**MAKALAH**

DEPLOYMENT DAN DISTRIBUSI APLIKASI RESEPKU BERBASIS DOCKER DAN DOCKER HUB



Disusun oleh:

Muhammad Rafli Aldian Jamil

(32602300044)

**UNIVERSITAS ISLAM SULTAN AGUNG SEMARANG**

**FAKULTAS TEKNOLOGI INDUSTRI**

**PRODI TEKNIK INFORMATIKA**

**2025**

# Daftar Isi

# BAB I PENDAHULUAN

## Latar Belakang

Di era digital, kebutuhan akan aplikasi manajemen resep semakin meningkat. Banyak orang ingin menyimpan, mencari, dan mengelola resep masakan secara digital agar lebih mudah diakses dan tidak mudah hilang. Namun, deployment aplikasi web seringkali menghadapi tantangan seperti perbedaan environment, dependency, dan konfigurasi server. Docker hadir sebagai solusi untuk mengemas aplikasi beserta seluruh dependensinya ke dalam container, sehingga deployment menjadi lebih mudah, konsisten, dan dapat direplikasi di berbagai platform. Docker Hub memudahkan distribusi image aplikasi ke berbagai server atau anggota tim.

## Rumusan Masalah

Bagaimana membangun, mendistribusikan, dan mendemokan aplikasi manajemen resep berbasis web menggunakan Docker dan Docker Hub agar deployment menjadi lebih mudah dan konsisten?

## Tujuan

1. Membangun aplikasi manajemen resep berbasis web.
2. Melakukan deployment aplikasi menggunakan Docker.
3. Mendistribusikan aplikasi ke Docker Hub.
4. Mendemokan aplikasi secara lokal.

# BAB II LANDASAN TEORI

## Docker

Docker adalah platform open-source yang memungkinkan developer mengemas aplikasi beserta seluruh dependensinya ke dalam sebuah container. Container ini dapat dijalankan di berbagai environment tanpa perlu konfigurasi ulang.

## Docker Compose

Docker Compose adalah tool untuk mendefinisikan dan menjalankan multi-container Docker applications menggunakan file YAML. Dengan Compose, developer dapat mengatur beberapa service sekaligus, seperti aplikasi, database, dan tool pendukung lainnya.

## Docker Hub

Docker Hub adalah layanan registry image Docker yang memungkinkan developer menyimpan dan mendistribusikan image secara online. Dengan Docker Hub, image aplikasi dapat di-pull dan dijalankan di mana saja.

## MariaDB

MariaDB adalah sistem manajemen basis data relasional yang merupakan fork dari MySQL. MariaDB banyak digunakan karena performa dan kompatibilitasnya yang tinggi.

## Flask

Flask adalah framework web berbasis Python yang ringan dan mudah digunakan untuk membangun aplikasi web. Flask mendukung pengembangan aplikasi dengan arsitektur modular dan fleksibel.

## PhpMyAdmin

PhpMyAdmin adalah aplikasi berbasis web untuk mengelola database MySQL/MariaDB. Dengan phpMyAdmin, pengguna dapat membuat, mengedit, dan menghapus database serta tabel secara visual.

# BAB III METODOLOGI

Penelitian ini menggunakan metode kualitatif dengan pendekatan rekayasa perangkat lunak. Metode kualitatif dipilih karena penelitian ini berfokus pada proses pengembangan, deployment, dan distribusi aplikasi secara deskriptif, tanpa melibatkan pengumpulan data numerik atau statistik.

Tahapan metodologi yang dilakukan meliputi:

## Analisis Kebutuhan

Pada tahap ini dilakukan identifikasi kebutuhan fungsional dan non-fungsional aplikasi, seperti:

1. Pengguna dapat menambah, mencari, mengedit, menghapus, dan melihat resep.
2. Aplikasi dapat dijalankan secara containerized.
3. Database terintegrasi dan mudah dikelola melalui phpMyAdmin.
4. Deployment dan distribusi aplikasi mudah dilakukan.

## Perancangan Sistem

Tahap ini meliputi:

1. Perancangan arsitektur aplikasi (menggunakan Docker Compose untuk multi-container).
2. Perancangan struktur folder proyek.
3. Perancangan Dockerfile untuk membungkus aplikasi Flask.
4. Perancangan docker-compose.yml untuk mengatur service Flask, MariaDB, dan phpMyAdmin.

## Implementasi

ada tahap ini dilakukan:

1. Pembuatan kode program aplikasi dengan Flask.
2. Pembuatan Dockerfile dan docker-compose.yml.
3. Build image aplikasi.
4. Testing lokal menggunakan Docker Compose.
5. Push image ke Docker Hub.

## Pengujian

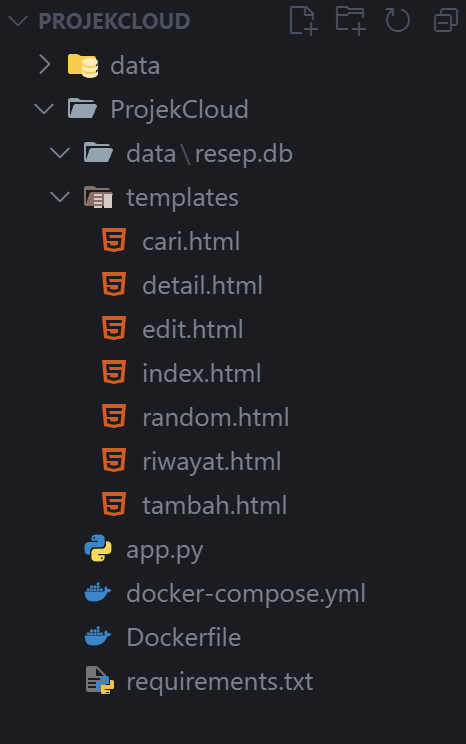
Aplikasi diuji secara lokal menggunakan Docker Compose untuk memastikan semua fitur berjalan dengan baik, serta diuji deployment-nya dengan menarik image dari Docker Hub. Pengujian dilakukan dengan menjalankan aplikasi, mengakses fitur-fitur utama, dan memastikan integrasi antar service berjalan lancar.

Setiap tahapan didokumentasikan secara naratif dan didukung dengan screenshot serta penjelasan langkah-langkah yang dilakukan.

# BAB IV IMPLEMENTASI

## Striktur Folder Proyek

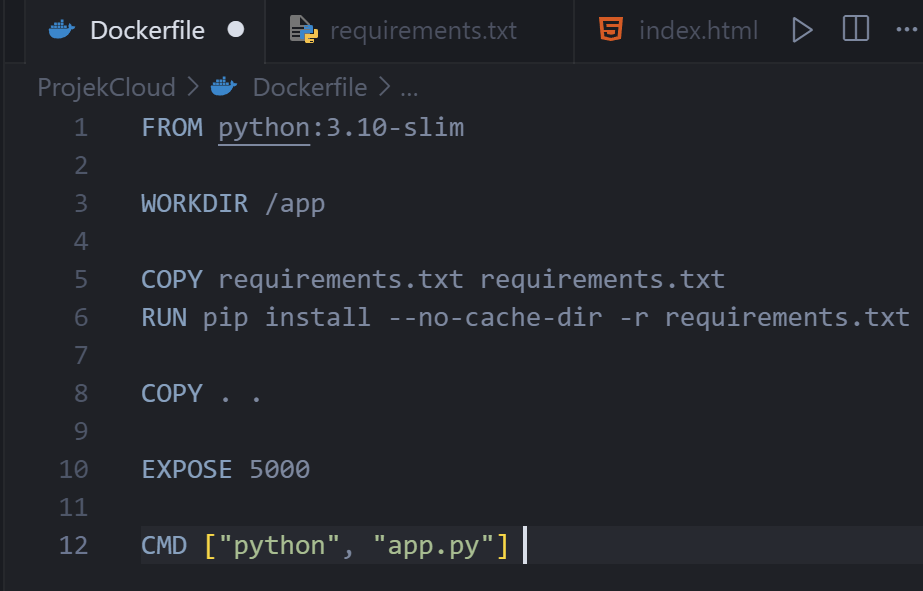
Struktur folder aplikasi Resepku adalah sebagai berikut:



Gambar 4. 1 Struktur folder proyek

## Pembuatan Dockerfile

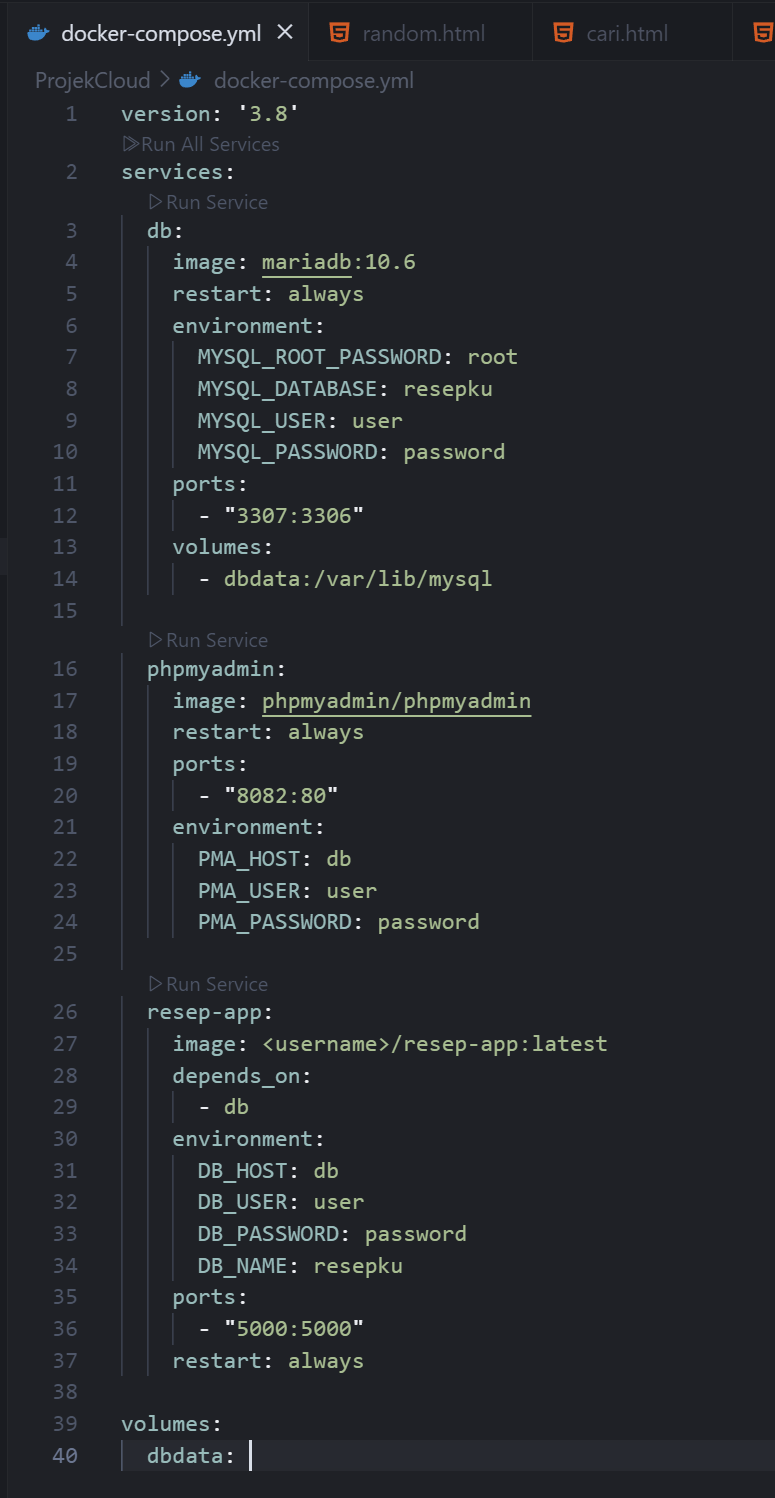
Dockerfile digunakan untuk membungkus aplikasi Flask ke dalam sebuah image. Berikut adalah isi Dockerfile:



Gambar 4. 2 Isi file Dockerfile

## Pembuatan docker-compose.yml

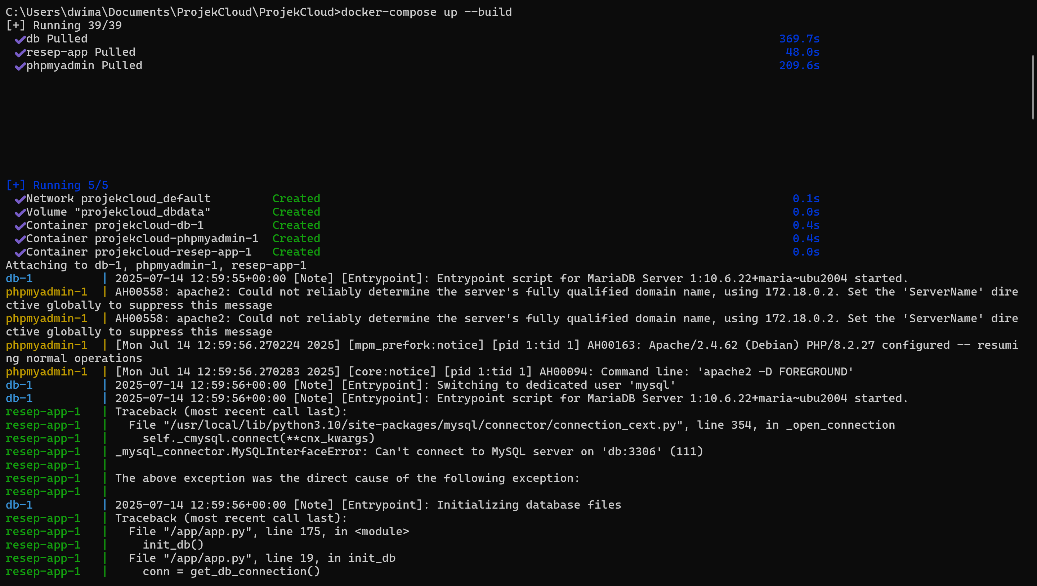
File docker-compose.yml digunakan untuk mengatur tiga service utama, yaitu aplikasi Flask, database MariaDB, dan phpMyAdmin.



Gambar 4. 3 Isi file docker-compose.yml

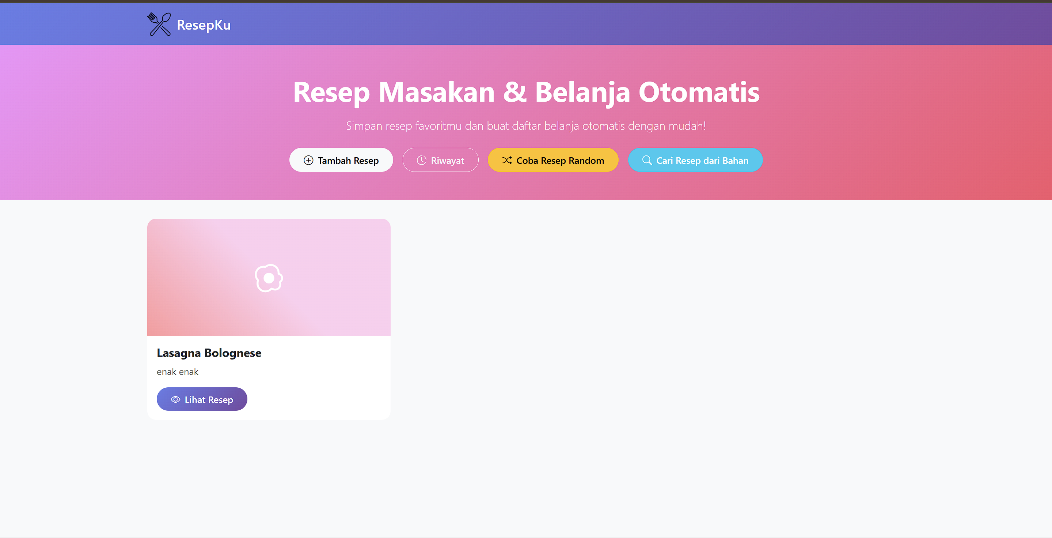
## Build dan Testing Lokal

Build dan testing aplikasi dilakukan dengan perintah docker-compose up –build.



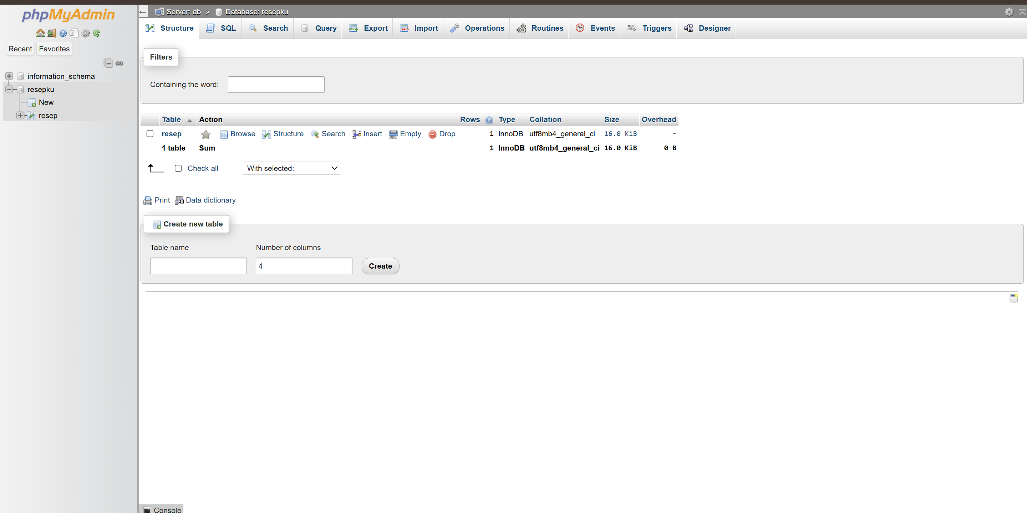
Gambar 4. 4 Hasil build di terminal

Setelah build selesai, aplikasi dapat diakses melalui browser pada alamat http://localhost:5000 dan phpMyAdmin pada <http://localhost:8082>.



Gambar 4. 5 Tampilan home aplikasi

Gambar 4.5 menunjukkan tampilan halaman utama (home) aplikasi Resepku setelah berhasil dijalankan di browser pada alamat http://localhost:5000. Pada halaman ini, pengguna dapat melihat daftar resep yang sudah tersimpan di database. Terdapat juga navigasi ke fitur-fitur lain seperti tambah resep, cari resep, random resep, dan riwayat pencarian. Tampilan home ini menjadi pusat akses utama bagi pengguna untuk mengelola dan mencari resep yang diinginkan.

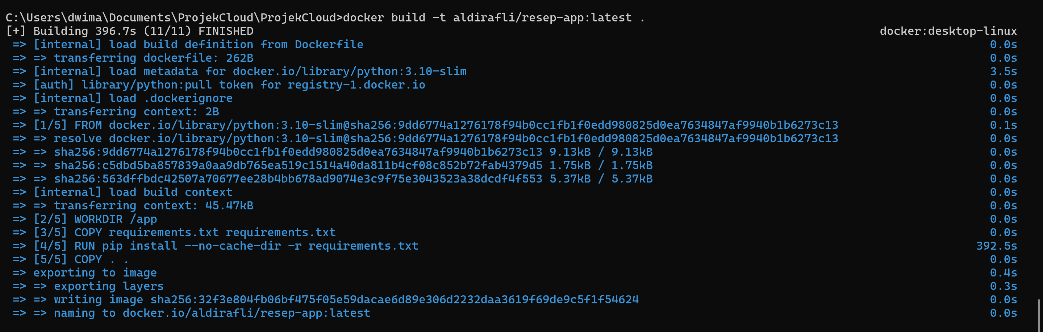


Gambar 4. 6 Tampilan phpMyAdmin

Gambar 4.6 memperlihatkan tampilan phpMyAdmin yang diakses melalui http://localhost:8082. phpMyAdmin digunakan untuk memudahkan pengelolaan database MariaDB secara visual. Pada tampilan ini, pengguna dapat melihat daftar database, tabel-tabel yang ada, serta melakukan operasi seperti menambah, mengedit, atau menghapus data secara langsung. phpMyAdmin sangat membantu dalam proses administrasi database tanpa perlu menggunakan perintah SQL secara manual.

## Build dan Push ke Docker Hub

Setelah aplikasi berjalan baik secara lokal, langkah selanjutnya adalah build image untuk Docker Hub:



Gambar 4. 7 Proses Build Image di Terminal

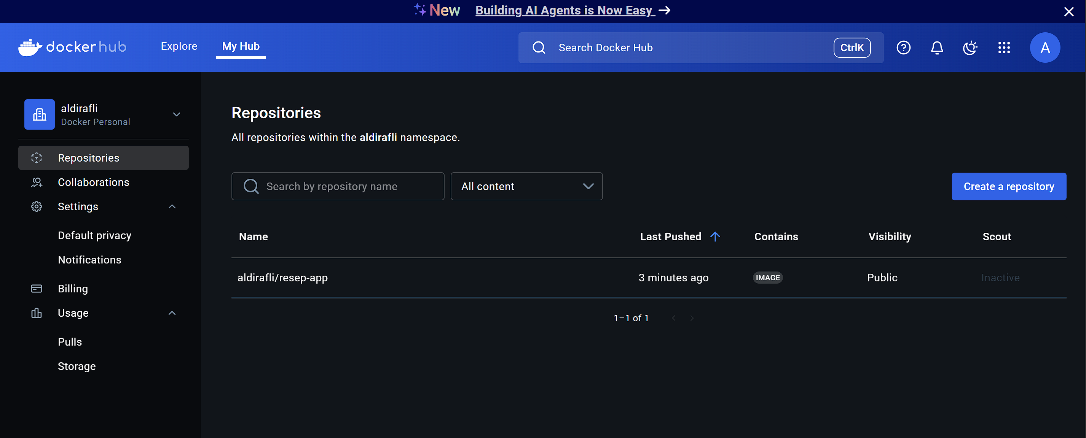
Proses ini dilakukan di terminal pada direktori proyek. Pada tahap ini, Docker akan membaca instruksi dari file Dockerfile, menginstall dependencies, menyalin source code, dan menghasilkan sebuah image aplikasi yang siap untuk dijalankan atau di-push ke Docker Hub. Output di terminal menampilkan tahapan build dan status keberhasilan build image.



Gambar 4. 8 Proses push image ke Docker Hub

Gambar 4.8 memperlihatkan proses push image aplikasi ke Docker Hub menggunakan perintah:

Pada tahap ini, image yang sudah berhasil dibuild akan diunggah ke repository Docker Hub milik pengguna. Proses push ini memungkinkan image aplikasi dapat diakses dan digunakan di komputer atau server lain tanpa perlu build ulang. Output di terminal menunjukkan progress upload dan konfirmasi jika push berhasil.



Gambar 4. 9 Repository Aplikasi di Docker Hub

Gambar 4.9 menampilkan halaman repository aplikasi di Docker Hub melalui browser. Pada halaman ini, dapat dilihat bahwa image aplikasi dengan tag latest sudah berhasil diunggah dan tersedia secara publik atau privat sesuai pengaturan repository. Informasi seperti nama repository, tag, ukuran image, dan waktu upload juga dapat dilihat di halaman ini. Hal ini menandakan bahwa aplikasi sudah siap untuk didistribusikan dan di-deploy di lingkungan lain.

## Update docker-compose.yml

Setelah image tersedia di Docker Hub, file docker-compose.yml diupdate agar service resep-app menggunakan image dari Docker Hub:

services:

db:

image: mariadb:10.6

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_DATABASE: resepku

MYSQL\_USER: user

MYSQL\_PASSWORD: password

ports:

- "3307:3306"

volumes:

- dbdata:/var/lib/mysql

phpmyadmin:

image: phpmyadmin/phpmyadmin

restart: always

ports:

- "8082:80"

environment:

PMA\_HOST: db

PMA\_USER: user

PMA\_PASSWORD: password

resep-app:

image: aldirafli/resep-app:latest

depends\_on:

- db

environment:

DB\_HOST: db

DB\_USER: user

DB\_PASSWORD: password

DB\_NAME: resepku

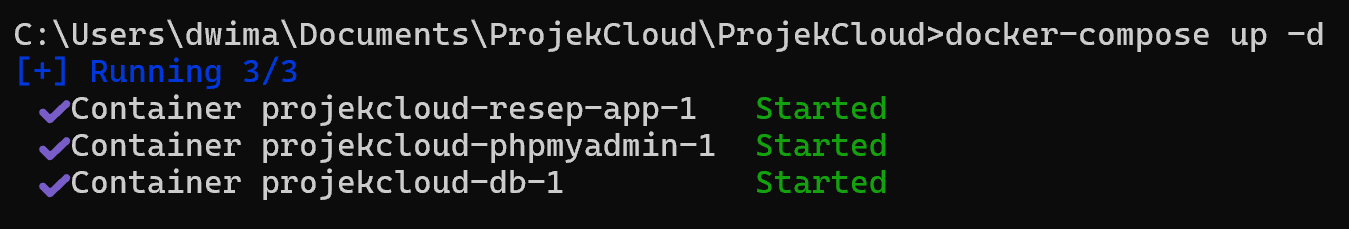
ports:

- "5000:5000"

restart: always

volumes:

dbdata:

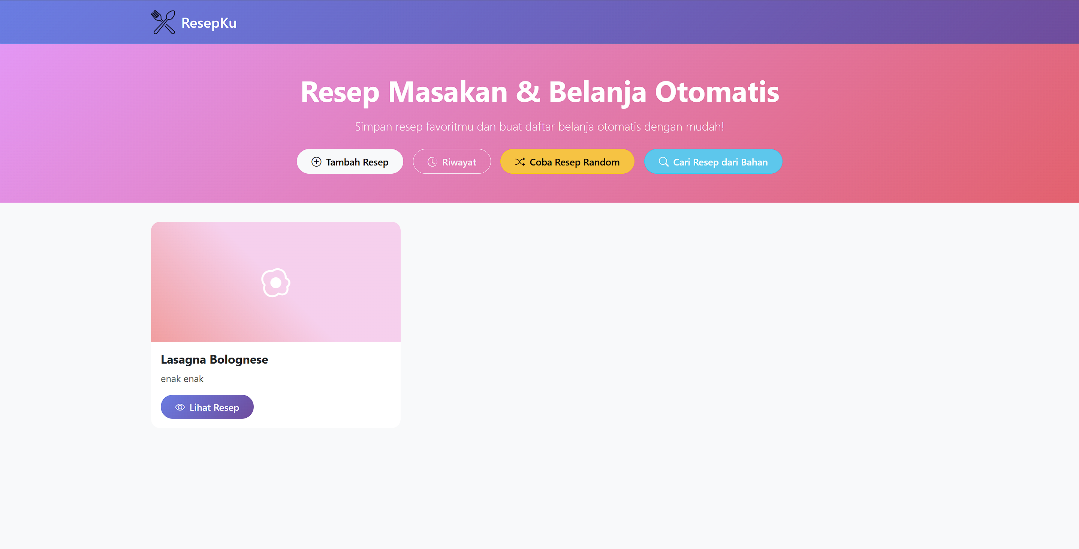
Lalu jalankan aplikasi dengan docker-compose up -d

Gambar 4. 10 Tampilan Saat Menjalankan Aplikasi

## Penjelasan Fitur Aplikasi

Aplikasi Resepku memiliki beberapa fitur utama, yaitu:

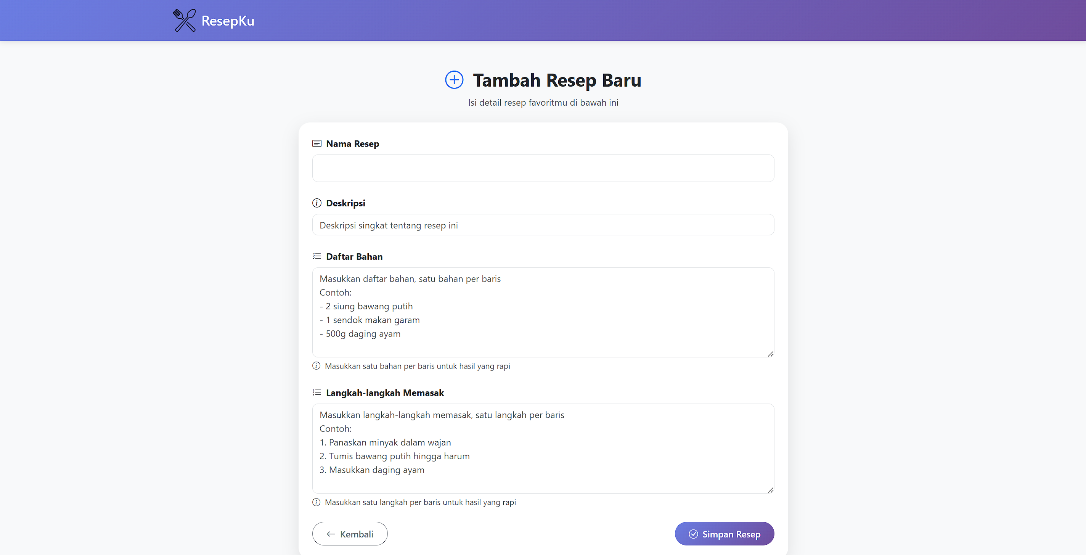
### Tampilan Home



Gambar 4. 11 Tampilan Home Aplikasi

Gambar ini menunjukkan halaman utama aplikasi Resepku yang diakses melalui http://localhost:5000. Pada halaman ini, pengguna dapat melihat daftar resep yang sudah tersimpan di database. Terdapat juga menu navigasi ke fitur-fitur lain seperti tambah resep, cari resep, random resep, dan riwayat pencarian. Tampilan home ini menjadi pusat akses utama bagi pengguna untuk mengelola dan mencari resep.

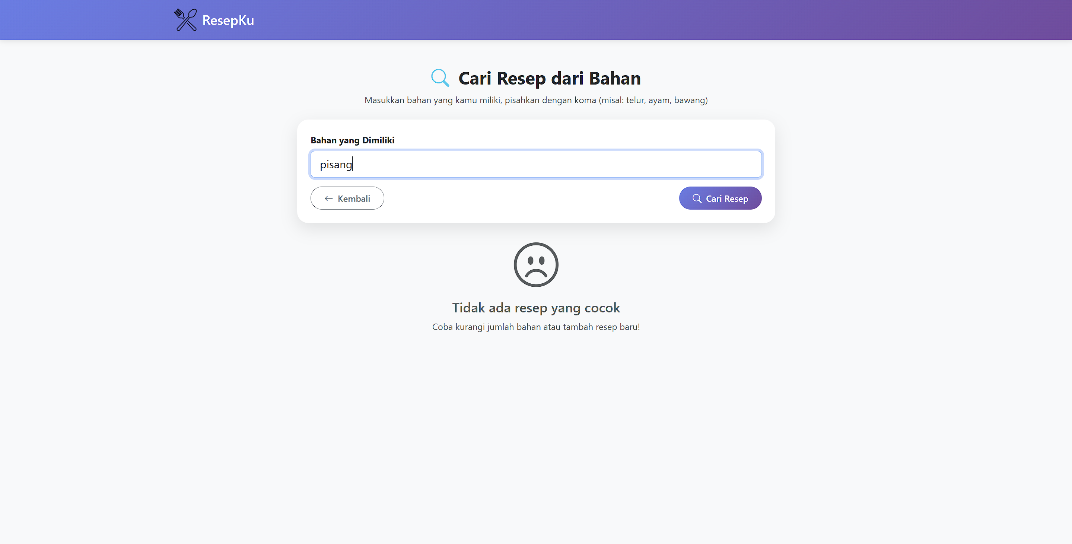
### Tambah Resep



Gambar 4. 12 Tampilan Tambah Resep

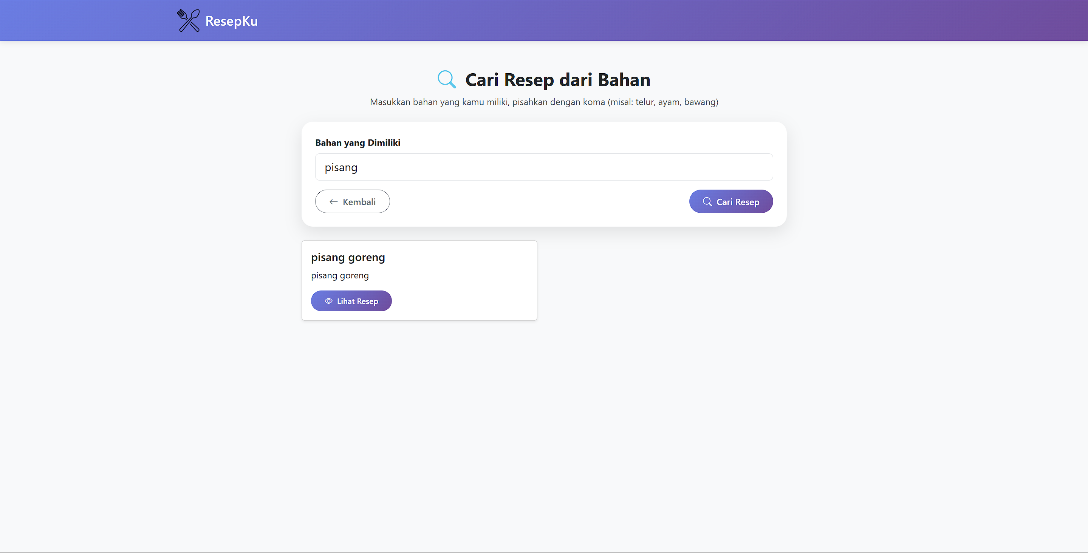
Gambar 4.12 memperlihatkan form tambah resep yang dapat diakses melalui menu “Tambah Resep”. Pengguna dapat mengisi nama resep, bahan-bahan, dan langkah-langkah pembuatan pada form ini.

### Cari Resep



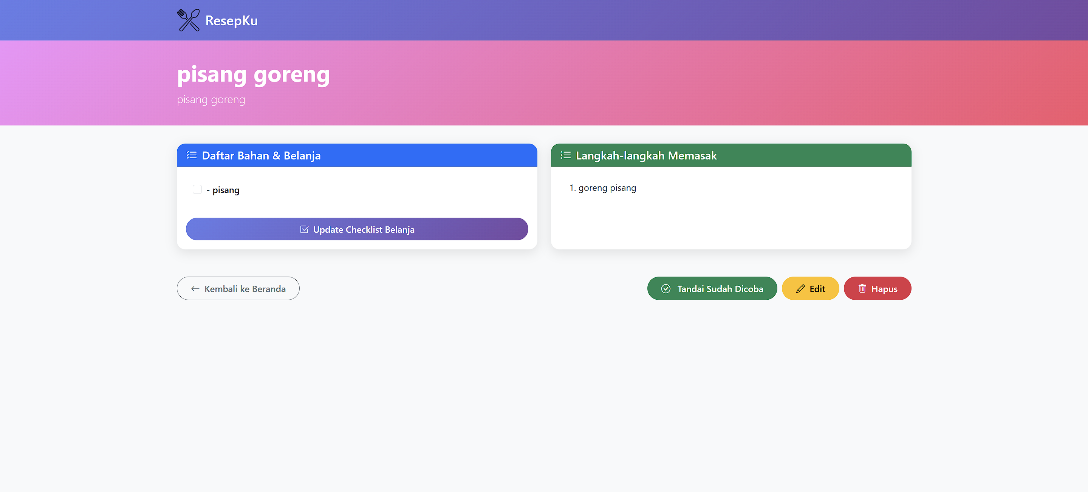
Gambar 4. 13 Tampilan Cari Resep

Gambar 4.13 menampilkan halaman pencarian resep, di mana pengguna dapat memasukkan kata kunci untuk mencari resep berdasarkan nama atau bahan tertentu.



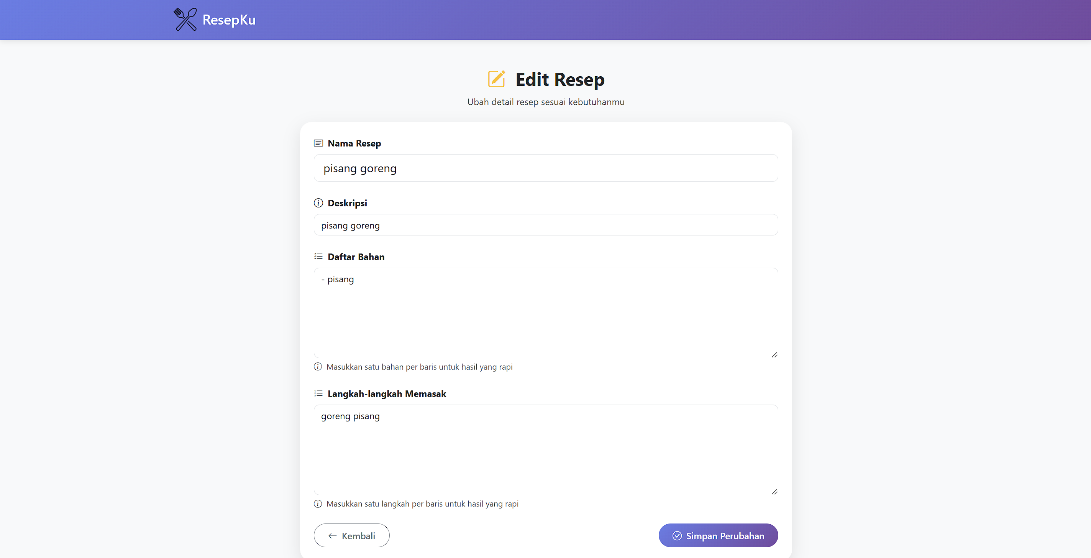
Gambar 4. 14 Tampilan Hasil Pencarian Resep

Setelah melakukan pencarian, aplikasi akan menampilkan daftar hasil resep yang sesuai dengan kata kunci yang dimasukkan. Pengguna dapat memilih salah satu resep untuk melihat detailnya.



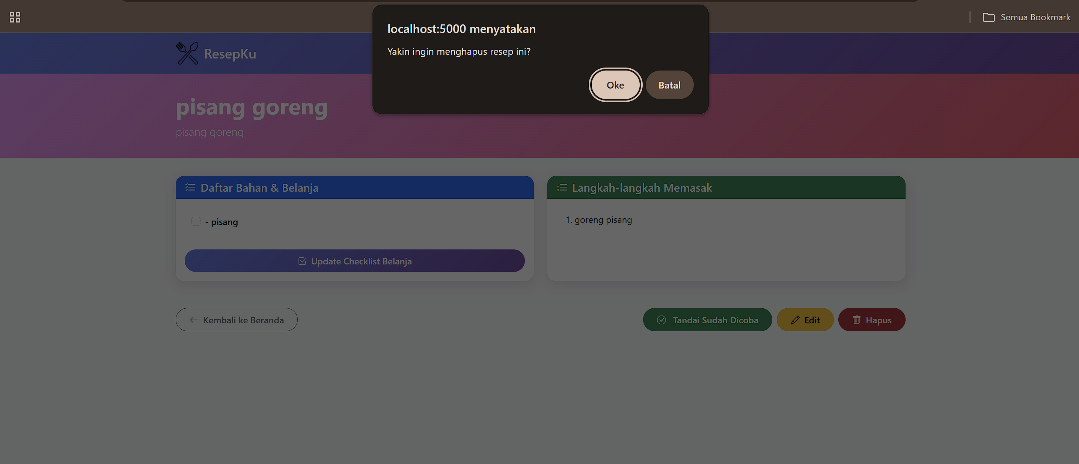
Gambar 4. 15 Tampilan Detail Resep

Gambar ini menunjukkan halaman detail resep yang menampilkan informasi lengkap mengenai resep yang dipilih, seperti nama resep, bahan-bahan, dan langkah-langkah pembuatan.



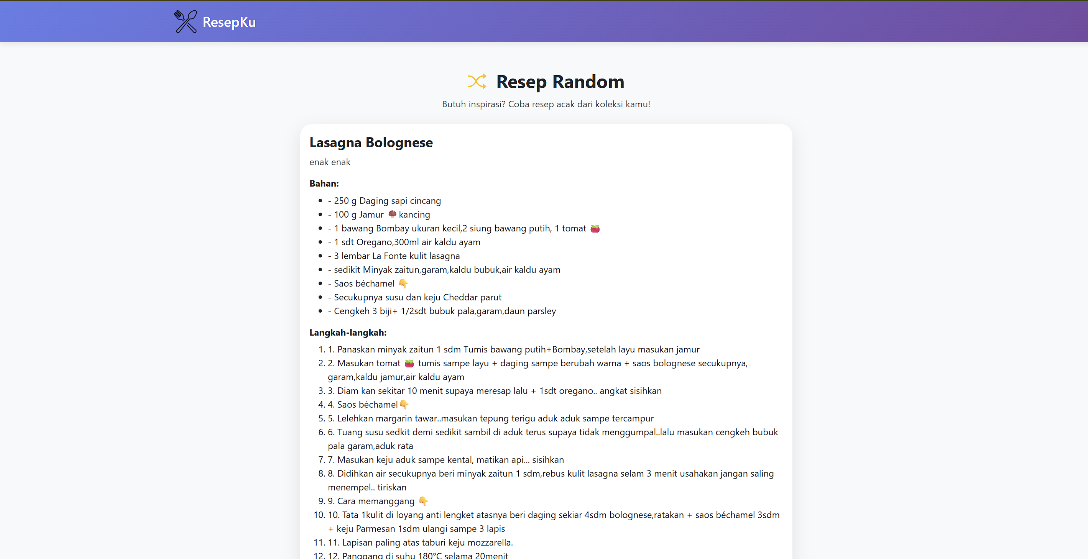
Gambar 4. 16 Form Edit Resep

Gambar ini memperlihatkan form edit resep yang dapat diakses dari halaman detail resep. Pengguna dapat memperbarui informasi resep yang sudah ada melalui form ini.



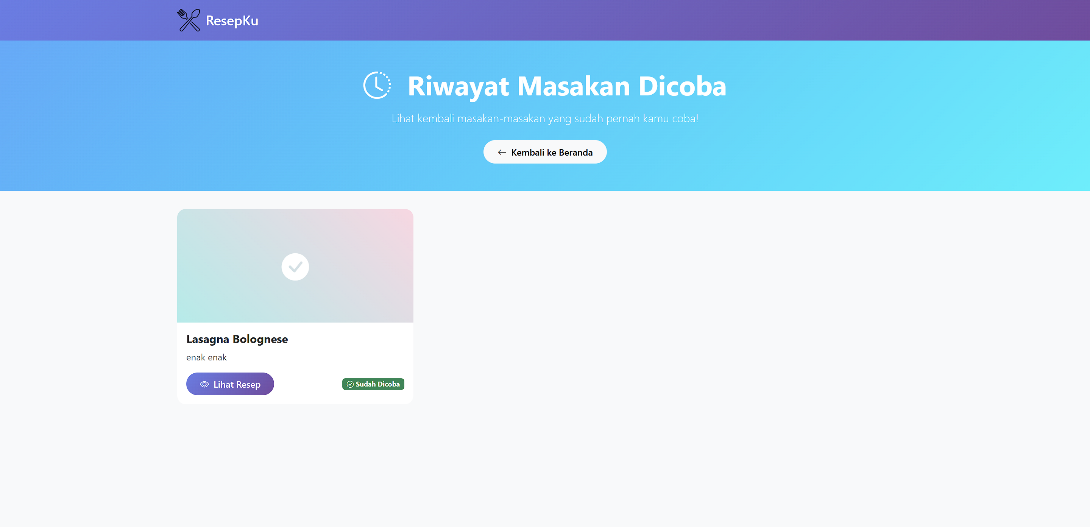
Gambar 4. 17 Tampilan Hapus Resep

Gambar ini menampilkan tampilan konfirmasi sebelum menghapus resep. Fitur ini memastikan bahwa pengguna benar-benar ingin menghapus resep dari database.



Gambar 4. 18 Tampilan Random Resep

Gambar ini memperlihatkan fitur random resep, di mana aplikasi akan menampilkan satu resep secara acak dari database. Fitur ini membantu pengguna yang bingung ingin memasak apa.



Gambar 4. 19 Tampilan Riwayat Pencarian

Gambar 4.19 menampilkan halaman riwayat pencarian, di mana pengguna dapat melihat daftar kata kunci atau resep yang pernah dicari sebelumnya. Fitur ini memudahkan pengguna untuk mengakses kembali resep yang pernah dicari.

# BAB V HASIL DAN PEMBAHASAN

## Hasil

Pada bab ini dijelaskan hasil dari proses implementasi, deployment, serta pengujian aplikasi Resepku yang telah dilakukan secara bertahap. Setiap tahapan didokumentasikan dengan narasi dan screenshot untuk memperjelas hasil yang diperoleh.

### Build dan Testing Lokal

Setelah seluruh file konfigurasi dan kode aplikasi selesai dibuat, proses build dan testing dilakukan secara lokal menggunakan Docker Compose. Perintah yang digunakan adalah:

docker-compose up –build

Hasil build ditunjukkan pada Gambar 4.4, di mana seluruh service berhasil dijalankan tanpa error. Aplikasi dapat diakses melalui browser pada alamat http://localhost:5000 dan phpMyAdmin pada http://localhost:8082.

### Tampilan Home Aplikasi

Setelah aplikasi berjalan, halaman utama aplikasi Resepku dapat diakses. Pada Gambar 4.5, terlihat daftar resep yang sudah tersimpan di database, serta menu navigasi ke fitur-fitur lain. Hal ini menunjukkan bahwa integrasi antara aplikasi Flask dan database MariaDB berjalan dengan baik.

### Pengelolaan Database dengan phpMyAdmin

Pada Gambar 4.6, tampak tampilan phpMyAdmin yang digunakan untuk mengelola database MariaDB. Melalui phpMyAdmin, dapat dilihat struktur tabel, data resep, serta melakukan operasi CRUD secara visual. Hal ini memudahkan proses administrasi database tanpa perlu perintah SQL manual.

### Build dan Push Image ke Docker Hub

Proses build image aplikasi dilakukan dengan perintah:

docker build -t aldirafli/resep-app:latest .

Hasil build image ditunjukkan pada Gambar 4.7. Setelah itu, image di-push ke Docker Hub menggunakan perintah:

docker push aldirafli/resep-app:latest

Proses push ini berhasil ditunjukkan pada Gambar 4.8. Verifikasi di Docker Hub dapat dilihat pada Gambar 4.9, di mana image aplikasi sudah tersedia di repository Docker Hub.

### Update dan Deploy dari Docker Hub

Setelah image tersedia di Docker Hub, file docker-compose.yml diupdate agar service resep-app menggunakan image dari Docker Hub. Aplikasi kemudian dijalankan kembali dan dapat diakses seperti sebelumnya, menandakan proses distribusi dan deployment berjalan lancar.

### Pengujian Fitur Aplikasi

Setiap fitur aplikasi diuji dan didokumentasikan sebagai berikut:

1. Tampilan Home: Menampilkan daftar resep dan menu navigasi (Gambar 4.11).
2. Tambah Resep: Form tambah resep dapat digunakan(Gambar 4.12).
3. Cari Resep: Fitur pencarian menampilkan hasil sesuai kata kunci (Gambar 4.13 dan 4.14).
4. Detail Resep: Informasi lengkap resep dapat diakses (Gambar 4.15).
5. Edit Resep: Form edit dapat digunakan untuk memperbarui data resep (Gambar 4.16).
6. Hapus Resep: Konfirmasi hapus muncul sebelum data dihapus (Gambar 4.17).
7. Random Resep: Fitur random menampilkan resep acak (Gambar 4.18).
8. Riwayat Pencarian: Daftar riwayat pencarian tampil dengan baik (Gambar 4.19).

## Pembahasan

### Analisis Proses Deployment

Proses deployment menggunakan Docker dan Docker Hub memberikan banyak keuntungan, di antaranya:

1. Konsistensi Lingkungan: Dengan Docker, aplikasi dapat dijalankan di berbagai sistem operasi tanpa perlu khawatir perbedaan environment.
2. Kemudahan Distribusi: Image aplikasi yang sudah di-push ke Docker Hub dapat di-pull dan dijalankan di komputer atau server mana pun dengan mudah.
3. Manajemen Multi-Container: Docker Compose memudahkan pengelolaan beberapa service sekaligus (aplikasi, database, phpMyAdmin) dalam satu perintah.

### Analisis Fitur Aplikasi

Setiap fitur aplikasi berjalan sesuai dengan kebutuhan pengguna:

1. CRUD Resep: Pengguna dapat menambah, mencari, mengedit, dan menghapus resep dengan mudah.
2. Pencarian dan Random: Fitur pencarian dan random resep memudahkan pengguna menemukan inspirasi masakan.
3. Riwayat Pencarian: Fitur ini membantu pengguna mengakses kembali resep yang pernah dicari.
4. phpMyAdmin: Memudahkan pengelolaan database secara visual, sangat membantu untuk debugging dan administrasi data.

### Kendala dan Solusi

Beberapa kendala yang dihadapi selama proses implementasi antara lain:

1. Error pada build/push image: Biasanya disebabkan oleh kesalahan penulisan nama image atau belum login ke Docker Hub. Solusi: pastikan nama image benar dan sudah login.
2. Masalah port bentrok: Jika port yang digunakan sudah dipakai aplikasi lain, perlu mengganti port di file docker-compose.yml.
3. Integrasi antar service: Pastikan environment variable dan dependensi antar service sudah benar diatur di docker-compose.

### Evaluasi

Secara keseluruhan, aplikasi Resepku berhasil di-deploy dan didistribusikan dengan baik menggunakan Docker dan Docker Hub. Semua fitur berjalan sesuai harapan, dan deployment menjadi lebih mudah, cepat, dan konsisten.

# BAB VI KESIMPULAN

Berdasarkan hasil implementasi, pengujian, dan pembahasan yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi Resepku berhasil dibangun dan diimplementasikan menggunakan framework Flask untuk backend, MariaDB sebagai database, serta phpMyAdmin sebagai alat bantu administrasi database. Seluruh fitur utama aplikasi, seperti tambah, cari, edit, hapus, random resep, dan riwayat pencarian, dapat berjalan dengan baik sesuai kebutuhan pengguna.
2. Penggunaan Docker dan Docker Compose sangat memudahkan proses deployment aplikasi. Dengan mengemas aplikasi dan seluruh dependensinya ke dalam container, proses setup menjadi lebih cepat, konsisten, dan minim error, baik di lingkungan pengembangan maupun produksi.
3. Distribusi aplikasi melalui Docker Hub terbukti efektif untuk membagikan image aplikasi ke berbagai komputer atau server. Proses build, push, dan pull image dapat dilakukan dengan mudah, sehingga aplikasi dapat dijalankan di mana saja tanpa perlu konfigurasi ulang yang rumit.
4. Manajemen multi-container dengan Docker Compose memungkinkan integrasi yang lancar antara aplikasi, database, dan phpMyAdmin. Semua service dapat dijalankan, dihentikan, dan dikelola secara bersamaan hanya dengan satu perintah, sehingga sangat efisien untuk pengembangan dan deployment.
5. Pengujian aplikasi menunjukkan bahwa seluruh fitur berjalan sesuai harapan. Pengguna dapat mengelola resep dengan mudah, melakukan pencarian, melihat detail, mengedit, menghapus, serta mendapatkan rekomendasi resep secara acak. Fitur riwayat pencarian juga menambah kenyamanan pengguna dalam mengakses resep yang pernah dicari.
6. phpMyAdmin sangat membantu dalam administrasi database, baik untuk keperluan pengujian, debugging, maupun pemeliharaan data. Pengelolaan database menjadi lebih mudah dan visual tanpa perlu perintah SQL manual.
7. Dokumentasi dan penjelasan langkah-langkah deployment yang detail sangat penting untuk memastikan proses dapat direplikasi oleh siapa saja. Dengan dokumentasi yang baik, anggota tim lain atau pengguna baru dapat dengan mudah memahami dan menjalankan aplikasi.
8. Kendala yang muncul selama proses implementasi, seperti error pada build image, push ke Docker Hub, atau pengaturan environment, dapat diatasi dengan troubleshooting yang tepat. Hal ini menunjukkan pentingnya pemahaman terhadap tools yang digunakan serta ketelitian dalam mengikuti setiap langkah.
9. Penggunaan teknologi container seperti Docker sangat relevan untuk pengembangan aplikasi modern, terutama dalam konteks DevOps, Continuous Integration, dan Continuous Deployment (CI/CD). Pengalaman ini menjadi bekal penting untuk pengembangan aplikasi skala lebih besar di masa depan.
10. Secara keseluruhan, proyek ini membuktikan bahwa deployment aplikasi berbasis web dapat dilakukan dengan cara yang efisien, terstandarisasi, dan mudah didistribusikan menggunakan Docker dan Docker Hub, sehingga aplikasi siap digunakan dan dikembangkan lebih lanjut.

# DAFTAR PUSTAKA

Docker Documentation. https://docs.docker.com/

Docker Hub. https://hub.docker.com/

Flask Documentation. https://flask.palletsprojects.com/

MariaDB Documentation. https://mariadb.com/kb/en/documentation/

phpMyAdmin Documentation. https://www.phpmyadmin.net/docs/

[Link GitHub Proyek] https://github.com/username/nama-repo