1. Kenapa perlu menyediakan fault tollerance?
    - Mengurangi down time.
    - Manajemen Kegagalan
    - Meningkatkan Reliability
    - Skalabilitas program menjadi lebih baik
2. Bagaimana cara menyediakan Fault Tollerance pada microservices?
    - Circuit breaker
    - Retry mechanism
    - Timeouts
    - Bulkheads
    - Dekomposisi Gracious
    - Back Pressure
3. Jelaskan rancangan sistem fault tollerance pada proyek microservices SPT!
    - Untuk merancang sistem fault tolerance pada proyek microservices FRK dan FED, kita perlu mempertimbangkan berbagai aspek untuk memastikan bahwa aplikasi tetap dapat beroperasi secara stabil meskipun terjadi kegagalan pada salah satu komponen. Berikut adalah langkah-langkah yang dapat diambil :
        - Autoscaling

          Konfigurasikan autoscaling untuk menyesuaikan jumlah instance microservice dengan permintaan saat ini. Dengan menggunakan autoscaling, sistem akan secara otomatis menambah atau mengurangi jumlah instance berdasarkan beban kerja.
        - Replikasi dan Skalabilitas

          Replikasi microservices FRK dan FED di beberapa instance atau node untuk meningkatkan ketersediaan aplikasi. Gunakan alat manajemen kontainer seperti Kubernetes untuk mengelola replikasi dan otomatisasi penyeimbangan beban. Pertimbangkan untuk mengelompokkan microservice berdasarkan fungsi dan membaginya ke dalam kluster yang terpisah.
        - Health Check

          Setiap instance microservice harus memiliki mekanisme pemeriksaan kesehatan yang terus menerus. Konfigurasikan sistem untuk mendeteksi instance yang tidak responsif atau bermasalah dan menggantikannya dengan instance baru secara otomatis.
        - Load Balancing

Gunakan load balancer untuk mendistribusikan lalu lintas antara instance microservice yang aktif. Load balancer harus dapat mendeteksi instance yang tidak responsif dan mengalihkan lalu lintas ke instance yang berfungsi dengan baik.

- Graceful Degradation

  Implementasikan mekanisme graceful degradation untuk memastikan bahwa aplikasi dapat berfungsi dalam kondisi yang tidak optimal. Misalnya, jika layanan FRK atau FEK tidak tersedia, aplikasi harus tetap memungkinkan pengguna untuk mengakses fungsionalitas lain yang masih tersedia.
- Circuit Breaker Pattern

  Terapkan pola circuit breaker untuk melindungi aplikasi dari kegagalan beruntun yang dapat menyebabkan kegagalan total. Circuit breaker harus dapat mengidentifikasi kegagalan secara cepat dan menonaktifkan sementara layanan yang bermasalah.
- Logging dan Monitoring

  Terapkan sistem logging yang kuat untuk melacak aktivitas aplikasi dan mencatat kejadian penting serta kesalahan. Gunakan sistem monitoring untuk memantau kesehatan dan kinerja microservices serta infrastruktur secara keseluruhan
- Data Replication dan Backup

  Pastikan data FRK dan FED direplikasi di beberapa lokasi untuk mencegah kehilangan data. Atur rutinitas backup data untuk memastikan bahwa data yang penting dapat dipulihkan jika terjadi kegagalan sistem.