

**IMPLEMENTASI *DOCKER* PADA *WEBSITE* DAFTAR ADUAN  
MASYARAKAT(DAM) BERBASIS *LARAVEL***



**Disusun Oleh :**

**NAMA : ARJUN KHOIRUL ABROR**

**NIM : 32602200044**

**PRODI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG**

**2025**

## DAFTAR ISI

|                                      |            |
|--------------------------------------|------------|
| <b>HALAMAN JUDUL .....</b>           | <b>i</b>   |
| <b>DAFTAR ISI.....</b>               | <b>ii</b>  |
| <b>DAFTAR GAMBAR.....</b>            | <b>iii</b> |
| <b>PENDAHULUAN.....</b>              | <b>1</b>   |
| 1.1    Latar Belakang.....           | 1          |
| 1.2    Tujuan Penulisan .....        | 1          |
| <b>PEMBAHASAN .....</b>              | <b>2</b>   |
| 2.1 <i>Laravel</i> .....             | 2          |
| 2.2 <i>Docker</i> .....              | 2          |
| 2.3 <i>Docker Compose</i> .....      | 2          |
| <b>IMPLEMENTASI .....</b>            | <b>3</b>   |
| 3.1    Struktur <i>Project</i> ..... | 3          |
| 3.2    Pemilihan Teknologi.....      | 3          |
| 3.3    Implementasi Sistem.....      | 3          |
| <b>HASIL DAN PEMBAHASAN .....</b>    | <b>4</b>   |
| 4.1    Proses Implementasi .....     | 4          |
| 4.2    Tampilan Aplikasi .....       | 4          |
| <b>PENUTUP.....</b>                  | <b>5</b>   |
| 5.1    Kesimpulan.....               | 5          |
| <b>REFERENSI.....</b>                |            |
| <b>LAMPIRAN.....</b>                 |            |

## **DAFTAR GAMBAR**

|   |   |
|---|---|
| Gambar 3. 1 Struktur Folder Project .....           | 3 |
| Gambar 4. 1 Daftar Container Aktif.....             | 4 |
| Gambar 4. 2 Tampilan Halaman Utama Website DAM..... | 4 |

# PENDAHULUAN

## 1.1 Latar Belakang

Dalam era perkembangan teknologi informasi yang pesat, tuntutan terhadap proses pengembangan perangkat lunak yang lebih cepat, portabel, dan efisien menjadi sebuah keniscayaan. Laravel telah memantapkan dirinya sebagai salah satu framework PHP yang sangat populer, menawarkan kemudahan dalam membangun aplikasi web yang kompleks dan modern. Meskipun demikian, proses konfigurasi lingkungan pengembangan untuk Laravel yang melibatkan pengaturan dependensi seperti PHP, Composer, Nginx, dan MySQL secara manual sering kali menjadi tantangan teknis.

Docker memungkinkan pengembang untuk membungkus aplikasi beserta seluruh dependensinya ke dalam sebuah kontainer (container) yang ringan dan terisolasi. Dengan pendekatan ini, proyek Laravel dapat dijalankan secara konsisten di berbagai lingkungan komputasi tanpa memerlukan konfigurasi ulang yang rumit. Laporan ini mendokumentasikan implementasi Docker pada sebuah proyek website To-Do-List yang sudah ada. Proyek ini di-deploy menggunakan arsitektur *multi-container* yang mencakup layanan untuk Laravel, Nginx, dan MySQL. Perlu ditekankan bahwa fokus utama laporan ini adalah pada proses integrasi dan deployment aplikasi Laravel menggunakan Docker dan Docker Compose, bukan pada pengembangan fitur aplikasi itu sendiri.

## 1.2 Tujuan Penulisan

Tujuan dari penulisan makalah ini adalah sebagai berikut:

1. Menjelaskan proses implementasi *Docker* pada *project Laravel*.
2. Menyajikan tahapan konfigurasi *Dockerfile* dan *docker-compose.yml* pada *project Laravel* yang sudah ada.
3. Menunjukkan keberhasilan deployment aplikasi *Laravel* ke dalam *environment Docker multi-container*

## PEMBAHASAN

### 2.1 *Laravel*

Laravel adalah sebuah framework aplikasi web berbasis PHP yang menggunakan pola arsitektur *Model-View-Controller* (MVC). *Framework* ini dirancang untuk menyederhanakan dan mempercepat proses pengembangan dengan menyediakan serangkaian fitur *esensial*, seperti sistem *routing* yang elegan, *middleware*, ORM (*Object-Relational Mapping*) bernama *Eloquent*, mekanisme migrasi basis data, serta antarmuka baris perintah (CLI) *Artisan* yang kuat.

### 2.2 *Docker*

*Docker* merupakan sebuah *platform open-source* yang berfungsi untuk membangun, mendistribusikan, dan menjalankan aplikasi di dalam sebuah unit terisolasi yang disebut kontainer. Setiap kontainer mengemas kode aplikasi beserta seluruh pustaka dan dependensi yang dibutuhkanannya. Hal ini menjamin bahwa aplikasi dapat berjalan dengan cara yang sama dan konsisten di lingkungan mana pun, mulai dari laptop pengembang hingga server produksi, tanpa perlu instalasi manual yang rentan kesalahan

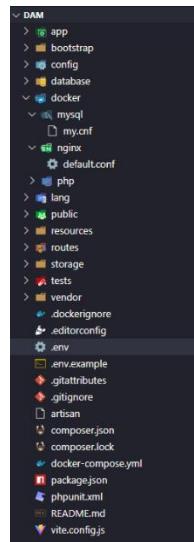
### 2.3 *Docker Compose*

*Docker Compose* adalah sebuah alat bantu (tool) yang melengkapi ekosistem *Docker*, dirancang khusus untuk mendefinisikan dan menjalankan aplikasi yang terdiri dari banyak kontainer (*multi-container*). Melalui sebuah file konfigurasi tunggal berformat YAML bernama `docker-compose.yml`, pengembang dapat mengatur seluruh layanan aplikasi (misalnya, kontainer untuk web server, aplikasi, dan basis data). Dengan satu perintah, *Docker Compose* akan membangun dan menjalankan keseluruhan arsitektur layanan yang saling terhubung secara otomatis.

## IMPLEMENTASI

### 3.1 Struktur *Project*

*Project Laravel DAM* ini di-clone dari *repository* pihak ketiga dan kemudian dikonfigurasi ke dalam *environment Docker*.



Gambar 3. 1 Struktur Folder *Project*

### 3.2 Pemilihan Teknologi

Laravel dipilih karena mendukung pengembangan aplikasi web modern dengan fitur MVC, routing, dan keamanan bawaan. *MySQL* digunakan sebagai basis data relasional. *Docker* dan *Docker Compose* mempermudah *deploy multi-container* secara konsisten. *Spatie Laravel Permission* digunakan untuk mengatur hak akses berbasis peran (*role*).

### 3.3 Implementasi Sistem

Aplikasi dikembangkan dengan Laravel dan dijalankan dalam tiga *container* (*Laravel*, *Nginx*, *MySQL*) menggunakan *Docker Compose*. Struktur *database* dibuat dengan *migration* dan data awal dimasukkan lewat *seeder*. Fitur utama seperti input laporan, *upload* dokumentasi, dan autentikasi role dikembangkan dan diuji langsung dalam lingkungan *Docker* untuk memastikan kestabilan dan integrasi antar komponen.

## HASIL DAN PEMBAHASAN

### 4.1 Proses Implementasi

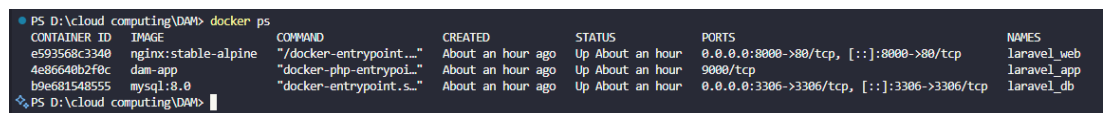
Setelah konfigurasi *Docker* selesai, jalankan:

```
Docker-compose up -d
```

Perintah ini memanggil *Dockerfile* dan *Docker-compose.yml* untuk membuat dan menjalankan 3 *service*: *Laravel App*, *MySQL*, dan *Nginx*. Cek status container dengan *Status container* dapat diperiksa dengan perintah:

```
Docker ps
```

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



| CONTAINER ID | IMAGE               | COMMAND                  | CREATED           | STATUS           | PORTS                                       | NAMES       |
|--------------|---------------------|--------------------------|-------------------|------------------|---|-------------|
| e593568c3340 | nginx:stable-alpine | "/docker-entrypoint..."  | About an hour ago | Up About an hour | 0.0.0.0:8000->80/tcp, [::]:8000->80/tcp     | laravel_web |
| 4e86648b2f0c | dam-app             | "docker-php-entryp..."   | About an hour ago | Up About an hour | 9000/tcp                                    | laravel_app |
| b9e681548555 | mysql:8.0           | "docker-entrypoint.s..." | About an hour ago | Up About an hour | 0.0.0.0:3306->3306/tcp, [::]:3306->3306/tcp | laravel_db  |

Gambar 4. 1 Daftar *Container* Aktif

Setelah itu, untuk melakukan migrasi *database*, dijalankan perintah:

```
Docker exec -it Laravel-app php artisan migrate
```

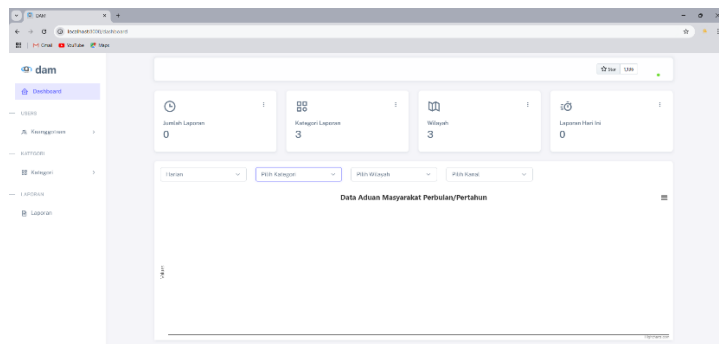
Proses ini memastikan semua tabel *Laravel* berhasil dibuat di dalam container *MySQL*.

### 4.2 Tampilan Aplikasi

Setelah *container* berjalan dan migrasi sukses, aplikasi *Laravel* dapat diakses melalui *browser* pada alamat:

```
http://localhost:8080
```

Halaman utama *website* DAM akan tampil, menandakan *deployment* berhasil dan *Laravel* berjalan normal di *Docker*.



Gambar 4. 2 Tampilan Halaman Utama *Website* DAM

## PENUTUP

### 5.1 Kesimpulan

Sistem pelaporan digital kepolisian yang dibangun menggunakan Laravel dan *Docker* telah berhasil diimplementasikan sebagai solusi untuk pengelolaan laporan masyarakat yang lebih modern, efisien, dan terdokumentasi dengan baik. Dengan memanfaatkan *framework* Laravel, pengembangan aplikasi menjadi lebih terstruktur dan cepat, sementara *Docker* membantu menciptakan lingkungan kerja yang konsisten serta mempermudah proses *deployment* dan pengujian di berbagai mesin tanpa konfigurasi manual yang kompleks.

Aplikasi ini mendukung autentikasi pengguna berbasis peran (*role-based access control*), memungkinkan diferensiasi hak akses antara petugas dan admin. Proses *input* laporan masyarakat telah disesuaikan agar mendukung pengunggahan dokumentasi (gambar/foto), serta disimpan secara aman di server.

Selain itu, arsitektur berbasis *container* memungkinkan sistem ini mudah diintegrasikan dengan layanan lain di masa mendatang, seperti *chatbot* pelaporan via WhatsApp, analitik laporan menggunakan AI, dan sistem notifikasi otomatis. Semua komponen aplikasi, termasuk kode, konfigurasi *Docker*, dan dokumentasi teknis telah disediakan dalam satu repositori GitHub agar mempermudah kolaborasi pengembangan lebih lanjut.

Melalui pengembangan sistem ini, diharapkan kepolisian dapat lebih *responsif* terhadap laporan masyarakat, mengurangi kesalahan pencatatan, serta meningkatkan kualitas pelayanan publik secara menyeluruh. Proyek ini membuktikan bahwa pemanfaatan teknologi modern dapat secara signifikan meningkatkan efisiensi dalam sistem pelayanan masyarakat, khususnya di institusi publik seperti kepolisian.



## REFERENSI

- [1] Dokumentasi *Laravel*. *Laravel Documentation*. <https://Laravel.com/docs>
- [2] Dokumentasi *Docker*. *Docker Docs*. <https://docs.Docker.com>
- [3] Repository Implementasi *Docker*  
[https://github.com/arjunkhoirul/ArjunKhoirulAbror\\_TugasDocker.git](https://github.com/arjunkhoirul/ArjunKhoirulAbror_TugasDocker.git)

## **LAMPIRAN**

Nama: Arjun Khoirul Abror

NIM: 32602200044

Link Project Github:

[https://github.com/arjunkhoirul/ArjunKhoirulAbror\\_TugasDocker.git](https://github.com/arjunkhoirul/ArjunKhoirulAbror_TugasDocker.git)