 <p><b>MODUL IV</b> <b>CCS 210 SISTEM OPERASI</b></p>		
<b>Judul</b>	<b>SISTEM PROSESOR DAN METODE PEMROSESAN JOB</b>	
<b>Penyusun</b>	<b>Distribusi</b>	<b>Perkuliahan</b>
<b>Nixon Erzed</b>	<b>FASILKOM</b> UNIVERSITAS ESA UNGGUL	Pertemuan – IV ON LINE

**Tujuan :**

Mahasiswa mengetahui sistem prosesor, metode-metode pemrosesan job dan memahami bagaimana metode tersebut diimplementasikan dalam sistem prosesor

**Materi:**

Sistem Prosesor

- Sistem Monoprosesor
- Sistem Multiprosesor
- Sistem Prosesor Terdistribusi

Metode Pemrosesan Job

- Batch System
- Multiprocessing
- Distributed Processing
- Multitasking
  - Multiprogramming
  - Timesharing
- Real Time processing
- Paralel Processing

**Referensi :**

1. Modern Operating System 3th Edition Andrew S Tanembaun 2009
2. Operating System, Internals and design Principles, William Stallings 7<sup>th</sup> Ed. 2012
3. Operating System Concepts, Abraham Silberschatz, 9th Ed, 2012
4. Sistem Operasi, Bambang Haryanto, Rev.5 2012
5. Arsitektur dan Organisasi Komputer, William Stalling, Prehalindo

## SISTEM KOMPUTER

Dalam konteks pembahasan pemrosesan job, sistem komputer dibedakan berdasarkan struktur prosesor yang dimiliki, yaitu sistem monoprosesor, sistem multiprosesor, dan sistem proses terdistribusi.

Prosesor merupakan bagian utama dari sistem komputer karena processor berfungsi untuk mengontrol keseluruhan jalannya sebuah sistem komputer dan digunakan sebagai pusat operasi dengan fungsi melakukan perhitungan dan menjalankan task. Kecepatan eksekusi processor tergantung apalagi pada frekuensinya, satuan adalah MHz (MegaHertz) atau GHz (1 GigaHertz = 1000 MegaHertz).

Untuk mengingatkan kembali komponen dasar prosesor adalah sebagai berikut:

**Unit Kontrol** adalah elemen CPU bertanggung jawab mengontrol komputer sehingga terjadi sinkronisasi kerja antarkomponen dalam menjalankan fungsi-fungsi operasinya. Tugas dari unit kendali ini antara lain:

- Mengatur dan mengendalikan alat-alat masukan (input) dan keluaran (output).
- Mengambil instruksi-instruksi dari memori utama.
- Mengambil data dari memori utama (jika diperlukan) untuk diproses.
- Mengirim instruksi ke ALU bila ada perhitungan aritmetika atau perbandingan logika serta mengawasi kerja dari ALU.
- Menyimpan hasil proses ke memori utama.

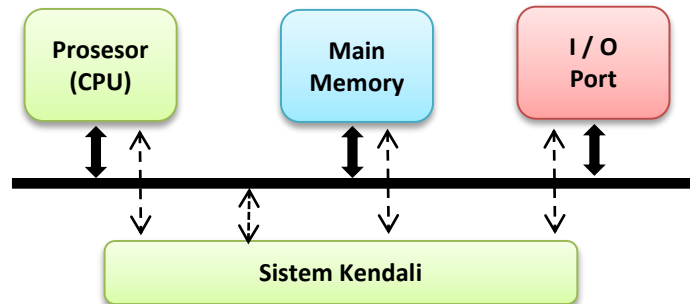
**Register** merupakan alat penyimpanan kecil yang mempunyai kecepatan akses cukup tinggi, yang digunakan untuk menyimpan data dan/atau instruksi yang sedang diproses. Memori ini bersifat sementara, biasanya digunakan untuk menyimpan data saat diolah ataupun data untuk pengolahan selanjutnya. Secara analogi, register ini dapat diibaratkan sebagai ingatan di otak bila kita melakukan pengolahan data secara manual, sehingga otak dapat diibaratkan sebagai CPU, yang berisi ingatan-ingatan, satuan kendali yang mengatur seluruh kegiatan tubuh dan mempunyai tempat untuk melakukan perhitungan dan perbandingan logika.

**ALU**, unit yang bertugas untuk melakukan operasi aritmetika dan operasi logika berdasar instruksi yang ditentukan. ALU sering di sebut mesin bahasa karena bagian ini ALU terdiri dari dua bagian, yaitu unit arithmetika dan unit logika boolean yang masing-masing memiliki spesifikasi tugas tersendiri. Tugas utama dari ALU adalah melakukan semua perhitungan aritmetika yang terjadi sesuai dengan instruksi program. ALU melakukan semua operasi aritmetika dengan dasar penjumlahan sehingga sirkuit elektronik yang digunakan disebut adder.

**CPU Interconnections** adalah sistem koneksi dan bus yang menghubungkan komponen internal CPU, yaitu ALU, unit kontrol dan register-register dan juga dengan bus-bus eksternal CPU yang menghubungkan dengan sistem lainnya, seperti memori utama, peranti masukan /keluaran.

### Sistem Monoprosesor

Sistem Monoprosesor adalah sistem komputer dengan prosesor tunggal, yang melayani pemrosesan job dengan cara : monoprogramming, multiprogramming dan timesharing.

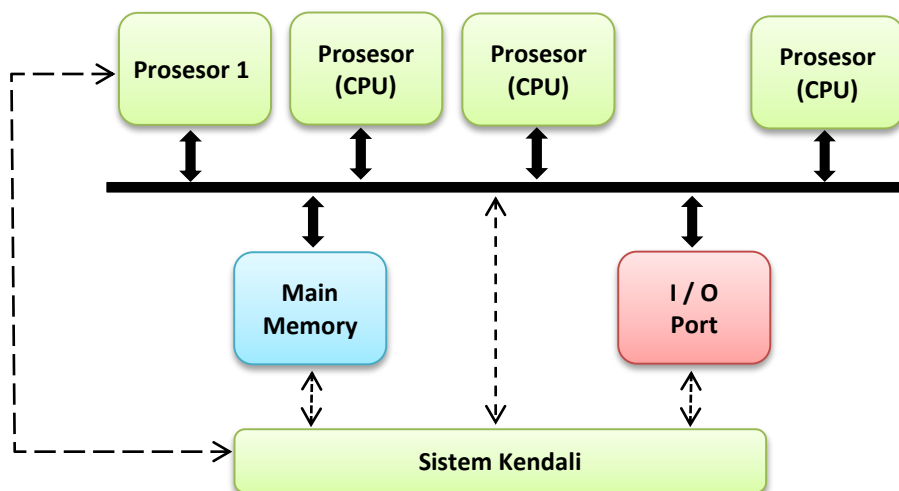


Dalam implementasi monoprogramming seluruh sumber daya sistem komputer akan dialokasikan untuk job/program berjalan. Sedangkan pada multiprogramming dan time sharing harus dilakukan penjadwalan yang ketat sehingga seluruh job dapat dilayani

### Sistem Multiprosesor

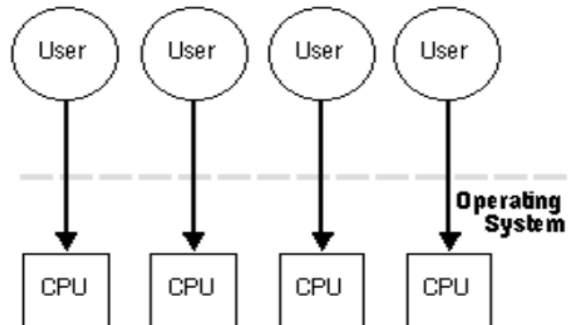
Sistem Multiprosesor adalah sistem komputer yang dilayani oleh sekumpulan prosesor. Processor set tersebut dapat berupa kumpulan prosesor yang bertipe dan berkemampuan sama (symetric multiprocessor) atau kumpulan prosesor dengan tipe dan kemampuan berbeda-beda (asymetric multiprocessor).

Processor set tersebut terpasang pada sistem mainboard yang sama dan bekerja di ruang memory yang sama dengan sistem kendali yang sama



### Symetric Multiprocessor

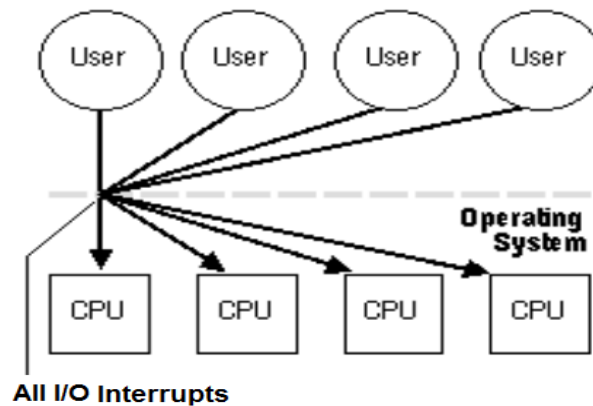
Pada symetric multiprocessor semua prosesor bertype dan berkemampuan sama sehingga job membutuhkan layanan dapan dilayani oleh sembarang prosesor.



### Asymetric Multiprocessor

Pada asymetric multiprocessor, terdapat prosesor yang difungsikan sebagai master processor dan prosesor lain nya difungsikan sebagai slave.

Job yang datang akan diatur oleh master processor untuk dikerjakan oleh prosesor yang sesuai.

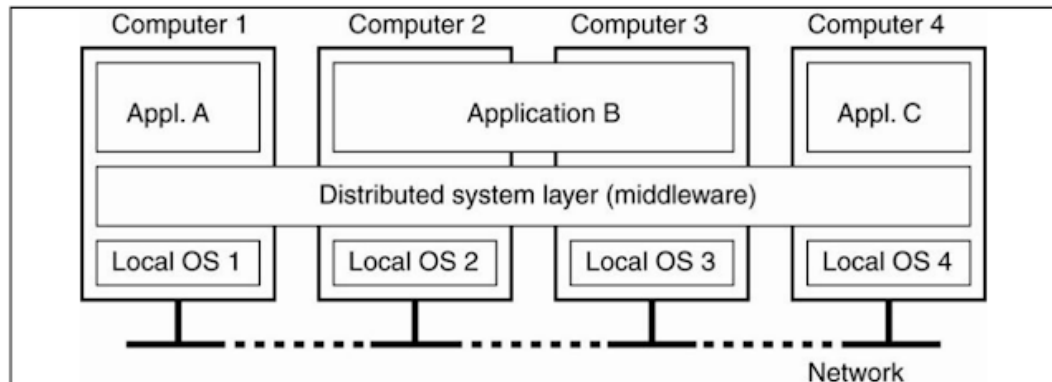


### Sistem Prosesor Terdistribusi

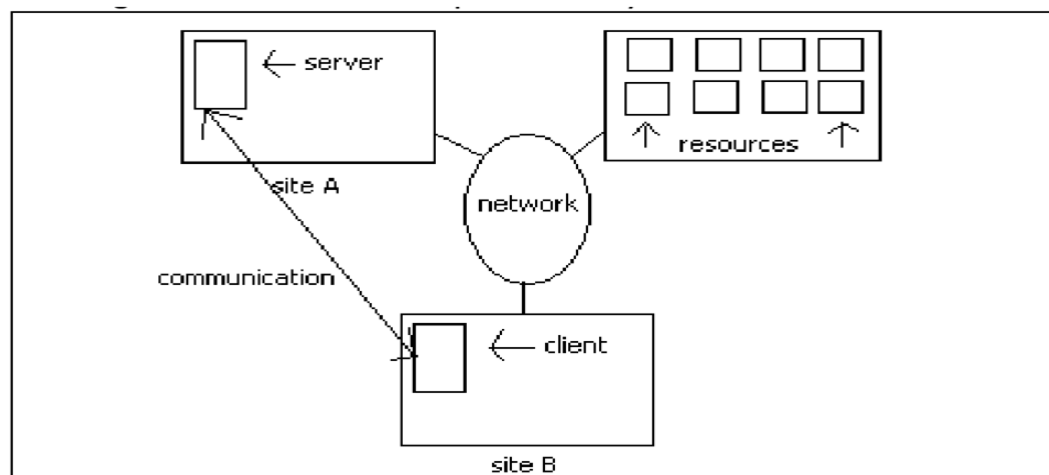
Sistem Prosesor Terdistribusi dapat dipandang sebagai sistem komputer yang dibentuk oleh sekumpulan komputer otonom yang diintegrasikan, sehingga terlihat oleh user sebagai sebuah sistem komputer.

Setiap komputer akan memiliki satu prosesor yang terpasang pada satu sistem mainboard dan sistem memory, dan setiap komputer memiliki sistem kendali dan sistem clock sendiri. Untuk Sistem kendali dari sistem Komputer Terdistribusi berada disebuah perangkat perantara (Middleware) merupakan komponen perantara yang memungkinkan client dan server (lapisan aplikasi dan sistem operasi) saling terhubung dan berkomunikasi satu sama lain. Tujuan utama

layanan middleware adalah untuk membantu memecahkan interkoneksi beberapa aplikasi dan masalah interoperabilitas.



Struktur sistem terdistribusi dapat digambarkan sebagai berikut:



Sistem terdistribusi walaupun rumit pengelolaannya, merupakan sistem yang sangat fleksibel untuk dikembangkan dan mampu memberi layanan yang luas dengan biaya yang murah. Secara lebih rinci berikut adalah fleksibilitas yang ditawarkan :

- Membagi beban kerja pada perangkat yang tersedia dengan cara yang efektif.
- Dapat menambah komponen secara individu tanpa harus menduplikasi sistem
- Fasilitas local dapat disesuaikan dengan kebutuhan local
- Memungkinkan pertumbuhan sistem secara terus menerus.
- Susunan sistem bisa disesuaikan dengan pola organisasi perusahaan
- Memungkinkan beberapa bagian/local mengadakan percobaan konsep baru dan fasilitas baru untuk mengurangi resiko kegagalan sistem secara keseluruhan

## METODA PEMROSESAN JOB / TASK

Sistem komputer memiliki banyak cara dalam memberikan tanggapan terhadap permintaan eksekusi suatu program. Setiap metoda memiliki karakteristik yang khas yang cocok dengan sifat-sifat atau kebutuhan proses. Metoda juga dikembangkan mengikuti spesifikasi perangkat keras.

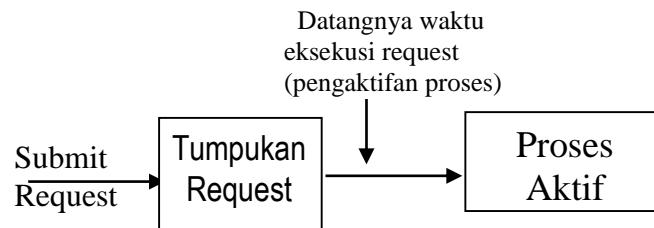
### 1. BATCH SYSTEM

Pada dasarnya Batch system merupakan **pengumpulan** dari job-job dalam satu periode pemrosesan.

→ alasan2 implementasi batch system :

mengoptimalkan pemanfaatan sumber daya system  
menunggu kelengkapan proses →

1. berdasarkan waktu/periode  
sistem akuntansi, sub proses posting  
→ input data transaksi ke buku besar (general ledger)  
→ data-data transaksi ditumpuk dalam jurnal sebelum diposting
2. berdasarkan kuantitas  
menunggu lengkapnya parameter proses



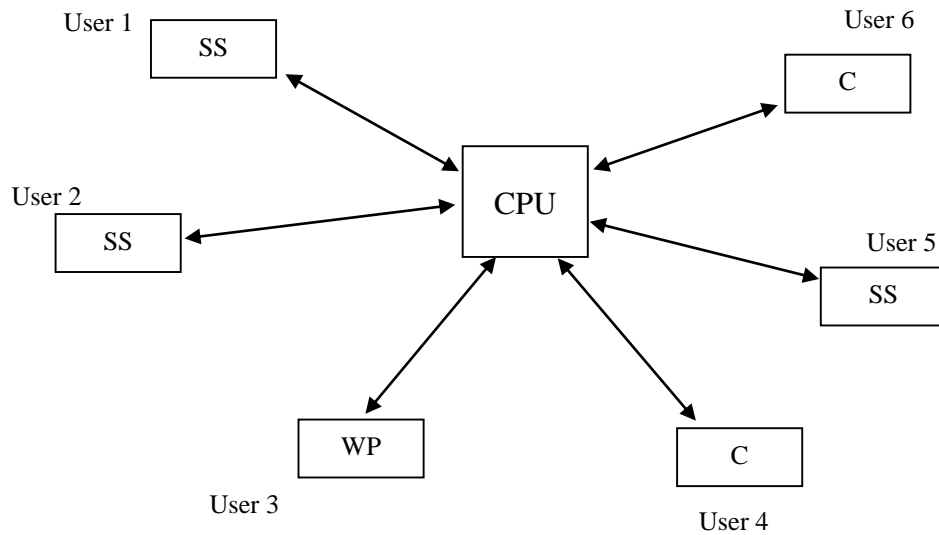
Contoh batch system populer :

- ✘ Resident Monitor
- ✘ Overlap operasi I/O dengan CPU

### Resident monitor

Resident monitor adalah program kecil yang selalu residen di memory, yang akan mem-batch request eksekusi job/proses dari user. Job-job yang ditumpuk terlebih dahulu akan diurutkan sebelum diproses oleh CPU. Tujuannya untuk mereduksi waktu menunggu CPU.

Resident Monitor akan memindahkan/mengatur kendali antar job.



Jika semua User pada saat yang hampir bersamaan merequest eksekusi proses, maka → resident monitor akan mengurutkan terlebih dahulu, sehingga eksekusi untuk proses yang sama dapat dilakukan secara berurutan → mereduksi waktu yang dibutuhkan untuk switch antar proses → backup dan restore status proses

Proses yang sama tidak memerlukan reset status CPU secara total

Reset penuh → backup data lama, backup aplikasi lama, restore data baru dan aplikasi baru

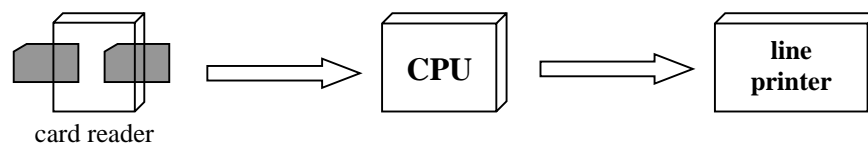
Dengan resident monitor kebutuhan waktu prosesor untuk backup dan restore aplikasi dapat direduksi

### Overlap operasi I/O dengan CPU

Untuk mengatasi ketidak seimbangan kecepatan I/O device dengan kecepatan CPU, maka kebutuhan layanan operasi I/O tidak secara langsung dikelola oleh CPU, tapi terlebih dahulu akan ditumpuk. Ketika I/O melakukan pengaksesan tumpukan (store/get) data, maka CPU akan mengerjakan/melayani proses lain.

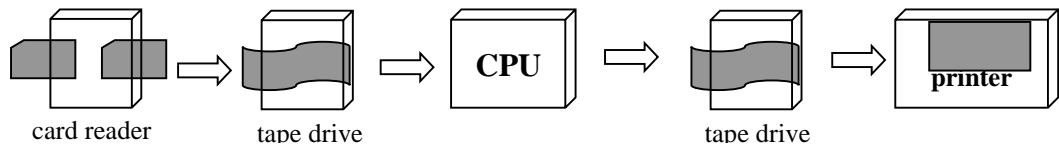
Berikut ini adalah ilustrasi overlap operasi I/O dengan CPU

⌘ **on-line processing** (*tidak diterapkan overlap operasi*)



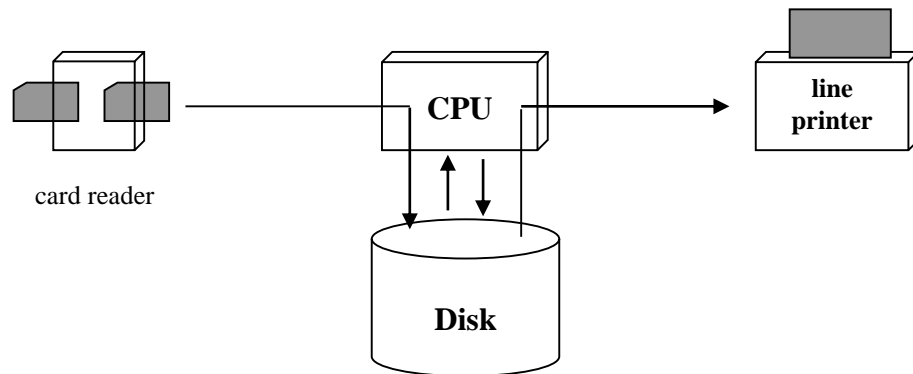
(a) On line processing

- ✘ **off-line processing** (menerapkan overlap operasi)



(b) Off line processing

- ✘ **Spooling** (overlap operasi pada komputer modern)



## 2. Multiprocessing System

→ multitasking yang dikerjakan dengan mesin multiprocessor

Mesin multiprocessor : CPU dengan lebih dari satu prosesor, tapi 1 memory, system bus dan lain-lain.

Dibedakan atas :

### a. Symetric multiprocessor

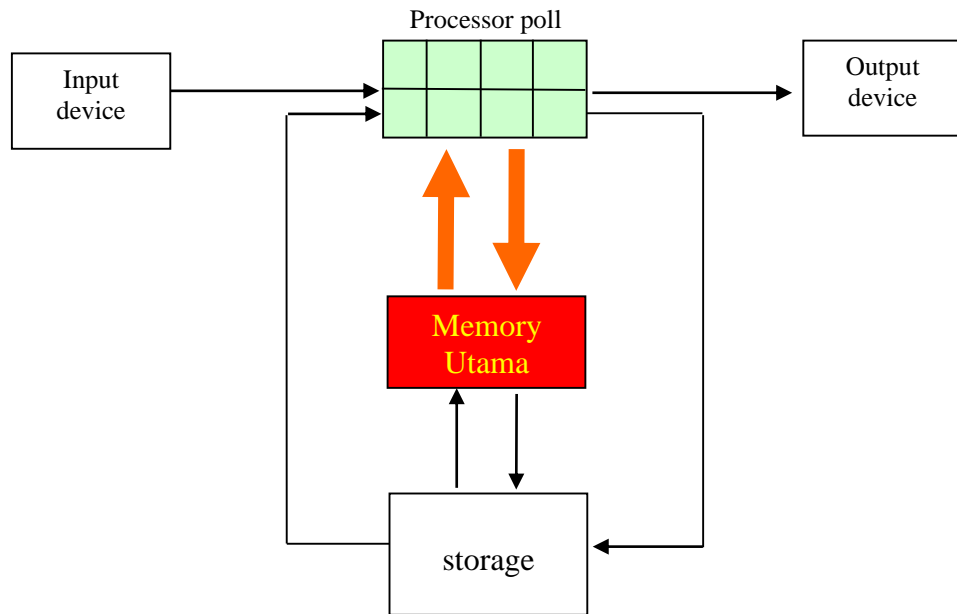
Tiap-tiap pemroses mempunyai model, kemampuan dan fungsi yang sama.

Dalam pelaksanaan proses → proses dapat dialokasikan ke sembarang prosesor

### b. Asymetric multiprocessor.

Sebuah pemroses berfungsi sebagai master processor, dan pemroses yang lain berfungsi sebagai slave.



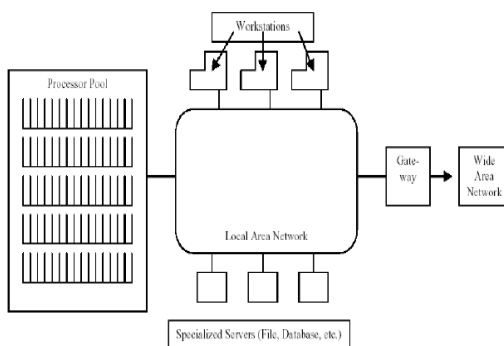


### 3. Distributed processing System

- multitasking yang dikerjakan dengan mesin distributed processor
- distributed processor** : Kumpulan prosesor yang tidak menggunakan memory atau clock secara bersama-sama (berada dalam sistem-sistem/CPU-CPU yang terpisah)
- sekumpulan system computer dengan satu system operasi

Sistem Terdistribusi → menggunakan OS yang khas

Misalnya : AMOEBA, Angel dsb



#### 4. MULTITASKING

Lingkup pemrosesan multitasking:

	Mono processor	Multi processor	Ditributed processor
Monoprogramming	√	√ Paralel processing unt sebuah proses kompleks	X
Multi tasking { Multiprogrammig	√	√	√
Time sharing	√	√	√

##### Monoprocessor :

1  $\mu$ P, 1 sistem memory, 1 sistem bus eksternal, 1 sistem clock

##### Multiprocessor :

Bebearapa  $\mu$ P, 1 sistem memory, 1 sistem bus eksternal, 1 sistem clock

##### Distributed processor :

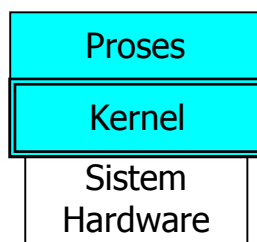
Bebearapa  $\mu$ P yang masing-masing memiliki : sistem memory, sistem bus eksternal, dan sistem clock

#### 4.1 Monoprogramming pada monoprosesor:

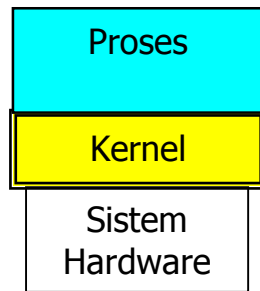
- o hanya ada satu proses yang dilayani oleh sistem  
→ seluruh sumber daya teralokasi untuk proses tersebut
- o pada dasarnya selain proses juga aktif dalam sistem komputer kernel OS yang dibundel kedalam proses  
→ sumber daya sistem dikendalikan oleh proses

Monoprogramming diimplementasikan dengan 2 cara :

- a. Proses membundel fungsi-fungsi OS kedalam dirinya



- b. Fungsi-fungsi OS menjadi interface antara proses dengan Hardware



#### 4.2 Multiprogramming >< Timesharing

Metode pemrosesan job yang memungkinkan pada saat yang sama terdapat lebih dari satu proses yang membutuhkan layanan sistem

- o multiprogramming
- o timesharing

a. Multiprogramming System

berbasiskan program, ketika sebuah proses mendapat jatah waktu pemroses, maka akan dikerjakan sampai tuntas

b. Time Sharing System

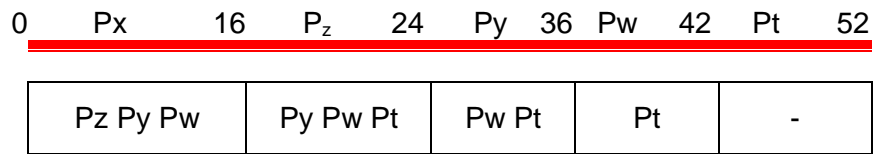
berbasiskan sub program (thread), ketika sebuah proses mendapat jatah waktu pemroses, maka akan dikerjakan selama batas waktu tertentu → Waktu Kuanta (quantum time)

Jika batas waktu habis → proses ditunda, dan prosesor diberikan pada proses lain.

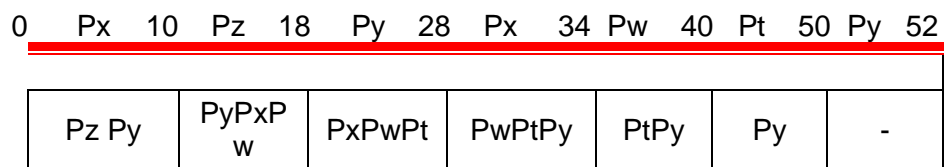
Id proses	Waktu Kedatangan <i>log-on time</i> (LT)	Estimasi Waktu Proses (RTE)
P <sub>x</sub>	0	16
P <sub>z</sub>	2	8
P <sub>y</sub>	2	12
P <sub>w</sub>	12	6
P <sub>t</sub>	18	10

Simulasi jika multiprogramming dan time sharing yang digunakan dengan prinsip FIFO

**Multiprogramming**



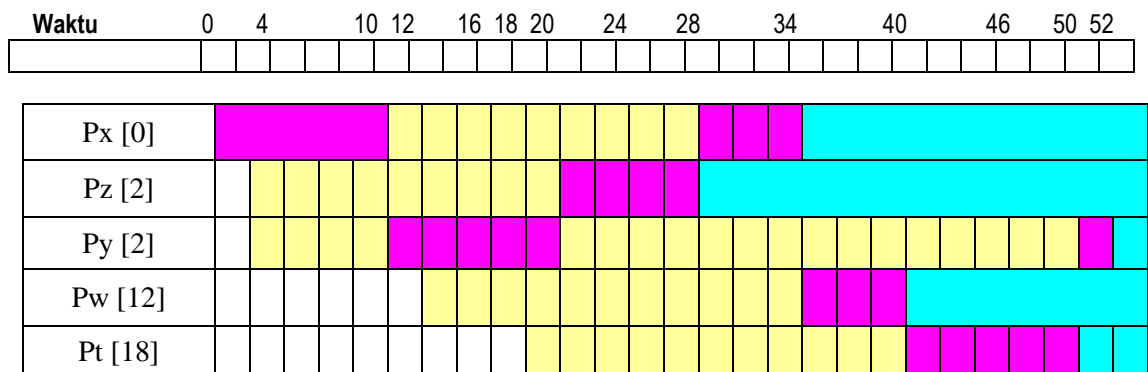
**Time sharing dengan QT =10**



Gant chart dari pengelolaan proses tersebut dapat disajikan sbb:



**Time sharing dengan QT = 10**



### 4.3 Model Multitasking

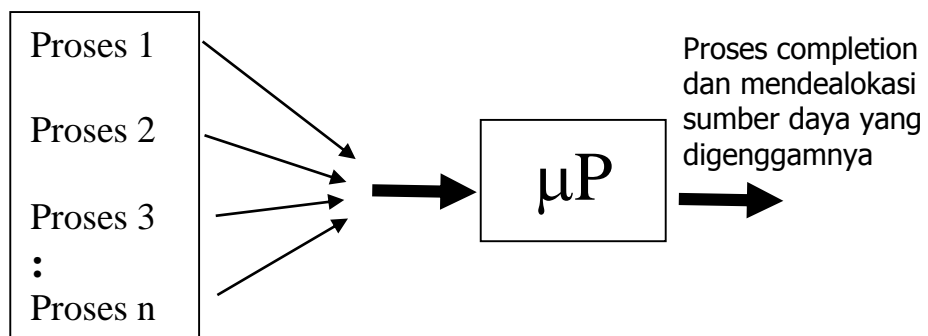
#### 4.3.1 Multiprogramming System

→ implementasi penjadwalan non preemptive

##### a. Pada Monoprocessor

→ terdapat antrian proses untuk menunggu alokasi waktu pemroses

→ begitu sebuah proses mendapat alokasi waktu pemroses, akan dikerjakan hingga selesai dan kemudian men-dealokasi seluruh sumber daya sistem yang dikuasainya



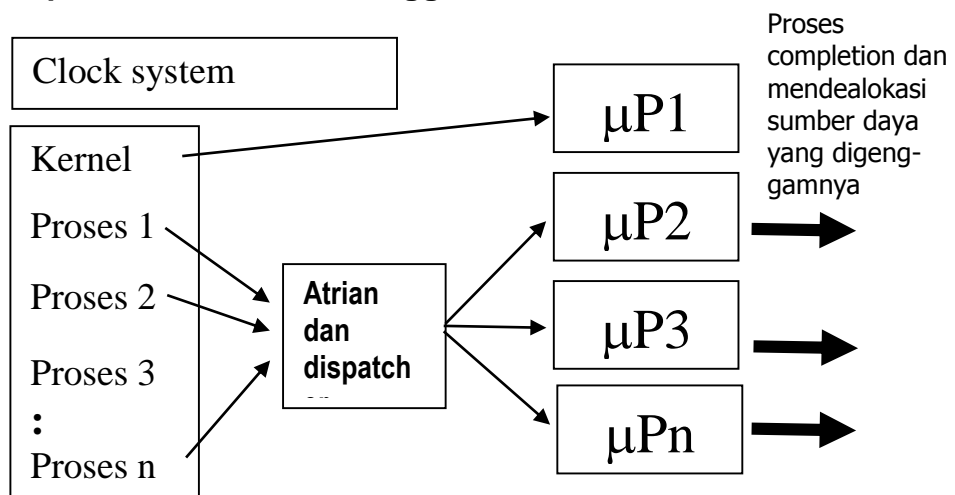
##### b. Pada multiprocessor

Pada umumnya akan dialokasikan sebuah pemroses untuk mengeksekusi kernel → master prosesor. Proses-proses akan antri untuk mendapatkan alokasi waktu pemroses :

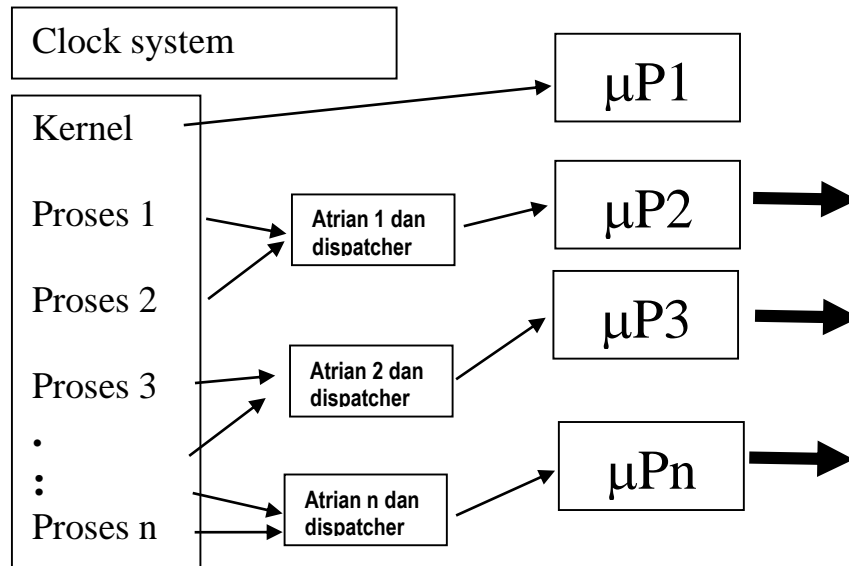
→ [antrian tunggal](#)

→ [banyak antrian](#); ada mekanisme untuk memindahkan proses antar antrian

#### Implementasi Antrian Tunggal :



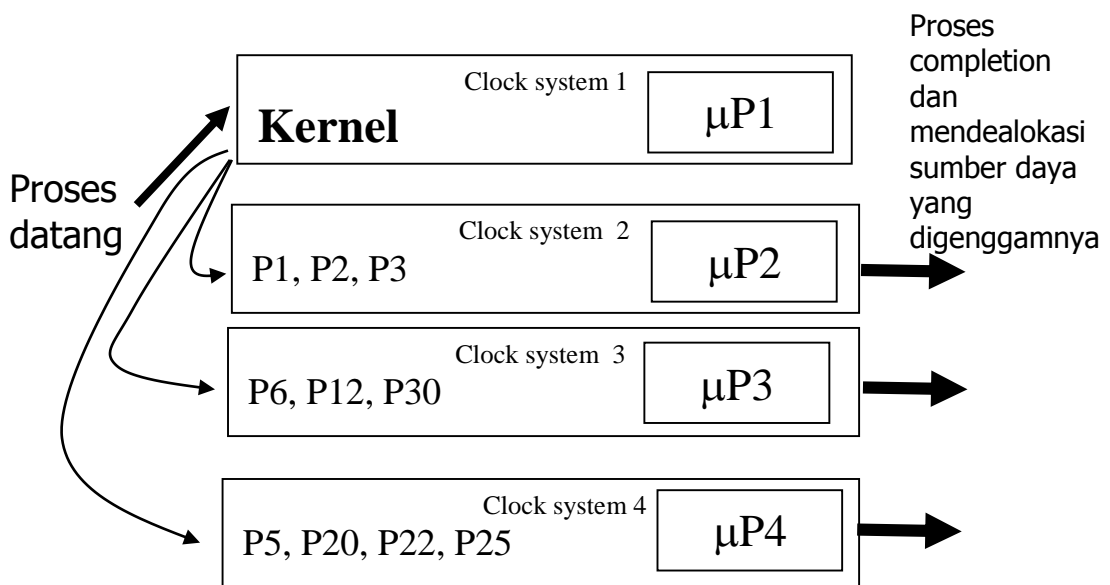
### Implementasi Multi Antrian



#### c. Pada distributed processor

Mirip dengan multiprosesor, perbedaannya adalah "lokasi" antrian

→ resiko terjadinya keadaan terdapat mesin yang "kosong" sementara pada mesin yang lain terdapat antrian → [masalah loadbalancing](#)

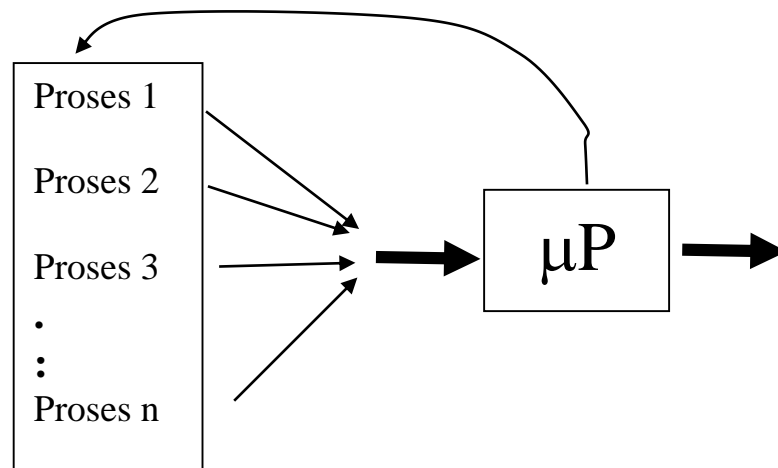


### 4.3.2. Time Sharing System

→ implementasi penjadwalan preemptive

#### a. Pada monoprocessor

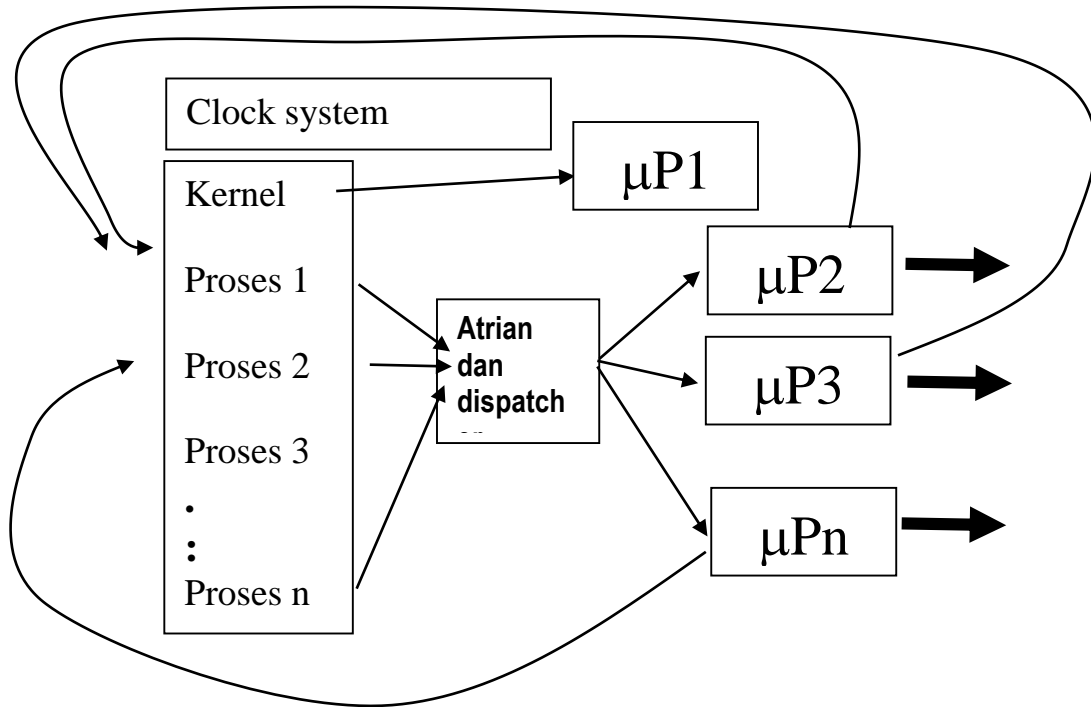
→ terdapat antrian proses untuk menunggu alokasi waktu pemroses  
→ setiap proses akan dapat alokasi waktu pemroses secara terbatas dan kembali masuk antrian jika jatah waktunya habis sementara jobnya belum selesai



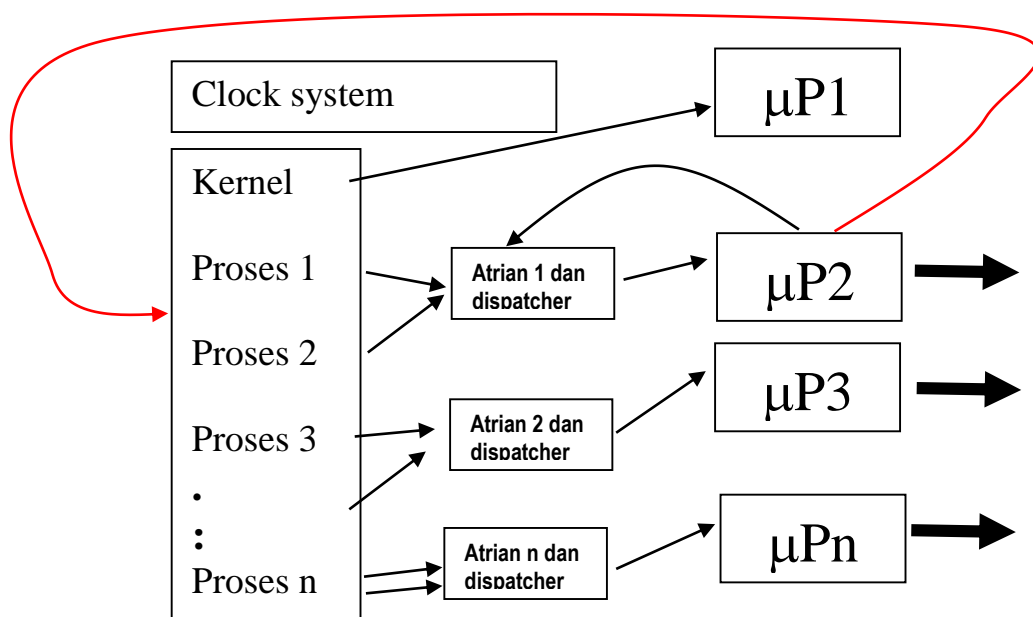
#### b. Pada multiprocessor

→ Proses-proses akan antri untuk mendapatkan alokasi waktu pemroses  
→ model antrian antrian tunggal → proses kena time out akan masuk kembali ke antrian  
→ pada model banyak Antrian  
    Untuk proses yang kena time out → terdapat dua kemungkinan kembali ke antrian semula atau dipindah ke antrian lain

### Implementasi Antrian Tunggal



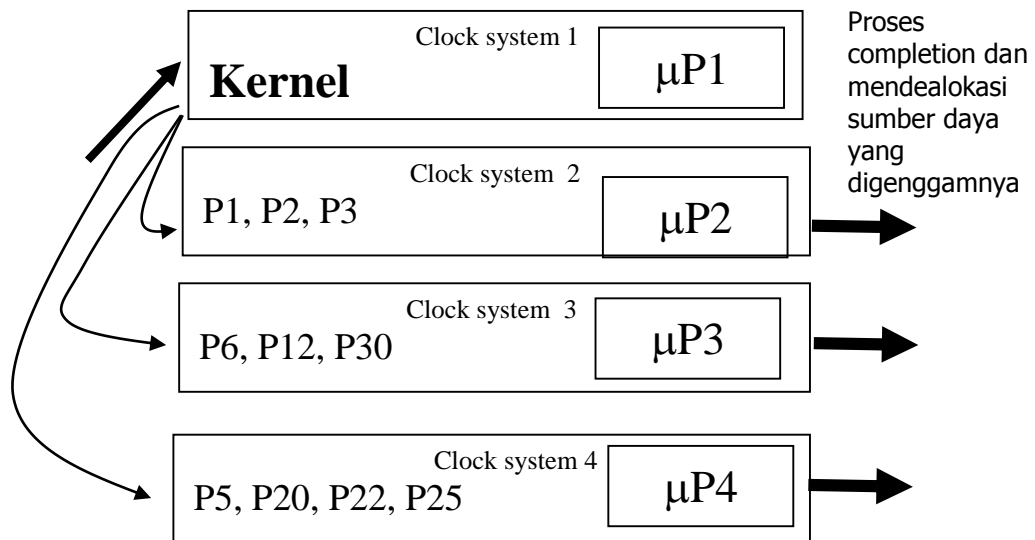
### Implementasi Multi Antrian





### c. Pada distributed processor

- proses yang “time-out”, secara fisik mesti kembali keantrian semula
- untuk memisahkan proses “time out” kemesin yang berbeda → arsitektur prosesor harus indentik



## 5. MONOPROGRAMMING PADA MULTIPROCESSOR

- melakukan pemrosesan paralel terhadap sebuah program yang kompleks
- proses dibagi menjadi sub-sub proses yang relatif independen dan menganalisa ketergantungan antar sub proses → untuk mendapatkan kemungkinan mengeksekusi sub proses tersebut secara bersamaan
- membangun graf keterdahuluan

## 6. REAL TIME SYSTEM

- sistem waktu seketika  
misal : ATM (update database account), aplikasi multimedia

Sistem yang diterapkan pada operasi yang menuntut ketepatan waktu dari prosesor atau data flow → waktu tunggu berada dalam limit tertentu

- sistem waktu seketika/real time system → Real Time Operating System (RTOS)