

**LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**  
**JOBSHEET STACK**



**VANESA MARDIANA PUTRI**

**244107020129**

**KELAS TI-1B**

**PRODI D-IV TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

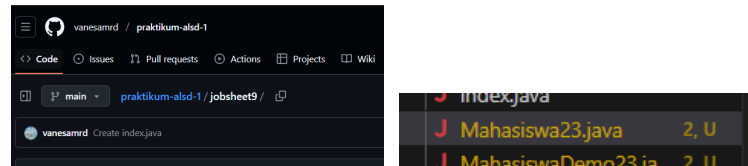
## A. PERCOBAAN

### 1. Percobaan 1 : Mahasiswa Mengumpulkan Tugas

#### a. Class Mahasiswa

Cara :

1. Buat folder baru bernama Jobsheet9 di dalam repository Praktikum ASD. Buat file baru, beri nama Mahasiswa.java



2. Lengkapi class Mahasiswa dengan atribut yang telah digambarkan di dalam class diagram Mahasiswa, yang terdiri dari atribut nama, nim, kelas, dan nilai.

```
public class Mahasiswa23 {  
    String nama;  
    String nim;  
    String kelas;  
    double nilai;  
}
```

3. Tambahkan konstruktor berparameter pada class Mahasiswa sesuai dengan class diagram Mahasiswa. Berikan nilai default nilai = -1 sebagai nilai awal ketika tugas belum dinilai.

```
Mahasiswa23() {  
}  
  
Mahasiswa23(String nama, String nim, String kelas) {  
    this.nama = nama;  
    this.nim = nim;  
    this.kelas = kelas;  
    nilai = -1;  
}
```

4. Tambahkan method tugasDinilai() yang digunakan untuk mengeset nilai ketika dilakukan penilaian tugas mahasiswa.

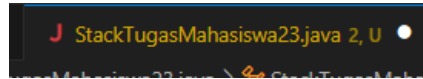
```
void tugasDinilai(int nilai) {  
}
```

#### b. Class StackTugasMahasiswa

Cara :

1. Setelah membuat class Mahasiswa, selanjutnya perlu dibuat class StackTugasMahasiswa.java sebagai tempat untuk

mengelola tumpukan tugas. Class StackTugasMahasiswa merupakan penerapan dari struktur data Stack.



2. Lengkapi class StackTugasMahasiswa dengan atribut yang telah digambarkan di dalam class diagram StackTugasMahasiswa, yang terdiri dari atribut stack, size, dan top.

```
public class StackTugasMahasiswa23 {  
    Mahasiswa23[] stack;  
    int top;  
    int size;  
}
```

3. Tambahkan konstruktor berparameter pada class StackTugasMahasiswa untuk melakukan inisialisasi kapasitas maksimum data tugas mahasiswa yang dapat disimpan di dalam Stack, serta mengeset indeks awal dari pointer top.

```
public StackTugasMahasiswa23(int size) {  
    this.size = size;  
    stack = new Mahasiswa23[size];  
    top = -1;  
}
```

4. Selanjutnya, buat method isFull bertipe boolean untuk mengecek apakah tumpukan tugas mahasiswa sudah terisi penuh sesuai kapasitas.

```
public boolean isFull() {  
    if (top == size - 1) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

5. Pada class StackTugasMahasiswa, buat method isEmpty bertipe boolean untuk mengecek apakah tumpukan tugas masih kosong.

```
public boolean isEmpty() {  
    if (top == -1) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

6. Untuk dapat menambahkan berkas tugas ke dalam tumpukan Stack, maka buat method push. Method ini menerima parameter mhs yang berupa object dari class Mahasiswa.

```
public void push(Mahasiswa23 mhs) {
    if (isFull()) {
        top++;
        stack[top] = mhs;
    } else {
        System.out.println(x:"Stack penuh! Tidak bisa menambahkan tugas lagi.");
    }
}
```

7. Penilaian tugas mahasiswa yang dilakukan oleh dosen dilakukan dengan menggunakan method pop untuk mengeluarkan tugas yang akan dinilai. Method ini tidak menerima parameter apapun namun mempunyai nilai kembalian berupa object dari class Mahasiswa.

```
public Mahasiswa23 pop() {
    if (!isEmpty()) {
        Mahasiswa23 m = stack[top];
        top--;
        return m;
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas untuk dinilai.");
        return null;
    }
}
```

8. Buat method peek untuk dapat mengecek tumpukan tugas mahasiswa yang berada di posisi paling atas.

```
public Mahasiswa23 peek() {
    if (!isEmpty()) {
        return stack[top];
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan.");
        return null;
    }
}
```

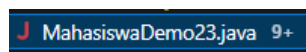
9. Tambahkan method print untuk dapat menampilkan semua daftar tugas mahasiswa pada Stack.

```
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:"");
}
```

#### c. Class Utama

Cara :

1. Buat file baru, beri nama MahasiswaDemo.java



2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi main.

```
public class MahasiswaDemo23 {
    Run | Debug
    public static void main(String[] args) {
```

- Di dalam fungsi main, lakukan instansiasi object StackTugasMahasiswa bernama stack dengan nilai parameternya adalah 5.

```
StackTugasMahasiswa23 stack = new StackTugasMahasiswa23(size:5);
Scanner scan = new Scanner(System.in);
```

- Deklarasikan Scanner dengan nama variabel scan dan variabel pilih bertipe int.

```
StackTugasMahasiswa23 stack = new StackTugasMahasiswa23(size:5);
Scanner scan = new Scanner(System.in);
int pilih;
```

- Tambahkan menu untuk memfasilitasi pengguna dalam memilih operasi Stack dalam mengelola data tugas mahasiswa menggunakan struktur perulangan do-while.

```
do {
    System.out.println(x:"\nMenu:");    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 : ");    Replace this use of System.out by a logger.
    pilih = scan.nextInt();
    scan.nextLine();

    switch (pilih) {
        case 1:
            System.out.print(s:"Nama: ");    Replace this use of System.out by a logger.
            String nama = scan.nextLine();
            System.out.print(s:"NIM: ");    Replace this use of System.out by a logger.
            String nim = scan.nextLine();
            System.out.print(s:"Kelas: ");    Replace this use of System.out by a logger.
            String kelas = scan.nextLine();
            Mahasiswa23 mhs = new Mahasiswa23(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 the platform-independent line separator.
            break;

        case 2:
            Mahasiswa23 dinilai = stack.pop();
            if (dinilai != null) {
                System.out.println("Menilai tugas dari " + dinilai.nama);    Replace this use of System.out by a logger.
                System.out.print(s:"Masukkan nilai (0-100): ");    Replace this use of System.out by a logger.
                int nilai = scan.nextInt();
                dinilai.nilai = nilai;
                System.out.printf(format:"Nilai Tugas %s adalah %d\n", dinilai.nama, dinilai.nilai);    %n should be used in place of \n to produce the platform-independent line separator.
            } else {
                System.out.println(x:"Tidak ada tugas yang dapat dinilai.");    Replace this use of System.out by a logger.
            }
            break;
    }
} while (pilih >= 1 && pilih <= 4);

scan.close();
}
```

```
case 3:
    Mahasiswa23 lihat = stack.peek();
    if (lihat != null) {
        System.out.println("Tugas terakhir dikumpulkan oleh " + lihat.nama);    Replace this use of System.out by a logger.
    } else {
        System.out.println(x:"Tidak ada tugas yang dapat dilihat.");    Replace this use of System.out by a logger.
    }
    break;

case 4:
    System.out.println(x:"Daftar Semua Tugas");    Replace this use of System.out by a logger.
    System.out.println(x:"Nama\tNim\tKelas");    Replace this use of System.out by a logger.
    stack.print();
    break;

default:
    System.out.println(x:"Pilihan tidak valid");    Replace this use of System.out by a logger.
}

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

scan.close();
}
```

6. Compile dan run program.

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih : 1
Nama: Vanesa
NIM: 1001
Kelas: 1B
Tugas Vanesa 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: Dila
NIM: 1003
Kelas: 1B
Tugas Dila berhasil dikumpulkan
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih : 3
Tugas terakhir dikumpulkan oleh Dila
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih : 4
Daftar Semua Tugas
Nama  Nim  Kelas
Vanesa  1001  1B
Erik  1002  1B
Dila  1003  1B
```

Jawaban Pertanyaan :

1. Kode yang perlu dimodifikasi :

```
lic void print() {
    for (int i = 0; i > 0; i++) { "i" is incremented and will never reach "stop con
        System.out.println(stack[i].nama + "\t" + stack[i].nim + "\t" + stack[i].kelas)
    }
    System.out.println(x:""); Replace this use of System.out by a logger.
}
```

2. Jumlah data : 5

```
StackTugasMahasiswa23 stack = new StackTugasMahasiswa23(size:5);
Scanner scan = new Scanner(System.in);
```

- Melakukan pengecekan apakah stack penuh atau tidak apabila if tidak ada dan stack penuh maka akan terjadi error `ArrayIndexOutOfBoundsException`.

```
case 5:
    Mahasiswa23 bawah = stack.bottom();
    if (bawah != null) {
        System.out.println(x:"Tugas terbawah:");
        System.out.println(bawah.nama + " - " + bawah.nim);
    } else {
        System.out.println(x:"Stack masih kosong!");
    }
    break;
```

4.

```
public Mahasiswa23 bottom() {
    if (!isEmpty()) {
        return stack[0];
    } else {
        return null;
    }
}
```

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

5.

```
break;
case 6:
    System.out.println("Jumlah tugas terkumpul: " + stack.jumlahTugas());
    break;
default:
```

```
5. Tugas Terbawah
6. Jumlah Tugas Terkumpul
Pilih :
```

6.

## 2. Konversi Nilai Tugas ke Biner

Cara :

- Buka kembali file `StackTugasMahasiswa.java`, tambahkan method `konversiDesimalKeBiner` dengan menerima parameter kode bertipe `int`.

```
public String konversiDesimalKeBiner(int nilai) {
    StackKonversi23 stack = new StackKonversi23();
    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;
}
```

- Buat file baru bernama `StackKonversi.java`

```
StackKonversi23.java
StackKonversi23.java
StackSurat23.java
```

- Tambahkan empat method yaitu `isEmpty`, `isFull`, `push`, dan `pull` sebagai operasi utama Stack pada class `StackKonversi`.

```

public class StackKonversi23 {
    int[] tumpukanBiner;
    int size;
    int top;

    public StackKonversi23() {
        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");
        } else {
            top++;
            tumpukanBiner[top] = data;
        }
    }

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

```

4. Agar nilai tugas mahasiswa dikonversi ke dalam bentuk biner setelah dilakukan penilaian, maka tambahkan baris kode program pada method pop di class MahasiswaDemo.

```

        System.out.printf(format:"Nilai Tugas %s adalah %d\n", dinil);
        String biner = stack.konversiDesimalKeBiner(nilai);
        System.out.println("Nilai Biner Tugas : " + biner);
    }
    break;

```

5. Compile dan run program.

```

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih : 2
Menilai tugas dari Dila
Masukkan nilai (0-100): 98
Nilai Tugas Dila adalah 98
Nilai Biner Tugas : 1100010

```

Jawaban Pertanyaan :



1. Method konversiDesimalKeBiner bekerja dengan: membagi bilangan desimal secara berulang oleh 2 dan mencatat sisanya (0 atau 1), menyusun sisa bagi dari terakhir ke pertama untuk membentuk string biner, berhenti ketika bilangan sudah habis (kode = 0).
2. Mengubah kondisi perulangan menjadi while (kode != 0) hanya aman untuk bilangan desimal positif. Hasil konversinya akan sama seperti kondisi while (kode > 0). Namun, jika inputnya bilangan negatif, program akan terjebak dalam infinite loop karena nilai kode tidak pernah mencapai 0 secara benar.

## B. TUGAS

Main :

```
import java.util.Scanner;    Move this file to a named package.

public class SuratDemo23 {
    Run | Debug
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);    Resource leak: 'input' is never closed
        StackSurat23 stack = new StackSurat23(size:5);
        int pilih;

        do {
            System.out.println(x:"Pilihan Menu :");    Replace this use of System.out.println
            System.out.println(x:"1. Terima Surat Izin");    Replace this use of System.out.println
            System.out.println(x:"2. Proses Surat Izin");    Replace this use of System.out.println
            System.out.println(x:"3. Lihat Surat Izin Terakhir");    Replace this use of System.out.println
            System.out.println(x:"4. Cari Surat");    Replace this use of System.out.println
            System.out.println(x:"Pilih : ");    Replace this use of System.out.println
            pilih = input.nextInt();

            switch (pilih) {
                case 1:
                    System.out.println(x:"Masukkan data Surat Izin");    Replace this use of System.out.println
                    System.out.print(s:"Masukkan ID Surat Izin: ");    Replace this use of System.out.print
                    String idSurat = input.next();
                    System.out.print(s:"Masukkan Nama Mahasiswa: ");    Replace this use of System.out.print
                    String namaMahasiswa = input.next();
                    System.out.print(s:"Masukkan Kelas: ");    Replace this use of System.out.print
                    String kelas = input.next();
                    System.out.print(s:"Masukkan Jenis Izin: ");    Replace this use of System.out.print
                    char jenisIzin = input.next().charAt(index:0);
                    if (jenisIzin != 'S' && jenisIzin != 'I') {
                        System.out.println(x:"Jenis Izin tidak valid.");    Replace this use of System.out.println
                        break;
                    }

                    case 2:
                        if (stack.isEmpty()) {
                            System.out.println(x:"Tidak ada Surat Izin yang dikumpulkan.");    Replace this use of System.out.println
                            break;
                        }
                        Surat23 suratDiproses = stack.pop();
                        System.out.println("Memproses Surat Izin Dari " + suratDiproses.namaMahasiswa);
                        suratDiproses.cetakSurat();
                        break;

                    case 3:
                        if (stack.isEmpty()) {
                            System.out.println(x:"Tidak ada Surat Izin yang dikumpulkan.");    Replace this use of System.out.println
                            break;
                        }
                        Surat23 suratTerakhir = stack.peek();
                        System.out.println("Surat Izin Terakhir : " + suratTerakhir.namaMahasiswa);    Repl
                        suratTerakhir.cetakSurat();
                        break;

                    case 4:
                        System.out.println(x:"Cari Surat Berdasarkan Nama Mahasiswa");    Replace this use of System.out.println
                        System.out.print(s:"Masukkan Nama Mahasiswa: ");    Replace this use of System.out.print
                        String cari = input.next();
                        stack.cariSurat(cari);
                        break;
                    default:
                        System.out.println(x:"Pilihan tidak valid.");    Replace this use of System.out by
                        break;
                }
            } while (pilih >= 1 && pilih <= 4);
        }
    }
}
```

Surat23.java :

```
public class Surat23 {    Move this file to a named package.
    String idSurat, namaMahasiswa, kelas;    Declare "namaMahasiswa" and all following declarations on
    int durasi;
    char jenisIzin;

    public Surat23(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 cetakSurat() {
        System.out.println("ID Surat      : " + idSurat);    Replace this use of System.out by a logg
        System.out.println("Nama Mahasiswa : " + namaMahasiswa);    Replace this use of System.out by
        System.out.println("Kelas        : " + kelas);    Replace this use of System.out by a logger
        System.out.println("Jenis Izin    : " + jenisIzin);    Replace this use of System.out by a lo
        System.out.println("Durasi       : " + durasi + " hari");    Replace this use of System.out
        System.out.println(x:"-----");    Replace this use of System.out by a
    }
}
```

StackSurat23.java :

```
1 public class StackSurat23 {    Move this file to a named package.
2     Surat23[] stack;
3     int top, size;    Declare "size" on a separate line.
4
5     public StackSurat23(int size) {
6         this.size = size;
7         stack = new Surat23[size];
8         top = -1;
9     }
10
11     public boolean isEmpty() {    Rename this method name to match the regular expression '^[a-z][a-zA-
12         if (top == -1) {    Replace this if-then-else statement by a single return statement.
13             return true;
14         } else {
15             return false;
16         }
17     }
18
19     public boolean IsFull() {    Rename this method name to match the regular expression '^[a-z][a-zA-2
20         if (top == size -1) {    Replace this if-then-else statement by a single return statement.
21             return true;
22         } else {
23             return false;
24         }
25     }
26
27     public void push(Surat23 surat) {
28         if (!IsFull()) {
29             top++;
30             stack[top] = surat;
31         } else {
32             System.out.println(x:"Isi Stack Penuh! Tidak bisa menambahkan data lagi");    Replace this
33         }
34     }
35
36     public Surat23 pop() {
37         if (!isEmpty()) {
38             Surat23 n = stack[top];
39             top--;
40             return n;
41         } else {
42             System.out.println(x:"Stack masih kosong! Tidak ada surat yang bisa diverifikasi");    Repl
43             return null;
44         }
45     }
46 }
```

```

public Surat23 peek() {
    if (!isEmpty()) {
        return stack[top];
    } else {
        System.out.println(x:"Stack masih kosong! Tidak ada surat yang dikumpulkan.");
        return null;
    }
}

public void cariSurat(String cari) {
    if (isEmpty()) {
        System.out.println(x:"Stack kosong! Tidak ada surat yang dikumpulkan.");
        return;
    } else {
        boolean found = false;
        for (int i = 0; i <= top; i++) {
            if (stack[i].namaMahasiswa.equalsIgnoreCase(cari)) {
                System.out.println(
                    x:"-----");
                System.out.println("Surat yang ditemukan: " + stack[i].namaMahasiswa);
                stack[i].cetakSurat();
                found = true;
            }
        }
        if (!found) {
            System.out.printf(format:"Surat dengan nama %s tidak ditemukan.", cari);
        }
    }
}
}

```

Hasil run :

```

.\java\jdk_16\praktikum-alsd-1_fa7eb693\bin 'SuratDemo23'
Pilihan Menu :
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
Pilih :
1
Masukkan data Surat Izin
Masukkan ID Surat Izin: 1
Masukkan Nama Mahasiswa: vanesa
Masukkan Kelas: 1B
Masukkan Jenis Izin: I
Masukkan Durasi Surat: 4
Pilihan Menu :
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
Pilih :
1
Masukkan data Surat Izin
Masukkan ID Surat Izin: 2
Masukkan Nama Mahasiswa: Putri
Masukkan Kelas: 1A
Masukkan Jenis Izin: S
Masukkan Durasi Surat: 1
Pilihan Menu :
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
Pilih :
3
Surat Izin Terakhir : Putri
ID Surat : 2
Nama Mahasiswa : Putri
Kelas : 1A
Jenis Izin : S
Durasi : 1 hari
-----

```

```

-----
Pilihan Menu :
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
Pilih :
2
Memproses Surat Izin Dari Putri
ID Surat : 2
Nama Mahasiswa : Putri
Kelas : 1A
Jenis Izin : S
Durasi : 1 hari
-----

```

```
Pilihan Menu :
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
Pilih :
4
Cari Surat Berdasarkan Nama Mahasiswa
Masukkan Nama Mahasiswa: vanesa
-----
Surat yang ditemukan: vanesa
ID Surat      : 1
Nama Mahasiswa : vanesa
Kelas        : 1B
Jenis Izin    : I
Durasi        : 4 hari
-----
```