

Organisasi dan Arsitektur Komputer

## **Pertemuan 2: Pengenalan Memori**

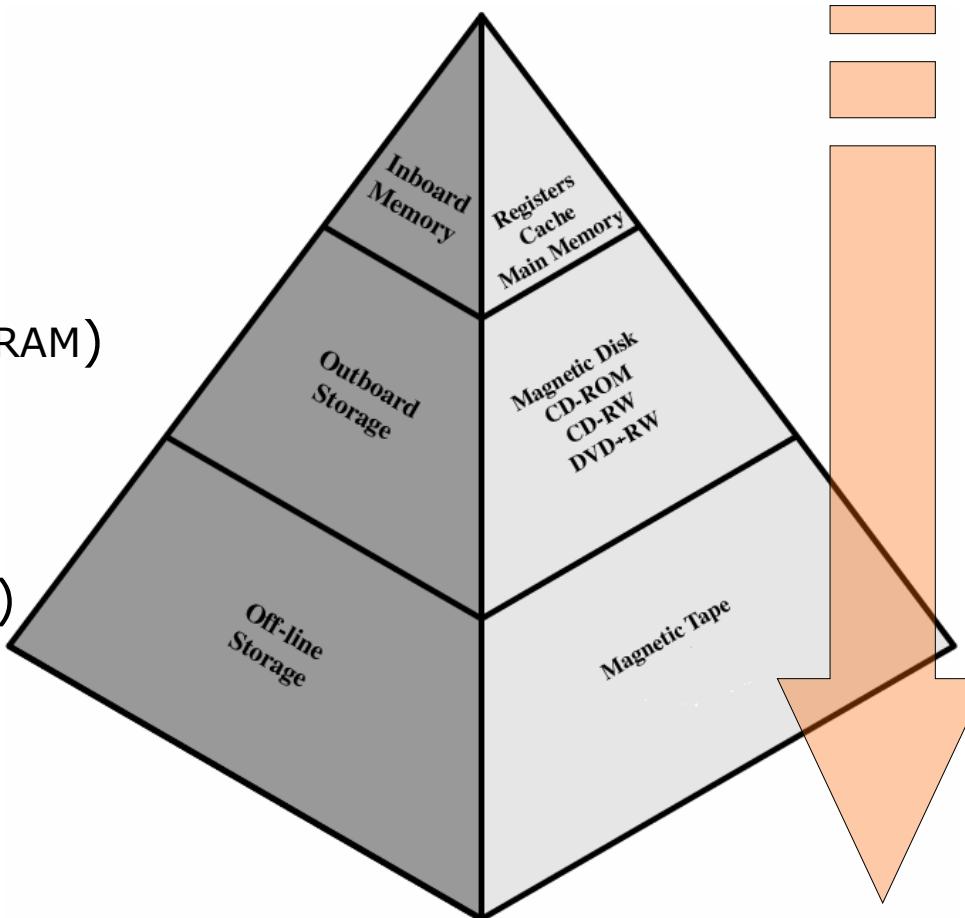
# Memory

1. Hierarki Memory
2. Karakteristik Memory
3. Organisasi Memory
  - a. Sel Memory
4. Memory Utama
  - a. RAM
    - i. SRAM
    - ii. DRAM
    - iii. Advanced DRAM
  - b. ROM
5. Error Checking
  - a. Parity
  - b. Hamming Code
6. Memory Sekunder
  - a. RAID
  - b. Memory Optic

# Hierarki dan Karaktieristik Memori

# Hirarki Memori

Registers  
L1 Cache  
L2 Cache  
Main memory (RAM)  
Disk cache  
Disk (Harddisk)  
Optical (CD, DVD)  
Magnetic tape



- **Biaya** per bit makin murah
- **Kapasitas** makin besar
- **Waktu akses** makin lama
- **Frekuensi** diakses oleh prosesor makin jarang

# Karakteristik Memori (1)

---

*Memory* diklasifikasikan berdasarkan:

- (1) Lokasi
- (2) Kapasitas
- (3) Satuan transfer
- (4) Cara akses
- (5) Performansi
- (6) Jenis fisik
- (7) Karakteristik fisik
- (8) Organisasi memori

# Karakteristik Memori (2)

---

## (1) Lokasi:

- ❖ *Internal* : dapat diakses oleh prosesor tanpa melalui I/O
  - Register
  - Cache memory
  - Main memory (RAM)
- ❖ *External* : untuk mengaksesnya harus melalui I/O
  - Harddisk, Diskette, Magnetic Tape
  - Flashdisk
  - CDROM, dll

## (2) Kapasitas:

- ❖ ***Adalah kemampuan menampung data dalam satuan tertentu (byte atau word)***
- ❖ Satu byte = 8 bit
- ❖ Satu word = 8, 16, atau 32 bit (tergantung pada pembuat prosesor, Intel: satu word = 16 bit, IBM 370: 32 bit)

# Karakteristik Memori (3)

---

## (3) Satuan transfer:

- ❖ Memori internal:
  - ***Adalah banyaknya bit yang dapat dibaca/ditulis dari/ke memori dalam setiap detik***
  - Adalah **setara** dengan banyaknya jalur data yang terhubung ke memori (lebar bus)
  - Biasanya sebanyak satu word, tetapi dapat lebih banyak lagi (misal: 32, 64, atau 128 bit)
- ❖ Memori eksternal
  - Digunakan satuan **block** yang ukurannya lebih dari satu word
- ❖ Satuan alamat: (*addressable unit*)
  - ***Adalah ukuran memori terkecil yang dapat diberi alamat tersendiri***
  - Besarnya tergantung pembuat prosesor (Intel: 1 byte atau 8 bit),

# Karakteristik Memori (4)

---

## (4) Cara akses:

### ❖ *Sequential access*

- Akses ke memori dilakukan secara **berurutan**
- Digunakan mekanisme *shared read/write*
- Waktu akses sangat variabel, bergantung pada lokasi data yang akan dituju dan data sebelumnya
- Contoh: Magnetic tape

### ❖ *Direct access*

- Akses ke memori langsung menuju ke **lokasi terdekat**, diteruskan dengan sedikit pencarian dan perhitungan
- Setiap blok/record mempunyai alamat unik berdasarkan lokasi fisik
- Digunakan mekanisme *shared read/write*
- Waktu aksesnya variabel (berbeda-beda) dan bergantung pada lokasi data yang akan dituju dan lokasi data sebelumnya
- Contoh: harddisk

# Karakteristik Memori (5)

---

## ❖ *Random access*

- Akses ke memori dilakukan secara *random langsung* ke alamat yang dituju
- Setiap alamat memori mempunyai alamat unik
- Waktu aksesnya konstan dan tidak bergantung pada urutan akses sebelumnya
- Contoh: *main memory*

## ❖ *Associative*

- Pencarian data di memori dilakukan dengan **membandingkan** seluruh word secara **bersamaan**, tidak berdasarkan alamat
- Waktu akses konstan dan tidak bergantung pada lokasi dan urutan akses sebelumnya
- Contoh: *cache memory*

# Karakteristik Memori (6)

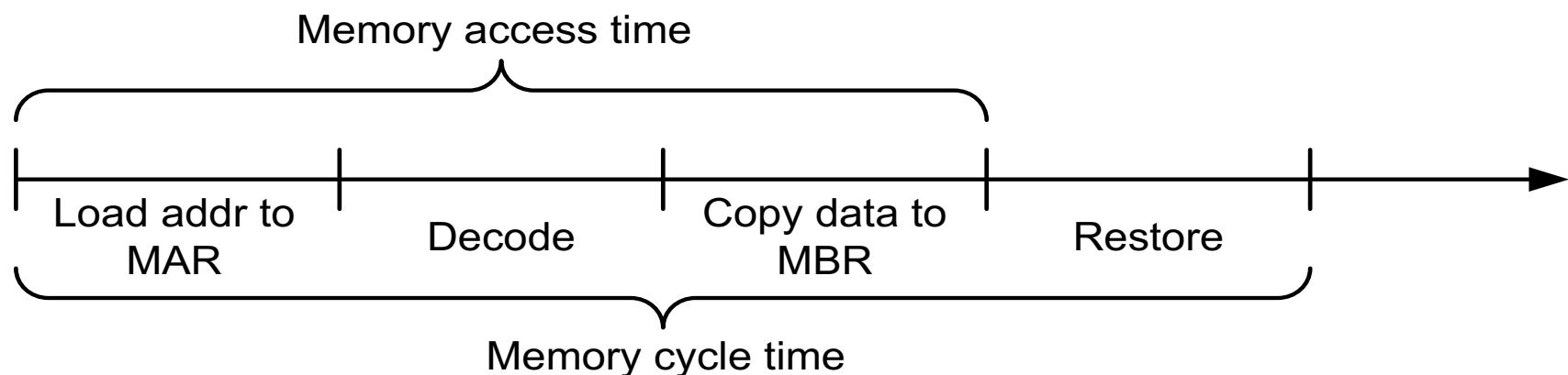
## (5) Performansi:

### ❖ Waktu akses (*latency*)

- Waktu antara perintah akses (baca atau tulis) sampai didapatkannya data di MBR atau data dari MBR telah disalin ke lokasi memori tertentu

### ❖ Waktu siklus memori

- Waktu dimulainya suatu operasi memori sampai memori siap melaksanakan operasi berikutnya (lebih penting)
- Waktu akses + waktu untuk perubahan signal jalur data sebelum akses kedua



# Karakteristik Memori (7)

---

## ❖ *Transfer rate*

- **Adalah waktu rata-rata perpindahan data**
- RAM: 1/waktu siklus
- Non-RAM:

$$T_N = T_A + N/R$$

$T_N$  = Waktu rata-rata untuk baca/tulis sejumlah  
N bit

$T_A$  = Rata-rata waktu akses

N = jumlah bit

R = transfer rate (bit per second)

# Karakteristik Memori (8)

---

## (6) Jenis fisik:

- ❖ Semikonduktor: RAM, flashdisk
- ❖ Magnetik: harddisk, *magnetic tape*
- ❖ Optik: CD, DVD

## (7) Karakteristik fisik:

- ❖ *Volatile*: nilainya hilang bila tegangan listrik tidak ada
- ❖ *Non-volatile*: nilainya TIDAK hilang (tetap ada) meskipun TIDAK ada tegangan listrik
- ❖ *Erasable*: nilainya dapat dihapus (semikonduktor, magnetik)
- ❖ *Non-erasable*: nilainya tidak dapat dihapus (ROM)

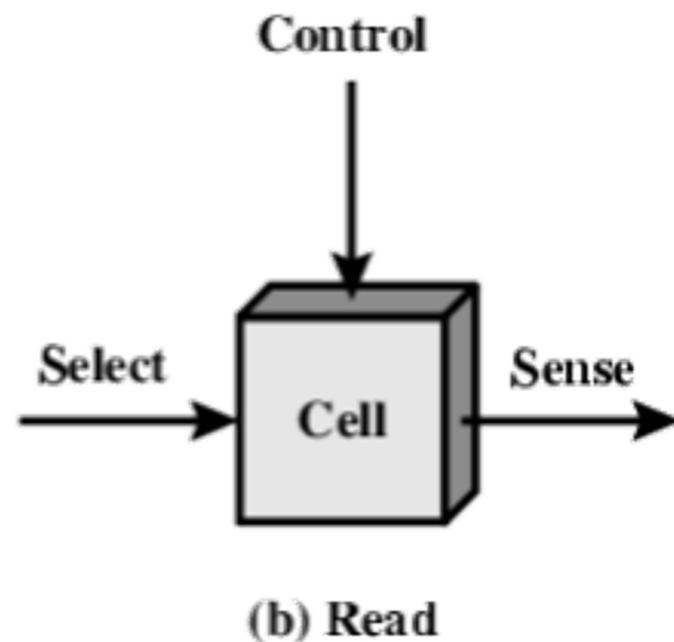
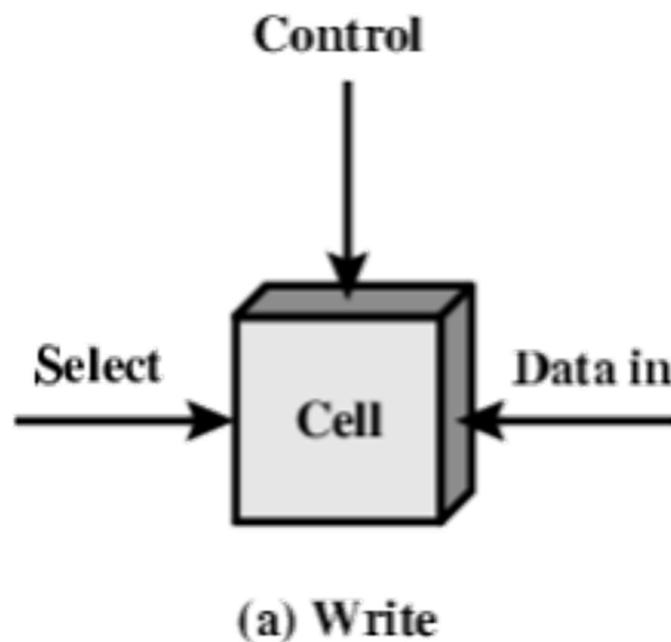
## (8) Organisasi memori:

- ❖ Penyusunan bit untuk membentuk word

# **Organisasi Memori**

# Sel Memori (1)

- Setiap memori tersusun dari rangkaian sel-sel memori:



# **Sel Memori (2)**

---

- **Sifat-sifat sel memori:**
  - ❖ Dapat memberikan 2 kondisi (1 atau 0)
  - ❖ Dapat ditulisi (minimal satu kali) dengan cara mengubah kondisi sel memori
  - ❖ Dapat dibaca
- **Tiap sel terdiri dari 3 terminal:**
  - ❖ **Select:** untuk **memilih** sel memori yang akan dibaca/ditulisi
  - ❖ **Control:** untuk menentukan jenis operasi **write/read**
  - ❖ **Data:**
    - Write: untuk mengubah kondisi sel dari 1 ke 0 atau sebaliknya
    - Read: untuk membaca kondisi sel memori

# **Sel Memori (3)**

---

- Apa **syarat device** yang dapat digunakan untuk menyimpan data biner ?
  - ❖ Hanya dapat menyimpan 2 macam nilai/kondisi (*true-false* atau 1-0) yang stabil
  - ❖ Mempunyai pembatas yang lebar yang dapat memisahkan kedua nilai/kondisi secara tegas
  - ❖ Mampu menangani perubahan nilai/kondisi dalam waktu yang tidak terbatas
  - ❖ Nilai/kondisinya tidak rusak pada saat dibaca
- Implementasinya dengan apa ?
  - ❖ Semikonduktor:
    - Gabungan antara transistor dan kapasitor
    - Gabungan beberapa transistor

# Sel Memori (4)

- ❖ Gabungan antara transistor dan kapasitor (**memori dinamis**)

- Capasitor***

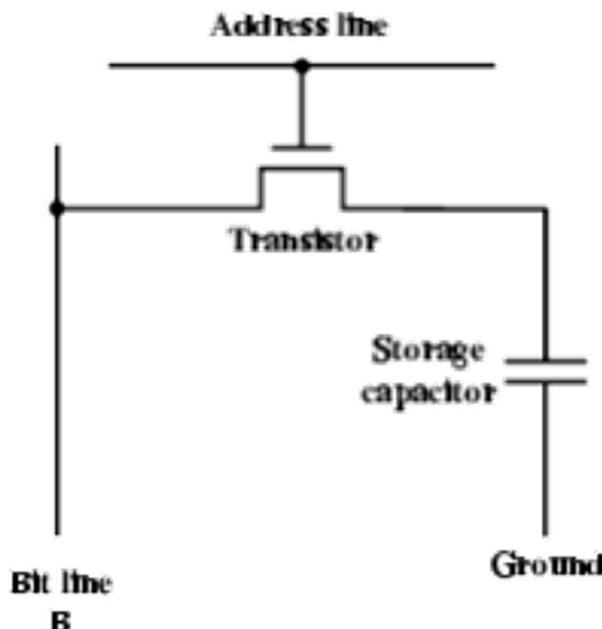
- + Untuk menyimpan nilai data

- Address line:***

- + Untuk mengaktifkan sel memori (transistor)

- Bit line:***

- + Untuk membaca/menulis data dari/ke sel memori

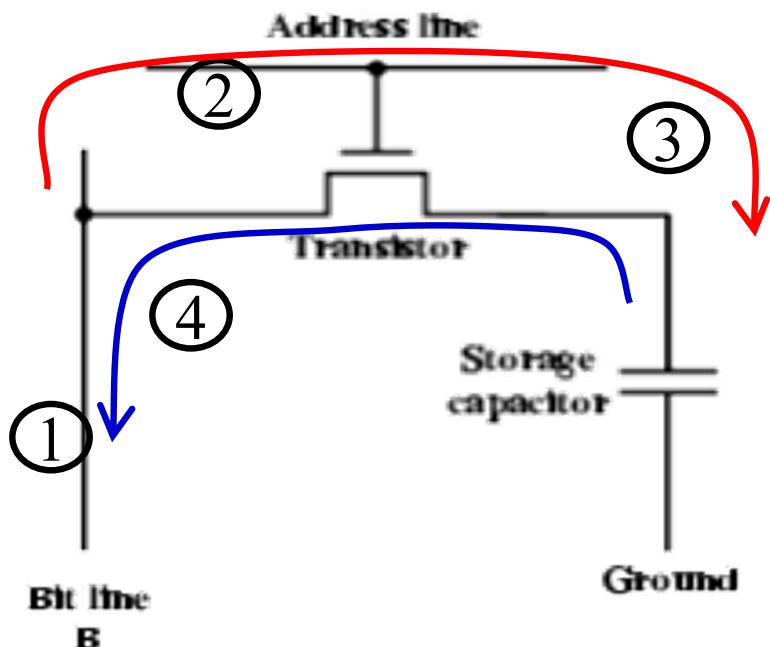


# Sel Memori (5)

- ❖ Gabungan antara transistor dan kapasitor (memori dinamis) (cont'd)

## ■ Cara kerja

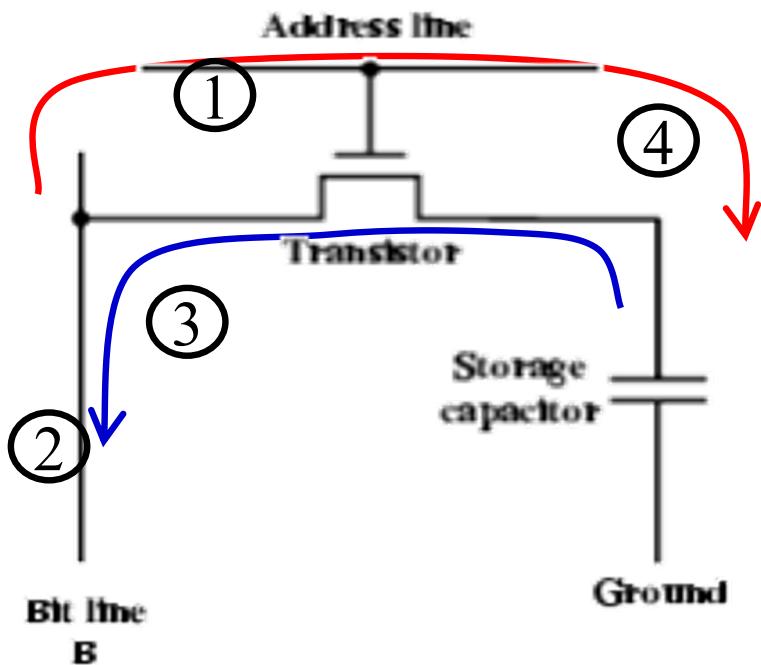
### + Simpan data (*write*):



1. Signal yang akan ditulis dihubungkan ke *bit line* (high voltage = 1, low voltage = 0)
2. Signal diberikan ke *address line* → transistor on
3. Untuk simpan bit 1 → arus mengalir masuk **ke** kapasitor → kapasitor diisi muatan
4. Untuk simpan bit 0 → arus mengalir **dari** ke kapasitor → muatan kapasitor dikosongkan

# Sel Memori (6)

## ❖ Cara kerja: (cont'd)

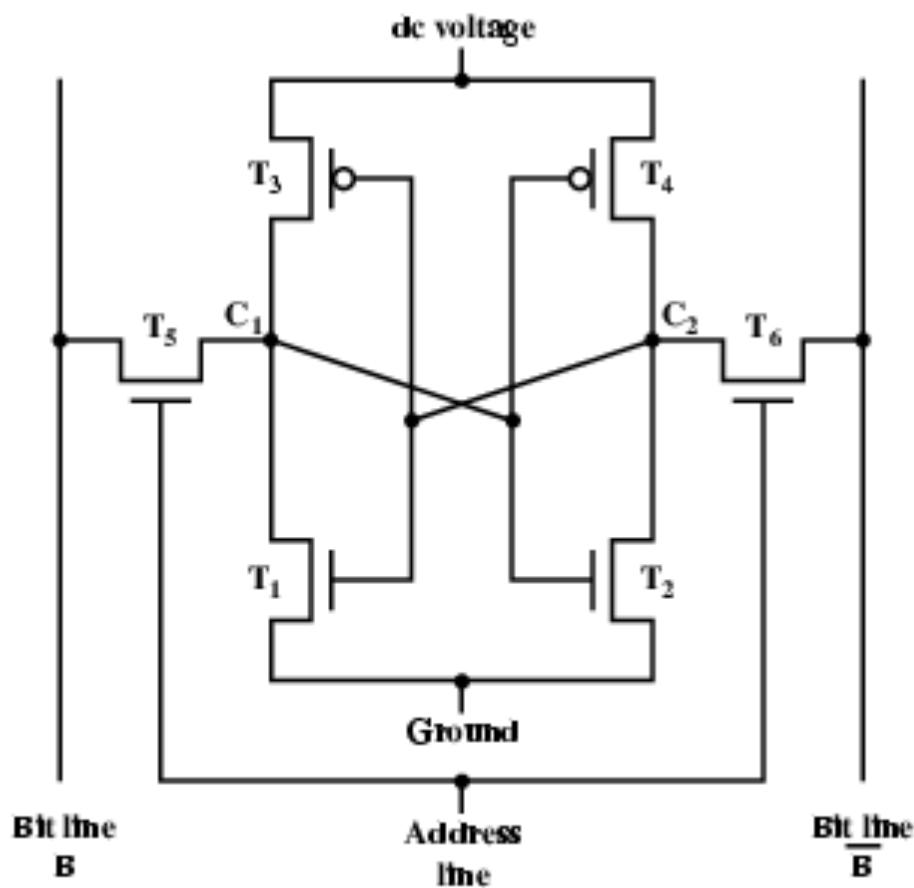


## + Baca data (*read*):

1. *Address line* diberi signal → transistor on
2. Muatan kapasitor dialirkan ke *bit line/sense amplifier*
3. *Sense amplifier* membandingkan muatan kapasitor dengan tegangan reference untuk menentukan apakah data bernilai 1 atau 0
4. Muatan kapasitor berkurang → perlu di-*refresh* → disebut **memori dinamis**

# Sel Memori (7)

- ❖ Gabungan antara beberapa transistor (**memori statis**)



- Status T<sub>1</sub> selalu berlawanan dengan T<sub>3</sub>, tetapi selalu sama dengan T<sub>4</sub>
- Status T<sub>2</sub> selalu berlawanan dengan T<sub>4</sub>, tetapi selalu sama dengan T<sub>3</sub>
- Nilai bit line  $\overline{B}$  berlawanan dengan nilai bit line B
- Signal **high**:  
T<sub>1</sub>=off, T<sub>3</sub>=on  $\rightarrow$  titik C<sub>1</sub>=high  
 $\rightarrow$  bit line B = **HIGH**  
T<sub>2</sub>=on, T<sub>4</sub>=off  $\rightarrow$  titik C<sub>2</sub> = LOW
- Signal **low**:  
T<sub>1</sub>=on, T<sub>3</sub>=off  $\rightarrow$  titik C<sub>1</sub>=low  
 $\rightarrow$  bit line B = **LOW**  
T<sub>2</sub>=off, T<sub>4</sub>=on  $\rightarrow$  titik C<sub>2</sub> = HIGH

# **Memory Utama:**

## **RAM dan ROM**

# **Memory Utama:**

# **RAM**

# **RAM (Random Access Memory) (1)**

## ➤ Dynamic RAM (DRAM)

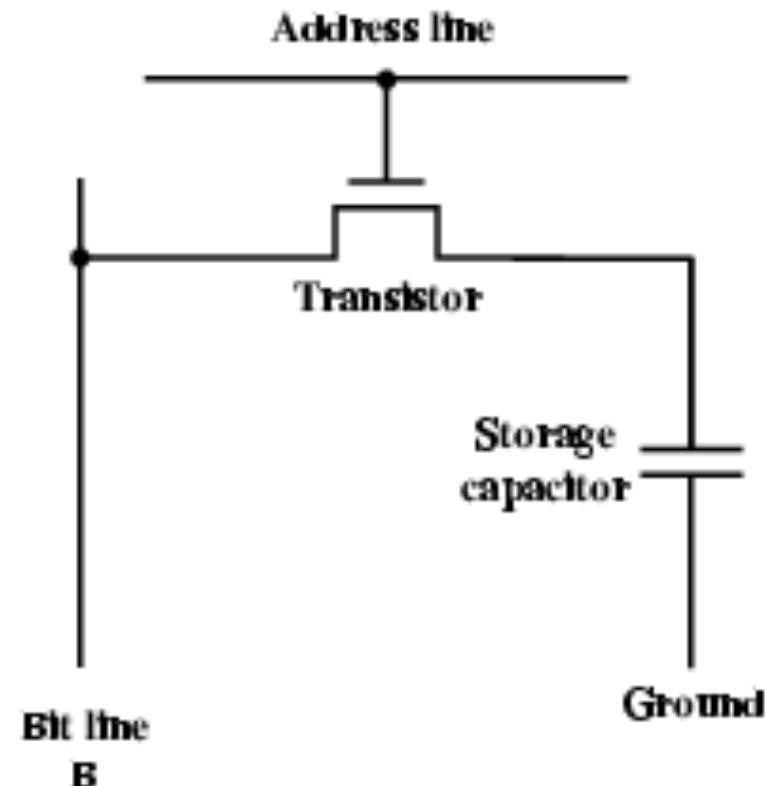
### ❖ Sel DRAM:

- Data berupa muatan listrik yang disimpan di kapasitor

### ❖ Mengapa disebut dynamic RAM ?

- Muatan listrik yang disimpan di kapasitor cenderung mengalami kebocoran, sehingga harus selalu di-*refresh*

### ❖ DRAM digunakan untuk *main memory*



# RAM (Random Access Memory) (2)

## ➤ Static RAM (SRAM)

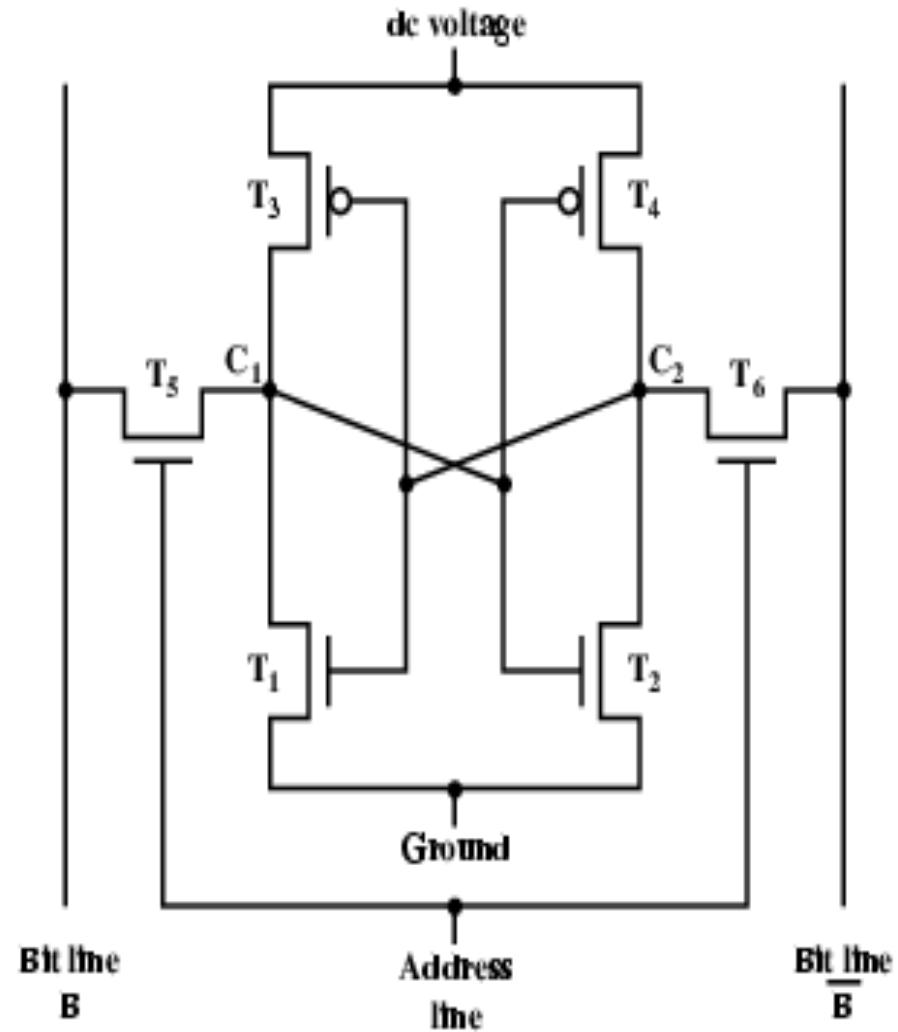
### ❖ Sel SRAM:

- Disusun dari beberapa transistor (*flip-flop*)

### ❖ Mengapa disebut static RAM ?

- Selama masih ada listrik, maka data yang disimpan tidak hilang, sehingga tidak perlu di-*refresh*

### ❖ SRAM digunakan untuk *cache memory*



# DRAM vs SRAM

---

## DRAM

- (+) sederhana
  - (+) dimensi lebih kecil
  - (+) murah
  - (+) kapasitas besar
  - (-) perlu rangkaian *refresh*
  - (-) biaya rangkaian *refresh*
    - memori berukuran besar
    - lebih mahal daripada biaya memori itu sendiri
  - (-) lebih lambat
- Sama-sama **volatile**

## SRAM

- (-) lebih kompleks
- (-) dimensi lebih besar
- (-) lebih mahal
- (-) kapasitas kecil
- (+) tidak perlu di-*refresh*
- (+) tidak perlu rangkaian *refresh*

# **Advanced DRAM (1)**

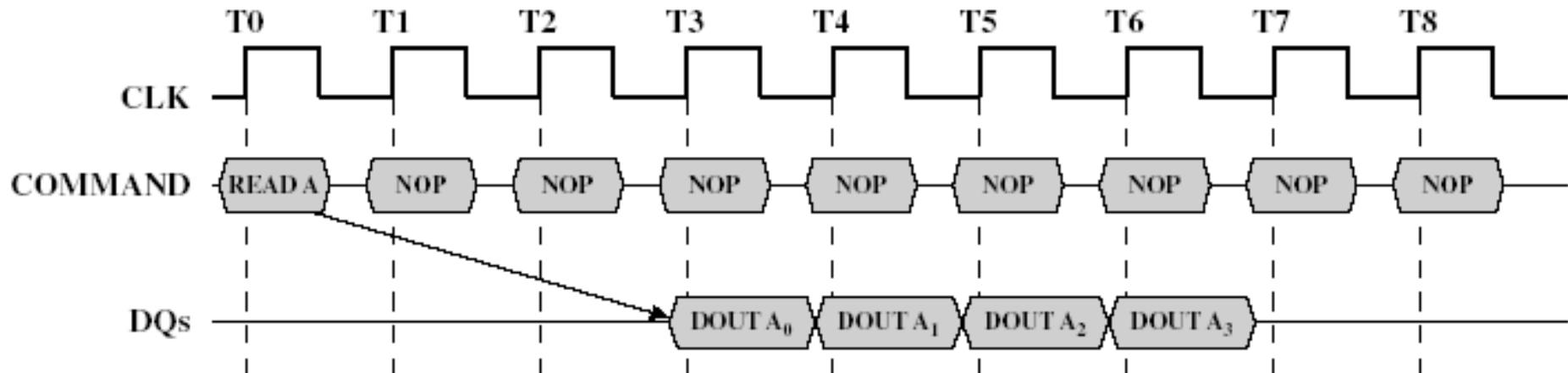
---

## ➤ *Synchronous DRAM (SDRAM)*

- ❖ Pertukaran data didasarkan pada *signal clock* eksternal **tanpa wait state**
- ❖ Kecepatan **sesuai** dengan kecepatan prosesor atau bus memori
- ❖ Selama proses pencarian data, CPU dapat melakukan tugas lain (**tidak perlu menunggu**, karena CPU tahu kapan data sudah tersedia)

# Advanced DRAM (2)

- ❖ Contoh: *SDRAM read timing*
  - $Burst length = 4 = T3-T6$ ,  $CAS latency = 2 = T1-T2$

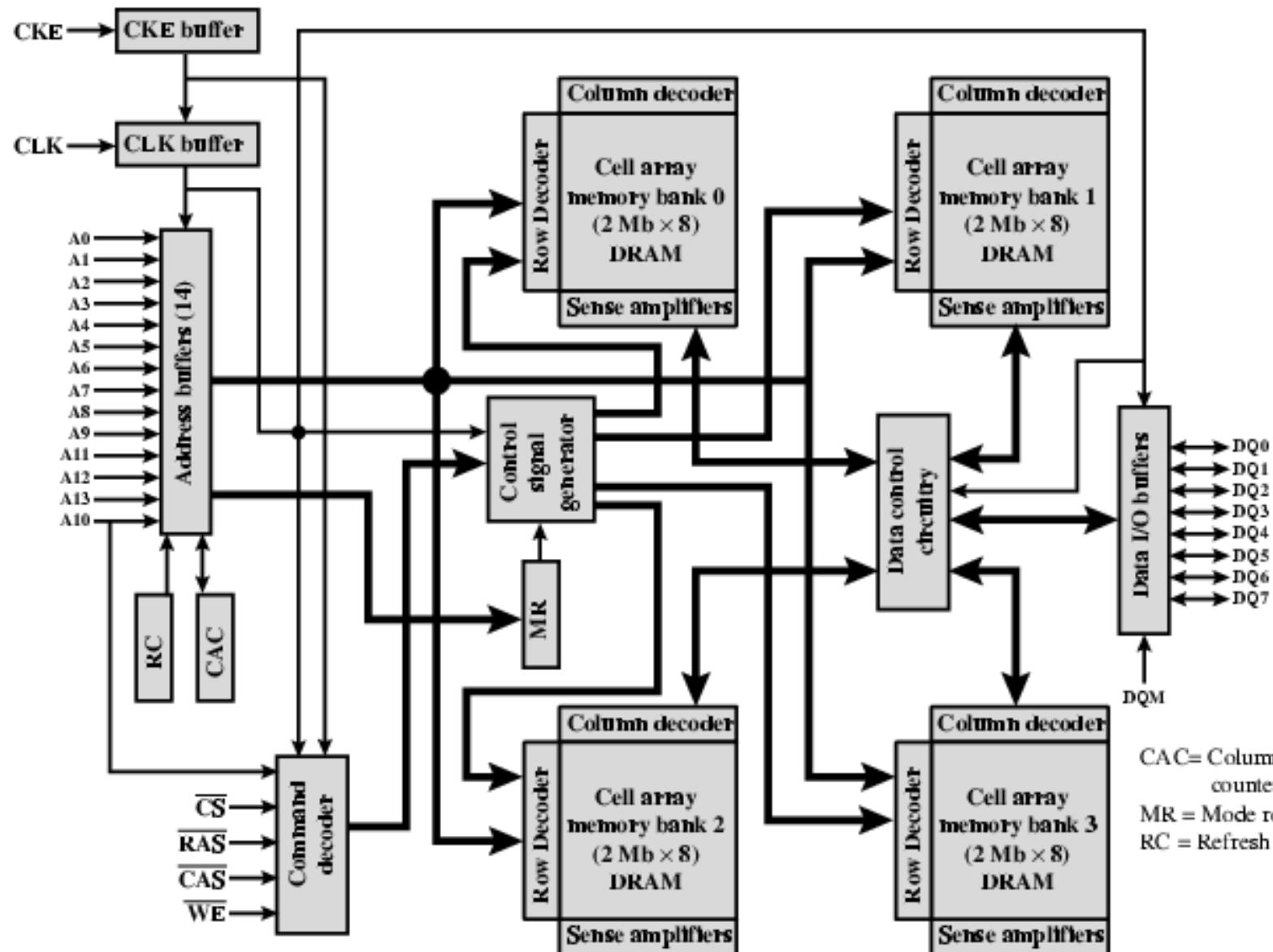


- ❖ *Burst length* = banyaknya *clock* untuk mengirimkan data (1, 2, 4, 8, full page)
- ❖ *Latency* = banyaknya *clock* yang diperlukan untuk persiapan sebelum data dikirimkan

- *Double Data Rate – SDRAM (DDR-SDRAM)*
  - ❖ Sama seperti SDRAM, tetapi dapat mengirimkan data 2x dalam satu *clock*

# Advanced DRAM (3)

Contoh  
IBM  
64Mb  
SDRAM



# Advanced DRAM (4)

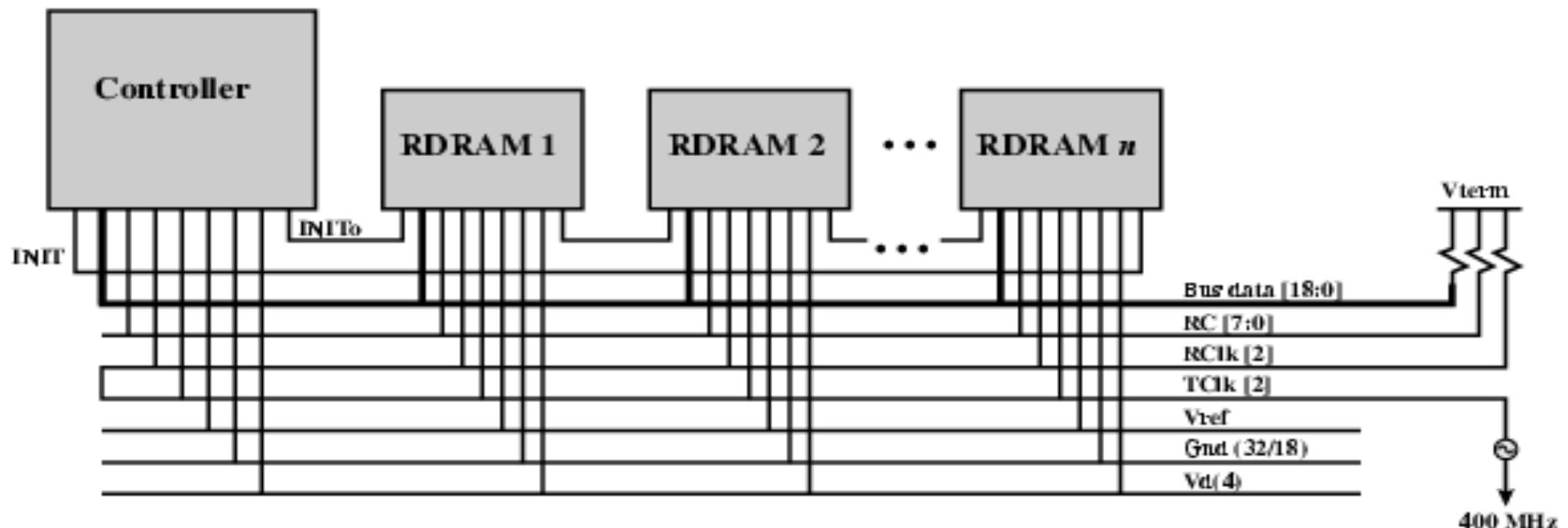
---

## ➤ *Rambus DRAM (RDRAM)*

- ❖ Diadopsi oleh Intel untuk Pentium dan Itanium
- ❖ Pesaing model SDRAM
- ❖ Bus alamat support s.d. 320 chip RDRAM
- ❖ Transfer data s.d. 1,6 GBps (2 x 800 MBps)
  - *Impedance, clock, dan signal* didefinisikan secara tepat
- ❖ Jalur data sebanyak 18 bit (16 data, 2 parity)
- ❖ Jalur RC (8 bit) digunakan untuk alamat dan signal control
- ❖ *Memory request* tidak menggunakan *RAS, CAS, R/W atau CE*, tetapi melalui *high speed bus* yang memuat informasi:
  - alamat yang diinginkan
  - jenis operasi
  - jumlah byte dalam operasi
- ❖ Pengiriman data secara *synchronous*

# Advanced DRAM (5)

## ➤ Struktur RDRAM



# **Memory Utama:**

## **ROM**

# **“Read” Only Memory (ROM)**

---

- *ROM (Read Only Memory)*
  - *PROM (Programmable ROM)*
  - *EPROM (Erasable PROM)*
  - *EEPROM (Electrically EPROM)*
  - *Flash Memory*

# **Read Only Memory (ROM)**

---

- Data bersifat permanen (*non-volatile*)
- Nilai data dihasilkan dari kombinasi rangkaian logika di dalamnya
- Penulisan data dilakukan pada saat pembuatan chip
- Data tidak dapat dihapus
- Aplikasi:
  - ❖ *Store firmware*
  - ❖ *Booting (BIOS)*

# **Programmable ROM (PROM)**

---

- Data ditulis **sesudah** chip dibuat
- Digunakan bila jumlahnya tidak terlalu besar (*bukan mass product*) dan memori memuat data khusus
- ***Non-volatile***
- Hanya dapat ditulisi **satu kali**
- Penulisan data dilakukan secara elektrik dengan PROM programmer
  - Awalnya bernilai 1 semua
  - Sel memori “dibakar” (burning-in) dengan arus listrik yang cukup besar, lokasi bit akan terbakar dan menunjukkan sebuah nilai (0)
- **Data tidak dapat dihapus**
- Lebih fleksibel daripada ROM
- Contoh aplikasi: RFID

# Erasable PROM (EPROM)

---

- Data ditulis **sesudah** chip dibuat
- *Non-volatile*
- Proses baca dan tulis secara elektrik
- Data dapat dihapus dan ditulisi **berulang-ulang** dengan radiasi ultraviolet
  - Sinar tersebut melewati celah di kumpulan chip
- Data dihapus dalam satuan **chip** (**semua data dihapus**)
- Lebih mahal daripada PROM

# **Electrically EPROM (EEPROM)**

---

- Data ditulis **sesudah** chip dibuat
- *Non-volatile*
- Data dapat ditulis dan dihapus **kapan saja** secara elektrik, data tidak perlu dihapus terlebih dahulu
- Data dihapus dalam satuan **byte**
- Lebih mahal daripada EPROM
- Kurang rapat dibanding EPROM

# Flash Memory

- *Non-volatile*
- Biaya dan fungsionalitas berada diantara EPROM dan EEPROM
- Data dihapus dalam satuan **blok memori**
- Lebih rapat dibanding EEPROM

# Jenis Memori Semikonduktor

---

Memory Type	Category	Erasure	Write Mechanism	Volatility	
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Electrically	Volatile	
Read-only memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile	
Programmable ROM (PROM)			Electrically		
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level			
Electrically Erasable PROM (EEPROM)		Electrically, byte-level			
Flash memory		Electrically, block-level			

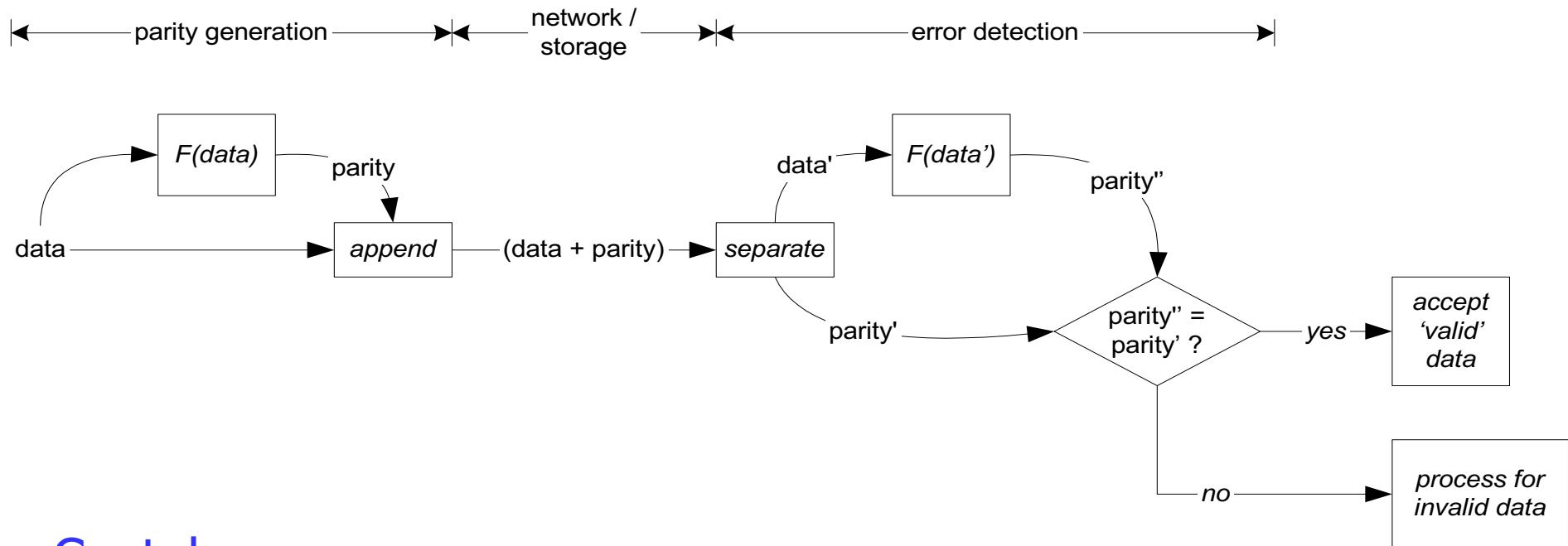
# **Error Check: Parity & Hamming Code**

# Penanganan Kesalahan Dengan **Paritas** (1)

---

- Mendeteksi kesalahan data pada **level bit**
- **Bit paritas:**
  - ❖ bit ekstra yang ditambahkan pada suatu unit data terkecil
  - ❖ digunakan dalam proses pengecekan kebenaran data ketika data akan disimpan atau dikirim
  - ❖ dihasilkan oleh fungsi generator paritas
- **Jenis paritas:**
  - ❖ **Paritas genap (even):**
    - Menambahkan sebuah bit sehingga total bit '1' suatu word berjumlah **genap**
  - ❖ **Paritas ganjil (odd):**
    - Menambahkan sebuah bit sehingga total bit '1' suatu word berjumlah **ganjil**
- Dapat mendeteksi **kesalahan bit berjumlah ganjil**, lokasi bit tidak bisa diketahui

## Penanganan Kesalahan Dengan Paritas (2)



Contoh:

Data = **1001001**, jenis paritas = paritas genap

Jika bit yang dibaca **10010011** →

data dianggap valid

Jika bit yang dibaca **10110011** →

data dianggap **tidak** valid

Jika bit yang dibaca **10110010** →

data dianggap valid !!

# Penanganan Kesalahan Dengan ***Hamming Code Single Bit***

---

- Biasa digunakan dalam konteks *Error Control* (deteksi maupun koreksi)
- *Codeword* = bit-bit data + bit paritas/kontrol
- *Hamming distance*: jumlah perbedaan bit dari dua buah *codeword*
  - ❖ *Hamming distance*  $n + 1 \rightarrow$  Dapat mendeteksi  $n$  error
  - ❖ *Hamming distance*  $2n + 1 \rightarrow$  Dapat me-recover  $n$  error
- Contoh *codeword* dengan 7 bit informasi dan 1 bit paritas genap:

0000000	0	}
0000001	1	
0000010	1	
0000011	0	

Antar *codeword*  
terdapat 2 bit berbeda

*Hamming distance-nya* = 2  $\rightarrow$  hanya dapat mendeteksi 1 bit error

# Penanganan Kesalahan Dengan ***Hamming Code Multi Bit*** (1)

---

- Blok data sebanyak **M** digit dikodekan menjadi **N** digit (**N>M**)
  - Ditulis dengan notasi (N,M)
    - Misal *codeword* terdiri dari 7 bit data + 4 bit *check* → (11,7)
  - Jumlah bit *codeword* harus memenuhi persamaan:  
$$2^K - 1 \geq M+K$$
K = jumlah bit kontrol; M = jumlah bit data
  - Posisi (letak) bit-bit K ditentukan dengan rumus  $2^x$ ; x = 0, 1, 2, 3, ...
  - Posisi bit dimulai dari posisi ke-1, **bukan ke-0 !!!**
  - $M/N$  = *code rate* atau *code efficiency*
  - $1 - M/N$  = *redundancy*
- Dapat mendekripsi kesalahan sebanyak 2 bit
- Dapat menentukan posisi bit yang error jika terjadi kesalahan sebanyak 1 bit
- *Recovery* dilakukan dengan meng-*inverse* bit pada posisi yang salah

# Penanganan Kesalahan Dengan ***Hamming Code Multi Bit*** (2)

---

- Contoh bit-bit data: 1001101 (7 bit)  
Jumlah bit *codeword*:  $2^K - 1 \geq M+K$   
Jika  $K = 3 \rightarrow 2^3 - 1 \geq 7 + 3$  (salah)  
Jika  $K = 4 \rightarrow 2^4 - 1 \geq 7 + 4$  (ok)

Posisi bit:	11	10	9	8	7	6	5	4	3	2	1
Bit-bit data:	1	0	0	K	1	1	0	K	1	K	K

- Cara menentukan bit-bit K:
  - ❖ Lakukan penjumlahan modulo 2 (biner) **semua nomor posisi bit data yang bernilai 1**

$$\begin{array}{r} 11 = 1011 \\ 7 = 0111 \\ 6 = 0110 \\ 3 = 0011 \\ \hline \end{array} + \quad \text{posisi ke: 8421}$$

- Hasil:

Posisi bit:	11	10	9	8	7	6	5	4	3	2	1
Bit-bit data:	1	0	0	1	1	1	0	0	1	0	1

# Penanganan Kesalahan Dengan ***Hamming Code Multi Bit*** (3)

## ➤ Pendeksiian kesalahan (*decoder*):

- ❖ Jumlahkan (modulo 2) semua posisi bit yang bernilai 1 (**termasuk bit check**)
  - Jika hasilnya 0 → tidak terjadi kesalahan
  - Jika hasilnya ≠ 0 → hasil penjumlahan merupakan posisi bit yang salah

$$\begin{array}{r} 11 = 1011 \\ 8 = 1000 \\ 7 = 0111 \\ 6 = 0110 \\ 3 = 0011 \\ 1 = 0001 \\ \hline \text{tidak terjadi error} \rightarrow 0000 + \end{array}$$

## ➤ Contoh jika terjadi *error* sebanyak **1 bit** (bit ke-11):

$$\begin{array}{r} 8 = 1000 \\ 7 = 0111 \\ 6 = 0110 \\ 3 = 0011 \\ 1 = 0001 \\ \hline 1011 + \end{array} \leftarrow \text{terjadi kesalahan pada bit ke-1011 (ke-11)}$$

## ➤ Recovery: invert bit ke-11

# **Penanganan Kesalahan Dengan**

## ***Hamming Code Multi Bit (4)***

- Contoh jika terjadi *error* sebanyak **2 bit** (bit ke-11 dan bit ke-1):

$$\begin{array}{r}
 8 = 1000 \\
 7 = 0111 \\
 6 = 0110 \\
 3 = 0011 \\
 \hline
 & 1010 \leftarrow
 \end{array}$$

kesalahan terdeteksi, posisi bit yang salah tidak diketahui

- Contoh jika terjadi *error* sebanyak **3 bit** (bit ke-11, bit ke-1, dan bit ke-10):

10	=	1010
8	=	1000
7	=	0111
6	=	0110
3	=	0011
<hr/>		
		0000

kesalahan tidak terdeteksi !

# **Memory Sekunder:**

# **RAID dan Memori Optik**

# **Memory Sekunder: RAID**

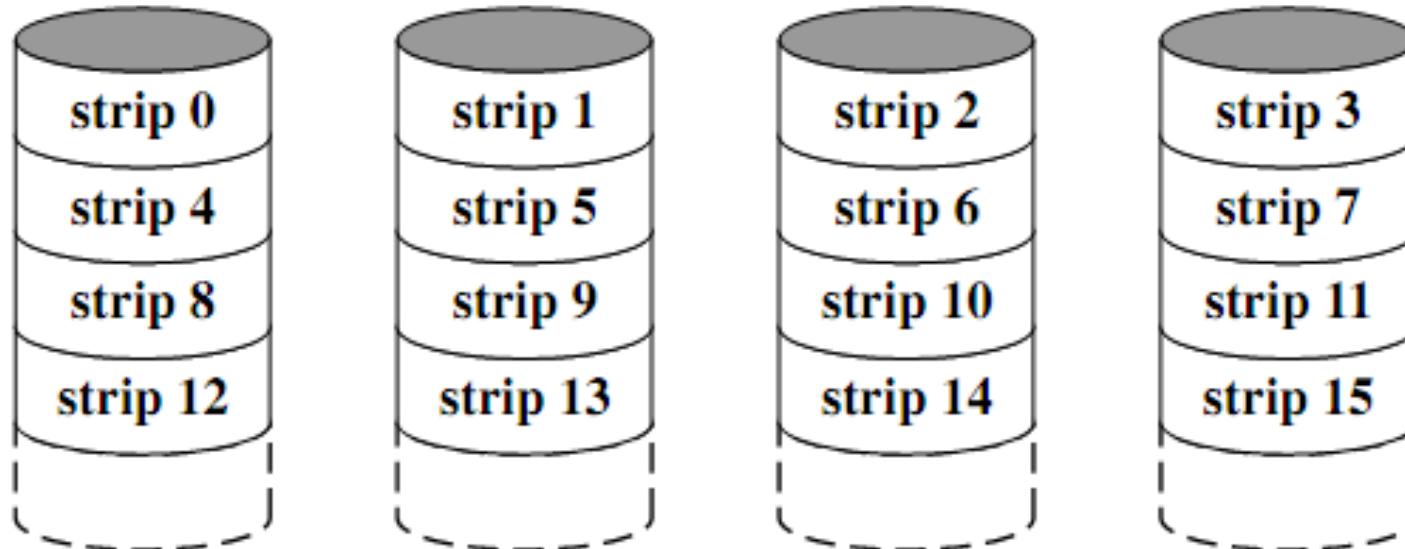
# **RAID (1)**

---

- Redundant Array of Independent Disks
- Redundant Array of Inexpensive Disks
- 7 cara konfigurasi (level 0 – level 6)
- Beberapa disk fisik yang dipandang sebagai satu drive logic oleh OS
- Data didistribusikan sepanjang drive fisik
- Dapat memanfaatkan kapasitas redundant untuk menyimpan mekanisme pengecekan error

# RAID 0 (2)

---

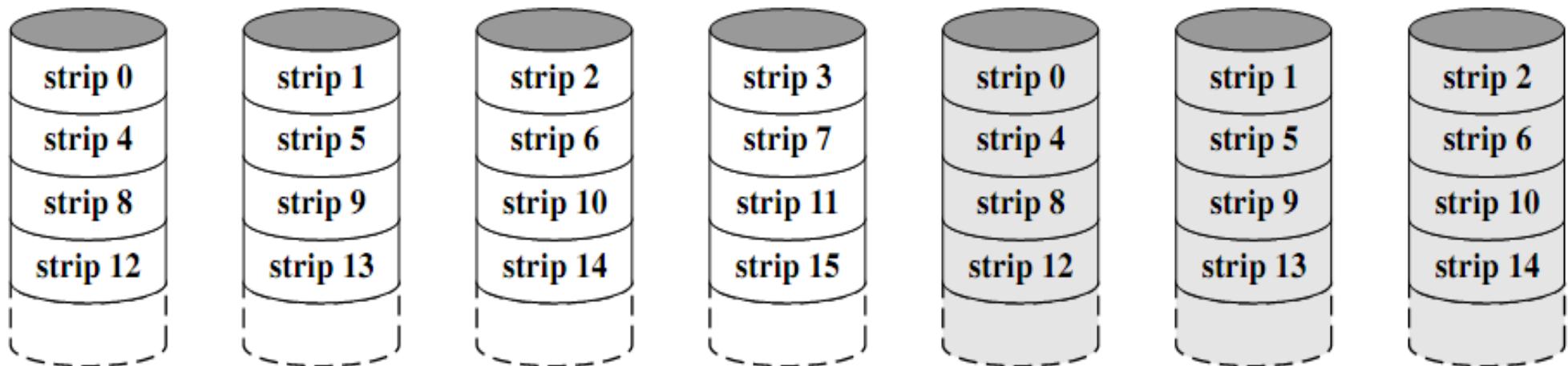


(a) RAID 0 (Nonredundant)

- Tidak ada redundancy
- RAID 0 menggunakan proses **striping**
  - ❖ Data disimpan ke dalam beberapa disk secara parallel untuk mempercepat proses baca/tulis
- Striping bersifat Round Robin

# RAID 1 (2)

---

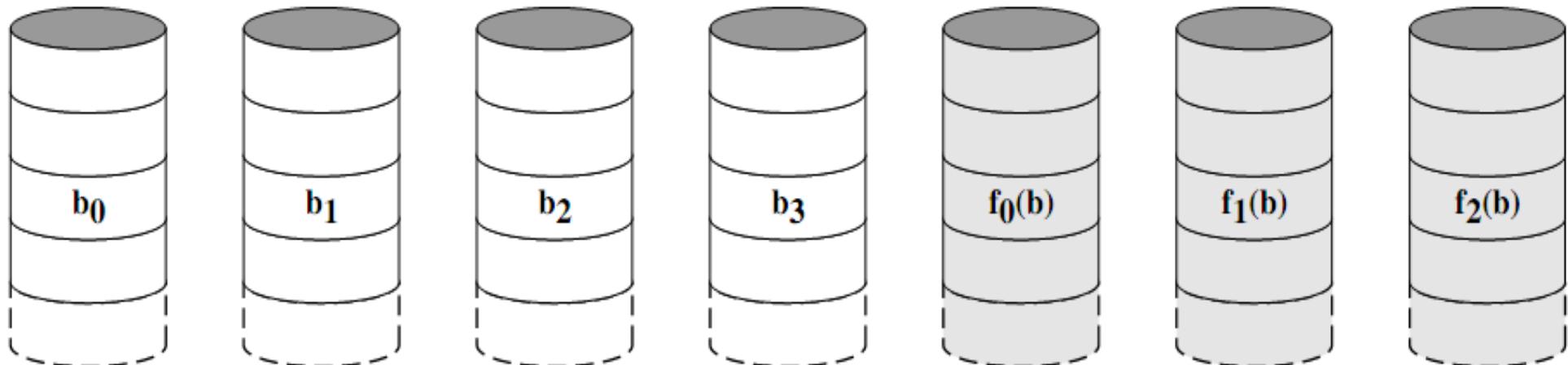


(b) RAID 1 (Mirrored)

- **Mirrored Disks**
  - ❖ Penyimpanan pada suatu disik akan diduplikat di disk lain
- **Data juga mengalami striping**
- **Setiap data mempunyai 2 copy**
  - ❖ Read dapat dilakukan pada salah satu copy
  - ❖ Write dilakukan di kedua copy
- **Recovery sederhana**
  - ❖ faulty disk ditukar & re-mirror
  - ❖ Tidak ada down time
- **(-) mahal**

## RAID 2 (2)

---

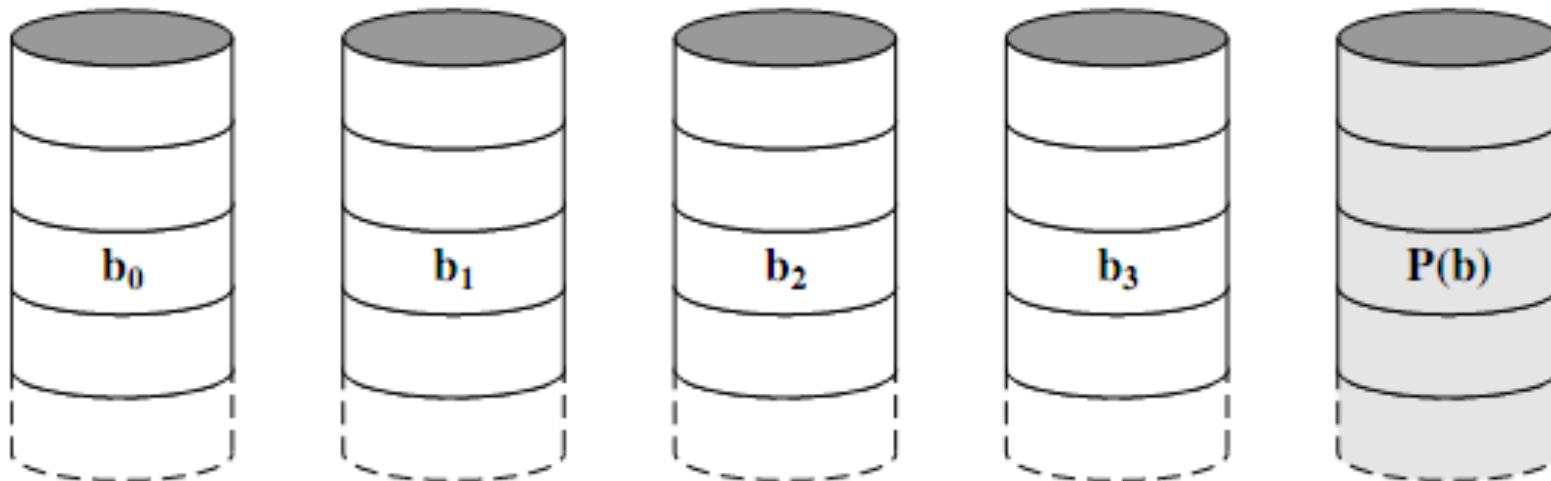


(c) RAID 2 (Redundancy through Hamming code)

- RAID 2 menggunakan **striping level byte**
- Error correction menggunakan **Hamming code**
  - ❖ Bit hamming code tersimpan dalam beberapa disk
- Redundancy banyak
  - ❖ Mahal
  - ❖ Tidak digunakan

# RAID 3 (2)

---

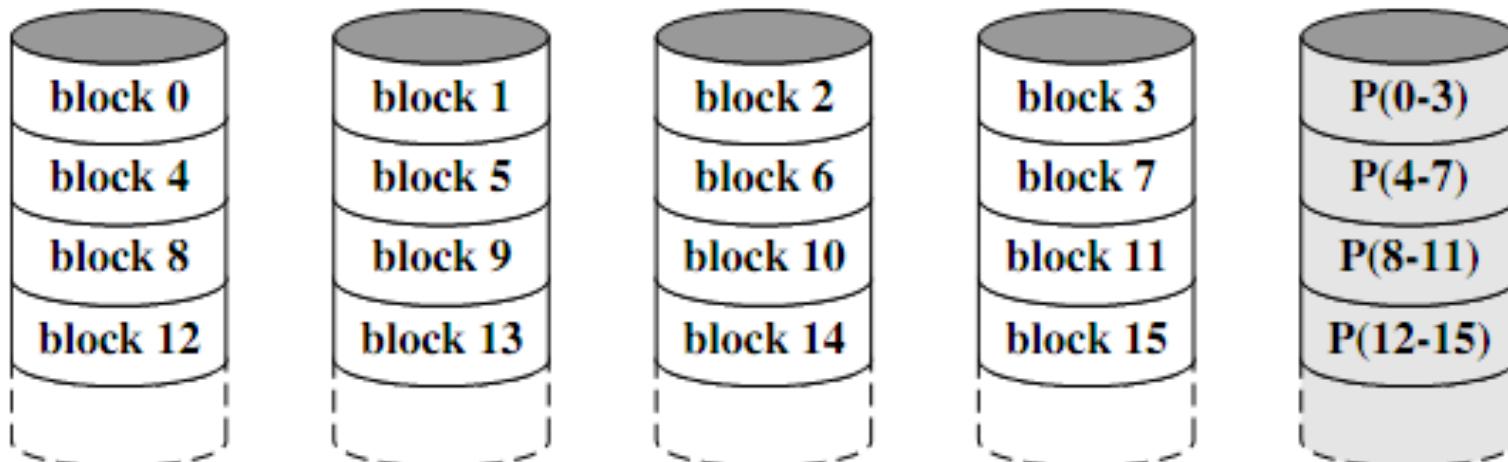


(d) RAID 3 (Bit-interleaved parity)

- Mirip RAID 2
- Hanya satu redundant disk, berapa pun besarnya array
- Pengecekan error menggunakan **parity** bit
- Data pada drive yang gagal bias direkonstruksi dari data yang selamat dan info parity
- Transfer rate sangat tinggi (karena **striping**)

# RAID 4 (2)

---

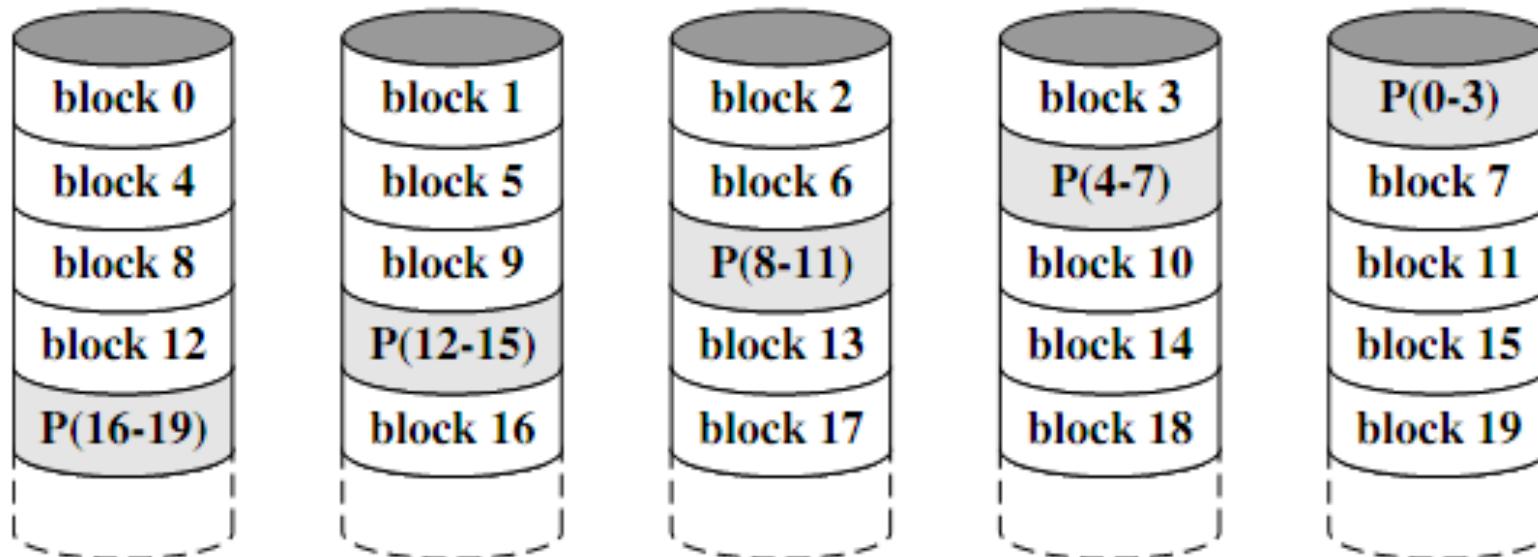


(e) RAID 4 (Block-level parity)

- Setiap disk beroperasi independent
  - ❖ Tidak paralel
- Bagus untuk I/O request rate tinggi
- Stripe berukuran besar (block)
- Bit-bit parity dikalkulasikan sepanjang stripe pada setiap disk
- Parity disimpan dalam parity disk
- (-) jika parity disk gagal, tidak ada mekanisme pengecekan error

# RAID 5 (2)

---

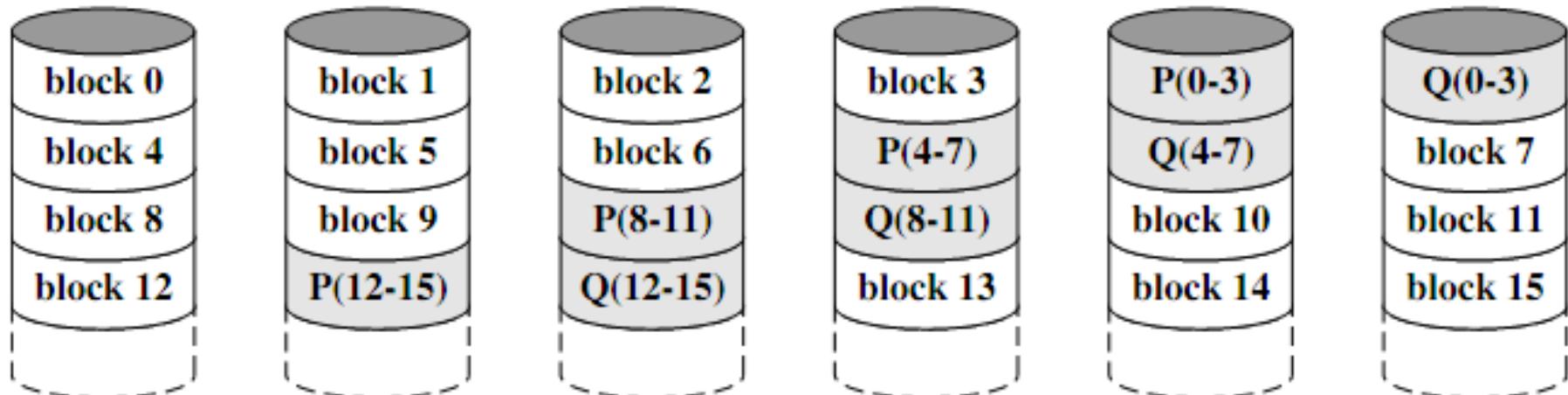


(f) RAID 5 (Block-level distributed parity)

- Seperti RAID 4
- Perbedaan: Parity di striped di semua disk
- Alokasi Round robin untuk semua parity stripe
- Mengatasi bottleneck parity disk pada RAID 4
- Digunakan secara umum dalam network server

# RAID 6 (2)

---



(g) RAID 6 (Dual redundancy)

- Dua parity calculation
- Disimpan pada block berbeda pada setiap disk
- User requirement untuk N data memerlukan N+2 kapasitas
- data availability tinggi
  - ❖ Tiga disk gagal untuk data loss
  - ❖ Write penalty signifikan

# Rangkuman RAID

---

Category	Level	Description	Disk Required	Data Availability	Large I/O Data Transfer Capacity	Small I/O Request Rate
Striping	0	Nonredundant	$N$	Lower than single disk	Very high	Very high for both read and write
Mirroring	1	Mirrored	$2N$	Higher than RAID 2, 3, 4, or 5; lower than RAID 6	Higher than single disk for read; similar to single disk for write	Up to twice that of a single disk for read; similar to single disk for write
Parallel access	2	Redundant via Hamming code	$N + m$	Much higher than single disk; comparable to RAID 3, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
	3	Bit-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
Independent access	4	Block-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 5	Similar to RAID 0 for read; significantly lower than single disk for write	Similar to RAID 0 for read; significantly lower than single disk for write
	5	Block-interleaved distributed parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 4	Similar to RAID 0 for read; lower than single disk for write	Similar to RAID 0 for read; generally lower than single disk for write
	6	Block-interleaved dual distributed parity	$N + 2$	Highest of all listed alternatives	Similar to RAID 0 for read; lower than RAID 5 for write	Similar to RAID 0 for read; significantly lower than RAID 5 for write

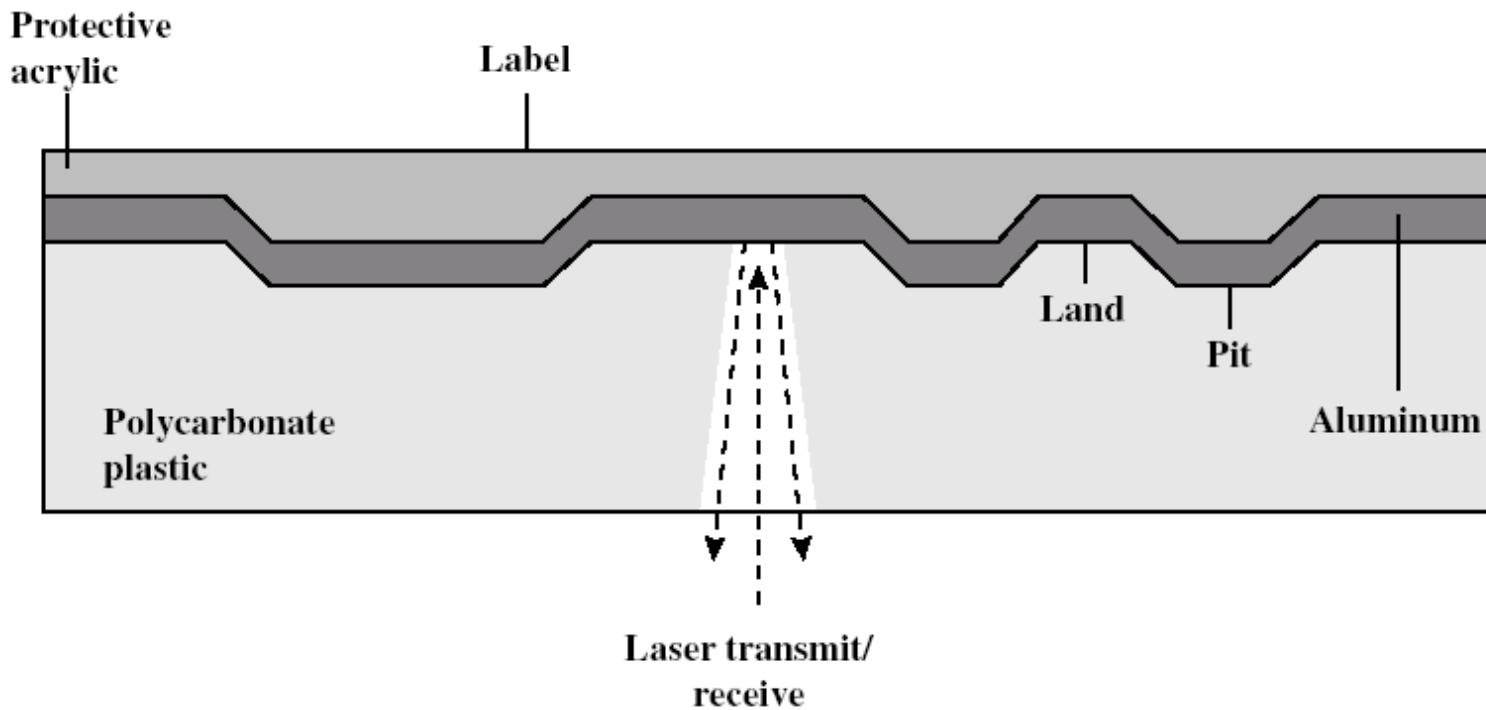
$N$  = number of data disks;  $m$  proportional to  $\log N$

# **Memory Sekunder:**

## **Memori Optik**

# CD ROM (1)

---



- ❖ Data disimpan dalam bentuk pits
- ❖ Cara baca dengan memantulkan sinar laser
- ❖ Kerapatan data konstan
- ❖ Kecepatan linear konstan

# CD ROM (3)

---

- ❖ Kelebihan CD ROM dibanding CD:
  - Lebih *rugged*
  - Mempunyai *error-correction*
- ❖ Permukaan CD ROM terdiri dari satu *track* berbentuk spiral
- ❖ Awal track berada di dekat pusat
- ❖ Track terbagi menjadi sektor yang ukurannya sama
- ❖ Data rate tetap → putaran disk di dekat pusat lebih cepat
- ❖ CD audio:
  - Kecepatan linear konstan
  - Single speed = 1.2 m/s
  - Panjang track (spiral) = **5.27 km**
  - Lama waktu = 4391 seconds = 73.2 menit
- ❖ Kecepatan yang lain merupakan kelipatannya (misal 8x, 24x, 52x, dll), referensi diambil dari kecepatan *floppy disc* = 150 kBps

# **CD R/RW**

---

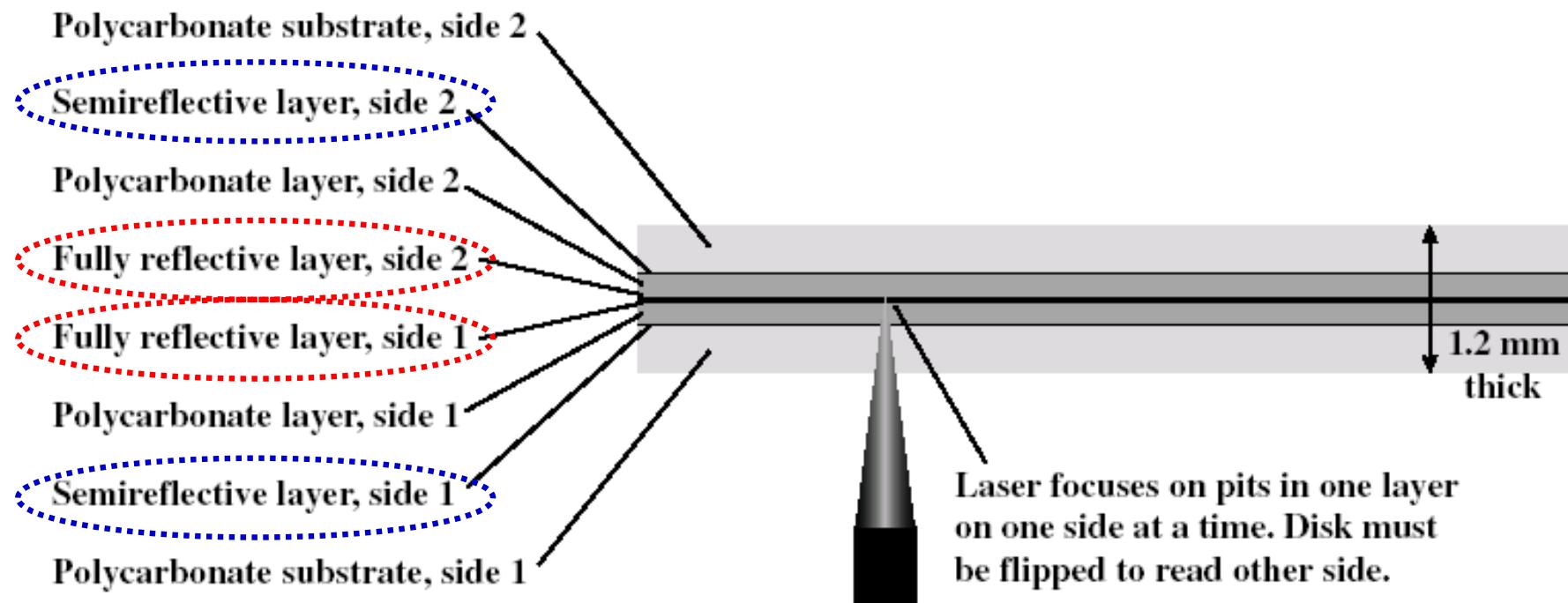
- *CD-Recordable (CD-R)*
  - ❖ Hanya dapat ditulisi **satu kali saja**
  - ❖ Digunakan untuk membuat *copy* data dalam jumlah kecil
  - ❖ Kompatibel dengan drive CD ROM
- *CD-RW*
  - ❖ Dapat ditulisi **berkali-kali**
  - ❖ Kebanyakan kompatibel dengan drive CD ROM
  - ❖ Sebagai penyimpanan data sekunder
  - ❖ Mempunyai kehandalan tinggi
  - ❖ Tahan lama

# DVD (1)

---

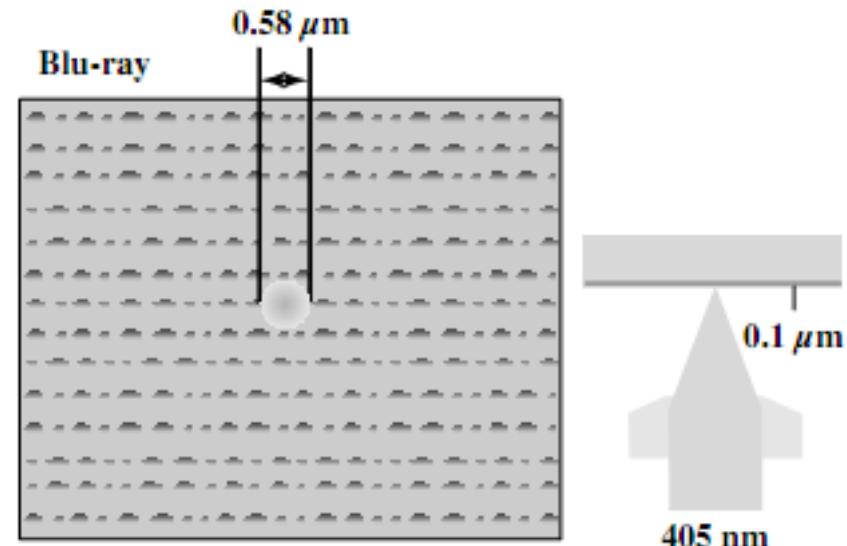
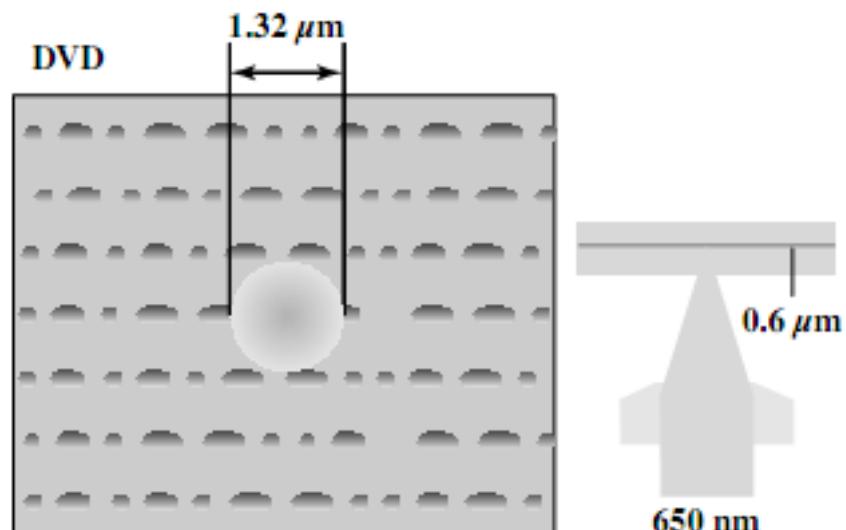
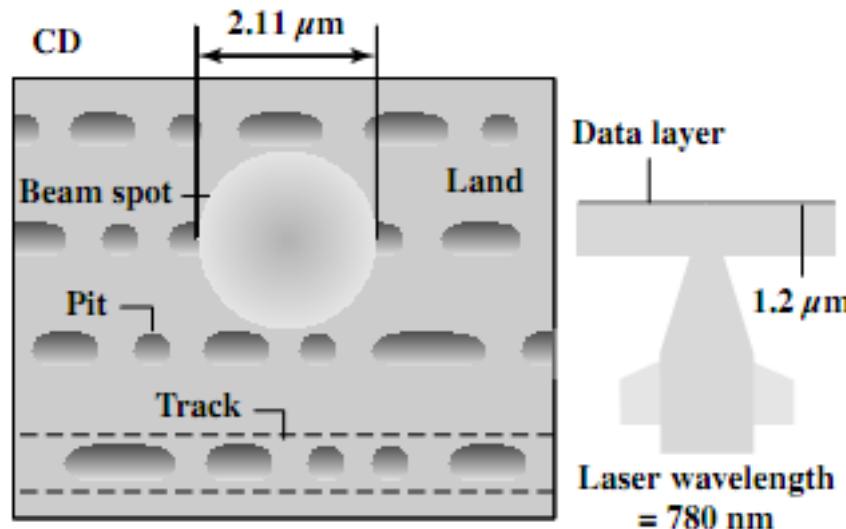
- DVD = *Digital Video Disk*
  - ❖ Hanya menyimpan data video saja
- DVD = *Digital Versatile Disk*
  - ❖ Dapat menyimpan data komputer dan data video
- Macam-macam DVD:
  - (a) DVD-ROM
    - DVD-5: satu sisi dan satu lapis, kapasitas total = 4,37 GB
    - DVD-9: satu sisi dan dua lapis dimana kapasitas setiap lapisan adalah 4,37 GB dan 7,95 GB, sehingga kapasitas total menjadi 12,32 GB
    - DVD-10: dua sisi masing-masing satu lapis, kapasitas total sebesar 8,74 GB
    - DVD-18: dua sisi masing-masing dua lapis, kapasitas totalnya sebesar 15,9 GB
  - (b) DVD-RW (Readable-Writeable)

# DVD (2)



(b) DVD-ROM, double-sided, dual-layer - Capacity 17 GB

# Karakteristik memori optik



# Referensi

- Stalling, William. 2016. "*Computer Organization and Architecture: Designing for Performance*". 10<sup>th</sup> edition