|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Muhammad Ariq Hafidl Fatanah G1F024014** | **FOR dan WHILE Java** | **01 Oktober 2024** |

**[No. 1] Identifikasi Masalah:**

1. Uraikan permasalahan dan variabel

Tuliskan kembali soal:

**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!

Pada contoh soal yang terdapat kesalahan hanya contoh soal 1, untuk soal 2 dan 3 tidak ada kesalahan

**[No.1] Analisis dan Argumentasi**

1.1 Evaluasi penyebab kesalahan

Pada kode for (double  y <= 15; y = 0;  y++)  Terdapat kesalahan yaitu pada bentuk umum for dimana inisialiasi y<= 15 dan kondisi y=0 posisinya terbalik. Kemudian mengganti tipe data double ke int agar kode lebih efesien. Sehingga perbaikan kode programnya menjadi { for (int y = 0; y <= 15; y++) {.

Dan untuk merekomendasikan 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

Kita dapat menambahkan kata kunci continue, Kata kunci continue digunakan untuk melanjutkan ke iterasi berikutnya dari perulangan. Dan untuk menghasilkan luaran seperti 0, 2, 4, 6 adalah dengan membatasi kondisi y <= 8 sehingga output yang dikeluarkan adalah 0, 2, 4, 6.

1.2 Rekomendasi kode 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 Masukan Input: 7  
       \*  
      \*\*\*  
     \*\*\*\*\*  
    \*\*\*\*\*\*\*  
   \*\*\*\*\*\*\*\*\*  
  \*\*\*\*\*\*\*\*\*\*\*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*  
Rekomendasikan kode untuk menghasilkan luaran tersebut!

Berikut adalah kode programnya:

import java.util.Scanner;

public class ForBersarang1 {

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

// Perulangan untuk baris

for (int t = 1; t <= tinggi; t++) {

// Menghitung Jumlah Spasi per Baris

for (int s = tinggi; s > t; s--) {

System.out.print(" "); // Mencetak spasi

}

// Menghitung Jumlah Bintang per Baris

for (int b = 1; b <= (2 \* t - 1); b++) {

System.out.print("\*"); // Mencetak bintang

}

System.out.println(); // Membuat Baris Baru

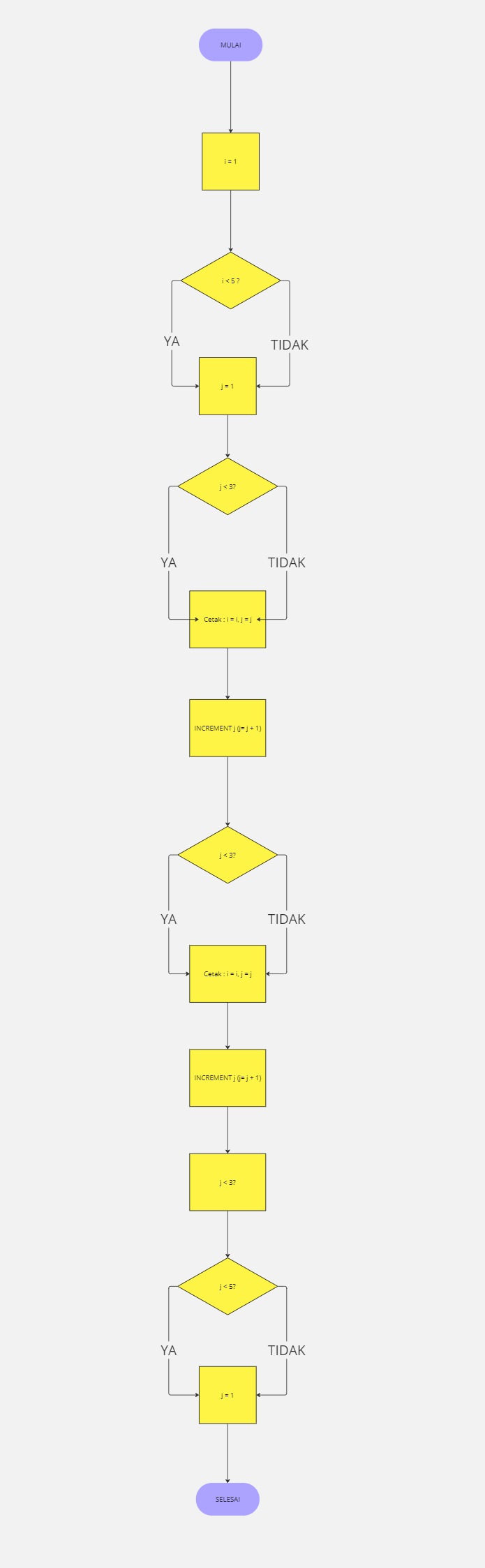
}

}

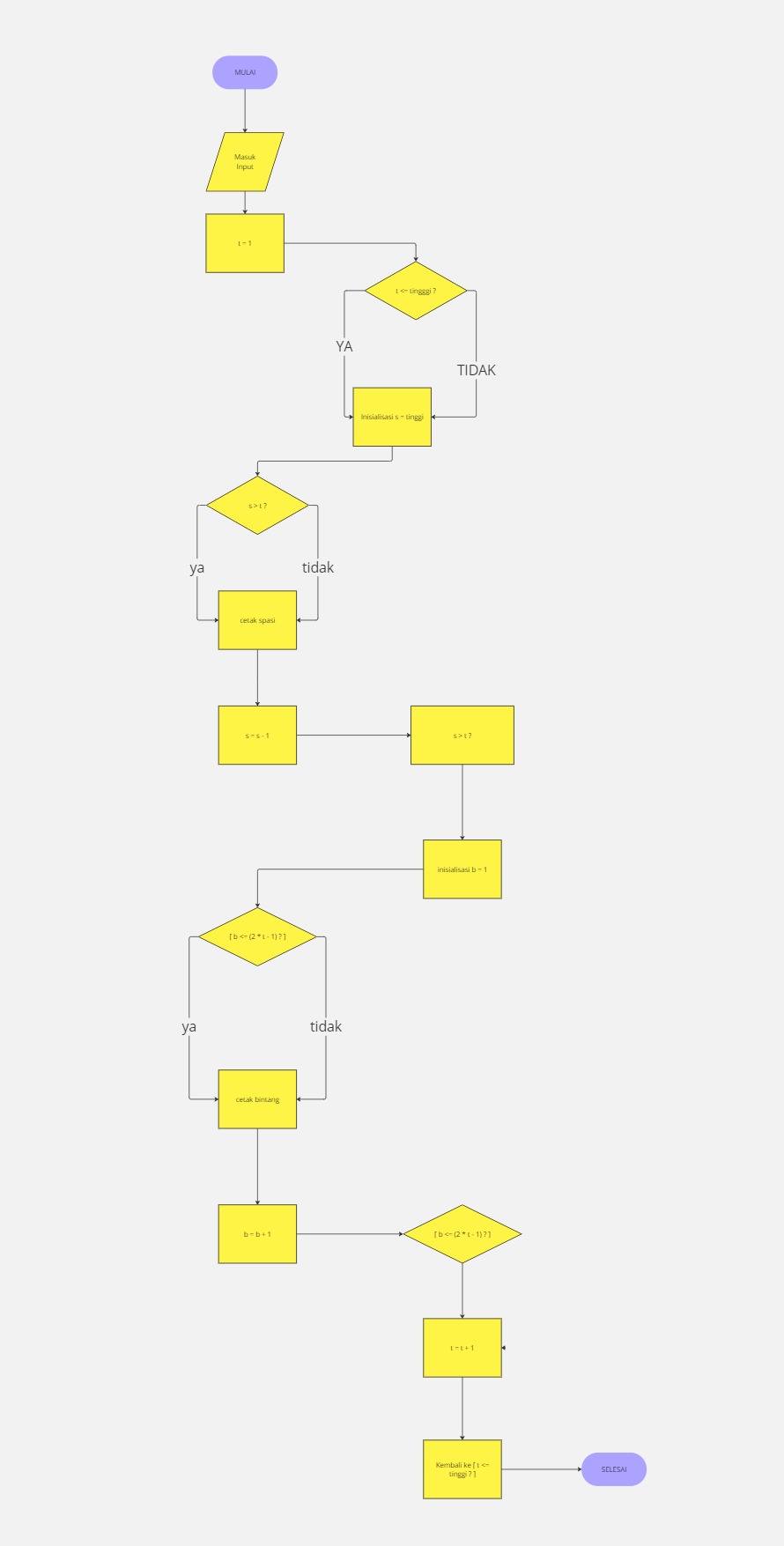
}

* 1. Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

Flowchart Latihan 1.2



Flowchart Latihan 1.3



**[No.1 ] Penyusunan Algoritma dan Kode Program**

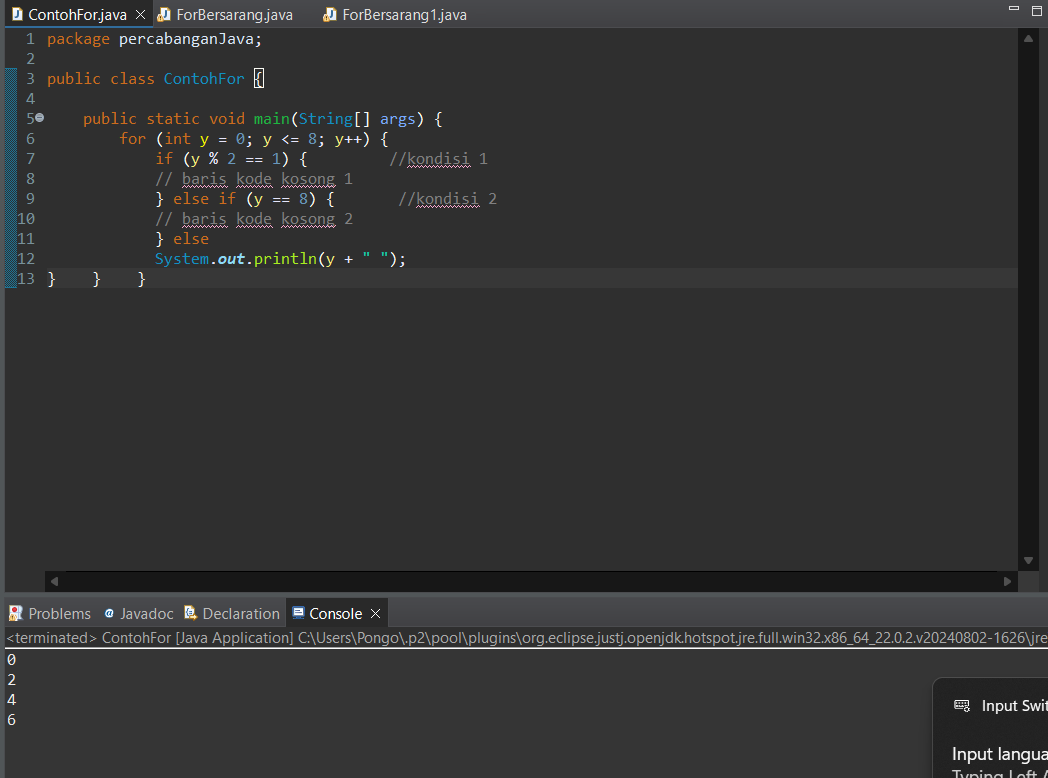
1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

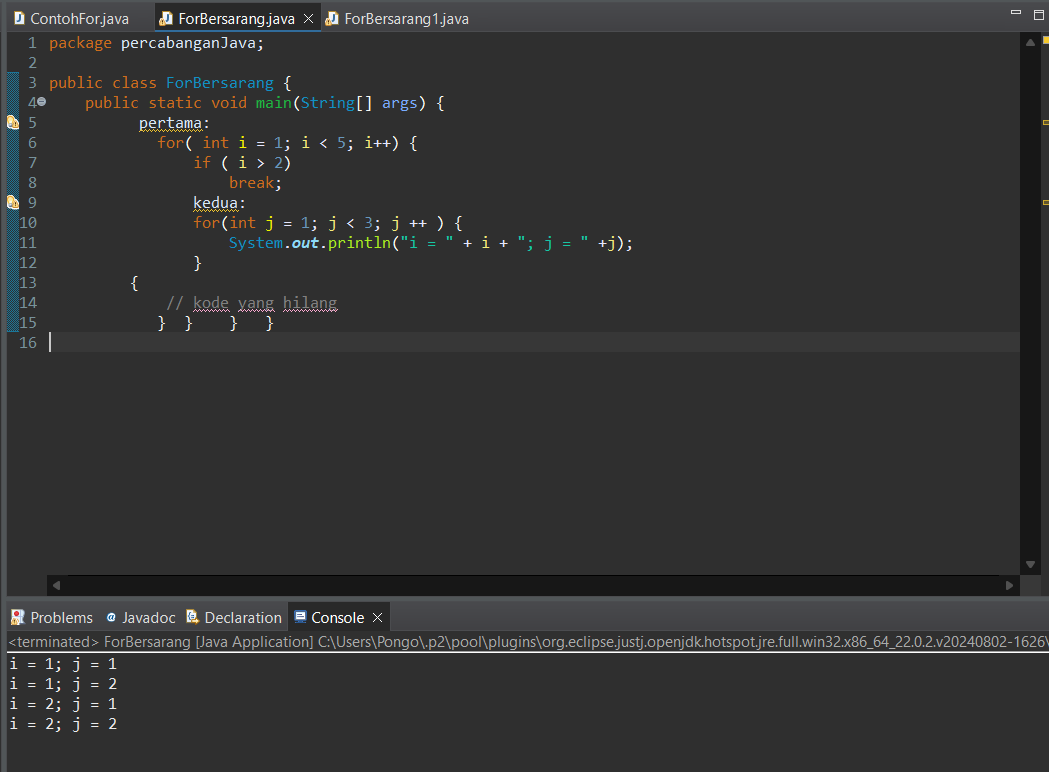
1. Mulai
2. Mengakses package
3. Mengimpor java jika diperlukan scanner
4. Mendeklarasikan public void
5. Menuliskan perulangan apa yang akan dibuat
6. Mengeprint output
7. selesai
8. Kode program dan luaran
9. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

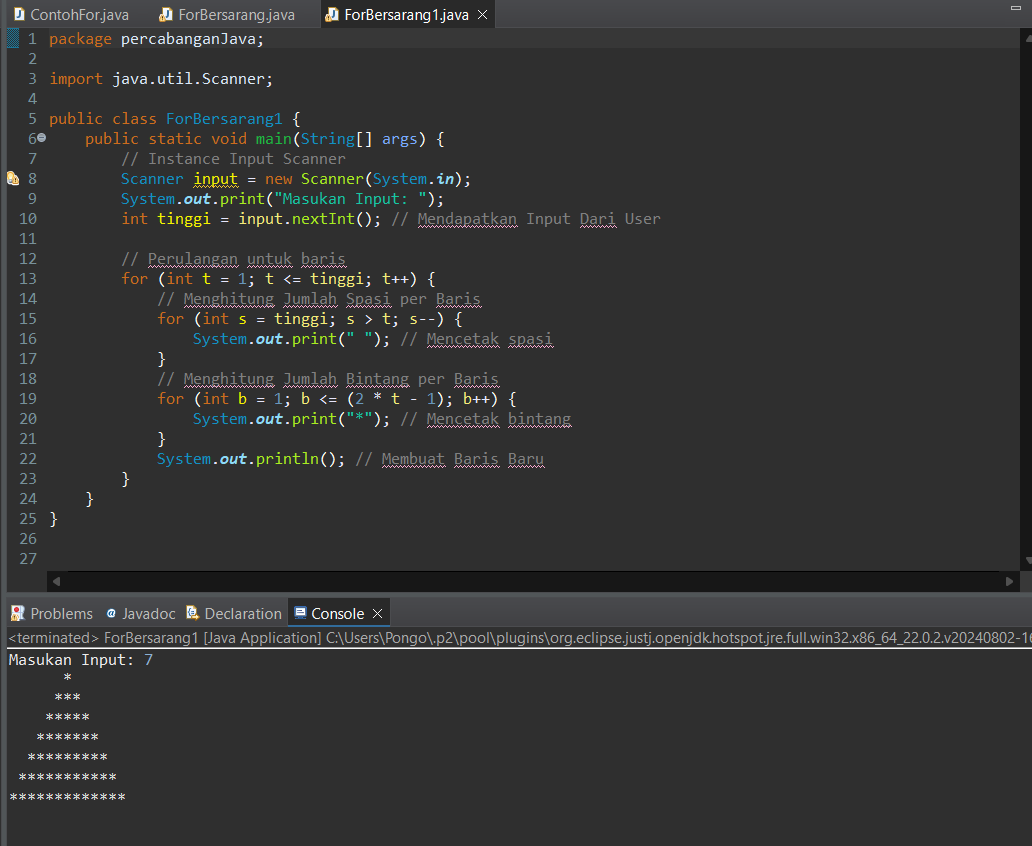
* 1. **Kode program**



* 1. **Kode Program**

****

* 1. **Kode Program**

****

1. Analisa luaran yang dihasilkan
   1. Luaran sudah sesuai dengan program yang disusun, program menghasilkan ouput 0, 2, 4, 6. Karena kode melakukan perulangan bilangan genap dengan membatasi y=0, y <= 8.
   2. Luaran sudah sesuai dengan program yang disusun. Program ini mencetak kombinasi nilai i dan j untuk nilai i yang kurang dari atau sama dengan 2. Setelah i mencapai 3, perulangan berhenti karena perintah break.
   3. Program ini berhasil mencetak piramida bintang dengan tinggi yang ditentukan oleh pengguna.

Dengan menghitung jumlah spasi dan bintang secara tepat, program menghasilkan bentuk piramida yang simetris. Setiap baris memiliki bintang yang selalu berjumlah ganjil, dan jumlah spasi yang diperlukan untuk mempertahankan bentuk piramida.

**[No.1] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawaban Analisa:

Kesimpulan dari program di atas penggunaan perulangan for dalam konteks pemrograman memberikan efisiensi dalam mengontrol jumlah iterasi. Melalui pemahaman yang mendalam tentang struktur ini, kita dapat menulis kode yang tidak hanya berfungsi dengan baik, tetapi juga mudah dipahami.

**[No. 2] Identifikasi Masalah:**

1. Uraikan permasalahan dan variabel

Tuliskan kembali soal:

**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  
            }  
        }  
}

**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!

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!

2.3. 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!

2.4.     Rancang diagram flowchart dari Latihan 2.1, Latihan 2.2, dan Latihan 2.3!

**[No.2] Analisis dan Argumentasi**

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!

**Jawab:** Sebelum perubahan, program menghentikan loop ketika i mencapai 4, sehingga hanya mencetak angka 1, 2, dan 3. Namun setelah adanya perubahan if (i % 3 == 0) { , ketika i mencapai 3 (kelipatan 3), program akan melanjutkan ke iterasi berikutnya tanpa mencetak 3, sehingga menghasilkan output 1, 2, 5, dan 6. Perubahan dari break ke continue mengubah perilaku loop. break menghentikan loop, sedangkan continue melanjutkan ke iterasi berikutnya,

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!

**Jawab:** Ketika jumlah yang diulang = 0, luaran program dengan while adalah tidak ada output karena kondisi i < jumlah tidak terpenuhi. Setelah saya mencoba mencobanya saya dapat menyimpulkan bahwa Ketika menggunakan do while, program akan tetap menjalankan satu iterasi sebelum memeriksa kondisi, sehingga menghasilkan satu kali output dari System.out.println(kata) meskipun jumlah = 0. Berbeda dengan while memeriksa kondisi sebelum di eksekusi sehingga tidak menampilkan output apapun Ketika jumlah = 0.

2.3  Rekomendasikan kode untuk menyelesaikan Pseudocode tersebut!

package percabanganJava;

import java.util.Scanner;

public class HitungNilaiRataRata {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int idPelajaran = 0;

double nilaiPelajaran = 0;

double nilaiRataRata = 0;

System.out.print("Masukkan jumlah pelajaran: ");

int jumlahPelajaran = scanner.nextInt();

while (idPelajaran < jumlahPelajaran) {

System.out.print("Masukkan nilai pelajaran: ");

nilaiPelajaran = scanner.nextDouble();

nilaiRataRata = (nilaiPelajaran + nilaiRataRata) / 2;

idPelajaran++;

}

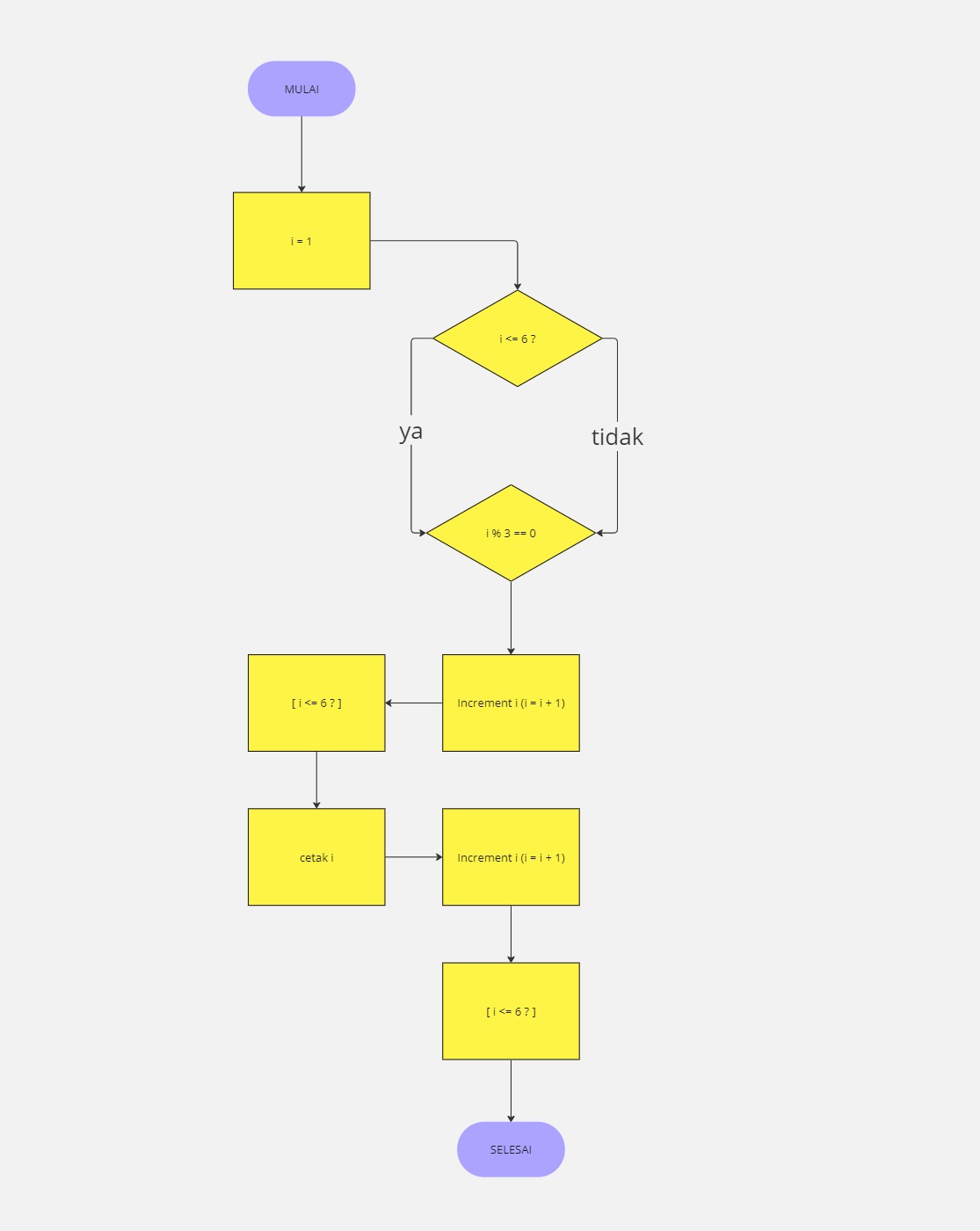
System.out.println("Nilai rata-rata: " + nilaiRataRata);

}

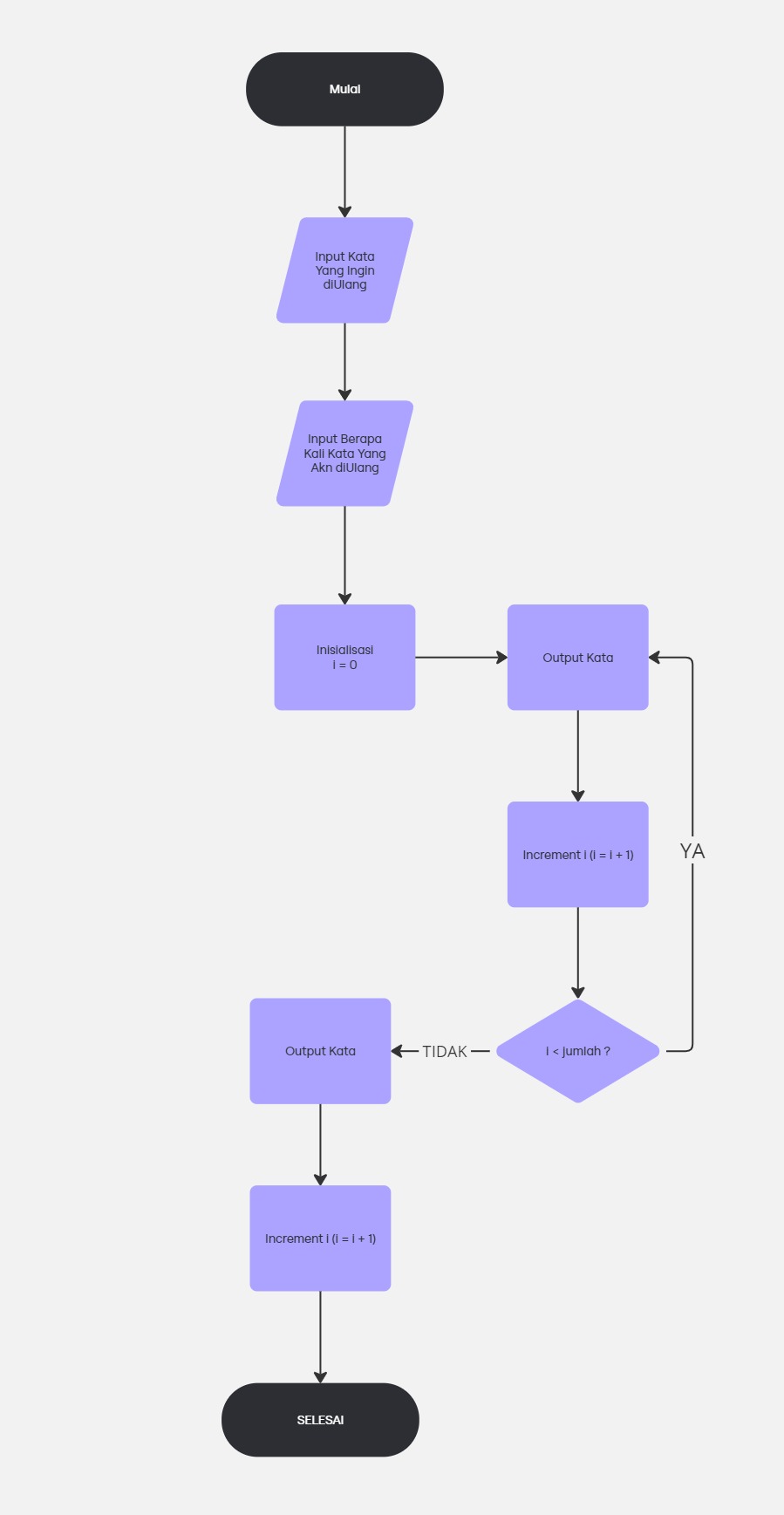
}

2.4 Rancang diagram flowchart dari Latihan 2.1, Latihan 2.2, dan Latihan 2.3!

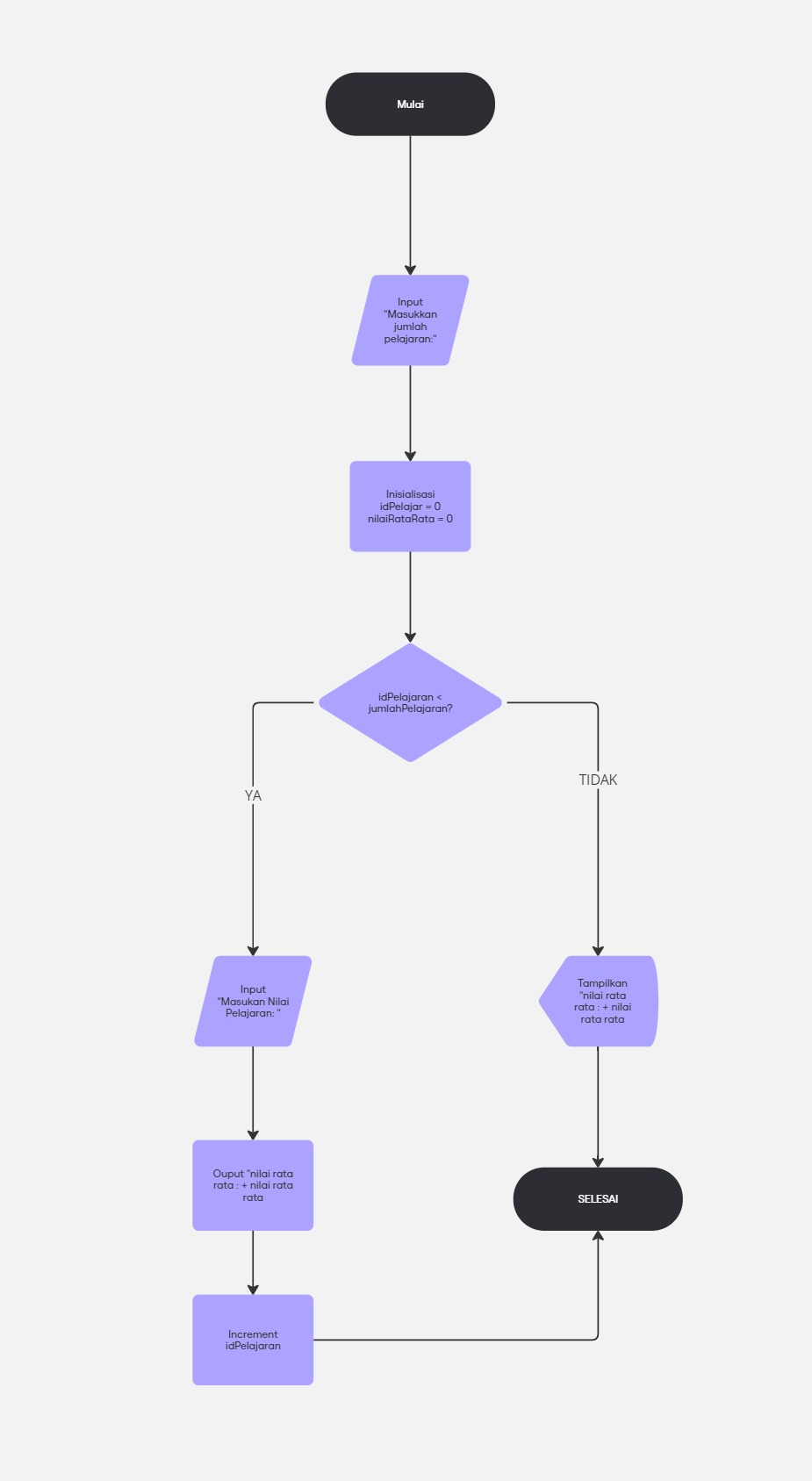
Flowchart 2.1



Flowchart 2.2



Flowchart 2.3



**[No.2 ] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Algoritma Contoh 4

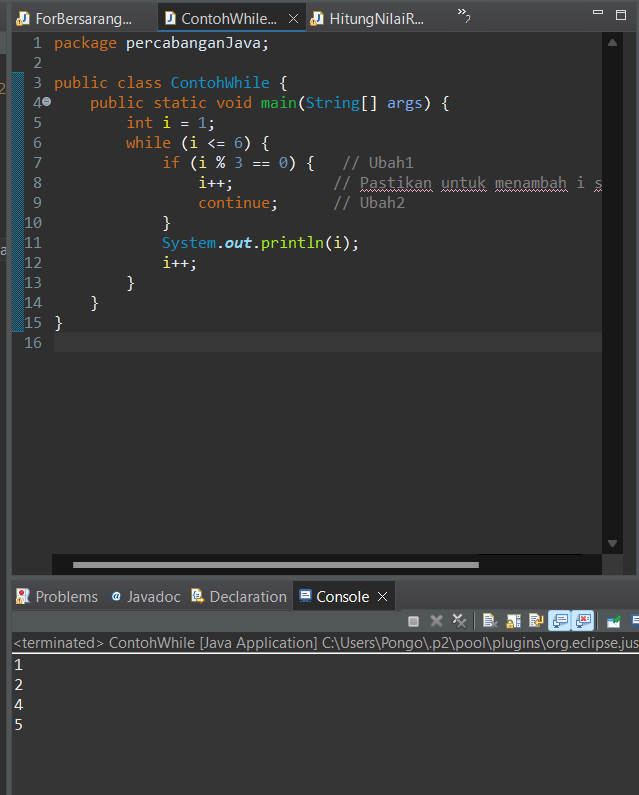
1. Inisialisasi I = 1
2. Selama I <= 6:
3. Jika i%3 == 0, tambahkan i lanjutkan
4. tampilkan i
5. tambahkan i
6. selesai

Algoritma Contoh 5

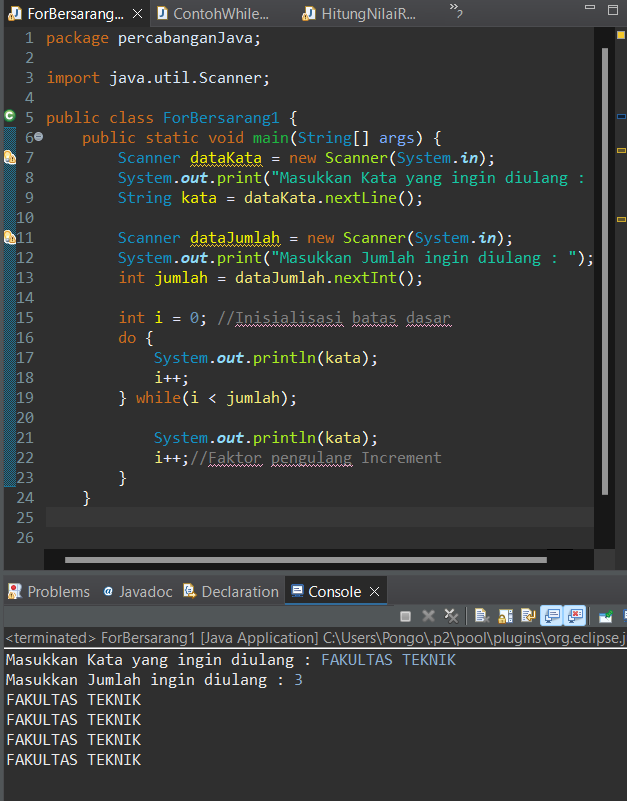
1. Inisialisasi scanner.
2. Minta input kata dan jumlah.
3. Set i = 0.
4. Tampilkan kata.
5. Tambahkan i.
6. Ulangi sampai i < jumlah.
7. Selesai.
8. Kode program dan luaran
9. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

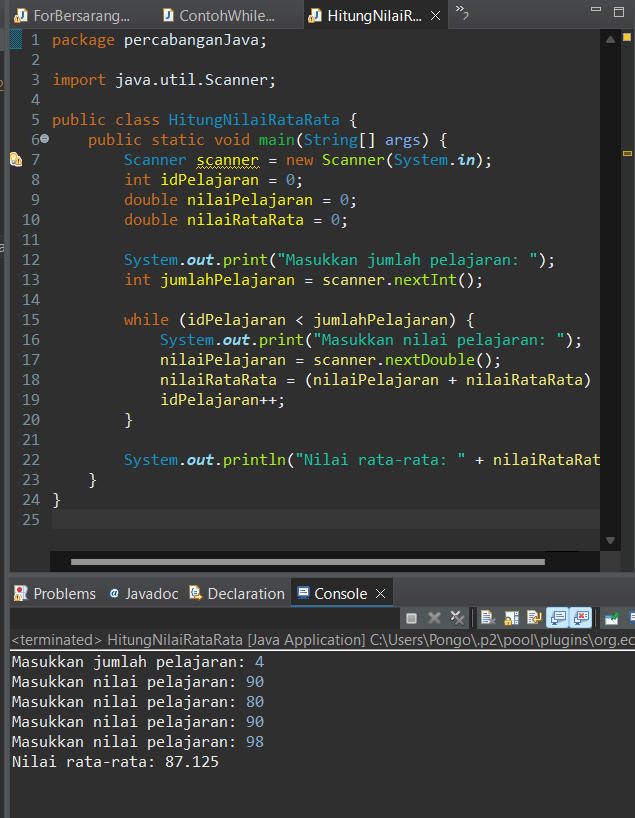
**KODE CONTOH 4**

****

**KODE CONTOH 5**

****

**KODE UNTUK MENYELESAIKAN PSEUDOCODE**

****

1. Analisa luaran yang dihasilkan

**Contoh 4**: Luaran sesuai dengan kode program dimana tadi saya mengganti i%3 == 0 dan menambahkan continue sehingga luaran dihasilkan adalah 1,2,4,5.

**Contoh 5**: Luaran sesuai dan tepat dengan kode program yang telah dibuat. Program mencetak Fakultas Teknik sebanyak 3 kali dari loop do...while. Setelah loop, kata yang sama dicetak sekali lagi, sehingga total ada 4 kali pencetakan.

**Contoh Puseodocode**: Luaran sudah sesuai dengan kode program yang dibuat, program ini dimaksudkan untuk menghitung rata-rata nilai pelajaran.

(Tuliskan penjelasan dari program yang dibuat, apakah kode dan luaran sudah benar?)

**[No.2] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawaban Analisa:

Berdasarkan analisis kode program yang telah dibahas, terdapat beberapa permasalahan terkait dengan penggunaan struktur kontrol dan perhitungan yang tidak akurat. Contoh 4 menunjukkan penggunaan while dan continue untuk mengabaikan angka kelipatan 3, sementara Contoh 5 mengimplementasikan do...while untuk mencetak kata sesuai input pengguna, memastikan minimal satu kali pencetakan.

**Refleksi**

Mengenai materi for dan while, saya belajar bahwa pemilihan antara keduanya sangat bergantung pada konteks, seperti jumlah iterasi yang diketahui atau tidak. Melalui latihan, saya menyadari betapa krusialnya memperhatikan logika agar tidak terjadi kesalahan, seperti infinite loop.