

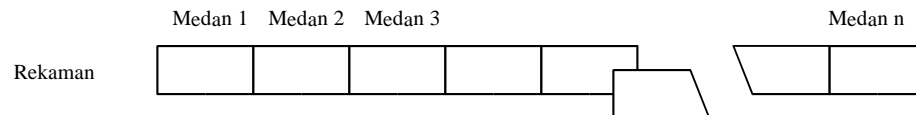
SISTEM BERKAS

Oleh : La Ode Abdul Jumar

1. Berkas

Berkas(file) adalah kumpulan sejumlah komponen yang bertipe data sama, yang jumlahnya tidak tertentu. Banyaknya data dalam berkas dapat ditambah jika diperlukan. Dalam pascal berkas menyediakan data yang nantinya akan digunakan oleh suatu program. Berkas dapat berupa *disk file* yang media penyimpanannya berupa cakram magnetis, kartu plong, dan sejenisnya atau berupa piranti logika yang sering digunakan.

Satu aspek penting dari berkas adalah bahwa data yang ada didalam berkas bisa digunakan oleh sembarang program yang tipe datanya disesuaikan dengan kebutuhan. Berkas tersusun dari sejumlah rekaman dimana masing-masing rekaman tersusun atas sejumlah medan. Hubungan antara medan, rekaman dan berkas :



Dalam pascal, berkas mempunyai sifat sebagai berkas berurutan (*sequential file*) dimana untuk membaca file ke 10 maka pembacaan dilakukan dari nomor ke 1, 2,3 dan seterusnya. Dalam pascal juga ada fasilitas untuk pembacaan berkas secara acak(*random file*).

MENDEKLARASIKAN BERKAS

Deklarasi berkas bisa ditempatkan pada bagian deklarasi tipe atau pada deklarasi var. Deklarasi berkas bisa dipastikan diwali dengan deklarasi rekaman, bentuk umum deklarasi rekaman :

Type *Pengenal* = **File of** *Pegenall*;

Jika ditempatkan dalam deklarasi Var, maka bentuknya adalah :

Var *Pengenal* : **File of** *Pegenall*;

Dengan **File of** : Kata baku yang menunjukkan bahwa *pengenal* adalah bertipe data berkas (**File**);

Pengenal : Nama peubah yang akan dinyatakan sebagai bertipe data **File**

Pegenall : Tipe data Komponen berkas.

Parameter *Pegenall* yang menunjukkan tipe data komponen berkas disebut tipe dasar (*Base type*) dari berkas.

OPERASI PADA BERKAS

Statemen-statemen dasar yang perlu dihafalkan untuk bisa bekerja dengan berkas. Adapun statemen-statemen tersebut adalah :

1. Statemen-statemen Dasar

1.1 Statemen Assign

Statemen ini digunakan untuk memilih berkas yang akan dioperasikan sesuai dengan tipe berkas yang kita deklarasikan, dengan kata lain statemen ini digunakan untuk membuka suatu berkas.

Bentuk umum statemen assign :

Assign(bksvar,nmbks);

Dengan bksvar : Pengenal yang di deklarasikan dalam variabel

Nmbks : Nama berkas yang tersimpan dalam media penyimpanan

1.2 Statemen Close

Statemen close digunakan untuk menutup berkas. Bentuk umu dari statemen ini adalah :

Close(bksvar)

1.3 Statemen Rewrite

Statemen ini digunakan untuk membangun berkas baru seperti dinyatakan dalam parameter bksvar pada saat membuka sebuah berkas baru. statemen ini sekaligus menyiapkan berkas untuk diproses dan penuding berkas diset diawal berkas yaitu pada posisi 0, jika berkas sebelumnya sudah terisi dengan data maka data yang ada akan terhapus.

Bentuk umum dari statemen rewrite adalah :

Rewrite(bksvar);

1.4 Statemen Reset

Statemen ini digunakan untuk menggerakkan penuding rekaman pada awal berkas, yaitu pada komponen nomor 0. bentuk umu dari statemen reset adalah :

Reset(bksvar);

1.5 Statemen Write

Statemen ini digunakan untuk menuliskan komponen berkas kedalam media penyimpanan. Bentuk umu dari statemen write adalah :

Write(bksvar,komvar);

Dengan bksvar : Pengenal yang di deklarasikan dalam variabel

Komvar : Pengenal yang menunjukkan nama peubah/variabel dari komponen berkas

1.6 Statemen Read

Statemen ini digunakan untuk membaca komponen berkas dari media penyimpanan. Bentuk umum dari statemen read adalah :

Read (bksvar,komvar);

2. Statemen dan Fungsi Tambahan

2.1 Statemen Seek

Statemen seek digunakan untuk menggerakkan penuding kesuatu komponen tertentu. Bentuk umum dari statemen seek adalah :

Seek(bksvar,nokom);

Dengan bksvar : Pengenal yang di deklarasikan dalam variabel

Nokom : variabel yang menunjukkan komponen yang akan dituju

2.2 Fungsi FilePos

Fungsi filepos ini digunakan untuk mengetahui dimana penuding berada. Bentuk umum dari fungsi filepos adalah :

Filepos(bksvar);

2.3 Fungsi FileSize

Fungsi filesize ini digunakan untuk mengetahui cacah komponen dari suatu berkas. Bentuk umum dari fungsi filesize adalah :

Filesize(bksvar);

2.4 Fungsi EOF

Fungsi EOF digunakan untuk mengetahui apakah penuding berada pada posisi akhir berkas(*end of file =eof*) atau tidak. Jika ya maka fungsi ini akan bernilai true, jika tidak maka fungsi ini akan bernilai false.

Bentuk umum dari fungsi EOF adalah :

EOF(bksvar);

2.5 Statemen Flush

Statemen ini digunakan untuk mengosongkan penyangga (buffer) dari berkas bksvar, dengan demikian akan memastikan bahwa isi penyangga telah dituliskan dalam cakram setelah suatu statemen write selesai dikerjakan. Bentuk umum dari statemen flush adalah :

Flush(bksvar);

2.6 Statemen Erase

Statemen erase digunakan untuk menghapus berkas yang dinyatakan dalam perubah bksvar. Bentuk umum dari statemen erase adalah :

Erase(**bksvar**);

2.7 Statemen Rename

Stemen rename digunakan untuk memberi nama baru pada berkas yang dinyatakan oleh bksvar. Bentuk umum dari statemen rename adalah :

Rename(bksvar,nm_baru);

Dengan bksvar : Pengenal yang di deklarasikan dalam variabel

Nm_baru : Nama baru yang akan diberikan sebagai pengganti nama lama.

Contoh Program dengan menggunakan Pascal:

PROGRAM SISTEM_BERKAS;

```
uses winCRT;
```

```
const Garis = '-----';
```

```
TYPE JUMLAH = 1..100;
```

```
daftar = RECORD
```

```
    Nama_Mhs : string[25];
```

```
    No_Mhs : string[8];
```

```
    Alamat : string[50];
```

```
end;
```

```
D_Nilai = file of daftar;
```

```
var Daf_MHS : D_Nilai;
```

```
    Nilai_Mhs : daftar;
```

```
    I,J,N : integer;
```

```
    lagi : char;
```

```
    key : string[8];
```

```
    POSISIX, POSISIY: INTEGER;
```

```
{#=====#}
```

```
{# Fungsi Pencarian#}
```

```
{#=====#}
```

```
function cari(key : string): longint;
```

```
var ketemu : boolean;
```

```
begin
```

```
    CLRSCR;
```

```
    {buka file}
```

```
    assign(Daf_MHS,'BERKAS.DAT');
```

```
    {$I- }
```

```

reset(Daf_MHS);
{$I+}
if IOresult <> 0 then
rewrite(Daf_MHS);
writeln;
write(' Masukkan nim yang ingin dicari : ');
read(key);
ketemu := false;
cari := -1;
seek(Daf_MHS, 0);
writeln(' Data yang anda cari : ');
while not eof(Daf_MHS) and (ketemu=false) do
begin

```

```

    read (Daf_MHS,Nilai_Mhs);
    if key = Nilai_Mhs.No_Mhs then
        begin
            ketemu := true;
            writeln;
            writeln(' nim    : ', Nilai_Mhs.No_Mhs);
            writeln(' nama   : ', Nilai_Mhs>Nama_Mhs);
            writeln(' alamat : ', Nilai_Mhs.Alat);
            writeln;
            write('PRESS ENTER TO CONTINUE...');
            readln;
            end
        Else
            if key <> Nilai_Mhs.No_Mhs then
                writeln(' tidak terdaftar ');
                writeln;write;
                readln;
            end;
        end;
end;

```

```

    {#=====#}
    {# Prosedur untuk menulis ke file#}
    {#=====#}

```

PROCEDURE tulis;

BEGIN

CLRSCR;

assign(Daf_MHS,'BERKAS.DAT');

```

rewrite(Daf_MHS);
I := 0;
repeat
  inc(I);
  clrscr;
  writeln('MENGENAL DATA MAHASISWA');
  writeln('=====');
  writeln;
  writeln('MAHASISWA KE-',I:2);
  writeln('=====');
  with Nilai_Mhs do
    begin
      write('Nama Mahasiswa : ');readln>Nama_Mhs);
      write('Nomor Mahasiswa : ');readln>No_Mhs);
      write('Alamat      : ');readln(Alamat);
    end;
  write(Daf_MHS,Nilai_Mhs);
  gotoxy(1,10);
  write('Akan mengisi lagi ? <Y>a/<T>idak: ');
  readln(lagi);
until not(lagi in['y','Y']);
close(Daf_MHS);
WRITELN;

```

END;

```

{#=====#}
{# Prosedur untuk menampilkan file#}
{#=====#}

```

PROCEDURE baca;

```

var Daf_MHS : D_Nilai;
    Nilai_Mhs : daftar;
    I,J,N    : integer;
    lagi     : char;
BEGIN
CLRSCR;
assign(Daf_MHS,'BERKAS.DAT');
reset(Daf_MHS);
N := filesize(Daf_MHS);
writeln(":28,'DATA MAHASISWA');
writeln(":28,'-----');

```

```

writeln;
writeln(Garis,Garis);
write('| No | Nama Mahasiswa | No Mhs ');
writeln(' | Alamat ');
writeln(Garis,Garis);
for I := 1 to N do
  begin
    gotoxy(1,I+6); write('| ',I:2,' | ');
    read(Daf_MHS,Nilai_mhs);
    with Nilai_Mhs do
      begin
        gotoxy(8,I+6);write>Nama_Mhs);
        gotoxy(26,I+6);write('| ',No_Mhs);
        gotoxy(39,I+6);writeln(' |',Alamat);
      end;
    end;
  writeln(Garis,Garis);
  close(Daf_MHS);
  writeln;
  write('PRESS ENTER TO CONTINUE...');
  readln;
END;

```

```

{#=====#}
{# Prosedur menu#}
{#=====#}

```

```

PROCEDURE menu;
var pil : integer;
begin
  clrscr;
  FOR POSISIY := 1 TO 20 DO
    BEGIN
      GOTOXY(1,POSISIY);
      WRITELN('#');
    END;
  FOR POSISIX := 1 TO 32 DO
    BEGIN
      GOTOXY(POSISIX,20);
      WRITELN('#');
    END;
  FOR POSISIY := 1 TO 30 DO
    BEGIN

```

```

        GOTOXY(32,20-POSISIY);
        WRITELN('#');
    END;
    FOR POSISIX := 1 TO 30 DO
        BEGIN
            GOTOXY(32-POSISIX,1);
            WRITELN('#');
        END;
    GOTOXY(3,3);
    WRITELN(' PROGRAM DATA MAHASISWA');
    GOTOXY(3,4);
    WRITELN(' +=+=+=+=+=+=+=+=+=+=+=');
    GOTOXY(3,5);
    WRITELN;
    GOTOXY(3,6);
    WRITELN(' *****');
    GOTOXY(3,7);
    writeln(' | MENU      |');
    GOTOXY(3,8);
    writeln(' | 1. Masukan Data |');
    GOTOXY(3,9);
    writeln(' | 2. Tampilkan Data |');
    GOTOXY(3,10);
    writeln(' | 3. cari Data     |');
    GOTOXY(3,11);
    writeln(' | 4. Keluar        |');
    GOTOXY(3,12);
    WRITELN(' *****');
    GOTOXY(3,13);
    WRITELN;
    repeat
    GOTOXY(3,14);
    write(' Masukan Piliahan anda : ');
    readln(pil);
    until (pil= 1) or (pil= 2) or (pil =3)or (pil=4);
    case pil of
    1 :
        begin
            tulis;
            menu;
        end;
    2 :
        begin

```



```

    baca;
    menu;
    end;
3 :
    begin
    cari(key);
    menu;
    end;
4 :
    begin
    CLRSCR;
    WRITELN;
    BEGIN
    FOR POSISIY := 1 TO 20 DO
        BEGIN
        GOTOXY(1,POSISIY);
        WRITELN('#');
        END;
    FOR POSISIX := 1 TO 75 DO
        BEGIN
        GOTOXY(POSISIX,20);
        WRITELN('#');
        END;
    FOR POSISIY := 1 TO 19 DO
        BEGIN
        GOTOXY(75,20-POSISIY);
        WRITELN('#');
        END;
    FOR POSISIX := 1 TO 74 DO
        BEGIN
        GOTOXY(75-POSISIX,1);
        WRITELN('#');
        END;
    GOTOXY(12,10);
    WRITE('LAODE ABDUL JUMAR');
    WRITELN(' (ABDULLAH UMAR AL-MUNAWY)');
    GOTOXY(14,13);
    WRITELN('BERUSAHALAH MENJADI YANG TERBAIK KAWAN');
    GOTOXY(28,6);
    WRITELN('SALAM DARI SAYA');
    GOTOXY(28,7);
    WRITELN('=====');
    GOTOXY(65,18);

```

```
WRITELN('JUMAR');  
GOTOXY(1,22);  
END;  
end;  
end;  
end;
```

```
{#=====#}  
{# program utama#}  
{#=====#}
```

```
BEGIN  
menu;  
end.
```

Daftar Pustaka

Bahan Perkuliahan Ilmu Komputer