

**LAPORAN**  
**PEMROGRAMAN BERORIENTASI OBJEK**



Nama : M. Akbar Amiruddin  
Nim : 13020220023  
Kelas : A1

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS MUSLIM INDONESIA**  
**MAKASSAR**

**2024**

## 1. BacaString.java

```
7 import java.io.BufferedReader;
8 import java.io.IOException;
9 import java.io.InputStreamReader;
10 import javax.swing.*;
11
12 public class BacaString {
13     /**
14      * @param args
15      * @throws IOException
16      */
17
18     public static void main(String[] args) throws IOException {
19         // TODO Auto-generated method stub
20
21         /* Kamus */
22         String str;
23         BufferedReader dataIn = new BufferedReader(new
24             InputStreamReader(System.in));
25
26         /* Program */
27         System.out.print ("\nBaca string dan Integer: \n");
28         System.out.print("masukkan sebuah string: ");
29         str= dataIn.readLine();
30         System.out.print ("String yang dibaca : "+ str);
31     }
32 }
```

### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac BacaString.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java BacaString

Baca string dan Integer:
masukkan sebuah string: Akbar
String yang dibaca : Akbar
C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>
```

Codingan di atas adalah program sederhana dalam bahasa pemrograman Java yang bertujuan untuk membaca sebuah string yang dimasukkan oleh pengguna melalui input keyboard, dan kemudian menampilkannya kembali ke layar.

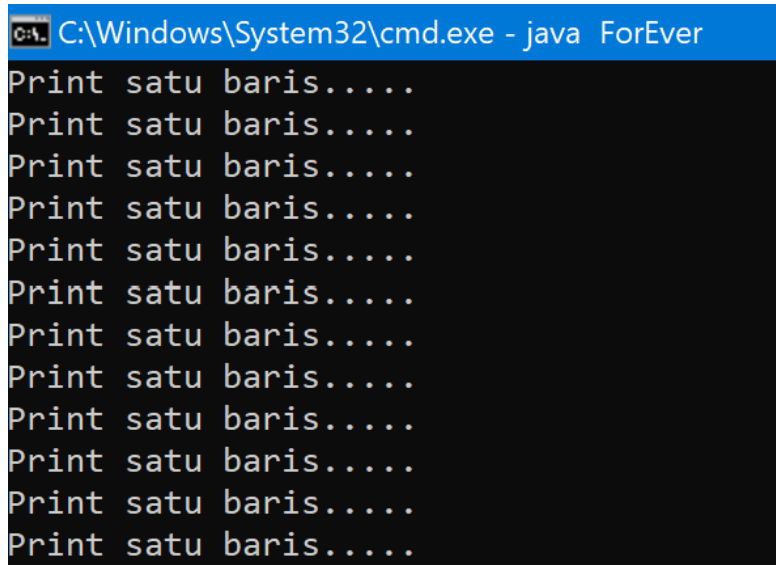
1. **\*\*Import Statements\*\***: Pada bagian awal program, terdapat beberapa pernyataan `import` yang mengimpor kelas-kelas yang diperlukan dari paket Java. Dalam hal ini, `java.io.BufferedReader`, `java.io.IOException`, `java.io.InputStreamReader`, dan `javax.swing.JOptionPane` diimpor.
2. **\*\*Class Declaration\*\***: Program ini memiliki satu kelas yang disebut `BacaString`.
3. **\*\*Main Method\*\***: Method `main()` adalah titik masuk utama untuk program Java. Program dimulai dari sini. Method `main()` memiliki satu parameter, yaitu array of strings `args`, yang dapat digunakan untuk meneruskan argumen dari baris perintah saat menjalankan program.
4. **\*\*Variabel\*\***: Dua variabel dideklarasikan di dalam method `main()`. Pertama, `str` yang bertipe `String` untuk menyimpan string yang dimasukkan oleh pengguna. Kedua, `datAIn` yang bertipe `BufferedReader` untuk membaca input dari keyboard.
5. **\*\*Input dan Output\*\***: Program meminta pengguna untuk memasukkan sebuah string dengan menampilkan pesan "masukkan sebuah string: ". Kemudian, string yang dimasukkan oleh pengguna dibaca menggunakan `readLine()` dari objek `datAIn` (`BufferedReader`) dan disimpan dalam variabel `str`. Selanjutnya, program mencetak kembali string yang dimasukkan oleh pengguna dengan pesan "String yang dibaca : ".
6. **\*\*Exception Handling\*\***: Method `main()` menggunakan pernyataan `throws IOException` untuk menangani pengecualian yang mungkin terjadi saat membaca input dari keyboard dengan `readLine()`.

Program tersebut adalah contoh sederhana untuk membaca dan menampilkan sebuah string yang dimasukkan oleh pengguna dalam bahasa Java.

## 2. ForEver.java

```
C: > Users > ASUS > Documents > PBO > Tugas 3 > Tugas3_SourceCode > ForEver.java > ForE
7 public class ForEver {
8     /**
9     * @param args
10    */
11
12    public static void main(String[] args) {
13        // TODO Auto-generated method stub
14
15        /* Program */
16        System.out.println("Program akan looping, akhiri dengan ^c");
17        while (true)
18        { System.out.print ("Print satu baris.....\n");
19        }
20    }
21 }
```

### Output:



```
C:\Windows\System32\cmd.exe - java ForEver
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
```

Codingan di atas adalah program Java sederhana yang menggunakan loop `while (true)` untuk melakukan iterasi tanpa henti.

1. **\*\*Class Declaration\*\***: Program ini memiliki satu kelas yang disebut `ForEver`.
2. **\*\*Main Method\*\***: Method `main()` adalah titik masuk utama untuk program Java. Program dimulai dari sini. Method `main()` memiliki satu parameter, yaitu array of strings `args`, yang dapat digunakan untuk meneruskan argumen dari baris perintah saat menjalankan program.
3. **\*\*Output Pesan\*\***: Program mencetak pesan "Program akan looping, akhiri dengan ^c" menggunakan `System.out.println()`. Pesan ini memberikan informasi kepada pengguna bahwa program akan melakukan looping tanpa henti dan dapat dihentikan dengan menekan kombinasi tombol ^c (Ctrl + c).
4. **\*\*Looping Tanpa Henti\*\***: Setelah mencetak pesan, program masuk ke dalam loop `while (true)`. Ini berarti bahwa kondisi `true` akan selalu dievaluasi sebagai benar, sehingga loop akan terus berjalan tanpa henti.
5. **\*\*Cetak Baris\*\***: Di dalam loop, program mencetak pesan "Print satu baris....." menggunakan `System.out.print()` diikuti dengan karakter newline `\n`.

Karena tidak ada cara untuk keluar dari loop `while (true)` secara normal (karena kondisi `true` selalu benar), program akan terus mencetak pesan tersebut tanpa henti sampai dihentikan secara paksa oleh pengguna, biasanya dengan menekan kombinasi tombol ^c (Ctrl + c) untuk menghentikan eksekusi program.

### 3. If1.java

```

C: > Users > ASUS > Documents > PBO > Tugas 3 > Tugas3_SourceCode > If1.java > If1
7  /* contoh pemakaian IF satu kasus */
8  /* membaca nilai integer, menuliskan nilainya jika positif */
9  import java.util.Scanner;
10
11  public class If1 {
12  /**
13   * @param args
14   */
15
16      public static void main(String[] args) {
17          // TODO Auto-generated method stub
18
19          /* Kamus */
20          Scanner masukan=new Scanner(System.in);
21          int a;
22
23          /* Program */
24          System.out.print ("Contoh IF satu kasus \n");
25          System.out.print ("Ketikkan suatu nilai integer : ");
26          a = masukan.nextInt();
27          if (a >= 0)
28              System.out.print ("\nNilai a positif "+ a);
29          }
30  }

```

### Output:

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac If1.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 7

Nilai a positif 7
C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>

```

Codingan di atas adalah program sederhana dalam bahasa pemrograman Java yang membaca nilai integer yang dimasukkan oleh pengguna melalui keyboard dan menampilkan nilai tersebut jika nilainya positif.

1. **\*\*Import Statements\*\***: Program menggunakan kelas `Scanner` dari paket `java.util` untuk membaca input dari pengguna. Oleh karena itu, kita perlu mengimpor kelas `Scanner`.
2. **\*\*Class Declaration\*\***: Program memiliki satu kelas yang disebut `If1`.
3. **\*\*Main Method\*\***: Method `main()` adalah titik masuk utama untuk program Java. Program dimulai dari sini. Method `main()` memiliki satu parameter, yaitu array of strings `args`, yang dapat digunakan untuk meneruskan argumen dari baris perintah saat menjalankan program.

4. **\*\*Variabel\*\***: Program mendeklarasikan variabel `masukan` dari tipe `Scanner` untuk membaca input dari pengguna, dan variabel `a` dari tipe `int` untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. **\*\*Output Pesan\*\***: Program mencetak pesan "Contoh IF satu kasus" dan "Ketikkan suatu nilai integer : " untuk meminta pengguna memasukkan nilai integer.
6. **\*\*Input Nilai\*\***: Program menggunakan `masukan.nextInt()` untuk membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `a`.
7. **\*\*Percabangan IF\*\***: Program menggunakan pernyataan `if` untuk mengecek apakah nilai `a` lebih besar dari atau sama dengan nol. Jika nilai `a` positif atau nol, maka program akan mencetak pesan "Nilai a positif" diikuti dengan nilai `a`.

Jika pengguna memasukkan nilai negatif, tidak akan ada output yang dihasilkan karena tidak ada pernyataan `print` yang terkait dengan nilai negatif. Program hanya menampilkan nilai jika nilainya positif atau nol.

#### 4. If2.java

```
C: > Users > ASUS > Documents > PBO > Tugas 3 > Tugas3_SourceCode > J If2.java > If2.java
7  import java.util.Scanner;
8  /* contoh pemakaian IF dua kasus komplementer */
9  /* Membaca sebuah nilai, */
10 /* menuliskan 'Nilai a positif , nilai a', jika a >=0 */
11 /* 'Nilai a negatif , nilai a', jika a <0 */
12
13 public class If2 {
14     /**
15     * @param args
16     */
17
18     public static void main(String[] args) {
19         // TODO Auto-generated method stub
20
21         /* Kamus */
22         int a;
23         Scanner masukan=new Scanner(System.in);
24
25         /* Program */
26         System.out.print ("Contoh IF dua kasus \n");
27         System.out.print ("Ketikkan suatu nilai integer :");
28         a=masukan.nextInt();
29         if (a >= 0){
30             System.out.println ("Nilai a positif "+ a);
31         }else /* a < 0 */
32         {
33             System.out.println ("Nilai a negatif "+ a);
34         }
35     }
36 }
```

## Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac If2.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :7
Nilai a positif 7

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac If2.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :-7
Nilai a negatif -7

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>_
```

Codingan di atas adalah program sederhana dalam bahasa pemrograman Java yang menggunakan struktur percabangan IF dua kasus untuk menentukan apakah nilai yang dimasukkan oleh pengguna adalah positif atau negatif, dan kemudian menampilkan pesan sesuai dengan kondisi tersebut.

1. **\*\*Import Statement\*\***: Program menggunakan kelas `Scanner` dari paket `java.util` untuk membaca input dari pengguna. Oleh karena itu, kelas `Scanner` diimpor.
2. **\*\*Class Declaration\*\***: Program memiliki satu kelas yang disebut `If2`.
3. **\*\*Main Method\*\***: Method `main()` adalah titik masuk utama untuk program Java. Method `main()` memiliki satu parameter, yaitu array of strings `args`, yang dapat digunakan untuk meneruskan argumen dari baris perintah saat menjalankan program.
4. **\*\*Variabel\*\***: Program mendeklarasikan variabel `a` sebagai tipe data `int` untuk menyimpan nilai yang dimasukkan oleh pengguna, dan variabel `masukan` sebagai objek `Scanner` untuk membaca input dari pengguna.
5. **\*\*Output Pesan\*\***: Program mencetak pesan "Contoh IF dua kasus" dan "Ketikkan suatu nilai integer : " untuk meminta pengguna memasukkan nilai integer.
6. **\*\*Input Nilai\*\***: Program menggunakan `masukan.nextInt()` untuk membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `a`.
7. **\*\*Percabangan IF\*\***: Program menggunakan struktur percabangan IF untuk mengecek apakah nilai `a` lebih besar dari atau sama dengan nol. Jika kondisi tersebut benar, program akan mencetak pesan "Nilai a positif" diikuti dengan nilai `a`. Jika kondisi tersebut salah, program akan menjalankan blok else yang mencetak pesan "Nilai a negatif" diikuti dengan nilai `a`.

Program ini akan menampilkan pesan yang sesuai tergantung pada nilai yang dimasukkan oleh pengguna, yaitu "Nilai a positif" jika nilai lebih besar dari atau sama dengan nol, dan "Nilai a negatif" jika nilai kurang dari nol.

## 5. If3.java

```
C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode> J If3.java > If3 > 0
7  import java.util.Scanner;
8  /* contoh pemakaian IF tiga kasus */ /* Membaca sebuah nilai, */
9  /* menuliskan 'Nilai a positif , nilai a', jika a >0 */
10 /* 'Nilai Nol , nilai a', jika a = 0 */
11 /* 'Nilai a negatif , nilai a', jika a <0 */
12
13 public class If3 {
14     /**
15     * @param args
16     */
17
18     public static void main(String[] args) {
19         // TODO Auto-generated method stub
20
21         /* Kamus */
22         int a;
23         Scanner masukan=new Scanner(System.in);
24
25         /* Program */
26         System.out.print ("Contoh IF tiga kasus \n");
27         System.out.print ("Ketikkan suatu nilai integer :");
28         a=masukan.nextInt();
29         if (a > 0){
30             System.out.println ("Nilai a positif "+ a);
31         }else if (a == 0){
32             System.out.println ("Nilai Nol "+ a);
33         }else /* a > 0 */ {
34             System.out.println ("Nilai a negatif "+ a);
35         }
36     }
37 }
```

### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac If3.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :7
Nilai a positif 7

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac If3.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :0
Nilai Nol 0

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac If3.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :-7
Nilai a negatif -7
```



Codingan di atas adalah program sederhana dalam bahasa pemrograman Java yang menggunakan struktur percabangan IF tiga kasus untuk menentukan apakah nilai yang dimasukkan oleh pengguna adalah positif, nol, atau negatif, dan kemudian menampilkan pesan sesuai dengan kondisi tersebut.

1. **\*\*Import Statement\*\***: Program menggunakan kelas `Scanner` dari paket `java.util` untuk membaca input dari pengguna. Oleh karena itu, kelas `Scanner` diimpor.
2. **\*\*Class Declaration\*\***: Program memiliki satu kelas yang disebut `If3`.
3. **\*\*Main Method\*\***: Method `main()` adalah titik masuk utama untuk program Java. Method `main()` memiliki satu parameter, yaitu array of strings `args`, yang dapat digunakan untuk meneruskan argumen dari baris perintah saat menjalankan program.
4. **\*\*Variabel\*\***: Program mendeklarasikan variabel `a` sebagai tipe data `int` untuk menyimpan nilai yang dimasukkan oleh pengguna, dan variabel `masukan` sebagai objek `Scanner` untuk membaca input dari pengguna.
5. **\*\*Output Pesan\*\***: Program mencetak pesan "Contoh IF tiga kasus" dan "Ketikkan suatu nilai integer :" untuk meminta pengguna memasukkan nilai integer.
6. **\*\*Input Nilai\*\***: Program menggunakan `masukan.nextInt()` untuk membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `a`.
7. **\*\*Percabangan IF\*\***: Program menggunakan struktur percabangan IF tiga kasus untuk menentukan kondisi nilai `a`.
  - Jika nilai `a` lebih besar dari nol, program akan mencetak pesan "Nilai a positif" diikuti dengan nilai `a`.
  - Jika nilai `a` sama dengan nol, program akan mencetak pesan "Nilai Nol" diikuti dengan nilai `a`.
  - Jika nilai `a` kurang dari nol, program akan mencetak pesan "Nilai a negatif" diikuti dengan nilai `a`.

Program ini akan menampilkan pesan yang sesuai tergantung pada nilai yang dimasukkan oleh pengguna, yaitu "Nilai a positif" jika nilai lebih besar dari nol, "Nilai Nol" jika nilai sama dengan nol, dan "Nilai a negatif" jika nilai kurang dari nol.

## 6. KasusBoolean.java

```

7  /* Ekspresi kondisional dengan boolean */
8  public class KasusBoolean {
9      /**
10     * @param args
11     */
12
13     public static void main(String[] args) {
14         // TODO Auto-generated method stub
15
16         /* Kamus */
17         boolean bool;
18
19         /* Algoritma */
20         bool = true;
21         if(bool) {
22             System.out.print("true\n");
23         } else
24             System.out.print("false\n");
25         if(!bool) {
26             System.out.print("salah\n");
27         } else
28             System.out.print("benar\n");
29     }
30 }

```

### Output:

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac KasusBoolean.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java KasusBoolean
true
benar

```

Codingan di atas adalah program sederhana dalam bahasa pemrograman Java yang menunjukkan contoh penggunaan ekspresi kondisional dengan tipe data boolean.

1. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan sebuah variabel bernama `bool` dengan tipe data boolean. Variabel ini akan digunakan untuk menyimpan nilai boolean.
2. **\*\*Inisialisasi Variabel\*\***: Variabel `bool` diinisialisasi dengan nilai `true`.
3. **\*\*Percabangan IF\*\***: Program menggunakan struktur percabangan IF untuk mengecek nilai dari variabel `bool`.
  - Jika nilai `bool` adalah `true`, maka blok pertama dari pernyataan `if` dijalankan. Ini berarti program akan mencetak "true".
  - Setelah itu, program menggunakan percabangan IF lagi untuk mengecek apakah nilai dari `!bool` (negasi dari `bool`) adalah `true` atau `false`. Karena nilai `bool` adalah `true`, nilai dari `!bool` adalah `false`, sehingga blok kedua dari pernyataan `if` dijalankan. Ini berarti program mencetak "benar".

Karena `bool` bernilai `true`, maka blok pertama dari percabangan `if` dijalankan dan mencetak "true". Selanjutnya, karena `bool` bernilai `true`, maka blok kedua dari

percabangan `if` yang mengecek `!bool` (yang bernilai `false`) dijalankan dan mencetak "benar".

## 7. KasusSwitch.java

```
C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode> J KasusSwitch.java > ...
7  import java.util.Scanner;
8
9  public class KasusSwitch {
10     /**
11     * @param args
12     */
13
14     public static void main(String[] args) {
15         // TODO Auto-generated method stub
16
17         /* Kamus */
18         char cc;
19         Scanner masukan=new Scanner(System.in);
20
21         /* Program */
22         System.out.print ("Ketikkan sebuah huruf, akhiri dengan RETURN \n");
23         cc=masukan.next().charAt(0);
24         switch (cc) {
25             case 'a': { System.out.print ("Yang anda ketik adalah a\n"); break;}
26             case 'u': { System.out.print ("Yang anda ketik adalah u\n"); break;}
27             case 'e': { System.out.print ("Yang anda ketik adalah e\n"); break;}
28             case 'i': { System.out.print ("Yang anda ketik adalah i\n"); break;}
29             case 'o': { System.out.print ("Yang anda ketik adalah o\n"); break;}
30             default:
31                 System.out.print ("Yang anda ketik adalah huruf mati\n");
32             }
33         }
34     }
35 }
```

### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac KasusSwitch.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac KasusSwitch.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
z
Yang anda ketik adalah huruf mati
```

Codingan di atas adalah program sederhana dalam bahasa pemrograman Java yang menggunakan struktur switch-case untuk memeriksa huruf yang dimasukkan oleh pengguna dan menampilkan pesan sesuai dengan huruf yang dimasukkan.

1. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan sebuah variabel `cc` dengan tipe data `char` yang akan digunakan untuk menyimpan huruf yang dimasukkan oleh pengguna. Selain itu, program juga membuat objek `Scanner` untuk membaca input dari pengguna.

2. **\*\*Input Huruf\*\***: Program mencetak pesan "Ketikkan sebuah huruf, akhiri dengan RETURN" untuk meminta pengguna memasukkan sebuah huruf. Setelah itu, program menggunakan `masukan.next().charAt(0)` untuk membaca huruf yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `cc`.
3. **\*\*Struktur Switch-case\*\***: Program menggunakan struktur switch-case untuk memeriksa nilai dari variabel `cc`.
  - Jika nilai `cc` adalah 'a', maka program akan mencetak pesan "Yang anda ketik adalah a".
  - Jika nilai `cc` adalah 'u', maka program akan mencetak pesan "Yang anda ketik adalah u".
  - Jika nilai `cc` adalah 'e', maka program akan mencetak pesan "Yang anda ketik adalah e".
  - Jika nilai `cc` adalah 'i', maka program akan mencetak pesan "Yang anda ketik adalah i".
  - Jika nilai `cc` adalah 'o', maka program akan mencetak pesan "Yang anda ketik adalah o".
  - Jika nilai `cc` tidak sama dengan 'a', 'u', 'e', 'i', atau 'o', maka program akan mencetak pesan "Yang anda ketik adalah huruf mati".
4. **\*\*Break Statement\*\***: Setiap kasus dalam struktur switch-case diakhiri dengan pernyataan `break` untuk keluar dari struktur switch-case setelah kasus yang sesuai telah dieksekusi.

Output dari program akan tergantung pada huruf yang dimasukkan oleh pengguna. Jika huruf yang dimasukkan oleh pengguna tidak termasuk dalam 'a', 'u', 'e', 'i', atau 'o', maka program akan mencetak "Yang anda ketik adalah huruf mati".

## 8. Konstant.java

```
7 import java.util.Scanner;
8 /* Membaca jari-jari, menghitung luas lingkaran */
9 /* latihan pemakaian konstanta */
10
11 public class Konstant {
12     /**
13     * @param args
14     */
15
16     public static void main(String[] args) {
17         // TODO Auto-generated method stub
18
19         /* Kamus */
20         final float PHI = 3.1415f;
21         float r;
22         Scanner masukan=new Scanner(System.in);
23
24         /* program */ /* baca data */
25         System.out.print ("Jari-jari lingkaran =");
26         r = masukan.nextFloat();
27
28         /* Hitung dan tulis hasil */
29         System.out.print ("Luas lingkaran = "+ (PHI * r * r)+"\n");
30         System.out.print ("Akhir program \n");
31     }
32 }
```

### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac Konstant.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java Konstant
Jari-jari lingkaran = 7
Luas lingkaran = 153.9335
Akhir program
```

Codingan di atas adalah program sederhana dalam bahasa pemrograman Java yang bertujuan untuk menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan oleh pengguna.

1. **\*\*Deklarasi Konstanta\*\***: Program mendefinisikan konstanta `PHI` dengan nilai 3.1415f. Konstanta ini merepresentasikan nilai  $\pi$  (pi) dalam matematika, yang digunakan untuk menghitung luas lingkaran.
2. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan variabel `r` dengan tipe data float. Variabel ini akan digunakan untuk menyimpan nilai jari-jari lingkaran yang dimasukkan oleh pengguna.
3. **\*\*Import Scanner\*\***: Program mengimpor kelas `Scanner` dari paket `java.util` untuk membaca input dari pengguna.
4. **\*\*Input Jari-Jari Lingkaran\*\***: Program mencetak pesan "Jari-jari lingkaran =" untuk meminta pengguna memasukkan nilai jari-jari lingkaran. Setelah itu, program menggunakan `masukan.nextFloat()` untuk membaca nilai float yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `r`.
5. **\*\*Perhitungan Luas Lingkaran\*\***: Program menghitung luas lingkaran menggunakan rumus  $\pi r^2$ , di mana `PHI` adalah konstanta  $\pi$  yang sudah didefinisikan sebelumnya, dan `r` adalah jari-jari lingkaran yang dimasukkan oleh pengguna.
6. **\*\*Output\*\***: Program mencetak hasil perhitungan luas lingkaran dengan menggunakan `System.out.print()`. Selain itu, program juga mencetak pesan "Akhir program" untuk menandakan bahwa program telah selesai dijalankan.

Program ini memungkinkan pengguna untuk menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan dan menggunakan konstanta untuk nilai  $\pi$ .

### 9. Max2.java

```

7  import java.util.Scanner;
8  /* Maksimum dua bilangan yang dibaca */
9
10 public class Max2 {
11     /**
12     * @param args
13     */
14
15     public static void main(String[] args) {
16         // TODO Auto-generated method stub
17
18         /* Kamus */
19         int a, b;
20         Scanner masukan=new Scanner(System.in);
21
22         /* Program */
23         System.out.print ("Maksimum dua bilangan : \n");
24         System.out.print ("Ketikkan dua bilangan, pisahkan dg RETURN : \n");
25         a=masukan.nextInt();
26         b=masukan.nextInt();
27         System.out.println ("Ke dua bilangan : a = "+ a+" b = "+b);
28         if (a >= b){
29             System.out.println ("Nilai a yang maksimum "+ a);
30         }else /* a > b */{
31             System.out.println ("Nilai b yang maksimum: "+ b);
32         }
33     }
34 }

```

### Output:

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac Max2.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
7
17
Ke dua bilangan : a = 7 b = 17
Nilai b yang maksimum: 17

```

Codingan di atas adalah program sederhana dalam bahasa pemrograman Java yang bertujuan untuk menentukan nilai maksimum dari dua bilangan yang dimasukkan oleh pengguna.

1. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan dua variabel `a` dan `b` dengan tipe data `int`. Variabel ini akan digunakan untuk menyimpan dua bilangan yang dimasukkan oleh pengguna.
2. **\*\*Import Scanner\*\***: Program mengimpor kelas `Scanner` dari paket `java.util` untuk membaca input dari pengguna.

3. **\*\*Input Bilangan\*\***: Program mencetak pesan "Maksimum dua bilangan" dan "Ketikkan dua bilangan, pisahkan dg RETURN :\" untuk meminta pengguna memasukkan dua bilangan. Setelah itu, program menggunakan `masukan.nextInt()` dua kali untuk membaca dua bilangan yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `a` dan `b` secara berturut-turut.
4. **\*\*Output\*\***: Program mencetak kembali kedua bilangan yang dimasukkan oleh pengguna dengan pesan "Ke dua bilangan : a = \" diikuti dengan nilai `a`, dan "b = \" diikuti dengan nilai `b`.
5. **\*\*Percabangan IF\*\***: Program menggunakan struktur percabangan IF untuk mengecek mana di antara dua bilangan yang lebih besar.
  - Jika nilai `a` lebih besar dari atau sama dengan nilai `b`, maka program mencetak pesan "Nilai a yang maksimum" diikuti dengan nilai `a`.
  - Jika nilai `a` tidak lebih besar dari atau sama dengan nilai `b`, maka program mencetak pesan "Nilai b yang maksimum" diikuti dengan nilai `b`.

Program ini memungkinkan pengguna untuk memasukkan dua bilangan, dan kemudian menentukan nilai maksimum di antara keduanya. Outputnya akan menampilkan nilai maksimum dari dua bilangan yang dimasukkan.

#### 10. PriFor.java

```
7  import java.util.Scanner;
8  /* Baca N, Print 1 s/d N dengan FOR */
9
10 public class PriFor {
11     /**
12     * @param args
13     */
14
15     public static void main(String[] args) {
16         // TODO Auto-generated method stub
17
18         /* Kamus */
19         int i,N;
20         Scanner masukan=new Scanner(System.in);
21
22         /* Program */
23
24         /* program */
25         System.out.print ("Baca N, print 1 s/d N ");
26         System.out.print ("N = ");
27
28         N=masukan.nextInt();
29         for (i = 1; i <= N; i++){
30             System.out.println (i);
31         }
32         System.out.println ("Akhir program \n");
33     }
```

### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac PriFor.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java PriFor
Baca N, print 1 s/d N N = 7
1
2
3
4
5
6
7
Akhir program
```

Codingan di atas adalah sebuah program sederhana dalam bahasa pemrograman Java yang bertujuan untuk membaca sebuah bilangan bulat N dari pengguna, dan kemudian mencetak angka dari 1 hingga N menggunakan perulangan `for`.

1. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan dua variabel `i` dan `N` dengan tipe data `int`. Variabel `i` digunakan sebagai variabel loop untuk perulangan `for`, sedangkan variabel `N` digunakan untuk menyimpan bilangan bulat yang akan dimasukkan oleh pengguna.
2. **\*\*Import Scanner\*\***: Program mengimpor kelas `Scanner` dari paket `java.util` untuk membaca input dari pengguna.
3. **\*\*Input Bilangan N\*\***: Program mencetak pesan "Baca N, print 1 s/d N" dan "N = " untuk meminta pengguna memasukkan sebuah bilangan bulat. Kemudian, program menggunakan `masukan.nextInt()` untuk membaca bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `N`.
4. **\*\*Perulangan for\*\***: Program menggunakan perulangan `for` untuk mencetak angka dari 1 hingga N.
  - Variabel `i` diinisialisasi dengan nilai 1.
  - Pada setiap iterasi, program mencetak nilai variabel `i`.
  - Setelah mencetak nilai, variabel `i` akan bertambah satu.
  - Perulangan akan terus berlanjut selama nilai `i` kurang dari atau sama dengan nilai `N`.
5. **\*\*Output\*\***: Setelah perulangan selesai, program mencetak pesan "Akhir program" untuk menandakan bahwa program telah selesai dijalankan.

Output dari program ini akan mencetak angka dari 1 hingga N sesuai dengan nilai yang dimasukkan oleh pengguna.

### 11. PrintIterasi.java



```

7  import java.util.Scanner;
8  /* Baca N, */
9  /* Print i = 1 s/d N dengan ITERATE */
10
11  public class PrintIterasi {
12  /**
13   * @param args
14   */
15
16   public static void main(String[] args) {
17   // TODO Auto-generated method stub
18
19   /* Kamus : */
20   int N;
21   int i;
22   Scanner masukan=new Scanner(System.in);
23
24   /* Program */
25   System.out.print ("Nilai N >0 = "); /* Inisialisasi*/
26   N = masukan.nextInt();
27   i = 1; /* First Elmt */
28   System.out.print ("Print i dengan ITERATE : \n");
29   for (;;){
30   System.out.println(i); /* Proses */
31   if (i == N)
32   break; /* Kondisi Berhenti */
33   else {
34   i++; /* Next Elmt */
35   }
36   } /* (i == N) */
37   }
38  }

```

### Output:

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac PrintIterasi.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java PrintIterasi
Nilai N >0 = 7
Print i dengan ITERATE :
1
2
3
4
5
6
7

```

Codingan di atas adalah sebuah program dalam bahasa pemrograman Java yang bertujuan untuk membaca sebuah bilangan bulat N dari pengguna, kemudian mencetak nilai i dari 1 hingga N menggunakan iterasi.

1. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan dua variabel `N` dan `i` dengan tipe data `int`. Variabel `N` digunakan untuk menyimpan bilangan bulat yang akan dimasukkan oleh pengguna, sedangkan variabel `i` digunakan sebagai variabel loop untuk iterasi.
2. **\*\*Import Scanner\*\***: Program mengimpor kelas `Scanner` dari paket `java.util` untuk membaca input dari pengguna.

3. **\*\*Input Bilangan N\*\***: Program mencetak pesan "Nilai N >0 = " untuk meminta pengguna memasukkan sebuah bilangan bulat yang lebih besar dari 0. Kemudian, program menggunakan `masukan.nextInt()` untuk membaca bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `N`.
4. **\*\*Inisialisasi Variabel i\*\***: Program menginisialisasi variabel `i` dengan nilai 1 sebagai elemen pertama yang akan dicetak.
5. **\*\*Perulangan dengan ITERATE\*\***: Program menggunakan perulangan `for` tanpa kondisi untuk melakukan iterasi. Ini berarti perulangan akan berjalan terus menerus hingga kondisi terminasi terpenuhi di dalamnya.
  - Pada setiap iterasi, program mencetak nilai `i`.
  - Kemudian, program memeriksa apakah nilai `i` sudah sama dengan nilai `N`.
  - Jika nilai `i` sudah sama dengan nilai `N`, maka perulangan akan dihentikan dengan menggunakan pernyataan `break`.
  - Jika nilai `i` belum sama dengan nilai `N`, maka nilai `i` akan ditambah 1 untuk mencetak nilai selanjutnya.
6. **\*\*Output\*\***: Setelah perulangan selesai, program tidak mencetak pesan apapun. Program hanya mencetak nilai i dari 1 hingga N sesuai dengan nilai yang dimasukkan oleh pengguna.

Dengan demikian, output dari program ini akan mencetak angka dari 1 hingga N sesuai dengan nilai yang dimasukkan oleh pengguna. Program akan berhenti mencetak ketika nilai i sudah sama dengan nilai N.

## 12. PrintRepeat.java

```
7  import java.util.Scanner;
8  public class PrintRepeat {
9      /**
10     * @param args
11     */
12
13     public static void main(String[] args) {
14         // TODO Auto-generated method stub
15
16         /* Kamus : */
17         int N;
18         int i;
19         Scanner masukan=new Scanner(System.in);
20
21         /* Program */
22         System.out.print ("Nilai N >0 = ");
23
24         /* Inisialisasi */
25         N = masukan.nextInt();
26         i = 1;
27
28         /* First Elmt */
29         System.out.print ("Print i dengan REPEAT: \n");
30         do{
31             System.out.print (i+"\n"); /* Proses */
32             i++; /* Next Elmt */
33         }
34         while (i <= N); /* Kondisi pengulangan */
35     }
36 }
```

### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac PrintRepeat.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java PrintRepeat
Nilai N >0 = 6
Print i dengan REPEAT:
1
2
3
4
5
6
```

Codingan di atas adalah sebuah program dalam bahasa pemrograman Java yang bertujuan untuk membaca sebuah bilangan bulat N dari pengguna, dan kemudian mencetak nilai i dari 1 hingga N menggunakan perulangan `do-while`.

1. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan dua variabel `N` dan `i` dengan tipe data `int`. Variabel `N` digunakan untuk menyimpan bilangan bulat yang akan dimasukkan oleh pengguna, sedangkan variabel `i` digunakan sebagai variabel loop untuk perulangan `do-while`.
2. **\*\*Import Scanner\*\***: Program mengimpor kelas `Scanner` dari paket `java.util` untuk membaca input dari pengguna.
3. **\*\*Input Bilangan N\*\***: Program mencetak pesan "Nilai N >0 = " untuk meminta pengguna memasukkan sebuah bilangan bulat yang lebih besar dari 0. Kemudian, program menggunakan `masukan.nextInt()` untuk membaca bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `N`.
4. **\*\*Inisialisasi Variabel i\*\***: Program menginisialisasi variabel `i` dengan nilai 1 sebagai elemen pertama yang akan dicetak.
5. **\*\*Perulangan dengan REPEAT\*\***: Program menggunakan perulangan `do-while` untuk mencetak angka dari 1 hingga N.
  - Pada setiap iterasi, program mencetak nilai `i`.
  - Kemudian, program menambahkan nilai `i` dengan 1 untuk mencetak nilai selanjutnya.
  - Perulangan akan terus berlanjut selama nilai `i` kurang dari atau sama dengan nilai `N`.
6. **\*\*Output\*\***: Setelah perulangan selesai, program tidak mencetak pesan apapun. Program hanya mencetak nilai i dari 1 hingga N sesuai dengan nilai yang dimasukkan oleh pengguna.

Dengan demikian, program ini akan mencetak angka dari 1 hingga N sesuai dengan nilai yang dimasukkan oleh pengguna. Program akan berhenti mencetak ketika nilai i sudah lebih besar dari nilai N.

### 13. PrintWhile.java

```
6  import java.util.Scanner;
7  /* Baca N, Print i = 1 s/d N dengan WHILE */
8
9  public class PrintWhile {
10     /**
11     * @param args
12     */
13
14     public static void main(String[] args) {
15         // TODO Auto-generated method stub
16
17         /* Kamus : */
18         int N;
19         int i;
20         Scanner masukan=new Scanner(System.in);
21
22         /* Program */
23         System.out.print ("Nilai N >0 = "); /* Inisialisasi */
24         N = masukan.nextInt();
25         i = 1; /* First Elmt */
26         System.out.print ("Print i dengan WHILE: \n");
27         while (i <= N) /* Kondisi pengulangan */
28         { System.out.println (i); /* Proses */
29           i++; /* Next Elmt */
30         }; /* (i > N) */
31     }
32 }
```

#### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac PrintWhile.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java PrintWhile
Nilai N >0 = 5
Print i dengan WHILE:
1
2
3
4
5
```

Codingan di atas adalah sebuah program dalam bahasa pemrograman Java yang bertujuan untuk membaca sebuah bilangan bulat N dari pengguna, dan kemudian mencetak nilai i dari 1 hingga N menggunakan perulangan `while`.

1. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan dua variabel `N` dan `i` dengan tipe data `int`. Variabel `N` digunakan untuk menyimpan bilangan bulat yang akan dimasukkan oleh pengguna, sedangkan variabel `i` digunakan sebagai variabel loop untuk perulangan `while`.
2. **\*\*Import Scanner\*\***: Program mengimpor kelas `Scanner` dari paket `java.util` untuk membaca input dari pengguna.

3. **\*\*Input Bilangan N\*\***: Program mencetak pesan "Nilai N >0 = " untuk meminta pengguna memasukkan sebuah bilangan bulat yang lebih besar dari 0. Kemudian, program menggunakan `masukan.nextInt()` untuk membaca bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `N`.
4. **\*\*Inisialisasi Variabel i\*\***: Program menginisialisasi variabel `i` dengan nilai 1 sebagai elemen pertama yang akan dicetak.
5. **\*\*Perulangan dengan WHILE\*\***: Program menggunakan perulangan `while` untuk mencetak angka dari 1 hingga N.
  - Pada setiap iterasi, program mencetak nilai `i`.
  - Kemudian, program menambahkan nilai `i` dengan 1 untuk mencetak nilai selanjutnya.
  - Perulangan akan terus berlanjut selama nilai `i` kurang dari atau sama dengan nilai `N`.
6. **\*\*Output\*\***: Setelah perulangan selesai, program tidak mencetak pesan apapun. Program hanya mencetak nilai `i` dari 1 hingga N sesuai dengan nilai yang dimasukkan oleh pengguna.

Dengan demikian, program ini akan mencetak angka dari 1 hingga N sesuai dengan nilai yang dimasukkan oleh pengguna. Program akan berhenti mencetak ketika nilai `i` sudah lebih besar dari nilai `N`.

#### 14. PrintWhile1.java

```
7  import java.util.Scanner;
8  /* Baca N, */
9  /* Print i = 1 s/d N dengan while (ringkas) */
10
11  public class PrintWhile1 {
12  /**
13   * @param args
14   */
15
16      public static void main(String[] args) {
17          // TODO Auto-generated method stub
18
19          /* Kamus : */
20          int N;
21          int i = 1;
22          Scanner masukan=new Scanner(System.in);
23
24          /* Program */
25          System.out.print ("Nilai N >0 = ");
26          N = masukan.nextInt();
27          System.out.print ("Print i dengan WHILE (ringkas): \n");
28          while (i <= N){
29              System.out.println (i++);
30          } /* (i > N)*/
31      }
32  }
```

## Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac PrintWhile1.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java PrintWhile1
Nilai N >0 = 7
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
7
```

Codingan di atas adalah sebuah program dalam bahasa pemrograman Java yang bertujuan untuk membaca sebuah bilangan bulat N dari pengguna, dan kemudian mencetak nilai i dari 1 hingga N menggunakan perulangan `while` dalam versi yang lebih ringkas.

1. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan dua variabel `N` dan `i` dengan tipe data `int`. Variabel `N` digunakan untuk menyimpan bilangan bulat yang akan dimasukkan oleh pengguna, sedangkan variabel `i` digunakan sebagai variabel loop untuk perulangan `while`.
2. **\*\*Import Scanner\*\***: Program mengimpor kelas `Scanner` dari paket `java.util` untuk membaca input dari pengguna.
3. **\*\*Input Bilangan N\*\***: Program mencetak pesan "Nilai N >0 = " untuk meminta pengguna memasukkan sebuah bilangan bulat yang lebih besar dari 0. Kemudian, program menggunakan `masukan.nextInt()` untuk membaca bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `N`.
4. **\*\*Inisialisasi Variabel i\*\***: Program menginisialisasi variabel `i` dengan nilai 1 sebagai elemen pertama yang akan dicetak.
5. **\*\*Perulangan dengan WHILE\*\***: Program menggunakan perulangan `while` untuk mencetak angka dari 1 hingga N.
  - Pada setiap iterasi, program mencetak nilai `i`.
  - Kemudian, program menambahkan nilai `i` dengan 1 menggunakan operator postfix `++` untuk mencetak nilai selanjutnya.
  - Perulangan akan terus berlanjut selama nilai `i` kurang dari atau sama dengan nilai `N`.
6. **\*\*Output\*\***: Setelah perulangan selesai, program tidak mencetak pesan apapun. Program hanya mencetak nilai i dari 1 hingga N sesuai dengan nilai yang dimasukkan oleh pengguna.

Dengan demikian, program ini akan mencetak angka dari 1 hingga N sesuai dengan nilai yang dimasukkan oleh pengguna. Program akan berhenti mencetak ketika nilai i sudah lebih besar dari nilai N.

## 15. PrintXinterasi.java

```
7  import java.util.Scanner;
8  /* contoh baca nilai x, */
9  /* Jumlahkan nilai yang dibaca dengan ITERATE */
10
11  public class PrintXinterasi {
12  /**
13   * @param args
14   */
15
16   public static void main(String[] args) {
17   // TODO Auto-generated method stub
18
19   /* Kamus : */
20   int Sum=0;
21   int x;
22   Scanner masukan=new Scanner(System.in);
23
24   /* Program */
25   System.out.print ("Masukkan nilai x (int), akhiri dg 999: ");
26   x = masukan.nextInt(); /* First Elmt */
27   if (x == 999){
28   System.out.print ("Kasus kosong \n");
29   }else{ /* MInimal ada satu data yang dijumlahkan*/
30   Sum = x; /* Inisialisasi; invariant !! */
31   for (;;) {
32   System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");
33   x = masukan.nextInt(); /* Next Elmt */
34   if(x==999)
35   break;
36   else{
37   Sum = Sum + x; /* Proses */
38   }
39   }
40   }
41   System.out.println("Hasil penjumlahan = "+ Sum); /* Terminasi */
42   }
43 }
```

### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac PrintXinterasi.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java PrintXinterasi
Masukkan nilai x (int), akhiri dg 999: 5
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 15
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 30
```

Codingan di atas adalah sebuah program dalam bahasa pemrograman Java yang bertujuan untuk membaca sejumlah nilai x dari pengguna dan kemudian menjumlahkan nilai-nilai tersebut menggunakan perulangan `for` dan penggunaan `break` statement untuk menghentikan perulangan ketika nilai x yang dimasukkan oleh pengguna adalah 999.

1. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan dua variabel, yaitu `Sum` untuk menyimpan hasil penjumlahan nilai-nilai x dan `x` untuk menyimpan nilai

- x yang dibaca dari pengguna. Selain itu, program juga mendeklarasikan objek `masukan` dari kelas `Scanner` untuk membaca input dari pengguna.
2. **\*\*Input Nilai x\*\***: Program mencetak pesan "Masukkan nilai x (int), akhiri dg 999: " untuk meminta pengguna memasukkan nilai x. Kemudian, program menggunakan `masukan.nextInt()` untuk membaca nilai x yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `x`.
  3. **\*\*Perulangan dengan FOR\*\***: Program menggunakan perulangan `for` untuk membaca nilai x dari pengguna dan menjumlahkannya.
    - Pada setiap iterasi, program mencetak pesan "Masukkan nilai x (int), akhiri dg 999: " untuk meminta pengguna memasukkan nilai x selanjutnya.
    - Kemudian, program menggunakan `masukan.nextInt()` untuk membaca nilai x yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `x`.
    - Program kemudian memeriksa apakah nilai x yang dimasukkan adalah 999. Jika ya, perulangan dihentikan menggunakan pernyataan `break`.
    - Jika nilai x bukan 999, program akan menjumlahkannya dengan nilai `Sum`.
  4. **\*\*Output Hasil Penjumlahan\*\***: Setelah perulangan selesai, program mencetak hasil penjumlahan dari semua nilai x yang dimasukkan oleh pengguna.

Dengan demikian, program ini akan mencetak hasil penjumlahan dari semua nilai x yang dimasukkan oleh pengguna, dengan menghentikan proses penjumlahan ketika pengguna memasukkan nilai 999.

## 16. PrintXRepeat.java

```
7  import java.util.Scanner;
8  /* contoh baca nilai x, */
9  /* Jumlahkan nilai yang dibaca dengan ITERATE */
10
11  public class PrintXRepeat {
12  /**
13   * @param args
14   */
15   public static void main(String[] args) {
16       // TODO Auto-generated method stub
17
18       /* Kamus : */
19       int Sum;
20       int x;
21       Scanner masukan=new Scanner(System.in);
22
23       /* Program */
24       System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");
25       x = masukan.nextInt(); /* First Elmt */
26       if (x == 999){
27           System.out.print("Kasus kosong \n");
28       }else { /* Minimal ada satu data yang dijumlahkan*/
29           Sum = 0; /* Inisialisasi; invariant !! */
30           do{
31               Sum = Sum + x; /* Proses */
32               System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");
33               x = masukan.nextInt(); /* Next Elmt */
34           } while (x != 999); /* Kondisi pengulangan */
35           System.out.println ("Hasil penjumlahan = "+Sum);
36           /* Terminasi */
37       }
38   }
39 }
```



### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac PrintXRepeat.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 20
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 40
```

Program di atas merupakan contoh program dalam bahasa Java yang membaca nilai x dari pengguna dan menjumlahkannya menggunakan perulangan `do-while`.

1. **\*\*Deklarasi Variabel\*\***: Program mendeklarasikan dua variabel, yaitu `Sum` untuk menyimpan hasil penjumlahan nilai-nilai x dan `x` untuk menyimpan nilai x yang dibaca dari pengguna. Selain itu, program juga mendeklarasikan objek `masukan` dari kelas `Scanner` untuk membaca input dari pengguna.
2. **\*\*Input Nilai x\*\***: Program mencetak pesan "Masukkan nilai x (int), akhiri dg 999 : " untuk meminta pengguna memasukkan nilai x. Kemudian, program menggunakan `masukan.nextInt()` untuk membaca nilai x yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `x`.
3. **\*\*Perulangan dengan do-while\*\***: Program menggunakan perulangan `do-while` untuk membaca nilai x dari pengguna dan menjumlahkannya.
  - Program memulai perulangan dengan melakukan proses penjumlahan `Sum = Sum + x`.
  - Kemudian, program mencetak pesan "Masukkan nilai x (int), akhiri dg 999 : " untuk meminta pengguna memasukkan nilai x selanjutnya.
  - Program menggunakan `masukan.nextInt()` untuk membaca nilai x yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `x`.
  - Perulangan akan berlanjut selama nilai x yang dimasukkan bukan 999 (`while (x != 999)`).
4. **\*\*Output Hasil Penjumlahan\*\***: Setelah perulangan selesai, program mencetak hasil penjumlahan dari semua nilai x yang dimasukkan oleh pengguna.

Dengan demikian, program akan terus meminta pengguna memasukkan nilai x hingga pengguna memasukkan nilai 999. Setelah itu, program akan mencetak hasil penjumlahan dari semua nilai x yang dimasukkan pengguna kecuali nilai 999.

### 17. PrintXWhile.java

```

7   import java.util.Scanner;
8   /* contoh baca nilai x, */
9   /* Jumlahkan nilai yang dibaca dengan WHILE */
10
11  public class PrintXWhile {
12      /**
13       * @param args
14       */
15
16      public static void main(String[] args) {
17          // TODO Auto-generated method stub
18
19          /* Kamus : */
20          int Sum;
21          int x;
22          Scanner masukan=new Scanner(System.in);
23
24          /* Program */
25          Sum = 0; /* Inisialisasi */
26          System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");
27          x = masukan.nextInt();/* First Elmt*/
28          while (x != 999) /* Kondisi berhenti */ {
29              Sum = Sum + x; /* Proses */
30              System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");
31              x = masukan.nextInt();/* First Elmt*/
32          }
33          System.out.println("Hasil penjumlahan = "+ Sum); /*
34          Terminasi */
35      }
36  }
37

```

### Output:

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac PrintXWhile.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 20
Masukkan nilai x (int), akhiri dg 999 : 20
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 50

```

Program tersebut merupakan contoh penggunaan perulangan `while` untuk membaca nilai `x` dari pengguna dan menjumlahkannya.

1. **\*\*Deklarasi Kamus\*\***: Variabel-variabel yang dideklarasikan untuk menyimpan nilai-nilai yang akan digunakan dalam program, seperti `Sum` untuk menyimpan hasil penjumlahan dan `x` untuk menyimpan nilai yang akan dibaca.
2. **\*\*Membaca Input\*\***: Program meminta pengguna untuk memasukkan nilai `x`. Nilai ini akan diinisialisasi ke dalam variabel `x`.

3. **\*\*Perulangan While\*\***: Program menggunakan perulangan `while` untuk terus membaca nilai `x` dari pengguna selama nilai `x` yang dimasukkan tidak sama dengan 999. Ketika nilai `x` sama dengan 999, perulangan berhenti.
4. **\*\*Proses Penjumlahan\*\***: Di dalam perulangan, setiap nilai `x` yang dimasukkan oleh pengguna akan ditambahkan ke dalam variabel `Sum`.
5. **\*\*Membaca Input Lagi\*\***: Setelah setiap iterasi, program meminta pengguna untuk memasukkan nilai `x` kembali.
6. **\*\*Output Hasil\*\***: Setelah perulangan berakhir karena pengguna memasukkan nilai 999, program mencetak hasil penjumlahan dari semua nilai `x` yang dimasukkan.

Dengan demikian, program ini membaca nilai `x` yang dimasukkan oleh pengguna, menjumlahkannya, dan mencetak hasil penjumlahannya ketika pengguna mengakhiri input dengan memasukkan nilai 999.

## 18. SubProgram.java

```
7  import java.util.Scanner;
8  /* Contoh program yang mengandung prosedur dan fungsi */
9  /* prosedur/fungsi */
10
11  public class SubProgram {
12  /**
13   * @param args
14   */
15
16   public static int maxab (int a, int b){
17   /* mencari maksimum dua bilangan bulat */
18   return ((a >= b) ? a : b);
19   }
20
21   public static void tukar (int a, int b)
22   { /* menukar dua bilangan bulat */
23   int temp;
24   temp = a;
25   a = b;
26   b = temp;
27   System.out.println ("Ke dua bilangan setelah tukar: a = "+ a +" b = "+ b);
28   }
29
30   /*** Program Utama ***/
31   public static void main(String[] args) {
32   // TODO Auto-generated method stub
33   /* Membaca dua bilangan integer */
34   /* Menuliskan maksimum dua bilangan yang dibaca
35   dg memanggil fungsi */
36   /* Menukar kedua bilangan dengan 'prosedur' */
37   int a, b;
38   Scanner masukan=new Scanner(System.in);
39   System.out.print ("Maksimum dua bilangan\n");
40   System.out.print ("Ketikkan dua bilangan, pisahkan dg RETURN : \n");
41   a = masukan.nextInt();
42   b = masukan.nextInt();
43   System.out.println ("Ke dua bilangan : a = "+ a +" b = "+ b);
44   System.out.println ("Maksimum = " + (maxab(a, b)));
45   System.out.print("Tukar kedua bilangan...\n");
46   tukar (a, b);
47   }
48 }
```

### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac SubProgram.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
6
7
Ke dua bilangan : a = 6 b = 7
Maksimum = 7
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 7 b = 6
```

Program `SubProgram` adalah contoh program Java yang mengandung prosedur dan fungsi.

1. **\*\*Fungsi `maxab`\*\*:**
  - Fungsi ini menerima dua parameter bertipe integer, `a` dan `b`, dan mengembalikan nilai maksimum dari kedua bilangan tersebut.
  - Fungsi menggunakan operator ternary untuk memeriksa apakah `a` lebih besar dari atau sama dengan `b`. Jika ya, fungsi mengembalikan nilai `a`, jika tidak, fungsi mengembalikan nilai `b`.
2. **\*\*Prosedur `tukar`\*\*:**
  - Prosedur ini menerima dua parameter bertipe integer, `a` dan `b`, dan bertujuan untuk menukar nilai kedua bilangan tersebut.
  - Variabel sementara `temp` digunakan untuk menyimpan nilai sementara saat pertukaran dilakukan.
  - Setelah pertukaran, prosedur mencetak nilai kedua bilangan yang telah ditukar.
3. **\*\*Program Utama\*\*:**
  - Program utama dimulai dengan mendeklarasikan dua variabel integer `a` dan `b`.
  - Pengguna diminta untuk memasukkan dua bilangan integer.
  - Setelah membaca kedua bilangan, program mencetak nilai kedua bilangan yang dimasukkan.
  - Program memanggil fungsi `maxab(a, b)` untuk menemukan nilai maksimum dari dua bilangan yang dimasukkan dan mencetak hasilnya.
  - Selanjutnya, program memanggil prosedur `tukar(a, b)` untuk menukar nilai kedua bilangan dan mencetak nilai kedua bilangan setelah pertukaran.

Dengan demikian, program ini memberikan contoh penggunaan prosedur dan fungsi dalam Java untuk menemukan nilai maksimum dari dua bilangan dan menukar nilai kedua bilangan.

## 19. Tempair.java

```
7  import java.util.Scanner;
8  /* contoh pemakaian IF tiga kasus : wujud air */
9
10 public class Tempair {
11     /**
12     * @param args
13     */
14
15     public static void main(String[] args) {
16         // TODO Auto-generated method stub
17
18         /* Kamus : */
19         int T;
20         Scanner masukan=new Scanner(System.in);
21
22         /* Program */
23         System.out.print ("Contoh IF tiga kasus \n");
24         System.out.print ("Temperatur (der. C) = ");
25         T=masukan.nextInt();
26         if (T < 0) {
27             System.out.print ("Wujud air beku \n"+ T);
28         }else if ((0 <= T) && (T <= 100)){
29             System.out.print ("Wujud air cair \n"+ T);
30         }else if (T > 100){
31             System.out.print ("Wujud air uap/gas \n"+ T);
32         };
33     }
34 }
```

### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>javac Tempair.java

C:\Users\ASUS\Documents\PBO\Tugas 3\Tugas3_SourceCode>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 77
Wujud air cair
77
```

Program `Tempair` adalah contoh penggunaan struktur percabangan `if-else if-else` untuk menentukan wujud air berdasarkan suhu yang dimasukkan pengguna dalam derajat Celsius.

1. `import java.util.Scanner;`: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca input dari pengguna.
2. `public class Tempair { ... }`: Mendefinisikan kelas `Tempair`, yang merupakan kelas utama dari program.
3. `public static void main(String[] args) { ... }`: Method utama (`main`) dari program. Program akan dimulai dari sini ketika dijalankan.
4. `int T;`: Mendeklarasikan variabel `T` yang akan digunakan untuk menyimpan suhu yang dimasukkan pengguna.

5. ``Scanner masukan=new Scanner(System.in);``: Membuat objek Scanner untuk menerima input dari pengguna melalui keyboard.
6. ``System.out.print ("Contoh IF tiga kasus \n");``: Menampilkan pesan untuk memberi tahu pengguna tentang jenis program yang dijalankan.
7. ``System.out.print ("Temperatur (der. C) = ");``: Menampilkan pesan meminta pengguna untuk memasukkan suhu dalam derajat Celsius.
8. ``T=masukan.nextInt();``: Menggunakan objek Scanner untuk membaca nilai integer yang dimasukkan pengguna dan menyimpannya dalam variabel ``T``.
9. ``if (T < 0) { ... }``: Memeriksa apakah suhu yang dimasukkan pengguna kurang dari 0. Jika benar, program akan mencetak bahwa air dalam wujud beku.
10. ``else if ((0 <= T) && (T <= 100)){ ... }``: Memeriksa apakah suhu berada di antara 0 dan 100. Jika benar, program akan mencetak bahwa air dalam wujud cair.
11. ``else if (T > 100){ ... }``: Memeriksa apakah suhu lebih dari 100. Jika benar, program akan mencetak bahwa air dalam wujud uap atau gas.
12. ``};``: Penutup dari struktur ``if-else if-else``.

Program ini mengambil input dari pengguna, yaitu suhu dalam derajat Celsius, dan kemudian menentukan wujud air berdasarkan suhu tersebut.