

Contoh Soal Desain Aplikasi Backend Menggunakan Message Broker dan Memcache

1. Sistem Pemesanan Makanan Online

Desain:

- **Message Broker:** Digunakan untuk mengelola komunikasi antara berbagai komponen sistem, seperti:
 - **Pesanan:** Saat pelanggan memesan makanan, pesan order dipublikasikan ke broker.
 - **Dapur:** Mengonsumsi pesan order dari broker untuk mempersiapkan makanan.
 - **Kurir:** Mengonsumsi pesan order dari broker untuk mengantarkan makanan.
- **Database:** Digunakan untuk menyimpan menu makanan, data profil pengguna, dan informasi status pemesanan untuk meningkatkan performa.

Manfaat:

- Dekupling komponen sistem, meningkatkan skalabilitas dan fleksibilitas.
- Komunikasi real-time antara komponen sistem untuk memastikan kelancaran proses pemesanan.
- Peningkatan performa dengan caching data statis dan dinamis.

2. Aplikasi Percakapan Real-time

Desain:

- **Message Broker:** Digunakan untuk mendistribusikan pesan chat antar pengguna secara real-time:
 - **Pengguna:** Mengirim pesan chat ke broker.
 - **Server Chat:** Mengonsumsi pesan chat dari broker dan meneruskan ke pengguna yang dituju.
- **Memcache:** Digunakan untuk caching data pengguna, status percakapan, dan informasi avatar untuk meningkatkan kecepatan respons.

Manfaat:

- Komunikasi real-time antara pengguna untuk interaksi yang lancar.
- Peningkatan kecepatan respons dengan caching data pengguna dan percakapan.
- Penurunan beban server dengan mengurangi akses database yang berulang.

3. Sistem Agregator Berita

Desain:

- **Message Broker:** Digunakan untuk mengelola aliran data berita dari berbagai sumber:
 - **Sumber Berita:** Menerbitkan berita ke broker.
 - **Mesin Agregasi:** Mengonsumsi berita dari broker, menganalisis, dan mengkategorikannya.
 - **Aplikasi Pengguna:** Menerima berita yang dikategorikan dari broker.
- **Database:** Digunakan untuk menyimpan berita yang telah dianalisis dan dikategorikan untuk meningkatkan efisiensi.

Manfaat:

- Pengumpulan dan agregasi data berita yang terpusat dan efisien.
- Pengiriman berita yang terpersonalisasi dan relevan kepada pengguna.
- Peningkatan performa dengan caching data berita yang telah diolah.

4. Platform E-commerce dengan Pesan Instan

Desain:

- **Message Broker:** Digunakan untuk memfasilitasi komunikasi antara pelanggan dan penjual:
 - **Pelanggan:** Mengirim pesan chat ke penjual melalui broker.
 - **Penjual:** Mengonsumsi pesan chat dari broker dan meresponnya.
 - **Notifikasi Pesan:** Broker mendistribusikan notifikasi pesan kepada pengguna yang relevan.
- **Memcache:** Digunakan untuk caching data produk, informasi penjual, dan riwayat percakapan untuk meningkatkan performa.

Manfaat:

- Komunikasi real-time antara pelanggan dan penjual untuk meningkatkan pengalaman berbelanja.
- Notifikasi pesan yang instan dan terjamin.
- Peningkatan performa dengan caching data produk dan percakapan.

5. Sistem Rekomendasi Konten

Desain:

- **Message Broker:** Digunakan untuk mendistribusikan data interaksi pengguna dan aktivitas mereka:
 - **Aplikasi Pengguna:** Mengirim data interaksi pengguna ke broker.

- **Mesin Rekomendasi:** Mengkonsumsi data interaksi dari broker, menganalisisnya, dan menghasilkan rekomendasi konten.
- **Aplikasi Pengguna:** Menerima rekomendasi konten yang dipersonalisasi dari broker.
- **Memcache:** Digunakan untuk caching model rekomendasi, data profil pengguna, dan riwayat interaksi untuk meningkatkan efisiensi.

Manfaat:

- Rekomendasi konten yang dipersonalisasi dan relevan berdasarkan interaksi dan aktivitas pengguna.
- Pengalaman pengguna yang lebih menarik dan interaktif.
- Peningkatan performa dengan caching model rekomendasi dan data pengguna.

6. Sistem Notifikasi Pemesanan Tiket

Desain:

- **Message Broker:** Digunakan untuk mengelola dan mendistribusikan notifikasi pemesanan tiket:
 - **Aplikasi Pengguna:** Menerbitkan permintaan pemesanan tiket ke broker.
 - **Sistem Notifikasi:** Mengkonsumsi permintaan dari broker dan mengirimkan notifikasi pemesanan ke pengguna.
- **Memcache:** Digunakan untuk caching data acara, jadwal, dan status pemesanan untuk mempercepat akses dan respons.

Manfaat:

- Penyampaian notifikasi pemesanan tiket secara real-time kepada pengguna.
- Peningkatan performa dengan caching data acara dan status pemesanan.
- Mengurangi beban server dengan caching data yang sering diakses.

7. Sistem Shopping Cart

Desain:

- **Message Broker:** Digunakan untuk mengelola aliran data keranjang belanja:
 - **Aplikasi Pengguna:** Mengirim data produk yang ditambahkan ke keranjang ke broker.
 - **Sistem Checkout:** Mengkonsumsi data dari broker untuk memproses pembayaran.
- **Memcache:** Digunakan untuk caching data keranjang belanja sementara, informasi pengguna, dan detail produk untuk meningkatkan performa.

Manfaat:

- Pengelolaan data keranjang belanja yang terpusat dan efisien.
- Peningkatan kecepatan akses dan respons dengan caching data keranjang belanja.
- Komunikasi real-time antara pengguna dan sistem checkout untuk pengalaman belanja yang lebih baik.

8. Product Catalog Menggunakan Elasticsearch

Desain:

- **Message Broker:** Digunakan untuk mendistribusikan pembaruan data produk:
 - **Sistem Inventaris:** Menerbitkan pembaruan data produk ke broker.
 - **Elasticsearch:** Mengonsumsi pembaruan dari broker untuk mengindeks data produk.
- **Elasticsearch:** Digunakan untuk mengindeks dan mencari data produk dengan cepat dan efisien.
- **Memcache:** Digunakan untuk caching data produk yang sering dicari dan diakses untuk meningkatkan performa.

Manfaat:

- Pencarian produk yang cepat dan efisien dengan Elasticsearch.
- Pembaruan data produk yang real-time dan terdistribusi dengan message broker.
- Peningkatan performa dengan caching data produk yang sering diakses.

Catatan: Anda dapat memilih untuk membuat program atau hanya mendesain aplikasi dengan menggunakan skenario yang telah disampaikan di atas dengan SKPL yang lengkap.