|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Sevia Ayu Wandinda**  **G1F021005** | **FOR dan WHILE Java** | **4 September 2024** |
| **1.a Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variabel   **Contoh 1: Salin dan tempel kode program berikut ke Eclipse.**  public class ContohFor{  public static void main(String[] args) {         for (double  y <= 15; y = 0;  y++) {                         if (y % 2 == 1) {         //kondisi 1                     // baris kode kosong 1                 } else if (y == 8) {       //kondisi 2                     // baris kode kosong 2                 } else                 System.out.println(y + " ");     }    }    }  **Luaran:** Exception in thread "main" java.lang.Error: Unresolved compilation problems:      Syntax error on token "<=", = expected     Type mismatch: cannot convert from double to boolean      at ContohFor.main(ContohFor.java:5)  **Contoh 2:** Salin dan tempel kode program berikut ke Eclipse.  public class ForBersarang {          public static void main(String[] args) {               pertama:                 for( int i = 1; i < 5; i++) {                     kedua:                     for(int j = 1; j < 3; j ++ ) {                         System.out.println("i = " + i + "; j = " +j);                     }             if ( i == 2) {                  // kode yang hilang                  }  }    }   }  **Contoh 3:** Salin dan tempel kode program berikut ke Eclipse.  import java.util.Scanner;  public class ForBersarang {         public static void main(String[] args){             //Instance Input Scanner             Scanner input = new Scanner(System.in);             System.out.print("Masukan Input: ");             int tinggi = input.nextInt(); //Mendapatkan Input Dari User             for(int t=tinggi; t>=1; t--){                 //Menghitung Jumlah Tinggi Piramida                 for(int s=tinggi; s>=t; s--){                     //Menghitung Jumlah Spasi per Baris                     System.out.print("\*");                 }                  System.out.println(); //Membuat Baris Baru             }         }     }  **Luaran:**  Masukan Input: 7 **\* \*\* \*\*\* \*\*\*\* \*\*\*\*\* \*\*\*\*\*\* \*\*\*\*\*\*\***  **Latihan 1** 1.1.     Evaluasi penyebab kesalahan dan perbaiki kode pada Contoh 1!            Rekomendasikan kata kunci yang tepat diletakkan pada baris kode yang kosong 1 dan 2 untuk dapat menghasilkan luaran berikut: Luaran contoh 1: 0  2  4  6  1.2.    Cermati contoh kode 2 pada kode  //baris kode kosong.            Rekomendasikan kode yang tepat menggunakan break atau continue terhadap pertama atau kedua agar menghasilkan luaran berikut: Luaran Contoh 2: i = 1; j = 1 i = 1; j = 2 i = 2; j = 1 i = 2; j = 2   1.3.    Cermati kode contoh 3. Apabila ingin menghasilkan luaran berikut: Luaran berbentuk piramida Masukan Input: 7        \*       \*\*\*      \*\*\*\*\*     \*\*\*\*\*\*\*    \*\*\*\*\*\*\*\*\*   \*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*         Rekomendasikan kode untuk menghasilkan luaran tersebut!  1.4.  Analisa diagram flowchart dari Latihan 1.2 dan 1.3! | | |
| **1.b Analisis dan Argumentasi** | | |
| 1. 1) Saya mengusulkan agar masalah ini dapat diselesaikan dengan   1.1 Dalam loop for, ekspresi yang digunakan adalah false. Pada kondisi if, syarat bilangan ganjil adalah y % 2 != 0 (bukan y % 2 == 1), karena hasil modulus suatu bilangan desimal akan selalu mengambang, dan bilangan ganjil akan menghasilkan modulus nol hasilnya harus 0.  1.2 Ada dua loop bersarang. Tujuan kita adalah berhenti mengulangi setelah i == 2. Untuk melakukan ini, kita dapat menggunakan break pada loop terluar yang diberi label pertama.  1.3 Gunakan dua loop bersarang sehingga kode program membuat piramida dengan jumlah baris berdasarkan masukan, dimulai dengan bintang di bagian atas hingga membentuk piramida.  2) Alasan solusi ini adalah  1.1 Menggunakan break atau continue stop atau melewatkan iterasi tertentu. Untuk menghasilkan hasil yang diinginkan, kita perlu menghentikan iterasi kedua setelah i == 2.  1.2 Biarkan kode program mencetak kombinasi i dan j untuk i == 1 dan i == 2, lalu hentikan perulangan ketika i = = 2.  1.3 Pada baris pertama piramida, harus ada lebih banyak ruang sebelum mencetak bintang, dan semakin sedikit ruang seiring bertambahnya tinggi piramida. Jarak secara bertahap berkurang tergantung pada ketinggian pengguna yang masuk. Bintang-bintang dicetak dalam jumlah yang semakin banyak dalam pola piramida (1, 3, 5, 7, dst). Hal ini dilakukan dengan menggunakan rumus (2 \* t - 1) untuk menghitung jumlah bintang per baris.  Perbaikan kode program dengan cara  a. 1.1  public class ContohFor {          public static void main(String[] args) {              for (double y = 0; y <= 15; y++) {                  if (y % 2 != 0) {                      continue;                  } else if (y == 8) {                      break;                  } else {                      System.out.println((int)y + " ");                  }              }          }      }  b. 1.2  public class Contoh2 {      public static void main(String[] args) {          pertama:          for (int i = 1; i < 5; i++) {              kedua:              for (int j = 1; j < 3; j++) {                  System.out.println("i = " + i + "; j = " + j);              }              if (i == 2) {                  break pertama;              }          }      }  }  c. 1.3  import java.util.Scanner;  public class Contoh3 {      public static void main(String[] args) {          Scanner input = new Scanner(System.in);          System.out.print("Masukan Input: ");          int tinggi = input.nextInt();          for (int t = 1; t <= tinggi; t++) {                for (int s = tinggi - t; s > 0; s--) {                  System.out.print(" ");              }                for (int b = 1; b <= (2 \* t - 1); b++) {                  System.out.print("\*");              }              System.out.println();          }      }  } | | |
| **1.c Penyusunan Algoritma dan Kode Program** | | |
| 1. Algoritma 2. Mulai 3. Buka aplikasi Vscode 4. Memperbaiki kode program 5. Menentukan luaran yang sudah ditentukan 6. Selesai 7. Kode program dan luaran   a. 1.1  b. 1.2    c. 1.3    Flowchart 1.2    1.3   1. Analisa luaran yang dihasilkan  Luaran sudah sesuai dengan kode program yang disusun.  Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data. | | |
| **1.d Kesimpulan** | | |
| **Analisa**  Dalam program ini, saya menggunakan bentuk kelas public karena semua kelas di Java harus dideklarasikan dalam ruang lingkup publik agar dapat diakses oleh seluruh program. Kelas public juga digunakan karena ini adalah entry point program (berisi metode main).  Pada Contoh 1  Perbaikan dilakukan dengan menambahkan inisialisasi, kondisi, dan pembaruan yang benar pada loop for, karena struktur Java mengharuskan setiap loop memiliki urutan yang jelas: inisialisasi, kondisi, dan pembaruan variabel.  Pada Contoh 2  Continue digunakan untuk menghentikan loop dalam pada kondisi tertentu agar iterasi berikutnya dapat dieksekusi pada loop luar. Ini diterapkan untuk memperbaiki perilaku loop bersarang.  Pada Contoh 3  Perbaikan dilakukan dengan menambahkan logika spasi dan bintang yang benar dalam loop bersarang untuk menghasilkan piramida bintang sesuai input pengguna. | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Sevia Ayu Wandinda**  **G1F021005** | **FOR dan WHILE Java** | **4 September 2024** |
| **2.a Identifikasi Masalah:** | | |
| Uraikan permasalahan dan variabel  **Contoh 4:** Salin dan tempel kode program berikut ke Eclipse.  public class ContohWhile{  public static void main(String[] args) {     int i=1;         while(i<=6){             System.out.println(i);             i++;                  if(i==4){                              break;        //ubah1              }}}  **Luaran:** 1 2 3  **Contoh 5:** Salin dan tempel kode program berikut ke Eclipse.  import java.util.Scanner;  public class ForBersarang {          public static void main(String[] args) {             Scanner dataKata = new Scanner(System.in);             System.out.print("Masukkan Kata yang ingin diulang : ");             String kata = dataKata.nextLine();                          Scanner dataJumlah = new Scanner(System.in);             System.out.print("Masukkan Jumlah ingin diulang : ");             int jumlah = dataJumlah.nextInt();                          int i = 0; //Inisialisasi batas dasar             while(i < jumlah){                 System.out.println(kata);                 i++;//Faktor pengulang Increment             }         } }   1. **Luaran Contoh 5:** Masukkan Kata yang ingin diulang : Fakultas Teknik Masukkan Jumlah ingin diulang : 5 Fakultas Teknik Fakultas Teknik Fakultas Teknik Fakultas Teknik Fakultas Teknik   **Latihan 2**  2.1.  Ubahlah baris kode pada Contoh 4         //Ubah1 menjadi  if(i % 3 == 0){   ◊ running, periksa hasilnya         //Ubah2 menjadi continue;  ◊ running, periksa hasilnya         Evaluasi perbandingan luaran sebelum dan setelah diubah! Simpulkan maksud dari perubahan tersebut!   1. 2.2. Cermati Contoh 5. Periksa luaran, bila ketika di eksekusi, jumlah yang diulang = 0!         Evaluasi luaran, bila kode diubah menjadi do … while dengan masukan sama jumlah yang diulang = 0.        Simpulkan perbedaan while dan do … while! 2. 2.3. Bila diketahui pernyataan pseudocode berikut:        [1]    inisiasi idPelajaran        [2]    inisiasi nilai pelajaran        [3]    inisiasi nilai rata-rata        [4]    Minta pengguna untuk menuliskan jumlah pelajaran        [5]    Ketika idPelajaran lebih kecil dari jumlah pelajaran        [6]    Minta pengguna untuk menuliskan nilai pelajaran        [7]    Hitung nilai rata-rata = (nilai pelajaran + nilai rata-rata) / 2             [8]    Tambah satu ke idPelajaran        [9]    Tampilkan nilai rata-rata        Rekomendasikan kode untuk menyelesaikan Pseudocode tersebut! 3. 2.4.     Rancang diagram flowchart dari Latihan 2.1, Latihan 2.2, dan Latihan 2.3! | | |
| **2.b Analisis dan Argumentasi** | | |
| 1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara   Membuat variabel untuk menyimpan ID pelajaran, nilai pelajaran, dan rata-rata. Dan meminta pengguna untuk memasukkan jumlah pelajaran. Menggunakan loop while untuk meminta input nilai pelajaran sesuai dengan jumlah yang dimasukkan. Dengan menghitung rata-rata secara bertahap dengan menambahkan nilai baru dan membagi dengan total pelajaran yang telah   1. Alasan solusi ini karena dengan inisialisasi variabel bisa menyimpan ID Pelajaran, nilai Pelajaran, dan rata-rata. Dan juga menggunkan Input jumlah Pelajaran agar pengguna untuk memasukkan jumlah Pelajaran. Output Rata-Rata menampilkan hasil perhitungan rata-rata kepada pengguna. 2. [2.1] Evaluasi perbandin gan luaran sebelum dan setelah diubah! Simpulkan maksud dari perubahan tersebut!   a. Evaluasi  Sebelum diubah (Ubah1) Program mencetak angka dari 1 hingga 3, kemudian berhenti ketika i mencapai 4. Setelah diubah (Ubah1) Program akan berhenti pada nilai i yang merupakan kelipatan dari 3, yaitu pada saat i mencapai 3. Setelah diubah (Ubah2) Program akan mencetak angka 1 dan 2, tetapi akan melewatkan angka 3 karena ada pernyataan continue, sehingga tidak akan mencetak angka tersebut dan melanjutkan ke iterasi berikutnya.  b. Kesimpulan  Menggunakan break menghentikan loop ketika kondisi tertentu terpenuhi. Untuk menggunakan continue melewatkan iterasi saat kondisi terpenuhi dan melanjutkan ke iterasi berikutnya.  a. Evaluasi [2.2]  [while] Jika jumlahnya yang diulang = 0, maka luarannya tidak ada output karena kondisi awal tidak terpenuhi. Sedangkan [do while] Tidak ada output juga, tetapi struktur do while memastikan bahwa blok kode dijalankan setidaknya sekali sebelum memeriksa kondisi.  b. Kesimpulan   * While loop mengecek kondisi sebelum menjalankan blok kode. Jika kondisi tidak terpenuhi (jumlah = 0), maka tidak ada output. * Do while loop menjalankan blok kode setidaknya sekali sebelum memeriksa kondisi. Namun, dalam kasus ini, jika jumlah = 0, tidak akan ada output juga. | | |
| **2.c Penyusunan Algoritma dan Kode Program** | | |
| 1. Algoritma   Algoritma menghitung rata-rata nilai pelajaran   1. Mulai 2. Buka Aplikasi Vscode 3. Inisialisasi variabel 4. Input jumlah pelajaran 5. Looping untuk input nilai 6. Output nilai rata-rata 7. Selesai 8. Kode program dan luaran      * 1. Analisa luaran yang dihasilkan   Luaran sudah sesuai dengan kode program yang disusun.  Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.  Flowchart 2.1    Flowchart 2.2    Flowchart 2.3 | | |
| **2.d Kesimpulan** | | |
| **Analisa**  kesimpulan dari analisa ini adalah bahwa penggunaan loop while dan kondisi break atau continue sangat penting dalam mengimplementasikan algoritma yang kompleks seperti menghitung rata-rata nilai pelajaran atau mengulang input kata sesuai dengan jumlah yang dimasukkan. | | |
|  | | |

**Refleksi**

Dilakukan secara individu namun saya sering ngobrol dengan teman sekelas untuk bertukar pikiran dan memecahkan masalah bersama. Diskusi ini berguna untuk memahami berbagai pendekatan untuk memecahkan masalah dan meningkatkan desain aplikasi. Berkolaborasi dalam diskusi ini juga membantu saya melihat bagaimana orang lain menghadapi tantangan serupa.