

## Bahasa Pemrograman Object Oriented Programming

Kuliah 8 :: Pengantar ke C++

Julio Adisantoso

## C++ as a Better C

Program sederhana C++

```
#include <iostream>
using namespace std;
int main() {
    int a;
    cin >> a;
    int b;
    cin >> b;
    int sum=a+b;
    cout << sum << endl;
    return 0;
}
```

Bandingkan dengan program C!

Slide 1

## Function Overloading

Memungkinkan membuat fungsi dengan nama sama.

```
#include <iostream>
using namespace std;

int kuadrat(int x) { return x*x; }
double kuadrat(double x) { return x*x; }

int main() {
    cout << kuadrat(7) << endl
         << kuadrat(7.5) << endl;
    return 0;
}
```

Slide 2

## Function Template

Memungkinkan membuat "cetakan" fungsi.

```
#include <iostream>
using namespace std;
template <class T>
T maksimum(T n1, T n2, T n3) {
    T besar = n1;
    if (n2>besar) besar=n2;
    if (n3>besar) besar=n3;
    return besar;
}

int main() {
    int i1, i2, i3;
    cin >> i1 >> i2 >> i3;
    cout << maksimum(i1,i2,i3) << endl << endl;
    double d1, d2, d3;
    cin >> d1 >> d2 >> d3;
    cout << maksimum(d1,d2,d3) << endl;
    return 0;
}
```

Slide 3

## Default Argument

Memungkinkan membuat fungsi dengan nilai default.

```
#include <iostream>
using namespace std;
double bagi(double a, double b=1.0) {
    return a/b;
}

int main() {
    cout << bagi(5.0,2.0) << endl;
    cout << bagi(5.0) << endl;
    return 0;
}
```

Slide 4

## Contoh C++ : Struct

```
#include <iostream>
using namespace std;

struct Time {
    int hour; // 0-23
    int minute; // 0-59
    int second; // 0-59
};
typedef struct Time TIME;

TIME t;
```

Slide 5

## Contoh C++ : Prosedur

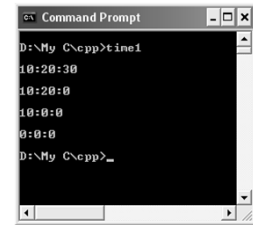
```
void setTime(int h=0, int m=0, int s=0)
{
    t.hour=h;
    t.minute=m;
    t.second=s;
}

void print(void)
{
    cout << endl
         << t.hour << ":"
         << t.minute << ":"
         << t.second << endl;
}
```

Slide 6

## Contoh C++ : Main Routine

```
main() {
    setTime(10,20,30);
    print();
    setTime(10,20);
    print();
    setTime(10);
    print();
    setTime();
    print();
    return 0;
}
```



Slide 7

OOP

Slide 8

## Prinsip Dasar

- Membungkus prosedur dan data dalam satu obyek → **Encapsulation**.
- OOP memodelkan obyek yang ada di dunia nyata ke dalam software obyek dalam pemrograman.
- Implementasi dalam bentuk **Class**.
- Berfungsi sebagai ADT (**Abstract Data Type**).

Slide 9

## Definisi Class Time

```
class Time {
public:
    Time(); // default constructor
    void setTime(int, int, int); // member function
    void print();
private:
    int hour; // data members
    int minute;
    int second;
};
```

Slide 10

## Implementasi Class Time

```
Time :: Time()
{
    hour = minute = second = 0;
}

void Time :: setTime(int h=0,int m=0,int s=0)
{
    hour = h;
    minute = m;
    second = s;
}
```

Slide 11

## Implementasi Class Time

---

```
void Time :: print()
{
    cout << (hour < 10 ? "0" : "")
        << hour << ":"
        << (minute < 10 ? "0" : "")
        << minute << ":"
        << (second < 10 ? "0" : "")
        << second << ":"
        << endl;
}
```

Slide 10

## Contoh Running

---

```
#include <iostream>
using namespace std;
#include "time.h"

main() {
    Time t1, t2;
    t1.setTime(10, 20, 30);
    t2.setTime(10);
    t1.print;
    t2.print;
}
```

Slide 11