

# **Laporan Website Backend**



Kelompok 11

Witriatna Kosahi (535220125)

Maryanto (535220130)

Carlouis Fernando Haryadi (535220156)

# Daftar Isi

<b>HALAMAN JUDUL.....</b>	<b>I</b>
<b>DAFTAR ISI.....</b>	<b>II</b>
<b>BAB I      PENDAHULUAN.....</b>	<b>1</b>
1.1 LATAR BELAKANG.....	1
1.2 RUMUSAN MASALAH.....	1
1.3 TUJUAN KEGIATAN.....	2
1.4 MANFAAT KEGIATAN.....	2
1.5 METODOLOGI KEGIATAN.....	2
<b>BAB II      TEORI.....</b>	<b>4</b>
2.1 METODOLOGI.....	4
2.2 TIMELINE.....	5
2.3 NODE.JS.....	8
2.4 MONGO.DB.....	8
2.5 KONDISI TERAKHIR PROJECT AKHIR.....	9
2.6 PEMBAGIAN TUGAS.....	18
<b>BAB III      PEMBAHASAN DAN IMPLEMENTASI .....</b>	<b>19</b>
3.1 FRAMEWORK .....	19
3.2 IMPLEMENTASI.....	21
3.3 STRUKTUR KODE .....	22
3.4 PEMBAHASAN.....	23
<b>BAB IV      KESIMPULAN.....</b>	<b>24</b>
4.1 KESIMPULAN.....	24
<b>DAFTAR PUSTAKA.....</b>	<b>25</b>
<b>LAMPIRAN.....</b>	<b>26</b>

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pada era globalisasi ini, perkembangan teknologi informasi dan internet telah mengubah pola perilaku masyarakat dalam berbelanja. Terutama dalam konteks produk konsumen seperti kopi, di mana minuman ini telah menjadi bagian penting dari gaya hidup modern. Di samping itu, tren minat akan kopi kualitas tinggi dan variasi rasa semakin meningkat di kalangan konsumen.

Sejalan dengan pergeseran preferensi konsumen ini, industri kopi menghadapi tuntutan untuk meningkatkan aksesibilitas produk mereka. Perusahaan kopi, baik yang berskala besar maupun kecil, harus mengadaptasi strategi pemasaran dan distribusi mereka agar dapat mencapai target pasar yang lebih luas.

Dalam konteks ini, pembuatan sebuah platform pembelian kopi *online* muncul sebagai solusi yang relevan dan berpotensi dalam memenuhi kebutuhan konsumen yang terus berkembang. Platform ini diharapkan dapat menyediakan kemudahan bagi konsumen untuk menjelajahi dan membeli berbagai jenis kopi dari kenyamanan rumah mereka sendiri.

Selain memenuhi kebutuhan konsumen, kehadiran platform pembelian kopi *online* juga diharapkan dapat memberikan manfaat bagi pelaku industri kopi. Dengan adopsi teknologi digital, perusahaan kopi dapat memperluas jangkauan pasar mereka, meningkatkan visibilitas dalam dunia digital, memberikan edukasi kepada pelanggan tentang kopi, menyediakan platform penjualan yang efisien, membangun komunitas pecinta kopi, serta memperkuat branding dan pemasaran untuk tetap relevan dan bersaing di era digital yang penuh tantangan ini.

### 1.2 Rumusan Masalah

1. Bagaimana menyediakan layanan pelanggan yang responsif dan membantu bagi pembeli kopi online?
2. Bagaimana menangani pertanyaan, keluhan, dan pengembalian barang dengan efektif?
3. Bagaimana jika ada tambahan menu dan lokasi toko?
4. Bagaimana proses pembuatan *website* GulaJava?

## 1.3 Tujuan Kegiatan

Tujuan:

1. Mengevaluasi kinerja *website* kopi dalam mencapai tujuan bisnis seperti peningkatan penjualan dan kepuasan pelanggan.
2. Untuk mengidentifikasi tren penggunaan *website*, seperti lalu lintas pengunjung dan tingkat konversi, untuk mendukung perencanaan strategi pemasaran.
3. Untuk mengevaluasi perform konten *website*, seperti deskripsi produk dan artikel blog, untuk menentukan efektivitas dalam menarik pengunjung.

## 1.4 Manfaat Kegiatan

1. Manfaat Akademis, laporan ini dapat memberikan pengalaman belajar dalam pembuatan *website* kopi
2. Manfaat Praktis, hasil laporan ini diharapkan dapat menjadi bahan pertimbangan untuk para peminat kopi dalam pengambilan keputusan pembelian dan para pelaku usaha *coffee* berkaitan dengan masalah pengaruh promosi dan penjualan online
3. Bagi penulis diharapkan dapat menerapkan ilmu yang sudah didapatkan dan menambah pengetahuan baru terhadap kegiatan ini khususnya pada bagian *backend*.

## 1.5 Metodologi Kegiatan

Metode *incremental* adalah pendekatan dalam pengembangan proyek, khususnya dalam bidang *Software Development*, di mana proyek dibagi menjadi serangkaian rilis yang lebih kecil dan dapat dikelola. Tujuan utama dari pendekatan ini adalah untuk memberikan manfaat yang cepat dan berkelanjutan sepanjang proses pengembangan, dengan mengintegrasikan umpan balik dari pengguna pada tahap awal. Konsep-konsep dasar metode *Incremental* adalah :

1. Pengembangan Bertahap: Proyek dibagi menjadi sejumlah bagian atau "*increments*" yang dapat dikembangkan secara berurutan atau paralel. Setiap *increment* mencakup sejumlah fitur atau fungsi tertentu yang menambahkan nilai ke produk akhir.
2. Pengiriman Bertahap: Setelah setiap tahapan selesai, versi baru dari produk tersebut dirilis. Ini memungkinkan pengguna untuk menguji dan menggunakan sistem dalam kondisi semi-lengkap dan memberikan umpan balik yang dapat digunakan untuk memperbaiki atau menyesuaikan pengembangan selanjutnya.
3. Integrasi Terus-menerus: Proses integrasi berlangsung sepanjang pengembangan, di mana fitur-fitur baru secara terus-menerus diintegrasikan ke dalam sistem yang ada. Hal ini memungkinkan pengidentifikasian dan perbaikan masalah lebih awal dalam siklus pengembangan.

Metode *incremental* memiliki beberapa kelebihan dan kekurangan yaitu :

Kelebihan *Incremental* :

1. Pengiriman Cepat dan Berkelanjutan: Pendekatan ini memungkinkan organisasi untuk mengirimkan fungsi-fungsi penting lebih cepat kepada pengguna, memberikan mereka nilai lebih awal dan sering.
2. Fleksibilitas dalam Perubahan: Dengan menerima umpan balik secara berkelanjutan dari pengguna, tim pengembang dapat membuat penyesuaian pada fitur mendatang berdasarkan apa yang paling diinginkan atau dibutuhkan oleh pengguna.
3. Pengurangan Risiko: Karena sistem diuji dan direvisi secara bertahap, risiko yang berkaitan dengan proyek secara keseluruhan (seperti kesalahan besar atau kegagalan proyek) dapat diminimalkan.

Kekurangan *Incremental* :

1. Perencanaan dan Desain Awal yang Kompleks: Membutuhkan perencanaan awal yang cermat untuk menentukan bagaimana sistem akan dibagi menjadi *increments* yang bermakna.
2. Kemungkinan Biaya Lebih Tinggi: Biaya keseluruhan bisa menjadi lebih tinggi karena revisi dan penyesuaian terus-menerus berdasarkan umpan balik pengguna.
3. Kesulitan dalam Integrasi: Integrasi fitur-fitur baru dengan sistem yang sudah ada bisa jadi menantang, terutama jika arsitektur sistem awal tidak dirancang untuk mendukung perluasan atau modifikasi yang mudah.

# BAB II

## TEORI

### 2.1 Metodologi

Awalnya, pembuatan *website* GulaJava direncanakan menggunakan metode *Waterfall*, namun kemudian diputuskan untuk menggunakan metode *Incremental*.

Penggunaan metode *incremental* dalam *project* kami:

*Inkremen 1: Pengembangan Awal dan Prototyping*

- Membuat desain *Website*
- Membuat *Front-End website*.
- Mengubah *Front-End* menjadi menggunakan *EJS*.

*Inkremen 2: Penetapan Metodologi dan Pembaharuan Teknologi*

- Menentukan metodologi yang akan digunakan dalam pembuatan proyek.
- Mengupdate *website* dengan menambahkan fitur-fitur menggunakan *Angular* dan *JS*.
- Membuat *Database* menggunakan *Formspree* untuk kritik & saran.
- Mengubah metodologi pengembangan.

*Inkremen 3: Integrasi Database dan Pengembangan Fitur*

- Mengubah *Database* menjadi *MongoDB* untuk Kritik & Saran.
- Membuat fitur-fitur baru seperti *sort*, *filter* & *search*.
- Peningkatan *UI* dan Penambahan Fitur Pengguna
- Mengupdate dan memperbaiki *CSS* dari *website*.
- Membuat *layout* dan *CSS* dari *login* & *signup page*.
- Membuat *wishlist* menu.
- Menghubungkan *Login* & *signup page* ke *MongoDB*
- Memperbarui desain *CSS*

#### *Inkremen 5: Pengembangan Lanjutan dan Backend*

- Membuat model-model yang akan dihubungkan ke dalam database *MongoDB*
- Membuat *layout* dan *page Admin side* dan menghubungkan *page admin* ke *Database*.

#### *Inkremen 6: Peningkatan dan Penambahan Fitur Final*

- Membuat *page* baru untuk *wishlist*, *cart*, dan pesanan.
- Mengupdate beberapa fitur dan program.
- Menyelesaikan *user side*.

Dan terakhir Evaluasi dan Iterasi Berkelanjutan

## 2.2 Timeline

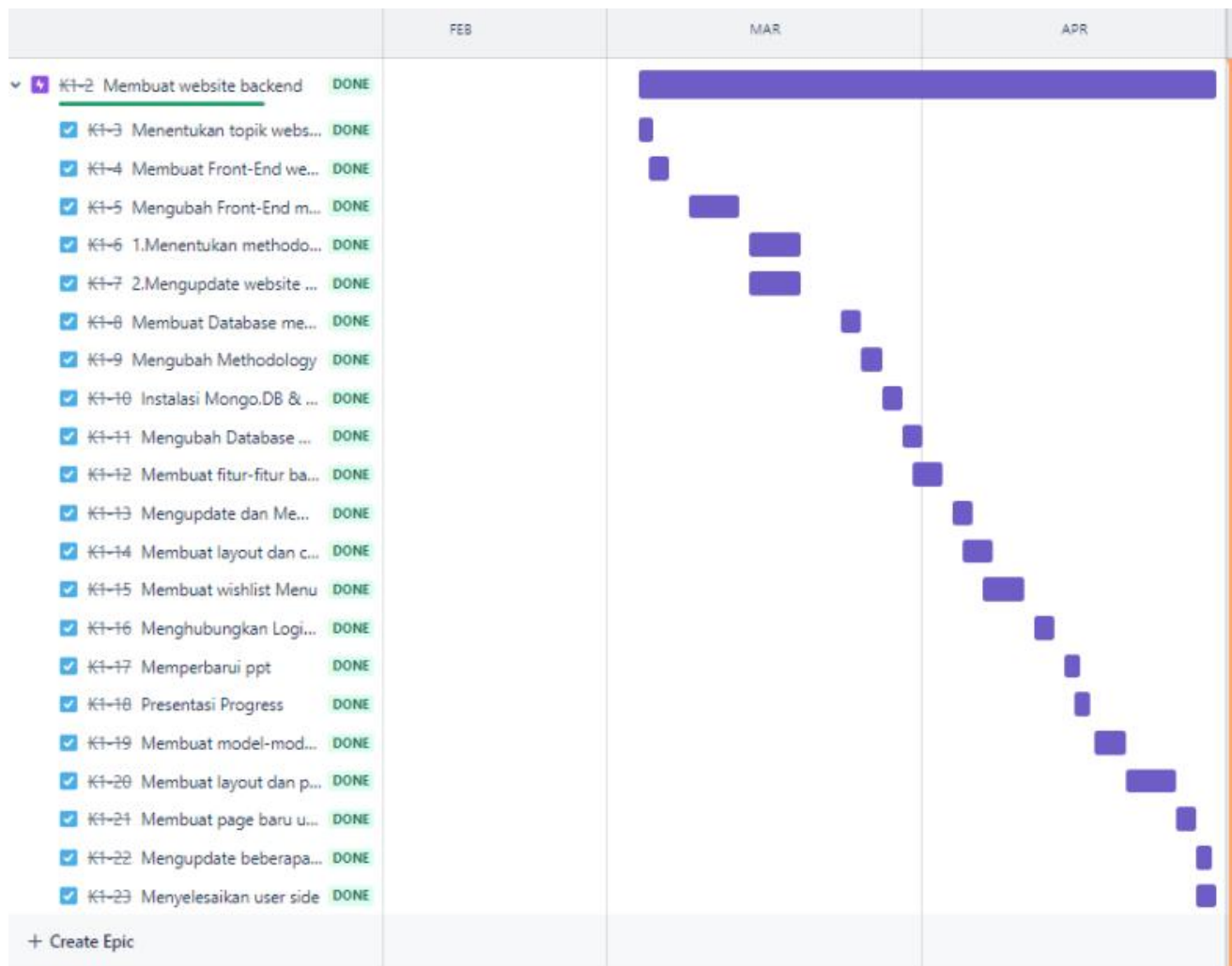
No	Tanggal	Kegiatan
1	4 Maret 2024	Menentukan topik <i>website</i>
2	5 Maret 2024	Membuat <i>Front-End website</i>
3	13 Maret 2024	Mengubah <i>Front-End</i> menjadi menggunakan <i>EJS</i>
4	19 Maret 2024	1. Menentukan <i>methodology</i> yang akan digunakan dalam pembuatan <i>project</i>  2. Mengupdate <i>website</i> dengan menambahkan fitur-fitur menggunakan <i>Angular</i> dan <i>js</i> .
5	25 Maret 2024	Membuat <i>Database</i> menggunakan <i>Formspree</i> untuk kritik & saran
6	27 Maret 2024	Mengubah <i>Methodology</i>

7	29 Maret 2024	Instalasi <i>MongoDB</i> & <i>Connect</i> <i>vscode</i> dengan <i>Mongo.DB</i>
8	30 Maret 2024	Mengubah <i>Database</i> menjadi <i>MongoDB</i> untuk Kritik & Saran
9	1 April 2024	Membuat fitur-fitur baru seperti <i>sort</i> , <i>filter</i> & <i>search</i> dan Membuat ppt presentasi
10	5 April 2024	<i>Mengupdate</i> dan Memperbagus <i>css</i> dari <i>website</i>
11	7 April 2024	Membuat <i>layout</i> dan <i>css</i> dari <i>login</i> & <i>signup page</i>
12	10 April 2024	Membuat <i>wishlist</i> Menu
13	13 April 2024	Menghubungkan <i>Login</i> & <i>signup page</i> ke <i>MongoDB</i> untuk <i>Create</i> dan <i>Read</i>
14	15 April 2024	Memperbarui ppt
15	16 April 2024	Presentasi Progres
16	18 - 20 April 2024	Membuat model-model yang akan dihubungkan ke dalam <i>database Mongo.DB</i>
17	21 - 25 April 2024	Membuat <i>layout</i> dan <i>page Admin side</i> dan menghubungkan <i>page admin</i> ke <i>Database</i>



18	26- 27 April 2024	Membuat <i>page</i> baru untuk <i>wishlist</i> , <i>cart</i> , dan pesanan
19	28 April 2024	<i>Mengupdate</i> beberapa fitur dan program
20	29 April 2024	Menyelesaikan <i>user side</i>

### Penggambaran Timeline project :



Gambar 1. Penggambaran Timeline project.

## 2.3 Node.js

*Node.js* dalam proyek ini digunakan sebagai pengganti *HTML* untuk pengembangan *backend*, hal itu memiliki beberapa alasan yaitu :

1. Kinerja Tinggi: *Node.js* dibangun di atas mesin *JavaScript V8* dari *Chrome*, yang diketahui untuk kinerjanya yang cepat. Ini membuat *Node.js* sangat efisien dalam menangani permintaan secara bersamaan dengan waktu respons yang singkat.
2. Skalabilitas: *Node.js* dirancang untuk menangani aplikasi yang *scalable* secara horizontal. Dengan arsitektur *non-blokirnya*, ia mampu menangani banyak koneksi bersamaan dengan sedikit *overhead*. Ini membuatnya ideal untuk aplikasi *real-time* yang membutuhkan banyak interaksi pengguna.
3. Ekosistem NPM: *Node.js* memiliki repositori paket *JavaScript* terbesar di dunia, *npm* (*Node Package Manager*). Ini berarti pengembang dapat dengan mudah menggunakan berbagai paket dan alat yang tersedia di ekosistem *npm* untuk mempercepat pengembangan dan mengatasi masalah yang umumnya dihadapi dalam pengembangan web.
4. Konsistensi Bahasa: Penggunaan *JavaScript* di kedua sisi, baik di *frontend* maupun *backend*, memungkinkan pengembang untuk menggunakan keterampilan yang sama dan berbagi kode antara klien dan *server*. Ini dapat mengurangi kompleksitas pengembangan dan mempercepat siklus pengembangan.
5. Komunitas yang Besar: *Node.js* memiliki komunitas yang besar dan aktif, yang berarti ada banyak sumber daya, tutorial, dan dukungan yang tersedia bagi pengembang. Hal ini membuatnya mudah untuk mendapatkan bantuan ketika menghadapi masalah atau tantangan tertentu dalam pengembangan.

## 2.4 MongoDB

Pembuatan *website* GulaJava menggunakan *database MongoDB*. *MongoDB* adalah sebuah sistem manajemen basis data (DBMS) yang bersifat dokumen dan berbasis *NoSQL*. *Database MongoDB* disebut dengan "*collections*", yang memuat sekumpulan dokumen dalam format *BSON* (*Binary JSON*). *BSON* merupakan format dokumen biner yang menyerupai *JSON* dan memungkinkan penyimpanan dan manipulasi data yang lebih kompleks.

*MongoDB* menyediakan fleksibilitas yang tinggi dalam menyimpan data, karena tidak memerlukan skema kaku seperti pada basis data relasional. Ini memungkinkan pengembang untuk menyimpan data dengan struktur yang berbeda-beda di dalam *collection* yang sama.

*MongoDB* juga memiliki beberapa fitur utama yang membantu dalam pembuatan *website* yaitu :

1. **Dokumen:** Data disimpan dalam format dokumen *BSON* yang mirip dengan *JSON*. Setiap dokumen adalah entitas tunggal yang dapat menyimpan sejumlah bidang dan nilai.
2. **Collection:** Kumpulan dokumen yang setara dengan tabel dalam basis data relasional. Setiap dokumen dalam *collection* tidak harus memiliki skema yang sama.
3. **Indeks:** *MongoDB* mendukung pembuatan indeks untuk meningkatkan kinerja pencarian data. Indeks dapat dibuat pada satu atau beberapa bidang dalam dokumen.
4. **Query:** *MongoDB* menyediakan bahasa kueri yang kuat untuk melakukan pencarian, penyaringan, dan pembaruan data.
5. **Agregasi:** *MongoDB* menyediakan fasilitas untuk melakukan operasi agregasi yang kompleks, seperti pengelompokan (*grouping*), pengurutan (*sorting*), dan perhitungan statistik.
6. **Replikasi:** *MongoDB* mendukung replikasi otomatis untuk meningkatkan ketersediaan dan ketahanan terhadap kegagalan.
7. **Sharding:** *MongoDB* mendukung *sharding* untuk menangani volume data yang besar dengan membagi data ke beberapa *node*.

## 2.5 Kondisi terakhir project akhir

Kondisi terakhir dari proyek akhir berupa *website* kopi adalah sebagai berikut: terdapat dua sisi, yaitu sisi klien dimana konsumen dapat melakukan pembelian kopi, dan sisi admin yang digunakan untuk mengatur *website* kopi. Ada beberapa fitur yang tersedia dalam *website* ini yaitu :

### 1. *Client side*

#### A. *Navbar dan Footer*

*Navbar* merupakan bagian teratas dari *website* yang berfungsi untuk mempermudah para konsumen mengakses berbagai bagian yang berbeda dari *website*, misalnya dari halaman beranda ke menu dan sebagainya.

[Beranda](#)[Menu](#)[Kritik Saran](#)[Wishlist](#)[Cart](#)[Pesanan](#)[Settings](#)[Logout](#)

### Gambar 2. Tampilan *Navbar*

Sedangkan *footer* adalah bagian bawah dari *website* yang berisi informasi tentang GulaJava, seperti media sosial, alamat, dan nomor telepon.



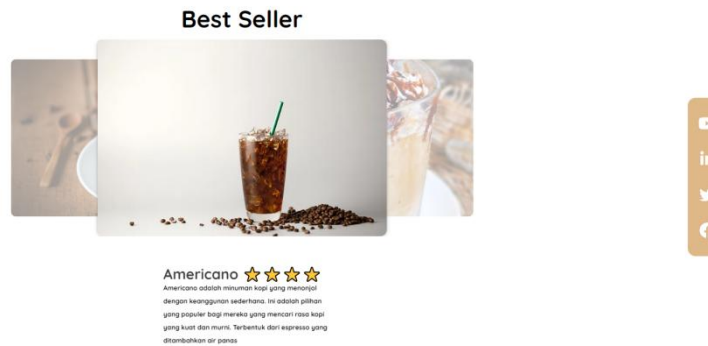
**Gambar 3.** Tampilan *Footer*

## B. Beranda

Beranda merupakan halaman awal yang pertama kali muncul saat mengakses *website*. Pada bagian beranda, terdapat penjelasan mengenai asal usul *website* kopi GulaJava. Selanjutnya, terdapat tampilan *best seller* yang memuat menu-menu favorit yang sering dibeli oleh konsumen. Setelahnya, terdapat bagian kritik dan saran yang berfungsi sebagai sarana bagi konsumen untuk memberikan *feedback* jika mereka merasa ada kekurangan dalam pengalaman menggunakan *website* kami.



**Gambar 4.** Tampilan beranda



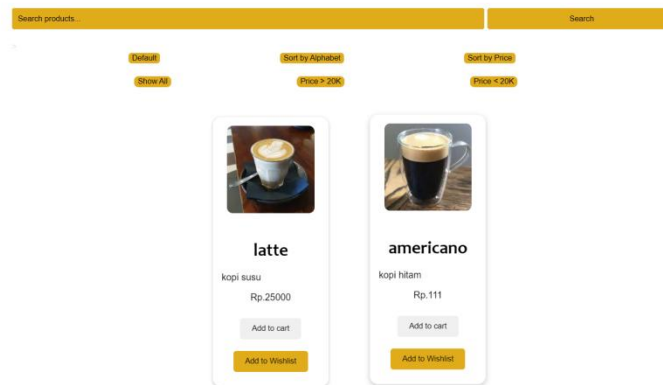
**Gambar 5.** Tampilan *Best seller*

**Gambar 6.** Tampilan kritik dan saran

### C. Menu

Di dalam bagian menu, terdapat beberapa fitur yang mempermudah konsumen dalam mencari menu yang mereka inginkan. Pertama adalah fitur pencarian (*search*), di mana konsumen dapat mencari menu yang mereka inginkan dengan mudah. Kedua, terdapat fitur pengurutan (*sort*) yang memungkinkan konsumen mengurutkan menu sesuai preferensi mereka, misalnya berdasarkan abjad atau harga. Fitur selanjutnya adalah filter, yang memungkinkan konsumen untuk menyaring menu sesuai dengan kriteria tertentu, seperti harga di bawah Rp.20.000.

Selain itu, di bagian menu juga terdapat fitur "*add to wishlist*" dan "*add to cart*". Fitur "*add to wishlist*" memungkinkan konsumen untuk menyimpan menu yang dipilih ke dalam daftar keinginan (*wishlist*), sehingga menu tersebut tetap tersedia untuk dilihat bahkan setelah keluar dari *website*. Sedangkan fitur "*add to cart*" digunakan ketika konsumen ingin melakukan pembelian, dengan menambahkan menu yang dipilih ke dalam keranjang belanja.



**Gambar 7.** Tampilan Menu

#### D. *Wishlist*

*Wishlist* merupakan bagian dimana konsumen dapat menyukai atau menyimpan menu-menu yang tersedia ke dalam daftar keinginan mereka. Menu-menu yang sudah dimasukkan ke dalam *wishlist* akan tersimpan di akun konsumen, bahkan ketika mereka keluar dari *website* ini.

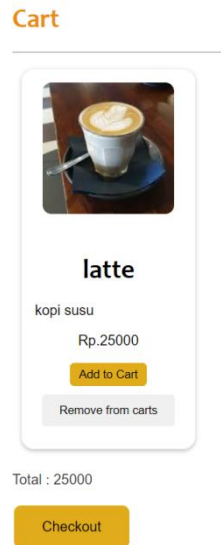
#### Wishlist



**Gambar 8.** Tampilan *Wishlist*

#### E. *Cart*

*Cart* adalah bagian dimana konsumen sudah ingin melakukan pembayaran, di dalam *cart* terdiri dari menu yang akan di beli, total harga keseluruhan menu dan juga tombol yang akan mengarahkan ke bagian menu.



**Gambar 9.** Tampilan *Cart*

#### F. Pesanan

Di dalam bagian pesanan, terdapat *invoice* pesanan yang akan dibayar oleh konsumen. Setelah pesanan dibayar, status *invoice* akan diperbarui menjadi "sudah dibayar".

##### Daftar Pesanan

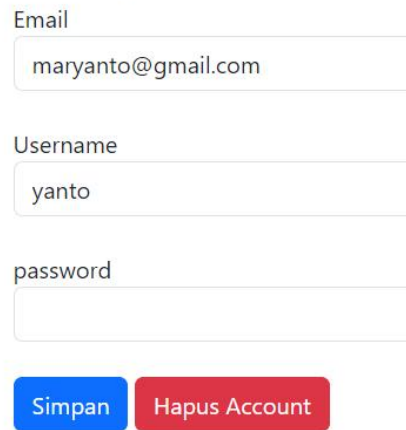
Pesanan 663654a416bf06e9b7f0879c
Total Harga: 25000
Status: Belum Selesai
Detail Pesanan:
Nama latte Harga : 25000

**Gambar 10.** Tampilan Pesanan

#### G. Settings

Di bagian ini, para konsumen dapat mengatur dan mengubah akun email, mengganti nama *username* konsumen dan juga mengganti *password*. konsumen juga bisa menghapus akunnya.

## Setting Account



Email

maryanto@gmail.com

Username

yanto

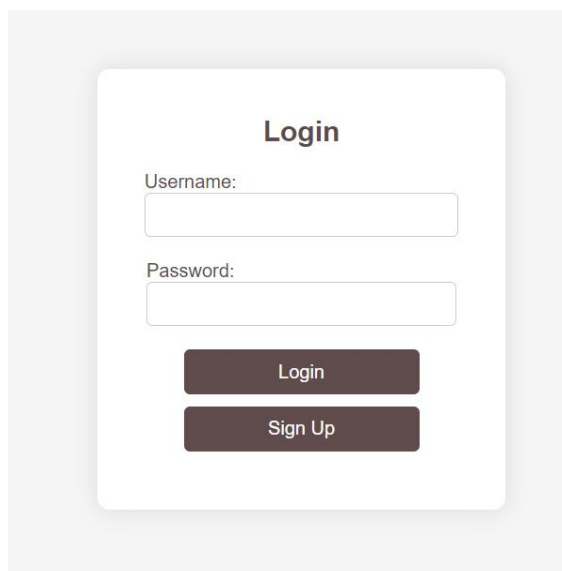
password

Simpan Hapus Account

**Gambar 11.** Tampilan *Settings*

### *H. Account*

*Account* adalah bagian yang sangat penting untuk melakukan pembelian kopi. Ketika pengunjung membuka *website* dan belum *login*, mereka hanya bisa mengakses bagian beranda. Untuk melakukan pembelian, pengunjung perlu *login* atau melakukan registrasi terlebih dahulu jika belum memiliki akun.



Login

Username:

Password:

Login Sign Up

**Gambar 12.** Tampilan *Login*



**Sign Up**

Email:

Username:

Password:

**Sign Up**

**Gambar 13.** Tampilan *Sign Up*

## 2. *Admin side*

### A. Pesanan

Admin memiliki fungsi untuk melihat pesanan pelanggan yang telah memesan minuman. Pada bagian ini, admin memiliki fitur yang memungkinkan mereka untuk menyelesaikan atau membatalkan pesanan yang masuk.

#### Pesanan



Order id	Total Price	Status	Customer name	Action
663654a416bf06e9b7f0879c	Rp.25000	Belum Selesai	yanto	<a href="#">Selesai</a> <a href="#">Tolak</a>

**Gambar 14.** Tampilan Pesanan Admin

### B. Menu

Admin memiliki fungsi untuk melakukan beberapa tindakan terhadap daftar menu. Mereka dapat menambahkan menu baru ke dalam daftar menu, menghapus menu yang sudah ada, dan mengedit detail dari menu, seperti nama menu, penjelasan menu, harga menu, dan gambar menu.

## Menu

Tambah Menu				
Name	Description	Price	Images	Action
latte	kopi susu	25000		<a href="#">edit</a> <a href="#">hapus</a>
americano	kopi hitam	111		<a href="#">edit</a> <a href="#">hapus</a>

**Gambar 15.1.** Tampilan Menu Admin

## Tambah Menu

Nama:

Deskripsi:

Harga:

Image  
 No file chosen

**Gambar 15.2.** Tampilan untuk Menambah Menu dari Admin

### C. Saran

Admin memiliki fungsi untuk mengedit isi kritik dan saran yang sudah diberikan oleh para pelanggan. Selain itu, mereka juga dapat menghapus kritik dan saran yang telah diberikan oleh pelanggan jika diperlukan.

## Saran

Email	Message	Action
maryanto@gmail.com	lol	<a href="#">edit</a> <a href="#">hapus</a>

**Gambar 16.** Tampilan Saran Admin

## D. Setting

Admin memiliki fungsi untuk melakukan beberapa tindakan terhadap daftar toko. Mereka dapat menambahkan toko baru yang akan segera dibuka ke dalam daftar toko, menghapus toko dari daftar toko yang sudah dibuka, dan mengedit detail dari toko yang terdaftar, seperti alamat, jam buka, dan kontak yang dimiliki oleh toko tersebut.

## Setting Toko

Tambah Data Toko			
Alamat	Jam Buka	Kontak	action
untar	00:12	1221	<a href="#">edit</a> <a href="#">hapus</a>

**Gambar 17.** Tampilan Setting Admin

## 2.6 Pembagian Tugas

Pembagian tugas untuk setiap anggota kelompok adalah sebagai berikut:

Carlouis Fernando Hariyadi:

- Membuat tampilan *website*.
- Membuat *database MongoDB* pada *website*.
- Mengimplementasikan fitur *CRUD* pada *website*.
- Membuat fitur untuk mengedit dan menghapus akun.
- Membuat *admin side*.
- Membuat video penjelasan seluruh *admin side*, menu, kritik saran, *cart*, pesanan, *login*, register, dan *setting*.
- Membuat laporan.

Maryanto:

- Membuat tampilan *website*.
- Mengimplementasikan fitur *wishlist*.
- Mengimplementasikan fitur *cart*.
- Mengimplementasikan fitur *search*.
- Mengimplementasikan fitur kritik dan saran.
- Membuat video penjelasan *navbar*, *footer*, dan beranda
- Membuat laporan.

Witriatna Kosahi:

- Membuat tampilan *website*.
- Mengimplementasikan fitur *sort*.
- Mengimplementasikan fitur filter.
- Mengimplementasikan fitur *order*.
- Membuat video penjelasan *wishlist*, *powerpoint* latar belakang, metodologi, dan kesimpulan.
- Mengedit video.
- Membuat laporan.

# BAB III

## IMPLEMENTASI DAN PEMBAHASAN

### 3.1 Framework

*Website* GulaJava menggunakan tiga *framework* utama dalam pengembangannya, yaitu *MongoDB*, *EJS*, dan *Express*. Dibawah adalah penjelasan mengenai implementasi ketiga *framework* pada *website* GulaJava.

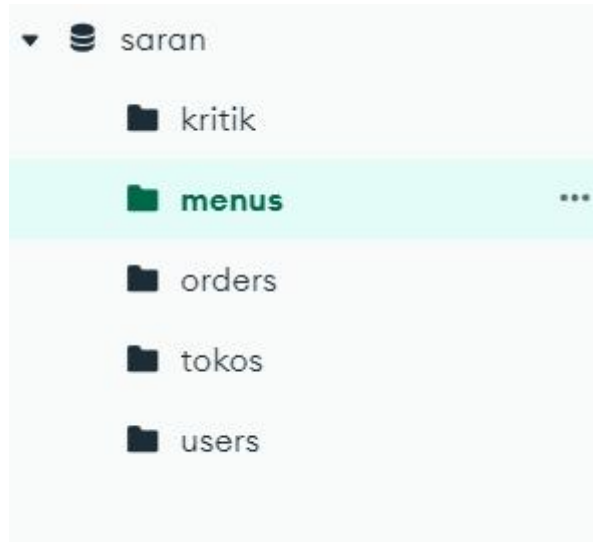
#### 1. *MongoDB*

Pengembangan *website* ini menggunakan *MongoDB* karena fleksibilitasnya dalam menyimpan data, yang memudahkan pengelolaan dan manipulasi data. *MongoDB* juga dapat menangani data dalam jumlah besar secara efisien.

Penggunaan *MongoDB* dalam *website* ini terutama terjadi di bagian akun, di mana *MongoDB* membuat dan menyimpan data saat pertama kali pengguna mendaftar (*Sign up*), dan saat pengguna melakukan *login*, *database* membaca dan mengingat data yang telah disimpan sebelumnya. Saat data mengalami perubahan, *MongoDB* akan melakukan *update* ke data yang terbaru, serta dapat menghapus data yang tidak diperlukan.

Fitur *Admin side* dalam *website* GulaJava terhubung ke *MongoDB*, sehingga perubahan yang dilakukan di dalam *web Admin side* akan tercermin langsung dalam *database MongoDB*. Hal ini terjadi dalam beberapa *page Admin side*, antara lain:

1. *Kritik*: Data yang dimasukkan ke dalam *database* meliputi nama, email, dan pesan yang disampaikan oleh pengguna.
2. *Menus*: Menampilkan data menu yang telah ditambahkan oleh admin ke dalam *database*.
3. *Orders*: Menampilkan data *invoice* yang mencakup *orderid*, total harga, dan status pesanan yang disimpan dalam *database*.
4. *Tokos*: Bertanggung jawab untuk menambahkan data baru seperti alamat, jam buka, dan kontak toko ke dalam *database*.
5. *Users*: Menyimpan data pengguna seperti alamat email, nama, dan *password* ke dalam *database MongoDB*.



**Gambar 18.** Tampilan *Database*

## 2. *EJS(Embedded Javascript)*

Dengan menggunakan *EJS*, *website* yang dibuat akan memiliki tampilan yang lebih dinamis. *EJS* memungkinkan penyisipan kode *JavaScript* langsung ke dalam *file HTML*, sehingga memungkinkan pembuatan tampilan yang dinamis dan responsif berdasarkan data yang diperoleh dari server atau basis data.

Selain itu, *website* ini menggunakan *framework Express*, yang dipilih karena kemudahannya dalam penggunaan dan kesesuaiannya dengan *Node.js*.

## 3. *Express*

*Express* adalah sebuah *framework* untuk *Node.js* yang menyediakan alat dan fitur yang mempermudah pengembangan server *HTTP*. *Framework* ini membantu dalam penanganan rute, autentikasi, dan interaksi dengan *database* dalam pengembangan *website*. Kelebihan penggunaan *Express* adalah menyediakan struktur dasar untuk aplikasi *web* dan *API*, menyederhanakan pembuatan dan manajemen server, serta mendukung banyak *plugin* dan *middleware* untuk memperluas fungsinya. Dengan demikian, penggunaan *Express* dalam pengembangan web membantu meningkatkan efisiensi dan keteraturan dalam pemrosesan permintaan dan respons.

## 3.2 Implementasi

### 1. *Express* dan *EJS*

Untuk mengonversi *file HTML* ke format *EJS*, pertama-tama buka terminal dan inisialisasi proyek menggunakan perintah “*npm init -y*”. Ini akan menghasilkan *file package.json* secara otomatis. Selanjutnya, instal *Express* dan *EJS* dengan memasukkan “*npm install express ejs*” ke dalam terminal. Setelah instalasi selesai, buat *file* bernama *index.js* untuk menulis kode server. Terakhir, ubah ekstensi semua *file HTML*, dari *.html* menjadi *.ejs*. Sebagai contoh, *file* yang bernama *menu.html* harus diubah menjadi *menu.ejs*.

### 2. *MongoDB*

Untuk menyambungkan *MongoDB* ke dalam proyek, langkah pertama adalah instal *MongoDB*, kemudian instal *extension "MongoDB for VSCode"* pada *Visual Studio Code*. Buka terminal, jalankan perintah “*npm install mongoose*” untuk memasang *Mongoose*. Setelah instalasi selesai, buka *file index.js* dan tambahkan kode “*const mongoose = require('mongoose')*” untuk mengimpor modul *Mongoose*. Terakhir, ikuti langkah-langkah pada gambar 16 untuk menyambungkan *mongoose* ke *MongoDB*.

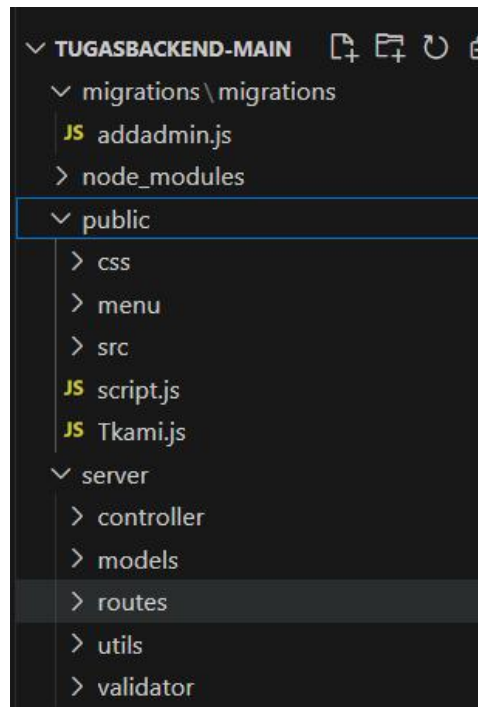
```
// Connect to MongoDB
mongoose.connect('mongodb://localhost:27017/saran', {
  useNewUrlParser: true,
  useUnifiedTopology: true
});

const db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));
db.once('open', function () {
  console.log('Connected to MongoDB');
});
```

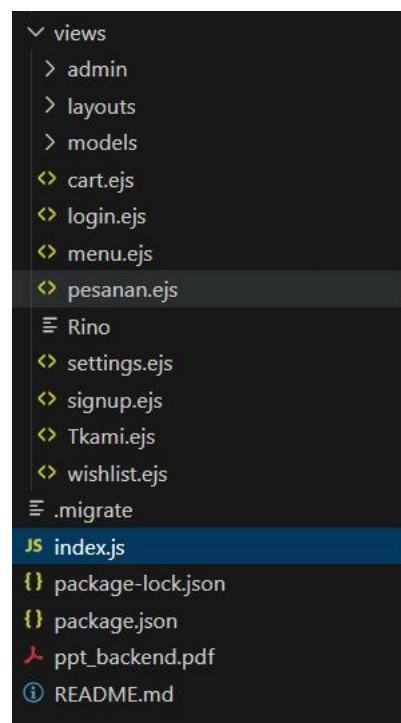
**Gambar 19.** Program untuk menyambungkan Mongoose ke MongoDB

Ganti 'mongodb://localhost:27017/saran' dengan *URL* koneksi yang sesuai dengan konfigurasi *MongoDB* yang diinginkan.

### 3.3 Struktur Kode



**Gambar 20.** Struktur program 1



**Gambar 21.** Struktur Program 2



### 3.4 Pembahasan

Dalam proses implementasi, pengembangan *website* ini menghadapi berbagai tantangan dan pertimbangan yang perlu dibahas. Salah satu tantangan utama adalah dalam mengelola koneksi dan operasi basis data *MongoDB*, terutama dalam menangani akses data yang kompleks dan transaksi. Dalam hal ini, penggunaan *library* dan modul tertentu dapat membantu mempermudah pengelolaan basis data.

Kombinasi *framework MongoDB*, *EJS*, dan *Express* membawa keuntungan dalam pengembangan proyek *website* ini. *MongoDB* memberikan fleksibilitas dalam penyimpanan data yang cocok untuk aplikasi dengan skema data yang berubah-ubah. *EJS* memungkinkan pembuatan tampilan dinamis dengan mudah, sementara *Express* menyediakan kerangka kerja yang kuat untuk mengelola rute dan logika *website*. Dengan terus memperbaharui dan meningkatkan struktur kode serta mempertimbangkan solusi terbaik dalam menghadapi tantangan teknis, proyek ini akan berhasil mencapai tujuan yang ditetapkan.

## KESIMPULAN

*Website* GulaJava adalah sebuah platform penjualan kopi melalui *website* yang memungkinkan akses yang mudah bagi konsumen tanpa perlu mengunduh aplikasi khusus seperti yang umumnya terjadi pada merek dan *brand* kopi lainnya. Pengembangan awal *website* ini mengadopsi metode *waterfall*, namun karena kebutuhan untuk terus melakukan pembaharuan, pendekatan pengembangan kemudian diubah menjadi *incremental*. Perubahan dari metode *waterfall* ke metode *incremental* terbukti berhasil. Dalam pendekatan ini, Waktu pengembangan yang lebih singkat dan penambahan fitur-fitur dan perbaikan dilakukan secara bertahap.

Keuntungan utama dari penggunaan *website* GulaJava adalah kemudahan akses bagi konsumen. Dengan hanya melakukan *login*, konsumen dapat langsung membeli kopi yang mereka inginkan. *Website* ini juga menyediakan fitur-fitur seperti pencarian, penyortiran, dan penyaringan untuk memudahkan konsumen dalam memilih menu. Selain itu, adanya fitur *wishlist* memungkinkan konsumen untuk menyimpan pilihan mereka untuk pembelian di masa mendatang.

Sementara itu, sisi admin *website* juga memberikan kemudahan bagi pengelola. Admin dapat dengan mudah mengatur menu, termasuk menambahkan dan menghapus menu serta mengatur harga. Mereka juga memiliki kemampuan untuk menangani pesanan, baik menyelesaikan maupun membatalkan pesanan, serta mengakses berbagai fitur yang mempermudah pengelolaan.

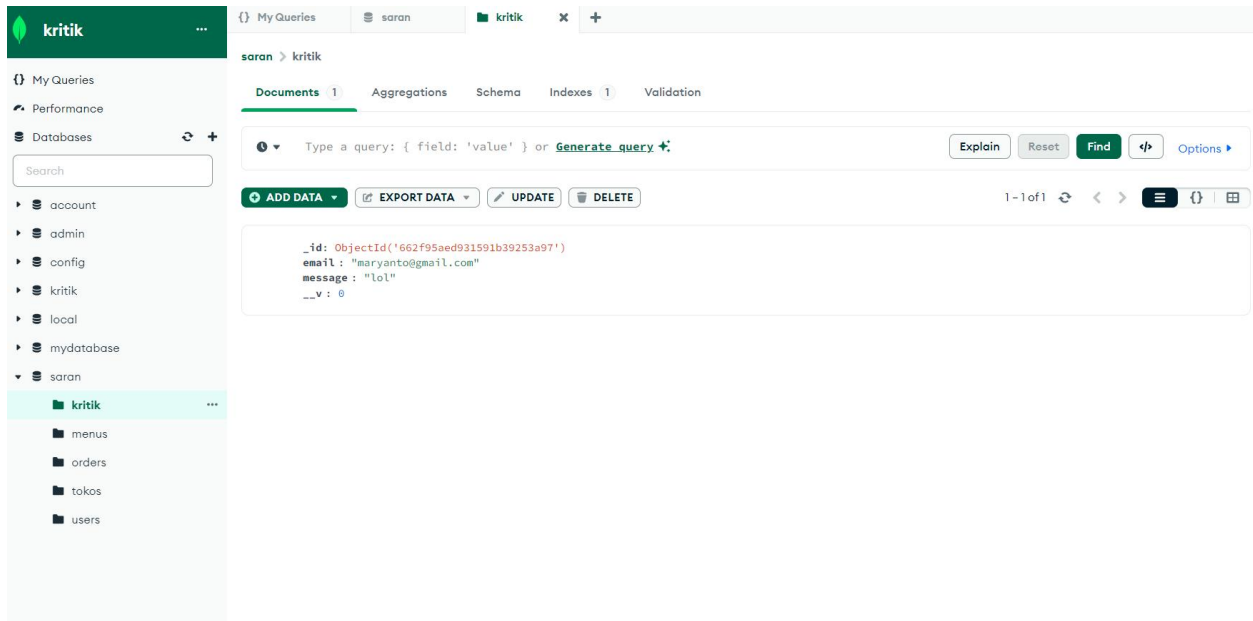
Proses pengembangan *website* ini berlangsung selama sekitar dua bulan. Pada minggu pertama, fokus pada konseptualisasi dan pembuatan bagian awal *website* menggunakan *HTML* dan *CSS*. Minggu berikutnya, bahasa pemrograman diubah menjadi *EJS*, dan kemudian di minggu-minggu berikutnya, *website* terus menerima penambahan fitur-fitur baru.

Untuk pengembangan masa depan, disarankan untuk terus memperluas fitur-fitur yang ada dan meningkatkan tampilan untuk mempertahankan daya tarik visual yang menarik bagi pengguna. Selain itu, penting untuk terus mengembangkan strategi pemeliharaan dan peningkatan dalam segala aspek.

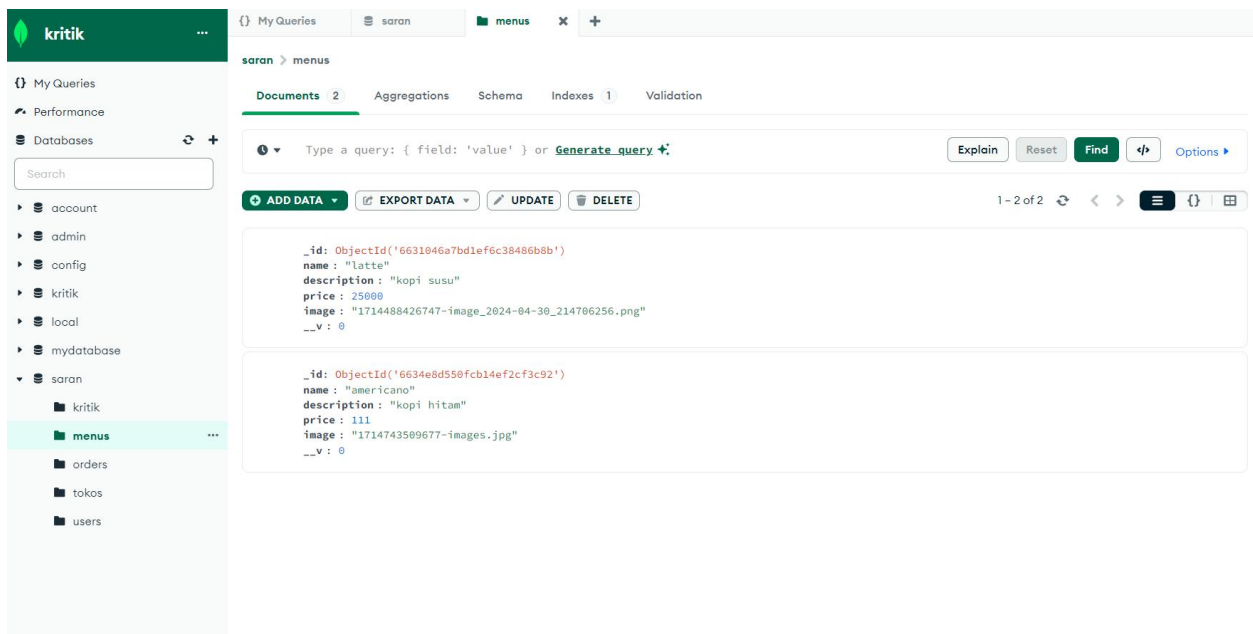
## DAFTAR PUSTAKA

- [1] Mamikos, “7 Contoh Hasil dan Pembahasan Laporan Berbagai Kegiatan serta Cara Membuatnya”, mamikos, 29 oktober 2023[Online]. Tersedia : [https://mamikos.com/info/contoh-hasil-dan-pembahasan-laporan-berbagai-kegiatan-pljr/#goog\\_rewarded](https://mamikos.com/info/contoh-hasil-dan-pembahasan-laporan-berbagai-kegiatan-pljr/#goog_rewarded) [Diakses : 05 Mei 2024].
- [2] D. Mania, “CRUD App Node Js, Express, MongoDB & EJS Templating Engine”,youtube,18 Maret 2021[Online]. Tersedia : <https://www.youtube.com/playlist?list=PL6u82dzQtlfvJoAWdyf5mUxPQRnNKCMGt> [Diakses : 23 April 2024].
- [3] A. Ceylan Öncül, “Playlist Carousel”,Codepen, 13 Juni 2020[Online]. Tersedia : <https://codepen.io/aybukeceylan/pen/RwrRPoO> [Diakses : 5 Maret 2024].

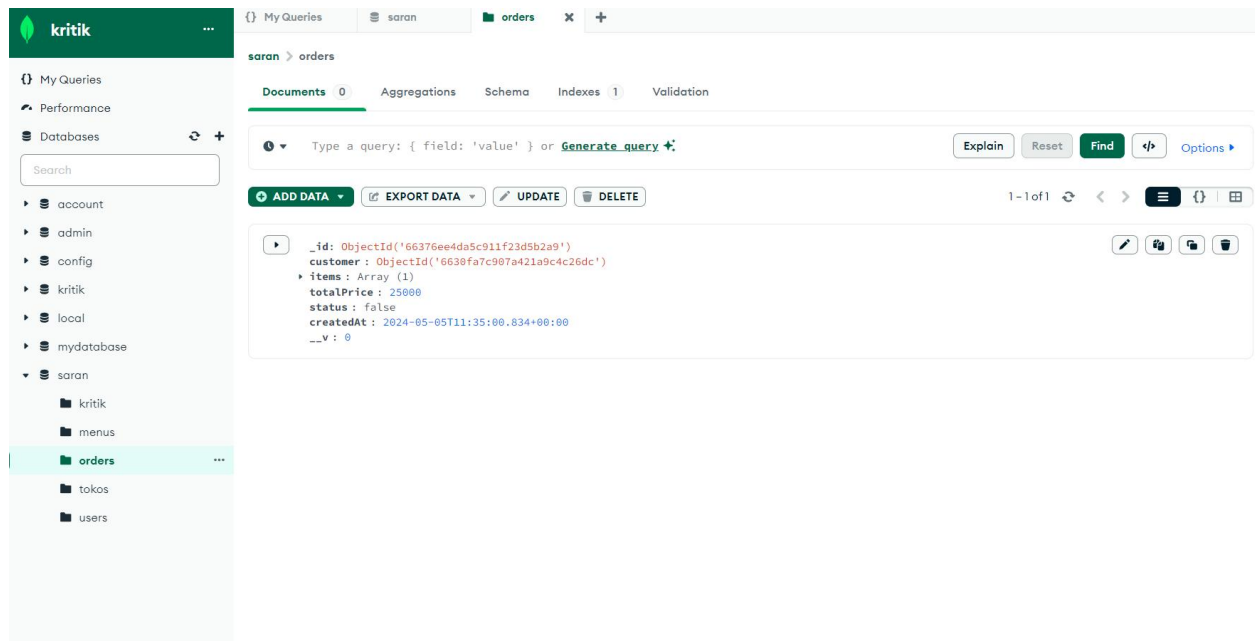
# LAMPIRAN



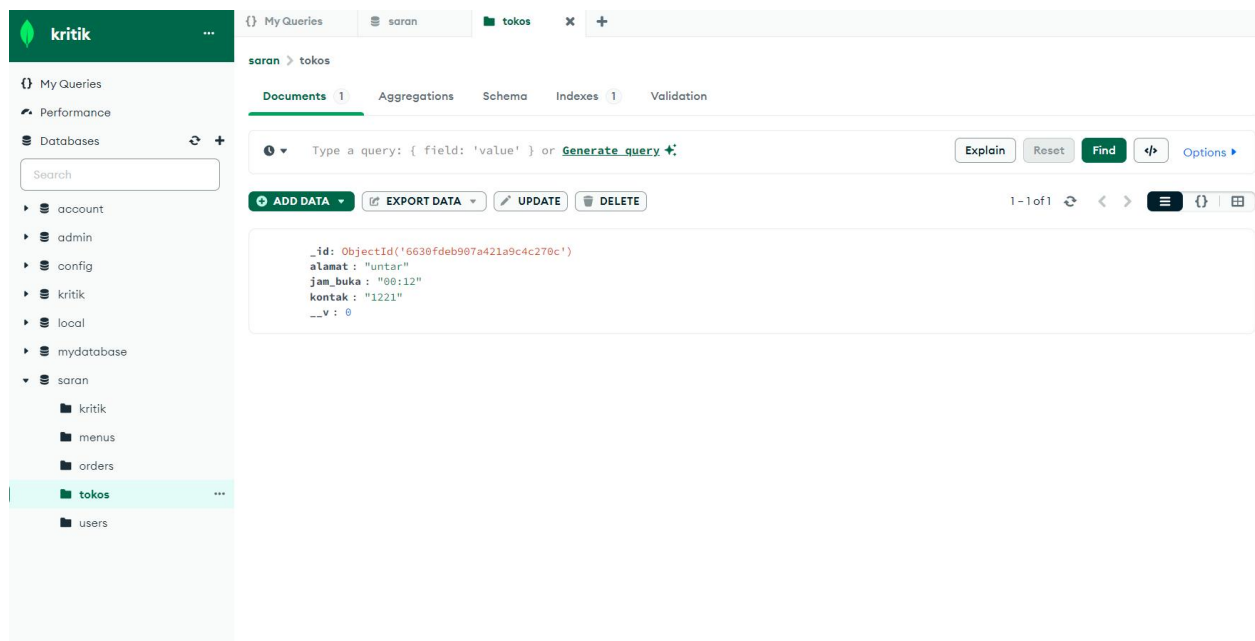
Lampiran 1. Tampilan MongoDB untuk database bagian kritik dan saran.



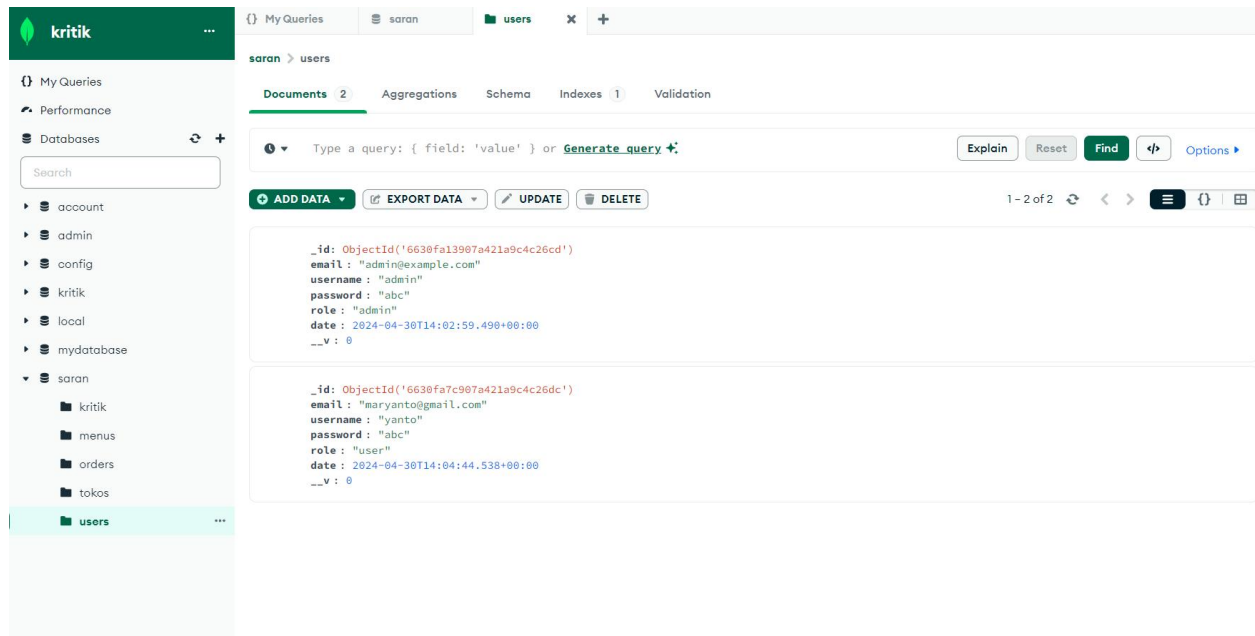
Lampiran 2. Tampilan MongoDB untuk database bagian menu.



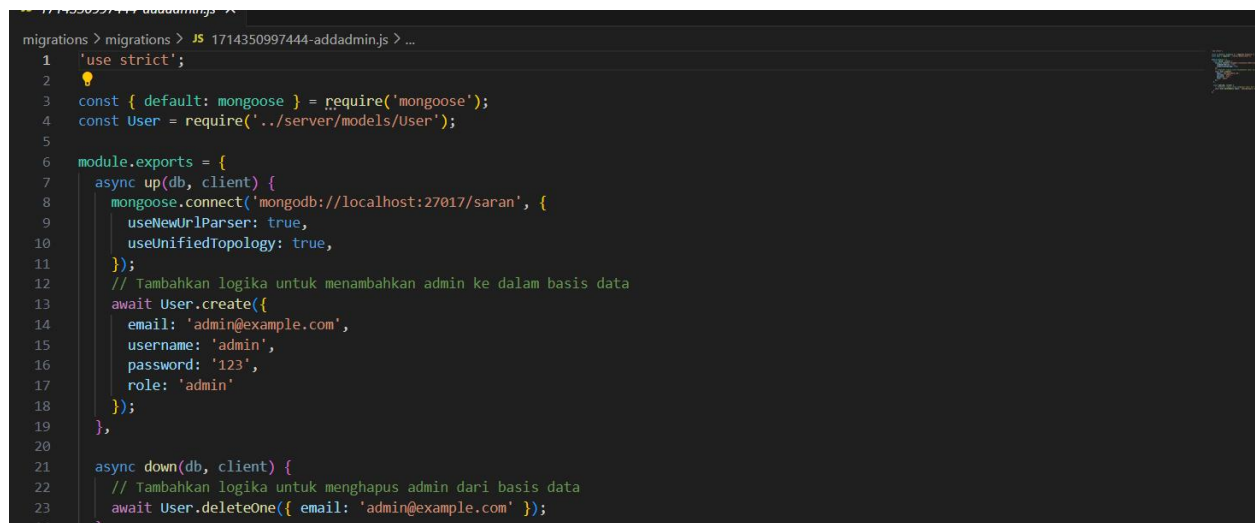
**Lampiran 3.** *Tampilan MongoDB untuk database bagian pesanan.*



**Lampiran 4.** *Tampilan MongoDB untuk Database bagian toko(admin side).*



Lampiran 5. Tampilan MongoDB untuk dataBase bagian Users.



Lampiran 6. membuat addadmin menggunakan namafile.js.

```

JS index.js > ...
1  const express = require('express');
2  const app = express();
3  const port = 3000;
4  const { errors } = require('celebrate');
5
6  const ejs = require('ejs');
7  const mongoose = require('mongoose');
8  const bodyParser = require('body-parser');
9  const { default: AuthenticationController } = require('./server/controller/AuthenticationController');
10 const api = require('./server/routes/api');
11 const adminRoutes = require('./server/routes/adminroute');
12 const userRoutes = require('./server/routes/userroute');
13
14 const session = require('express-session');
15 const { default: axios } = require('axios');
16
17
18 // Middleware to parse JSON and urlencoded bodies
19 app.use(bodyParser.urlencoded({ extended: false }));
20 app.use(bodyParser.json());
21
22 // set template engine ejs
23 app.set('view engine', 'ejs');

```

Lampiran 7. macam - macam modul yang digunakan.

**LINK YOUTUBE :**

**Demo Program :** <https://youtu.be/rHYhXHYQOy0>

**Powerpoint :** <https://youtu.be/6WEIjAQGYRs>