**LAPORAN HASIL PRAKTIKUM ALSD**

**JOBSHEET 10 QUEUE**



Faiva Puspa Sahara

244107020036

TI – 1E

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

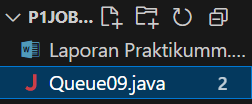
POLITEKNIK NEGERI MALANG

2025

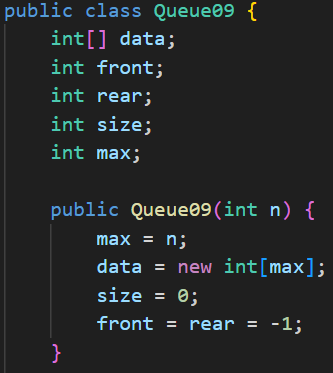
**HASIL PRAKTIKUM**

* 1. **Percobaan 1 (Mahasiswa Mengumpulkan Tugas)**

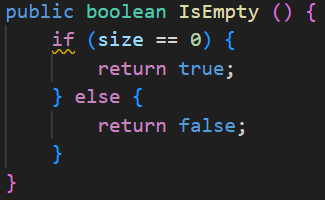
1. Buat folder baru bernama **P1Jobsheet10**. Buat file baru, beri nama **Queue09.java**



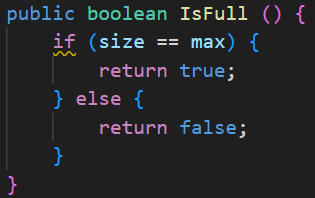
1. Tambahkan atribut-atribut Queue sesuai diagram class, kemudian tambahkan pula konstruktornya



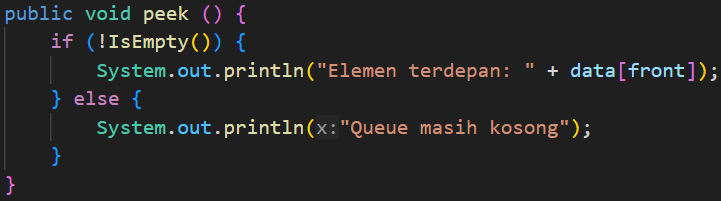
1. Buat method **IsEmpty** bertipe boolean yang digunakan untuk mengecek apakah queue kosong.



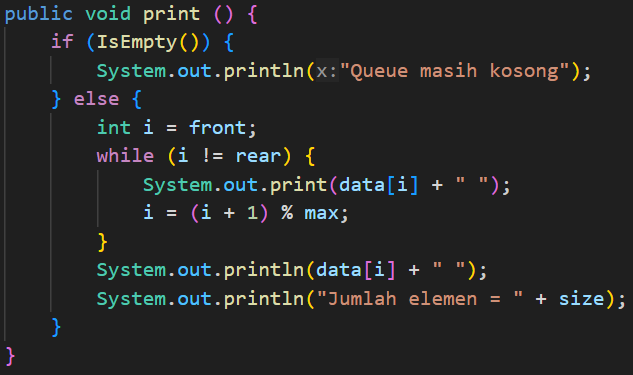
1. Buat method **IsFull** bertipe boolean yang digunakan untuk mengecek apakah queue sudah penuh.



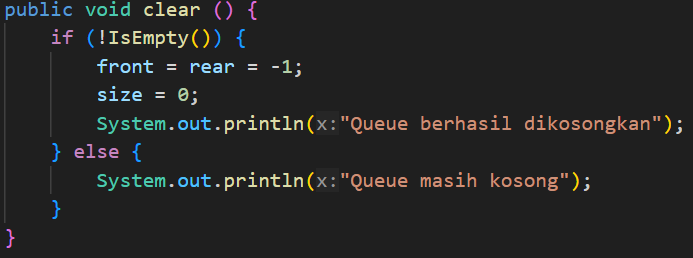
1. Buat method **peek** bertipe void untuk menampilkan elemen queue pada posisi paling depan.



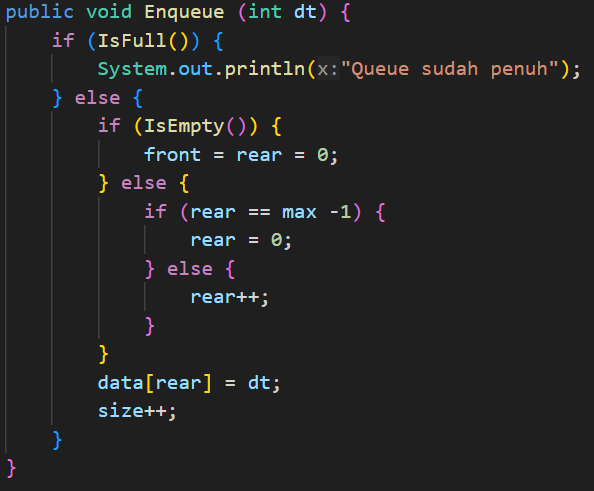
1. Buat method **print** bertipe void untuk menampilkan seluruh elemen pada queue mulai dari posisi front sampai dengan posisi rear.



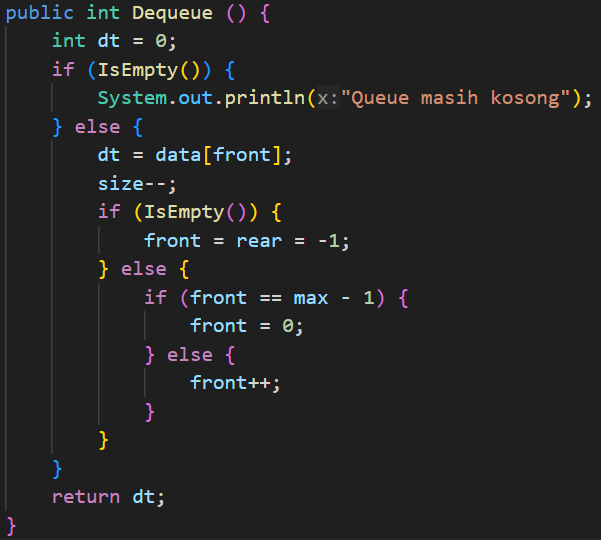
1. Buat method **clear** bertipe void untuk menghapus semua elemen pada queue



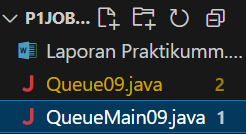
1. Buat method **Enqueue** bertipe void untuk menambahkan isi queue dengan parameter **dt** yang bertipe integer

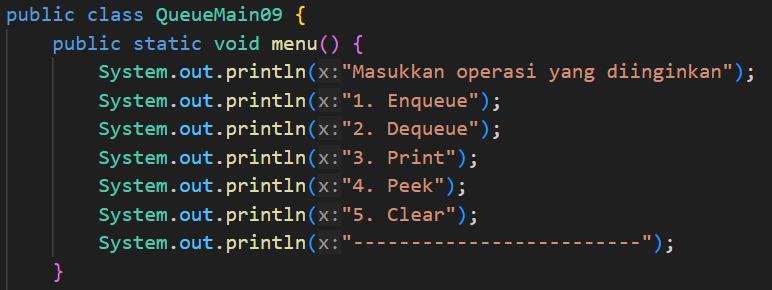


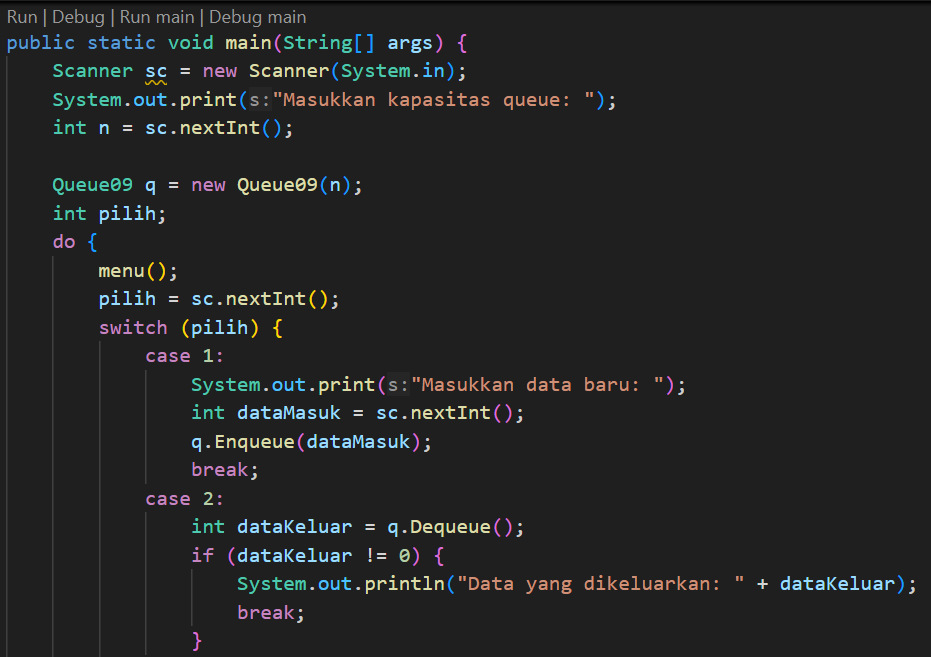
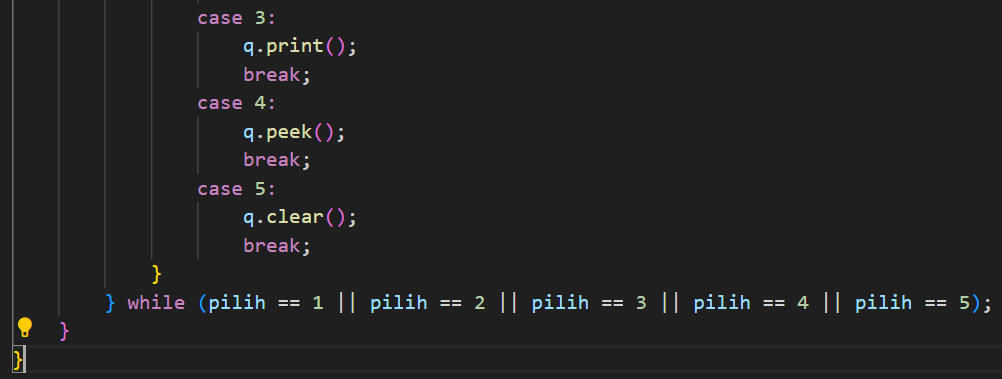
1. Buat method **Dequeue** bertipe int untuk mengeluarkan data pada queue di posisi belakang



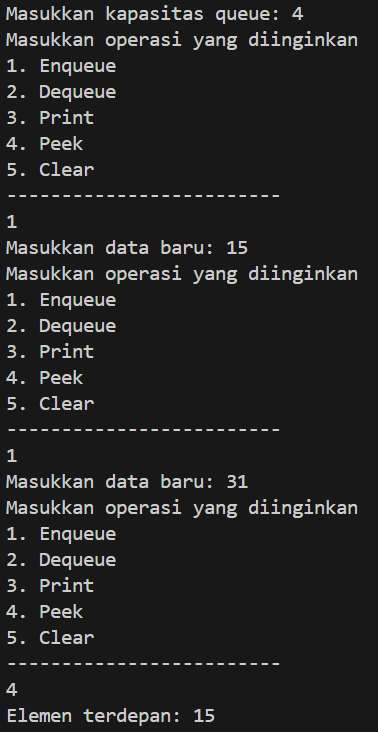
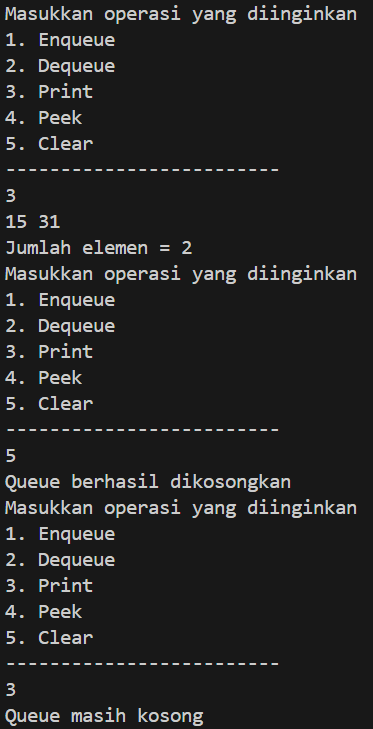
1. Selanjutnya, buat class baru dengan nama **QueueMain** tetap pada package **Praktikum1**. Buat method **menu** bertipe void untuk memilih menu program pada saat dijalankan.

* 



* y
* y
* y

1. Run Program

1. Commit dan push ke Github

* **Pertanyaan**

1. Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan sama dengan verifikasi hasil percobaan! Bagian mana yang perlu diperbaiki?

* Kode Programnya

public void print() {

for (int i = 0; i >= 0; i++) {

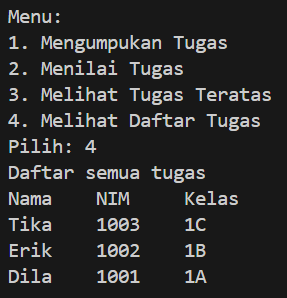
System.out.println(stack[i].nama + "\t" + stack[i].nim + "\t" + stack[i].kelas);

}

System.out.println("");

}

* Outputnya



1. Berapa banyak data tugas mahasiswa yang dapat ditampung di dalam Stack? Tunjukkan potongan kode programnya!

* Banyaknya data tugas mahasiswa yag ditampung, maksimal ada 5.

StackTugasMahasiswa09 stack = new StackTugasMahasiswa09(5);

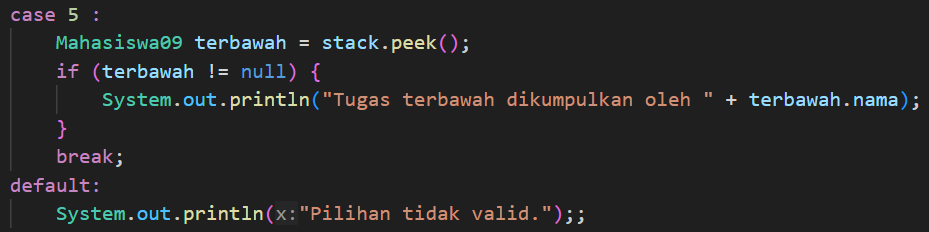
1. Mengapa perlu pengecekan kondisi **!isFull()** pada method **push**? Kalau kondisi if-else tersebut dihapus, apa dampaknya?

* Kalau pengecekan kondisi !**isFull** dihapus, dampaknya bisa fatal. Tanpa batasan, program akan tetap mencoba menambahkan elemen meskipun kapasitas sudah penuh.

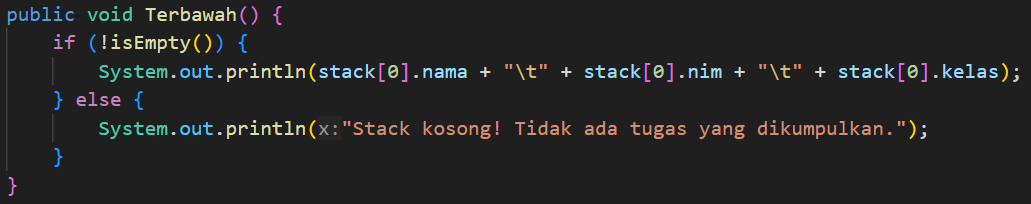
1. Modifikasi kode program pada class **MahasiswaDemo** dan **StackTugasMahasiswa** sehingga pengguna juga dapat melihat mahasiswa yang pertama kali mengumpulkan tugas melalui operasi lihat tugas terbawah!

* Kode program – pada class **MahasiswaDemo09** tambahkan menu 5 dan tambahkan case 5 di switch

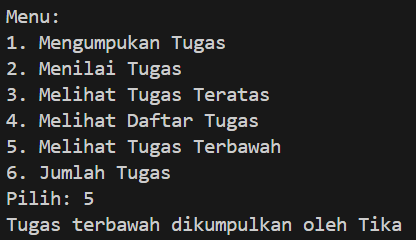




* Kode program – pada class **StackTugasMahasiswa09** tambakan method **void Terbawah**



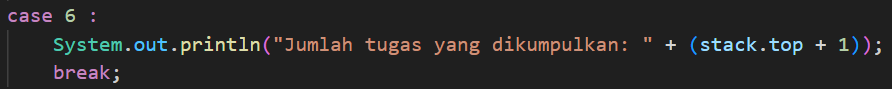
* Outputnya



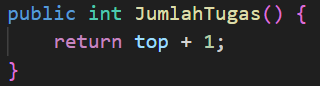
1. Tambahkan method untuk dapat menghitung berapa banyak tugas yang sudah dikumpulkan saat ini, serta tambahkan operasi menunya!

* Kode program – pada class **MahasiswaDemo09** tambahkan menu 6 dan tambahkan case 6 di switch

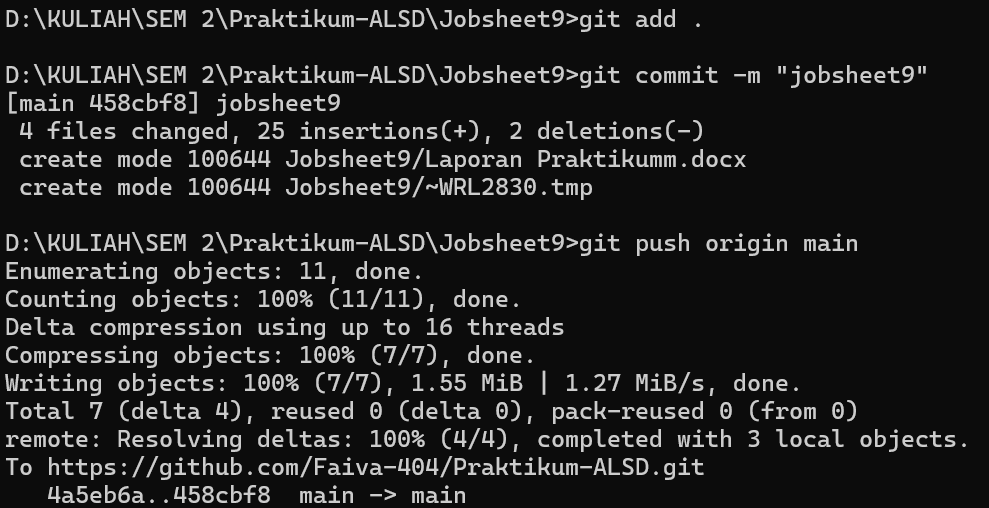




* Kode program – pada class **StackTugasMahasiswa09** tambakan method **JumlahTugas**

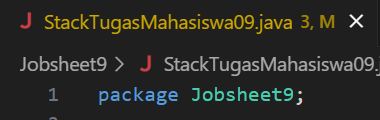


1. **Commit dan push kode program ke Github**



* 1. **Percobaan 2 (Konversi Nilai Tugas ke Biner)**

1. Buka kembali file **StackTugasMahasiswa09.java**



1. Tambahkan method **konversiDesimalKeBiner** dengan menerima parameter **kode** bertipe int

public String konversiDesimalKeBiner(int nilai) {

StackKonversi09 stack = new StackKonversi09();

while (nilai > 0) {

int sisa = nilai % 2;

stack.push(sisa);

nilai = nilai / 2;

}

String biner = new String();

while (!stack.isEmpty()) {

biner += stack.pop();

}

return biner;

}

1. Tambahkan empat method yaitu **isEmpty**, **isFull**, **push**, dan **pull** sebagai operasi utama Stack pada class **StackKonversi09**

package Jobsheet9;

public class StackKonversi09 {

int[] tumpukanBiner;

int size;

int top;

public StackKonversi09() {

this.size = 32;

tumpukanBiner = new int[size];

top = -1;

}

public boolean isEmpty() {

return top == -1;

}

public boolean isFull() {

return top == size - 1;

}

public void push(int data) {

if (!isFull()) {

top++;

tumpukanBiner[top] = data;

} else {

System.out.println("Stack penuh! Tidak bisa menambahkan data lagi.");

}

}

public int pop() {

if (!isEmpty()) {

int data = tumpukanBiner[top];

top--;

return data;

} else {

System.out.println("Stack kosong! Tidak ada data untuk diambil.");

return -1;

}

}

}

public int pop() {

if (!isEmpty()) {

int data = tumpukanBiner[top];

top--;

return data;

} else {

System.out.println("Stack kosong! Tidak ada data untuk diambil.");

return -1;

}

}

}

1. Agar nilai tugas mahasiswa dikonversi ke dalam bentuk biner setelah dilakukan penilaian, maka tambahkan baris kode program pada method **pop** di class **MahasiswaDemo09**

case 2 :

Mahasiswa09 dinilai = stack.pop();

if (dinilai != null) {

System.out.println("Menilai tugas dari " + dinilai.nama);

System.out.print("Masukkan nilai (0-100): ");

int nilai = sc.nextInt();

dinilai.tugasDinilai(nilai);

System.out.printf("Nilai Tugas %s adalah %d\n", dinilai.nama, nilai);

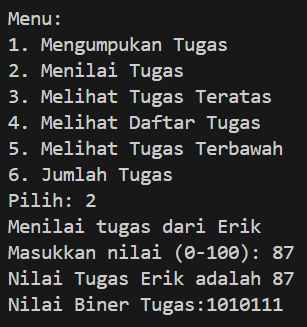
String biner = stack.konversiDesimalKeBiner(nilai);

System.out.println("Nilai Biner Tugas:" + biner);

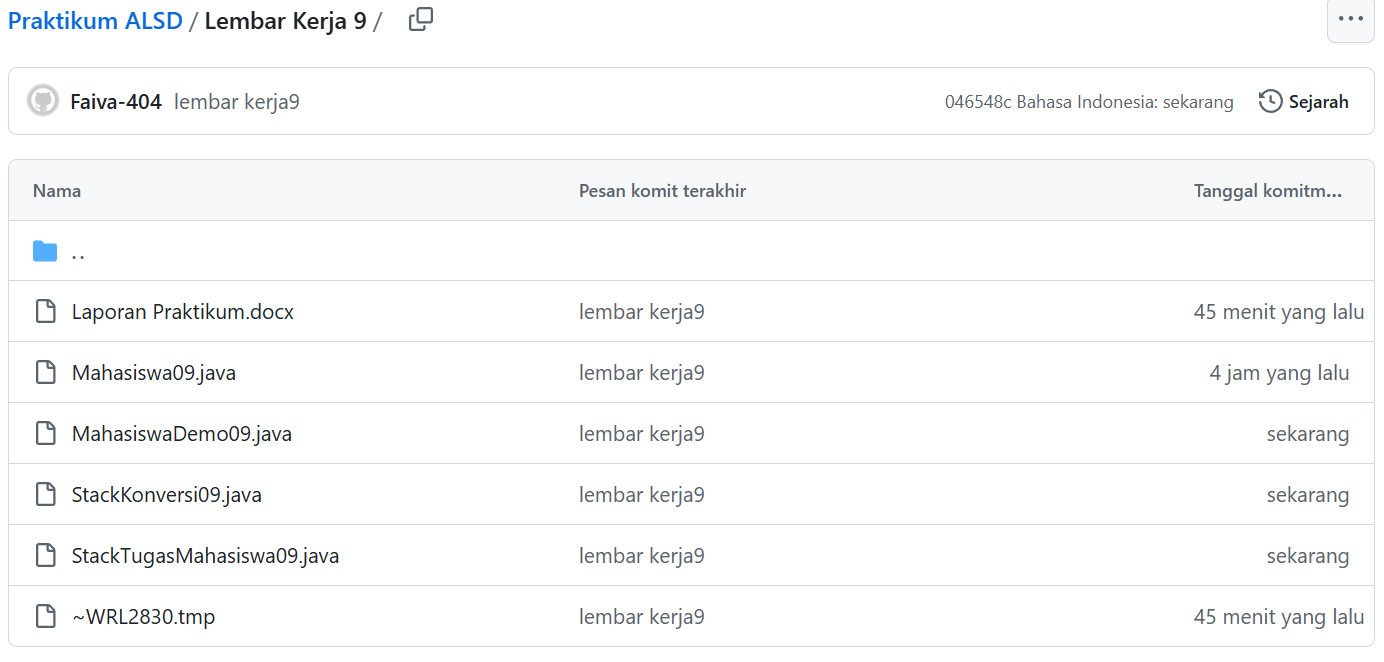
}

break;

1. Run Program



1. Commit dan Push ke Github



* **Pertanyaan**

1. Jelaskan alur kerja dari method **konversiDesimalKeBiner**!

* **Membagi angka secara berulang dengan 2** sambil mencatat **sisa hasil bagi** di setiap langkah. Sisa pembagian ini nantinya akan membentuk angka biner.

1. Pada method **konversiDesimalKeBiner**, ubah kondisi perulangan menjadi **while (kode != 0)**, bagaimana hasilnya? Jelaskan alasannya!

* Kode Programnya

public String konversiDesimalKeBiner(int nilai) {

StackKonversi09 stack = new StackKonversi09();

while (kode != 0) {

int sisa = nilai % 2;

stack.push(sisa);

nilai = nilai / 2;

}

String biner = new String();

while (!stack.isEmpty()) {

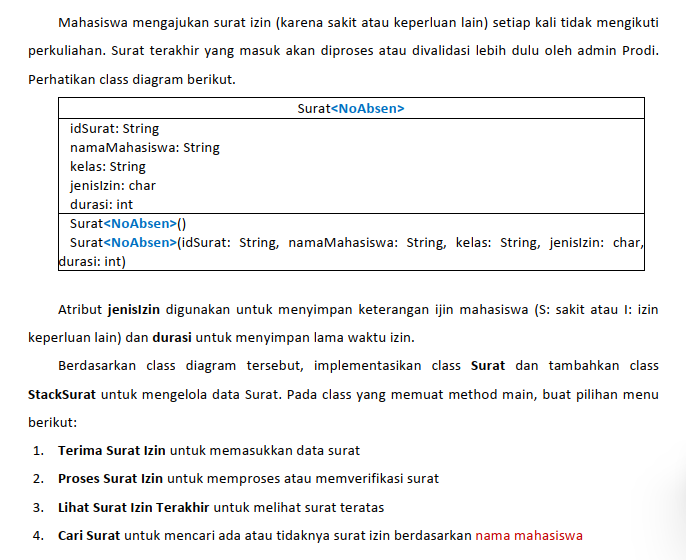
biner += stack.pop();

}

return biner;

}

* Penjelasan :
* program akan mengalami error karena variabel **kode** **tidak dideklarasikan atau tidak ada dalam metode.**

* 1. **Latihan Praktikum**
* **Class Surat09.java**

package Jobsheet9;

public class Surat09 {

String idSurat;

String namaMahasiswa;

String kelas;

char jenisIzin;

int durasi;

public Surat09() {

}

public Surat09(String idSurat, String namaMahasiswa, String kelas, char jenisIzin, int durasi) {

this.idSurat = idSurat;

this.namaMahasiswa = namaMahasiswa;

this.kelas = kelas;

this.jenisIzin = jenisIzin;

this.durasi = durasi;

}

}

* **Class StackSurat09.java**

package Jobsheet9;

public class StackSurat09 {

Surat09[] stack;

int size;

int top;

public StackSurat09(int size) {

this.size = size;

stack = new Surat09[size];

top = -1;

}

public boolean isFull() {

return top == size - 1;

}

public boolean isEmpty() {

return top == -1;

}

public void push(Surat09 surat) {

if (!isFull()) {

top++;

stack[top] = surat;

} else {

System.out.println("Stack penuh! Tidak bisa menambahkan surat lagi.");

}

}

public Surat09 pop() {

if (!isEmpty()) {

Surat09 surat = stack[top];

top--;

return surat;

} else {

System.out.println("Stack kosong! Tidak ada surat untuk diambil.");

return null;

}

}

public Surat09 peek() {

if (!isEmpty()) {

return stack[top];

} else {

System.out.println("Stack kosong! Tidak ada surat yang dikumpulkan.");

return null;

}

}

public boolean search(String namaMahasiswa) {

for (int i = top; i >= 0; i--) {

if (stack[i].namaMahasiswa.equals(namaMahasiswa)) {

return true;

}

}

return false;

}

}

System.out.println("Stack kosong! Tidak ada surat untuk diambil.");

return null;

}

}

public Surat09 peek() {

if (!isEmpty()) {

return stack[top];

} else {

System.out.println("Stack kosong! Tidak ada surat yang dikumpulkan.");

return null;

}

}

public boolean search(String namaMahasiswa) {

for (int i = top; i >= 0; i--) {

if (stack[i].namaMahasiswa.equals(namaMahasiswa)) {

return true;

}

}

return false;

}

}

* **Class SuratDemo09.java**

package Jobsheet9;

import java.util.Scanner;

public class SuratDemo09 {

public static void main(String[] args) {

StackSurat09 stack = new StackSurat09(5);

Scanner sc = new Scanner(System.in);

int pilih;

do {

System.out.println("\nMenu: ");

System.out.println("1. Terima Surat Izin");

System.out.println("2. Proses Surat Izin");

System.out.println("3. Lihat Surat Izin Terakhir");

System.out.println("4. Cari Surat Izin");

System.out.print("Pilih : ");

pilih = sc.nextInt();

sc.nextLine();

switch (pilih) {

case 1 :

if (stack.isFull()) {

System.out.println("Stack penuh! Tidak bisa menambahkan surat lagi.");

} else {

System.out.print("ID Surat : ");

String idSurat = sc.nextLine();

System.out.print("Nama Mahasiswa : ");

String namaMahasiswa = sc.nextLine();

System.out.print("Kelas : ");

String kelas = sc.nextLine();

System.out.print("Jenis Izin (S/I): ");

char jenisIzin = sc.next().charAt(0);

System.out.print("Durasi Izin : ");

int durasi = sc.nextInt();

Surat09 surat = new Surat09(idSurat, namaMahasiswa, kelas, jenisIzin, durasi);

stack.push(surat);

System.out.println("Surat izin dari " + surat.namaMahasiswa + " berhasil diterima.");

}

break;

case 2 :

Surat09 surat = stack.pop();

if (surat != null) {

System.out.println("Surat izin dari " + surat.namaMahasiswa + " berhasil diproses.");

}

break;

case 3 :

Surat09 suratTerakhir = stack.peek();

if (suratTerakhir != null) {

System.out.println("Surat izin terakhir: ");

System.out.println("ID Surat : " + suratTerakhir.idSurat);

System.out.println("Nama Mahasiswa: " + suratTerakhir.namaMahasiswa);

System.out.println("Jenis Izin : " + suratTerakhir.jenisIzin + " (S/I)");

System.out.println("Durasi Izin : " + suratTerakhir.durasi + " hari");

}

break;

case 4 :

System.out.print("Nama Mahasiswa yang dicari: ");

String cariNama = sc.nextLine();

boolean ditemukan = stack.search(cariNama);

if (ditemukan) {

System.out.println("Surat izin untuk " + cariNama + " ditemukan.");

} else {

System.out.println("Surat izin untuk " + cariNama + " tidak ditemukan.");

}

break;

default:

System.out.println("Pilihan tidak valid. Silakan coba lagi.");

break;

}

} while (pilih >= 1 && pilih <= 4);

}

}

char jenisIzin = sc.next().charAt(0);

System.out.print("Durasi Izin : ");

int durasi = sc.nextInt();

Surat09 surat = new Surat09(idSurat, namaMahasiswa, kelas, jenisIzin, durasi);

stack.push(surat);

System.out.println("Surat izin dari " + surat.namaMahasiswa + " berhasil diterima.");

}

break;

case 2 :

Surat09 surat = stack.pop();

if (surat != null) {

System.out.println("Surat izin dari " + surat.namaMahasiswa + " berhasil diproses.");

}

break;

case 3 :

Surat09 suratTerakhir = stack.peek();

if (suratTerakhir != null) {

System.out.println("Surat izin terakhir: ");

System.out.println("ID Surat : " + suratTerakhir.idSurat);

System.out.println("Nama Mahasiswa: " + suratTerakhir.namaMahasiswa);

System.out.println("Jenis Izin : " + suratTerakhir.jenisIzin + " (S/I)");

System.out.println("Durasi Izin : " + suratTerakhir.durasi + " hari");

}

break;

case 4 :

System.out.print("Nama Mahasiswa yang dicari: ");

String cariNama = sc.nextLine();

boolean ditemukan = stack.search(cariNama);

if (ditemukan) {

System.out.println("Surat izin untuk " + cariNama + " ditemukan.");

} else {

System.out.println("Surat izin untuk " + cariNama + " tidak ditemukan.");

}

break;

default:

System.out.println("Pilihan tidak valid. Silakan coba lagi.");

break;

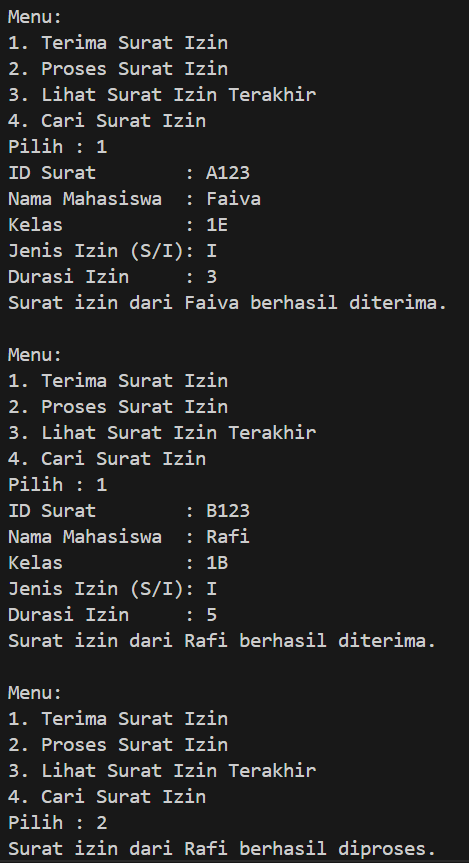
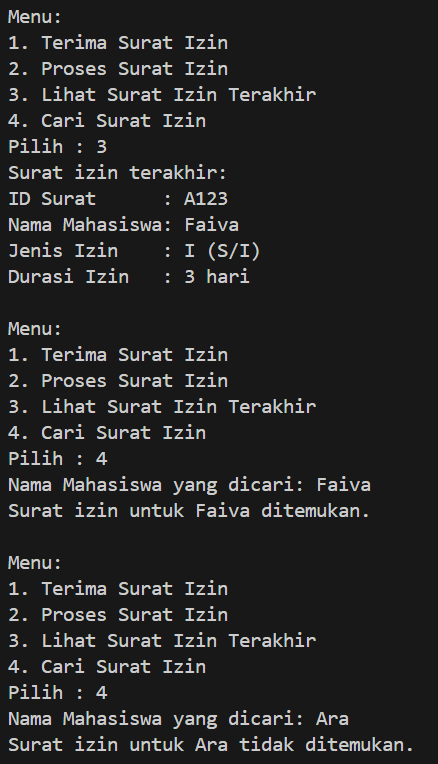
}

} while (pilih >= 1 && pilih <= 4);

}

}

* Run Program

* Commit dan Push ke Github

