

# **LAPORAN PRAKTIKUM**

## **ALGORITMA DAN PEMROGRAMAN LANJUT**

### **PERTEMUAN KE-9**



**Disusun Oleh :**

**NAMA : Raden Isnawan Argi Aryasatya**

**NIM : 195410257**

**JURUSAN : Informatika**

**JENJANG : S1**

**SEMESTER ANTARA**

**Sekolah Tinggi Manajemen Informatika Komputer**

**AKAKOM**

**YOGYAKARTA**

**2021**

## **PERTEMUAN KE-9 (METHOD)**

### **TUJUAN**

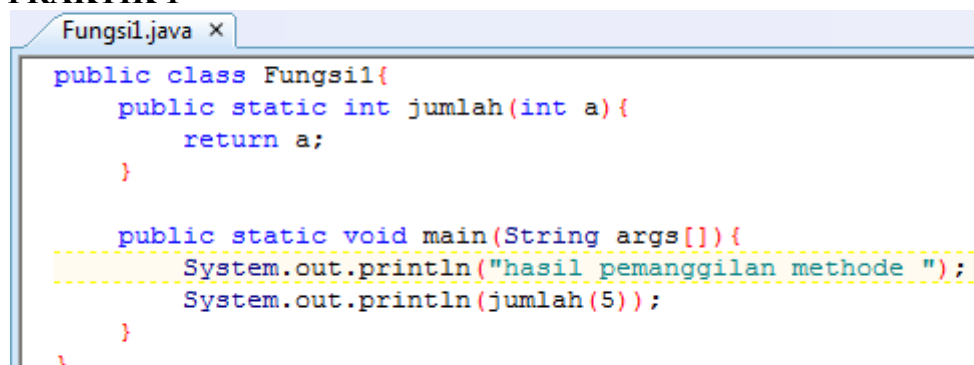
Mahasiswa dapat menyelesaikan kasus dengan menggunakan method dengan parameter, membuat method overloading dan menggunakan method-method bawaan yang ada di java.

### **DASAR TEORI**

1. Method dengan parameter (object). Ada dua buah parameter. Parameter formal adalah parameter yang tertulis dalam definisi method. Parameter aktual parameter yang berada pada inputan langsung pada saat penggunaan method tersebut. Parameter bisa lebih dari satu dengan dipisahkan tanda koma. Yang perlu diperhatikan pada saat pemanggilan method adalah jumlah, urutan dan tipe parameter aktual harus sesuai dengan jumlah urutan dan tipe parameter formal.
2. Overloading. Bahasa java mendukung method overloading , java dapat membedakan beberapa method dengan nama yang sama di dalam sebuah kelas namun parameternya berbeda. Hal ini sangat menguntungkan karena memudahkan kita dalam mengingat nama method.
3. Method bawaan. Contoh:  
Numeric : sqrt,pow,ceil,floor,max,min dan String:StrCpy,Strcat,strlen,equals

---

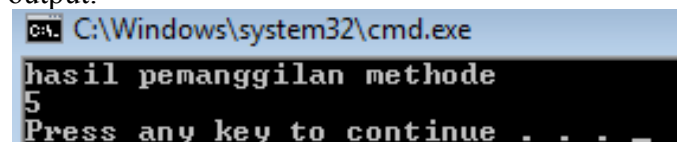
### **PRAKTIK 1**



```
public class Fungsi1{
    public static int jumlah(int a){
        return a;
    }

    public static void main(String args[]){
        System.out.println("hasil pemanggilan method ");
        System.out.println(jumlah(5));
    }
}
```

output:



```
C:\Windows\system32\cmd.exe
hasil pemanggilan method
5
Press any key to continue . . . _
```

penjelasan:

program di atas adalah contoh sederhana sebuah penggunaan sebuah fungsi atau method yang menggunakan parameter. Untuk membuat program tersebut, langsung saja kita awali dengan mendeklarasikan nama class yaitu dengan menuliskan baris kode **public class Fungsi1**. Kemudian kita buat method bernama jumlah dengan menuliskan baris kode **public static int jumlah(int a)**. Method tersebut memiliki parameter bernama a, yang kemudian kita kembalikan nilai valuenya dengan kode **return a**. Selanjutnya kita buat main method yang eksekusinya diawali dengan **public static void main(String args[])**. Di dalam nya ada pesan atau kalimat **System.out.println("hasil pemanggilan method ")**, lalu kita panggil method jumlah dengan memberi nilai ke parameter a yaitu dengan angka 5. Hal tersebut kita lakukan dengan menuliskan sebuah baris kode yaitu **System.out.println(jumlah(5))**.

## PRAKTIK 2

```
Fungsi2.java x Fungsi1.java
public class Fungsi2{
    public static int jumlah(int a){
        return (a + a);
    }
    public static void main(String args[]){
        System.out.println("Panggil method jumlah dengan parameter 5");
        System.out.println(jumlah(5));
        System.out.println("Panggil method jumlah dengan parameter 15");
        System.out.println(jumlah(15));
    }
}
```

output:

```
C:\Windows\system32\cmd.exe
Panggil method jumlah dengan parameter 5
10
Panggil method jumlah dengan parameter 15
30
Press any key to continue . . . _
```

penjelasan:

program ini lumayan mirip dengan program pada praktik 1 tadi, hanya saja di program ini sedikit berbeda di bagian saat kita mengembalikan (return) parameter a. Jadi, untuk membuat program tersebut pertama-tama kita deklarasikan terlebih dahulu nama class pada program yaitu dengan menuliskan baris kode **public class Fungsi2**. Kemudian langsung saja kita buat method beserta parameternya yaitu dengan menuliskan **public static int jumlah(int a)**. Lalu kita kembalikan parameter a untuk mengembalikan nilai a sekaligus mendapatkan hasil dari perhitungan tambah  $a+a$  yaitu dengan menuliskan baris kode **return (a + a)**.

selanjutnya kita buat main method dengan mengawali eksekusi dengan menuliskan sebuah baris kode yaitu **public static void main(String args[])**. Di dalamnya langsung kita tampilkan pesan **System.out.println("Panggil method jumlah dengan parameter 5")** yang kita ikuti dengan pemberian nilai atau value untuk parameter a dengan angka 5 yaitu dengan menuliskan baris kode **System.out.println(jumlah(5))**. Lalu kita tampilkan pesan **System.out.println("Panggil method jumlah dengan parameter 15")** yang kita ikuti dengan pemberian nilai untuk parameter a dengan angka 15 yaitu dengan menuliskan baris kode **System.out.println(jumlah(15))**. Hasilnya adalah hasil penjumlahan dari  $a+a$ . Saat kita memberikan angka 5 ke parameter a, maka hasilnya adalah 10 karena  $5+5$ . Saat kita memberikan angka 15 ke parameter a, maka hasilnya adalah 30 karena  $15+15$ .

## PRAKTIK 3

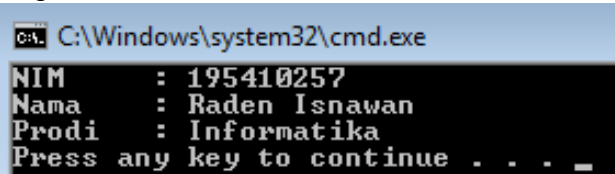
```
Mahasiswa.java x Fungsi2.java Fungsi1.java
1 public class Mahasiswa{
2     String nim;
3     String nama;
4     String prodi;
5
6     public void setMhs(String nim, String nama, String prodi)
7     {
8         this.nim = nim;
9         this.nama = nama;
10        this.prodi = prodi;
11    }
12    public void tampil(){
13        System.out.println("NIM \t: " + nim);
14        System.out.println("Nama \t: " + nama);
15        System.out.println("Prodi \t: " + prodi);
16    }
17 }
```

```

15     }
16     public static void main (String[] args){
17         Mahasiswa mhs = new Mahasiswa();
18         mhs.setMhs("195410257","Raden Isnawan","Informatika");
19         mhs.tampil();
20     }
21 }

```

output:



```

C:\Windows\system32\cmd.exe
NIM : 195410257
Nama : Raden Isnawan
Prodi : Informatika
Press any key to continue . . . _

```

penjelasan:

Untuk membuat program di atas, kita awali dengan deklarasi nama class yaitu dengan menuliskan baris kode **public class Mahasiswa** yang diikuti dengan pembuatan variabel-variabel bertipe data String yaitu **String nim**, **String nama**, **String prodi**. Dibawahnya kita buat method untuk mengakses dan inisialisasi setiap variabel supaya nanti bisa diberi value atau nilai di main method. Nama method tersebut adalah setMhs. Untuk membuat method tersebut, kita langsung saja tulis baris kode **public void setMhs(String nim, String nama, String prodi)**. Di dalamnya kita gunakan keyword **this** untuk mereferensikan setiap variabel sekaligus untuk membedakan antara variabel lokal dan variabel instance. Hal tersebut dilakukan dengan menuliskan baris kode:

```

this.nim = nim;
this.nama = nama;
this.prodi = prodi;

```

Selanjutnya kita buat method bernama tampil() untuk menampilkan hasil output di program yaitu dengan menuliskan baris kode **public void tampil()**. Pesan yang ditampilkan:

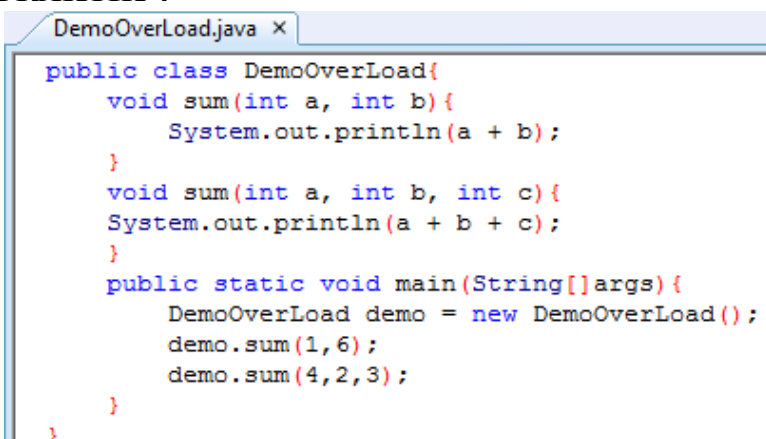
```

System.out.println("NIM \t: " + nim);
System.out.println("Nama \t: " + nama);
System.out.println("Prodi \t: " + prodi);

```

Kita buat main method yang eksekusinya diawali dengan **public static void main (String[] args)**. Di dalamnya kita lakukan instansiasi atau pembuatan objek bernama mhs dengan menuliskan baris kode **Mahasiswa mhs = new Mahasiswa()**. Setelah itu kita referensikan objek mhs terhadap method setMhs untuk memberikan nilai yaitu dengan menulis **mhs.setMhs("195410257","Raden Isnawan","Informatika")**. Data pertama adalah nim, data kedua adalah nama, dan data ketiga adalah prodi. Untuk tampil output kita panggil method tampil() dengan menuliskan **mhs.tampil()**;

## PRAKTIK 4

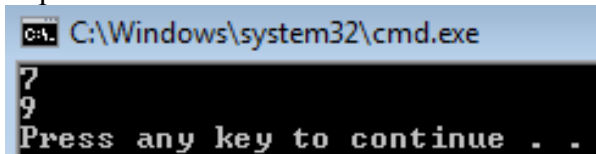


```

DemoOverLoad.java x
public class DemoOverLoad{
    void sum(int a, int b){
        System.out.println(a + b);
    }
    void sum(int a, int b, int c){
        System.out.println(a + b + c);
    }
    public static void main(String[] args){
        DemoOverLoad demo = new DemoOverLoad();
        demo.sum(1,6);
        demo.sum(4,2,3);
    }
}

```

output:



```
C:\Windows\system32\cmd.exe
7
9
Press any key to continue . .
```

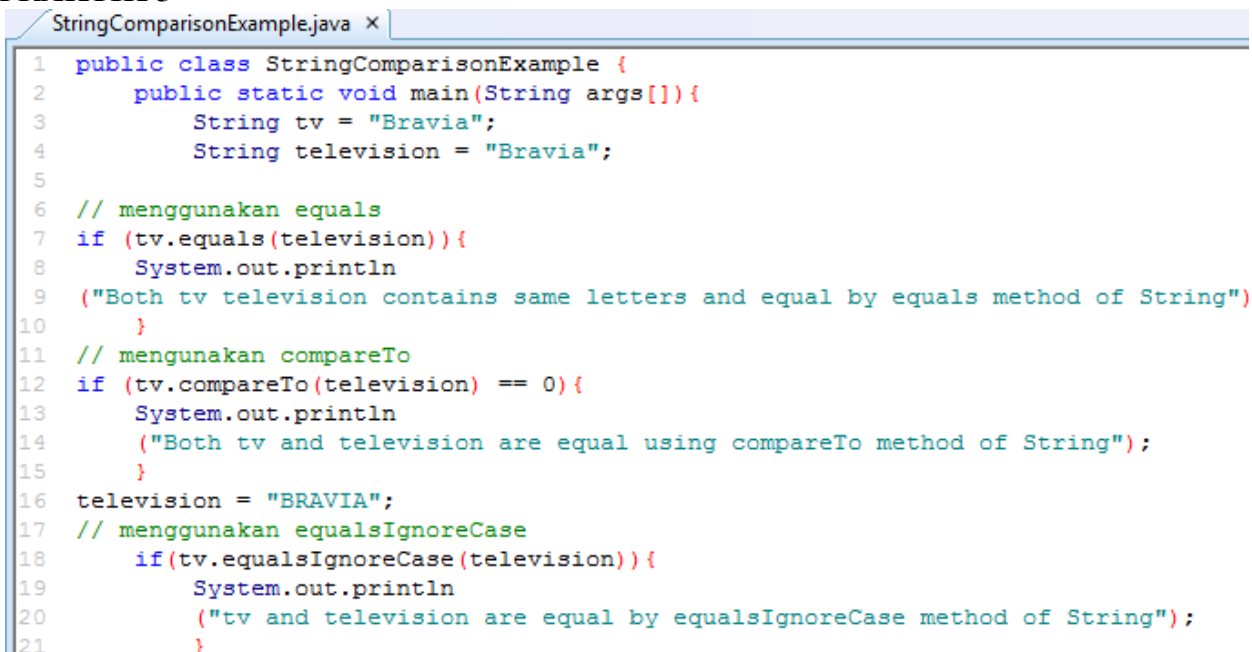
penjelasan:

method overloading adalah membedakan beberapa method dengan nama yang sama di dalam sebuah kelas namun parameternya berbeda. Hal ini sangat menguntungkan karena memudahkan kita dalam mengingat nama method. Jadi tujuan utama method overloading adalah untuk memudahkan pembacaan program dengan cepat. Untuk melakukan overloading method ada dua cara, yang pertama adalah dengan mengganti jumlah argumen/parameter, yang kedua adalah dengan mengganti tipe data.

Untuk membuat program di atas, langsung kita deklarasikan nama class dengan menuliskan baris kode **public class DemoOverLoad**. Di dalamnya kita langsung buat method bernama sum dengan parameter variabel bertipe data integer yaitu variabel a dan b. Kita buat method tersebut dengan menuliskan baris kode **void sum(int a, int b)**. Method tersebut nantinya akan menampilkan hasil penjumlahan antara nilai dari variabel a dan b yaitu **System.out.println(a + b)**. Kemudian method tersebut kita overload dengan method baru tetapi memiliki nama sama yang bisa dibuat dengan menuliskan baris kode **void sum(int a, int b, int c)**. Method tersebut merupakan method yang memiliki nama sama dengan method sebelumnya, tetapi kita tambahkan variabel integer c di dalamnya. Method tersebut nanti akan menampilkan hasil penjumlahan antara ketiga variabel tersebut yaitu **System.out.println(a + b + c)**.

Kita buat main method nya yang diawali dengan **public static void main(String[]args)**. Di dalamnya kita lakukan instansiasi atau pembuatan objek bernama demo dengan menuliskan baris kode **DemoOverLoad demo = new DemoOverLoad()**. Di dalamnya kita beri nilai atau data kepada kedua method yang telah kita buat tadi yaitu dengan menuliskan **demo.sum(1,6)** untuk method sum pertama dan **demo.sum(4,2,3)** untuk method sum kedua. Method sum pertama menghasilkan penjumlahan 1+6 yaitu 7, sementara method sum kedua menghasilkan penjumlahan 4+2+3 yaitu 9.

## PRAKTIK 5



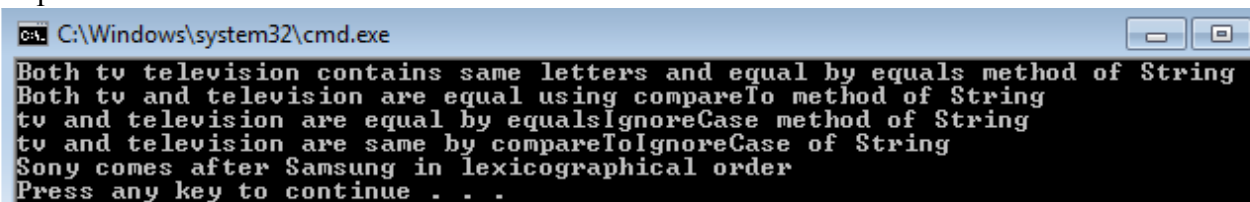
```
StringComparisonExample.java x
1 public class StringComparisonExample {
2     public static void main(String args[]){
3         String tv = "Bravia";
4         String television = "Bravia";
5
6         // menggunakan equals
7         if (tv.equals(television)){
8             System.out.println
9             ("Both tv television contains same letters and equal by equals method of String")
10        }
11        // menggunakan compareTo
12        if (tv.compareTo(television) == 0){
13            System.out.println
14            ("Both tv and television are equal using compareTo method of String");
15        }
16        television = "BRAVIA";
17        // menggunakan equalsIgnoreCase
18        if (tv.equalsIgnoreCase(television)){
19            System.out.println
20            ("tv and television are equal by equalsIgnoreCase method of String");
21        }
22    }
23 }
```

```

22         // menggunakan compareToIgnoreCase
23         if (tv.compareToIgnoreCase(television) == 0){
24             System.out.println
25                 ("tv and television are same by compareToIgnoreCase of String");
26         }
27         String sony = "Sony";
28         String samsung = "Samsung";
29         // menggunakan compareTo
30         if (sony.compareTo(samsung) > 0){
31             System.out.println
32                 ("Sony comes after Samsung in lexicographical order");
33         } else if (sony.compareTo(samsung) < 0){
34             System.out.println
35                 ("Sony comes before Samsung in lexicographical order");
36         }
37     }
38 }

```

output:



```

C:\Windows\system32\cmd.exe
Both tv television contains same letters and equal by equals method of String
Both tv and television are equal using compareTo method of String
tv and television are equal by equalsIgnoreCase method of String
tv and television are same by compareToIgnoreCase of String
Sony comes after Samsung in lexicographical order
Press any key to continue . . .

```

penjelasan:

program di atas adalah contoh program yang menggunakan berbagai method bawaan Java untuk membandingkan dua buah string. Untuk membuat program di atas, kita awali dengan deklarasi nama class yaitu dengan menuliskan baris kode **public class StringComparisonExample**. Class tersebut bermodifikasi public yaitu keyword yang berfungsi untuk menentukan sifat akses ke semua member yang mengikutinya (di bawahnya), sehingga memiliki sifat dapat di akses dari manapun, dan dapat di akses dari dalam class itu sendiri, dari anak class (derived class) dan juga dari luar class.

Lalu dilanjutkan dengan **public static void main(String args[])**. Public, dapat diartikan bahwa argumen atau metode main() merupakan sebuah metode yang bersifat publik. void merupakan suatu tipe data yang menyatakan bahwa deklarasi code tidak memerlukan nilai balik atau return. String menyatakan bahwa argumen yang terdapat pada code java tersebut merupakan String. Args merupakan argumen bertipe data string yang mengandung array, Args dapat di misalkan sebagai memory untuk menampung program.

Kita buat dua buah variabel yaitu tv dan television. Keduanya memiliki value string yang sama yaitu "Bravia". Kita tulis kodenya dengan **String tv = "Bravia"** dan **String television = "Bravia"**. Kemudian kita pakai method equals. Metode equals() membandingkan dua String berdasarkan kontennya. Sehingga apabila dua buah String memiliki huruf-huruf yang sama, diurutkan yang sama dan besar-kecil yang sama maka dua String ini akan dianggap sama. Kita langsung saja tulis **if (tv.equals(television))** yang artinya "jika variabel tv equals television, maka...". Kemudian dilanjutkan dengan menampilkan pesan ("**Both tv television contains same letters and equal by equals method of String**"). Karena bernilai TRUE (karena dua buah string memiliki huruf, urutan, dan besae-kecil yang sama), sehingga output berupa tersebut berhasil ditampilkan oleh program.

Method kedua yang kita gunakan adalah compareTo. compareTo membandingkan dua buah String berdasarkan urutan leksikografisnya (posisi kamus). Method ini akan mengembalikan sebuah nilai int. Jika nilai yang dikembalikan kurang dari nol, maka String yang pertama memiliki posisi kamus lebih dulu. Jika nilai yang dikembalikan sama dengan nol maka kedua String memiliki posisi kamus



yang sama. Jika nilai yang dikembalikan lebih dari nol maka String yang kedua (yang dikirimkan ke parameter compareTo) memiliki posisi kamus lebih dulu. Karena nilai yang dikembalikan sama dengan nol yaitu **if (tv.compareTo(television) == 0)**, maka output bernilai TRUE dan akan menampilkan pesan kalimat ("**Both tv and television are equal using compareTo method of String**").

Method ketiga yang kita buat dan gunakan adalah equalsIgnoreCase. method equalsIgnoreCase membandingkan dua String dengan mengabaikan besar-kecilnya. Maka dari itu kita ubah String milik variabel television dari "Bravia" menjadi "BRAVIA". Setelah itu, langsung kita tulis **if(tv.equalsIgnoreCase(television))**. Karena equalsIgnoreCase mengabaikan besar-kecil huruf, maka tetap bernilai "BRAVIA" dan "Bravia" tetap dianggap sama dan tetap bernilai TRUE, sehingga program menampilkan pesan ("**tv and television are equal by equalsIgnoreCase method of String**").

Method keempat yang kita gunakan adalah compareToIgnoreCase. Metode ini mengabaikan besar-kecil String seperti halnya equalsIgnoreCase() dan mengembalikan nilai yang sama dengan metode compareTo(). Kita buat baris kode **if (tv.compareToIgnoreCase(television) == 0)**, dan program menghasilkan TRUE sehingga menampilkan pesan kalimat ("**tv and television are same by compareToIgnoreCase of String**")

kemudian kita coba lagi menggunakan compareTo. Kita buat variabel sony yang kita beri nilai "Sony" dan variabel samsung yang kita beri nilai "Samsung". Untuk itu kita tulis **String sony = "Sony"** dan **String samsung = "Samsung"**. Jika kondisi seleksi yang terpenuhi adalah **if (sony.compareTo(samsung) > 0)**, maka muncul pesan ("**Sony comes after Samsung in lexicographical order**"). Jika kondisi yang terpenuhi adalah **else if (sony.compareTo(samsung) < 0)**, maka muncul pesan ("**Sony comes before Samsung in lexicographical order**"). Karena nilai yang dikembalikan lebih dari nol maka String yang kedua (yang dikirimkan ke parameter compareTo) memiliki posisi kamus lebih dulu sehingga Samsung menempati urutan pertama dan Sony menempati urutan kedua sehingga pesan yang muncul adalah "Sony comes after Samsung in lexicographical order".

---

## LATIHAN 1

```
MahasiswaEdit.java x
public class MahasiswaEdit{
    String nim;
    String nama;
    String prodi;
    double ipk;

    public void setMhs(String nim, String nama, String prodi, double ipk){
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
        this.ipk = ipk;
    }
    public void tampil(){
        System.out.println("NIM \t: " + nim);
        System.out.println("Nama \t: " + nama);
        System.out.println("Prodi \t: " + prodi);
        System.out.println("IPK \t: " + ipk);
    }
    public static void main (String[] args){
        MahasiswaEdit mhs = new MahasiswaEdit();
        mhs.setMhs("195410257", "Raden Isnawan", "Informatika", 4.0);
        mhs.tampil();
    }
}
```

output:

```
C:\Windows\system32\cmd.exe
NIM      : 195410257
Nama     : Raden Isnawan
Prodi    : Informatika
IPK      : 4.0
Press any key to continue . . .
```

penjelasan:

program ini sebenarnya sama saja dengan program di praktik 3, hanya saja di program ini kita tambahkan variabel ipk. Untuk membuat program ini, kita deklarasikan nama class yaitu dengan menuliskan baris kode **public class MahasiswaEdit** yang diikuti dengan pembuatan variabel-variabel bertipe data String dan double yaitu **String nim, String nama, String prodi, double ipk**. Dibawahnya kita buat method untuk mengakses dan referensi setiap variabel supaya nanti bisa diberi value atau nilai di main method. Nama method tersebut adalah setMhs. **Untuk membuat method tersebut, kita langsung saja tulis baris kode public void setMhs(String nim, String nama, String prodi)**. Di dalamnya kita gunakan keyword this untuk mereferensikan setiap variabel sekaligus untuk membedakan antara variabel lokal dan variabel instance. Hal tersebut dilakukan dengan menuliskan baris kode:

```
this.nim = nim;
this.nama = nama;
this.prodi = prodi;
this.ipk = ipk;
```

Selanjutnya kita buat method bernama tampil() untuk menampilkan hasil output di program yaitu dengan menuliskan baris kode **public void tampil()**. Pesan yang ditampilkan:

```
System.out.println("NIM \t: " + nim);
System.out.println("Nama \t: " + nama);
System.out.println("Prodi \t: " + prodi);
System.out.println("IPK \t: " + ipk);
```

Kita buat main method yang eksekusinya diawali dengan **public static void main (String[]args)**. Di dalamnya kita lakukan instansiasi atau pembuatan objek bernama mhs dengan menuliskan baris kode Mahasiswa mhs = new Mahasiswa(). Setelah itu kita referensikan objek mhs terhadap method setMhs untuk memberikan nilai yaitu dengan menulis mhs.setMhs("195410257","Raden Isnawan","Informatika",4.0). Data pertama adalah nim, data kedua adalah nama, data ketiga adalah prodi, dan data keempat adalah ipk. Untuk menampilkan output kita panggil method tampil() dengan menuliskan **mhs.tampil();**.

## LATIHAN 2

```
MainMahasiswa.java x
1  class Mahasiswa{
2      public int sks;
3      public double ipk;
4      public String data;
5
6      // method mutator
7      public void setSks(int sks) {
8          this.sks=sks;
9      }
10     public void setIpk(double ipk) {
11         this.ipk=ipk;
12     }
13     public void setData(String data) {
14         this.data=data;
15     }
```

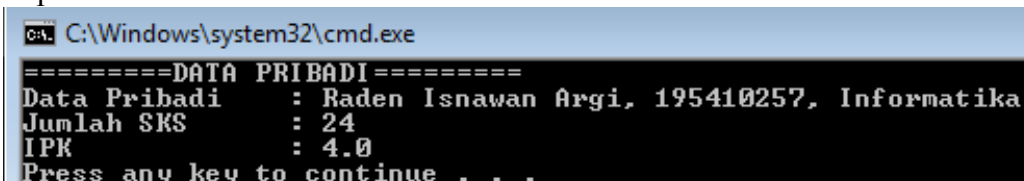


```

16 //method accessor
17     public int getSks() {
18         return sks;
19     }
20     public double getIpk() {
21         return ipk;
22     }
23     public String getData() {
24         return data;
25     }
26 }
27
28 public class MainMahasiswa{
29     public static void main(String args[]){
30
31         Mahasiswa mahasiswa=new Mahasiswa();
32         mahasiswa.setSks(24);
33         mahasiswa.setIpk(4.0);
34         mahasiswa.setData("Raden Isnawan Argi, 195410257, Informatika");
35
36         System.out.println("=====DATA PRIBADI=====");
37         System.out.println("Data Pribadi\t: "+ mahasiswa.getData());
38         System.out.println("Jumlah SKS\t: "+ mahasiswa.getSks());
39         System.out.println("IPK\t\t\t: "+ mahasiswa.getIpk());
40     }
41 }

```

output:



```

C:\Windows\system32\cmd.exe
=====DATA PRIBADI=====
Data Pribadi      : Raden Isnawan Argi, 195410257, Informatika
Jumlah SKS        : 24
IPK               : 4.0
Press any key to continue . . .

```

penjelasan:

Di program ini, kita menerapkan method mutator dan accessor dan pemrograman berorientasi objek beserta konsep inheritance (pewarisan). Inheritance atau Pewarisan/Penurunan adalah konsep pemrograman dimana sebuah class dapat 'menurunkan' property dan method yang dimilikinya kepada class lain. Konsep inheritance digunakan untuk memanfaatkan fitur 'code reuse' untuk menghindari duplikasi kode program.

Method mutator adalah sebuah metode yang digunakan untuk menetapkan atau mengubah nilai dari sebuah private field, skema ini biasanya ditandai dengan awalan nama "set". Sementara method accessor adalah sebuah metode yang digunakan untuk mengembalikan nilai dari sebuah private field, skema ini biasanya ditandai dengan awalan nama "get". Kita menggunakan 2 method tersebut untuk menyembunyikan data dari objek. Buffer yang disediakan oleh method ini memungkinkan kita untuk mengubah data dari balik layar dan menerapkan atau mengaplikasikan validasi nilai-nilai yang telah kita atur di dalam field.

Untuk membuat program di atas, kita buat parent class yaitu **class Mahasiswa** lalu kita buat tiga variabel yang masing-masing bertipe data integer, double, dan String. Kita tulis variabel-variabel tersebut dengan **public int sks**, **public double ipk**, dan **public String data**.

Kemudian di baris 6-15 kita buat method mutator untuk menetapkan nilai suatu variabel, dan kita juga menggunakan keyword "this" untuk mengakses variabel target. Contoh penerapannya dalam program tersebut adalah:

```

public void setSks(int sks) {
    this.sks=sks;
}

```

di baris 16-25 kita buat method accessor untuk mengembalikan nilai field/variabel dengan menggunakan keyword "return". Method ini mengembalikan value/nilai dengan tipe data yang sama sesuai korespondensi private fieldnya, misal (String);. Contoh penerapannya dalam program tersebut adalah:

```
public int getSks() {  
    return sks;  
}
```

kemudian kita buat child class dengan menuliskan **public class MainMahasiswa**. Isi dari child class tersebut adalah main method yang eksekusinya diawali dengan **public static void main(String args[])**. Public, dapat diartikan bahwa argumen atau metode main() merupakan sebuah metode yang bersifat publik. void merupakan suatu tipe data yang menyatakan bahwa deklarasi code tidak memerlukan nilai balik atau return. String menyatakan bahwa argumen yang terdapat pada code java tersebut merupakan String. Args merupakan argumen bertipe data string yang mengandung array, Args dapat di misalkan sebagai memory untuk menampung program.

Selanjutnya kita lakukan instansiasi atau pembuatan objek bernama mahasiswa dengan menuliskan baris kode **Mahasiswa mahasiswa=new Mahasiswa()**. Di dalamnya kita beri nilai kepada setiap variabel yaitu dengan menuliskan:

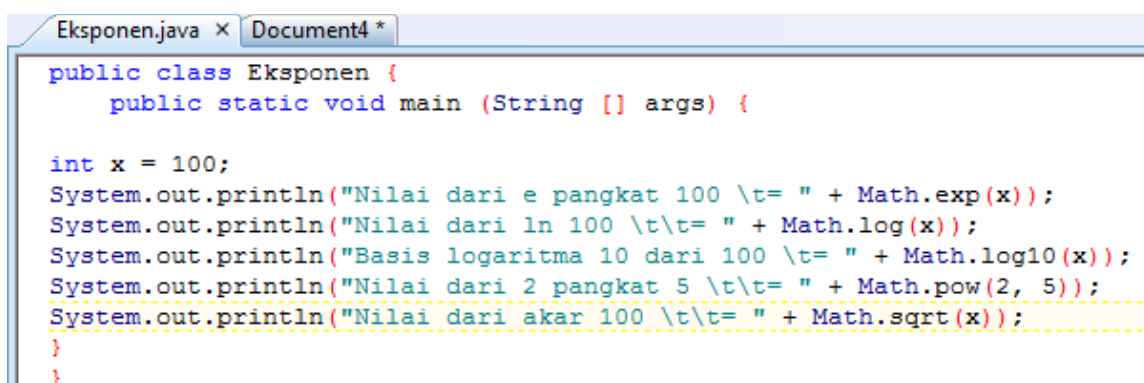
```
mahasiswa.setSks(24);  
mahasiswa.setIpk(4.0);  
mahasiswa.setData("Raden Isnawan Argi, 195410257, Informatika");
```

Setelah kita berikan nilai, kita tampilkan data-data tersebut dengan get yaitu dengan menuliskan:

```
System.out.println("=====DATA PRIBADI=====");  
System.out.println("Data Pribadi\t: "+ mahasiswa.getData());  
System.out.println("Jumlah SKS\t: "+ mahasiswa.getSks());  
System.out.println("IPK\t\t: "+ mahasiswa.getIpk());
```

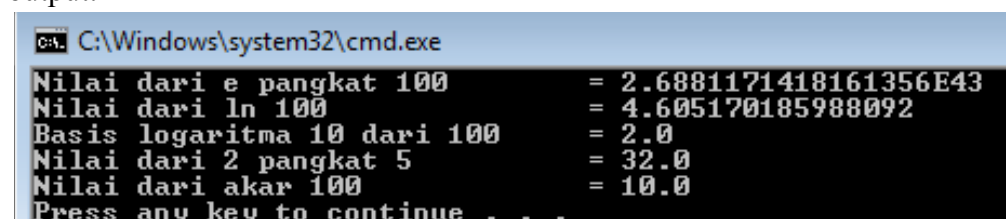
---

## TUGAS 1



```
Eksponen.java x Document4 *  
public class Eksponen {  
    public static void main (String [] args) {  
  
        int x = 100;  
        System.out.println("Nilai dari e pangkat 100 \t= " + Math.exp(x));  
        System.out.println("Nilai dari ln 100 \t\t= " + Math.log(x));  
        System.out.println("Basis logaritma 10 dari 100 \t= " + Math.log10(x));  
        System.out.println("Nilai dari 2 pangkat 5 \t\t= " + Math.pow(2, 5));  
        System.out.println("Nilai dari akar 100 \t\t= " + Math.sqrt(x));  
    }  
}
```

output:



```
C:\Windows\system32\cmd.exe  
Nilai dari e pangkat 100          = 2.6881171418161356E43  
Nilai dari ln 100                 = 4.605170185988092  
Basis logaritma 10 dari 100      = 2.0  
Nilai dari 2 pangkat 5           = 32.0  
Nilai dari akar 100              = 10.0  
Press any key to continue . . .
```

penjelasan:

apa itu eksponen? Bilangan Eksponen adalah bentuk dari sebuah bilangan yang dikalikan dengan bilangan yang sama dan di ulang-ulang, atau lebih mudahnya kita bisa menyebutnya sebagai perkalian yang diulang-ulang. Eksponen juga bisa dikenal sebagai pangkat yang akan menunjukkan nilai derajat ke pangkatan. Untuk membuat program di atas, kita beri nama class dengan **public class Eksponen** lalu kita buat variabel x yang kita beri nilai 100.

Class Math menyediakan lima method yang berhubungan dengan eksponen yaitu:

exp(x), mengembalikan nilai e pangkat x.

log(x), mengembalikan nilai logaritma natural dari x..

log10(x), mengembalikan nilai dari basis logaritma 10 dari x..

pow(a, b), mengembalikan nilai dari a pangkat b

sqrt(x), mengembalikan nilai dari akar x, untuk  $x \geq 0$

## TUGAS 2

### Fungsi5a

```
Fungsi5a.java x
1  public class Fungsi5a{
2      public float jumlah(int a, float b){
3          System.out.println(" a = "+a+" dan b = "+b); // 1.
4          a = 12;
5          b = 14.5f;
6          System.out.println(" a = "+a+" dan b = "+b); // 2.
7          return (a + b);
8      }
9      public static void main(String args[]){
10         Fungsi5a obyek = new Fungsi5a();
11         int a; float b;
12         a = 17;
13         b = 12.3f;
14
15         System.out.println("Panggil method jumlah dengan parameter a dan b")
16         System.out.println(obyek.jumlah(a, b)); // 3.
17         System.out.println(" a = "+a+" dan b = "+b); // 4.
18     }
19 }
```

output:

```
C:\Windows\system32\cmd.exe
Panggil method jumlah dengan parameter a dan b
a = 17 dan b = 12.3
a = 12 dan b = 14.5
26.5
a = 17 dan b = 12.3
Press any key to continue . . .
```

penjelasan:

kita awali program dengan mendeklarasikan nama class yaitu dengan menuliskan baris kode **public class Fungsi5a**. Kemudian langsung kita buat method bertipe data integer dan float bernama jumlah dengan menuliskan baris kode **public float jumlah(int a, float b)**. Di dalamnya kita buat perintah untuk menampilkan sebuah kalimat yaitu **System.out.println(" a = "+a+" dan b = "+b);**. Pesan tersebut memiliki variabel a dan b yang belum di deklarasi sehingga harus dibuat di main method nanti. Setelah itu kita buat variabel a dan b yaitu **a = 12** dan **b = 14.5f** untuk pesan selanjutnya yang akan ditampilkan yaitu **System.out.println(" a = "+a+" dan b = "+b);** yang akan me-return berupa penjumlahan yaitu **return (a + b)**. Hal ini nantinya akan menampilkan hasil penjumlahan 12+14,5. Selanjutnya kita buat main method diawali dengan **public static void main**

(**String args[]**). Di main method tersebut kita lakukan instansiasi atau pembuatan objek bernama obyek yang dituliskan dengan baris kode **Fungsi5a obyek = new Fungsi5a()**. Di dalamnya kita beri nilai ke variabel a dan variabel b untuk pesan pertama yang belum memiliki nilai di baris ke-3 tadi. Kita buat kedua variabel itu dengan menuliskan **a = 17** dan **b = 12.3**. Kemudian kita beri kalimat pembuka pada program yaitu **System.out.println("Panggil method jumlah dengan parameter a dan b")**.

di main method kita akan memanggil method jumlah dengan **System.out.println(obyek.jumlah(a, b))**. Hal ini akan menghasilkan output:

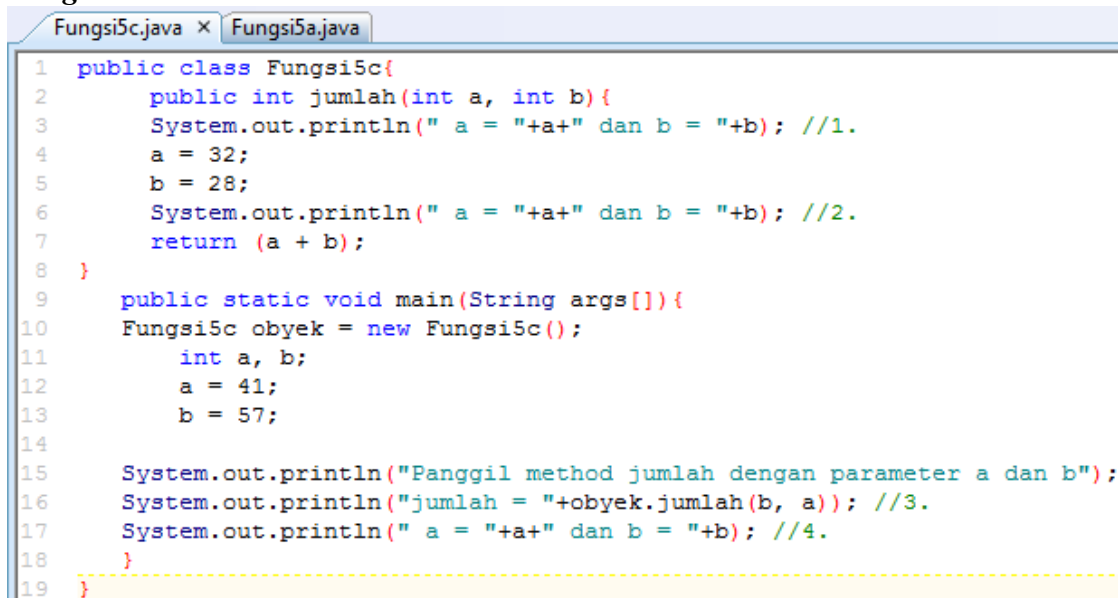
**a = 17 dan b = 12.3**

**a = 12 dan b = 14.5**

**26** (hasil penjumlahan a+b atau 12+14.5)

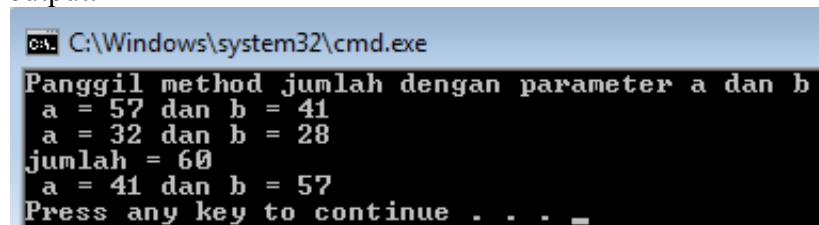
terakhir kita tambahkan **System.out.println(" a = "+a+" dan b = "+b)** yang menghasilkan output: **a = 17 dan b = 12.3** (outputnya bukan a = 12 dan b = 14.5) karena perintah ini mengikuti variabel beserta data yang ada di dalam main method.

### Fungsi5c



```
1 public class Fungsi5c{
2     public int jumlah(int a, int b){
3         System.out.println(" a = "+a+" dan b = "+b); //1.
4         a = 32;
5         b = 28;
6         System.out.println(" a = "+a+" dan b = "+b); //2.
7         return (a + b);
8     }
9     public static void main(String args[]){
10        Fungsi5c obyek = new Fungsi5c();
11        int a, b;
12        a = 41;
13        b = 57;
14
15        System.out.println("Panggil method jumlah dengan parameter a dan b");
16        System.out.println("jumlah = "+obyek.jumlah(b, a)); //3.
17        System.out.println(" a = "+a+" dan b = "+b); //4.
18    }
19 }
```

output:



```
C:\Windows\system32\cmd.exe
Panggil method jumlah dengan parameter a dan b
a = 57 dan b = 41
a = 32 dan b = 28
jumlah = 60
a = 41 dan b = 57
Press any key to continue . . . _
```

penjelasan:

program ini hampir sama dengan program di tugas 1 walaupun ada sedikit perubahan. Pertama-tama kita mendeklarasikan nama class yaitu dengan menuliskan baris kode **public class Fungsi5c**. Kemudian langsung kita buat method bertipe data integer bernama jumlah dengan menuliskan baris kode **public int jumlah(int a, int b)**. Di dalamnya kita buat perintah untuk menampilkan sebuah kalimat yaitu **System.out.println(" a = "+a+" dan b = "+b)**. Pesan tersebut memiliki variabel a dan b yang belum di deklarasi sehingga harus dibuat di main method nanti. Setelah itu kita buat variabel a dan b yaitu **a = 32** dan **b = 28** untuk pesan selanjutnya yang akan ditampilkan yaitu

**System.out.println(" a = "+a+" dan b = "+b);** yang akan me-return berupa penjumlahan yaitu **return (a + b)**. Baris kode tersebut nantinya akan menampilkan hasil penjumlahan 32+28.

kita buat main method diawali dengan **public static void main (String args[])**. Public, dapat diartikan bahwa argumen atau metode main() merupakan sebuah metode yang bersifat publik. void merupakan suatu tipe data yang menyatakan bahwa deklarasi code tidak memerlukan nilai balik atau return. String menyatakan bahwa argumen yang terdapat pada code java tersebut merupakan String. Args merupakan argumen bertipe data string yang mengandung array, Args dapat di misalkan sebagai memory untuk menampung program.

Di main method tersebut kita lakukan instansiasi atau pembuatan objek bernama obyek yang dituliskan dengan baris kode **Fungsi5c obyek = new Fungsi5c()**. Di dalamnya kita beri nilai ke variabel a dan variabel b untuk pesan pertama yang belum memiliki nilai di baris ke-3 tadi. Kita buat kedua variabel itu dengan menuliskan **a = 41** dan **b = 57**. Kemudian kita beri kalimat pembuka pada program yaitu **System.out.println("Panggil method jumlah dengan parameter a dan b")**.

di main method kita akan memanggil method jumlah dengan **System.out.println(obyek.jumlah(b, a))**. Hal ini akan menghasilkan output:

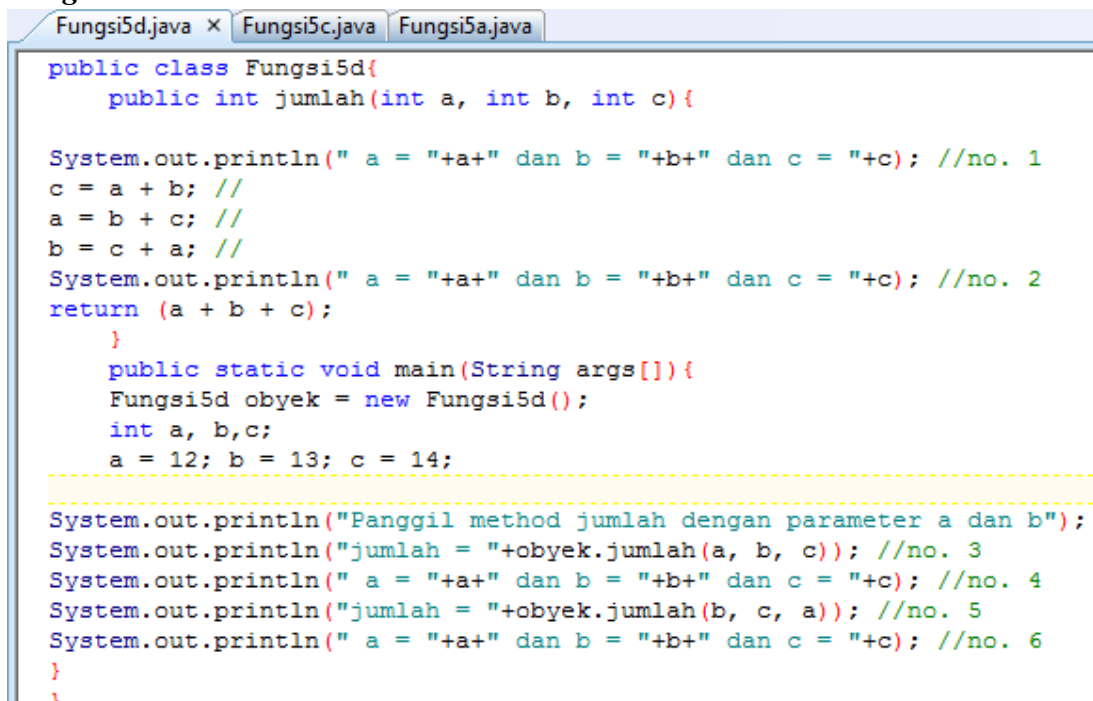
**a = 57 dan b = 41** (karena b,a bukan a,b(dibalik))

**a = 32 dan b = 28**

**jumlah 60** (hasil penjumlahan a+b atau 32+28)

terakhir kita tambahkan **System.out.println(" a = "+a+" dan b = "+b)** yang menghasilkan output:  
**a = 41 dan b = 57**

### *Fungsi5d*



```
public class Fungsi5d{
    public int jumlah(int a, int b, int c){

        System.out.println(" a = "+a+" dan b = "+b+" dan c = "+c); //no. 1
        c = a + b; //
        a = b + c; //
        b = c + a; //
        System.out.println(" a = "+a+" dan b = "+b+" dan c = "+c); //no. 2
        return (a + b + c);
    }
    public static void main(String args[]){
        Fungsi5d obyek = new Fungsi5d();
        int a, b,c;
        a = 12; b = 13; c = 14;

        System.out.println("Panggil method jumlah dengan parameter a dan b");
        System.out.println("jumlah = "+obyek.jumlah(a, b, c)); //no. 3
        System.out.println(" a = "+a+" dan b = "+b+" dan c = "+c); //no. 4
        System.out.println("jumlah = "+obyek.jumlah(b, c, a)); //no. 5
        System.out.println(" a = "+a+" dan b = "+b+" dan c = "+c); //no. 6
    }
}
```

output di halaman selanjutnya

output:

```
C:\Windows\system32\cmd.exe
Panggil method jumlah dengan parameter a dan b
a = 12 dan b = 13 dan c = 14
a = 38 dan b = 63 dan c = 25
jumlah = 126
a = 12 dan b = 13 dan c = 14
a = 13 dan b = 14 dan c = 12
a = 41 dan b = 68 dan c = 27
jumlah = 136
a = 12 dan b = 13 dan c = 14
Press any key to continue . . . _
```

penjelasan:

program kita awali dengan mendeklarasikan nama yaitu **public class Fungsi5d**. Kemudian langsung kita buat method **public int jumlah(int a, int b, int c)**. Di dalamnya kita buat pesan yang variabel a dan b nya belum diberi nilai sehingga harus diberi nilai di main method nanti yaitu **System.out.println(" a = "+a+" dan b = "+b+" dan c = "+c);**. Kemudian buat variabel dan kita beri perhitungan penjumlahan di dalamnya yaitu:

**c = a + b;**

**a = b + c;**

**b = c + a;**

lalu kita cetak lagi **System.out.println(" a = "+a+" dan b = "+b+" dan c = "+c);** yang akan menampilkan dan me-return hasil penjumlahan berupa **return (a + b + c)**.

lalu kita buat main method yang diawali dengan **public static void main(String args[])**. Public, dapat diartikan bahwa argumen atau metode main() merupakan sebuah metode yang bersifat publik. void merupakan suatu tipe data yang menyatakan bahwa deklarasi code tidak memerlukan nilai balik atau return. String menyatakan bahwa argumen yang terdapat pada code java tersebut merupakan String. Args merupakan argumen bertipe data string yang mengandung array, Args dapat di misalkan sebagai memory untuk menampung program.

Di dalam main method itu kita buat variabel **int a, b, c** dan masing-masing variabel kita beri nilai dengan menuliskan baris kode **a = 12; b = 13; c = 14**. Kemudian kita tampilkan kalimat **System.out.println("Panggil method jumlah dengan parameter a dan b")** untuk membuka program.

**System.out.println("jumlah = "+obyek.jumlah(a, b, c));** akan menghasilkan output:

**a = 12 dan b = 13 dan c = 14**

**a = 38 dan b = 63 dan c = 25**

**jumlah 126** (hasil penjumlahan a+b+c atau 38+63+25)

**System.out.println(" a = "+a+" dan b = "+b+" dan c = "+c);** akan menghasilkan output:

**a = 12 dan b = 13 dan c = 14**

**System.out.println("jumlah = "+obyek.jumlah(b, c, a));** akan menghasilkan output:

**a = 13 dan b = 14 dan c = 12**

**a = 41 dan b = 68 dan c = 27**

**jumlah 136** (hasil penjumlahan a+b+c atau 41+68+27)

**System.out.println(" a = "+a+" dan b = "+b+" dan c = "+c);** akan menghasilkan output:

**a = 12 dan b = 13 dan c = 14**

---

kesimpulan di halaman selanjutnya



## **KESIMPULAN**

Di pertemuan ke-9 ini, saya berhasil memenuhi tujuan praktikum modul ini yaitu dapat menyelesaikan kasus dengan menggunakan method dengan parameter, membuat method overloading dan menggunakan method-method bawaan yang ada di java. Selain menerapkan ketiga konsep diatas, ternyata di pertemuan ke-9 ini saya juga mendapatkan cukup banyak wawasan tentang OOP (Object Oriented Programming) atau jika di bahasa indonesiakan akan menjadi PBO (Pemrograman Berorientasi Objek). OOP beserta konsep-konsepnya banyak kita terapkan terutama di bagian Latihan dan Tugas. Selain itu, saya mendapat banyak wawasan tentang method built-in atau bawaan yang ada di bahasa pemrograman Java beserta kegunaannya.

**Terima Kasih**