

Laporan Hasil Pratikum
Algoritma Dan Struktur Data

Jobsheet 9



Nama : Zacky Rio Orlando

NIM : 244107020086

Kelas : 1E

Program Studi D-IV Teknik Informatika

Jurusan Teknologi Informasi

Praktikum

2025

2.1 Percobaan 1 : Mahasiswa Mengumpulkan Tugas

Class Mahasiswa

```
public class Mahasiswa27 {  
    String nim, nama, kelas;  
    int nilai;  
  
    Mahasiswa27 (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 StackTugasMahasiswa27

```
public class StackTugasMahasiswa27 {
    Mahasiswa27[] stack;
    int top;
    int size;

    public StackTugasMahasiswa27 (int size) {
        this.size = size;
        stack = new Mahasiswa27[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(Mahasiswa27 mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan tugas lagi.");
        }
    }

    public Mahasiswa27 pop() {
        if (!isEmpty()) {
            Mahasiswa27 m = stack[top];
            top--;
            return m;
        } else {
            System.out.println("Stack kosong! Tidak ada tugas untuk dinilai.");
            return null;
        }
    }

    public Mahasiswa27 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);
        }
        System.out.println("");
    }
}
```

Class MahasiswaDemo27

```
import java.util.Scanner;
public class MahasiswaDemo27 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int pilih;
        StackTugasMahasiswa27 stack = new StackTugasMahasiswa27(5);

        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();
                    Mahasiswa27 mhs = new Mahasiswa27(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n",
mhs.nama);
                    break;
                case 2:
                    Mahasiswa27 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 :
                    Mahasiswa27 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);
    }
}
```

Hasil Outputnya

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
```

2.1.3 Pertanyaan

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

Jawaban :

Class Mahasiswa27

```
public class Mahasiswa27 {  
    String nim, nama, kelas;  
    int nilai;  
  
    Mahasiswa27 (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 StackTugasMahasiswa27

```
public class StackTugasMahasiswa27 {
    Mahasiswa27[] stack;
    int top;
    int size;

    public StackTugasMahasiswa27 (int size) {
        this.size = size;
        stack = new Mahasiswa27[size];
        top = -1;
    }

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

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

    public void push(Mahasiswa27 mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan tugas
lagi.");
        }
    }

    public Mahasiswa27 pop() {
        if (!isEmpty()) {
            Mahasiswa27 m = stack[top];
            top--;
            return m;
        } else {
            System.out.println("Stack kosong! Tidak ada tugas untuk
dinilai.");
            return null;
        }
    }

    public Mahasiswa27 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();
    }
}
```


Class MahasiswaDemo27

```
import java.util.Scanner;

public class MahasiswaDemo27 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int pilih;
        StackTugasMahasiswa27 stack = new StackTugasMahasiswa27(5);

        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();
                    Mahasiswa27 mhs = new Mahasiswa27(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n", mhs.nama);
                    break;
                case 2:
                    Mahasiswa27 dinilai = stack.pop();
                    if (dinilai != null) {
                        System.out.println("Menilai tugas dari " + dinilai.nama);
                        int nilai;
                        do {
                            System.out.print("Masukkan nilai (0-100): ");
                            nilai = scan.nextInt();
                            if (nilai < 0 || nilai > 100) {
                                System.out.println("Nilai tidak valid. Harus antara
0 hingga 100.");
                            }
                        } while (nilai < 0 || nilai > 100);
                        dinilai.tugasDinilai(nilai);
                        System.out.printf("Nilai Tugas %s adalah %d\n",
dinilai.nama, nilai);
                    }
                    break;
                case 3 :
                    Mahasiswa27 lihat = stack.peek();
                    if (lihat != null) {
                        System.out.println("Tugas terakhir dikumpulkan oleh " +
lihat.nama);
                    }
                    break;
                case 4 :
                    System.out.println("=====");
                    System.out.println("Daftar Semua Tugas yang Terkumpul:");
                    System.out.println("Nama\tNIM\tKelas");
                    stack.print();
                    break;
                default :
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilih >= 1 && pilih <= 4);
    }
}
```

Hasil 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 yang Terkumpul:

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 yang Terkumpul:

Nama	NIM	Kelas
Erik	1002	1B
Dila	1001	1A

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

Jawaban :

Stack bisa menampung maksimal 5 data tugas mahasiswa

Potongan kodenya, pada Class MahasiswaDemo27

```
StackTugasMahasiswa27 stack = new StackTugasMahasiswa27(5);
```

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

Jawaban :

Pengecekan **!isFull()** pada method **push()** penting untuk memastikan bahwa stack belum penuh sebelum menambahkan data baru. Jika pengecekan ini dihapus, saat stack sudah mencapai kapasitas maksimum dan kita tetap mencoba menambahkan data, program akan mencoba mengakses indeks di luar batas array, yang akan menyebabkan error **ArrayIndexOutOfBoundsException**. Selain itu, hal ini bisa merusak urutan dan struktur data stack. Oleh karena itu, pengecekan ini berfungsi untuk mencegah penambahan data berlebih dan menjaga agar program tetap berjalan dengan aman.

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!

Jawaban :

Class Mahasiswa27

```
public class Mahasiswa27 {  
    String nim, nama, kelas;  
    int nilai;  
  
    Mahasiswa27 (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 StackTugasMahasiswa27

```
public class StackTugasMahasiswa27 {
    Mahasiswa27[] stack;
    int top;
    int size;

    public StackTugasMahasiswa27 (int size) {
        this.size = size;
        stack = new Mahasiswa27[size];
        top = -1;
    }

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

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

    public void push(Mahasiswa27 mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan tugas lagi.");
        }
    }

    public Mahasiswa27 pop() {
        if (!isEmpty()) {
            Mahasiswa27 m = stack[top];
            top--;
            return m;
        } else {
            System.out.println("Stack kosong! Tidak ada tugas untuk dinilai.");
            return null;
        }
    }

    public Mahasiswa27 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();
    }

    public Mahasiswa27 bottom() {
        if (!isEmpty()) {
            return stack[0];
        } else {
            System.out.println("Stack kosong! Tidak ada tugas yang dikumpulkan.");
            return null;
        }
    }
}
```

Class MahasiswaDemo27

```
import java.util.Scanner;

public class MahasiswaDemo27 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int pilih;
        StackTugasMahasiswa27 stack = new StackTugasMahasiswa27(5);

        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.println("5. Melihat Tugas Terbawah");
            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();
                    Mahasiswa27 mhs = new Mahasiswa27(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n", mhs.nama);
                    break;
                case 2:
                    Mahasiswa27 dinilai = stack.pop();
                    if (dinilai != null) {
                        System.out.println("Menilai tugas dari " + dinilai.nama);
                        int nilai;
                        do {
                            System.out.print("Masukkan nilai (0-100): ");
                            nilai = scan.nextInt();
                            if (nilai < 0 || nilai > 100) {
                                System.out.println("Nilai tidak valid. Harus antara 0 hingga
100.");
                            }
                        } while (nilai < 0 || nilai > 100);
                        dinilai.tugasDinilai(nilai);
                        System.out.printf("Nilai Tugas %s adalah %d\n", dinilai.nama, nilai);
                    }
                    break;
                case 3:
                    Mahasiswa27 lihat = stack.peek();
                    if (lihat != null) {
                        System.out.println("Tugas terakhir dikumpulkan oleh " + lihat.nama);
                    }
                    break;
                case 4:
                    System.out.println("=====");
                    System.out.println("Daftar Semua Tugas yang Terkumpul:");
                    System.out.println("Nama\tNIM\tKelas");
                    stack.print();
                    break;
                case 5:
                    Mahasiswa27 terbawah = stack.bottom();
                    if (terbawah != null) {
                        System.out.println("Tugas pertama dikumpulkan oleh " + terbawah.nama);
                    }
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilih >= 1 && pilih <= 5);
    }
}
```

Hasil Output

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: 3

Tugas terakhir dikumpulkan oleh Erik

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

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

Jawaban :

Class Mahasiswa27

```
public class Mahasiswa27 {  
    String nim, nama, kelas;  
    int nilai;  
  
    Mahasiswa27 (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 StackTugasMahasiswa27

```
public class StackTugasMahasiswa27 {
    Mahasiswa27[] stack;
    int top;
    int size;

    public StackTugasMahasiswa27 (int size) {
        this.size = size;
        stack = new Mahasiswa27[size];
        top = -1;
    }

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

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

    public void push(Mahasiswa27 mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan tugas lagi.");
        }
    }

    public Mahasiswa27 pop() {
        if (!isEmpty()) {
            Mahasiswa27 m = stack[top];
            top--;
            return m;
        } else {
            System.out.println("Stack kosong! Tidak ada tugas untuk dinilai.");
            return null;
        }
    }

    public Mahasiswa27 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();
    }

    public Mahasiswa27 bottom() {
        if (!isEmpty()) {
            return stack[0];
        } else {
            System.out.println("Stack kosong! Tidak ada tugas yang dikumpulkan.");
            return null;
        }
    }

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

Class MahasiswaDemo27

```
import java.util.Scanner;

public class MahasiswaDemo27 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int pilih;
        StackTugasMahasiswa27 stack = new StackTugasMahasiswa27(5);

        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.println("5. Melihat Tugas Terbawah");
            System.out.println("6. Lihat Jumlah Tugas yang Berkumpul");
            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();
                    Mahasiswa27 mhs = new Mahasiswa27(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n", mhs.nama);
                    break;
                case 2:
                    Mahasiswa27 dinilai = stack.pop();
                    if (dinilai != null) {
                        System.out.println("Menilai tugas dari " + dinilai.nama);
                        int nilai;
                        do {
                            System.out.print("Masukkan nilai (0-100): ");
                            nilai = scan.nextInt();
                            if (nilai < 0 || nilai > 100) {
                                System.out.println("Nilai tidak valid. Harus antara 0 hingga 100.");
                            }
                        } while (nilai < 0 || nilai > 100);
                        dinilai.tugasDinilai(nilai);
                        System.out.printf("Nilai Tugas %s adalah %d\n", dinilai.nama, nilai);
                    }
                    break;
                case 3:
                    Mahasiswa27 lihat = stack.peek();
                    if (lihat != null) {
                        System.out.println("Tugas terakhir dikumpulkan oleh " + lihat.nama);
                    }
                    break;
                case 4:
                    System.out.println("=====");
                    System.out.println("Daftar Semua Tugas yang Berkumpul:");
                    System.out.println("Nama\tNIM\tKelas");
                    stack.print();
                    break;
                case 5:
                    Mahasiswa27 terbawah = stack.bottom();
                    if (terbawah != null) {
                        System.out.println("Tugas pertama dikumpulkan oleh " + terbawah.nama);
                    }
                    break;
                case 6:
                    int jumlah = stack.jumlahTugas();
                    System.out.println("Jumlah tugas yang sudah dikumpulkan: " + jumlah);
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilih >= 1 && pilih <= 6);
    }
}
```

Hasil Output

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Lihat Jumlah Tugas yang Terkumpul
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. Lihat Jumlah Tugas yang Terkumpul
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. Lihat Jumlah Tugas yang Terkumpul
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. Lihat Jumlah Tugas yang Terkumpul
Pilih: 6
Jumlah tugas yang sudah dikumpulkan: 3
```

6. Commit dan push kode program ke Github

2.2 Percobaan 2 : Konversi Nilai Tugas ke Biner

Class Mahasiswa27

```
public class Mahasiswa27 {
    String nim, nama, kelas;
    int nilai;

    Mahasiswa27 (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 StackTugasMahasiswa27

```
public class StackTugasMahasiswa27 {
    Mahasiswa27[] stack;
    int top;
    int size;

    public StackTugasMahasiswa27 (int size) {
        this.size = size;
        stack = new Mahasiswa27[size];
        top = -1;
    }

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

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

    public void push(Mahasiswa27 mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan tugas
lagi.");
        }
    }

    public Mahasiswa27 pop() {
        if (!isEmpty()) {
            Mahasiswa27 m = stack[top];
            top--;
            return m;
        } else {
            System.out.println("Stack kosong! Tidak ada tugas untuk
dinilai.");
            return null;
        }
    }

    public Mahasiswa27 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();
    }

    public Mahasiswa27 bottom() {
        if (!isEmpty()) {
            return stack[0];
        } else {
            System.out.println("Stack kosong! Tidak ada tugas yang
dikumpulkan.");
            return null;
        }
    }

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

    public String konversiDesimalKeBiner (int nilai) {
        StackKonversi27 stack = new StackKonversi27();
        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 StackKonversi27

```
public class StackKonversi27 {
    int[] tumpukanBiner;
    int size;
    int top;

    public StackKonversi27() {
        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 MahasiswaDemo27

```
import java.util.Scanner;

public class MahasiswaDemo27 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int pilih;
        StackTugasMahasiswa27 stack = new StackTugasMahasiswa27(5);

        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.println("5. Melihat Tugas Terbawah");
            System.out.println("6. Lihat Jumlah Tugas yang Terkumpul");
            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();
                    Mahasiswa27 mhs = new Mahasiswa27(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n",
mhs.nama);
                    break;
```

```

        case 2:
            Mahasiswa27 dinilai = stack.pop();
            if (dinilai != null) {
                System.out.println("Menilai tugas dari " +
dinilai.nama);
                int nilai;
                do {
                    System.out.print("Masukkan nilai (0-100): ");
                    nilai = scan.nextInt();
                    if (nilai < 0 || nilai > 100) {
                        System.out.println("Nilai tidak valid.
Harus antara 0 hingga 100.");
                    }
                } while (nilai < 0 || nilai > 100);
                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;
        case 3 :
            Mahasiswa27 lihat = stack.peek();
            if (lihat != null) {
                System.out.println("Tugas terakhir dikumpulkan oleh
" + lihat.nama);
            }
            break;
        case 4 :
            System.out.println("=====");
            System.out.println("Daftar Semua Tugas yang
Terkumpul:");
            System.out.println("Nama\tNIM\tKelas");
            stack.print();
            break;
        case 5:
            Mahasiswa27 terbawah = stack.bottom();
            if (terbawah != null) {
                System.out.println("Tugas pertama dikumpulkan oleh " +
terbawah.nama);
            }
            break;
        case 6:
            int jumlah = stack.jumlahTugas();
            System.out.println("Jumlah tugas yang sudah
dikumpulkan: " + jumlah);
            break;
        default :
            System.out.println("Pilihan tidak valid.");
    }
    } while (pilih >= 1 && pilih <= 6);
}

```

Hasil Output

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

2.2.3 Pertanyaan

1. Jelaskan alur kerja dari method **konversiDesimalKeBiner**!

Jawaban :

Method **konversiDesimalKeBiner** berfungsi untuk mengubah nilai desimal menjadi biner dengan menggunakan struktur stack agar urutan biner tetap benar. Caranya yaitu, nilai desimal dibagi terus-menerus dengan 2, dan setiap sisa bagi (0 atau 1) disimpan ke dalam stack. Setelah nilai desimal menjadi 0, isi stack tersebut diambil satu per satu (pop) dan disusun menjadi string biner. Karena stack bekerja dengan prinsip LIFO (Last In First Out), hasil akhir biner akan memiliki urutan yang benar dari atas ke bawah.

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

Jawaban :

Hasilnya tetap sama karena kedua kondisi tersebut akan menghentikan perulangan saat nilai sudah menjadi nol. Selama nilai yang dikonversi adalah bilangan bulat positif, tidak ada perbedaan dalam hasil akhirnya.

2.4 Latihan Praktikum

Mahasiswa mengajukan surat izin (karena sakit atau keperluan lain) setiap kali tidak mengikuti perkuliahan. Surat terakhir yang masuk akan diproses atau divalidasi lebih dulu oleh admin Prodi.

Perhatikan class diagram berikut.

Surat<NoAbsen>
idSurat: String namaMahasiswa: String kelas: String jenisIzin: char durasi: int
Surat<NoAbsen>() Surat<NoAbsen>(idSurat: String, namaMahasiswa: String, kelas: String, jenisIzin: char, durasi: int)

Atribut jenisIzin digunakan untuk menyimpan keterangan izin mahasiswa (S: sakit atau I: izin keperluan lain) dan durasi untuk menyimpan lama waktu izin.

Berdasarkan class diagram tersebut, implementasikan class Surat dan tambahkan class StackSurat untuk mengelola data Surat. Pada class yang memuat method main, buat pilihan menu

berikut:

1. Terima Surat Izin untuk memasukkan data surat
2. Proses Surat Izin untuk memproses atau memverifikasi surat
3. Lihat Surat Izin Terakhir untuk melihat surat teratas
4. Cari Surat untuk mencari ada atau tidaknya surat izin berdasarkan **nama mahasiswa**

Jawaban :

Class Surat27

```
public class Surat27 {
    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin;
    int durasi;

    public Surat27() {}

    public Surat27(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          : " + namaMahasiswa);
        System.out.println("Kelas       : " + kelas);
        System.out.println("Jenis Izin   : " + (jenisIzin == 'S' ? "Sakit"
: "Izin Lain"));
        System.out.println("Durasi Izin  : " + durasi + " hari");
    }
}
```

Class StackSurat27

```
public class StackSurat27 {
    Surat27[] stack;
    int top;
    int size;

    public StackSurat27(int size) {
        this.size = size;
        stack = new Surat27[size];
        top = -1;
    }

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

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

    public void push(Surat27 surat) {
        if (!isFull()) {
            stack[++top] = surat;
            System.out.println("Surat berhasil ditambahkan.");
        } else {
            System.out.println("Stack penuh. Tidak bisa menambahkan surat.");
        }
    }

    public Surat27 pop() {
        if (!isEmpty()) {
            return stack[top--];
        } else {
            System.out.println("Stack kosong. Tidak ada surat untuk diproses.");
            return null;
        }
    }

    public Surat27 peek() {
        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong. Tidak ada surat terakhir.");
            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("Surat ditemukan.");
                stack[i].tampilkan();
                ditemukan = true;
                break;
            }
        }
        if (!ditemukan) {
            System.out.println("Surat dengan nama " + nama + " tidak ditemukan.");
        }
    }
}
```

Class SuratDemo27

```
import java.util.Scanner;

public class SuratDemo27 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        StackSurat27 stack = new StackSurat27(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 berdasarkan Nama");
            System.out.println("0. Keluar");
            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 Mahasiswa: ");
                    String nama = scan.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = scan.nextLine();
                    System.out.print("Jenis Izin (S/I): ");
                    char jenis = scan.nextLine().toUpperCase().charAt(0);
                    System.out.print("Durasi Izin (hari): ");
                    int durasi = scan.nextInt();
                    scan.nextLine();
                    Surat27 surat = new Surat27(id, nama, kelas, jenis, durasi);
                    stack.push(surat);
                    break;

                case 2:
                    Surat27 diproses = stack.pop();
                    if (diproses != null) {
                        System.out.println("Memproses surat berikut:");
                        diproses.tampilkan();
                    }
                    break;

                case 3:
                    Surat27 terakhir = stack.peek();
                    if (terakhir != null) {
                        System.out.println("Surat terakhir yang diterima:");
                        terakhir.tampilkan();
                    }
                    break;

                case 4:
                    System.out.print("Masukkan nama mahasiswa: ");
                    String cariNama = scan.nextLine();
                    stack.cariSurat(cariNama);
                    break;

                case 0:
                    System.out.println("Terima kasih.");
                    break;

                default:
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilih != 0);
    }
}
```

Hasil Output

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan Nama
0. Keluar
Pilih: 1
ID Surat: 1001
Nama Mahasiswa: Dila
Kelas: 1A
Jenis Izin (S/I): S
Durasi Izin (hari): 2
Surat berhasil ditambahkan.
```

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan Nama
0. Keluar
Pilih: 1
ID Surat: 1002
Nama Mahasiswa: Erik
Kelas: 1B
Jenis Izin (S/I): I
Durasi Izin (hari): 3
Surat berhasil ditambahkan.
```


Menu:

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan Nama
0. Keluar

Pilih: 2

Memproses surat berikut:

ID Surat : 1002
Nama : Erik
Kelas : 1B
Jenis Izin : Izin Lain
Durasi Izin : 3 hari

Menu:

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan Nama
0. Keluar

Pilih: 3

Surat terakhir yang diterima:

ID Surat : 1001
Nama : Dila
Kelas : 1A
Jenis Izin : Sakit
Durasi Izin : 2 hari

Menu:

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan Nama
0. Keluar

Pilih: 4

Masukkan nama mahasiswa: Dila

Surat ditemukan:

ID Surat : 1001
Nama : Dila
Kelas : 1A
Jenis Izin : Sakit
Durasi Izin : 2 hari