Nama : Andre Fransisko

Kelas : TK2A

Matkul : Pemograman Microservice

Rangkuman dari dokumen yang disediakan adalah tentang berbagai konsep terkait dengan arsitektur microservices, service orchestration, service choreography, komunikasi RPI, dan penggunaan database per service. Berikut adalah penjelasan lengkap untuk setiap slide yang terdapat dalam dokumen tersebut:

-Keuntungan Service Orchestration

Poin Utama: Service orchestration memudahkan pembuatan aplikasi karena logika bisnis terpusat di Aggregation Microservices. Hal ini juga membuat kode menjadi lebih mudah dimengerti karena fokus pada satu tempat.

Implikasi: Dengan menggunakan service orchestration, pengembang dapat dengan mudah mengelola dan memahami logika bisnis aplikasi mereka.

-Keuntungan Menggunakan RPI

Poin Utama: RPI (Remote Procedure Invocation) merupakan pendekatan yang sederhana dan mudah digunakan untuk komunikasi antara layanan. Biasanya digunakan untuk komunikasi tipe Request-Reply dan proses yang bersifat synchronous.

Implikasi: Penggunaan RPI dapat mempermudah interaksi antar layanan dalam aplikasi yang memerlukan respons cepat dan sinkron.

-Kelebihan Arsitektur Microservices

Poin Utama: Arsitektur microservices memiliki kelebihan dalam hal kemudahan pemahaman, pengembangan, skalabilitas, dan fleksibilitas teknologi. Layanan yang relatif kecil memudahkan pengembangan, pemeliharaan, pengujian, dan penyebaran aplikasi.

Implikasi: Dengan arsitektur microservices, tim pengembang dapat bekerja lebih efisien, mengadopsi teknologi baru dengan mudah, dan menyesuaikan skala aplikasi sesuai kebutuhan.

Masalah di Komunikasi RPI

Poin Utama: Terdapat beberapa masalah yang mungkin timbul dalam implementasi komunikasi RPI, seperti proses yang memakan waktu, pengiriman data berulang, dan kompleksitas dalam membuat proses paralel.

Implikasi: Pengembang perlu memperhatikan masalah-masalah yang mungkin muncul saat menggunakan komunikasi RPI untuk memastikan kinerja aplikasi tetap optimal.

Permasalahan Banyak Jenis Frontend

Poin Utama: Terdapat berbagai jenis permasalahan yang dapat muncul terkait dengan kompleksitas pengembangan frontend aplikasi.

Implikasi: Pengembang perlu mempertimbangkan berbagai faktor yang memengaruhi pengembangan frontend agar dapat menghasilkan aplikasi yang optimal dan sesuai dengan kebutuhan pengguna.

Keuntungan Service Choreography

Poin Utama: Service choreography memberikan keuntungan dalam hal interaksi yang lebih dinamis antara layanan, proses bisnis yang lebih fleksibel, dan kolaborasi yang lebih efektif.

Implikasi: Dengan menggunakan service choreography, aplikasi dapat dirancang untuk beradaptasi dengan perubahan bisnis dan lingkungan dengan lebih baik.

Kenapa Harus Database per Service?

Poin Utama: Penggunaan database per layanan memastikan bahwa setiap layanan tidak saling tergantung satu sama lain, memungkinkan setiap layanan menggunakan database sesuai kebutuhan, dan mengurangi kompleksitas dalam interaksi antar layanan.

Implikasi: Dengan pendekatan ini, pengembang dapat meningkatkan isolasi antar layanan, meningkatkan fleksibilitas, dan mengurangi risiko kesalahan dalam pengelolaan data.

Contoh Kasus

Poin Utama: Slide ini menyajikan contoh kasus tanpa penjelasan lebih lanjut. Mungkin digunakan untuk mengilustrasikan penerapan konsep-konsep sebelumnya dalam situasi nyata.

Implikasi: Contoh kasus dapat membantu pengembang memahami bagaimana konsep-konsep yang telah dipelajari dapat diterapkan dalam konteks yang lebih konkret.

Pembagian Aplikasi Microservices

Poin Utama: Pembagian aplikasi menjadi layanan-layanan kecil (microservices) seperti Merchant, Product, dan Shipping.

Implikasi: Dengan membagi aplikasi menjadi layanan-layanan kecil, pengembang dapat meningkatkan modularitas, skalabilitas, dan fleksibilitas aplikasi mereka.

Dengan memahami dan menerapkan konsep-konsep tersebut, pengembang dapat merancang, mengembangkan, dan mengelola aplikasi dengan lebih efisien, skalabilitas, dan fleksibilitas yang lebih baik sesuai dengan kebutuhan bisnis dan teknis yang ada.