LAPORAN FINAL PROJECT

SISTEM OPERASI

"Membangun Web Server Berbasis Linux Menggunakan Apache/Nginx dengan SSL"

Dosen Pengampu : Ferdi Cahyadi, S.Kom., M.Cs.

Disusun Oleh :

Kelompok 5

Anggota Kelompok :

Siti Melka Rehngenana	2401020132
Fitri Arfiana	2401020135
Aisyah Nazwa Ramadhani	2401020136
Christine Simbolon	2401020158
Salsadilla Frisca Anjani	2401020164

PRODI TEKNIK INFORMATIKA

FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN

UNIVERSITAS MARITIM RAJA ALI HAJI

T.A 2024/2025

ABSTRAK

Perkembangan teknologi informasi yang pesat menuntut adanya sistem layanan berbasis web yang andal, aman, dan stabil. Kini Web server menjadi komponen utama dalam penyediaan layanan tersebut, khususnya dalam pengelolaan dan distribusi informasi melalui jaringan internet. Pada proyek akhir mata kuliah Sistem Operasi ini, dilakukan pembangunan web server berbasis sistem operasi Linux menggunakan Apache sebagai web server, serta penerapan Secure Socket Layer (SSL) untuk meningkatkan keamanan komunikasi data antara client dan server.

Sistem web server dibangun pada lingkungan Ubuntu Server yang dijalankan dalam mesin virtual. Apache digunakan sebagai web server utama karena sifatnya yang open source, stabil, serta memiliki dukungan komunitas yang luas. Selain itu, pengamanan server dilakukan dengan penerapan SSL/HTTPS untuk mengenkripsi data yang dikirimkan serta konfigurasi firewall menggunakan UFW (Uncomplicated Firewall) guna membatasi akses jaringan yang tidak diizinkan.

Tujuan dari proyek ini adalah membangun dan mengonfigurasi web server yang dapat diakses melalui protokol HTTPS dengan dukungan firewall untuk meningkatkan keamanan server. Metode yang digunakan meliputi instalasi sistem operasi Ubuntu Server pada mesin virtual, instalasi dan konfigurasi Apache Web Server, penerapan SSL menggunakan sertifikat digital, serta konfigurasi firewall menggunakan UFW. Hasil pengujian menunjukkan bahwa web server dapat berjalan dengan baik, dapat diakses melalui browser klien menggunakan HTTPS, dan terlindungi dari akses yang tidak diinginkan melalui konfigurasi firewall yang tepat. Dengan demikian, proyek ini diharapkan dapat menjadi referensi dasar dalam memahami implementasi web server berbasis Linux yang aman.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Sistem operasi merupakan perangkat lunak inti yang berperan penting dalam mengelola sumber daya komputer dan menjadi penghubung antara perangkat keras dan pengguna. Dalam dunia teknologi informasi, sistem operasi Linux banyak digunakan pada server karena bersifat open source, stabil, aman, dan fleksibel. Salah satu penerapan Linux yang paling umum adalah sebagai web server. Web server berfungsi untuk menyimpan, memproses, dan menyajikan halaman web kepada pengguna melalui jaringan. Apache merupakan salah satu web server paling populer yang banyak digunakan pada sistem operasi Linux.

Selain ketersediaan layanan, aspek keamanan menjadi hal yang sangat penting. Penggunaan protokol HTTP tanpa enkripsi berisiko terhadap penyadapan data. Oleh karena itu, diperlukan penerapan Secure Socket Layer (SSL) untuk mengamankan komunikasi data dengan menggunakan protokol HTTPS. Selain itu, firewall juga diperlukan untuk melindungi server dari akses yang tidak sah. Berdasarkan hal tersebut, proyek ini bertujuan untuk membangun dan mengonfigurasi web server berbasis Linux menggunakan Apache yang dilengkapi dengan SSL dan firewall sebagai bentuk penerapan konsep sistem operasi dan keamanan jaringan.

1.2 RUANG LINGKUP

Ruang lingkup pada proyek ini meliputi :

1. Instalasi Ubuntu Server pada mesin virtual.
2. Instalasi dan konfigurasi Apache Web Server.
3. Penerapan SSL/HTTPS pada web server.
4. Konfigurasi firewall menggunakan UFW.
5. Pengujian akses web server dari sisi klien.

1.3 RUMUSAN MASALAH

Berdasarkan latar belakang tersebut, rumusan masalah dalam proyek ini adalah :

1. Bagaimana cara membangun web server berbasis Linux menggunakan Apache?
2. Bagaimana penerapan SSL untuk meningkatkan keamanan web server?

3. Bagaimana konfigurasi firewall untuk melindungi server dari akses yang tidak sah?

1.4 TUJUAN PROJECT

Tujuan dari pelaksanaan proyek akhir ini adalah :

1. Memahami proses instalasi dan konfigurasi web server berbasis Linux.
2. Mengimplementasikan SSL/HTTPS pada web server.
3. Menerapkan firewall sebagai sistem keamanan server.
4. Menganalisis hasil pengujian web server yang telah dibangun.

BAB II

LANDASAN TEORI

2.1 SISTEM OPERASI LINUX

Linux merupakan sistem operasi open source yang dikembangkan pertama kali oleh Linus Torvalds dan berbasis pada sistem operasi Unix. Linux banyak digunakan pada lingkungan server karena memiliki tingkat stabilitas yang tinggi, aman, serta mampu berjalan dalam waktu lama tanpa perlu reboot. Sistem operasi Linux juga dikenal memiliki manajemen proses dan memori yang baik, sehingga sangat cocok digunakan untuk server dengan beban kerja yang berkelanjutan.

Linux tersedia dalam berbagai distribusi (distro), seperti Ubuntu, Debian, CentOS, Fedora, dan Red Hat. Setiap distribusi memiliki karakteristik masing-masing, namun secara umum memiliki kernel Linux yang sama. Pada proyek ini digunakan Ubuntu Server karena merupakan salah satu distribusi Linux yang populer dan banyak digunakan pada lingkungan server pendidikan maupun industri.

Keunggulan Linux sebagai sistem operasi server antara lain :

1. Bersifat open source sehingga dapat digunakan dan dikembangkan secara bebas.
2. Memiliki tingkat keamanan yang tinggi karena dukungan sistem permission dan user management.
3. Stabil dan jarang mengalami crash.
4. Mendukung berbagai layanan server seperti web server, database server, dan file server.
5. Didukung oleh komunitas besar dan dokumentasi yang lengkap.

Dengan keunggulan tersebut, Linux menjadi pilihan utama sebagai sistem operasi dasar dalam pembangunan web server pada proyek ini

2.2 WEB SERVER APACHE

Apache HTTP Server adalah perangkat lunak web server open source yang dikembangkan oleh Apache Software Foundation. Apache berfungsi untuk menerima permintaan (request) dari klien melalui protokol HTTP atau HTTPS dan mengirimkan respon berupa halaman web. Apache menjadi salah satu web server yang paling banyak digunakan di dunia karena memiliki kestabilan tinggi, fleksibel, dan mendukung berbagai modul tambahan.

Apache dapat berjalan pada berbagai sistem operasi, namun memiliki performa yang sangat baik pada sistem operasi Linux.

Beberapa fungsi utama Apache Web Server adalah :

1. Menyediakan layanan web berbasis HTTP dan HTTPS.
2. Mengelola permintaan klien dan mengirimkan respon berupa konten web.
3. Mendukung konfigurasi virtual host untuk mengelola lebih dari satu website dalam satu server.
4. Mendukung modul keamanan seperti SSL/TLS.

Dalam proyek ini, Apache digunakan sebagai web server utama untuk menampilkan halaman web statis dan mengelola koneksi HTTPS yang aman.

2.3 SECURE SOCKET LAYER (SSL/HTTPS)

Secure Socket Layer (SSL) atau Transport Layer Security (TLS) adalah protokol keamanan yang digunakan untuk mengenkripsi komunikasi data antara klien dan server. SSL memastikan bahwa data yang dikirimkan tidak dapat dibaca atau dimodifikasi oleh pihak yang tidak berwenang. HTTPS merupakan pengembangan dari HTTP yang menggunakan SSL/TLS sebagai lapisan keamanannya. Dengan menggunakan HTTPS, data seperti username, password, dan informasi sensitif lainnya akan dikirimkan dalam bentuk terenkripsi.

Perbedaan utama antara HTTP dan HTTPS adalah :

- HTTP tidak menggunakan enkripsi sehingga data dikirim dalam bentuk teks biasa.
- HTTPS menggunakan enkripsi SSL/TLS sehingga lebih aman dari penyadapan data.

Dalam proyek ini digunakan SSL jenis self-signed certificate yang dibuat secara mandiri untuk keperluan pembelajaran dan pengujian pada jaringan lokal.

Sertifikat SSL self-signed merupakan sertifikat digital yang dibuat dan ditandatangani oleh server itu sendiri tanpa melibatkan Certificate Authority (CA) resmi. Sertifikat jenis ini umumnya digunakan pada lingkungan pengujian atau pengembangan lokal.

Namun, karena sertifikat self-signed tidak berasal dari CA yang dipercaya, browser pada sisi client akan menampilkan peringatan keamanan seperti “Not Secure”. Oleh karena itu, diperlukan proses validasi tambahan di sisi client agar sertifikat tersebut dapat dikenali dan dipercaya oleh sistem operasi dan browser.

2.4 FIREWALL (UFW/IPTABLES)

Firewall merupakan sistem keamanan jaringan yang berfungsi untuk mengontrol lalu lintas data masuk dan keluar dari server. Firewall menentukan port dan layanan mana yang diizinkan atau diblokir.

UFW (Uncomplicated Firewall) adalah firewall bawaan Ubuntu yang dirancang agar mudah digunakan. UFW merupakan antarmuka sederhana dari iptables yang memungkinkan administrator mengatur aturan firewall dengan perintah yang lebih mudah dipahami.

Fungsi utama firewall dalam proyek ini adalah :

1. Mengizinkan akses ke port yang diperlukan seperti SSH (22), HTTP (80), dan HTTPS (443).
2. Menutup akses ke port lain yang tidak diperlukan.
3. Meningkatkan keamanan server dari potensi serangan.

Iptables merupakan firewall tingkat lanjut yang memberikan kontrol lebih detail terhadap lalu lintas jaringan. Namun, dalam proyek ini digunakan UFW karena lebih sederhana dan sesuai untuk pembelajaran dasar. Dengan penerapan firewall, web server yang dibangun menjadi lebih aman dan terlindungi dari akses yang tidak sah.

BAB III

PERANCANGAN ARSITEKTUR

3.1 ARSITEKTUR SISTEM

Arsitektur sistem yang diterapkan pada proyek ini menggunakan model client–server, di mana server bertindak sebagai penyedia layanan web dan client sebagai pengguna yang mengakses layanan tersebut. Server dibangun menggunakan Ubuntu Server 22.04 LTS yang dijalankan pada mesin virtual (VirtualBox), sementara client menggunakan sistem operasi Windows dengan browser sebagai media akses.

Komponen utama dalam arsitektur sistem ini meliputi :

1. Client (Browser Windows)

   Client merupakan perangkat pengguna yang digunakan untuk mengakses website. Browser seperti Google Chrome atau Mozilla Firefox digunakan untuk mengirimkan request ke server melalui protokol HTTP dan HTTPS. Client juga melakukan resolusi domain lokal melalui konfigurasi file hosts.
2. Virtual Machine (Ubuntu Server)

   Ubuntu Server diinstal pada mesin virtual untuk mensimulasikan lingkungan server secara terisolasi. Sistem operasi ini berperan sebagai pengelola sumber daya dan layanan server, termasuk Apache, SSL, dan firewall.
3. Apache Web Server

   Apache berfungsi sebagai web server utama yang menerima permintaan dari client dan menyajikan halaman web. Apache dikonfigurasi untuk melayani:
   - Port 80 (HTTP)
   - Port 443 (HTTPS)

   Selain itu, Apache juga dikonfigurasi menggunakan Virtual Host agar website dapat diakses melalui domain lokal tertentu.
4. SSL/TLS (Secure Socket Layer)

   SSL/TLS diterapkan untuk mengenkripsi komunikasi antara client dan server. Sertifikat yang digunakan adalah self-signed certificate, yang dibuat menggunakan OpenSSL dan digunakan untuk pengujian pada jaringan lokal.
5. Firewall (UFW)

Firewall UFW digunakan sebagai sistem keamanan jaringan untuk mengatur lalu lintas data masuk ke server. Firewall dikonfigurasi agar hanya membuka port yang diperlukan, yaitu port 22 (SSH), 80 (HTTP), dan 443 (HTTPS).

Dengan kombinasi komponen tersebut, arsitektur sistem dirancang untuk menyediakan layanan web yang aman, stabil, dan terkontrol, sesuai dengan tujuan proyek.

3.2 DIAGRAM ARSITEKTUR

Diagram arsitektur sistem menggambarkan alur komunikasi antara client dan server dalam proyek ini. Secara umum, arsitektur sistem dapat direpresentasikan sebagai berikut :

Client (Browser) → HTTPS → Apache Web Server → Website

Penjelasan diagram arsitektur :

- Client mengakses website menggunakan browser.
- Request dikirim melalui protokol HTTPS.
- Firewall UFW memfilter request berdasarkan port yang diizinkan.
- Apache Web Server menerima dan memproses request.
- Website yang tersimpan pada server dikirimkan kembali ke client.

Diagram ini menunjukkan bahwa seluruh komunikasi antara client dan server telah diamankan menggunakan SSL/TLS, serta dilindungi oleh firewall untuk mencegah akses yang tidak sah.

3.3 ALUR KERJA SISTEM

Alur kerja sistem menjelaskan proses yang terjadi mulai dari client mengakses website hingga halaman web ditampilkan. Alur kerja pada sistem ini adalah sebagai berikut :

1. Client membuka browser pada sistem operasi Windows.
2. Client mengetikkan domain lokal website pada address bar browser.
3. Sistem client melakukan resolusi domain melalui file hosts yang telah dikonfigurasi.
4. Browser mengirimkan request ke server melalui protokol HTTPS.
5. Firewall (UFW) memeriksa request dan memastikan port yang digunakan diizinkan.
6. Request diteruskan ke Apache Web Server.
7. Apache memproses request dan mengambil file website dari direktori DocumentRoot.
8. Apache mengirimkan respon berupa halaman web melalui koneksi HTTPS.
9. Browser menerima dan menampilkan halaman web kepada client.

Dengan alur kerja tersebut, sistem memastikan bahwa setiap akses website dilakukan secara aman, terenkripsi, dan sesuai dengan aturan keamanan yang telah ditetapkan.

3.4 KETERKAITAN PERANCANGAN DENGAN IMPLEMENTASI

Perancangan arsitektur sistem ini menjadi dasar pelaksanaan implementasi pada BAB IV. Seluruh komponen yang dirancang telah direalisasikan melalui tahapan berikut :

- Instalasi dan konfigurasi Apache Web Server (Progres 1)
- Konfigurasi Virtual Host dan domain lokal (Progres 1)
- Implementasi SSL/HTTPS menggunakan self-signed certificate (Progres 2)
- Penerapan Firewall UFW dan pengujian keamanan (Progres 4)

Dengan demikian, perancangan arsitektur sistem tidak hanya bersifat konseptual, tetapi telah diwujudkan secara nyata dalam implementasi web server berbasis Linux yang aman.

BAB IV

IMPLEMENTASI

4.1 LINGKUNGAN IMPLEMENTASI SISTEM

Lingkungan implementasi merupakan aspek fundamental dalam pembangunan sistem web server karena menentukan stabilitas, keamanan, dan keberhasilan layanan yang dijalankan. Pada proyek ini, sistem web server dibangun pada lingkungan virtual menggunakan Oracle VirtualBox. Penggunaan virtualisasi bertujuan untuk mensimulasikan kondisi server sesungguhnya tanpa memengaruhi sistem operasi utama pada perangkat pengguna.

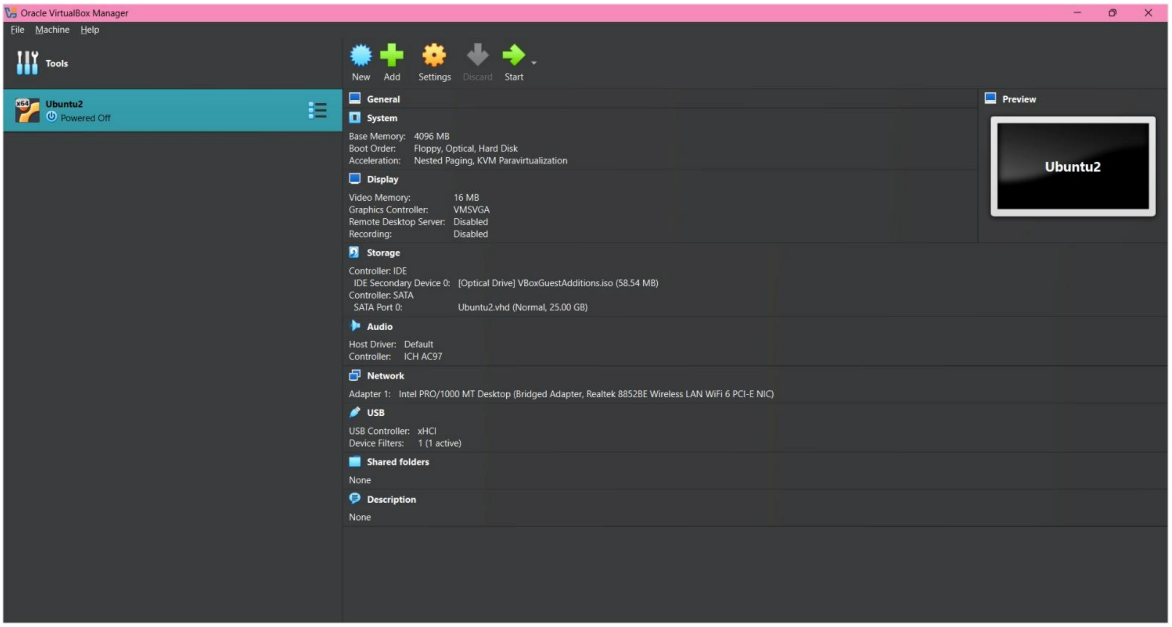
Server menggunakan sistem operasi Ubuntu Server 22.04 LTS yang dijalankan sebagai mesin virtual. Ubuntu Server dipilih karena memiliki tingkat stabilitas yang tinggi, dukungan pembaruan keamanan jangka panjang, serta dokumentasi resmi yang lengkap. Seluruh layanan server, termasuk Apache Web Server, SSL, dan firewall, dikonfigurasi dan dijalankan di dalam sistem ini.

Implementasi dilakukan pada jaringan lokal, di mana sistem client menggunakan sistem operasi Windows dengan browser sebagai media pengujian. Model client–server ini memungkinkan pengujian layanan web secara langsung dan realistis sesuai dengan skenario penggunaan web server pada umumnya.

Dalam proses implementasi, alamat IP server yang digunakan dapat mengalami perubahan. Hal ini disebabkan oleh konfigurasi jaringan pada mesin virtual yang menggunakan mekanisme pemberian alamat IP secara dinamis (DHCP), baik pada mode NAT maupun Bridged Adapter. Perubahan alamat IP juga dapat dipengaruhi oleh pergantian jaringan yang digunakan, seperti jaringan kampus, jaringan rumah, atau hotspot. Meskipun demikian, perubahan alamat IP tersebut tidak mempengaruhi proses instalasi, konfigurasi, maupun fungsi layanan web server yang dijalankan, karena seluruh pengujian selalu dilakukan menggunakan alamat IP terbaru yang diberikan oleh jaringan.

Langkah Implementasi Lingkungan Sistem :

1. Membuat mesin virtual baru menggunakan Oracle VirtualBox.
2. Menginstal sistem operasi Ubuntu Server 22.04 LTS pada mesin virtual.



Gambar 4 . 1 Tampilan Ubuntu Server pada VirtualBox

3. Mengatur konfigurasi jaringan menggunakan mode Bridged Adapter agar server memperoleh alamat IP dari jaringan lokal.

```
christinesimbolon@christinesimbolon-VirtualBox:~$ hostname -I
10.34.126.218
```

Gambar 4 . 2 Informasi IP Address server

Catatan : Perbedaan alamat IP pada dokumentasi disebabkan oleh penggunaan jaringan yang berbeda dan mekanisme DHCP.

4. Melakukan login ke sistem server menggunakan akun administrator.
5. Memverifikasi konektivitas jaringan dengan memastikan server memperoleh IP Address yang valid.

Langkah-langkah tersebut dilakukan untuk memastikan lingkungan server siap digunakan sebelum instalasi layanan utama. Konfigurasi jaringan menjadi aspek penting karena menentukan keberhasilan komunikasi antara server dan client pada jaringan lokal.

4.2 INSTALASI DAN KONFIGURASI WEB SERVER APACHE

Tahap awal implementasi sistem adalah instalasi Apache Web Server. Apache berfungsi sebagai layanan utama yang menangani permintaan dari client dan menyajikan konten web melalui protokol HTTP dan HTTPS. Instalasi Apache dilakukan setelah sistem operasi Ubuntu Server siap digunakan.

Proses instalasi diawali dengan pembaruan repository sistem menggunakan perintah apt update. Langkah ini bertujuan untuk memastikan bahwa paket yang digunakan merupakan versi terbaru dan kompatibel dengan sistem. Setelah itu, Apache diinstal menggunakan package manager bawaan Ubuntu.

Setelah proses instalasi selesai, dilakukan pemeriksaan status layanan Apache menggunakan perintah systemctl status apache2. Pemeriksaan ini penting untuk memastikan bahwa Apache berjalan sebagai service dan siap menerima permintaan dari client. Status active (running) menunjukkan bahwa Apache telah berhasil diinstal dan berjalan dengan baik.

Sebagai tahap verifikasi awal, dilakukan pengujian akses Apache melalui browser client menggunakan alamat IP server. Apabila halaman Apache2 Ubuntu Default Page berhasil ditampilkan pada browser, maka dapat disimpulkan bahwa Apache mampu melayani permintaan HTTP dengan baik.

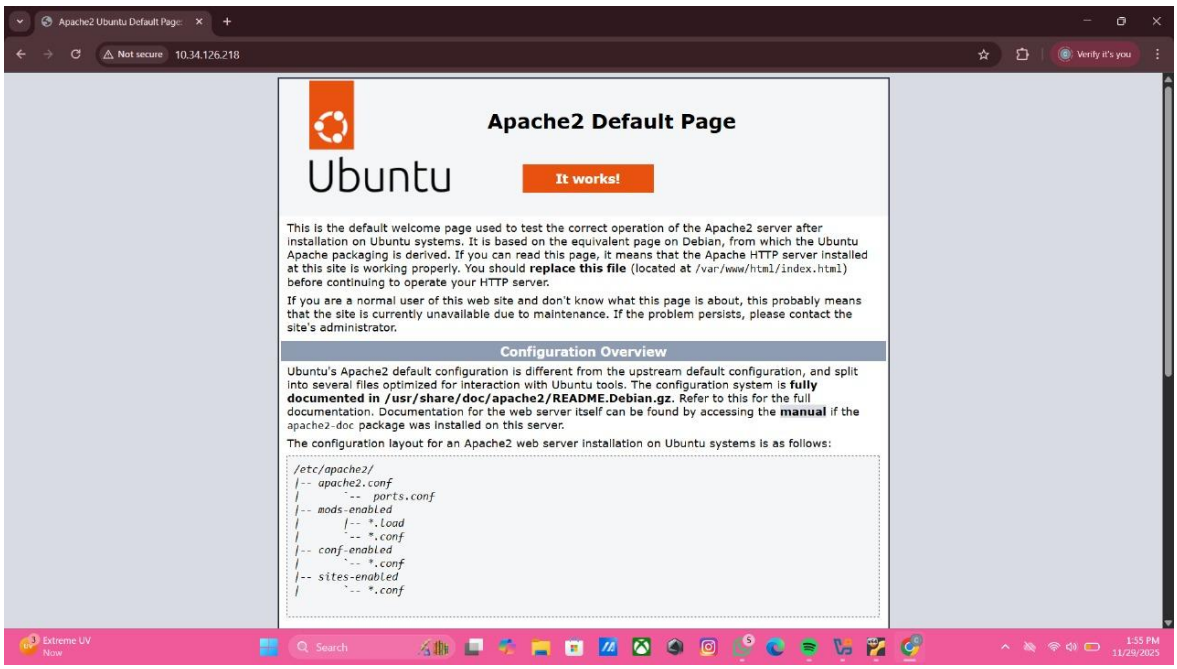
Langkah Implementasi Instalasi Apache :

1. Melakukan pembaruan repository sistem menggunakan perintah apt update.
2. Menginstal Apache Web Server melalui package manager Ubuntu.
3. Sistem menjalankan layanan Apache secara otomatis setelah instalasi selesai.
4. Memeriksa status layanan Apache menggunakan perintah systemctl status apache2.

```
christinesimbolon@christinesimbolon-VirtualBox:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.serveice; enabled; vendor preset:
     Active: active (running) since Tue 2024-11-29 12:45:10 WIB; 5min ago
     Main PID: 1234 (apache2)
     Tasks: 8 (limit: 4672)
     Memory: 15.3M
     CGroup: /system.slice/apache2.service
             ├─ <-1234 /usr/sbin/apache2 -k start
             ├─ <-1278 /usr/sbin/apache2 -k start
             └─ <-1280 /usr/sbin/apache2 -k start

Nov 29 12:45:10 server systemd[1]: Started The Apache HTTP Server.
Nov 29 12:45:10 server apache2[1234]: AH00558: apache2: Could not reliably determine
                                      the server's fully qualified domain name, using
Nov 29 12:45:10 server apache2[1234]: AH00568: ServerName set to 127.0.0.1
Nov 29 12:45:10 server apache2[1234]: AH00163: Apache/2.4.51 (Ubuntu) configured —
                                      resuming normal operations
christinesimbolon@christinesimbolon-VirtualBox:~$
```

Gambar 4 . 3 Status layanan Apache

5. Menguji akses web server melalui browser client menggunakan alamat IP server.



Gambar 4 . 4 Halaman default Apache pada browser client

Tahapan ini memastikan bahwa Apache Web Server terpasang dengan benar dan siap melayani permintaan client.

4.3 KONFIGURASI VIRTUAL HOST / SERVER BLOCK

Setelah Apache berhasil dijalankan, tahap berikutnya adalah konfigurasi Virtual Host. Virtual Host memungkinkan satu web server mengelola website dengan domain yang berbeda secara terpisah. Pada proyek ini, Virtual Host digunakan untuk menggantikan halaman default Apache dengan website kelompok serta mengaksesnya menggunakan domain lokal.

Tahap konfigurasi dimulai dengan pembuatan direktori website pada path /var/www/kelompok5 yang berfungsi sebagai DocumentRoot. Di dalam direktori tersebut dibuat file index.html yang berisi halaman web statis sebagai tampilan utama website kelompok.

Selanjutnya, dibuat file konfigurasi Virtual Host pada direktori /etc/apache2/sites-available. File konfigurasi ini berisi pengaturan ServerName, DocumentRoot, serta direktori log akses dan error. Setelah konfigurasi selesai, Virtual Host diaktifkan menggunakan perintah a2ensite dan Apache direload agar konfigurasi baru dapat diterapkan.

Agar domain lokal dapat dikenali oleh sistem client, dilakukan pengaturan pada file hosts di sistem operasi Windows. File ini berfungsi sebagai DNS lokal yang memetakan domain ke alamat IP server. Setelah konfigurasi selesai, dilakukan pengujian dengan mengakses domain lokal melalui browser client.

Langkah Implementasi Virtual Host :

1. Membuat direktori website pada path /var/www/kelompok5.

```
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo mkdir -p /var/www/kelompok5
```

Gambar 4 . 5 Struktur direktori website

2. Menyusun file index.html sebagai halaman utama website.

```
christinesimbolon-VirtualBox:~$ sudo nano /var/www/kelompok5/index.html
```

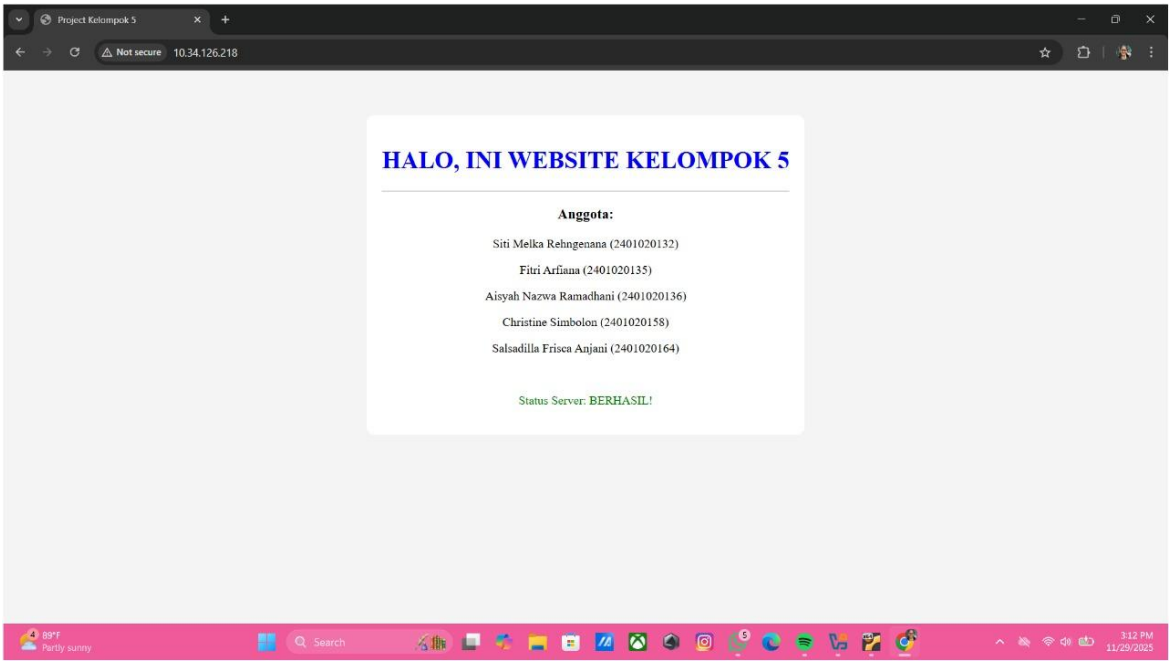
Gambar 4 . 6 Isi file index.html

3. Membuat file konfigurasi Virtual Host pada direktori /etc/apache2/sites-available.

```
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo nano /etc/apache2/sites-available/kelompok5.conf
```

Gambar 4 . 7 File konfigurasi Virtual Host Apache

4. Mengatur parameter ServerName dan DocumentRoot pada file konfigurasi.
5. Mengaktifkan Virtual Host menggunakan perintah a2ensite.
6. Melakukan reload layanan Apache agar konfigurasi diterapkan.
7. Mengatur file hosts pada sistem client untuk pemetaan domain lokal ke IP server.
8. Menguji akses domain lokal melalui browser client.



Gambar 4 . 8 Pengujian akses domain lokal melalui HTTP

Keberhasilan akses domain lokal menunjukkan bahwa konfigurasi Virtual Host telah berjalan sesuai dengan perancangan.

4.4 IMPLEMENTASI KEAMANAN MENGGUNAKAN SSL / HTTPS

Keamanan komunikasi data merupakan aspek penting dalam layanan web, terutama untuk mencegah penyadapan dan manipulasi data. Oleh karena itu, pada proyek ini diterapkan Secure Socket Layer (SSL) untuk mengamankan koneksi antara client dan server melalui protokol HTTPS.

Implementasi SSL diawali dengan mengaktifkan modul SSL pada Apache. Setelah itu, dibuat sertifikat SSL self-signed menggunakan OpenSSL. Sertifikat self-signed dipilih karena sesuai untuk keperluan pembelajaran dan pengujian pada jaringan lokal. Sertifikat ini digunakan untuk mengamankan komunikasi antara client dan server melalui protokol HTTPS. Sertifikat yang dihasilkan berupa file .crt sebagai sertifikat publik dan .key sebagai private key yang disimpan pada direktori sistem.

Setelah sertifikat berhasil dibuat, Apache dikonfigurasi untuk melayani koneksi HTTPS pada port 443. Konfigurasi ini mencakup pemanggilan file sertifikat dan private key pada Virtual Host HTTPS. Apache kemudian direstart untuk menerapkan konfigurasi keamanan tersebut.

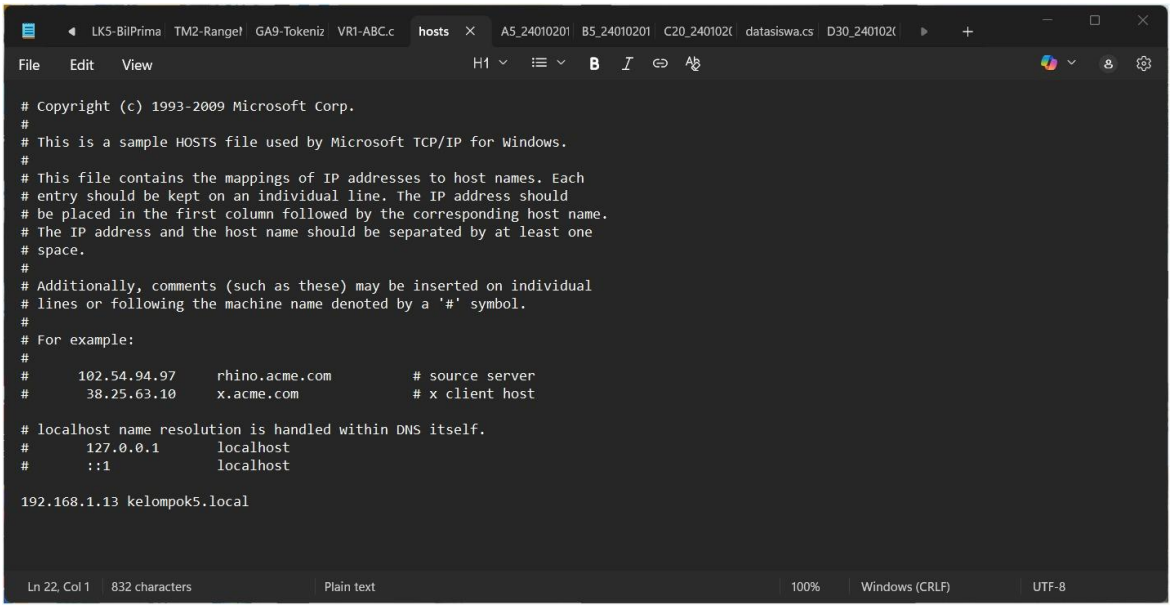
Pengujian dilakukan dengan mengakses website menggunakan protokol HTTPS dari browser client. Meskipun browser menampilkan peringatan keamanan karena sertifikat tidak dikeluarkan oleh Certificate Authority resmi, website tetap dapat diakses dan koneksi telah terenkripsi. Hal ini menunjukkan bahwa SSL telah berfungsi dengan baik dalam mengamankan komunikasi data.

Langkah Implementasi SSL / HTTPS :

1. Mengaktifkan modul SSL pada Apache menggunakan perintah a2enmod ssl.
2. Membuat sertifikat SSL self-signed menggunakan OpenSSL.

Gambar 4 . 9 Proses pembuatan sertifikat SSL self-signed

3. Menyimpan file sertifikat dan private key pada direktori sistem yang aman.
4. Mengonfigurasi Virtual Host HTTPS pada port 443.



Gambar 4 . 10 Konfigurasi Virtual Host HTTPS

5. Menghubungkan file sertifikat dan private key ke konfigurasi Apache.

```
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo cp /etc/ssl/certs/apache-selfsigned.crt /var/www/kelompok5/sertifikat_baru.crt
```

Gambar 4 . 11 Menghubugkan file sertifikat dan private key ke konfigurasi Apache

6. Melakukan restart Apache untuk menerapkan konfigurasi keamanan.

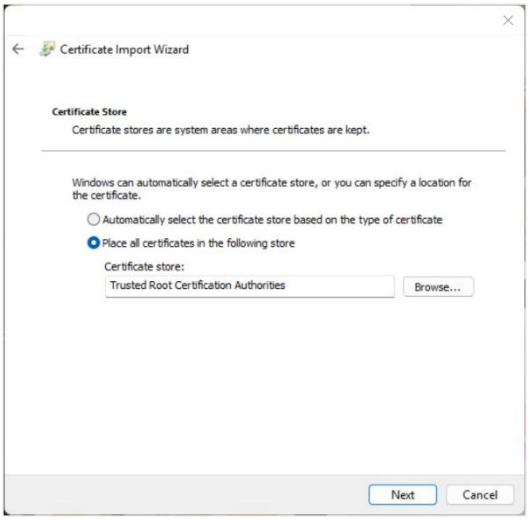
```
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo systemctl restart apache2
```

Gambar 4 . 12 Melakukan Restart Apache untuk menerapkan konfigurasi keamanan

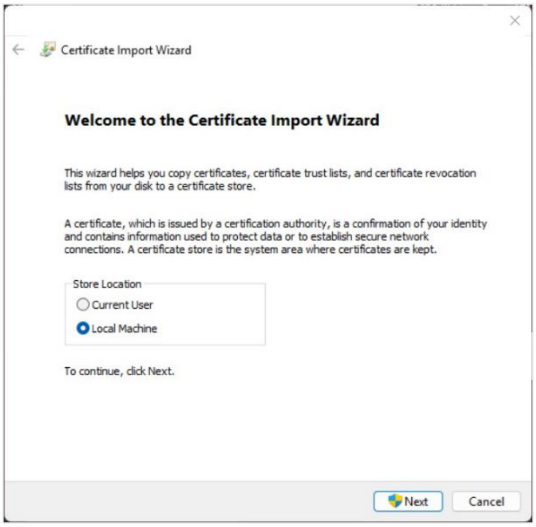
7. Konfigurasi Hardening SSL pada sisi client (Windows).

   Agar browser pada sisi client mempercayai sertifikat Self-Signed yang telah dibuat dan menghilangkan peringatan “Not Secure”, sertifikat publik (*.crt) diekspor dari server dan diinstal ke dalam sistem Windows.

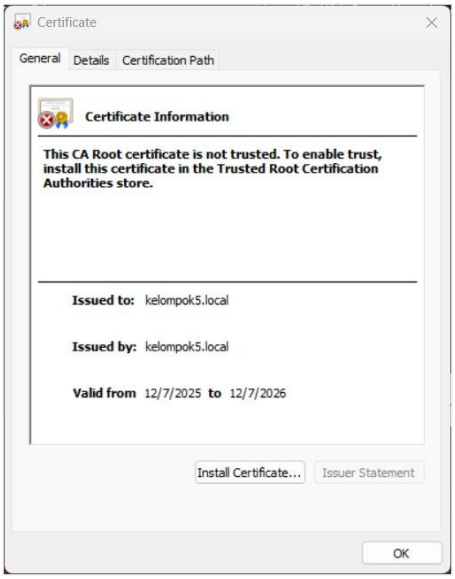
   - Unduh file sertifikat dari server.
   - Lakukan instalasi sertifikat ke dalam Trusted Root Certification Authorities pada Windows.



Gambar 4 . 13 Trusted Root Certification Authorities



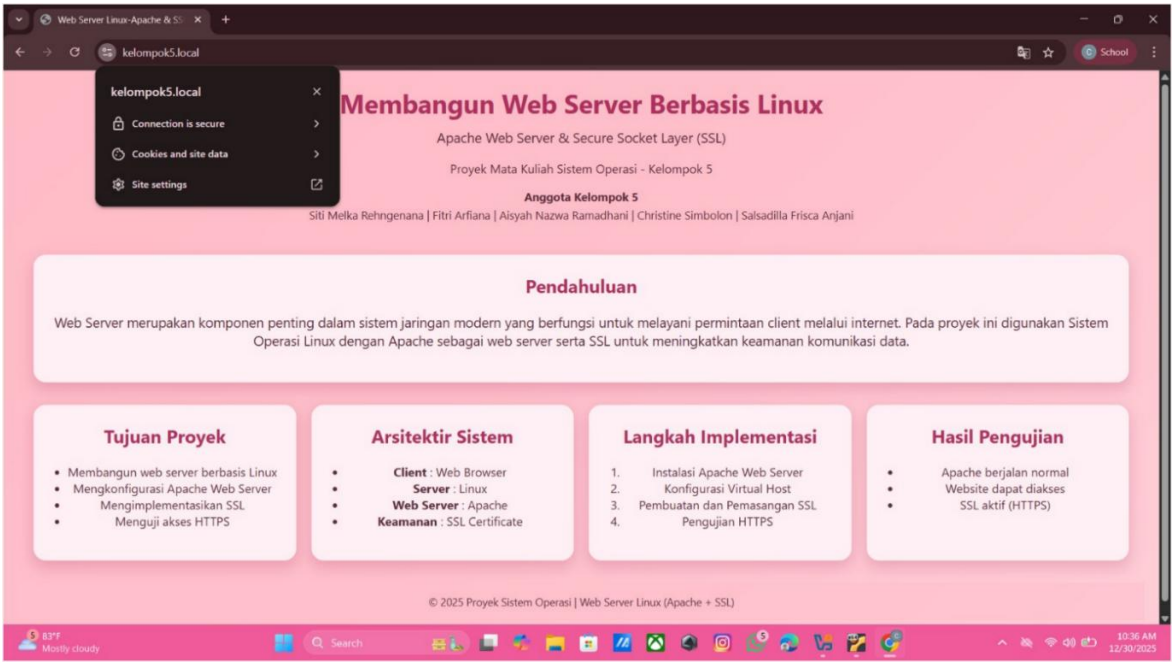
Gambar 4 . 14 Certificate Import Wizard



Gambar 4 . 15 Certificate Information

- Hal ini dilakukan untuk memvalidasi identitas server secara lokal sehingga browser menampilkan indikator “Connection is Secure” (Gembok Hijau).

8. Menguji akses website menggunakan protokol HTTPS melalui browser client.



Gambar 4 . 16 Pengujian akses website menggunakan HTTPS

4.5 RINGKASAN IMPLEMENTASI SISTEM

Berdasarkan seluruh tahapan implementasi yang telah dilakukan, sistem web server berbasis Linux telah berhasil dibangun dan dikonfigurasi sesuai dengan perencanaan. Apache Web Server berjalan dengan stabil, Virtual Host berfungsi dengan baik, SSL berhasil mengamankan koneksi, dan sistem siap untuk diuji lebih lanjut pada tahap pengujian dan analisis.

Bab ini menjadi dasar bagi bab selanjutnya, yaitu Bab V, yang akan membahas secara mendalam hasil pengujian dan analisis terhadap sistem yang telah diimplementasikan.

BAB V

PENGUJIAN DAN ANALISIS

5.1 TUJUAN DAN METODOLOGI PENGUJIAN

Pengujian sistem bertujuan untuk memverifikasi bahwa web server yang dibangun telah berjalan secara fungsional, aman, dan stabil. Selain itu, pengujian juga dilakukan untuk menilai kesesuaian antara hasil implementasi dengan perencanaan yang tertuang dalam proposal dan progres proyek.

Metode pengujian yang digunakan adalah pengujian fungsional (functional testing), yaitu dengan melakukan pengujian langsung terhadap fungsi-fungsi utama sistem dari sisi client. Pengujian dilakukan dalam kondisi sistem sudah berada pada tahap final dan firewall telah diaktifkan.

Tahapan Umum Pengujian Sistem :

1. Memastikan seluruh layanan server berada dalam kondisi aktif.
2. Mengaktifkan firewall UFW sebelum proses pengujian.
3. Melakukan pengujian akses dari sisi client menggunakan browser.
4. Mendokumentasikan hasil pengujian dalam bentuk tabel dan screenshot.
5. Melakukan analisis terhadap hasil pengujian yang diperoleh.

5.2 SKENARIO DAN TABEL PENGUJIAN SISTEM

Untuk mempermudah analisis dan meningkatkan kejelasan hasil pengujian, proses pengujian disajikan dalam bentuk tabel pengujian. Setiap pengujian dikaitkan dengan screenshot sebagai bukti nyata implementasi.

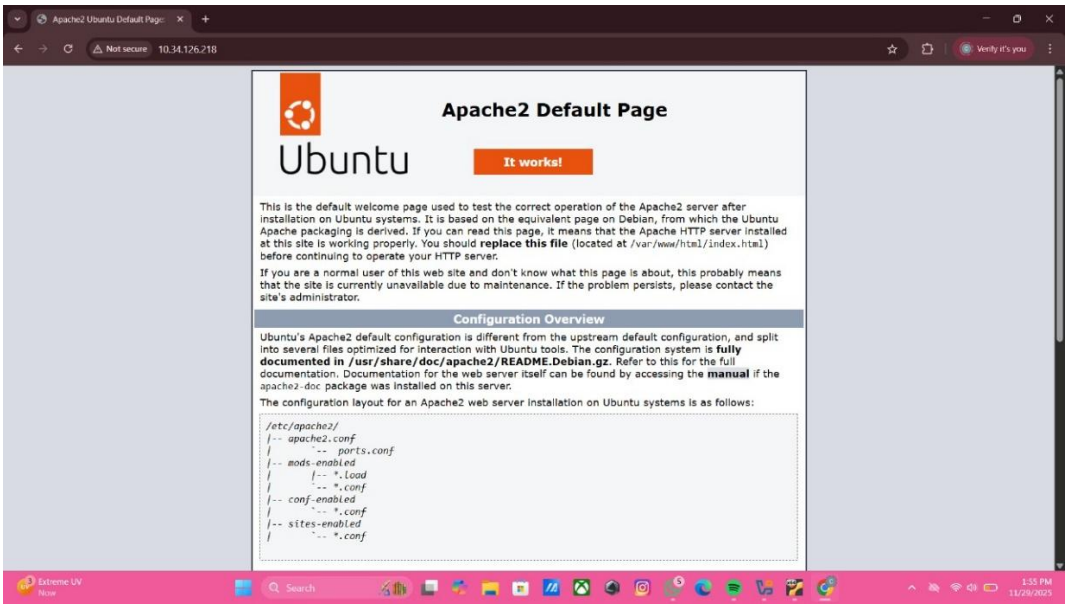
Pengujian layanan web server dilakukan melalui beberapa skenario akses untuk memastikan sistem berfungsi sesuai dengan yang diharapkan.

Langkah Umum Pengujian Layanan Web Server :

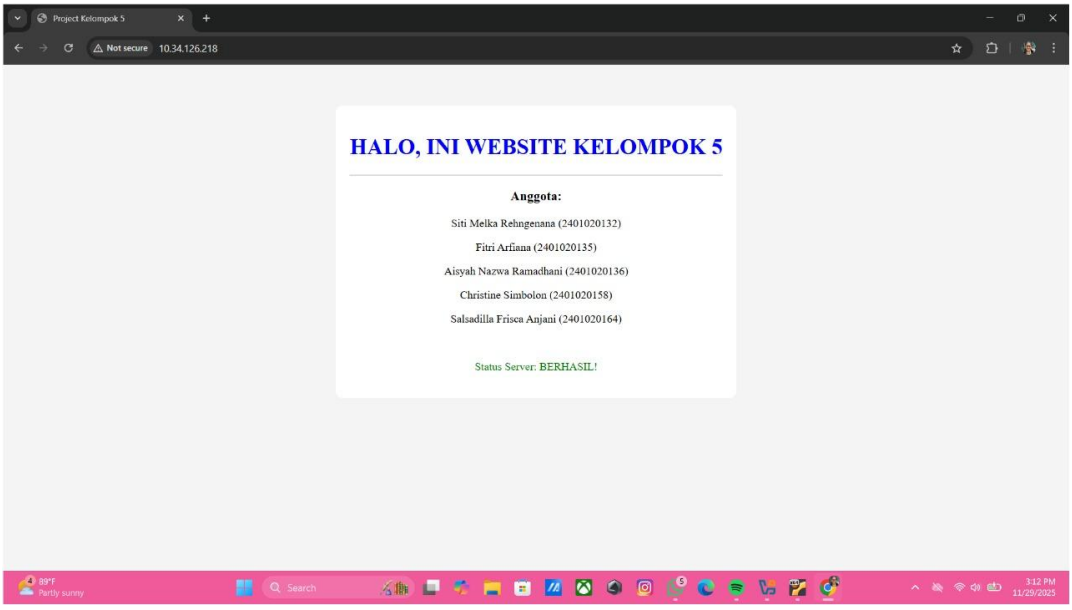
1. Mengakses web server menggunakan protokol HTTP melalui alamat IP.
2. Mengakses website menggunakan domain lokal.
3. Mengakses website menggunakan protokol HTTPS.

No	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Akses web server melalui HTTP	Mengakses http://IP_SERVER melalui browser client	Halaman Apache tampil	Berhasil
2.	Akses domain lokal	Mengakses http://kelompok5.local	Website kelompok tampil	Berhasil

Tabel 5 . 1 Pengujian Layanan Web Server (HTTP)



Gambar 4 . 17 Halaman default Apache pada browser client



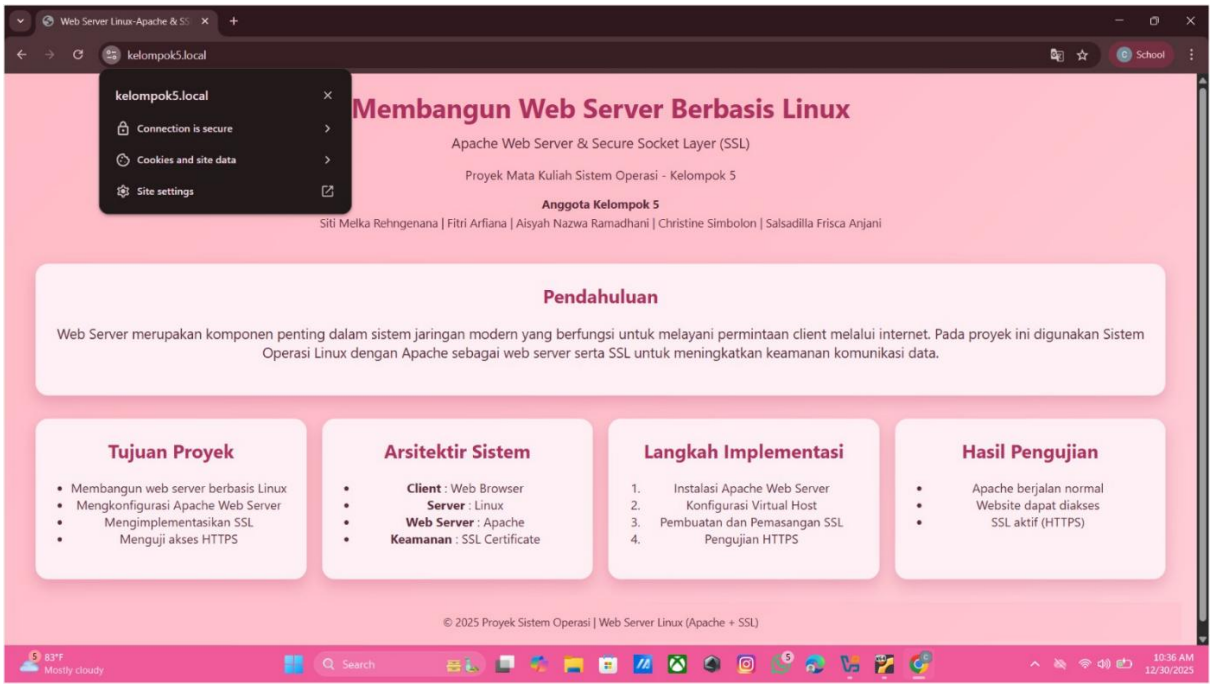
Gambar 4 . 18 Pengujian akses domain lokal melalui HTTP

Analisis :

Hasil pengujian menunjukkan bahwa Apache Web Server mampu melayani permintaan HTTP dengan baik, baik melalui alamat IP maupun domain lokal. Hal ini membuktikan bahwa instalasi Apache dan konfigurasi Virtual Host telah berhasil dilakukan.

No	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Akses website melalui HTTPS	Mengakses https://kelompok5.local	Website tampil dengan koneksi terenkripsi	Berhasil (Indikator Gembok Hijau / Connection is Secure Muncul)
2.	Validasi SSL	Browser menampilkan koneksi HTTPS	SSL aktif (self-signed)	Berhasil

Tabel 5 . 2 Pengujian Layanan Web Server (HTTPS)



Gambar 4 . 19 Pengujian akses website menggunakan HTTPS

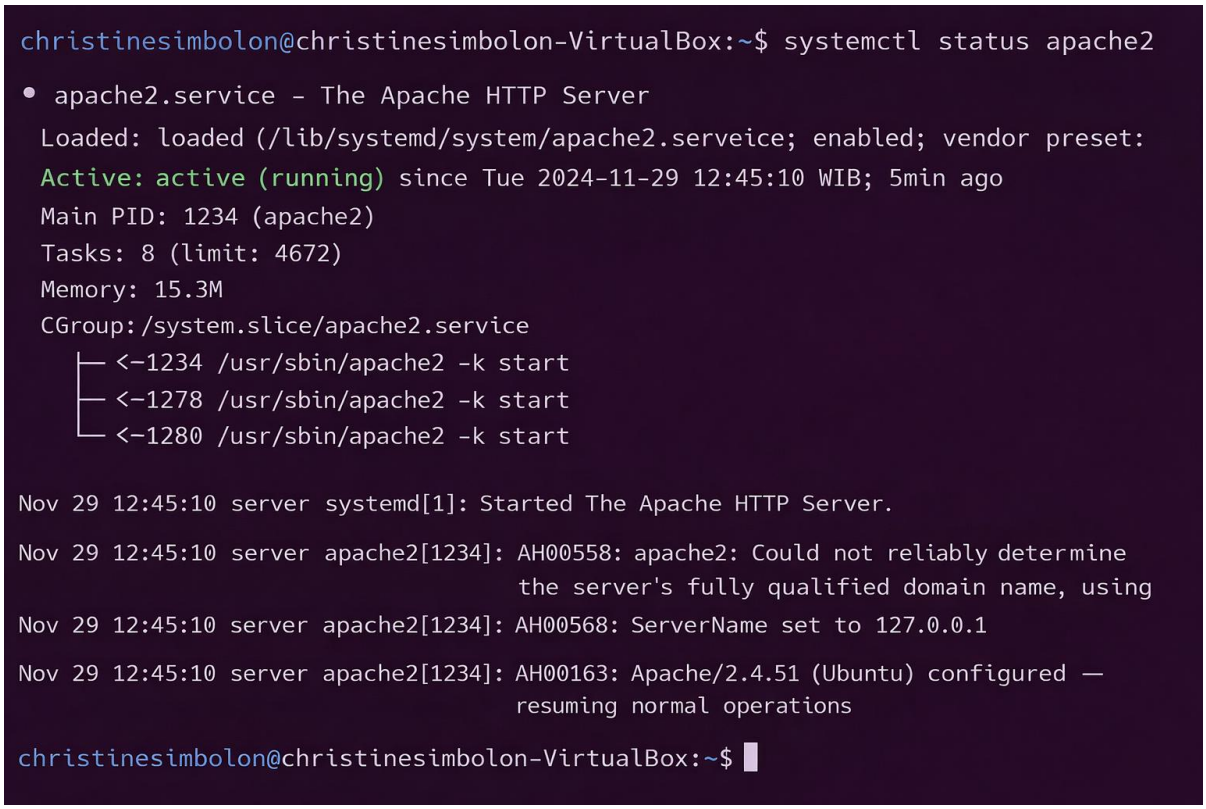
Analisis :

Setelah dilakukan instalasi sertifikat SSL pada sisi client sebagai bagian dari proses hardening, browser berhasil mengenali sertifikat yang digunakan oleh server. Indikator keamanan pada browser menunjukkan status “Connection is secure” yang ditandai dengan munculnya gembok

hijau. Hal ini membuktikan bahwa proses enkripsi SSL/HTTPS telah berjalan dengan baik dan identitas server telah berhasil diverifikasi di sisi client, sehingga komunikasi data dapat dilakukan secara aman tanpa peringatan keamanan.

No	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Status Apache	Menjalankan *systemctl status apache2*	Status active (running)	Berhasil
2.	Stabilitas layanan	Apache tetap berjalan selama pengujian	Tidak terjadi crash	Berhasil

Tabel 5 . 3 Pengujian Status Layanan Apache



Gambar 4 . 20 Status layanan Apache

Analisis :

Apache berjalan stabil selama proses pengujian tanpa mengalami gangguan. Hal ini menunjukkan bahwa konfigurasi Apache telah dilakukan dengan benar dan sistem memiliki kestabilan yang baik.

5.3 PENGUJIAN KEAMANAN MENGGUNAKAN FIREWALL

Pengujian keamanan dilakukan untuk memastikan bahwa firewall UFW berfungsi sesuai dengan perancangan, yaitu membatasi akses jaringan hanya pada port yang diperlukan.

Langkah Pengujian Firewall :

1. Mengaktifkan firewall menggunakan perintah ufw enable.
2. Membuka port 22, 80, dan 443.
3. Memverifikasi status firewall.
4. Menguji akses layanan web setelah firewall aktif.

No	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Aktivasi firewall	Menjalankan *ufw enable*	Firewall aktif	Berhasil
2.	Akses port HTTP	Mengakses port 80	Akses diizinkan	Berhasil
3.	Akses port HTTPS	Mengakses port 443	Akses diizinkan	Berhasil
4.	Akses port lain	Mengakses port selain 22, 80, 443	Akses ditolak	Berhasil

Tabel 5 . 4 Pengujian Firewall (UFW)

```
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Gambar 5 . 1 Status firewall UFW

```
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW IN    Anywhere
443/tcp                    ALLOW IN    Anywhere
22/tcp                     ALLOW IN    Anywhere
80/tcp (v6)                ALLOW IN    Anywhere (v6)
443/tcp (v6)               ALLOW IN    Anywhere (v6)
22/tcp (v6)                ALLOW IN    Anywhere (v6)
```

Gambar 5 . 2 Daftar aturan firewall

Gambar 5 . 3 Akses website setelah firewall aktif

Analisis :

Firewall UFW berhasil diaktifkan dan hanya mengizinkan akses pada port yang telah ditentukan. Website tetap dapat diakses dengan normal meskipun firewall aktif, yang menunjukkan bahwa konfigurasi keamanan tidak mengganggu layanan utama sistem.

5.4 ANALISIS KESELURUHAN HASIL PENGUJIAN

Berdasarkan seluruh pengujian yang telah dilakukan, sistem web server berbasis Linux yang dibangun telah memenuhi seluruh target luaran yang ditetapkan pada proposal proyek. Apache Web Server mampu berjalan secara stabil dan melayani permintaan client dengan baik. Konfigurasi Virtual Host memungkinkan website diakses menggunakan domain lokal, sehingga mempermudah pengelolaan layanan web.

Perbedaan alamat IP server yang terlihat pada beberapa hasil dokumentasi pengujian merupakan kondisi yang wajar dalam lingkungan jaringan dinamis. Alamat IP server diperoleh secara otomatis melalui layanan DHCP sesuai dengan jaringan yang digunakan oleh mesin virtual. Oleh karena itu, perbedaan alamat IP tidak mempengaruhi validitas hasil pengujian, karena fokus pengujian terletak pada keberhasilan layanan Apache, penerapan HTTPS, serta efektivitas firewall dalam mengamankan server.

Implementasi SSL/HTTPS berhasil meningkatkan keamanan komunikasi data melalui mekanisme enkripsi, sedangkan firewall UFW berfungsi sebagai lapisan keamanan tambahan dengan membatasi akses jaringan yang tidak diperlukan. Kombinasi antara web server, SSL,

dan firewall menjadikan sistem lebih aman, stabil, dan sesuai dengan konsep hardening dasar pada sistem operasi Linux.

Dengan demikian, hasil pengujian membuktikan bahwa sistem web server yang dibangun telah berfungsi secara optimal dan layak digunakan sebagai implementasi pembelajaran administrasi server berbasis Linux.

BAB VI

PENUTUP

6.1 KESIMPULAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa konfigurasi SSL/HTTPS pada web server telah berhasil diterapkan dengan baik. Sertifikat SSL self-signed yang digunakan mampu mengamankan komunikasi data antara client dan server melalui enkripsi HTTPS.

Selain itu, melalui proses hardening pada sisi client dengan melakukan instalasi sertifikat ke dalam Trusted Root Certification Authorities, browser berhasil memverifikasi identitas server dan menampilkan indikator keamanan berupa “Connection is secure”. Hal ini menunjukkan bahwa sistem telah memenuhi aspek keamanan komunikasi data secara menyeluruh.

Secara keseluruhan, proyek ini memberikan pemahaman praktis mengenai pengelolaan sistem operasi Linux dalam konteks layanan server, mulai dari instalasi, konfigurasi layanan, hingga penerapan keamanan dasar. Hasil yang dicapai menunjukkan bahwa sistem telah memenuhi seluruh target luaran proyek dan mampu berjalan sesuai dengan konsep client–server yang aman dan terkontrol.

6.2 SARAN PENGEMBANGAN

Untuk pengembangan lebih lanjut, sistem web server yang telah dibangun dapat ditingkatkan dengan menggunakan sertifikat SSL resmi dari Certificate Authority seperti Let’s Encrypt agar koneksi HTTPS dapat dikenali sebagai aman oleh browser tanpa peringatan keamanan. Selain itu, web server dapat diimplementasikan pada jaringan publik atau layanan cloud hosting sehingga dapat diakses secara luas dan diuji pada lingkungan produksi yang sesungguhnya.

Pengembangan selanjutnya juga dapat mencakup penerapan sistem monitoring dan logging untuk memantau kinerja server serta mendeteksi potensi gangguan atau serangan keamanan secara lebih dini. Dari sisi aplikasi, website yang masih bersifat statis dapat dikembangkan menjadi aplikasi web dinamis dengan menambahkan backend dan basis data. Dengan pengembangan tersebut, sistem web server tidak hanya berfungsi sebagai sarana

pembelajaran dasar, tetapi juga dapat menjadi fondasi untuk implementasi layanan web yang lebih kompleks dan profesional.

Untuk pengembangan selanjutnya, disarankan agar sertifikat SSL yang digunakan berasal dari Certificate Authority (CA) resmi agar sistem dapat langsung dipercaya oleh browser tanpa memerlukan proses instalasi sertifikat secara manual pada sisi client.

DAFTAR PUSTAKA

Apache Software Foundation. (2024). *Apache HTTP Server documentation*. https://httpd.apache.org/docs/

Canonical Ltd. (2023). *Ubuntu server documentation*. https://ubuntu.com/server/docs

Kurniawan, E. (2020). *Administrasi server Linux*. Yogyakarta: Andi Offset.

OpenSSL Project. (2023). *OpenSSL documentation*. https://www.openssl.org/docs/

Stallings, W. (2018). *Operating systems: Internals and design principles* (9th ed.). Pearson Education.

Ubuntu Documentation Team. (2023). *Uncomplicated firewall (UFW) documentation*. https://help.ubuntu.com/community/UFW

Widodo, A., & Prasetyo, D. (2021). Implementasi web server berbasis Linux menggunakan Apache dan SSL. *Jurnal Teknologi Informasi dan Komputer*, 9(2), 85–92.

Yuliana, R., & Saputra, A. (2020). Analisis penerapan firewall UFW pada server Linux untuk meningkatkan keamanan jaringan. *Jurnal Informatika dan Sistem Informasi*, 6(1), 45–52.

LAMPIRAN

LAMPIRAN 1 SOURCE CODE TAMPILAN WEB (INDEX.HTML)

Lokasi File : /var/www/kelompok5/index.html

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title>Web Server Linux - Apache & SSL</title>
    <style>
        body {
            margin: 0;
            font-family: "Segoe UI", Arial, sans-serif;
            background: linear-gradient(135deg, #ffb6c1, #ffc0cb, #ffe4e9);
            color: #5a2a3a;
        }

        header {
            padding: 20px 20px;
            text-align: center;
            background: rgba(255, 192, 203, 0.6);
        }

        header h1 {
            margin: 0;
            font-size: 34px;
            color: #b03060;
        }

        header p {
            margin-top: 10px;
            font-size: 18px;
        }

        .badge {
```

```
    display: inline-block;
    margin-top: 15px;
    padding: 8px 20px;
    background: #ffd1dc;
    border-radius: 20px;
    font-weight: bold;
}

.container {
    max-width: 1100px;
    margin: auto;
    padding: 0px 20px;
}

.grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 20px;
    margin-top: 30px;
}

.card {
    background: #fff0f5;
    padding: 25px;
    border-radius: 16px;
    box-shadow: 0 8px 20px rgba(176,48,96,0.25);
}

.card h2 {
    margin-top: 0;
    color: #b03060;
}

ul, ol {
    padding-left: 20px;
}
```

```
        footer {
            text-align: center;
            padding: 15px;
            background: rgba(255, 192, 203, 0.6);
            font-size: 14px;
            margin-top: 30px;
        }
    </style>
</head>
<body>

<header>
    <h1>Membangun Web Server Berbasis Linux</h1>
    <p>Apache Web Server & Secure Socket Layer (SSL)</p>
    <div class="badge">Proyek Mata Kuliah Sistem Operasi — Kelompok 5</div>

    <div style="margin-top: 15px; font-size: 15px; line-height: 1.6;">
          <b>Anggota Kelompok 5</b>
          <br>Siti Melka Rehngenana | Fitri Arfiana | Aisyah Nazwa Ramadhani |
Christine Simbolon | Salsadilla Frisca Anjani</br>
    </div>
</header>
<header>

<div class="container">

    <div class="card">
        <h2>Pendahuluan</h2>
        <p>
            Web server merupakan komponen penting dalam sistem jaringan modern
            yang berfungsi untuk melayani permintaan client melalui internet.
            Pada proyek ini digunakan sistem operasi Linux dengan Apache sebagai
            web server serta SSL untuk meningkatkan keamanan komunikasi data.
        </p>
    </div>

    <div class="grid">
```

```
<div class="card">
    <h2>Tujuan Proyek</h2>
    <ul>
        <li>Membangun web server berbasis Linux</li>
        <li>Mengonfigurasi Apache Web Server</li>
        <li>Mengimplementasikan SSL</li>
        <li>Menguji akses HTTPS</li>
    </ul>
</div>

<div class="card">
    <h2>Arsitektur Sistem</h2>
    <ul>
        <li><b>Client</b> : Web Browser</li>
        <li><b>Server</b> : Linux</li>
        <li><b>Web Server</b> : Apache</li>
        <li><b>Keamanan</b> : SSL Certificate</li>
    </ul>
</div>

<div class="card">
    <h2>Langkah Implementasi</h2>
    <ol>
        <li>Instalasi Apache Web Server</li>
        <li>Konfigurasi Virtual Host</li>
        <li>Pembuatan & pemasangan SSL</li>
        <li>Pengujian HTTPS</li>
    </ol>
</div>

<div class="card">
    <h2>Hasil Pengujian</h2>
    <ul>
        <li>Apache berjalan normal</li>
        <li>Website dapat diakses</li>
        <li>SSL aktif (HTTPS)</li>
    </ul>
```

```
        </div>


    </div>
</div>

<footer>
    © 2025 Proyek Sistem Operasi | Web Server Linux (Apache + SSL)
</footer>

</body>
</html>
```

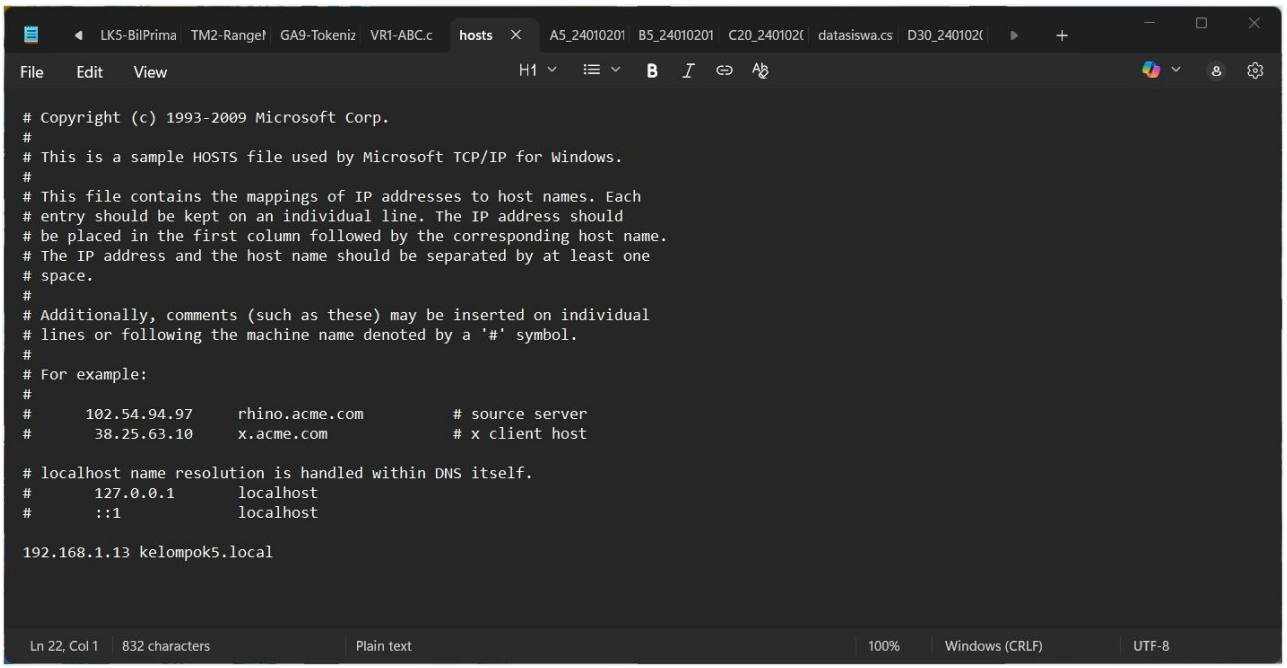
LAMPIRAN 2 KONFIGURASI VIRTUAL HOST

Lokasi File : /etc/apache2/sites-available/kelompok5.conf

```
<VirtualHost *:80>
    ServerAdmin admin@localhost
    ServerName kelompok5.local

    Redirect permanent / https://kelompok5.local/

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

<VirtualHost *:443>
     ServerName kelompok5.local
     DocumentRoot /var/www/kelompok5

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

LAMPIRAN 3 KONFIGURASI DNS LOKAL (HOSTS FILE - CLIENT SIDE)

Lokasi File : C:\Windows\System32\drivers\etc\hosts



LAMPIRAN 4 STATUS KONFIGURASI FIREWALL (UFW)

Perintah : sudo ufw status verbose

```
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo ufw allow 22/tcp
[sudo] password for christinesimbolon:
Skipping adding existing rule
Skipping adding existing rule (v6)
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo ufw allow 80/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo ufw allow 443/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW IN    Anywhere
443/tcp                    ALLOW IN    Anywhere
22/tcp                     ALLOW IN    Anywhere
80/tcp (v6)                ALLOW IN    Anywhere (v6)
443/tcp (v6)               ALLOW IN    Anywhere (v6)
22/tcp (v6)                ALLOW IN    Anywhere (v6)
```

LAMPIRAN 5 LOKASI PENYIMPANAN SERTIFIKAT SSL (BUKTI FILE SISTEM)

Perintah : sudo ls -l /etc/ssl/certs/apache-selfsigned.crt /etc/ssl/private/apache-selfsigned.key

```
christinesimbolon@christinesimbolon-VirtualBox:~$ sudo ls -l /etc/ssl/certs/apache-selfsigned.crt /etc/ssl/private/apache-selfsigned.key
[sudo] password for christinesimbolon:
-rw-r--r-- 1 root root 1578 Dec  7 15:48 /etc/ssl/certs/apache-selfsigned.crt
-rw------- 1 root root 1704 Dec  7 15:46 /etc/ssl/private/apache-selfsigned.key
```