

# PRAKTIKUM 14

## OPERASI FILE

### 14.1 TUJUAN PRAKTIKUM

#### Tujuan Umum

Mahasiswa dapat memahami operasi file pada bahasa pemrograman C

#### Tujuan Khusus

Mahasiswa dapat :

1. Memahami Implementasi read, write dan append file pada bahasa C
2. Memahami Implementasi algoritma kriptografi klasik menggunakan media file pada bahasa C

### 14.2 PELAKSANAAN PRAKTIKUM

Penggunaan operasi FILE dalam sebuah program kadangkala sangat dibutuhkan dalam sebuah program yang sesungguhnya. Kita membutuhkan file sebagai tempat penampung data-data selama operasi program. Jika selama ini kita menggunakan variabel biasa, data yang biasanya diolah program hanya akan tersimpan sementara dalam memory dan akan hilang ketika program berhenti atau ditutup. Berbeda dengan memory, penyimpanan data berbasis file akan tersimpan terus walaupun program telah ditutup maupun komputer telah dimatikan.

Pada dasarnya, operasi file terbagi 3 jenis. Read (R), Write(W), dan Append(A). Read artinya membaca isi file, write menulis data ke file dan append menambahkan data ke baris terakhir dalam file.

Saat berurusan dengan file, ada dua jenis file yang harus diketahui yakni:

1. File teks
2. File biner

File teks adalah file .txt normal yang dapat Anda buat dengan mudah menggunakan Notepad atau editor teks sederhana. Saat Anda membuka file-file itu, Anda akan melihat semua konten di dalam file sebagai teks biasa. Anda dapat dengan mudah mengedit atau menghapus konten. Karena sifatnya yang mudah dibaca maka file teks memberikan keamanan paling sedikit dan membutuhkan ruang penyimpanan yang lebih besar.

File biner sebagian besar adalah file .bin di komputer Anda. Alih-alih menyimpan data dalam teks biasa, file menyimpannya dalam bentuk biner (0 dan 1). Keuntungan file biner adalah dapat menyimpan jumlah data yang lebih banyak, tidak mudah dibaca dan memberikan keamanan yang lebih baik daripada file teks.

#### 14.2.1 Struktur Dasar

```
main(){
    FILE *in=fopen(namafile,mode);
    // Pengolahan teks
    fclose(in);
}
```

- FILE \*in=fopen(namafile, mode); Langkah pertama kita wajib membuka file terlebih dahulu dengan perintah fopen. Fungsi fopen diikuti dengan 2 variabel diantaranya namafile dan mode. Nama file bebas kamu tentukan sendiri.

- Setelah dibuka, file tersebut wajib kita close dengan perintah `fclose(namavariabel)`. Karena diawal kita membuka file lalu menyimpannya dalam variabel pointer “in”, maka diakhir program juga wajib menutup file dengan `fclose(in)`;
- Pengolahan teks meliputi didalamnya pemanggilan fungsi lanjutan tergantung mode pada `fopen`.

Berikut ini mode-mode yang bisa digunakan :

File Mode	Meaning of Mode	During Inexistence of file
r	Open for reading.	If the file does not exist, <code>fopen()</code> returns NULL.
rb	Open for reading in binary mode.	If the file does not exist, <code>fopen()</code> returns NULL.
w	Open for writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
wb	Open for writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a	Open for append. i.e, Data is added to end of file.	If the file does not exists, it will be created.
ab	Open for append in binary mode. i.e, Data is added to end of file.	If the file does not exists, it will be created.
r+	Open for both reading and writing.	If the file does not exist, <code>fopen()</code> returns NULL.
rb+	Open for both reading and writing in binary mode.	If the file does not exist, <code>fopen()</code> returns NULL.
w+	Open for both reading and writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
wb+	Open for both reading and writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a+	Open for both reading and appending.	If the file does not exists, it will be created.
ab+	Open for both reading and appending in binary mode.	If the file does not exists, it will be created.

**14.2.2 Menulis dan Membaca File Teks**

Untuk membaca dan menulis ke file teks, kita dapat menggunakan fungsi `fprintf()` dan `fscanf()` yang merupakan versi file `printf ()` dan `scanf ()`. Satu-satunya perbedaan adalah bahwa, `fprint` dan `fscanf` menggunakan pointer ke struktur `FILE`.

### Praktikum 14.1 Menulis File Teks

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  main() {
5      int num;
6      FILE *fptr;
7      fptr = fopen("program.txt", "w");
8      if (fptr == NULL)
9      {
10         printf("Error!");
11         exit(1);
12     }
13     printf("Input Nilai: ");
14     scanf("%d", &num);
15     fprintf(fptr, "%d", num);
16     fclose(fptr);
17     printf("Nilai telah tersimpan di file teks\n");
18     printf("Silahkan cek file program.txt");
19 }
```

### Praktikum 14.2 Membaca File Teks

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  main() {
5      int num;
6      FILE *fptr;
7      if ((fptr = fopen("program.txt", "r")) == NULL){
8          printf("Error!");
9          // program berhenti jika file tidak ditemukan.
10         exit(1);
11     }
12     fscanf(fptr, "%d", &num);
13     printf("Isi file teks : %d", num);
14     fclose(fptr);
15 }
```

#### 14.2.2 Menulis dan Membaca File Biner

Untuk membaca dan menulis ke file biner, kita dapat menggunakan fungsi `fwrite()` dan `fread()`. Ada 4 argumen yang digunakan pada fungsi-fungsi tsb, alamat data, ukuran data, jumlah tipe data dan pointer file.

```
fwrite(address_data, size_data, numbers_data, pointer_to_file);
fread(address_data, size_data, numbers_data, pointer_to_file);
```

### Praktikum 14.3 Menulis File Biner

```
1  #include<stdio.h>
2
3  struct rec {
4      int x,y,z;
5  };
6
7  main() {
8      int counter;
9      FILE *ptr_myfile;
10     struct rec my_record;
11
12     ptr_myfile=fopen("test.bin","wb");
13     if (!ptr_myfile) {
14         printf("Unable to open file!");
15         return 1;
16     } else {
17         for ( counter=1; counter <= 10; counter++) {
18             my_record.x= counter;
19             my_record.y= counter+1;
20             my_record.z= counter+counter;
21             fwrite(&my_record, sizeof(struct rec), 1, ptr_myfile);
22         }
23         printf("Berhasil menulis file biner");
24     }
25     fclose(ptr_myfile);
26
27 }
```

Untuk membaca file biner dapat menggunakan powershell di windows dengan mengetikkan “format-hex nama\_file”, seperti contoh dibawah ini.

```
PS D:\STSN\AJAR\SDAP\Pertemuan\Modul> format-hex test.bin

Path: D:\STSN\AJAR\SDAP\Pertemuan\Modul\test.bin

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  01 00 00 00 02 00 00 00 02 00 00 00 02 00 00 00  .....
00000010  03 00 00 00 04 00 00 00 03 00 00 00 04 00 00 00  .....
00000020  06 00 00 00 04 00 00 00 05 00 00 00 08 00 00 00  .....
00000030  05 00 00 00 06 00 00 00 0A 00 00 00 06 00 00 00  .....
00000040  07 00 00 00 0C 00 00 00 07 00 00 00 08 00 00 00  .....
00000050  0E 00 00 00 08 00 00 00 09 00 00 00 10 00 00 00  .....
00000060  09 00 00 00 0A 00 00 00 12 00 00 00 0A 00 00 00  .....
00000070  0B 00 00 00 14 00 00 00                                .....
```

## Praktikum 14.4 Membaca File Biner

```
1  #include<stdio.h>
2
3  struct rec {
4      int x,y,z;
5  };
6
7  main() {
8      int counter;
9      FILE *ptr_myfile;
10     struct rec my_record;
11
12     ptr_myfile=fopen("test.bin","rb");
13     if (!ptr_myfile) {
14         printf("Unable to open file!");
15         return 1;
16     } else {
17         for ( counter=1; counter <= 10; counter++) {
18             fread(&my_record, sizeof(struct rec), 1, ptr_myfile);
19             printf("x: %d\ty: %d\tz: %d\n", my_record.x, my_record.y, my_record.z);
20         }
21     }
22     fclose(ptr_myfile);
23 }
```

### Latihan

Buatlah program implementasi vigenere cipher untuk men-enkripsi dan men-dekripsi file.