan lokal pada suatu organisasi yang memiliki infrastruktur serta pengguna internet yang besar.

Aktivitas pengunduhan konten pada lingkungan Universitas Diponegoro khususnya Jurusan Teknik Elektro dari internet yang cukup banyak dan bisa tidak sebanding dengan bandwidth yang ada. Dalam penelitian ini, untuk memenuhi kebutuhan mengunduh konten dirancang penyedia Torrent pada jaringan lokal sehingga pengguna tidak selalu mengunduh melalui Internet. Oleh karena itu, pengguna internet harus mengunggah kembali pada torrent server di jaringan lokal dalam bentuk file torrent untuk dibagikan ke pengguna lain.

2. Metode

2.1 Perancangan Sistem

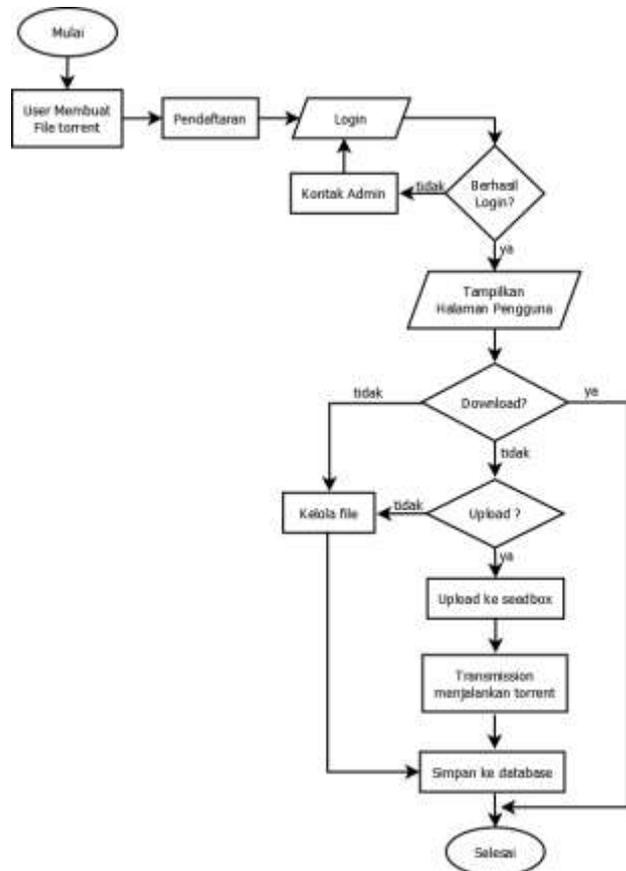
Torrent Server merupakan server yang menyediakan berkas Torrent yang dibutuhkan oleh pengguna yang hendak berbagi konten. Perancangan sistem meliputi :

- Apache dan PHP sebagai *web server* menjalankan aplikasi antarmuka pengguna serta menjalankan aplikasi *tracker* dan antarmuka pengguna.
- Transmission Daemon sebagai klien pada *server* yang bertugas untuk mengunduh konten asli dari pengguna yang mengunggah *file* Torrent ke *server* selain itu untuk melakukan kegiatan *seeding*. Transmission Daemon akan secara otomatis mengunduh konten asli dari pengguna pada saat pengguna mengunggah *file* Torrent ke *seedbox*. Pada Transmission telah di konfigurasi agar secara otomatis mengunduh konten asli pada direktori filetorrent.
- MySQL server berfungsi sebagai pengolahan basis data pada antarmuka pengguna dan Opentracker.
- Opentracker berfungsi untuk menemukan pengguna torrent satu sama lain agar bisa saling berbagi file satu sama lain melalui jaringan bittorrent. Opentracker berjalan melalui protokol HTTP (Hypertext Transfer Protocol) karena berbasis PHP.

2.2 Perancangan Antarmuka Pengguna

Cara kerja sistem ini secara umum adalah sebagai berikut:

- Pengguna membuat *file* Torrent pada komputer klien
- Pengguna melakukan pendaftaran di halaman utama agar bisa masuk ke antarmuka pengguna.
- Pengguna mengunggah *file* Torrent ke server untuk dibagikan ke pengguna lain
- Pengguna dapat mengunggah kembali *file* Torrent ke server dengan tujuan agar server dapat melakukan *seeding* ketika pemilik *konten asli* tidak melakukan aktifitas *seeding*.
- Transmission akan mengunduh *file* dari pengguna yang memiliki *konten asli* sesaat setelah pengguna mengunggah *file* Torrent ke server melalui Transmission Daemon.



Gambar 1. Diagram alir antarmuka pengguna

Dalam antarmuka web ini memisahkan proses yang utama menjadi 2 bagian, yaitu:

1. Admin

Admin secara umum bertugas sebagai berikut :

- Mengawasi semua file memiliki konten kurang baik pada sistem tanpa sepengetahuan pengguna maupun sepengetahuan pengguna.
- Bertugas untuk memberikan kembali dan mengganti kata sandi pada pengguna.
- Memonitor aktifitas *seeding* pada server.

4. Menghapus file asli yang diunggah oleh pengguna pada server.
5. Mengelola pengguna seperti mengangkat pengguna biasa menjadi admin, menghapus dan mengubah data pengguna.

2. Pengguna

Pengguna di sistem diharuskan login terlebih dahulu untuk dapat masuk ke menu yang telah ditentukan, sedangkan untuk guest hanya dapat melihat menu utama, tiap – tiap user memiliki pilihan menu khusus tersendiri seperti melihat profil, mengelola *file* Torrent yang diunggah, mengunduh semua *file* Torrent, mengunggah *file* Torrent.

2.3 Pemantauan Jaringan

2.3.1 Perancangan SNMP

SNMP adalah standar manajemen jaringan pada TCP/IP. Gagasan di balik SNMP adalah bagaimana supaya informasi yang dibutuhkan untuk manajemen jaringan bisa dikirim menggunakan TCP/IP. SNMP berfungsi untuk memantau pemakaian bandwidth pada server. Pada perancangan SNMP, konfigurasi dilakukan pada berkas `/etc/snmp/snmpd.conf` yang diisi dengan

```
rocommunity public localhost
syslocation "Torrent Server"
```

konfigurasi selanjutnya mengubah konfigurasi standar dari `snmpd` dengan perintah

```
# nano /etc/snmp/snmpd.conf
```

Beberapa konfigurasi yang diubah adalah

```
SNMPDRUN=yes
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I
-smux -p /var/run/snmpd.pid -c
/etc/snmp/snmpd.conf'
```

SNMP bisa dikatakan berjalan apabila berhasil menampilkan data dari agen `SNMPD` melalui `localhost`.

2.3.2 Perancangan MRTG

MRTG (*Multi Router Traffic Grapher*) berfungsi untuk menampilkan dalam bentuk grafik aktivitas dari SNMP.

Perintah yang digunakan untuk memasang MRTG adalah sebagai berikut

```
# apt-get install mrtg
# sudo mkdir /var/www/mrtg
# chmod 777 /var/www/mrtg
```

Berkas konfigurasi dari MRTG dibuat dengan perintah

```
# cfgmaker --global 'WorkDir:/var/www/mrtg' --
ifref=name --ifdesc=eth --global 'Options[_]:
bits' --output /etc/mrtg.cfg public@localhost
```

Perintah `cfgmaker` menghasilkan `mrtg.cfg` yang merupakan berkas konfigurasi dari MRTG. Berkas konfigurasi `mrtg.cfg` digunakan untuk membuat berkas `index.html` yang nantinya akan menampilkan grafik pemakaian jaringan. Berkas `index.html` dapat dibuat dengan perintah

```
# indexmaker /etc/mrtg.cfg >
/var/www/mrtg/index.html
# env LANG=C /usr/bin/mrtg /etc/mrtg.cfg
```

Grafik dari MRTG dapat dilihat pada `10.31.17.217/mrtg` yang akan menampilkan pemakaian *bandwidth* pada *server*.

3. Hasil dan Analisa

Sistem yang akan diterapkan pada *Torrent server* adalah *Opentracker* sebagai *tracker* dan *Transmission Daemon* sebagai klien pada *server* untuk melayani *seeding* untuk pengguna, halaman antarmuka pengguna sebagai sarana pengguna untuk berbagi *file* Torrent. SNMP serta MRTG digunakan untuk memantau pemakaian jaringan yang dipakai oleh *server*.

3.1 Hasil Pengujian Transmission Daemon

Pengujian *Transmission Daemon* dilakukan ketika pengguna mengunggah *file* Torrent ke halaman antarmuka pengguna. Pada saat pengguna mengunggah ke *seedbox*, maka *Transmission Daemon* akan mengunduh secara otomatis *file* Torrent pada direktori `filetorrent.direktori` `filetorrent` adalah direktori yang digunakan untuk menampung *file* Torrent.

Pengujian ini membandingkan antara kecepatan mengunduh dari *file* asli melalui Torrent dari *server* ke klien maupun dari klien ke *server*. *transmission* memiliki kecepatan unduh 3,25 MB/s. Berdasarkan pengamatan, kecepatan unduh tidak stabil diantara 400 KB/s hingga 3.25MB/s. Proses mengunduh melalui torrent tidak langsung dimulai karena torrent harus melakukan permintaan ke *tracker* untuk mencari *piece file* ke semua pengguna yang memiliki *piece file* lain maupun *file* asli untuk mengunduh *file*.

Pengujian selanjutnya adalah server melakukan kegiatan *seeding* atau mengunggah *file* ke komputer klien yang sedang mengunduh *file* asli. Berikut ini adalah tampilan ketika pengujian kecepatan unggah pada *server* menggunakan *Transmission Daemon* dan kecepatan unduh pada klien dengan menggunakan *utorrent*.

Pengujian *seeding* dari *server* ke pengguna hampir sama, kecepatan naik turun dengan perkiraan 5,74 MB/s pada *server* dan 4,7 MB/s pada pengguna. Prosesnya sama dengan pengujian sebelumnya. Proses mengunduh maupun melalui torrent tidak langsung dimulai karena

torrent harus melakukan permintaan ke *tracker* untuk mencari *piece file* ke *seeder* yang dibutuhkan untuk mengunduh *file*. Hal ini berbeda dengan mengunduh langsung melalui HTTP yang bisa langsung memulai proses mengunduh konten asli dari pengguna.

3.2 Hasil Pengujian Opentracker

Pengujian diawali dengan mencantumkan *tracker* <http://10.31.17.217/tracker/announce.php> pada saat membuat file Torrent kemudian memberikan tanda centang pada opsi *start seeding* agar *server* bisa mengunduh konten asli dari pengguna yang memiliki konten asli.

Opentracker bisa dikatakan bekerja dengan baik apabila antar pengguna satu dengan yang lain dapat saling berbagi *file* menggunakan Torrent. Setelah *file* Torrent sudah jadi kemudian diunggah ke antarmuka *web* untuk pengguna dan ke *seedbox*. Apabila *seedbox* berhasil melakukan kegiatan *seeding*, dapat disimpulkan bahwa *tracker* berhasil berjalan pada jaringan lokal.

Intouchables (2011)

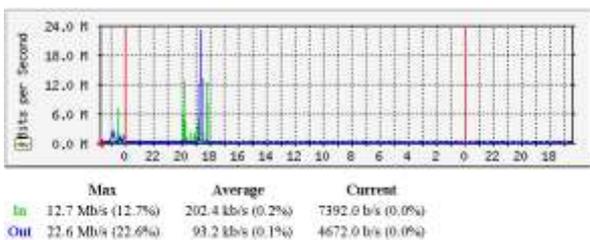
<http://10.31.17.217:80>
 Last Announce: Today 01:05:26 AM (got 2 peers)
 Next announce in 4 minutes
 Last Scrape: Today 12:51:20 AM
 Seeders: 2
 Leechers: 0
 Downloads: 4

Gambar 2. Tampilan Tracker pada Transmission

Pada gambar 2 terlihat bahwa Transmission Daemon berhasil mengunduh *file* “Intouchables (2011)” dengan baik ke *seedbox* untuk tujuan *seeding*. Hal ini dapat terlihat bahwa *tracker* dapat melihat jumlah *seeder* tak berapa kali konten asli diunduh.

3.2 Hasil Pengujian MRTG

Pengujian *Multi Router Traffic Grapher* yang menunjukkan berapa *bandwidth* yang dipakai ketika Torrent *server* bekerja. Berikut adalah gambar 3 merupakan grafik pada MRTG ketika *server* melayani 3 buah *peers* dan 2 *seeder* yang sedang mengunggah berkas asli ke *server*.



Gambar 3. Grafik MRTG

Grafik berwarna hijau menandakan *incoming traffic* atau *server* sedang mengunduh berkas asli dari pengguna yang

nantinya *server* akan melakukan *seeding*. Pada saat pengujian, transfer data maksimal sebesar 12,7 Mbps, transfer data rata-rata 202,4 kbps dan pada saat data diambil. Apabila dikonversi menjadi Byte maka kecepatan maksimal menjadi 1,525 MBps. Hasil ini sedikit berbeda jauh karena kondisi jaringan lokal yang sedang menurun.

Grafik berwarna biru menandakan *outcoming traffic* atau bisa dikatakan *server* sedang melakukan kegiatan *seeding* atau mengunggah berkas asli ke komputer pengguna. Pada saat pengujian, transfer data maksimal sebesar 22,6 Mbps, transfer data rata-rata 93,2 kbps dan pada saat data diambil. Apabila dikonversi menjadi Byte maka kecepatan maksimal menjadi 2,825 MBps, hal ini berbeda jauh dengan pengujian mengunduh sebenarnya karena *server* tidak bekerja sendirian dalam melakukan kegiatan *seeding*.

3.8 Perbandingan Mengunduh Melalui Torrent dengan HTTP

Parameter perbandingan yang dipakai untuk menguji kecepatan mengunduh dengan menggunakan data yang sebelumnya telah diambil melalui MRTG dan hasil pengamatan mengunduh melalui Internet Download manager melalui protokol HTTP.

Kecepatan yang dihasilkan pada saat data diambil sekitar 1,743 MBps menurut pengamatan pada saat jaringan sedang tidak banyak dipakai sekitar pukul 2.00 malam. Hasil kecepatan mengunduh tidak jauh berbeda dengan mengunduh menggunakan protokol Torrent ketika pengujian *seeding* dari *server* ke pengguna, kecepatan yang dapat dilihat melalui MRTG 2,825 MBps. Maka dari itu, dapat disimpulkan bahwa berbagi *file* melalui Torrent tidak selalu lebih lambat dibandingkan melalui HTTP.

4. Kesimpulan

Dari hasil implementasi dan pengujian dapat disimpulkan bahwa:

1. LDAP pada penelitian ini tidak berhasil berjalan karena adanya modul apache untuk LDAP mengalami kerusakan sehingga tidak dapat mengautentifikasi ke LDAP.
2. Modul Apache untuk LDAP mengalami kegagalan sehingga menyebabkan kinerja *server* menurun, untuk itu dilakukan konfigurasi ulang terhadap *web server*.
3. Transmission Daemon berjalan stabil pada *server* ketika melakukan proses *seeding* apabila tidak ada pengguna lain yang melakukan *seeding*, sehingga aktifitas berbagi konten dapat berjalan.
4. Kecepatan *seeding* pada *server* maupun pada pengguna yang dihasilkan pada saat pengamatan biasa melalui *µTorrent* dan Transmission 4,7 MBps hingga 5,74 MBps sedangkan kecepatan mengunduh konten dari pengguna berkisar antara 400 KBps hingga 3,2MBps.

5. Kecepatan unggah *file* asli ke *server* maupun ke pengguna yang terlihat pada MRTG ketika mengunggah atau *seeding* 2,825 MBps, sedangkan kecepatan mengunduh *file* asli ke *server* berada pada 1,525 MBps.
6. Kecepatan yang dihasilkan ketika mengunduh langsung dari *server* pada saat data diambil 1.743 MBps, sedangkan pengunduhan dari *server* melalui Torrent 2.825 MBps.
7. Berdasarkan data pada saat pengujian, berbagi *file* melalui Torrent relatif lebih cepat jika dibandingkan melalui HTTP.

Referensi

- [1]. Chandra Ian, K., 2010. *Teknik Berbagi Objek lewat Jaringan P2P*. Jakarta : Penerbit Elex Media Komputindo.
- [2]. Cohen Bram, *The BitTorrent Protocol Specification*, 25 Juni 2009, <http://bittorrent.org/beps/bep_0003.html> [diakses 30 Oktober 2012]
- [3]. Emka Luthfi, *Mengenal & Membuat Encoding Decoding BitTorrent Bencode dengan PHP*, Club Teknologi Informasi dan Komunikasi, 9 Juni 2010, <<http://dotexe.wordpress.com/2010/06/09/mengenal-membuat-encoding-decoding-bittorrent-bencode-dengan-php/>> [diakses 11 Oktober 2012]
- [4]. Hayder. H., "Object-oriented Programming with PHP5", Desember 2007
- [5]. Huda Miftakhul, Bunafit Komputer, 2010, *Membuat Aplikasi Database dengan Java, MySQL, dan NetBeans*. Jakarta : Penerbit Elex Media Komputindo.
- [6]. More Hawes, *The BitTorrent Protocol*, <<http://www.morehawes.co.uk/the-bittorrent-protocol>> [diakses 31 Oktober 2012]
- [7]. Nugroho, B., *Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL*, Cetakan Pertama, 2004.
- [8]. Lerdorf. R., P. MacIntyre. , and K. Tatroe. "Programming PHP, 2nd Editon". O'Reilly. April 2006.
- [9]. Seedbox. <http://id.wikipedia.org/wiki/Seedbox> (diakses tanggal 31 Oktober 2012 pukul 01.42 WIB).
- [10]. Richards, R., "Pro PHP XML and Web Services". 2006
- [11]. Sklar, D., "Learning PHP 5". O'Reilly. June 2004.