

LAPORAN JOBSHEET 9

KONSEP DASAR PEMROGRAMAN

Mata Kuliah : Algoritma dan Struktur Data

Dosen : **Mungki Astiningrum, S.T., M.Kom.**



Alfreda Dhaifullah Mahezwara

244107020219

Kelas : 1A

Absen : 04

**PROGRAM STUDI TEKNIK INFORMATIKA JURUSAN
TEKNOLOGI INFORMASI POLITEKNIK NEGERI
MALANG TAHUN 2025**

Percobaan 1

```
package Pertemuan_10;
```

```
public class Mahasiswa04 {  
    String nim;  
    String nama;  
    String kelas;  
    int nilai;  
  
    Mahasiswa04 () {  
  
    }  
  
    Mahasiswa04 (String nm, String nim, String kls) {  
        nama = nm;  
        this.nim = nim;  
        kelas = kls;  
        nilai = -1;  
    }  
  
    void tugasDinilai (int nilai) {  
        this.nilai = nilai;  
    }  
}
```

```
package Pertemuan_10;
```

```
public class StackTugasMahasiswa {  
    Mahasiswa04 [] stack;  
    int size;  
    int top;  
  
    public StackTugasMahasiswa (int size) {  
        this.size = size;  
        stack = new Mahasiswa04[size];  
        top = -1;  
    }  
  
    // method yang digunakan untuk mengecek apakah stack sudah terisi penuh sesuai kapasitas  
    public boolean isFull () {
```

```

        if (top == size-1) {
            return true;
        } else {
            return false;
        }
    }

}

// untuk mengecek apakah tumpukan masih kossong
public boolean isEmpty () {
    if (top == -1) {
        return true;
    } else {
        return false;
    }
}

// untuk menambah tugas ke tumpukan stack
public void push (Mahasiswa04 mhs) {
    if (!isFull()) {
        top++;
        stack[top] = mhs;
    } else {
        System.out.println("stack penuh! Tidak bisa menambahkan tugas lagi");
    }
}

// method untuk mengeluarkan tugas yang akan dinilai
public Mahasiswa04 pop () {
    if (!isEmpty()) {
        Mahasiswa04 m = stack [top];
        ;
        return m;
    } else {
        System.out.println("Stack kossong! Tidak ada tugas untuk dinilai.");
        return null;
    }
}

// untuk mengecek tumpukan tugas paling atas
public Mahasiswa04 peek () {

```

```

        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong! Tidak ada tugas yang dikumpulkan");
            return null;
        }
    }
}

// untuk menampilkan seluruh daftar tugas mahasiswa
public void print () {
    for (int i = 0; i <= top; i++) {
        System.out.println(stack[i].nama + "\t" + stack[i].nim + "\t" + stack[i].kelas);
    }
    System.out.println("");
}
}

```

```

package Pertemuan_10;

import java.util.Scanner;

public class MahasiswaDemo04 {

    public static void main(String[]args) {

        Scanner scan = new Scanner (System.in);

        int pick;

        StackTugasMahasiswa stack = new StackTugasMahasiswa(5);

        do {

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

            System.out.println("1. Megumpulkan 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: ");

            pick = scan.nextInt();

            scan.nextLine();

            switch (pick) {

                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();
        Mahasiswa04 mhs = new Mahasiswa04(nama, nim, kelas);
        stack.push(mhs);
        System.out.printf(" Tugas %s berhasil dikumpulkan \n", mhs.nama);
        break;
    case 2 :
        Mahasiswa04 dinilai = stack.pop();
        if (dinilai != null) {
            System.out.println("Menulai tugas daru " + dinilai.nama);
            System.out.println("Masukan nilai (0-100): ");
            int nilai = scan.nextInt();
            scan.nextLine();
            dinilai.tugasDinilai(nilai);
            System.out.printf("nilai tugas %s adalah %d\n", dinilai.nama, nilai);
        }
        break;
    case 3:
        Mahasiswa04 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 tudak valid");
    }
} while (pick >= 1 && pick <=4 );
scan.close();
}
}

```

Menu:

1. Megumpulkan tugas
 2. Menilai tugas
 3. Melihat tugas teratas
 4. Melihat daftar tugas
 5. Melihat tugas pertama
 6. cek berapa banyak tugas yang dikumpulkan
- pilih:

1. Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan sama dengan verifikasi hasil percobaan! Bagian mana yang perlu diperbaiki?
 - Dengan menghapus syntax top— pada method pop
2. Berapa banyak data tugas mahasiswa yang dapat ditampung di dalam stack ? tunjukan potongan kode programnya !
 - `StackTugasMahasiswa stack = new StackTugasMahasiswa(5);`
3. Mengapa perlu pengecekan kondisi `!isFull()` pada method push? Kalau kondisi if-else tersebut dihapus, apa dampaknya?
 - diperlukan untuk memastikan bahwa elemen baru hanya dapat ditambahkan ke stack jika stack belum penuh
4. 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 Mahasiswa04 peekBottom() {
    if (!isEmpty()) {
        return stack[0]; // Elemen pertama dalam stack
    } else {
        System.out.println("Stack kosong! Tidak ada tugas yang dikumpulkan.");
        return null;
    }
}
```
5. Tambahkan method untuk dapat menghitung berapa banyak tugas yang sudah dikumpulkan saat ini, serta tambahkan operasi menunya!

```
public int jumlahTugas() {
    return top + 1; // Karena indeks top dimulai dari -1
}
```
6. Commit dan push kode program ke github

Percobaan 2

```
package Pertemuan_10;

import java.util.Stack;

public class StackKonversi {
    int [] tumpukanBiner;

    int size;

    int top;

    public StackKonversi(int size) {
        this.size = size;
        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! Tidak bisa menambahkan data.");
        } else {
            top++;
            tumpukanBiner[top] = data;
        }
    }

    public int pop() {
        if (isEmpty()) {
            System.out.println("Stack kosong! Tidak ada data untuk diambil.");
            return -1;
        } else {
            int data = tumpukanBiner[top];
            top--;
            return data;
        }
    }
}

```

1. Jelaskan alur kerja dari method BkonversiDecimalKebiner

Inisialisasi Stack:

Membuat objek StackKonversi dengan kapasitas tertentu (dalam contoh ini, kapasitasnya adalah 5).

Stack digunakan untuk menyimpan sisa hasil pembagian bilangan desimal dengan 2.

Menghitung Sisa Pembagian:

Selama nilai desimal (nilai) lebih besar dari 0:

Hitung sisa pembagian nilai % 2 (sisa ini adalah digit biner).

Simpan sisa tersebut ke dalam stack menggunakan push.

Perbarui nilai desimal dengan nilai / 2 untuk iterasi berikutnya.

TUGAS

```
package Pertemuan_9;

import java.util.Scanner;
import java.util.Stack;

class Surat {

    String idSurat;

    String namaMahasiswa;

    String kelas;

    char jenisIzin; // S = Sakit, I = Izin

    int durasi;

    public Surat() {

    }

    public Surat(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 tampilkan() {

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

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

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

        System.out.println("Jenis Izin   : " + jenisIzin);

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

    }

}

class StackSurat {

    Stack<Surat> daftarSurat = new Stack<>();

    public void terimaSurat(Surat surat) {

        daftarSurat.push(surat);

    }

}
```



```

public void prosesSurat() {
    if (!daftarSurat.isEmpty()) {
        Surat diproses = daftarSurat.pop();
        System.out.println("Surat berikut telah diproses:");
        diproses.tampilkan();
    } else {
        System.out.println("Tidak ada surat yang bisa diproses.");
    }
}

public void lihatSuratTerakhir() {
    if (!daftarSurat.isEmpty()) {
        System.out.println("Surat Terakhir:");
        daftarSurat.peek().tampilkan();
    } else {
        System.out.println("Tidak ada surat yang tersedia.");
    }
}

public void cariSurat(String namaMahasiswa) {
    boolean ditemukan = false;
    for (Surat s : daftarSurat) {
        if (s.namaMahasiswa.equalsIgnoreCase(namaMahasiswa)) {
            System.out.println("Surat ditemukan:");
            s.tampilkan();
            ditemukan = true;
            break;
        }
    }
    if (!ditemukan) {
        System.out.println("Surat dari mahasiswa dengan nama " + namaMahasiswa + " tidak ditemukan.");
    }
}

public class surat {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        StackSurat stackSurat = new StackSurat();
    }
}

```

```

int pilihan;

do {
    System.out.println("\n=== MENU SURAT IZIN MAHASISWA ===");
    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("5. Keluar");
    System.out.print("Pilih menu: ");
    pilihan = input.nextInt();
    input.nextLine(); // konsumsi newline

    switch (pilihan) {
        case 1:
            System.out.print("ID Surat      : ");
            String id = input.nextLine();
            System.out.print("Nama Mahasiswa : ");
            String nama = input.nextLine();
            System.out.print("Kelas        : ");
            String kelas = input.nextLine();
            System.out.print("Jenis Izin (S/I): ");
            char jenis = input.next().charAt(0);
            System.out.print("Durasi (hari)  : ");
            int durasi = input.nextInt();
            input.nextLine(); // konsumsi newline
            stackSurat.terimaSurat(new Surat(id, nama, kelas, jenis, durasi));
            break;

        case 2:
            stackSurat.prosesSurat();
            break;

        case 3:
            stackSurat.lihatSuratTerakhir();
            break;

        case 4:
            System.out.print("Masukkan nama mahasiswa: ");
            String cariNama = input.nextLine();

```

```
        stackSurat.cariSurat(cariNama);

        break;

    case 5:

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

        break;

    default:

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

    }

    } while (pilihan != 5);

    input.close();

}

}
```

https://github.com/AlfredaDhaifullah04/Semester-2/tree/master/Pertemuan_10