

# Pengelolaan File

Modul **Praktikum**  
C++  
**Dasar Pemrograman Komputer**

Semester Genap 2017/2018

## MODUL 8

### PENGELOLAAN FILE

#### A. Tujuan

Setelah mempelajari bab ini, diharapkan mahasiswa dapat:

1. Mengetahui jenis file teks maupun file biner.
2. Membedakan jenis file teks maupun file biner.
3. Menangani file teks maupun file biner dalam Bahasa C.
4. Membuat aplikasi yang melibatkan pengelolaan file teks maupun file biner dalam Bahasa C.

#### B. Petunjuk

1. Awali setiap aktivitas Anda dengan doa agar Anda lancar dalam belajar.
2. Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik.
3. Kerjakan tugas-tugas praktikum dengan baik, jujur, dan sabar.
4. Tanyakan kepada instruktur apabila ada hal-hal yang kurang jelas.

#### C. Alat dan Bahan

1. PC / laptop yang memenuhi kapasitas untuk pemrograman Bahasa C++.
2. Software MinGW.
3. Modul praktikum tentang file.

#### D. Dasar Teori

**File** adalah sebuah organisasi dari sejumlah record. Masing-masing record bisa terdiri dari satu atau beberapa field. Setiap field terdiri dari satu atau beberapa byte.

##### 1. Membuka File

Untuk membuka atau mengaktifkan file, fungsi yang digunakan adalah fungsi `fopen()`. File dapat berupa file biner atau file teks. File biner adalah file yang pola penyimpanan di dalam disk dalam bentuk biner, yaitu seperti bentuk pada memori (RAM) komputer. File teks adalah file yang pola penyimpanan datanya dalam bentuk karakter. Penambahan yang perlu dilakukan untuk menentukan mode teks atau biner adalah "t" untuk file teks dan "b" untuk file biner. Prototype fungsi `fopen()` ada di header fungsi "stdio.h".

Bentuk umum:

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

Keterangan :

- namafile adalah nama dari file yang akan dibuka/diaktifkan.
- mode adalah jenis operasi file yang akan dilakukan terhadap file.

Jenis-jenis operasi file:

- r : menyorakan file hanya dapat dibaca (file harus sudah ada)
- w : menyatakan file baru akan dibuat/diciptakan (file yang sudah ada akan dihapus)
- a : untuk membuka file yang sudah ada dan akan dilakukan proses penambahan data (jika file belum ada, otomatis akan dibuat)
- r+ : untuk membuka file yang sudah ada dan akan dilakukan proses pembacaan dan penulisan.
- w+ : untuk membuka file dengan tujuan untuk pembacaan atau penulisan.

Jika file sudah ada, isinya akan dihapus. a+ : untuk membuka file, dengan operasi yang akan dilakukan berupa perekaman maupun pembacaan. Jika file sudah ada, isinya akan dihapus.

Contoh:

```
pf = fopen("COBA.TXT", "w");
```

Untuk menutup file, fungsi yang digunakan adalah fclose(). Prototype fungsi fclose() ada di header file "stdio.h".

Bentuk umum:

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

atau

```
int fcloseall (void);
```

## 2. Memproses File

### a. Menulis Karakter

Untuk menulis sebuah karakter, bentuk yang digunakan adalah:

```
putc (int ch, file *fp)
```

Keterangan:

- fp adalah pointer file yang dihasilkan oleh fopen()
- ch adalah karakter yang akan ditulis.

### b. Membaca Karakter

Untuk membaca karakter dari file, fungsi yang digunakan adalah:

```
getc (file *fp);
```

Keterangan:

- fp adalah pointer file yang dihasilkan oleh fopen()
- Fungsi feof(), digunakan untuk mendeteksi akhir file pada saat membaca data foef (file \*fp).

#### c. Membaca dan Menulis String

Fungsi untuk membaca dan menulis string adalah: fgets() dan fputs().

Bentuk umumnya:

```
fgets (char *str, int p, file *fp);
```

```
fputs (char *str, file *fp);
```

#### d. Membaca dan Menulis Blok Data

Fungsi untuk membaca dan menulis blok data adalah: fread() dan fwrite().

Bentuk umumnya:

```
fread (void *buffer, int b_byte, int c, file *fp);
```

```
fwrite (void *buffer, int b_byte, int c, file *fp);
```

Keterangan:

- buffer adalah pointer ke sebuah area di memori yang menampung data yang akan dibaca dari file.
- b\_byte adalah banyaknya byte yang akan dibaca atau ditulis ke file.
- c adalah banyaknya item yang dibaca/ditulis.

#### e. Membaca dan Menulis File yang Terformat

Jika diinginkan, data bilangan dapat disimpan ke dalam file dalam keadaan terformat, fungsi yang digunakan adalah:

```
fprintf (ptr_file, "string control", daftar argument);
```

```
fscanf (pts_file, "string control", daftar argument);
```

File sekuensial berisi rekord-rekord data yang tidak mengenal posisi baris atau nomor rekord pada saat aksesnya, dan setiap record dapat mempunyai lebar yang berbeda-beda. Akses terhadapnya selalu dimulai dari awal file dan berjalan satu persatu menuju akhir dari file. Dengan demikian, penambahan file hanya dapat dilakukan terhadap akhir file, dan akses terhadap baris tertentu harus dimulai dari awal file. Fungsi baku yang terkait dengan file sekuensial ini antara lain adalah fprintf, fscanf, dan rewind.

## E. Latihan

1. Eksekusi program berikut, jelaskan keluaran program hasil eksekusi tersebut!

```
//-----
#include <vcl.h>
#pragma hdrstop
#include <stdio.h>
#include <conio.h>
#define CTRL_Z 26
//-----

#pragma argsused
int main(int argc, char* argv[])
{
    FILE *pf; /* pointer ke file */
    char kar;

    if((pf = fopen("COBA.TXT", "w")) == NULL) /* ciptakan file */
    {
        cputs("File tak dapat diciptakan !\r\n");
        exit(1); /* selesai */
    }
    while((kar=getche()) != CTRL_Z)
        putc(kar, pf); /* tulis ke file */
    fclose(pf); /* tutup file */
    getch();
    return 0;
}
```

2. Jelaskan apa keluaran program berikut! Mengapa demikian?

```
//-----
#include <vcl.h>
#pragma hdrstop
#include <stdio.h>
#include <conio.h>
|
typedef struct
{
    long NIM;
    char Nama[25];
    float IPK;
} DataMhs;
|
//-----

#pragma argsuse

int main(int argc, char* argv[])
{
    FILE *MHS;
    DataMhs RekMhs;

    FILE *BIL;
    int i;
```

```
FILE *KAR;
char c;

getch();
return 0;
}
```

3. Perhatikan program berikut. Eksekusi program tersebut, dan jelaskan apa keluarannya!

```
//-----
#include <vcl.h>
#pragma hdrstop
#include <stdio.h>
#include <conio.h>

typedef struct
{
    long NIM;
    char Nama[25];
    float IPK;
} DataMhs;

//-----

#pragma argsused
int main(int argc, char* argv[])
{
    FILE *MHS;
    DataMhs RekMhs;
    MHS = fopen("kuliah.dat", "w");
    printf("NIM = "); scanf("%s", &RekMhs.NIM);
    printf("Nama = "); scanf("%s", &RekMhs.Nama);
    printf("IPK = "); scanf("%f", &RekMhs.IPK);
    fwrite(&RekMhs, sizeof(RekMhs), 1, MHS);
    fclose(MHS);
    getch();
    return 0;
}
```

4. Eksekusi Program berikut!. Apa keluarannya?

```
//-----
#include <vcl.h>
#pragma hdrstop
#include <stdio.h>
#include <conio.h>

typedef struct
{
    long NIM;
    char Nama[25];
    float IPK;
} DataMhs;

//-----

#pragma argsused
```

```

int main(int argc, char* argv[])
{
    /* Tulis file ke file kuliah.dat */
    FILE *MHS;
    DataMhs RekMhs;
    printf("Tulis ke file kuliah.dat\n");
    MHS = fopen("kuliah.dat", "w");
    printf("NIM = "); scanf("%ld", &RekMhs.NIM);
    printf("Nama = "); scanf("%s", &RekMhs>Nama);
    printf("IPK = "); scanf("%f", &RekMhs>IPK);
    fwrite(&RekMhs, sizeof(RekMhs), 1, MHS);
    fclose(MHS);

    /* Baca file dari file kuliah.dat */
    printf("Buka File kuliah.dat\n");
    FILE *BukaMhs;
    BukaMhs = fopen("kuliah.dat", "r");
    fread(&RekMhs, sizeof(RekMhs), 1, BukaMhs);
    printf("NIM = %ld \n", RekMhs.NIM);
    printf("Nama = %s \n", RekMhs>Nama);
    printf("IPK = %2.2f \n", RekMhs>IPK);
    fclose(BukaMhs);
    getch();
    return 0;
}

```

5. Program berikut menyajikan penanganan file sekuensial tentang data nasabah yang berisi tiga field, yaitu nomor identitas (account), nama (name), dan posisi tabungannya (balance) untuk (1) menyajikan yang tabungannya bernilai nol, (2) berstatus kredit, dan (3) berstatus debit. File data tersimpan dengan nama klien.dat.

Eksekusi program tersebut, tuliskan mengapa hasil demikian!

```

//-----
#include <vcl.h>
#pragma hdrstop
#include <stdio.h>
#include <conio.h>
//-----
#pragma argsused

int main(int argc, char* argv[])
{
    int request, account;
    float balance;
    char name[25];

    FILE *cfPtr;
    if ( (cfPtr = fopen("klien.dat", "r+")) == NULL )
    {
        printf("File could not be opened\n");
    }
    else
    {

```

```

printf ( "Enter request\n"
"1 - List accounts with zero balances\n"
"2 - List accounts with credit balances\n"
"3 - List accounts with debit balances\n"
"4 - End of run\n? " ) ;
scanf( "%d", &request );

while (request != 4)
{
fscanf (cfPtr, "%d%s%f", &account, name, &balance); switch (request)
{
case 1:
printf ("\nAccounts with zero balances:\n");
while ( !feof(cfPtr) )
{
if (balance == 0)
printf ("%10d%-13s7.2f\n", account, name, balance);
fscanf (cfPtr, "%d%s%f", &account, name, &balance);
}
break;

case 2:
printf ("\nAccounts with credit balances:\n");
while ( !feof(cfPtr) )
{
if (balance < 0)
printf ("%10d%-13s7.2f\n", account, name, balance);
fscanf (cfPtr, "%d%s%f", &account, name, &balance);
}
break;
rewind(cfPtr);
printf( "\n? " );
scanf ("%d", &request);
}
}
getch();
return 0;
}

```

## F. Tugas Praktikum

1. Apa perbedaan file teks dengan file biner?
2. Buat kesimpulan terhadap latihan 1 sampai dengan 5 yang telah Anda buat!
3. Buat program sorting untuk mengurutkan string, dimana program masukan maupun keluarnya berbentuk array/larik, yang disimpan dalam bentuk file. Anda boleh memilih file biner maupun teks. Contoh:

Masukkan:

S	B	a	A	o	E	W	p	C	x
---	---	---	---	---	---	---	---	---	---

Keluaran:

a	o	p	x	A	B	C	E	S	W
---	---	---	---	---	---	---	---	---	---



### **G. Tugas Rumah**

1. Seperti pada latihan 5, buatlah program aplikasi lain yang dapat digunakan untuk kepentingan pembelajaran di sekolah.
1. Berikan listing program, dokumentasi programnya, dan demonstrasikan hasilnya pada asisten.