PROPOSAL

Implementasi Web Server Berbasis Linux Menggunakan Nginx dengan Virtual Host, SSL dan Firewall

Disusun untuk Memenuhi Tugas Akhir Mata Kuliah

Sistem Operasi

Dosen Pengampu:

Ferdi Chahyadi, S.Kom., M.Cs

Disusun oleh:

Kelompok Wanmut

2401020001	Amira Azza Nuradiba
2401020006	Zafira Khairunnisa
2401020017	Rashika Kasih Putri
2401020025	Nurul Syafika

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN

UNIVERSITAS MARITIM RAJA ALI HAJI

2025/2026

LEMBAR PENGESAHAN

Implementasi Web Server Berbasis Linux Menggunakan Nginx dengan Virtual Host, SSL dan Firewall

Disusun Oleh:

2401020001 Amira Azza Nuradiba
2401020006 Zafira Khairunnisa
2401020017 Rashika Kasih Putri
2401020025 Nurul Syafika

Laporan ini telah disetujui sebagai Laporan Final Project Mata Kuliah Sistem Operasi.

Dosen Pengampu	Ketua Kelompok
Ferdi Chahyadi, Skom, M.Cs.	**Rashika Kasih Putri**
NIP. 198902222018031001.	NIM:2401020017

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi menuntut tersedianya layanan web yang handal, cepat, dan aman. Salah satu keterampilan dasar yang harus dikuasai mahasiswa Sistem Operasi adalah kemampuan menginstal, mengonfigurasi, dan mengamankan sebuah web server. Sistem operasi Linux merupakan platform yang banyak digunakan dalam dunia industry sebagai server karena stabil, ringan, dan memiliki tingkat keamanan yang tinggi.

Melalui proyek ini, mahasiswa diharapkan mampu memahami proses instalasi web server (Apache atau Nginx), membuat virtual host, menerapkan enkripsi HTTPS menggunakan SSL, serta mengamankan server menggunakan firewall. Proyek ini relevan dalam pembelajaran karena mencakup konsep manajemen layanan, keamanan jaringan, dan pemahaman struktur sistem operasi berbasis Linux.

1.2 Rumusan Masalah

Rumusan masalah dalam proyek ini adalah:

1. Bagaimana melakukan instalasi dan konfigurasi web server berbasis Linux menggunakan Apache atau Nginx?
2. Bagaimana menambahkan dan mengelola virtual host untuk domain lokal maupun domain publik?
3. Bagaimana menerapkan protokol HTTPS menggunakan SSL baik self-signed maupun Let’s Encrypt?
4. Bagaimana melakukan konfigurasi firewall untuk mengamankan akses ke web server?
5. Bagaimana memastikan server berjalan dengan baik melalui proses pengujian HTTP, HTTPS, dan firewall rule?
6. Bagaimana menyusun dokumentasi teknis dan laporan final sesuai standar?

1.3 Tujuan Proyek

Adapun tujuan dari proyek ini adalah:

1. Memahami dan mengimplementasikan web server berbasis Linux (Apache/Nginx).

2. Mengonfigurasi virtual host untuk domain lokal maupun publik.
3. Mengimplementasikan SSL (self-signed atau Let’s Encrypt).
4. Mengonfigurasi firewall pada Linux untuk meningkatkan keamanan server.
5. Melakukan pengujian akses publik, keamanan, dan fungsionalitas web server.
6. Menyusun laporan dan dokumentasi teknis sesuai standar proyek sistem operasi.

1.4 Manfaat

Manfaat yang diharapkan dari proyek akhir ini meliputi:

1. Meningkatkan kemampuan teknis dalam administrasi server Linux.
2. Memahami penerapan web hosting, domain, dan SSL secara nyata.
3. Menguasai konfigurasi keamanan dasar melalui firewall.
4. Menghasilkan proyek yang dapat dijadikan portofolio DevOps/System Administrator.

1.5 Batasan Masalah

Agar proyek tetap sesuai ruang lingkup dan dapat dikerjakan dalam waktu yang tersedia, batasan masalah ditetapkan sebagai berikut:

1. Sistem operasi yang digunakan hanya berbasis Linux (Ubuntu/Debian/CentOS).
2. Web server yang digunakan hanya Apache atau Nginx.
3. SSL yang digunakan hanya self-signed atau Let’s Encrypt (jika domain tersedia).
4. Pengujian dilakukan pada jaringan lokal atau publik sederhana.
5. Firewall yang digunakan hanya UFW atau iptables.
6. Tidak mencakup konfigurasi load balancer, reverse proxy lanjutan, ataupun containerization.
7. Fokus pengujian pada HTTP, HTTPS, serta rule firewall.

1.6 Output Akhir

Output akhir proyek ini sesuai dengan instruksi yaitu:

1. Server web berjalan dengan HTTP dan HTTPS.
2. Virtual host berfungsi dengan domain lokal/publik.
3. Firewall berfungsi sesuai konfigurasi (UFW/iptables).
4. Dokumentasi dan laporan final lengkap.

BAB II
LANDASAN TEORI

2.1 Linux Server

Linux Server adalah sistem operasi open-source yang digunakan secara luas sebagai platform server karena stabil, aman, ringan, dan memiliki komunitas pengembang besar. Beberapa distribusi yang umum digunakan untuk web server antara lain Ubuntu Server, Debian, CentOS, dan Rocky Linux.

Linux menyediakan berbagai fitur yang sangat mendukung layanan web, seperti:

1) Sistem manajemen layanan (systemd)
2) Package manager (APT, YUM, DNF)
3) Tools jaringan (netplan, ifconfig, ip)
4) Firewall bawaan seperti UFW atau iptables
5) File system yang stabil dan aman

Dengan karakteristik ini, Linux menjadi pilihan utama sebagai sistem operasi web server.

2.2 Web Server

Web server adalah aplikasi yang bertugas menerima permintaan HTTP/HTTPS dari klien (browser) dan memberikan respon berupa halaman web.

2.2.1 Apache HTTP Server

Apache adalah web server yang paling banyak digunakan di dunia. Kelebihannya adalah:

1. Konfigurasi fleksibel dengan file *.conf*
2. Mendukung berbagai modul tambahan (mod_ssl, mod_rewrite, dll.)
3. Mudah digunakan untuk virtual host
4. Berbasis proses (MPM Prefork/Worker/Event)

2.2.2 Nginx

Nginx adalah web server modern yang ringan dan cepat. Keunggulannya:

1. Arsitektur event-driven, efisien menangani banyak koneksi
2. Cocok untuk server statis dan reverse proxy
3. Resource lebih ringan dibanding Apache
4. Performa tinggi dan stabil

Kedua web server tersebut didukung penuh pada sistem operasi Linux dan dapat dikombinasikan dengan SSL untuk keamanan layanan.

2.3 Virtual Host

Virtual host adalah mekanisme yang memungkinkan satu server melayani beberapa domain atau subdomain secara bersamaan.

Contohnya:

- `example.local`
- `app.local`

Virtual host digunakan untuk memisahkan konfigurasi website dalam satu server fisik. Pada Apache, virtual host diatur melalui file:

`/etc/apache2/sites-available/*.conf`

Pada Nginx:

`/etc/nginx/sites-available/*.conf`

Virtual host sangat penting dalam deployment website profesional.

2.4 SSL (Secure Socket Layer) dan HTTPS

2.4.1 SSL (Secure Socket Layer)

SSL adalah protokol keamanan yang mengenkripsi koneksi antara server dan klien. Implementasinya menggunakan sertifikat digital.

Fungsi SSL:

1) Melindungi data selama transmisi
2) Menghindari penyadapan (sniffing)
3) Mengautentikasi identitas server
4) Mengaktifkan protokol HTTPS

Jenis Sertifikat SSL:

1. **Self-Signed SSL**

   Dibuat oleh administrator sendiri, tidak divalidasi pihak ketiga. Cocok untuk pembelajaran dan server lokal.

2. **Let’s Encrypt SSL**

   Sertifikat gratis yang valid secara global. Cocok untuk hosting publik.

2.4.2 HTTPS

HTTPS adalah versi aman dari HTTP. Browser menampilkan ikon “gembok” jika sertifikat SSL valid.

2.5 Firewall UFW dan Iptables

Firewall adalah sistem yang mengatur izin akses jaringan berdasarkan aturan tertentu.

A. UFW (Uncomplicated Firewall)

Firewall sederhana dan mudah digunakan pada Ubuntu/Debian

Contoh rule:

- `sudo ufw allow 80`
- `sudo ufw allow 443`

B. Iptables

Firewall tingkat lanjut yang mengatur paket secara detail

Contoh rule:

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Firewall sangat penting untuk mengamankan server dari serangan dan akses ilegal.

BAB III
METODOLOGI PERANCANGAN

3.1 Pendekatan Proyek

Proyek ini menggunakan pendekatan praktikum terstruktur yang terdiri dari perancangan, implementasi, pengujian, dan dokumentasi. Setiap tahap dilaksanakan berurutan namun iteratif: bila ditemukan masalah pada tahap pengujian, tim kembali ke tahap implementasi/perbaikan sampai memenuhi indikator keberhasilan.

Metode kerja:

1. Agile mini-sprint selama 1 minggu per fase (minggu 11–15).
2. Kolaborasi**:** penggunaan version control (Git) untuk manajemen konfigurasi dan dokumentasi.
3. Dokumentasi berkelanjutan: setiap langkah difoto (screenshot), dicatat perintah, dan log disimpan.

3.2 Tahapan Pengerjaan Proyek

1. **Perancangan Web Server**
   - Menentukan OS Linux yang digunakan
   - Menentukan Web Server (Apache atau Nginx)
   - Mendesain struktur direktori web
   - Menentukan domain local
2. **Instalasi Linux Server**
   - Update repository
   - Instalasi Apache/Nginx
   - Konfigurasi IP statis
   - Pengujian dasar akses halaman default
3. **Konfigurasi Virtual Host**
   - Membuat direktori website
   - Membuat file .conf untuk domain
   - Mengaktifkan virtual host
   - Pengujian pada browser

4. **Konfigurasi SSL (HTTPS)**
    - Membuat sertifikat self-signed
    - Mengonfigurasi SSL pada Apache/Nginx
    - Mengaktifkan port 443
    - Pengujian akses melalui HTTPS
5. **Konfigurasi Firewall**
    - Mengizinkan port 22, 80, 443
    - Mengaktifkan firewall
    - Memverifikasi rule keamanan
6. **Pengujian Layanan**
    - HTTP dan HTTPS
    - Error log
    - Ketersediaan layanan
    - Rule firewall
7. **Dokumentasi Laporan**
    - Screenshot konfigurasi
    - Penjelasan teknis
    - Kesimpulan & rekomendasi

3.3 Jadwal Pelaksanaan

Proyek ini dilaksanakan selama 5 minggu dengan tahapan sebagai berikut:

Minggu	Kegiatan
11	Membuat proposal
12	Instalasi Linux Server, Apache/Nginx, domain lokal, SSL self-signed
13	Konfigurasi virtual host + HTTPS, penataan struktur web
14	Pengujian (akses publik, firewall, load test kecil)
15	Dokumentasi final & presentasi

3.4 Perangkat dan Kebutuhan Sistem

1. **Perangkat Keras**
    a. Laptop/PC

b. RAM minimal 4–8 GB
c. Jaringan LAN/WiFi
d. Storage SSD/HDD

2. **Perangkat Lunak**
   a. Linux Server (Ubuntu/Debian)
   b. Apache atau Nginx
   c. OpenSSL / Certbot
   d. UFW/Iptables
3. **Kebutuhan Jaringan**
   a. IP statis (opsional)
   b. Port 80 dan 443 terbuka

3.5 Pengumpulan Data

Data dikumpulkan melalui:

1. Observasi selama instalasi
2. Logs dari server
3. Screenshot konfigurasi
4. Pengujian HTTP/HTTPS
5. Catatan error dan perbaikan

3.6 Indikator Keberhasilan

Proyek dianggap berhasil jika:

1. Server dapat diakses melalui HTTP
2. Server dapat diakses melalui HTTPS
3. Virtual host berjalan sesuai domain
4. Firewall berfungsi dan memblokir port yang tidak diizinkan
5. Laporan akhir tersusun lengkap

BAB IV
IMPLEMENTASI

Pada tahap proposal ini, implementasi belum dilakukan. Oleh karena itu, bagian ini hanya menjelaskan apa saja langkah implementasi yang akan dilakukan ketika proyek mulai dikerjakan, yaitu:

1. **Instalasi Server**

   Menyiapkan Linux Server, melakukan update, dan memasang Apache atau Nginx. Pengujian dilakukan dengan membuka halaman default melalui browser.
2. **Konfigurasi Virtual Host**

   Membuat direktori website, membuat file konfigurasi virtual host, dan menguji domain lokal menggunakan file /etc/hosts.
3. **Konfigurasi SSL**
   - Membuat self-signed SSL menggunakan OpenSSL
   - Atau memasang Let’s Encrypt jika domain publik tersedia
   - Mengaktifkan virtual host HTTPS dan menguji koneksi aman
4. **Firewall**

   Mengizinkan port 22, 80, 443, dan memblokir port yang tidak diperlukan menggunakan UFW atau iptables.
5. **Rencana Pengujian**

   Pengujian meliputi:
   - Akses HTTP/HTTPS
   - Pengujian virtual host
   - Cek firewall
   - Test sederhana menggunakan ab atau curl

BAB V
HASIL DAN PEMBAHASAN

Karena implementasi belum dilakukan, maka pada BAB ini berisi prediksi hasil dan rencana bagaimana hasil tersebut akan dianalisis**.**

5.1 Hasil Yang Diharapkan

Setelah implementasi, diharapkan:

1. Server dapat diakses melalui HTTP dan HTTPS.
2. Virtual host menampilkan website sesuai domain.
3. Sertifikat SSL berfungsi (meski self-signed menampilkan peringatan).
4. Firewall bekerja dan port lain tertutup.
5. Server mampu menangani permintaan dasar tanpa error.

5.2 Rencana Pembahasan

Pembahasan akan meliputi:

1. Keberhasilan konfigurasi web server
2. Fungsi SSL dan keamanan HTTPS
3. Efektivitas firewall
4. Stabilitas server berdasarkan uji sederhana
5. Kendala yang muncul dan solusi yang diterapkan

5.3 Evaluasi Keberhasilan

Keberhasilan proyek dinilai dari:

1. Server berfungsi tanpa error
2. HTTPS aktif
3. Firewall menerapkan aturan dengan benar
4. Dokumentasi lengkap dan sesuai prosedur

BAB VI
PENUTUP

6.1 Kesimpulan

Berdasarkan rancangan dan penyusunan proposal proyek ini, dapat disimpulkan bahwa pembangunan web server berbasis Linux dengan konfigurasi Apache/Nginx, virtual host, SSL, dan firewall merupakan proses yang penting dalam memahami dasar administrasi server modern. Proyek ini dirancang untuk memberikan pengalaman langsung kepada mahasiswa dalam mengelola layanan web, meningkatkan keamanan menggunakan HTTPS, serta menerapkan mekanisme perlindungan server melalui firewall.

Melalui tahapan yang telah direncanakan dalam proposal, mahasiswa diharapkan mampu menguasai proses instalasi server, konfigurasi domain lokal, pembuatan sertifikat SSL, pengamanan akses jaringan, dan pengujian layanan web secara mandiri. Dengan demikian, proyek ini menjadi sarana pembelajaran praktis yang mendukung kompetensi dalam bidang sistem operasi dan administrasi jaringan.