

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Muhammad Andi Syaifullah
Nim : 13020220015
Dosen : Mardiyah Hasnawi, S.Kom.,M.T.

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR

2024

• Program 1

-OutPut Program class BacaString Dan Penjelasan Programnya

C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java BacaString.java

```
Baca string dan Integer:  
masukkan sebuah string: 13  
String yang dibaca : 13
```

Tujuan Program:

Program ini memiliki tujuan untuk membaca sebuah string dari pengguna melalui konsol dan menampilkan string tersebut kembali ke pengguna. Hal ini membantu pemahaman tentang input/output dasar dalam bahasa Java.

Keyword dan Tipe Data:

1. `import java.io.BufferedReader;` : Mengimpor kelas `BufferedReader` dari paket `java.io`, digunakan untuk membaca input dari pengguna.
2. `import java.io.IOException;` : Mengimpor kelas `IOException` dari paket `java.io`, yang menangani pengecualian input/output.
3. `import java.io.InputStreamReader;` : Mengimpor kelas `InputStreamReader` dari paket `java.io`, membantu dalam membaca input stream.
4. `import javax.swing.*;` : Mengimpor seluruh kelas dari paket `javax.swing`, meskipun dalam program ini kelas ini tidak digunakan.
5. `public class BacaString {` : Mendeklarasikan kelas `BacaString`, yang berisi program utama.

Main Method:

```
public static void main(String[] args) throws  
IOException {
```

Metode utama `main`, yang merupakan titik masuk eksekusi program. Metode ini melemparkan pengecualian `IOException` karena membaca input dapat menimbulkan kesalahan.

Variable Declaration:

```
String str;
```

```
BufferedReader datAIn = new BufferedReader(new  
InputStreamReader(System.in));
```

1. `String str;`: Mendeklarasikan variabel `str` dengan tipe data `String`, yang akan digunakan untuk menyimpan string yang dibaca.
2. `BufferedReader datAIn = new BufferedReader(new InputStreamReader(System.in));`: Membuat objek `BufferedReader` bernama `datAIn` untuk membaca input dari pengguna. `InputStreamReader(System.in)` digunakan untuk mengonversi input stream dari `System.in` (konsol) ke karakter.

User Input:

```
System.out.print ("\nBaca string dan Integer:  
\n");  
System.out.print("masukkan sebuah string: ");  
str= datAIn.readLine();
```

1. `System.out.print ("\nBaca string dan Integer: \n");`: Menampilkan pesan ke konsol untuk memberi petunjuk kepada pengguna.
2. `System.out.print("masukkan sebuah string: ");`: Menampilkan pesan ke konsol untuk meminta pengguna memasukkan sebuah string.
3. `str= datAIn.readLine();`: Membaca string yang dimasukkan oleh pengguna menggunakan `BufferedReader` dan menyimpannya dalam variabel `str`.

Output:

```
System.out.print ("String yang dibaca : "+ str);
```

Menampilkan string yang telah dibaca dari pengguna ke konsol.

Program ini dirancang untuk membaca input string dari pengguna dan menampilkannya kembali. Dengan menggunakan

`BufferedReader`, program dapat mengatasi masalah pembacaan input yang lebih kompleks dibandingkan dengan `Scanner`.

• Program 2

-OutPut Program class ForEver Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java ForEver.java
Program akan looping, akhiri dengan ^c
Print satu baris .....
Print satu baris .....
Print satu baris .....
Print satu baris .....
^C
```

Tujuan Program:

Program ini dirancang untuk melakukan loop tanpa akhir (infinite loop) yang mencetak satu baris setiap detiknya. Tujuannya adalah untuk memberikan contoh penggunaan perulangan tanpa batas dan penggunaan penanganan pengecualian (`InterruptedException`) untuk menghentikan program dengan penekanan tombol ^c (Ctrl + C).

Keyword dan Tipe Data:

1. `public class ForEver {` : Mendeklarasikan kelas `ForEver`, yang berisi program utama.
2. `public static void main(String[] args) {` : Mendeklarasikan metode utama `main`, titik awal eksekusi program.
3. `System.out.println("Program akan looping, akhiri dengan ^c");` : Menampilkan pesan ke konsol untuk memberi tahu pengguna cara menghentikan program.
4. `try { ... } catch (InterruptedException e) { ... }` : Menggunakan blok try-catch untuk menangani pengecualian `InterruptedException`.
5. `while (true) { ... }` : Membuat loop tanpa akhir.

Program Loop:

```

try {
    while (true) {
        System.out.print("Print satu baris .....
\n");
        Thread.sleep(1000);
    }
} catch (InterruptedException e) {
    System.out.println("Program dihentikan.");
}

```

1. `while (true) { ... }`: Membuat loop tanpa akhir.
2. `System.out.print("Print satu baris \n");`: Mencetak satu baris ke konsol setiap detik.
3. `Thread.sleep(1000);`: Memberikan jeda (pause) selama 1000 milidetik (1 detik) menggunakan `Thread.sleep(1000)`.
4. Blok `catch (InterruptedException e) { ... }`: Menangkap pengecualian `InterruptedException` yang dapat terjadi saat menggunakan `Thread.sleep`. Program memberikan pesan "Program dihentikan" ketika pengecualian terjadi.

Output:

Program akan mencetak pesan "Program akan looping, akhiri dengan ^c" ke konsol dan kemudian memasuki loop tanpa akhir yang mencetak "Print satu baris" setiap detiknya. Program dapat dihentikan dengan menekan tombol ^c, yang akan menimbulkan pengecualian `InterruptedException` dan menampilkan pesan "Program dihentikan."

Program ini memberikan contoh sederhana penggunaan loop tanpa akhir dan cara menghentikan program secara manual menggunakan pengecualian.

• Program 3

-OutPut Program class If1 Dan Penjelasan Programnya

C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java If1.java
Contoh IF satu kasus
Ketikkan suatu nilai integer : 13
Nilai a positif 13

Tujuan Program:

Program ini memiliki tujuan untuk memperkenalkan konsep dasar penggunaan pernyataan kondisional ``if`` dalam bahasa pemrograman Java. Tujuannya adalah untuk meminta pengguna memasukkan nilai integer, kemudian memeriksa apakah nilai tersebut non-negatif (lebih besar atau sama dengan 0). Jika benar, program akan mencetak pesan bahwa nilai tersebut positif.

Keyword dan Tipe Data:

1. ``import java.util.Scanner;``: Mengimpor kelas ``Scanner`` dari paket ``java.util``, yang digunakan untuk membaca input dari pengguna.
2. ``public class If1 {``: Mendeklarasikan kelas ``If1``, yang berisi program utama.
3. ``public static void main(String[] args) {``: Mendeklarasikan metode utama ``main``, titik awal eksekusi program.
4. ``Scanner masukan=new Scanner(System.in);``: Membuat objek ``Scanner`` bernama ``masukan`` untuk membaca input dari pengguna.
5. ``int a;``: Mendeklarasikan variabel ``a`` dengan tipe data ``int`` untuk menyimpan nilai integer yang dimasukkan oleh pengguna.

Program Input dan Kondisional:

```
System.out.print ("Contoh IF satu kasus \n");  
System.out.print ("Ketikkan suatu nilai integer :  
");  
a = masukan.nextInt();  
if (a >= 0)
```

```
System.out.print ("\nNilai a positif "+ a);
```

1. `System.out.print ("Contoh IF satu kasus \n");`: Menampilkan pesan ke konsol untuk memberikan petunjuk kepada pengguna.
2. `System.out.print ("Ketikkan suatu nilai integer : ");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai integer.
3. `a = masukan.nextInt();`: Membaca nilai integer dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `a`.
4. `if (a >= 0)`: Pernyataan kondisional `if` yang memeriksa apakah nilai `a` lebih besar atau sama dengan 0.
5. `System.out.print ("\nNilai a positif "+ a);`: Jika kondisi `if` benar, program akan mencetak pesan bahwa nilai `a` adalah positif.

Output:

Program akan menampilkan pesan instruksi, meminta pengguna memasukkan nilai integer, dan kemudian memberikan informasi apakah nilai tersebut positif. Jika nilai yang dimasukkan non-negatif, maka program akan mencetak pesan bahwa nilai tersebut positif.

Program ini memberikan contoh sederhana penggunaan pernyataan kondisional `if` untuk melakukan pengujian sederhana dan memberikan output berdasarkan kondisi yang terpenuhi atau tidak terpenuhi.

• Program 4

-OutPut Program class If2 Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java If2.java
Contoh IF dua kasus
Ketikkan suatu nilai integer :14
Nilai a positif 14
```



Tujuan Program:

Program ini bertujuan untuk memperkenalkan konsep penggunaan pernyataan kondisional `if-else` dalam bahasa pemrograman.

Java. Tujuannya adalah meminta pengguna memasukkan nilai integer, dan kemudian memeriksa apakah nilai tersebut non-negatif atau negatif. Program akan mencetak pesan tergantung pada kondisi nilai `a`.

Keyword dan Tipe Data:

1. `import java.util.Scanner;` : Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class If2 {` : Mendeklarasikan kelas `If2`, yang berisi program utama.
3. `public static void main(String[] args) {` : Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `int a;` : Mendeklarasikan variabel `a` dengan tipe data `int` untuk menyimpan nilai integer yang dimasukkan oleh pengguna.
5. `Scanner masukan=new Scanner(System.in);` : Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Input dan Kondisional:

```
System.out.print ("Contoh IF dua kasus \n");
System.out.print ("Ketikkan suatu nilai integer
:");
a=masukan.nextInt();
if (a >= 0){
    System.out.println ("Nilai a positif "+ a);
}else{
    System.out.println ("Nilai a negatif "+ a);
}
```


1. `System.out.print ("Contoh IF dua kasus \n");`: Menampilkan pesan ke konsol untuk memberikan petunjuk kepada pengguna.
2. `System.out.print ("Ketikkan suatu nilai integer :");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai integer.
3. `a=masukan.nextInt();`: Membaca nilai integer dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `a`.
4. `if (a >= 0){`: Pernyataan kondisional `if` yang memeriksa apakah nilai `a` lebih besar atau sama dengan 0.
5. `System.out.println ("Nilai a positif "+ a);`: Jika kondisi `if` benar, program akan mencetak pesan bahwa nilai `a` adalah positif.
6. `else { ... }`: Bagian `else` dari pernyataan kondisional yang dijalankan jika kondisi `if` tidak terpenuhi.
7. `System.out.println ("Nilai a negatif "+ a);`: Jika kondisi `if` tidak terpenuhi, program akan mencetak pesan bahwa nilai `a` adalah negatif.

Output:

Program akan menampilkan pesan instruksi, meminta pengguna memasukkan nilai integer, dan kemudian memberikan informasi apakah nilai tersebut positif atau negatif. Program ini menggunakan struktur `if-else` untuk menentukan tindakan yang akan diambil berdasarkan kondisi.

Program ini memberikan contoh penggunaan pernyataan kondisional `if-else` untuk melakukan pengujian sederhana dan memberikan output berdasarkan kondisi yang terpenuhi atau tidak terpenuhi.

• Program 5

-OutPut Program class If3 Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java If3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :15
Nilai a positif 15
```



Tujuan Program:

Program ini bertujuan untuk memperkenalkan konsep penggunaan pernyataan kondisional `if-else` dalam bahasa pemrograman Java. Tujuannya adalah meminta pengguna memasukkan nilai integer, dan kemudian memeriksa apakah nilai tersebut positif, nol, atau negatif. Program akan mencetak pesan tergantung pada kondisi nilai `a`.

Keyword dan Tipe Data:

1. `import java.util.Scanner;` : Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class If3 {` : Mendeklarasikan kelas `If3`, yang berisi program utama.
3. `public static void main(String[] args) {` : Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `int a;` : Mendeklarasikan variabel `a` dengan tipe data `int` untuk menyimpan nilai integer yang dimasukkan oleh pengguna.
5. `Scanner masukan=new Scanner(System.in);` : Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Input dan Kondisional:

```
System.out.print ("Contoh IF tiga kasus \n");
System.out.print ("Ketikkan suatu nilai integer
:");
a=masukan.nextInt();
if (a > 0){
    System.out.println ("Nilai a positif "+ a);
}else if (a == 0){
    System.out.println ("Nilai Nol "+ a);
}else {
    System.out.println ("Nilai a negatif "+ a);
}
```

}

1. `System.out.print ("Contoh IF tiga kasus \n");`: Menampilkan pesan ke konsol untuk memberikan petunjuk kepada pengguna.
2. `System.out.print ("Ketikkan suatu nilai integer :");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai integer.
3. `a=masukan.nextInt();`: Membaca nilai integer dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `a`.
4. `if (a > 0){`: Pernyataan kondisional pertama yang memeriksa apakah nilai `a` lebih besar dari 0.
5. `System.out.println ("Nilai a positif "+ a);`: Jika kondisi pertama terpenuhi, program akan mencetak pesan bahwa nilai `a` adalah positif.
6. `else if (a == 0){`: Pernyataan kondisional kedua yang memeriksa apakah nilai `a` sama dengan 0.
7. `System.out.println ("Nilai Nol "+ a);`: Jika kondisi kedua terpenuhi, program akan mencetak pesan bahwa nilai `a` adalah nol.
8. `else { ... }`: Bagian `else` dari pernyataan kondisional yang dijalankan jika kedua kondisi sebelumnya tidak terpenuhi.
9. `System.out.println ("Nilai a negatif "+ a);`: Jika kondisi `else` terpenuhi, program akan mencetak pesan bahwa nilai `a` adalah negatif.

Output:

Program ini akan menampilkan pesan instruksi, meminta pengguna memasukkan nilai integer, dan kemudian memberikan informasi apakah nilai tersebut positif, nol, atau negatif. Program menggunakan struktur `if-else if-else` untuk menentukan tindakan yang akan diambil berdasarkan kondisi.

Program ini memberikan contoh penggunaan pernyataan kondisional `if-else if-else` untuk melakukan pengujian sederhana dan

memberikan output berdasarkan kondisi yang terpenuhi atau tidak terpenuhi.

• Program 6

-OutPut Program class KasusBoolean Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java KasusBoolean.java  
true  
benar
```

Tujuan Program:

Program ini bertujuan untuk memberikan pemahaman tentang penggunaan tipe data boolean dan operasi logika dalam bahasa pemrograman Java. Program tersebut mendeklarasikan variabel boolean `bool`, memberikan nilai `true` kepadanya, dan kemudian mengevaluasi dan mencetak beberapa pernyataan berdasarkan kondisi boolean.

Keyword dan Tipe Data:

1. `public class KasusBoolean {`: Mendeklarasikan kelas `KasusBoolean`, yang berisi program utama.
2. `public static void main(String[] args) {`: Mendeklarasikan metode utama `main`, titik awal eksekusi program.
3. `boolean bool;`: Mendeklarasikan variabel `bool` dengan tipe data `boolean`.

Program Logika Boolean:

```
bool = true;  
if (bool) {  
    System.out.print("true\n");  
} else {  
    System.out.print("false\n");  
}  
if (!bool) {
```

```
System.out.print("salah\n");  
} else {  
System.out.print("benar\n");
```

1. `bool = true;`: Memberikan nilai `true` ke variabel boolean `bool`.
2. `if (bool) { ... } else { ... }`: Pernyataan kondisional yang mengevaluasi apakah nilai `bool` adalah `true`. Jika `true`, maka blok pertama akan dieksekusi; jika tidak, blok kedua yang dieksekusi.
3. `System.out.print("true\n");`: Mencetak "true" ke konsol jika nilai `bool` adalah `true`.
4. `System.out.print("false\n");`: Mencetak "false" ke konsol jika nilai `bool` adalah `false`.
5. `if (!bool) { ... } else { ... }`: Pernyataan kondisional yang mengevaluasi apakah nilai `bool` tidak sama dengan `true` (negasi dari nilai `bool`). Jika `true`, maka blok pertama akan dieksekusi; jika tidak, blok kedua yang dieksekusi.
6. `System.out.print("salah\n");`: Mencetak "salah" ke konsol jika nilai `bool` tidak sama dengan `true`.
7. `System.out.print("benar\n");`: Mencetak "benar" ke konsol jika nilai `bool` sama dengan `true`.

Output:

Program ini akan mencetak hasil berdasarkan nilai variabel boolean `bool`. Jika `bool` adalah `true`, maka program akan mencetak "true" dan "benar". Jika `bool` adalah `false`, maka program akan mencetak "false" dan "salah".

Program ini memberikan contoh penggunaan tipe data boolean dan operasi logika seperti negasi (!) dalam konteks pernyataan kondisional.

• Program 7

-OutPut Program class KasusSwitch Dan Penjelasan Programnya

C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
benar
Yang anda ketik adalah huruf mati

Tujuan Program:

Program ini bertujuan untuk memberikan pemahaman tentang penggunaan pernyataan `switch` dalam bahasa pemrograman Java. Program meminta pengguna memasukkan sebuah huruf dan mengevaluasi input tersebut menggunakan pernyataan `switch`. Berdasarkan huruf yang dimasukkan, program akan memberikan output sesuai dengan kasus yang cocok.

Keyword dan Tipe Data:

1. `import java.util.Scanner;`: Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class KasusSwitch {`: Mendeklarasikan kelas `KasusSwitch`, yang berisi program utama.
3. `public static void main(String[] args) {`: Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `char cc;`: Mendeklarasikan variabel `cc` dengan tipe data `char` untuk menyimpan karakter yang dimasukkan oleh pengguna.
5. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Pernyataan Switch:

```
System.out.print ("Ketikkan sebuah huruf, akhiri  
dengan RETURN \n");  
cc = masukan.next().charAt(0);  
switch (cc) {  
    case 'a': {
```

```

        System.out.print (" Yang anda ketik adalah
a\n");
        break;
    }
    case 'u': {
        System.out.print (" Yang anda ketik adalah
u\n");
        break;
    }
    case 'e': {
        System.out.print (" Yang anda ketik adalah
e\n");
        break;
    }
    case 'i': {
        System.out.print (" Yang anda ketik adalah
i\n");
        break;
    }
    case 'o': {
        System.out.print (" Yang anda ketik adalah
o\n");
        break;
    }
    default:
        System.out.print (" Yang anda ketik adalah
huruf mati\n");
}

```

1. `System.out.print ("Ketikkan sebuah huruf, akhiri dengan RETURN \n");`: Menampilkan pesan instruksi ke konsol untuk meminta pengguna memasukkan sebuah huruf.
2. `cc = masukan.next().charAt(0);`: Membaca karakter yang dimasukkan oleh pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `cc`.
3. `switch (cc) { ... }`: Pernyataan `switch` yang mengevaluasi nilai dari variabel `cc`.

4. ``case 'a': { ... }``: Kasus pertama yang dieksekusi jika nilai ``cc`` sama dengan `'a'`.
5. ``System.out.print (" Yang anda ketik adalah a\n");``: Mencetak pesan ke konsol jika kasus `'a'` terpenuhi.
6. ``break;``: Menghentikan eksekusi ``switch`` setelah kasus `'a'` terpenuhi.
7. Sama seperti kasus `'a'`, program memiliki kasus untuk `'u'`, `'e'`, `'i'`, dan `'o'`, masing-masing dengan pernyataan yang sesuai.
8. ``default: System.out.print (" Yang anda ketik adalah huruf mati\n");``: Bagian default yang dieksekusi jika tidak ada kasus yang cocok dengan nilai ``cc``.

Output:


Program ini akan menampilkan pesan instruksi, meminta pengguna memasukkan huruf, dan kemudian memberikan informasi tentang huruf yang dimasukkan. Kasus `'a'`, `'u'`, `'e'`, `'i'`, dan `'o'` akan memberikan output sesuai, sedangkan huruf lainnya akan masuk ke kasus default.

Program ini memberikan contoh penggunaan pernyataan ``switch`` untuk menentukan tindakan yang akan diambil berdasarkan nilai variabel ``cc``.

• Program 8

-OutPut Program class Konstant Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java Konstant.java
Jari-jari lingkaran =23
Luas lingkaran = 1661.8535
Akhir program
```



Tujuan Program:

Program ini bertujuan untuk menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan oleh pengguna. Program ini juga menggunakan konstanta (`final float PHI``) untuk menyimpan

nilai π (Pi) sebagai konstanta yang tidak dapat diubah selama eksekusi program.

Keyword dan Tipe Data:

1. `import java.util.Scanner;` : Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class Konstant` { : Mendeklarasikan kelas `Konstant`, yang berisi program utama.
3. `public static void main(String[] args)` { : Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `final float PHI = 3.1415f;` : Mendeklarasikan konstanta `PHI` dengan tipe data `float`. Nilai konstanta ini tidak dapat diubah selama eksekusi program.
5. `float r;` : Mendeklarasikan variabel `r` dengan tipe data `float` untuk menyimpan jari-jari lingkaran.
6. `Scanner masukan=new Scanner(System.in);` : Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Menghitung Luas Lingkaran:

```
System.out.print ("Jari-jari lingkaran =");  
r = masukan.nextFloat();  
System.out.print ("Luas lingkaran = "+ (PHI * r *  
r)+"\n");  
System.out.print ("Akhir program \n");
```

1. `System.out.print ("Jari-jari lingkaran =");` : Menampilkan pesan instruksi ke konsol untuk meminta pengguna memasukkan jari-jari lingkaran.
2. `r = masukan.nextFloat();` : Membaca nilai jari-jari dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `r`.

3. ``System.out.print ("Luas lingkaran = "+ (PHI * r * r)+"\n");``: Menghitung luas lingkaran dengan menggunakan rumus πr^2 dan mencetak hasilnya ke konsol.

4. ``System.out.print ("Akhir program \n");``: Menampilkan pesan akhir ke konsol.

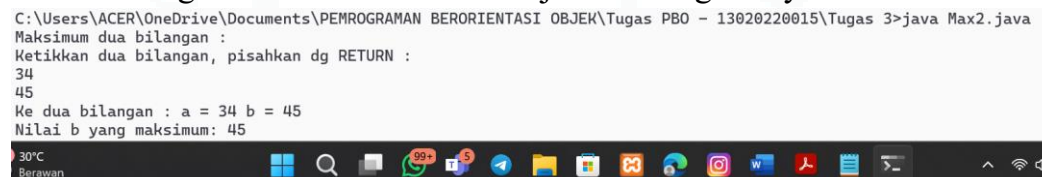
Output:

Program ini akan meminta pengguna memasukkan nilai jari-jari lingkaran, menghitung luas lingkaran, dan menampilkan hasilnya ke konsol. Nilai π (Pi) diambil dari konstanta ``PHI``.

Program ini memberikan contoh penggunaan konstanta untuk menyimpan nilai yang tetap dan tidak dapat diubah selama eksekusi program, serta penggunaan rumus matematika sederhana untuk menghitung luas lingkaran.

• Program 9

-OutPut Program class Max2 Dan Penjelasan Programnya



```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java Max2.java
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
34
45
Ke dua bilangan : a = 34 b = 45
Nilai b yang maksimum: 45
```

Tujuan Program:

Program ini bertujuan untuk meminta pengguna memasukkan dua bilangan, dan kemudian menentukan nilai maksimum di antara kedua bilangan tersebut. Program menggunakan struktur kondisional ``if-else`` untuk membandingkan nilai ``a`` dan ``b`` dan mencetak nilai maksimum ke konsol.

Keyword dan Tipe Data:

1. ``import java.util.Scanner;``: Mengimpor kelas ``Scanner`` dari paket ``java.util``, yang digunakan untuk membaca input dari pengguna.

2. `public class Max2 {`: Mendeklarasikan kelas `Max2`, yang berisi program utama.
3. `public static void main(String[] args) {`: Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `int a, b;`: Mendeklarasikan dua variabel `a` dan `b` dengan tipe data `int` untuk menyimpan dua bilangan yang dimasukkan oleh pengguna.
5. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Menentukan Nilai Maksimum:

```
System.out.print ("Maksimum dua bilangan : \n");
System.out.print ("Ketikkan dua bilangan, pisahkan
dg RETURN : \n");
a=masukan.nextInt(); b=masukan.nextInt();
System.out.println ("Ke dua bilangan : a = "+ a+"
b = "+b);
if (a >= b){
    System.out.println ("Nilai a yang maksimum "+
a);
} else {
    System.out.println ("Nilai b yang maksimum: "+
b);
}
```

1. `System.out.print ("Maksimum dua bilangan : \n");`: Menampilkan pesan ke konsol untuk memberi tahu pengguna tujuan program.
2. `System.out.print ("Ketikkan dua bilangan, pisahkan dg RETURN : \n");`: Menampilkan pesan instruksi ke konsol untuk meminta pengguna memasukkan dua bilangan, dipisahkan oleh RETURN.
3. `a=masukan.nextInt(); b=masukan.nextInt();`: Membaca dua bilangan dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `a` dan `b`.
4. `System.out.println ("Ke dua bilangan : a = "+ a+" b = "+b);`: Menampilkan nilai bilangan yang dimasukkan oleh pengguna ke konsol.

5. `if (a >= b) { ... } else { ... }`: Pernyataan kondisional `if-else` yang membandingkan nilai `a` dan `b`.

6. `System.out.println ("Nilai a yang maksimum "+ a);`: Jika `a` lebih besar atau sama dengan `b`, program akan mencetak bahwa nilai maksimum adalah `a`.

7. `System.out.println ("Nilai b yang maksimum: "+ b);`: Jika `a` kurang dari `b`, program akan mencetak bahwa nilai maksimum adalah `b`.

Output:

Program ini akan meminta pengguna memasukkan dua bilangan, membandingkannya, dan kemudian mencetak nilai maksimum ke konsol. Jika kedua bilangan sama, program akan memilih salah satu sebagai nilai maksimum.

Program ini memberikan contoh penggunaan pernyataan kondisional `if-else` untuk menentukan tindakan yang akan diambil berdasarkan kondisi yang terpenuhi atau tidak terpenuhi.

• Program 10

-OutPut Program class PriFor Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java PriFor.java
Baca N, print 1 s/d N N = 3
1
2
3
Akhir program
```



Tujuan Program:

Program ini bertujuan untuk membaca nilai N dari pengguna, kemudian mencetak angka dari 1 hingga N ke konsol. Program menggunakan pernyataan perulangan `for` untuk mencapai tujuan ini.

Keyword dan Tipe Data:

1. `import java.util.Scanner;`: Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.

2. `public class PriFor {`: Mendeklarasikan kelas `PriFor`, yang berisi program utama.
3. `public static void main(String[] args) {`: Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `int i, N;`: Mendeklarasikan dua variabel `i` dan `N` dengan tipe data `int`. `i` digunakan sebagai variabel iterasi, dan `N` untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Looping dan Cetak Angka:

```
System.out.print ("Baca N, print 1 s/d N ");
System.out.print ("N = ");
N=masukan.nextInt();
for (i = 1; i <= N; i++){
    System.out.println (i);
}
System.out.println ("Akhir program \n");
```

1. `System.out.print ("Baca N, print 1 s/d N ");`: Menampilkan pesan instruksi ke konsol untuk memberitahu pengguna tujuan program.
2. `System.out.print ("N = ");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai `N`.
3. `N=masukan.nextInt();`: Membaca nilai `N` dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `N`.
4. `for (i = 1; i <= N; i++){ ... }`: Pernyataan perulangan `for` yang mencetak angka dari 1 hingga `N`. Variabel `i` digunakan sebagai variabel iterasi, dan `i++` digunakan untuk meningkatkannya setiap kali iterasi dilakukan.
5. `System.out.println (i);`: Mencetak nilai `i` ke konsol pada setiap iterasi.
6. `System.out.println ("Akhir program \n");`: Menampilkan pesan akhir ke konsol setelah selesai mencetak angka.

Output:

-

Program ini akan meminta pengguna memasukkan nilai N, kemudian mencetak angka dari 1 hingga N ke konsol. Setelah selesai mencetak, program akan menampilkan pesan akhir.

Program ini memberikan contoh penggunaan pernyataan perulangan `for` untuk melakukan tugas yang berulang dengan iterasi yang jelas, yaitu mencetak angka dari 1 hingga N.

• Program 11

-OutPut Program class PrintIterasi Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java PrintIterasi.java
Nilai N >0 = 3
Print i dengan ITERATE :
1
2
3
```

Tujuan Program:

Program ini bertujuan untuk membaca nilai N dari pengguna, kemudian mencetak angka dari 1 hingga N ke konsol. Program menggunakan pernyataan perulangan `for` dengan bentuk "for (;;)" yang bersifat tak terbatas dan menggunakan instruksi `break` untuk menghentikan perulangan ketika nilai `i` sama dengan `N`.

Keyword dan Tipe Data:

1. `import java.util.Scanner;`: Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class PrintIterasi {`: Mendeklarasikan kelas `PrintIterasi`, yang berisi program utama.
3. `public static void main(String[] args) {`: Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `int N;`: Mendeklarasikan variabel `N` dengan tipe data `int` untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. `int i;`: Mendeklarasikan variabel `i` dengan tipe data `int` untuk menyimpan nilai iterasi.

Program Looping dan Cetak Angka:

```
``java
System.out.print ("Nilai N >0 = ");
N = masukan.nextInt();
i = 1;
System.out.print ("Print i dengan ITERATE : \n");
for (;;) {
    System.out.println(i);
    if (i == N)
        break;
    else {
        i++;
    }
}
```

1. `System.out.print ("Nilai N >0 = ");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai N yang lebih besar dari 0.
2. `N = masukan.nextInt();`: Membaca nilai N dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `N`.
3. `i = 1;`: Menginisialisasi variabel iterasi `i` dengan nilai 1.
4. `System.out.print ("Print i dengan ITERATE : \n");`: Menampilkan pesan instruksi ke konsol.
5. `for (;;) { ... }`: Pernyataan perulangan `for` yang tidak memiliki kondisi terminasi dan akan dihentikan menggunakan instruksi `break`.
6. `System.out.println(i);`: Mencetak nilai `i` ke konsol pada setiap iterasi.
7. `if (i == N) break;`: Pernyataan kondisional yang menggunakan instruksi `break` untuk menghentikan perulangan jika nilai `i` sama dengan `N`.
8. `else { i++; }`: Jika kondisi di atas tidak terpenuhi, nilai `i` akan diinkrementasi untuk mencapai nilai berikutnya.

Output:

Program ini akan meminta pengguna memasukkan nilai N, kemudian mencetak angka dari 1 hingga N ke konsol. Program menggunakan


perulangan tak terbatas dan menggunakan instruksi `break` untuk menghentikan perulangan ketika nilai `i` sama dengan `N`.

Program ini memberikan contoh penggunaan perulangan tak terbatas dengan menggunakan instruksi `break` untuk menghentikan perulangan pada kondisi tertentu.

• Program 12

-OutPut Program class PrintRepeat Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java PrintRepeat.java
Nilai N >0 = 3
Print i dengan REPEAT:
1
2
3
```



Tujuan Program:

Program ini bertujuan untuk membaca nilai N dari pengguna, kemudian mencetak angka dari 1 hingga N ke konsol. Program menggunakan pernyataan perulangan `do-while` untuk mencapai tujuan ini.

Keyword dan Tipe Data:

1. `import java.util.Scanner;`: Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class PrintRepeat {`: Mendeklarasikan kelas `PrintRepeat`, yang berisi program utama.
3. `public static void main(String[] args) {`: Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `int N;`: Mendeklarasikan variabel `N` dengan tipe data `int` untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. `int i;`: Mendeklarasikan variabel `i` dengan tipe data `int` untuk menyimpan nilai iterasi.
6. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Looping dan Cetak Angka:


```

System.out.print ("Nilai N >0 = ");
N = masukan.nextInt();
i = 1;
System.out.print ("Print i dengan REPEAT: \n");
do {
    System.out.print (i+"\n");
    i++;
} while (i <= N);

```

1. `System.out.print ("Nilai N >0 = ");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai N yang lebih besar dari 0.
2. `N = masukan.nextInt();`: Membaca nilai N dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `N`.
3. `i = 1;`: Menginisialisasi variabel iterasi `i` dengan nilai 1.
4. `System.out.print ("Print i dengan REPEAT: \n");`: Menampilkan pesan instruksi ke konsol.
5. `do { ... } while (i <= N);`: Pernyataan perulangan `do-while` yang mencetak angka dari 1 hingga N.
6. `System.out.print (i+"\n");`: Mencetak nilai `i` ke konsol pada setiap iterasi.
7. `i++;`: Menambahkan nilai `i` untuk mencapai nilai berikutnya.
8. `while (i <= N);`: Kondisi pengulangan yang menentukan apakah perulangan akan dilanjutkan berdasarkan nilai `i` dan `N`.

Output:

Program ini akan meminta pengguna memasukkan nilai N, kemudian mencetak angka dari 1 hingga N ke konsol. Program menggunakan pernyataan perulangan `do-while` yang memastikan setidaknya satu iterasi dilakukan sebelum mengevaluasi kondisi.

Program ini memberikan contoh penggunaan pernyataan perulangan `do-while`, yang berguna ketika Anda ingin memastikan bahwa blok pernyataan di dalamnya dieksekusi setidaknya sekali, bahkan jika kondisi pengulangan awalnya salah.

• Program 13

-OutPut Program class PrintWhile Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java PrintWhile.java
Nilai N >0 = 5
Print i dengan WHILE:
1
2
3
4
5
```

Tujuan Program:

Program ini bertujuan untuk membaca nilai N dari pengguna, kemudian mencetak angka dari 1 hingga N ke konsol. Program menggunakan pernyataan perulangan `while` untuk mencapai tujuan ini.

Keyword dan Tipe Data:

1. `import java.util.Scanner;`: Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class PrintWhile {`: Mendefinisikan kelas `PrintWhile`, yang berisi program utama.
3. `public static void main(String[] args) {`: Mendefinisikan metode utama `main`, titik awal eksekusi program.
4. `int N;`: Mendefinisikan variabel `N` dengan tipe data `int` untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. `int i;`: Mendefinisikan variabel `i` dengan tipe data `int` untuk menyimpan nilai iterasi.
6. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Looping dan Cetak Angka:

```
System.out.print ("Nilai N >0 = ");
N = masukan.nextInt();
i = 1;
System.out.print ("Print i dengan WHILE: \n");
while (i <= N)
{
```

```
System.out.println (i);  
i++;  
};
```

1. `System.out.print ("Nilai N >0 = ");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai N yang lebih besar dari 0.
2. `N = masukan.nextInt();`: Membaca nilai N dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `N`.
3. `i = 1;`: Menginisialisasi variabel iterasi `i` dengan nilai 1.
4. `System.out.print ("Print i dengan WHILE: \n");`: Menampilkan pesan instruksi ke konsol.
5. `while (i <= N) { ... };`: Pernyataan perulangan `while` yang mencetak angka dari 1 hingga N.
6. `System.out.println (i);`: Mencetak nilai `i` ke konsol pada setiap iterasi.
7. `i++;`: Menambahkan nilai `i` untuk mencapai nilai berikutnya.
8. `};`: Kurung kurawal penutup untuk menandakan akhir dari blok pernyataan dalam perulangan `while`.

Output:


Program ini akan meminta pengguna memasukkan nilai N, kemudian mencetak angka dari 1 hingga N ke konsol. Program menggunakan pernyataan perulangan `while` yang mengeksekusi blok pernyataan selama kondisi yang diberikan benar.

Program ini memberikan contoh penggunaan pernyataan perulangan `while` untuk melakukan tugas yang berulang selama kondisi tertentu terpenuhi.

• Program 14

-OutPut Program class PrintWhile1 Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PB0 - 13020220015\Tugas 3>java PrintWhile1.java  
Nilai N >0 = 6  
Print i dengan WHILE (ringkas):  
1  
2  
3  
4  
5  
6
```



Tujuan Program:

Program ini bertujuan untuk membaca nilai N dari pengguna, kemudian mencetak angka dari 1 hingga N ke konsol. Program menggunakan pernyataan perulangan `while` dengan pendekatan ringkas untuk mencapai tujuan ini.

Keyword dan Tipe Data:

1. `import java.util.Scanner;`: Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class PrintWhile1 {`: Mendeklarasikan kelas `PrintWhile1`, yang berisi program utama.
3. `public static void main(String[] args) {`: Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `int N;`: Mendeklarasikan variabel `N` dengan tipe data `int` untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. `int i = 1;`: Mendeklarasikan dan menginisialisasi variabel `i` dengan nilai 1 sebagai inisialisasi awal iterasi.
6. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Looping dan Cetak Angka:

```
System.out.print ("Nilai N >0 = ");
N = masukan.nextInt();
System.out.print ("Print i dengan WHILE (ringkas):
\n");
while (i <= N){
    System.out.println (i++);
}
```

1. `System.out.print ("Nilai N >0 = ");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai N yang lebih besar dari 0.

2. `'N = masukan.nextInt();'`: Membaca nilai N dari pengguna menggunakan objek `'Scanner'` dan menyimpannya dalam variabel `'N'`.
3. `'System.out.print ("Print i dengan WHILE (ringkas): \n");'`: Menampilkan pesan instruksi ke konsol.
4. `'while (i <= N){ ... }'`: Pernyataan perulangan `'while'` yang mencetak angka dari 1 hingga N dengan pendekatan ringkas.
5. `'System.out.println (i++);'`: Mencetak nilai `'i'` ke konsol pada setiap iterasi dan secara bersamaan meningkatkan nilai `'i'` dengan operator `'++'`.

Output:

Program ini akan meminta pengguna memasukkan nilai N, kemudian mencetak angka dari 1 hingga N ke konsol. Program menggunakan pernyataan perulangan `'while'` dengan pendekatan ringkas untuk melakukan tugas ini.

Program ini memberikan contoh penggunaan pernyataan perulangan `'while'` dengan pendekatan ringkas menggunakan operator `'++'` untuk mencetak dan meningkatkan nilai variabel iterasi dalam satu pernyataan.

• Program 15

-OutPut Program class PrintXinterasi Dan Penjelasan Programnya

C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java PrintXinterasi.java

```
Masukkan nilai x (int), akhiri dg 999: 34
Masukkan nilai x (int),akhiri dg 999 : 999
Hasil penjumlahan = 34
```



Tujuan Program:

Program ini bertujuan untuk menghitung jumlah nilai yang dimasukkan pengguna (bilangan bulat) sampai pengguna memasukkan angka 999. Program menggunakan perulangan tak terbatas dan menggunakan instruksi `'break'` untuk menghentikan perulangan ketika pengguna memasukkan angka 999.

Keyword dan Tipe Data:

1. ``import java.util.Scanner;``: Mengimpor kelas ``Scanner`` dari paket ``java.util``, yang digunakan untuk membaca input dari pengguna.
2. ``public class PrintXinterasi {``: Mendeklarasikan kelas ``PrintXinterasi``, yang berisi program utama.
3. ``public static void main(String[] args) {``: Mendeklarasikan metode utama ``main``, titik awal eksekusi program.
4. ``int Sum=0;``: Mendeklarasikan variabel ``Sum`` dengan tipe data ``int`` untuk menyimpan jumlah nilai yang dimasukkan pengguna.
5. ``int x;``: Mendeklarasikan variabel ``x`` dengan tipe data ``int`` untuk menyimpan nilai yang dimasukkan pengguna setiap iterasi.
6. ``Scanner masukan=new Scanner(System.in);``: Membuat objek ``Scanner`` bernama ``masukan`` untuk membaca input dari pengguna.

Program Looping dan Penjumlahan Nilai:

```
System.out.print ("Masukkan nilai x (int), akhiri  
dg 999: ");  
x = masukan.nextInt();  
if (x == 999){  
    System.out.print ("Kasus kosong \n");  
}else{  
    Sum = x;  
    for (;;){  
        System.out.print ("Masukkan nilai x (int),  
akhiri dg 999 : ");  
        x = masukan.nextInt();  
        if(x==999)  
            break;  
        else{  
            Sum = Sum + x;  
        }  
    }  
}  
System.out.println("Hasil penjumlahan = "+ Sum);
```

1. ``System.out.print ("Masukkan nilai x (int), akhiri dg 999: ");``: Menampilkan pesan untuk meminta pengguna memasukkan nilai ``x`` (bilangan bulat) dan mengakhiri jika pengguna memasukkan 999.
2. ``x = masukan.nextInt();``: Membaca nilai ``x`` dari pengguna menggunakan objek ``Scanner`` dan menyimpannya dalam variabel ``x``.
3. ``if (x == 999){ ... }``: Pernyataan kondisional yang mengecek apakah pengguna langsung memasukkan angka 999 tanpa memasukkan nilai ``x``.
4. ``System.out.print ("Kasus kosong \n");``: Menampilkan pesan bahwa kasusnya kosong jika pengguna langsung memasukkan 999.
5. ``else { ... }``: Blok else yang dijalankan jika pengguna memasukkan nilai ``x`` selain 999.
6. ``Sum = x;``: Menginisialisasi variabel ``Sum`` dengan nilai ``x``.
7. ``for (;;) { ... }``: Pernyataan perulangan tak terbatas yang akan dihentikan menggunakan instruksi ``break`` ketika pengguna memasukkan 999.
8. ``System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");``: Menampilkan pesan untuk meminta pengguna memasukkan nilai ``x`` dan mengakhiri jika pengguna memasukkan 999.
9. ``x = masukan.nextInt();``: Membaca nilai ``x`` dari pengguna menggunakan objek ``Scanner`` dan menyimpannya dalam variabel ``x``.
10. ``if (x==999) break;``: Pernyataan kondisional yang menggunakan instruksi ``break`` untuk menghentikan perulangan jika pengguna memasukkan 999.
11. ``else { Sum = Sum + x; }``: Blok else yang dijalankan jika pengguna memasukkan nilai ``x`` selain 999, dan nilai ``x`` ditambahkan ke ``Sum``.
12. ``System.out.println("Hasil penjumlahan = "+ Sum);``: Menampilkan hasil penjumlahan nilai-nilai yang dimasukkan pengguna ke konsol.

Output:

Program ini akan meminta pengguna memasukkan nilai ``x`` (bilangan bulat) sampai pengguna memasukkan angka 999. Program kemudian akan menghitung dan mencetak hasil penjumlahan nilai-nilai

tersebut ke konsol. Jika pengguna langsung memasukkan 999, program akan memberikan pesan bahwa kasusnya kosong.

• Program 16

-OutPut Program class PrintXRepeat Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java PrintXRepeat.java
Masukkan nilai x (int), akhiri dg 999 : 45
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 45
```

Tujuan Program:

Program ini bertujuan untuk menghitung jumlah nilai yang dimasukkan pengguna (bilangan bulat) sampai pengguna memasukkan angka 999. Program menggunakan pernyataan perulangan `do-while` untuk mencapai tujuan ini.

Keyword dan Tipe Data:

1. `import java.util.Scanner;`: Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class PrintXRepeat {`: Mendefinisikan kelas `PrintXRepeat`, yang berisi program utama.
3. `public static void main(String[] args) {`: Mendefinisikan metode utama `main`, titik awal eksekusi program.
4. `int Sum;`: Mendefinisikan variabel `Sum` dengan tipe data `int` untuk menyimpan jumlah nilai yang dimasukkan pengguna.
5. `int x;`: Mendefinisikan variabel `x` dengan tipe data `int` untuk menyimpan nilai yang dimasukkan pengguna setiap iterasi.
6. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Looping dan Penjumlahan Nilai:

```
System.out.print ("Masukkan nilai x (int), akhiri
dg 999 : ");
x = masukan.nextInt();
```



```

if (x == 999){
    System.out.print("Kasus kosong \n");
}else {

    Sum = 0;
    do{
        Sum = Sum + x;
        System.out.print ("Masukkan nilai x (int),
akhiri dg 999 : ");
        x = masukan.nextInt();
    } while (x != 999);
    System.out.println ("Hasil penjumlahan =
"+Sum);

}

```

1. `System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai `x` (bilangan bulat) dan mengakhiri jika pengguna memasukkan 999.
2. `x = masukan.nextInt();`: Membaca nilai `x` dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `x`.
3. `if (x == 999){ ... }`: Pernyataan kondisional yang mengecek apakah pengguna langsung memasukkan angka 999 tanpa memasukkan nilai `x`.
4. `System.out.print("Kasus kosong \n");`: Menampilkan pesan bahwa kasusnya kosong jika pengguna langsung memasukkan 999.
5. `else { ... }`: Blok else yang dijalankan jika pengguna memasukkan nilai `x` selain 999.
6. `Sum = 0;`: Menginisialisasi variabel `Sum` dengan nilai 0.
7. `do { ... } while (x != 999);`: Pernyataan perulangan `do-while` yang akan terus berjalan selama pengguna memasukkan nilai `x` bukan 999.
8. `Sum = Sum + x;`: Menambahkan nilai `x` ke `Sum`.
9. `System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai `x` dan mengakhiri jika pengguna memasukkan 999.

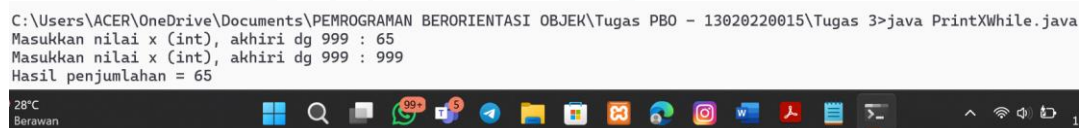
10. `x = masukan.nextInt();`: Membaca nilai `x` dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `x`.
11. `while (x != 999);`: Kondisi pengulangan yang menentukan apakah perulangan `do-while` akan dilanjutkan berdasarkan nilai `x`.

Output:

Program ini akan meminta pengguna memasukkan nilai `x` (bilangan bulat) sampai pengguna memasukkan angka 999. Program kemudian akan menghitung dan mencetak hasil penjumlahan nilai-nilai tersebut ke konsol. Jika pengguna langsung memasukkan 999, program akan memberikan pesan bahwa kasusnya kosong. Program menggunakan pernyataan perulangan `do-while` untuk memastikan setidaknya satu iterasi dilakukan sebelum mengevaluasi kondisi pengulangan.

• Program 17

-OutPut Program class PrintXWhile Dan Penjelasan Programnya



C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java PrintXWhile.java
Masukkan nilai x (int), akhiri dg 999 : 65
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 65

The screenshot shows a Windows taskbar at the bottom with various application icons. The main window displays the output of a Java program. The first input is 65, and the second is 999. The program calculates the sum of these two numbers, which is 65, and prints it out.

Tujuan Program:

Program ini bertujuan untuk menghitung jumlah nilai yang dimasukkan pengguna (bilangan bulat) sampai pengguna memasukkan angka 999. Program menggunakan pernyataan perulangan `while` untuk mencapai tujuan ini.

Keyword dan Tipe Data:

1. `import java.util.Scanner;`: Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class PrintXWhile {`: Mendeklarasikan kelas `PrintXWhile`, yang berisi program utama.

3. `public static void main(String[] args) {`: Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `int Sum;`: Mendeklarasikan variabel `Sum` dengan tipe data `int` untuk menyimpan jumlah nilai yang dimasukkan pengguna.
5. `int x;`: Mendeklarasikan variabel `x` dengan tipe data `int` untuk menyimpan nilai yang dimasukkan pengguna setiap iterasi.
6. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Looping dan Penjumlahan Nilai:

```
Sum = 0; /* Inisialisasi */
System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");
x = masukan.nextInt();/* First Elmt*/
while (x != 999) /* Kondisi berhenti */ {
    Sum = Sum + x; /* Proses */
    System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");
    x = masukan.nextInt();/* Next Elmt*/
}
System.out.println("Hasil penjumlahan = "+ Sum);
/* Terminasi */
```

1. `Sum = 0;`: Menginisialisasi variabel `Sum` dengan nilai 0.
2. `System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai `x` (bilangan bulat) dan mengakhiri jika pengguna memasukkan 999.
3. `x = masukan.nextInt();`: Membaca nilai `x` dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `x`.
4. `while (x != 999)`: Pernyataan perulangan `while` yang akan terus berjalan selama pengguna memasukkan nilai `x` bukan 999.
5. `Sum = Sum + x;`: Menambahkan nilai `x` ke `Sum`.
6. `System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");`: Menampilkan pesan untuk meminta pengguna memasukkan nilai `x` dan mengakhiri jika pengguna memasukkan 999.

7. ``x` = masukan.nextInt();``: Membaca nilai ``x`` dari pengguna menggunakan objek ``Scanner`` dan menyimpannya dalam variabel ``x``.
8. ``System.out.println("Hasil penjumlahan = "+ Sum);``: Menampilkan hasil penjumlahan nilai-nilai yang dimasukkan pengguna ke konsol.

Output:

Program ini akan meminta pengguna memasukkan nilai ``x`` (bilangan bulat) sampai pengguna memasukkan angka 999. Program kemudian akan menghitung dan mencetak hasil penjumlahan nilai-nilai tersebut ke konsol. Program menggunakan pernyataan perulangan ``while`` untuk memastikan bahwa kondisi pengulangan dievaluasi sebelum setiap iterasi dilakukan.

• Program 18

-OutPut Program class SubProgram Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java SubProgram.java
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
23
34
Ke dua bilangan : a = 23 b = 34
Maksimum = 34
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 34 b = 23
```

Tujuan Program:

Program ini memiliki tujuan untuk mencari nilai maksimum dari dua bilangan yang dimasukkan oleh pengguna dan menukar nilai kedua bilangan tersebut. Program ini menggunakan subprogram (metode) untuk mencapai tujuan ini.

Keyword dan Tipe Data:

1. ``import java.util.Scanner;``: Mengimpor kelas ``Scanner`` dari paket ``java.util``, yang digunakan untuk membaca input dari pengguna.
2. ``public class SubProgram {``: Mendeklarasikan kelas ``SubProgram``, yang berisi program utama dan subprogram (metode).

3. `public static int maxab (int a, int b) { ... }`: Mendeklarasikan subprogram `maxab` yang mengembalikan nilai maksimum dari dua bilangan `a` dan `b`.
4. `public static void tukar (int a, int b) { ... }`: Mendeklarasikan subprogram `tukar` yang menukar nilai dua bilangan `a` dan `b`.
5. `public static void main(String[] args) { }`: Mendeklarasikan metode utama `main`, titik awal eksekusi program.
6. `int a, b;`: Mendeklarasikan variabel `a` dan `b` dengan tipe data `int` untuk menyimpan dua bilangan yang dimasukkan pengguna.
7. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Subprogram maxab:

```
public static int maxab (int a, int b){  
    return ((a >= b) ? a : b);  
}
```

- `int maxab (int a, int b)`: Mendeklarasikan subprogram `maxab` dengan tipe data kembalian `int` yang menerima dua parameter `a` dan `b`.
- `return ((a >= b) ? a : b);`: Mengembalikan nilai maksimum antara `a` dan `b` menggunakan operator kondisional.

Subprogram tukar:

```
public static void tukar (int a, int b)  
{  
    int temp; temp = a; a = b;  
    b = temp;  
    System.out.println ("Ke dua bilangan setelah  
tukar: a = "+ a +" b = "+ b);  
}
```

- `void tukar (int a, int b)`: Mendeklarasikan subprogram `tukar` tanpa nilai kembalian yang menerima dua parameter `a` dan `b`.
- `int temp; temp = a; a = b; b = temp;`: Menukar nilai `a` dan `b` menggunakan variabel sementara `temp`.

- `System.out.println ("Ke dua bilangan setelah tukar: a = "+ a +" b = "+ b);`: Menampilkan nilai kedua bilangan setelah pertukaran.

Program Utama:

```
a = masukan.nextInt(); b = masukan.nextInt();
System.out.println ("Ke dua bilangan : a = "+ a +"
b = "+ b);
System.out.println ("Maksimum = " + (maxab (a,
b)));
System.out.print("Tukar kedua bilangan... \n");
tukar (a, b);
```

1. `a = masukan.nextInt(); b = masukan.nextInt();`: Membaca dua bilangan dari pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `a` dan `b`.
2. `System.out.println ("Ke dua bilangan : a = "+ a +" b = "+ b);`: Menampilkan nilai kedua bilangan yang dimasukkan pengguna.
3. `System.out.println ("Maksimum = " + (maxab (a, b)));`: Menampilkan nilai maksimum dari dua bilangan menggunakan subprogram `maxab`.
4. `tukar (a, b);`: Memanggil subprogram `tukar` untuk menukar nilai kedua bilangan dan menampilkan hasilnya.

Output:

Program ini akan meminta pengguna memasukkan dua bilangan, kemudian menampilkan nilai maksimum dari kedua bilangan tersebut dan menukar nilai kedua bilangan. Hasil pertukaran nilai juga akan ditampilkan ke konsol.

• Program 19

-OutPut Program class Tempair Dan Penjelasan Programnya

```
C:\Users\ACER\OneDrive\Documents\PEMROGRAMAN BERORIENTASI OBJEK\Tugas PBO - 13020220015\Tugas 3>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = 45
Wujud air cair
45
```

Tujuan Program:

Program ini bertujuan untuk menentukan wujud air berdasarkan suhu (temperatur) yang dimasukkan oleh pengguna. Program menggunakan pernyataan kondisional `if-else if` untuk mengevaluasi tiga kasus: air beku, air cair, dan air uap/gas.

Keyword dan Tipe Data:

1. `import java.util.Scanner;`: Mengimpor kelas `Scanner` dari paket `java.util`, yang digunakan untuk membaca input dari pengguna.
2. `public class Tempair {`: Mendeklarasikan kelas `Tempair`, yang berisi program utama.
3. `public static void main(String[] args) {`: Mendeklarasikan metode utama `main`, titik awal eksekusi program.
4. `int T;`: Mendeklarasikan variabel `T` dengan tipe data `int` untuk menyimpan suhu (temperatur) yang dimasukkan pengguna.
5. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna.

Program Penentuan Wujud Air:

```
System.out.print ("Contoh IF tiga kasus \n");
System.out.print ("Temperatur (der. C) = ");
T = masukan.nextInt();
if (T < 0) {
    System.out.print ("Wujud air beku \n" + T);
} else if ((0 <= T) && (T <= 100)) {
    System.out.print ("Wujud air cair \n" + T);
} else if (T > 100) {
    System.out.print ("Wujud air uap/gas \n" + T);
};
```

1. `System.out.print ("Contoh IF tiga kasus \n");`: Menampilkan pesan ke konsol untuk memberikan informasi tentang program.

-

2. ``System.out.print ("Temperatur (der. C) = ");``: Menampilkan pesan untuk meminta pengguna memasukkan suhu dalam derajat Celsius.
3. ``T = masukan.nextInt();``: Membaca nilai suhu dari pengguna menggunakan objek ``Scanner`` dan menyimpannya dalam variabel ``T``.
4. ``if (T < 0) { ... }``: Pernyataan kondisional yang mengecek apakah suhu kurang dari 0.
 - ``System.out.print ("Wujud air beku \n" + T);``: Menampilkan pesan bahwa air dalam keadaan beku jika suhu kurang dari 0.
5. ``else if ((0 <= T) && (T <= 100)) { ... }``: Pernyataan kondisional kedua yang mengecek apakah suhu berada dalam rentang 0 hingga 100.
 - ``System.out.print ("Wujud air cair \n" + T);``: Menampilkan pesan bahwa air dalam keadaan cair jika suhu berada dalam rentang 0 hingga 100.
6. ``else if (T > 100) { ... }``: Pernyataan kondisional ketiga yang mengecek apakah suhu lebih dari 100.
 - ``System.out.print ("Wujud air uap/gas \n" + T);``: Menampilkan pesan bahwa air dalam keadaan uap atau gas jika suhu lebih dari 100.
7. ``};``: Tanda kurung kurawal penutup untuk menyelesaikan blok ``if-else if``.

Output:

Program ini akan meminta pengguna memasukkan suhu dalam derajat Celsius. Berdasarkan suhu yang dimasukkan, program akan menentukan dan mencetak wujud air (beku, cair, atau uap/gas) ke konsol. Jika suhu tidak masuk ke dalam salah satu kasus, program tidak memberikan output tambahan.