

IMPLEMENTASI LOAD BALANCING SISTEM MAIL SERVER SERTA PENINGKATAN CBPOLICYD MENGGUNAKAN ZIMBRA

Mohammad Hud¹, Riad Sahara², Syahid Abdullah³

¹²³Program Studi Teknik Informatika, Fakultas Informatika, Universitas Siber Asia

Jl. Harsono RM No.1 Ps. Minggu, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12550

¹ 2024.mohammadhud@student.unsia.ac.id, ² riadsahara@lecturer.unsia.ac.id, ³ syahidabdullah@lecturer.unsia.id

Abstract

In the rapidly evolving technological era, security and availability of email services are pivotal aspects requiring special attention. Email servers play a crucial role in providing fast and up-to-date access to information, however with the growth in the number of clients, server workload can significantly increase. Threats to email services, particularly from spam attacks, can lead to performance losses and serious security risks. Therefore, in this study, the author attempts to find preventive solutions and enhance the security of email servers, which are deemed essential. By implementing load balancing on zimbra multi servers using zimbra proxy, the effectiveness of evenly distributing server workload is demonstrated. Additionally, integration with the security features of cbpolicyd on zimbra can provide an optimal additional security layer. This research, employing experimental and analytical methods, aims to improve the availability of email services, distribute workload, and strengthen email server security with advanced filtering policies and message management. The implementation results successfully integrate mailbox servers with load balancing using zimbra proxy, and the application of cbpolicyd successfully manages more sophisticated security policies. The significant contribution of this research lies in optimizing the performance and security of the email system, offering contemporary solutions to face challenges in the current digital era.

Keywords : mail server, load balancing, zimbra proxy, cbpolicyd

Abstrak

Dalam era teknologi yang berkembang pesat seperti saat ini, keamanan dan ketersediaan layanan email menjadi aspek utama yang memerlukan perhatian khusus. Server email memiliki peran utama dalam menyediakan akses yang cepat dan terkini terhadap informasi, namun dengan pertumbuhan jumlah klien, beban kerja server dapat meningkat secara signifikan. Ancaman pada layanan email, terutama dari serangan spam, dapat mengakibatkan kerugian kinerja dan risiko keamanan yang cukup serius. Oleh karena itu, pada penelitian kali ini penulis mencoba untuk mencari solusi pencegahan dan peningkatan keamanan server email yang menjadi esensial. Dengan melakukan implementasi *load balancing* pada zimbra multi server menggunakan zimbra proxy membuktikan langkah efektif dalam mendistribusikan kinerja server secara merata. Selain itu integrasi dengan fitur keamanan *cbpolicyd* pada zimbra dapat memberikan lapisan keamanan tambahan yang optimal. Penelitian ini, melalui metode eksperimen dan analisis, bertujuan untuk meningkatkan ketersediaan layanan email, membagi beban kerja, dan memperkuat keamanan server email dengan kebijakan filtrasi dan manajemen pesan yang canggih. Hasil implementasi berhasil mengintegrasikan server mailbox dengan metode *load balancing* menggunakan zimbra Proxy, dan penerapan *cbpolicyd* sukses mengatur kebijakan keamanan yang lebih canggih. Kontribusi signifikan dari penelitian ini terletak pada optimalisasi kinerja dan keamanan sistem email, dan diharapkan dapat memberikan solusi terkini untuk menghadapi tantangan di era digital saat ini.

Kata kunci : server email, load balancing, zimbra proxy, cbpolicyd

1. PENDAHULUAN

Pada era teknologi yang terus berkembang, terutama dalam komunikasi, perkembangan ini terjadi begitu cepat. Kebutuhan akan informasi yang cepat semakin meningkat, terutama dalam mengakses informasi terkini. Salah satu faktor utama yang memengaruhi kecepatan akses pengiriman email adalah *server* email. *Server* email berperan sebagai pelayan yang menerima permintaan dari klien yang terhubung ke *server* email yang ingin diakses. Sebagai pusat pengelolaan, semakin banyaknya klien akan meningkatkan beban kerja *server* email. Oleh karena itu, spesifikasi komputer, terutama *server* email, harus optimal agar dapat melayani permintaan dari banyak klien. Dengan meningkatnya jumlah layanan email, hal ini menyebabkan situs *webmail* bekerja lebih banyak pada *server* yang melayani permintaan. Akibatnya, performa *server* dapat menurun dan sering terjadi gangguan pada layanan email [1]. Tingkat kenyamanan dan keamanan juga menurun sebagai akibat dari semakin populernya teknologi email saat ini. Spam adalah salah satu alasan mengapa keamanan email menurun. Spam adalah pesan elektronik yang tidak diinginkan atau tidak diminta yang dikirimkan kepada sejumlah besar orang secara massal, seringkali untuk tujuan iklan, promosi, atau penipuan. Jika pesan spam tidak diatasi dengan cepat, ini dapat menyebabkan *server* email menjadi terbebani, memperlambat kinerja, dan bahkan memicu kegagalan *server*. Untuk mencegah hal ini terjadi pada *server* email suatu organisasi, dibutuhkan tindakan pencegahan keamanan yang dilakukan pada *server* email [2].

Menurut [3] untuk menangani jumlah layanan email yang terus meningkat, dibutuhkan sistem *server* yang handal dalam melayani trafik yang besar dan transaksi email yang tinggi. Dengan meliputi pembagian trafik masuk dan keluar pada *mail server* tersebut agar skalabilitas dari *server* dapat meningkat. Salah satu solusi dari permasalahan ini adalah penerapan metode *load balancing* pada *multi server*. *Load balancing* pada *server* merupakan salah satu cara yang dapat digunakan untuk meningkatkan kinerja dan ketersediaan *server*. Dengan cara membagi permintaan layanan yang datang ke beberapa *server* sekaligus, sehingga beban yang ditanggung oleh masing-masing *server* lebih sedikit [3], [4]. Ketika *mail server* sudah aktif dan sedang melakukan tugasnya, disaat itulah seringkali berbagai jenis ancaman datang seperti banyak spam, masuknya virus, ip *server* terblacklist, gagal mengirim pesan, pesan masuk spam folder dan

performa *mail server* menurun [5]. Faktor keamanan merupakan salah satu faktor terpenting yang perlu diperhatikan pada *server* email [6]. Oleh sebab itu diperlukan fitur keamanan lebih untuk mengantisipasi atau mencegah ancaman itu sebelum masuk kedalam *server* email. Dengan menggunakan fitur tambahan seperti *cbpolicyd* pada *mail server* zimbra, dapat membantu mengidentifikasi dan menghapus pesan spam sebelum mencapai tujuan. Selain itu dengan mengaktifkan *cbpolicyd*, seorang sistem *administrator* dapat membuat pembatasan, mencegah spam, dan memastikan alur email, sehingga menjaga skalabilitas *server* email. Dengan demikian, implementasi *cbpolicyd* pada *mail server* zimbra penting untuk meningkatkan keamanan *server* email [5].

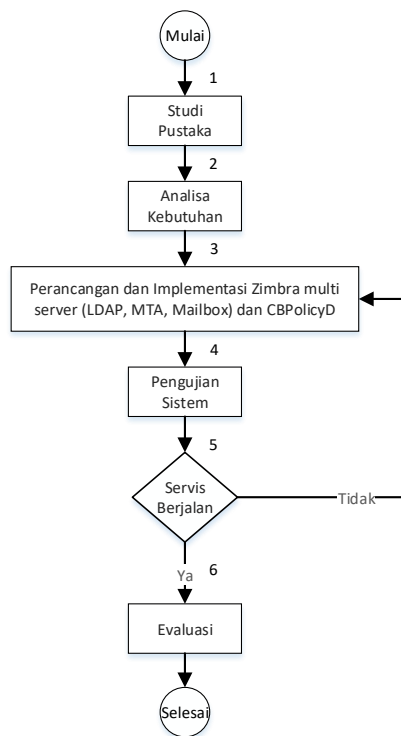
Diharapkan dengan mengimplementasikan sistem *load balancing* pada *server* email dapat memberikan manfaat besar dalam meningkatkan ketersediaan layanan email dalam hal kinerja sistem dengan membagi beban kerja secara merata serta pengimplementasian *security cbpolicyd* dapat meningkatkan keamanan dengan kebijakan filtrasi dan manajemen pesan yang lebih canggih. Sehingga penelitian ini bertujuan untuk mengimplementasikan *load balancing* pada *mail server* untuk kestabilan *server* dengan cara manajemen yang baik serta peningkatan *security* menggunakan *cbpolicyd*. Hal tersebut merupakan perbedaan penelitian dari penelitian sebelumnya yang sudah pernah dilakukan.

Selain itu penelitian ini diharapkan dapat memberikan kontribusi yang signifikan sebagai berikut:

- Berkelanjutan untuk pengembangan sistem email di era digital seperti saat ini.
- Industri dan organisasi dapat menggunakan implementasi *loadbalancing* dan kebijakan keamanan menggunakan *cbpolicyd* untuk meningkatkan kinerja serta ketangguhan dalam sistem email mereka.
- Dapat menjadi landasan untuk penelitian lebih lanjut dalam membantu memahami dan menerapkan teknologi email yang aman dan handal.

2. METODOLOGI PENELITIAN

Pada bagian ini akan dijelaskan bagaimana proses keseluruhan penelitian ini dilakukan. Untuk memperjelas jalannya penelitian ini secara lebih detail dan metode yang akan dilakukan dapat dilihat pada Gambar 1.



Gambar 1. Diagram Metode Penelitian

- 1) Untuk tahap awal dilakukan studi pustaka serta observasi dengan membaca karya ilmiah, artikel, dan jurnal untuk memperoleh informasi mengenai *mail server*, *load balancing* dan *cbpolicyd* sehingga dapat mendukung kelancaran penelitian ini.
- 2) Pada tahap kedua dilakukan analisa kebutuhan perangkat dan sistem untuk melakukan implementasi *load balancing* menggunakan *zimbra proxy* dan peningkatan keamanan dengan *cbpolicyd* pada *mail server* menggunakan *zimbra* [15].
- 3) Tahap ketiga dilakukan perancangan topologi sistem. Percobaan pada penelitian ini menggunakan *virtual machine* (VM) [16] dengan konfigurasi *zimbra multi server* menggunakan 4 *server* VM, yaitu satu buah *server* LDAP dan DNS, dua buah *server mailbox*, dan satu buah *server* MTA dan *Proxy*. Semua *server* tersebut menggunakan sistem operasi linux ubuntu versi 20.04 LTS.
- 4) Pada tahap ini dilakukan pengujian sistem dengan parameter pengujian distribusi beban menggunakan *zimbra proxy* dan kinerja keamanan *cbpolicyd*. Pengujian *load balancing*

dilakukan dengan melakukan kirim pesan antar *user* pada *server mailbox* yang berbeda [16]. Pengujian fitur keamanan *cbpolicyd* dilakukan dengan memastikan bahwa semua fungsi yang telah diatur dalam *cbpolicyd* berjalan sesuai yang diharapkan termasuk aturan kebijakan, identifikasi spam, dan pengelolaan pesan email yang masuk [17].

- 5) Jika tahap pengujian sistem berhasil dilakukan akan dilanjutkan ke tahap evaluasi. Namun jika tahap pengujian sistem tidak berhasil, maka akan kembali ke tahap perancangan topologi sistem untuk merancang ulang topologi sistem.
- 6) Terakhir tahap evaluasi, pada tahap ini akan dilakukan evaluasi terhadap implementasi metode *load balancing* dan fitur *cbpolicyd* tersebut dengan menggunakan parameter pengujian yang telah ditentukan. Dari hasil evaluasi, maka akan terlihat nilai hasil dari uji parameter yang telah dilakukan.

2.1. Analisis

Setelah melakukan analisa kebutuhan perangkat dan sistem VM untuk melakukan percobaan implementasi. Pada tabel 1 dijelaskan kebutuhan spesifikasi perangkat. Serta konfigurasi alamat IP sistem yang dibutuhkan pada implementasi kali ini dapat dilihat pada tabel 2.

TABEL 1. SPESIFIKASI PERANGKAT DAN SISTEM

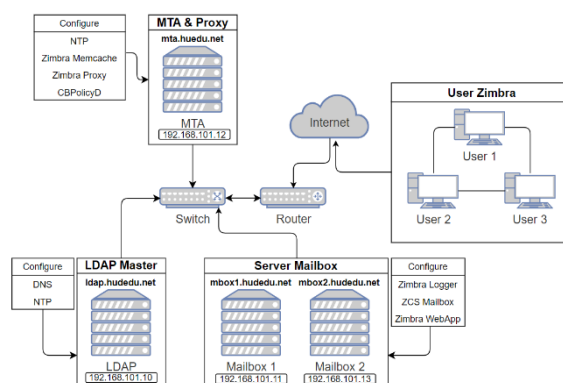
No	Nama Server	Spesifikasi	Sistem Operasi
1	Server LDAP & DNS	CPU 4 Core RAM 8 GB HDD 20 GB	Ubuntu 20.04
2	Server Mailbox1	CPU 4 Core RAM 8 GB HDD 20 GB	Ubuntu 20.04
3	Server MTA & PROXY	CPU 4 Core RAM 8 GB HDD 20 GB	Ubuntu 20.04
4	Server Mailbox2	CPU 2 Core RAM 4 GB HDD 20 GB	Ubuntu 20.04
5	User Zimbra	CPU 4 Core RAM 4 GB HDD 10 GB	Windows 10

TABEL II. ALAMAT IP SISTEM

No	Nama Perangkat	IP Address	Hostname
1	Server LDAP & DNS	192.168.101.10/24	ldap.hudedu.net
2	Server Mailbox1	192.168.101.11/24	mbox1.hudedu.net
3	Server MTA & PROXY	192.168.101.12/24	mta.hudedu.net
4	Server Mailbox2	192.168.101.13/24	mbox2.hudedu.net
5	User Zimbra	DHCP	-

2.2. Desain Penelitian

Pada design penelitian dari sistem *load balancing* dan peningkatan *cbpolicyd* ini terdapat empat bagian mesin *virtual* seperti diperlihatkan pada Gambar 2 yang mana topologi jaringan terdiri dari *server* MTA, *server* LDAP, *server* mailbox 1 dan 2, serta *user* atau pengguna zimbra [19].



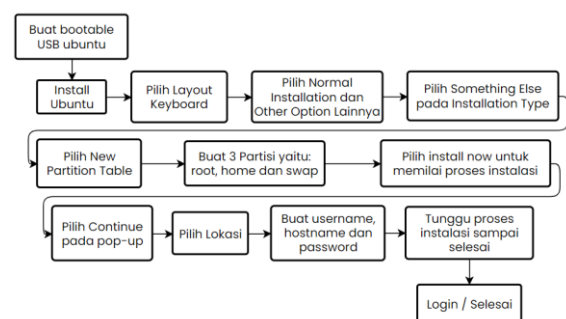
Gambar 2. Topologi Jaringan

Pada bagian yang pertama terdiri dari satu buah *server* MTA dan *Proxy*, *Server* MTA bertugas untuk menerima serta mengirim email dan meneruskannya ke masing-masing *mailbox* yang ada serta dipasang fitur *cbpolicyd* sebagai peningkatan keamanan *server* email. Sedangkan *proxy* bertugas sebagai pembagi beban dua *server* mailbox. Selanjutnya *server* MTA tersebut dikonfigurasi dengan alamat IP 192.168.101.12. Kemudian pada bagian kedua adalah *server* LDAP dan DNS dengan alamat IP 192.168.101.10, LDAP bertugas sebagai tempat penyimpanan data seluruh konfigurasi *user* dan *password*. Sedangkan DNS merupakan nama *domain* untuk mengakses *webmail*. Bagian ketiga terdapat *server* mailbox

untuk menerima request dari pengakses serta menempatkan data ke dalam media penyimpanan. *Server* mailbox terdiri dari 2 buah *server*, keduanya menggunakan alamat IP dan hostname yang berbeda yaitu *mbox1.hudedu.net* dengan alamat IP 192.168.101.11 dan *mbox2.hudedu.net* dengan alamat IP 192.168.101.13. Terakhir untuk bagian yang keempat adalah *user* atau pengguna *webmail*, bertugas untuk melakukan pengujian nantinya *user* tidak perlu tahu dimana akunnya tersimpan apakah di *mailbox* 1 atau *mailbox* 2, *user* hanya cukup mengakses laman *mail.hudedu.net* dan sistem akan secara otomatis mengarahkan sesuai *mailbox* masing-masing *user*.

2.3. Implementasi Sistem

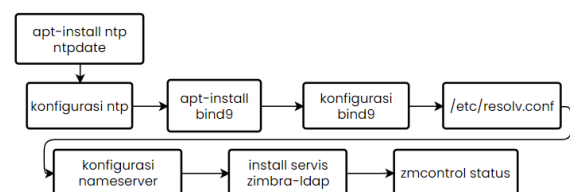
1) Instalasi Linux Ubuntu 20.04 LTS



Gambar 3. Tahapan Instalasi Ubuntu

Proses instalasi Ubuntu *Server* 20.04 LTS dimulai dengan mengunduh file ISO dari situs resmi Ubuntu dan membuat media instalasi seperti USB *bootable*. Setelah *booting* dari media tersebut, pilih opsi "*Install Ubuntu Server*" dan ikuti langkah-langkah untuk konfigurasi bahasa, lokasi, keyboard, serta pengaturan jaringan. Selanjutnya, tentukan partisi instalasi, konfigurasikan pengguna dan kata sandi administratif. Terakhir selesaikan proses dengan instalasi *bootloader*, dan setelah *reboot*, sistem Ubuntu *Server* 20.04 LTS sudah siap digunakan.

2) Instalasi LDAP dan DNS

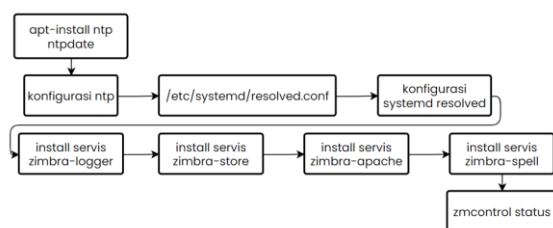


Gambar 4. Tahapan Instalasi LDAP

Untuk menginstal LDAP dan DNS pada Zimbra dilakukan pada satu server yang sama. Pertama-

tama instal dan konfigurasi NTP dengan perintah "`apt-install ntp ntpdate`" dan sesuaikan konfigurasi NTP di "`/etc/ntp.conf`". Lanjutkan dengan instalasi BIND9 menggunakan "`apt-install bind9`". Selanjutnya, edit "`/etc/resolv.conf`" untuk menambahkan alamat DNS server. Setelah itu, instal servis Zimbra LDAP dengan mengunduh dan mengekstrak paket instalasi Zimbra, lalu jalankan "`./install.sh`" dan pilih opsi instalasi LDAP. Terakhir, masuk sebagai pengguna Zimbra, dan cek status layanan LDAP menggunakan "`zmcontrol status`".

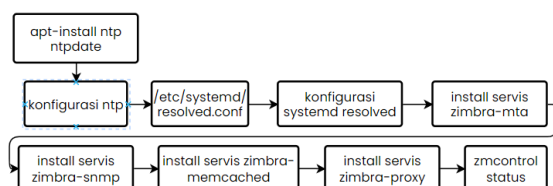
3) Instalasi Mailbox



Gambar 5. Tahapan Instalasi Mailbox

Tahap instalasi mailbox dimulai dengan instalasi dan konfigurasi NTP agar waktu pada server mailbox sinkron dengan server LDAP. Gunakan perintah "`apt-install ntp`" untuk instalasi, lalu sesuaikan konfigurasi di "`/etc/ntp.conf`". Selanjutnya, setting "`/etc/resolv.conf`" dan konfigurasi `systemd resolved` untuk menambahkan alamat DNS server. Instal servis Zimbra Logger, Store, Apache, dan Spell dengan berurutan menggunakan perintah "`su - zimbra`" dan jalankan installer Zimbra. Pilih servis instalasi yang sesuai untuk mailbox, seperti "`zimbra-logger`", "`zimbra-store`", "`zimbra-apache`", dan "`zimbra-spell`". Terakhir, cek status servis menggunakan "`zmcontrol status`". Lakukan instalasi mailbox tersebut pada server mailbox 1 dan 2.

4) Instalasi MTA dan Proxy

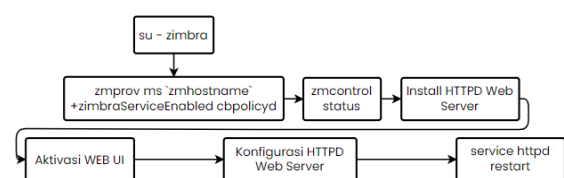


Gambar 6. Tahapan Instalasi MTA

Tahap instalasi MTA dan Proxy dimulai dengan instalasi dan konfigurasi NTP agar waktu

pada server MTA sinkron dengan server LDAP. Gunakan perintah "`apt-install ntp`" untuk instalasi, lalu sesuaikan konfigurasi di "`/etc/ntp.conf`". Selanjutnya, setting "`/etc/resolv.conf`" dan konfigurasi `systemd resolved` untuk menambahkan alamat DNS server. Setelah itu, lakukan instalasi servis Zimbra MTA melalui perintah "`su - zimbra`" dan eksekusi installer Zimbra. Lanjutkan dengan memilih instalasi servis Zimbra SNMP, Memcached, dan Proxy sesuai dengan urutan instalasi yang dibutuhkan. Lalu cek status servis menggunakan "`zmcontrol status`".

5) Aktivasi CBPolicyD



Gambar 7. Tahapan Aktivasi CBPolicyD

`CBPolicyD` diaktifkan pada sisi server MTA, langkah pertama adalah masuk ke user Zimbra dengan perintah "`su - zimbra`". Selanjutnya, gunakan perintah "`zmprov ms 'zmhostname' +zimbraServiceEnabled cbpolicyd`" untuk mengaktifkan `cbpolicyd`. Lalu periksa status dengan "`zmcontrol status`" untuk memastikan layanan telah aktif. Setelah itu, instal dan konfigurasi HTTPD web server, restart layanan HTTPD dengan perintah "`service httpd restart`".

2.4. Pengujian Sistem

Pada tahap pengujian implementasi awal akan dilihat dua hal. Pertama memastikan bahwa sistem *load balancing* pada mailbox 1 dan mailbox 2 sudah menyatu pada server MTA. Kedua memastikan apakah pengaturan aturan kebijakan dan peningkatan keamanan berhasil diimplementasikan dan dapat diakses dengan baik.

1) Pengujian Load Balancing

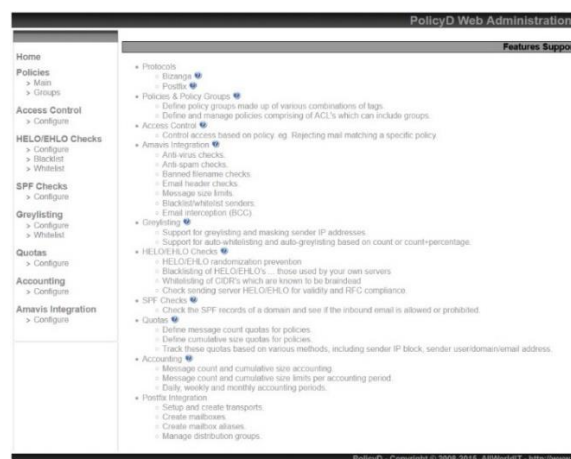
Dapat dilihat pada Gambar 8 server mailbox 1 dan mailbox 2 sudah berhasil disatukan dengan metode *load balancing* menggunakan zimbra Proxy, Dengan begitu akan tercipta integrasi yang seimbang antara kedua server mailbox. halaman ini dapat diakses oleh administrator melalui laman <https://mail.hudedu.net:9071/zimbraAdmin>.

Server	Layanan	Waktu
mbx2.hudedu.net	mailbox	17 Jan 2024 19:04
	stats	17 Jan 2024 19:04
	zimlet	17 Jan 2024 19:04
	zimbraAdmin	17 Jan 2024 19:04
	spell	17 Jan 2024 19:04
	zimbra	17 Jan 2024 19:04
	zmconfigd	17 Jan 2024 19:04
	service	17 Jan 2024 19:04
	logger	17 Jan 2024 19:04
ldap.hudedu.net	zmconfigd	17 Jan 2024 19:04
	stats	17 Jan 2024 19:04
	ldap	17 Jan 2024 19:04
mta.hudedu.net	stats	17 Jan 2024 19:04
	zmconfigd	17 Jan 2024 19:04
	amavis	17 Jan 2024 19:04
	mta	17 Jan 2024 19:04
	memcached	17 Jan 2024 19:04
	snmp	17 Jan 2024 19:04
	proxy	17 Jan 2024 19:04
	opendkim	17 Jan 2024 19:04
	antispam	17 Jan 2024 19:04
	antivirus	17 Jan 2024 19:04
mbx1.hudedu.net	zmconfigd	17 Jan 2024 19:04
	mailbox	17 Jan 2024 19:04
	zimlet	17 Jan 2024 19:04
	logger	17 Jan 2024 19:04
	zimbraAdmin	17 Jan 2024 19:04
	stats	17 Jan 2024 19:04
	service	17 Jan 2024 19:04
	zimbra	17 Jan 2024 19:04
	spell	17 Jan 2024 19:04

Gambar 8. Servis load balancing

2) Pengujian CBPolicyD

Pada Gambar 9 antarmuka *cbpolicyd* sudah dapat diakses dan dilakukan pengaturan keamanan, halaman ini dapat diakses melalui laman <http://mail.hudedu.net:8880/webui/index.php>. Dengan akses ke halaman web ini seorang administrator dapat secara efektif mengkonfigurasi dan menyesuaikan kebijakan keamanan untuk meningkatkan kinerja dan keamanan sistem email.



Gambar 9. Antarmuka CBPolicyD

3. HASIL DAN PEMBAHASAN

Untuk mengetahui apakah implementasi ini berjalan dengan baik, maka akan dilakukan

pengujian hasil lebih lanjut dari implementasi tersebut. Proses pengujian dibagi menjadi dua tahap yaitu pengujian *load balancing* zimbra proxy dan pengujian keamanan *cbpolicyd*.

1) Pengujian Load Balancing

Pada tahap ini akan dilakukan pengujian apakah server mailbox 1 dan 2 sudah terhubung dengan menggunakan zimbra proxy. Dengan melakukan skenario kirim pesan antar dua user zimbra pada server mailbox yang berbeda, user 1 dari server mailbox 1 dan user 2 dari server mailbox 2.

Beranda - Kelola - Akun - user1@hudedu.net	
ID:	7217447d-79fa-4916-a59c-7b6e
Dibuat:	17 Januari 2024 19:02:15
Server:	mbx1.hudedu.net
Email:	user1@hudedu.net
Status:	Aktif
Kuota:	0.003 MB dari 5 MB
Masuk Terakhir:	17 Januari 2024 19:03:42

Gambar 10. User Zimbra 1

Beranda - Kelola - Akun - user2@hudedu.net	
ID:	b53ea7f3-74d3-4e08-9cce-7ca
Dibuat:	17 Januari 2024 19:01:07
Server:	mbx2.hudedu.net
Email:	user2@hudedu.net
Status:	Aktif
Kuota:	0.003 MB dari 5 MB
Masuk Terakhir:	17 Januari 2024 19:03:10

Gambar 11. User Zimbra 2

Untuk mengetahui skenario pengujian berjalan sesuai dengan lancar atau tidak akan dilihat melalui log zimbra pada server mta dengan perintah seperti pada Gambar 12.

```
root@mta:~# /opt/zimbra/libexec/zmmsgtrace -r @hudedu.net
```

Gambar 12. Perintah Log

Hasil dari log diatas akan memberikan seluruh informasi tentang transaksi email dengan domain @hudedu.net. Pada Gambar 13 dapat dilihat bahwa user1@hudedu.net berhasil mengirim pesan ke user2@hudedu.net begitupun sebaliknya dengan status sent.

```
Message ID '1507279785.1.1705493012516.JavaMail.zimbra@hudedu.net'
user1@hudedu.net -->
Recipient user2@hudedu.net
Jan 17 19:03:33 - mbox2.hudedu.net (192.168.101.13) --> 127.0.0.1:10026 (127.0.0.1:10026) status sent
Jan 17 19:03:33 - mta --> 127.0.0.1:10032 (127.0.0.1:10032) status sent
Jan 17 19:03:34 - mta --> mbox1.hudedu.net:7025 (192.168.101.11:7025) status sent

Message ID '1824773764.6.1705493043802.JavaMail.zimbra@hudedu.net'
user2@hudedu.net -->
Recipient user1@hudedu.net
Jan 17 19:04:04 - mbox1.hudedu.net (192.168.101.11) --> 127.0.0.1:10026 (127.0.0.1:10026) status sent
Jan 17 19:04:04 - mta --> 127.0.0.1:10032 (127.0.0.1:10032) status sent
Jan 17 19:04:04 - mta --> mbox2.hudedu.net:7025 (192.168.101.13:7025) status sent
```

Gambar 13. Log Pesan Terkirim

Dengan begitu dapat disimpulkan ketika *user 1* mengirim pesan kepada *user 2*, *zimbra proxy* akan bekerja dengan menerima pesan tersebut dan menempatkannya pada *mailbox* yang sesuai dengan lokasi *user* tersebut berada.

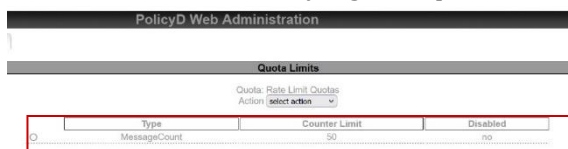
2) Pengujian CBPolicyD

Pengujian *cbpolicyd* dilakukan dengan membuat aturan atau kebijakan baru, dalam hal ini akan dilakukan aturan *rate limit* atau batas pengiriman pesan. Aturan ini digunakan untuk mengontrol jumlah permintaan atau pesan tertentu dalam batas waktu tertentu.

TABEL III. SKENARIO RATE LIMIT

No	User	Limit	Durasi
1	User1	50 Pesan	1800 Detik
2	User2	Bypass Rules	Bypass Rules

Untuk menguji keberhasilan implementasi kebijakan *rate limit* pada *cbpolicyd*, penulis melakukan percobaan dengan dua skenario berbeda. Pertama, untuk pengguna *user1*, akan dibuat batasan sebanyak 50 pesan dalam jangka waktu 1800 detik (30 menit). Pengguna tersebut akan diuji untuk dilihat apakah sistem dapat secara efektif mengenali dan membatasi jumlah pesan yang dikirim dalam batas waktu yang telah ditentukan. Kedua, untuk pengguna *user2*, akan ditambahkan aturan yang bernama "*bypass_users_limit*", yang berarti pengguna ini dikecualikan dari aturan-aturan yang ditetapkan, termasuk *rate limit*. Pengujian pada *user2* bertujuan untuk memastikan bahwa mekanisme pengecualian berfungsi dengan baik dan pengguna tersebut dapat mengirim pesan tanpa terkena batasan *rate limit* yang diterapkan.



PolicyD Web Administration			
Quota Limits			
Quota	Rate Limit	Quota	Action
MessageCount	50		no

Gambar 14. Quota Limits

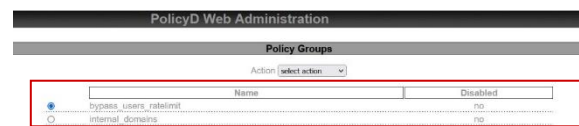
Gambar 14 merupakan aturan batas pengiriman pesan, dengan maksimal yang dapat dikirim yaitu 50 pesan.



PolicyD Web Administration						
Quota List						
Policy	Name	Track	Period	Verdict	Date	Disabled
Rate Limit	Rate Limit	Sender user@domain	1800	REJECT	Store your quota for sending email is full	no

Gambar 15. Quota List

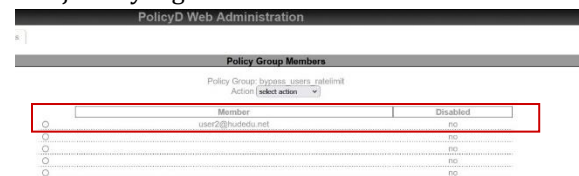
Pada bagian *quota list* terdapat aturan batas waktu pengiriman pesan sebelum di *reject* oleh sistem. Dapat dilihat pada Gambar 15, jika *user* masuk kedalam daftar *rate limit* maka akan dibolehkan mengirim pesan hanya dalam periode 30 menit, jika lebih dari waktu yang telah ditentukan maka sistem akan memberikan notifikasi gagal mengirim pesan atau *reject*.



PolicyD Web Administration		
Policy Groups		
Name	Disabled	
bypass_users_ratelimit	no	
internal_domains	no	

Gambar 16. Policy Group

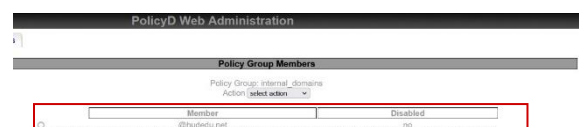
Selanjutnya bagian *policy group* atau kelompok kebijakan, digunakan untuk menerapkan aturan-aturan kebijakan tertentu untuk pengguna. Dalam hal ini terdapat dua kebijakan yaitu *bypass_users_ratelimit* yang dapat menambahkan pengguna yang hendak di kecualikan, sedangkan *internal_domains* digunakan untuk menentukan *domain internal* yang dikecualikan dari aturan kebijakan yang telah dibuat.



PolicyD Web Administration		
Policy Group Members		
Member	Disabled	
user2@hudedu.net	no	

Gambar 17. Daftar User Limit

Pengguna dengan nama *user2* di tambahkan pada daftar grup *bypass_users_ratelimit*.

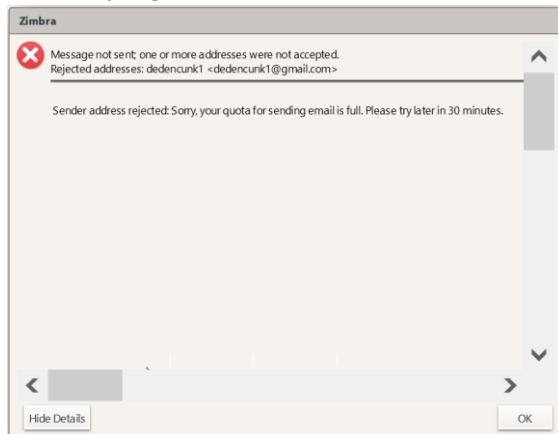


PolicyD Web Administration		
Policy Group Members		
Member	Disabled	
@hudedu.net	no	

Gambar 18. Daftar Internal Domain

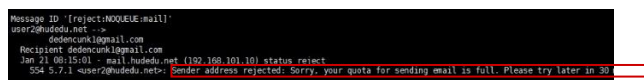
Domain @hudedu.net ditambahkan pada daftar grup *internal_domains*. Setelah semua kebijakan sudah diatur maka akan dilanjutkan dengan

percobaan mengirim pesan sesuai dengan skenario yang telah dibuat.



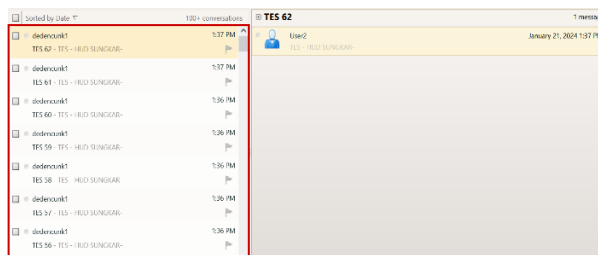
Gambar 19. Notifikasi *Rejected*

Setelah melakukan percobaan mengirim 50 pesan lebih dalam waktu 30 menit. Akan muncul notifikasi pesan gagal terkirim dengan status *reject* dikarenakan kuota sudah penuh seperti Gambar 19, Untuk memastikan dapat dilihat melalui log zimbra.



Gambar 20. Log Pesan Gagal Terkirim

Pengujian selanjutnya dilakukan percobaan kirim pesan lebih dari 50 oleh pengguna yang telah dikecualikan dari aturan kebijakan yaitu *user2*.



Gambar 21. Hasil *Bypass Limit*

Terlihat pada Gambar 21, *user2* berhasil mengirim lebih dari 50 pesan dalam periode waktu 30 menit tanpa ada notifikasi pengiriman gagal atau *rejected*.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan pembahasan yang sudah dilakukan dapat ditarik kesimpulan sebagai berikut:

- 1) Dalam penelitian ini telah berhasil mengintegrasikan server *mailbox* menggunakan metode *load balancing* dengan *zimbra proxy*, serta kebijakan keamanan email menggunakan *cbpolicyd* berhasil di terapkan.
- 2) Implementasi *load balancing* menggunakan *zimbra proxy* dapat meningkatkan kinerja dan skalabilitas sistem secara keseluruhan. Dengan *zimbra proxy* beban kerja *server* dapat terdistribusi dengan merata.
- 3) Peningkatan keamanan menggunakan *cbpolicyd* memudahkan *administrator* untuk mengatur kebijakan pengiriman email. Dengan membatasi jumlah pesan yang dikirimkan dapat membantu dalam mengoptimalkan penggunaan sumber daya *server* email serta mencegah *overload* sehingga meningkatkan kinerja *server*.

Adapun saran dari penelitian ini adalah sebagai berikut:

- 1) Setelah *server* email berjalan dengan baik, lakukan pemantauan pembaharuan versi *zimbra* dan sistem operasi yang digunakan. Hal ini perlu dilakukan agar mendapatkan keamanan optimal dan menjaga ketersediaan berbagai perbaikan performa yang telah diterapkan oleh pengembang.
- 2) Selain itu lakukan pemantauan penggunaan *server* serta potensi ancaman keamanan pada *server* email, sehingga dapat dengan cepat terdeteksi dan melakukan respon terhadap temuan untuk menjaga keamanan sistem secara keseluruhan.

Daftar Pustaka:

- [1] K. Wibowo, I. Fitri, and D. Hidayatullah, "Implementasi Load Balancing Web Server Menggunakan Apache di Ubuntu 16.04.," *SISFOTENIKA*, vol. 10, no. 1, p. 50, Jan. 2020, doi: 10.30700/jst.v10i1.773.
- [2] G. D. Prakoso and H. R. Andrian, "Implementasi Keamanan Server Zimbra Menggunakan SpamAssassin dan Whitelist Pada Linux CentOS 7," 2017.
- [3] S. D. Riskiono and D. Pasha, "ANALISIS METODE LOAD BALANCING DALAM MENINGKATKAN KINERJA WEBSITE E-LEARNING," *J. Teknoinfo*, vol. 14, no. 1, p. 22, Jan. 2020, doi: 10.33365/jti.v14i1.466.
- [4] B. M. & Akmaludin, "Implementasi Zimbra

- Proxy Server Sebagai Load Balancing Dengan Skema Multi Server,” *J. Inf. Syst. Applied, Manag. Account. Res.*, vol. 2, no. 4, pp. 7–13, 2018, [Online]. Available: <http://journal.stmikjayakarta.ac.id/index.php/jisamar/article/view/51>
- [5] H. Mukhtar, Daniel Adi Putra Sitorus, and Yulia Fatma, “Analisa Dan Implementasi Security Mail Server,” *J. Fasilkom*, vol. 10, no. 1, pp. 25–32, 2020, doi: 10.37859/jf.v10i1.1906.
- [6] H. Supendar and Royman, “Zimbra Collaboration Suite Dalam Membangun Web Server Berbasis Linux,” vol. 1, no. 1, pp. 13–26, 2016.
- [7] A. D. Riawati, M. Irfan, Khaeruddin, and A. Faruq, “High Availability Dynamic Sharding Database Server Dengan Metode Fail Over dan Clusterin,” *J. Manaj. Inform. Sist. Inf.*, vol. 5, pp. 1–10, 2022.
- [8] Ley 25.632, “Deduplication Enabled Secure E-mail Server on Cloud Environment using Virtual Data Optimizer,” *IEEE*, pp. 270–275, 2020.
- [9] S. D. Riskiono and D. Pasha, “Analisis Perbandingan Server Load Balancing dengan Haproxy & Nginx dalam Mendukung Kinerja Server E- Learning,” *J. Telekomun. dan Komput.*, vol. 10, no. 3, p. 135, Dec. 2020, doi: 10.22441/incomtech.v10i3.8751.
- [10] K. Marzuki, N. Hanif, and I. P. Hariyadi, “Application of Domain Keys Identified Mail, Sender Policy Framework, Anti-Spam, and Anti-Virus: The Analysis on Mail Servers,” *Int. J. Electron. Commun. Syst.*, vol. 2, no. 2, pp. 65–73, Dec. 2022, doi: 10.24042/ijecs.v2i2.13543.
- [11] A. Tantoni, M. T. A. Zaen, and W. Bagye, “Simulasi Pemilihan Hardware Jaringan Komputer Berdasarkan Penyesuaian Skalabilitas Jaringan (Studi Kasus: STMIK Lombok),” *MISI (Jurnal Manaj. Inform. Sist. Informasi)*, vol. 3, no. 1, pp. 24–34, 2020, [Online]. Available: <https://e-journal.stmiklombok.ac.id/index.php/misi/article/view/133>
- [12] I. T. Singh, T. R. Singh, and T. Sinam, “Server Load Balancing with Round Robin Technique in SDN,” in *2022 International Conference on Decision Aid Sciences and Applications, DASA 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 503–505. doi: 10.1109/DASA54658.2022.9765287.
- [13] F. M. Nurzaman, F. Chahyadi, and M. R. Rathnomi, “Analisis Perbandingan Performa Load Balancer Nginx Dan Haproxy Pada Docker,” *J. Sustain. J. Has. Penelit. dan Ind. Terap.*, vol. 11, no. 01, pp. 16–25, 2022.
- [14] A. Yannuri, M. Iwan Wahyuddin, J. Sawo Manila, P. Minggu, K. Jakarta Selatan, and D. Khusus, “Design and Build Mail Server Systems Using Zimbra 8.8.15 and Antispam on Proxmox Mail Gateway 5.2,” 2020. [Online]. Available: <https://iocscience.org/ejournal/index.php/mantik>
- [15] I. Synacor, “Zimbra Collaboration Administrator Guide,” zimbra.github.io. Accessed: Jan. 21, 2024. [Online]. Available: <https://zimbra.github.io/adminguide/latest/>
- [16] R. Azhar, H. Santoso, and B. Krismono, “Pengaruh Implementasi Kernel Based Virtual Machine Pada Server Vps Terhadap Pemakaian Cpu Memory Dan Harddisk,” *J. Inform. dan Rekayasa Elektron.*, vol. 5, no. 1, pp. 140–152, 2022, doi: 10.36595/jire.v5i1.522.
- [17] Wiki.zimbra, “Zimbra Collaboration Proxy Guide,” Wiki Zimbra. [Online]. Available: https://wiki.zimbra.com/wiki/Zimbra_Proxy_Manual:Overview_And_Planning_For_Zimbra_Proxy
- [18] Wiki.zimbra, “CBPolicyD Management,” Wiki Zimbra. [Online]. Available: https://wiki.zimbra.com/wiki/CBPolicyD_Management
- [19] R. Nurbahri, T. Ahmad, K. Imtihan, B. Wire, and M. Ryansyah, “JIRE (Jurnal Informatika & Rekayasa Elektronika) <http://e-journal.stmiklombok.ac.id/index.php/jire> Volume 3, No 1, April 2020,” *Implementasi Jar. Inter-Vlanrouting Berbas. Mikrotik Rb260Gs Dan Mikrotik Rb1100Ahx4*, vol. 4, no. 1, pp. 13–18, 2020, [Online]. Available: <http://jurnal.pustakagalerimandiri.co.id/index.php/pustakadata/article/view/89>