**POLITEKNIK NEGERI MALANG**

**TEKNOLOGI INFORMASI**

**TEKNIK INFORMATIKA**

****

**Mohammad Ariq Baihaqi**

**244107020161**

**TI – 1A**

**16**

**2.1.1 Langkah-langkah Percobaan**

package Minggu10;

public class Mahasiswa16 {

    String nim;

    String nama;

    String kelas;

    int nilai;

    Mahasiswa16(String nama, String nim, String kelas) {

        this.nama = nama;

        this.nim = nim;

        this.kelas = kelas;

    }

    void tugasDinilai(int nilai) {

        this.nilai = nilai;

    }

}

package Minggu10;

public class StackTugasMahasiswa16 {

    Mahasiswa16[] stack;

    int top;

    int size;

    public StackTugasMahasiswa16(int size) {

        this.size = size;

        stack = new Mahasiswa16[size];

        top = -1;

    }

    public boolean isFull() {

       if (top == size - 1) {

            return true;

        } else {

            return false;

        }

    }

    public boolean isEmpty() {

        if (top == -1) {

            return true;

        } else {

            return false;

        }

    }

 public void push(Mahasiswa16 m) {

        if (!isFull()) {

            top++;

            stack[top] = m;

        } else {

            System.out.println("Stack penuh! Tdiak bisa menambahkan tugas lagi");

        }

    }

    public Mahasiswa16 pop() {

        if (!isEmpty()) {

            Mahasiswa16 m = stack[top];

            top--;

            return m;

        } else {

            System.out.println("Stack kosong! Tdiak ada tugas untuk dinilai");

            return null;

        }

    }

    public Mahasiswa16 peek() {

        if (!isEmpty()) {

            return stack[top];

        } else {

            System.out.println("Stack kosong! Tdiak ada tugas untuk dikumpulkan");

            return null;

        }

    }

    public void print() {

        for (int i = 0; i <= top; i++) {

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

        }

        System.out.println("");

    }

}

public Mahasiswa16 pop() {

        if (!isEmpty()) {

            Mahasiswa16 m = stack[top];

            top--;

            return m;

        } else {

            System.out.println("Stack kosong! Tdiak ada tugas untuk dinilai");

            return null;

        }

    }

    public Mahasiswa16 peek() {

        if (!isEmpty()) {

            return stack[top];

        } else {

            System.out.println("Stack kosong! Tdiak ada tugas untuk dikumpulkan");

            return null;

        }

    }

    public void print() {

        for (int i = 0; i <= top; i++) {

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

        }

        System.out.println("");

    }

}

package Minggu10;

import java.util.Scanner;

public class MahasiswaDemo16 {

    public static void main(String[] args) {

        StackTugasMahasiswa16 stack = new StackTugasMahasiswa16(5);

        Scanner scan = new Scanner(System.in);

        int pilih;

        do {

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

            System.out.println("1. Mengumpulkan Tugas");

            System.out.println("2. Menilai Tugas");

            System.out.println("3. Melihat Tugas Teratas");

            System.out.println("4. Melihat Daftar Tugas");

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

            pilih = scan.nextInt();

            scan.nextLine();

            switch (pilih) {

                case 1:

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

                    String nama = scan.nextLine();

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

                    String nim = scan.nextLine();

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

                    String kelas = scan.nextLine();

                    Mahasiswa16 mhs = new Mahasiswa16(nama, nim, kelas);

                    stack.push(mhs);

                    System.out.printf("Tugas %s berhasil dikumpulkan\n", mhs.nama);

                    break;

                case 2:

                    Mahasiswa16 dinilai = stack.pop();

                    if (dinilai != null) {

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

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

                        int nilai = scan.nextInt();

                        dinilai.tugasDinilai(nilai);

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

                    }

                    break;

                case 3:

                    Mahasiswa16 lihat = stack.peek();

                    if (lihat != null) {

                        System.out.println("Tugas terakhir dikumpulkan oleh " + lihat.nama);

                    }

                    break;

                case 4:

                    System.out.println("Daftar semua tugas");

                    System.out.println("Nama\tNIM\tKelas");

                    stack.print();

                    break;

                default:

                    System.out.println("Pilihan tidak valid.");

            }

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

    }

}

dinilai.tugasDinilai(nilai);

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

                    }

                    break;

                case 3:

                    Mahasiswa16 lihat = stack.peek();

                    if (lihat != null) {

                        System.out.println("Tugas terakhir dikumpulkan oleh " + lihat.nama);

                    }

                    break;

                case 4:

                    System.out.println("Daftar semua tugas");

                    System.out.println("Nama\tNIM\tKelas");

                    stack.print();

                    break;

                default:

                    System.out.println("Pilihan tidak valid.");

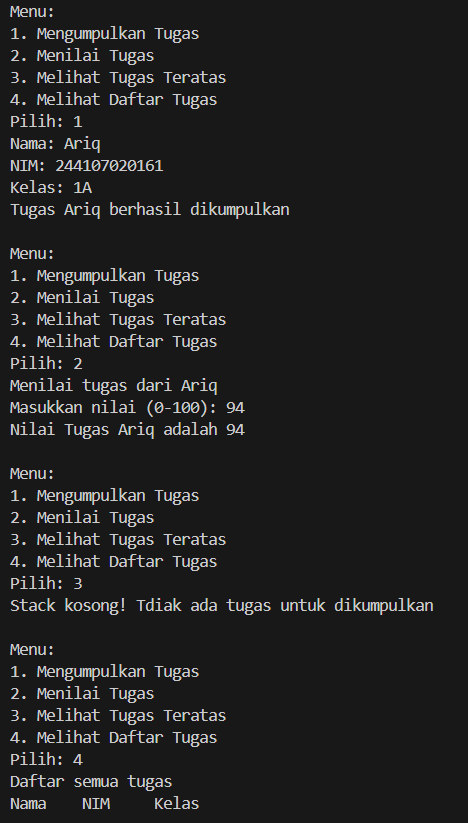
            }

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

    }

}

**OUTPUT**



**2.1.3 Pertanyaan**

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

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

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

* Stack dapat menampung 5 tugas Mahasiswa
* StackTugasMahasiswa16 stack = new StackTugasMahasiswa16(5);

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

* Untuk mencegah menambahkan elemen ketika stack sudah penuh
* Jika dihapus bisa crash saat pengguna mencoba menambahkan lebih banyak elemen daripada kapasitas stack

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!

public Mahasiswa16 bottom() {

        if (!isEmpty()) {

            return stack[0];

        } else {

            System.out.println("Stack kosong! Tidak ada tugas.");

            return null;

        }

    }

System.out.println("5. Melihat Tugas Terbawah");

case 5:

                    Mahasiswa16 bawah = stack.bottom();

                    if (bawah != null) {

                        System.out.println("Tugas pertama dikumpulkan oleh " + bawah.nama);

                    }

                    break;

1. Tambahkan method untuk dapat menghitung berapa banyak tugas yang

sudah dikumpulkan saat ini, serta tambahkan operasi menunya!

* public int jumlahTugas() {

return top + 1;

}

* public int jumlahTugas() {

return top + 1;

}

* System.out.println("6. Jumlah Tugas yang Dikumpulkan");

case 6:

System.out.println("Jumlah tugas yang sudah dikumpulkan: " + stack.jumlahTugas());

break;

1. Commit dan push kode program ke Github

**2.2.1 Langkah-langkah Percobaan**

**Method konversiDesimalKeBiner**

public String konversiDesimalKeBiner(int nilai) {

        StackKonversi16 stack = new StackKonversi16 ();

        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;

    }

**Class StackKoversi**

package Minggu10;

public class StackKonversi16 {

    int[] tumpukanBiner;

    int size;

    int top;

    public StackKonversi16() {

        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()) {

            System.out.println("Stack penuh");

        } else {

            top++;

            tumpukanBiner[top] = data;

        }

    }

    public int pop() {

        if (isEmpty()) {

            System.out.println("Stack kosong");

            return -1;

        } else {

            int data = tumpukanBiner[top];

            top--;

            return data;

        }

    }

}

public boolean isFull() {

        return top == size - 1;

    }

    public void push(int data) {

        if (isFull()) {

            System.out.println("Stack penuh");

        } else {

            top++;

            tumpukanBiner[top] = data;

        }

    }

    public int pop() {

        if (isEmpty()) {

            System.out.println("Stack kosong");

            return -1;

        } else {

            int data = tumpukanBiner[top];

            top--;

            return data;

        }

    }

}

**Class MahasiswaDemo**

case 2:

                    Mahasiswa16 dinilai = stack.pop();

                    if (dinilai != null) {

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

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

                        int nilai = scan.nextInt();

                        dinilai.tugasDinilai(nilai);

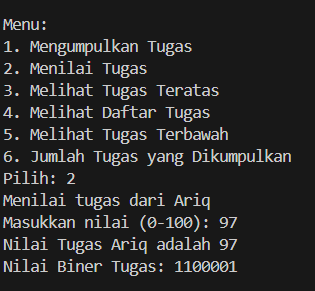
                        System.out.printf("Nilai Tugas %s adalah %d\n", dinilai.nama, nilai);

                        String biner = stack.konversiDesimalKeBiner(nilai);

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

                    }

**OUTPUT**



**2.2.3 Pertanyaan**

1. Jelaskan alur kerja dari method konversiDesimalKeBiner!

* Buat stack kosong untuk menyimpan hasil sisa pembagian 2
* Hitung nilai % 2
* Push hasil sisa ke dalam stack
* Bagi nilai dengan 2
* Ambil semua isi stack satu per
* Setiap pop, tambahkan ke String biner
* Pop mengeluarkan bit-bit dalam urutan yang benar dari atas ke bawah
* Kembalikan String ke biner

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

* Hasilnya akan sama, karena keduanya menghentikan lopp saat nilai menjadi nol
* Pada bilangan positif, nilai > 0 dan nilai != 0 artinya sama

**2.4 Latihan Praktikum**

**Class Surat**

package Minggu10;

public class Surat16 {

        String idSurat;

        String namaMahasiswa;

        String kelas;

        char jenisIzin;

        int durasi;

        public Surat16() {

        }

        public Surat16(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;

        }

        public void displaySurat() {

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

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

            System.out.println("Kelas        : " + kelas);

            System.out.println("Jenis Izin   : " + (jenisIzin == 'S' ? "Sakit" : "Izin"));

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

        }

    }

**Class StackSurat**

package Minggu10;

import java.util.Stack;

public class StackSurat16 {

    private Stack<Surat16> stackSurat = new Stack<>();

    public void terimaSurat(Surat16 surat) {

        stackSurat.push(surat);

    }

    public Surat16 prosesSurat() {

        if (!stackSurat.isEmpty()) {

            return stackSurat.pop();

        }

        return null;

    }

    public Surat16 lihatSuratTerakhir() {

        if (!stackSurat.isEmpty()) {

            return stackSurat.peek();

        }

        return null;

    }

    public Surat16 cariSurat(String namaMahasiswa) {

        for (int i = stackSurat.size() - 1; i >= 0; i--) {

            Surat16 s = stackSurat.get(i);

            if (s.namaMahasiswa.equalsIgnoreCase(namaMahasiswa)) {

                return s;

            }

        }

        return null;

    }

}

Class SuratDemo

package Minggu10;

import java.util.Scanner;

public class SuratDemo16 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        StackSurat16 stack = new StackSurat16();

        int pilihan;

        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 berdasarkan Nama Mahasiswa");

            System.out.println("0. Keluar");

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

            pilihan = sc.nextInt();

            sc.nextLine();

            switch (pilihan) {

                case 1:

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

                    String id = sc.nextLine();

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

                    String nama = sc.nextLine();

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

                    String kelas = sc.nextLine();

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

                    char jenis = sc.nextLine().toUpperCase().charAt(0);

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

                    int durasi = sc.nextInt();

                    Surat16 surat = new Surat16(id, nama, kelas, jenis, durasi);

                    stack.terimaSurat(surat);

                    System.out.println("Surat berhasil ditambahkan.");

                    break;

                case 2:

                    Surat16 diproses = stack.prosesSurat();

                    if (diproses != null) {

                        System.out.println("Surat berikut telah diproses:");

                        diproses.displaySurat();

                    } else {

                        System.out.println("Tidak ada surat untuk diproses.");

                    }

                    break;

                case 3:

                    Surat16 terakhir = stack.lihatSuratTerakhir();

                    if (terakhir != null) {

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

                        terakhir.displaySurat();

                    } else {

                        System.out.println("Belum ada surat.");

                    }

                    break;

                case 4:

                    System.out.print("Masukkan nama mahasiswa yang dicari: ");

                    String cariNama = sc.nextLine();

                    Surat16 ditemukan = stack.cariSurat(cariNama);

                    if (ditemukan != null) {

                        System.out.println("Surat ditemukan:");

                        ditemukan.displaySurat();

                    } else {

                        System.out.println("Surat tidak ditemukan.");

                    }

                    break;

                case 0:

                    System.out.println("Keluar dari program.");

                    break;

                default:

                    System.out.println("Pilihan tidak valid.");

            }

        } while (pilihan != 0);

    }

    }

break;

                case 2:

                    Surat16 diproses = stack.prosesSurat();

                    if (diproses != null) {

                        System.out.println("Surat berikut telah diproses:");

                        diproses.displaySurat();

                    } else {

                        System.out.println("Tidak ada surat untuk diproses.");

                    }

                    break;

                case 3:

                    Surat16 terakhir = stack.lihatSuratTerakhir();

                    if (terakhir != null) {

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

                        terakhir.displaySurat();

                    } else {

                        System.out.println("Belum ada surat.");

                    }

                    break;

                case 4:

                    System.out.print("Masukkan nama mahasiswa yang dicari: ");

                    String cariNama = sc.nextLine();

                    Surat16 ditemukan = stack.cariSurat(cariNama);

                    if (ditemukan != null) {

                        System.out.println("Surat ditemukan:");

                        ditemukan.displaySurat();

                    } else {

                        System.out.println("Surat tidak ditemukan.");

                    }

                    break;

                case 0:

                    System.out.println("Keluar dari program.");

                    break;

                default:

                    System.out.println("Pilihan tidak valid.");

            }

        } while (pilihan != 0);

    }

    }

case 0:

                    System.out.println("Keluar dari program.");

                    break;

                default:

                    System.out.println("Pilihan tidak valid.");

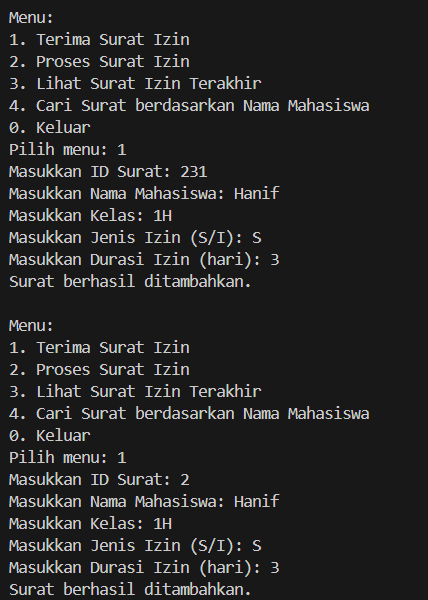
            }

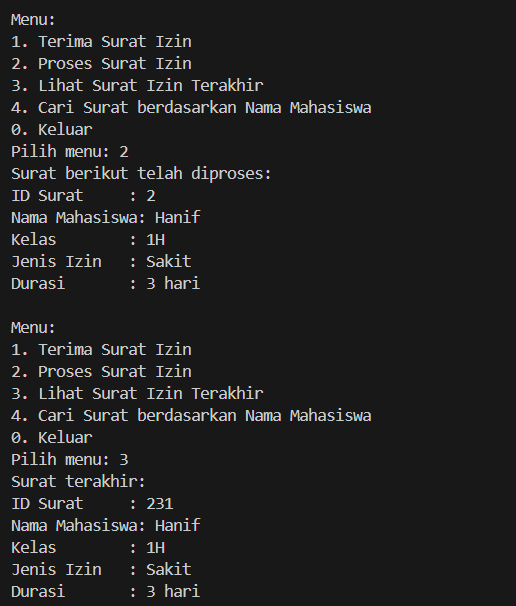
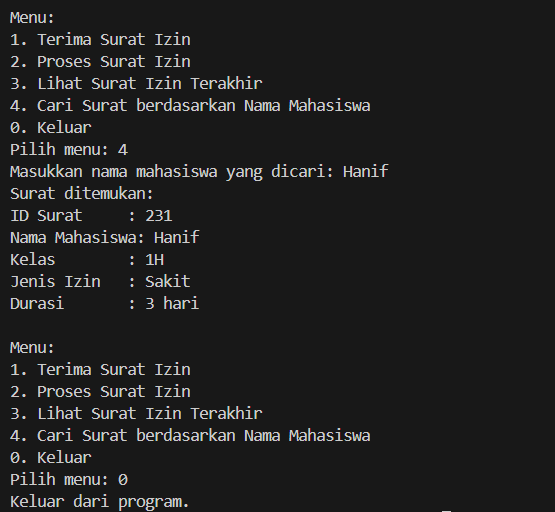
        } while (pilihan != 0);

    }

    }

**OUTPUT**

****

**** ****

**Link Github :** [**https://github.com/Ariqq16/semester2**](https://github.com/Ariqq16/semester2)