

Membuat Berkas File Text Pada Pascal taryanarx@yahoo.com

Target Pencapaian

Siswa dapat membuat File bertipe Text, memasukan data dan menampilkan data dari file text tersebut

FILE (BERKAS)

File adalah kumpulan byte-byte yang disimpan dalam media penyimpanan. Merupakan komponen yang bertipe data sama, yang jumlahnya tidak tentu, yang dapat ditambah atau dikurangi jika dibutuhkan.

Pointer file adalah bagian yang menunjuk ke komponen file yang akan diakses (dibaca atau direkam) untuk keperluan pengaksesan file (akan dibahas kemudian).

File pada Pascal dikenal dalam 3 jenis, yaitu :

1. File Text
2. File bertipe
3. File tidak bertipe

Selain kita akan mempelajari tentang bagaimana membuat sebuah file atau menambahkan isi suatu file, kita dapat pula melakukan manipulasi File, yaitu :

1. Menggunakan parameter
2. Mengenai Atribut File
3. Menghapus file
4. Mengubah nama file
5. Mengenai tanggal dan waktu file
6. Mencari file
7. Mengecek keberadaan file
8. Memberikan directory file

1. File Text

1.1. Karakteristik

- Berisi data karakter ASCII
- Tiap record boleh memiliki panjang yang bervariasi
- Setiap record diakhiri tanda EOL (end of Line)
- Hanya dapat diakses secara sequensial (berurutan).
- Isi filenya dapat dilihat oleh perintah dos type atau editor text

1.2. Membuat file text

Urutan Prosesnya :

1. Mendeklarasikan variabel file

`Var NmVar:TEXT;`

Dengan :

NmVar : Nama variabel file text

2. Menghubungkan variabel file dengan nama file

`Assign (NmVar, nama file);`

Dengan :

NmVar : Nama variabel file text

nama file : Nama file dalam bentuk string, format 8:3 penamaan dos, ditulis dalam bentuk string.

3. Membuat file text aktif

`Rewrite(NmVar);`

Dengan :

NmVar : Nama variabel file text yang sudah di- assign

4. Menulis ke dalam file text

`Write / writeln (NmVar, data item1, data item 2, ...)`

Dengan :

NmVar : Nama variabel file text

Data item : text / string yang akan dituliskan, atau bisa juga berupa isi suatu variabel

5. Menutup file

`Close (NmVar);`

Dengan:

NmVar : Nama variabel file text

Contoh :

```
Program membuat_file_text_namafile_HALLO_TXT;
Uses crt;
Var var_teks:TEXT;
Begin
    Clrscr;
    Assign(var_teks,'Hallo.txt');
    Rewrite(var_teks);
    Writeln(var_teks,'Hallo... ini program pertamaku!');
    Writeln(var_teks,'Contoh file teks');
    Writeln(var_teks,'-----');
    Close(var_teks);
End.
```

1.3. Membaca File Text

Urutan Prosesnya :

1. Mendeklarasikan variabel file

```
Var NmVar:TEXT;
```

2. Menghubungkan variabel file dengan nama file

```
Assign (NmVar, nama file)
```

3. Membaca isi file dan menampilkannya di layar

```
While not eof (NmVar) do  
Begin  
Read / readln ( NmVar, data item 1, data item 2, ... );  
Write / writeln ( data item1, data item 2, ... );  
End;
```

4. **Menutup file**

```
Close (NmVar);
```

Contoh :

```
Program membaca_file_HALLO_TXT;  
Uses crt;  
Var var_teks:TEXT;  
    Pesan:string;  
Begin  
    Clrscr;  
    Assign(var_teks,'Hallo.txt');  
    Reset(var_teks);  
    While not eof (var_teks) do  
        Begin  
            Readln ( var_teks, pesan );  
            Writeln (pesan);  
        End;  
    Close(var_teks);  
End.
```

1.4. Menambah isi File Text

Urutan Prosesnya :

1. Mendeklarasikan variabel file

```
Var NmVar:TEXT;
```

2. Menghubungkan variabel file dengan nama file

```
Assign (NmVar, nama file)
```

3. Menambah isi file

```
Append(NmVar)
```

4. **Menampilkannya di layar**

```
Write / writeln (NmVar, data item1, data item 2, ...)
```

5. **Menutup file**

```
Close (NmVar);
```

Contoh :

```
Program menambah_isi_file_HALLO_TXT;
Uses crt;
Var var_teks:TEXT;
    Pesan:string;
Begin
    Clrscr;
    Assign(var_teks,'Hallo.txt');
    Append(var_teks);
    Writeln(var_teks,'Ini penambahan data file hallo.txt!');
    Writeln(var_teks,'Contoh file teks');
    Writeln(var_teks,'-----');
    Close(var_teks);
End.
```

1.5. Metode Pembacaan file oleh Turbo Pascal

- ❑ Menggunakan statemen operasi READ / READLN
- ❑ Bergantung pada tipe data variabel yang digunakannya :
 - ◆ Variabel berjenis numerik (byte, integer, real)
 - Tanda batas akhir pembacaan variabel jika ditemui blank(spasi), EOL (end of line), EOF (End of File) atau TAB.
 - Jika sebelum membaca data (atau dengan kata lain tidak ada datanya) pointer pascal menemukan EOL / EOF, maka variabel tersebut akan diisi NOL.
 - Jika string yang dibaca oleh variabel numerik tidak sah, maka terjadi kesalahan I/O : *Invalid numerik format.*
 - ◆ Variabel berjenis string
 - Karakter-karakter file akan dibaca sampai ditemui EOL / EOF tercapai atau lebih besar dari ukuran stringnya.
 - ◆ Variabel berjenis Char
 - Yang dibaca hanya satu karakter saja.
- ❑ Agar pembacaan file sesuai dengan yang diharapkan, kadangkala perlu kita deklarasikan secara jelas jumlah byte yang disediakan untuk suatu variabel, atau mesti kita selipkan pencetakan spasi diantara dua variabel yang akan dibaca.
- ❑ Perlu diperhatikan kesesuaian tipe data, antara yang dituliskan dengan metode WRITE/Writeln dan dengan yang akan kita baca.

CONTOH: BERKAS FILE

Membuat File Text

```
Uses Crt;
Var
  Berkas : Text;
  Kode : String[5];
  Nama : String[20];
  Harga : Longint;
  Jumlah: Longint;
  Lagi,Jawab : Char;
  Ada : Boolean;
Begin
  Assign(Berkas,'BARANG.TXT');
  {$I-}
  Reset(Berkas);
  {$I+}
  Append(Berkas);
  Lagi := 'Y';
  While upcase(lagi)='Y' do
  begin
    Clrscr;
    Write('kode barang :');Readln(Kode);
    Write('Nama      :');Readln(Nama);
    Write('Harga      :');Readln(Harga);
    Write('Jumlah      :');Readln(jumlah);
    Writeln(Berkas,kode:5,nama:20,harga:10,jumlah:10);
    Writeln;
    Write(' Lagi Y/T : ');Readln(lagi);
  End;
  Close(berkas);
End.
```

Program Untuk Menampilkan Data;

```
Uses Crt;
Var
  Berkas : Text;
  Kode : String[5];
  Nama : String[20];
  Harga : Longint;
  Jumlah: Longint;
  Total : real;

  Lagi,Jawab : Char;
  Ada : Boolean;
Begin
  Assign(Berkas,'BARANG.TXT');
  Reset(Berkas);
  Clrscr;
  While not EOF(berkas) Do
  Begin
    Readln(Berkas,kode,nama,harga,jumlah);
    Total := Harga * jumlah;
    writeln(kode:5,nama:20,harga:10,jumlah:10,total:15:2);
  End;
  Readln;
  Close(berkas);
End.
```