

**LAPORAN HASIL PRAKTIKUM ALGORITMA DAN STRUKTUR
DATA**

Jobsheet 9



Adi Luhung

244107020088

Kelas 1E

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

2025

1. Praktikum

1. Percobaan 1

1. Membuat class Mahasiswa01.java dan menambahkan konstruktor untuk inialisasi atribut, kemudian menambahkan method tugasDinilai () untuk mengatur nilai ketika dilakukan penilaian tugas mahasiswa.

```
package Jobsheet9;

public class Mahasiswa01 {
    String nim, nama, kelas;
    int nilai;

    Mahasiswa01() {

    }
    Mahasiswa01(String nim, String nama, String kelas) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        nilai = -1;
    }
    void tugasDinilai(int nilai) {
        this.nilai = nilai;
    }
}
```

2. Membuat class StackTugasMahasiswa01.java sebagai tempat untuk mengelola tumpukan tugas. Class StackTugasMahasiswa01 merupakan penerapan dari struktur data Stack. Terdapat konstruktor dan beberapa method didalam class StackTugasMahasiswa, yaitu:

- a. Konstruktor untuk inialisasi atribut didalam class StackTugasMahasiswa01.

```
package Jobsheet9;

public class StackTugasMahasiswa01 {
    Mahasiswa01[] stack;
    int top, size;

    public StackTugasMahasiswa01(int size) {
        this.size = size;
        stack = new Mahasiswa01[size];
        top = -1;
    }
}
```

- b. Method isFull () untuk mengecek apakah tumpukan tugas mahasiswa sudah terisi penuh sesuai kapasitas

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

- c. Method `isEmpty()` untuk mengecek apakah tumpukan tugas masih kosong.

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

- d. Method `push()` untuk dapat menambahkan berkas tugas ke dalam tumpukan Stack.

```
public void push (Mahasiswa01 mhs) {
    if (!isFull()) {
        top++;
        stack[top] = mhs;
    } else {
        System.out.println("Stack penuh! Tidak bisa menambahkan tugas lagi.");
    }
}
```

- e. Method `pop()` untuk mengeluarkan tugas yang akan dinilai.

```
public Mahasiswa01 pop () {
    if (!isEmpty()) {
        Mahasiswa01 m = stack[top];
        top--;
        return m;
    } else {
        System.out.println("Stack kosong! Tidak ada tugas untuk dinilai.");
        return null;
    }
}
```

- f. Method `peek()` untuk dapat mengecek tumpukan tugas mahasiswa yang berada di posisi paling atas

```
public Mahasiswa01 peek () {
    if (!isEmpty()) {
        return stack[top];
    } else {
        System.out.println("Stack kosong! Tidak ada tugas yang dikumpulan.");
        return null;
    }
}
```

- g. 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("");
}
```

3. Membuat class MahasiswaDemo01.java untuk menjalankan program.

```
package Jobsheet9;
import java.util.Scanner;

public class MahasiswaDemo01 {
    public static void main(String[] args) {
        StackTugasMahasiswa01 stack = new StackTugasMahasiswa01(5);
        Scanner scan = new Scanner (System.in);
        int pilih = 0;

        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();
                    Mahasiswa01 mhs = new Mahasiswa01 (nim, nama, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n", mhs.nama);
                    break;

                case 2:
                    Mahasiswa01 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:
                    Mahasiswa01 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);
        scan.close();
    }
}
```

4. Hasil kode program.

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

5. Push kode program ke github

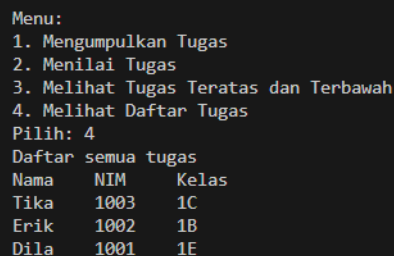
```
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git push origin main
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 8 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (24/24), 8.54 KiB | 1.42 MiB/s, done.
Total 24 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 5 local objects.
remote: This repository moved. Please use the new location:
remote:  https://github.com/Kumaaan/Praktikum_ASD.git
To https://github.com/Kumaaan/Praktikum-ASD.git
63c181e..85476ee  main -> main
```

1.1 Pertanyaan

1. Bagian yang perlu diperbaiki adalah pada bagian method `print()`, tepatnya seperti ini:

```
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("");  
}
```

Dengan kode program yang sudah di modifikasi diatas, maka saat daftar tugas ditampilkan akan dimulai dari data terakhir yang dimasukkan, sesuai seperti algoritma pada struktur data stack. Contoh:



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

2. Data yang dapat ditampung oleh Stack pada kode program tersebut adalah 5, hal ini diatur pada class `MahasiswaDemo`, dibawah ini:

```
public static void main(String[] args) {  
    StackTugasMahasiswa01 stack = new StackTugasMahasiswa01(5);  
    Scanner scan = new Scanner (System.in);  
    int pilih = 0;
```

3. Method `isFull()` dibutuhkan pada method `push()` untuk membatasi, dan memberi peringatan saat ingin menambahkan data baru dalam kondisi stack full. Jika kondisi `if-else` pada method `isFull()` dihapus maka akan terjadi error saat ingin menambahkan data dalam kondisi stack full.
4. Memodifikasi kode program agar dapat menampilkan daftar terbawah pada stack

a. Pada class `MahasiswaDemo1`

```
case 3:  
Mahasiswa01 lihat = stack.peek();  
Mahasiswa01 look = stack.bottom();  
if (lihat != null) {  
    System.out.println("Tugas terakhir dikumpulkan oleh " + lihat.nama);  
}  
if (look != null ) {  
    System.out.println("Tugas terakhir dikumpulkan oleh " + look.nama);  
}  
break;
```

b. Pada class StackTugasMahasiswa01

```
public Mahasiswa01 peek() {
    if (!isEmpty()) {
        return stack[top];
    } else {
        System.out.println("Stack kosong! Tidak ada tugas yang dikumpulan.");
        return null;
    }
}
public Mahasiswa01 bottom() {
    if (!isEmpty()) {
        return stack[0];
    } else {
        System.out.println("Stack kosong! Tidak ada tugas yang dikumpulan.");
        return null;
    }
}
```

Dengan modifikasi tersebut, saat pengguna memilih menu 3, program akan menampilkan data teratas, dan terbawah pada stack. Contoh:

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

5. Memodifikasi program agar dapat menampilkan total tugas yang sudah dikumpulkan.

a. Pada class MahasiswaDemo01

```
do {
    System.out.println("\nMenu:");
    System.out.println("1. Mengumpulkan Tugas");
    System.out.println("2. Menilai Tugas");
    System.out.println("3. Melihat Tugas Teratas dan Terbawah");
    System.out.println("4. Melihat Daftar Tugas");
    System.out.println("5. Melihat Jumlah Total Tugas");
    System.out.print("Pilih: ");
    pilih = scan.nextInt();
    scan.nextLine();
    switch (pilih) {
```

```
        case 4:
            System.out.println("Daftar semua tugas");
            System.out.println("Nama\tNIM\tKelas");
            stack.print();
            break;

        case 5:
            int total = stack.total();
            System.out.println("Jumlah total tugas yang sudah dikumpulkan: " + total);
            break;

        default:
            System.out.println("Pilihan tidak valid.");
    }
} while (pilih >= 1 && pilih <= 5);
```

Setelah program dimodifikasi, maka dapat menampilkan total dari tugas yang sudah dikumpulkan. Contoh:

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas dan Terbawah
4. Melihat Daftar Tugas
5. Melihat Jumlah Total Tugas
Pilih: 1
Nama: Dila
NIM: 1001
Kelas: 1E
Tugas Dila berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas dan Terbawah
4. Melihat Daftar Tugas
5. Melihat Jumlah Total Tugas
Pilih: 1
Nama: Erik
NIM: 1002
Kelas: 1B
Tugas Erik berhasil dikumpulkan
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas dan Terbawah
4. Melihat Daftar Tugas
5. Melihat Jumlah Total Tugas
Pilih: 1
Nama: Tika
NIM: 1003
Kelas: 1C
Tugas Tika berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas dan Terbawah
4. Melihat Daftar Tugas
5. Melihat Jumlah Total Tugas
Pilih: 5
Jumlah total tugas yang sudah dikumpulkan: 3
```


6. Push kode program ke github.

```
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git add .
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git commit -m "Jobsheet 9"
[main 7dcc15c] Jobsheet 9
 2 files changed, 31 insertions(+), 6 deletions(-)
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 738 bytes | 738.00 KiB/s, done.
Total 5 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
remote: This repository moved. Please use the new location:
remote:  https://github.com/Kumaaan/Praktikum_ASD.git
To https://github.com/Kumaaan/Praktikum-ASD.git
```

2. Percobaan 2

1. Menambahkan method konversiDesimalBiner pada class

StackTugasMahasiswa01 untuk merubah nilai mahasiswa menjadi bilangan biner.

```
public String konversiDesimalBiner(int nilai) {
    StackKonversi01 stack = new StackKonversi01();
    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;
}
```

2. Membuat class baru dengan nama `StackKonversi01` yang berfungsi untuk membedakan Stack mahasiswa dengan Stack yang digunakan untuk biner karena tipe data yang digunakan berbeda.

```
package Jobsheet9;

public class StackKonversi01 {
    int[] tumpukanBiner;
    int size, top;

    public StackKonversi01() {
        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("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;
        }
    }
}
```

3. Memodifikasi kode program pada class `MahasiswaDemo01` agar dapat menampilkan bilangan desimal yang sudah dikonversikan ke bilangan biner.

```
case 2:
    Mahasiswa01 dinilai = stack.pop();
    if (dinilai != null) {
        System.out.println("Menilai tugas dari " + dinilai.nama);
        System.out.print("Masukkan nilai (0-100): ");
        int nilai = scan.nextInt();
        dinilai.tugasDinilai(nilai);
        System.out.printf("Nilai Tugas %s adalah %d\n", dinilai.nama, nilai);
        String biner = stack.konversiDesimalBiner(nilai);
        System.out.println("Nilai Biner Tugas: " + biner);
    }
    break;
```

4. Hasil running kode program

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas dan Terbawah
4. Melihat Daftar Tugas
5. Melihat Jumlah Total Tugas
Pilih: 1
Nama: Tika
NIM: 1003
Kelas: 1C
Tugas Tika berhasil dikumpulkan

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

2.1 Pertanyaan

1. Alur kerja dari method konversiDesimalkeBiner.

```
public String konversiDesimalBiner(int nilai) {    //Menerima nilai yang diberikan
    StackKonversi01 stack = new StackKonversi01();    //Membuat objek untuk menyimpan sisa
    hasil bagi nilai / 2
    while (nilai > 0) {    //Mencari nilai sisa bagi yang akan di push ke stack dalam urutan
    terbalik
        int sisa = nilai % 2;
        stack.push(sisa);
        nilai = nilai / 2;
    }
    String biner = new String();    //Membuat tempat untuk menyimpan nilai yang dikirim dari
    stack
    while (!stack.isEmpty()) {    //Memindahkan nilai yang ada didalam stack ke biner dalam
    urutan yang benar
        biner += stack.pop();
    }
}
```

2. Jika modifikasi yang dimaksud adalah `while (nilai != 0)`, seperti dibawah ini:

```
public String konversiDesimalBiner(int nilai) {
    StackKonversi01 stack = new StackKonversi01();
    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;
}
```

Maka tidak akan terjadi error dan program akan berjalan normal. Hal tersebut karena nilai yang dimasukkan adalah nilai positif, jika yang dimasukkan adalah nilai negatif akan terjadi *infinite loop*. Tetapi dalam program ini hal tersebut bukanlah masalah karena yang akan dimasukkan adalah nilai tugas dari seorang mahasiswa, sehingga tidak akan bernilai negatif.

2. Latihan Praktikum

1. Membuat kode program untuk mengelola data surat izin para mahasiswa.\

a. Class Surat01

```
package Jobsheet9;

public class Surat01 {
    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin;
    int durasi;
    boolean terima;

    Surat01() {

    }
    Surat01(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;
    }
}
```

b. Class StackSurat01

```
package Jobsheet9;

public class StackSurat01 {
    Surat01[] stack;
    int top, size;

    public StackSurat01(int size) {
        this.size = size;
        stack = new Surat01[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 (Surat01 srt) {
        if (!isFull()) {
            top++;
            stack[top] = srt;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan data lagi.");
        }
    }
    public Surat01 pop () {
        if (!isEmpty()) {
            Surat01 m = stack[top];
            top--;
            return m;
        } else {
            System.out.println("Stack kosong! Tidak ada surat untuk diproses.");
            return null;
        }
    }
    public Surat01 peek() {
        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong! Tidak ada surat yang dikumpulan.");
            return null;
        }
    }
}
```

```

public void printCari(int hasil) {
    String keterangan = "";
    if (stack[hasil].jenisIzin == 'S') {
        keterangan = "Sakit";
    } else if (stack[hasil].jenisIzin == 'I') {
        keterangan = "Izin";
    } else {

    }
    System.out.printf("%-12s %-9s %-9s %-14s %-10d", stack[hasil].idSurat,
        stack[hasil].namaMahasiswa, stack[hasil].kelas, keterangan, stack[hasil].durasi);
}
public void print() {
    String keterangan = "";
    for (int i = top; i >= 0; i--) {
        if (stack[i].jenisIzin == 'S') {
            keterangan = "Sakit";
        } else if (stack[i].jenisIzin == 'I') {
            keterangan = "Izin";
        } else {
            System.out.println("Keterangan izin tidak valid!");
        }
        System.out.printf("%-12s %-9s %-9s %-14s %-10d\n", stack[i].idSurat,
            stack[i].namaMahasiswa, stack[i].kelas, keterangan, stack[i].durasi);
    }
}
}
}

```

c. Class SearchingSurat01

```

package Jobsheet9;

public class SearchingSurat01 {
    public int searchingSurat(StackSurat01 stack, String cariNama, int size) {
        int posisi = -1;
        for (int j = 0; j <= stack.top; j++) {
            if (stack.stack[j].namaMahasiswa.equalsIgnoreCase(cariNama)) {
                posisi = j;
                break;
            }
        }
        return posisi;
    }
}

```

d. Class SuratDemo1

```
package Jobsheet9;
import java.util.Scanner;
public class SuratDemo01 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Masukkan jumlah mahasiswa: ");
        int size = sc.nextInt();
        sc.nextLine();
        StackSurat01 stack = new StackSurat01(size);
        int pilih;
        String terima = "";

        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");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Id Surat: ");
                    String idSurat = sc.nextLine();
                    System.out.print("Nama: ");
                    String namaMahasiswa = sc.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = sc.nextLine();
                    System.out.print("Jenis Izin: ");
                    char jenisIzin = sc.next().charAt(0);
                    System.out.print("Durasi Izin: ");
                    int durasi = sc.nextInt();
                    sc.nextLine();
                    Surat01 srt = new Surat01 (idSurat, namaMahasiswa, kelas, jenisIzin, durasi);
                    stack.push(srt);
                    System.out.printf("Surat %s berhasil dikumpulkan\n", srt.namaMahasiswa);
                    break;

                case 2:
                    if (!stack.isEmpty()) {
                        Surat01 trm = new Surat01();
                        System.out.println("Daftar semua Surat:");
                        System.out.println("ID Surat      Nama      Kelas      Keterangan      Durasi");
                        stack.print();
                        System.out.println("");
                        System.out.printf("Apakah surat %s diterima? (y/n)", stack.peek().namaMahasiswa);
                        terima = sc.nextLine();
                        if (terima.equalsIgnoreCase("y")) {
                            trm = stack.pop();
                            System.out.printf("Surat dari %s diterima.\n", trm.namaMahasiswa);
                        } else if (terima.equalsIgnoreCase("n")) {
                            System.out.printf("Surat dari %s tidak diterima.",
                                stack.peek().namaMahasiswa);
                        } else {
                            System.out.println("Keterangan yang anda masukkan tidak vali!");
                        }
                    } else {
                        System.out.println("Tidak ada surat untuk diproses.");
                    }
                    break;
            }
        } while (true);
    }
}
```

```

        case 3:
            Surat01 lihat = stack.peek();
            if (lihat != null) {
                System.out.println("Surat terakhir dikumpulkan oleh " + lihat.namaMahasiswa);
                System.out.println("");
            }
            break;

        case 4:
            SearchingSurat01 pencari = new SearchingSurat01();
            System.out.print("Masukkan Nama Mahasiswa: ");
            String cariNama = sc.nextLine();
            int hasil = pencari.searchingSurat(stack, cariNama, size);
            if (hasil != -1) {
                System.out.println("Detail Surat: ");
                System.out.println("ID Surat      Nama      Kelas      Keterangan      Durasi
(Hari)");

                stack.printCari(hasil);
                System.out.println("");
            } else {
                System.out.println("Surat tidak ditemukan.\n");
            }
            break;

        default:
            System.out.println("Pilihan tidak valid.\n");
            break;
    }

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

```

2. Hasil kode program

a. Case 1

```

Masukkan jumlah mahasiswa: 3

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 1
Id Surat: 1001
Nama: Dila
Kelas: 1E
Jenis Izin: S
Durasi Izin: 2
Surat Dila berhasil dikumpulkan

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 1
Id Surat: 1002
Nama: Erik
Kelas: 1E
Jenis Izin: I
Durasi Izin: 1
Surat Erik berhasil dikumpulkan

```

```

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 1
Id Surat: 1003
Nama: Tika
Kelas: 1E
Jenis Izin: S
Durasi Izin: 1
Surat Tika berhasil dikumpulkan

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: █

```


b. Case 2

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 2
Daftar semua Surat:
ID Surat  Nama    Kelas  Keterangan  Durasi (Hari)
1003      Tika    1E     Sakit       1
1002      Erik    1E     Izin        1
1001      Dila    1E     Sakit       2

Apakah surat Tika diterima? (y/n)y
Surat dari Tika diterima.

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 2
Daftar semua Surat:
ID Surat  Nama    Kelas  Keterangan  Durasi (Hari)
1002      Erik    1E     Izin        1
1001      Dila    1E     Sakit       2

Apakah surat Erik diterima? (y/n)y
Surat dari Erik diterima.
```

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 2
Daftar semua Surat:
ID Surat  Nama    Kelas  Keterangan  Durasi (Hari)
1001      Dila    1E     Sakit       2

Apakah surat Dila diterima? (y/n)n
Surat dari Dila tidak diterima.

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 2
Daftar semua Surat:
ID Surat  Nama    Kelas  Keterangan  Durasi (Hari)
1001      Dila    1E     Sakit       2

Apakah surat Dila diterima? (y/n)
```

c. Case 3

```
Apakah surat Dila diterima? (y/n)n
Surat dari Dila tidak diterima.
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 3
Surat terakhir dikumpulkan oleh Dila

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 
```

d. Case 4

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 4
Masukkan Nama Mahasiswa: Dila
Detail Surat:
ID Surat    Nama    Kelas    Keterangan    Durasi (Hari)
1001        Dila    1E       Sakit         2

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: █
```

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 4
Masukkan Nama Mahasiswa: Erik
Detail Surat:
ID Surat    Nama    Kelas    Keterangan    Durasi (Hari)
1002        Erik    1E       Izin          1

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: █
```

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 5
Pilihan tidak valid.
```

3. Push ke github

```
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git add .
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git commit -m "jobsheet 9"
[main 99fbb95] jobsheet 9
7 files changed, 260 insertions(+), 1 deletion(-)
create mode 100644 Jobsheet9/SearchingSurat01.java
create mode 100644 Jobsheet9/StackKonversi01.java
create mode 100644 Jobsheet9/StackSurat01.java
create mode 100644 Jobsheet9/Surat01.java
create mode 100644 Jobsheet9/SuratDemo01.java
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git push origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 3.20 KiB | 1.60 MiB/s, done.
Total 10 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
remote: This repository moved. Please use the new location:
remote:   https://github.com/Kumaaan/Praktikum_ASD.git
To https://github.com/Kumaaan/Praktikum_ASD.git
7dcc15c..99fbb95  main -> main
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> █
```

