

ORGANISASI BERKAS SEKUENSIAL

Dalam organisasi ini, record-record direkam secara berurutan pada waktu berkas ini dibuat dan harus diakses secara berurutan pada waktu berkas ini digunakan sebagai input. Berkas sekuensial sangat cocok untuk akses yang sekuensial, misalnya dalam aplikasi dimana sebagian besar atau semua rekaman akan diproses. Sebagai contoh adalah membuat daftar mahasiswa dalam sebuah program studi. Melakukan akses secara sekuensial berarti proses akan berpindah dari satu rekaman ke rekaman berikutnya secara langsung.

1	2	3	...	i	i+1	i+2	...	n-1	n
---	---	---	-----	---	-----	-----	-----	-----	---

Pencarian secara sekuensial adalah memproses rekaman – rekaman dalam berkas sesuai urutan keberadaan rekaman – rekaman tersebut sampai ditemukan rekaman yang diinginkan atau semua rekaman terbaca. Berkas sekuensial juga dapat diproses secara tunggal dan langsung, jika diketahui subskripnya. Tetapi bagaimana kalau subskrip yang dimiliki bukan identitas utama rekaman, misal ,Nama Mahasiswa` pada file berikut ini :

Nama Mahasiswa	Nomor Induk Mahasiswa	Jenjang	Program Studi	Dosen Wali	SPP	Data lain
Dian Kartika	11.50.001	S1	Sistem Informasi	Made	500.000	...
Syda Arlin	11.50.011	S1	Sistem Informasi	Sugeng	500.000	...
Yuniati	11.30.012	DIII	Manajemen Informatika	Made	400.000	...
Sunaryono	11.30.013	DIII	Manajemen Informatika	Made	400.000	...
Zainul	11.55.024	S1	Teknik Informatika	Sugeng	500.000	...
Yenny Noorma	11.50.021	S1	Sistem Informasi	Sugeng	500.000	...
Mustafa	11.50.025	S1	Sistem Informasi	Sugeng	500.000	...
Kusmiyati	11.55.027	S1	Teknik Informatika	Made	500.000	...
Susiana	11.55.028	S1	Teknik Informatika	Made	500.000	...
Dewi Dwi	11.30.014	DIII	Manajemen Informatika	Made	400.000	...

Pembacaan harus dilakukan secara sekuensial.Rekaman demi rekaman, sampai ,Nama Mahasiswa` yang sesuai ditemukan. Misalnya untuk pembacaan rekaman dengan ,Nama Mahasiswa` = Sunaryono, diperlukan probe (akses terhadap lokasi yang berbeda) sejumlah 4 kali. Yang harus dilakukan agar kinerja pembacaan rekaman lebih baik, maka rekaman-rekaman dalam berkas mahasiswa tersebut diurutkan untuk mendapatkan pengurutan yang linier berdasarkan nilai kunci rekaman.Baik secara alphabetis maupun numeris. Hasil pengurutannya adalah sbb :

Nama Mahasiswa	Nomor Induk Mahasiswa	Jenjang	Program Studi	Dosen Wali	SPP	Data lain
Yuniati	11.30.012	DIII	Manajemen Informatika	Made	400.000	...
Sunaryono	11.30.013	DIII	Manajemen Informatika	Made	400.000	...
Dewi Dwi	11.30.014	DIII	Manajemen Informatika	Made	400.000	...
Dian Kartika	11.50.001	S1	Sistem Informasi	Made	500.000	...
Syda Arlin	11.50.011	S1	Sistem Informasi	Sugeng	500.000	...
Yenny Noorma	11.50.021	S1	Sistem Informasi	Sugeng	500.000	...
Mustafa	11.50.025	S1	Sistem Informasi	Sugeng	500.000	...
Zainul	11.55.024	S1	Teknik Informatika	Sugeng	500.000	...
Kusmiyati	11.55.027	S1	Teknik Informatika	Made	500.000	...
Susiana	11.55.028	S1	Teknik Informatika	Made	500.000	...

Berkas diatas berisi rekaman mahasiswaurut berdasar ,Nomor Induk Mahasiswa'.Kolom ,Nomor Induk Mahasiswa' menunjukkan nilai yangurut dari kecil ke besar. Dengan demikian , hanya $n/2$ rekaman yang perlu diperiksa rekaman-demi-rekaman untuk menemukan rekaman yang diinginkan. Kalau pembacaan diteruskan melewati posisi dimana rekaman seharusnya berada (mengingat berkas sudah diurutkan), maka proses pencarian dihentikan. Untuk membaca ,Sunaryono' hanya diperlukan 2 probe, lebih kecil disbanding berkas sebelum diurutkan.Namun teknik tersebut masih kurang memuaskan untuk berkas dengan jumlah rekaman yang lebih besar.

A. PENCARIAN BINER (BINARY SEARCH)

Pencarian Biner dalah membandingkan kunci yang dicari dengan rekaman pada posisi tengah dari berkas. Bila sama (Kasus 1) rekaman yang diinginkan sudah ditemukan. Jika tidak sama (kasus 2), berarti separuh rekaman-rekaman dalam berkas akan dieliminasi dari perbandingan yang selanjutnya. Bila yang terjadi pada kasus 2, maka proses perbandingan terhadap rekaman pada posisi di tengah dilanjutkan menggunakan rekaman-rekaman yang tersisa. Jumlah probe (yang diperlukan untuk membaca sebuah rekaman) pada sebuah berkas dengan rekaman yang sudah diurutkan, dapat diperkecil dengan menggunakan teknik pencarian biner. Jika kunci cari < kunci tengah maka bagian berkas mulai dari kunci tengah sampai akhir berkas dieliminiasi. sebaliknya jika kunci cari >

kunci tengah, maka bagian berkas mulai dari depan sampai dengan kunci tengah dieliminasi.

Dengan mengulang proses perbandingan terhadap rekaman tengah, maka lokasi rekaman yang diinginkan akan ditemukan atau diketahui bahwa rekaman yang diinginkan tersebut tidak berada dalam berkas. Algoritma pencarian biner:

```
Proc pencarian_biner
*/ n buah rekaman dalam berkas diurutkan menaik menurut kunci
rekaman */
    AWAL := 1
    AKHIR := n
    While AWAL <= AKHIR do
        Tengah := [(AWAL+AKHIR)/2]
        If kunci(cari) = kunci(tengah)
            Then pencarian berakhir
        Else if kunci(cari) > kunci(tengah)
            Then AWAL := Tengah + 1
        Else AKHIR := tengah - 1
    End
    Rekaman tidak ditemukan
End pencarian_biner
```

B. PENCARIAN INTERPOLASI

Pencarian interpolasi (asumsinya kunci rekaman numeris) menentukan posisi yang akan dibandingkan berikutnya berdasar posisi yang di estimasi dari sisa rekaman yang belum diperiksa. Pencarian interpolasi tidak mencari posisi tengah, seperti algoritma pencarian biner, melainkan menentukan posisi berikutnya.

```

Proc pencarian_interpolasi
*/ n buah rekaman dalam berkas diurutkan menaik menurut kunci
rekaman */
    AWAL := 1
    AKHIR := n
    While AWAL <= AKHIR do

        BERIKUT := [AWAL +  $\frac{\text{kunci(cari)} - \text{kunci(AWAL)}}{\text{kunci(AKHIR)} - \text{kunci(AWAL)}} (\text{AKHIR} - \text{AWAL})$ ]

        If kunci(cari) = kunci(BERIKUT)
            Then pencarian berakhir
        Else if kunci(cari) > kunci(BERIKUT)
            Then AWAL := BERIKUT + 1
        Else AKHIR := BERIKUT - 1

    End
    Rekaman tidak ditemukan
End pencarian_interpolasi

```