

Praktikum VII Dasar Pemograman



Disusun Oleh :

Muhammad Irsyad Dimas Abdillah

2341720088

Jl, Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65141

Phone : (0341) 404424, 404425

Percobaan 1








1. Source code

```
1  import java.util.Scanner;
2
3  public class ForKelipatan20 {
4      public static void main(String[] args) {
5          Scanner input = new Scanner(System.in);
6
7          System.out.print("Masukkan bilangan kelipatan (1-9): ");
8          int kelipatan = input.nextInt();
9          int total = 0;
10         int counter = 0;
11
12         for (int i = 1; i <= 50; i++) {
13             if (i % kelipatan == 0) {
14                 total += i;
15                 counter++;
16             }
17         }
18
19         System.out.println("Banyaknya bilangan " + kelipatan + " dari 1 sampai 50 adalah " + counter);
20         System.out.println("Total bilangan kelipatan " + kelipatan + " adalah " + total);
21     }
22 }
```

2. Output

```
PS C:\Users\62857\Downloads\POLINEMA\kuliah\Pengumpulan-Tugas>
rs\62857\AppData\Roaming\Code\User\workspaceStorage\d14a2dde5
Masukkan bilangan kelipatan (1-9): 5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan kelipatan 5 adalah 275
PS C:\Users\62857\Downloads\POLINEMA\kuliah\Pengumpulan-Tugas>
```

3. Push ke github

Pengumpulan-Tugas / Pertemuan 7 / 		
 Dimas0824 Tugas pertemuan ke 7 now 		
Name	Last commit message	Last commit date
 ..		
 DoWhileCuti20.java	Tugas pertemuan ke 7	now
 ForKelipatan20.java	Tugas pertemuan ke 7	now
 WhileGaji20.java	Tugas pertemuan ke 7	now

Pertanyaan

1. Terdapat tiga komponen perulangan pada sintaks FOR. Berdasarkan Percobaan 1 tersebut, sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program yang telah dibuat!

Jawab:

- a. `'int i =1;'` adalah inisialisasi, yang menetapkan nilai awal variabel iterasi `i` menjadi 1.
 - b. `i <= 50;` adalah kondisi, yang menentukan apakah loop harus terus berlanjut atau berhenti. Selama nilai `i` kurang dari atau sama dengan 50, loop akan terus berlanjut.
 - c. `i++;` adalah iterasi, yang meningkatkan nilai variabel iterasi `i` sebesar 1 setelah setiap iterasi.
2. Jelaskan alur kerja dari potongan kode program berikut!

Jawab:

Potongan kode program tersebut merupakan bagian dari struktur kontrol perulangan for dalam bahasa pemrograman Java. Berikut adalah penjelasan alur kerjanya:

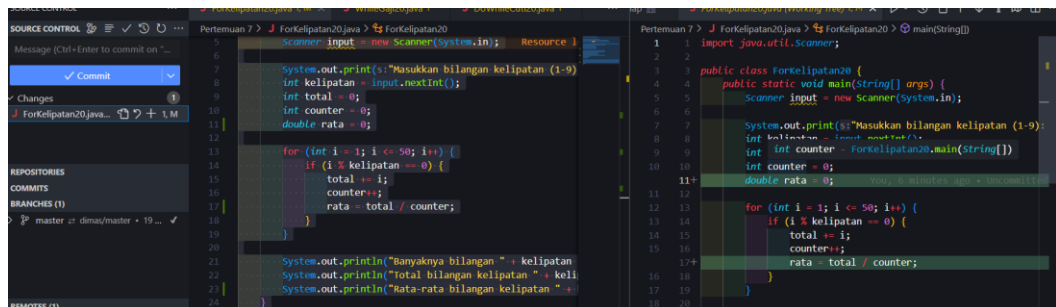
- a. Inisialisasi: Loop dimulai dengan nilai `i` adalah 1.
- b. Kondisi: Selama nilai `i` kurang dari atau sama dengan 50, loop akan terus berlanjut.
- c. Iterasi: Nilai `i` akan bertambah satu setelah setiap iterasi.
- d. Pernyataan dalam Loop: Dalam setiap iterasi, kode program akan memeriksa apakah `i` adalah kelipatan dari variabel kelipatan. Jika ya (`i % kelipatan == 0`), maka dua hal terjadi:
 - Nilai `i` ditambahkan ke variabel `total` (`total += i;`).
 - Variabel `counter` bertambah satu (`counter++;`).

Jadi, loop ini akan berjalan sebanyak 50 kali, dan dalam setiap iterasi, jika nilai `i` adalah kelipatan dari variabel kelipatan, maka nilai `i` akan ditambahkan ke `total`, dan `counter` akan bertambah satu. Pada akhirnya, variabel `total` akan berisi jumlah total dari semua bilangan yang merupakan kelipatan dari variabel kelipatan, dan variabel `counter` akan berisi banyaknya bilangan tersebut.

- a. Source code

b. Output

c. Push




4. Buatlah file baru dengan nama WhileKelipatanNoAbsen.java. Buatlah kode program dengan tujuan serupa tetapi menggunakan WHILE. Push dan commit kode program ke github.


Jawab:

```
1  import java.util.Scanner;
2
3  public class WhileKelipatan20 {
4      public static void main(String[] args) {
5          Scanner input = new Scanner(System.in);
6
7          System.out.print("Masukkan bilangan kelipatan (1-9): ");
8          int kelipatan = input.nextInt();
9          int total = 0;
10         int counter = 0;
11         double rata = 0;
12
13         int i = 1;
14         while (i <= 50) {
15             if (i % kelipatan == 0) {
16                 total += i;
17                 counter++;
18                 rata = total / counter;
19             }
20             i++;
21         }
22
23         System.out.println("Banyaknya bilangan " + kelipatan + " dari 1 sampai 50 adalah " + counter);
24         System.out.println("Total bilangan kelipatan " + kelipatan + " adalah " + total);
25         System.out.println("Rata-rata bilangan kelipatan " + kelipatan + " adalah " + rata);
26     }
27 }
28
```

push ke github

Pengumpulan-Tugas / Pertemuan 7 / WhileKelipatan20.java

 Dimas0824 pertanyaan 4 percobaan 1

Code Blame 27 lines (23 loc) · 847 Bytes  Code 55% faster with GitHub

```
1  import java.util.Scanner;
2
3  public class WhileKelipatan20 {
4      public static void main(String[] args) {
5          Scanner input = new Scanner(System.in);
6
7          System.out.print("Masukkan bilangan kelipatan (1-9): ");
8          int kelipatan = input.nextInt();
9          int total = 0;
10         int counter = 0;
11         double rata = 0;
```

Percobaan 2

1. Source code

```
1  import java.util.Scanner;
2
3  public class WhileGaji20 {
4      public static void main(String[] args) {
5          Scanner input = new Scanner(System.in);
6
7          System.out.print("Masukkan jumlah karyawan: ");
8          int jumlahKaryawan = input.nextInt();
9
10         int i = 0;
11         double totalGajiLembur = 0;
12
13         while (i < jumlahKaryawan) {
14             System.out.println("pilihan jabatan - Direktur, Manajer, Karyawan");
15             System.out.print("Masukkan jabatan karyawan ke-: " + (i + 1) + ": ");
16             String jabatan = input.next();
17             System.out.print("Masukkan jam lembur: ");
18             int jumlahJamLembur = input.nextInt();
19             i++;
20
21             int gajiLembur = 0;
22
23             if (jabatan.equalsIgnoreCase("direktur")) {
24                 continue;
25             } else if (jabatan.equalsIgnoreCase("manajer")) {
26                 gajiLembur = jumlahJamLembur * 100000;
27             } else if (jabatan.equalsIgnoreCase("karyawan")) {
28                 gajiLembur = jumlahJamLembur * 75000;
29             }
30             totalGajiLembur += gajiLembur;
31         }
32         System.out.println("Total gaji lembur: " + totalGajiLembur);
33     }
34 }
```

2. Output

```
99a3c227d\rednat.java\juc_ws\pengumpulan-tugas_e612b40c(bin)
Masukkan jumlah karyawan: 3
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 1: manajer
Masukkan jam lembur: 1
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 2: direktur
Masukkan jam lembur: 10
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 3: karyawan
Masukkan jam lembur: 5
Total gaji lembur: 475000.0
PS C:\Users\62857\Downloads\POLINEMA\kuliah\Pengumpulan-Tugas>
```

3. Push dan commit

Pengumpulan-Tugas / Pertemuan 7 /		
Dimas0824 Tugas pertemuan ke 7 now		
Name	Last commit message	Last commit date
..		
DoWhileCuti20.java	Tugas pertemuan ke 7	now
ForKelipatan20.java	Tugas pertemuan ke 7	now
WhileGaji20.java	Tugas pertemuan ke 7	now

Pertanyaan

1. Tunjukkan bagian kode program yang digunakan sebagai syarat untuk menghentikan perulangan WHILE! Berapa kali perulangan dilakukan?

Jawab: kode untuk menghentikan perulangan adalah $i < \text{jumlahKaryawan}$. Ini berarti bahwa selama nilai i kurang dari jumlahKaryawan, perulangan akan terus berlanjut. Jadi jika input jumlahKaryawan 5, maka perulangan akan dilakukan sampai 5 kali.

2. Pada potongan kode berikut, Apa yang sebenarnya terjadi jika variabel jabatan berisi nilai "DIREKTUR"? Apa peran CONTINUE yang dituliskan di dalam sintaks perulangan?

Jawab: fungsi continue dalam program ini adalah untuk melewati penghitungan gaji lembur untuk karyawan dengan jabatan "Direktur".

3. Mengapa komponen update $i++$ diletakkan di posisi tengah, tidak di bagian akhir statement? Pindahkan $i++$ di bagian akhir, lalu jalankan kembali program dengan memasukkan "direktur" sebagai jabatan karyawan pertama. Apa yang terjadi? Jelaskan!

Jawab: memindahkan $i++$ ke bagian akhir pernyataan akan menyebabkan loop tak terbatas.

```
Masukkan jumlah karyawan: 3
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 1: direktur
Masukkan jam lembur: 10
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 1: manajer
Masukkan jam lembur: 1
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 2: karyawan
Masukkan jam lembur: 5
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 3: manajer
Masukkan jam lembur: 10
Total gaji lembur: 1475000.0
PS C:\Users\62857\Downloads\POLINEMA\kuliah\Peng
```

4. Modifikasi kode program untuk handle jabatan yang invalid seperti contoh berikut:

Jawab: dengan menambahkan kode `i--;` pada jabatan tidak valid, yang bertujuan untuk mengulangi input user sampai benar.

```
1 import java.util.Scanner;
2
3 public class WhileGaji20 {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6
7         System.out.print("Masukkan jumlah karyawan: ");
8         int jumlahKaryawan = input.nextInt();
9
10        int i = 0;
11        double totalGajilembur = 0;
12        int gajilembur = 0;
13
14        while (i < jumlahKaryawan) {
15            System.out.println("pilihan jabatan - Direktur, Manajer, Karyawan");
16            System.out.print("Masukkan jabatan karyawan ke-: " + (i + 1) + ": ");
17            String jabatan = input.next();
18            System.out.print("Masukkan jam lembur: ");
19            int jumlahJamLembur = input.nextInt();
20            i++;
21
22            if (jabatan.equalsIgnoreCase("direktur")) {
23                continue;
24            } else if (jabatan.equalsIgnoreCase("manajer")) {
25                gajilembur = jumlahJamLembur * 100000;
26            } else if (jabatan.equalsIgnoreCase("karyawan")) {
27                gajilembur = jumlahJamLembur * 75000;
28            } else {
29                System.out.println("Jabatan tidak valid.");
30                i--; //mengurangi jabatan yg tidak valid
31                continue;
32            }
33            totalGajilembur += gajilembur;
34        }
35        System.out.println("Total gaji lembur: " + totalGajilembur);
36    }
37 }
```

```
aji20'
Masukkan jumlah karyawan: 3
Masukkan jumlah karyawan: 3
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 1: manajer
Masukkan jam lembur: 10
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 2: direktur
Masukkan jam lembur: 5
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 3: pegawai
Masukkan jam lembur: 4
Jabatan tidak valid.
pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-: 3: karyawan
Masukkan jam lembur: 4
Total gaji lembur: 1300000.0
PS C:\Users\62857\Downloads\POLINEMA\kuliah\Pengumpu
```

5. Push dan commit kode program ke github

Jawab:

```
← Files  master  Pengumpulan-Tugas / Pertemuan 7
/ WhileGaji20.java  ↑ Top

Code  Blame  Raw  Copy  Download  Edit  Close

3      public class WhileGaji20 {
11         double totalGajilembur = 0;
12         int gajilembur = 0;
13
14         while (i < jumlahKaryawan) {
15             System.out.println("pilihan jabatan - Direktur, Manajer, K");
16             System.out.print("Masukkan jabatan karyawan ke-: " + (i +
17             String jabatan = input.next();
18             System.out.print("Masukkan jam lembur: ");
19             int jumlahJamLembur = input.nextInt();
20             i++;
21
22             if (jabatan.equalsIgnoreCase("direktur")) {
23                 continue;
24             } else if (jabatan.equalsIgnoreCase("manajer")) {
25                 gajilembur = jumlahJamLembur * 100000;
26             } else if (jabatan.equalsIgnoreCase("karyawan")) {
27                 gajilembur = jumlahJamLembur * 75000;
28             } else {
29                 System.out.println("Jabatan tidak valid.");
30                 i--; //mengurangi jabatan yg tidak valid
31                 continue;
32             }
33             totalGajilembur += gajilembur;
34         }
35         System.out.println("Total gaji lembur: " + totalGajilembur);
36     }
```


Percobaan 3

1. Source code

```
1  import java.util.Scanner;
2
3  public class DoWhileCuti20 {
4      public static void main(String[] args){
5          Scanner sc = new Scanner(System.in);
6
7          // deklarasi
8          int jatahCuti, jumlahHari;
9          String konfirmasi;
10
11         //main
12         System.out.println("Jatah cuti: ");
13         jatahCuti = sc.nextInt();
14
15         do{
16             System.out.println("Apakah Anda ingin mengambil cuti (y/t)? ");
17             konfirmasi = sc.next();
18             if(konfirmasi.equalsIgnoreCase("y")){
19                 System.out.println("Jumlah hari: ");
20                 jumlahHari = sc.nextInt();
21
22                 if(jumlahHari <= jatahCuti){
23                     jatahCuti -= jumlahHari;
24                     System.out.println("Sisa jatah cuti: " + jatahCuti);
25                 }else{
26                     System.out.println("Sisa jatah cuti Anda tidak mencukupi");
27                     break;
28                 }
29             }
30         }while (jatahCuti > 0);
31     }
32 }
33
```

2. Output

```
-Tugas_e612b40c\bin' 'DowhileCuti20'
Jatah cuti: 12
Apakah Anda ingin mengambil cuti (y/t)? y
Jumlah hari: 4
Sisa jatah cuti: 8
Apakah Anda ingin mengambil cuti (y/t)? y
Jumlah hari: 5
Sisa jatah cuti: 3
Apakah Anda ingin mengambil cuti (y/t)? y
Jumlah hari: 4
Sisa jatah cuti Anda tidak mencukupi
PS C:\Users\62857\Downloads\POLINEMA\kuliah\P
```

3. Push and commit

Pengumpulan-Tugas / Pertemuan 7 /		
Dimas0824 Tugas pertemuan ke 7 now		
Name	Last commit message	Last commit date
..		
DoWhileCuti20.java	Tugas pertemuan ke 7	now
ForKelipatan20.java	Tugas pertemuan ke 7	now
WhileGaji20.java	Tugas pertemuan ke 7	now

Pertanyaan

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

Jawab: Sintaks BREAK berguna untuk menghentikan eksekusi dari perulangan.

Selanjutnya program akan mengeksekusi kode setelah blok perulangan.

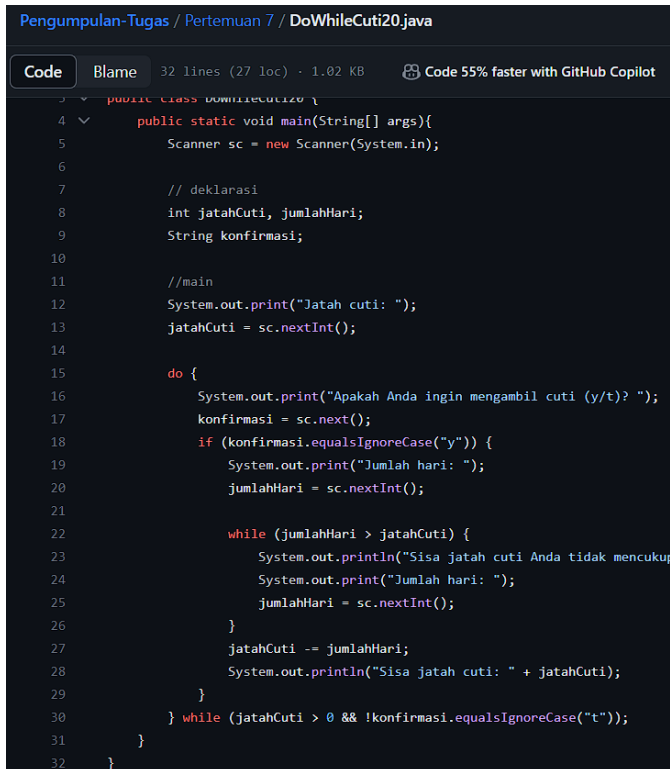
2. Modifikasi kode program sehingga jika jumlah hari cuti yang ingin diambil lebih besar daripada jatah yang tersisa, program tidak berhenti sehingga pengguna masih memiliki kesempatan untuk mengisi jumlah hari sesuai jatah cuti.

Jawab:

```
1  import java.util.Scanner;
2
3  public class DowhileCuti20 {
4      public static void main(String[] args){
5          Scanner sc = new Scanner(System.in);
6
7          // deklarasi
8          int jatahCuti, jumlahHari;
9          String konfirmasi;
10
11         //main
12         System.out.print("Jatah cuti: ");
13         jatahCuti = sc.nextInt();
14
15         do {
16             System.out.print("Apakah Anda ingin mengambil cuti (y/t)? ");
17             konfirmasi = sc.next();
18             if (konfirmasi.equalsIgnoreCase("y")) {
19                 System.out.print("Jumlah hari: ");
20                 jumlahHari = sc.nextInt();
21
22                 while (jumlahHari > jatahCuti) {
23                     System.out.println("Sisa jatah cuti Anda tidak mencukupi");
24                     System.out.print("Jumlah hari: ");
25                     jumlahHari = sc.nextInt();
26                 }
27                 jatahCuti -= jumlahHari;
28                 System.out.println("Sisa jatah cuti: " + jatahCuti);
29             }
30         } while (jatahCuti > 0 && !konfirmasi.equalsIgnoreCase("t"));
31     }
32 }
33
```

3. Push dan commit kode program ke github

Jawab:



```
Pengumpulan-Tugas / Pertemuan 7 / DoWhileCuti20.java
Code Blame 32 lines (27 loc) · 1.02 KB Code 55% faster with GitHub Copilot

1 public class DoWhileCuti20 {
2
3     public static void main(String[] args){
4         Scanner sc = new Scanner(System.in);
5
6         // deklarasi
7         int jatahCuti, jumlahHari;
8         String konfirmasi;
9
10        //main
11        System.out.print("Jatah cuti: ");
12        jatahCuti = sc.nextInt();
13
14        do {
15            System.out.print("Apakah Anda ingin mengambil cuti (y/t)? ");
16            konfirmasi = sc.next();
17            if (konfirmasi.equalsIgnoreCase("y")) {
18                System.out.print("Jumlah hari: ");
19                jumlahHari = sc.nextInt();
20
21                while (jumlahHari > jatahCuti) {
22                    System.out.println("Sisa jatah cuti Anda tidak mencukupi");
23                    System.out.print("Jumlah hari: ");
24                    jumlahHari = sc.nextInt();
25                }
26                jatahCuti -= jumlahHari;
27                System.out.println("Sisa jatah cuti: " + jatahCuti);
28            } while (jatahCuti > 0 && !konfirmasi.equalsIgnoreCase("t"));
29        }
30    }
31 }
32 }
```

4. Pada saat input konfirmasi, ketikkan “t”, apa yang terjadi? Mengapa demikian?

Jawab: kode program akan dihentikan sebelum memasuki blok perulangan karena input equals y. Dengan demikian ada 2 kondisi yang harus terpenuhi untuk melakukan perulangan yaitu ketika $jatahcute > 0$, dan input konfirmasi tidak sama dengan t, maka perulangan akan dilakukan.

5. Modifikasi kode program sehingga saat pengguna mengetikkan “t” sebagai input konfirmasi, maka program akan berhenti.



```
1 import java.util.Scanner;
2
3 public class DoWhileCuti20 {
4     public static void main(String[] args){
5         Scanner sc = new Scanner(System.in);
6
7         // deklarasi
8         int jatahCuti, jumlahHari;
9         String konfirmasi;
10
11        //main
12        System.out.print("Jatah cuti: ");
13        jatahCuti = sc.nextInt();
14
15        do {
16            System.out.print("Apakah Anda ingin mengambil cuti (y/t)? ");
17            konfirmasi = sc.next();
18            if (konfirmasi.equalsIgnoreCase("y")) {
19                System.out.print("Jumlah hari: ");
20                jumlahHari = sc.nextInt();
21
22                while (jumlahHari > jatahCuti) {
23                    System.out.println("Sisa jatah cuti Anda tidak mencukupi");
24                    System.out.print("Jumlah hari: ");
25                    jumlahHari = sc.nextInt();
26                }
27                jatahCuti -= jumlahHari;
28                System.out.println("Sisa jatah cuti: " + jatahCuti);
29            } while (jatahCuti > 0 && !konfirmasi.equalsIgnoreCase("T"));
30        }
31    }
32 }
33 }
```

6. Push dan commit kode program ke github

Pengumpulan-Tugas / Pertemuan 7 // DoWhileCuti20.java

Code Blame 32 lines (27 loc) · 1.02 KB Code 55% faster with GitHub Copilot

```
1  public class DoWhileCuti20 {
2
3  }
4  public static void main(String[] args){
5      Scanner sc = new Scanner(System.in);
6
7      // deklarasi
8      int jatahCuti, jumlahHari;
9      String konfirmasi;
10
11     //main
12     System.out.print("Jatah cuti: ");
13     jatahCuti = sc.nextInt();
14
15     do {
16         System.out.print("Apakah Anda ingin mengambil cuti (y/t)? ");
17         konfirmasi = sc.next();
18         if (konfirmasi.equalsIgnoreCase("y")) {
19             System.out.print("Jumlah hari: ");
20             jumlahHari = sc.nextInt();
21
22             while (jumlahHari > jatahCuti) {
23                 System.out.println("Sisa jatah cuti Anda tidak mencukupi");
24                 System.out.print("Jumlah hari: ");
25                 jumlahHari = sc.nextInt();
26             }
27             jatahCuti -= jumlahHari;
28             System.out.println("Sisa jatah cuti: " + jatahCuti);
29         }
30     } while (jatahCuti > 0 && !konfirmasi.equalsIgnoreCase("t"));
31 }
32 }
```