

|  |
| --- |
| **Algoritma dan Struktur Data 1** |
|  |
| **Modul 5** |
| **Fungsi dan Prosedur Program** |

**Disusun oleh:**

**Dwi Intan Af’idah, S.T., M.Kom**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK HARAPAN BERSAMA**

**TAHUN AJARAN 2020/2021**

Daftar Isi

[Daftar Isi ii](#_Toc57737402)

[1 Pengantar 1](#_Toc57737403)

[2 Cara Membuat Fungsi di Java 2](#_Toc57737404)

[2.1 Fungsi Tanpa Parameter 3](#_Toc57737405)

[2.2 Fungsi dengan Parameter dan Retrun 4](#_Toc57737406)

[3 Fungsi Static dan Non Static 5](#_Toc57737407)

[4 Variabel Global dan Lokal 6](#_Toc57737408)

[5 Tugas 11: Fungsi 7](#_Toc57737409)

1. Pengantar

* Pada contoh program di pembahasan sebelumnya, kita hanya menulis kode intruksi pada fungsi main() saja.
* Fungsi main() adalah fungsi utama dalam program Java. Semua kode yang kita tulis di dalamnya, akan langsung dieksekusi.
* Tapi masalahnya sekarang:
* “Bagaimana kalau kita membuat program yang cukup besar, apakah kita masih bisa menulis semua kodenya di dalam fungsi main()?”
* Bisa-bisa saja, tapi kurang efektif dan akan menghabiskan banyak tenaga untuk mengetik kodenya.
* Belum lagi kalau ada error…
* “Lalu solusinya bagaimana?”
* Solusinya menggunakan prosedur/fungsi.
* Prosedur/fungsi dapat memecah program menjadi sub-sub program **(subrutin),** sehingga kita bisa membuat program lebih efisien.
* Penggunaan prosedur/fungsi dapat mengurangi pengetikan kode yang berulang-ulang.
* Pada kesempatan ini, kita akan belajar menggunakan prosedur/fungsi pada Java untuk membuat program.
* Pertama, kita kenalan dulu dengan prosedur dan fungsi. Setelah itu, dilanjutkan dengan contoh program.
* Prosedur, Fungsi, dan Method itu sama.
* Prosedur adalah sebutan untuk fungsi yang tidak mengembalikan nilai. Fungsi ini biasanya ditandai dengan kata kunci **void.**
* Fungsi adalah sebutan untuk fungsi yang mengembalikan nilai.
* Method adalah fungsi yang berada di dalam Class. Sebutan ini, biasanya digunakan pada OOP.

1. Cara Membuat Fungsi di Java

Fungsi harus dibuat atau ditulis di dalam class.

Struktur dasarnya seperti ini:

*static* TypeDataKembalian namaFungsi(){

// statemen atau kode fungsi

}

Penjelasan:

* Kata kunci static, artinya kita membuat fungsi yang dapat dipanggil tanpa harus membuat instansiasi objek. 🡪 Nanti saya jelaskan.
* TypeDataKembalian adalah tipe data dari nilai yang dikembalikan setelah fungsi dieksekusi.
* namaFungsi() adalah nama fungsinya. Biasanya ditulis dengan huruf kecil di awalnya. Lalu, kalau terdapat lebih dari satu suku kata, huruf awal di kata kedua ditulis kapital.
* Contoh 1:

*static* void ucapSalam(){

System.out.println("Selamat Pagi");

}

* Tipe data void artinya kosong, fungsi tersebut tidak mengebalikan nilai apa-apa.
* Contoh 2:

*static* TipeData namaFungsi(TipeData namaParameter, TipeData namaParameterLain){

// kode fungsi

}

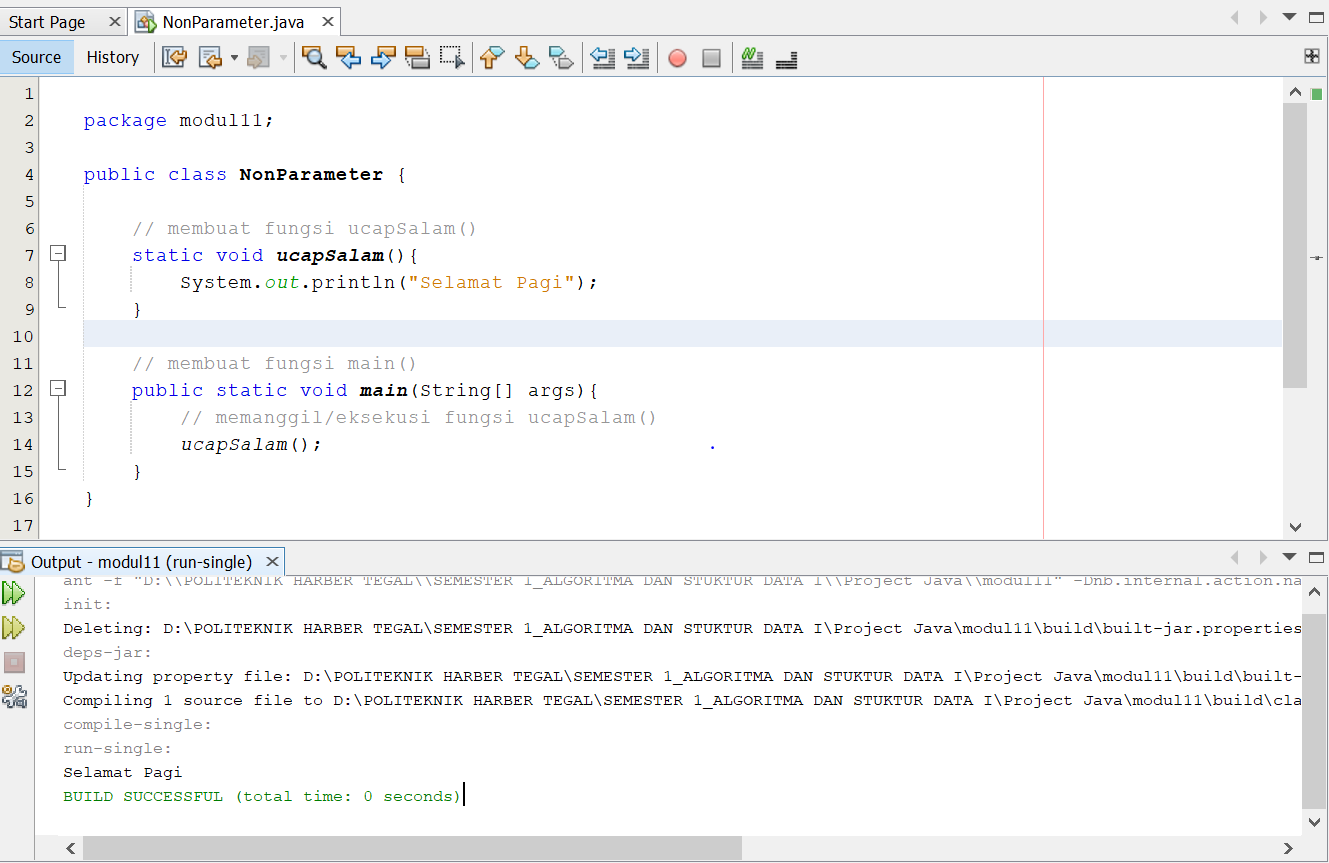
*static* int luasPersegi(int sisi){

int luas = sisi \* sisi;

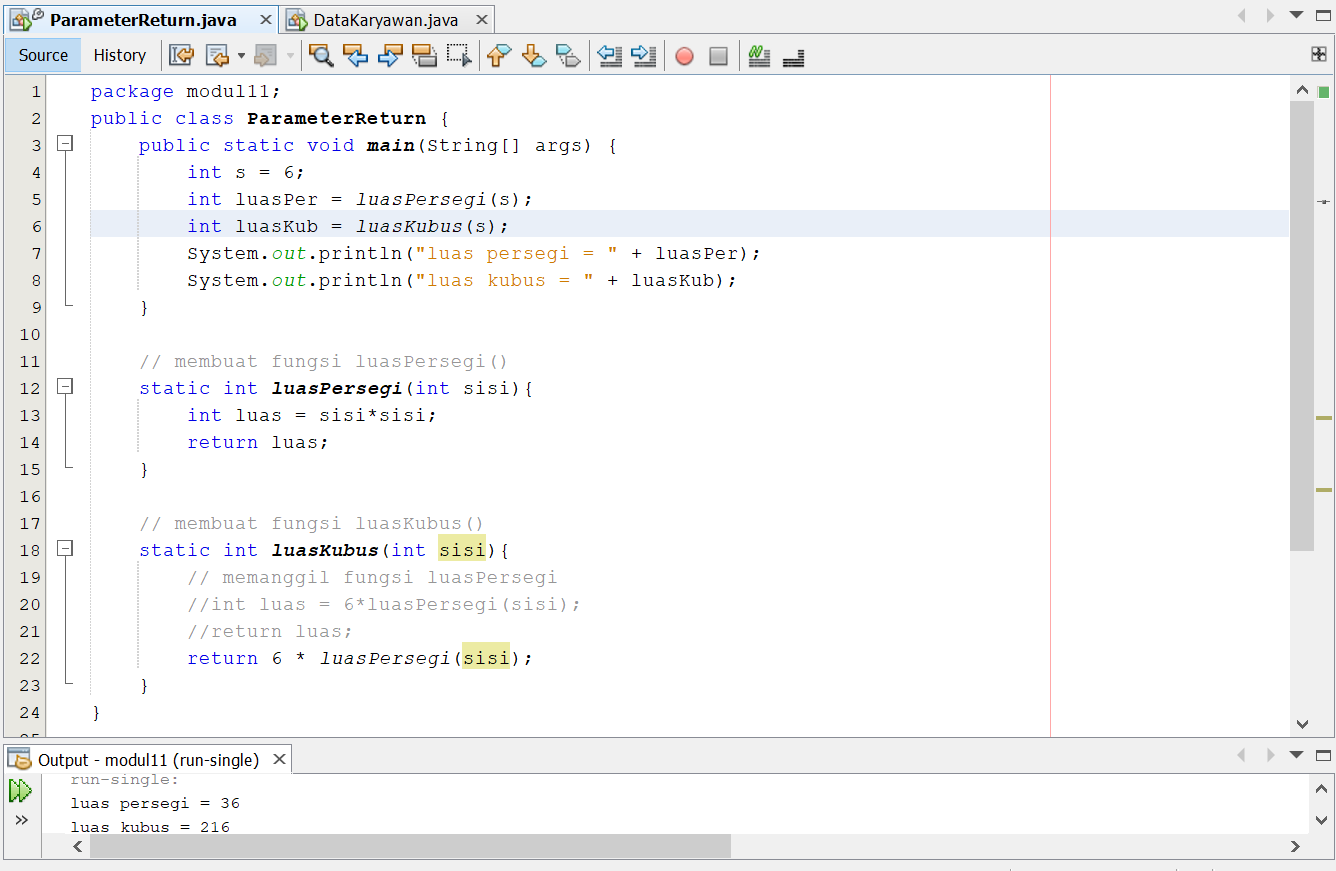
return luas;

}

* Pada contoh tersebut, kita membuat sebuah parameter bernama sisi.
* Kemudian fungsi akan mengembalikan nilai (return) dengan tipe int (integer) dari variabel luas.
  1. Fungsi Tanpa Parameter



* 1. Fungsi dengan Parameter dan Retrun



1. Fungsi Static dan Non Static

Pada contoh-contoh diatas, kita menggunakan kata kunci static sebelum membuat fungsi.

Kata kunci static akan membuat fungsi dapat dieksekusi langsung, tanpa harus membuat instansiasi objek dari class.

*public* *class* FungsiStatic {

// Fungsi non-static

void makan(String makanan){

System.out.println("Hi!");

System.out.println("Saya sedang makan " + makanan);

}

// fungsi static

*static* void minum(String minuman){

System.out.println("Saya sedang minum " + minuman);

}

// fungsi main

*public* *static* void main(String[] args) {

// pemanggilan fungsi static

minum("Kopi");

// mambuat instansiasi objek saya dari class FungsiStatic

FungsiStatic saya = new FungsiStatic();

// pemanggilan fungsi non-static

saya.makan("Nasi Goreng");

}

}

1. Variabel Global dan Lokal

* Variabel Global 🡪 variabel yang bisa diakses dari semua fungsi
* Variabel Lokal 🡪 variabel yang hanya bisa diakses di dalam fungsi tempat dia berada

*class* ProgramKu{

// ini variabel global

*static* String nama = "Programku";

*static* String version = "1.0.0";

*static* void help(){

// ini variabel lokal

String nama = "Program Anda";

// mengakses variabel global di dalam fungso help()

System.out.println("Nama: " + nama);

System.out.println("Versi: " + version);

}

*public* *static* void main(String args[]){

// panggil fungsi help()

help();

System.out.println("Nama: " + nama);

System.out.println("Versi: " + version);

}

}

Hasil Output

Nama: Petani Kode

Versi: 1.0.0

Nama: Programku

Versi: 1.0.0

1. Tugas 11: Fungsi

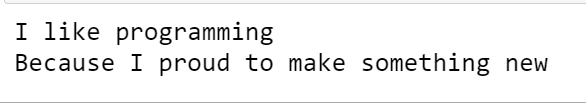
Tulislah kode **program** untuk menjelaskan baris kode untuk kasus di bawah ini:

(Disarankan terdapat kalimat **comment** tentang **nama creator** dan penjelasan **kode program**)

1. Membuat 2 Fungsi (pengembangan dari contoh di subbab 1.1)

Dengan ketentuan:

1. Terdapat 2 fungsi
2. Fungsi pertama 🡪 berisi perintah prinln (“Because I proud to make something new”)
3. Fungsi kedua 🡪 berisi perintah println (“I like programming”)
4. Dengan output program seperti di bawah ini:



1. Fungsi dan Parameter

Dengan ketentuan:

1. Fungsi: luas persegi panjang
2. Panggil fungsi luas persegi panjang dengan 🡪 panjang1=5 dan lebar1=10
3. Panggil fungsi luas persegi panjang dengan 🡪 panjang2=100 dan lebar2=50