



# Sistem Operasi

[www.esaunggul.ac.id](http://www.esaunggul.ac.id)

HARDWARE

Pertemuan 3

Dosen Pengampu: Arief Ichwani (2018)

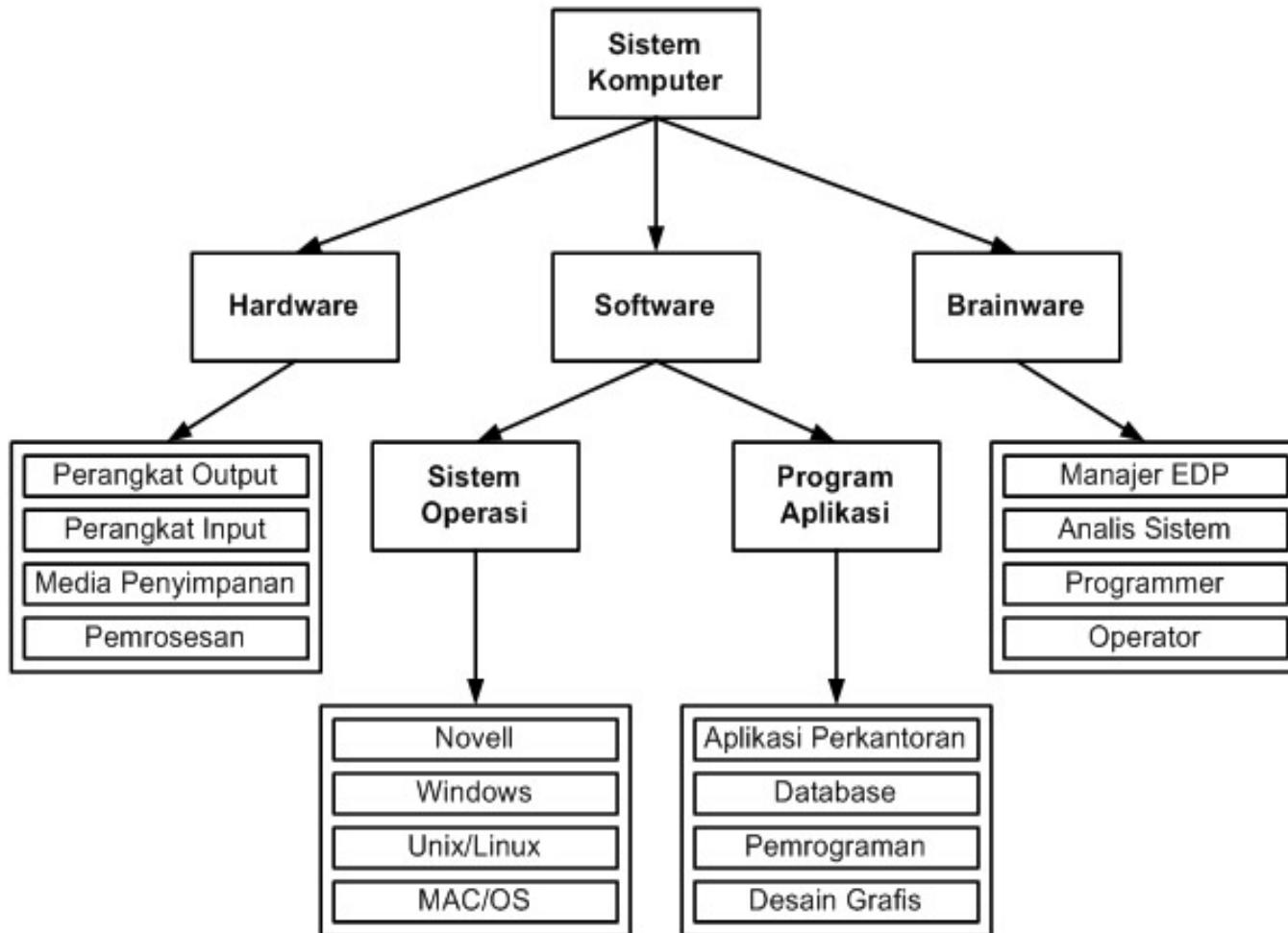
Prodi Sistem Informasi - Fakultas Ilmu Komputer

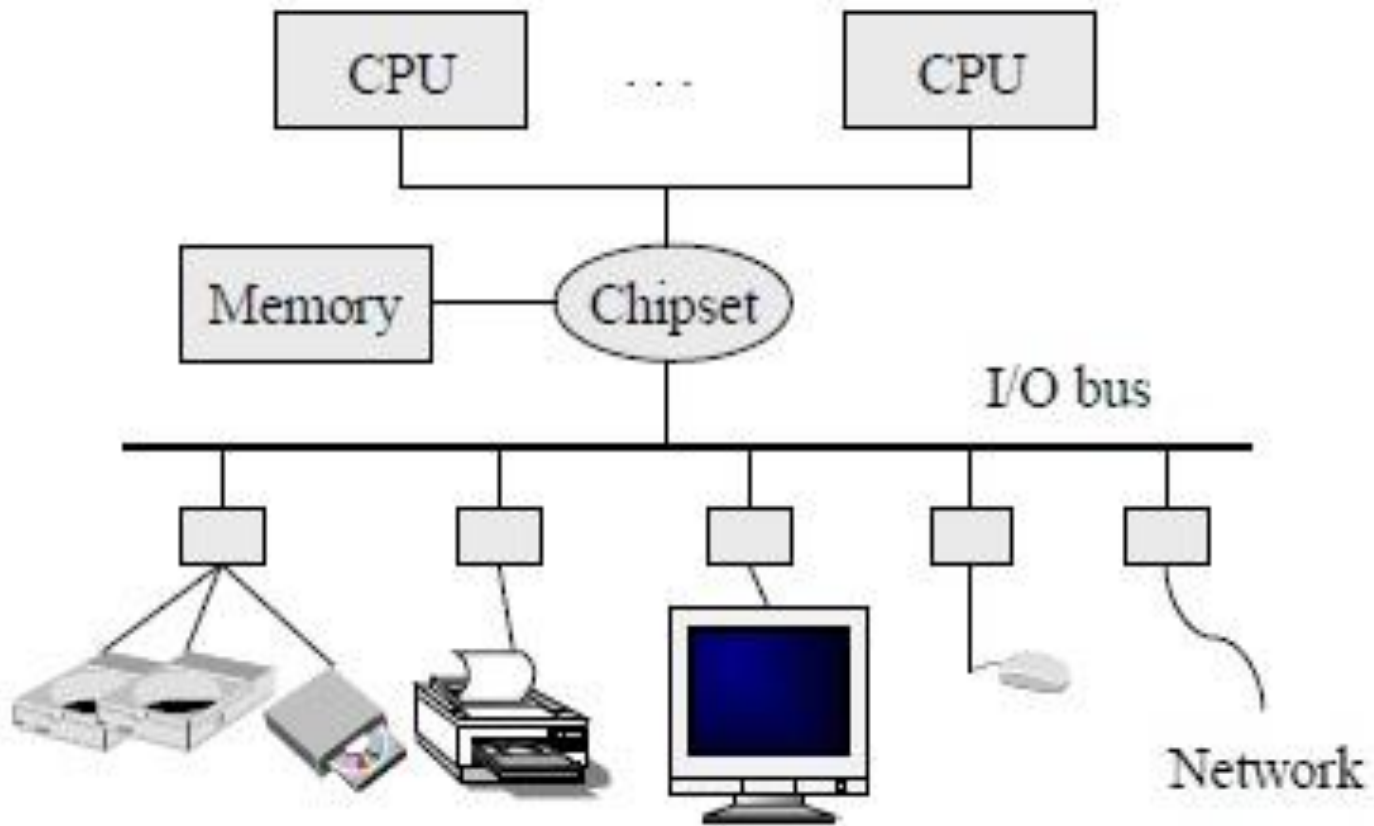
# Tujuan

- perangkat keras komputer dan bagaimana hubungannya dengan Melihat komponen-komponen an sistem operasi

# Overview

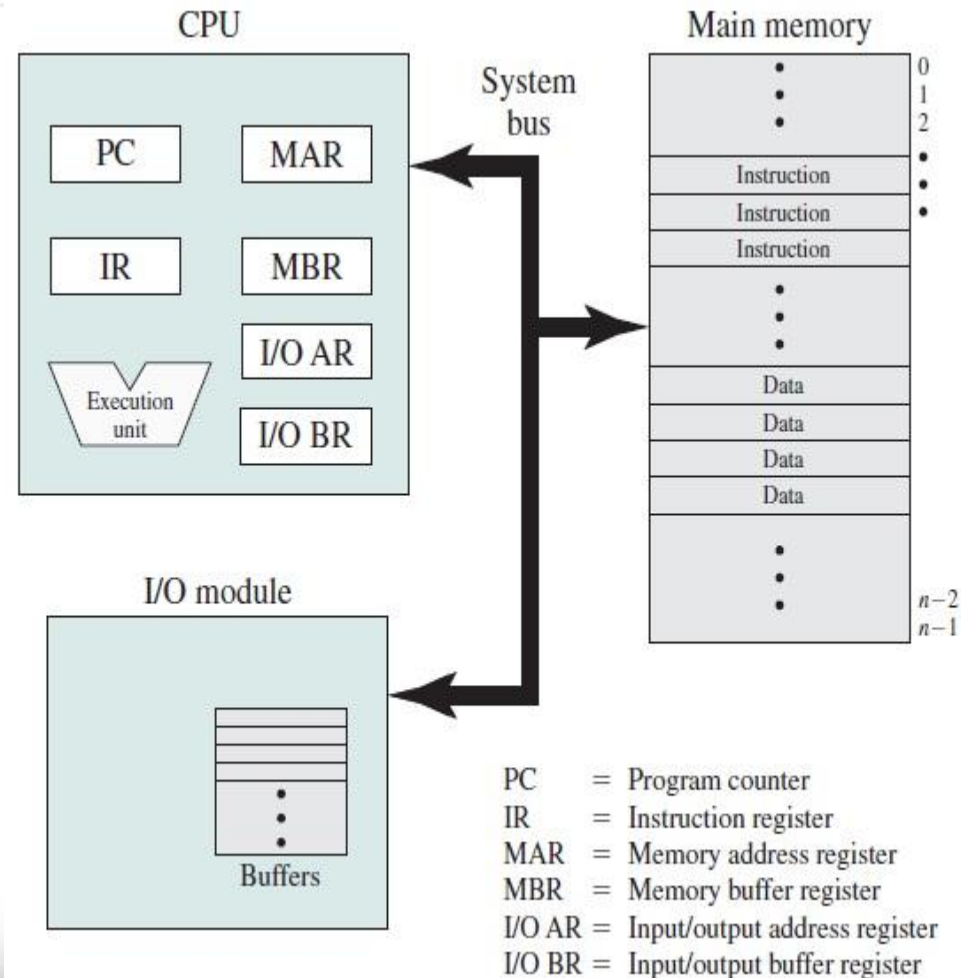
- **Bagian-bagian Perangkat Keras**
  - **CPU**
  - **Memory**
  - **Perangkat I/O**
  - **Bus**
- **Melakukan booting**





# Central Processing Unit(CPU)

- Otak komputer
- Mengeksekusi program: sekumpulan instruksi sekuensial
- Mengendalikan I/O



# Organisasi CPU terdiri dari

- **ALU** (Arithmetic and Logical Unit) : untuk melakukan komputasi atau pengolahan data aktual
- **CU** (Control Unit): untuk mengontrol perpindahan data dan instruksi ke/ dari CPU dan juga mengontrol ALU
- **Register** memory internal CPU merupakan jenis memori yang terdapat pada prosesor dan sebagai memori internal processor, kecepatan 5 sampai 10 kali di bandingkan memori utama, digunakan untuk menyimpan instruksi-instruksi dan data yang sedang diproses oleh CPU.

# Jenis dan Fungsi Register

Register terbagi dua yaitu :

1. *General Purpose Register (Register Serbaguna)*
2. *Pointer Register*
3. *Index Register*
4. *Segment Register*



## ➤ ***General Purpose Register (Register Serbaguna)***

### **Register AX (Accumulator register)**

berfungsi sebagai tempat Sementara hasil suatu operasi arithmetika atau logika

### **Register BX (Base Register)**

berfungsi untuk menyimpan alamat offset data yang terletak di memori

## ❑ Register CX (Counter Register)

- Counter register adalah register serbaguna yang berfungsi sebagai:
- Pencacah untuk operasi loop (CX dan ECX)
- Pencacah untuk operasi shift dan rotate (CL)
- Pencacah (counter) untuk operasi string (CX)

## ❑ Register DX (Data register)

- Data register adalah register serbaguna yang berfungsi sebagai :
- Penyimpan hasil perkalian 16 bit (DX-AX) dan 32 bit (EDX-EAX).
- Penyimpan hasil pembagian (DX-AX dan EDX-EAX)
- Penyimpan data hexadesimal (kode ASCII) di reg DL untuk dicetak di layar monitor.

➤ ***Pointer Register***

- Register ini untuk menunjukkan alamat sebuah data di lokasi memori, dipakai saat operasi perpindahan data (dari/ke memori), operasi stack (PUSH/POP) dan penunjukkan alamat suatu instruksi

➤ ***Index Register***

- Sama dengan pointer register, sering digunakan untuk menunjukkan alamat sebuah data di lokasi memori pada operasi string.

## ➤ *Segment Register*

Segment register membentuk alamat memori untuk data, yang termasuk ke dalam segment register antara lain :

- Code segment -> untuk menunjukkan alamat instruksi berikutnya.
- Data segment -> untuk menunjukkan alamat data pada transfer register
- Extra segment -> register tambahan untuk operasi string
- Stack segment -> untuk memanggil suatu prosedur dan mengarah ke program utama.
- FS dan GS register -> register tambahan u/ segmen memori yang besar.

# SIKLUS INSTRUKSI

Siklus instruksi adalah suatu siklus instruksi dasar yang dikerjakan oleh CPU di dalam melakukan eksekusi suatu instruksi.

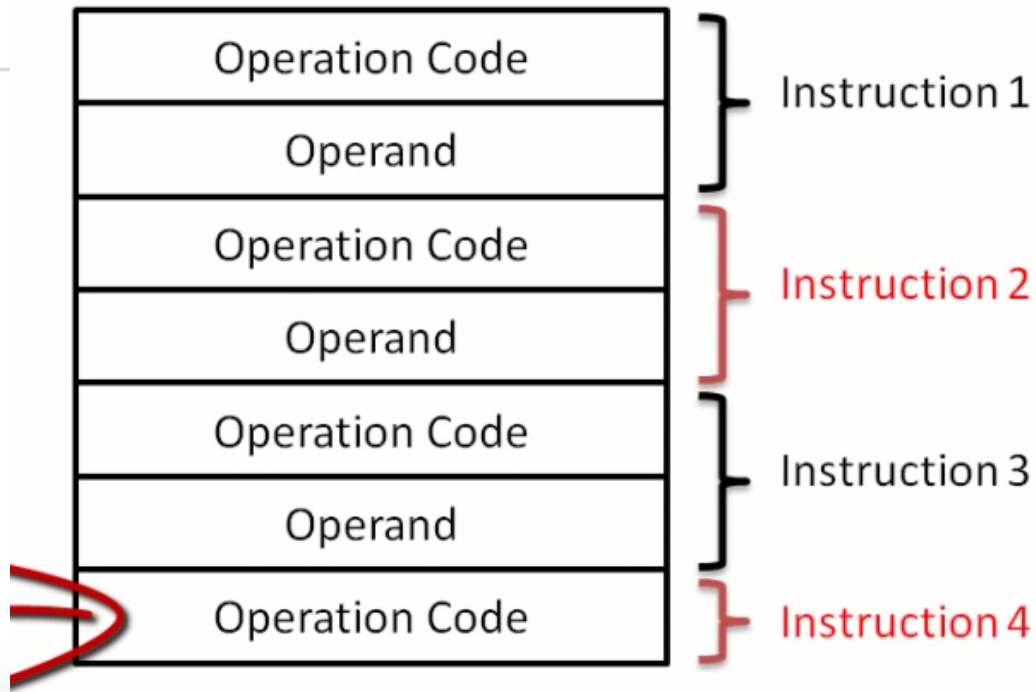
- Fetch : proses pengambilan data dari memory ke CPU terus menerus diulang oleh CPU selama komputer hidup atau membaca instruksi berikutnya dari memory kedalam CPU
- Decode : Tahap menterjemahkan instruksi kedalam perintah-perintah yang dapat dimengerti oleh komputer
- Execute : instruksi menjalankan program yang telah dikompilasi oleh CPU

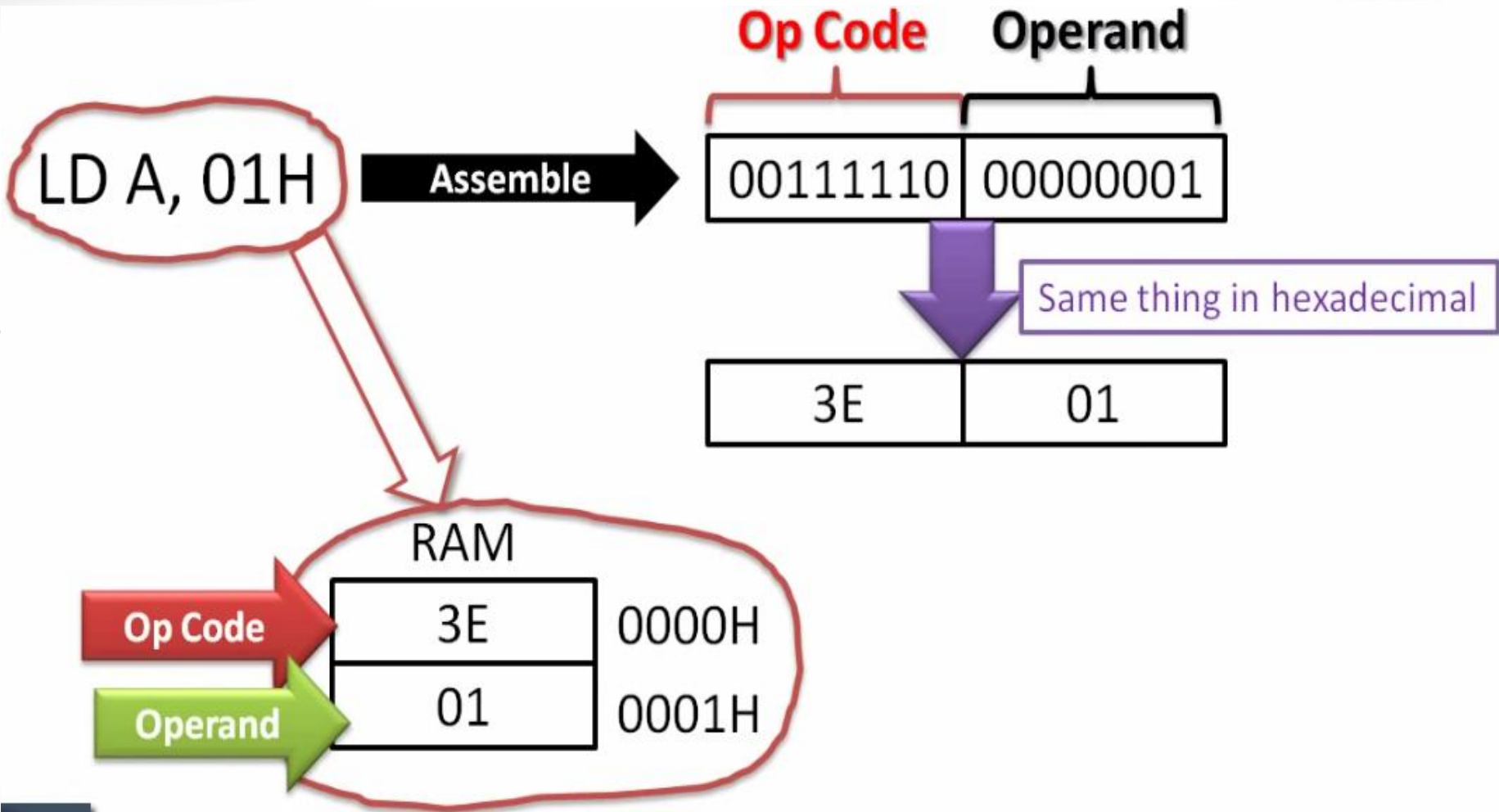


Instruction format

- 0001 = Load AC from memory
- 0010 = Store AC to memory
- 0101 = Add to AC from memory

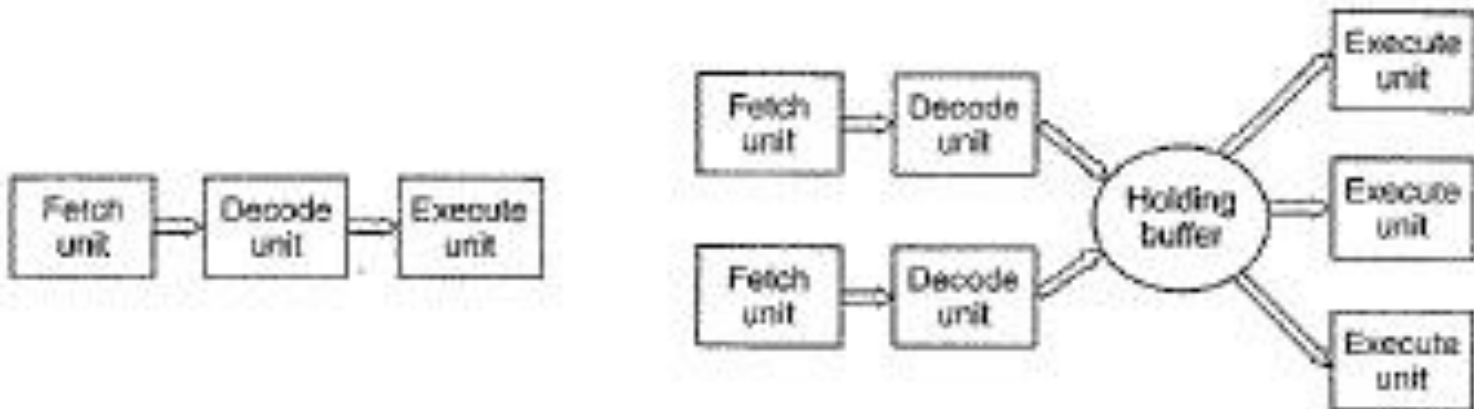
Partial list of opcodes





# Pipeline dan Superscalar (1)

- Beberapa teknologi untuk mempercepat proses dengan mengeksekusi lebih dari 1 instruksi tiap saat
- Pipeline: menyediakan 3 unit (dalam 1 line) dan dapat bekerja paralel
- Superscalar: membagi menjadi beberapa unit dalam beberapa line dengan dilengkapi buffer untuk menampung hasilnya





# Pipeline dan Superscalar (2)

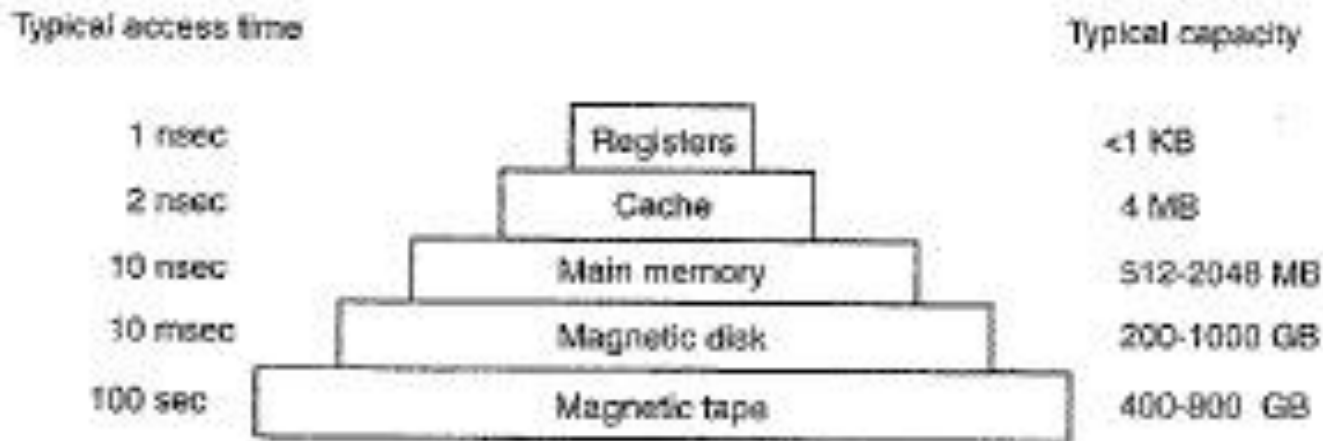
- Masalah pipeline: instruksi yang sudah diambil dari memory harus dijalankan, padahal ada instruksi percabangan
- Masalah superscalar: ketidakterurutan eksekusi instruksi

# Sirkuit atau jenis yang digunakan CPU

1. Program Counter (PC): berisi alamat lokasi memori dimana instruksi harus diambil.
2. Memory Address Register (MAR): register yang digunakan untuk menampung alamat data atau instruksi kepada memori utama yang akan diambil atau yang akan diletakan.
3. Memory Data Register (MDR): register dari unit kontrol komputer yang berisi data yang akan disimpan dalam penyimpanan komputer (misalnya RAM), atau data setelah mengambil dari penyimpanan komputer
4. Instruction Register (IR) : register yang menyimpan instruksi yang sedang dijalankan

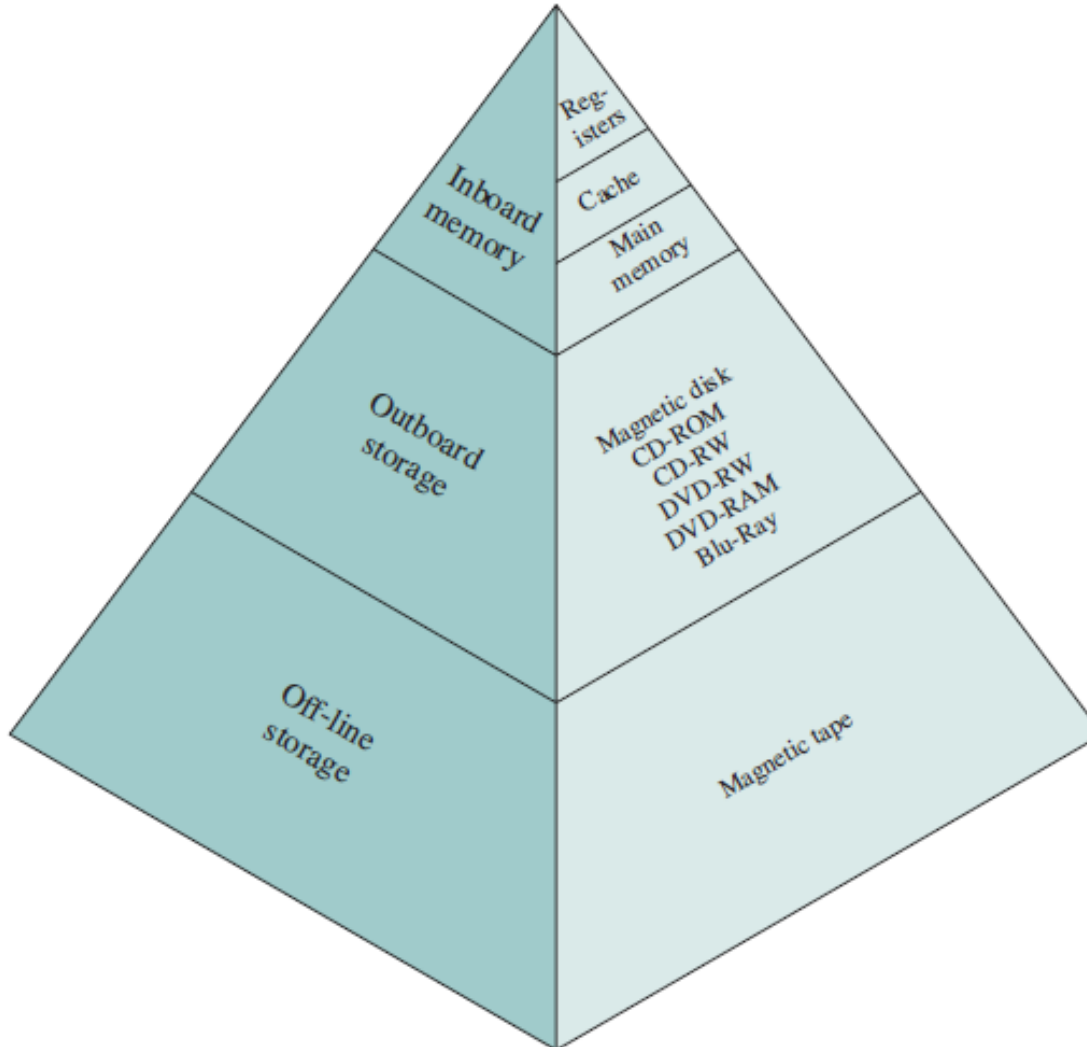
# Memory

- Memory ideal: cepat, kapasitas besar dan murah
- Biasanya hanya sebagian saja yang terpenuhi
- Solusi: hirarki memory



- Fungsi SO untuk mengatur transfer antar hirarki
- Data yang sering diakses diletakkan di memory yang lebih cepat

# Hirarki Memory



# Caching

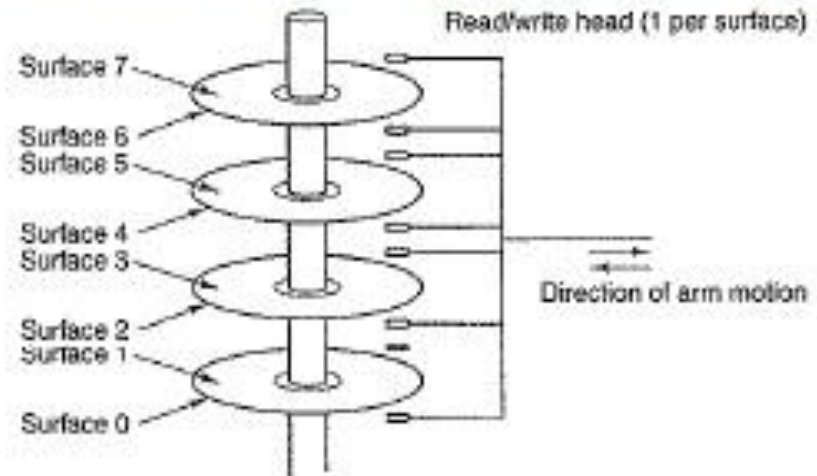
- Jika suatu memory word dibutuhkan SO akan memeriksa apakah ada di cache
- Jika ada → cache hit, maka memory utama tidak perlu diakses
- Jika tidak → cache miss, perlu mengakses memory utama yang jauh lebih lambat
- Cache juga dibuat berlevel, masing-masing level berbeda kecepatannya

# Memory Utama

- Volatility → hilangnya data yang disimpan setelah tiada arus listrik
- RAM, volatile
- ROM, nonvolatile namun tidak bisa diubah datanya
- EEPROM (*Electrically Erasable Programmable Read Only Memory*) dan flash memory, nonvolatile dan bisa diubah datanya
- Flash memory memiliki keterbatasan banyaknya penghapusan data
- CMOS (Complementary Metal–Oxide–Semiconductor), volatile namun bertenaga baterai yang tahan lama

# DISK

- Kapasitas besar, relatif murah
- Namun lambat, karena mekanis,
- 5400 – 10800 rpm
- Suatu jalur melingkar pada tiap surface disebut *track* yang terdiri dari sejumlah *sector*
- Gabungan track dengan radius sama membentuk *cylinder*
- Head harus digeser ke cylinder yang tepat, surface diputar sampai ke sector yang tepat, lalu akses data 50 – 160 MB/s
- SO berfungsi memetakan alamat logical menjadi akses fisik, diperumit dengan *virtual memory*



# Perangkat I/O

- Perangkat dan controller
- Tugas controller untuk memetakan alamat logical menjadi alamat fisik
- Perangkat I/O pun punya interface agar standar
- *Device driver* adalah s/w yang berkomunikasi dengan controller
- Device driver perlu berjalan di mode kernel dengan cara
  - Relink driver kepada SO lalu *reboot*
  - Memberitahu adanya driver kepada SO, *reboot* dan SO mencari driver yg dibutuhkan
  - On-the-fly



## Perangkat I/O (2)

- Controller dilengkapi register
- Register bisa dipetakan ke address space dari SO, sehingga bisa diakses oleh program tanpa instruksi khusus
- Masalah I/O: lebih lambat dari processor
  - Programmed I/O, processor harus terus-menerus memeriksa I/O module dengan busy waiting
  - I/O module memberitahu driver via interrupt bila proses I/O sudah selesai
  - Menggunakan h/w khusus: DMA (Direct Memory Access) yang bekerja paralel tanpa processor harus terlibat

# Bus

- Melayani transfer data antar perangkat
- ISA (Industrial Standard Architecture), teknologi lama (8 – 16 bit)
- PCI (Peripheral Component Interconnect), lebih cepat (32 - 64bit)
- IDE bus, peripheral standar spt. Disk dan CD ROM
- USB bus
- SCSI bus untuk perangkat yang cepat
- Conflict terjadi jika interrupt yang sama dipakai 2 perangkat berbeda

# Booting

- Memanggil BIOS (Basic Input Output System)
- Memeriksa kapasitas RAM
- Memeriksa perangkat I/O yang terhubung ke bus (ISA dan PCI)
- Memeriksa boot device sesuai urutan yang tersimpan di BIOS
- Membaca boot sector, yaitu tempat SO disimpan, dan meload ke memory (oleh bootstrap loader)
- SO akan memeriksa device driver
- Diakhiri dengan login