UG COLORING THE GLOBAL FUTURE

1. PENGENALAN SISTEM TERDISTRIBUSI

Kemal Ade Sekarwati

1. Mengapa Sistem terdistribusi ?

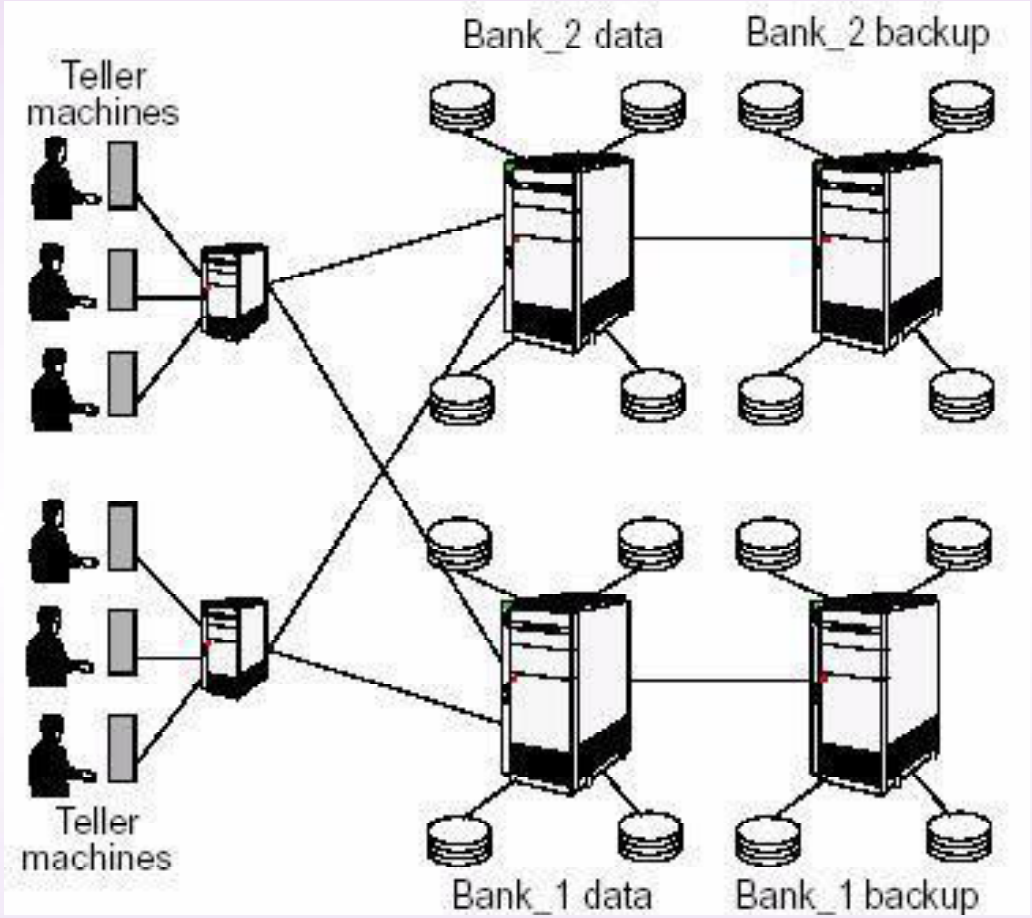
- Komputer-komputer yang terdistribusi secara geografis.
- Komunikasi melalui koneksi kabel/fibre/wireless/.
- Keuntungan : interaksi, koorporasi dan pemakaian bersama sumber daya, mengurangi biaya, meningkatkan kinerja dan ketersediaan.

Definisi Sistem Terdistribusi

Sistem Terdistribusi adalah Sekumpulan komputer otonom yang terhubung ke suatu jaringan, dimana bagi pengguna sistem terlihat sebagai satu komputer.

Komputer otonomi : walaupun komputer tidak terhubung ke jaringan, komputer tersebut tetap dapat berjalan.

Dengan menjalankan sistem terdistribusi, komputer dapat melakukan : Koordinasi Aktifitas dan berbagi sumber daya : hardware, software dan data



Gambar 1.1: Contoh sistem terdistribusi, Automatic Banking (teller machine) System

Proses :

- Dieksekusi secara konkuren.
- Berinteraksi untuk mencapai tujuan umum.
- Aktifitasnya saling koordinasi dan bertukar informasi melalui pesan yang ditransfer melalui jaringan komunikasi.

Dengan definisi tersebut diatas, apakah internet merupakan suatu sistem terdistribusi ?

2. Contoh Sistem Terdistribusi

- Sistem Telepon : ISDN, PSTN
- Manajemen Jaringan : Adminstrasi sesumber jaringan
- Network File System (NFS) : Arsitektur untuk mengakses sistem file melalui jaringan
- WWW : Arsitektur client/server yang diterapkan di atas infrastruktur internet, Shared Resource (melalui URL)
- dll

3. Keuntungan Sistem Terdistribusi

- Performance
- Distribution
- Reliability (Fault tolerance)
- Incremental Growth
- Sharing Data/Resources

4. Permasalahan dalam Sistem Terdistribusi

Kelemahan pada sistem terdistribusi adalah :

- Kesulitan dalam membangun perangkat lunak :

  bahasa pemrogramman, sistem operasi dll.
- Masalah Jaringan : merancang & mengimplementasikan sistem.
- Masalah Keamanan : berbagi data/sumber daya →

  berkaitan dengan keamanan data dll.

5. Karakteristik Sistem Terdistribusi

Hal yang diperhatikan dalam membangun sistem terdistribusi :

a. Transparency (Kejelasan)
b. Communication (Komunikasi)
c. Performance & Scalability (Kinerja dan Ruang Lingkup)
d. Heterogenity (Keanekaragaman)
e. Opennes (Keterbukaan)
f. Reliability dan Fault Tolerancy (Kehandalan dan Toleransi Kegagalan)
g. Security (Kemanan)

a. Transparency

- Access transparency

  Local & remote resources dapat diakses dengan operasi yg sama

- Location transparency

  Resource dapat diakses tanpa tahu di mana lokasinya.

  Bagaimana pendapat Anda mengenai hyperlink & URL?

- Migration (Mobility) transparency

  Resource dan klien dapat berpindah tanpa mempengaruhi operasi pemakai atau program

- Replication transparency

  Pemakai maupun pemrogram aplikasi tidak perlu mengetahui adanya replikasi *resource*, yang dapat meningkatkan kehandalan dan unjuk kerja.

- Concurency transparency

  Beberapa proses dapat sama-sama menggunakan suatu *resource* tanpa saling interferensi.

  Bagaimana jika beberapa pemakai secara bersamaan akan mengubah suatu berkas?

- Failure transparency

  Pemakai dan pemrogram aplikasi dapat menyelesaikan tugasnya walaupun ada kegagalan hardware atau software.

- Performance transparency

  Sistem dapat dikonfigurasi ulang untuk meningkatkan unjuk kerja, sejalan dengan perubahan beban sistem.

- *Scaling transparency :*

  Sistem dan aplikasi mudah bertambah luas tanpa perubahan struktur sistem dan algoritma aplikasi.

b. Communication

Komponen pada sistem terdistribusi harus melakukan komunikasi dalam suatu urutan sebagai berikut :

- Infrastruktur jaringan (interkoneksi dan software jaringan)
- Metode dan Model komunikasi yang cocok

  Metode komunikasi : Send, Receive, Remote Procedure Call
- Model Komunikasi : client - server communication, groupmulticast

c. Performance and Scalability

Faktor yang mempengaruhi kinerja (performance) :

- Kinerja dari pada personal workstations
- Kecepatan infrastruktur komunikasi
- Fleksibilitas dalam membagi beban kerja

contoh : apabila terdapat prosesor (workstation) yang idle maka dapat dialokasikan secara otomatis untuk mengerjakan tugas-tugas user.

Scalability

Sistem tetap harus memperhatikan efesiensi walaupun terdapat penambahan secara signifikan user atau sumber daya yang terhubung :

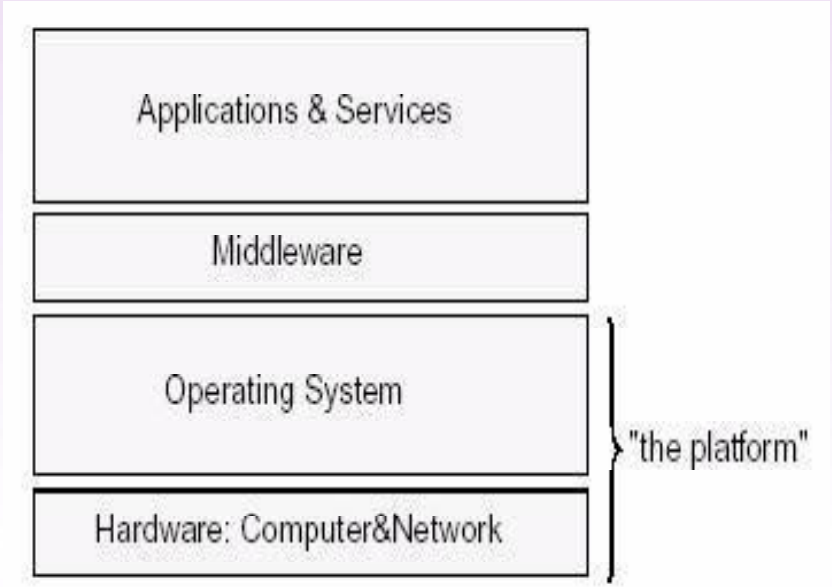
- Cost (biaya) penambahan sumber daya (resources) harus reasonable.
- Penurunan kinerja (performance) diakibatkan oleh penambahan user atau sumber daya harus terkontrol.

d. Heterogenity

Aplikasi yang terdistribusi biasa berjalan dalam keberagaman :

- Hardware : mainframes, workstations, PC's, server dll.
- Software : UNIX, MS Windows, IMB OS/2, LINUX dll.
- Devices : teller machine, robot, sistem manufacturing dll.
- Network dan Protocol : Ethernet, FDDI, ATM, TCP/IP dll

Melihat keaneka ragaman di atas maka salah satu solusi yang bisa diterapkan adalah Middleware : berfungsi sebagai jembatan untuk komunikasi dan proses.

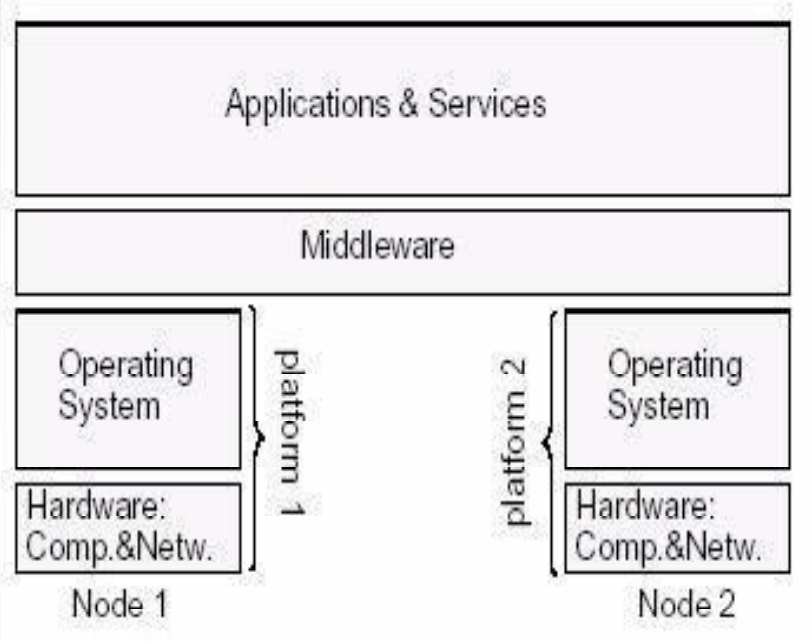


Gambar 1.2: Arsitektur sofware pada sistem terdistribusi

e. Opennes

Hal terpenting yang harus dimiliki oleh sistem terdistribusi adalah opennes (keterbukaan) dan flexibility (fleksibilitas) :

- Setiap layanan (services) harus dapat diakses oleh semua user.
- Mudah dalam implementasi, install dan debug services;
- User dapat membuat dan menginstall service yang telah dibuat oleh user tersebut.



Gambar 1.3: Sistem Terdistribusi pada dua titik

Aspek kunci pada opennes :

- Interface dan Protocol yang standard (seperti protokol komunikasi di internet)
- Support terhadap keanekaragaman. (dengan membuat midleware seperti CORBA)

f. Reliability dan Fault Tolerance

Availability : kalau mesin mati (down), sistem tetap harus berjalan dengan jumlah layananan yang tersisa.

- Komponen yang sangat vital (critical resources) berjumlah seminimal mungkin.

- Software dan Hardware harus direplikasi : kalau terjadi kegagalan / error maka yang lain akan menangani.

- Data dalam sistem tidak boleh hilang.
  copy file disimpan secara redundan pada server lain, tapi tetap harus dijaga konsistensi datanya.

Fault Tolerance : Sistem harus bisa mendeteksi kegagalan dan melakukan tindakan dengan dasar sebagai berikut :

Mask the fault (menutupi kegagalan) : tugas harus dapat dilanjutkan dengan menurunkan kinerja tapi tanpa terjadi kehilangan data atau informasi.

Fail Gracefully : membuat suatu antisipasi terhadap suatu kegagalan ke suatu prosedur yang telah direncanakan dan memungkinkan untuk menghentikan proses dalam waktu yang singkat tanpa menghilangkan informasi atau data.

g. Security

- Confidentiality : keamanan terhadap data yang diakses oleh user yang tidak diperbolehkan (unauthorizes user)

- Integrity : keamanan terhadap kelengkapan dan autentikasi data.

- Availability : Menjaga agar resource dapat selalu diakses.

- Hal lain yang harus dijamin dalam sistem terdistribusi : penggunaan rerources yang tepat oleh user yang berlainan.

6. Model dalam Sistem Terdistribusi

Model dalam sistem terdistribusi :

a. Model Arsitektur (Architectural Models)

b. Model Interaksi (Interaction Models)

c. Model Kegagalan (Failure Models)

a. Architectural Models

cara kerja antar komponen sistem dan bagaimana komponen tsb berada pada sistem terdistribusi :

- Client - Server Model
- Proxy Server
- Peer processes (peer to peer)

- *Client-Server Mo*del

  Model client-server biasanya berbasiskan protokol request/reply.

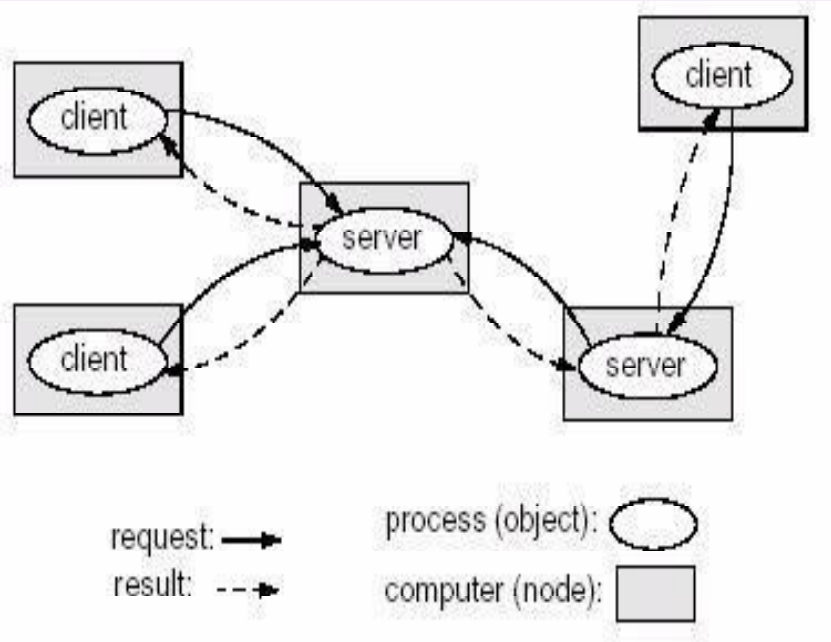
  Contoh :

  Implementasi RPC (Remote Procedure Calling) dan RMI (Remote Method Invocation) :

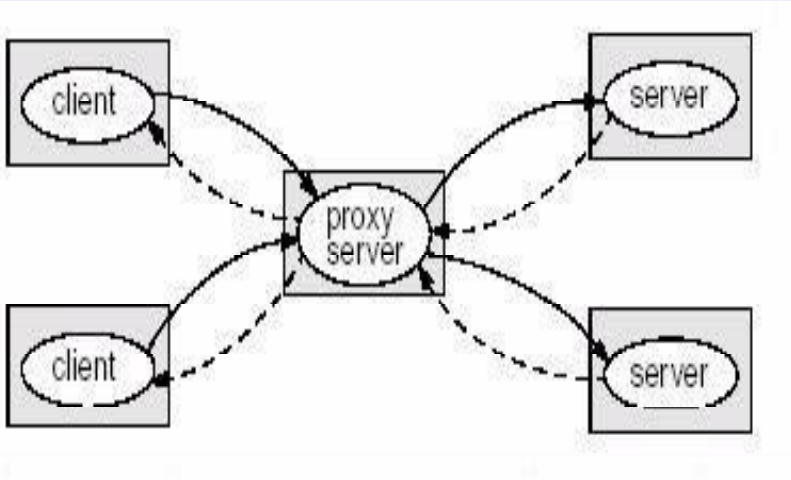
  - client mengirimkan request berupa pesan ke server untuk mengakses suatu service.
  - server menerima pesan tersebut dan mengeksekusi request client dan mereply hasil ke client

- *Proxy Server*

- menyediakan hasil copy (replikasi) dari resource yang diatur oleh server lain
- dipakai untuk menyimpan hasil copy web resources.
- Ketika client melakukan request ke server, proxy server diperiksa apakah yang diminta oleh client terdapat pada proxy server.
- diletakkan pada setiap client atau dapat dipakai bersama oleh beberapa client. Tujuannya adalah meningkatkan performance dan availibity dengan mencegah frekwensi akses ke server.



Gambar 1.4 : Model arsitektur client-server



Gambar 1.5: Model Proxy Server

- *Peer Process*

Semua proses (object) mempunyai peran yang sama.

- Proses berinteraksi tanpa ada nya perbedaan antara client dan server.
- Pola komunikasi yang digunakan berdasarkan aplikasi yang digunakan.
- Merupakan model yang paling general dan fleksible.

b. Interaction Models

dibagi menjadi dua bagian :

- Synchronous distributed system
- Asynchronous distributed system

Synchronous Distributed System

- Batas atas dan batas bawah waktu pengeksekusian dapat diset.
- Pesan yang dikirim, diterima dalam waktu yang sudah di tentukan
- Fluktuasi ukuran antara waktu lokal berada dalam suatu batasan.

Beberapa hal yang penting untuk diperhatikan synchronous distributed sistem :

- terdapat satu waktu global
- dapat memprediksi perilaku (waktu)
- dimungkinkan dan aman untuk menggunakan mekanisme timeout dalam mendekteksi error atau kegagalan dalam proses atau komunikasi

Asynchronous Distributed System

Banyak sistem terdistribusi yang menggunakan model interaksi ini (termasuk Internet)

- Tidak ada batasan dalam waktu pengkeksekusian.
- Tidak ada batasan dalam delay transmission (penundaan pengiriman)
- Tidak ada batasan terhadap fluktuasi waktu lokal.

Asynchronous system secara parktek lebih banyak digunakan.

c. Failure Models

Kegagalan apa saja yang dapat terjadi dan bagaimana efek yang ditimbulkan ?

- Omission Faluires
- Arbitary Failures
- Timing Failures

Kegagalan dapat terjadi pada proses atau kanal komunikasi. Penyebabnya bisa berasal dari hardware ataupun software.

Model Kegagalan (Failure Models) dibutuhkan dalam membangun suatu sistem dengan prediksi terhadap kagagalan yang mungkin terjadi.

Ommision Failures :

ketika prosesor dan kanal komunikasi mengalami kegagalan untuk melakukan hal yang seharusnya dilakukan.

Dikatakan tidak mempunyai ommision failures apabila :

- Terjadi keterlambatan (delayed) tetapi akhirnya tetap tereksekusi.
- Sebuah aksi dieksekusi walaupun terdapat kesalahan pada hasil.

Dengan synchronous system, ommision failures dapat dideteksi dengan timeouts.

Arbitary Failures

- kegagalan yang paling buruk dalam sistem.
- Tahapan proses atau komunikasi diabaikan atau yang tidak diharapkan terjadi dieksekusi → hasil yang diharapkan tidak terjadi atau megeluarkan hasil yang salah.

Timing Failures

- dapat terjadi pada synchronous system, dimana batas waktu diatur untuk eksekusi proses, komunikasi dan fluktuasi waktu.
- Timing Failures terjadi apabila waktu yang telah ditentukan terlampaui.