

## **BAB 2 SELECTION & REPETITION**

### **2.1 Tujuan Pembelajaran**

1. Mahasiswa diharapkan untuk memahami struktur dalam pemilihan/ selection dan struktur pengulangan/ repetition.
2. Mahasiswa diharapkan memahami bentuk umum dari setiap struktur pemilihan/ selection dan struktur pengulangan/ repetition.

### **2.2 Dasar Teori**

#### **2.2.1 Struktur Pemilihan / Selection**

Struktur pemilihan atau percabangan adalah suatu cara pemecahan masalah dengan memilih statement atau perintah tertentu berdasarkan berdasarkan suatu kondisi. Statement atau perintah tertentu akan dijalankan apabila memenuhi ketentuan yang telah didefinisikan sebelumnya. Pernyataan percabangan memungkinkan suatu statement dieksekusi, hanya jika suatu kondisi terpenuhi atau tidak terpenuhi.

Untuk menentukan suatu kondisi tertentu apakah benar atau salah, ya atau tidak; diperlukan pemahaman tentang beberapa struktur yang mendukung pengambilan keputusan. Dalam hal pemrograman, harus mengenal operator yang dipakai dalam pengambilan keputusan seperti operator relasi dan logika.

##### **2.2.1.1 If**

Statement If memungkinkan sebuah program untuk dapat memilih beberapa operasi untuk dieksekusi berdasarkan beberapa pilihan.

- **Bentuk Umum If**

```
if (kondisi){  
    Pernyataan-1;  
    ...  
    Pernyataan-n;  
}
```

Jika kondisi benar maka pernyataan-1 sampai dengan pernyataan-n akan dijalankan, sedangkan jika kondisi salah maka program akan keluar dari struktur if.

- **Bentuk Umum IF – Else**

```

if (kondisi)
    Pernyataan-1;
else
    Pernyataan-2;

```

Jika kondisi benar maka pernyataan akan dijalankan, dan jika kondisi salah maka pernyataan-2 yang akan dijalankan.

- **Bentuk Umum Nested If**

```

If (kondisi1)
    Pernyataan-1; else if (kondisi 2)
    Pernyataan-2; else if (kondisi 3)
    Pernyataan-2;
    ...
    ... else
    pernyataan;

```

### 2.2.1.2 Switch

Switch adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan statement untuk dieksekusi, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksi. Setiap ungkapan diungkapkan dengan sebuah nilai integral konstan, seperti sebuah nilai dengan tipe byte, short, int atau char.

- **Bentuk Umum Switch**

```

Switch (Ekspresi)
{

    case nilai1 : {
        Statement1;
        Statement1;
    }

    break;

```

```

case nilai2 : {
    Statement2;
    Statement2;
}

break;
default : {
Statement1;
    Statement1;
}

break;
}

```

Keterangan :

- a. Case : digunakan sebagai label yang menandai awal eksekusi deret instruksinya hingga ditemukan pernyataan break.
- b. Default : label yang digunakan atau dieksekusi apabila label case tidak ada yang memenuhi.
- c. Break : Perintah mengontrol alur program, berfungsi untuk keluar dari satu blok kondisi ataupun iterasi.

**Contoh syntax selection If**

```

package com.ibik.pbo.selections;

import java.util.Scanner;

public class HandphoneMain {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String merek = "Samsung";
        Integer battery = 10;
        int stock;
        Boolean isSell = false;

        System.out.print("Enter name of tipe phone:");
        String tipe = input.nextLine();

        System.out.print("Enter color of phone:");
        String color = input.nextLine();

        System.out.print("Enter battery health:");
        battery = input.nextInt();

        if(color.equals("red")) {
            stock = 5; isSell=true;
        }else if(color.equals("blue")){
            stock = 2; isSell =true;
        }else {
            stock = 0;
        }

        //output
        System.out.println("\n====="+merek+"-"+tipe+"=====");
        System.out.println("Baterai "+battery+"% ");
        if(battery < 20) {

            System.out.print("(need charching)");
        }
        System.out.println("Color "+color+"\nStock:"+stock);
        System.out.println("====="+((isSell) ? "for sale" :
"sold").toUpperCase() + "=====" );

        input.close();

    }
}

```

### Contoh syntax selection Switch:

```
package com.ibik.pbo.selections;

import java.util.Scanner;

public class HandphoneMain {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String merek = "Samsung";
        Integer battery = 10;
        int stock;
        Boolean isSell = false;

        System.out.print("Enter name of tipe phone:");
        String tipe = input.nextLine();

        System.out.print("Enter color of phone:");
        String color = input.nextLine();

        System.out.print("Enter battery health:");
        battery = input.nextInt();

        switch (color) {
            case "red": {
                stock = 5;
                break;
            }
            case "blue": {
                stock = 2;
                break;
            }
            default:
                stock = 0;
        }

        //output
        System.out.println("\n====="+merek+"-"+tipe+"=====");
        System.out.println("Baterai "+battery+"% ");
        if(battery < 20) {
            System.out.print("(need charching)");
        }
        System.out.println("Color "+color+"\nStock:"+stock);
        System.out.println("====="+((isSell) ? "for sale" :
"sold").toUpperCase() + "=====" );

        input.close();
    }
}
```

### 2.2.2 Struktur Pengulangan/ Repetition

Struktur pengulangan digunakan untuk mengulang Sebuah / kelompok instruksi diulang untuk jumlah pengulangan tertentu. Beberapa hal penting pada struktur pengulangan adalah adanya inisialisasi, kondisi, dan iterasi. Ada empat struktur pengulangan yang dapat digunakan pada bahasa JAVA, yaitu: for, while,

do...while, dan foreach.

Berkaitan dengan proses pengulangan, pemrograman JAVA juga menyediakan pernyataan *break* (untuk mengakhiri pengulangan) , *goto* , dan *continue*(untuk melakukan pengulangan selanjutnya) .

#### 2.2.2.1 Struktur For

Struktur pengulangan for biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah pengulangannya.

- **Bentuk Umum For**

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

**Keterangan:**

Inisialisasi : pernyataan untuk menyatakan keadaan awal dari variabel kontrol.

Kondisi : ekspresi relasi yang menyatakan kondisi untuk keluar dari pengulangan.

Iterasi : pengatur perubahan nilai variabel kontrol.

#### 2.2.2.2 Struktur While

Struktur ini digunakan bila jumlah perulangannya belum diketahui. Proses perulangan akan terus berlanjut selama kondisinya bernilai benar (true) dan akan berhenti bila kondisinya bernilai salah.

- **Bentuk Umum While**

```
While (kondisi){  
    Pernyataan  
    ...  
}
```

#### 2.2.2.3 Do While

Pada dasarnya struktur pengulangan ini sama saja dengan struktur while, hanya saja pada proses pengulangan dengan while, seleksi berada di while yang

letaknya diatas, sementara pada pengulangan do ... while, seleksi berada dibawah batas pengulangan. Jadi dengan menggunakan struktur do .. while sekurang – kurangnya akan terjadi satu kali pengulangan.

- **Bentuk Umum Do While**

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

### Contoh syntax repetition

```
package com.ibik.pbo.repetitions;  
  
import java.util.Scanner;  
  
public class RepetitionMain {  
  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        int number;  
        int a=0;  
  
        System.out.print("Enter the number of looping:");  
  
        number = input.nextInt();  
  
        //Looping number  
        System.out.println("\n==USE WHILE==");  
        while (a < number) {  
            System.out.print(a);  
            a++;  
        }  
  
        System.out.println("\n==USE DO WHILE==");  
        int b = 0;  
        do {  
            System.out.print(b);  
            b++;  
        } while (b < number);  
  
        System.out.println("\n==USE FOR==");  
        for (int c = 0; c < number; c++) {  
            System.out.print(c);  
        }  
  
        input.close();  
    }  
}
```

## 2.3 Latihan Pembelajaran

1. Buatlah project JAVA dengan nama Pembelajaran-02-Selection-Repetition.

2. Buatlah program main JAVA untuk menentukan apakah nilai inputan NPM anda termasuk bilangan PRIMA. Jika inputan NPM adalah bilangan prima, maka mencetak “*Termasuk bilangan PRIMA*”, jika tidak maka mencetak “*Bukan termasuk bilangan prima karena dapat dibagi dengan 2*”.
- Simpan dengan nama file Latihan02.java pada proyek anda.

3. Buatlah program main JAVA untuk membuat program pemesanan menu

Menu Sarapan:

Nasi Goreng

Bubur Ayam

Soto Ayam

makanan dengan pilihan seperti dibawah ini:

Jika user memilih menu 1 maka mencetak “Anda memesan Nasi Goreng dengan harga Rp 22.000,-”.

Jika memilih menu 2 maka mencetak “Anda memesan Bubur Ayam dengan harga Rp15.000,- ”.

Jika memilih menu 3 maka mencetak “Anda memesan Soto Ayam dengan harga Rp 25.000,- ”.

Jika memasukan pilihan yang tidak ada di menu maka mencetak “Maaf menu yang anda masukan salah.”

Simpanlah dengan nama file Latihan03.java pada proyek anda.

4. Buatlah program main JAVA untuk mencari nilai faktorisasi. Dengan ketentuan nilai inputan factor tidak boleh kurang dari 20. Jika NPM anda GENAP maka hasil nilai faktorisasi diurutkan secara Ascending dan GANJIL diurutkan secara Descending.

Contoh:

**ASC:** Faktor 25 adalah 1, 5, 25

**DESC:** Faktor 25 adalah 25, 5, 1

5. Buatlah program main JAVA untuk mencetak Persegi seperti gambar



```
Enter the number of row:7
* * * * * * * * * 
* * * * * *   *   *
* * * * *     *   *
* * * *      *   *
* * *        *   *
* *         *   *
*          *   *
*           *   *
*            *   *
*             *   *
*              *   *
*               *   *
*                *   *
*                 *   *
*                  *   *
```

Simpanlah dengan nama file Latihan05.java pada proyek anda.

- Pada class main JAVA buatlah source code untuk menjalankan program yang sama seperti soal nomor 3 (menampilkan dan memasukkan pilihan menu). Namun harga yang ditampilkan sesuai dengan nilai balik pada method getPrices.

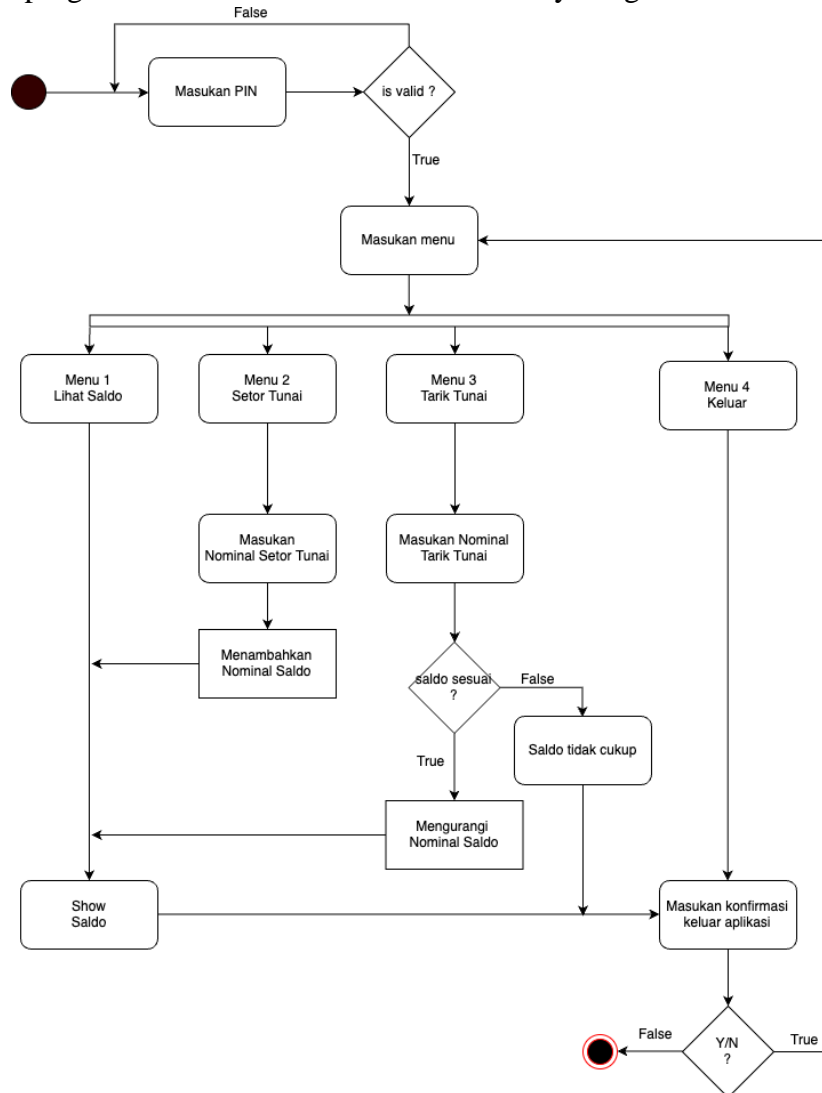
Dan jika sudah memenuhi kebutuhannya maka aplikasi akan menampilkan pilihan “Apakah anda akan memesan kembali ? (y/n)”, jika memilih ‘y’ maka kembali mengisi menu pilihan sedangkan jika ‘n’ maka program selesai dan menampilkan “Terimakasih telah memesan.”

Simpanlah dengan nama Latihan07.java pada projek anda.

8. Buatlah rancangan Class Diagram berdasarkan soal nomor 7.

9. Buatlah rancangan Activity Diagram berdasarkan soal nomor 7.

10. Buatlah program main JAVA berdasarkan Activity Diagram dibawah ini:



Activity diagram diatas ialah program sederhana mengenai program ATM, dimana nilai PIN ialah NPM anda. Jika tidak cocok inputan PIN maka kembali diminta untuk memasukan PIN. Sedangkan jika berhasil maka akan dihadapkan dengan 4 buah menu, yaitu Lihat Saldo, Setor Tunai, Tarik Tunai dan Keluar.

Simpanlah dengan nama file Latihan10.java pada proyek anda.