



PEMROGRAMAN LANJUT

03 | Const

Keyword const

- Digunakan untuk mendefinisikan konstanta yaitu nilai yang tidak dapat diubah. Bisa digunakan pada:
 - Variabel
 - Pointer
 - Argumen fungsi dan return type
 - Variabel yang menjadi anggota class
 - Objek dari sebuah class
 - Method yang menjadi anggota class

Constant Variable

- Nilai variabel tidak dapat diubah
- Constant variable harus diberikan nilai awal ketika dideklarasikan
- Manakah line yang error di contoh program dibawah ini?

```
int main {  
    const int i = 10;  
    const int j = i + 10;  
    i++;  
}
```

Pointer dengan keyword const

- Ada dua cara penggunaan keyword const untuk pointer:

1. Pointer yang menunjuk constant variable

```
const int* u;
```

```
char const* v;
```

2. Constant pointer

```
int x = 1;  
int* const w = &x;
```

diperlukan jika ingin menyimpan variable yang dapat diubah nilainya tetapi dengan lokasi yang tetap

Apa artinya code dibawah ini?

```
const int* const x;
```

Const untuk argumen fungsi dan return type

- Ketika ditambah keyword const maka nilainya tidak dapat diubah

```
void f(const int i) {  
    i++;  
}
```

```
const int g() {  
    return 1;  
}
```

Const untuk variabel yg menjadi anggota class

- Adakah yg error dengan line program dibawah?
- Apa yg harus dilakukan agar tidak error?

```
class Test {  
    const int i;  
public:  
    Test (int x) { i = x; }  
};  
int main() {  
    Test t(10);  
    Test s(20);  
}
```

Const untuk objek dari sebuah class

- Kalau objek dibuat const maka nilai dari anggota kelas tidak akan pernah dapat diubah selama objek tersebut dipakai

```
const class_name object;
```

```
const Test r(30);
```

- Tidak dapat mengakses non-const function

Const untuk method yang menjadi anggota class

- Method yang dibuat const tidak bisa memodifikasi variable anggota class

```
class StarWars {
public:
    int i;
    StarWars(int x) // constructor
    { i = x; }
    int falcon() const // constant function
    { /* can do anything but will not modify any data members */
    cout << "Falcon has left the Base"; }
    int gamma() { i++; }
};

int main() {
    StarWars objOne(10); // non const object
    const StarWars objTwo(20); // const object
    objOne.falcon(); // No error
    objTwo.falcon(); // No error
    cout << objOne.i << objTwo.i;
    objOne.gamma(); // No error
    objTwo.gamma(); // Compile time error }
```


Keyword mutable

- Digunakan pada variabel class sehingga ketika terdapat objek const maka nilai dari data member dapat diubah

```
class Zee {
    int i;
    mutable int j;
public:
    Zee() {
        i = 0;
        j = 0;
    }

    void fool() const {
        i++;    // will give error
        j++;    // works, because j is mutable
    }
};

int main() {
    const Zee obj;
    obj.fool();
}
```