

Algoritma dan Struktur Data

Jobsheet 09 : Stack



Oleh:

Gagah Irlangsa Eka Yuanata (244107060076)

Kelas SIB-1B / 10

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65141

Link Github : <https://github.com/GagahIr/Praktikum-ASD>

2.1.1 Langkah-langkah Percobaan

```
StackTugasMahasiswa10.jav

public class StackTugasMahasiswa10 {
    Mahasiswa10[] stack;
    int top, size;

    public StackTugasMahasiswa10(int size) {
        this.size = size;
        stack = new Mahasiswa10[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(Mahasiswa10 mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan tugas lagi");
        }
    }

    public Mahasiswa10 pop() {
        if (!isEmpty()) {
            Mahasiswa10 m = stack[top];
            top--;
            return m;
        } else {
            System.out.println("Stack kosong! Tidak ada tugas untuk dinilai!");
            return null;
        }
    }

    public Mahasiswa10 peek() {
        if (isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong! Tidak ada tugas yang dikumpulkan");
            return null;
        }
    }

    public void print() {
        for (int i = top; i >= 0; i--) {
            System.out.println(stack[i].nama + "\t" + stack[i].nim + "\t" + stack[i].kelas);
        }
        System.out.println("");
    }
}
```

```

MahasiswaDemo10.java

import java.util.Scanner;

public class MahasiswaDemo10 {
    public static void main(String[] args) {
        StackTugasMahasiswa10 stack = new StackTugasMahasiswa10(5);
        int pilih;
        Scanner scan = new Scanner(System.in);
        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();
                    Mahasiswa10 mhs = new Mahasiswa10(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n", mhs.nama);
                    break;
                case 2:
                    Mahasiswa10 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 " + dinilai.nama, nilai );
                    }
                    break;
                case 3:
                    Mahasiswa10 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);
    }
}

```

```

public class Mahasiswa10 {
    String nama, nim, kelas;
    int nilai;

    Mahasiswa10(String nama, String nim, String kelas){
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
    }

    Mahasiswa10(){

    }

    void TugasDiNilai(int nilai){
        this.nilai = nilai;
    }
}

```

2.1.2 Verifikasi Hasil Percobaan

```

PS C:\Users\ACER\Documents\#Jobsheet#S2\Praktikum Code\Praktikum-ASD>
Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat daftar Tugas
Pilih: 1
Nama: Dila
NIM: 1001
Kelas: 1A
Tugas Dila berhasil dikumpulkan

Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat daftar Tugas
Pilih: 1
Nama: Erik
NIM: 1002
Kelas: 1B
Tugas Erik berhasil dikumpulkan

Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat daftar Tugas
Pilih: 1
Nama: Tika
NIM: 1003
Kelas: 1C
Tugas Tika berhasil dikumpulkan

```

```

Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat daftar Tugas
Pilih: 4
Daftar semua tugas
Nama    NIM    Kelas
Tika    1003   1C
Erik    1002   1B
Dila    1001   1A

```

```

Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat daftar Tugas
Pilih: 2
Menilai tugas dari Tika
Masukkan nilai (0-100): 87
Nilai Tugas 87 adalah Tika

```

2.1.3 Pertanyaan

1. Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan **sama** dengan verifikasi hasil percobaan! Bagian mana yang perlu diperbaiki?
2. Berapa banyak data tugas mahasiswa yang dapat ditampung di dalam Stack? Tunjukkan potongan kode programnya!
3. Mengapa perlu pengecekan kondisi **!isFull()** pada method **push**? Kalau kondisi if-else tersebut dihapus, apa dampaknya?

- 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 untuk dapat menghitung berapa banyak tugas yang sudah dikumpulkan saat ini, serta tambahkan operasi menunya!
- Commit dan push kode program ke Github**

Jawaban

- Sebelum

```
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(x:"");
}
```

Sesudah

```
public void print() {
    for (int i = top; i >= 0; i--) {
        System.out.println(stack[i].nama + "\t" + stack[i].nim + "\t" + stack[i].kelas);
    }
    System.out.println(x:"");
}
```

Karena stack maka elemen terakhir yang masuk berada di posisi paling atas. Jadi menampilkan dari yang input terbaru ke yang terlama, harus mulai mencetak dari indeks top ke bawah.

- StackTugasMahasiswa10 stack = new StackTugasMahasiswa10(size:5);**
- Jumlah data tugas mahasiswa yang dapat ditampung di dalam stack adalah sebanyak 5
- Agar tidak stack overflow, yaitu saat kita mencoba menambahkan data ke dalam stack yang sudah penuh. Ketika if else dihapus maka ketika top sudah mencapai batas maksimal ($top == size - 1$) dan tetap menambahkan elemen maka yang terjadi **ArrayIndexOutOfBoundsException** karena kita mencoba mengakses indeks yang tidak tersedia pada stack.

```
Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat daftar Tugas
5. Melihat Tugas Terbawah
Pilih: 5
Tugas pertama dikumpulkan oleh Dila
```

-

```
public Mahasiswa10 TugasTerbawah() {
    if (!isEmpty()) {
        return stack[0];
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan");
        return null;
    }
}
```

```

case 5:
    Mahasiswa10 terbawah = stack.TugasTerbawah();
    if (terbawah != null) {
        System.out.println("Tugas pertama dikumpulkan oleh " + terbawah.nama);
    }
    break;

```

```

Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat daftar Tugas
5. Melihat Tugas Terbawah
6. Melihat jumlah tugas yang dikumpulkan
Pilih: 6
Jumlah tugas yang dikumpulkan adalah: 3

```

5.

```

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

```

```

case 6:
    System.out.println("Jumlah tugas yang dikumpulkan adalah: " + stack.jumlahTugas());
    break;

```

2.2.1 Langkah-langkah Percobaan

```

case 2:
    Mahasiswa10 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);
    }
    break;

```

```

public String konversiDesimalKeBiner(int nilai) {
    StackKonversi stack = new StackKonversi();
    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;
}

```

```

public class StackKonversi {
    int[] tumpukanBiner;
    int size, top;

    public StackKonversi() {
        this.size = 32; // asumsi 32 bit
        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;
        }
    }
}

```

2.2.2 Verifikasi Hasil Percobaan

```
Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat daftar Tugas
5. Melihat Tugas Terbawah
6. Melihat jumlah tugas yang dikumpulkan
Pilih: 2
Menilai tugas dari Dila
Masukkan nilai (0-100): 89
Nilai Tugas Dila adalah 89
Nilai biner tugas: 1011001
```

2.2.3 Pertanyaan

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

JAWABAN

1. Langkah 1: Konversi desimal ke biner menggunakan sisa bagi 2
Lalu ambil sisa (0 atau 1), masukkan ke stack, bagi nilai dengan 2
Langkah 2: Ambil dari stack untuk membentuk string biner
ambil satu per satu dari stack
return biner; hasil akhir string biner

```
public String konversiDesimalKeBiner(int nilai) {
    StackKonversi stack = new StackKonversi();
    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;
}
```

2.

Error karena variable kode belum dideklarasikan. Tetapi jika **nilai != 0** Secara logika hasilnya tetap sama karena bilangan positif nilai akan bernilai 0. Tetapi ketika nilai bernilai negatif, maka perulangan tidak akan berhenti sedangkan nilai > 0 lebih aman karena tidak mungkin negatif