

Nama : Sang Made Diva Wikananda  
Nim : 21101143  
Kelas : Cloud Computing (Q)

### UAS

1. Elastisitas dalam cloud computing memang sangat penting! Ini seperti kemampuan sistem untuk secara dinamis menyesuaikan diri dengan perubahan beban kerja tanpa perlu intervensi manual. Kelebihan elastisitas ini sangat penting dalam pengelolaan sumber daya karena bisa mengoptimalkan penggunaan infrastruktur. Saat beban kerja meningkat, sistem dapat secara otomatis menambah kapasitasnya, dan ketika beban berkurang, kapasitasnya bisa dikurangi. Jadi, kita bisa hemat biaya dan tetap efisien.
2. IaaS lebih seperti memberikan organisasi fondasi dan bahan bangunan (infrastruktur, jaringan, penyimpanan) sementara SaaS adalah paket lengkap dengan cerita yang sudah jadi (aplikasi siap pakai). Jadi, kapan memilih IaaS atau SaaS? Itu tergantung kebutuhan dan kontrol yang diinginkan. Kalau organisasi butuh kebebasan lebih dalam mengelola dan mengonfigurasi infrastrukturnya sendiri, IaaS bisa jadi pilihan. Misalnya, buat organisasi yang punya kebutuhan khusus atau aplikasi internal yang kompleks. Tapi, kalau fokusnya pada efisiensi dan penggunaan aplikasi tanpa perlu khawatir sama infrastruktur, SaaS lebih menggiurkan. Bayangkan kayak punya petugas yang nyiapin dan merawat semua perangkat dan software buat kita. Jadi, intinya, IaaS buat yang suka DIY (Do It Yourself), sementara SaaS buat yang pengen "ready to use."
3. Docker dan konsep containerization menyajikan solusi yang sangat efisien dalam pengembangan dan pengelolaan aplikasi. Kontainer berfungsi sebagai unit mandiri yang memuat aplikasi beserta semua dependensinya, memastikan konsistensi dan portabilitas lintas lingkungan. Keberadaan kontainer memberikan kepastian bahwa aplikasi yang berjalan di satu lingkungan, seperti laptop pengembang, akan konsisten dengan lingkungan produksi. Hal ini menghilangkan potensi masalah yang mungkin timbul akibat perbedaan konfigurasi. Pentingnya kontainer juga tercermin dalam kemampuannya untuk bergerak secara fleksibel antar lingkungan, dari pengembangan lokal hingga implementasi di cloud. Ini membantu meningkatkan efisiensi pengembangan dan mempercepat siklus pengembangan. Selain itu, penggunaan kontainer juga mempermudah proses implementasi (deployment) aplikasi. Dengan kontainer, aplikasi dapat diemas dengan semua dependensinya dan diimplementasikan dengan mudah di berbagai platform. Dengan demikian, Docker dan containerization memberikan fondasi yang solid untuk mempercepat dan menyederhanakan pengembangan serta pengelolaan aplikasi secara menyeluruh.
4. Redundansi dan ketersediaan tinggi adalah kunci dalam lingkungan cloud computing karena mereka menjamin kelangsungan operasional dan kinerja yang optimal. Bayangkan cloud computing seperti pusat kendali di mana data dan aplikasi kita "tinggal." Jadi, kalau ada gangguan atau kegagalan, harus ada rencana cadangan yang bisa diandalkan. Faktor-

faktor ini penting karena mengurangi risiko downtime. Ketersediaan tinggi memastikan bahwa layanan tetap berjalan meski ada kegagalan pada satu komponen. Dengan redundansi, data dan beban kerja bisa disalin dan didistribusikan di beberapa lokasi atau server sehingga jika satu mati, yang lain bisa mengambil alih tanpa gangguan. Namun, mencapai tingkat ketersediaan yang diinginkan tidak selalu mudah. Tantangannya termasuk manajemen kompleksitas infrastruktur yang lebih tinggi, biaya tambahan untuk implementasi redundansi, dan memastikan koordinasi yang efektif antar berbagai komponen sistem. Selain itu, tetap menghadapi ancaman seperti serangan siber, bencana alam, atau kesalahan manusia yang dapat mengganggu ketersediaan. Jadi, sementara ketersediaan tinggi itu sangat diinginkan, mencapainya seringkali melibatkan keseimbangan yang cermat antara biaya, kompleksitas, dan keamanan.

5. Keamanan di cloud public dan private adalah dua sisi koin yang perlu diperhatikan dengan cermat. Cloud public menawarkan kenyamanan dan skala ekonomis, sementara cloud private memberikan kontrol dan keamanan lebih langsung. Cloud public cenderung memberikan tingkat keamanan yang tinggi karena penyedia cloud bertanggung jawab untuk mengamankan infrastruktur dan menyediakan layanan keamanan terkini. Namun, organisasi harus memastikan bahwa mereka memahami dan mengelola dengan benar pengaturan keamanan berbasis peran dan izin. Di sisi lain, cloud private memberikan tingkat kontrol yang lebih besar karena infrastruktur dikelola secara internal. Organisasi dengan kebutuhan keamanan tinggi, seperti di industri keuangan atau kesehatan, sering memilih cloud private untuk menjaga data sensitif. Kapan sebaiknya memilih cloud public atau private? Itu tergantung pada kebutuhan dan prioritas organisasi. Cloud public bagus untuk skala, elastisitas, dan biaya yang lebih rendah, sementara cloud private lebih sesuai untuk organisasi yang membutuhkan kendali penuh atas keamanan dan ingin menjaga data di bawah kendali langsung mereka. Beberapa organisasi bahkan memilih solusi campuran (hybrid), menggabungkan keuntungan dari keduanya. Intinya, pemilihan antara cloud public dan private seharusnya selaras dengan tujuan bisnis dan kebutuhan keamanan khusus organisasi.
6. Virtualisasi dan containerization adalah dua konsep yang berbeda, dan keduanya memiliki peran unik dalam konteks pengembangan dan implementasi aplikasi. Virtualisasi menciptakan lingkungan virtual yang terisolasi di atas sistem operasi host. Setiap mesin virtual memiliki sistem operasi sendiri dan memerlukan sumber daya yang signifikan. Ini memberikan isolasi penuh, tetapi bisa menjadi berat secara resource. Sementara itu, containerization, terutama menggunakan Docker, membagi sistem operasi host menjadi beberapa kontainer yang berbagi kernel. Mereka lebih ringan dan membagi sumber daya lebih efisien daripada mesin virtual. Kelebihan utamanya adalah portabilitas dan konsistensi lintas lingkungan. Apa yang berjalan di lingkungan pengembangan akan sama dengan lingkungan produksi. Kelebihan menggunakan Docker dan kontainer termasuk efisiensi sumber daya, waktu respons yang lebih cepat, dan kemudahan dalam pengelolaan. Kontainer mudah dipindahkan, dideploy, dan dikelola, membuat siklus pengembangan dan

implementasi lebih cepat dan efisien. Jadi, jika efisiensi, portabilitas, dan manajemen yang mudah adalah prioritas, Docker dan kontainer bisa menjadi pilihan yang lebih unggul dibandingkan dengan pendekatan virtualisasi tradisional.

7. Skalabilitas horizontal dalam arsitektur cloud itu seperti keajaiban yang memungkinkan sistem tumbuh seiring dengan tuntutan tanpa harus membebani instans tunggal. Bayangkan sistem yang bisa berkembang dengan cara menambah lebih banyak instans sejalan dengan meningkatnya lalu lintas!. Manfaatnya memang signifikan, terutama ketika ada lonjakan lalu lintas tiba-tiba. Dengan menambahkan lebih banyak instans, sistem bisa menyebar beban kerja dan memastikan performa tetap optimal. Hal ini juga menghindari risiko kegagalan tunggal yang bisa terjadi jika hanya mengandalkan satu instans. Skalabilitas horizontal juga memungkinkan efisiensi penggunaan sumber daya. Saat lonjakan lalu lintas mereda, kita bisa dengan mudah mengurangi jumlah instans untuk menghindari pemborosan sumber daya. Jadi, intinya, konsep skalabilitas horizontal itu seperti membekali sistem dengan kemampuan regenerasi dan fleksibilitas untuk mengatasi tantangan lonjakan lalu lintas. Itu adalah salah satu elemen kunci untuk membuat aplikasi dan layanan tetap responsif dan dapat diandalkan dalam lingkungan cloud.
8. Perbandingan antara SaaS dan FaaS adalah seperti memilih antara paket komplit dengan aplikasi siap pakai (SaaS) atau fokus pada fungsi spesifik tanpa perlu mengelola infrastruktur (FaaS). SaaS cocok untuk kebutuhan organisasi yang menginginkan solusi lengkap tanpa harus repot dengan pengelolaan infrastruktur atau pengembangan kode khusus. Ini seperti menyewa rumah yang sudah furnished, siap pakai tanpa perlu repot-repot membangun atau mengonfigurasi. Sementara FaaS, seperti serverless computing, lebih fokus pada pengembangan fungsi-fungsi kecil dan spesifik. Cocok untuk situasi di mana kita ingin fokus pada kode atau fitur tertentu tanpa harus mengelola infrastruktur secara menyeluruh. Ini seperti membangun blok bangunan satu per satu sesuai kebutuhan. Kapan lebih masuk akal menggunakan FaaS atau SaaS? Jika organisasi butuh solusi all-in-one tanpa repot, SaaS mungkin lebih tepat. Tapi jika fokusnya pada pengembangan fungsi-fungsi spesifik tanpa harus pikirin infrastruktur, FaaS bisa jadi pilihan yang lebih hemat waktu dan efisien. Jadi, intinya, pilihannya tergantung pada kebutuhan dan prioritas organisasi dalam hal pengelolaan infrastruktur dan fokus pengembangan.
9. Docker Hub itu seperti pusat keberlanjutan untuk ekosistem kontainer. Ini menyediakan repositori sentral yang memfasilitasi manajemen kontainer dengan mudah. Ada beberapa keuntungan menarik dalam menggunakan Docker Hub, terutama dalam konteks repositori publik. Pertama, Docker Hub menyediakan tempat sentral untuk menyimpan dan mengelola image kontainer. Ini memudahkan kolaborasi dan berbagi di antara pengembang. Misalnya, jika ada gambar keren yang sudah dibuat oleh komunitas, kita bisa dengan mudah mengunduhnya dari Docker Hub. Kedua, repositori publik seperti Docker Hub mempercepat siklus pengembangan dengan menyediakan akses cepat ke image-image umum dan populer. Kita bisa mendapatkan image dasar dengan mudah dan membangun

aplikasi kita di atasnya. Ketiga, Docker Hub memiliki fitur otomatisasi dan integrasi yang mempermudah proses CI/CD (Continuous Integration/Continuous Deployment). Kita bisa mengotomatiskan alur kerja untuk membangun, menguji, dan mendeploy kontainer dengan cepat dan efisien. Namun, penting juga untuk mempertimbangkan keamanan dan privasi. Dalam konteks bisnis atau proyek yang memerlukan tingkat keamanan lebih tinggi, menggunakan repositori kontainer pribadi atau repositori yang dikelola sendiri mungkin lebih sesuai. Jadi, sementara Docker Hub memfasilitasi manajemen kontainer dengan cara yang luar biasa, keuntungan penggunaannya tergantung pada kebutuhan spesifik dan tingkat keamanan yang diinginkan.

10. Keamanan data dalam cloud computing memang jadi prioritas utama. Beberapa langkah konkret yang dapat diambil untuk meningkatkan keamanan data dalam lingkungan cloud melibatkan kombinasi kebijakan, teknologi, dan kesadaran.

- Enkripsi:

Pastikan data disandikan selama penyimpanan dan pengiriman. Enkripsi end-to-end dapat membantu melindungi data dari potensi ancaman di seluruh siklus hidupnya.

- Manajemen Akses:

Terapkan kontrol akses yang ketat. Hanya berikan izin akses yang diperlukan kepada pengguna dan kelompok, dan perbarui izin secara teratur sesuai perubahan peran atau tanggung jawab.

- Audit dan Monitoring:

Aktifkan audit dan monitoring secara terus-menerus. Dengan melacak aktivitas dan mengaudit log, organisasi bisa mendeteksi potensi ancaman atau aktivitas yang mencurigakan.

- Kebijakan Keamanan yang Ketat:

Tentukan dan terapkan kebijakan keamanan yang ketat. Ini bisa mencakup penggunaan kata sandi yang kuat, kebijakan retensi data, dan aturan akses yang jelas.

- Pemulihan Bencana dan Cadangan Data:

Pastikan ada rencana pemulihan bencana yang baik dan lakukan pencadangan data secara teratur. Jika terjadi masalah, organisasi harus dapat memulihkan data mereka dengan cepat.

- Pendidikan dan Pelatihan Karyawan:

Tingkatkan kesadaran keamanan di antara karyawan. Pelatihan tentang praktik keamanan data dan ancaman siber membantu mengurangi risiko human error.

- **Pemilihan Penyedia Cloud yang Terpercaya:**

Pilih penyedia cloud yang memiliki reputasi baik dalam hal keamanan. Pastikan penyedia cloud menyediakan alat dan fitur keamanan yang memadai.

- **Keamanan Aplikasi:**

Pastikan aplikasi yang di-host di cloud juga aman. Melakukan pengujian keamanan secara teratur dan mengimplementasikan praktik pengembangan aman bisa membantu melindungi data.

Ini adalah beberapa langkah umum yang dapat diambil, tetapi penting untuk memahami bahwa keamanan data di cloud merupakan tanggung jawab bersama antara penyedia cloud dan organisasi pengguna. Terus memantau perkembangan teknologi dan menerapkan tindakan keamanan yang relevan adalah kunci untuk menjaga keamanan data di lingkungan cloud.