

Academic Year 2021/2022

# Pemrograman Berorientasi Objek

Rofilde Hasudungan

Lecturer

Department of Informatic

Universitas Muhammadiyah Kalimantan Timur

September 24, 2021

# Konversi Data

Casting dan Parsing merupakan metode untuk mengkonversi/merubah suatu tipe data. Perbedaan keduanya ialah casting mengkonversi nilai pada tipe data yang mirip seperti double ke integer atau sebaliknya, float ke integer atau sebaliknya, double ke float atau sebaliknya dan lainnya. Sedangkan parsing ialah mengkonversi nilai pada tipe data yang berbeda seperti integer ke string atau sebaliknya, double ke string dan sebaliknya, float ke string dan sebaliknya.

# Contoh: Menghitung Luas Persegi

```
1 import javax.swing.JOptionPane;
2
3 class HitungPersegi {
4     public static void main(String[] args) {
5         double panjang, lebar, luas;
6         String p = JOptionPane.showInputDialog("Masukan panjang");
7         String l = JOptionPane.showInputDialog("Masukan tinggi persegi");
8         // Konversi dari string ke double
9         panjang = Double.parseDouble(p);
10        lebar = Double.parseDouble(l);
11        luas = panjang * lebar;
12        JOptionPane.showMessageDialog(null, "Luas Persegi = " + luas);
13    }
14 }
```

# Luaran

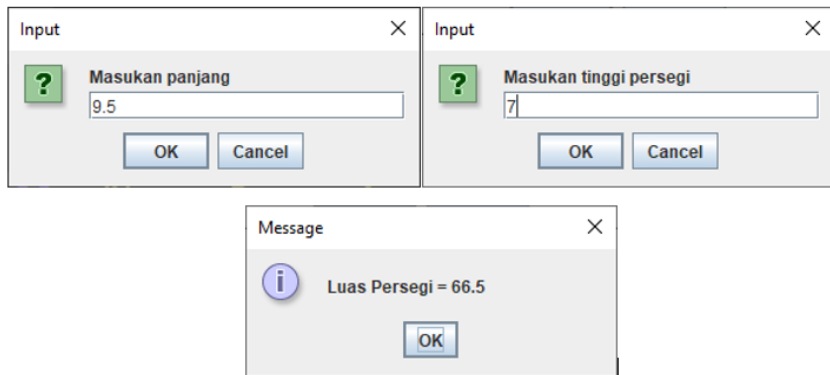


Figure: Luaran Gui

# Contoh: Pembagian bilangan bulat

```
1 import javax.swing.JOptionPane;
2
3 public class PembagianBilanganBulat {
4     public static void main(String[] args) {
5         int a, b;
6         double hasil_bagi;
7         // Parsing dari string ke integer
8         a = Integer.parseInt(JOptionPane.showInputDialog("Masukan a"));
9         b = Integer.parseInt(JOptionPane.showInputDialog("Masukan b"));
10        // Casting dari integer ke double
11        hasil_bagi = (double) a / b;
12
13        JOptionPane.showMessageDialog(null, a+"/"+b + " = "+hasil_bagi);
14    }
15 }
```

# Luaran Hasil Pembagian

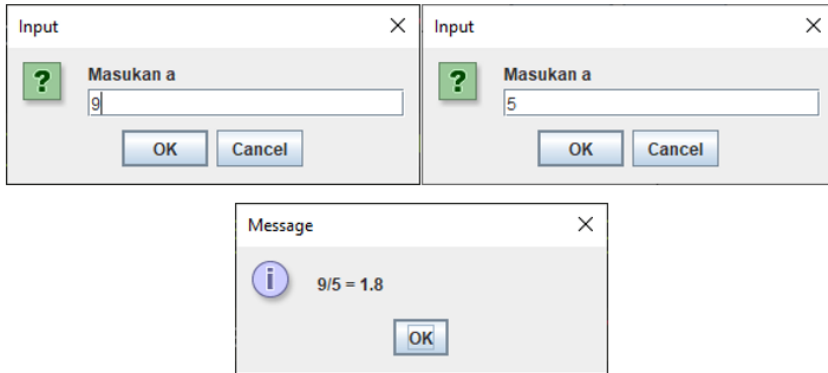


Figure: Luaran Hasil Pembagian

# Selection

- ① If (One-way selection)
- ② If..else (Alternative Selection)
- ③ If..if..else (Multiple Alternative Selection)
- ④ Switch..case

# Contoh: Menebak Hasil Penjumlahan

## LISTING 3.1 AdditionQuiz.java

```
1 import java.util.Scanner;
2
3 public class AdditionQuiz {
4     public static void main(String[] args) {
5         int number1 = (int)(System.currentTimeMillis() % 10);
6         int number2 = (int)(System.currentTimeMillis() * 7 % 10);
7
8         // Create a Scanner
9         Scanner input = new Scanner(System.in);
10
11         System.out.print(
12             "What is " + number1 + " + " + number2 + "? ");
13
14         int answer = input.nextInt();
15
16         System.out.println(
17             number1 + " + " + number2 + " = " + answer + " is " +
18             (number1 + number2 == answer));
19     }
20 }
```

generate number1  
generate number2

show question

display result



Seleksi yang pertama ialah **if** atau disebut juga dengan *one-way condition*. Kondisi ini memiliki satu kondisi dan tidak memiliki alternatifnya. Berikut ialah format dari *one-way condition*

---

```
1 if(boolean-expression) {  
2     // block code  
3 }
```

---

**boolean-expression** ialah ekspresi dari operasi perbandingan dan/atau logika atau boolean.

# Contoh One-way condition

```
1 import java.util.Scanner;
2
3 public class ExOneWayCondition {
4     public static void main(String[] args) {
5         System.out.println("Ayo Pergi Kamping");
6         System.out.println("Ayu Bawa Pop Mie");
7         Scanner input = new Scanner(System.in);
8         System.out.print("Apakah Yuli ikut [1 - untuk ikut]? ");
9         int yuli_ikut = input.nextInt();
10
11         if (yuli_ikut == 1) {
12             System.out.println("Yuli bawa tenda");
13         }
14
15         System.out.println("Ayo Berangkat");
16     }
17 }
```

# Alternative Condition

Seleksi yang kedua ialah **if..else** atau disebut juga dengan *alternative condition*. Kondisi ini memiliki satu kondisi/lebih dan kondisi alternatifnya, sehingga jika kondisi yang disyaratkan tidak memenuhi, maka kondisi alternatif yang akan dijalankan. Perhatikan format berikut

---

```
1 if(boolean-expression) {  
2     // block code  
3 } else {  
4     // Alternatif code-block  
5 }
```

---

**else** ialah alternatif kondisi.

# Contoh One-way condition

```
1 import java.util.Scanner;
2 public class ExAlternativeCondition {
3     public static void main(String[] args) {
4         System.out.println("Ayo Pergi Kamping");
5         System.out.println("Ayu Bawa Pop Mie");
6         Scanner input = new Scanner(System.in);
7         System.out.print("Apakah Yuli ikut [1 - untuk ikut]? ");
8         int yuli_ikut = input.nextInt();
9
10        if (yuli_ikut == 1) {
11            System.out.println("Yuli bawa tenda");
12        } else {
13            System.out.println("Kita harus cari penggantinya");
14            System.out.println("Ajak Anto");
15        }
16        System.out.println("Ayo Berangkat");
17    }
18 }
```

# Multiple dan default Condition dan

Seleksi yang ketiga ialah **if..else if** yakni memiliki lebih dari satu kondisi dan default kondisi, sehingga jika tidak ada kondisi memenuhi, maka kondisi default yang akan dijalankan. Perhatikan format berikut

---

```
1  if(boolean-expression) {  
2      // block code  
3  } else if(boolean-expression){  
4      // Alternatif code-block  
5  } else if(boolean-expression){  
6      // Alternatif code-block  
7  } ... alternatif yang lain...  
8
```

---

**else if** ialah alternatif kondisi.

```
1 import java.util.Scanner;
2 public class ExMultipleCondition {
3     public static void main(String[] args) {
4         double nilai;
5         Scanner input = new Scanner(System.in);
6         System.out.println("Berapakah nilai Anda? ");
7         nilai = input.nextDouble();
8         if(nilai >= 80 && nilai <= 100) {
9             System.out.println("A");
10        } else if(nilai >= 70 && nilai < 80) {
11            System.out.println("B");
12        } else if(nilai >= 60 && nilai < 70) {
13            System.out.println("C");
14        } else if(nilai >= 50 && nilai < 60) {
15            System.out.println("D");
16        } else if(nilai >= 0 && nilai < 50) {
17            System.out.println("E");
18        }
19    }
20 }
```

# Multiple Condition + Alternative

Seleksi yang kedua ialah **if..else** atau disebut juga dengan *alternative condition*. Kondisi ini memiliki satu kondisi/lebih dan kondisi alternatifnya, sehingga jika kondisi yang disyaratkan tidak memenuhi, maka kondisi alternatif yang akan dijalankan. Perhatikan format berikut

---

```
1  if(boolean-expression) {  
2      // block code  
3  } else if(boolean-expression){  
4      // Alternatif code-block  
5  } else if(boolean-expression){  
6      // Alternatif code-block  
7  } ... alternatif yang lain...  
8
```

---

**else if** ialah alternatif kondisi terakhir (default).

# Contoh

```
1  import java.util.Scanner;
2  public class ExMultipleCondition {
3      public static void main(String[] args) {
4          double nilai;
5          Scanner input = new Scanner(System.in);
6          System.out.println("Berapakah nilai Anda? ");
7          nilai = input.nextDouble();
8          if(nilai >= 80 && nilai <= 100) {
9              System.out.println("A");
10         } else if(nilai >= 70 && nilai < 80) {
11             System.out.println("B");
12         } else if(nilai >= 60 && nilai < 70) {
13             System.out.println("C");
14         } else if(nilai >= 50 && nilai < 60) {
15             System.out.println("D");
16         } else if(nilai >= 0 && nilai < 50) {
17             System.out.println("E");
18         } else {
19             System.out.println("Nilai tidak valid. Harus diantara 0-100");
```



```
20     }  
21     }  
22 }  
23
```

---

# Switch..Case

Seleksi kondisi switch..case berbeda dengan if, dimana jika if menggunakan ekspresi yang menghasilkan boolean, sehingga bisa menggunakan operator perbandingan dan kombinasi dengan operator logika, maka switch..case lebih sederhana, karena seleksi ini mengecek ketersediaan case yang ada.

# Format Switch..case

```
1 TipeData VARIABLE;  
2 switch(VARIABLE) {  
3     case CASE_1:  
4         // BLOCK-CODE  
5         break;  
6     case CASE_2:  
7         // BLOCK-CODE  
8         break;  
9     case CASE_3:  
10        // BLOCK-CODE  
11        break;  
12    case ...:  
13        // BLOCK-CODE  
14        break;  
15    default:  
16        // BLOCK-CODE  
17        break;  
18 }
```

# Contoh Implementasi Switch...case

```
1 import java.util.Scanner;
2
3 public class ExSwitchCase {
4     public static void main(String[] args) {
5         int menu;
6         Scanner input = new Scanner(System.in);
7         System.out.println("-----Pilih Menu-----");
8         System.out.println("1. Nasi Goreng");
9         System.out.println("2. Nasi Ayam");
10        System.out.println("3. Mie Rebus");
11        System.out.println("x. Keluar");
12        System.out.print("Menu pilihan Anda? ");
13        menu = input.nextInt();
14        switch (menu) {
15            case 1:
16                System.out.println("Nasi Goreng Sedang Disiapkan.!");
17                break;
18            case 2:
19                System.out.println("Nasi Ayam Sedang Disiapkan.!");
```

```
20         System.out.println("Ayam Sedang Dipotong.!");
21         break;
22     case 3:
23         System.out.println("Mie telah direbus!");
24         break;
25     default:
26         System.out.println("Terimakasih!");
27         break;
28     }
29 }
30 }
```

---

# Kondisi Bersarang

Kondisi bersarang ialah kondisi didalam kondisi. Hal ini memungkinkan karena kebutuhan dalam pengembangan aplikasi.

# Contoh Kondisi Bersarang

```
1 import java.util.Scanner;
2 public class ExNestedCondition {
3     public static void main(String[] args) {
4         double nilai;
5         Scanner input = new Scanner(System.in);
6         System.out.print("Berapakah nilai Anda? ");
7         nilai = input.nextDouble();
8         if(nilai >= 60) {
9             System.out.print("Selamat Anda Lulus dengan Nilai ");
10            if(nilai >= 80 && nilai <= 100) {
11                System.out.println("A");
12            } else if(nilai >= 70 && nilai < 80) {
13                System.out.println("B");
14            } else {
15                System.out.println("C");
16            }
17        } else {
18            System.out.print("Anda Gagal. Nilai Anda ");
19            if(nilai >= 50) {
```

```
20         System.out.println("D");
21     } else {
22         System.out.println("E");
23     }
24 }
25
26 }
27 }
```

---



# Looping

- ① for loop
- ② while loop
- ③ Nested Looping
- ④ Stop Looping
- ⑤ Jump Looping

# For Loop

Perulangan for merupakan perulangan yang sederhana. Perulangan ini melakukan iterasi seperti dari 1 hingga 100. Berikut ialah formatnya:

---

```
1 for (initial-action; loop-continuation-condition; action-after-each-  
    iteration) {  
2     // Loop body;  
3 }
```

---

- ❶ initial-action ialah awal perulangan akan dilaksanakan
- ❷ loop-continuation-condition ialah kondisi iterasi tetap berlangsung
- ❸ action-after-each-iteration ialah umumnya step/langkah pada iterasi

# Contoh: For Loop

---

```
1 public class ExForLoop {  
2     public static void main(String[] args) {  
3         for(int i=0; i <= 14; i++) {  
4             System.out.println("Hello ke-" + i);  
5         }  
6     }  
7 }
```

---

# Contoh for loop 2

```
1 import java.util.Scanner;
2
3 public class ExForLoop {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6         System.out.print("Masukan jumlah angka = ");
7         int n = input.nextInt();
8         for(int i=1; i <= n; i++) {
9             System.out.print("Masukan angka ke-" + i + " = ");
10            System.out.println(input.nextInt());
11            System.out.println("---");
12        }
13    }
14 }
```

# While Loop

While loop merupakan variant style dari perulangan pada Java. Perulangan ini memiliki format sebagai berikut:

---

```
1 while(loop-continuation-condition) {  
2     // loop body  
3 }
```

---

- ❶ loop-continuation-condition ialah kondisi dimana iterasi berlangsung
- ❷ loop body ialah block-code yang akan diulang sesuai dengan kondisi.

# Contoh while loop

---

```
1 public class ExWhileLoop {
2     public static void main(String[] args) {
3         int count = 0;
4         while (count < 10) {
5             System.out.println("Welcome to Java!");
6             count++;
7         }
8     }
9 }
```

---

# Tebak Angka

---

```
1 import java.util.Scanner;
2
3 public class TebakAngka {
4     public static void main(String[] args) {
5         boolean looping = true;
6         int angka;
7         Scanner input = new Scanner(System.in);
8         System.out.println("Tebak Angka 0-10");
9         while (looping) {
10             System.out.print("Masukan Angka = ");
11             angka = input.nextInt();
12             if (angka == 7) {
13                 looping = false;
14                 System.out.println("Hore... Tebakanmu Benar.!!");
15             } else {
16                 System.out.println("Coba lagi sampai berhasil y..");
17             }
18         }
19     }
```





# Nested Loop

Perulangan Bersarang (Nested loop) ialah perulangan didalam perulangan.

---

```
1 public class ExNestedLoop {  
2     public static void main(String[] args) {  
3         for (int i = 1; i <= 10; i++) {  
4             for (int j = 1; j <= 10; j++) {  
5                 System.out.print((i*j) + " ");  
6             }  
7             System.out.println();  
8         }  
9     }  
10 }
```

---

# Stop Perulangan

Adakalanya kita ingin menghentikan perulangan karena misalkan suatu kondisi telah tercapai. Untuk menghentikan perulangan dapat menggunakan `break`.

# Contoh Stop Perulangan

```
1 import java.util.Scanner;
2
3 public class ExBreakLooping {
4     public static void main(String[] args) {
5         int jumlah=0;
6         Scanner input = new Scanner(System.in);
7         for(long i = 1; i <= 1000; i++) {
8             System.out.print(i + " ");
9             if(i > 1 && i % 10 == 0) {
10                 System.out.print("\nMau Ditampilkan lagi? [1/0]");
11                 if (input.nextInt() == 0) {
12                     break;
13                 }
14             }
15         }
16     }
17 }
```

# Melompati Perulangan

Selain memberhentikan perulangan, kita juga dapat melompati perulangan. Bedanya mungkin terdapat kode yang dilewati ketika perulangan.

---

```
1 import java.util.Scanner;
2 public class ExJumpLooping {
3     public static void main(String[] args) {
4         int lompat;
5         Scanner input = new Scanner(System.in);
6         System.out.print("Masukan bilangan yang dilompati? [1-10]");
7         lompat = input.nextInt();
8         for(int i = 1; i <= 10; i++) {
9             if(i == lompat) {
10                 continue;
11             }
12             System.out.print(i + " ");
13         }
14     }
15 }
```