

BAB 3

ORGANISASI BERKAS SEQUENTIAL

Pengertian Berkas Sequential

Organisasi berkas sequential adalah merupakan cara yang paling dasar untuk mengorganisasikan kumpulan record-record dalam sebuah berkas. Dalam organisasi berkas sequential, 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.

Jadi dalam organisasi berkas sequential, bukan berarti bahwa record-record tersebut disimpan dalam urutan numerik. Jika kita ingin menambahkan suatu record pada berkas sequential, maka record tersebut akan terletak pada akhir berkas.

Organisasi berkas sequential dapat terdiri dari record-record yang berbeda jenis.

Contoh:

Pada gambar, dalam sistem penggajian terpadu (Intergrated personnel-payroll system) mempunyai sebuah berkas pegawai (Employee file) yang terdiri dari dua jenis record, yaitu:

Personnel Record

rec type	empno	name	address	marital status	sex	home log	mil code	educ	so

Payroll Record

rec type	empno	fund type	account	base salrate	% time	resp dept	date start	date change

Record-record pada berkas tersebut tidak memerlukan format dan ukuran yang sama. Pada contoh tersebut, berkas disortir berdasarkan : EMP-NO, REC-TYPE

Proses

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

Keuntungan dan Keterbatasan

Adapun keuntungan utama dari teknik organisasi berkas sequential adalah kemampuan untuk mengakses record berikutnya secara tepat.

Sedangkan keterbatasan dari organisasi berkas sequential adalah kita tidak dapat mengakses langsung pada record yang diinginkan.

Pola Akses

Pola Akses adalah penentuan akses berdasarkan field tertentu. Selama pola akses, berkas sequential 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 sequential berdasarkan urutan yang sesuai dengan pola aksesnya, jangan sebaliknya.

Contoh:

Berkas gaji yang disusun secara sequential 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 = 1502800

Dimana NIP tersebut belum tersortir.

Media Penyimpanan Berkas Sequential

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

Beberapa alasan untuk menyimpan berkas sequential 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 sequential, 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 :

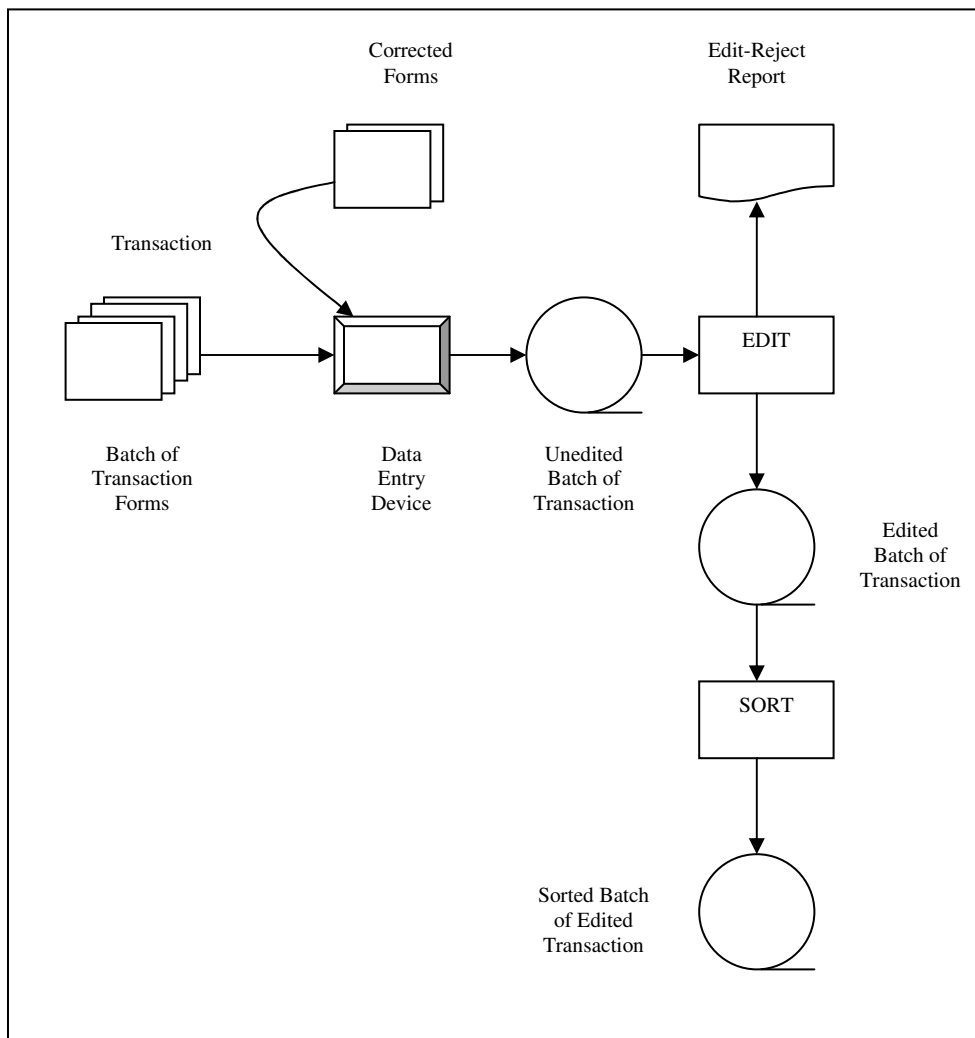
Sebuah sistem akan dikonfigurasi dengan 2 tape drive pada satu saluran dan 2 disk drive pada saluran lain. Jika volume data besar, yang dihasilkan oleh sebuah program dari 2 berkas sekuensial, maka akan menguntungkan bila berkas-berkas tersebut diletakkan pada saluran terpisah, daripada diletakkan pada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

Pembuatan Berkas Sequential

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

Pembuatan berkas transaksi sequential meliputi tugas-tugas:

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



Pembuatan Berkas Laporan

Dalam pembuatan berkas laporan sequential dikenal 3 jenis record :

1) **Header Record;**

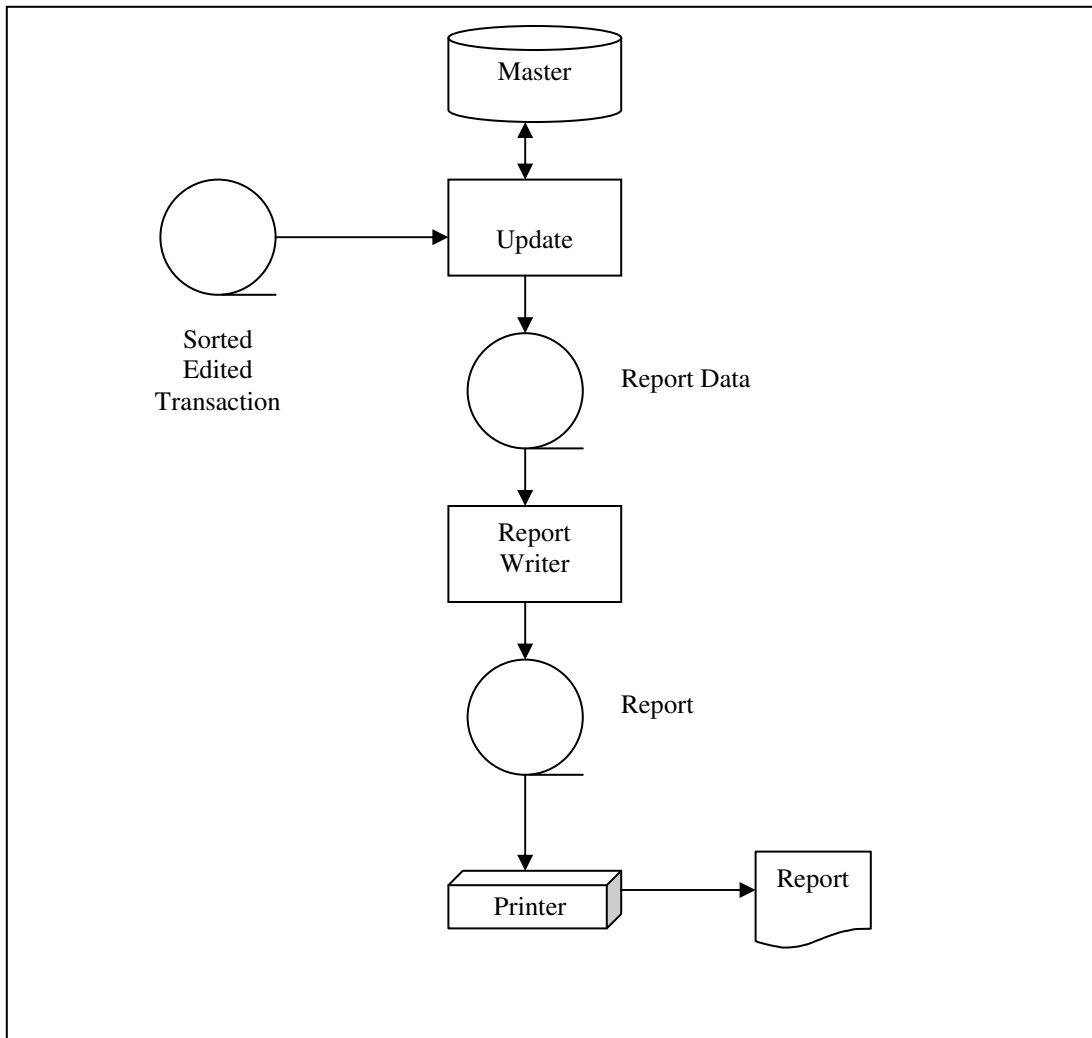
Mencakup report header, page header dan group header. Dikenal sebagai informasi pengenalan (Identifying Information).

2) **Detail Record;**

Mencakup isi laporan yang umumnya disusun dalam kolom.

3) **Footer Record;**

Mencakup report footer, page footer dan group footer. Dikenal sebagai informasi ringkasan (Summary Information).



Retrieval Terhadap Berkas Sequential

Record pada berkas sequential di retrieve secara berurutan. Urutan dimana record tersebut ditulis pada berkas menentukan urutan dimana record tersebut di dapat kembali.

Retrieve dari sebuah berkas dapat dibagi 2, yaitu : Report Generation dan Inquiry, yang bergantung pada jumlah data yang dihasilkan.

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

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

Contoh :

- Berapa 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 = 0029207 memerlukan pengaksesan record sebanyak 10 dari 100 record yang ada dalam berkas mahasiswa.

$$\text{Hit Ratio} = 10/100 = 0.1$$

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

Update Terhadap Berkas Sequential

Telah kita ketahui bahwa 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		
101	Bimo	75
102	Amalia	70
103	Seno	60

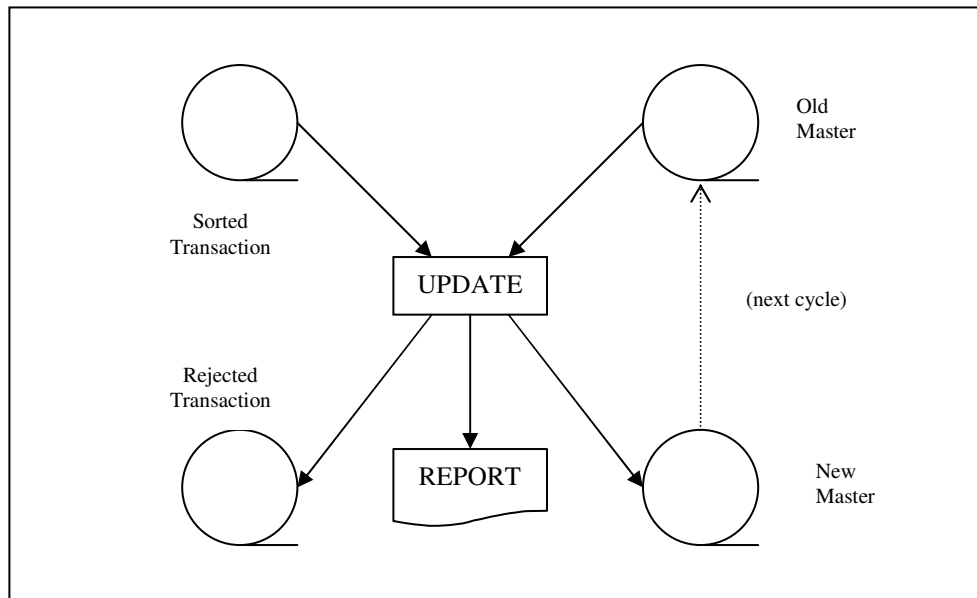
Master File			
101	Bimo	Jl. A	50
102	Seno	Jl. C	30
103	Henni	Jl. Z	50
104	Pandu	Jl. D	70

File Activity Ratio = $(1 + 1) / 4 = 0.5$

- 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 di tempat, karena untuk meng-update biasanya diperlukan berkas baru sebagai pengganti berkas lama.

Di bawah ini akan ditunjukkan gambar system flow diagram untuk meng-update sebuah berkas sequential.



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, dan seterusnya.

JENIS UPDATE

Ada 3 jenis update yang akan dapat dilaksanakan pada master file :

- 1) Insert a new record
- 2) Delete an existing record
- 3) Modify an existing record

MENANGANI KESALAHAN

Dalam pelaksanaan update, dapat ditemukan beberapa kesalahan seperti :

- (a) Insert a record that already exists
- (b) Delete a record that does not exist
- (c) Modify a record that does not exist

Contoh :

Master	Trans	Type	
101	101	1	
102	103	2	1 : Delete
103	105	1	2 : Insert
104	107	3	3 : Modify
	101	2	

File Activity Ratio = 1/4

Contoh :

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

Master	Trans	Type	
111		2	
111		1	1 : Delete
96		3	2 : Insert
400		1	3 : Modify
96		1	
111		2	
400		3	
342		3	
96		2	

File Activity Ratio = 4/10 = 0,4