

PRAKTIKKUM V

PEWARISAN 2

1.. Pewarisan Berganda

Konsep pewarisan berganda adalah itu kelas turunan dapat memiliki lebih dari satu kelas induk.

Bentuk Umum Pewarisan Induk Berganda :

```
class nama_kelas_turunan : hak akses kelas_induk1, hak akses kelas_induk2, hak akses kelas_indukN
```

1. Listing Program

```
#include <iostream.h>
#include <conio.h>

class induk1
{
    protected:
    int a;

    public:
    void masuk_a(){
    cout << "Masukkan Integer Pertama :";
    cin >> a;
    }
};

class induk2
{
    protected:
    int b;

    public:
    void masuk_b(){
    cout << "Masukkan Integer Kedua :";
    cin >> b;
    }
};

// Membuat kelas turunan dengan 2 kelas induk
class anak:public induk1,public induk2
{
    public:
    int tambah(){
        return (a+b);
    }

    int kali(){
        return (a*b);
    }
}
```

```

        void tampil()
        {
            cout << "Hasil Penjumlahan= " << tambah() << endl;
            cout << "Hasil Perkalian =" << kali() << endl;
        }
};

void main()
{
    anak x;

    x.masuk_a();
    x.masuk_b();

    x.tampil(0;
}

```

2. Konstruktor dalam penurunan jika memiliki argumen.

Bentuk Umum

```

konstruktor-turunan(daftar-argumen)
induk1(daftar-argumen), induk2(daftar-argumen),..., indukN(daftar-
argumen)
{
    // isi dan fungsi konstruktor untuk kelas turunan
    ....
    ....
}

```

Listing Program

```

#include<iostream.h>

class tambah
{
public:
    tambah(int i, int j){
        cout << "Hasil Penjumlahan" << i << "dan" << j << "adalah"
            << (i+j) << endl;
    }
};

class kali
{
public:
    kali (int i, int j){
        cout << Hasil Perkalian " << i << "dan" << j << "adalah" << (i*j) <<
        endl;
    }
};

```

```

class tampil : public tambah, public kali
{
    int pzr1,par2,par3,par4;

    public:
        tampil(int a, int b, int c, int d);
        void par();
};

// Definisi konstruktor untuk kelas anak
// Pisahkan parameter a dan b untuk konstruktor kelas tambah
// Pisahkan parameter c dan d untuk konstruktor kelas kali

tampil :: tampil(int a, int b, int c, int d): tambah(a,b), kali(c,d)
{
    par1=a;
    par2=b;
    par3=c;
    par4=d;
}

void tampil :: par()
{
    cout << "Parameter 1=" << par1 << endl;
    cout << "Parameter 2=" << par2 << endl;
    cout << "Parameter 3=" << par3 << endl;
    cout << "Parameter 4=" << par4 << endl;
}

void main()
{
    tampil obj(30,15,20,2);

    obj.par();
}

```