Tugas 2 Makassar, 07 Maret 2025

**PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Siti Safira Tawetubun

Stambuk : 13020230217

Frekuensi : TI\_ SD-13 (B3)

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

**PROGRAM STUDI TEKNIK INFORMATIKA**

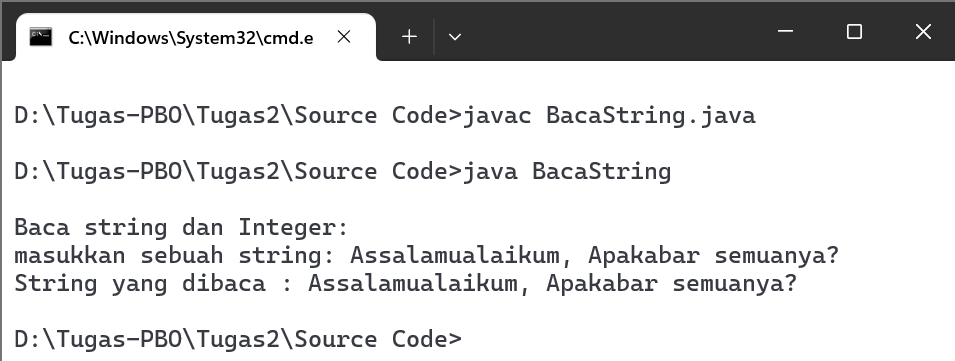
**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS MUSLIM INDONESIA**

**MAKASSAR**

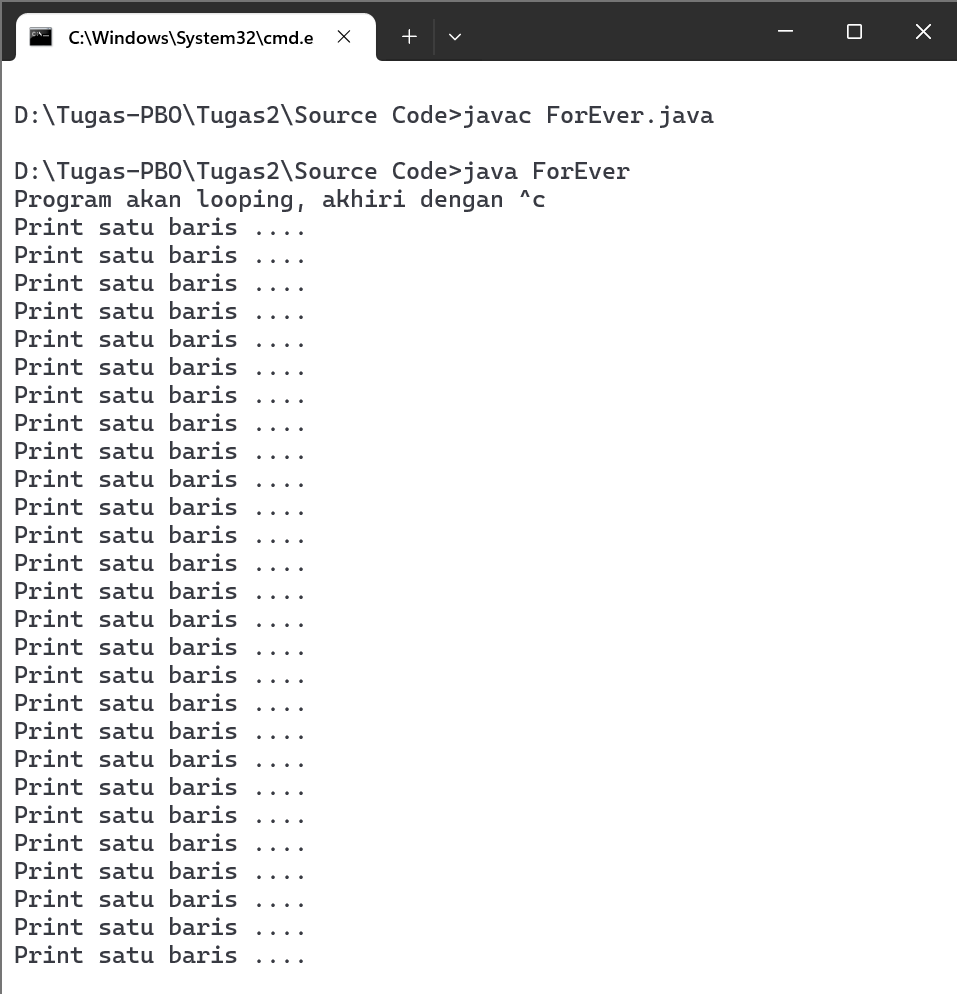
**2025**

1. Output



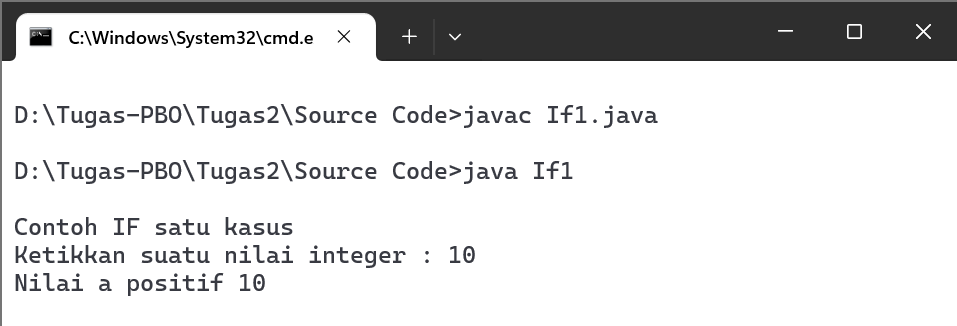
* Deklarasi dan Inisialisasi
* String str : mendeklarasikan variabel str dengan tipe data String. Variabel str ini belum di inisialisasi, inisialisasi dilakukan ketika user memberikan inputan.
* BufferedReader datAIn = new BufferedReader (new InputStreamReader (System.in)); : mendeklarasikan variabel datAIn dengan tipe data BufferedReader, digunakan untuk membaca teks dari inputan. datAIn di inisialisasi dengan membuat objek Buffered Reader baru. Objek ini menggunakan Input Stream Reader untuk membaca dari System.in (standard input, biasanya keyboard). Ini menyiapkan BufferedReader untuk membaca input berbasis karakter dari pengguna.
* Input
* str = datAIn.readLine();: Baris ini adalah tempat program menerima input dari pengguna.
* datAIn.readLine(): Membaca satu baris teks lengkap dari input stream (dalam hal ini, dari keyboard). Input akan dibaca hingga menemukan karakter newline (Enter).
* str = ...: Hasil pembacaan (string yang dimasukkan pengguna) disimpan dalam variabel str.
* Output
* System.out.print("String yang dibaca : " + str + "\n");: Baris ini menampilkan output ke konsol.
* System.out.print(...): Mencetak teks ke standard output (biasanya layar konsol).
* "String yang dibaca : " + str: Menggabungkan string literal "String yang dibaca : " dengan nilai dari variabel str. Ini menghasilkan string yang akan ditampilkan.

1. Output



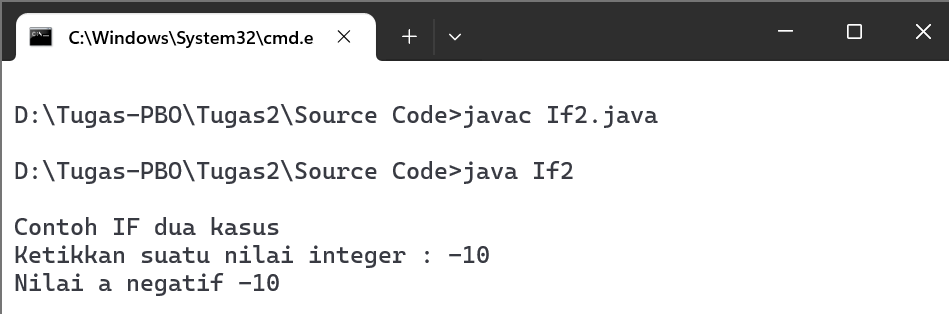
* Proses Looping
* while (true) { ... }: Memulai loop tak terbatas. Kondisi true selalu terpenuhi, sehingga loop akan terus berjalan selamanya. Karena kondisi loop selalu benar, blok kode di dalamnya akan dieksekusi terus-menerus tanpa henti.
* Output
* System.out.println("Program akan looping, akhiri dengan ^c");: Menampilkan pesan ke konsol yang memberi tahu pengguna bahwa program akan melakukan looping tanpa henti. Cara menghentikan program (menggunakan kombinasi tombol Ctrl + C).
* System.out.print("Print satu baris ....\n");: Menampilkan teks "Print satu baris ...." ke konsol setiap kali loop dieksekusi. Teks ini akan dicetak berulang kali selama program berjalan, dan setiap kali loop dijalankan, output baru akan ditampilkan di konsol.

1. Output



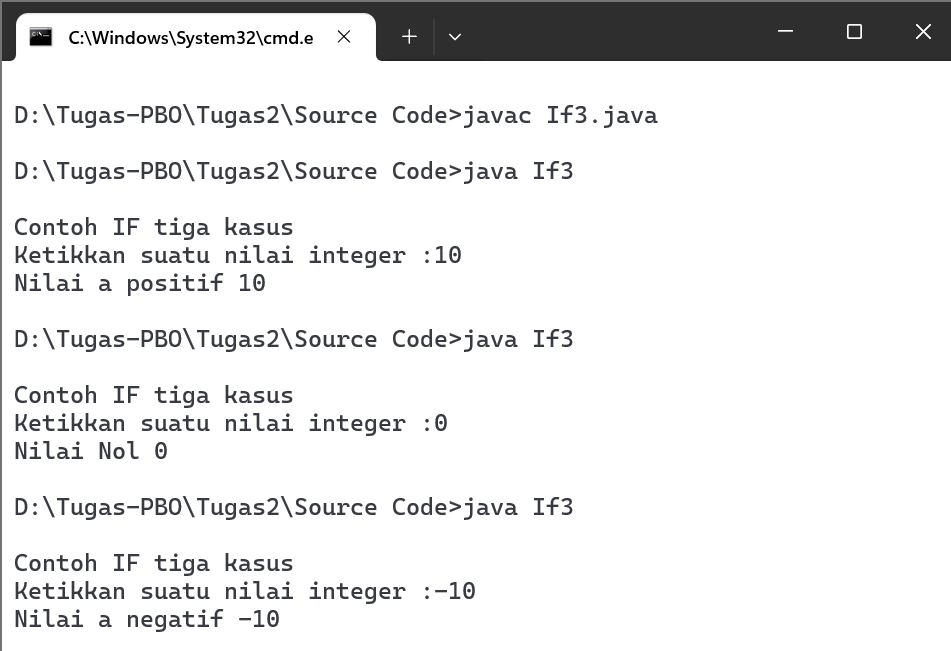
* Deklarasi Variabel
* Scanner masukan = new Scanner(System.in);: **Deklarasi dan Inisialisasi**: Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* int a;: **Deklarasi**: Mendeklarasikan variabel a dengan tipe data int (integer) yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
* Input dan output
* System.out.print("\nContoh IF satu kasus \n");:Menampilkan pesan ke konsol yang menjelaskan bahwa program ini adalah contoh penggunaan pernyataan IF.
* System.out.print("Ketikkan suatu nilai integer : ");: Meminta pengguna untuk memasukkan sebuah nilai integer.
* a = masukan.nextInt();: Membaca input integer dari pengguna menggunakan objek masukan. Nilai yang dimasukkan oleh pengguna disimpan dalam variabel a.
* Proses If dan Output
* if (a >= 0) { ... }: Memeriksa apakah nilai variabel a lebih besar dari atau sama dengan 0. Jika kondisi ini terpenuhi (artinya nilai a adalah positif atau nol), maka blok kode di dalam pernyataan IF akan dieksekusi.
* System.out.print("Nilai a positif " + a + "\n");: Menampilkan pesan ke konsol jika nilai a adalah positif. Pesan ini mencetak nilai dari variabel a yang telah dimasukkan oleh pengguna, dengan keterangan bahwa nilainya positif.

1. Output



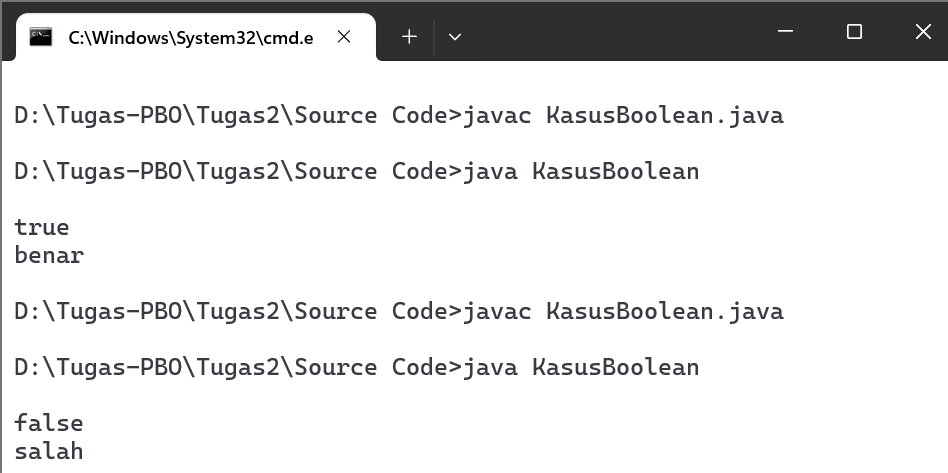
* Deklarasi Variabel
* int a;: Mendeklarasikan variabel a dengan tipe data int (integer) yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
* Scanner masukan = new Scanner(System.in);: Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* Input dan output
* System.out.print("\nContoh IF dua kasus \n");: Menampilkan pesan ke konsol yang menjelaskan bahwa program ini adalah contoh penggunaan pernyataan IF dengan dua kasus.
* System.out.print("Ketikkan suatu nilai integer : ");: Meminta pengguna untuk memasukkan sebuah nilai integer.
* a = masukan.nextInt();: Membaca input integer dari pengguna menggunakan objek masukan. Nilai yang dimasukkan oleh pengguna disimpan dalam variabel a.
* Proses If dan Output
* if (a >= 0) { ... }: Memeriksa apakah nilai variabel a lebih besar dari atau sama dengan 0. Jika kondisi ini terpenuhi (artinya nilai a adalah positif atau nol), maka blok kode di dalam pernyataan IF akan dieksekusi.
* System.out.println("Nilai a positif " + a);: Menampilkan pesan ke konsol jika nilai a adalah positif atau nol. Pesan ini mencetak nilai dari variabel a yang telah dimasukkan oleh pengguna, dengan keterangan bahwa nilainya positif.
* else { ... }:Jika kondisi di dalam pernyataan IF tidak terpenuhi (artinya nilai a kurang dari 0), maka blok kode di dalam ELSE akan dieksekusi.
* System.out.println("Nilai a negatif " + a);: Menampilkan pesan ke konsol jika nilai a adalah negatif. Pesan ini mencetak nilai dari variabel a, dengan keterangan bahwa nilainya negatif.

1. Output



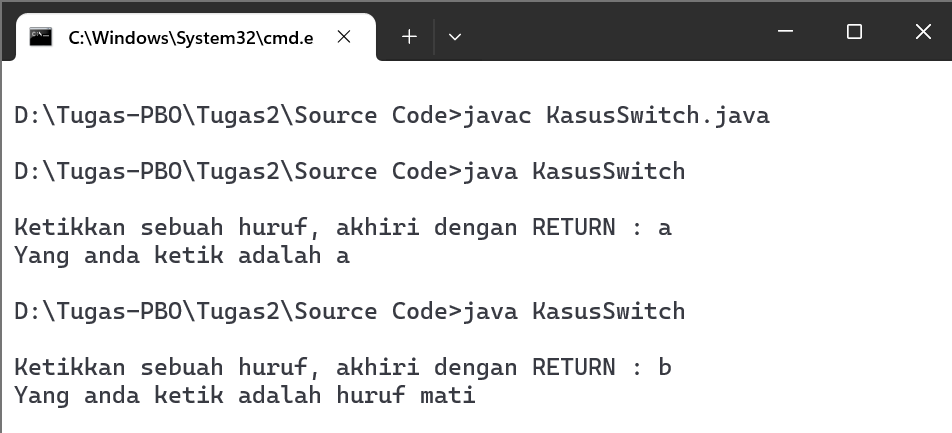
* Deklarasi dan Inisialisasi Variabel
* int a;: Mendeklarasikan variabel a dengan tipe data int (integer) yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
* Scanner masukan = new Scanner(System.in);: Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* Input dan Output
* System.out.print("\nContoh IF tiga kasus \n");: Menampilkan pesan ke konsol yang menjelaskan bahwa program ini adalah contoh penggunaan pernyataan IF dengan tiga kasus.
* System.out.print("Ketikkan suatu nilai integer : ");: Meminta pengguna untuk memasukkan sebuah nilai integer.
* a = masukan.nextInt();:Membaca input integer dari pengguna menggunakan objek masukan. Nilai yang dimasukkan oleh pengguna disimpan dalam variabel a.
* Proses dan Output
* if (a > 0) { ... }:Memeriksa apakah nilai variabel a lebih besar dari 0. Jika kondisi ini terpenuhi (artinya nilai a adalah positif), maka blok kode di dalam pernyataan IF akan dieksekusi.
  + System.out.println("Nilai a positif " + a);: Menampilkan pesan ke konsol jika nilai a adalah positif. Pesan ini mencetak nilai dari variabel a yang telah dimasukkan oleh pengguna, dengan keterangan bahwa nilainya positif.
* else if (a == 0) { ... }:Memeriksa apakah nilai variabel a sama dengan 0. Jika kondisi ini terpenuhi, maka blok kode di dalam ELSE IF akan dieksekusi.
  + System.out.println("Nilai Nol " + a);: Menampilkan pesan ke konsol jika nilai a adalah nol. Pesan ini mencetak nilai dari variabel a, dengan keterangan bahwa nilainya adalah nol.
* else { ... }:Jika kedua kondisi sebelumnya tidak terpenuhi (artinya nilai a kurang dari 0), maka blok kode di dalam ELSE akan dieksekusi.
  + System.out.println("Nilai a negatif " + a);: pesan ke konsol jika nilai a adalah negatif. Pesan ini mencetak nilai dari variabel a, dengan keterangan bahwa nilainya negatif.

1. Output



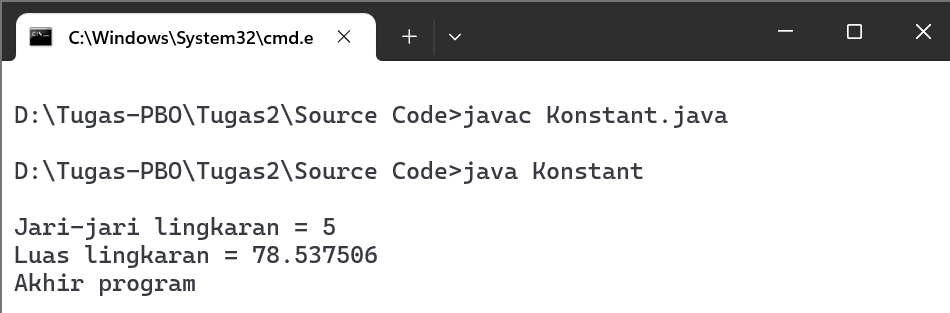
* Deklarasi dan Inisialisasi Variabel
* boolean bool;: Mendeklarasikan variabel bool dengan tipe data boolean, yang hanya dapat menyimpan nilai true atau false.
* bool = true;: Menginisialisasi variabel bool dengan nilai true.
* Output
* if (bool) { System.out.print("\ntrue\n"); } else System.out.print ("\nfalse\n");:Memeriksa nilai dari variabel bool. Jika nilai bool adalah true maka program akan mencetak “true” ke konsol. Jika tidak, program akan mencetak “false”.
* if (!bool) { System.out.print("salah\n");}else System.out.print ("benar\n");: Memeriksa negasi dari nilai variabel bool. Operator negasi (!) membalikkan nilai boolean. Jika bool adalah true, maka !bool akan menjadi false.

1. Output



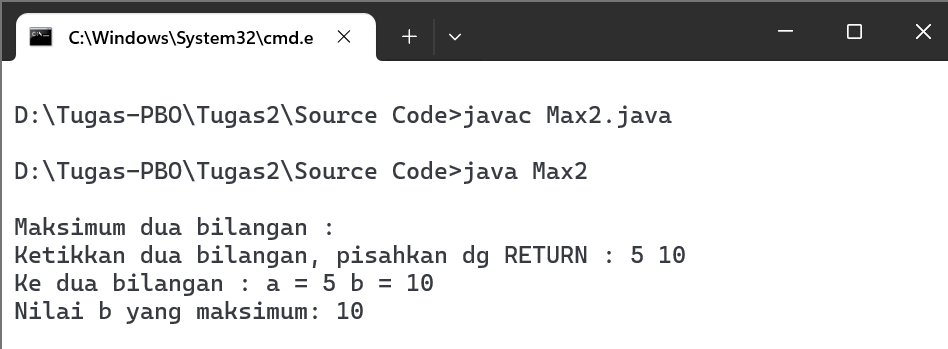
* Deklarasi Variabel
* char cc;:Mendeklarasikan variabel cc dengan tipe data char, yang akan digunakan untuk menyimpan karakter yang dimasukkan oleh pengguna.
* Scanner masukan = new Scanner(System.in);: Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* Input dan Output
* System.out.print("Ketikkan sebuah huruf, akhiri dengan RETURN : ");:Menampilkan pesan ke konsol yang meminta pengguna untuk memasukkan sebuah huruf.
* cc = masukan.next().charAt(0);: Membaca input dari pengguna sebagai string dan mengambil karakter pertama dari string tersebut. Nilai karakter yang dimasukkan oleh pengguna disimpan dalam variabel cc.
* Proses Switch
* switch (cc) { ... }:Memeriksa nilai dari variabel cc untuk menentukan tindakan yang sesuai berdasarkan karakter yang dimasukkan.
* case 'a': { ... }:Jika nilai dari cc adalah 'a', maka blok kode ini akan dieksekusi. Mencetak "Yang anda ketik adalah a".
* case 'u': { ... }:Jika nilai dari cc adalah 'u', maka blok kode ini akan dieksekusi. Mencetak "Yang anda ketik adalah u".
* case 'e': { ... }:Jika nilai dari cc adalah 'e', maka blok kode ini akan dieksekusi. Mencetak "Yang anda ketik adalah e".
* case 'i': { ... }: Jika nilai dari cc adalah 'i', maka blok kode ini akan dieksekusi. Mencetak "Yang anda ketik adalah i".
* case 'o': { ... }:Jika nilai dari cc adalah 'o', maka blok kode ini akan dieksekusi. Mencetak "Yang anda ketik adalah o".
* default: { ... }:Jika nilai dari cc tidak cocok dengan salah satu kasus di atas, maka blok kode dalam default akan dieksekusi. Mencetak "Yang anda ketik adalah huruf mati".

1. Output



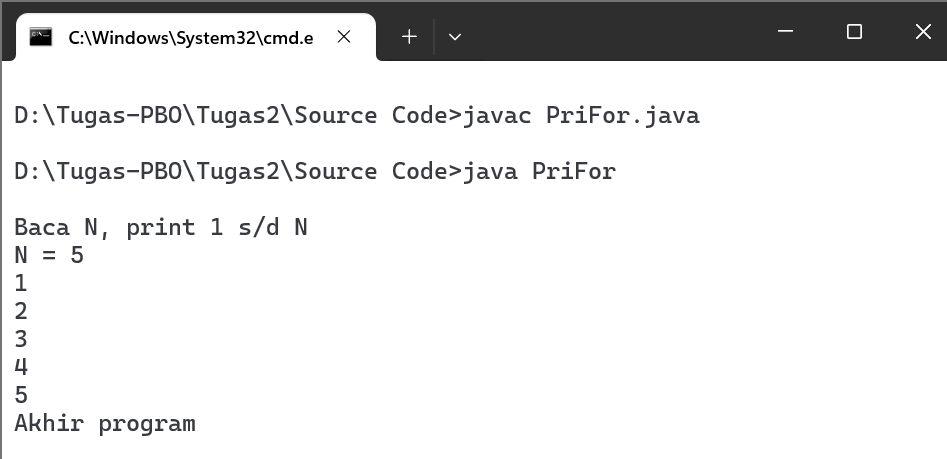
* Deklarasi dan Inisialisasi
* final float PHI = 3.1415f;
  + final: Menandakan bahwa PHI adalah konstanta (nilai tidak bisa diubah setelah inisialisasi).
  + 3.1415f: Nilai konstanta π (phi) yang digunakan untuk menghitung luas lingkaran.
  + f: Menunjukkan bahwa nilai tersebut bertipe float.
* float r;: Mendeklarasikan variabel r untuk menyimpan jari-jari lingkaran (tipe data float).
* Scanner masukan = new Scanner(System.in); : Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* Input dan Output
* System.out.print("\nJari-jari lingkaran = "); r = masukan.nextFloat(); : Menampilkan pesan untuk meminta input jari-jari lingkaran. Membaca input pengguna menggunakan nextFloat() dan menyimpannya ke variabel r.
* System.out.print("Luas lingkaran = " + (PHI \* r \* r) + "\n"); : Menghitung luas lingkaran dengan rumus π × r² dan langsung mencetak hasilnya dalam satu baris. PHI \* r \* r: Ekspresi matematika untuk menghitung luas.
* System.out.print("Akhir program \n"); : Menampilkan pesan penutup bahwa program selesai dijalankan.

1. Output



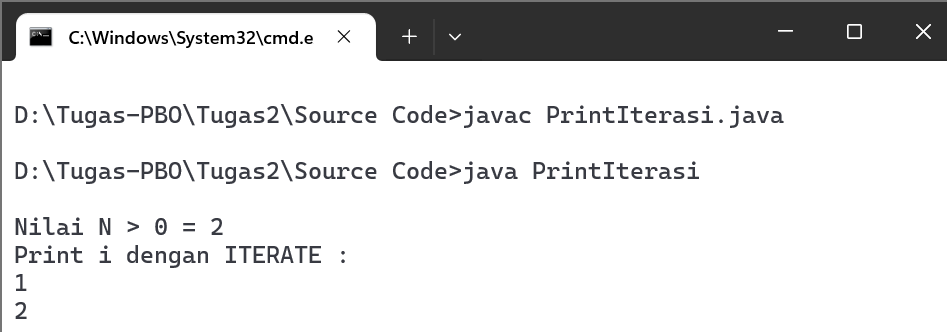
* Deklarasi dan Inisialisasi
* int a, b;: Mendeklarasikan dua variabel bertipe data int (integer) yang akan digunakan untuk menyimpan dua bilangan yang dimasukkan oleh pengguna.
* Scanner masukan = new Scanner(System.in);:Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* Input dan Output
* System.out.print("Ketikkan dua bilangan, pisahkan dg RETURN : "); : Menampilkan pesan ke konsol yang meminta pengguna untuk memasukkan dua bilangan.
* a = masukan.nextInt(); dan b = masukan.nextInt(); :  Membaca dua bilangan integer dari input pengguna menggunakan objek masukan. Nilai yang dimasukkan disimpan dalam variabel a dan b.
* System.out.println("Ke dua bilangan : a = " + a + " b = " + b); : Menampilkan kedua bilangan yang telah dimasukkan oleh pengguna ke konsol.

1. Output



* Deklarasi dan Inisialisasi
* int i, N; Mendeklarasikan dua variabel bertipe data int (integer):
  + i: Variabel counter untuk loop.
  + N: Variabel untuk menyimpan batas atas deret yang akan dicetak.
* Scanner masukan = new Scanner(System.in); Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* Input dan Output
* System.out.print("\nBaca N, print 1 s/d N "); System.out.print("\nN = "); : Menampilkan pesan ke konsol yang meminta pengguna untuk memasukkan nilai N.
* N = masukan.nextInt(); : Membaca input integer dari pengguna menggunakan objek masukan dan menyimpan nilai yang dimasukkan ke dalam variabel N.
* for (i = 1; i <= N; i++){ System.out.println(i); }; : Mencetak deret bilangan dari 1 hingga N ke konsol menggunakan loop for.
* i = 1: Inisialisasi variabel i dengan nilai 1.
* i <= N: Kondisi yang diperiksa pada setiap iterasi loop. Loop akan terus berjalan selama i kurang dari atau sama dengan N.
* i++: Increment variabel i setelah setiap iterasi.
* System.out.println(i);: Mencetak nilai i (bilangan dalam deret) ke konsol pada setiap iterasi loop.

1. Output



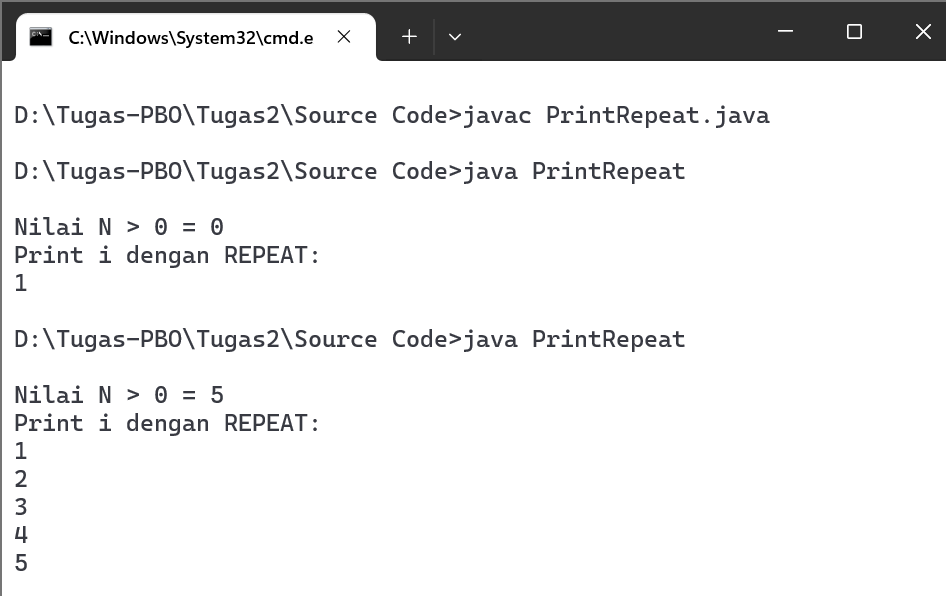
* Deklarasi dan Inisialisasi
* int N; : Mendeklarasikan variabel N bertipe data int (integer) yang akan digunakan untuk menyimpan batas atas deret yang akan dicetak.
* int i; : Mendeklarasikan variabel i bertipe data int (integer) yang akan digunakan sebagai counter untuk iterasi.
* Scanner masukan = new Scanner(System.in); : Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* i = 1; : Menginisialisasi variabel counter i dengan nilai 1.
* Input dan Output
* System.out.print("\nNilai N > 0 = "); : Menampilkan pesan ke konsol yang meminta pengguna untuk memasukkan nilai N yang lebih besar dari 0.
* N = masukan.nextInt();: Membaca input integer dari pengguna menggunakan objek masukan dan menyimpan nilai yang dimasukkan ke dalam variabel N.
* System.out.print("Print i dengan ITERATE : \n"); : Menampilkan pesan bahwa program akan mencetak nilai i dengan iterasi.

for (;;) { System.out.println(i); if (i == N) break; else { i++;}

} : Mencetak deret bilangan dari 1 hingga N ke konsol menggunakan loop for tanpa kondisi awal, kondisi perulangan, atau increment.

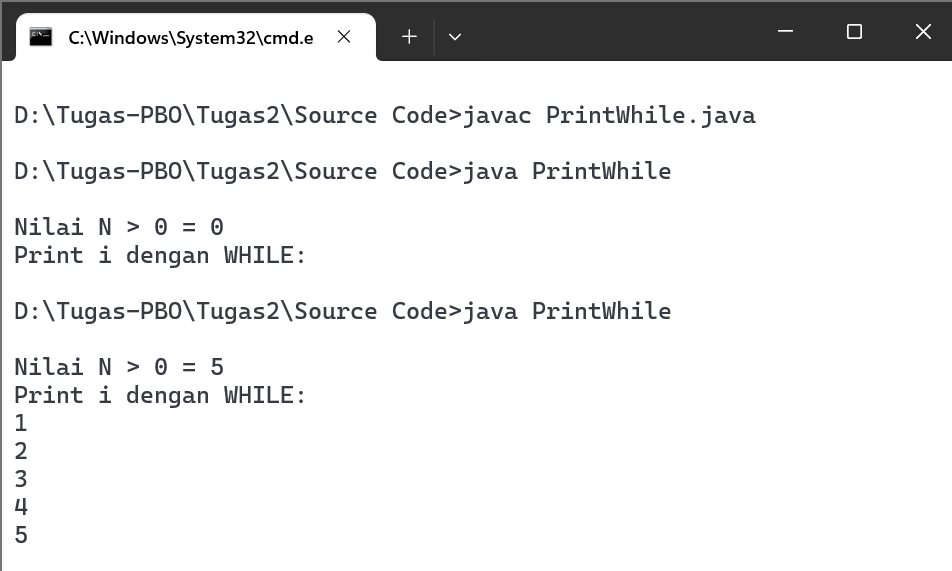
* System.out.println(i);: Mencetak nilai i (bilangan dalam deret) ke konsol pada setiap iterasi loop.
* if (i == N) break;: Memeriksa apakah i sama dengan N. Jika ya, maka loop akan berhenti menggunakan pernyataan break.
* else { i++; }: Jika i tidak sama dengan N, maka i akan diinkremen.

1. Output



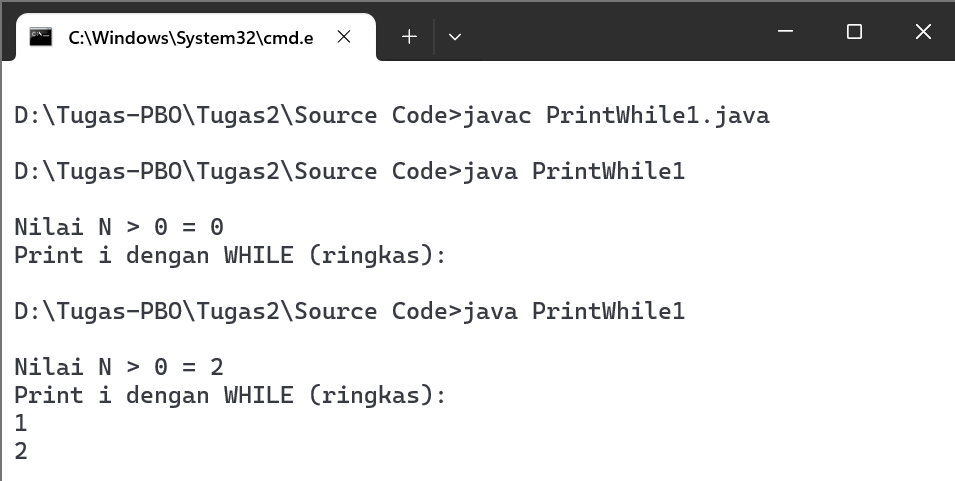
* Deklarasi dan Inisialisasi
* int N; : Mendeklarasikan variabel N bertipe data int (integer) yang akan digunakan untuk menyimpan batas atas deret yang akan dicetak.
* int i; : Mendeklarasikan variabel i bertipe data int (integer) yang akan digunakan sebagai counter untuk perulangan.
* Scanner masukan = new Scanner(System.in); : Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* i = 1; : Menginisialisasi variabel counter i dengan nilai 1.
* Input dan Output
* System.out.print("\nNilai N > 0 = "); : Menampilkan pesan ke konsol yang meminta pengguna untuk memasukkan nilai N yang lebih besar dari 0.
* N = masukan.nextInt(); : Membaca input integer dari pengguna menggunakan objek masukan dan menyimpan nilai yang dimasukkan ke dalam variabel N.
* System.out.print("Print i dengan REPEAT: \n"); : Menampilkan pesan bahwa program akan mencetak nilai i dengan pengulangan (REPEAT).
* do { System.out.print(i+"\n"); i++; } while (i <= N); : Mencetak deret bilangan dari 1 hingga N ke konsol menggunakan loop do-while.
* System.out.print(i+"\n");: Mencetak nilai i (bilangan dalam deret) ke konsol diikuti dengan baris baru pada setiap iterasi loop.
* i++;: Menginkremen nilai i setelah mencetak nilai tersebut.
* while (i <= N);: Memeriksa apakah i kurang dari atau sama dengan N. Jika ya, maka loop akan berlanjut. Jika tidak, loop akan berhenti.

1. Output



* Deklarasi dan Inisialisasi
* int N; : Mendeklarasikan variabel N bertipe data int (integer) yang akan digunakan untuk menyimpan batas atas deret yang akan dicetak.
* int i; : Mendeklarasikan variabel i bertipe data int (integer) yang akan digunakan sebagai counter untuk perulangan.
* Scanner masukan = new Scanner(System.in); : Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* i = 1; : Menginisialisasi variabel counter i dengan nilai 1.
* Input dan Output
* System.out.print("\nNilai N > 0 = "); : Menampilkan pesan ke konsol yang meminta pengguna untuk memasukkan nilai N yang lebih besar dari 0.
* N = masukan.nextInt(); : Membaca input integer dari pengguna menggunakan objek masukan dan menyimpan nilai yang dimasukkan ke dalam variabel N.
* System.out.print("Print i dengan WHILE: \n"); : Menampilkan pesan bahwa program akan mencetak nilai i dengan pengulangan (WHILE).
* while (i <= N) { System.out.println(i); i++; }; : Mencetak deret bilangan dari 1 hingga N ke konsol menggunakan loop while.
* while (i <= N): Memeriksa apakah i kurang dari atau sama dengan N. Jika ya, maka blok kode di dalam loop akan dieksekusi. Jika tidak, loop akan berhenti.
* System.out.println(i);: Mencetak nilai i (bilangan dalam deret) ke konsol pada setiap iterasi loop.
* i++;: Menginkremen nilai i setelah mencetak nilai tersebut.

1. Output

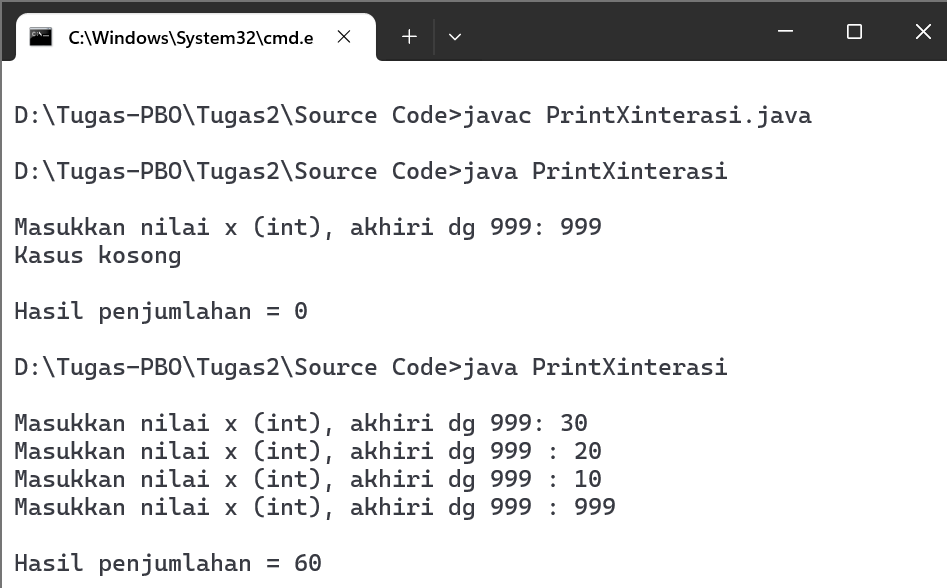


* Deklarasi dan Inisialisasi
* int N; : Mendeklarasikan variabel N bertipe data int (integer) yang akan digunakan untuk menyimpan batas atas deret yang akan dicetak.
* int i; : Mendeklarasikan variabel i bertipe data int (integer) yang akan digunakan sebagai counter untuk perulangan.
* Scanner masukan = new Scanner(System.in); : Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.
* i = 1; : Menginisialisasi variabel counter i dengan nilai 1.
* Input dan Output
* System.out.print("\nNilai N > 0 = "); : Menampilkan pesan ke konsol yang meminta pengguna untuk memasukkan nilai N yang lebih besar dari 0.
* N = masukan.nextInt(); : Membaca input integer dari pengguna menggunakan objek masukan dan menyimpan nilai yang dimasukkan ke dalam variabel N.
* System.out.print("Print i dengan WHILE (ringkas): \n");

while (i <= N) { System.out.println(i++);} : Mencetak deret bilangan dari 1 hingga N ke konsol menggunakan loop while dengan increment variabel i yang diringkas.

* System.out.print("Print i dengan WHILE (ringkas): \n");: Menampilkan pesan sebelum memulai loop.
* while (i <= N): Memeriksa apakah i kurang dari atau sama dengan N. Jika ya, maka blok kode di dalam loop akan dieksekusi. Jika tidak, loop akan berhenti.
* System.out.println(i++);: Mencetak nilai i ke konsol dan kemudian menginkremen nilai i setelahnya (post-increment).

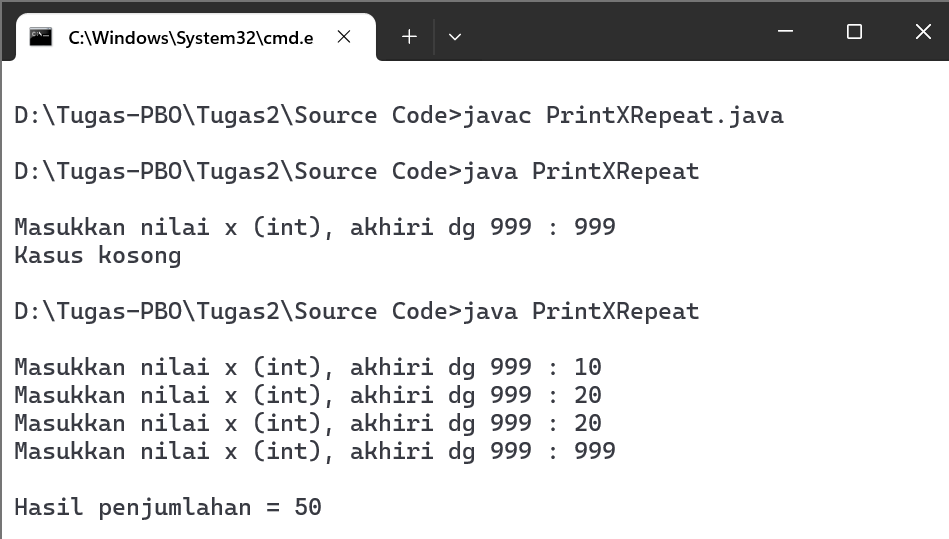
1. Output



1. Inisialisasi Program: Program dimulai dengan mendeklarasikan kelas dan metode utama, menginisialisasi variabel, dan membuat objek Scanner.

* int Sum = 0;: Mendeklarasikan variabel Sum bertipe data int (integer) yang akan digunakan untuk menyimpan hasil penjumlahan nilai-nilai x dan diinisialisasi dengan nilai 0.
* int x; : Mendeklarasikan variabel x bertipe data int (integer) yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
* Scanner masukan = new Scanner(System.in); : Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.

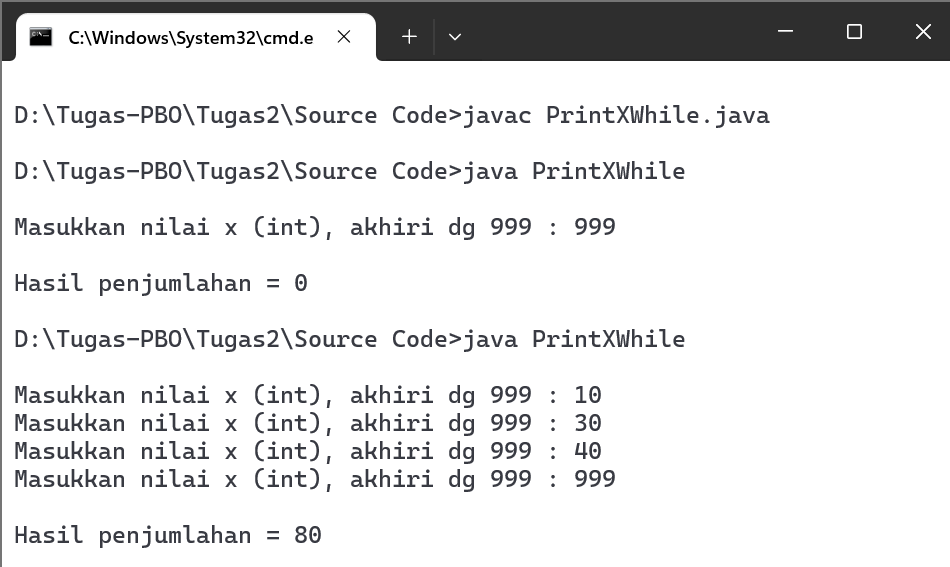
1. Membaca Input Pertama: Program meminta pengguna untuk memasukkan nilai x dan membaca nilai tersebut.
2. Pemeriksaan Kasus Kosong: Program memeriksa apakah nilai pertama yang dimasukkan adalah 999. Jika ya, program mencetak pesan bahwa kasusnya kosong dan berhenti.
3. Iterasi Penjumlahan: Jika nilai pertama bukan 999, program masuk ke dalam loop tak terbatas untuk terus membaca nilai x dari pengguna dan menambahkannya ke variabel Sum sampai pengguna memasukkan nilai 999.
4. Menampilkan Hasil Akhir: Program menampilkan hasil penjumlahan dari semua nilai x yang dimasukkan sebelum nilai 999.
5. Output



1. Inisialisasi Program: Program dimulai dengan mendeklarasikan kelas dan metode utama, menginisialisasi variabel, dan membuat objek Scanner.

* int Sum; Mendeklarasikan variabel Sum bertipe data int (integer) yang akan digunakan untuk menyimpan hasil penjumlahan nilai-nilai x.
* int x; : Mendeklarasikan variabel x bertipe data int (integer) yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
* Scanner masukan = new Scanner(System.in); : Membuat objek Scanner bernama masukan untuk membaca input dari keyboard.

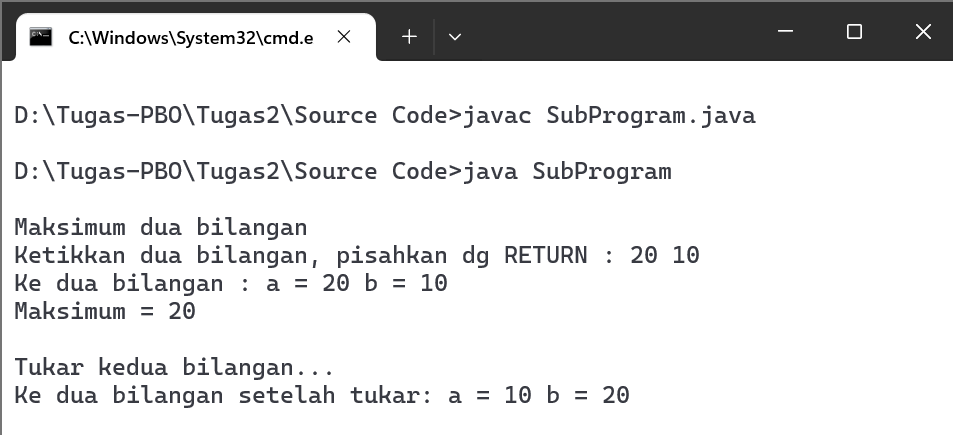
1. Membaca Input Pertama: Program meminta pengguna untuk memasukkan nilai x dan membaca nilai tersebut.
2. Pemeriksaan Kasus Kosong: Program memeriksa apakah nilai pertama yang dimasukkan adalah 999. Jika ya, program mencetak pesan bahwa kasusnya kosong dan berhenti.
3. Iterasi Penjumlahan dengan do-while Loop:
   * Jika nilai pertama bukan 999, program masuk ke dalam loop do-while untuk terus membaca nilai x dari pengguna dan menambahkannya ke variabel Sum.
   * Loop akan selalu dieksekusi setidaknya sekali, kemudian kondisi x != 999 diperiksa.
4. Menampilkan Hasil Akhir: Program menampilkan hasil penjumlahan dari semua nilai x yang dimasukkan sebelum nilai 999.
5. Output



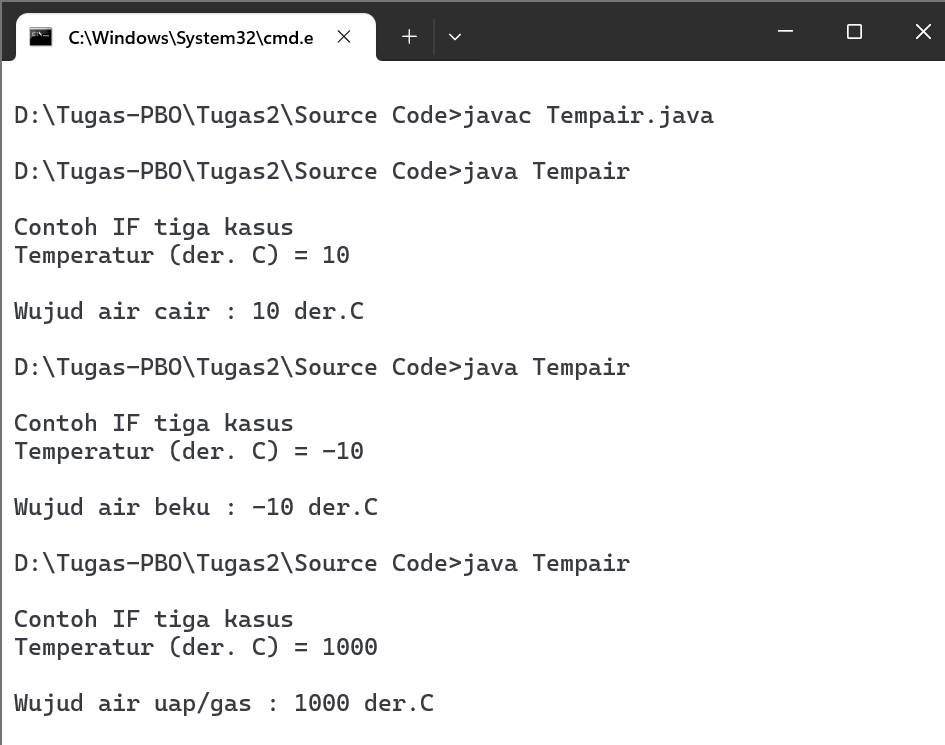
1. Inisialisasi Program: Program dimulai dengan mendeklarasikan kelas dan metode utama, menginisialisasi variabel, dan membuat objek Scanner.

* int Sum; Mendeklarasikan variabel Sum bertipe data int (integer) yang akan digunakan untuk menyimpan hasil penjumlahan nilai-nilai x.
* int x; : Mendeklarasikan variabel x bertipe data int (integer) yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
* Sum = 0; : Menginisialisasi variable Sum dengan nilai 0.
* Scanner masukan = new Scanner(System.in); : Membuat objek Scanner bernama masukan untuk membaca input dari keyboard

1. Membaca Input Pertama: Program meminta pengguna untuk memasukkan nilai integer x.
2. Iterasi Penjumlahan dengan WHILE Loop:
   * Program menggunakan loop while untuk terus meminta input dari pengguna dan menjumlahkan semua nilai yang dimasukkan hingga pengguna memasukkan 999.
   * Setiap kali nilai baru dimasukkan, program menambahkannya ke total (Sum) dan meminta input lagi.
3. Menampilkan Hasil Akhir: Setelah pengguna memasukkan 999, loop berhenti, dan program menampilkan total penjumlahan semua nilai yang dimasukkan sebelumnya.
4. Output



1. Inisialisasi Program: Program dimulai dengan mendeklarasikan kelas dan metode utama serta menginisialisasi variabel dan objek Scanner.
2. Membaca Input Pengguna: Program meminta pengguna untuk memasuk kan dua bilangan integer dan membaca nilai tersebut.
3. Menampilkan Input Bilangan: Program mencetak kedua bilangan yang telah dimasukkan ke konsol.
4. Menghitung Nilai Maksimum: Program memanggil fungsi maxab, yang menghitung nilai maksimum dari kedua bilangan dan mengembalikannya ke metode utama untuk ditampilkan.
5. Menukar Bilangan:
   * Program memanggil prosedur tukar, yang menukar nilai kedua bilangan dan mencetak hasilnya ke konsol.
   * Perlu dicatat bahwa meskipun prosedur mencetak hasil tukar, perubahan tidak berpengaruh pada variabel asli di metode utama karena Java menggunakan pass-by-value untuk tipe data primitif.
6. Output



1. Inisialisasi Program: Program dimulai dengan mendeklarasikan kelas dan metode utama serta menginisialisasi variabel dan objek Scanner.
2. Membaca Input Pengguna: Program meminta pengguna untuk memasukkan temperatur air.
3. Memeriksa Kondisi Temperatur: Program menggunakan pernyataan if-else if-else untuk memeriksa rentang temperatur dan menentukan wujud air yang sesuai.

* **Kasus 1:**T < 0 : Jika temperatur kurang dari 0, program menampilkan pesan bahwa wujud air adalah beku.
* **Kasus 2:**(0 <= T) && (T <= 100) : Jika temperatur antara 0 dan 100 (inklusif), program menampilkan pesan bahwa wujud air adalah cair.
* **Kasus 3:**T > 100 : Jika temperatur lebih dari 100, program menampilkan pesan bahwa wujud air adalah uap/gas.

1. Menampilkan Output Wujud Air: Program menampilkan pesan yang sesuai dengan wujud air berdasarkan temperatur yang dimasukkan oleh pengguna.