



Membuat File Text

File text adalah suatu file yang pola penyimpanannya datanya dalam bentuk karakter. Sehingga kalau suatu variabel bertipe int (2 byte) dengan isi 10000, maka akan disimpan dalam bentuk karakter 10000 (5 karakter) sehingga file yang dibuat besarnya 5 byte.

Contoh program untuk membuat file text :

```
1 #include <stdio.h>
2 #include <conio.h>
3 #include <String.h>
4 main()
5 {
6     FILE *f;
7     char st[81]="";
8     f=fopen("strings.dat","w");
9     if (f!=NULL)
10    {
11        do{
12            printf("Masukan string : ");gets(st);
13            fprintf(f,"%s\n",st);
14        }while (strlen(st)>0);
15        fclose(f);
16        printf("File selesai di buat");
17    }
18    else
19    {
20        printf("File gagal dibuat");
21    }
22    getch();
23    return 0;
24 }
```

Keterangan program :

- Baris 6 : Perintah FILE *f berarti deklarasi variabel f yang merupakan pointer ke suatu file .
- Baris 8 : membuat link / menghubungkan variabel file f dengan file strings.dat dengan mode operasi adalah w (write).
- Baris 9 : Jika file sukses di buat (variabel f menunjuk ke suatu file/tidak NULL) maka program akan mengerjakan perintah baris 11 – 16.
- Baris 11 dan 14: Perulangan selama string yang dimasukan panjangnya lebih dari 0.
- Baris 12 : Baca sebuah string, masukan ke variabel st

- Baris 13 : Tulis string st ke file f, dan kemudian diberikan tanda pindah baris (\n).
- Baris 15 : Menutup hubungan/link ke file f.
- Baris 18 : Jika file tidak bisa dibuat, maka akan menampilkan pesan File gagal dibuat.

Jika data yang dimasukan ingin disimpan di bagian belakang file, maka mode yang digunakan adalah mode "a"

Membaca File Text

Mode yang dipakai dalam membaca data text adalah mode "r". File yang dibaca bisa berekstensi apa saja.

Perhatikan perintah berikut :

```
1 #include <stdio.h>
2 #include <conio.h>
3 #include <String.h>
4 main()
5 {
6     FILE *f;
7     char st[81]="";
8     f=fopen("strings.dat","r");
9     if (f!=NULL)
10    {
11        clrscr();
12        while (fscanf(f,"%s",st)!=EOF) //(fgets(st,80,f)!=NULL)
13        {
14            printf("%s",st);
15        }
16        fclose(f);
17        printf("File sudah dibaca");
18    }
19    else
20    {
21        printf("File gagal dibaca");
22    }
23    getch();
24    return 0;
25 }
26
```

Keterangan program :

- Baris 8 : Buka sebuah file (strings.dat) dalam mode "r" (read).
- Baris 9 – 18 : jika variabel F tidak NULL (file sukses di buka), maka program akan mengeksekusi program baris 11 – 17.
- Baris 12 : Perulangan selama fscanf belum mencapai EOF (end of file). Fscanf akan mengambil data dari file bertipe string (%s") disimpan di st. Alternatif lain adalah menggunakan perintah fgets.
- Baris 14 : Menuliskan data yang telah dibaca.

Membuat File Biner

File biner adalah suatu file yang pola penyimpanannya adalah dengan menuliskan data secara biner. Sebagai contoh, jika data yang disimpan ke file tersebut bertipe int (2 byte) dengan isi variabel 10000, maka akan disimpan sebesar 2 byte dengan isi : 2710_h. Tipe jenis ini sangat cocok untuk penyimpanan data dalam bentuk struct. Pembacaan file biner dilakukan perblok.

Contoh program :

```
1 #include <stdio.h>
2 #include <conio.h>
3 main()
4 {
5     FILE *f;
6     int i;
7     f=fopen("dataint.abc", "wb");
8     if(f!=NULL)
9     {
10        do{
11            printf("Masukan angka : ");scanf("%d",&i);
12            if(i!=0)
13                fwrite(&i,sizeof(int),1,f);
14        }while (i!=0);
15        fclose(f);
16        printf("File telah selesai dibuat");
17    }
18    else
19    {
20        printf("Data tidak bisa dibuat");
21    }
22    getch();
23    return 0;
24 }
```

Keterangan program :

- Baris 7 : Buka file dalam mode "wb" (write binary).
- Baris 8 – 17 : Jika file bisa dibuat, maka akan melakukan perintah-perintah di baris 10 – 16.
- Baris 10 – 14 : Perulangan selama data yang dibaca dari keyboard tidak nol.
- Baris 11 : Baca suatu integer
- Baris 12-13 : Jika data tidak nol maka disimpan dalam file f sebanyak 1 data sebesar 2 byte (sizeof(int)) yang ada di alamat &i.
- Baris 15 : Menutup file f.

Membaca File Biner

Untuk membaca file biner, mode operasi filenya adalah "rb" (read binary). Pembacaan data untuk tipe biner dilakukan perblok, sehingga kita harus tahu besar data yang akan dibaca.

Perhatikan program di bawah ini :

```
1 #include <stdio.h>
2 #include <conio.h>
3 main()
4 {
5     FILE *f;
6     int i;
7     f=fopen("dataint.abc","rb");
8     if(f!=NULL)
9     {
10        while(fread(&i,sizeof(int),1,f)==1)
11            printf("angka : %i\n",i);
12        fclose(f);
13        printf("File telah selesai dibaca");
14    }
15    else
16    {
17        printf("Data tidak bisa dibaca");
18    }
19    getch();
20    return 0;
21 }
```

Keterangan program :

- Baris 7 : Buka file dataint.abc dalam mode rb (read binary).
- Baris 8 – 13 : Jika file bisa dibuka, maka akan mengeksekusi perintah baris 10 – 13
- Baris 10 : Lakukan pembacaan data fread, simpan data ke alamat I (&i), sebesar 2 byte (sizeof(int)), sebanyak 1 buah integer, dari file f. Pembacaan ini diulang selama fread == 1. fread menghasilkan nilai banyak data yang dibaca.
- Baris 11 : Menuliskan data hasil pembacaan (i) ke layar
- Baris 12 : Menutup file f.

Membaca File Biner Secara Acak

Ada kalanya kita ingin mengambil data dari posisi tengah file. Untuk itu, ada suatu fungsi `fseek`, yang gunanya untuk memindahkan kursor file ke posisi tertentu.

Posisi awal pencarian ada 3 jenis yaitu `SEEK_SET` (dari awal file), `SEEK_CUR` (dari posisi sekarang), `SEEK_END` (dari akhir file).

Contoh perintah :

- `fseek(f,10,SEEK_SET)`, pindahkan pointer file ke posisi (byte) ke-10 dari awal file.
- `fseek(f,5,SEEK_CUR)`, pindahkan pointer file 5 byte dari posisi pointer sekarang.
- `fseek(f,1,SEEK_END)`, pindahkan pointer posisi paling akhir (1 byte sebelum EOF).

Jadi kalau kita mempunyai file biner yang digunakan untuk menyimpan data int, maka kalau kita ingin mengambil data ke-5 perintah `fseek`nya adalah `fseek(f,5*sizeof(int),SEEK_SET)`.

Contoh dibawah ini akan menampilkan data dari posisi 3 dan seterusnya.

```
1 #include <stdio.h>
2 #include <conio.h>
3 main()
4 {
5     FILE *f;
6     int i;
7     f=fopen("dataint.abc","rb");
8     if(f!=NULL)
9     {
10        fseek(f,3*sizeof(int),SEEK_SET);
11        while(fread(&i,sizeof(int),1,f)==1)
12            printf("angka : %i\n",i);
13        fclose(f);
14        printf("File telah selesai dibaca");
15    }
16    else
17    {
18        printf("Data tidak bisa dibaca");
19    }
20    getch();
21    return 0;
}
```