

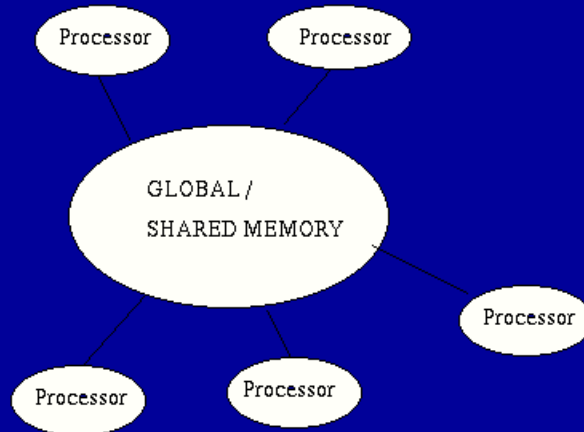
Dasar Komunikasi Antar Prosesor (Interprocessor Communication)

Kudang B. Seminar

Bagaimana Prosesor Saling Berkomunikasi Data?

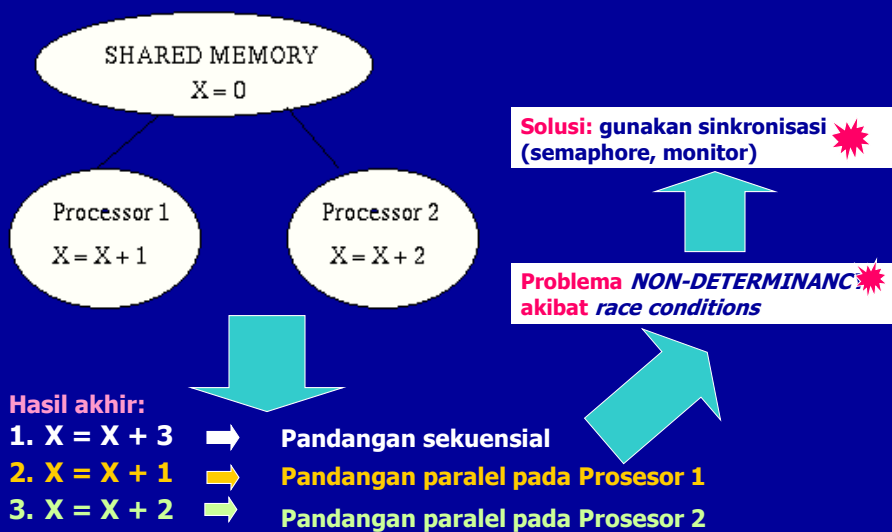
- Menggunakan **memori persekutuan** (*shared memory*) atau **variabel persekutuan** (*shared variable*)
- Menggunakan **pertukaran pesan** (*message passing*)

Komunikasi Dengan Memori Persekutuan



Problema: akses simultan pada alamat memori yang sama!

Ilustrasi Problema Akses Simultan



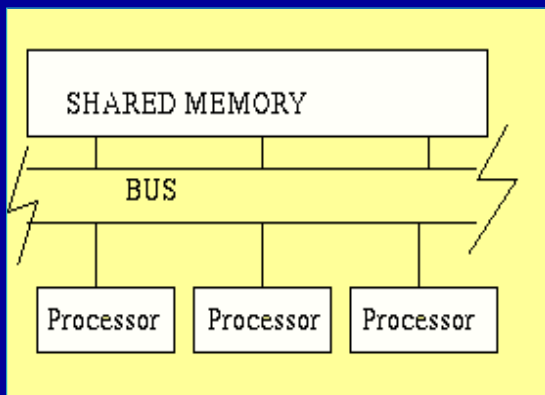
Kategori komputer *shared memori*

- Exclusive Read, Exclusive Write (EREW)
- Concurrent Read, Exclusive Write (CREW)
- Exclusive Read, Concurrent Write (ERCW)
- Concurrent Read, Concurrent Write (CRCW)

Contoh Komputer dengan Memori

Persekutuan: ***Sequent & Encore***

Menggunakan bus berkecepatan tinggi yang menghubungkan prosesor dan memori



Bandwidth dari bus menjadi pembatas

Jumlah prosesor yg besar menimbulkan masalah kompetisi penggunaan bus (*bus contention*)

Pilihan implementasi memori persekutuan

- Centralized Memory
- Distributed Memory

	Centralized	Distributed
Jumlah prosesor	Besar (100 – 1000)	Sedang (10 – 100)
Kemampuan teoritis	Tinggi	Sedang
Ekspansi	Tak terbatas	Terbatas
Level utilisasi	Sulit	Lebih mudah
Paradigma	Revolutionay parallel programming	Evolutionay parallel programming

Penjumlahan sejumlah m bilangan menggunakan sebanyak N prosesor

Komputer sekuensial

```
sum := A0
FOR i=1 TO m-1
  sum := sum + Ai
ENDFOR
```

Time Complexity: $\Theta(m)$

Komputer *shared memory*

```
Global_sum := 0
FOR all Pi WHERE 0 <= i <= N-1
  Local_sum := 0
  Calculate local sum of m/N numbers
  LOCK
  Global_sum := Global_sum + Local_sum
  UNLOCK
ENDFOR
```

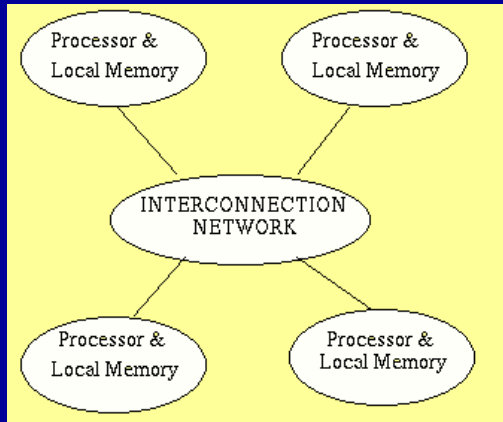
Note: **Global_sum** is a shared variable each processor must have mutually exclusive access to it

Time Complexity: $\Theta((m / N) + N) + S$

S: waktu utk sinkronisasi

INTERCONNECTION NETWORK and MESSAGE PASSING

N prosesor memiliki unit memori pribadi, tidak ada memori persekutuan (*shared memori*).



Komunikasi antar prosesor dengan *mengirim* dan *menerima* pesan (*message*)

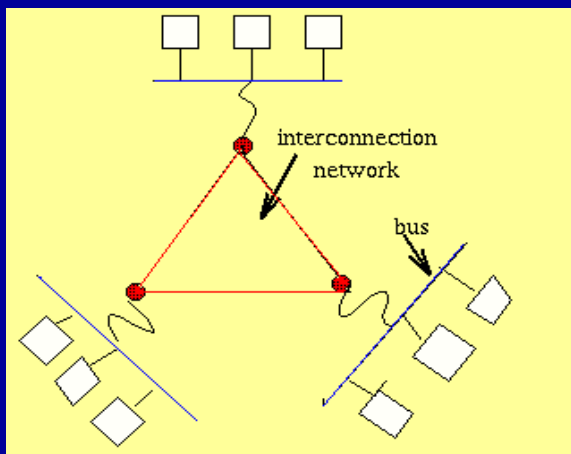


Prosesor P1
Menerima ($x, P2$)

Prosesor P2
Mengirim ($x, P1$)

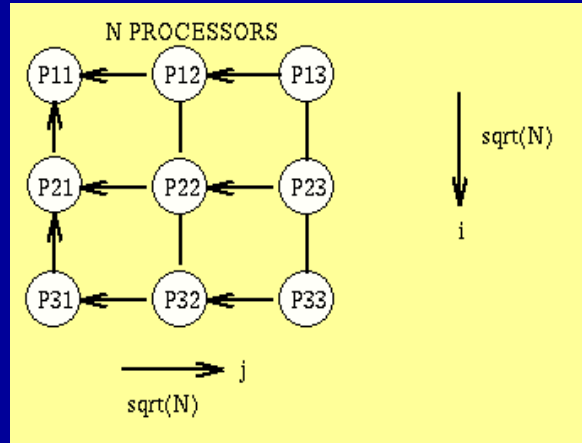
Disain Hibrid (Kombinasi)

Komunikasi *Intracluster* dan *Intercluster*



The processors operate **synchronously** and a **global clock** is used to ensure **lockstep** operation.

Penjumlahan m bilangan dengan N prosesor dan memori tersebar



Setiap prosesor menghitung **sum** lokal dari sejumlah m/N bilangan. Setiap prosesor mengirimkan **sum** lokal ke prosesor lain (pd waktu yang tepat) sampai **sum** global sudah berada pada prosesor **P11**

Penjumlahan m bilangan dengan N prosesor dan memori tersebar

```

FORALL Pij                                     * Semua prosesor aktif
  local_sum of m / N numbers
ENDFOR

FORALL Pij j = sqrt(N) downto 2               * Semua prosesor kol j aktif
  Pij passes local sum to Pij-1
  Pij-1 calculates local sum = new_local_sum + old_local_sum
ENDFOR

FORALL Pi1 i = sqrt(N) downto 2               * 1 prosesor di kol i aktif
  Pi1 passes local sum to Pi-1,1
  local sum = new_local_sum + old_local_sum
ENDFOR
  
```

Time Complexity: $\Theta(m/N + 2(\text{sqrt}(N)) - 2 + C)$

dimana C adalah waktu untuk komunikasi

Ilustrasi penjumlahan

tahap ke-

1

10	10	10
10	10	10
10	10	10

Initially

2

10	20	-
10	20	-
10	20	-

$j = 3$

3

30	-	-
30	-	-
30	-	-

$j = 2$

4

30	-	-
60	-	-
-	-	-

$i = 3$

5

90	-	-
-	-	-
-	-	-

$i = 2$

i.e. 4 additions and
4 communications

$$(2\sqrt{N} - 2)$$