Laporan Praktikum

Algoritma Pemrograman



Disusun Oleh :

**Ahda Rindang Al-Amin (2311531003)**

Dosen Pengampu : Dr. Wahyudi, S.T, M.T

Asisten Labor : Rahmad Dwirizki Olders

Departemen Informatika

Fakultas Teknologi Informasi

Universitas Andalas

Tahun 2023

**Ternary Operator, continue dan switch Statement**

1. **Tujuan Praktikum**
2. Memahami penggunaan **?** operator (ternary) pada bahasa pemrograman Java.
3. Memahami penggunaan **continue** pada bahasa pemrograman Java.
4. Memahami penggunaan **switch** pada bahasa pemrograman Java.
5. **Pendahuluan**
6. **The ? Operator**

One of Java’s most fascinating operators is the ?. The ? operator is often used to replace if-else statements of this general form:

if (*condition*)

var = *expression1*;

else

var = *expression2*;

Here, the value assigned to *var* depends upon the outcome of the condition controlling the **if**.

The **?** is called a *ternary operator* because it requires three operands. It takes the general form

*Exp1* ? *Exp2* : *Exp3*;

where *Exp1* is a **boolean** expression, and *Exp2* and *Exp3* are expressions of any type other than **void**. The type of *Exp2* and *Exp3* must be the same (or compatible), though. Notice the use and placement of the colon.

The value of a **?** expression is determined like this: *Exp1* is evaluated. If it is true, then *Exp2* is evaluated and becomes the value of the entire **?** expression. If *Exp1* is false, then *Exp3* is evaluated and its value becomes the value of the expression.

1. **Use continue**

It is possible to force an early iteration of a loop, bypassing the loop’s normal control structure. This is accomplished using **continue**. The **continue** statement forces the next iteration of the loop to take place, skipping any code between itself and the conditional expression that controls the loop. Thus, continue is essentially the complement of **break**.

In **while** and **do-while** loops, a **continue** statement will cause control to go directly to the conditional expression and then continue the looping process. In the case of the **for**, the iteration expression of the loop is evaluated, then the conditional expression is executed, and then the loop continues.

Good uses of **continue** are rare. One reason is that Java provides a rich set of loop statements that fit most applications. However, for those special circumstances in which early iteration is needed, the **continue** statement provides a structured way to accomplish it.

1. **The switch Statement**

One of Java’s selection statements is the **switch**. The **switch** provides for a multiway branch. Thus, it enables a program to select among several alternatives. Although a series of nested **if** statements can perform multiway tests, for many situations the **switch** is a more efficient approach. It works like this: the value of an expression is successively tested against a list of constants. When a match is found, the statement sequence associated with that match is executed. The general form of the **switch** statement is

switch(*expression*) {

case *constant1*:

*statement sequence*

break;

case *constant2:*

*statement sequence*

break;

case *constant3:*

*statement sequence*

break;

.

.

default:

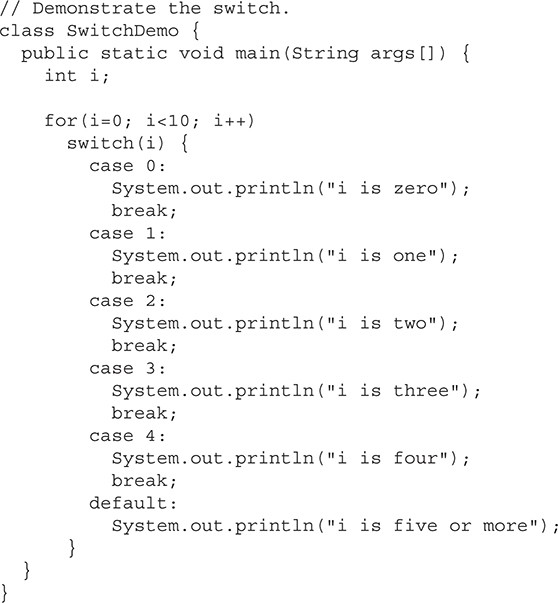
*statement sequence*

}

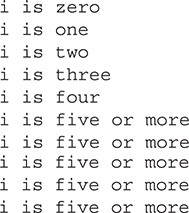
Each value specified in the **case** statements must be a unique constant expression (such as a literal value). Duplicate **case** values are not allowed. The type of each value must be compatible with the type of *expression*.

The **default** statement sequence is executed if no **case** constant matches the expression. The **default** is optional; if it is not present, no action takes place if all matches fail. When a match is found, the statements associated with that **case** are executed until the **break** is encountered or, in the case of **default** or the last **case**, until the end of the **switch** is reached.

The following program demonstrates the **switch**:



The output produced by this program is shown here:



As you can see, each time through the loop, the statements associated with the **case** constant that matches **i** are executed. All others are bypassed. When **i** is five or greater, no **case** statements match, so the **default** statement is executed.

Technically, the **break** statement is optional, although most applications of the **switch** will use it. When encountered within the statement sequence of a **case**, the **break** statement causes program flow to exit from the entire **switch** statement and resume at the next statement outside the **switch**.

However, if a **break** statement does not end the statement sequence associated with a **case**, then all the statements at and following the matching **case** will be executed until a **break** (or the end of the **switch**) is encountered.

1. **Metode Praktikum**
2. **Ternary**

Program berikut menggunakan ternary pada string hasil. Jika IPK > 2.75 bernilai true, maka statement setelah tanda ? (sebelum : ) dijalankan. Jika bernilai false maka statement setelah : dijalankan.

public static void main(String[] args) {

double IPK;

String stat1="Anda Lulus Sangat Memuaskan";

Scanner input=new Scanner(System.***in***);

System.***out***.print("Input IPK Anda = ");

IPK=input.nextDouble();

input.close();

String hasil = (IPK > 2.75)?stat1:"Anda Lulus Memuaskan";

System.***out***.println(hasil);

}

Berikut contoh output dari program tersebut :

1. IPK > 2.75 bernilai false

Input IPK Anda = 2

Anda Lulus Memuaskan

1. IPK > 2.75 bernilai true

Input IPK Anda = 3.5

Anda Lulus Sangat Memuaskan

1. **Continue Statement**

Program berikut menggunakan continue statement dalam kurung kurawal if. Kondisi if dimana i==5 OR i==8, jika bernilai true, maka continue akan dijalankan. Yang berarti program dipaksa untuk lanjut sehingga 5 dan 8 tidak akan muncul pada output.

public static void main(String[] args) {

int i=1;

while (i<=10) {

if ((i==5)||(i==8)) {

i++;

continue;

}

System.***out***.println(i);

i++;

}

}

Berikut output dari program tersebut:

1

2

3

4

6

7

9

10

1. **Switch statement**

Program berikut menggunakan switch statement. Untuk nilai n yang diinputkan, akan sesuai dengan case yang dijalankan. Misal nilai n yang diinput adalah 1, maka case 1 dijalankan, dan seterusnya. Untuk nilai n yang tidak memiliki case, atau pada program ini nilai n lebih dari 3, maka default akan dijalankan.

public static void main(String[] args) {

Scanner in = new Scanner(System.***in***);

System.***out***.print("Nama: ");

String nama=in.nextLine();

System.***out***.print("Umur: ");

int umur=in.nextInt();

System.***out***.print("Inputkan Pilihan Anda (1-3): ");

int n = in.nextInt();

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

System.***out***.println("Umur: " + umur);

System.***out***.println("Pilihan: " + n);

switch(n) {

case 1:

System.***out***.println("Anda memilih Anies");

break;

case 2:

System.***out***.println("Anda memilih Prabowo");

break;

case 3:

System.***out***.println("Anda memilih Ganjar");

break;

default:

System.***out***.println("Pilihan anda bukan 1 sampai 3");

}

System.***out***.println("Terimakasih sudah memilih");

in.close();

}

Berikut contoh output dari progam tersebut:

Nama: Ahda

Umur: 18

Inputkan Pilihan Anda (1-3): 3

Nama: Ahda

Umur: 18

Pilihan: 3

Anda memilih Ganjar

Terimakasih sudah memilih

Dapat dilihat bahwa nilai n yang diinput adalah 3, sehingga case 3 dijalankan.

1. **Kesimpulan Praktikum**

Pada bahasa pemrograman Java, terdapat **operator ternary**, **continue**, dan **switch**. **Operator ternary** adalah bentuk lain **if-else** yang lebih sederhana. Pernyataan **continue** digunakan untuk memaksakan iterasi awal dari sebuah loop, melewati struktur kontrol normal dari loop tersebut. Sedangkan, **switch** menyediakan cabang multiway. Dengan demikian, ini memungkinkan program untuk memilih di antara beberapa alternatif. Meskipun serangkaian pernyataan **nested if** dapat melakukan pengujian multiway, dalam banyak situasi **switch** merupakan pendekatan yang lebih efisien.