**JOBSHEET 7**

**LOOPING**



19

Muhammad Alif Febriansyah

2341720025

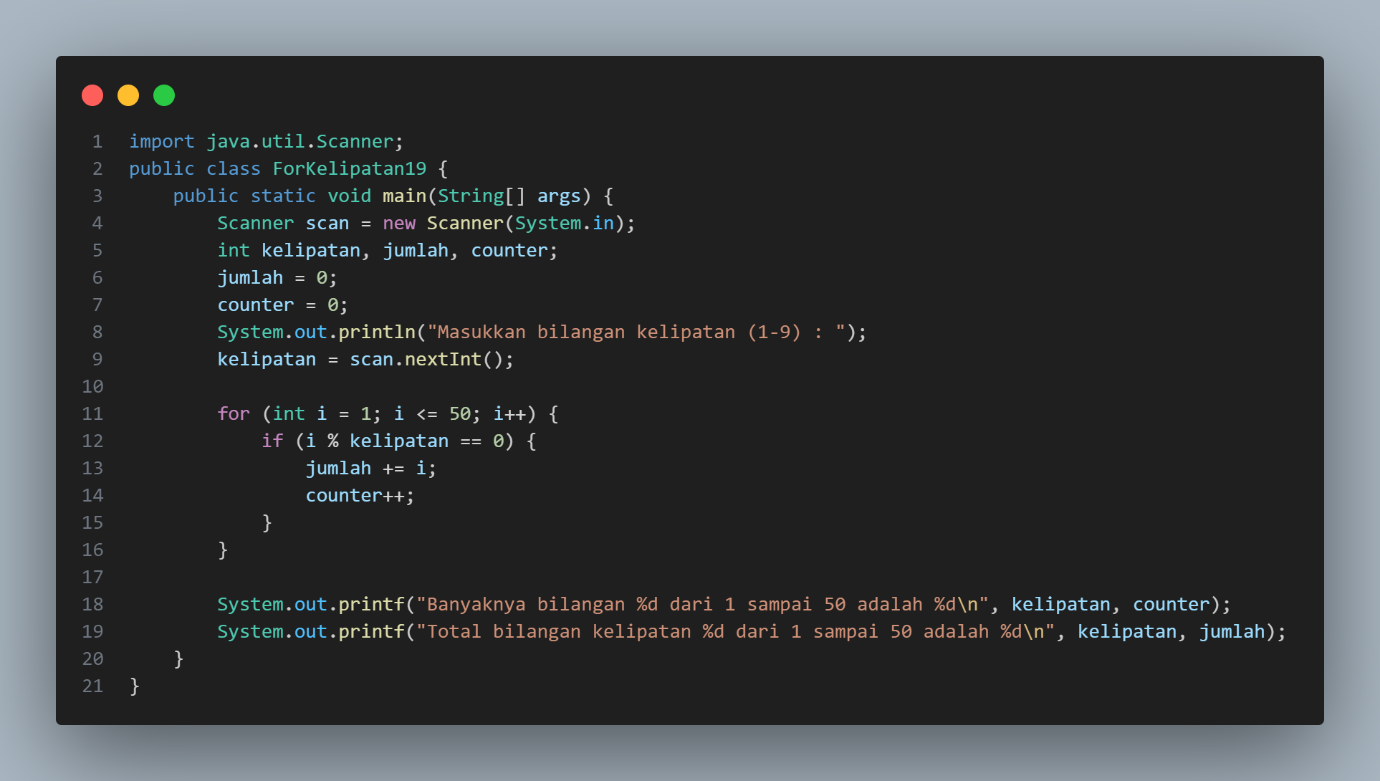
<https://github.com/alfbrynn>

Teknologi Informasi\_Teknik Informatika

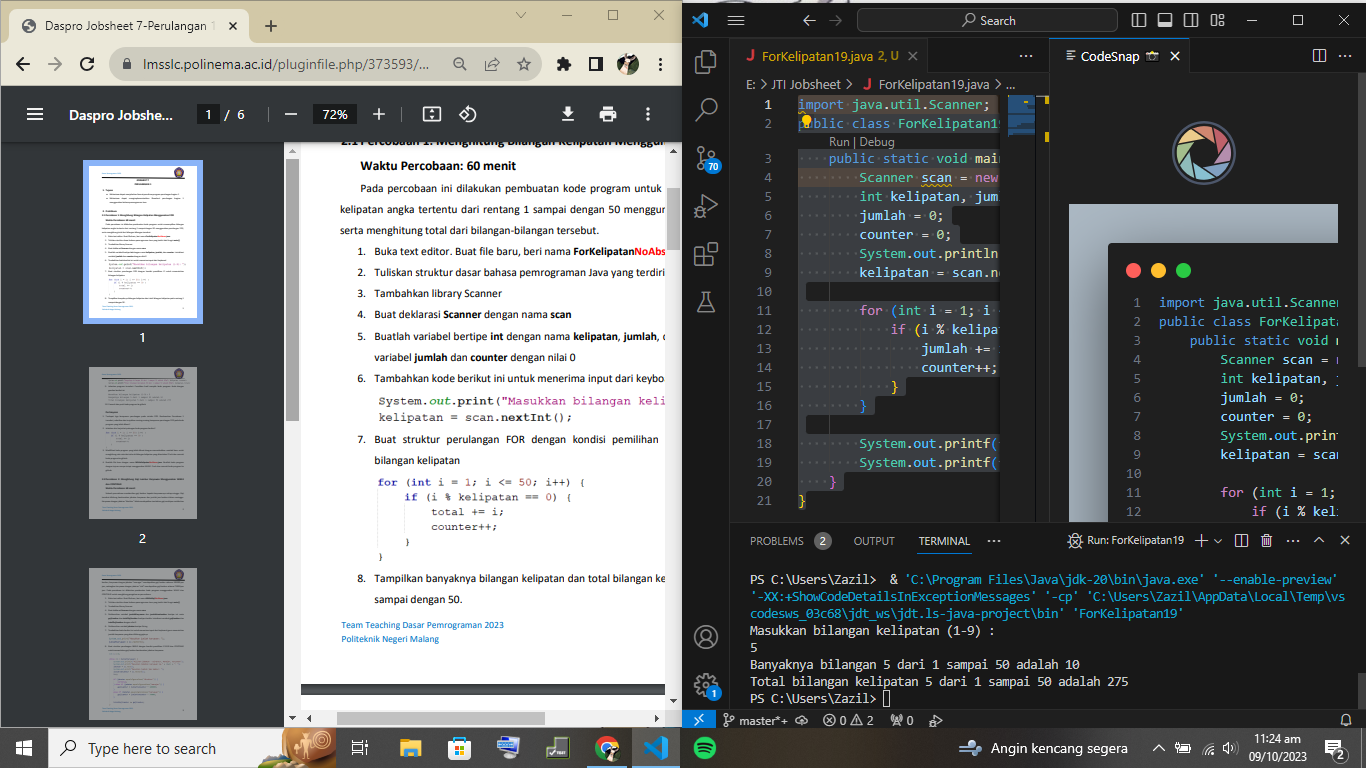
Polinema

2023/2024

Percobaan 1



Output



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!

Komponen perulangan pada sintaks FOR adalah:

* + Inisialisasi: `int i = 1;` (inisialisasi variabel i dengan nilai 1)
  + Kondisi: `i <= 50;` (kondisi untuk menjalankan perulangan selama i kurang dari atau sama dengan 50)
  + Perubahan: `i++` (increment i setiap kali iterasi)

1. Jelaskan alur kerja dari potongan kode program berikut!

Untuk menjelaskan alur kerja dari potongan kode program tersebut:

1. Inisialisasi variabel: Variabel `i` diinisialisasi dengan nilai 1, yang akan digunakan sebagai penghitung untuk iterasi perulangan.

2. Perulangan for: Perulangan akan berjalan sebanyak 50 kali, dimulai dari 1 hingga 50.

3. Pengecekan kelipatan: Pada setiap iterasi, dilakukan pengecekan apakah `i` merupakan kelipatan dari bilangan yang dimasukkan pengguna (`kelipatan`). Ini dilakukan dengan menggunakan operator modulo (`%`). Jika `i` adalah kelipatan, maka akan memasuki blok if.

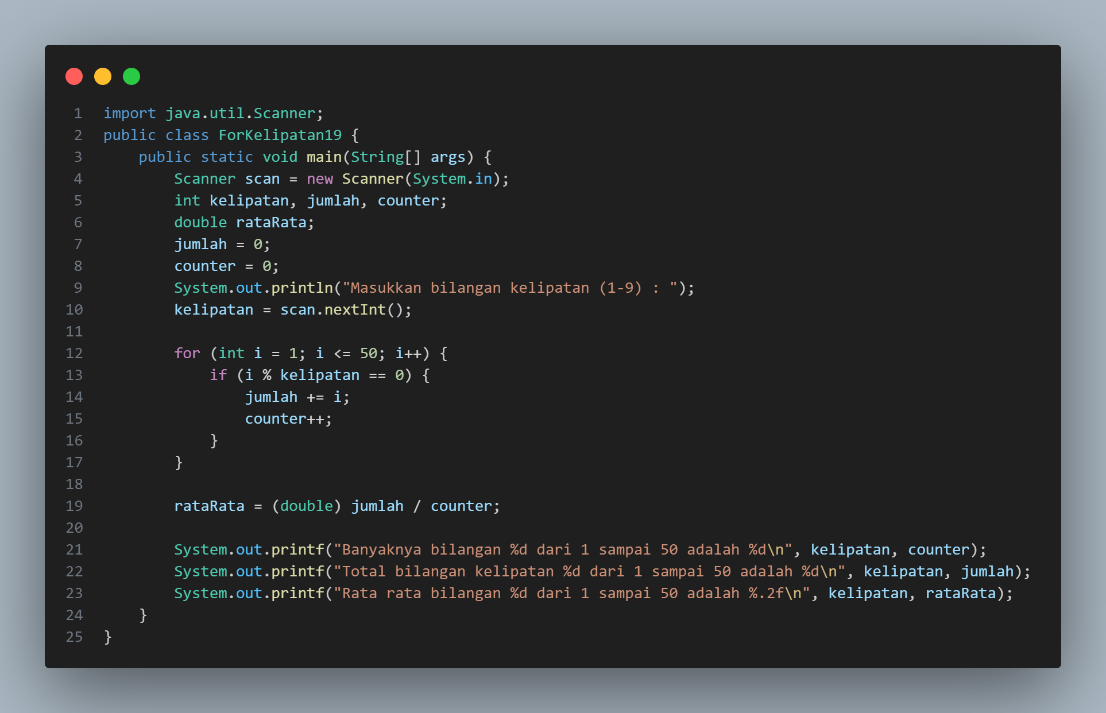
4. Penjumlahan dan pencatatan: Jika `i` adalah kelipatan, maka nilai `i` akan ditambahkan ke variabel `jumlah`, yang digunakan untuk menghitung total dari bilangan-bilangan kelipatan. Selain itu, variabel `counter` akan diincrement untuk menghitung banyaknya bilangan kelipatan.

5. Perubahan nilai `i`: Setelah setiap iterasi, nilai `i` akan diincrement menggunakan pernyataan `i++`.

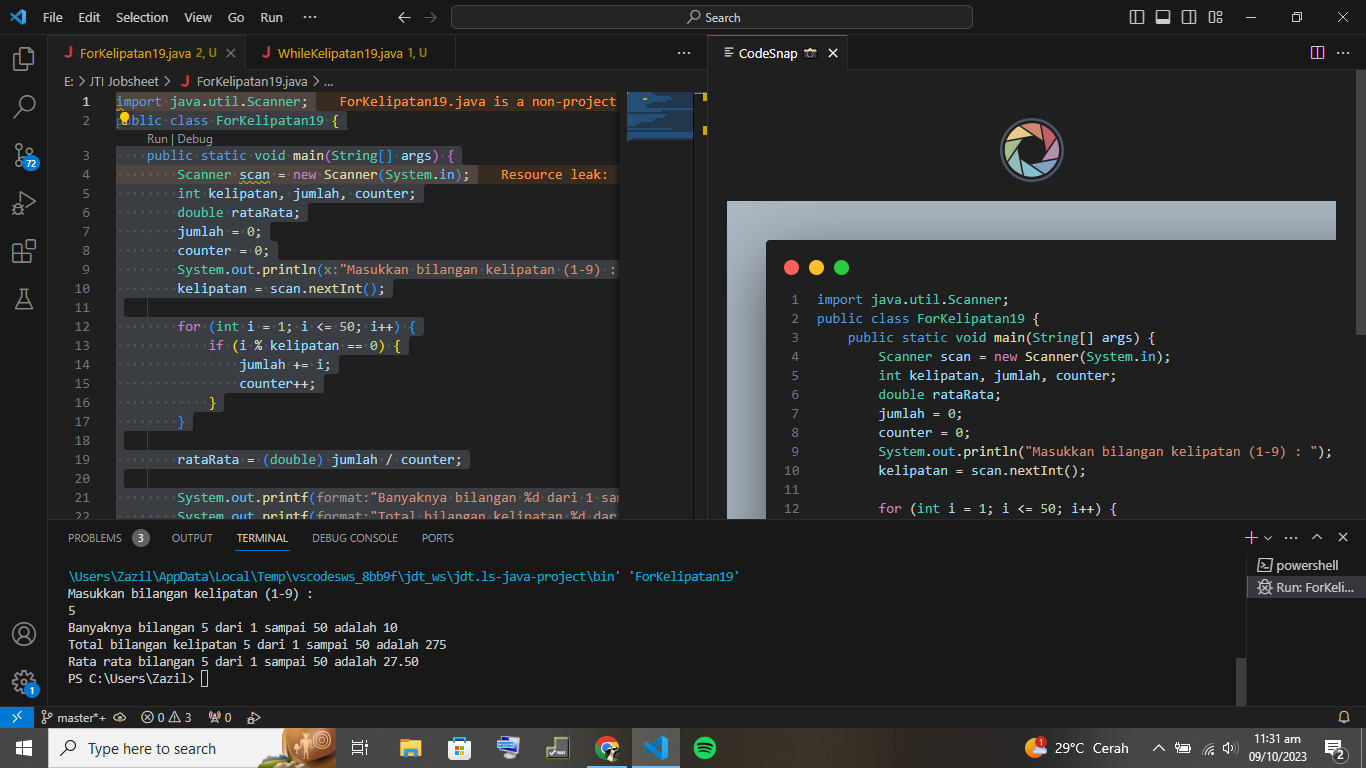
6. Pengecekan batas perulangan: Perulangan akan terus berlanjut selama nilai `i` masih kurang dari atau sama dengan 50.

Jadi, potongan kode program ini melakukan iterasi dari 1 hingga 50, memeriksa apakah setiap bilangan adalah kelipatan dari bilangan yang dimasukkan pengguna, dan mengakumulasi total serta mencatat banyaknya bilangan kelipatan.

1. 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.



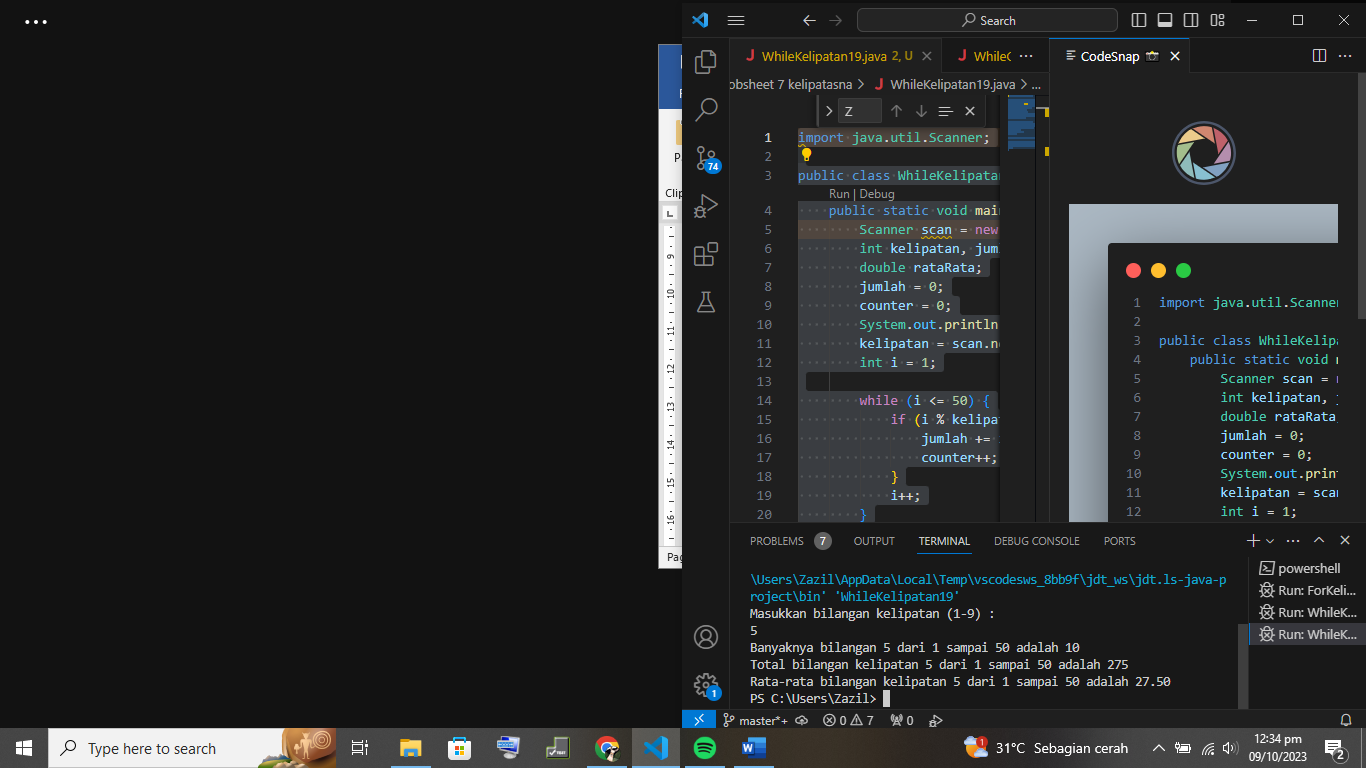
Output



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



Output



percobaan 2

import java.util.Scanner;

public class WhileGaji19 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int jumlahKaryawan, jumlahJamLembur;

        double gajiLembur, totalGajiLembur;

        gajiLembur = 0;

        totalGajiLembur = 0;

        String jabatan;

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

        jumlahKaryawan = sc.nextInt();

        int i = 0;

        while (i < jumlahKaryawan) {

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

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

            jabatan = sc.next();

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

            jumlahJamLembur = sc.nextInt();

            i++;

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

                continue;

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

                gajiLembur = jumlahJamLembur \* 100000;

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

                gajiLembur = jumlahJamLembur \* 75000;

            }

            totalGajiLembur += gajiLembur;

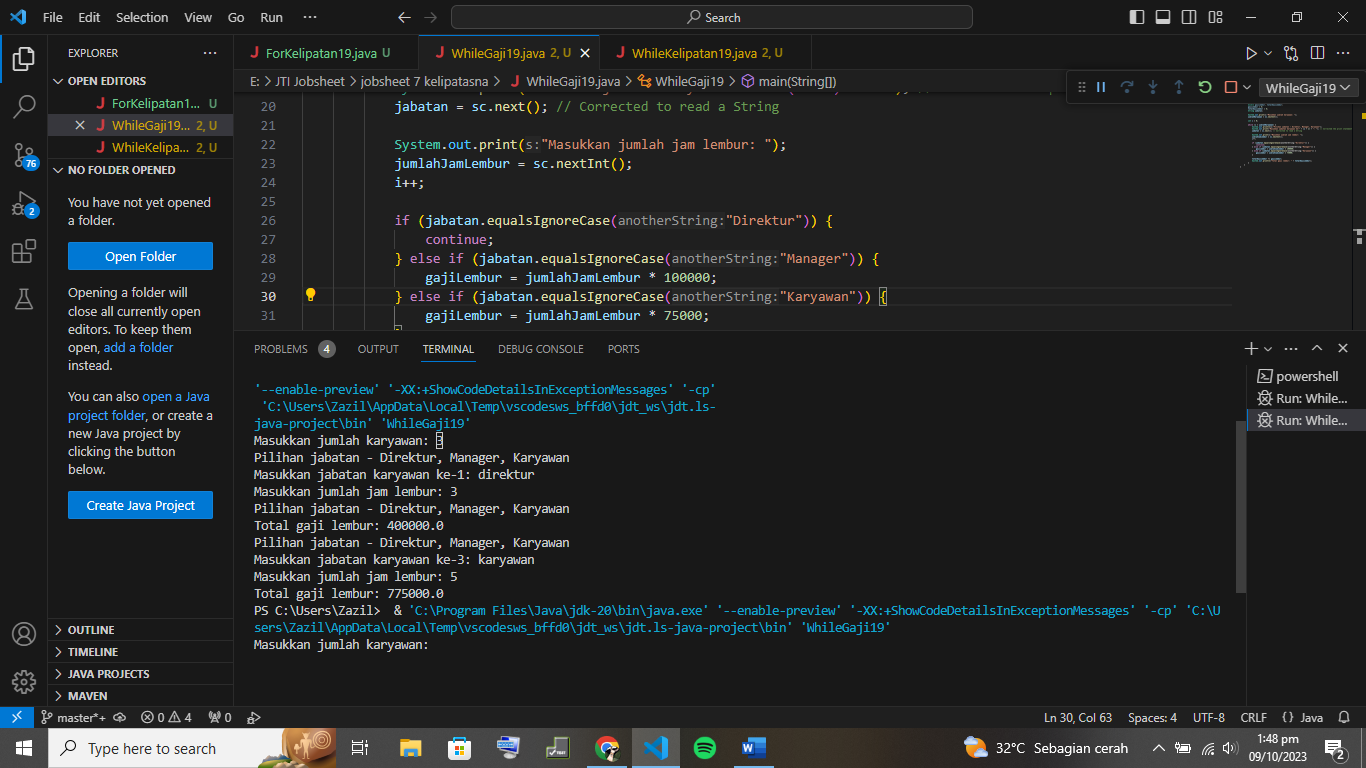
            System.out.println("Total gaji lembur: " + totalGajiLembur);

        }

    }

}

Output



Pertanyaan 1

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

Perulangan WHILE dalam program ini akan berhenti saat variable ‘I’ mencapai nilai yang sama dengan ‘JumlahKaryawan’, sesuai dengan kondisi while ‘I < jumlahKaryawan’.

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

Jika variabel `jabatan` berisi nilai "DIREKTUR", maka kondisi `if (jabatan.equalsIgnoreCase("Direktur"))` akan dievaluasi sebagai true, dan pernyataan `continue` akan memulai iterasi perulangan baru tanpa mengeksekusi kode di bawahnya dalam lingkup while. Dalam konteks ini, jika jabatan adalah "DIREKTUR", perulangan akan melanjutkan ke iterasi berikutnya tanpa menghitung gaji lembur dan mencetak total gaji lembur.

1. 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!

Pembaruan `i++` diletakkan di posisi tengah karena keperluan perulangan dalam kode ini adalah untuk mendapatkan input dan menghitung gaji lembur untuk setiap karyawan. Memindahkan `i++` ke bagian akhir akan mengakibatkan peningkatan `i` terjadi setelah semua perhitungan dan pencetakan gaji lembur dilakukan, sehingga perulangan akan keliru menghitung gaji lembur untuk karyawan selanjutnya. Saat "direktur" dimasukkan sebagai jabatan karyawan pertama, karena perulangan terus berlanjut tanpa melakukan perhitungan gaji lembur, program akan mengabaikan input "direktur" dan melanjutkan ke karyawan berikutnya tanpa memperhitungkan gaji lembur untuk direktur.

1. Modifikasi program untuk menghandle jabatan yang invalid

import java.util.Scanner;

public class WhileGaji19 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int jumlahKaryawan, jumlahJamLembur;

        double gajiLembur, totalGajiLembur;

        gajiLembur = 0;

        totalGajiLembur = 0;

        String jabatan;

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

        jumlahKaryawan = sc.nextInt();

        int i = 0;

        while (i < jumlahKaryawan) {

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

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

            jabatan = sc.next();

            if (!(jabatan.equalsIgnoreCase("Direktur") || jabatan.equalsIgnoreCase("Manager") || jabatan.equalsIgnoreCase("Karyawan"))) {

                System.out.println("Jabatan tidak valid. Silakan masukkan jabatan jabatan yang valid.3 ");

                continue;

            }

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

            jumlahJamLembur = sc.nextInt();

            i++;

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

                continue;

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

                gajiLembur = jumlahJamLembur \* 100000;

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

                gajiLembur = jumlahJamLembur \* 75000;

            }

            totalGajiLembur += gajiLembur;

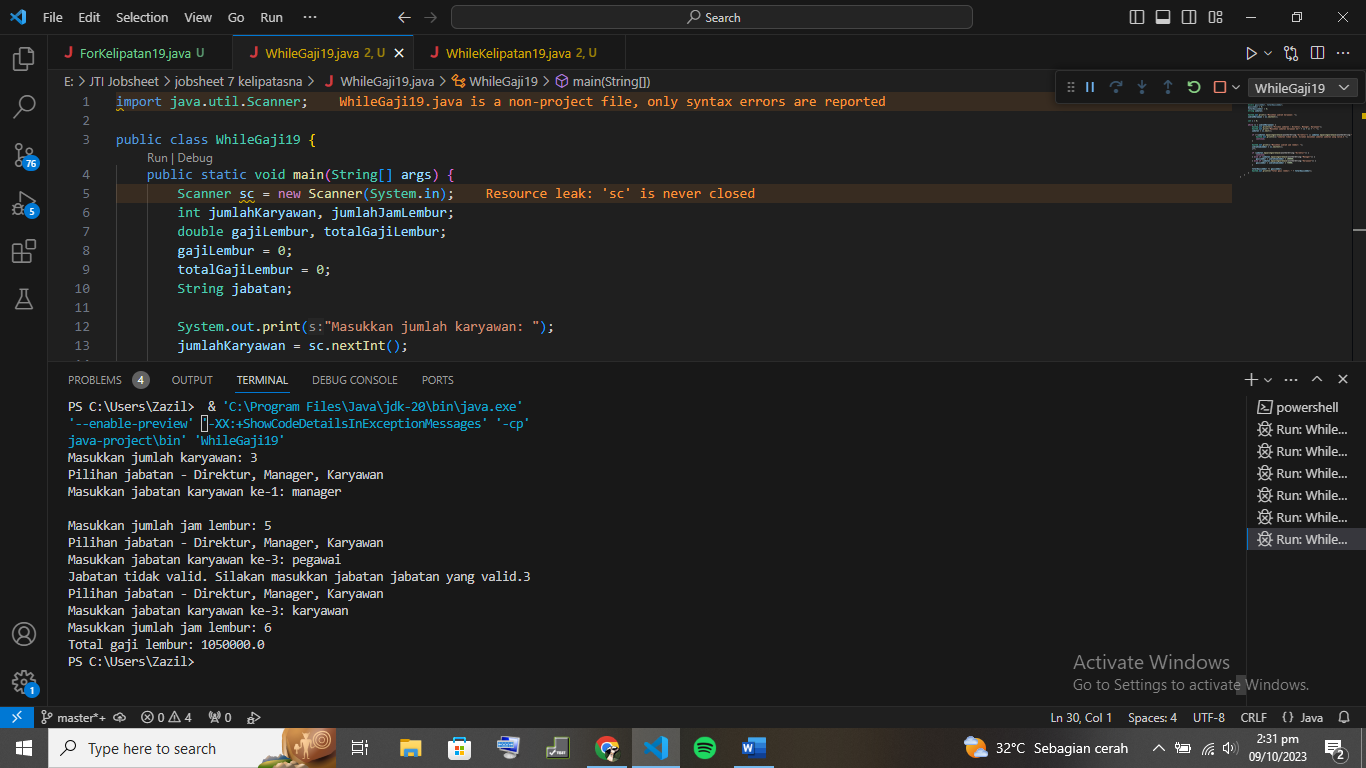
            System.out.println("Total gaji lembur: " + totalGajiLembur);

        }

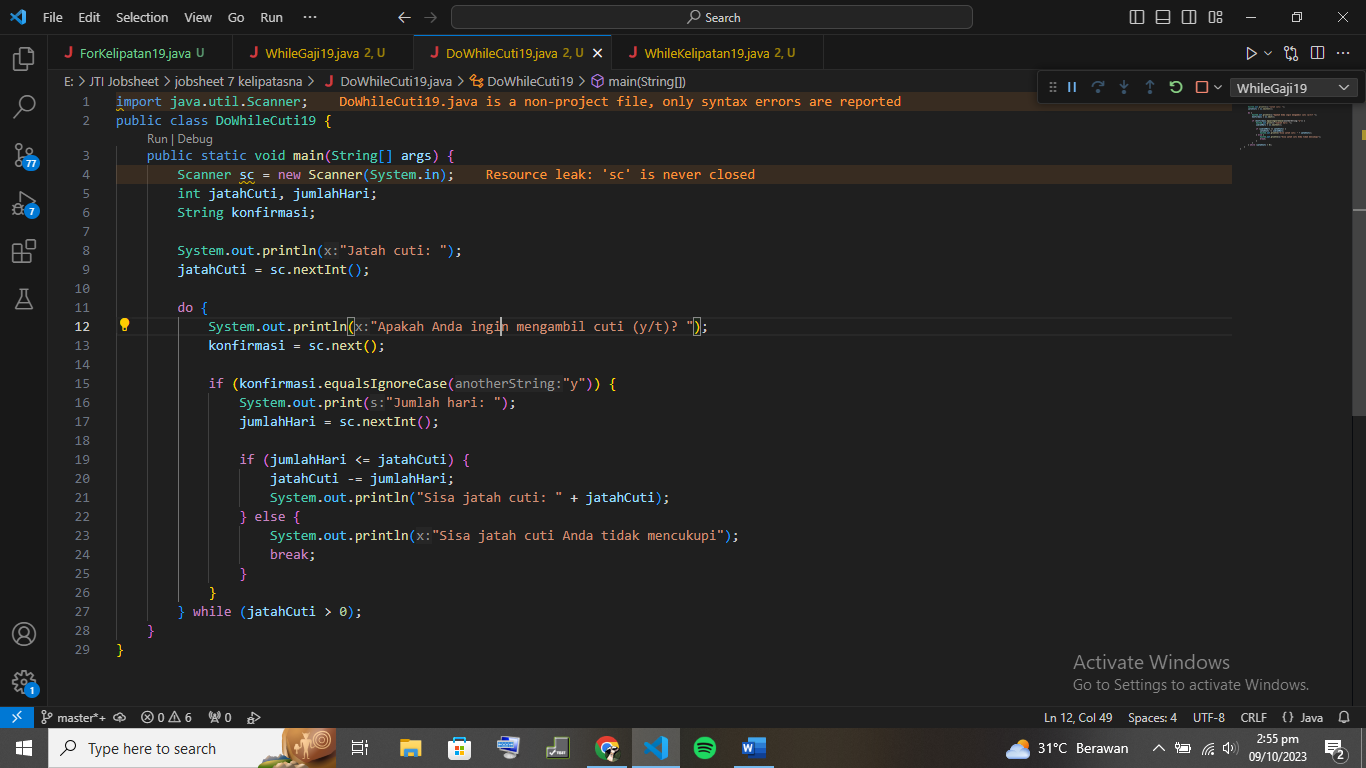
    }

}

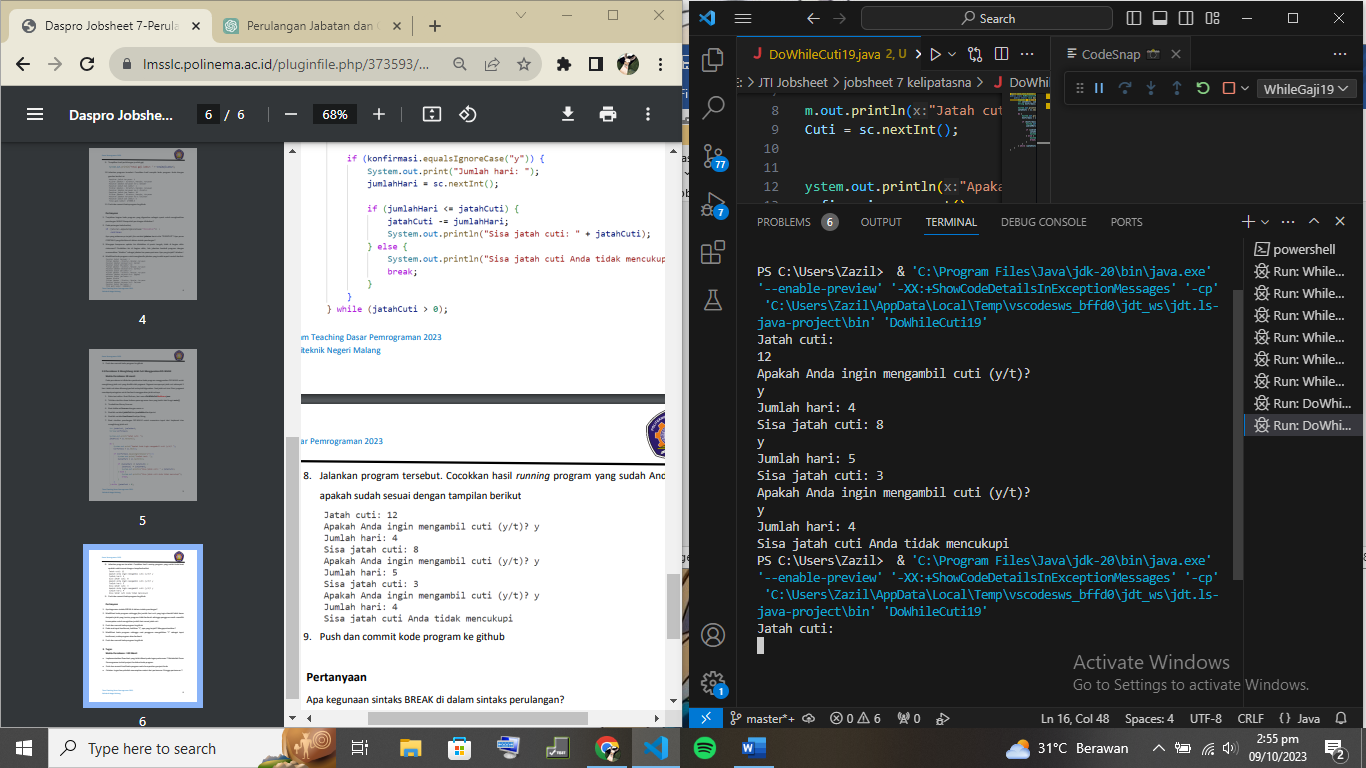
output



Percobaan 3



Output



Pertanyaan

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

Untuk menghentikan eksekusi darisebuah loop

1. 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.

public class DoWhileCuti119 {

    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();

        outerLoop: 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");

                    // Tambahkan inner loop untuk memungkinkan pengisian ulang jumlah hari

                    while (true) {

                        System.out.print("Apakah Anda ingin mengisi jumlah hari kembali (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);

                                break;  // Keluar dari inner loop

                            } else {

                                System.out.println("Sisa jatah cuti Anda tidak mencukupi");

                            }

                        } else {

                            break outerLoop;  // Keluar dari outer loop jika pengguna tidak ingin isi jumlah hari lagi

                        }

                    }

                }

            }

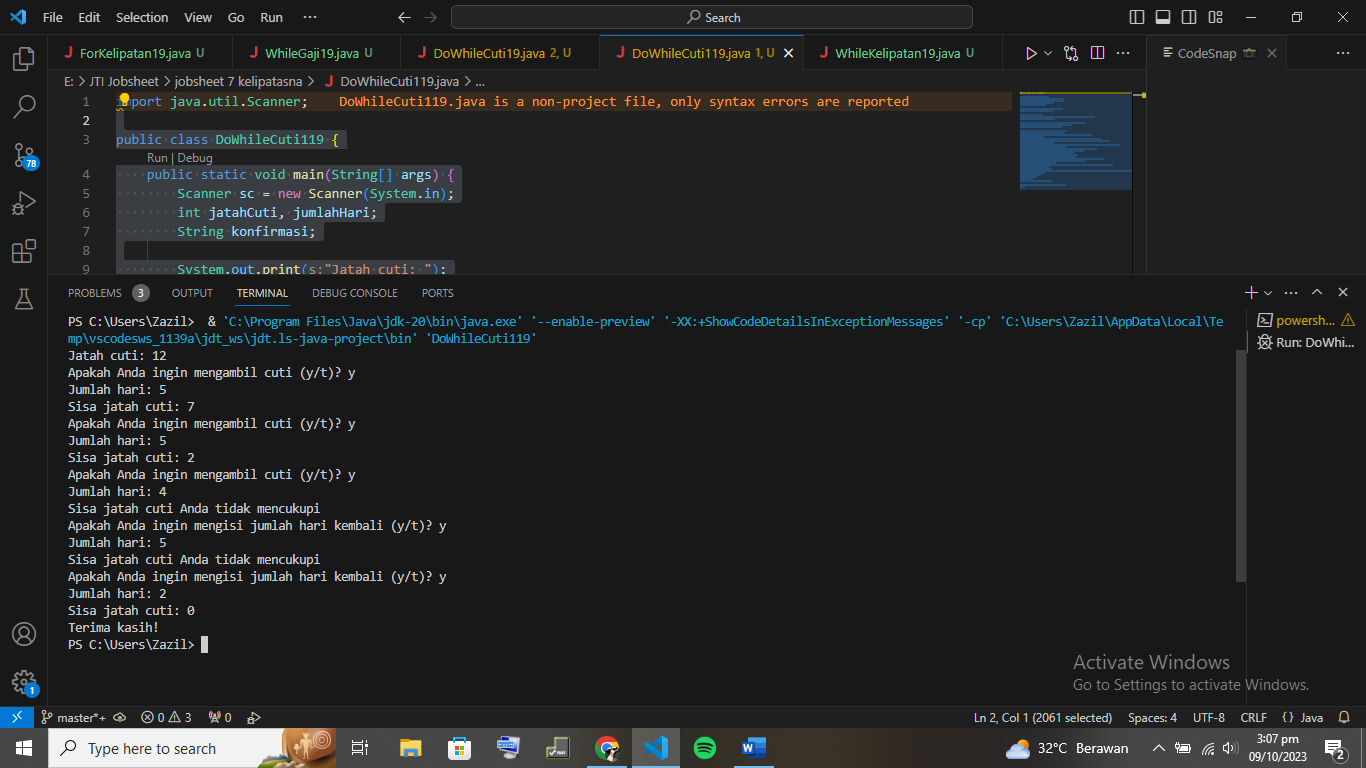
        } while (jatahCuti > 0);

        System.out.println("Terima kasih!");

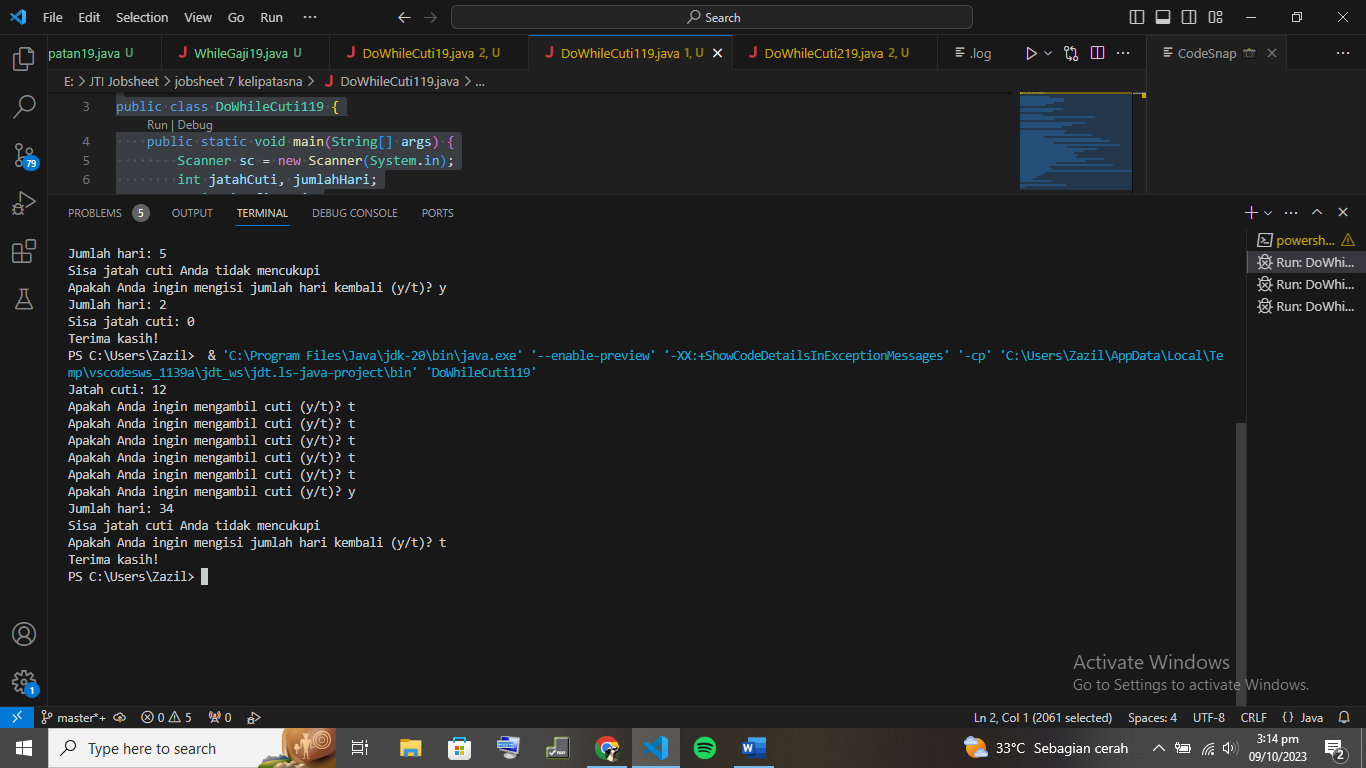
    }

}

Output



1. Push dan commit kode program ke github
2. Pada saat input konfirmasi, ketikkan “t”, apa yang terjadi? Mengapa demikian?



Jika user input t maka program akan melanjutkan ke Langkah selanjutnya sesuai dengan alur looping. Ini karena t tidaksaa dengan y yang diharapkan dalam percabangan if

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

import java.util.Scanner;

public class DoWhileCuti219 {

    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;  // Lanjutkan ke iterasi berikutnya jika sisa jatah cuti tidak mencukupi

                }

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

                break;  // Keluar dari loop jika pengguna memilih untuk tidak mengambil cuti

            } else {

                System.out.println("Input tidak valid. Silakan masukkan y atau t.");

            }

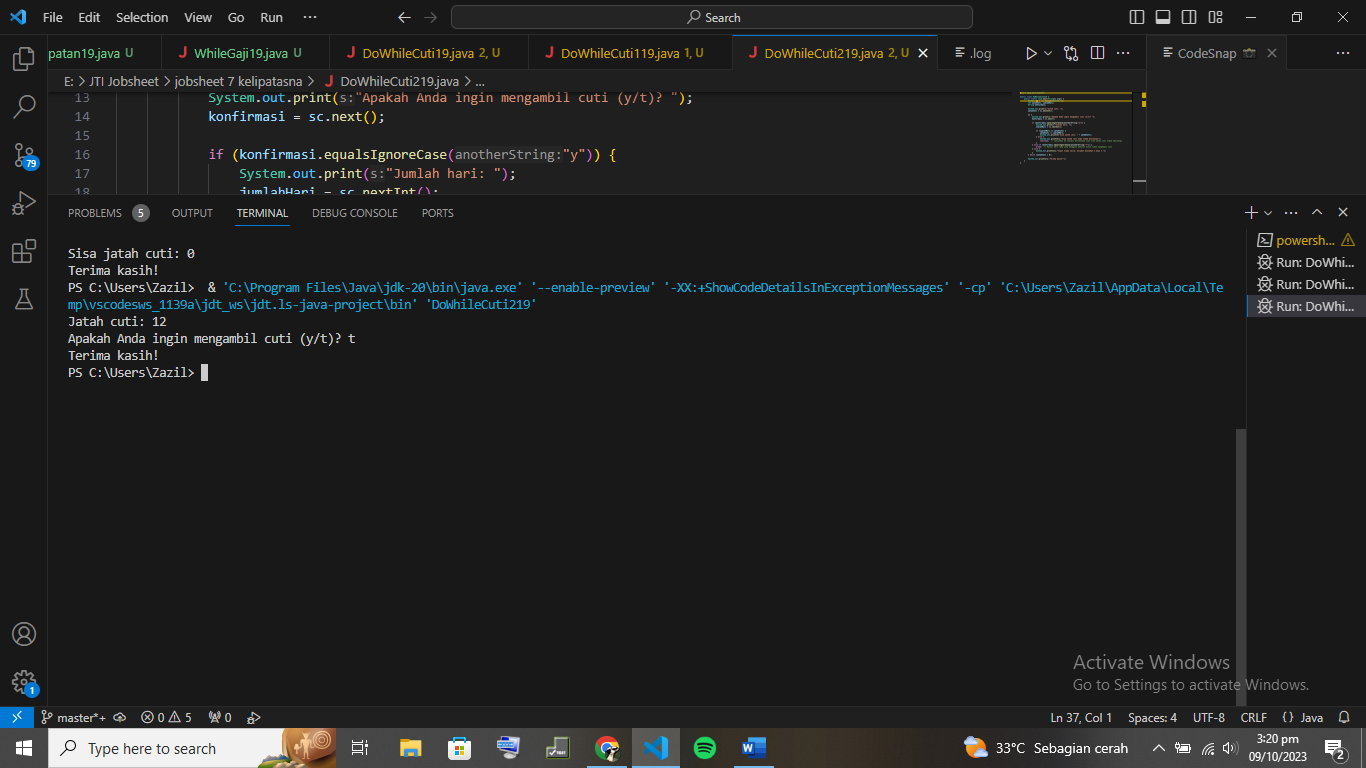
        } while (jatahCuti > 0);

        System.out.println("Terima kasih!");

    }

}

Output



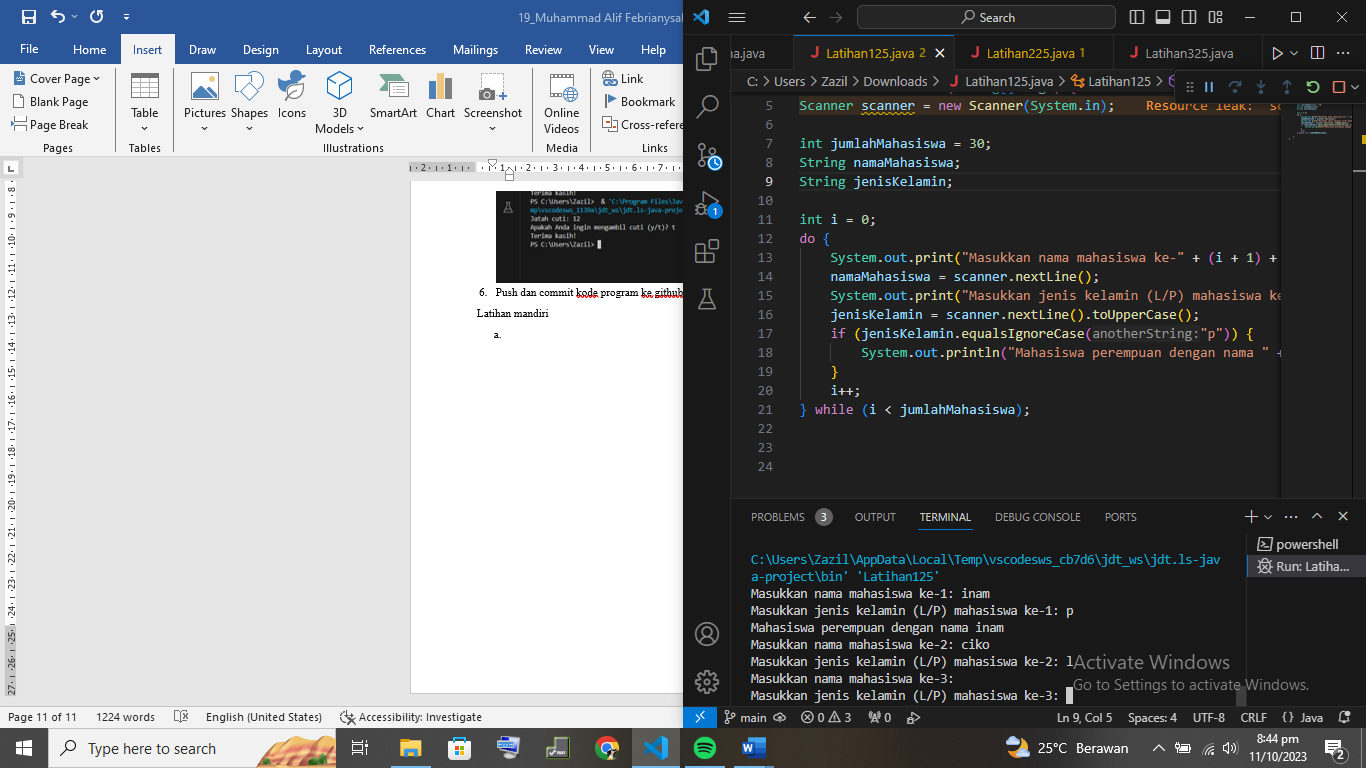
1. Push dan commit kode program ke github

Latihan mandiri

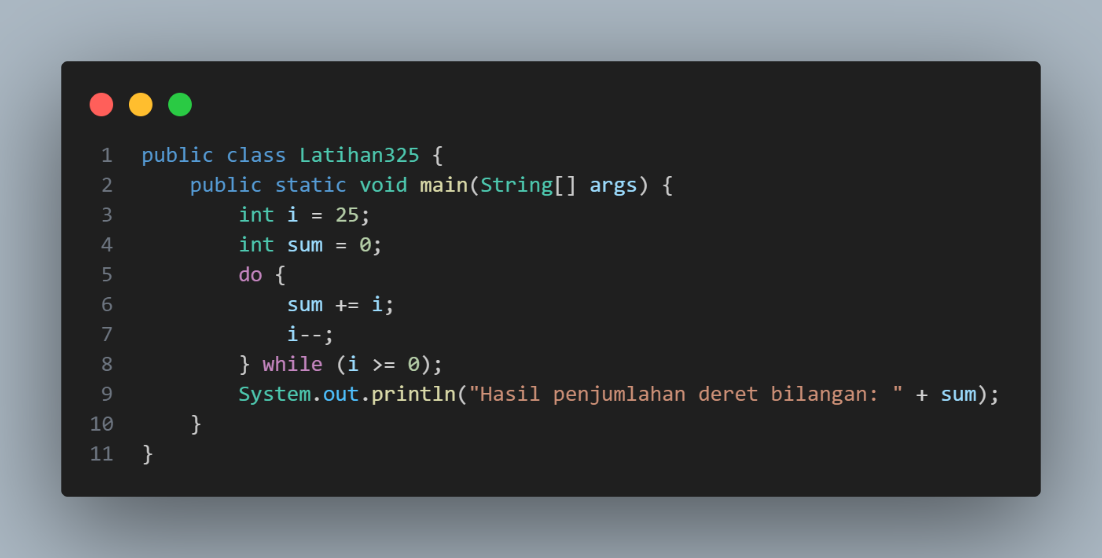
* 1. Pengguna memasukkan nama dan jenis kelamin dari 30 mahasiswa di suatu kelas. Nama-nama mahasiswa yang ditampilkan hanya yang berjenis kelamin Perempuan



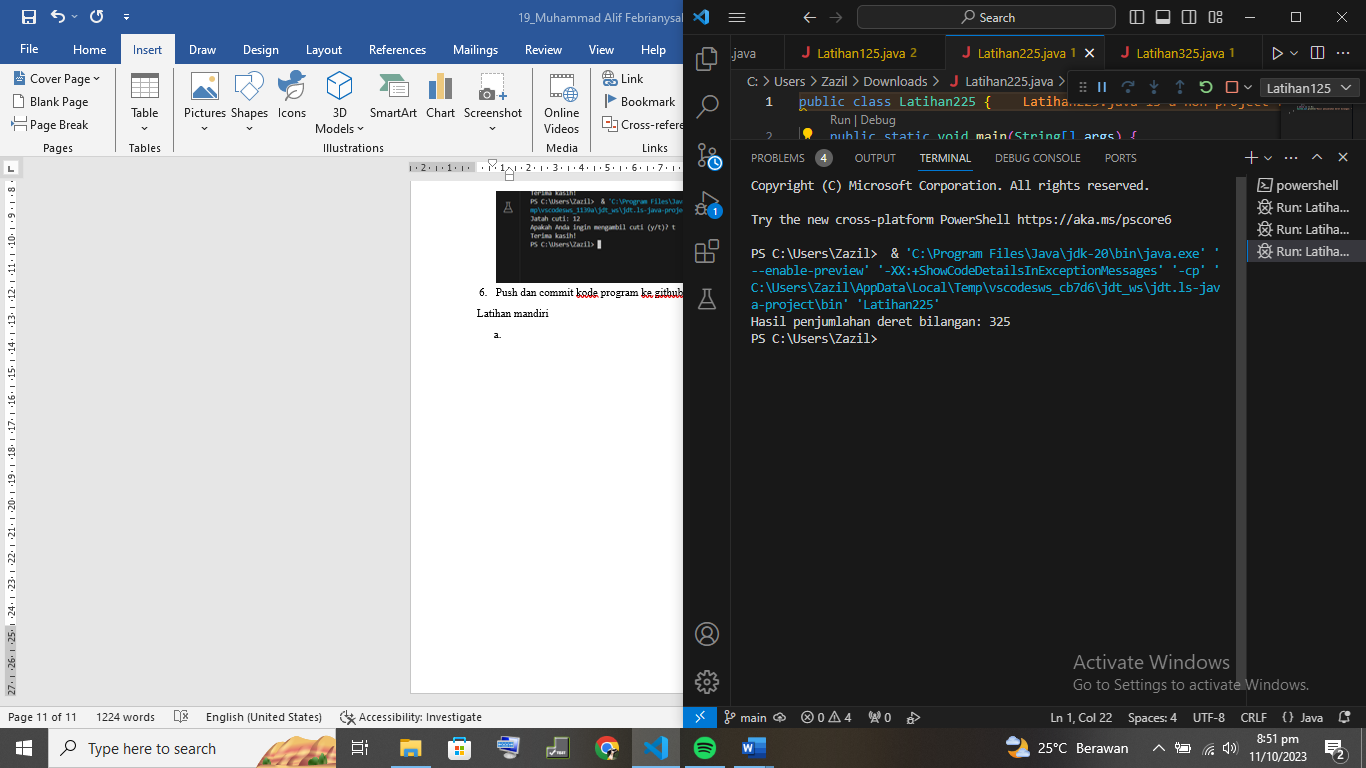
Output



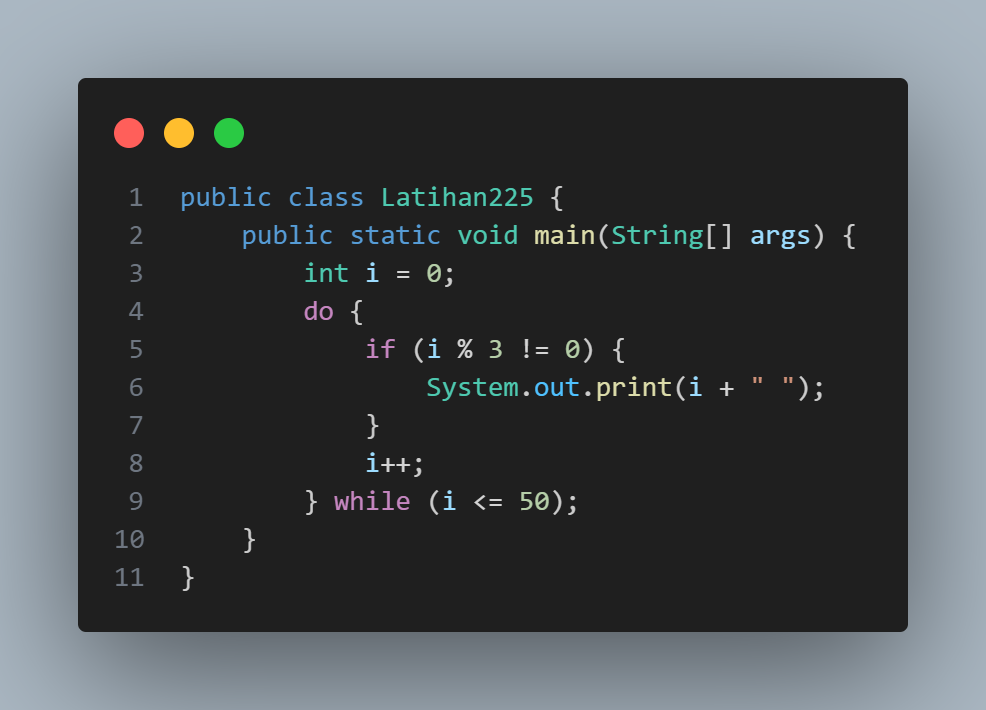
* 1. Menampilkan hasil penjumlahan deret bilangan 25 sampai dengan 1



Output



* 1. Menampilkan deret bilangan 1 sampai 50, kecuali bilangan kelipatan 3 (1 2 4 5 7 8 10 … 47 49 50)



Output

