

**ANALISIS DAN IMPLEMENTASI *REVERSE PROXY* SEBAGAI MEDIA
KOMUNIKASI *CLIENT SERVER* MENGGUNAKAN *APACHE*
(Studi Kasus Pada Lab. Jaringan Komputer IST AKPRIND YOGYAKARTA)**

Ariyadi Dwi Utomo¹, Rr.Yuliana Rachmawati², Catur Iswahyudi³

Program Studi Teknik Informatika, Fakultas Teknologi Industri

Institut Sains & Teknologi AKPRIND Yogyakarta

Email :idaydwimo@gmail.com¹, yuliana@akprind.ac.id², catur@akprind.ac.id³.

ABSTRACT

Internet in this postmodernist era was not something more thing a myth of development of civilization for humans and the world even sounds have been very familiar in the ear, both in a village and in a city very close in daily activity. The growth of internet technology of course was balanced with network technology computer which shelter internet acces wherever they for every user both to loked for an information and interaction needed.

Once of the balancing method we can used reverse proxy, a proxy which has an apposite purpose with forward proxy intermediatry users on the internet to access the web server that existed on LAN, so the users on the internet like direct access the web servers that they meant actually user on internet was access the web server on LAN through the reverse proxy.

The conclusion that could be taken from this research is based on comparison test result analysis of web server access which was used reverse proxy media there was an increase of time spent by clients towards the access on the web server with a number of 60% and reducing the burden of web server performance in serving client demand used reverse proxy media of 54%, so in the optimization and stability of web server access could be declared to have succeed as desired.

Keywords: reverse proxy, web servers apache, proxy server

INTISARI

Internet di zaman postmodernis seperti saat ini bukanlah sesuatu hal lagi yang menjadi mitos perkembangan peradaban zaman pada manusia juga dunia bahkan kedengarannya sudah sangat tak asing di telinga, baik di desa maupun di kota internet sangatlah dekat dalam kehidupan sehari-hari. Perkembangan teknologi internet ini tentu diimbangi dengan teknologi jaringan komputer yang menaungi akses internet dimanapun ia berada. Internet adalah barang komoditas bagi setiap penggunanya baik untuk mencari informasi maupun sampai keperluan berinteraksi.

Salah satu metode penyeimbang beban *web* dapat menggunakan *reverse proxy*, yaitu *proxy* yang memiliki maksud yang berkebalikan dengan *forward proxy* yaitu menjadi perantara *user-user* di internet terhadap akses ke *web-web server* yang berada pada LAN, sehingga seolah-olah *user* di internet mengakses langsung *web server* yang dimaksud padahal sesungguhnya user di internet mengakses *web-web server* yang terdapat di LAN melalui *reverse proxy* tersebut.

Kesimpulan yang diambil dari penelitian ini adalah berdasarkan perbandingan analisis hasil pengujian akses *web server* menggunakan media *reverse proxy* terjadi peningkatan waktu yang dihabiskan oleh klien terhadap akses pada *web server* dengan sejumlah 60% dan pengurangan beban kinerja *web server* dalam melayani permintaan klien menggunakan media *reverse proxy* sejumlah 54%, sehingga dalam pengoptimalisasi dan kestabilan akses *web server* dapat dinyatakan telah berhasil sesuai yang diinginkan.

Kata kunci: *Reverse proxy, web server apache, proxy server*

PENDAHULUAN

Internet di zaman postmodernis seperti saat ini bukanlah sesuatu hal lagi yang menjadi mitos perkembangan peradaban zaman pada manusia juga dunia, dan sangatlah dekat dalam kehidupan sehari-hari bahkan telah termasuk sebagai kebutuhan yang primer sebagaimana sandang, pangan dan papan. Fenomena peningkatan jumlah pengguna ini harus disiasati karena klien belum sepenuhnya merasakan secara optimal akses *web* yang ada dikarenakan beban permintaan yang terlalu padat dan terkadang pula *web server* mengalami *server* mati atau kegagalan *server* jika terlalu banyak permintaan sehingga *web server* tidak dapat menanganinya.

Salah satu metode penyeimbang beban *web* dapat menggunakan *reverse proxy*, yaitu *proxy* yang memiliki maksud yang berkebalikan dengan *forward proxy* yaitu menjadi perantara *user-user* di internet terhadap akses ke *web-web server* yang berada pada LAN, sehingga seolah-olah *user* di internet mengakses langsung *web server* yang dimaksud padahal sesungguhnya *user* di internet mengakses *web-web server* yang terdapat di LAN melalui *reverse proxy* tersebut. Tentunya komputer yang bertindak sebagai *reverse proxy* tersebut memiliki IP publik sehingga dapat diakses dari Internet.

Berdasarkan permasalahan di atas, penulis melakukan penelitian tentang "Analisis Dan Implementasi *Reverse Proxy* Sebagai Media Komunikasi *Client Server* Menggunakan *Apache*". Salah satu konsep *reverse proxy* adalah penyeimbang beban *web*, yaitu *reverse proxy* menjadi perantara *user-user* di internet terhadap akses ke *web-web server* yang berada pada LAN.

TINJAUAN PUSTAKA

Penelitian ini dikembangkan dari beberapa literatur pustaka sebagai referensi penelitian yang berhubungan dengan objek pembahasan. Diantaranya adalah penelitian oleh

(Nuriba, 2014), tentang Analisis Availabilitas *Load balancing* Pada *Web server* Lokal. Yang membahas implementasi *load balancing* dan menganalisa hasil availabilitas tersebut. Teknologi *load balancing* diimplementasikan menggunakan *Linux Virtual Server (LVS)*. Parameter yang dianalisa adalah *throughput* dan *response time*. Dari hasil analisa terhadap availabilitas *load balancing* yang telah dilakukan, sistem *load balancing* dapat menjadi salah satu solusi yang efektif dan efisien untuk menciptakan sistem yang handal dengan tingkat ketersediaan tinggi, khususnya sebagai *web server*.

(Noviyanto, 2015), adalah penelitian tentang Perancangan Dan Implementasi *Load balancing Reverse Proxy* Menggunakan *Haproxy* Pada Aplikasi *Web*. Menanggulangi permasalahan yang diakibatkan oleh beban *traffic*. Membagi beban secara merata kepada kedua *serverbackend*. *Request* yang datang dari *client* akan diterima oleh *balancer* dan akan diarahkan pada *server* yang siap menerima *request*, sedangkan pada *singleserver* tidak mampu menanggulangi beban trafik yang berlebih pada jaringan. Dengan metode *Load balancing* dan menggunakan *HAProxy* sebagai penyeimbang beban *traffic* yang terjadi pada *server*. Hasilnya adalah semua *request* akan langsung dilayani dan apabila *request* belum selesai dilayani *request-request* yang masuk tidak akan bisa dilayani atau *request time out* sehingga *server* yang sedang sibuk berkemungkinan kecil untuk mati/*down*.

(Prismana, 2016), Implementasi *Load Balancing* Pada *Web server* Dengan Menggunakan *Apache*. Peningkatan permintaan klien menyebabkan *web server* sibuk melayani permintaan dan sangat memungkinkan *web server* tidak mampu melayani permintaan klien. Hal tersebut dapat mengakibatkan *web server* menjadi *overload* dan akhirnya *web server* menjadi *down* salah satu mekanisme untuk lebih mengoptimalkan sumber daya *web server* yang ada adalah dengan menggunakan *loadbalancing* yang akan menyeimbangkan beban pada suatu *web server* yang sibuk sehingga dapat mempercepat waktu respon dari *web*. Berdasarkan hasil pengujian setelah diimplementasikan *apache* sebagai *loadbalancer* diperoleh bahwa setiap *web server* mendapatkan beban yang berurutan serta sama tanpa memiliki beban lebih, dan *web server* memiliki waktu tanggap lebih tinggi.

('Ala, 2016), tentang Perancangan Sistem *Proxy Server* Menggunakan Protokol WCCPv2 dengan Konfigurasi Multi Router. Desain dan implementasi *ProxyServer* menggunakan protokol WCCP (*WebCache Communication Protocol*) dengan konfigurasi multi router. *Proxyserver* yang dibuat menggunakan aplikasi *Squid*. Pengujian *proxyserver* yang dilakukan dalam dua kondisi, kondisi pertama pengujian yang dilakukan ketika *Squid* berada dalam keadaan mati dan pengujian kedua dilakukan ketika *Squid* berada dalam keadaan hidup. Perbandingan nilai rerata beberapa parameter selama pengujian menunjukkan bahwa *Squid* telah dapat melayani permintaan klien dengan cukup baik. Hal ini ditandai dengan jumlah permintaan dengan respon HIT berada di atas permintaan dengan respon MISS. Parameter *transaction* dan *successfultransaction*, yang secara berurutan mengalami kenaikan serta *responsetime* dan *failedtransaction* yang mengalami penurunan. *Proxyserver* juga mencakup pengujian kemampuan *Squid* untuk melakukan caching terhadap paket – paket pembaruan sistem operasi. Penurunan waktu unduh dan kenaikan kecepatan unduh paket disebabkan oleh permintaan klien yang sudah dapat dipenuhi secara lokal oleh *proxyserver*.

Landasan teori yang digunakan dalam penelitian ini adalah penjelasan teori dari buku yang berhubungan dengan penelitian ini, diantaranya :

Jaringan komputer adalah suatu himpunan interkoneksi sejumlah komputer *autonomous*. Dalam bahasa yang populer dapat dijelaskan bahwa jaringan komputer adalah kumpulan beberapa komputer (dan perangkat lain seperti *router*, *switch*, dan sebagainya) yang saling terhubung satu sama lain melalui media perantara. Media perantara ini bisa berupa kabel ataupun media tanpa kabel (nirkabel) (Sofana, 2013).

Proxy dapat dipahami sebagai pihak ketiga yang berdiri ditengah-tengah antara pihak kedua yang saling berhubungan dan berfungsi sebagai perantara, sehingga pihak pertama dan kedua tidak secara langsung berhubungan, akan tetapi berhubungan melalui perantara yaitu *proxy*. Pengertian *proxy* adalah *server* yang menyediakan suatu layanan untuk meneruskan setiap permintaan user kepada *server*lain yang terdapat di internet (Sora, 2015).

Server adalah sebuah sistem komputer yang menyediakan jenis layanan (*service*) tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan (*network operating system*)(Priyono, 2012).

Pengertian *Domain Name System* Digunakan untuk mentraslasi dari *IP address* ke *namahost* yang lebih mudah diingat sehingga klien dapat terhubung ke *web server* atau *email server* menggunakan nama domain bukan *IP address*. Saat berselancar dari satu *website* ke *website* lainnya atau mengirim *email* tentu saja *domain name system* telah berperan di dalamnya (Susilo, 2013).

Reverse Proxy merupakan sebuah *proxy* yang menghadap ke internet serta digunakan sebagai *front-end* berfungsi untuk mengontrol serta melindungi dari akses ke sebuah jaringan pribadi, adapun tugas tugasnya seperti, *load balancing*, *cahcing*, deskripsi dan otentikasi.(Turman, 2017).

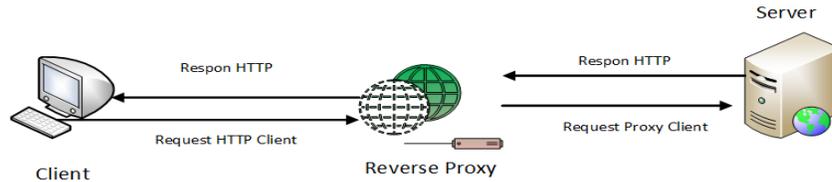
Penyusunan Analisis Dan Implementasi *Reverse Proxy* Sebagai Media Komunikasi *Client Server* Menggunakan *Apache* adalah mengobservasi mengumpulkan berbagai keluhan yang terjadi dilapangan, meneliti pada berbagai literatur yang berhubungan dengan penelitian studi terhadap system sejenis yang sudah dibangun agar mendapatkan gambaran secara menyeluruh terhadap sistem yang akan dibangun. Kemudian merancang sistem, setiap data yang telah dikumpulkan selanjutnya dilakukan analisis dan perancangan. Perancangan dimulai dari desain jaringan yang ada dan alur sistem yang akan diterapkan. Lalu manajemen sistem yang dilakukan dengan mengaplikasikan seluruh rancangan untuk manajemen beban kerja *server*.

Alat yang digunakan dalam penelitain adalah perangkat keras dan lunak dengan spesifikasi sebagai berikut:

1. *Hardware* : prosesor: AMD A8 6410 APU (4CPU) 2.0 GHz, RAM:2 GB DDR 3, hardisk: 500 GB, LCD: LED 14" , DVD-ROM, VGA: AMD Radeon R5 Graphics

2. *Hardware* : processor: Intel Xeon 4-Core E3-1231v3 3.4GHz/ 1600MHz, RAM:4 GB DDR 3, hardisk: 250 GB, LCD: LED 14", VGA: NVIDIA Geforce® GT 540M CUDA™ . 2 GB.
3. *Software* :Linux OS 12.04, *Apache2*, *Browser* firefox, *Puty*, *Webserver Stress Tool*.

Rancangan arsitektur merupakan gambaran alur proses dari jaringan komputer dibangun menjadi dua rancangan jaringan arsitektur di lingkungan *server*. Untuk menangani request konten internet yang diakses pada setiap komputer melalui jaringan lokal laboratorium komputer jaringan menggunakan *reverse proxy* dapat dilihat pada gambar 1



Gambar 1 Arsitektur sistem menggunakan *Reverse Proxy*

Pada gambar 1 menjelaskan tentang proses akses internet komputer *client* dialihkan ke *reverse proxy* sebelum terhubung dengan sumber internet, dengan ketentuan alur proses yang sudah dikonfigurasi pada *reverse proxy*, rancang jaringan ini dibangun dengan terdiri dari sebuah *server* dengan 1 buah *reverse proxy* dan digunakan sebagai *web accelerator* serta satu buah klien sebagai tester. *Reverse proxy* bertindak sebagai *gateway* penghubung antara klien dan *server*, serta berfungsi sebagai penyeimbang beban untuk meringankan kinerja *server*, sehingga *request* dari klien tidak langsung menuju *server* melainkan melalui *reverse proxy*. Dengan demikian beban *request* yang diterima oleh *server* diperingan.

PEMBAHASAN

Proses pengerjaan *server* dibutuhkan *software* yang berfungsi sebagai media pelayanan *webserver*, pada instalasi *apache2* dengan melakukan perintah pada terminal *linux* yang untuk mendapatkan *mod-proxy* sebelum melakukan konfigurasi dilakukan perintah untuk menginstall paket *apache2* dapat dilihat pada gambar 2

```
apt-getinstall libapache2-mod-proxy-html libxml2-dev
```

Gambar 2 Instalasi *Apache2*

Setelah proses penginstallan selesai maka paket pendukung untuk konfigurasi *reverse proxy* menggunakan *apache2*, kemudian mengaktifkan modul pendukung agar dapat berfungsi sebagai *reverse proxy* dapat dilihat pada gambar 3

```
a2enmod proxy_http
a2enmod proxy_ajp
a2enmod rewrite
a2enmod deflate
a2enmod headers
a2enmod proxy_connect
a2enmod proxy_html
```

Gambar 3 Modul pendukung *Reverse Proxy*

Proses mengkonfigurasi modul *proxy* pada direktori tempat *module-module* yang dibutuhkan oleh *apache2* untuk melakukan konfigurasi dengan menambahkan konfigurasi seperti pada gambar 4

```
ProxyRequests Off
<Proxy *>
    AddDefaultCharset off
    Order deny,allow
    Deny from all
    Allow from all
</Proxy>
```

Gambar 4 Konfigurasi modul *Reverse Proxy*

Konfigurasi *domain name server* pada *server utama* untuk memanggil *server1* dan *server2* dengan melakukan konfigurasi pada */etc/apache2/sites-available*, seperti pada gambar 5 dan pada gambar 6

```
server1@jarinet4: /etc/apache2/sites-available
GNU nano 2.2.6 File: server1.lab.akprind.ac.id.conf
<VirtualHost *:80>
  ServerName server1.lab.akprind.ac.id
  ProxyRequests off
  ProxyPass / http://192.168.1.5/
</VirtualHost>
```

Gambar 5 Konfigurasi *domain name server* pada *server1*

```
skripsi@serverLab: ~
GNU nano 2.2.6 File: server2.lab.akprind.ac.id.conf
<VirtualHost *:80>
  ServerName server2.lab.akprind.ac.id
  ProxyRequests off
  ProxyPass / http://192.168.1.4/
</VirtualHost>
```

Gambar 6 Konfigurasi *domain name server* pada *server2*

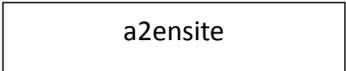
Gambar 5 dan gambar 6 menjelaskan proses konfigurasi *domain name server* sebagai sebuah *nama url* untuk memanggil *server* lokal yang berada dibawah *server utama* dengan nama *url http://server1.lab.akprind.ac.id* sebagai *server1* yang berada dan dengan nama *url http://server2.lab.akprind.ac.id* sebagai *server2*.

Konfigurasi *virtualhost* pada *server utama* berfungsi untuk melakukan pemanggilan *server* lokal, yaitu *server1* dan *server2* secara bergantian sesuai permintaan dengan melakukan konfigurasi *virtualhost* pada *reverse proxy* setelah menginstal *apache2* dan mengaktifkan modul *reverse proxy* untuk melakukan pemanggilan *server* lokal yang berada dibawah *server utama*, seperti pada gambar 7

```
<VirtualHost *:80>
  ServerName skripsi.lab.akprind.ac.id
  ProxyRequests Off
  <Proxy *>
    Order deny,allow
    Allow from all
  </Proxy>
  ProxyPass /balancer-manager !
  ProxyPass /balancer://mycluster/ stickysession=JSESSIONID nofailover=On
  ProxyPassReverse / http://server1.lab.akprind.ac.id/
  ProxyPassReverse / http://server2.lab.akprind.ac.id/
  <Proxy balancer://mycluster>
    BalancerMember http://server1.lab.akprind.ac.id/ route=public1
    BalancerMember http://server2.lab.akprind.ac.id/ route=public2
    ProxySet lbmethod=byrequests
  </Proxy>
  <Location /balancer-manager>
    SetHandler balancer-manager
  </Location>
  <Directory "/Library/WebServer/Documents">
    AllowOverride AuthConfig
  </Directory>
```

Gambar 7 Konfigurasi *virtual host reverse proxy* untuk *server* lokal

Proses selanjutnya *a2ensite* adalah perintah untuk melakukan pengaktifan *virtual host*, seperti pada gambar 8



Gambar 8 Perintah untuk mengaktifkan *Virtual Host*

Proses untuk memastikan apakah semua *server* terhubung dengan *server* utama adalah dengan mengecek konektivitas pada *reverse proxy* dengan melakukan perintah *ping* pada masing-masing ip *server* seperti pada gambar 9

```
skripsi@serverLab:~$ ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
64 bytes from 192.168.1.5: icmp_seq=1 ttl=64 time=0.399 ms
64 bytes from 192.168.1.5: icmp_seq=2 ttl=64 time=0.169 ms
64 bytes from 192.168.1.5: icmp_seq=3 ttl=64 time=0.350 ms
64 bytes from 192.168.1.5: icmp_seq=4 ttl=64 time=0.290 ms
^C
--- 192.168.1.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.169/0.302/0.399/0.085 ms
skripsi@serverLab:~$ ping 192.168.1.4
PING 192.168.1.4 (192.168.1.4) 56(84) bytes of data.
64 bytes from 192.168.1.4: icmp_seq=1 ttl=64 time=0.393 ms
64 bytes from 192.168.1.4: icmp_seq=2 ttl=64 time=0.325 ms
64 bytes from 192.168.1.4: icmp_seq=3 ttl=64 time=0.295 ms
64 bytes from 192.168.1.4: icmp_seq=4 ttl=64 time=0.168 ms
```

Gambar 9 Perintah pengujian koneksi pada *reverse proxy server*

Pengujian dilakukan dengan mengakses alamat *skripsi.lab.akprind.ac.id* sebagaimana yang telah dikonfigurasi pada *reverse proxy* yang terhubung dengan *server* lokal yang berada dibawah *server* utama yang telah dilakukan konfigurasi pada *virtualhostapache2* proses pengujian dilakukan untuk menguji keberhasilan konfigurasi di *virtualhostapacheserver* utama seperti pada gambar 10



Gambar 10 Pengujian akses *skripsi.lab.akprind.ac.id*

Tabel 1 menunjukkan data keseluruhan dari hasil uji peruser yang menghitung beban rata-rata akses secara langsung tanpa menggunakan *reverse proxy* menggunakan aplikasi *Webserver Stress Tools*.

Tabel 1 Data keseluruhan hasil uji per user

No	Name	Clicks	Avg. Click Time [ms]	Bytes	Kbit/s
1	Uji 1	78	512	932,646	186.66
2	Uji 2	72	582	860,904	164.36
3	Uji 3	65	747	765,248	126.06
4	Uji 4	59	623	705,463	153.58
5	Uji 5	52	571	621,764	167.51
6	Uji 6	32	1,337	371,023	69.35
7	Uji 7	38	650	454,366	147.13
8	Uji 8	33	673	394,581	142.17
9	Uji 9	28	565	334,796	169.42
10	Uji 10	20	835	239,140	114.55
Rata-Rata		47.7	709.5	567,993	144.079

Tabel 1 menunjukkan bahwa beban rata-rata akses secara langsung tanpa menggunakan *reverse proxy* mendapatkan rata-rata *clicks* 47.7, *Avg. Click Time* 709.5 (ms) dan 567,993 *Bytes*.

Tabel 2 Data keseluruhan hasil uji perURL

No	Name	Clicks	Time Spent (ms)	Avg. Click Time (ms)
1	Uji 1	77	72,240	32,922
2	Uji 2	71	76,300	28,139
3	Uji 3	64	61,824	26,261
4	Uji 4	58	67,724	23,089
5	Uji 5	50	57,776	21,478
6	Uji 6	31	42,908	18,275
7	Uji 7	36	36,216	16,019
8	Uji 8	32	30,498	12,757
9	Uji 9	25	5,262	11,329
10	Uji 10	19	14,888	8,321
Rata-Rata		46.3	46,564	19,859

Tabel 2 menunjukkan bahwa waktu rata-rata akses secara langsung tanpa menggunakan *reverse proxy* mendapatkan rata-rata *clicks* 46.3, *Time Spent* 46.564 (ms) dan 19.859 *Avg. Time* (ms).

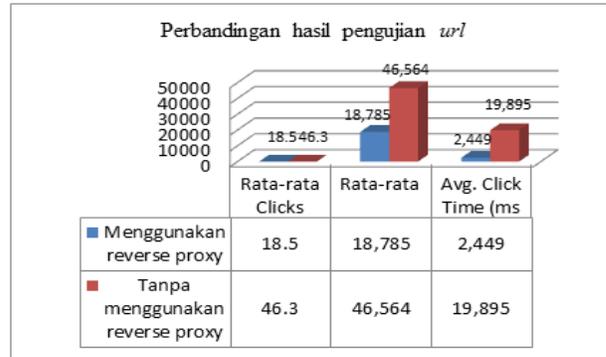
Pada bagian ini penulis akan menganalisis perbandingan hasil pengujian waktu rata-rata akses *url* dan akses *user* tanpa menggunakan *reverse proxy* dan dengan menggunakan *reverse proxy* berdasarkan dari data yang di dapat dari hasil uji menggunakan nilai rata-rata *clicks*, *Time Spent* (ms) dan *Avg. Click Time* (ms). Hasil pengujian berdasarkan akses *url* ditunjukkan pada tabel 3

Tabel 3 Perbandingan hasil pengujian *url*

Pengujian	Rata-rata Clicks	Rata-rata Time Spent (ms)	Avg. Click Time (ms)
Menggunakan <i>reverse proxy</i>	18.5	18,785	2,449
Tanpa menggunakan <i>reverse proxy</i>	46.3	46,564	19.859

Tabel 3 menunjukkan hasil waktu rata-rata akses *url* menggunakan *reverse proxy* dan akses *url* tanpa menggunakan *reverse proxy*. Hasil pengujian *url* menggunakan *reverse*

proxymendapatkan waktu rata-rata clicks 18.5 dengan Time Spent 18,785 (ms) dan waktu rata-rata clicks per url2,449AVG. Time(ms). Dan hasil pengujian urltanpa menggunakan reverse proxy adalah mendapatkan waktu rata-rata clicks 46.3 dengan Time Spent 46,564 (ms) dan waktu rata-rata clicks per url19.859 AVG. Time(ms).



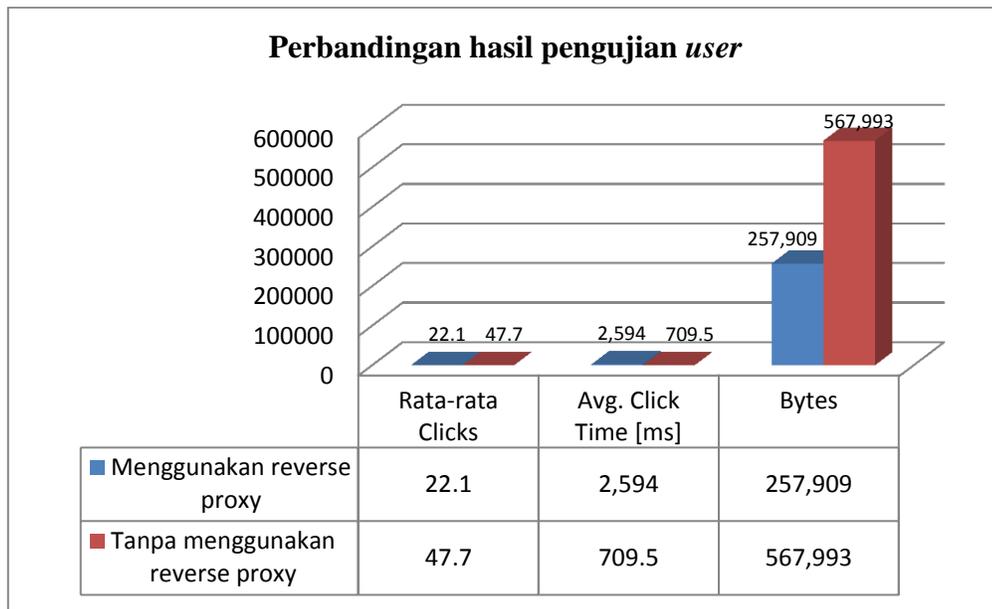
Gambar 11 Grafik perbandingan hasil pengujian url

Gambar 11 menunjukkan hasil waktu rata-rata akses *url* menggunakan *reverse proxy* dan akses *url* tanpa menggunakan *reverse proxy*. Hasil akses pengujian *url* menggunakan *reverse proxy* adalah mendapatkan waktu rata-rata clicks 18.5 dengan *Time Spent* 19,946 (ms) dan waktu rata-rata clicks per *url* 2,449 *AVG. Time*(ms). Dan hasil akses pengujian *url* tanpa menggunakan *reverse proxy* adalah mendapatkan waktu rata-rata clicks 46.3 dengan *Time Spent* 46,564 (ms) dan waktu rata-rata clicks per *url* 19.859 *AVG. Time*(ms). Hasil pengujian berdasarkan akses *user* ditunjukkan pada tabel 4

Tabel 4 Perbandingan hasil pengujian user

Pengujian	Rata-rata Clicks	Avg. Click Time [ms]	Bytes
Menggunakan <i>reverse proxy</i>	22.1	2,594	257,909
Tanpa menggunakan <i>reverse proxy</i>	47.7	709.5	567,993

Tabel 4 menunjukkan hasil beban rata-rata akses *user* menggunakan *reverse proxy* dan akses *user* tanpa menggunakan *reverse proxy*. Hasil akses pengujian *user* menggunakan *reverse proxy* adalah mendapatkan beban rata-rata clicks 22.1 dengan 2,594 *Avg. Click Time* (ms), dan 257,909 *bytes* per *user*. Hasil akses pengujian *user* tanpa menggunakan *reverse proxy* adalah mendapatkan beban rata-rata clicks 47.7 dengan 709.5 *Avg. Click Time* (ms), dan 567,993 *bytes* per *user* 257,909.



Gambar 12 Perbandingan hasil pengujian user

Gambar 12 menunjukkan hasil beban rata-rata akses user menggunakan *reverse proxy* dan akses user tanpa menggunakan *reverse proxy*. Hasil akses pengujian user menggunakan *reverse proxy* adalah mendapatkan beban rata-rata *clicks* 22.1 dengan 2,594 *Avg. ClickTime* (ms), dan 257,909 *bytes* per user. Hasil akses pengujian user tanpa menggunakan *reverse proxy* adalah mendapatkan beban rata-rata *clicks* 47.7 dengan 709.5 *Avg. ClickTime* (ms), dan 567,993 *bytes* per user.

Berdasarkan analisis hasil perbandingan tabel 3 dan tabel 4 terjadi peningkatan waktu yang dihabiskan oleh klien terhadap akses pada *web server* dengan menggunakan media *reverse proxy* sejumlah 59% dan pengurangan beban kinerja *web server* dalam melayani permintaan klien sejumlah 54%, sehingga dalam pengoptimalisasi dan kestabilan akses *web server* dapat dinyatakan berhasil.

KESIMPULAN

Kesimpulan yang dapat diambil dari skripsi ini adalah sebagai berikut:

1. Peningkatan proses pelayanan permintaan dari klien ke *web server* dengan menggunakan *reverse proxy* lebih cepat dibandingkan tanpa menggunakan *reverse proxy*.
2. Dari hasil pengujian konfigurasi menggunakan *reverse proxy* memperoleh hasil peningkatan waktu yang dihabiskan oleh klien terhadap akses pada *web server* dengan menggunakan media *reverse proxy* sejumlah 60% dan pengurangan beban kinerja *web server* dalam melayani permintaan klien sejumlah 54%.
3. *Reverse proxy* menjadi sebuah jembatan penghubung antara klien dengan *server* sehingga keberadaan *server* akan terjamin keamanannya karena *server* tidak terhubung secara langsung kepada klien.

Saran

Berdasarkan percobaan terhadap aplikasi, ternyata masih terdapat kesalahan dalam menampilkan isi dari tampilan, sehingga perlu adanya penyempurnaan dan pengembangan sistem dalam hal sebagai berikut:

1. Melakukan penambahan metode optimalisasi agar hasil pelayanan lebih maksimal agar hasil pelayanan *web server* kepada klien lebih optimal.
2. Melakukan penambahan jumlah *server* sebagai pembanding proses layanan permintaan dari klien.

3. Selain menggunakan *Webserver Stress Tools*, dapat melakukan pengujian *webserver* dengan menggunakan *Apache Benchmark (AB)*, *ApacheJmeter*, dan *WAPT (Web Application Performance Testing)* sebagai perbandingan.

DAFTAR PUSTAKA

- 'Ala, F. M. (2016). "Perancangan Sistem Proxy Server Menggunakan Protokol WCCPV2 dengan Konfigurasi Multi Router". *Jurnal Teknologi dan Sistem Komputer, Vol.4, No.2, April 2016 (e-ISSN: 2338-0403)*.
- Noviyanto, A. B. (2015). "Perancangan Dan Implementasi Load Balancing Reverse Proxy Menggunakan Haproxy Pada Aplikasi Web". *Jurnal JARKOM Vol. 3 No. 1 Desember 2015 Teknik Informatika, IST AKPRIND Yogyakarta ISSN:2338-6313*.
- Nuriba, D. (2014). ANALISIS AVAILABILITAS LOAD BALANCING PADA WEB SERVER LOKAL. TA - S1 *Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro Semarang*.
- Prismana, I. G. (2016). Implementasi Load Balancing Pada Web Server Dengan Menggunakan Apache . *Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya. Volume 5 Nomer 2 Tahun 2016*.
- Priyono, D. T. (2012). PEMBANGUNAN SERVER PROXY SQUID MENGGUNAKAN UBUNTU SERVER 11.10 PADA SEKOLAH TINGGI KEGURUAN ILMU PENIDIDKAN PGRI PACITA. *Indonesian Journal on Networking and Security* .
- Sofana. (2013). Membangun Jaringan Internet. *Informatika Bandung*.
- Sora, N. (2015, 06 14). *Pengertian Proxy dan Fungsinya Serta Cara Kerjanya*. Retrieved from Pengertianku: <http://www.pengertianku.net/2015/06/pengertian-proxy-dan-fungsinya-serta-cara-kerjanya.html>
- Susilo, I. (2013). PEMBANGUNAN WEB SERVER MENGGUNAKAN DEBIAN SERVER UNTUK MEDIA PEMBELAJARAN DI SEKOLAH MENENGAH KEJURUAN (SMK) NEGERI 1 SRAGEN . *Indonesian Journal on Networking and Security*.
- Turman. (2017, Mei 10). *Kunci Komputer*. Retrieved from Kunci Komputer: <http://www.kuncikomputer.com/umum/penjelasan-proxy-server-tujuan-jenis-proxy-dan-penjelasan-lainnya-terlengkap/>