**SISTEM MANAJEMEN OPERASIONAL DAN REKOMENDASI GAYA RAMBUT PADA WOX’S BARBERSHOP MENGGUNAKAN FRAMEWORK LARAVEL**

**TUGAS AKHIR**

****

**Oleh:**

**I DEWA GEDE AGUNG WAHYU BRAHMANTHA (220030097)**

**INSTITUT TEKNOLOGI DAN BISNIS**

**(ITB) STIKOM BALI**

**2025**

**SISTEM MANAJEMEN OPERASIONAL DAN REKOMENDASI GAYA RAMBUT PADA WOX’S BARBERSHOP MENGGUNAKAN FRAMEWORK LARAVEL**

**DIAJUKAN SEBAGAI SALAH SATU SYARAT UNTUK MENYUSUN TUGAS AKHIR PROGRAM STUDI S1-SISTEM INFORMASI**

****

**Oleh:**

**I DEWA GEDE AGUNG WAHYU BRAHMANTHA (220030097)**

**INSTITUT TEKNOLOGI DAN BISNIS**

**(ITB) STIKOM BALI**

**2025**

# **PERSETUJUAN SEMINAR DAN SIDANG TUGAS AKHIR**

**SISTEM MANAJEMEN OPERASIONAL DAN REKOMENDASI GAYA RAMBUT PADA WOX’S BARBERSHOP MENGGUNAKAN FRAMEWORK LARAVEL**

**Oleh:**

**I DEWA GEDE AGUNG WAHYU BRAHMANTHA (220030097)**

|  |  |  |
| --- | --- | --- |
| **Dosen Pembimbing** | **Tanda Tangan** | **Tanggal** |
| Pande Putu Gede Putra Pertama, S.T., M.T | ………………… | ……………... |
| Ni Wayan Deriani, SE., M.Kom | ………………… | ……………… |

Denpasar, ………………..

Mengetahui,

Ketua Program Studi Sistem Informasi

Pande Putu Gede Putra Pertama, S.T., M.T

*Halaman ini sengaja dikosongkan*

# **PERNYATAAN KEASLIAN**

Saya yang bertanda tangan dibawah ini:

NIM : 220030097

Nama : I Dewa Gede Agung Wahyu Brahmantha

Jenjang Studi : Strata Satu (S1)

Program Studi : Sistem Informasi

Tempat, Tgl. Lahir : Gianyar, 05 Maret 2004

Alamat : Br. Batur

NIK : 5104010503040001

Dengan ini menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang pernah digunakan untuk memperoleh **Sarjana Komputer (S.Kom)** di suatu perguruan Tinggi dan sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam Daftar Pustaka.

Denpasar,…..………………

(I Dewa Gede Agung Wahyu Brahmantha)

*Halaman ini sengaja dikosongkan*

**SISTEM MANAJEMEN OPERASIONAL DAN REKOMENDASI GAYA RAMBUT PADA WOX’S BARBERSHOP MENGGUNAKAN FRAMEWORK LARAVEL**

# **ABSTRAK**

Seiring dengan meningkatnya jumlah pelanggan, Wox's Barbershop menghadapi tantangan dalam pencatatan transaksi, manajemen pemesanan, dan pelaporan keuangan yang efisien dan akurat. Proses manual saat ini kurang efektif, rentan terhadap kesalahan, dan memakan waktu, sehingga berdampak pada kepuasan pelanggan. Selain itu, seringnya pelanggan bertanya tentang gaya rambut yang cocok menunjukkan adanya kebutuhan akan fitur rekomendasi untuk membantu para tukang cukur dalam memberikan saran yang cepat dan konsisten. Oleh karena itu, dibutuhkan sistem manajemen operasional berbasis website, tidak hanya untuk mengelola data pelanggan, karyawan, layanan, pemesanan, dan transaksi secara lebih sistematis dan menghasilkan laporan keuangan, tetapi juga untuk menawarkan rekomendasi gaya rambut yang dipersonalisasi. Penelitian ini bertujuan untuk mengatasi kendala operasional Wox's Barbershop, khususnya dalam hal pemesanan, transaksi, dan keuangan, melalui sebuah sistem yang terintegrasi. Situs web manajemen berbasis Laravel yang dikembangkan diharapkan dapat meningkatkan efisiensi, meningkatkan akurasi data, mempercepat pelaporan keuangan, dan memungkinkan pemantauan dan pengambilan keputusan secara real-time, sehingga mendorong pertumbuhan bisnis yang modern dan kompetitif.

**Kata Kunci**: Barbershop, Website, Laravel.

*Halaman ini sengaja dikosongkan*

**SISTEM MANAJEMEN OPERASIONAL DAN REKOMENDASI GAYA RAMBUT PADA WOX’S BARBERSHOP MENGGUNAKAN FRAMEWORK LARAVEL**

# ***ABSTRACT***

*As Wox's Barbershop experiences increasing customer numbers, it faces challenges in efficient and accurate transaction recording, booking management, and financial reporting. Current manual processes are less effective, prone to errors, and time-consuming, impacting customer satisfaction. Additionally, frequent customer inquiries about suitable hairstyles highlight a need for a recommendation feature to assist barbers in providing quick and consistent advice. Therefore, an operational management system based on a website is required, not only to manage customer, employee, service, booking, and transaction data more systematically and generate financial reports, but also to offer personalized hairstyle recommendations. This research aims to address Wox's Barbershop's operational constraints, particularly in booking, transactions, and finance, through an integrated system. The developed Laravel-based management website is expected to enhance efficiency, improve data accuracy, accelerate financial reporting, and enable real-time monitoring and decision-making, thereby fostering modern and competitive business growth.*

***Keywords****: Barbershop, Website, Laravel.*

*Halaman ini sengaja dikosongkan*

# **KATA PENGANTAR**

Berkat Rahmat Tuhan Yang Maha Esa, penulis dapat menyelesaikan Tugas Akhir yang berjudul "**SISTEM MANAJEMEN OPERASIONAL DAN REKOMENDASI GAYA RAMBUT PADA WOX’S *BARBERSHOP* MENGGUNAKAN FRAMEWORK LARAVEL**" sesuai dengan waktu yang telah direncanakan.Selanjutnya penulis menyampaikan Terima Kasih kepada :

1. Rektor ITB STIKOM Bali Bapak Dr. Dadang Hermawan
2. Bapak Dr. Roy Rudolf Huizen, ST., MT selaku Wakil Rektor I.
3. Ibu Dr. Ni Luh Putri Srinadi, SE., MM.Kom selaku Wakil Rektor II.
4. Bapak Yudi Agusta, Ph.D. selaku Wakil Rektor III
5. Bapak Dian Pramana S.Kom.,M.Kom selaku Dekan Fakultas Informatika dan Komputer ITB STIKOM BALI.
6. Bapak Pande Putu Gede Putra Pertama, S.T., M.T selaku Kaprodi Sistem Informasi ITB STIKOM Bali dan Dosen Pembimbing Utama yang telah membimbing penulis selama melaksanakan Tugas Akhir.
7. Orang tua beserta keluarga yang selalu memberikan doa, semangat, motivasi dan dukungan penuh dalam pembuatan Tugas Akhir.
8. Sahabat, teman-teman, dan berbagai pihak yang memberikan dukungan dan bantuan kepada penulis untuk menyelesaikan Tugas Akhir.

Semoga hasil penulisan Tugas Akhir ini bermanfaat bagi pihak yang berkepentingan.

Denpasar, April 2025

Penulis

*Halaman ini sengaja dikosongkan*

# **DAFTAR ISI**

[PERSETUJUAN SEMINAR DAN SIDANG TUGAS AKHIR i](#_Toc201756331)

[PERNYATAAN KEASLIAN iii](#_Toc201756332)

[ABSTRAK v](#_Toc201756333)

[*ABSTRACT* vii](#_Toc201756334)

[KATA PENGANTAR ix](#_Toc201756335)

[DAFTAR ISI xi](#_Toc201756336)

[DAFTAR TABEL xiv](#_Toc201756337)

[DAFTAR GAMBAR xvi](#_Toc201756338)

[BAB I PENDAHULUAN 1](#_Toc201756339)

[1.1 Latar Belakang 1](#_Toc201756340)

[1.2 Rumusan Masalah 2](#_Toc201756341)

[1.3 Tujuan Penelitian 3](#_Toc201756342)

[1.4 Manfaat Penelitian 3](#_Toc201756343)

[1.5 Ruang Lingkup 3](#_Toc201756344)

[1.6 Sistematika Penulisan 5](#_Toc201756345)

[BAB II TINJAUAN PUSTAKA 7](#_Toc201756346)

[2.1 *State of The Art* 7](#_Toc201756347)

[2.2 Wox’s *Barbershop* 9](#_Toc201756348)

[2.3 Sistem Informasi Manajemen 10](#_Toc201756349)

[2.4 *Booking* 10](#_Toc201756350)

[2.5 *Website* 10](#_Toc201756351)

[2.6 HTML 10](#_Toc201756352)

[2.7 *CSS* 11](#_Toc201756353)

[2.8 *Tailwind CSS* 11](#_Toc201756354)

[2.9 *JavaScript* 12](#_Toc201756355)

[2.10 *PHP* 13](#_Toc201756356)

[2.11 *Framework Laravel* 13](#_Toc201756357)

[2.12 *Visual Studio Code* 14](#_Toc201756358)

[2.13 *MYSQL* 15](#_Toc201756359)

[2.14 *Flowchart* 15](#_Toc201756360)

[2.15 *Class Diagram* 17](#_Toc201756361)

[2.16 *Entity Relationship Diagram (ERD)* 19](#_Toc201756362)

[2.17 Draw*.oi* 20](#_Toc201756363)

[2.18 *Xampp* 21](#_Toc201756364)

[2.19 Metode *Prototype* 21](#_Toc201756365)

[2.20 *Black Box Testing* 21](#_Toc201756366)

[2.21 SUS (*System* *Usability* *Scale*) 22](#_Toc201756367)

[BAB III METODE PENELITIAN 23](#_Toc201756368)

[, 3.1 Pengumpulan Data 23](#_Toc201756369)

[3.2 Analisis Sistem 24](#_Toc201756370)

[3.3 Perancangan Sistem 25](#_Toc201756371)

[3.4 Pengujian Sistem 25](#_Toc201756372)

[3.5 Penulisan Laporan 26](#_Toc201756373)

[3.6 Arsitektur Sistem 26](#_Toc201756374)

[BAB IV HASIL DAN PEMBAHASAN 29](#_Toc201756375)

[4.1 Hasil Analisis 29](#_Toc201756376)

[4.2 Perancangan Sistem 29](#_Toc201756377)

[4.3 Perancangan Basis Data 29](#_Toc201756378)

[4.4 Perancangan Struktur Framework Laravel 29](#_Toc201756379)

[4.5 Perancangan Antarmuka 29](#_Toc201756380)

[4.5.1 Halaman Untuk User 29](#_Toc201756381)

[4.5.2 Halaman Untuk Admin 29](#_Toc201756382)

[4.5.3 Halaman Untuk Owner 29](#_Toc201756383)

[4.6 Implementasi Sistem 29](#_Toc201756384)

[4.6.1 Halaman Untuk User 29](#_Toc201756385)

[4.6.2 Halaman Untuk Admin 29](#_Toc201756386)

[4.6.3 Halaman Untuk Owner 29](#_Toc201756387)

[4.7 Pengujian Sistem 29](#_Toc201756388)

[4.8 Pengujian User 29](#_Toc201756389)

[BAB V PENUTUP 30](#_Toc201756390)

[5.1 Kesimpulan 30](#_Toc201756391)

[5.2 Penutup 30](#_Toc201756392)

[DAFTAR PUSTAKA 31](#_Toc201756393)

# **DAFTAR TABEL**

[Tabel 2. 1 *State of The Art* 7](#_Toc197504841)

[Tabel 2. 2 Simbol-simbol *Flowchart* 16](#_Toc197504842)

[Tabel 2. 3 *Class Diagram* 18](#_Toc197504843)

[Tabel 2. 4 *Entity Relationship Diagram (ERD)* 19](#_Toc197504844)

*Halaman ini sengaja dikosongkan*

# **DAFTAR GAMBAR**

[Gambar 2. 1 Gambar Wox’s *Barbersho*p 9](#_Toc197334395)

[Gambar 2. 2 Logo *HTML* 11](#_Toc197334396)

[Gambar 2. 3 Logo CSS 11](#_Toc197334396)

[Gambar 2. 4 Logo *Tailwind CSS* 11](#_Toc197334396)

[Gambar 2. 5 Logo *JavaScript* 13](#_Toc197334398)

[Gambar 2. 6 Logo *PHP* 13](#_Toc197334399)

[Gambar 2. 7 Logo *Laravel* 14](#_Toc197334400)

[Gambar 2. 8 Logo Visual Studio Code 14](#_Toc197334401)

[Gambar 2. 9 Logo *MYSQ*L 15](#_Toc197334402)

[Gambar 2. 10 Logo Draw.io 20](#_Toc197334403)

[Gambar 2. 11 Logo Xampp 21](#_Toc197334404)

[Gambar 3. 1 Gambar Metode *Prototype* 23](#_Toc197975886)

[Gambar 3. 2 Gambar Arsitektur Sistem 27](#_Toc197975887)

*Halaman ini sengaja dikosongkan*

# **BAB I** **PENDAHULUAN**

1. **Latar Belakang**

Pelanggan memesan barang atau layanan tertentu di tempat atau bisnis. Dalam kasus *barbershop*, pemesanan berarti pelanggan memilih jadwal tertentu untuk perawatan rambut atau perawatan lainnya. Untuk memastikan layanan diberikan tepat waktu dan tanpa penundaan, pemesanan dapat dilakukan dengan berbagai cara, seperti secara manual melalui telepon atau *WhatsApp*, atau melalui sistem pemesanan *online* yang lebih terorganisir. Dengan pemesanan yang efektif, pelanggan dapat menghindari antrean panjang dan memastikan bahwa layanan diterima sesuai jadwal yang telah ditentukan. Menurut [1] menyatakan bahwa sistem *booking* *online* telah mengalami evolusi signifikan, memungkinkan pelanggan untuk melakukan pemesanan secara mandiri melalui platform digital, yang meningkatkan efisiensi operasional dan pengalaman pelanggan.

Wox's *Barbershop*, yang berlokasi strategis di Jalan Raya Semabaung, Gianyar, telah menjadi destinasi andalan bagi para pria yang ingin tampil rapi dan *stylish* sejak didirikan pada tahun 2023. Meskipun terbilang baru, *barbershop* ini berhasil menarik perhatian berkat komitmennya dalam menghadirkan layanan potong rambut berkualitas tinggi dengan suasana yang nyaman dan ramah. Setiap harinya, antara lima belas hingga dua puluh pelanggan datang untuk menikmati berbagai layanan, mulai dari potongan rambut modern, *styling*, hingga perawatan rambut dan kumis.

Seiring meningkatnya jumlah pelanggan, Wox's *Barbershop* menghadapi tantangan dalam pencatatan transaksi, manajemen pemesanan, dan penyusunan laporan keuangan yang efisien dan akurat. Proses manual yang digunakan selama ini mulai dirasa kurang efektif, rawan kesalahan, dan memakan waktu, sehingga dapat memengaruhi kepuasan pelanggan. Selain itu, berdasarkan hasil wawancara dengan pemilik *barbershop*, banyak pelanggan yang sering menanyakan rekomendasi gaya rambut yang sesuai dengan bentuk kepala dan jenis rambut mereka. Hal ini menunjukkan adanya kebutuhan untuk menghadirkan fitur rekomendasi gaya rambut yang dapat membantu barber dalam memberikan saran yang tepat secara cepat dan konsisten. Oleh karena itu, dibutuhkan sebuah sistem manajemen operasional berbasis *website* yang tidak hanya membantu dalam pengelolaan data pelanggan, karyawan, layanan, pemesanan, transaksi, dan pembuatan laporan keuangan secara lebih terstruktur, juga memerlukan fitur rekomendasi gaya rambut untuk meningkatkan kualitas layanan yang lebih personal dan sesuai dengan keinginan pelanggan. Menurut [2] penggunaan *framework* *Laravel* dapat memaksimalkan pengembangan *website* dan *website* yang di hasilkan lebih dinamis. Sementara itu, menurut [3], sistem informasi manajemen berbasis web mampu mengatasi berbagai masalah transaksi dan laporan keuangan, serta meningkatkan efisiensi dan profesionalitas layanan *barbershop*, karena sistem ini layak digunakan untuk mengelola data manajemen dan transaksi agar dapat tercatat dengan rapi dan baik.

Sistem manajemen informasi berbasis web dipilih karena fleksibel, mudah diakses, dan efisien. Sistem ini memungkinkan pengguna mengelola data kapan saja dan di mana saja selama terhubung ke internet. Penelitian ini bertujuan membantu Wox's *Barbershop* dalam mengatasi kendala operasional, khususnya dalam pengelolaan pemesanan, transaksi, dan keuangan. Dengan sistem terintegrasi, diharapkan efisiensi meningkat, pencatatan lebih akurat, laporan keuangan lebih cepat disusun, serta pengawasan dan pengambilan keputusan dapat dilakukan secara *real-time*. Sistem ini juga mendorong pertumbuhan bisnis yang lebih modern dan kompetitif.

Studi ini menunjukkan bahwa sistem terintegrasi mampu mempermudah penyusunan laporan keuangan, meningkatkan efisiensi operasional, dan akurasi pencatatan. *Website* manajemen yang dikembangkan menggunakan *framework* *Laravel* ini menawarkan antarmuka menarik dan responsif, serta mendukung modul pencatatan, pemesanan, dan rekomendasi. *Laravel* dipilih karena memiliki struktur yang rapi, *sintaks* mudah dipahami, dan fitur lengkap. Sistem ini diharapkan dapat meningkatkan kualitas layanan, pengalaman pelanggan, dan mendukung pertumbuhan bisnis yang lebih modern dan profesional.

## **Rumusan Masalah**

Bagaimana cara membantu Wox’s *Barbershop* dalam mengelola operasional dan memberikan rekomendasi gaya rambut secara lebih efektif melalui sistem berbasis *website* menggunakan *framework* *Laravel*?

1. **Tujuan Penelitian**

Membantu Wox’s *Barbershop* dalam mengelola operasional dan memberikan rekomendasi gaya rambut secara lebih efektif melalui sistem berbasis website menggunakan framework *Laravel*.

1. **Manfaat Penelitian**

Hasil penelitian ini diharapkan dapat membantu Wox's *Barbershop* dalam mengelola operasional harian secara lebih efisien, termasuk pengelolaan data pelanggan, karyawan, layanan, pemesanan, transaksi, dan pembuatan laporan keuangan secara lebih terstruktur. Sistem ini dirancang untuk meminimalkan kesalahan manusia dalam proses pencatatan manual, meningkatkan akurasi data, serta mempercepat proses pelayanan. Selain itu, fitur rekomendasi gaya rambut yang disesuaikan dapat membantu pelanggan yang merasa bingung dalam memilih model potongan rambut yang sesuai.

1. **Ruang Lingkup**

Adapun ruang lingkup penelitian ini adalah sebagai berikut :

1. Penelitian ini dilakukan berdasarkan studi kasus pada Wox’s *Barbershop* sebagai lokasi penelitian.
2. Pengguna sistem ini terdiri dari tiga hak akses, yaitu:
3. Admin
4. Mengelola data pengguna (membaca, menambah, mengubah).
5. Mengelola transaksi serta laporan keuangan dan operasional.
6. Melakukan pemantauan dan pengaturan sistem, termasuk pengaturan keamanan dan *backup.*
7. Pegawai
8. Mengelola layanan pelanggan.
9. Mengelola data pelanggan dan jadwal pelayanan.
10. Memberikan layanan pelanggan, menangani konfirmasi pemesanan, dan mengelola komplain.
11. Pelanggan
12. Melakukan pemesanan *online* untuk mendapatkan layanan *barbershop*.
13. Melihat riwayat transaksi dan pemesanan yang telah dilakukan.
14. Mendapatkan rekomendasi potongan rambut yang sesuai dengan karakteristik pribadi (misalnya, bentuk wajah dan jenis rambut).
15. **Mendapatkan layanan potong rambut gratis** setelah melakukan kunjungan sebanyak 10 kali, sebagai bentuk *loyalty* *reward*.
16. Data utama yang dikelola dalam sistem meliputi data pelanggan, karyawan, layanan, pemesanan, transaksi, dan keuangan.
17. Laporan yang dihasilkan dari sistem manajemen operasional dan rekomendasi gaya rambut ini mencakup laporan data pelanggan, karyawan, layanan, pemesanan, transaksi, dan keuangan, serta laporan tren gaya rambut yang paling diminati berdasarkan pilihan pelanggan dan hasil rekomendasi sistem.
18. Perancangan proses bisnis yang digunakan dalam sistem ini mengacu pada alur kerja yang mengintegrasikan pemesanan *online*, transaksi, dan manajemen pelanggan untuk meningkatkan efisiensi operasional.
19. Perancangan basis data yang digunakan dimaksudkan untuk menyimpan data secara terstruktur. Ini memastikan integrasi antara modul yang mencakup pelanggan, karyawan, layanan, reservasi, transaksi, laporan keuangan, dan sistem rekomendasi gaya rambut. Ini membantu mengelola operasi *barbershop* secara efisien dan terpusat.
20. Sistem *booking* Wox’s *Barbershop* ini menggunakan beberapa teknologi, di antaranya:
21. *Framework* *Larave 10*, untuk pengembangan aplikasi berbasis web.
22. *MySQL*, sebagai sistem manajemen basis data untuk menyimpan informasi pelanggan dan transaksi.
23. *JavaScript* dan *AJAX*, untuk meningkatkan interaktivitas dan responsivitas sistem.
24. *Tailwind* *CSS* dan *CSS*, untuk desain antarmuka pengguna yang *modern* dan fleksibel.
25. Development *tools* yang digunakan dalam penelitian ini adalah *Visual Studio Code*, sebagai *Integrated Development Environment (IDE)* untuk pengembangan dan pengujian sistem.
26. Metode pengujian yang akan digunakan pada sistem ini mencakup:
27. Pengujian *Black Box* – Pengujian sistem dilakukan menggunakan metode *black* *box* testing untuk memastikan semua fitur berjalan dengan baik sesuai dengan fungsionalitas yang dirancang.
28. Metode *System Usability Scale (SUS)* – Menggunakan kuesioner SUS untuk mengukur tingkat kegunaan sistem berdasarkan pengalaman pengguna, sehingga dapat diketahui sejauh mana sistem ini mudah digunakan dan diterima oleh pengguna.
29. **Sistematika Penulisan**

**BAB I PENDAHULUAN**

Bab ini membahas latar belakang penelitian yang menjelaskan urgensi pengembangan sistem, rumusan masalah yang menjadi fokus penelitian, tujuan penelitian yang ingin dicapai, manfaat yang diperoleh dari penelitian ini, ruang lingkup penelitian untuk membatasi cakupan penelitian, serta sistematika penulisan yang memberikan gambaran keseluruhan isi penelitian.

**BAB II TINJAUAN PUSTAKA**

Bab ini memuat kajian pustaka yang mencakup teori-teori pendukung yang relevan dengan penelitian ini, penelitian terdahulu yang menjadi referensi, serta konsep-konsep yang digunakan dalam pengembangan sistem manajemen operasional dan rekomendasi gaya rambut.

**BAB III METODOLOGI PENELITIAN**

Bab ini menjelaskan metode penelitian yang digunakan, mencakup metode pengembangan sistem, tahapan penelitian, analisis kebutuhan sistem, desain sistem (Class Diagram, ERD, dan desain antarmuka), serta metode pengujian yang digunakan untuk mengevaluasi sistem.

**BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi hasil perancangan sistem manajemen operasional dan rekomendasi gaya rambut yang mencakup mencakup *Flowchart*, *Class* *Diagram, ERD* (*Entity Relationship Diagram*), perancangan basis data, desain antarmuka dan pengujian sistem yang dilakukan menggunakan metode yang telah ditentukan.

**BABV KESIMPULAN DAN SARAN**

Bab ini menyajikan kesimpulan dari hasil penelitian yang telah dilakukan, menjelaskan kontribusi sistem terhadap peningkatan layanan dan operasional Wox’s *Barbershop*, serta memberikan saran untuk pengembangan lebih lanjut guna meningkatkan kualitas sistem di masa mendatang

# **BAB II TINJAUAN PUSTAKA**

1. ***State of The Art***

Berikut adalah beberapa penelitian yang digunakan sebagai referensi dalam penelitian ini, dapat dilihat pada tabel 2.1.

Tabel 2. 1 State of The Art

| **No** | **Penulis** | **Judul Penelitian** | **Hasil Penelitian** |
| --- | --- | --- | --- |
| 1 | I Kadek Prasta Dwiutama | Sistem Informasi *Booking* *Barbershop* Bad Punz Bali Menggunakan *Framework* *Laravel* | Penelitian ini menghasilkan sistem informasi *booking* untuk Bad Punz *Barbershop* yang memungkinkan pelanggan memesan layanan potong rambut sesuai jadwal, melihat model rambut, riwayat transaksi, dan produk seperti pomade. Sistem ini juga menampilkan grafik transaksi dan *booking* bulanan untuk membantu pemilik memantau tren, serta mengirim notifikasi *booking* melalui *WhatsApp [2]*. |
| 2 | Setiawan R.Kurniadi D.Saleh M. | Rancang Bangun Aplikasi *Booking* dan Transaksi *Barbershop* Berbasis Web | Penelitian ini menghasilkan sistem yang dibangun dengan sukses memenuhi persyaratan digital untuk operasi *barbershop*. Aplikasi berbasis web ini menggunakan PHP dan *MySQL* dan memiliki antarmuka yang mudah digunakan. Pemesanan antrean *online*, pengelolaan jenis layanan dan harga, pencatatan dan pelaporan transaksi, dan pemberian nomor antrean otomatis adalah fitur utama yang berhasil diterapkan. Dengan sistem ini, pelanggan dapat memesan layanan dengan lebih efisien tanpa harus datang langsung dan menunggu lama, sementara manajer dapat melacak seluruh operasi melalui laporan yang disajikan secara otomatis. Seluruh fungsi berjalan dengan baik sesuai yang dirancang, menurut pengujian metode *black box*. Selain itu, karena aplikasi ini dianggap dapat mempercepat proses *booking* dan meningkatkan efisiensi pelayanan di *barbershop*, hasil uji coba menunjukkan respons pengguna yang positif [4]. |
| 3 | Dihin Muriyatmoko, Shoffin Nahwa Utama, Triana Harmini , Muhammad Fazrin Tammy | Sistem Informasi Manajemen *Barbershop* Unit Usaha Unida Gontor (U3) Berbasis *Website* | Penelitian ini menghasilkan sistem informasi manajemen *barbershop* berbasis *website* menggunakan *Laravel* dan *MySQL*. Sistem ini dirancang untuk menggantikan pencatatan manual di *Barbershop* U3 Gontor, mencakup fitur pencatatan transaksi, inventaris, laporan keuangan, dan manajemen akun dengan hak akses berbeda. Pengujian menggunakan metode *black-box* dan *framework* PIECES menunjukkan sistem berjalan baik dan mendapat respon puas dari pengguna. Sistem ini dinilai layak digunakan karena mampu meningkatkan efisiensi dan ketertiban dalam pengelolaan barbershop [3]. |

Tabel 2.1 menggambarkan rangkaian penelitian sebelumnya dan temuan penelitian ini. Misalnya, sistem informasi booking yang dikembangkan oleh I Kadek Prasta Dwiutama dan Dihin Muriyatmoko et al. berfokus pada pengelolaan layanan dan transaksi di salon secara digital dengan menggunakan *framework* Laravel. Demikian pula, penelitian oleh Setiawan, Kurniadi, dan Saleh berfokus pada kemudahan pelanggan untuk memesan dan melakukan transaksi. Namun, keunggulan utama dari sistem manajemen operasional yang dibuat untuk Wox's *Barbershop* adalah rekomendasi gaya rambut yang disesuaikan dengan jenis rambut dan bentuk kepala pelanggan. Dibandingkan dengan penelitian sebelumnya, fitur ini memberikan nilai tambahan berupa saran personal yang membantu pelanggan memilih gaya rambut yang paling sesuai dengan preferensi dan kebutuhan mereka. Ini membuat sistem ini lebih baik dalam hal personalisasi layanan.

1. **Wox’s *Barbershop***

Wox's *barbershop*, yang berlokasi strategis di Jalan Raya Semabaung, Gianyar, telah menjadi destinasi andalan bagi para pria yang ingin tampil rapi dan *stylish* sejak didirikan pada tahun 2023. Meskipun terbilang baru, *barbershop* ini berhasil menarik perhatian berkat komitmennya dalam menghadirkan layanan potong rambut berkualitas tinggi dengan suasana yang nyaman dan ramah. Setiap harinya, antara lima belas hingga dua puluh pelanggan datang untuk menikmati berbagai layanan, mulai dari potongan rambut modern, *styling*, hingga perawatan rambut dan kumis, dapat dilihat pada gambar 2.1.

****

Gambar 2. 1 Gambar Wox’s Barbershop

1. **Sistem Informasi Manajemen**

Sistem Informasi Manajemen (SIM) merupakan sistem yang menyediakan informasi yang akurat dan tepat waktu yang diperlukan untuk memfasilitasi proses pengambilan keputusan serta memungkinkan organisasi untuk menjalankan fungsi perencanaan, pengendalian, dan operasional secara efektif [5]. Melalui penerapan SIM, organisasi dapat meningkatkan efisiensi dan efektivitas operasionalnya, serta mendorong pengambilan keputusan yang lebih cepat dan tepat berdasarkan informasi yang tersedia

1. ***Booking***

*Booking* atau pemesanan merupakan proses di mana pelanggan mengatur atau menjadwalkan layanan tertentu, seperti reservasi kamar hotel, tiket perjalanan, atau layanan lainnya, melalui sistem yang memungkinkan pengelolaan ketersediaan, harga, dan informasi pelanggan secara efisien. Dalam konteks pariwisata dan perhotelan, sistem *booking* telah berevolusi menjadi platform digital yang memungkinkan pelanggan melakukan reservasi secara *online*, memberikan kenyamanan dan efisiensi dalam proses pemesanan.​

Menurut [1] menyatakan bahwa sistem *booking* *online* telah mengalami evolusi signifikan, memungkinkan pelanggan untuk melakukan reservasi secara mandiri melalui platform digital, yang meningkatkan efisiensi operasional dan pengalaman pelanggan.

1. ***Website***

*Website* merupakan salah satu alat yang digunakan untuk memberikan informasi dan mempromosikan suatu produk agar dikenal dan diketahui oleh masyarakat luas. Dalam era digital, *website* menjadi media penting bagi organisasi, institusi, maupun pelaku usaha untuk menyampaikan informasi secara cepat, efisien, dan dapat diakses kapan saja. Fungsi utama dari *website* tidak hanya sebagai sarana penyampaian informasi, tetapi juga sebagai media interaktif yang memungkinkan terjadinya komunikasi dua arah antara pemilik situs dan pengunjung [6].

1. **HTML**

*HTML (HyperText Markup Language)* adalah bahasa *markup* standar yang digunakan untuk membuat dan menyusun struktur halaman web. *HTML* terdiri dari elemen-elemen yang ditandai dengan *tag*, yang digunakan untuk menentukan berbagai komponen dalam halaman web seperti teks, gambar, tautan, dan elemen multimedia lainnya. Dengan HTML, pengembang web dapat merancang kerangka dasar dari sebuah situs web, yang kemudian dapat diperkaya dengan CSS dan *JavaScript* untuk tampilan dan interaktivitas yang lebih kompleks, dapat dilihat pada gambar 2.2 [7].



Gambar 2. 2 Logo HTML

1. ***CSS***

*CSS (Cascading Style Sheets)* adalah bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, termasuk tata letak, jenis huruf, warna, dan elemen visual lainnya. CSS memungkinkan pemisahan antara konten dan presentasi, sehingga memudahkan pengelolaan dan pemeliharaan desain web. Dengan *CSS*, pengembang dapat mengontrol format tampilan pada halaman *HTML* dan *XHTML*, serta dapat diterapkan pada berbagai dokumen *XML* lainnya, dapat dilihat pada gambar 2.3 [7].

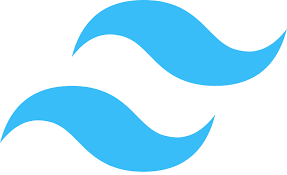


Gambar 2. 3 Logo CSS

1. ***Tailwind CSS***

Dengan menggabungkan kelas-kelas kecil seperti p-4, *text-center*, atau bg-blue-500, *Tailwind* *CSS* memungkinkan pengembang untuk membangun antarmuka pengguna secara cepat dan efisien dengan menggunakan kelas-kelas utilitas langsung di dalam elemen *HTML*. Ini membedakan kerangka kerja *CSS* tradisional yang bergantung pada komponen siap pakai. Metode ini memudahkan proses *styling*, memastikan tampilan yang konsisten, dan membantu mengoptimalkan ukuran *file* *CSS* dengan fitur seperti *compiler* dan *purge* *Just-In-Time* *(JIT).*

Menurut [8], penggunaan *Tailwind CSS* sebagai *framework utility-first* sangat membantu dalam pengembangan *website gallery* karena dapat mempercepat proses desain antarmuka, memberikan tampilan yang responsif, serta memudahkan penyesuaian gaya tanpa menulis *CSS* secara manual, dapat dilihat pada gambar 2.4.



Gambar 2. 4 Logo *Tailwind CSS*

1. ***JavaScript***

*JavaScript* adalah bahasa pemrograman yang digunakan untuk membangun *website* dari sisi klien *(client-side)* dengan tujuan membuat halaman web menjadi lebih dinamis dan interaktif. Melalui *JavaScript*, pengembang dapat menambahkan fitur seperti validasi formulir, animasi, dan interaksi pengguna lainnya yang memungkinkan halaman web merespon *input* pengguna tanpa perlu *refresh* halaman. *JavaScript* bekerja dengan memanipulasi elemen *HTML* dan *CSS* di browser pengguna secara langsung, yang membuat aplikasi web lebih cepat dan responsif. Selain itu, *JavaScript* memungkinkan pengelolaan data di sisi klien, seperti memproses informasi atau mengambil data dari API, tanpa perlu komunikasi dengan server setiap kali ada perubahan.

Menurut [9] menyatakan bahwa *JavaScript* adalah bahasa pemrograman yang memungkinkan pembuatan aplikasi web yang interaktif dan dinamis dengan memanfaatkan skrip yang ditempelkan dalam dokumen *HTML*. *JavaScript* memiliki kemampuan untuk mengelola *event*, memanipulasi *DOM (Document Object Model)*, serta melakukan pemrosesan data di sisi klien, tanpa harus melakukan komunikasi dengan server. Kemampuan ini menjadikan *JavaScript* sebagai bahasa utama dalam pengembangan aplikasi web modern , dapat dilihat pada gambar 2.5.



Gambar 2. 5 Logo JavaScript

1. ***PHP***

*PHP* adalah bahasa pemrograman skrip yang dirancang khusus untuk pengembangan web sisi server. *PHP* memungkinkan pembuatan halaman web dinamis dan interaktif dengan mengolah data dari server dan menghasilkan konten *HTML* yang ditampilkan di browser pengguna. Bahasa ini mendukung berbagai fitur seperti pengolahan formulir, manajemen sesi, dan koneksi ke *database*, menjadikannya pilihan populer dalam pengembangan aplikasi web.​

Menurut [10] menyatakan bahwa, PHP digunakan dalam perancangan dan implementasi *website* layanan akademik di SMA Negeri 07 Bandar Lampung. Dalam penelitian tersebut, *PHP* digunakan bersama dengan *MariaDB* sebagai *database* server untuk membangun sistem informasi akademik berbasis web. Penggunaan *PHP* memungkinkan pengelolaan data akademik secara efisien dan interaktif, serta mempermudah akses informasi bagi pengguna , dapat dilihat pada gambar 2.6.



Gambar 2. 6 Logo PHP

1. ***Framework Laravel***

*Laravel* adalah *framework* PHP berbasis *open-source* yang dirancang untuk membangun aplikasi web modern dengan sintaksis yang elegan dan ekspresif. Dikembangkan oleh Taylor Otwell pada tahun 2011, *Laravel* mengikuti pola arsitektur *Model-View-Controller (MVC)* dan dilengkapi dengan berbagai fitur seperti *Eloquent ORM*, *Blade templating engine*, serta sistem *routing* dan migrasi *database* yang memudahkan pengembangan aplikasi web yang *skalabel* dan *maintainable*.

Menurut [11] menyatakan bahwa penggunaan *Laravel* dalam pengembangan sistem informasi manajemen jasa fotografi berbasis *website* memberikan kemudahan dalam pengelolaan data pemesanan dan pelanggan, serta meningkatkan efisiensi operasional melalui fitur-fitur yang disediakan oleh *framework* tersebut, dapat dilihat pada gambar 2.7.



Gambar 2. 7 Logo Laravel

1. ***Visual Studio Code***

Visual Studio Code (VS Code) adalah editor kode sumber ringan namun kuat yang dikembangkan oleh Microsoft. Dirancang untuk pengembangan aplikasi web dan perangkat lunak, VS Code mendukung berbagai bahasa pemrograman, termasuk *JavaScript*, *Python*, dan PHP. Dengan fitur seperti *IntelliSense* untuk penyelesaian kode otomatis, *debugging* bawaan, kontrol versi Git, dan ekstensibilitas melalui *plugin*, VS Code menjadi salah satu editor yang populer di kalangan pengembang. Menurut [12] penggunaan VS Code sebagai tools utama dalam proses pengembangan sistem karena kemudahan dan kelengkapan fiturnya,dapat dilihat pada gambar 2.8.



Gambar 2. 8 Logo Visual Studio Code

1. ***MYSQL***

*MySQL* adalah sistem manajemen basis data relasional (RDBMS) *open-source* yang menggunakan bahasa *SQL (Structured Query Language)* untuk mengelola dan memanipulasi data. Dikenal karena kecepatan, keandalan, dan kemudahan penggunaannya, *MySQL* banyak digunakan dalam pengembangan aplikasi web dan sistem informasi. Sebagai perangkat lunak *open-source*, *MySQL* dapat digunakan secara gratis dan mendukung berbagai platform sistem operasi.

MySQL atau dibaca *My* *Sequel* adalah sistem manajemen basis data yang sering disingkat sebagai *DBMS*. *MySQL* adalah sistem manajemen basis data open-source. *MySQL* dapat digunakan untuk membuat dan mengelola basis data serta isinya, dari yang terkecil hingga yang sangat besar, dan menyampaikan informasi kepada penggunanya. *MySQL* juga termasuk dalam *RDBMS* atau sistem manajemen basis data relasional, di mana dalam struktur basis datanya, saat proses pengambilan data menggunakan metode basis data relasional dan menjadi penghubung antara perangkat lunak dan server basis data. Hal yang perlu diingat adalah bahwa *MySQL* dapat digunakan secara gratis, dapat dilihat pada gambar 2.9 [13].



Gambar 2. 9 Logo MYSQL

1. ***Flowchart***

Flowchart adalah cara penulisan algoritma dengan menggunakan notasi grafis. Flowchart merupakan gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu.

Dengan menggunakan *flowchart* akan memudahkan untuk melakukan pengecekan bagian-bagian yang terlupakan 4 dalam analisis masalah. *Flowchart* dapat membantu analis dan *programmer* untuk memecahkan masalah kedalam segmen yang lebih kecil dan membantu dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* didefinisikan juga sebagai bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart dapat juga merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut, dapat dilihat pada tabel 2. 2 [14].

Tabel 2. 2 Simbol-simbol Flowchart

| **Nama** | **Lambang** | **Keterangan** |
| --- | --- | --- |
| *Terminator Symbo*l |  | Simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan. |
| *Processing Symbol* |  | Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer. |
| *Input-Output Symbol* |  | Simbol yang menyatakan proses *input* dan *output* tanpa tergantung dengan jenis peralatannya. |
| *Manual Operation Symbol* |  | Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer. |
| Simbol Dokumen |  | Simbol yang menyatakan *input* berasal dari dokumen dalam bentuk kertas atau *output* dicetak ke kertas. |
| *Decision Symbol* |  | Simbol pemilihan proses berdasarkan kondisi yang ada. |
| *Disk and Online Storage Symbol* |  | Simbol yang menyatakan *input* yang berasal dari *disk* atau disimpan ke *disk*. |
| *Manual Input Symbol* |  | Simbol untuk pemasukan data secara manual *online* *keyboard*. |
| *Flow Symbol* |  | Simbol untuk Menunjukkan arus dari suatu proses |
| *Symbol* Penghubung |  | Simbol untuk Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain |

1. ***Class Diagram***

*Class* *diagram* merupakan konteks *Unified* Modeling *Language* (UML) adalah diagram struktur statis yang menggambarkan kelas‑kelas pada suatu sistem, atribut dan operasi setiap kelas, serta hubungan antar kelas seperti asosiasi, generalisasi, agregasi, dan dependensi. Diagram ini digunakan untuk memodelkan struktur konseptual maupun desain rinci, dan sekaligus dapat dijadikan dasar untuk pembuatan kode dalam paradigma berorientasi objek.

Menurut [15], *class* diagram merupakan komponen sentral dari pengembangan perangkat lunak berbasis model (*model*-*driven* *development*) dan digunakan untuk memodelkan *static design view* dari suatu sistem, di mana kelas-kelas sebagai ‘*template’* objek menyimpan data (atribut) dan aksi (metode) yang akan dilaksanakan

Tabel 2. 3 Class Diagram

| **No** | **Simbol** | **Keterangan** |
| --- | --- | --- |
| 1 |  | *Class* : Simbol yang memiliki fungsi sebagai representasi dari entitas dalam sistem. |
| 2 |  | *Nary Association* : Simbol yang memiliki fungsi sebagai upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | Generalization:Simbol yang memiliki fungsi sebagaihubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Collaboration* :Simbol yang memiliki fungsi sebagai deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*. |
| 5 |  | *Realization* : Simbol yang memiliki fungsi sebagai operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency*: Simbol yang memiliki fungsi sebagai hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
| 7 |  | *Association*: Simbol yang memiliki fungsi sebagai Apa yang menghubungkan antara objek satu dengan objek lainnya. |

1. ***Entity Relationship Diagram (ERD)***

*Entity Relationship Diagram (ERD)* adalah representasi grafis yang digunakan untuk memodelkan struktur data dan hubungan antar entitas dalam suatu sistem informasi. *ERD* membantu dalam merancang basis data dengan menggambarkan entitas, atribut, dan relasi yang ada, sehingga memudahkan pemahaman tentang bagaimana data saling berinteraksi. Dengan menggunakan *ERD*, pengembang sistem dapat memastikan bahwa basis data yang dirancang sesuai dengan kebutuhan informasi dan proses bisnis yang ada, dapat dilihat pada tabel 2.4 [16].

Tabel 2. 4 Entity Relationship Diagram (ERD)

| **Notasi** | **Keterangan** |
| --- | --- |
|  | Entitas, sebuah barang atau obyek yang dapat dibedakan dari obyek lain. |
|  | Relasi, asosiasi 2 atau lebih entitas dan berupa kata kerja. |
|  | Atribut, properti yang dimiliki setiap entitas yang akan disimpan datanya. |

1. **Draw*.io***

*Draw.io*, yang kini dikenal sebagai diagrams.net, adalah aplikasi diagram berbasis web yang memungkinkan pengguna untuk membuat berbagai jenis diagram seperti diagram alur, diagram UML, diagram jaringan, dan lainnya. Aplikasi ini bersifat *open-source* dan dapat digunakan secara gratis tanpa memerlukan instalasi perangkat lunak tambahan. Draw.io menyediakan antarmuka yang intuitif dengan fitur *drag-and-drop*, serta integrasi dengan layanan penyimpanan *cloud* seperti Google Drive dan OneDrive, memudahkan kolaborasi dan penyimpanan diagram secara *online*. Dengan kemampuannya yang luas, Draw.io menjadi alat yang populer di kalangan profesional TI, pendidik, dan pengembang perangkat lunak untuk merancang dan mendokumentasikan sistem secara visual.

Dalam jurnal "Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis *Internet of Things*" oleh Arumsari, Maulindar, dan Pradana (2023), Draw.io digunakan sebagai salah satu alat bantu dalam proses perancangan sistem. Penelitian ini bertujuan untuk membangun sistem yang dapat mendeteksi tanda-tanda kebakaran dini menggunakan berbagai sensor dan mengirimkan notifikasi melalui aplikasi Telegram. Dalam pengembangan sistem ini, Draw.io dimanfaatkan untuk membuat diagram alur dan perancangan sistem secara visual, yang membantu tim pengembang dalam memahami dan merancang struktur sistem secara keseluruhan. Penggunaan Draw.io, bersama dengan alat lain seperti *Fritzing* dan *Arduino* *IDE*, memungkinkan perancangan sistem yang lebih terstruktur dan efisien, dapat dilihat pada gambar 2.10 [17].



Gambar 2. 10 Logo Draw.io

1. ***Xampp***

*XAMPP* adalah web server *opensource* yang berjalan pada sistem operasi *cross-platform* (Windows, Linux, MacOS). Semua yang diperlukan untuk mengelola *website* tersedia di *XAMPP* seperti *Apache, MySQL/MariaDB, PHP,* dan *Perl*. Meski program di dalamnya lengkap, *XAMPP* tetap merupakan web server yang sederhana dan ringan, dapat dilihat pada gambar 2.11 [18].



Gambar 2. 11 Logo Xampp

1. **Metode *Prototype***

Metode Prototype adalah model pengembangan sistem yang menekankan pada pembuatan versi awal (*prototype*) dari sistem yang akan dibangun. Metode ini melibatkan interaksi langsung dengan pengguna untuk mendapatkan umpan balik secara cepat dan berkelanjutan terhadap desain dan fungsi sistem. Melalui pendekatan ini, pengguna dapat melihat dan mencoba sistem lebih awal, sehingga kebutuhan dan keinginan mereka bisa dipahami dengan lebih baik sebelum pengembangan sistem dilakukan secara penuh. *Prototype* yang dibuat dapat direvisi berkali-kali sampai mencapai kesesuaian dengan kebutuhan pengguna.

Menurut [19], metode *prototype* memungkinkan pengembang dan pengguna bekerja sama secara intensif dalam mengembangkan sistem melalui tahapan perancangan, pembuatan *prototype*, evaluasi, dan revisi hingga menghasilkan sistem akhir yang sesuai kebutuhan. Dengan pendekatan ini, kesalahan dalam memahami kebutuhan pengguna dapat diminimalkan, dan hasil akhir sistem dapat lebih efektif dan efisien.

1. ***Black Box Testing***

*Black Box Testin*g adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas sistem tanpa memeriksa struktur internal atau kode program. Dalam pendekatan ini, penguji memberikan *input* ke sistem dan mengamati *output* yang dihasilkan untuk menentukan apakah sistem berperilaku sesuai dengan spesifikasi yang diharapkan. Metode ini sangat berguna untuk mengidentifikasi kesalahan dalam fungsi sistem, antarmuka pengguna, dan interaksi antar komponen, tanpa memerlukan pengetahuan mendalam tentang implementasi internal perangkat lunak.

Menurut [20] menyatakan bahwa metode Black Box Testing diterapkan untuk menguji sistem informasi perpustakaan berbasis web. Penelitian ini menggunakan teknik *Equivalence* *Partitioning* untuk mengidentifikasi dan mengelompokkan data *input* yang relevan dalam pengujian. Hasil pengujian menunjukkan bahwa beberapa formulir dalam sistem menghasilkan *output* yang tidak sesuai dengan yang diharapkan, dengan tingkat validitas sistem sebesar 75%. Kesalahan tersebut kemungkinan disebabkan oleh kesalahan pada baris kode (*syntax*), sehingga diperlukan pembaruan kode untuk memperbaiki *bug* dan meningkatkan kualitas *system*.

1. **SUS (*System* *Usability* *Scale*)**

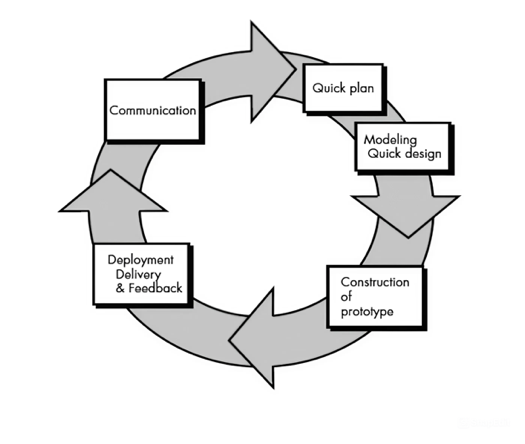
SUS merupakan Alat evaluasi *usability* yang diciptakan oleh John Brooke pada tahun 1986, *System Usability Scale* (SUS) terdiri dari sepuluh pernyataan dengan skala *Likert* untuk mengukur persepsi subjektif pengguna terhadap kegunaan suatu produk atau sistem yang sederhana namun andal. Menurut [21] menyatakan bahwa metode SUS digunakan untuk menilai tingkat kegunaan situs web Shopee. Dengan skor rata-rata 67,08, situs tersebut dikategorikan sebagai "OK" atau "*Marginally* *Acceptable*", menunjukkan bahwa situs tersebut cukup mudah digunakan, tetapi juga menunjukkan bahwa situs tersebut cukup dapat digunakan namun masih memiliki ruang untuk peningkatan.

# **BAB III METODE PENELITIAN**

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut:

## **3.1 Pengumpulan Data**

**Dalam penelitian ini, pengumpulan data dilakukan terlebih dahulu melalui observasi, wawancara, dan studi dokumen untuk memperoleh informasi kebutuhan sistem secara jelas.** Setelah itu, metode Prototype digunakan untuk mengembangkan sistem. Metode ini adalah model pengembangan sistem yang menekankan pada pembuatan versi awal atau prototype dari sistem yang akan dibangun. Metode ini melibatkan interaksi langsung dengan pengguna untuk mendapatkan umpan balik cepat dan berkelanjutan terhadap desain dan fungsi sistem. Dengan cara ini, pengguna dapat melihat dan mencoba sistem lebih awal, sehingga mereka dapat mengetahui kebutuhan dan keinginan sistem. Protokol dapat diubah berkali-kali untuk memenuhi kebutuhan pengguna [19].



Gambar 3. 1 Gambar Metode Prototype

Keterangan :

1. ***Communication***

Pada tahap *Communication*, dilakukan analisis kebutuhan sistem dengan mengumpulkan data. Pengumpulan data ini dilakukan dengan cara wawancara langsung dengan pihak terkait untuk mendapatkan kebutuhan sistem yang diinginkan pengguna. Dari wawancara tersebut, dapat dianalisis mengenai sistem yang sedang berjalan.

1. ***Quick Plan***

Tahap *Quick Plan* merupakan tahap lanjutan dari proses *Communication*. Tahapan ini adalah pembuatan desain sederhana yang menggambarkan desain secara singkat tentang sistem yang akan dibuat. Pada tahap ini dihasilkan desain yang telah disepakati dengan pengguna

1. ***Modelng Quick Design***

Pada tahap *Modeling Quick Design*, mulai dilakukan perancangan sistem sesuai dengan kebutuhan pengguna. *Modeling* ini juga dapat memperkirakan pengkodean yang akan digunakan. Proses *modeling* ini dilakukan dengan merancang struktur data, arsitektur *software*, representasi *interface*, dan *unified modeling language (UML).*

1. ***Construction of Prototype***

Setelah melakukan tahap *modeling*, selanjutnya adalah tahap *Construction of prototype*. Pada tahap ini, mulai dilakukan pengkodean, yaitu membangun aplikasi web sesuai dengan perancangan pada tahap *modeling*.

1. *Deployment, Delivery & Feedback*

Tahap *Deployment*, *Delivery* & *Feedback* ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisis, desain, dan pengkodean, maka sistem yang sudah jadi akan digunakan oleh pengguna. Kemudian, sistem yang telah dibangun dilakukan pemeliharaan secara berkala dan berkomunikasi dengan pengguna mengenai aplikasi yang dipakai untuk mendapatkan tanggapan mengenai aplikasi tersebut dari mereka. Pengujian sistem dilakukan pada saat pengujian dan setelah aplikasi digunakan, sehingga sistem yang dihasilkan benar-benar sesuai dengan kebutuhan pengguna.

* 1. **Analisis Sistem**

Analisis sistem adalah proses untuk mengidentifikasi masalah yang terjadi pada objek penelitian melalui analisis. Analisis yang dilakukan pada penelitian ini meliputi analisis pengguna, analisis data, dan analisis proses. Sistem yang akan dibangun bertujuan untuk mendukung proses perancangan system manajemen operasional di Wox’s *Barbershop*. Berdasarkan hasil analisis permasalahan, pada tahap ini akan ditentukan batasan-batasan serta fitur-fitur yang akan dikembangkan dalam sistem. Hasil dari analisis ini akan digunakan sebagai pedoman dalam merancang sistem yang dibutuhkan.

* 1. **Perancangan Sistem**

Perancangan sistem adalah tahap untuk merancang dan menggambarkan bagaimana sistem akan dibentuk baik dari sisi logika data, alur sistem, hingga antarmuka pengguna. Adapun tahapan perancangan sistem dalam penelitian ini meliputi:

1. **Perancangan Model Sistem**
2. ***Flowchart***

Digunakan untuk menggambarkan alur proses sistem secara menyeluruh.

1. ***Class Diagram***Digunakan untuk menggambarkan struktur objek dan hubungan antar kelas dalam sistem.
2. **Perancangan Basis Data**
3. ***Entity Relationship Diagram (ERD)***

Digunakan untuk memodelkan hubungan antar entitas dalam basis data.

1. **Basis Data Konseptual**

Merancang skema awal basis data secara konseptual berdasarkan hasil *ERD*.

1. **Struktur Tabel (Relasional)**

Mendefinisikan struktur tabel dan tipe data secara rinci.

1. **Perancangan Antarmuka Pengguna**

**Desain Antarmuka**Menyusun tampilan halaman-halaman utama sistem (*mockup* atau *wireframe*).

1. **Perangkat Lunak Pendukung**Untuk menunjang proses perancangan sistem, digunakan beberapa perangkat lunak sebagai berikut:
2. **Visual Studio Code**: Editor kode untuk menyusun struktur antarmuka dan logika sistem.
3. ***MySQL****:* Sistem manajemen basis data yang digunakan untuk menyimpan data sistem.
4. **Google Chrome**: Digunakan untuk menguji tampilan sistem berbasis web.
5. ***Laravel Framework****: Framework* *PHP* yang mendukung struktur MVC dan pengembangan sistem web.

## **Pengujian Sistem**

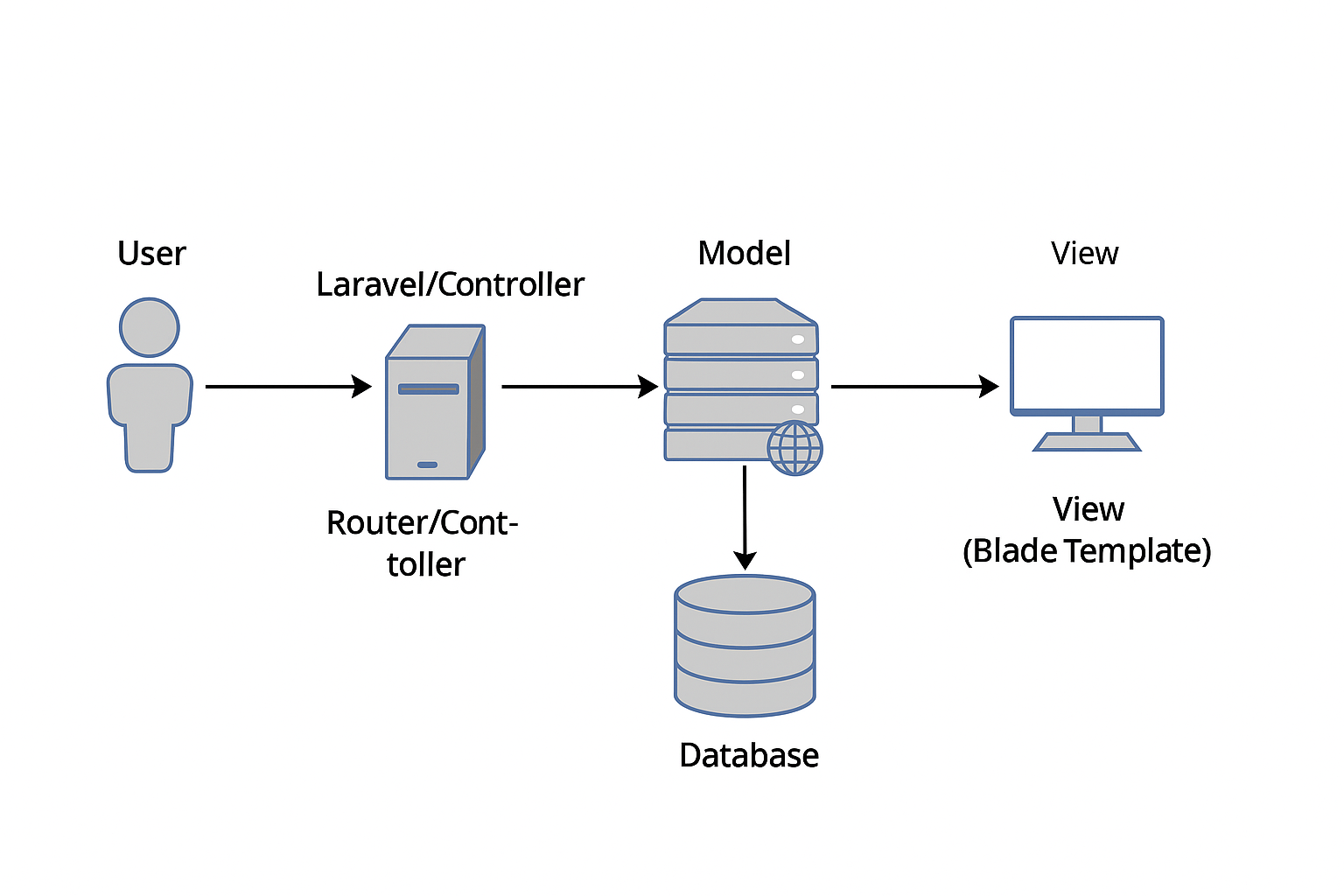
Pada tahap pengujian, sistem yang telah dibuat akan diuji menggunakan metode *black box testing* untuk menemukan kesalahan dan memastikan kesesuaian sistem dengan kebutuhan pengguna, Pengujian akan disesuaikan dengan ketentuan yang telah ditetapkan, mencakup berbagai aspek yang relevan untuk memastikan kualitas dan kinerja sistem. Setiap langkah dalam proses pengujian akan didokumentasikan secara rinci, dan hasilnya akan dirangkum dalam tabel rangkuman hasil pengujian sistem. Tabel ini akan mencakup data terkait keberhasilan atau kegagalan setiap uji, serta catatan mengenai setiap masalah yang ditemukan dan langkah-langkah yang diambil untuk mengatasinya. Dengan demikian, tabel rangkuman hasil pengujian akan memberikan gambaran menyeluruh tentang kondisi dan kinerja sistem setelah melalui proses pengujian.

## **Penulisan Laporan**

Pada tahap penulisan laporan, dilakukan penyusunan laporan penelitian yang mencakup semua tahap mulai dari metode pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, hingga tahap pengujian sistem. Laporan ini diakhiri dengan kesimpulan dan saran.

## **Arsitektur Sistem**

Arsitektur *Model-View-Controller (MVC)* digunakan oleh sistem manajemen operasional ini karena *Laravel*, sebuah *framework* berbasis *PHP*, secara *default* menerapkan pola tersebut. Arsitektur *MVC* membagi logika aplikasi menjadi tiga bagian utama: Model (logika bisnis dan basis data), *View* (tampilan antarmuka), dan *Controller* (penanganan permintaan pengguna). Dengan membedakan masalah, pemisahan ini mempermudah pengembangan dan pemeliharaan kode. Ini membuat kode lebih terorganisir dan dapat dikembangkan ulang secara modular. Arsitektur *MVC* dapat diimplementasikan dengan sukses dalam aplikasi web berbasis jadwal potong rambut ini berkat fitur otentikasi dan *routing Laravel*, dapat dilihat pada gambar 3.2



Gambar 3. 2 Gambar Arsitektur Sistem

Keterangan :

1. Pelanggan mengirim permintaan (HTTP *request*) lewat browser ke server (misal, memilih jadwal *booking*, memilih pilihan rekomendasi rambut).
2. *Laravel* melakukan *routing* untuk mengarahkan permintaan tersebut ke *Controller* yang sesuai.
3. *Controller* menerima permintaan, memvalidasi *input*, lalu berinteraksi dengan Model untuk mengambil atau menyimpan data di basis data.
4. Model mengeksekusi operasi *database* (misalnya menambah entri *booking* atau mengambil rekomendasi gaya rambut).
5. Setelah data siap, *Controller* meneruskan hasilnya ke *View*.
6. *View (Blade template Laravel)* menggabungkan data dengan tampilan HTML Tailwind CSS dan CSS untuk menghasilkan halaman web akhir.
7. Server mengirimkan *response* (halaman *HTML*) kembali ke browser pelanggan. Siklus ini memastikan data bergerak dari pengguna ke server dan kembali secara terstruktur.

*Halaman ini sengaja dikosongkan*

# **BAB IV HASIL DAN PEMBAHASAN**

## **Hasil Analisis**

Berdasarkan data yang diperoleh dan dianalisis, dapat diidentifikasi beberapa permasalahan yang menjadi dasar dalam pembuatan rancangan sistem. Hasil analisis ini memberikan gambaran secara umum mengenai komponen–komponen yang diperlukan untuk membangun sebuah sistem yang sesuai dengan kebutuhan operasional Wox’s Barbershop. Dalam penelitian ini terdapat beberapa analisis kebutuhan yang dijabarkan, yaitu analisis pengguna, analisis data, dan analisis proses, antara lain :

* 1. **Analisis Pengguna**

Dari hasil analisis ini, terdapat 3 (tiga) pengguna yang dapat mengakses sistem. Adapun peran pengguna tersebut pada Tabel 4.1 Analisis Pengguna.

Tabel 4. 1 Analisis Pengguna

| **No** | **Pengguna** | **Hak Akses** | **Keterangan** |
| --- | --- | --- | --- |
| 1 | Admin | Kelola data pengguna, transaksi, laporan keuangan, pengaturan sistem | Admin berperan sebagai pengelola utama sistem. |
| 2 | Pegawai | Kelola layanan pelanggan, data pelanggan, jadwal, konfirmasi pemesanan | Pegawai berinteraksi langsung dengan pelanggan dan operasional harian. |
| 3 | Pelanggan | Melakukan pemesanan online, melihat riwayat, mendapatkan rekomendasi, loyalty reward | Pelanggan menggunakan sistem untuk memperoleh layanan barbershop. |

* 1. **Analisis Data**

Analisis data dilakukan untuk mengidentifikasi data apa saja yang dibutuhkan dalam sistem agar dapat mendukung kebutuhan pengguna. Data tersebut dapat dilihat pada Tabel 4.2 Analisis Data.

Tabel 4. 2 Analisis Data

| **No** | **Data yang Dikelola** | **Deskripsi** | **Pengguna Terkait** |
| --- | --- | --- | --- |
| 1 | Data Pelanggan | Identitas, kontak, riwayat transaksi, status loyalty | Admin, Pegawai, Pelanggan |
| 2 | Data Pegawai | Identitas, jadwal kerja, hak akses | Admin, Pegawai |
| 3 | Data Layanan | Jenis layanan, harga, durasi | Admin |
| 4 | Data Pemesanan | ID pemesanan, pelanggan, layanan, jadwal, status | Admin, Pegawai, Pelanggan |
| 5 | Data Transaksi | Detail pembayaran, metode bayar, jumlah transaksi | Admin |
| 6 | Data Rekomendasi Gaya Rambut | Basis data gaya rambut sesuai jenis rambut dan bentuk wajah | Pelanggan, Pegawai |
| 7 | Data Laporan | Rekap data pelanggan, transaksi, tren gaya rambut | Admin |

* 1. Analisis Proses

Analisis proses dilakukan untuk menggambarkan alur bisnis yang berjalan pada sistem, mulai dari registrasi hingga pelaporan. Proses-proses tersebut ditunjukkan pada Tabel 4.3 Analisis Proses.

Tabel 4. 3 Analisis Proses

| **No** | **Proses** | **Deskripsi** | **Aktor Terkait** |
| --- | --- | --- | --- |
| 1 | Registrasi & Login | Pelanggan membuat akun dan login ke sistem | Pelanggan |
| 2 | Pemesanan Online | Pelanggan memilih layanan dan jadwal, sistem mencatat pemesanan | Pelanggan, Pegawai |
| 3 | Konfirmasi Pemesanan | Admin/Pegawai mengelola dan mengonfirmasi booking pelanggan | Admin, Pegawai |
| 4 | Rekomendasi Gaya Rambut | Sistem memberikan saran gaya rambut berdasarkan data pelanggan | Pelanggan, Pegawai |
| 5 | Layanan di Barbershop | Pegawai memberikan layanan sesuai pemesanan | Pegawai |
| 6 | Transaksi & Pembayaran | Sistem mencatat detail transaksi setelah layanan selesai | Pelanggan, Admin |
| 7 | Loyalty Reward | Sistem menghitung jumlah kunjungan pelanggan, memberikan reward gratis potong rambut setelah 10 kali kunjungan | Pelanggan |
| 8 | Pelaporan | Admin/Owner mengakses laporan keuangan, transaksi, tren gaya rambut | Admin, Owner |

Berdasarkan hasil analisis yang telah dilakukan terhadap pengguna, data, dan proses, dapat disimpulkan bahwa sistem manajemen operasional dan rekomendasi gaya rambut pada Wox’s Barbershop membutuhkan integrasi yang mampu mengakomodasi kebutuhan seluruh aktor yang terlibat. Analisis pengguna menunjukkan adanya tiga peran utama dengan hak akses yang berbeda, analisis data mengidentifikasi informasi penting yang harus dikelola secara terstruktur, sedangkan analisis proses menggambarkan alur bisnis utama yang harus didukung oleh sistem. Hasil analisis ini menjadi dasar dalam perancangan sistem yang akan dibahas pada subbab berikutnya, sehingga sistem yang dikembangkan dapat berjalan sesuai kebutuhan operasional barbershop.

## **Perancangan Sistem**

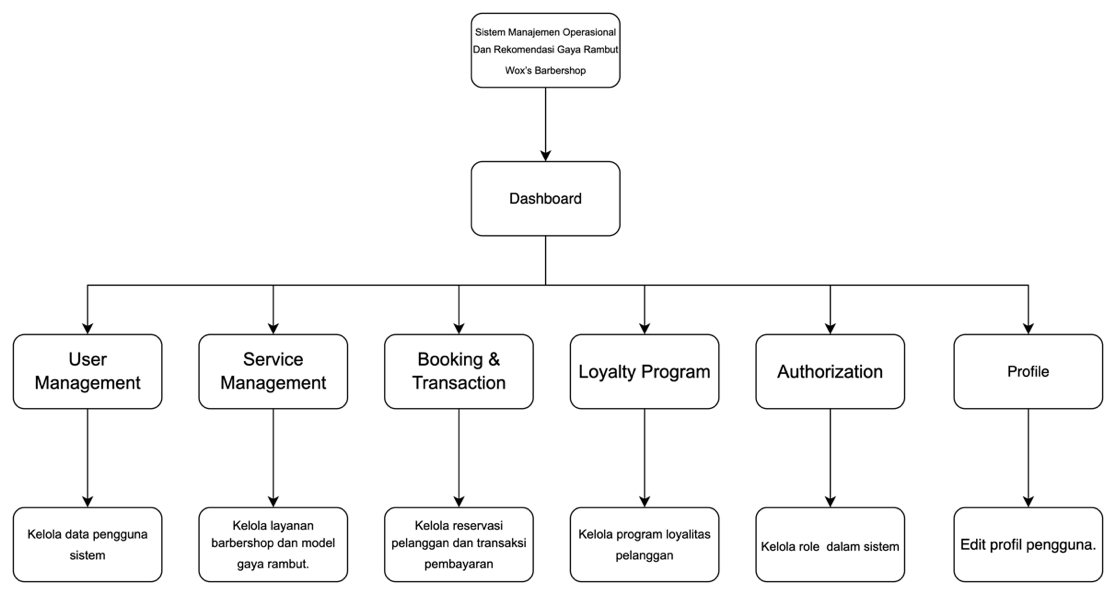
Perancangan sistem merupakan penggambaran, perencanaan dan pembuatan sketsa dari sistem yang dibangun untuk menjelaskan secara detail bagaimana sistem tersebut berjalan.

* 1. Struktur Menu

Struktur menu menggambarkan hierarki navigasi dalam sistem. Dengan adanya struktur menu, pengguna dapat lebih mudah memahami alur penggunaan aplikasi serta fitur-fitur yang tersedia.

* 1. Struktur Menu Admin

Admin memiliki akses penuh terhadap seluruh fitur sistem, mulai dari pengelolaan data, transaksi, hingga pengaturan hak akses. Struktur menu Admin dapat dilihat pada **Gambar 4.1 Struktur Menu Admin**.

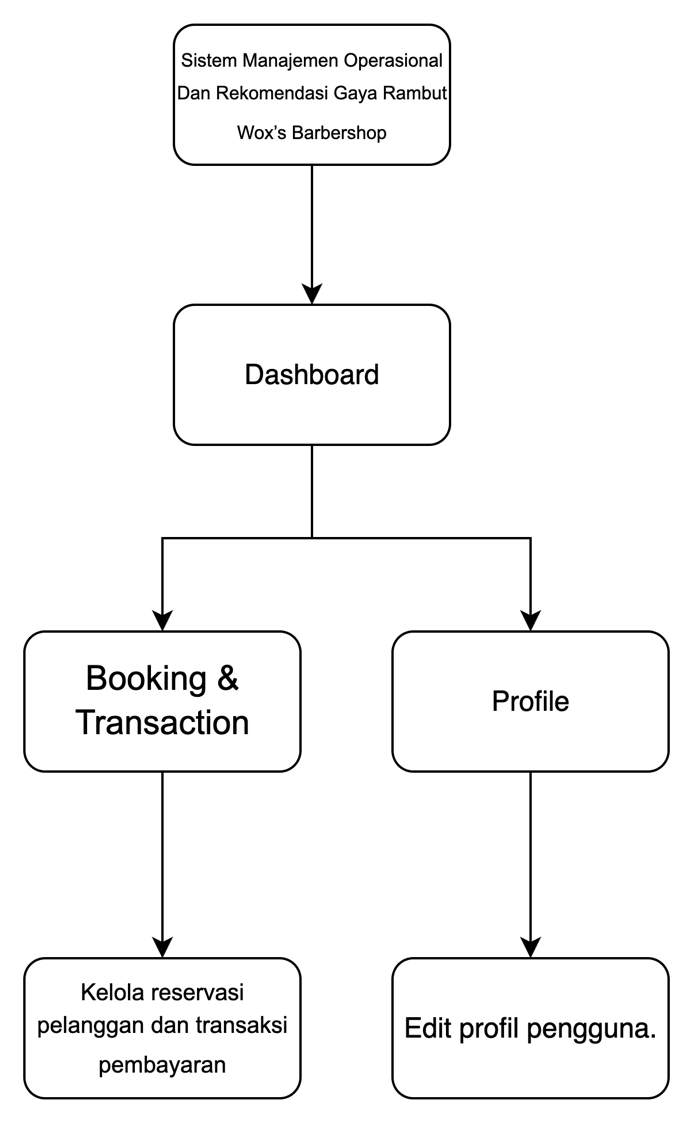


Gambar 4. 1 Struktur Menu Admin

Menu Admin terdiri atas:

* + 1. **Dashboard** berfungsi sebagai panel utama yang menampilkan statistik dan ringkasan sistem. Melalui fitur ini, admin dapat memantau performa barbershop, melihat data reservasi, transaksi, serta informasi penting lainnya secara real-time sehingga memudahkan pengambilan keputusan.
    2. **User Management** digunakan untuk mengelola data pengguna sistem. Fitur ini mencakup penambahan pengguna baru, pengeditan data, hingga penghapusan akun yang tidak aktif. Dengan demikian, data pengguna tetap terstruktur, akurat, dan mudah diakses.
    3. **Service Management** bertugas untuk mengatur layanan yang tersedia di barbershop. Melalui fitur ini, admin dapat menambahkan atau mengubah daftar layanan seperti potong rambut, perawatan wajah, serta mengelola model gaya rambut sesuai tren yang berkembang.
    4. **Booking & Transaction** berfungsi untuk mengelola reservasi pelanggan sekaligus pencatatan transaksi pembayaran. Pelanggan dapat melakukan pemesanan jadwal secara online, sementara sistem secara otomatis mencatat detail pembayaran agar transaksi lebih transparan dan efisien.
    5. **Loyalty Program** dirancang untuk meningkatkan kepuasan pelanggan dengan memberikan reward atau poin setiap kali melakukan transaksi. Poin tersebut dapat ditukar dengan diskon atau layanan tertentu, sehingga mampu mendorong loyalitas pelanggan terhadap barbershop.
    6. **Authorization Role** adalah fitur yang mengatur role setiap pengguna. Dengan adanya pengaturan ini, akses terhadap sistem menjadi lebih aman karena setiap pengguna hanya dapat mengakses menu dan fitur sesuai dengan hak akses yang telah ditentukan.
  1. **Struktur Menu Pegawai**

Pegawai memiliki akses yang berfokus pada pengelolaan transaksi harian dan konfirmasi reservasi pelanggan. Struktur menu Pegawai dapat dilihat pada **Gambar 4.2 Struktur Menu Pegawai**.

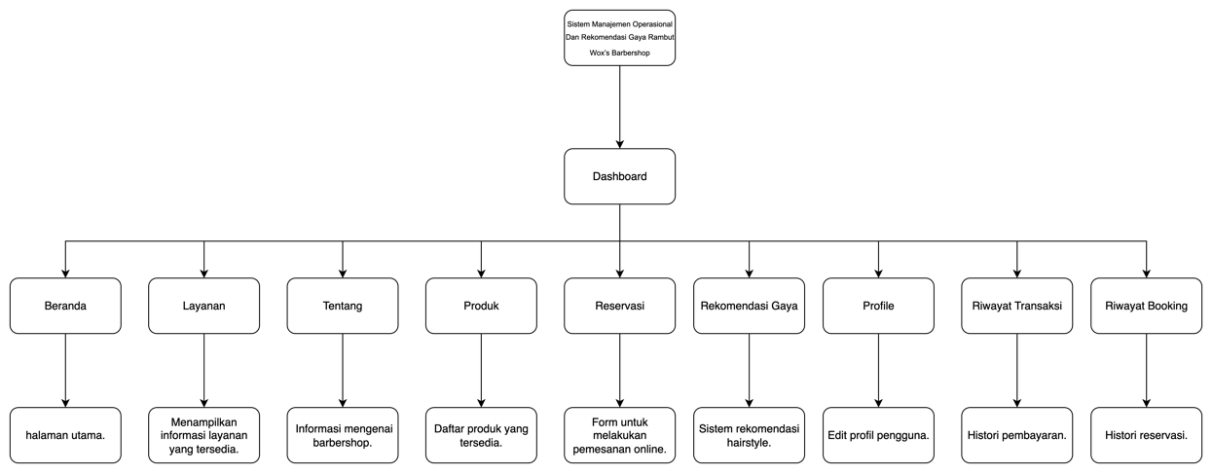


Gambar 4. 2 Struktur Menu Pegawai

Menu Pegawai terdiri atas:

1. **Dashboard** berfungsi sebagai panel utama yang menampilkan statistik dan ringkasan sistem. Melalui fitur ini, pegawai dapat memantau performa barbershop, melihat data reservasi, transaksi, secara real-time sehingga memudahkan pengambilan keputusan.
2. **Booking & Transaction** berfungsi untuk mengelola reservasi pelanggan sekaligus pencatatan transaksi pembayaran. Pelanggan dapat melakukan pemesanan jadwal secara online, sementara sistem secara otomatis mencatat detail pembayaran agar transaksi lebih transparan dan efisien.
3. **Struktur Menu Pelanggan**

Pelanggan memiliki akses untuk menggunakan layanan barbershop, melakukan reservasi, mengakses sistem rekomendasi gaya rambut, serta melakukan pembayaran online. Struktur menu Pelanggan dapat dilihat pada **Gambar 4.3 Struktur Menu Pelanggan**.



Gambar 4. 3 Struktur Menu Pelanggan

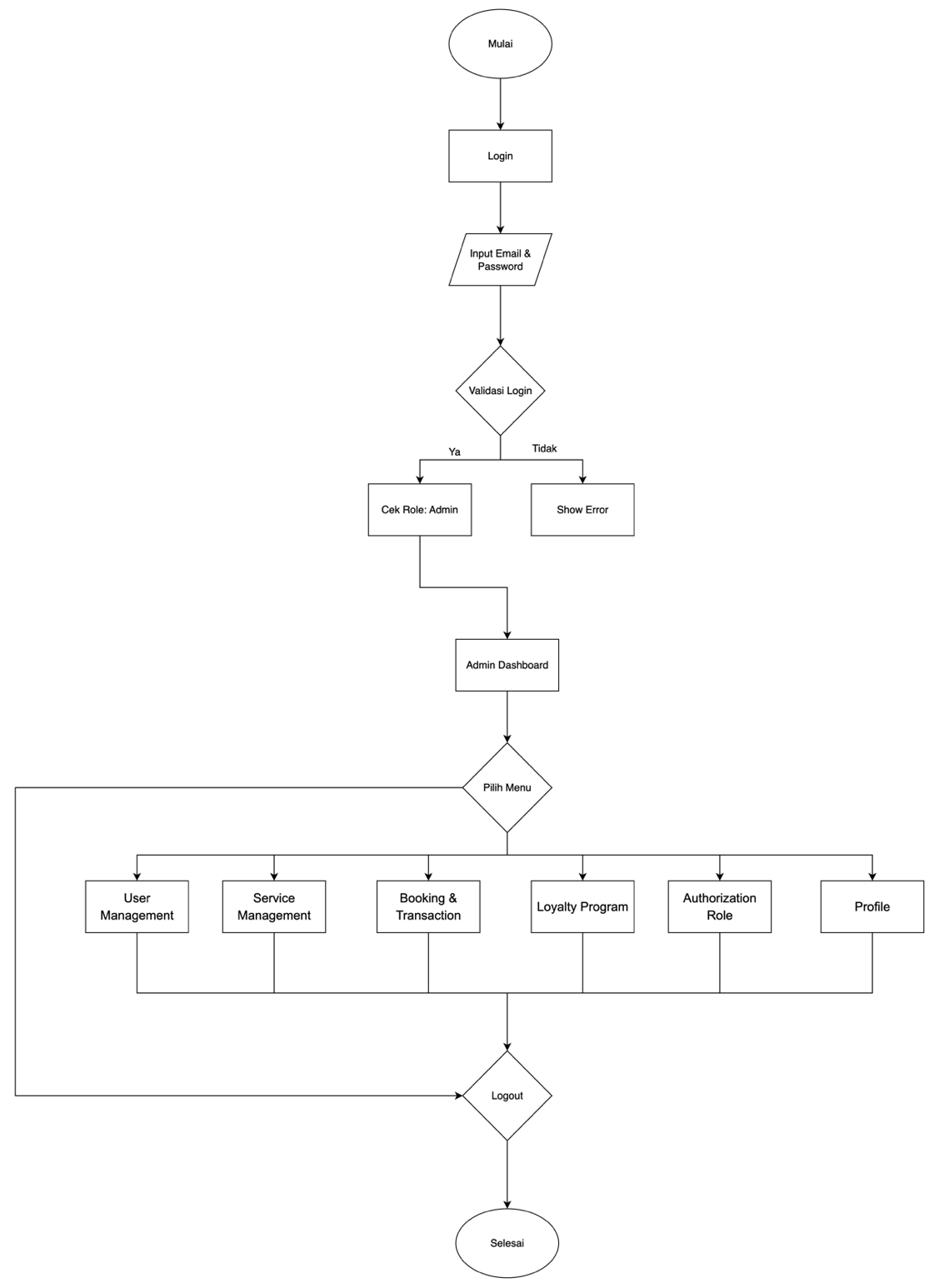
Menu Pelanggan terdiri atas:

* + 1. **Navigasi Utama – Beranda** berfungsi sebagai halaman utama yang menampilkan tampilan awal sistem kepada pelanggan. Pada halaman ini ditampilkan informasi umum barbershop dan akses cepat ke berbagai fitur lain.
    2. **Navigasi Utama – Layanan** menampilkan daftar layanan yang tersedia di barbershop, seperti potong rambut, perawatan wajah, atau layanan tambahan lain. Hal ini membantu pelanggan mengetahui pilihan layanan sebelum melakukan reservasi.
    3. **Navigasi Utama – Tentang** menyediakan informasi mengenai barbershop, termasuk profil, sejarah singkat, serta visi dan misi. Fitur ini bertujuan untuk memperkenalkan identitas barbershop kepada pelanggan.
    4. **Navigasi Utama – Produk** menampilkan daftar produk yang tersedia, seperti pomade, shampoo, atau perawatan rambut lainnya, sehingga pelanggan dapat melihat dan memilih produk yang sesuai dengan kebutuhan mereka.
    5. **Navigasi Utama – Reservasi** menyediakan form online yang memudahkan pelanggan melakukan pemesanan jadwal kunjungan. Dengan adanya fitur ini, pelanggan dapat memilih layanan, barber, dan waktu kunjungan sesuai preferensi.
    6. **Navigasi Utama – Rekomendasi Gaya** merupakan fitur sistem rekomendasi hairstyle yang membantu pelanggan menemukan gaya rambut yang sesuai berdasarkan preferensi, tren, atau karakteristik pribadi mereka.
    7. **Menu User – Profile** memungkinkan pelanggan untuk mengedit dan memperbarui data profil pribadi, seperti nama, nomor telepon, email, atau preferensi lainnya agar tetap sesuai dengan kebutuhan pengguna.
    8. **Menu User – Riwayat Transaksi** menampilkan histori pembayaran yang telah dilakukan pelanggan. Fitur ini mempermudah pelanggan dalam memantau catatan keuangan serta memastikan transaksi tercatat dengan baik.
    9. **Menu User – Riwayat Booking** berisi daftar histori reservasi yang pernah dilakukan. Melalui fitur ini, pelanggan dapat melihat jadwal reservasi sebelumnya sekaligus menjadi referensi untuk melakukan pemesanan ulang.
  1. Flowchart

Flowchart digunakan untuk menggambarkan alur kerja sistem secara menyeluruh. Dengan adanya flowchart, alur proses mulai dari login hingga laporan dapat dipahami dengan lebih jelas.

* + - * 1. Flowchart Admin

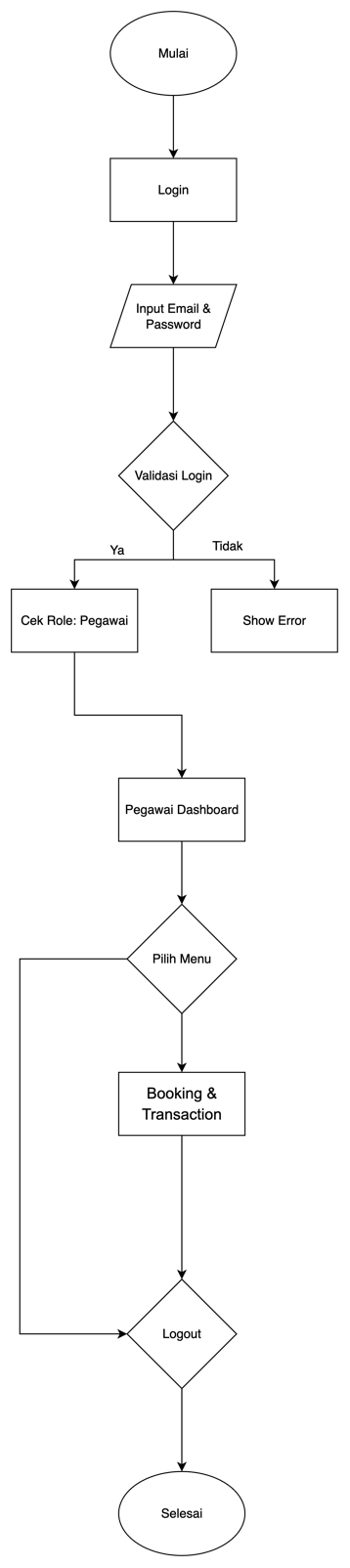
Flowchart Admin menggambarkan alur proses yang dapat dilakukan oleh Admin, mulai dari login hingga pengelolaan sistem. Flowchart Admin dapat dilihat pada **Gambar 4.4 Flowchart Admin**.



Gambar 4. 4 Flowchart Admin

* + - * 1. Flowchart Pegawai

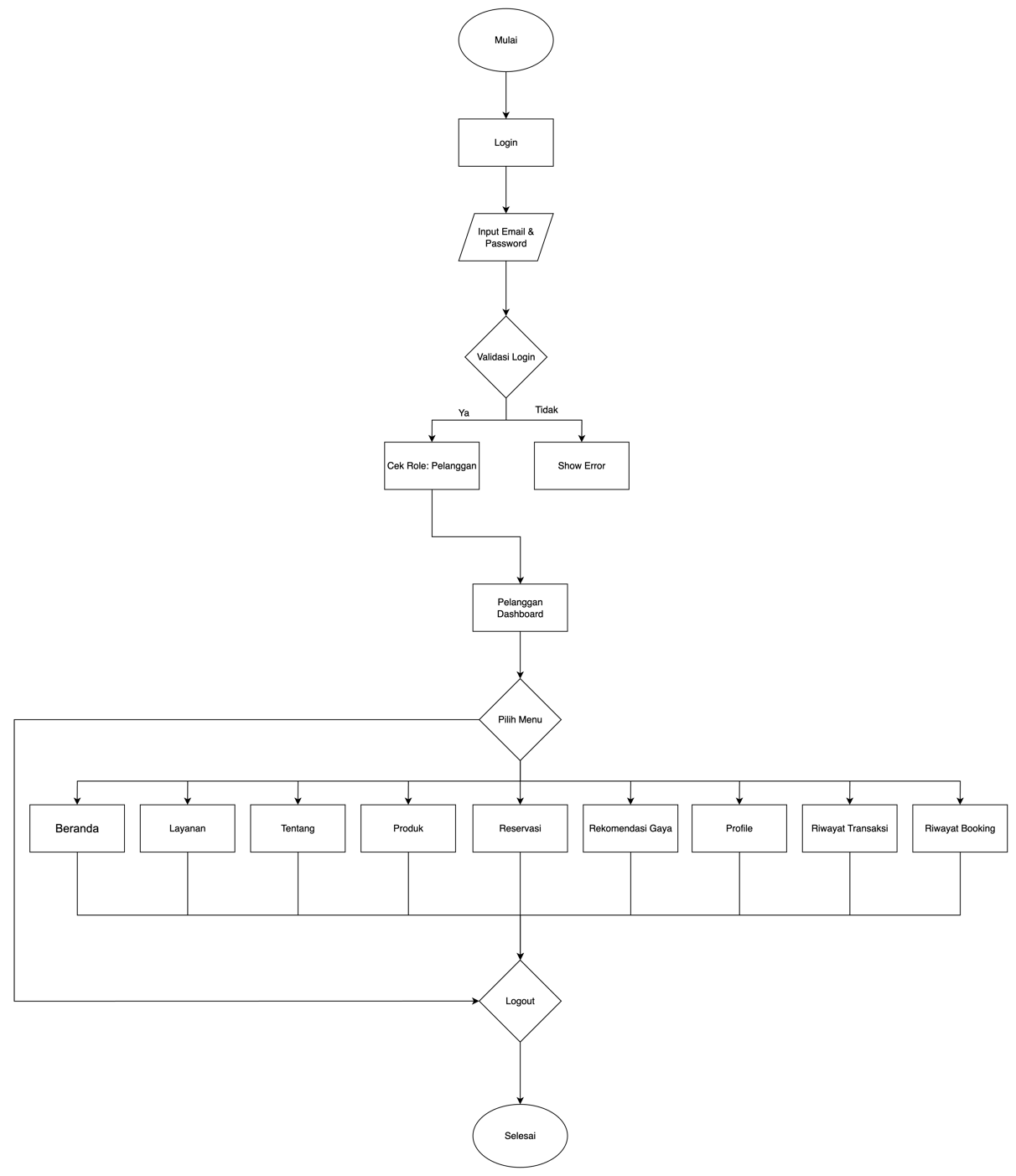
Flowchart Pegawai menggambarkan alur proses yang dapat dilakukan oleh Pegawai, mulai dari login hingga pengelolaan reservasi dan transaksi. Flowchart Pegawai dapat dilihat pada Gambar 4.5 Flowchart Pegawai.



Gambar 4. 5 Flowchart Pegawai

* + - * 1. Flowchart Pelanggan

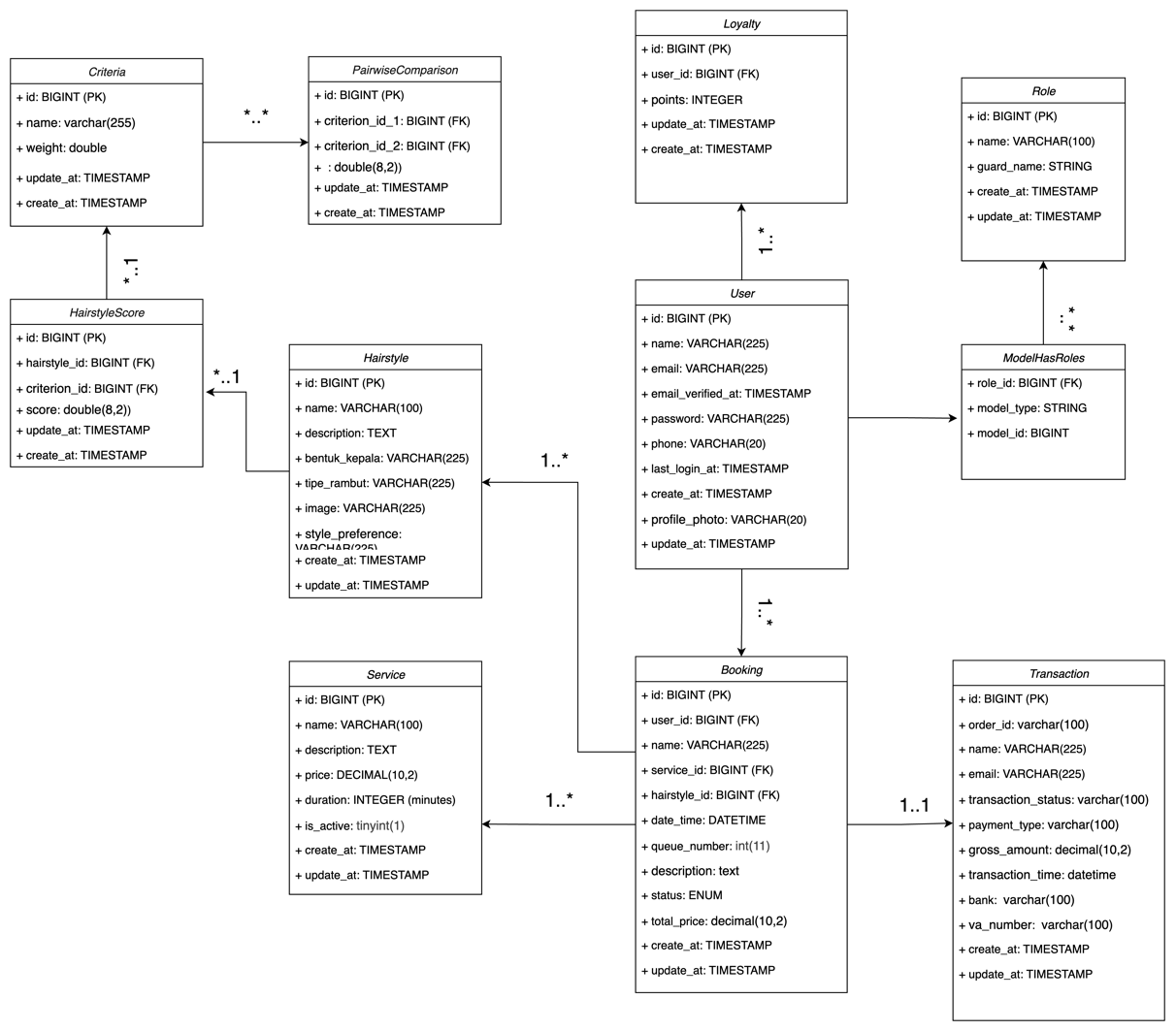
Flowchart Pelanggan menggambarkan alur proses yang dapat dilakukan oleh Pelanggan, mulai dari login hingga melakukan reservasi layanan dan pembayaran. Flowchart Pelanggan dapat dilihat pada **Gambar 4.6 Flowchart Pelanggan.**



Gambar 4.6 Flowchart Pelanggan

* 1. Class Diagram

Class Diagram digunakan untuk menggambarkan struktur objek beserta atribut dan metode, serta hubungan antar kelas dalam sistem. Diagram ini membantu pengembang memahami logika objek yang ada pada sistem. Class Diagram ditunjukkan pada **Gambar 4.6 Class Diagram Sistem**.



Gambar 4. 6 Class Diagram Sistem

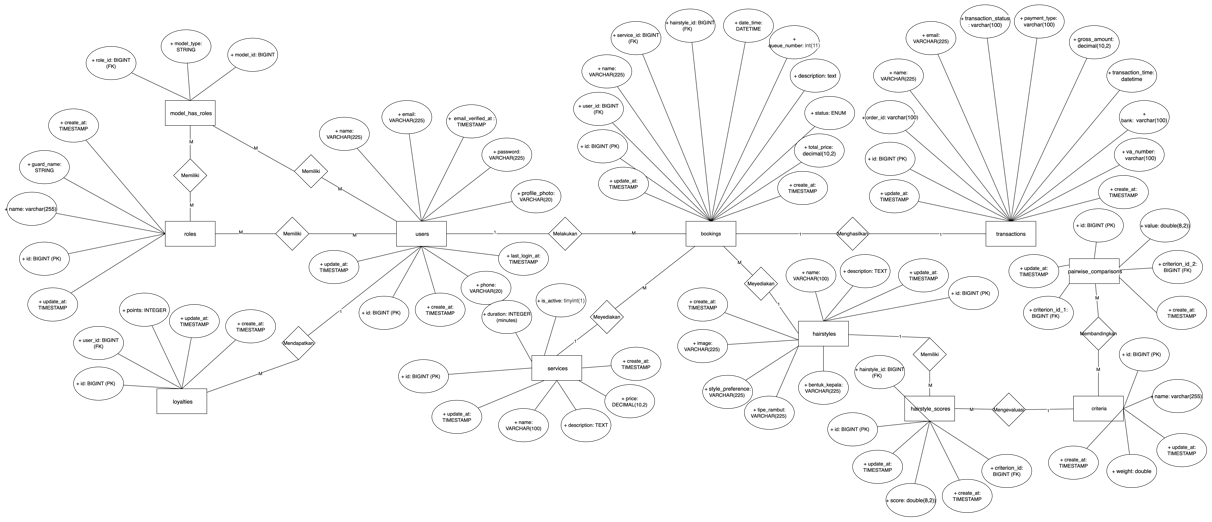
Class Diagram pada Gambar 4.6 menggambarkan hubungan antar kelas yang ada dalam sistem barbershop. Kelas-kelas utama yang terlibat antara lain **User, Booking, Transaction, Service, Hairstyle, Loyalty, Role, dan Criteria.** Setiap kelas memiliki atribut serta relasi yang saling terhubung untuk mendukung proses bisnis, seperti pemesanan layanan, transaksi pembayaran, sistem loyalitas, serta rekomendasi gaya rambut. Misalnya, kelas **User** berhubungan dengan **Booking** untuk mencatat reservasi, sementara kelas **Transaction** menangani detail pembayaran. Kelas **Criteria, PairwiseComparison, dan HairstyleScore** digunakan untuk mendukung sistem rekomendasi hairstyle berbasis preferensi pengguna. Dengan class diagram ini, pengembang dapat memahami alur interaksi antar objek dalam sistem secara menyeluruh sehingga memudahkan dalam proses implementasi dan pengembangan lebih lanjut.

## **Perancangan Basis Data**

Perancangan basis data merupakan tahap untuk menggambarkan struktur data yang digunakan dalam sistem. Basis data yang dirancang harus mampu menampung seluruh data yang dibutuhkan oleh sistem, serta memastikan hubungan antar entitas dapat berjalan dengan baik. Tahapan perancangan basis data dalam penelitian ini terdiri dari Entity Relationship Diagram (ERD), perancangan basis data konseptual, serta struktur tabel relasional.

* + - 1. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram (ERD)* digunakan untuk memodelkan hubungan antar entitas dalam basis data. ERD memberikan gambaran mengenai entitas utama yang terlibat dalam sistem, atribut yang dimiliki, serta hubungan antar entitas tersebut.



Gambar 4. 7 Entity Relationship Diagram Sistem

Keterangan:

1. Users - Bookings

Satu user dapat memiliki banyak booking. Contohnya, user dengan ID 1 dapat memiliki beberapa record di tabel bookings dengan user\_id yang sama. Sebaliknya, setiap booking hanya dimiliki oleh satu user, yang ditunjukkan oleh nilai user\_id yang unik di setiap record bookings.

1. Service – Bookings

Satu service dapat dipesan oleh banyak user. Contohnya, service Hair Cut (service\_id: 1) dapat dipesan oleh beberapa user. Dalam basis data, relasi one-to-many antara tabel services dan bookings memungkinkan satu service memiliki banyak record bookings dengan service\_id yang sama. Sebaliknya, setiap booking hanya merujuk ke satu service, yang ditunjukkan oleh nilai service\_id yang unik pada setiap record bookings.

1. User – Loyalty

Satu user dapat memiliki banyak riwayat loyalty point. Sistem loyalty menggunakan metode akumulasi dimana setiap kali user mengumpulkan 10 poin, mereka mendapatkan potongan gratis dan poin akan direset ke nol. Dalam basis data, relasi one-to-many antara tabel users dan loyalties memungkinkan satu user memiliki banyak record loyalties dengan user\_id yang sama. Setiap record loyalty hanya dimiliki oleh satu user tertentu, yang ditunjukkan oleh nilai user\_id yang unik pada setiap record.

1. Hairstyle – HairstyleScores

Setiap hairstyle dapat memiliki beberapa skor berdasarkan kriteria yang berbeda. Contohnya, hairstyle Undercut memiliki nilai 8.5 untuk Face Shape, 9.0 untuk Hair Type, dan 8.0 untuk Style. Dalam basis data, relasi one-to-many antara tabel hairstyles dan hairstyle\_scores memungkinkan satu hairstyle memiliki banyak record scores dengan hairstyle\_id yang sama. Setiap skor hanya terkait dengan satu hairstyle, yang ditunjukkan melalui foreign key hairstyle\_id yang unik.

1. Criteria – HairstyleScores

Satu kriteria dapat menilai banyak hairstyle. Contohnya, kriteria "Face Shape" menilai Undercut, Pompadour, dan Buzz Cut. Dalam basis data, relasi one-to-many antara tabel criterias dan hairstyle\_scores memungkinkan satu criteria memiliki banyak record scores dengan criteria\_id yang sama. Setiap skor hanya merujuk pada satu kriteria, yang ditunjukkan melalui foreign key criteria\_id yang unik.

1. Booking – Transaction

Satu booking menghasilkan satu transaksi. Contoh: booking Hair Cut tanggal 15 Agustus menghasilkan transaksi Rp 50.000. Dalam basis data, relasi one-to-one antara tabel bookings dan transactions memungkinkan satu booking memiliki satu record transaction dengan booking\_id yang sesuai. Setiap transaksi hanya merujuk pada satu booking tertentu, yang ditunjukkan melalui foreign key booking\_id yang unik.

1. User – Role

Satu user dapat memiliki beberapa role. Contohnya, user Admin dapat memiliki role Admin dan Staff sekaligus. Dalam basis data, relasi many-to-many antara users dan roles diimplementasikan melalui tabel perantara model\_has\_roles. Satu user dapat memiliki banyak record di tabel perantara dengan user\_id yang sama, dan satu role juga dapat dimiliki oleh banyak user, yang ditunjukkan melalui record yang terkait dengan role\_id yang sama.

1. Criteria – PairwiseComparison (Self-Referencing)

Satu user dapat memiliki beberapa role. Hubungan many-to-many antara users dan roles diimplementasikan melalui tabel perantara model\_has\_roles, yang memungkinkan satu user memiliki banyak role dan satu role dapat dimiliki oleh banyak user.

* + - 1. Basis Data Konseptual

Basis data konseptual dirancang berdasarkan Entity Relationship Diagram (ERD) untuk memastikan integritas dan konsistensi data. Skema konseptual berikut menggambarkan entitas utama, atribut, dan hubungan antar entitas tanpa memperhatikan implementasi fisik basis data.

Tabel 4. 4 Skema Konseptual Basis Data

| **No** | **Entitas** | **Atribut Utama** | **Tipe Hubungan** | **Entitas Terkait** |
| --- | --- | --- | --- | --- |
| 1 | Users | id, name, email, password, phone, email\_verified\_at, last\_login\_at,, profile\_photo, created\_at, updated\_at | One-to-Many | bookings, loyalties |
|  |  |  | Many-to-Many | roles |
| 2 | roles | id, name, guard\_name, created\_at, updated\_at | Many-to-Many | users |
| 3 | Services | id, name, description, price, duration, is\_active  created\_at, updated\_at | One-to-Many | bookings |
| 4 | bookings | id, user\_id, name, service\_id, hairstyle\_id,date\_time, queue\_number, description, status, total\_price, created\_at, updated\_at | One-to-One | transactions |
|  |  |  | Many-to-One | users, services, hairstyles |
| 5 | transactions | id, order\_id, name , email, transaction\_status, payment\_type, gross\_amount , transaction\_time, bank, va\_number, created\_at, updated\_at | One-to-One | bookings |
| 6 | loyalties | id, user\_id, type, points, created\_at, updated\_at | Many-to-One | users |
| 7 | hairstyles | id, name, description, bentuk\_kepala, tipe\_rambut, image, style\_preference, created\_at, updated\_at | One-to-Many | hairstyle\_scores |
| 8 | criterias | id, name, weight, created\_at, updated\_at | One-to-Many | hairstyle\_scores |
| 9 | hairstyle\_scores | id, hairstyle\_id, criterion\_id, score, created\_at, updated\_at | Many-to-One | hairstyles, criterias |
| 10 | model\_has\_roles | role\_id, model\_type, model\_id | Perantara Many-to-Many | users, roles |
| 11 | pairwise\_comparisons | id, criterion\_id\_1, criterion\_id\_2, value, created\_at, updated\_at | Self-Referencing | criterias |

Keterangan :

1. Users-bookings: Satu user dapat melakukan banyak booking.
2. Users-loyalties: Satu user dapat memiliki banyak riwayat loyalty.
3. Users-roles: Relasi many-to-many melalui tabel perantara model\_has\_roles.
4. Services-bookings: Satu service dapat dipesan dalam banyak booking.
5. Hairstyles -bookings: Satu hairstyles dapat dipesan dalam banyak booking.
6. Bookings-transactions: Satu booking menghasilkan satu transaksi (one-to-one).
7. Hairstyles-hairstyle\_scores: Satu hairstyle dapat memiliki banyak skor.
8. Criterias-hairstyle\_scores: Satu kriteria dapat digunakan untuk banyak skor.
9. Model\_has\_roles: Tabel perantara many-to-many antara users dan roles.
10. Criterias-pairwise\_comparisons: Self-referencing untuk perbandingan berpasangan.
    * + 1. Struktur Tabel (Relasional)

Struktur tabel dirancang secara rinci dengan menyesuaikan tipe data, constraint, dan kunci asing (foreign key) untuk memastikan integritas data. Berikut adalah definisi lengkap dari semua tabel yang digunakan dalam sistem:

Tabel 4. 5 Users

| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | id | Bigint | Primary Key, Auto\_Increment | ID Unik Pengguna |
| 2 | name | Varchar(255) | Not Null | Nama Lengkap Pengguna |
| 3 | email | Varchar(255) | Unique, Not Null | Alamat Email (Unik) |
| 4 | password | Varchar(255) | Not Null | Kata Sandi Terenkripsi |
| 5 | phone | Varchar(20) | Not Null | Nomor Telepon |
| 6 | email\_verified\_at | Timestamp | Null | Verifikasi Email |
| 7 | last\_login\_at | Timestamp | Null | Tanggal Terakhir Login |
| 8 | profile\_photo | Varchar(255) | Null | Photo Yang Diupload |
| 9 | created\_at | Timestamp | Default Current\_Timestamp | Waktu Dibuat |
| 10 | updated\_at | Timestamp | Default Current\_Timestamp On Update | Waktu Diperbarui |

Tabel 4.4 User digunakan untuk menyimpan data pengguna sistem. Tabel ini memiliki atribut utama berupa user\_id sebagai primary key, serta atribut lain yang ada di dalam tabel. Data dari tabel ini akan digunakan pada proses autentikasi dan pengelolaan profil pengguna. Struktur tabel berikutnya menjelaskan data roles yang disediakan dalam Tabel 4. 6 Roles.

Tabel 4. 6 Roles

| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | id | Bigint | Primary Key, Auto\_Increment | ID Unik Role |
| 2 | name | Varchar(255) | Unique, Not Null | Nama Role (Contoh: Admin) |
| 3 | guard\_name | Varchar(255) | - | Mekanisme Otentikasi |
| 4 | created\_at | Timestamp | Default Current\_Timestamp | Waktu Dibuat |
| 5 | updated\_at | Timestamp | Default Current\_Timestamp On Update | Waktu Diperbarui |

**Tabel 4.6 Roles** digunakan untuk menyimpan informasi mengenai peran (role) dari setiap pengguna sistem. Setiap role memiliki **id** yang bersifat unik dan dihasilkan secara otomatis melalui AUTO\_INCREMENT. Kolom **name** menyimpan nama role, misalnya Admin, Pegawai, atau Pelanggan, yang bersifat unik dan tidak boleh kosong. Dengan adanya tabel ini, sistem dapat mengatur hak akses pengguna sesuai dengan role yang dimilikinya. Struktur tabel berikutnya menjelaskan data model\_has\_role yang disediakan dalam Tabel 4. 7 Model Has Roles.

Tabel 4. 7 Model Has Roles

| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | role\_id | Bigint | Foreign Key (Roles.Id) | Merujuk Ke Id Role |
| 2 | model\_type | String | - | Jenis Model (Umumnya User) |
| 3 | model\_id | Bigint | Foreign Key (Users.Id) | Merujuk Ke Id User |

Tabel 4.7 Model Has Role digunakan sebagai tabel penghubung (junction table) antara tabel Users dan Roles. Dengan adanya tabel ini, setiap pengguna dapat memiliki lebih dari satu role, dan setiap role dapat diberikan kepada lebih dari satu pengguna (hubungan many-to-many).Kolom role\_id merupakan foreign key yang merujuk ke id pada tabel Roles, sedangkan kolom model\_id merupakan foreign key yang merujuk ke id pada tabel Users. Kolom model\_type digunakan untuk mendefinisikan jenis model yang terkait dengan role, walaupun dalam implementasi ini umumnya berisi nilai "User". Struktur tabel berikutnya menjelaskan data services yang disediakan Tabel 4. 8 Services.

Tabel 4. 8 Services

| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | id | Bigint | Primary Key, Auto\_Increment | Id Unik Layanan |
| 2 | name | Varchar(255) | Not Null | Nama Layanan |
| 3 | price | Decimal(10,2) | Not Null | Harga Layanan |
| 4 | duration | Int | Not Null | Durasi (Menit) |
| 5 | description | Text | Null | Deskripsi Layanan |
| 6 | is\_active | Tinyint | - | Penanda Aktif/Nonaktif |
| 7 | created\_at | Timestamp | Default Current\_Timestamp | Waktu Dibuat |
| 8 | updated\_at | Timestamp | Default Current\_Timestamp On Update | Waktu Diperbarui |

Tabel 4.8 Services digunakan sebagai tabel yang menyimpan data mengenai layanan yang tersedia pada barbershop, seperti di dalam table ini. Tabel ini akan berelasi dengan tabel Booking untuk mencatat pemesanan layanan oleh pengguna. Struktur tabel berikutnya menjelaskan detail pemesanan layanan melalui Tabel 4. 9 Bookings.

Tabel 4. 9 Bookings

| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | id | Bigint | Primary Key, Auto\_Increment | ID Unik Booking |
| 2 | user\_id | Bigint | Foreign Key (Users.Id) | Merujuk Ke ID Pengguna |
| 3 | name | Varchar(255) | Not Null | Nama User Booking |
| 4 | service\_id | Bigint | Foreign Key (Services.Id) | Merujuk Ke ID Layanan |
| 5 | hairstyle\_id | Bigint | Foreign Key (Hairstyle.Id) | Merujuk Ke ID Layanan |
| 6 | date\_time | Datetime | - | Tanggal Booking |
| 7 | queue\_number | Int(11) | Null | Nomor Antrean Booking |
| 8 | description | Text | Null | Deskripsi Booking |
| 9 | total\_price | Decimal | Null | Total Harga Booking |
| 10 | status | Enum | Default 'Pending' | Status: Pending/Confirmed/In Progress/Completed/Cancelled |
| 11 | created\_at | Timestamp | Default Current\_Timestamp | Waktu Dibuat |
| 12 | updated\_at | Timestamp | Default Current\_Timestamp On Update | Waktu Diperbarui |

Tabel 4.9 Booking digunakan untuk mencatat pemesanan layanan oleh pengguna. Tabel ini memiliki foreign key user\_id yang merujuk ke tabel User, service\_id yang merujuk ke tabel Service dan hairstyle\_id yang merujuk ke table Hairstyles. Dengan demikian, setiap record pada tabel Booking dapat menghubungkan pengguna tertentu dengan layanan yang dipilih. Struktur tabel berikutnya menjelaskan rincian pembayaran yang dilakukan oleh pengguna malalui Tabel 4. 10 Transactions.

Tabel 4. 10 Transactions

| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | id | Bigint | Primary Key, Auto\_Increment | Id Unik Transaksi |
| 2 | order\_id | Varchar(100) | - | Id Order |
| 3 | transaction\_status | Varchar(50) | - | Status Pembayaran |
| 4 | payment\_type | Varchar(50) | Null | Metode Pembayaran |
| 5 | gross\_amount | Decimal | Null | Total Bayar |
| 6 | transaction\_time | Datetime | Default Current\_Timestamp | Waktu Transaksi Dibuat |
| 7 | bank | Varchar(50) | Null | Bank yang digunakan |
| 8 | va\_number | Varchar(50) | Null | Virtual account number |
| 9 | created\_at | Timestamp | Default Current\_Timestamp | Waktu Dibuat |
| 10 | updated\_at | Timestamp | Default Current\_Timestamp On Update | Waktu Diperbarui |
| 11 | name | Varchar(225) | Null | Nama yang bertransaksi |
| 12 | email | Varchar(225) | Null | Email yang bertransaksi |

**Tabel 4.10 Transactions**Tabel ini digunakan untuk mencatat seluruh data transaksi pembayaran yang terjadi dalam sistem. Setiap transaksi memiliki **id** unik yang dihasilkan otomatis melalui AUTO\_INCREMENT.Struktur tabel berikutnya menjelaskan *loyalty* *reward* yang di peroleh pengguna malalui Tabel 4. 11 Loyalties.

Tabel 4. 11 Loyalties

| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT | ID unik loyalty |
| 2 | user\_id | BIGINT | FOREIGN KEY (users.id) | Merujuk ke ID pengguna |
| 3 | points | INT | NOT NULL | Jumlah poin |
| 4 | created\_at | Timestamp | Default Current\_Timestamp | Waktu Dibuat |
| 5 | updated\_at | Timestamp | Default Current\_Timestamp On Update | Waktu Diperbarui |

Tabel 4. 12 **Loyalties** digunakan untuk mencatat seluruh riwayat poin loyalitas yang diperoleh maupun digunakan oleh pengguna. Kolom **id** berfungsi sebagai kunci primer dengan nilai unik dan otomatis bertambah. Kolom **user\_id** menjadi foreign key yang menghubungkan tabel ini dengan tabel **Users**, sehingga setiap riwayat poin selalu terkait dengan pengguna tertentu.Kolom **type** berisi informasi jenis aktivitas poin, yaitu earned (poin diperoleh) atau used (poin digunakan). Selanjutnya, kolom **points** menyimpan jumlah poin yang berubah, dapat berupa nilai positif. Kolom **created\_at** mencatat waktu ketika data riwayat poin dibuat secara otomatis oleh sistem. Struktur tabel berikutnya menjelaskan hairstyles yang di ada dalam sistem malalui Tabel 4. 12 Hairstyles.

Tabel 4. 13 Hairstyles

| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | id | Bigint | Primary Key, Auto\_Increment | Id Unik Hairstyle |
| 2 | name | Varchar(255) | Not Null | Nama Gaya Rambut |
| 3 | description | Text | Null | Deskripsi Hairstyle |
| 4 | bentuk\_kepala | Varchar(255) | - | Bentuk Kepala User |
| 5 | tipe\_rambut | Varchar(255) | - | Tipe Rambut User |
| 6 | image | Varchar(255) | Null | Url Gambar |
| 7 | style\_preference | Varchar(255) | Null |  |
| 8 | created\_at | Timestamp | Default Current\_Timestamp | Waktu Dibuat |
| 9 | updated\_at | Timestamp | Default Current\_Timestamp On Update | Waktu Diperbarui |

**Tabel 4.12 Hairstyles** menyimpan data gaya rambut yang tersedia di barbershop. Setiap gaya rambut memiliki ID unik, nama, deskripsi, dan URL gambar untuk visualisasi. Tabel ini berfungsi sebagai referensi utama dalam memberikan pilihan gaya rambut kepada pelanggan. Struktur tabel berikutnya menjelaskan criterias yang digunakan dalam proses pengambilan keputusan melalui Tabel 4. 13 Criterias.

Tabel 4. 14 Criterias

| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | id | Bigint | Primary Key, Auto\_Increment | ID Unik Kriteria |
| 2 | name | Varchar(255) | Not Null | Nama Kriteria |
| 3 | weight | Double | NULL | Bobot Kriteria (AHP) |
| 4 | created\_at | Timestamp | Default Current\_Timestamp | Waktu Dibuat |
| 5 | updated\_at | Timestamp | Default Current\_Timestamp On Update | Waktu Diperbarui |

Tabel 4.13 Criterias ini menyimpan data kriteria yang digunakan dalam proses pengambilan keputusan, misalnya untuk rekomendasi gaya rambut. Setiap kriteria memiliki ID unik, nama kriteria, serta bobot (weight) yang digunakan dalam perhitungan AHP (Analytical Hierarchy Process).Selanjutnya, hubungan antara kriteria dengan gaya rambut dicatat pada **Tabel 4.14 Hairstyle Scores**.

Tabel 4. 15 Hairstyle Scores

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| 1 | id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT | ID unik skor |
| 2 | hairstyle\_id | BIGINT | FOREIGN KEY (hairstyles.id) | Merujuk ke ID hairstyle |
| 3 | criterion\_id | BIGINT | FOREIGN KEY (criteria.id) | Merujuk ke ID kriteria |
| 4 | score | FLOAT | NOT NULL | Nilai skor (misal: 8.5) |
| 5 | created\_at | Timestamp | Default Current\_Timestamp | Waktu Dibuat |
| 6 | updated\_at | Timestamp | Default Current\_Timestamp On Update | Waktu Diperbarui |

**Tabel 4.14 Hairstyle Scores** Tabel ini berfungsi mencatat skor setiap gaya rambut berdasarkan kriteria tertentu. Kolom **hairstyle\_id** menghubungkan tabel ini dengan **Hairstyles**, sedangkan **criteria\_id** menghubungkannya dengan **Criterias**. Kolom **score** berisi nilai penilaian (misalnya 8.5) sesuai hasil evaluasi. Data skor ini mendukung proses analisis dalam pemilihan gaya rambut terbaik. Struktur berikutnya dijelaskan pada **Tabel 4.15 Pairwise Comparisons**.

Tabel 4. 16 Pairwise Comparisons

| **No** | **Column** | **Type** | **Constraints** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT | ID unik perbandingan |
| 2 | criterion\_id \_1 | BIGINT | FOREIGN KEY (criterias.id) | Kriteria yang dibandingkan |
| 3 | criterion\_id \_2 | BIGINT | FOREIGN KEY (criterias.id) | Kriteria pembanding |
| 4 | value | FLOAT | NOT NULL | Nilai perbandingan (AHP) |
| 5 | created\_at | Timestamp | Default Current\_Timestamp | Waktu Dibuat |
| 6 | updated\_at | Timestamp | Default Current\_Timestamp On Update | Waktu Diperbarui |

**Tabel 4.15 Pairwise Comparisons** Tabel ini menyimpan hasil perbandingan berpasangan antar kriteria dalam metode AHP. Kolom **criteria\_id** dan **compared\_criteria\_id** masing-masing merepresentasikan kriteria yang dibandingkan, sedangkan kolom **value** menyimpan nilai perbandingan. Dengan adanya tabel ini, sistem dapat menghitung bobot kriteria secara lebih objektif.

## **Perancangan Struktur *Framework Laravel***

Perancangan struktur framework Laravel bertujuan untuk menjelaskan bagaimana sistem yang dikembangkan diorganisir dalam kerangka kerja Laravel. Dengan adanya struktur yang jelas, proses pengembangan, pemeliharaan, serta pengembangan lanjutan sistem dapat dilakukan secara lebih mudah dan terarah.

Laravel dipilih karena merupakan framework PHP yang modern, mendukung arsitektur **Model-View-Controller (MVC),** serta memiliki dokumentasi yang lengkap. MVC pada Laravel membantu memisahkan antara logika aplikasi (Model), tampilan (View), dan pengendali alur (Controller).Berikut merupakan perancangan struktur framework laravel pada system ini.

Tabel 4. 17 Struktur Framework Laravel

|  |  |  |
| --- | --- | --- |
| **No** | **Struktur** | **Keterangan** |
| 1 | Model | Pada folder ini terdapat beberapa file yang disimpan seperti User.php, Booking.php, Service.php, Hairstyle.php, Transaction.php, Loyalty.php, Criteria.php, Dashboard.php |
| 2 | View | Pada folder ini terdapat beberapa file yang disimpan app.blade.php, guest.blade.php, footer.blade.php, head.blade.php, navigation.blade.php, scripts.blade.php, sidebar.blade.php, login.blade.php, register.blade.php, forgot-password.blade.php, confirm-password.blade.php, reset-password.blade.php, verify-email.blade.php, dashboard.blade.php, welcome.blade.php, rekomendasi.blade.php, index.blade.php, create.blade.php, edit.blade.php, show.blade.php, transaction.blade.php. |
| 3 | **Controller** | Pada folder ini terdapat beberapa file yang disimpan seperti Controller.php, BookingController.php, PaymentController.php, MidtransCallbackController.php, ProfileController.php, DashboardController.php, RecommendationController.php, HairstyleController.php, LoyaltyController.php, PermissionController.php, RoleController.php, ServiceController.php, TransactionController.php, UserController.php, AuthenticatedSessionController.php, ConfirmablePasswordController.php, EmailVerificationNotificationController.php, EmailVerificationPromptController.php, NewPasswordController.php, PasswordController.php, PasswordResetLinkController.php, RegisteredUserController.php, VerifyEmailController.php. |

## **Perancangan Antarmuka**

Perancangan antarmuka pengguna akan menyusun tampilan halaman-halaman utama sistem (mockup atau wireframe). Antarmuka ini akan dikembangkan menggunakan Tailwind CSS dan CSS untuk desain yang modern dan fleksibel. Perancangan antarmuka ini merupakan bagian dari tahap Modeling Quick Design pada metode Prototype. Halaman-halaman antarmuka akan dirancang untuk tiga jenis pengguna utama: Pelanggan, Admin, dan Pegawai.

### **Halaman Untuk Pelanggan**

**1. Rancangan Halaman Landing Page**

Halaman ini menampilkan tampilan awal yang pertama kali muncul saat pelanggan membuka aplikasi. Pada bagian ini disajikan informasi singkat mengenai Wox’s Barbershop. Rancangan halaman landing page dapat di lihat pada Gambar 4.8.



Gambar 4. 8 Rancangan Halaman Landing Page Pelanggan

* + - 1. **Rancangan Halaman Registrasi**

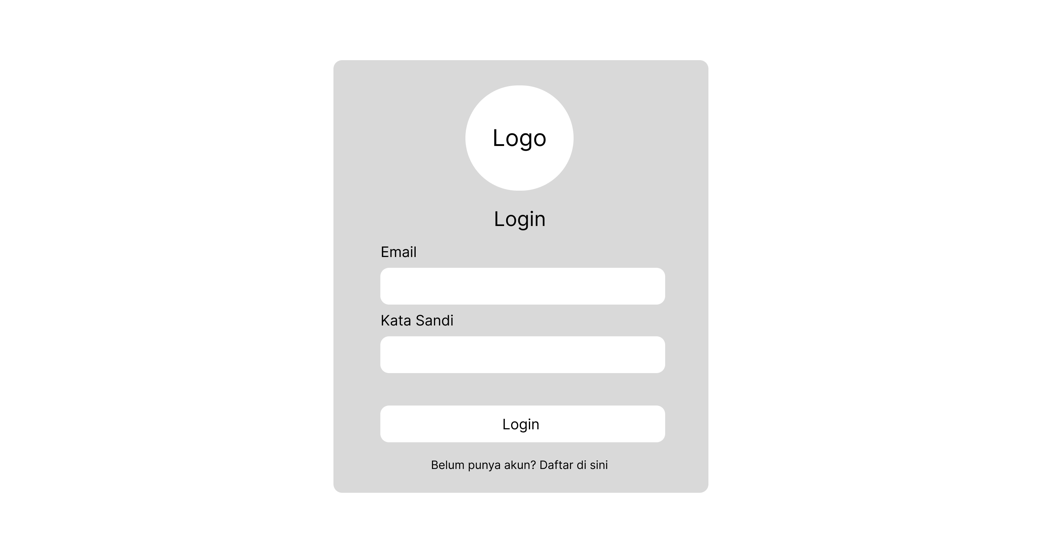
Halaman ini menampilkan **registrasi** digunakan oleh pelanggan untuk membuat akun baru yang akan digunakan untuk masuk ke dalam sistem. Rancangan halaman **registrasi** dapat di lihat pada Gambar 4.9.



Gambar 4. 9 Rancangan Halaman Registrasi Pelanggan

* + - 1. **Rancangan Halaman Login**

Halaman ini menampilkan login memungkinkan pelanggan masuk ke dalam sistem. Rancangan halaman **login** dapat di lihat pada Gambar 4.10.



Gambar 4. 10 Rancangan Halaman Login Pelanggan

* + - 1. **Rancangan Halaman Layanan**

Halaman ini menampilkan daftar layanan barbershop. Rancangan halaman **layanan** dapat di lihat pada Gambar 4.10.



Gambar 4. 11 Rancangan Halaman Layanan

* + - 1. **Rancangan Halaman Tentang**

Halaman ini menampilkan informasi singkat mengenai barbershop. Rancangan halaman **tentang** dapat di lihat pada Gambar 4.12.



Gambar 4. 12 Rancangan Halaman Tentang

* + - 1. **Rancangan Halaman Produk & Testimoni**

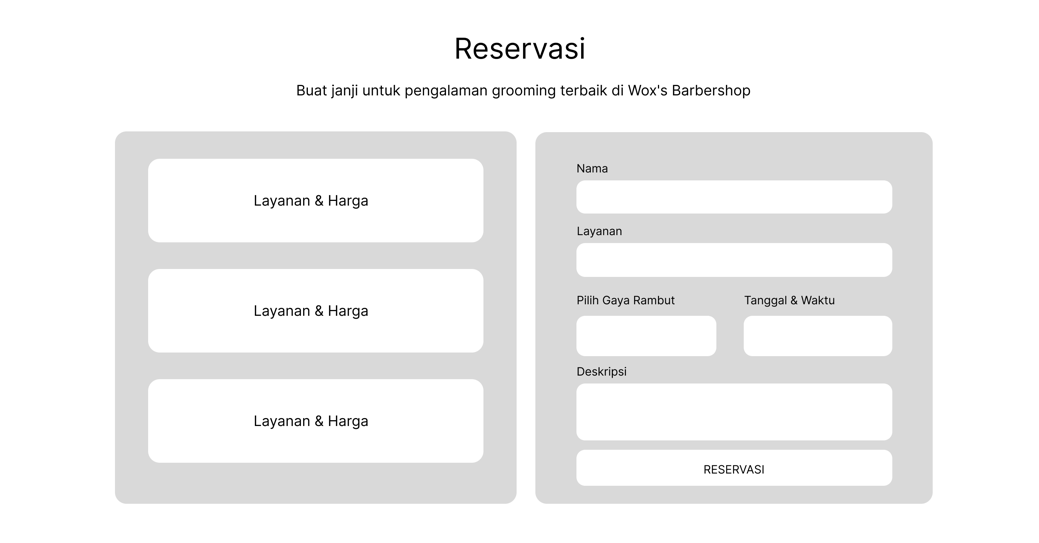
Halaman ini menampilkan daftar produk barbershop yang tersedia dengan deskripsi. Selain itu, disediakan juga bagian testimoni yang berisi ulasan dari pelanggan mengenai pengalaman mereka menggunakan layanan maupun produk. Rancangan halaman **produk** dapat di lihat pada Gambar 4.13.



Gambar 4. 13 Rancangan Halaman Produk & Testimoni

* + - 1. **Rancangan Halaman Reservasi**

Halaman ini menampilkan reservasi yang dirancang untuk memudahkan pelanggan dalam melakukan pemesanan layanan. Rancangan halaman **reservasi** dapat di lihat pada Gambar 4.14.



Gambar 4. 14 Rancangan Halaman Reservasi

* + - 1. **Rancangan Halaman Galeri & Footer**

Halaman ini menampilkan galeri dokumentasi foto dan aktivitas di barbershop. Sementara itu, bagian footer berisi informasi barbershop. Footer juga dilengkapi dengan navigasi cepat ke halaman utama. Rancangan halaman **galeri dan footer** dapat di lihat pada Gambar 4.15.



Gambar 4. 15 Rancangan Halaman Galeri & Footer

* + - 1. **Rancangan Halaman Rekomendasi Gaya Rambut**

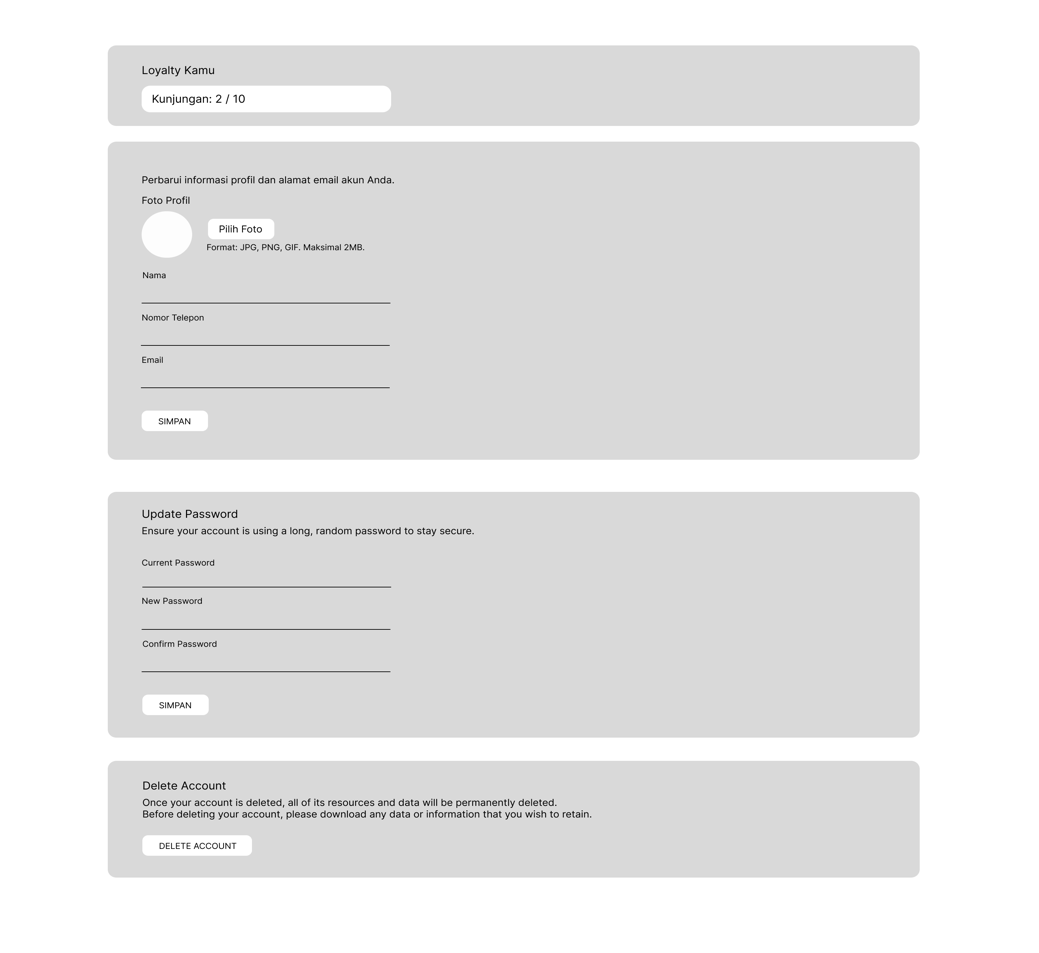
Halaman ini menampilkan rekomendasi gaya rambut menampilkan pilihan model sesuai bentuk wajah, jenis rambut, dan tren terbaru. Pengguna dapat melihat katalog bergambar lengkap dengan nama dan deskripsi. Rancangan halaman **rekomendasi gaya rambut** dapat di lihat pada Gambar 4.16.



Gambar 4. 16 Rancangan Halaman Rekomendasi Gaya Rambut

* + - 1. **Rancangan Halaman Profile & Loyalty**

Halaman ini menampilkan profile & loyalty merupakan fitur yang menampilkan informasi pribadi pelanggan sekaligus memberikan gambaran mengenai status program loyalitas yang dimiliki. Rancangan halaman **profile dan loyalty** dapat di lihat pada Gambar 4.17.



Gambar 4. 17 Rancangan Halaman Profile & Loyalty

* + - 1. **Rancangan Halaman Transaksi**

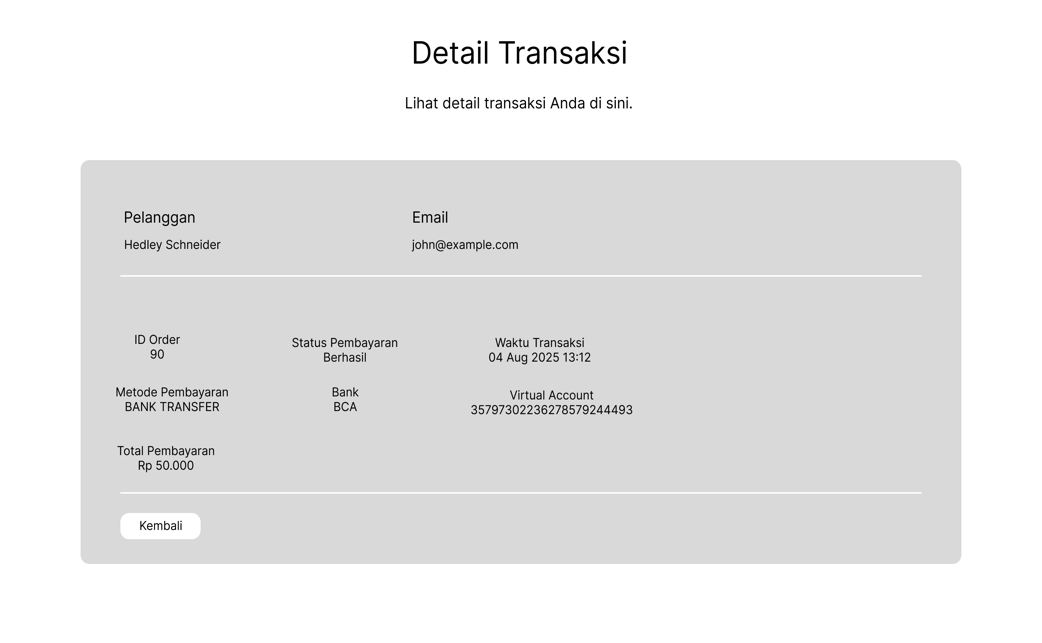
Halaman ini menampilkan transaksi yang dirancang untuk menampilkan keseluruhan riwayat transaksi pelanggan. Rancangan halaman **transaksi** dapat di lihat pada Gambar 4.18.



Gambar 4. 18 Rancangan Halaman Transaksi

* + - 1. **Rancangan Halaman Detail Transaksi**

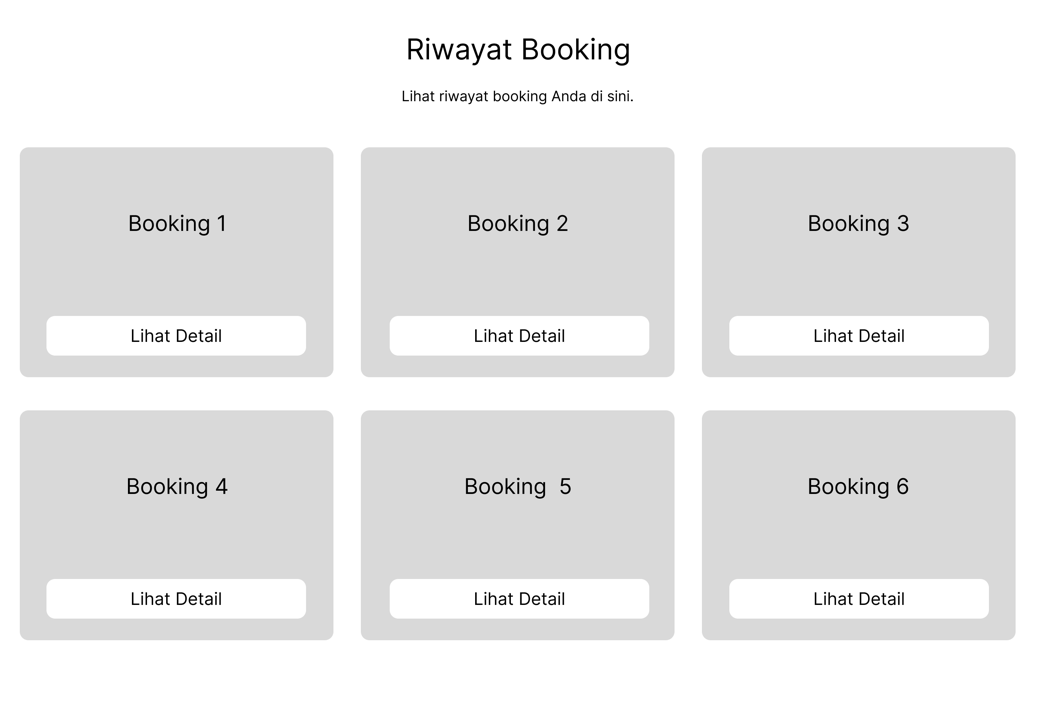
Halaman ini menampilkan detail transaksi digunakan untuk menampilkan informasi terkait satu transaksi tertentu yang dipilih oleh pelanggan. Rancangan halaman **detail transaksi** dapat di lihat pada Gambar 4.19.



Gambar 4. 19 Rancangan Halaman Detail Transaksi

* + - 1. **Rancangan Halaman Booking**

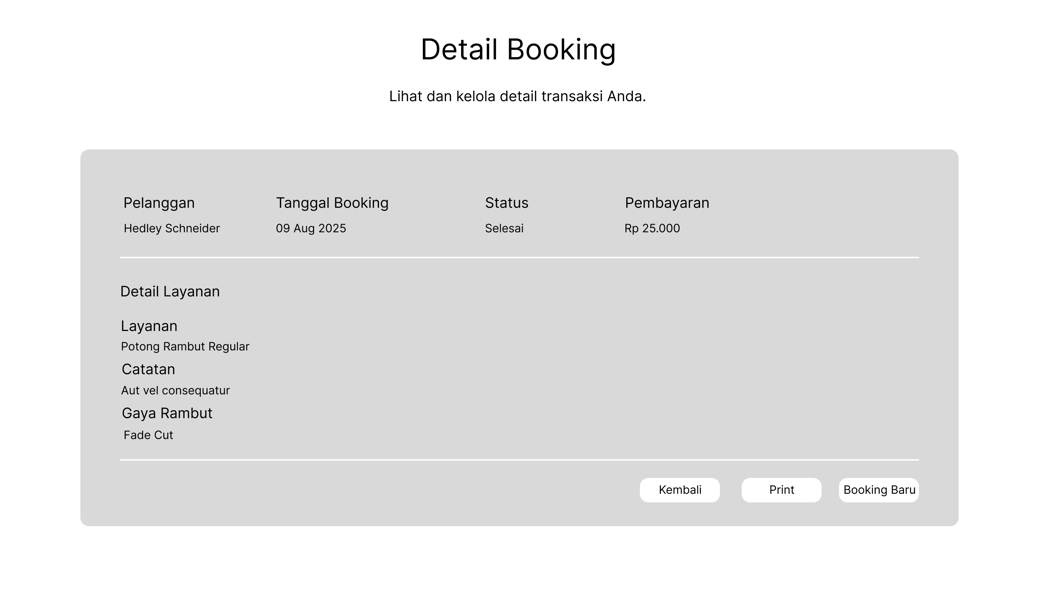
Halaman ini menampilkan booking digunakan untuk menampilkan keseluruhan riwayat booking pelanggan. Rancangan halaman **booking** dapat di lihat pada Gambar 4.20.



Gambar 4. 20 Rancangan Halaman Booking

* + - 1. **Rancangan Halaman Detail Booking**

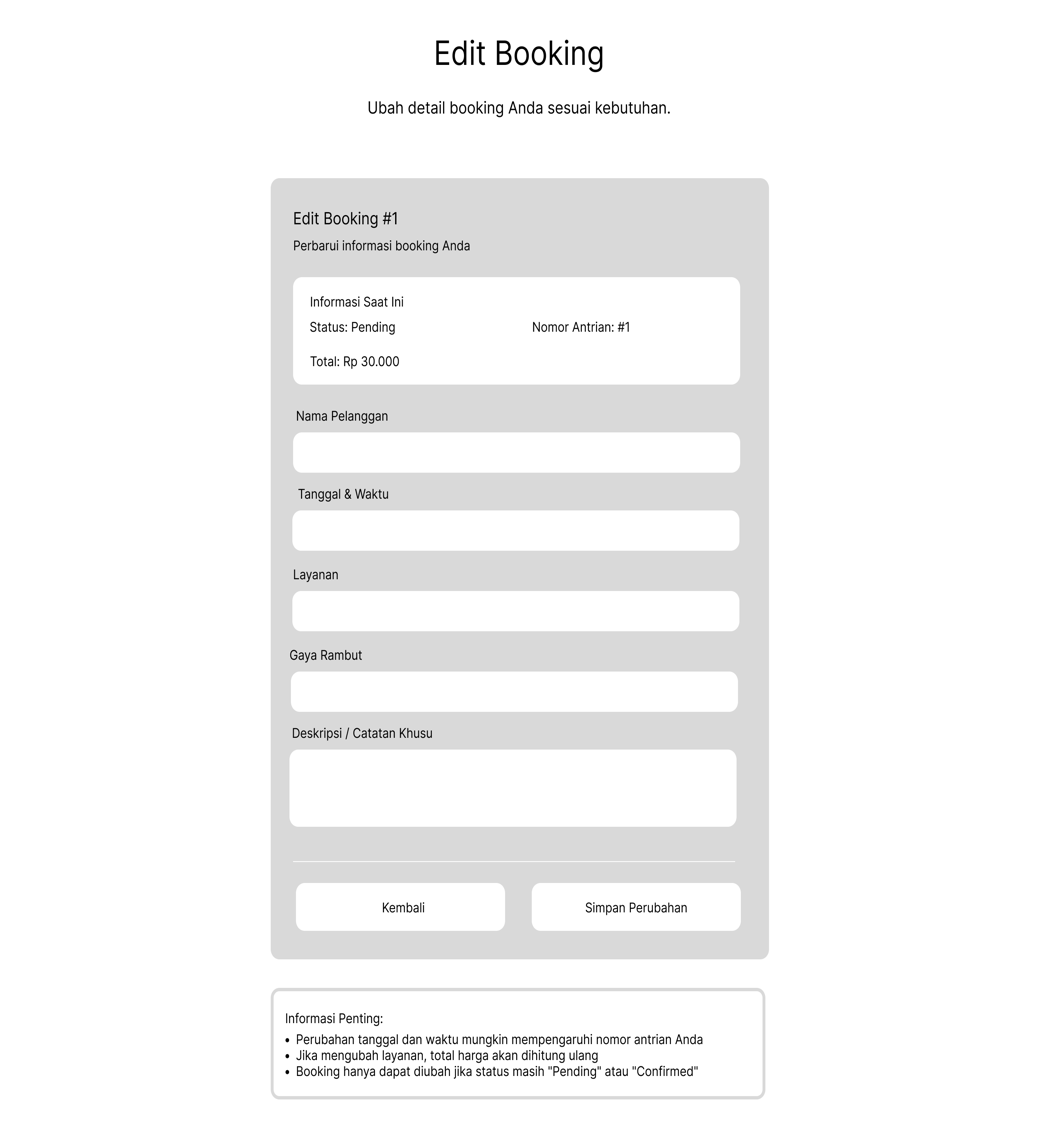
Halaman ini menampilkan detail booking digunakan untuk menampilkan informasi lengkap dari salah satu booking yang dipilih pelanggan. Rancangan halaman **detail booking** dapat di lihat pada Gambar 4.21.



Gambar 4. 21 Rancangan Halaman Detail Booking

* + - 1. Rancangan Halaman Edit Booking

Halaman ini menampilkan edit booking digunakan untuk mengubah data pada booking yang sudah dibuat pelanggan. Rancangan halaman **edit booking** dapat di lihat pada Gambar 4.22.

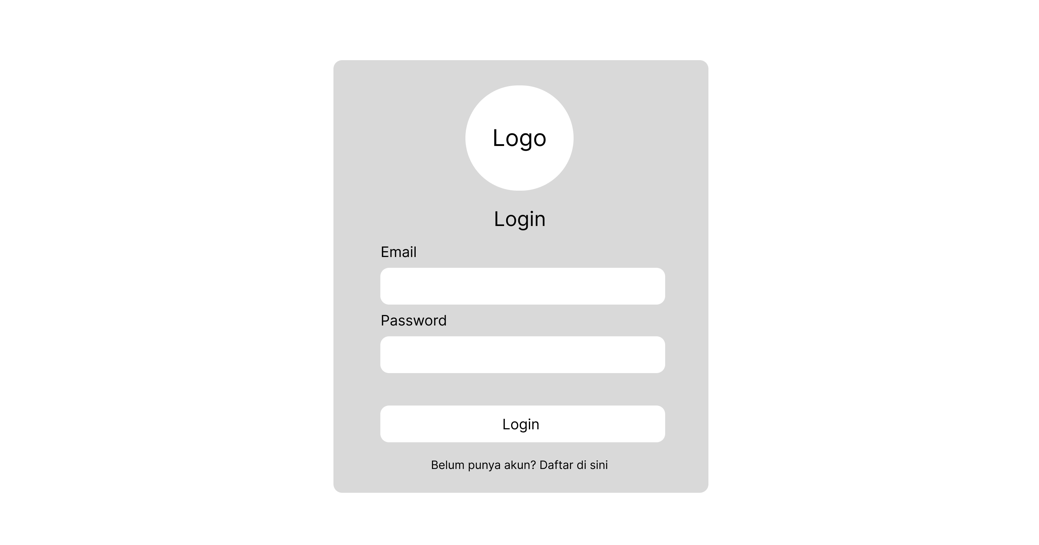


Gambar 4. 22 Rancangan Halaman Edit Booking

### **Halaman Untuk *Admin***

1. Rancangan Halaman Login Admin

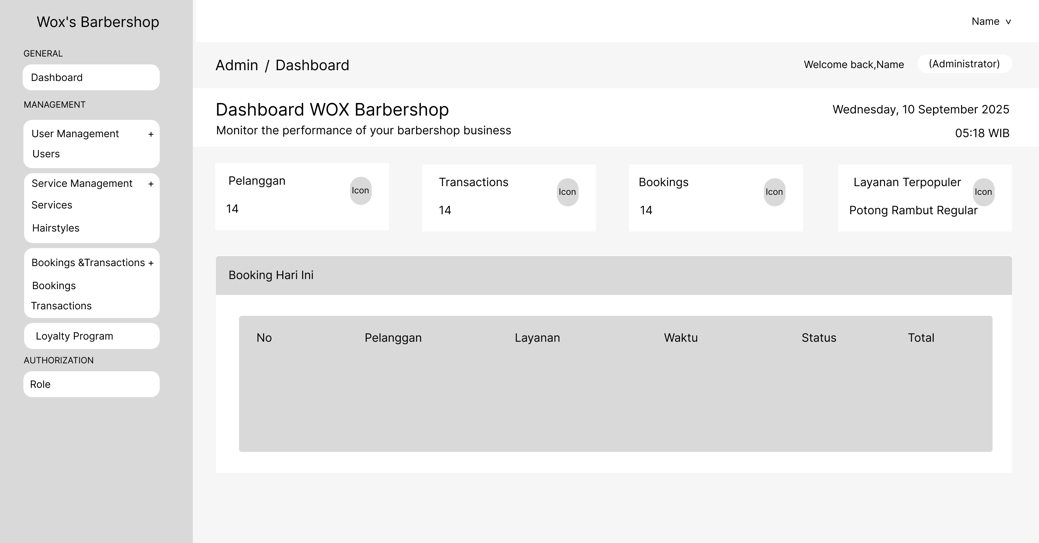
Halaman ini menampilkan autentikasi awal sebelum admin dapat mengakses sistem. Admin perlu memasukkan kredensial yang valid agar dapat masuk dan melanjutkan ke halaman dashboard. Rancangan halaman **login admin** dapat di lihat pada Gambar 4.23.



Gambar 4. 23 Rancangan Halaman Login Admin

1. Rancangan Halaman Dashboard Admin

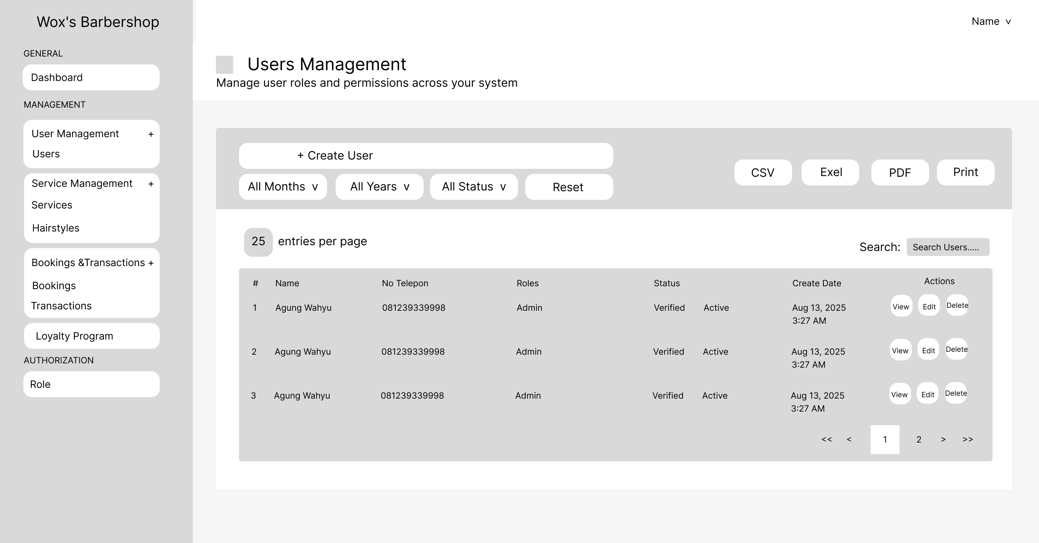
Halaman ini menampilkan ringkasan informasi utama yang membantu admin memantau kondisi sistem secara cepat dan efisien. Rancangan halaman **dashboard admin** dapat di lihat pada Gambar 4.24.



Gambar 4. 24 Rancangan Halaman Dashboard Admin

1. Rancangan Halaman User Management

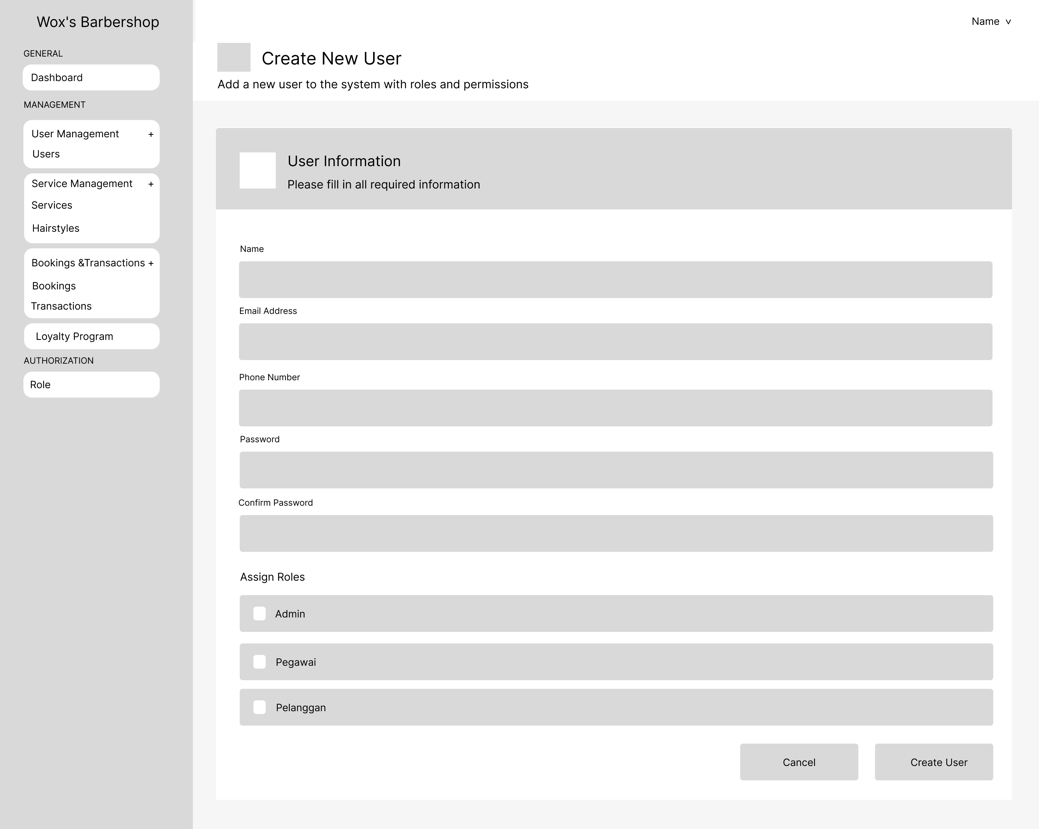
Halaman ini menampilkan data pengguna yang dapat di kelola oleh admin. Rancangan halaman user management dapat dilihat pada Gambar 4.25.



Gambar 4. 25 Rancangan Halaman User Management

1. Rancangan Halaman Create User

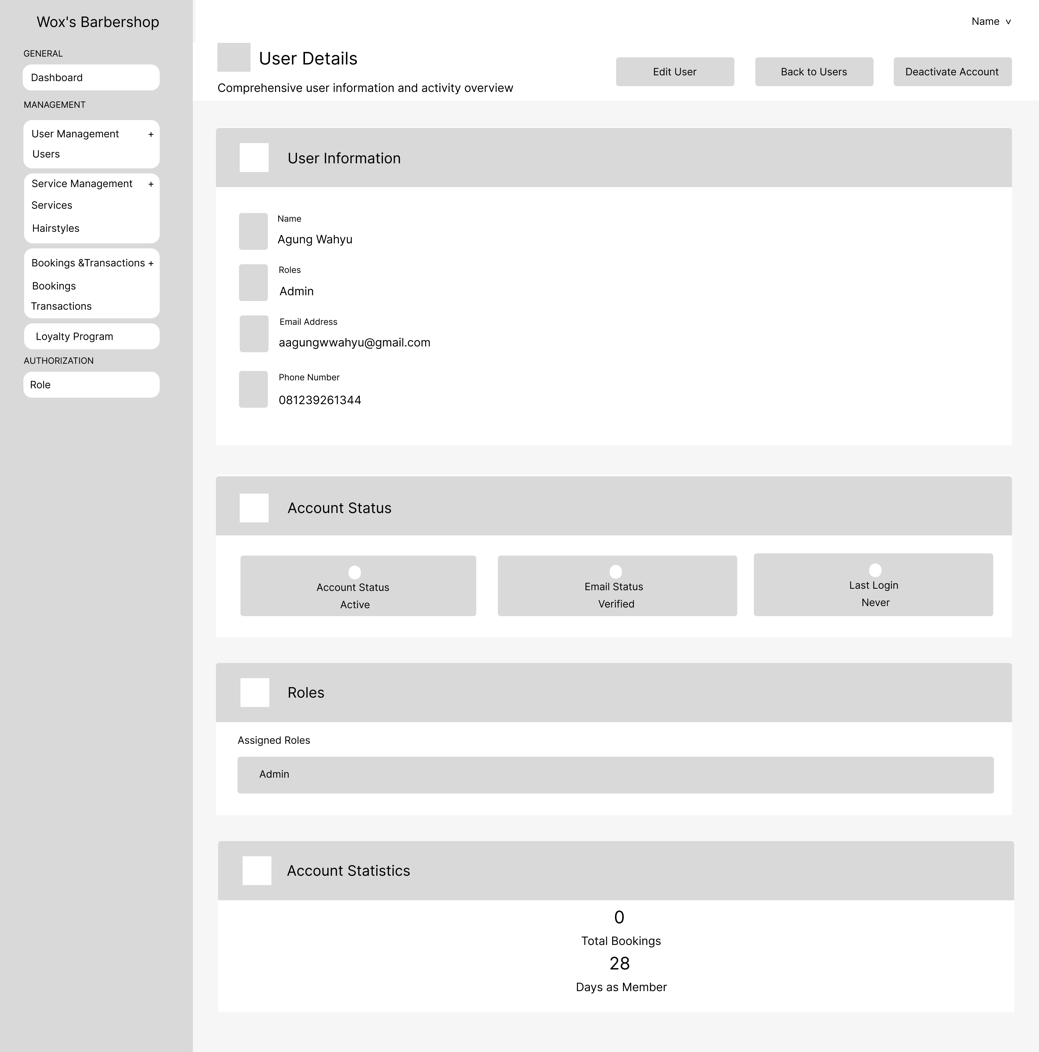
Halaman ini menampilkan create user yang digunakan untuk menambahkan data pengguna baru ke dalam sistem. Rancangan halaman create user dapat dilihat pada Gambar 4.26.



Gambar 4. 26 Rancangan Halaman Create User

1. Rancangan Halaman Show User

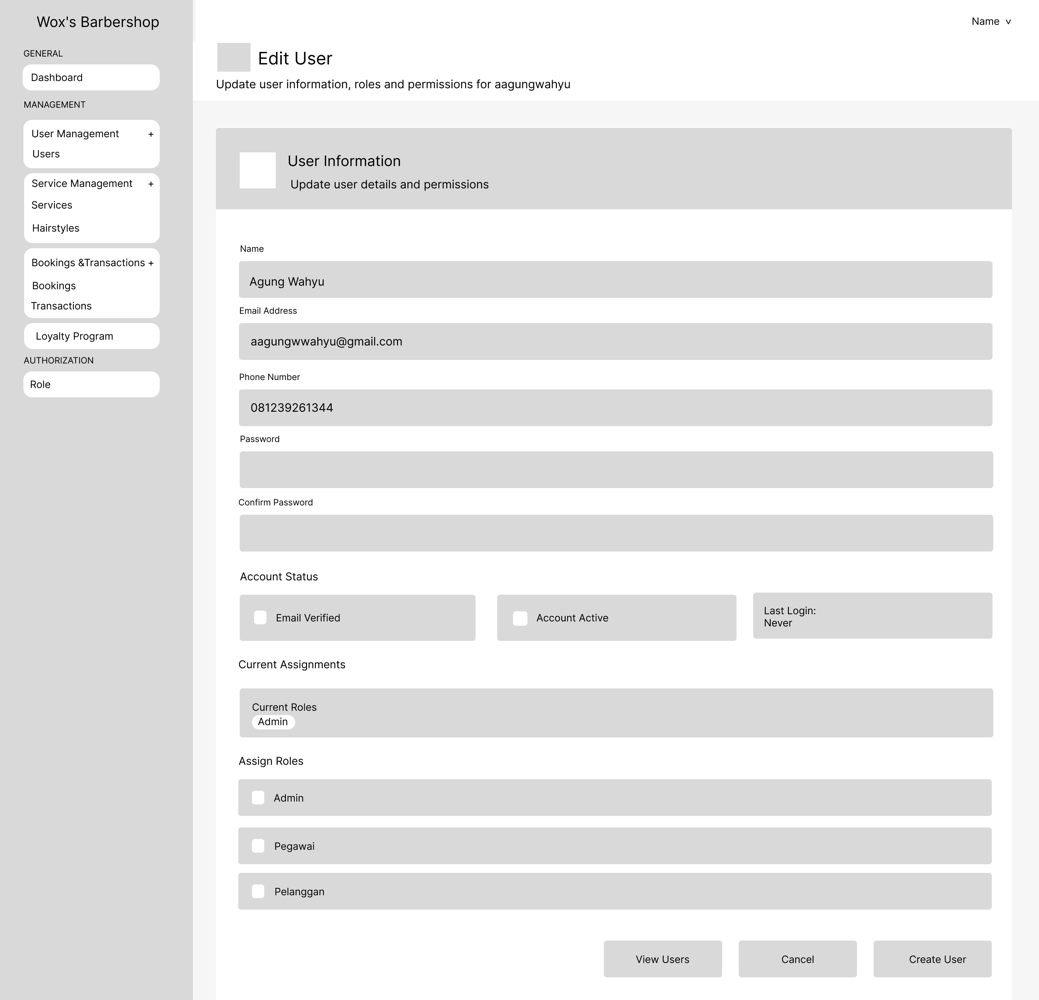
Halaman ini menampilkan data satu pengguna yang sudah terdaftar di dalam sistem. Rancangan halaman show user dapat dilihat pada Gambar 4.27.



Gambar 4. 27 Rancangan Halaman Show User

1. Rancangan Halaman Edit User

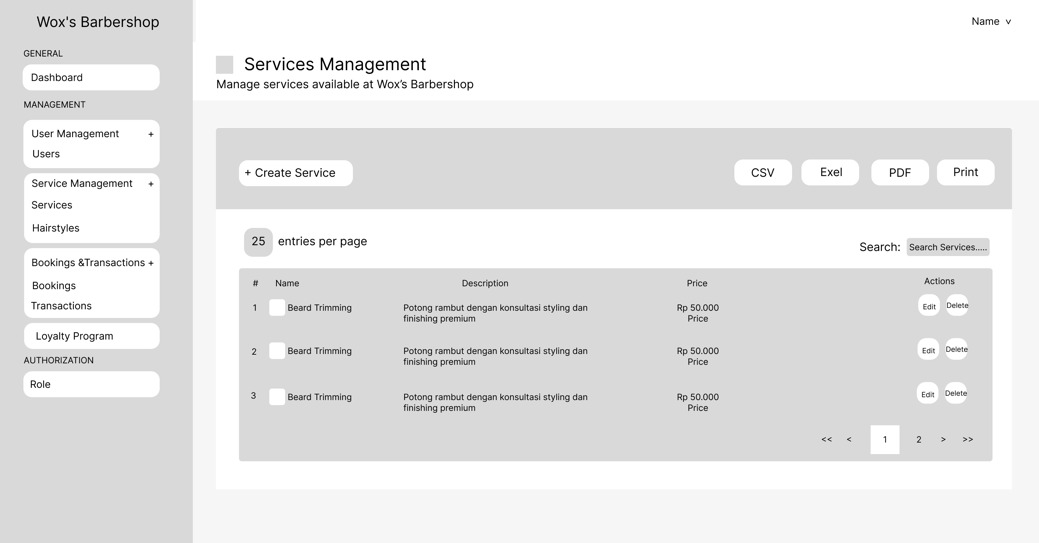
Halaman ini menampilkan data satu pengguna yang sudah terdaftar di dalam system yang bisa di edit oleh admin. Rancangan halaman edit user dapat dilihat pada Gambar 4.28.



Gambar 4. 28 Rancangan Halaman Edit User

1. Rancangan Halaman Services

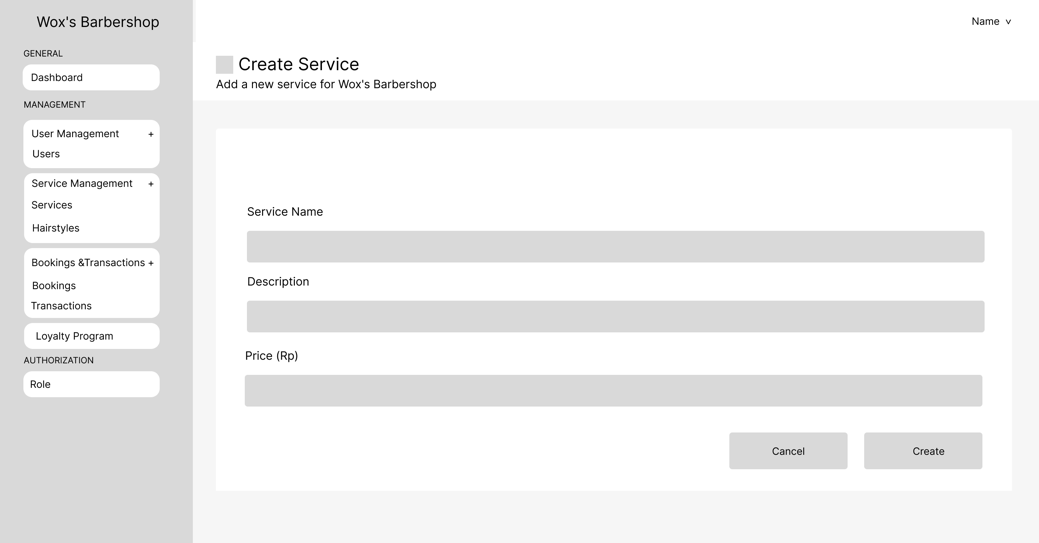
Halaman ini menampilkan data services yang dapat di kelola oleh admin. Rancangan halaman services dapat dilihat pada Gambar 4.29.



Gambar 4. 29 Rancangan Halaman Services

1. Rancangan Halaman Create Service

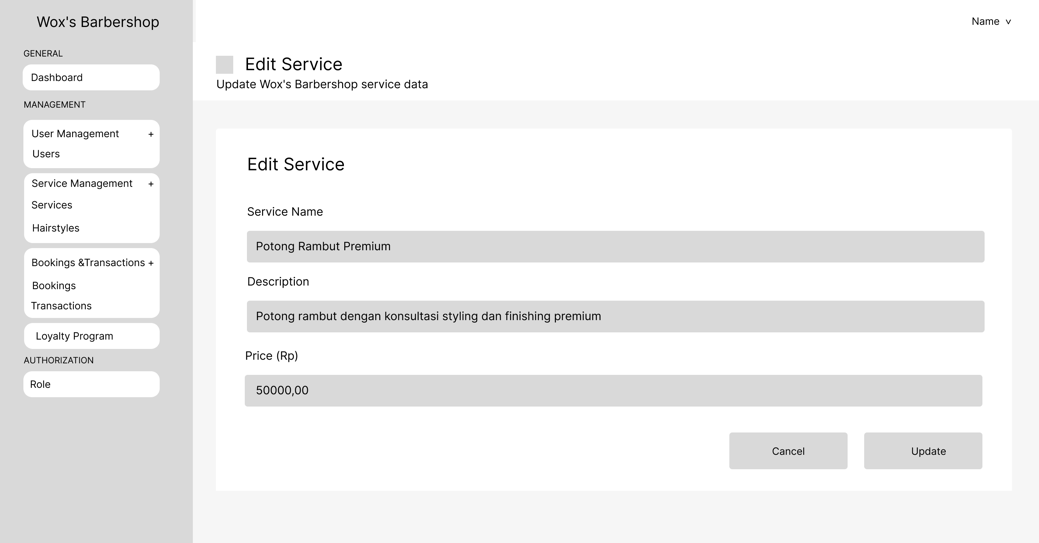
Halaman ini menampilkan create services yang digunakan untuk menambah data services. Rancangan halaman create services dapat dilihat pada Gambar 4.30.



Gambar 4. 30 Rancangan Halaman Create Service

1. Rancangan Halaman Edit Service

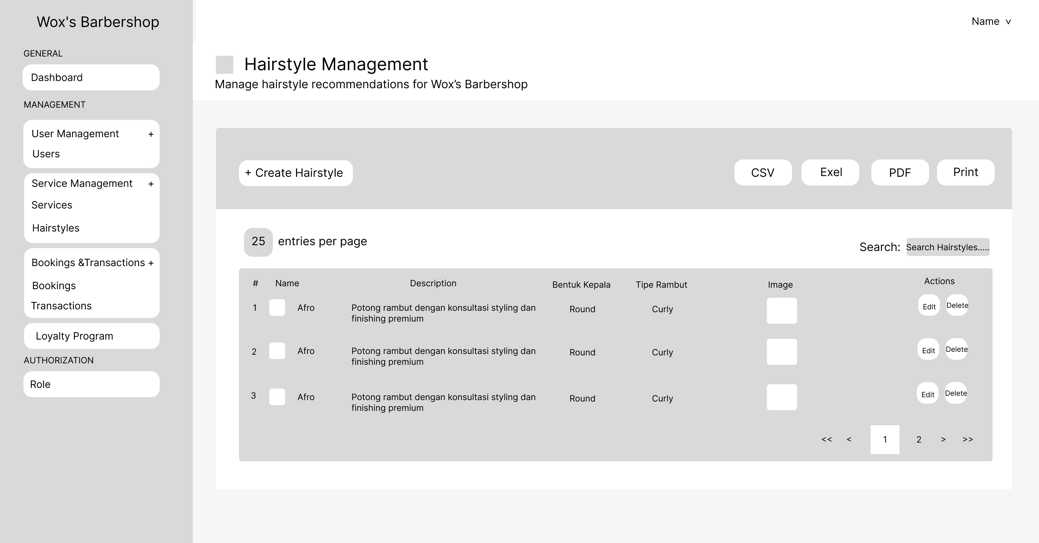
Halaman ini menampilkan edit services yang digunakan untuk melihat satu data services yang bisa diedit. Rancangan halaman edit services dapat dilihat pada Gambar 4.31.



Gambar 4. 31 Rancangan Halaman Edit Service

1. Rancangan Halaman Hairstyle

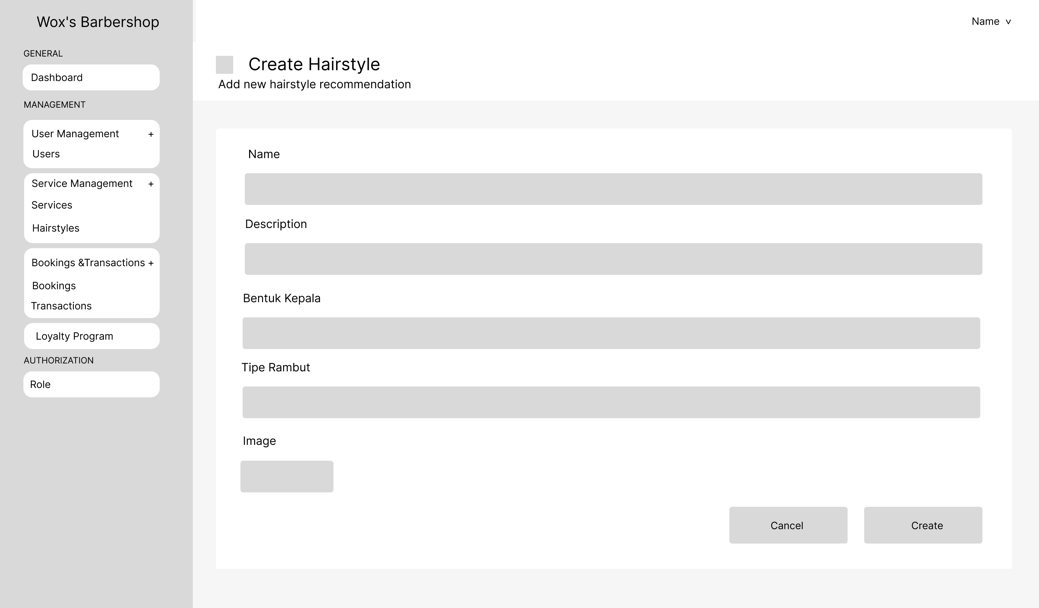
Halaman ini menampilkan data hairstyle yang dapat di kelola oleh admin. Rancangan halaman hairstyle dapat dilihat pada Gambar 4.32.



Gambar 4. 32 Rancangan Halaman Hairstyle

1. Rancangan Halaman Create Hairstyle

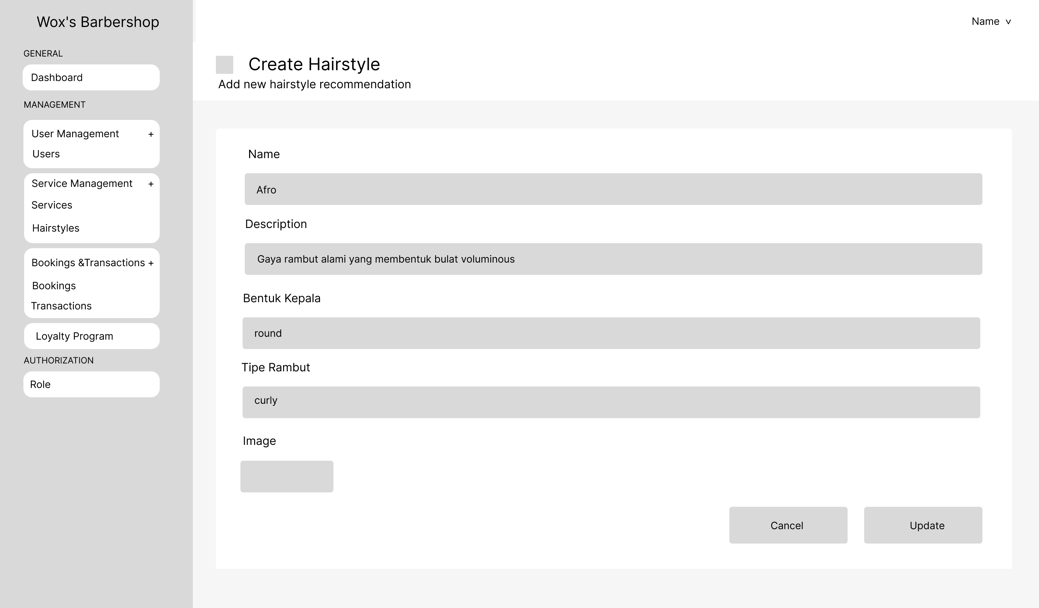
Halaman ini menampilkan create hairstyle yang digunakan untuk menambah data hairstyle. Rancangan halaman create hairstyle dapat dilihat pada Gambar 4.33.



Gambar 4. 33 Rancangan Halaman Create Hairstyle

1. Rancangan Halaman Edit Hairstyle

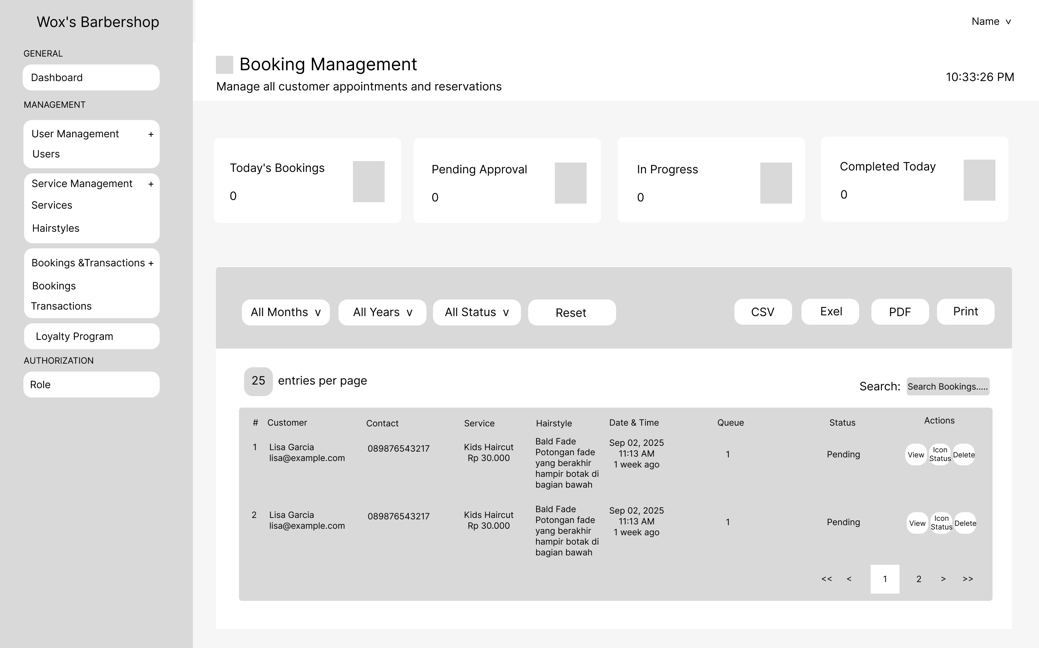
Halaman ini menampilkan edit hairstyle yang digunakan untuk melihat satu data hairstyle yang bisa diedit. Rancangan halaman edit hairstyle dapat dilihat pada Gambar 4.34.



Gambar 4. 34 Rancangan Halaman Edit Hairstyle

1. Rancangan Halaman Booking

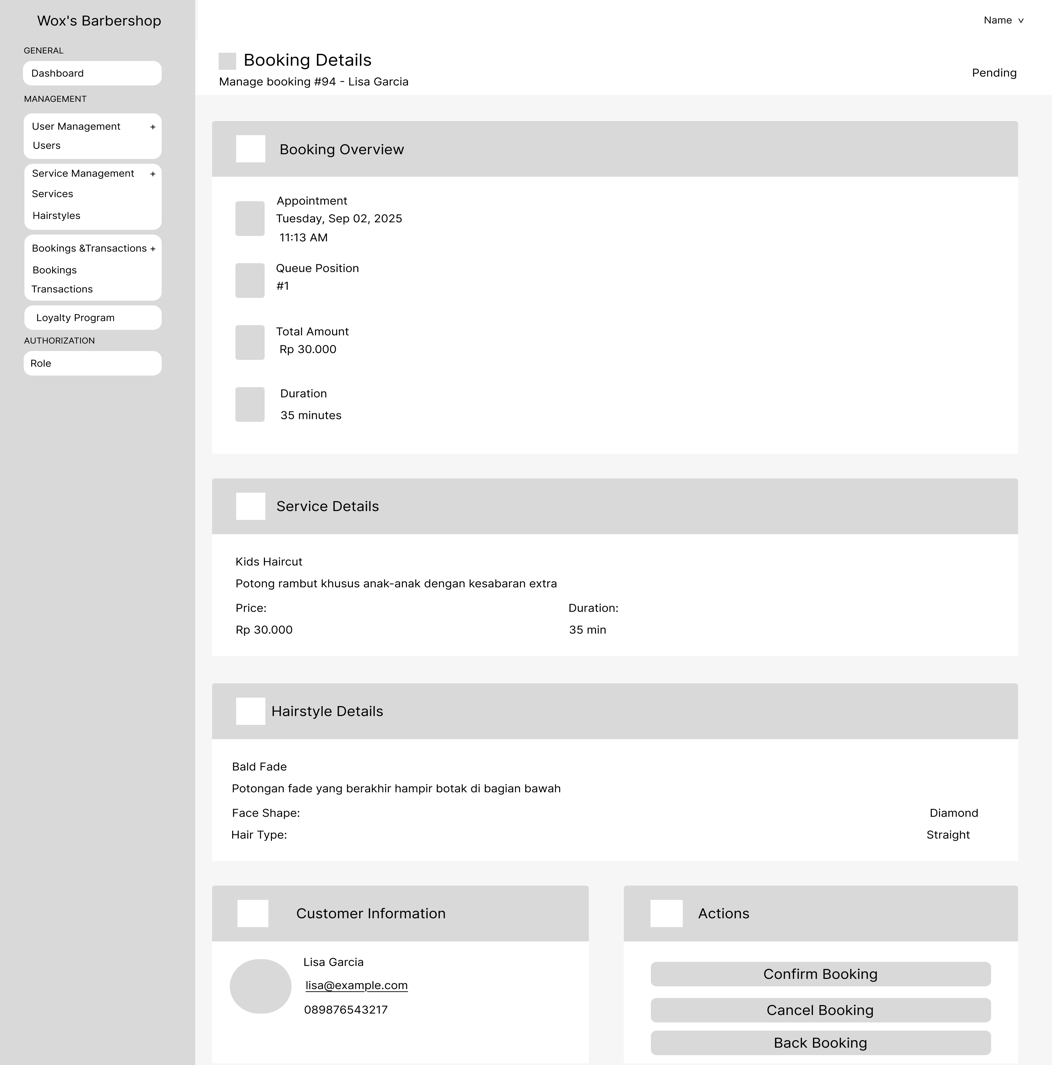
Halaman ini menampilkan data booking yang dapat dikelola oleh admin. Rancangan halaman booking dapat dilihat pada Gambar 4.35.



Gambar 4. 35 Rancangan Halaman Booking

1. Rancangan Halaman Detail Booking

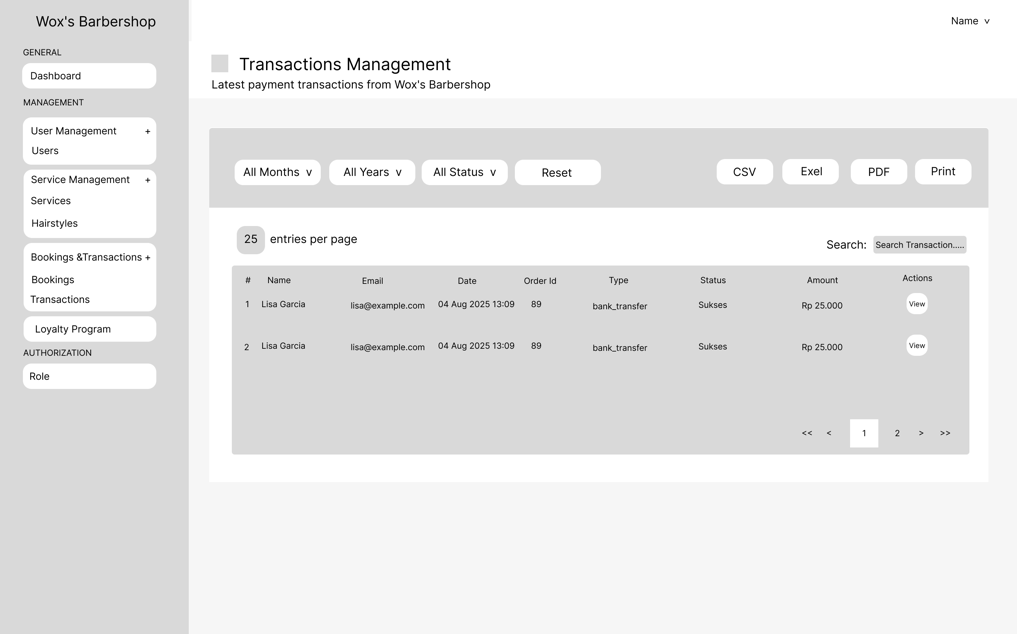
Halaman ini menampilkan data booking yang digunakan untuk melihat satu data booking. Rancangan halaman detail booking dapat dilihat pada Gambar 4.36.



Gambar 4. 36 Rancangan Halaman Detail Booking

1. Rancangan Halaman Transaction

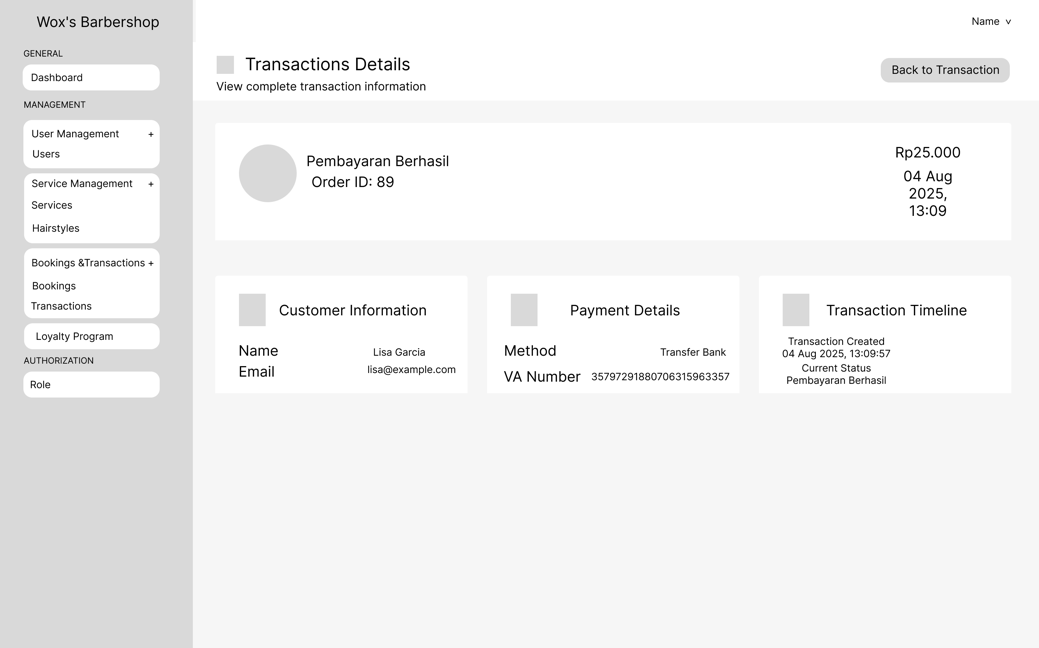
Halaman ini menampilkan data transaction yang dapat dikelola oleh admin. Rancangan halaman transactions dapat dilihat pada Gambar 4.37.



Gambar 4. 37 Rancangan Halaman Transaction

1. Rancangan Halaman Detail Transactions

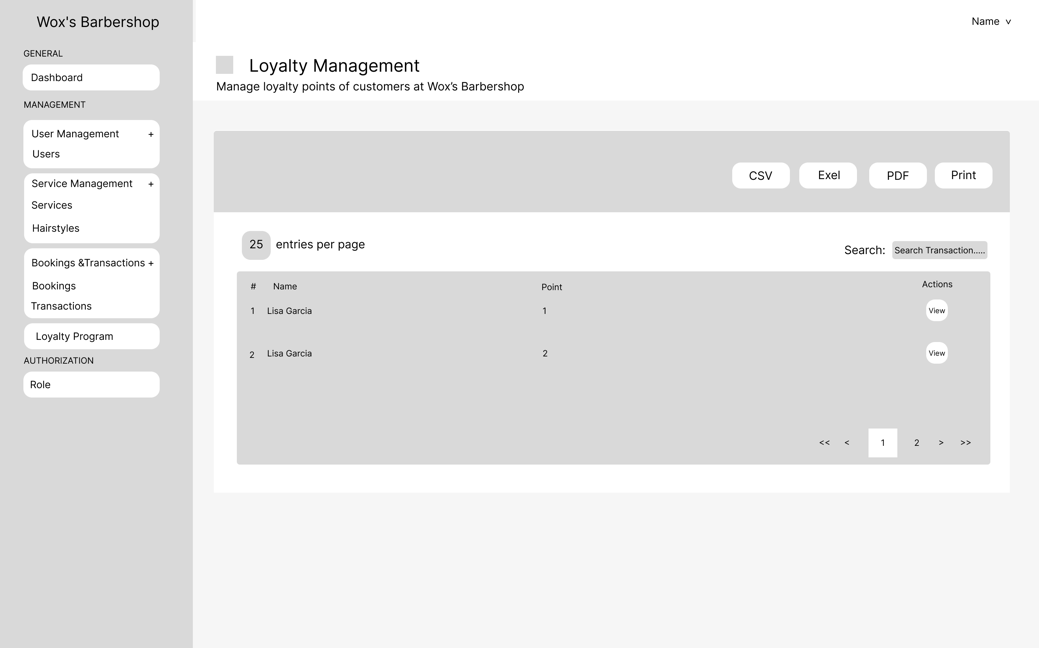
Halaman ini menampilkan data transactions yang digunakan untuk melihat satu data transactions. Rancangan halaman detail transactions dapat dilihat pada Gambar 4.38.



Gambar 4. 38 Rancangan Halaman Detail Transaction

1. Rancangan Halaman Loyalty Program

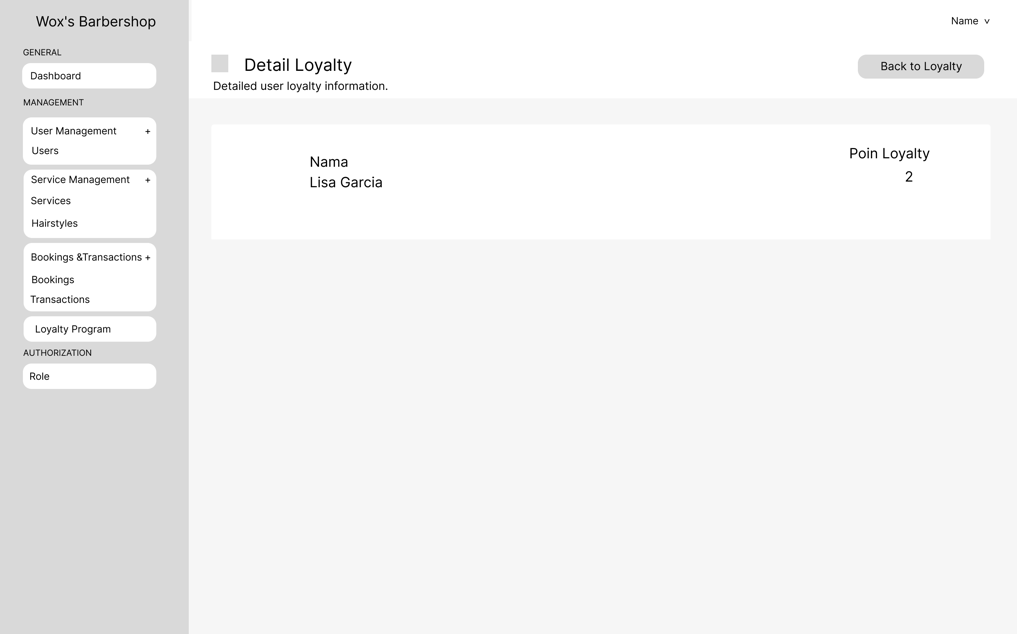
Halaman ini menampilkan data loyalty program yang dapat dikelola oleh admin. Rancangan halaman loyalty program dapat dilihat pada Gambar 4.39.



Gambar 4. 39 Rancangan Halaman Loyalty

1. Rancangan Halaman Detail Loyalty Program

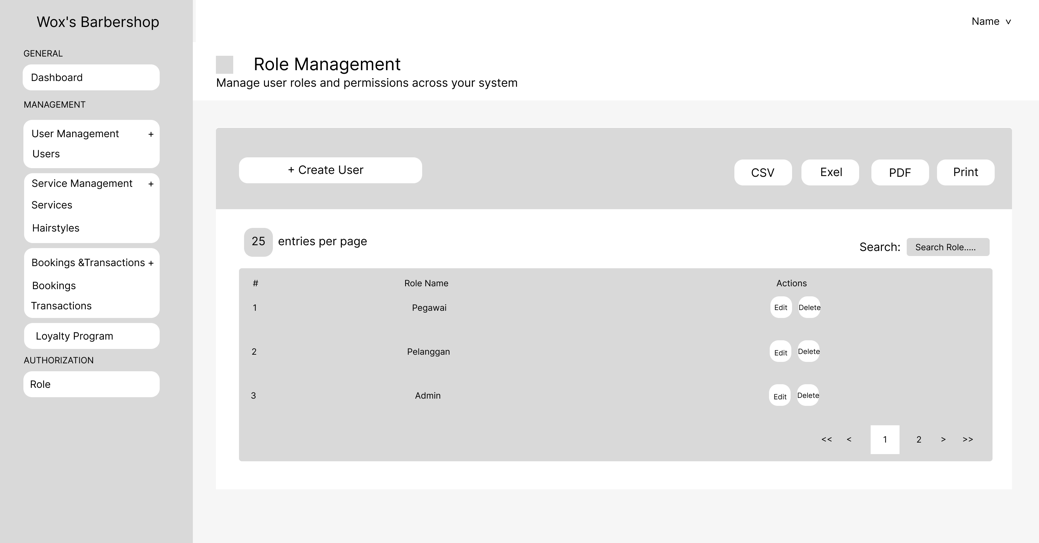
Halaman ini menampilkan data loyalty program yang digunakan untuk melihat satu data loyalty program. Rancangan halaman detail loyalty program dapat dilihat pada Gambar 4.40.



Gambar 4. 40 Rancangan Halaman Detail Loyalty

1. Rancangan Halaman Role

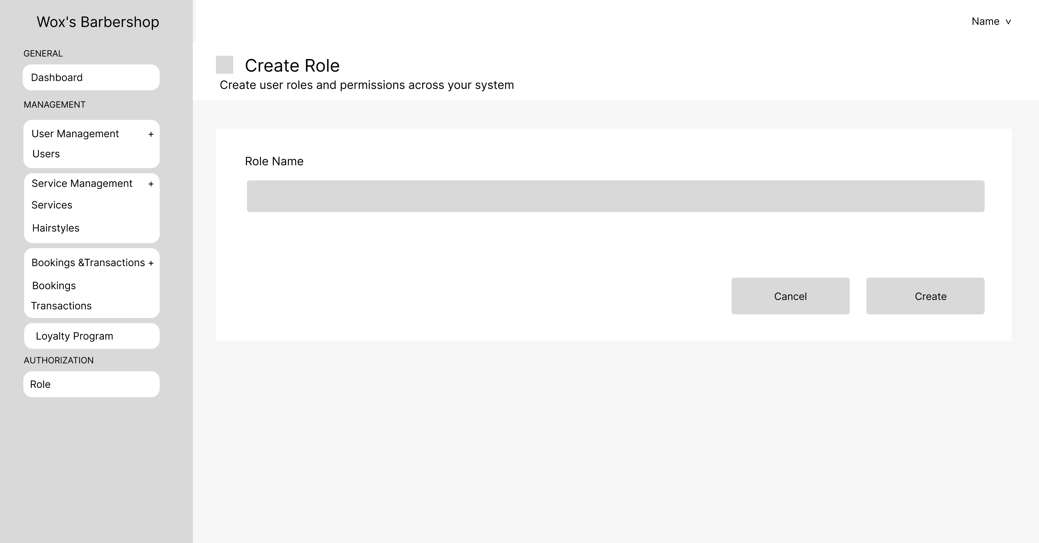
Halaman ini menampilkan data role yang dapat dikelola oleh admin. Rancangan halaman role dapat dilihat pada Gambar 4.41



Gambar 4. 41 Rancangan Halaman Role

1. Rancangan Halaman Create Role

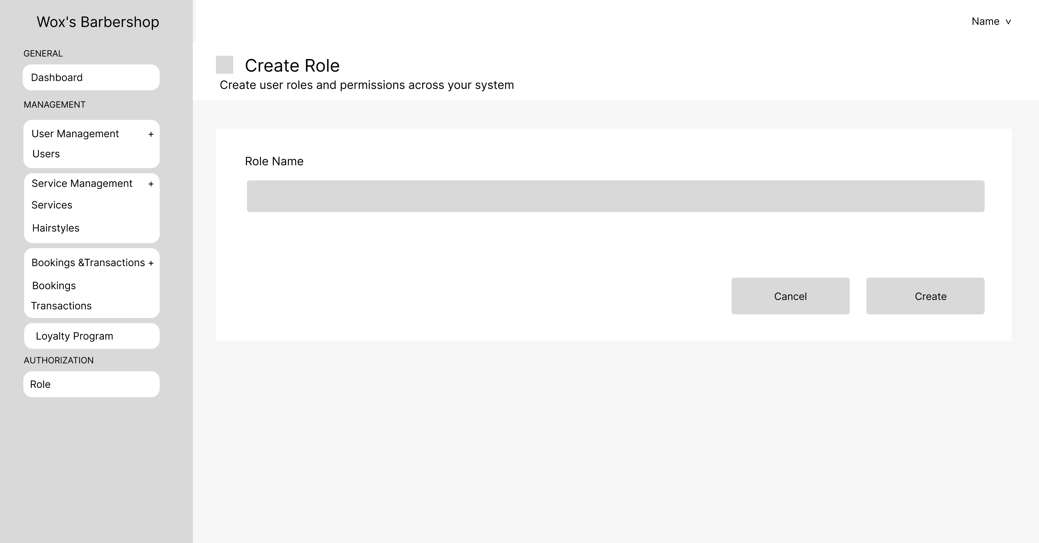
Halaman ini menampilkan create role yang digunakan untuk menambah data role. Rancangan halaman create role dapat dilihat pada Gambar 4.42.



Gambar 4. 42 Rancangan Halaman Create Role

1. Rancangan Halaman Edit Role

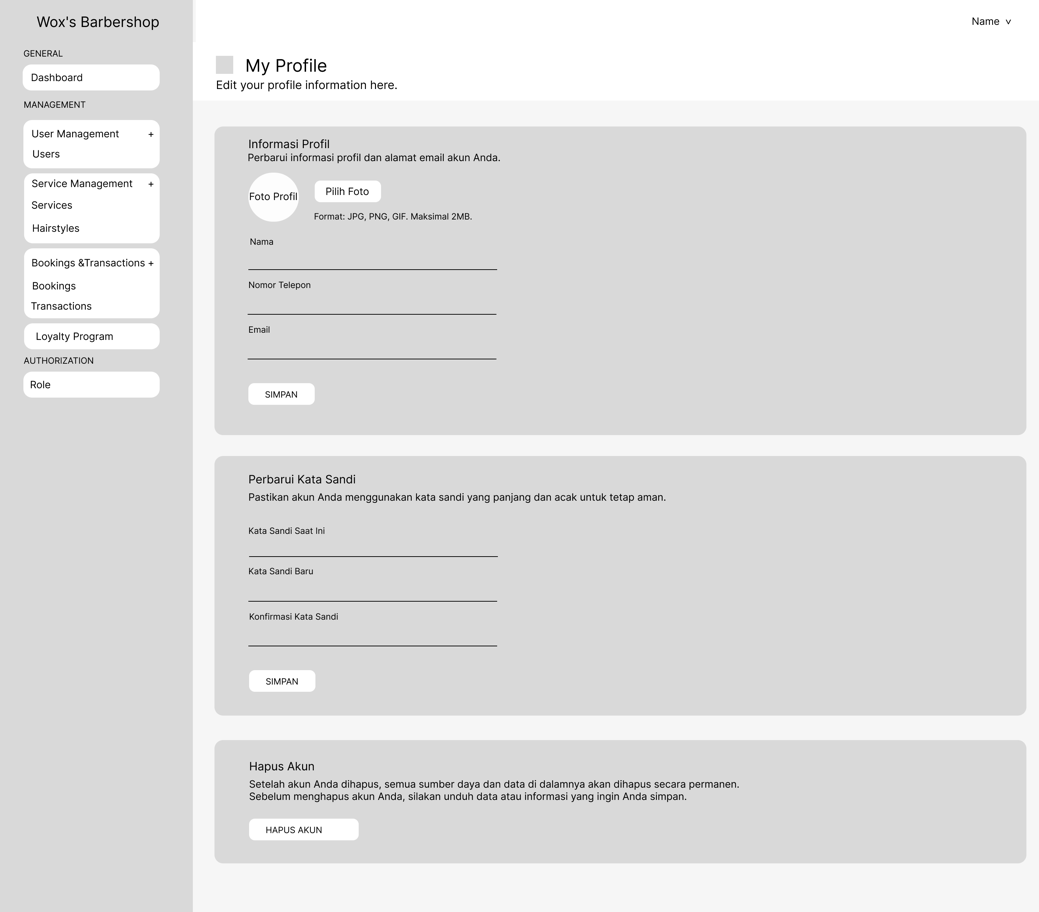
Halaman ini menampilkan edit role yang digunakan untuk melihat satu data role yang bisa diedit. Rancangan halaman edit role dapat dilihat pada Gambar 4.43.



Gambar 4. 43 Rancangan Halaman Edit Role

1. Rancangan Halaman Profile

Halaman ini menampilkan profile untuk melihat informasi pribadi admin. Rancangan halaman edit role dapat dilihat pada Gambar 4.44.

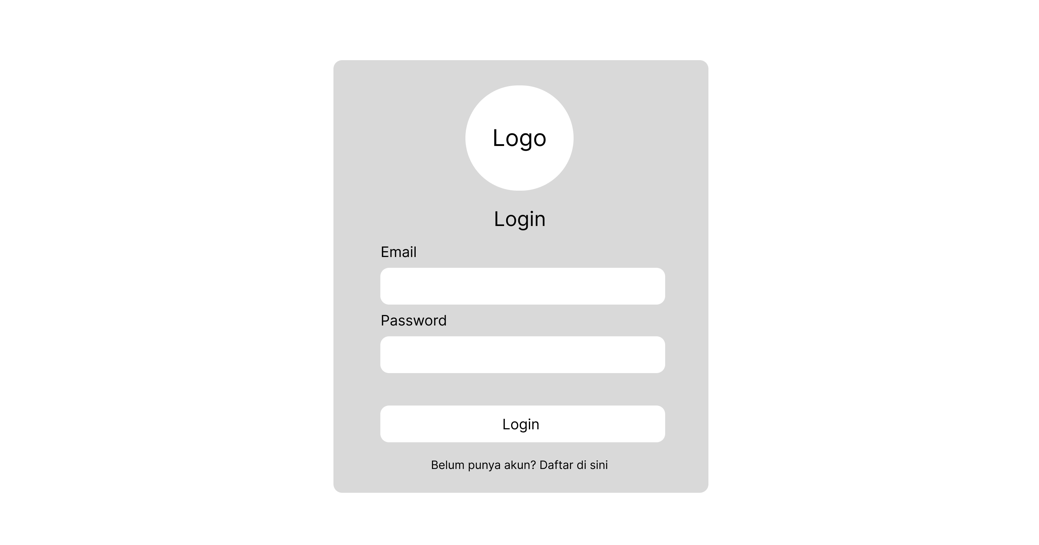


Gambar 4. 44 Rancangan Halaman Profile

### **Halaman Untuk *Pegawai***

* 1. Rancangan Halaman Login Pegawai

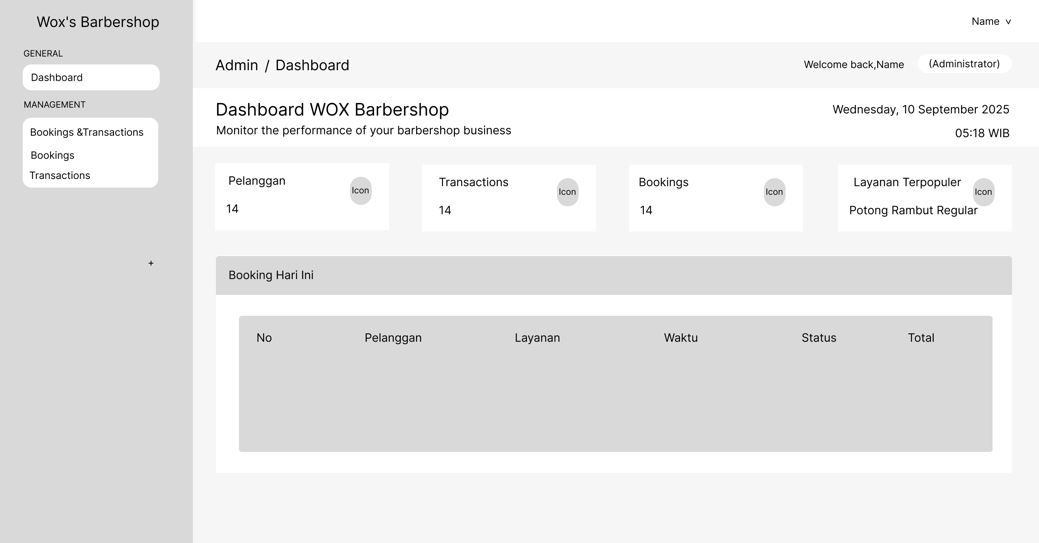
Halaman ini menampilkan autentikasi awal sebelum pegawai dapat mengakses sistem. Pegawai perlu memasukkan kredensial yang valid agar dapat masuk dan melanjutkan ke halaman dashboard. Rancangan halaman **login** pegawai dapat di lihat pada Gambar 4.45.



Gambar 4. 45 Rancangan Halaman Login Pegawai

1. Rancangan Halaman Dashboard Pegawai

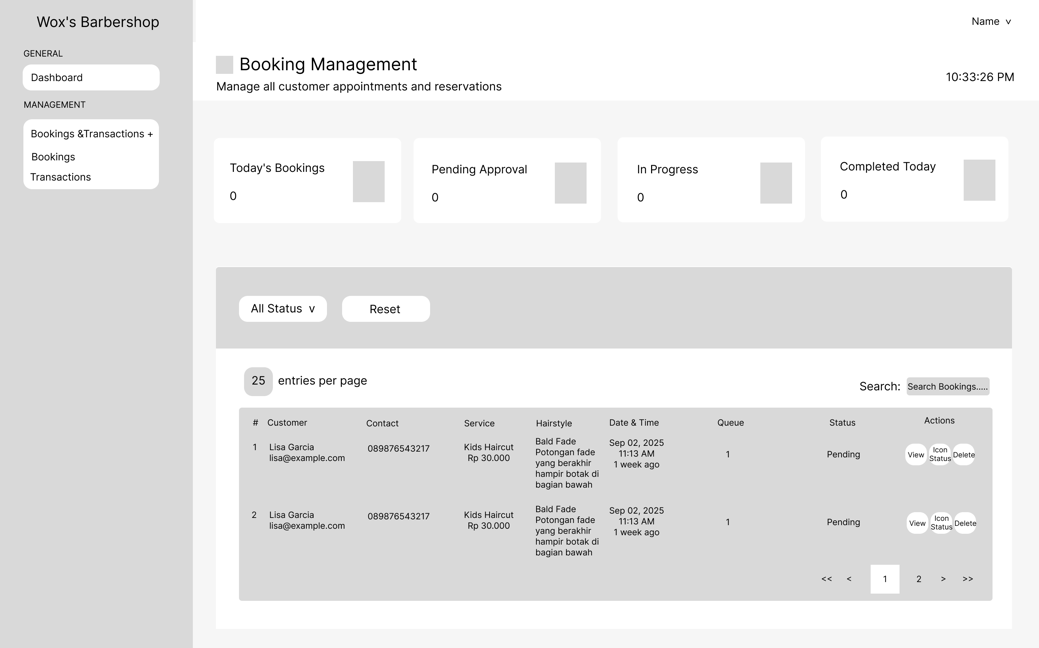
Halaman ini menampilkan ringkasan informasi utama yang membantu pegawai memantau kondisi sistem secara cepat dan efisien. Rancangan halaman **dashboard** pegawai dapat di lihat pada Gambar 4.46.



Gambar 4. 46 Rancangan Halaman Dashboard Pegawai

* 1. Rancangan Halaman Booking

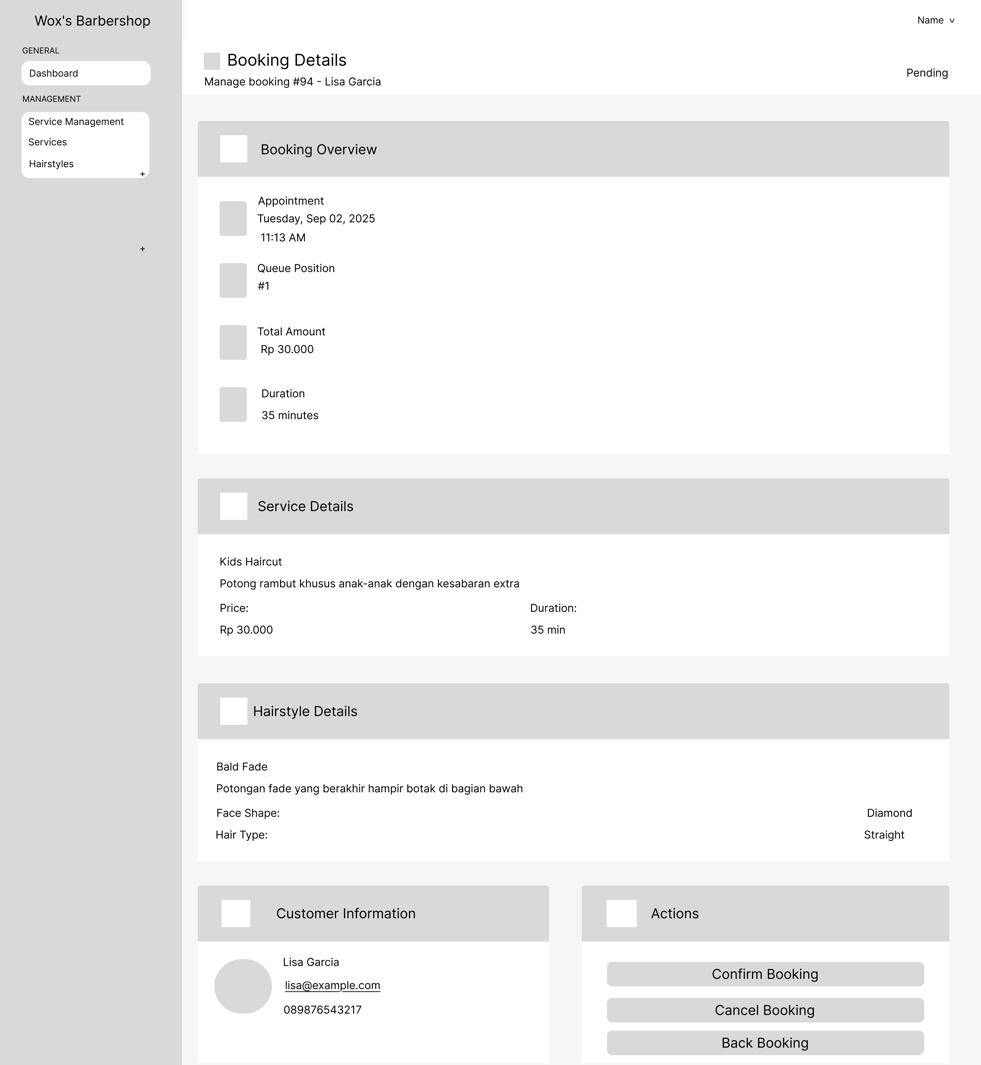
Halaman ini menampilkan data booking yang dapat dikelola oleh pegawai. Rancangan halaman booking dapat dilihat pada Gambar 4.47.



Gambar 4. 47 Rancangan Halaman Booking

* 1. Rancangan Halaman Detail Booking

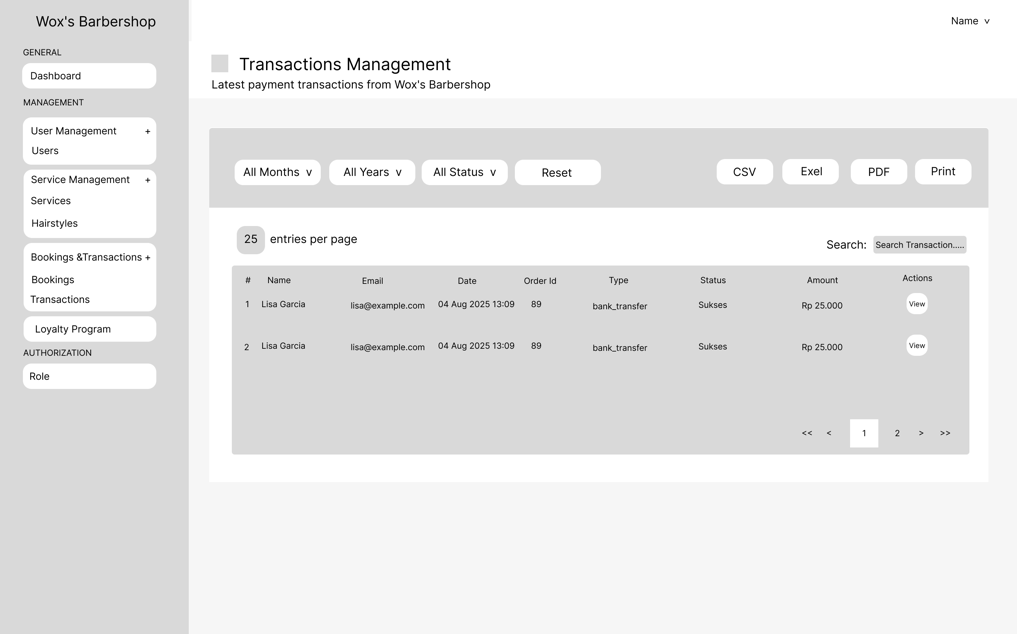
Halaman ini menampilkan satu data booking yang digunakan untuk melihat detailnya. Rancangan halaman detail booking dapat dilihat pada Gambar 4.48.



Gambar 4. 48 Rancangan Halaman Detail Booking

* 1. Rancangan Halaman Transaction

Halaman ini menampilkan data transaction yang dapat dikelola oleh pegawai. Rancangan halaman transactions dapat dilihat pada Gambar 4.49.



Gambar 4. 49 Rancangan Halaman Transaction

* 1. Rancangan Halaman Detail Transactions

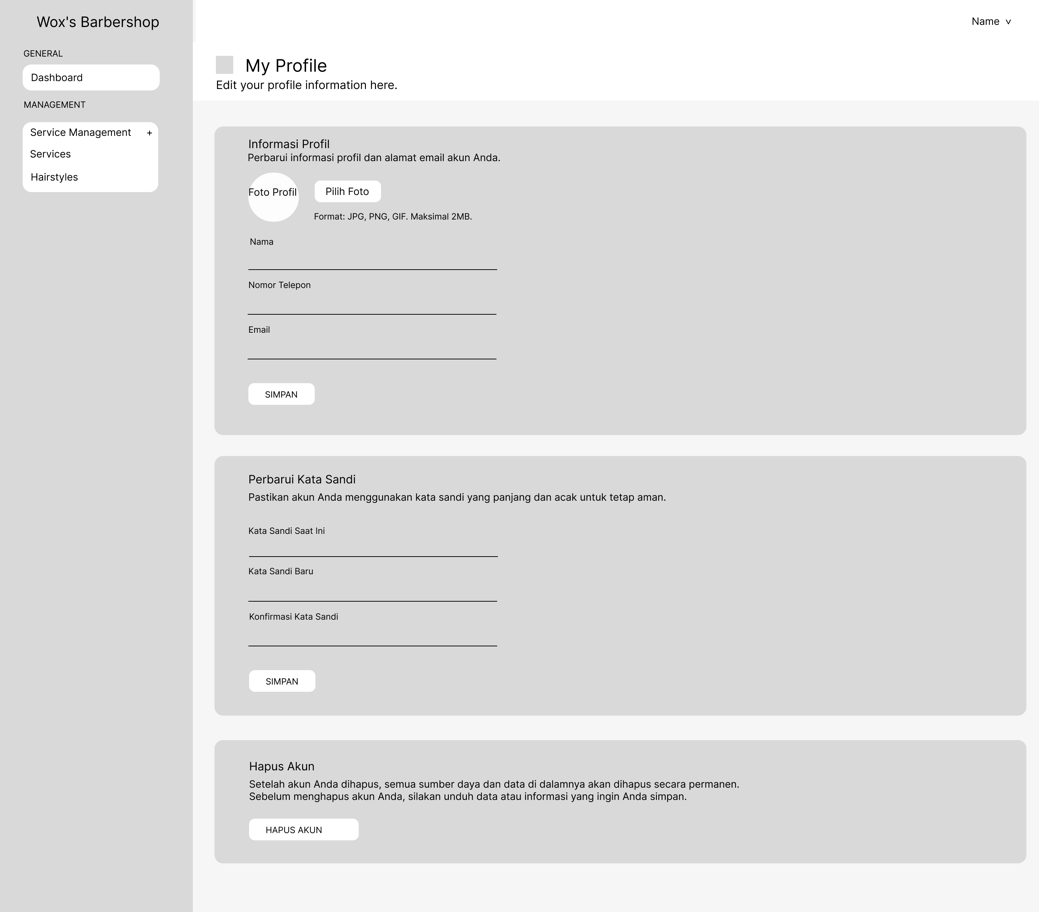
Halaman ini menampilkan data transactions yang digunakan untuk melihat satu data transactions. Rancangan halaman detail transactions dapat dilihat pada Gambar 4.50 .



Gambar 4. 50 Rancangan Halaman Detail Transaction

* 1. Rancangan Halaman Profile

Halaman ini menampilkan profile untuk melihat informasi pribadi pegawai. Rancangan halaman edit role dapat dilihat pada Gambar 4.51.



Gambar 4. 51 Rancangan Halaman Profile

## **Implementasi Sistem**

Tahap implementasi sistem dilakukan dengan membangun aplikasi web berdasarkan rancangan yang telah disusun sebelumnya. Proses implementasi ini mencakup pembangunan antarmuka, integrasi database, serta penerapan fungsionalitas utama seperti pemesanan online, rekomendasi gaya rambut, dan manajemen operasional. Implementasi ini merupakan bagian dari tahap Construction of Prototype pada metode Prototype.

### **Halaman Untuk *User***

**1. Implementasi Halaman Landing Page**  
Halaman landing page sudah diimplementasikan dengan tampilan modern dan responsif menggunakan Tailwind CSS. Halaman ini menampilkan informasi layanan, promosi, serta tombol login dan registrasi.  
Gambar 4.xx Implementasi Halaman Landing Page Pelanggan

**2. Implementasi Halaman Registrasi**  
Form registrasi pelanggan sudah diimplementasikan dengan validasi data, memungkinkan pelanggan membuat akun baru dengan mudah.  
Gambar 4.xx Implementasi Halaman Registrasi Pelanggan

**3. Implementasi Halaman Login**  
Halaman login sudah dilengkapi validasi email dan password, serta notifikasi error jika login gagal.  
Gambar 4.xx Implementasi Halaman Login Pelanggan

**4. Implementasi Halaman Dashboard**  
Dashboard pelanggan menampilkan status pemesanan terakhir, loyalty point, serta menu navigasi utama.  
Gambar 4.xx Implementasi Halaman Dashboard Pelanggan

**5. Implementasi Halaman Pemesanan Layanan**  
Pelanggan dapat memilih layanan, stylist, serta menjadwalkan waktu pemesanan secara interaktif.  
Gambar 4.xx Implementasi Halaman Pemesanan Layanan

**6. Implementasi Halaman Riwayat Pemesanan**  
Halaman riwayat menampilkan daftar transaksi dan status pemesanan pelanggan.  
Gambar 4.xx Implementasi Halaman Riwayat Pemesanan

**7. Implementasi Halaman Rekomendasi Gaya Rambut**  
Sistem menampilkan rekomendasi gaya rambut sesuai data pelanggan.  
Gambar 4.xx Implementasi Halaman Rekomendasi Gaya Rambut

**8. Implementasi Halaman Loyalty Reward**  
Progress reward pelanggan ditampilkan dalam bentuk visual (progress bar).  
Gambar 4.xx Implementasi Halaman Loyalty Reward

**9. Implementasi Halaman Profil**  
Halaman profil memungkinkan pelanggan memperbarui data diri dan foto profil.  
Gambar 4.xx Implementasi Halaman Profil Pelanggan

**10. Implementasi Halaman Notifikasi**  
Pelanggan dapat melihat notifikasi terkait jadwal, promo, dan pemesanan.  
Gambar 4.xx Implementasi Halaman Notifikasi Pelanggan

**11. Implementasi Halaman Ulasan Layanan**  
Pelanggan dapat memberikan ulasan berupa rating dan komentar.  
Gambar 4.xx Implementasi Halaman Ulasan Pelanggan

**12. Implementasi Halaman Logout**  
Halaman ini menampilkan konfirmasi sebelum keluar dari sistem.  
Gambar 4.xx Implementasi Halaman Logout Pelanggan

### **Halaman Untuk *Admin***

**1. Implementasi Halaman Login Admin**  
Halaman login untuk admin diimplementasikan dengan form sederhana yang mendukung validasi username dan password. Jika data salah, sistem akan menampilkan pesan kesalahan.  
Gambar 4.xx Implementasi Halaman Login Admin

**2. Implementasi Halaman Dashboard Admin**  
Dashboard admin menampilkan ringkasan data pengguna, transaksi, serta grafik laporan untuk mempermudah pemantauan sistem.  
Gambar 4.xx Implementasi Halaman Dashboard Admin

**3. Implementasi Halaman Manajemen User**  
Admin dapat mengelola data pelanggan dan pegawai, termasuk menambahkan, mengedit, dan menghapus data.  
Gambar 4.xx Implementasi Halaman Manajemen User

**4. Implementasi Halaman Manajemen Layanan**  
Admin dapat menambahkan layanan baru, mengubah detail layanan, serta menghapus layanan yang tidak aktif.  
Gambar 4.xx Implementasi Halaman Manajemen Layanan

**5. Implementasi Halaman Manajemen Hairstyle**  
Halaman ini memungkinkan admin mengunggah gambar hairstyle, memberi deskripsi, serta mengatur kategori gaya rambut.  
Gambar 4.xx Implementasi Halaman Manajemen Hairstyle

**6. Implementasi Halaman Transaksi**  
Admin dapat melihat daftar transaksi yang dilakukan pelanggan, termasuk status pembayaran dan detail pesanan.  
Gambar 4.xx Implementasi Halaman Transaksi Admin

**7. Implementasi Halaman Laporan Keuangan**  
Halaman laporan keuangan menampilkan pendapatan bulanan, pengeluaran, dan keuntungan dalam bentuk tabel serta grafik.  
Gambar 4.xx Implementasi Halaman Laporan Keuangan

**8. Implementasi Halaman Laporan Operasional**  
Admin dapat melihat laporan aktivitas harian pegawai, termasuk jumlah layanan yang telah diselesaikan.  
Gambar 4.xx Implementasi Halaman Laporan Operasional

**9. Implementasi Halaman Pengaturan Sistem**  
Halaman ini digunakan untuk mengatur informasi umum sistem seperti nama barbershop, kontak, dan jam operasional.  
Gambar 4.xx Implementasi Halaman Pengaturan Sistem

**10. Implementasi Halaman Notifikasi Admin**  
Admin dapat melihat notifikasi terkait pemesanan baru, pembatalan, dan ulasan pelanggan.  
Gambar 4.xx Implementasi Halaman Notifikasi Admin

**11. Implementasi Halaman Profil Admin**  
Profil admin dapat diperbarui melalui halaman ini, termasuk mengganti foto profil dan password.  
Gambar 4.xx Implementasi Halaman Profil Admin

**12. Implementasi Halaman Logout Admin**  
Halaman logout menampilkan konfirmasi sebelum admin keluar dari sistem untuk menjaga keamanan akun.  
Gambar 4.xx Implementasi Halaman Logout Admin

### **Halaman Untuk *Pegawai***

**1. Implementasi Halaman Login Pegawai**  
Halaman login pegawai diimplementasikan dengan form sederhana untuk masuk ke dalam sistem.  
Gambar 4.xx Implementasi Halaman Login Pegawai

**2. Implementasi Halaman Dashboard Pegawai**  
Dashboard pegawai menampilkan jadwal layanan, notifikasi pemesanan baru, dan ringkasan aktivitas harian.  
Gambar 4.xx Implementasi Halaman Dashboard Pegawai

**3. Implementasi Halaman Pengelolaan Jadwal**  
Pegawai dapat mengatur jadwal kerja serta melihat jadwal layanan yang sudah dipesan pelanggan.  
Gambar 4.xx Implementasi Halaman Pengelolaan Jadwal

**4. Implementasi Halaman Konfirmasi Pemesanan**  
Halaman ini memungkinkan pegawai untuk menerima atau menolak pemesanan pelanggan.  
Gambar 4.xx Implementasi Halaman Konfirmasi Pemesanan

**5. Implementasi Halaman Layanan Berlangsung**  
Pegawai dapat melihat layanan yang sedang berlangsung dan mencatat progres pengerjaan.  
Gambar 4.xx Implementasi Halaman Layanan Berlangsung

**6. Implementasi Halaman Komplain Pelanggan**  
Pegawai dapat membaca dan menindaklanjuti komplain pelanggan terkait layanan yang diberikan.  
Gambar 4.xx Implementasi Halaman Komplain Pelanggan

**7. Implementasi Halaman Riwayat Layanan**  
Halaman ini menampilkan daftar layanan yang pernah ditangani pegawai sebelumnya.  
Gambar 4.xx Implementasi Halaman Riwayat Layanan

**8. Implementasi Halaman Profil Pegawai**  
Pegawai dapat memperbarui data pribadi, termasuk kontak dan foto profil.  
Gambar 4.xx Implementasi Halaman Profil Pegawai

**9. Implementasi Halaman Notifikasi Pegawai**  
Pegawai dapat melihat notifikasi terkait pemesanan baru, pembatalan, atau perubahan jadwal.  
Gambar 4.xx Implementasi Halaman Notifikasi Pegawai

**10. Implementasi Halaman Penilaian Kinerja**  
Halaman ini menampilkan penilaian dan ulasan dari pelanggan terhadap layanan yang diberikan pegawai.  
Gambar 4.xx Implementasi Halaman Penilaian Pegawai

**11. Implementasi Halaman Pengaturan Akun Pegawai**  
Pegawai dapat mengubah password dan preferensi akun melalui halaman ini.  
Gambar 4.xx Implementasi Halaman Pengaturan Akun Pegawai

**12. Implementasi Halaman Logout Pegawai**  
Halaman logout menampilkan konfirmasi sebelum pegawai keluar dari sistem.  
Gambar 4.xx Implementasi Halaman Logout Pegawai

## **Pengujian Sistem**

Pengujian sistem dilakukan menggunakan metode **Black Box Testing** untuk memverifikasi apakah fungsi sistem berjalan sesuai kebutuhan. Setiap pengujian berfokus pada input, output, dan reaksi sistem.

Tabel 4.xx Rangkuman Hasil Black Box Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario Uji** | **Input** | **Output yang Diharapkan** | **Output Aktual** | **Status** |
| 1 | Login dengan data benar | Username & Password valid | Masuk ke dashboard | Masuk ke dashboard | Berhasil |
| 2 | Login dengan data salah | Username/Password salah | Muncul pesan error | Muncul pesan error | Berhasil |
| 3 | Pemesanan layanan | Pilih layanan, tanggal, jam | Pesanan tersimpan | Pesanan tersimpan | Berhasil |
| 4 | Pemesanan layanan tanpa memilih jam | Pilih layanan & tanggal, tanpa jam | Pesan error "Jam belum dipilih" | Pesan error tampil | Berhasil |
| 5 | Riwayat transaksi | Klik menu Riwayat | Data transaksi tampil | Data transaksi tampil | Berhasil |
| 6 | Pengelolaan user (Admin) | Tambah data user baru | Data tersimpan ke database | Data tersimpan | Berhasil |
| 7 | Pengelolaan jadwal (Pegawai) | Update jadwal layanan | Jadwal tersimpan | Jadwal tersimpan | Berhasil |
| 8 | Logout | Klik tombol logout | Sistem keluar ke halaman login | Sistem keluar | Berhasil |

## **Pengujian *User***

Selain *black box testing*, pengujian sistem juga akan menggunakan metode System Usability Scale (SUS). Kuesioner SUS akan digunakan untuk mengukur tingkat kegunaan sistem berdasarkan persepsi subjektif pengguna, sehingga dapat diketahui sejauh mana sistem ini mudah digunakan dan diterima oleh pengguna. Hasil pengujian *user* akan memberikan umpan balik penting untuk perbaikan dan peningkatan kualitas sistem. Tahap pengujian ini juga merupakan bagian dari "Deployment, Delivery & Feedback" pada metode Prototype. Tabel terkait: Akan ada tabel yang menyajikan hasil kuesioner SUS, termasuk skor rata-rata SUS dan interpretasinya.

**BAB V  
PENUTUP**

* 1. **Kesimpulan**
  2. **Penutup**

**DAFTAR PUSTAKA**

[1] R. Goecke, “*The Evolution Of Online Booking Systems*,” dalam *Handbook Of E-Tourism*, 2022. Doi: 10.1007/978-3-030-48652-5\_27.

[2] I. K. P. Dwiutama, “Sistem Informasi *Booking Barbershop* Bad Punz Bali Menggunakan *Framework* *Laravel*,” Skripsi, Stikom Bali : Denpasar, 2023.

[3] D. Muriyatmoko, “Sistem Informasi Manajemen *Barbershop* Unit Usaha Unida Gontor (U3) Berbasis *Website*,” *Jurnal Informatika Polinema*, Vol. 9, No. 3, 2023, Doi: 10.33795/Jip.V9i3.1274.

[4] R. Setiawan, D. Kurniadi, dan M. Saleh, “Rancang Bangun Aplikasi Booking Dan Transaksi *Barbershop* Berbasis Web,” *Jurnal Algoritma*, Vol. 17, No. 2, 2021, Doi: 10.33364/Algoritma/V.17-2.452.

[5] A. Syahputra, R. Wiranti, dan W. Astita, “Peran Sistem Informasi Manajemen Organisasi Dalam Pengambilan Keputusan,” *Jurnal Manajemen Sistem Informasi (Jmasif)*, Vol. 1, No. 1, 2022, Doi: 10.35870/Jmasif.V1i1.67.

[6] E. Nurlailah Dan K. R. N. Wardani, “Perancangan *Website* Sebagai Media Informasi Dan Promosi Oleh-Oleh Khas Kota Pagaralam,” *Jipi (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, Vol. 8, No. 4, 2023, Doi: 10.29100/Jipi.V8i4.4006.

[7] H. Sama Dan E. Hartanto, “Studi Deskriptif Evolusi *Website* Dari Html1 Sampai Html5 Dan Pengaruhnya Terhadap Perancangan Dan Pengembangan Website,” *Conference On Management, Business, Innovation, Education And Social Sciences (Combines)*, Vol. 1, No. 1, 2021.

[8] F. Rifandi, T. V. Adriansyah, Dan R. Kurniawati, “*Website Gallery Development Using Tailwind Css Framework,*” *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, Vol. 6, No. 2, 2022, Doi: 10.37339/E-Komtek.V6i2.937.

[9] M. Dirgaf, Masnur, Dan Merlina, “Pengertian *Javascript*,” *Jurnal Sintaks Logika*, Vol. 1, No. 1, 2021.

[10] Y. T. Utami, R. Sofyan, Dan Rd. I. A. Pribadi, “Perancangan Dan Implementasi *Website* Layanan Akademik Di SMA Negeri 07 Bandar Lampung,” *Inovasi Pembangunan : Jurnal Kelitbangan*, Vol. 9, No. 02, 2021, Doi: 10.35450/Jip.V9i02.255.

[11] A. Yunisa dan R. Amalia, “Perancangan Sistem Informasi Manajemen Jasa Fotografi Berbasis *Website* Menggunakan *Framework Laravel* (Studi Kasus: Haydey Moment),” *Jurnal Informatika Multi*, Vol. 1, No. 1, 2023.

[12] D. P. A. Erlangga, D. Hartanti, Dan H. Lubis, “Perancangan Sistem Informasi Geografis Sekolah Luar Biasa Dengan Metode *Extreme* *Programming*,” *Journal Of Informatic And Information Security*, Vol. 3, No. 1, 2022, Doi: 10.31599/Jiforty.V3i1.1220.

[13] J. Wahyudi, M. Asbari, I. Sasono, T. Pramono, Dan D. Novitasari, “*Database* *Management Education In Mysql*,” *Edumaspul: Jurnal Pendidikan*, Vol. 6, No. 2, 2022, Doi: 10.33487/Edumaspul.V6i2.4570.

[14] J. R. Fauzi, “Algoritma Dan *Flowchart* Dalam Menyelesaikan Suatu Masalah,” *Jurnal Hukum Progresif.*, No. 20330044, 2023.

[15] S. Al-Fedaghi, “*Classes In Object-Oriented Modeling* *(Uml):* *Further Understanding And Abstraction,”* Jun 2021, Doi: 10.22937/Ijcsns.2021.21.5.21.

[16] R. Nurmasari, S. Pinem, Dan U. Nurkhalifah, “Perancangan Pengelolaan Data Pelabuhan Perikanan Nusantara (Ppn) Pelabuhan Ratu Menggunakan *Entity Relationship Diagram (Erd),*” *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, Vol. 9, No. 1, 2023.

[17] F. T. Arumsari, J. Maulindar, dan A. I. Pradana, “Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis *Internet Of Things*,” *Infotech Journal*, Vol. 9, No. 1, 2023, Doi: 10.31949/Infotech.V9i1.5317.

[18] M. A. Maksum, “Pengertian *Xampp*, Fungsi, Dan Cara Menggunakannya,” 2022.

[19] Nurhadi dan M. Ridwan, “Sistem Informasi Inventaris Berbasis Web Menggunakan Metode Prototype,” *Jurnal Multidisiplin Madani*, Vol. 2, No. 9, 2022, Doi: 10.55927/Mudima.V2i9.1143.

[20] Uminingsih, M. N. Ichsanudin, M. Yusuf, Dan S. Suraya, “Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula,” *Storage: Jurnal Ilmiah Teknik Dan Ilmu Komputer*, Vol. 1, No. 2, 2022, Doi: 10.55123/Storage.V1i2.270.

[21] F. G. Sembodo, G. F. Fitriana, Dan N. A. Prasetyo, “Evaluasi *Usability* *Website* Shopee Menggunakan *System Usability Scale (Sus)*,” *Journal Of Applied Informatics And Computing*, Vol. 5, No. 2, 2021, Doi: 10.30871/Jaic.V5i2.3293.

*Halaman ini sengaja dikosongkan*