**SOAL - SOAL KUIS DSAD**

Session I

1. To share resources and loads are the reason of why Distributed System is important (T)

Jawab :

Sistem terdistribusi adalah kumpulan komponen yang independen, yang saling terhubung melalui jaringan dan terlihat seolah-olah seperti single sistem yang saling terhubung.

1. CIA is Confidentiality, Integrity, and Avalaibility (T)

Jawab :

Confidentiality : Terautorisasi

Integrity : Keutuhan data terjaga, dan kalaupun ada perubahan, developer harus tahu

Avalaibility : Tersedia dan dapat digunakan

1. Middleware is heterogeneity(T)

Middleware dapat mengenkapsulasi / menyembunyikan keberagaman platform dalam suatu sistem, sehingga setiap komponen dapat berkomunikasi

1. Mengapa DS mampu mengenkapsulasi kompleksitasnya?

DS mampu mengenkapsulasi kompleksitasnya karena ada middleware yang menyembunyikan keberagaman platform yang ada dalam sistem tersebut.

1. Exlplain concurency, if there is no, what will happen?

Consurency adalah suatu kondisi dimana terdapat 2 proses atau lebih mengakses resource yang sama. Jika tidak terjadi, maka sistem / resorce nya akan menjadi kacau. Kemungkinan tidak ada proses yang berjalan (tidak mereply satupun).

1. Write and explain at least 3 challenges of DS!

3challenges of DS :

1. Heterogeneity (dapat menyembunyikan komplesitasnya)
2. Openness (dapat digunakan kembli oleh developer yang lain)
3. Security (memenuhi CIA(confidentiality, integrity, dan Avalailability))
4. Scalability (Dapat diperbesar kapasitasnya)
5. Concurrency (dapat diakses 2 atau lebih proses sekaligus)
6. Fault tolerance (tetap dapat berjalan keika terdapt kesalahan pada salah satu komponen)
7. Resiliency (dapat mememberi solusi terhadap masalah yang dihadapi oleh dirinya sendiri)
8. Transparent (terlihat seolah-oleh eperti single system)

Session II

1. In DS Layer, platform consist of hardware and software and middleware (F)

False, because DS Layer consist of Application(Service), Middleware(Corba, RPC, dan RMI), and Platform, where platform consist of hardware and software (OS)

1. Type of Network connectionless and connection-oriented (T)

True. Connectionless : hanya mengirim request, dan tidak bertanggung jawab dengan kesalahan atau ketidaklengkapan data.

1. DS layer consist of 3 layer; application, middleware, and platform (T)

True. Application : service, Middleware : mengenkapsulasi keberagaman, Platform : (OS + hardware)

1. Why DS cheaper than CS?

DS lebih murah daripada CS karena DS tidak menggunakan mainframe yang mahal dan dapat berbagi beban sehingga tampak seolah-olah seperti single system, karena perubahan pada CS harus mengubah pusat yang besar (pengaruh nya besar).

1. Why open system may expose vulnerability ?

Karena bersifat open, maka dapat diutak-atik, dan kelemahannya terlihat, jadi tidak aman.

Opennes 🡪 proses komunikasinya (dapat diakses) melalui jaringan, sehingga dapat ditemuka kelemahannya

Dokumentasinya terbuka, rahasianya bocor, pdhl dokumentasinya penting agr dapat di-extend.

1. Explain proxy and cache !

Katika seseorang menggunakan proxy untuk mengakses suatu alamat web, cache menyimpan alamat tersebut, sehingga ketika suatu alamat yang sama diakses dengan proxy yang sama, maka dapat langsung segera di load karena sudah tersimpan di cache tadi.

Ketika client request ke server, pertama dilakukan memeriksa proxy server apakah yang diminta client terdapt pada proxy, kemudian cache melakukan penyimpanan local untuk IP yang pernah diakses, sehingga dapat meningakatkan kerja dan mengurangi beban kerja.

Session III

1. Multicast adalah tipe komunikasi antara single proses dan lebih dari 1 proses

True. Multicast terjadi pada satu proses ke lebih dari satu proses.

1. Syncronuous is non-blocking (F)

False, karena syncronuous is blocking, yaitu proses lain dapat terjadi setelah request-responds telah selesai dilakukan.

1. There are 3 protocol to commerce on IPC (F)

False, there are 4 protocol to commerce IPC, namely :

1. Connect (establish connection)
2. Send (transmitting)
3. Receive (accepting)
4. Destroy (de-allocate connection)
5. Explain the beneficial of using socket than file, pipes, and shared memory!

Keutungannya adalah socket dapat berjalan di berbeda tempat (host). A socket instance may be used by a group of working processes

Koneksi melalui jaringan

Komunikasi nya 2 arah (dibandingkan dengan pipes)

1. Explain how IPC works through a distinguished file!

By file, two or more process can be communicated in the same host inside the operating system.

File harus menulis ke disk dulu, kalau memori terganggu, file gak akan

1. What actually IPC is?

Sebenarnya IPC adalah bagaimana 2 proses atau lebih dapat berkomunikasi.

Session IV

1. Both RMI and RPC must have a stub to do remote invocation (T)

True, karena stub digunakan sebagai perantara antara client dengan skeleton di server, jadi stub itu harus ada sebelum melakukan remote invocation, dan stub juga digunakan untuk marshall dan unmarshall.

1. In RMI skeleton called proxy (F)

False, skeleton bukan proxy, melainkan stub lah yang disebut juga proxy.

Dispatcher 🡪 menerima dan meneruskan request

Skeleton 🡪 yang melakuka marshal dan unmarshall

1. Middleware (like in either RPC or RMI) will automate marshall and unmarshall process

True, karena di dalam RPC dan RMI terdapat komponen yang dpat melakukan marshall dan unmarshal.

1. What the main differences (you feel) between socket and RPC? Give the clear example!

* Perbedaan antara socket da RPC adalah socket yang terdiri ari IP+Port digunkan sebagai alat komunikasi antara suatu process dengan process lain. Sedangkan RPC digunakan client untuk memanggil fungsi / prosedur yang ada di proses lain (server).
* Dalam RPC, socket tidak lagi diperdulikan, dengan artian : bahwa RPC telah mengautomatisasi pekerjaan socket (dalam hal ini marsall dan unmarshalling). Sedangkan pada socket menggunakan socket programming, kita masih harus mengatur bagaimana request akan tiba & bagaimana hasil (result) akan dikirimkan.

Contoh :

Dalam socket 🡪 kita harus mengimplement serializable pada class

Pada RPC 🡪hal tersebut tidak diperlukan lagi

Encrypt 🡪 menggunakan algoritma tertentu (belum tentu bisa dikembalikan ke bentuk awalnya)

1. How RMI works? (Dah tau, kn?)

Proxy menangkap request

1. What is the main reason for independence process to invoke procedure from another process? Give the clear example!Agar tidak perlu membuat prosedur yang berulang pada suatu proses, jadi hanya tinggal memanggil posedur saja. Contoh : proses login di akademik memanggil sebuah prosedur login do weserver, begitu juga proses login di zimbra.

Session V

1. Preserves system invariant merupakan *goal* dari durable.  
    Jawaban : FALSE karena itu merupakan *goal* dari consistency.
2. Sebutkan kekurangan dari Nested Transaction ?

Child tergantung kepada parent, parent ngabort satu proses, maka semua proses harus di abbort

1. What does Concurrency mean? (Dah tau jg, kn?)
2. Cara mendeteksi deadlock.

Mengatasi apa yang udah terjadi (WFG)

1. Type locking (Ya tau lah ya…:D)

Tipe locking ada 2, yaitu :

Read locking 🡪 shared loking

Write locking 🡪 exclusive locking

1. Two phase adalah cara untuk mengatasi (dealing with) nested transaction

False, karena itu adalah phase dalam distributed

Di nested hanya commit aja phase nya

**Semoga Bermanfaat ☺\_MH\_070**