

BAB 4. ORGANISASI FILE

I. FILE URUT

Organisasi file secara urut memungkinkan pengaksesan record di file secara berurutan.

1. Identification division.

Isi dan bentuk dari identification division untuk organisasi file urut sama dengan program yang tidak menggunakan file

2. Environment division.

Pada file urut yang berbeda adalah Input-Output Section pada File-Control paragraph karena informasi mengenai file urut disebutkan di sini.

Bentuk :

INPUT-OUTPUT SECTION.

FILE-CONTROL

SELECT nama_file ASSIGN TO { DISK
PRINTER }

[;ORGANIZATION IS [LINE] SEQUENTIAL]

[; ACCESS MODE IS SEQUENTIAL]

[; FILE STATUS IS nama_data].

- ASSIGN clause menunjukkan bentuk dari file yang akan dipergunakan, yaitu dapat berupa disk-file atau print-file.
- ORGANIZATION IS LINE SEQUENTIAL atau ORGANIZATION IS SEQUENTIAL boleh dipilih, bila tidak disebutkan dianggap default yaitu ORGANIZATION IS SEQUENTIAL, dan clause ACCESS MODE IS SEQUENTIAL boleh tidak disebutkan.
- FILE STATUS clause dipergunakan untuk menunjukan jenis kesalahan yang terjadi dari suatu hasil proses file. Nama-data yang disebutkan harus terlebih dahulu dideklarasikan dahulu pada working-storage section.
- Nilai status organisasi file urut :

Nilai Status	Jenis kesalahan
00	Tidak terjadi kesalahan
10	Akhir dari file (End of file)
30	File tidak ada(file not found)
34	Disk penuh (Disk space full)
91	Struktur file rusak

Contoh :

File-control

Select Rekaman Assign to disk

Organization is Sequential

Access Mode is sequential.

Select cetak assign to printer.

3. Data division

Pada file urut harus mengandung FD (file description) entry sbb :

FD nama-file

; LABEL { RECORD IS } { STANDARD }
 { RECORD ARE } { OMITTED }

[; VALUE OF FILE_ID IS nama_file di label]

[; DATA { RECORD IS } nama-record-1 [, nama-record-2]
 { RECORDS ARE }]

[; LINAGE IS { nama-data-1 } LINES [; WITH FOOTING AT { nama-data-2 }
 { integer-5 }] { integer-6 }]

[; LINES AT TOP { nama-data-3 }
 { integer-7 }]
 [; LINES AT BOTTOM { nama-data-4 }
 { integer-8 }]

- LABEL RECORD dan LABEL RECORDS Clause
 Menunjukkan apakah file yang digunakan mempunyai label atau tidak.
 LABEL RECORD IS OMITTED digunakan untuk card-file atau print-file yang tidak mempunyai label.
 LABEL RECORD IS STANDARD digunakan untuk disk-file yang mempunyai label.
- VALUE OF FILE-ID Clause
 Menunjukkan informasi mengenai file yang mempunyai label di disk. Label di disk ini berupa suatu nama-file yang berisi data yang direkamkan tersebut.
- DATA RECORD atau DATA RECORDS Clause
 Menunjukkan nama-record dalam file, sifatnya optional, hanya sebagai dokumentasi saja.
- LINAGE Clause
 Pencetakan output di printer dapat dilakukan dengan 2 cara, yaitu :
 - a. Dengan menganggap printer sebagai print-file, yang berbentuk organisasi file secara sequential, dengan menggunakan statement WRITE pada procedure division.
 - a. Dengan tanpa menggunakan print-file, tetapi langsung menghubungkan alat cetak printer sebagai nama-mnemonik pada paragraph SPECIAL-NAMES, yang kemudian menggunakan statement DISPLAY... UPON.

LINAGE Clause digunakan untuk mengatur pencetakan output diprinter yang mempergunakan cara pertama, yaitu sebagai print-file yang menggunakan statement WRITE, menunjukkan jumlah baris yang akan dicetak per halamannya, jumlah dari baris kosong paling atas (TOP MARGIN) dan jumlah dari baris kosong paling bawah (BOTTOM MARGIN).

LINAGE IS menunjukkan tubuh dari laporan, daerah laporan yang akan dicetak meliputi judul, isi dan footingnya.

WITH FOOTING AT menunjukkan daerah letak permukaan foot-note/footing (untuk pencetakan suatu total atau keterangan-keterangan untuk laporan).

LINES AT BOTTOM menunjukkan sejumlah baris yang tidak dipergunakan pada ujung bawah laporan.

Untuk lebih lengkap lihat bab 2 hal 6.

- Label record is Omitted atau label records are omitted digunakan untuk print-file yang tidak mempunyai label dan Value of File-id clause tidak dituliskan.
- Label record is Standard atau Label Records are standard digunakan untuk disk-file yang mempunyai label.

Contoh :

```
Data Division.
File-Section.
FD rekaman
    Label record is standard
    Value of file-Id is 'd:\nama.dat'
    Data record is record-relasi.
01 record-relasi.
    02 nama          pic A(25).
    02 alamat        pic X(25).
FD cetak
    Label record is omitted
    Linage          is 52 lines
                   With footing at 48
                   Lines at top 7
                   Lines at bottom 7
    Data record is baris-cetak.
01 baris-cetak     pic x (100).
```

Contoh ini menggunakan dua buah file dengan maksud data akan dibaca kemudian akan dicetak di print-file.

4. Procedure division.
Statement yang tersedia adalah:

a. OPEN verb

Digunakan untuk membuka file yang akan diakses.

Bentuk :

```
OPEN { INPUT      nama-file1 [, nama-file2],.... }
      { OUTPUT     nama-file3 [, nama-file4],.... }
      { I-O        nama-file5 [, nama-file6],.... }
      { EXTEND     nama-file7 [, nama-file8],.... }
```

- OPEN INPUT menunjukan file dibuka untuk tujuan sebagai input atau data akan dibaca dengan statemen READ.

- OPEN OUTPUT menunjukan file akan dibuka untuk direkam atau ditulisi dengan statemen WRITE. Jika ingin menambah data maka gunakan OPEN EXTEND.
- OPEN I-O menunjukan file dibuka sebagai file input dan output, yaitu file dibuka untuk dapat dibaca dengan statement READ kemudian memungkinkan untuk merekamkan kembali pada posisi yang sama dengan REWRITE.

Contoh : Open input rekaman
Output cetak.

b. CLOSE verb

Setelah operasi selesai, maka semua file harus ditutup

Bentuk :

CLOSE nama-file-1 [WITH LOCK] [, nama-file2 [WITH LOCK]]

LOCK dipergunakan bila file yang sudah ditutup tidak bisa dibuka kembali selama proses masih berlangsung.

Contoh : Close rekaman, cetak.

c. WRITE verb

Digunakan untuk merekamkan data ke dalam record di file yang telah dibuka dengan statemen OPEN pada disk-file atau mencetak di printer menggunakan print-file.

Bentuk :

WRITE nama-record [FROM nama-data-1]

$$\left(\left\{ \begin{array}{l} \text{BEFORE} \\ \text{AFTER} \end{array} \right\} \left\{ \begin{array}{l} \text{ADVANCING} \\ \text{nama-file2} \\ \text{integer} \\ \text{PAGES} \end{array} \right\} \left\{ \begin{array}{l} [\text{LINE}] \\ [\text{LINES}] \end{array} \right\} \right) \\ \left(;\text{AT} \left\{ \begin{array}{l} \text{END OF PAGE} \\ \text{EOF} \end{array} \right\} \text{statemen-imperatip} \right)$$

- WRITE harus diikuti nama-record dengan level number 01 yang dihubungkan dengan suatu file yang dibuka.
- BEFORE option digunakan untuk menunjukkan banyaknya spasi baris percetakan antara baris yang sedang dicetak dengan baris percetakan selanjutnya.
- AFTER option digunakan untuk menunjukkan banyaknya spasi baris antara baris yang sedang dicetak pada kertas di printer dengan baris percetakan sebelumnya.
- ADVANCING option digunakan untuk menunjukkan banyaknya spasi baris yang digunakan BEFORE atau AFTER yang ditunjukkan oleh nilai nama-data2 atau integer.
- PAGE option digunakan bila data akan dicetak sebelum atau sesudah printer menduduki halaman baru selanjutnya.
- EOF phrase menunjukan baris yang terakhir dalam satu halaman percetakan. Letak dari EOF ditunjukkan oleh WITH FOOTING AT pada LINAGR clause di paragraph FD dalam DATA DIVISION. Kalau digunakan phrase ini, maka daerah footnote harus diebutkan.

Contoh : Write baris-cetak from record-relasi.

d. READ verb

Digunakan untuk membaca record yang ada di file.

Bentuk :

```
READ nama-file record [INTO nana-data]
    [; AT END satatement-imperatip ]
```

Contoh : 1) READ rekaman AT END GO TO selesai.
2) READ rekaman INTO rec-out GO TO selesai.

e. REWRITE verb

Digunakan untuk merekam ulang record yang sudah pernah direkam. Record yang direkam ulang adalah record yang terakhir dibaca. Pada Statemen REWRITE, file harus dibuka sebagai file Input-Output dengan statemen OPEN I-O.

Bentuk : REWRITE nama-record [FROM nama-data]

Contoh : **Rewrite** record relasi.
Rewrite record-relasi from record relasi-out.

II. FILE INDEX

Pada file index dimungkinkan pembacaan data secara langsung pada reord yang dituju. Pada file index akan dicipitakan dua buah file, yaitu file data dan file kunci yang berisi control index yang berfungsi sebagai penunjuk dari posisi record tertentu.

1. Identification division.

Isi dan bentuk dari identification division untuk organisasi file urut sama dengan program yang tidak menggunakann file

2. Environment division.

Pada file urut yang berbeda adalah Input-Output Section pada File-Control paragraph karena informasi mengenai file urut disebutkan di sini.

Bentuk :

```
INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT nama_file ASSIGN TO { DISK
                             PRINTER }

[ ;ORGANIZATION IS INDEX ]
( [ ; ACCESS MODE IS { SEQUENTIAL
                       RANDOM
                       DYNAMIC } ] )
; RECORD KEY IS Nama-data1
[ ; FILE STATUS IS nama_data2 ]
```

- **Sequential** access mode, urutan record di file disajikan dalam keadaan urut dari kecil ke besar berdasarkan field kuncinya.
- **Random** access urutan dan record di file tidak urut. Tetapi pengaksesan record diatur oleh programmer berdasarkan nilai kunci.
- **Dynamic** access, programmer dapat menggunakan file tersebut baik secara sequensial atau secara random.
- **Record Key** is clause menunjukkan nama data yang dipergunakn sebagai field kunci yang sudah disebutkan terlebih dahulu pada file description entry. Nama data ini dapat berupa group data item atau elementary data item yang berbentuk alphanumeric.
- **File Status** is clause dipergunakan untuk menunjukkan jenis kesalahan dari suatu hasil operasi file.

Nilai status file index

Nilai Status	Jenis Kesalahan
00	Tidak terjadi kesalahan
10	End of file
21	Tidak bisa merekam pada sequential mode, file tidak urut
22	Nilai kunci tidak unik
23	Record tidak ketemu
24	Disk full
30	File tidak ada
91	Struktur file rusak

Contoh :

File-control

Select file-mhs Assign to disk

Organization is Indexed

Access Mode is Random

Record key is No-Mhs

File status is status-salah.

3. Data division.
Data divisi pada file index sama dengan pada file urut
4. Procedure division.
Statement yang tersedia adalah:

a. OPEN verb

Digunakan untuk membuka file yang akan diakses.

Bentuk :

OPEN { INPUT nama-file1 [, nama-file2],....
 OUTPUT nama-file3 [, nama-file4],....
 I-O nama-file5 [, nama-file6],.... }

Statement yang boleh pada file index

ACCESS MODE	STATEMENT	PHRASE PADA STATEMENT OPEN		
		INPUT	OUTPUT	I-O
SEQUENTIAL	CLOSE	Boleh	Boleh	Boleh
	READ	Boleh	Tidak	Boleh
	WRITE	Tidak	Boleh	Tidak
	REWRITE	Tidak	Tidak	Boleh
	START	Boleh	Tidak	Boleh
	DELETE	Tidak	Tidak	Boleh
RANDOM	CLOSE	Boleh	Boleh	Boleh
	READ	Boleh	Tidak	Boleh
	WRITE	Tidak	Boleh	Boleh
	REWRITE	Tidak	Tidak	Boleh
	START	Tidak	Tidak	Tidak
	DELETE	Tidak	Tidak	Boleh
DYNAMIC	CLOSE	Boleh	Boleh	Boleh
	READ	Boleh	Tidak	Boleh
	WRITE	Tidak	Boleh	Boleh
	REWRITE	Tidak	Tidak	Boleh
	START	Boleh	Tidak	Boleh
	DELETE	Tidak	Tidak	Boleh

b. CLOSE verb

Bentuk :

CLOSE nama-file-1 [WITH LOCK] [, nama-file2 [WITH LOCK]]

c. WRITE verb

Bentuk :

WRITE nama-record [FROM nama-data-1]
 [; INVALID KEY statement imperative]

- **Invalid key** menunjukkan data pada record key dari field kunci harus unik, kalau tidak data tidak akan direkam dan statement imperative akan dikerjakan bila disebutkan.

d. READ verb

Bentuk 1:

READ nama-file [NEXT] RECORD [INTO nama-data]
 [; AT END satatement-imperatip]

Bentuk2 :

READ nama-file RECORD [INTO nama-data-1]
 [; KEY IS nama-data-2]
 [; INVALID KEY statement-imperatif]

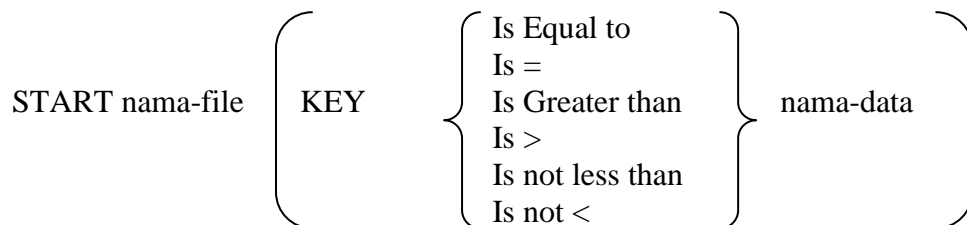
- **Next**, untuk sequential access mode bentuk1 digunakan tanpa menggunakan Next option dan bila menggunakan access mode Dynamic untuk pembacaan data secara urut maka NEXT option digunakan.
- Untuk access mode secara random atau dynamic dengan pembacaan data random, maka digunakan bentuk2.
- **Invalid key** menunjukkan data pada record key dari field kunci harus unik, kalau tidak data tidak akan direkam dan statement imperative akan dikerjakan bila disebutkan.
- **Into** phrase digunakan untuk memindahkan hasil record yang dibaca ke lokasi lain yang disebutkan oleh nama-data.

Contoh : 1) READ file-mhs atau
 READ file-mhs INTO record-mhs-out.
 2) READ file-mhs NEXT RECORD atau
 READ file-mhs NEXT RECORD INTO record-mhs-out
 3) READ file-mhs
 Key is No-mhs
 Invalid key move 'T' to hasil-cari.

e. START verb

Digunakan untuk menempatkan pembacaan record mulai pada record yang mengandung nilai field kunci tertentu. Statemen ini hanya boleh digunakan untuk file yang diakses secara sequential atau secara dynamic yang dibuka sebagai file urut. File dibuka dengan OPEN INPUT atau I-O.

Bentuk :



[; INVALID KEY statement imperative]

- Is equal to atau Is = , maka penempatan posisi akan berada pada record yang nilai kuncinya sama dengan nilai yang disebutkan.
- Is greater than atau Is > atau Is not less than atau Is not < , penempatan posisi akan berada pada record yang nilainya lebih besar dari nilai nama-data yang disebutkan
- Bila Key phrase tidak disebutkan, maka penempatan posisi akan berada pada record yang berisi nilai field kunci yang terakhir ada di storage.

Contoh : MOVE '850409' TO NIP
 START file-peg KEY IS` EQUAL NIP
 Invalid key move'T' to hasil-cari.

f. REWRITE verb

Untuk file sequential, sebelum record dirubah, record harus dibaca terlebih dahulu dengan statement read. Untuk file Random atau Dynamic, record yang akan dirubah ditentukan langsung dari field kuncinya.

Bentuk :

```
REWRITE nama-record [FROM nama-data ]  
[ ; INVALID KEY statemen imperative ]
```

g. DELETE verb

Digunakan untuk untuk menghapus record tertentu di file. Untuk file sequential, sebelum record dihapus, record harus dibaca terlebih dahulu dengan statement read. Untuk file Random atau Dynamic, record yang akan dihapus ditentukan langsung dari field kuncinya.

Bentuk :

```
DELETE nama-file RECORD[ ;INVALID KEY statemen imperative]
```

Contoh : MOVE '850409' TO Nip

```
DELETE File-peg RECORD
```

```
INVALID KEY DISPLAY ' Tidak ada data'.
```

Membandingkan Program Memasukkan Data

Contoh 1 : file sequential

Identification division.
Program-id. testFile.
Environment division.
Input-Output section.
File-control
 Select pegawai assign to disk
 Organization is sequential
 File status is status-salah.
Data division.
File section.
FD pegawai
 Label record is standard
 Value of file-id is 'c:\ peg.dat'
 Data record is rec-peg.
01 rec-peg.
 02 nama pic x(10).
 02 alamat pic x(20).
 02 nip pic 9(6).
Working-storage section.
77 status-salah pic xx.
01 sudah-benar pic x.
 88 benar value 'Y'.
 88 belum value 'T'.
01 input-lagi pic x.
 88 lagi value 'Y'.
 88 tidak value 'T'.
Screen section.
01 layar-data.
 02 Line 4 column 5 'Nama :'.
 02 column plus 2 pic x(10) to nama.
 02 Line 5 column 5 'Alamat :'.
 02 Column plus 2 pic x(20) to alamat.
 02 Line 6 column 5 'NIP :'.
 02 Column plus 2 pic z(6) To nip full.
Procedure division.
Buka-file.
 Open output pegawai.
Mulai.
 Move space to sudah-benar
 Perform input until benar
 Perform rekam-data
 Perform tambah-lagi.
 If lagi go to mulai.

(1)

Contoh 2 : file index

.....
File-control
 Select pegawai assign to disk
 Organization is index
 Access mode is sequential
 Record key is nip
 File status is status-salah.
Data division.
File section.
FD pegawai
 Label record is standard
 Value of file-id is 'c:\ peg.dat'
 Data record is rec-peg.
01 rec-peg.
 02 nama pic x(10).
 02 alamat pic x(20).
 02 nip pic 9(6).
Working-storage section.
01 rec-peg-o
 nama-o pic x(10).
 alamat-o pic x(20).
 nip-o pic 9(6).
77 status-salah pic xx.
03 sudah-benar pic x.
 88 benar value 'Y'.
 88 belum value 'T'.
01 input-lagi pic x.
 88 lagi value 'Y'.
 88 tidak value 'T'.
Screen section.
01 layar-data.
 02 Line 4 column 5 'Nama :'.
 02 Column plus 2 pic x(10) to nama-o.
 02 Line 5 column 5 'Alamat :'.
 02 Column plus 2 pic x(20) to alamat-o.
 02 Line 6 column 5 'NIP :'.
 02 Column plus 2 pic z(6) To nip-o full.
Procedure division.
Buka-file.
 Open output pegawai.
Mulai.
 Move space to sudah-benar
 Perform input until benar
 Perform rekam-data
 Perform tambah-lagi.
 If lagi go to mulai.

(a)

Selesai.
 Close pegawai
 Stop run.
 Input-data.
 Display layar-data
 Accept layar data.
 Display (12, 5) 'Sudah benar (Y/T) ? '
 Accept (,) sudah-benar.
 Rekam-data.
 Move space to status-salah.
Write rec-peg.
 Perform Tanya-rekam.
 Tanya-rekam.
 If status-salah = '34'
 display (12,5) 'disk full'.
 Tambah-lagi.
 Display (12, 5)
 'Masukan data lagi (Y/T) ? '
 Accept (,) input-lagi.

(2)

Selesai.
 Close pegawai
 Stop run.
 Input-data.
 Display layar-data
 Accept layar data.
 Display (10, 5) 'Sudah benar (Y/T) ? '
 Accept (,) sudah-benar.
 Rekam-data.
 Move space to status-salah.
Write rec-peg from rec-peg-o
 Perform Tanya-rekam.
 Tanya-rekam.
 If status-salah = '21'
 display (12,5) 'data tidak urut'.
 If status-salah = '24'
 Display (12, 5) 'diskfull'.
 Tambah-lagi.
 Display (15, 5)
 'Masukan data lagi (Y/T) ? '
 Accept (,) input-lagi.

(b)

Mencetak semua data ke printer Contoh 3 : file sequential

.....
 File-control
 Select pegawai assign to disk
 Organization is sequential
 Access mode is sequential.
 Select cetak assign to printer.
 Data division.
 File section.
 FD pegawai
{sama dengan contoh 1}.....
 FD cetak
 Label record is omitted
 Data record is baris-cetak.
 01 baris-cetak pic x(100).
 Working-storage section.
 01 baca pic x.
 88 habis value 'Y'.
 01 judul.
 02 filler pic x(10) value spaces.
 02 filler pic x(20) value
 'DAFTAR PEGAWAI'.
 01 judul2.
 02 filler pic x(13) value 'NAMA'.
 02 filler pic x(23) value 'ALAMAT'.

(1)

02 filler pic x(8) value 'NIP'.
 01 garis.
 02 Filler pic x(65) value all'-'.
 Procedure division.
 Buka-file.
 Open input pegawai
 Output cetak.
 Mulai.
 Perform cetak judul
 Perform cetak-isi until habis-dibaca
 Perform cetak garis.
 Selesai.
 Close pegawai,cetak.
 Stop run.
 Cetak judul.
 Write baris-cetak from judul after
 advancing page.
 Write baris-cetak from garis after
 advancing 2 lines.
 Write baris-cetak from judul2
 Write baris-cetak from garis.
 Write baris-cetak from rec-peg.

(2)

Cetak-isi.
 Read pegawai
 At end move 'Y' to baca
 Cetak-garis.
 Write baris-cetak from garis.
 (3)

Mencari Data Tertentu dengan File Sequential dan Index

Contoh 4 : file sequential

.....
 File-control
 Select pegawai assign to disk
 Organization is sequential
 File status is status-salah.
 Data division.
 File section.
 FD pegawai
 ... {sama dengan contoh 1}....
 Working-storage section.
 77 nip-cari pic 9(6).
 01 hasil-cari pic x.
 88 ketemu value 'Y'.
 88 tidak value 'T'.
 01 baca pic x.
 88 habis value 'Y'.
 01 cari pic x.
 88 lagi value 'Y'.
 88 tdk value 'T'.
 Screen section.
 01 layar-hasil.
 02 line 4 column 5 'Nama :'.
 02 Column plus 2 pic x(10) from
 nama.
 02 Line 5 column 5 'Alamat :'.
 02 Column plus 2 pic x(20) from
 alamat.
 02 Line 6 column 5 'NIP :'.
 02 Column plus 2 pic z(6) from nip.

Procedure division.
 Buka-file.
 Open input pegawai.
 Mulai.
 Display 'Masukan nip yang dicari :'
 Accept nip-cari.
 Perform cari-data until habis or ketemu
If ketemu
Display layar-hasil
 Else
 Display (4, 5) 'data tidak ada'.
 Perform Ada-lagi.
 If lagi
 go to buka-file.
 Selesai.
 Close pegawai
 Stop run.
Cari-data.
Read pegawai at end move 'Y' to baca
If nip = nip-cari
Move 'Y' to hasil-cari.
 Ada-lagi.
 Display (8,5) 'cari lagi (Y/T) ?'
 Accept (,) cari with beep.

1)

(2)

Contoh 5 : file index, mode sequential

.....
File-control
 Select pegawai assign to disk
 Organization is index
 Access mode is sequential
 Record key is nip.
Data division.
..... {sama dengan contoh 4}.....
Procedure division.
Buka-file.
 Open input pegawai.
Mulai.
 Display 'Masukan nip yang dicari : '
 Accept nip-cari.
 Perform Cari-data
 If ketemu
 Perform Baca-data
 Display layar-hasil
 Else
 Display (4, 5) 'data tidak ada'.
 Perform Ada-lagi.
 If lagi
 go to Buka-file.
Selesai.
..... {sama dengan contoh 4}.....

Cari-data.
 Move 'Y' to hasil-cari
 Move nip-cari to nip
 Start file-mhs
 Key is equal to nip
 Invalid key move't to hasil-cari.
Baca-data.
 Read file-mhs.
Ada-lagi.
..... {sama dengan contoh 4}.....

Contoh 6 : file index, mode random

.....
File-control
 Select pegawai assign to disk
 Organization is index
 Access mode is random
 Record key is nip.
Data division.
..... {sama dengan contoh 4}.....
Procedure division.
Buka-file.
 Open input pegawai.
Mulai.
 Display 'Masukan nip yang dicari : '
 Accept nip-cari.
 Perform Cari-data
 If ketemu
 Display layar-hasil
 Else
 Display (4, 5) 'data tidak ada'.
 Perform Ada-lagi.
 If lagi
 go to Buka-file.
Selesai.
..... {sama dengan contoh 4}.....

Cari-data.
 Move 'Y' to hasil-cari
 Move nip-cari to nip
 Read file-mhs
 Key is nip
 Invalid key move't' to hasil-cari.

Ada-lagi.
..... {sama dengan contoh4}.....