

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

JOBSHEET 9



SHAFIQA NABILA MAHARANI KHOIRUNNISA

244107020221

TI – 1B

2.1.1 Langkah-langkah Percobaan

A. Class Mahasiswa

```
public class Mahasiswa20 {  
    String nama;  
    String nim;  
    String kelas;  
    int nilai;  
  
    public Mahasiswa20(String nama, String nim, String kelas) {  
        this.nama = nama;  
        this.nim = nim;  
        this.kelas = kelas;  
        this.nilai = -1;  
    }  
  
    void tugasDinilai(int nilai) {  
        this.nilai = nilai;  
    }  
}
```

Move this file to a named package.

B. Class StackTugasMahasiswa

```

public class StackTugasMahasiswa20 {
    Mahasiswa20[] stack;
    int top;
    int size;

    public StackTugasMahasiswa20(int size) {
        this.size = size;
        this.stack = new Mahasiswa20[size];
        this.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(Mahasiswa20 mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println(x:"Stack penuh! Tidak bisa menambahkan Tugas Lagi.");
        }
    }
}

```

C. Class Utama

```

import java.util.Scanner; // Move this file to a named package.
public class MahasiswaDemo {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in); // Resource leak: 'sc' is never closed
        StackTugasMahasiswa20 stack = new StackTugasMahasiswa20(size:5);
        int pilih;
        do {
            System.out.println(x:"\n Menu"); // Replace this use of System.out by a logger.
            System.out.println(x:"1. Mengumpulkan Tugas"); // Replace this use of System.out by a logger.
            System.out.println(x:"2. Menilai Tugas"); // Replace this use of System.out by a logger.
            System.out.println(x:"3. Melihat Tugas Teratas"); // Replace this use of System.out by a logger.
            System.out.println(x:"4. Melihat Daftar Tugas"); // Replace this use of System.out by a logger.
            System.out.print(s:"Pilih Menu : "); // Replace this use of System.out by a logger.
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print(s:"Nama: "); // Replace this use of System.out by a logger.
                    String nama = sc.nextLine();
                    System.out.print(s:"NIM: "); // Replace this use of System.out by a logger.
                    String nim = sc.nextLine();
                    System.out.print(s:"Kelas: "); // Replace this use of System.out by a logger.
                    String kelas = sc.nextLine();
                    Mahasiswa20 mhs = new Mahasiswa20(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf((format:"Tugas %s berhasil dikumpulkan\n", mhs.nama); // %n should be used in place of \n to produce a new line.
                    break;

                case 2:
                    Mahasiswa20 dinilai = stack.pop();
            }
        } while (pilih != 0);
    }
}

```

Hasil run :

Menu

1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Terastas
4. Melihat Daftar Tugas

Pilih Menu : 1

Nama: Naufal

NIM: 1001

Kelas: TI 1H

Tugas Naufal berhasil dikumpulkan

Menu

1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Terastas
4. Melihat Daftar Tugas

Pilih Menu : 1

Nama: Fika

NIM: 1002

Kelas: TI 1B

Tugas Fika berhasil dikumpulkan

Menu

1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Terastas
4. Melihat Daftar Tugas

Pilih Menu : 1

Nama: Vanesa

NIM: 1003

Kelas: TI 1B

Tugas Vanesa berhasil dikumpulkan

Menu

1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Terastas
4. Melihat Daftar Tugas

Pilih Menu : 3

Tugas terakhir dikumpulkan oleh Vanesa

Menu

1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Terastas
4. Melihat Daftar Tugas

Pilih Menu : 4

Daftar semua tugas

Nama	NIM	Kelas
Naufal	1001	TI 1H
Fika	1002	TI 1B
Vanesa	1003	TI 1B

Menu

1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Terastas
4. Melihat Daftar Tugas

Pilih Menu : 2

Menilai tugas dari Vanesa

Masukkan Nilai (0-100): 87

Nilai Tugas Vanesa adalah 87

2.1.3 Pertanyaan

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

Mahasiswa20.java

```
public class Mahasiswa20 {    Move this file to a named package.
    String nim, nama, kelas;    Declare "nama" and all following declarations on a separate line. [+1 location]
    int nilai;

    public Mahasiswa20(String nim, String nama, String kelas) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.nilai = -1;
    }

    public void tugasDinilai(int nilai) {
        this.nilai = nilai;
    }

    @Override
    public String toString() {
        return "NIM: " + nim + ", Nama: " + nama + ", Kelas: " + kelas + ", Nilai: " +
            (nilai == -1 ? "Belum Dinilai" : nilai);
    }
}
```

StackTugasMahasiswa20.java

```
public void push(Mahasiswa20 mhs) {
    if (!isFull()) {
        top++;
        stack[top] = mhs;
        System.out.println("Tugas " + mhs.nama + " berhasil ditambahkan.");
    } else {
        System.out.println(x:"Stack penuh! Tidak dapat menambahkan tugas.");
    }
}
```

```

public Mahasiswa20 pop() {
    if (!isEmpty()) {
        Mahasiswa20 mhs = stack[top];
        top--;
        System.out.println("Tugas " + mhs.nama + " dikeluarkan untuk dinilai.");
        return mhs;
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas untuk dinilai.");
        return null;
    }
}

```

```

public void print() {
    if (isEmpty()) {
        System.out.println(x:"Stack kosong.");
    } else {
        System.out.println(x:"Daftar tugas mahasiswa:");
        for (int i = top; i >= 0; i--) {
            System.out.println(stack[i]);
        }
    }
}

```

MahasiswaDemo20.java

```

System.out.println("Nilai tugas " + nilai + " selesai.");
dinilai.tugasDinilai(nilai);
System.out.println("Tugas " + dinilai.nama + " telah dinilai.");

```

```

case 3:
    Mahasiswa20 atas = stack.peek();
    if (atas != null)
        System.out.println("Tugas Teratas: " + atas);
    else
        System.out.println(x:"Stack kosong.");
    break;

case 4:
    stack.print();
    break;

```

2. Berapa banyak data tugas mahasiswa yang dapat ditampung di dalam Stack? Tunjukkan 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?

Pengecekan !isFull() mencegah penambahan elemen ke dalam stack jika stack sudah penuh. Jika if-else dihapus Akan terjadi ArrayIndexOutOfBoundsException saat mencoba menambahkan elemen ke array yang sudah penuh dan Program bisa crash/error saat runtime.

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!

Tambahkan method baru pada StackTugasMahasiswa :

```
public Mahasiswa20 bottom() {  
    if (!isEmpty()) {  
        return stack[0];  
    } else {  
        return null;  
    }  
}
```

Tambahkan menu di MahasiswaDemo :

```
case 6:  
    Mahasiswa20 bawah = stack.bottom();  
    if (bawah != null) {  
        System.out.println(x:"Tugas terbawah:");    Replace this use of System.out by a logger.  
        System.out.println(bawah.nama + " - " + bawah.nim);    Replace this use of System.out by a logger.  
    } else {  
        System.out.println(x:"Stack masih kosong!");    Replace this use of System.out by a logger.  
    }  
}
```

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

Tambahkan Method baru di StackTugasMahasiswa :


```
public int jumlahTugas() {  
    return top + 1;  
}
```

Tambahkan menu di MahasiswaDemo :

```
case 7:  
    System.out.println("Jumlah tugas yang sudah dikumpulkan: " + stack.jumlahTugas());  
    break;  
}
```

2.2 Percobaan 2: Konversi Nilai Tugas ke Biner

```
public class StackKonversi20 {    Move this file to a named package.  
    int[] tumpukanBiner;  
    int size;  
    int top;  
  
    public StackKonversi20() {  
        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(x:"Stack penuh! Tidak bisa menambahkan data.");    Replace this use of System.out by a log  
        } else {  
            top++;  
            tumpukanBiner[top] = data;  
        }  
    }  
  
    public int pop() {  
        if (isEmpty()) {  
            System.out.println(x:"Stack kosong");    Replace this use of System.out by a logger.  
        }  
    }  
}
```

```

case 2:
    Mahasiswa20 dinilai = stack.pop();
    if (dinilai != null) {
        System.out.println("Menilai tugas dari " + dinilai.nama);
        System.out.print(s:"Masukkan Nilai (0-100): ");
        int nilai = sc.nextInt();
        dinilai.tugasDinilai(nilai);
        System.out.printf(format:"Nilai Tugas %s adalah %d\n", dinilai.nama, nilai);
        String biner = stack.konversiDesimalkeBiner(nilai);
        System.out.println("Nilai Biner Tugas: " + biner);
    }
    break;

```

```

public String konversiDesimalkeBiner(int nilai) {
    StackKonversi20 stack = new StackKonversi20();
    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;
}

```

Hasil run :

```

Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Terastast
4. Melihat Daftar Tugas
Pilih Menu : 2
Menilai tugas dari Vanesa
Masukkan Nilai (0-100): 87
Nilai Tugas Vanesa adalah 87
Nilai Biner Tugas: 1010111

```

2.2.3 Pertanyaan

1. Jelaskan alur kerja dari method konversiDesimalKeBiner!

Method konversiDesimalKeBiner bekerja dengan cara mengubah bilangan desimal menjadi bentuk biner menggunakan bantuan struktur data stack. Pertama, method ini membuat objek stack untuk menyimpan sisa pembagian dari bilangan desimal dengan angka 2. Selanjutnya, selama nilai desimal masih lebih besar dari 0, bilangan tersebut dibagi 2 secara berulang dan setiap sisa pembagian (hasil modulus) dimasukkan ke dalam stack. Proses ini akan menyimpan digit-digit biner dalam urutan terbalik karena sifat stack adalah Last In First Out (LIFO). Setelah proses pembagian selesai, isi stack dikeluarkan satu per satu untuk dicetak sehingga menghasilkan representasi bilangan dalam bentuk biner dengan urutan yang benar. Misalnya, jika input desimal adalah 10, maka hasil akhirnya adalah 1010.

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

Jika kondisi perulangan dalam method konversiDesimalKeBiner diubah menjadi while(kode != 0), secara umum hasil yang diperoleh tetap sama untuk semua bilangan desimal positif yang lebih besar dari nol. Namun, ketika input yang diberikan adalah nol (0), maka perulangan tidak akan dijalankan karena kondisi kode != 0 akan bernilai salah. Akibatnya, tidak ada angka yang dimasukkan ke dalam stack dan tidak ada output yang ditampilkan. Hal ini serupa dengan kondisi awal (kode > 0) karena kedua kondisi tersebut sama-sama tidak mengeksekusi perulangan jika input adalah 0. Oleh karena itu, meskipun kondisi diubah, tidak ada perbedaan hasil yang berarti, tetapi tetap disarankan untuk menambahkan pengecekan khusus agar jika input adalah nol, program tetap dapat menampilkan output "0" sebagai hasil konversi yang benar.

2.4 Latihan Praktikum

```
public class StackSurat20 {    Move this file to a named package.
    Surat20[] data;
    int top;

    public StackSurat20(int kapasitas) {
        data = new Surat20[kapasitas];
        top = -1;
    }

    public boolean isEmpty() {
        return top == -1;
    }

    public boolean isFull() {
        return top == data.length - 1;
    }

    public void push(Surat20 s) {
        if (!isFull()) {
            data[++top] = s;
        } else {
            System.out.println(x:"Stack penuh, tidak dapat menambahkan surat!");    Replac
        }
    }

    public Surat20 pop() {
        if (!isEmpty()) {
            return data[top--];
        } else {
            System.out.println(x:"Stack kosong, tidak ada surat yang dapat diproses.");
            return null;
        }
    }
}
```

```

public class Surat20 {
    String idSurat, namaMahasiswa, kelas;
    char jenisIzin;
    int durasi;

    public Surat20(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");
        System.out.println(x:"-----");
    }
}

```

```

import java.util.Scanner;    Move this file to a named package.
public class SuratDemo20 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StackSurat20 stack = new StackSurat20(kapasitas:10);
        int pilihan;

        do {
            System.out.println(x:"\n== Menu Layanan Surat Izin Mahasiswa ==");    Replace this use of System.out by a logger.
            System.out.println(x:"1. Terima Surat Izin");    Replace this use of System.out by a logger.
            System.out.println(x:"2. Proses Surat Izin");    Replace this use of System.out by a logger.
            System.out.println(x:"3. Lihat Surat Izin Terakhir");    Replace this use of System.out by a logger.
            System.out.println(x:"4. Cari Surat Izin (berdasarkan nama mahasiswa)");    Replace this use of System.out by a logger.
            System.out.println(x:"5. Tampilkan Semua Surat");    Replace this use of System.out by a logger.
            System.out.println(x:"0. Keluar");    Replace this use of System.out by a logger.
            System.out.print(s:"Pilih menu: ");    Replace this use of System.out by a logger.
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print(s:"ID Surat      : ");    Replace this use of System.out by a logger.
                    String id = sc.nextLine();
                    System.out.print(s:"Nama Mahasiswa : ");    Replace this use of System.out by a logger.
                    String nama = sc.nextLine();
                    System.out.print(s:"Kelas        : ");    Replace this use of System.out by a logger.
                    String kelas = sc.nextLine();
                    System.out.print(s:"Jenis Izin (S/I) : ");    Replace this use of System.out by a logger.
                    char jenis = sc.nextLine().charAt(index:0);
                    System.out.print(s:"Durasi (hari) : ");    Replace this use of System.out by a logger.
                    int durasi = sc.nextInt();

```

Hasil run :

```
=== Menu Layanan Surat Izin Mahasiswa ===
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Tampilkan Semua Surat
0. Keluar
Pilih menu: 1
ID Surat      : 1001
Nama Mahasiswa : Fika
Kelas        : TI 1B
Jenis Izin (S/I): S
Durasi (hari) : 2

=== Menu Layanan Surat Izin Mahasiswa ===
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Tampilkan Semua Surat
0. Keluar
Pilih menu: 1
ID Surat      : 1002
Nama Mahasiswa : Naufal
Kelas        : TI 1H
Jenis Izin (S/I): I
Durasi (hari) : 2
```

=== Menu Layanan Surat Izin Mahasiswa ===

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Tampilkan Semua Surat
0. Keluar

Pilih menu: 1

ID Surat : 1003
Nama Mahasiswa : Vanesa
Kelas : TI 1B
Jenis Izin (S/I): S
Durasi (hari) : 3

=== Menu Layanan Surat Izin Mahasiswa ===

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Tampilkan Semua Surat
0. Keluar

Pilih menu: 2

Surat yang diproses:

ID Surat : 1003
Nama Mahasiswa : Vanesa
Kelas : TI 1B
Jenis Izin : S
Durasi : 3 hari

=== Menu Layanan Surat Izin Mahasiswa ===

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Tampilkan Semua Surat
0. Keluar

Pilih menu: 3

Surat izin terakhir:

ID Surat : 1002
Nama Mahasiswa : Naufal
Kelas : TI 1H
Jenis Izin : I
Durasi : 2 hari

=== Menu Layanan Surat Izin Mahasiswa ===

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Tampilkan Semua Surat
0. Keluar

Pilih menu: 4

Masukkan nama mahasiswa: Naufal

Surat ditemukan:

ID Surat	: 1002
Nama Mahasiswa	: Naufal
Kelas	: TI 1H
Jenis Izin	: I
Durasi	: 2 hari

=== Menu Layanan Surat Izin Mahasiswa ===

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Tampilkan Semua Surat
0. Keluar

Pilih menu: 5

Semua surat yang tersimpan:

ID Surat : 1002
Nama Mahasiswa : Naufal
Kelas : TI 1H
Jenis Izin : I
Durasi : 2 hari

ID Surat : 1001
Nama Mahasiswa : Fika
Kelas : TI 1B
Jenis Izin : S
Durasi : 2 hari
