

LAPORAN HASIL PRAKTIKUM ALSD
JOBSHEET 9 STACK



Faiva Puspa Sahara

244107020036

TI – 1E

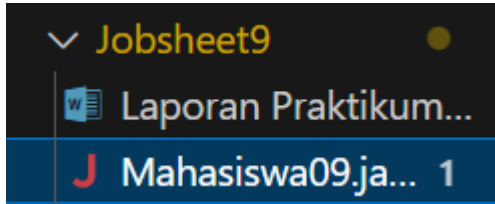
PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2025

HASIL PRAKTIKUM

2.1 Percobaan 1 (Mahasiswa Mengumpulkan Tugas)

A. Class Mahasiswa09.java

1. Buat folder baru bernama **Jobsheet9**. Buat file baru, beri nama **Mahasiswa09.java**



2. Lengkapi class **Mahasiswa09** dengan atribut yang telah digambarkan di dalam class diagram Mahasiswa, yang terdiri dari atribut **nama**, **nim**, **kelas**, dan **nilai**

```
public class Mahasiswa09 {  
    String nama;  
    String nim;  
    String kelas;  
    int 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

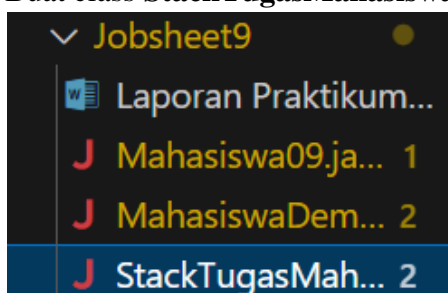
```
public Mahasiswa09() {  
    }  
    public Mahasiswa09(String nama, String nim, String kelas) {  
        this.nama = nama;  
        this.nim = nim;  
        this.kelas = kelas;  
        nilai = -1;  
    }  
}
```

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

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

B. Class StackTugasMahasiswa09.java

5. Buat class **StackTugasMahasiswa09.java** tempat untuk mengelola tumpukan tugas.



6. Lengkapi class **StackTugasMahasiswa09** dengan atribut yang telah digambarkan di dalam class diagram StackTugasMahasiswa, yang terdiri dari atribut **stack**, **size**, dan **top**

```
public class StackTugasMahasiswa09 {  
    Mahasiswa09[] stack;  
    int size;  
    int top;
```

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

```
public StackTugasMahasiswa09 (int size) {  
    this.size = size;  
    stack = new Mahasiswa09[size];  
    top = -1;  
}
```

8. 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;  
    }  
}
```

9. Buat method **isEmpty** bertipe boolean untuk mengecek apakah tumpukan tugas masih kosong

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

10. Buat method **push**. Method ini menerima parameter **mhs** yang berupa object dari class **Mahasiswa09**

```
public void push(Mahasiswa09 mhs) {  
    if (!isFull()) {  
        top++;  
        stack[top] = mhs;  
    } else {  
        System.out.println("Stack penuh! Tidak bisa  
menambahkan tugas lagi.");  
    }  
}
```

11. 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 **Mahasiswa09**

```

public Mahasiswa09 pop() {
    if (!isEmpty()) {
        Mahasiswa09 m = stack[top];
        top--;
        return m;
    } else {
        System.out.println("Stack kosong! Tidak ada tugas
untuk dinilai.");
        return null;
    }
}

```

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

```

public Mahasiswa09 peek() {
    if (!isEmpty()) {
        return stack[top];
    } else {
        System.out.println("Stack kosong! Tidak ada tugas yang
dikumpulkan.");
        return null;
    }
}

```

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

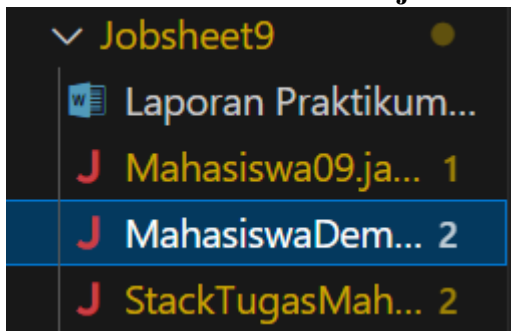
```

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("");
}

```

C. Class Utama

14. Buat file **MahasiswaDemo09.java**



15. Tuliskan struktur dasar Bahasa pemrograman java yang terdiri dari fungsi **main**

```

package Jobsheet9;

import java.util.Scanner;
public class MahasiswaDemo09 {
    public static void main(String[] args) {

```

16. Di dalam fungsi **main**, lakukan instansiasi object StackTugasMahasiswa bernama **stack** dengan nilai parameternya adalah 5. Deklarasikan Scanner dengan nama variabel **scan** dan variabel **pilih** bertipe int

```
public static void main(String[] args) {
    StackTugasMahasiswa09 stack = new
    StackTugasMahasiswa09(5);
    Scanner sc = new Scanner(System.in);
    int pilih;
```

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

```
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();
    switch (pilih) {
        case 1 :
            System.out.print("Nama: ");
            String nama = sc.next();
            System.out.print("NIM: ");
            String nim = sc.next();
            System.out.print("Kelas: ");
            String kelas = sc.next();
            Mahasiswa09 mhs = new Mahasiswa09(nama, nim,
kelas);
            stack.push(mhs);
            System.out.printf("Tugas %s berhasil
dikumpulkan\n", mhs.nama);
            break;
        case 2 :
            Mahasiswa09 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 :
            Mahasiswa09 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);
}
```

18. Run Program

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

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

19. Commit dan push ke Github

```
C:\Windows\System32\cmd.e  X  +  v

D:\KULIAH\SEM 2\Praktikum-ALSD>git add .


D:\KULIAH\SEM 2\Praktikum-ALSD>git commit -m "jobsheet9"
[main 4a5eb6a] jobsheet9
 8 files changed, 356 insertions(+)
 create mode 100644 CaseMethod/fiks/Akademik.java
 create mode 100644 CaseMethod/fiks/AkademikMain.java
 create mode 100644 CaseMethod/fiks/Mahasiswa.java
 create mode 100644 CaseMethod/fiks/MataKuliah.java
 create mode 100644 CaseMethod/fiks/NilaiMahasiswa.java
 create mode 100644 Jobsheet9/Mahasiswa09.java
 create mode 100644 Jobsheet9/MahasiswaDemo09.java
 create mode 100644 Jobsheet9/StackTugasMahasiswa09.java

D:\KULIAH\SEM 2\Praktikum-ALSD>git push origin main
info: please complete authentication in your browser...
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 4.04 KiB | 690.00 KiB/s, done.
Total 13 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Faiva-404/Praktikum-ALSD.git
   fb8ed63..4a5eb6a  main -> main





D:\KULIAH\SEM 2\Praktikum-ALSD>git push origin main
Everything up-to-date

D:\KULIAH\SEM 2\Praktikum-ALSD>
```

20. Compile dan run program

Praktikum-ALSD / Jobsheet9 / 



Faiva-404 jobsheet9 4a5eb6a · 1 hour ago  History		
Name	Last commit message	Last commit date
..		
 Mahasiswa09.java	jobsheet9	1 hour ago
 MahasiswaDemo09.java	jobsheet9	1 hour ago
 StackTugasMahasiswa09.java	jobsheet9	1 hour ago

➤ Pertanyaan

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

⇒ Kode Programnya

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

⇒ Outputnya

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

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

⇒ Banyaknya data tugas mahasiswa yang ditampung, maksimal ada 5.

```
StackTugasMahasiswa09 stack = new StackTugasMahasiswa09(5);
```

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

⇒ Kalau pengecekan kondisi **!isFull** dihapus, dampaknya bisa fatal. Tanpa batasan, program akan tetap mencoba menambahkan elemen meskipun kapasitas sudah penuh.

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!

⇒ Kode program – pada class **MahasiswaDemo09** tambahkan menu 5 dan tambahkan case 5 di switch

```
System.out.println(x:"5. Lihat Tugas Terbawah");
case 5 :
    Mahasiswa09 terbawah = stack.peek();
    if (terbawah != null) {
        System.out.println("Tugas terbawah dikumpulkan oleh " + terbawah.nama);
    }
    break;
default:
    System.out.println(x:"Pilihan tidak valid.");;
```

⇒ Kode program – pada class **StackTugasMahasiswa09** tambahkan method **void Terbawah**

```
public void Terbawah() {
    if (!isEmpty()) {
        System.out.println(stack[0].nama + "\t" + stack[0].nim + "\t" + stack[0].kelas);
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan.");
    }
}
```

⇒ Outputnya

```
Menu:
1. Mengumpukan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah Tugas
Pilih: 5
Tugas terbawah dikumpulkan oleh Tika
```

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

⇒ Kode program – pada class **MahasiswaDemo09** tambahkan menu 6 dan tambahkan case 6 di switch

```
System.out.println(x:"6. Jumlah Tugas");
case 6 :
    System.out.println("Jumlah tugas yang dikumpulkan: " + (stack.top + 1));
    break;
```

⇒ Kode program – pada class **StackTugasMahasiswa09** tambahkan method **JumlahTugas**

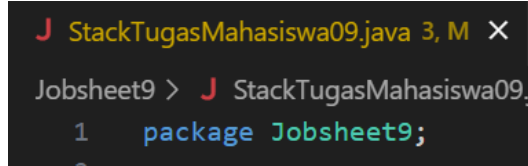
```
public int JumlahTugas() {
    return top + 1;
}
```

6. Commit dan push kode program ke Github

```
D:\KULIAH\SEM 2\Praktikum-ALSD\Jobsheet9>git add .
D:\KULIAH\SEM 2\Praktikum-ALSD\Jobsheet9>git commit -m "jobsheet9"
[main 458cbf8] jobsheet9
 4 files changed, 25 insertions(+), 2 deletions(-)
 create mode 100644 Jobsheet9/Laporan Praktikum.docx
 create mode 100644 Jobsheet9/~WRL2830.tmp
D:\KULIAH\SEM 2\Praktikum-ALSD\Jobsheet9>git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.55 MiB | 1.27 MiB/s, done.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To https://github.com/Faiva-404/Praktikum-ALSD.git
 4a5eb6a..458cbf8  main -> main
```


2.2 Percobaan 2 (Konversi Nilai Tugas ke Biner)

1. Buka kembali file **StackTugasMahasiswa09.java**



```
J StackTugasMahasiswa09.java 3, M X
Jobsheet9 > J StackTugasMahasiswa09_
1 package Jobsheet9;
```

2. Tambahkan method **konversiDesimalKeBiner** dengan menerima parameter **kode** bertipe int

```
public String konversiDesimalKeBiner(int nilai) {
    StackKonversi09 stack = new StackKonversi09();
    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;
}
```

3. Tambahkan empat method yaitu **isEmpty**, **isFull**, **push**, dan **pull** sebagai operasi utama Stack pada class **StackKonversi09**

```
package Jobsheet9;

public class StackKonversi09 {
    int[] tumpukanBiner;
    int size;
    int top;

    public StackKonversi09() {
        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()) {
            top++;
            tumpukanBiner[top] = data;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan data lagi.");
        }
    }
}
```

```

        public int pop() {
            if (!isEmpty()) {
                int data = tumpukanBiner[top];
                top--;
                return data;
            } else {
                System.out.println("Stack kosong! Tidak ada data untuk
diambil.");
                return -1;
            }
        }
    }
}

```

4. Agar nilai tugas mahasiswa dikonversi ke dalam bentuk biner setelah dilakukan penilaian, maka tambahkan baris kode program pada method **pop** di class **MahasiswaDemo09**

```

case 2 :
    Mahasiswa09 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);
        String biner =
stack.konversiDesimalKeBiner(nilai);
        System.out.println("Nilai Biner Tugas:" +
biner);
    }
    break;

```

5. Run Program

```

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

```

6. Commit dan Push ke Github

Praktikum ALSD / Lembar Kerja 9 /

Faiva-404 lembar kerja9 046548c Bahasa Indonesia: sekarang Sejarah

Nama	Pesan komit terakhir	Tanggal komitm...
..		
Laporan Praktikum.docx	lembar kerja9	45 menit yang lalu
Mahasiswa09.java	lembar kerja9	4 jam yang lalu
MahasiswaDemo09.java	lembar kerja9	sekarang
StackKonversi09.java	lembar kerja9	sekarang
StackTugasMahasiswa09.java	lembar kerja9	sekarang
~WRL2830.tmp	lembar kerja9	45 menit yang lalu

➤ Pertanyaan

1. Jelaskan alur kerja dari method **konversiDesimalKeBiner**!
⇒ Membagi angka secara berulang dengan 2 sambil mencatat sisa hasil bagi di setiap langkah. Sisa pembagian ini nantinya akan membentuk angka biner.
2. Pada method **konversiDesimalKeBiner**, ubah kondisi perulangan menjadi **while (kode != 0)**, bagaimana hasilnya? Jelaskan alasannya!
⇒ Kode Programnya

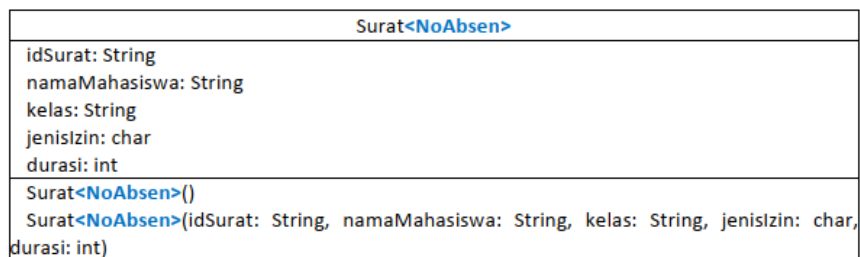
```
public String konversiDesimalKeBiner(int nilai) {  
    StackKonversi09 stack = new StackKonversi09();  
    while (kode != 0) {  
        int sisa = nilai % 2;  
        stack.push(sisa);  
        nilai = nilai / 2;  
    }  
    String biner = new String();  
    while (!stack.isEmpty()) {  
        biner += stack.pop();  
    }  
    return biner;  
}
```

⇒ Penjelasan :

- program akan mengalami error karena variabel **kode** tidak dideklarasikan atau tidak ada dalam metode.

2.3 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.



Atribut **jenisIzin** digunakan untuk menyimpan keterangan ijin 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**

➤ **Class Surat09.java**

```
package Jobsheet9;

public class Surat09 {
    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin;
    int durasi;

    public Surat09() {

    }

    public Surat09(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 StackSurat09.java**

```
package Jobsheet9;

public class StackSurat09 {
    Surat09[] stack;
    int size;
    int top;

    public StackSurat09(int size) {
        this.size = size;
        stack = new Surat09[size];
        top = -1;
    }

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

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

    public void push(Surat09 surat) {
        if (!isFull()) {
            top++;
            stack[top] = surat;
        } else {
            System.out.println("Stack penuh! Tidak bisa
menambahkan surat lagi.");
        }
    }

    public Surat09 pop() {
        if (!isEmpty()) {
            Surat09 surat = stack[top];
            top--;
            return surat;
        } else {
```

```

        System.out.println("Stack kosong! Tidak ada
surat untuk diambil.");
        return null;
    }
}

    public Surat09 peek() {
        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong! Tidak ada
surat yang dikumpulkan.");
            return null;
        }
    }

    public boolean search(String namaMahasiswa) {
        for (int i = top; i >= 0; i--) {
            if
(stack[i].namaMahasiswa.equals(namaMahasiswa)) {
                return true;
            }
        }
        return false;
    }
}

```

➤ Class SuratDemo09.java

```

package Jobsheet9;

import java.util.Scanner;
public class SuratDemo09 {
    public static void main(String[] args) {
        StackSurat09 stack = new StackSurat09(5);
        Scanner sc = new Scanner(System.in);
        int pilih;

        do {
            System.out.println("\nMenu: ");
            System.out.println("1. Terima Surat Izin");
            System.out.println("2. Proses Surat Izin");
            System.out.println("3. Lihat Surat Izin
Terakhir");
            System.out.println("4. Cari Surat Izin");
            System.out.print("Pilih : ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1 :
                    if (stack.isFull()) {
                        System.out.println("Stack penuh!
Tidak bisa menambahkan surat lagi.");
                    } else {
                        System.out.print("ID Surat
: ");
                        String idSurat = sc.nextLine();
                        System.out.print("Nama Mahasiswa
: ");
                        String namaMahasiswa =
sc.nextLine();
                        System.out.print("Kelas
: ");
                        String kelas = sc.nextLine();
                        System.out.print("Jenis Izin
(S/I): ");

```

```

char jenisIzin =
sc.next().charAt(0);
System.out.print("Durasi Izin
: ");

int durasi = sc.nextInt();
Surat09 surat = new
Surat09(idSurat, namaMahasiswa, kelas, jenisIzin,
durasi);

stack.push(surat);
System.out.println("Surat izin
dari " + surat.namaMahasiswa + " berhasil diterima.");
}
break;
case 2 :
Surat09 surat = stack.pop();
if (surat != null) {
System.out.println("Surat izin
dari " + surat.namaMahasiswa + " berhasil diproses.");
}
break;
case 3 :
Surat09 suratTerakhir = stack.peek();
if (suratTerakhir != null) {
System.out.println("Surat izin
terakhir: ");
System.out.println("ID Surat
: " + suratTerakhir.idSurat);
System.out.println("Nama
Mahasiswa: " + suratTerakhir.namaMahasiswa);
System.out.println("Jenis Izin
: " + suratTerakhir.jenisIzin + " (S/I)");
System.out.println("Durasi Izin
: " + suratTerakhir.durasi + " hari");
}
break;
case 4 :
System.out.print("Nama Mahasiswa yang
dicari: ");

String cariNama = sc.nextLine();
boolean ditemukan =
stack.search(cariNama);
if (ditemukan) {
System.out.println("Surat izin
untuk " + cariNama + " ditemukan.");
} else {
System.out.println("Surat izin
untuk " + cariNama + " tidak ditemukan.");
}
break;
default:
System.out.println("Pilihan tidak
valid. Silakan coba lagi.");
break;
}
} while (pilih >= 1 && pilih <= 4);
}
}

```

➤ Run Program

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih : 1
ID Surat      : A123
Nama Mahasiswa : Faiva
Kelas       : 1E
Jenis Izin (S/I): I
Durasi Izin   : 3
Surat izin dari Faiva berhasil diterima.

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih : 1
ID Surat      : B123
Nama Mahasiswa : Rafi
Kelas       : 1B
Jenis Izin (S/I): I
Durasi Izin   : 5
Surat izin dari Rafi berhasil diterima.

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih : 2
Surat izin dari Rafi berhasil diproses.
```

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih : 3
Surat izin terakhir:
ID Surat      : A123
Nama Mahasiswa: Faiva
Jenis Izin    : I (S/I)
Durasi Izin   : 3 hari

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih : 4
Nama Mahasiswa yang dicari: Faiva
Surat izin untuk Faiva ditemukan.

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih : 4
Nama Mahasiswa yang dicari: Ara
Surat izin untuk Ara tidak ditemukan.
```

➤ Commit dan Push ke Github

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22631.5189]
(c) Microsoft Corporation. All rights reserved.

D:\WULIAH\SEM 2\Praktikum-ALSD\Jobsheet9>git add .
D:\WULIAH\SEM 2\Praktikum-ALSD\Jobsheet9>git commit -m "Jobsheet9"
[main 48eb468] Jobsheet9
 5 files changed, 153 insertions(+)
 create mode 100644 Jobsheet9/StackSurat09.java
 create mode 100644 Jobsheet9/Surat09.java
 create mode 100644 Jobsheet9/SuratDemo09.java
 create mode 100644 Jobsheet9/~WRL3150.tmp
D:\WULIAH\SEM 2\Praktikum-ALSD\Jobsheet9>git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 222.40 KiB | 6.01 MiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Faiva-H04/Praktikum-ALSD.git
   bbcbb26..48eb468  main -> main
```

Praktikum ALSD / Lembar Kerja 9		
Faiva-404 Lembar Kerja 9 48eb468 Bahasa Indonesia: 1 jam yang lalu Sejarah		
Nama	Pesan komit terakhir	Tanggal komit...
..		
Laporan Praktikum.docx	Lembar Kerja 9	1 jam yang lalu
Mahasiswa09.java	lembar kerja9	15 jam yang lalu
MahasiswaDemo09.java	lembar kerja9	11 jam yang lalu
StackKonversi09.java	lembar kerja9	11 jam yang lalu
StackSurat09.java	Lembar Kerja 9	1 jam yang lalu
StackTugasMahasiswa09.java	lembar kerja9	11 jam yang lalu
Surat09.java	Lembar Kerja 9	1 jam yang lalu
SuratDemo09.java	Lembar Kerja 9	1 jam yang lalu
~WRL3150.tmp	Lembar Kerja 9	1 jam yang lalu