



Institut Informatika & Bisnis
DARMAJAYA
Yayasan Alfian Husin



**Kampus
Merdeka**
INDONESIA JAYA

**MERDEKA
BELAJAR**

DATA SCIENCE DARMAJAYA
“YOUR BEST FUTURE IN DATA”

PERTEMUAN KE: 14

Paralel Computing

KULIAH

OLEH: Nurjoko



Capaian Pembelajaran

1. Memahami dasar-dasar komputasi paralel, termasuk konsep pemrosesan bersamaan, distribusi tugas, dan sinkronisasi.
2. Mengetahui berbagai arsitektur komputasi paralel, seperti sistem terdistribusi, komputer kluster, dan superkomputer
3. Memahami model pemrograman paralel seperti Pthreads, OpenMP, dan MPI, serta kemampuan mereka untuk mengkoordinasikan dan menyinkronkan tugas.
4. Mengetahui perbedaan antara komputasi SIMD, MIMD, dan SPMD, dan dapat mengidentifikasi situasi di mana setiap jenis dapat diterapkan
5. Mengenali peran komputasi paralel dalam pengolahan dan analisis data besar, dan kemampuannya untuk memberikan solusi yang efisien

Pengenalan *Parallel Computing*

Terminologi

□ **Pengolahan Paralel :**

Pengolahan informasi yang ditekankan pada manipulasi elemen data yang dimiliki oleh satu atau lebih dari satu proses secara bersamaan dalam rangka menyelesaikan sebuah problem.

□ **Komputer Paralel :**

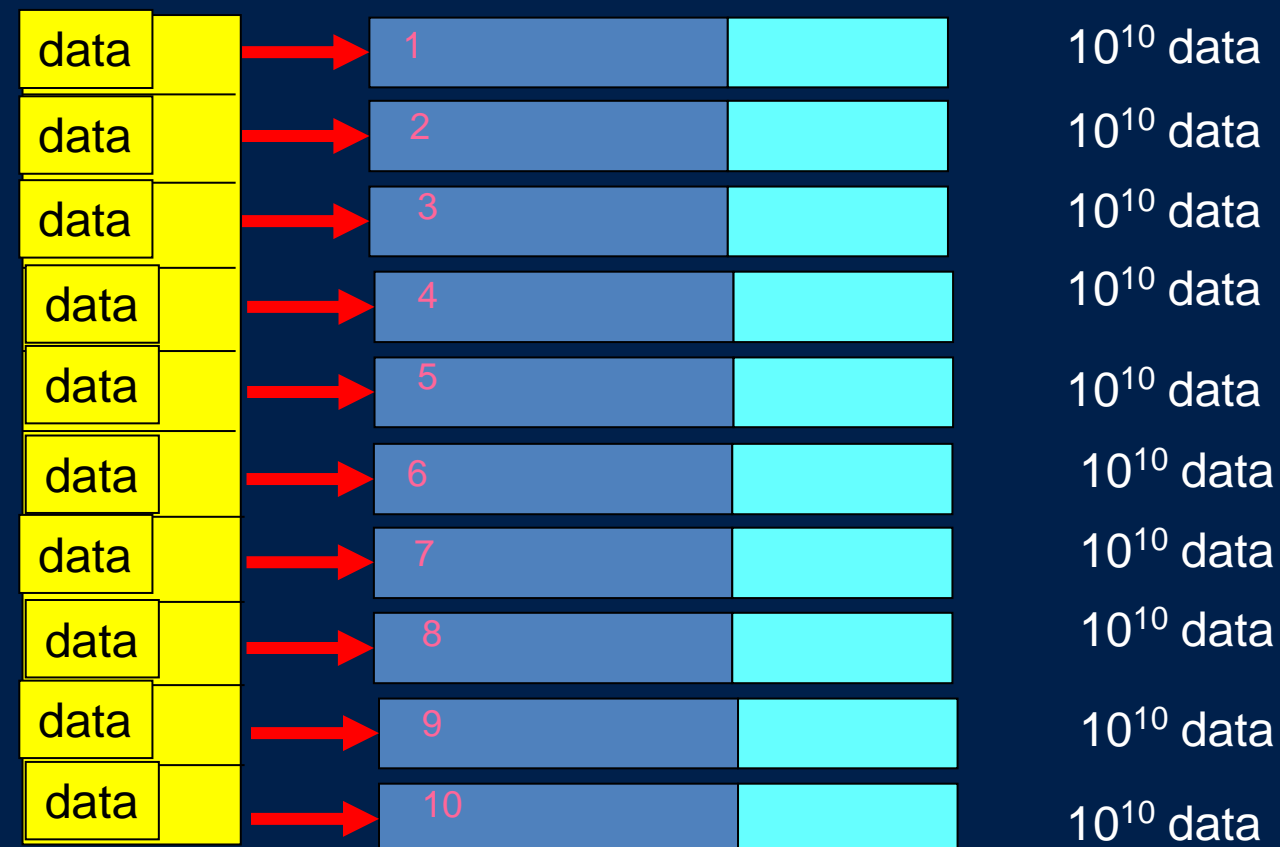
Komputer multi-prosesor dengan kemampuan melakukan pengolahan paralel.

□ **Supercomputer :**

sebuah general-purpose computer yang mampu menyelesaikan problem dengan kecepatan komputasi sangat tinggi. Semua superkomputer kontemporer adalah komputer paralel. Beberapa di antaranya memiliki prosesor yang sangat kuat dalam jumlah yang relatif sedikit, sementara yang lainnya dibangun oleh mikroprosesor dalam jumlah yang cukup besar.

Mengapa menggunakan pemrosesan paralel?

Kecepatan proses tiap prosesor:
 10^{11} data/s

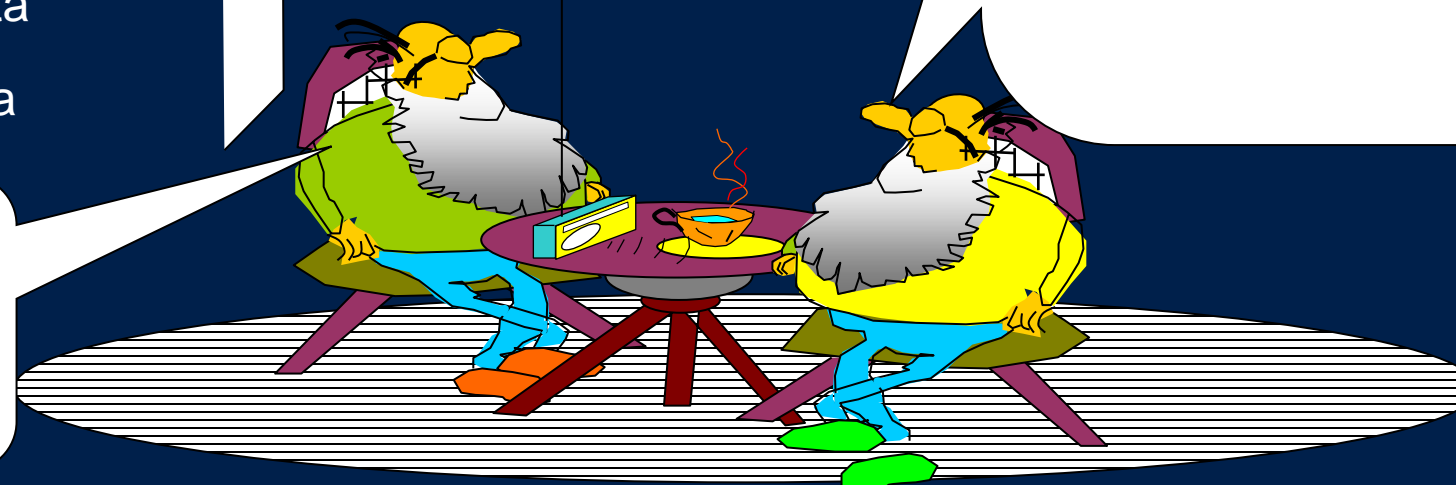


total data dalam 1 detik:
 $10^{10} \times 10 = 10^{11}$ data

Kita gunakan pemrosesan paralel.

Lebih dari satu prosesor yang bekerja, sehingga batasan transmisi data tersebut bisa diatasi.

Datanya banyak sekali, dan kita perlu kecepatan tinggi. Bagaimana ini?

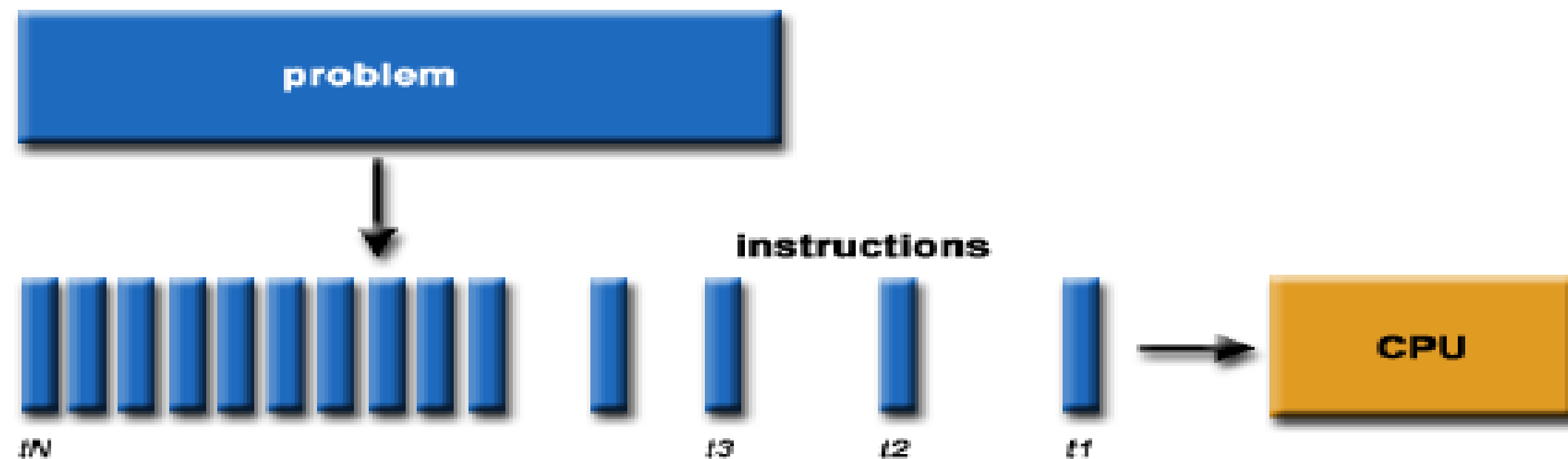




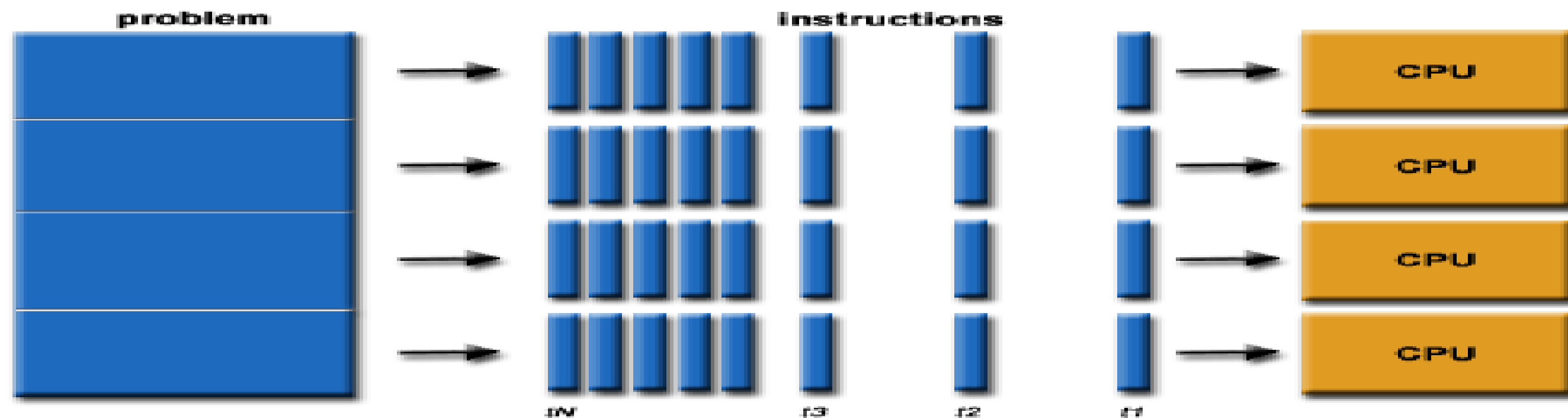
ALASAN UTAMA DALAM PENGGUNAAN KOMPUTASI PARALEL, YAITU :

1. Keinginan untuk menyelesaikan masalah dalam waktu yang lebih singkat dan keinginan untuk menyelesaikan masalah yang lebih besar.
2. Masalah efisiensi biaya, penggunaan sumberdaya non-lokal yang tersedia maupun keterbatasan fisik dari sumberdaya komputasi itu sendiri.

Software tradisional umumnya dibuat untuk komputasi serial (Gambar 1) yang dijalankan oleh prosesor tunggal dimana sebuah persoalan dipecah ke dalam instruksi yang dieksekusi secara berurutan dan hanya satu instruksi yang boleh dieksekusi pada saat yang sama



Pada komputasi paralel (Gambar 2) sebuah persoalan dipecah menjadi beberapa bagian yang dapat diselesaikan pada saat yang bersamaan. Setiap bagian selanjutnya dipecah menjadi instruksi yang berurutan dan masing-masing bagian dikerjakan oleh prosesor yang berbeda. Sumber daya komputasi yang dimaksud dapat berupa : sebuah komputer tunggal dengan beberapa prosesor; beberapa komputer yang terhubung dalam jaringan; maupun kombinasi dari keduanya.

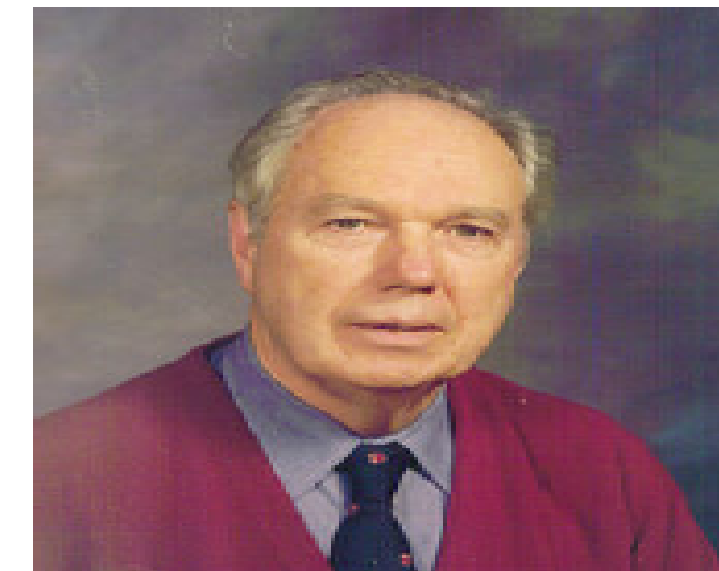


- Arsitektur *Parallel Computing*

Taksonomi Flynn

- ▶ Klasifikasi arsitektur komputer (1966)
- ▶ Membedakan arsitektur komputer multiprosesor berdasarkan dimensi instruksi dan data. Kemudian tiap dimensi dibagi menjadi single atau multiple

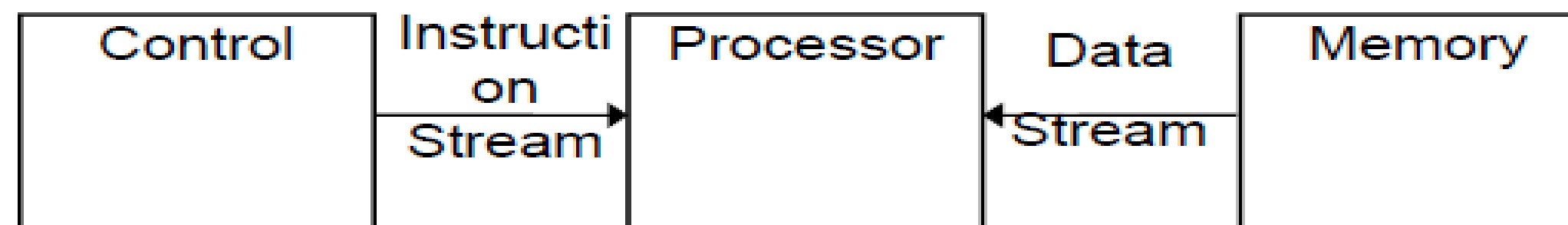
	Single Instruction	Multiple Instruction
Single Data	SISD	MISD
Multiple Data	SIMD	MIMD



Michael J. Flynn
(1903-1957)

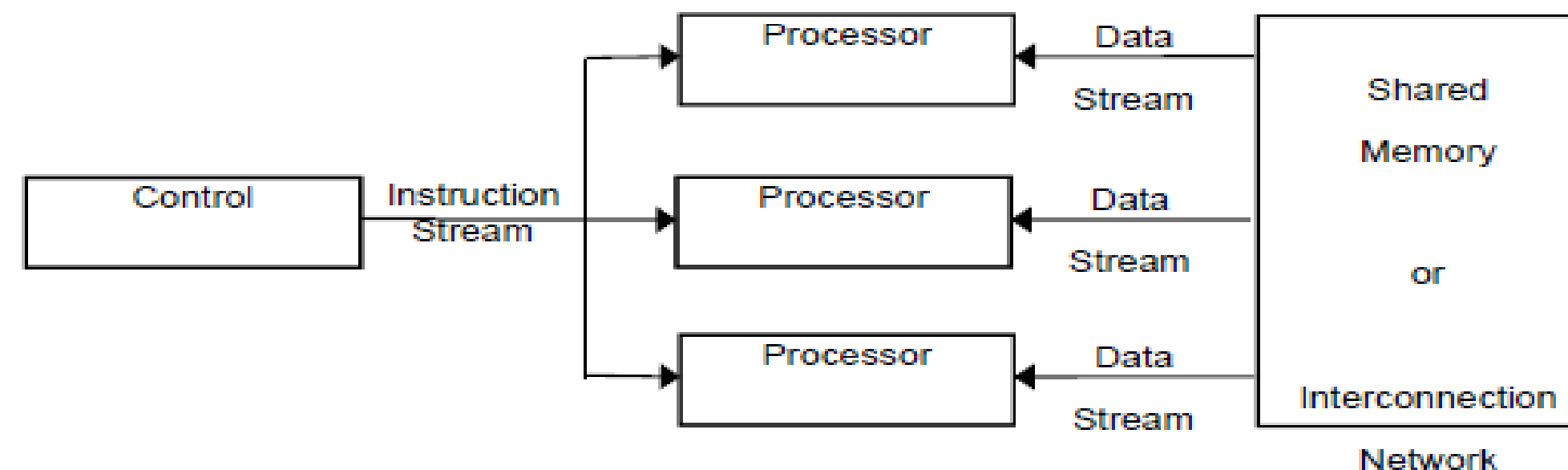
1. SISD (Single Instruction stream, Single Data stream)

- ❑ Komputer tunggal yang mempunyai satu unit kontrol, satu unit prosesor dan satu unit memori
- ❑ Instruksi dilaksanakan secara berurut tetapi boleh juga overlap dalam tahapan eksekusi (overlap)
- ❑ Satu alur instruksi didecode untuk alur data tunggal



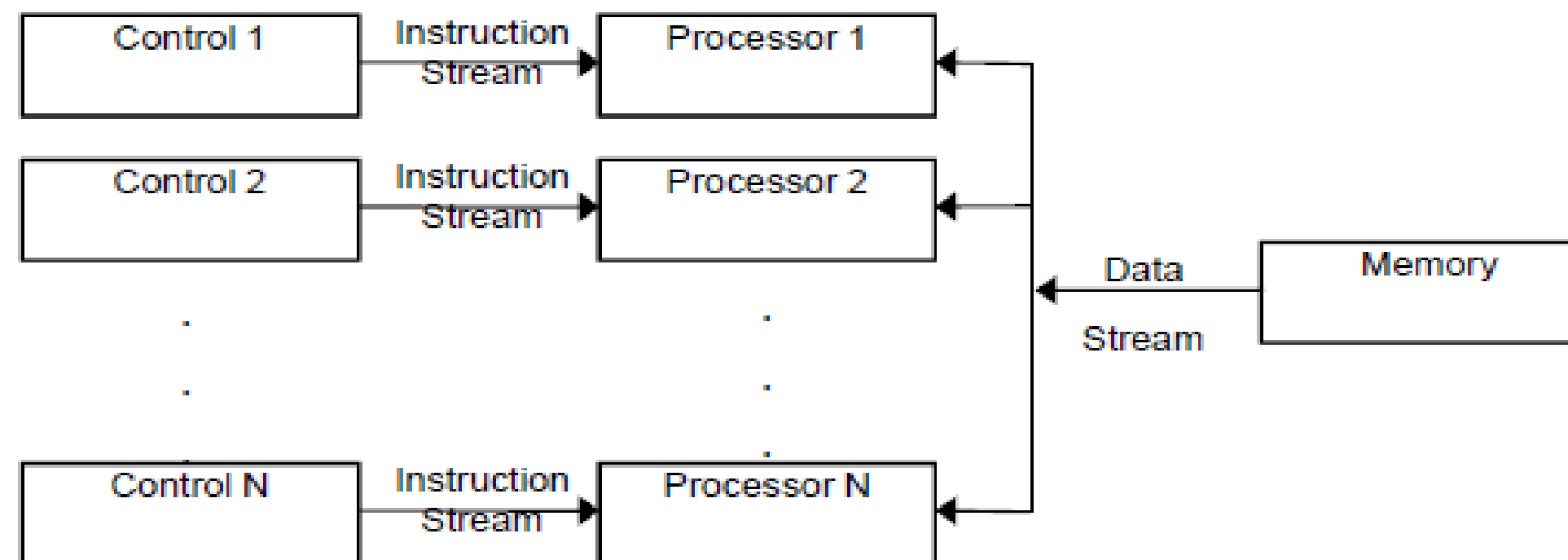
2. SIMD (Single Instruction stream, Multiple Data stream)

- Komputer yang mempunyai beberapa unit prosesor di bawah satu supervisi satu unit common control. Setiap prosesor menerima instruksi yang sama dari unit kontrol, tetapi beroperasi pada data yang berbeda.



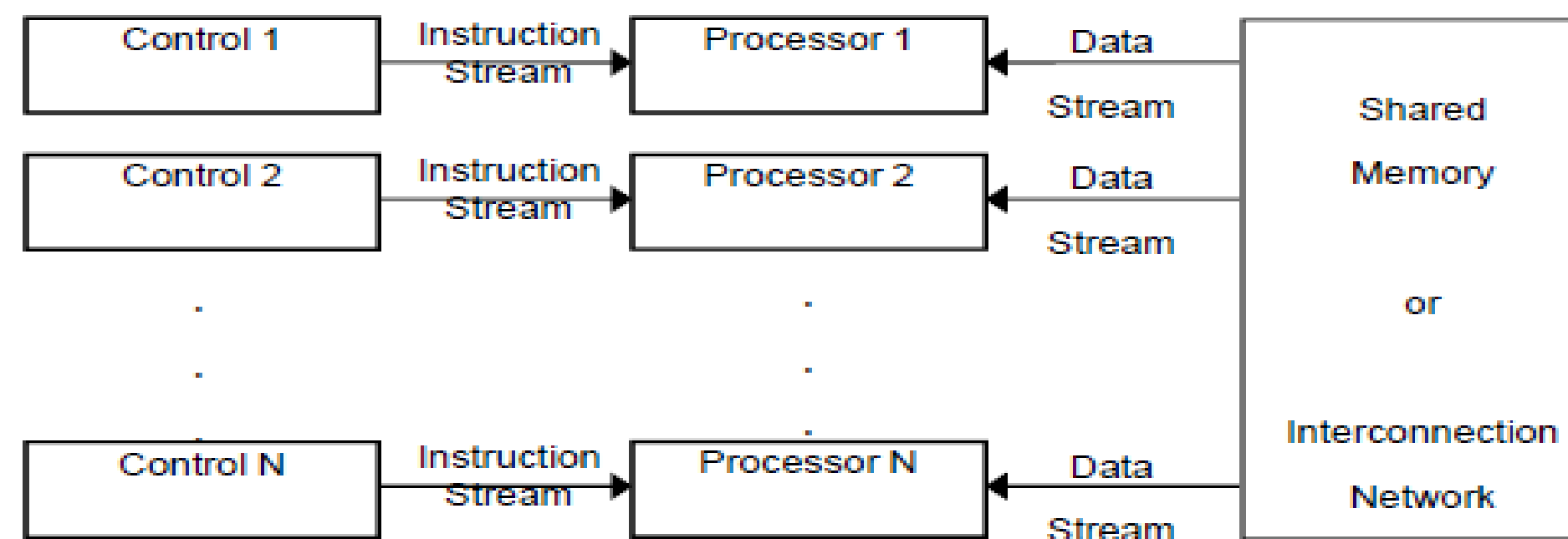
3. MISD (Multiple Instruction stream, Single Data stream)

- Sampai saat ini struktur ini masih merupakan struktur teoritis dan belum ada komputer dengan model ini.



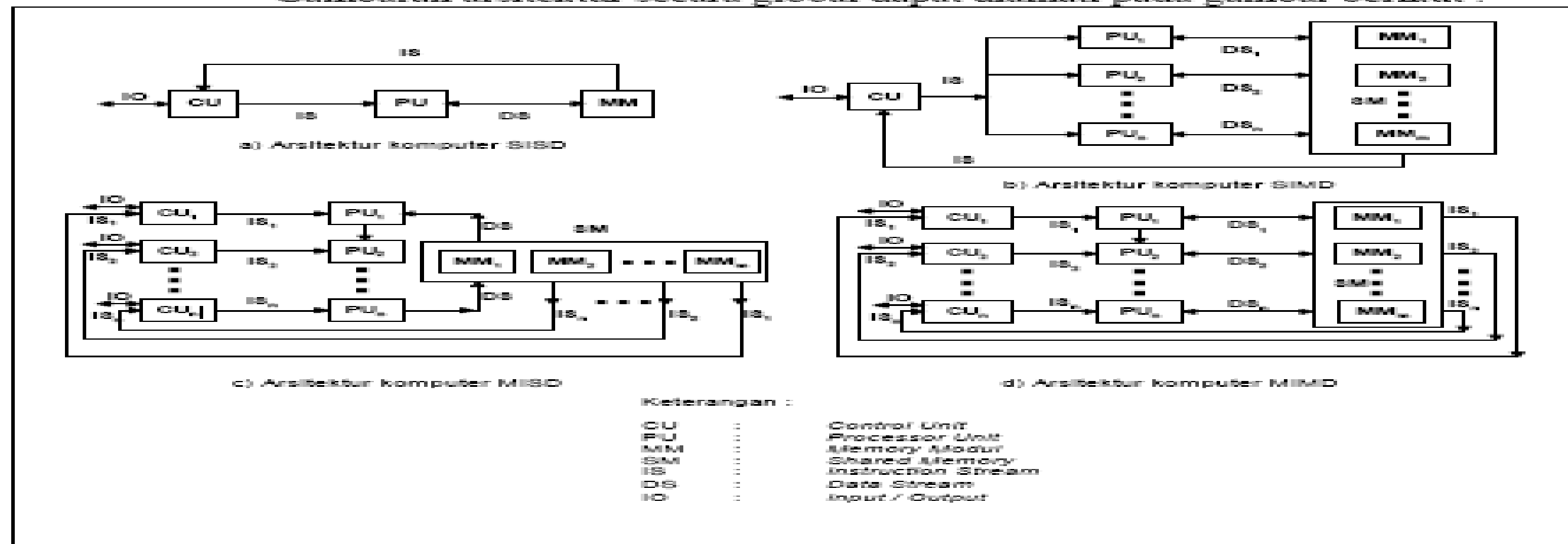
4. MIMD (Multiple Instruction stream, Multiple Data stream)

- Organisasi komputer yang memiliki kemampuan untuk memproses beberapa program dalam waktu yang sama. Pada umumnya multiprosesor dan multikomputer termasuk dalam kategori ini.



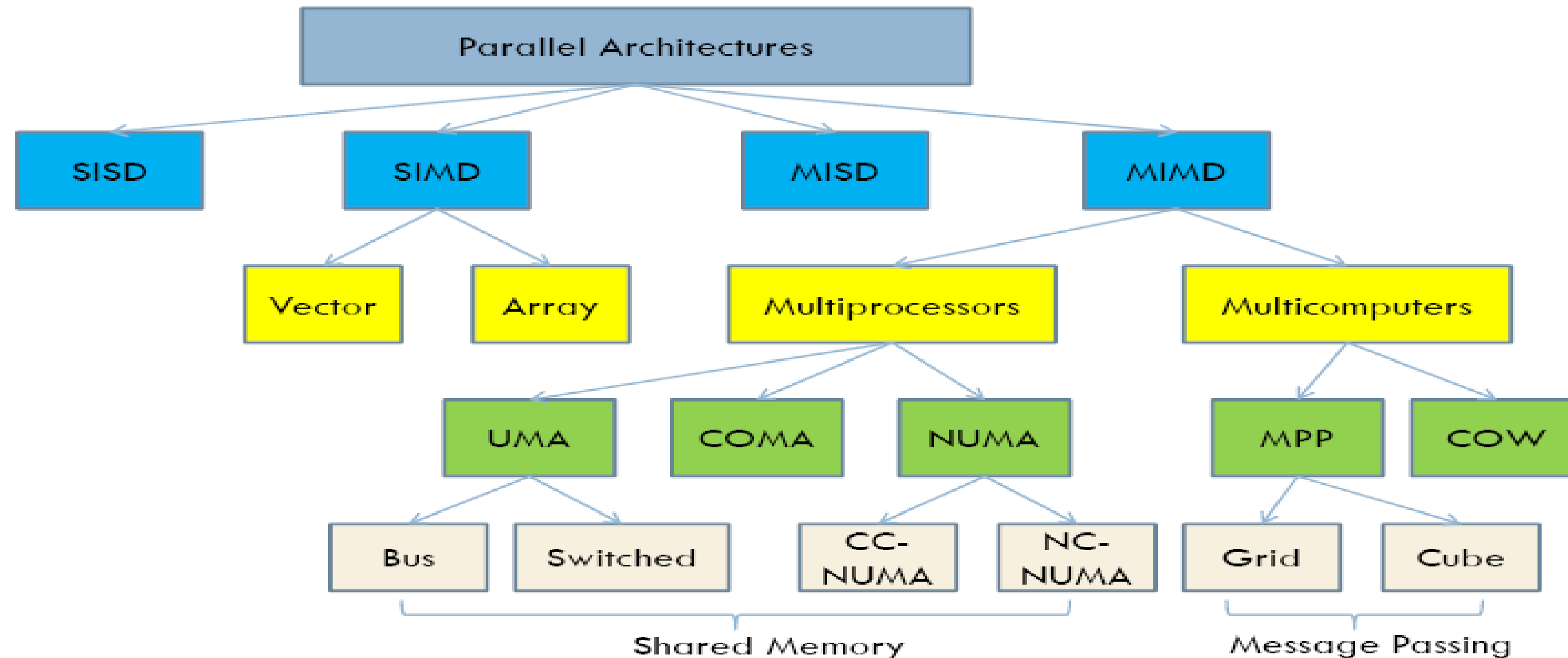
- MIMD dibagi menjadi 2 grup:
 - ▣ Multiprocessor yang menggunakan memory bersama.
 - ▣ Multicomputer.

Gambaran arsitektur secara global dapat diamati pada gambar berikut :



Gambar 1 Pengelompokan Arsitektur Komputer Menurut Flynn

A Taxonomy of Parallel Computers

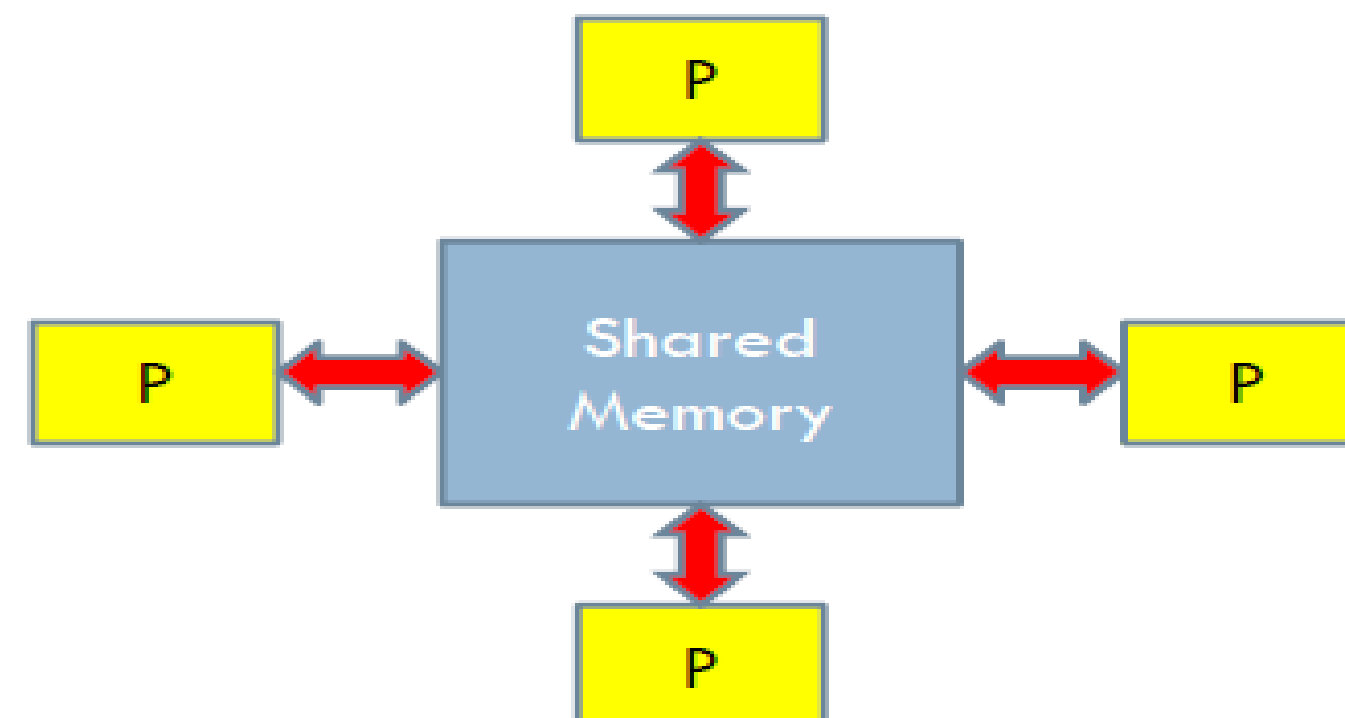


□ Keterangan:

- UMA : Uniform Memory Access
- NUMA : Non Uniform Memory Access
- COMA : Cache Only Memory Access
- MPP : Massively Parallel Processor
- COW : Cluster of Workstations
- CC-NUMA : Cache Coherent NUMA
- NC-NUMA : Non Cache NUMA

Multiprocessor

- Sistem multiprocessor adalah suatu komputer yang mempunyai lebih dari satu CPU pada motherboardnya. Jika sistem operasi dibangun untuk memanfaatkan kelebihan ini, maka SO tersebut dapat menjalankan proses-proses berbeda (atau thread-thread berbeda yang dimiliki oleh proses yang sama) pada CPU-CPU berbeda.



Processors are connected with memory via the memory bus (which is fast), or switches

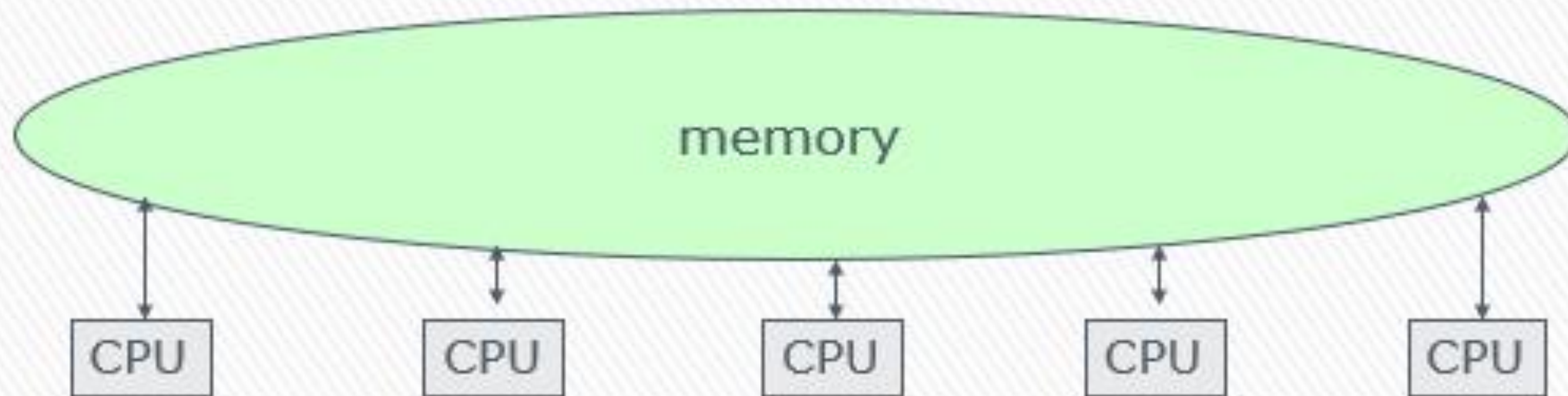
Arsitektur Memori Komputer Paralel

Arsitektur memori paralel:

1. **Shared memory**: prosesor dapat mengakses semua memori sebagai space alamat global
 - 1.1 Uniform Memory Access (UMA)
 - 1.2 Non-Uniform Memory Access (NUMA)
2. **Distributed memory**: prosesor mempunyai memori lokal sendiri. Inter-prosesor memori perlu networking
3. **Hybrid distributed-shared memory**: menggunakan shared dan distributed

Shared Memory Multiprocessors

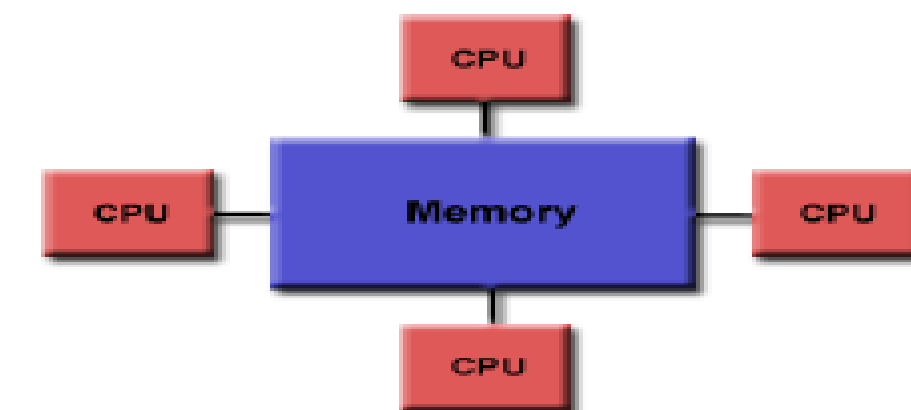
- 3 Model Shared Memory Multiprocessors:
 - ▣ Uniform Memory Access (UMA)
 - ▣ Nonuniform Memory Access (NUMA)
 - ▣ Cache Only Memory Architecture (COMA)
- Yang membedakan dari ketiga model diatas adalah bagaimana memori dan pheripheral device di shared atau didistribusikan



Shared Memory

Shared Memory (UMA)

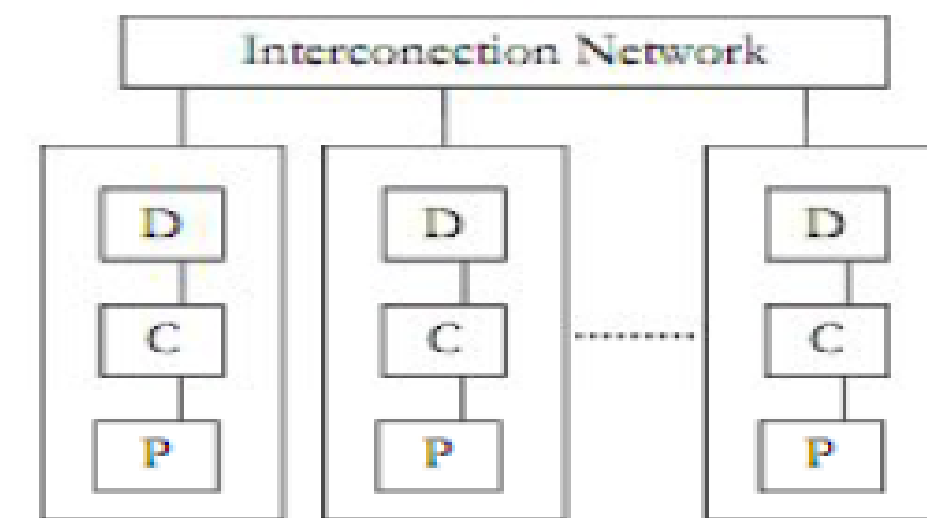
- ▶ Karakteristik: semua prosesor dapat mengakses semua memori sebagai ruang alamat global
 - ▶ Multi prosesor dapat beroperasi secara independen namun sharing memori
 - ▶ Perubahan di lokasi memori oleh satu prosesor dapat dilihat oleh prosesor lain
-
- ▶ Prosesor identik, misalnya mesin SMP
 - ▶ Upto 8 prosesor
 - ▶ Mempunyai akses dan waktu akses yang sama ke memori
 - ▶ Kadang disebut CC-UMA (cache coherent)
 - ▶ cache coherent: jika satu prosesor update suatu lokasi di memori, semua prosesor mengetahui update tsb
 - ▶ Koherensi dilakukan di level hardware



Shared Memory (UMA)

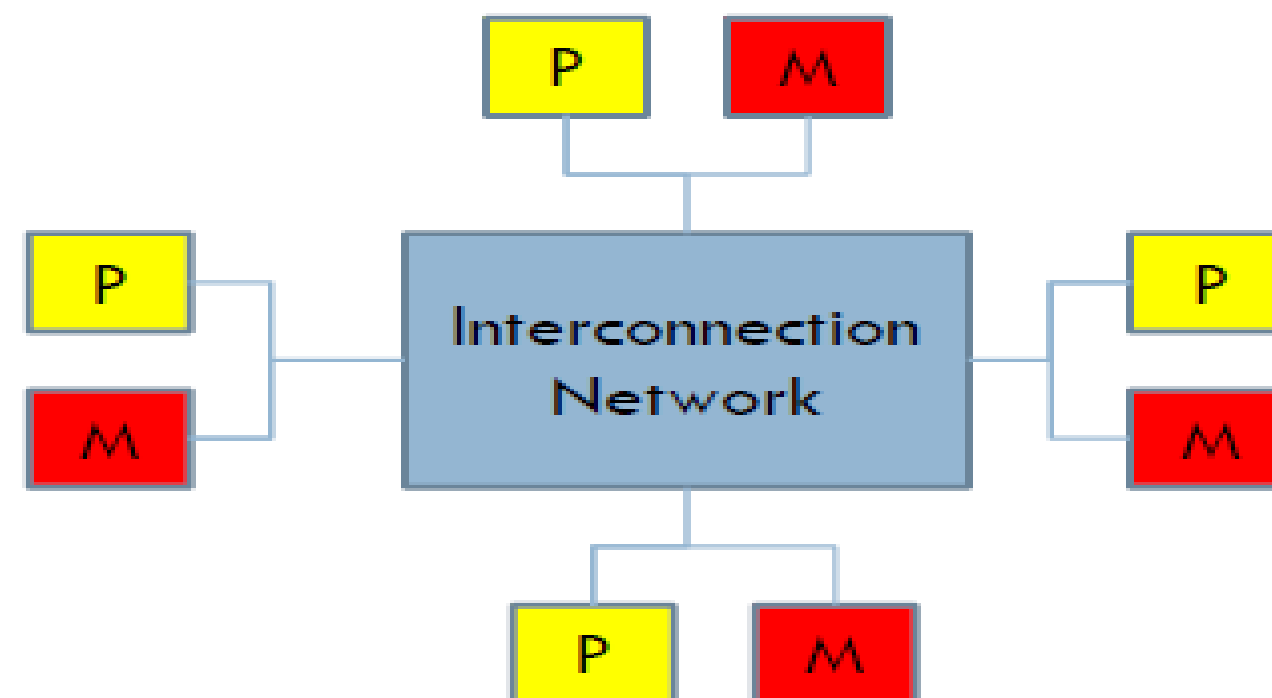
Cache Only Memory Access (COMA)

- ❑ COMA adalah multiprocessor yang hanya menggunakan cache memory
- ❑ COMA dapat ditemukan pada NUMA machines, dimana pendistribusioan main memory dirubah ke cache. Disini tidak ada hirarki memori pada setiap node computer
- ❑ Semua cache berasal dari ruang alamat global. Akses ke cache jauh dibantu oleh direktori cache yang didistribusikan. Tergantung kepada interkoneksi jaringan yang digunakan, terkadang direktori digunakan untuk membantu penempatan copian dari blok-blok cache
- ❑ Penempatan awak data tidak penting karena data akan menempati tempat dimana data tersebut akan digunakan



Multicomputer

- ❑ **Sistem Multicomputer**
Multicomputer dapat dianggap berupa suatu komputer NUMA loosely atau cluster yang tightly coupled.
- ❑ Multicomputer biasanya digunakan ketika diperlukan power komputasi tinggi tetapi lingkungan mempunyai ruang fisik atau tenaga listrik terbatas.



Processors have private address space.
Interprocess communication via message passing only
In clusters, computers are connected via LAN

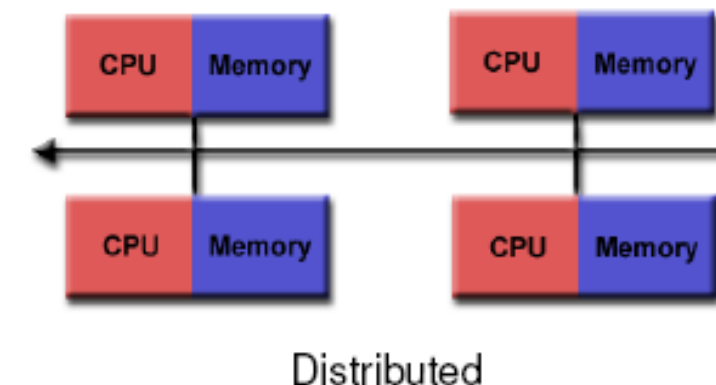
Bookmarks

Options

- Pengolahan Paralel
- Konsep Pengolahan Paralel
- Umpan Balik
- Lisensi

Distributed Memory

- ▶ Tiap prosesor mempunyai memori lokal sendiri
 - ▶ Sehingga prosesor dapat beroperasi secara independen
 - ▶ Perubahan ke lokal memori tidak membawa efek ke memori lain
 - ▶ Konsep cache coherence tidak berlaku
 - ▶ Jika memerlukan interprosesor, tugas programmer secara eksplisit mendefinisikan bagaimana dan kapan data akan dikomunikasikan



Pengolahan Paralel

@2012,Eko Didik Widianto

Pengolahan Paralel

Konsep Pengolahan Paralel

Arsitektur Komputer

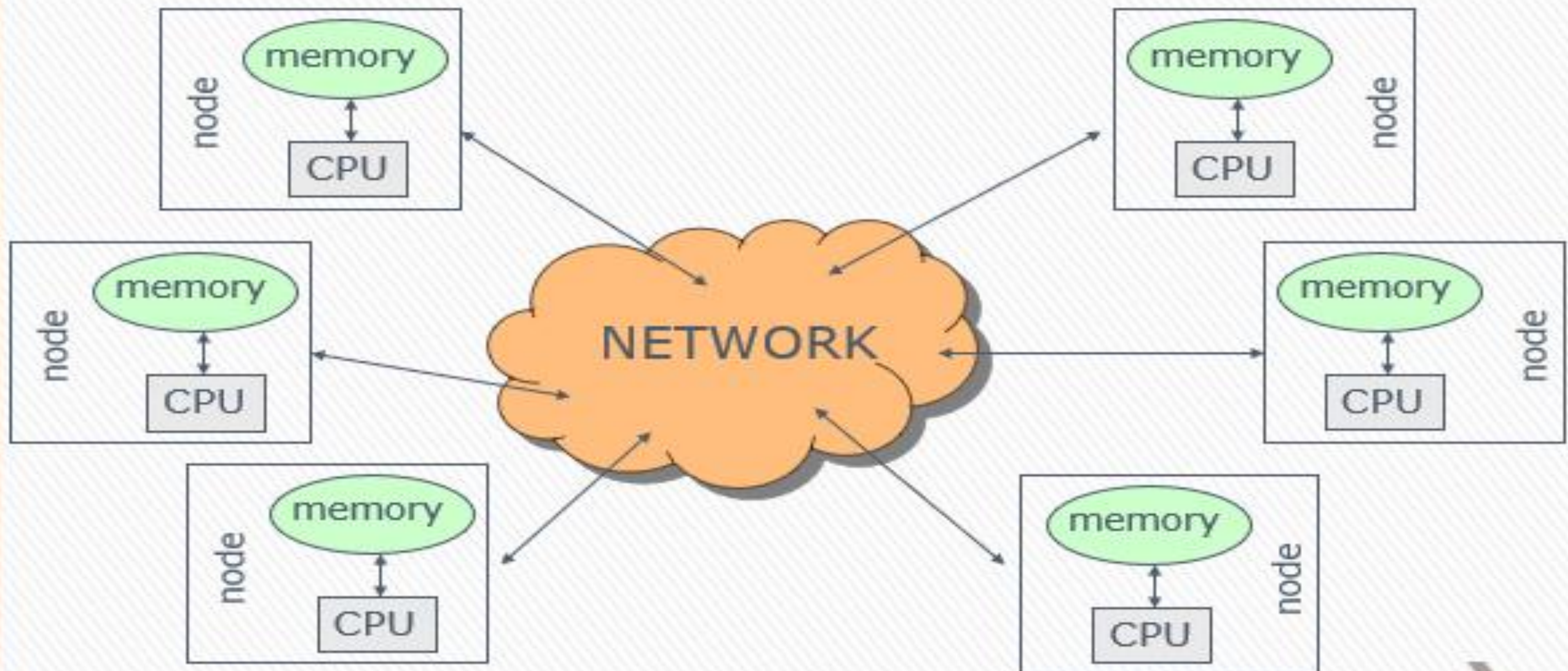
Taksonomi Flynn

Arsitektur Memori Paralel

Umpan Balik

Lisensi

Memory



Hybrid Memori

- ▶ Terdiri dari arsitektur memori shared dan distributed

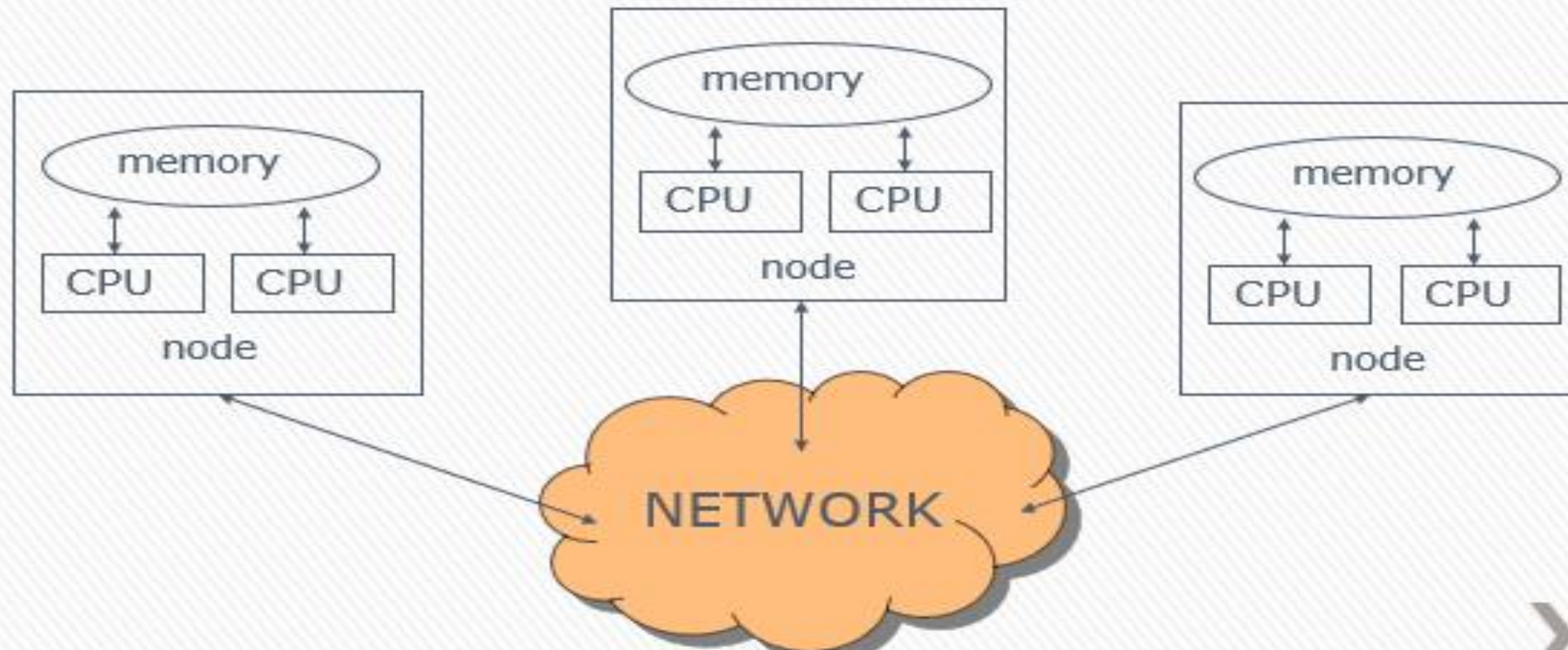
- ▶ Komponen memori shared biasanya mesin SMP koheren
 - ▶ Prosesor di mesin SMP mempunyai akses global ke memori mesin tersebut
- ▶ Komponen distributed adalah jaringan SMP multiple
 - ▶ SMP hanya tahu|memorinya saja
 - ▶ Komunikasi jaringan diperlukan untuk memindahkan data dari satu SMP ke lainnya



Hybrid

- ▶ Trend ke depan menggunakan tipe arsitektur memori ini

Mixed Architectures





- Ide awal komputasi grid dimulai dengan adanya distributed computing, yaitu mempelajari penggunaan komputer terkoordinasi yang secara fisik terpisah atau terdistribusi.
- Distributed computing. Seperti yang tercermin dari namanya, distributed computing berarti komputasi yang terdistribusi.

- Proses komputasi tidak terjadi dalam satu komputer saja akan tetapi didistribusikan ke beberapa komputer.
- Analoginya seperti kerja kelompok membuat kliping, semua anggota kelompok mencari bahan-bahan berdasarkan pembagian tugas kemudian bahan tersebut akhirnya dikumpulkan menjadi satu berbentuk kliping sebagai bentuk karya kelompok.
- Analogi lain bisa berupa kelompok-kelompok kerja lain semisal kantor, pabrik, dll. Intinya proses tersebar dalam kelompok, namun menghasilkan satu output. Distributed computing adalah salah satu contoh parallel processing (pemrosesan paralel).



- Sistem terdistribusi membutuhkan aplikasi yang berbeda dengan sistem terpusat.
- Kemudian berkembang lagi menjadi parallel computing yang merupakan teknik komputasi secara bersamaan dengan memanfaatkan beberapa komputer secara bersamaan.

Grid Computing vs Komputasi Paralel

- Komputasi paralel bersifat homogen (seluruh resource yang digunakan Serupa) sedangkan komputasi grid bisa bersifat heterogen (berasal dari Bermacam macam resource yang berbeda) ataupun homogen.
- Komputasi paralel hanya terdiri dari satu cluster sedangkan komputasi grid bisa terdiri dari beberapa cluster
- Pembagian job execution dalam komputasi paralel harus sama besar antar tiap resource, sedangkan pada komputasi grid tidak harus sama

Pemrograman Paralel

- Teknik pemrograman komputer yang memungkinkan eksekusi perintah/operasi secara bersamaan baik dalam komputer dengan satu (prosesor tunggal) ataupun banyak (prosesor ganda dengan mesin paralel) CPU.
- Tujuan utama dari pemrograman paralel adalah untuk meningkatkan performa komputasi. Semakin banyak hal yang bisa dilakukan secara bersamaan (dalam waktu yang sama), semakin banyak pekerjaan yang bisa diselesaikan.
- Komputasi paralel membutuhkan:
 - algoritma
 - bahasa pemrograman
 - compiler

Pemrograman Paralel

Ada 2 tehnik pemrograman pararel :

1. Message Passing Interface (MPI).

- MPI adalah sebuah standard pemrograman yang memungkinkan pemrogram (programmer) untuk membuat sebuah aplikasi yang dapat dijalankan secara paralel.
- Proses yang dijalankan oleh sebuah aplikasi dapat dibagi untuk dikirimkan ke masing - masing compute node yang kemudian masing - masing compute node tersebut mengolah dan mengembalikan hasilnya ke komputer head node. Untuk merancang aplikasi paralel tentu membutuhkan banyak pertimbangan - pertimbangan diantaranya adalah latensi dari jaringan dan lama sebuah tugas dieksekusi oleh prosesor.



Pemrograman Paralel

Kegunaan MPI yang lain adalah :

1. Menulis kode paralel secara portable,
2. Mendapatkan performa yang tinggi dalam pemrograman paralel,
3. Menghadapi permasalahan yang melibatkan hubungan data irregular atau dinamis yang tidak begitu cocok dengan model data paralel.



Pemrograman Paralel

2. PVM (Parallel Virtual Machine)

- a. Paket software yang mendukung pengiriman pesan untuk komputasi parallel antara komputer.
- b. PVM dapat berjalan diberbagai macam variasi UNIX atau pun windows dan telah portable untuk banyak arsitektur seperti PC, workstation, multiprocessor dan superkomputer.



Learning Objective n

Fill in



CONCLUSION

Fill in



REFERENCES

Fill in IEEE Style

[Introduction to Parallel Computing Tutorial | HPC @ LLNL](#)

<https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial##Whatis>



Institut Informatika & Bisnis
DARMAJAYA
Yayasan Alfian Husin



**Kampus
Merdeka**
INDONESIA JAYA

**MERDEKA
BELAJAR**

THANK YOU!!

DATA SCIENCE DARMAJAYA “YOUR BEST FUTURE IN DATA”