

JOBSHEET 13
DASAR PEMROGRAMAN



ALVINO VALERIAN D.R
2341720027
D-IV TEKNIK INFORMATIKA

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023

Pertanyaan!

1. Apakah fungsi tanpa parameter selalu digunakan untuk fungsi "void" saja?

Tidak, fungsi tanpa parameter tidak selalu digunakan untuk fungsi "void" saja.

Fungsi tanpa parameter dapat mengembalikan nilai (non-void) dengan menentukan tipe data pengembalian di deklarasi fungsi

2. Apakah bisa kalimat "Thank you for.....dst" dituliskan tanpa fungsi

UcapanTerimaKasih? modifikasi kode program tersebut untuk dapat menampilkan

suatu kalimat tanpa dan dengan menggunakan fungsi!

```
public class TerimaKasih_05 {  
    Run | Debug  
    public static void main(String[] args) {  
        System.out.print("Thank you for being  
        "you inspired in me a love for learni  
    }  
}
```

Thank you for being the best teacher in the world.

you inspired in me a love for learning and made me feel like i could ask you anything.

PS D:\alvino\1.Daspro prak\minggu13>

3. Apakah keuntungan menggunakan fungsi di dalam program?

fungsi membantu meningkatkan struktur dan organisasi program, membuatnya lebih efisien dan mudah dikelola

Pertanyaan 2!

1. Apakah kegunaan suatu parameter di dalam fungsi?

Parameters digunakan dalam fungsi untuk menerima input dari pemanggil fungsi.

Mereka memungkinkan fungsi untuk menerima nilai atau data yang dapat berbeda setiap kali fungsi dipanggil, sehingga fungsi dapat dijalankan dengan argumen yang bervariasi.

2. Apakah parameter sama dengan variabel? jelaskan!

Parameters digunakan dalam fungsi untuk menerima input dari pemanggil fungsi.

Mereka memungkinkan fungsi untuk menerima nilai atau data yang dapat berbeda setiap kali fungsi dipanggil, sehingga fungsi dapat dijalankan dengan argumen yang bervariasi.

3. Apakah parameter di dalam Bahasa pemrograman Java hanya digunakan untuk melewati data input saja? Bagaimana dengan data output?

Tidak, parameter di dalam Bahasa pemrograman Java tidak hanya digunakan untuk melewati data input. juga digunakan untuk mengembalikan data output.

Dengan menggunakan kata kunci `return` dalam sebuah metode, nilai yang dihasilkan di dalam fungsi dapat dikembalikan dan diakses di tempat pemanggilan. Jadi, parameter dapat berperan dalam menerima input dan mengembalikan output dari sebuah fungsi atau metode dalam Java.

Pertanyaan3!

1. Jelaskan mengenai, kapan suatu fungsi membutuhkan nilai kembalian (return)!

Jika fungsi ingin menggunakan nilai dan pengembalian nilai

2. Apakah bisa ditambahkan statement `System.out.println` di dalam fungsi dengan nilai

kembalian? Apa pengaruh nya?

Bisa `System.out.println` akan mencetak output yang berasal dari nilai kembalian pada fungsi yang dikembalikan dan hal ini sama sekali tidak mempengaruhi nilai dari nilai kembalian.

3. Apakah bisa fungsi tanpa nilai kembalian di dalam fungsi main dipanggil tanpa dilewatkan ke variabel? seperti pada percobaan 1? Jelaskan!

Ya, fungsi tanpa nilai kembalian dapat dipanggil langsung dalam fungsi main tanpa perlu dilewatkan ke variabel.

Pertanyaan4!

1. Berdasarkan uji coba pada percobaan 4, manakah fungsi yang akan eksekusi pertamakali? Jelaskan!

Fungsi main yang akan dieksekusi terlebih dahulu dikarenakan fungsi main() merupakan titik awal eksekusi program Java. Setiap program Java harus memiliki fungsi main() dan akan berisi instruksi-instruksi yang akan dijalankan oleh program.

2. Manakah tata cara penulisan fungsi benar di dalam class? Di atas fungsi main ataukah dibawah fungsi main? Jelaskan!

penempatan fungsi-fungsi selain fungsi main di atas atau di bawah fungsi main bersifat fleksibel dan tergantung pada preferensi pengembang dan struktur logis program yang diinginkan.

3. Modifikasi program di atas dengan menambahkan fungsi **UcapanTambahan()** dengan input parameter String. Fungsi **UcapanTambahan()** berisi ucapan tambahan yang ingin disampaikan kepada penerima ucapan.

```
1  import java.util.Scanner;
2
3  public class TerimaKasih_05 {
4      public static String PenerimaUcapan(){
5          Scanner sc=new Scanner(System.in);
6          System.out.println("Tuliskan Nama Orang Yang Ingin Kamu Beri Ucapan: ");
7          String namaOrang = sc.nextLine();
8          sc.close();
9          return namaOrang;
10     }
11     public static void UcapanTerimakasih(){
12         String nama = PenerimaUcapan();
13         System.out.println("Thank " + nama + " you for being the best teacher in the world.\n"+
14             "you inspired in me a love for learning and made me feel like i could ask you anything.");
15     }
16     public static void ucapanTambahan(String ucapanTambahan){
17         System.out.println(ucapanTambahan);
18     }
19     public static void main(String[] args){
20         UcapanTerimakasih();
21         String ucapanTambahan = ("I love You More");
22         ucapanTambahan(ucapanTambahan);
23     }
24
25
26
27
28 }
```

Pertanyaan 5!

1. Jelaskan mengapa penulisan parameter di praktikum 5 di tulis dengan **int... a!**
int... a adalah **varargs** ,dapat menerima sejumlah argumen, dan akan dianggap sebagai elemen-elemen dari array **a** di dalam fungsi.
2. Sebutkan kegunaan varargs dalam implementasi kode program untuk menyelesaikan permasalahan dalam dunia nyata! (minimal 3)

Mengatasi Jumlah Argumen Variabel: Varargs memungkinkan kita menangani situasi di mana jumlah argumen yang diperlukan dapat bervariasi., Penggunaan yang **Fleksibel:**

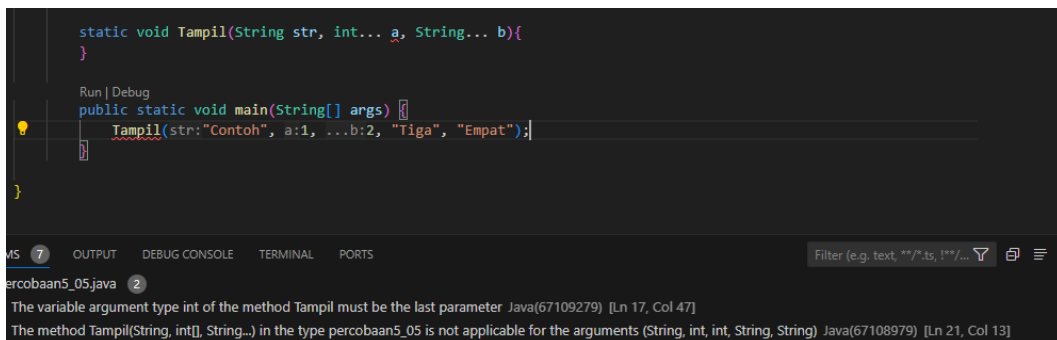
Varargs memberikan fleksibilitas saat kita tidak tahu pasti berapa banyak argumen yang akan digunakan oleh pemanggil fungsi. Hal ini memudahkan pemrogram untuk menggunakan metode dengan cara yang lebih intuitif tanpa perlu membuat banyak variasi metode.,

Mengurangi Kode Berulang:

Dengan **varargs**, kita dapat menghindari penulisan banyak **overloading** metode untuk menangani jumlah argumen yang berbeda.

3. Bisakah kita menggunakan dua tipe data **varaargs** dalam satu fungsi?Berikan contohnya!

Tidak bisa varargs harus menggunakan 1 tipe data yang konsisten



```
static void Tampil(String str, int... a, String... b){
}

public static void main(String[] args) {
    Tampil(str:"Contoh", a:1, ...b:2, "Tiga", "Empat");
}
```

The variable argument type int of the method Tampil must be the last parameter Java(67109279) [Ln 17, Col 47]

The method Tampil(String, int[], String...) in the type percobaan5_05 is not applicable for the arguments (String, int, int, String, String) Java(67108979) [Ln 21, Col 13]

Pertanyaan6!

1. Sebutkan tahapan dan urutan eksekusi pada percobaan 6 di atas!

Deklarasi Variabel,Panggil Fungsi HitungLuas,Cetak Luas Persegi Panjang,Cetak Volume Balok,Penutup Scanner

2. Apakah output dari program dibawah ini kemudian jelaskan alur jalannya programtersebut!

```
01234567  
PS D:\alvino\1.Daspro prak\minggu13>
```

Alur akan dimulai dari fungsi psvm. Command pada line 17 akan dieksekusi untuk pertama kali guna mengisi variabel temp. Command pada line 17 akan masuk ke parameter fungsi Jumlah, sehingga nanti nilai dari temp akan menjadi 2. setelah itu, command pada baris ke 18 akan menjadi yang kedua untuk dieksekusi dengan cara melewatinya parameter dari fungsi TampilJumlah. Setelah memasuki fungsi TampilJumlah bil 1 akan bernilai sama dengan temp kemudian bil2 bernilai sama dengan 5. Karena pada fungsi TampilJumlah terdapat

TampilHinggaKei(Jumlah(bil,bil2)), maka command yang berada didalam kurung yang akan diselesaikan terlebih dahulu maka pada saat memasuki fungsi TampilJumlah akan langsung diarahkan pada fungsi Jumlah, setelah dari fungsi Jumlah, command ini bernilai 7, setelah itu diarahkan ke fungsi TampilHinggaKei, melalui parameter fungsi tersebut sehingga menggantikan nilai menjadi 7. Program akan ditutup dengan sout j didalam perulangan dimana perulangan ini akan dimulai saat j yang bernilai 1 kurang dari atau sama dengan l yang bernilai 7

3. Pada saat apakah fungsi yang kita buat harus menggunakan parameter atau tidak?Pada saat apakah fungsi yang kita buat harus memiliki nilai kembalian atau tidak?Jelaskan!

Fungsi sebaiknya menggunakan parameter ketika memerlukan data atau informasi dari luar untuk diolah di dalam fungsi tersebut. Parameter memungkinkan fungsi menerima input yang dapat bervariasi sesuai dengan kebutuhan.

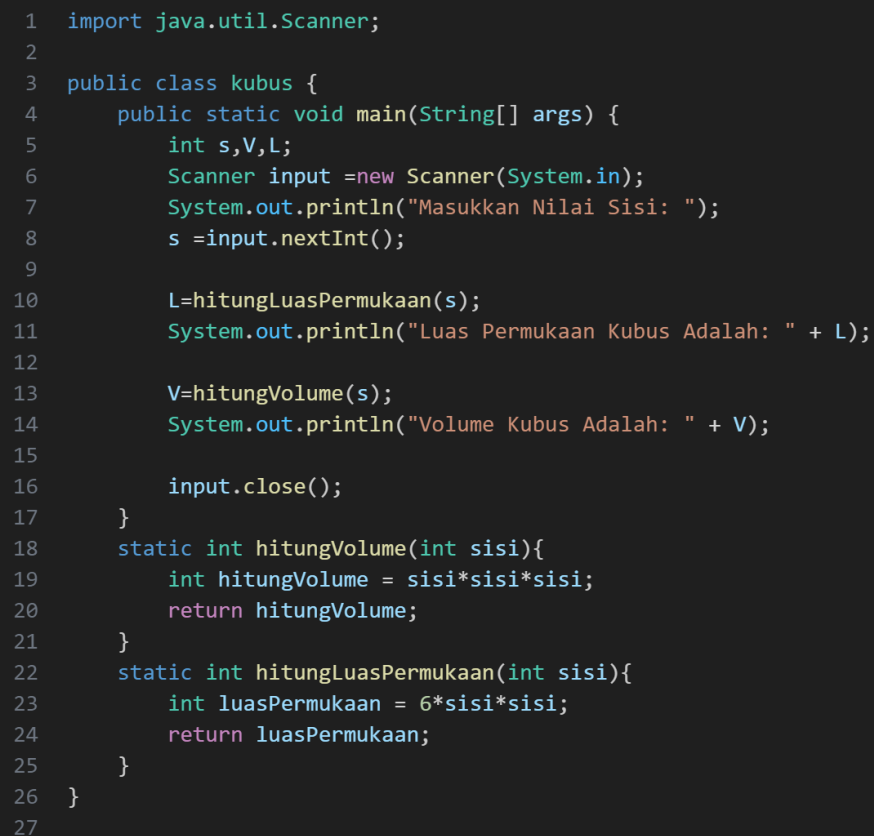
Fungsi sebaiknya memiliki nilai kembalian ketika hasil dari eksekusi fungsi tersebut perlu diakses atau digunakan di tempat pemanggilan.

tidak selalu setiap fungsi harus memiliki parameter atau nilai kembalian. Beberapa fungsi mungkin tidak memerlukan input tambahan (parameter)

atau tidak perlu mengembalikan nilai (nilai kembalian) tergantung pada tujuan dan tugas spesifik yang diemban oleh fungsi tersebut.

TUGAS 3

1



```
1  import java.util.Scanner;
2
3  public class kubus {
4      public static void main(String[] args) {
5          int s,V,L;
6          Scanner input =new Scanner(System.in);
7          System.out.println("Masukkan Nilai Sisi: ");
8          s =input.nextInt();
9
10         L=hitungLuasPermukaan(s);
11         System.out.println("Luas Permukaan Kubus Adalah: " + L);
12
13         V=hitungVolume(s);
14         System.out.println("Volume Kubus Adalah: " + V);
15
16         input.close();
17     }
18     static int hitungVolume(int sisi){
19         int hitungVolume = sisi*sisi*sisi;
20         return hitungVolume;
21     }
22     static int hitungLuasPermukaan(int sisi){
23         int luasPermukaan = 6*sisi*sisi;
24         return luasPermukaan;
25     }
26 }
27
```

```

1  import java.util.Scanner;
2
3  public class TugasMahasiswa {
4      static Scanner input = new Scanner(System.in);
5
6      public static void inputNilai(int[][] nilaiMhs) {
7          for (int i = 0; i < nilaiMhs.length; i++) {
8              for (int j = 0; j < nilaiMhs[i].length; j++) {
9                  System.out.print("Nilai [" + i + "][" + j + "] = ");
10                     nilaiMhs[i][j] = input.nextInt();
11             }
12         }
13     }
14
15     public static void tampilNilai(int[][] nilaiMhs) {
16         for (int i = 0; i < nilaiMhs.length; i++) {
17             for (int j = 0; j < nilaiMhs[i].length; j++) {
18                 System.out.println("Nilai mhs" + i + " hari" + j + "=" + nilaiMhs[i][j]);
19             }
20         }
21     }
22
23     public static int getHarikeNilaiMax(int[][] nilaiMhs) {
24         int hariKe = 0;
25         int nilaiMax = nilaiMhs[0][0];
26         for (int j = 0; j < nilaiMhs[0].length; j++) {
27             int totalHari = 0;
28             for (int i = 0; i < nilaiMhs.length; i++) {
29                 totalHari += nilaiMhs[i][j];
30             }
31             if (totalHari > nilaiMax) {
32                 nilaiMax = totalHari;
33                 hariKe = j;
34             }
35         }
36         return hariKe + 1;
37     }
38
39     public static int[] getMhskeNilaiMax(int[][] nilaiMhs) {
40         int mhsKe = 0;
41         int hariKe = 0;
42         int nilaiMax = nilaiMhs[0][0];
43
44         for (int i = 0; i < nilaiMhs.length; i++) {
45             for (int j = 0; j < nilaiMhs[i].length; j++) {
46                 if (nilaiMhs[i][j] > nilaiMax) {
47                     nilaiMax = nilaiMhs[i][j];
48                     mhsKe = i;
49                     hariKe = j;
50                 }
51             }
52         }
53
54         int[] hasil = {hariKe + 1, mhsKe + 1, nilaiMax};
55         return hasil;
56     }
57
58     public static void main(String[] args) {
59         int[][] dataNilai = new int[5][7];
60
61         inputNilai(dataNilai);
62         tampilNilai(dataNilai);
63
64         int hariTertinggi = getHarikeNilaiMax(dataNilai);
65         System.out.println("Nilai Tertinggi terdapat pada Hari ke-" + hariTertinggi);
66
67         int[] hasil = getMhskeNilaiMax(dataNilai);
68         System.out.println("Nilai Tertinggi: " + hasil[2] + " pada minggu ke-" + hasil[0] +
69             " oleh Mahasiswa ke-" + hasil[1]);
70     }
71 }
72

```



```

1  import java.util.Scanner;
2
3  public class TugasMahasiswa {
4      static Scanner input = new Scanner(System.in);
5
6      public static void inputNilai(int[][] nilaiMhs) {
7          for (int i = 0; i < nilaiMhs.length; i++) {
8              for (int j = 0; j < nilaiMhs[i].length; j++) {
9                  System.out.print("Nilai [" + i + "][" + j + "] = ");
10                     nilaiMhs[i][j] = input.nextInt();
11             }
12         }
13     }
14
15     public static void tampilNilai(int[][] nilaiMhs) {
16         for (int i = 0; i < nilaiMhs.length; i++) {
17             for (int j = 0; j < nilaiMhs[i].length; j++) {
18                 System.out.println("Nilai mhs" + i + " hari" + j + "=" + nilaiMhs[i][j]);
19             }
20         }
21     }
22
23     public static int getHarikeNilaiMax(int[][] nilaiMhs) {
24         int hariKe = 0;
25         int nilaiMax = nilaiMhs[0][0];
26         for (int j = 0; j < nilaiMhs[0].length; j++) {
27             int totalHari = 0;
28             for (int i = 0; i < nilaiMhs.length; i++) {
29                 totalHari += nilaiMhs[i][j];
30             }
31             if (totalHari > nilaiMax) {
32                 nilaiMax = totalHari;
33                 hariKe = j;
34             }
35         }
36         return hariKe + 1;
37     }
38
39     public static int[] getMhskeNilaiMax(int[][] nilaiMhs) {
40         int mhsKe = 0;
41         int hariKe = 0;
42         int nilaiMax = nilaiMhs[0][0];
43
44         for (int i = 0; i < nilaiMhs.length; i++) {
45             for (int j = 0; j < nilaiMhs[i].length; j++) {
46                 if (nilaiMhs[i][j] > nilaiMax) {
47                     nilaiMax = nilaiMhs[i][j];
48                     mhsKe = i;
49                     hariKe = j;
50                 }
51             }
52         }
53
54         int[] hasil = {hariKe + 1, mhsKe + 1, nilaiMax};
55         return hasil;
56     }
57
58     public static void main(String[] args) {
59         int[][] dataNilai = new int[5][7];
60
61         inputNilai(dataNilai);
62         tampilNilai(dataNilai);
63
64         int hariTertinggi = getHarikeNilaiMax(dataNilai);
65         System.out.println("Nilai Tertinggi terdapat pada Hari ke-" + hariTertinggi);
66
67         int[] hasil = getMhskeNilaiMax(dataNilai);
68         System.out.println("Nilai Tertinggi: " + hasil[2] + " pada minggu ke-" + hasil[0] +
69             " oleh Mahasiswa ke-" + hasil[1]);
70     }
71 }
72

```