PERTEMUAN 10 · SUB-CPMK 3.2

Konfigurasi Web Server Apache & Nginx

Arsitektur, virtual host, SSL/TLS dasar, dan tuning performa

Administrasi Sistem (CEN3303) · D3 Teknik Komputer · Politeknik Negeri Padang
Ir. H. A. Mooduto, M.Kom. · Semester Ganjil 2026/2027

Target hari ini

Sub-CPMK 3.2

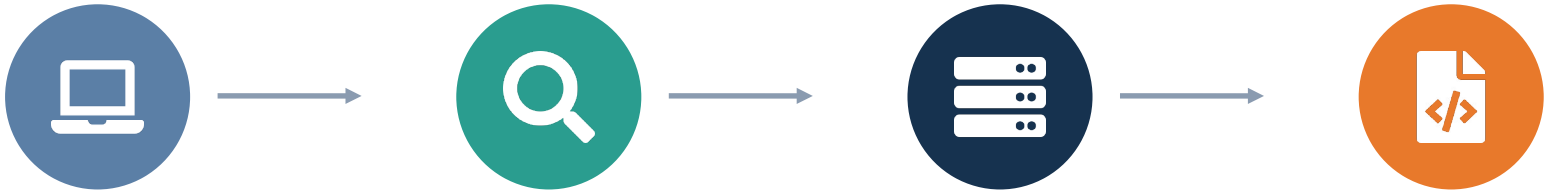
Mahasiswa mampu mengonfigurasi web server (Apache/Nginx) dasar untuk hosting layanan internal.

- Memahami arsitektur & cara kerja web server
- Mengonfigurasi virtual host
- Menjelaskan SSL/TLS & tuning dasar

Alur pertemuan

1	Cara kerja & arsitektur	25'
2	Virtual host & konfigurasi	35'
3	SSL/TLS & tuning	20'
4	Troubleshoot & studi kasus	20'
5	Praktikum lab	170'

Apa yang terjadi saat kita mengetik URL?



Browser

IP dari DHCP
(Pertemuan 9)

DNS

portal.lab.pnp
→ 192.168.10.20

Web server

baca Host header,
pilih situs

Respons

HTML · CSS · JSON
+ status code

Kode status yang paling sering ditemui admin

200	301	403	404	502
OK	Moved Permanently	Forbidden	Not Found	Bad Gateway

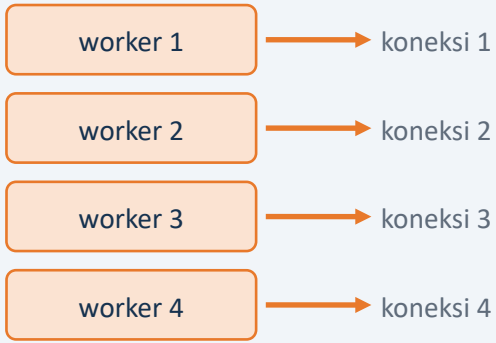
01

Arsitektur: Apache vs Nginx

Dua web server paling populer, dua filosofi penanganan koneksi yang berbeda

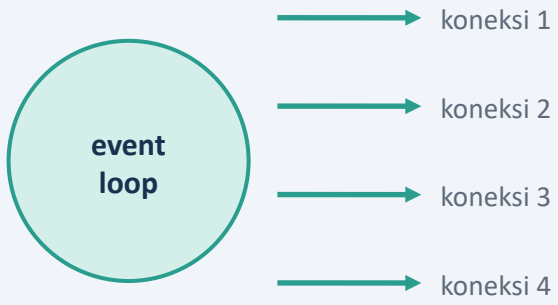
Dua model penanganan koneksi

Apache — process/thread



1 koneksi = 1 worker. Mudah dipahami, tetapi memori ikut naik seiring jumlah koneksi.

Nginx — event-driven



Sedikit worker melayani ribuan koneksi secara non-blocking — hemat memori saat trafik tinggi.

Kapan memilih Apache, kapan Nginx?

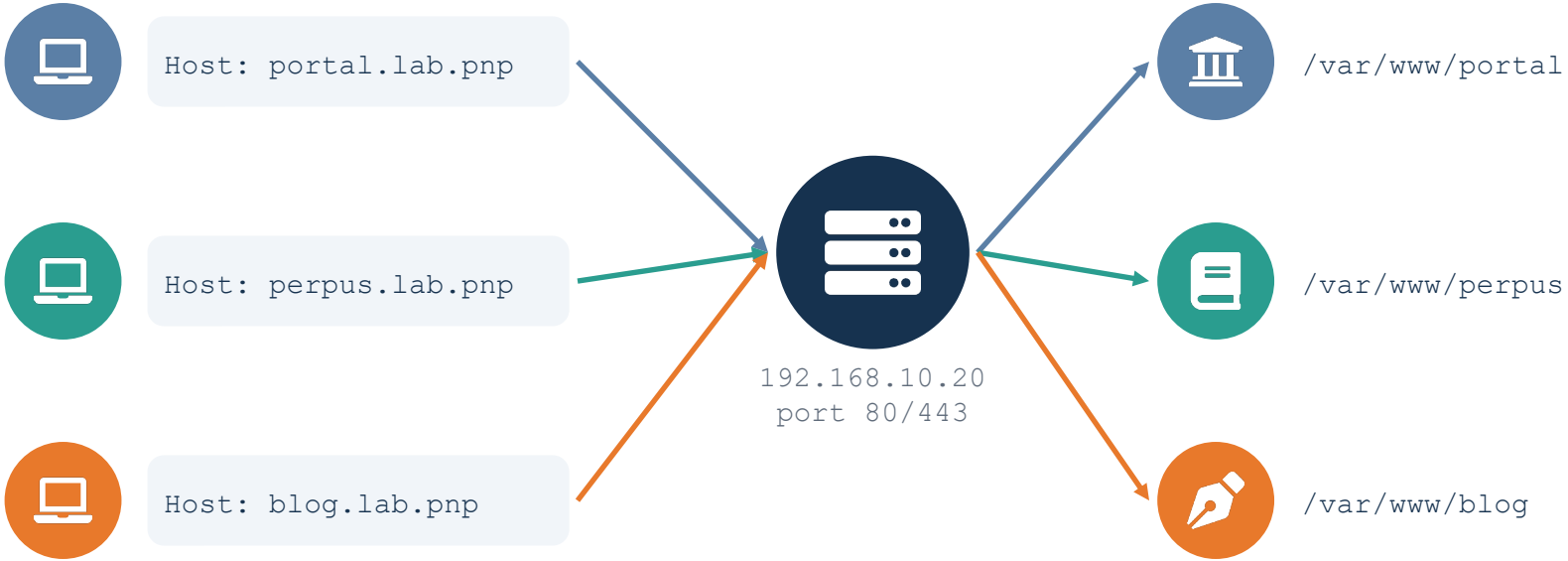
Aspek	Apache HTTP Server	Nginx
Model	Process/thread per koneksi (MPM)	Event-driven asynchronous
Konfigurasi per-direktori	.htaccess didukung	Tidak ada .htaccess
Beban sangat tinggi	Cenderung lebih berat	Lebih ringan & efisien
Konten dinamis (PHP)	Bisa langsung (mod_php)	Lewat PHP-FPM / backend
Peran umum	Shared hosting, aplikasi lama	Reverse proxy, load balancer, situs trafik tinggi
Perintah layanan (Ubuntu)	systemctl reload apache2	systemctl reload nginx

Tidak harus memilih salah satu: keduanya sering dipakai bersama — Nginx di depan, Apache di belakang (studi kasus).

Peta berkas penting di Ubuntu Server

	Nginx	Apache
Konfigurasi utama	`/etc/nginx/nginx.conf`	`/etc/apache2/apache2.conf`
Situs tersedia	`/etc/nginx/sites-available/`	`/etc/apache2/sites-available/`
Situs aktif (symlink)	`/etc/nginx/sites-enabled/`	`/etc/apache2/sites-enabled/`
Document root	`/var/www/<situs>/`	`/var/www/<situs>/`
Log akses & error	`/var/log/nginx/`	`/var/log/apache2/`
User layanan	`www-data`	`www-data`

Name-based virtual host: 1 IP, banyak situs



Server mencocokkan header Host dengan server_name / ServerName untuk menentukan document root mana yang dilayani.

Membuat virtual host di Nginx

```
# /etc/nginx/sites-available/portal
server {
    listen 80;
    server_name portal.lab.pnp;
    root /var/www/portal;
    index index.html;

    access_log /var/log/nginx/portal.access.log;
    error_log  /var/log/nginx/portal.error.log;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Langkah aktivasi

1 Buat document root

```
sudo mkdir -p /var/www/portal
```

2 Aktifkan situs

```
sudo ln -s
/etc/nginx/sites-available/portal \
  /etc/nginx/sites-enabled/
```

3 Uji sintaks

```
sudo nginx -t
```

4 Terapkan

```
sudo systemctl reload nginx
```

Padanannya di Apache

```
# /etc/apache2/sites-available/perpus.conf
<VirtualHost *:80>
    ServerName  perpus.lab.pnp
    DocumentRoot /var/www/perpus

    <Directory /var/www/perpus>
        AllowOverride All
        Require all granted
    </Directory>

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

Aktivasi (gaya Debian)

```
sudo a2ensite perpus
sudo a2dissite 000-default
sudo apache2ctl configtest
sudo systemctl reload apache2
```

Padanan

server_name → ServerName
root → DocumentRoot
nginx -t → apache2ctl configtest

Menghubungkan nama domain ke web server

Cara produksi: DNS server

Tambahkan record di zone file dari Pertemuan 7:

```
web      IN  A      192.168.10.20
portal   IN  CNAME  web
perpus   IN  CNAME  web
blog     IN  CNAME  web
```

Satu A record + beberapa CNAME → semua nama menuju server yang sama.

Cara cepat uji lab: hosts

Linux /etc/hosts · Windows C:
\Windows\System32\drivers\etc\hosts

```
192.168.10.20  portal.lab.pnp
192.168.10.20  perpus.lab.pnp
192.168.10.20  blog.lab.pnp
```

Hanya berlaku di komputer itu sendiri — cocok untuk menguji sebelum DNS siap, bukan pengganti DNS.

SSL/TLS: mengamankan HTTP menjadi HTTPS

Kerahasiaan

Data terenkripsi — kata sandi tidak bisa disadap

Integritas

Isi halaman tidak bisa diubah di tengah jalan

Autentikasi

Sertifikat membuktikan identitas server

Lab: sertifikat self-signed · Produksi: Let's Encrypt (gratis, otomatis)

```
# Lab: buat sertifikat self-signed
sudo openssl req -x509 -nodes -days 365 \
  -newkey rsa:2048 \
  -keyout /etc/ssl/private/lab.key \
  -out /etc/ssl/certs/lab.crt

# Nginx
server {
  listen 443 ssl;
  server_name portal.lab.pnp;
  ssl_certificate     /etc/ssl/certs/lab.crt;
  ssl_certificate_key /etc/ssl/private/lab.key;
  ssl_protocols TLSv1.2 TLSv1.3;
}
# Produksi: sudo certbot --nginx -d contoh.ac.id
```

Tuning dasar tanpa menambah hardware

Kompresi gzip

Respons teks diperkecil hingga ±70% → halaman lebih cepat dimuat

Cache konten statis

Browser menyimpan gambar/CSS/JS, tidak perlu unduh ulang

Worker & koneksi

Sesuaikan jumlah worker dengan jumlah core CPU

Load balancing

Bila satu server tak cukup, bagi beban ke beberapa backend

```
# /etc/nginx/nginx.conf (potongan)
worker_processes auto;
events { worker_connections 1024; }

http {
  gzip on;
  gzip_types text/css application/javascript
             application/json;
  keepalive_timeout 65;
}

# di dalam server { }
location ~* \.(png|jpg|css|js)$ {
  expires 7d;
}
```

Verifikasi & troubleshooting

Layanan & port

```
systemctl status nginx
sudo ss -tlnp | grep -E ':80|:443'
```

Uji dari klien

```
curl -I http://portal.lab.pnp
curl -kI https://portal.lab.pnp
```

Uji vhost tanpa DNS

```
curl -H "Host: blog.lab.pnp" \
  http://192.168.10.20
```

Baca log

```
sudo tail -f /var/log/nginx/error.log
journalctl -u nginx --since today
```

403 permission/owner direktori

404 root atau nama berkas salah

502 backend mati / salah alamat

bind() failed port 80 dipakai layanan lain

Studi kasus: e-commerce melambat saat promo

Situs berbasis Apache melambat drastis saat lonjakan pengunjung, padahal spesifikasi server memadai.



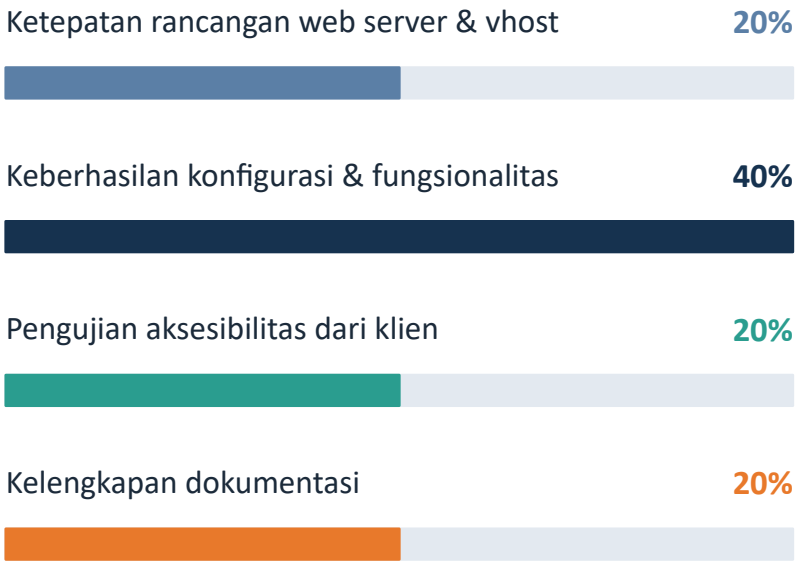

```
location /static/ { root /var/www/toko; expires 7d; }
location / {
    proxy_pass http://127.0.0.1:8080;
    proxy_set_header Host $host;
}
```

Praktikum & penilaian (170 menit)

Tugas lab

1. Instal Nginx di VM server, buka port 80/443 di firewall
1. Buat 3 virtual host: portal, perpus, blog
1. Daftarkan namanya di DNS server Pertemuan 7
1. Aktifkan HTTPS (self-signed) untuk portal + redirect 80→443
1. Uji dari klien (browser & curl), dokumentasikan hasil

Rubrik Pertemuan 10 · bobot 2,5% nilai akhir



Cek pemahaman

1	**Model penanganan koneksi Nginx?**	Event-driven asynchronous
2	**Fitur agar satu server melayani banyak domain?**	Virtual host
3	**Penerbit sertifikat SSL gratis & otomatis?**	Let's Encrypt
4	**Beda model Apache vs Nginx & dampaknya?**	Worker per koneksi vs event loop; Nginx lebih hemat memori di beban tinggi
5	**3 situs kampus di 1 server — apa yang perlu?**	3 vhost (server_name beda) + record DNS/CNAME ke IP yang sama

Rangkuman

Arsitektur

Apache: worker per koneksi · Nginx: event-driven, unggul di beban tinggi.

Virtual host

Satu IP melayani banyak domain berdasarkan header Host.

SSL & tuning

HTTPS melindungi data; gzip & cache mempercepat tanpa hardware baru.

Pertemuan 11

Konfigurasi File Sharing Server (Samba/NFS/FTP). Baca Bab 10 Buku Ajar sebelum kelas.

Referensi: Buku Ajar Bab 9 · Nemeth et al. (2021) · Nginx Beginner's Guide · Apache HTTP Server Documentation