

# TIF1202 – Pemrograman Berorientasi Objek

## HO 02 - Mengenal I/O Stream

Opim Salim Sitompul

Department of Information Technology  
Universitas Sumatera Utara



# Outline

- 1 Tujuan
- 2 Pendahuluan
- 3 Manipulator
- 4 Menggunakan <iomanip>
- 5 Menuliskan output berformat
- 6 Menggunakan input/output karakter
- 7 Menggunakan input baris getline()
- 8 Latihan Pemrograman



# Tujuan

- Dalam kuliah ini mahasiswa akan diperkenalkan pada:
  - Konsep dasar I/O stream
  - Bagaimana menggunakan header file
  - Memahami manipulator
  - Bagaimana memformat input dan output
  - Menggunakan anggota fungsi untuk input/output karakter
  - Menggunakan anggota fungsi untuk input satu baris



# Pendahuluan

- I/O stream adalah sederetan aliran karakter yang dituliskan ke layar tampilan atau dibaca dari keyboard
- Ada 2 operasi input/output standard yang menggunakan I/O stream tersebut:
  - cin untuk input
  - cout untuk output
- I/O stream didefinisikan sebagai kelas dalam header file `iostream.h` atau `istream.h` dan `ostream.h`
- Header file disertakan menggunakan direktif `#include` tanpa menuliskan “.h”



# Pendahuluan

```
1 //Contoh2_1.cpp
2 #include <iostream>
3 #include <string>
4
5 using namespace std;
6
7 class Nama
8 {
9     private:
10         string nama_depan;
11         string nama_belakang;
12
13     public:
14         void ambilNama();
15         void displayNama();
16 };
```



# Pendahuluan

```
1 void Nama::ambilNama()  
2 {  
3     cout << "Tuliskan_nama_depan_dan_nama_  
         belakang: ";  
4     cin >> nama_depan >> nama_belakang;  
5 }  
6  
7 void Nama::displayNama()  
8 {  
9     cout << "Halo, " << nama_depan << "  
10         << nama_belakang << endl;  
11 }
```



# Pendahuluan

```
1 int main()  
2 {  
3     Nama satuNama;  
4  
5     satuNama.ambilNama();  
6     satuNama.displayNama();  
7  
8     return 0;  
9 }
```



# Manipulator

- Digunakan untuk menyaring output I/O stream

```
1 // Contoh2_2.cpp
2 #include <iostream>
3
4 using namespace std;
5
6 class HexOct
7 {
8     private:
9         int number;
10
11     public:
12         void getNumber();
13         void displayNumber();
14 };
```





# Manipulator

```
1 void HexOct::getNumber()
2 {
3     cout << "Please_give_me_a_number:_";
4     cin >> number;
5 }
6
7 void HexOct::displayNumber()
8 {
9     cout << "Decimal:_ " << number << "\t\
10         tHexadecimal:_ "
11         << hex << number << endl;
12     cout << "Decimal:_ " << dec << number
13         << "\tOctal:_ " << oct << number << endl;
14 }
```



# Manipulator

```
1  int main()  
2  {  
3      HexOct aNumber;  
4  
5      aNumber.getNumber();  
6      aNumber.displayNumber();  
7  
8      return 0;  
9  }
```

- Anggota variabel number yang berjenis integer ditampilkan dalam bentuk heksadesimal, dan oktal menggunakan manipulator hex dan oct.



# Menggunakan <iomanip>

- Berkas judul `iomanip.h` memuat fungsi-fungsi pustaka untuk memanipulasi output.
- Manipulator yang digunakan adalah `setbase()` yang terdapat di dalam file header `iomanip.h`.
- Selain itu juga digunakan manipulator `setiosflags` dan `showbase` untuk menampilkan basis bilangan (0 untuk octal) dan (0x atau 0X untuk hexadecimal).
- Manipulator `setiosflags` dan `uppercase` digunakan untuk menuliskan basis bilangan hexadecimal dalam huruf besar (0X).



# Menggunakan <iomanip>

```
1 //Contoh2_3.cpp
2 #include <iostream>
3 #include <iomanip>
4
5 using namespace std;
6
7 class NumSetbase
8 {
9     private:
10         int number;
11     public:
12         void getNumber();
13         void displayNumber();
14 };;
```



## Menggunakan <iomanip>

```
1 void NumSetbase::getNumber()
2 {
3     cout << "Please_give_me_a_number:_";
4     cin >> number;
5 }
6
7 void NumSetbase::displayNumber()
8 {
9     cout<< setiosflags(ios::showbase);
10    cout<< setiosflags(ios::uppercase);
11    cout<<"Decimal_"<<number<<"_in_octal:_";
12    cout<<setbase(8)<<number<<endl;
13    cout<<"Decimal_"<<setbase(10)<<number;
14    cout<<"_in_Hexadecimal:_"<<setbase(16)
15        <<number<<endl;
16 }
```



# Menggunakan <iomanip>

```
1  int main()
2  {
3      NumSetbase aNumber;
4
5      aNumber.getNumber();
6      aNumber.displayNumber();
7
8      return 0;
9  }
```



# Menuliskan output berformat

- Fungsi-fungsi iomanip berikut dapat digunakan untuk membantu penulisan output dalam bentuk yang berformat.

```
1 //Contoh2_4.cpp
2 #include <iostream>
3 #include <iomanip>
4
5 using namespace std;
6
7 class NumSetPrecision
8 {
9     private:
10         float number;
11     public:
12         void Initialize() {number = 0.0;}
```



# Menuliskan output berformat

```

1      void getNumber(float number) {this->number
      =number;}
2      float returnNumber() {return number;}
3      void displayNumber();
4  };
5
6  void NumSetPrecision::displayNumber()
7  {
8      cout << setfill('-');
9      cout << setprecision(5) << showpoint << setw
      (9);
10     cout << number << endl;
11 }
    
```





# Menuliskan output berformat

```
1  int main()  
2  {  
3      NumSetPrecision oneNumber;  
4      NumSetPrecision twoNumber;  
5  
6      oneNumber.Initialize();  
7      oneNumber.displayNumber();  
8      twoNumber.getNumber(3.14159);  
9      twoNumber.displayNumber();  
10  
11     return 0;  
12 }
```



# Menuliskan output berformat

- Fungsi `std::setfill('-')` mengisi field-field yang kosong dengan karakter '-' yang diberikan.
- Fungsi `std::setprecision(5)` menentukan angka presisi di belakang titik desimal (termasuk tanda titik)
- Fungsi `std::showpoint` memperlihatkan angka nol yang perlu dituliskan setelah titik desimal apabila presisi yang diberikan tidak terpenuhi.
- Fungsi `std::setw(9)` menentukan lebar field maksimum.



# Menuliskan output berformat

```
1 // Namafile: Contoh2_5.cpp
2 // Membalik bilangan bulat
3 #include <iostream>
4 #include <iomanip>
5 using namespace std;
6 class BalikBilangan
7 {
8     private:
9         int bilangan;
10        int hasil;
11        int pjgBil;
12    public:
13        void ambilBilangan();
14        void balik();
15        void displayHasil();
16 };
```



# Menuliskan output berformat

```

1 void BalikBilangan::ambilBilangan()
2 {
3     cout << "Membalik_bilangan\n";
4     cout << "Berikan_sebuah_bilangan_bulat:_";
5     cin >> bilangan;
6 }
7
8 void BalikBilangan::balik()
9 {
10     int blk=0, sisa=bilangan;
11     pjgBil = 0;
12     while(sisa)
13     {
14         hasil = (blk * 10) + (sisa%10);
15         sisa /= 10;
16         blk = hasil;
17         pjgBil++;

```



# Menuliskan output berformat

```
1 void BalikBilangan::displayHasil()
2 {
3     cout << "Bilangan_Anda_sekarang_=";
4     cout << setfill('0') << setw(pjgBil);
5     cout << hasil << endl;
6 }
7
8 int main()
9 {
10     BalikBilangan bil;
11
12     bil.ambilBilangan();
13     bil.balik();
14     bil.displayHasil();
15
16     return 0;
17 }
```



# Menggunakan input/output karakter

- Kelas `iostream` memiliki anggota fungsi untuk meminta input karakter dan anggota fungsi menuliskan karakter.
- Anggota fungsi untuk input karakter adalah `cin.get()` dan anggota fungsi untuk output karakter adalah `cout.put()`.



# Menggunakan input/output karakter

```

1 //Contoh2_6.cpp
2 #include <iostream>
3
4 using namespace std;
5
6 class Kalimat
7 {
8     private:
9         char text[128];
10        int ct;
11    public:
12        void getText();
13        void displayText();
14 };
    
```



# Menggunakan input/output karakter

```
1 void Kalimat::getText()
2 {
3     int i=0;
4     char ch;
5
6     cout<<"Tuliskan_sembarang_kalimat:_"<<endl;
7     do
8     {
9         cin.get(ch);
10        text[i++] = ch;
11    }
12    while(ch != '\n');
13    text[i] = '\0';
14    ct = i;
15 }
```





# Menggunakan input/output karakter

```

1 void Kalimat::displayText()
2 {
3     char ch;
4
5     cout << "Anda_menuliskan:_" << endl;
6     for(int i=0; i<ct; i++)
7     {
8         ch = text[i];
9         switch(ch)
10        {
11            case '_' : cout<<"<SPC>="<<(int)ch<<endl;
12                       break;
13            case '\n' : cout<<"<CR>="<<(int)ch<<endl;
14                       break;

```



# Menggunakan input/output karakter

```
1      default : if ((ch >= '0') && (ch <= '9'))
2          cout<<ch<<"="<<(int)ch<<" (Angka)"<<endl;
3          else
4          cout<<ch<<"="<<(int)ch<<" (Huruf)"<<endl;
5      }
6  }
7  cout.put ('\n');
8  }
9
10 int main()
11 {
12     Kalimat satuTeks;
13
14     satuTeks.getText();
15     satuTeks.displayText();
16     return 0;
17 }
```



# Menggunakan input baris getline()

- Berguna untuk menerima input dalam satu baris termasuk tanda spasi.

```

1 // Contoh2_7.cpp
2 #include <iostream>
3 #include <cstring>
4 using namespace std;
5 class Movie
6 {
7     private:
8         char name[64];
9         char first_star[64];
10        char second_star[64];
11    public:
12        void Initialize() {}
13        void getMovie(char *, char *, char *);

```



# Menggunakan input baris getline()

```
1  void show_movie();
2  void get_movie();
3  };
4
5  Void Movie::getMovie(char *movie_name, char *
    first, char *second)
6  {
7      strcpy(name, movie_name);
8      strcpy(first_star, first);
9      strcpy(second_star, second);
10 }
```



## Menggunakan input baris getline()

```
1 void Movie::show_movie()      //member functions
2 {
3     cout << "Nama_film:_ " << "\"" << name << "\" "
4         << endl;
5     cout << "Pemeran_utama:_ " << first_star << "
6         << "dan_"
7         << second_star << endl << endl;
8 }
9 void Movie::get_movie()
10 {
11     cout << "Nama_film:_ ";
12     cin.getline(name, 64);
13     cout << "Pemeran_utama:_ ";
14     cin.getline(first_star, 64);
15     cout << "Pemeran_kedua:_ ";
```



## Menggunakan input baris getline()

```
1    cin.getline(second_star, 64);
2 }
3
4 int main ()
5 {
6     Movie starwars;
7     Movie myMovie;
8
9     starwars.getMovie("Star_Wars:_The_Force_
        Awakens", "Daisy_Ridley", "John_Boyega");
10    starwars.show_movie();
11    myMovie.get_movie();
12    myMovie.show_movie();
13
14    return 0;
15 }
```



# Latihan Pemrograman

- 1 Dengan menggunakan anggota fungsi `cin.get()` buatlah sebuah program C++ untuk meminta user menuliskan sebuah kalimat. Program kemudian menampilkan output berapa banyak huruf hidup pada kalimat tersebut.
- 2 Dengan menggunakan anggota fungsi `cin.get()` buatlah sebuah program C++ untuk mentabulasi banyaknya masing-masing karakter abjad dan tabulasi banyaknya karakter angka 0 – 9 pada sebuah kalimat yang diminta sebagai input. Gunakan anggota fungsi `cout.put()` untuk menuliskan hasil tabulasi dimaksud.



# Latihan Pemrograman

- 8 Buatlah sebuah program C++ sebagai berikut. Buatlah sebuah kelas yang memuat anggota variabel sebuah array karakter sejumlah 26 elemen. Kemudian acaklah bilangan sebanyak 26 buah yang mewakili huruf-huruf abjad (range bilangan 1 - 26). Simpanlah ke dalam array. Mintalah input sebuah huruf kepada user, kemudian carilah apakah huruf tersebut ada di dalam array. Jika ada, sebutkan posisi huruf itu di dalam array. Beritahu huruf dimaksud tidak ada dalam array, jika tidak ditemukan. Kemudian tampilkanlah daftar huruf yang ada dalam array, dengan output yang ditampilkan adalah sebagai berikut:





# Latihan Pemrograman

- 1 Berikan sebuah huruf: A
- 2 Huruf yang dicari tidak ada dalam array.
- 3 Q J D B K D M Z Y T C I C O I B B B Q Z T B Q  
Y D Q
- 4
- 5 Berikan sebuah huruf: Y
- 6 Huruf yang dicari ada di posisi ke 26
- 7 H L E D O J I T J D D G G R I C P S K F X C M  
W M \*Y\*
- 8
- 9 Berikan sebuah huruf: N
- 10 Huruf yang dicari ada di posisi ke 13 dan 26
- 11 Z C E R Z W A H T R C Z \*N\* X S B H L K A S A  
E Q Q \*N\*



# Latihan Pemrograman

- 1 Berikan sebuah huruf: E
- 2 Huruf yang dicari ada di posisi ke 5, 14 dan 15
- 3 U K Q Y \*E\* N W T N H N M N \*E\* \*E\* T W F M M  
T C G P V T
- 4
- 5 Berikan sebuah huruf: Q
- 6 Huruf yang dicari ada di posisi ke 3, 9, 11 dan 14
- 7 H T \*Q\* K R G A I \*Q\* T \*Q\* U R \*Q\* V J D T K  
T T G M P U A

