

**LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN  
MODUL 5 – OPERASI FILE**



Muhammad Firyanul Rizky

1708561006

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS UDAYANA  
TAHUN 2018**

# **BAB I**

## **PEMBAHASAN**

### **1.1. Deskripsi**

Pada praktikum ini akan dipraktekan tentang operasi-operasi file pada bahasa pemrograman C.

### **1.2. Tujuan Umum**

Setelah mengikuti praktikum ini, mahasiswa mampu menggunakan berbagai operasi file yang terdapat pada bahasa pemrograman C dalam memecahkan suatu permasalahan.

### **1.3. Indikator Pencapaian**

Adapun indikator pencapaian dari modul praktikum ini adalah mahasiswa dapat mengimplementasikan berbagai operasi file dalam penyelesaian permasalahan-permasalahan yang ada ke dalam sebuah program sederhana.

## **BAB II**

### **LANDASAN TEORI**

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

#### **2.1. 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 polapenyimpanan 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 pada header fungsi “**stdio.h**”

## 2.2. 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.

### 2.2.1. Jenis-jenis operasi file:

- **r** : menandakan 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");
```

## 2.3. 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);
```

## 2.4. Melaksanakan Proses File

### 2.4.1. 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.

```
#include "stdio.h"
#include "conio.h"
#define CTRL_Z 26

void main(){

    file *pf; // pointer ke file

    char kar;

    // ciptakan file
    if((pf = fopen("COBA.TXT", "w")) == NULL) {
        cputs("File tak dapat diciptakan
        !\r\n"); exit(1);
    }

    while((kar=getche()) != CTRL_Z)
        putc(kar, pf); // tulis ke file

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

### **2.4.2. 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)

```
#include "stdio.h"
#include "conio.h"

void main(){

    file *pf; // pointer ke file

    char kar;

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

    while((kar=getc(pf)) != EOF)
        putchar(kar); // tampilkan ke layar

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

### **2.4.3. 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)
```

#### 2.4.4. Membaca dan Menulis Blok Data

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

Bentuk umum :

```
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.

```
#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){ cputs("File tak dapat diciptakan
        !\r\n"); exit(1);
    }

    do{

        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 = getch();

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

    fclose(f_struktur);
}
```

### **2.4.5. 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);
```

```
#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) {
        cputs("File tak dapat dibuka
        !\r\n"); exit(1);
    }

    do {
        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 = getche();
    } while(jawaban == 'y' || jawaban == 'Y');

    fclose(pformat);
    getch();

}
```





```

        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);

}

```

## **BAB III**

### **PEMBAHASAN**

#### **3.1. Penjelasan Program**

Pada saat user pertama kali menjalankan program, program akan menampilkan pilihan menu awal yang tersedia. Jika User memilih pilihan 1, maka program akan memanggil fungsi `'tambah() ;'` untuk menambah data, user akan diperintahkan untuk memasukkan nama, usia dan jenis kelamin kedalam list. Program akan kembali ke Menu Awal setelah user memilih menu 1 pada Menu Kembali.

Jika user memilih pilihan 2, maka program akan memanggil fungsi `'tampil() ;'` untuk menampilkan data yang tersimpan pada file.

Jika user memilih pilihan 3, maka program akan memanggil fungsi `'edit() ;'` untuk mengubah data, user akan diperintahkan untuk memasukkan nomor index data yang diubah, Nama, Umur, dan jenis kelamin.

Jika user memilih pilihan 4, maka program akan memanggil fungsi `'hapus() ;'` untuk menghapus data, user akan diperintahkan untuk memasukkan nomor index data yang ingin dihapus.

Jika user memilih pilihan 5, maka program akan memanggil fungsi `'simpan() ;'` untuk menyimpan data dan keluar dari program. Data yang telah dibuat akan disimpan sebagai `biodata.txt` dan terletak dalam folder yang sama dengan `'exe'` program tersebut.

Jika User memilih pilihan 6, maka program akan memanggil fungsi `'menu_baru() ;'` untuk memulai program penggabungan dua list yang merepresentasikan gabungan dari konsep struct, pointer dan single link list. Dimana akan ada dua list yang memiliki masing - masing 2 input dengan 3 data yang kemudian akan digabung bersamaan menjadi satu list. Sistem kerjanya adalah menggunakan list berulang dengan masing - masing memegang 3 variabel angka sebagai isi dari list tersebut, kemudian ada pointer yang menginisialisasi masing - masing list. Dan data ini akan sama - sama bertemu di pointer baru setelah pointer ekor dari list pertama bertemu dengan pointer kepala list kedua.

## 3.2. Tampilan Program

### 3.2.1. Menu Awal

```
"C:\Users\Firyanul Rizky\Desktop\Praktikum Pemrograman\Bahasa C\Program Operasi File (Baca,Tampil,Edit)_1708561006.exe"
Program Sederhana Operasi File
=====
+----- BIODATA MAHASISWA -----+
| [1] Tambah biodata Mahasiswa      |
| [2] Tampil biodata Mahasiswa      |
| [3] Edit biodata Mahasiswa        |
| [4] Hapus biodata Mahasiswa       |
| [5] Simpan & Keluar               |
| [6] Penggabungan List dengan Pointer |
+-----+
Pilih : _
```

### 3.2.2. Menu 1 : Tambah Biodata Mahasiswa

```
"C:\Users\Firyanul Rizky\Desktop\Praktikum Pemrograman\Bahasa C\Program Operasi File (Baca,Tampil,Edit)_1708561006.exe"
Program Sederhana Operasi File
=====
+----- BIODATA MAHASISWA -----+
| [1] Tambah biodata Mahasiswa      |
| [2] Tampil biodata Mahasiswa      |
| [3] Edit biodata Mahasiswa        |
| [4] Hapus biodata Mahasiswa       |
| [5] Simpan & Keluar               |
| [6] Penggabungan List dengan Pointer |
+-----+
Pilih : 1
Masukan Nama Mahasiswa : Muhammad Firyanul Rizky
Masukan usia : 19
Masukan Jenis Kelamin : Laki - Laki
Kembali Ke Menu ?
[1] Ya
[2] Tidak
Ketik Pilihan : _
```

### 3.2.3. Menu 2 : Tampil Biodata Mahasiswa

```
"C:\Users\Firyanul Rizky\Desktop\Praktikum Pemrograman\Bahasa C\Program Operasi File (Baca,Tampil,Edit)_1708561006.exe"
Program Sederhana Operasi File
=====
+----- BIODATA MAHASISWA -----+
| [1] Tambah biodata Mahasiswa      |
| [2] Tampil biodata Mahasiswa      |
| [3] Edit biodata Mahasiswa        |
| [4] Hapus biodata Mahasiswa       |
| [5] Simpan & Keluar               |
| [6] Penggabungan List dengan Pointer |
+-----+
Pilih : 2
0 Muhammad Firyanul Rizky 19 Laki - Laki
Kembali Ke Menu ?
[1] Ya
[2] Tidak
Ketik Pilihan : _
```

### 3.2.4. Menu 3 : Edit Biodata Mahasiswa

```
"C:\Users\Firyanul Rizky\Desktop\Praktikum Pemrograman\Bahasa C\Program Operasi File (Baca,Tampil,Edit)_1708561006.exe"
Program Sederhana Operasi File
=====
+----- BIODATA MAHASISWA -----+
| [1] Tambah biodata Mahasiswa      |
| [2] Tampil biodata Mahasiswa      |
| [3] Edit biodata Mahasiswa        |
| [4] Hapus biodata Mahasiswa       |
| [5] Simpan & Keluar               |
| [6] Penggabungan List dengan Pointer |
+-----+

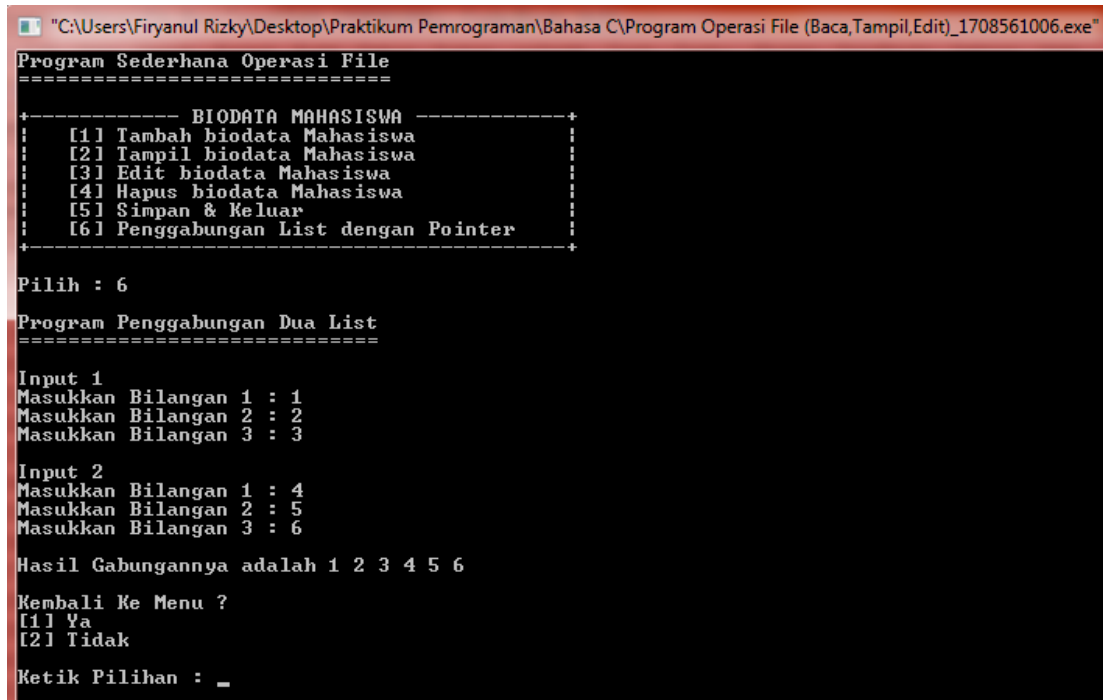
Pilih : 3
Masukan Index biodata Mahasiswa : 0
Masukan Nama Mahasiswa : FiryaNULL
Masukan usia : 18
Masukan Jenis Kelamin : Pria
Kembali Ke Menu ?
[1] Ya
[2] Tidak
Ketik Pilihan : _
```

### 3.2.5. Menu 4 : Hapus Biodata Mahasiswa

```
"C:\Users\Firyanul Rizky\Desktop\Praktikum Pemrograman\Bahasa C\Program Operasi File (Baca,Tampil,Edit)_1708561006.exe"
Program Sederhana Operasi File
=====
+----- BIODATA MAHASISWA -----+
| [1] Tambah biodata Mahasiswa      |
| [2] Tampil biodata Mahasiswa      |
| [3] Edit biodata Mahasiswa        |
| [4] Hapus biodata Mahasiswa       |
| [5] Simpan & Keluar               |
| [6] Penggabungan List dengan Pointer |
+-----+

Pilih : 4
Masukan Index Yang Ingin Dihapus : 0
Kembali Ke Menu ?
[1] Ya
[2] Tidak
Ketik Pilihan :
```

### 3.2.6. Menu 6 : Penggabungan List dengan Pointer



```
"C:\Users\Firyanul Rizky\Desktop\Praktikum Pemrograman\Bahasa C\Program Operasi File (Baca,Tampil,Edit)_1708561006.exe"
Program Sederhana Operasi File
=====
+----- BIODATA MAHASISWA -----+
| [1] Tambah biodata Mahasiswa    |
| [2] Tampil biodata Mahasiswa    |
| [3] Edit biodata Mahasiswa      |
| [4] Hapus biodata Mahasiswa     |
| [5] Simpan & Keluar             |
| [6] Penggabungan List dengan Pointer |
+-----+

Pilih : 6

Program Penggabungan Dua List
=====

Input 1
Masukkan Bilangan 1 : 1
Masukkan Bilangan 2 : 2
Masukkan Bilangan 3 : 3

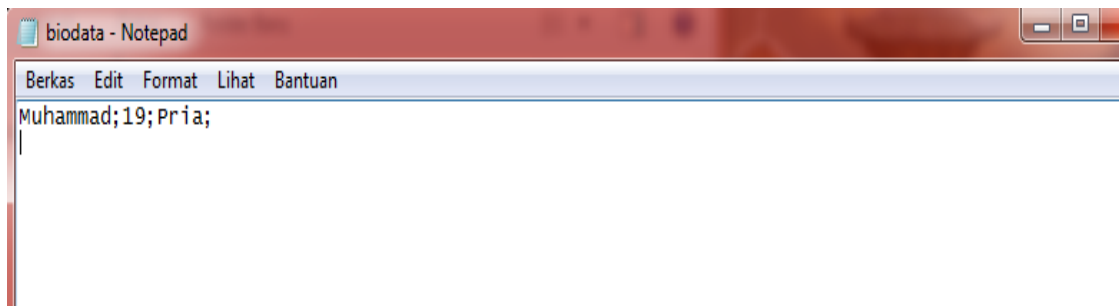
Input 2
Masukkan Bilangan 1 : 4
Masukkan Bilangan 2 : 5
Masukkan Bilangan 3 : 6

Hasil Gabungannya adalah 1 2 3 4 5 6

Kembali Ke Menu ?
[1] Ya
[2] Tidak

Ketik Pilihan : _
```

### 3.2.7. Isi dari Biodata.txt



```
biodata - Notepad
Berkas Edit Format Lihat Bantuan
Muhammad;19;Pria;
|
```

### 3.3. Sysntax Program Keseluruhan

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

struct biodata {
    char nama[100];
    int usia;
    char gender[10];
};

struct biodata a[100];
int index=0;

int tambah()
{
    printf("Masukan Nama Mahasiwa : ");
    scanf("%[^\\n]",a[index].nama);fflush(stdin);
    printf("Masukan usia      : ");
    scanf("%d",&a[index].usia);fflush(stdin);
    printf("Masukan Jenis Kelamin : ");
    scanf("%[^\\n]",a[index].gender);fflush(stdin);
    index++;
    kembali();
}

int edit()
{
    int keindex;
    printf ("Masukan Index biodata Mahasiswa : ");
    scanf ("%d",&keindex);fflush(stdin);
    printf ("Masukan Nama Mahasiswa : ");
    scanf ("%[^\\n]",a[keindex].nama);
    printf ("Masukan usia      : ");
    scanf ("%d",&a[keindex].usia);fflush(stdin);
    printf ("Masukan Jenis Kelamin : ");
    scanf ("%[^\\n]",a[keindex].gender);
    kembali();
}

int hapus()
{
    int keindex,i;
    printf ("Masukan Index Yang Ingin Dihapus : ");
    scanf ("%d",&keindex);
    for (i=keindex;i<index;i++)
    {
        a[i]=a[i+1];
    }
    index--;kembali();
}

int simpan()
```

```

{
    int i,pilih;
    FILE *fp;
    fp=fopen("biodata.txt","w");
    for (i=0;i<index;i++)
    {
        fprintf(fp,"%s;%d;%s;\n",a[i].nama,a[i].usia,a[i].gender);
    }
    fclose(fp);
    printf("Baca File dulu ?\n");
    scanf("%d",pilih);
    if(pilih==1)
    {
        FILE *fp;
        fp=fopen("biodata.txt","r");
        while
(fscanf(fp,"%[^;];%d;%[^;];\n",a[index].nama,&a[index].usia,a[index]
.gender)!=EOF)
        {
            index++;
        }
        fclose(fp);
    }
    else {
        printf("Input Salah !\n");
    }
    kembali();
}

int tampil()
{
    int pil;
    int i;
    for (i=0;i<index;i++)
    {
        printf          ("%d          %s          %d
%s\n",i,a[i].nama,a[i].usia,a[i].gender);
    }
    kembali();
}

struct node {
    int data;
    struct node *next;
}*head,*head1,*temp,*tail,*tail1;

int menu_baru(){
    int pil,i,bil;
    printf("Program Penggabungan Dua List\n");
    printf("=====\n\n");
    printf("Input 1\n");
    for(i=1;i<=3;i++){
        temp=(struct node *)malloc(sizeof(struct node));
        printf("Masukkan Bilangan %d : ", i);
    }
}

```



```

        scanf("%d", &bil);
        if(head==NULL) {
            tail=head=temp;
            head->data=bil;
            head->next=NULL;
        }
        else{
            temp->data=bil;
            tail->next=temp;
            tail=temp;
            tail->next=NULL;
        }
    }

    printf("\nInput 2\n");
    for(i=1;i<=3;i++){

        temp=(struct node *)malloc(sizeof(struct node));

        printf("Masukkan Bilangan %d : ", i);
        scanf("%d", &bil);
        if(head1==NULL) {
            tail1=head1=temp;
            head1->data=bil;
            head1->next=NULL;
        }
        else{
            temp->data=bil;
            tail1->next=temp;
            tail1=temp;
            tail1->next=NULL;
        }
    }
    tail->next=head1;
    struct node *a;
    a=head;
    printf("\nHasil Gabungannya adalah ");
    while(a!=NULL)
    {
        printf("%d ",a->data);
        a=a->next;
    }
    printf("\n");
    kembali();
}

int kembali()
{
    int pil;
    printf("\nKembali Ke Menu ?\n");
    printf("[1] Ya\n");
    printf("[2] Tidak\n\n");
    printf("Ketik Pilihan : ");
    scanf("%d",&pil);

```

```

        if(pil==1)
        {
            main();
        }
        else {
            system("cls");
            printf("\nTerimakasih !\n");
        }
    }

int main()
{
    kembali:
    system("cls");
    int menu;
    printf("Program Sederhana Operasi File\n");
    printf("=====\n\n");
    printf("+----- BIODATA MAHASISWA -----+\n");
    printf("|    [1] Tambah biodata Mahasiswa      |\n");
    printf("|    [2] Tampil biodata Mahasiswa      |\n");
    printf("|    [3] Edit biodata Mahasiswa        |\n");
    printf("|    [4] Hapus biodata Mahasiswa       |\n");
    printf("|    [5] Simpan & Keluar                |\n");
    printf("|    [6] Penggabungan List dengan Pointer |\n");
    printf("+-----+\n");
    printf ("\nPilih : ");
    scanf ("%d",&menu); fflush(stdin);
    printf ("\n");
    switch(menu)
    {
        case 1: tambah();break;
        case 2: tampil();break;
        case 3: edit();break;
        case 4: hapus();break;
        case 5: simpan();break;
        case 6: menu_baru();break;
    }
    if(menu!=6)
    {
        printf("Pilihan Salah !\n");
        return 0;
    }
}

```