

Tugas MK. ITI

“Instalasi Mesin Virtualisasi Docker dan Membuat Aplikasi Web Menggunakan Docker Compose”

Dosen Pengampu : R. Dimas Adityo



Disusun Oleh :

Bima Sangraloka (2314311010)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BHAYANGKARA SURABAYA**

2020

KATA PENGANTAR

Puji syukur saya ucapkan kehadiran Allah SWT yang telah memberikan rahmat serta hidayah kepada kita semua, sehingga berkat Karunia-Nya penulis dapat menyelesaikan makalah ini guna memenuhi Tugas Akhir UAS untuk mata kuliah Jaringan Komputer, dengan judul *“Instalasi Mesin Virtualisasi Docker dan Membuat Aplikasi Web Menggunakan Docker Compose”*

Saya menyadari bahwa dalam penulisan makalah ini tidak terlepas dari bantuan banyak pihak yang dengan tulus memberikan doa, saran dan kritik sehingga makalah ini dapat terselesaikan.

Saya menyadari sepenuhnya bahwa makalah ini masih jauh dari sempurna dikarenakan terbatasnya pengetahuan yang saya miliki. Oleh karena itu, kami mengharapkan segala bentuk saran serta masukan bahkan kritik yang membangun dari berbagai pihak. Dalam penyusunan makalah ini penulis berharap semoga makalah ini dapat bermanfaat bagi penulis sendiri maupun kepada pembaca umumnya.

Penyusun

Daftar Isi

KATA PENGANTAR.....	2
BAB I	4
PENDAHULUAN.....	4
1. 1 Latar Belakang.....	4
1.2 Rumusan Masalah	4
1.3 Tujuan Masalah.....	4
BAB II	5
PEMBAHASAN	5
2.1. Instalasi Docker Desktop	5
2.2 Cara Membuat Docker Compose	6
2.3 Cara menjalankan Docker Compose.....	6
BAB III.....	11
PENUTUP	11
3.1. Kesimpulan.....	11
3.2. Saran.....	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang semakin berkembang pesat, kebutuhan akan teknologi yang efisien dan fleksibel dalam mengelola aplikasi web menjadi semakin penting. Salah satu teknologi yang kini banyak digunakan untuk keperluan tersebut adalah Docker, sebuah platform open-source yang memungkinkan pembuatan, pengiriman, dan menjalankan aplikasi dalam kontainer yang terisolasi. Kontainer Docker memberikan lingkungan yang konsisten dari tahap pengembangan hingga produksi, sehingga mengurangi masalah yang terkait dengan perbedaan lingkungan.

Docker mempermudah proses deployment aplikasi dengan cara mengemas aplikasi dan seluruh dependensinya ke dalam sebuah kontainer. Hal ini memungkinkan aplikasi untuk berjalan di berbagai lingkungan tanpa perlu khawatir mengenai kompatibilitas sistem. Salah satu penggunaan umum dari Docker adalah untuk menginstal dan mengelola webserver seperti Nginx. Nginx adalah web server yang cepat dan ringan, yang dapat digunakan sebagai server web, reverse proxy, load balancer, dan HTTP cache. Kombinasi Docker dan Nginx menawarkan solusi yang handal dan mudah diatur untuk hosting aplikasi web.

Dengan memanfaatkan Docker untuk menginstal Nginx sebagai webserver dan PHP sebagai bahasa pemrograman, pengembang dapat menciptakan lingkungan pengembangan yang konsisten dan efisien. Docker memastikan bahwa aplikasi yang dikembangkan akan berjalan dengan cara yang sama di lingkungan pengembangan, pengujian, dan produksi. Hal ini mengurangi waktu yang dibutuhkan untuk setup dan konfigurasi lingkungan, serta mengurangi risiko kesalahan yang disebabkan oleh perbedaan lingkungan.

1.2 Rumusan Masalah

1. Bagaimana cara instalasi *Docker* pada *desktop Windows*?
2. Bagaimana cara instalasi *Docker Compose* ?
3. Bagaimana cara konfigurasi *PHP* dan *MariaDB* untuk konten *web*?

1.3 Tujuan Masalah

1. Untuk mengetahui cara instalasi *Docker* pada *desktop Windows*.
2. Untuk mengetahui cara instalasi *Docker Compose*..
3. Untuk mengetahui cara konfigurasi konfigurasi *PHP* dan *MariaDB* untuk konten *web*.

BAB II

PEMBAHASAN

2.1. Instalasi Docker Desktop untuk Windows

Docker for Windows pertama kali dirilis dalam versi beta pada bulan Maret 2016, dengan versi stabilnya dirilis pada bulan Juli 2016. Docker for Windows memungkinkan pengguna untuk menjalankan dan mengelola kontainer Docker di mesin Windows, memberikan pengalaman yang lebih terpadu dan optimal dibandingkan dengan solusi sebelumnya yang mengandalkan mesin virtual.

Pertama-tama sebelum mengunduh Docker desktop pastikan spesifikasi yang digunakan mencukupi persyaratan minimum dari docker-nya, berikut persyaratan minimum Docker for Windows :

- Windows 11 64-bit atau Windows 10 64-bit
- 4GB RAM
- Mengaktifkan virtualisasi pada BIOS perangkat
- Mengaktifkan fitur WSL 2 pada Windows

Setelah melihat persyaratan untuk mengunduh, kunjungi situs resmi Docker <https://docs.docker.com/desktop/install/windows-install/> , lalu ikuti langkah berikut :

- Unduh installer dan tunggu hingga selesai mengunduh
- Jalankan installer dan lakukan instalasi sesuai petunjuk, yang mencakup persetujuan lisensi dan konfigurasi opsi instalasi
- Tunggu hingga selesai menginstall

Setelah berhasil menginstall Docker Desktop, buka Docker desktop pada menu start dan tunggu hingga selesai melakukan setup dan menampilkan *dashboard*. Untuk memverifikasi instalasi lakukan langkah berikut:

- Buka *Command Prompt*
- Untuk memverifikasi bahwa Docker sudah terinstall dengan benar, jalankan perintah :

```
docker --version
```

- Untuk memastikan Docker dapat menjalankan *container* dengan benar, jalankan perintah :

```
docker run Hello-World
```

2.2 Instalasi Docker Compose

Pada Docker Desktop biasanya sudah termasuk Docker Compose, jadi tidak diperlukan instalasi tambahan untuk Docker Compose. Untuk membuat Docker Compose ikuti langkah langkah berikut :

- Buat direktori proyek baru untuk aplikasi yang akan dikembangkan
- Buka Visual Studio Code dan pilih “Open Folder”
- Pilih folder yang sudah dibuat tadi
- Di dalam direktori proyek, buat file bernama “**docker-compose.yml**”
- Konfigurasi “**docker-compose.yml**” seperti berikut :

```
version: "3.9"
services:
  nginx:
    image: nginx:latest
    container_name: nginx-container
    ports:
      - 80:80
```

- Jalankan perintah “**docker-compose up -d**” untuk menginstall dan menjalankan Nginx
- Jalankan perintah “**docker ps**” untuk memeriksa apakah sudah berjalan dengan baik
- Untuk memastikan nginx terinstall pada docker compose cek pada web browser dengan mengetikan URL “<http://alamat-ip-server>”. Atau “localhost” yang nantinya akan muncul tulisan “**Welcome to Nginx**”

2.3 Konfigurasi PHP dan database MariaDB untuk konten web

Setelah Nginx berjalan dengan lancar langkah berikutnya adalah instalasi PHP dan database MariaDB dan konfigurasinya

1. Instalasi dan Konfigurasi PHP

- Buat direktori PHP pada direktori proyek Anda
- Unduh template atau buat code PHP untuk web, di sini saya menggunakan template dari <https://github.com/rapidcode-technologies-private-limited/php-e-commerce>
- Masuk ke dalam direktori PHP Anda dan buat file “**Dockerfile**”
- Tambahkan seperti berikut:

```
FROM php:7.0-fpm
RUN docker-php-ext-install mysqli pdo pdo_mysql
RUN docker-php-ext-enable mysqli
```

- Buat direktori baru Nginx pada proyek anda
- Di dalam direktori Nginx buat file baru bernama “**default.conf**”

- Konfigurasi seperti berikut ini:

```
server {  
  
    listen 80 default_server;  
    root /php;  
    index index.html index.php;  
  
    charset utf-8;  
  
    location / {  
        try_files $uri $uri/ /index.php?$query_string;  
    }  
  
    location = /favicon.ico { access_log off; log_not_found off; }  
    location = /robots.txt { access_log off; log_not_found off; }  
  
    access_log off;  
    error_log /var/log/nginx/error.log error;  
  
    sendfile off;  
  
    client_max_body_size 100m;  
  
    location ~ .php$ {  
        fastcgi_split_path_info ^(.+\.php)(/.+)$;  
        fastcgi_pass php:9000;  
        fastcgi_index index.php;  
        include fastcgi_params;  
        fastcgi_read_timeout 300;  
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
        fastcgi_intercept_errors off;  
        fastcgi_buffer_size 16k;  
        fastcgi_buffers 4 16k;  
    }  
  
    location ~ /\.ht {  
        deny all;  
    }  
}
```

- Buat lagi file “**Dockerfile**” dalam direktori Nginx dan konfigurasi seperti ini:

```
FROM nginx  
COPY ./default.conf /etc/nginx/conf.d/default.conf
```

- Lakukan update pada “**docker-compose.yml**” seperti berikut ini :

```
version: "3.9"
services:
  nginx:
    build: ./nginx/
    ports:
      - 80:80

    volumes:
      - ./php:/var/www/html/

  php:
    build: ./php/
    expose:
      - 9000
    volumes:
      - ./php:/var/www/html/
```

- Jalankan perintah “**docker-compose up -d**” untuk menginstall semua image dan menjalankannya
- Jalankan perintah “**docker ps**” untuk melihat bahwa nginx dan php sudah berjalan
- Untuk memastikan bahwa PHP sudah berjalan, kembali cek pada web browser mengetikkan “**localhost**”

2. Instalasi dan Konfigurasi database MariaDB

- Update “**docker-compose.yml**” seperti berikut ini :

```

version: "3.9"
version: "3.9"
services:
  nginx:
    build: ./nginx/
    ports:
      - 80:80

    volumes:
      - ./php:/var/www/html/

  php:
    build: ./php/
    expose:
      - 9000
    volumes:
      - ./php:/var/www/html/

```

```

db:
  image: mariadb
  volumes:
    - mysql-data:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: bima1 (pass root bebas)
    MYSQL_DATABASE: dbweb (nama database bebas)

```

```

volumes:
  mysql-data:

```

- Jalankan perintah “**docker-compose up -d**” untuk menginstall dan menjalankan image Mariadb
- Jalankan perintah “**docker exec -it “ID kontainer database” /bin/sh**” untuk mengkonfigurasi database
- Buat user baru untuk database

```
CREATE USER 'namauser'@'%' IDENTIFIED BY "passworduser"
```

- Berikan privileges kepada user baru yang sudah dibuat

```
GRANT ALL PRIVILEGES ON *.* TO 'namauser'@'%';
FLUSH PRIVILEGES ;
```

- Keluar dari MariaDB shell

exit

- Muat informasi penyimpanan produk ke database

```
cat > db-load-script.sql <<-EOF
```

USE "nama database";

```
CREATE TABLE products (id mediumint(8) unsigned NOT NULL auto_increment,Name  
varchar(255) default NULL,Price varchar(255) default NULL, ImageUrl varchar(255) default  
NULL,PRIMARY KEY (id)) AUTO_INCREMENT=1;
```

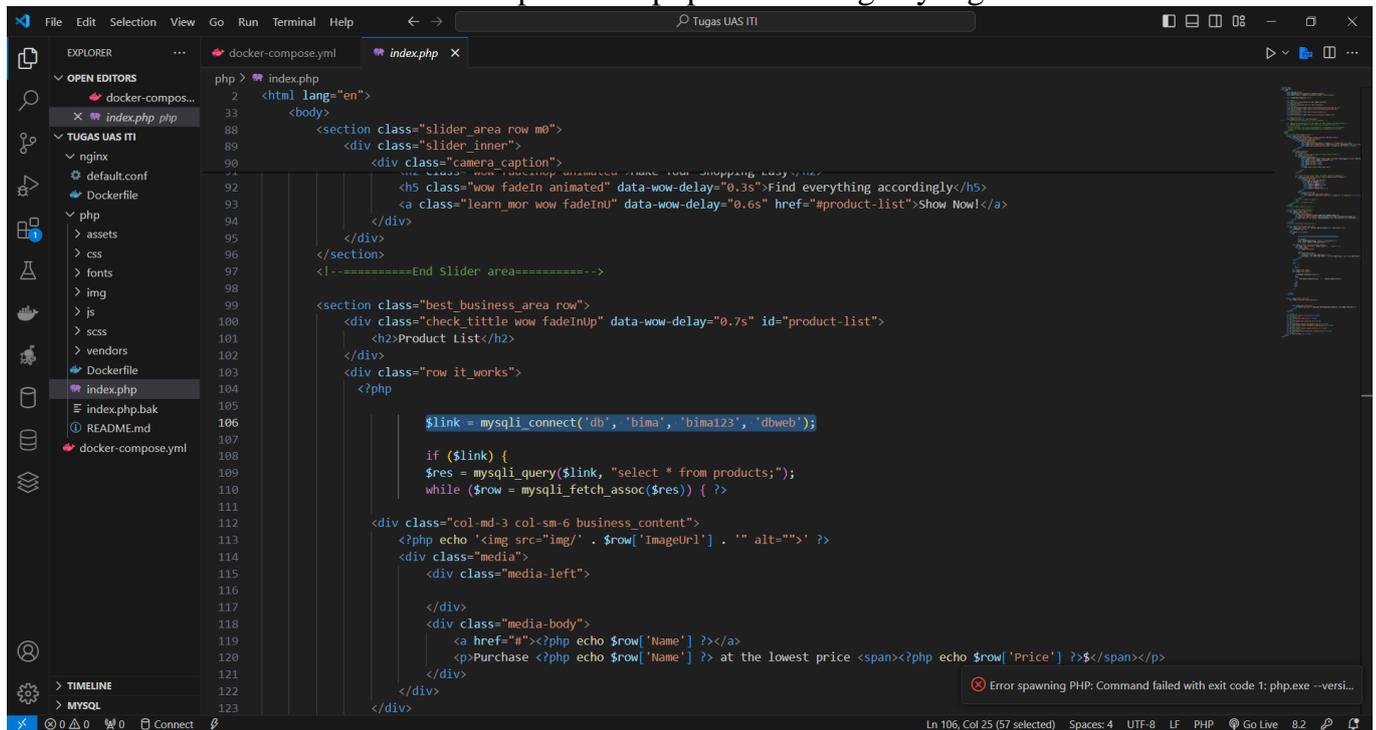
- Jalankan SQL script

```
mariadb -u root -pmariadb < db-load-script.sql
```

- Exit dari MariaDB kontainer

exit

- Pastikan link database pada inexphp sesuai dengan yang dibuat tadi

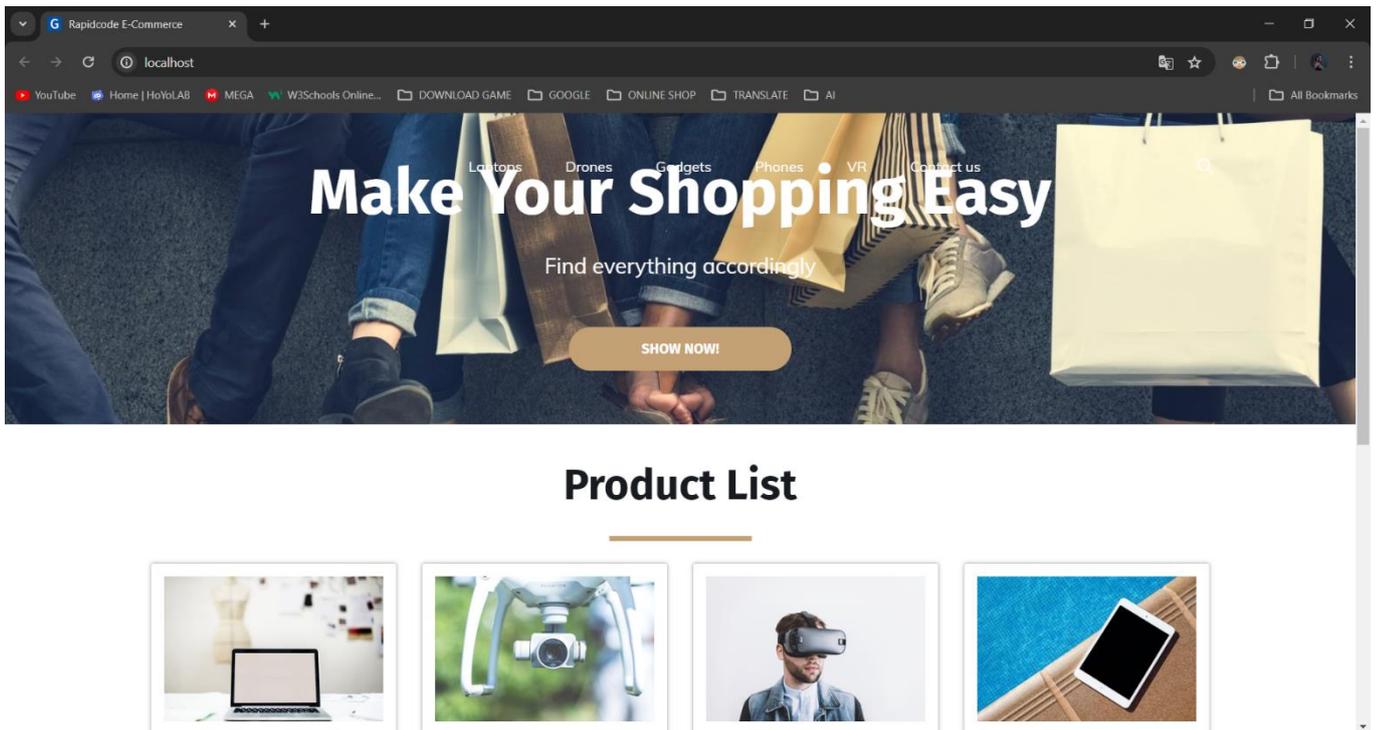


The screenshot shows a code editor with the following PHP code in `index.php`:

```
106 $link = mysqli_connect('db', 'bima', 'bima123', 'dweb');  
107  
108 if ($link) {  
109   $res = mysqli_query($link, "select * from products;");  
110   while ($row = mysqli_fetch_assoc($res)) { ?>  
111  
112 <div class="col-md-3 col-sm-6 business_content">  
113 <?php echo "<img src='img/' . $row['ImageUrl'] . "' alt=''> ?>  
114 <div class="media">  
115   <div class="media-left">  
116  
117   </div>  
118   <div class="media-body">  
119     <a href="#">?php echo $row['Name'] ?></a>  
120     <p>Purchase <?php echo $row['Name'] ?> at the lowest price <span>?php echo $row['Price'] ?></span></p>  
121   </div>  
122 </div>  
123
```

An error message is visible at the bottom right: "Error spawning PHP: Command failed with exit code 1: php.exe --versi..."

- Cek lagi URL pada website maka akan muncul produk yang sudah dimasukkan di database tadi



BAB III

PENUTUP

3.1. Kesimpulan

Setelah kita mengimplementasikan *docker* dengan *docker compose* untuk membuat webserver menggunakan *nginx* dan bahasa *php* serta *database Mariaadb*, kita menjadi tahu cara membuat website yang mudah dan cepat menggunakan software yang satu ini

3.2. Saran

Bagi pembaca makalah ini, semoga makalah ini dapat menjadi bermanfaat bagi semua. Kepada generasi muda untuk lebih bijak dalam penggunaan perkembangan teknologi saat ini.