



Aplikasi Tiket Booking Kapal Tomok - Ajibata

Pengembangan Aplikasi Terdistribusi

Members



Benyamin Sibarani



David Silalahi



Desrico Siallagan



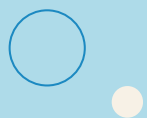
Alvina Siallagan



Listra Sidabutar



Yessi Slpahutar



Outline

01 Perbedaan Monolith dan
Microservice

02 Microservice User

03 Microservice Database

04 Microservice Server

05 Microservice Pull

06 Microservice Push

Fungsi Fungsi yang akan di bangun

01

Fungsi Register Pengguna

1. Penumpang Mendaftarkan Akun
2. Akun pemilik kapal ditambahkan oleh admin.

02

Fungsi Login Pengguna

Pengguna dapat masuk dan mengakses fitur aplikasi

03

Fungsi Mengelola Rute

Menambahkan dan mengedit rute keberangkatan kapal

04

Fungsi Mengelola Kapal

Menambah dan mengedit Informasi Kapal

05

Fungsi Mengelola Nahkoda

Menambahkan dan mengedit Informasi Nahkoda

06

Fungsi Mengelola Jadwal

Menambah jadwal dengan informasi rute, kapal, nahkoda, waktu keberangkatan, stok tiket, dan harga per tiket

07

Fungsi Memesan Tiket

Penumpang memesan tiket kapal sesuai jadwal yang tersedia

Perbedaan Monolith dan Microservice

Monolith

- Monolith bergantung pada satu server yang menangani semua fungsi atau program. Jika server tersebut mengalami masalah atau tidak berjalan, maka seluruh fungsi atau program dalam monolith tidak akan berjalan.



Microservice

- Terdapat beberapa service yang masing-masing menangani fungsi-fungsi tertentu.
- Setiap service berjalan di servernya sendiri atau di server yang berbeda-beda antara service. Dengan demikian,
- Seluruh program atau fungsi tidak ditangani oleh satu server tunggal lagi.
- Jika salah satu server yang menangani suatu service mengalami masalah atau tidak berjalan, hal ini tidak akan berdampak pada seluruh aplikasi secara keseluruhan dan tidak akan mengganggu service yang lain yang tetap berjalan dengan baik.



Microservice User Interface

- Definisi:
 - Bagian dari aplikasi yang berinteraksi langsung dengan pengguna.
 - Biasanya berupa aplikasi web atau aplikasi seluler.
- Karakteristik:
 - Decoupled Frontend and Backend:
 - Frontend (antarmuka pengguna) tidak terikat langsung dengan backend tertentu.
 - Antarmuka hanya mengetahui endpoint API yang disediakan oleh berbagai layanan microservice.
- Fungsi:
 - Memuat konten secara dinamis dari berbagai layanan microservice.
 - Contoh:
 - Memuat jadwal kapal dari service jadwal.
 - Menampilkan informasi user dari service user.

Microservice Database

- Defenisi:
 - Setiap layanan microservice memiliki database tersendiri.
- Keuntungan:
 - Isolasi dan Otonomi:
 - Mendukung isolasi dan otonomi masing-masing layanan.
 - Minimalkan Dampak Masalah:
 - Masalah pada satu layanan tidak mempengaruhi layanan lainnya.
- Implementasi:
 - Contoh dalam Aplikasi Pemesanan Jadwal Kapal:
 - Service user, service kapal, service nahkoda, service rute, service jadwal, dan service order memiliki database tersendiri.
 - Jenis Database:
 - Semua layanan dapat menggunakan jenis database yang sama (misalnya MySQL di port 3306).
 - Setiap layanan memiliki skema database yang berbeda

Microservice Server

- Definisi:
 - Infrastruktur yang mendukung pelaksanaan layanan microservice secara independen.
- Karakteristik:
 - Isolasi Layanan:
 - Setiap layanan berjalan pada server terpisah atau dalam container terpisah dengan port yang berbeda.
 - Independen:
 - Setiap layanan dapat di-deploy, di-skalakan, dan diperbarui secara independen.
- Keuntungan:
 - Isolasi Kegagalan:
 - Masalah pada satu layanan tidak mempengaruhi layanan lain.
- Contoh Implementasi:
 - Service user:
 - Berjalan di server dan port tertentu.
 - Service jadwal:
 - Berjalan di server dan port lain.
 - Toleransi Kegagalan:
 - Jika server service kapal down, service user tetap berfungsi.



Pull Model

- Defenisi:
 - berarti client atau service meminta (menarik) data atau informasi dari server atau service lain secara terpisah.
- Keuntungan:
 - Isolasi dan Otonomi:
 - Mendukung isolasi dan otonomi masing-masing layanan.
 - Minimalkan Dampak Masalah:
 - Masalah pada satu layanan tidak mempengaruhi layanan lainnya.
- Implementasi:
 - Contoh dalam Aplikasi Pemesanan Jadwal Kapal:
 - Service user, service kapal, service nahkoda, service rute, service jadwal, dan service order memiliki database tersendiri.
 - Jenis Database:
 - Semua layanan dapat menggunakan jenis database yang sama (misalnya MySQL di port 3306).
 - Setiap layanan memiliki skema database yang berbeda

Push Model

1. Push Data/Informasi:

- Server atau service mengirimkan data atau informasi ke client atau service lain tanpa diminta terlebih dahulu.

2. Contoh Implementasi:

- Pembaruan stok tiket dalam aplikasi jadwal.

3. Notifikasi Pembaruan Stok:

- Service jadwal mendorong informasi langsung ke client saat ada perubahan stok tiket.

4. Proses Otomatis:

- Pesan notifikasi muncul di user interface setelah tindakan seperti pemesanan jadwal selesai.

5. Update Real-Time:

- Service jadwal mengirimkan pembaruan stok tiket langsung ke UI saat terjadi perubahan

Alternative Resources

Thanks!

