|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Khusnul Hidayati**  **G1F024032** | **For dan While Java** | **05 Oktober 2024** |

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

1. Uraikan permasalahan dan variabel

Jawab:

**UNIT 1 FOR**

Pernyataan for dapat mengendalikan proses berulang dengan jumlah perulangan yang sudah ditentukan.

Bentuk umum For:

for (inisialisasi; kondisi; iterasi) {  
   pernyataan;  
}

Makna bagian for:  
1.    inisialisasi untuk memberikan nilai kepada variabel yang digunakan untuk mengontrol pengulangan.   
2.    kondisi untuk mengontrol pengulangan untuk dilanjutkan atau diakhiri.   
3.    Penaikan (increment) atau penurunan (decrement) untuk menaikkan atau menurunkan nilai variabel perulangan.

**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  
\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*\*  
\*\*\*\*\*\*\*

**Latihan 1**

1.1.    Analisa tujuan penulisan kata kunci continue dan break pada Contoh 1!   
          Buat perubahan nilai angka pada variabel di   
          //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!

Jawab:

* Pada perubahan 1 kode akan mencetak angka 0 hingga 15, tapi karena ada break di anggka 8, jadi loop hanya mencetak 0, 2, 4, 6.
* Pada perubahan 2 hasil yang dicetak hanya angka ganjil, yaitu 1, 3, 5, 7, 9, 11, 13, 15.
* Pada perubahan 3 saat kode tersebut digunakan, program akan mencetak angka 0, 2, 4, 6, dan 8. Tetapi Continue akan melewati angka genap, dan Break akan menghentikan loop setelah y mencapai 9, jadi hasil yang dicetak adalah 1, 3, 5, 7.

1.2.    Buat perubahan kode pada Contoh 2 di baris //Ubah1 menjadi  
          a. continue pertama;  lalu  running, periksa hasilnya  
          b. break pertama; lalu running, periksa hasilnya  
          c. continue kedua; lalu running, periksa hasilnya  
          Analisa perbedaan perubahan kode pada Ubah 1 untuk setiap poin (a), (b), dan (c)

Jawab:

* Perubahan 1, melanjutkan loop pertama, hasilnya yaitu mencetak semua kombinasi i dari 1 sampai 4 tetapi melewatkan output untuk j saat i adalah 2.
* Perubahan 2, menghentikan loop pertama, sehingga program hanya mencetak output untuk i = 1 dan i = 2, dan ketika i mencapai 2, program langsung keluar dari kedua loop.
* Perubahan 3, ketika i mencapai 2, program akan terus melanjutkan iterasi di loop kedua (j), dan akan mencetak nilai j yang lain untuk i = 2.

1.3.    Cermati kode contoh 3. Apabila ingin menghasilkan luaran berikut:  
Luaran:  
Masukan Input: 7  
\*\*\*\*\*\*\*  
\*\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*  
\*\*\*  
\*\*  
\*  
           Susunlah analisa kode untuk menghasilkan luaran tersebut!

1.4.      Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

Jawab:

1. Latihan 1.2

Analisis diagram flowchart:

Program ini memiliki struktur yang jelas, terdiri dari dua loop bersarang yang mencetak setiap kombinasi dua variabel. Berhubungan dengan kondisi yang ditentukan, keputusan flowchart menunjukkan apakah program dapat melanjutkan atau tidak. Ini meningkatkan fleksibilitas dalam alur kerja program.

Inisialisasi for i = 1

Inisialisasi for j = 1

Cetak i : j

Apakah lanjut?

Ya Tidak

Lanjut ke j

Akhiri for j

Akhiri for i

1. Latihan 1.3

Analisis diageam flowchart:

Program ini menghasilkan pola bintang terbalik berdasarkan input yang diberikan. Struktur loop yang digunakan efektif untuk mencetak baris bintang secara berurutan, dan langkah-langkah di dalam flowchart menunjukkan proses yang logis dari input hingga output.

Input tinggi

Inisialisasi for t = tinggi

Inisialisasi for s = 1

Cetak “\*”

Akhiri For s

Cetak baris baru

Akhiri For t

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

Jawab:

Menurut saya solusi yang dapat diberkan adalah dengan menganalisis dampah perubahan loop, seperti mengubah batasan iterasi dan logika dalam percabangan, yang memungkinkan program untuk menceta angka atau pola bintang sesuai kebutuhan. Dengan memodifikasi kode program dengan tepat, kita dapat membuat hasil keluaran yang diharapkan, seperti mencetak angka ganjil dan membuat piramida terbalik. Analisis ini bertujuan untuk memahami cara kerja control alur dalam pemrograman untuk menghasilkan output yang efektif dan efisien.

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

1. Algoritma

Jawab:

Algoritma adalah langkah-langkah penyelesaian masalah.  
a) Latihan 1.1

* Mulai
* Menganalisa program
* Mengubah //Ubah 1 menjadi  for (int y = 0; y <= 15; y++) {
* Running dan periksa perubahan outputnya
* Mengubah //Ubah 2 menjadi  if (y % 2 == 0)
* Running dan periksa perubahan outputnya
* Mengubah //Ubah 3 menjadi else if (y == 9)
* Running dan periksa kembali hasil outputnya
* Menganalisa dampak perubahannya terhadap luaran setelah di running
* Selesai

b) Latihan 1.2

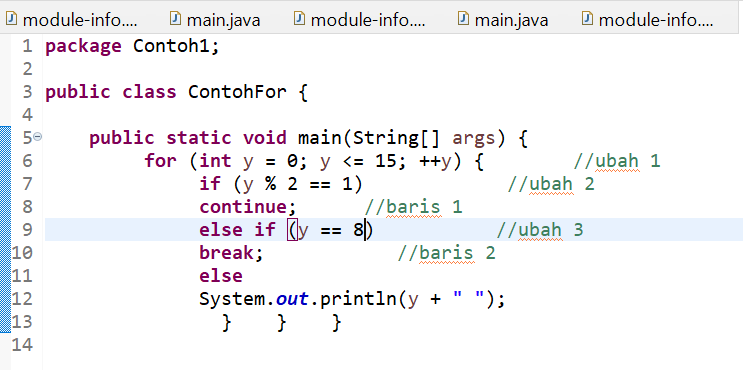
* Mulai
* Mengubah //Ubah1 menjadi continue pertama;
* Running dan periksa hasilnya
* Mengubah //Ubah1 menjadi break pertama;
* Running dan periksa hasilnya
* Mengubah //Ubah1 menjadi continue kedua;
* Running dan periksa hasilnya
* Menganalisis perbedaan berubahan //Ubah1 untuk setiap percobaan.
* Selesai

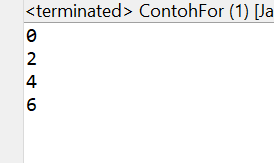
c) Latihan 1.3

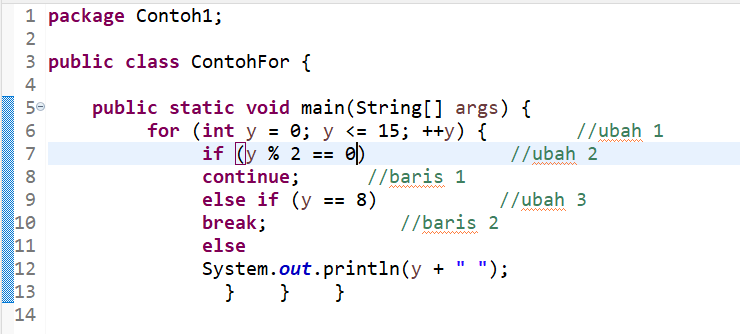
* Mulai
* Mengganti baris for(int t=tinggi; t>=1; t--){ pada contoh 3 menjadi **for**(**int** t=1; t<=tinggi; t++) {
* Running dan masukkan angka sesuia yang tertera dalam soal
* Selesai

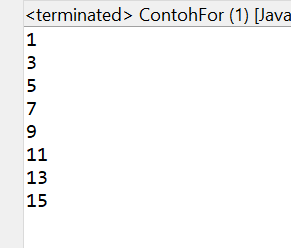
1. Kode program dan luaran
2. Screenshot/ Capture potongan kode dan hasil luaran

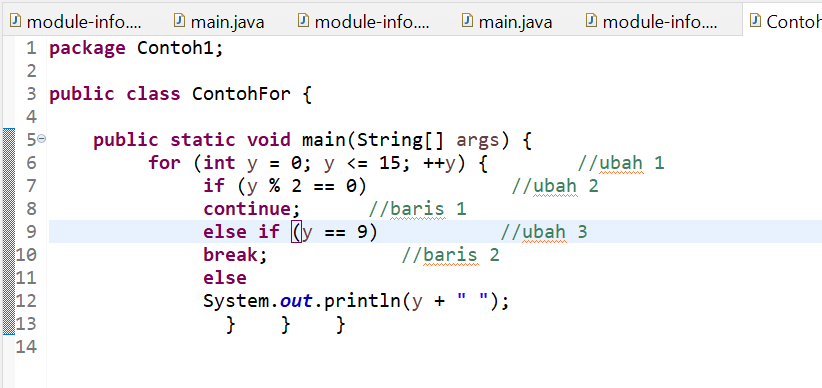
* Latihan 1.1

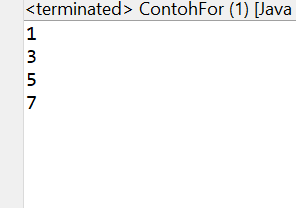




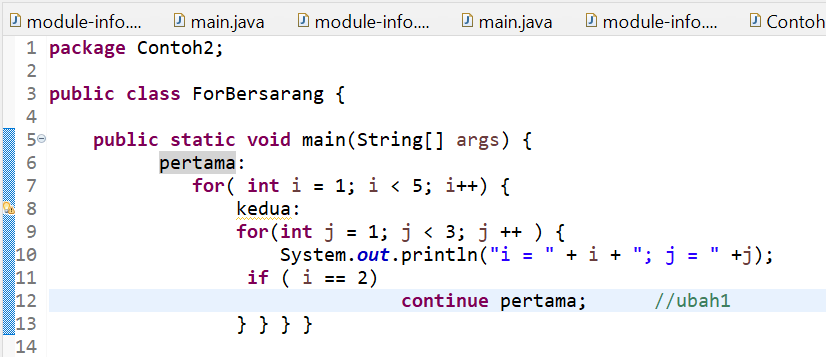


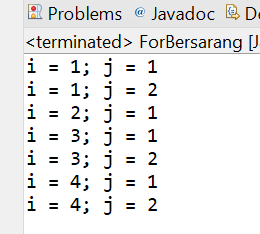


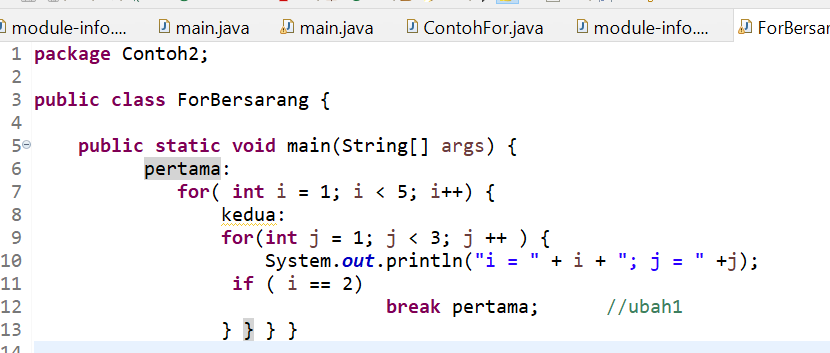


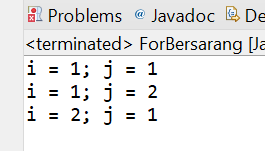


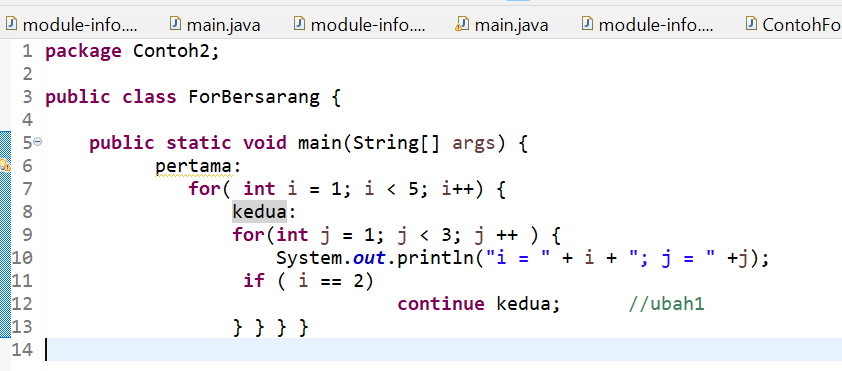
* Latihan 1.2

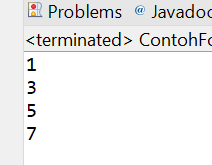




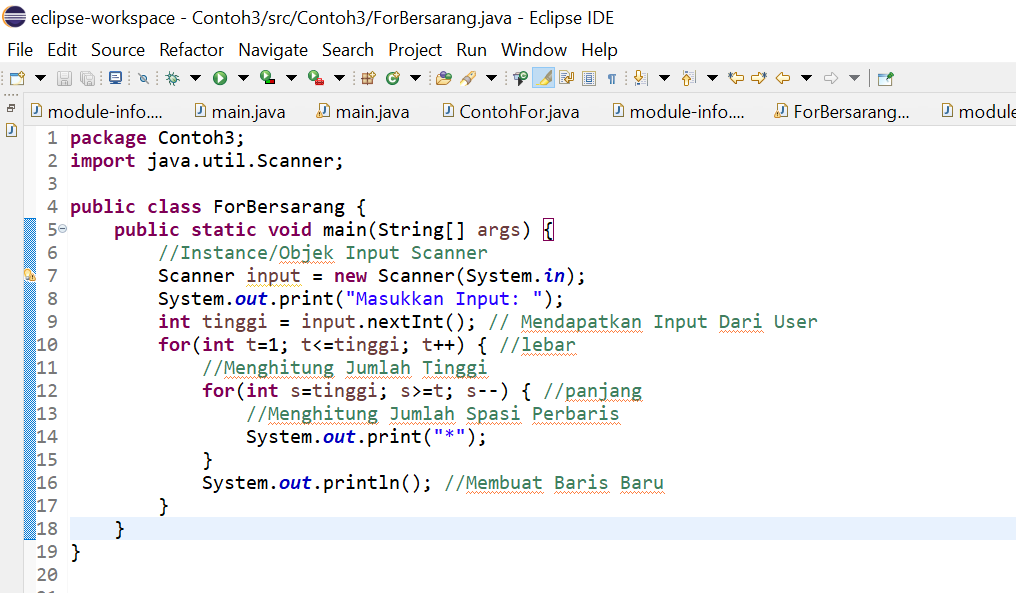


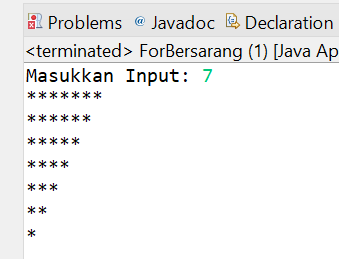






* Latihan 1.3





1. Analisa luaran yang dihasilkan   
   Jawab:

Luaran sudah telah dengan program yang disusun.   
Tipe data yang ditampilkanjuga sudah sesuai dengan kebutuhan dan permintaan data.

**[No.1] Kesimpulan**

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Jawab:

Dengan mempertimbangkan permasalahan, algoritma, dan kode program, dapat disimpulkan bahwa programmer yang memahami cara kerja pernyataan for, continue, dan break dapat mengendalikan alur program dengan baik. Dengan mengubah batasan iterasi dan logika percabangan, program dapat mencetak angka atau pola bintang sesuai kebutuhan.

1. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab:

Dasar alasan pengambilan keputusan ini adalah untuk meningkatkan efisiensi dan efektivitas kode program. Dengan melakukan analisis mendalam terhadap setiap perubahan yang dilakukan pada kode, programmer dapat lebih memahami pengaruh dari setiap struktur kontrol dan variabel, yang pada akhirnya akan menghasilkan output yang lebih sesuai dengan yang diinginkan.

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

1. Uraikan permasalahan dan variabel

Jawab:

**UNIT 2 WHILE**

Pernyataan while berguna untuk melakukan proses yang berulang. Pernyataan akan dijalankan terus menerus selama kondisi bernilai true (benar). Oleh karena itu, kita harus membuat kondisi suatu saat ungkapan bernilai salah agar perulangan berakhir.

Bentuk umum While seperti:

while (kondisi) {   
        pernyataan;  
}

**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!

Jawab:

* Ketika menggunakan break, saat nilai i mencapai 4, maka loop langsung dihentikan. Berarti nilai yang di cetak hanya 1, 2, 3. Fungsi break digunakan untuk keluar dari loop secara langsung, sehingga tidak ada nilai yang dikeluarkan lagi setelah break.
* Ketika menggunakan continue, saat nilai I mencapai 4, continue akan membuat program melompati sisa iterasi dalam loop untuk siklus saat itu dan langsung melanjutkan ke iterasi berikutnya.

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!

Jawab:

* Pada kondisi if (count % 3 == 0)  , program mencetak semua angka dari 0 sampai 20 yang habis dibagi 3 tanpa sisa.
* Setelah diganti menjadi if (count % 5 == 0)  , program hanya mencetak angka dai 0 sampai 20 yang habis dibagi 5.
* Kegunaan % yaitu untuk mendapatkan sisa dari pembagian dua angka, atau biasanya digunakan untuk memeriksa apakah suatu angka adalah kelipatan dari angka lain.

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!

Jawab:

* Saat menggunakan while (count < 0) {  , tidak ada eksekusi dari blok kode karena kondisi tidak terpenuhi sehingga hasil luarannya adalah kosong (tidak ada output).
* Saat mengubah baris kode while pada Contoh 5 menjadi do … while, blok kode dieksekusi sekali terlebih dahulu, dan pada eksekusi pertama, count adalah 0, yang memenuhi syarat count % 3 == 0. Sehingga luaran yang dihasilkan adalah 0.

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

Jawab:

1. Latihan 2.1

Analisis Diagram Flawchart:

Diagram tersebut menggambarkan logika pengulangan menggunakan pernyataan while dalam sebuah program. Diagram ini dimulai dengan inisialisasi variabel i dan memeriksa kondisinya untuk menentukan apakah program harus melanjutkan pencetakan nilai. Dengan menggunakan percabangan, flowchart menunjukkan bagaimana pernyataan break dapat menghentikan perulangan ketika i mencapai nilai 4, sehingga hanya nilai 1, 2, dan 3 yang dicetak. Jika break tidak digunakan, program akan terus berlanjut hingga i melebihi 6. Secara keseluruhan, flowchart ini membantu memahami bagaimana kontrol aliran berfungsi dalam loop, serta dampak dari pernyataan kontrol seperti break dan continue dalam menentukan output program. Diagram ini juga menekankan pentingnya perancangan logika yang jelas agar pengulangan dapat berjalan sesuai dengan yang diharapkan.

v

Inisialisasi variabel i = 1

While i <= 6 ?

Print i

If i == 4 ?

**Yes No**

i++

Break

Back to while

1. Contoh 5

Analisis Diagram Flawchart:

Diagram tersebut menggambarkan penggunaan pengulangan dengan pernyataan while, di mana variabel count diinisialisasi dengan 0 dan kemudian diperiksa apakah nilainya kurang dari 20. Jika kondisi ini terpenuhi, program melanjutkan untuk memeriksa apakah count habis dibagi 5. Jika ya, nilai count dicetak. Jika tidak, count akan ditambah 1. Proses ini berulang sampai count mencapai 20. Diagram ini menunjukkan bagaimana operator modulus (%) berfungsi untuk mengidentifikasi kelipatan, serta bagaimana pengulangan dapat digunakan untuk mencetak hanya angka yang memenuhi syarat tertentu, yaitu angka yang habis dibagi 5.

Inisialisasi count = 0

While count < 20 ?

Yes

If count % 5 == 0 ?

Count ++

Print count

Back to while

1. Latihan 2.3

Analisis Diagram Flawchart:

Diagram ini berfokus pada penggunaan struktur do...while, dimulai dengan inisialisasi count ke 0. Dalam struktur ini, blok kode di dalam do akan selalu dieksekusi setidaknya sekali, sehingga nilai count dicetak sebelum memeriksa kondisi apakah count kurang dari 0. Meskipun pada iterasi pertama count adalah 0 (tidak memenuhi syarat untuk melanjutkan perulangan), nilai tersebut tetap dicetak. Diagram ini menggarisbawahi perbedaan penting antara do...while dan while, yaitu bahwa do...while menjamin eksekusi blok kode setidaknya satu kali. Secara keseluruhan, kedua diagram ini memberikan wawasan yang jelas tentang cara kerja pengulangan dan pengambilan keputusan dalam pemrograman, serta bagaimana struktur kontrol memengaruhi alur program dan output yang dihasilkan.

Inisialisasi count = 0

do

Print count

While count < 0 ?

**Yes No**

Back to do

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

Jawab:

Jika kita memahami struktur kontrol perulangan dengan baik, khususnya tentang penggunaan pernyataan while, do... while, break, dan continue, kita dapat menyelesaikan masalah ini. Perilaku setiap struktur kontrol berbeda tergantung pada kondisi yang ditetapkan; misalnya, break memungkinkan kita keluar dari loop sepenuhnya saat kondisi tertentu terpenuhi, sedangkan continue melewatkan iterasi tertentu dan melanjutkan ke iterasi berikutnya. Dengan memahami perbedaan ini, programmer dapat menulis kode dengan lebih efektif dan tepat sasaran. Mereka dapat memperbaiki kode dengan mengganti kondisi loop agar lebih sesuai dengan kebutuhan logika, menggunakan do... while untuk memastikan blok kode dieksekusi setidaknya sekali, dan memastikan penggunaan operator modulus dengan benar untuk memfilter dan memanipulasi angka, sehingga output program menjadi lebih relevan dan informatif.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Latihan 2.1

* Mulai
* Menganalisis program
* Mengubah //Ubah 1 menjadi continue;
* Running dan periksa hasilnya
* Menganalisa dampak perubahan terhadap luaran setelah running
* Menguraikan kegunaan break dan continue
* Selesai

1. Latihan 2.2

* Mulai
* Mengubah  //Ubah2 menjadi if (count % 5 == 0)
* Running dan periksa hasilnya
* Menganalisa dampaknya perubahan terhadap luaran setelah running
* Menguraikan kegunaan %.
* Selesai

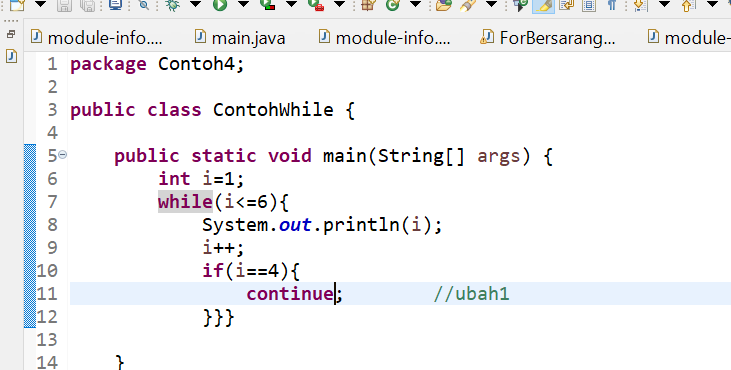
1. Latihan 2.3

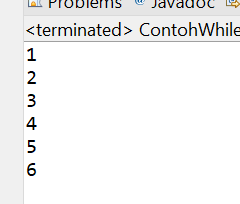
* Mulai
* Mengubah //Ubah1 menjadi while (count < 0) {
* Running dan periksa hasilnya
* Mengubah baris kode while pada Contoh 5 menjadi do … while dengan persyaratan yang sama  while (count < 0).
* Running dan periksa hasilnya
* Membandingkan hasil kedua luaran
* Selesai

1. Latihan 2.4
2. Kode program dan luaran
3. Screenshot/ Capture potongan kode dan hasil luaran

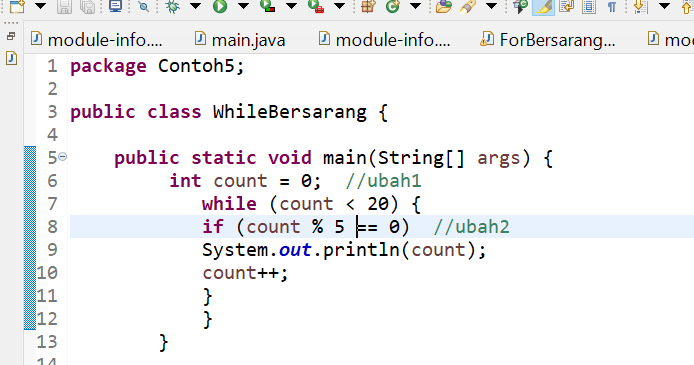
Beri komentar pada kode yang di Screenshot

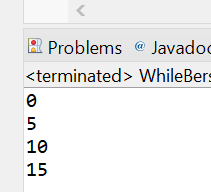
* Latihan 2.1



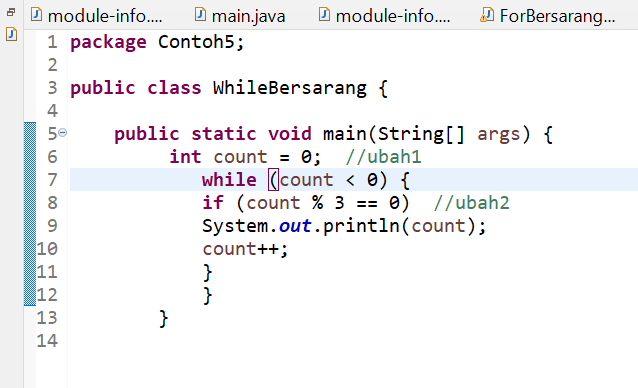


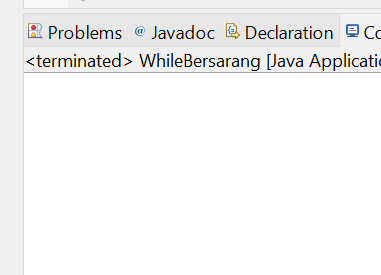
* Latihan 2.2

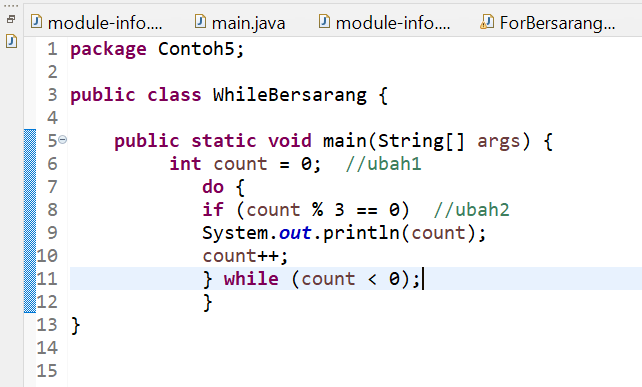


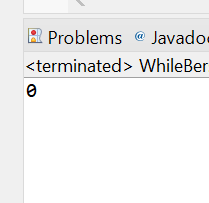


* Latihan 2.3









1. Analisa luaran yang dihasilkan   
   Jawab:

Luaran telah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan juga sudah sesuai dengan kebutuhan dan permintaan data.

**[No.2] Kesimpulan**

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Jawab:

Dari analisis, dapat disimpulkan bahwa dengan memahami struktur kontrol perulangan dan dampaknya, programmer dapat menulis kode yang lebih efektif dan efisien. Mengganti kondisi loop dan memanfaatkan break serta continue memungkinkan penyesuaian yang tepat terhadap output program sesuai kebutuhan.

1. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab:

Pengambilan keputusan dalam kasus ini didasarkan pada kebutuhan untuk meningkatkan kejelasan dan efisiensi dalam kode. Dengan memahami logika di balik pernyataan kontrol dan hasil dari setiap perubahan, programmer dapat mengoptimalkan kode untuk mencapai hasil yang diinginkan.

**Refleksi**

(Tuliskan singkat tentang pengalaman belajar, pemaknaan pengetahuan yang baru, tantangan yang dihadapi pada minggu tersebut. Ringkasan singkat dari semua soal, bukan per soal)

Jawab:

Saya belajar tentang konsep kontrol alur dalam pemrograman selama minggu ini, terutama pernyataan for dan while dan penggunaan break dan continue. Pengalaman ini sangat bermanfaat karena membantu saya memahami sintaks dan bagaimana perubahan kecil pada kode berdampak pada output. Meskipun saya menghadapi kesulitan membedakan penggunaan break dan continue, dengan adanya latihan ini membuat saya mulai memahami kapan masing-masing digunakan dengan benar. Selain itu, memiliki pemahaman tentang apa yang harus dilakukan sambil memberikan perspektif baru, terutama untuk menjamin eksekusi. Secara keseluruhan, latihan ini meningkatkan kemampuan saya dalam menganalisis kode dan menyesuaikan logika untuk mencapai hasil yang diinginkan dan mendorong saya untuk terus belajar konsep pemrograman yang lebih kompleks.