

# **LAPORAN PRAKTIKUM**

## **PEMROGRAMAN BERORIENTASI OBJEK**

### **PERTEMUAN KE-4**



**Disusun Oleh :**

**NAMA : Raden Isnawan Argi Aryasatya**

**NIM : 195410257**

**JURUSAN : Teknik Informatika**

**JENJANG : S1**

**KELAS : TI-5**

**Laboratorium Terpadu**  
**Sekolah Tinggi Manajemen Informatika Komputer**  
**AKAKOM**  
**YOGYAKARTA**

**2021**

## ***PERTEMUAN KE-4*** **(KONSTRUKTOR)**

### **TUJUAN**

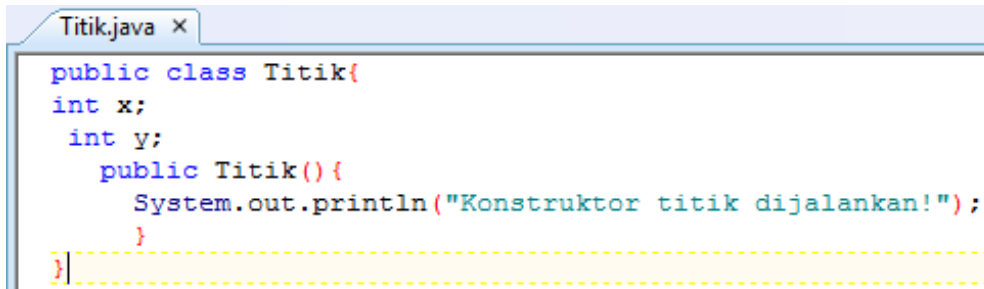
Dapat membuat dan menggunakan konstruktor

### **DASAR TEORI**

Konstruktor adalah sebuah tipe khusus dari method yang digunakan untuk membuat dan menginisialisasi sebuah object baru. Berikut ini adalah property dari konstruktor :

1. konstruktor memiliki nama yang sama dengan class
  2. konstruktor tidak memiliki return value
  3. konstruktor tidak dapat dipanggil secara langsung, namun harus dipanggil dengan menggunakan operator new pada pembentukan sebuah class.
- 

### **PRAKTIK 1: Membuat Konstruktor**



```
Titik.java x
public class Titik{
    int x;
    int y;
    public Titik(){
        System.out.println("Konstruktor titik dijalankan!");
    }
}
```

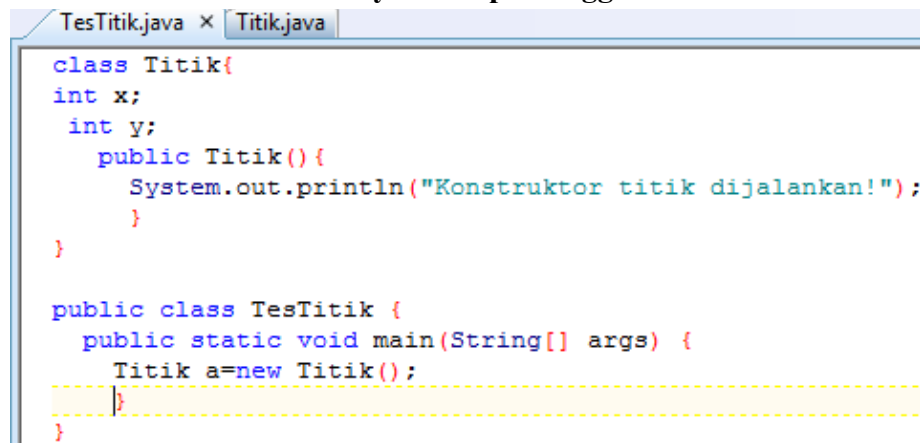
penjelasan:

Pada praktik 1 ini, kita diinstruksikan untuk membuat atau mendeklarasikan konstruktor. Lalu pada gambar di atas, manakah yang disebut dengan konstruktor? Tentu saja baris kode berikut:

```
public Titik(){
    System.out.println("Konstruktor titik dijalankan!");
}
```

Cara membuat constructor adalah dengan menuliskan nama method constructor sama seperti nama class. Nama class di atas adalah “Titik” maka dari itu nama konstruktor juga harus “Titik”. Pastikan memberikan modifier public kepada Constructor, karena ia akan dieksekusi saat pembuatan objek (instance).

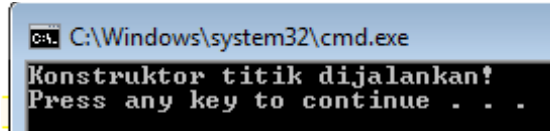
### **PRAKTIK 2: Membuat obyek dan pemanggilan konstruktor**



```
TesTitik.java x Titik.java
class Titik{
    int x;
    int y;
    public Titik(){
        System.out.println("Konstruktor titik dijalankan!");
    }
}

public class TesTitik {
    public static void main(String[] args) {
        Titik a=new Titik();
    }
}
```

output:



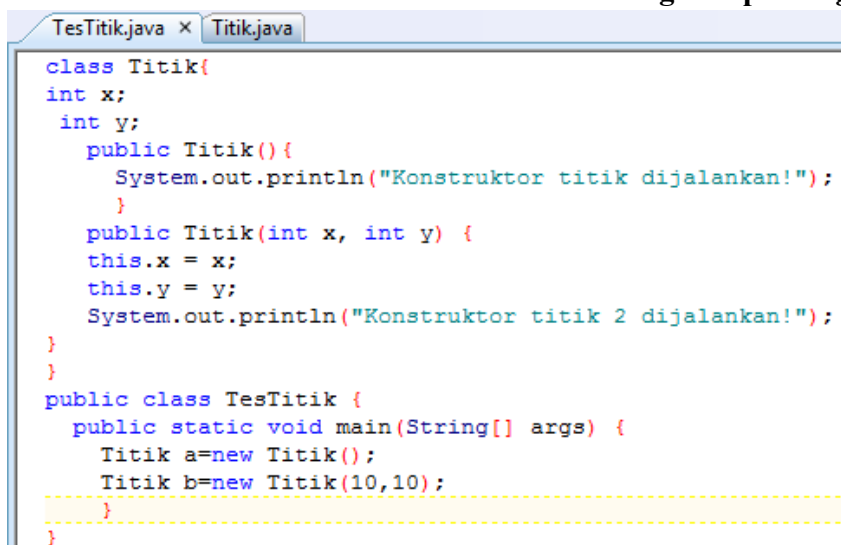
```
C:\Windows\system32\cmd.exe
Konstruktor titik dijalankan!
Press any key to continue . . .
```

penjelasan:

Dalam script program di atas, public class TesTitik berfungsi untuk mendeklarasikan class bersifat public bernama TesTitik yang gunanya untuk mengawali penginisialisasian objek sekaligus memanggil konstruktor. Lalu deklarasikan **public static void main(String args[]){** untuk mengawali eksekusi . public, yang berarti metode ini bisa dipanggil dan digunakan didalam Class atau diluar Class. public, yang berarti metode ini bisa dipanggil dan digunakan didalam Class atau diluar Class. void, yang berarti bahwa metode ini tidak mengirimkan nilai balik. String[], adalah tipe data objek yang menangani serangkaian karakter-karakter yang berjenis array. args, adalah variabel objek.

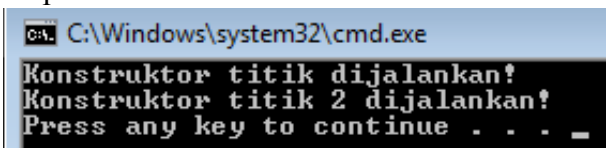
Selanjutnya, kita buat objek dari class Titik dengan **Titik a=new Titik();**. Maka otomatis isi dari konstruktor lah yang akan muncul yaitu **"Konstruktor titik dijalankan!"**. Hal itu terjadi karena nama class dan nama konstruktor sama sehingga saat pembuatan objek dari class Titik maka value dari konstruktor Titik lah yang muncul.

### PRAKTIK 3: Membuat konstruktor overloading dan pemanggilan konstruktor



```
TesTitik.java x Titik.java
class Titik{
    int x;
    int y;
    public Titik(){
        System.out.println("Konstruktor titik dijalankan!");
    }
    public Titik(int x, int y) {
        this.x = x;
        this.y = y;
        System.out.println("Konstruktor titik 2 dijalankan!");
    }
}
public class TesTitik {
    public static void main(String[] args) {
        Titik a=new Titik();
        Titik b=new Titik(10,10);
    }
}
```

output:



```
C:\Windows\system32\cmd.exe
Konstruktor titik dijalankan!
Konstruktor titik 2 dijalankan!
Press any key to continue . . .
```

penjelasan:

Pertama-tama saya jelaskan dulu apa itu konstruktor overloading. Jadi, konstruktor overloading adalah mekanisme dimana kita dapat membuat Konstruktor lebih dari satu dalam satu Class, tapi dengan ketentuan setiap Konstruktor memiliki Parameter yang berbeda, bisa berbeda jumlah parameternya ataupun berbeda tipe data parameternya. Lalu, di script program di atas mana yang merupakan konstruktor overloading? Tentu saja baris kode berikut:

```

public Titik(int x, int y) {
    this.x = x;
    this.y = y;
    System.out.println("Konstruktor titik 2 dijalankan!");
}

```

Konstruktor overloading tersebut kita tempatkan tepat dibawah konstruktor pertama tadi. Pertama, kita buat konstruktor overloading dengan mendeklarasikan **public Titik(int x, int y)**. Lalu untuk mengakses komponen dalam konstruktor tersebut, gunakan “this” yaitu `this.x=x`; dan `this.y=y`;. Selanjutnya cetak output dengan **System.out.println("Konstruktor titik 2 dijalankan!");**.

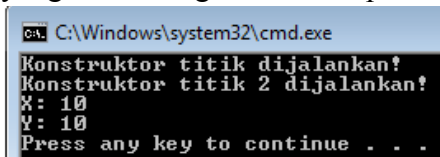
Setelah ini langsung saja kita buat objek baru bernama “b” dengan **Titik b=new Titik(10,10);**. Nah yang jadi pertanyaan disini adalah, kenapa nilai dari x dan y yaitu 10 dan 10 tidak muncul saat kita memunculkan output program? Tentu saja karena kita tidak mendeklarasikan dan mencetak output objek b dan menginisiasikannya dengan x dan y. Untuk melakukan itu, maka masukkan baris kode seperti ini di dalam class TesTitik:

```

System.out.println("X: " + b.x);
System.out.println("Y: " + b.y);

```

yang akan menghasilkan output seperti berikut:



```

C:\Windows\system32\cmd.exe
Konstruktor titik dijalankan!
Konstruktor titik 2 dijalankan!
X: 10
Y: 10
Press any key to continue . . .

```

## PRAKTIK 4:Overloading konstruktor

```

public class Pegawai
{
    String NamaPegawai;
    int IdPegawai;
    String PosisiPegawai;

    Pegawai (String nama,int Id, String posisi) // konstruktor
    {
        NamaPegawai = nama;
        IdPegawai = Id;
        PosisiPegawai = posisi;
    }

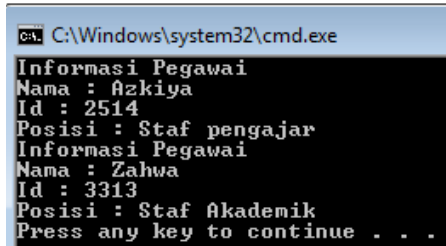
    Pegawai () //konstruktor
    {
        NamaPegawai = "Azkiya";
        IdPegawai = 2514;
        PosisiPegawai = "Staf pengajar";
    }

    void Show()
    {
        System.out.println("Informasi Pegawai");
        System.out.println("Nama : "+NamaPegawai);
        System.out.println("Id : "+IdPegawai);
        System.out.println("Posisi : "+PosisiPegawai);
    }

    public static void main(String args[])
    {
        Pegawai pegawai1 = new Pegawai();
        Pegawai pegawai2 = new Pegawai("Zahwa",3313,"Staf Akademik");
        pegawai1.Show();
        pegawai2.Show();
    }
}

```

output:



```
C:\Windows\system32\cmd.exe
Informasi Pegawai
Nama : Azkiya
Id : 2514
Posisi : Staf pengajar
Informasi Pegawai
Nama : Zahwa
Id : 3313
Posisi : Staf Akademik
Press any key to continue . . .
```

penjelasan:

dalam program ini, public class Pegawai berfungsi untuk mendeklarasikan class bernama Pegawai yang gunanya untuk mengumpulkan fungsi maupun variabel/atribut. Terdapat beberapa atribut seperti NamaPegawai(String), IdPegawai(int), dan PosisiPegawai(String). Lalu langsung buat konstruktor pertama bernama Pegawai dengan atribut konstruktor di dalamnya seperti String nama,int Id, String posisi. Lakukan itu dengan menuliskan **Pegawai (String nama,int Id, String posisi)**. Di dalam konstruktor tersebut, inisialisasikan atribut-atribut milik konstruktor Pegawai dengan atribut-atribut milik Pegawai, yaitu dengan menuliskan baris kode berikut:

```
NamaPegawai = nama;
IdPegawai = Id;
PosisiPegawai = posisi;
```

Setelah selesai membuat konstruktor pertama, kita langsung membuat konstruktor overloading yaitu Pegawai (). Di dalamnya kita beri nilai pada atribut-atributnya seperti: NamaPegawai = "Azkiya"; IdPegawai = 2514; PosisiPegawai = "Staf pengajar";

Lalu kita buat method yang bakal digunakan untuk konstruktor pertama dan konstruktor overloading yang fungsinya untuk menampilkan output yang bernama Show()

```
System.out.println("Informasi Pegawai")      = menampilkan tulisan "Informasi Pegawai"
System.out.println("Nama : "+NamaPegawai);    = menampilkan nama pegawai
System.out.println("Id : "+IdPegawai);        = menampilkan id pegawai
System.out.println("Posisi : "+PosisiPegawai); = menampilkan posisi pegawai
```

Tulis **public static void main(String args[])** sebelum kita membuat objek. Disini kita membuat 2 objek yaitu pegawai1 yaitu objek untuk konstruktor overloading dan pegawai 2 yang justru untuk konstruktor pertama. Objek pegawai1 untuk konstruktor overloading karena tadi saat mendeklarasikan konstruktor, atribut beserta nilai nya sudah diisi yaitu **NamaPegawai = "Azkiya"; IdPegawai = 2514; PosisiPegawai = "Staf pengajar";**. Sementara pegawai2 untuk konstruktor pertama karena pada saat pertama kali mendeklarasikan konstruktor, kita menulis atribut dan menginisialisasinya dengan atribut milik class Pegawai, bukan menulis atribut dan memberikan value/nilai di dalamnya. Kita baru mengisi value/nilai konstruktor pertama pada saat pembuatan objek. Nah, untuk membuat kedua objek yang sudah saya jelaskan di atas tadi, kita bisa menuliskan baris kode berikut:

```
Pegawai pegawai1 = new Pegawai();
Pegawai pegawai2 = new Pegawai("Zahwa",3313,"Staf Akademik");
```

Untuk menampilkan output, kita panggil method Show() yang tadi sudah kita buat. Untuk memanggil method tersebut, kita bisa menuliskan baris kode berikut:

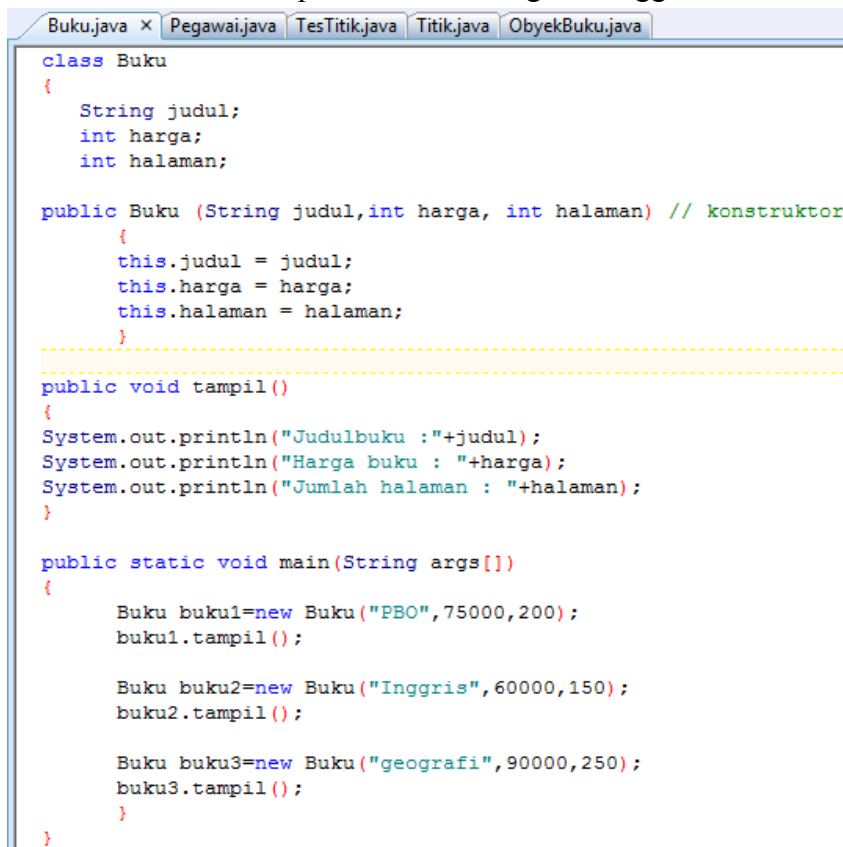
```
pegawai1.Show();
pegawai2.Show();
```

Maka tercetaklah data Informasi Pegawai seperti yang sudah saya cantumkan gambar outputnya tadi.

---

## LATIHAN

Modifikasi class buku pada modul 1 dengan menggunakan konstruktor !



```
class Buku
{
    String judul;
    int harga;
    int halaman;

    public Buku (String judul,int harga, int halaman) // konstruktor
    {
        this.judul = judul;
        this.harga = harga;
        this.halaman = halaman;
    }

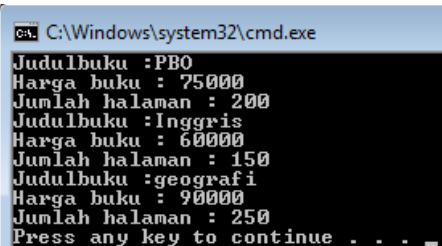
    public void tampil ()
    {
        System.out.println("Judulbuku :"+judul);
        System.out.println("Harga buku : "+harga);
        System.out.println("Jumlah halaman : "+halaman);
    }

    public static void main(String args[])
    {
        Buku buku1=new Buku("PBO",75000,200);
        buku1.tampil();

        Buku buku2=new Buku("Inggris",60000,150);
        buku2.tampil();

        Buku buku3=new Buku("geografi",90000,250);
        buku3.tampil();
    }
}
```

Output:



```
C:\Windows\system32\cmd.exe
Judulbuku :PBO
Harga buku : 75000
Jumlah halaman : 200
Judulbuku :Inggris
Harga buku : 60000
Jumlah halaman : 150
Judulbuku :geografi
Harga buku : 90000
Jumlah halaman : 250
Press any key to continue . . .
```

penjelasan:

Inti dari latihan 1 ini adalah memodifikasi class Buku supaya memiliki konstruktor. Berikut saya jelaskan proses pengerjaannya:

Pertama, kita buat class bernama Buku yang berfungsi untuk mendeklarasikan class bernama Buku yang gunanya untuk mengumpulkan fungsi maupun variabel/atribut. Terdapat beberapa atribut seperti judul(String), harga(int), dan halaman(String). Lalu buat konstruktor bernama Buku dengan atribut konstruktor di dalamnya seperti String judul,int harga, String halaman. Lakukan itu dengan menuliskan public Buku (String judul, int harga, int halaman). Di dalam konstruktor tersebut, inisialisasikan atribut-atribut milik konstruktor Buku dengan atribut-atribut milik class Buku, yaitu dengan menuliskan baris kode berikut:

**this.judul = judul;**

**this.harga = harga;**

**this.halaman = halaman;**

Lalu buat method tampil() yang berfungsi untuk menampilkan output yaitu dengan menuliskan:

```
System.out.println("Judulbuku :"+judul);
System.out.println("Harga buku : "+harga);
System.out.println("Jumlah halaman : "+halaman);
```

Tulis **public static void main(String args[])** sebelum kita membuat objek.

Buat objek buku1 dengan **Buku buku1=new Buku("PBO",75000,200);** dan tampilkan dengan **buku1.tampil();**

Buat objek buku2 dengan **Buku buku2=new Buku("Inggris",60000,150);** dan tampilkan dengan **buku2.tampil();**

Buat objek buku1 dengan **Buku buku3=new Buku("geografi",90000,250);** dan tampilkan dengan **buku3.tampil();**

dengan begitu, kita telah berhasil membuat program dengan konstruktor yang memunculkan 3 buah buku beserta data-data nya.

---

## TUGAS

Buatlah class Komputer lengkap dengan atribut, method dan konstruktor !

```
Komputer.java x
1  public class Komputer {
2
3      String merk;
4      String graphic;
5      String ram;
6      String os;
7      String processor;
8
9      public Komputer(String merk, String graphic,
10 String ram, String os, String processor){
11          this.merk = merk;
12          this.graphic = graphic;
13          this.ram = ram;
14          this.os = os;
15          this.processor = processor;
16      }
17
18      public String getMerk(){
19          return merk;
20      }
21
22      public String getGraphic(){
23          return graphic;
24      }
25
26      public String getRam(){
27          return ram;
28      }
29
30      public String getOs(){
31          return os;
32      }
33
34      public String getProcessor(){
35          return processor;
36      }
37      @Override
38      public String toString(){
39
40          return("Aku punya komputer baru merknya " + this.getMerk() +
41              ", graphic card nya " + this.getGraphic() +
42              ", kapasitas RAM nya " + this.getRam()+
43              ", operating system nya " + this.getOs() +
44              ", dan processor nya " + this.getProcessor() +
45              ". Aku senang sekali.");
```

```

46     }
47     public static void main(String[] args) {
48         Komputer canggi = new Komputer
49             ("HP OMEN 15-DC0036TX", "NVIDIA GeForce GTX 1050 Ti",
50             "16 GB DDR4", "Windows 10", "Intel Core i7");
51         System.out.println(canggi.toString());
52     }
53 }

```

Output:

penjelasan:

Sebelum saya menjelaskan proses penulisan script programnya, saya akan tunjukkan letak atribut, method, dan konstruktor nya. Untuk atribut, bisa ditemukan dibawah public class Komputer yaitu **String merk; String graphic; String ram; String os; dan String processor;**. Untuk method bisa kita temukan di baris-baris kode yang menggunakan method getter yaitu **public String getMerk(), public String getGraphic(), public String getRam(), public String getOs(), dan public String getProcessor()** dan juga ada pada method **toString()**.

Untuk konstruktor bisa ditemukan di baris kode 9-16 berikut:

```

public Komputer(String merk, String graphic,
String ram, String os, String processor){
    this.merk = merk;
    this.graphic = graphic;
    this.ram = ram;
    this.os = os;
    this.processor = processor;
}

```

Untuk proses penulisan script programnya, kita mulai dari membuat class bernama Komputer yang memiliki modifier public yang berfungsi untuk mendeklarasikan class bernama Komputer yang gunanya untuk mengumpulkan fungsi maupun variabel/atribut. Di dalamnya sudah ada atribut-atribut yang saya tuliskan tadi. Setelah itu kita deklarasikan konstruktor dengan **public Komputer(String merk, String graphic, String ram, String os, String processor)**. Di dalam konstruktor tersebut, inisialisasikan atribut-atribut milik konstruktor Komputer dengan atribut-atribut milik class Komputer, yaitu dengan cara menuliskan **this.merk = merk; this.graphic = graphic; this.ram = ram; this.os = os; this.processor = processor;**

Jika sudah, maka kita buat method-method yang dibutuhkan. Untuk yang getter saya beri satu contohnya saja yaitu:

```

public String getMerk(){
    return merk;
}

```

Method getter ini akan membantu kita untuk mengakses data yang di enkapsulasi dengan modifier private yang tidak dapat diakses langsung dari luar class. Getter mempunyai nilai kembalian sesuai dengan tipe data yang diambil. Kita menggunakan ini supaya tipe data yang diambil nantinya bisa di return.

Lalu setelah selesai membuat baris-baris kode getter method, kita langsung membuat method **toString()**. Method ini digunakan untuk merepresentasikan sebuah objek, kedalam tipe String. Jika kita menampilkan sebuah objek, maka secara implisit (tidak kita ketahui) sebenarnya compiler



sudah memanggil method toString(). Kita dapat melakukan overriding pada method tersebut, dengan begitu, kita bisa mengimplementasikannya untuk memperoleh output yang diinginkan, seperti atribut-atribut dari suatu objek. Dalam kasus dalam tugas ini, di dalam toString() terdapat return yang berfungsi untuk membuat kalimat + atribut yang nantinya merupakan output program yang telah kita buat.

Lalu kita return dengan membuat kalimat sesuka kita ditambah menginisialisasi method-method getter di dalamnya contohnya dengan menuliskan **"Aku punya komputer baru merknya " + this.getMerk()**

Lalu, kita membuat objek bernama "canggih" dan menginisialisasi objek tersebut dengan argumen di konstruktor dengan baris kode dibawah:

```
public static void main(String[] args){  
    Komputer canggih = new Komputer  
        ("HP OMEN 15-DC0036TX"," NVIDIA GeForce GTX 1050 Ti",  
        "16 GB DDR4", "Windows 10", " Intel Core i7");
```

Terakhir, kita tampilkan outputnya dengan: **System.out.println(canggih.toString());**

---

## KESIMPULAN

Di dalam laporan ini, kita belajar banyak hal tentang konstruktor. Diawali dari praktik 1 yaitu cara mendeklarasikan konstruktor sampai ke tugas yang intinya adalah kompilasi dari modul 1-4 yaitu membuat program yang ada atribut, method, dan konstruktor nya. Hal utama yang kita pelajari di pertemuan ini adalah jenis-jenis konstruktor yaitu:

### **Konstruktor Default:**

Apabila anda tidak mendeklarasikan satu pun konstruktor, maka Java secara otomatis menambahkan konstruktor default ke dalam class yang kita buat walaupun tidak kelihatan pada kode program. Apabila kita mendeklarasikan satu atau lebih konstruktor maka java tidak akan menambahkan kostruktor default.

### **Konstruktor Overloading:**

Pada konstruktor juga berlaku overloading, artinya boleh mendeklarasikan lebih dari satu konstruktor, asalkan memiliki parameter yang berbeda – beda.

Pada konstruktor :

1. Memiliki konsep yang sama dengan overloading method.
2. Dibedakan berdasarkan paramater (jumlah atau tipe data).

Struktur penulisan konstruktor:

```
<modifier> <className> (<parameter>) {  
    <statement>  
}
```

**Terima Kasih**