Lab Pemrograman I

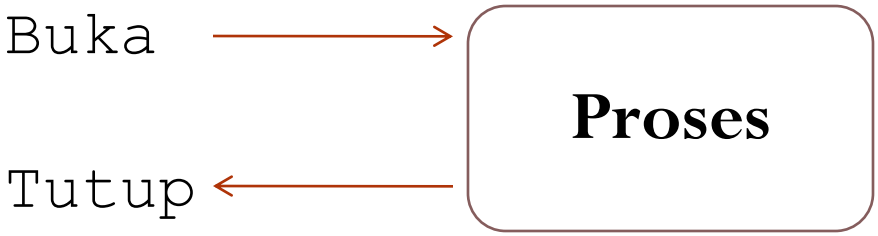
File

Dosen : Eko Budi Setiawan

Tahapan Operasi File

Tahapan Operasi File pada dasarnya meliputi tiga tahapan :

- Membuka /Mengaktifkan File
- Melaksanakan Proses File
- Menutup File



Membuka / Mengaktifkan File

Sebelum file dapat diakses (dibaca atau ditulisi), mula-mula file harus diaktifkan lebih dahulu. Fungsi yang digunakan yaitu fopen(). Bentuk deklarasinya adalah :

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

Dimana :
namafile : menyatakan nama dari file yang akan diaktifkan
mode : jenis operasi yang akan dilakukan terhadap file tsb

Jenis-jenis Operasi File

Jenis-jenis operasi File adalah sebagai berikut :

r : Menyatakan file hanya akan dibaca, jika file belum ada maka tidak akan berhasil

w : Menyatakan bahwa file baru diciptakan. Jika file tersebut sudah ada dalam disk, isi yang lamanya akan terhapus.

a : Untuk membuka file yang sudah ada untuk ditambah dengan data, jika file belum ada maka akan dibuat yang baru

r+ : sama dengan "r" tapi juga dapat ditulisi

w+ : sama dengan "w" tapi juga dapat dibaca

Jenis-jenis Operasi File [fopen]

- Berhasil tidaknya operasi pengaktifkan file dapat dilihat pada keluarn fungsi fopen().

- Jika Keluaran fungsi berupa NULL, berarti pengaktifkan gagal → misalnya membuka file dengan mode 'r' tapi filenya belum ada

- Contoh

```
FILE *pf; //deklarasi variabel pf
pf = fopen("COBA.TXT", "w");
```

  - Menciptakan dan mengaktifkan file bernama "COBA.TXT"
  - dengan mode yaitu "w" (mode penulisan ke file)
  - dan menempatkan pointer FILE ke variabel pointer pf

Jenis-jenis Operasi File [fopen]

❖Contoh Bentuk pengaktifan File

```
FILE *pf;
pf=fopen("COBA.TXT","a"); //membuka file
if(pf==NULL)
{
  printf("File tidak bisa dibuka\n");
}
```

Keterangan :

- pf akan diisi dengan keluaran dari fungsi **`fopen()`**
- Jika nilainya NULL, maka akan mencetak “File tidak bisa dibuka”

Menutup File

Apabila suatu file tidak diproses lagi, file perlu ditutup. Hal ini sangat penting terutama jika melakukan pemrosesan file yang jumlahnya lebih dari satu. Untuk menutup file, fungsi yang digunakan adalah **`fclose()`**. Bentuk Deklarasinya adalah :

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

Fungsi **`fclose()`** menghasilkan keluaran berupa nol jika operasi penutupan file berhasil dilakukan

Jenis FILE

- **File Biner** : File yang pola penyimpanan didalam disk berbentuk biner, yaitu seperti bentuk pada memori RAM (komputer). Digunakan untuk menyimpan data kompleks, e.g : struct
- **File Teks** : File yang pola penyimpanan datanya dalam bentuk karakter
- Penentuan mode teks dan mode biner :

  t untuk mode teks

  b untuk mode biner

Contoh :

"rt"	:	mode file adalah teks dan file hendak dibaca
"rt+"	:	mode file adalah teks dan file bisa dibaca dan ditulisi. Bisa juga ditulis : "r+t"
"rb"	:	mode file adalah biner dan file hendak dibaca

Operasi Penyimpanan dan Pembacaan File

Operasi Penyimpanan Karakter di File

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

Operasi Pembacaan Karakter di File

```
int fgetc(FILE, FILE *ptr_file);
```

Operasi Penyimpanan Integer di File

```
int putw(intnilai, FILE *ptr_file);
```

Operasi Pembacaan Integer di File

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

Operasi Penyimpanan dan Pembacaan File

Operasi Penyimpanan String di File

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

Operasi Pembacaan String di File

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

Menghapus dan Mengganti nama File

Menghapus File

```
int remove (char *namafile);
```

Jika penghapusan file berhasil, output = 0

Mengganti Nama File

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

Jika operasi penggantian berhasil, output = 0

Praktikum [1] – isi file karakter

```
#include<stdio.h>

main()
{
    FILE *pf;
    char kar;
    //membuka file
    pf=fopen("COBA.TXT","w");
    if(pf==NULL)
    {
        printf("File tidak bisa dibuka\n");
    }
    printf("Ketikkan apa saja, akhiri dengan ENTER.\n");
    printf("Program akan membaca perkarakter");
    printf(" dan menyimpannya dalam file COBA.TXT\n\n");
    //memasukkan karakter per karakter
    while((kar = getchar()) != '\n')
    fputc(kar,pf);
    //menutup file
    fclose(pf);
}
```

Praktikum [2] – menambah isi file

```
#include<stdio.h>

main()
{
    FILE *pd;
    char kar;
    //Buka file
    pd=fopen("COBA.TXT","r+");
    if(pd==NULL)
    {
        printf("File tidak dapat dibuka \n");
    }
    //Baca karakter per karakter sampai ketemu End Of FIle
    while((kar=fgetc(pd)) != EOF)
        fputchar(kar);
    while((kar=getchar()) != '\n')
        fputc(kar,pd);
fclose(pd); //tutup file
}
```

Praktikum [3] – menulis file biner

```
#include<stdio.h>

main()
{
    FILE *pf;
    int i,data,nilai;
    pf=fopen("data.dat","wb");
    if(pf==NULL)
    {
        printf("File tidak bisa dibuka\n");
    }
    printf("Masukkan banyaknya data\n");scanf("%d",&data);

    for(i=1;i<=data;i++)
    {
        printf("data yang disimpan : ");scanf("%d",&nilai);
        putw(nilai,pf);
    }
printf("\nOke. Data sudah disimpan di file\n");
fclose(pf);
}
```

Praktikum [4] – membaca file biner

```
#include <stdio.h>

main()
{
   FILE *pf; /* ptr ke file */
   int nilai, nomor = 0;
   /* Buka file biner untuk dibaca */
   if((pf=fopen("data.DAT","rb")) == NULL)
      {
         printf("File gagal dibuka.\n");
      }
      printf("Isi file data.DAT : \n");
   while(1) /* file berhasil dibuka */
      {
         nilai = getw(pf); /* Baca sebuah int dr file */
         if (feof(pf) != 0) break; /*Jika akhir file, keluar loop*/
         printf("%2d. %d \n", ++nomor, nilai); /* Tampilkan ke layar */
      }
fclose(pf); /* Tutup file */
}
```

Praktikum [5] – membaca file teks

```
#include<stdio.h>
#include<string.h>

main()
{
    FILE *pf;
    char nama[20];
    pf=fopen("coba.txt","r");
    if(pf==NULL)
    {
        printf("File gagal dibuka\n");
    }
    while (fgets(nama,6,pf))
    printf("%s\n",nama);
    fclose(pf);
}
```

Praktikum [6] – mengganti nama file

```c
#include <stdio.h>

#define PJG 65
main()
{
    int kode;
    char namafilelama[PJG], namafilebaru[PJG];
    printf("Nama file yang akan diganti : ");gets(namafilelama);
    printf("Nama file yang baru : ");gets(namafilebaru);
    kode = rename(namafilelama, namafilebaru);
    if(kode == 0)
        printf("Nama file sudah diganti\n");
    else
    printf("Gagal dalam mengganti nama\n");
}
```

Praktikum [7] – menghapus file

```
#include <stdio.h>
#define PJG 65
main()
{
    int kode;
    char namafile[PJG];
    printf("Nama file yang akan dihapus : ");
    gets(namafile);
    kode = remove(namafile);
    if(kode == 0)
        printf("File sudah dihapus\n");
    else
        printf("Gagal dalam menghapus file\n");
}
```