

| | |
|--------------------|---|
| Author | : Muhamad Almupid |
| Judul | : Dokumentasi Konfigurasi Apache2 |
| Tempat dan tanggal | : Kantor MV Site PIK – 28 Juli 2025 |
| Refrensi | : https://httpd.apache.org/docs/2.4/configuring.html |

A. Pendahuluan

Dalam dunia web, **Apache HTTP Server (Apache2)** adalah salah satu server web paling dominan dan fundamental. Dikenal karena **stabilitas, kinerja, dan fleksibilitasnya**, Apache2 menjadi pilihan utama bagi jutaan situs web, dari skala kecil hingga besar. Sebagai perangkat lunak *open-source* yang terus berkembang, Apache2 memainkan peran krusial dalam menghubungkan pengguna dengan informasi di internet, menjadikannya pilar tak tergantikan dalam infrastruktur web global.

Dokumentasi server ini disusun sebagai dasar acuan dalam penyusunan kebijakan teknis yang mencakup perumusan tujuan, penetapan sasaran, serta perancangan sistem server secara menyeluruh. Dokumentasi ini mendukung proses pengembangan dan pemeliharaan infrastruktur server, serta sebagai panduan teknis bagi pembaca yang membutuhkan pemahaman mendalam terhadap konfigurasi dan mekanisme kerja sistem yang diterapkan.

Seluruh konfigurasi dan tahapan implementasi yang tercantum dalam dokumen ini disusun berdasarkan hasil kajian komparatif dari berbagai referensi. Konfigurasi yang digunakan merupakan versi yang telah melalui proses evaluasi dan dipilih sebagai alternatif paling optimal oleh penulis berdasarkan kebutuhan sistem dan praktik terbaik (best practices) di bidang manajemen server.

B. Landasan Teori

1. Definisi

Apache2 adalah singkatan dari **Apache HTTP Server**, sebuah perangkat lunak server web *open-source* yang paling banyak digunakan di dunia. Fungsinya utama adalah untuk **menyajikan konten web** (seperti halaman HTML, gambar, video, dan file lainnya) kepada pengguna internet yang memintanya melalui *browser* web mereka.

2. Efektivitas Domain Name System

Efektivitas **platform manajemen dan dukungan jarak jauh** sangatlah tinggi dan memberikan dampak signifikan pada operasional IT modern. Keefektifan ini dapat diukur dari berbagai aspek kunci :

1. **Stabilitas dan Keandalan:** Apache2 telah teruji waktu dan dikenal sangat stabil serta andal.
2. **Fleksibilitas dan Kustomisasi:** Apache2 sangat fleksibel berkat arsitektur modularnya.
3. **Dukungan Komunitas Luas:** Sebagai proyek *open-source* yang sudah lama ada, Apache2 memiliki komunitas yang sangat besar dan aktif.
4. **Kompatibilitas Lintas Platform:** Apache2 dapat berjalan di berbagai sistem operasi, termasuk Linux, Windows, dan macOS, memberikan fleksibilitas dalam pilihan infrastruktur.
5. **Penanganan Konten Dinamis:** Apache2 sangat efektif dalam memproses konten dinamis secara internal, misalnya melalui modul seperti `mod_php`.

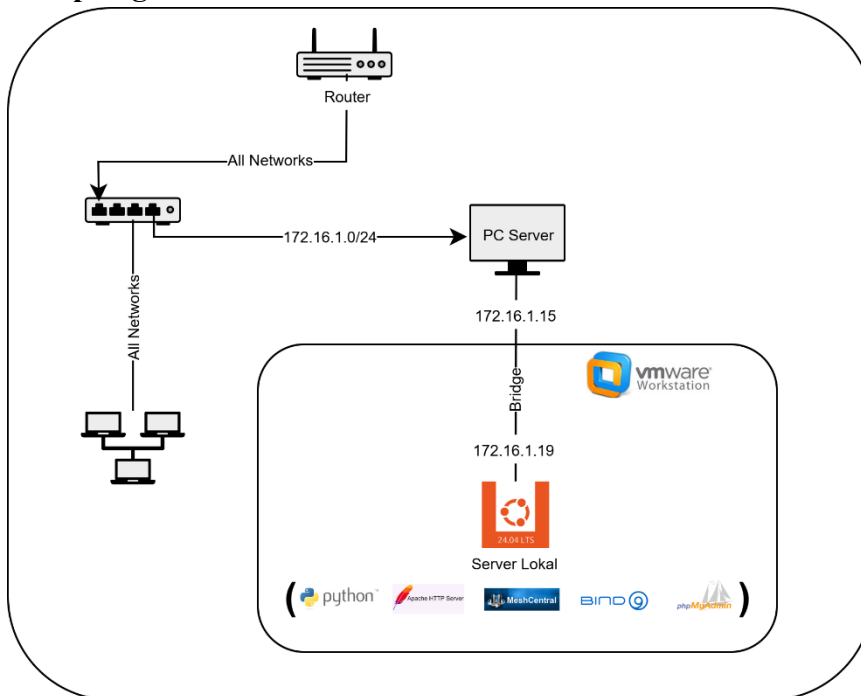
6. **Kematangan dan Keamanan:** Karena sudah lama digunakan dan diuji oleh jutaan orang, Apache2 dianggap sangat matang dalam hal keamanan.

3. Tujuan Domain Name System

Tujuan utama dari **Apache HTTP Server (Apache2)** adalah untuk **menyajikan konten web** dan memfasilitasi komunikasi antara server dan klien (biasanya *web browser*).

C. Informasi Server

1. Topologi Server



2. Persyaratan Server

Memahami dasar pengetahuan dan pemahaman terkait terminologi IT yang akan digunakan oleh server.

Software :

- Apache2
- Allow port 80,443 (jika sudah menginstall firewall)

OS :

- Linux
- Linux

3. Sistem Operasi dan Versi

Menggunakan sistem operasi platform Linux/BSD/Unix. Hypervisor yang digunakan adalah **Vmware Workstation** tujuan utamanya yaitu *user friendly* dan memiliki fitur-fitur yang cocok untuk server lokal skala kecil. OS yang digunakan untuk server adalah **Ubuntu Server 24.04**.

4. Langkah-langkah Konfigurasi Dasar Server Apache2

1. Install paket Apache2

```
root@modernvet:~# apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-lubuntu8.7).
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
root@modernvet:~#
```

2. Enable, start, cek status dan restart apache2.service

```
root@modernvet:~# systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
root@modernvet:~# systemctl start apache2
root@modernvet:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-07-28 05:39:44 UTC; 21h ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 14847 (apache2)
    Tasks: 6 (limit: 4548)
   Memory: 14.0M (peak: 29.0M)
      CPU: 3.844s
   CGroup: /system.slice/apache2.service
           └─14847 /usr/sbin/apache2 -k start
             └─19341 /usr/sbin/apache2 -k start
               └─19342 /usr/sbin/apache2 -k start
                 └─19343 /usr/sbin/apache2 -k start
                   └─19344 /usr/sbin/apache2 -k start
                     └─19345 /usr/sbin/apache2 -k start

Jul 28 05:39:44 modernvet systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 28 05:39:44 modernvet systemd[1]: Started apache2.service - The Apache HTTP Server.
Jul 29 00:00:03 modernvet systemd[1]: Reloading apache2.service - The Apache HTTP Server...
Jul 29 00:00:03 modernvet systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
root@modernvet:~# systemctl restart apache2
root@modernvet:~#
```

- Search [http://\[IP Server\]](http://[IP Server])



3. Penjelasan direktori Apache2

Apache2, terutama pada sistem operasi berbasis Linux seperti Debian/Ubuntu, memiliki struktur direktori yang terorganisir dengan baik untuk mengelola konfigurasi dan kontennya. Berikut adalah beberapa direktori penting yang perlu Anda ketahui:

1. Direktori konfigurasi Utama

- **/etc/apache2/** Ini adalah direktori utama tempat semua file konfigurasi Apache2 disimpan. Direktori ini adalah pusat kendali untuk seluruh server Apache.

- **/etc/apache2/apache2.conf** Ini adalah file konfigurasi utama Apache. File ini berisi pengaturan global untuk server, dan seringkali juga bertanggung jawab untuk menyertakan (include) file-file konfigurasi lainnya dari subdirektori.

- **/etc/apache2/ports.conf** File ini menentukan port mana yang akan didengarkan oleh Apache. Secara *default*, Apache mendengarkan port 80 (HTTP) dan port 443 (HTTPS/SSL) jika modul SSL diaktifkan.

2. Direktori untuk Virtual Host dan Modul

- **/etc/apache2/sites-available/** Direktori ini berisi file konfigurasi untuk Virtual Host yang tersedia. Setiap file di sini mewakili konfigurasi untuk situs web atau aplikasi tertentu yang dapat di-*host* oleh Apache. File-file ini adalah "template" atau konfigurasi yang siap digunakan, tetapi belum aktif.

- **/etc/apache2/sites-enabled/** Direktori ini berisi Virtual Host yang sudah diaktifkan. Biasanya, file di sini adalah symbolic link (tautan simbolis) ke file yang ada di sites-available. Anda bisa mengaktifkan atau menonaktifkan Virtual Host menggunakan perintah seperti `a2ensite` dan `a2dissite`.

- **/etc/apache2/mods-available/** Direktori ini menyimpan file konfigurasi untuk modul-modul Apache yang tersedia. Modul ini adalah tambahan fungsionalitas untuk server Apache, seperti dukungan SSL (`mod_ssl`), penulisan ulang URL (`mod_rewrite`), atau integrasi dengan bahasa pemrograman (`mod_php`).

- **/etc/apache2/mods-enabled/** Direktori ini berisi modul-modul Apache yang sudah diaktifkan, dalam bentuk symbolic link ke file di mods-available. Anda bisa mengaktifkan atau menonaktifkan modul menggunakan perintah `a2enmod` dan `a2dismod`.

- **/etc/apache2/conf-available/** dan **/etc/apache2/conf-enabled/** Direktori ini serupa dengan sites-available dan sites-enabled, tetapi digunakan untuk fragmen konfigurasi umum yang bukan bagian dari Virtual Host tertentu. Contohnya bisa berupa pengaturan kompresi atau header HTTP global. Mereka juga diaktifkan/dinonaktifkan menggunakan perintah `a2enconf` dan `a2disconf`.

3. Direktori Konten Web (Document Root)

- **/var/www/** Ini adalah lokasi standar untuk menyimpan file situs web Anda di banyak distribusi Linux. Secara *default*, file root untuk situs web Apache seringkali terletak di `/var/www/html/`: Ini adalah direktori Document Root default Apache. Ketika Anda pertama kali menginstal Apache, halaman "It Works!" atau halaman default lainnya akan disajikan dari sini. Untuk situs web Anda, Anda bisa membuat subdirektori di dalamnya (misalnya, `/var/www/nama_situs_anda/`) atau mengonfigurasi Virtual Host untuk menunjuk ke lokasi lain.

4. Direktori Log

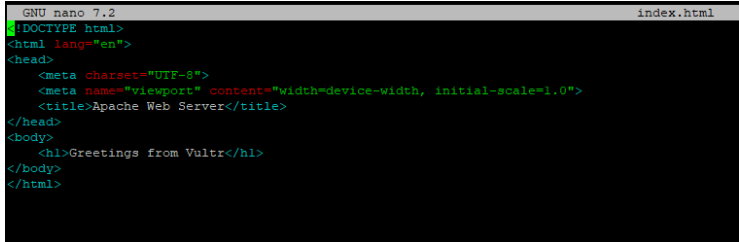
- **/var/log/apache2/** (atau **/var/log/httpd/** pada beberapa sistem) Direktori ini berisi **file log Apache**. **access.log**: Mencatat setiap permintaan yang diterima oleh server, termasuk informasi seperti alamat IP klien, tanggal/waktu, URL yang diminta, kode status HTTP, dan ukuran respons. **error.log**: Mencatat kesalahan yang terjadi pada server, yang sangat penting untuk *debugging* dan pemecahan masalah. (Pastikan *timezone* server sudah di setting)

Memahami struktur direktori ini sangat penting untuk mengelola, mengonfigurasi, dan memecahkan masalah server Apache2 Anda secara efektif. Pada tahap selanjutnya pembaca diharapkan sudah memiliki DNS Server dan sudah konfigurasi DNS Server (tutorial DNS Server terdapat pada **Dokumentasi Konfigurasi BIND9**).

4. Membuat Konten

- Buat direktori tampilan **website** di **/var/www/** lalu buat file **.html** di dalam direktori **website** **/var/www/website/index.html** dan isi file **index.html** dengan tampilan web statis sesuai selera (bisa dapatkan refrensi dari google).

```
root@modernvet:/var/www# mkdir website
root@modernvet:/var/www/website# nano index.html
```



```
GNU nano 7.2 index.html
1 DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Apache Web Server</title>
7 </head>
8 <body>
9   <h1>Greetings from Vultr</h1>
10 </body>
11 </html>
```

5. Membuat Virtual Host

Fitur yang memungkinkan satu server Apache untuk melayani lebih dari satu website (domain atau subdomain) dari satu alamat IP yang sama atau bahkan dari beberapa IP.

- Edit file **/etc/apache2/apache2.conf** dan masukan **ServerName**. Ini berfungsi untuk mencegah peringatan error karena apache tidak mengenali nama host default nya.

```
GNU nano 7.2 apache2.conf
1 ServerName modernvet.net
2 # This is the main Apache server configuration file. It contains the
3 # configuration directives that give the server its instructions.
4 # See http://httpd.apache.org/docs/2.4/ for detailed information about
5 # the directives and /usr/share/doc/apache2/README.Debian about Debian specific
6 # hints.
7 #
```

- Buat file virtual host di **/etc/apache2/sites-available/** (nama file yang dibuat **website.conf**)

```
root@modernvet:/etc/apache2/sites-available# nano website.conf
```

```
GNU nano 7.2 /etc/apache2/sites-available/website.conf
<VirtualHost *:80>
    ServerName modernvet.net
    ServerAdmin webmaster@modernvet.net

    DocumentRoot /var/www/website/

    <Directory /var/www/website>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/website_error.log
    CustomLog ${APACHE_LOG_DIR}/website_access.log combined
</VirtualHost>
```

<VirtualHost *:80>

ServerName modernvet.net

ServerAdmin webmaster@modernvet.net

DocumentRoot /var/www/website/

<Directory /var/www/website>

Options Indexes FollowSymLinks

AllowOverride All

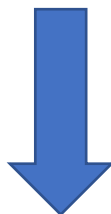
Require all granted

</Directory>

ErrorLog \${APACHE_LOG_DIR}/website_error.log

CustomLog \${APACHE_LOG_DIR}/website_access.log combined

</VirtualHost>



- **<VirtualHost *:80>** Ini mendeklarasikan blok virtual host untuk semua koneksi di **port 80** (HTTP). Tanda ***** berarti menerima permintaan dari semua IP.
- **ServerName modernvet.net** Nama domain utama yang digunakan untuk virtual host ini. Apache akan menggunakan blok ini untuk permintaan ke **modernvet.net**.
- (Opsional) Tambahkan, contoh: **ServerAlias www.modernvet.net** berguna jika klien mencari **modernvet.net** atau **www.modernvet.net** maka akan muncul konten yang sama sesuai konfigurasi. (***.modernvet.net**) untuk mengarahkan semua subdomain ke virtual host yang sama.
- **ServerAdmin webmaster@modernvet.net** Alamat email admin situs. Akan ditampilkan di halaman error (misalnya 500 Internal Server Error) sebagai kontak teknis.
- **DocumentRoot /var/www/website** Lokasi folder utama situs web, semua file HTML, PHP, dan lainnya akan dilayani dari sini. Baris ini bekerja dengan mencari direktori bukan file, direktori **website** sudah disiapkan penulis sebelumnya dan berisi konten web.
- **<Directory /var/www/website> ... </Directory>** Konfigurasi akses untuk folder DocumentRoot.
 - **Options Indexes FollowSymLinks**
 Indexes: jika tidak ada index.html, Apache bisa menampilkan daftar file di direktori (bisa dimatikan dengan menambah - untuk keamanan).
 FollowSymLinks: izinkan Apache mengikuti symlink (shortcut) dalam direktori.
 - **AllowOverride All** Mengizinkan file .htaccess di dalam folder ini menimpa pengaturan Apache. Diperlukan untuk CMS seperti WordPress.
 - **Require all granted** Mengizinkan akses ke semua klien. Jika tidak diset, bisa ditolak secara default.
- **ErrorLog \${APACHE_LOG_DIR}/website_error.log** Lokasi file untuk log error (misalnya file tidak ditemukan, permission denied), file **website_erro.log** akan otomatis terbuat di **/var/log/apache2**.
- **CustomLog \${APACHE_LOG_DIR}/modernvet_access.log combined** Log semua permintaan HTTP yang masuk. Format combined mencatat data lengkap (alamat IP, agen browser, referer, dll). file **website_erro.log** akan otomatis terbuat di **/var/log/apache2**.

- Enable file virtual host **website.conf** dan disable virtual host default

```
root@modernvet:/etc/apache2/sites-available# a2ensite website.conf
Enabling site website.
To activate the new configuration, you need to run:
    systemctl reload apache2

root@modernvet:/etc/apache2/sites-available# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@modernvet:/etc/apache2/sites-available#
```

- Reload, restart, dan cek status Apache2

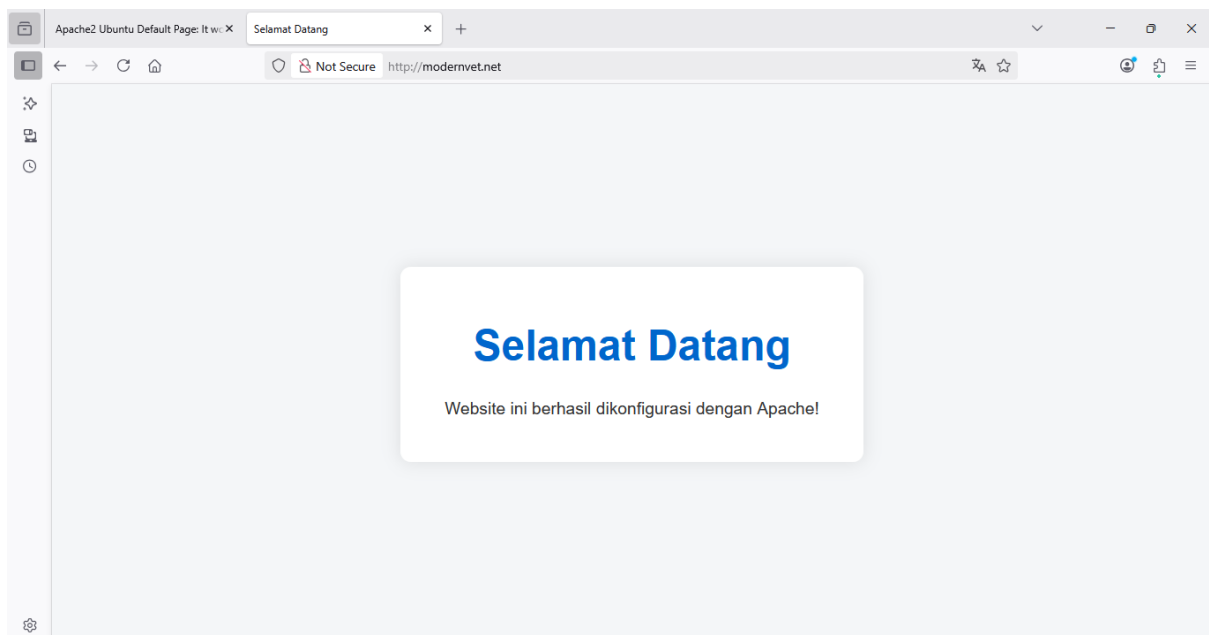
```
root@modernvet:~# systemctl reload apache2.service
root@modernvet:~# systemctl restart apache2.service
root@modernvet:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-07-29 08:18:07 UTC; 8s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 22073 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 22076 (apache2)
    Tasks: 6 (limit: 4548)
   Memory: 13.8M (peak: 14.2M)
      CPU: 49ms
   CGroup: /system.slice/apache2.service
           └─22076 /usr/sbin/apache2 -k start
             └─22078 /usr/sbin/apache2 -k start
               └─22079 /usr/sbin/apache2 -k start
                 └─22080 /usr/sbin/apache2 -k start
                   └─22081 /usr/sbin/apache2 -k start
                     └─22082 /usr/sbin/apache2 -k start

Jul 29 08:18:07 modernvet systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 29 08:18:07 modernvet systemd[1]: Started apache2.service - The Apache HTTP Server.
root@modernvet:~# \
```

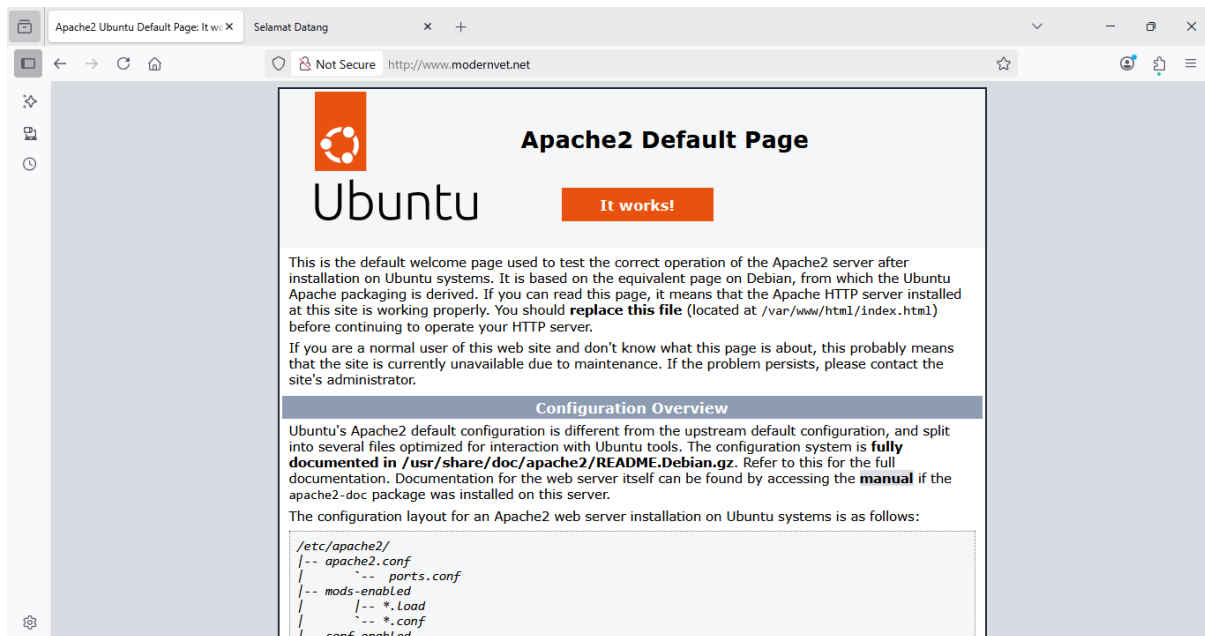
- Cek konfigurasi Apache2 **apachectl configtest**

```
root@modernvet:~# apachectl configtest
Syntax OK
root@modernvet:~#
```

6. Uji Coba Konfigurasi



9999999999%% Berhasil



Mengapa masih muncul tampilan default jika kita searh dengan subdomain? Karena Apache memilih *default virtual host* jika tidak menemukan yang cocok dengan nama host yang diminta, untuk mencegah hal ini kita dapat mengarahkan semua subdomain ke tampilan website baru pada virtual host yang kita buat lalu tambahkan **ServerAlias** atau buat virtual host kosong sebagai fallback, ikuti langkah nomor 7 di bawah ini.

7. Membuat Virtual Host kosong sebagai Fallback

Buat VirtualHost yang **menangani semua permintaan yang tidak dikenal**, dan arahkan ke direktori kosong atau tidak valid.

- Buat folder *dummy* di **/var/www/** (nama folder yang dibuat blackhole)

```
root@modernvet:/var/www# mkdir blackhole
```

biarkan folder ini kosong

- Buat file virtual host di **/etc/apache2/sites-available/** (nama file yang dibuat **000-blackhole.conf**), file ini diberi nama dengan awalan **000-** supaya selalu dibaca pertama dan menjadi fallback jika tidak ada **ServerName** yang cocok.

```
root@modernvet:/etc/apache2/sites-available# cp 000-default.conf 000-blackhole.conf
```

bisa buat dengan perintah **nano** atau copy file default dengan perintah **cp**

```
GNU nano 7.2                                000-blackhole.conf *
<VirtualHost *:80>
    ServerName _
    DocumentRoot /var/www/blackhole
    <Directory /var/www/blackhole>
        Require all denied
    </Directory>
</VirtualHost>
```



```

<VirtualHost *:80>

    ServerName _

    DocumentRoot /var/www/blackhole

    <Directory /var/www/blackhole>

        Require all denied

    </Directory>

</VirtualHost>

```

- **<VirtualHost *:80>** Untuk Akses dari HTTP
- **ServerName _**
Fungsi: Menentukan nama domain utama yang akan ditangani oleh virtual host ini.
Tanda `_` biasanya digunakan sebagai *placeholder* untuk permintaan yang tidak cocok dengan `ServerName` manapun, sehingga bisa dijadikan fallback.
Bukan domain valid, tetapi diterima oleh Apache sebagai nama host "sementara".
- **DocumentRoot /var/www/blackhole**
Fungsi: Menentukan direktori yang akan digunakan sebagai root dari dokumen web.
Dalam kasus ini, Apache diarahkan ke folder **/var/www/blackhole**, biasanya folder kosong atau tidak ada konten web. Tujuannya adalah agar tidak menampilkan konten apapun jika subdomain tidak dikenal.
- **Directory /var/www/blackhole> ... </Directory>** Konfigurasi akses untuk folder DocumentRoot.
Require all denied berfungsi Menolak semua akses HTTP ke direktori ini. Artinya, semua permintaan ke virtual host ini akan mendapat respon **"403 Forbidden"**, tanpa pengecualian.
- **</VirtualHost>**
Fungsi: Menutup blok virtual host yang dimulai sebelumnya.

- Aktivasi virtual host, dan restart **apache2.service**

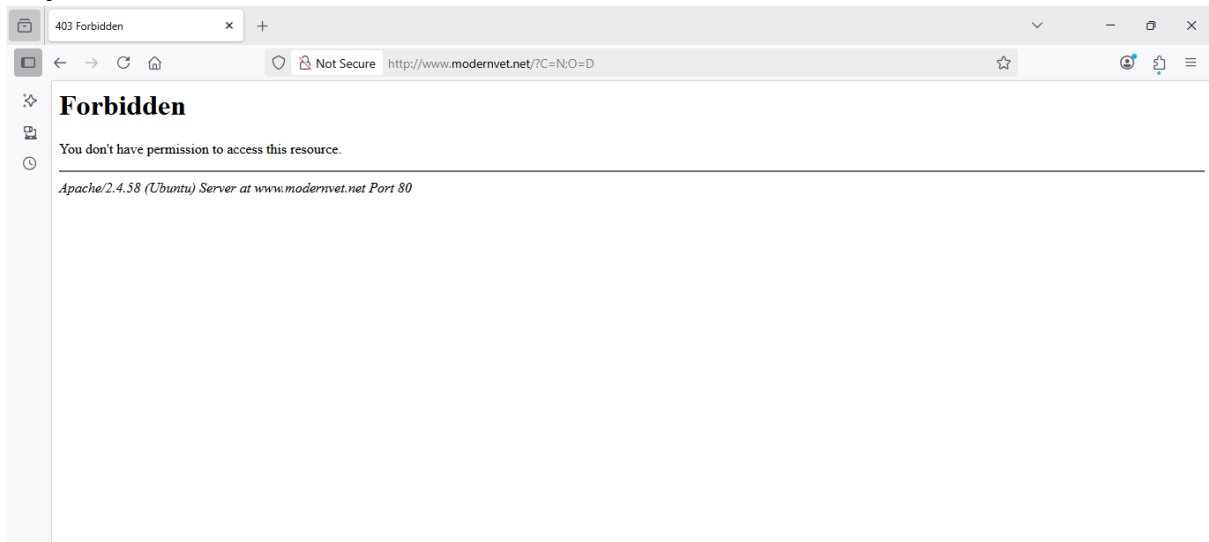
```

root@modernvet:/etc/apache2/sites-available# a2ensite 000-blackhole.conf
Enabling site 000-blackhole.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@modernvet:/etc/apache2/sites-available# systemctl reload apache2.service
root@modernvet:/etc/apache2/sites-available# systemctl restart apache2.service
root@modernvet:/etc/apache2/sites-available# █

```



- Uji coba



tes ke semua subdomain dan pastikan **access denied**.

5. Membuat Self-Signed TLS

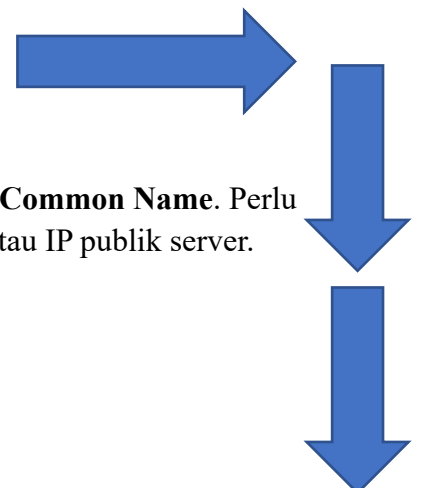
Server web Apache mendengarkan permintaan koneksi pada port HTTP port 80 secara default. HTTP memungkinkan permintaan yang tidak terenkripsi antara server web dan peramban pengguna. HTTPS memungkinkan koneksi terenkripsi menggunakan sertifikat SSL yang valid untuk mengautentikasi server web dengan peramban pengguna.

TLS (Transport Layer Security) adalah **protokol keamanan** yang digunakan untuk mengenkripsi komunikasi antara **client (browser)** dan **server (web)**.

1. Buat Sertifikat Self-Signed

```
root@modernvet:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/wselfsigned.key -out /etc/ssl/certs/wselfsigned.crt
.....
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:JKT
Locality Name (eg, city) []:JKT
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MPET
Organizational Unit Name (eg, section) []:MPET
Common Name (e.g. server FQDN or YOUR name) []:modernvet.net
Email Address []:webmaster@modernvet.net
root@modernvet:~#
```

**openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/wselfsigned.key -out /etc/ssl/certs/wselfsigned.crt**



Isi perintah dengan benar, baris terpenting adalah baris yang meminta **Common Name**. Perlu memasukkan nama host yang akan digunakan untuk mengakses server, atau IP publik server.

- **openssl** Ini adalah alat baris perintah untuk membuat dan mengelola sertifikat OpenSSL, kunci, dan file lainnya.
- **req -x509** Ini menetapkan bahwa kami ingin menggunakan manajemen permintaan penandatanganan sertifikat (CSR) X.509. X.509 adalah standar infrastruktur kunci publik yang dipatuhi TLS untuk manajemen kunci dan sertifikat.
- **-nodes** Ini memberi tahu OpenSSL untuk melewati opsi pengamanan sertifikat kita dengan frasa sandi. Kita membutuhkan Apache agar dapat membaca berkas, tanpa campur tangan pengguna, saat server dinyalakan. Frasa sandi akan mencegah hal ini terjadi, karena kita harus memasukkannya setiap kali server dinyalakan ulang.
- **-days 365** Opsi ini mengatur lamanya masa berlaku sertifikat. Kami menetapkannya selama satu tahun di sini. Banyak peramban modern akan menolak sertifikat apa pun yang masa berlakunya lebih dari satu tahun.
- **-newkey rsa:2048** Ini menetapkan bahwa kita ingin membuat sertifikat baru dan kunci baru secara bersamaan. Kita tidak membuat kunci yang diperlukan untuk menandatangani sertifikat pada langkah sebelumnya, jadi kita perlu membuatnya bersama sertifikat tersebut. **rsa:2048** Bagian ini memerintahkannya untuk membuat kunci RSA sepanjang 2048 bit.
- **-keyout** Baris ini memberi tahu OpenSSL di mana menempatkan file kunci privat yang dihasilkan yang sedang kita buat.
- **-out** Ini memberi tahu OpenSSL di mana harus menempatkan sertifikat yang kita buat.

2. Deklarasi *Self-Signed Certificate* ke Virtual Host

- Buat file virtualhost untuk HTTPS, di sini penulis ingin mengubah **website** pada konfigurasi di atas menggunakan HTTPS, dan edit virtualhost nya.

```
root@modernvet:/etc/apache2/sites-available# mv website.conf website-ssl.conf

GNU nano 7.2                               website-ssl.conf *
<VirtualHost *:443>
    ServerName modernvet.net
    ServerAdmin webmaster@modernvet.net

    DocumentRoot /var/www/website/

    <Directory /var/www/website>
        Options -Indexes +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/wselfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/wselfsigned.key

    ErrorLog ${APACHE_LOG_DIR}/website_error.log
    CustomLog ${APACHE_LOG_DIR}/website_access.log combined
</VirtualHost>
```

<VirtualHost *:443>

SSLEngine on

SSLCertificateFile /etc/ssl/certs/wselfsigned.crt

SSLCertificateKeyFile /etc/ssl/private/wselfsigned.key



- **<VirtualHost *:443>** Port 443 karena web akan di akses melalui HTTPS
- **SSLEngine on**
Fungsi: Mengaktifkan modul SSL untuk Virtual Host ini.
Artinya: Apache akan mendengarkan permintaan HTTPS (port 443) dan memproses koneksi terenkripsi.
- **SSLCertificateFile /etc/ssl/certs/wsselfsigned.crt**
Fungsi: Menentukan lokasi file sertifikat SSL yang akan digunakan.
File **.crt** ini berisi *public key* yang digunakan untuk mengenkripsi data antara server dan klien.
Sertifikat ini akan dikirim ke browser agar bisa melakukan validasi terhadap koneksi aman.
- **SSLCertificateKeyFile /etc/ssl/private/wsselfsigned.key**
Fungsi: Menentukan lokasi private key dari sertifikat.
File **.key** ini digunakan server untuk mendekripsi informasi yang dienkripsi oleh klien.
Harus dijaga tetap rahasia. Jika bocor, maka koneksi aman bisa disusupi.

3. Aktifkan SSL dan virtual Host

- Enable ssl, enable virtual host, dan restart apache2.service

```
root@modernvet:/etc/apache2/sites-available# a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
root@modernvet:/etc/apache2/sites-available# a2ensite website-ssl.conf
Site website-ssl already enabled
root@modernvet:/etc/apache2/sites-available# systemctl reload apache2.service
root@modernvet:/etc/apache2/sites-available#
```

a2enmod ssl Mengaktifkan modul SSL di Apache.

a2ensite website-ssl.conf Mengaktifkan virtual host konfigurasi untuk HTTPS.

systemctl restart apache2.service

4. Uji Coba dan Buat Virtual Host Untuk Redirect HTTP ke HTTPS



ketika melakukan search hanya dengan DNS maka akan otomatis menggunakan protokol HTTP, jika ingin mengakses web dengan ssl kita harus menambahkan protokol HTTPS di pencaharian (`https://<namadomain>`). Sangat tidak efisien bukan??? Mari kita buat virtual host redirect port 80 HTTP ke port 443 HTTPS.

- Buat virtual host baru untuk redirect

```
root@modernvet:/etc/apache2/sites-available# nano website.conf
```


5. Membuat Virtual Host kosong sebagai Fallback Untuk port 443 (HTTPS)

Pada langkah-langkah sebelumnya penulis sudah membuat virtualhost kosong sebagai fallback. Namun, itu hanya berlaku di port 80 (HTTP), jika klien search menggunakan protokol 443 (HTTPS) ditambah dengan subdomain maka akan masuk ke website yang sudah dibuat dengan TLS/SSL.

- Buat file virtual host di **/etc/apache2/sites-available/** (nama file yang dibuat **000-blackhole-ssl.conf**), file ini diberi nama dengan awalan **000-** supaya selalu dibaca pertama dan menjadi fallback jika tidak ada ServerName yang cocok.

```
root@modernvet:/etc/apache2/sites-available# nano 000-blackhole-ssl.conf
```

```
GNU nano 7.2                                000-blackhole-ssl.conf
<VirtualHost *:443>
    ServerName _
    DocumentRoot /var/www/blackhole
    <Directory /var/www/blackhole>
        Require all denied
    </Directory>

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/wselfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/wselfsigned.key
</VirtualHost>
```

<VirtualHost *:443>

ServerName _

DocumentRoot /var/www/blackhole

<Directory /var/www/blackhole>

Require all denied

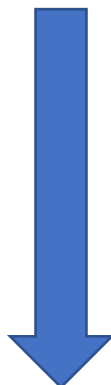
</Directory>

SSLEngine on

SSLCertificateFile /etc/ssl/certs/wselfsigned.crt

SSLCertificateKeyFile /etc/ssl/private/wselfsigned.key

</VirtualHost>



- **<VirtualHost *:443>** Untuk Akses dari HTTPS
- **ServerName _**
Fungsi: Menentukan nama domain utama yang akan ditangani oleh virtual host ini.
Tanda `_` biasanya digunakan sebagai *placeholder* untuk permintaan yang tidak cocok dengan `ServerName` manapun, sehingga bisa dijadikan fallback.
Bukan domain valid, tetapi diterima oleh Apache sebagai nama host "sementara".
- **DocumentRoot /var/www/blackhole**
Fungsi: Menentukan direktori yang akan digunakan sebagai root dari dokumen web.
Dalam kasus ini, Apache diarahkan ke folder `/var/www/blackhole`, biasanya folder kosong atau tidak ada konten web. Tujuannya adalah agar tidak menampilkan konten apapun jika subdomain tidak dikenal.
- **Directory /var/www/blackhole> ... </Directory>** Konfigurasi akses untuk folder `DocumentRoot`.
Require all denied berfungsi menolak semua akses HTTPS ke direktori ini. Artinya, semua permintaan ke virtual host ini akan mendapat respon **"403 Forbidden"**, tanpa pengecualian.
- **SSLEngine on**
Fungsi: Mengaktifkan modul SSL untuk Virtual Host ini.
Artinya: Apache akan mendengarkan permintaan HTTPS (port 443) dan memproses koneksi terenkripsi.
- **SSLCertificateFile /etc/ssl/certs/wselfsigned.crt**
Fungsi: Menentukan lokasi file sertifikat SSL yang akan digunakan.
File `.crt` ini berisi *public key* yang digunakan untuk mengenkripsi data antara server dan klien.
Sertifikat ini akan dikirim ke browser agar bisa melakukan validasi terhadap koneksi aman.
- **SSLCertificateKeyFile /etc/ssl/private/wselfsigned.key**
Fungsi: Menentukan lokasi private key dari sertifikat.
File `.key` ini digunakan server untuk mendekripsi informasi yang dienkripsi oleh klien.
Harus dijaga tetap rahasia. Jika bocor, maka koneksi aman bisa disusupi.
- **</VirtualHost>**
Fungsi: Menutup blok virtual host yang dimulai sebelumnya.

Opsional menambahkan SSL pada virtualhost ini tergantung tampilan apa yang ingin ditampilkan ketika klien akses dengan HTTPS

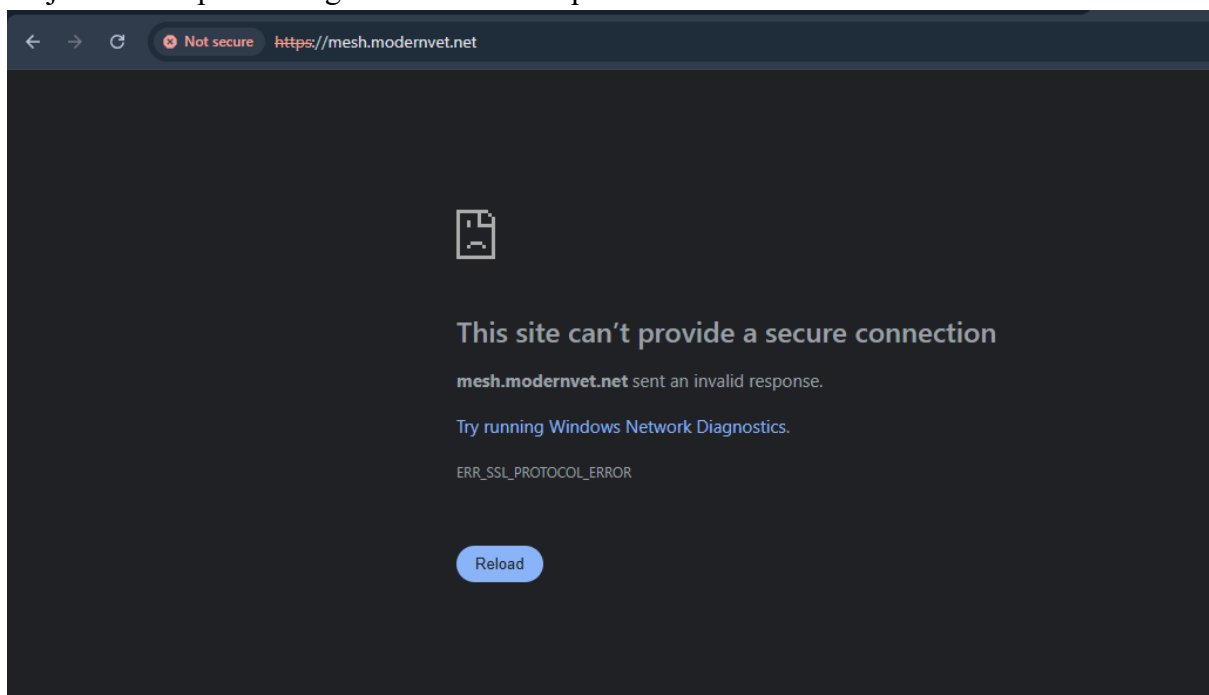
- Enable virtual host dan restart apache2.service

```
root@modernvet:/etc/apache2/sites-available# a2ensite 000-blackhole-ssl.conf
Site 000-blackhole-ssl already enabled
root@modernvet:/etc/apache2/sites-available# systemctl restart apache2.service
root@modernvet:/etc/apache2/sites-available#
```


- Uji coba dan perbandingan virtual host dengan SSL



- Uji coba dan perbandingan virtual host tanpa SSL



Sampai pada tahap ini, konfigurasi web server Apache2 telah cukup memadai untuk memenuhi kebutuhan dasar. Langkah selanjutnya dapat diikuti apabila pembaca memiliki website atau aplikasi web yang berjalan pada port tertentu untuk diakses melalui peramban web.

6. Reverse Proxy dengan mod_proxy

Reverse Proxy memungkinkan Apache2 meneruskan permintaan (request) dari klien ke server backend lain (misalnya aplikasi yang berjalan di port tertentu seperti Node.js, Python, atau layanan lain). Apache akan berperan sebagai perantara antara klien dan backend, menyembunyikan backend dari pengguna.

1. Aktifkan Modul Proxy di Apache

```
root@modernvet:/etc/apache2/sites-available# a2enmod proxy
Module proxy already enabled
root@modernvet:/etc/apache2/sites-available# a2enmod proxy_http
Considering dependency proxy for proxy_http:
Module proxy already enabled
Module proxy_http already enabled
root@modernvet:/etc/apache2/sites-available# a2enmod rewrite
Module rewrite already enabled
```

a2enmod proxy Modul inti untuk fungsionalitas *proxy*.

a2enmod proxy_http Diperlukan untuk meneruskan permintaan HTTP.

a2enmod rewrite Berguna untuk menulis ulang URL, yang sering kali diperlukan dalam konfigurasi *reverse proxy*.

2. Membuat Virtual Host

Sebelumnya penulis sudah membuat website, website yang dibuat penulis adalah **dashboard nvr** yang berjalan pada port 5000. Pembaca diharapkan sudah memiliki website yang berjalan pada port tertentu atau bisa lihat Dokumentasi **Redirect Port with Python**.

```
root@modernvet:/etc/apache2/sites-available# nano nvr.conf
GNU nano 7.2 nvr.conf
<VirtualHost *:80>
    ServerName nvr.modernvet.net

    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:5000/
    ProxyPassReverse / http://127.0.0.1:5000/

    ErrorLog ${APACHE_LOG_DIR}/nvr_error.log
    CustomLog ${APACHE_LOG_DIR}/nvr_access.log combined
</VirtualHost>
```

<VirtualHost *:80>

ServerName nvr.modernvet.net

ProxyPreserveHost On

ProxyPass / http://127.0.0.1:5000/

ProxyPassReverse / http://127.0.0.1:5000/

ErrorLog \${APACHE_LOG_DIR}/nvr_error.log

CustomLog \${APACHE_LOG_DIR}/nvr_access.log combined

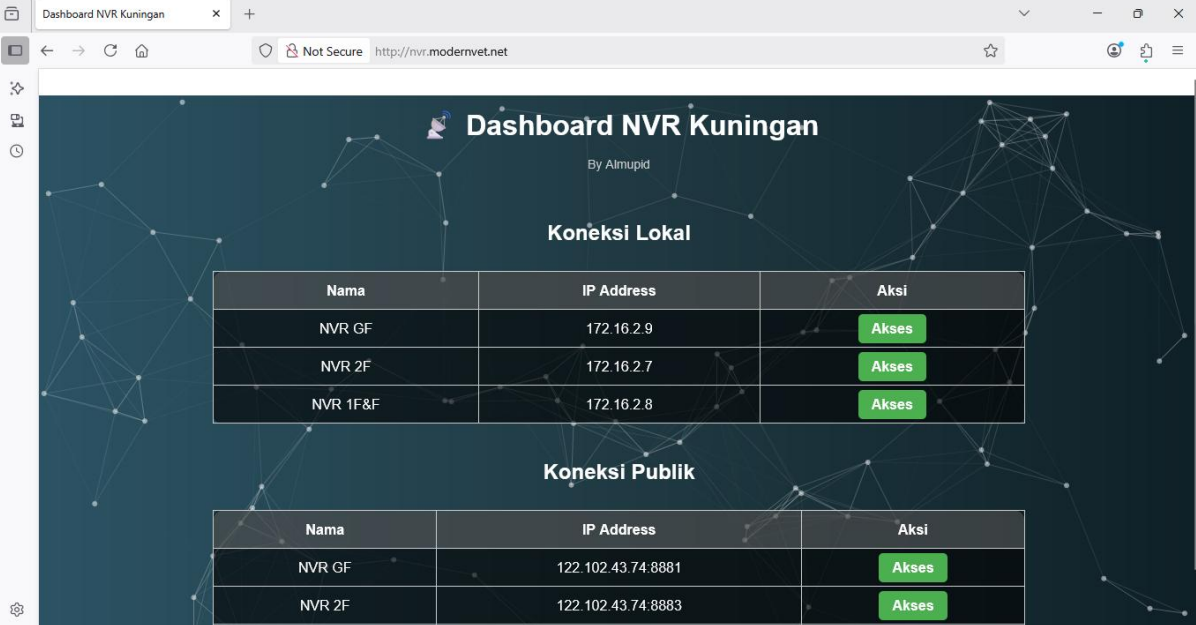
</VirtualHost>



- **ProxyPreserveHost On** Direktif ini akan meneruskan header Host asli dari klien ke server backend. Tanpa ini, server backend akan melihat proxy sebagai Host.
- **ProxyPass / http://127.0.0.1:5000/** Direktif utama reverse proxy. Ini memberitahu Apache untuk meneruskan semua permintaan (/) ke server yang berjalan di http://127.0.0.1:5000/.
- **ProxyPassReverse / http://127.0.0.1:5000/** Direktif ini menyesuaikan header respons dari server backend, seperti Location header, sehingga mengarah ke alamat proxy (contoh.com) alih-alih alamat backend (127.0.0.1:5000).
 ###127.0.0.1 adalah ip localhost server, jika konten berjalan pada server berbeda bisa disesuaikan.
 ###Untuk direktif lainnya sudah dijelaskan pada langkah-langkah sebelumnya.

- Enable virtual host, restart apache2.service, dan uji coba

```
root@modernvet:/etc/apache2/sites-available# a2ensite nvr.conf
Enabling site nvr.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@modernvet:/etc/apache2/sites-available# systemctl restart apache2.service
root@modernvet:/etc/apache2/sites-available#
```



Dashboard NVR Kuningan
By Almupid

Koneksi Lokal

| Nama | IP Address | Aksi |
|----------|------------|-----------------------|
| NVR GF | 172.16.2.9 | Akses |
| NVR 2F | 172.16.2.7 | Akses |
| NVR 1F&F | 172.16.2.8 | Akses |

Koneksi Publik

| Nama | IP Address | Aksi |
|--------|--------------------|-----------------------|
| NVR GF | 122.102.43.74:8881 | Akses |
| NVR 2F | 122.102.43.74:8883 | Akses |

8. Reverse Proxy dengan Load Balancer Server

Load balancing Apache2 adalah proses mendistribusikan beban trafik jaringan atau aplikasi ke beberapa server Apache2 yang tersedia secara merata. Tujuannya adalah untuk mencegah satu server kewalahan menangani semua permintaan, sehingga meningkatkan kinerja, ketersediaan, dan keandalan aplikasi web.



1. Aktifkan Modul Balancer

```
root@modernvet:/etc/apache2/sites-available# a2enmod proxy_balancer
Considering dependency proxy for proxy_balancer:
Module proxy already enabled
Considering dependency alias for proxy_balancer:
Module alias already enabled
Considering dependency slotmem_shm for proxy_balancer:
Enabling module slotmem_shm.
Enabling module proxy_balancer.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@modernvet:/etc/apache2/sites-available# a2enmod lbmethod_byrequests
Considering dependency proxy_balancer for lbmethod_byrequests:
Considering dependency proxy for proxy_balancer:
Module proxy already enabled
Considering dependency alias for proxy_balancer:
Module alias already enabled
Considering dependency slotmem_shm for proxy_balancer:
Module slotmem_shm already enabled
Module proxy_balancer already enabled
Enabling module lbmethod_byrequests.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@modernvet:/etc/apache2/sites-available#
```

a2enmod proxy_balancer

a2enmod lbmethod_byrequest Mengaktifkan metode load balancing berdasarkan **jumlah request** yang diterima oleh masing-masing backend.

2. Buat File Global dan Virtual Host

- Buat file global di **/etc/apache2/conf-available/**

```
root@modernvet:/etc/apache2/conf-available# nano balancer-manager.conf
GNU nano 7.2 conf-enabled/balancer-manager.conf
<Proxy balancer://mycluster>
    BalancerMember http://172.16.1.19:5000
    BalancerMember http://172.16.1.19:5001
</Proxy>

<Location /balancer-manager>
    SetHandler balancer-manager
    Require ip 172.16.1.0/24
</Location>
```

```
<Proxy balancer://mycluster>
    BalancerMember http://172.16.1.19:5000
    BalancerMember http://172.16.1.19:5001
</Proxy>
```

```
<Location /balancer-manager>
    SetHandler balancer-manager
    Require ip 172.16.1.0/24
</Location>
```

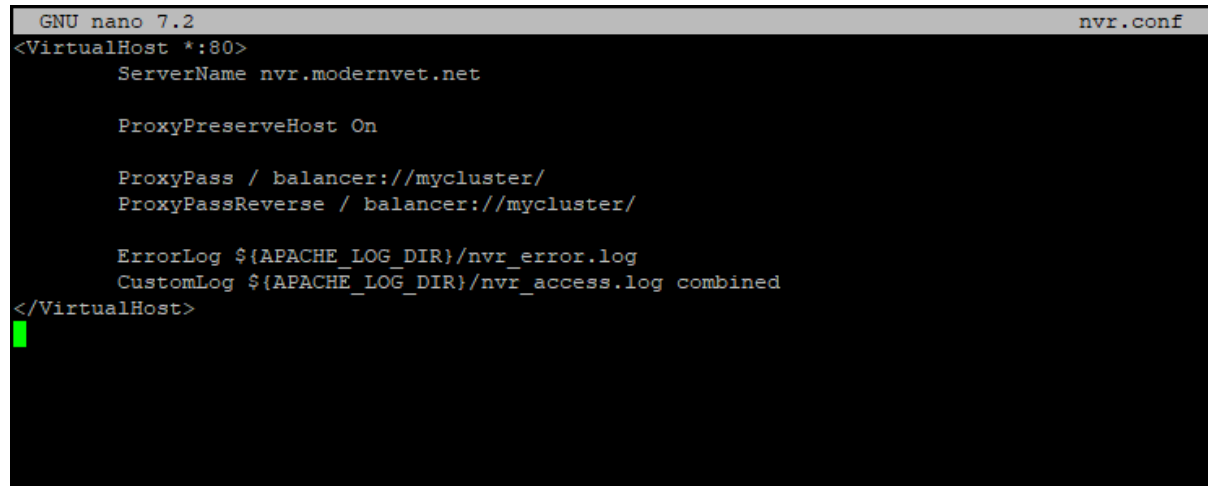
Bagian Proxy Balancer

- **balancer://mycluster** mendefinisikan nama kluster load balancer dengan nama **mycluster**.
- **BalancerMember** mendefinisikan backend server yang menjadi anggota dari kluster tersebut.

Bagian Location balancer-manager

- **<Location /balancer-manager>** menyatakan URL path yang digunakan untuk mengakses dashboard balancer Apache.
- **SetHandler balancer-manager** mengaktifkan halaman manajemen balancer yang disediakan oleh Apache.
- **Require ip 172.16.1.0/24** hanya mengizinkan akses ke halaman ini dari jaringan 172.16.1.0/24 (yaitu hanya IP dalam subnet tersebut yang bisa membuka /balancer-manager). Gunakan **Require all granted** agar bisa diakses dari mana saja.

- Buat/edit virtual host



```
GNU nano 7.2 nvr.conf
<VirtualHost *:80>
    ServerName nvr.modernvet.net

    ProxyPreserveHost On

    ProxyPass / balancer://mycluster/
    ProxyPassReverse / balancer://mycluster/

    ErrorLog ${APACHE_LOG_DIR}/nvr_error.log
    CustomLog ${APACHE_LOG_DIR}/nvr_access.log combined
</VirtualHost>
```

```
<VirtualHost *:80>
    ServerName nvr.modernvet.net

    ProxyPreserveHost On

    ProxyPass / balancer://mycluster/
    ProxyPassReverse / balancer://mycluster/

    ErrorLog ${APACHE_LOG_DIR}/nvr_error.log
    CustomLog ${APACHE_LOG_DIR}/nvr_access.log combined
</VirtualHost>
```

- **ProxyPreserveHost On** Direktif ini akan meneruskan header Host asli dari klien ke server backend. Tanpa ini, server backend akan melihat proxy sebagai Host.
- **ProxyPass / balancer://mycluster/** Mengarahkan semua permintaan (/) ke kluster balancer mycluster, yang didefinisikan di file balancer-manager.conf.
- **ProxyPassReverse / balancer://mycluster/** Menyesuaikan URL balik (response Location atau Redirect) agar tetap sesuai domain frontend.

3. Aktifkan file konfigurasi dan restart apache2.service

```
root@modernvet:/etc/apache2# a2enconf balancer-manager
Conf balancer-manager already enabled
root@modernvet:/etc/apache2# a2ensite nvr.conf
Site nvr already enabled
root@modernvet:/etc/apache2# systemctl restart apache2.service
```

- Uji coba, dan refresh-refresh halaman

The screenshot shows a web browser window with the URL <http://nvr.modernvet.net>. The page has a dark blue background with a network diagram. It contains two tables: 'Koneksi Lokal' and 'Koneksi Publik'. Each table has three columns: 'Nama', 'IP Address', and 'Aksi'.

| Koneksi Lokal | | |
|---------------|------------|-----------------------|
| Nama | IP Address | Aksi |
| NVR GF | 172.16.2.9 | Akses |
| NVR 2F | 172.16.2.7 | Akses |
| NVR 1F&F | 172.16.2.8 | Akses |

| Koneksi Publik | | |
|----------------|--------------------|-----------------------|
| Nama | IP Address | Aksi |
| NVR GF | 122.102.43.74:8881 | Akses |
| NVR 2F | 122.102.43.74:8883 | Akses |
| NVR 1F&F | 122.102.43.74:8882 | Akses |

- Cek balancer-manager

The screenshot shows a web browser window with the URL <http://172.16.1.19/balancer-manager>. The page title is 'Load Balancer Manager for 172.16.1.19'. It displays server information and a table of balancer status.

Server Version: Apache/2.4.58 (Ubuntu) OpenSSL/3.0.13
Server Built: 2025-07-14T16:22:22
Balancer changes will NOT be persisted on restart.
Balancers are inherited from main server.
ProxyPass settings are inherited from main server.

LoadBalancer Status for [balancer://mycluster](#) [p28e43231_mycluster]

| MaxMembers | StickySession | DisableFailover | Timeout | FailoverAttempts | Method | Path | Active |
|------------|---------------|-----------------|---------|------------------|------------|------|--------|
| 2 [2 Used] | (None) | Off | 0 | 1 | byrequests | / | Yes |

| Worker URL | Route | RouteRedir | Factor | Set | Status | Elected | Busy | Load | To | From |
|---|-------|------------|--------|-----|---------|---------|------|------|-----|------|
| http://172.16.1.19:5000 | | | 1.00 | 0 | Init Ok | 2 | 0 | -100 | 908 | 8.0K |
| http://172.16.1.19:5001 | | | 1.00 | 0 | Init Ok | 1 | 0 | 100 | 454 | 4.0K |

Apache/2.4.58 (Ubuntu) Server at 172.16.1.19 Port 80

Pastikan **status init OK** untuk ke 2 alamat server web.

9. Membuat Virtual Host Untuk Web Application

Pada tahap ini (9) pembaca di harapkan sudah install **MeshCentral** untuk contoh konfigurasi **Apache2 proxy_wstunnel** dan **rewrite**. Tutorial bisa dilihat pada **Dokumentasi Teknikal MeshCentral**.

1. Edit file konfigurasi meshcentral

```
GNU nano 7.2 /opt/meshcentral/meshcentral-data/config.json
["$schema": "https://raw.githubusercontent.com/Ylianst/MeshCentral/master/meshcentral-config-schema.json",
  "_comment1_": "This is a simple configuration file, all values and sections that start with underscore (_) are ignored. Edit a section and remove the _ in front of",
  "_comment2_": "See node_modules/meshcentral/sample-config-advanced.json for a more advanced example.",
  "Settings": {
    "cert": "mesh.modernvet.net",
    "WANOnly": true,
    "LANOnly": true,
    "sessionKey": "MyReallySecretPassword1",
    "port": 8080,
    "aliasPort": 443,
    "redisPort": 80,
    "_redisAliasPort": 80
  },
  "domains": {
    "": {
      "title": "MyServer",
      "title2": "Servername",
      "minify": true,
      "newAccounts": true,
      "userNameIsEmail": true
    }
  },
  "letsencrypt": {
    "_comment_": "Requires NodeJS 8.x or better, Go to https://letsdebug.net/ first before trying Let's Encrypt.",
    "email": "myemail@mydomain.com",
    "names": "mesh.modernvet.net",
    "skipChallengeVerification": true,
    "production": false
  }
}]
```

edit file ini sesuaikan pada gambar

2. Buat virtual host untuk meshcentral

Buat virtual host dengan ssl dan direktif lainnya dan 1 virtual host lagi untuk redirect http ke https.

- Virtual host dengan ssl dan direktif lainnya

```
GNU nano 7.2 meshcentral-ssl.conf
<VirtualHost *:443>
    ServerName mesh.modernvet.net
    ServerAdmin webmaster@modernvet.net

    ProxyPreserveHost On
    SSLProxyEngine on

    RewriteEngine on
    RewriteCond %{HTTP:Upgrade} websocket [NC]
    RewriteCond %{HTTP:Connection} upgrade [NC]
    RewriteRule /(.*) wss://172.16.1.19:8080/$1 [P,L]

    ProxyPass / https://172.16.1.19:8080/
    ProxyPassReverse / https://172.16.1.19:8080/

    SSLEngine on
    SSLCertificateFile /opt/meshcentral/meshcentral-data/webserver-cert-public.crt
    SSLCertificateKeyFile /opt/meshcentral/meshcentral-data/webserver-cert-private.key

    ErrorLog ${APACHE_LOG_DIR}/mesh_error.log
    CustomLog ${APACHE_LOG_DIR}/mesh_access.log combined
</VirtualHost>
```

<VirtualHost *:443>

ServerName mesh.modernvet.net

ServerAdmin webmaster@modernvet.net

ProxyPreserveHost On
SSLProxyEngine on

RewriteEngine on
RewriteCond %{HTTP:Upgrade} websocket [NC]
RewriteCond %{HTTP:Connection} upgrade [NC]
RewriteRule /(.*) wss://172.16.1.19:8080/\$1 [P,L]

ProxyPass / https://172.16.1.19:8080/
ProxyPassReverse / https://172.16.1.19:8080/

SSLEngine on
SSLCertificateFile /opt/meshcentral/meshcentral-data/webserver-cert-public.crt
SSLCertificateKeyFile /opt/meshcentral/meshcentral-data/webserver-cert-private.key

ErrorLog \${APACHE_LOG_DIR}/mesh_error.log
CustomLog \${APACHE_LOG_DIR}/mesh_access.log combined
</VirtualHost>

- **ProxyPreserveHost On** : Memastikan header `Host` asli diteruskan ke backend.
SSLProxyEngine on : Mengaktifkan ****proxy** ke backend **HTTPS**** (jika backend-nya HTTPS).
- ****Rewrite dan WebSocket Support****
Mengarahkan permintaan WebSocket ke backend 172.16.1.19:8080 menggunakan protokol ws://. Baris ini penting untuk aplikasi real-time seperti MeshCentral, yang menggunakan WebSocket untuk remote control.
- Certificate MeshCentral ada di **/opt/meshcentral/meshcentral-data**

- Buat virtual host redirect http ke https

```
GNU nano 7.2 meshcentral.conf
VirtualHost *:80>
    ServerName mesh.modernvet.net
    Redirect permanent / https://mesh.modernvet.net/
</VirtualHost>
```

- Aktifkan modul dan restart apache2.service

```
root@modernvet:/etc/apache2/sites-available# a2enmod proxy_wstunnel rewrite
Considering dependency proxy for proxy_wstunnel:
Module proxy already enabled
Module proxy_wstunnel already enabled
Module rewrite already enabled
root@modernvet:/etc/apache2/sites-available# 
root@modernvet:/etc/apache2/sites-available# systemctl restart apache2.service
```

Catatan: Konfigurasi IP, port, dan lain-lain pada dokumen ini dapat disesuaikan dengan kebutuhan masing-masing. Konfigurasi pada dokumen ini ditujukan untuk satu server utama (Master Server). Apabila pembaca memiliki lebih dari satu server dengan konten yang berbeda, mohon untuk menyesuaikan setiap langkah konfigurasi sesuai dengan struktur dan kebutuhan masing-masing server.