

OPERASI FILE

Mira Suryani, S.Pd., M.Kom
Jatinangor, 13 Desember 2018

S-1 Teknik Informatika



From West Java for Indonesia to the World through SDGs

www.unpad.ac.id



Tujuan

- Mahasiswa mampu memahami tentang tipe data file baik file teks maupun file biner (binary file)
- Mampu membuat program yang terstruktur dan modular menggunakan file untuk menyimpan data ke eksternal storage



Operasi File

- Dalam bahasa pemrograman C++ dapat dilakukan operasi file dengan menggunakan ifstream, ofstream, dan fstream.
 - **ifstream** adalah kelas file stream yang memiliki fungsionalitas **untuk membaca file**. Digunakan pada saat file hanya untuk dibaca saja, tidak untuk ditulis.
 - **ofstream** adalah sebuah kelas file stream yang memiliki fungsionalitas **untuk menulis kedalam file**. Digunakan pada saat file hanya untuk ditulis saja, tidak untuk dibaca.
 - **fstream** adalah sebuah kelas file stream yang memiliki fungsionalitas untuk **membaca dan menulis kedalam file**. Digunakan pada saat file kedua fungsionalitas membaca dan menulis dibutuhkan.



Fungsi-fungsi untuk operasi file:

<code>ios::in</code>	If <i>FileName</i> is a new file, then it gets created fine as an empty file. If <i>FileName</i> already exists, then it is opened and its content is made available for processing
<code>ios::out</code>	If <i>FileName</i> is a new file, then it gets created fine as an empty file. Once/Since it gets created empty, you can write data to it. If <i>FileName</i> already exists, then it is opened, its content is destroyed, and the file becomes as new. Therefore you can create new data to write to it. Then, if you save the file, which is the main purpose of this mode, the new content is saved it.*This operation is typically used when you want to save a file
<code>ios::binary</code>	Open dalam mode binary.
<code>ios::ate</code>	If <i>FileName</i> is a new file, data is written to it and subsequently added to the end of the file. If <i>FileName</i> already exists and contains data, then it is opened and data is written in the current position
<code>ios::app</code>	If <i>FileName</i> is a new file, <u>data</u> is written to it. If <i>FileName</i> already exists and contains data, then it is opened, the compiler goes to the end of the file and adds the new data to it.
<code>ios::trunc</code>	If <i>FileName</i> already exists, its content is destroyed and the file becomes as new



Operasi File Text

- File Teks : file yang pola penyimpanan datanya dalam bentuk karakter.
- Dipakai untuk menyimpan data seperti karakter atau string.

Operasi dasar pada file pada prinsipnya terbagi menjadi 3 tahap, yaitu:

- membuka atau mengaktifkan file (Open File)
- melaksanakan pemrosesan file
- menutup file (Close File)



A. Membuka file (Open File)

- Sebelum suatu file dapat diproses, file harus dibuka terlebih dahulu. Sebelum file dibuka, terlebih dahulu obyek file harus didefinisikan.

- Sintaksnya:

```
ofstream nama_obyek;
```

- perintah ofstream dapat dijalankan dengan menyertakan file header `fstream.h`
- Setelah itu, suatu file dapat dibuka dengan perintah

```
nama_obyek open("nama file dan path");
```



B. Menulis ke File

- Sintaknya:
 - *nama_obyek << ... {isian data yang akan disimpan ke File}*

C. Menutup File (Close File)

- Setelah pemrosesan file selesai, file dapat ditutup menggunakan perintah
- Sintaknya:
 - *nama_obyek.close();*



Contoh fungsi operasi file teks.

- *void bukaTulisIsiFile(char nFile[]);*
- *void tambahIsiFile(char nFile[], char isi[]);*
- *void periksaFile(char nFile[]);*
- *void cetakIsiFilePerKarakter(char nFile[]);*
- *void isiFilePerKarakter(char nFile[]);*
- *int cariJumlahHurufA(char nFile[]);*



Contoh 1.

Program berikut ini untuk menulis teks ke dalam file

```
void bukaTulisIsiFile(char nFile[]){
    ofstream fileteks;
    fileteks.open(nFile);
    fileteks << "Selamat Datang Mahasiswa Teknik Informatika
                Unpad "<< endl;
    fileteks << "dalam Mata Kuliah Algoritma dan
                Pemrograman"<< endl;
    fileteks << "Semester 1 th ajaran baru" << endl;
    fileteks << "Selamat Belajar dan Semoga Sukses " << endl;
    fileteks.close();
}
```

- Perintah *fileteks.open(nFile)*; akan membuka file berdasarkan nama File dari parameter Fungsi yang diberikan oleh pemanggil fungsi misalkan nama File = "algoritma.txt"
- Apabila file tersebut belum ada maka akan dibuat secara otomatis, dan apabila sudah ada, isi file algoritma.txt akan terhapus.



D. Menambah Data pada File

- Suatu file yang sudah ada sebelumnya dapat ditambah dengan data yang baru (tidak menghapus data lama). Caranya dengan menambahkan perintah `ios::app` pada `open()`.
 - `nama_objek.open("nama file", ios::app);`

```
void tambahIsiFile(char nFile[], char isi[]){
    ofstream fileteks;
    fileteks.open(nFile, ios::app);
    fileteks << endl;
    fileteks << "Oleh: Akmal dan Mira"<< endl;
    fileteks << isi << endl;

    fileteks.close();
}
```

- Penambahan data bisa langsung di dalam fungsi atau bisa diberikan melalui nilai parameter dari fungsi

```
fileteks << "Oleh: Akmal"<< endl;
```

```
fileteks << isi << endl;
```



E. Memeriksa Keberhasilan Operasi File

- Jika file dibuka maka ada kemungkinan ternyata file tidak ada. Dalam C++ tersedia function untuk memeriksa kondisi-kondisi pada operasi file, sehingga kesalahan saat eksekusi dapat dikendalikan.
- Function yang dimaksud adalah fail().

```
void periksaFile(char nFile[]){
    ifstream fileteks;           // u/ membaca file
    fileteks.open(nFile);
    if (fileteks.fail())
        cout<< "File tak dapat dibuka / tidak ditemukan";
    else
        cout << "File ditemukan " << endl;

    fileteks.close();
}
```



F. Operasi Berbasis Karakter

- Operasi file dapat dilakukan dalam bentuk karakter.
- Misalnya proses penyimpanan data ke file dilakukan setiap karakter, atau membaca data file karakter per karakter. Operasi ini didukung oleh function **put()** dan **get()**.

Contoh 4:

- Program untuk menyimpan data karakter per karakter ke dalam file.

```
void isiFilePerKarakter(char nFile[]) {  
    ofstream fileteks;  
    fileteks.open(nFile);  
    fileteks.put('A');  
    fileteks.put('B');  
    fileteks.put('C');  
    fileteks.close();  
}
```



Contoh 5.

Program untuk membaca file karakter per karakter sekaligus menampilkan semua isi file

```
void cetakIsiFilePerKarakter(char nFile[]) {
    char karakter;
    ifstream fileteks;
    fileteks.open(nFile);
    while(!fileteks.eof())
    {
        fileteks.get(karakter);
        cout << karakter;
    }
    fileteks.close();
}
```



Contoh 6.

Fungsi berikut digunakan untuk menghitung secara sequensial banyak huruf 'A' yang terdapat dalam file text

```
int cariJumlahHurufA(char nFile[]){
    char karakter;
    int jumlahA=0;
    ifstream fileteks;
    fileteks.open(nFile);
    while(!fileteks.eof())
    {
        fileteks.get(karakter);
        if (karakter == 'A' || karakter == 'a')
            jumlahA=jumlahA+1;
    }
    fileteks.close();
    return jumlahA;
}
```



Pemanggilan dari program utama sebagai berikut :

```
int main(){
    char namaFile[]="algoritma.txt";
    char isi[]="Motto : Hari ini harus lebih baik dari
                hari kemaren ";
    bukaTulisIsiFile(namaFile);
    tambahIsiFile(namaFile,isi);
    periksaFile(namaFile);
//    isiFilePerKarakter(namaFile);
    cetakIsiFilePerKarakter(namaFile);
    cout << "Jumlah huruf A = " <<
        cariJumlahHurufA(namaFile);
    return 0;
}
```



Isi file “algoritma.txt” yang dilihat melalui Notepad adalah sebagai berikut :

```
File Edit Format View Help
Selamat Datang Mahasiswa Teknik Informatika Unpad
dalam Mata Kuliah Algoritma dan Pemrograman
Semester 1 th ajaran baru
Selamat Belajar dan Semoga Sukses

Oleh: Akmal
Motto : Hari ini harus lebih baik dari hari kemaren
```



Latihan

1. Buatlah program untuk menghitung jumlah karakter vocal dan konsonan dalam suatu file. Inputnya adalah nama file text dan pathnya.
2. Buatlah program C++ untuk menghitung jumlah karakter tertentu, misalnya karakter 'A'. Input berupa nama file dan karakter yang akan dihitung.
3. Misalkan suatu file teks berisi listing program C++. Buatlah program untuk menghitung pasangan kurung kurawal yang ada pada file teks tersebut.



Latihan

4. Buatlah program C++ untuk melakukan enkripsi shift chipper suatu file teks (dengan asumsi semua karakter huruf adalah huruf kapital). Inputnya adalah file teks yang akan dienkripsi dan besar pergeseran (integer). Outputnya adalah file teks hasil enkripsi.

Hint:

Ide dasar shift chipper adalah mengubah setiap karakter huruf ke karakter huruf lain. Misalkan pergeserannya 2, maka berikut ini karakter hasil enkripsi :

awal A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

hasil C D E F G H I J K L M N O P Q R S T U V W X Y Z A B

Sehingga misal diberikan suatu teks

C++ IS EASY,

maka hasil enkripsinya adalah

E++ KU GCUA



Operasi File Biner (Binary File)

- File Biner : file yang pola penyimpanan di dalam disk berbentuk biner, yaitu seperti bentuk pada memori RAM (komputer). Dipakai untuk menyimpan data kompleks, misal : struct.
- Contoh : Mengelola Data Mahasiswa ke dalam File

```
#include <iostream>
#include <fstream>
#include <conio.h>
#include <string.h>

using namespace std;
void isiFileMhs(char nFile[],int& n);
void cetakFileMhs(char nFile[], int n);
void searchingFileMhs(char nFile[], int n);
struct mahasiswa {
    char npm[10];
    char nama[50];
};

int main() {
    char namaFile[]="mahasiswa.dat";
    int nData;

    isiFileMhs(namaFile,nData);
    cetakFileMhs(namaFile,nData);

}
```



```
void isiFileMhs(char nFile[],int& n){
    // kode dibawah ini adalah untuk write data ke dalam file.

    mahasiswa mhs;

    ofstream fo;                // menulis

    cout << "Masukkan jumlah mahasiswa: ";
    cin >> n;

    fo.open(nFile, ios::binary | ios::app);
    for(int i=1; i<=n; i++) {
        cout << "\nMahasiswa " << i << endl;
        cout << "NPM   : ";   cin >> mhs.npm;
        cout << "Nama  : ";   cin >> mhs.nama;
        fo.write((char *) &mhs, sizeof(mhs)); // menulis ke file
    }
    fo.close();
}
```



```
void cetakFileMhs(char nFile[], int n){
    // * kode dibawah ini adalah untuk read data dari file.

    mahasiswa mhs;

    ifstream fi;                                // membaca

    fi.open(nFile, ios::binary);
    int j = 1;
    while(fi.read((char *) &mhs, sizeof(mhs))) {
        cout << "\nMahasiswa: " << j++;
        cout << "\nNPM: " << mhs.npm;
        cout << "\nNama: " << mhs.nama;
    }
    fi.close();
}
```

mahasiswa - Notepad

File Edit Format View Help

123 ali ÀÑ@ °p(Øp(Äÿ(eÂ¹wr2šüpyÿÿZ#LwYüwÀÑ@ Øp(234 susi ÀÑ@ °p(Øp(Äÿ(eÂ¹w ^



**ANY
QUESTIONS?**



Sesi Berakhir
TERIMA KASIH