ISSN 2085-4579

Implementasi *Distributed File Server* pada *Apache Web Server*

Novita Belinda Wunarso, Andrias Rusli, Michelle Angelica, Arabella Margaret Salim

Program Studi Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia

Diterima 5 Desember 2013

Disahkan 23 Desember 2013

***Abstract*—In this research, a *Distributed File Server* (DFS) is developed to manage the need of a large storage space in the server, especially when the multimedia files are saved permanently in the web server. In the development, *Apache Web Server* is used with 1 computer as the main server, 2 computers as file servers, and 1 computer as the client who sends request. The result from the implementation of the DFS is the usage of main server's storage space can be reduced by 99,77% from the full usage condition, causing an optimization in the web server. Another parameter is also being tested by the implementation of the DFS, which is the index.html page's average access time. When only one server is being used, the average access time is 2,327 second. Whereas, when three servers are being used, the average access time is 5,577 second.**

Index Terms*—Apache, distributed file server, web server optimization, average access time

I. PENDAHULUAN

Dewasa ini, perkembangan teknologi semakin pesat, terutama teknologi internet. Untuk bertukar informasi ataupun data, seseorang tidak harus melakukannya secara manual lagi seperti berkirim surat atau menggunakan *usb drive* untuk memindahkan file. Dengan adanya teknologi internet, masyarakat dapat melakukan *sharing* file yang mereka miliki melalui halaman web, di mana file tersebut akan diunggah ke *web server*. Orang-orang yang membutuhkan file tersebut kemudian dapat mengunduhnya secara langung.

Namun, mengunggah file ke *web server* menyebabkan penggunaan *storage space* yang cukup banyak, terutama jika jumlah file yang diunggah bertambah dan penyimpanannya bersifat permanen. Dengan lalu lintas data yang sangat padat di dalam *world-wide web* (WWW), merupakan hal yang penting untuk mencapai performa yang dapat diukur dari web server[1]. Keseluruhan performa dan pemanfaatan sumber daya dapat ditingkatkan dengan menyebarkan permintaan dokumen di antara kumpulan web server [1].

Oleh karena itu, dikembangkan sebuah *Distributed File Server* (DFS) sebagai salah satu cara untuk mengurangi penggunaan *storage space* web server. Server yang terdistribusi tersusun atas banyak file dalam berbagai format yang dimaksudkan untuk dibagi melalui jaringan [2]. Setiap server tersusun atas folder terdedikasi yang dibagi di antara semua server. Setiap server dilengkapi dengan izin untuk hanya membaca file dari server lain [2].

DFS menggunakan arsitektur yang mirip dengan P2P atau Peer to Peer, dimana informasi didistribusikan di antara *member nodes* , bukannya terkonsentrasi pada satu server [3]. Arsitektur P2P dapat membantu mengurangi beban storage dan memungkinkan pembagian beban dengan menggunakan infrastruktur yang ada dan menggabungkan sumber daya dari tempat yang berbeda [3].

Dengan penerapan DFS, penggunaan *storage* untuk penyimpanan file yang pada awalnya dibebankan seluruhnya hanya pada *single* server akan dibagi ke beberapa server lainnya yang kemudian akan berfungsi sebagai server penyimpan dan penyedia file yang dibutuhkan.

Secara keseluruhan, bagian kedua dalam

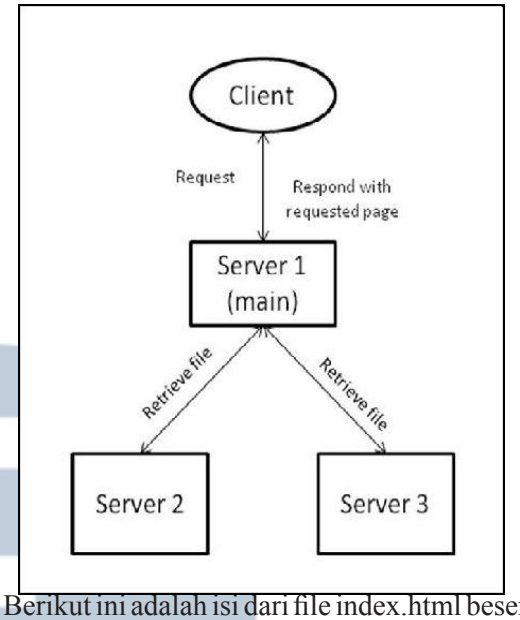
paper ini akan menjelaskan mengenai metode yang digunakan pada penelitian, bagian ketiga akan memaparkan hasil yang diperoleh beserta pembahasannya, dan bagian keempat berisi kesimpulan mengenai penelitian serta saran.

II. METODE

Pada implementasi *Distributed File Server* (DFS) ini, terdapat satu server utama yang terhubung melalui jaringan *wireless* dengan dua server sebagai penyimpan dan penyedia file serta satu klien pengirim *request.*

Server yang digunakan adalah *Apache Web Server* dengan direktori *root* /var/www yang berjalan pada OS Linux Mint sebagai lokasi penyimpanan file HTML, PHP *script* maupun file multimedia yang akan ditampilkan pada halaman web. Sebelumnya /var/www pada masing-masing server telah diatur *permission*nya menjadi 777 agar dapat dibaca, dieksekusi dan ditulis ke dalamnya oleh server lain pada saat eksekusi *script*.

Berdasarkan arsitektur sistem DFS pada Gambar 1, setiap kali klien melakukan *request* atas halaman web yang di-*host* pada server 1 (main) melalui ipaddress-server-1/index.html, maka, PHP *script* yang berada di server 1 akan dieksekusi. Terdapat dua PHP *script* yang digunakan yaitu image1.php dan image2.php, dimana ketika dieksekusi, server 1 akan mengambil file gambar dari server 2 dan server 3, kemudian merespon klien dengan menampilkan halaman web yang diminta.



Berikut ini adalah isi dari file index.html beserta image1.php dan image2.php yang digunakan.

index.html :

```
<html>
    <head>
    <title>Welcome!</title>
    </head>
    <body>
        <center>Hello!
        Welcome to this
        website!<br/> <img
        src="/image1.
        php"><br/> <img
        src="/image2.php">
        </center>
    </body>
</html>
```

Pemanggilan PHP *script* dilakukan dengan meletakkan /image1.php dan /image2.php sebagai *source* dari tag <img>.

image1.php :

```
<?php
    $command="wget ip-
```

```
        serverkedua/image1.
        jpg

        /var/www"; shell_
        exec($command);
        $image="image1.
        jpg";

        header("Content-
        type: image/jpg");
        readfile($image);
        unlink($image);
        exit();

?>
```

image2.php :

```
<?php

        $command="wget ip-
        serverketiga/image2.
        jpg

        /var/www"; shell_
        exec($command);
        $image="image2.
        jpg";
        header("Content-
        type: image/jpg");
        readfile($image);
        unlink($image);
        exit();

?>
```

Variabel $command pada masing-masing PHP *script* image1.php dan image2.php menampung perintah wget[4] untuk mengambil file gambar image1.jpg dan image2.jpg dari /var/www server penyimpan dan penyedia file yaitu server 2 dan server 3 lalu menyimpannya pada /var/www server 1. Fungsi shell_exec() kemudian akan mengeksekusi perintah di dalam variabel $command di *shell* server 1[5]. Selanjutnya gambar yang telah diambil akan disimpan di dalam variabel $image. Fungsi readfile() bertugas membaca file gambar pada variabel $image[6]. Fungsi header()yang berisi *content type* dari file berfungsi untuk memberi tahu pada browser bahwa gambar yang akan ditampilkan adalah bertipe JPG[7]. Setelah gambar berhasil ditampilkan pada halaman web, fungsi unlink() yang berfungsi untuk menghapus file[8] akan menghapus file gambar secara permanen dari /var/www server 1. Fungsi exit() pada bagian akhir PHP *script* akan mengakhiri script yang berjalan[9].

III. HASIL DAN PEMBAHASAN

Pada kondisi awal, seperti dapat dilihat pada Tabel 1, besarnya *storage space* server utama yang digunakan adalah sebesar 0,480 kilobyte, yang terdiri dari file index.html berukuran 0,164 kilobyte, serta image1.php dan image2.php yang masing-masing berukuran 0,158 kilobyte. Sesaat setelah server utama berhasil mengambil file gambar sebesar 80,9 kilobyte dari server 2 dan sebesar 123,4 kilobyte dari server 3 kemudian menyimpannya pada /var/www server 1, *storage space* server yang digunakan bertambah menjadi 204,78 kilobyte.

Penggunaan *space* sebesar 204,78 kilobyte tersebut adalah dalam kondisi apabila di dalam PHP *script* tidak terdapat fungsi unlink() atau file gambar disimpan permanen seluruhnya oleh server 1. Pada kondisi tersebut, server 1 atau *main server* menampung index.html, image1.php, image2.php, image1.jpg dan image2.jpg.

Tabel 1. Penggunaan Storage Space Main Server

Kondisi Awal		
Contents in Main Server	**Quantity**	**Storage Space Usage**
index.html	1	0,164 kB
image1.php	1	0,158 kB
image2.php	1	0,158 kB
	Total Usage	0,480 kB

Kondisi jika file gambar disimpan permanen atau PHP script tanpa fungsi unlink()		
index.html	1	0,164 kB
image1.php	1	0,158 kB
image2.php	1	0,158 kB
image1.jpg	1	80,9 kB
image2.jpg	1	123,4 kB
Total Usage		204,78 kB
PHP script dengan fungsi unlink()		
index.html	1	0,164 kB
image1.php	1	0,158 kB
image2.php	1	0,158 kB
Total Usage		0,480 kB

kemudian dihapus dari folder /var/www server 1. Pada kondisi akhir, *storage space* server yang digunakan adalah sebesar 0,480 kilobyte atau sama dengan kondisi awal *storage space* server.

Pada Gambar 2, dengan asumsi bahwa penggunaan *storage* secara penuh jika gambar disimpan permanen bersama file HTML dan PHP *script* sebesar 204,78 kilobyte adalah 100% dan kondisi akhir sebesar 0,480 kilobyte setelah fungsi unlink() dieksekusi adalah 0,23%, terjadi pengurangan penggunaan *storage space* di *main server* sebesar 99,77%.

$$\begin{aligned} \textit{Full usage} &= 204{,}78\ kB = 100\ \% \\ \textit{Kondisi akhir} &= 0{,}480\ kB \\ &= \frac{0{,}480}{204{,}78} \times 100\% \\ &= 0{,}23023.. \approx 0{,}23\ \% \\ \textit{Pengurangan penggunaan} &= 100\% - 0{,}23\% = 99{,}77\% \end{aligned}$$

Gambar 2. Persentase Pengurangan Penggunaan Storage Space di Main Server

Parameter lain yang juga diukur di dalam percobaan ini adalah waktu akses rata-rata halaman index.html yang ditunjukkan pada Tabel 2.

Tabel 2. Waktu Akses Rata-Rata Halaman Index.html dalam Second (s)

Jumlah Server	Testing ke-1	Testing ke-2	Testing ke-3	Rata-Rata
1	2,61 s	2,04 s	2,33 s	2,327 s
3	5,23 s	5,07 s	6,43 s	5,577 s

Sedangkan, jika menggunakan tiga server atau file gambar harus diambil terlebih dahulu dari server 2 dan server 3, waktu akses rata-rata index.html yang dibutuhkan adalah 5,577 s.

IV. SIMPULAN

Hasil percobaan ini menunjukkan bahwa penerapan DFS dapat mengurangi penggunaan storage space server sebesar 99,77% dari penggunaan jika file multimedia disimpan secara permanen di server. Dengan demikian, optimisasi web server dapat dilakukan melalui implementasi *Distributed File Server*. Selain itu, parameter lain yang diukur yaitu waktu akses rata-rata halaman index.html berdasarkan jumlah server yang dipakai menunjukkan bahwa penggunaan satu server membutuhkan waktu akses yang lebih cepat, sebesar 2,327 s, jika dibandingkan penggunaan tiga server yang membutuhkan waktu akses 5,577 s.

Kekurangan dari sistem ini adalah dari segi kemampuan akses file, waktu yang dibutuhkan untuk me-*load* halaman web dan berhasil ditampilkannya file pada halaman web. Akses file bersifat terbatas karena server utama tidak dapat mengambil file dari server lain apabila file tersebut tidak tersimpan pada direktori /var/www/ server yang bersangkutan. Waktu yang diperlukan untuk me-*load* halaman web menjadi lebih lama karena konten atau file multimedia yang ditampilkan di halaman web harus diambil dari beberapa server yang berbeda, terlebih lagi jika ada banyak klien yang melakukan *request* halaman web secara bersamaan. Apabila file multimedia yang disimpan di server lainnya diubah nama atau tipenya, maka dapat menyebabkan gagal

ditampilkannya file tersebut pada halaman web karena tidak sesuai dengan nama dan tipe file yang dituliskan di dalam PHP *script*.

Kelebihan dari sistem ini adalah dari segi penggunaan *storage space* web server dan jumlah file yang dapat disimpan. Dengan pendistribusian penyimpanan file ke beberapa server sekaligus, penggunaan *storage space* server utama dapat dikurangi dan jumlah file yang dapat disimpan akan bertambah sesuai dengan banyaknya server yang digunakan di dalam sistem.

Hambatan yang dihadapi di dalam penelitian mengenai *Distributed File Server* (DFS) ini adalah percobaan hanya dilakukan dua kali. Selain itu, file multimedia yang digunakan untuk percobaan hanya berupa file gambar berukuran kecil yang berasal dari dua server. Untuk jenis file lainnya dan jumlah server yang lebih banyak belum dilakukan uji coba. Mengenai waktu yang dibutuhkan untuk kecepatan akses, hanya dilakukan pencatatan sebanyak tiga kali per jumlah server yang digunakan.

Oleh karena itu, pada penelitian selanjutnya, sebaiknya dilakukan uji coba untuk lebih banyak jenis file multimedia dengan ukuran file yang lebih besar, misalnya video atau mp3, pada jumlah server terdistribusi yang lebih banyak untuk memastikan bahwa metode yang dipaparkan dapat berfungsi dengan baik. Selain itu, pencatatan waktu untuk mengakses halaman web perlu dilakukan lebih dari tiga kali agar lebih akurat.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Pak Yustinus Eko Soelistio atas bimbingannya dalam penelitian dan penyelesaian paper ini, terutama pada saat penentuan topik penelitian dan pelaksanaan percobaan. Penulis juga berterima kasih kepada orang tua penulis dan rekan-rekan dari jurusan Sistem Informasi Universitas Multimedia Nusantara, atas dukungan, motivasi dan bantuannya dalam proses penyelesaian paper ini.

REFERENSI

[1] Q. Li, B. Moon, "Distributed Cooperative Apache Web Server‖ , *WWW10*, hal. 555, 2001.

[2] A. Singh, A.K. Sharma, A. Kumar, B.R. Ambedkar, ―Multi-Client Multi-Instance Secured Distributed File Server,‖ *IJCA,* vol. 19, no.9, hal. 3, 2011.

[3] R. Hasan, Z. Anwar, W. Yurick, L. Brumbaugh, R. Campbell, ―A Survey of Peer-to-Peer Storage Techniques for Distributed File Systems‖, *ITCC*, vol. 2, hal. 205, 2005.

[4] GNU Wget 1.13.4 Manual. Available : http://www.gnu.org/software/wget/manual/wget.html

[5] PHP : shell_exec - Manual. Available: http://www.php.net/manual/en/function.shell-exec.php

[6] PHP : readfile – Manual. Available: http://www.php.net/manual/en/function.readfile.php

[7] PHP : header – Manual. Available: http://www.php.net/manual/en/function.header.php

[8] PHP : unlink – Manual. Available: http://www.php.net/manual/en/function.unlink.php

[9] PHP : exit – Manual. Available: www.php.net/manual/en/function.exit.php