

PENDAHULUAN

1. Kebutuhan akan Pengolahan Paralel

Motivasi :

- . Pengolahan data numerik dalam jumlah yang sangat besar.
- . Kebutuhan akan ketersediaan data yang senantiasa up to date.

Contoh 1.1. :

Simulasi sirkulasi global laut di Oregon State University.

Lautan dibagi ke dalam 4096 daerah membentang dari timur ke barat, 1024 daerah membentang dari utara ke selatan dan 12 lapisan. Berarti terdapat sekitar 50 juta daerah berdimensi tiga.

Satu iterasi mampu mensimulasikan sirkulasi lautan untuk jangka waktu 10 menit dan membutuhkan sekitar 30 milyar kalkulasi floating point. Para ahli kelautan ingin menggunakan model tersebut untuk mensimulasikan sirkulasi lautan untuk periode 1 tahun.

Pengolahan Paralel :

- . pengolahan informasi yang menekankan pada manipulasi data-data elemen secara simultan.
- . dimaksudkan untuk mempercepat komputasi dari sistem komputer dan menambah jumlah keluaran yang dapat dihasilkan dalam jangka waktu tertentu.

Komputer Paralel :

- . Komputer yang memiliki kemampuan untuk melakukan pengolahan paralel.

Throughput :

- . Banyaknya keluaran yang dihasilkan per unit waktu.

Peningkatan throughput dapat dilakukan dengan :

- . Meningkatkan kecepatan operasi

- Meningkatkan jumlah operasi yang dapat dilakukan dalam satu waktu tertentu (concurrency).

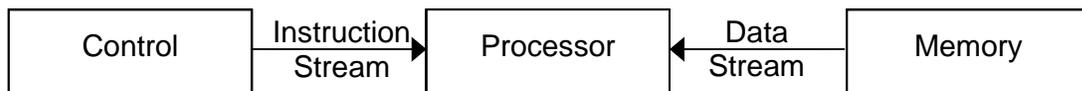
2. Paradigma Pengolahan Paralel

2.1. M. J. FLYNN

Pengklasifikasian oleh Flynn, dikenal sebagai Taksonomi Flynn, membedakan komputer paralel ke dalam empat kelas berdasarkan konsep aliran data (data stream) dan aliran instruksi (instruction stream), sebagai : SISD, SIMD, MISD, MIMD.

SISD (Single Instruction stream, Single Data stream)

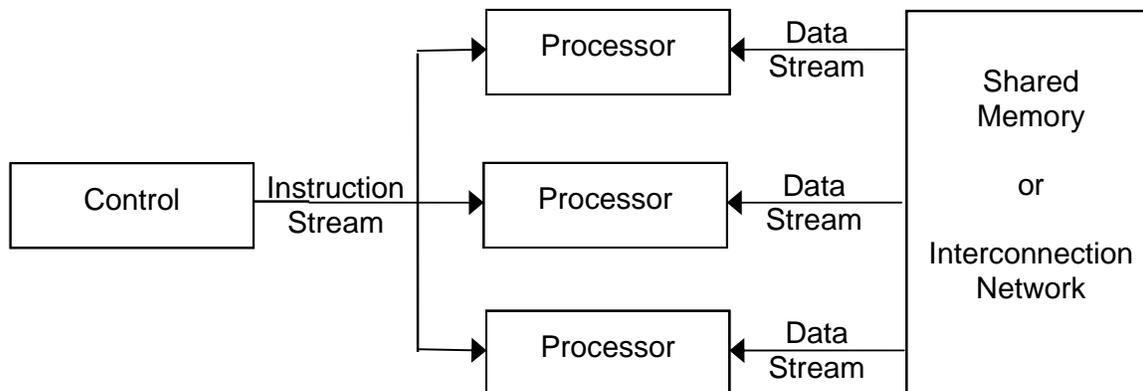
Komputer tunggal yang mempunyai satu unit kontrol, satu unit prosesor dan satu unit memori.



[Akl 1989]

SIMD (Single Instruction stream, Multiple Data stream)

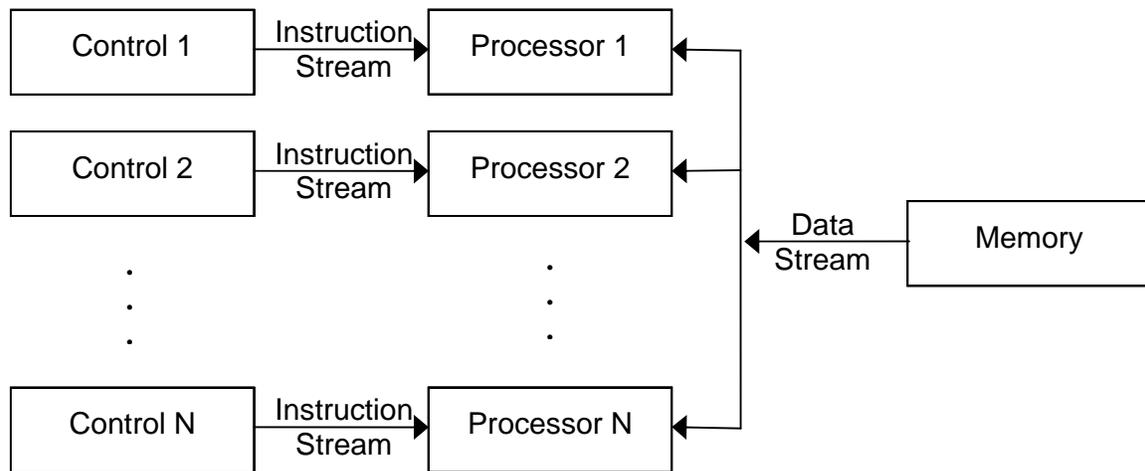
Komputer yang mempunyai beberapa unit prosesor di bawah satu supervisi satu unit common control. Setiap prosesor menerima instruksi yang sama dari unit kontrol, tetapi beroperasi pada data yang berbeda.



[Akl 1989]

MISD (Multiple Instruction stream, Single Data stream)

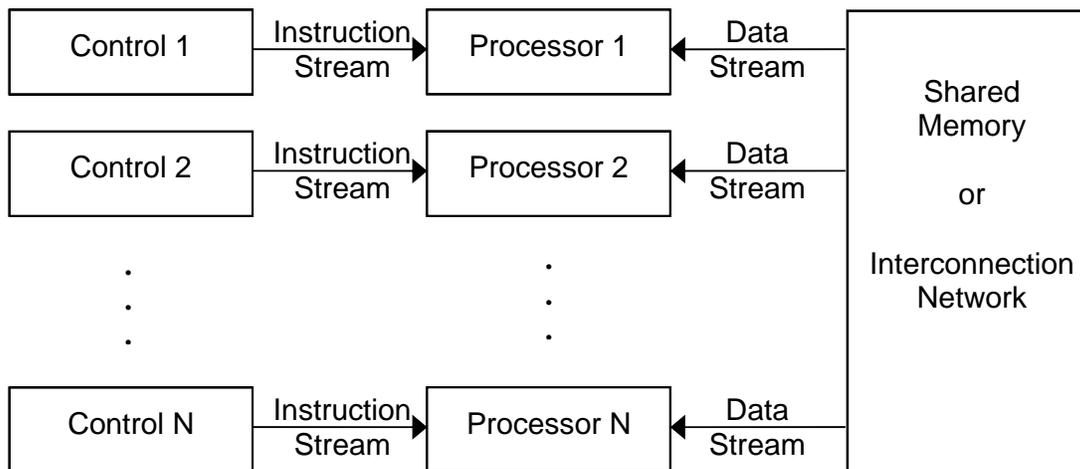
Sampai saat ini struktur ini masih merupakan struktur teoritis dan belum ada komputer dengan model ini.



[Akl 1989]

MIMD (Multiple Instruction stream, Multiple Data stream)

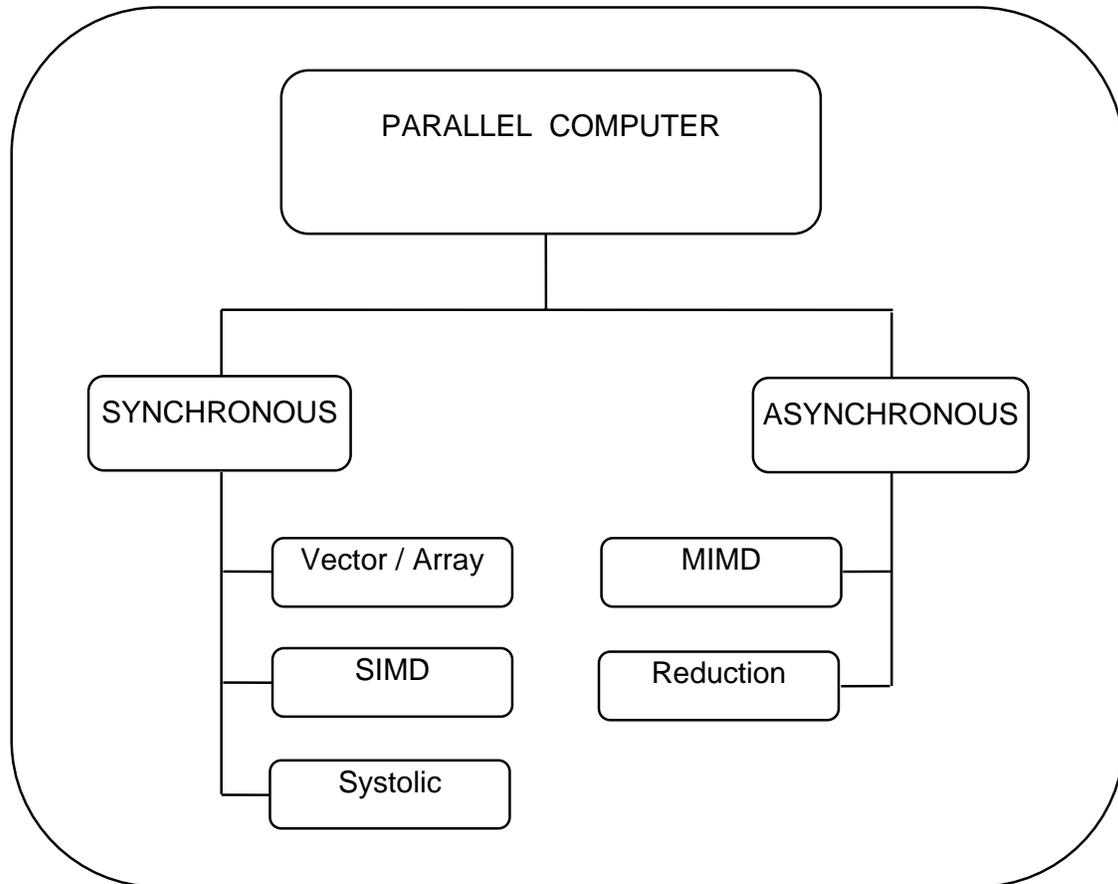
Organisasi komputer yang memiliki kemampuan untuk memproses beberapa program dalam waktu yang sama. Pada umumnya multiprosesor dan multikomputer termasuk dalam kategori ini.



[Akl 1989]

2.2. T.G. LEWIS

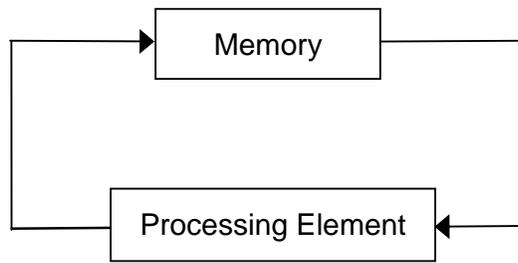
T.G. Lewis membedakan komputer paralel ke dalam dua kelas, berdasarkan ada atau tidak adanya common global clock, sebagai : synchronous dan asynchronous.



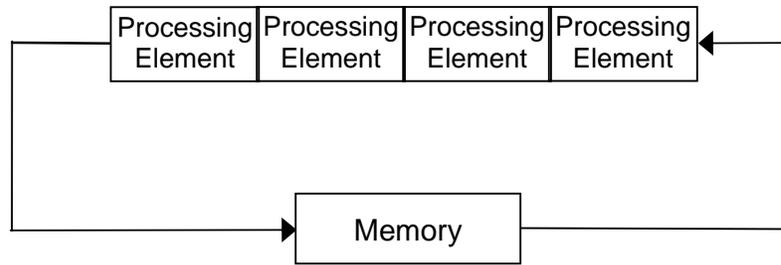
Gambar 1.
Taksonomi Parallel Computing oleh T.G. Lewis dkk. [Lewis,1992]

Synchronous :

- Pada komputer paralel yang termasuk dalam kategori ini terdapat koordinasi yang mengatur beberapa operasi untuk dapat berjalan bersamaan sedemikian hingga tidak ada ketergantungan antar operasi.
- Parallelism yang termasuk dalam kategori ini adalah vector/array parallelism, SIMD dan systolic parallelism.
- Systolic parallel computer adalah multiprocessor dimana data didistribusikan dan dipompa dari memory ke suatu array prosesor sebelum kembali ke memory.



Gambar 2
Model Komputasi Tradisional (SISD) [Quinn 1994]



Gambar 3
Model Komputasi Systolic Array [Quinn 1994]

Asynchronous :

- Pada komputer paralel yang termasuk dalam kategori asynchronous, masing-masing prosesor dapat diberi tugas atau menjalankan operasi berbeda dan masing-masing prosesor melaksanakan operasi tersebut secara sendiri-sendiri tanpa perlu koordinasi.
- Paradigma yang juga termasuk dalam kategori ini adalah MIMD dan reduksi.
- Paradigma reduksi adalah paradigma yang berpijak pada konsep graph reduksi. Program dengan model reduksi diekspresikan sebagai graph alur data. Komputasi berlangsung dengan cara mereduksi graph dan program berhenti jika graph akhirnya hanya mempunyai satu simpul.

2.3. MICHAEL J. QUINN

Quinn membedakan paralelisma ke dalam dua jenis : Data Parallelism dan Control Parallelism.

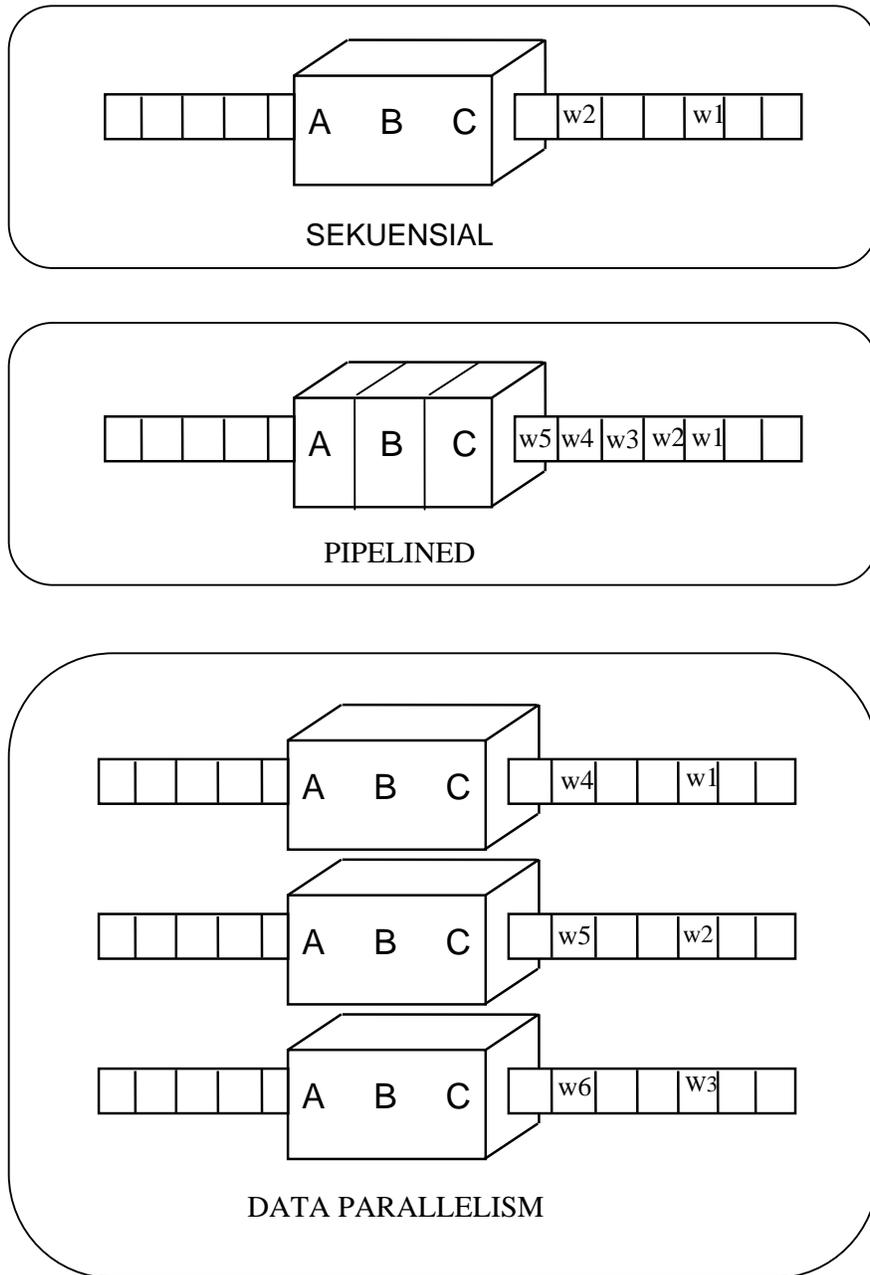
Data Parallelism :

penerapan operasi yang sama secara simultan terhadap elemen-elemen dari kumpulan data.

Control Parallelism :

penerapan operasi-operasi berbeda terhadap elemen-elemen data yang berbeda secara bersamaan. Pada control parallelism dapat terjadi aliran data antar proses-proses dan kemungkinan terjadi aliran data yang kompleks/rumit.

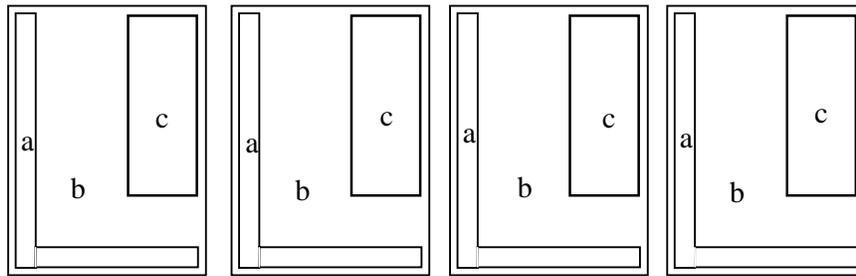
Pipeline merupakan satu kasus khusus dari control parallelism dimana aliran data membentuk jalur yang sederhana.



Gambar 4.
Ilustrasi perbandingan pipelining dengan data parallelism. [Quinn 1994]

Contoh :

Perhatikan ke-empat taman yang harus dirawat berikut ini :



Taman 1

Taman 2

Taman 3

Taman 4

a. Tanaman Pagar

b. Lapangan rumput

c. Kebun bunga

Nama Pekerjaan	Pekerja
1. Merapihkan tanaman pagar	Ali , Budi , Cipto, Dadang, Edi
2. Memangkas rumput	Frans, Gugun
3. Menanam bibit bunga	Heru, Indra, Joko
4. Menyiram taman	Ali

Pekerjaan 4 dapat dilakukan jika ketiga pekerjaan 1, 2 dan 3 telah selesai. Pekerjaan 1, 2 dan 3 dapat dilakukan secara bersamaan, sebagai contoh control parallelism. Masing-masing pekerjaan adalah contoh data parallelism. Sementara pekerjaan 4 dikerjakan pada sebuah taman, pekerjaan 1, 2 dan 3 dapat dikerjakan pada satu taman yang lain.

3. 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.

Throughput :

banyaknya keluaran yang dihasilkan per unit waktu

Pipeline :

Pada komputasi pipelined, komputasi dibagi ke dalam sejumlah langkah yang masing-masing disebut sebagai segmen, atau stage. Output dari sebuah segmen menjadi input segmen yang lain.

4. Analisa Algoritma Paralel

Pada saat sebuah algoritma digunakan untuk memecahkan sebuah problem, maka performance dari algoritma tersebut akan dinilai. Hal ini berlaku untuk algoritma sekuensial maupun algoritma paralel. Penampilan sebuah algoritma pengolahan paralel dapat dinilai dari beberapa kriteria, seperti running time dan banyaknya prosesor yang digunakan.

4.1. Running Time

Running time adalah waktu yang digunakan oleh sebuah algoritma untuk menyelesaikan masalah pada sebuah komputer paralel dihitung mulai dari saat algoritma mulai hingga saat algoritma berhenti. Jika prosesor-prosesornya tidak mulai dan selesai pada saat yang bersamaan, maka running time dihitung mulai saat komputasi pada prosesor pertama dimulai hingga pada saat komputasi pada prosesor terakhir selesai.

4.1.1. Counting Steps

Untuk menentukan running time, secara teoritis dilakukan analisa untuk menentukan waktu yang dibutuhkan sebuah algoritma dalam mencari solusi dari sebuah masalah. Hal ini dilakukan dengan cara menghitung banyaknya operasi dasar, atau step (langkah), yang dilakukan oleh algoritma untuk keadaan terburuknya (worst case).

Langkah-langkah yang diambil oleh sebuah algoritma dibedakan ke dalam dua jenis yaitu :

a. Computational step

Sebuah computational step adalah sebuah operasi aritmetika atau operasi logika yang dilakukan terhadap sebuah data dalam sebuah prosesor.

b. Routing step.

Pada routing step, sebuah data akan melakukan perjalanan dari satu prosesor ke prosesor lain melalui shared memory atau melalui jaringan komunikasi.

Contoh 4.1. :

Perhatikan sebuah file komputer dengan n entri berbeda.

Pada file tersebut akan diperiksa apakah x terdapat di dalamnya.

Dengan algoritma sekuensial, keadaan terburuknya (worst case) untuk menemukan x membutuhkan n langkah, dimana tiap langkah adalah membandingkan x dengan sebuah entri pada file. Keadaan terburuk terjadi jika x ternyata sama dengan entri terakhir pada file atau x tidak terdapat pada file tersebut.

Dengan EREW SM SIMD (Exclusive Read Exclusive Write Shared Memory SIMD) komputer dengan N prosesor, dimana $N \leq n$, pada worst casenya dibutuhkan $\log N + \frac{n}{N}$ langkah.

Misalkan P_1, P_2, \dots, P_N prosesor-prosesor pada EREW SM SIMD komputer tersebut.

Proses pencarian entri yang sama dengan x adalah :

-. Broadcasting, x dikomunikasikan pada semua prosesor dengan cara

1. P_1 membaca x dan mengkomunikasikan dengan P_2 .
2. P_1 dan P_2 secara simultan mengkomunikasikan x dengan P_3 dan P_4
3. P_1, P_2, P_3 dan P_4 secara simultan meng-komunikasikan x dengan P_5, P_6, P_7 dan P_8 .

Dan seterusnya hingga semua prosesor mengenal x .

Proses ini dilakukan dalam $\log N$ langkah.

-. Searching, File dimana x akan dicari dibagi ke dalam sub file dan secara simultan dilakukan pencarian oleh prosesor-prosesor :

P_1 mencari pada $\frac{n}{N}$ entri pertama,

P_2 mencari pada n/N entri kedua,
 P_3 mencari pada n/N entri ketiga,
 P_N mencari pada n/N entri ke-N.
 Proses ini membutuhkan n/N langkah.

Jadi total langkah yang dibutuhkan oleh algoritma tersebut adalah : $\log N + n/N$ langkah.

4.1.2. Speedup

Pengukuran speedup sebuah algoritma paralel adalah salah satu cara untuk mengevaluasi kinerja algoritma tersebut.

Speedup adalah perbandingan antara waktu yang diperlukan algoritma sekuensial yang paling efisien untuk melakukan komputasi dengan waktu yang dibutuhkan untuk melakukan komputasi yang sama pada sebuah mesin pipeline atau paralel.

$$\text{Speedup} = \frac{\text{Worst case running time dari algoritma sekuensial terefisien}}{\text{Worst case running time dari algoritma paralel}}$$

Contoh 4.2. :

Dari contoh 4.1.

Running time proses searching dengan mesin sekuensial adalah $O(n)$.

Running time dari proses searching pada komputer EREW SM SIMD adalah $O(n/N)$.

Jadi speedup = $O(N)$.

4.2. Banyaknya Prosesor (Number of Processor)

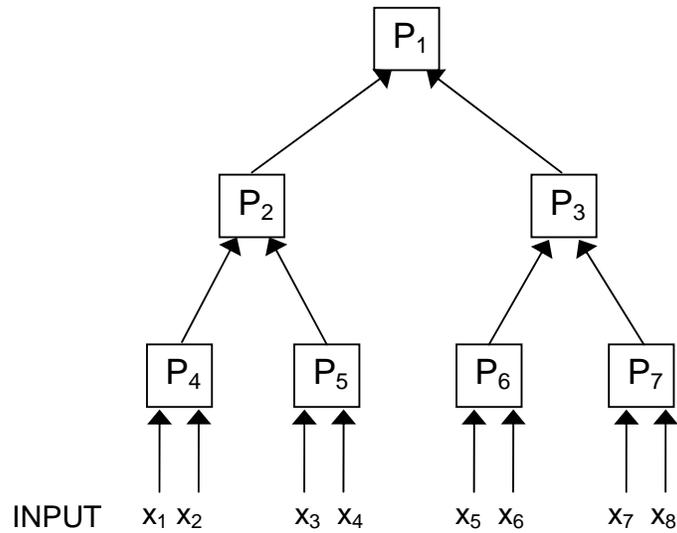
Semakin banyak prosesor yang digunakan semakin tinggi biaya untuk memperoleh solusi sebuah problem. Hal ini terjadi karena perlu dipertimbangkan biaya pengadaan prosesor dan perawatannya.

Jumlah prosesor yang tergantung dari n , n =ukuran problem, dinyatakan sebagai $p(n)$. Kadang-kadang jumlah prosesor tidak tergantung pada ukuran problem.

Contoh 4.3. :

Perhatikan n bilangan x_1, x_2, \dots, x_n yang akan dijumlahkan.

Dengan menggunakan komputer tree-connected SIMD dengan $\log n$ level dan $n/2$ daun, dibutuhkan pohon dengan ukuran $(n-1)$ atau $p(n) = n - 1$.
Ilustrasi untuk $n = 8$.



Sedangkan pada contoh 4.1. , banyaknya prosesor, N , tidak tergantung pada ukuran problem, n .