



KONSTRUKTOR & DESTRUKTOR

Oleh :

Rasim

Ilmu Komputer - UPI

KONSTRUKTOR

- Adalah fungsi anggota yang mempunyai nama yang sama dengan nama kelasnya
- Tidak mempunyai nilai balik
- Harus diletakkan pada bagian publik

JENIS

- Default constructor: **konstruktor** yang menginisialisasi objek dengan nilai(-nilai) default yang ditentukan oleh perancang kelas. Dalam deklarasi kelas, ctor ini tidak memiliki parameter formal.
- User-dened constructor: **konstruktor** yang menginisialisasi objek dengan nilai(-nilai) yang diberikan oleh pemakai kelas pada saat objek diciptakannya.



KEGUNAAN

- Mengalokasikan ruang bagi sebuah objek
- Memberikan nilai awal terhadap anggota data sebuah objek

Contoh

```
Class Kompleks{
```

```
    double re;
```

```
    double im;
```

```
public :
```

```
    Kompleks();
```



Konstruktor

```
    void info();
```

```
        Kompleks::Kompleks{
```

```
            ....  
        }
```

```
}
```

KONSTRUKTOR dg ARGUMENT

```
class buku {
    char judul[35];
    char pengarang[25];
    int jumlah;

public:
    buku(char *Judul, char *Pengarang, int Jumlah);
    void info();
};

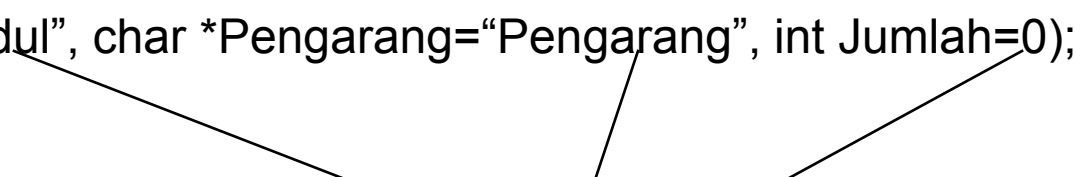
int main()
{
    //clrscr();
    buku novel("Siaga Merah","Alistair Maclean",12);
    novel.info();

    system("PAUSE");
    return (0);
}

buku::buku(char *Judul, char *Pengarang, int
Jumlah){
    strcpy(judul,Judul);
    strcpy(pengarang, Pengarang);
    jumlah=Jumlah;
}
void buku::info(){
    printf("Judul          : %s\n",judul);
    printf("Pengarang      :
%s\n",pengarang );
    printf("Jumlah          : %d\n",jumlah );
}
```

ARGUMENT BAWAAN

```
class buku {  
    char judul[35];  
    char pengarang[25];  
    int jumlah;  
  
    public:  
        buku(char *Judul="Judul", char *Pengarang="Pengarang", int Jumlah=0);  
        void info();  
};
```



Nilai Default

ARGUMENT BAWAAN...2

```
int main()
{
    buku novel1;
    buku novel2("Siaga Merah");
    buku novel3("Siaga Merah","Alistair Maclean");
    buku novel4("Siaga Merah","Alistair",12);
    novel1.info();
    novel2.info();
    novel3.info();
    novel4.info();

    system("PAUSE");
    return (0);
}
```


OVERLOAD KONSTRUKTOR

```
Class berbagimemori{
    union{
        int x;
        char y[2];
    }mem;

    public:
        berbagimemori() { mem.x=0;}
        berbagimemori(int A) {mem.x=A;}
        berbagimemori(char A0, char A1) {
            mem.y[0] = A0;
            mem.y[1] = A1;
        }

        void info();
}
```

```
Int main(){
    berbagimemori m1;
    berbagimemori m2(0x1234);
    berbagimemori m3(0x27, 0x58);

    m1.info();
    m2.info();
    m3.info();

    system("PAUSE");
    return (0);
}
```

DESTRUKTOR

- Adalah fungsi anggota kelas yang akan dijalankan secara otomatis pada saat objek akan sirna
- Nama destruktur sama dengan nama konstruktor, tetapi diawali tanda (~)

CONTOH

```
Class Bilangan {
```

```
    int l;
```

```
public:
```

```
    Bilangan();
```

```
    ~Bilangan();
```

```
    void info();
```

```
}
```



Destruktor

ALOKASI DINAMIS

Ada beberapa jenis objek yang dapat digunakan di dalam program C++:

- Automatic object: diciptakan jika ada deklarasi objek di dalam blok eksekusi dan dimusnahkan pada saat blok yang mengandung deklarasi tersebut selesai eksekusi
- Static object: diciptakan satu kali pada saat program dimulai dan dimusnahkan pada saat program selesai
- Free store object: diciptakan dengan operator new dan dimusnahkan dengan operator delete. Kedua hal ini harus secara eksplisit dilakukan oleh pengguna kelas/objek.

CONTOH

```
Class Mobil{  
    char *nama;  
    int cc;  
    int jml;  
  
    public:  
        Mobil(char * Nama, int CC, int JML);  
        ~Mobil();  
        void info();  
};
```

```
Int main(){
    Mobil Sedan("Chorolla",1800,6);
    Mobil PickUp("Datsun",1000,2);

    Sedan.info();
    PickUp.info();
}
Mobil::Mobil(char *Nama, int CC, int JML){
    nama=new char[25];
    strcpy(nama,Nama);
}
```

```
Mobil::~~Mobil(){
    delete [ ] nama;
}
```

```
Void Mobil::info(){
    printf("Informasi Mobil :\n");
    printf("Nama   : %s \n",nama);
    printf("CC      : %d \n",cc);
    printf("Jumlah  : %d \n",jml);
}
```