****

**Laporan Praktikum Bab 3**

**PENGULANGAN**

**Pemrograman Dasar TIF – F**



**Nama:**

**Muhammad Ghazy Humaidi (245150200111071)**

**Asisten:**

I Putu Natha Kusuma

Gabriel Christhoper

**Program Studi Teknik Informatika**

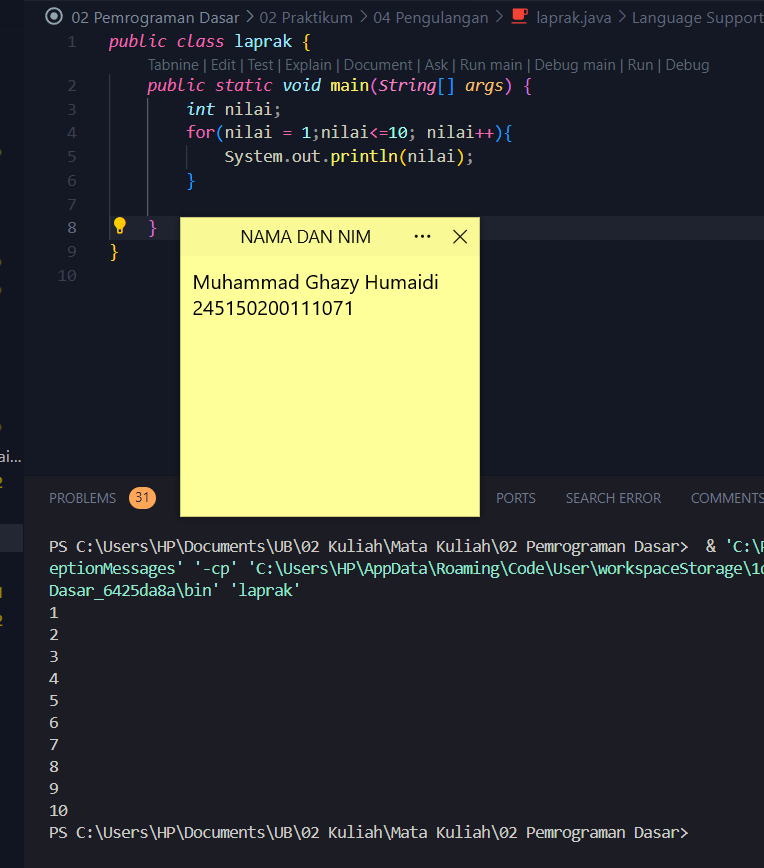
**Fakultas Ilmu Komputer**

**Universitas Brawijaya**

**2024**

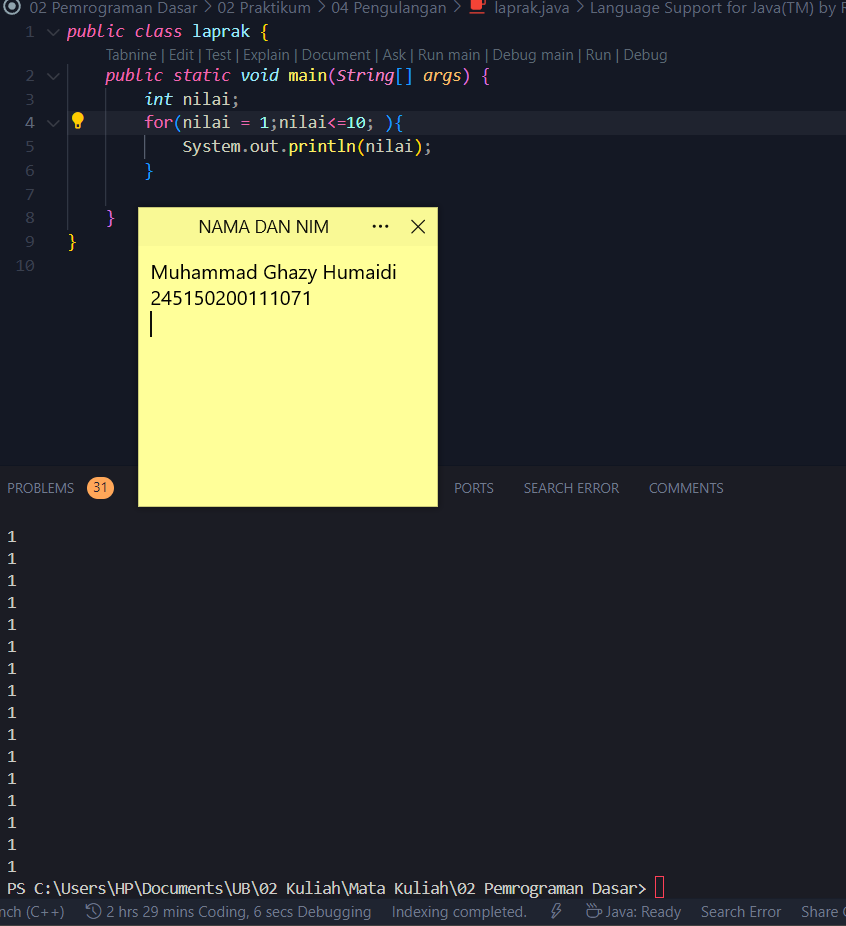
# Tugas Praktikum

1. Screenshoot Kode:



Sebenarnya tidak ada yang salah dalam kode tersebut, tetapi jika kita hanya menggunakan satu variable, maka proses deklarasi dan inisialisasi dapat dilakukan di dalam for looping agar mudah untuk dibaca dan tidak berceceran.

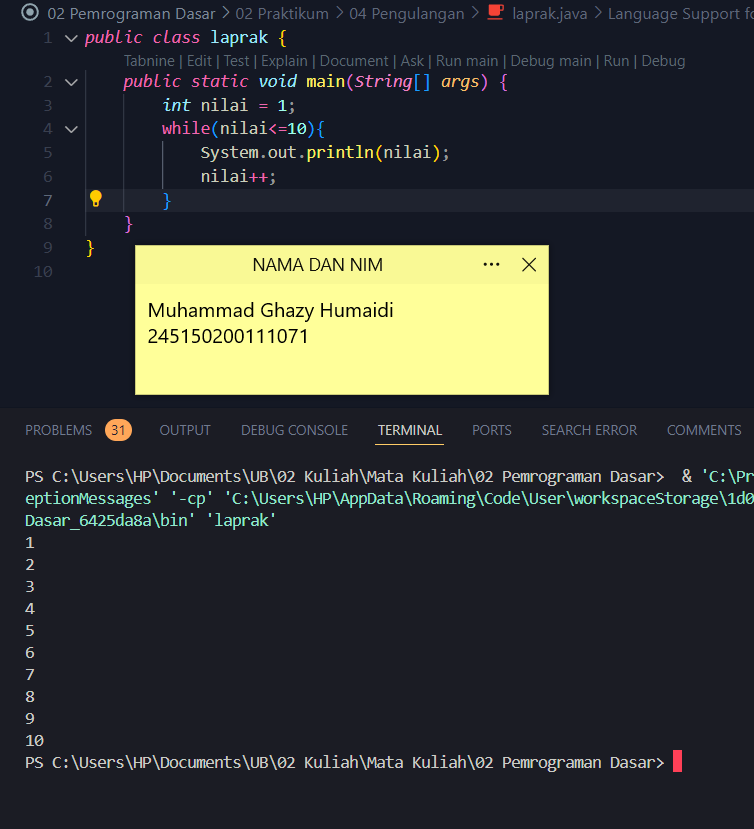
1. Variabel nilai berfungsi sebagai nilai awal di dalam sebuah loop yang nantinya akan dilakukan increment atau decrement tergantung dengan kebutuhan dari penggunaan loop.
2. Screenshoot kode:



Maka akan terjadi infinite loop, karena variable nilai akan tetap sama tanpa ada peningkatan dan ketika dilakukan pengkondisian, maka nilainnya akan selalu benar dan loop akan dijalankan terus menerus.

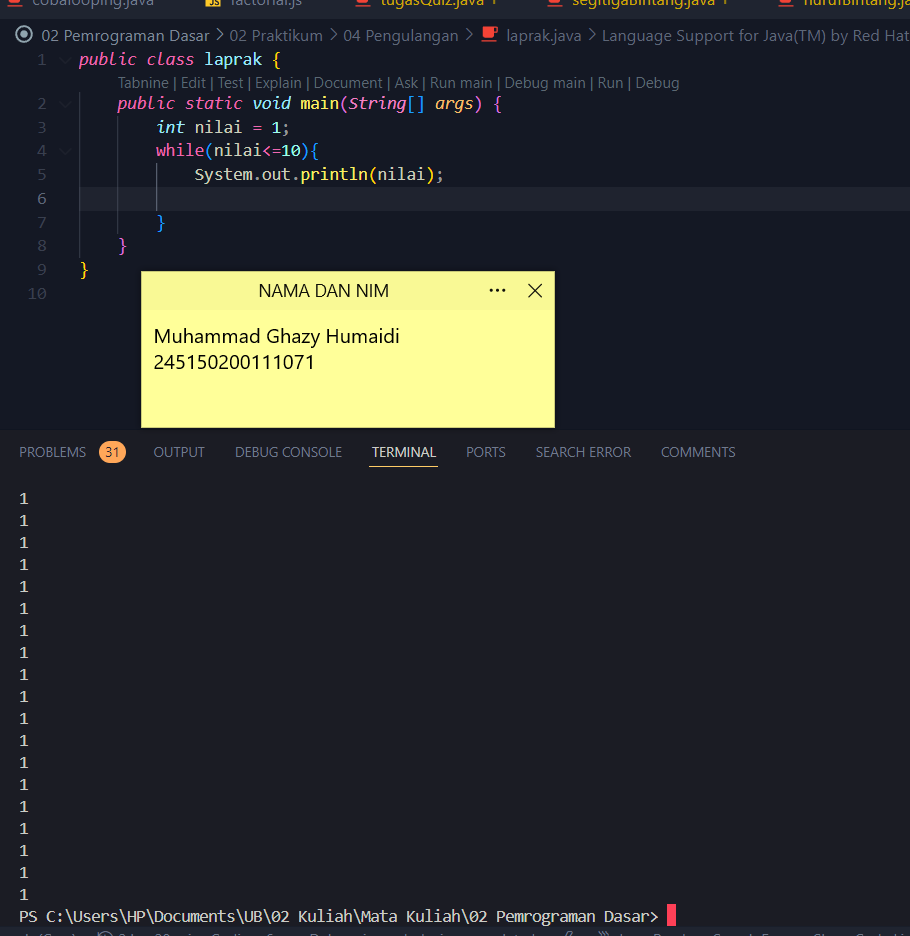
1. Jika initializationExpress dihapuskan, maka akan terjadi error karena tidak ada nilai awal dan tidak ada nilai yang hendak ditingkatkan. Jika loopCondition dihapuskan, maka akan terjadi infinite loop karena for loop tersebut tidak tau kapan variable nilai harus berhenti. Jika stepExpression dihapuskan, maka akan terjadi infinite loop karena variable nilai akan tetap sama dan ia akan selalu memenuhi pengkondisian di dalam loop selama-lamanya.

1. Screenshoot code:



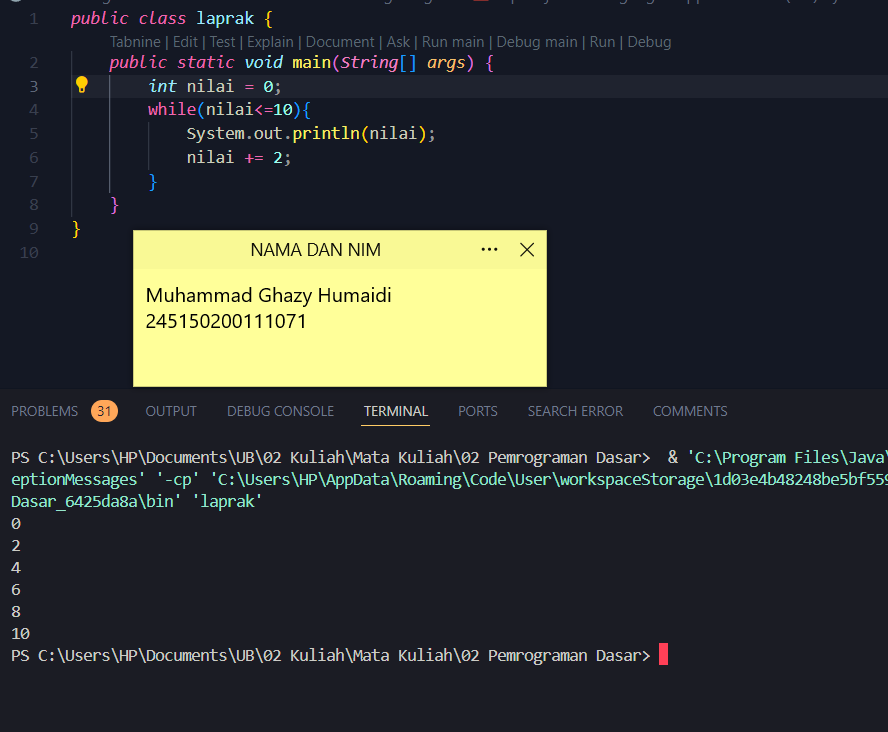
Kode tersebut tidak ada kesalahan dan variable while loop akan berjalan dimana variable nilai akan terus ditambah dan ditampilkan di console hingga variable nilai lebih kecil sama dengan 10 dan jika kondisi sudah tidak terpenuhi maka while loop akan selesai.

1. Perbedaan for dan while terdapat pada penempatan stepExpression yang fleksibel dan struktur yang sederhana. Jika kita melihat while, perbedaan terdapat pada peletakkan sintaks yang sederhana dan pemisahan struktur looping. Sedangkan untuk for sendiri, semua struktur looping disimpan di satu baris dan lebih terstruktur.
2. Screenshoot code:

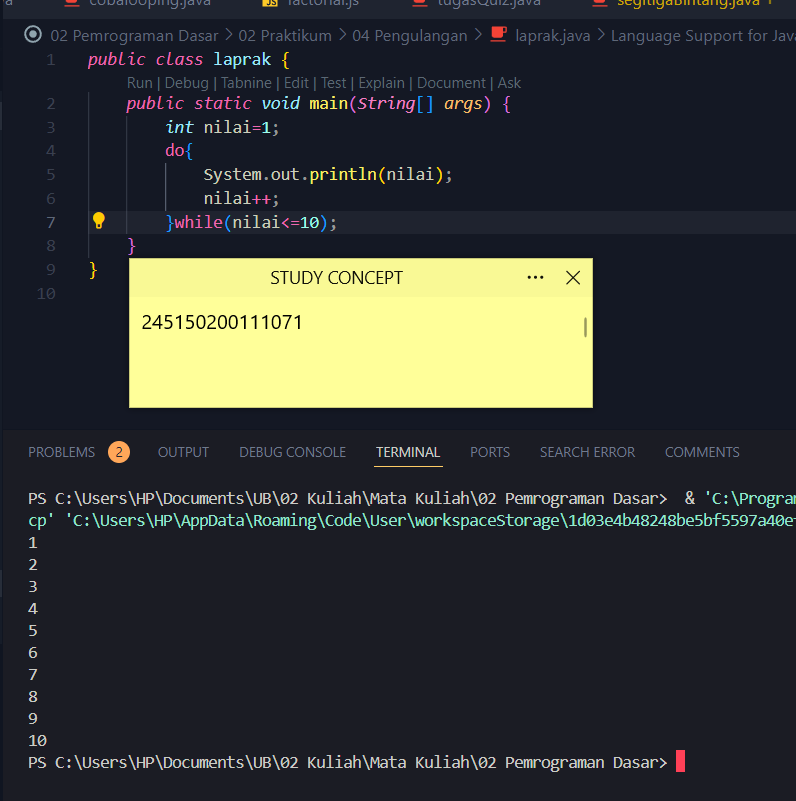


Maka akan terjadi infinite loop, karena nilai dari variable nilai tidak akan bertambah sedangkan ia akan selalu mengecek kondisi di dalam loop dan akan selalu bernilai true.

1. Screenshoot Code:

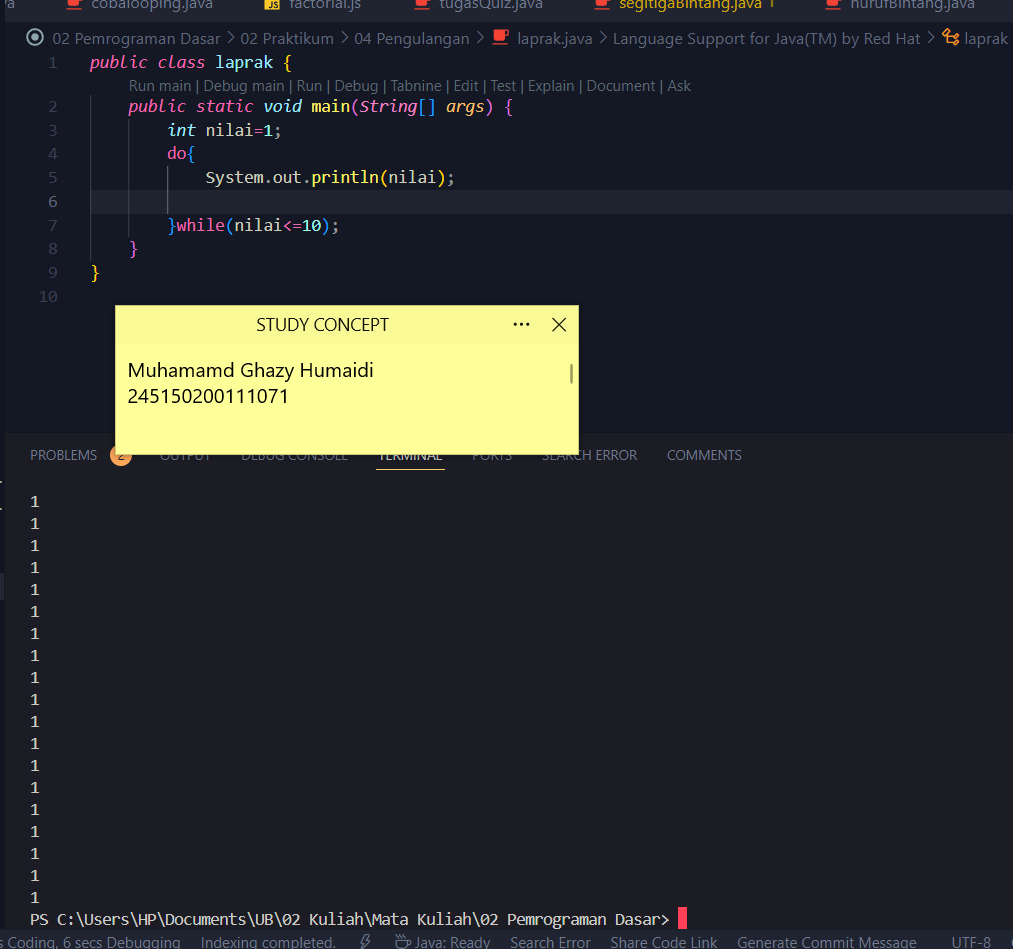


1. Screenshoot code:



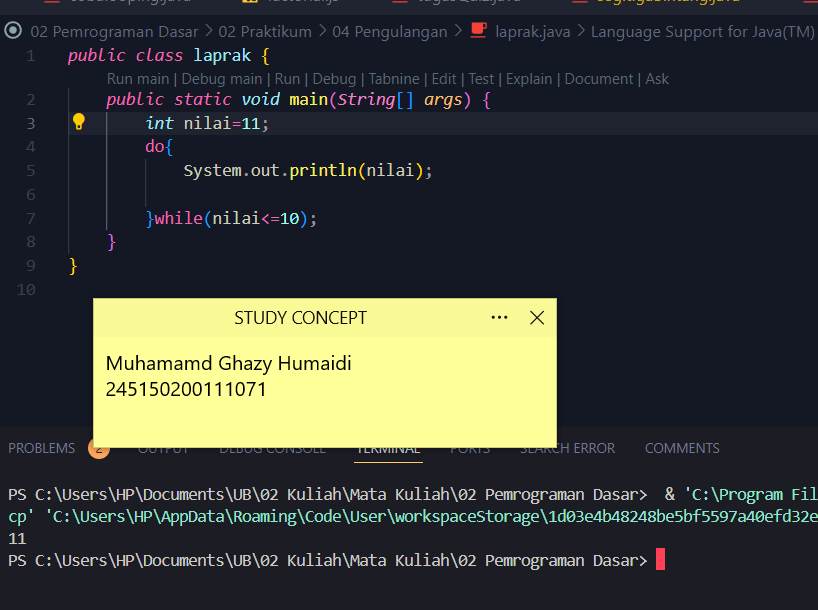
Jadi cara kerja dari kode diatas adalah ketika kita menjalankan kode tersebut, looping akan dijalankan setidaknya sekali kemudian melakukan pengecekan kondisi sesuai dengan apa yang ada di dalam keyword while, dan jika kondisi terpenuhi, maka looping akan dilakukan kembali hingga kondisi di dalam keyword while tidak terpenuhi dan bernilai false.

1. Perbedaan antara for, while dan do while adalah. dalam penggunaan, bagian increment dan decrement. Jika kita lihat dari sisi penggunaan, maka for loop biasa digunakan jika increment yang ditetapkan telah diketahui dengan jelas batasannya, sedangkan untuk while sendiri, jika increment yang ditetapkan tidak diketahui angka pastinya, dan untuk do while sendiri memiliki sifat yang sama seperti while, tetapi ia memiliki ciri khas yaitu seminimalnya kode akan dijalankan sekali. Dilihat dari bagian increment, for akan disimpan di awal setelah kondisi, sedangkan untuk while dan do while increment/decrement harus di definisikan di dalam scope block dari while/do while itu sendiri
2. Screenshoot Code:



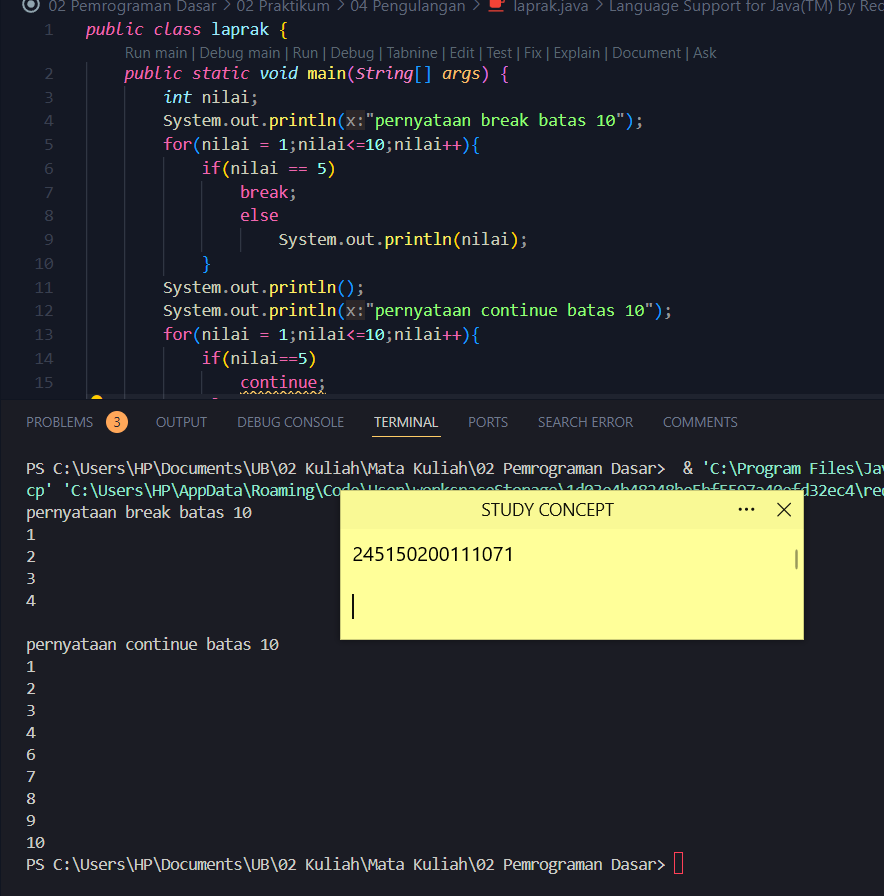
Maka code akan mengalami infinite loop karena pengkondisian akan selalu bernilai true dan hal tersebut disebabkan oleh variable nilai yang seharusnya bisa bertambah sehingga bisa keluar dari pengkondisian di dalam loop, tetapi hal tersebut tidak terjadi semenjak nilai++ dihilangkan dan variable nilai akan tetap sama hingga kapanpun.

1. Screenshot code:



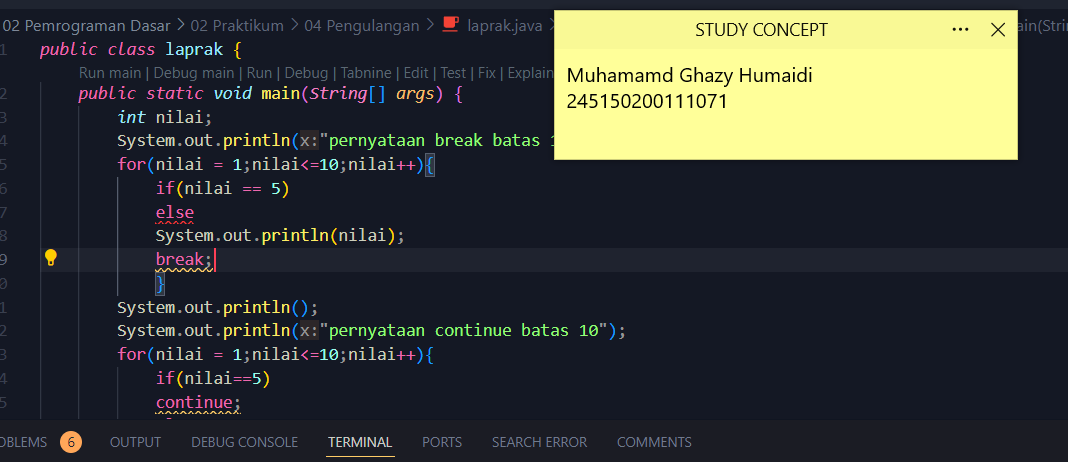
Maka yang akan terjadi adalah akan tampil angka 11, karena kita menggunakan do while yang dimana loop akan dilakukan setidaknya sekali ketika kode dijalankan. Dan setelah kode dijalankan maka akan melakukan pemeriksaan looping dengan variable nilai, dan itu akan bernilai false sehingag looping tidak adakn dilakukan.

1. Screenshoot Code:



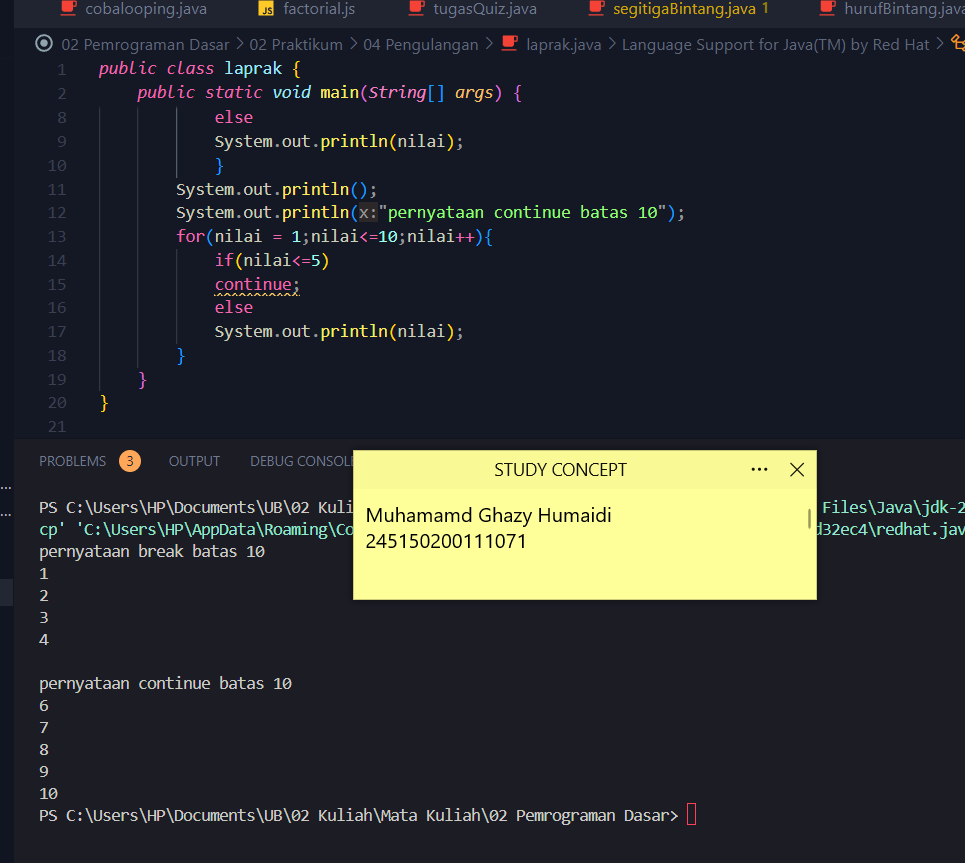
Jadi di kode tersebut terdapat dua loop, loop yang pertama akan terhenti di looping yang ke 5 karena ada keyword break. Keyword tersebut berguna untuk mengeluarkan kita dari loop yang sedang berlangsung. Sedangkan untuk for loop yang kedua, itu pada looping yang ke 5 akan melakukan skip pada pengulangan saat ini dan lanjut ke pengulangan selanjutnya. Maka dapat dilihat dengan jelas, jika angka 5 pada output for loop kedua itu tidak ada. Karena ketika di dalam loop dan bertemu dengan keyword continue, maka looping langsung lompat ke loopingselanjutnya.

1. Jadi untuk break sendiri jika dalam sebuah looping bertemu dengan keyword break ini, maka secara otomatis akan langsung keluar dari looping dan looping akan selesai. Sedangkan untuk keyword continue ini sendiri jika bertemu di dalam looping, maka looping yang sekarang akan dilewat dan akan di lanjut ke increment yang selanjutnya.
2. Screenshoot code:



Akan terjadi error karena else tiba tiba munucl tanpa adanya keyword if yang mendahuluinya. Karena pada awalnya kita menggunakan java shorthand agar penggunaan if dna else lebih sederhana, sehingga kurung kurawal tidak dibutuhkan, namun ketika dipidahkan ke baris lain, maka kurung kurawal dibutuhkan agar if mengetahui batasannya.

1. Hasil screenshoot code:



Maka variable nilai akan ditampilkan mulai dari angka 6, kenapa tidak angka 5? Karena ketika variable nilai bernilai 5, itu akan masuk ke dalam pengkondisian dan akan langsung dilakukan continue dan lanjut ke looping selanjutnya.

TUGAS PRAKTIKUM

|  |  |
| --- | --- |
| 1. | Hasil tangkapan layer:    Penjelasan :  Ketika kita menjalan kode java diatas, maka kita dapat memilih melakukan beberapa operasi berdasarkan angka yang ada disana, jika kita memilih angka 1, maka kita akan melakukan kalkulasi dari luas dan keliling dari persegi Panjang, jika kita memilih angka 2, maka kita akan melakukan kalkulasi dari luas dan keliling lingkara, dan jika kita memilih angka 3, maka kita akan menghitung luas dan keliling dari segitiga sembarang. Untuk keliling dari segitiga sembarang sendiri relative mudah karena kita hanya menghitung Panjang dari setiap sisi yang di-input. Sedangkan untuk luas dari segitiga sembarang itu lumayan kompleks dimana kita harus berurusan dengan rumus di bawah ini:  Luas segitiga sembarang = √(s(s-a)(s-b)(s-c) |
| 2. | |  |  | | --- | --- | |  | TugasPraktikum2.java | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | import *java.util.Scanner*;  *public* *class* tugasPraktikum2 {  *public* *static* *void* main(*String*[] *args*) {  *Scanner* input = new Scanner(System.in);          System.out.print("Berat badan (kg) : ");  *int* berat = input.nextInt();            System.out.print("Tinggi badan (m) : ");  *double* tinggi = input.nextDouble();  *double* imt = berat / (tinggi \* tinggi);            if(imt > 30){              System.out.printf("IMT = %.2f Termasuk Kegemukan", imt );          }else if(imt > 25){              System.out.printf("IMT = %.2f Termasuk Gemuk", imt );          }else if(imt > 18.5){              System.out.printf("IMT = %.2f Termasuk Normal", imt );          }else if(imt <= 18.5){              System.out.printf("IMT = %.2f Termasuk Kurus", imt );          }      }  } |   Hasil tangkapan layar:    Penjelasan:  User memasukakn input berat badan dan tinggi badan, nantinya program akan melakukan kalkulasi dengan rumus yang digunakan untuk menentukan IMT (Indeks Massa Tubuh), kemudian setelah hasil kalkulasi tersebut sudah ada, maka hasil kalkulasi tersebut akan dilakukan pengecekan sesuai dengan hasil kalkulasi IMT yang nantinya akan menentukan apakah IMT tersebut termasuk kegemukan, gemuk, normal, atau kurus. |
| 3. | |  |  | | --- | --- | |  | TugasPraktikum3.java | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | import *java.util.Scanner*;  *public* *class* tugasPraktikum3 {  *public* *static* *void* main(*String*[] *args*) {  *Scanner* input = new Scanner(System.in);            System.out.print("Jam Kerja : ");  *int* jamKerja = input.nextInt();  *int* lembur = 0;  *int* denda = 0;          if(jamKerja >= 60){              lembur = jamKerja % 60;              jamKerja = jamKerja - lembur;          }else if(jamKerja <= 50){              denda = 50 - jamKerja;          }  *int* upah = jamKerja \* 5000;          denda = denda \* 1000;          lembur = lembur \* 6000;  *int* total = upah + lembur - denda;            System.out.println("Upah = Rp." + upah);          System.out.println("Lembur = Rp." + lembur);          System.out.println("Denda = Rp." + denda);          System.out.println("---------------------");          System.out.println("Total = Rp." + total);      }  } |   Hasil tangkapan layar:    Penjelasan:  Untuk menghitung upah sendiri, kita bisa melakukan modulo terhadap jam kerja yang ada, maka jika jam kerja lebih dari 60 (batas maksimal jam kerja), maka akan secara otomatis akan mendapatkan jam lembur, karena sisa bagi dari batas maksimal jam kerja. Setelah kita mendapatkan nilai jam kerja dan lembut, kita tinggal mengalikan jam kerja dan jam lembur tersebut dengan harga per jam yang telah ditentukan. Sedangkan untuk menentukan denda sendiri, kita melakukan pengkondisian terlebih dahulu apakah nilai jam lebih kecil dari batas minimum jam kerja (yaitu 50 jam), ketika benar, maka kita lakukan pengurangan antara batas minimum jam kerja dan input jam kerja yang dimasukkan user kemudian mengkalikan dengan harga dari denda setiap jam nya. Setelah itu kita lakukan kalkulasi keseluruhan dengan menambahkan antara upah dengan lembur lalu mengurangi nya dengan denda. |