

Praktikum

FILE

A. TUJUAN

1. Menjelaskan tentang struktur file
2. Menjelaskan tentang tahap-tahap operasi pada file
3. Menjelaskan tentang fungsi untuk penyimpanan dan pembacaan file per-karakter
4. Menjelaskan tentang file biner dan file teks
5. Menjelaskan tentang operasi penyimpanan dan pembacaan file per-int
6. Menjelaskan tentang operasi penyimpanan dan pembacaan file per-blok
7. Menjelaskan cara membaca dan menyimpan data string pada file
8. Menjelaskan cara mengakses file biner secara acak
9. Menjelaskan cara menghapus file
10. Menjelaskan cara mengganti nama file

B. PERCOBAAN

1. Menciptakan dan mengisi file dengan data karakter dari keyboard

```
/* File program: fputc.c */
#include <stdio.h>
#include <stdlib.h>

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);                             /* selesai */
    }

    printf("Ketikkan apa saja, akhiri dengan ENTER.\n");
    printf("Program akan membaca perkarakter");
    printf(" dan menyimpannya dalam file COBA.TXT\n\n");
```

```

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

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

```

2. Membaca isi file per karakter

```

/* File program: fgetc.c */
#include <stdio.h>
#include <stdlib.h>

main()
{
    FILE *pf;
    char kar;

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

    while((kar=fgetc(pf)) != EOF) /* baca kar dari file */
        putchar(kar);               /* tampilkan ke layar*/

    printf("\n");
    fclose(pf);                     /* tutup file */
}

```

3. Menyimpan data bertipe int menggunakan putw()

```

/* File program: _putw.c */
#include <stdio.h>
#include <stdlib.h>

main( )
{
    FILE *pf;                                /* ptr-ke-FILE */
    int nilai, sudah_benar;
    char jawab;

    /* ciptakan file*/
    if((pf=fopen("BILANGAN.DAT", "wb")) == NULL )
    {
        printf("file gagal diciptakan!\n");
        exit(1);
    }
}

```

```

printf("MENYIMPAN DATA INTEGER KE FILE\n");

do {
    printf("\nBilangan yang akan disimpan: ");
    scanf("%d", &nilai); /* baca nilai dr keyboard */
    _putw(nilai, pf); /* baca bilangan ke file */
    printf("memasukkan data lagi (Y/T)? ");

    do
    {
        jawab = getchar(); /* baca jawaban dari keyboard */
        sudah_benar = ((jawab == 'Y') || (jawab == 'y') ||
            (jawab == 'T') || (jawab == 't'));
    } while(! sudah_benar);
} while (jawab == 'y' || jawab == 'Y');

printf("\nOke. Data sudah disimpan dalam file.\n");
fclose(pf); /*menutup file */
}

```

4. Membaca isi file biner menggunakan getw()

```

/* File program : _getw.c */
#include <stdio.h>
#include <stdlib.h>

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

    /* Buka file biner untuk dibaca */
    if((pf=fopen("BILANGAN.DAT","rb")) == NULL)
    {
        printf("File gagal dibuka.\n");
        exit(1);
    }

    printf("Isi file BILANGAN.DAT : \n");

    while(1) /* file berhasil dibuka */
    {
        nilai = _getw(pf); /* Baca sebuah int dari file */
        if (feof(pf) != 0) /* Jika akhir file, keluar loop */
            break;
        /* Tampilkan ke layar */
        printf("%2d. %d \n", ++nomor, nilai);
    }
}

```

```

    fclose(pf);
}
/* Tutup file */

```

5. Menyimpan data bertipe struktur ke file menggunakan fwrite()

```

/* File program : fwrite.c */
#include <stdio.h>
#include <stdlib.h>

main()
{
    FILE *f_struktur;
    char jawaban;
    int sudah_benar;

    struct {
        char judul[30];
        char pengarang[30];
        int jumlah;
    } buku; /* var buku bertipe struktur */

    /* Buka file */
    if((f_struktur = fopen("DAFBUKU.DAT", "wb")) == NULL)
    {
        printf("File tidak dapat diciptakan !\n");
        exit(1);
    }

    do {
        fflush(stdin); /* Hapus isi penampung keyboard */
        printf("Judul buku      : ");
        gets(buku.judul);
        printf("Nama pengarang      : ");
        gets(buku.pengarang);
        printf("Jumlah buku          : ");
        scanf("%d", &buku.jumlah);
        fflush(stdin); /* Hapus isi penampung keyboard */

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

        printf("\nMau merekam data lagi [Y/T] ?");

        do {
            jawaban = getchar();
            sudah_benar = ((jawaban == 'Y') || (jawaban == 'y')
                || (jawaban == 'T') || (jawaban == 't'));
        } while(!sudah_benar);
    }
}

```

```

        printf("\n");

    } while(jawaban == 'Y' || jawaban == 'y');

    fclose(f_struktur);          /* Tutup file */
}

```

6. Membaca data bertipe struktur ke file menggunakan fread()

```

/* File program : fread.c */
#include <stdio.h>
#include <stdlib.h>

main()
{
    FILE *f_struktur;
    int i=1;

    struct {
        char judul[30];
        char pengarang[30];
        int jumlah;
    } buku;          /* var buku bertipe struktur */

    /* Buka file */
    if((f_struktur = fopen("DAFBUKU.DAT", "rb")) == NULL)
    {
        printf("File tidak dapat dibuka !\n");
        exit(1);
    }

    printf("%2s. %-30s %-30s %s\n\n", "No", "Judul Buku",
        "Nama Pengarang", "Jumlah");

    /* diulang selama masih ada sebuah record terbaca dalam
    file */
    while(fread(&buku, sizeof(buku), 1, f_struktur) == 1)
        printf("%2d. %-30s %-30s %4d\n", i++, buku.judul,
            buku.pengarang, buku.jumlah);

    printf("\n");
    fclose(f_struktur);      /* Tutup file */
}

```

7. Membaca isi file teks

```
/* File program : fgets.c */
#include <stdio.h>
#include <stdlib.h>

#define PANJANG 256

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

    printf("PROGRAM UNTUK MELIHAT ISI FILE TEKS\n\n");
    printf("Masukkan nama file : ");
    gets(namafile);

    printf("\nIsi file %s adalah sbb :\n", namafile);
    if((f_teks=fopen(namafile,"rt")) == NULL)
    {
        printf("File gagal dibuka\n");
        exit(1);
    }

    while(fgets(string, sizeof string, f_teks)!= NULL);
        printf("%s\n\n", string);

    fclose(f_teks);
}
```

8. Membaca kemudian menyalin isi file teks

```
/* File program : fputs.c */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define PANJANG 256

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

    printf("PROGRAM UNTUK MENYALIN ISI FILE TEKS\n\n");
    printf("Masukkan nama file input : ");
    gets(namafile_inp);
```

```

printf("Masukkan nama file output: ");
gets(namafile_out);

/* Buka file input */
if((pf_input=fopen(namafile_inp,"r+")) == NULL)
{
    printf("File input gagal dibuka\n");
    exit(1);
}

/* Buka file output */
if((pf_output=fopen(namafile_out,"w+")) == NULL)
{
    printf("File output gagal dibuka\n");
    exit(1);
}

/* menampilkan isi file input, merubahnya ke huruf besar
dan menyalinnya ke file output */
while(fgets(string, sizeof string, pf_input) != NULL)
{
    printf("\nIsi file %s adalah :\n",namafile_inp);
    printf("%s\n", string);
    strupr(string);
    fputs(string, pf_output);
}

fcloseall();

/* Buka file output */
if((pf_output=fopen(namafile_out,"r+")) == NULL)
{
    printf("File output gagal dibuka\n");
    exit(1);
}

/* tampilkan isi file output */
printf("\nIsi dari file %s adalah : \n",namafile_out);
while(fgets(string, sizeof string, pf_output) != NULL)
    printf("%s\n\n",string);

fclose(pf_output);
}

```

9. Membaca isi file secara random

```
/* File program : baca_acak.c */
#include <stdio.h>
#include <stdlib.h>

main()
{
    struct {
        char judul[30];
        char pengarang[30];
        int jumlah;
    } buku;                                /* var buku bertipe struktur */

    FILE *pf;
    char jawab;
    int i, no_record, sudah_benar;
    long int offset_byte;

    /* Buka file */
    if((pf = fopen("DAFBUKU.DAT", "rb")) == NULL)
    {
        printf("File tidak dapat dibuka !\n");
        exit(1);
    }

    do
    {
        i = 1;
        printf("Nomor record dari data yg mau ditampilkan : ");
        scanf("%d", &no_record);

        offset_byte = (no_record-1) * sizeof(buku);
        fseek(pf, offset_byte, SEEK_SET);

        if(fread(&buku, sizeof(buku), 1, pf) == 0)
            printf("Nomor record tdk dikenali!\n");
        else
        {
            printf("\n%2s. %-30s %-30s %s\n\n", "No",
                "Judul Buku", "Nama Pengarang", "Jumlah");
            printf("%2d. %-30s %-30s %4d\n", i++, buku.judul,
                buku.pengarang, buku.jumlah);
        }

        printf("\nMau mencoba lagi (Y/T)? ");
    }
```



```

do
{
    jawab = getchar();    /* baca jawaban dr keyboard */
    sudah_benar = ((jawab == 'Y') || (jawab == 'y') ||
        (jawab == 'T') || (jawab == 't'));
} while(! sudah_benar);

} while (jawab == 'y' || jawab == 'Y');

printf("\n");
fclose(pf);                /* Tutup file */
}

```

10. Mengganti isi suatu record secara random

```

/* File program : gantirec.c */
#include <stdio.h>
#include <stdlib.h>

#define SATU_RECORD 1

main()
{
    struct {
        char judul[30];
        char pengarang[30];
        int jumlah;
    } buku;                /* var buku bertipe struktur */

    FILE *pf;
    char jawab;
    int i, no_record, sudah_benar;
    long int offset_byte;

    /* Buka file */
    if((pf = fopen("DAFBUKU.DAT", "rb")) == NULL)
    {
        printf("File tidak dapat dibuka !\n");
        exit(1);
    }

    do
    {
        i = 1;
        printf("Nomor record dari data yg mau ditampilkan : ");
        scanf("%d", &no_record);
    }
}

```

```

offset_byte = (no_record-1) * sizeof(buku);
fseek(pf, offset_byte, SEEK_SET);

if(fread(&buku, sizeof(buku), 1, pf) == 0)
    printf("Nomor record tdk dikenali!\n");
else
{
    printf("\n%2s. %-30s %-30s %s\n\n", "No", "Judul
        Buku", "Nama Pengarang", "Jumlah");
    printf("%2d. %-30s %-30s %4d\n", i++, buku.judul,
        buku.pengarang, buku.jumlah);
}

printf("\nMau mencoba lagi (Y/T)? ");

do
{
    jawab = getchar();    /* baca jawaban dr keyboard */
    sudah_benar = ((jawab == 'Y') || (jawab == 'y') ||
        (jawab == 'T') || (jawab == 't'));
} while(! sudah_benar);

} while (jawab == 'y' || jawab == 'Y');

printf("\n");
fclose(pf);                                /* Tutup file */
}

```

11. Mengganti nama file

```

/* File program : gantinama.c */
#include <stdio.h>
#include <stdlib.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");
}

```

12. Menghapus file

```

/* File program : hapusfile.c */
#include <stdio.h>
#include <stdlib.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");
}

```

C. SOAL

1. a. Buatlah program untuk menambahkan tulisan :

BELAJAR BAHASA C

ke dalam file **COBA.TXT** yang dipakai pada program **fputc.c** di atas. Caranya, ubahlah mode “w” pada program tsb dengan mode “a” dan gantilah komentar “**File tidak dapat diciptakan!**” dengan “**File tidak dapat dibuka!**”

- b. Tampilkan isi file **COBA.TXT** yang baru dengan menggunakan NOTEPAD atau dengan menjalankan program **fgetc.c**. Perhatikan hasilnya.