

# Organisasi dan arsitektur komputer

Pertemuan 13:  
Pepirline Dan RISC

# PIPELINE

- ▶ Pipeline:

Mesin yang melaksanakan beberapa komputasi yang berbeda secara bersamaan, namun pada saat itu setiap komputasi akan berada dalam tahapan eksekusi yang berbeda

# KATEGORI PIPELINE

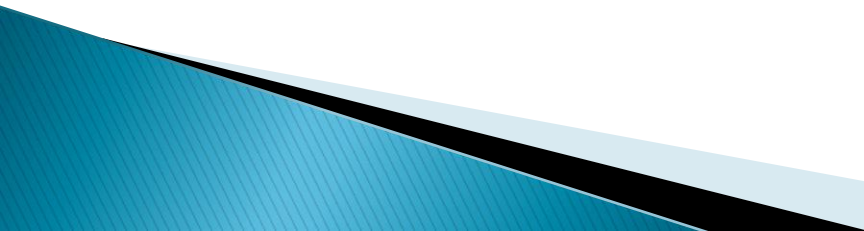
## 1. Pipeline Unit Arithmetic

Berguna untuk operasi vektor

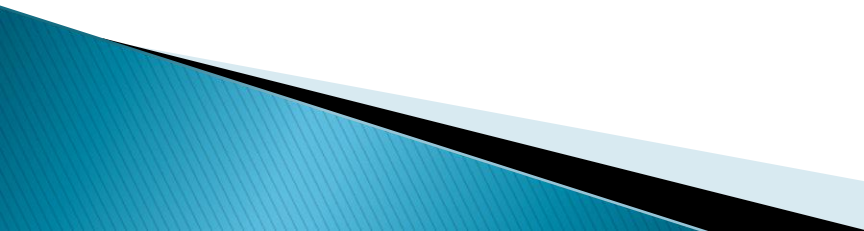
## 2. Pipeline Unit Instruction

Berguna untuk komputer yang mempunyai set instruksi yang sederhana

# TEKNIK DASAR YANG DIGUNAKAN DALAM MERANCANG SUPERKOMPUTER DAN RISC

1. Pendekodean instruksi pipelined
  2. Beberapa unit fungsional pipelined yang beroperasi secara bersamaan
  3. Bank memori interleaved tak sinkron
  4. Cache instruksi dan data independen
  5. Sejumlah bus untuk mentransfer data, alamat dan signal kontrol
- 

# PEMROSESAN VEKTOR LEBIH CEPAT DARIPADA PEMROSESAN SKALAR

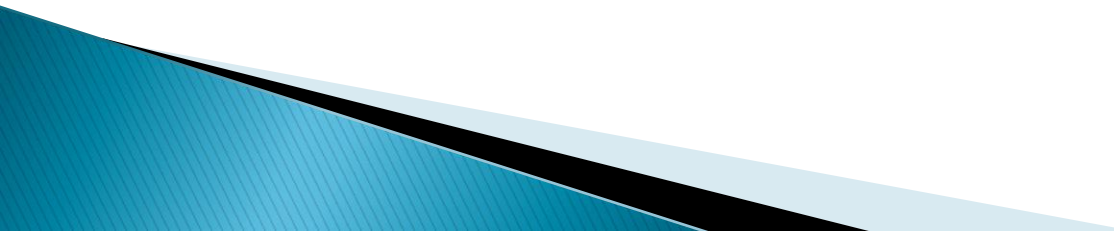
1. Berkurangnya kontensi memori karena adanya akses memori yang lebih sedikit
  2. Berkurangnya pendekodean instruksi
  3. Tingkah lakunya bisa diramalkan, hal ini khususnya penting bagi:
    - Pengindeksan implisit dan akses memori
    - Pencabangan implisit
- 

# REDUCED INSTRUCTION SET COMPUTER (RISC)

Fitur Utama:

- ▶ General Purpose Register dalam jumlah yang amat banyak
- ▶ Menggunakan teknologi compiler untuk mengoptimalkan penggunaan register
- ▶ Instruction Set yang sedikit dan sederhana
- ▶ Pendekatan umum dalam instruksi pipeline
- ▶ Memimpin untuk:  
Set eksekusi yang besar dan lebih banyak mode pengalamatan

# KARAKTERISTIK RISC

- ▶ Satu instruksi persiklus
  - ▶ Operasi register to register
  - ▶ Mode pengalamatan yang sederhana
  - ▶ Format instruksi yang sederhana
  - ▶ Desain hardwired (tanpa microcode)
  - ▶ Format instruksi yang fix
  - ▶ Proses compile yang cepat
- 

# PIPELINING PADA RISC

- Terdapat berbagai macam instruksi pada register to register
- Siklus Instruksi memiliki 2 Fase:
  1. I : Instruction Fetch (Pengambilan Instruksi)
  2. E : Execute (Melakukan operasi ALU dengan register input dan output)
- Operasi Load dan Store memiliki 3 Fase:
  1. I : Instruction Fetch
  2. E : Execute (Menghitung alamat memori)
  3. D : Memory (Operasi register ke memori atau memori ke register)



# Effects of Pipelining

Load  $rA \leftarrow M$   
 Load  $rB \leftarrow M$   
 Add  $rC \leftarrow rA + rB$   
 Store  $M \leftarrow rC$   
 Branch X

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| I | E | D |   |   |   |   |   |   |   |
|   |   |   | I | E | D |   |   |   |   |
|   |   |   |   |   |   | I | E |   |   |
|   |   |   |   |   |   |   | I | E | D |
|   |   |   |   |   |   |   |   | I | E |

(a) Sequential execution

Load  $rA \leftarrow M$   
 Load  $rB \leftarrow M$   
 Add  $rC \leftarrow rA + rB$   
 Store  $M \leftarrow rC$   
 Branch X  
 NOOP

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| I | E | D |   |   |   |   |   |   |   |
|   | I |   | E | D |   |   |   |   |   |
|   |   |   | I |   | E |   |   |   |   |
|   |   |   |   |   |   | I | E | D |   |
|   |   |   |   |   |   |   | I |   | E |
|   |   |   |   |   |   |   |   | I | E |

(b) Two-stage pipelined timing

Load  $rA \leftarrow M$   
 Load  $rB \leftarrow M$   
 NOOP  
 Add  $rC \leftarrow rA + rB$   
 Store  $M \leftarrow rC$   
 Branch X  
 NOOP

|   |   |   |   |   |   |   |   |  |  |
|---|---|---|---|---|---|---|---|--|--|
| I | E | D |   |   |   |   |   |  |  |
|   | I | E | D |   |   |   |   |  |  |
|   |   | I | E |   |   |   |   |  |  |
|   |   |   | I | E |   |   |   |  |  |
|   |   |   |   | I | E | D |   |  |  |
|   |   |   |   |   | I | E |   |  |  |
|   |   |   |   |   |   | I | E |  |  |

(c) Three-stage pipelined timing

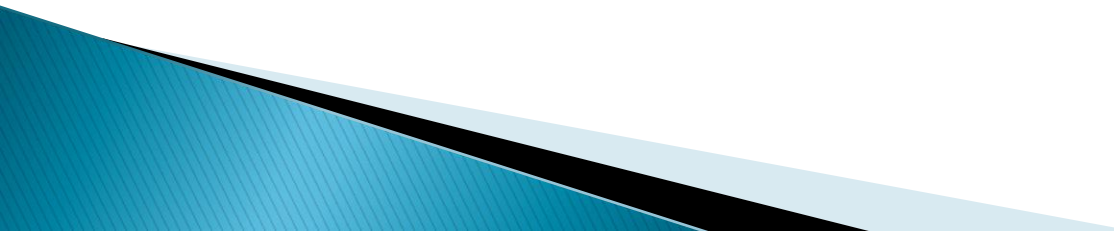
Load  $rA \leftarrow M$   
 Load  $rB \leftarrow M$   
 NOOP  
 NOOP  
 Add  $rC \leftarrow rA + rB$   
 Store  $M \leftarrow rC$   
 Branch X  
 NOOP  
 NOOP

|   |                |                |                |                |                |                |                |                |                |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| I | E <sub>1</sub> | E <sub>2</sub> | D              |                |                |                |                |                |                |
|   | I              | E <sub>1</sub> | E <sub>2</sub> | D              |                |                |                |                |                |
|   |                | I              | E <sub>1</sub> | E <sub>2</sub> |                |                |                |                |                |
|   |                |                | I              | E <sub>1</sub> | E <sub>2</sub> |                |                |                |                |
|   |                |                |                | I              | E <sub>1</sub> | E <sub>2</sub> | D              |                |                |
|   |                |                |                |                | I              | E <sub>1</sub> | E <sub>2</sub> |                |                |
|   |                |                |                |                |                | I              | E <sub>1</sub> | E <sub>2</sub> |                |
|   |                |                |                |                |                |                | I              | E <sub>1</sub> | E <sub>2</sub> |

(d) Four-stage pipelined timing

# OPTIMALISASI PIPELINE

Percabangan yang dikenal Delay Branch

1. Tidak akan ada efeknya sampai suatu eksekusi instruksi selesai
  2. Instruksi percabangan akhirnya mengalami delay
- 

# Comparison of processors

|   | Complex Instruction Set<br>(CISC) Computer |               |                | Reduced Instruction<br>Set (RISC) Computer |               | Superscalar |                |                |
|---|--|---------------|----------------|--|---------------|-------------|----------------|----------------|
| <b>Characteristic</b>                           | IBM<br>370/168                             | VAX<br>11/780 | Intel<br>80486 | SPARC                                      | MIPS<br>R4000 | PowerPC     | Ultra<br>SPARC | MIPS<br>R10000 |
| <b>Year developed</b>                           | 1973                                       | 1978          | 1989           | 1987                                       | 1991          | 1993        | 1996           | 1996           |
| <b>Number of<br/>instructions</b>               | 208  | 303           | 235            | 69   | 94            | 225         |                |                |
| <b>Instruction size (bytes)</b>                 | 2-6  | 2-57          | 1-11           | 4  | 4             | 4           | 4              | 4              |
| <b>Addressing modes</b>                         | 4  | 22            | 11             | 1  | 1             | 2           | 1              | 1              |
| <b>Number of general-<br/>purpose registers</b> | 16   | 16            | 8              | 40 - 520                                   | 32            | 32          | 40 - 520       | 32             |
| <b>Control memory size<br/>(Kbits)</b>          | 420  | 480           | 246            | —  | —             | —           | —              | —              |
| <b>Cache size (KBytes)</b>                      | 64   | 64            | 8              | 32   | 128           | 16-32       | 32             | 64             |