

IMPLEMENTASI *OPENROUTESERVICE* DAN *MIDTRANS* PADA SISTEM E-BOOKING JASA MAKEUP ARTIST BERBASIS WEB

Danendra Ganijan Sunarso^{1*}, Arip Solehudin², Agung Susilo Yuda Irawan³

Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang

Keywords:

E-Booking;
Makeup Artist;
OpenRouteService;
Midtrans;
Payment Gateway.

Correspondent Email:

2210631170062@student.unsika.ac.id

Perkembangan industri jasa kecantikan, khususnya layanan Makeup Artist, mendorong kebutuhan terhadap sistem pemesanan yang efektif, transparan, dan terintegrasi. Proses pemesanan yang masih dilakukan secara manual melalui pesan instan memiliki beberapa kendala, seperti risiko bentrok jadwal, perhitungan biaya transportasi yang tidak konsisten, serta verifikasi pembayaran yang masih dilakukan secara manual. Penelitian ini bertujuan untuk mengimplementasikan *OpenRouteService* dan *Midtrans* pada sistem e-booking jasa Makeup Artist berbasis web. Metode pengembangan sistem menggunakan Software Development Life Cycle dengan model prototyping. Sistem dibangun menggunakan Next.js sebagai framework pengembangan web dan Supabase sebagai basis data. *OpenRouteService* digunakan untuk memperoleh data jarak tempuh berdasarkan koordinat lokasi klien dan MUA, sedangkan biaya transportasi dihitung oleh sistem melalui logika program berdasarkan jarak dan tarif yang telah ditentukan. *Midtrans* digunakan untuk memproses dan memvalidasi pembayaran secara online. Pengujian dilakukan menggunakan Blackbox Testing dan User Acceptance Testing. Hasil penelitian menunjukkan bahwa sistem mampu mendukung proses pemesanan mandiri, pengelolaan jadwal, perhitungan biaya transportasi otomatis, serta validasi pembayaran secara real-time. Implementasi sistem ini dapat meningkatkan efisiensi operasional dan transparansi layanan Makeup Artist.



Copyright © 2026 The Author(s), Published by Jurnal Informatika dan Teknik Elektro Terapan. This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

The growth of the beauty service industry, particularly Makeup Artist services, has increased the need for an effective, transparent, and integrated booking system. Manual booking processes through instant messaging still face several problems, such as scheduling conflicts, inconsistent transportation cost calculation, and manual payment verification. This study aims to implement OpenRouteService and Midtrans in a web-based e-booking system for Makeup Artist services. The system development method applies the Software Development Life Cycle using the prototyping model. The system was developed using Next.js as the web development framework and Supabase as the database service. OpenRouteService is used to obtain travel distance data based on the coordinates of the client and MUA locations, while transportation costs are calculated by the system through program logic based on distance and predetermined rates. Midtrans is used to process and validate online payments. Testing was conducted using Blackbox Testing and User Acceptance Testing. The results show that the system can support self-service booking, schedule management, automatic transportation cost calculation, and real-time payment validation. The implementation of this system can improve operational efficiency and service transparency for Makeup Artist businesses.

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong berbagai sektor usaha untuk

melakukan transformasi digital, termasuk pada industri jasa kecantikan. Salah satu layanan yang mengalami perkembangan cukup pesat

adalah jasa Makeup Artist atau MUA. Profesi MUA tidak hanya berperan sebagai penyedia layanan tata rias, tetapi juga menjadi bagian dari industri kreatif yang memanfaatkan media digital untuk promosi, komunikasi, dan pelayanan pelanggan. Media sosial seperti Instagram dan TikTok sering digunakan sebagai media promosi portofolio, sedangkan komunikasi pemesanan umumnya dilakukan melalui aplikasi pesan instan [5], [6].

Pada praktiknya, sebagian besar proses pemesanan jasa MUA masih dilakukan secara manual melalui WhatsApp atau Instagram Direct Message. Proses tersebut memungkinkan calon klien untuk menanyakan jadwal, harga layanan, lokasi, serta metode pembayaran secara langsung kepada penyedia jasa. Namun, proses manual tersebut memiliki beberapa kelemahan, seperti waktu respons yang tidak pasti, risiko kesalahan pencatatan, kemungkinan terjadinya bentrok jadwal atau double-booking, serta kurangnya transparansi dalam perhitungan total biaya layanan [2], [3], [7].

Permasalahan lain yang cukup penting pada layanan MUA adalah perhitungan biaya transportasi. Sebagian besar MUA menyediakan layanan panggilan atau home service, sehingga biaya transportasi menjadi komponen penting dalam total biaya pemesanan. Perhitungan biaya transportasi yang dilakukan secara manual dapat menimbulkan ketidaktepatan dan kurang transparan bagi klien. Selain itu, proses pembayaran yang masih menggunakan transfer manual juga mengharuskan MUA untuk memeriksa bukti pembayaran satu per satu, sehingga menambah beban administrasi dan memperlambat proses konfirmasi pesanan [8], [9], [10].

Untuk mengatasi permasalahan tersebut, diperlukan sistem e-booking berbasis web yang mampu mengelola proses pemesanan secara terpusat. Sistem pemesanan berbasis web telah banyak diterapkan untuk meningkatkan efisiensi layanan, mempercepat proses transaksi, dan mengurangi kesalahan pencatatan pada berbagai bidang jasa [2], [4], [7]. Dalam penelitian ini, sistem tidak hanya

menyediakan fitur pemesanan dan manajemen jadwal, tetapi juga mengintegrasikan layanan pihak ketiga untuk meningkatkan otomatisasi proses bisnis.

OpenRouteService dapat digunakan untuk menghitung jarak tempuh berdasarkan titik lokasi klien dan MUA, sedangkan *Midtrans* dapat digunakan sebagai payment gateway untuk memproses pembayaran secara online. Integrasi payment gateway pada aplikasi berbasis web telah terbukti dapat mendukung transaksi non-tunai dan meningkatkan efisiensi operasional melalui validasi pembayaran yang lebih otomatis [1], [14], [15]. Oleh karena itu, integrasi *OpenRouteService* dan *Midtrans* menjadi bagian penting dalam sistem e-booking jasa Makeup Artist yang dikembangkan.

Penelitian ini berfokus pada implementasi *OpenRouteService* dan *Midtrans* pada sistem e-booking jasa Makeup Artist berbasis web. Kebaruan penelitian ini terletak pada integrasi fitur perhitungan biaya transportasi otomatis dan pembayaran online dalam satu alur pemesanan. Dengan adanya sistem ini, proses pemesanan diharapkan menjadi lebih efisien, transparan, dan mampu mengurangi risiko kesalahan administrasi.

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan *OpenRouteService* pada sistem e-booking jasa Makeup Artist untuk menghitung jarak dan biaya transportasi secara otomatis.
2. Mengimplementasikan *Midtrans* sebagai payment gateway untuk mendukung proses pembayaran online.
3. Menguji fungsionalitas sistem menggunakan Blackbox Testing dan tingkat penerimaan pengguna menggunakan User Acceptance Testing.

2. TINJAUAN PUSTAKA

2.1 Sistem E-Booking

Sistem e-booking merupakan sistem pemesanan berbasis elektronik yang memungkinkan pengguna melakukan reservasi layanan atau produk secara daring. Sistem ini memberikan kemudahan bagi pengguna dalam melakukan pemesanan tanpa harus datang langsung atau menunggu konfirmasi manual dari penyedia jasa. Pada layanan jasa, sistem e-booking berperan penting dalam mengelola data pelanggan, jadwal, layanan, dan status transaksi secara terpusat [2], [4].

Dalam konteks jasa Makeup Artist, sistem e-booking dapat membantu klien melihat daftar layanan, memilih jadwal, menentukan lokasi, dan melakukan pembayaran secara mandiri. Bagi penyedia jasa, sistem ini dapat membantu mengelola jadwal, memantau pesanan, mengurangi risiko double-booking, serta mempercepat proses administrasi [3], [7], [10].

2.2 Makeup Artist

Makeup Artist merupakan penyedia jasa profesional di bidang tata rias yang melayani berbagai kebutuhan, seperti acara pernikahan, wisuda, pemotretan, acara formal, maupun acara khusus lainnya. Model bisnis MUA umumnya bersifat fleksibel dan mobile karena penyedia jasa sering kali mendatangi lokasi klien. Karakteristik tersebut membuat sistem pemesanan jasa MUA membutuhkan pengelolaan jadwal yang baik serta perhitungan biaya transportasi yang akurat [5], [6], [8].

2.3 OpenRouteService

OpenRouteService merupakan layanan berbasis Application Programming Interface yang menyediakan berbagai fungsi geospasial, seperti routing, geocoding, dan perhitungan jarak. Pada sistem e-booking jasa Makeup Artist, *OpenRouteService* digunakan untuk menghitung jarak tempuh antara lokasi MUA dan lokasi klien. Hasil perhitungan jarak tersebut kemudian digunakan sebagai dasar perhitungan biaya transportasi secara otomatis [11], [12], [13].

2.4 Midtrans

Midtrans merupakan salah satu payment gateway yang digunakan untuk memproses transaksi pembayaran secara online. Layanan ini mendukung berbagai metode pembayaran, seperti virtual account, e-wallet, kartu kredit, dan transfer bank. Pada sistem e-booking jasa Makeup Artist, *Midtrans* digunakan untuk membuat transaksi pembayaran, menampilkan instruksi pembayaran, dan memperbarui status pembayaran berdasarkan respons sistem [1], [14], [15].

2.5 Model Prototyping

Model prototyping merupakan salah satu model dalam Software Development Life Cycle yang menekankan proses pengembangan sistem secara iteratif. Model ini memungkinkan pengembang membuat rancangan awal sistem, memperoleh umpan balik dari pengguna, kemudian melakukan perbaikan hingga sistem sesuai dengan kebutuhan. Model ini cocok digunakan pada pengembangan sistem yang berorientasi pada pengalaman pengguna, seperti sistem e-booking berbasis web [1], [19].

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan pengembangan sistem dengan metode Software Development Life Cycle model prototyping. Model ini dipilih karena sistem yang dikembangkan membutuhkan evaluasi dan umpan balik langsung dari pengguna, terutama dalam aspek alur pemesanan, tampilan antarmuka, dan kemudahan penggunaan. Model prototyping juga digunakan pada penelitian pengembangan aplikasi berbasis web karena memungkinkan sistem dibangun secara bertahap berdasarkan kebutuhan pengguna [1], [19].

Tahapan penelitian terdiri dari analisis kebutuhan, perancangan cepat, evaluasi prototipe awal, pembangunan prototipe, evaluasi pengguna, perbaikan prototipe, dan implementasi sistem.

3.1 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk mengidentifikasi permasalahan yang terjadi pada proses pemesanan jasa Makeup Artist. Pengumpulan data dilakukan melalui observasi

proses bisnis, wawancara dengan pihak MUA, dan studi literatur terhadap penelitian terdahulu yang relevan. Berdasarkan hasil analisis, ditemukan bahwa sistem harus mampu mengelola proses pemesanan, jadwal, biaya transportasi, pembayaran, serta laporan transaksi secara terpusat [2], [3], [7].

Kebutuhan fungsional sistem disusun berdasarkan fitur utama yang diperlukan oleh pengguna. Kebutuhan ini mencakup fitur yang dapat digunakan oleh tiga aktor utama, yaitu *Client*, MUA, dan Admin.

Tabel 1 Kebutuhan Fungsional

No	Kebutuhan	Keterangan
1	Autentikasi Akun	Sistem memfasilitasi pengguna untuk melakukan pendaftaran akun dan proses masuk (<i>login</i>) guna mengenali tingkat akses pengguna, baik sebagai <i>Client</i> , MUA, maupun Admin.
2	Manajemen Hak Akses Admin	Sistem membatasi wewenang Admin secara ketat. Admin hanya dapat mengubah peran (<i>role</i>) dan menghapus akun pengguna, namun tidak diberikan akses untuk melihat transaksi, menghapus ulasan, atau mereset data operasional.
3	Manajemen Layanan dan Harga	memiliki akses penuh untuk mengubah daftar paket layanan dan <i>add-ons</i> secara dinamis, sementara <i>Client</i> dapat melihat daftar tersebut secara transparan.
4	Manajemen Portofolio	MUA dapat mengunggah dan mengelola foto-foto hasil riasan pada galeri portofolio untuk meningkatkan daya tarik layanan bagi calon pelanggan.
5	Pemesanan Jadwal (Booking)	Klien dapat melakukan pemesanan jadwal secara mandiri melalui formulir reservasi, dan MUA dapat memantau serta mengelola seluruh data pesanan yang masuk melalui dasbor operasional
6	Otomatisasi Biaya Transportasi	Sistem mengambil titik koordinat lokasi klien, kemudian menggunakan <i>OpenRouteService</i> untuk memperoleh jarak tempuh. Biaya transportasi dihitung oleh sistem berdasarkan jarak dan tarif yang telah ditentukan.
7	Otomatisasi Pembayaran	Sistem mengintegrasikan API <i>Midtrans</i> untuk memproses transaksi elektronik, membangkitkan <i>Virtual Account</i> , dan memperbarui status pesanan menjadi lunas secara instan tanpa verifikasi manual.
8	Ulasan Pelanggan	<i>Client</i> dapat memberikan penilaian dan ulasan terhadap layanan yang telah selesai, yang kemudian akan ditampilkan untuk membangun kredibilitas penyedia jasa.
9	Laporan Keuangan	Sistem menyediakan rekapitulasi pendapatan otomatis bagi MUA berdasarkan transaksi pemesanan yang telah berstatus selesai.
10	Manajemen Kalender	Sistem menyediakan antarmuka kalender interaktif bagi MUA untuk mengatur ketersediaan waktu operasional dan menutup (<i>block</i>) tanggal tertentu secara mandiri.
11	Reschedule Jadwal	Sistem menyediakan antarmuka kalender interaktif bagi MUA untuk mengatur ketersediaan waktu operasional dan menutup (<i>block</i>) tanggal tertentu secara mandiri.

Tabel 1 menyajikan kebutuhan fungsional sistem yang diperoleh dari hasil analisis kebutuhan pengguna. Kebutuhan tersebut meliputi autentikasi akun, manajemen layanan, manajemen portofolio, pemesanan jadwal, perhitungan biaya transportasi otomatis, pembayaran online, ulasan pelanggan, manajemen kalender, reschedule jadwal, dan laporan keuangan. Kebutuhan fungsional ini menjadi dasar dalam proses perancangan dan implementasi sistem.

Selain kebutuhan fungsional, sistem juga memerlukan kebutuhan non-fungsional untuk mendukung kinerja dan aksesibilitas aplikasi. Kebutuhan non-fungsional mencakup perangkat keras, perangkat lunak, koneksi internet, serta kompatibilitas browser yang digunakan oleh pengembang dan pengguna.

Tabel 2 Kebutuhan Non-Fungsional Pengembang

No	Nama Perangkat	Spesifikasi
1	Laptop	- Os Windows - Prosesor AMD Ryzen 7 - RAM 11 GB
2	Visual Studio Code	1.107
3	Next.js	V15.5.6
4	Tailwind CSS	V4.1.18
5	Supabase (PostgreSQL)	18.1
6	Google Chrome	143.0.7499.169

Tabel 3 Kebutuhan Non-Fungsional Pengguna

No	Nama Perangkat	Spesifikasi
1	Laptop	- Os : Windows - Intel i3 - Ram 8GB
2	Smartphone	- Os : Android/iOS - Ram : 4GB
3	Internet	2 Mb/s
4	Google Chrome	143.0.7499.169

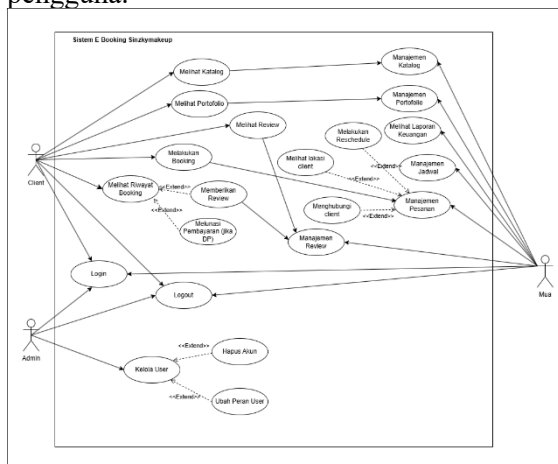
Tabel 2 dan 3 menjelaskan kebutuhan non-fungsional sistem yang harus dipenuhi agar aplikasi dapat berjalan dengan baik. Dari sisi pengembang, sistem membutuhkan perangkat yang mendukung proses coding, pengujian, dan integrasi layanan eksternal. Dari sisi pengguna, sistem harus dapat diakses melalui laptop

maupun smartphone dengan koneksi internet yang stabil. Kebutuhan non-fungsional ini penting untuk memastikan sistem dapat digunakan secara optimal pada berbagai perangkat.

3.2 Perancangan Sistem

Tahap perancangan sistem dilakukan untuk menggambarkan alur kerja, struktur data, dan interaksi antar pengguna sebelum sistem diimplementasikan. Perancangan sistem dilakukan menggunakan Unified Modeling Language dan perancangan basis data. Pada tahap ini, aktor yang terlibat dalam sistem terdiri dari *Client*, MUA, dan Admin. Perancangan sistem diperlukan agar kebutuhan pengguna dapat diterjemahkan ke dalam model sistem yang lebih terstruktur [16], [17].

Use case diagram digunakan untuk menggambarkan hubungan antara aktor dengan fitur sistem. Diagram ini membantu memperjelas batasan hak akses masing-masing pengguna.

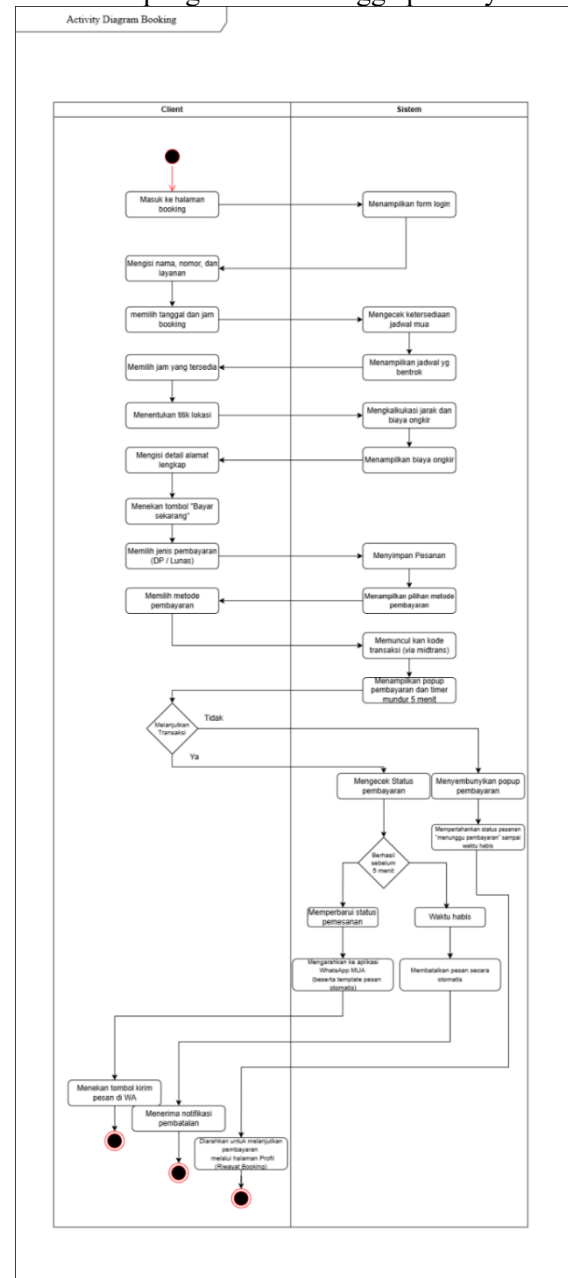


Gambar 1 Use Case Diagram Sistem E-Booking Sinzky Makeup

Berdasarkan Gambar 1, sistem e-booking jasa Makeup Artist melibatkan tiga aktor utama, yaitu *Client*, MUA, dan Admin. *Client* memiliki hak akses untuk melihat katalog layanan, melihat portofolio, melihat ulasan, melakukan pemesanan, melakukan pembayaran, melihat riwayat booking, dan memberikan ulasan setelah layanan selesai digunakan. Aktor MUA berperan sebagai pengelola utama layanan yang dapat mengatur katalog, portofolio, jadwal ketersediaan, pesanan masuk, ulasan, serta laporan keuangan. Sementara itu, Admin memiliki kewenangan untuk mengelola data pengguna dan hak akses pada sistem. Use case

diagram ini menunjukkan bahwa sistem dirancang dengan pembagian hak akses yang jelas sesuai dengan peran masing-masing pengguna.

Selain use case diagram, activity diagram digunakan untuk menggambarkan alur aktivitas sistem, khususnya pada proses booking dan pembayaran. Diagram ini menunjukkan urutan aktivitas yang dilakukan oleh *Client* dan sistem mulai dari pengisian data hingga pembayaran.

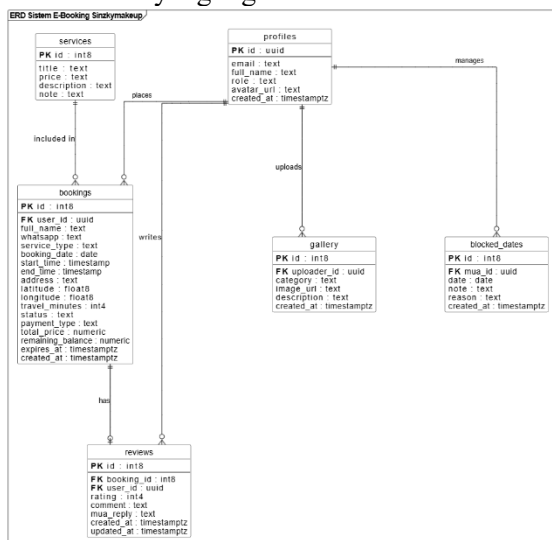


Gambar 2 Activity Diagram Melakukan Booking & Pembayaran

Gambar 2 menjelaskan alur aktivitas pemesanan dan pembayaran yang dilakukan

oleh *Client*. Proses dimulai ketika *Client* memilih layanan Makeup Artist, menentukan tanggal acara, mengisi data pemesanan, dan memasukkan lokasi acara. Setelah lokasi dimasukkan, sistem mengambil titik koordinat lokasi *Client* dan mengirimkannya ke *OpenRouteService* untuk memperoleh jarak tempuh. Data jarak tersebut kemudian diproses oleh sistem untuk menghitung biaya transportasi berdasarkan tarif yang telah ditentukan. Setelah total biaya ditampilkan, *Client* dapat memilih metode pembayaran dan sistem meneruskan transaksi ke *Midtrans*. Jika pembayaran berhasil, status pesanan diperbarui secara otomatis dan jadwal yang dipilih tercatat sebagai pesanan aktif.

Perancangan basis data dilakukan menggunakan Entity Relationship Diagram. ERD digunakan untuk menggambarkan relasi antar entitas yang digunakan dalam sistem.



Gambar 3 Entity Relationship Diagram

Gambar 3 menunjukkan rancangan hubungan antar entitas dalam basis data sistem e-booking jasa Makeup Artist. Entitas utama dalam sistem meliputi profiles, services, bookings, gallery, reviews, dan blocked_dates. Entitas profiles digunakan untuk menyimpan data pengguna, baik *Client*, MUA, maupun Admin. Entitas services menyimpan data layanan yang dapat dipesan oleh *Client*. Entitas bookings menjadi pusat transaksi karena menyimpan data pemesanan, jadwal, lokasi, biaya, dan status pembayaran. Entitas reviews digunakan untuk menyimpan ulasan pelanggan, sedangkan blocked_dates digunakan untuk mencatat tanggal yang tidak tersedia.

Perancangan ERD ini bertujuan menjaga keterhubungan data agar proses pemesanan, pembayaran, dan pengelolaan jadwal dapat berjalan secara terstruktur.

3.3 Implementasi Sistem

Sistem dikembangkan dalam bentuk aplikasi web menggunakan Next.js sebagai framework utama. Next.js digunakan karena mendukung pengembangan aplikasi web modern berbasis React serta dapat menangani kebutuhan frontend dan backend dalam satu kerangka kerja [16]. Supabase digunakan sebagai layanan basis data berbasis PostgreSQL serta mendukung autentikasi pengguna dan penyimpanan data [17]. Sistem juga menggunakan integrasi *OpenRouteService* dan *Midtrans* untuk mendukung proses bisnis utama [13], [15].

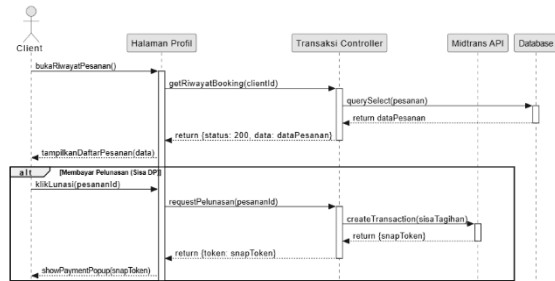
Implementasi *OpenRouteService* dilakukan pada proses booking. Ketika klien menentukan lokasi acara, sistem mengambil titik koordinat lokasi tersebut, kemudian mengirimkannya ke *OpenRouteService* untuk menghitung jarak dari lokasi MUA menuju lokasi klien. Jarak yang diperoleh kemudian dikalikan dengan tarif transportasi yang telah ditentukan sehingga sistem dapat menampilkan biaya transportasi secara otomatis [13].



Gambar 4 Sequence Diagram Booking, Integrasi *OpenRouteService*, dan *Midtrans*

Gambar 4 menggambarkan alur teknis proses booking yang melibatkan integrasi *OpenRouteService* dan *Midtrans*. Proses dimulai ketika *Client* mengisi data booking dan menentukan lokasi acara. Sistem mengambil koordinat lokasi *Client*, kemudian mengirimkan koordinat tersebut bersama koordinat lokasi MUA ke *OpenRouteService* untuk memperoleh data jarak tempuh. Setelah jarak diperoleh, sistem menghitung biaya transportasi melalui logika program berdasarkan tarif yang telah ditentukan. Selanjutnya, total biaya pemesanan

dikirimkan ke *Midtrans* untuk menghasilkan instruksi pembayaran. Alur ini menunjukkan bahwa *OpenRouteService* digunakan untuk memperoleh data jarak, sedangkan perhitungan biaya dan proses pembayaran dikelola oleh sistem.



Gambar 5 Sequence Diagram Pelunasan Pembayaran Menggunakan *Midtrans*

Gambar 5 menjelaskan alur pelunasan pembayaran menggunakan *Midtrans*. Proses dimulai ketika *Client* membuka riwayat booking dan memilih pesanan yang masih memiliki sisa tagihan. Sistem kemudian mengambil data pesanan dari basis data dan membuat transaksi pelunasan melalui *Midtrans*. Setelah *Midtrans* menghasilkan instruksi pembayaran, *Client* dapat menyelesaikan pelunasan sesuai metode yang tersedia. Apabila pembayaran berhasil, sistem menerima respons transaksi dan memperbarui status pembayaran menjadi lunas pada basis data. Alur ini menunjukkan bahwa *Midtrans* tidak hanya digunakan pada pembayaran awal, tetapi juga pada proses pelunasan transaksi.

3.4 Pengujian Sistem

Pengujian sistem dilakukan menggunakan dua metode, yaitu Blackbox Testing dan User Acceptance Testing. Blackbox Testing digunakan untuk menguji fungsi-fungsi utama sistem tanpa melihat struktur kode program. Pengujian ini dilakukan terhadap fitur registrasi, login, booking, perhitungan biaya transportasi, pembayaran, manajemen jadwal, manajemen pesanan, dan ulasan [18].

Tabel 4 Skenario Pengujian Blackbox Testing

No	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Status
1	Registrasi	Pengguna mengisi data registrasi secara lengkap dan valid	Sistem menyimpan data akun baru dan mengarahkan pengguna ke halaman login	Sesuai
2	Login	Pengguna masuk menggunakan email dan password yang valid	Sistem memvalidasi akun dan mengarahkan pengguna sesuai role, yaitu <i>Client</i> , MUA, atau Admin	Sesuai
3	Booking	<i>Client</i> mengisi formulir	Sistem menyimpan data booking dan	Sesuai

		pemesanan secara lengkap	menampilkan rincian pesanan	
4	Validasi Jadwal	<i>Client</i> memilih tanggal dan slot waktu yang sudah terisi	Sistem menolak pemesanan dan menampilkan peringatan bahwa jadwal tidak tersedia	Sesuai
5	<i>OpenRouteService</i>	<i>Client</i> memilih lokasi acara menggunakan marker atau input lokasi	Sistem menghitung jarak lokasi dan menampilkan biaya transportasi secara otomatis	Sesuai
6	<i>Midtrans</i>	<i>Client</i> melanjutkan pembayaran setelah formulir booking selesai	Sistem membuat transaksi pembayaran dan mengarahkan <i>Client</i> ke halaman pembayaran <i>Midtrans</i>	Sesuai
7	Pelunasan Pembayaran	<i>Client</i> melakukan pelunasan pada pesanan yang berstatus DP	Sistem membuat transaksi pelunasan dan memperbarui status pembayaran setelah berhasil	Sesuai
8	Dashboard MUA	MUA melihat daftar pesanan masuk dan menggunakan tombol navigasi atau WhatsApp	Sistem menampilkan pesanan masuk dan membuka akses ke Google Maps atau WhatsApp	Sesuai
9	Manajemen Admin	Admin mengubah role atau menghapus akun pengguna	Sistem memperbarui role atau menghapus akun sesuai aksi yang dilakukan Admin	Sesuai
10	Ulasan	<i>Client</i> mengirimkan ulasan setelah layanan selesai	Sistem menyimpan ulasan dan menampilkannya pada halaman ulasan atau beranda	Sesuai

Tabel 4 menyajikan skenario Blackbox Testing yang digunakan untuk menguji fungsionalitas utama sistem. Setiap skenario pengujian disusun berdasarkan input, proses, dan output yang diharapkan. Pengujian ini bertujuan memastikan bahwa setiap fitur berjalan sesuai dengan kebutuhan yang telah ditentukan.

User Acceptance Testing digunakan untuk mengetahui tingkat penerimaan pengguna terhadap sistem. Responden terdiri dari calon klien, MUA, dan admin. Penilaian dilakukan berdasarkan aspek fungsionalitas, kemudahan penggunaan, efisiensi, dan keandalan sistem [19].

Tabel 5 Skala Penilaian User Acceptance Testing

No	Keterangan	Skor
1	Sangat Tidak Setuju	1
2	Tidak Setuju	2
3	Netral	3
4	Setuju	4
5	Sangat Setuju	5

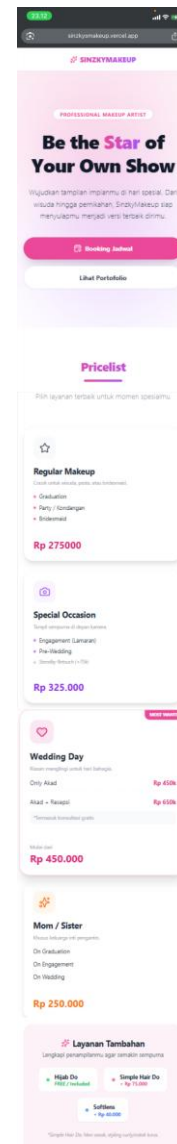
Tabel 5 menjelaskan skala penilaian yang digunakan dalam User Acceptance Testing. Skala ini digunakan untuk mengukur persepsi pengguna terhadap sistem berdasarkan beberapa aspek penilaian. Hasil dari penilaian ini digunakan untuk mengetahui apakah sistem telah sesuai dengan kebutuhan pengguna akhir.

4. HASIL DAN PEMBAHASAN

4.1 Hasil Implementasi Sistem

Hasil dari penelitian ini adalah sistem e-booking jasa Makeup Artist berbasis web yang dapat digunakan oleh klien untuk melakukan pemesanan layanan secara mandiri. Sistem ini juga menyediakan fitur manajemen operasional bagi MUA dan fitur manajemen pengguna bagi admin. Pengembangan sistem berbasis web dipilih karena mampu menyediakan akses layanan yang fleksibel melalui berbagai perangkat dan mendukung pengelolaan data secara terpusat [2], [4], [16].

Halaman utama sistem menampilkan informasi layanan, daftar harga, portofolio, dan akses menuju halaman booking. Halaman ini dirancang sebagai pintu masuk utama bagi pengguna yang ingin mengetahui informasi layanan sebelum melakukan pemesanan.





Gambar 6 Halaman Beranda

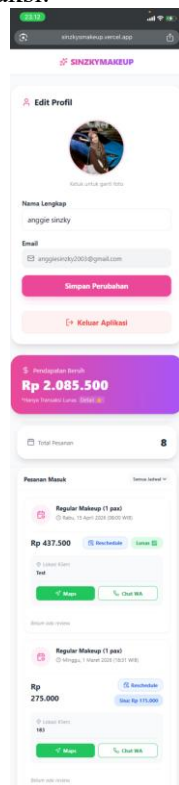
Gambar 6 menampilkan halaman beranda sistem e-booking jasa Makeup Artist. Halaman ini dirancang sebagai halaman utama yang memberikan informasi awal kepada pengguna mengenai layanan yang tersedia. Pada halaman beranda, pengguna dapat melihat informasi singkat mengenai jasa Makeup Artist, daftar layanan atau pricelist, serta akses menuju halaman booking dan portofolio. Desain halaman beranda dibuat untuk memudahkan calon *Client* memahami layanan yang ditawarkan sebelum melakukan pemesanan. Halaman ini juga berfungsi sebagai pintu masuk utama menuju fitur-fitur penting dalam sistem. Setelah pengguna memahami informasi layanan, pengguna dapat melakukan pemesanan melalui halaman booking. Halaman booking menjadi bagian utama dari sistem karena memuat alur pemesanan, pemilihan jadwal, penentuan lokasi, perhitungan biaya, dan pembayaran.



Gambar 7 Halaman Booking

Gambar 7 menunjukkan tampilan halaman booking yang digunakan oleh *Client* untuk melakukan pemesanan layanan. Pada halaman ini, *Client* dapat mengisi data diri, memilih paket layanan, menentukan tanggal acara, memasukkan lokasi, serta melihat rincian biaya. Halaman booking juga menjadi bagian penting dalam integrasi *OpenRouteService* karena lokasi yang dimasukkan oleh *Client* digunakan untuk menghitung biaya transportasi. Setelah seluruh data diisi, sistem akan menampilkan total biaya yang harus dibayar dan mengarahkan *Client* ke proses pembayaran melalui *Midtrans*. Tampilan ini dirancang agar proses pemesanan dapat dilakukan secara mandiri dan mudah dipahami oleh pengguna. Selain fitur untuk *Client*, sistem juga menyediakan dashboard khusus bagi MUA. Dashboard ini digunakan untuk memantau

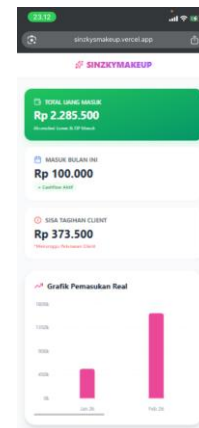
pesanan masuk, mengelola jadwal, dan melihat informasi transaksi.



Gambar 8 Halaman Profil MUA

Gambar 8 memperlihatkan tampilan dashboard MUA yang digunakan untuk mengelola aktivitas operasional. Melalui dashboard ini, MUA dapat melihat daftar pesanan masuk, status pembayaran, detail lokasi *Client*, serta informasi jadwal layanan. Dashboard juga menyediakan akses cepat untuk menghubungi *Client* melalui WhatsApp dan membuka navigasi lokasi melalui aplikasi peta. Dengan adanya dashboard ini, MUA dapat memantau seluruh pesanan secara lebih terpusat dan mengurangi ketergantungan pada pencatatan manual. Hal ini mendukung efisiensi proses administrasi layanan Makeup Artist.

Sistem juga menyediakan halaman laporan keuangan yang digunakan MUA untuk melihat rekapitulasi transaksi. Fitur ini membantu MUA memantau pendapatan dari pesanan yang telah masuk ke dalam sistem.



Gambar 9 Halaman Laporan Keuangan

Gambar 9 menampilkan halaman laporan keuangan yang digunakan oleh MUA untuk memantau pendapatan dari transaksi pemesanan. Halaman ini menyajikan informasi seperti total uang masuk, pemasukan pada periode tertentu, dan sisa tagihan dari *Client* yang belum melunasi pembayaran. Selain itu, halaman laporan juga dapat menampilkan daftar transaksi berdasarkan status pembayaran. Fitur laporan keuangan ini membantu MUA dalam melakukan rekapitulasi pendapatan secara otomatis sehingga proses pencatatan keuangan menjadi lebih praktis dan terorganisir.

4.2 Implementasi *OpenRouteService*

OpenRouteService diimplementasikan untuk memperoleh jarak tempuh antara lokasi MUA dan lokasi klien pada proses pemesanan. Ketika klien menentukan lokasi acara melalui peta, sistem mengambil titik koordinat lokasi tersebut dalam bentuk latitude dan longitude. Titik koordinat ini kemudian digunakan sebagai parameter untuk mengirimkan permintaan rute ke *OpenRouteService*. Penggunaan API dalam sistem web memungkinkan aplikasi memanfaatkan layanan eksternal untuk memperoleh data rute dan jarak tanpa perlu membangun fungsi pemetaan dari awal [11], [12].

Hasil yang diperoleh dari *OpenRouteService* berupa informasi jarak tempuh antara lokasi MUA dan lokasi klien. Data jarak tersebut kemudian diproses oleh sistem melalui logika perhitungan biaya transportasi yang telah ditentukan di dalam program. Dengan demikian, *OpenRouteService* tidak secara langsung menghitung biaya transportasi, melainkan menyediakan data jarak yang

menjadi dasar perhitungan biaya. Perhitungan biaya transportasi dilakukan oleh sistem berdasarkan tarif yang telah ditetapkan. Dengan adanya alur ini, klien dapat mengetahui estimasi biaya transportasi secara langsung sebelum melakukan pembayaran sehingga proses pemesanan menjadi lebih transparan [13].

Berdasarkan Gambar 4, proses perhitungan biaya transportasi diawali ketika klien mengisi data booking dan menentukan titik lokasi acara melalui peta. Sistem mengambil koordinat lokasi klien, kemudian mengirimkannya bersama koordinat lokasi MUA ke *OpenRouteService* untuk memperoleh data jarak tempuh. Setelah jarak diperoleh, sistem menjalankan logika perhitungan biaya transportasi pada kode program berdasarkan jarak dan tarif yang telah ditentukan. Hasil perhitungan tersebut kemudian ditambahkan ke biaya layanan sehingga menghasilkan total biaya pemesanan yang ditampilkan kepada klien.

4.3 Implementasi *Midtrans*

Midtrans diimplementasikan sebagai payment gateway untuk mendukung proses pembayaran pada sistem e-booking. Setelah klien mengisi data pemesanan dan sistem menampilkan total biaya, klien dapat memilih jenis pembayaran, yaitu pembayaran uang muka atau pembayaran penuh. Payment gateway membantu aplikasi berbasis web dalam memproses pembayaran digital dan mengurangi ketergantungan pada validasi manual [1], [14], [15].

Pada proses booking awal, alur pembayaran ditunjukkan pada Gambar 4. Setelah data booking, biaya layanan, dan biaya transportasi selesai dihitung, sistem mengirimkan data transaksi ke *Midtrans*. *Midtrans* kemudian menghasilkan instruksi pembayaran yang dapat digunakan oleh klien untuk menyelesaikan transaksi. Setelah pembayaran berhasil, sistem menerima respons transaksi dan memperbarui status pembayaran pada basis data.

Selain pembayaran awal, sistem juga mendukung proses pelunasan bagi klien yang sebelumnya memilih pembayaran uang muka. Berdasarkan Gambar 5, proses pelunasan dilakukan melalui halaman riwayat booking. Ketika klien memilih pesanan yang masih memiliki sisa tagihan, sistem mengambil data pemesanan dari basis data dan membuat transaksi pelunasan melalui *Midtrans*. Setelah

Midtrans menghasilkan instruksi pembayaran, klien dapat menyelesaikan pelunasan. Apabila pembayaran berhasil, sistem memperbarui status pembayaran menjadi lunas.

Dengan adanya integrasi *Midtrans*, proses validasi pembayaran menjadi lebih cepat karena MUA tidak perlu memeriksa bukti transfer secara manual. Hal ini dapat mengurangi risiko kesalahan verifikasi, mempercepat konfirmasi pesanan, dan memberikan kepastian status transaksi bagi klien maupun MUA.

4.4 Hasil Pengujian Blackbox

Pengujian Blackbox dilakukan untuk memastikan bahwa setiap fitur sistem berjalan sesuai dengan kebutuhan yang telah ditentukan. Metode ini menilai kesesuaian input dan output sistem tanpa melihat struktur kode program [18]. Pengujian dilakukan terhadap fitur utama, seperti registrasi, login, booking, perhitungan biaya transportasi, pembayaran, manajemen jadwal, dan ulasan.

Tabel 6 Hasil Pengujian Fitur Registrasi dan Login

No	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Registrasi	Pengguna mengisi nama lengkap, email, password, dan konfirmasi password dengan data valid	Sistem menyimpan data akun baru dan menampilkan notifikasi registrasi berhasil	Sistem berhasil menyimpan data akun baru dan mengarahkan pengguna ke halaman login	Sesuai
2	Registrasi	Pengguna mengosongkan salah satu kolom pada form registrasi	Sistem menolak proses registrasi dan menampilkan pesan bahwa data wajib diisi	Sistem menampilkan pesan validasi pada kolom yang belum diisi	Sesuai
3	Registrasi	Pengguna mengisi password dan konfirmasi password yang tidak sama	Sistem menolak registrasi dan menampilkan pesan bahwa password tidak sesuai	Sistem menampilkan pesan kesalahan konfirmasi password	Sesuai
4	Login	Pengguna memasukkan email dan password yang valid	Sistem memvalidasi akun dan mengarahkan pengguna ke halaman sesuai role	Sistem berhasil mengarahkan pengguna ke halaman sesuai role, yaitu <i>Client</i> , MUA, atau Admin	Sesuai
5	Login	Pengguna memasukkan email atau password yang salah	Sistem menolak login dan menampilkan pesan kesalahan	Sistem menampilkan pesan bahwa email atau password tidak valid	Sesuai
6	Logout	Pengguna menekan tombol keluar aplikasi	Sistem menghapus sesi login dan mengarahkan pengguna ke halaman login	Sistem berhasil menghapus sesi pengguna dan mengarahkan ke	Sesuai

				halaman login	
--	--	--	--	---------------	--

Tabel 6 menunjukkan hasil pengujian fitur registrasi dan login. Pengujian dilakukan untuk memastikan bahwa sistem dapat menerima input data pengguna, memvalidasi email dan kata sandi, serta membuat sesi login sesuai dengan role pengguna. Berdasarkan hasil pengujian, fitur registrasi dan login dapat berjalan sesuai dengan skenario yang telah ditentukan.

Tabel 7 Hasil Pengujian Fitur Booking

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	<i>Client</i> mengisi formulir pemesanan secara lengkap	Sistem menyimpan data booking dan menampilkan rincian pesanan	Sistem berhasil menyimpan data booking dan menampilkan detail pesanan	Sesuai
2	<i>Client</i> memilih tanggal dan waktu layanan	Sistem memvalidasi ketersediaan jadwal	Sistem berhasil menampilkan status ketersediaan jadwal	Sesuai
3	<i>Client</i> memilih jadwal yang sudah terisi	Sistem menolak pemesanan dan menampilkan peringatan jadwal tidak tersedia	Sistem mendeteksi bentrok jadwal dan menampilkan peringatan	Sesuai
4	<i>Client</i> mengirim formulir pemesanan untuk melanjutkan pembayaran	Sistem memproses data booking dan meneruskan transaksi ke payment gateway	Sistem berhasil memproses booking dan mengarahkan ke proses pembayaran	Sesuai

Tabel 7 menyajikan hasil pengujian fitur booking. Pengujian ini dilakukan dengan memasukkan data pemesanan, memilih jadwal, dan menyimpan pesanan ke dalam sistem. Hasil pengujian menunjukkan bahwa sistem mampu memproses booking secara mandiri oleh *Client* dan menyimpan data pesanan dengan status yang sesuai. Selain itu, sistem dapat membantu mengurangi risiko double-booking melalui pengelolaan jadwal yang terpusat.

Tabel 8 Hasil Pengujian Fitur Perhitungan Biaya Transportasi

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	<i>Client</i> menentukan lokasi acara melalui peta	Sistem mengambil titik koordinat lokasi <i>Client</i>	Sistem berhasil mengambil data latitude dan longitude dari lokasi yang dipilih	Sesuai
2	Sistem mengirim koordinat lokasi <i>Client</i> dan lokasi MUA ke <i>OpenRouteService</i>	<i>OpenRouteService</i> mengembalikan data jarak tempuh	Sistem berhasil memperoleh data jarak tempuh berdasarkan koordinat lokasi	Sesuai
3	Sistem menerima data jarak dari <i>OpenRouteService</i>	Sistem menghitung biaya transportasi melalui logika program	Sistem berhasil menghitung biaya transportasi berdasarkan jarak dan tarif yang telah ditentukan	Sesuai

4	Sistem menampilkan total biaya pemesanan	Total biaya terdiri dari biaya layanan dan biaya transportasi	Sistem berhasil menampilkan total biaya pemesanan kepada <i>Client</i>	Sesuai
---	--	---	--	--------

Tabel 8 menjelaskan hasil pengujian fitur perhitungan biaya transportasi. Pengujian dilakukan dengan memasukkan lokasi *Client* dan menghitung jarak menggunakan *OpenRouteService*. Hasil pengujian menunjukkan bahwa sistem dapat memperoleh data jarak dan menghitung biaya transportasi berdasarkan tarif yang telah ditentukan. Fitur ini berjalan sesuai kebutuhan karena mampu menampilkan biaya transportasi secara otomatis kepada *Client*.

Tabel 9 Hasil Pengujian Fitur Pembayaran

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	<i>Client</i> memilih metode pembayaran setelah booking	Sistem membuat transaksi pembayaran melalui <i>Midtrans</i>	Sistem berhasil membuat transaksi pembayaran	Sesuai
2	<i>Client</i> melanjutkan pembayaran melalui instruksi <i>Midtrans</i>	<i>Midtrans</i> menampilkan instruksi pembayaran	Sistem berhasil mengarahkan <i>Client</i> ke halaman atau instruksi pembayaran	Sesuai
3	<i>Client</i> melakukan pembayaran penuh	Sistem memperbarui status pembayaran menjadi lunas	Sistem berhasil memperbarui status pembayaran setelah transaksi berhasil	Sesuai
4	<i>Client</i> melakukan pembayaran uang muka atau DP	Sistem mencatat pembayaran awal dan menyimpan sisa tagihan	Sistem berhasil menyimpan status pembayaran DP dan sisa tagihan	Sesuai
5	<i>Client</i> melakukan pelunasan dari riwayat booking	Sistem membuat transaksi pelunasan dan memperbarui status pembayaran	Sistem berhasil memproses pelunasan dan memperbarui status menjadi lunas	Sesuai

Tabel 9 menunjukkan hasil pengujian fitur pembayaran menggunakan *Midtrans*. Pengujian dilakukan dengan membuat transaksi berdasarkan data booking dan total biaya yang telah dihitung. Hasil pengujian menunjukkan bahwa sistem dapat mengirimkan data transaksi ke *Midtrans*, menghasilkan instruksi pembayaran, dan memperbarui status pembayaran berdasarkan respons sistem. Dengan demikian, fitur pembayaran dapat mendukung proses validasi transaksi secara otomatis.

Berdasarkan seluruh hasil pengujian Blackbox, fitur utama sistem dapat berjalan sesuai dengan skenario yang dirancang. Sistem mampu memvalidasi input pengguna, menyimpan data booking, menghitung biaya transportasi, menghasilkan transaksi pembayaran, dan memperbarui status pesanan.

4.5 Hasil User Acceptance Testing

User Acceptance Testing dilakukan untuk mengetahui tingkat penerimaan pengguna terhadap sistem. UAT digunakan untuk menilai apakah sistem telah memenuhi kebutuhan pengguna akhir dari sisi fungsionalitas, kemudahan penggunaan, efisiensi, dan keandalan [19]. Pengujian melibatkan tiga jenis pengguna, yaitu calon *Client*, MUA, dan Admin.

Tabel 10 Hasil UAT Responden Calon *Client*

No	Aspek Penilaian	Indikator Penilaian	Hasil Penilaian	Kategori
1	Fungsionalitas	Sistem membantu <i>Client</i> melihat layanan, memilih jadwal, melakukan booking, menghitung biaya transportasi, dan melakukan pembayaran	Sangat baik	Sangat Layak
2	Efisiensi	Sistem mempercepat proses pemesanan dibandingkan pemesanan manual	Sangat baik	Sangat Layak
3	Usability	Tampilan dan alur pengisian form mudah dipahami oleh pengguna	Sangat baik	Sangat Layak
4	Keandalan	Sistem dapat menampilkan informasi booking dan riwayat pesanan dengan baik	Sangat baik	Sangat Layak
5	Rekapitulasi	Total skor 947 dari skor maksimum 1080	98,63%	Sangat Layak

Tabel 10 menunjukkan hasil User Acceptance Testing dari responden calon *Client*. Berdasarkan hasil penilaian, sistem dinilai membantu *Client* dalam melihat informasi layanan, memilih jadwal, mengetahui total biaya, dan melakukan pembayaran secara mandiri. Hal ini menunjukkan bahwa sistem dapat meningkatkan kemudahan akses layanan bagi pelanggan.

Tabel 11 Hasil UAT Responden MUA

No	Aspek Penilaian	Indikator Penilaian	Hasil Penilaian	Kategori
1	Fungsionalitas	Sistem membantu MUA memantau pesanan, rincian tagihan, layanan, dan portofolio	Sangat baik	Sangat Layak
2	Keandalan	Sistem mendukung laporan keuangan dan validasi perubahan jadwal	Sangat baik	Sangat Layak
3	Usability	Sistem memudahkan MUA mengelola kalender dan operasional layanan	Sangat baik	Sangat Layak
4	Rekapitulasi	Total skor 35 dari skor maksimum 35	100%	Sangat Layak

Tabel 11 menyajikan hasil User Acceptance Testing dari responden MUA. Berdasarkan hasil penilaian, sistem dinilai dapat membantu mengurangi pekerjaan administratif, terutama dalam pengelolaan jadwal, pemantauan pesanan, perhitungan biaya transportasi, dan

pengecekan pembayaran. Dengan demikian, sistem dapat mendukung peningkatan efisiensi operasional MUA.

Tabel 12 Hasil UAT Responden Admin

No	Aspek Penilaian	Indikator Penilaian	Hasil Penilaian	Kategori
1	Fungsionalitas	Sistem dapat menampilkan dan membaca data pengguna	Sangat baik	Sangat Layak
2	Keandalan	Sistem dapat memperbarui role pengguna secara akurat	Sangat baik	Sangat Layak
3	Keandalan	Sistem dapat menghapus akun tanpa mengganggu integritas data	Sangat baik	Sangat Layak
4	Fungsionalitas	Sistem membatasi hak akses Admin agar tidak melihat data transaksi sensitif	Sangat baik	Sangat Layak
5	Rekapitulasi	Total skor 20 dari skor maksimum 20	100%	Sangat Layak

Tabel 12 menjelaskan hasil User Acceptance Testing dari responden Admin. Hasil penilaian menunjukkan bahwa fitur manajemen pengguna dapat digunakan untuk melihat daftar akun, mengubah role, dan menghapus pengguna yang tidak diperlukan. Fitur ini mendukung pengelolaan sistem agar tetap terstruktur dan sesuai dengan pembagian hak akses.

Berdasarkan hasil UAT secara keseluruhan, sistem dinilai dapat diterima oleh pengguna karena mampu membantu proses pemesanan, meningkatkan transparansi biaya, mempercepat konfirmasi pembayaran, dan mengurangi pekerjaan administrasi manual.

4.6 Pembahasan

Hasil penelitian menunjukkan bahwa integrasi *OpenRouteService* dan *Midtrans* dapat meningkatkan kualitas sistem e-booking jasa Makeup Artist. *OpenRouteService* memberikan solusi terhadap masalah perhitungan biaya transportasi yang sebelumnya dilakukan secara manual. Dengan adanya perhitungan otomatis, sistem dapat memberikan estimasi biaya yang lebih transparan kepada *Client* [13].

Midtrans memberikan solusi terhadap proses pembayaran yang sebelumnya masih bergantung pada transfer manual dan verifikasi bukti pembayaran. Dengan payment gateway, proses pembayaran menjadi lebih terstruktur dan status pembayaran dapat diperbarui secara otomatis. Temuan ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa integrasi payment gateway pada aplikasi berbasis web dapat meningkatkan efisiensi transaksi dan mendukung pembayaran non-tunai [1], [14], [15].

Selain itu, sistem e-booking yang dikembangkan juga membantu mengurangi risiko double-booking melalui pengelolaan jadwal berbasis sistem. *Client* dapat melakukan pemesanan secara mandiri, sedangkan MUA dapat memantau jadwal dan pesanan melalui dashboard. Dengan demikian, sistem ini tidak hanya berfungsi sebagai media pemesanan, tetapi juga sebagai alat bantu manajemen operasional jasa MUA [2], [3], [10].

Dibandingkan dengan proses manual, sistem yang dikembangkan memberikan beberapa keunggulan, yaitu proses pemesanan lebih cepat, biaya transportasi dapat dihitung otomatis berdasarkan lokasi, pembayaran dapat diproses secara online, MUA dapat mengelola jadwal dan pesanan secara terpusat, serta risiko kesalahan pencatatan dapat dikurangi [2], [7], [10].

Namun, sistem ini juga memiliki keterbatasan. Integrasi pembayaran pada penelitian ini masih menggunakan mode sandbox, sehingga belum memproses transaksi keuangan secara nyata. Selain itu, sistem masih berbasis web dan belum dikembangkan dalam bentuk aplikasi mobile native. Oleh karena itu, pengembangan berikutnya dapat diarahkan pada integrasi mode production, pengembangan aplikasi mobile, dan penambahan notifikasi otomatis.

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Sistem e-booking jasa Makeup Artist berbasis web berhasil dikembangkan untuk mendukung proses pemesanan layanan, pengelolaan jadwal, pengelolaan pesanan, perhitungan biaya transportasi, dan pembayaran online secara terintegrasi.
2. *OpenRouteService* berhasil diimplementasikan untuk memperoleh data jarak tempuh antara lokasi MUA dan lokasi *Client* berdasarkan koordinat lokasi. Data jarak tersebut kemudian digunakan oleh sistem sebagai dasar perhitungan biaya transportasi melalui logika program, sehingga proses perhitungan biaya menjadi lebih cepat, konsisten, dan transparan.
3. *Midtrans* berhasil diimplementasikan sebagai payment gateway untuk mendukung proses pembayaran online. Sistem dapat membuat transaksi pembayaran,

menampilkan instruksi pembayaran, dan memperbarui status pembayaran berdasarkan respons transaksi.

4. Hasil pengujian Blackbox menunjukkan bahwa fitur utama sistem dapat berjalan sesuai dengan skenario yang telah dirancang, meliputi fitur registrasi, login, booking, perhitungan biaya transportasi, pembayaran, manajemen jadwal, dan pengelolaan pesanan.
5. Hasil User Acceptance Testing menunjukkan bahwa sistem dapat diterima oleh pengguna karena mampu mempermudah proses pemesanan, meningkatkan transparansi biaya, dan membantu efisiensi operasional layanan Makeup Artist.
6. Kelebihan sistem ini terletak pada integrasi *OpenRouteService* dan *Midtrans* dalam satu alur booking. Adapun keterbatasannya adalah penggunaan *Midtrans* masih berada pada mode sandbox dan sistem belum tersedia dalam bentuk aplikasi mobile native.
7. Pengembangan selanjutnya dapat dilakukan dengan mengaktifkan mode production pada payment gateway, menambahkan notifikasi otomatis, mengembangkan aplikasi mobile, serta menambahkan fitur pelacakan status layanan secara real-time.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang yang telah memberikan dukungan dalam proses penyusunan penelitian ini. Penulis juga mengucapkan terima kasih kepada Bapak Arip Solehudin, M.Kom. selaku dosen pembimbing I dan Bapak Agung Susilo Yuda Irawan, M.Kom. selaku dosen pembimbing II yang telah memberikan arahan, masukan, dan bimbingan selama proses penelitian. Ucapan terima kasih juga disampaikan kepada pihak *Sinzkymakeup* yang telah memberikan kesempatan, informasi, dan data yang dibutuhkan dalam pengembangan sistem e-booking jasa Makeup Artist berbasis web ini.

DAFTAR PUSTAKA

- [1] M. I. Surya Pratama, "Integrasi Payment Gateway pada Aplikasi Point of Sales Berbasis Website Menggunakan Framework ReactJS

- (Studi Kasus: Toko Adida Pratama),” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3, 2025, doi: 10.23960/jitet.v13i3.7099.
- [2] R. Bilah, S. R. Andani, and I. Gunawan, “Perancangan Sistem Informasi Pemesanan Jasa Make-Up Artist Berbasis Web,” *Jurnal Teknologi dan Sistem Informasi*, 2022.
 - [3] C. Agusniar, S. Siregar, and A. Rasyid, “Sistem Booking Makeup Artist Berbasis Web Menggunakan Algoritma SHA-256,” *Jurnal Sistem Informasi*, 2024.
 - [4] N. Pratiwi and P. Widodo, “Aplikasi Sistem Informasi Pemesanan Jasa Fotografi Berbasis Web,” *Jurnal Teknologi Informasi dan Komputer*, 2023.
 - [5] A. Lestari and R. Wahyudi, “Social Media Marketing Jasa Make Up Artist dalam Meningkatkan Minat Konsumen,” *Jurnal Komunikasi dan Bisnis*, 2023.
 - [6] Rachmah, A. Putri, and F. Rahman, “Strategi Pemasaran Jasa Makeup Melalui Media Sosial,” *Jurnal Ekonomi Kreatif dan Digital*, 2024.
 - [7] H. Patasik, A. Nirwana, and R. Layuk, “Rancang Bangun Aplikasi Pencarian dan Pemesanan Jasa Makeup Artist Berbasis Web Service,” *Jurnal Ilmiah Teknologi Informasi*, 2022.
 - [8] S. Faadhilla and D. Kurniadi, “Sistem Informasi Penyedia Jasa Make-Up Berbasis Web,” *Jurnal Algoritma*, 2023.
 - [9] P. B. Suta, L. Santiari, and W. C. A. Pratami, “Sistem Informasi Pemesanan Jasa Make-Up Artist Salon Komala Santika Berbasis Web,” *Jurnal Sistem dan Informatika*, 2024.
 - [10] Wadda, A. Ayuningtyas, and P. Arum, “Pengembangan Sistem Pemesanan Online Makeup Artist Berbasis Website,” *Jurnal Teknologi Informasi*, 2024.
 - [11] M. Muharom, A. Prasetyo, and R. Nugroho, “Pemanfaatan Application Programming Interface pada Sistem Informasi Berbasis Web,” *Jurnal Informatika*, 2022.
 - [12] A. Falah and O. Komarudin, “Implementasi RESTful API pada Pengembangan Sistem Informasi,” *Jurnal Teknologi dan Rekayasa Perangkat Lunak*, 2023.
 - [13] Widyantoro and Prayogo, “Pemanfaatan Layanan Pemetaan Digital dalam Perhitungan Rute dan Jarak,” *Jurnal Geospasial Indonesia*, 2023.
 - [14] Maulana, A. Hidayat, and R. Sari, “Implementasi Payment Gateway pada Aplikasi Transaksi Online,” *Jurnal Sistem Informasi dan Teknologi*, 2022.
 - [15] Nursansiwati and Armiani, “Penerapan Payment Gateway dalam Sistem Pembayaran Digital,” *Jurnal Ekonomi dan Teknologi Digital*, 2022.
 - [16] R. Kamal and H. Gunaryati, “Pengembangan Aplikasi Web Menggunakan Framework Next.js,” *Jurnal Teknologi Informasi*, 2023.
 - [17] Fatikasari and Fargazzi, “Pemanfaatan Supabase sebagai Backend-as-a-Service dalam Pengembangan Aplikasi Web,” *Jurnal Informatika dan Sistem Informasi*, 2025.
 - [18] Wahyudi and Alameka, “Pengujian Perangkat Lunak Menggunakan Metode Black Box Testing,” *Jurnal Rekayasa Perangkat Lunak*, 2023.
 - [19] Budisaputro, Anardani, Riyanto, and Kusdwidjati, “User Acceptance Testing pada Sistem Informasi Berbasis Web,” *Jurnal Sistem Informasi*, 2023.