**Dokumen Teknis**

**Rancang Bangun Website Commanditaire Vennootschap (C.V.) Pausoan Material**

**Tugas Proyek Akhir Semester**

**Mata Kuliah: Pengembangan Aplikasi Terdistribusi**

**Dipersiapkan oleh:**

|  |  |
| --- | --- |
| NIM 11322002 | Erlangga Abel Napitupulu |
| NIM 11322018 | Maranatha Siahaan |
| NIM 11322033 | Eka Lumbanraja |
| NIM 11322047 | Olivia Apriani |

**Untuk: Institut Teknologi Del**

**2024**



**PROYEK PENGEMBANGAN APLIKASI TERDISTRIBUSI INSTITUT TEKNOLOGI DEL 2024**

*Nomor dokumen: DT-PASTI-02 Tanggal: 2024-05-15 Jumlah Halaman: 14*

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc166708998)

[1 Pendahuluan 3](#_Toc166708999)

[1.1 Deskripsi Umum Aplikasi 3](#_Toc166709000)

[1.2 Karakteristik Pengguna Aplikasi 5](#_Toc166709001)

[1.3 Fungsi pada Aplikasi 5](#_Toc166709002)

[2 Tampilan Aplikasi 7](#_Toc166709003)

[3 Pengujian Aplikasi 9](#_Toc166709004)

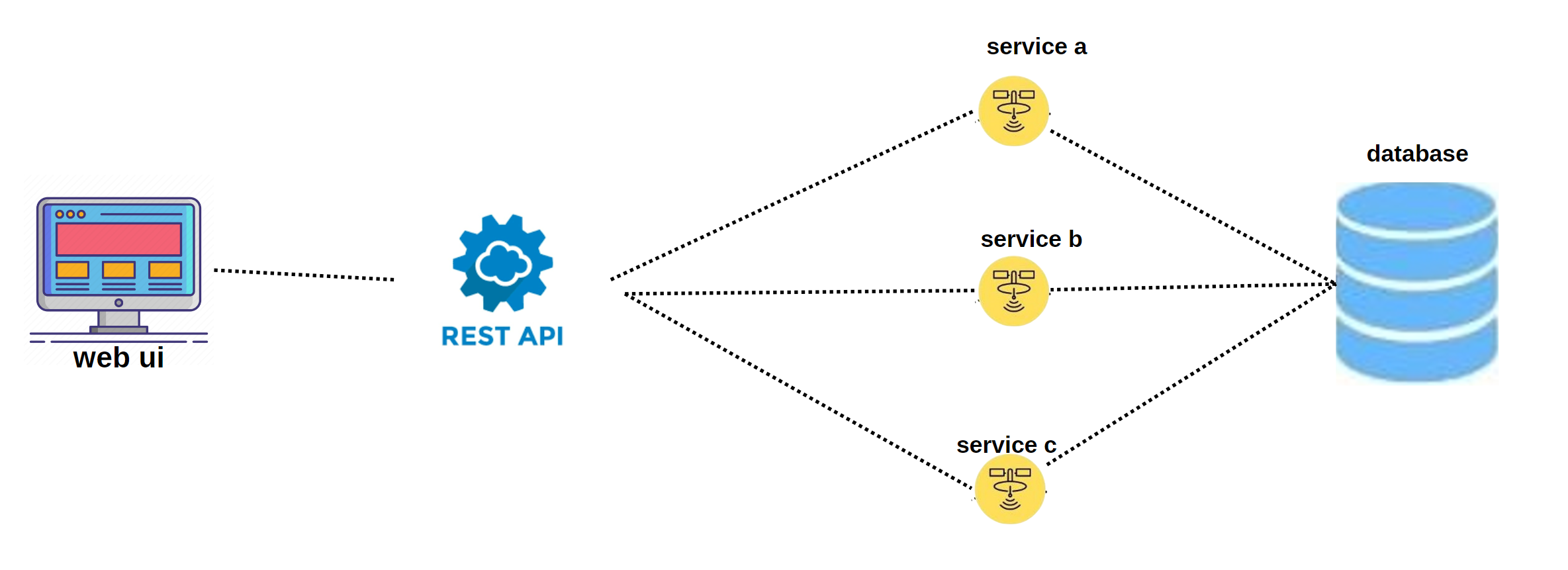
# Pendahuluan

Pada bab 1 berisi tentang deskripsi umum aplikasi, karakteristik pengguna aplikasi, dan fungsi pada aplikasi.

# Deskripsi Umum Aplikasi

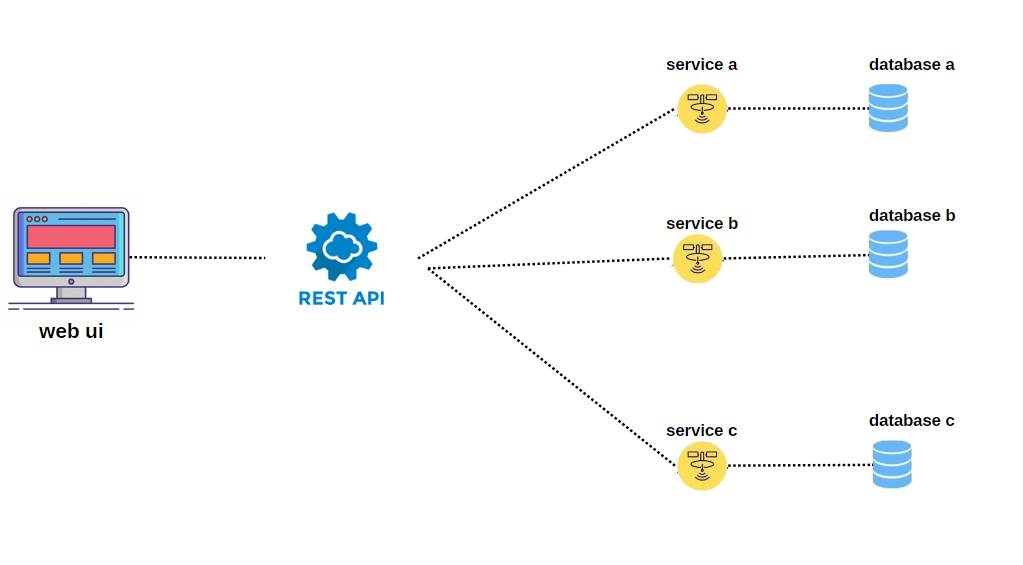
Website ini dibangun dengan menggunakan 2 bahasa pemrograman. Bagian *back-end* dan *front-end* dibangun dengan bahasa yang berbeda. Untuk bagian *front-end* digunakan bahasa PHP, khususnya *framework Laravel* dan untuk bagian *back-end* digunakan bahasa *Go.*

Perbedaan arsitektur *monolith* dan *microservice* yaitu dalam arsitektur *monolith*, seluruh logika aplikasi diimplementasikan dalam satu prosesor atau aplikasi tunggal. Semua komponen seperti antarmuka pengguna, basis data, dan logika server berada dalam satu kesatuan.

****

**Gambar 1. Arsitektur Monolith**

Sedangkan website C.V. Pausoan Material menggunakan arsitektur *microservice.* Sesuai dengan namanya, arsitektur ini dirancang pada sebuah sistem untuk membagi *service* menjadi *service* yang lebih kecil. Dengan ini, maka setiap *service* akan memiliki *database*nya masing-masing. *Service* yang ada pada website seperti services\_auth, services\_gallery, services\_order, services\_review, services\_aboutus, services\_employee, services\_slider, services\_category, services\_cart*. Service* akan berjalan secara *independent* sehingga satu *service* tidak akan memengaruhi *service* lainnya. Namun, meskipun berdiri sendiri, setiap *service* ini akan tetap saling terhubung antara satu dengan yang lainnya. Antar *service* akan saling berkomunikasi dengan HTTP *Request* berstandar 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 antar *service* sehingga saat satu *service* sedang tidak dapat dijalankan, maka *service* yang lainnya masih tetap dapat berjalan.



**Gambar 2. Arsitektur Microservice**

Pada Gambar 2 terlampir arsitektur microservice yang digunakan pada pembangunan aplikasi Website C.V. Pausoan Material. REST API digunakan sebagai penghubung antara *web ui* dengan *service* yang tersedia sehingga memungkinkan *service* dapat dikonsumsi oleh *web ui*. Hal inilah yang menyebabkan *service* dapat diakses oleh beberapa *platform* tanpa harus terbatas pada satu bahasa pemrograman saja. Sama halnya dengan pembangunan Website C.V. Pausoan Material ini. . Bagian *back-end* dan *front-end* dibangun dengan bahasa yang berbeda. Untuk bagian *front-end* digunakan bahasa PHP, khususnya *framework Laravel* dan untuk bagian *back-end* digunakan bahasa *Go.* Meskipun memiliki bahasa yang berbeda, tetapi aplikasi akan tetap dapat berjalan dengan adanya REST API.

Pada bagian *back-end,* setiap *service* memiliki *database*-nya masing-masing. Pada web ini terdapat tujuh *service* dan tujuh *database. Service* yang ada pada website seperti *services\_category, services\_product, services\_cart, services\_comment, services\_profile, services\_ulasan, dan services\_aboutus.* Setiap *service* dirancang memiliki *port* yang berbeda sehingga *port* tidak akan bertabrakan.

Komponen dalam Arsitektur Microservices:

* User Interface: Antarmuka pengguna yang terpisah untuk setiap layanan, memungkinkan pembaruan dan perbaikan independen.
* Database: Setiap microservice memiliki basis data sendiri, memastikan isolasi data dan kinerja optimal.
* Server: Layanan server independen yang menangani logika aplikasi untuk setiap fungsi spesifik.
* Pull & Push Model: Menggunakan model pull untuk mengumpulkan data dari berbagai layanan dan model push untuk mengirim pembaruan atau notifikasi ke pengguna.

# Karakteristik Pengguna Aplikasi

Pada Website C.V. Pausoan Material, terdapat tiga kategori pengguna. Kategori pengguna pada aplikasi adalah pemilik toko, pelanggan dan *guest.* Karakteristik pengguna aplikasi dilampirkan pada Tabel 1.

**Tabel 1. Karakteristik Pengguna Aplikasi**

| **User Group/Role** | **Fungsi** | **Hak Akses** |
| --- | --- | --- |
| Pemilik Toko | 1. Mengelola About Us 2. Mengelola Kategori 3. Mengelola Produk 4. Mengelola slide show 5. Mengelola Karyawan 6. Autentikasi 7. Tambah Keranjang 8. Mengelola Galeri 9. Memesan Produk 10. Request Produk Jasa | Akses ke menu *Home* untuk memperbaharui data |
| *Guest* | 1. Melihat Produk | Akses ke menu *Home* untuk melihat daftar produk dan ulasan produk |
| Pelanggan | 1. Memberi ulasan produk | Akses ke menu *Home* untuk memesan produk |

# Fungsi pada Aplikasi

Berikut adalah beberapa fungsi yang ada dalam Proyek ini, antara lain:

1. Mengelola About Us

Fungsi ini digunakan oleh pemilik toko untuk menambahkan tentang identitas toko.

1. Mengelola Kategori

Fungsi ini digunakan oleh pemilik toko untuk menambah dan menghapus kategori untuk produk yang ada.

1. Mengelola Produk

Pemilik toko menggunakan fungsi ini untuk dapat mengelola daftar produk, termasuk menambah, mengedit, atau menghapus produk dari katalog mereka.

1. Memberi Ulasan Toko

Fungsi ini digunakan oleh pelanggan untuk dapat memberikan kritik dan saran terkait toko memberikan umpan balik yang berguna untuk pemilik toko.

1. Mengelola Slide show

Fungsi ini digunakan oleh pemilik toko untuk menambah, mengganti dan menghapus slide show.

1. Mengelola Karyawan

Fungsi ini digunakan oleh pemilik toko untuk menambah, mengganti dan menghapus daftar karyawan yang bekerja di toko.

1. Autentikasi

Fungsi ini digunakan oleh pemilik toko dan pelanggan untuk registrasi, login dan logout.

1. Tambah Keranjang

Fungsi ini digunakan oleh pelanggan untuk menambahkan produk ke keranjang.

1. Mengelola Galeri

Fungsi ini digunakan oleh pemilik toko untuk menambah, mengedit dan menghapus foto-foto yang ada di website mereka.

1. Memesan Produk

Fungsi ini digunakan oleh pelanggan untuk memesan produk yang ada di Website C.V. Pausoan Material.

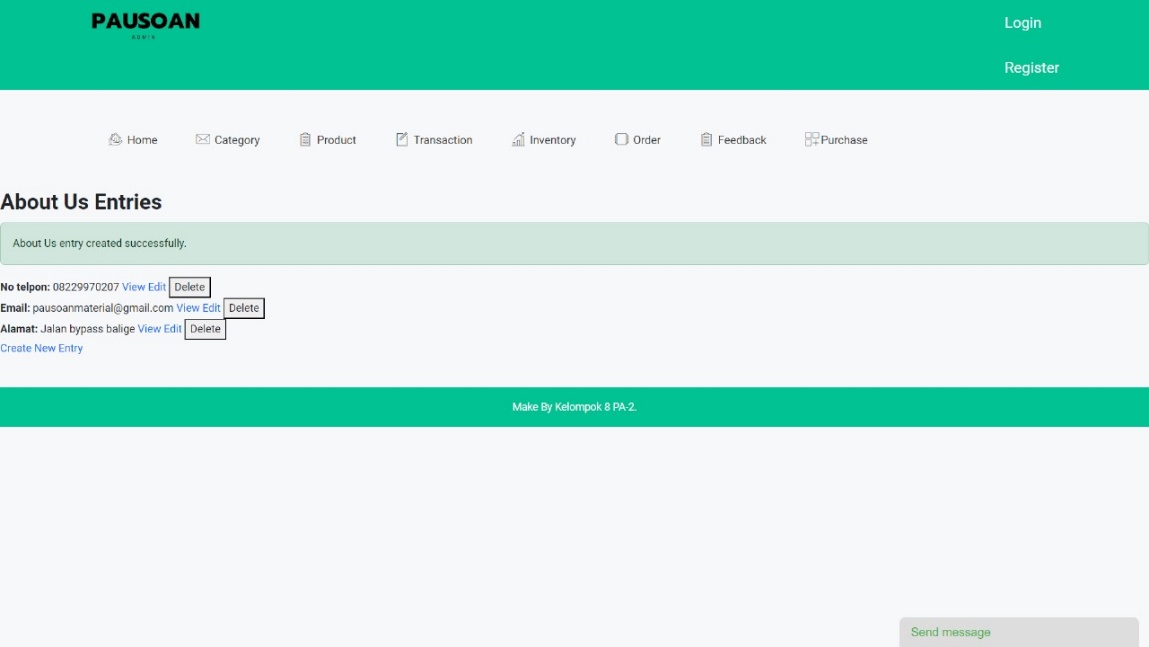
1. Request Produk Jasa

Fungsi ini digunakan oleh pelanggan untuk merequest produk atau jasa yang mereka inginkan agar toko menjualnya.

# Tampilan Aplikasi

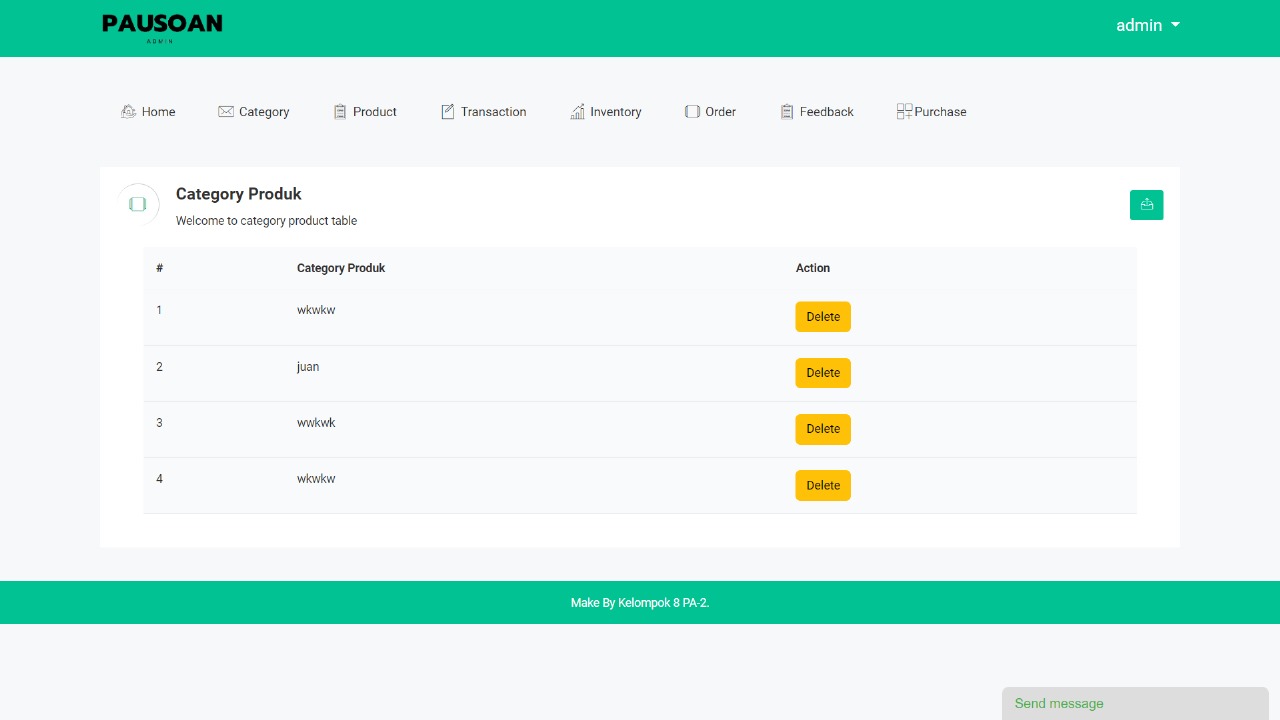
Pada bab 2 berisi tentang tampilan aplikasi yang telah dibangun.

1. Halaman About Us

****

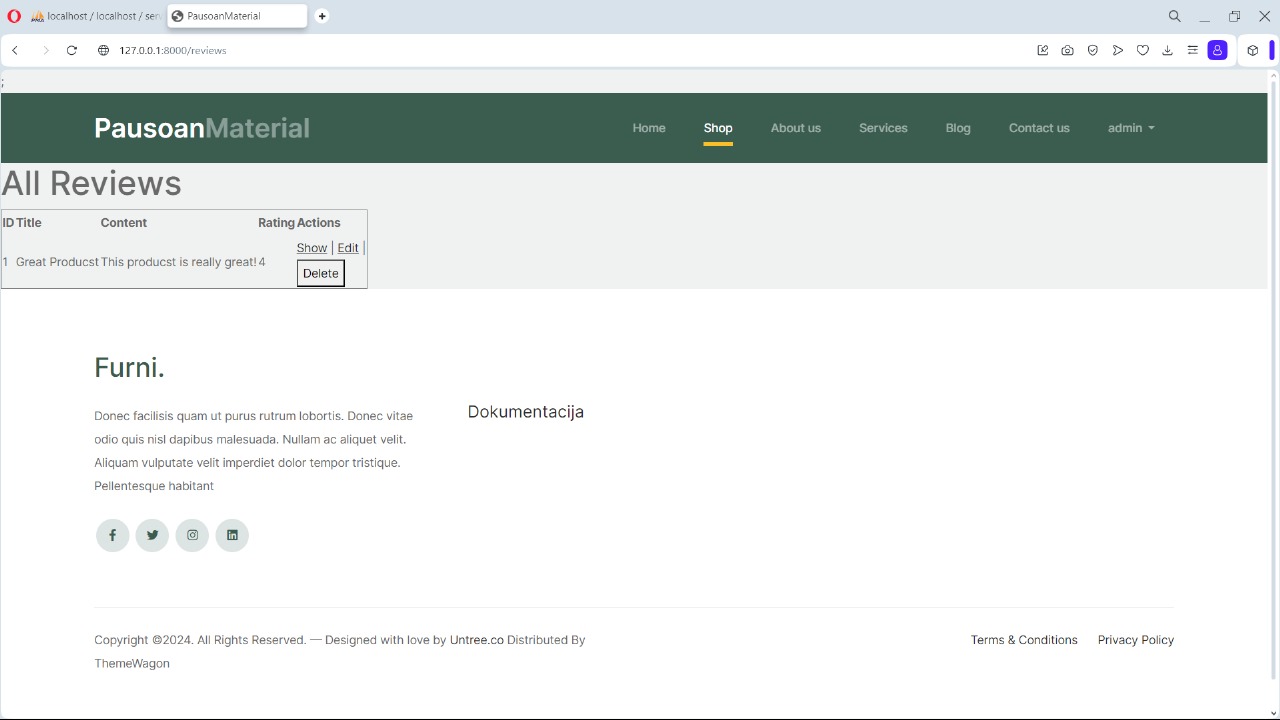
**Gambar 3. Halaman About Us**

1. Halaman Kategori

****

**Gambar 4. Halaman Kategori**

1. Halaman Produk
2. Halasan Ulasan Toko

****

**Gambar 5. Halaman Ulasan Toko**

1. Halaman Slide show.
2. Mengelola Karyawan
3. Halaman Request Produk Jasa

# Pengujian Aplikasi

Pada pengujian ini, kami berfokus pada microservice yang sedang down.

1. Pengujian About Us

**Tabel 2. Pengujian About us**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Butir Uji** | Pengujian *about us down* | | |
| **Tujuan** | Untuk mengetahui apakah *service* lain berjalan saat *service about us* down | | |
| **Tanggal Pengujian** | 15/05/2024 | | |
| **Penguji** | Kelompok 02 | | |
| **Skenario Pengujian** | | | |
| * Menonaktifkan service *about us* * Menjalankan aplikasi | | | |
| **Kriteria Evaluasi Hasil** | | | |
| Berhasil menjalankan *service* lain, meskipun *service about us* sedang *down* | | | |
| **Kasus dan Hasil Pengujian** | | | |
| **Data Masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| *Service about us* dinonaktifkan | *Service* lain tetap berjalan | Sesuai yang diharapkan | [X] diterima  [ ] ditolak |
| **Catatan** | | | |
| 1. *About us down*        1. *Service* kategori tetap berjalan | | | |

1. Pengujian Kategori

**Tabel 3. Pengujian Kategori**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Butir Uji** | Pengujian Kategori *down* | | |
| **Tujuan** | Untuk mengetahui apakah *service* lain berjalan saat *service* Kategori down | | |
| **Tanggal Pengujian** | 15/05/2024 | | |
| **Penguji** | Kelompok 02 | | |
| **Skenario Pengujian** | | | |
| * Menonaktifkan service *about us* * Menjalankan aplikasi | | | |
| **Kriteria Evaluasi Hasil** | | | |
| Berhasil menjalankan *service* lain, meskipun *service* Kategori sedang *down* | | | |
| **Kasus dan Hasil Pengujian** | | | |
| **Data Masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| *Service* Kategori  dinonaktifkan | *Service* lain tetap berjalan | Sesuai yang diharapkan | [X] diterima  [ ] ditolak |
| **Catatan** | | | |
| 1. Kategori *down*        1. *About us* tetap berjalan | | | |

1. Pengujian Review

**Tabel 4. Pengujian Review**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Butir Uji** | Pengujian Review *down* | | |
| **Tujuan** | Untuk mengetahui apakah *service* lain berjalan saat *service* Review *down* | | |
| **Tanggal Pengujian** | 15/05/2024 | | |
| **Penguji** | Kelompok 02 | | |
| **Skenario Pengujian** | | | |
| * Menonaktifkan service Review * Menjalankan aplikasi | | | |
| **Kriteria Evaluasi Hasil** | | | |
| Berhasil menjalankan *service* lain, meskipun *service* Review sedang *down* | | | |
| **Kasus dan Hasil Pengujian** | | | |
| **Data Masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| *Service* Review  dinonaktifkan | *Service* lain tetap berjalan | Sesuai yang diharapkan | [X] diterima  [ ] ditolak |
| **Catatan** | | | |
| 1. Review *down*        1. *About us* tetap berjalan | | | |

**KESIMPULAN:** Dengan menggunakan *microservice,* setiap *service* pada website yang kami bangun dapat berjalan secara independent. Artinya, ketika service lain sedang down, hal ini tidak akan memengaruhi service lainnya.