

# **MODUL MATA KULIAH PRAKTIKUM PEMROGRAMAN LANJUT**

## **Modul 4: Akses File**



Dosen Pengampu:  
**Herryawan Pujiharsono, S.T., M.Eng.**  
**Slamet Indriyanto, S.T., M.T.**  
**Prasetyo Yuliantoro, S.T., M.T.**

**PROGRAM STUDI S1 TEKNIK TELEKOMUNIKASI  
FAKULTAS TEKNIK TELEKOMUNIKASI DAN ELEKTRO  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2019**

## I. TUJUAN

1. Menulis data ke dalam sebuah file
2. Membaca data dari sebuah file
3. Meng-*update* data dalam sebuah file
4. Meng-*edit* data dalam sebuah file

## II. DASAR TEORI

### A. LIBRARY AKSES FILE

Proses pengaksesan file di dalam bahasa pemrograman C++ dapat dilakukan dengan menambahkan library *fstream.h* pada bagian header. Library tersebut menyediakan tiga jenis *class* yang dapat digunakan untuk pengaksesan file, yaitu:

1. *ifstream* yang berfungsi untuk menangani pembacaan data dari sebuah file,
2. *ofstream* yang berfungsi untuk menangani penulisan data dari sebuah file
3. *fstream* yang berfungsi untuk menangani pembacaan dan penulisan data sekaligus.

*Class* tersebut digunakan untuk membuat sebuah *object* agar proses penanganan file dapat dilakukan. Proses pembuatan *object* hampir mirip dengan proses pendeklarasian variabel, yaitu sebagai berikut.

```
[ifstream/ofstream/fstream] [nama object];
```

Berikut adalah contoh proses pembuatan *object* bernama *file* dengan *class ifstream*:

```
ifstream file;
```

Setelah *object* dibuat, proses untuk pembacaan atau penulisan data dapat dilakukan dengan urutan langkah sebagai berikut:

1. Membuka file

Proses pembukaan file dapat dilakukan dengan perintah *open*("[nama file.ext]") dengan format sebagai berikut:

```
[nama object].open("[nama file]");
```

Contoh:

```
file.open("contoh.txt");
```

Perintah tersebut akan membuka file dengan nama *contoh.txt* pada folder dimana kode program dibuat. Jika file tersebut tidak ada, maka perintah tersebut akan membuatkan file dengan nama tersebut.

2. Melakukan pembacaan atau penulisan data

Proses pembacaan atau penulisan data dapat dilakukan setelah proses pembukaan file. Perintah yang digunakan untuk proses pembacaan dan penulisan file masing-masing akan dibahas pada subbab selanjutnya.

3. Menutup file

Proses penutupan file dilakukan dengan menggunakan perintah *close()* dengan format sebagai berikut.

```
[nama object].close();
```

Contoh:

```
file.close();
```

Perintah harus selalu ditambahkan di bagian akhir kode program sehingga file yang diakses dapat diakses oleh program yang lain.

## B. TULIS DATA KE FILE

Perintah yang digunakan untuk menuliskan data ke dalam sebuah file adalah *<<* dengan format sebagai berikut.

```
[nama object] << "[Isi data]";
```

Format tersebut hampir sama seperti pada *cout*, hanya output dari perintah tersebut tidak lagi di *command prompt* atau *console*, tetapi masuk dalam sebuah file. Contoh:

```
file << "Hello World";
```

Perintah tersebut akan memasukkan data berupa tulisan *Hello World* ke dalam sebuah file. *file* merupakan object yang dibuat dengan *class ofstream* dan tulisan *Hello World* dapat digantikan data yang lain, baik berupa karakter maupun angka.

## C. BACA DATA DARI FILE

Perintah yang digunakan untuk membaca data dari sebuah file adalah *getline* dengan format sebagai berikut.

```
[nama object].getline ([Varchar], [Panjang data], [karakter pemisah jika ada];
```

Contoh:

```
file.getline(baca, 25, '_');
```

Perintah tersebut akan membaca data yang ada di dalam sebuah file dan kemudian menampungnya dalam variabel bertipe **char** dengan nama *baca*. Panjang data yang dapat ditampung pada variabel *baca* dibatasi sampai 25 karakter (huruf atau angka) atau sampai menemukan karakter underscore ( \_ ).

Proses pembacaan data dalam sebuah file juga dapat dilakukan dimulai dari posisi karakter tertentu (tidak dari awal file) menggunakan perintah *seekg* dengan format sebagai berikut.

```
[nama object].seekg([posisi],ios::[beg/cur/end]);
```

Berdasarkan perintah tersebut, penentuan posisi dapat dilakukan dengan 3 cara, yaitu:

- ios::beg → penentuan posisi ditentukan dari depan/awal file
- ios::cur → penentuan posisi ditentukan dari posisi saat itu
- ios::end → penentuan posisi ditentukan dari belakang/akhir file

Contoh:

```
file.seekg(12,ios::beg);
```

Perintah tersebut membaca data mulai dari posisi 12 karakter dari depan. Jika isi file adalah sebagai berikut.

### ***Pemrograman Lanjut***

Maka 12 karakter dari dari depan sebagai berikut.

P	e	m	r	o	g	r	a	m	a	n		L	a	n	j	u	t	EOF
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

sehingga yang terbaca adalah:

***Lanjut***

## **D. UPDATE DATA**

Secara garis besar, proses update data sama dengan proses penulisan data ke dalam sebuah file, yaitu menggunakan perintah <<. Namun, yang membedakan adalah pada proses update data, data yang baru tidak menumpuki atau menghapus data yang sebelum sudah tersimpan pada file tersebut (*overwriting*). Untuk melakukan hal tersebut, terdapat modifikasi pada perintah untuk membuka data, yaitu *open*, dengan menambahkan parameter atau argumen tambahan di dalamnya, yaitu *ios::app*, sehingga format untuk perintah membuka file adalah sebagai berikut.

```
[nama object].open("[nama file]", ios::app);
```

Contoh:

```
file.open("contoh.txt", ios::app);
```

Argumen *ios::app* bertujuan untuk menempatkan posisi di bagian akhir file sehingga ketika terdapat data baru, data tersebut dimulai dari bagian akhir file sehingga data yang baru tidak menumpuki data lama tetapi malah menambahkan (*append*) data yang ada di dalam file.

## E. EDIT DATA

Prinsip dari proses edit data hampir sama dengan update data, yaitu menempatkan karakter pada suatu posisi di dalam file sehingga data yang baru akan dimulai dari posisi tersebut. Yang membedakan adalah pada edit data, posisi tidak ditempatkan pada bagian akhir file, tetapi di antara bagian awal dan akhir file. Untuk menempatkan posisi tersebut, digunakan perintah *seekp* dengan format sama seperti perintah *seekg*, yaitu sebagai berikut:

```
[nama object].seekp([posisi],ios::[beg/cur/end]);
```

Contoh:

```
file.seekp(12,ios::beg);
```

Perintah tersebut akan menempatkan data yang baru pada posisi karakter ke-12 dari depan. Data yang baru kemudian akan menumpuki karakter yang terdapat pada posisi 12 sampai sejumlah karakter pada data baru.

Jika data yang lama adalah sebagai berikut:

**Pemrograman Lanjut**

Dan data yang baru yang akan dimasukkan adalah:

**Prolan**

Maka isi data pada file tersebut menjadi sebagai berikut:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P	e	m	r	o	g	r	a	m	a	n		L	a	n	j	u	t	EOF
P	e	m	r	o	g	r	a	m	a	n		P	r	o	l	a	n	EOF

**Pemrograman Prolan**

### III. PERCOBAAN

**Catatan:** Buatlah sebuah folder dan simpan semua program tersebut dalam 1 folder yang sama karena file yang ditulis/dibaca juga tersimpan dalam folder tersebut.

#### 1. Latihan 1- Tulis data ke file

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      ofstream file;
8      char teks[50];
9      int i;
10
11     file.open("data.txt");
12     for (i=1; i<=3; i++)
13     {
14         cout << "Masukkan teks ke-" << i << " : ";
15         cin.getline(teks, 50);
16         file << teks << endl;
17     }
18     file.close();
19     return 0;
20 }
```

- Apa [*nama object*] yang digunakan untuk mengakses file pada kode program di atas?
- Apa fungsi dari *ofstream* yang terdapat pada baris 7?
- Sebutkan perintah apa saja yang digunakan pada kode program tersebut untuk menuliskan data ke dalam sebuah file dan apa fungsi dari masing-masing perintah tersebut?
- Apa nama file yang diakses pada kode program di atas dan apa ekstensi dari file tersebut?
- Gantilah ekstensi file tersebut dengan ekstensi yang lain seperti .xls, .dat, atau ekstensi buatan sendiri dan coba cek kembali dengan aplikasi yang sesuai apakah data berhasil dimasukkan ke dalam file tersebut!
- Simpulkan bagaimana proses menulis data ke sebuah file berdasarkan kode program tersebut?

## 2. Latihan 2 – Baca data dari file

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      ifstream file;
8      char Teks[50];
9      int i;
10
11     file.open("data.txt");
12     while (file.good())
13     {
14         file.getline (Teks,50,'\n');
15         cout << Teks << endl;
16     }
17     file.close();
18     return 0;
19 }
```

- Apa fungsi *ifstream* pada kode program baris ke-7?
- Sebutkan perintah apa saja yang digunakan pada kode program tersebut untuk membaca data dari sebuah file dan apa fungsi dari masing-masing perintah tersebut?
- Ubah file yang diakses pada kode program tersebut dengan file buatan sendiri yang dibuat pada latihan 1 dan cek kembali apakah data yang terbaca sama dengan data dimasukkan ke dalam file tersebut!
- Simpulkan bagaimana proses menulis data ke sebuah file berdasarkan kode program tersebut?

### 3. Latihan 3 – Baca data dari file (2)

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      ifstream file;
8      char Teks[50];
9      int i;
10
11     file.open("data.txt");
12
13     file.seekg(1, ios::beg);
14     file.getline (Teks, 50, '\n');
15     cout << Teks << endl;
16
17     file.seekg(6, ios::cur);
18     file.getline (Teks, 50, '\n');
19     cout << Teks << endl;
20
21     file.seekg(-4, ios::end);
22     file.getline (Teks, 50, '\n');
23     cout << Teks << endl;
24
25     file.close();
26     return 0;
27 }
```

- Apa maksud/fungsi dari `ios::beg` (baris 13), `ios::cur` (baris 17), dan `ios::end` (baris 21)?
- Ganti angka 1, 6 dan -4 pada baris tersebut dengan angka yang lain dan amati efek yang dihasilkan dari angka tersebut!
- Simpulkan apa fungsi dari `seekg` pada kode program tersebut!



#### 4. Latihan 4 – Update Data

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      ofstream file;
8      char teks[50];
9      int i;
10
11     file.open("data.txt", ios::app);
12     cout << "Masukkan data baru : ";
13     cin.getline (teks, 50);
14     file << teks << endl;
15     file.close();
16     return 0;
17 }
```

- Bandingkan hasil file yang dihasilkan antara kode program pada latihan 4 dengan latihan 1!
- Apakah terdapat perbedaan perintah yang digunakan untuk mengakses file antara kode program pada latihan 4 dengan latihan 1?
- Perintah apa yang menyebabkan file yang dihasilkan kode program latihan 4 berbeda dengan latihan 1?
- Simpulkan bagaimana proses update data berdasarkan kode program tersebut

## 5. Latihan 5 – Edit Data

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      fstream file;
8      char teks[50];
9      int pos;
10
11     file.open("data.txt");
12     cout << "Tentukan posisi yang akan diubah: ";
13     cin >> pos;
14     cout << "Masukkan data baru : ";
15     cin.ignore();
16     cin.getline(teks, 50);
17     file.seekp(pos);
18     file << teks << endl;
19
20     file.close();
21     return 0;
22 }
```

- Apa fungsi dari variabel `pos`?
- Apa fungsi dari perintah `seekp` dan apa bedanya dengan perintah `seekg` (latihan 3)?
- Apa fungsi `cin.ignore()` yang terdapat pada baris 15?
- Simpulkan bagaimana proses edit data berdasarkan kode program tersebut!

## IV. TUGAS

1. Rancanglah kode program yang mengkombinasikan tipe data *struct* dengan akses file sehingga menjadi sebuah database yang tersimpan dalam komputer dengan format nama file: **[NIM].[3 huruf nama pertama]** (Contoh: Nama Tony Stark, NIM: 17101789 → nama file database: 17101789.ton).

### Catatan:

- Perancangan terdiri dari notasi algoritma (pseudocode atau *flowchart*) dan kode program + hasil program

- **Field/Isi database (variabel di dalam tipe data *struct*) yang dibuat oleh setiap praktikan tidak boleh sama**

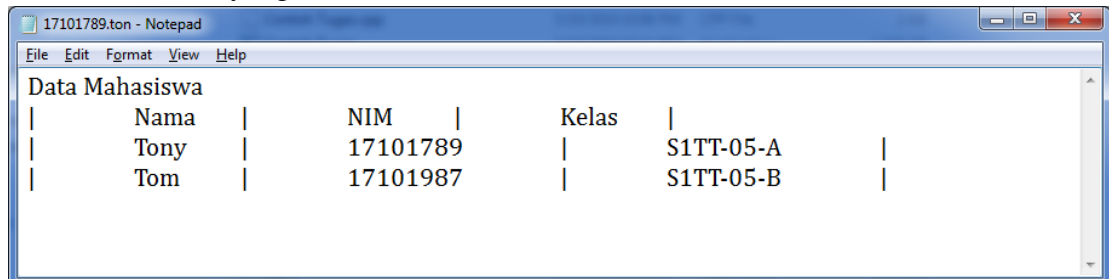
Contoh hasil program:



```
"E:\CodeBlocks\File 1\Contoh Tugas.exe"
Data ke-1
Nama : Tony
NIM : 17101789
Kelas : S1TT-05-A
Data ke-2
Nama : Tom
NIM : 17101987
Kelas : S1TT-05-B
Data Mahasiswa
|      Nama      |      NIM      |      Kelas      |
|      Tony      |      17101789  |      S1TT-05-A  |
|      Tom       |      17101987  |      S1TT-05-B  |

Process returned 0 (0x0)   execution time : 40.123 s
Press any key to continue.
```

Contoh hasil file yang dihasilkan:



```
17101789.ton - Notepad
File Edit Format View Help
Data Mahasiswa
|      Nama      |      NIM      |      Kelas      |
|      Tony      |      17101789  |      S1TT-05-A  |
|      Tom       |      17101987  |      S1TT-05-B  |
```