# JOBSHEET 9

# LAPORAN HASIL PRAKTIKUM

# ALGORITMA DAN STRUKTUR

# DATA

MUHAMMAD DAFFI FIROS ZAIDAN

244107020182

TI 1E

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2025**

## Percobaan 1: Mahasiswa Mengumpulkan Tugas

Kode Program

Class Mahasiswa

```
public class Mahasiswa17 {
    String nim, nama, kelas;
    int nilai;

    public Mahasiswa17() {

    }
    public Mahasiswa17(String nama, String nim, String kelas ) {
        this.nama = nama;
        this.nim = nim;
        this.kelas  = kelas;
        nilai = -1;

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

    }
}
```

## Class StackTugasMahasiswa

```
public class StackTugasMahasiswa17 {
    Mahasiswa17[] stack;
    int size, top;

    public StackTugasMahasiswa17(int size) {
        this.size = size;
        stack = new Mahasiswa17[size];
        top = -1;
    }

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

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

    void push(Mahasiswa17 mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan tugas lagi.");
        }
    }

    Mahasiswa17 pop() {
        if (!isEmpty()) {
            Mahasiswa17 m = stack[top];
            top--;
            return m;
        } else {
            System.out.println("Stack kosong, Tidak ada tugas untuk dinilai.");
            return null;
        }
    }

    Mahasiswa17 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 = 0; i <= top; i++) {
            System.out.println(stack[i].nama + "\t" + stack[i].nim + "\t" +
stack[i].kelas);
        }
    }
}
```

Class MahasiswaDemo

```
import java.util.Scanner;
public class MahasiswaDemo17 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StackTugasMahasiswa17 stack = new StackTugasMahasiswa17(5);
        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 = sc.nextInt();
            sc.nextLine();
            switch (pilih) {
                case 1:
                    System.out.print("Nama: ");
                    String nama = sc.nextLine();
                    System.out.print("NIM: ");
                    String nim = sc.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = sc.nextLine();
                    Mahasiswa17 mhs = new Mahasiswa17(nama, nim,
kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil
dikumpulkan\n", mhs.nama);
                    break;
                case 2:
                    Mahasiswa17 dinilai = stack.pop();
                    if (dinilai != null) {
                        System.out.println("Menilai tugas dari " +
dinilai.nama);
                        System.out.print("Masukkan nilai (0-100):
");
                        int nilai = sc.nextInt();
                        dinilai.tugasDinilai(nilai);
                        System.out.printf("Nilai Tugas %s adalah
%d\n", dinilai.nama, nilai);
                    }
                    break;
                case 3:
                    Mahasiswa17 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);
    }
```

## Output

```
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

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih:
```

## Pertanyaan

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

   Jawaban:

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

   Analisa:

   Bagian yang perlu diperbaiki adalah **fungsi print() pada class StackTugasMahasiswa**.Pada fungsi tersebut, urutan perulangan harus diubah dari menaik (i++) menjadi menurun (i--).Hal ini sesuai dengan prinsip kerja stack (LIFO), sehingga data yang terakhir masuk (top) akan tercetak paling atas, dan data yang pertama masuk akan tercetak paling bawah.

2. Berapa banyak data tugas mahasiswa yang dapat ditampung di dalam Stack? Tunjukkan potongan kode programnya!

   Jawaban:

```
StackTugasMahasiswa17 stack = new StackTugasMahasiswa17(5);
```

   Analisa:

   **stack dapat menampung maksimal 5 data tugas mahasiswa**.

3. Mengapa perlu pengecekan kondisi !isFull() pada method push? Kalau kondisi if-else tersebut dihapus, apa dampaknya?

   Jawaban:

   Pengecekan !isFull() pada method push penting agar data tidak ditambahkan saat stack sudah penuh. Jika pengecekan ini dihapus, program bisa error karena mencoba menambah data melebihi kapasitas stack, sehingga terjadi ArrayIndexOutOfBoundsException dan program bisa berhenti tiba-tiba

4. Modifikasi kode program pada class MahasiswaDemo dan StackTugasMahasiswa sehinggapengguna juga dapat melihat mahasiswa yang pertama kali mengumpulkan tugas melalui operasi lihat tugas terbawah!

   Jawaban:
   Class StackTugasMahasiswa ditambahkan fungsi berikut

```
public Mahasiswa17 peekBottom() {
        if (!isEmpty()) {
            return stack[0];
        } else {
            System.out.println("Stack kosong! Tidak ada tugas yang
dikumpulkan.");
            return null;
        }
    }
```

   Class MahasiswoDemo ditambahkan Menu No.5 dan Switch Case No. 5

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

```
case 5:
    Mahasiswa17 terbawah = stack.peekBottom();
    if (terbawah != null) {
        System.out.println("Tugas pertama dikumpulkan oleh " +
terbawah.nama);
    }
    break;
```

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

   Jawaban:

   pada class StackTugasMahasiswa Menambahkan fungsi jumlahTugas

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

   Pada Class MahasiswaDemo menambahkan menu no.6 dan switch casenya

6. Commit dan push kode program ke Github

## Percobaan 2: Konversi Nilai Tugas ke Biner

Kode Program

Class StackTugasMahasiswa

Menambahkan fungsi konversi biner

```
public String konversiDesimalKeBiner(int nilai) {
        StackKonversi17 stack = new StackKonversi17();
        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;
    }
}
```

Class StackKonversi

```
public class StackKonversi17 {
    int[] tumpukanBiner;
    int size;
    int top;

    public StackKonversi17() {
        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;
        }
    }
}
```

Class MahasiswaDemo

Menambahkan kode berikut pada class

```
String biner = stack.konversiDesimalKeBiner(nilai);
System.out.println("Nilai Biner Tugas: " + biner);
```

Output

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

## Pertanyaan

1. Jelaskan alur kerja dari method konversiDesimalKeBiner!

   Jawaban:

   Method konversiDesimalKeBiner bekerja dengan cara membagi angka desimal dengan 2 berulang-ulang, lalu setiap sisa pembagiannya dimasukkan ke stack. Setelah itu, sisa-sisa tadi diambil dari stack satu per satu sehingga membentuk angka biner yang benar. Stack dipakai agar urutan biner-nya jadi dari belakang ke depan.

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

   Jawaban:

   Jika kondisi perulangan diubah menjadi while (kode != 0), hasilnya akan tetap benar asalkan variabel kode di-update di dalam perulangan, misalnya dengan kode = kode / 2;. Hal ini karena perulangan akan terus berjalan selama nilai kode belum menjadi nol, sama seperti saat menggunakan while (nilai > 0). Namun, jika variabel kode tidak diubah di dalam loop, maka perulangan tidak akan pernah berhenti (infinite loop). Jadi, yang terpenting adalah variabel pada kondisi perulangan harus selalu di-update agar loop bisa selesai dan hasil konversi tetap benar.

## Latihan Praktikum

Kode Program
Class Surat

```
public class Surat17 {
    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin;
    int durasi;

    public Surat17() {}

    public Surat17(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;
    }
}
```

## Class StackSurat

```
public class StackSurat17 {
    Surat17[] stack;
    int size, top;

    public StackSurat17(int size) {
        this.size = size;
        stack = new Surat17[size];
        top = -1;
    }

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

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


    public void push(Surat17 surat) {
        if (!isFull()) {
            top++;
            stack[top] = surat;
            System.out.println("Surat berhasil diterima.");
        } else {
            System.out.println("Stack penuh, tidak bisa menerima surat baru.");
        }
    }

    public Surat17 pop() {
        if (!isEmpty()) {
            Surat17 s = stack[top];
            top--;
            System.out.println("Surat atas nama " + s.namaMahasiswa + " telah diproses.");
            return s;
        } else {
            System.out.println("Tidak ada surat yang bisa diproses.");
            return null;
        }
    }

    public Surat17 peek() {
        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Tidak ada surat di stack.");
            return null;
        }
    }


    public void cariSurat(String nama) {
        boolean ditemukan = false;
        for (int i = 0; i <= top; i++) {
            if (stack[i].namaMahasiswa.equalsIgnoreCase(nama)) {
                System.out.println("\nSurat Ditemukan :");
                System.out.println("ID Surat        : " + stack[i].idSurat);
                System.out.println("Nama Mahasiswa  : " + stack[i].namaMahasiswa);
                System.out.println("Kelas           : " + stack[i].kelas);
                System.out.println("Jenis Perizinan : " + stack[i].jenisIzin);
                System.out.println("Lama Waktu Izin : " + stack[i].durasi);
                ditemukan = true;
                break;
            }
        }
        if (!ditemukan) {
            System.out.println("\nSurat atas nama \"" + nama + "\" tidak ditemukan.");
        }
    }
}
```

Class MainSurat

```
import java.util.Scanner;

public class MainSurat17 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StackSurat17 stack = new StackSurat17(10);
        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. Lihat Surat Izin Terakhir");
            System.out.println("4. Cari Surat Izin (berdasarkan nama
mahasiswa)");
            System.out.println("5. Keluar");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("ID Surat: ");
                    String id = sc.nextLine();
                    System.out.print("Nama Mahasiswa: ");
                    String nama = sc.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = sc.nextLine();
                    System.out.print("Jenis Izin (S/I): ");
                    char jenis = sc.nextLine().charAt(0);
                    System.out.print("Durasi (hari): ");
                    int durasi = sc.nextInt();
                    sc.nextLine();
                    Surat17 surat = new Surat17(id, nama, kelas, jenis,
durasi);
                    stack.push(surat);
                    break;
                case 2:
                    stack.pop();
                    break;
                case 3:
                    Surat17 terakhir = stack.peek();
                    if (terakhir != null) {
                        System.out.println("Surat terakhir dikumpulkan
oleh : " + terakhir.namaMahasiswa);
                    }
                    break;
                case 4:
                    System.out.print("Masukkan nama mahasiswa yang
dicari: ");
                    String cari = sc.nextLine();
                    stack.cariSurat(cari);
                    break;
                case 5:
                    System.out.println("Keluar...");
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilih != 5);
    }
}
```

Output

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Keluar
Pilih: 1
ID Surat: 100
Nama Mahasiswa: Daffi
Kelas: 1E
Jenis Izin (S/I): s
Durasi (hari): 1
Surat berhasil diterima.

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Keluar
Pilih: 3
Surat terakhir dikumpulkan oleh : Daffi
```

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Keluar
Pilih: 4
Masukkan nama mahasiswa yang dicari: Daffi

Surat Ditemukan :
ID Surat        : 100
Nama Mahasiswa  : Daffi
Kelas           : 1E
Jenis Perizinan : s
Lama Waktu Izin : 1

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Keluar
Pilih: 2
Surat atas nama Daffi telah diproses.
```

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Keluar
Pilih: 3
Tidak ada surat di stack.

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin (berdasarkan nama mahasiswa)
5. Keluar
Pilih: 5
Keluar...
PS D:\Tugas Kuliah\SEMESTER 2\ASLD\Praktikum ASD\Praktikum-ASD>
```