19

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN KE-13

****

**Disusun oleh :**

**NAMA : FERDI DIRGANTARA**

**NIM : 175410039**

**JURUSAN : TEKNIK INFORMATIKA**

**JENJANG : S1**

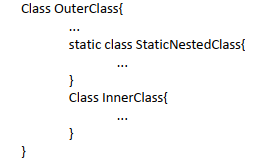
LABORATORIUM TERPADU

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTERAKAKOMYOGYAKARTA

2019

1. **TEORI SINGKAT**

Java mengijinkan kita mendefiniskan suatu kelas di dalam kelas lain, hal ini disebut sebagai *nested class*. Disebut nested karena class bersifat tersarang terhadap kelas – kelas utamanya, seperti halnya blok penyeleksian (if, for) yang tersarang pada blok penyeleksian lainnya atau method yang tersarang pada method lainnya. *nested class* dibagi menjadi dua kategori, yaitu *static* dan *non static nested class*. Istilah *inner class* biasa digunakan untuk *non static nested  
class*.

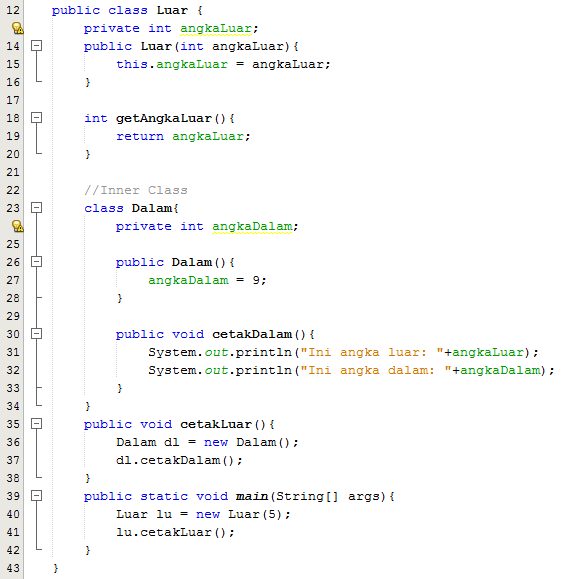


Nested class merupakan member dari kelas Outer(kelas terluar). Non static nested class (inner class) dapat mengakses semua member dari kelas Outer bahkan yang mempunyai hak akses private. Static nested class tidak mempunyai akses kepada member dari kelas Outer. Sebagai member dari Outer class, sebuah nested class dapat dideklarasikan dengan hak akses private, public, proteced, sedangkan Outer class hanya bisa dideklarasikan dengan hak akses  
public.

Anonymous inner class adalah sebuah inner class yang dideklarasikan tanpa memberikan nama kelas. Anonymous class akan membuat kode program menjadi lebih ringkas karena dengan Anonymous class kita dapat mendeklarasikan dan meng- instant suatu kelas sekaligus dalam satu langkah.

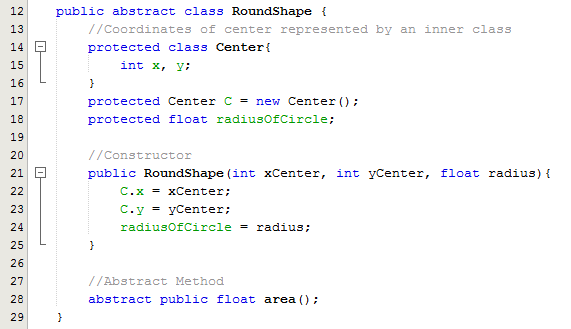
1. **PEMBAHASAN LISTING**

Praktik 1



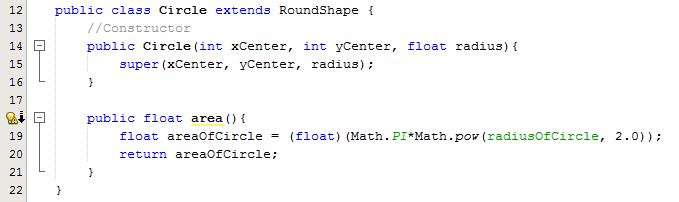
Nested class yang terjadi pada program diatas ditunjukkan dengan kelas **Luar** yang menjadi outer class dan kelas **Dalam** yang menjadi inner class. Pada saat program dijalankan program akan menampilkan statement yang terdapat didalam method **cetakDalam** yang ditulis pada inner class (Kelas **Dalam**).

Praktik 2



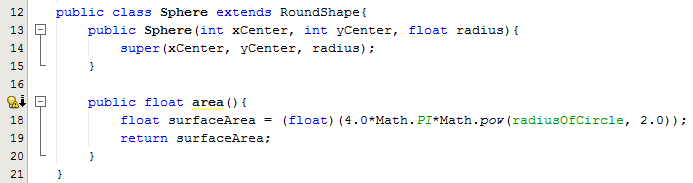
Kelas abstrak **RoundShape** memiliki inner class yaitu kelas **Center** dengan attribut **radiusOfCircle** yang digunakan untuk menyimpan nilai radius lingkaran dan juga terdapat method abstract **area** pada program diatas.

Praktik 3



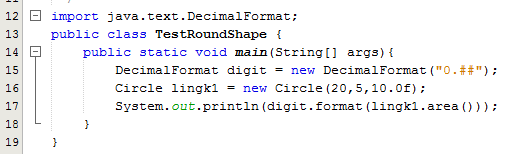
Kelas **Circle** diatas merupaka kelas turunan dari kelas **RoundShape** yang didalamnya terdapat konstruktor untuk kelas **Circle** itu sendiri untuk memberikan nilai pada attribut **xCenter, yCenter** dan **radius** yang terdapat pada kelas induk, dan overriding pada method **area** untuk menghitung luas lingkaran.

Praktik 4



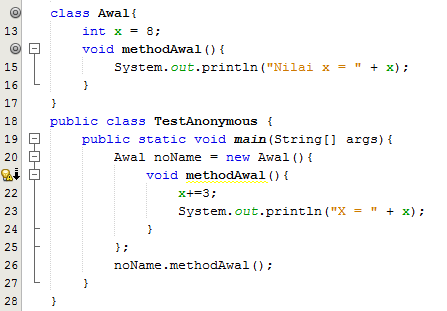
Kelas **Sphere** diatas juga merupakan kelas turunan dari kelas **RoundShape** dan memiliki konstruktor dan method yang sama seperti kelas **Circle** sebelumnya.

Praktik 5



Kelas **TestRoundShape** diatas digunakan untuk menguji kelas **Circle** dan menampilkan hasil perhitungan yang dilakukan pada metho **area**.

Praktik 6



Program diatas akan menampilkan output X=11 yang mana nilai 11 tersebut diperoleh dari penjumlahan 8 dengan 3 dan kemudian ditampilkan melalui method **methodAwal** yang terdapat pada kelas **TestAnonymous**.

1. **PEMBAHASAN LATIHAN**
2. **TUGAS**
3. **KESIMPULAN**
4. **LISTING**

*Terlampir*