

## **BAB II**

### **PEMBAHASAN**

#### **A. Pengertian Jaringan Terdistribusi**

Sistem terdistribusi adalah sebuah sistem yang komponennya berada pada jaringan komputer. Komponen tersebut saling berkomunikasi dan melakukan koordinasi hanya dengan pengiriman pesan (*message passing*). Atau sebuah sistem yang terdiri dari beberapa komponen software atau hardware yang independent yang berkomunikasi dan berkoordinasi melalui message parsing baik sinkron maupun asinkron yang terlihat satu kesatuan dan dirancang untuk menghasilkan fasilitas komputasi terintegrasi.

Saya kasih contoh dari sistem terdistribusi ini yaitu:

- Internet, global jaringan interkoneksi computer yang berkomunikasi melalui IP (Internet Protocol) Protocol.
- Intranet, jaringan teradministrasi terpisah dengan batasan pada kebijakan keamanan local.
- dan masih banyak lagi

Sistem Terdistribusi adalah Sekumpulan komputer otonom yang terhubung ke suatu jaringan, dimana bagi pengguna sistem terlihat sebagai satu computer. Maksud komputer otonomi adalah walaupun komputer tidak terhubung ke jaringan, komputer tersebut tetap dapat berjalan. Dengan menjalankan sistem terdistribusi, komputer dapat melakukan :

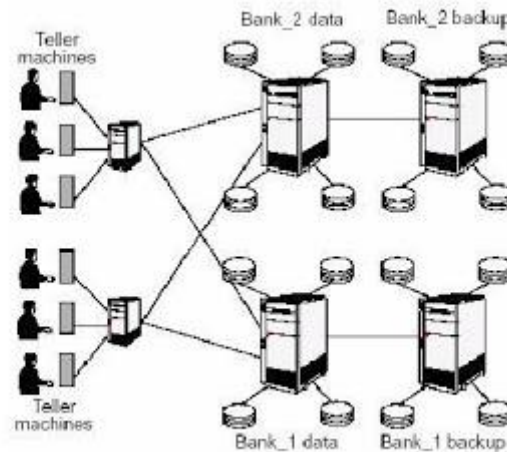
- Koordinasi Aktifitas
- Berbagi sumber daya : hardware, software dan data

Dengan definisi tersebut diatas maka internet sesungguhnya bukanlah suatu sistem terdistribusi, melainkan infrastruktur dimana sistem terdistribusi dapat di aplikasikan pada jaringan tersebut.

#### **Contoh Sistem Terdistribusi**

- Sistem Telepon
  - ISDN, PSTN

- Manajemen Jaringan
  - Administrasi sumber jaringan



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

- 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)

## B. Karakteristik Sistem Terdistribusi

Karakteristik yang dimiliki system terdistribusi, adalah sebagai berikut:

### 1. *No global clock* (Keterbatasan dalam Global Clock)

Hal ini menyebabkan kesulitan dalam mensinkronkan waktu seluruh komputer/perangkat yang terlibat. Dapat berpengaruh pada pengiriman pesan/data, seperti saat beberapa proses berebut ingin masuk ke critical session.

- Dalam pemakaian bersama atas sumber daya diperlukan beberapa hal, yaitu:
  - Dibutuhkan hardware dan software yang mendukung
  - Memerlukan resource manager
  - Perlunya suatu hubungan antara resource dengan pihak yang menggunakannya.
  - Terdapat client-server, remote evolution, code on demand, dan mobile agent.
- Terdapat batasan pada ketepatan proses sinkronisasi clock pada sistem terdistribusi, oleh karena asynchronous message passing
- Pada sistem terdistribusi, tidak ada satu proses tunggal yang mengetahui global state sistem saat ini (disebabkan oleh concurrency dan message passing)

## 2. *Independent failure*

- Setiap komponen/perangkat dapat mengalami kegagalan namun komponen/perangkat lain tetap berjalan dengan baik.
- Kemungkinan adanya kegagalan proses tunggal yang tidak diketahui.
- Proses tunggal mungkin tidak peduli pada kegagalan sistem keseluruhan

## 3. *Concurrency of components*

Pengaksesan suatu komponen/sumber daya (segala hal yang dapat digunakan bersama dalam jaringan komputer) secara bersamaan.

- Setiap komponen Hardware atau Software bersifat otonom.
- Sinkronisasi dan koordinasi dengan message passing.
- Sharing resource
- Masalah umum dalam system concurrent:
  - Deadlock
  - Lifelock
  - Komunitas yang tidak handal

Contoh: Beberapa pemakai browser mengakses halaman web secara bersamaan.

## **C. Implementasi Sistem Terdistribusi**

Sistem jaringan berbasis banyak digunakan di banyak area aplikasi seperti sistem cerdas dan sistem terdistribusi. Untuk merancang sistem terdistribusi, pengolahan informasi terdistribusi dan terminal cerdas diperlukan. Tulisan ini berkaitan dengan pertimbangan desain dan implementasi dari sistem didistribusikan menggunakan arsitektur jaringan. Node jaringan, disebut node cerdas, dirancang untuk mendukung CAN (controller area network) protokol dan metode penjadwalan pesan baru diusulkan untuk mentransfer data antara node ini. Setup eksperimental dibangun untuk melakukan kontrol kecepatan empat subsistem DC-motor. Hasil eksperimen memverifikasi bahwa sistem terdistribusi yang diusulkan memberikan kinerja kontrol diterima.

Sebuah sistem terdistribusi adalah kumpulan komputer independen yang muncul untuk para penggunaanya sebagai sistem yang koheren tunggal. Hal ini tentu merupakan bentuk ideal dari sistem terdistribusi, di mana “implementasi detail” membangun sistem yang kuat dari banyak sederhana sistem sepenuhnya disembunyikan dari pengguna.

### **Contoh sistem terdistribusi**

Mungkin yang paling sederhana dan paling terkenal contoh dari sistem terdistribusi adalah kumpulan server Web-atau lebih tepatnya, server menerapkan protokol HTTP-yang bersama-sama menyediakan database terdistribusi hypertext dan dokumen multimedia yang kita kenal sebagai World Wide Web. Contoh lain termasuk komputer dari jaringan lokal yang memberikan pandangan yang seragam dari sistem file terdistribusi dan kumpulan komputer di Internet yang menerapkan Name Service Domain (DNS).

Sebuah versi lebih canggih dari sistem terdistribusi adalah XT3 (dan XT4) seri komputer paralel oleh Cray. Ini adalah mesin performa tinggi yang terdiri dari kumpulan node komputasi yang dihubungkan oleh jaringan berkecepatan tinggi-latency rendah. Sistem operasi, UNICOS / lc, menyajikan pengguna dengan lingkungan Linux standar setelah login, tapi transparan jadwal masuk sesi selama beberapa node login yang tersedia. Namun, pelaksanaan pekerjaan komputasi paralel pada XT3 dan XT4 biasanya membutuhkan programmer untuk secara eksplisit mengatur koleksi dari menghitung node dalam kode aplikasi menggunakan versi XT3-spesifik paralel umum. Terlepas dari kenyataan bahwa sistem dalam contoh ini semua sama (karena mereka memenuhi definisi sistem terdistribusi), ada juga banyak perbedaan di antara mereka. The World Wide Web dan DNS, misalnya, keduanya beroperasi pada skala global. Sistem file terdistribusi, di sisi lain, beroperasi pada skala LAN, sedangkan superkomputer beroperasi pada skala yang lebih kecil memanfaatkan jaringan berkecepatan tinggi yang dirancang khusus untuk menghubungkan semua node nya.

### **Mengapa kita menggunakan sistem terdistribusi?**

Alternatif untuk menggunakan sistem terdistribusi adalah memiliki sistem terpusat besar, seperti mainframe. Untuk banyak aplikasi ada sejumlah alasan ekonomi dan teknis yang membuat sistem terdistribusi jauh lebih menarik daripada rekan-rekan mereka terpusat. Biaya. Lebih baik harga / kinerja selama hardware komoditas yang digunakan untuk komponen computers. Kinerja. Dengan menggunakan gabungan pengolahan dan kapasitas penyimpanan banyak node, tingkat kinerja dapat dicapai yang berada di luar jangkauan mesin terpusat.

Distribusi yang melekat. Beberapa aplikasi, seperti email dan web (di mana pengguna yang tersebar di seluruh dunia), didistribusikan secara alami. Ini termasuk

kasus di mana pengguna secara geografis serta ketika sumber daya tunggal (misalnya, printer, data) harus dibagi. Keandalan. Dengan memiliki komponen berlebihan dampak hardware dan kesalahan perangkat lunak pada pengguna dapat dikurangi. Namun, keunggulan ini sering diimbangi oleh masalah berikut yang dihadapi selama penggunaan dan pengembangan sistem terdistribusi.

Komponen baru jaringan. Jaringan diperlukan untuk menghubungkan node yang independen dan subjek keterbatasan kinerja. Selain keterbatasan ini, jaringan juga merupakan poin baru yang potensial kegagalan. Keamanan. Karena sistem terdistribusi terdiri dari beberapa komponen ada lebih elemen yang dapat dikompromikan dan harus, karena itu, harus diamankan. Hal ini membuat lebih mudah untuk kompromi sistem terdistribusi.

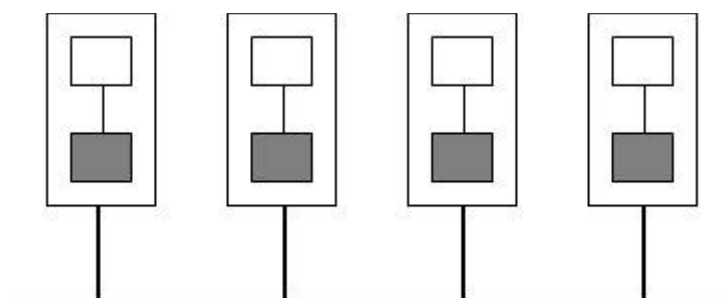
Kompleksitas perangkat lunak. Perangkat lunak yang didistribusikan lebih kompleks dan sulit untuk mengembangkan dari perangkat lunak konvensional, oleh karena itu lebih mahal untuk mengembangkan dan ada kemungkinan besar kesalahan memperkenalkan.

### **Hardware dan Software Arsitektur**

Karakteristik kunci dari definisi kita tentang sistem terdistribusi adalah bahwa hal itu mencakup aspek hardware (komputer independen) dan aspek perangkat lunak (melakukan tugas dan memberikan layanan). Dari sudut pandang hardware sistem terdistribusi umumnya diterapkan pada multicomputers. Dari sudut pandang perangkat lunak mereka umumnya diimplementasikan sebagai sistem operasi terdistribusi atau middleware.

### **Multicomputers**

Multicomputer terdiri dari node komputasi terpisah yang terhubung satu sama lain melalui jaringan (Gambar 1). Multicomputers umumnya berbeda satu sama lain dalam tiga cara:



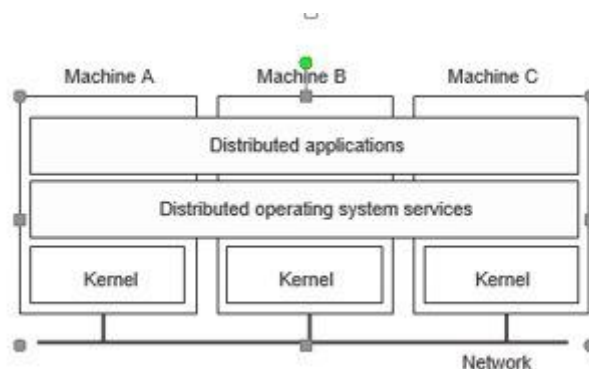
## Multicomputers

### Gambar : A.Multicomputers

1. Sumber daya Node. Ini termasuk prosesor, jumlah memori, jumlah penyimpanan sekunder, dll tersedia pada setiap node.
2. Koneksi jaringan. Koneksi jaringan antara berbagai node dapat memiliki dampak besar pada fungsi dan aplikasi bahwa sistem tersebut dapat digunakan untuk. A multi-komputer dengan jaringan bandwidth yang sangat tinggi lebih cocok untuk aplikasi yang aktif data saham atas node dan memodifikasi sejumlah besar data bersama. Sebuah jaringan bandwidth yang lebih rendah, bagaimanapun, adalah cukup untuk aplikasi di mana ada pembagian kurang intens data.
3. Homogenitas. Sebuah multicomputer homogen adalah salah satu di mana semua node yang sama, bahwa adalah mereka didasarkan pada arsitektur fisik yang sama (misalnya prosesor, bus sistem, memori, dll). Sebuah multicomputer heterogen adalah salah satu tempat node tidak diharapkan sama.

Salah satu ciri umum dari semua jenis multicomputers adalah bahwa sumber daya pada setiap node tertentu tidak dapat diakses secara langsung oleh node lain. Semua akses ke sumber daya remote akhirnya mengambil bentuk permintaan yang dikirim melalui jaringan ke node di mana sumber daya yang berada.

### Sistem Operasi Terdistribusi



## Sistem Operasi

### Gambar : Sistem Operasi Terdistribusi

Sebuah sistem operasi terdistribusi (DOS) adalah sebuah sistem operasi yang dibangun, dari bawah ke atas, untuk menyediakan layanan terdistribusi. Dengan demikian, DOS mengintegrasikan layanan terdistribusi kunci ke arsitektur (Gambar 2).

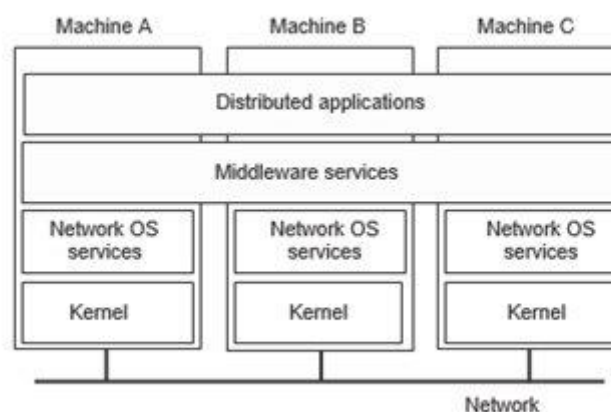
Layanan ini dapat mencakup didistribusikan memori bersama, tugas tugas untuk prosesor, masking kegagalan, penyimpanan didistribusikan, interprocess komunikasi, transparan.

berbagi sumber daya, manajemen sumber daya didistribusikan, dll Sebuah properti kunci dari sistem operasi terdistribusi adalah bahwa ia berusaha untuk tingkat yang sangat tinggi transparansi, idealnya menyediakan gambar sistem tunggal. Artinya, dengan ideal pengguna DOS tidak akan menyadari bahwa mereka, pada kenyataannya, bekerja pada sistem terdistribusi.

Sistem operasi terdistribusi pada umumnya menganggap multicomputer homogen. Mereka juga umumnya lebih cocok untuk lingkungan LAN daripada lingkungan wide-area network. Pada hari-hari awal penelitian sistem terdistribusi, didistribusikan sistem operasi mana topik utama yang menarik. Sebagian besar penelitian terfokus pada cara mengintegrasikan layanan terdistribusi ke dalam sistem operasi, atau pada cara mendistribusikan layanan sistem operasi tradisional. Saat ini, bagaimanapun, penekanan telah bergeser lebih ke arah sistem middleware. Alasan utama untuk ini adalah bahwa middleware lebih fleksibel (yaitu, tidak mengharuskan pengguna menginstal dan menjalankan sistem operasi tertentu), dan lebih cocok untuk multicomputers heterogen dan wide-area.

## Middleware

Sedangkan DOS mencoba untuk menciptakan sebuah sistem khusus untuk aplikasi terdistribusi, tujuan middleware adalah untuk menciptakan sistem antarmuka independen untuk aplikasi terdistribusi.



Middleware

Gambar : Sebuah sistem middleware

Seperti ditunjukkan dalam Gambar 3 middleware terdiri dari lapisan layanan ditambahkan antara orang-orang dari jaringan biasa OS 1 dan aplikasi yang sebenarnya. Layanan ini memfasilitasi pelaksanaan aplikasi terdistribusi dan berusaha untuk menyembunyikan heterogenitas (baik hardware dan software) dari arsitektur sistem yang mendasarinya.

Prinsip Tujuan middleware, yaitu menaikkan tingkat abstraksi untuk didistribusikan pemrograman, dicapai dalam tiga cara:

- (1) mekanisme komunikasi yang lebih nyaman dan kurang rawan kesalahan daripada lewat pesan dasar,
- (2) dari OS, jaringan protokol, bahasa pemrograman, dll dan
- (3) standar layanan (seperti layanan penamaan, layanan transaksi, layanan keamanan, dll).

Untuk membuat integrasi berbagai layanan lebih mudah, dan untuk meningkatkan transparansi, middleware biasanya didasarkan pada paradigma tertentu, atau model, untuk menggambarkan distribusi dan komunikasi. Karena paradigma adalah pendekatan yang menyeluruh untuk bagaimana didistribusikan

Sistem harus dikembangkan, ini sering memanifestasikan dirinya dalam model pemrograman tertentu seperti 'Semuanya adalah file', panggilan prosedur remote, dan objek terdistribusi. Memberikan paradigma tersebut secara otomatis menyediakan abstraksi untuk programmer untuk mengikuti, dan memberikan arahan untuk bagaimana untuk merancang dan mengatur aplikasi terdistribusi.

### **Sistem terdistribusi dan komputasi paralel**

Sistem komputasi paralel bertujuan untuk meningkatkan kinerja dengan menggunakan beberapa prosesor untuk menjalankan sebuah aplikasi. Mereka datang dalam dua rasa: sistem shared-memori dan sistem memori terdistribusi. Mantan penggunaan beberapa prosesor yang berbagi bus tunggal dan memori sub-sistem. Yang terakhir adalah sistem terdistribusi dalam arti sistem yang kita bicarakan di sini dan menggunakan node komputasi independen terhubung melalui jaringan (misalnya, multicomputer a). Meskipun janji peningkatan kinerja, pemrograman paralel tetap sulit dan jika perawatan tidak diambil kinerja mungkin berakhir menurun daripada meningkatkan.



## **Sistem terdistribusi dalam konteks**

Studi tentang sistem terdistribusi berkaitan erat dengan dua bidang lainnya, Jaringan dan Sistem Operasi. Hubungan ke jaringan harus cukup jelas, sistem terdistribusi mengandalkan jaringan untuk menghubungkan komputer individu bersama-sama. Ada garis tipis dan kabur antara ketika seseorang berbicara tentang pengembangan jaringan dan mengembangkan sistem terdistribusi. Seperti yang akan kita bahas nanti pembangunan (dan studi) sistem terdistribusi kekhawatiran itu sendiri dengan isu-isu yang timbul ketika sistem yang dibangun dari komponen jaringan yang saling berhubungan, bukan rincian komunikasi dan jaringan protokol.

Hubungan dengan sistem operasi mungkin kurang jelas. Untuk membuat generalisasi yang luas sistem operasi bertanggung jawab untuk mengelola sumber daya dari sebuah sistem komputer, dan menyediakan akses ke sumber daya tersebut secara independen aplikasi (dan berurusan dengan isu-isu seperti sinkronisasi, keamanan, dll yang muncul). Studi tentang sistem terdistribusi dapat dilihat sebagai mencoba untuk menyediakan jenis yang sama akses umum terhadap sumber daya didistribusikan (dan juga berurusan dengan isu-isu yang muncul).

## **D. Keuntungan dan Kerugian Jaringan Terdistribusi**

### **Keuntungan :**

#### **a. Performance**

Kumpulan dari beberapa prosesor akan memberikan kinerja yang lebih baik dari pada komputer yang terpusat. Begitu juga kalau dilihat dari sisi biaya.

#### **b. Distribution**

#### **c. Reliability (Fault tolerance)**

Apabila salah satu komponen terjadi kerusakan, system tetap dapat berjalan

#### **d. Incremental Growth**

Mudah dalam melakukan penambahan komputer/komponen

#### **e. Sharing Data/Resources**

Berbagi data adalah salah satu hal yang pokok pada kebanyakan aplikasi

### **Kelemahan :**

#### **a. Kesulitan dalam membangun perangkat lunak.**

Kesulitan yang akan dihadapi antara lain : bahasa pemrograman yang harus dipakai, sistem operasi dll.

b. Masalah Jaringan

Karena sistem terdistribusi di implementasikan dalam jaringan komputer, maka isu2 yang berkaitan dengan jaringan komputer akan menjadi pertimbangan utamadalam merancang dan mengimplementasikan sistem.

c. Masalah Keamanan

Karena pada sistem terdistribusi berbagi data/sumber daya merupakan hal yang mutlak maka muncul masalah2 yang berkaitan dengan keamanan data dll.

**E. Hal-hal yang harus diperhatikan dalam membangun sebuah system jaringan terdistribusi**

Karakteristik yang harus diperhatikan adalah :

a. Transparency (Kejelasan)

1) Access transparency

Sumber daya lokal dan remote di akses dengan menggunakan operasi yang sama.

2) Location transparency

Pengguna sistem tidak tahu mengetahui keberadaan hardware dan software (CPU, file dan data).

3) Migration (Mobility) transparency

Sumber daya (baik berupa Hardware dan/atau software) dapat bebas berpindah tanpa mengubah sistem penamaan.

4) Replication transparency

Sistem bebas untuk menambah file atau sumber daya tanpa diketahui oleh user (dalam rangkan meningkatkan kinerja)

5) Concurency transparency

User tidak akan mengetahui keberadaan user lain dalam sistem, walaupun user tersebut menggunakan sumber daya yang sama.

6) Failure transparency

Aplikasi harus dapat menyelesaikan proses nya walaupun terdapat kegagalan pada beberapa pada komponen sistem.

7) Performance transparency

Beban kerja yang bervariasi tidak akan menyebabkan turunnya kinerja sistem, hal ini dapat di capai dengan melakukan otomatisasi konfigurasi terhadap perubahan beban.

b. Communication (Komunikasi)

Sistem melakukan urutan komunikasi. Beberapa hal yang harus diperhatikan adalah:

- infrastruktur jaringan (interkoneksi dan software jaringan)
- Metode dan Model komunikasi yang cocok. Metode komunikasi :
  - ☐ Send
  - ☐ Receive
  - ☐ Remote Procedure Call
- Model Komunikasi
  - ☐ client - server communication : pertukaran pesan antara dua proses dimana satu proses (client) menggunakan / meminta layanan pada server dan server menyediakan hasil dari proses tersebut.
  - ☐ groupmulticast : target dari pesan yang dikirimkan adalah gabungan dari proses, yang berasal dari suatu grup.

c. Performance dan scalability

- Performance:

Ada beberapa faktor yang mempengaruhi kinerja (performance) dari pada sistem terdistribusi :

  - ☐ Kinerja dari pada personal workstations
  - ☐ Kecepatan infrastruktur komunikasi
  - ☐ Fleksibilitas dalam membagi beban kerja : contoh, apabila terdapat prosesor (workstation) yang idle maka dapat di alokasikan secara otomatis untuk mengerjakan tugas2 user.
- Scalability

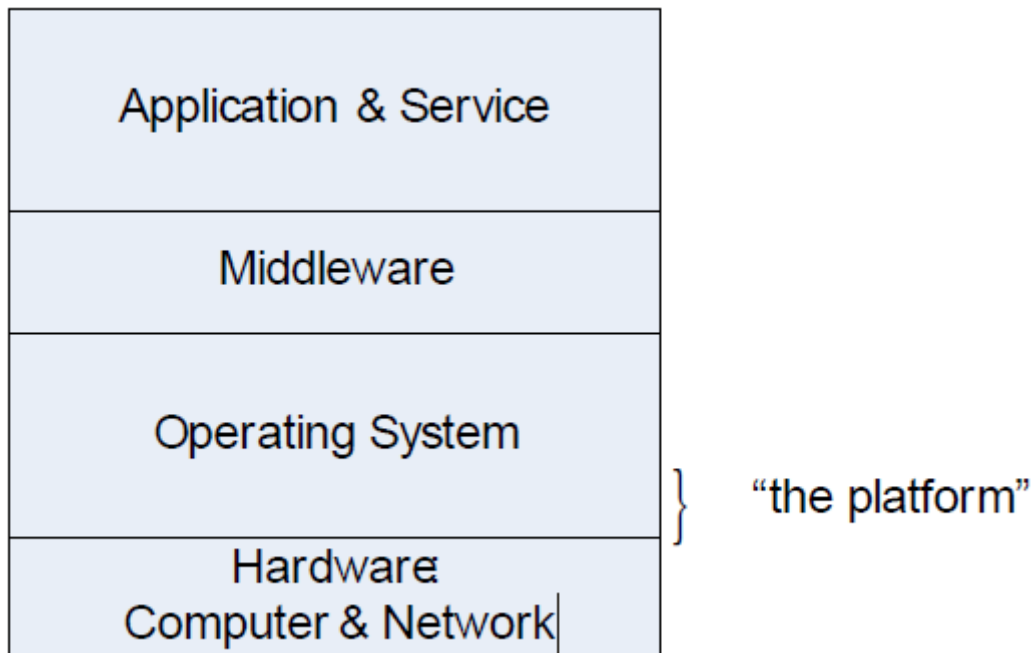
Sistem tetap harus memperhatikan efisiensi walaupun terdapat penambahan secara signifikan user atau sumber daya yang terhubung:

  - ☐ 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.
- Untuk mendukung keanekaragaman maka arsitektur perangkat lunak sistem jaringan terdistribusi adalah:



e. Openess (Keterbukaan)

- Setiap layanan (services) harus dapat di akses oleh semua user.
- Mudah dalam implementasi, install dan debug services;
- User dapat membuat dan menginstall service
- Aspek kunci pada opennes :
  - ☐ Interface dan Protocol yang standard (seperti protokol komunikasi diinternet)
  - ☐ Support terhadap keanekaragaman (dengan membuat midleware (ORB) seperti CORBA atau Java RMI)

f. Reliability

Salah satu tujuan dalam membangun system terdistribusi adalah memungkinkan untuk melakukan improvisasi terhadap kehandalan sistem.

- Availability : kalau mesin mati (down), sistem tetap harus berjalan dengan jumlah layanan yang tersisa.
- Dalam sistem terdistribusi componen yang sangat vital (critical resources) berjumlah se minimal mungkin. Yang dimaksud dengan critical resources adalah komponen yang harus ada untuk menjalankan sistem terdistribusi.
- Masing - masing Software dan Hardware harus di replikasi : kalau terjadi kegagalan / error maka yang lain akan menangani.
- Data dalam sistem tidak boleh hilang, copy dari file tersebut disimpan secara redundan pada server lain, tapi tetap harus dijaga konsistensi datanya.

g. Fault Tolerance

Fault Tolerance : Sistem harus bias 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 di rencanakan dan memungkinkan untuk menghentikan proses dalam waktu yang singkat tanpa menghilangkan informasi atau data.

h. Security (Keamanan)

- Confidentiality :keamanan terhadap data yang di akses oleh user yang tidak di perbolehkan (unauthorizes user)
- Integrity: keamanan terhadap kelengkapan dan autentikasi data.
- Availability: Menjaga agar resource dapat selalu di akses.
- Antisipasi terhadap free access (penggunaan resource terhadap user yang semestinya)