

**PEMBUATAN WEBSITE TO DO LIST
DAN MENGGUNAKAN DOCKER**



Disusun Oleh :

Ahmad Rafi

32602200035

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025**

DAFTAR ISI

DAFTAR ISI	ii
DAFTAR GAMBAR	iii
DAFTAR TABEL	iv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan.....	1
BAB II TINJAUAN PUSTAKA	2
2.1 To-do list	2
2.2 Laravel	2
2.3 Docker	2
2.4 Laragon	2
BAB III IMPLEMENTASI	3
3.1 Struktur Project	3
3.2 Docker-compose.yml	3
3.3 Dockerfile.....	3
3.4 Konfigurasi Environment (.env)	3
BAB IV HASIL DAN PEMBAHASAN	4
4.1 Instalasi Docker.....	4
4.2 Proses Implementasi	4
4.3 Tampilan Aplikasi	5
4.4 Unggah Aplikasi ke Dockerhub	5
4.5 Mengamankan Docker Image dari Tampering	6
BAB V PENUTUP	8
5.1 Kesimpulan	8
REFRENSI	9
LAMPIRAN.....	10

DAFTAR GAMBAR

Gambar 3. 1 Struktur Folder Project	3
Gambar 4. 1 Tampilan Docker Setelah di Install	4
Gambar 4. 2 Tampilan Docker Version	4
Gambar 4. 3 Docker PS.....	5
Gambar 4. 4 Tampilan Utama Websitenya.....	5
Gambar 4. 5 Tampilan Push Docker	6
Gambar 4. 6 Tampilan DockerHub	6
Gambar 4. 7 Gambar 4. 8 Fitur Docker Content Trust (DCT).....	7

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi informasi telah mengubah cara individu dan organisasi mengelola aktivitas sehari-hari. Aplikasi berbasis web, seperti to-do list, menjadi solusi populer karena kemudahan akses dan fungsionalitasnya. Aplikasi ini membantu mencatat, mengatur, dan memantau tugas, sehingga pekerjaan dapat direncanakan dengan lebih terstruktur dan prioritas lebih mudah ditentukan. Dalam lingkungan kerja, to-do list juga mendukung kolaborasi tim dan pelacakan progres pekerjaan (Randi 2025).

Namun, pengembangan aplikasi sering menghadapi kendala perbedaan konfigurasi antara sistem lokal dan server. Untuk itu, Docker hadir sebagai solusi dengan menyediakan lingkungan kerja yang konsisten melalui container. Laravel, sebagai framework PHP yang populer, menawarkan struktur kode rapi dan manajemen database yang efisien melalui migrasi dan Eloquent ORM. Ketika Laravel dipadukan dengan Docker, proses pengembangan menjadi lebih mudah dan siap untuk dijalankan di berbagai platform, termasuk cloud (Aryo Wibowo and Trias Hanggara 2023).

1.2 Tujuan

Adapun tujuan dari makalah ini :

1. Menjelaskan proses deployment aplikasi ke dalam container menggunakan Docker.
2. Menyajikan tahapan konfigurasi Dockerfile dan docker-compose.yml pada project Laravel yang sudah ada.
3. Memberikan gambaran tentang penggunaan Docker dalam pengemasan dan deployment aplikasi.

BAB II

TINJAUAN PUSTAKA

2.1 To-do list

Aplikasi to-do list merupakan salah satu bentuk aplikasi manajemen tugas yang bertujuan untuk membantu pengguna dalam mencatat, mengatur, dan memantau pekerjaan yang harus diselesaikan. Secara umum, aplikasi ini bekerja dengan menyediakan antarmuka bagi pengguna untuk menuliskan daftar tugas, memberi tenggat waktu (deadline), dan menandai tugas yang sudah diselesaikan (Imani, Pamungkas, and Jamaluddin 2024).

2.2 Laravel

Laravel adalah salah satu framework PHP yang paling populer dan banyak digunakan oleh para pengembang web. Framework ini dibuat untuk memudahkan proses pembuatan aplikasi web dengan menyediakan berbagai fitur yang lengkap, seperti sistem routing, autentikasi, manajemen database, hingga keamanan. (Sinlae et al. 2024).

2.3 Docker

Docker adalah sebuah platform yang digunakan untuk membuat, menjalankan, dan mengelola aplikasi di dalam container. Container ini bisa diibaratkan seperti "kotak" yang berisi semua hal yang dibutuhkan oleh sebuah aplikasi, mulai dari kode program, dependensi, hingga konfigurasi sistem, sehingga aplikasi bisa berjalan dengan cara yang sama di mana pun, baik di komputer pengembang, server, maupun cloud (Megantara et al. 2022).

2.4 Laragon

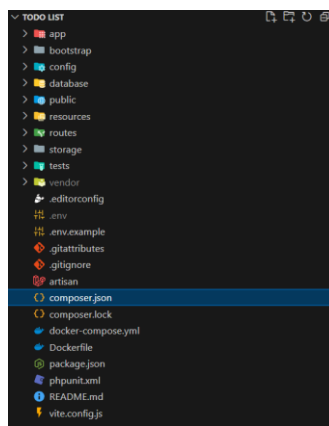
Laragon adalah software lokal development environment yang ringan dan mudah digunakan, khususnya untuk pengembangan aplikasi web berbasis PHP seperti Laravel. Dengan Laragon, pengguna bisa dengan cepat membuat server lokal lengkap dengan Apache/Nginx, MySQL, PHP, dan berbagai tool lain hanya dalam beberapa klik, tanpa perlu instalasi dan konfigurasi manual yang rumit (Andarsyah, Yuda Pratama, and Kishendrian 2022).

BAB III

IMPLEMENTASI

3.1 Struktur Project

Pada project ini yaitu pembuatan website catatan atau to-do list yang di konfigurasi dengan docker container, dibawah merupakan struktur projectnya.



Gambar 3. 1 Struktur *Folder Project*

3.2 Docker-compose.yml

File Docker-compose.yml untuk membuat dan menjalankan dua container :

1. Laravel app yang berjalan di localhost:8000
2. MySQL database yang terhubung dengan Laravel

3.3 Dockerfile

Dockerfile berfungsi untuk membuat image Docker secara otomatis dengan mendefinisikan semua kebutuhan aplikasi, seperti dependensi, file, dan perintah yang perlu dijalankan, sehingga mempermudah proses setup dan deployment.

3.4 Konfigurasi Environment (.env)

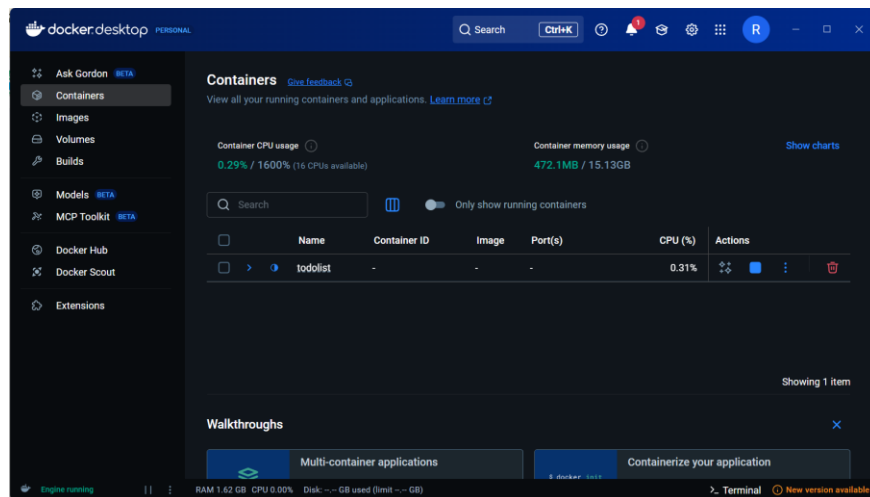
File .env digunakan untuk menyimpan konfigurasi environment seperti nama database, user, password, atau port yang dibutuhkan aplikasi dan berkaitan dengan docker-compose.yml.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Instalasi Docker

Untuk mengunduh docker bisa melalui website resmi docker, <https://www.docker.com/>, kemudian pilih docker desktop for windows, kemudian klik tombol download for windows dan simpan file installer di komputer.



Gambar 4. 1 Tampilan *Docker* Setelah di Install

Setelah instalasi, perintah `docker -v` digunakan untuk memastikan bahwa Docker telah terpasang dan berjalan dengan baik.

```
PS E:\todo list> docker -v
Docker version 28.3.0, build 38b7060
PS E:\todo list> |
```

Gambar 4. 2 Tampilan *Docker* Version

4.2 Proses Implementasi

Setelah konfigurasi docker selesai, jalankan :

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

Perintah `docker-compose up --build -d` digunakan untuk membangun ulang image Docker dari Dockerfile jika ada perubahan (`--build`), lalu menjalankan semua container yang didefinisikan di file `docker-compose.yml` (`up`). Kemudian cek status container :

```
docker ps
```


Hasilnya menampilkan daftar container aktif lengkap dengan nama, status, dan port mapping.

```
E:\todo list>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d49456bd63e	todolist-app	"docker-php-entrypoi..."	4 days ago	Up 2 minutes	0.0.0.0:8000->8000/tcp, [::]:8000->8000/tcp	laravel_app
a31cf9778dcc	mysql:8	"docker-entrypoint.s..."	4 days ago	Up 5 minutes	0.0.0.0:3306->3306/tcp, [::]:3306->3306/tcp	laravel_mys

```
ql
```

Gambar 4. 3 Docker PS

4.3 Tampilan Aplikasi

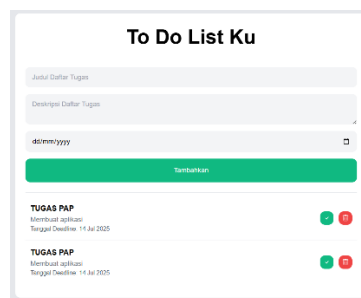
Setelah konfigurasi selesai, kemudian jalankan project nya dengan mengakses :

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

setelah container running kemudian akses alamat kemudian copy ke browser :

<http://localhost:8000>

Kemudian akan muncul tampilan halaman utama webstie to-do list, kalau website sudah mucul maka konfigurasi docker nya sudah benar.



Gambar 4. 4 Tampilan Utama Websitenya

4.4 Unggah Aplikasi ke Dockerhub

Sebelum unggah aplikasi ke dockerhub terlebih dahulu masuk ke akun docker

```
docker login
```

kemudian tag pada image lokal beasiswa-app agar sesuai dengan format penamaan DockerHub (username/nama-repo:tag). Ini penting agar image bisa dikenali saat proses push.

```
docker tag todolist-app:latest rafimars345/laravel-php83:latest
```

Kemudian unggah image yang sudah ditandai tadi ke repository DockerHub.

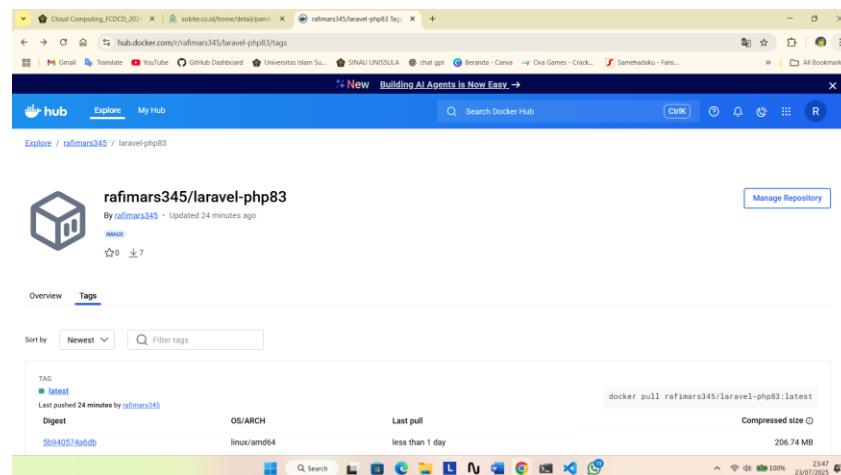
```
docker push rafimars345/laravel-php83:latest
```

```
E:\todo list>docker push rafimars345/laravel-php83:latest
The push refers to repository [docker.io/rafimars345/laravel-php83]
ae4752096b7b: Pushed
7d6417d18a80: Pushed
019c2b29e4af: Pushed
5f70bf18a086: Pushed
609e2e21d35c: Pushed
cb55ee1cfd8: Pushed
6ad9f0fbf4bb: Pushed
392ee7e58c4a: Pushed
e3609dd02db8: Pushed
f3e97e9460ff: Pushed
44801396762b: Pushed
1a2c067fb9f8: Pushed
e691f3ea106b: Pushed
af92c2132539: Pushed
20424b33f94c: Pushed
a483da8ab3e9: Pushed
latest: digest: sha256:5b940574a6db6e3a7011e53713ab665aa8ceb4f783717883da9c867747e37ac9 size: 3874
```

Gambar 4. 5 Tampilan *Push Docker*

Setelah di unggah link image bisa di akses melalui :

<https://hub.docker.com/r/rafimars345/laravel-php83/tags>



Gambar 4. 6 Tampilan *DockerHub*

Melalui tautan ini, siapa pun dapat melihat dan mengunduh image Docker bernama beasiswa-app yang telah dipublikasikan . Image tersebut dapat diunduh menggunakan perintah

```
docker pull rafimars345/laravel-php83:latest
```

4.5 Mengamankan Docker Image dari Tampering

Keamanan dalam distribusi Docker image sangat penting untuk memastikan bahwa image yang digunakan tidak dimodifikasi (tampered) oleh pihak yang tidak berwenang. Salah satu cara untuk mengamankan image adalah dengan mengaktifkan fitur Docker Content Trust (DCT).

Dengan Docker Content Trust, setiap image yang di-push atau di-pull akan diverifikasi menggunakan tanda tangan digital (digital signature). Ini

membantu memastikan bahwa image berasal dari sumber terpercaya dan belum dimodifikasi di luar kendali pengembang.

Untuk mengaktifkan fitur ini pada Windows PowerShell, digunakan perintah berikut :

```
$env:DOCKER_CONTENT_TRUST = "1"
```

```
PS E:\todo list> $env:DOCKER_CONTENT_TRUST = "1"
```

Gambar 4. 7 Gambar 4. 8 Fitur Docker Content Trust (DCT)

BAB V

PENUTUP

5.1 Kesimpulan

Implementasi Docker dalam pengembangan aplikasi to-do list berbasis Laravel memberikan banyak manfaat, terutama dalam hal konsistensi lingkungan kerja dan kemudahan deployment. Dengan menggunakan Docker, proses instalasi dan konfigurasi menjadi lebih sederhana karena semua kebutuhan aplikasi dikemas dalam container yang siap dijalankan kapan saja dan di mana saja tanpa perlu penyesuaian ulang.

Selain itu, Laravel yang digunakan sebagai framework inti juga memberikan struktur pengembangan yang rapi dan efisien, mulai dari manajemen database hingga pengelolaan routing dan autentikasi. Kombinasi antara Laravel dan Docker terbukti mampu mempercepat proses pengembangan serta meminimalkan risiko error akibat perbedaan sistem antara server lokal dan server produksi.

Secara keseluruhan, penggunaan Docker tidak hanya mempermudah proses pengembangan dan distribusi aplikasi, tetapi juga membuat aplikasi lebih siap untuk diimplementasikan pada lingkungan cloud atau server skala besar secara stabil dan efisien.

REFRENSI

- Andarsyah, Roni, Christian Yuda Pratama, and Hanan Destiarin Kishendrian. 2022. "Implementasi Code Coverage Pada Chatbot Telegram Sebagai Media Alternatif Sistem Informasi." *Jurnal Teknik Informatika* 14(2): 9568.
- Aryo Wibowo, Adhitya, and Buce Trias Hanggara. 2023. "Pembangunan Aplikasi Logistik Berbasis Web Dengan Integrasi Google Tasks API (Studi Kasus: Toko Alwan)." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* 7(7): 3145–52. <http://j-ptiik.ub.ac.id>.
- Imani, Zuriah Syahda, Achmad Fanani Ade Pamungkas, and Muhammad Jamaluddin. 2024. "Efektivitas Teknik To-Do List Terhadap Time Management Mahasiswa Psikologi UIN Malang Dalam Belajar." *Jurnal Psikologi* 1(4): 9.
- Megantara, Rama Aria, Farrikh Alzami, Ricardus Anggi Pramunendar, and Dwi Puji Prabowo. 2022. "Pengembangan Dan Implementasi Docker Untuk Memaksimalkan Utilitas Server Universitas Pada Masa Covid-19." *Transmisi* 24(2): 48–54.
- Randi, Randi Afif. 2025. "Rancang Bangun Aplikasi Manajemen Tugas Menggunakan Laravel." *Jurnal Teknologi Informasi Dan Komunikasi* 16(1): 6–13.
- Sinlae, Fried, Eko Irwanda, Zaky Maulana, and Vicky Eka Syahputra. 2024. "Penggunaan Framework Laravel Dalam Membangun Aplikasi Website Berbasis PHP." *Jurnal Siber Multi Disiplin* 2(2): 119–32.

LAMPIRAN

Link gituhub :

https://github.com/AhmadRafi345/32602200035_Tugas_Docker_CloudComputing_AhmadRafi.git