

Aplikasi Resep Full-Stack

Dibangun dengan Vue.js, PHP (Slim Framework), & Docker

Presentasi Proyek

Tim Pengembang



Abdullah Azzam Rabbani



Achmar Ichsan Anthony



La Ode Achmed Sayyed Purnomo



Yudistira Agung Kurnia

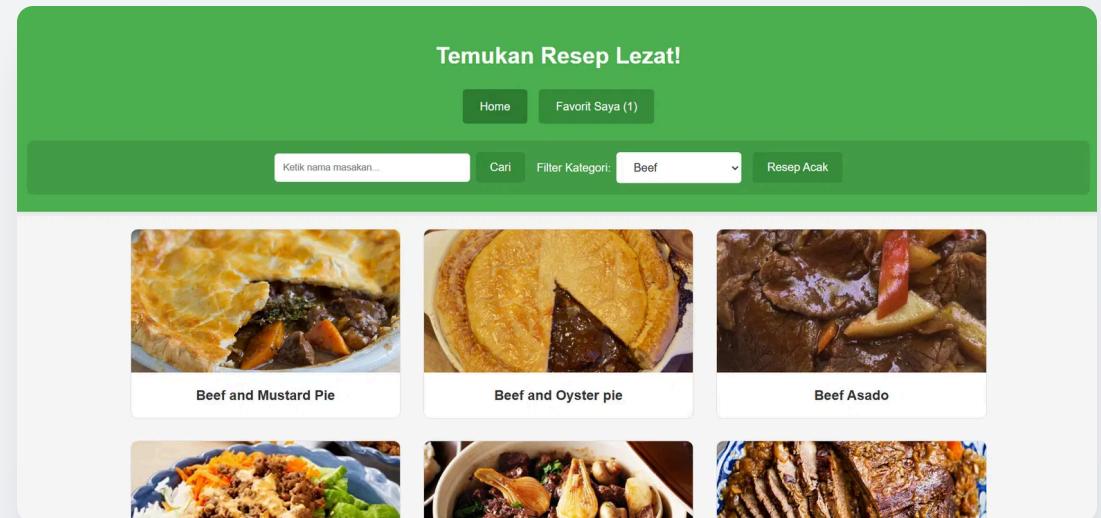


Masalah:

Pengguna sering kesulitan untuk menemukan ide dan resep masakan yang beragam dari satu sumber yang cepat dan terpercaya.

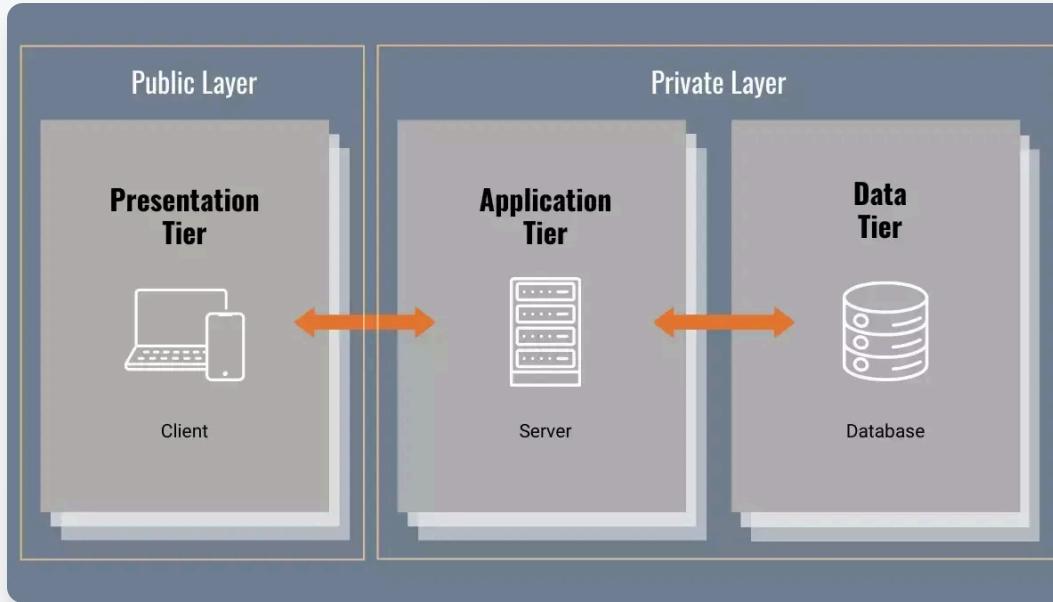
Tujuan Proyek:

- ✓ Membangun aplikasi web modern (SPA) yang interaktif dan responsif
- ✓ Mengimplementasikan arsitektur full-stack dengan pemisahan yang jelas
- ✓ Menciptakan lingkungan pengembangan terisolasi dengan Docker
- ✓ Melakukan refaktorisasi dari skrip backend sederhana ke Slim Framework



Arsitektur Sistem

Proyek ini menggunakan Arsitektur 3-Tier yang Dikontainerisasi



Tier 1: Frontend (Aplikasi Klien)

Teknologi: Vue.js

Tugas: Mengelola semua yang dilihat dan diinteraksikan oleh pengguna di browser.

Tier 2: Backend (Aplikasi Server)

Teknologi: PHP & Slim Framework

Tugas: Bertindak sebagai API Gateway atau perantara. Menerima permintaan dari frontend, memprosesnya, dan berkomunikasi secara aman dengan layanan eksternal.

Tier 3: Layanan Eksternal

Teknologi: TheMealDB API

Tugas: Menyediakan data utama resep masakan.

Teknologi yang Digunakan

Stack teknologi modern untuk aplikasi resep yang responsif dan skalabel

Frontend

Vue.js 3

 Framework utama untuk membangun antarmuka pengguna yang reaktif dan modern.

Vite

 Perkakas build yang sangat cepat untuk pengembangan frontend.

Vue Router

 Mengelola navigasi antar halaman tanpa memuat ulang halaman.

Pinia

 Mengelola state aplikasi, seperti daftar resep favorit dan hasil pencarian.

Backend



PHP 8.2

Bahasa pemrograman utama di sisi server.



Slim Framework

Micro-framework yang ringan dan cepat untuk membangun API yang terstruktur.



Apache

Web server yang berjalan di dalam kontainer untuk melayani aplikasi Slim.



Composer

Manajer dependensi untuk menginstal dan mengelola pustaka PHP.

Lingkungan



Docker

Platform untuk membuat, menyebarkan, dan menjalankan aplikasi dalam kontainer.



Docker Compose

Alat untuk mendefinisikan dan menjalankan aplikasi multi-kontainer.



Alur Pengguna (User Flow)

Perjalanan pengguna dari awal hingga akhir saat menggunakan aplikasi

1 Kunjungan Awal

Pengguna membuka aplikasi dan langsung melihat halaman utama dengan fitur pencarian.



2 Pencarian & Penemuan

- Pengguna mengetik nama masakan, ATAU
- Pengguna memilih filter berdasarkan kategori, ATAU
- Pengguna menekan tombol "Resep Acak" untuk inspirasi



3 Melihat Hasil

Aplikasi menampilkan galeri resep yang relevan.



4 Mendalami Detail

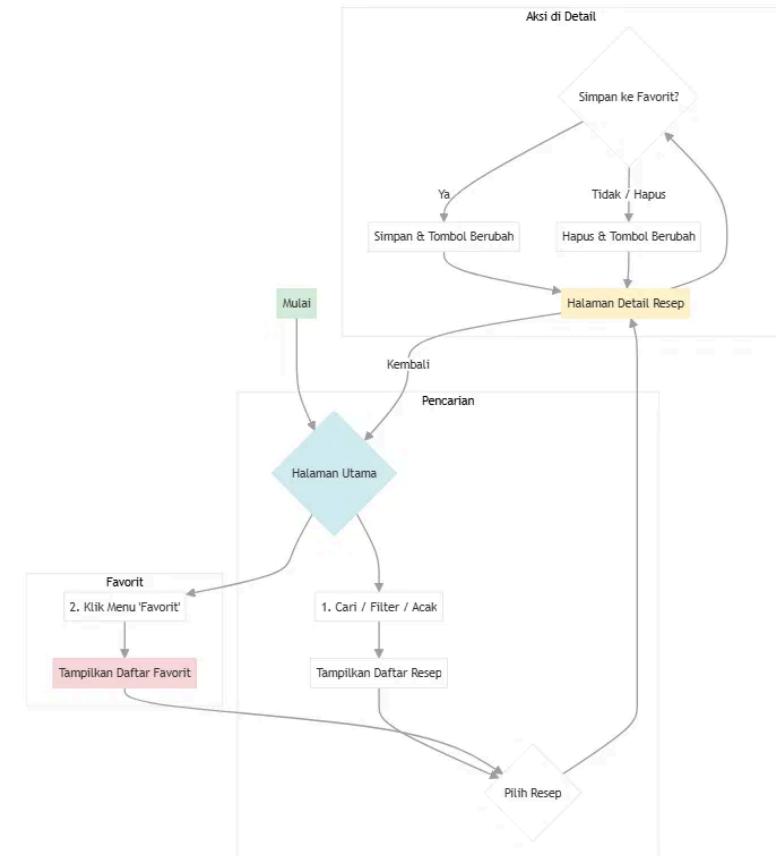
Pengguna mengklik sebuah resep dan diarahkan ke halaman detail yang menampilkan bahan, instruksi, dan video.



5 Menyimpan & Mengakses Favorit

Pengguna dapat menyimpan resep yang disukai dan mengaksesnya kembali di halaman "Favorit Saya".

Diagram Alur Pengguna (User Flow)



Untuk menyimpan, Anda bisa mengambil tangkap layar (screenshot) dari diagram ini.

Fitur-Fitur Utama

Kemampuan inti yang membuat aplikasi resep ini bermanfaat bagi pengguna



Pencarian Resep Dinamis

Mencari resep berdasarkan nama dengan hasil yang ditampilkan secara real-time saat pengguna mengetik.

"Temukan resep pasta favorit Anda dalam hitungan detik."



Filter Kategori

Mengelompokkan resep berdasarkan jenisnya (misal: "Seafood", "Dessert") untuk memudahkan penemuan.

"Ingin hidangan penutup? Cukup pilih kategori dan temukan inspirasi."



Fitur "Resep Acak"

Solusi cepat untuk pengguna yang mencari inspirasi tanpa tahu pasti apa yang mereka inginkan.

"Bingung mau masak apa? Biarkan aplikasi memilihkan untuk Anda."



Halaman Detail Komprehensif

Informasi resep yang disajikan secara jelas dan terstruktur, termasuk bahan-bahan dan langkah-langkah.

"Semua yang Anda butuhkan untuk memasak dalam satu tampilan."



Integrasi Video YouTube

Membantu pengguna dengan tutorial visual langsung dari halaman resep tanpa perlu meninggalkan aplikasi.

"Lihat cara profesional menyiapkan hidangan ini langsung di aplikasi."

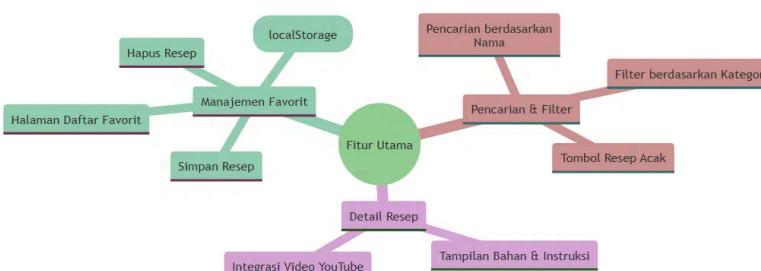


Penyimpanan Favorit Lokal

Menggunakan localStorage browser untuk menyimpan resep favorit tanpa perlu login atau membuat akun.

"Simpan resep favorit Anda untuk diakses kapan saja, bahkan offline."

Fitur-Fitur Utama Aplikasi Resep



Struktur Proyek

Struktur folder yang terorganisir adalah kunci proyek yang mudah dikelola

RecipeApp/

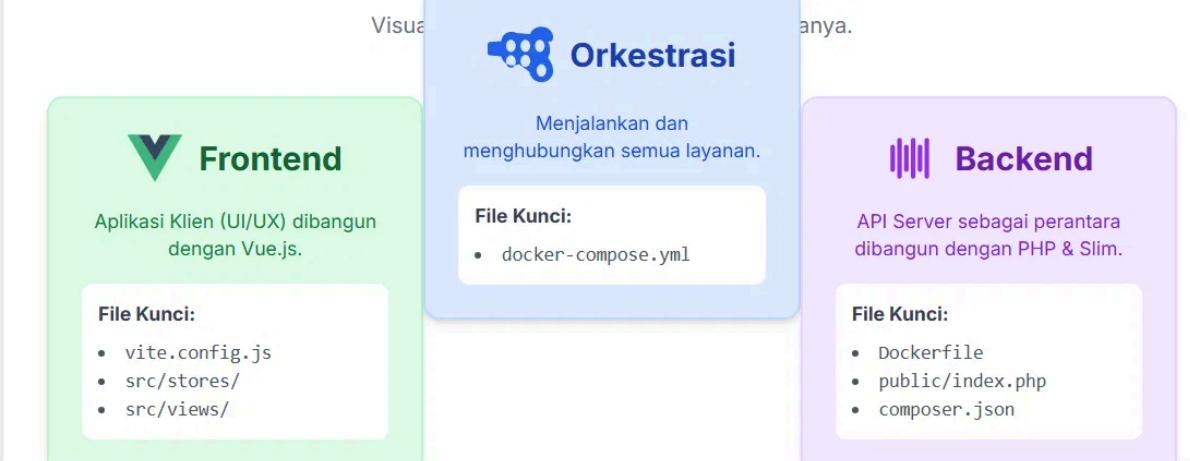
```
└── backend/
    ├── public/
    │   ├── .htaccess
    │   └── index.php
    ├── composer.json
    └── Dockerfile

    └── frontend/
        ├── src/
        │   ├── stores/
        │   ├── components/
        │   ├── views/
        │   └── assets/
        ├── vite.config.js
        └── package.json
```

File-File Kunci:

- 📄 **docker-compose.yml**: Mengatur orkestrasi kontainer, menghubungkan frontend dan backend.
- 📄 **vite.config.js**: Mengkonfigurasi proxy API untuk mengarahkan permintaan ke backend.
- 📄 **Dockerfile**: Mendefinisikan lingkungan backend dengan PHP dan Apache.
- 📄 **composer.json**: Menentukan dependensi PHP, termasuk Slim Framework.

Struktur Proyek RecipeApp



Alur Kerja Data & Kode

Contoh: Proses Pencarian Resep dari UI hingga API

1 UI (Frontend)

Pengguna mengetik "Arrabiata" dan menekan tombol "Cari" di TheHeader.vue.

2 State Management

Aksi ini memanggil fungsi handleSearch() di mealStore.js.

3 API Call (Frontend)

Fungsi handleSearch() melakukan fetch ke URL backend: /api/proxy/search.php? s=Arrabiata.

4 Routing (Backend)

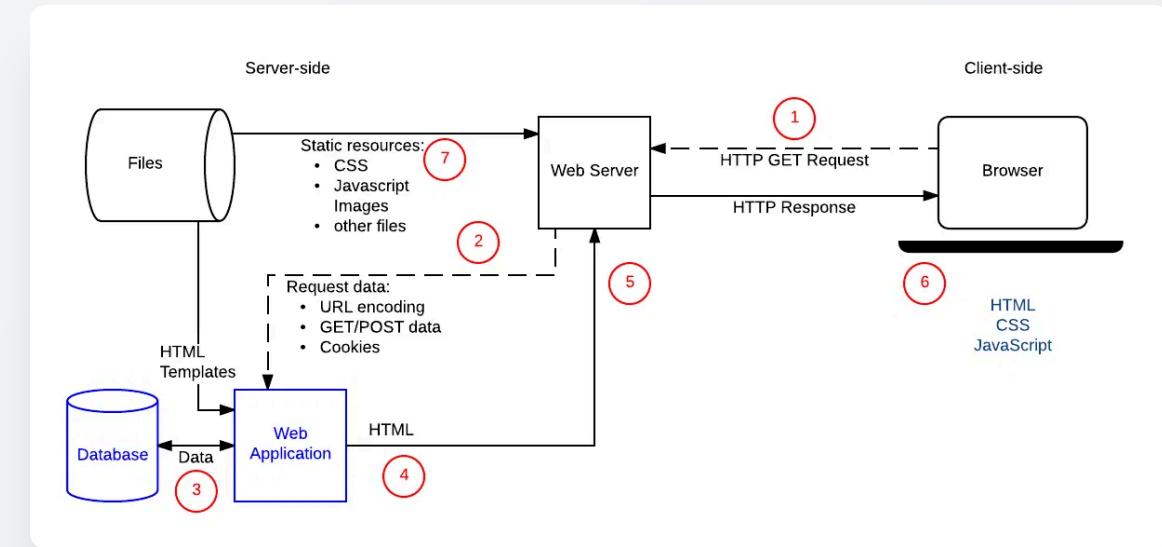
vite.config.js meneruskan permintaan /api ke kontainer backend. Router Slim mencocokkan pola URL.

5 Logic (Backend)

Skrip PHP menerima "search.php" sebagai endpoint dan melakukan request cURL ke TheMealDB API.

6 Response & Update

Backend meneruskan respons JSON ke frontend. mealStore.js memperbarui state dan UI menampilkan resep.



Alur Data



Frontend



Backend



API Eksternal

Kesimpulan & Pengembangan Selanjutnya

Refleksi pencapaian dan visi masa depan untuk aplikasi resep

Kesimpulan

- ✓ Berhasil membangun aplikasi full-stack yang fungsional dengan teknologi modern (Vue.js, PHP Slim, Docker).
- ✓ Berhasil mengimplementasikan lingkungan pengembangan berbasis Docker yang terisolasi dan portabel.
- ✓ Berhasil melakukan refaktorisasi backend ke Slim Framework, menciptakan fondasi yang lebih terstruktur dan skalabel.
- ✓ Memahami dan menyelesaikan berbagai tantangan teknis, mulai dari konfigurasi server, manajemen dependensi, hingga debugging lintas lapisan.
- ✓ Menciptakan antarmuka pengguna yang intuitif dan responsif dengan fitur pencarian dinamis dan penyimpanan favorit lokal.

Pengembangan Selanjutnya

- 💡 **Database & Autentikasi**
Mengganti localStorage dengan database (misal: MySQL) dan menambahkan sistem login agar pengguna dapat menyimpan favoritnya secara permanen.
- 💡 **Fitur "Upload Resep"**
Memungkinkan pengguna untuk berkontribusi resep mereka sendiri, memperkaya konten aplikasi.
- 💡 **PWA (Progressive Web App)**
Membuat aplikasi dapat diinstal di perangkat pengguna dan berfungsi secara offline untuk pengalaman yang lebih baik.
- 💡 **Testing & Deployment**
Menulis unit test dan end-to-end test untuk memastikan kualitas kode, serta menyebarkan aplikasi ke layanan cloud agar dapat diakses oleh publik.

Terima Kasih!

Proyek ini memiliki fondasi yang kuat untuk dikembangkan lebih lanjut