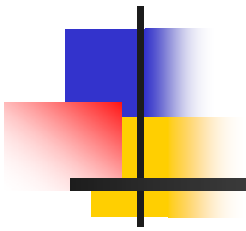


OPERASI FILE



SASARAN

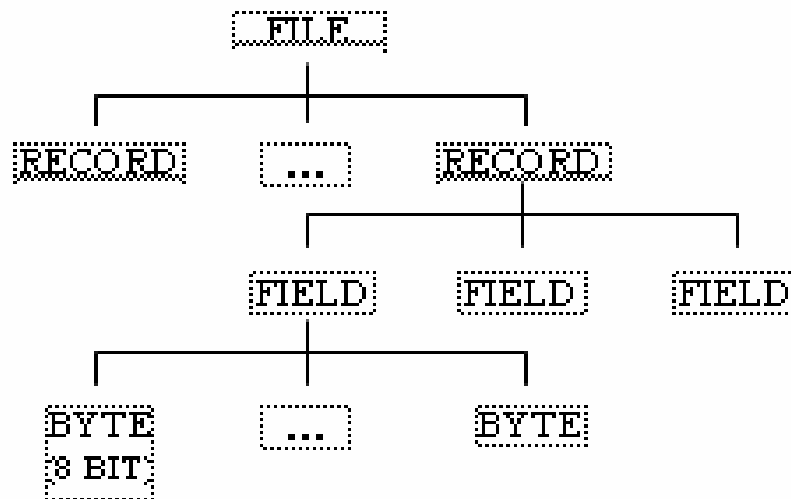


Setelah menyelesaikan bab ini, anda diharapkan dapat:

- Menjelaskan tentang struktur file
- Menjelaskan tentang tahap-tahap operasi pada file
- Menjelaskan tentang fungsi untuk penyimpanan dan pembacaan file per-karakter

DASAR STRUKTUR FILE

- Penyimpanan suatu data dalam disk berupa suatu file.
- Gambar struktur file :

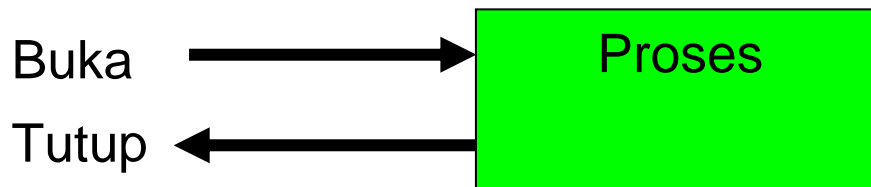


Catatan : *record* adalah nama lain dari struktur (*struct*).



DASAR FILE – Cont. 1

- Tahapan Operasi File :
 1. Membuka/mengaktifkan file
 2. Melaksanakan proses file
 3. Menutup file





MEMBUKA / AKTIFKAN FILE

- Bentuk deklarasi :

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

Keterangan :

- **namafile** berupa nama dari file yang akan diaktifkan
- **mode** berupa jenis operasi yang akan dilakukan terhadap file
- prototipe ada pada file **stdio.h**



JENIS OPERASI FILE

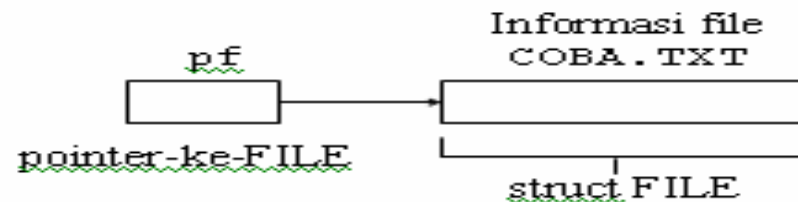
- **r** menyatakan file hanya akan dibaca, jika file belum ada maka tidak akan berhasil.
- **w** menyatakan bahwa file baru diciptakan. Jika file tersebut sudah ada dalam disk, isinya yang lama akan terhapus.
- **a** untuk membuka file yang sudah ada untuk ditambah dengan data, jika file belum ada akan dibuat yang baru.
- **r+** sama dengan "r" tetapi selain file dapat dibaca, file juga dapat ditulisi.
- **w+** sama dengan "w" tetapi selain file dapat ditulisi, file juga dapat dibaca.
- **a+** sama dengan "w" tetapi selain file dapat ditulisi, file juga dapat dibaca.

JENIS OPERASI FILE – Cont. 1

- Berhasil tidaknya operasi pengaktifan file dapat dilihat pada keluaran fungsi *fopen()*.
- Jika keluaran fungsi berupa NULL (suatu makro yang didefinisikan pada file **stdio.h**), berarti operasi pengaktifan file gagal → misal 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-ke-FILE ke variabel pointer pf





JENIS OPERASI FILE – Cont. 2

■ Contoh Bentuk pengaktifan file :

```
if (pf = fopen("COBA.TXT", "w") == NULL)
{
    printf("File tidak dapat diciptakan !\n");
    exit(1);           //keluar dari program
}
```

Keterangan :

- > pf akan diisi dengan keluaran dari fungsi *fopen()*.
- > Jika nilainya NULL, maka akan mencetak "File tidak dapat diciptakan", setelah itu program dihentikan.

MENUTUP FILE

- Apabila file sudah tidak diproses lagi, maka file tersebut ditutup, karena adanya keterbatasan jumlah file yang dapat dibuka secara serentak.
- Perintah yang digunakan : `fclose()`;
- Bentuk deklarasi : `int fclose(FILE *pf);`
- Bentuk deklarasi yang lain :

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

prototype yang digunakan : `stdio.h`

OPERASI FILE

Operasi Penyimpanan dan Pembacaan File Per Karakter

A. OPERASI PENYIMPANAN FILE

- Penyimpanan karakter ke file menggunakan perintah : `fputc()`.

- Bentuk deklarasi :

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

- **ptr_file** adalah pointer-ke-FILE yang berisi keluaran dari `fopen()`,
- **kar** berupa karakter yang akan disimpan dalam file.

CONTOH PROGRAM TULIS

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

Bisa dilihat hasilnya dengan notepad

```
main()
{
```

```
    FILE *pf;           /* Pointer-ke-FILE */
    char kar;
```

```
    /* Ciptakan file */
```

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

```
    {
```

```
        printf("file tak dapat diciptakan!\r\n");
        exit(1);
```

```
    }
```

```
    //Masukkan karakter per karakter sampai ditekan ENTER
```

```
    while((kar=getchar()) != '\n')
```

```
        fputc(kar, pf);
```

```
    fclose(pf);        /* tutup file */
```

```
}
```

OPERASI FILE – Cont. 1

B. OPERASI PEMBACAAN FILE

- Pembacaan karakter dari suatu file memakai perintah : `fgetc()`.
- Bentuk deklarasi :

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

PROSES PEMBACAAN FILE PER KARAKTER :

1. Buka file COBA.TXT dengan mode “r”
Jika tidak berhasil dibuka maka
 - beri keterangan pada layar bahwa file tak ada
 - selesai
2. Baca sebuah karakter dari file
Jika karakter sama dengan EOF (tanda akhir file) maka ke langkah 4
3. Tampilkan karakter ke layar dan kembali ke langkah 2
4. Tutup file
5. Selesai

CONTOH PROGRAM BACA TULIS

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

main()
{
    FILE *pf;                /* Pointer-ke-FILE */
    char kar;
    /* Ciptakan file */
    if ((pf = fopen("COBA.TXT","r+")) == NULL)
    {
        printf("file tak dapat diciptakan!\r\n");
        exit(0);
    }
    while((kar=fgetc(pf)) != EOF) /* baca kar dari file */
        putchar(kar);

    while((kar=getchar()) != '\n') /*baca kar dr keyboard*/
        fputc(kar, pf);

    fclose(pf);
}
```