

TUGAS
INFRASTRUKTUR TEKNOLOGI INFORMASI
INSTALLASI DOCKER DEKSTOP & WEB
ECOMMERCE



DUSUSUN OLEH :
LUTFIA FEBRIANTI
2014311031

TEKNIK INFORMATIKA

UNIVERSITAS BHAYANGKARA SURABAYA
2023/2024

KATA PENGANTAR

Puji syukur saya panjatkan kehadirat Allah SWT yang telah memberikan rahmat serta karunia-Nya kepada saya sehingga saya berhasil menyelesaikan Makalah ini yang alhamdulillah tepat pada waktunya yang berjudul “INSTALLASI DOCKER DEKSTOP & WEB ECOMMERCE”.

Diharapkan Makalah ini dapat memberikan informasi kepada kita semua tentang Instalasi Mesin Docker Dekstop & Web Ecommerce. Penulis menyadari bahwa makalah ini masih jauh dari sempurna, oleh karena itu kritik dan saran dari semua pihak yang bersifat membangun selalu saya harapkan demi perbaikan dari kesalahan-kesalahan dalam penulisan makalah ini.

Akhir kata, Saya sampaikan terima kasih kepada semua pihak yang telah berperan serta dalam penyusunan makalah ini dari awal sampai akhir. Semoga Allah SWT senantiasa meridhai segala usaha kita. Amin.

Surabaya , 09 Juni 2024

DAFTAR ISI

HALAMAN

JUDUL.....	1
KATA PENGANTAR.....	2
DAFTAR ISI.....	3

BAB I

A. LATAR BELAKANG.....	4
B. TUJUAN PENULISAN.....	4

BAB II

A. Proses Mengunduh Docker, Cek Version dan Instalasi Library.....	5
B. Proses Pembuatan Folder HTML dan File-File	6
C. Proses Koneksi dan Tampilan Web-App Saat Berjalan.....	11

BAB III

A. KESIMPULAN	14
B. SARAN	14

BAB 1

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi informasi yang pesat mendorong kebutuhan akan solusi efisien dalam pengelolaan aplikasi. Virtualisasi, terutama melalui platform containerization seperti Docker, memungkinkan penggunaan sumber daya komputasi yang lebih optimal. Docker Desktop memberikan kemudahan bagi pengguna untuk mengelola container di lingkungan pengembangan mereka, mempercepat proses pengembangan dan penerapan aplikasi.

Dalam konteks e-commerce, keandalan dan skalabilitas aplikasi web sangat penting. Docker menyediakan isolasi lingkungan yang memastikan aplikasi berjalan konsisten di berbagai platform, serta memudahkan pengelolaan dependensi. Dengan Docker, pengembang dapat menjalankan seluruh stack aplikasi, termasuk server web dan database, dalam container yang terisolasi, mengurangi risiko konflik dan meningkatkan efisiensi deployment.

Makalah ini bertujuan memberikan panduan praktis untuk instalasi Docker Desktop dan konfigurasi aplikasi web e-commerce menggunakan Docker. Diharapkan, teknologi ini dapat meningkatkan produktivitas dan kualitas aplikasi, serta memberikan pemahaman dasar tentang penggunaan Docker untuk mengembangkan dan mengelola aplikasi e-commerce yang andal dan skalabel.

B. Tujuan Pembelajaran

Makalah ini dirancang untuk memenuhi persyaratan ujian akhir semester Infrastruktur Teknologi Informasi, sekaligus mengembangkan pemahaman mendalam tentang Konfigurasi LEMP (Linux, Nginx, MySQL, PHP) dan Wordpress menggunakan Docker, yang merupakan teknologi kontainerisasi terdepan yang mendukung efisiensi dan fleksibilitas dalam pengembangan aplikasi web.

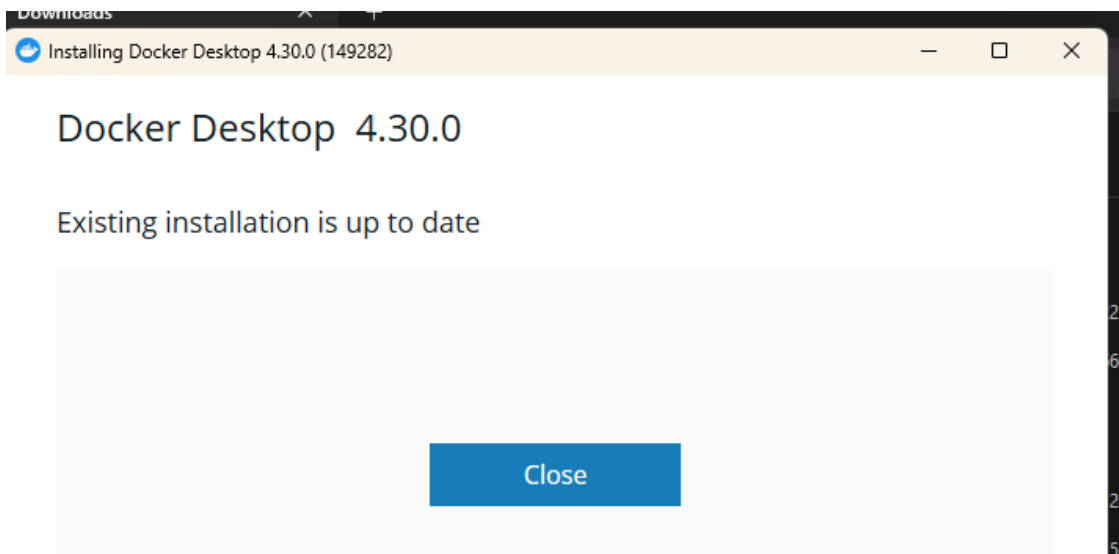
BAB II

PEMBAHASAN



A. Proses Mengunduh Docker, Cek Versi dan Instalasi library

Untuk menjalankan aplikasi ini dengan benar, kita perlu melalui proses mengunduh beberapa library yang diperlukan. Berikut adalah langkah-langkah untuk mengunduh library-library tersebut:

1. **Buka situs resmi Docker:**
Kunjungi <https://www.docker.com/products/docker-desktop>.
2. **Klik tombol "Download Docker Desktop":**
Pilih versi untuk Windows dan simpan file instalasi di komputer Anda.



3. **Download Git :** Sistem kontrol versi yang memungkinkan pengembang melacak, mengelola, dan berkolaborasi dalam pengembangan perangkat lunak dengan mudah dan efisien.

 Git-2.45.2-64-bit.exe	05/06/2024 20:55	Application	66.535 KB
 Docker Desktop Installer.exe	03/06/2024 18:28	Application	487.594 KB

4. Jika ingin mengecek versi Docker:
Jalankan perintah docker --version di terminal.

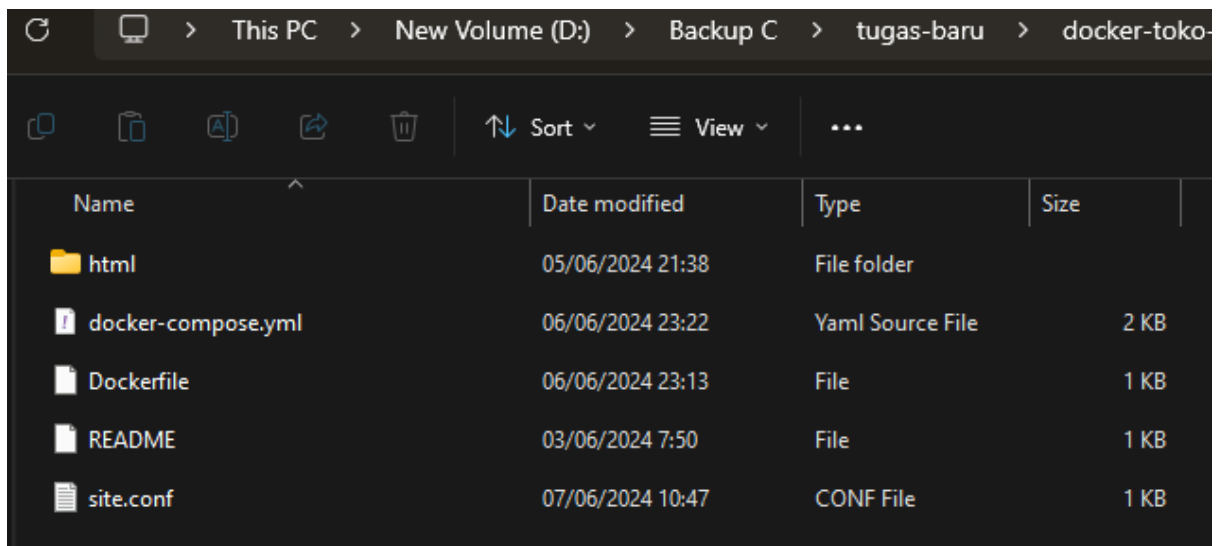
```
ASUS S333JQ@LAPTOP-3NV80MJF MINGW64 /d/Backup C/tugas-baru/docker-toko-online-latest
test
$ docker --version
Docker version 26.1.1, build 4cf5afa
```

5. Jika ingin mengunduh Nginx:
Jalankan perintah docker pull nginx di terminal.

```
MINGW64:/d/Backup C/tugas-baru/docker-toko-online-latest
ASUS S333JQ@LAPTOP-3NV80MJF MINGW64 /d/Backup C/tugas-baru/docker-toko-online-latest
test
$ docker pull nginx:latest
```

B. Proses Pembuatan Folder HTML dan File-File

6. docker-compose up --build
7. Lalu buat file baru docker-compose.yml dan isi dengan layanan service seperti berikut:



8. Dengan konfigurasi ini, kami menjalankan beberapa layanan dalam Docker Compose, termasuk server web dengan Nginx, server PHP, basis data MySQL, dan PHPMyAdmin. Setiap layanan diatur dengan pengaturan spesifik seperti restart, port, volume, dan dependensi untuk memastikan aplikasi berjalan dengan lancar dalam lingkungan Docker.

```
version: '3.7'
# Kita jalankan docker-compose up ( ngepull folder docekr layanan dan
website yang kita buat
services:
  web:
    image: nginx:latest
    restart: always
    ports:
      - "8081:80"
      # Yang merujuk pada layanan web di docker kompos bagian volume
      kita menautakan file site cofig kita ke default confid pada nginx
    volumes:
      - ./html:/html
      - ./site.conf:/etc/nginx/conf.d/default.conf
    depends_on:
      - php
    networks:
      - nginx-network
      # tambakan layanan service php dengan mengikutsertakan docker
      file yang telah dibuat, lalu kembali membuat site config
  php:
    build:
      dockerfile: Dockerfile
      context: .
    volumes:
      - ./html:/html
    networks:
      - nginx-network
  mysql-db:
    image: mysql:5.7
    restart: always
    depends_on:
      - php
    environment:
      MYSQL_ROOT_PASSWORD: root
    volumes:
```

```

    - mysql-data:/var/lib/mysql
networks:
  - nginx-network
phpmyadmin:
  image: phpmyadmin/phpmyadmin:5.0.1
  restart: always
  environment:
    PMA_HOST: mysql-db
  ports:
    - "8080:80"
  networks:
    - nginx-network

volumes:
  mysql-data:

networks:
  nginx-network:
    driver: bridge

```

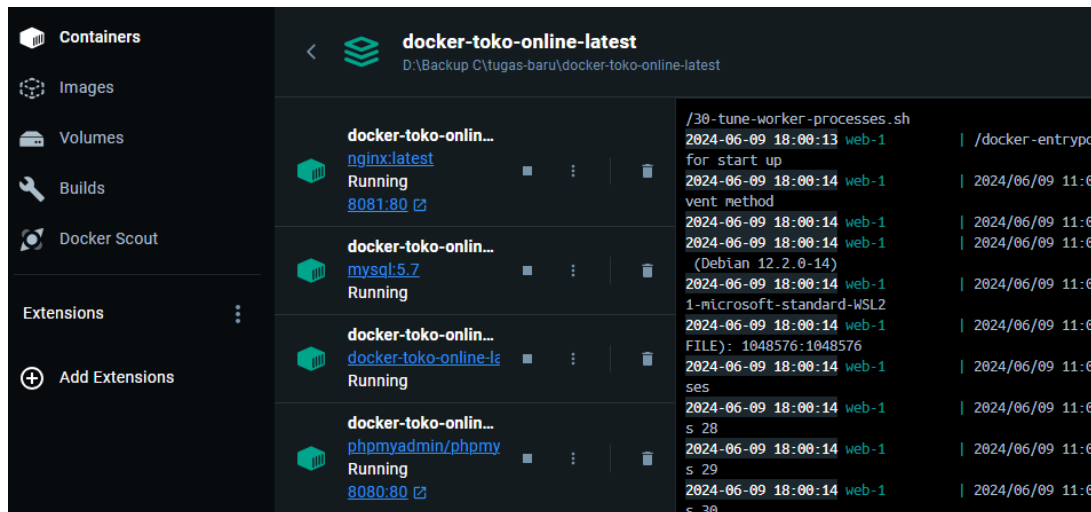
9. Untuk membangun, mengonfigurasi, dan menjalankan layanan-layanan yang didefinisikan dalam file docker-compose.yml. Ini akan membuat semua kontainer yang diperlukan berdasarkan konfigurasi yang ada dalam file tersebut dan menjalankannya.

```

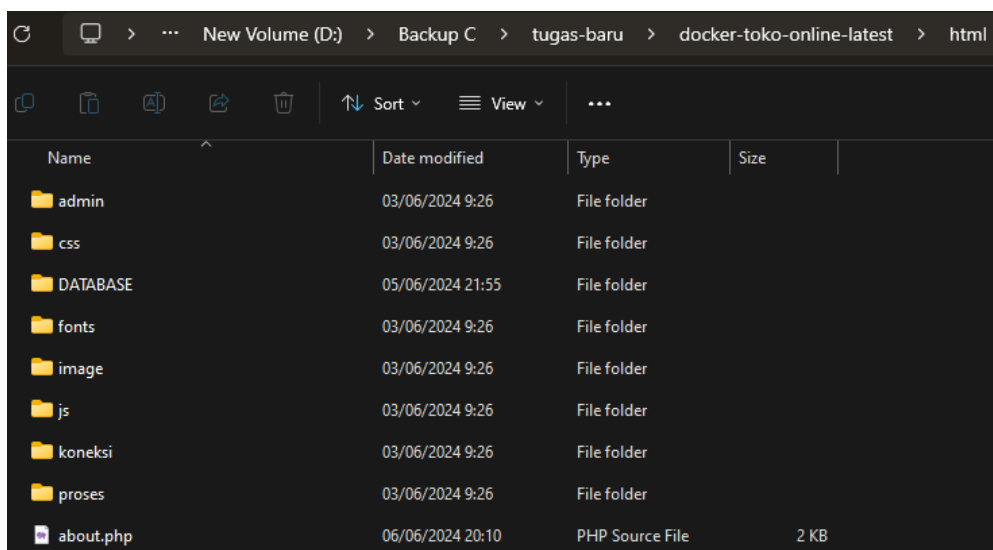
ASUS S333JQ@LAPTOP-3NV80MJF MINGW64 /d/Backup C/tugas-baru/docker-toko-online-latest
$ docker-compose up
time="2024-06-09T18:00:11+07:00" level=warning msg="D:\\Backup C\\tugas-baru\\docker-toko-online-latest\\docker-compose.yml: 'version' is obsolete"
Container docker-toko-online-latest-php-1 Created
Container docker-toko-online-latest-phpmyadmin-1 Created
Container docker-toko-online-latest-web-1 Created
Container docker-toko-online-latest-mysql-db-1 Created
Attaching to mysql-db-1, php-1, phpmyadmin-1, web-1
phpmyadmin-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.19.0.2. Set the 'ServerName' directive globally to suppress this message
phpmyadmin-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.19.0.2. Set the 'ServerName' directive globally to suppress this message
phpmyadmin-1 | [Sun Jun 09 11:00:13.467578 2024] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.38 (Debian) PHP/7.4.1 configured -- resuming normal operations
phpmyadmin-1 | [Sun Jun 09 11:00:13.467750 2024] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
php-1 | [09-Jun-2024 11:00:13] NOTICE: fpm is running, pid 1
php-1 | [09-Jun-2024 11:00:13] NOTICE: ready to handle connections

```


10. Jika dijalankan di Docker Desktop, kontainer akan menampilkan empat layanan server yang sedang berjalan dan aktif. Setiap layanan memiliki port yang ditetapkan untuk akses, dengan server yang sesuai. Misalnya, server Nginx akan menggunakan port 8081 dan server PHPMyAdmin akan menggunakan port 8080.



11. Lalu, kami membuat file HTML yang digunakan untuk menampung file-file aplikasi web PHP yang telah kami unduh dari GitHub. Lalu, kami membuat file HTML yang digunakan untuk menampung file-file aplikasi web PHP yang telah kami unduh dari GitHub.



12. Buat file bernama `dockerfile` = File teks yang mendefinisikan lingkungan dan konfigurasi yang diperlukan untuk membangun image container.

```
# layanan fpm asecgimanager alat untuk mempercepat kinerja situs web
FROM php:7.2-fpm
# didalam ny ada perintah run untuk menjalankan dan menginstal mysql php data
objek(pdo) data mysql
RUN docker-php-ext-install mysqli pdo pdo_mysql
```

13. Buat File Bernama `site.conf` = File konfigurasi untuk server web Nginx. Dalam file ini, pengguna dapat menentukan konfigurasi server seperti pengaturan rute, pengaturan proxy, aturan pemetaan, dan pengaturan lainnya yang diperlukan untuk mengonfigurasi server Nginx sesuai dengan kebutuhan aplikasi web yang dijalankan.

```
server {
    listen 80;
    index index.php index.html;
    server_name localhost;
    error_log /var/log/nginx/error.log;
    access_log /var/log/nginx/access.log;
    root /html;

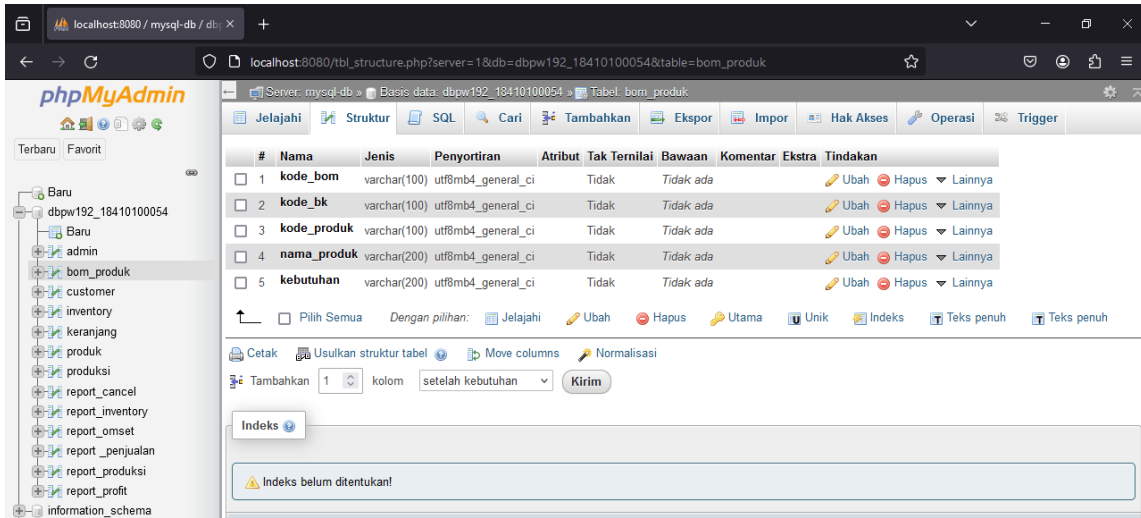
    location / {
        try_files $uri $uri/ /index.php;
    }

    location ~ /\.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.(php))(/.+)$;
        fastcgi_pass php:9000;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
```

```
fastcgi_param PATH_INFO $fastcgi_path_info;
}
}
```

C. Proses Koneksi dan Tampilan Web-App Saat Berjalan

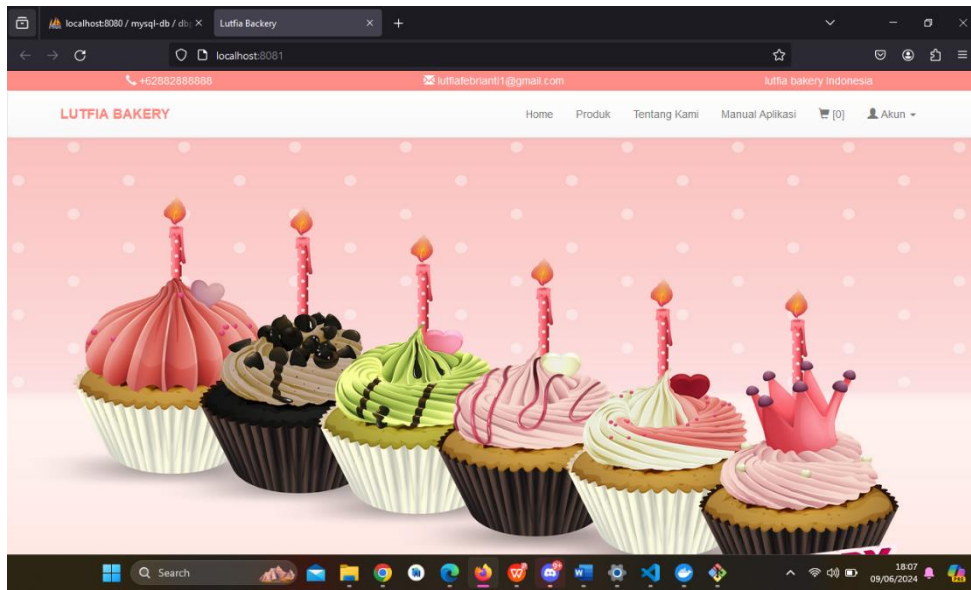
14. Buatlah database Di phpMyAdmin seperti gambar berikut



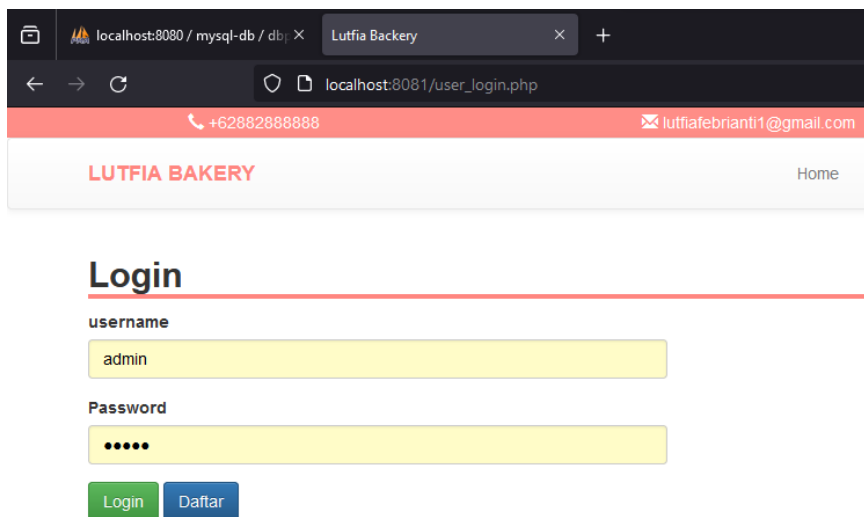
15. Dalam file koneksi.php, dilakukan proses pengoneksian antara web server dengan Docker untuk memastikan bahwa aplikasi web dapat berinteraksi dengan database yang berjalan di dalam kontainer Docker.

```
File Edit Selection View Go ... docker-t
Dockerfile docker-compose.yml koneksi.php x site.conf header.php.html footer.ph
html > koneksi > koneksi.php
1 <?php
2 $conn = mysqli_connect("mysql-db", "root", "root", "dbpw192_18410100054");
3
4
```

16. Berikut Tampilan Web-App saat dan setelah dijalankan



17. Berikut tampilan Produk hingga login.



localhost:8080 / mysql-db / db; x Lutfia Bakery x +

localhost:8081/register.php

+62882888888 lutfiabrian1@gmail.com lutfia bakery Indonesia

LUTFIA BAKERY Home Produk Tentang Kami Manual Aplikasi [0] Akun

Register

Nama	<input type="text" value="Nama"/>	Email	<input type="text" value="Email"/>
username	<input type="text" value="admin"/>	No Telp	<input type="text" value="+62"/>
Password	<input type="password" value="....."/>	Konfirmasi Password	<input type="text" value="Konfirmasi Password"/>
<input type="button" value="Register"/>			

localhost:8080 / mysql-db / db; x Lutfia Bakery x +

localhost:8081/produk.php

+62882888888 lutfiabrian1@gmail.com lutfia bakery Indonesia

LUTFIA BAKERY Home Produk Tentang Kami Manual Aplikasi [0] Akun

Produk Kami



Sari roti



Maryam
Rp.15,000



Kue tart coklat
Rp.100,000

BAB III

PENUTUP

A. Kesimpulan

Dari semua yang telah dibahas, dapat disimpulkan bahwa menggunakan Docker untuk mengelola aplikasi web memberikan banyak manfaat, termasuk isolasi lingkungan yang konsisten, pengembangan yang lebih efisien, dan penyebaran yang mudah. Dengan Docker, pengguna dapat membuat lingkungan pengembangan yang bersih dan terisolasi, menjalankan berbagai layanan server dalam kontainer yang dapat diatur dengan cepat, dan memastikan portabilitas aplikasi di berbagai lingkungan. Penggunaan file Dockerfile dan docker-compose.yml memungkinkan pengaturan dan konfigurasi yang konsisten, yang membuat proses pengembangan dan penyebaran menjadi lebih terstruktur dan mudah dipelihara.

B. SARAN

Sebagai saran, penting untuk selalu mengikuti praktik terbaik dalam menggunakan Docker, seperti memahami konsep dasar Docker, memastikan keamanan kontainer, memonitor dan mengelola kontainer dengan baik, serta menjaga file konfigurasi dan skrip Docker tetap terorganisir dan terdokumentasi dengan baik. Selain itu, selalu perbarui dan perbaiki kontainer secara berkala untuk menghindari kerentanan keamanan dan menjaga ketersediaan aplikasi. Dengan mengikuti saran-saran ini, pengguna dapat memaksimalkan manfaat dari penggunaan Docker dalam pengembangan dan pengelolaan aplikasi web.