**LAPORAN PRAKTIKUM**

**JOBSHEET 7**

**Perulangan 1**

****

Oleh:

**Ekya Muhammad Hasfi Fadlilurrahman**

**2341720111**

**Jl. Soekarno Hatta No.9 Telp. (0341) 404424**

**KECAMATAN LOWOKWARU**

**KODE POS : 65141**

**TAHUN PELAJARAN 2023/2024**

**2. Praktikum**

**2.1** **Percobaan 1 : Menghitung Bilangan Kelipatan Menggunakan FOR Waktu Percobaan: 60 menit**

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. Buka text editor. Buat file baru, beri nama ForKelipatanNoAbsen.java

2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi main()

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 dan counter dengan nilai 0

6. Tambahkan kode berikut ini untuk menerima input dari keyboard

7. Buat 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

10. Commit dan push kode program ke github

import java.util.Scanner;

public class ForKelipatan08 {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int kelipatan, jumlah = 0, counter = 0;

        System.out.print("Masukkan bilangan kelipatan (1-9): ");

        kelipatan = scan.nextInt();

        for (int i = 1; i <= 50; i++) {

            if (i % kelipatan == 0) {

                jumlah += i;

                counter++;

            }

        }

        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);

        System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah);

    }

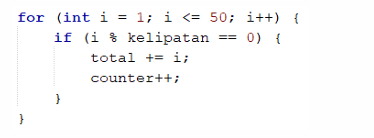
}

**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!

- Perulangan FOR memiliki 3 komponen yaitu inisialisasi, kondisi, dan update. Dalam kode program tersebut int i = 1 berperan sebagai inisiaisasi, i <= 50 berperan sebagai kondisi, dan i++ berperan sebagai update.

2. Jelaskan alur kerja dari potongan kode program berikut!



- Pertama kode akan menginisialisasi nilai i = 1, dan nilainya akan terus ditambah 1 hingga i mencapai nilai 50. Jika nilai i % kelipatan = 0 maka variabel total akan ditambah dengan nilai I dan counter akan bertambah 1.

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 ModifForKelipatan08 {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int kelipatan, jumlah = 0;

        double rata, counter = 0;

        System.out.print("Masukkan bilangan kelipatan (1-9): ");

        kelipatan = scan.nextInt();

        for (int i = 1; i <= 50; i++) {

            if (i % kelipatan == 0) {

                jumlah += i;

                counter++;

            }

        }

        rata = jumlah / counter;

        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %.0f\n", kelipatan, counter);

        System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah);

        System.out.printf("Rata-rata dari dari seluruh bilangan kelipatan %d adalah %.2f", kelipatan, rata);

    }

}



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 WhileKelipatan08 {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int kelipatan, jumlah = 0;

        double rata, counter = 0;

        System.out.print("Masukkan bilangan kelipatan (1-9): ");

        kelipatan = scan.nextInt();

        int i = 1;

        while (i <= 50) {

            if (i % kelipatan == 0) {

                jumlah += i;

                counter++;

            }

            i++ ;

        }

        rata = jumlah / counter;

        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %.0f\n", kelipatan, counter);

        System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah);

        System.out.printf("Rata-rata dari dari seluruh bilangan kelipatan %d adalah %.2f", kelipatan, rata);

    }

}



**2.2 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. Buka text editor. Buat file baru, beri nama WhileGajiNoAbsen.java

2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi main()

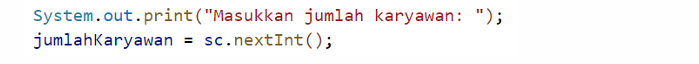
3. Tambahkan library Scanner

4. Buat deklarasi Scanner dengan nama scan

5. Deklarasikan variabel jumlahKaryawan dan jumlahJamLembur bertipe int serta gajiLembur dan totalGajiLembur bertipe double. Inisialisasi variabel gajiLembur dan totalGajiLembur dengan nilai 0

6. Deklarasikan variabel jabatan bertipe String

7. Tambahkan kode berikut ini untuk menerima input dari keyboard guna menentukan jumlah karyawan yang akan dihitung gajinya

****

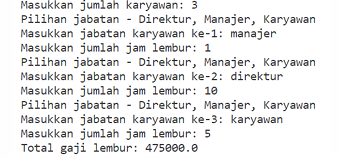
8. Buat struktur perulangan WHILE dengan kondisi pemilihan IF-ELSE dan CONTINUE untuk menentukan gaji lembur berdasarkan jabatan karyawan

****

9. Tampilkan hasil perhitungan jumlah gaji



10. Jalankan program tersebut. Cocokkan hasil compile kode program Anda dengan gambar berikut ini



11. Push dan commit kode program ke github.

import java.util.Scanner;

public class WhileGaji08 {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int jumlahJamLembur, jumlahKaryawan;

        double gajiLembur = 0, totalGajiLembur = 0;

        String jabatan;

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

        jumlahKaryawan = scan.nextInt();

        int i = 0;

        while (i < jumlahKaryawan) {

            System.out.println("Pilihan jabatan - Direktur, Manajer, Karyawan");

            System.out.print("Masukkan jabatan ke-" + (i+1) + ": ");

            jabatan = scan.next();

            System.out.print("Masukkan jumlah jam lembur: ");

            jumlahJamLembur = scan.nextInt();

            i++;

            if (jabatan.equalsIgnoreCase("direktur")) {

                continue;

            }else if (jabatan.equalsIgnoreCase("manajer")) {

                gajiLembur = jumlahJamLembur \* 100000;

            }else if (jabatan.equalsIgnoreCase("karyawan")) {

                gajiLembur = jumlahJamLembur \* 75000;

            }

            totalGajiLembur += gajiLembur;

        }

        System.out.println("Total gaji lembur: " + totalGajiLembur);

    }

}



**Pertanyaan**

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

**-** i < jumlah karyawan. Perulangan dihentikan saat nilai i sama dengan nilai jumlah karyawan.

2, Pada potongan kode berikut,



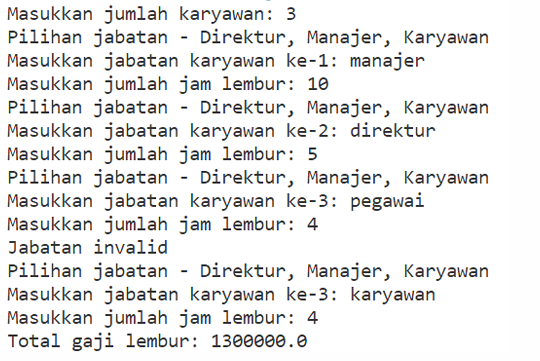
Apa yang sebenarnya terjadi jika variabel jabatan berisi nilai “DIREKTUR”? Apa peran CONTINUE yang dituliskan di dalam sintaks perulangan?

- Perintah continue hampir sama dengan break hanya saja continue menghentikan satu kali perulangan sedangkan break menghentikan perulangan secara keseluruhan. Continue juga dapat digunakan dalam looping untuk melewati program dan lanjut untuk loop selanjutnya.

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!

- Jika kita memasukkan “direktur” maka i++ tidak akan dibaca karena ada perintah continue yang berfungsi untuk melewati program dan akan lanjut ke loop selanjutnya dikarenakan i++ tidak terbaca. Dan karena jumlah i tidak berubah maka itu juga akan mempengaruhi jumlah output yang dikeluarkan.

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

****

import java.util.Scanner;

public class ModifWhileGaji08 {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int jumlahJamLembur, jumlahKaryawan;

        double gajiLembur = 0, totalGajiLembur = 0;

        String jabatan;

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

        jumlahKaryawan = scan.nextInt();

        int i = 0;

        while (i < jumlahKaryawan) {

            System.out.println("Pilihan jabatan - Direktur, Manajer, Karyawan");

            System.out.print("Masukkan jabatan ke-" + (i+1) + ": ");

            jabatan = scan.next();

            System.out.print("Masukkan jumlah jam lembur: ");

            jumlahJamLembur = scan.nextInt();

            i++;

            if (jabatan.equalsIgnoreCase("direktur")) {

                continue;

            }else if (jabatan.equalsIgnoreCase("manajer")) {

                gajiLembur = jumlahJamLembur \* 100000;

            }else if (jabatan.equalsIgnoreCase("karyawan")) {

                gajiLembur = jumlahJamLembur \* 75000;

            }else if (!jabatan.equalsIgnoreCase("direktur") && !jabatan.equalsIgnoreCase("manajer") && !jabatan.equalsIgnoreCase("karyawan")) {

                i--;

                System.out.println("Jabatan invalid");

            }

            totalGajiLembur += gajiLembur;

        }

        System.out.println("Total gaji lembur: " + totalGajiLembur);

    }

}

5. Push dan commit kode program ke github

**2.3 Percobaan 3: Menghitung Jatah Cuti Menggunakan DO-WHILE**

Pada percobaan ini dilakukan pembuatan kode program menggunakan DO-WHILE untuk menghitung jatah cuti yang dimiliki oleh pagawai. Pegawai mempunyai jatah cuti sebanyak 5 hari. Jatah cuti akan dikurangi perhati setiap kali digunakan. Saat jatah cuti sisa 2 hari, pegawai mendapat peringatan untuk berhenti menggunakan jatah cutinya.

1. Buka text editor. Buat file baru, beri nama DoWhileCutiNoAbsen.java

2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi main()

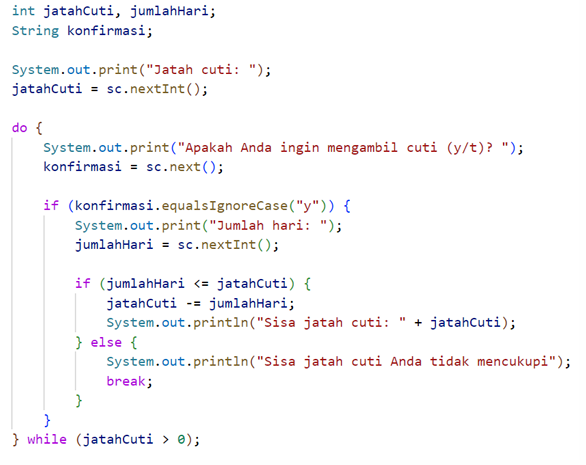
3. Tambahkan library Scanner

4. Buat deklarasi Scanner dengan nama sc

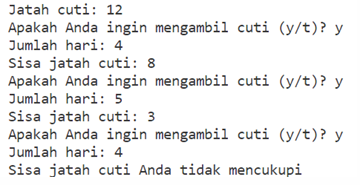
5. Buatlah variabel jatahCuti dan jumlahHari bertipe int

6. Buatlah variabel konfirmasi bertipe String

7. Buat struktur perulangan DO-WHILE untuk menerima input dari keyboard dan menghitung jatah cuti



8. Jalankan program tersebut. Cocokkan hasil running program yang sudah Anda buat apakah sudah sesuai dengan tampilan berikut



9. Push dan commit kode program ke github

import java.util.Scanner;

public class DoWhileCuti08 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int jatahCuti, jumlahHari;

        String konfirmasi;

        System.out.print("Jatah cuti: ");

        jatahCuti = sc.nextInt();

        do{

            System.out.print("Apakah anda ingin mengambil cuti (y/t)? ");

            konfirmasi = sc.next();

            if (konfirmasi.equalsIgnoreCase("y")) {

                System.out.print("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;

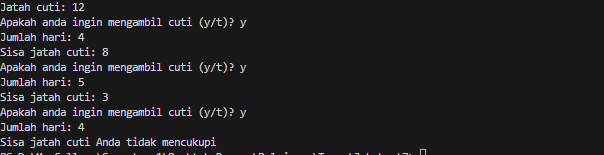
                }

            }

        }while (jatahCuti > 0);

    }

}



**Pertanyaan**

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

- BREAK berfungsi mengeluarkan program yang berjalan dari looping

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 mengisikan jumlah hari sesuai jatah cuti.

import java.util.Scanner;

public class ModifDoWhileCuti08 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int jatahCuti, jumlahHari;

        String konfirmasi;

        System.out.print("Jatah cuti: ");

        jatahCuti = sc.nextInt();

        do{

            System.out.print("Apakah anda ingin mengambil cuti (y/t)? ");

            konfirmasi = sc.next();

            if (konfirmasi.equalsIgnoreCase("y")) {

                System.out.print("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");

                    continue;

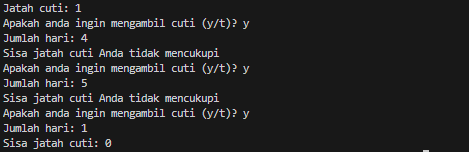
                }

            }

        }while (jatahCuti > 0);

    }

}



3. Push dan commit kode program ke github

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

- Pertanyaan “Apakah anda ingin mengambil cuti?” akan diulang terus menerus hingga kita menginput y jika nilai yang kita input lebih dari 0 karena program akan diulang sampai jatahCuti <=0.

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

-

import java.util.Scanner;

public class Modif2DoWhileCuti08 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int jatahCuti, jumlahHari;

        String konfirmasi;

        System.out.print("Jatah cuti: ");

        jatahCuti = sc.nextInt();

        do{

            System.out.print("Apakah anda ingin mengambil cuti (y/t)? ");

            konfirmasi = sc.next();

            if (konfirmasi.equalsIgnoreCase("y")) {

                System.out.print("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");

                    continue;

                }

            }else {

                break;

            }

        }while (jatahCuti > 0);

    }

}

6. Push dan commit kode program ke github

**Tugas**

• Implementasikan flowchart yang telah dibuat pada tugas pertemuan 7 Matakuliah Dasar Pemrograman terkait project ke dalam kode program

• Push dan commit hasil kode program anda ke repository project Anda

• Catatan: tugas hanya boleh menerapkan materi dari pertemuan 1 hingga pertemuan 7.