

# TOPIK 1: Deployment Offline untuk Multi-Project PHP & JSP

## ❖ Narasi Proyek

Mahasiswa diminta untuk membangun sebuah server lokal berbasis Ubuntu di Proxmox yang mampu menampung **dua jenis aplikasi web** secara simultan:

- Satu aplikasi berbasis **PHP + MariaDB/MySQL**
- Satu aplikasi berbasis **Java JSP** yang dijalankan menggunakan Apache Tomcat

Kedua aplikasi ini harus dapat diakses melalui alamat **virtual host (subdomain lokal)** dalam jaringan offline, tanpa harus melakukan perubahan manual pada file hosts di sisi client. Konfigurasi dilakukan secara penuh dari sisi server menggunakan DNS resolver lokal (dnsmasq atau bind).

## ❖ Capaian Output

- VM Ubuntu Server aktif di Proxmox dan terkoneksi dalam jaringan virtual
- Aplikasi PHP (misal: loginapp) berjalan dengan koneksi database
- Aplikasi JSP (misal: portofolioapp) dapat diakses melalui Tomcat via NGINX reverse proxy
- Virtual host:
  - `http://login.localhost` untuk PHP
  - `http://portofolio.localhost` untuk JSP
- Konfigurasi DNS lokal agar semua client dapat mengakses domain lokal tanpa setting manual
- Sistem backup otomatis untuk kedua aplikasi
- Dokumentasi lengkap dengan konfigurasi NGINX, Tomcat, dan database

## ❖ Tahapan Implementasi

### A. Persiapan Server

1. Instal Ubuntu Server minimal 2 core, 2 GB RAM di Proxmox
2. Atur IP statis dan aktifkan SSH

### B. Instalasi Software

1. Install Apache2 atau NGINX
2. Install PHP 8.x dan MariaDB

3. Install Tomcat 9 (untuk JSP)

4. Install dnsmasq untuk DNS resolver lokal

### C. Setup Project

1. Clone atau copy project PHP ke /var/www/loginapp/

2. Copy project JSP (WAR/jsp folder) ke /var/lib/tomcat9/webapps/portofolioapp/

### D. Konfigurasi Virtual Host

1. Buat konfigurasi NGINX:

#### ⌘ Contoh untuk PHP:

```
server {  
    listen 80;  
    server_name login.localhost;  
    root /var/www/loginapp;  
  
    index index.php;  
    location ~ \.php$ {  
        include snippets/fastcgi-php.conf;  
        fastcgi_pass unix:/run/php/php8.1-fpm.sock;  
    }  
}
```

#### ⌘ Contoh untuk JSP (via reverse proxy ke Tomcat):

```
server {  
    listen 80;  
    server_name portofolio.localhost;  
  
    location / {  
        proxy_pass http://localhost:8080/portofolioapp/;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}
```

### E. Konfigurasi DNS Lokal

1. Install dnsmasq di server:

```
sudo apt install dnsmasq
```

2. Tambahkan konfigurasi domain:

```
echo "address=/login.localhost/192.168.100.10" >> /etc/dnsmasq.conf  
echo "address=/portofolio.localhost/192.168.100.10" >> /etc/dnsmasq.conf  
sudo systemctl restart dnsmasq
```

3. Atur client agar menggunakan IP server sebagai DNS utama

F. Backup Otomatis

1. Buat skrip backup .sh untuk file & database
2. Jadwalkan dengan cron untuk harian atau mingguan

❖ **Dokumentasi Wajib**

- Screenshots akses aplikasi dari client
- File konfigurasi .conf, .xml, .sh
- Struktur folder project
- Diagram jaringan (jika perlu)

## TOPIK 2: Instalasi dan Konfigurasi CyberPanel pada Server Offline Ubuntu

### ❖ Narasi Proyek

Mahasiswa diminta untuk mengubah server lokal (Ubuntu) yang berjalan di Proxmox menjadi **panel manajemen server berbasis GUI (CyberPanel)**. Panel ini akan digunakan untuk:

- Men-deploy website berbasis WordPress dan HTML/PHP
- Melakukan pengelolaan domain lokal
- Menyediakan layanan FTP dan backup

Semua proses dilakukan **tanpa koneksi internet**, sehingga CyberPanel dan seluruh dependensinya harus diinstal **secara offline**.

### ❖ Capaian Output

- CyberPanel aktif di `https://<ip>:8090`
- Minimal dua website aktif:
  - WordPress lokal
  - Website statis HTML atau PHP
- Akses domain lokal tanpa edit `hosts`
- FTP user aktif untuk masing-masing web
- Backup otomatis aktif di CyberPanel
- SSL self-signed aktif untuk simulasi keamanan
- Dokumentasi konfigurasi, screenshot, dan panel

### ❖ Tahapan Implementasi

#### A. Persiapan Offline

1. Unduh seluruh paket `.deb` CyberPanel dan dependensinya dari internet sebelumnya
2. Upload paket ke server via Proxmox ISO atau SCP

#### B. Instalasi CyberPanel Offline

1. Gunakan installer lokal:

```
chmod +x install.sh  
./install.sh --offline
```

2. Ikuti wizard hingga selesai

C. Akses Panel dan Login

1. Buka browser: `https://<ip-server>:8090`

2. Login admin (default: admin / 1234567)

D. Setup Website

1. Buat dua website:

`owp.localhost` (WordPress)

`ohtml.localhost` (HTML)

2. Upload konten web via FTP / File Manager CyberPanel

E. DNS Lokal

Gunakan `dnsmasq` seperti topik 1 untuk mapping domain lokal

F. Keamanan

1. Aktifkan SSL self-signed

2. UFW: buka port 80, 443, 8090, 21 (FTP)

3. Tambahkan backup otomatis mingguan dari menu panel

❖ **Dokumentasi Wajib**

- Screenshot Dashboard CyberPanel
- Screenshot tampilan kedua website
- Konfigurasi FTP, SSL, Backup
- File `.conf`, `.sh`, `.zip` WordPress, dll

# TOPIK 3: Deployment Project JS Menggunakan Docker

## ❖ Narasi Proyek

Mahasiswa membangun dan men-deploy aplikasi berbasis **JavaScript modern** (Next.js, React.js, Express.js, atau lainnya) ke dalam container **Docker**, secara **offline** di server Ubuntu Proxmox. Proyek ini wajib memanfaatkan:

- Dockerfile
- `docker-compose`
- NGINX sebagai reverse proxy
- Konfigurasi domain lokal tanpa edit `hosts`

## ❖ Capaian Output

- Docker aktif dan terkonfigurasi offline
- Aplikasi JS (misal: portofolioapp berbasis React.js) berjalan di container
- Reverse proxy via NGINX (port 80 → 3000 internal container)
- Domain lokal `http://project.localhost` aktif
- Backup volume Docker (opsional)
- File Dockerfile & `docker-compose.yml` terdokumentasi
- Screenshot hasil dan dokumentasi lengkap

## ❖ Tahapan Implementasi

### A. Instalasi Docker Offline

1. Unduh `.deb` paket Docker dan Compose di PC online
2. Upload dan instal di server Ubuntu offline:

```
sudo dpkg -i docker-ce*.deb
```

### B. Siapkan Aplikasi

1. Clone project JS: React.js, Express.js, Next.js, dst
2. Struktur folder disesuaikan

### C. Buat Dockerfile

**Contoh untuk Express.js:**

```
FROM node:18

WORKDIR /app

COPY . .

RUN npm install

CMD ["npm", "start"]
```

D. Buat `docker-compose.yml`

```
version: '3'

services:
  web:
    build: .
    ports:
      - "3000:3000"
```

E. Jalankan Container

```
docker-compose up -d
```

F. Reverse Proxy dengan NGINX

1. Install NGINX
2. Konfigurasi virtual host untuk `project.localhost`:

```
server {
    listen 80;
    server_name project.localhost;

    location / {
        proxy_pass http://localhost:3000;
    }
}
```

G. DNS Resolver Lokal (`dnsmasq`)

```
echo "address=/project.localhost/192.168.100.10" >> /etc/dnsmasq.conf
sudo systemctl restart dnsmasq
```

❖ **Dokumentasi Wajib**

- Dockerfile dan `docker-compose.yml`
- Screenshot dashboard Docker (`docker ps`, `docker images`)
- Screenshot aplikasi berjalan di browser
- File `.conf` `nginx`