

LAPORAN PRAKTIKUM 7

DASAR PEMROGRAMAN



Rangga Dwi Saputra

2341720248

Kelas 1B

Prodi D-IV Teknik Informatika

Jobsheet 7

Perulangan 1

A. Percobaan 1: Menghitung Bilangan Kelipatan Menggunakan FOR

Pada percobaan ini dilakukan pembuatan kode program untuk menampilkan bilangan kelipatan angka tertentu dari rentang 1 sampai dengan 50 menggunakan perulangan FOR, serta menghitung total dari bilangan-bilangan tersebut.

1. Buat file baru di Visual Studio Code dengan nama file **ForKelipatanNoAbsen.java**
2. Masukkan fungsi main() sebagai struktur dasar
3. Tambahkan library Scanner
4. Buat deklarasi Scanner dengan nama scan
5. Buatlah variabel bertipe int dengan nama **kelipatan, jumlah, dan counter**.
Inisialisasi variabel **jumlah=0** dan **counter=0**
6. Tambahkan kode untuk menerima input dari keyboard
7. Struktur perulangan FOR dengan kondisi pemilihan IF untuk menentukan bilangan kelipatan
8. Tampilkan banyaknya bilangan kelipatan dan total bilangan kelipatan pada rentang 1 sampai dengan 50.
9. Jalankan program tersebut. Cocokkan hasil compile kode program Anda dengan gambar berikut ini

Masukkan bilangan kelipatan (1-9): 5

Banyaknya bilangan 5 dari 1 sampai 50 adalah 10

Total bilangan kelipatan 5 dari 1 sampai 50 adalah 275

10. Berdasarkan program didapatkan hasil sebagai berikut;

```
import java.util.Scanner;
public class ForKelipatan23 {
    Run | Debug
    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        int kelipatan, jumlah=0, counter=0;

        System.out.println(x:"Masukkan bilangan kelipatan (1-9) : ");
        kelipatan = scan.nextInt();

        // for (inisialisasi; kondisi; update) { statement1; statement2;}
        for ( int i = 1; i <=50; i++) {
            if (i % kelipatan == 0) {
                jumlah += i; //menghitung akumulasi dari banyaknya nilai
                counter++; //menghitung banyaknya nilai
            }
        }

        System.out.printf(format:"Banyaknya bilangan %d dari 1 sampai 50 adalah %d", kelipatan, counter);
        System.out.printf(format:"\nTotal bilangan kelipatan %d dari 1 sampai 50 adalah %d", kelipatan, jumlah);
    }
}
```

Output: sama dengan yang diminta

```
(rednat.java\jdk_ws\Pertemuan7_95551798\bin Forkelipatan23
Masukkan bilangan kelipatan (1-9) :
5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan kelipatan 5 dari 1 sampai 50 adalah 275
PS D:\Semester 1\Dasar Pemrograman\Pertemuan7>
```

11. Commit dan push kode program ke github



```
1  import java.util.Scanner;
2  public class Forkelipatan23 {
3      public static void main(String[] args) {
4
5          Scanner scan = new Scanner(System.in);
6          int kelipatan, jumlah=0, counter=0;
7
8          System.out.println("Masukkan bilangan kelipatan (1-9) : ");
9          kelipatan = scan.nextInt();
10
11         // for (inisialisasi; kondisi; update) { statement1; statement2;}
12         for ( int i = 1; i <=50; i++) {
13             if (i % kelipatan == 0) {
14                 jumlah += i; //menghitung akumulasi dari banyaknya nilai
15                 counter++; //menghitung banyaknya nilai
16             }
17         }
18         System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d", kelipatan, counter);
19         System.out.printf("\ntotal bilangan kelipatan %d dari 1 sampai 50 adalah %d", kelipatan, jumlah);
20     }
21 }
```

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!



```
// for (inisialisasi; kondisi; update) { statement1; statement2;}
for ( int i = 1; i <=50; i++) {
    if (i % kelipatan == 0) {
        jumlah += i;
        counter++;
    }
}
```

2. Jelaskan alur kerja dari potongan kode program berikut!

```
for (int i = 1; i <= 50; i++) {
    if (i % kelipatan == 0) {
        total += i;
        counter++;
    }
}
```

- a) Perulangan diawali dengan melakukan inisialisasi **i=1**
- b) Tentukan kondisi **i <= 50**. Kemudian Evaluasi kondisi :
 - Kondisi **i** akan terus berlanjut selama kelipatannya tidak melebihi 50.

- Counter++ akan menghitung banyaknya nilai dari kelipatan yang dihasilkan
- Total += akan mengakumulasi nilai dari banyaknya kelipatan yang dihasilkan. akan melakukan update. Ulangi kembali langkah nomor 2

3. Modifikasi kode program yang telah dibuat dengan menambahkan variabel baru untuk menghitung rata-rata dari seluruh bilangan kelipatan yang ditentukan! Push dan commit kode program ke github.

```
import java.util.Scanner;
public class ForKelipatan23 {
    Run | Debug
    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        int kelipatan, jumlah;
        double average=0;

        System.out.println("Masukkan bilangan kelipatan (1-9) : ");
        kelipatan = scan.nextInt();

        // for (inisialisasi; kondisi; update) { statement1; statement2;}
        for (int i = 1; i <= 50; i++) {
            if (i % kelipatan == 0) {
                jumlah += i; //menghitung akumulasi dari banyaknya nilai
                counter++; //menghitung banyaknya nilai
                average = jumlah / counter; //menghitung rata-rata
            }
        }

        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d", kelipatan, counter);
        System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d", kelipatan, jumlah);
        System.out.printf("Jumlah rata-rata adalah: %f", average);
    }
}
```

Output:

```
Masukkan bilangan kelipatan (1-9) :
5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan kelipatan 5 dari 1 sampai 50 adalah 275
Jumlah rata-rata adalah: 27.000000
PS D:\Semester 1\Dasar Pemrograman\Pertemuan7>
```

Push dan Commit ke Github:

```
Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git init
Reinitialized existing Git repository in D:/Semester 1/Dasar Pemrograman/Pertemuan7/.git/

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git add ForKelipatan23.java

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git commit -m "modifikasi + average"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git branch -M main

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git remote add origin https://github.com/RanggaDwiSaputra/ForKelipatan23.git
error: remote origin already exists.

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git push -u origin main
Everything up-to-date
branch 'main' set up to track 'origin/main'

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$
```

```
import java.util.Scanner;
public class ForKelipatan23 {
    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        int kelipatan, jumlah=0, counter=0;
        double average=0;

        System.out.println("Masukkan bilangan kelipatan (1-9) : ");
        kelipatan = scan.nextInt();

        // for (inisialisasi; kondisi; update) { statement1; statement2;}
        for (int i = 1; i <= 50; i++) {
            if (i % kelipatan == 0) {
                jumlah += i; //menghitung akumulasi dari banyaknya nilai
                counter++; //menghitung banyaknya nilai
                average = jumlah / counter; //menghitung rata-rata
            }
        }

        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d", kelipatan, counter);
        System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d", kelipatan, jumlah);
        System.out.printf("Jumlah rata-rata adalah: %f", average);
    }
}
```

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

```
import java.util.Scanner;

public class WhileKelipatan23 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int kelipatan, jumlah=0, counter=0;

        System.out.println("Masukkan bilangan kelipatan (1-9) : ");
        kelipatan = scan.nextInt();

        /*inisiasi
        while (kondisi) {
            statement;
        }
        update; */
        int i = 1;
        while (i <= 50) {
            if (i % kelipatan == 0) {
                jumlah += i;
                counter++;
            }
            i++;
        }
        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d", kelipatan, counter);
        System.out.printf("\nTotal bilangan kelipatan %d dari 1 sampai 50 adalah %d", kelipatan, jumlah);
    }
}
```

Output

```
Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git init
Reinitialized existing Git repository in D:/Semester 1/Dasar Pemrograman/Pertemuan7/.git/

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git add WhileKelipatan23.java

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git commit -m "add while"
[main 707cd0e] add while
1 file changed, 30 insertions(+)
create mode 100644 WhileKelipatan23.java

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git branch -M main

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git remote add origin https://github.com/PutraRanggaDwi/WhileKelipatan23.git
error: remote origin already exists!
```

```
Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4)
Delta compression using up to 4 threads
Compressing objects: 100% (3/3)
Writing objects: 100% (3/3)
Total 3 (delta 1), reused 0 (delta 0), pushed 3 (delta 0)
remote: Resolving deltas: 0% (0/1)
To https://github.com/PutraRanggaDwi/WhileKelipatan23.git
fd48a64..707cd0e main -> main
branch 'main' set up to track remote branch 'main' from 'origin'

Rangga Dwi Saputra@LAPTOP-36IOUL75 MINGW64 /d/Semester 1/Dasar Pemrograman/Pertemuan7 (main)
$
```

```
public class WhileKelipatan23 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int kelipatan, jumlah=0, counter=0;

        System.out.println("Masukkan bilangan kelipatan (1-9) : ");
        kelipatan = scan.nextInt();

        /*inisiasi
        while (kondisi) {
            statement;
        }
        update; */
        int i = 1;
        while (i <= 50) {
            if (i % kelipatan == 0) {
                jumlah += i;
                counter++;
            }
            i++;
        }
        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d", kelipatan, counter);
        System.out.printf("\nTotal bilangan kelipatan %d dari 1 sampai 50 adalah %d", kelipatan, jumlah);
    }
}
```

B. Percobaan 2: Menghitung Gaji Lembur Karyawan Menggunakan WHILE dan CONTINUE

Sebuah perusahaan memberikan gaji lembur kepada karyawannya setiap minggu. Gaji tersebut dihitung berdasarkan jabatan karyawan dan jumlah jam lembur dalam seminggu. Karyawan dengan jabatan “direktur” tidak mendapatkan tambahan gaji meskipun melakukan lembur, karyawan dengan jabatan “manager” mendapatkan gaji lembur sebesar 100000 per jam, sedangkan karyawan dengan jabatan “staf” mendapatkan gaji lembur sebesar 75000 per jam. Pada percobaan ini dilakukan pembuatan kode program menggunakan WHILE dan CONTINUE untuk menghitung pengeluaran perusahaan.

1. Buat file baru di Visual Studio Code dengan nama file **ForKelipatanNoAbsen.java**
2. Masukkan fungsi main() sebagai struktur dasar
3. Tambahkan library Scanner
4. Buat deklarasi Scanner dengan nama scan
5. variabel **jumlahKaryawan** dan **jumlahJamLembur** bertipe int serta **gajiLembur** dan **totalGajiLembur** bertipe double.
6. Inisialisasi variabel **gajiLembur=0** dan **totalGajiLembur=0**
7. Deklarasikan variabel **jabatan** bertipe String
8. Tambahkan kode untuk menerima input dari keyboard guna menentukan jumlah karyawan yang akan dihitung gajinya
9. Buat struktur perulangan **WHILE** dengan kondisi pemilihan **IF-ELSE** dan **CONTINUE** untuk menentukan gaji lembur berdasarkan jabatan karyawan
10. Tampilkan hasil perhitungan jumlah gaji

```
Users\Rangga Dwi Saputra\AppData\Roaming\Code\User\workspace
WhileGaji23'
Masukkan Jumlah Karyawan:
3
Pilihan Jabatan - Direktur, Manager, Karyawan
Masukkan jabatan Karyawan ke-1 :
manager
Masukkan Jumlah jam Lembur
1
Pilihan Jabatan - Direktur, Manager, Karyawan
Masukkan jabatan Karyawan ke-2 :
direktur
Masukkan Jumlah jam Lembur
10
Pilihan Jabatan - Direktur, Manager, Karyawan
Masukkan jabatan Karyawan ke-3 :
karyawan
Masukkan Jumlah jam Lembur
5
Total Gaji Lembur: 475000.0
PS D:\Semester 1\Dasar Pemrograman\Pertemuan7> |
```

11. Berikut program dari output yang dihasilkan:

```
1  import java.util.Scanner;
2  public class WhileGaji23 {
3      public static void main(String[] args) {
4          Scanner scan = new Scanner(System.in);
5          int jumlahKaryawan, jumlahJamLembur;
6          double gajiLembur=0, totalGajiLembur=0;
7          String jabatan;
8
9          System.out.println("Masukkan Jumlah Karyawan: ");
10         jumlahKaryawan = scan.nextInt();
11
12         int i = 0;
13         while (i < jumlahKaryawan) {
14             System.out.println("Pilihan Jabatan - Direktur, Manager, Karyawan");
15             System.out.println("Masukkan jabatan Karyawan ke-" + (i+1) + " : ");
16             jabatan = scan.next();
17             System.out.println("Masukkan Jumlah jam Lembur");
18             jumlahJamLembur = scan.nextInt();
19             i++;
20
21             if (jabatan.equalsIgnoreCase("direktur")) {
22                 continue;
23             } else if (jabatan.equalsIgnoreCase("Manager")) {
24                 gajiLembur = jumlahJamLembur * 100000;
25             } else if (jabatan.equalsIgnoreCase("Karyawan")) {
26                 gajiLembur = jumlahJamLembur * 75000;
27             }
28             totalGajiLembur += gajiLembur;
29         }
30         System.out.println("Total Gaji Lembur: " +totalGajiLembur);
31     }
32 }
33
34
```

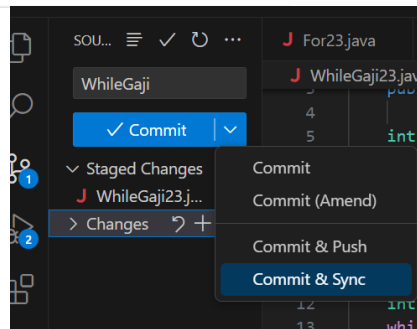
12. Push dan Commit ke GitHub

```
import java.util.Scanner;
public class WhileGaji23 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int jumlahKaryawan, jumlahJamLembur;
        double gajiLembur=0, totalGajiLembur=0;
        String jabatan;

        System.out.println("Masukkan Jumlah Karyawan: ");
        jumlahKaryawan = scan.nextInt();

        int i = 0;
        while (i < jumlahKaryawan) {
            System.out.println("Pilihan Jabatan - Direktur, Manager, Karyawan");
            System.out.println("Masukkan jabatan Karyawan ke-" + (i+1) + " : ");
            jabatan = scan.next();
            System.out.println("Masukkan Jumlah jam Lembur");
            jumlahJamLembur = scan.nextInt();
            i++;

            if (jabatan.equalsIgnoreCase("direktur")) {
                continue;
            } else if (jabatan.equalsIgnoreCase("Manager")) {
                gajiLembur = jumlahJamLembur * 100000;
            } else if (jabatan.equalsIgnoreCase("Karyawan")) {
                gajiLembur = jumlahJamLembur * 75000;
            }
        }
    }
}
```



Pertanyaan

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

```
6 double gajiLembur=0, totalGajiLembur=0;
7 String jabatan;
8
9 System.out.println("Masukkan Jumlah Karyawan:");
10 jumlahKaryawan = scan.nextInt();
11
12 int i = 0;
13 while (i < jumlahKaryawan) {
14     System.out.println("Pilihan Jabatan - Direktur, Manager, Karyawan");
15     System.out.println("Masukkan jabatan Karyawan ke-" + (i+1) + " : ");
16     jabatan = scan.next();
17     System.out.println("Masukkan Jumlah jam Lembur");
18     jumlahJamLembur = scan.nextInt();
19     i++;
20
21     if (jabatan.equalsIgnoreCase("direktur")) {
22         continue;
23     } else if (jabatan.equalsIgnoreCase("Manager")) {
24         gajiLembur = jumlahJamLembur * 100000;
25     } else if (jabatan.equalsIgnoreCase("Karyawan")) {
26         gajiLembur = jumlahJamLembur * 75000;
27     }
28     totalGajiLembur += gajiLembur;
29 }
```

Syarat

Perulangan

Syarat untuk menghentikan perulangan while ditentukan pada inputan berapa jumlah karyawan. Program akan tetap berajalan sampai pada jumlah yang sudah diinputkan

```
Masukkan Jumlah Karyawan:
2
Pilihan Jabatan - Direktur, Manager, Karyawan
Masukkan jabatan Karyawan ke-1 :
manager
Masukkan Jumlah jam Lembur
5
Pilihan Jabatan - Direktur, Manager, Karyawan
Masukkan jabatan Karyawan ke-2 :
karyawan
Masukkan Jumlah jam Lembur
4
Total Gaji Lembur: 800000.0
PS D:\Semester 1\Dasar Pemrograman\Pertemuan7>
```

Pada inputan berjumlah 2, maka program akan berulang sebanyak 2 kali saja. Setelah itu menjalankan program penutup.

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

Pada studi kasus disebutkan bahwa direktur tidak mendapatkan gaji tambahan, fungsi "continue" adalah melanjutkan program selanjutnya meskipun keluar dari ketentuan. Pada kondisi if memakai fungsi

`equalsIgnoreCase`, jadi pemberian nama direktur/DIREKTUR akan sama saja karena penggunaan huruf kapital/tidak akan diabaikan

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!

```
25 } else if (jabatan.equalsIgnoreCase(anotherString: "Manager")) {
26     gajiLembur = jumlahJamLembur * 75000;
27 }
28 i++;
29 totalGajiLembur += gajiLembur;
30 }
31
32 System.out.println("Total Gaji Lembur: " + totalGajiLembur);
33 }
34 }
35 }
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Pilihan Jabatan - Direktur, Manager, Karyawan
Masukkan jabatan Karyawan ke-1 :
direktur
Masukkan Jumlah jam Lembur
4
Pilihan Jabatan - Direktur, Manager, Karyawan
Masukkan jabatan Karyawan ke-1 :
Karyawan
Masukkan Jumlah jam Lembur
8
Pilihan Jabatan - Direktur, Manager, Karyawan
Masukkan jabatan Karyawan ke-2 :

komponen `i++` diletakkan ditengah untuk agar perhitungan dimulai ketika program mulai dijalankan, jika diletakkan di akhir maka perhitungan ke-1 akan dibaca 2 kali.

4. Modifikasi kode program untuk menghandle jabatan yang invalid:

```
22 continue;
23 } else if (jabatan.equalsIgnoreCase(anotherString: "Manager")) {
24     gajiLembur = jumlahJamLembur * 100000;
25 } else if (jabatan.equalsIgnoreCase(anotherString: "Karyawan")) {
26     gajiLembur = jumlahJamLembur * 75000;
27 } else { System.out.println(x: "Jabatan Invalid");
28     i--;
29     continue;}
30 totalGajiLembur += gajiLembur;
31 }
32
33 System.out.println("Total Gaji Lembur: " + totalGajiLembur);
34 }
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

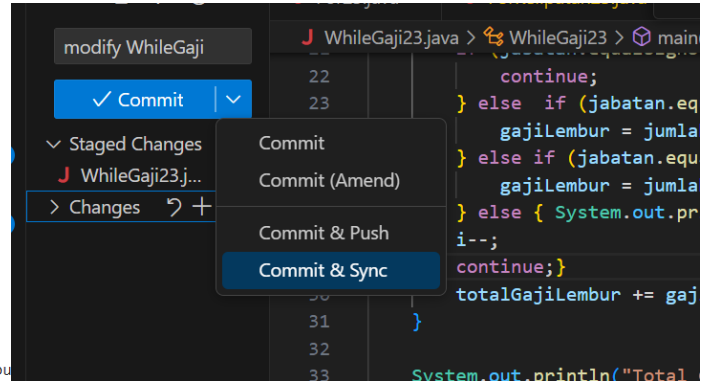
Masukkan jabatan Karyawan ke-3 :
pegawai
Masukkan Jumlah jam Lembur
4
Jabatan Invalid
Pilihan Jabatan - Direktur, Manager, Karyawan
Masukkan jabatan Karyawan ke-3 :
Karyawan
Masukkan Jumlah jam Lembur
5
Total Gaji Lembur: 1375000.0
PS D:\Semester 1\Dasar Pemrograman\Pertemuan7> █

```

i++;
if (jabatan.equalsIgnoreCase("direktur")) {
    continue;
} else if (jabatan.equalsIgnoreCase("Manager")) {
    gajiLembur = jumlahJamLembur * 100000;
} else if (jabatan.equalsIgnoreCase("Karyawan")) {
    gajiLembur = jumlahJamLembur * 75000;
} else { System.out.println("Jabatan Invalid");
i--;
continue;}
totalGajiLembur += gajiLembur;
}

System.out.println("Total Gaji Lembur: " +totalGajiLembur);
}

```



C. Percobaan 3: Menghitung Jatah Cuti Menggunakan DO-WHILE

1. Buat file di Visual Studio Code dengan nama **DoWhile23.java**
2. Buat struktur dasar fungsi main().
3. Masukkan library Scanner
4. Buat variable **jatahCuti** dan **jumlahHari** menggunakan tipe data int
5. Buat variable **konfirmasi** menggunakan tipe data String
6. Masukkan kode Scanner untuk menginput nilai jatahCuti
7. Buat struktur data **Do-While** sebagai berikut

```

import java.util.Scanner;
public class DowhileCuti23 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int jatahCuti, jumlahHari;
        String konfirmasi;

        System.out.println(x:"Jatah Cuti: ");
        jatahCuti = sc.nextInt();

        do {
            System.out.println(x:"Apakah anda ingin mengambil cuti? (y/n)");
            konfirmasi = sc.next();

            if (konfirmasi.equalsIgnoreCase(anotherString:"y")) {
                System.out.println(x:"Jumlah Hari: ");
                jumlahHari = sc.nextInt();

                if (jumlahHari < jatahCuti) {
                    jatahCuti -= jumlahHari;
                    System.out.println("Sisa jatah cuti: " +jatahCuti);
                } else {
                    System.out.println(x:"Sisa jatah cuti anda tidak mencukupi");
                    break;
                }
            }
        } while (jatahCuti > 0);
    }
}

```

8. Akan menghasilkan output sebagai berikut:

```

at DownWhileCuti23.main(DownWhileCuti23.java:9)
PS D:\Semester 1\Dasar Pemrograman\Pertemuan7> d:; cd 'd:\va\jdk-17\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages\workspaceStorage\451a33cf8e4ca1a75666996c0167d8c3\redha
Jatah Cuti:
12
Apakah anda ingin mengambil cuti? (y/n)
y
Jumlah Hari:
4
Sisa jatah cuti: 8
Apakah anda ingin mengambil cuti? (y/n)
y
Jumlah Hari:
5
Sisa jatah cuti: 3
Apakah anda ingin mengambil cuti? (y/n)
y
Jumlah Hari:
4
Sisa jatah cuti anda tidak mencukupi
PS D:\Semester 1\Dasar Pemrograman\Pertemuan7>

```

```

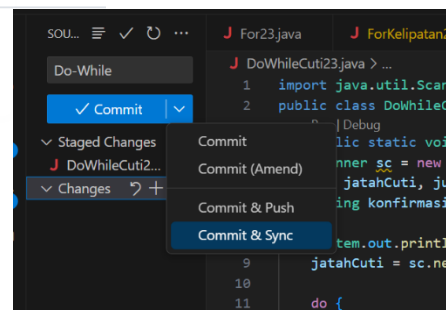
public class DoWhileCuti23 {
    System.out.println("Jatah Cuti: ");
    jatahCuti = sc.nextInt();

    do {
        System.out.println("Apakah anda ingin mengambil cuti? (y/n)");
        konfirmasi = sc.next();

        if (konfirmasi.equalsIgnoreCase("y")) {
            System.out.println("Jumlah Hari: ");
            jumlahHari = sc.nextInt();

            if (jumlahHari < jatahCuti) {
                jatahCuti -= jumlahHari;
                System.out.println("Sisa jatah cuti: " + jatahCuti);
            } else {
                System.out.println("Sisa jatah cuti anda tidak mencukupi");
                break;
            }
        }
    } while (jatahCuti > 0);
}

```



Pertanyaan

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?
Sintaks 'break' digunakan dalam perulangan dalam bahasa pemrograman untuk menghentikan perulangan secara paksa dan keluar dari perulangan saat kondisi tertentu terpenuhi
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.

```

20         jatahcuti -= jumlahHari;
21         System.out.println("Sisa jatahcuti: " + jatahcuti);
22     } else {
23         System.out.println("Sisa jatahcuti anda tidak mencukupi");
24         continue;
25     }
26 }
27 while (jatahcuti > 0);
28 }

```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORT

```

Sisa jatahcuti anda tidak mencukupi > d;;
va\jdk-17\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages
ser\workspaceStorage\451a33cf8e4ca1a75666996c0167d8c
Jatahcuti:
12
Apakah anda ingin mengambil cuti? (y/n)
y
Jumlah Hari:
15
Sisa jatahcuti anda tidak mencukupi
Apakah anda ingin mengambil cuti? (y/n)
y
Jumlah Hari:
7
Sisa jatahcuti: 5
Apakah anda ingin mengambil cuti? (y/n)

```

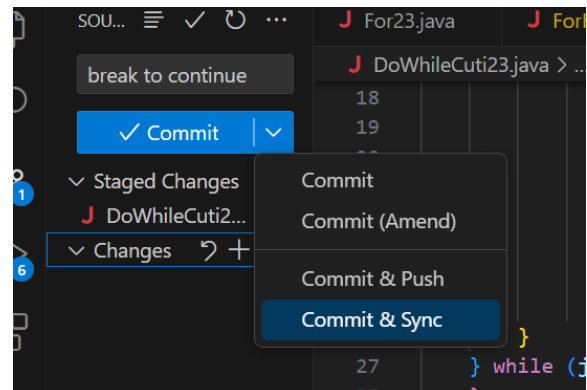
kode 'break' diganti dengan 'continue'.

3. Push dan commit kode program ke github

```

if (jumlahHari < jatahcuti) {
    jatahcuti -= jumlahHari;
    System.out.println("Sisa jatahcuti: " + jatahcuti);
} else {
    System.out.println("Sisa jatahcuti anda tidak mencukupi");
    continue;
}
} while (jatahcuti > 0);
}

```



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

Program akan terus mengulangi pertanyaan, karena program tidak ada kondisi yang menjalankan inputan "t"

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

```
22     } else {
23         System.out.println(x:"Sisa jatah cuti anda tidak mencukupi");
24         continue;
25     }
26 }else if (konfirmasi.equalsIgnoreCase(anotherString:"t")){ System.out.println(x:"Program dihentikan");
27     break; }
28 } while (jatahCuti > 0 );
29 }
```

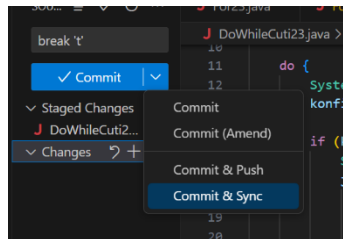
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Semester 1\Dasar Pemrograman\Pertemuan7> & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Rangga Dwi Saputra\AppData\Roaming\Code\User\workspaceStorage\451a33cf8e4ca1a75666996c0167d8\redhat.java\jdt_ws\Pertemuan7_95551790\bin' 'DoWhileCuti23'

Jatah Cuti:
12
Apakah anda ingin mengambil cuti? (y/n)
t
Program dihentikan

6. Push dan commit kode program ke github

```
        continue;
    }
    }else if (konfirmasi.equalsIgnoreCase("t")){ System.out.println("Program dihentikan");
    break; }
} while (jatahCuti > 0 );
}
```



TUGAS KELOMPOK

PENERAPAN SISTEM PERULANGAN KEDALAM PROJECT

1. Fitur Login

```
1  import java.util.Scanner;
2
3  public class login11 {
4
5      public static void main(String[] args) {
6          String username = "user";
7          String password = "password";
8
9          Scanner scanner = new Scanner(System.in);
10         boolean login = false;
11
12         System.out.println("*****LOGIN*****");
13
14         while (!login) {
15             System.out.print("Masukkan username: ");
16             String inputUsername = scanner.nextLine();
17
18             if (inputUsername.equals(username)) {
19                 System.out.print("Masukkan password: ");
20                 String inputPassword = scanner.nextLine();
21
22                 if (inputPassword.equals(password)) {
23                     System.out.println("Login berhasil. Selamat datang, " + inputUsername + "!");
24                     login = true;
25                 } else {
26                     System.out.println("Login gagal. Password salah.");
27                 }
28             } else {
29                 System.out.println("Login gagal. Username tidak ditemukan. Silakan coba lagi.");
30                 continue; // Meminta pengguna untuk mengisi ulang username
31             }
32         }
33
34         scanner.close();
35     }
36 }
```

2. Fitur Transaksi

```
1  import java.util.Scanner;
2
3  public class Transaksi{
4      public static void main(String[] args) {
5          //inisialisasi
6          Scanner input = new Scanner(System.in);
7          int beratBarang, jarak;
8          double tarifDasar =10000, tarifDasarKg =1000, tarifDasarKm =2000;
9          double biayaBerat, biayaJarak, biayaTotal, jumlahBarang, biayaAkhir=0;
10
11         System.out.println("Masukkan jumlah barang: ");
12         jumlahBarang = input.nextInt();
13
14         int i = 0;
15         //melakukan perulangan sesuai jumlah barang yang akan dikirim
16         while (i < jumlahBarang) {
17             System.out.println ("Masukkan berat barang ke-" + (i+1) + ": ");
18             beratBarang = input.nextInt();
19             System.out.println ("Masukkan jarak pengiriman barang ke-" + (i+1) + ": ");
20             jarak = input.nextInt();
21             i++;
22             //menentukan harga berdasarkan data yang telah diinputkan
23             if(beratBarang <= 5){
24                 biayaBerat=0;
25             }else{
26                 biayaBerat=(beratBarang-5)*tarifDasarKg;
27             }
28             if(jarak <= 100){
29                 biayaJarak=0;
30             }else{
31                 biayaJarak=(jarak-100)*tarifDasarKm;
32             }
33             biayaTotal = tarifDasar+biayaBerat+biayaJarak;
34             biayaAkhir += biayaTotal;
35         }
36         System.out.println("estimasi biaya pengiriman: " +biayaAkhir);
37     }
38 }
39
```