**LAPORAN HASIL PRAKTIKUM DASAR PEMROGRAMAN**

**JOBSHEET 7**

**PERULANGAN PADA BAHASA JAVA**



MUHAMMAD AMMAR HAFIZH

(2341720074)

D-IV TEKNIK INFORMATIKA – 1E

**Jurusan Teknologi Informasi**

**Politeknik Negeri Malang**

**1. TUJUAN**

● Mahasiswa dapat menjelaskan format penulisan program perulangan bagian 1

● Mahasiswa dapat mengimplementasikan flowchart perulangan bagian 1 menggunakan bahasa pemrograman Java

**1. PERULANGAN**

**1.1 Menghitung Kelipatan Pada Sebuah Bilangan**

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.

Kode program yang akan kita gunakan seperti di bawah.

package SCRIPT;

import java.util.Scanner;

/\*\*

\* ForKelipatan17

\*/

public class ForKelipatan17 {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int kelipatan, jumlah = 0, counter = 0,total = 0;

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

kelipatan = scan.nextInt();

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

if (i % kelipatan == 0) {

total += 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,total);

}

}

Jika kita mengikuti kode di atas maka program yang akan keluar adalah seperti ini.



**1.2 Pertanyaan**

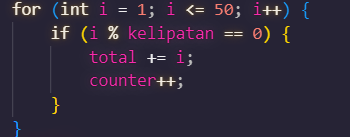
1. Terdapat tiga komponen perulangan pada sintaks FOR. Berdasarkan Percobaan 1 tersebut, sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program yang telah dibuat!

* (int = i) Bertujuan untuk mendeklarasikan i sebagai variable tipe data integer

(i <= 50) Bertujuan untuk mengoperasikan aritmatika yaitu i kurang sama dengan 50 yang harus terpenuhi

(i ++) Bertujuan untuk mengupdate nilai variable i dengan cara increment

2. Jelaskan alur kerja dari potongan kode program berikut!



* Pertama tama for akan membuat i menjadi tipe data integer dan kondisi yang harus terpenuhi adalah i kurang sama dengan 50 dan jika i belum lebih dari 50 akan mengupdate nilainya increment setelah itu dilanjut pemilihan if dengan kondisi i modulus kelipatan sama dengan 0 jika kondisi terpenuhi maka variable total ditambah dengan nilai variable i dan variable counter akan mengalami increment skema ini akan berulang sampai for memenuhi kondisi atau nilai i lebih dari 50.

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.

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

**1.3 Menghitung Gaji Lembur**

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.

Jika kita menemukan kasus seperti di atas maka kode program yang akan kita gunakan adalah seperti berikut.

package SCRIPT;

import java.util.Scanner;

public class WhileGaji17 {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int jumlahKaryawan, jumlahJamLembur;

double gajiLembur = 0, totalGajiLembur = 0;

String jabatan;

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

jumlahKaryawan = scan.nextInt();

int i = 0;

while (i < jumlahKaryawan) {

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

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

jabatan = scan.next();

System.out.print("Masukan 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;

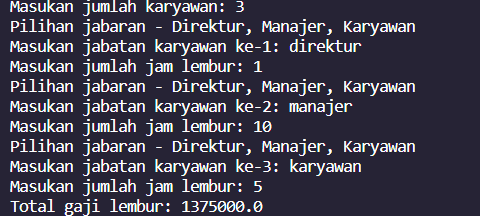
}

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

}

}

Dengan kode program di atas makan program yang akan muncul adalah seperti berikut.

****

**1.4 Pertanyaan**

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

while (i < jumlahKaryawan) {

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

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

jabatan = scan.next();

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

jumlahJamLembur = scan.nextInt();

i++;

bisa dilihat code yang diberi warna kuning adalah syarat perulangan while jika user menginput 2 maka while akan melakukan perulangan sekali dan seterusnya.

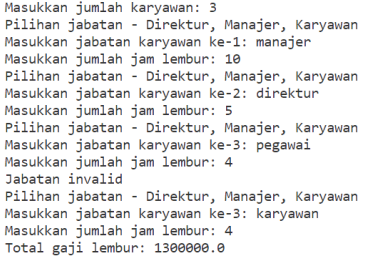
2. Pada potongan kode berikut, Apa yang sebenarnya terjadi jika variabel jabatan berisi nilai “DIREKTUR”? Apa peran CONTINUE yang dituliskan di dalam sintaks perulangan?

* Jika user menginput jabatan direktur maka tidak ada perintah untuk meng-operasikan jam lembur dan continue berperan untuk melanjutkan computer membaca program.

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 memindahkan i++ ke bagian akhir maka jika kita menginput direktur akan terus menerus terduplicate karena code program continue

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



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

continue;

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

gajiLembur = jumlahJamLembur \* 100000;

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

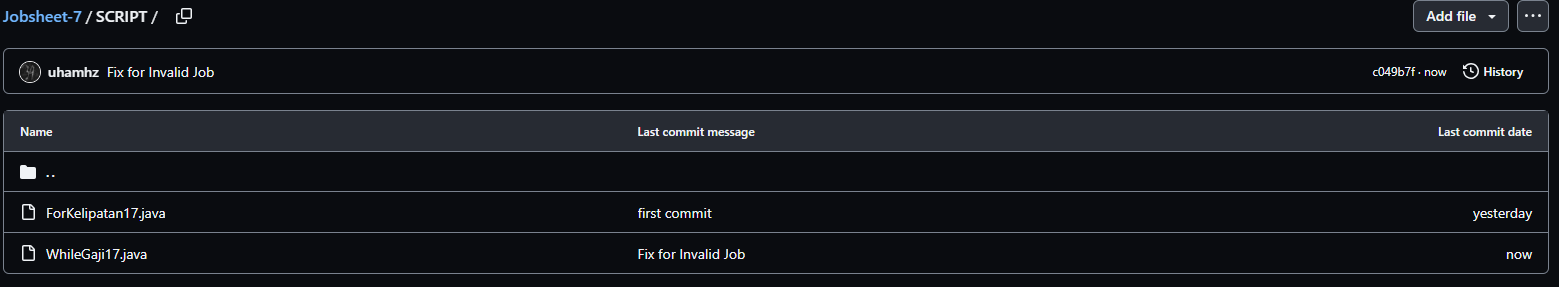
gajiLembur = jumlahJamLembur \* 75000;

} else {

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

}

5. Push dan commit kode program ke github

****

**1.4 Menghitung Jatah Cuti**

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.

Dengan kasus di atas maka kita akan membuat program seperti berikut.

package SCRIPT;

import java.util.Scanner;

public class DoWhileCuti17 {

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;

}

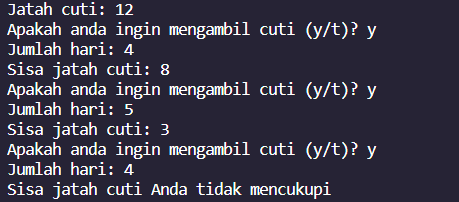
}

} while (jatahCuti > 0);

}

}

Dengan kode program di atas akan menghasilkan program seperti berikut.



**1.5 Pertanyaan**

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

* Break berfungsi untuk menghentikan perulangan jika kita menghapus break maka perulangan akan tetap berlangsung walau sudah melewati batas jumlah cuti yang kita dapat

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.

if (jumlahHari <= jatahCuti) {

jatahCuti -= jumlahHari;

System.out.println("Sisa jatah cuti: " + jatahCuti);

} else {

System.out.printf("Invalid sisa jatah cuti Anda %d !\n", jatahCuti);

continue;

}

3. Push dan commit kode program ke github



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

* Tetap mengalami perulangan karena tidak ada perintah jika user menginput “t” maka menghentikan perulangan.

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

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.printf("Invalid sisa jatah cuti Anda %d !\n", jatahCuti);

continue;

}

} else if (konfirmasi.equalsIgnoreCase("t")) {

break;

}

6. Push dan commit kode program ke github

****