

# Rancangan Aplikasi E-Payment Restoran Sop Duren 97 berbasis Web dengan Implementasi Algoritma FCFS menggunakan React JS dan PostgreSQL

Muhammad Ihsan<sup>1</sup>, Agung Slamet Riyadi<sup>1</sup>, Ire Puspa Wardhani<sup>2\*</sup>

<sup>1</sup>Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma

<sup>2</sup>Magister Teknologi Informasi, STMIK Jakarta STI&K

\*[irepuspa@gmail.com](mailto:irepuspa@gmail.com)

**Abstrak**— Perkembangan teknologi informasi telah mengubah cara berinteraksi dan bertransaksi dalam bisnis, termasuk melalui platform online atau e-commerce. Banyak restoran masih menerapkan sistem pelayanan konvensional, yang dapat menimbulkan kendala saat pelanggan memesan makanan secara langsung. Hal ini berpotensi membuat pelayan menjadi kelelahan dan menghadapi kesulitan dalam pelayanan. Sebagai solusi, banyak bisnis restoran mengadopsi penggunaan QR Code untuk memesan makanan dan memfasilitasi pembayaran. Dalam konteks ini, penelitian ini bertujuan untuk mengimplementasikan aplikasi web E-Duren di Restoran Sop Duren 97 dengan mengintegrasikan teknologi payment gateway midtrans untuk memproses pembayaran. Sistem pesanan akan diatur menggunakan algoritma First Come First Serve (FCFS) guna meningkatkan efisiensi pelayanan dan memodernisasi pelayanan di restoran yang masih menggunakan metode konvensional. Hasil pengujian dari User Acceptance Test (UAT) dengan sampel 30 orang menunjukkan indeks persentase sebesar 83,1%, dengan kesimpulan bahwa aplikasi website E-Duren sudah berjalan lancar dan sesuai dengan kepuasan pengguna. Ujicoba aplikasi dengan *blackbox testing* untuk menguji fungsionalitas aplikasi untuk menunjukkan bahwa semua fungsi aplikasi telah berjalan lancar.

## Article History:

Received: Dec 22, 2023

Revised: Jan 15, 2024

Accepted: Jan 19, 2024

Published: Feb 6, 2024

**Kata Kunci**— FCFS, Midtrans, Order, Pembayaran, Pemesanan, Restoran.

DOI: 10.22441/jitkom.v8i1.002

## I. PENDAHULUAN

Pemanfaatan teknologi informasi dalam dunia bisnis memberikan dampak yang besar terhadap proses bisnis agar dapat berjalan secara efektif dan efisien. Salah satu dampak positifnya adalah pemanfaatan sarana internet sebagai media promosi, aktivitas transaksi jual-beli dan proses memperkenalkan produk dapat dilakukan secara online, mengurangi biaya promosi, serta menjangkau pelanggan dari berbagai lokasi dengan lebih luas dan cepat. Perubahan kebiasaan akibat perkembangan teknologi mendorong semakin populer dan luasnya penjualan melalui platform online atau disebut sebagai e-commerce.

Selain itu, perubahan lain yang cukup signifikan adalah penggunaan teknologi QR Code pada bisnis kuliner ini. QR Code menjadi salah satu alat yang efektif dalam mempermudah berbagai aktivitas pelanggan, terutama dalam hal transaksi dan berinteraksi pada layanan bisnis kuliner. [1] Penggunaan QR Code ini memungkinkan pelanggan mudah melihat menu makanan, melakukan pembayaran, dan melakukan tindakan lainnya dengan cara yang lebih efisien. [2] Penelitian ini merupakan penelitian terhadap tamu atau pengunjung penjualan yang dapat melihat menu secara langsung dari perangkat smartphone pengguna dalam bentuk halaman web

yang dapat diakses melalui scan QR Code yang akan diterapkan pada setiap meja pada restoran bernama Sop Duren 97 yang berlokasi di Ruko Permata Cibubur, Blok C5, Cileungsi Kidul, Kecamatan Cileungsi, Kabupaten Bogor. Selama ini Restoran Sop Duren 97 menjual berbagai menu makanan, minuman dan juga makanan olahan dari bahan buah durian ini terkendala dan terkait pada lamanya waktu pelayanan pesanan yang menimbulkan ketidakpuasan pelanggan apabila tidak dilayani secara cepat. [3] Kendala ini disebabkan oleh peningkatan jumlah pelanggan dan minimnya jumlah karyawan, membutuhkan waktu lebih lama proses pada pemesanan. Penelitian ini menghasilkan aplikasi berbasis web dengan nama *E-Duren*, sebuah aplikasi yang dibuat dengan tujuan melakukan modernisasi pelayanan restoran yang dilengkapi dengan berbagai macam fitur. Salah satu fiturnya adalah fitur pemesanan makanan dengan metode pembayaran melalui QR Code, yang bisa dibayar secara langsung melalui aplikasi dengan sistem transaksi payment gateway via midtrans atau membayar di kasir. Selain itu pada aplikasi ini dibuat secara otomatis history pemesanan pada admin yang merupakan data Riwayat pemesanan pada restoran tersebut. [4]

Web E-Duren menggunakan salah satu algoritma antrian First Come First Serve (FCFS) untuk membuat efisiensi waktu terhadap proses pemesanan hingga pembayaran yang dimana

akan disusun dan diurutkan berdasarkan pesanan prioritas yang masuk pertama untuk diproses terlebih dahulu tanpa memperdulikan tingkat kepentingan. [5]

## II. LITERATURE REVIEW

Pada penelitian yang telah dilakukan oleh Eldies, 2021 memberikan contoh tentang bagaimana algoritma FCFS digunakan pada aplikasi pemesanan dan pembayaran pada restoran Sari Idaman yang sangat membantu untuk mempercepat proses pencatatan pesanan secara manual ketika restoran sedang dalam kondisi ramai yang bisa mencapai 120 nota transaksi pada hari biasa dan dua kali lipat ketika hari libur [1].

Pada penelitian sebelumnya yang dilakukan oleh Anam & Wahanggara, 2018 memberikan contoh bagaimana pengimplementasian algoritma FCFS yang menggunakan perhitungan rumus *time around* untuk dilakukan untuk menguji waktu yang dibutuhkan untuk melakukan proses pemesanan. Hasilnya adalah ketika pesanan masuk secara bersamaan dari 3 orang dengan waktu tunggu adalah 12.33 menit sedangkan untuk pesanan di waktu yang acak dari 5 orang menghasilkan waktu tunggu selama 42.4 menit [2].

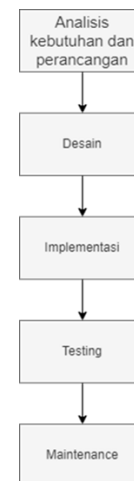
Penggunaan algoritma FCFS juga dapat digabungkan dengan beberapa metode lain seperti pada penelitian yang dilakukan oleh Syofian & Damar, 2020 yang mengaplikasikan metode Haversine dan FCFS secara bersamaan untuk mendapatkan dan mengelola jarak tempuh dan urutan pelayanan pemesanan makanan pelanggan pada toko TIXZ KITCHEN [3].

Selain mengimplementasikan algoritma FCFS, aplikasi web E-Duren juga melakukan implementasi payment gateway midtrans untuk membantu dari segi pembayaran yang menurut penelitian dari Puspasari & Maulina, 2019 yang menjelaskan bagaimana Midtrans berperan dalam memfasilitasi pengguna dalam melakukan pembayaran online dengan aman dan memberikan kemampuan bagi admin dan vendor untuk memantau status pemesanan online yang dilakukan oleh pelanggan [4].

Selain itu juga terdapat penelitian yang dilakukan oleh Habibirrahman, Putra, & Hanggara, 2022 menunjukkan bahwa penerapan payment gateway midtrans pada toko kue De Tasty mendapatkan penilaian sebesar 74,13 berdasarkan berdasarkan survey kuesioner terhadap 20 responden menunjukkan bahwa aplikasi web dengan metode pembayaran yang diimplementasikan oleh midtrans sudah mendapat grades acceptable dan dapat digunakan untuk melakukan pembayaran [5].

## III. METODOLOGI PENELITIAN

Metode yang digunakan dalam pengembangan aplikasi E-Duren berbasis website ini [6] adalah dengan menggunakan SDLC atau (Software Development Life Cycle) dengan model Waterfall. Proses pembuatannya mengikuti urutan yang dimulai analisis, desain, kode, pengujian dan pemeliharaan. Dalam penggunaan model dari Waterfall memiliki tahap-tahap yang digunakan. Seperti: Analisis Kebutuhan (Requirement) terlihat pada gambar skema penelitian berikut ini :



Gambar 1. Skema Penelitian

Pada Tahap awal penelitian ini, penulis memulai dengan melakukan pengumpulan sumber data dan analisis yang bertujuan untuk memahami apa yang diharapkan oleh pengguna dan informasi yang berhubungan dengan pembuatan sebuah halaman web untuk dapat lebih memudahkan penulis dalam perancangan halaman web. [7] Tahap Desain, Pada tahap kedua, akan dilakukan sebuah perancangan desain perangkat lunak yang mencakup struktur data, arsitektur perangkat lunak, user interface, serta prosedur pengkodean untuk mempermudah proses tahap implementasi. Desain perangkat lunak ini akan menggunakan teknik UML (*Unified Modeling Language*) termasuk Use Case Diagram untuk menggambarkan interaksi antara sistem dan aktor, serta *Class Diagram* untuk merepresentasikan struktur dan hubungan antar objek dalam sistem, *Activity Diagram* digunakan untuk menggambarkan aliran proses bisnis dan interaksi antara aktivitas dalam system. Tahap Implementasi, pada tahap ini, penulis memulai pembuatan program yang akan disesuaikan dengan desain pada tahap sebelumnya. Implementasi aplikasi E-Duren akan dilakukan dengan menggunakan teknologi framework dari Bahasa pemrograman *JavaScript* yaitu React JS [8] untuk membangun antarmuka pengguna (user interface), Express JS untuk mengelola server dan routing, Firebase untuk keperluan backend, serta PostgreSQL sebagai database untuk menyimpan data aplikasi. Selain itu, dalam proses penulisan kode, penulis akan menggunakan Text editor Visual Studio Code, yang merupakan salah satu text editor populer dan mendukung banyak fitur untuk membantu dalam pengembangan perangkat lunak. Peneliti menggunakan perangkat keras berupa Laptop Lenovo ideaPad 130-14IKB dengan spesifikasi perangkat keras dan perangkat lunak yang mencakup:

- Processor : Intel Core i3-7020U 2.30GHz
- Memori : 12GB DDR4
- VGA : Intel HD Graphics 620
- Storage : SSD 512 GB, HDD 1TB
- OS : Windows 10 Home
- Text editor : Visual Studio Code
- Database : PostgreSQL dengan integrasi DBeaver
- Browser : Google Chrome dan Opera GX

Tahap Integrasi dan Testing, Pada tahap Integrasi dan Testing ini dilakukan setelah selesai pembuatan modul, modul-modul tersebut akan digabungkan untuk membentuk aplikasi E-Duren. Selanjutnya, akan dilakukan pengujian untuk memastikan perangkat lunak yang telah dikembangkan sesuai dengan fungsi dan desain yang telah ditentukan. [9] Tahap pengujian aplikasi, terdapat beberapa pengujian yang akan dilakukan, pertama adalah menggunakan metode Blackbox Testing dengan tipe Functional Testing, di mana aplikasi akan diuji berdasarkan fungsionalitas yang telah dijelaskan. Selain itu, akan digunakan Teknik All pair testing untuk memastikan pengujian berbagai kombinasi fitur dan situasi yang mungkin terjadi pada aplikasi. Lalu terdapat juga pengujian algoritma untuk menguji penerapan algoritma FCFS dalam implementasi di web E-Duren. Dan terakhir terdapat pengujian UAT (User Acceptance Test) menggunakan metode skala likert. Proses pengujian akan dilakukan melalui website browser Google Chrome, [10] sehingga aplikasi dapat diuji secara menyeluruh dengan berbagai skenario penggunaan. Tahap Evaluasi dan Maintenance, Ini merupakan tahap terakhir dalam model Waterfall yang dimana tahap ini berisi pemeliharaan perangkat lunak untuk memperbaiki kesalahan yang ada setelah testing perangkat lunak.

Metode penelitian yang digunakan mengenai “Aplikasi Web E-Payment [11] Restoran Sop Duren 97 [12] Dengan Implementasi Algoritma FCFS [13][14][15] Menggunakan React Js Dan Postgresql” dalam penelitian ini dapat digambarkan melalui proses dari metode SDLC dengan model waterfall terlihat pada gambar-1, yaitu diawali dengan melakukan analisa kebutuhan dan perancangan, desain halaman, implementasi, testing, analisis evaluasi dan maintenance untuk dapat memelihara sistem dari website tersebut. [16][17]

Hal pertama yang dilakukan adalah diawali dengan melakukan analisa dari segi perangkat lunak dan perancangan sistem untuk kebutuhan dalam aplikasi yang dibuat, lalu tahap kedua yang dilakukan adalah membuat desain halaman dari *website* dengan sistem yang ingin dibuat seperti halaman awal, halaman *login*, halaman, dan halaman lainnya, kemudian tahap ketiga adalah pengimplementasian dari desain menjadi sebuah *website* dengan proses koding yang ditulis, lalu tahap keempat akan dilakukan testing menggunakan metode *blackbox* testing untuk melihat apakah semua komponen dalam *website* tersebut sudah berjalan dengan baik atau belum, kemudian akan dilakukan analisis evaluasi dan *maintenance* untuk dapat memelihara sistem dari *website* tersebut dengan melakukan *survey* dengan menggunakan skala likert untuk mendapatkan pendapat dan saran yang bisa digunakan untuk *maintenance* fitur yang dirasa masih kurang.

#### A. Analisa Kebutuhan dan Perancangan

Pada tahap ini, dilakukan pengumpulan data dan informasi dari berbagai sumber untuk memahami secara mendalam kebutuhan pengguna terhadap *website* E-Duren. Analisis ini bertujuan untuk mengidentifikasi fitur-fitur yang diperlukan guna mempercepat layanan restoran, termasuk proses pemesanan dan pembayaran, serta mengatasi masalah-masalah dalam pelayanan restoran.

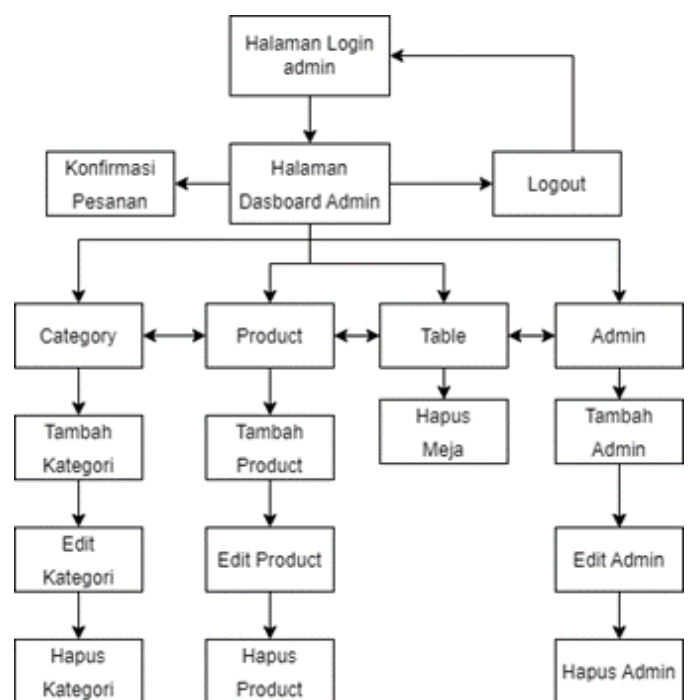
Berdasarkan data yang diperoleh, restoran Sop Duren 97 memerlukan modernisasi pelayanan dan pembayaran untuk meningkatkan efisiensi dan kecepatan layanan. Oleh karena itu, diperlukan pembuatan *website* E-Duren yang dapat diakses secara langsung oleh pelanggan. Fitur utama yang dibutuhkan dalam pembuatan *website* ini adalah pemesanan makanan yang terhubung langsung dengan admin, serta fleksibilitas dalam pembayaran, baik melalui sistem *payment gateway* midtrans maupun langsung di kasir. Selain itu, admin juga harus dapat mengelola data pelanggan dan mengubah menu, termasuk menambah, menghapus, dan mengeditnya.

#### B. Struktur Navigasi

Terdapat 2 struktur navigasi yang digunakan dalam perancangan *website* E-Duren, yaitu

##### 1. Struktur Navigasi Admin

Gambaran umum mengenai struktur navigasi admin dapat dilihat pada gambar 2 berikut ini.



Gambar 1. Struktur Navigasi Admin

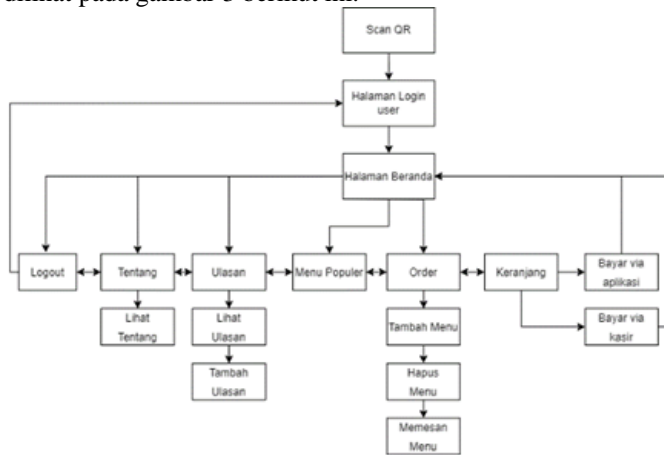
Dalam navigasi admin, proses dimulai dari halaman *login* admin, di mana administrator harus memasukkan *username* dan *password* untuk masuk ke halaman *Dashboard* admin. Halaman *Dashboard* admin memiliki enam menu, yaitu *Dashboard*, *Category*, *Product*, *Table*, *Admin*, dan *Logout*. Pada halaman *Dashboard* admin, terdapat beberapa proses, seperti konfirmasi pemesanan dengan mengubah data yang ada, dan juga proses *logout* untuk keluar dari halaman admin.

Selanjutnya, terdapat halaman *Category* yang berisikan daftar kategori menu yang dihidangkan di restoran Sop Duren 97. Halaman ini memiliki tiga proses, yaitu menambah, mengedit, dan menghapus kategori. Kemudian, terdapat halaman *Product* yang berisi daftar produk dari menu restoran. Halaman ini memiliki tiga proses, yaitu menambah produk, mengedit produk, dan menghapus produk. Selain itu, ada

halaman *Table* yang berisikan data customer yang tersimpan di *database* setiap kali *customer login*. Halaman ini memiliki proses hapus data. Terakhir, ada halaman Admin yang menampilkan daftar akun admin yang dapat melakukan proses tambah data, edit, dan hapus data di *website* E-Duren.

## 2. Struktur Navigasi User

Gambaran umum mengenai struktur navigasi admin dapat dilihat pada gambar 3 berikut ini.



Gambar 2. Struktur Navigasi User

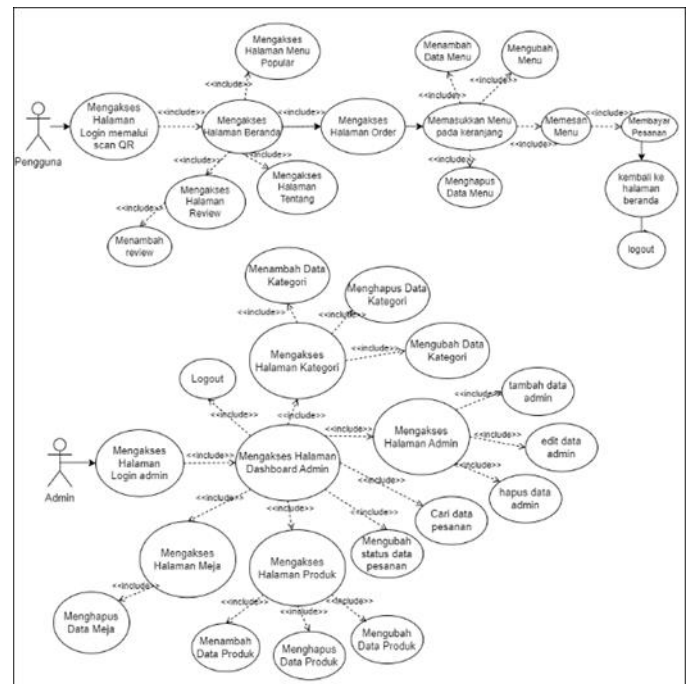
Dalam navigasi *user*, proses dimulai dari pengguna yang melakukan *scan QR* di meja restoran dan akan diarahkan ke halaman *login user*. Setelah itu, pengguna diminta untuk memasukkan nama serta nomor telepon, dan akan diarahkan ke halaman beranda. Pada halaman beranda, terdapat beberapa menu, yaitu Tentang, Ulasan/Review, Popular, Order, dan Logout.

Pada halaman Tentang, pengguna dapat melihat profil dari restoran Sop Duren 97. Selanjutnya, ada halaman Ulasan yang berisikan hasil ulasan yang ditambahkan oleh pengguna lain. Kemudian, ada halaman Menu Popular yang berisikan menu-menu yang paling sering dibeli atau direkomendasikan dari restoran Sop Duren 97. Terdapat tombol Logout untuk keluar dari halaman beranda menuju halaman *login user* kembali.

Selanjutnya, ada halaman order yang berisikan menu makanan atau minuman yang bisa dipesan oleh pengguna. Pengguna dapat melakukan proses tambah, edit, dan hapus menu, serta memesan menu. Setelah itu, ada halaman Keranjang di mana pengguna dapat memastikan apakah pesanan yang dipesan sudah sesuai atau belum sebelum membayarnya. Di halaman ini, terdapat 2 proses, yaitu bayar di kasir atau bayar langsung. Setelah selesai, pengguna akan dibawa kembali ke halaman beranda.

## C. Use Case Diagram

Use case digunakan untuk mengidentifikasi interaksi antara pengguna dan sistem, seperti yang terlihat pada gambar 4 berikut ini.



Gambar 3. Use Case Diagram

Pada Use Case Diagram pada gambar 4 akan terdapat 2 aktor, yaitu *user* dan *admin*. Proses untuk *user* dimulai dari mengakses halaman *login* melalui *scan QR*. Kemudian, *user* diarahkan ke halaman beranda yang berisi beberapa menu yang dapat dipilih, seperti halaman menu *popular*, halaman tentang, halaman *review*, dan halaman order.

Pada halaman *review*, *user* dapat memasukkan *review* atau ulasan mereka, yang nantinya akan ditampilkan di halaman *review*. Sementara itu, pada halaman order, *user* dapat memasukkan menu ke dalam keranjang dengan proses tambah, ubah, dan hapus data menu. Setelah itu, *user* dapat memesan menu yang sudah ada di dalam keranjang dan membayar pesanan. Setelah selesai, *user* akan diarahkan kembali ke halaman beranda dan dapat melakukan *logout*.

Proses untuk *admin* dimulai dengan mengakses halaman *login admin*. Jika data *login* benar, *admin* akan masuk ke *Dashboard admin*. Di halaman *Dashboard*, *admin* memiliki akses ke beberapa halaman, seperti halaman meja, halaman produk, halaman kategori, dan halaman admin.

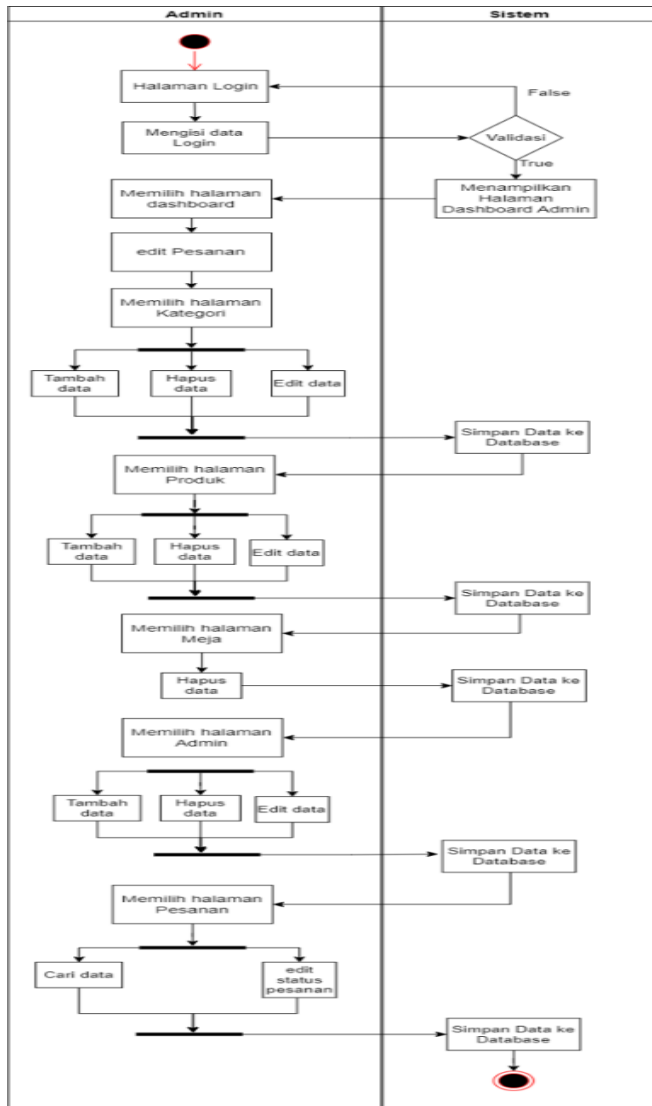
Di halaman meja, *admin* dapat menambah dan menghapus data meja. Pada halaman kategori, *admin* dapat menambah, menghapus, dan mengedit kategori. Di halaman produk, *admin* bisa menambah, menghapus, dan mengedit produk yang ada di *website* E-Duren. Selain itu, di halaman admin sendiri, terdapat 3 proses yang sama, yaitu menambah, mengedit, dan menghapus data admin.

## D. Activity Diagram

Activity diagram digunakan untuk menggambarkan alur atau urutan aktivitas dalam suatu proses atau sistem yang ada di *website* E-Duren. Activity diagram sendiri dibagi menjadi 2 bagian, yaitu activity diagram admin, dan activity diagram *user*.



1. Activity Diagram Admin



Gambar 4. Activity Diagram Admin

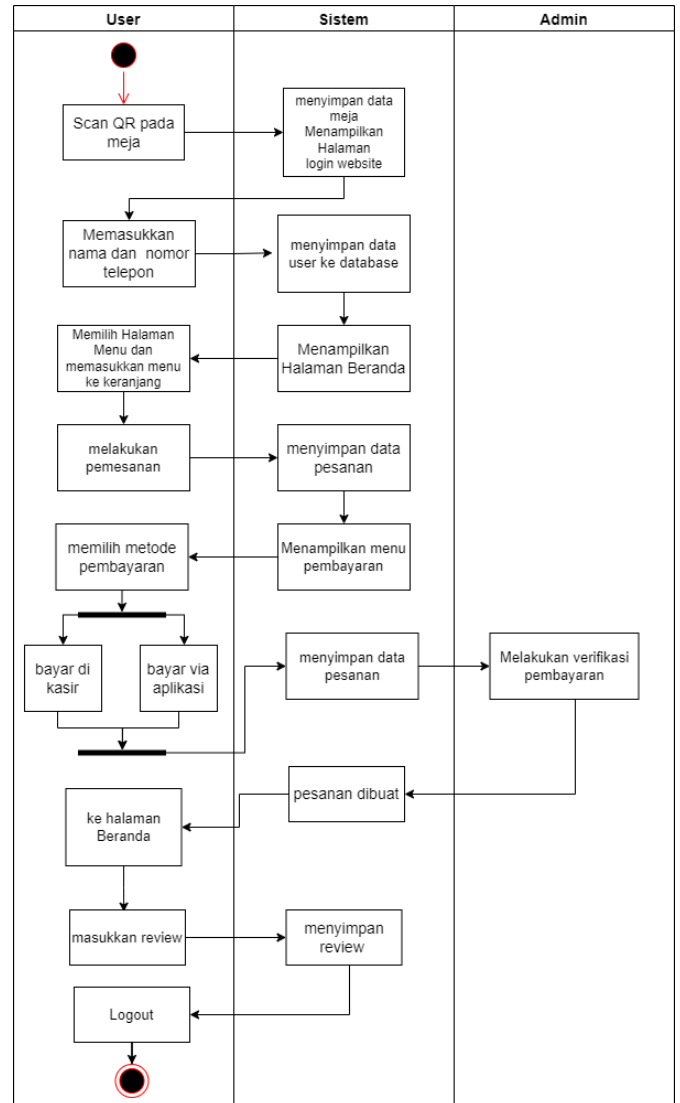
Pada Diagram aktivitas admin pada gambar 5 menunjukkan terdapat 2 kontainer yang berisi admin dan sistem yang mengatur jalannya proses dalam *website*. Langkah pertama adalah ketika admin menuju halaman *login* admin dan mengisi data *login*. Jika data yang dimasukkan salah, admin akan kembali ke halaman *login*. Namun, jika datanya benar, sistem akan menampilkan halaman *Dashboard* admin.

Pada halaman *Dashboard*, admin memiliki beberapa pilihan, seperti melakukan edit pesanan untuk pengguna yang belum membayar dan pengguna yang sudah membayar. Selanjutnya, admin dapat masuk ke halaman kategori di mana dia dapat melakukan proses tambah, edit, dan hapus kategori apa saja yang ada.

Data dari proses ini akan disimpan dalam *database* oleh sistem. Selanjutnya, admin dapat masuk ke halaman produk untuk menambah, menghapus, serta mengedit produk makanan dan minuman yang ada di *website* E-Duren. Setelah admin selesai melakukan proses tersebut, data akan dimasukkan ke dalam *database* oleh sistem. Admin juga

memiliki akses untuk melihat daftar tamu yang masuk ke *website* dengan memilih menu *table* dan melakukan proses hapus data tamu. Terakhir, admin dapat menambah data admin melalui halaman admin dan melakukan proses tambah, edit, serta hapus data admin, dan sistem akan menyimpan data yang diubah ke dalam *database*.

2. Activity Diagram User



Gambar 5. Activity Diagram User

Pada diagram aktivitas seperti gambar 6 diatas, langkah pertama dimulai ketika *user* melakukan *scan QR* yang ada di meja restoran. Selanjutnya, *user* akan diarahkan ke halaman *login user*, di mana *user* diminta untuk memasukkan nama dan nomor telepon. Setelah itu, *user* akan diarahkan ke halaman beranda, di mana terdapat beberapa menu yang bisa dipilih oleh *user*, seperti halaman tentang, ulasan/*review*, menu populer, order, dan juga *logout*.

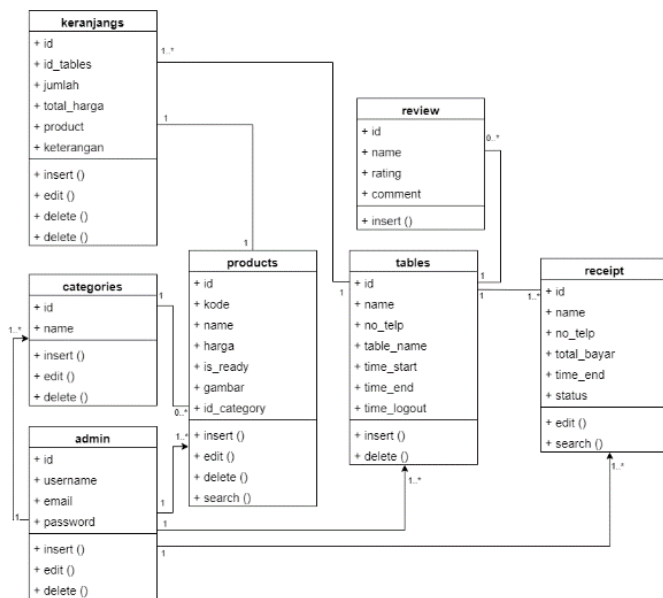
Pada halaman tentang, *user* dapat melihat profil dari restoran Sop Duren 97. Kemudian, *user* juga dapat mengakses halaman ulasan yang berisikan hasil ulasan yang ditambah oleh *user*. Selanjutnya, ada halaman menu populer yang berisikan menu yang paling sering dibeli atau

direkomendasikan dari restoran Sop Duren 97. *User* juga dapat kembali ke halaman *login user* dengan memilih tombol *logout*.

Selanjutnya, ada halaman order yang berisikan menu makanan atau minuman yang dapat dipesan oleh *user*. *User* dapat melakukan penambahan, mengedit, dan menghapus menu, serta memesan menu. Setelah memesan menu, *user* akan dibawa kembali ke halaman beranda.

### E. Class Diagram

*Class diagram* digunakan untuk menggambarkan struktur serta deskripsi dari *class*, atribut, metode, dan hubungan antara setiap objek. Gambar 7 menampilkan *class diagram* ini.



Gambar 6. Class Diagram

Pada *class diagram* yang dibuat, terdapat 7 *class* yang memiliki masing-masing tujuan, yaitu:

#### 1. Admin:

*Admin*: Kelas *admin* memiliki 4 atribut, yaitu *id*, *username*, *email*, serta *password* untuk melakukan login. Kelas *admin* juga memiliki 3 operasi, yaitu *insert*, *edit*, dan *delete*. *Admin* bertanggung jawab untuk mengelola data admin dalam sistem. Kelas *admin* memiliki relasi dengan tabel *categories*, *products*, *tables*, serta *receipts* yang dimana bernilai *one-to-many optional*. Dengan data 1 *admin* dapat mengelola banyak data kategori, produk, tabel, dan *receipt*. Sedangkan 1 data data kategori, produk, tabel, dan *receipt* hanya dapat dikelola oleh *admin*.

#### 2. Categories:

Kelas *categories* memiliki 2 atribut, *id* dan *nama*. Kelas *categories* memiliki 3 operasi, yaitu *insert*, *edit*, dan *delete*. *Categories* berperan dalam mengelola informasi tentang kategori makanan dan minuman yang ada di restoran. dalam segi relasinya sendiri. Kelas *categories* memiliki relasi dengan tabel *products* yang dimana bernilai *one-to-many optional*. Dengan data 1 kategori dapat dimiliki oleh banyak produk sedangkan 1 produk hanya memiliki 1 kategori.

#### 3. Products:

Kelas *products* memiliki 7 atribut, yaitu *id*, *kode*, *name*, *harga*, *is\_ready*, *gambar*, dan *id\_category*. Kelas *products* juga memiliki 4 operasi, yaitu *insert*, *edit*, *delete*, dan *search*. *Products* bertugas untuk mengelola data produk makanan dan minuman yang dapat dipesan oleh pengguna. Kelas *products* memiliki relasi dengan tabel *categories* dan tabel keranjangs, yang dimana untuk relasi dengan tabel keranjangs adalah *one-to-one*, dengan arti 1 produk hanya dimiliki 1 keranjang dan 1 keranjang hanya dapat memiliki 1 produk.

#### 4. Tables:

Kelas *tables* memiliki 6 atribut, yaitu *id*, *name*, *no\_telp*, *table\_name*, *time\_start*, *time\_end*, *time\_logout*. Kelas *tables* memiliki 2 operasi, yaitu *insert* dan *delete*. *Tables* berfungsi untuk mengelola informasi tentang meja-meja yang tersedia di restoran. kelas *tables* memiliki relasi dengan tabel keranjangs, tabel *review*, tabel *receipt*. Untuk relasi dengan tabel *review* memiliki relasi *one-to-mandatory many* dengan arti 1 table dapat memiliki atau tidak memiliki *review* dan 1 *review* dimiliki 1 table, sedangkan *review* hanya dimiliki oleh 1 table. Untuk relasi dengan tabel keranjangs yang dimana memiliki relasi *one-to-many mandatory* dengan arti 1 table dapat memiliki 1 atau lebih keranjangs dan 1 keranjangs hanya dimiliki 1 table. Untuk relasi dengan tabel *receipt* memiliki relasi *one-to-many mandatory* dengan arti 1 table dapat memiliki lebih dari 1 *receipt* dan 1 *receipt* hanya dimiliki 1 table.

#### 5. Receipt:

Kelas *receipt* memiliki 6 atribut, yaitu *id*, *name*, *no\_telp*, *total\_bayar*, *time\_start*, *status*. Kelas *receipt* memiliki 2 operasi, yaitu *search* dan *edit*. *Receipt* bertugas untuk mengelola data pemesanan dan pembayaran yang dilakukan oleh pengguna. Untuk relasi tabel *receipt* memiliki relasi dengan tabel *tables*, dengan relasi *one-to-many mandatory*.

#### 6. Keranjangs:

Kelas keranjangs memiliki 6 atribut, yaitu *id*, *id\_tables*, *jumlah*, *total\_harga*, *products*, *keterangan*. Kelas keranjangs juga memiliki 3 operasi, yaitu *insert*, *edit*, serta *delete*. Keranjangs bertanggung jawab untuk mengelola keranjang belanja pengguna sebelum melakukan pemesanan. untuk relasi tabel keranjangs memiliki relasi dengan tabel *products* dengan relasi *one-to-one* dan tabel *tables* dengan *many mandatory- to-one*.

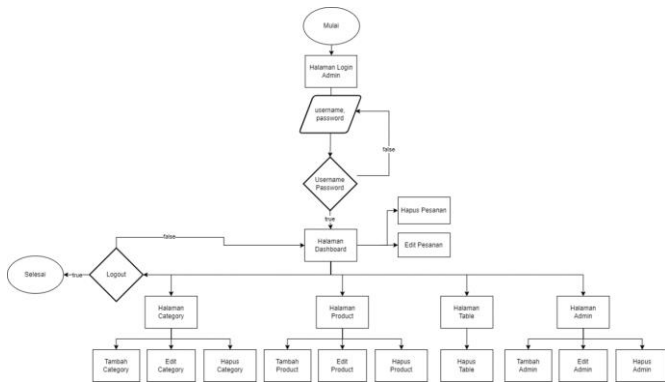
#### 7. Review:

Kelas *Review* memiliki 4 atribut, yaitu *id*, *name*, *rating*, serta *comment*. Kelas *review* hanya memiliki 1 operasi, yaitu *insert*. *Review* digunakan untuk mengelola data ulasan dan penilaian yang diberikan oleh pengguna terhadap produk atau layanan restoran. untuk tabel *review* memiliki relasi dengan tabel *tables* dengan relasi *mandatory options-to-one*.

### F. Flowchart Sistem

Flowchart akan dibagi menjadi 2 bagian, yaitu *flowchart admin*, dan *flowchart user*.

### 1. Flowchart Admin



Gambar 7. Flowchart Admin

Dalam alur *flowchart* admin, proses dimulai dengan admin mengakses halaman *login* admin. Kemudian, admin diminta untuk memasukkan informasi *login* menggunakan kombinasi *username* dan *password*. Apabila informasi yang dimasukkan benar, admin akan diarahkan ke halaman Dashboard admin. Tetapi, jika terdapat kesalahan dalam informasi yang dimasukkan, admin akan kembali ke halaman login. Dari halaman Dashboard, admin memiliki beberapa opsi untuk mengelola pesanan, termasuk mengubah pesanan dan menghapus pesanan.

Selain itu, admin juga memiliki wewenang untuk mengelola hal lain seperti kategori. Admin dapat menambah, mengedit, dan menghapus kategori sesuai kebutuhan. Selanjutnya, admin juga dapat mengelola produk dengan melakukan proses tambah, edit, dan hapus produk. Di halaman admin, admin memiliki akses untuk mengelola data meja dengan proses penghapusan meja.

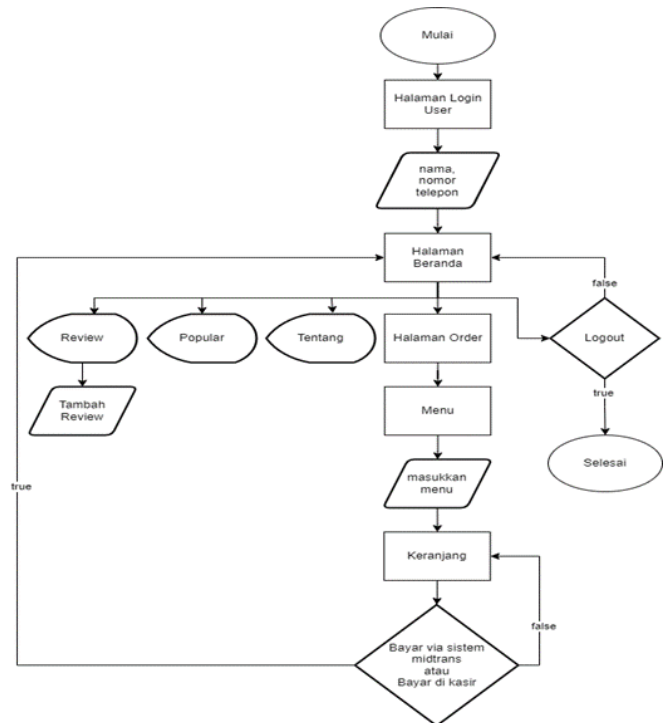
Terakhir, admin dapat mengelola akun admin lain dengan proses menambah, mengedit, dan menghapus akun admin. Setelah selesai dalam urusannya, admin dapat melakukan proses *Logout* dan keluar dari *website* admin E-Duren. Dengan begitu, admin memiliki kendali penuh atas berbagai aspek dalam sistem *website* E-Duren melalui halaman-halaman yang telah disebutkan di atas.

### 2. Flowchart User

Dalam *flowchart* user, proses dimulai ketika pengguna masuk ke restoran dan melakukan pemindaian kode QR di meja. Setelah itu, pengguna akan diarahkan ke halaman login user, di mana mereka diminta untuk memasukkan data berupa nama dan nomor telepon. Setelah pengguna mengisi data tersebut, mereka akan diarahkan ke halaman beranda. Di halaman beranda, pengguna dapat melihat menu populer, informasi tentang restoran, dan juga ulasan dari pengunjung lain. Selain itu, pengguna memiliki opsi untuk menambahkan ulasan mereka sendiri. Jika pengguna telah selesai dengan pesannya, mereka dapat melakukan *logout*. Selain itu, ada tombol "Order" yang mengarahkan pengguna ke halaman order.

Di halaman order, pengguna dapat melihat menu yang tersedia di restoran Sop Duren 97. Pengguna memiliki kesempatan untuk memasukkan menu-menu yang diinginkan

ke dalam keranjang. Setelah itu, pengguna dapat melanjutkan ke halaman pembayaran. Pembayaran dapat dilakukan di kasir atau langsung melalui aplikasi menggunakan layanan midtrans. Setelah selesai melakukan pembayaran, pengguna akan diarahkan kembali ke halaman beranda. Di sini, mereka dapat menambahkan ulasan tambahan jika diinginkan sebelum melakukan *logout* untuk menyelesaikan proses.



Gambar 8. Flowchart User

Dengan demikian, pengguna dapat dengan mudah melakukan pemesanan makanan dan memberikan ulasan melalui langkah-langkah yang telah dijelaskan dalam *flowchart* tersebut.

### G. Perancangan Teknik Pengujian Perangkat Lunak

Pengujian pada aplikasi web E-Duren ini dilakukan dengan *black box* testing, testing algoritma FCFS, dan testing *user acceptance test* (UAT).

### H. Black Box Testing

Dalam penelitian ini pengujian dilakukan fokus utama diberikan kepada pengalaman para pengguna, sehingga dapat dipastikan bahwa semua fitur dapat berjalan dengan tepat dan sesuai dengan harapan. Skenario pengujian mencakup berbagai situasi penggunaan untuk memastikan bahwa aplikasi berfungsi dengan baik dalam kondisi nyata. Pengujian fungsional ini penting Untuk memastikan bahwa semua fitur dalam aplikasi beroperasi dengan lancar dan memberikan nilai tambah yang maksimal bagi pengguna. Hasil pengujian selanjutnya disajikan dalam bentuk tabel sehingga lebih mudah untuk dievaluasi

### I. Testing Algoritma FCFS

Pengujian selanjutnya yang akan dilakukan pada aplikasi E-Duren adalah pengujian waktu antrian menggunakan Algoritma FCFS (*First-Come, First-Served*). Pengujian ini

bertujuan untuk mengukur waktu tunggu rata-rata pemesanan dalam aplikasi.

*J. User Acceptance Testing (UAT)*

Skala likert adalah skala penelitian yang digunakan untuk mengukur sikap dan pendapat. Skala ini digunakan untuk melengkapi kuesioner yang mengharuskan responden menunjukkan tingkat persetujuan terhadap serangkaian pertanyaan. Pengujian UAT (*User Acceptance Test*) menggunakan metode skala Likert melibatkan pengukuran penerimaan dan kepuasan pengguna terhadap perangkat lunak yang telah dikembangkan.

Adapun rumus perhitungan Skala Likert untuk mendapatkan tingkat kepuasan adalah sebagai berikut :

a. Total skor

$$Total\ Skor = T \times Pn \tag{1}$$

T adalah total jumlah responden yang memilih, sedangkan Pn adalah pilihan angka skor Likert.

b. Interpretasi Skor Perhitungan

Untuk mendapatkan hasil interpretasi, langkah pertama adalah mengetahui skor tertinggi (X) dan skor terendah (Y) untuk setiap item penilaian dengan rumus berikut:

$$Y = \frac{Jumlah\ Responden \times Nilai\ Tertinggi}{Jumlah\ Pertanyaan} \tag{2}$$

$$X = \frac{Jumlah\ Responden \times Nilai\ Terendah}{Jumlah\ Pertanyaan} \tag{3}$$

c. Index %

$$Index\ \% = \frac{JTotal\ skor}{Y} \times 100 \tag{4}$$

Sebelum menyelesaikan kita juga harus mengetahui *interval* (rentang jarak)

$$I = interval$$

$$I = 100 / Jumlah\ Skor\ (Likert)$$

Tabel 1. Interval Penilaian Skala Likert

Indeks %	Keterangan
0% - 19,99%	Sangat tidak setuju
20% - 39,99%	Tidak setuju
40% - 59,99%	Netral
60% - 79,99%	Setuju
80% - 100%	Sangat setuju

Berdasarkan Tabel 1, dapat dilihat interval penilaian Skala Likert yang memiliki jarak terendahnya 0% hingga jarak tertingginya 100%. Interval ini akan digunakan untuk pembandingan apakah hasil pengujian sudah setuju atau belum.

IV. METODOLOGI PENELITIAN

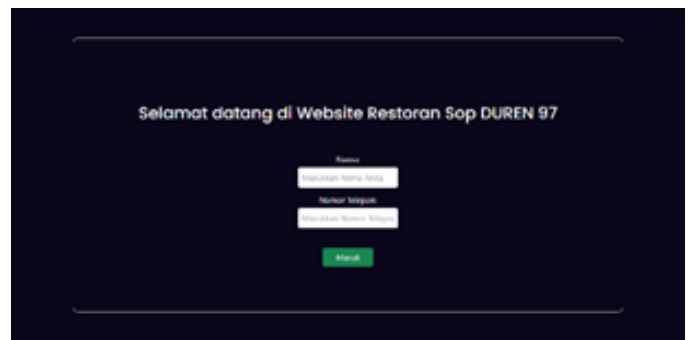
A. Implementasi

Berdasarkan rancangan antarmuka atau *user interface*, didapatkan hasil penerapan *user interface* aplikasi web E-Duren. Pengguna dapat melaksanakan beragam aktivitas melalui menu-menu yang tersedia dalam aplikasi web ini. Berikut ini adalah hasil dari tampilan antarmuka pengguna yang telah selesai dibuat.

1. Halaman Login user

Halaman *Login User* adalah tempat di mana pengguna

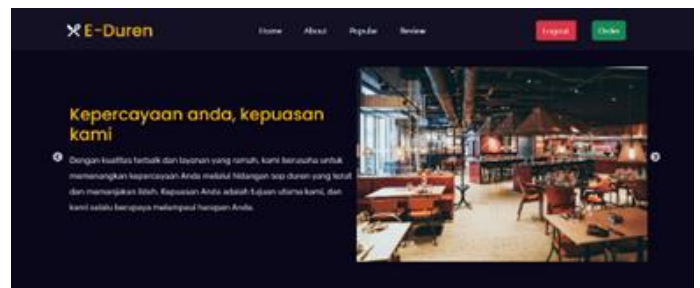
masuk ke dalam *platform* web E-Duren. Pada halaman ini, terdapat dua kotak teks yang memungkinkan pengguna untuk memasukkan nama dan nomor telepon mereka. Setelah mengisi kedua informasi tersebut, pengguna dapat menekan tombol "*Login*" untuk masuk ke dalam *website* E-Duren. Halaman ini memungkinkan pengguna untuk mengakses fitur-fitur yang hanya tersedia untuk pengguna yang telah memasukkan nama dan nomor telepon dalam sistem web E-Duren



Gambar 9 *Login User*

2. Halaman Beranda bagian navigasi dan banner

Halaman beranda dirancang dengan tata letak yang mudah dibaca. Terdapat logo di atas kiri dan tombol-tombol navigasi seperti *Home*, *About*, *Popular*, dan *Review* di bagian tengah navigasi. Di sebelah kanan navigasi, terdapat tombol *Logout* untuk keluar dari website dan *Order* untuk menuju halaman order untuk memesan makanan. Bagian *banner* terdiri dari slogan, teks, dan gambar yang dapat digeser untuk tampilan menarik. Halaman ini memberikan pengalaman pengguna yang menarik dan mudah diakses.

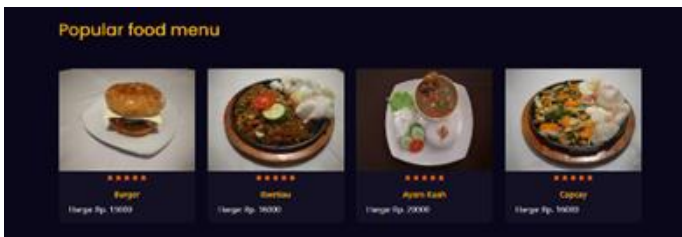


Gambar 10. Navigasi dan *banner*

3. Halaman beranda bagian *Popular*

Di bagian popular atau rekomendasi, terdapat sebuah teks besar yang menampilkan kalimat *Menu popular*. Di bawahnya, terdapat empat kartu (*card*) yang masing-masing berisi gambar dari menu favorit. Setiap kartu memiliki gambar bintang yang menunjukkan nilai atau rating makanan tersebut. Selain itu, setiap kartu juga menampilkan nama menu dan harganya di bagian bawah kartu.

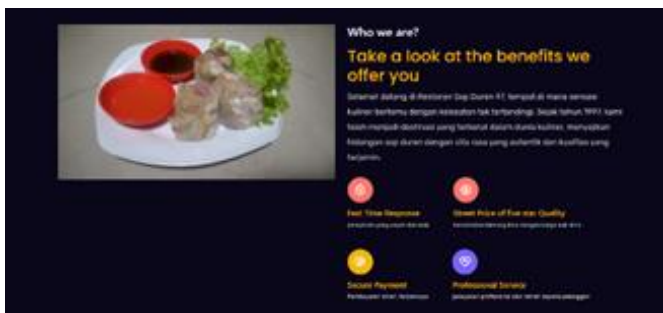




Gambar 11. *Popular*

#### 4. Halaman beranda bagian tentang

Bagian tentang pada halaman tersebut dibagi menjadi dua kolom. Di kolom kiri, terdapat gambar restoran. Sementara itu, di kolom kanan, terdapat teks yang berisi penjelasan tentang sejarah restoran dan berbagai fitur yang disediakan oleh restoran tersebut.



Gambar 12. *Tentang*

#### 5. Halaman beranda bagian review

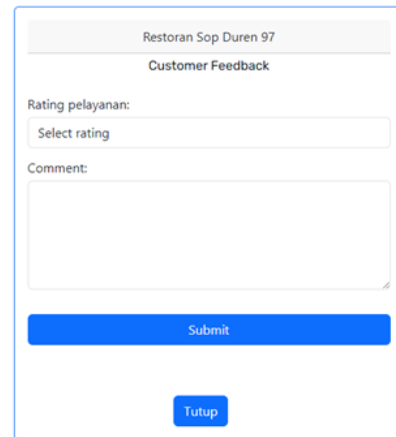
Bagian *review* pada halaman tersebut akan dibagi menjadi dua kolom. Di kolom kanan, terdapat gambar makanan yang di *review*. Di kolom kiri, terdapat bintang yang menunjukkan hasil *review* dari pelanggan. Selain itu, ada teks yang menampilkan nama pelanggan dan tombol untuk menambahkan *review* baru.



Gambar 13. *Review*

#### 6. Halaman beranda bagian tambah review

Halaman tambah *review* akan memiliki judul besar yang menampilkan nama restoran. Di bawahnya, terdapat teks yang mengajak pelanggan untuk memberikan *feedback* mengenai layanan dan produk restoran. Kemudian, akan ada sebuah *dropdown* yang berisi nilai dari bintang 1 hingga 5, yang memungkinkan pelanggan memberikan penilaian. Selanjutnya, terdapat input untuk menuliskan komentar dan *review* di sebuah *textarea*. Terakhir, akan ada tombol "*Submit*" untuk mengirimkan *review* dan tombol "*Tutup*" untuk menutup halaman tambah *review*.



Gambar 14. *Tambah review*

#### 7. Halaman beranda bagian footer

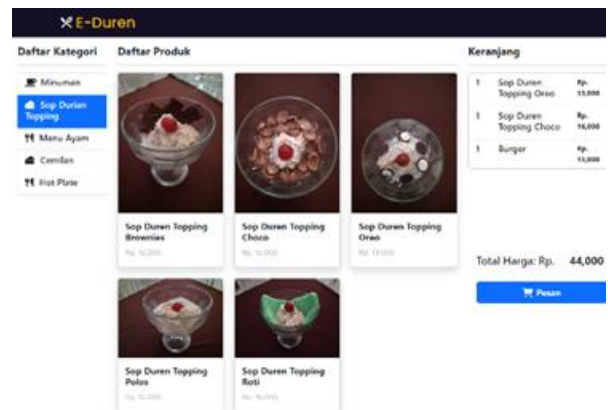
Bagian *footer* akan menampilkan gambar logo restoran di bagian atas, dan di bawahnya terdapat teks "*Copyright*" yang menandakan hak cipta dari restoran tersebut.



Gambar 15. *Footer*

#### 8. Halaman Order

Halaman "*Order*" memiliki 3 kolom. Di bagian navigasi terdapat logo restoran. Bagian kiri kolom menampilkan kategori-kategori menu yang tersedia terdiri dari text. Bagian tengah menampilkan card berisi gambar, nama menu, dan harga untuk setiap menu. Bagian kanan menampilkan hasil pesanan dengan satu kotak yang berisi informasi jumlah, nama menu makanan, dan harga. Di bagian bawah kanan terdapat teks yang menunjukkan total harga dari menu yang telah dimasukkan, diikuti oleh tombol "*Pesan*" untuk menyelesaikan pesanan.

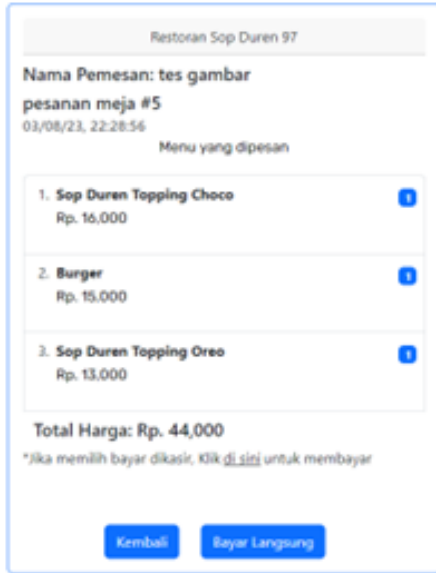


Gambar 16. *Order*

#### 9. Halaman keranjang

Pada halaman "*Keranjang*", terdapat teks nama restoran di bagian atas. Di bawahnya, terdapat teks yang memberikan informasi tentang nama pemesan, nomor meja, dan jam pemesanan. Kemudian, terdapat sebuah kotak yang

menampilkan informasi tentang menu yang dipesan, termasuk nomor, nama produk, harga, dan jumlah pesanan. Di bawah kotak tersebut, terdapat teks yang menunjukkan total harga pesanan. Di bawahnya lagi, terdapat teks yang memberikan informasi tentang proses pembayaran. Selanjutnya, terdapat dua tombol di bagian paling bawah halaman. Tombol kanan digunakan untuk pembayaran langsung, sedangkan tombol kiri digunakan untuk kembali ke halaman sebelumnya.



Gambar 17. Keranjang

**B. Hasil Pengujian Aplikasi**

Berdasarkan rancangan pengujian aplikasi, dalam pembangunan aplikasi ini telah dilakukan uji coba sistem dengan pengujian blackbox dengan hasil yang terdokumentasi dalam Tabel 2 berikut ini.

Tabel 2. *BlackBox* Halaman *Login User*

No.	Halaman yang Diuji	Aksi	Reaksi Sistem <i>website</i>		Hasil
			Benar	Salah	
1	Login User	Memasukkan nama dan nomor telepon <i>user</i>	Masuk ke halaman beranda	Tetap di halaman <i>login user</i>	Berhasil
2		Hanya memasukkan nama	Muncul pesan <i>error</i> dan tidak menuju halaman beranda	Menuju halaman beranda	Berhasil
3		Hanya memasukkan nomor telepon	Muncul pesan <i>error</i> dan tidak menuju halaman beranda	Menuju halaman beranda	Berhasil
4		Tidak memasukkan nama dan nomor telepon	Muncul pesan <i>error</i> dan tidak menuju halaman beranda	Menuju halaman beranda	Berhasil

Tabel 3. *BlackBox* halaman Beranda

No	Halaman yang Diuji	Aksi	Reaksi Sistem <i>website</i>		Hasil
			Benar	Salah	
1	Beranda bagian navigasi dan banner	Menekan tombol <i>Home</i>	Halaman akan berada diatas	Tidak terjadi apa-apa	Berhasil
2		Menekan tombol <i>About</i>	Halaman akan pindah ke bagian <i>about</i>	Tidak terjadi apa-apa	Berhasil
3		Menekan tombol <i>Popular</i>	Halaman akan pindah ke bagian <i>popular</i>	Tidak terjadi apa-apa	Berhasil
4		Menekan tombol <i>Review</i>	Halaman akan pindah ke bagian <i>review</i>	Tidak terjadi apa-apa	Berhasil
5		Menekan tombol geser banner	Banner akan bergeser dan menampilkan konten lain	Banner tidak bergerak	Berhasil
6	Beranda bagian <i>review</i>	Menekan tombol geser <i>review</i>	<i>Review</i> akan bergeser dan menampilkan <i>review</i> lain	<i>Review</i> tidak bergerak	Berhasil
7		Menekan tombol tambah <i>review</i>	Masuk ke halaman tambah <i>review</i>	Tetap di halaman beranda	Berhasil
8		Menekan tombol <i>logout</i>	Menuju halaman <i>login user</i>	Tetap di halaman beranda	Berhasil
9		Menekan tombol <i>order</i>	Menuju halaman <i>order</i>	Tetap di halaman beranda	Berhasil

Tabel 4. *BlackBox* halaman Order

No	Halaman yang Diuji	Aksi	Reaksi Sistem <i>website</i>		Hasil
			Benar	Salah	
1	Order Bagian kategori	Menekan tombol Kategori Minuman	Bagian produk akan menampilkan menu minuman	Bagian produk tidak menampilkan menu minuman	Berhasil
2		Menekan tombol Kategori <i>Topping</i> durian	Bagian produk akan menampilkan menu <i>topping</i> durian	Bagian produk tidak menampilkan menu <i>topping</i> durian	Berhasil
3		Menekan tombol produk	Bagian produk	Bagian produk	Berhasil

		Kategori Menu ayam	akan menampilkan menu ayam	tidak menampilkan menu ayam	
4		Menekan tombol Kategori Cemilan	Bagian produk akan menampilkan menu cemilan	Bagian produk tidak menampilkan menu cemilan	Berhasil
5		Menekan tombol Kategori Hot plate	Bagian produk akan menampilkan menu hot plate	Bagian produk tidak menampilkan menu hot plate	Berhasil
6	Order bagian produk	Menekan menu di daftar produk	Menu akan masuk kedalam keranjang	Menu tidak masuk keranjang	Berhasil
7	Menekan menu di keranjang	Akan muncul detail pesanan	Tidak terjadi apa apa	Berhasil	Menekan menu di keranjang
8	User mengubah jumlah porsi dan menambahkan keterangan dan menekan tombol simpan	Perubahan akan tersimpan dan muncul di keranjang	Data tidak tersimpan dan tidak muncul di keranjang	Berhasil	User mengubah jumlah porsi dan menambahkan keterangan dan menekan tombol simpan
9	User menghapus menu dengan menekan tombol hapus pesanan	Menu akan hilang dari keranjang	Menu masih ada di keranjang	Berhasil	User menghapus menu dengan menekan tombol hapus pesanan
10	User memesan menu yang dipilih dengan menekan tombol pesan	Menuju halaman pembayaran	Tetap di halaman Order	Berhasil	User memesan menu yang dipilih dengan menekan tombol pesan

Tabel 5. *BlackBox* halaman pembayaran

No.	Halaman yang Diuji	Aksi	Reaksi Sistem website		Hasil
			Benar	Salah	
1	Pembayaran	Menekan tombol disini	Masuk ke halaman beranda	Tetap di halaman pembayaran	Berhasil
2		Menekan tombol kembali	Kembali ke halaman Order	Tetap di halaman pembayaran	Berhasil
3	Bagian bayar langsung	Menekan tombol bayar langsung	Muncul <i>popup</i> pembayaran khusus dari	Tidak terjadi apa apa	Berhasil

			midtrans		
4		User memilih metode pembayaran via midtrans	Muncul nomor <i>Virtual Account</i> yang dapat dibayar	Nomor <i>Virtual Account</i> tidak keluar	Berhasil
5		Melakukan pembayaran	Kembali ke halaman beranda	Tetap berada di halaman pembayaran	Berhasil

Tabel 6. *BlackBox* halaman Login admin

No	Halaman yang Diuji	Aksi	Reaksi Sistem website		Hasil
			Benar	Salah	
1	Login Admin	Memasukkan <i>username</i> dan <i>password</i> dengan benar	Masuk ke halaman <i>Dashboard</i> admin	Tetap di halaman <i>login</i> admin	Berhasil
2		Memasukkan <i>username</i> dan <i>password</i> yang salah	Muncul pesan <i>error</i> dan tidak menuju halaman <i>Dashboard</i>	Tetap di halaman <i>login</i> admin	Berhasil
3		Hanya memasukkan <i>username</i>	Muncul pesan <i>error</i> dan tidak menuju halaman <i>Dashboard</i>	Menuju halaman <i>Dashboard</i>	Berhasil
4		Hanya memasukkan <i>password</i>	Muncul pesan <i>error</i> dan tidak menuju halaman <i>Dashboard</i>	Menuju halaman <i>Dashboard</i>	Berhasil
5		Tidak memasukkan <i>username</i> dan <i>password</i>	Muncul pesan <i>error</i> dan tidak menuju halaman <i>Dashboard</i>	Menuju halaman <i>Dashboard</i>	Berhasil

Tabel 7. *BlackBox* halaman daftar admin

No	Halaman yang Diuji	Aksi	Reaksi Sistem website		Hasil
			Benar	Salah	
1	Daftar Admin	Memasukkan <i>username</i> , <i>email</i> dan <i>password</i>	Masuk ke halaman <i>login</i>	Tetap di halaman daftar admin	Berhasil
2		Hanya memasukkan <i>username</i>	Muncul pesan <i>error</i> dan akun tidak terbuat	Menuju halaman <i>login</i>	Berhasil
3		Hanya memasukkan <i>email</i>	Muncul pesan <i>error</i> dan akun tidak	Menuju halaman <i>login</i>	Berhasil

4	Hanya memasukkan <i>password</i>	Muncul pesan <i>error</i> dan akun tidak terbuat	Menuju halaman <i>login</i>	Berhasil
5	Tidak memasukkan <i>username</i> dan <i>password</i>	Muncul pesan <i>error</i> dan akun tidak terbuat	Menuju halaman <i>login</i>	Berhasil
6	Menekan tombol sudah punya akun	Masuk ke halaman <i>login</i>	Tetap di halaman daftar <i>user</i>	Berhasil

Table 8. *BlackBox* halaman *Dashboard* admin

No	Halaman yang Diuji	Aksi	Reaksi Sistem <i>website</i>		Hasil
			Benar	Salah	
1	Dashboard admin	Melakukan pencarian data pesanan	Data akan muncul sesuai pencarian	Data tidak akan muncul	Berhasil
2		Melakukan perubahan status pembayaran dengan menekan tombol status	Status berubah menjadi <i>true</i>	Status tidak berubah	Berhasil
3	Bagian kategori	Melakukan penambahan data baru	Data kategori akan bertambah dan muncul pada tabel	Data kategori tidak bertambah	Berhasil
4		Mengubah data kategori dengan menekan tombol edit	Data kategori berhasil dirubah dan muncul	Tidak terjadi perubahan pada data kategori	Berhasil
5		Menghapus data kategori dengan menekan tombol hapus	Data kategori hilang dari tabel	Data kategori tidak hilang dari tabel	Berhasil
6	Bagian produk	Melakukan penambahan data baru	Data produk akan bertambah dan muncul pada tabel	Data produk tidak bertambah	Berhasil
7		Mengubah data produk dengan menekan tombol edit	Data produk berhasil dirubah dan muncul	Tidak terjadi perubahan pada data produk	Berhasil
8		Menghapus data produk dengan menekan	Data produk hilang dari tabel	Data produk tidak hilang dari	Berhasil

9	Bagian table	Melakukan pencarian data produk	Data akan muncul sesuai pencarian	Data tidak akan muncul	Berhasil
10		Melakukan pencarian data <i>table</i>	Data akan muncul sesuai pencarian	Data tidak akan muncul	Berhasil
11	Bagian admin	Menghapus data <i>table</i>	Data <i>table</i> hilang dari <i>table</i>	Data <i>table</i> tidak hilang dari <i>table</i>	Berhasil
12		Melakukan penambahan data baru	Data admin akan bertambah dan muncul pada tabel	Data admin tidak bertambah	Berhasil
13	Bagian admin	Mengubah data admin dengan menekan tombol edit	Data admin berhasil dirubah dan muncul	Tidak terjadi perubahan pada data admin	Berhasil
14		Menghapus data admin dengan menekan tombol hapus	Data admin hilang dari tabel	Data admin tidak hilang dari tabel	Berhasil
15	Bagian <i>logout</i>	<i>Logout</i>	Ke halaman <i>login</i> admin	Tetap di halaman <i>Dashboard</i>	Berhasil

Berdasarkan hasil *blackbox* testing menunjukkan bahwa aplikasi web E-duren yang telah dibuat telah berhasil secara fungsionalitas dan memenuhi syarat yang diharapkan. Fitur-fitur yang diujikan dalam *blackbox* testing berjalan dengan baik dan sesuai dengan spesifikasi yang telah ditetapkan sebelumnya. Semua interaksi antarmuka pengguna telah diuji secara menyeluruh dan memberikan respons yang konsisten. Tidak ditemukan kesalahan fatal yang dapat mengganggu penggunaan normal aplikasi. Oleh karena itu, aplikasi web E-duren telah siap untuk tahap pengujian selanjutnya atau peluncuran ke pengguna.

### C. Pengujian Waktu Pemesanan

Hasil pengujian dengan 5 pengguna dan didapat data rata-rata waktu tunggu seperti pada tabel 9 berikut.

Tabel 9. Pengujian waktu pemesanan

Nama Pemesan	Waktu Masuk	Waktu Order	Lama Eksekusi
Surya	21:41	21:45	4
Danar	21:41	21:46	5
Adit	21:44	21:46	2
Asep	21:44	21:46	2
Dimas	21:45	21:46	1

Waktu =  $4 + 5 + 2 + 2 + 1 = 14 / 5 = 2,8$  menit untuk rata-rata pemesanan dan setelah dilakukan percobaan dengan 5 orang menggunakan aplikasi e-duren rata-rata waktu pemesanan melalui website adalah 2,8 menit.



#### D. Pengujian UAT

Hasil dari pengambilan data ini digunakan untuk mengevaluasi kelayakan dan efektivitas aplikasi yang telah dibangun, serta untuk memberikan wawasan mengenai bagaimana pengguna merasakan dan berinteraksi dengan aplikasi tersebut. Berdasarkan hasil uji dengan kuesioner dapat diperhitungkan hasil indeks di bawah ini.

$$\text{Total skor} = (0 \times 1) + (1 \times 2) + (42 \times 3) + (65 \times 4) + (72 \times 5) = 748$$

$$\text{Skor tertinggi (Y)} = \frac{\text{Jumlah responden} \times \text{nilai tertinggi}}{\text{jumlah pertanyaan}} = 30 \times 5 \times 6 = 900$$

$$\text{Skor terendah (X)} = \frac{\text{Jumlah responded} \times \text{nilai terendah}}{\text{jumlah pertanyaan}} = 30 \times 1 \times 6 = 180$$

$$\text{Interval (I)} = \frac{100}{\text{Jumlah skor Likert}} = \frac{100}{5} = 20$$

$$\text{Indeks \%} = \frac{\text{Total skor}}{\text{Skor tertinggi (Y)}} \times 100 = \frac{748}{900} \times 100 = 83,1\%$$

Hasil pengujian menunjukkan bahwa aplikasi telah mencapai tingkat persetujuan sebesar 83,1%. Berdasarkan interval penilaian, angka ini berada dalam kategori "Sangat Setuju", menandakan bahwa mayoritas responden memiliki pandangan positif terhadap aplikasi ini.

#### V. KESIMPULAN

Pada penelitian ini, aplikasi website E-Duren berhasil dibuat, yang membuat pengguna dapat melakukan proses pemesanan makanan pada halaman order, lalu pengguna juga dapat melakukan proses pembayaran baik secara langsung di kasir, maupun secara langsung dari aplikasi menggunakan midtrans. Selain itu pengguna juga dapat memberikan feedback dan rating pada pelayanan dan website restoran Sop Duren 97. Berdasarkan hasil pengujian blackbox yang telah dilakukan, hasilnya sudah berhasil dengan target yang diharapkan dari segi fungsionalitas. Sedangkan dari hasil pengujian waktu didapatkan rata-rata waktu pemesanan dari 5 orang adalah 2,8 menit. Berdasarkan hasil User Acceptance Test (UAT) menggunakan metode skala likert menunjukkan bahwa aplikasi website E-Duren mendapatkan indeks persentase 83,1% yang dapat disimpulkan bahwa aplikasi website E-Duren sudah dapat berjalan lancar. Aplikasi web E-Duren dapat diakses melalui QR yang di scan pada meja atau melalui link berikut ini : <https://e-duren-site.vercel.app/table/3>

Aplikasi Website E-Duren dapat ditingkatkan melalui pengembangan aplikasi yang menghubungkan antara admin dengan dapur, memungkinkan untuk meningkatkan efisiensi dalam proses pelayanan di Restoran Sop Duren 97. Selain itu, dalam tahap pengembangan pembayaran, dapat dilakukan peralihan dari mode pengujian ("sandbox") ke mode produksi ("production") agar dapat diterapkan secara langsung di dalam restoran. Selain pengembangan tersebut, penelitian selanjutnya dapat mempertimbangkan penerapan algoritma lain untuk menganalisis perbedaan penggunaan algoritma dalam konteks yang berbeda. Dengan melibatkan algoritma yang beragam, akan memungkinkan pemahaman yang lebih baik tentang kelebihan dan kekurangan masing-masing algoritma dalam situasi tertentu.

#### DAFTAR PUSTAKA

- [1] G. Eldies, "Rancang Bangun Aplikasi Android Untuk Pemesanan Dan Pembayaran Menggunakan Metode First Come First Served (FCFS) - Repository Politeknik Negeri Jakarta," *Pnj.ac.id*, Aug. 2021.
- [2] N. S. Anam, "Implementasi Metode FCFS (First Come First Served) pada Aplikasi Pemesanan Makanan Menggunakan QR Code Berbasis Web service - Repository UM Jember," *Unmuhjember.ac.id*, 2018.
- [3] S. Syofian and A. A. Damar, "Implementasi Algoritma First Come First Served Dan Haversine Pada Aplikasi Pemesanan Makanan Berbasis Mobile," *Jurnal Sains & Teknologi Fakultas Teknik Universitas Darma Persada*, vol. 10, no. 1, pp. 31–40, 2020, Accessed: Feb. 04, 2024. [Online]. Available: <https://unsada.e-journal.id/jst/article/view/75>
- [4] M. Mega and D. Maulina, "Implementasi Payment Gateway Menggunakan Midtrans Pada Marketplace Travnesia.com," *Mobile and Forensics*, vol. 1, no. 1, pp. 22–22, Sep. 2019, doi: <https://doi.org/10.12928/mf.v1i1.997>.
- [5] M. A. Habibirrahman, N. Putra, and B. T. Hanggara, "Pengembangan Sistem Pemesanan Kue Berbasis Website Menggunakan Midtrans Webservice Sebagai Payment Gateway (Studi Kasus: Toko Kue De Tasty)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 2, pp. 597–604, 2022.
- [6] R. Abdullah, "Pemrograman Web Untuk Pemula" *Media Komputindo*, Jakarta, 2015.
- [7] H. Alatas, "Responsive web design dengan php dan bootstrap," *Lokomedia*, 2013.
- [8] A. Afriansyah and R. Annisa, "Pengembangan Sistem ERP HRD PT Queen Network Nusantara Menggunakan Framework React," *Electrician: Jurnal Rekayasa dan Teknologi Elektro*, vol. 17, no. 1, pp. 120–127, Jan. 2023, doi: <https://doi.org/10.23960/elc.v17n1.2439>.
- [9] L. D. Ariany, "Sistem Pendukung Keputusan Pemilihan Wajib Pajak Daerah Self-Assessment Teladan Kota Batam dengan Metode SMART - Repository Universitas Internasional Batam," *Uib.ac.id*, Aug. 2018.
- [10] F. A. Batubara, "Perancangan Website Pada PT. Ratu Enim Palembang," *Jurnal Ilmu Pengetahuan Dan Teknologi Terapan "REINTEK" (Rekayasa Inovasi Teknologi)*, vol. 7, no. 1, 2024.
- [11] Alfian, P. Sokibi and L. Magdalena, "Penerapan Payment Gateway pada Aplikasi Marketplace Waroeng Mahasiswa Menggunakan Midtrans," *Jurnal Informatika Universitas Pamulang*, Vol. 5, No. 3, 2020.
- [12] M. W. Atmodjo, "Restoran Permasalahannya," *Andi Offset*, Yogyakarta, 2005.
- [13] G. Chizoba, "Comparative Analysis between First-Come-First-Serve (FCFS) and Shortest-Job-First (SJF) Scheduling Algorithms," *Unijos.edu.ng*, 2019, doi: <https://doi.org/2320%E2%80%9393088X>.
- [14] S. Pemasinghe and S. Rajapaksha, "Comparison of CPU Scheduling Algorithms: FCFS, SJF, SRTF, Round Robin, Priority Based, and Multilevel Queuing," *2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC)*, Hyderabad, India, 2022, pp. 318–323, doi: [10.1109/R10-HTC54060.2022.9929533](https://doi.org/10.1109/R10-HTC54060.2022.9929533).
- [15] D. Fratiwi and N. Mariana, "Metode FCFS dalam menunjang Sistem Layanan Antrian Pebagian Dana Pensiun Studi Kasus Kantor Pos Bongsari," *Proceeding Sendu*, 2020.
- [16] N. Normah, "Sistem Informasi Konsultasi Kesehatan Berbasis Web," *Jurnal Sistem Informasi*, vol. 5, no. 2, pp. 196–200, Jan. 2016, doi: <https://doi.org/10.51998/jsi.v5i2.106>.
- [17] R. Muhammad, "Aplikasi Penjadwalan Guru Berbasis Web Dengan Framework Bootstrap Studi Kasus di Madrasah Wahid Hasyim Yogyakarta - UTDI Repository," *Utdi.ac.id*, Feb. 2017.