

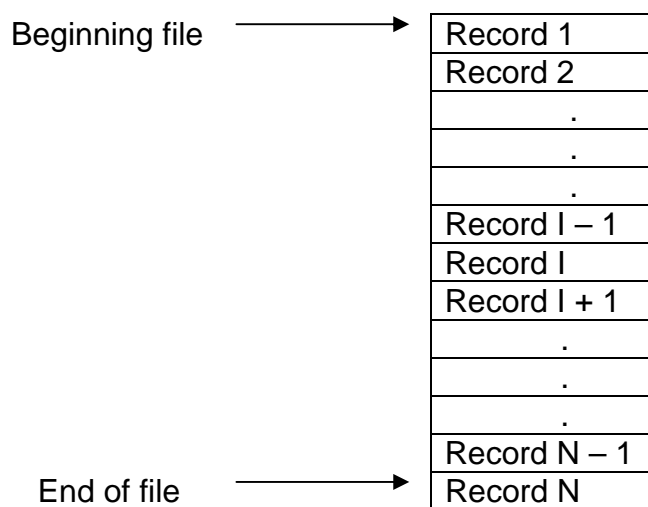
ORGANISASI BERKAS SEKUENSIAL

Organisasi berkas sekuensial adalah merupakan cara yang paling dasar untuk mengorganisasikan kumpulan record record dalam sebuah berkas.

Dalam organisasi berkas sekuensial, pada waktu record ini dibuat, record-record direkam secara berurutan.

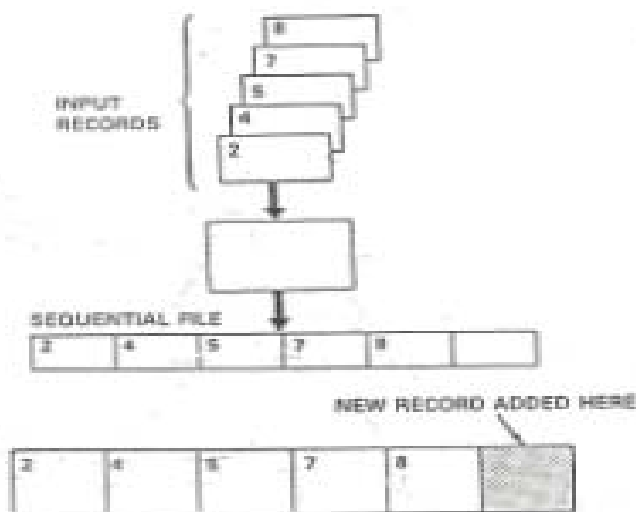
Record pertama ditempatkan pada posisi pertama dalam berkas, record kedua ditempatkan pada posisi kedua dalam berkas dan seterusnya.

Begitu pula pada waktu pengaksesan dan pada waktu berkas ini digunakan sebagai input, record-record harus diakses secara berurutan.



Gambar 1. Struktur Berkas Sekuensial

SEQUENTIAL ORGANIZATION



Gambar 2. Penambahan Berkas Sekuensial

Jadi dalam organisasi berkas sekuensial, bukan berarti bahwa record-record tersebut disimpan dalam urutan numerik .

Jika kita ingin menambahkan suatu record pada berkas sekuensial, maka record tersebut akan terletak pada **akhir berkas** .

Organisasi berkas sekuensial dapat terdiri dari record-record yang berbeda jenis

Contoh:

Dalam sistem penggajian terpadu (*intergrated personnel-payroll system*) mempunyai sebuah berkas pegawai (*employee file*) yang terdiri dari dua jenis record, yaitu:

personnel record dan payroll record

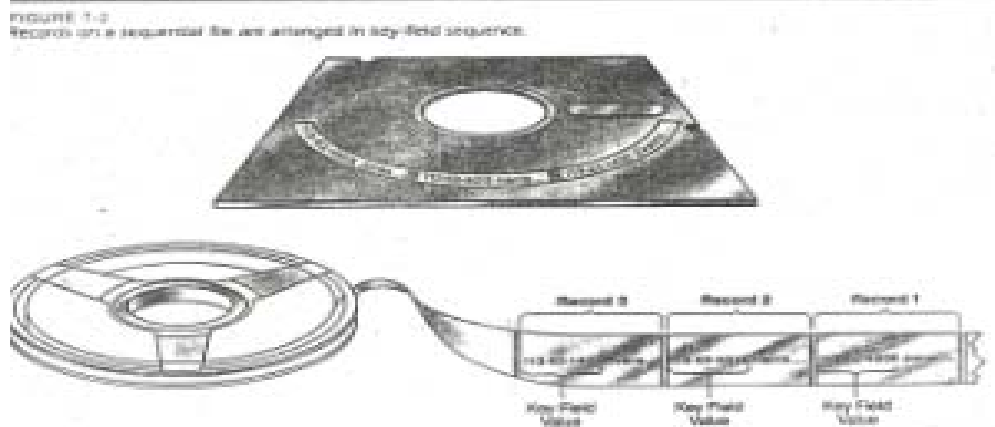
PERSONNEL										
REC TYPE	EMPNO	NAME	ADDRESS	MARITAL STATUS	SEX	HOMER LOC	PHY. CODE	DEPT OR TRNG
1										

PAYROLL										
REC TYPE	EMPNO	PLANT	ACCOUNT	RATE	DATE START	DATE CHANGE
2										

Gambar 3. Jenis Record

Record-record pada berkas tersebut tidak memerlukan format dan ukuran yang sama. Pada contoh tersebut, berkas disortir berdasarkan:

EMP-NO, REC – TYPE



Gambar 4. Record dalam Berkas Sekuensial

PROSES

Karena record-record dalam organisasi berkas sekuensial harus diakses secara berurutan, maka berkas sekuensial lebih sering menggunakan batch processing dari pada interactive processing.

Keuntungan dan Keterbatasan

Adapun keuntungan utama dari teknik organisasi berkas sekuensial adalah kemampuan untuk mengakses record berikutnya secara cepat. Keterbatasan dari organisasi berkas sekuensial adalah kita tidak dapat mengakses langsung pada record yang diinginkan.

POLA AKSES

Pola akses adalah penentuan akses berdasarkan field tertentu. Selama pola akses, berkas sekuensial dapat dipasangkan dengan record-record yang sudah diurut pada berkas, maka waktu aksesnya sangat baik .

Jadi kita harus menentukan pola akses terlebih dahulu, kemudian baru menentukan organisasi berkas sekuensial berdasarkan urutan yang sesuai dengan pola aksesnya, jangan sebaliknya .

Contoh:

Berkas gaji yang disusun secara sekuensial berdasarkan NIP, hendak diakses berdasarkan NAMA, maka program tidak baik .

Juga tidak baik mengakses record dengan urutan sebagai berikut:

NIP: 15024508 , NIP: 15024607, NIP: 15024115 ,NIP: 15028001
dimana NIP tersebut belum tersortir.

MEDIA PENYIMPANAN BERKAS SEKUENSIAL

Berkas sekuensial dapat disimpan dalam SASD, seperti magnetic tape atau pada DASD, seperti magnetic disk.

Beberapa alasan untuk menyimpan berkas sekuensial pada DASD:

- Pada umumnya komputer dihubungkan dengan sedikit tape drive, sehingga tidak cukup untuk menunjang program aplikasi yang banyak membutuhkan berkas sekuensial .

Contoh:

Jika 3 berkas sekuensial, seperti master file, transaction file, dan update master file yang digunakan oleh sebuah program, karena hanya ada 2 tape drive, maka salah satu dari ketiga berkas tersebut disimpan dalam disk.

- Sistem yang dikonfigurasi untuk fungsi berkas tertentu, selalu disimpan dalam disk .

Contoh:

Printer hanya dapat menerima semua berkas yang akan dicetak, bila terlebih dahulu berkas tersebut disimpan dalam disk. Jadi bila kita ingin membuat sebuah berkas laporan, maka harus ditentukan dari disk ke printer. Karakteristik lalu lintas saluran dan kapasitas saluran pada sistem dapat dibuat menguntungkan dengan cara memisahkan berkas-berkas dalam media penyimpanan .

Contoh:

Sebuah sistem akan dikonfigurasi dengan 2 tape drive pada satu saluran dan 2 disk drive pada saluran lain. Jika volume data besar yang dihasilkan oleh sebuah program dari 2 berkas sekuensial, maka akan menguntungkan bila berkas-berkas tersebut diletakkan pada saluran terpisah, dari pada diletakkan pada peralatan yang salurannya digunakan bersama-sama.

PEMBUATAN BERKAS SEKUENSIAL

Pembuatan berkas sekuensial meliputi penulisan record-record dalam serangkaian yang diinginkan pada media penyimpanan.

Pembuatan berkas transaksi sekuensial meliputi tugas tugas:

- Pengumpulan data.
- Perubahan data dalam bentuk bahasa yang dapat dibaca oleh mesin.
- Pengeditan data.
- Pemeriksaan transaksi yang ditolak.
- Penyortiran edit data.

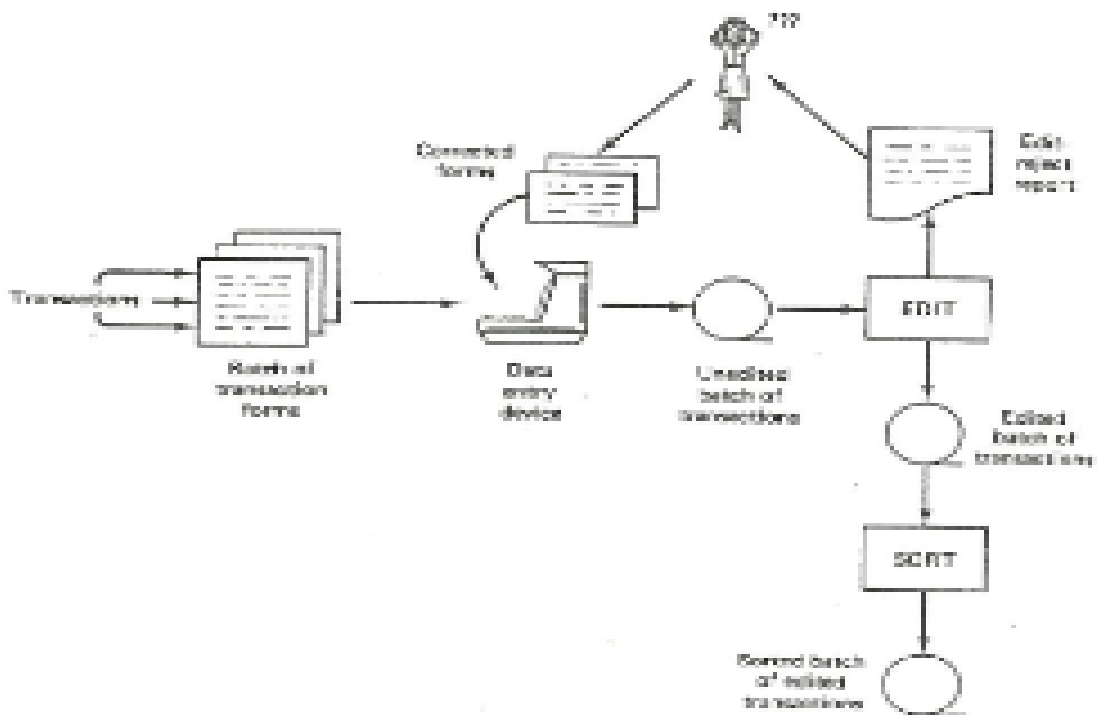


Figure 11-6. System flow for creating a sequential transaction file.

Gambar 5. Pembuatan Berkas Transaksi Sekuensial

PEMBUATAN BERKAS LAPORAN

Dalam Pembuatan berkas laporan sekuensial dikenal 3 jenis record:

- **Header Record**
Mencakup report header, page header, dan group header. Dikenal sebagai informasi pengenal (identifying information).
- **Detail Record**
Mencakup isi laporan yang umumnya disusun dalam kolom.
- **Footer Record**
Mencakup report footer, page footer, dan group footer. Dikenal sebagai informasi ringkasan (summary information).

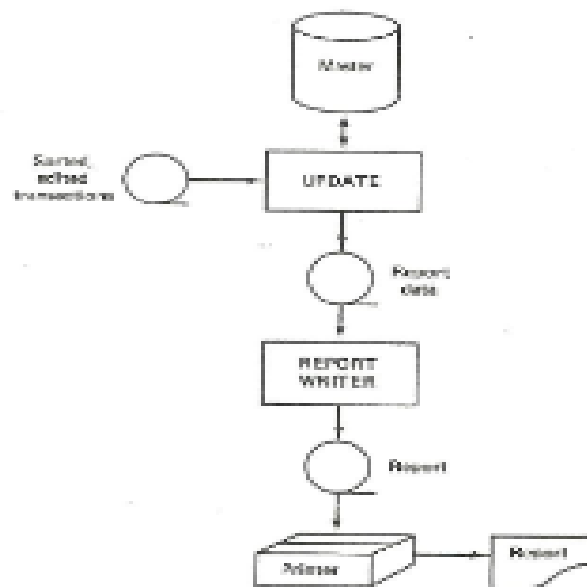


Figure 11-8 Creating a report file.

Gambar 6. Pembuatan Berkas Laporan

RIETRIEVAL TERHADAP BERKAS SEKUENSIAL

Record pada berkas sekuensial di *retrieve* secara berurutan. Urutan dimana record tersebut ditulis pada berkas, menentukan urutan dimana record tersebut didapat kembali.

Retrieve dari sebuah berkas sekuensial dibagi 2, yaitu: **report generation** dan **inquiry** yang bergantung pada jumlah data yang dihasilkan.

Pada umumnya bekas sekuensial diakses dalam model report generation, karena record-record harus diakses secara berurutan, tentunya lebih efisien mengakses setiap record dari berkas tersebut.

Inquiry dari berkas sekuensial mengalami hambatan, karena organisasi berkas ini memerlukan pengaksesan record secara satu persatu. Namun ada inquiry yang memerlukan pengaksesan semua record dari berkas

Contoh:

- Beberapa jumlah mahasiswa yang berumur di atas 20 tahun?
- Berapa jumlah pegawai yang mempunyai gaji di bawah Rp. 1.000.000.- ?

HIT RATIO

Banyaknya record yang harus diakses untuk mendapatkan informasi yang diinginkan dibagi dengan banyaknya record dalam berkas tersebut.

Contoh:

Inquiry NPM: 10109207 memerlukan pengaksesan record sebanyak 10 dari 100 record yang ada dalam berkas mahasiswa.

$$\text{Hit} = \frac{10}{100} = 0.1$$

- Semakin rendah hit ratio, semakin tidak baik bila menggunakan organisasi sekuensial.
- Semakin tinggi hit ratio, semakin baik bila menggunakan organisasi sekuensial.

UPDATE TERHADAP BERKAS SEKUENSIAL

Master file berisi data yang relatif tetap, tetapi kadang kadang kita perlu mengadakan perubahan pada berkas tersebut.

Hal ini kita sebut sebagai proses update.

Frekuensi dimana sebuah master file harus di *update* bergantung pada faktor faktor:

- Tingkat perubahan data.
- Ukuran dari master file.
- Kebutuhan yang mendesak dari data yang sedang berjalan pada master file.
- File Activity Ratio

FILE ACTIVITY RATIO

Banyaknya record pada master file yang di *update* dibagi dengan banyaknya record pada master file.

Contoh:

Transaction file

101	Bimo	75
102	Amalia	70
103	Seno	60

Master File

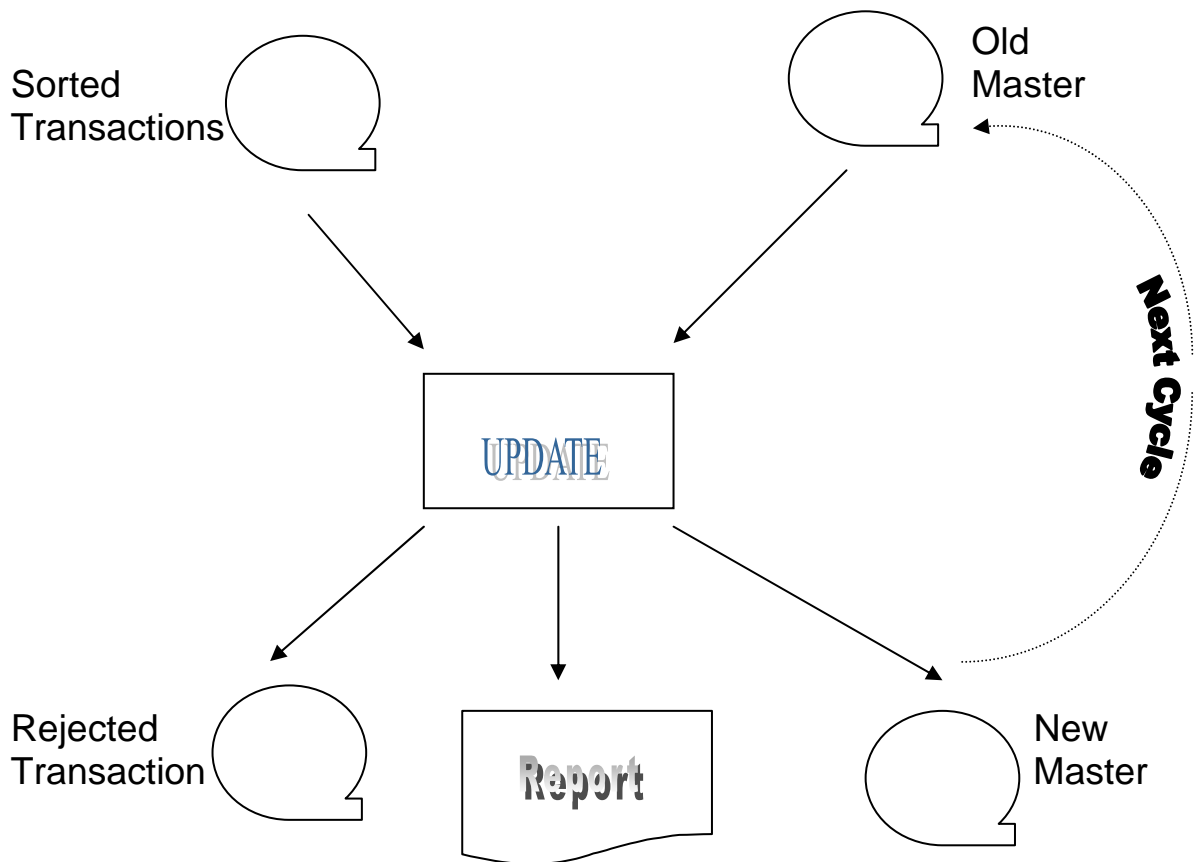
101	Bimo	jl.A	50
103	Seno	jl.C	30
104	Henni	jl.Z	50
105	Pandu	jl.D	70

$$\text{File Activity Ratio} = \frac{1+1}{4} = 0.5$$

- Semakin tinggi file activity ratio, semakin lama proses peng-*update*-an master file.
- Semakin tinggi kebutuhan akan data yang baru pada master file, maka semakin sering file tersebut diakses.
- Semakin sering master file di-*update*, semakin tinggi biaya pemrosesannya.

Kebanyakan berkas sequential tidak dapat di-*update* langsung ditempat, karena untuk meng-*update* biasanya diperlukan berkas baru sebagai pengganti berkas lama.

Pada gambar 7 menunjukkan *system flow diagram* untuk meng-*update* sebuah berkas sekuensial.



Gambar 7. System Flow Diagram untuk Update Berkas Sekuensial

GENERATION FILE

Selama next cycle pada proses update, new master file yang sekarang akan menjadi old master file. Menjadi banyaknya master file inilah yang disebut sebagai Generation File. File yang mempunyai nama yang sama, tetapi berbeda nomor generasinya. Jika old master sekarang merupakan generasi 1, maka new master berikutnya merupakan generasi 2, new master pada next cycle menjadi generasi 3, dst.

JENIS UPDATE

Ada 3 jenis update yang dapat dilakukan pada master file:

- Insert a new record
- Delete an existing record
- Modify an existing record

Menangani Kesalahan

Dalam pelaksanaan update dapat ditemukan beberapa kesalahan, seperti:

- Insert record that already exists
- Delete a record that does not exist
- Modify a record that does not exist

Contoh:

<u>Master File</u>	<u>Trans -Type</u>		
101	101	1	
102	103	2	1 : Delete
103	105	1	2 : Insert
104	107	3	3 : Modify
	101	2	

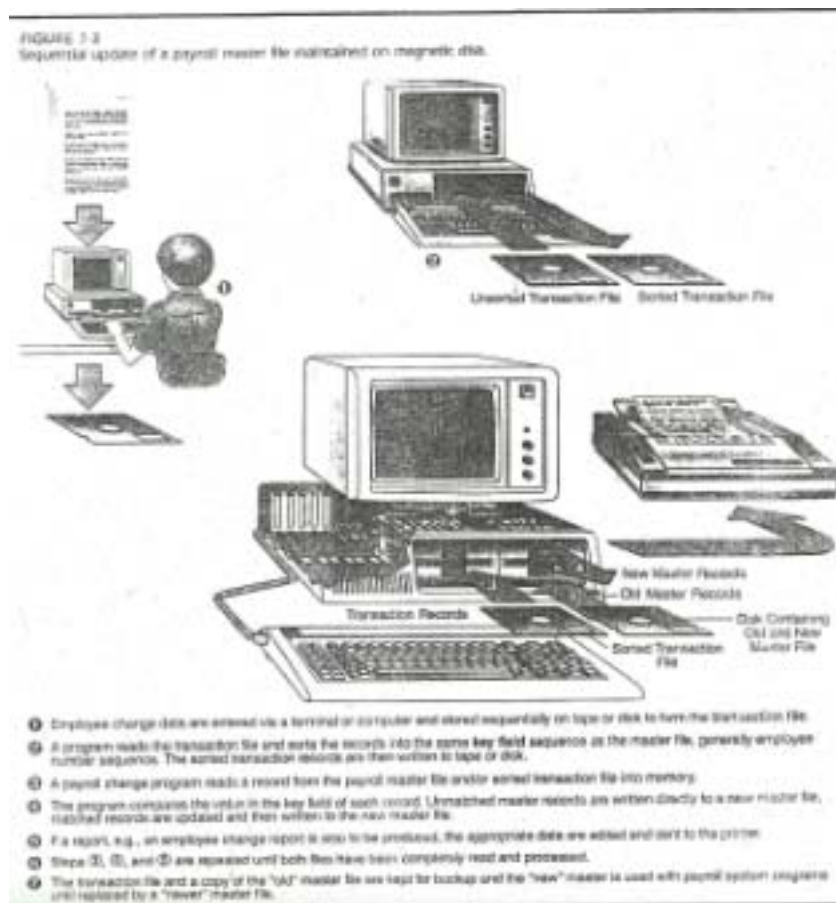
$$\text{File Activity Ratio} = \frac{1}{4}$$

Contoh:

Sebuah master file berisi 10 record. Transaksi yang akan diproses adalah sebagai berikut:

<u>Rec-Id</u>	<u>Trans - Type</u>	
111	2	
111	1	
96	3	
400	1	1 : Insert
96	1	2 : Delete
111	2	3 : Modify
400	3	
342	3	
96	2	

$$\text{File Activity Ratio} = \frac{4}{10}$$



Gambar 8. Update Berkas Sekuensial pada Payroll Master File

DEKLARASI BERKAS SEQUENTIAL DALAM BAHASA COBOL

```
SELECT [OPTIONAL] filename ASSIGN TO [implementor-name] ...
    [ RESERVE integer [AREAS] ]
    [ AREA ]
```

```
[ ORGANIZATION is SEQUENTIAL ]
[ ACCESS MODE is SEQUENTIAL ]
[ FILE STATUS is data-name ].
```

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

```
SELECT RPT-FILE ASSIGN TO OUTPUT.
SELECT TRANS-FILE ASSIGN TO TAPE1.
```

```
SELECT MSTR-FILE-IN ASSIGN TO TAPE3.
SELECT MSTR-FILE-OUT ASSIGN TO TAPE2.
```

I/O CONTROL.

```
MULTIPLE FILE TRANS-FILE POSITION 2.
```

.....