18

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN KE-3

****

**Disusun oleh :**

**NAMA : FERDI DIRGANTARA**

**NIM : 175410039**

**JURUSAN : TEKNIK INFORMATIKA**

**JENJANG : S1**

LABORATORIUM TERPADU

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTERAKAKOMYOGYAKARTA

2019

1. **TEORI SINGKAT**

**Method**

Sebuah method menjelaskan behaviour dari sebuah object. Method juga dikenal sebagai fungsi atau prosedur. Pendeklarasian method biasa dituliskan seperti berikut ini :

<modifier> <returnType> <name>(<parameter>\*) {

<statement>\*

}

dimana,

<modifier> dapat menggunakan beberapa modifier yang berbeda

<returnType> dapat berupa seluruh tipe data, termasuk void

<name> identifier atas class

<parameter> ::= <tipe\_parameter> <nama\_parameter>[,]

**Method Tanpa Return Value**

Jenis method ini ditandai dengan return type yang berupa void dan pada bagian statement tidak terdapat keyword return.

**Method Dengan Return Value**

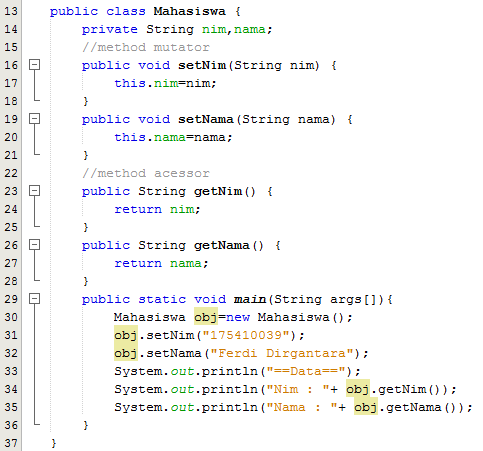
Jenis method ini ditandai dengan return type selain void dan pada bagian statement terdapat keyword return.

**Method Overloading**

Bahasa java mendukung method overloading , java dapat membedakan beberapa method dengan nama yang sama di dalam sebuah kelas namun parameternya berbeda.

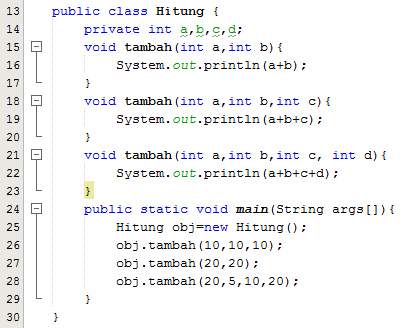
1. **PEMBAHASAN LISTING**

Kelas Mahasiswa



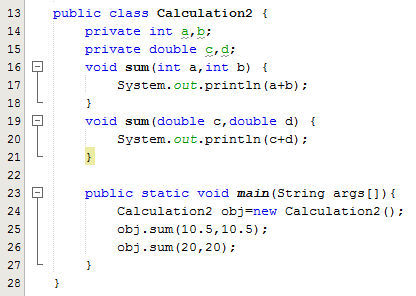
* Kelas Mahasiswa diatas digunakan untuk menambahkan data mahasiswa dengan beberapa method yang telah disediakan.
* Method setNim() digunakan untuk mengatur atau menambahkan nim mahasiswa.
* Method setNama() digunakan untuk mengatur atau menambahkan nama mahasiswa.
* Method getNim() digunakan untuk mengambil Nim dari mahasiswa dan memberikan nilai return berupa nim mahasiswa.
* Method getNama() digunakan untuk mengambil nama dari mahasiswa dan memberikan nilai return berupa nama mahasiswa.
* Pada method main() dibuat objek untuk kelas Mahasiswa dengan nama obj dan kemudian digunakan untuk setNim dan setNama dan juga data tersebut kemudian ditampilkan dengan menggunakan method getNim dan getNama.

Kelas Hitung



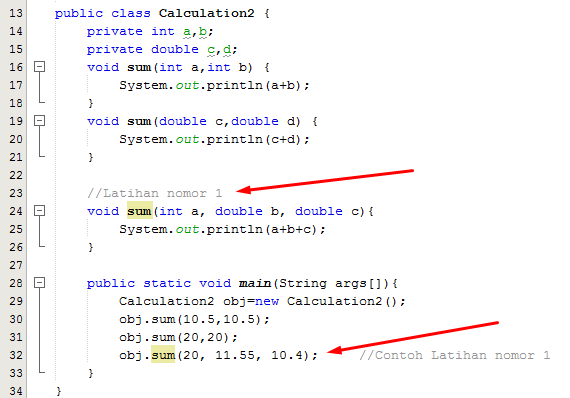
* Kelas Hitung diatas merupakan contoh kelas mutuator dan method accessor yang digunakan untuk menghitung nilai suatu bilangan dengan yang diakses dengan method tambah beserta beberapa parameter yang berbeda yang nanti akan disesuaikan langsung oleh programnya.
* Dari ketiga method tambah() diatas memiliki parameter yang berbeda-beda yang mana program akan menjalankan method yang nilai parameternya sesuai dengan data yang dimasukkan pada saat method tambah() diakses pada method main().

Kelas Calculation2



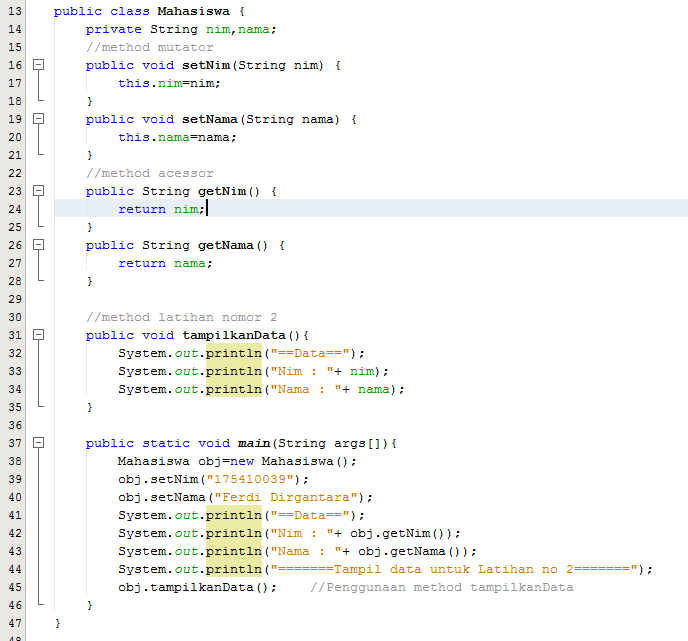
* Method Calculation2 diatas adalah contoh implementasi method overloading yang mana saat menggunakan method sum diatas dengan nilai parameter yang berbeda akan memberikan output yang berbeda juga.

1. **PEMBAHASAN LATIHAN**
2. Modifikasi praktik 3 dengan menambahkan method sum dengan tipe parameter yang berbeda selanjutnya panggil melalui main program



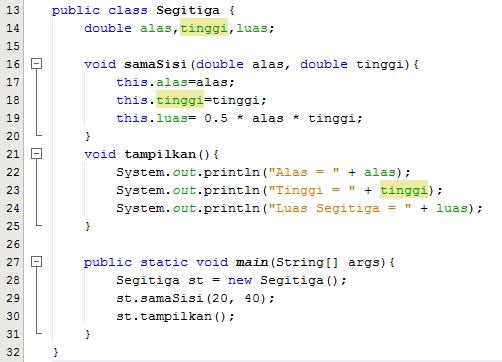
Ditambahkan sebuah method overloading untuk menerima 3 buah parameter pada method sum() dan juga menambahkan method sum().

1. Modifikasi praktik 1 dengan menambahkan method TampilkanData yang digunakan untuk menampilkan data nim dan nama



Program diatas membuat method tampilkanData() yang digunakan untuk menampilkan data mirip seperti yang tertera pada method main pada praktikum.

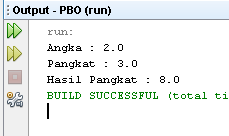
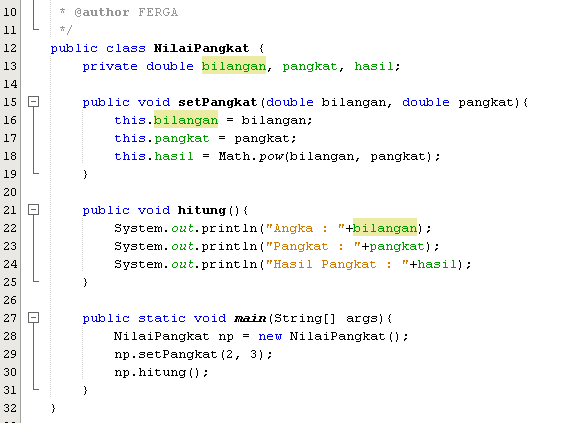
1. Buatlah kelas lengkap dengan atribut dan operasi untuk menentukan luas segitiga



Program diatas menghitung luas segitiga dengan mengakses method samaSisi() untuk memasukkan alas dan tinggi dari segitiga sama sisi yang ingin dihitung luasnya dan yang ditampilkan pada saat program dijalankan adalah alas, tinggi dan luas segitiganya.

1. **TUGAS**

Buatlah program lengkap dengan atribut dan operasi yang digunakan untuk menghitung nilai pangkat dari suatu bilangan.



1. **KESIMPULAN**

Mahasiswa dapat membuat, menggunakan, dan membedakan method-method yang terdapat pada pemrograman Java seperti method tanpa return value, method dengan return value dan method overloading.

1. **LISTING**

*Terlampir*