

MODUL 8

METHOD



CAPAIAN PEMBELAJARAN

Mahasiswa dapat memahami, membuat dan menyelesaikan kasus dengan menggunakan method tanpa parameter



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. TextPad
2. JDK



DASAR TEORI

1. Definisi Method / Function

Fungsi digunakan untuk mempermudah didalam membuat sebuah program, terutama program yang besar dan banyak melakukan beberapa hal yang sama. Fungsi memiliki ciri-ciri sebagai berikut:

1. Memiliki nama dari fungsi tersebut.
2. Memiliki tugas spesifik tertentu.
3. Memiliki sekumpulan statement atau perintah untuk melakukan tugas tersebut.
4. Mengembalikan sebuah nilai kepada fungsi lain yang memanggil atau menggunakannya (jika perlu).

2. Fungsi di Dalam Bahasa Java

Dalam bahasa java terdapat 2 macam fungsi yaitu :

1. Fungsi yang mengembalikan/menghasilkan nilai (non void function)
2. Fungsi tidak mengembalikan/menghasilkan nilai (void function)

3. Deklarasi Function

Pada dasarnya, cara mendeklarasikan fungsi di dalam bahasa Java serupa dengan bahasa C. Berikut ini adalah cara mendeklarasikan fungsi di dalam bahasa Java:

<Modifier><Return type><function name><parameter list>

Dimana :

Modifier : sebuah *access modifier*(*public, private, protected*), yang dapat dikombinasikan dengan tipe *modifier* lain (*static, final, abstract*).

modifier static digunakan untuk mendefinisikan *member class*, sehingga metode/fungsi dapat diakses langsung dengan nama kelas (Untuk member/fungsi non static maka dipanggil melalui instant atau dengan membuat objek terlebih dahulu)

Return Type : type nilai pengembalian dari fungsi (*int,String,boolean dll*)

Function name : nama fungsi yang nantinya akan di panggil di dalam program

Parameter list : adalah **nama dan tipe variable** yang akan digunakan untuk menyimpan nilai yang dibutuhkan oleh fungsi tersebut. Jika terdapat lebih dari 1 parameter, maka parameter ditulis dipisahkan dengan koma.

Method tanpa parameter

Method tanpa parameter adalah method yang tidak memiliki parameter sama sekali.

Contoh :

```
public class Method1{
    public static void buatGaris(){
        System.out.println("=====");
    }
    public static void main(String args[]){
        buatGaris();
    }
}
```

Membuat dan memanggil fungsi

Fungsi didalam java dapat diletakkan disembarang tempat selama masih ada didalam class. Bisa sebelum program utama (public static void main) atau sesudahnya, Fungsi yang sudah dibuat dapat di panggil oleh fungsi yang lain atau dari dalam program utama.

Pemanggilan fungsi dilakukan dengan memanggil **nama fungsi** diikuti **parameternya (lihat contoh pada program)**.

a. Fungsi yang menghasilkan nilai (*Non Void Function*)

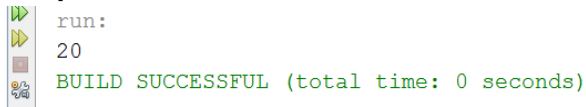
Adalah fungsi yang ketika kembali ke program utamanya disertai dengan membawa suatu nilai.

Untuk mengembalikan nilai kedalam nama fungsi menggunakan perintah **return**.

Contoh Program 1:

```
public class FungsiJava {  
    //membuat fungsi non void  
    static int jumlah2Bilangan(int a,int b){  
        return a+b;  
    }  
    public static void main(String[] args) {  
        // TODO code application logic here  
        int a=10, b=10;  
        //pemanggilan fungsi  
        System.out.println(jumlah2Bilangan(a,b));  
    }  
}
```

Output :



```
run:  
20  
BUILD SUCCESSFUL (total time: 0 seconds)
```

b. Fungsi yang tidak menghasilkan nilai (*void Function*)

Sebuah fungsi tidak harus selalu mengembalikan nilai. Tipe dari fungsi yang tidak dapat mengembalikan nilai adalah **void**. Berikut adalah contoh fungsi yang tidak mengembalikan nilai, yaitu untuk menampilkan kata "Hello" ke layar sebanyak **n** kali.

Contoh Program 2:

```

public class FungsiVoid {
    //membuat fungsi
    static void FungsiHello(int n) {
        int i;
        for (i = 0; i < n; i++) {
            System.out.printf("Hello\n");
        }
    }
    public static void main(String args[]){
        //pemanggilan fungsi
        FungsiHello(5);
    }
}

```

Output :

```

run:
Hello
Hello
Hello
Hello
Hello
Hello
BUILD SUCCESSFUL (total time: 0 seconds)

```



RAKTIK

Praktik 1. Menggunakan method tanpa parameter

```

public class Method1{
    public static void cetak(){
        System.out.println("STMIK AKAKOM");
    }
    public static void main(String args[]){
        cetak();
    }
}

```

Method diatas bersifat static dan bertipe void. Method yang bersifat static bisa langsung dipanggil dengan nama methodnya saja. Sebuah method juga bisa dipanggil lebih dari satu kali.

Praktik 2. Pemanggilan method lebih dari sekali

```
public class Method2{
    public static void cetakKalimat(){
        System.out.println("Di dalam method kalimat");
    }

    public static void main(String args[]){
        cetakKalimat();
        System.out.println("Di dalam main");
        cetakKalimat();
    }
}
```

Praktik 3. Menggunakan method dalam perulangan

```
public class Method3{
    public static void cetak(){
        System.out.println("STMIK AKAKOM");
    }
    public static void main(String args[]){
        for (int i = 0; i<10;i++)
            cetak();
    }
}
```

Praktik 4. Method tanpa parameter dengan nilai balik (return type)

```
public class Method4{
    public static int jumlah(){
        int a = 8, b = 10;
        return (a + b);
    }

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

Praktik 5. Memanggil method dengan menciptakan objek

```
public class Method5{
    public int jumlah(){
        int a = 8, b = 10;
        return (a + b);
    }

    public static void main(String args[]){
        Method5 obyek = new Method5();
    }
}
```

```
        System.out.println("Hasil pemanggilan method jumlah");  
        System.out.println(obyek.jumlah());  
    }  
}
```



LATIHAN

1. Buatlah program dengan menggunakan method untuk menampilkan data pribadi Anda seperti NIK, Nama, Jenis kelamin, Umur, Alamat ! Panggil method tersebut melalui program utama !
2. Diberikan oleh dosen pengampu



TUGAS

Diberikan oleh dosen pengampu pada saat praktikum



REFERENSI

1. Abdul Kadir, 2012, Algoritma & Pemrograman menggunakan Java, Penerbit Andi, Yogyakarta

MODUL 9 METHOD



CAPAIAN PEMBELAJARAN

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



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. TextPad
2. JDK



DASAR TEORI

Pada modul sebelumnya telah dijelaskan mengenai method tanpa parameter. Selain method tanpa parameter ada method yang menggunakan parameter.

1. Method dengan parameter (object)

Method ada yang mempunyai parameter. Ada 2 buah parameter yaitu

- 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.

Contoh :

```
public float jumlah(int a, float b) {  
    return (a+b);  
}
```

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, bayangkan bila program pada class Gambar harus diberi nama drawInterger(int i), drawString(String s), drawDouble(double d). Method *overloading* dibedakan oleh jumlah dan jenis tipe data parameternya

Contoh method overloading :

```
public class Gambar{  
    public void draw(int i){  
        .....  
    }  
    public void draw(String s){  
        .....  
    }  
    public void draw(double d){  
        .....  
    }  
    public void draw(int i, double d){  
        .....  
    }  
}
```

3. Method bawaan

(Numeric : sqrt,pow,ceil,floor,max,min dan String:StrCpy,Strcat,strlen,equals)



RAKTIK

1. Praktik 1

Tuliskan dan jalankan program berikut ini

```
1. public class Fungsi1{
2. public static int jumlah(int a){
3. return a;
4. }
5. public static void main(String args[]){
6. System.out.println("Hasil pemanggilan method ");
7. System.out.println(jumlah(5));
8. }
9. }
```

Parameter pada baris ke-2 disebut sebagai parameter formal, dan pada baris ke-7 disebut parameter aktual.

2. Praktik 2

Tuliskan program berikut :

```
1. public class Fungsi2{
2. public static int jumlah(int a){
3. return (a + a);
4. }
5. public static void main (String args[]){
6. System.out.println("Panggil method jumlah dengan parameter 5");
7. System.out.println(jumlah(5));
8. System.out.println("Panggil method jumlah dengan parameter 15");
9. System.out.println(jumlah(15));
10. }
11. }
```

3. Praktik 3.

```
1. class Mahasiswa
2. {
3. String nim;
4. String nama;
```

```

5. String prodi;
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. {
14. System.out.println("Judulbuku :"+nim);
15. System.out.println("Harga buku : "+nama);
16. System.out.println("Jumlah halaman : "+prodi);
17. }
18. public static void main(String[] args)
19. {
20. Mahasiswa mhs = new Mahasiswa();
21. mhs.setMhs("145410012","Nisa","Informatika");
22. mhs.tampil();
23. }
24. }

```

4. Praktik 4. Method overloading dengan perbedaan jumlah parameter

```

1. class DemoOverload {
2.
3.     void sum(int a, int b){
4.         System.out.println(a + b);
5.     }
6.     void sum(int a, int b, int c){
7.         System.out.println(a + b + c);
8.     }
9.
10.    public static void main(String[] args) {
11.        DemoOverload demo = new DemoOverload();
12.        demo.sum(1, 6);
13.        demo.sum(4, 2, 3);
14.    }
15. }

```

5. Praktik 5 Menggunakan method bawaan dari java

```

1. public class StringComparisonExample {
2. public static void main(String args[]) {
3. String tv = "Bravia";
4. String television = "Bravia";
5. // menggunakan equals
6. if (tv.equals(television)) {

```

```
7. System.out.println("Both tv and television contains same
   letters and equal by equals method of String");
8. }
9.
10. // menggunakan compareTo
11. if (tv.compareTo(television) == 0) {
12.   System.out.println("Both tv and television are equal using
      compareTo method of String");
13. }
14. television = "BRAVIA";
15. // menggunakan equalsIgnoreCase
16. if (tv.equalsIgnoreCase(television)) {
17.   System.out.println("tv and television are equal by
      equalsIgnoreCase method of String");
18. }
19. // menggunakan compareToIgnoreCase
20. if (tv.compareToIgnoreCase(television) == 0) {
21.   System.out.println("tv and television are same by
      compareToIgnoreCase of String");
22. }
23. String sony = "Sony";
24. String samsung = "Samsung";
25. // menggunakan compareTo
26. if (sony.compareTo(samsung) > 0) {
27.   System.out.println("Sony comes after Samsung in
      lexicographical order");
28. } else if (sony.compareTo(samsung) < 0) {
29.   System.out.println("Sony comes before Samsung in
      lexicographical order");
30. }
31. }
32. }
```



LATIHAN

1. Modifikasi praktik 3 dengan menambahkan variable ipk dengan tipe data double, serta modifikasi juga pada method dan program utamanya kemudian lihat hasil outpunya !



TUGAS

1. Buat program dengan menggunakan method bawaan java selain dari yang sudah di praktikkan
2. Diberikan oleh dosen pengampu pada saat praktikum



REFERENSI

2. Abdul Kadir, 2012, Algoritma & Pemrograman menggunakan Java, Penerbit Andi, Yogyakarta

MODUL 10

ITERASI SELEKSI DALAM METHOD



CAPAIAN PEMBELAJARAN

Mahasiswa dapat menyelesaikan kasus dengan menggabungkan konsep iterasi, seleksi dalam method serta dapat memanggil method dari class lain



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. TextPad
2. JDK



DASAR TEORI

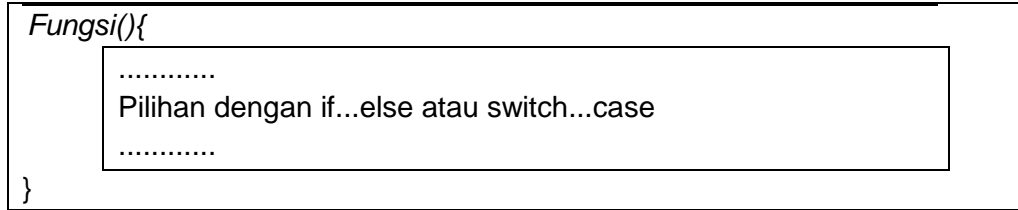
Pernyataan seleksi, perulangan dan fungsi sudah dibahas pada pertemuan sebelumnya. Pada dasarnya pemakaian ketiganya dapat digabungkan dalam suatu array, baik untuk data tunggal maupun data berupa array.

Untuk modul kali ini akan dipraktekkan beberapa program yang menggabungkan seleksi, perulangan dalam suatu fungsi.

Bentuk iterasi / perulangan dalam fungsi adalah sebagai berikut.

```
Fungsi(){  
    .....  
    Perulangan dengan for, do...while, atau while  
    .....  
}
```

Bentuk seleksi dalam fungsi adalah sebagai berikut.



RAKTIK

Praktik 1

1. Cobalah program berikut, amati hasilnya dan jelaskan dalam laporan.

```
import java.util.Scanner;  
  
public class ProyekIterasiFungsi {  
  
    public static void cetakUlang(int nUlang){  
        for (int i=0;i<nUlang;i++){  
  
            System.out.println("Cetak ke " +(i+1));  
  
        }  
  
    }  
  
    public static void main(String[] args) {  
  
        int nUlang;  
        Scanner scan = new Scanner(System.in);  
        System.out.print("Akan dicetak berapa kali :?");  
        nUlang = scan.nextInt();  
        cetakUlang(nUlang);  
  
    }  
}
```

2. Modifikasi program 1 agar fungsi dapat mencetak kalimat :yang dimasukkan oleh pengguna sebanyak jumlah yang diinginkan. Jelaskan hasil modifikasi pada laporan.
3. Cobalah program untuk menghitung jumlah dari n buah bilangan (array) berikut, amati hasilnya dan jelaskan pada laporan.

```
import java.util.Scanner;

public class ProyekHitungJumlah {

    public static int hitungJumlah(int[] x){
        int jum=0;
        for (int i=0;i<x.length;i++){
            jum += x[i];
        }
        return jum;
    }

    public static void main(String[] args) {
        int data[] = new int[10];
        int jumlah;
        Scanner scan = new Scanner(System.in);
        for (int i=0;i<10;i++){
            System.out.println("Masukkan data ke-
"+(i+1)+":");
            data[i] = scan.nextInt();
        }

        jumlah = hitungJumlah(data);

        System.out.println("Jumlah data = "+jumlah);
    }
}
```

4. Modifikasi program 3 agar fungsi dapat menghitung **rata-rata dari n buah bilangan (array)**. Jelaskan hasil modifikasi pada laporan.
5. Cobalah fungsi berikut ini yang digunakan untuk mengecek apakah suatu bilangan tersebut ganjil atau genap. Amati hasilnya dan jelaskan pada laporan.

```
import java.util.Scanner;

public class ProyekCekGenap1 {

    public static boolean cekGenap(int a){
        boolean status;
```

```

        if ((a % 2) == 0){
            status = true;
        }else {
            status = false;
        }
        return status;
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int x;
        boolean genap;
        System.out.print("Masukkan bilangan : ");

        x = scan.nextInt();

        genap = cekGenap(x);

        if (genap){
            System.out.println("Bilangan yang dicek ternyata genap");
        }else{
            System.out.println("Bilangan yang dicek ternyata ganjil");
        }
    }
}

```

6. Cobalah program berikut yang digunakan untuk mengecek bilangan tersebut ganjil atau genap dalam suatu array. Amati hasilnya dan jelaskan pada laporan.

```

import java.util.Scanner;

public class ProyekCekGanjilGenapArray {

    public static boolean[] cekGanjilGenapArray(int[] x){
        boolean hasil[] = new boolean[10];
        for (int i=0;i<x.length;i++){
            if ((x[i] % 2) == 0){
                hasil[i] = true;
            }else {
                hasil[i] = false;
            }
        }
        return hasil;
    }
}

```



```

public static void main(String[] args) {
    int data[] = new int[10];
    boolean hasilCek[] = new boolean[10];
    Scanner scan = new Scanner(System.in);
    for (int i=0;i<10;i++){
        System.out.println("Masukkan data ke-"+(i+1)+":");
        data[i] = scan.nextInt();
    }
    hasilCek = cekGanjilGenapArray(data);
    System.out.println("=====");
    System.out.println("==Hasil Pengecekan=====");
    System.out.println("=====");

    for (int i=0;i<10;i++){
        System.out.print(" "+data[i]);
        System.out.print(" "+hasilCek[i]);
        System.out.println();
    }
}

```

7. Modifikasi program 6 tersebut agar jika hasil pengecekan berupa bilangan genap maka akan tertulis genap, dan jika hasil pengecekan berupa bilangan ganjil maka akan tertulis ganjil. Jelaskan hasil modifikasi pada laporan.



LATIHAN

1. Cobalah program berikut yang digunakan untuk mengkonversi masukan berupa angka menjadi huruf. Sedangkan ketentuan konversi adalah sebagai berikut :

Nilai 81 – 100 → A
 Nilai 61 – 80 → B
 Nilai 41 – 60 → C
 Nilai 21 – 40 → D
 Nilai < =20 → E

Jelaskan hasilnya

```

import java.util.Scanner;

public class KonversiNilai {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        double[] data = new double[10];
        char [] keluar = new char[10];
        for (int i=0;i<10;i++){
            System.out.print("data ke-"+(i+1)+":");
            data[i] = scan.nextDouble();
        }
        System.out.println("=====");
        System.out.println("|    Hasil Konversi    |");
        System.out.println("=====");
        keluar = konversiNilai(data);
        //tampil hasil
        for (int i=0;i<10;i++){
            System.out.println(data[i]+"    "+keluar[i]);
        }
    }

    public static char[] konversiNilai(double[] x) {
        int n = x.length;

        char hasil[] = new char[n];

        for (int i = 0; i < n; i++) {
            if ((x[i] > 80) && (x[i] <= 100)) {
                hasil[i] = 'A';
            } else if ((x[i] > 60) && (x[i] <= 80)) {
                hasil[i] = 'B';
            } else if ((x[i] > 40) && (x[i] <= 60)) {
                hasil[i] = 'C';
            } else if ((x[i] > 20) && (x[i] <= 40)) {
                hasil[i] = 'D';
            } else if (x[i] <= 20) {
                hasil[i] = 'E';
            }
        }
        return hasil;
    }
}

```



TUGAS

3. Modifikasi program latihan tersebut dengan menggunakan while dan switch....case. Jelaskan hasil modifikasi pada laporan.
4. Diberikan oleh dosen pengampu pada saat praktikum



REFERENSI

1. Abdul Kadir, 2012, Algoritma & Pemrograman menggunakan Java, Penerbit Andi, Yogyakarta