

JAWABAN HANYA REFERENSI DAN BELUM TENTU BENAR

1. Jelaskan istilah scalability dalam konteks sistem terdistribusi serta jelaskan pula pentingnya dalam sistem terdistribusi!

Skalabilitas (scalability) adalah kemampuan suatu sistem untuk meningkatkan kapasitas atau kinerja dengan menanggapi peningkatan beban atau permintaan yang lebih tinggi. Dalam konteks sistem terdistribusi, skalabilitas mengacu pada kemampuan sistem untuk mempertahankan kinerja yang baik atau bahkan meningkatkan kinerja seiring dengan peningkatan jumlah pengguna, volume data, atau kebutuhan komputasi.

Pentingnya skalabilitas dalam sistem terdistribusi :

a. Peningkatan Kapasitas

Sistem terdistribusi harus mampu menanggapi peningkatan beban dengan meningkatkan kapasitasnya. Skalabilitas memungkinkan sistem untuk menangani pertumbuhan data atau pengguna tanpa mengalami penurunan kinerja atau kegagalan sistem.

b. Penghematan Biaya

Dengan adanya skalabilitas, perusahaan atau organisasi dapat membangun infrastruktur yang sesuai dengan kebutuhan saat ini, sambil memungkinkan untuk mudah diperluas di masa depan. Hal ini menghindari pemborosan sumber daya dan investasi yang tidak perlu pada awalnya.

c. Fleksibilitas

Sistem yang dapat diskalakan memungkinkan perubahan dan adaptasi lebih cepat terhadap kebutuhan bisnis yang berubah. Perubahan dalam permintaan atau pola penggunaan dapat dengan mudah diakomodasi dengan meningkatkan sumber daya secara proporsional.

d. Kepuasan Pengguna

Dalam sistem terdistribusi yang skalabel, pengguna akan mendapatkan pengalaman yang lebih baik karena waktu tanggapan lebih cepat dan ketersediaan layanan yang tinggi. Ini meningkatkan kepuasan pengguna dan mempertahankan loyalitas mereka.

e. Daya Tahan dan Ketersediaan

Skalabilitas juga berhubungan dengan daya tahan (resilience) dan ketersediaan (availability). Dengan sistem yang dapat diskalakan, jika satu bagian sistem mengalami kegagalan, beban dapat dialihkan ke bagian lain yang masih berfungsi, menjaga ketersediaan layanan.

f. Kemudahan Pemeliharaan

Sistem yang baik dalam hal skalabilitas umumnya lebih mudah untuk dikelola dan dipelihara. Penambahan atau pengurangan sumber daya dapat dilakukan dengan lancar tanpa mengganggu operasi sistem secara keseluruhan.

JAWABAN HANYA REFERENSI DAN BELUM TENTU BENAR

2. Berikan satu contoh komputasi tersebar, lengkap dengan roles serta responsibilities, serta penjelasannya

Contoh komputasi tersebar yaitu Sistem Penyimpanan Awan (Cloud Storage)

Roles dan Responsibilities:

- **Frontend Interface:**
Role: Mengelola antarmuka pengguna yang memungkinkan pengguna untuk mengakses, mengunggah, mengunduh, dan mengelola berkas di cloud storage.
Responsibilities:
 - Menyediakan antarmuka pengguna yang ramah dan mudah digunakan untuk mengakses dan mengelola berkas di cloud storage.
 - Memvalidasi dan meneruskan permintaan pengguna ke backend server sesuai kebutuhan.
 - Menampilkan informasi terkait status berkas dan transaksi kepada pengguna.
- **Backend Server:**
Role: Menangani logika bisnis, manajemen data, dan interaksi dengan berbagai komponen dalam sistem penyimpanan awan.
Responsibilities:
 - Mengelola logika bisnis terkait penyimpanan, pengambilan, pembaruan, dan penghapusan berkas di cloud storage.
 - Menangani otorisasi dan otentikasi pengguna.
 - Mengelola infrastruktur terkait penanganan permintaan pengguna.
- **Data Storage Cluster:**
Role: Menyimpan dan mengelola data pengguna secara terdistribusi.
Responsibilities:
 - Menyimpan dan mempertahankan data berkas pengguna dengan keandalan dan ketersediaan tinggi.
 - Mengelola replikasi dan distribusi data untuk memastikan ketersediaan dan performa tinggi.
 - Melakukan penanganan beban kerja untuk memastikan pengelolaan data yang efisien.
- **Load Balancer:**
Role: Mengatur aliran lalu lintas dari frontend server ke backend server untuk memastikan keseimbangan beban kerja
Responsibilities:
 - Memantau kinerja server dan memutuskan distribusi lalu lintas untuk mencegah overloading pada satu server.
 - Menyediakan toleransi kegagalan dengan mengalihkan lalu lintas ke server yang berfungsi jika satu server mengalami masalah.
- **Security and Authentication Module:**
Role: Menangani keamanan dan otentikasi pengguna serta data dalam sistem.
Responsibilities:

JAWABAN HANYA REFERENSI DAN BELUM TENTU BENAR

- Mengelola otentikasi pengguna untuk memastikan akses yang sah ke berkas dan sistem.
 - Mengimplementasikan kebijakan keamanan, enkripsi, dan kontrol akses untuk melindungi data pengguna dari akses yang tidak sah.
3. Sebuah mesin client mengirim data dengan panjang pesan 10 kilo byte ke server dalam sebuah lingkungan terdistribusi, yang memiliki latency 5 mili detik dan data transfer rate 16 mega bits per detik. Hitunglah message transmission time untuk data tersebut dalam satuan detik.

$$\text{Message Transmission Time} = \frac{\text{Message Size}}{\text{Data Transfer Rate}} + \text{Latency}$$

Namun, sebelum itu, kita perlu memastikan bahwa satuan pengukuran digunakan konsisten. Latency sudah dalam satuan detik (5 mili detik = 0,005 detik), namun data transfer rate dalam satuan megabit per detik. Kita perlu mengonversi ukuran pesan dari kilobyte ke bit:

$$\text{Message Size (bits)} = \text{Message Size (kilobytes)} \times 8 \times 1024$$

Sekarang, mari kita substitusikan nilai-nilai ini ke dalam rumus:

$$\begin{aligned} \text{Message Transmission Time} \\ = \frac{\text{Message Size(bits)}}{\text{Data Transfer Rate(bits per second)}} + \text{Latency(seconds)} \end{aligned}$$

$$\begin{aligned} \text{Message Transmission Time} \\ = \frac{10 \text{ kilobytes} \times 8 \times 1024}{\text{Data Transfer Rate 16 megabits per second}} + 0.005 \text{ s} \end{aligned}$$

$$\text{Message Transmission Time} \approx 0.0512 \text{ s}$$

4. Modifikasi kode

```
public void run() {
    try {
        // Menerima jari-jari lingkaran dari client
        String radiusString = in.readUTF();

        try {
            // Mengubah teks jari-jari menjadi tipe data double
            double radius = Double.parseDouble(radiusString);

            // Menghitung luas lingkaran
            double area = Math.PI * Math.pow(radius, 2);

            // Mengirim luas lingkaran kembali ke client dalam bentuk
            teks
```

JAWABAN HANYA REFERENSI DAN BELUM TENTU BENAR

```
        out.writeUTF("Luas lingkaran dengan jari-jari " + radius +  
" adalah: " + area);  
    } catch (NumberFormatException e) {  
        // Mengirim pesan kesalahan jika format jari-jari tidak  
valid  
        out.writeUTF("Kesalahan format, tidak bisa menghitung luas  
lingkaran");  
    }  
  
    } catch (IOException e) {  
        System.out.println("IO: " + e.getMessage());  
    }  
}
```