

Nama & NPM	Topik:	Tanggal:
Donna Zelvya Gizella Alvera Deadora G1F024018	For dan While	6 Oktober 2024

UNIT 1 FOR

[No. 1] 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 (int y = 0; y <= 10; ++y) { //ubah 1
            if (y % 2 == 1) //ubah 2
                continue; //baris 1
            else if (y == 8) //ubah 3
                break; //baris 2
            else
                System.out.println(y + " ");
        } } }
```

Luaran contoh 1:

```
0
2
4
6
```

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)
```

```
break kedua; //ubah1
```

```
} } } }
```

Luaran Contoh 2:

```
i = 1; j = 1
```

```
i = 1; j = 2
```

```
i = 2; j = 1
```

```
i = 3; j = 1
```

```
i = 3; j = 2
```

```
i = 4; j = 1
```

```
i = 4; j = 2
```

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 contoh 3:

```
Masukan Input: 7
```

```
*
```

```
**
```

```
***
```

Penjelasan :

Ketiga kode program diatas adalah contoh dari penggunaan loop dalam java, yaitu loop for untuk menghasilkan output yang diinginkan dan memasukkan nilai. For adalah proses perulangan blok kode dengan jumlah angka yang sudah ditentukan. Tipe data yang digunakan pada program tersebut adalah tipe data int yang digunakan untuk menyimpan atau menyatakan data berupa bilangan bulat.

Latihan 1

1.1. Analisa tujuan penulisan kata kunci `continue` dan `break` pada Contoh 1!

Penjelasan :

- Tujuan penulisan dari kata kunci `continue` adalah untuk menghentikan satu perulangan jika kondisi tertentu terjadi, dan berlanjut pada perulangan berikutnya dalam loop. Sebagai contoh pada program di contoh 1 pada kode program `if (y % 2 == 1)` yang berarti jika `y` adalah angka ganjil, maka program akan melewati percetakan dan melanjutkan ke `y` berikutnya.
- Tujuan penulisan kata kunci `break` adalah untuk menghentikan perulangan sebelum perulangan tersebut selesai atau lebih awal daripada yang ditentukan. Pada kode program contoh 1, jika `y` sama dengan 8, maka perulangan akan berhenti dan tidak melanjutkan ke angka yang selanjutnya.

Buat perubahan nilai angka pada variabel di contoh 1

//Ubah 1 menjadi `for (int y = 0; y <= 15; y++)` { lalu running, periksa hasilnya

//Ubah 2 menjadi `if (y % 2 == 0)` lalu running, periksa hasilnya

//Ubah 3 menjadi `else if (y == 9)` lalu running, periksa hasilnya

Analisa dampaknya perubahan ini terhadap luaran setelah running!

Penjelasan :

- Ubah 1 menjadi `for (int y = 0; y <= 15; y++)` , maka setelah di running, output yang dihasilkan adalah 0, 2, 4, 6. Perulangan beroperasi dari 0 sampai dengan 15, dan outputnya juga dipengaruhi oleh kondisi `if` dan `else`.
- Ubah 2 menjadi `if (y % 2 == 0)` , maka program akan mengecek apakah `y` angka genap, dan jika angka genap, kata kunci `continue` akan dijalankan dan semua bilangan genap akan dilewatkan dan hanya mencetak angka ganjil saja.

- c. Ubah 3 menjadi `else if (y == 9)` , maka program akan berhenti dijalankan ketika y sudah mencapai 9 karena kata kunci break telah dijalankan. Angka 9 tidak akan ikut dicetak dan perulangan akan berakhir.

1.2. Buat perubahan kode pada Contoh 2 di baris //Ubah1 menjadi

- a. `continue` pertama; lalu running, periksa hasilnya

Penjelasan :

Ketika break kedua diubah menjadi continue pertama, maka outputnya adalah sama ketika menggunakan kata kunci break kedua dan yang membedakan adalah proses atau alurnya. Perulangan pertama (pertama) adalah perulangan yang mengontrol nilai i dan akan berjalan dari i=1 sampai 1<5. Perulangan kedua (kedua) adalah perulangan yang mengontrol nilai j dan akan berjalan dari j=1 sampai j<3. Saat i=2, program mencetak hasil untuk j=1, yaitu i=2 dan j=1. Ketika program sampai pada kondisi if (i == 2), kata kunci continue pertama akan dijalankan. Ketika kata kunci continue pertama dijalankan, maka program akan melompat atau melanjutkan ke perulangan berikutnya dari loop pertama. Setelah continue pertama, i akan terus bertambah (increment) dan program akan terus mencetak i dan j sampai batas yang telah ditentukan.

- b. `break` pertama; lalu running, periksa hasilnya

Penjelasan :

Break pertama berarti ketika i = 2, maka setelahnya hentikan proses perulangan dan i yang ke-3 sampai i ke-4 tidak akan dilanjutkan.

- c. `continue` kedua; lalu running, periksa hasilnya

Penjelasan :

Ketika menggunakan continue kedua, maka saat I = 2, program akan langsung melompat dan melanjutkan ke pengulangan berikutnya di dalam perulangan kedua. Namun, perulangan kedua akan menyelesaikan semua pengulangan untuk i = 2 sebelum lanjut ke i = 3 dan i = 4.

Analisa perbedaan perubahan kode pada Ubah 1 untuk setiap poin (a), (b), dan (c)!

1.3. Cermati kode contoh 3. Apabila ingin menghasilkan luaran berikut:

Luaran:

Masukan Input: 7

```
****
***
**
*
```

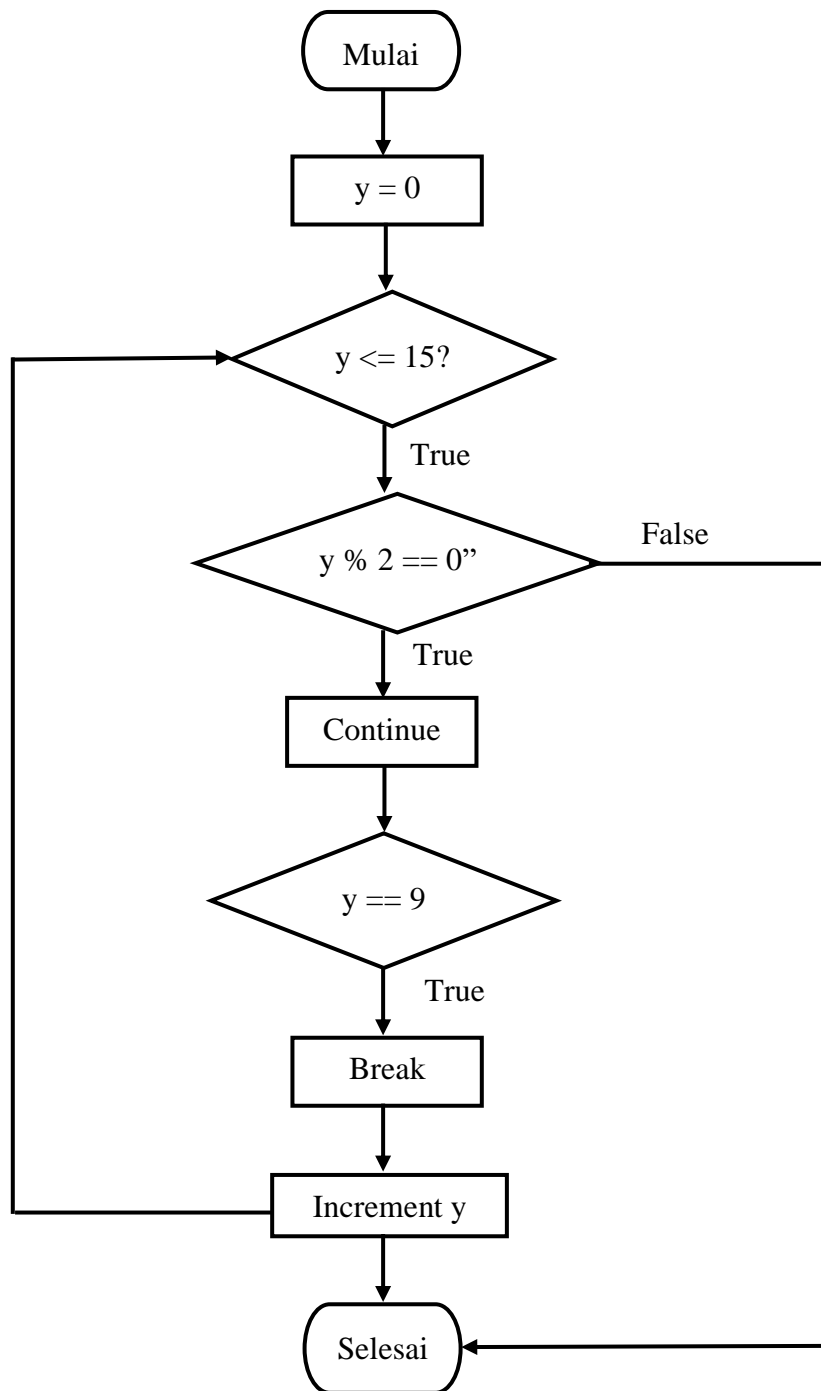
Susunlah analisa kode untuk menghasilkan luaran tersebut!

Penjelasan :

Pada contoh 3, program yang dirancang adalah untuk membuat segitiga dengan variabel input yang dimasukkan oleh pengguna dengan memanggil kode program `import java.util.Scanner;`. Kode program `for(int t=tinggi; t>=1; t--)` diubah menjadi `for(int t=1; t<=tinggi; t++)` untuk mengatur jumlah baris yang akan dicetak. Dimulai dari 1 hingga nilai tinggi yang dimasukkan, yaitu 7. Baris 1: 7 bintang dicetak, baris 2: 6 bintang dicetak, baris 3: 5 bintang dicetak, baris 4: 4 bintang dicetak, baris 5: 3 bintang dicetak, baris 6: 2 bintang dicetak, dan baris 7: 1 bintang dicetak. Jadi, outputnya adalah pola bintang yang menurun dari jumlah maksimum yang pengguna input atau masukkan.

1.4. Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

a. Analisa Flowchart Latihan 1.1

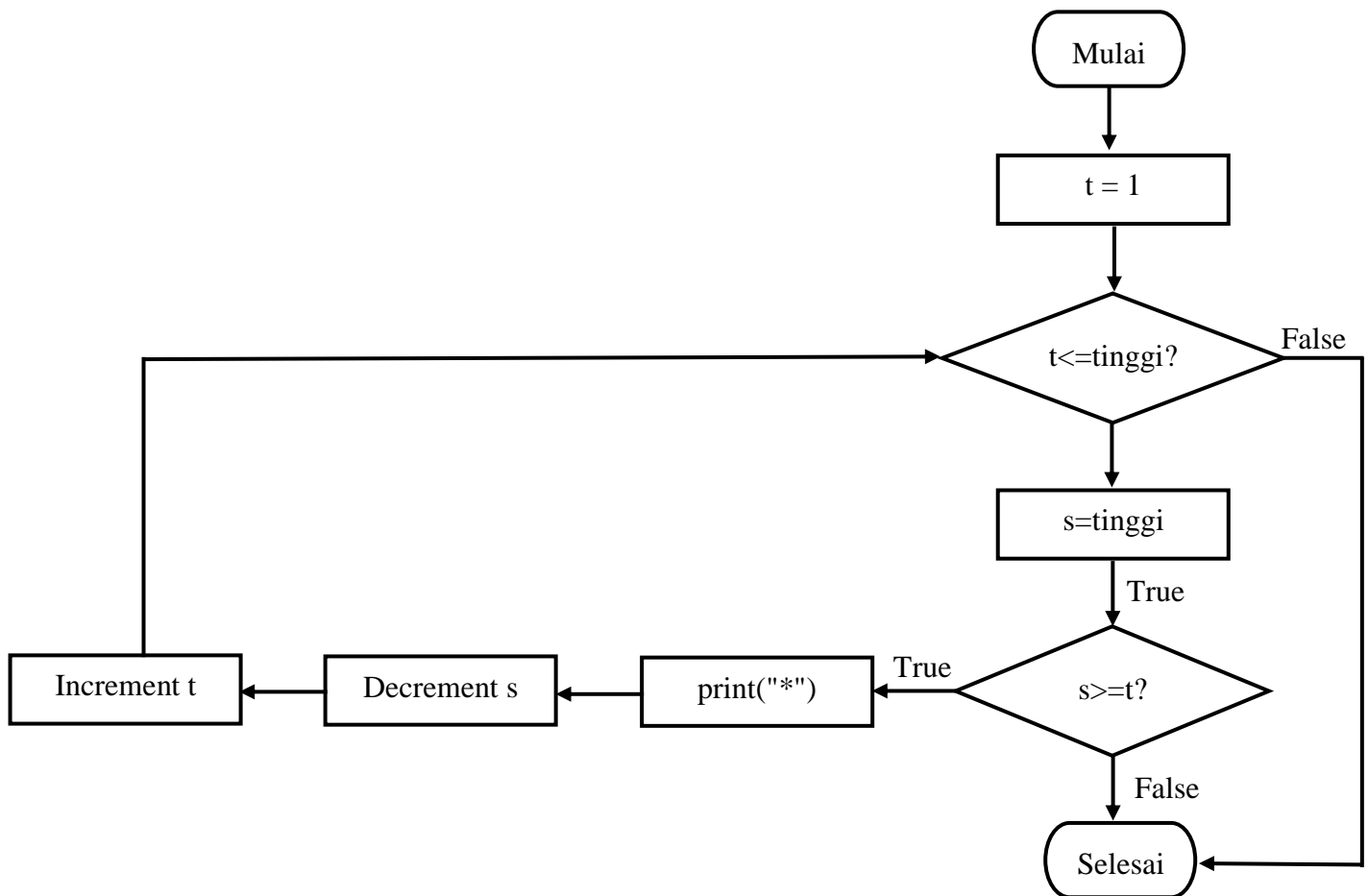


Gambar Flowchart Latihan 1.2

1. Mulai: Proses dimulai
2. Inisialisasi: Nilai dari variabel y adalah 0
3. Cek kondisi: Memeriksa apakah y kurang dari atau sama dengan 15. Jika tidak, program berhenti.
4. Perintah jika benar: Jika y adalah bilangan ganjil (%2), maka program melanjutkan ke pengulangan berikutnya (continue) tanpa mencetak y. Jika y mencapai 9. Maka program akan menjalankan perintah break. Jika y adalah bilangan genap dan tidak sama dengan 9, maka nilai y akan dicetak.

5. Increment: Nilai y ditambah 1 dan kembali ke kondisi loop.
6. Selesai

b. Analisa Flowchart Latihan 1.3



Gambar Flowchart Latihan 1.3

1. Mulai: Proses dimulai
2. Inisialisasi: Mengambil input tinggi dari pengguna
3. Loop tinggi: Mengatur tinggi piramida dari yang dimasukkan pengguna hingga 1
4. Kondisi Loop: Memeriksa apakah t lebih besar atau sama dengan 1 untuk melanjutkan proses. Menghitung jumlah bintang yang dicetak berdasarkan nilai t.
5. Perintah jika benar: Mencetak bintang untuk setiap pengulangan dari loop dan mencetak baris baru setelah selesai.

[No.1] Analisis dan Argumentasi

Dalam ketiga contoh kode program yang diberikan, struktur perulangan yang digunakan adalah for dengan kata kunci continue dan break. Kata kunci continue digunakan untuk melewati perulangan jika kondisi tertentu terpenuhi. Contohnya pada contoh 1 digunakan untuk melewati angka ganjil dan hanya mencetak angka genap. Kata kunci break

digunakan untuk menghentikan suatu pengulangan jika sudah mencapai kondisi tertentu. Contohnya saat perulangan sudah mencapai 8, maka break akan dijalankan. Pada ketiga contoh tersebut, pengguna diminta untuk mengubah beberapa kode program lalu cek dan Analisa hasilnya.

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma

- Algoritma Kode Program 1.1

- (a) Mulai
- (b) Inisialisasi variabel $y = 0$
- (c) Cek kondisi $y \leq 15$
- (d) Jika y bilangan genap, kata kunci continue dijalankan
- (e) Jika y mencapai 9, kata kunci break dijalankan
- (f) Mencetak nilai y
- (g) Selesai.

- Algoritma Kode Program 1.2

- (a) Mulai
- (b) Inisialisasi variabel i dengan 1
- (c) i akan terus berjalan selama $i < 5$
- (d) Inisialisasi variabel j dengan 1
- (e) j akan terus berjalan selama $j < 3$
- (f) Jika $i = 2$, kata kunci continue akan dijalankan
- (g) Jika $i = 2$, kata kunci break dijalankan
- (h) Increment j dengan 1 ($j++$)
- (i) Increment i dengan 1 ($i++$)
- (j) Menampilkan kombinasi i dan j
- (k) Selesai.

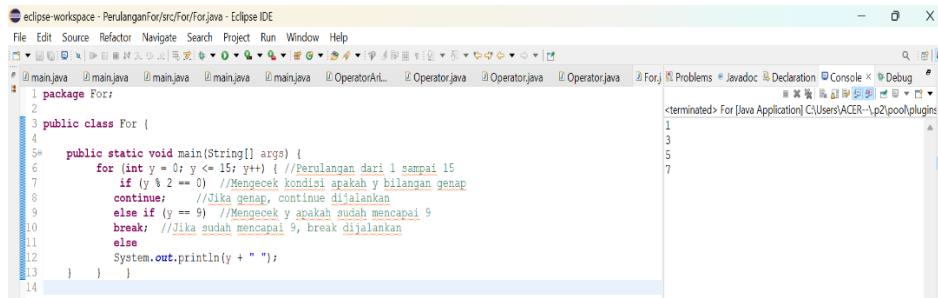
- Algoritma Kode Program 1.3

- (a) Mulai
- (b) Membuat objek scanner untuk membaca input pengguna
- (c) Menampilkan pesan "Masukan Input:"
- (d) Membuat variabel s untuk mencetak bintang (*)
- (e) Memasukkan jumlah bintang
- (f) Menampilkan piramida
- (g) Selesai.

2) Kode program dan luaran

a) Screenshot/ Capture potongan kode dan hasil luaran

Latihan 1.1

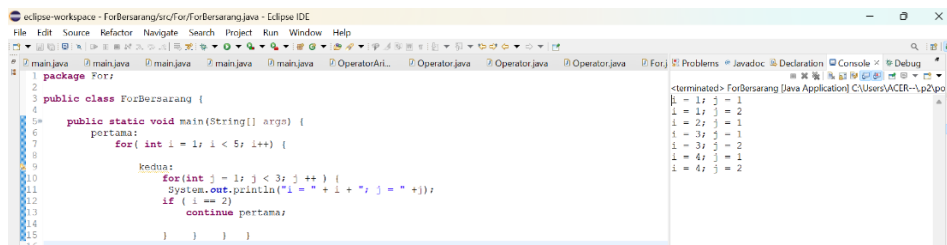


```

1 package For;
2
3 public class For {
4
5     public static void main(String[] args) {
6         for (int y = 0; y <= 15; y++) { //Perulangan dari 1 sampai 15
7             if (y % 2 == 0) //Mengecek kondisi apakah y bilangan genap
8                 continue; //Jika genap, continue dijalankan
9             else if (y == 9) //Mengecek y apakah sudah mencapai 9
10                 break; //Jika sudah mencapai 9, break dijalankan
11             else
12                 System.out.println(y + " ");
13         }
14     }
15 }

```

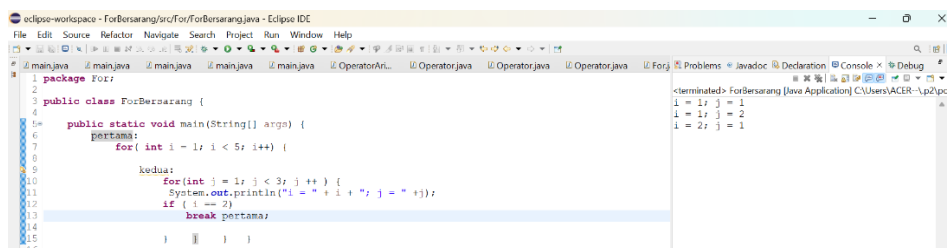
Latihan 1.2



```

1 package For;
2
3 public class ForBersarang {
4
5     public static void main(String[] args) {
6         pertama:
7         for (int i = 1; i <= 5; i++) {
8
9             kedua:
10             for (int j = 1; j <= 3; j++) {
11                 System.out.println("i = " + i + ", j = " + j);
12                 if (i == 2)
13                     continue pertama;
14             }
15         }
16     }
17 }

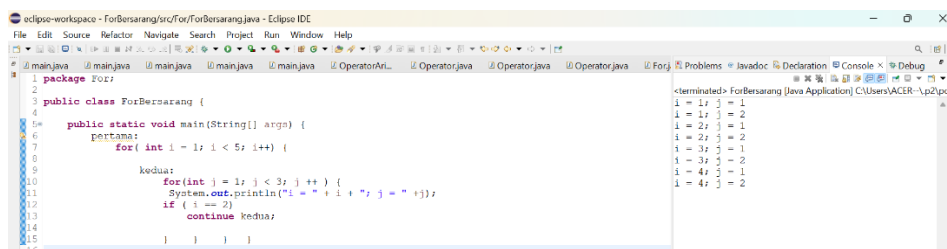
```



```

1 package For;
2
3 public class ForBersarang {
4
5     public static void main(String[] args) {
6         pertama:
7         for (int i = 1; i <= 5; i++) {
8
9             kedua:
10             for (int j = 1; j <= 3; j++) {
11                 System.out.println("i = " + i + ", j = " + j);
12                 if (i == 2)
13                     break pertama;
14             }
15         }
16     }
17 }

```

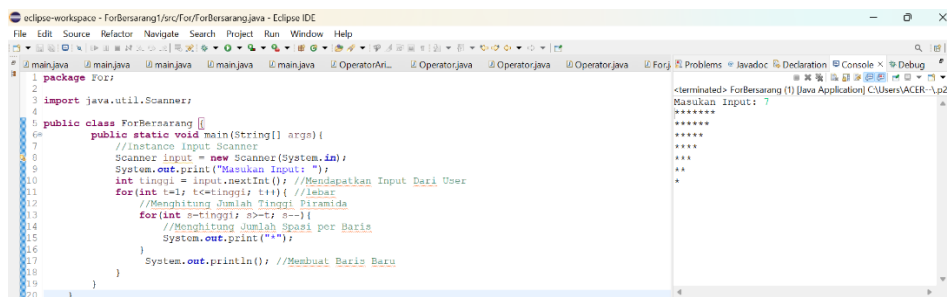


```

1 package For;
2
3 public class ForBersarang {
4
5     public static void main(String[] args) {
6         pertama:
7         for (int i = 1; i <= 5; i++) {
8
9             kedua:
10             for (int j = 1; j <= 3; j++) {
11                 System.out.println("i = " + i + ", j = " + j);
12                 if (i == 2)
13                     continue kedua;
14             }
15         }
16     }
17 }

```

Latihan 1.3



```

1 package For;
2
3 import java.util.Scanner;
4
5 public class ForBersarang {
6
7     public static void main(String[] args) {
8         //Instance Input Scanner
9         Scanner input = new Scanner(System.in);
10         System.out.print("Masukan Input: ");
11         int tinggi = input.nextInt(); //Mendapatkan Input Dari User
12         for (int i = 1; i <= tinggi; i++) { //Lebar
13             //Menghitung Jumlah Tinggi Piramida
14             for (int k = tinggi; k >= i; k--) {
15                 System.out.print(" ");
16             }
17             System.out.println(); //Membuat Baris Baru
18         }
19     }
20 }

```

b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan pada setiap program sudah tepat dan sesuai dengan perintah yang ada dengan mengubah beberapa kode program dan memasukkan angka untuk membentuk piramida dengan Bintang. Contohnya pada program

latihan 1.1, luaran yang dihasilkan adalah bilangan ganjil (bilangan genap dilewatkan) dengan batas y adalah 9 (9 tidak termasuk).

[No.1] Kesimpulan

1) Analisa

Dari ketiga contoh program loop for yang diberikan, dapat dilihat bagaimana implementasi atau penerapan dari perulangan for dengan menggunakan kata kunci continue dan break. Pada contoh 1, continue digunakan untuk melewati bilangan genap (hanya bilangan ganjil yang dicetak) dan break digunakan untuk memberhentikan perulangan ketika sudah mencapai $y = 9$, sehingga 9 tidak ikut dicetak. Pada contoh 2, program yang dijalankan adalah menampilkan kombinasi antara nilai i dan j. Ketika i mencapai 2, break akan dijalankan dan ketika continue dijalankan, maka akan melanjutkan ke perulangan selanjutnya. Pada contoh 3, program yang dirancang adalah meminta pengguna untuk memasukkan nilai yang nantinya akan membentuk segitiga dengan Bintang.

UNIT 2 WHILE

[No. 2] Identifikasi Masalah:

1) 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.

```

public class WhileBersarang {
    public static void main(String[] args) {
        int count = 0; //ubah1
        while (count < 20) {
            if (count % 3 == 0) //ubah2
                System.out.println(count);
            count++;
        }
    }
}

```

Luaran:

```

0
3
6
9
12
15
18

```

Latihan 2

- 2.1. Buat perubahan nilai angka pada variabel di Contoh 4

//Ubah 1 menjadi continue; lalu running, periksa hasilnya

Analisa dampaknya perubahan terhadap luaran setelah running dan uraikan kegunaan break dan continue!

Penjelasan :

Ketika menggunakan break, program akan mencetak angka 1, 2, 3 dan ketika mencapai 4, maka kata kunci break akan dijalankan untuk menghentikan semua perulangan. Kata kunci break berfungsi untuk menghentikan perulangan sepenuhnya sebelum kondisi perulangan mencapai batas yang telah ditentukan. Ketika break diubah menjadi continue, maka program akan mencetak angka 1, 2, 3, 4, 5, dan 6. Hal ini karena ketika $1 = 4$, continue akan dijalankan dan akan melompat kembali ke awal perulangan while. Langkah-langkah berikutnya di dalam perulangan tidak akan dijalankan. Namun, pernyataan $i++$ tetap dijalankan sebelum memasuki loop lagi.

2.2. Buat perubahan nilai angka pada variabel di Contoh 5

//Ubah2 menjadi `if (count % 5 == 0)` lalu running, periksa hasilnya

Analisa dampaknya perubahan terhadap luaran setelah running dan uraikan kegunaan % untuk angka yang berbeda pada perintah tersebut!

Penjelasan :

Operator modulus (%) adalah operator yang digunakan untuk menghitung sisa pembagian antara dua bilangan. Dengan `if (count % 3 == 0)`, maka output yang dicetak adalah angka atau bilangan yang ketika dibagi 3 hasilnya 0 (bilangan habis dibagi 3). Ketika `if (count % 3 == 0)` diubah menjadi `if (count % 5 == 0)`, maka outputnya adalah bilangan atau angka yang ketika dibagi 5 hasilnya 0 (bilangan habis dibagi 5).

2.3. Buat perubahan nilai angka pada variabel di

//Ubah1 menjadi `while (count < 0) {` lalu running, periksa hasilnya

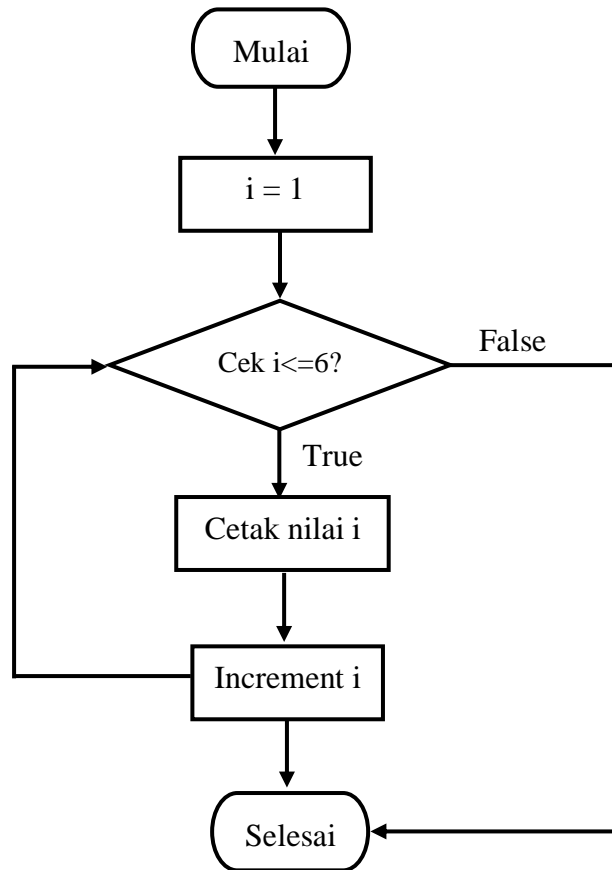
Ubahlah baris kode `while` pada Contoh 5 menjadi `do ... while` dengan persyaratan yang sama `while (count < 0)`. Bandingkan hasil luaran antara menggunakan `while` dan `do ... while`!

Penjelasan :

Ketika diubah menjadi `while (count < 0) {` lalu running, tidak ada output yang dihasilkan karena 0 tidak kurang dari 0 sehingga kondisi `count < 0` false (salah atau tidak terpenuhi). Juga tidak ada perulangan yang terjadi karena perulangan hanya akan terjadi jika count kurang dari 0. Ketika diubah menjadi `do...while`, output yang dihasilkan adalah 0. Hal ini karena dengan `do...while`, kode akan dijalankan setidaknya sekali sebelum kondisi diperiksa. Jadi, pada perulangan pertama, Dengan `do...while`, blok kode di dalam `do` dieksekusi setidaknya sekali sebelum kondisi diperiksa. Jika true akan Kembali ke `do`, tetapi jika false maka perulangan akan berhenti.

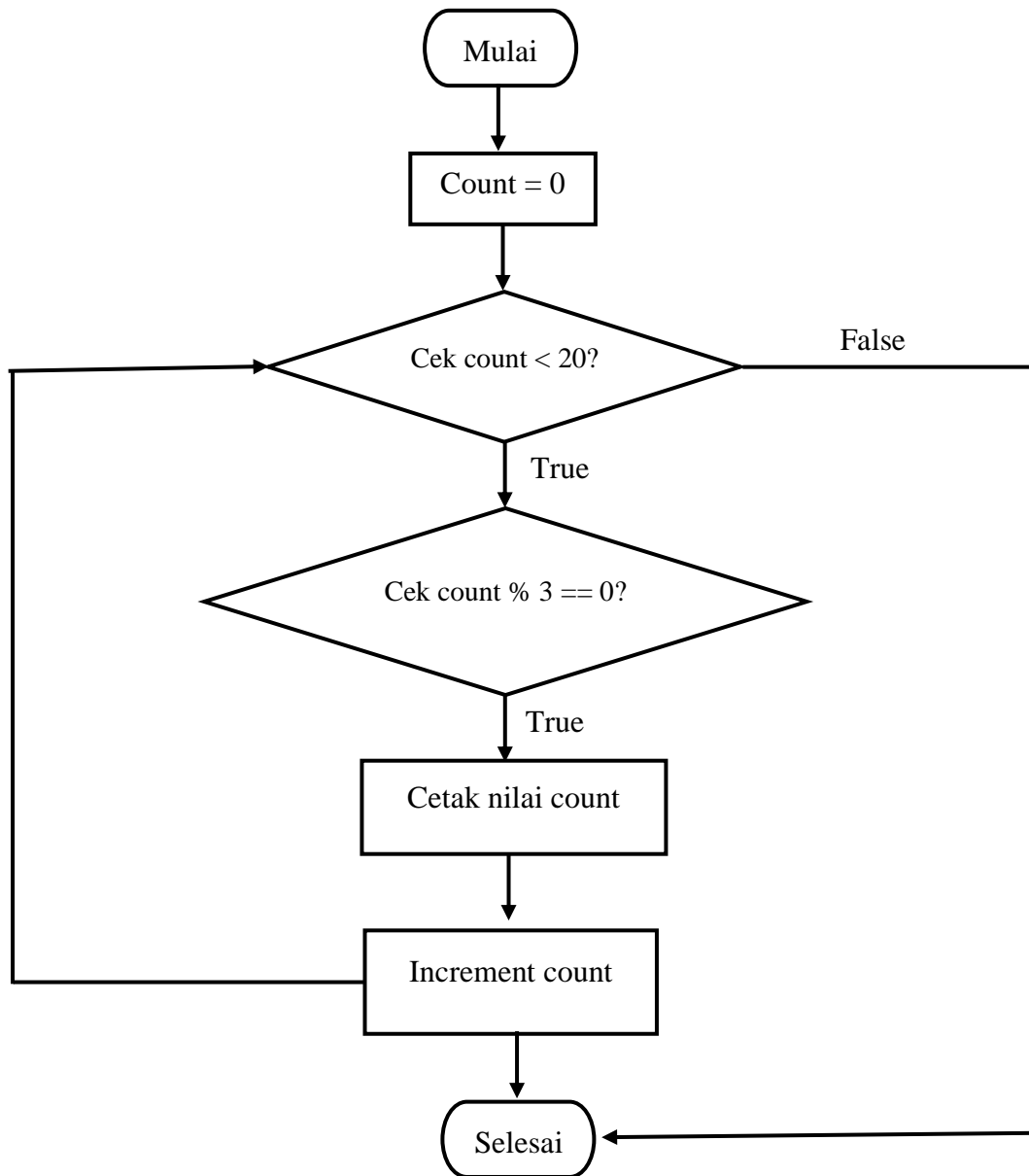
2.4. Analisa diagram flowchart dari Latihan 2.1, Contoh 5, dan Latihan 2.3!

a. Analisa flowchart 2.1



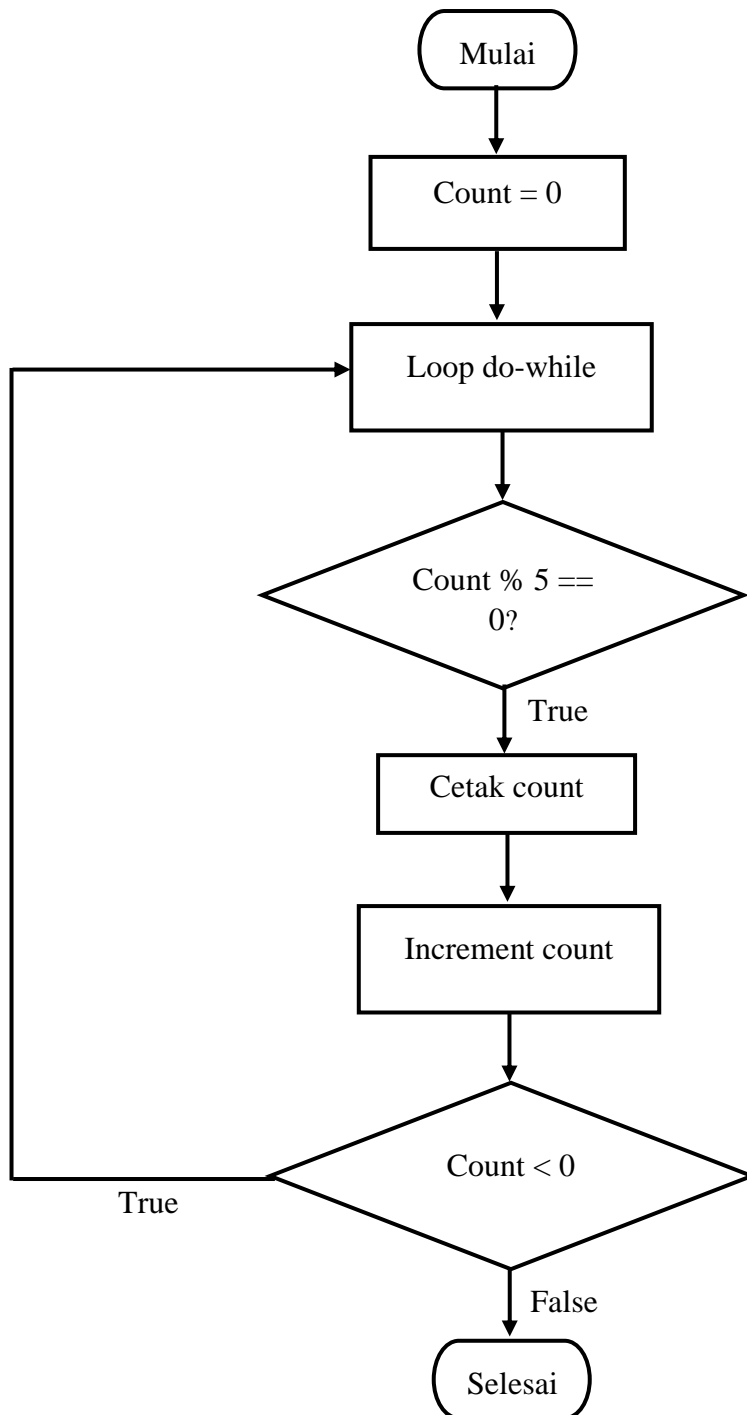
1. Mulai: Proses akan dimulai
2. Inisialisasi: Inisialisasi i dengan 1
3. While: Cek kondisi $i \leq 6$, jika kondisi benar, maka lanjut ke cetak nilai i. Jika salah, program akan berakhir
4. Cetak nilai i: Nilai i akan ditampilkan
5. Increment i: Nilai i ditambah 1
6. Proses akan kembali untuk memeriksa kondisi $i \leq 6$
7. Selesai: Proses telah selesai

b. Analisa Flowchart contoh 5



1. Mulai: Proses akan dimulai
2. Inisialisasi: Inisialisasi count dengan 0
3. Cek count<20: Memeriksa apakah count kurang dari 20. Jika true (benar), maka akan dilanjutkan ke program berikutnya
4. Cek count % 3 == 0: Memeriksa apakah count habis dibagi 3. Jika true (benar), maka akan ke program selanjutnya. Jika false (salah), kembali lagi ke langkah 3.
5. Cetak nilai count: Nilai count akan dicetak
6. Increment count: Nilai count ditambah 1
7. Jika count >= 20, program berakhir.

c. Analisa Flowchart Latihan 2.3



1. Mulai: Proses akan dimulai
2. Count = 0: Inisialisasi count dengan angka 0
3. Loop do while: Loop do-while akan dijalankan
4. Count % 5 == 0?: Memeriksa apakah count habis dibagi (tanpa sisa bagi) dengan 5. Jika count habis dibagi 5, maka kondisi akan bernilai true (benar) dan program akan mencetak nilai count. Jika false (salah), maka program akan melewati langkah cetak dan melanjutkan ke langkah berikutnya.
5. Increment count: Setelah memeriksa dan mencetak nilai count, program akan menambahkan nilai count sebanyak 1.

6. $\text{Count} < 10$: Program akan memeriksa apakah count masih < 0 . Jika true (benar), maka program kembali ke langkah 3 untuk pengulangan berikutnya. Jika false (salah), maka program akan berakhir.
7. Selesai.

[No.2] Analisis dan Argumentasi

Dari kedua kode program yang diberikan merupakan penerapan while, yaitu proses perulangan blok kode dalam jumlah angka tidak terhitung dan proses pengulangan blok pernyataan dilakukan selama kondisinya bernilai true. Selain while, terdapat do-while yang merupakan proses perulangan blok kode dengan jumlah angka tidak terhitung dan proses pengulangan suatu blok pernyataan dieksekusi paling tidak satu kali dan selama kondisinya bernilai true. Saya mengusulkan pada program yang diberikan untuk mengubah kode program sesuai dengan perintah, seperti mengubah kata kunci, while menjadi do-while, dan lain-lain.

[No.2] Penyusunan Algoritma dan Kode Program

1) Algoritma

- Algoritma Latihan 2.1

- (a) Mulai
- (b) Inisialisasi variabel
- (c) Selama $i \leq 6$, cetak nilai i
- (d) Increment nilai i
- (e) Jika $i = 4$, continue akan dijalankan
- (f) Perulangan terus berlanjut sampai $i \leq 6$
- (g) Menampilkan output
- (h) Selesai.

- Algoritma Latihan 2.2

- (a) Mulai
- (b) Inisialisasi variabel
- (c) Selama $\text{count} < 20$, cek kondisi menggunakan operator modulus
- (d) Jika count habis dibagi 5, cetak nilai count ke layar
- (e) Selesai.

- Algoritma Latihan 2.3

- (a) Mulai
- (b) Inisialisasi variabel
- (c) Cek kondisi count dengan operator modulus
- (d) Jika count habis dibagi 5, cetak nilai count

- (e) Ulangi langkah tersebut selama $\text{count} < 0$
 - (f) Selesai.
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran

Latihan 2.1

```

package Loop;

public class While {

    public static void main(String[] args) {
        int i=1; //inisialisasi variabel
        while(i<=6) { //Cek kondisi
            //Perintah kode yang dijalankan jika kondisi benar
            System.out.println(i);
            i++; //Increment
            if(i==4) { //Cek apakah i=4
                continue;
            }
        }
    }
}

```

Latihan 2.2

```

package Loop;

public class WhileBersarang {

    public static void main(String[] args) {
        int count = 0; //inisialisasi variabel
        while (count < 20) { //Cek kondisi count selama < 20
            if (count % 5 == 0) { //Cek apakah count habis dibagi 5
                System.out.println(count);
                count++; //Increment
            }
        }
    }
}

```

Latihan 2.3

```

package Loop;

public class WhileBersarang {

    public static void main(String[] args) {
        int count = 0; //inisialisasi variabel
        do { //Eksekusi satu kali
            if (count % 5 == 0) { //Cek kondisi count habis dibagi 5
                System.out.println(count);
                count++;
            }
        } while (count < 20);
    }
}

```

- b) Analisa luaran yang dihasilkan

Mengubah kata kunci maupun kode program pada loop tentunya mempengaruhi luaran atau outputnya. Setelah mengubah beberapa kode program atau kata kunci sesuai dengan perintah, luaran yang dihasilkan sudah sesuai dan tepat setelah mengubah beberapa kode program atau kata kunci sesuai dengan perintah. Setiap perubahan pada kata kunci maupun kode program tentunya memberikan pemahaman mengenai fungsi dari break, continue, dan operator modulus dalam konteks loop.

[No.2] Kesimpulan

- 1) Analisa

Pada contoh 4 dan contoh 5, kode program menerapkan struktur perulangan, yaitu while dan do-while. Pada while, proses pengulangan blok pernyataan dilakukan selama kondisinya bernilai true. Dan pada do-while, proses pengulangan suatu blok pernyataan dieksekusi paling tidak satu kali dan selama kondisinya bernilai true. Di contoh 4 dan 5 juga pengguna diminta mengubah beberapa kata kunci dan kode

program. Kata kunci break dan kata kunci continue yang memiliki kegunaan masing-masing, serta penggunaan operator modulus (%) untuk menentukan sisa bagi antar dua bilangan atau angka. Setiap perubahan dan penabahan pada program, maka akan mempengaruhi output yang ditampilkan.

Refleksi

Pada tugas kali ini, materi yang dibahas dan di praktekan adalah konsep serta struktur perulangan khususnya menggunakan for dan while. Tentu saja setiap minggunya Saya mendapatkan pengalaman belajar dan pengetahuan baru. Saya mendapatkan pemahaman mengenai bagaimana konsep for, while, dan do-while, fungsi dari break dan continue, mengubah kata kunci lalu menganalisis outputnya, dan lain-lain. Tantangan yang Saya hadapi adalah masih bingung dalam membedakan while dan do while. Selain itu, Saya masih sering kebingungan dalam membuat flowchart. Namun, Saya terus berlatih dan mempelajari setiap hal yang dirasanya perlu pemahaman lebih lanjut baik melalui teman maupun menonton youtube.