

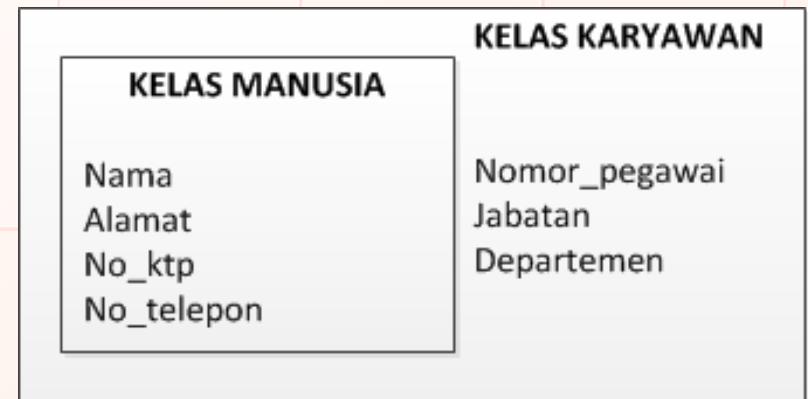
PERTEMUAN 3
PEMOGRAMAN
BERORIENTASI OBJEK

L/O/G/O

INHERITANCE

- Pengertian Pewarisan (Inheritance) yaitu Hubungan dua buah kelas atau lebih dimana ada kelas yang memiliki atribut dan metode yang sama dengan kelas lain beserta atribut dan metode tambahan yang merupakan sifat khusus kelas yang menjadi turunannya
- Kelas yang merupakan kelas turunan biasa disebut kelas anak (subclass) dan kelas yang menjadi dasar penurunan adalah kelas orang tua (superclass)
- Biasanya kelas anak adalah kelas dengan jenis yang khusus dari kelas orang tua.

INHERITANCE



Ilustrasi kelas inheritance

Ilustrasi lingkup kelas inheritance

INHERITANCE

- Misalkan yaitu kelas manusia dan kelas karyawan maka kelas manusia akan menjadi orang tua karena bagaimanapun seorang karyawan adalah manusia.
- Kelas manusia memiliki atribut nama, alamat, no_ktp, no_telepon dll sedangkan karyawan memiliki atribut yang dimiliki oleh manusia ditambah dengan atribut nomor_pegawai, jabatan, departemen dll
- Karena semua atribut manusia dibutuhkan oleh karyawan dan karyawan membutuhkan atribut tambahan nomor_pegawai, jabatan dan departemen maka disimpulkan bahwa kelas karyawan merupakan kelas turunan dari manusia karena semua atribut manusia dibutuhkan oleh karyawan

INHERITANCE



Ilustrasi Kelas Pewarisan Secara Hirarki

INHERITANCE

```
class Manusia
  private nama : string
  private alamat : string
  private no_ktp : string
  private no_telepon : string

  manusia()
  {end constructor}

  procedure setNama(nama : string)
    this.nama <- nama
  {end procedure}

  procedure SetAlamat (alamat : string)
    this.alamat <- alamat
  {end procedure}

  procedure setNoKTP (no_ktp : string)
    this.no_ktp <- no_ktp
  {end procedure}

  procedure setNoTelepon (no_telepon : string)
    this.no_telepon <- no_telepon
  {end procedure}
```

```
function getName () -> string
  -> this.nama
{end function}

function getAddress() -> string
  -> this.alamat
{end function}

function getNoKTP() -> string
  -> this.no_ktp
{end function}

function getNoTelepon() -> string
  -> this.no_telepon
{end function}
{end class}
```

Mendeklarasikan kelas manusia sebagai kelas manusia disertai metode get set

INHERITANCE

```
class Karyawan : Manusia
  private nomor_pegawai : string
  private jabatan : string
  private departemen : string

  Karyawan ()
  {end constructor}

  procedure setNomorPegawai (nomor_pegawai : string
    this.nomor_pegawai <- nomor_pegawai
  {end procedure}

  procedure setJabatan(jabatan : string)
    this.jabatan <- jabatan
  {end procedure}

  procedure setDepartemen (departemen : string)
    this.departemen <- departemen
  {end procedure}
```

```
function getNomorpegawai () -> string
  -> this.nomor_pegawai
{end function}

function getJabatan () -> string
  -> this.jabatan
{end function}

function getDepartemen () -> string
  -> this.departemen
{end function}
{end class}
```

Mendeklarasikan kelas karyawan sebagai kelas anak atau kelas turunan dari kelas manusia beserta metode get dan set milik kelas karyawan

INHERITANCE

```
class Main
  procedure main()
    kmanusia : Manusia
    kkaryawan : Karyawan

    kmanusia <- new Manusia()
    kmanusia.setNama("Gina")
    kmanusia.setAlamat("Bandung")
    kmanusia.setNoKTP ("320.120.777")
    kmanusia.setNoTelepon("08188899977")

    output (kmanusia.getNama () )
    output (kmanusia.getAlamat () )
    output (kmanusia.getNoTelepon() )
```

```
    kkaryawan <- new Karyawan()
    kkaryawan.setNama("Gina")
    kkaryawan.setNoKTP("32-.120.777")
    kkaryawan.setAlamat("Bandung")
    kkaryawan.setNoTelepon("081888899977")
    kkaryawan.setNomorPegawai("13202020")
    kkaryawan.setJabatan("-")
    kkaryawan.setJabatan("-")

    output (kkaryawan.getNama () )
    output (kkaryawan.getAlamat () )
    output (kkaryawan.getNoKTP () )
    output (kkaryawan.getNoTelepon () )
    output (kkaryawan.getNomorPegawai () )
    output (kkaryawan.getJabatan () )
    output (kkaryawan.getDepartemen () )
  {end procedure}
{end class}
```

Mendeklarasikan kelas main atau kelas utama yang mengeksekusi kelas-kelas lain seperti kelas manusia dan kelas karyawan. Kelas main mengisi semua atribut pada kelas manusia dan kelas karyawan menggunakan metode set dan menampilkan isi semua atribut

INHERITANCE

- Kelas anak atau kelas turunan dari kelas lain juga memiliki semua atribut yang dimiliki oleh kelas orangtuanya. Kelas anak jika dieksekusi adalah sebuah objek yang berdiri sendiri dan memiliki semua atribut orang tua yang bisa diisi, sedangkan kelas orang tua juga merupakan objek sendiri
- Misalkan didalam kelas orangtua memiliki atribut nama, maka kelas anak juga memiliki atribut nama yang harus diisi masing-masing. Kelas anak dan kelas orangtua jika dieksekusi akan menjadi objek yang berbeda dan atribut masing-masing

Atribut kelas orangtua
+ atribut kelas anak

Kelas Anak

Atribut kelas orangtua

Kelas Orangtua

INHERITANCE

```
class Titik
  private x : integer
  private y : integer
  Titik ()
    x <- 0
    y <- 0
  {end Titik}

  public getX () -> integer
    -> x
  {end getX}

  public getY () -> integer
    -> y
  {end getY}
(end class)
```

```
class Titik3D : Titik
  private z : integer

  Titik3D ()
    z <- 0
  {end Titik 3D}

  public getZ () -> integer
    -> z
  {end getZ}
{end class}
```

INHERITANCE

- Keuntungan dari pewarisan adalah tidak perlu mengutak atik kode kelas yang membutuhkan tambahan atribut atau metode saja karena tinggal membuat kelas turunannya tanpa harus mengubah kode kelas dasarnya
- Kelas dasar akan mewariskan semua atribut dan metodenya kecuali konstruktor dan destruktur yang memiliki izin akses public dan protector ke kelas turunannya dengan izin akses yang sama dengan kelas dasar
- Ketika sebuah kelas urunan objeknya saat eksekusi, maka secara implisit konstruktor kelas dasar dipanggil dahulu baru kemudian konstruktor kelas turunannya dijalankan. Begitu juga saat objek dimusnahkan maka sestruktur kelas dasar atau orangtua akan dijalankan diawal destruktur kelas turunan atau kelas anak

INHERITANCE

Kelas turunan juga dapat mengimplementasikan metode yang ada pada kelas dasar, maka pada saat eksekusi yang dipanggil adalah metode yang ada pada kelas turunan

```
class Titik3D : Titik
  private z : integer

  Titik3D()
    z <- 0
  {end constructor}

  public function getX() -> integer
    -> 0
  {end function}

  public function getZ() -> integer
    -> z
  {end function}
{end class}
```

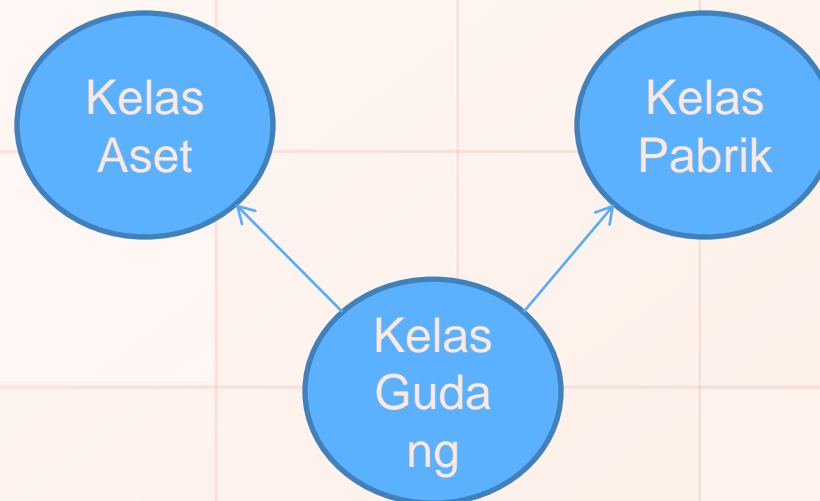
Maka pada saat metode `getX()` dijalankan dengan kode:

```
Titik3D p <- new Titik3D()
Integer x <- p.getX()
```

Variable `x` akan berisi nilai 0 karena yang dijalankan adalah metode `getX` yang ada pada kelas `Titik3D`

INHERITANCE

- Bahasa pemrograman Java dan PHP hanya mendukung pewarisan tunggal (single inheritance) yaitu kelas turunan atau kelas anak hanya dapat diturunkan dari sebuah kelas dasar atau kelas orangtua.
- Bahasa C++ mendukung pewarisan lebih dari satu atau pewarisan ganda atau jamak (multiple inheritance dimana sebuah kelas turunan atau kelas anak dapat diturunkan dari satu atau lebih kelas dasar atau kelas orangtua

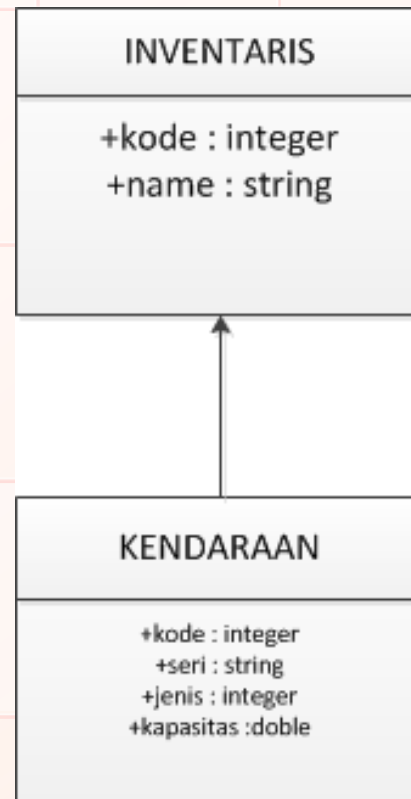


INHERITANCE

- Jika orangtua kelas anak lebih dari satu kelas, maka semua metode dan atribut yang bersifat public dan protected dari semua kelas orangtua akan dimiliki secara otomatis oleh kelas anak
- Metode dan atribut kelas turunan atau kelas anak tidak dapat diakses langsung pada kelas dasar atau orangtua tanpa mendeklarasikan perintah pembuatan objek anak (*new* objek anak)

INHERITANCE

Buatlah kelas-kelas pewarisan sesuai dengan gambar berikut ini



INHERITANCE

Buatlah kelas-kelas pewarisan sesuai dengan gambar berikut ini

