



PROGRAM STUDI
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

MATA KULIAH
ORGANISASI DAN ARSITEKTUR
KOMPUTER

Desain Sistem Memori

- ✓ Konsep Dasar
- ✓ *Cache Memory*
- ✓ *Fungsi Cache-Mapping*
- ✓ *Main Memory*
- ✓ *Virtual Memory*
- ✓ *Read Only Memory*

Tim pengampu

Sistem Komputer, Komunikasi dan Keamanan Data

T.A. 2020

Capaian Pembelajaran

Mahasiswa dapat memahami tentang konsep dasar memori, *cache memory*, *main memory*, *secondary memory* dan *virtual memory*.



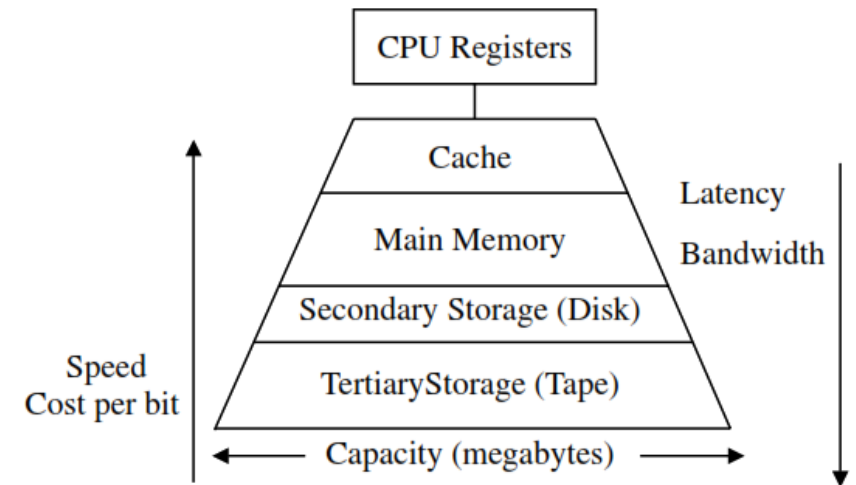
Konsep Dasar

Secara umum, memori diukur dari beberapa parameter:

- Access type
- Cycle time
- Capacity
- Latency
- Bandwidth
- Cost

Sehingga pada hirarki memori, *cache* berada pada posisi tertinggi dengan kapasitas terkecil (*capacity*) namun memiliki kecepatan (*cycle time*, *latency*, *bandwidth*) dan harga tertinggi (*cost*). Diikuti *main memory* (*solid-state semiconductor*), *secondary memory* (*disk*) dan *tertiary memory* (*tape*).

Konsekuensi penekanan harga (*cost*) memunculkan memori dengan kecepatan lebih rendah dengan kapasitas besar (*capacity*)



Gambar 9.1 Memory hierarchy

Parameter pengukur

Access type, merujuk kepada cara mengakses selama operasi *read* and *write*.

Cycle time, jarak antara waktu memulai operasi *read* hingga operasi *read* berikutnya.

Capacity, terukur dengan satuan *byte*.

Latency, interval waktu antara request dan akses kepada bit pertama sebuah informasi didalam memori.

Bandwidth, merupakan ukuran akan jumlah bit/ detik yang mampu diakses.

Cost, pengukuran berupa harga / megabyte.

	Access type	Capacity	Latency	Bandwidth	Cost/MB
CPU registers	Random	64–1024 bytes	1–10 ns	System clock rate	High
Cache memory	Random	8–512 KB	15–20 ns	10–20 MB/s	\$500
Main memory	Random	16–512 MB	30–50 ns	1–2 MB/s	\$20–50
Disk memory	Direct	1–20 GB	10–30 ms	1–2 MB/s	\$0.25
Tape memory	Sequential	1–20 TB	30–10,000 ms	1–2 MB/s	\$0.025

Tabel 9.1. Parameter pada Hirarki Memori

Cache Memory

Diperkenalkan oleh **Maurice Wilkes** pada 1965, ia membedakan dua jenis *main memory* : *conventional* dan *slave memory*.

Slave memory merupakan level ke-dua dari *conventional memory* yang memiliki kecepatan tinggi, yang sekarang lebih dikenal dengan *cache memory* (berasal dari makna kata *cache* : lokasi aman untuk menyimpan dan menyembunyikan sesuatu)

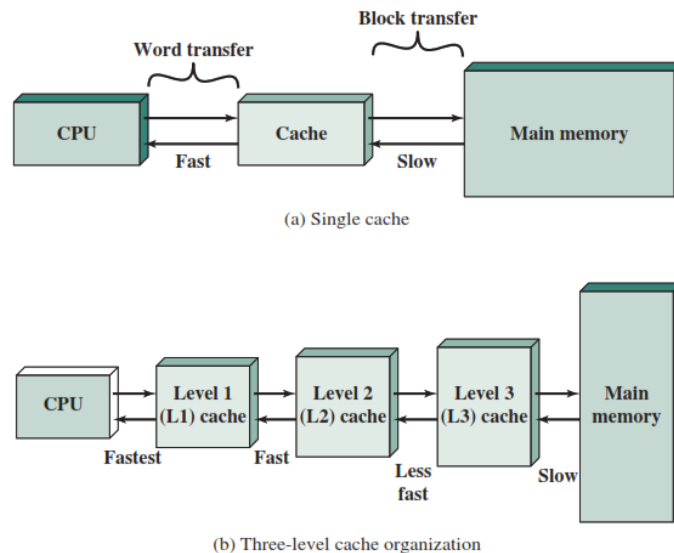
Tujuan dari pemanfaatan *cache memory* adalah untuk menyimpan informasi yang sering digunakan CPU, dengan lokasi yang lebih dekat kepada CPU dibanding lokasi *main memory*. Sehingga, ada sebagian data aktif yang berada didalam *main memory* diduplikasi kedalam *cache memory* dalam durasi waktu tertentu .

Maka, ketika CPU me-request data dari *memory* pencarian akan dilakukan pada *cache memory* terlebih dahulu. Jika data ditemukan (*cache hit*) maka data akan dimuat untuk dilibatkan dalam eksekusi oleh CPU dan jika tidak ditemukan (*cache miss*) maka pencarian akan dilakukan pada *main memory*. Dengan asumsi data ditemukan di *main memory*, maka data tersebut akan diduplikasikan ke *cache memory*.

Cache Size

Desain cache semakin berkembang dimana keberadaannya merupakan salah satu faktor pendukung dalam meningkatkan kinerja CPU.

Semakin besar kapasitas *cache memory* maka semakin baik pula kinerja CPU, namun disaat yang sama akan memperbesar biaya (cost) yang dibutuhkan.



Gambar 9.2. penggunaan dan tingkat kecepatan transfer

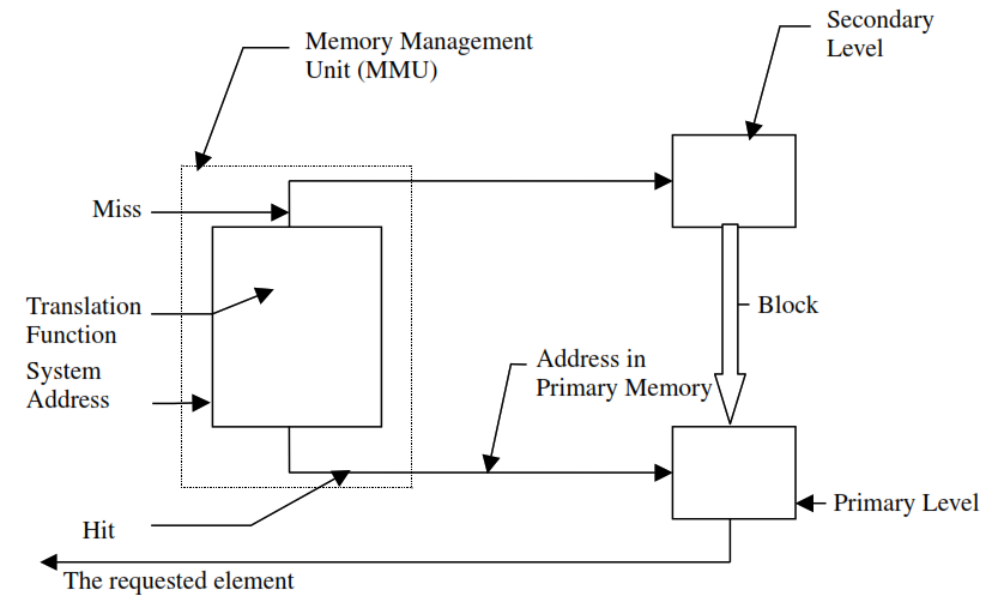
Processor	Type	Year of Introduction	L1 Cache ^a	L2 Cache	L3 Cache
IBM 360/85	Mainframe	1968	16–32 kB	—	—
PDP-11/70	Minicomputer	1975	1 kB	—	—
VAX 11/780	Minicomputer	1978	16 kB	—	—
IBM 3033	Mainframe	1978	64 kB	—	—
IBM 3090	Mainframe	1985	128–256 kB	—	—
Intel 80486	PC	1989	8 kB	—	—
Pentium	PC	1993	8 kB/8 kB	256–512 kB	—
PowerPC 601	PC	1993	32 kB	—	—
PowerPC 620	PC	1996	32 kB/32 kB	—	—
PowerPC G4	PC/server	1999	32 kB/32 kB	256 kB to 1 MB	2 MB
IBM S/390 G6	Mainframe	1999	256 kB	8 MB	—
Pentium 4	PC/server	2000	8 kB/8 kB	256 kB	—
IBM SP	High-end server/supercomputer	2000	64 kB/32 kB	8 MB	—
CRAY MTA ^b	Supercomputer	2000	8 kB	2 MB	—
Itanium	PC/server	2001	16 kB/16 kB	96 kB	4 MB
Itanium 2	PC/server	2002	32 kB	256 kB	6 MB
IBM POWER5	High-end server	2003	64 kB	1.9 MB	36 MB
CRAY XD-1	Supercomputer	2004	64 kB/64 kB	1 MB	—
IBM POWER6	PC/server	2007	64 kB/64 kB	4 MB	32 MB
IBM z10	Mainframe	2008	64 kB/128 kB	3 MB	24–48 MB
Intel Core i7 EE 990	Workstation/server	2011	6 × 32 kB/32 kB	1.5 MB	12 MB
IBM zEnterprise 196	Mainframe/server	2011	24 × 64 kB/128 kB	24 × 1.5 MB	24 MB L3 192 MB L4

Tabel 9.2. Contoh cache memory pada berbagai processor

Cache-Mapping Function

Request untuk mengakses data pada memori dilakukan oleh CPU dengan memberikan alamat lokasi data yang diinginkan. Alamat yang diberikan CPU bisa berupa alamat yang berada di cache (*cache hit*), atau jika tidak bisa ditemukan di *main memory*.

Pada gambar 9.3 *system address* adalah alamat data yang di *request* oleh CPU, lalu diterjemahkan oleh *translation function* (MMU). Jika alamat ditemukan (*cache hit*), maka data dari cache (*primary level*) dikirimkan kembali ke CPU. Namun jika alamat tersebut tidak ditemukan (*cache miss*), maka data akan dicari di *main memory* (*secondary level*) untuk kemudian disimpan di *cache* (*primary level*) sehingga dapat dikirimkan kembali ke CPU.



Gambar 9.3. address mapping operation

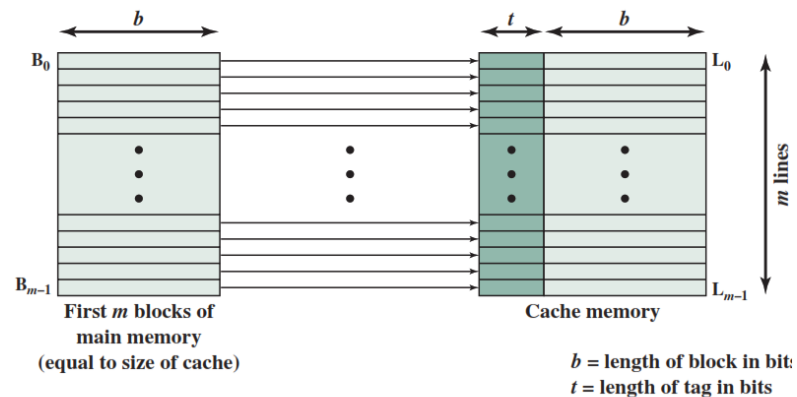
Cache Memory Organization

Direct, teknik paling sederhana dalam pengaturan mapping.

Yaitu dengan memetakan setiap 1 *main memory block* pada 1 *cache line*. Menuliskan alamat dalam 3 bagian.

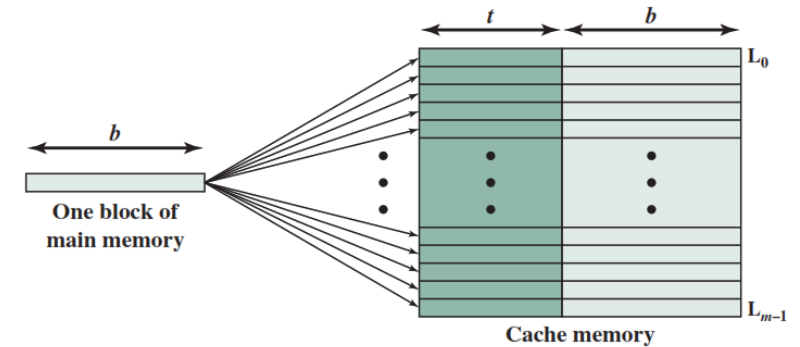
Associative, meletakkan *main memory block* ke dalam *cache line* manapun. Menuliskan alamat dalam 2 bagian.

Set associative, *cache* dibagi menjadi sejumlah *set*, setiap *set* terdiri dari sejumlah *block*. Menuliskan alamat dalam 3 bagian.

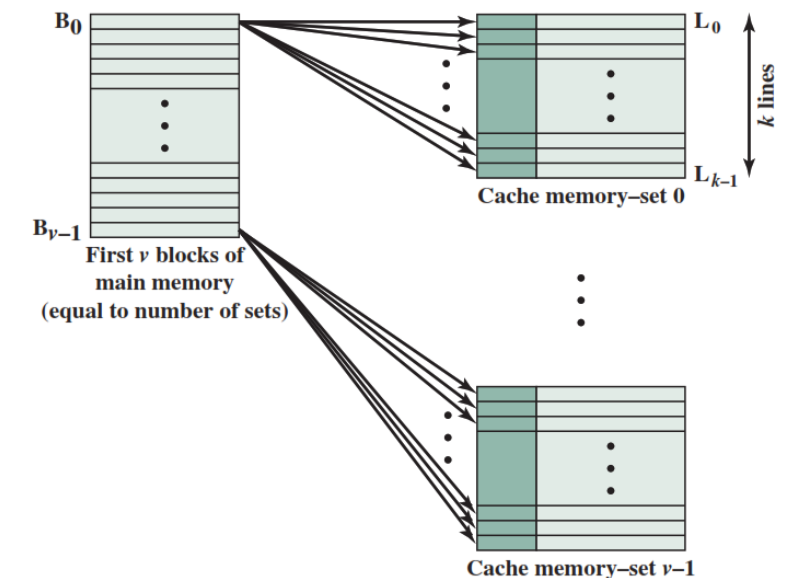


(a) Direct mapping

b = length of block in bit
 t = length of tag in bits



(b) Associative mapping



(a) v associative-mapped caches

Gambar 9.4. cache memory organization (*direct, associative, set associative*)

Main Memory

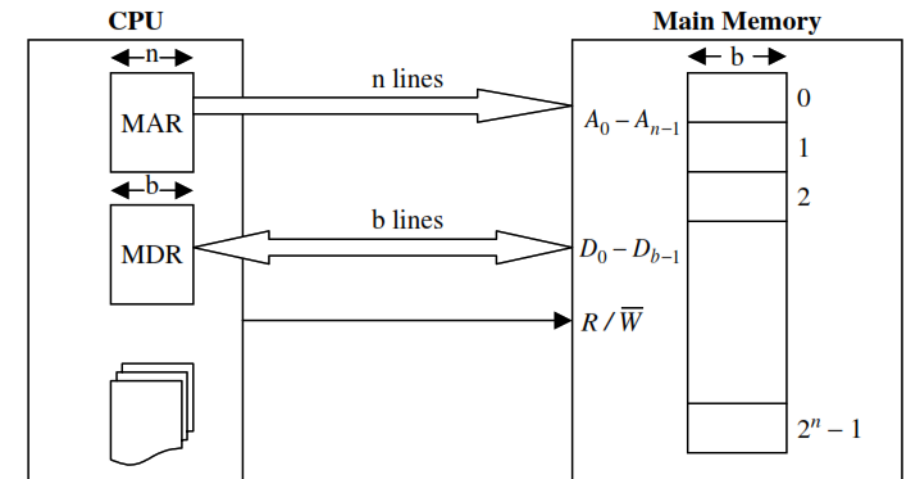
Main Memory atau sering disebut *random access memory* (RAM) menyediakan ruang penyimpanan utama bagi komputer.

Bersifat **volatile**, yaitu jenis penyimpanan yang akan kehilangan kemampuan menyimpan data didalamnya saat asupan listrik terganggu/ hilang.

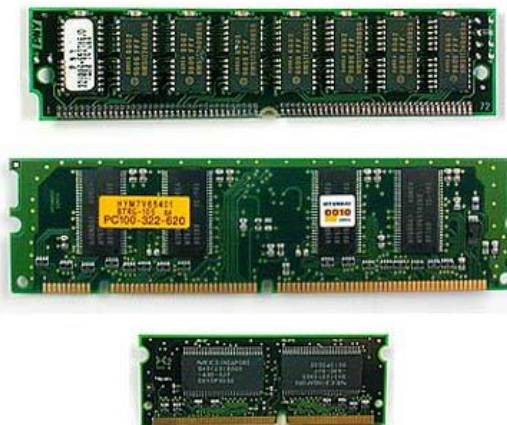
Terdapat 2 jenis RAM :

Dynamic RAM (DRAM), tiap *memory cell*-nya terdiri dari sebuah *transistor* dan sebuah *capasitor*, dimana *capasitor* membutuhkan asupan daya berkala untuk tetap dapat menyimpan bit 1.

Static RAM (SRAM), tersusun dari rangkaian 4 hingga 6 *transistor* yang akan tetap menyimpan nilai bit 1 hingga nilainya dirubah.



Gambar 9.5. korelasi *main memory* dengan CPU (atas)
Contoh RAM (bawah)



Read Only Memory

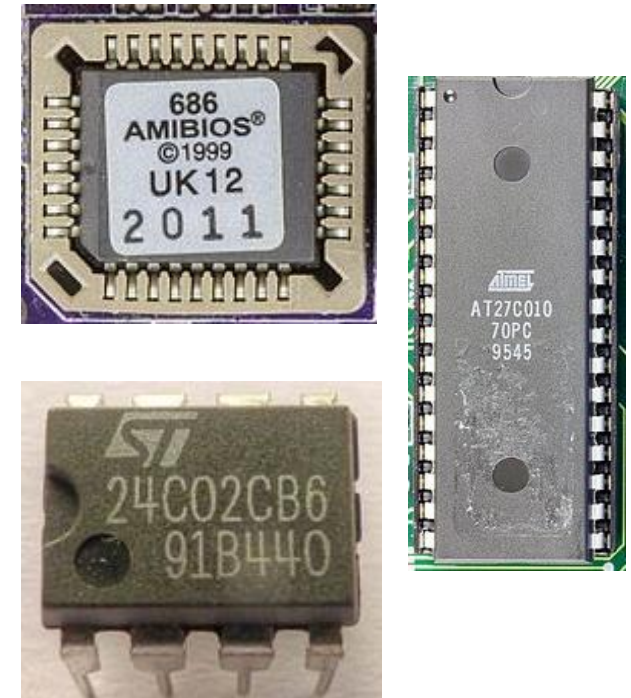
ROM merupakan jenis memori yang nilai datanya permanen/ tidak dapat dirubah (*read-only = unerasable*), bersifat *nonvolatile*.

Dimanfaatkan untuk microprogramming, dengan pemanfaatan sbb :

- *Library subroutine* untuk fungsi- fungsi yang sering digunakan
- *System Program*
- Tabel fungsi

Jenis- jenis ROM :

- *Programmable ROM* (PROM)
- *Erasable Programmable ROM* (EPROM)
- *Electrically Erasable Programmable ROM* (EEPROM)



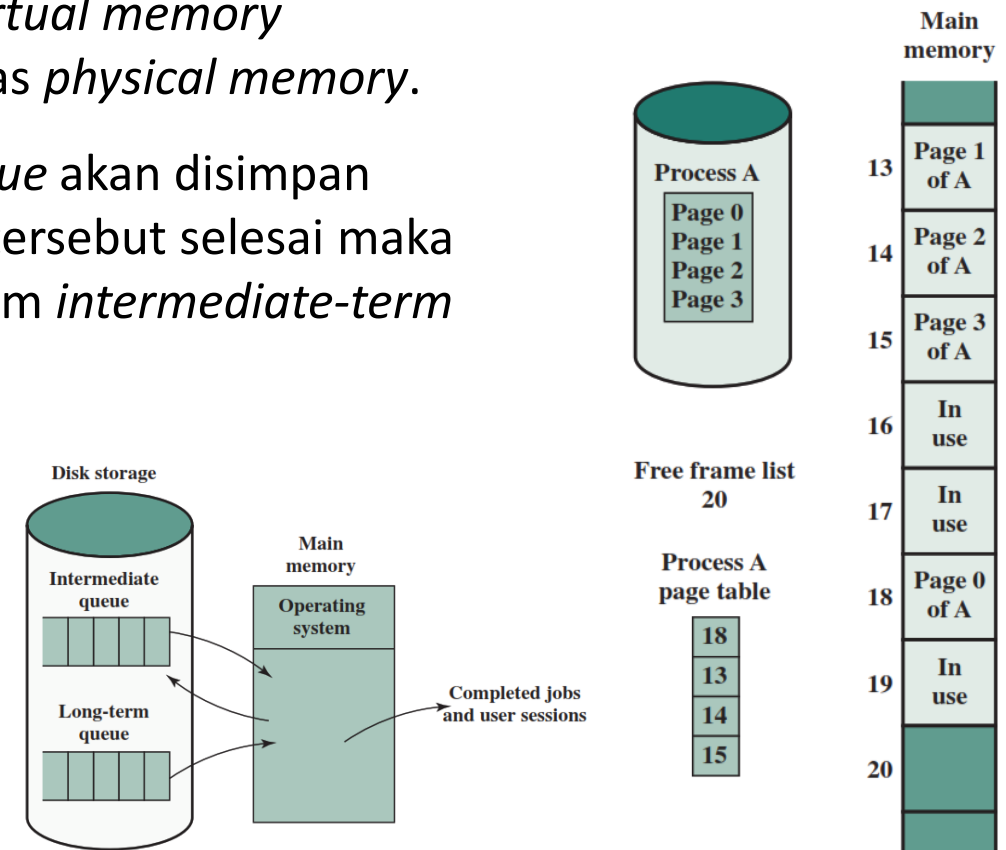
Gambar 9.6. ROM (atas), EPROM (kanan), EEPROM (bawah)

Virtual Memory

Bertujuan untuk mengoptimalkan kinerja komputer, *virtual memory* memanfaatkan *hard disk* untuk menambah kapasitas *physical memory*.

Swapping, proses yang terjadwal dalam *long-term queue* akan disimpan kedalam disk hingga ia dieksekusi, dan saat proses tersebut selesai maka ia keluar dari main memory dan dimasukkan kedalam *intermediate-term queue*.

Paging, proses dibagi menjadi potongan - potongan program (*page*) dimasukkan kedalam potongan-potongan memory (*frame*). SO menciptakan *page table* dimana didalamnya tersimpan daftar lokasi *frame* dimana *page* tersimpan.



Gambar 9.8. *swapping* (kiri), *paging* (kanan)

Magnetic Disk

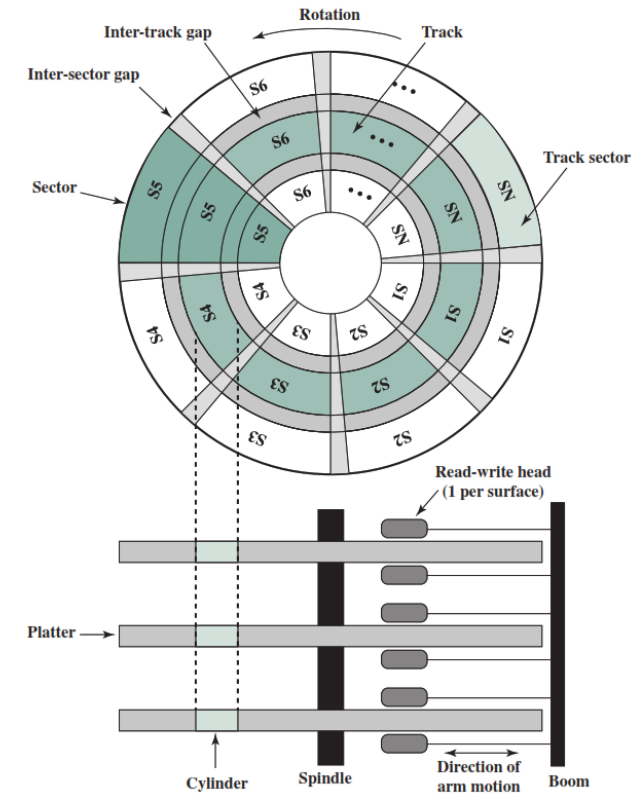
Media berupa piringan dengan lapisan berbahan bermagnet, berputar pada **spindle** pada kecepatan tertentu (*rpm*).

Sebuah hard disk bisa tersusun dari beberapa **platter** (fisik), tersusun dan berputar pada sebuah **spindle**. Piringan **platter** terbagi menjadi sejumlah **track**, dan setiap **track** terbagi menjadi sejumlah **sector** (logic).

Proses baca/ tulis dilakukan oleh **head** yang dapat mengakses setiap sisi dari **platter** dengan menggerakkan **arm** sehingga mendapatkan posisi **sector** yang dituju.

Seperangkat arm digunakan untuk memastikan head dapat mengakses lokasi sector yang dituju, namun hanya 1 arm yang diperbolehkan bergerak dalam setiap operasi baca/ tulis.

Cylinder merupakan sejumlah **track** serupa yang berada pada **platter** yang berbeda.



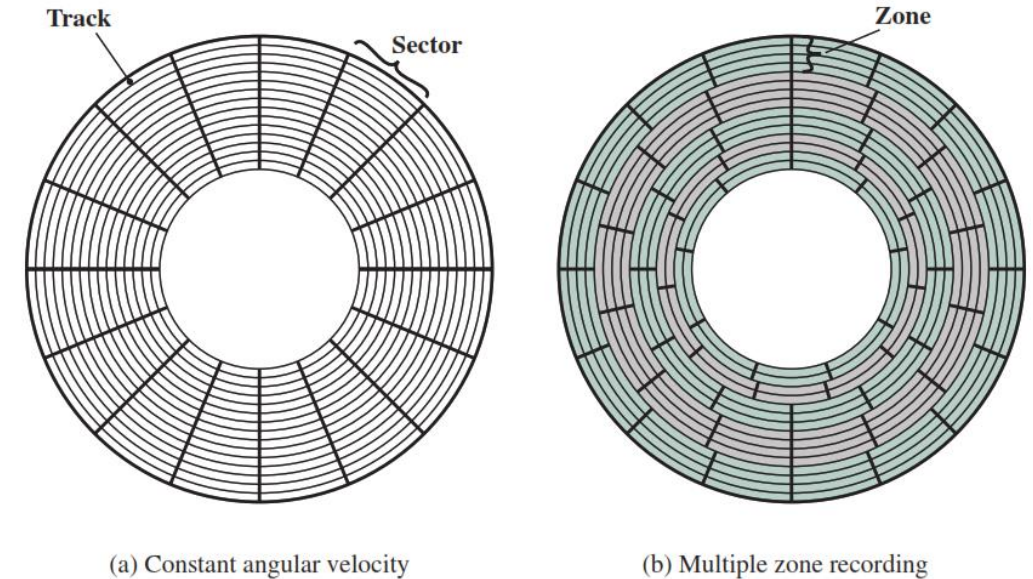
Gambar 9.9. tata letak disk

Magnetic Disk - Lanjt.

Pada hard disk modern, pembagian *sector* tidak lagi menggunakan **constant angular velocity** (gambar 9.10 a) yang membagi track berdasar sudut yang sama. $15\text{track} \times 16\text{sector/track} = 240\text{sector}$.

Menggunakan **multiple zone recording** yang menggabungkan beberapa *track* terdekat kedalam sebuah *zone* sehingga kapasitas tiap *zone* sama.

Pada gambar 9.10b terdapat terbagi 5 **zone**, dimana setiap **zone** tersusun dari beberapa **track**. Dimulai dari **zone** terdalam tersusun dari 2 **track** yang terbagi menjadi 9 **sector**; **zone** berikutnya tersusun dari 2 **track** dan terbagi menjadi 11 **sector**; **zone-3** tersusun dari 3 **track** dan terbagi menjadi 12 **sector**; **zone-4** tersusun dari 3 **track** dan 14 **sector**; **zone-5** tersusun dari 3 **track** dan 16 **sector**.



Gambar 9.10. angular sector (a), MZR (b)

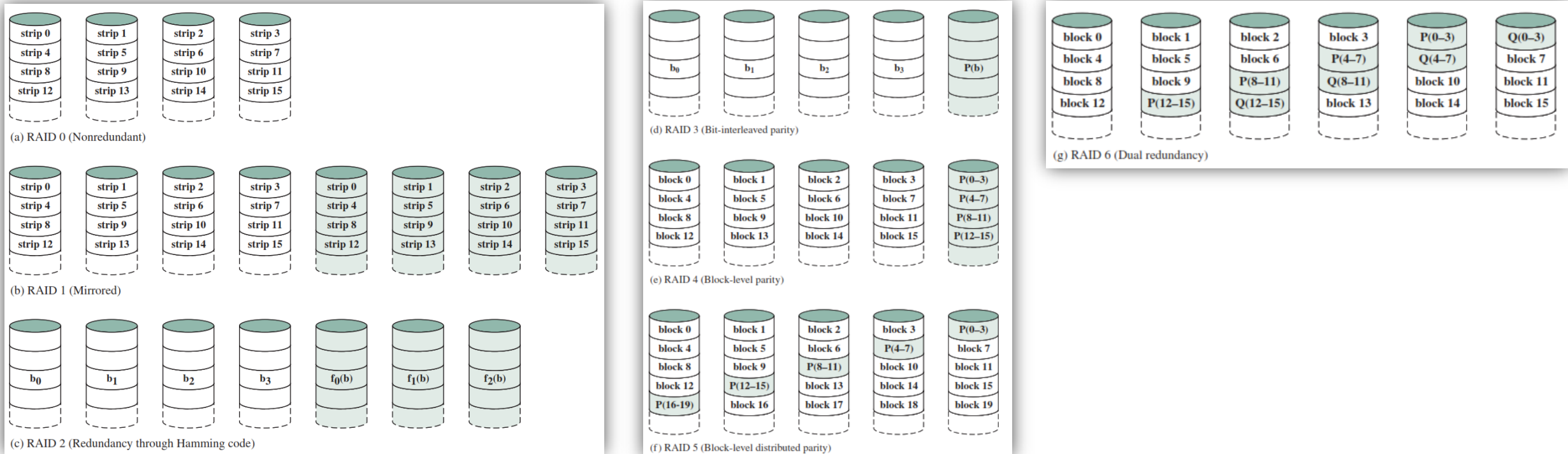
Redundant Array of Independent Disks (RAID)

RAID merupakan sejumlah *physical disk drive* yang dilihat oleh sistem operasi sebagai sebuah *logical drive*.

Category	Level	Description	Disks 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

Tabel 9.3. RAID Levels

Redundant Array of Independent Disks (RAID) – Lanjt.



Tabel 9.11. Layout of RAID

SUMBER PUSTAKA

- Mostafa dan Hesham.2005.Fundamentals Of Computer Organization And Architecture.New Jersey : Wiley Interscience
- W. Stallings.2016. Computer Organization and Architecture.Hoboken:Pearson Education
- A.S. Tanenbaum.Structured Computer Organization.New Jersey : Pearson Prentice Hall



THANKS

ANY QUESTIONS?