MAKALAH

Pengembangan Aplikasi Kasir Sederhana Berbasis CMS dengan Menggunakan Laravel

Untuk Memenuhi Persyaratan
Tugas ke-4 Pemrograman Web

Disusun Oleh :
Umi Kalsum 12221942

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN ILMU KOMPUTER EL RAHMA YOGYAKARTA

2023/2024

1. Pendahuluan

1.1 Latar Belakang

Sistem manajemen konten (CMS) telah menjadi komponen penting dalam berbagai aplikasi, termasuk aplikasi kasir. CMS memungkinkan pengelolaan konten yang lebih mudah dan terstruktur, sehingga bisnis dapat mengoptimalkan operasi sehari-hari mereka. Dalam konteks aplikasi kasir, CMS membantu dalam pengelolaan produk, transaksi, dan laporan keuangan dengan lebih efisien.

Laravel adalah salah satu framework PHP yang populer dan sering digunakan untuk pengembangan aplikasi web. Dikenal dengan sintaks yang elegan dan fitur-fitur canggih, Laravel menyediakan alat dan sumber daya yang diperlukan untuk membangun aplikasi web yang robust dan scalable. Dengan menggunakan Laravel, pengembangan aplikasi kasir yang memiliki CMS menjadi lebih cepat dan mudah, karena framework ini mendukung berbagai fitur seperti routing, middleware, dan ORM yang kuat.

1.2 Rumusan Masalah

Pembuatan CMS untuk aplikasi kasir bertujuan untuk mengatasi berbagai masalah yang sering dihadapi oleh bisnis dalam mengelola transaksi dan data penjualan. Salah satu masalah utama adalah kesulitan dalam mengelola dan mengorganisasi data produk, pelanggan, dan transaksi secara efisien. Banyak bisnis kecil dan menengah masih menggunakan metode manual atau sistem yang tidak terintegrasi, sehingga rentan terhadap kesalahan dan memerlukan waktu yang lebih lama dalam proses pengelolaan.

Tujuan utama dari pengembangan aplikasi ini adalah untuk menyediakan solusi yang terintegrasi dan efisien bagi bisnis dalam mengelola data penjualan dan transaksi. Aplikasi ini diharapkan dapat meningkatkan akurasi dan efisiensi operasional dengan menyediakan antarmuka yang user-friendly dan fitur-fitur yang memudahkan pengelolaan data. Selain itu, aplikasi ini juga bertujuan untuk mendukung bisnis dalam membuat keputusan yang lebih baik berdasarkan data yang lebih terstruktur dan mudah diakses.

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk menyediakan solusi CMS yang efektif dan efisien bagi aplikasi kasir. Solusi ini diharapkan dapat membantu bisnis dalam mengelola data produk, transaksi, dan laporan keuangan secara lebih terstruktur dan mudah diakses. Selain itu, penelitian ini juga berfokus pada pemanfaatan dan pendalaman fitur-fitur Laravel sebagai framework PHP untuk membangun CMS yang andal dan scalable.

Penelitian ini juga ditujukan untuk memenuhi penilaian tugas mata kuliah "Pemrograman Web", di mana mahasiswa diharapkan dapat mengaplikasikan pengetahuan dan keterampilan yang telah dipelajari selama perkuliahan dalam sebuah proyek nyata yang relevan dengan kebutuhan industri.

1.4 Manfaat Penelitian

Penelitian ini memiliki manfaat praktis dan teoritis yang signifikan. Dari sudut pandang praktis, pengembangan CMS untuk aplikasi kasir diharapkan dapat memberikan solusi yang lebih efisien dan efektif dalam mengelola data penjualan dan transaksi. Solusi ini dapat membantu bisnis dalam meningkatkan produktivitas dan mengurangi kesalahan yang sering terjadi dalam pengelolaan data secara manual.

Dari sudut pandang teoritis, penelitian ini akan menambah wawasan tentang penggunaan dan implementasi framework Laravel dalam pengembangan aplikasi web. Penelitian ini juga diharapkan dapat memberikan kontribusi pada literatur yang ada mengenai pengembangan CMS dan aplikasi kasir, sehingga dapat menjadi referensi bagi penelitian-penelitian selanjutnya.

Sebagai mahasiswa, penelitian ini memberikan kesempatan untuk mengaplikasikan teori dan konsep yang telah dipelajari dalam mata kuliah "Pemrograman Web" ke dalam sebuah proyek nyata. Hal ini akan meningkatkan pemahaman dan keterampilan teknis dalam pengembangan aplikasi web menggunakan Laravel, serta memberikan pengalaman praktis yang berharga dalam merancang dan mengimplementasikan solusi perangkat lunak yang kompleks.

2. Tinjauan Pustaka

2.1 Pengertian CMS (Content Management System)

Content Management System (CMS) adalah sebuah aplikasi perangkat lunak yang memungkinkan pengguna untuk membuat, mengelola, dan memodifikasi konten digital secara mudah tanpa memerlukan keahlian teknis yang mendalam. CMS menyediakan antarmuka pengguna yang intuitif dan alat-alat yang mempermudah proses pengelolaan konten.

Fungsi utama dari CMS meliputi:

- Pengelolaan Konten: Memungkinkan pengguna untuk menambah, mengedit, dan menghapus konten dengan mudah.
- Pengelolaan Media: Memungkinkan penyimpanan dan pengorganisasian gambar, video, dan file lainnya.
- Pengelolaan Pengguna: Memberikan kemampuan untuk mengatur hak akses dan peran pengguna yang berbeda dalam sistem.

- Pengaturan Situs: Memungkinkan konfigurasi tampilan dan fungsi situs web tanpa perlu mengubah kode sumber.

2.2 Aplikasi Kasir

Aplikasi kasir adalah perangkat lunak yang digunakan oleh bisnis untuk memproses transaksi penjualan. Fungsinya meliputi pencatatan penjualan, pengelolaan stok barang, dan pembuatan laporan keuangan. Aplikasi ini membantu bisnis dalam mengotomatisasi proses penjualan, sehingga meningkatkan efisiensi operasional dan mengurangi kesalahan manual.

Pada aplikasi kasir yang dibahas dalam penelitian ini, hanya mencakup dua fitur utama: **menu** dan **order**. Fitur menu digunakan untuk mengelola daftar produk yang tersedia, sedangkan fitur order digunakan untuk memproses transaksi penjualan dari pelanggan. Kedua fitur ini adalah inti dari operasi aplikasi kasir dan memastikan bahwa proses penjualan berjalan dengan lancar dan terorganisir.

2.3 Laravel Framework

Laravel adalah salah satu framework PHP yang paling populer dan digunakan secara luas dalam pengembangan aplikasi web. Kelebihan Laravel dalam pengembangan aplikasi web antara lain:

- **Sintaks yang Elegan:** Laravel menawarkan sintaks yang bersih dan elegan, yang memudahkan pengembang untuk menulis dan memahami kode.
- **Modular dan Skalabel:** Laravel memiliki struktur modular yang memungkinkan pengembangan aplikasi yang skalabel dan mudah diatur.
- **Banyak Fitur Bawaan:** Laravel dilengkapi dengan berbagai fitur bawaan seperti routing, middleware, autentikasi, dan ORM (Eloquent) yang mempermudah pengembangan aplikasi.
- **Komunitas yang Besar:** Laravel didukung oleh komunitas yang besar dan aktif, menyediakan dokumentasi yang baik, tutorial, dan berbagai paket tambahan yang dapat digunakan untuk mempercepat pengembangan.
- **Keamanan:** Laravel menyediakan berbagai fitur keamanan seperti proteksi CSRF (Cross-Site Request Forgery) dan hashing password yang membantu melindungi aplikasi dari ancaman keamanan.
- **Artisan CLI:** Laravel menyediakan tool command-line bernama Artisan yang membantu dalam otomatisasi tugas-tugas rutin seperti migrasi database dan seeding.

Dengan semua kelebihan ini, Laravel menjadi pilihan yang kuat untuk membangun aplikasi web yang kompleks dan membutuhkan waktu pengembangan yang efisien.

3. Pembahasan

3.1 Instalasi dan Konfigurasi Laravel

Langkah-langkah Instalasi Laravel

1. **Persiapan Lingkungan Pengembangan:**
   - Pastikan komputer telah terinstal PHP versi terbaru (minimal PHP 7.3).
   - Instalasi Composer, dependency manager untuk PHP, diperlukan untuk mengelola paket dan dependensi dalam proyek Laravel.
2. **Menginstal Composer:**
   - Unduh dan instal Composer dari https://getcomposer.org/.
   - Setelah instalasi, pastikan Composer dapat diakses dari command line dengan menjalankan perintah

```
composer create-project --prefer-dist laravel/laravel nama_proyek
```

3. **Membuat Proyek Laravel Baru:**
   - Jalankan perintah berikut di command line untuk membuat proyek Laravel baru:

```
composer create-project --prefer-dist laravel/laravel nama_proyek
```

4. **Menjalankan Server Laravel:**
   - Setelah proyek berhasil dibuat, navigasikan ke direktori proyek dengan perintah:

```
cd nama_proyek
```

   - Jalankan server pengembangan Laravel dengan perintah:

```
php artisan serve
```

   - Akses aplikasi Laravel melalui browser di alamat `http://localhost:8000`.
5. **Konfigurasi Database:**
   - Buka file `.env` di direktori proyek dan konfigurasi pengaturan database sesuai dengan informasi koneksi database yang dimiliki, contohi:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=nama_database
DB_USERNAME=username_database
DB_PASSWORD=password_database
```

Referensi untuk langkah-langkah instalasi ini dapat ditemukan di dokumentasi resmi Laravel:

https://laravel.com/docs/11.x/installation

3.2 Pengembangan Fitur CMS

Memasuki pembahasan tentang pengembangan fitur, pada sesi ini akan difokuskan dengan penerapan aplikasi yang kita kembangkan yaitu daftar menu (menu) dan transaksi (order). Pembahasan difokuskan pada Schema (Migration), Model, View (Blade), dan Controller.

Dengan menggunakan schema (migration), model, view (Blade), dan controller secara bersama-sama, Laravel menyediakan pendekatan pengembangan yang terstruktur dan terorganisir untuk membangun aplikasi web yang skalabel dan mudah diatur.

3.2.1. Pembuatan model kelas

Seperti yang dijelaskan pada point 3.1, kita harus membuat tabel terlebih dahulu, laravel menyediakan perintah untuk membuat kelas skema database, model, dan controller pada perintah artisan.

```
php artisan make:model nama_model --controller --resource --requests
```

Pada aplikasi kasir ini, kita membutuhkan 2 tabel utama yaitu menu dan order, serta satu tabel untuk menghubungkan menu dan order. Maka kita perlu membuat 2 model seperti di atas dan 1 skema untuk tabel penghubung..
Link referensi :
https://laravel.com/docs/11.x/migrations#generating-migrations
https://laravel.com/docs/11.x/eloquent#generating-model-classes

3.2.2. Menentukan skema tabel

Schema dalam Laravel direpresentasikan melalui migration. Migration digunakan untuk mengelola struktur database secara terorganisir. Dengan migration, pengembang dapat membuat tabel, menambahkan kolom, dan memodifikasi skema database lainnya menggunakan sintaks PHP yang ekspresif. Migration memungkinkan perubahan skema database dijalankan secara konsisten di berbagai lingkungan pengembangan dan produksi.

Kita perlu menentukan skema dari tabel yang sudah terbuat, pada folder **/database/migrations/** akan terbentuk 3 file skema yang terbuat dari langkah 3.2.1 . Kita perlu merubah skema pada file tersebut.
Berikut adalah contoh skema pada *menu* :

```php
public function up(): void
{
    Schema::create('menus', function (Blueprint $table) {
        $table->id();
        $table->timestamps();
        $table->string("name");
```

```
            $table->integer("price");
            $table->string("type");
            $table->longText("description");
            $table->string("image");
        });
    }
```

Setelah semua skema tabel ditentukan, jalankan perintah migrasi dengan:

```
php artisan migrate
```

3.2.3 Mengatur Struktur Model

Model dalam Laravel adalah representasi dari tabel database. Model menghubungkan aplikasi dengan database dan memungkinkan pengembang untuk berinteraksi dengan data menggunakan PHP. Model Laravel seringkali terkait langsung dengan tabel database tertentu, dan menyediakan metode untuk melakukan operasi seperti menyimpan data, mengambil data, dan melakukan operasi database lainnya.

Berikut adalah contoh Model untuk *menu* pada aplikasi kasir.

```
class Menu extends Model
{
    protected $fillable = [
        'name',
        'price',
        'type',
        'description',
        'image'
    ];
    use HasFactory;
    public function orders()
    {
        return $this->belongsToMany(Order::class, 'order_menus')
            ->withPivot('quantity', 'price', 'note');
    }
}
```

3.2.4 Membuat tampilan View (blade)

View dalam Laravel menggunakan Blade sebagai template engine. Blade memungkinkan pengembang untuk membuat template HTML dengan struktur PHP yang bersih dan ekspresif. Blade menyediakan fitur seperti inheritance layout, sections, partials, looping, dan conditional statements, yang membuat proses pembuatan tampilan lebih efisien dan mudah dipelajari. Blade juga mendukung caching, yang membantu meningkatkan performa aplikasi dengan menyimpan hasil kompilasi template.

Berikut adalah contoh view (blade) layout tampilan utama pada aplikasi kasir :

```
<!DOCTYPE html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">


  <!-- Bootstrap CSS -->
                                                                                <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"

integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASj
C" crossorigin="anonymous">
                                                                              <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js
"

integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVX
M"
    crossorigin="anonymous"></script>
  <title>{{ $title }}</title>

</head>
<body>
  <div id="navbar">
    @include('layouts.sidebar')
  </div>
  @yield('content')

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"

integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5Kk
N"
    crossorigin="anonymous"></script>
                                                                            <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"

integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4
Q"
    crossorigin="anonymous"></script>
                                                                            <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"

integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY
l"
    crossorigin="anonymous"></script>
</body>

</html>
```

3.2.5 Menentukan pengaturan pada controller

Controller dalam Laravel bertanggung jawab untuk mengatur aliran aplikasi dan mengelola permintaan HTTP. Controller berisi metode-metode yang memproses permintaan yang diterima dari routing dan memanipulasi data sebelum menampilkan

hasilnya ke dalam view atau memberikan respons HTTP lainnya. Controller memisahkan logika aplikasi dari tampilan, mempromosikan reusabilitas kode, dan membuat aplikasi lebih mudah untuk dipelihara dan dikembangkan.

Berikut adalah contoh kode pada controller MenuController :

```
class MenuController extends Controller
{

    public function index()
    {
        $list_menu = Menu::all();
                                                                    return
view('menu.index',['title'=>'menu','list_menu'=>$list_menu]);

    }
    public function create()
    {
        return view('menu.create',['title'=>'Create enu']);
    }
    public function store(Request $request): RedirectResponse
    {
        $request->validate([
                                        'image'                                 =>
'required|image|mimes:jpeg,jpg,png|max:2048',
            'name'          => 'required|min:3',
            'description'   => 'required|min:10',
            'price'         => 'required|numeric',
            'type'         => 'required|min:3'
        ]);
        $image = $request->file('image');
        $image_name = $image->hashName();
        $image->storeAs('menu',$image_name );

        Menu::create([
            'image'         => $image_name,
            'name'          => $request->name,
            'description'   => $request->description,
            'price'         => $request->price,
            'type'          => $request->type
        ]);

           return  redirect()->route('menu.index')->with(['success'=>'Data
berhasil disimpan!']);
    }

}
```

3.3 Pengujian Sistem

Pengujian Fitur Menu (Daftar Menu)

1. **Pengujian Create Menu:**
    - Pengujian ini bertujuan untuk memastikan bahwa pengguna dapat menambahkan menu baru ke dalam sistem dengan benar.

    - Langkah-langkah pengujian meliputi:
        - Memasuki halaman tambah menu.
        - Mengisi formulir dengan informasi yang sesuai (nama menu, harga, kategori, dll.).
        - Memvalidasi bahwa data yang dimasukkan sesuai dengan format yang diharapkan.
        - Menyimpan menu baru dan memverifikasi bahwa data telah tersimpan dengan benar di database.
        - Mengonfirmasi bahwa menu baru dapat ditampilkan secara akurat di daftar menu.
2. **Pengujian Index Menu:**
    - Pengujian ini mengevaluasi kemampuan sistem dalam menampilkan daftar menu yang ada kepada pengguna.
    - Langkah-langkah pengujian meliputi:
        - Memastikan bahwa halaman daftar menu menampilkan semua menu yang tersedia.
        - Memverifikasi bahwa setiap entri menu menampilkan informasi yang relevan seperti nama, harga, dan kategori.
        - Menyaring dan mencari menu berdasarkan kriteria tertentu (misalnya, berdasarkan kategori atau nama menu).

Pengujian Fitur Order (Transaksi)

1. **Pengujian Create Order:**
    - Pengujian ini menguji proses pembuatan order baru oleh pengguna untuk memastikan transaksi dapat dilakukan dengan sukses.
    - Langkah-langkah pengujian meliputi:
        - Memasuki halaman order baru.
        - Memilih menu yang diinginkan dari daftar menu yang tersedia.
        - Menambahkan jumlah item yang dipesan.
2. **Pengujian Index Order:**
    - Pengujian ini mengevaluasi kemampuan sistem dalam menampilkan daftar order yang telah dibuat kepada pengguna.
    - Langkah-langkah pengujian meliputi:
        - Memastikan bahwa halaman daftar order menampilkan semua order yang telah dibuat.
        - Memverifikasi bahwa setiap entri order menampilkan informasi seperti nomor meja, nama pelanggan, dan total harga.

4. Hasil dan Pembahasan

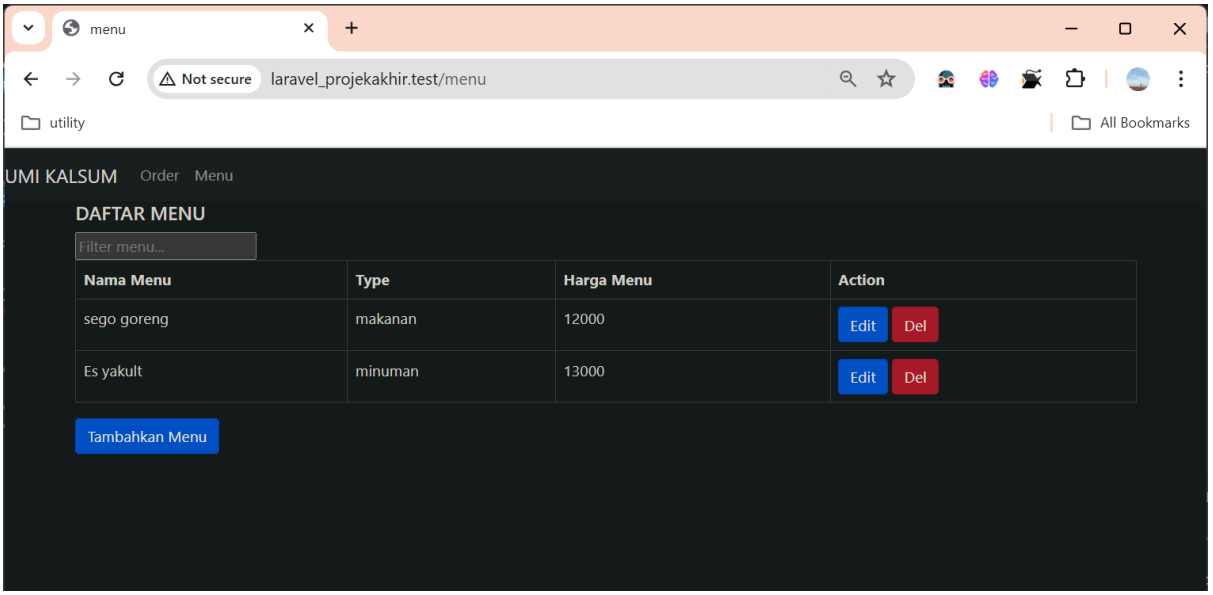
4.1 Hasil Implementasi

Source Code dapat diakses pada tautan berikut :
https://github.com/Umi-kalsum/pemrograman_web_tugas_4.git

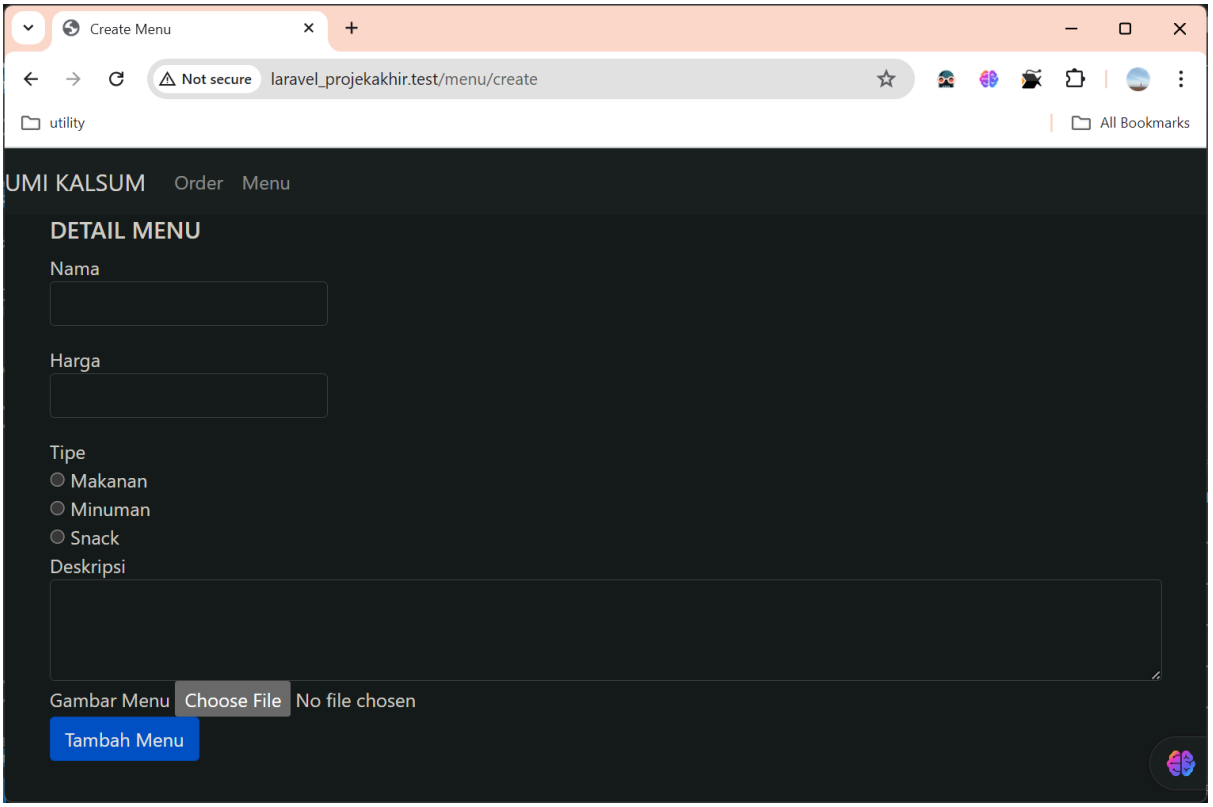
Video presentasi dapat diakses pada tautan berikut : https://youtu.be/rx83humMt_4

Berikut adalah tampilan antarmuka pengguna (UI) dan fungsionalitas yang tersedia.

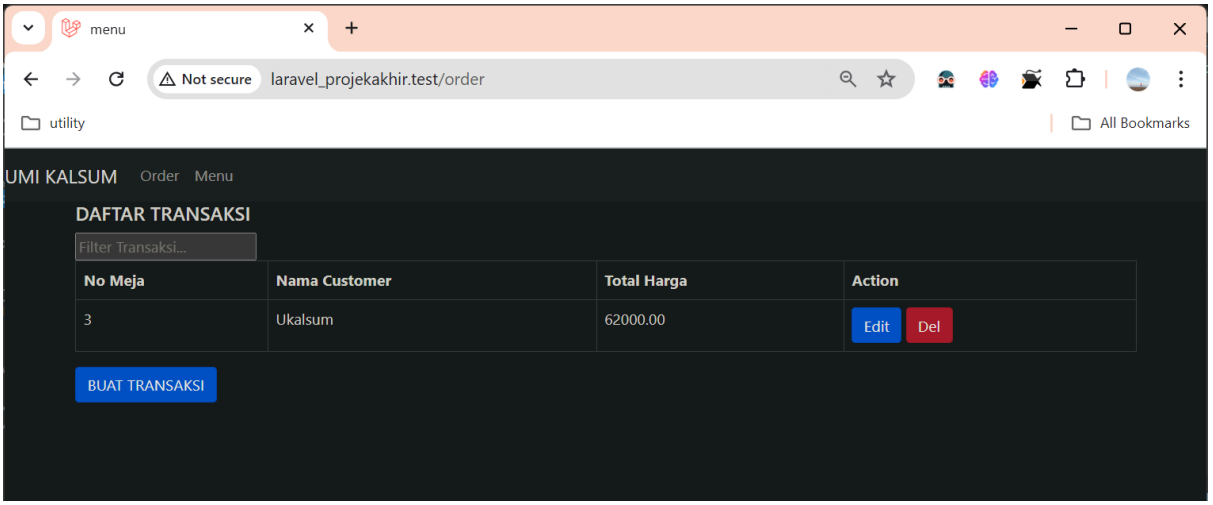
Tampilan antarmuka Daftar Menu



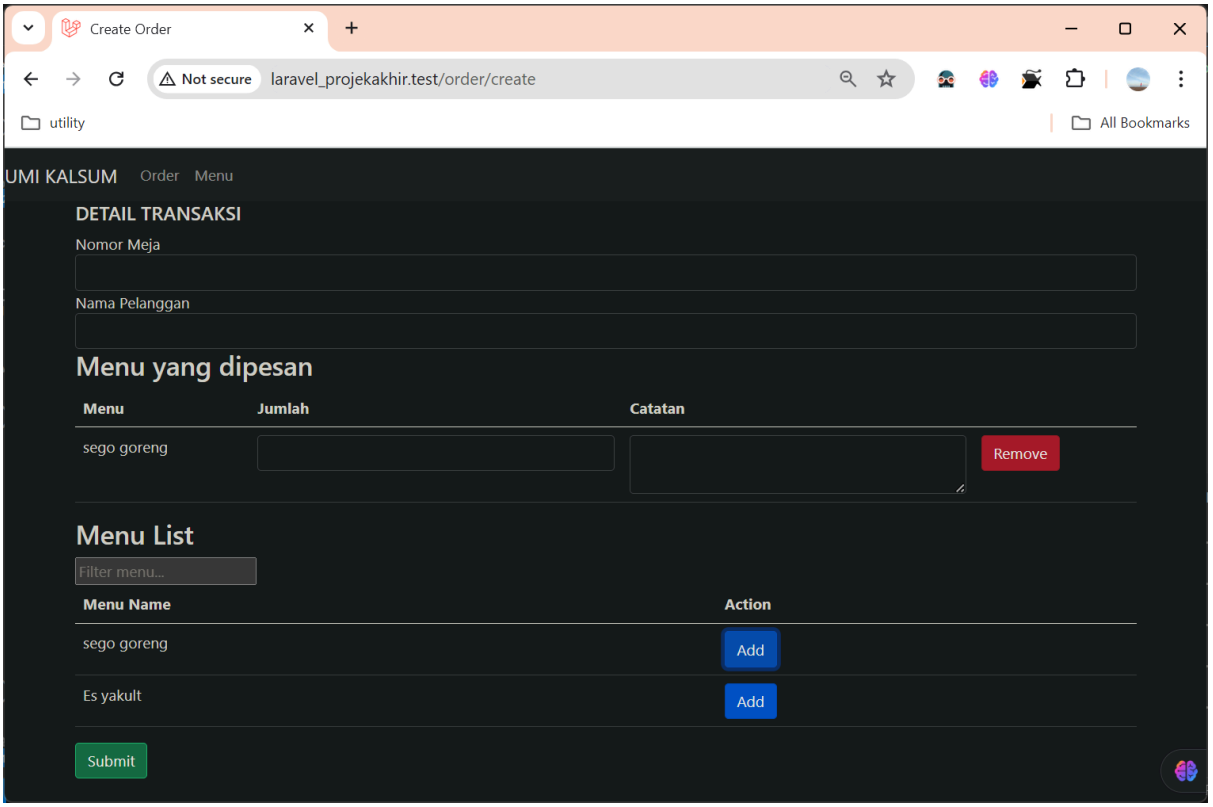
Tampilan antarmuka Tambah Menu



Tampilan antarmuka Daftar Transaksi



Tampilan antarmuka Buat Transaksi



4.2 Analisis Hasil

Kelebihan Aplikasi

- **Aplikasi dapat diakses:**
  - Aplikasi dapat diakses dari berbagai perangkat dengan koneksi internet, memungkinkan akses yang mudah untuk pengguna di lokasi yang berbeda.
- **Dapat membuat daftar menu:**
  - Pengguna dapat menambahkan menu baru ke dalam sistem dengan mudah melalui fitur create menu, memungkinkan untuk pengelolaan menu yang fleksibel.
- **Dapat melihat daftar menu:**

    - Pengguna dapat melihat semua daftar menu yang tersedia di aplikasi, mempermudah pengelolaan dan peninjauan menu yang ada.
- **Dapat melihat semua daftar menu ketika proses membuat transaksi:**
    - Saat membuat transaksi, pengguna dapat melihat semua daftar menu yang tersedia, memudahkan dalam pemilihan menu yang ingin dipesan oleh pelanggan.
- **Dapat membuat transaksi:**
    - Pengguna dapat membuat transaksi dengan menginputkan pesanan dari daftar menu yang tersedia, memfasilitasi proses penjualan secara efisien.
- **Dapat melihat transaksi yang sudah dibuat:**
    - Pengguna dapat melihat riwayat transaksi yang telah dibuat, termasuk detail pesanan dan informasi transaksi lainnya, membantu dalam pelacakan dan pencatatan penjualan.

Kekurangan Aplikasi

- **Belum mengimplementasi edit dan delete data pada menu:**
    - Saat ini, aplikasi belum mendukung fitur untuk mengedit atau menghapus data menu yang sudah ada. Hal ini dapat menghambat fleksibilitas pengelolaan menu dalam jangka panjang.
- **Belum mengimplementasi edit dan delete data pada transaksi:**
    - Fitur untuk mengedit atau menghapus data transaksi yang sudah dibuat belum diimplementasikan. Ini dapat menyulitkan dalam memperbaiki atau menghapus data yang salah atau tidak diperlukan.
- **Belum menampilkan konfirmasi sebelum membuat pesanan:**
    - Aplikasi saat ini tidak menampilkan konfirmasi kepada pengguna sebelum transaksi atau pesanan dibuat. Fitur ini dapat membantu mengurangi kesalahan input dan memastikan kepuasan pelanggan.

Penilaian kelebihan dan kekurangan ini penting untuk memahami kemampuan dan keterbatasan aplikasi kasir yang telah dikembangkan, serta memberikan arah untuk pengembangan selanjutnya guna meningkatkan fungsionalitas dan pengalaman pengguna secara keseluruhan.

5. Daftar Pustaka

Daftar referensi yang digunakan dalam penulisan makalah ini.

- Laravel Documentation. (n.d.). Laravel Installation. Retrieved from https://laravel.com/docs/11.x/installation
- Laravel Documentation. (n.d.). Migrations - Laravel. Retrieved from https://laravel.com/docs/11.x/migrations#generating-migrations
- Laravel Documentation. (n.d.). Eloquent ORM - Laravel. Retrieved from https://laravel.com/docs/11.x/eloquent#generating-model-classes
- Laravel Documentation. (n.d.). Controllers - Laravel. Retrieved from https://laravel.com/docs/11.x/controllers#writing-controllers
- Laravel Documentation. (n.d.). Blade Templates - Laravel. Retrieved from https://laravel.com/docs/11.x/blade#main-content