

Dokumen Teknis

Website Cafe Friendzone

Menggunakan Arsitektur Microservice

Tugas Proyek Akhir Semester

Mata Kuliah: Pengembangan Aplikasi Terdistribusi

Dipersiapkan oleh:

11322013	Felix Aldi Simanjuntak
11322028	Christian Jhon Panjaitan
11322042	Kesia Rotua Sihombing
11322058	Dian Anggi Belita Sitanggang

Untuk:

Institut Teknologi Del
2022



PROYEK PENGEMBANGAN APLIKASI TERDISTRIBUSI
INSTITUT TEKNOLOGI DEL 2022

1 Pendahuluan

Cafe Friendzone, sebuah oase kuliner yang menghadirkan hidangan lezat dan suasana nyaman, kini dihadapkan pada tantangan sistem pemesanan manual yang tidak lagi efisien. Antrian panjang dan waktu tunggu yang lama, terutama di jam-jam sibuk, menjadi hambatan bagi pelanggan dan staf kafe. Metode pemesanan melalui WhatsApp pun tak luput dari kendala. Chat yang menumpuk dan telepon yang sibuk seringkali menyulitkan pelanggan dalam memesan, dan staf kafe pun kesulitan menangani pemesanan dengan cepat. Menyadari hal ini, dapat disimpulkan, bahwa sistem Cafe Friendzone saat ini dihadapkan pada beberapa tantangan, seperti sistem pemesanan manual yang tidak efisien dan kurangnya informasi bagi pelanggan. Untuk mengatasinya, implementasi sistem pemesanan terintegrasi berbasis website dengan arsitektur yang tepat menjadi penting. Dua pilihan arsitektur utama yang perlu dipertimbangkan adalah Monolith dan Microservices. Masing-masing memiliki kelebihan dan kekurangan yang perlu dipahami dengan seksama.

1.1 Perbedaan arsitektur Monolith dan Microservices

Arsitektur Monolith adalah pendekatan tradisional dalam pengembangan perangkat lunak di mana seluruh aplikasi dibangun sebagai satu kesatuan tunggal. Dalam konteks Cafe Friendzone, ini berarti bahwa semua fitur, seperti antarmuka pengguna, basis data, server, dan model pull & push, diimplementasikan dalam satu prosesor.

Sementara itu, arsitektur Microservices memecah aplikasi menjadi serangkaian layanan yang independen dan terpisah. Setiap layanan memiliki prosesor yang independen, yang mengelola fungsi spesifik dari aplikasi. Misalnya, ada layanan untuk antarmuka pengguna, layanan basis data, layanan server, dan lain-lain.

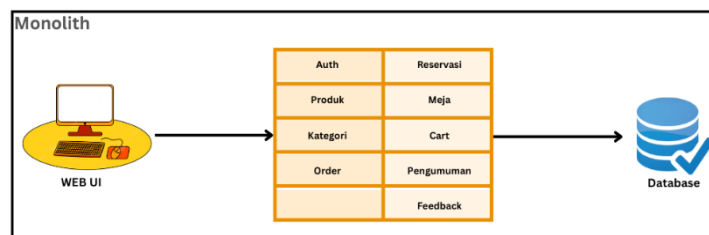
Perbedaan utama antara kedua arsitektur ini adalah pada cara prosesor dikelola. Dalam Monolith, semua logika aplikasi ditangani oleh satu prosesor, sementara dalam Microservices, setiap layanan memiliki prosesor independen. Ini memiliki implikasi yang signifikan dalam hal skalabilitas, fleksibilitas, dan pengelolaan sistem. Arsitektur Monolith memiliki kelebihan dalam hal kesederhanaan dan kemudahan pengembangan awal. Namun, kelemahan utamanya adalah skalabilitas yang terbatas dan kesulitan dalam memperbarui bagian-bagian tertentu tanpa memengaruhi keseluruhan sistem.

Di sisi lain, arsitektur Microservices menawarkan skalabilitas yang lebih tinggi dan memungkinkan pengembangan dan pembaruan yang lebih cepat. Namun, membutuhkan manajemen yang lebih kompleks untuk koordinasi antara layanan, dan infrastruktur yang lebih rumit untuk mengelola komunikasi antara layanan.

Dalam konteks Cafe Friendzone, penggunaan arsitektur Microservices dipilih untuk meningkatkan efisiensi dan fleksibilitas dalam pengelolaan sistem informasi. Dengan memecah aplikasi menjadi layanan-layanan yang independen, Cafe Friendzone dapat mengatasi masalah seperti pemesanan meja dan makanan yang dilakukan secara manual, serta meningkatkan kemampuan dalam memantau meja yang sudah dipesan oleh pelanggan. Selain itu, dengan arsitektur Microservices, Cafe Friendzone dapat lebih responsif terhadap kebutuhan pelanggan dan mengurangi waktu yang diperlukan untuk menangani pemesanan.

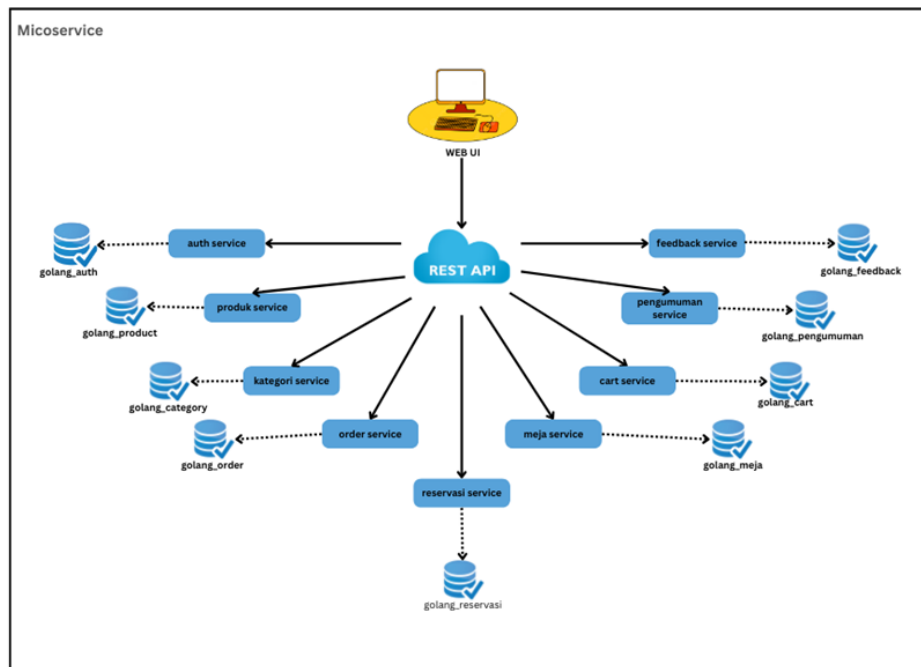
1.2 Diagram

Program ini dirancang dengan memanfaatkan dua bahasa pemrograman yang berbeda. Bagian back-end dan front-end dibangun dalam bahasa yang berbeda-beda. Bahasa pemrograman Go digunakan pengerjaan back-end, sementara untuk pengerjaan bagian front-end fokus menggunakan bahasa PHP, dengan penekanan pada kerangka kerja Laravel.



Gambar 1 Diagram monolith

Dalam konteks website Cafe Friendzone, arsitektur Monolith akan menyatukan semua fitur seperti autentikasi pengguna, manajemen keranjang, kategori produk, sistem feedback, manajemen meja, fitur pemesanan, pengumuman, manajemen produk, dan fitur booking meja ke dalam satu kesatuan. Artinya, semua logika aplikasi, basis data, dan server akan tergabung menjadi satu entitas yang besar.



Gambar 2 Diagram Microservice

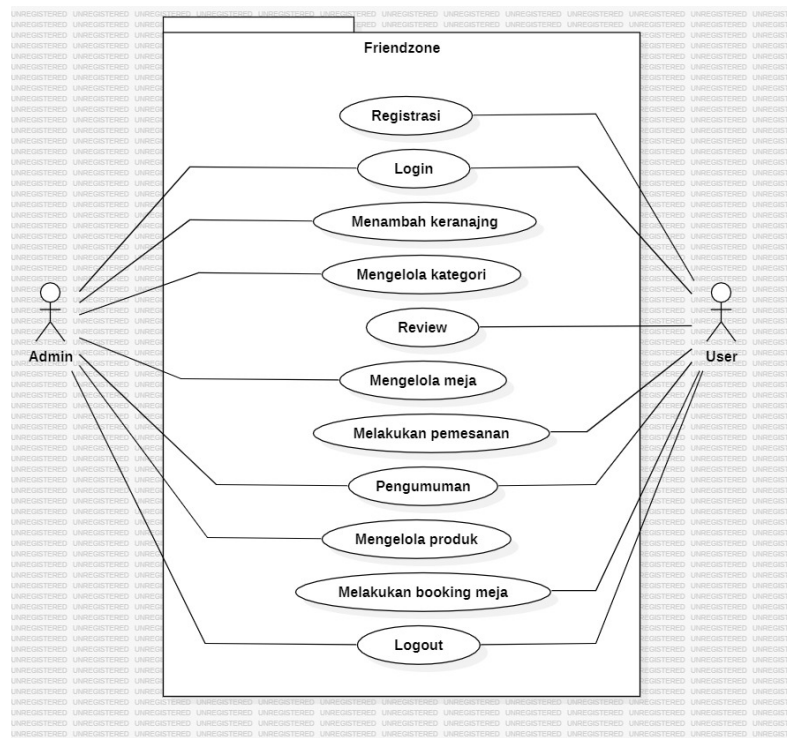
Sedangkan untuk “microservice”, arsitektur ini dirancang dalam sebuah sistem untuk memecah layanan (service) menjadi unit-unit yang lebih kecil. Dengan pendekatan ini, setiap unit akan memiliki database tersendiri. Pada bagian *back-end*, setiap *service* memiliki *database*-nya masing-masing. Ada beberapa *service/database* yang terdapat dalam website ini yakni autentikasi, keranjang, kategori, feedback, crud meja, pemesanan, pengumuman, produk, dan booking meja. Arsitektur microservice yang digunakan pada pembangunan *website* Café Friendzone. Setiap unit *service* akan beroperasi secara mandiri, memastikan bahwa satu pun unit tidak akan mengganggu unit lainnya. Meskipun demikian, meski berdiri sendiri, setiap unit tetap berinteraksi satu sama lain. Komunikasi antar unit dilakukan melalui HTTP Request yang mengikuti standar REST API. Setiap *database* pada masing-masing *service* memiliki *port* yang sama. Namun, *port* yang digunakan pada setiap *service* tersebut akan berbeda. Hal ini ditujukan sebagai pemisah antarservice sehingga apabila satu *service* tidak dapat dijalankan, maka *service* yang lainnya masih tetap dapat berjalan.

2 Desain Rancangan Aplikasi

Pada bab 2 berisi tentang desain rancangan aplikasi, yaitu *use case diagram*.

2.1 Use Case Diagram

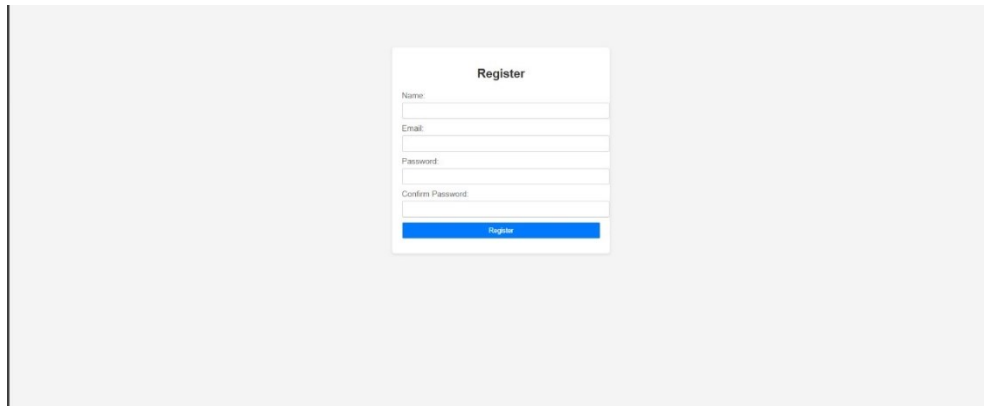
Website ini akan dapat memiliki dua role yakni oleh staff (admin) dan customer (user). Admin akan bertanggung jawab dalam mengelola system dan customer sebagai user akan menggunakan web berdasarkan kebutuhan untuk melakukan pemesanan dan pem-bookingan meja pada Café Friendzone. User harus melakukan registrasi dahulu agar memiliki akun serta dapat mengakses website Café Friendzone.



Gambar 3 Use Case Diagram Website Café Friendzone

3 Tampilan Aplikasi

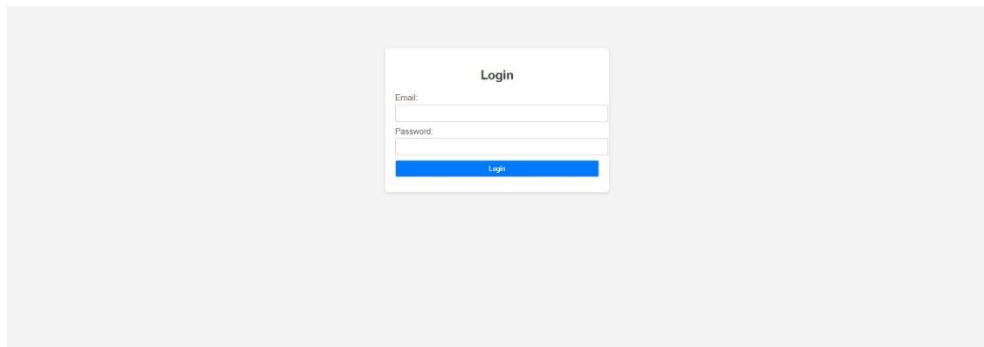
1. Tampilan Halaman Registrasi



The screenshot shows a 'Register' form centered on a light gray background. The form has a title 'Register' at the top. Below the title are four input fields: 'Name:', 'Email:', 'Password:', and 'Confirm Password:'. At the bottom of the form is a blue button labeled 'Register'.

Gambar 4 Tampilan Halaman Registrasi

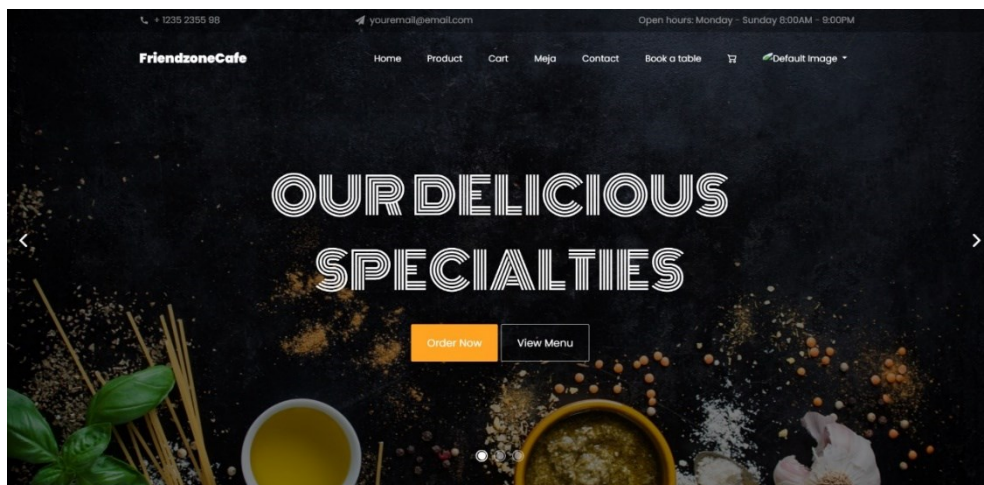
2. Tampilan Halaman Login



The screenshot shows a 'Login' form centered on a light gray background. The form has a title 'Login' at the top. Below the title are two input fields: 'Email:' and 'Password:'. At the bottom of the form is a blue button labeled 'Login'.

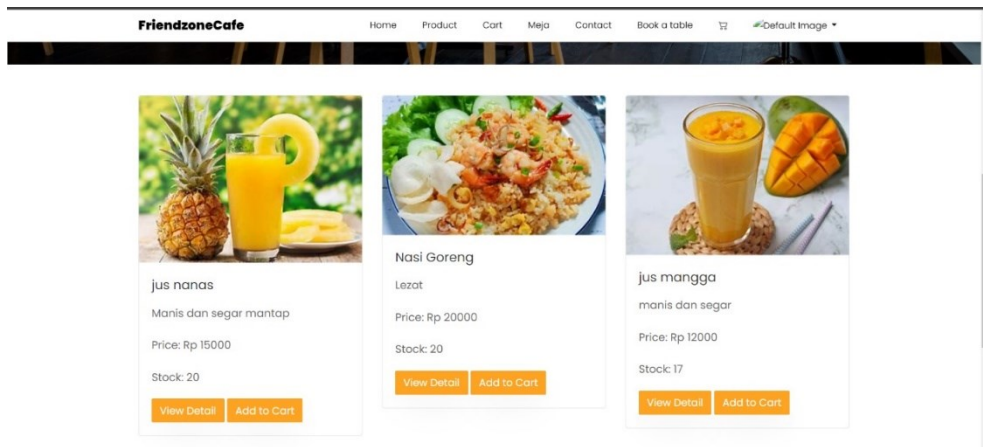
Gambar 5 Tampilan Halaman Login

3. Dashboard User



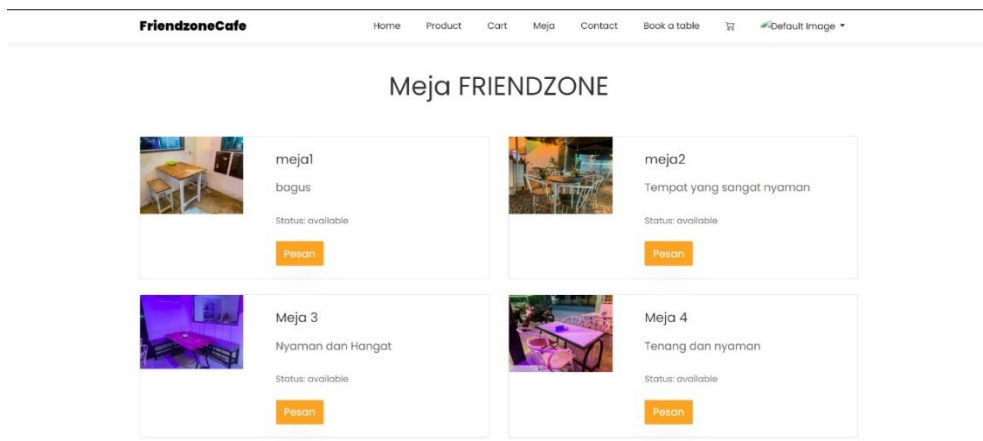
Gambar 6 Dashboard User

4. Tampilan Halaman Produk user



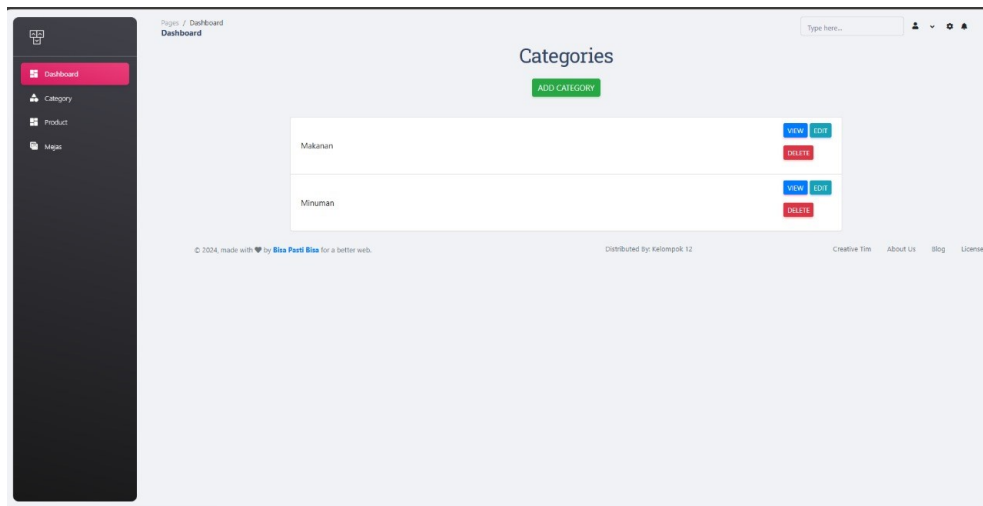
Gambar 7 Tampilan Halaman Produk user

5. Tampilan Halaman dashboard user



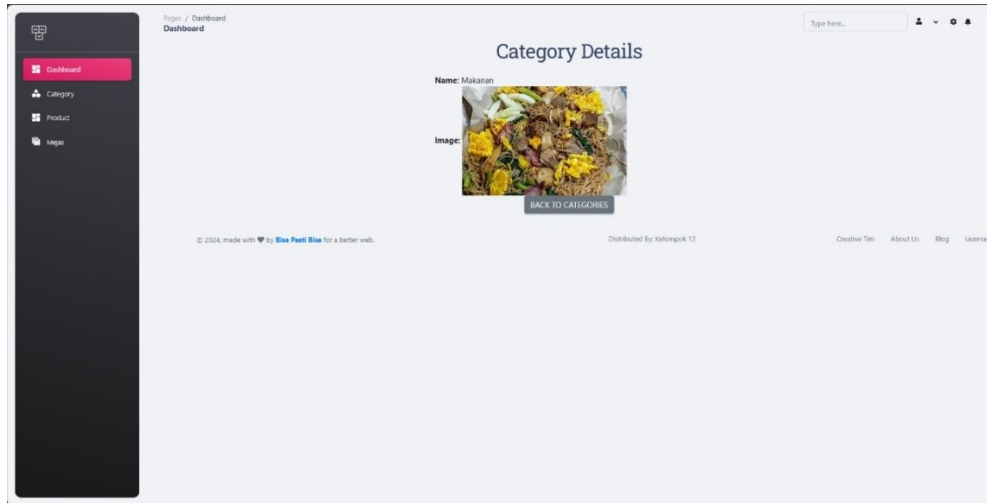
Gambar 8 Tampilan Halaman dashboard user

6. Tampilan Halaman Categories



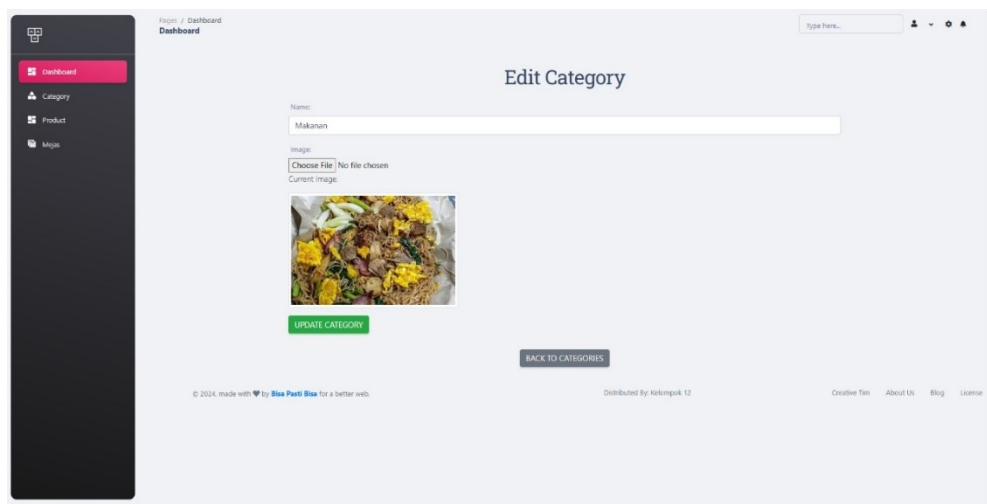
Gambar 9 Tampilan Halaman Categories

7. Tampilan Halaman Detail Categories



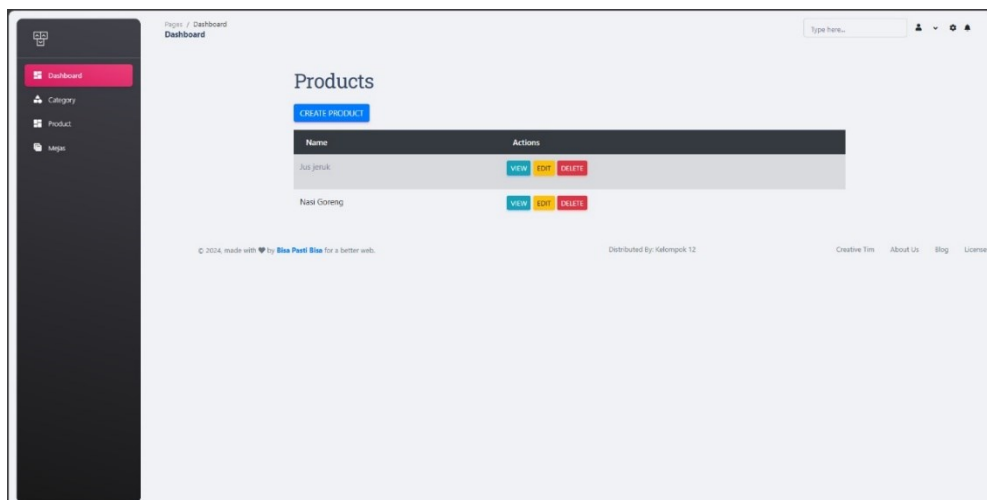
Gambar 10 Tampilan Halaman Detail Categories

8. Halaman Edit Categories



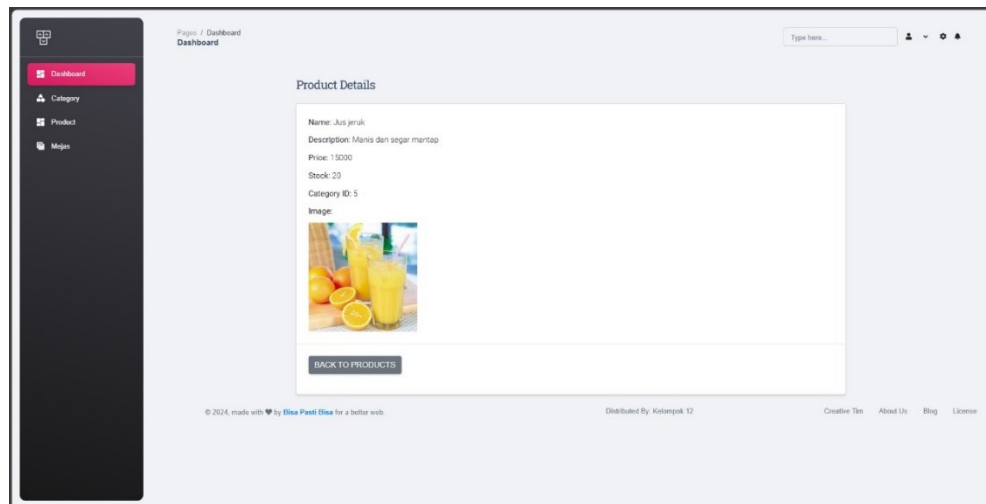
Gambar 11 Halaman Edit Categories

9. Halaman Produk



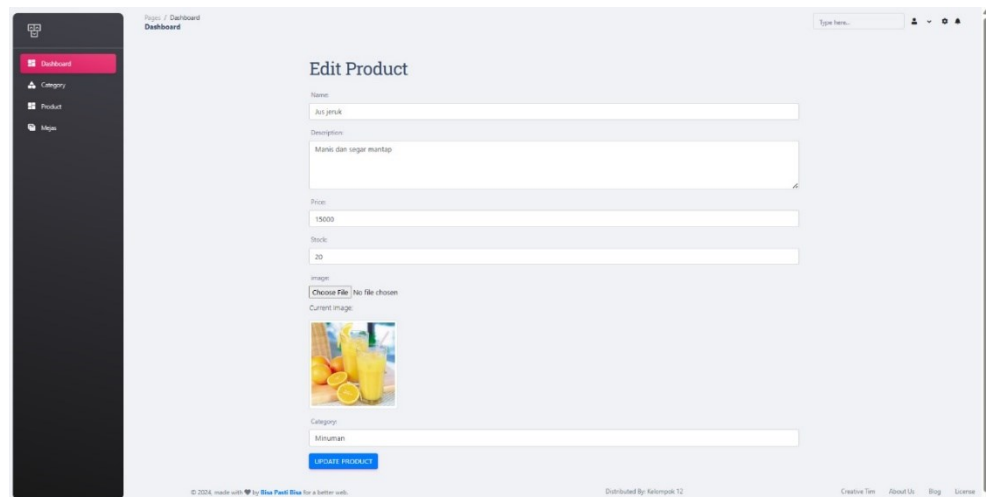
Gambar 12 Halaman Produk

10. Halaman Detail Produk



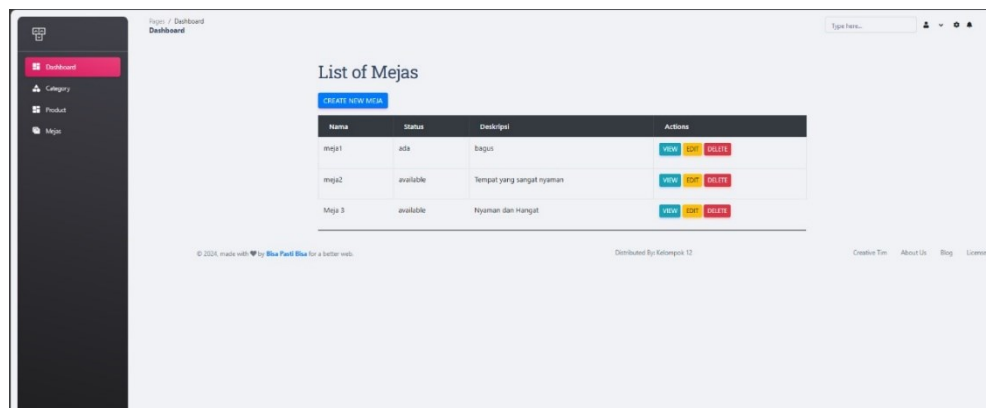
Gambar 13 Halaman Detail Produk

11. Halaman Edit Produk



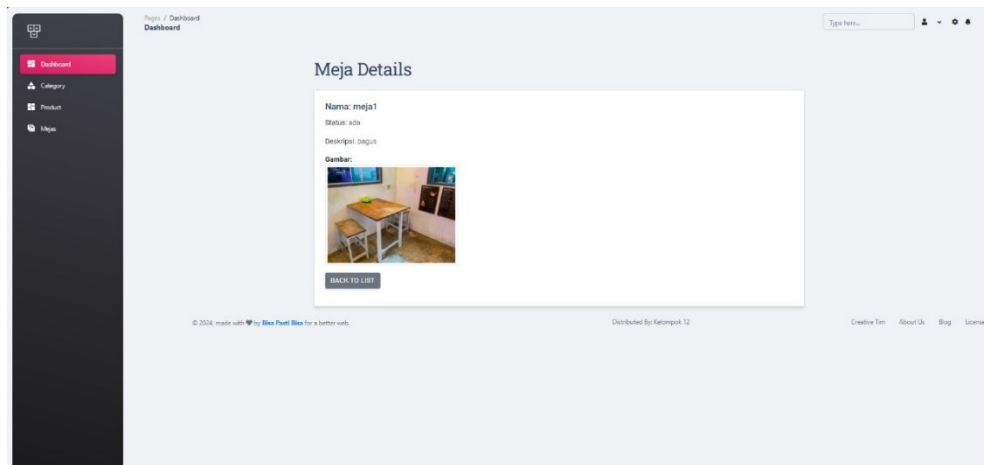
Gambar 14 Halaman Edit Produk

12. Halaman Meja



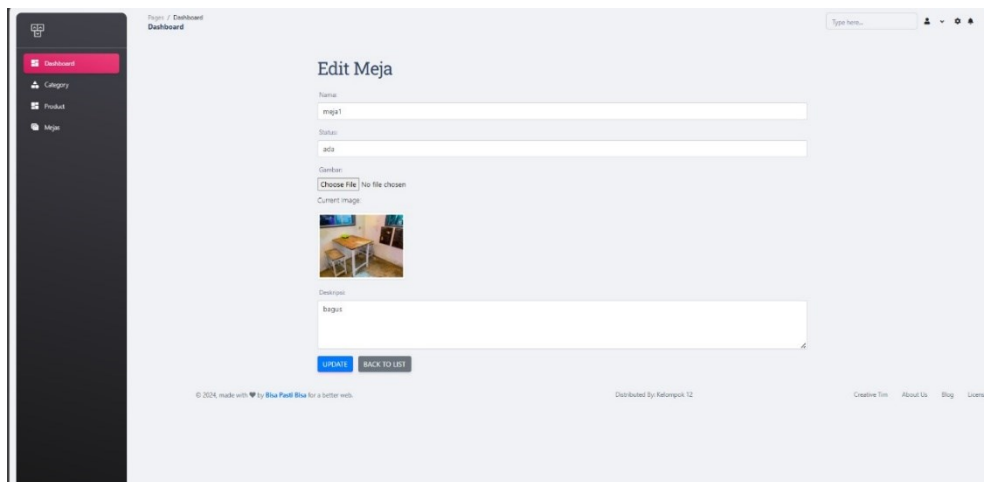
Gambar 15 Halaman Meja

13. Halaman Detail Meja



Gambar 16 Halaman Detail Meja

14. Halaman Edit Meja



Gambar 17 Halaman Edit Meja

4 Pengujian Aplikasi

Pada bab ini berisi tentang tampilan website yang telah dibangun.

Table 1 Pengujian service category down

Nama Butir Uji	Pengujian <i>service category</i> down		
Tujuan	Untuk mengetahui apakah <i>service</i> lain berjalan saat <i>service category</i> down		
Tanggal Pengujian	15/05/2023		
Penguji	Kelompok 12		
Skenario Pengujian			
<ul style="list-style-type: none">- Menonaktifkan <i>service category</i>- Menjalankan website			
Kriteria Evaluasi Hasil			
Berhasil menjalankan <i>service</i> produk, meskipun <i>service category</i> sedang down			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
<i>Service category</i> dinonaktifkan	Service produk berjalan	Sesuai yang diharapkan	[X] diterima [] ditolak
Catatan			

←

→

🔄

📄

🔍

🔗

🌟

🔒

🔧

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

🔌

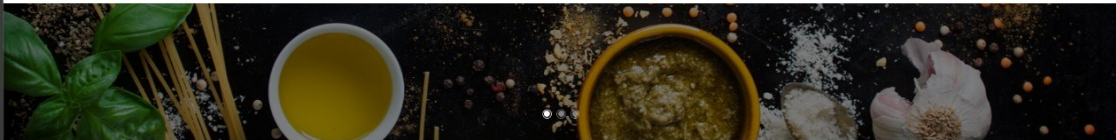


🔌

🔌

🔌

🔌

Table 2 Pengujian service authentication down

Nama Butir Uji	Pengujian <i>service authentication down</i>		
Tujuan	Untuk mengetahui apakah <i>service</i> lain berjalan saat <i>service authentication down</i>		
Tanggal Pengujian	15/05/2023		
Penguji	Kelompok 12		
Skenario Pengujian			
<ul style="list-style-type: none">- Menonaktifkan <i>service authentication</i>- Menjalankan aplikasi			
Kriteria Evaluasi Hasil			
Berhasil menjalankan <i>service</i> produk, meskipun <i>service authentication</i> sedang <i>down</i>			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
<i>Service authentication</i> dinonaktifkan	Service produk berjalan	Sesuai yang diharapkan	[X] diterima [] ditolak
Catatan			
<div><div><div>FriendzoneCafe</div><div>HomeProductCartMenuContactBook a table🔍#Default Image +</div><div></div><div><div><div><div><div>jus nenas</div><div>Manis dan segar mantap.</div><div>Price: Rp 15000</div><div>Stock: 20</div><div><div>view detail</div><div>Add to Cart</div></div></div></div><div><div><div><div>Nasi Goreng</div><div>Lazat</div><div>Price: Rp 20000</div><div>Stock: 20</div><div><div>view detail</div><div>Add to Cart</div></div></div></div><div><div><div><div>jus mangga</div><div>manis dan segar</div><div>Price: Rp 12000</div><div>Stock: 17</div><div><div>view detail</div><div>Add to Cart</div></div></div></div></div></div></div></div></div></div>			

KESIMPULAN: Ketika *service* lain sedang *down*, hal ini tidak akan memengaruhi *service* lainnya, hal itu disebabkan karena menggunakan *microservice*, setiap *service* pada aplikasi yang kami bangun dapat berjalan secara independent.