

KELOMPOK 14

PEMBANGUNAN
APLIKASI MANAGEMENT
INVENTORY APOTEK
DUMA MENGGUNAKAN
ARSITEKTUR
MICROSERVICE



About Us



Horas Sidabalok



Mario Manurung



Nania Pangaribuan



Vanessa Siahaan

Deskripsi

Pembangunan Aplikasi berbasis mobile ini dirancang khusus untuk Sistem Inventory Apotek Duma. Aplikasi ini dibangun dengan menggunakan dua bahasa pemrograman yang berbeda untuk bagian back-end dan front-end. Untuk bagian back-end, digunakan bahasa Go yang dikenal dengan performanya yang tinggi dan efisien. Sedangkan untuk bagian front-end, digunakan bahasa Dart dengan framework Flutter, yang menawarkan kemampuan untuk membuat antarmuka pengguna yang responsif dan menarik. Aplikasi ini menggunakan arsitektur microservice, di mana sistem dibagi menjadi beberapa layanan kecil yang berdiri sendiri. Setiap layanan memiliki database-nya masing-masing dan berfungsi secara independen.





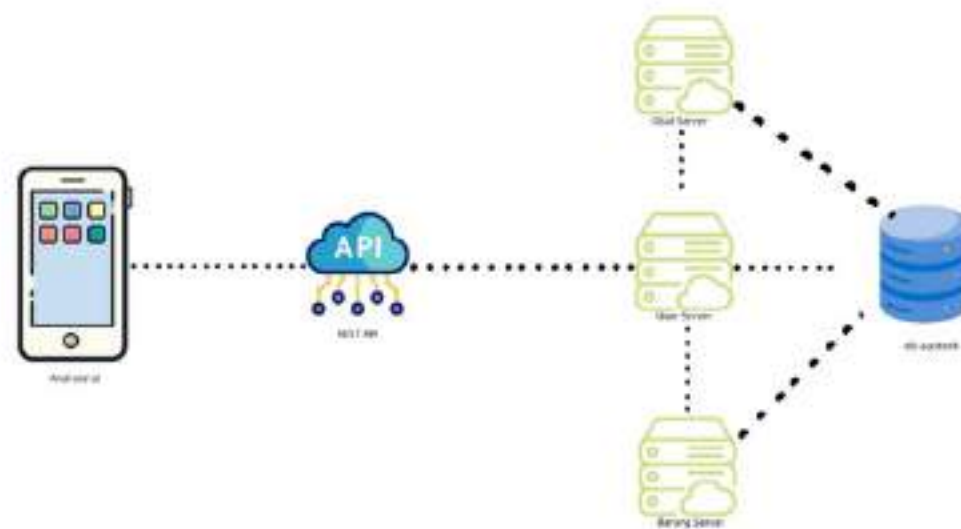
Arsitektur Monolith

Arsitektur monolith adalah pendekatan tradisional dalam pengembangan perangkat lunak di mana semua komponen aplikasi dikelompokkan dalam satu kode basis besar yang di-deploy sebagai satu unit pada server. Kesalahan pada satu bagian dapat menyebabkan kegagalan total, menjadikannya kurang ideal untuk aplikasi yang memerlukan fleksibilitas, skalabilitas, dan ketahanan tinggi seiring pertumbuhan.

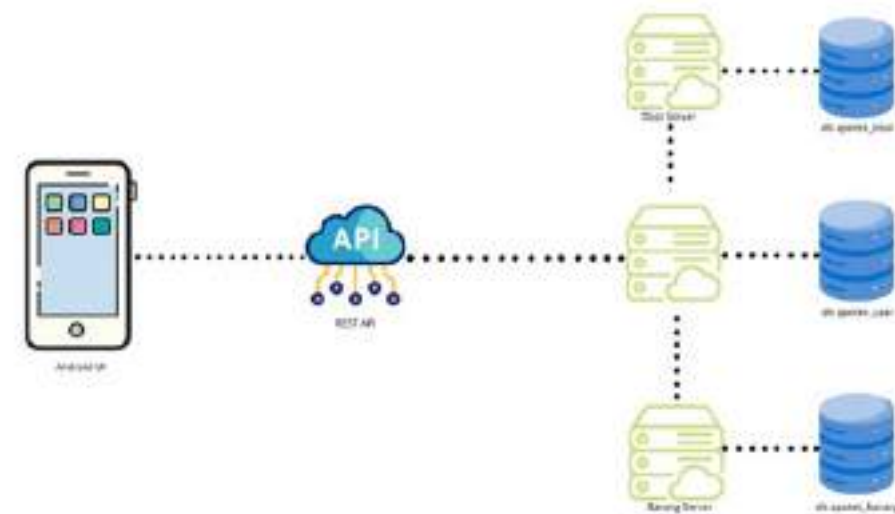
Arsitektur Microservice

Setiap layanan dirancang untuk beroperasi di port yang berbeda, memastikan kinerja yang optimal dan memungkinkan pemeliharaan yang terpisah. Dengan arsitektur ini, aplikasi dapat lebih fleksibel, efisien, dan mudah dikembangkan serta dikelola.

Architecture Monolith



Architecture Microservice



Fungsi

Adapun fungsi yang terdapat pada Aplikasi Management Inventory Apotek Duma ini, yaitu:

1. Fungsi login

Fungsi login digunakan oleh admin dan staff untuk dapat masuk dan mengakses aplikasi.

2. Fungsi mengelola obat

Fungsi mengelola obat digunakan oleh admin dan staff untuk menambahkan, mengubah, dan menghapus obat. Hal ini bertujuan untuk dapat menampilkan dan mengelolah data obat.

3. Fungsi mengelola barang

Fungsi mengelola barang digunakan oleh admin dan staff untuk menambahkan, mengubah, dan menghapus barang. Hal ini bertujuan untuk dapat menampilkan dan mengelolah data barang.

4. Fungsi mengelola kategori obat

Fungsi mengelola kagtegori obat digunakan oleh admin dan staff untuk menambahkan, mengubah, dan menghapus kategori obat. Hal ini bertujuan untuk dapat menampilkan dan mengelolah data obat.

5. Fungsi mengelola kategori barang

Fungsi mengelola kategori barang digunakan oleh admin dan staff untuk menambahkan, mengubah, dan menghapus kategori barang. Hal ini bertujuan untuk dapat menampilkan dan mengelolah data barang.

6. Fungsi menambah akun staff

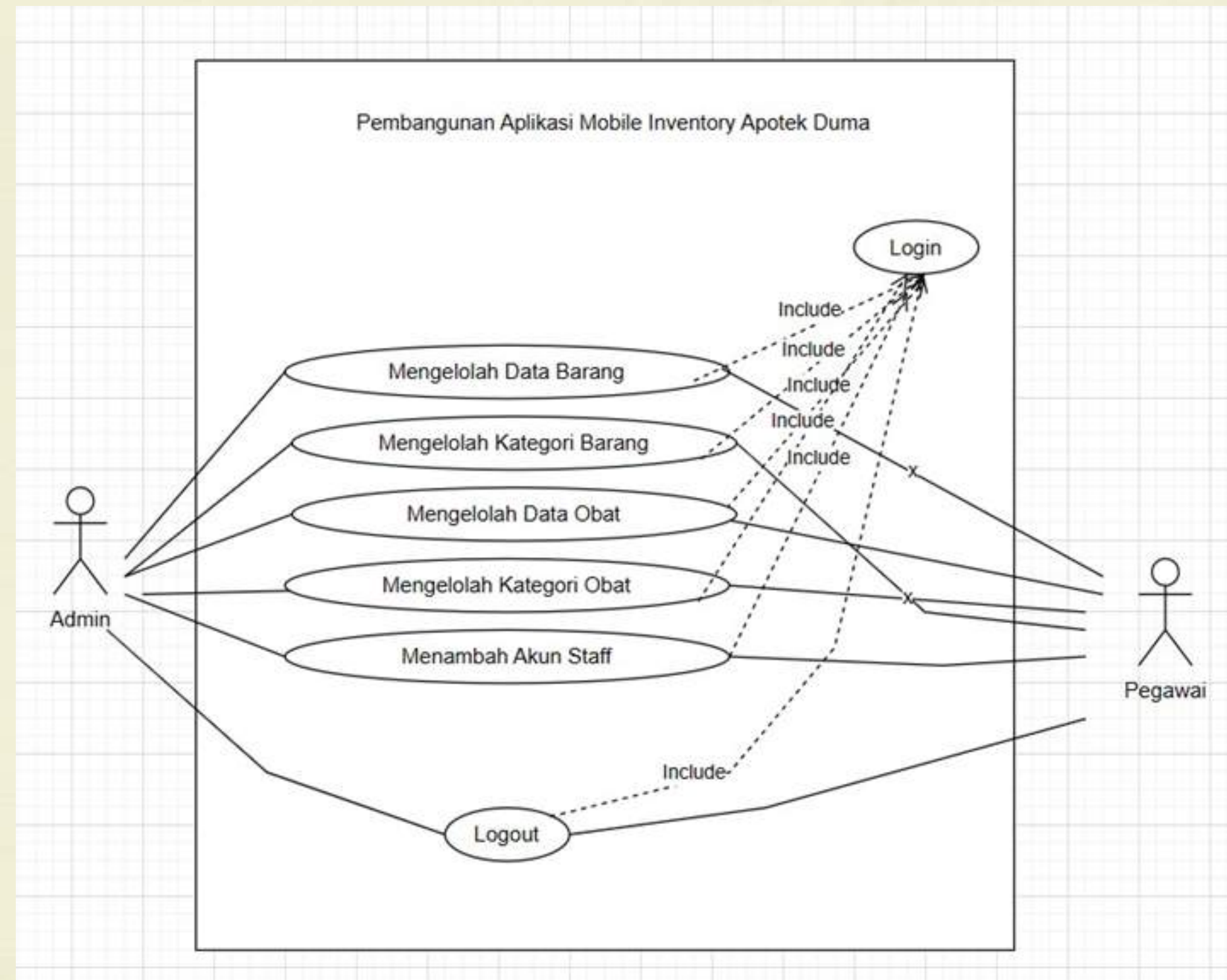
Fungsi menambah akun staff digunakan oleh admin untuk menambahkan akun staff. Hal ini bertujuan untuk dapat menampilkan dan menambah akun staff pada aplikasi.

7. Fungsi logout

Fungsi logout digunakan oleh admin dan staff untuk dapat keluar dari aplikasi.



USE CASE DIAGRAM



Demo Proyek





Kesimpulan

Microservice adalah pendekatan arsitektur dalam pengembangan perangkat lunak yang memecah aplikasi besar menjadi sejumlah layanan kecil yang mandiri. Setiap layanan atau "microservice" menjalankan satu fungsi bisnis tertentu dan berkomunikasi dengan layanan lain melalui antarmuka API. Karakteristik utama dari microservices meliputi modularitas, di mana aplikasi dipecah menjadi beberapa layanan kecil yang masing-masing memiliki fungsionalitas spesifik, serta independensi yang memungkinkan setiap layanan dikembangkan, diuji, dan dideploy secara mandiri tanpa bergantung pada layanan lain. Skalabilitas menjadi lebih efisien karena setiap layanan dapat diskalakan secara independen. Selain itu, microservices memungkinkan penggunaan teknologi yang beragam, di mana setiap layanan dapat dibangun menggunakan teknologi dan bahasa pemrograman yang berbeda sesuai kebutuhan spesifiknya. Pengelolaan data juga dilakukan secara terpisah, dengan setiap layanan biasanya memiliki basis data sendiri, yang mengurangi ketergantungan antar layanan. Komunikasi antar layanan dilakukan melalui API atau sistem pesan, biasanya menggunakan protokol HTTP/REST atau gRPC. Dengan pendekatan ini, pengembangan dan pemeliharaan aplikasi menjadi lebih fleksibel, mudah diadaptasi, dan tangguh terhadap perubahan teknologi atau kebutuhan bisnis.

**THANK
YOU**

Kelompok 14

