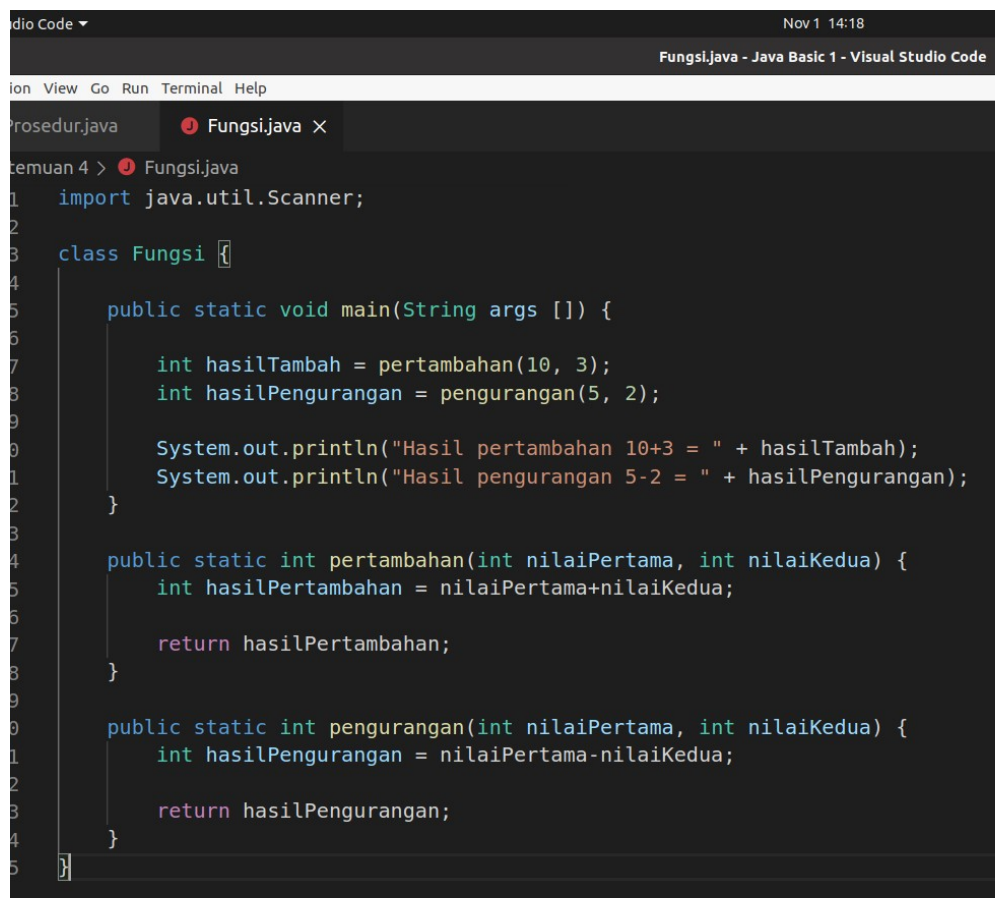


# Modul Pemrograman Berbasis Objek (PBO)

## Class Dan Object

Setelah pada modul modul sebelum nya kita mempelajari basic dari bahasa pemrograman java, saat ini kita akan memulai membahas tentang pbo itu sendiri atau nama keren nya object oriented programming (oop)

Sebelum membahas apa itu oop, mari kita lihat kodingan pada modul fungsi



```
1 import java.util.Scanner;
2
3 class Fungsi {
4     public static void main(String args []) {
5         int hasilTambah = pertambahan(10, 3);
6         int hasilPengurangan = pengurangan(5, 2);
7
8         System.out.println("Hasil pertambahan 10+3 = " + hasilTambah);
9         System.out.println("Hasil pengurangan 5-2 = " + hasilPengurangan);
10    }
11
12    public static int pertambahan(int nilaiPertama, int nilaiKedua) {
13        int hasilPertambahan = nilaiPertama+nilaiKedua;
14
15        return hasilPertambahan;
16    }
17
18    public static int pengurangan(int nilaiPertama, int nilaiKedua) {
19        int hasilPengurangan = nilaiPertama-nilaiKedua;
20
21        return hasilPengurangan;
22    }
23 }
```

Pada kodingan diatas, kita membuang fungsi static dari pertambahan dan pengurangan. Tujuan pembuatan fungsi tersebut agar jika kita ingin menggunakan berkali kali kita tidak perlu menulis kode tersebut secara berulang kali, cukup 1x dibuat fungsi maka kode tersebut dapat dipanggil berkali kali.

Nah, sekarang bagaimana semisal teman teman telah membuat sampai 100 fungsi, dimana tiap fungsinya mempunyai tugas khusus masing masing ?

Walau fungsi itu menghemat teman teman dalam penulisan kode, tetapi jika teman teman membuat sampai 100 fungsi atau lebih tentunya hal ini akan membuat teman teman sulit melakukan perawatan/perbaikan pada kode tersebut.

Terus bagaimana solusi nya ?

Bayangkan teman teman dapat membuat pengelompokan fungsi fungsi tersebut dan disimpan pada file yang berbeda

Semisal

Fungsi penambahan, pengurangan, perkalian pembagian masuk kedalam kelompok aritmatika dan disimpan pada file Aritmatika.java

Fungsi menghitung mean, median, modus, standar deviasi, dan varians masuk ke dalam kelompok statistika dan disimpan pada file Statistika.java

Nah setelah teman teman melakukan pengelompokan seperti itu, teman teman tinggal mengakses masing masing kelompok jika ingin menggunakan salah satu fungsi yang terdapat pada kelompok tersebut

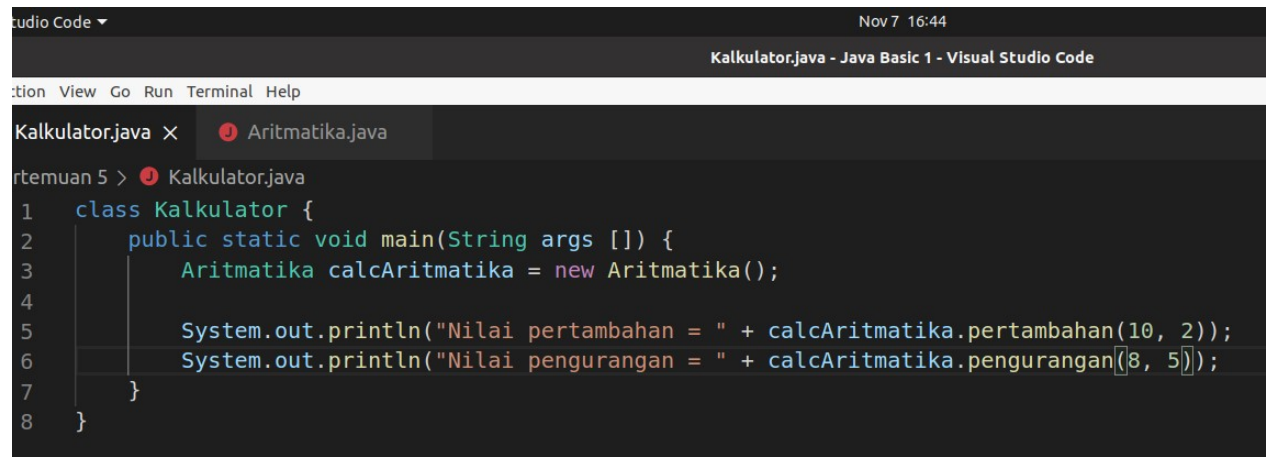
Hal inilah yang kita sebut sebagai class, mari kita mulai bikin contoh kodingan diatas

Pertama, buatlah suatu folder bernama kalkulator

Pada folder tersebut buatlah File Aritmatika.java dan tulis kode di bawah ini

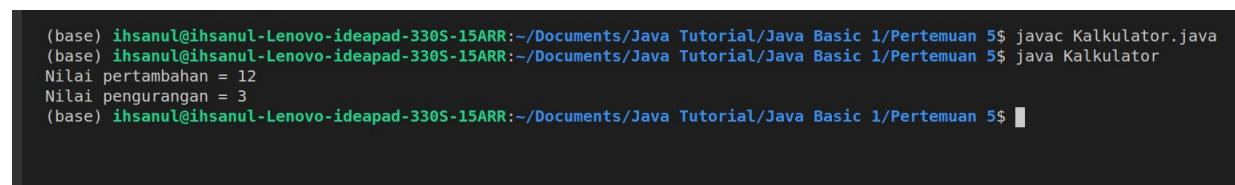
```
class Aritmatika {  
    public int pertambahan(int angkaPertama, int angkaKedua) {  
        return angkaPertama+angkaKedua;  
    }  
  
    public int pengurangan(int angkaPertama, int angkaKedua) {  
        return angkaPertama-angkaKedua;  
    }  
}
```

Lalu, buat File Kalkulator.java masih di folder yang sama, dan tulis kode dibawah ini didalam file Kalkulator.java



```
1 class Kalkulator {
2     public static void main(String args []) {
3         Aritmatika calcAritmatika = new Aritmatika();
4
5         System.out.println("Nilai pertambahan = " + calcAritmatika.pertambahan(10, 2));
6         System.out.println("Nilai pengurangan = " + calcAritmatika.pengurangan(8, 5));
7     }
8 }
```

Jalankan file Kalkulator.java pada terminal



```
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 5$ javac Kalkulator.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 5$ java Kalkulator
Nilai pertambahan = 12
Nilai pengurangan = 3
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 5$
```

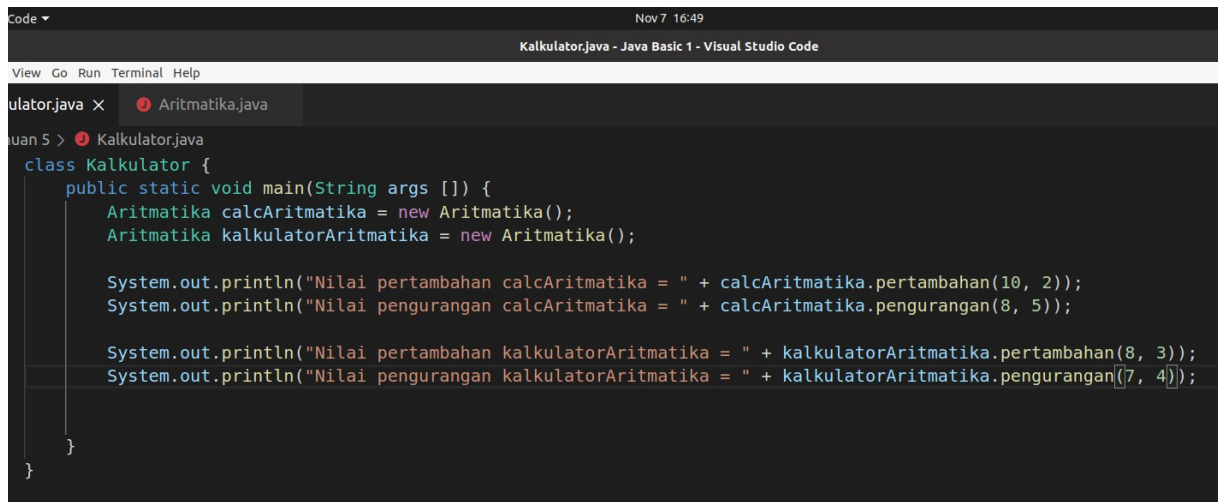
Seperti yang teman teman liat, kita tidak membuat fungsi pertambahan pada file Kalkulator.java, tetapi kenapa pertambahan ini bisa dilakukan?

Alasan nya, karena kita telah membuat class Aritmatika, dan memanggil class Aritmatika pada Kalkulator.java dan menyimpan nya pada variabel calcAritmatika

calcAritmatika ini dapat kita sebut sebagai objek atau instance. Objek adalah sesuatu yang menampung class pada modul lain. Variabel calcAritmatika menampung class Aritmatika yang terdapat pada file Aritmatika.java, sehingga kita dapat menggunakan berbagai fungsi yang ada pada class Aritmatika didalam class Kalkulator yang telah kita buat.

Lalu, apakah class Aritmatika hanya bisa ditampung lebih dari 1 variabel ? Jawaban nya, bisa

Seperti fungsi yang dapat digunakan berkali kali, kita juga dapat menampung class Aritmatika berkali kali pada variabel yang berbeda beda



```
Code Nov 7 16:49
Kalkulator.java - Java Basic 1 - Visual Studio Code

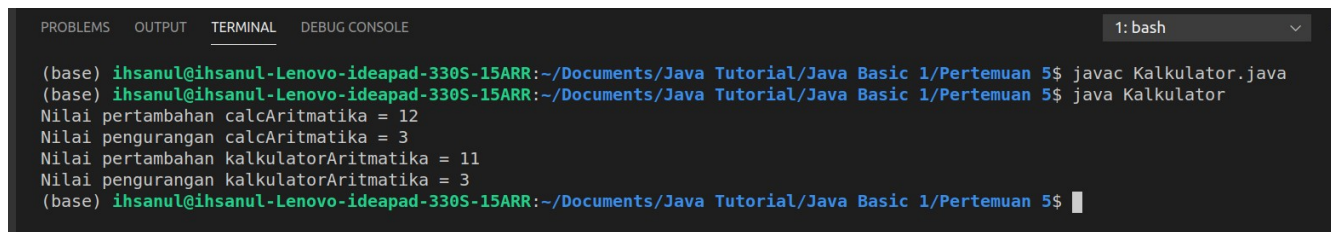
View Go Run Terminal Help

Kalkulator.java x Aritmatika.java

uan 5 > Kalkulator.java
class Kalkulator {
    public static void main(String args []) {
        Aritmatika calcAritmatika = new Aritmatika();
        Aritmatika kalkulatorAritmatika = new Aritmatika();

        System.out.println("Nilai pertambahan calcAritmatika = " + calcAritmatika.pertambahan(10, 2));
        System.out.println("Nilai pengurangan calcAritmatika = " + calcAritmatika.pengurangan(8, 5));

        System.out.println("Nilai pertambahan kalkulatorAritmatika = " + kalkulatorAritmatika.pertambahan(8, 3));
        System.out.println("Nilai pengurangan kalkulatorAritmatika = " + kalkulatorAritmatika.pengurangan(7, 4));
    }
}
```

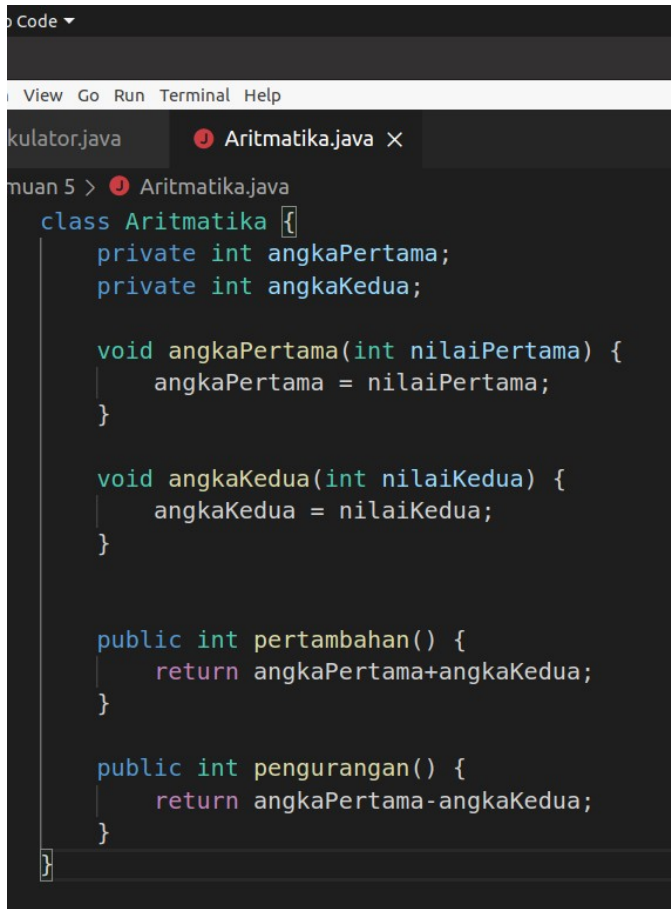


```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash

(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 5$ javac Kalkulator.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 5$ java Kalkulator
Nilai pertambahan calcAritmatika = 12
Nilai pengurangan calcAritmatika = 3
Nilai pertambahan kalkulatorAritmatika = 11
Nilai pengurangan kalkulatorAritmatika = 3
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 5$
```

Dapat kita lihat diatas, kita membuat sebuah objek baru bernama kalkulatorAritmatika yang menampung class Aritmatika didalam nya. Hal ini membuat kita juga dapat menggunakan fungsi yang ada didalam class Aritmatika pada variabel kalkulatorAritmatika

Terdapat pertanyaan lain, selain fungsi, apakah kita dapat menaruh variabel pada class Aritmatika? Tentu saja bisa juga



```
class Aritmatika {
    private int angkaPertama;
    private int angkaKedua;

    void angkaPertama(int nilaiPertama) {
        angkaPertama = nilaiPertama;
    }

    void angkaKedua(int nilaiKedua) {
        angkaKedua = nilaiKedua;
    }

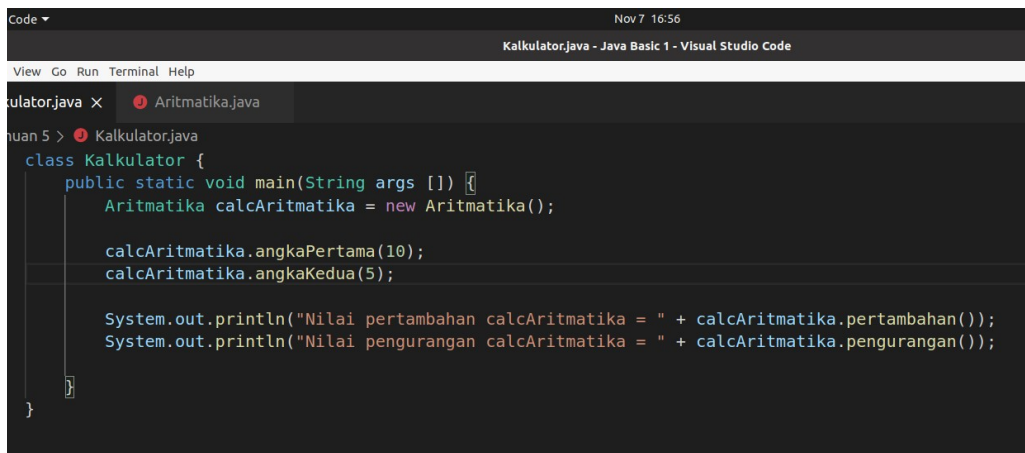
    public int pertambahan() {
        return angkaPertama+angkaKedua;
    }

    public int pengurangan() {
        return angkaPertama-angkaKedua;
    }
}
```

Seperti yang kita lihat diatas, kita dapat menaruh variabel pada class Aritmatika. Variabel tersebut kita namakan sebagai angkaPertama dan angkaKedua. Lalu, untuk mengisi variabel tersebut, kita membuat suatu prosedur baru bernama angkaPertama dan angkaKedua juga.

Dan dapat kita lihat juga, pada fungsi pertambahan dan pengurangan kita menggunakan variabel angkaPertama dan angkaKedua.

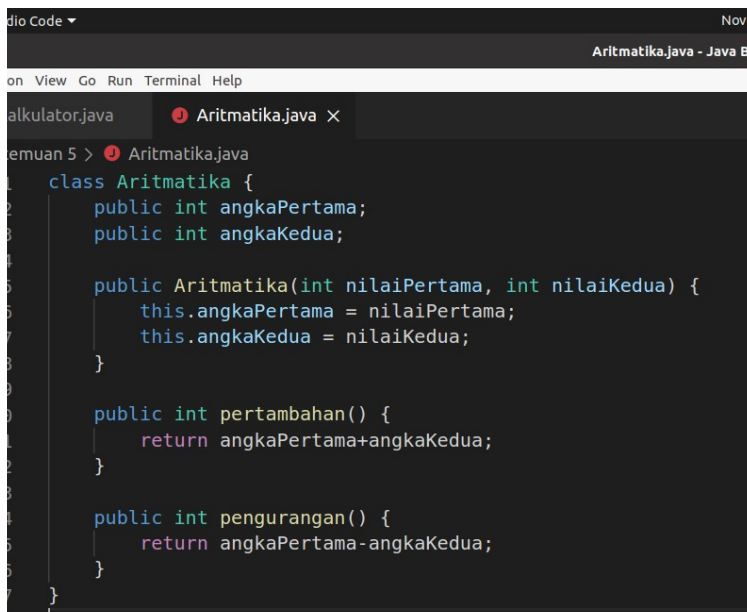
Kembali pada file Kalkulator.java

A screenshot of the Visual Studio Code editor. The top bar shows 'Code' and 'Nov 7 16:56'. The title bar says 'Kalkulator.java - Java Basic 1 - Visual Studio Code'. The editor has two tabs: 'Kalkulator.java' (active) and 'Aritmatika.java'. The code in Kalkulator.java is as follows:

```
1  class Kalkulator {  
2      public static void main(String args []) {  
3          Aritmatika calcAritmatika = new Aritmatika();  
4  
5          calcAritmatika.angkaPertama(10);  
6          calcAritmatika.angkaKedua(5);  
7  
8          System.out.println("Nilai pertambahan calcAritmatika = " + calcAritmatika.pertambahan());  
9          System.out.println("Nilai pengurangan calcAritmatika = " + calcAritmatika.pengurangan());  
10     }  
11 }
```

Kembali pada file Kalkulator.java, kita kali ini memanggil fungsi angkaPertama dan angkaKedua, sehingga variabel yang terdapat pada class Aritmatika terisi dengan nilai 10 dan 5. Setelah terisi, kita memanggil fungsi pertambahan dan pengurangan tanpa perlu memasukkan parameter karena fungsi pertambahan dan pengurangan menggunakan angka dari variabel angkaPertama dan angkaKedua

Terdapat cara lain juga untuk mengisi variabel pada class Aritmatika

A screenshot of the Visual Studio Code editor. The top bar shows 'Visual Studio Code' and 'Nov 7 16:56'. The title bar says 'Aritmatika.java - Java Basic 1 - Visual Studio Code'. The editor has two tabs: 'Kalkulator.java' and 'Aritmatika.java' (active). The code in Aritmatika.java is as follows:

```
1  class Aritmatika {  
2      public int angkaPertama;  
3      public int angkaKedua;  
4  
5      public Aritmatika(int nilaiPertama, int nilaiKedua) {  
6          this.angkaPertama = nilaiPertama;  
7          this.angkaKedua = nilaiKedua;  
8      }  
9  
10     public int pertambahan() {  
11         return angkaPertama+angkaKedua;  
12     }  
13  
14     public int pengurangan() {  
15         return angkaPertama-angkaKedua;  
16     }  
17 }
```

Kali ini, kita tidak membuat prosedur seperti cara pertama. Tetapi, kita menggunakan sesuatu bernama constructor. Constructor akan otomatis tercipta pertama kali ketika program dijalankan, tetapi kita bisa juga membuatnya secara manual dengan cara menuliskan nama Constructor tersebut sama persis dengan nama class yang dibuat dan menyuruh constructor tersebut untuk membuat kita dapat mengisi variabel angkaPertama dan angkaKedua



Notes :

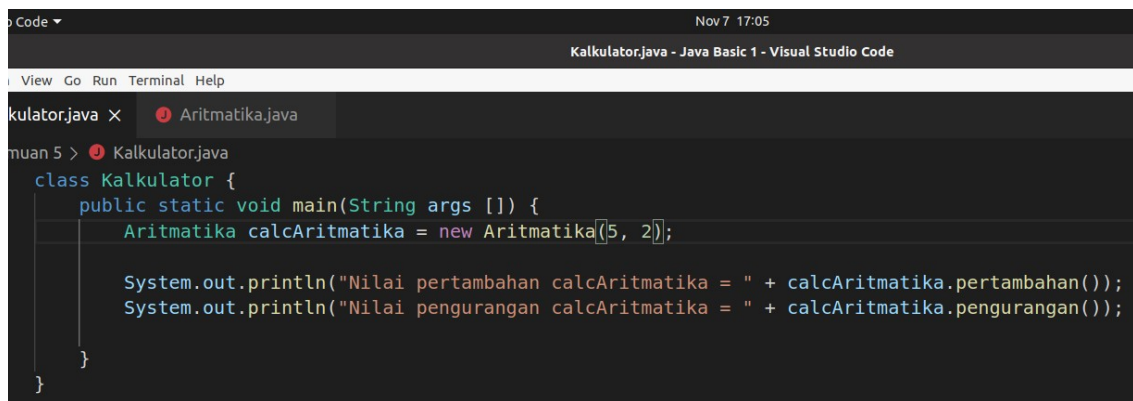
---

Terdapat kata this pada constructor, apa itu this ?

this digunakan untuk merujuk variable yang berada diluar fungsi pada class. Jadi kata this diatas, merujuk kepada variable angkaPertama dan angkaKedua yang berada diluar fungsi constructor.

---

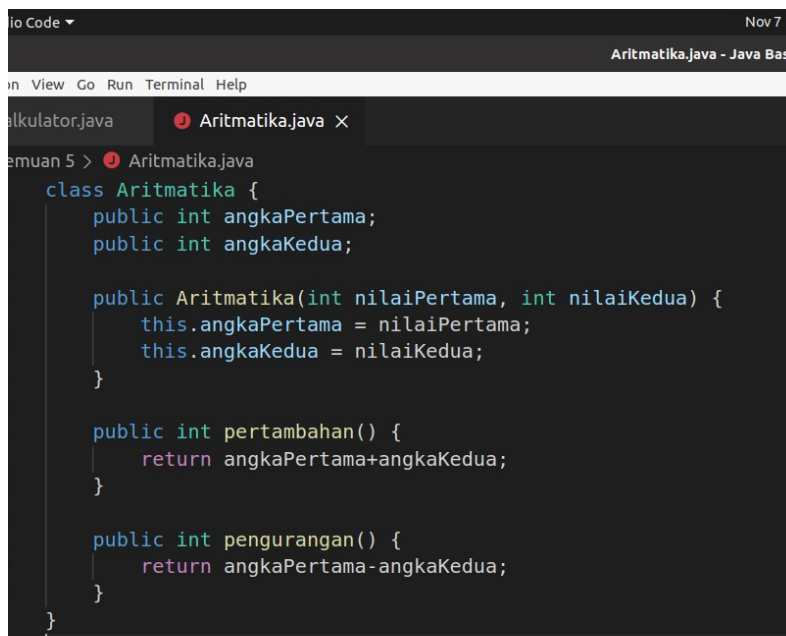
Mari kita lihat pada file Kalkulator.java



```
Nov 7 17:05
Kalkulator.java - Java Basic 1 - Visual Studio Code
File View Go Run Terminal Help
Kalkulator.java x Aritmatika.java
Kalkulator.java
1 1. Kalkulator.java
2  class Kalkulator {
3      public static void main(String args []) {
4          Aritmatika calcAritmatika = new Aritmatika(5, 2);
5
6          System.out.println("Nilai pertambahan calcAritmatika = " + calcAritmatika.pertambahan());
7          System.out.println("Nilai pengurangan calcAritmatika = " + calcAritmatika.pengurangan());
8      }
9  }
```

pada variabel calcAritmatika, kita menambahkan angka 5 dan 2 sebagai parameter class Aritmatika nya. 5 dan 2 ini akan terisi secara otomatis pada constructor dan disimpan pada variabel angkaPertama dan angkaKedua.

Mari kita lihat kembali file Aritmatika.java



```
Nov 7
Aritmatika.java - Java Basic 1 - Visual Studio Code
File View Go Run Terminal Help
Kalkulator.java x Aritmatika.java x
Aritmatika.java
1 1. Aritmatika.java
2  class Aritmatika {
3      public int angkaPertama;
4      public int angkaKedua;
5
6      public Aritmatika(int nilaiPertama, int nilaiKedua) {
7          this.angkaPertama = nilaiPertama;
8          this.angkaKedua = nilaiKedua;
9      }
10
11     public int pertambahan() {
12         return angkaPertama+angkaKedua;
13     }
14
15     public int pengurangan() {
16         return angkaPertama-angkaKedua;
17     }
18 }
```

Selama ini kita telah menyebut variabel, fungsi dan prosedur. Tetapi, pada oop terdapat sebutan lain, variabel dapat kita sebut sebagai properti. Fungsi dan prosedur dapat kita sebut sebagai method. Dan sebutan tersebut yang akan kita gunakan ketika menerapkan class dan objek.

Notes :

--

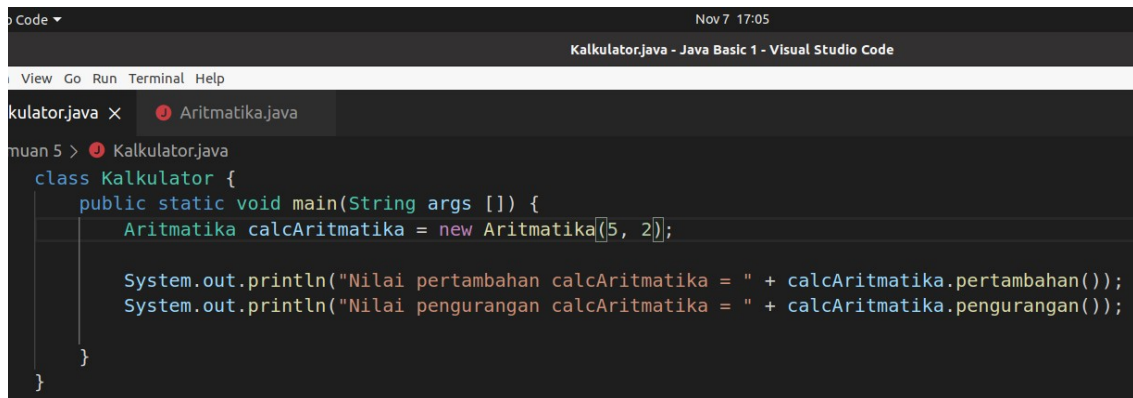
Pada beberapa bahasa pemrograman lain, ketika kita menggunakan suatu class maka kita telah menerapkan oop dan semisal terdapat variabel dan fungsi pada class tersebut maka kita tidak menggunakan sebutan variabel dan fungsi melainkan properti dan method, sedangkan kode yang tidak menggunakan class maka tetap disebut sebagai variabel dan fungsi/prosedur.

Pada bahasa pemrograman java, keseluruhan kode yang kita buat menggunakan class dan objek. Hal ini disebabkan karena java menerapkan full oop pada bahasa pemrograman nya, untuk modul modul berikutnya, kita akan menggunakan properti dan method ketimbang variabel dan fungsi

--

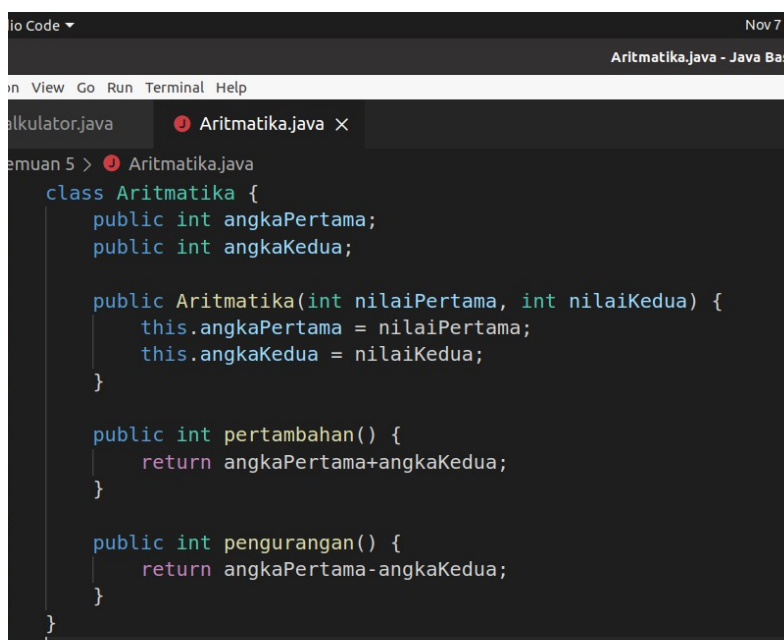


Modul class dan objek selesai sampai disini, tetapi mungkin kalian memiliki beberapa pertanyaan sebagai berikut



```
Nov 7 17:05
Kalkulator.java - Java Basic 1 - Visual Studio Code
View Go Run Terminal Help
Kalkulator.java x Aritmatika.java
Kalkulator.java
muan 5 > Kalkulator.java
class Kalkulator {
    public static void main(String args []) {
        Aritmatika calcAritmatika = new Aritmatika(5, 2);

        System.out.println("Nilai pertambahan calcAritmatika = " + calcAritmatika.pertambahan());
        System.out.println("Nilai pengurangan calcAritmatika = " + calcAritmatika.pengurangan());
    }
}
```



```
Nov 7
Aritmatika.java - Java Bas
View Go Run Terminal Help
Kalkulator.java x Aritmatika.java x
Aritmatika.java
muan 5 > Aritmatika.java
class Aritmatika {
    public int angkaPertama;
    public int angkaKedua;

    public Aritmatika(int nilaiPertama, int nilaiKedua) {
        this.angkaPertama = nilaiPertama;
        this.angkaKedua = nilaiKedua;
    }

    public int pertambahan() {
        return angkaPertama+angkaKedua;
    }

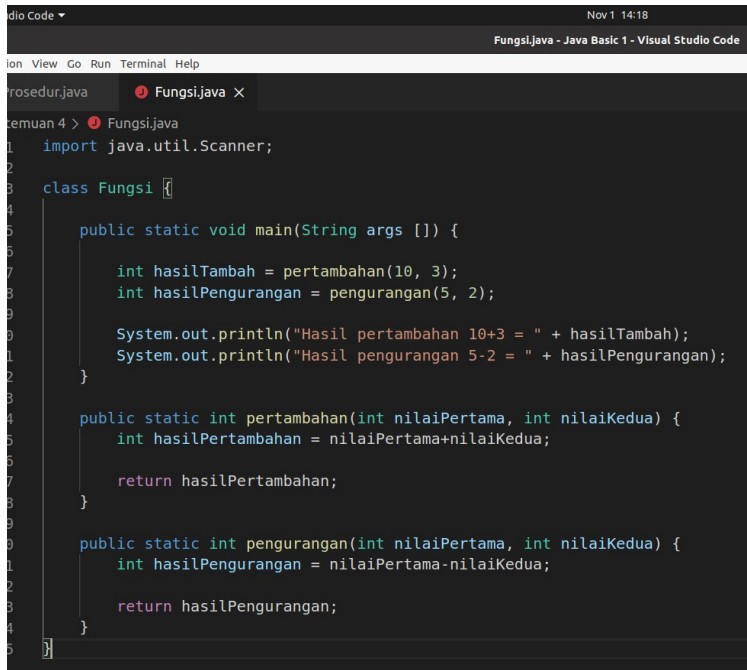
    public int pengurangan() {
        return angkaPertama-angkaKedua;
    }
}
```

Pada class Kalkulator terdapat static void main, kenapa pada class Aritmatika tidak terdapa hal tersebut?

static void main tersebut menandakan bahwa file utama/yang pertama kali dijalankan adalah class Kalkulator , sedangkan class Aritmatika hanya sebagai pendukung/modul yang dapat digunakan pada class Kalkulator. Kalian tidak dapat menjalankan class Aritmatika, tanpa adanya class utama dan akan mengakibatkan error.

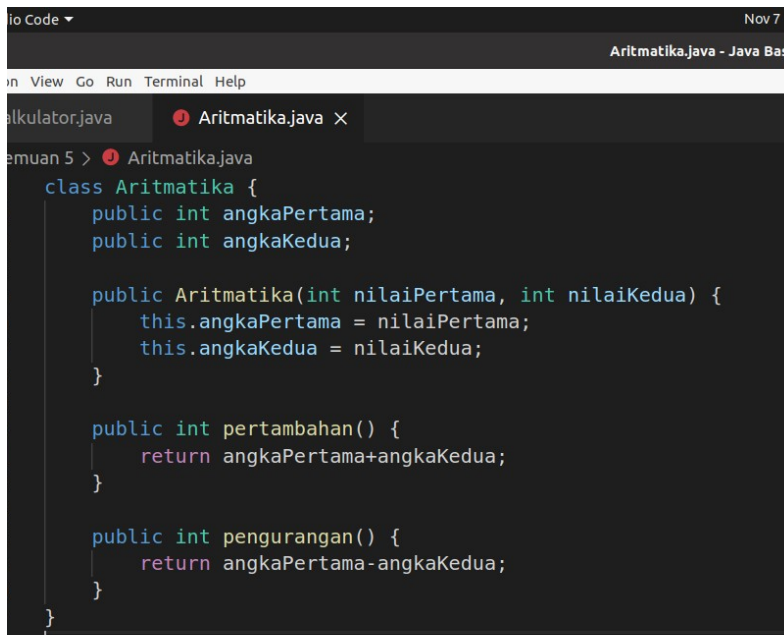
Lalu, pertanyaan berikutnya. Pada modul fungsi, kita menggunakan tulisan static untuk membuat fungsi, tetapi pada file Aritmatika.java kita tidak menggunakan kata kata static pada pembuatan method

Contoh, pada file Fungsi.java di modul pekan lalu :



```
1 import java.util.Scanner;
2
3 class Fungsi {
4
5     public static void main(String args []) {
6
7         int hasilTambah = pertambahan(10, 3);
8         int hasilPengurangan = pengurangan(5, 2);
9
10        System.out.println("Hasil pertambahan 10+3 = " + hasilTambah);
11        System.out.println("Hasil pengurangan 5-2 = " + hasilPengurangan);
12    }
13
14    public static int pertambahan(int nilaiPertama, int nilaiKedua) {
15        int hasilPertambahan = nilaiPertama+nilaiKedua;
16
17        return hasilPertambahan;
18    }
19
20    public static int pengurangan(int nilaiPertama, int nilaiKedua) {
21        int hasilPengurangan = nilaiPertama-nilaiKedua;
22
23        return hasilPengurangan;
24    }
25 }
```

File Aritmatika.java :



```
1 class Aritmatika {
2     public int angkaPertama;
3     public int angkaKedua;
4
5     public Aritmatika(int nilaiPertama, int nilaiKedua) {
6         this.angkaPertama = nilaiPertama;
7         this.angkaKedua = nilaiKedua;
8     }
9
10    public int pertambahan() {
11        return angkaPertama+angkaKedua;
12    }
13
14    public int pengurangan() {
15        return angkaPertama-angkaKedua;
16    }
17 }
```

Terdapat kata static pada pembuatan fungsi pertambahan dan pengurangan pada file Fungsi.java, sedangkan tidak ada kata static pada pembuatan method pertambahan dan pengurangan pada class Aritmatika

Kata static menandakan bahwa fungsi tersebut dapat langsung dijalankan pada class tersebut, tanpa perlu membuat objek

```
Nov 7 17:05
Kalkulator.java - Java Basic 1 - Visual Studio Code
View Go Run Terminal Help
Kalkulator.java x Aritmatika.java
nuan 5 > Kalkulator.java
class Kalkulator {
    public static void main(String args []) {
        Aritmatika calcAritmatika = new Aritmatika(5, 2);

        System.out.println("Nilai pertambahan calcAritmatika = " + calcAritmatika.pertambahan());
        System.out.println("Nilai pengurangan calcAritmatika = " + calcAritmatika.pengurangan());
    }
}
```

Lalu, mengapa banyak kata public yang telah kita buat pada modul ini? Ini akan kita bahas pada modul modul berikutnya

1. Buatlah laporan terkait praktikum modul PBO - Class Dan Object
2. Pada modul ini, kita baru membuat class Aritmatika. Buatlah 1 class baru bernama Statistik pada file baru bernama Statistik.java dan masukan rumus mean, median, dan modus, lalu gunakan class tersebut pada Kalkulator.java dan jalankan
3. Screenshot kode dan hasil kode yang dijalankan. Masukan hasil screenshot tersebut dan jelaskan.

1. Pada penamaan file gunakan format berikut

Contoh :

## 2. Format laporan bertipe pdf

3. Pada cover praktikum, masukan beberapa hal dibawah ini
  - Praktikum ke berapa
  - Judul Praktikum
  - Nama
  - Nim
4. Tenggat waktu mengerjakan adalah 1 pekan semenjak tugas diberikan