

TUGAS 3

Nama : Nur Bakti Widjaya

NIM : 13020220134

Kelas : A4

Matkul : Pemrograman Berorientasi Objek

Kode Program 1

```
D:\Tugas 3\Tugas3_SourceCode>java BacaString  
  
Baca string dan Integer:  
masukkan sebuah string: Assalamualaikum  
String yang dibaca : Assalamualaikum
```

Penjelasan :

1. **Import Paket:** Program mengimpor kelas-kelas yang diperlukan untuk operasi masukan/keluaran (input/output).
2. **Deklarasi Kelas:** Kelas utama program diberi nama **BacaString**.
3. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
4. **Inisialisasi BufferedReader:** Inisialisasi objek **BufferedReader** untuk membaca masukan dari konsol.
5. **Prompt Pengguna:** Menampilkan pesan kepada pengguna untuk memasukkan sebuah string.
6. **Membaca Masukan:** Membaca masukan string dari pengguna menggunakan metode **readLine()**.
7. **Menampilkan Kembali String:** Menampilkan kembali string yang dimasukkan oleh pengguna ke konsol.
8. **Penanganan Kesalahan:** Penanganan kemungkinan kesalahan yang mungkin terjadi saat pembacaan masukan.

Kode Program 2

```
Print satu baris ....  
Print satu baris ....  
Print satu baris ....  
Print satu baris ....  
Print satu baris ....
```

Penjelasan :

1. **Deklarasi Kelas:** Kelas utama program diberi nama **ForEver**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Loop Tanpa Akhir:** Program menggunakan konstruksi loop **while (true)** yang membuatnya terus berjalan tanpa henti. Pernyataan **true** membuat kondisi loop selalu benar, sehingga loop tidak akan pernah berhenti kecuali program dipaksa untuk dihentikan.
4. **Pesanan Konsol:** Sebuah pesan cetak ditampilkan di awal program, memberi tahu pengguna bahwa program akan terus berjalan tanpa henti, dan memberikan instruksi untuk menghentikan program menggunakan kombinasi tombol **^c** (biasanya digunakan untuk menghentikan program di lingkungan terminal Unix/Linux).
5. **Loop Internal:** Di dalam loop, pesan "Print satu baris" akan terus dicetak ke konsol tanpa henti.
6. **Komponen 'TODO':** Komentar **TODO** menandakan bahwa ada pekerjaan yang perlu dilakukan atau tambahan yang perlu ditambahkan di bagian ini. Biasanya, ini digunakan untuk menunjukkan bagian-bagian kode yang belum sepenuhnya diimplementasikan atau direncanakan.

Kode Program 3

```
D:\Tugas 3\Tugas3_SourceCode>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 6

Nilai a positif 6
```

Penjelasan :

1. **Import Paket:** Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca masukan dari pengguna.
2. **Deklarasi Kelas:** Kelas utama program diberi nama **If1**.
3. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
4. **Deklarasi Variabel:** Sebuah objek **Scanner** dengan nama **masukan** dideklarasikan untuk membaca masukan dari pengguna. Selain itu, sebuah variabel **a** dengan tipe data **int** juga dideklarasikan untuk menyimpan nilai integer yang dimasukkan oleh pengguna.
5. **Prompt Pengguna:** Program mencetak pesan ke konsol untuk meminta pengguna untuk memasukkan sebuah nilai integer.
6. **Membaca Masukan:** Program menggunakan metode **nextInt()** dari objek **Scanner** untuk membaca masukan integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel **a**.

7. **Struktur Kondisional IF:** Program menggunakan struktur kondisional **if** untuk mengecek apakah nilai yang dimasukkan oleh pengguna adalah positif atau tidak. Jika nilai **a** lebih besar atau sama dengan 0, maka program akan mencetak pesan yang menyatakan bahwa nilai **a** positif.
8. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.

Kode Program 4

```
D:\Tugas 3\Tugas3_SourceCode>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :4
Nilai a positif 4
```

Penjelasan :

1. **Import Paket:** Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca masukan dari pengguna.
2. **Deklarasi Kelas:** Kelas utama program diberi nama **If2**.
3. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
4. **Deklarasi Variabel:** Sebuah objek **Scanner** dengan nama **masukan** dideklarasikan untuk membaca masukan dari pengguna. Selain itu, sebuah variabel **a** dengan tipe data **int** juga dideklarasikan untuk menyimpan nilai integer yang dimasukkan oleh pengguna.
5. **Prompt Pengguna:** Program mencetak pesan ke konsol untuk meminta pengguna untuk memasukkan sebuah nilai integer.
6. **Membaca Masukan:** Program menggunakan metode **nextInt()** dari objek **Scanner** untuk membaca masukan integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel **a**.
7. **Struktur Kondisional IF:** Program menggunakan struktur kondisional **if** untuk mengecek apakah nilai yang dimasukkan oleh pengguna positif atau negatif. Jika nilai **a** lebih besar atau sama dengan 0, maka program akan mencetak pesan yang menyatakan bahwa nilai **a** positif. Jika tidak, program akan mencetak pesan yang menyatakan bahwa nilai **a** negatif.
8. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.

Kode Program 5

```
D:\Tugas 3\Tugas3_SourceCode>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :8
Nilai a positif 8
```

Penjelasan :

1. **Import Paket:** Program tidak mengimpor paket tambahan.
2. **Deklarasi Kelas:** Kelas utama program diberi nama **If3**.
3. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
4. **Deklarasi Variabel:** Sebuah variabel **a** dengan tipe data **int** dideklarasikan untuk menyimpan nilai integer yang dimasukkan oleh pengguna. Sebuah objek **Scanner** dengan nama **masukan** juga dideklarasikan untuk membaca masukan dari pengguna.
5. **Prompt Pengguna:** Program mencetak pesan ke konsol untuk meminta pengguna untuk memasukkan sebuah nilai integer.
6. **Membaca Masukan:** Program menggunakan metode **nextInt()** dari objek **Scanner** untuk membaca masukan integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel **a**.
7. **Struktur Kondisional IF:** Program menggunakan struktur kondisional **if** dengan tiga kasus. Pertama, program memeriksa apakah nilai **a** lebih besar dari 0. Jika iya, maka program akan mencetak pesan yang menyatakan bahwa nilai **a** positif. Jika tidak, program akan memeriksa apakah nilai **a** sama dengan 0. Jika iya, maka program akan mencetak pesan yang menyatakan bahwa nilai **a** adalah nol. Jika kedua kondisi tersebut tidak terpenuhi, maka program akan mencetak pesan yang menyatakan bahwa nilai **a** negatif.
8. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.

Kode Program 6

```
D:\Tugas 3\Tugas3_SourceCode>java KasusBoolean
true
benar
```

Penjelasan :

1. **Deklarasi Kelas:** Kelas utama program diberi nama **KasusBoolean**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Sebuah variabel **bool** dengan tipe data **boolean** dideklarasikan untuk menyimpan nilai boolean.
4. **Inisialisasi Variabel:** Variabel **bool** diinisialisasi dengan nilai **true**.
5. **Struktur Kondisional IF:** Program menggunakan struktur kondisional **if** untuk mengevaluasi nilai boolean yang disimpan dalam variabel **bool**. Jika nilai **bool** adalah **true**, maka program akan mencetak "true". Jika tidak, program akan mencetak "false". Kemudian, program menggunakan struktur kondisional **if** yang lain dengan operator negasi **!** untuk mengevaluasi nilai kebalikan dari

bool. Jika nilai **bool** adalah **false**, maka program akan mencetak "salah". Jika tidak, program akan mencetak "benar".

6. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.

Kode Program 7

```
D:\Tugas 3\Tugas3_SourceCode>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
g
Yang anda ketik adalah huruf mati
```

Penjelasan :

1. **Deklarasi Kelas:** Kelas utama program diberi nama **KasusSwitch**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Sebuah variabel **cc** dengan tipe data **char** dideklarasikan untuk menyimpan huruf yang dimasukkan oleh pengguna. Sebuah objek **Scanner** dengan nama **masukan** juga dideklarasikan untuk membaca masukan dari pengguna.
4. **Prompt Pengguna:** Program mencetak pesan ke konsol untuk meminta pengguna untuk memasukkan sebuah huruf, dan menunjukkan bahwa input harus diakhiri dengan tombol RETURN.
5. **Membaca Masukan:** Program menggunakan metode **next().charAt(0)** dari objek **Scanner** untuk membaca huruf yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel **cc**.
6. **Struktur Kontrol switch-case:** Program menggunakan struktur kontrol **switch-case** untuk mengevaluasi nilai yang disimpan dalam variabel **cc**. Program akan menjalankan blok kode yang sesuai dengan nilai **cc** yang diberikan oleh pengguna. Jika **cc** adalah 'a', 'u', 'e', 'i', atau 'o', program akan mencetak pesan yang menyatakan huruf yang dimasukkan oleh pengguna. Jika **cc** tidak sesuai dengan nilai-nilai tersebut, program akan mencetak pesan yang menyatakan bahwa huruf yang dimasukkan adalah "huruf mati" (tidak termasuk dalam vokal).
7. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.

Kode Program 8

```
D:\Tugas 3\Tugas3_SourceCode>java Konstant
Jari-jari lingkaran =55
Luas lingkaran = 9503.037
Akhir program
```

Penjelasan :

1. **Deklarasi Kelas:** Kelas utama program diberi nama **Konstant**.

2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel Konstanta:** Sebuah konstanta **PHI** dideklarasikan dengan kata kunci **final** untuk menyimpan nilai Phi (π). Konstanta ini memiliki tipe data **float** dan nilainya diinisialisasi dengan 3.1415.
4. **Deklarasi Variabel:** Sebuah variabel **r** dengan tipe data **float** dideklarasikan untuk menyimpan jari-jari lingkaran yang dimasukkan oleh pengguna. Sebuah objek **Scanner** dengan nama **masukan** juga dideklarasikan untuk membaca masukan dari pengguna.
5. **Prompt Pengguna:** Program mencetak pesan ke konsol untuk meminta pengguna untuk memasukkan jari-jari lingkaran.
6. **Membaca Masukan:** Program menggunakan metode **nextFloat()** dari objek **Scanner** untuk membaca masukan float yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel **r**.
7. **Menghitung dan Menampilkan Luas Lingkaran:** Program menghitung luas lingkaran menggunakan rumus πr^2 (Phi dikalikan jari-jari kuadrat) dan kemudian mencetak hasilnya ke konsol.
8. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.

Kode Program 9

```
D:\Tugas 3\Tugas3_SourceCode>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
1
3
Ke dua bilangan : a = 1 b = 3
Nilai b yang maksimum: 3
```

Penjelasan :

1. **Deklarasi Kelas:** Kelas utama program diberi nama **Max2**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Dua variabel **a** dan **b** dengan tipe data **int** dideklarasikan untuk menyimpan dua bilangan yang dimasukkan oleh pengguna. Sebuah objek **Scanner** dengan nama **masukan** juga dideklarasikan untuk membaca masukan dari pengguna.
4. **Prompt Pengguna:** Program mencetak pesan ke konsol untuk meminta pengguna untuk memasukkan dua bilangan, dan menunjukkan bahwa bilangan-bilangan tersebut harus dipisahkan oleh tombol RETURN.
5. **Membaca Masukan:** Program menggunakan metode **nextInt()** dari objek **Scanner** untuk membaca dua bilangan integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel **a** dan **b**.

6. **Menampilkan Bilangan:** Setelah membaca masukan, program mencetak kembali bilangan yang dimasukkan oleh pengguna ke konsol.
7. **Struktur Kondisional IF-ELSE:** Program menggunakan struktur kondisional **if-else** untuk mengecek mana di antara dua bilangan yang lebih besar. Jika nilai **a** lebih besar atau sama dengan **b**, maka program akan mencetak bahwa nilai **a** adalah nilai maksimum. Jika tidak, maka program akan mencetak bahwa nilai **b** adalah nilai maksimum.
8. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.

Kode Program 10

```
D:\Tugas 3\Tugas3_SourceCode>java PriFor
Baca N, print 1 s/d N N = 6
1
2
3
4
5
6
Akhir program
```

Penjelasan :

1. **Deklarasi Kelas:** Kelas utama program diberi nama **PriFor**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Dua variabel **i** dan **N** dengan tipe data **int** dideklarasikan untuk menyimpan nilai indeks iterasi loop dan bilangan **N** yang dimasukkan oleh pengguna. Sebuah objek **Scanner** dengan nama **masukan** juga dideklarasikan untuk membaca masukan dari pengguna.
4. **Prompt Pengguna:** Program mencetak pesan ke konsol untuk meminta pengguna untuk memasukkan nilai **N**.
5. **Membaca Masukan:** Program menggunakan metode **nextInt()** dari objek **Scanner** untuk membaca bilangan integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel **N**.
6. **Loop For:** Program menggunakan loop **for** untuk mengulangi pernyataan di dalam blok loop sebanyak **N** kali. Dimulai dari nilai **i** sama dengan 1, setiap iterasi akan mencetak nilai **i** ke konsol, kemudian nilai **i** akan bertambah satu. Loop akan berakhir saat nilai **i** lebih besar dari **N**.
7. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.
8. **Pesan Akhir:** Setelah loop selesai dieksekusi, program mencetak pesan akhir ke konsol.

Kode Program 11

```
D:\Tugas 3\Tugas3_SourceCode>java PrintIterasi
Nilai N >0 = 7
Print i dengan ITERATE :
1
2
3
4
5
6
7
```

Penjelasan :

1. **Deklarasi Kelas:** Program memiliki kelas utama bernama **PrintIterasi**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Variabel **N** dan **i** dideklarasikan untuk menyimpan nilai **N** dan nilai iterasi, serta objek **Scanner** untuk masukan.
4. **Prompt Pengguna:** Pesan meminta pengguna untuk memasukkan nilai **N** ditampilkan.
5. **Membaca Masukan:** Nilai **N** dimasukkan oleh pengguna dan disimpan dalam variabel **N**.
6. **Inisialisasi Iterasi:** Variabel **i** diinisialisasi dengan nilai 1 sebagai awal iterasi.
7. **Loop for Tanpa Batas:** Program menggunakan loop **for** tanpa kondisi berhenti, dengan peningkatan nilai **i** setiap iterasi.
8. **Periksa Kondisi:** Nilai **i** dicetak setiap iterasi. Jika **i** sama dengan **N**, loop berhenti menggunakan pernyataan **break**.
9. **Penanganan Kesalahan:** Program tidak menyertakan penanganan kesalahan.

Kode Program 12

```
D:\Tugas 3\Tugas3_SourceCode>java PrintRepeat
Nilai N >0 = 4
Print i dengan REPEAT:
1
2
3
4
```

Penjelasan :

1. **Deklarasi Kelas:** Program memiliki kelas utama bernama **PrintRepeat**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Variabel **N** dan **i** dideklarasikan untuk menyimpan nilai **N** dan iterasi, serta objek **Scanner** untuk masukan.

4. **Prompt Pengguna:** Pesan meminta pengguna untuk memasukkan nilai N ditampilkan.
5. **Membaca Masukan:** Nilai N dimasukkan oleh pengguna dan disimpan dalam variabel N.
6. **Inisialisasi Iterasi:** Variabel **i** diinisialisasi dengan nilai 1.
7. **Loop do-while:** Program mencetak nilai **i** dan meningkatkan nilainya. Loop ini berjalan setidaknya sekali dan berlanjut selama nilai **i** kurang dari atau sama dengan N.
8. **Periksa Kondisi:** Kondisi **while (i <= N)** digunakan sebagai kondisi penghentian iterasi.
9. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.

Kode Program 13

```
D:\Tugas 3\Tugas3_SourceCode>java PrintWhile
Nilai N >0 = 12
Print i dengan WHILE:
1
2
3
4
5
6
7
8
9
10
11
12
```

Penjelasan :

1. **Deklarasi Kelas:** Kelas utama program diberi nama **PrintWhile**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Dua variabel **N** dan **i** dengan tipe data **int** dideklarasikan untuk menyimpan nilai N yang dimasukkan oleh pengguna dan nilai iterasi saat ini. Sebuah objek **Scanner** dengan nama **masukan** juga dideklarasikan untuk membaca masukan dari pengguna.
4. **Prompt Pengguna:** Program mencetak pesan ke konsol untuk meminta pengguna untuk memasukkan nilai N.
5. **Membaca Masukan:** Program menggunakan metode **nextInt()** dari objek **Scanner** untuk membaca bilangan integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel N.
6. **Inisialisasi Iterasi:** Variabel **i** diinisialisasi dengan nilai 1 sebagai langkah awal iterasi.
7. **Loop while:** Program menggunakan loop **while** untuk mencetak nilai **i** dan kemudian meningkatkan nilai **i** dengan 1 setiap iterasi. Loop akan terus berjalan selama nilai **i** kurang dari atau sama dengan nilai N yang dimasukkan oleh pengguna.
8. **Periksa Kondisi:** Kondisi **while (i <= N)** digunakan sebagai kondisi pengulangan.
9. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.

Kode Program 14

```
D:\Tugas 3\Tugas3_SourceCode>java PrintWhile1
Nilai N >0 = 3
Print i dengan WHILE (ringkas):
1
2
3
```

Penjelasan :

1. **Deklarasi Kelas:** Kelas utama program diberi nama **PrintWhile1**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Variabel **N** dideklarasikan untuk menyimpan nilai **N** yang dimasukkan oleh pengguna. Variabel **i** diinisialisasi dengan nilai 1.
4. **Membaca Masukan:** Program menggunakan objek **Scanner** untuk membaca bilangan integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel **N**.
5. **Loop while (Ringkas):** Program menggunakan loop **while** untuk mencetak nilai **i** dan meningkatkan nilai **i** dengan 1 setiap iterasi. Loop ini berjalan selama nilai **i** kurang dari atau sama dengan nilai **N** yang dimasukkan oleh pengguna. Peningkatan nilai **i** dilakukan dengan menggunakan operator **++** dalam pemanggilan fungsi **System.out.println()**.
6. **Periksa Kondisi:** Kondisi **while (i <= N)** digunakan sebagai kondisi pengulangan.
7. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang disertakan dalam program ini.

Kode Program 15

```
D:\Tugas 3\Tugas3_SourceCode>java PrintXinterasi
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 4
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 11
```

Penjelasan :

1. **Deklarasi Kelas:** Program memiliki kelas utama bernama **PrintXinterasi**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Variabel **Sum** dan **x** dideklarasikan untuk menyimpan total penjumlahan dan nilai yang dibaca dari pengguna. Objek **Scanner** juga dideklarasikan.
4. **Prompt Pengguna:** Pesan meminta pengguna untuk memasukkan nilai **x** ditampilkan.
5. **Membaca Masukan Pertama:** Nilai pertama dari pengguna dibaca dan disimpan dalam variabel **x**.

6. **Periksa Kondisi Pertama:** Program memeriksa apakah nilai **x** sama dengan **999**. Jika ya, program menganggap kasusnya kosong.
7. **Iterasi untuk Membaca Nilai:** Jika nilai **x** bukan **999**, maka nilai tersebut dimasukkan ke dalam total penjumlahan **Sum**. Program terus meminta pengguna untuk memasukkan nilai **x** hingga pengguna memasukkan **999**. Penggunaan **for** loop tanpa kondisi berhenti dan **break** memungkinkan keluar dari loop.
8. **Penanganan Kesalahan:** Tidak ada penanganan kesalahan yang diimplementasikan secara eksplisit.
9. **Cetak Total Penjumlahan:** Setelah pengguna memasukkan nilai **999**, total penjumlahan dari nilai-nilai yang dimasukkan oleh pengguna dicetak.

Kode Program 16

```
D:\Tugas 3\Tugas3_SourceCode>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 1
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 3
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 6
```

Penjelasan :

1. **Deklarasi Kelas:** Program memiliki kelas utama bernama **PrintXRepeat**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Variabel **Sum** dan **x** dideklarasikan untuk menyimpan total penjumlahan dan nilai yang dibaca dari pengguna. Objek **Scanner** juga dideklarasikan.
4. **Prompt Pengguna:** Pesan meminta pengguna untuk memasukkan nilai **x** ditampilkan.
5. **Membaca Masukan Pertama:** Nilai pertama dari pengguna dibaca dan disimpan dalam variabel **x**.
6. **Periksa Kondisi Pertama:** Program memeriksa apakah nilai **x** sama dengan **999**. Jika ya, program menganggap kasusnya kosong.
7. **Iterasi untuk Membaca Nilai:** Jika nilai **x** bukan **999**, maka nilai tersebut dimasukkan ke dalam total penjumlahan **Sum**. Program terus meminta pengguna untuk memasukkan nilai **x** hingga pengguna memasukkan **999**. Ini dilakukan menggunakan loop **do-while** yang berjalan setidaknya sekali.
8. **Periksa Kondisi Pengulangan:** Pengulangan dilanjutkan selama nilai **x** yang dimasukkan oleh pengguna tidak sama dengan **999**.
9. **Cetak Total Penjumlahan:** Setelah pengguna memasukkan nilai **999**, total penjumlahan dari nilai-nilai yang dimasukkan oleh pengguna dicetak.

Kode Program 17

```
D:\Tugas 3\Tugas3_SourceCode>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 4
Masukkan nilai x (int), akhiri dg 999 : 3
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 12
```

Penjelasan :

1. **Deklarasi Kelas:** Program memiliki kelas utama bernama **PrintXWhile**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Variabel **Sum** dan **x** dideklarasikan untuk menyimpan total penjumlahan dan nilai yang dibaca dari pengguna. Objek **Scanner** juga dideklarasikan.
4. **Inisialisasi Total Penjumlahan:** Variabel **Sum** diinisialisasi dengan nilai 0.
5. **Prompt Pengguna:** Pesan meminta pengguna untuk memasukkan nilai **x** ditampilkan.
6. **Membaca Masukan Pertama:** Nilai pertama dari pengguna dibaca dan disimpan dalam variabel **x**.
7. **Iterasi untuk Membaca Nilai:** Program menggunakan loop **while** untuk memeriksa apakah nilai **x** yang dimasukkan oleh pengguna bukan **999**. Jika bukan, nilai **x** ditambahkan ke dalam total penjumlahan **Sum**. Program terus meminta pengguna untuk memasukkan nilai **x** hingga pengguna memasukkan **999**.
8. **Periksa Kondisi Pengulangan:** Pengulangan dilanjutkan selama nilai **x** yang dimasukkan oleh pengguna tidak sama dengan **999**.
9. **Cetak Total Penjumlahan:** Setelah pengguna memasukkan nilai **999**, total penjumlahan dari nilai-nilai yang dimasukkan oleh pengguna dicetak.

Kode Program 18

```
D:\Tugas 3\Tugas3_SourceCode>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
2
9
Ke dua bilangan : a = 2 b = 9
Maksimum = 9
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 9 b = 2
```

Penjelasan :

1. **Deklarasi Kelas:** Program memiliki kelas utama yang diberi nama **SubProgram**.

2. **Fungsi maxab:** Program mendefinisikan sebuah fungsi bernama **maxab** yang mengambil dua parameter bertipe integer (**a** dan **b**). Fungsi ini bertujuan untuk mencari nilai maksimum dari dua bilangan bulat yang diberikan.
3. **Prosedur tukar:** Program mendefinisikan sebuah prosedur bernama **tukar** yang mengambil dua parameter bertipe integer (**a** dan **b**). Proses ini bertujuan untuk menukar nilai antara dua variabel.
4. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
5. **Membaca Dua Bilangan:** Program meminta pengguna untuk memasukkan dua bilangan integer. Input ini disimpan dalam variabel **a** dan **b**.
6. **Pemanggilan Fungsi:** Program mencetak nilai maksimum dari dua bilangan yang dimasukkan dengan memanggil fungsi **maxab** dan menyertakan nilai **a** dan **b** sebagai argumen.
7. **Pemanggilan Prosedur:** Program mencetak pesan untuk menukar dua bilangan dan memanggil prosedur **tukar** dengan menyertakan nilai **a** dan **b** sebagai argumen. Proses tukar ini dilakukan di dalam prosedur.

Kode Program 19

```
D:\Tugas 3\Tugas3_SourceCode>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 30
Wujud air cair
30
```

Penjelasan :

1. **Deklarasi Kelas:** Program memiliki kelas utama yang diberi nama **Tempair**.
2. **Metode Utama (main):** Metode **main** adalah titik masuk eksekusi program.
3. **Deklarasi Variabel:** Variabel **T** dideklarasikan untuk menyimpan nilai temperatur yang dimasukkan oleh pengguna. Objek **Scanner** juga dideklarasikan untuk membaca masukan dari pengguna.
4. **Prompt Pengguna:** Program mencetak pesan ke layar untuk meminta pengguna memasukkan temperatur air dalam derajat Celsius.
5. **Membaca Masukan:** Program menggunakan objek **Scanner** untuk membaca masukan berupa temperatur yang dimasukkan oleh pengguna, kemudian disimpan dalam variabel **T**.
6. **Pengkondisian:** Program menggunakan struktur pengkondisian **if-else if-else** untuk menentukan wujud air berdasarkan suhu yang dimasukkan. Terdapat tiga kondisi yang diuji:
 - Jika suhu kurang dari 0, program mencetak bahwa air berada dalam wujud beku.
 - Jika suhu di antara 0 dan 100 (inklusif), program mencetak bahwa air berada dalam wujud cair.
 - Jika suhu lebih dari 100, program mencetak bahwa air berada dalam wujud uap atau gas.

7. **Cetak Hasil:** Program mencetak hasil penentuan wujud air berdasarkan kondisi yang terpenuhi.