

# **Laporan Jobsheet 9 Algoritma dan Struktur Data**

## **Stack**



244107020151

Ghazwan Ababil

Teknik Informatika / TI – E (11)

Politeknik Negeri Malang

Jurusan Teknologi Informasi

2025

## 1. Praktikum

### 1.1. Percobaan 1: Mahasiswa Mengumpulkan Tugas

#### a. Class Mahasiswa

1. Membuat folder baru Jobsheet9 di dalam repository Praktikum-ASD. Kemudian membuat class baru dengan nama Mahasiswa11.java
2. Menambahkan atribut, method, dan konstruktor pada class Mahasiswa11 sesuai dengan class diagram

```
package Jobsheet9;

public class Mahasiswa11 {
    String nim,nama,kelas;
    int nilai;

    Mahasiswa11(){};
    Mahasiswa11(String nama, String nim, String kelas){
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        nilai = -1;
    }

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

#### b. Class StackTugasMahasiswa

3. Membuat class baru dengan nama StackTugasMahasiswa11.java
4. Menambahkan atribut, method, dan konstruktor sesuai dengan class diagram.  
method **isFull** untuk mengecek apakah tumpukan tugas mahasiswa sudah terisi penuh sesuai kapasitas, method **isEmpty** untuk mengecek apakah tumpukan tugas masih kosong, method **push** untuk menambahkan tugas ke dalam tumpukan stack, method **pop** untuk melakukan penilaian dan mengeluarkan tugas yang akan dinilai, method **peek** untuk mengecek tugas yang berada pada tumpukan paling atas, dan method **print** untuk menampilkan semua daftar tugas mahasiswa pada stack

```

package Jobsheet9;

public class StackTugasMahasiswa11 {
    Mahasiswa11[] stack;
    int size,top;

    StackTugasMahasiswa11(int size){
        this.size = size;
        stack = new Mahasiswa11[size];
        top = -1;
    }

    boolean isFull(){
        return top == size -1;
    }

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

    void push (Mahasiswa11 mhs){
        if (!isFull()) stack[++top] = mhs;
        else System.out.println("Stack Penuh! Tidak bisa
menambahkan tugas lagi.");
    }

    Mahasiswa11 pop(){
        if(top != -1){
            Mahasiswa11 m = stack[top];
            top--;
            return m;
        }else {
            System.out.println("Stack kosong! Tidak ada tugas
untuk dinilai.");
            return null;
        }
    }

    Mahasiswa11 peek(){
        if (!isEmpty()) return stack[top];
        else{
            System.out.println("Stack kosong! Tidak ada tugas
yang dikumpulkan");
            return null;
        }
    }

    void print(){
        for (int i = 0; i <= top; i++) {
            System.out.printf("%s \t %s \t
%s",stack[i].nama,stack[i].nim,stack[i].kelas);
            System.out.println();
        }
        System.out.println();
    }
}

```

c. Class Utama (MahasiswaDemo)

5. Membuat class baru dengan nama MahasiswaDemo11.java
6. Menambahkan fungsi main, kemudian melakukan instansiasi object stack berserta scanner didalam main dan menambahkan menu untuk pemilihan dalam mengoperasikan stack dalam mengelola data mahasiswa dengan struktur do-while

```

package Jobsheet9;

import java.util.Scanner;

public class MahasiswaDemol1 {
    public static void main(String[] args) {
        StackTugasMahasiswa11 stack = new StackTugasMahasiswa11(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();
                    Mahasiswa11 mhs = new Mahasiswa11(nama,nim,kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil
dikumpulkan\n",mhs.nama);
                    break;
                case 2:
                    Mahasiswa11 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:
                    Mahasiswa11 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 \t NIM \t Kelas");
                    stack.print();
                    break;
                default:
                    System.out.println("Pilihan tidak valid");
                    break;
            }
        } while (pilih > 0 && pilih < 5);
    }
}

```

d. Output Kode program

```
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: 3
Tugas terakhir dikumpulkan oleh Erik

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
Dila      1001     1A
Erik      1002     1B
Tika      1003     1C

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 Tika adalah 87

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

e. Pertanyaan

1. Memodifikasi kode program agar output yang dihasilkan sama dengan verifikasi hasil percobaan. Agar daftar semua tugas yang dikumpulkan pada stack ditampilkan secara terurut dari yang terakhir kali mengumpulkan, maka diperlukan memodifikasi method print untuk mengubah kondisi dan inisialisasi perulangan untuk print pada stack menjadi

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

## Output kode program

```
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: 3
Tugas terakhir dikumpulkan oleh Erik

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 Tika adalah 87

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

2. Pada stack tersebut, dapat menampung maksimal 5 data tugas mahasiswa karena pada demo class, stack diinstansiasikan dengan size 5 menggunakan konstruktor berparameter, yang kemudian memproses parameter tersebut sebagai size dari stack. Potongan kode program MahasiswaDemo11.java

```
// MahasiswaDemo11.java
StackTugasMahasiswa11 stack = new StackTugasMahasiswa11(5);
```

### Potongan kode program StackTugasMahasiswa11.java

```
// StackTugasMahasiswa11.java
StackTugasMahasiswa11(int size){
    this.size = size;
    stack = new Mahasiswa11[size];
    top = -1;
}
```

3. Pengecekan kondisi **!isFull()** pada method **push** dilakukan untuk mengecek terlebih dahulu apakah stack dalam kondisi penuh atau tidak, jika kondisi stack sudah dalam keadaan penuh, maka penambahan data dengan method push tidak dijalankan, begitu juga sebaliknya. Jika kondisi if-else tersebut dihapus, maka saat kondisi stack dalam

keadaan penuh dan melakukan penambahan data dengan method push, akan menyebabkan error array index out of bounds yaitu indeks stack ingin diakses melebihi batas size yang ditentukan yaitu size - 1.

4. Memodifikasi class MahasiswaDemo11 dan StackTugasMahasiswa11 agar pengguna dapat melihat mahasiswa yang pertama kali mengumpulkan tugas melalui menu lihat tugas terbawah

Modifikasi class StackTugasMahasiswa11

```
Mahasiswa11 peekBottom(){
    if (!isEmpty()) return stack[0];
    else{
        System.out.println("Stack kosong! Tidak ada tugas yang
dikumpulkan");
        return null;
    }
}
```

Modifikasi class MahasiswaDemo11

```
do{
//. . .
System.out.println("5. Melihat Tugas Terbawah");

//. . .
    case 5:
        Mahasiswa11 lihatTerbawah = stack.peekBottom();
        if (lihatTerbawah != null)
            System.out.println("Tugas pertama dikumpulkan oleh "
+ lihatTerbawah.nama);
        break;
//. . .
} while (pilih > 0 && pilih < 6);
```

Output kode program

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

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
Pilih: 11
Pilihan tidak valid
```



5. Menambahkan method untuk dapat menghitung berapa banyak tugas yang sudah dikumpulkan saat ini, dan menambahkan menunya

Menambahkan method pada class StackTugasMahasiswa11

```
int totalTugas() {  
    return top + 1;  
}
```

Menambahkan opsi pada class MahasiswaDemo11

```
do{  
    //. . .  
  
    System.out.println("6. Melihat Tugas yang sudah dikumpulkan saat ini");  
  
    //. . .  
    case 6:  
        int totalTugas = stack.totalTugas();  
        if (totalTugas == 0)  
            System.out.println("Tidak ada tugas yang dikumpulkan saat ini.");  
        else  
            System.out.println("Total tugas yang sudah dikumpulkan saat ini: " + totalTugas);  
        break;  
    //. . .  
} while (pilih > 0 && pilih < 7);
```

Output kode program

```
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Melihat Tugas yang sudah dikumpulkan saat ini  
Pilih: 6  
Tidak ada tugas yang dikumpulkan saat ini.  
  
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Melihat Tugas yang sudah dikumpulkan saat ini  
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  
5. Melihat Tugas Terbawah  
6. Melihat Tugas yang sudah dikumpulkan saat ini  
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  
5. Melihat Tugas Terbawah  
6. Melihat Tugas yang sudah dikumpulkan saat ini  
Pilih: 6  
Total tugas yang sudah dikumpulkan saat ini: 2  
  
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Melihat Tugas yang sudah dikumpulkan saat ini  
Pilih: 2  
Menilai tugas dari Erik  
Masukkan nilai (0-100): 87  
Nilai tugas Erik adalah 87  
  
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Melihat Tugas yang sudah dikumpulkan saat ini  
Pilih: 6  
Total tugas yang sudah dikumpulkan saat ini: 1
```

```

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
Pilih: 2
Menilai tugas dari Dila
Masukkan nilai (0-100): 87
Nilai tugas Dila adalah 87

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
Pilih: 6
Tidak ada tugas yang dikumpulkan saat ini.

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
Pilih: 11
Pilihan tidak valid

```

## 6. Push dan commit kode program ke github

```

PS D:\Java\Praktikum-ASD> git add .
PS D:\Java\Praktikum-ASD> git commit -m "Jobsheet 9 Percobaan 1 dan pertanyaan Stack Mahasiswa"
[main 8fc3aa9] Jobsheet 9 Percobaan 1 dan pertanyaan Stack Mahasiswa
3 files changed, 155 insertions(+)
create mode 100644 Jobsheet9/Mahasiswa11.java
create mode 100644 Jobsheet9/MahasiswaDemo11.java
create mode 100644 Jobsheet9/StackTugasMahasiswa11.java
PS D:\Java\Praktikum-ASD> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.80 KiB | 306.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ghazwanz/Praktikum-ASD.git
290d2b4..8fc3aa9 main -> main
branch 'main' set up to track 'origin/main'.

```

## 1.2. Percobaan 2: Konversi Nilai Tugas ke Biner

- Menambahkan method konversiDesimalKeBiner dengan menerima parameter dengan tipe int

```

String konversiDesimalKeBiner(int nilai){
    StackKonversi11 stack = new StackKonversi11();
    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;
}

```

- Membuat file baru dengan nama StackKonversi11.java untuk memisahkan stack mahasiswa dengan stack yang akan digunakan untuk biner. Atribut dan method pada

class StackKonversi11 kurang lebih hampir sama seperti stack pada class StackTugasMahasiswa11 sebelumnya

```
package Jobsheet9;

public class StackKonversi11 {
    int[] tumpukanBiner;
    int size,top;

    StackKonversi11(){
        this.size = 32; // asumsi 32 bit
        tumpukanBiner = new int[size];
        top = -1;
    }

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

    boolean isFull(){
        return top == size-1;
    }

    void push(int data){
        if (isFull()) System.out.println("Stack penuh");
        else tumpukanBiner[++top] = data;
    }

    int pop(){
        if (isEmpty()) {
            System.out.println("Stack kosong");
            return -1;
        } else{
            int data = tumpukanBiner[top];
            top--;
            return data;
        }
    }
}
```

- c. Menambahkan method konversiDesimalKeBiner untuk mengonversi nilai setelah melakukan penilaian tugas pada method pop di class MahasiswaDemo11

```
case 2:
    Mahasiswa11 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\n",dinilai.nama,nilai);
        String biner = stack.konversiDesimalKeBiner(nilai);
        System.out.println("Nilai Biner Tugas: " + biner);
    }
    break;
```

d. Output kode program

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
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
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
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
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
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
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
Pilih: 2
Menilai tugas dari Tika
Masukkan nilai (0-100): 87
Nilai tugas Tika adalah 87
Nilai Biner Tugas: 1010111
```

e. Push dan commit kode program ke github

```
PS D:\Java\Praktikum-ASD> git commit -m "Jobsheet 9 Percobaan 2 Stack Konversi Biner"
[main 45cea44] Jobsheet 9 Percobaan 2 Stack Konversi Biner
3 files changed, 51 insertions(+)
create mode 100644 Jobsheet9/StackKonversi11.java
PS D:\Java\Praktikum-ASD> git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.05 KiB | 269.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/ghazwanz/Praktikum-ASD.git
8fc3aa9..45cea44 main -> main
branch 'main' set up to track 'origin/main'.
PS D:\Java\Praktikum-ASD>
```

f. Pertanyaan

1. Alur method konversiDesimalKeBiner berjalan dengan menerima parameter nilai dengan tipe int, kemudian memproses parameter tersebut dengan perulangan selama nilai > 0 maka nilai akan dimodulus dan disimpan hasil modulus ke variabel sisa, lalu variabel sisa akan dimasukkan ke dalam stack konversi dan membagi nilai dengan dua hingga nilai kurang dari 0. Setelah seluruh modulus telah dipush ke stack, dilakukan untuk perulangan hingga stack konversi kosong untuk

mengeluarkan seluruh sisa bagi dari top hingga data pertama pada stack konversi untuk menyimpan hasil konversi dengan menggunakan method pop. Setelah itu hasil konversi biner yang telah disimpan pada string digunakan sebagai nilai kembalian pada method konversiDesimalKeBiner

2. Mengubah kondisi perulangan menjadi while(nilai !=0)

```
String konversiDesimalKeBiner(int nilai){
    StackKonversi11 stack = new StackKonversi11();
    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;
}
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
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
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
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
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
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
5. Melihat Tugas Terbawah
6. Melihat Tugas yang sudah dikumpulkan saat ini
Pilih: 2
Menilai tugas dari Tika
Masukkan nilai (0-100): 87
Nilai tugas Tika adalah 87
Nilai Biner Tugas: 101011
```

Saat dijalankan hasil output sama seperti saat dengan kondisi perulangan menggunakan while(nilai>0), karena kondisi nilai terendah saat pembagian yaitu 0 sehingga kondisi nilai > 0 dapat dikatakan sama dengan nilai != 0 jika nilai yang diinputkan merupakan bilangan positif, jika nilai diinputkan dengan bilangan negatif, perulangan dengan kondisi while(nilai > 0) tidak akan dijalankan karena nilai memiliki nilai kurang dari 0, sedangkan untuk kondisi perulangan while(nilai !=0)

maka perulangan akan tetap dijalankan seperti saat nilai merupakan bilangan positif, hanya saja untuk biner dengan nilai 1 akan menghasilkan nilai -1. Sehingga kondisi `while(nilai > 0)` dan `while(nilai != 0)` dapat dikatakan sama jika nilai yang diinputkan merupakan bilangan positif.

## 2. Latihan

Membuat sistem untuk perizinan mahasiswa dengan konsep stack

Class Surat11.java

```
package Jobsheet9;

public class Surat11 {
    String idSurat, namaMahasiswa, kelas;
    char jenisIzin;
    int durasi;

    Surat11() {
    };

    Surat11(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;
    }

    void tampilSurat() {
        System.out.println("ID Surat: " + idSurat);
        System.out.println("Nama Mahasiswa: " + namaMahasiswa);
        System.out.println("Kelas: " + kelas);
        String jenis = (jenisIzin == 'I') ? "Izin" : (jenisIzin == 'S') ?
"Sakit" : "Lainnya";
        System.out.printf("Jenis Izin: %s (%s)\n", jenisIzin, jenis);
        System.out.println("Durasi: " + durasi + " hari");
    }
}
```

Class StackSurat11.java

```

package Jobsheet9;

public class StackSurat11 {
    Surat11[] stack;
    int top,size;
    StackSurat11(int size) {
        this.size = size;
        stack = new Surat11[size];
        top = -1;
    }
    boolean isEmpty() {
        return top == -1;
    }
    boolean isFull() {
        return top == size - 1;
    }
    void push(Surat11 surat) {
        if (isFull())
            System.out.println("Stack penuh, surat tidak bisa ditambahkan " +
surat.idSurat);
        else {
            stack[++top] = surat;
            System.out.println("surat " + surat.idSurat + " berhasil
ditambahkan ke stack");
        }
    }
    Surat11 pop() {
        if (isEmpty()) {
            System.out.println("Stack kosong, tidak ada surat yang bisa
diambil");
            return null;
        } else {
            return stack[top--];
        }
    }
    Surat11 peek() {
        if (isEmpty()){
            System.out.println("Stack kosong, tidak ada surat yang bisa
dilihat");
            return null;
        }
        else return stack[top];
    }

    Surat11 searchStack(String mhs) {
        for (int i = top; i >= 0; i--) {
            if (stack[i].namaMahasiswa.equals(mhs))
                return stack[i];
        }
        System.out.println("Surat dengan nama " + mhs + " tidak ditemukan di
stack");
        return null;
    }
}

```

Class SuratDemo11.java



```

package Jobsheet9;
import java.util.Scanner;

public class SuratDemoll {
    public static void main(String[] args) {
        StackSurat11 stack = new StackSurat11(5);
        Scanner scan = 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. Melihat Surat Izin Terakhir");
            System.out.println("4. Cari Surat Izin Mahasiswa");
            System.out.print("Pilih: ");
            pilih = scan.nextInt();
            scan.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("ID Surat: ");
                    String id = scan.nextLine();
                    System.out.print("Nama: ");
                    String nama = scan.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = scan.nextLine();
                    System.out.println("Jenis Izin (I/S/L): ");
                    System.out.println("I (Izin) | S (Sakit) | L
(Lainnya)");
                    System.out.print("Pilih: ");
                    char jenis = scan.nextLine().charAt(0);
                    jenis = Character.toUpperCase(jenis);
                    System.out.print("Durasi (hari): ");
                    int durasi = scan.nextInt();
                    Surat11 surat = new
Surat11(id,nama,kelas,jenis,durasi);
                    stack.push(surat);
                    break;
                case 2:
                    Surat11 diproses = stack.pop();
                    if (diproses != null) {
                        System.out.println("Memproses surat dari " +
diproses.namaMahasiswa);
                        System.out.print("Konfirmasi surat: ");
                        String konfirmasi = scan.nextLine();
                        if (konfirmasi.equalsIgnoreCase("ya")){
                            System.out.printf("Surat %s dari %s telah
diverifikasi\n",diproses.idSurat,diproses.namaMahasiswa);
                            diproses.tampilSurat();
                        }
                        else
                            System.out.printf("Surat %s dari %s
ditolak\n",diproses.idSurat,diproses.namaMahasiswa);
                        }
                    break;
            }
        }
    }
}

```

```

case 3:
    Surat11 lihat = stack.peek();
    if (lihat != null)
        System.out.println("Surat terakhir dikumpulkan oleh
" + lihat.namaMahasiswa);
    break;
case 4:
    System.out.print("Masukkan nama mahasiswa: ");
    String mhs = scan.nextLine();
    Surat11 cari = stack.searchStack(mhs);
    if (cari != null) {
        System.out.println("Surat ditemukan:");
        cari.tampilSurat();
    }
    break;
default:
    System.out.println("Pilihan tidak valid");
    break;
    }
} while (pilih > 0 && pilih < 5);
}
}

```

## Hasil output

```

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Melihat Surat Izin Terakhir
4. Cari Surat Izin Mahasiswa
Pilih: 1
ID Surat: S001
Nama: Dila
Kelas: 1A
Jenis Izin (I/S/L):
I (Izin) | S (Sakit) | L (Lainnya)
Pilih: I
Durasi (hari): 2
surat S001 berhasil ditambahkan ke stack

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Melihat Surat Izin Terakhir
4. Cari Surat Izin Mahasiswa
Pilih: 3
Surat terakhir dikumpulkan oleh Dila

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Melihat Surat Izin Terakhir
4. Cari Surat Izin Mahasiswa
Pilih: 4
Masukkan nama mahasiswa: Dila
Surat ditemukan:
ID Surat: S001
Nama Mahasiswa: Dila
Kelas: 1A
Jenis Izin: I (Izin)
Durasi: 2 hari

```

```

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Melihat Surat Izin Terakhir
4. Cari Surat Izin Mahasiswa
Pilih: 2
Memproses surat dari Dila
Konfirmasi surat: ya
Surat S001 dari Dila telah diverifikasi
ID Surat: S001
Nama Mahasiswa: Dila
Kelas: 1A
Jenis Izin: I (Izin)
Durasi: 2 hari

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Melihat Surat Izin Terakhir
4. Cari Surat Izin Mahasiswa
Pilih: 3
Stack kosong, tidak ada surat yang bisa dilihat

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Melihat Surat Izin Terakhir
4. Cari Surat Izin Mahasiswa
Pilih: 4
Masukkan nama mahasiswa: Dila
Stack kosong, tidak ada surat yang bisa dicari

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Melihat Surat Izin Terakhir
4. Cari Surat Izin Mahasiswa
Pilih: 2
Stack kosong, tidak ada surat yang bisa diambil

```

```

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Melihat Surat Izin Terakhir
4. Cari Surat Izin Mahasiswa
Pilih: 1
ID Surat: S002
Nama: Erik
Kelas: 1B
Jenis Izin (I/S/L):
I (Izin) | S (Sakit) | L (Lainnya)
Pilih: S
Durasi (hari): 3
surat S002 berhasil ditambahkan ke stack

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Melihat Surat Izin Terakhir
4. Cari Surat Izin Mahasiswa
Pilih: 2
Memproses surat dari Erik
Konfirmasi surat: tidak
Surat S002 dari Erik ditolak

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Melihat Surat Izin Terakhir
4. Cari Surat Izin Mahasiswa
Pilih: 22
Pilihan tidak valid

```

Commit dan Push kode program ke github

```

PS D:\Java\Praktikum-ASD> git add .
PS D:\Java\Praktikum-ASD> git commit -m "Jobsheet 9 Latihan Stack Sistem Layanan Perizinan Mahasiswa"
[main 998a0de] Jobsheet 9 Latihan Stack Sistem Layanan Perizinan Mahasiswa
3 files changed, 157 insertions(+)
create mode 100644 Jobsheet9/StackSurat11.java
create mode 100644 Jobsheet9/Surat11.java
create mode 100644 Jobsheet9/SuratDemo11.java
PS D:\Java\Praktikum-ASD> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.02 KiB | 690.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/ghazwanz/Praktikum-ASD.git
45cea44..998a0de main -> main
branch 'main' set up to track 'origin/main'.
PS D:\Java\Praktikum-ASD>

```