

# Temu-13

Pipelining

**Ali Tarmuji, S.T., M.Cs.**

[alitarmuji@tif.uad.ac.id](mailto:alitarmuji@tif.uad.ac.id)

---

# PARALEL PROCESSING & PIPELINING

# PARALEL PROCESSING

- Paralel processing/ Pemrosesan Paralel adalah komputasi dua atau lebih tugas pada waktu bersamaan dengan tujuan untuk mempersingkat waktu penyelesaian tugas-tugas tersebut
- Caranya dengan mengoptimalkan resource pada sistem komputer yang ada untuk mencapai tujuan yang sama.
- Pemrosesan paralel dapat mempersingkat waktu eksekusi suatu program dengan cara membagi suatu program menjadi bagian-bagian yang lebih kecil yang dapat dikerjakan pada masing-masing prosesor secara bersamaan.

# PARALEL PROCESSING

- Tujuan utama dari pemrosesan paralel adalah untuk meningkatkan performa komputasi.
- Semakin banyak hal yang bisa dilakukan secara bersamaan (dalam waktu yang sama), semakin banyak pekerjaan yang bisa diselesaikan.
- Analogi yang paling gampang adalah, bila kita dapat merebus air sambil memotong-motong bawang saat akan memasak, waktu yang dibutuhkan akan lebih sedikit dibandingkan bila mengerjakan hal tersebut secara berurutan (serial). Atau waktu yg dibutuhkan memotong bawang akan lebih sedikit jika dikerjakan berdua/paralel/bersamaan

# PARALEL PROCESSING

- Performa dalam pemrosesan paralel diukur dari berapa banyak peningkatan kecepatan (speed up) yang diperoleh dalam menggunakan teknik paralel.
- Secara informal, bila anda memotong bawang sendirian membutuhkan waktu 1 jam dan dengan bantuan teman, berdua bisa melakukannya dalam 1/2 jam maka memperoleh peningkatan kecepatan sebanyak 2 kali.

# PARALEL PROCESSING

- Adapun proses kerja , pemrosesan paralel membagi beban kerja dan mendistribusikannya pada komputer-komputer lain yang terdapat dalam sistem untuk menyelesaikan suatu masalah.
- Sistem yang akan dibangun akan tidak akan menggunakan komputer yang didedikasikan secara khusus untuk keperluan pemrosesan paralel melainkan menggunakan komputer yang telah ada.
- Artinya, sistem ini nantinya akan terdiri dari sejumlah komputer dengan spesifikasi berbeda yang akan bekerjasama untuk menyelesaikan suatu masalah

# Tipe-tipe Parallelisme

## 1. Tipe *Result Parallelisme/ Embarrassingly Parallel/ Perfect Parallel*

- adalah tipe parallelisme dimana komputasinya dapat dibagi menjadi beberapa tugas independen yang mempunyai struktur sama.
- Data struktur suatu tugas dibagi menjadi beberapa bagian yang berstruktur sama.
- Contoh tugas yang bisa diselesaikan dengan Result Parallelism adalah Simulasi Montecarlo.

# Tipe-tipe Paralelisme

## 2. Specialist Paralelisme

- Cara kerjanya dengan mengerjakan beberapa tugas secara bersamaan pada prosesor yang berbeda.
- Setiap komputer mengerjakan tugas tertentu.
- Contohnya penggunaannya adalah pada simulasi pabrik kimia, satu prosesor mensimulasikan proses sebelum reaksi kimia, satu prosesor mensimulasikan reaksi pada tahap awal, dan prosesor lainnya mensimulasikan proses penyulingan hasil, dan seterusnya.



# Tipe-tipe Paralelisme

## 3. Agenda Paralelisme

- Tipe paralelisme ini mempunyai daftar yang harus dikerjakan oleh sistem komputer.
- Semua komputer yang terdapat pada sistem dapat mengakses daftar tersebut.
- Pada Model MW (Manager Worker) terdapat pengelompokan komputer menjadi dua yaitu :
  - Manager : bertugas memulai perhitungan, memonitor kemajuan tugas, melayani permintaan worker. User berkomunikasi dengan sistem komputer melalui komputer yang berfungsi sebagai manager ini.
  - Worker : mengerjakan tugastugas yang diberikan oleh manager. Kerja komputer ini dimulai setelah ada perintah dari manager dan diakhiri oleh manager. Pesan Terdistribusi dan Lingkungan Pemrograman (Distributed Messaging and Programming Enviroment) hubungannya dengan pemrosesan paralel ialah pesan sebagai sesuatu ( objek ) pada pemrosesan itu sendiri sehingga kita harus mendeskripsikan pesan ( message ) itu .

# PIPELINING

- Pipeline adalah suatu cara yang digunakan untuk melakukan sejumlah kerja secara bersama tetapi dalam tahap yang berbeda yang dialirkan secara kontinu pada unit pemrosesan. Dengan cara ini, maka unit pemrosesan selalu bekerja
- Teknik pipeline ini dapat diterapkan pada berbagai tingkatan dalam sistem komputer. Bisa pada level yang tinggi, misalnya program aplikasi, sampai pada tingkat yang rendah, seperti pada instruksi yang dijaankan oleh microprocessor

- Pada microprocessor yang tidak menggunakan pipeline, satu instruksi dilakukan sampai selesai, baru instruksi berikutnya dapat dilaksanakan. Sedangkan dalam microprocessor yang menggunakan teknik pipeline, ketika satu instruksi sedang diproses, maka instruksi yang berikutnya juga dapat diproses dalam waktu yang bersamaan. Tetapi, instruksi yang diproses secara bersamaan ini, ada dalam tahap proses yang berbeda. Jadi, ada sejumlah tahapan yang akan dilewati oleh sebuah instruksi.

- Dengan penerapan pipeline ini pada microprocessor akan didapatkan peningkatan dalam unjuk kerja microprocessor. Hal ini terjadi karena beberapa instruksi dapat dilakukan secara parallel dalam waktu yang bersamaan. Secara kasarnya diharapkan akan didapatkan peningkatan sebesar  $X$  kali dibandingkan dengan microprocessor yang tidak menggunakan pipeline, apabila tahapan yang ada dalam satu kali pemrosesan instruksi adalah  $X$  tahap.

- Karena beberapa instruksi diproses secara bersamaan ada kemungkinan instruksi tersebut samasama memerlukan resource yang sama, sehingga diperlukan adanya pengaturan yang tepat agar proses tetap berjalan dengan benar. Sedangkan ketergantungan terhadap data, bisa muncul, misalnya instruksi yang berurutan memerlukan data dari instruksi yang sebelumnya. Kasus Jump, juga perlu perhatian, karena ketika sebuah instruksi meminta untuk melompat ke suatu lokasi memori tertentu, akan terjadi perubahan program counter, sedangkan instruksi yang sedang berada dalam salah satu tahap proses yang berikutnya mungkin tidak mengharapkan terjadinya perubahan program counter.

- Teknik pipeline yang diterapkan pada microprocessor, dapat dikatakan sebuah arsitektur khusus. Ada perbedaan khusus antara model microprocessor yang tidak menggunakan arsitektur pipeline dengan microprocessor yang menerapkan teknik ini. Pada microprocessor yang tidak menggunakan pipeline, satu instruksi dilakukan sampai selesai, baru instruksi berikutnya dapat dilaksanakan. Sedangkan dalam microprocessor yang menggunakan teknik pipeline, ketika satu instruksi sedang diproses, maka instruksi yang berikutnya juga dapat diproses dalam waktu yang bersamaan. Tetapi, instruksi yang diproses secara bersamaan ini, ada dalam tahap proses yang berbeda

- Jadi, ada sejumlah tahapan yang akan dilewati oleh sebuah instruksi.

Misalnya sebuah microprocessor menyelesaikan sebuah instruksi dalam 4 langkah. Ketika instruksi pertama masuk ke langkah 2, maka instruksi berikutnya diambil untuk diproses pada langkah 1 instruksi tersebut. Begitu seterusnya, ketika instruksi pertama masuk ke langkah 3, instruksi kedua masuk ke langkah 2 dan instruksi ketiga masuk ke langkah 1.

- Kenapa komputer menggunakan teknik Pipelining??  
Drive for computing speed never ends.  
Improvements from architecture or organization point of view are limited  
Clock speed enhancement is done, but more improvement should be sought from  
instruction execution perspective, instead of hardware design  
Flynn's Taxonomy : SISD (Single Instruction Single stream of Data), SIMD (Single Instruction Multiple stream of data) or MIMD – Parallel  
Parallel Processor : may be a solution  
Use two processors (or more, instead of one) in a computer system  
How do it runs the code ? (program)  
Suppose, we have a problem :  
 $C = (A^2 + B^2)$



# Instruksi Pipeline

- Tahapan pipeline

Mengambil instruksi dan membufferkannya  
Ketika tahapan kedua bebas tahapan pertama mengirimkan instruksi yang dibufferkan tersebut

Pada saat tahapan kedua sedang mengeksekusi instruksi, tahapan pertama memanfaatkan siklus memori yang tidak dipakai untuk mengambil dan membufferkan instruksi berikutnya