

PRAKTIKUM 3

PERCABANGAN DAN PERULANGAN

A. TUJUAN PEMBELAJARAN

1. Memahami logika percabangan
2. Memahami logika perulangan
3. Memakai percabangan dan perulangan yang tepat
4. Memahami pemberian kondisi untuk perulangan

B. DASAR TEORI

Percabangan di dalam Java terdapat 2 macam, yaitu dengan memakai if dan switch. Percabangan if dipakai jika kita menginginkan suatu pernyataan itu dilakukan dengan syarat tertentu yang bernilai benar. Sintaks dari if adalah sebagai berikut:

```
if (ekspresi_boolean) {  
    Pernyataan1;  
}
```

Pernyataan1 akan dikerjakan kalau ekspresi_boolean bernilai true. Percabangan if-else dipakai untuk mengeksekusi salah satu dari 2 pernyataan dari syarat tertentu yang pada pada if yang dapat bernilai benar atau salah. Sintaks dari if-else adalah sebagai berikut:

```
if (ekspresi_boolean) {  
    Pernyataan1;  
} else {  
    Pernyataan2;  
}
```

Pernyataan1 akan dikerjakan kalau ekspresi_boolean bernilai true. Kalau ekspresi_boolean bernilai false, maka Pernyataan2 akan dikerjakan. Percabangan else-if dipakai untuk memberikan kondisi tertentu pada bagian else. Sintaks dari else-if adalah sebagai berikut:

```
if (ekspresi_boolean1) {
    Pernyataan1;
} else if (ekspresi_boolean2) {
    Pernyataan2;
}
```

Ketika ekspresi_boolean bernilai false, maka alur program akan menuju ke bagian else. Selanjutnya Pernyataan2 diatas akan dikerjakan kalau ekspresi_boolean2 bernilai true.

Percabangan switch dipakai pada saat kita ingin memberikan kondisi dengan beberapa syarat yang identik yang masing-masing mempunyai pernyataan yang berbeda-beda. Pada Java, nilai yang dilewatkan pada switch harus bertipe int, short, byte atau char. Sintaks dari switch adalah sebagai berikut:

```
switch (ekspresi) {
    case nilai1: Pernyataan1;
        break;
    case nilai2: Pernyataan2;
        break;
    default: Pernyataan3;
}
```

Ketika ekspresi bernilai nilai1, maka alur program akan mengeksekusi Pernyataan1. Selanjutnya break menyebabkan alur program keluar dari daerah switch. Kalau ekspresi bernilai nilai2, maka alur program akan mengeksekusi Pernyataan2. Apabila ekspresi mempunyai nilai yang tidak sama dengan nilai1 dan nilai2, maka alur program akan menuju ke bagian default dan kemudian mengeksekusi Pernyataan3. Tipe data yang

diperbolehkan untuk ekspresi adalah byte, short, int dan char saja. Tetapi untuk Java diatas versi 7 ditambahkan tipe data String.

Perulangan di dalam Java terdapat 3 macam, yaitu for, while dan do-while. Perulangan for dipakai pada saat kita melakukan perulangan dengan jumlah yang sudah diketahui pasti. Sintaks dari for adalah sebagai berikut:

```
for (inisialisasi; kondisi; perubah) {  
    Pernyataan;  
}
```

Perulangan while dipakai pada saat kita melakukan perulangan dengan jumlah yang belum diketahui pasti. Pernyataan pada while akan dikerjakan setelah pengecekan kondisi pada while bernilai true. Sintaks dari while adalah sebagai berikut:

```
while (kondisi) {  
    Pernyataan;  
}
```

Perulangan do-while dipakai pada saat kita melakukan perulangan dengan jumlah yang belum diketahui pasti. Pernyataan pada do akan dikerjakan terlebih dahulu, baru setelah itu dilakukan pengecekan kondisi pada while. Sintaks dari do-while adalah sebagai berikut:

```
do {  
    Pernyataan;  
} while (kondisi);
```

Kita dapat memberikan kondisi tertentu pada saat terjadi perulangan. Kondisi yang mungkin terjadi pada perulangan terdapat 2 macam, yaitu break dan continue. Break menyebabkan suatu kondisi untuk keluar dari perulangan. Sedangkan continue menyebabkan suatu kondisi untuk melanjutkan ke tahapan selanjutnya pada perulangan.

C. TUGAS PENDAHULUAN

1. Sebutkan dan jelaskan berbagai macam sintaks percabangan yang digunakan di Java!
2. Sebutkan dan jelaskan berbagai macam sintaks perulangan yang digunakan di Java!

D. PERCOBAAN

Percobaan 1 : Percabangan menggunakan if, if-else dan else-if

```
class IfElseName {
    public static void main (String args[]) {
        char firstInitial = 'a';
        System.out.println("Masukkan huruf awal nama anda: ");
        try {
            firstInitial = (char)System.in.read();
        } catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }

        if (firstInitial == 'a')
            System.out.println("Nama anda pasti Asep!");
        else if (firstInitial == 'b')
            System.out.println("Nama anda pasti Brodin!");
        else if (firstInitial == 'c')
            System.out.println("Nama anda pasti Cecep!");
        else
            System.out.println("Nama anda tidak terkenal!");
    }
}
```

Percobaan 2 : Percabangan menggunakan switch

```
class SwitchName {
    public static void main (String args[]) {
        char firstInitial = 'a';
        System.out.println("Masukkan huruf awal nama anda:");
        try {
            firstInitial = (char)System.in.read();
        } catch (Exception e) {
```

```

        System.out.println("Error: " + e.toString());
    }

    switch (firstInitial) {
        case 'a':
            System.out.println("Nama anda pasti Asep!");
        case 'b':
            System.out.println("Nama anda pasti Brodin!");
        case 'c':
            System.out.println("Nama anda pasti Cecep!");
        default:
            System.out.println("Nama anda tidak terkenal!");
    }
}
}

```

Percobaan 3 : Percabangan menggunakan switch dengan break

```

class SwitchNameBreak {
    public static void main (String args[]) {
        char firstInitial = 'a';
        System.out.println("Masukkan huruf awal nama anda:");
        try {
            firstInitial = (char)System.in.read();
        } catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }

        switch (firstInitial) {
            case 'a':
                System.out.println("Nama anda pasti Asep!");
                break;
            case 'b':
                System.out.println("Nama anda pasti Brodin!");
                break;
            case 'c':
                System.out.println("Nama anda pasti Cecep!");
                break;

```

```

        default:
            System.out.println("Nama anda tidak terkenal!");
    }
}
}

```

Percobaan 4 : Perulangan menggunakan for

```

class ForCount {
    public static void main (String args[]) {
        int count=1;
        for (int i=0; i<9; i++) {
            for (int j=0; j<i+1; j++) {
                System.out.print(count);
            }
            count++;
            System.out.println();
        }
    }
}

```

Percobaan 5 : Perulangan menggunakan while

```

class WhileCoun
t {
    public static void main (String args[]) {
        int count=1;
        int i=0;
        while (i<9) {
            int j=0;
            while (j<i+1) {
                System.out.print(count);
                j++;
            }
            count++;
            System.out.println();
            i++;
        }
    }
}

```

```
    }  
  }  
}
```

Percobaan 6 : Perulangan dengan break

```
class BreakLoop {  
    public static void main (String args[]) {  
        int i = 0;  
        do {  
            System.out.println("Iterasi ke " + i);  
            i++;  
            if (i > 10) break;  
        }  
        while (true);  
    }  
}
```

Percobaan 7 : Perulangan dengan continue

```
public class ContinueLoop {  
    public static void main(String args[]) {  
        int a, b;  
        for(a=0;a<2;a++)  
            for(b=0;b<3;b++) {  
                if (b==1) continue;  
                System.out.println("a=" + a + " ; b=" + b);  
            }  
    }  
}
```

Percobaan 8 : Pemakaian label pada kondisi break

```
public class BreakLabel {  
    public static void main(String args[]) {  
        int a, b;
```

```

    Mulai:
    for (a=0;a<2;a++)
        for (b=0;b<3;b++) {
            if (b==1) break Mulai;
            System.out.println("a=" + a + " ; b=" + b);
        }
    }
}

```

Percobaan 9 : Pemakaian label pada kondisi continue

```

public class ContinueLabel {
    public static void main(String args[]) {
        int a, b;
        Mulai:
        for (a=0;a<2;a++)
            for (b=0;b<3;b++) {
                if (b==1) continue Mulai;
                System.out.println("a=" + a + " ; b=" + b);
            }
    }
}

```

E. LATIHAN

Latihan 2 : Menampilkan bilangan faktorial

Bilangan bulat faktorial n, ditulis dengan n! adalah dihasilkan dari mengalikan dari 1 sampai dengan n. Contohnya $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$. Buatlah program untuk menampilkan tabel hasil faktorial dari suatu bilangan yang diinputkan (tampilan bilangan rata kanan)

Contoh tampilan :

Masukkan bilangan faktorial: 7

n	n!

1	1

2	2
3	6
4	24
5	120
6	720
7	5040

Latihan 4 : Menampilkan deret bilangan genap

Buatlah program untuk menampilkan deret bilangan genap dari 2 sampai dengan 20 kecuali kelipatan 6.

Contoh tampilan :

2 4 8 10 14 16 20

F. TUGAS

Tugas 1 : Deret Fibonacci

Buatlah program untuk menampilkan deret Fibonacci

Contoh tampilan :

Masukkan berapa deret Fibonacci? 8

8 deret Fibonacci = 1 1 2 3 5 8 13 21

Tugas 2 : Menentukan tahun kabisat

Buatlah program untuk menentukan suatu tahun kabisat atau bukan dimana tahun dibatasi mulai dari tahun 1900 sampai dengan tahun 2005.

Contoh tampilan:

Masukkan tahun (1900-2005) : 1923

1923 bukan tahun kabisat

Masukkan tahun (1900-2005) : 1898

Maaf, tahun input dibawah 1900

Masukkan tahun (1900-2005) : 1996

1996 adalah tahun kabisat

Masukkan tahun (1900-2005) : 2008

Maaf, tahun input diatas 2005

LAPORAN RESMI

Kumpulkan hasil percobaan di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat.