Tugas II Makassar, 07 Maret 2025

**PRAKTIKUM**

**PEMROGRAMAN WEB**



Nama : Farah Tsabitaputri Az Zahra

Stambuk : 13020230268

Dosen : Mardiyyah Hasnawi, S.Kom., M.T.

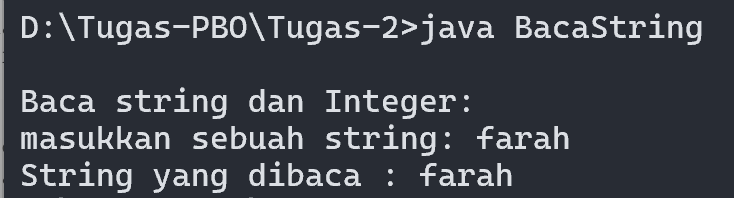
**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS MUSLIM INDONESIA**

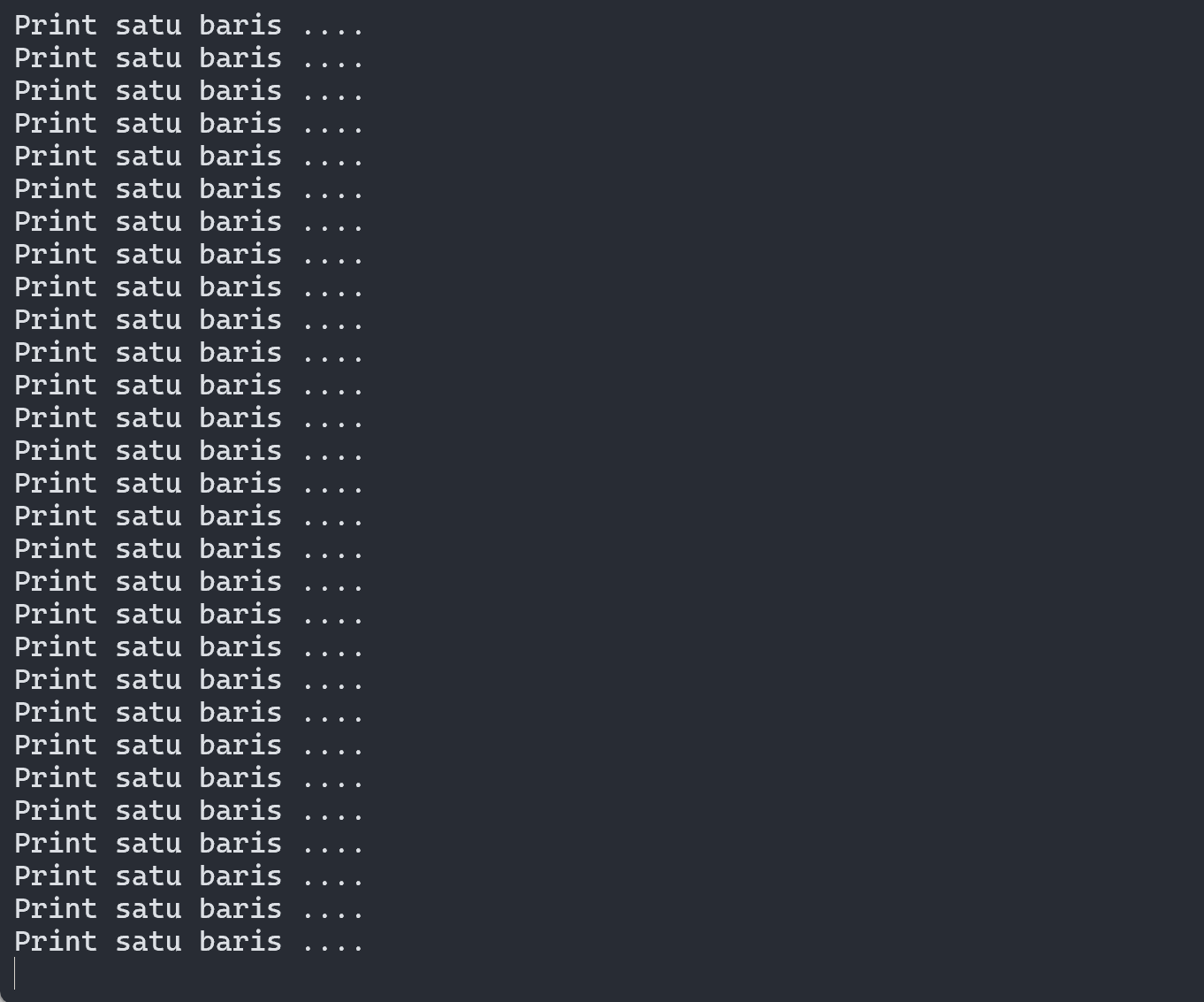
**MAKASSAR**

**2025**

1. 

Penjelasan :

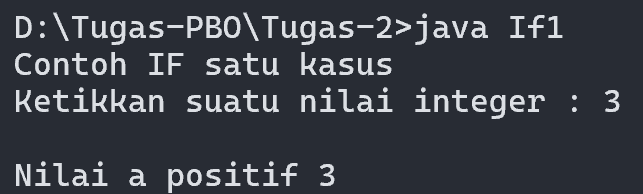
Output di atas merupakan hasil dari program sederhana yang digunakan untuk membaca input dari pengguna dalam bentuk string menggunakan BufferedReader. Pada awal program, terdapat impor beberapa library dari java.io, yaitu BufferedReader, IOException, dan InputStreamReader, yang berfungsi untuk menangani input dari pengguna dan kemungkinan kesalahan saat membaca data. Kemudian, program mendeklarasikan kelas BacaString yang berisi metode main, yang merupakan titik awal eksekusi program.

Di dalam metode main, pertama-tama, sebuah objek BufferedReader dibuat dengan cara membungkus InputStreamReader(System.in). Objek ini digunakan untuk membaca input dari keyboard dalam bentuk teks. Selanjutnya, program menampilkan pesan kepada pengguna untuk memasukkan sebuah string. Pengguna kemudian mengetikkan inputnya, dan nilai tersebut disimpan dalam variabel str menggunakan metode readLine. Setelah input dibaca, program menampilkan kembali string yang dimasukkan oleh pengguna ke layar dengan menggunakan System.out.print. Karena metode readLine dapat menyebabkan IOException, program menangani kemungkinan kesalahan tersebut dengan menambahkan throws IOException pada deklarasi metode main. Program ini cukup sederhana namun mendemonstrasikan cara dasar membaca input dari pengguna dalam bahasa Java.

Penjelasan :

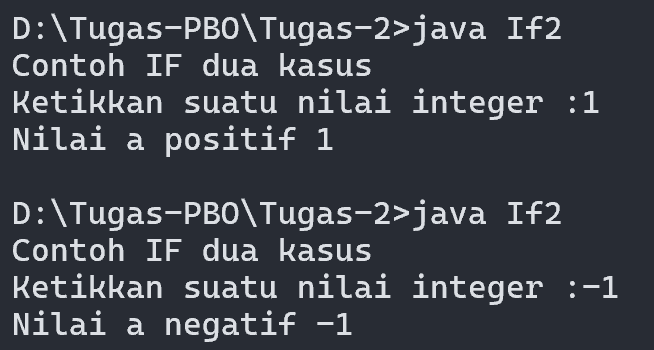
Output di atas adalah hasil dari program sederhana yang menjalankan perulangan tanpa henti (infinite loop) menggunakan perulangan while(true). Program dimulai dengan mendeklarasikan kelas ForEver, yang berisi metode main, sebagai titik awal eksekusi. Di dalam metode main, pertama-tama program mencetak pesan "Program akan looping, akhiri dengan ^c", yang memberi tahu pengguna bahwa program akan terus berjalan hingga dihentikan secara paksa menggunakan kombinasi tombol Ctrl + C di terminal atau command prompt.

Setelah itu, program memasuki perulangan while(true), yang berarti kondisi dalam while selalu benar, sehingga kode di dalamnya akan dieksekusi terus-menerus tanpa berhenti. Di dalam perulangan ini, program mencetak teks "Print satu baris ...." ke layar berulang kali. Karena tidak ada perintah untuk keluar dari perulangan, program akan terus mencetak baris tersebut tanpa henti sampai pengguna secara manual menghentikannya. Program ini merupakan contoh dari perulangan tak terbatas yang sering digunakan dalam sistem yang membutuhkan proses berjalan terus-menerus, seperti server atau program pemantauan, tetapi dalam banyak kasus, perulangan seperti ini perlu disertai dengan kondisi berhenti agar tidak menyebabkan program macet atau menggunakan sumber daya secara berlebihan.

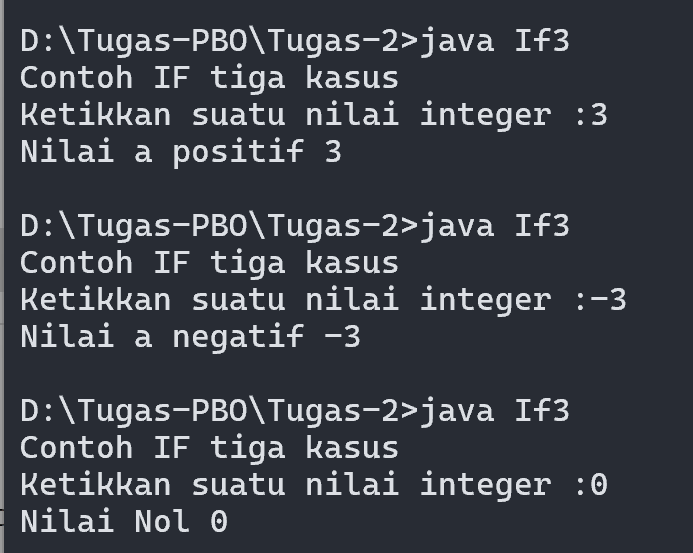
1. 

Penjelasan :

Program Java di atas adalah contoh sederhana penggunaan pernyataan if untuk satu kasus. Program ini digunakan untuk membaca sebuah bilangan bulat (integer) dari pengguna dan menampilkan nilainya hanya jika bilangan tersebut bernilai positif atau nol. Program dimulai dengan mengimpor kelas Scanner dari pustaka java.util, yang digunakan untuk membaca input dari keyboard. Kemudian, dideklarasikan kelas If1 yang berisi metode main, tempat eksekusi utama program berlangsung.Di dalam metode main, pertama-tama, program membuat objek Scanner bernama masukan untuk menerima input dari pengguna. Selanjutnya, sebuah variabel a bertipe integer dideklarasikan untuk menyimpan nilai yang diinputkan oleh pengguna. Program kemudian menampilkan teks "Contoh IF satu kasus" dan meminta pengguna untuk memasukkan sebuah bilangan integer. Input tersebut dibaca menggunakan masukan.nextInt dan disimpan dalam variabel a. Setelah nilai a diperoleh, program mengecek apakah a bernilai positif atau nol dengan menggunakan pernyataan if (a >= 0). Jika kondisi ini benar, program akan mencetak "Nilai a positif " diikuti dengan nilai a. Namun, jika a bernilai negatif, program tidak akan mencetak apa pun karena tidak ada perintah untuk menangani kondisi selain a >= 0. Dengan kata lain, hanya bilangan positif dan nol yang akan ditampilkan, sementara bilangan negatif akan diabaikan tanpa ada output.

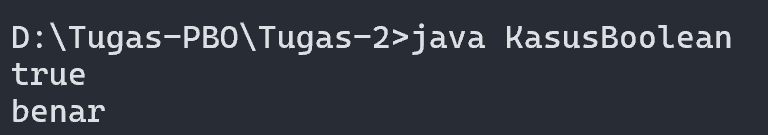
1. 

Penjelasan :

Output di atas adalah hasil dari program contoh penggunaan if-else untuk menangani dua kemungkinan kasus yang saling berlawanan atau komplementer. Program ini berfungsi untuk membaca sebuah bilangan bulat (integer) dari pengguna, lalu menentukan apakah bilangan tersebut bernilai positif atau negatif dan menampilkannya ke layar. Program diawali dengan mengimpor kelas Scanner dari pustaka java.util, yang digunakan untuk membaca input dari keyboard. Kemudian, dideklarasikan kelas If2 yang berisi metode main, tempat eksekusi program berlangsung. Di dalam metode main, variabel a bertipe integer dideklarasikan untuk menyimpan input dari pengguna. Sebuah objek Scanner bernama masukan juga dibuat untuk membaca nilai yang diketikkan pengguna. Setelah itu, program menampilkan teks "Contoh IF dua kasus" sebagai informasi awal kepada pengguna. Kemudian, pengguna diminta untuk mengetikkan sebuah bilangan integer, yang kemudian dibaca menggunakan masukan.nextInt dan disimpan dalam variabel a. Berikutnya, program melakukan pengecekan menggunakan if-else. Jika nilai a lebih besar dari atau sama dengan nol (a >= 0), maka program akan mencetak "Nilai a positif" diikuti dengan nilai a. Namun, jika nilai a kurang dari nol (a < 0), maka blok else akan dieksekusi, dan program akan mencetak "Nilai a negatif" diikuti dengan nilai a. Dengan demikian, program selalu memberikan output sesuai dengan tanda dari bilangan yang dimasukkan oleh pengguna.

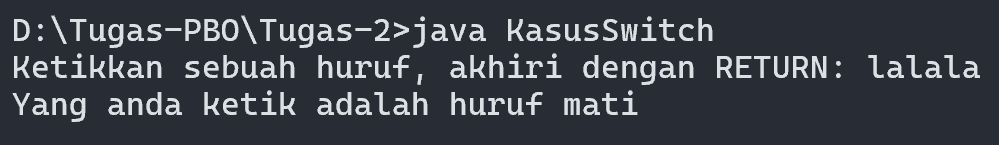
Penjelasan :

Output di atas adalah hasil dari program contoh penggunaan if-else if-else untuk menangani tiga kemungkinan kasus dalam menentukan tanda suatu bilangan. Program ini membaca sebuah bilangan bulat (integer) yang dimasukkan oleh pengguna, lalu mengevaluasi apakah bilangan tersebut positif, nol, atau negatif, kemudian menampilkan hasilnya di layar. Program diawali dengan mengimpor kelas Scanner dari pustaka java.util agar dapat membaca input dari keyboard. Setelah itu, dideklarasikan kelas If3 yang berisi metode main, tempat eksekusi utama program berlangsung. Di dalam metode main, program mendeklarasikan variabel a bertipe integer untuk menyimpan nilai yang dimasukkan oleh pengguna. Selanjutnya, sebuah objek Scanner bernama masukan dibuat untuk menerima input dari pengguna. Program kemudian menampilkan teks "Contoh IF tiga kasus" sebagai pengantar, lalu meminta pengguna untuk mengetikkan sebuah bilangan integer. Nilai yang diketikkan pengguna dibaca menggunakan masukan.nextInt dan disimpan dalam variabel a. Setelah mendapatkan input, program mengecek nilainya menggunakan if-else if-else. Jika a lebih besar dari nol (a > 0), program mencetak "Nilai a positif" diikuti dengan nilai a. Jika a sama dengan nol (a == 0), program mencetak "Nilai Nol" diikuti dengan nilai a. Jika tidak memenuhi kedua kondisi sebelumnya (berarti a < 0), program menjalankan blok else dan mencetak "Nilai a negatif" diikuti dengan nilai a. Dengan logika ini, program selalu memberikan output yang sesuai dengan tanda dari bilangan yang dimasukkan oleh pengguna, apakah itu positif, nol, atau negatif.

1. 

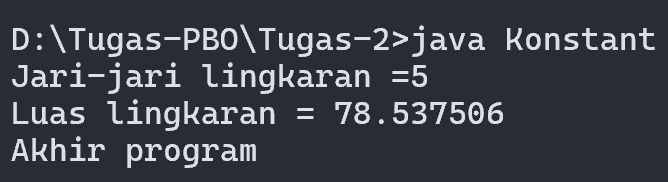
Penjelasan :

Output di atas merupakan hasil dari contoh program penggunaan ekspresi kondisional dengan tipe data boolean. Program ini menunjukkan bagaimana variabel boolean dapat digunakan dalam pernyataan if-else untuk mengontrol alur eksekusi berdasarkan nilai true atau false. Program diawali dengan mendeklarasikan kelas KasusBoolean, yang berisi metode main, sebagai titik awal eksekusi program. Di dalam metode main, terdapat deklarasi variabel boolean bernama bool, yang digunakan untuk menyimpan nilai true atau false. Pada awalnya, bool diinisialisasi dengan nilai true. Setelah itu, program menjalankan pernyataan if-else pertama untuk memeriksa nilai dari bool. Jika bool bernilai true (yang memang sudah diinisialisasi demikian), maka program mencetak "true". Jika bool bernilai false, maka program akan mencetak "false", tetapi dalam kasus ini, kondisi tersebut tidak akan terjadi karena bool sudah diatur ke true. Selanjutnya, program menjalankan pernyataan if-else kedua dengan kondisi !bool (negasi dari bool). Karena bool bernilai true, maka !bool akan menjadi false, sehingga blok if tidak dieksekusi, dan program langsung menjalankan bagian else, yang mencetak "benar". Dengan kata lain, program akan selalu mencetak "true" dan "benar", karena variabel bool diatur ke true sejak awal. Jika bool diubah menjadi false, maka output akan berubah menjadi "false" dan "salah". Program ini menggambarkan cara kerja tipe data boolean dalam pengambilan keputusan menggunakan pernyataan if-else.

1. 

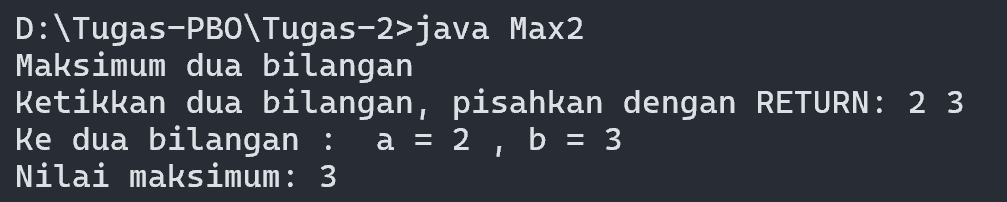
Penjelasan :

Output di atas adalahhasil dari program contoh penggunaan switch-case untuk menentukan apakah huruf yang dimasukkan oleh pengguna merupakan huruf vokal (a, u, e, i, o) atau bukan. Program ini dimulai dengan mengimpor kelas Scanner agar dapat membaca input dari keyboard. Kemudian, di dalam metode main, sebuah objek Scanner bernama masukan dibuat untuk menerima input dari pengguna. Selanjutnya, program meminta pengguna untuk mengetikkan satu huruf dan menekan tombol Enter. Huruf yang dimasukkan oleh pengguna kemudian disimpan dalam variabel cc bertipe char, yang diambil menggunakan masukan.next.charAt(0). Metode ini digunakan untuk membaca karakter pertama dari string yang diketikkan oleh pengguna. Setelah mendapatkan input, program menggunakan pernyataan switch-case untuk memeriksa apakah cc termasuk salah satu dari huruf vokal (a, u, e, i, o). Jika pengguna mengetikkan salah satu huruf tersebut, program akan mencetak pesan yang menunjukkan huruf yang dimasukkan. Setiap case diakhiri dengan break; agar program tidak menjalankan case lain setelahnya. Jika pengguna memasukkan huruf selain a, u, e, i, o, program akan menjalankan default case, yang mencetak pesan "Yang anda ketik adalah huruf mati", menunjukkan bahwa huruf tersebut bukan huruf vokal (dalam bahasa Indonesia, "huruf mati" berarti huruf konsonan). Terakhir, setelah semua proses selesai, program menutup objek Scanner dengan masukan.close; untuk mencegah kebocoran sumber daya. Dengan cara ini, program dapat mengklasifikasikan huruf yang diketikkan oleh pengguna sebagai vokal atau konsonan dengan mudah.

1. 

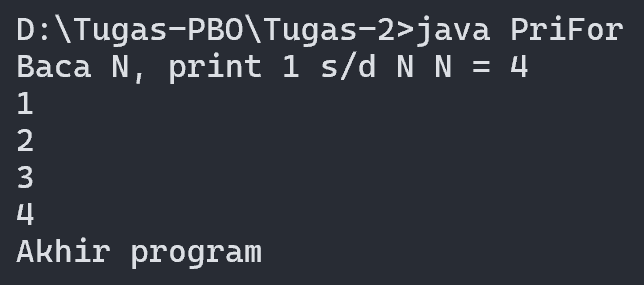
Penjelasan :

Output di atas adalah hasil dari program yang digunakan untuk menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan oleh pengguna. Program ini dimulai dengan mendeklarasikan konstanta PHI dengan nilai 3.1415, yang merupakan pendekatan nilai π (pi). Konstanta ini menggunakan kata kunci final, sehingga nilainya tidak bisa diubah selama program berjalan. Di dalam metode main, program mendeklarasikan variabel r bertipe float untuk menyimpan jari-jari lingkaran yang diinputkan oleh pengguna. Objek Scanner bernama masukan dibuat untuk membaca input dari keyboard. Program kemudian meminta pengguna memasukkan nilai jari-jari dengan menampilkan pesan "Jari-jari lingkaran =", lalu membaca input tersebut menggunakan masukan.nextFloat. Setelah mendapatkan nilai jari-jari, program menghitung luas lingkaran menggunakan rumus π × r × r, di mana PHI digunakan sebagai nilai π. Hasil perhitungan ditampilkan dengan perintah System.out.print, yang mencetak "Luas lingkaran =" diikuti hasil perhitungan. Terakhir, program mencetak pesan "Akhir program" sebagai tanda bahwa eksekusi telah selesai. Program ini menunjukkan penggunaan konstanta, cara membaca input, serta perhitungan sederhana dalam Java.

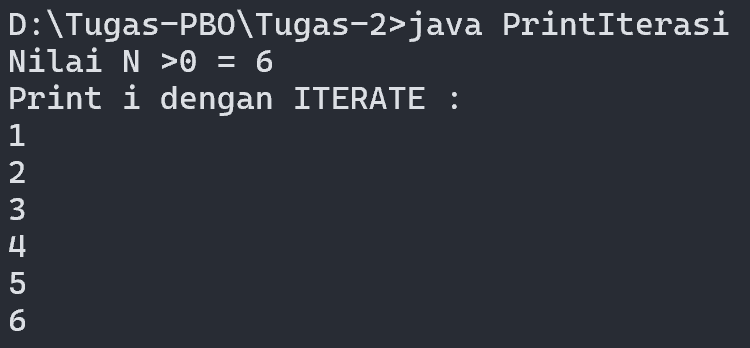
1. 

Penjelasan :

Output di atas adalah hasil dari program yang digunakan untuk menentukan bilangan terbesar dari dua bilangan yang dimasukkan oleh pengguna. Program dimulai dengan mengimpor kelas Scanner agar dapat membaca input dari keyboard. Di dalam metode main, program mendeklarasikan dua variabel integer, a dan b, untuk menyimpan dua bilangan yang diinputkan. Kemudian, objek Scanner bernama masukan dibuat untuk menerima input dari pengguna. Program menampilkan pesan "Maksimum dua bilangan" sebagai pengantar, lalu meminta pengguna memasukkan dua bilangan dengan menekan Enter setelah masing-masing bilangan. Setelah kedua bilangan dibaca menggunakan masukan.nextInt(), program menampilkan kembali nilai yang dimasukkan dalam format "Ke dua bilangan: a = ..., b = ...". Selanjutnya, program menggunakan if-else untuk menentukan bilangan terbesar. Jika a lebih besar atau sama dengan b, program mencetak "Nilai maksimum" diikuti dengan nilai a. Jika tidak (a < b), maka program mencetak "Nilai maksimum" diikuti dengan nilai b. Dengan cara ini, program dapat membandingkan dua bilangan dan menampilkan nilai yang lebih besar di antara keduanya.

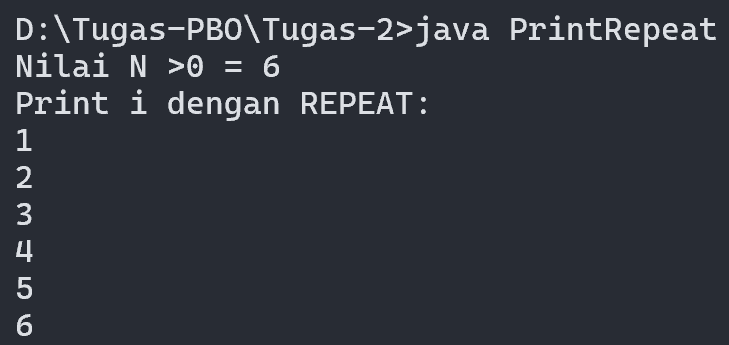
1. 

Penjelasan :

Output di atas merupakan hasil dari program yang digunakan untuk mencetak bilangan dari 1 hingga N menggunakan perulangan FOR. Program diawali dengan mengimpor kelas Scanner agar bisa membaca input dari pengguna. Di dalam metode main, program mendeklarasikan dua variabel integer, i sebagai penghitung dalam perulangan dan N sebagai batas akhir angka yang akan dicetak. Selanjutnya, program membuat objek Scanner bernama masukan untuk menerima input dari pengguna. Program meminta pengguna memasukkan nilai N dengan menampilkan pesan "Baca N, print 1 s/d N", lalu membaca input tersebut menggunakan masukan.nextInt. Setelah nilai N diperoleh, program masuk ke perulangan FOR, di mana variabel i dimulai dari 1, terus bertambah 1 setiap iterasi (i++), dan berhenti ketika mencapai N. Pada setiap iterasi, nilai i akan dicetak ke layar. Setelah perulangan selesai, program mencetak "Akhir program", menandakan bahwa eksekusi telah berakhir. Dengan cara ini, program dapat mencetak deret angka dari 1 hingga N secara otomatis sesuai input pengguna.

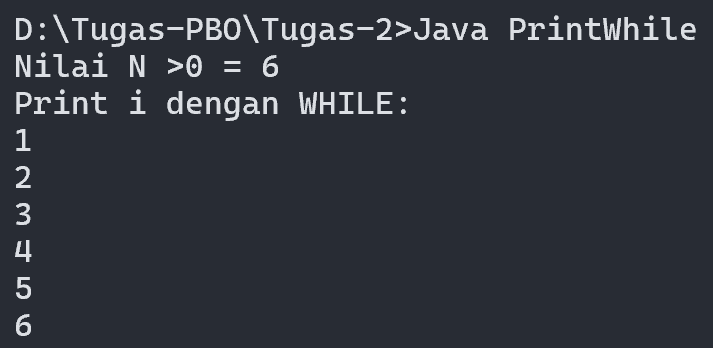
Penjelasan :

Output di atas merupakan hasil dari program yang digunakan untuk mencetak bilangan dari 1 hingga N menggunakan perulangan FOR tanpa kondisi awal dan akhir langsung (disebut infinite loop dengan break). Program dimulai dengan mengimpor kelas Scanner agar bisa membaca input dari pengguna. Di dalam metode main, program mendeklarasikan dua variabel integer: N untuk menyimpan batas akhir angka yang akan dicetak, dan i sebagai penghitung perulangan. Program kemudian membuat objek Scanner bernama masukan untuk membaca input dari pengguna. Program meminta pengguna memasukkan nilai N dengan menampilkan pesan "Nilai N >0 =", lalu membaca input tersebut menggunakan masukan.nextInt. Setelah nilai N diperoleh, variabel i diinisialisasi dengan 1 sebagai nilai awal. Selanjutnya, program menggunakan perulangan FOR tanpa batasan eksplisit (for (;;)), yang berarti akan berjalan terus-menerus hingga dihentikan secara manual. Di dalam perulangan, nilai i dicetak pada setiap iterasi. Program kemudian memeriksa apakah i sudah sama dengan N. Jika iya, maka perulangan dihentikan menggunakan break. Jika tidak, nilai i akan bertambah 1 (i++), dan perulangan akan terus berjalan. Dengan cara ini, program akan mencetak angka 1 hingga N, lalu berhenti saat mencapai N, menggunakan mekanisme break untuk keluar dari perulangan tak terbatas.

1. 

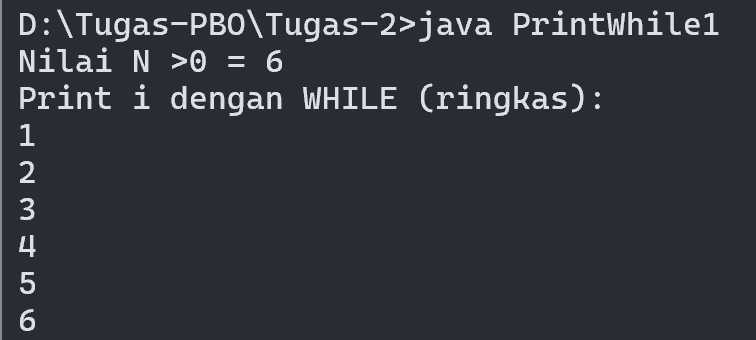
Penjelasan :

Output di atas merupakan hasil dari program yang digunakan untuk mencetak bilangan dari 1 hingga N menggunakan perulangan DO-WHILE. Program diawali dengan mengimpor kelas Scanner agar dapat membaca input dari pengguna. Di dalam metode main(), terdapat dua variabel integer: N untuk menyimpan batas akhir angka yang akan dicetak, dan i sebagai penghitung perulangan. Program kemudian membuat objek Scanner bernama masukan untuk membaca input dari pengguna. Program meminta pengguna memasukkan nilai N dengan menampilkan pesan "Nilai N >0 =", lalu membaca input tersebut menggunakan masukan.nextInt(). Setelah nilai N diperoleh, variabel i diinisialisasi dengan 1 sebagai nilai awal. Selanjutnya, program masuk ke dalam perulangan DO-WHILE. Pada setiap iterasi, program mencetak nilai i, lalu menaikkan nilainya sebanyak 1 (i++). Perulangan ini akan terus berjalan selama kondisi i <= N terpenuhi. Karena menggunakan DO-WHILE, program akan selalu mengeksekusi perintah setidaknya satu kali, bahkan jika nilai N yang dimasukkan pengguna kurang dari 1. Dengan cara ini, program dapat mencetak angka dari 1 hingga N, menggunakan perulangan DO-WHILE yang memastikan proses berjalan minimal sekali sebelum mengecek kondisi penghentian.

1. 

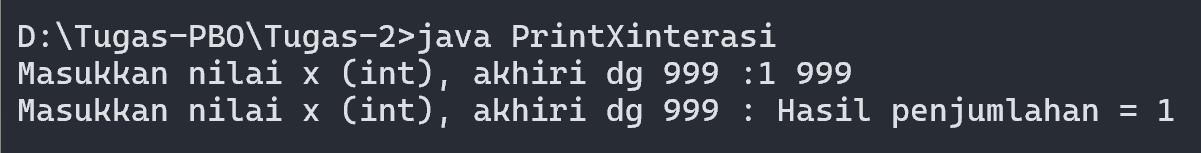
Penjelasan :

Output di atas merupakan hasil dari program yang digunakan untuk mencetak bilangan dari 1 hingga N menggunakan perulangan WHILE. Program diawali dengan mengimpor kelas Scanner agar bisa membaca input dari pengguna. Di dalam metode main(), terdapat dua variabel integer: N sebagai batas akhir angka yang akan dicetak, dan i sebagai penghitung perulangan. Program kemudian membuat objek Scanner bernama masukan untuk membaca input dari pengguna. Program meminta pengguna memasukkan nilai N dengan menampilkan pesan "Nilai N >0 =", lalu membaca input tersebut menggunakan masukan.nextInt(). Setelah nilai N diperoleh, variabel i diinisialisasi dengan 1 sebagai angka awal yang akan dicetak. Selanjutnya, program masuk ke dalam perulangan WHILE yang memiliki kondisi i <= N. Selama kondisi ini terpenuhi, program akan mencetak nilai i. Setelah mencetak angka, nilai i akan bertambah 1 (i++), sehingga perulangan terus berjalan sampai i melebihi N, yang akan menyebabkan perulangan berhenti. Dengan cara ini, program akan mencetak angka dari 1 hingga N menggunakan perulangan WHILE, yang cocok digunakan ketika jumlah iterasi belum diketahui sebelumnya tetapi bergantung pada suatu kondisi tertentu.

1. 

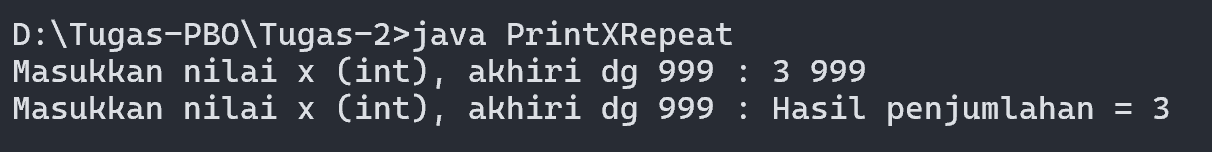
Penjelasan :

Output di atas merupakan hasil dari program yang digunakan untuk mencetak bilangan dari 1 hingga N menggunakan perulangan WHILE dengan cara yang lebih ringkas. Program diawali dengan mengimpor kelas Scanner agar dapat membaca input dari pengguna.Di dalam metode main, terdapat dua variabel integer: N sebagai batas akhir angka yang akan dicetak, dan i yang diinisialisasi langsung dengan 1 sebagai nilai awal. Program kemudian membuat objek Scanner bernama masukan untuk membaca input dari pengguna.Program meminta pengguna memasukkan nilai N dengan menampilkan pesan "Nilai N >0 =", lalu membaca input tersebut menggunakan masukan.nextInt. Setelah itu, program mencetak pesan "Print i dengan WHILE (ringkas):" sebagai penanda sebelum mencetak angka.Selanjutnya, program menggunakan perulangan WHILE dengan kondisi i <= N. Selama kondisi ini terpenuhi, program akan mencetak nilai i, lalu langsung menambah nilai i sebesar 1 dalam satu baris (System.out.println(i++);). Dengan cara ini, program lebih ringkas karena nilai i bertambah langsung dalam perintah cetak, sehingga tidak perlu menuliskan i++ di baris terpisah.Perulangan akan berhenti ketika nilai i melebihi N, dan program selesai. Metode ini membuat kode lebih sederhana namun tetap berfungsi dengan baik untuk mencetak bilangan dari 1 hingga N.

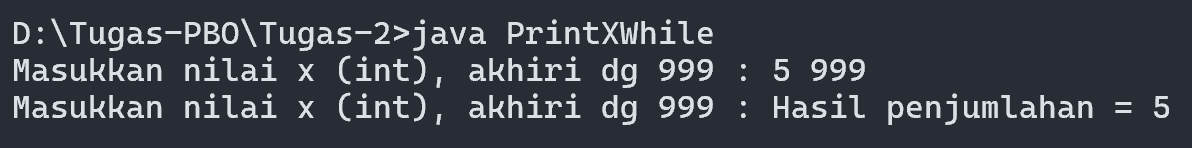
1. 

Penjelasan :

Output di atas merupakan hasil dari program yang berfungsi untuk menjumlahkan beberapa bilangan integer yang dimasukkan oleh pengguna, dengan proses iterasi menggunakan perulangan for (;;) (loop tak terbatas).Pertama, program mengimpor kelas Scanner agar dapat membaca input dari pengguna. Di dalam metode main, terdapat dua variabel utama: Sum yang digunakan untuk menyimpan hasil penjumlahan (diinisialisasi dengan 0) dan x sebagai variabel penyimpan angka yang dimasukkan pengguna. Program kemudian meminta pengguna memasukkan angka pertama dengan menampilkan pesan "Masukkan nilai x (int), akhiri dg 999 :".Jika angka pertama yang dimasukkan adalah 999, program langsung menampilkan "Kasus kosong" dan berhenti tanpa melakukan perhitungan. Namun, jika angka pertama bukan 999, program mulai menjumlahkan angka tersebut ke dalam variabel Sum.Selanjutnya, program masuk ke dalam perulangan for (;;) yang akan terus berjalan tanpa batas hingga pengguna memasukkan 999. Di dalam loop ini, program terus meminta pengguna memasukkan angka baru, dan jika angka yang dimasukkan bukan 999, maka angka tersebut akan dijumlahkan ke dalam Sum. Jika pengguna memasukkan 999, perulangan langsung dihentikan dengan perintah break.Terakhir, setelah perulangan selesai, program mencetak hasil penjumlahan seluruh angka yang telah dimasukkan pengguna sebelum 999. Dengan cara ini, program dapat membaca dan menjumlahkan angka sebanyak yang diinginkan oleh pengguna hingga mereka memutuskan untuk berhenti dengan memasukkan 999.

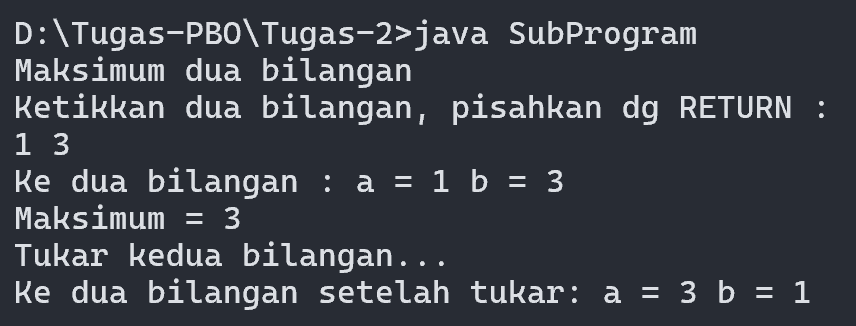
1. 

Penjelasan :

 Output di atas merupakan hasil dari program yang berfungsi untuk menjumlahkan beberapa bilangan integer yang dimasukkan oleh pengguna menggunakan perulangan do-while. Pertama, program mengimpor kelas Scanner untuk membaca input dari pengguna. Di dalam metode main, terdapat dua variabel utama: Sum untuk menyimpan hasil penjumlahan dan x sebagai variabel penyimpan angka yang dimasukkan pengguna. Program meminta pengguna memasukkan angka pertama dengan menampilkan pesan "Masukkan nilai x (int), akhiri dg 999 :". Jika angka pertama yang dimasukkan adalah 999, maka program akan langsung mencetak "Kasus kosong" dan berhenti tanpa melakukan proses penjumlahan. Namun, jika angka pertama bukan 999, maka program mulai melakukan penjumlahan menggunakan perulangan do-while. Dalam perulangan do-while, setiap angka yang dimasukkan akan ditambahkan ke dalam Sum, lalu program meminta pengguna memasukkan angka berikutnya. Proses ini terus berulang hingga pengguna memasukkan 999, yang menjadi kondisi penghentian perulangan. Setelah perulangan selesai, program mencetak hasil penjumlahan seluruh angka yang telah dimasukkan sebelum 999Dengan cara ini, program memastikan bahwa minimal satu angka dijumlahkan sebelum menghentikan perulangan, selama angka pertama yang dimasukkan bukan 999.

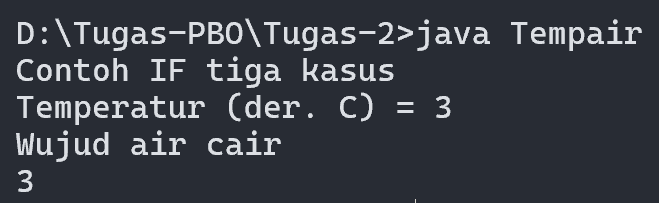
Penjelasan :

Output di atas merupakan hasil dari program yang berfungsi untuk menjumlahkan sejumlah bilangan integer yang dimasukkan oleh pengguna menggunakan perulangan while. Pertama, program mengimpor kelas Scanner untuk membaca input dari pengguna. Di dalam metode main, terdapat dua variabel utama: Sum untuk menyimpan hasil penjumlahan dan x sebagai variabel penyimpan angka yang dimasukkan pengguna.Program meminta pengguna memasukkan angka pertama dengan menampilkan pesan "Masukkan nilai x (int), akhiri dg 999 :". Jika angka yang dimasukkan bukan 999, maka program masuk ke dalam perulangan while.Di dalam perulangan while, setiap angka yang dimasukkan akan langsung ditambahkan ke dalam Sum. Setelah itu, program kembali meminta pengguna memasukkan angka berikutnya. Proses ini akan terus berulang hingga pengguna memasukkan 999, yang menjadi tanda bahwa proses perulangan harus berhenti.Setelah perulangan selesai, program mencetak hasil penjumlahan seluruh angka yang telah dimasukkan sebelum 999. Dengan cara ini, program dapat menjumlahkan sekumpulan angka yang diinput oleh pengguna secara berulang, tanpa batasan jumlah angka yang dapat dijumlahkan.

1. 

Penjelasan :

Output di atas merupakan hasil dari program yang merupakan contoh penggunaan prosedur dan fungsi dalam Java untuk mengolah dua bilangan bulat yang dimasukkan oleh pengguna. Program ini memiliki dua metode tambahan selain metode utama main, yaitu fungsi maxab(int a, int b) dan prosedur tukar(int a, int b). Fungsi maxab(int a, int b) digunakan untuk mencari bilangan terbesar antara dua bilangan yang dimasukkan. Fungsi ini menggunakan operator ternary (? :) untuk membandingkan dua bilangan dan mengembalikan nilai yang lebih besar. Misalnya, jika pengguna memasukkan angka 5 dan 8, maka fungsi ini akan mengembalikan nilai 8 sebagai bilangan terbesar. Selain itu, terdapat prosedur tukar(int a, int b) yang bertujuan untuk menukar nilai dari dua bilangan yang diberikan. Prosedur ini menggunakan variabel sementara (temp) untuk menukar nilai a dan b. Namun, karena Java menggunakan pass by value untuk tipe data primitif, perubahan nilai dalam prosedur ini hanya berlaku di dalam prosedur itu sendiri dan tidak mempengaruhi nilai a dan b di dalam metode main. Akibatnya, meskipun prosedur menampilkan hasil pertukaran, nilai sebenarnya di dalam main() tetap tidak berubah. Jalannya program dimulai dengan meminta pengguna memasukkan dua bilangan integer. Setelah itu, program menampilkan bilangan yang telah dimasukkan, kemudian memanggil fungsi maxab() untuk menentukan bilangan terbesar dan mencetak hasilnya. Selanjutnya, program memanggil prosedur tukar(), tetapi karena Java menggunakan pass by value, pertukaran nilai tidak benar-benar terjadi dalam main(). Jika ingin pertukaran nilai benar-benar terjadi di dalam main(), maka harus menggunakan array atau objek sebagai parameter agar Java menggunakan konsep pass by reference.

1. 

Penjelasan :

Output di atas merupakan hasil dari program contoh penggunaan percabangan IF dengan tiga kasus untuk menentukan wujud air berdasarkan suhu yang dimasukkan oleh pengguna. Program ini pertama-tama meminta pengguna untuk memasukkan temperatur air dalam derajat Celsius melalui input keyboard. Setelah itu, program menggunakan struktur if-else if untuk mengevaluasi suhu yang dimasukkan dan menentukan apakah air berada dalam keadaan beku, cair, atau gas. Jika suhu yang dimasukkan kurang dari 0 derajat Celsius, maka program akan mencetak bahwa air berada dalam keadaan beku. Jika suhu berada dalam rentang 0 hingga 100 derajat Celsius (termasuk 0 dan 100), maka program akan mencetak bahwa air berada dalam keadaan cair. Sedangkan jika suhu lebih dari 100 derajat Celsius, program akan mencetak bahwa air berada dalam keadaan uap atau gas. Sebagai contoh, jika pengguna memasukkan suhu -5, program akan menampilkan "Wujud air beku". Jika pengguna memasukkan suhu 50, program akan menampilkan "Wujud air cair". Dan jika pengguna memasukkan suhu 150, program akan menampilkan "Wujud air uap/gas". Program ini mencerminkan konsep dasar dalam ilmu fisika mengenai perubahan wujud zat dan menunjukkan bagaimana struktur if-else dapat digunakan untuk membuat keputusan dalam pemrograman.