

## PERANCANGAN WEB SERVER DAN PROXY SERVER DENGAN SISTEM OPERASI JARINGAN DEBIAN MENGGUNAKAN VIRTUAL MACHINE

Aji Syahputra<sup>1</sup>, Mufida Khairani<sup>2</sup>, Yunita Sari Siregar<sup>3</sup>

syah151515@gmail.com<sup>1</sup>, mufida.khairani@gmail.com<sup>2</sup>, yunitasarisiregar1990@gmail.com<sup>3</sup>

<sup>123</sup>Universitas Harapan Medan

### Abstract

A Web Server works by receiving requests from clients to be processed and returned to the client in the form of website pages. In addition, a Web Server is one of the client needs because a Web Server has a large storage capacity and fast access. Broadly speaking, the Web Server has a role in processing various data requested by the client. Proxy Server is a server computer or computer program that acts as a gateway to servers and clients. The Proxy Server is not visible to the client computer, because the Web Server that receives requests from the Proxy Server will interpret these requests as if the request came directly from the client computer, not from the Proxy Server. Having many functions, Proxy Server can be used as filtering because Proxy Server operates at a higher level and has more thorough control over network access. Debian is an operating system composed of software packages released as free software. Debian provides services for managing network processes on network management systems. In addition, servers created by debian do not need to be connected to the internet. In short, Debian is an operating system that has many functions. Debian can be installed on Virtual Machine devices, one of the Virtual Machine devices is Vmware Workstation Pro 16.

**Keyword:** server, debian, virtual machine, operating system.

### Abstrak

Sebuah Web Server bekerja dengan cara menerima permintaan dari klien untuk di proses dan di kembalikan kepada klien dalam bentuk laman website. Selain itu, Web Server menjadi salah satu kebutuhan client sebab Web Server memiliki kapasitas penyimpanan yang besar dan akses yang cepat. Secara garis besar, Web Server memiliki peran dalam memproses berbagai data yang diminta oleh client. Proxy Server adalah sebuah komputer server atau program komputer yang bertindak sebagai gateway terhadap server dan klien. Proxy Server tidak terlihat oleh komputer klien, karena Web Server yang menerima permintaan dari Proxy Server akan menginterpretasikan permintaan-permintaan tersebut seolah-olah permintaan itu datang secara langsung dari komputer klien, bukan dari Proxy Server. Memiliki banyak fungsi, Proxy Server dapat digunakan sebagai penyaring karena memang Proxy Server beroperasi pada level yang lebih tinggi dan memiliki kontrol yang lebih menyeluruh terhadap akses jaringan. Debian adalah sistem operasi yang tersusun dari paket-paket perangkat lunak yang dirilis sebagai perangkat lunak bebas. Debian menyediakan layanan untuk mengatur proses jaringan pada sistem pengatur jaringan. Selain itu, server yang dibuat oleh debian tidak perlu terhubung ke internet. Secara singkatnya, debian adalah salah satu sistem operasi yang memiliki banyak fungsi. Debian dapat di install pada perangkat Mesin Virtual, salah satu perangkat Mesin Virtual adalah Vmware Workstation Pro 16.

**Kata Kunci:** server, debian, mesin virtual, sistem operasi.

### 1. PENDAHULUAN

Dengan perkembangan bidang teknologi yang semakin pesat, internet menjadi sarana yang paling banyak digunakan oleh seluruh orang. Internet merupakan sumber ilmu pengetahuan di era saat ini. Dimana sebuah internet sangat berpengaruh meningkatkan pengetahuan dan wawasan yang dapat digunakan sarana untuk kegiatan berfikir, belajar maupun berkomunikasi. Internet dapat menghubungkan hampir setiap orang di dunia ini, dengan adanya internet, seseorang bisa berkomunikasi saat itu juga

dalam waktu singkat meskipun terpisah jarak yang jauh. Internet juga mengandung beragam pengetahuan dan informasi yang tidak ada habisnya.

Server merupakan suatu sistem yang memiliki layanan khusus berupa penyimpanan data. Pada dasarnya, Data yang disimpan didalam server berupa informasi dan beragam jenis dokumen yang kompleks. Dalam perkembangan teknologi saat ini, server umumnya digunakan oleh berbagai jenis aktivitas yang berhubungan dengan internet. Server sangat berperan penting dalam menyediakan layanan akses lebih cepat

untuk mengirim atau menerima data maupun informasi yang tersedia pada server. Beberapa jenis server antara lain *web server*, *dns server*, *mail server*, *ftp server*, *dhcp server* dan *proxy server*.

Keseluruhan jenis server ini bisa di konfigurasi atau di buat dengan sistem operasi jaringan linux yaitu debian. Perancangan *web server* dan *proxy server* ini dapat digunakan sebagai media pengujian bahwasanya *web server* dan *proxy server* dapat dimanfaatkan dengan baik. Dengan adanya *web server*, client dapat mengakses permintaan *Hypertext Transfer Protocol Secure (HTTPS)* kepada server dalam bentuk website. Sedangkan kontrol akses juga sangat di perlukan agar tidak terjadi penyalahgunaan internet oleh user. Dengan adanya *Bloking Website* atau *Web Filtering*, situs yang berbahaya dapat dapat di blokir agar tidak di akses oleh user. Maka dari itu peran *Proxy server* sangat penting karena memiliki kegunaan yang besar bagi user.

Dengan adanya teknologi *virtual machine*, debian dapat di install tanpa harus melakukan instalasi pada komputer/laptop. Kemudian langsung melakukan konfigurasi *web server* dan *proxy server*. *Web server* dan *proxy server* tidak lepas dari pengelolahan dalam jaringan, kedua jenis *server* ini memiliki peran penting bagi para *client*. Selain itu dengan adanya server yang dibuat melalui debian lebih optimal dalam waktu efisien dan biaya. Salah satu keunggulan pada debian adalah, debian tidak membutuhkan spesifikasi *hardware* yang tinggi.

Berdasarkan latar belakang di atas, yang membahas tentang *web server* dan *proxy server* yang diinstall menggunakan *virtual machine* dan sistem operasi linux yaitu debian, maka penulis menyimpulkan untuk membahas penelitian dengan judul mengenai “Perancangan *Web Server* dan *Proxy Server* Dengan Debian Menggunakan *Virtual Machine*”.

## 2. LANDASAN TEORI

### A. Website

Era modern mempermudah manusia dalam melakukan banyak hal. Penemuan internet yang revolusioner mampu menghubungkan berbagai sektor secara *daring*. Kini, berbagai informasi tersedia dan dapat diakses menggunakan internet melalui *website* atau situs *web*. Internet merupakan jaringan komunikasi yang menghubungkan satu media elektronik dengan media lainnya. Dengan adanya internet, berbagai macam situs atau media sosial dapat diakses dengan mudah, salah satunya *Website*. *web* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia, dan lainnya pada jaringan internet[1].

Terkadang istilah *server* disebut sebagai *web server*. Namun orang lebih suka menyebutnya sebagai *server* saja. Sebuah *server* didukung dengan prosesor yang bersifat scalabel dan RAM yang besar, juga

dilengkapi dengan sistem operasi khusus. Sistem operasi ini berbeda dengan sistem operasi yang biasanya. Jika kita biasa menggunakan sistem operasi seperti windows, macOS, maka sistem operasi dari *server* ini berbeda[1].

*Server* sendiri terdiri atas beberapa jenis untuk memenuhi kebutuhan banyak *client*, maka dari itu. Berikut ini adalah beberapa Jenis-jenis *server* antara lain :

1. *Server Platform*: *Server* platform adalah hardware atau software dasar untuk sistem yang bertindak sebagai mesin yang mengerakkan *server*.
2. *Server Aplikasi*: Juga dikenal sebagai middleware, berfungsi sebagai pengolah perintah dari *client* dan database.
3. *Server Audio / Video*: Menyediakan kemampuan multimedia ke situs *web* dengan membantu pengguna untuk menyiaran streaming konten multimedia.
4. *Chat Server*: Untuk menangani chatting beberapa user.
5. *Fax Server*: *Fax Server* memungkinkan user untuk mengirim dan menerima fax melalui jaringan komputer.
6. *FTP Server*: File Transfer Protocol (FTP) untuk transfer data, dari atau ke *client*.
7. *Groupware Server*: Ini adalah software yang dirancang agar memungkinkan pengguna untuk bekerja sama, terlepas dari lokasi, melalui internet atau intranet perusahaan dan berfungsi bersama-sama dalam suasana virtual.
8. *IRC Server*: Internet Relay Chat Server (IRC Server) memungkinkan user untuk berdiskusi lewat jaringan.
9. *List Server*: Menyediakan cara yang lebih baik dalam mengelola milis. *Server* dapat berupa diskusi interaktif terbuka untuk orang-orang atau daftar satu arah yang memberikan pengumuman, buletin atau iklan.
10. *Mail Server*: *Server* yang bertugas untuk menangani email.
11. *News Server*: Ini berfungsi sebagai sumber distribusi dan pengiriman untuk banyak kelompok berita umum, didekati melalui jaringan berita USENET.
12. *ProxyServer*: *Proxy server* bertindak sebagai mediator antara *client* dan *server* diluar jaringan untuk menyaring atau membatasi permintaan, mengingkatkan kinerja koneksi dan sharing.
13. *Telnet Server*: Untuk Log On ke komputer host dengan cara remote dari komputer lain
14. *Virtual Server*: *Virtual Server* adalah metode mengurangi jumlah *Server* fisik, jadi pada satu *server* fisik seolah-olah ada beberapa *Server*.

15. *Web server*: Menyediakan konten ke *web browser*. *Web server* Berkommunikasi dengan menggunakan *HyperText Transfer Protocol (HTTP)*.

### B. Web Server

Pada umumnya *web server* berperan sebagai *server* yang memberikan layanan kepada komponen yang meminta informasi berkaitan dengan *web*, dalam *web* yang telah dirancang dalam internet. *Web server* adalah *software* yang menjadi tulang belakang dari *world wide web (WWW)* yang pertama kali tercipta sekitar tahun 1980an. *Web server* menunggu permintaan dari *client* yang menggunakan *Browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla Firefox*, dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*[2].

### C. Proxy Server

*Proxy server* bekerja dengan menjembatani komputer ke Internet. Program Internet seperti *browser*, *download manager* dan lain-lain berhubungan dengan *proxy server*, dan *proxy server* tersebut yang akan berkomunikasi dengan *server* lain di Internet. Pada umumnya *proxy server* digunakan untuk mengamankan jaringan komputer pribadi yang terhubung dengan jaringan publik. Jadi, dari *proxy server* tersebut maka biasanya *server* diletakkan di antara aplikasi *server* dengan aplikasi *client*, dimana aplikasi *client* berupa *web browser*, *client FTP* dan lainnya sedangkan aplikasi *server* berupa *server FTP* dan *web server*[3].

*Proxy server* memiliki lebih banyak fungsi daripada *router* yang memiliki fitur *packet filtering* karena memang *proxy server* beroperasi pada level yang lebih tinggi dan memiliki kontrol yang lebih menyeluruh terhadap akses jaringan. *Proxy server* yang berfungsi sebagai sebuah "agen keamanan" untuk sebuah jaringan pribadi, umumnya dikenal sebagai *firewall*. Sebuah *proxy server* memiliki fungsi-fungsi tambahan dibandingkan dengan *server* biasa dalam sebuah jaringan. Diantara fungsi-fungsi tersebut adalah *connection sharing*, *filtering*, dan *caching*[4].

### D. Pemodelan Berorientasi Objek Menggunakan Unified Modeling Language (UML)

UML adalah sebuah metode pemodelan yang digunakan untuk memvisualisasikan sebuah perancangan sistem berorientasi objek. Selain itu UML juga telah menjadi standar visualisasi, perancangan, pendokumentasian sistem, bahkan saat ini menjadi standar dalam penyusunan sebuah *blue print*[5].

## 3. METODE PENELITIAN

Adapun metode penelitian yang penulis gunakan adalah sebagai berikut.

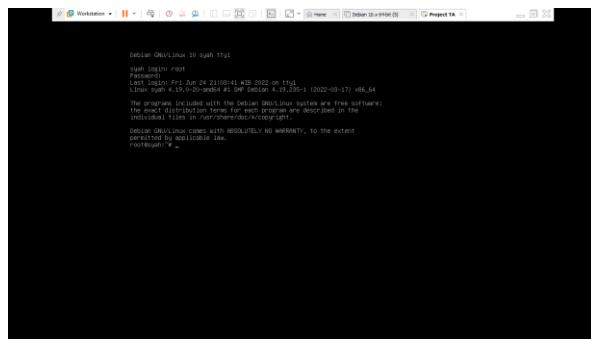
1. Studi Literatur  
Pada tahap ini penulis mengumpulkan bahan dan data sebagai referensi dari berbagai jurnal, skripsi dan sumber lainnya yang berkaitan dengan penulisan tugas akhir. Metode ini dilakukan untuk memperoleh informasi yang terkait dengan *web server* debian menggunakan *virtual machine*.
2. Analisis  
Pada tahap ini penulis mengolah data dari hasil studi literatur yang kemudian dilakukan analisa terhadap cara kerja *web server* dan *proxy server* dengan debian menggunakan *virtual machine*.
3. Perancangan Sistem  
Pada tahap ini penulis melakukan perancangan sistem yang meliputi konfigurasi ip, konfigurasi *web server*, konfigurasi *proxy server* dan koneksi antara *client* dengan *server*.
4. Uji Coba dan Implementasi  
Pada tahap ini penulis melakukan uji coba sistem kepada, uji coba ini dilakukan oleh *server* dan *client*, guna mengetahui apakah *web server* dan *proxy server* agar dapat berjalan dengan baik saat diakses oleh *client*.
5. Dokumentasi  
Tahap ini berisi mengenai laporan dan kesimpulan akhir dari penelitian dan pengujian dalam bentuk skripsi.

## 4. HASIL DAN PEMBAHASAN

Pada bagian ini akan dilakukan pembahasan hasil terhadap rancangan yang dibuat dalam penelitian ini, mulai dari mengaktifkan *server* hingga dapat terhubung serta dapat diakses oleh seluruh *client*, berikut akan diuraikan.

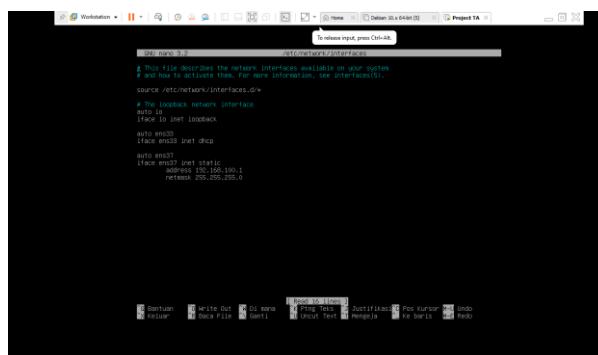
### A. Perancangan Web Server Dan Proxy Server

*Server* berperan penting dalam menyediakan layanan akses lebih cepat untuk mengirim atau menerima data maupun informasi yang tersedia pada *server*. Data yang disimpan melalui *server* berupa informasi dan beragam jenis dokumen yang kompleks. Layanan tersebut ditujukan khusus untuk *client* yang berkebutuhan dalam menyediakan informasi untuk pengguna atau pengunjungnya. Berikut gambar 1 adalah tampilan *server* debian yang terinstal pada *Virtual Machine*.



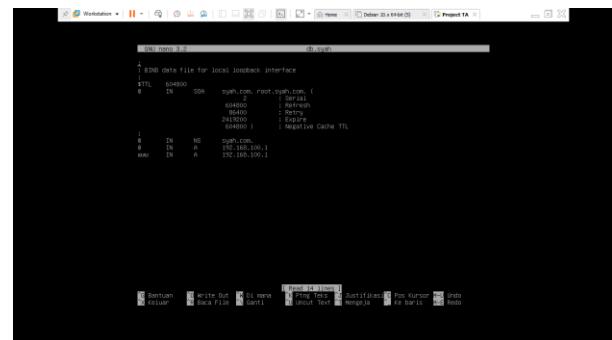
Gambar 1. Tampilan Interface Debian

Setiap komputer yang ingin terhubung kedalam jaringan, membutuhkan sebuah IP. IP sendiri ialah alamat unik yang diberikan pada komputer sehingga dapat terhubung kedalam jaringan. Pada saat membuat *server*, yang terlebih dahulu di lakukan adalah melakukan konfigurasi *Interface*, pengaturan antar muka ini berfungsi agar server mendapatkan ip dari internet secara dhcp, selain itu ip juga dimasukan secara manual, dan saat melakukan pengaturan pada cleint, ip harus disamakan dengan ip yang ada pada server. Berikut adalah tampilan *Interface* pada *server* dapat di lihat pada gambar 2.



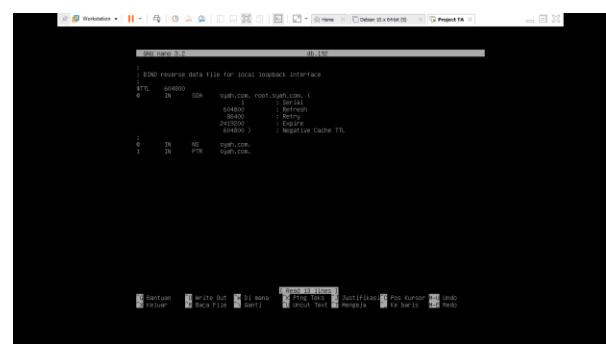
Gambar 2. Konfigurasi IP Debian

*Domain Name System (DNS)* adalah sistem yang menghubungkan *Uniform Resource Locator (URL)* dengan *Internet Protocol Address (IP)*. Dengan menggunakan *Domain Name System (DNS)*, *Domain Name System (DNS)* memiliki fungsi untuk mencocokan nama atau domain situs web kedalam angka-angka yang disebut dengan IP. Sehingga server mengerti siatus apa yang di minta oleh *client*. *Client* tidak perlu memasukan IP secara langsung karena *Domain Name System (DNS)* mengubah *Internet Protocol (IP)* menjadi inisial yang memudahkan *Client*. Berikut adalah tampilan konfigurasi *Domain Name System (DNS)* pada gambar 3.



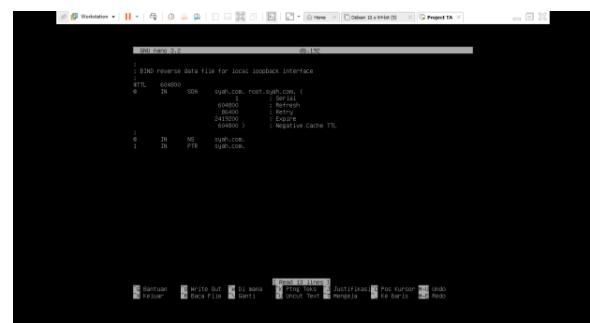
Gambar 3. Konfigurasi "db.syah"

Pada gambar 3 dilakukan konfigurasi untuk mengganti "localhost" menjadi doamian yang ingin digunakan dan setelah itu masukan alamat IP yang digunakan.



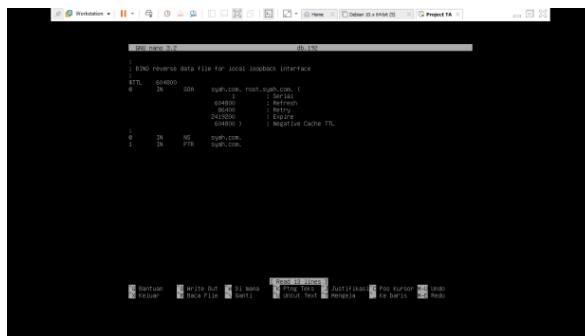
Gambar 4. Tampilan Konfigurasi "db.192"

Hal yang sama juga di lakukan pada gambar 4, mengubah "localhost" menjadi domain yang akan digunakan.



Gambar 5. Tampilan Konfiguras "named.conf.default-zones"

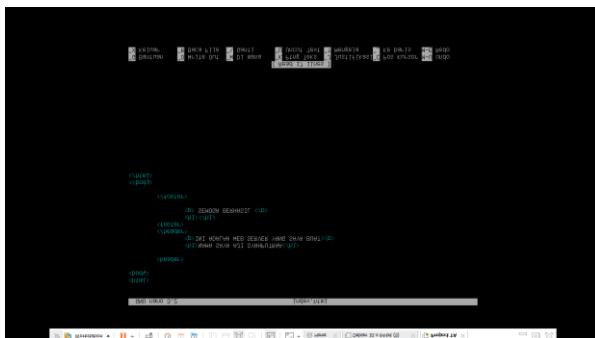
Lakukan konfigurasi pada debian dengan perintah nano /etc/bind/named.conf.default-zones" lalu isi dengan domain dan alamat ip yang ditulis secara terbalik seperti pada gambar 5.



Gambar 6. Tampilan Konfigurasi “resolv.conf”

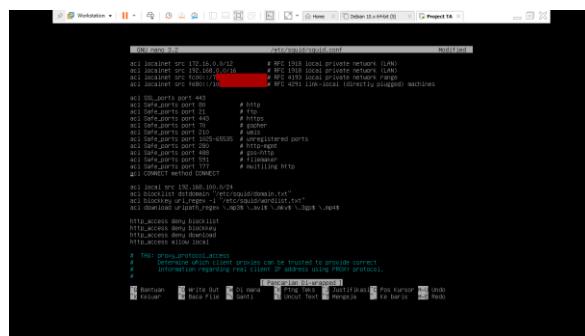
Pada gambar 6, ketikan perintah resolv.conf, kemudian pada nameservers, masukan alamat ip domain yang telah dibuat pada server saat melakukan konfigurasi di awal. Fungsi gunanya agar web server dapat diakses ketika memasukan *Domain Name System*(DNS).

Selanjutnya adalah melakukan konfigurasi langsung pada *Web Server* serta mengubah isi dari *Web Server* tersebut seperti yang di inginkan dan agar dapat di akses seperti yang terlihat pada gambar 7.



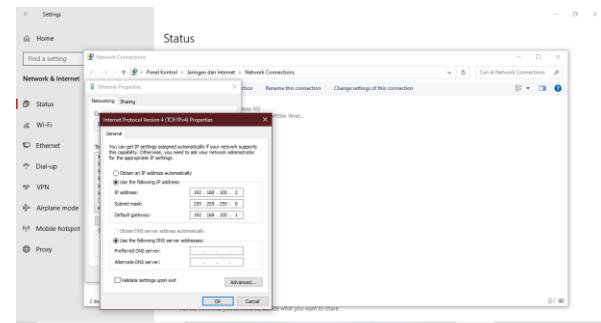
Gambar 7. Tampilan Konfigurasi “html.index”

Setelah mengubah isi dari *Web Server*, Selanjutnya melakukan konfigurasi *Proxy Server* seperti yang terlihat pada gambar 8.

Gambar 8. Tampilan Konfigurasi *Proxy Server*

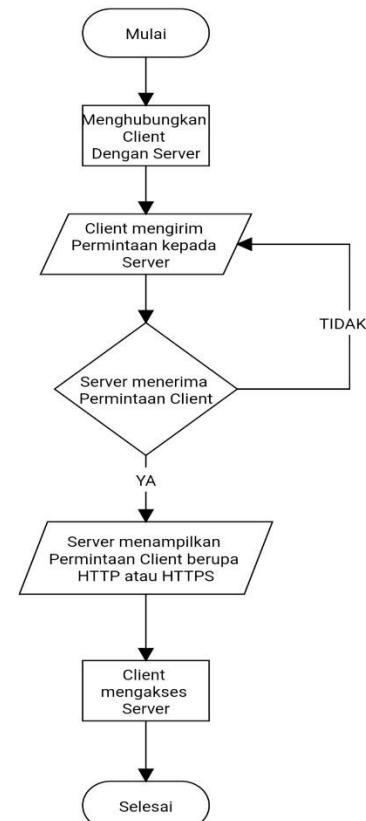
Ketika seluruh konfigurasi pada *server* sudah di lakukan, langkah penyelesaiannya adalah menghubungkan antar *server* dengan *client*. *Server*

terhubung dengan *client* menggunakan kabel UTP *Cross-OVER*. Berikut gambar 9 adalah tampilan Konfigurasi IP pada *client*.

Gambar 9. Tampilan Konfigurasi IP pada *Client*

## B. Flowchart Web Server

Dalam membuat *web server*, dibutuhkan sebuah flowchart untuk menggambarkan alur dari langkah-langkah pada *web server*. Berikut adalah *flowchart web server* yang dapat di lihat pada gambar 10

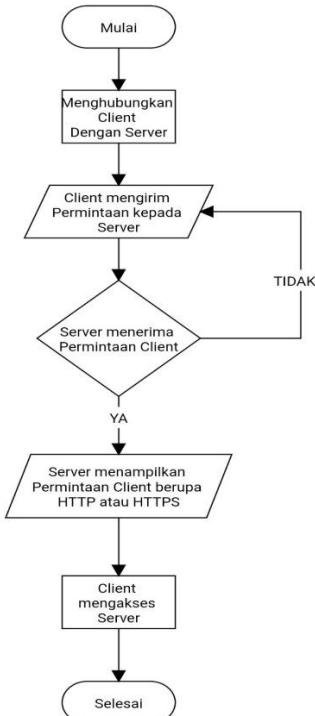
Gambar 10. Flowchart *Web Server*

Pada gambar 10 dijelaskan alur dari *web server*, di mana tahap awal adalah menghubungkan antara *client* dan *server*, setelah *client* terhubung dengan *server*, *client* mengakses *web server* berupa laman web yang di akses pada *browser client*, saat *server* berhasil

menerima permintaan *client*, *server* akan menampilkan laman *web* berupa HTML atau CSS di *browser client*. Tetapi jika tidak berhasil, langkah yang harus di lakukan adalah mengulangi permintaan pada *server*.

### C. Flowchart Proxy Server

Dalam membuat Proxy server, terdapat langkah-langkah yang digambarkan dalam bentuk simbol atau diagram yang memiliki Alur. Berikut adalah *flowchart proxy server* yang di lihat pada gambar 11.



Gambar 11. *Flowchart Proxy Server*

Pada gambar 11 dijelaskan alur dari penggunaan proxy yang di lakukan oleh *server* kepada *client*, jadi ketika client mengakses sesuatu yang telah di *filter* oleh *server*, otomatis laman yang di minta tidak akan tertampilkan pada *browser client*.

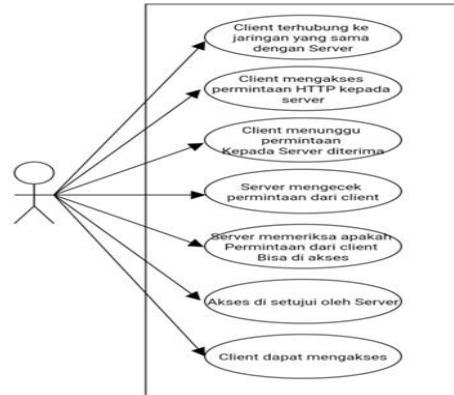
### D. Unified Modeling Language(UML)

*Unified Modelling Language* (UML) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997. UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan *blueprint* sebuah *software*.

#### 1. Use Case Diagram

*Use case diagram* adalah satu dari berbagai jenis diagram UML (*Unified Modelling Language*) yang

menggambarkan hubungan interaksi antara sistem dan aktor. *Use Case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. *Use case diagram* mengenai web server dan proxy server dapat di lihat pada gambar 12 berikut.

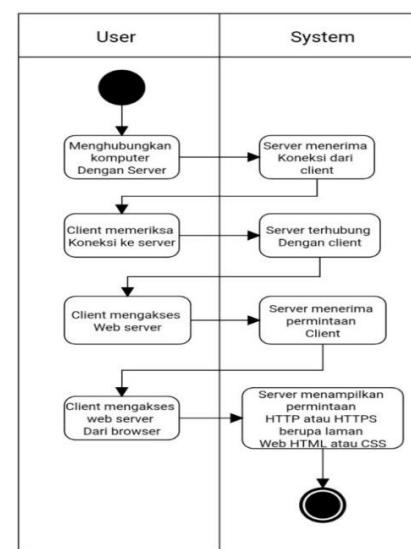


Gambar 12. *Use Case Diagram Web server & Proxy*

Pada gambar 12 dijelaskan mengenai alur *use case diagram*, yang pertama yaitu *client* terhubung ke jaringan yang sama dengan *server*, kemudian *client* meminta akses kepada *server*, setelah itu beberapa saat *client* menunggu permintaan disetujui oleh *server*, *server* kemudian memeriksa permintaan dari *client* apakah situs yang di akses oleh *client* tidak berbahaya, ketika permintaan di setujui oleh *server*, maka *client* dapat mengakses permintaan HTTPS atau HTTP berupa laman *server*.

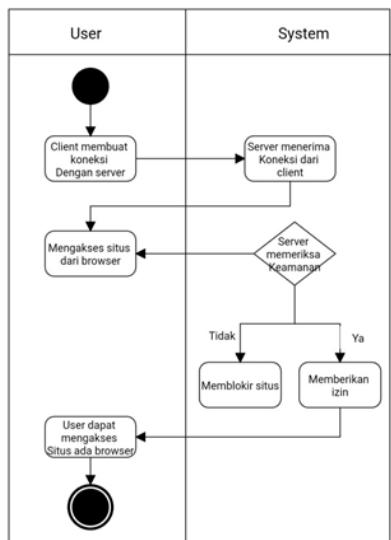
### 2. Activity Diagram

pada gambar 13 dan gambar 14 akan di jelaskan alur *Activity Diagram* mengenai *Web Server* dan *Proxy server*.



Gambar 13. *Activity Diagram web server*

Pada gambar 13 di jelaskan bagaimana alur dari *web server* yang di akses oleh *client* dalam bentuk *Activity Diagram*.



Gambar 14. Activity Diagram Proxy Server

Pada Gambar 14 dijelaskan bagaimana cara kerja *proxy* sebagai *filtering* untuk user/clinet dan bentuk *Diagram Activity*.

#### E. Proses Pengujian Web server dan Proxy Server

Pengujian adalah proses menjalankan dan mengevaluasi sebuah perangkat lunak secara manual maupun otomatis untuk menguji apakah sistem yang di rancang sudah memenuhi persyaratan atau belum, atau untuk menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya. Setelah menyelesaikan konfigurasi *server* dan melakukan koneksi kepada *client*, maka akan di uji coba apakah rancangan *web server* dan *proxy server* dapat berjalan dengan baik.

##### 1. Pengujian Koneksi Antara Server dan Internet

Berikut adalah proses pengujian koneksi antara *server* dan internet, *server* akan mencoba melakukan ping ke google.com.

```

$ ping google.com
PING google.com (209.85.128.139) 56(84) bytes of data.
64 bytes from de-in-f139.149.128.139: icmp_seq=1 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=2 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=3 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=4 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=5 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=6 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=7 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=8 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=9 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=10 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=11 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=12 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=13 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=14 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=15 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=16 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=17 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=18 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=19 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=20 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=21 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=22 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=23 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=24 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=25 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=26 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=27 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=28 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=29 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=30 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=31 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=32 ttl=128 time=0.96 ms
64 bytes from de-in-f139.149.128.139: icmp_seq=33 ttl=128 time=0.96 ms

```

Gambar 15. Uji Coba Koneksi google.com

Berdasarkan proses pengujian koneksi di atas dapat disimpulkan bahwa koneksi dengan google.com dapat berjalan dengan baik seperti pada gambar 15.

##### 2. Pengujian Koneksi Antara Server dan Client

Selanjutnya kan di lakukan proses pengujian koneksi. Berikut proses pengujian koneksi antara *server* dan *client*, *server* akan melakukan ping ke IP 192.168.100.2 yang merupakan *IP Client*.

```

$ ping 192.168.100.2
PING 192.168.100.2 (192.168.100.2) 64(84) bytes of data.
64 bytes from 192.168.100.2: icmp_seq=1 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=2 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=3 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=4 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=5 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=6 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=7 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=8 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=9 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=10 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=11 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=12 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=13 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=14 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=15 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=16 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=17 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=18 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=19 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=20 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=21 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=22 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=23 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=24 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=25 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=26 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=27 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=28 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=29 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=30 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=31 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=32 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=33 ttl=128 time=0.96 ms

```

Gambar 16. Uji Coba Koneksi antara server dan client

Berdasarkan proses pengujian koneksi di atas dapat disimpulkan bahwa koneksi antara *server* dan *client* dapat berjalan dengan baik seperti pada gambar 16.

##### 3. Pengujian Domain Name System (DNS) pada Server

Berikut adalah proses melakukan uji coba *Domain Name System* (DNS) pada *server*. Server melakukan ping kepada domain syah.com.

```

$ ping syah.com
PING syah.com (192.168.100.2) 64(84) bytes of data.
64 bytes from 192.168.100.2: icmp_seq=1 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=2 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=3 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=4 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=5 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=6 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=7 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=8 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=9 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=10 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=11 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=12 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=13 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=14 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=15 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=16 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=17 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=18 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=19 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=20 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=21 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=22 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=23 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=24 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=25 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=26 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=27 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=28 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=29 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=30 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=31 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=32 ttl=128 time=0.96 ms
64 bytes from 192.168.100.2: icmp_seq=33 ttl=128 time=0.96 ms

```

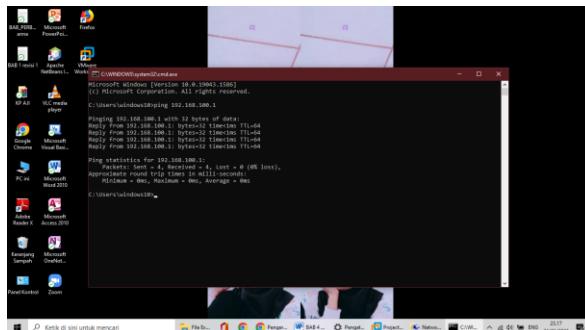
Gambar 17. Uji Coba Domain Name System (DNS) pada server

Berdasarkan proses pengujian koneksi di atas dapat disimpulkan bahwa koneksi dengan syah.com dapat berjalan dengan baik seperti pada gambar 17.

##### 4. Pengujian Koneksi Antara Client dan Server

Berikut proses pengujian koneksi antara *server* dan *client*, *server* akan melakukan ping ke IP 192.168.100.1 yang merupakan *IP Client*. Ping

merupakan perintah yang biasa digunakan untuk melakukan pengecekan koneksi pada sebuah komputer yang saling terhubung satu dengan yang lainnya. Cara kerja ping dengan membuat perangkat client atau server mengirim paket ICMP\_ECHO dan beberapa data lainnya dalam satu host. Kemudian host tujuan akan memberikan balasan yang sama.

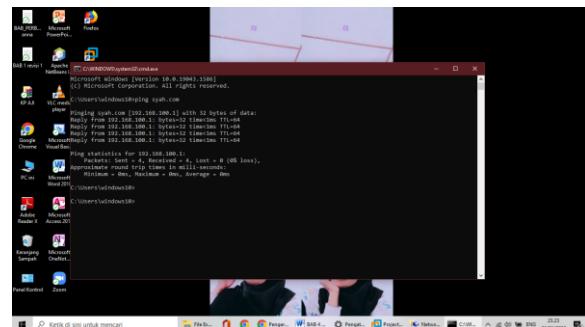


Gambar 18. Uji Coba Antara *Client* dan *Server*

Berdasarkan proses pengujian koneksi di atas dapat disimpulkan bahwa koneksi antara *server* dan *client* dapat berjalan dengan baik seperti pada gambar 18.

##### 5. Pengujian Domain Name System (DNS) pada Client

Berikut adalah proses melakukan uji coba *Domain Name System* (DNS) pada *client*. *Client* melakukan ping kepada domain syah.com. Pengujian ini dilakukan setelah seluruh konfigurasi sudah selesai, untuk melakukan pengujian yang pertama kali harus dilakukan adalah melakukan restart pada bing9, fungsi ini agar komputer atau server memulai ulang seluruh service dan dapat menjalankan dns dengan baik saat dilakukan pengujian oleh client.

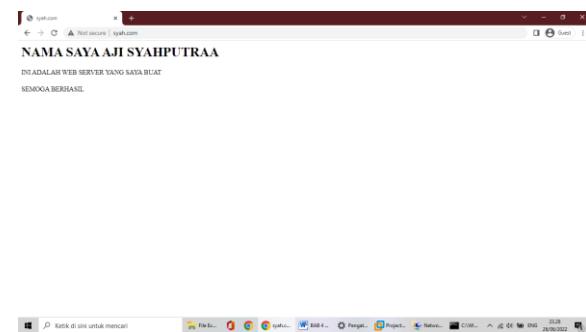


Gambar 19. Uji Coba Domain Name System (DNS) pada *Client*

Berdasarkan proses pengujian koneksi di atas dapat disimpulkan bahwa koneksi dengan syah.com dapat berjalan dengan baik seperti pada gambar 19.

##### 6. Pengujian Web server pada Browser Client

Berikut merupakan tahap untuk menguji *web server* pada *client*, *Web server* menunggu permintaan atau *request* dari *client*. Permintaan tersebut berupa HTTP atau HTTPS.

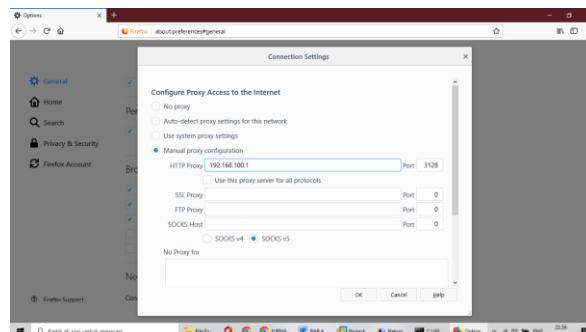


Gambar 20. Uji Coba Web server pada *Browser Client*

Berdasarkan proses pengujian koneksi pada gambar 20 di atas dapat disimpulkan bahwa *web server* dengan *domain* syah.com dapat berjalan dengan baik di *browser client*

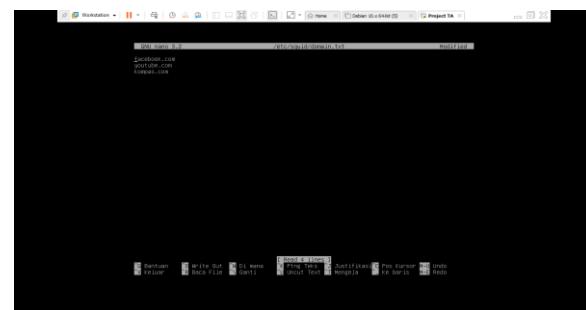
##### 7. Pengujian Proxy Server pada Client

Sebelum melakukan uji coba *proxy server* pada *client*, atur terlebih dahulu *proxy* pada *browser* dengan memasukan ip server debian seperti gambar 21.



Gambar 21. Mengatur IP *Proxy* pada *browser*

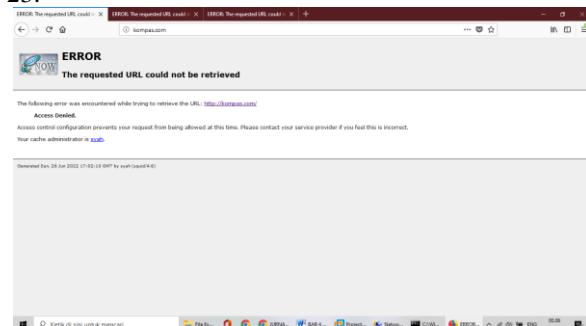
Setelah itu masukan IP *server* pada *proxy Browser*, kemudian lakukan uji coba proxy pada *client* seperti pada gambar 22.



Gambar 22. Daftar Website yang di bloking oleh *Proxy Server*

8. Uji Coba Mengakses Kompas.com

Pada situs kompas.com di lakukan uji coba langsung menggunakan browser *client* seperti gambar 23.

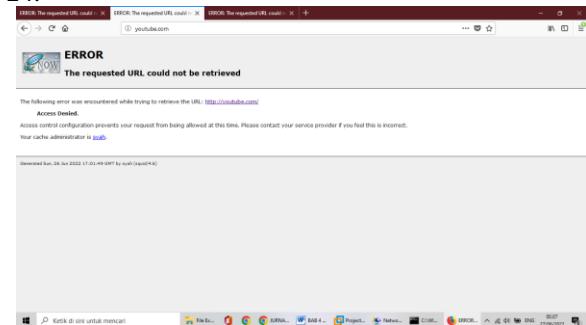


**Gambar 23.** Uji Coba mengakses web Kompas.com

Pada gambar 23, pengujian bloking pada situs kompas.com berhasil di lakukan

9. Uji Coba Mengakses Youtube.com

Setelah berhasil melakukan uji coba bloking situs sebelumnya, kemudian di lanjutkan dengan melakukan uji coba pada situs Youtube.com seperti pada gambar 24.

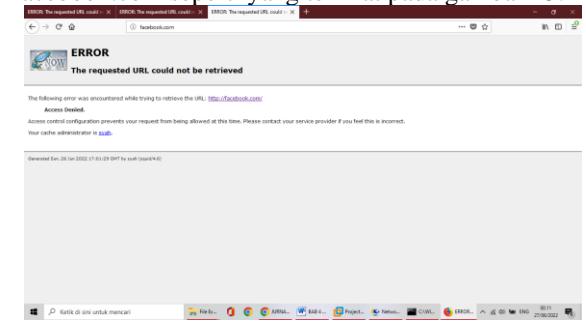


**Gambar 24.** Uji coba mengakses Youtube.com

Pada gambar 24, pengujian bloking pada situs Youtube.com berhasil di lakukan

10. Uji Coba mengakses Facebook.com

Dan yang terakhir dilakukan uji coba pada situs Facebook.com seperti yang terlihat pada gambar 25.



**Gambar 25.** Uji Coba Mengakses Facebook.com

Berdasarkan proses pengujian koneksi di atas dapat disimpulkan bahwa *proxy server* dapat berjalan dengan baik di browser *client*.

11. Pengujian Sistem Menggunakan Blackbox

Pada tahap ini akan di lakukan pengujian sistem untuk memeriksa apakah suatu perangkat lunak yang di hasilkan sudah dapat di jalankan sesuai dengan standar tertentu. Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada perangkat lunak yang di uji. Ada pun teknik pengujian yang di lakukan yaitu pengujian *Blackbox*. *Blackbox* adalah suatu metode pengujian pada fungsionalitas atau kegunaan dari suatu software. Blackbox hanya akan menjangkau input dan output sistem software tanpa adanya pengetahuan terkait internal program. Pengujian ini berfokus pada persyaratan fungsional sistem. Pengujian berikut di lakukan guna memeriksa tingkat keakuratan sistem.

Hasil pengujian web server dengan menggunakan sistem pengujian Blackbox dapat dilihat pada tabel 1 berikut.

**Tabel 1.** Pengujian Sistem *Blackbox* pada *Web Server*

Aktivitas Pengujian	Realisasi yang di harapkan	Hasil pengujian
Melakukan koneksi IP dhcp dari internet	Server mengambil IP dari internet secara DHCP	Berhasil
Melakukan uji coba koneksi dengan internet dan berhasil melakukan ping ke google.com	Server terhubung dengan internet dan berhasil melakukan ping ke google.com	Berhasil
Melakukan ping pada ip Server : 192.168.10.1	Koneksi berjalan dengan baik	Berhasil
Melakukan uji coba koneksi pada client	Koneksi pada client berhasil di lakukan	Berhasil
Melakukan uji coba client kepada server	Koneksi pada server berhasil di lakukan	Berhasil

Melakukan uji coba DNS 'syah.com' pada server dengan	DNS dapat terhubung dengan baik	Berhasil
Melakukan uji coba DNS 'syah.com' pada Client melalui web browser	DNS dapat di akses oleh client	Berhasil

Hasil pengujian *Proxy Server* dengan menggunakan sistem pengujian *Blackbox* dapat di lihat pada tabel 2 berikut.

**Tabel 2.** Pengujian Sistem *Blackbox* pada *Proxy Server*

Aktivitas Pengujian	Realisasi yang di harapkan	Hasil pengujian
Mengakses youtube.com pada browser Client	Youtube.com terblokir oleh proxy pada browser Client	Berhasil
Mengakses Kompas.com pada browser Client	Kompas.com terblokir oleh proxy pada browser Client	Berhasil
Mengakses komiku.id pada browser Client	Komiku.id terblokir oleh proxy pada browser client	Berhasil

## 5. KESIMPULAN

Dari perancangan *web server* dan proxy server dengan *debian* menggunakan *virtual machine* dapat di simpulkan sebagai berikut ini.

1. Dengan melakukan perancangan menggunakan *virtual machine*, mendapatkan manfaat dari kinerja yang tinggi tanpa mengeluarkan biaya yang biasa dikeluarkan menggunakan *hardware* sendiri
2. Agar bisa terhubung dengan *server*, *client* tidak bisa begitu saja langsung terhubung, karena *client* harus melakukan pengaturan IP dengan *class* yang sama oleh *Server*.
3. *Web server* yang di ciptakan hanya dapat di akses oleh *client* yang terhubung dengan *server* secara langsung
4. Untuk mengoptimalkan penerapan *proxy server* kepada *client*, *browser client* harus di atur sedemikian rupa dengan cara memasukkan IP *server* pada *proxy browser* tersebut.

5. Konfigurasi pada *server* harus di lakukan secara hati-hati, karena jika ada sedikit kesalahan, akan terjadi *error* pada *server*.

## 6. DAFTAR PUSTAKA

- [1]. Arunawati Putri, Anindya. *Optimasi Apache Web Server Menggunakan Varnish Web Cache Dan Reverse Proxy Nginx*. 2020.
- [2]. Chandra, Albert Yakobus. "Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request." *Jurnal Sistem Dan Informatika (JSI)*, vol. 14, no. 1, 2019, pp. 48–56, doi:10.30864/jsi.v14i1.248.
- [3]. Dwiyatno, Saleh, and Wiwin Susilawati. "Membangun Web Server Menggunakan Debian Server Sebagai Pendukung Media Pembelajaran Di Sma Negeri 1 Baros." *JSII (Jurnal Sistem Informatika)*, vol. 3, 2017, pp. 7–14, doi:10.30656/jsii.v3i0.127.
- [4]. Hermawan, Y., et al. "Perancangan Aplikasi Client-Server Untuk Sistem Informasi Inventori Studi Kasus Di Toko Bangunan Santoso." *Jurnal Infra*, 2015.
- [5]. Ridwan, A. "Analisis Perbandingan Performa Apache Web Server Dan Nginx Menggunakan Apache Jmeter." *Jurnal Teknologi ITP*, vol. 8, no. 2, 2020, pp. 87–92, doi:10.21063/jtif.2020.v8.2.87-92.
- [6]. Soepomo, Prof. "Analisis Dan Perancangan Proxy Server Menggunakan Virtual Machine." *JSTIE (Jurnal Sarjana Teknik Informatika) (E-Journal)*, vol. 2, no. 3, 2014, pp. 1–9.
- [7]. Tedyyana, Agus, and Rezki Kurniati. "Membuat Web Server Menggunakan Dinamic Domain Name System Pada IP Dinamis." *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, vol. 7, no. 1, 2016, pp. 1–10.
- [8]. Tegal, Y. P. T. *Rancang Bangun Web Server Menggunakan Linux Debian 5 Pada Smk Ypt Tegal*. no. 9, 2017, pp. 77–79.
- [9]. Yuisar, et al. "Analisa Pemanfaatan Proxy Server Sebagai Media Filtering Dan Caching Pada Jaringan Komputer." *Jurnal Media Infotama*, vol. 11, no. 1, 2015, pp. 81–90.