

**DOKUMENTASI DAN IMPLEMENTASI APLIKASI WEB PHP &
MYSQL MENGGUNAKAN DOCKER**



Disusun oleh:
AGUNG WIDY SAPUTRO
(32602200032)

PRODI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG

2025

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan makalah yang berjudul “Dokumentasi dan Implementasi Aplikasi Web PHP & MySQL Menggunakan Docker” ini dengan baik dan tepat waktu.

Makalah ini disusun sebagai dokumentasi teknis dari proses pengembangan dan *containerization* sebuah aplikasi web sederhana, yaitu kalkulator biaya listrik "HitungWatt". Tujuan utama dari penulisan makalah ini adalah untuk memberikan panduan langkah demi langkah yang jelas mengenai cara membungkus aplikasi berbasis PHP dan MySQL ke dalam lingkungan Docker sehingga menjadi portabel, konsisten, dan mudah untuk didistribusikan.

Penulis menyadari bahwa makalah ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga makalah ini dapat bermanfaat dan memberikan wawasan baru bagi para pembaca, khususnya yang tertarik pada teknologi pengembangan perangkat lunak modern.

Semarang, 14 Juli 2025



Agung Widy Saputro

DAFTAR ISI

HALAMAN JUDUL	1
KATA PENGANTAR	2
DAFTAR ISI	3
BAB I PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	4
1.3 Tujuan Penulisan	4
BAB II LANDASAN TEORI	5
2.1 Aplikasi Web PHP & MySQL	5
2.2 Teknologi Docker	5
2.3 Dockerfile	5
2.4 Docker Compose	5
BAB III PEMBAHASAN & IMPLEMENTASI	6
3.1 Persiapan Struktur Proyek	6
3.2 Membuat Dockerfile untuk Layanan Web	6
3.3 Membuat Docker Compose untuk Orkestrasi	7
3.4 Penyesuaian Kode Aplikasi	8
3.5 Menjalankan Aplikasi	8
3.6 Hasil Akhir	9
BAB IV PENUTUP	10
4.1 Kesimpulan	10
4.2 Saran	10
4.3 Link Github.....	10
DAFTAR PUSTAKA	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengembangan aplikasi web modern seringkali dihadapkan pada tantangan "berjalan di komputer saya, tapi tidak di komputer lain". Perbedaan konfigurasi sistem operasi, versi *runtime* (seperti PHP), dan layanan (seperti database MySQL) dapat menyebabkan aplikasi gagal berjalan saat dipindahkan dari lingkungan pengembangan ke produksi.

Docker hadir sebagai solusi untuk masalah ini dengan teknologi *containerization*. Docker memungkinkan pengembang untuk membungkus aplikasi beserta seluruh dependensinya ke dalam sebuah unit standar yang disebut *container*. *Container* ini bersifat portabel dan dapat berjalan secara konsisten di lingkungan mana pun yang telah terinstal Docker.

Studi kasus dalam makalah ini adalah aplikasi "HitungWatt", sebuah kalkulator biaya listrik sederhana yang dibangun menggunakan PHP dan MySQL. Proses "dockerisasi" aplikasi ini akan didokumentasikan sebagai contoh praktis penerapan teknologi Docker dalam siklus pengembangan perangkat lunak.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah yang akan dibahas adalah sebagai berikut:

1. Bagaimana cara mempersiapkan struktur proyek aplikasi web untuk proses dockerisasi?
2. Bagaimana cara membuat `Dockerfile` untuk layanan web berbasis PHP-Apache?
3. Bagaimana cara menggunakan `Docker Compose` untuk menjalankan layanan web dan database MySQL secara bersamaan?
4. Bagaimana cara menjalankan dan melakukan inisialisasi database pada Docker?

1.3 Tujuan Penulisan

Tujuan dari penulisan makalah ini adalah:

1. Menyediakan panduan untuk melakukan dockerisasi aplikasi web PHP dan MySQL.
2. Mendokumentasikan konfigurasi `Dockerfile` dan `docker-compose.yml` yang diperlukan.
3. Menjadi referensi praktis bagi pengembang lain yang ingin mulai menggunakan Docker dalam proyek mereka.

BAB II

LANDASAN TEORI

2.1 Aplikasi Web PHP & MySQL

Aplikasi yang menjadi studi kasus adalah "HitungWatt", yang terdiri dari dua komponen utama:

1. Frontend & Backend (PHP): Kode PHP yang menangani logika bisnis (perhitungan biaya), menampilkan antarmuka pengguna (HTML & CSS), dan berkomunikasi dengan database.
2. Database (MySQL): Sistem manajemen basis data relasional yang digunakan untuk menyimpan data riwayat perhitungan.

2.2 Teknologi Docker

Docker adalah platform perangkat lunak yang memungkinkan pembuatan, pengujian, dan penerapan aplikasi dengan cepat menggunakan *container*. *Container* membungkus perangkat lunak dalam unit standar yang memiliki semua yang dibutuhkan untuk menjalankannya, termasuk kode, *runtime*, *system tools*, dan *library*.

2.3 Dockerfile

`Dockerfile` adalah sebuah file teks yang berisi serangkaian instruksi untuk membangun sebuah *image* Docker secara otomatis. Setiap instruksi dalam `Dockerfile` akan menciptakan sebuah *layer* baru pada *image*, sehingga proses *build* menjadi efisien.

2.4 Docker Compose

Docker Compose adalah alat untuk mendefinisikan dan menjalankan aplikasi Docker multi-kontainer. Dengan sebuah file konfigurasi berformat YAML (`docker-compose.yml`), kita dapat mengkonfigurasi semua layanan aplikasi (seperti server web dan database) dan menjalankannya dengan satu perintah sederhana.

BAB III

PEMBAHASAN & IMPLEMENTASI

3.1 Persiapan Struktur Proyek

Langkah pertama adalah menata semua file proyek ke dalam satu direktori utama. Struktur ini penting agar Docker dapat menemukan semua file yang dibutuhkan. [SCREENSHOT: Struktur folder proyek di File Explorer yang menunjukkan file `index.php`, `koneksi.php`, `lamp.jpg`, `Dockerfile`, dan `docker-compose.yml`]

3.2 Membuat Dockerfile untuk Layanan Web

Dockerfile ini akan menjadi resep untuk membangun *image* server web kita. File ini berbasis `php:apache`, menginstal ekstensi `mysqli` yang dibutuhkan, dan menyalin kode aplikasi ke dalamnya.

File: `Dockerfile`

```
DockerFile > ...
1 FROM php:8.2-apache
2 RUN docker-php-ext-install mysqli && docker-php-ext-enable mysqli
3 COPY . /var/www/html/
```

3.3 Membuat Docker Compose untuk Orkestrasi

`docker-compose.yml` adalah pusat kendali aplikasi kita. File ini mendefinisikan dua layanan utama: `web` dan `db`. Konfigurasi ini juga mencakup pengaturan kredensial database dan penambahan `command` khusus untuk mengatasi masalah kompatibilitas autentikasi MySQL 8.

File: `docker-compose.yml`

```
docker-compose.yml
1 services:
2   web:
3     build: .
4     ports:
5       - "8080:80"
6     volumes:
7       - ../var/www/html/
8     depends_on:
9       - db
10
11   db:
12     image: mysql:8.0
13     restart: always
14     command: --default-authentication-plugin=mysql_native_password
15     environment:
16       MYSQL_ROOT_PASSWORD: password_rahasia
17       MYSQL_DATABASE: db_listrik
18       MYSQL_USER: AgungWidysaputro
19       MYSQL_PASSWORD: Aws280803
20     volumes:
21       - db_data:/var/lib/mysql
22
23 volumes:
24   db_data:
```

3.4 Penyesuaian Kode Aplikasi

File `koneksi.php` perlu diubah. Di lingkungan Docker Compose, nama host untuk database bukan lagi `localhost`, melainkan nama layanan yang didefinisikan di `docker-compose.yml`, yaitu `db`.

File: `koneksi.php`

```

koneksi.php
1  <?php
2  $host = "db";
3  $user = "AgungWidySaputro";
4  $pass = "Aws280803";
5  $db = "db_listrik";
6
7  $koneksi = mysqli_connect(hostname: $host, username: $user, password: $pass, database: $db);
8
9  if (!$koneksi) {
10     die("Koneksi Gagal: " . mysqli_connect_error());
11 }
12 ?>

```

3.5 Menjalankan Aplikasi dengan Docker

Dengan semua file konfigurasi siap, aplikasi dijalankan menggunakan satu perintah dari terminal di dalam direktori proyek.

```
C:\xampp\htdocs\HitungWatt>docker-compose up -d
```

Perintah ini akan membangun *image* (jika belum ada) dan memulai semua layanan yang didefinisikan. Setelah kontainer berjalan, database `db_listrik` akan tercipta tetapi masih kosong. Kita perlu membuat tabel `tb_perhitungan` secara manual.

1. Masuk ke dalam kontainer database: `docker-compose exec db mysql -u`

`AgungWidySaputro -p` (Masukkan password `Aws280803` saat diminta)

2. Pilih database dan buat tabel: `USE db_listrik; CREATE TABLE tb_perhitungan (...);`

```

mysql> USE db_listrik;
Database changed
mysql> CREATE TABLE `tb_perhitungan` ( `id` int(11) NOT NULL AUTO_INCREMENT, `nama_alat` varchar(100) NOT NULL, `daya_watt` int(11) NOT NULL, `lama_pakai_jam` float NOT NULL, `jumlah_hari` int(11) NOT NULL, `tarif_per_kwh` decimal(10,2) NOT NULL, `total_biaya` decimal(10,2) NOT NULL, `tanggal_hitung` timestamp NOT NULL DEFAULT current_timestamp(), PRIMARY KEY (`id`) );
Query OK, 0 rows affected, 3 warnings (0.04 sec)

```

3.6 Hasil Akhir

Setelah tabel dibuat, aplikasi dapat diakses melalui browser di alamat `http://localhost:8080` dan semua fungsionalitasnya berjalan dengan baik, termasuk penyimpanan riwayat ke database.

HitungWatt ⚡
Prediksi Tagihan Listrik Anda

Nama Alat

Daya Alat (Watt)

Lama Pemakaian (Jam per Hari)

Jumlah Hari Pemakaian

Tarif Listrik per kWh (Rp)

Hitung Biaya

HitungWatt ⚡
Prediksi Tagihan Listrik Anda

Nama Alat

Daya Alat (Watt)

Lama Pemakaian (Jam per Hari)

Jumlah Hari Pemakaian

Tarif Listrik per kWh (Rp)

Hitung Biaya

Jumlah Hari Pemakaian

Tarif Listrik per kWh (Rp)

Hitung Biaya

Perkiraan Biaya untuk **kulkas** selama **30 hari** adalah:
Rp 234.041,40

Riwayat Perhitungan Terakhir

Alat	Daya	Pemakaian	Total Biaya
kulkas	900 Watt	6 jam/hari	Rp 234.041,40

BAB IV

PENUTUP

4.1 Kesimpulan

Proses dockerisasi aplikasi web "HitungWatt" telah berhasil dilakukan. Dengan menggunakan `Dockerfile` dan `Docker Compose`, aplikasi yang terdiri dari layanan PHP-Apache dan database MySQL dapat dijalankan dalam lingkungan yang terisolasi dan konsisten. Teknologi ini secara efektif menghilangkan masalah perbedaan konfigurasi antar mesin dan menyederhanakan proses deployment. Proses ini menunjukkan bahwa Docker adalah alat yang sangat kuat untuk pengembangan perangkat lunak modern.

4.2 Saran

Untuk pengembangan lebih lanjut, proses inisialisasi database dapat dibuat lebih otomatis. Daripada membuat tabel secara manual menggunakan `docker-compose exec`, kita dapat menempatkan file `.sql` yang berisi perintah `CREATE TABLE` ke dalam direktori `/docker-entrypoint-initdb.d/`. Dengan cara ini, MySQL akan secara otomatis menjalankan skrip tersebut saat pertama kali diinisialisasi, membuat proses setup menjadi sepenuhnya otomatis.

4.3 Link Github

<https://github.com/AgungWidySaputro28/HitungWatt.git>

DAFTAR PUSTAKA

- Docker Documentation. (2025). *Docker Compose overview*. Diakses dari <https://docs.docker.com/compose/>
- PHP. (2025). *PHP: Hypertext Preprocessor*. Diakses dari <https://www.php.net/>
- MySQL. (2025). *MySQL Documentation*. Diakses dari <https://dev.mysql.com/doc/>