

Pemrograman Bahasa C dengan Turbo C

Achmad Solichin
Sh-001@plasa.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.

Bab IX Operasi File

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) computer.
- ▶▶ 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 : menyarakan 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");

2 MENUTUP FILE

- ▶▶ 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);

3 MELAKSANAKAN PROSES FILE

- ▶▶ **Menulis Karakter**
 - ◆ Untuk menulis sebuah karakter, bentuk yang digunakan adalah :
putc(int ch, file *fp)
 - fp adalah pointer file yang dihasilkan oleh fopen()
 - ch adalah karakter yang akan ditulis.

Contoh Program :

```
#include "stdio.h"
#include "conio.h"
#define CTRL_Z 26
void main()
{
    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 */
    }
}
```

- ▶▶ **Membaca Karakter**
 - ◆ Untuk membaca karakter dari file, fungsi yang digunakan adalah :
getc(file *fp);
 - fp adalah pointer file yang dihasilkan oleh fopen()
 - Fungsi feof(), digunakan untuk mendeteksi akhir file.

- Pada saat membaca data foef(file *fp)

Contoh Program :

```
#include "stdio.h"
#include "conio.h"
void main()
{   file *pf;                               /* pointer ke file */
    char kar;
    clrscr();

    if((pf = fopen("COBA.TXT", "r")) == NULL) /* buka file */
    {   cputs("File tak dapat dibuka !\r\n");
        exit(1);                             /* selesai */
    }

    while((kar=getc(pf)) != EOF)
        putchar(kar);                        /* tampilkan ke layar */
    fclose(pf);                              /* tutup file */
}
```

▶▶ **Membaca dan Menulis String**

- ◆ Fungsi untuk membaca dan menulis string adalah : fgets() dan fputs()
- ◆ Bentuk Umum :
 fgets(char *str, int p, file *fp)
 fputs(char *str, file *fp)

▶▶ **Membaca dan Menulis Blok Data**

- ◆ Fungsi untuk membaca dan menulis blok data adalah : fread() dan fwrite()
- ◆ Bentuk umum :
 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 dibaca/ditulis.

Contoh Program :

```
#include "stdio.h"
#include "conio.h"
void main()
{   file *f_struktur;
    char jawaban;
    struct data_pustaka
    {   char judul[26];
        char pengarang[20];
        int jumlah;
    } buku;                               /* variabel buku bertipe struktur */

    /* buka file */

    if((f_struktur = fopen("DAFBUKU.DAT", "wb")) == NULL)/* buka file */
    {   cputs("File tak dapat diciptakan !\r\n");
        exit(1);                             /* selesai */
    }
}
```

```
do
{ clrscr();
  cputs("Judul Buku          :");
  gets(buku.judul);
  cputs("Nama Pengarang      :");
  gets(buku.pengarang);
  cputs("Jumlah buku           :");
  scanf("%i", &buku.jumlah);
  fflush(stdin);          /* Hapus isi penampung keyboard */

  /*Rekam sebuah data bertipe struktur */
  fwrite(&buku, sizeof(buku), 1, f_struktur);

  cputs("\r\nMau merekam data lagi (Y/T) ??");
  jawaban = getche();
}
while(jawaban == 'Y' || jawaban == 'y');

fclose(f_struktur);      /* tutup file */;
}
```

► 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);

Contoh Program :

```
#include "stdio.h"
#include "conio.h"
void main()
{ FILE *pformat;
  char jawaban;
  struct
  { int x;
    int y;
  } koordinat;

  /* Buka dan ciptakan file. Periksa kalau gagal dibuka */
  if((pformat = fopen("KOORDINAT.TXT", "w")) == NULL) /* buka file */
  { cputs("File tak dapat dibuka !\r\n");
    exit(1);          /* selesai */
  }
  do
  { clrscr();
    cputs("Masukkan data koordinat (bilangan integer)\r\n");
    cputs("Format : posisi x posisi y\r\n");
    cputs("Contoh : 20 30 [ENTER]\r\n");
    scanf("%i %i, &koordinat.x, &koordinat.y);
    fflush(stdin);

    /* Rekam ke file */
    fprintf(pformat, "%5i %5i\n", koordinat.x, koordinat.y);
    cputs("\r\nMenyimpan data lagi (Y/T) ??");
```

```
        jawaban = getch();
    }
    while(jawaban == 'y' || jawaban == 'Y');
    fclose(pformat);
    getch();
}
```

Contoh Program 2 :

```
#include <stdio.h>

FILE *in;
void BACA( int[ ] );
void CETAK( int[ ] );

void main() {
    int tabel[26] = {0};
    BACA(tabel);
    CETAK(tabel);
}

void BACA ( int huruf[ ] ) {
    char c;
    if (( in = fopen("data.txt", "r") ) == NULL)
        printf ("File tidak bisa dibaca\n");
    else
        while ( (ch = fgetc(in)) != EOF ) {
            c = (( c >= 97 ) || ( c <= 122 )) ? c - 32 : c ;
            if ( ( c >= 65 ) || ( c <= 90 ) )
                ++huruf [ c - 65 ];
        }
    fclose(in);
}

void CETAK ( int huruf[ ] ) {
    int counter;
    for ( counter = 0 ; counter <= 25 ; counter++ )
        printf ("\n%c%5d", counter + 65, huruf[counter] );
}
```

4 FILE SEKUENSIAL

File sekuensial berisi record-record data yang tidak mengenal posisi baris atau nomor record 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** . 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 debet. File data tersimpan dengan nama klien.dat.

```
#include <stdio.h>

void main() {
    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;

                case 3:
                    printf ("\nAccounts with debit 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);
        }
    }
}
```

```
    printf("End of run.\n");  
    fclose(cfPtr);  
}  
}
```