

# FILE & STREAM

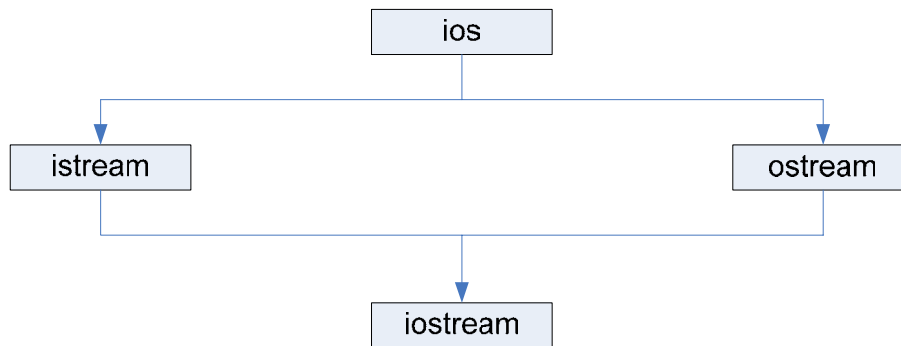
## Bab 5

### 5.1 FILE

File adalah sekumpulan data yang disimpan dalam media penyimpanan luar seperti disket/harddisk. Dalam C++ file adalah sebuah stream yang disimpan dalam media penyimpanan luar. Karena merupakan sebuah stream, operasi yang berlaku pada stream berlaku juga untuk file.

Stream adalah suatu logika device yang menghasilkan dan menerima informasi atau wadah yang digunakan untuk menampung keluaran dan menampung aliran data.

Hirarki I/O class :



Penjelasan :

- ios adalah virtual base class untuk class istream dan ostream. Berisi fasilitas dasar untuk melakukan proses input/output. Dideklarasikan untuk pointer ke buffer untuk tempat penyimpanan data sementara.
- Istream ( input stream ) mendefinisikan fasilitas untuk melakukan input suatu informasi. Di dalamnya didefinisikan fungsi `get ( )`, `getline ( )`, extractor operator `>>`
- Ostream ( output stream ) mendefinisikan fasilitas untuk melakukan setting terhadap output.
- Iostream : berisi semua fasilitas dari ios, istream, ostream ditambah beberapa fungsi untuk menyempurnakan kerja dari fungsi yang dideklarasikan pada base class.

Untuk melakukan proses file I/O, diperlukan file header `fstream.h` didalam program. Dalam file `fstream.h` didefinisikan beberapa class/object yang berhubungan dengan pemrosesan file, yaitu : `ifstream`, `ofstream`, `fstream` yang diturunkan dari `istream` dan `ostream`.

## 5.2 KETERHUBUNGAN STREAM DENGAN FILE

Jika mendeklarasikan suatu stream, dapat menghubungkan stream tersebut dengan file, dimana proses ini berhubungan dengan operasi terhadap file.

Tiga proses utama dalam mengelola file adalah :

1. membuka file
2. melakukan proses terhadap file
3. menutup file

Sebelum membuka file, harus mengetahui keadaan mode filenya. Keadaan yang perlu diketahui adalah :

1. untuk membuka file dengan tujuan output, digunakan ofstream.
2. untuk membuka file dengan tujuan input, digunakan ifstream.
3. untuk membuka file dalam keadaan input maupun output, gunakan fstream.

Fungsi open ( ) digunakan untuk membuka file.

**Bentuk umumnya adalah :**

Void open (char\* file\_name, int mode, int access) ;

Keterangan :

- dengan fungsi open ( ), menghubungkan stream dengan file yang bernama file\_name.
- nilai dari var mode akan menentukan bagaimana keadaan file jika dibuka.
- Variable access akan menentukan bagaimana metode pengaksesan terhadap file tersebut. Nilai ini berhubungan dengan atribut file yang ada pada DOS.

Nilai pada variable mode adalah :

1. ios :: app  
semua informasi yang ditulis ke dalam file (output) akan ditambahkan dibagian akhir file tersebut.
2. ios :: ate  
file akan dibuka dengan pointer file menunjuk pada akhir file.
3. ios :: in  
file akan dibuka sebagai input.
4. ios :: out  
file akan dibuka sebagai output
5. ios :: nocreate  
akan mengakibatkan kesalahan jika file tidak ada.
6. ios :: noreplace  
file yang dibuka tidak dapat diganti, atau mengakibatkan kesalahan jika file yang akan dibuka sudah ada.
7. ios :: trunc  
menyebabkan isi dari file yang sudah ada akan hilang

Nilai pada variable access adalah :

0 : normal file            4 : system file  
1 : read only file        8 : archive bit-set file  
2 : hidden file

#### Contoh penggunaan open ( ) :

Akan dibuka sebuah file sebagai output, dimana atribut dari file tersebut adalah read only dan nama file tersebut adalah tes.

Jawab :

```
Ofstream fout ;  
Fout.open ("tes", ios : : out, 1) ;
```

Jika dalam pembukaan file terdapat kesalahan, maka fungsi open ( ) akan mengembalikan nilai NULL.

Fungsi close ( ) untuk menutup file yang telah dibuka.

#### **Bentuk umumnya :**

```
Void close ( ) ;
```

#### Contoh penggunaan close ( ) :

```
Fout.close ( ) ;  
Fin.close ( ) ;  
Fio. Close ( ) ;
```

Didalam class ios terdapat pendefinisian fungsi : int eof ( ) digunakan untuk menentukan apakah pointer pada file telah mencapai akhir dari file tersebut.

### **5.3 DETEKSI ERROR**

Dalam class ios ada 4 (empat) buah fungsi untuk melakukan tes terhadap error yang terjadi didalam stream. Fungsi tersebut adalah :

1. int good ( )
2. int eof ( )
3. int bad ( )
4. int fail ( )

Keempat fungsi itu akan menghasilkan nilai 0 (nol) jika kondisi tersebut salah dan nilai bukan nol jika kondisi tersebut benar.

Jika error terjadi, maka stream harus dibersihkan dahulu dari kesalahan sebelum melanjutkan ke proses berikutnya. Fungsi yang digunakan untuk membersihkan kesalahan pada stream jika terjadi kesalahan : `clear ( )`

**Bentuk umumnya :**

`Void clear (int flags = 0 ) ;`

Fungsi `clear ( )` dalam keadaan default akan membersihkan seluruh flag. Dapat ditentukan flag mana yang akan dibersihkan, dengan memberi argument yang sesuai.

Fungsi `int rdstate ( )` digunakan untuk menentukan jenis kesalahan yang terjadi, dengan mengembalikan nilai integer kesalahan tersebut.

Nilai enum yang dikembalikan oleh fungsi `rdstate ( )` adalah :

1. `goodbit`  
0 jika tidak ada error dan 1 jika terjadi error
2. `eofbit`  
1 jika eof ditemukan dan 0 jika eof tidak ditemukan
3. `failbit`  
1 jika non fatal error ditemukan dan 0 jika tidak
4. `badbit`  
1 jika fatal error ditemukan dan 0 jika tidak.

#### **5.4 INPUT/OUTPUT KARAKTER**

Input/output pada file binary digunakan fungsi `get ( )`, `put ( )`.

**Bentuk Umumnya :**

`Istream &get (char& ch) ;`  
`Ostream &put (char ch) ;`

Fungsi `get ( )` akan membaca sebuah karakter dari stream dan karakter tersebut akan diletakkan pada variable `ch`.

Fungsi ini akan mengembalikan reference ke stream jika EOF ditemukan, maka nilai yang dikembalikan `NULL`.

Fungsi `put ( )` akan meletakkan sebuah karakter ke suatu stream dan akan mengembalikan reference ke stream.

Untuk membaca dan menulis file binary dalam bentuk blok data dapat menggunakan fungsi `read ( )` dan `write ( )`

**Bentuk Umum :**

Istream& read (unsigned char\* buf, int num) ;  
Ostream& write (const unsigned char\* buf, int num);

Fungsi read ( ) akan membaca sejumlah num byte dari stream dan akan meletakkan data tersebut ke dalam variable yang ditunjukkan oleh pointer buf.

Fungsi write ( ) akan menulis sejumlah num byte ke stream dari data yang ditunjuk oleh pointer buf.

Fungsi yang akan mengembalikan karakter yang terakhir dibaca ke dalam memori lagi adalah : putback ( )

**Bentuk umum :**

Istream& putback (char ch) ;

Keterangan :

- variable ch adalah karakter yang dibaca yang akan dikembalikan lagi ke stream ( didalam memori)