



Pemrograman Web

Sirojul Munir | rojulman@nurulfikri.ac.id

PHP - Object Oriented Programming

Sirojul Munir | rojulman@nurulfikri.ac.id

Class & Object

- ❑ Class : **Blue Print** dari Object
 - Gambar/Desain Rumah, Resep Masakan
- ❑ Object : **Instance** dari Class
 - Rumah , Hidangan di meja makan

Class & Object

Seorang arsitek ketika membangun **rumah**, ia akan menggambar atau menuangkan ide bentuk dan arsitektur rumah dalam sebuah **blue print**. Pada terminologi OOP, **blue print** dari rumah disebut dengan **class** dan rumah yang dibangun berdasarkan blue print disebut **object**.



Class Declarations

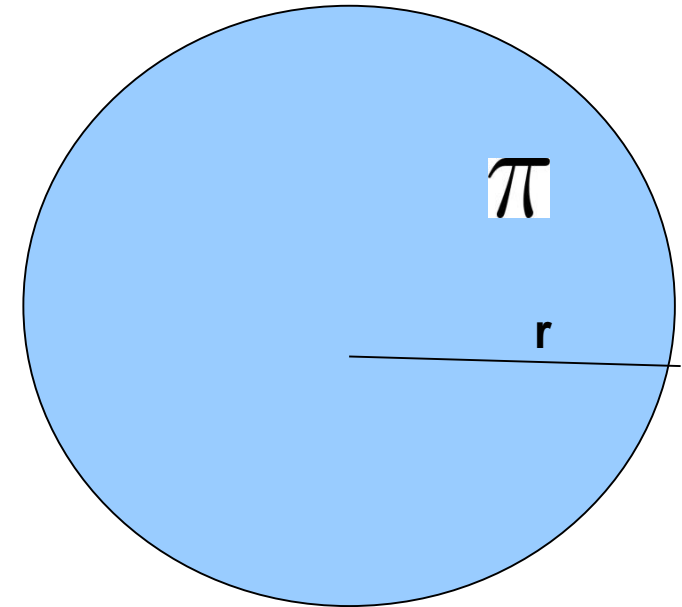
```
class Foo  
{  
  
}
```

```
$foo = new Foo();
```

```
$fii = new Foo();
```

Member Class

- ❑ Variable : **Properti & Konstanta** class
 - Jari , Konstanta PHI
- ❑ Method : **Behavior** Class
 - getLuas , getKeliling



Static Member Class → **ClassName::**

```
class Matematika
{
    public static $counter = 100;

    public static function tambahkan($a , $b)
    {
        return $a + $b;
    }
}
```

```
Matematika::$counter++;
echo 'Counter Sekarang : '. Matematika::$counter;
$x = Matematika::tambahkan(5,4);
echo "5 + 4 = " . $x;
```

Static Member Class → **self::**

```
class Matematika
{
    public static $counter = 100;

    public static function naikanCounter()
    {
        self::$counter++;
    }
}
```

```
Matematika::naikanCounter();
echo 'Counter Sekarang : '. Matematika::$counter;
```


Konstanta Class → self::

```
class Matematika
{
    const PHI = 3.14;

    public static function luasLingkaran($jari)
    {
        $luas = self::PHI * $jari * $jari;
        return $luas;
    }
}

echo 'NILAI PHI '. Matematika::PHI ;
$luasling = Matematika::luasLingkaran(8);
echo 'Luas Lingkaran Jari2nya 8 = ' . $luasling;
```

Member Class Variable → **:: non static**

```
class NilaiSiswa
{
    public $nama;
    public $nilai;
    public $sekolah='SDIT NF';
}
```

```
$ns1 = new NilaiSiswa();
$ns1->nama = 'Faiz Fikri';
$ns1->nilai = 80;
echo $ns1->nama .' Sekolah di ' . $ns1->sekolah;
```

Member Class Method → **:: non static**

```
class NilaiSiswa {  
    public $nama;  
    public $nilai;  
    public $sekolah='SDIT NF';  
  
    public function getHasil()  
    {  
        If ($this->nilai > 55 ) return 'Lulus';  
        else return 'Ngak Lulus';  
    }  
}
```

```
$ns1 = new NilaiSiswa();  
$ns1->nama = 'Faiz Fikri';  
$ns1->nilai = 83;  
echo $ns1->nama . ' Dinyatakan ' . $ns1->getHasil();
```

Constructor Class (1)

- ❑ method bernama `__construct()`
- ❑ baris kode program pada konstruktor akan dieksekusi ketika objek diciptakan
- ❑ Digunakan untuk inisialisasi variable class
- ❑ Contoh: *Untuk membuat objek NilaiSiswa harus ada nama dan nilainya*

Constructor Class (2)

```
class NilaiSiswa {  
    public $nama;  
    public $nilai;  
    public $sekolah='SDIT NF';  
  
    public function __construct($nama, $nilai)  
    {  
        $this->nama = $nama;  
        $this->nilai = $nilai;  
    }  
    public function getHasil()  
    {  
        if ($this->nilai > 55 ) return 'Lulus';  
        else return 'Ngak Lulus';  
    }  
}
```

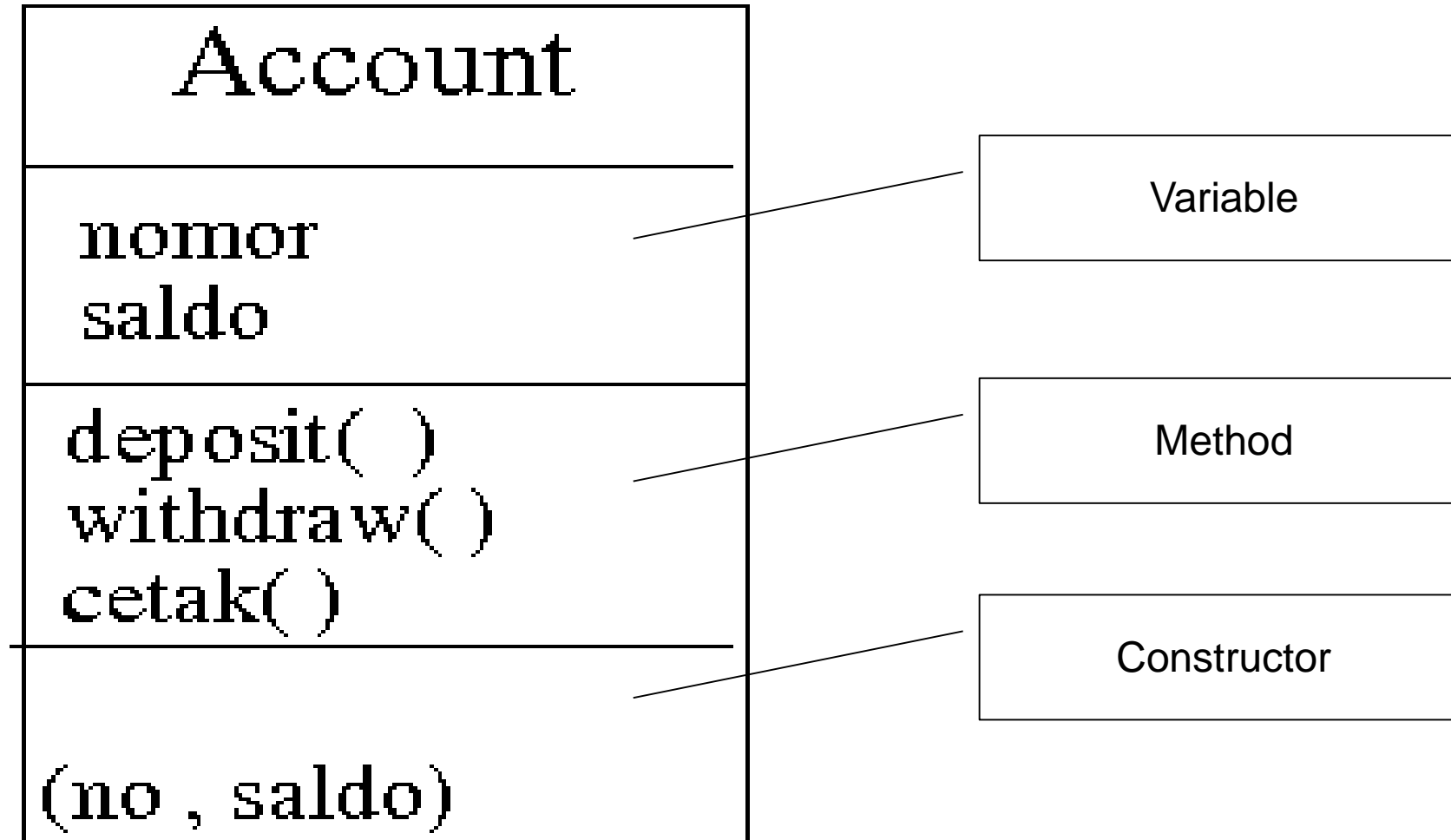
Constructor Class (3)

```
$ns1 = new NilaiSiswa('Rosa',90);  
$ns2 = new NilaiSiswa('Badu',30);  
  
echo $ns1->nama .' : '. $ns1->getHasil();  
  
echo $ns2->nama .' : '. $ns2->getHasil();
```

Access Modifier

- ❑ **public** : dapat diakses dari dalam maupun luar class
- ❑ **private** : hanya dapat diakses dari dalam class
- ❑ **protected** : dapat diakses dari dalam class dan class-class turunannya

Class Diagram



Class Account (1)

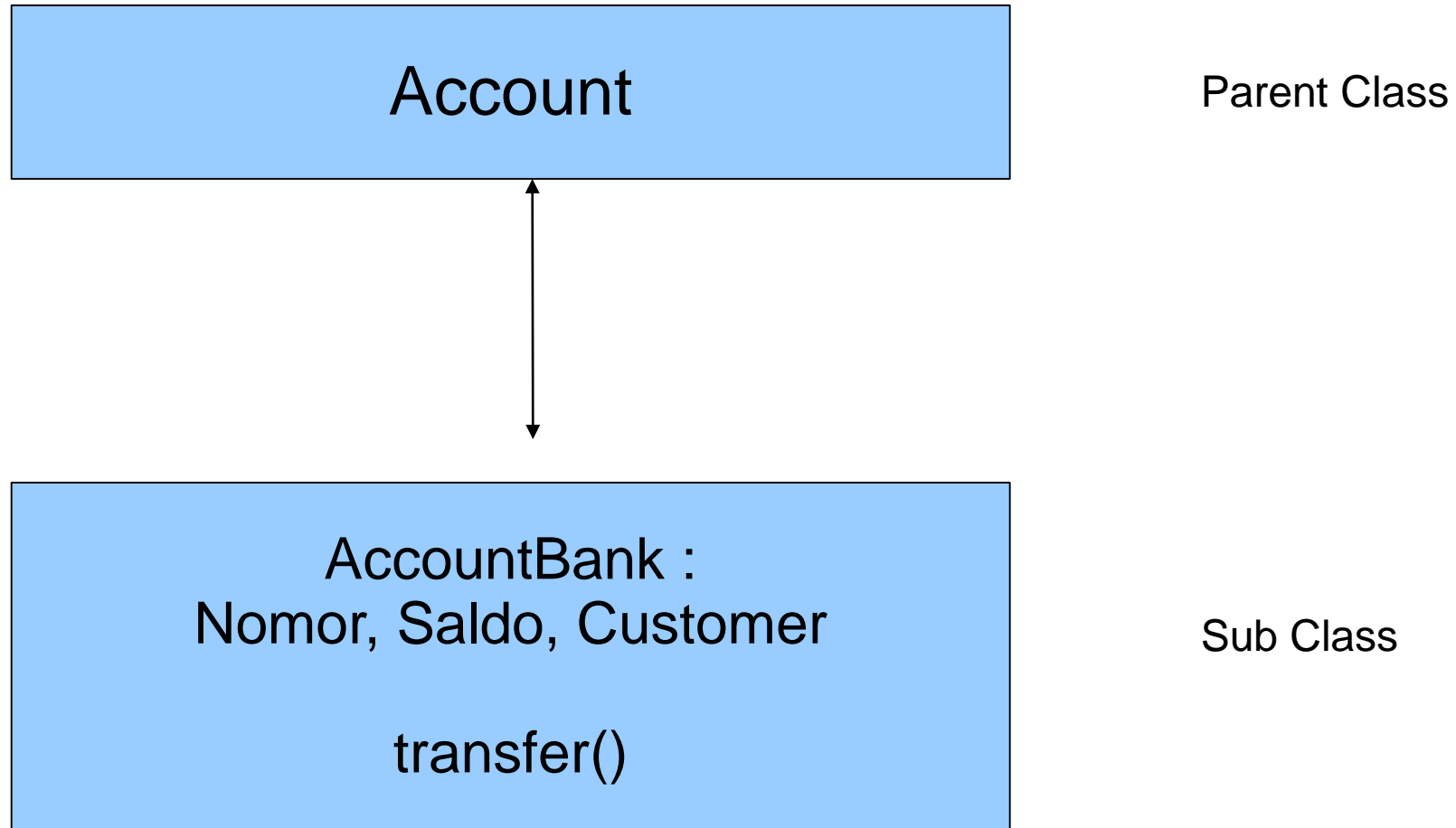
```
class Account {  
    public $nomor;  
    public $saldo;  
  
    public function __construct($no, $saldo)  
    {  
        $this->nomor = $no;  
        $this->saldo = $saldo;  
    }  
    public function deposit($uang)  
    {  
        $this->saldo = $this->saldo + $uang;  
    }  
  
    public function withdraw($uang)  
    {  
        $this->saldo = $this->saldo - $uang;  
    }  
  
    public function cetak() {  
        echo 'Nomor ' . $this->nomor . ' , Saldonya : ' . $this->saldo;  
    }  
}
```

Class Account (2)

```
$ac1 = new Account("020",4000);  
$ac1->deposit(200);  
$ac1->cetak();
```

```
$ac2 = new Account("045",2000);  
$ac2->withdraw(40);  
$ac2->cetak();
```

Inheretance → :: extends



Inheretance → :: extends

```
require_once 'class_account.php';
class BankAccount extends Account {

    public $customer;
    function __construct($no, $saldo_awal, $cust) {
        parent::__construct($no, $saldo_awal);
        $this->customer = $cust;
    }

    function cetak() {
        parent::cetak();
        echo ', Customer : ' . $this->customer;
    }

    function transfer($obj_account, $uang) {
        $obj_account->deposit($uang);
        $this->withdraw($uang);
    }
}
```

Package – Namespace (1)

- Fitur PHP > 5.4
- Pustaka program (Class – Class) tersimpan dalam direktori yang terstruktur sesuai dengan fungsi / kegunaannya
- Digunakan pada beberapa framework PHP
- Pustaka program dapat didistribusikan dalam GITHUB library oleh vendor / programmer yang membuatnya
- Keyword:

```
namespace keu\models; // direktori file keu\models
class Account{
    . . .
```

Package -- Namespace

- Cara penggunaan:

```
<?php
// to be handle by framework on bootstrap file
include_once 'keu\models\Account.php';
use keu\models\Account;

$sac1 = new Account('A001', 5000);
$sac1->cetak();
?>
```

Design Pattern DAO :: Data Access Object

Parent Class

DAO

Constructor:

- \$con

Variable Class:

- \$con = koneksi databse
- \$tbl_name

Method Class:

- getAll()



Sub Class

DAOSiswa, DAOKelas, DAO Jadwal

Constructor:

- \$con

-Method Class:

- findByPK()
- save()
- update()
- delete()

Let's Code !!

- Buat class dengan kriteria
 - Terdapat member variable : private, public
 - Terdapat konstruktor
 - Terdapat fungsi/variable static
 - Terdapat konstanta class
 - Terdapat member fungsi : private, public
- Buat file yang menggunakan class diatas
 - Didalamnya memanggil fungsi2 / variable yang dideklarasikan di class