



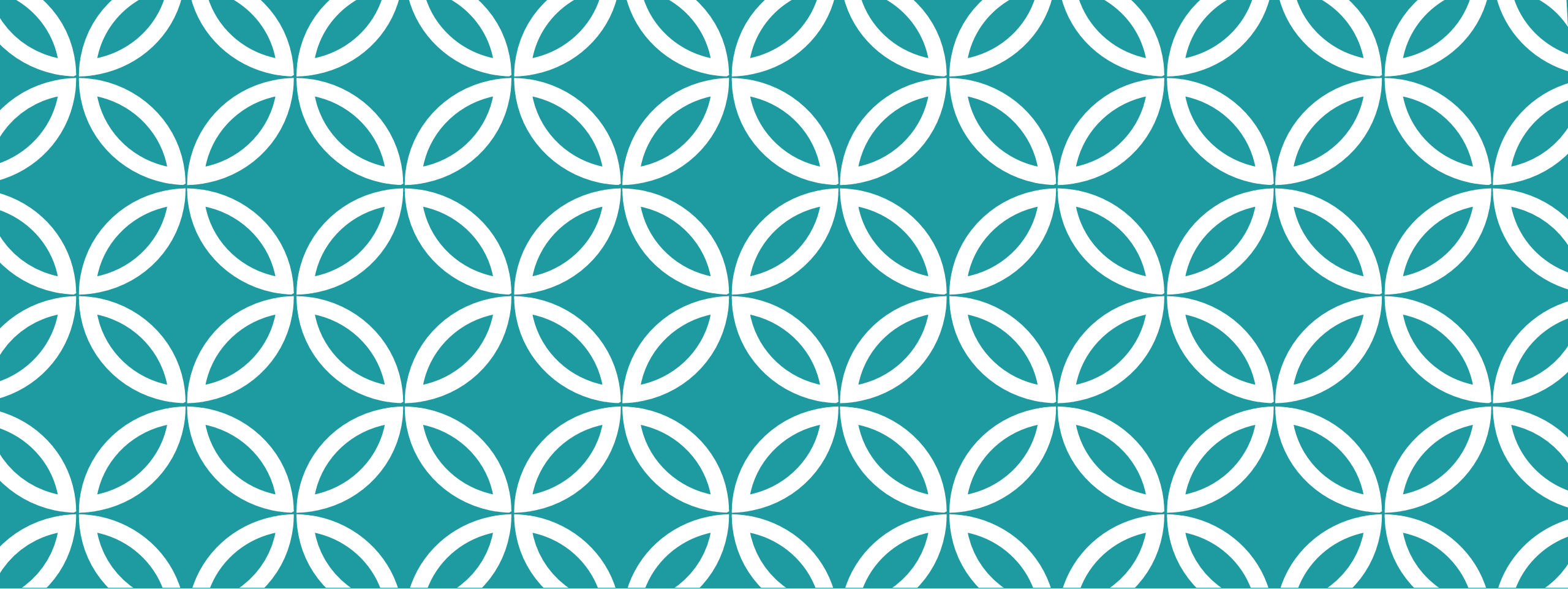
METHOD AND VARIABLE SCOPE

**Kuliah Algoritma &
Pemrograman**

Renny P. Kusumawardani, S.T.,
M.T., SCJP

TODAY ON BAHASA PEMROGRAMAN

- Pembuatan method
- Pembuatan objek (sedikit)
- Input dan Output



METHOD



PENGENALAN METHOD

- Method berfungsi untuk “membungkus” sekumpulan pernyataan perintah (statements)
- Statements yang terbungkus pada suatu method merupakan langkah-langkah untuk mewujudkan aksi / hasil tertentu
- Contoh:
 void makan() { ... }
 int hitungPangkat(int basis, int pangkat) { ... }

MANFAAT METHOD

- You don't have to do the typing over and over again
- It's less error prone
 - begitu sudah dibuat sekali, dapat dipergunakan berulang kali

STRUKTUR DASAR METHOD

Deklarasi method terdiri dari:

0. Pernyataan hak akses method dan apakah method bersifat static/tidak
1. **Tipe data** nilai balikan yang akan dihasilkan oleh method
2. **Nama** method
3. **Daftar argumen** dalam kurung buka – kurung tutup
4. Pembungkus statements pada method: **kurung keriting buka – kurung keriting tutup**
5. **Daftar statements** pada method
 - berada di antara kurung keriting buka – kurung keriting tutup

STRUKTUR DASAR METHOD (2)

0
Modifiers

1
Tipe data balikan

2
Nama

3
(Daftar argumen)

4
{

```
Statement1;  
Statement2;  
Statement3;  
...  
...  
statementN;  
return nilai_dengan_tipe_data_balikan;
```

4
}

CONTOH

Fungsi untuk menghitung $a + b$

- Masing-masing bertipe data double
- Menghasilkan balikan bertipe data double
- Dapat diakses secara public (dapat diakses oleh siapa saja)

```
public double tambahAB (double a, double b) {  
    double hasil = a + b;  
    return hasil;  
}
```

```
public double tambahAB (double a, double b) {  
    double hasil = a + b;  
    return hasil;  
}
```


KALKULATOR GRAVITASI, REVISITED

$$x(t) = 0.5 \times at^2 + v_i t + x_i$$

Previously, we have these statements:

```
double a = -10;  
double t = 10;  
double vi = 0;  
double xi = 0;  
double xt = 0.5*a*t*t;  
xt = xt + vi*t;  
xt = xt + xi;
```

Bagaimana bila kita ingin menghitung posisi benda dengan nilai t , atau v_i , atau x_i yang berbeda?

KALKULATOR GRAVITASI DENGAN METHOD

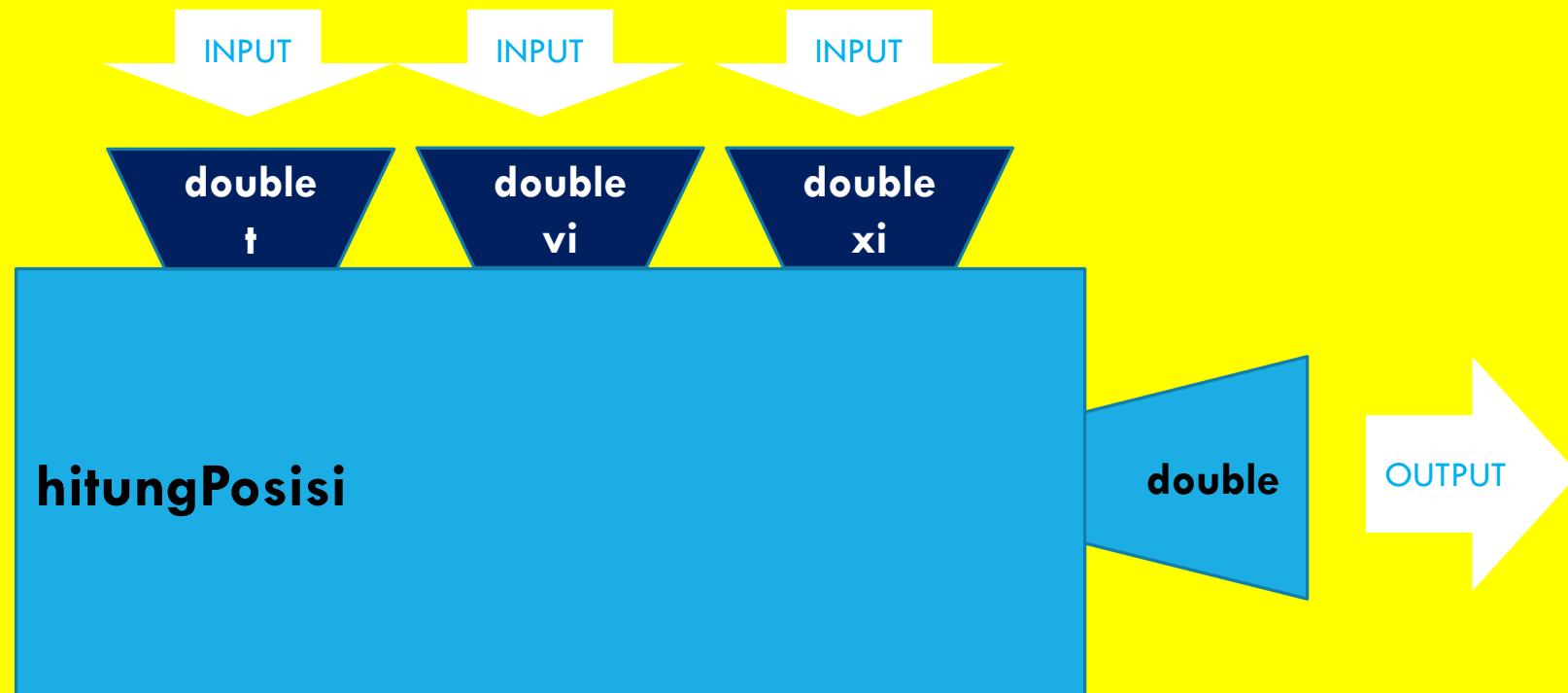
Dengan menggunakan method, susunan statements di atas dapat kita “bungkus” sebagai berikut:

```
double hitungPosisi(double t, double vi, double xi) {  
    double a = -10;  
    double xt = 0.5*a*t*t + vi*t + xi;  
    return xt;  
}
```

VOILA! Kita dapat menghitung posisi benda dengan nilai t , v_i , dan x_i yang berbeda-beda

WHAT IS A METHOD? A VISUAL AID

- Bayangkan sebuah method seperti sebuah alat
- Yang perlu diketahui pengguna hanyalah: apa tipe masukannya, serta apa tipe hasilnya



MENGGUNAKAN METHOD

Bagaimana anda dapat menggunakan method?

1. Method bisa jadi merupakan **anggota dari suatu instance kelas (objek)**, atau
2. Method bisa jadi merupakan **anggota static dari definisi suatu kelas**

Pada kasus 1, untuk menggunakan method kita memerlukan referensi ke suatu objek, sedangkan pada kasus 2 method dapat dipanggil secara langsung, namun memerlukan modifier static

PREVIOUSLY ON KALKULATOR GRAVITASI

```
class KalkulatorGravitasi {  
  
    public static void main(String[] args) {  
  
        double a = -10;  
        double t = 10;  
        double vi = 0;  
        double xi = 0;  
        double xt = 0.5*a*t*t + vi*t + xi;  
        System.out.println("Posisi suatu objek setelah " + t + " detik adalah " + xt + " m.");  
    }  
}
```

MENGGUNAKAN METHOD (2)

Kasus 1:

```
class KalkulatorGravitasi {  
  
    double hitungPosisi(double t, double vi, double xi) {  
        double a = -10;  
        double xt = 0.5*a*t*t + vi*t + xi;  
        return xt;  
    }  
  
    public static void main(String[] arguments) {  
        KalkulatorGravitasi kg = new KalkulatorGravitasi();  
        double t = 30;  
        double posisi = kg.hitungPosisi(t, 10, 0);  
        System.out.println("Posisi suatu objek setelah " + t + " detik adalah " + posisi  
        + " m.");  
    }  
}
```

MENGGUNAKAN METHOD (3)

Kasus 2:

```
class KalkulatorGravitasi {  
  
    static double hitungPosisi(double t, double vi, double xi) {  
        double a = -10;  
        double xt = 0.5*a*t*t + vi*t + xi;  
        return xt;  
    }  
  
    public static void main(String[] arguments) {  
        double t = 30;  
        double posisi = hitungPosisi(t, 10, 0);  
        System.out.println("Posisi suatu objek setelah " + t + " detik adalah " + posisi  
        + " m.");  
    }  
}
```

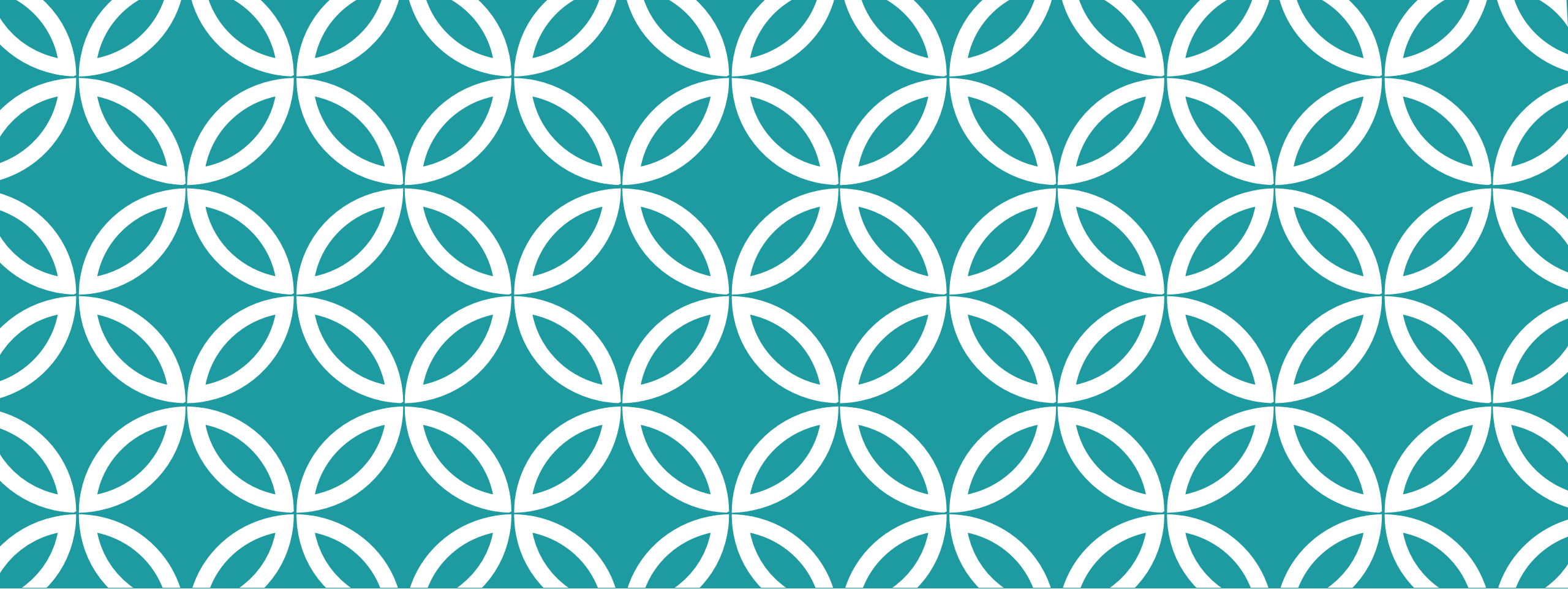
MENGAPA ADA DUA JENIS PEMANGGILAN METHOD?

1. Method sebagai anggota objek spesifik dari kelas

- Seringkali method merupakan ‘aksi/tindakan’ spesifik terhadap instance objek kelas bersangkutan
- Contoh: melakukan modifikasi pada nilai variabel anggota objek

2. Method sebagai anggota dari kelas itu sendiri

- Berguna untuk kasus dimana benar-benar hanya fungsionalitasnya saja yang kita butuhkan
- Biasanya untuk dipanggil dari kelas-kelas lain



VARIABLE SCOPE



VARIABLE SCOPE

- Tempat hidup variabel adalah di dalam blok statement ({}), tempat variabel tersebut didefinisikan (scope)
 - Variabel tersebut masih dapat diakses sepanjang blok statement tersebut masih sedang dieksekusi
 - Begitu keluar dari blok, variabel sudah tidak dapat diakses
- Parameter/argumen dari method adalah seperti mendefinisikan variabel baru di dalam method

CONTOH VARIABLE SCOPE

```
class KalkulatorGravitasi {
```

```
    double hitungPosisi(double t, double vi, double xi) {  
        double a = -10;  
        double xt = 0.5*a*t*t + vi*t + xi;  
        return xt;  
    }
```

```
    public static void main(String[] args) {  
        KalkulatorGravitasi myKG = new KalkulatorGravitasi();  
        double waktuTempuh = 30;  
        double posisi = myKG.hitungPosisi(waktuTempuh, 10, 0);  
        System.out.println("Posisi objek setelah " + waktuTempuh + " detik adalah " + xt);  
    }
```

```
}
```

Mengapa program ini error?