

TIF1202 – Pemrograman Berorientasi Objek

HO 04 - Operasi File I/O

Opim Salim Sitompul

Department of Information Technology
Universitas Sumatera Utara



Outline

- 1 Tujuan
- 2 Pendahuluan
- 3 Membuka File untuk Operasi Output
 - Writing Characters
 - Writing Alphabet at Random
 - Constructor dan Manipulator
- 4 Membuka File untuk Operasi Input
 - Menguji Keberhasilan File I/O
- 5 Operasi File Biner
 - Menulis File Biner
 - Membaca File Biner



Tujuan

- Dalam kuliah ini mahasiswa diharapkan:
 - Bagaimana membuka file
 - Bagaimana menutup file
 - Bagaimana memformat output
 - Bagaimana menguji keberhasilan operasi file
 - Membaca dan menulis file biner



Pendahuluan

- Apabila program membaca atau menuliskan informasi dari/ke file, program harus membuka file itu secara khusus dan harus pula dinyatakan dengan jelas apakah file itu hendak dibaca atau ditulis.
- Untuk operasi membaca dan menulis file digunakan kelas file stream.
- Untuk menggunakan file stream program menyertakan file header `fstream.h`



Pendahuluan

- Apabila program membuka file, harus diturunkan sebuah objek dari kelasnya berdasarkan jenis operasi I/O yang akan dilaksanakan.
 - Untuk operasi input: gunakan kelas **ifstream**
 - Untuk operasi output: gunakan kelas **ofstream**
- Contoh: mendeklarasikan sebuah objek berjenis **ofstream**
ofstream output_file;



Membuka File untuk Operasi Output

- Gunakan fungsi anggota `open` untuk membuka file:
 - `output_file.open("Namafile.ext");`
- Gunakan operator insertion untuk menuliskan output ke file:
 - `output_file << "Teks ini akan ditulis ke file" << endl;`
- Apabila tidak ada lagi output yang akan dituliskan, gunakan fungsi anggota **`close`** untuk menutup file:
 - `output_file.close();`



Membuka File untuk Operasi Output

```
1 //Contoh4_1.cpp
2 #include <fstream>
3 using namespace std;
4 class OutFile
5 {
6     private:
7         ofstream ofile;
8         char filename[80];
9     public:
10         OutFile(char fname[]) {strcpy(filename,
11                                     fname);}
12         void openFile() {ofile.open(filename);}
13         void writeFile();
14         void closeFile() {ofile.close();}
15         ~OutFile() {}
16 };
```



Membuka File untuk Operasi Output

```
16 void OutFile::writeFile()  
17 {  
18     ofile << "Nama:_Afry_Lianfauzan_Siregar" <<  
        endl;  
19     ofile << "NIM:_201402001" << endl;  
20     ofile << "Semester:_2" << endl;  
21     ofile << "Prog._Studi:_Teknologi_Informasi"  
        << endl;  
22     ofile << "Fasilkom-TI_USU_Medan" << endl;  
23 }
```



Membuka File untuk Operasi Output

```
24  int main()  
25  {  
26      OutFile myFile("Contoh4_1.txt");  
27  
28      myFile.openFile();  
29      myFile.writeFile();  
30      myFile.closeFile();  
31  
32      return 0;  
33  }
```



Writing Characters

- Menggunakan fungsi `put()` untuk Menulis File.
 - Class `alphabet` menyimpan nama file output dalam sebuah string yang berupa array karakter.
 - Constructor akan menyalin nama file yang dikirimkan sebagai argumen pada saat objek diturunkan.
 - Sebuah fungsi disediakan untuk menuliskan karakter-karakter alfabet ke dalam file.



Writing Characters

```
1 //Contoh4_2.cpp
2 #include <fstream>
3 #include <iomanip>
4 #include <cstring>
5 using namespace std;
6 class Alfabet
7 {
8     private:
9         char filename[80];
10    public:
11        Alfabet(char fname[]) {strcpy(filename,
12                                   fname);}
13        void writeFile();
14        ~Alfabet(){}
15 };
```



Writing Characters

```
15 void Alfabet::writeFile()  
16 {  
17     ofstream ofile(filename);  
18  
19     for(int hrf='A'; hrf <= 'Z'; hrf++)  
20         ofile.put((char) hrf);  
21  
22     ofile.close();  
23 }
```



Writing Characters

```
24  int main()  
25  {  
26      Alfabet myFile("Contoh4_2.txt");  
27  
28      myFile.writeFile();  
29  
30      return 0;  
31  }
```



Writing Alphabet at Random

- The following program generate a number of alphabet characters and save them into a file.
 - Alphabet characters are put into a character array.
 - A random number is generated in the range of 0-25, and then 65 will be added to make it an alphabet character.



Writing Characters

```
1 //Contoh4_3.cpp
2 #include <fstream>
3 #include <cstdlib>
4 #include <ctime>
5 #include <cstring> // Menggunakan fungsi
   string C
6 #define N 100
7
8 using namespace std;
9 class RandomAlfabet
10 {
11     private:
12         ofstream output_file;
13         char filename[80];
14         char random_char[N];
```



Writing Characters

```
15     public:
16         RandomAlfabet(char []);
17         void openFile() {output_file.open(filename
18             );}
19         void writeFile();
20         void create_random();
21         void print_chars();
22         void closeFile() {output_file.close();}
23         ~RandomAlfabet(){}
24     };
25 RandomAlfabet::RandomAlfabet(char fname[])
26 {
27     strcpy(filename, fname);
28 }
```



Writing Characters

```
29 void RandomAlfabet::create_random()
30 {
31     int num;
32     srand(time(0));
33     for(int i=0; i<N; i++)
34     {
35         num = rand() % 26;
36         random_char[i] = num + 65;
37     }
38 }
39
40 void RandomAlfabet::print_chars()
41 {
42     for(int i=0; i<N; i++)
43         printf("%c", random_char[i]);
44     printf("\n");
45 }
```



Writing Characters

```
46 void RandomAlfabet::writeFile()  
47 {  
48     for(int i=0; i<N; i++)  
49         output_file << random_char[i];  
50 }  
51  
52 int main()  
53 {  
54     RandomAlfabet myFile("Contoh4_3.txt");  
55     myFile.openFile();  
56     myFile.create_random();  
57     myFile.print_chars();  
58     myFile.writeFile();  
59     myFile.closeFile();  
60     return 0;  
61 }
```



Constructor dan Manipulator

- Menggunakan fungsi constructor untuk Menulis File
 - Fungsi constructor dapat digunakan untuk membuka file:
 - `ofstream ofile("Contoh4_3.txt");`
- Menggunakan manipulator dan anggota fungsi output
 - Dapat digunakan untuk melaksanakan operasi file.



Constructor dan Manipulator

```
1 //Contoh4_3.cpp
2 #include <fstream>
3 #include <iomanip>
4 #include <cstring>
5 using namespace std;
6 class OutFile
7 {
8     private:
9         char filename[80];
10    public:
11        OutFile(char fname[]) {strcpy(filename,
12                                fname);}
13        void writeFile();
14        ~OutFile(){}
15};
```



Constructor dan Manipulator

```
15 void OutFile::writeFile()
16 {
17     ofstream ofile(filename);
18
19     ofile << "_Dec_" << "_Oct_" << "_Hex_" <<
        endl;
20     for(int i=0; i <= 255; i++)
21         ofile << setw(5) << dec << i
22             << setw(5) << oct << i
23             << setw(5) << hex << i << endl;
24     ofile.close();
25 }
```



Constructor dan Manipulator

```
26  int main()
27  {
28      OutFile myFile("Contoh4_3.txt");
29
30      myFile.writeFile();
31
32      return 0;
33  }
```



Membuka File untuk Operasi Input

- Menggunakan kelas *ifstream*
 - Turunkan sebuah objek dari kelas *ifstream*.
ifstream input_file("namafile.ext");
 - Baca isi file baik per baris maupun per karakter, menggunakan operator extraction atau fungsi anggota.
 - Ketika membaca file, posisi umumnya bermula pada awal file dan membaca isinya hingga ditemukan akhir file.



Membuka File untuk Operasi Input

```
1 //Contoh4_5.cpp
2 #include <iostream>
3 #include <fstream>
4 #include <cstring>
5 using namespace std;
6 class InputFile
7 {
8     private:
9         ifstream rekord;
10         char namafile[80];
11     public:
12         InputFile(char fname[]) {
13             strcpy(namafile, fname);
14             void bukaFile() {rekord.open(namafile);}
15             void bacaFile();
16             void tutupFile() {rekord.close();}
17 }
```



Membuka File untuk Operasi Input

```
18 void InputFile::bacaFile()
19 {
20     char brs[80];
21     long htgBrs = 0L;
22
23     if(rekord.fail())
24         cerr << "Error_membuka_file:_\"
25             << namafile << endl;
26     else
27     {
28         while(!rekord.eof())
29         {
30             rekord.getline(brs, sizeof(brs));
31             cout << brs << endl;
```



Membuka File untuk Operasi Input

```
32     if ((htgBrs>0) && ((++htgBrs % 24) == 0))
33     {
34         cout << "PRESS_<ENTER>..." << endl;
35         cin.get();
36     }
37 }
38 }
39
40 int main()
41 {
42     InputFile satuFile("Contoh4_3.txt");
43     satuFile.bukaFile();
44     satuFile.bacaFile();
45     satuFile.tutupFile();
46     return 0;
47 }
```



Menguji Keberhasilan File I/O

Fungsi Anggota	Kegunaan
good	Return 1 jika operasi berhasil
eof	Return 1 jika ditemukan akhir file
fail	Return 1 jika terjadi error
bad	Jika dilakukan operasi tidak sah



Menguji Keberhasilan File I/O

```
1 //Contoh4_6.cpp
2 #include <iostream>
3 #include <fstream>
4 #include <cstring>
5 using namespace std;
6 class InputFile
7 {
8     private:
9         ifstream infile;
10        ofstream outfile;
11    public:
12        InputFile(char [], char []);
13        int bacaFile();
14        void tutupFile(){
15            infile.close(); outfile.close();}
16};
```



Menguji Keberhasilan File I/O

```
17 InputFile::InputFile(char asal[], char tuju
    [])
18 {
19     infile.open(asal);
20     if(infile.fail())
21     {
22         cerr << "Error_membuka_file:_\"
23             << asal << endl;
24         exit(0);
25     }
26     outfile.open(tuju);
27     if(outfile.fail())
28     {
29         cerr << "Error_membuka_file:_\"
30             << tuju << endl;
31         exit(0);
32     }
```



Menguji Keberhasilan File I/O

```
34  int InputFile::bacaFile()
35  {
36      char brs[80];
37      while(!infile.eof())
38      {
39          infile.getline(brs, sizeof(brs));
40          outfile << brs << endl;
41          if (outfile.fail())
42          {
43              cerr << "Error_menulis_file." << endl;
44              return 1;
45          }
46      }
47      return 0;
48  }
```



Menguji Keberhasilan File I/O

```
49  int main()
50  {
51      InputFile duaFile("Contoh4_3.txt", "
        Contoh4_5.txt");
52
53      duaFile.bacaFile();
54      duaFile.tutupFile();
55
56      return 0;
57  }
```



Operasi File Biner

- Secara default operasi file dilakukan dalam modus teks.
 - Program dapat mengenali tanda akhir file sebagai kode ASCII 26 (Ctrl-Z).
 - Hal yang sama tidak berlaku pada file biner karena nilai ASCII ini dapat ditemukan di sembarang tempat pada file.
- Membuka file dalam modus biner untuk input:
`ifstream input_file("namafile.ext", ios::binary);`
- Membuka file dalam modus biner untuk output:
`ofstream output_file("namafile.ext", ios::binary);`



Operasi File Biner

- Lakukan operasi input/output menggunakan fungsi anggota read dan write.
 - `input_file.read(buffer, sizeof(buffer));`
 - `output_file.write(buffer, sizeof(buffer));`



Menulis File Biner

```
1 //Contoh4_7.cpp
2 #include <iostream>
3 #include <fstream>
4
5 using namespace std;
6
7 class FileBiner
8 {
9     private:
10         ofstream outfile;
11     public:
12         FileBiner(char fname[]);
13         void tulisFile();
14         void tutupFile(){outfile.close();}
15 };
```



Menulis File Biner

```
16 FileBiner::FileBiner(char fname[])
17 {
18     outfile.open(fname, ios::binary);
19     if(outfile.fail())
20     {
21         cerr << "Error_membuka_file:_"
22         << fname << endl;
23         exit(0);
24     }
25 }
```



Menulis File Biner

```
26 void FileBiner::tulisanFile()  
27 {  
28     int htg;  
29     float harga;  
30  
31     for(htg=1; htg <=30; htg++)  
32     {  
33         harga = htg * 100.0F;  
34         outfile.write((char *) &harga, sizeof(  
35             float));  
36     }
```



Menulis File Biner

```
37  int main()  
38  {  
39      FileBiner satuFile("Contoh4_6.bin");  
40  
41      satuFile.tulisFile();  
42      satuFile.tutupFile();  
43  
44      return 0;  
45  }
```



Membaca File Biner

```
1 //Contoh4_8.cpp
2 #include <iostream>
3 #include <fstream>
4 #include <iomanip>
5 using namespace std;
6 class FileBiner
7 {
8     private:
9         ifstream infile;
10        ofstream outfile;
11    public:
12        FileBiner(char f1name[], char f2name[]);
13        void bacaFile();
14        void tutupFile(){infile.close();
15            outfile.close();}
16};
```



Membaca File Biner

```
17  FileBiner::FileBiner(char f1name[], char
    f2name[])
18  {
19      infile.open(f1name, ios::binary);
20      if(infile.fail())
21      {
22          cerr << "Error_membuka_file:_"
23              << f1name << endl;
24          exit(0);
25      }
26      outfile.open(f2name);
27      if(outfile.fail())
28      {
29          cerr << "Error_membuka_file:_"
30              << f2name << endl;
31          exit(0);
32      }
```



Membaca File Biner

```
34 void FileBiner::bacaFile()
35 {
36     int htg;
37     float harga;
38
39     while(!infile.eof())
40     {
41         infile.read((char *) &harga, sizeof(
42             float));
43         outfile << setprecision(2)
44         << setiosflags(ios::showpoint |
45             ios::fixed) << harga << endl;
46         cout << harga << endl;
47     }
48 }
```



Membaca File Biner

```
48  int main()
49  {
50      FileBiner duaFile("Contoh4_6.bin", "
        Contoh4_6.txt");
51
52      duaFile.bacaFile();
53      duaFile.tutupFile();
54
55      return 0;
56  }
```

