

Penjelasan:

Program ini adalah sebuah program Java sederhana yang menggunakan sebuah loop **while(true)** untuk terus mencetak pesan "Print satu baris....." dalam satu baris. Program ini akan terus melakukan loop tanpa henti sampai program dihentikan secara paksa dengan menekan **Ctrl + C** pada keyboard. Pesan "Program akan looping, akhiri dengan ^c" adalah pesan yang memberi tahu pengguna cara menghentikan program.

Output Program 3:

```
Contoh IF satu kasus
Ketikkan suatu nilai integer : 149

Nilai a positif 149
PS C:\Users\ASUS> 
```

Penjelasan:

Program ini meminta pengguna untuk memasukkan sebuah nilai integer. Jika nilai yang dimasukkan oleh pengguna lebih besar dari atau sama dengan 0, maka program akan mencetak pesan "Nilai a positif" diikuti dengan nilai yang dimasukkan oleh pengguna.

Output Program 4:

```
Contoh IF dua kasus
Ketikkan suatu nilai integer :-149
Nilai a negatif -149
```

Penjelasan:

Program ini meminta pengguna untuk memasukkan nilai integer. Kemudian program memeriksa apakah nilai yang dimasukkan lebih besar dari atau sama dengan 0. Jika ya, program mencetak pesan bahwa nilai tersebut positif. Jika tidak, program mencetak pesan bahwa nilai tersebut negatif.

Output Program 5:

```
Contoh IF tiga kasus
Ketikkan suatu nilai integer :0
Nilai Nol 0
PS C:\Users\ASUS> 
```

Penjelasan:

Program ini meminta pengguna untuk memasukkan nilai integer. Kemudian program memeriksa nilai yang dimasukkan dan mencetak pesan yang sesuai: "Nilai a positif" jika a lebih besar dari 0. "Nilai Nol" jika a sama dengan 0. "Nilai a negatif" jika a kurang dari 0.

Output Program 6:

```
true
benar
PS C:\Users\ASUS>
```

Penjelasan:

Program menginisialisasi bool sebagai true. Pernyataan if pertama mencetak "true\n" dengan benar karena bool bernilai true. Pernyataan if kedua tidak mencetak apa pun karena !bool bernilai false.

Output Program 7:

```
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a
```

Penjelasan:

Program ini meminta pengguna untuk memasukkan huruf. Kemudian program memeriksa huruf yang dimasukkan dan mencetak pesan yang sesuai: Pesan "Yang anda ketik adalah [huruf]" jika huruf yang dimasukkan adalah a, u, e, i, atau o. Pesan "Yang anda ketik adalah huruf mati" jika huruf yang dimasukkan bukan a, u, e, i, atau o.

Output Program 8:

```
Jari-jari lingkaran =12
Luas lingkaran = 452.37598
Akhir program
```

Penjelasan:

Program ini meminta pengguna untuk memasukkan nilai jari-jari lingkaran, kemudian menghitung dan menampilkan luas lingkaran dengan rumus $\text{PHI} * r * r$.

Penggunaan konstanta **PHI** di sini memungkinkan nilai phi (π) untuk tetap konsisten dan dapat digunakan ulang tanpa perlu menuliskan nilai tersebut secara langsung dalam rumus perhitungan.

Output Program 9:

```
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
1 4
Ke dua bilangan : a = 1 b = 4
Nilai b yang maksimum: 4
```

Penjelasan:

Program ini menggunakan **Scanner** untuk meminta pengguna memasukkan dua bilangan, kemudian membandingkan kedua bilangan tersebut menggunakan **if-else** statement untuk menentukan bilangan mana yang lebih besar. Jika **a** lebih besar dari atau sama dengan **b**, program akan mencetak nilai **a** sebagai nilai maksimum. Jika tidak, program akan mencetak nilai **b** sebagai nilai maksimum.

Ouput Program 10:

```
Baca N, print 1 s/d N N = 5
1
2
3
4
5
Akhir program
```

Penjelasan:

Program mendeklarasikan variabel untuk menyimpan input pengguna dan penghitung. Kemudian program meminta pengguna untuk memasukkan nilai N. Program membaca input pengguna dan menyimpannya nilai di N. Program menggunakan perulangan for untuk berulang dari 1 hingga N. Di dalam perulangan, program mencetak nilai penghitung saat ini (i). Kemudian program mencetak pesan yang menunjukkan akhir program.

Output Program 11:

```
Nilai N >0 = 9
Print i dengan ITERATE :
1
2
3
4
5
6
7
8
9
```

Penjelasan:

Program meminta pengguna untuk memasukkan nilai N. Program menginisialisasi i dengan 1. Program memasuki perulangan for. Di dalam perulangan, program mencetak nilai i ke konsol. Program terus

berulang dan menambah nilai i hingga i sama dengan N. Ketika i sama dengan N, perulangan dihentikan menggunakan break.

Output Program 12:

```
Nilai N >0 = 6
Print i dengan REPEAT:
1
2
3
4
5
6
PS C:\Users\ASUS> █
```

Penjelasan:

Program menginisialisasi i dengan 1. Program memasuki perulangan do-while. Di dalam perulangan, program mencetak nilai i ke konsol. Perulangan mengecek kondisi ($i \leq N$). Jika kondisinya benar, perulangan dijalankan lagi. Setelah setiap iterasi, i ditambah 1. Program terus berulang dan menambah nilai i hingga i melebihi N.

Output Program 13:

```
Nilai N >0 = 3
Print i dengan WHILE:
1
2
3
PS C:\Users\ASUS> █
```

Penjelasan:

Program menginisialisasi i dengan 1. Program memasuki perulangan while. Selama nilai i kurang dari atau sama dengan N: Program mencetak nilai i ke konsol. Setelah setiap iterasi, i ditambah 1. Perulangan while terus berlanjut hingga kondisi $i \leq N$ menjadi salah (ketika i melebihi N)

Output Program 14:

```
Nilai N >0 = 4
Print i dengan WHILE (ringkas):
1
2
3
4
PS C:\Users\ASUS>
```

Penjelasan:

Program menginisialisasi *i* dengan 1. Program memasuki perulangan while. Selama nilai *i* kurang dari atau sama dengan *N*: Program mencetak nilai *i* ke konsol (menggunakan *i* sebelum penambahan). *i* ditambah 1 setelah pencetakan (menggunakan *i*++). Perulangan while terus berlanjut hingga kondisi *i* <= *N* menjadi salah (ketika *i* melebihi *N*).

Output Program 15:

```
Masukkan nilai x (int), akhiri dg 999: 5
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 15
PS C:\Users\ASUS>
```

Penjelasan:

Program ini meminta pengguna memasukkan nilai integer (**x**) secara berulang dan menjumlahkannya. Jika pengguna memasukkan nilai **999**, program akan menganggap itu sebagai akhir dari input dan mencetak hasil penjumlahan dari semua nilai yang dimasukkan sebelumnya. Program ini menggunakan loop **for(;;)** yang merupakan bentuk loop tak terbatas. Loop ini akan terus berjalan sampai terjadi break. Pada setiap iterasi loop, program akan meminta pengguna memasukkan nilai **x**. Jika nilai **x** sama dengan **999**, loop akan dihentikan dengan break. Jika tidak, nilai **x** akan ditambahkan ke variabel **Sum** yang merupakan hasil penjumlahan dari semua nilai yang dimasukkan pengguna sebelumnya.

Output Program 16:

```
Masukkan nilai x (int), akhiri dg 999 : 11
Masukkan nilai x (int), akhiri dg 999 : 4
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 15
```

Penjelasan:

Jika x adalah 999, program selesai. Jika x bukan 999, program menginisialisasi Sum dengan 0. Program memasuki perulangan do-while. Di dalam perulangan: Nilai x ditambahkan ke Sum. Program meminta pengguna untuk memasukkan nilai x berikutnya. Perulangan dihentikan ketika nilai x yang dimasukkan adalah 999. Program mencetak hasil penjumlahan (Sum) ke konsol

Ouput Program 17:

```
Masukkan nilai x (int), akhiri dg 999 : 24
Masukkan nilai x (int), akhiri dg 999 : 6
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 30
```

Penjelasan:

Program menginisialisasi Sum dengan 0. Program meminta pengguna untuk memasukkan nilai x. Perulangan while dimulai. Di dalam perulangan: Jika x tidak sama dengan 999: Nilai x ditambahkan ke Sum. Program meminta pengguna untuk memasukkan nilai x berikutnya. Jika x sama dengan 999, perulangan berhenti. Program mencetak hasil penjumlahan (Sum) ke konsol

Output Program 18:

```
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
4
9
Ke dua bilangan : a = 4 b = 9
Maksimum = 9
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 9 b = 4
```

Penjelasan:

program meminta pengguna untuk memasukkan dua bilangan bulat, kemudian mencetak kedua bilangan tersebut dan nilai maksimumnya menggunakan fungsi **maxab**. Selanjutnya, program memanggil prosedur **tukar** untuk menukar nilai kedua bilangan, dan mencetak kedua bilangan setelah ditukar.

Output Program 19:

```
Contoh IF tiga kasus
Temperatur (der. C) = 9
Wujud air cair
9
```

Penjelasan:

Program ini memiliki penggunaan if tiga kasus untuk menentukan wujud air berdasarkan temperatur yang dimasukkan pengguna. Program membaca input temperatur, lalu menentukan dan menampilkan wujud air berdasarkan kondisi yang terpenuhi