

IF3055 - Manajemen I/O Disk

Henny Y. Zubir
STEI - ITB

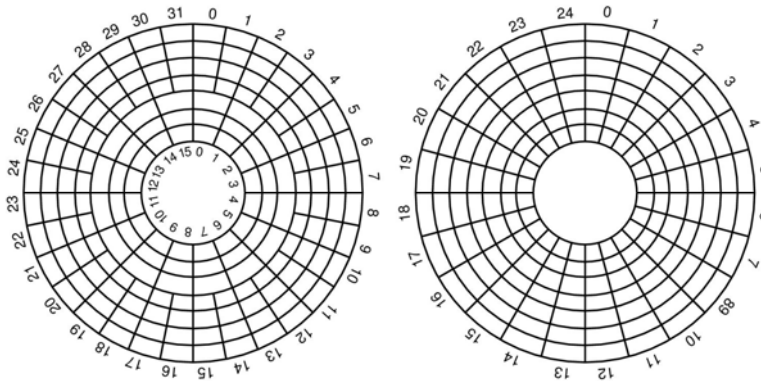


Ikhtisar

- Struktur Disk
- Performansi
- Penjadwalan Lengan Disk
- Pengelolaan Disk
- Keandalan Disk

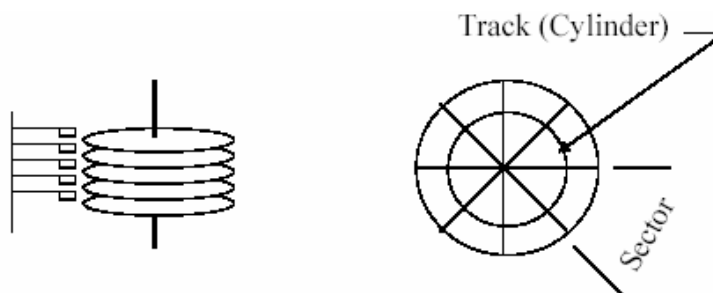


Struktur Disk (1)

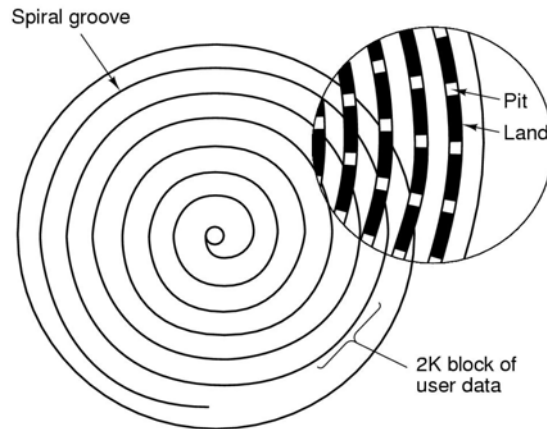


- Geometri fisik suatu disk dengan 2 zona
- Kemungkinan geometri virtual untuk disk ini

Struktur Disk (2)

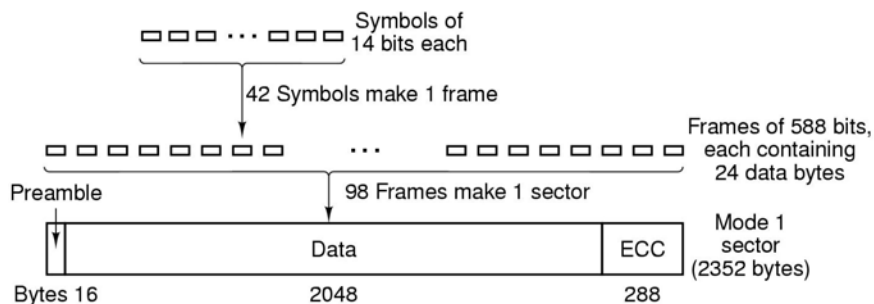


Struktur Disk (3)



- Struktur perekaman pada CD atau CD-ROM

Struktur Disk (4)



- Layout data logik pada CD-ROM

Penjadwalan Lengan Disk (1)

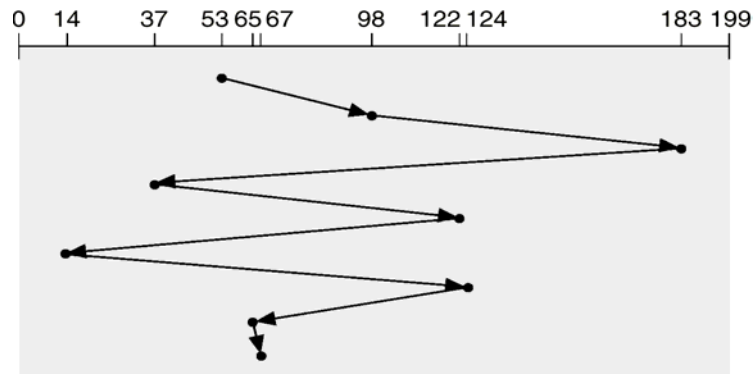
- Efisiensi penggunaan disk meliputi:
 - waktu akses
 - bandwidth disk
- **Waktu akses** memiliki dua komponen utama:
 - **Seek time:** waktu yang diperlukan untuk menggerakkan head ke silinder yang berisi sektor yang diinginkan
 - **Rotational latency:** waktu tambahan yg dibutuhkan untuk menunggu disk memutar sektor yang diinginkan ke head
- Seek time mendominasi → minimalkan seek time
- Seek time \approx seek distance
- **Bandwidth disk:** merupakan banyaknya byte yang ditransfer, dibagi dengan total waktu antara permintaan pertama layanan dan penyelesaian transfer terakhir

Penjadwalan Lengan Disk (2)

- Penjadwalan disk perlu dilakukan karena:
 - urutan cluster/blok yang dibaca mempengaruhi seek time → mempengaruhi kinerja I/O secara keseluruhan
 - pada sistem multiprogramming, permintaan pengaksesan disk lebih banyak daripada yang dapat dilayani
- Tujuan penjadwalan disk:
 - mengoptimalkan kinerja I/O
 - permintaan dilayani dengan urutan pergerakan mekanis yang minimum → meminimumkan pergerakan lengan disk (*seek time*)

Penjadwalan Lengan Disk: FCFS

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



- Pergerakan head sebanyak 66 silinder

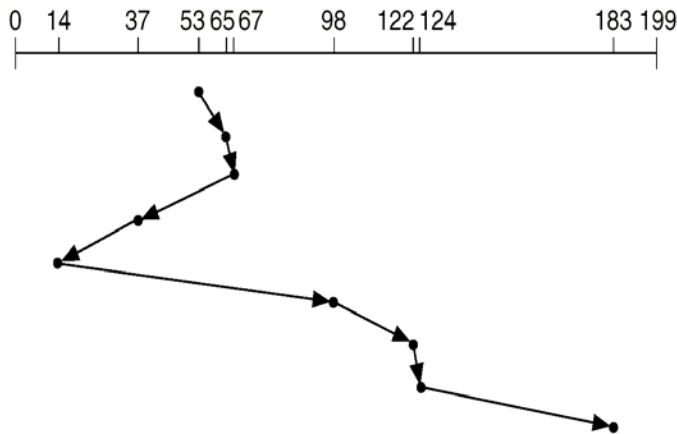
Penjadwalan Lengan Disk: SSTF (1)

- Memilih request dengan seek time minimum dari posisi head saat ini
- Mirip dengan penjadwalan SJF → dapat menyebabkan starvation pada beberapa request
- Contoh pd hal 12: total pergerakan head 236 silinder

Penjadwalan Lengan Disk: SSTF (2)

queue = 98, 183, 37, 122, 14, 124, 65, 67

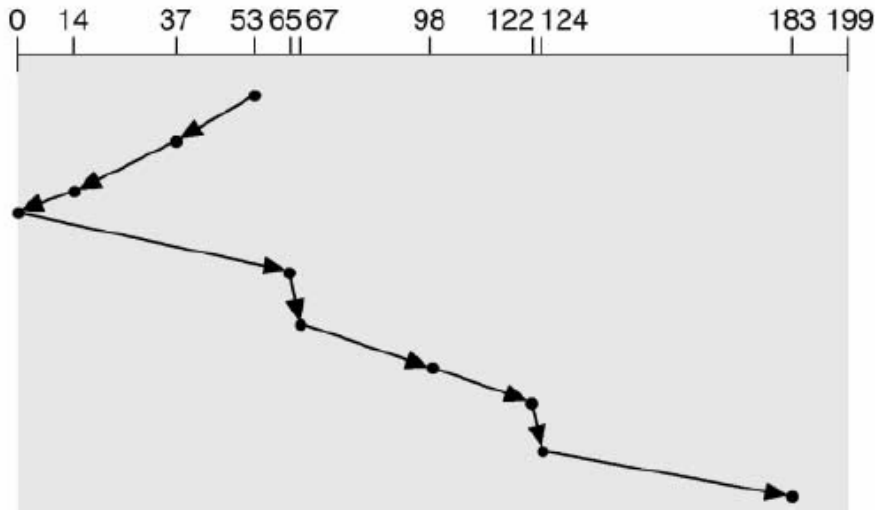
head starts at 53



Penjadwalan Lengan Disk: SCAN (1)

- Lengan disk mulai dari salah satu ujung disk, dan bergerak ke arah ujung lainnya
 - Melayani request hingga sampai ke ujung lainnya dari disk
 - Pergerakan head kemudian berbalik, dan terus melayani permintaan
- Disebut juga algoritma **elevator**
- Contoh: total pergerakan head=208 silinder

Penjadwalan Lengan Disk: SCAN (2)



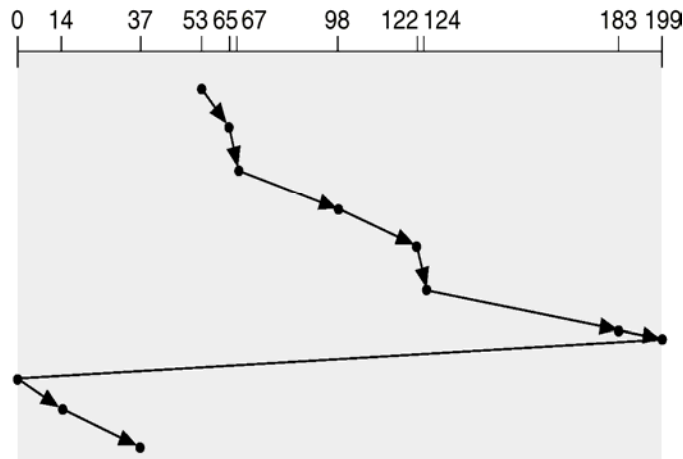
Penjadwalan Lengan Disk: C-SCAN (1)

- Memberikan waktu tunggu yg lebih sedikit variasinya dibandingkan dengan SCAN
- Head melayani request dengan berpindah dari satu ujung disk ke ujung lainnya. Jika telah mencapai ujung lainnya, head kembali ke awal disk tanpa melayani request dalam perjalanan kembalinya
- Menganggap silinder sebagai list sirkuler (terhubung langsung dari silinder terakhir ke silinder pertama)

Penjadwalan Lengan Disk: C-SCAN (2)

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



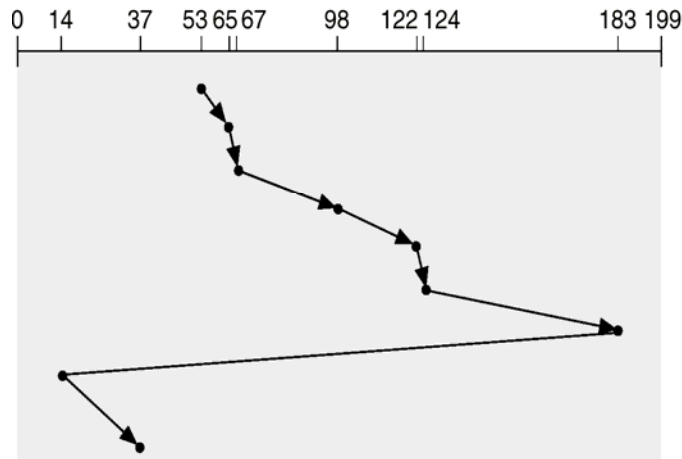
Penjadwalan Lengan Disk: C-LOOK (1)

- Variasi dari C-SCAN
- Lengan disk hanya berpindah sejauh request terakhir pada tiap arah, dan kemudian langsung berbalik arah tanpa sampai ke ujung disk

Penjadwalan Lengan Disk: C-LOOK (2)

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53

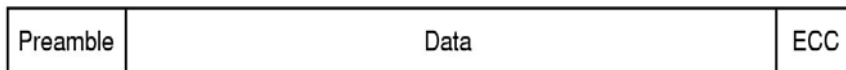


Pemilihan Algoritma Penjadwalan

- SSTF bersifat umum dan alami
- SCAN dan C-SCAN memberikan performansi lebih baik untuk sistem dengan akses ke disk sangat tinggi
- Performansi tergantung pada tipe dan banyaknya permintaan
- Request utk layanan disk dapat dipengaruhi oleh metode alokasi file
- Algoritma penjadwalan disk sebaiknya ditulis pada modul terpisah dari OS sehingga dapat diganti dgn algoritma lain jika perlu
- SSTF atau LOOK dapat menjadi algoritma default

Pengelolaan Disk: Formatting (1)

- **Low-level formatting (physical formatting):** membagi disk menjadi sektor sehingga dapat ditulis/dibaca oleh disk controller
 - mengisi disk dengan struktur data khusus untuk tiap sektor (header, data area, dan trailer)
 - Header dan trailer berisi informasi yang diperlukan oleh disk controller seperti **ECC (Error-Correcting Code)**
 - ECC digunakan untuk mengetahui **bad sector**

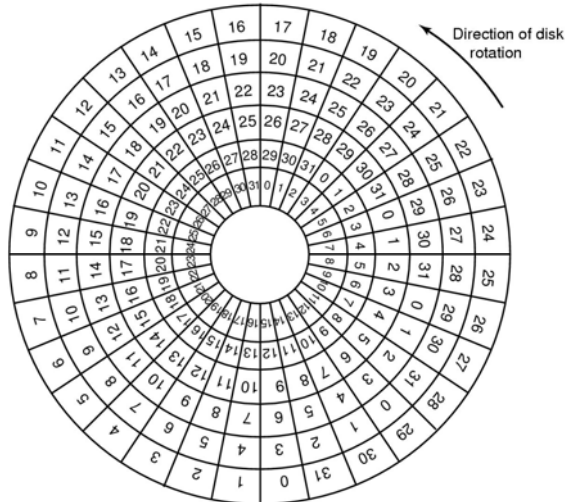


Pengelolaan Disk: Formatting (2)

- **Logical formatting:** membuat struktur data yang diperlukan OS untuk menyimpan data (membuat file system)
 - **Mempartisi** disk menjadi beberapa kelompok silinder
- Program tertentu dapat menggunakan partisi disk sebagai array blok logik tersendiri (bukan menggunakan struktur data sistem file)
 - Contoh: untuk keperluan basis data

Pengelolaan Disk: Formatting (3)

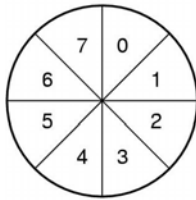
- Ilustrasi sayatan silinder



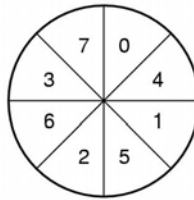
Pengelolaan Disk: Formatting (3)

- Disk drive dialamatkan sebagai array blok logik 1-dimensi yang berukuran besar, dimana blok logik merupakan unit transfer terkecil
- Array blok logik ini dipetakan menjadi sektor pada disk secara sekuensial
 - Sektor 0 merupakan sektor pertama dari track pertama pada silinder terluar
 - Pemetaan berlanjut terurut pada track → track pada silinder yang sama → silinder dari luar ke dalam

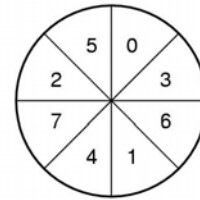
Pengelolaan Disk: Formatting (4)



(a)



(b)



(c)

- a. Tanpa interleaving
- b. Interleaving tunggal
- c. Interleaving ganda

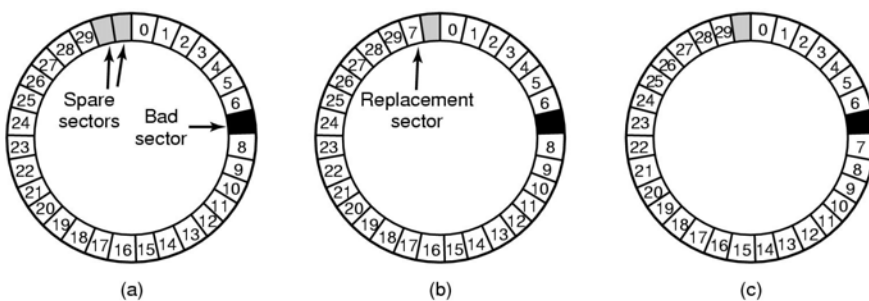
Pengelolaan Disk: Boot Block

- Boot Block
 - Program **bootstrap** digunakan untuk menjalankan program awal pada saat komputer baru dinyalakan
 - Program bootstrap dapat disimpan di ROM, namun tidak fleksibel
 - Kebanyakan sistem menyimpan **bootstrap loader** di ROM, yang fungsinya hanya untuk membawa bootstrap program dari disk → program bootstrap lebih fleksibel
 - Program bootstrap disimpan di **boot blocks**, pada lokasi yang tetap pada disk (**boot disk/system disk**)

Pengelolaan Disk: Bad Blocks (1)

- Pada disk sederhana seperti IDE, bad block ditangani secara manual
- Program khusus (seperti chkdsk) dijalankan untuk memeriksa block yang rusak dan ditandai; data dalam bad block biasanya hilang
- Pada disk yang lebih rumit seperti SCSI, pemeriksaan bad block dilakukan secara periodik
- Controller dapat mengganti block yang rusak dengan block cadangan

Pengelolaan Disk: Bad Blocks (2)



- Suatu track disk yang memiliki bad sector
- Mengganti bad sector dengan sector cadangan
- Memindahkan semua sector untuk mem-bypass bad sector

Pengelolaan Disk: Swap Space

- Swap-space — ruang disk yang digunakan oleh virtual memory sebagai ekstensi dari memori utama
- Swap-space dapat diambil dari sistem file biasa, atau yang lebih umum menggunakan partisi tersendiri
- Pengelolaan swap-space
 - 4.3BSD mengalokasikan swap space ketika proses mulai; menyimpan *text segment* (program) dan *data segment*
 - Kernel menggunakan *swap maps* untuk melacak penggunaan swap space
 - Solaris 2 mengalokasikan swap space hanya ketika page dipaksa keluar dari memori, bukan ketika page virtual memori pertama kali dibuat

Keandalan Disk

- Beberapa perbaikan dalam teknik penggunaan disk mencakup penggunaan beberapa disk yang saling bekerjasama
- Disk striping menggunakan sekelompok disk sebagai satu unit penyimpanan
- Skema RAID dapat meningkatkan performansi dan keandalan sistem penyimpanan dengan menyimpan data redundan
 - **Mirroring** atau **shadowing** untuk memelihara duplikat tiap disk
 - **Block interleaved parity** menggunakan lebih sedikit redundansi