

PROSES

PERT 3.

Sistem Terdistribusi

Konsep Proses

- Proses : suatu program yang sedang dieksekusi. Eksekusi proses dilakukan secara berurutan
- Dalam proses terdapat
 - ▣ Program counter : menunjukkan instruksi berikutnya yang akan dieksekusi
 - ▣ Stack : tempat penyimpanan temporary data yang dibutuhkan selama program dieksekusi
 - ▣ Data section : mengandung variabel global dari suatu proses
- Sistem operasi mengeksekusi berbagai jenis program.

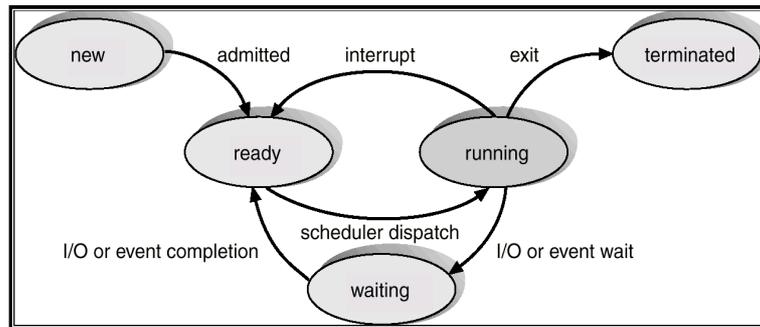
Perkembangan Konsep Proses

- Konsep proses berkembang sejalan dengan modus operasi sistem komputer :
 - ▣ Fase-fase monoprogramming : pada suatu saat di memori hanya sebuah program yang sedang dijalankan (mungkin disertai suatu monitor proses)
 - ▣ Fase-fase multiprogramming : pada suatu saat di memori bisa terdapat beberapa program yang sedang dijalankan

Status Proses

- Selama eksekusinya proses berada dalam salah satu dari status-status berikut ini :
 - ▣ **New** : proses sedang di-create
 - ▣ **Running** : instruksi-instruksi sedang dieksekusi
 - ▣ **Waiting** : proses sedang menunggu terjadinya event-event tertentu seperti untuk menunggu untuk menyelesaikan I/O
 - ▣ **Ready** : proses menunggu di-assign ke prosesor
 - ▣ **Terminated** : proses selesai eksekusi

Diagram Status Proses



Process Control Block (PCB)

- Agar suatu proses yang running, lalu waiting/ready, dan kemudian running kembali dapat meneruskan proses tanpa kehilangan konteks maka diperlukan struktur data untuk menyimpan informasi dan status proses, disebut Process Control Block (PCB)
- Satu blok digunakan untuk satu proses dan isinya bisa bervariasi dari proses ke proses

Thread

- *Thread adalah sebuah alur kontrol dari sebuah proses.*
- *Kontrol thread tunggal ini hanya memungkinkan proses untuk menjalankan satu tugas pada satu waktu.*
- *Sistem operasi modern telah memiliki konsep yang dikembangkan agar memungkinkan sebuah proses untuk memiliki eksekusi multi-threads*

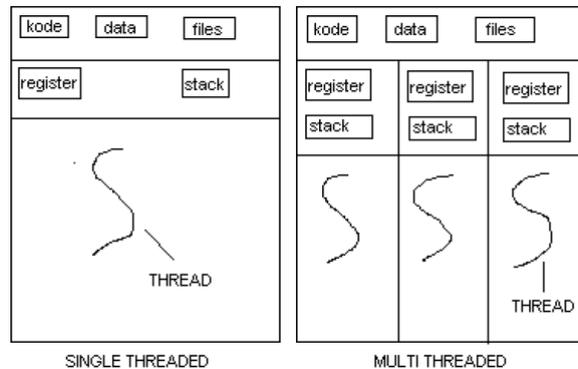
Keuntungan dari *multithreaded*

- *Meliputi peningkatan respon dari pengguna,*
- *Pembagian sumber daya proses,*
- *Ekonomis, dan kemampuan untuk mengambil keuntungan dari arsitektur multiprosesor.*

- *Thread merupakan unit dasar dari penggunaan CPU, yang terdiri dari Thread_ID, program counter, register set, dan stack.*

Perbedaan Thread

- Perbedaan antara proses dengan *thread tunggal* dan proses dengan *thread yang banyak* adalah: proses dengan *thread banyak* dapat mengerjakan lebih dari satu tugas pada satu satuan waktu.



Multi-threading

- Perangkat lunak pada PC modern dirancang secara *multi-threading*
- Contohnya sebuah *web browser* mempunyai *thread* untuk menampilkan gambar atau tulisan sedangkan *thread* yang lain berfungsi sebagai penerima data dari network.
- Kadang kala ada situasi dimana sebuah aplikasi diperlukan untuk menjalankan beberapa tugas yang serupa. Sebagai contohnya sebuah *web server* dapat mempunyai ratusan klien yang mengaksesnya secara *concurrent*.
- Kalau *web server* berjalan sebagai proses yang hanya mempunyai *thread tunggal* maka ia hanya dapat melayani satu klien pada satu satuan waktu. Bila ada klien lain yang ingin mengajukan permintaan maka ia harus menunggu sampai klien sebelumnya selesai dilayani. Solusinya adalah dengan membuat *web server* menjadi *multi-threading*.

Keuntungan Multi-*Threading* [1]

1. Responsi: Membuat aplikasi yang interaktif menjadi *multithreading* dapat membuat sebuah program terus berjalan meskipun sebagian dari program tersebut diblok atau melakukan operasi yang panjang, karena itu dapat meningkatkan respons kepada pengguna. Sebagai contohnya dalam web browser yang *multithreading*, sebuah thread dapat melayani permintaan pengguna sementara thread lain berusaha menampilkan image.

Keuntungan Multi-*Threading* [2]

- 2. Berbagi sumber daya:** thread berbagi memori dan sumber daya dengan thread lain yang dimiliki oleh proses yang sama. Keuntungan dari berbagi kode adalah mengizinkan sebuah aplikasi untuk mempunyai beberapa thread yang berbeda dalam lokasi memori yang sama.
- 3. Ekonomi:** dalam pembuatan sebuah proses banyak dibutuhkan pengalokasian memori dan sumber daya. Alternatifnya adalah dengan penggunaan thread, karena thread berbagi memori dan sumber daya proses yang memilikinya maka akan lebih ekonomis untuk membuat dan context switch thread.

Keuntungan Multi-*Threading* [3]

4. Utilisasi arsitektur *multiprocessor*: Keuntungan dari *multithreading* dapat sangat meningkat pada arsitektur *multiprocessor*, dimana setiap *thread* dapat berjalan secara paralel di atas processor yang berbeda. Pada arsitektur processor tunggal, CPU menjalankan setiap *thread* secara bergantian tetapi hal ini berlangsung sangat cepat sehingga menciptakan ilusi paralel, tetapi pada kenyataannya hanya satu *thread* yang dijalankan CPU pada satu-satuan waktu (satu-satuan waktu pada CPU biasa disebut *time slice* atau *quantum*).

Client – Server

- Pada sistem *client/server* harus terdapat satu atau beberapa server yang menyediakan layanan dan satu atau beberapa klien yang meminta layanan tersebut (tidak peduli apakah kondisi tersebut berada pada sebuah sistem jaringan ataupun *stand-alone*).
- Istilah server di sini bisa saja berupa komputer-komputer kelas server seperti IBM, HP, Compaq dll. Atau juga berupa software yang dapat dikategorikan berdasarkan layanannya misalnya *web server*, *application server*, *file server*, *database server*, *terminal server*, *mail server*, dll.

Perbedaan server dan client

- *Server adalah komputer yang dapat memberikan service ke client, sedangkan client adalah komputer yang mengakses beberapa service yang ada di server.*

- *Ketika client membutuhkan suatu service yang ada di server, dia akan mengirim request kepada server lewat jaringan. Jika request tersebut dapat dilaksanakan, maka server akan mengirim balasan berupa service yang dibutuhkan untuk saling berhubungan menggunakan Socket.*

Karakteristik

1. Karakteristik *Server*
 - a. Pasif
 - b. Menunggu *request*
 - c. Menerima *request*, memproses mereka dan mengirimkan balasan berupa *service*

2. Karakteristik *Client*
 - a. Aktif
 - b. Mengirim *request*
 - c. Menunggu dan menerima balasan dari *server*

Socket

- *Socket adalah sebuah endpoint untuk komunikasi didalam jaringan.*
- *Sepasang proses atau thread berkomunikasi dengan membangun sepasang socket, yang masing-masing proses memilikinya.*
- *Socket dibuat dengan menyambungkan dua buah alamat IP melalui port tertentu.*
- *Secara umum socket digunakan dalam client/server system, dimana sebuah server akan menunggu client pada port tertentu. Begitu ada client yang menghubungi server maka server akan menyetujui komunikasi dengan client melalui socket yang dibangun.*

Model Client-Server

□ a. **Arsitektur Mainframe**

Pada arsitektur ini, terdapat sebuah komputer pusat (host) yang memiliki sumber daya yang sangat besar, baik memori, processor maupun media penyimpanan. Melalui komputer terminal, pengguna mengakses sumber daya tersebut. Komputer terminal hanya memiliki monitor/keyboard dan tidak memiliki CPU. Semua sumber daya yang diperlukan terminal dilayani oleh komputer host. Model ini berkembang pada akhir tahun 1980-an.

Model Client-Server

b. Arsitektur File Sharing Pada arsitektur ini komputer server menyediakan file-file yang tersimpan di media penyimpanan server yang dapat diakses oleh pengguna.

Arsitektur file sharing memiliki keterbatasan, terutama jika jumlah pengakses semakin banyak serta ukuran file yang di sharing sangat besar. Hal ini dapat mengakibatkan transfer data menjadi lambat. Model ini populer pada tahun 1990-an.

Model Client-Server

c. Arsitektur Client/Server Karena keterbatasan sistem file sharing, dikembangkanlah arsitektur client/server. Salah satu hasilnya yaitu berupa software database server yang menggantikan software database berbasis file server. Dikenalkan pula RDBMS (Relational Database Management System). Dengan arsitektur ini, query data ke server dapat terlayani dengan lebih cepat karena yang ditransfer bukanlah file, tetapi hanyalah hasil dari query tersebut.

RPC (Remote Procedure Calls) memegang peranan penting pada arsitektur client/server

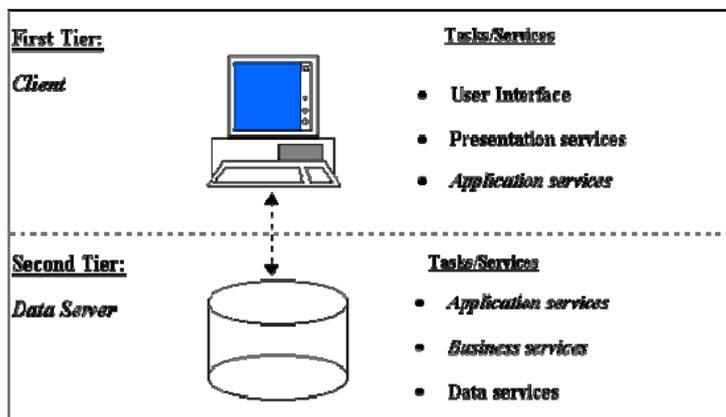
Model Two-tier

d. Model Two-tier : terdiri dari tiga komponen yang disusun menjadi dua lapisan: *client* (yang meminta service) dan *server* (yang menyediakan service). Tiga komponen tersebut yaitu :

- ▣ 1. User Interface, yaitu antar muka program aplikasi yang berhadapan dan digunakan langsung oleh user.
- ▣ 2. Manajemen proses
- ▣ 3. Database

Model Two-tier

- ▣ Model ini memisahkan peranan user interface dan database dengan jelas, sehingga terbentuk dua lapisan.



Model Two-tier

- Pada gambar tersebut, user interface yang merupakan bagian dari program aplikasi melayani input dari user. Input tersebut diproses oleh Manajemen Proses dan melakukan query data ke database (dalam bentuk perintah SQL). Pada database server juga bisa memiliki Manajemen Proses untuk melayani query tersebut, biasanya ditulis ke dalam bentuk Stored Procedure.

Model Three-tier

e. Model Three-tier

- Pada model ini disisipkan satu layer tambahan diantara user interface tier dan database tier. Tier tersebut dinamakan middle-tier.
- Middle-Tier terdiri dari bussiness logic dan rules yang menjembatani query user dan database, sehingga program aplikasi tidak bisa mengquery langsung ke database server, tetapi harus memanggil prosedur-prosedur yang telah dibuat dan disimpan pada middle-tier.
- Dengan adanya server middle-tier ini, beban database server berkurang.
- Jika query semakin banyak dan/atau jumlah pengguna bertambah, maka server-server ini dapat ditambah, tanpa merubah struktur yang sudah ada.

