

Algoritma dan Pemrograman Lanjut

Pertemuan Ke-13 Arsip (*File*) 2



Disusun Oleh :
Wilis Kaswidjanti, S.Si.,M.Kom.

**Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Pembangunan Nasional “Veteran”
Yogyakarta**

Algoritma dan Pemrograman Lanjut

Judul Materi : Arsip (*File*) 2

Deskripsi Materi : Materi ini membahas tipe data file, operasi-operasi file mengolah data dengan bahasa C++

Tujuan Instruksional Khusus :

1. Mendefinisikan dan menggunakan tipe data file
2. Mendeskripsikan tipe data file
3. Menggunakan tipe data file
4. Menjelaskan fungsi-fungsi pada tipe data file
5. Mengimplementasikan fungsi-fungsi file untuk memanipulasi file

Referensi :

- Buku Teks
Munir, Rinaldi (2005), *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, Buku 2, Edisi Ketiga, Penerbit Informatika Bandung, Bab 4, hal 115-168.
Charibaldi, N. (2004), *Modul Kuliah Algoritma Pemrograman II*, Edisi Kedua, Yogyakarta
- Buku Acuan/Referensi
Brassard, Gilles (1999), *Fundamentals of algorithma*, PrinteceHall.
Jarne, Stroustrup B. (1997), *C++ Programming language*, AT &T.
Kristanto, Andri (2003), *Algoritma pemrograman C++*, Graha Ilmu.
Schildt,Herbert (2000), *The Complete Reference C++*, McGraw-Hill.
Sedgewick, R. (2000), *Algoritma Third edition In C part 5*, Addison Wesley.

Arsip (*File*) 2

Operasi File

1. Membuka / Mengaktifkan File

```
FILE fopen(char *namafile, char *mode);
```

Keterangan mode :

r : Read only

w : Menyatakan file baru diciptakan. Operasi yang akan dilakukan adalah operasi perekaman data. Jika file tersebut sudah ada, isi yang lama akan dihapus.

a : Membuka file yang ada pada disk dan operasi yang akan dilakukan adalah operasi penambahan data pada file. Jika file belum ada, secara otomatis file akan dibuat .

r+ : Membuka file yang sudah ada, operasi yang dilakukan berupa pembacaan dan penulisan.

w+ : Membuka file untuk pembacaan/penulisan. Jika file sudah ada, isinya akan dihapus.

a+ : Membuka file, operasi yang dilakukan berupa perekaman dan pembacaan. Jika file sudah ada, isinya tak akan terhapus.

File biner adalah file yang pola penyimpanan di dalam disk adalah dalam bentuk biner, yaitu seperti bentuk dalam memori (RAM) komputer. Sedangkan file teks merupakan file yang pola penyimpanan datanya dalam bentuk karakter.

Keterangan mode :

rt : mode file adalah teks dan file hendak dibaca.

rt+ : mode file adalah teks dan file bisa dibaca atau ditulisi (= **r+t**).

rb : mode file adalah biner dan file hendak dibaca.

2. Menutup File

```
Int fclose(FILE *pf);
```

3. Menyimpan File per Character

```
Int fputc(int kar, FILE *ptr_file );
```

4. Membaca File per Character

```
Int fgetc(FILE *ptr_file );
```

Contoh program 1 :

```
/*-----*/
/* menciptakan & mengisi file dgn data karakter dari keyboard */
/*-----*/
#include <stdio.h>
#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
#define CTRL_Z 26
main()
{
    FILE *pf; /* pointer ke FILE */
    char kar;
    clrscr();
    if ((pf=fopen("COBA.TXT","w"))==NULL) /* ciptakan file */
    {
        cout<<" File tak dapat diciptakan !!\r\n"; /* tulis kelayar */
        exit(1); /* selesai */
    }
    while((kar=getche()) != CTRL_Z) /* baca kar dari kbd */
        fputc(kar,pf); /* tulis ke file */
    fclose(pf); /* tutup file */
}
```

Contoh program 2 :

```
/*-----*/
/* membaca isi file per karakter */
/*-----*/
#include <stdio.h>
#include <iostream.h>
#include <stdlib.h>
#include <conio.h>
main()
{
    FILE *pf; /* pointer ke FILE */
    char kar;
    clrscr();
    if((pf=fopen("COBA.TXT","r"))==NULL) /* buka file */
    {
        cout<<" File tak dapat dibuka !!\r\n"; /* tulis kelayar */
        exit(1); /* selesai */
    }
    while((kar = getc(pf)) != EOF) /* baca kar dari file */
        cout<<kar; /* tampilkan ke layar */
    fclose(pf); /* tutup file */
    getch();
}
```

5. Membaca File per Integer

```
Int getw(FILE *ptr_file);
```

6. Menyimpan File per Integer

```
Int putw(FILE *ptr_file);
```

Contoh program 3 :

```
/* ----- */
/* menyimpan data dengan putw() */
/* ----- */
#include <stdio.h>
#include <iostream.h>
#include <stdlib.h>
#include <conio.h>

main()
{
    FILE *pf;
    /*ptr ke FILE*/
    int nilai;
    char jawab;
```

```

clrscr();
if((pf=fopen("BILANGAN.DAT","wb"))==NULL) /*ciptakan file biner*/
{
    cout<<"File gagal diciptain !!\n";
    exit(1);
}
cout<<"Program untuk menyimpan data integer ke file.";

do
{
    cout<<"\r\n Bilangan yang akan disimpan : ";
    cin>>nilai; /*baca nilai dari keyboard*/
    putw(nilai, pf); /*tulis bilangan ke file*/
    cout<<" Memasukkan data lagi (Y/T)? : ";
    jawab =getche(); /*baca jawaban dari keyboard*/
}
while(jawab == 'y' || jawab == 'Y');

fclose(pf); /*tutup file*/
cout<<"\r\nOke. Data sudah disimpan pada file.\r\n";

getche();
}

```

Contoh program 4 :

```

/* -----*/
/* membaca isi file dengan getw() */
/* -----*/

#include <stdio.h>
#include <iostream.h>
#include <stdlib.h>
#include <conio.h>

main()
{
    FILE *pf; /*ptr ke file*/
    int nilai;
    int nomor = 0;

    clrscr();
    if((pf=fopen("BILANGAN.DAT","rb"))==NULL)/*buka file biner utk dibaca*/
    {
        cout<<" File gagal dibuka.\r\n";
        exit(1);
    }
    cout<<"\n Isi file BILANGAN.DAT : \r\n";
    while(1)
    {
        nilai = getw(pf); /* baca sebuah int dr file*/
        if(feof(pf) != NULL) break; /* jika akhir file,keluar loop*/
        cout<<" "<<+nomor<<" "<< nilai<<"\r\n";
    }
    fclose(pf); /* tutup file */
    getche();
}

```

7. Membaca File per Blok

Untuk menyimpan atau membaca data file dalam bentuk kesatuan blok (sejumlah byte), misalnya untuk tipe float atau struct (struktur)

```
int fread(void *buffer, int n, FILE *ptr_file);
```

8. Menyimpan File per Blok

```
int fwrite(void *buffer, int jum_byte, int n, FILE *ptr_file);
```

Contoh program 5 :

```
/* ----- */
/* menyimpan data bertipe struktur,dgn fwrite() */
/* ----- */

#include <stdio.h>
#include <iostream.h>
#include <stdlib.h>
#include <conio.h>

main()
{
    FILE *f_struktur;
    char jawab;

    struct data_pustaka
    { char judul[26];
      char pengarang[20];
      int jumlah;
    } buku;          /*variabel buku bertipe struktur*/

    if((f_struktur=fopen("DAFBUKU.DAT","wb"))==NULL)
    {
        cout<<"File tak dapat diciptakan !!\r\n";
        exit(1);
    }
    do
    {
        clrscr();
        cout<<" Judul buku          : "; cin>>buku.judul;
        cout<<" Nama pengarang       : "; cin>>buku.pengarang;
        cout<<" Jumlah buku           : "; cin>>buku.jumlah;
        fflush(stdin);          /*hapus isi penampung keyboard*/
        fwrite(&buku, sizeof(buku), 1, f_struktur); /*rekam sebuah data
                                                    struktur*/
        cout<<"\r\nMau merekam data lagi (Y/T) : "; jawab = getche();
    }
    while(jawab == 'Y' || jawab == 'y');
    fclose(f_struktur);        /*tutup file*/
}
```

Contoh program 6 :

```
/* -----*/
/* membaca sebuah struktur, dengan fread() */
/* -----*/

#include <stdio.h>
#include <iostream.h>
#include <stdlib.h>
#include <iomanip.h>
#include <conio.h>
#define JUM_BLOK 1          /*sebuah record*/

main()
{
    FILE *f_struktur;
    struct
    { char judul[26];
      char pengarang[20];
      int jumlah;
    } buku;                /*variabel struktur*/

    clrscr();
    if((f_struktur=fopen("DAFBUKU.DAT", "rb"))==NULL)
    {
        cout<<"File tak dapat dibuka\r\n";
        exit(1);
    }
    cout<<"\nDaftar buku :\n";
    cout<<"\n"<<"Judul"<<setw(26)<<"Pengarang"<<setw(20)<<"Jumlah\n";
    while(fread(&buku, sizeof(buku), JUM_BLOK, f_struktur) ==JUM_BLOK)
        cout<<buku.judul<< setw(26)<<buku.pengarang<<setw(20)<<
buku.jumlah<<endl;
    fclose(f_struktur);
    getch();
}
```

9. Membaca Data String dari File

Fungsi yang digunakan untuk membaca data string pada file yaitu **fgets()** untuk menyimpan string **str** ke dalam file. Dan **fputs()** untuk membaca string dari file sampai ditemukannya karakter baris-baru '\n' atau setelah (n-1) karakter, dengan **n** adalah panjang maksimal string yang dibaca per wktu-baca.

```
char *fgets(char *str, int n, FILE *ptr_file);
```

10. Menyimpan Data String dari File

```
int fputs(char *str, FILE *ptr_file);
```


Contoh program 7 :

```
/* -----*/
/*  membaca isi file teks dengan fgets()      */
/* -----*/
#include <stdio.h>
#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

#define PANJANG 256

main()
{
    FILE *f_teks;
    char string[PANJANG];
    char namafile[65];

    clrscr();

    cout<<"\n PROGRAM UNTUK MELIHAT ISI FILE TEKS\r\n";
    cout<<"\n Nama file : ";
    gets(namafile);

    if((f_teks=fopen(namafile, "rt"))==NULL)
    {
        cout<<"File tak dapat dibuka\r\n";
        exit(1);
    }

    /*baca string dari file selama masih ada*/
    while((fgets(string, PANJANG, f_teks)) != NULL)
    cout<<"\n Isi file :\n";
    cout<<" " <<string<<"\r";          /*tampilakan string ke layar*/
    fclose(f_teks);
    getch();
}
```

Contoh program 8 :

```
/* -----*/
/*      menyalin isi file teks                */
/* -----*/
#include <stdio.h>
#include <stdlib.h>
#include <iostream.h>
#include <conio.h>
#include <string.h>

#define PANJANG 256

main()
{
    FILE *pf_input, *pf_output;
    char string[PANJANG];
    char namafile_inp[65],namafile_out[65];

    clrscr();
```

```

cout<<"\n PROGRAM UNTUK MENYALIN ISI FILE TEKS\r\n";
cout<<"\n HURUF KECIL AKAN DIGANTI HURUF KAPITAL\r\n";
cout<<" Nama file input : ";
cin>>namafile_inp;
cout<<" Nama file output: ";
cin>>namafile_out;

if((pf_input=fopen(namafile_inp, "rt"))==NULL) /*buka file input*/
{ cout<<"File input tak dapat dibuka !!\r\n";
  exit(1);
}
if((pf_output=fopen(namafile_out, "wt"))==NULL) /*buka file output*/
{ cout<<"File output tak dapat dibuka !!\r\n";
  exit(1);
}

/*baca string dari file input & konversikan kehuruf kapital*/
/*kemudian rekam ke file output*/

while((fgets(string, PANJANG, pf_input)) != NULL)
{strupr(string);
  fputs(string, pf_output);
}
fcloseall(); /*tutup file input dan output*/
}

```

11. Mengganti Nama File

Fungsi yang berguna untuk menghapus file, yaitu **rename()**. Bentuk deklarasinya :

```
int rename(char *namafilelama, char *namafilebaru);
```

Contoh program 9 :

```

/* ----- */
/* Mengganti nama file */
/* ----- */

#include <stdio.h>
#include <iostream.h>
#include <conio.h>

main()
{
  int kode;
  char namafilelama[65], namafilebaru[65];

  clrscr();

  cout<<"\n Nama file yang akan diganti : ";
  cin>>namafilelama;
  cout<<"\n Nama file pengganti : ";
  cin>>namafilebaru;
}

```

```
kode = rename( namafilelama, namafilebaru );
if( kode == -1 )
    cout<<"  Gagal dalam mengganti nama file !!\r\n";
else
    cout<<"  Ok.Nama file sudah diganti !!\r\n";
getche();
}
```

PENUTUP

Bemacam-macam perintah untuk memanipulasi file yang terdapat pada bahasa C++. Perintah-perintah yang dipakai tersebut harus sesuai dengan tipe data yang disimpan pada file.

SOAL-SOAL

Buatlah contoh program untuk masing-masing operasi file.