

Kelompok : 3.0
Nama / NIM : M. Haykal Lazuardy / 607012400068
: Salman Ridhwan Qomarudin / 607012430004
: Vemas Seftaesa Dwi Setiawan / 607012400076
Kelas : D3SI 48-05
Kode MK / Nama MK : APSI / Analisis Perancangan Sistem Informasi
No dokumen : 1.0
Persetujuan Dosen : Anranur Uwaisy Marchiningrum
Pengampu

Judul Sistem

1. Deskripsi studi kasus

Sistem Informasi Tapal Kuda adalah sebuah platform digital berbasis web yang dirancang untuk mengelola operasional kedai kopi. Sistem ini mengintegrasikan fungsionalitas untuk pelanggan (Customer) yang dapat melihat menu, melakukan reservasi meja, dan memesan secara online. Sistem ini juga mencakup antarmuka Point of Sale (POS) untuk Kasir yang melayani transaksi langsung di tempat dan mengelola reservasi harian. Selain itu, terdapat dasbor Administrasi (Admin) untuk mengelola data master seperti menu, stok produk, pengguna, serta melihat laporan penjualan dan ulasan pelanggan.

2. Identifikasi aktor

Aktor	Peran
Pelanggan	Pengguna umum yang berinteraksi dengan sistem untuk melihat menu, melakukan registrasi, login, membuat reservasi, melakukan checkout pesanan online, dan memberikan ulasan.
Kasir	Staf operasional yang menggunakan sistem di kedai. Berinteraksi dengan sistem untuk login, memproses pesanan langsung (POS), dan mengelola reservasi yang masuk (menerima/menolak).
Admin	Pengguna dengan hak akses penuh (administrator). Berinteraksi dengan sistem untuk login, mengelola data master (Menu, Stok, Pengguna), melihat dasbor, memantau rating & ulasan, dan membuat laporan (penjualan, stok).

Payment <<Sistem>>	Aktor eksternal (sistem pihak ketiga). Berinteraksi dengan sistem untuk memvalidasi dan memproses pembayaran online (Qris/E-Wallet) saat checkout.
-----------------------	--

3. Identifikasi Fungsionalitas

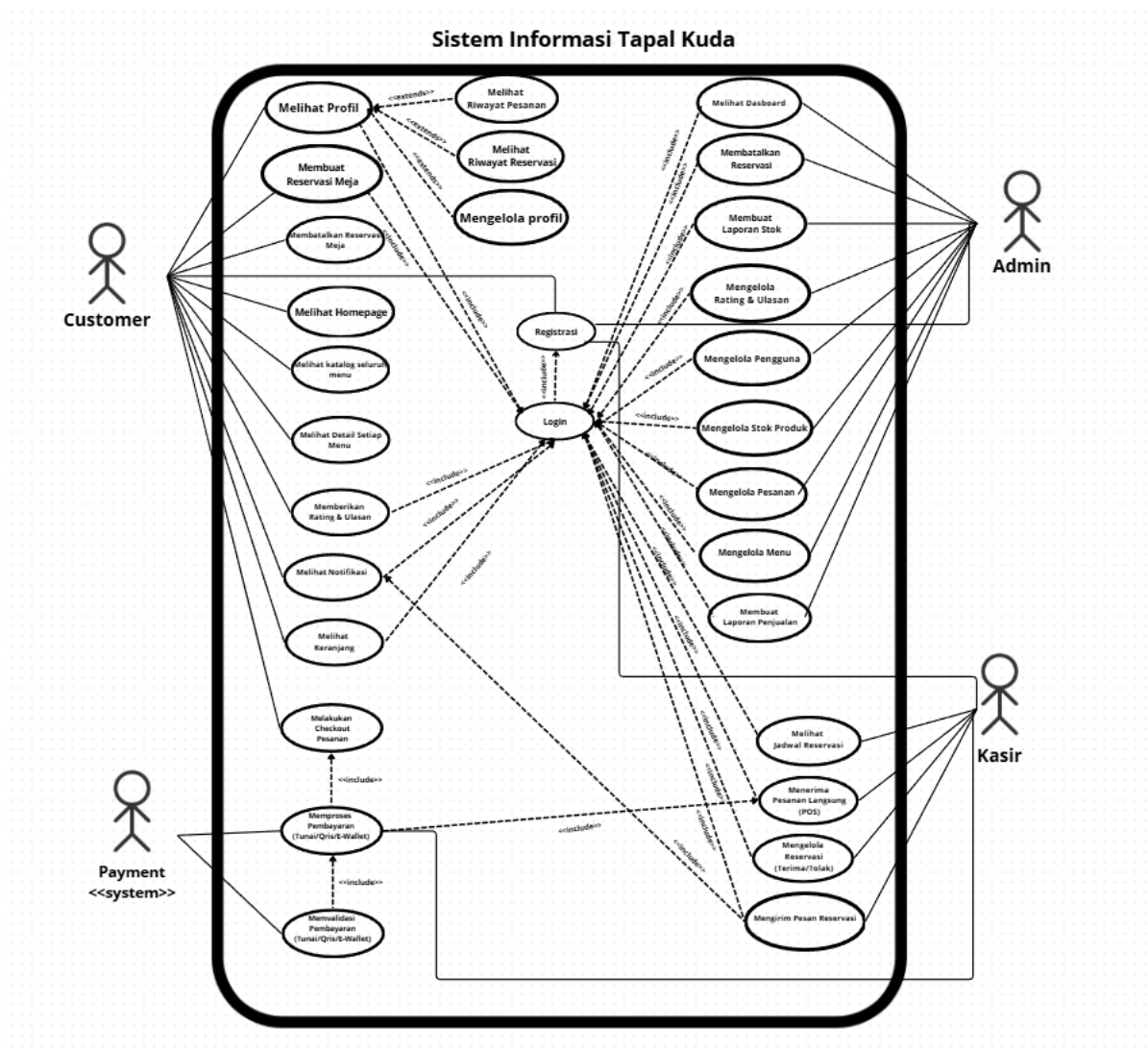
Berikut adalah daftar fungsionalitas (kandidat use case) untuk keseluruhan sistem:

- Login
- Logout
- Lupa Password
- Melakukan Registrasi
- Melihat Homepage
- Melihat Katalog Seluruh Menu
- Melihat Detail Setiap Menu
- Melihat Profil
- Mengelola Profil
- Melihat Riwayat Pesanan
- Melihat Riwayat Reservasi
- Membuat Reservasi Meja
- Membatalkan Reservasi Meja
- Melakukan Checkout Pesanan
- Memberikan Rating & Ulasan
- Melihat Notifikasi
- Melihat Keranjang
- Menerima Pesanan Langsung (POS)
- Mengelola Reservasi (Terima/Tolak)
- Melihat Jadwal Reservasi
- Mengirim Pesan Reservasi
- Melihat Dasbor
- Mengelola Pengguna
- Mengelola Stok Produk
- Mengelola Pesanan
- Mengelola Menu
- Membuat Laporan Penjualan

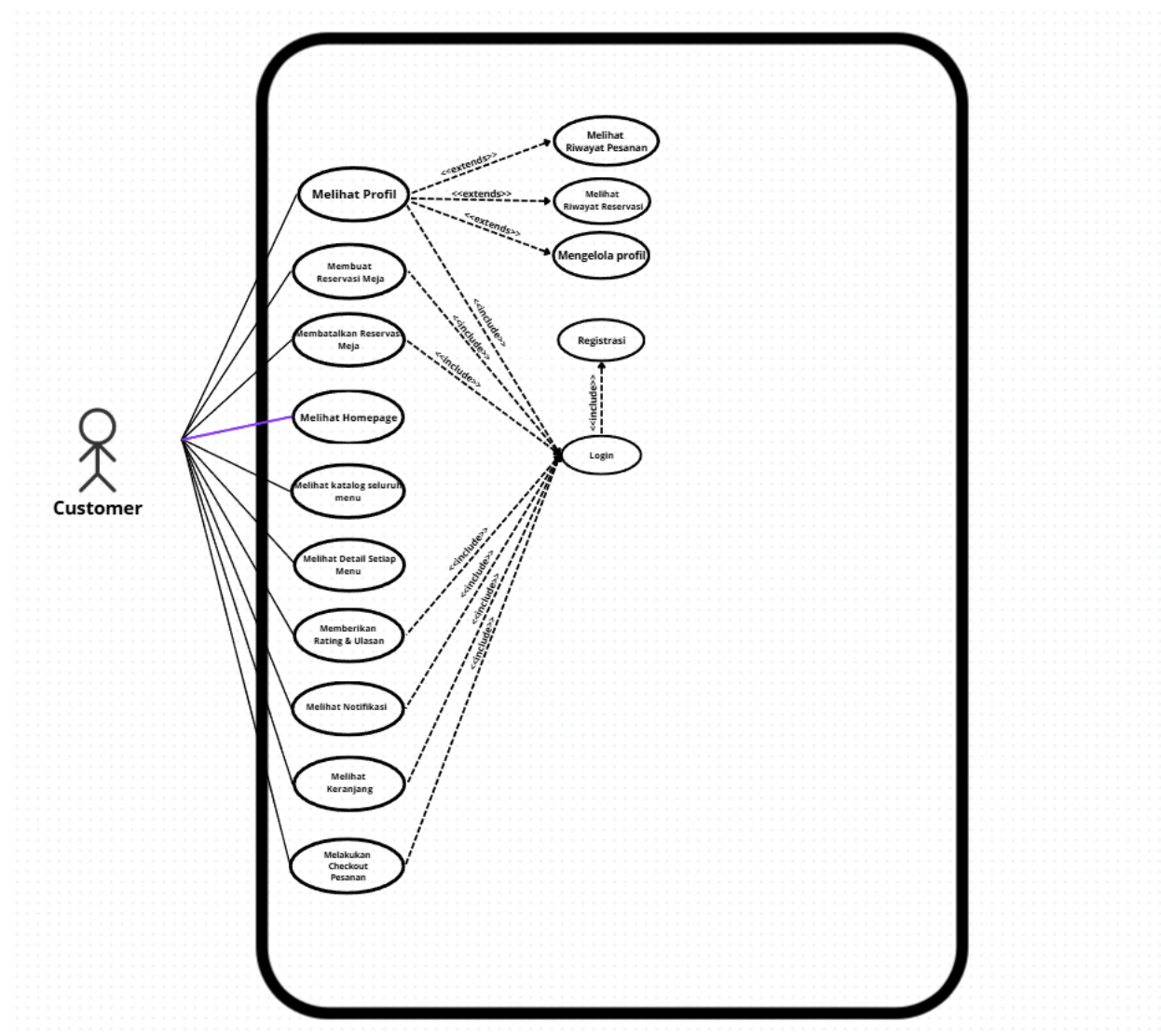
- Membuat Laporan Stok
- Mengelola Rating & Ulasan
- Memproses Pembayaran (Online/Qris/E-Wallet)
- Memvalidasi Pembayaran (Online/Qris/E-Wallet)
- (Internal) Update Stok Otomatis
- (Internal) Mengirim Notifikasi

4. Diagram Use Case

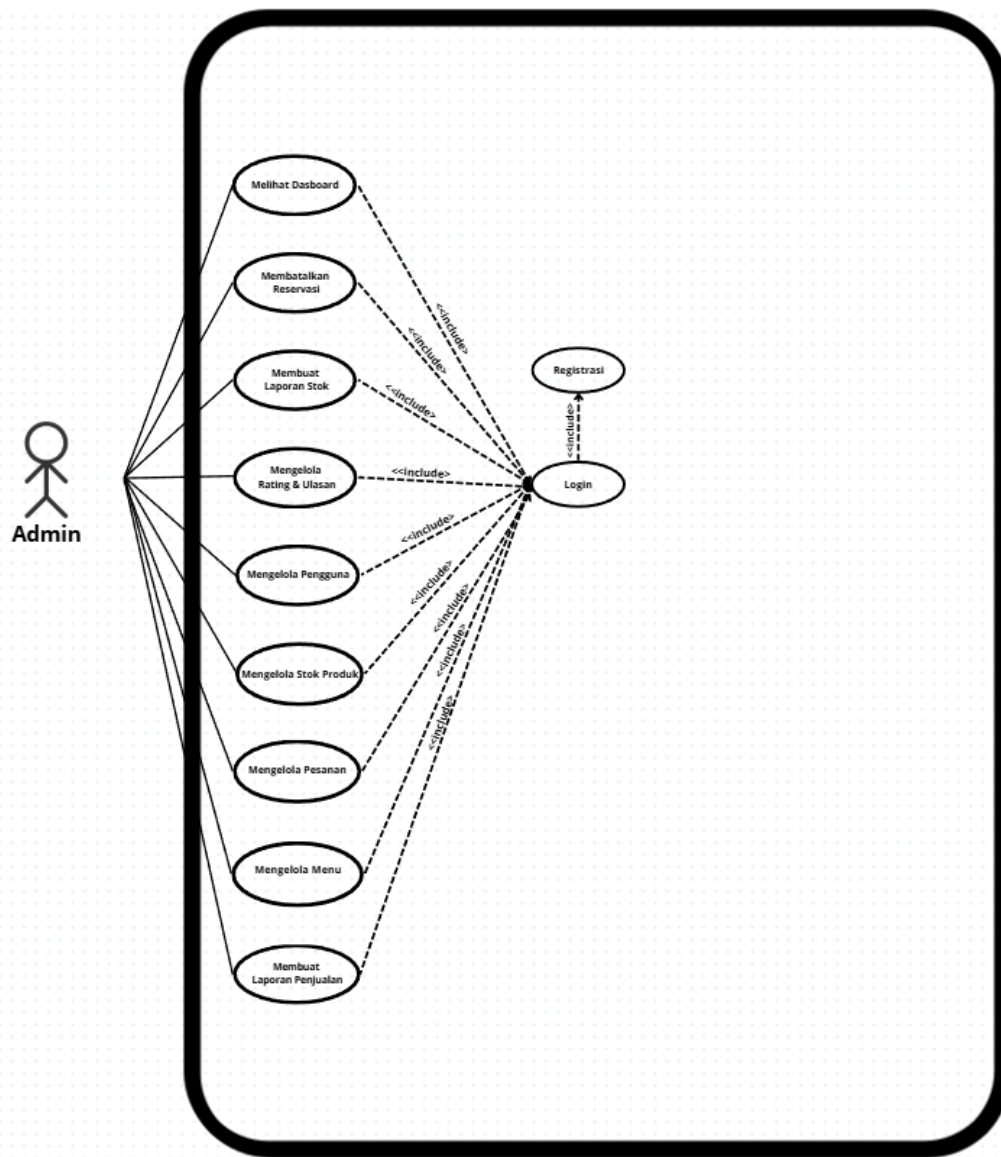
4.1. Diagram use case seluruh sistem



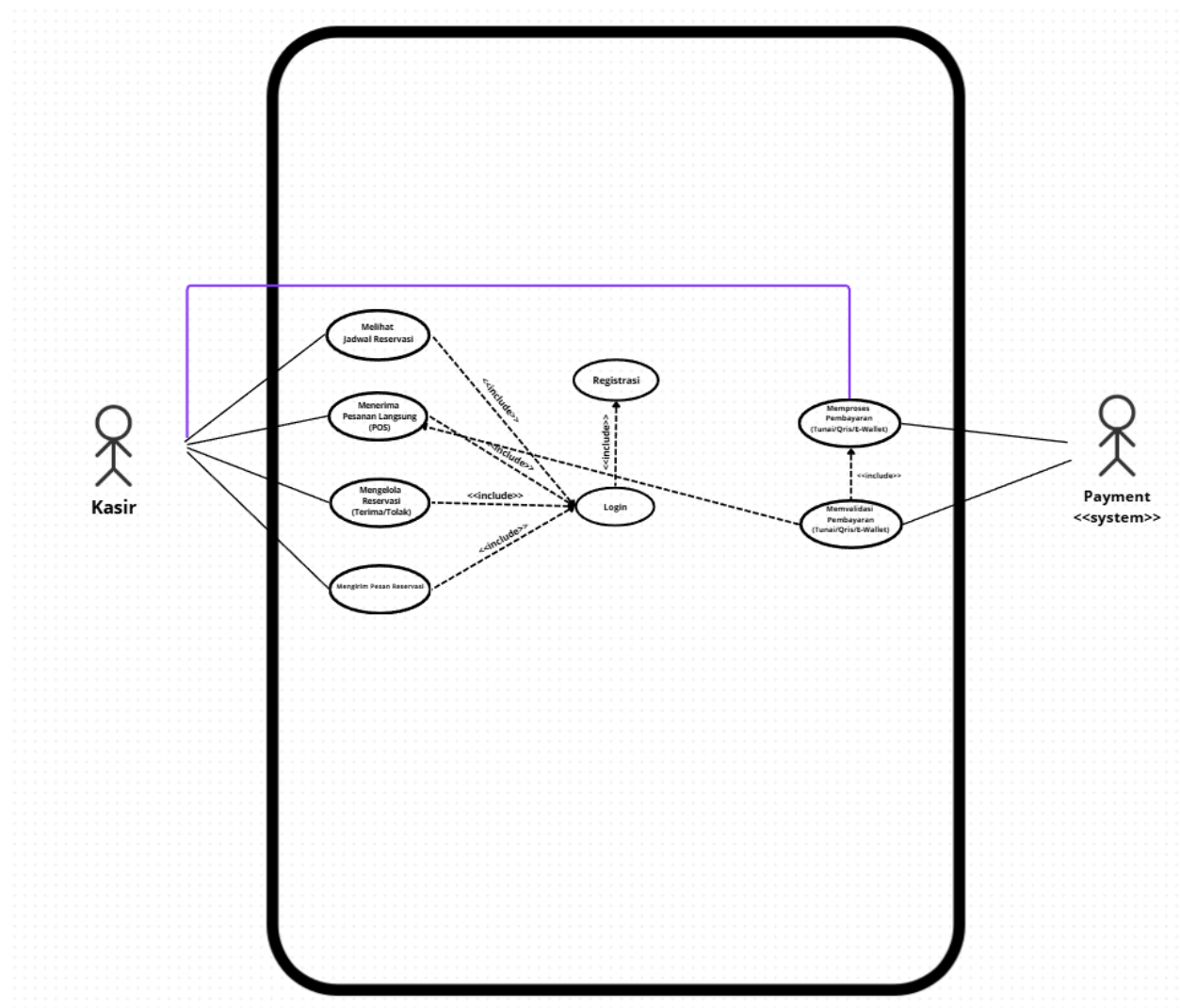
4.2. Diagram use case customer



4.3. Diagram use case admin



4.4. Diagram use case kasir



5. Skenario Use Case

1). Login

- Nama use case : Login
- Aktor : Customer, Kasir, Admin
- Deskripsi : Proses ini digunakan oleh pengguna terdaftar untuk masuk ke dalam sistem menggunakan kredensial mereka.
- Pre-Condition : Pengguna sudah memiliki akun yang terdaftar di sistem dan tidak sedang dalam keadaan login. Pengguna berada di halaman login.

Post Condition : Pengguna berhasil diautentikasi, sesi (session) untuk pengguna dibuat, dan sistem mengarahkan pengguna ke halaman yang sesuai dengan perannya.

Include : Melihat Dasbor, Mengelola Pengguna, Melihat Profil, Mengelola Pesanan

Skenario

Aktor	Sistem
1.Memasukkan email dan password	
2.Menekan Tombol “Login”.	
	3. Sistem memverifikasi kecocokan email dan password di database
	4.Sistem mengidentifikasi peran (role) pengguna
	5. Sistem membuat sesi login untuk pengguna
	6. Sistem mengarahkan pengguna ke halaman utama sesuai perannya

Skenario alternatif

Aktor	Sistem
	3a. Jika email dan password yang dimasukkan salah, sistem akan menampilkan pesan error, “Kombinasi email dan password salah.”
	3b. Sistem tetap berada di halaman login.

2). Melakukan Registrasi

Nama use case : Registrasi (atau melakukan registrasi)

Aktor : Pelanggan

Deskripsi : Proses ini memungkinkan pengguna baru (pelanggan) untuk membuat akun di sistem “Tapal Kuda”.

Pre-Condition : Pengguna sudah memiliki akun yang terdaftar di sistem dan tidak sedang dalam keadaan login. Pengguna berada di halaman login.

Post Condition : Akun baru untuk Customer berhasil dibuat dan datanya tersimpan di database.

Include : Login

Aktor	Sistem
1. Mengisi formulir registrasi (misal: Nama, Email, Password).	
2. Menekan tombol "Daftar".	
	3. Sistem memvalidasi data yang dimasukkan
	4. Sistem mengidentifikasi peran (role) pengguna
	5. Sistem membuat sesi login untuk pengguna
	6. Sistem mengarahkan pengguna ke halaman utama sesuai perannya

3). Melakukan Checkout Pesanan

Nama use case : Melakukan Checkout Pesanan
 Aktor : Pelanggan
 Deskripsi : Proses bagi Customer untuk memfinalisasi pesanan dari keranjang belanja dan melakukan pembayaran.
 Pre-Condition : Customer sudah Login dan memiliki setidaknya satu barang di "Melihat Keranjang".
 Post Condition : Pesanan berhasil dibuat, pembayaran diproses, dan notifikasi dikirim.
 Include : Memproses Pembayaran

Aktor	Sistem
1. Menekan tombol "Checkout" dari halaman "Melihat Keranjang".	
	2. Sistem menampilkan ringkasan pesanan dan total harga.
3. Menekan tombol "Bayar Sekarang".	
	4. Sistem memanggil <i>use case</i> Memproses Pembayaran (Tunai/Qris/E-Wallet)
	5. (Asumsi pembayaran berhasil) Sistem membuat catatan pesanan baru di <i>database</i> .
	6. Sistem menampilkan pesan "Pembayaran berhasil, pesanan sedang diproses."

4). Melihat Profil

Nama use case : Melihat Profil
Aktor : Customer
Deskripsi : Proses ini memungkinkan Customer untuk melihat data profil pribadi mereka, seperti nama dan email. Halaman ini juga menjadi titik awal untuk mengubah profil atau melihat riwayat.
Pre-Condition : Customer harus sudah Login
Post Condition : Sistem menampilkan halaman data profil Customer.
Include : Login
Extend : Melihat Profil

Skenario

Aktor	Sistem
1. Menekan menu "Profil Saya" (atau nama pengguna).	
	2. Sistem mengambil data profil (Nama, Email, No. HP) milik Customer yang sedang <i>login</i> dari <i>database</i> .
	3. Sistem menampilkan halaman Profil yang berisi data tersebut, serta menyediakan tautan/tombol untuk "Ubah Profil", "Melihat Riwayat Pesanan", dan "Melihat Riwayat Reservasi".

Skenario Alternatif No.3

Aktor	Sistem
3a. Customer menekan tombol "Ubah Profil".	
	3b. Sistem memulai <i>use case</i> Mengelola profil dan menampilkan formulir untuk mengubah data.

Skenario Alternatif No. 3

Aktor	Sistem
3c. Customer menekan tombol "Melihat Riwayat Pesanan".	
	3d. Sistem memulai <i>use case</i> Melihat Riwayat Pesanan dan menampilkan daftar riwayat pesanan Customer.

Skenario Alternatif No. 3

Aktor	Sistem
3e. Customer menekan tombol "Melihat Riwayat Reservasi".	
	3f. Sistem memulai <i>use case</i> Melihat Riwayat Reservasi dan menampilkan daftar riwayat reservasi Customer.

5). Mengelola profil

Nama use case : Mengelola profil
 Aktor : Customer
 Deskripsi : Proses ini memungkinkan Customer untuk mengubah data pribadi mereka, seperti Nama, Nomor HP, atau *Password*.
 Pre-Condition : Customer harus sudah Login dan telah menekan tombol "Ubah Profil" dari halaman Melihat Profil.
 Post Condition : Data profil Customer yang baru berhasil divalidasi dan disimpan di *database*. Sistem menampilkan pesan sukses.
 Include : Login

Skenario

Aktor	Sistem
	1. Sistem menampilkan formulir Ubah Profil dengan data Customer saat ini (Nama, No. HP, dll.).
2. Customer mengubah data di kolom yang diinginkan (misal: mengubah Nomor HP).	
3. Customer menekan tombol "Simpan Perubahan".	
	4. Sistem memvalidasi data yang baru dimasukkan (misal: format No. HP harus benar).
	5. Sistem memperbarui data Customer di <i>database</i> dengan informasi yang baru.
	6. Sistem menampilkan pesan "Profil berhasil diperbarui."
	7. Sistem mengarahkan kembali ke halaman Melihat Profil.

Skenario Alternatif No.4

Aktor	Sistem
	4a. Jika data yang dimasukkan tidak valid (misal: No. HP berisi huruf), sistem akan menampilkan pesan error, "Format Nomor HP tidak valid."
	4b. Sistem tetap berada di halaman formulir Ubah Profil dan menyorot kolom yang salah.

6). Melihat Riwayat Pesanan

Nama use case : Melihat Riwayat Pesanan
Aktor : Customer
Deskripsi : Proses ini memungkinkan Customer untuk melihat daftar semua transaksi/pesanan *online* yang pernah dilakukannya.
Pre-Condition : Customer harus sudah Login dan sedang berada di halaman Melihat Profil.
Post Condition : Sistem menampilkan halaman berisi daftar riwayat pesanan Customer.
Extend : Melihat Profil

Skenario

Aktor	Sistem
1. Menekan tombol/tautan "Riwayat Pesanan" (dari halaman Melihat Profil).	
	2. Sistem mengambil data semua pesanan <i>online</i> yang terkait dengan ID Customer tersebut dari <i>database</i> .
	3. Sistem menampilkan daftar riwayat pesanan (misal: ID Pesanan, Tanggal, Total Harga, Status Pesanan).

Skenario Alternatif No.2

Aktor	Sistem
	2a. Jika Customer belum pernah melakukan pesanan, sistem akan menampilkan pesan: "Anda belum memiliki riwayat pesanan."

7). Melihat Riwayat Reservasi

Nama use case : Melihat Riwayat Reservasi
Aktor : Customer
Deskripsi : Proses ini memungkinkan Customer untuk melihat daftar semua reservasi meja yang pernah dilakukannya.
Pre-Condition : Customer harus sudah Login dan (sesuai Skenario 4) sedang berada di halaman Melihat Profil.
Post Condition : Sistem menampilkan halaman berisi daftar riwayat reservasi Customer.
Extend : Melihat Profil

Skenario

Aktor	Sistem
1. Menekan tombol/tautan "Riwayat Reservasi" (dari halaman Melihat Profil).	
	2. Sistem mengambil data semua reservasi meja yang terkait dengan ID Customer tersebut dari <i>database</i> .
	3. Sistem menampilkan daftar riwayat reservasi (misal: Tanggal Reservasi, Jam, Jumlah Orang, Status).

Skenario Alternatif No.2

Aktor	Sistem
	2a. Jika Customer belum pernah melakukan reservasi, sistem akan menampilkan pesan: "Anda belum memiliki riwayat reservasi."

8). Membuat Reservasi Meja

Nama use case : Membuat Reservasi Meja
Aktor : Customer
Deskripsi : Proses ini memungkinkan Customer yang sudah *login* untuk memilih tanggal, waktu, dan jumlah orang untuk melakukan reservasi meja di kedai kopi.
Pre-Condition : Customer harus sudah Login

Post Condition : Data reservasi baru berhasil disimpan ke dalam sistem dengan status "Menunggu Konfirmasi". Sistem memicu notifikasi untuk kasir.

Include : Login

Skenario

Aktor	Sistem
1. Memilih tanggal reservasi.	
2. Memilih jam reservasi.	
3. Memasukkan jumlah orang.	
4. Menekan tombol "Buat Reservasi".	
	5. Sistem memvalidasi data (misal: tanggal tidak boleh di masa lalu, jam operasional).
	6. Sistem menyimpan data reservasi baru ke <i>database</i> dengan status "Menunggu Konfirmasi".
	7. Sistem memanggil fungsionalitas notifikasi untuk memberitahu Kasir.
	8. Sistem menampilkan pesan "Reservasi Anda telah berhasil dibuat dan sedang menunggu konfirmasi."

Skenario Alternatif No.5

Aktor	Sistem
	5a. Jika data yang dimasukkan tidak valid (misal: memilih jam yang sudah lewat), sistem akan menampilkan pesan error, "Tanggal atau jam tidak valid."
	5b. Sistem tetap berada di halaman formulir reservasi.

9). Membatalkan Reservasi Meja

Nama use case : Membatalkan Reservasi Meja

Aktor : Customer

Deskripsi : Proses ini memungkinkan Customer untuk membatalkan reservasi meja yang sebelumnya telah mereka buat.

Pre-Condition : Customer harus sudah Login dan (biasanya) berada di halaman "Riwayat Reservasi"

Post Condition : Status reservasi di *database* diubah menjadi "Dibatalkan". Sistem memicu notifikasi untuk kasir.

Include : Login, dan mengirim notifikasi

Skenario

Aktor	Sistem
1. Memilih salah satu reservasi yang berstatus "Diterima" atau "Menunggu Konfirmasi" dari daftar riwayat.	
2. Menekan tombol "Batalkan Reservasi".	
	3. Sistem menampilkan <i>pop-up</i> konfirmasi: "Apakah Anda yakin ingin membatalkan reservasi ini?"
4. Menekan tombol "Ya, Batalkan".	
	5. Sistem memeriksa aturan bisnis (misal: "Apakah pembatalan masih diizinkan?").
	6. Sistem mengubah status reservasi di <i>database</i> menjadi "Dibatalkan".
	7. Sistem memanggil fungsionalitas notifikasi (merujuk ke <i>use case</i> Mengirim Pesan Reservasi untuk memberitahu Kasir.
	8. Sistem menampilkan pesan "Reservasi Anda telah berhasil dibatalkan."

Skenario Alternatif No.5

Aktor	Sistem
	5a. Sistem mendeteksi bahwa reservasi tidak dapat dibatalkan (misal: kurang dari 1 jam sebelum waktu reservasi).
	5b. Sistem menampilkan pesan error: "Maaf, pembatalan tidak dapat dilakukan karena sudah terlalu dekat dengan waktu reservasi."

10). Melihat Homepage

Nama use case : Melihat Homepage

Aktor : Customer

Deskripsi : Proses ini memungkinkan Customer atau pengunjung untuk melihat halaman utama (landing page) dari sistem Tapal Kuda, yang biasanya berisi informasi umum, promo, atau menu unggulan.

Pre-Condition : Customer atau pengunjung membuka alamat web (URL) aplikasi.

Post Condition : Sistem menampilkan Halaman Utama.

Include : Login

Skenario

Aktor	Sistem
1. Membuka alamat web (URL) Sistem Informasi Tapal Kuda di browser.	
	2. Sistem mengambil data yang relevan untuk Halaman Utama (misal: data promo, menu unggulan).
	3. Sistem menampilkan Halaman Utama kepada pengguna.

11). Melihat katalog seluruh menu

Nama use case : Melihat katalog seluruh menu

Aktor : Customer

Deskripsi : Proses ini memungkinkan Customer untuk melihat semua menu yang dijual (Kopi, Makanan, Cemilan) dalam satu halaman. Halaman ini juga menyediakan filter berdasarkan kategori.

Pre-Condition : Customer harus sudah Login

Post Condition : Sistem menampilkan halaman katalog berisi semua menu, dan Customer dapat memilih menu untuk dilihat detailnya.

Include : Login

Skenario

Aktor	Sistem
1. Menekan tombol/menu "Katalog Menu" (setelah <i>login</i>).	
	2. Sistem mengambil semua data menu (Kopi, Makanan, Non-Kopi, Cemilan) dari <i>database</i> .
3. Sistem menampilkan halaman katalog yang berisi semua menu tersebut, beserta filter kategori di bagian atas.	

Skenario Alternatif No.3

Aktor	Sistem
4. Customer menekan salah satu filter (misal: "Kopi").	
	5. Sistem menyembunyikan menu dari kategori lain dan hanya menampilkan menu kategori "Kopi".

Skenario alternatif No.3

Aktor	Sistem
6. Customer mengklik gambar atau nama salah satu menu.	
	7. Sistem memulai <i>use case</i> Melihat Detail Setiap Menu.

12). Melihat Detail Setiap Menu

Nama use case : Melihat Detail Setiap Menu
 Aktor : Customer
 Deskripsi : Proses ini memungkinkan Customer untuk melihat informasi rinci dari satu menu spesifik, seperti deskripsi lengkap, harga, dan foto yang lebih besar.
 Pre-Condition : Customer harus sudah Login dan sedang berada di halaman Melihat katalog seluruh menu.
 Post Condition : Sistem menampilkan halaman detail untuk menu yang dipilih. Customer dapat melanjutkan untuk memesan (menambah ke keranjang) atau memberi ulasan
 Include : Login

Skenario

Aktor	Sistem
1. (Dari Skenario 11) Customer mengklik gambar atau nama salah satu menu.	
	2. Sistem mengambil data rinci (deskripsi, harga, foto, ulasan yang ada) untuk menu yang dipilih dari <i>database</i> .
	3. Sistem menampilkan halaman Detail Menu yang berisi semua informasi tersebut.

Skenario Alternatif No.3

Aktor	Sistem
3a. Customer menekan tombol "Tambah ke Keranjang".	
	3b.. Sistem memulai <i>use case</i> Melihat Keranjang [cite: Screenshot 2025-11-15 194218.png] (atau fungsionalitas keranjang).

Skenario Alternatif No.3

Aktor	Sistem
	3c. Sistem mendeteksi stok untuk menu ini adalah 0 (nol)
	3d. Sistem menampilkan halaman Detail Menu, namun dengan status "Stok Habis" dan tombol "Tambah ke Keranjang" dinonaktifkan.

13). Memberikan Rating & Ulasan

Nama use case : Memberikan Rating & Ulasan
Aktor : Customer
Deskripsi : Proses ini memungkinkan Customer untuk memberikan peringkat (bintang) dan ulasan tertulis untuk menu yang spesifik.
Pre-Condition : Customer harus sudah Login dan (sesuai Skenario 12) sedang berada di halaman Melihat Detail Setiap Menu.
Post Condition : Data rating dan ulasan baru dari Customer berhasil disimpan di *database* dan akan ditampilkan di halaman detail menu tersebut.
Include : Login

Skenario

Aktor	Sistem
1. (Dari Skenario 12) Customer menekan tombol "Beri Ulasan".	
	2. Sistem menampilkan formulir/area untuk memberikan rating (misal: 1-5 bintang) dan kolom teks untuk ulasan
3. Customer memilih jumlah bintang (misal: 5 bintang).	

4. Customer (opsional) menuliskan ulasan di kolom teks.	
5. Customer menekan tombol "Kirim Ulasan".	
	6. Sistem memvalidasi input (minimal harus memilih bintang).
	7. Sistem menyimpan data ulasan dan rating ke <i>database</i> , mengaitkannya dengan ID Customer dan ID Menu.
	8. Sistem menghitung ulang rata-rata rating untuk menu tersebut.
	9. Sistem menampilkan pesan "Ulasan Anda berhasil dikirim." dan me- <i>refresh</i> halaman detail (menampilkan ulasan yang baru ditambahkan).

Skenario Alternatif No.6

Aktor	Sistem
	6a. Jika Customer menekan "Kirim Ulasan" tetapi belum memilih bintang, sistem akan menampilkan pesan error: "Harap pilih rating (bintang) terlebih dahulu."
	6b. Sistem tetap berada di formulir ulasan.

14). Melihat Notifikasi

Nama use case : Melihat Notifikasi
 Aktor : Customer
 Deskripsi : Proses ini memungkinkan Customer untuk melihat daftar notifikasi yang dikirim oleh sistem (misalnya status reservasi diterima/ditolak, status pesanan selesai, dll).
 Pre-Condition : Customer harus sudah Login. (Sesuai diagram final, *use case* ini di-*<<include>>* oleh Login).
 Post Condition : Customer melihat daftar notifikasi. Notifikasi yang baru dilihat ditandai sebagai "sudah dibaca".
 Include : Login

Skenario

Aktor	Sistem
1. Menekan ikon "Lonceng Notifikasi".	

	2. Sistem mengambil semua data notifikasi (yang belum dibaca dan sudah dibaca) yang terkait dengan ID Customer dari <i>database</i> .
	3. Sistem menampilkan daftar notifikasi dalam <i>dropdown</i> atau halaman baru, membedakan notifikasi yang belum dibaca.
	4. Sistem secara otomatis menandai notifikasi yang baru ditampilkan sebagai "sudah dibaca" di <i>database</i> .

Skenario Alternatif No.2

Aktor	Sistem
	2a. Jika Customer tidak memiliki notifikasi sama sekali, sistem akan menampilkan pesan: "Tidak ada notifikasi baru."

15). Melihat Keranjang

Nama use case : Melihat Keranjang
 Aktor : Customer
 Deskripsi : Proses bagi Customer untuk meninjau item yang telah ditambahkan ke keranjang belanja sebelum melanjutkan ke *checkout*.
 Pre-Condition : Customer harus sudah Login. (Sesuai diagram final, *use case* ini di-
 <<include>> oleh Login).
 Post Condition : Sistem menampilkan daftar item di keranjang. Customer dapat melanjutkan ke *use case* Melakukan Checkout Pesanan.
 Include : Login

Skenario

Aktor	Sistem
1. Menekan ikon/tombol "Keranjang Belanja".	
	2. Sistem mengambil data item (nama, jumlah, harga) yang ada di keranjang Customer saat ini.
	3. Sistem menampilkan daftar item, total harga, dan tombol "Checkout".

Skenario Alternatif No.2

Aktor	Sistem
	2a. Jika tidak ada item yang ditemukan di keranjang, sistem akan menampilkan pesan: "Keranjang Anda kosong."
	2b. Sistem menonaktifkan tombol "Checkout".

Skenario Alternatif

Aktor	Sistem
4. Customer mengubah jumlah item (misal: dari 1 menjadi 2).	
	5. Sistem secara otomatis menghitung ulang total harga.

Skenario Alternatif

Aktor	Sistem
6. Customer menekan tombol "Hapus" pada salah satu item.	
	7. Sistem menghapus item tersebut dari keranjang.
	8. Sistem menghitung ulang total harga.

16). Menerima Pesanan Langsung (POS)

- Nama use case : Menerima Pesanan Langsung (POS)
- Aktor : Kasir
- Deskripsi : Proses bagi Kasir untuk melayani dan mencatat pesanan dari pelanggan yang datang langsung (*walk-in*) di kedai.
- Pre-Condition : Kasir harus sudah Login dan berada di antarmuka (halaman) Point of Sale.
- Post Condition : Pesanan berhasil dicatat, pembayaran diterima, dan stok produk (jika relevan) diperbarui. Struk tercetak.
- Include : Memproses Pembayaran (Tunai/Kartu), Update Stok Otomatis

Skenario

Aktor	Sistem
1. Memilih menu dan jumlah yang dipesan pelanggan di antarmuka POS.	

	2. Sistem menambahkan item ke keranjang POS dan menghitung total harga.
3. Mengonfirmasi pesanan ke pelanggan dan menekan tombol "Bayar".	
	4. Sistem menampilkan pilihan metode pembayaran (Tunai, Kartu/Debit, Qris/E-Wallet).
5. Memilih metode pembayaran (misal: "Tunai") dan memasukkan jumlah uang yang diterima dari pelanggan.	
	6. Sistem menghitung jumlah kembalian (jika ada).
	7. Sistem memanggil fungsionalitas Memproses Pembayaran (Tunai/Kartu) untuk mencatat transaksi keuangan.
	8. Sistem memanggil fungsionalitas Update Stok Otomatis untuk mengurangi stok item yang terjual.
	9. Sistem mengirim perintah cetak struk ke printer.
	10. Sistem menampilkan pesan "Transaksi Berhasil" dan siap untuk pesanan berikutnya.

Skenario Alternatif No.4

Aktor	Sistem
5a. Memilih metode "Kartu" atau "Qris".	
	5b. Sistem mencoba memproses pembayaran (misal: via EDC atau menampilkan QR code).
	5c. Sistem menerima respon "Pembayaran Gagal" (misal: kartu ditolak, <i>timeout</i>).
	5d. Sistem menampilkan pesan error: "Pembayaran Gagal. Silakan coba lagi atau ganti metode pembayaran."
6a. Kembali ke langkah 4 (memilih metode pembayaran).	

17). Mengelola Reservasi (Terima/Tolak)

Nama use case : Mengelola Reservasi (Terima/Tolak)
Aktor : Kasir
Deskripsi : Proses ini memungkinkan Kasir untuk meninjau permintaan reservasi meja yang masuk dari Customer dan memutuskan untuk Menerima atau Menolaknya.
Pre-Condition : Kasir harus sudah Login dan ada reservasi baru yang masuk dengan status "Menunggu Konfirmasi".
Post Condition : Status reservasi di *database* diubah menjadi "Diterima" atau "Ditolak". Sistem memicu *use case* Mengirim Pesan Reservasi untuk memberi tahu Customer.
Include : Mengirim Pesan Reservasi

Skenario

Aktor	Sistem
1. Memilih notifikasi reservasi baru atau membuka halaman "Manajemen Reservasi".	
	2. Sistem menampilkan daftar reservasi yang berstatus "Menunggu Konfirmasi".
3. Memilih salah satu reservasi untuk melihat detailnya (Nama, Tanggal, Jam, Jumlah Orang).	
4. Memeriksa ketersediaan meja (secara manual atau dibantu sistem).	
5. Menekan tombol "Terima".	
	6. Sistem mengubah status reservasi di <i>database</i> menjadi "Diterima".
	7. Sistem memanggil <i>use case</i> Mengirim Pesan Reservasi (untuk mengirim notifikasi "Reservasi Diterima" ke Customer).
	8. Sistem menampilkan pesan "Reservasi berhasil diterima."

Skenario Alternatif No.5

Aktor	Sistem
5a. Menekan tombol "Tolak" (misalnya karena meja penuh).	
	5b. Sistem mengubah status reservasi di <i>database</i> menjadi "Ditolak".

	5c. Sistem memanggil <i>use case</i> Mengirim Pesan Reservasi [cite: Screenshot 2025-11-15 194218.png] (untuk mengirim notifikasi "Reservasi Ditolak" ke Customer).
	5d. Sistem menampilkan pesan "Reservasi telah ditolak."

18). Melihat Jadwal Reservasi

Nama use case : Melihat Jadwal Reservasi
 Aktor : Kasir
 Deskripsi : Proses ini memungkinkan Kasir untuk melihat daftar atau kalender semua reservasi yang sudah dikonfirmasi (diterima) untuk hari ini atau hari-hari mendatang, agar dapat mempersiapkan meja.
 Pre-Condition : Kasir harus sudah Login. (Sesuai diagram , *use case* ini di-
 <<include>> oleh Login).
 Post Condition : Sistem menampilkan jadwal reservasi yang diminta.
 Include : Login

Skenario

Aktor	Sistem
1. Memilih menu/halaman "Jadwal Reservasi".	
2. Memilih tanggal yang ingin dilihat (sistem otomatis menampilkan hari ini sebagai <i>default</i>).	
	3. Sistem mengambil data semua reservasi yang berstatus "Diterima" dari <i>database</i> untuk tanggal yang dipilih.
	4. Sistem menampilkan jadwal dalam bentuk daftar atau kalender, diurutkan berdasarkan jam, berisi nama pelanggan dan jumlah orang.

Skenario Alternatif No.3

Aktor	Sistem
	3a. Jika tidak ada reservasi yang ditemukan untuk tanggal tersebut.
	3b. Sistem menampilkan pesan: "Tidak ada reservasi untuk tanggal ini."

19) Melihat Dasbor

Nama use case : Melihat Dasbor
Aktor : Admin
Deskripsi : Proses ini menampilkan halaman ringkasan (dasbor) yang berisi data visual operasional kedai kopi, seperti total penjualan, jumlah reservasi, dan ulasan terbaru.
Pre-Condition : Admin telah berhasil Login. (Sesuai diagram final, *use case* ini adalah *use case* pertama yang di-<<include>> oleh Login untuk Admin).
Post Condition : Halaman Dasbor dengan ringkasan data ditampilkan kepada Admin.
Include : Login

Skenario

Aktor	Sistem
1. (Otomatis setelah Admin berhasil Login).	
	2. Sistem mengumpulkan data agregat (misal: total penjualan hari ini, jumlah reservasi baru, ulasan terakhir, stok yang menipis).
	3. Sistem menampilkan Halaman Dasbor yang berisi semua ringkasan data tersebut dalam bentuk <i>widget</i> , angka, atau grafik.

Skenario Alternatif

Aktor	Sistem
4. Admin mengklik salah satu <i>widget</i> (misal: "Lihat Laporan Penjualan").	
	5. Sistem mengarahkan Admin ke <i>use case</i> terkait (misal: Membuat Laporan Penjualan).

20). Membatalkan Reservasi (Admin)

Nama use case : Membatalkan Reservasi
Aktor : Admin
Deskripsi : Proses ini memungkinkan Admin untuk membatalkan reservasi yang sudah ada (baik yang "Menunggu Konfirmasi" atau "Diterima"), mungkin karena *overbooking* atau alasan operasional.

Pre-Condition : Admin harus sudah Login. (Sesuai diagram final, *use case* ini di-
<<include>> oleh Login).

Post Condition : Status reservasi di *database* diubah menjadi "Dibatalkan". Sistem
memicu notifikasi untuk Customer.

Include : Login, Mengirim Notifikasi

Skenario

Aktor	Sistem
1. Membuka halaman "Manajemen Reservasi" (mungkin dari Dasbor).	
	2. Sistem menampilkan daftar semua reservasi.
3. Mencari dan memilih reservasi yang ingin dibatalkan.	
4. Menekan tombol "Batalkan Reservasi (Paksa)".	
	5. Sistem menampilkan <i>pop-up</i> konfirmasi: "Anda yakin ingin membatalkan reservasi milik Customer ini? Customer akan diberitahu."
6. Menekan tombol "Ya, Batalkan".	
	7. Sistem mengubah status reservasi di <i>database</i> menjadi "Dibatalkan".
	8. Sistem memanggil <i>use case</i> Mengirim Pesan Reservasi (untuk mengirim notifikasi "Reservasi Anda dibatalkan oleh Admin" ke Customer).
	9. Sistem menampilkan pesan "Reservasi berhasil dibatalkan."

Skenario Alternatif No.6

Aktor	Sistem
6a. Menekan tombol "Tidak" pada <i>pop-up</i> konfirmasi.	
	6b. Sistem menutup <i>pop-up</i> konfirmasi dan kembali ke halaman manajemen reservasi.

21). Membuat Laporan Stok

Nama use case : Membuat Laporan Stok

Aktor : Admin

Deskripsi : Proses ini memungkinkan Admin untuk men-*generate* dan mengunduh laporan yang berisi data sisa stok bahan baku atau produk yang ada di *database*.

Pre-Condition : Admin harus sudah Login. (Sesuai diagram final, *use case* ini di-*<<include>>* oleh Login).

Post Condition : Sistem berhasil men-*generate* file laporan (misalnya PDF atau Excel) yang berisi data stok, dan file tersebut terunduh oleh Admin.

Include : Login

Skenario

Aktor	Sistem
1. Memilih menu/halaman "Laporan".	
2. Memilih jenis laporan: "Laporan Stok".	
3. Memilih format file (misal: "PDF").	
4. Menekan tombol "Generate Laporan Stok".	
	5. Sistem mengumpulkan data stok terbaru (nama produk, sisa stok, satuan) dari <i>database</i> .
	6. Sistem membuat file laporan stok dalam format PDF.
	7. Sistem secara otomatis mengunduh (<i>download</i>) file laporan tersebut ke komputer Admin.

Skenario Alternatif No.5

Aktor	Sistem
	5a. Jika tidak ada data stok yang terdaftar di sistem.
	5b. Sistem menampilkan pesan: "Data stok tidak ditemukan."

22). Mengelola Rating & Ulasan

Nama use case : Mengelola Rating & Ulasan

Aktor : Admin

- Deskripsi : Proses ini memungkinkan Admin untuk melihat, memoderasi, atau menghapus ulasan dan rating yang diberikan oleh Customer untuk setiap menu.
- Pre-Condition : Admin harus sudah Login. (Sesuai diagram final, *use case* ini di-
<<include>> oleh Login).
- Post Condition : Ulasan yang tidak pantas telah disembunyikan atau dihapus dari tampilan publik.
- Include : Login

Skenario

Aktor	Sistem
1. Membuka halaman "Rating & Ulasan" (mungkin dari Dasbor).	
	2. Sistem mengambil semua data ulasan (nama menu, nama customer, isi ulasan, rating bintang) dari <i>database</i> .
	3. Sistem menampilkan daftar semua ulasan, diurutkan dari yang terbaru.

Skenario Alternatif No.3

Aktor	Sistem
4. Admin menemukan ulasan yang tidak pantas (misal: mengandung kata kasar).	
5. Admin menekan tombol "Sembunyikan" (Hide) pada ulasan tersebut.	
	6. Sistem mengubah status ulasan di <i>database</i> menjadi "Disembunyikan".
	7. Sistem menampilkan pesan "Ulasan telah disembunyikan dari publik."

Skenario Alternatif No.3

Aktor	Sistem
4a. Admin menemukan ulasan <i>spam</i> .	
5a. Admin menekan tombol "Hapus" pada ulasan tersebut.	
	5b. Sistem menampilkan <i>pop-up</i> konfirmasi: "Anda yakin ingin menghapus ulasan ini secara permanen?"

6a. Admin menekan "Ya, Hapus".	
	6b. Sistem menghapus data ulasan tersebut dari <i>database</i> .
	6c. Sistem menampilkan pesan "Ulasan telah dihapus."

23). Mengelola Pengguna

Nama use case : Mengelola Pengguna
 Aktor : Admin
 Deskripsi : Proses bagi Admin untuk menambah, mengubah, atau menghapus akun pengguna (terutama Kasir atau Admin lain) di dalam sistem.
 Pre-Condition : Admin harus sudah Login. (Sesuai diagram final *use case* ini di-
 <<include>> oleh Login).
 Post Condition : Data pengguna (misal: akun Kasir baru) berhasil dibuat, diubah, atau dihapus dari *database*.
 Include : Login

Skenario

Aktor	Sistem
1. Membuka halaman "Manajemen Pengguna" (dari Dasbor).	
	2. Sistem menampilkan daftar semua pengguna yang terdaftar (Customer, Kasir, Admin).
3. Menekan tombol "Tambah Pengguna Baru".	
	4. Sistem menampilkan formulir untuk data pengguna baru (Nama, Email, <i>Password</i> , Peran/Role).
5. Mengisi data (misal: "Kasir 1", "kasir1@tapalkuda.com", Peran: "Kasir").	
6. Menekan tombol "Simpan".	
	7. Sistem memvalidasi data (misal: email unik).
	8. Sistem membuat akun pengguna baru di <i>database</i> dengan peran "Kasir".
	9. Sistem menampilkan pesan "Pengguna Kasir berhasil dibuat."

Skenario Alternatif No.7

Aktor	Sistem
	7a. Jika email yang dimasukkan sudah terdaftar di <i>database</i> .
	7b. Sistem menampilkan pesan error: "Email sudah digunakan."
	7c. Sistem tetap berada di halaman formulir tambah pengguna.

24). Mengelola Stok Produk

Nama use case : Mengelola Stok Produk
 Aktor : Admin
 Deskripsi : Proses bagi Admin untuk menambah, mengubah, atau melihat data master bahan baku/produk (misal: biji kopi, susu) dan jumlah stoknya.
 Pre-Condition : Admin harus sudah Login. (Sesuai diagram final *use case* ini di-
 <<include>> oleh Login).
 Post Condition : Data stok (jumlah atau detail produk) berhasil diperbarui di *database*.
 Include : Login

Skenario

Aktor	Sistem
1. Membuka halaman "Manajemen Stok Produk".	
	2. Sistem menampilkan daftar semua produk/bahan baku dan sisa stoknya saat ini.
3. Memilih produk yang ingin diubah (misal: "Biji Kopi Robusta").	
4. Menekan tombol "Update Stok".	
	5. Sistem menampilkan <i>pop-up</i> atau formulir untuk mengubah jumlah stok.
6. Memasukkan jumlah stok baru (misal: "10000" gram).	
7. Menekan tombol "Simpan".	
	8. Sistem memvalidasi input (harus angka).

	9. Sistem menyimpan jumlah stok baru ke <i>database</i> . Sistem menampilkan pesan "Stok berhasil diperbarui."
--	--

Skenario Alternatif No.8

Aktor	Sistem
1. Membuka halaman "Manajemen Stok Produk".	8a. Jika input yang dimasukkan tidak valid (misal: "seribu" atau -50).
	8b. Sistem menampilkan pesan error: "Jumlah stok harus berupa angka positif"
	8c. Sistem tetap berada di <i>pop-up</i> /formulir.

25). Mengelola Pesanan

Nama use case : Mengelola Pesanan
 Aktor : Admin
 Deskripsi : Proses bagi Admin untuk melihat dan mengubah status pesanan *online* yang masuk dari Customer (misal: dari "Diproses" menjadi "Selesai" atau "Dikirim").
 Pre-Condition : Admin harus sudah Login.
 Post Condition : Status pesanan di *database* diperbarui. Sistem memicu notifikasi ke Customer.
 Include : Mengirim Notifikasi

Skenario

Aktor	Sistem
1. Membuka halaman "Manajemen Pesanan" (dari Dasbor).	
	2. Sistem menampilkan daftar semua pesanan <i>online</i> yang masuk (misal: yang berstatus "Diproses").
3. Memilih salah satu pesanan untuk diubah statusnya.	
4. Menekan tombol "Ubah Status".	
	5. Sistem menampilkan pilihan status baru (misal: "Selesai", "Dibatalkan").
6. Memilih status baru (misal: "Selesai").	

7. Menekan tombol "Simpan Perubahan".	
	8. Sistem memperbarui status pesanan di <i>database</i> menjadi "Selesai".
	9. Sistem memanggil <i>use case</i> Mengirim Notifikasi
	10. Sistem menampilkan pesan "Status pesanan berhasil diperbarui."

Skenario Alternatif No.2

Aktor	Sistem
	2a. Jika tidak ada pesanan <i>online</i> yang masuk.
	2b. Sistem menampilkan pesan: "Tidak ada pesanan yang perlu dikelola."

26). Mengelola Menu

Nama use case : Mengelola Menu
 Aktor : Admin
 Deskripsi : Proses bagi Admin untuk menambah, mengubah, atau menghapus data menu (kopi, makanan, dll) yang akan ditampilkan di aplikasi.
 Pre-Condition : Admin harus sudah Login.
 Post Condition : Data menu di *database* berhasil diperbarui (dibuat, diubah, atau dihapus).
 Include : Login

Skenario

Aktor	Sistem
1. (Dari Dasbor) Memilih menu "Manajemen Menu".	
	2. Sistem menampilkan daftar semua menu yang ada saat ini.
3. Menekan tombol "Tambah Menu Baru".	
	4. Sistem menampilkan formulir untuk data menu baru.
5. Mengisi data (misal: Nama Menu, Harga, Kategori, Deskripsi, Foto).	

6. Menekan tombol "Simpan".	
	7. Sistem memvalidasi data yang dimasukkan (misal: harga harus angka).
	8. Sistem menyimpan data menu baru ke <i>database</i> .
	9. Sistem menampilkan pesan "Menu baru berhasil ditambahkan." dan kembali ke daftar menu.

Skenario Alternatif No.7

Aktor	Sistem
	7a. Jika data yang dimasukkan tidak valid (misal: harga kosong atau berisi huruf).
	7b. Sistem menampilkan pesan error: "Data tidak valid. Periksa kembali input Anda."
	7c. Sistem tetap berada di halaman formulir tambah menu.

Skenario Alternatif (Menghapus Menu):

Aktor	Sistem
1. (Dari Daftar Menu) Menekan tombol "Hapus" pada salah satu menu.	
	2. Sistem menampilkan <i>pop-up</i> konfirmasi: "Anda yakin ingin menghapus menu ini?"
3. Menekan "Ya, Hapus".	
	4. Sistem menghapus data menu dari <i>database</i> .
	5. Sistem menampilkan pesan "Menu berhasil dihapus."

27). Membuat Laporan Penjualan

Nama use case : Membuat Laporan Penjualan
 Aktor : Admin
 Deskripsi : Proses ini memungkinkan Admin untuk men-*generate* dan mengunduh laporan penjualan berdasarkan rentang waktu tertentu.
 Pre-Condition : Admin harus sudah Login. (Sesuai diagram final [cite: Screenshot 2025-11-15 194135.png], *use case* ini di-`<<include>>` oleh Login).

Post Condition : Sistem berhasil men-*generate* file laporan (misalnya PDF atau Excel) dan file tersebut terunduh oleh Admin.

Include : Login

Skenario

Aktor	Sistem
1. Memilih menu/halaman "Laporan".	
2. Memasukkan tanggal awal dan akhir periode laporan penjualan	
3. Menekan tombol "Generate Laporan" (atau "Tampilkan").	
	4. Sistem mengumpulkan data transaksi penjualan sesuai rentang tanggal yang dimasukkan.
	5. Sistem membuat file laporan (misal: PDF).
	6. Sistem secara otomatis mengunduh file laporan ke komputer Admin.

Skenario Alternatif No.4

Aktor	Sistem
	4a. Jika data periode yang dimasukkan tidak ada, maka akan menampilkan pesan "Laporan penjualan tidak ada" (sesuai template).
	4b. Sistem tetap berada di halaman laporan.

28). Memproses Pembayaran (Tunai/Qris/E-Wallet)

Nama use case : Memproses Pembayaran (Tunai/Qris/E-Wallet)

Aktor : Customer, Payment <<Sistem>>

Deskripsi : Proses bagi sistem untuk menangani transaksi pembayaran online yang dipilih oleh Customer.

Pre-Condition : Customer telah memulai use case Melakukan Checkout Pesanan (Skenario 3) dan memilih metode pembayaran.

Post Condition : Status pembayaran (berhasil atau gagal) diterima oleh sistem. Jika berhasil, pesanan dibuat.

Include : Memvalidasi Pembayaran (Tunai/Qris/E-Wallet)

Skenario

Aktor	Sistem
1. (Dari Skenario 3) Customer memilih metode pembayaran (misal: "Qris").	
	2. Sistem menampilkan QR code yang harus dipindai.
3. Customer memindai (scan) QR code dan menyelesaikan pembayaran di aplikasi E-Wallet-nya.	
	4. Sistem memanggil <i>use case</i> Memvalidasi Pembayaran [cite: Screenshot 2025-11-15 194218.png] (yang akan berkomunikasi dengan Aktor Payment <<Sistem>>).
	5. (Asumsi validasi sukses) Sistem menerima status "BERHASIL".
	6. Sistem membuat catatan pesanan baru di <i>database</i> .
	7. Sistem menampilkan pesan "Pembayaran berhasil, pesanan sedang diproses." (kembali ke Skenario 3).

Skenario Alternatif No.5

Aktor	Sistem
	5a. Sistem menerima status "GAGAL" atau "TIMEOUT" dari Memvalidasi Pembayaran.
	5b. Sistem menampilkan pesan error: "Pembayaran Gagal. Silakan coba lagi."
	5c. Sistem tetap berada di halaman <i>checkout</i> (kembali ke Skenario 3).

29). Memvalidasi Pembayaran (Tunai/Qris/E-Wallet)

Nama use case : Memvalidasi Pembayaran (Tunai/Qris/E-Wallet)
 Aktor : Payment <<Sistem>>
 Deskripsi : Proses backend dimana sistem Tapal Kuda berkomunikasi dengan Aktor Payment <<Sistem>> (Payment Gateway) untuk mengonfirmasi status pembayaran yang telah diinisiasi oleh Customer.

Pre-Condition : Use case Memproses Pembayaran (Skenario 28) telah dimulai. Sistem sedang menunggu callback atau status dari Payment <<Sistem>>.

Post Condition : Sistem mendapatkan status "BERHASIL" atau "GAGAL" dan meneruskannya kembali ke use case Memproses Pembayaran.

Include : Memproses pembayaran

Skenario

Aktor	Sistem
1. (Setelah Customer membayar di Skenario 28) Mengirimkan callback (respon) ke sistem Tapal Kuda dengan status "BERHASIL" dan detail transaksi.	
	2. Menerima callback dan memverifikasi detail transaksi (misal: jumlah pembayaran sudah benar).
	3. Mencatat status "BERHASIL" di database.
	4. Mengembalikan status "BERHASIL" ke use case Memproses Pembayaran.

Skenario Alternatif No.1

Aktor	Sistem
1a. Mengirimkan callback (respon) ke sistem Tapal Kuda dengan status "GAGAL" (misal: dana tidak cukup, timeout, atau dibatalkan pengguna).	
	2a. Menerima callback dan mencatat status "GAGAL".
	3a. Mengembalikan status "GAGAL" ke use case Memproses Pembayaran (Skenario 28).

30). Mengirim Pesan Reservasi

Nama use case : Mengirim Pesan Reservasi

Aktor : Kasir

Deskripsi : Proses bagi Kasir untuk mengirimkan pesan konfirmasi atau pembatalan secara manual kepada Customer terkait status reservasi mereka.

Pre-Condition : Kasir harus sudah Login. (Sesuai diagram final [cite: Screenshot 2025-11-15 194135.png], use case ini di-<<include>> oleh Login). Kasir (kemungkinan) baru saja mengubah status di use case Mengelola Reservasi.

Post Condition : Pesan/notifikasi berhasil dikirim ke Customer yang dituju.

Include : Login

Skenario

Aktor	Sistem
1. Membuka halaman "Kirim Pesan Reservasi".	
	2. Sistem menampilkan daftar reservasi yang statusnya baru diubah.
3. Memilih reservasi yang ingin dikirim pesan.	
4. Memilih template pesan (misal: "Reservasi Diterima").	
5. Menekan tombol "Kirim Pesan".	
	6. Sistem mengambil data kontak Customer (email/nomor HP).
	7. Sistem mengirimkan pesan/notifikasi ke Customer tersebut.
	8. Sistem menampilkan pesan "Pesan reservasi berhasil dikirim."

Skenario Alternatif No.2

Aktor	Sistem
	2a. Jika tidak ada reservasi yang statusnya baru diubah (atau semua sudah dikirim pesan).
	2b. Sistem menampilkan pesan: "Tidak ada pesan reservasi yang perlu dikirim."