PROPOSAL TUGAS AKHIR

**PERANCANGAN APLIKASI *POINT OF SALES* BERBASIS *WEBSITE* PADA FITUR *BACK OFFICE* DI KANTIN UNIVERSITAS TELKOM DENGAN METODE *SCRUM***

**Oleh:**

**MUHAMMAD FASMA FAREGA**

**1202194168**



**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS REKAYASA INDUSTRI**

**UNIVERSITAS TELKOM**

**2023**

LEMBAR PENGESAHAN

Tugas Akhir dengan judul:

**PERANCANGAN APLIKASI *POINT OF SALES* BERBASIS *WEBSITE* PADA FITUR *BACK OFFICE* DI KANTIN UNIVERSITAS TELKOM DENGAN METODE *SCRUM***

Telah disetujui dan disahkan untuk *desk evaluation* proposal tugas akhir

Program Studi S1 Sistem Informasi

Fakultas Rekayasa Industri

Tahun 2023

Oleh

**MUHAMMAD FASMA FAREGA**

**1202194168**

Bandung, 07 Januari 2023

Disetujui oleh,

| Pembimbing 1 | Pembimbing 2 |
| --- | --- |
|  |  |
|
| Faishal Mufied Al-Anshary, S.Kom., M.Kom. MSc | Taufik Nur Adi, S.Kom., M.T., Ph.D. |
| NIP. 14900050 | NIP. 14830024 |

LEMBAR PERNYATAAN ORISINALITAS

ABSTRAK

ABSTRACT

KATA PENGANTAR

Daftar Isi

[**BAB I Pendahuluan 1**](#_1t3h5sf)

[I.1 Latar Belakang 1](#_sz1cbdd9qfig)

[I.2 Perumusan Masalah 2](#_2s8eyo1)

[I.3 Tujuan Penelitian 2](#_17dp8vu)

[I.4 Manfaat Penelitian 3](#_26in1rg)

[I.5 Batasan Penelitian 3](#_3rdcrjn)

[**BAB II Tinjauan Pustaka 5**](#_1ksv4uv)

[II.1 Aplikasi Point of Sales 5](#_b1ficogs7xvs)

[II.2 Website Back Office 5](#_2bb2qyyqv80)

[II.3 Metode Scrum 6](#_p7nang74gmq1)

[II.4 PHP 6](#_6aaf159h50bo)

[II.5 Laravel 7](#_m9i6d75dc3fo)

[II.6 Postman 8](#_7w0nv4mth74b)

[II.7 Midtrans 9](#_s1dgjv8q8koi)

[II.8 Penelitian Terdahulu 9](#_4tir2o3x35qi)

[Tabel II.1 Penelitian Terdahulu: Ilham Fannani 10](#_gmb1837yj76s)

[Tabel II.2 Penelitian Terdahulu: Nailul Marom 10](#_hayuufwz6t0k)

[Tabel II.3 Penelitian Terdahulu: Muhamad Noufal Ajriya Siddik 11](#_jmo8b4wuxupu)

[**BAB III Metodologi Penelitian 12**](#_rp0f3x5tqxn8)

[III.1 Model Konseptual 12](#_umfqojetp16s)

[Gambar III.1 Model Konseptual 12](#_cwmk405zg7ub)

[III.2 Sistematika Penyelesaian Masalah 13](#_1ci93xb)

[Gambar III.2 Metode Scrum 13](#_ze14e2o4sj84)

[III.3 Alasan Pemilihan Metode 15](#_zevighenw8l1)

[III.4 Pengumpulan Data 16](#_3whwml4)

[III.5 Pengolahan Data 16](#_2bn6wsx)

[III.6 Metode Evaluasi 17](#_qsh70q)

[III.7 Rencana Jadwal Kegiatan 18](#_at45un60xzy3)

[Gambar III.3 Sprint 18](#_t3tm01ct020d)

[**BAB IV Analisis dan Perancangan 19**](#_4uf7vgye49i)

[IV.1 Proses Bisnis 19](#_g5d944dy2v95)

[IV.2 Product Backlog 19](#_xqvs9368kz9t)

[IV.3 Use Case Diagram 20](#_f36dtrqfxgq4)

[IV.4 Use Case Scenario 21](#_c2vufjn2s9hk)

[IV.5 Class Diagram 25](#_2yneo3pgaiac)

[IV.6 Activity Diagram 26](#_tokkqf4r0b5e)

[IV.7 Sequence Diagram 29](#_5nb0ao5122ka)

[IV.8 Entitas Relasional Diagram 32](#_mkwh5945xzea)

[**DAFTAR PUSTAKA 34**](#_fz2xre1i1zmh)

Daftar Gambar

[Gambar III.1 Model Konseptual 11](#_cwmk405zg7ub)

[Gambar III.2 Metode Scrum 12](#_ze14e2o4sj84)

[Gambar III.3 Sprint 17](#_t3tm01ct020d)

Daftar Tabel

[Tabel II.1 Penelitian Terdahulu: Ilham Fannani 9](#_gmb1837yj76s)

[Tabel II.2 Penelitian Terdahulu: Nailul Marom 10](#_hayuufwz6t0k)

[Tabel II.3 Penelitian Terdahulu: Muhamad Noufal Ajriya Siddik 10](#_jmo8b4wuxupu)

Daftar Lampiran

# Pendahuluan

## **Latar Belakang**

Perkembangan teknologi informasi dalam dunia bisnis telah mengubah banyak aspek operasional, salah satunya adalah sistem penjualan atau yang biasa dikenal dengan Point of Sales (POS). POS adalah sistem yang mencatat transaksi penjualan dan mampu mengendalikan operasional bisnis secara real-time. Namun, dalam implementasinya, perancangan aplikasi POS berbasis website back office dengan metode Scrum membutuhkan pemahaman yang baik tentang kebutuhan pengguna dan lingkungan operasional. Salah satu masalah yang mungkin muncul adalah ketidaksesuaian antara fitur aplikasi dan kebutuhan pengguna. Oleh karena itu, dalam penelitian ini, akan dilakukan analisis kebutuhan pengguna dan pengujian aplikasi secara mendalam untuk memastikan bahwa aplikasi yang dihasilkan benar-benar membantu pengguna dalam mengatasi masalah yang mereka hadapi dalam operasional bisnis mereka (Putri & Wulandari, 2020).

Masalah yang sering dihadapi oleh tenan kantin di lingkungan universitas adalah kurangnya sistem yang terorganisir dan terintegrasi untuk manajemen produk. Tanpa adanya alat atau metode yang tepat, tenan kantin dapat mengalami kesulitan dalam memantau stok, mengelola persediaan, memperbarui menu, atau merencanakan produksi makanan yang efisien. Selain itu, ketidakmampuan untuk mengidentifikasi tren permintaan pelanggan atau mengukur keberhasilan menu dapat menghambat pertumbuhan dan keberlanjutan bisnis kantin.

Kantin Asrama Putra dan Kantin Asrama Putri merupakan contoh kantin yang berada di lingkungan Universitas Telkom. Disaat pagi menjelang siang dan sore menjelang malam adalah waktu yang sangat padat dikarenakan kebanyakan mahasiswa asrama makan di jam tersebut. Permasalahan yang dirasakan oleh pihak kantin sendiri disaat pendataan pesanan yang masih manual dan saat pembukuan setiap minggu atau bulannya dikarenakan pembukuan penjualan yang masih manual dan terkadang hilang atau si penjual yang harus mengkalkulasikan secara manual yang terkadang merepotkan si penjual.

Oleh karena itu, perlu dilakukan perancangan aplikasi point of sales berbasis website back office yang dapat membantu pengelola kantin Universitas Universitas Telkom dalam mengelola transaksi secara lebih efektif dan efisien. Metode pengembangan yang digunakan adalah Scrum, yang merupakan metode pengembangan software agile yang efektif dan efisien.

Diharapkan perancangan aplikasi ini dapat memberikan manfaat dan kontribusi bagi pengelola kantin Universitas Telkom dalam meningkatkan efektivitas dan efisiensi bisnis.

## **Perumusan Masalah**

Rumusan masalah yang dapat diambil dari judul "Perancangan Aplikasi Point of Sales Berbasis Website Back Office untuk Tenant Kantin Universitas Telkom Menggunakan Metode Scrum" adalah sebagai berikut:

1. Bagaimana merancang aplikasi point of sales berbasis website yang memungkinkan admin untuk mengelola, dan mengubah informasi akun tenant ?
2. Bagaimana merancang aplikasi point of sales berbasis website yang memungkinkan tenant untuk untuk menambah, menghapus, mengedit, dan mengatur menu makanan atau minuman ?
3. Bagaimana merancang aplikasi point of sales berbasis website yang memungkinkan admin dan tenant untuk melihat laporan penjualan harian, mingguan, atau bulanan ?

## **Tujuan Penelitian**

Tujuan dari perancangan Aplikasi Point of Sales Berbasis Website pada Fitur Back Office di Kantin Universitas Telkom dengan Metode Scrum adalah:

1. Merancang aplikasi point of sales berbasis website yang memungkinkan admin untuk secara efisien mengelola dan mengubah informasi akun tenant.
2. Merancang aplikasi point of sales berbasis website yang memungkinkan tenant untuk dengan mudah menambah, menghapus, mengedit, dan mengatur menu makanan atau minuman sebagai bagian dari operasional bisnis.
3. Merancang aplikasi point of sales berbasis website yang memungkinkan admin dan tenant untuk secara praktis melihat laporan penjualan harian, mingguan, atau bulanan guna mendapatkan wawasan yang diperlukan dalam pengambilan keputusan bisnis.

## **Manfaat Penelitian**

Diharapkan, penelitian ini akan memberikan manfaat dalam tiga aspek, yaitu:

1. Bagi tenant, Penelitian ini memberikan solusi teknologi untuk pengelolaan pembukuan dan transaksi yang lebih efisien, mempercepat proses transaksi, mengurangi kesalahan pencatatan, dan meningkatkan produktivitas serta keuntungan bisnis tenant di Universitas Telkom.
2. Bagi Universitas Telkom, Penelitian ini meningkatkan efisiensi operasional kantin, mengurangi antrian dan kerumunan, meningkatkan kepuasan mahasiswa dalam mengakses layanan makanan, dan menciptakan citra positif universitas sebagai lembaga yang responsif terhadap kebutuhan mahasiswa.
3. Bagi peneliti lain yang bergerak dalam sistem informasi pendidikan tinggi, Hasil rancangan aplikasi ini menjadi referensi dalam pengembangan solusi teknologi informasi untuk efisiensi dan kenyamanan di pendidikan tinggi, mendorong inovasi, dan memberikan kontribusi positif dalam pengembangan sistem informasi yang mendukung efektivitas operasional kampus.

## **Batasan Penelitian**

Batasan dari penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya berfokus pada perancangan dan implementasi aplikasi point of sales berbasis website pada fitur back office untuk kantin asrama di Universitas Telkom.
2. Aplikasi yang dirancang dalam penelitian ini hanya mencakup fungsi manajemen akun tenant, manajemen menu, dan laporan penjualan.
3. Meski metode scrum digunakan dalam perancangan aplikasi ini, penelitian ini tidak akan membahas secara detail tentang metode scrum itu sendiri. Fokus penelitian lebih kepada penerapan metode tersebut dalam konteks pengembangan aplikasi point of sales.
4. Evaluasi kinerja dan kegunaan aplikasi ini hanya akan dilakukan terhadap admin dan tenant Universitas Telkom yang menggunakan kantin universitas.

# Tinjauan Pustaka

## **Aplikasi Point of Sales**

Aplikasi Point of Sale (POS) adalah perangkat lunak yang digunakan untuk mengatur dan mempermudah berbagai kegiatan penjualan dan transaksi di toko atau bisnis ritel. POS sangat penting dalam operasional harian bisnis karena menggantikan sistem kasir tradisional dengan solusi digital yang lebih efisien dan akurat. Aplikasi POS umumnya memiliki fitur seperti pembayaran, manajemen inventaris, pelacakan penjualan, dan laporan keuangan.

Penggunaan aplikasi POS dalam konteks bisnis memiliki berbagai manfaat. Pengguna dapat meningkatkan efisiensi operasional, mengurangi kesalahan dalam penghitungan, mempercepat waktu transaksi, dan mengoptimalkan pengelolaan inventaris. Selain itu, aplikasi POS juga menyediakan data dan analisis yang canggih, yang memungkinkan pemilik bisnis untuk memahami tren penjualan, melacak performa produk, dan mengidentifikasi peluang bisnis baru (Suryadi & Setiawan, 2022).

## **Website Back Office**

Website Back Office adalah sebuah bagian penting dari sebuah situs web yang dirancang khusus untuk mengelola dan mengendalikan berbagai aspek dari situs tersebut. Sebagai inti dari operasional sebuah situs web, Back Office menyediakan alat dan fitur yang memungkinkan pemilik situs untuk mengelola konten, mengontrol akses pengguna, dan melacak kinerja situs. Dalam konteks umum, Website Back Office mencakup antarmuka administratif yang memungkinkan pengguna untuk mengubah atau menambahkan konten, mengelola pengguna dan izin akses, mengintegrasikan layanan pihak ketiga, dan melacak analitik situs web.

Dalam bidang pos (Point of Sale), Website Back Office memiliki peran yang krusial dalam menjalankan operasional bisnis. Back Office pos menyediakan alat yang memungkinkan pemilik bisnis untuk mengelola inventaris, mengatur harga, dan mengontrol transaksi penjualan. Dengan adanya Back Office, pemilik bisnis dapat melacak dan mengelola stok barang, memperbaharui harga produk, dan memantau penjualan harian. Selain itu, Back Office pos juga dapat menghasilkan laporan keuangan dan analitik yang membantu pemilik bisnis dalam mengambil keputusan yang lebih baik (Susanto & Utama, 2020).

## **Metode Scrum**

Metode Scrum adalah kerangka kerja manajemen proyek yang termasuk dalam metodologi Agile. Metode ini digunakan untuk mengelola proyek yang kompleks dan dinamis, terutama dalam pengembangan perangkat lunak. Fokus utama dari Scrum adalah mengatur waktu dan menggunakan pendekatan kolaboratif dan iteratif dalam menciptakan produk yang bernilai. Dalam Scrum, tim proyek bekerja dalam periode waktu yang disebut sprint, biasanya berlangsung selama 1 hingga 4 minggu.

Proses kerja dalam Scrum melibatkan tiga peran utama: Product Owner, Scrum Master, dan Tim Pengembang. Product Owner bertanggung jawab untuk menggambarkan kebutuhan pengguna dan merancang produk yang sesuai. Scrum Master berfungsi sebagai pemimpin tim yang memastikan penerapan Scrum yang efektif dan mengatasi hambatan yang mungkin muncul. Tim Pengembang terdiri dari individu yang bertanggung jawab untuk menyelesaikan pekerjaan dalam sprint.

Scrum juga menggunakan artefak seperti Product Backlog, Sprint Backlog, dan Burndown Chart untuk membantu mengelola proyek. Product Backlog adalah daftar tugas yang harus diselesaikan dalam proyek, sedangkan Sprint Backlog adalah subset dari Product Backlog yang harus diselesaikan dalam sprint tertentu. Burndown Chart digunakan untuk melacak kemajuan dan memperlihatkan transparansi dalam menyelesaikan pekerjaan (Arifianto, 2021).

## **PHP**

PHP (Hypertext Preprocessor) adalah sebuah bahasa pemrograman server-side yang dirancang khusus untuk pengembangan aplikasi web. PHP awalnya dibuat sebagai alat untuk membangun halaman web dinamis dan interaktif, di mana kode PHP dieksekusi di sisi server untuk menghasilkan HTML yang akan dikirim ke browser pengguna. PHP memiliki sintaks yang mirip dengan C dan bahasa pemrograman lainnya, membuatnya relatif mudah dipelajari dan digunakan oleh para pengembang.

Salah satu kelebihan utama PHP adalah kemampuannya untuk berinteraksi dengan berbagai jenis basis data, seperti MySQL, PostgreSQL, dan Oracle. Dengan PHP, pengembang dapat membuat aplikasi web yang dapat mengambil data dari basis data, melakukan manipulasi, dan menyimpan perubahan kembali ke basis data. Kemampuan ini menjadikan PHP sebagai salah satu bahasa pemrograman yang populer dalam pengembangan aplikasi web berbasis database.

Selain itu, PHP juga menyediakan berbagai fungsi dan pustaka bawaan yang mendukung pengembangan aplikasi web yang kompleks. PHP dapat digunakan untuk mengelola sesi pengguna, mengunggah file, memproses formulir, mengirim email, dan masih banyak lagi. Karena kepopulerannya, terdapat juga banyak kerangka kerja (framework) PHP seperti Laravel, CodeIgniter, dan Symfony yang membantu dalam pengembangan aplikasi web dengan pola desain yang terstruktur (Ardianto & Suryanto, 2021).

## **Laravel**

Laravel adalah sebuah kerangka kerja (framework) PHP yang open-source dan sangat populer dalam pengembangan aplikasi web. Dirancang dengan konsep MVC (Model-View-Controller), Laravel menyediakan struktur dan alat yang kuat untuk membangun aplikasi web yang efisien dan mudah dikelola. Laravel menawarkan berbagai fitur termasuk manajemen basis data yang canggih, routing yang fleksibel, sistem templating, keamanan yang kuat, serta berbagai pustaka dan komponen yang siap pakai.

Salah satu keunggulan utama Laravel adalah Eloquent ORM, yang memungkinkan pengembang untuk berinteraksi dengan basis data menggunakan kode PHP yang ekspresif dan mudah dipahami. Eloquent ORM menyederhanakan tugas-tugas seperti pengambilan data, manipulasi, dan penggunaan query kompleks, sehingga mempercepat pengembangan aplikasi web. Selain itu, Laravel juga menyediakan fitur migrasi basis data yang memudahkan dalam mengelola struktur basis data selama proses pengembangan dan pemeliharaan aplikasi.

Laravel juga dikenal dengan komunitas yang aktif dan luas, yang menyediakan dokumentasi yang lengkap, tutorial, dan sumber daya yang berguna. Komunitas ini juga berkontribusi dalam mengembangkan berbagai paket dan ekstensi yang dapat digunakan untuk memperluas fitur Laravel. Dengan adanya komunitas yang solid, pengembang dapat dengan mudah mencari solusi, berbagi pengetahuan, dan terus memperbarui keahlian mereka dalam pengembangan dengan menggunakan Laravel (Prasetyo & Cahyono 2020).

## **Postman**

Postman merupakan aplikasi yang sangat populer di kalangan pengembang perangkat lunak untuk menguji API (Application Programming Interface). Dikembangkan oleh Postman Inc., aplikasi ini menawarkan antarmuka yang mudah dipahami dan komprehensif untuk menguji, memvalidasi, dan mendokumentasikan API. Dengan Postman, pengembang dapat mengirimkan permintaan HTTP ke API, menangani respons, dan melakukan debugging untuk memastikan bahwa API berfungsi dengan baik.

Salah satu fitur utama dari Postman adalah kemampuannya untuk mengirimkan permintaan HTTP ke berbagai endpoint API. Pengembang dapat dengan mudah menentukan metode HTTP yang akan digunakan (seperti GET, POST, PUT, DELETE), melampirkan header dan parameter yang diperlukan, dan mengirimkan permintaan ke server. Postman juga menyediakan fitur pengelolaan koleksi permintaan, yang memungkinkan pengembang untuk mengorganisir dan menyimpan permintaan yang sering digunakan, sehingga pengujian dan debugging dapat dilakukan dengan lebih efisien.

Selain itu, Postman juga dilengkapi dengan fitur pemantauan API dan pengujian otomatis. Pengembang dapat membuat skrip pengujian yang dapat dijalankan secara otomatis untuk memverifikasi kinerja dan keandalan API. Hal ini memungkinkan pengembang untuk mengotomatisasi proses pengujian dan mempercepat siklus pengembangan (Ahmad Ramadhan, 2019).

## **Midtrans**

Midtrans adalah payment gateway yang memenuhi kebutuhan bisnis online dengan menawarkan layanan berbagai metode pembayaran. Layanan ini dimungkinkan Pekerjaan operator di lokasi menjadi lebih mudah dan omset meningkat. Pembayaran kartu, transfer bank, debit langsung, e-wallet, over the counter dan lainnya ditawarkan sebagai metode pembayaran (Muhammad Ramadhan Febrianto,2020).

Dengan menggunakan Midtrans, pembeli bisa mendapatkan manfaat proses pembayaran yang mudah dan nyaman. Selain itu, laporan pembayaran dapat ditangkap dengan baik, meminimalkan hilangnya kesalahan atau kesalahan yang salah dan tidak pantas atau dokumen dalam proses pengambilan data (Erick Febrianto,dkk,2018).

## **Penelitian Terdahulu**

| **Nama Peneliti** | **Judul** | **Masalah** |
| --- | --- | --- |
| Ilham Fannani | Perancangan Aplikasi POINT OF SALES Berbasis Website Togoide Studi Kasus UKM Laundry Menggunakan Metode Iterative dan Incremental | Penelitian ini mengangkat masalah Website Togoide Studi Kasus UMKM Laundry |
| **Solusi dan Hasil :** User Interface Pada Aplikasi Customer Layanan Pemesanan Makanan “EatAja” | | |
| **Perbandingan :** Penelitian terdahulu ini fokus pada perancangan aplikasi Point of Sales (POS) berbasis website untuk usaha kecil menengah (UKM) laundry. Metode yang digunakan adalah metode iterative dan incremental, yang berfokus pada pengembangan bertahap dan peningkatan secara berulang terhadap aplikasi tersebut. | | |

### Tabel II.01 Penelitian Terdahulu: Ilham Fannani

Sumber : (Fannani, 2020)

| **Nama Peneliti** | **Judul** | **Masalah** |
| --- | --- | --- |
| Nailul Marom | Perancangan User Interface Pada Aplikasi Customer Layanan Pemesanan Makanan “EatAja” Menggunakan Metode User Centered Design | Penelitian ini mengangkat masalah User Interface Pada Aplikasi Customer Layanan Pemesanan Makanan “EatAja” |
| **Solusi dan Hasil :** User Interface Pada Aplikasi Customer Layanan Pemesanan Makanan “EatAja” | | |
| **Perbandingan :** Penelitian terdahulu ini bertujuan untuk merancang antarmuka pengguna (user interface) pada aplikasi "EatAja" yang digunakan oleh pelanggan untuk pemesanan makanan. Metode yang digunakan adalah metode User Centered Design, yang berfokus pada kebutuhan dan pengalaman pengguna dalam merancang antarmuka yang intuitif dan mudah digunakan. | | |

### Tabel II.02 Penelitian Terdahulu: Nailul Marom

Sumber : (Nailul, 2020)

| **Nama Peneliti** | **Judul** | **Masalah** |
| --- | --- | --- |
| Muhamad Noufal Ajriya Siddik | Perancangan User Interface Mitra Sebagai Penerima Pesanan Pada Aplikasi EATAJA Mitra Dengan Metode Goal-Directed Design | Penelitian ini mengangkat masalah Perancangan User Interface Mitra Sebagai Penerima Pesanan Pada Aplikasi EATAJA |
| **Solusi dan Hasil :** Perancangan User Interface Mitra Sebagai Penerima Pesanan Pada Aplikasi EATAJA | | |
| **Perbandingan :** Penelitian terdahulu ini berfokus pada perancangan antarmuka pengguna bagi mitra yang bertindak sebagai penerima pesanan pada aplikasi "EATAJA Mitra". Metode yang digunakan adalah metode Goal-Directed Design, yang berfokus pada pemahaman tujuan dan kebutuhan pengguna dalam merancang antarmuka yang mendukung pencapaian tujuan tersebut. | | |

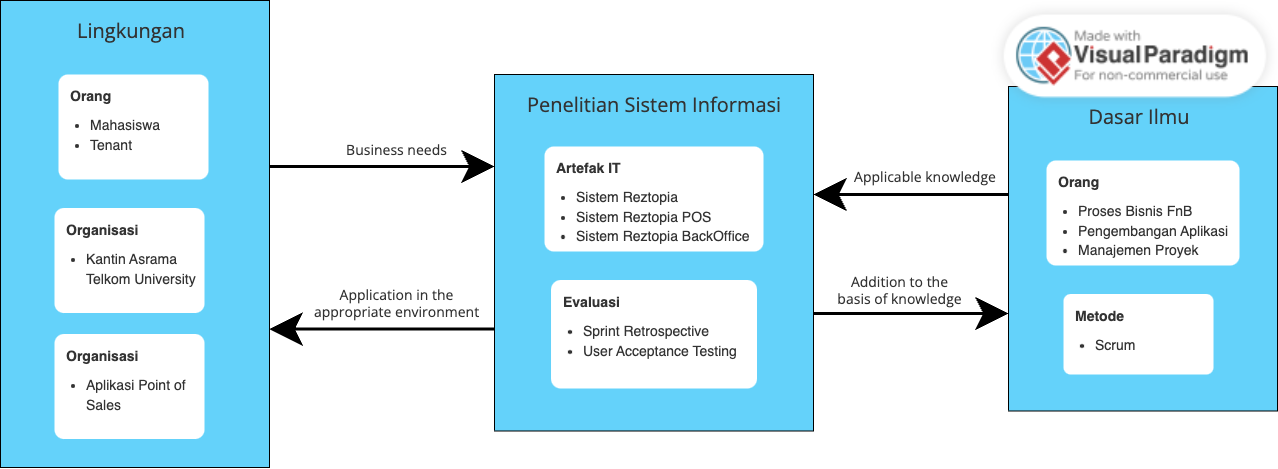
### Tabel II.03 Penelitian Terdahulu: Muhamad Noufal Ajriya Siddik

Sumber : (Siddik, 2020)

# 

# Metodologi Penelitian

## **Model Konseptual**

Pengembangan model konseptual merupakan representasi abstrak yang digunakan oleh peneliti untuk memahami, menerapkan, dan mengevaluasi penelitian. Dalam penelitian ini, digunakan model konseptual berdasarkan paradigma design science dengan tujuan mengembangkan solusi berbasis teknologi yang terkait dengan permasalahan dalam desain dan bisnis.

### Gambar III.01 Model Konseptual

Model konseptual tersebut digunakan untuk merancang aplikasi point of sales berbasis website khusus untuk kantin Telkom University dengan menggunakan metode Scrum. Aplikasi ini melibatkan lingkungan yang terdiri dari mahasiswa sebagai pelanggan dan tenant sebagai pihak yang berjualan. Organisasi yang terlibat adalah Kantin Telkom University sebagai UMKM yang memerlukan aplikasi ini untuk memudahkan transaksi dan manajemen penjualan. Teknologi yang digunakan meliputi website dan aplikasi point of sales yang terintegrasi dengan mesin kasir dan sistem inventaris.

Penelitian ini menggunakan artefak IT seperti reztopia, reztopia pos, dan reztopia backoffice dalam bidang Sistem Informasi. Evaluasi dilakukan melalui sprint retrospective dan user acceptance testing (UAT). Dasar ilmu yang digunakan meliputi proses bisnis F&B (Food and Beverage), pengembangan aplikasi, dan manajemen proyek dengan menggunakan metode Scrum. Dengan menggunakan model konseptual ini, diharapkan perancangan aplikasi point of sales berbasis website dapat membantu Kantin Telkom University dalam meningkatkan efisiensi transaksi dan manajemen penjualan (Pramono & Utami, 2021).

## **Sistematika Penyelesaian Masalah**

### Gambar III.02 Metode Scrum

Sistematika penyelesaian masalah dilakukan dengan tahapan sebagai berikut (Utami & Pramono, 2020):

1. Perencanaan awal (Initial Planning): Pada tahap ini, tim proyek harus menetapkan visi jangka panjang dan tujuan proyek yang akan dilakukan. Kemudian, tim perlu mengumpulkan kebutuhan proyek dari pihak terkait dan mengutamakan kebutuhan tersebut dalam daftar tugas proyek (Product Backlog). Ini dilakukan untuk memastikan bahwa proyek dapat menghasilkan solusi yang memenuhi kebutuhan pihak terkait dan sesuai dengan tujuan dan visi proyek.
2. Sprint Planning: Pada tahap ini, tim proyek akan memilih item-item dari Product Backlog yang akan dikerjakan pada periode waktu tertentu yang disebut Sprint, dan mencatat item-item tersebut ke dalam daftar tugas Sprint (Sprint Backlog). Selanjutnya, tim akan membagi tugas-tugas dalam Sprint Backlog menjadi bagian yang lebih kecil dan konkret, menentukan waktu dan usaha yang diperlukan untuk setiap tugas. Tujuan dari tahap ini adalah untuk mengoptimalkan penggunaan waktu dan sumber daya, sehingga proyek dapat diselesaikan dengan efisien.
3. Sprint: Pada tahap ini, tim proyek akan melaksanakan tugas-tugas yang direncanakan dalam Sprint Backlog dan berkoordinasi dengan anggota tim lainnya. Selama Sprint, anggota tim akan fokus pada tugas-tugas yang ditugaskan dan memastikan bahwa semua tugas tersebut diselesaikan sesuai dengan waktu yang ditentukan dalam Sprint Planning.
4. Daily Stand-up: Pertemuan harian singkat untuk melaporkan kemajuan, hambatan, dan koordinasi tugas antara anggota tim. Pertemuan harian ini bertujuan untuk memastikan bahwa semua anggota tim tetap berada pada jalur yang sama, menjaga proyek tetap berjalan sesuai jadwal, serta mengatasi masalah atau hambatan yang muncul sebelum menjadi masalah yang lebih besar.
5. Sprint Review: Pada tahap ini, tim proyek akan mengevaluasi hasil pengembangan, melakukan demonstrasi hasil kerja kepada pihak terkait, dan membahas umpan balik serta penyesuaian rencana. Hal ini dilakukan untuk memastikan bahwa hasil pengembangan sesuai dengan kebutuhan pihak terkait dan memperbaiki rencana jika diperlukan.
6. Sprint Retrospective: Pada tahap ini, tim proyek akan mengevaluasi proses dan hasil Sprint, mengidentifikasi area perbaikan dan merencanakan tindakan yang diperlukan, serta mengembangkan strategi untuk meningkatkan efektivitas dan efisiensi tim. Tujuan dari tahap ini adalah untuk memperbaiki proses pengembangan dan meningkatkan efisiensi tim untuk Sprint berikutnya.
7. Pengujian (Testing): Pada tahap ini, tim proyek akan menyusun skenario dan kasus uji, melaksanakan pengujian pada tingkat unit, integrasi, dan sistem, serta mencatat dan melacak bug serta penyelesaiannya. Tahap ini dilakukan untuk memastikan bahwa solusi yang dikembangkan dapat berfungsi dengan baik dan memenuhi kebutuhan pihak terkait.
8. Peluncuran (Release): Pada tahap ini, tim proyek akan menyusun dan memvalidasi dokumen peluncuran, melaksanakan rencana peluncuran, dan mendistribusikan solusi ke pengguna. Tahap ini bertujuan untuk memperkenalkan solusi yang telah dikembangkan ke lingkungan produksi atau pengguna akhir.
9. Kesimpulan (Conclusion): Tahap terakhir dalam metode Scrum adalah kesimpulan proyek. Pada tahap ini, dilakukan evaluasi keseluruhan proyek, penutupan proyek, dan transfer pengetahuan ke tim pemeliharaan atau proyek berikutnya. Selain itu, juga disusun laporan akhir proyek yang mencakup hasil proyek, pelajaran yang dipetik, dan rekomendasi untuk proyek selanjutnya. Hal ini bertujuan untuk memaksimalkan nilai bisnis yang dihasilkan oleh proyek dan meningkatkan pengembangan tim untuk proyek mendatang.

## **Alasan Pemilihan Metode**

Alasan pemilihan metode Scrum adalah karena Scrum memiliki beberapa keunggulan dalam pengembangan perangkat lunak, seperti yang dikemukakan oleh Septiandani & Fauziah (2019). Keunggulan-keunggulan tersebut antara lain:

1. Fleksibilitas dalam pengembangan: Scrum memungkinkan tim untuk dengan cepat dan efektif menyesuaikan perubahan kebutuhan atau prioritas dalam pengembangan perangkat lunak.
2. Kolaborasi yang baik: Scrum mendorong kerja sama dan memperkuat tim pengembang. Setiap anggota tim memiliki peran yang jelas, sehingga memastikan kolaborasi yang efektif dalam mencapai tujuan proyek.
3. Menghasilkan produk yang lebih baik: Dengan pendekatan yang berfokus pada nilai bisnis yang diinginkan, Scrum membantu tim pengembang untuk menghasilkan produk yang lebih baik dan lebih relevan dengan kebutuhan pengguna.
4. Responsif terhadap perubahan: Scrum memungkinkan tim untuk dengan cepat merespons perubahan yang terjadi pada produk atau lingkungan bisnis. Hal ini memungkinkan pengembang untuk mengatasi tantangan dan peluang baru dengan lebih efisien.

Dalam konteks proyek ini, metode Scrum dipilih karena melibatkan banyak pemangku kepentingan dan kebutuhan yang terus berkembang. Oleh karena itu, diperlukan pendekatan pengembangan perangkat lunak yang fleksibel dan adaptif seperti Scrum untuk menghadapi perubahan dengan efisien.

## **Pengumpulan Data**

Data dikumpulkan menggunakan teknik pengumpulan data sebagai berikut (Daryanto & Rahardjo, 2021):

1. Observasi

* Observasi langsung dilakukan di kantin Telkom University untuk memperoleh informasi tentang kebutuhan sistem Point of Sales dan proses bisnis di kantin tersebut.
* Observasi juga dilakukan melalui wawancara dengan beberapa pihak terkait di kantin, seperti owner kantin dan mahasiswa.

1. Wawancara

Wawancara dilakukan dengan owner kantin dan mahasiswa untuk mendapatkan informasi detail tentang proses bisnis dan kebutuhan sistem Point of Sales di kantin.

1. Studi dokumentasi

Studi dokumentasi dilakukan untuk mengumpulkan informasi tentang kebutuhan bisnis dan teknologi yang diperlukan untuk pengembangan aplikasi Point of Sales berbasis website.

Setelah data terkumpul, data tersebut akan dianalisis menggunakan teknik analisis data yang sesuai dengan jenis data yang dikumpulkan, seperti analisis kualitatif dan analisis kuantitatif. Hasil analisis data akan digunakan sebagai panduan dalam merancang sistem Point of Sales berbasis website yang sesuai dengan kebutuhan dan proses bisnis di kantin Telkom University.

## **Pengolahan Data**

Setelah proses pengumpulan data selesai, langkah-langkah berikut dilakukan dalam pengolahan data (Darmawan & Nugroho, 2021):

1. Pembersihan Data

Data yang telah terkumpul perlu dibersihkan dan dihapus data yang tidak relevan atau duplikat agar data yang digunakan dalam analisis lebih akurat.

1. Analisis Data

Data kemudian dianalisis menggunakan teknik analisis data yang sesuai, seperti analisis deskriptif, analisis korelasi, dan analisis regresi, sesuai dengan tujuan penelitian.

1. Visualisasi Data

Hasil analisis data divisualisasikan menggunakan grafik, tabel, dan diagram untuk mempermudah pemahaman dan interpretasi data.

Data yang telah diolah kemudian digunakan dalam perancangan sistem Point of Sales berbasis website yang sesuai dengan kebutuhan dan proses bisnis di kantin Telkom University. Selain itu, data juga digunakan untuk melakukan evaluasi terhadap aplikasi Point of Sales yang telah dikembangkan.

## **Metode Evaluasi**

Setelah pengembangan aplikasi Point of Sales berbasis website untuk pelanggan selesai, aplikasi tersebut akan dievaluasi menggunakan metode evaluasi berikut (Maulidia & Santoso, 2021):

1. User Acceptance Testing (UAT)

UAT dilakukan oleh pengguna yang telah ditentukan untuk menguji fungsionalitas dan kinerja aplikasi. Hasil dari UAT akan digunakan untuk menentukan apakah aplikasi siap digunakan secara langsung atau perlu dilakukan perbaikan.

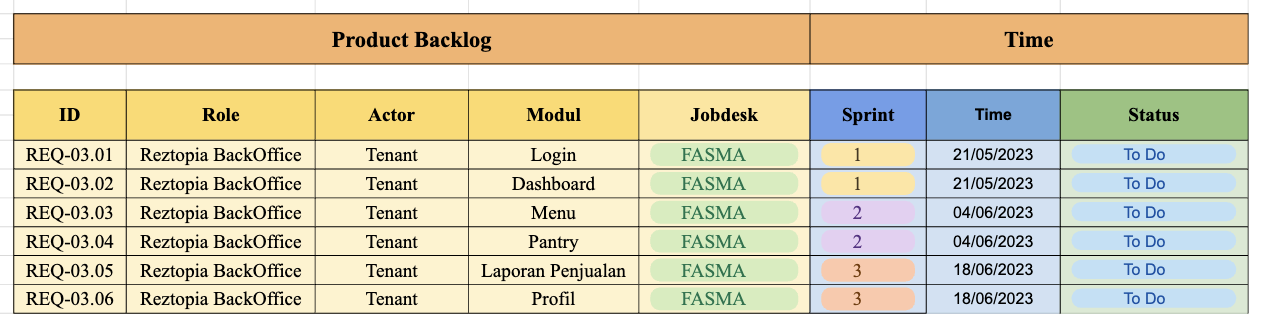
1. Testing Performance

Testing Performance dilakukan untuk menguji kemampuan aplikasi dalam menangani beban pengguna dan memastikan ketersediaan aplikasi saat digunakan secara bersamaan oleh banyak pengguna.

1. Usability Testing

Usability testing dilakukan untuk menguji sejauh mana aplikasi mudah digunakan oleh pengguna dan memenuhi kebutuhan pengguna.

## **Rencana Jadwal Kegiatan**

Sprint adalah suatu periode dalam metode Scrum di mana tim pengembang bekerja secara intensif untuk menyelesaikan item backlog yang telah dipilih selama Sprint Planning (Utami & Pramono, 2020).

### Gambar III.03 Sprint

# Analisis dan Perancangan

## **Proses Bisnis**

### Gambar IV.01 Proses Bisnis Back Office Pada Kantin Telkom University

Dalam konteks penggunaan aplikasi Point of Sales (POS) berbasis website untuk Tenant di Kantin Universitas Telkom, terdapat serangkaian proses bisnis yang dapat dijelaskan sebagai berikut. Tenant memulai dengan login ke website Reztopia BackOffice menggunakan akun yang terdaftar. Setelah login, Tenant dapat menambahkan kategori dan menu makanan atau minuman ke dalam sistem. Sementara itu, mahasiswa dapat login ke website Reztopia POS untuk membuat pesanan dan menyelesaikannya. Tenant dapat melihat daftar pesanan yang masuk, mengkonfirmasi pesanan yang telah selesai, dan melihat laporan penjualan untuk menganalisis kinerja usaha. Proses bisnis tersebut berakhir setelah melihat laporan penjualan. Proses ini memungkinkan Tenant untuk mengelola menu dan pesanan secara efisien, sementara mahasiswa dapat dengan mudah melakukan pemesanan melalui website Reztopia POS.

## **Product Backlog**

1. Fungsional

| **ID** | **Apps** | **Actor** | **Epic** | **Story** | **User Story** |
| --- | --- | --- | --- | --- | --- |
|
| REQ-02.01 | BackOffice | Tenant | Login | Melakukan Login | Sebagai tenant, saya ingin dapat login ke dalam aplikasi agar dapat mengakses fungsionalitasnya. |
| REQ-02.02 | BackOffice | Tenant | Logout | Melakukan Logout | Sebagai tenant, saya ingin melakukan logout dari aplikasi BackOffice ketika saya selesai menggunakannya untuk menjaga keamanan akun saya. |
| REQ-02.03 | BackOffice | Tenant | Dashboard | Melihat Dashboard | Sebagai tenant, saya ingin melihat dashboard yang menyediakan fitur laporan penjualan dan secara visual dalam bentuk grafik dan juga daftar produk terlaris. |
| REQ-02.04 | BackOffice | Tenant | Menu | Mengelola Kategori | Sebagai tenant, saya ingin dapat membuat, mengubah, dan menghapus kategori pada halaman menu dalam aplikasi BackOffice. |
| REQ-02.05 | BackOffice | Tenant | Menu | Mengelola Menu | Sebagai tenant, saya ingin dapat membuat, mengubah, dan menghapus menu pada halaman menu dalam aplikasi BackOffice. |
| REQ-02.06 | BackOffice | Tenant | Laporan Penjualan  Tenant | Melihat Laporan Penjualan | Sebagai tenant, saya ingin dapat melihat rincian penjualan harian atau periode tertentu seperti nama pemesan, nomor meja, dan total pembayaran |
| REQ-02.07 | BackOffice | Tenant | Profil | Mengelola Profil | Sebagai tenant, saya ingin dapat mengatur dan memperbarui informasi profil saya seperti nama, BackOffice, foto profil, dan kata sandi pada halaman profil. |
| REQ-02.10 | BackOffice | Admin | Login | Melakukan Login | Sebagai admin, saya ingin dapat login ke dalam aplikasi agar dapat mengakses fungsionalitasnya. |
| REQ-02.11 | BackOffice | Admin | Logout | Melakukan Logout | Sebagai admin, saya ingin melakukan logout dari aplikasi BackOffice ketika saya selesai menggunakannya untuk menjaga keamanan akun saya. |
| REQ-02.12 | BackOffice | Admin | Dashboard | Melihat Dashboard | Sebagai admin, saya ingin melihat dashboard yang menyediakan fitur laporan penjualan pada pujasera, jumlah orderan, jumlah tenant yang aktif, jumlah tenant tidak aktif dan juga daftar tenant terlaris |
| REQ-02.13 | BackOffice | Admin | Tenant | Mengelola Akun Tenant | Sebagai admin, saya ingin mengatur data profil, menambahkan data tenant baru, dan mengatur tenant untuk aktif maupun non aktif |
| REQ-02.14 | BackOffice | Admin | Laporan Penjualan  Tenant | Melihat Laporan Penjualan | Sebagai admin, saya ingin dapat melihat rincian penjualan harian atau periode tertentu seperti nama pemesan, nomor meja, dan total pembayaran dari setiap tenant |
| REQ-02.15 | BackOffice | Admin | Profil | Mengelola Profil | Sebagai tenant, saya ingin dapat mengatur dan memperbarui informasi profil saya seperti BackOffice, foto profil, dan kata sandi pada halaman profil. |

### Tabel IV.01 Kebutuhan Fungsionalitas Reztopia Fitur Back Office

1. Non Fungsional

| **No** | **Aspek** | **Detail** | **Deskripsi** |
| --- | --- | --- | --- |
|
| 1 | Operasional | Portability | Aplikasi Reztopia harus dapat berfungsi dengan baik di berbagai browser web dan responsif untuk berbagai ukuran layar. |
| Maintainability | Aplikasi Reztopia harus mudah diperbarui dan diperbaiki. |
| Availability | Aplikasi Reztopia harus selalu tersedia untuk pengguna. |
| Scalability | Aplikasi Reztopia harus dapat menangani peningkatan beban penggunaan jika diperlukan. |
| 2 | Performa | Response Time | Aplikasi Reztopia harus dapat memberikan respon yang cepat kepada pengguna. |
| Efficiency | Aplikasi Reztopia menggunakan sumber daya sistem secara efisien. |
| Reliability | Aplikasi Reztopia berfungsi secara konsisten dan dapat diandalkan. |
| 3 | Keamanan | Data Protection | Aplikasi Reztopia melindungi data pengguna melalui enkripsi data |
| Access Control | Aplikasi Reztopia mengontrol akses berdasarkan peran pengguna. |

### Tabel IV.02 Kebutuhan Fungsionalitas Reztopia Fitur Back Office

1. Aktor

| **No** | **Aktor** | **Deskripsi** |
| --- | --- | --- |
|
| 1 | Admin | Admin merupakan pengelola kantin Telkom University yang memanfaatkan aplikasi Reztopia untuk memfasilitasi manajemen tenant dan memastikan operasional kantin berjalan lancar dan efisien. |
| 2 | Tenant | Tenant merupakan pelaku usaha yang menjual makanan atau minuman yang beroperasi di kantin asrama Telkom University. Dengan bantuan aplikasi Reztopia, mereka mampu mengelola operasional usaha mereka dengan cara yang lebih efisien dan efektif. |

### Tabel IV.03 Aktor Reztopia Fitur Back Office

## **Use Case Diagram**

### Gambar IV.02 Use Case Diagram Reztopia Fitur Back Office

Diagram use case di atas menggambarkan berbagai interaksi yang mungkin dilakukan oleh aktor, yaitu 'Tenant' dan 'Admin', dengan sistem 'Reztopia BackOffice'. 'Tenant' dapat melakukan serangkaian aksi seperti login, logout, melihat dashboard, mengelola kategori dan menu, melihat laporan penjualan, dan mengelola profil. Sementara 'Admin' juga dapat melakukan aksi yang mirip, dengan tambahan beberapa aksi spesifik seperti melihat laporan penjualan seluruh tenant dan mengelola akun tenant. Diagram ini memberikan gambaran visual yang jelas tentang siapa yang dapat melakukan apa dalam sistem dan bagaimana interaksi mereka dengan berbagai fitur sistem.

## **Use Case Scenario**

| ID | REQ-02.01 |
| --- | --- |
| Title | Melakukan Login di Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang tenant dapat melakukan login ke dalam aplikasi Reztopia BackOffice untuk mengakses fungsionalitasnya. |
| Primary Actor | Tenant |
| Precondition | Tenant telah memilih opsi "Login" di Reztopia BackOffice. |
| Postcondition | Tenant berhasil login dan dapat mengakses halaman utama aplikasi. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi "Login" |  |
|  | 1. Menampilkan halaman login |
| 1. Memasukan data email dan password |  |
|  | 1. Memvalidasi data |
|  | 1. Jika data valid, maka :  * Memberikan akses ke aplikasi * Menampilkan halaman utama |
|  | 1. Jika data tidak valid, maka :  * Menampilkan pesan error * Kembali ke halaman login |
| Extension | - |
| Frequency of use | Proses ini dilakukan setiap kali tenant ingin masuk ke dalam aplikasi Reztopia BackOffice. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.04 Use Case Scenario REQ-02.01

| ID | REQ-02.02 |
| --- | --- |
| Title | Melakukan Register di Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang tenant dapat melakukan logout dari aplikasi Reztopia BackOffice setelah selesai menggunakannya, untuk menjaga keamanan akunnya. |
| Primary Actor | Tenant |
| Precondition | Tenant telah masuk ke dalam sistem Reztopia BackOffice. |
| Postcondition | Tenant berhasil melakukan logout dan kembali ke halaman login. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi Logout |  |
|  | 1. Menampilkan halaman Login |
| Extension | - |
| Frequency of use | Proses ini akan dilakukan oleh tenant setiap kali ingin logout dari aplikasi Reztopia BackOffice setelah selesai menggunakannya. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.05 Use Case Scenario REQ-02.02

| ID | REQ-02.03 |
| --- | --- |
| Title | Melihat Dashboard Tenant di Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang tenant dapat melihat dashboard di aplikasi Reztopia BackOffice yang menyediakan fitur laporan penjualan secara visual dalam bentuk grafik dan juga daftar produk terlaris. |
| Primary Actor | Tenant |
| Precondition | Tenant telah masuk ke dalam sistem Reztopia BackOffice. |
| Postcondition | Tenant berhasil melihat dashboard yang menampilkan laporan penjualan, grafik penjualan, dan daftar produk terlaris. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Masuk ke aplikasi Reztopia BackOffice |  |
|  | 1. Menampilkan dashboard |
|  | 1. Menampilkan laporan penjualan |
|  | 1. Menampilkan grafik penjualan |
|  | 1. Menampilkan daftar produk terlaris |
| Extension | - |
| Frequency of use | Proses ini dapat dilakukan oleh tenant setiap kali ingin melihat laporan penjualan, grafik penjualan, dan daftar produk terlaris pada dashboard. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.06 Use Case Scenario REQ-02.03

| ID | REQ-02.04 |
| --- | --- |
| Title | Mengelola Kategori Tenant di Reztopia BackOffice |
| Description | Pengguna dapat melihat seluruh transaksi pada website, baik itu transaksi penjualan harian maupun laporan penjualan keseluruhan |
| Primary Actor | Tenant |
| Precondition | Tenant telah masuk ke dalam sistem Reztopia BackOffice. |
| Postcondition | Tenant berhasil membuat, mengubah, atau menghapus kategori pada halaman menu. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi "Menu" |  |
|  | 1. Menampilkan halaman menu |
| 1. "Kelola Kategori" |  |
|  | 1. Menampilkan halaman kelola kategori |
| 1. Memilih opsi "Buat Kategori" |  |
|  | 1. Menampilkan halaman pembuatan kategori |
| 1. Memasukkan detail kategori |  |
|  | 1. Mengkonfirmasi pembuatan kategori |
| 1. Memilih opsi "Ubah Kategori" |  |
|  | 1. Menampilkan halaman pengubahan kategori |
| 1. Memilih kategori yang akan diubah |  |
|  | 1. Menampilkan halaman detail kategori |
| 1. Memperbarui detail kategori |  |
|  | 1. Mengkonfirmasi perubahan kategori |
| 1. Memilih opsi "Hapus Kategori" |  |
|  | 1. Menampilkan konfirmasi penghapusan kategori |
| 1. Mengkonfirmasi penghapusan kategori |  |
|  | 1. Melakukan penghapusan kategori |
| Extension | - |
| Frequency of use | Proses ini akan dilakukan oleh tenant ketika ingin membuat, mengubah, atau menghapus kategori pada halaman menu. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.07 Use Case Scenario REQ-02.04

| ID | REQ-02.05 |
| --- | --- |
| Title | Mengelola Menu di Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang tenant dapat membuat, mengubah, dan menghapus menu pada halaman menu dalam aplikasi Reztopia BackOffice. |
| Primary Actor | Tenant |
| Precondition | Tenant telah masuk ke dalam sistem Reztopia BackOffice. |
| Postcondition | Tenant berhasil membuat, mengubah, atau menghapus menu. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi “Menu” |  |
|  | 1. Menampilkan halaman menu |
| 1. Memilih opsi “Tambah Menu” |  |
| 1. Memasukan data menu |  |
|  | 1. Memvalidasi data menu |
|  | 1. Jika data valid :  * Menambahkan menu ke daftar menu * Menampilkan pesan sukses |
|  | 1. Jika data tidak valid :  * Menampilkan pesan error * Kembali ke halaman menu |
| 1. Memilih opsi “Ubah Menu” |  |
| 1. Memilih menu data perubahan menu |  |
| 1. Menyimpan data perubahan |  |
|  | 1. Memvalidasi data perubahan |
|  | 1. Jika data valid :   Mengubah menu yang dipilih  Menampilkan pesan sukse |
|  | 1. Jika data tidak valid :   Menampilkan pesan error  Kembali ke halaman menu |
| 1. Memilih opsi “Hapus Menu” |  |
| 1. Memilih menu yang akan dihapus |  |
| 1. Mengkonfirmasi penghapusan |  |
|  | 1. Memvalidasi konfirmasi |
|  | 1. Jika konfirmasi valid :  * Menghapus menu yang dipilih * Menampilkan pesan sukses |
|  | 1. Jika konfirmasi tidak valid :  * Menampilkan pesan error * Kembali ke halaman menu |
| Extension | - |
| Frequency of use | Proses ini akan dilakukan oleh tenant saat ingin membuat, mengubah, atau menghapus menu pada halaman menu Reztopia BackOffice. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.08 Use Case Scenario REQ-02.05

| ID | REQ-02.06 |
| --- | --- |
| Title | Melihat Laporan Penjualan Tenant di Reztopia BackOffice |
| Description | Melihat Laporan Penjualan Tenant di Reztopia BackOffice |
| Primary Actor | Tenant |
| Precondition | Tenant telah masuk ke dalam sistem Reztopia BackOffice. |
| Postcondition | Tenant berhasil melihat laporan penjualan sesuai dengan pilihan yang dipilih. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi “Laporan Penjualan” |  |
|  | 1. Menampilkan halaman laporan penjualan |
| 1. Memilih opsi “Rincian Harian” atau “Rincian Periode” |  |
| 1. Jika opsi yang dipilih adalah “Rincian Harian” :  * Memasukan tanggal penjualan |  |
| 1. Jika opsi yang dipilih adalah “Rincian Periode” :  * Memasukan tanggal penjualan |  |
|  | 1. Menampilkan laporan penjualan berdasarkan pilihan yang dipilih |
| Extension | - |
| Frequency of use | Proses ini dapat dilakukan oleh tenant kapan pun mereka ingin melihat laporan penjualan harian atau periode tertentu. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.09 Use Case Scenario REQ-02.06

| ID | REQ-02.07 |
| --- | --- |
| Title | Mengelola Profil Tenant di Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang tenant dapat mengatur dan memperbarui informasi profilnya seperti nama, BackOffice, foto profil, dan kata sandi pada halaman profil di Reztopia BackOffice. |
| Primary Actor | Tenant |
| Precondition | Tenant telah masuk ke dalam sistem Reztopia BackOffice. |
| Postcondition | Tenant berhasil memperbarui informasi profil, foto profil, atau kata sandi. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi “Profil” |  |
|  | 1. Menampilkan halaman profil |
| 1. Memilih opsi “Edit Profil” |  |
| 1. Memperbarui informasi profil |  |
| 1. Menyimpan perubahan |  |
|  | 1. Memvalidasi data |
|  | 1. Jika data valid :  * Memperbarui informasi profil dan menampilkan pesan sukses |
|  | 1. Jika data tidak valid :  * Menampilkan pesan error dan kembali ke halaman profil |
| 1. Memilih opsi “Ubah Foto Profil” |  |
| 1. Memilih foto profil baru |  |
| 1. Mengupload foto profil |  |
|  | 1. Memvalidasi foto profil |
|  | 1. Jika foto valid :  * Memperbarui foto profil dan menampilkan pesan sukses |
|  | 1. Jika foto tidak valid :  * Menampilkan pesan error dan kembali ke halam profil |
| 1. Memilih opsi “Ubah Kata Sandi” |  |
| 1. Memasukan kata sandi lama dan baru |  |
| 1. Memverifikasi kata sandi lama |  |
|  | 1. Memvalidasi kata sandi lama |
|  | 1. Jika kata sandi valid :  * Memperbarui kata sandi dan menampilkan pesan sukses |
|  | 1. Jika kata sandi tidak valid :  * Menampilkan pesan error dan kembali ke halaman profil |
| Extension | - |
| Frequency of use | Proses ini dapat dilakukan oleh tenant sesuai kebutuhan saat ingin mengatur dan memperbarui informasi profil, foto profil, atau kata sandi. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.10 Use Case Scenario REQ-02.07

| ID | REQ-02.10 |
| --- | --- |
| Title | Melakukan Login di Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang admin dapat melakukan login ke dalam aplikasi Reztopia BackOffice agar dapat mengakses fungsionalitasnya. |
| Primary Actor | Admin |
| Precondition | Admin telah memilih opsi "Login" di Reztopia BackOffice. |
| Postcondition | Admin berhasil login dan dapat mengakses halaman utama. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi “Login” |  |
|  | 1. Menampilkan halaman login |
| 1. Memasukan data username dan password |  |
|  | 1. Memvalidasi data |
|  | 1. Jika data valid, maka :  * Memberikan akses ke aplikasi * Menampilkan halaman utama |
|  | 1. Jika data tidak valid, maka :  * Menampilkan pesan error * Kembali ke halaman login |
| Extension | - |
| Frequency of use | Admin akan melalui proses login ini setiap kali ingin mengakses Reztopia BackOffice. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.11 Use Case Scenario REQ-02.10

| ID | REQ-02.11 |
| --- | --- |
| Title | Melakukan Logout dari Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang admin dapat melakukan logout dari aplikasi Reztopia BackOffice ketika selesai menggunakannya untuk menjaga keamanan akun. |
| Primary Actor | Admin |
| Precondition | Admin telah masuk ke dalam aplikasi Reztopia BackOffice. |
| Postcondition | Admin berhasil melakukan logout dan kembali ke halaman login. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi “Logout” |  |
|  | 1. Melakukan proses logout. |
|  | 1. Membersihkan sesi admin |
|  | 1. Menampilkan halaman login |
| Extension | - |
| Frequency of use | Admin akan menggunakan use case ini setiap kali ingin melakukan logout dari aplikasi Reztopia BackOffice. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.12 Use Case Scenario REQ-02.11

| ID | REQ-02.12 |
| --- | --- |
| Title | Melihat Dashboard Admin di Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang admin dapat melihat dashboard yang menyediakan informasi laporan penjualan pada pujasera, jumlah orderan, jumlah tenant yang aktif, jumlah tenant tidak aktif, dan daftar tenant terlaris. |
| Primary Actor | Admin |
| Precondition | Admin telah masuk ke dalam sistem Reztopia BackOffice. |
| Postcondition | Admin berhasil melihat dashboard dengan semua informasi yang tersedia. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi “Dashboard” |  |
|  | 1. Menampilkan dashboard |
|  | 1. Memperoleh data laporan penjualan pujasera |
|  | 1. Memperoleh data jumlah orderan |
|  | 1. Memperoleh data jumlah tenant aktif |
|  | 1. Memperoleh data jumlah tenant tidak aktif |
|  | 1. Memperoleh data daftar tenant terlaris |
| 1. Melihat laporan penjualan pujasera |  |
| 1. Melihat jumlah orderan |  |
| 1. Melihat jumlah tenant aktif |  |
| 1. Melihat jumlah tenant tidak aktif |  |
| 1. Melihat daftar tenant terlaris |  |
| Extension | - |
| Frequency of use | Proses ini akan dilakukan oleh admin setiap kali ingin melihat dashboard dengan informasi terkini tentang penjualan, orderan, dan status tenant. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.13 Use Case Scenario REQ-02.12

| ID | REQ-02.13 |
| --- | --- |
| Title | Mengelola Akun Tenant di Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang admin dapat mengelola akun tenant di aplikasi Reztopia BackOffice, termasuk mengatur data profil, menambahkan data tenant baru, serta mengatur status tenant sebagai aktif atau non-aktif. |
| Primary Actor | Admin |
| Precondition | Admin telah masuk ke dalam sistem Reztopia BackOffice. |
| Postcondition | Admin berhasil mengelola akun tenant sesuai dengan tindakan yang dilakukan. |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi “Data Profil” |  |
|  | 1. Menampilkan halaman data profil |
| 1. Memperbarui data profil |  |
| 1. Menyimpan data profil |  |
|  | 1. Memvalidasi data profil |
|  | 1. Jika data profil valid, maka :  * Memperbarui data profil * Menampilkan pesan sukses |
|  | 1. Jika data profil tidak valid, maka :  * Menampilkan pesan error * kembali ke halaman data profil |
| 1. Memilih opsi “Tambah Tenant” |  |
| 1. Memasukan data tenant baru |  |
| 1. Menyimpan data tenant baru |  |
|  | 1. Memvalidasi data tenant baru |
|  | 1. Jika data tenant baru valid, maka :  * Menambahkan tenant baru * Menampilkan pesan sukses |
|  | 1. Jika data tenant baru tidak valid, maka :  * Menampilkan pesan error * Kembali ke halaman data tenant |
| 1. Memilih tenant yang akan diatur |  |
| 1. Memilih opsi “Aktifkan” atau “Nonaktifkan” |  |
| 1. Konfirmasi Perubahan |  |
|  | 1. Memvalidasi konfirmasi |
|  | 1. Jika konfirmasi valid, maka :  * Mengatur status tenant * Menampilkan pesan sukses |
|  | 1. Jika konfirmasi tidak valid, maka :  * Menampilkan pesan error * Kembali ke halaman data tenant |
| Extension | - |
| Frequency of use | Proses ini akan dilakukan oleh admin ketika perlu mengatur dan mengelola akun tenant di Reztopia BackOffice. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.14 Use Case Scenario REQ-02.13

| ID | REQ-02.14 |
| --- | --- |
| Title | Melihat Laporan Penjualan Seluruh Tenant di Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang admin dapat melihat rincian penjualan harian atau periode tertentu dari setiap tenant di aplikasi Reztopia BackOffice, termasuk nama pemesan, nomor meja, dan total pembayaran. Admin juga dapat memilih tenant yang ingin dilihat laporan penjualannya. |
| Primary Actor | Admin |
| Precondition | Admin telah masuk ke dalam sistem Reztopia BackOffice. |
| Postcondition | Admin berhasil melihat rincian penjualan tenant. |
|  |  |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi "Laporan Penjualan" |  |
|  | 1. Menampilkan halaman laporan penjualan |
| 1. Memilih opsi "Rincian Harian" atau "Rincian Periode" |  |
| 1. Jika memilih "Rincian Harian":  * Memasukkan melihat laporan penjualan hari ini |  |
| 1. Jika memilih "Rincian Periode":  * Memasukkan rentang tanggal penjualan |  |
|  | 1. Menampilkan laporan penjualan berdasarkan pilihan |
| 1. Memilih opsi "Pilih Tenant" atau "Semua" |  |
| 1. Jika memilih "Pilih Tenant":  * Memilih tenant yang ingin dilihat laporan penjualannya |  |
| 1. Jika memilih "Semua":  * Tidak perlu memilih tenant |  |
|  | 1. Menampilkan laporan penjualan berdasarkan pilihan |
| Extension | - |
| Frequency of use | Proses ini dapat dilakukan oleh admin sesuai kebutuhan untuk melihat laporan penjualan dari seluruh tenant. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

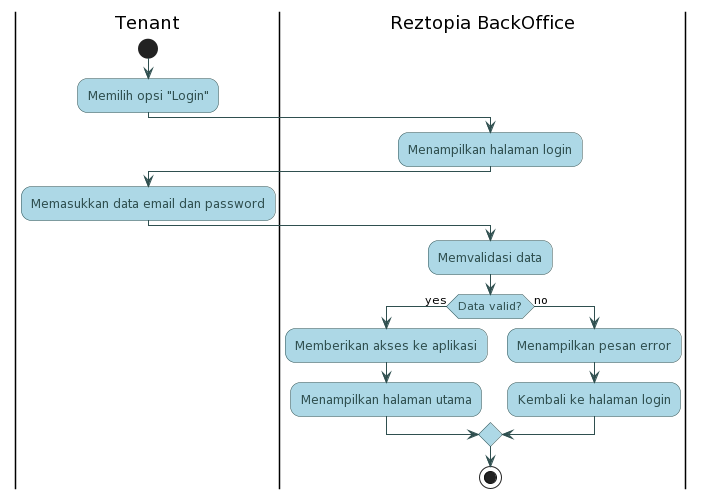
### Tabel IV.15 Use Case Scenario REQ-02.14

| ID | REQ-02.15 |
| --- | --- |
| Title | Mengelola Profil Admin di Reztopia BackOffice |
| Description | Use case ini menjelaskan bagaimana seorang admin dapat mengelola profilnya di aplikasi Reztopia BackOffice, termasuk mengatur isi profil, mengubah foto profil, dan mengubah kata sandi. |
| Primary Actor | Admin |
| Precondition | Admin telah masuk ke dalam sistem Reztopia BackOffice. |
| Postcondition | Admin berhasil mengatur dan memperbarui informasi profil seperti isi profil, foto profil, dan kata sandi. |
|  |  |
| Main Success Scenario | |
| Actor | Respon Sistem |
| 1. Memilih opsi “Profil" |  |
|  | 1. Menampilkan halaman profil |
| 1. Memilih opsi “Edit Isi Profil” |  |
| 1. Memperbarui informasi profil dan menyimpan perubahan |  |
|  | 1. Memvalidasi perubahan |
|  | 1. Jika perubahan valid, maka :  * Memperbarui profil * Menampilkan pesan sukses |
|  | 1. Jika perubahan tidak valid, maka :  * Menampilkan pesan error * Kembali ke halaman profil |
| 1. Memilih opsi “Ubah Foto baru dan menyimpan foto” |  |
|  | 1. Memvalidasi foto |
|  | 1. Jika foto valid, maka :  * Mengubah foto profil * Menampilkan pesan sukses |
|  | 1. Jika foto tidak valid, maka :  * Menampilkan pesan error * Kembali ke halaman profil |
| 1. Memilih opsi “Ubah Kata Sandi” |  |
| 1. Memasukan kata sandi lama dan baru dan menyimpan perubahan kata sandi |  |
|  | 1. Memvalidasi kata sandi |
|  | 1. Jika kata sandi valid, maka :  * Mengubah kata sandi * menampilkan pesan sukses |
|  | 1. Jika kata sandi tidak valid, maka :  * Menampilkan pesan error * Kembali ke halaman profil |
| Extension | - |
| Frequency of use | Admin dapat mengelola profilnya sesuai kebutuhan, seperti mengubah isi profil, foto profil, dan kata sandi. |
| Status | Disetujui |
| Owner | Tim TA Reztopia |
| Priority | Tinggi |

### Tabel IV.16 Use Case Scenario REQ-02.15

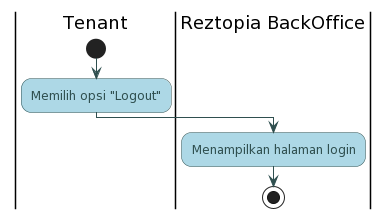
## **Activity Diagram**

Untuk Tenant :

1. Melakukan Login Tenant

### Gambar IV.03 Activity Diagram Melakukan Login Tenant

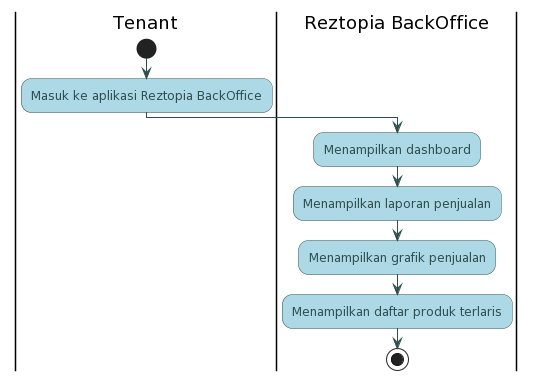
Diagram ini secara umum menggambarkan aktivitas login yang dilakukan oleh seorang tenant di aplikasi "Reztopia BackOffice". Tenant memulai dengan memilih opsi login, memasukkan data login, dan aplikasi akan memvalidasi data tersebut. Jika data valid, tenant diberikan akses ke aplikasi dan dapat mengakses fungsionalitasnya. Jika data tidak valid, tenant akan mendapatkan pesan error dan diarahkan kembali ke halaman login.

1. Melakukan Logout Tenant

### Gambar IV.04 Activity Diagram Melakukan Logout Tenant

Diagram ini secara umum menggambarkan aktivitas logout yang dilakukan oleh seorang tenant di aplikasi "Reztopia BackOffice". Aktor "Tenant" akan melakukan logout untuk menjaga keamanan akun setelah selesai menggunakan aplikasi tersebut.

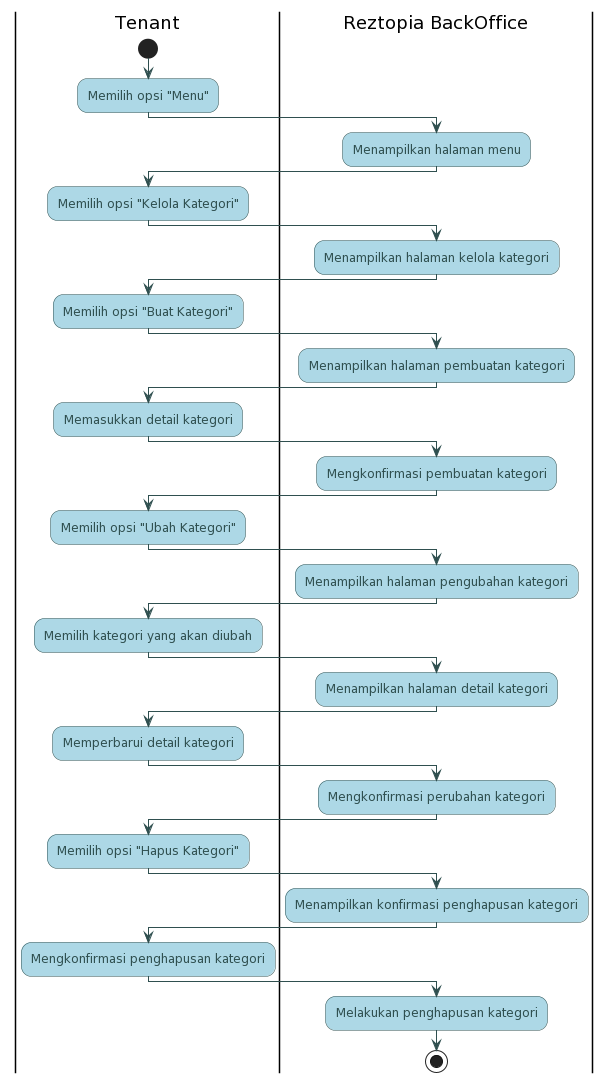
1. Melihat Dashboard Tenant



### Gambar IV.05 Activity Diagram Melihat Dashboard Tenant

Diagram ini secara umum menggambarkan aktivitas "Melihat Dashboard" yang dilakukan oleh seorang tenant di aplikasi "Reztopia BackOffice". Tenant dapat melihat dashboard yang menampilkan laporan penjualan dalam bentuk grafik serta daftar produk terlaris untuk memantau kinerja bisnis mereka.

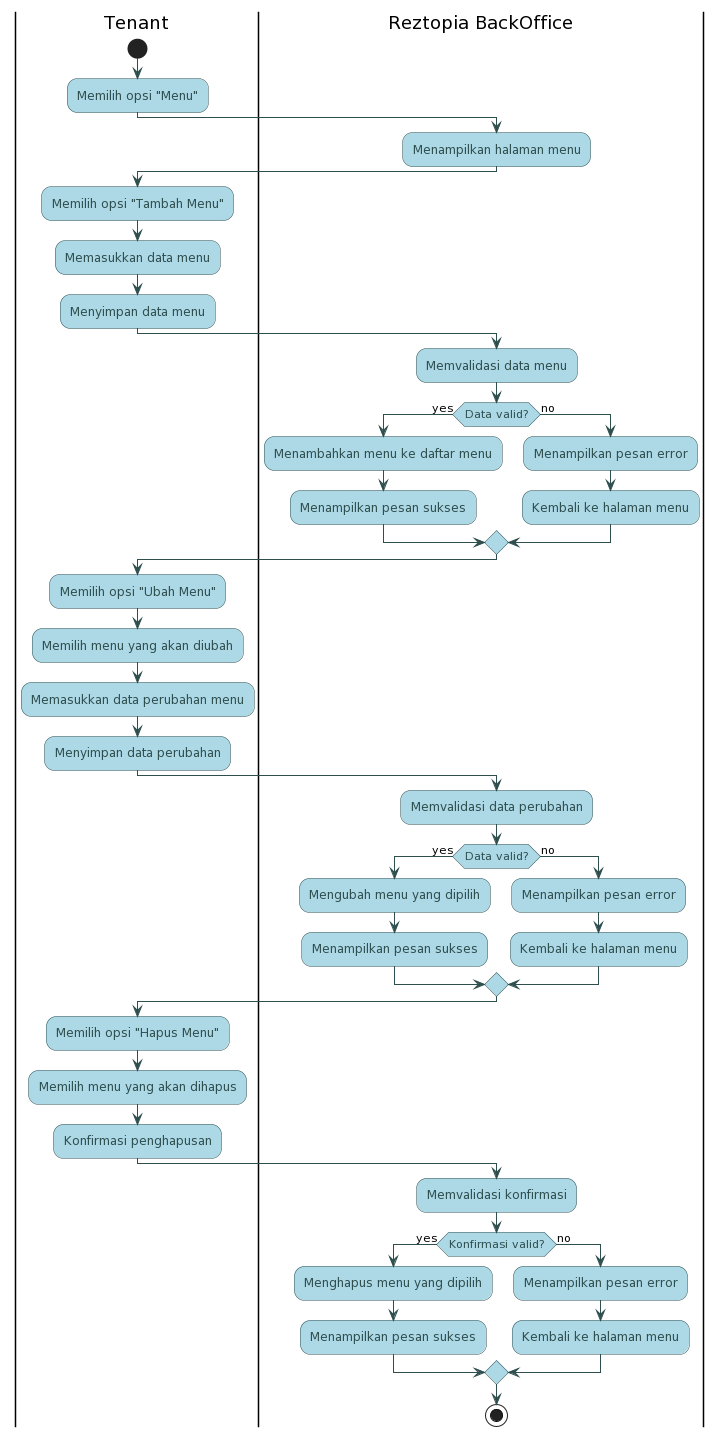
1. Mengelola Kategori



### Gambar IV.06 Activity Diagram Mengelola Kategori

Diagram ini secara umum menggambarkan aktivitas yang dilakukan oleh seorang tenant di aplikasi "Reztopia BackOffice" dalam mengelola kategori pada halaman menu. Tenant dapat membuat, mengubah, dan menghapus kategori sesuai kebutuhan mereka dalam mengatur menu aplikasi.

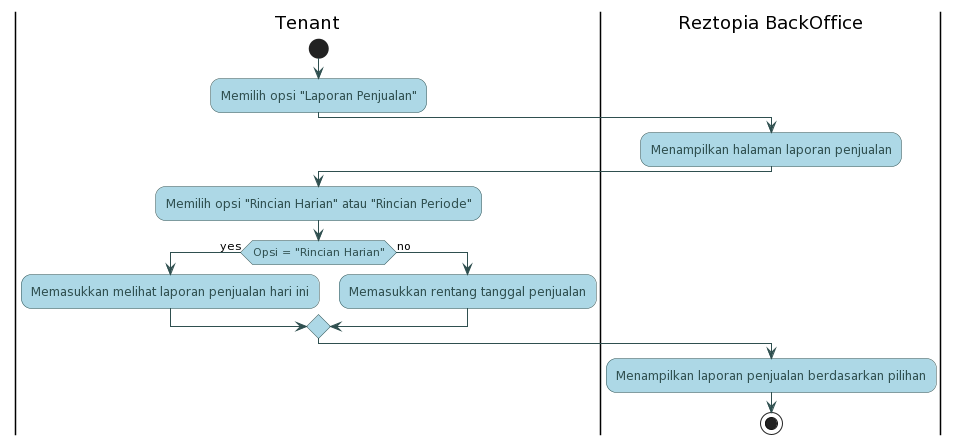
1. Mengelola Menu



### Gambar IV.07 Activity Diagram Mengelola Menu

Diagram ini secara umum menggambarkan aktivitas yang dilakukan oleh seorang tenant di aplikasi "Reztopia BackOffice" terkait dengan epic "Menu". Tenant memulai dengan memilih opsi "Menu" di aplikasi, yang kemudian menampilkan halaman menu. Tenant dapat melakukan tiga tindakan utama: "Tambah Menu", "Ubah Menu", dan "Hapus Menu". Setiap tindakan melibatkan penginputan data, validasi data, dan tindakan yang sesuai, diikuti dengan pesan sukses atau pesan error tergantung pada validasi.

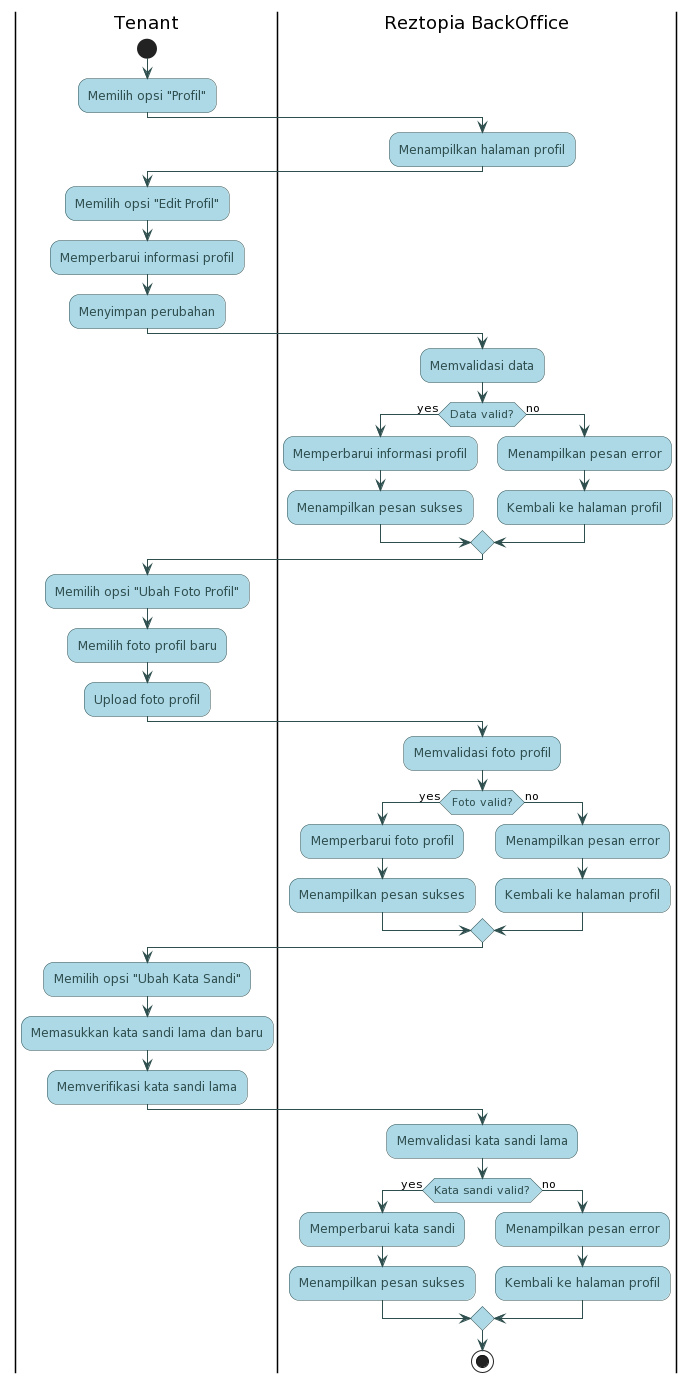
1. Melihat Laporan Penjualan Tenant



### Gambar IV.08 Activity Diagram Melihat Laporan Penjualan Tenant

Diagram aktivitas di atas menggambarkan alur yang terjadi ketika seorang penyewa (Tenant) memilih opsi "Laporan Penjualan" pada aplikasi Reztopia BackOffice. Tenant kemudian dapat memilih antara opsi "Rincian Harian" untuk melihat laporan penjualan pada hari ini atau opsi "Rincian Periode" untuk memasukkan rentang tanggal penjualan. Setelah memilih, Reztopia BackOffice akan menampilkan laporan penjualan sesuai pilihan Tenant.

1. Mengelola Profil Tenant

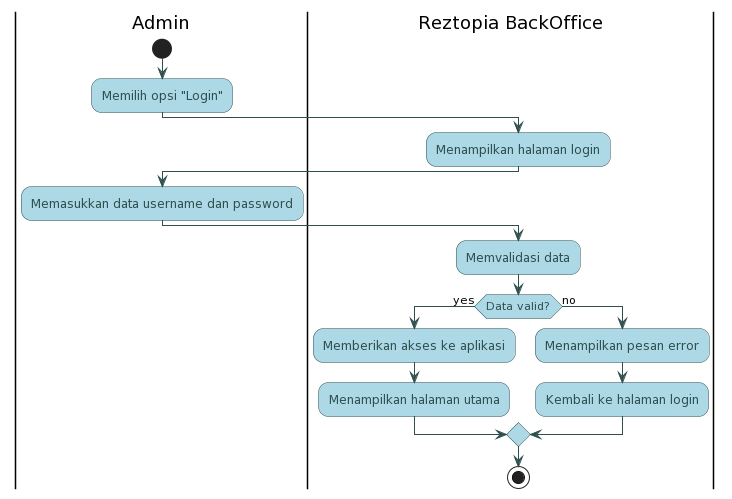


### Gambar IV.09 Activity Diagram Mengelola Profil Tenant

Diagram ini secara umum menggambarkan aktivitas yang dilakukan oleh seorang tenant di aplikasi "Reztopia BackOffice" terkait dengan epic "Profil". Tenant memulai dengan memilih opsi "Profil" di aplikasi, yang kemudian menampilkan halaman profil. Tenant dapat melakukan tiga tindakan utama: "Edit Profil" untuk memperbarui informasi profil, "Ubah Foto Profil" untuk mengganti foto profil, dan "Ubah Kata Sandi" untuk mengubah kata sandi. Setiap tindakan melibatkan penginputan data, validasi data, dan tindakan yang sesuai, diikuti dengan pesan sukses atau pesan error tergantung pada validasi.

Untuk Admin :

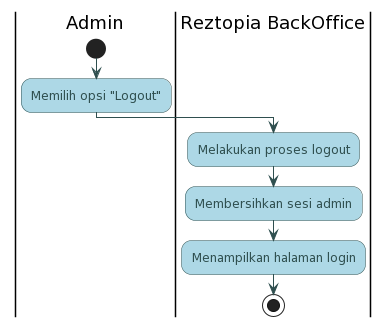
1. Melakukan Login Admin



### Gambar IV.10 Activity Diagram Melakukan Login Admin

Diagram ini secara umum menggambarkan aktivitas yang dilakukan oleh seorang admin di aplikasi "Reztopia BackOffice" terkait dengan epic "Login". Admin memulai dengan memilih opsi "Login" di aplikasi, yang kemudian menampilkan halaman login. Admin memasukkan data username dan password, dan aplikasi "Reztopia BackOffice" akan memvalidasi data tersebut. Jika data valid, admin akan diberikan akses ke dalam aplikasi dan melihat halaman utama. Namun, jika data tidak valid, aplikasi akan menampilkan pesan error dan mengarahkan admin kembali ke halaman login. Proses berakhir ketika login selesai (stop).

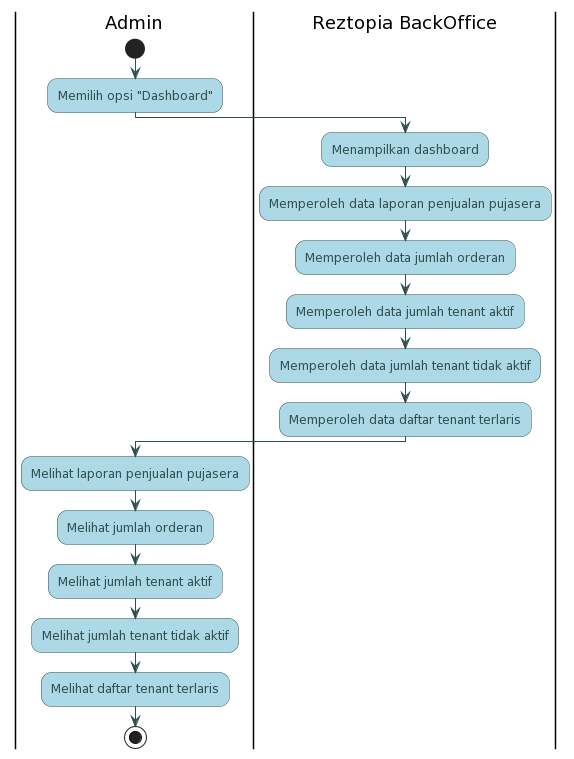
1. Melakukan Logout Admin



### Gambar IV.11 Activity Diagram Melakukan Logout Admin

Diagram ini secara umum menggambarkan aktivitas yang dilakukan oleh seorang admin di aplikasi "Reztopia BackOffice" terkait dengan epic "Logout". Admin memulai dengan memilih opsi "Logout" di aplikasi, yang kemudian melakukan proses logout. Selama proses logout, sesi admin dibersihkan dan halaman login ditampilkan. Proses berakhir ketika logout selesai (stop).

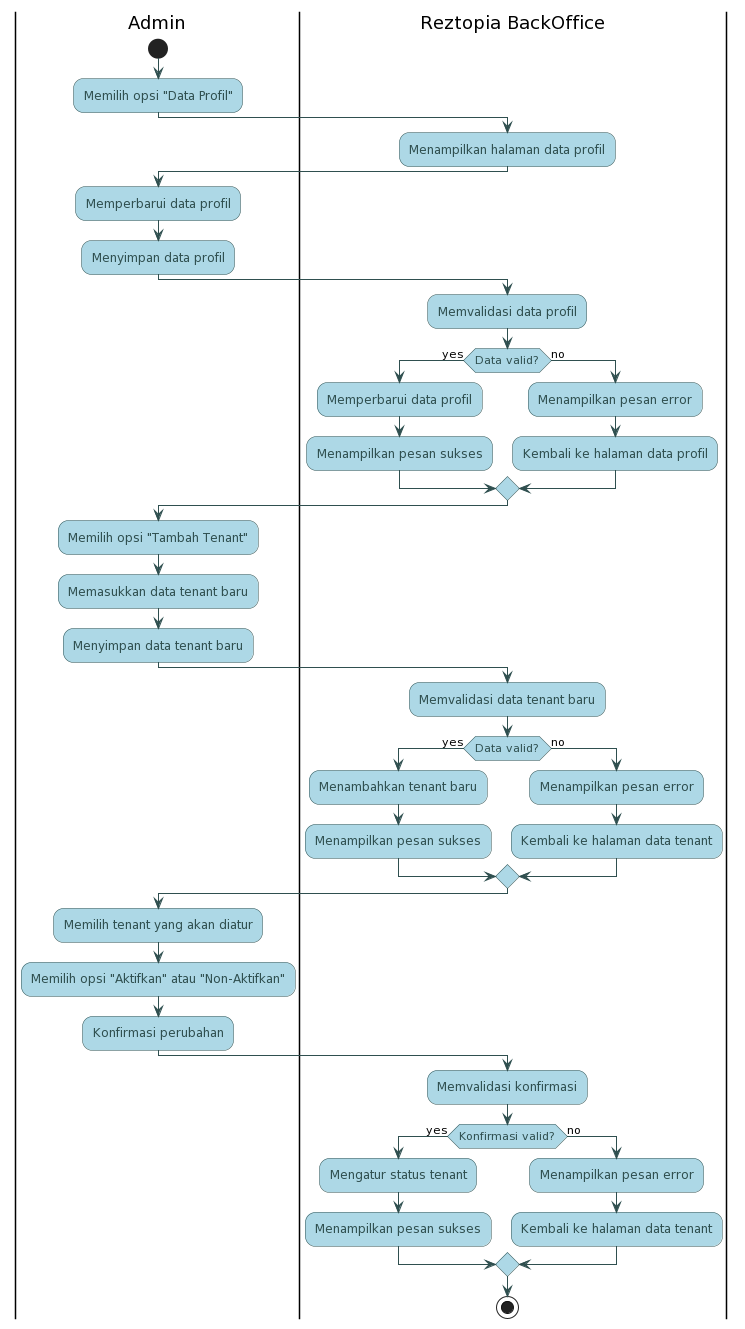
1. Melihat Dashboard Admin



### Gambar IV.12 Activity Diagram Melihat Dashboard Admin

Diagram ini secara umum menggambarkan aktivitas yang dilakukan oleh seorang admin di aplikasi "Reztopia BackOffice" terkait dengan epic "Dashboard". Admin memulai dengan memilih opsi "Dashboard" di aplikasi, yang kemudian menampilkan halaman dashboard. Admin dapat melihat beberapa informasi, termasuk laporan penjualan pujasera, jumlah orderan, jumlah tenant aktif, jumlah tenant tidak aktif, dan daftar tenant terlaris. Informasi tersebut diperoleh dari aplikasi "Reztopia BackOffice" dan ditampilkan kepada admin untuk memberikan pemahaman tentang performa pujasera dan aktivitas tenant secara keseluruhan.

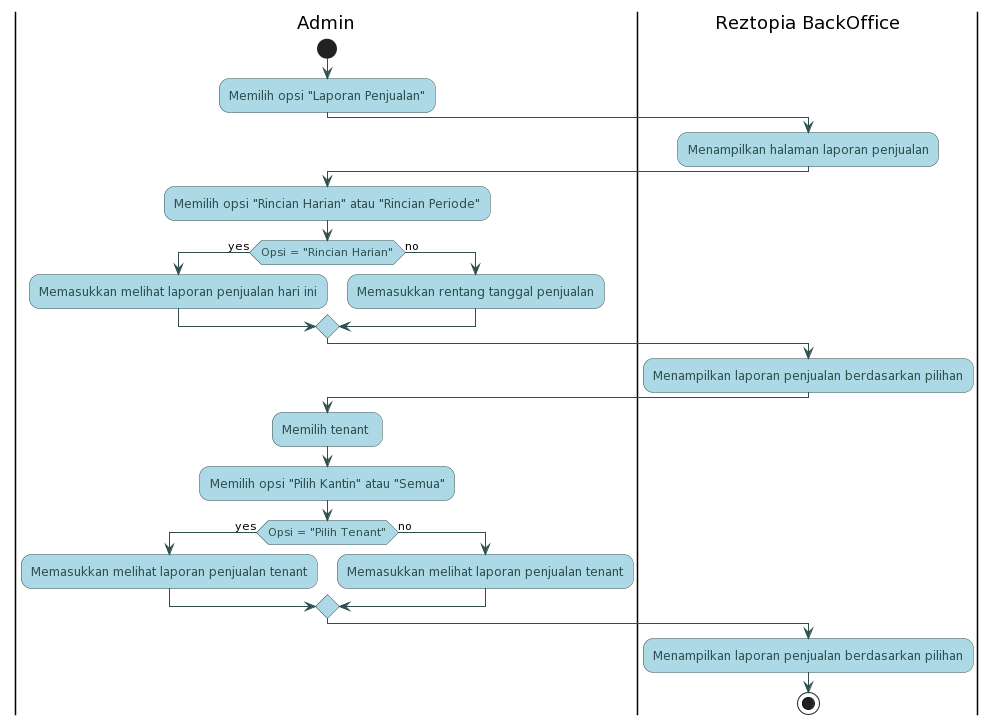
1. Mengelola Akun Tenant



### Gambar IV.13 Activity Diagram Mengelola Akun Tenant

Diagram ini secara umum menggambarkan aktivitas yang dilakukan oleh seorang admin di aplikasi "Reztopia BackOffice" terkait dengan epic "Tenant" dan story "Mengelola Akun Tenant". Admin dapat melakukan beberapa tindakan, yaitu "Memperbarui Data Profil", "Tambah Tenant", dan "Mengatur Tenant untuk Aktif atau Nonaktif". Setiap tindakan melibatkan penginputan data, validasi data, dan tindakan yang sesuai, diikuti dengan pesan sukses atau pesan error tergantung pada validasi. Diagram ini membantu memvisualisasikan alur aktivitas yang harus dilakukan oleh admin dalam mengelola akun tenant pada aplikasi "Reztopia BackOffice".

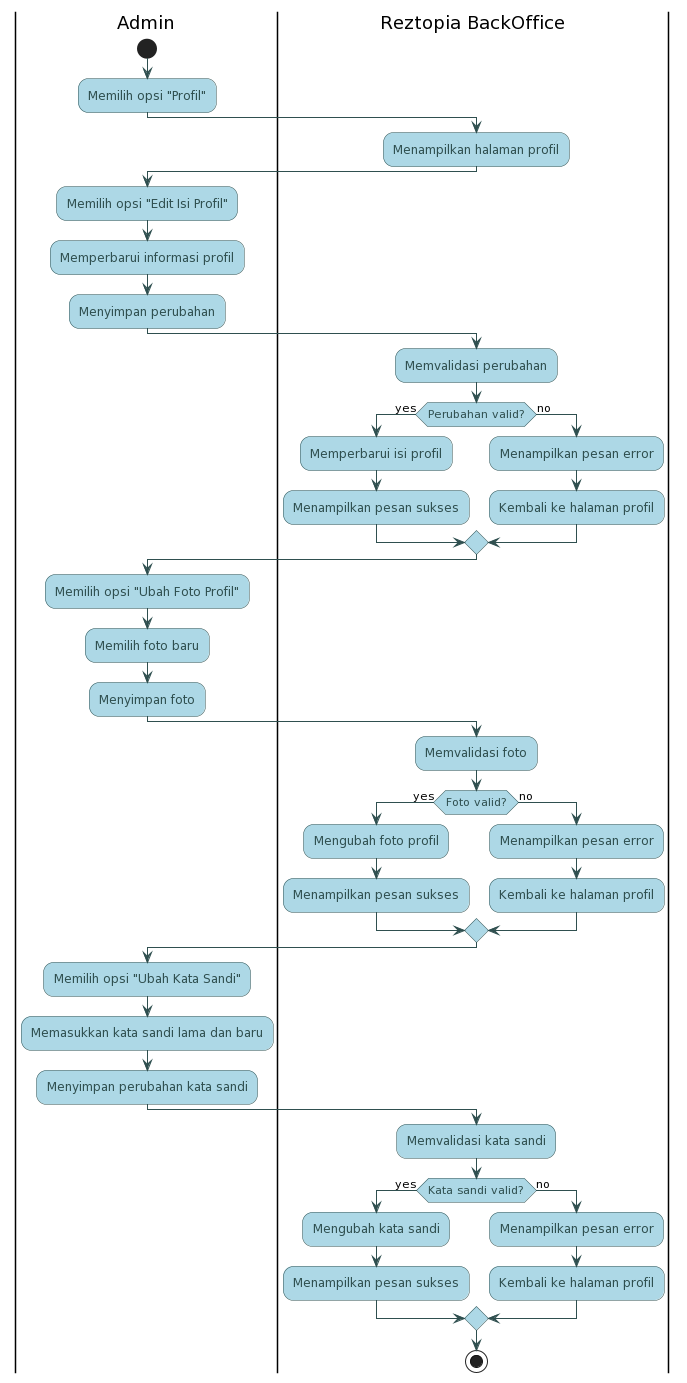
1. Melihat Laporan Penjualan Seluruh Tenant



### Gambar IV.14 Activity Diagram Melihat Laporan Penjualan Seluruh Tenant

Diagram aktivitas tersebut menggambarkan alur aktivitas yang terjadi ketika seorang "Admin" menggunakan aplikasi "Reztopia BackOffice" untuk melihat laporan penjualan. Admin dapat memilih opsi "Rincian Harian" atau "Rincian Periode" untuk melihat laporan penjualan berdasarkan tanggal, serta memilih tenant atau semua tenant untuk melihat laporan penjualan berdasarkan tenant. Diagram ini membantu memvisualisasikan langkah-langkah yang harus diikuti oleh Admin dalam melihat laporan penjualan secara jelas dan intuitif.

1. Mengelola Profil Admin



### Gambar IV.15 Activity Diagram Mengelola Profil Admin

Diagram ini secara umum menggambarkan aktivitas yang dilakukan oleh seorang admin di aplikasi "Reztopia BackOffice" terkait dengan epic "Profil". Admin memulai dengan memilih opsi "Profil" di aplikasi, yang kemudian menampilkan halaman profil. Admin dapat melakukan tiga tindakan utama: "Edit Isi Profil", "Ubah Foto Profil", dan "Ubah Kata Sandi". Setiap tindakan melibatkan penginputan data, validasi data, dan tindakan yang sesuai, diikuti dengan pesan sukses atau pesan error tergantung pada validasi.

## **Class Diagram**

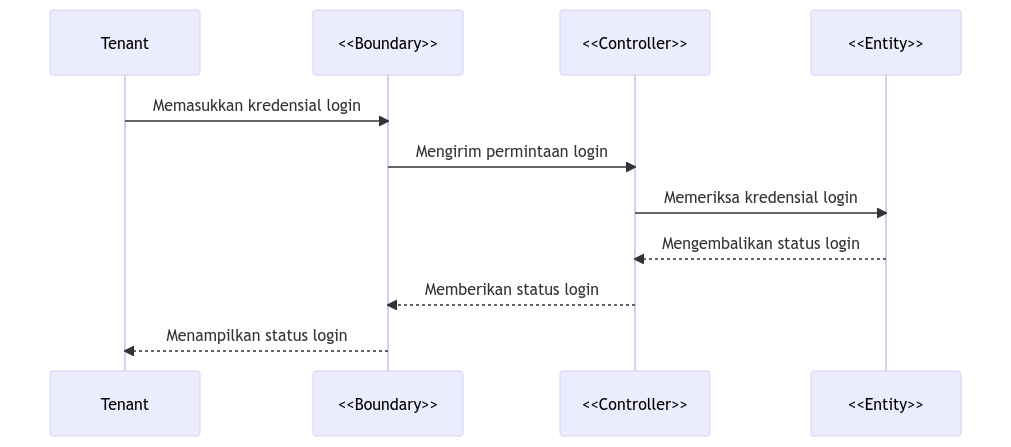
### Gambar IV.16 Class Diagram Reztopia Fitur Back Office

Diagram tersebut menggambarkan struktur hubungan antara kelas-kelas yang terlibat dalam sistem. Diagram ini mencakup beberapa kelas yang terkait dengan login, logout, dashboard, menu, laporan penjualan, dan profil. Terdapat dua aktor utama, yaitu Tenant dan Admin, yang memiliki akses ke fitur-fitur yang berbeda. Tenant dapat melakukan login, logout, melihat dashboard, mengelola kategori dan menu, melihat laporan penjualan, serta mengelola profil. Sementara itu, Admin memiliki fungsi yang lebih luas, termasuk login, logout, melihat dashboard admin, mengelola akun tenant, melihat laporan penjualan seluruh tenant, dan mengelola profil admin. Diagram ini memberikan gambaran umum mengenai struktur kelas dan hubungan antara mereka dalam sistem tersebut.

## **Sequence Diagram**

Untuk Tenant :

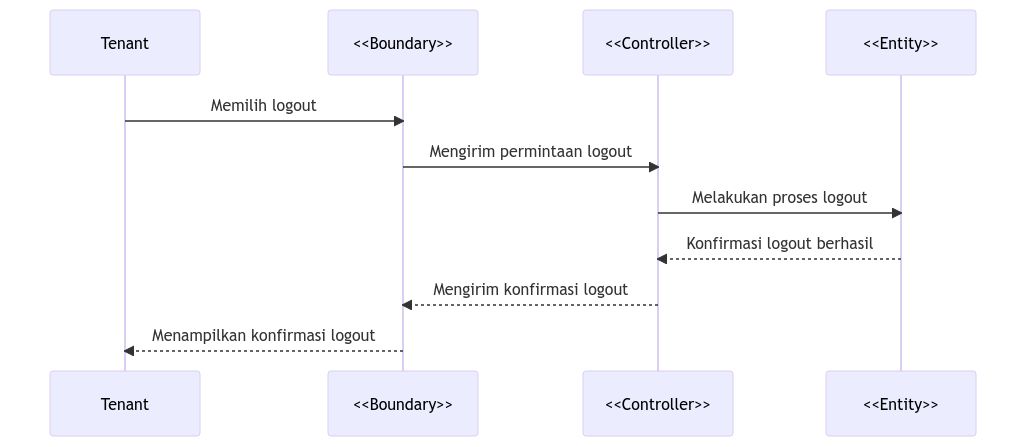
1. Melakukan Login Tenant



### Gambar IV.17 Sequence Diagram Melakukan Login Tenant

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Tenant memasukkan kredensial login ke dalam aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan login ke Controller. Controller berkomunikasi dengan Entity untuk memeriksa kredensial login, yang kemudian mengembalikan status login kepada Controller. Status login diberikan oleh Controller kepada Boundary dan ditampilkan kepada Tenant.

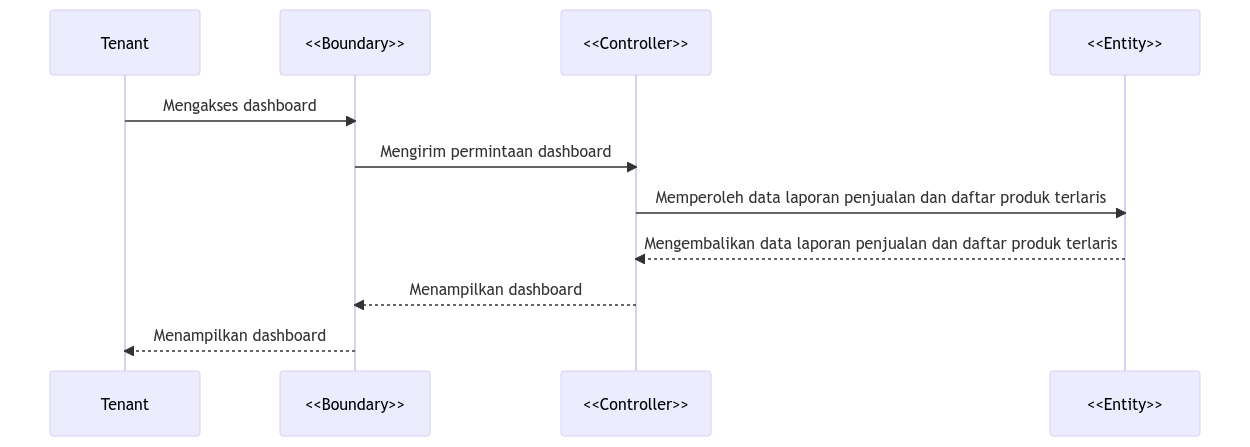
1. Melakukan Logout Tenant



### Gambar IV.18 Sequence Diagram Melakukan Logout Tenant

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Tenant memilih logout di aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan logout ke Controller. Controller berkomunikasi dengan Entity untuk melakukan proses logout, dan setelah berhasil, konfirmasi logout dikirimkan oleh Controller kepada Boundary. Konfirmasi logout kemudian ditampilkan oleh Boundary kepada Tenant.

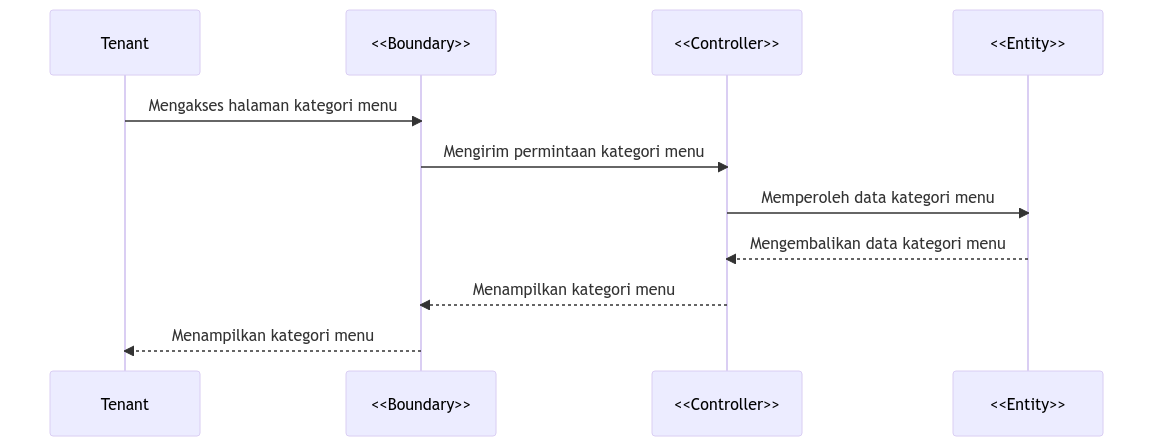
1. Melihat Dashboard Tenant



### Gambar IV.19 Sequence Diagram Melihat Dashboard Tenant

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Tenant mengakses dashboard di aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan dashboard ke Controller. Controller berkomunikasi dengan Entity untuk memperoleh data laporan penjualan dan daftar produk terlaris, yang kemudian ditampilkan oleh Controller kepada Boundary. Dashboard ditampilkan oleh Boundary kepada Tenant, termasuk fitur laporan penjualan dalam bentuk grafik dan daftar produk terlaris.

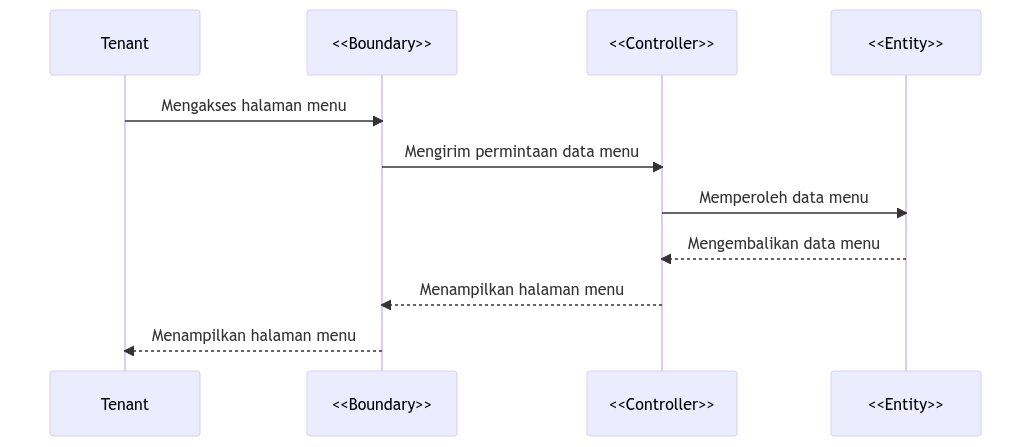
1. Mengelola Kategori



### Gambar IV.20 Sequence Diagram Mengelola Kategori

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Tenant mengakses halaman kategori menu pada aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan kategori menu ke Controller. Controller berkomunikasi dengan Entity untuk memperoleh data kategori menu, yang kemudian ditampilkan oleh Controller kepada Boundary. Kategori menu ditampilkan oleh Boundary kepada Tenant.

1. Mengelola Menu



### Gambar IV.21 Sequence Diagram Mengelola Menu

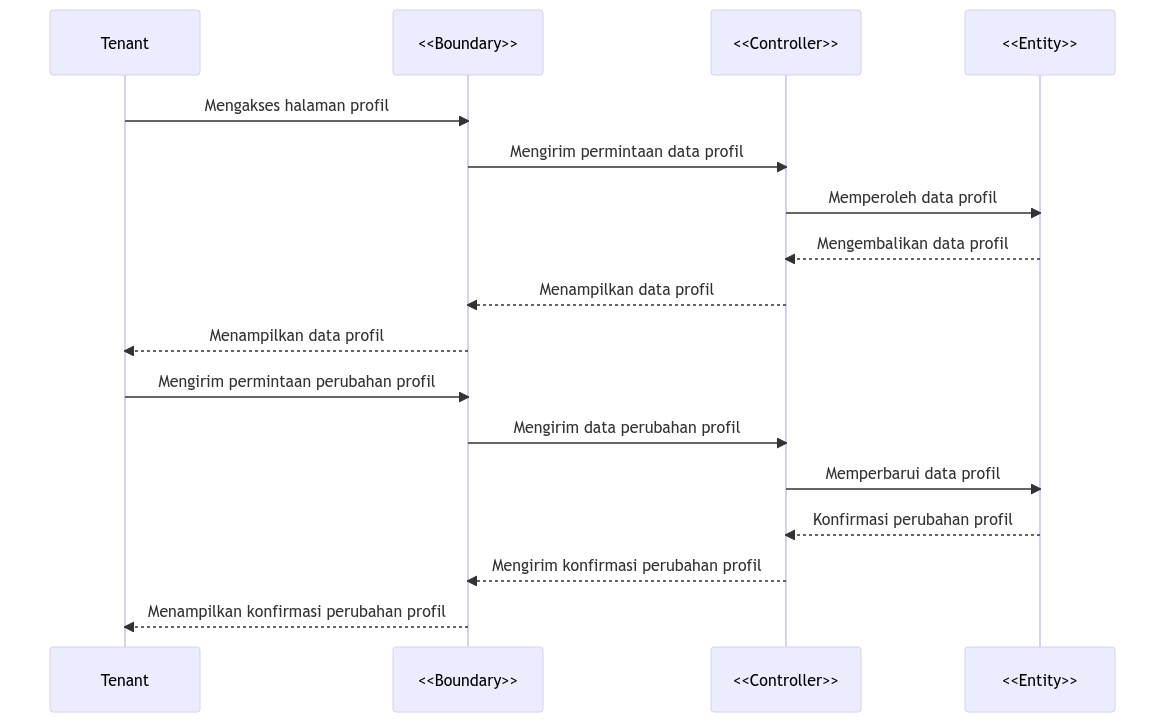
Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Tenant mengakses halaman menu pada aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan data menu ke Controller. Controller berkomunikasi dengan Entity untuk memperoleh data menu, yang kemudian ditampilkan oleh Controller kepada Boundary. Halaman menu ditampilkan oleh Boundary kepada Tenant.

1. Melihat Laporan Penjualan Tenant

### Gambar IV.22 Sequence Diagram Laporan Penjualan Tenant

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Tenant memilih laporan penjualan di aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan laporan penjualan ke Controller. Controller berkomunikasi dengan Entity untuk memperoleh data penjualan, yang kemudian ditampilkan oleh Controller kepada Boundary. Laporan penjualan ditampilkan oleh Boundary kepada Tenant, termasuk rincian seperti nama pemesan, nomor meja, dan total pembayaran.

1. Mengelola Profil Tenant

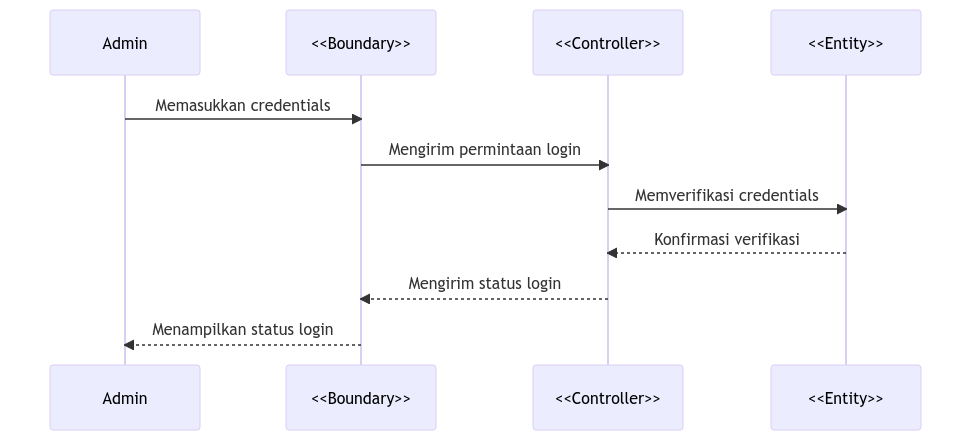


### Gambar IV.23 Sequence Diagram Mengelola Profil Tenant

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Tenant mengakses halaman profil pada aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan data profil ke Controller, yang berkomunikasi dengan Entity untuk memperoleh data profil. Data profil ditampilkan oleh Controller kepada Boundary, yang kemudian ditampilkan kepada Tenant. Tenant juga dapat mengirim permintaan perubahan profil melalui Boundary. Data perubahan profil dikirim oleh Boundary ke Controller, yang berkomunikasi dengan Entity untuk memperbarui data profil. Setelah perubahan profil berhasil, konfirmasi perubahan dikirimkan oleh Controller kepada Boundary, yang kemudian ditampilkan kepada Tenant.

Untuk Admin :

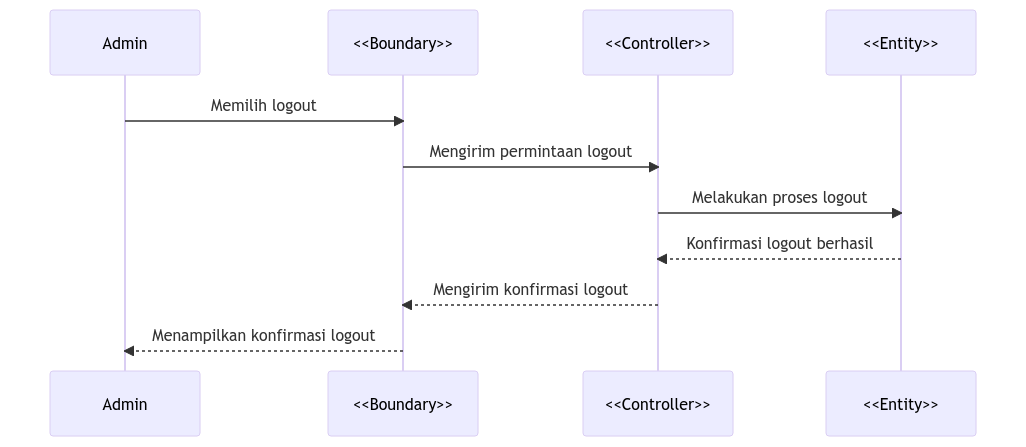
1. Melakukan Login Admin



### Gambar IV.24 Sequence Diagram Melakukan Login Admin

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Admin memasukkan credentials untuk login ke aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan login ke Controller. Controller berkomunikasi dengan Entity untuk memverifikasi credentials, dan setelah verifikasi berhasil, konfirmasi verifikasi dikirimkan oleh Entity kepada Controller. Status login dikirimkan oleh Controller kepada Boundary, yang kemudian ditampilkan oleh Boundary kepada Admin.

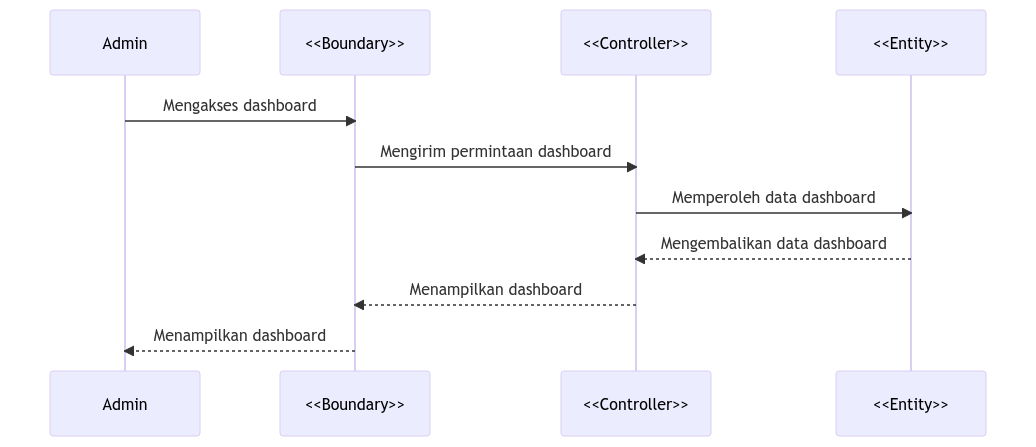
1. Melakukan Logout Admin



### Gambar IV.25 Sequence Diagram Melakukan Logout Admin

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Admin memilih logout di aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan logout ke Controller. Controller berkomunikasi dengan Entity untuk melakukan proses logout, dan setelah berhasil, konfirmasi logout dikirimkan oleh Controller kepada Boundary. Konfirmasi logout kemudian ditampilkan oleh Boundary kepada Admin.

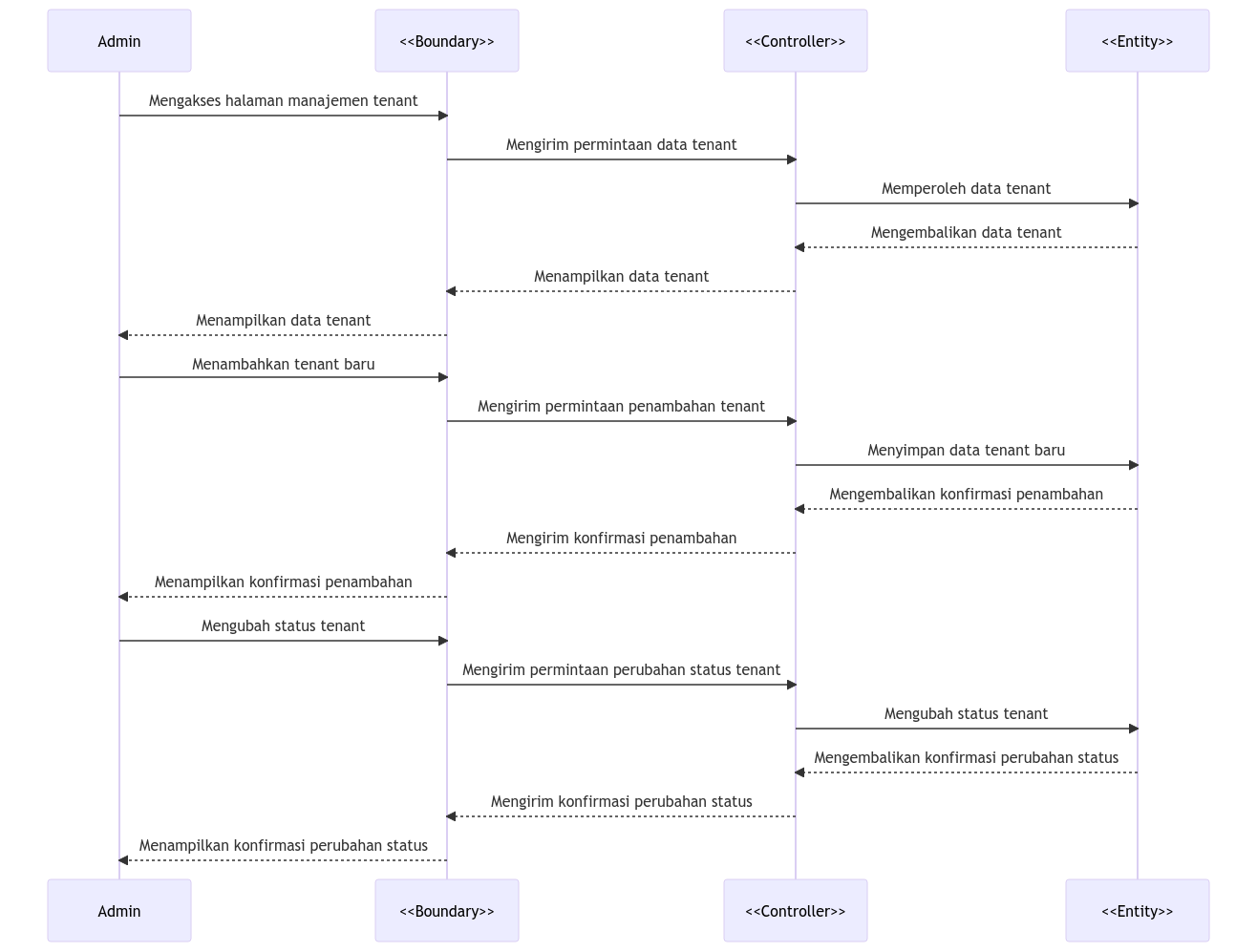
1. Melihat Dashboard Admin



### Gambar IV.26 Sequence Diagram Melihat Dashboard Admin

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Admin mengakses dashboard aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan dashboard ke Controller. Controller berkomunikasi dengan Entity untuk memperoleh data dashboard, yang kemudian ditampilkan oleh Controller kepada Boundary. Dashboard ditampilkan oleh Boundary kepada Admin.

1. Mengelola Akun Tenant



### Gambar IV.27 Sequence Diagram Mengelola Akun Tenant

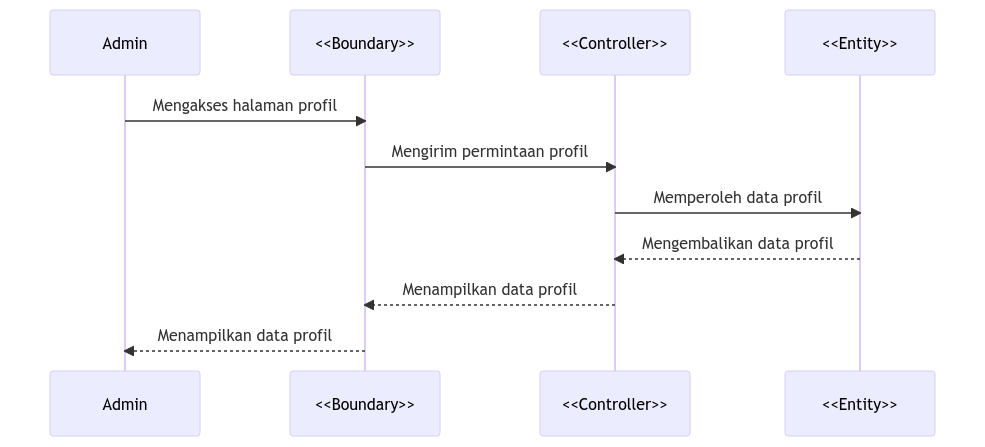
Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Admin mengakses halaman manajemen tenant di aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan data tenant ke Controller. Controller berkomunikasi dengan Entity untuk memperoleh data tenant, yang kemudian ditampilkan oleh Controller kepada Boundary. Data tenant ditampilkan oleh Boundary kepada Admin. Admin dapat menambahkan tenant baru dengan mengirimkan permintaan penambahan melalui Boundary. Controller menyimpan data tenant baru yang diterima dari Entity, dan konfirmasi penambahan dikirimkan kembali oleh Controller kepada Boundary. Konfirmasi penambahan ditampilkan oleh Boundary kepada Admin. Admin juga dapat mengubah status tenant dengan mengirimkan permintaan perubahan status melalui Boundary. Controller mengubah status tenant di Entity, dan konfirmasi perubahan status dikirimkan kembali oleh Controller kepada Boundary. Konfirmasi perubahan status ditampilkan oleh Boundary kepada Admin.

1. Melihat Laporan Penjualan Seluruh Tenant

### Gambar IV.28 Sequence Diagram Melihat Laporan Penjualan Seluruh Tenant

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Admin memilih laporan penjualan pada aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan laporan penjualan ke Controller. Controller berkomunikasi dengan Entity untuk mengambil data penjualan, yang kemudian ditampilkan oleh Controller kepada Boundary. Laporan penjualan ditampilkan oleh Boundary kepada Admin.

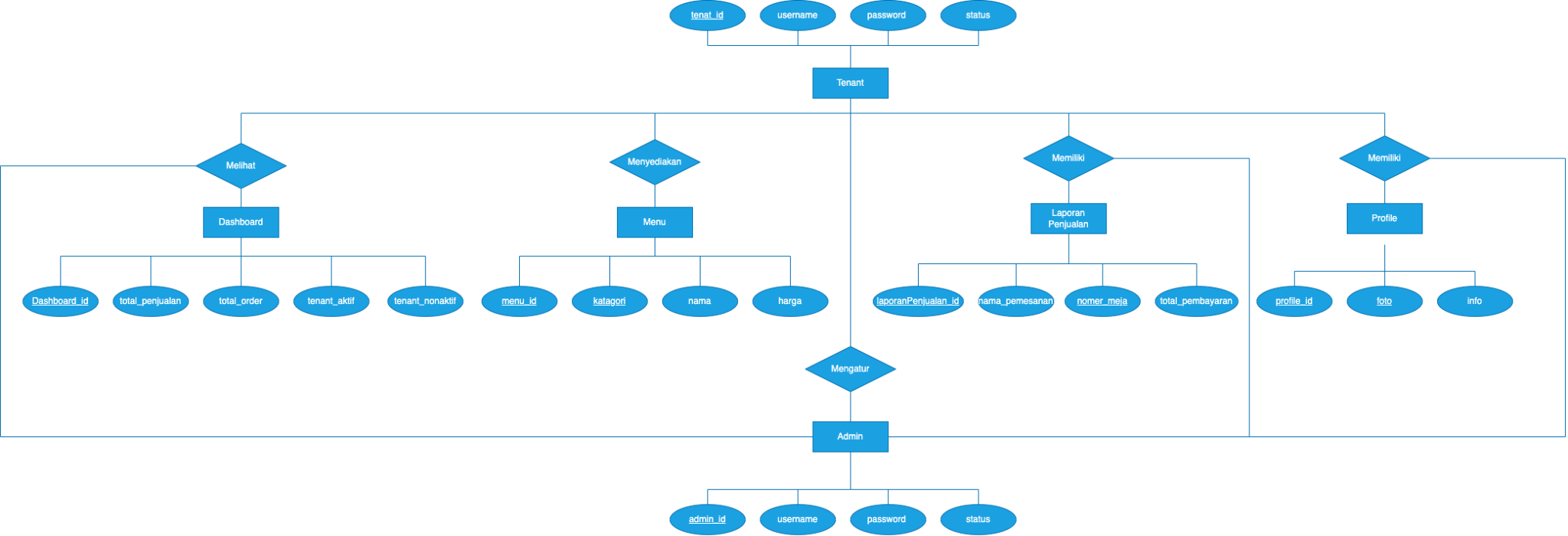
1. Mengelola Profil Admin



### Gambar IV.29 Sequence Diagram Mengelola Profil Admin

Gambaran umum singkat mengenai diagram ini adalah sebagai berikut: Admin mengakses halaman profil di aplikasi Reztopia BackOffice melalui Boundary. Boundary mengirim permintaan profil ke Controller. Controller berkomunikasi dengan Entity untuk memperoleh data profil, yang kemudian ditampilkan oleh Controller kepada Boundary. Data profil ditampilkan oleh Boundary kepada Admin.

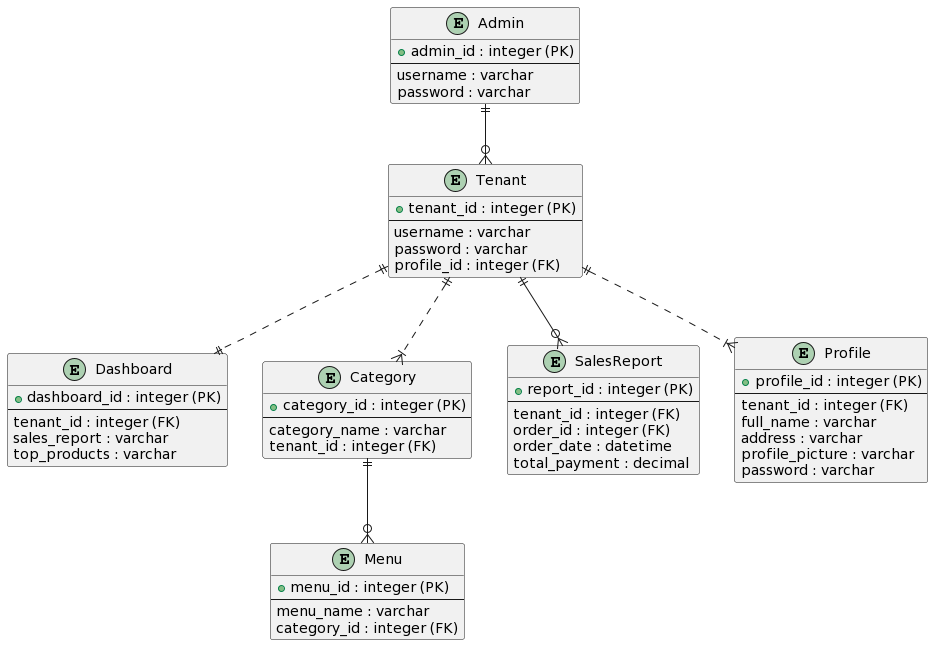
## **Entitas Relasional Diagram**



### Gambar IV.30 Entity Relationship Diagram Fitur Back Office

Diagram entitas relasional ini menggambarkan struktur data untuk aplikasi Reztopia BackOffice. Terdapat lima entitas utama: Tenant, Menu, Laporan Penjualan, Profil, dan Admin. Entitas Tenant dan Admin memiliki atribut seperti id, username, password, dan status. Sementara itu, entitas Menu dan Laporan Penjualan terhubung dengan Tenant melalui foreign key, menunjukkan bahwa setiap Tenant dapat memiliki banyak Menu dan Laporan Penjualan. Entitas Profil terkait dengan Tenant dan Admin, yang menandakan bahwa setiap Tenant dan Admin memiliki satu Profil. Entitas Dashboard terkait dengan Admin, yang berarti setiap Admin dapat melihat satu Dashboard. Diagram ini mendukung banyak cerita pengguna yang disebutkan dalam spesifikasi, termasuk pengelolaan menu oleh Tenant, melihat laporan penjualan, dan mengelola profil oleh Tenant dan Admin.

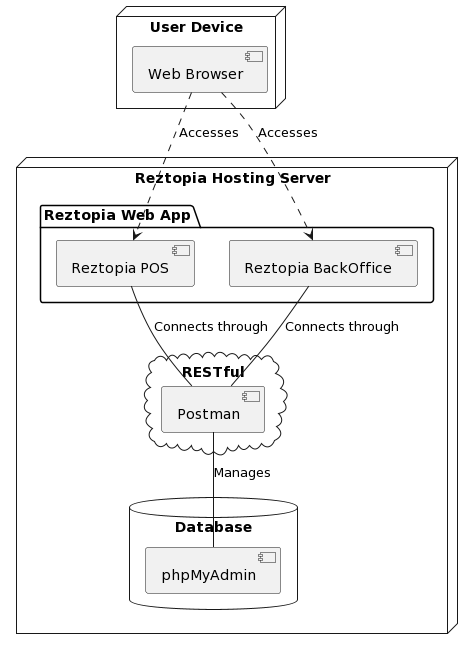
## **Relational Table**



### Gambar IV.31 Relational Table Diagram Fitur Back Office

Diagram entitas relasional yang diberikan menggambarkan struktur data dan hubungan antara entitas dalam aplikasi Reztopia BackOffice. Terdapat beberapa entitas utama, seperti Tenant, Dashboard, Category, Menu, SalesReport, Profile, dan Admin. Setiap entitas memiliki atribut-atribut yang relevan. Hubungan antara entitas ditunjukkan melalui kardinalitas, misalnya, Tenant memiliki hubungan "has" dengan Dashboard, "manages" dengan Category, "generates" dengan SalesReport, dan "has" dengan Profile. Diagram ini memberikan pandangan umum tentang bagaimana data terorganisir dan berinteraksi dalam aplikasi Reztopia BackOffice.

## **Deployment Diagram**



### Gambar IV.32 Deployment Diagram Fitur Back Office

Diagram di atas menggambarkan cara di mana sistem aplikasi web Reztopia dihosting dan diakses. Pengguna dapat mengakses dua aplikasi berbeda, yaitu Reztopia POS dan Reztopia BackOffice, melalui peramban web pada perangkat mereka. Kedua aplikasi ini di hosting di server Reztopia dan berkomunikasi melalui RESTful menggunakan Postman untuk menguji dan mengelola layanan API. Semua data yang dihasilkan dan dikelola oleh aplikasi disimpan dalam sebuah database yang dioperasikan melalui phpMyAdmin. Ini menciptakan ekosistem terintegrasi dimana pengguna dapat mengakses dan berinteraksi dengan aplikasi melalui web, sementara semua data dan fungsionalitas dikelola dan di hosting pada server Reztopia.

## **Low Fidelity**

1. Reztopia BackOffice - Login
2. Reztopia BackOffice - Dashboard
3. Reztopia BackOffice - Menu
4. Reztopia BackOffice - Laporan Penjualan
5. Reztopia BackOffice - Profil
6. Reztopia BackOffice - Mengelola Akun Tenant

# DAFTAR PUSTAKA

Putri, A. D., & Wulandari, A. (2020). Analisis Kepuasan Mahasiswa Terhadap Kualitas Pelayanan Kantin Asrama Universitas Telkom dengan Metode Importance Performance Analysis (IPA). Jurnal Ilmu Manajemen Dan Akuntansi Terapan, 11(2), 173-186.

Rizqi, A., & Wibisono, Y. (2018). Perancangan Aplikasi Point of Sales di Restoran untuk Menunjang Proses Transaksi. NeO-BIS (Ners and Electronics Oriented Business Information System), 2(1), 60-67.

Susanto, A., & Utama, W. G. (2020). "Perancangan Aplikasi Point of Sale (POS) dengan Fitur Back Office pada Toko Pakaian." Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK), 4(2), 95-101.

Arifianto, A. (2021). Implementasi Metode Scrum pada Proses Pengembangan Perangkat Lunak. Jurnal Sistem Informasi Bisnis, 11(1), 39-45.

Ardianto, A., & Suryanto, R. (2021). Implementasi Keamanan Web dengan PHP. Jurnal Informatika Upgris, 9(2), 79-88.

Prasetyo, A., & Cahyono, E. (2020). Pengembangan Aplikasi Berbasis Web dengan Menggunakan Laravel Framework. Jurnal Informatika Upgris, 8(1), 44-55.

Ahmad Ramadhan. (2019). "Pengujian RESTful API Menggunakan Postman pada Aplikasi Mobile." Jurnal Informatika, 4(2), 69-74.

Fannani, Ilham. 2020. Perancangan Aplikasi POINT OF SALES Berbasis Website Togoide Studi Kasus UKM Laundry Menggunakan Metode Iterative dan Incremental.

Marom, N. (2020). Perancangan User Interface Pada Aplikasi Customer Layanan Pemesanan Makanan “EatAja” Menggunakan Metode User Centered Design.

Siddik, M. N. A. (2020). Perancangan User Interface Mitra Sebagai Penerima Pesanan Pada Aplikasi EATAJA Mitra Dengan Metode Goal-Directed Design

Pramono, R., & Utami, A. P. (2021). Pengembangan Model Konseptual pada Penelitian Sistem Informasi Berbasis Design Science. Jurnal Ilmiah Teknologi Informasi Terapan (JITTER), 3(1), 24-31.

Utami, A. P., & Pramono, R. (2020). Analisis Perencanaan Awal dalam Metode Scrum pada Pengembangan Perangkat Lunak. Jurnal Sistem Informasi Bisnis, 10(2), 101-108.

Septiandani, D., & Fauziah, N. (2019). Analisis Kelebihan Metode Scrum dalam Pengembangan Perangkat Lunak. Jurnal Teknik Informatika, 12(2), 176-184.

Daryanto, A., & Rahardjo, S. (2021). Pemanfaatan Teknik Pengumpulan Data dalam Riset Kualitatif. Jurnal Ilmiah Ilmu Administrasi Publik, 10(1), 35-47.

Darmawan, I., & Nugroho, A. (2021). Pengolahan Data dan Analisis Kuantitatif dalam Penelitian Sosial. Jurnal Sosiologi dan Pendidikan, 17(1), 51-64.

Maulidia, E., & Santoso, H. B. (2021). Evaluasi Aplikasi Berbasis Website menggunakan Metode User Acceptance Testing. Jurnal Sistem Informasi, 17(1), 22-29.