

KONSEP DASAR

Tujuan :

- Menjelaskan prinsip abstraksi data
- Menjelaskan prinsip inheritance
- Menjelaskan prinsip polimorfisme

```

Class rekening
{
    int jumlah;
}
public:
    rekening operator+(rekening rekening1)
}

rekening operator+()
{
    rekening r_temp;
    r_temp = jumlah + rekening1.jumlah;
}

mainline()
{
    rekening rek1, rek2, rekTotal
    ...
    ...
    rekTotal = rek1 + rek2
    ...
    ...
}

```

Pengantar

- Pada desain berorientasi objek, fungsi dan data merupakan satu kesatuan.
- Gabungan antar keduanya menghasilkan objek-objek yang aktif yang mampu melakukan satu aktifitas tertentu seperti contoh-contoh sebelumnya.
- Konsep dasar yang dimiliki desain berorientasi objek adalah abstraksi data, penurunan (inheritance), dan polimorfisme.

Abstraksi Data

- Setelah objek dibuat, operator-operator yang diperlukan (seperti +, -, *, /) dapat didefinisikan.
- Kemampuan seperti ini disebut abstraksi data (data abstraction).
- Konsep ini dilakukan dengan menggunakan anggota fungsi yang disebut operator overloading.
- Misalkan penggunaan operator + untuk menjumlahkan objek rekening.

Inheritance (penurunan)

- Merupakan kemampuan sebuah objek untuk mewarisi ciri-ciri objek lain.
- Sebuah objek induk dapat memiliki beberapa objek turunan yang kesemuanya memiliki ciri-ciri objek induk tersebut.
- Objek turunan dapat pula menjadi objek induk bagi objek lainnya.
- Konsep ini diwujudkan pada struktur objek.
- Saat mendesain program faktor penurunan perlu dipertimbangkan untuk menghasilkan program yang efisien.
- Persamaan-persamaan yang dimiliki oleh beberapa class dapat digabung dalam sebuah class induk sehingga setiap class yang diturunkan cukup memuat hal-hal yang spesifik untuk class yang bersangkutan.

Contoh inheritance:

Class SiswaAngkatan2000 adalah Siswa

- Dalam contoh di atas, atribut class SiswaAngkatan2000 adalah sama dengan atribut class Siswa.
- Class Siswa disini merupakan gabungan dari objek-objek siswa dari seluruh angkatan.
- Selain inheritance, class juga mempunyai karakter 'mempunyai' (posession).

Contoh posession:

Class Kendaraan mempunyai Mesin

- Dalam contoh posession, class Kendaraan mempunyai class Mesin dimana tiap class memiliki atributnya masing-masing.

BINA NUSANTARA

Edisi : 1

Revisi : 0

September 2000

- Fungsi BacaPIN() dapat didefinisikan kembali pada objek turunannya, yaitu class Siswa2000 sebagai berikut:

Class Siswa2000 : public Siswa

```
{
public:
    BacaPIN()
    {
        Cetak "PIN Siswa2000 :."
        Baca PINsiswa2000
    }
}
```

BINA NUSANTARA

Edisi : 1

Revisi : 0

September 2000

Polimorfisme

- Kemampuan objek untuk menentukan anggota fungsi yang akan diaplikasikan terhadap objek tersebut, dan hal ini tergantung pada jenjang inheritance.
- Pada jenjang yang berbeda, beberapa fungsi dapat mempunyai nama yang sama namun dengan tugas yang berbeda.
- Fungsi tersebut dinamakan function overloading.

Misalnya : pada class Siswa terdapat fungsi BacaPIN()

Class Siswa

```
{
public:
    BacaPIN()
    {
        Cetak "PIN Siswa :."
        Baca PINsiswa
    }
}
```

BINA NUSANTARA

Edisi : 1

Revisi : 0

September 2000