

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN STRUKTUR DATA  
JOBSHEET 6**



Disusun Oleh :

Nama : Nawaf Azril Annaufal

Nim : 244107020047

Kelas : TI 1E

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLINEMA  
2025**

Percobaan 1:

a. Hasil penulisan kode program class Mahasiswa21.java

```
package Praktikum_ASD.Jobsheet9;

public class Mahasiswa21 {

    String nim, nama, kelas;
    int nilai;

    Mahasiswa21(String nim, String nama, String kelas) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        nilai = -1;
    }

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

}
```

b. Hasil penulisan kode program class StackTugasMahasiswa21.java

```
package Praktikum_ASD.Jobsheet9;

public class StackTugasMahasiswa21 {

    Mahasiswa21[] stack;
    int size, top;

    StackTugasMahasiswa21(int size) {
        this.size = size;
        stack = new Mahasiswa21[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(Mahasiswa21 mhs) {
        if(!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println("Stack Penuh!, Tidak bisa menambahkan tugas lagi");
        }
    }
}
```

```

public Mahasiswa21 pop() {
    if(!isEmpty()) {
        Mahasiswa21 m = stack[top];
        top--;
        return m;
    } else {
        System.out.println("Stack Kosong! Tidak ada tugas untuk dinilai.");
        return null;
    }
}

public Mahasiswa21 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);
    }
}
}

```



```
case 3:
    Mahasiswa21 lihat = stack.peek();
    if (lihat != null) {
        System.out.printf("Tugas teratas adalah %s\n", 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, silakan coba lagi.");
    break;
}

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

sc.close();

}

}
```

d. Hasil run kode program

```
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 telah 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 telah dikumpulkan
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 3
Tugas teratas adalah 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 telah dikumpulkan
```

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

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

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

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

Pertanyaan:

1. Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan sama dengan verifikasi hasil percobaan! Bagian mana yang perlu diperbaiki?
2. Berapa banyak data tugas mahasiswa yang dapat ditampung di dalam Stack? Tunjukkan potongan kode programnya!
3. Mengapa perlu pengecekan kondisi !isFull() pada method push? Kalau kondisi if-else tersebut dihapus, apa dampaknya?
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!
5. Tambahkan method untuk dapat menghitung berapa banyak tugas yang sudah dikumpulkan saat ini, serta tambahkan operasi menunya!
6. Commit dan push kode program ke Github

Jawab:

1. Statement yang perlu diperbaiki berada di class StackTugasMahasiswa.java pada method print yang masih menggunakan metode FIFO bukan LIFO

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

2. Keterangan size berada dalam inputan konstruktor object StackTugasMahasiswa21.java pada main sistem

```
StackTugasMahasiswa21 stack = new StackTugasMahasiswa21(5);
```

3. Untuk menghindari error stackoverflow, jika kondisi dihapus maka akan beresiko menyebabkan eror dapat terjadi



#### 4. Hasil modifikasi kode program

##### Class StackTugasMahasiswa21.java

```
public Mahasiswa21 bottom() {
    if(!isEmpty()) {
        return stack[0];
    } else {
        System.out.println("Stack kosong!, Tidak ada tugas yang dikumpulkan");
        return null;
    }
}
```

##### Class MahasiswaDemo21.java

```
case 5:
    Mahasiswa21 terbawah = stack.bottom();
    if (terbawah != null) {
        System.out.printf("Tugas terbawah adalah %s\n", terbawah.nama);
    }
    break;
```

#### 5. Hasil modifikasi kode program

##### Class StackTugasMahasiswa21.java

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

##### Class MahasiswaDemo21.java

```
case 6:
    int jumlah = stack.jumlahTugas();
    System.out.printf("Jumlah tugas yang dikumpulkan: %d\n", jumlah);
    break;
```

## 6. Hasil push ke github

```
<ezreals_@ LAPTOP-ODCVBL7M in ~/C/P/Jobsheet9>
< (main)* >» git commit -m "Percobaan 1 & Pertanyaan"
[main ed47ec3] Percobaan 1 & Pertanyaan
2 files changed, 18 insertions(+), 1 deletion(-)
<ezreals_@ LAPTOP-ODCVBL7M in ~/C/P/Jobsheet9>
< (main)* >» 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), 604 bytes | 604.00 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.
To https://github.com/ezreals-ares/Praktikum_ASD.git
4d1f9d4..ed47ec3 main -> main
<ezreals_@ LAPTOP-ODCVBL7M in ~/C/P/Jobsheet9>
< (main)* >»
```

### Percobaan 2:

#### a. Modifikasi pada class StackTugasMahasiswa21.java

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

b. Pembuatan kode program class StackKonversi21.java

```
package Praktikum_ASD.Jobsheet9;

public class StackKonversi21 {

    int[] tumpukanBiner;
    int size;
    int top;

    public StackKonversi21() {
        this.size = 32;
        tumpukanBiner = new int[size];
        top = -1;
    }

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

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

    public void push(int data) {
        if (isFull()) {
            System.out.println("Stack penuh, tidak bisa menambahkan data");
        } else {
            top++;
            tumpukanBiner[top] = data;
        }
    }

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

c. Modifikasi pada class mahasiswaDemo21.java

case 2:

```
Mahasiswa21 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();
    sc.nextLine();
    dinilai.tugasDinilai(nilai);
    System.out.printf("Nilai Tugas %s adalah %d\n", dinilai.nama, dinilai.nilai);
    String biner = stack.konversiDesimalKeBiner(nilai);
    System.out.println("Nilai Biner Tugas: " + biner);
}
break;
```

2. Hasil push ke GitHub

```
<ezreals_ @ LAPTOP-ODCVBL7M in ~/C/P/Jobsheet9>
< (main)* >» git commit -m "Percobaan 2"
[main e46f9dc] Percobaan 2
3 files changed, 75 insertions(+), 1 deletion(-)
create mode 100644 Jobsheet9/StackKonversi21.java
<ezreals_ @ LAPTOP-ODCVBL7M in ~/C/P/Jobsheet9>
< (main)* >» git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.24 KiB | 1.24 MiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/ezreals-ares/Praktikum_ASD.git
    ed47ec3..e46f9dc  main -> main
<ezreals_ @ LAPTOP-ODCVBL7M in ~/C/P/Jobsheet9>
< (main)* >»
```

### 3. Hasil run kode program

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

### Pertanyaan

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

### Jawab

1.
  - a. Membuat stack kosong untuk menyimpan sisa pembagian desimal dengan 2
  - b. Mengulangi proses pembagian nilai desimal dengan 2
  - c. Membangun String Binner dari stack dengan mengambil dari stack satu per satu (pop), lalu gabungkan menjadi satu
  - d. Mengembalikan hasil
2. Jika merubah yang awalnya while (nilai > 0) menjadi while (nilai != 0) maka hasilnya tidak akan berubah dan program akan tetap berjalan dengan benar selama nilai yang diberikan adalah positif

## Latihan

### 1. Hasil penulisan kode program class Surat21.java

```
package Praktikum_ASD.Jobsheet9;

public class Surat21 {

    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin;
    int durasi;

    public Surat21(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;
    }
}
```

## 2. Hasil penulisan kode program class StackSurat21.java

```
package Praktikum_ASD.Jobsheet9;

public class StackSurat21 {
    Surat21[] stack;
    int size, top;

    public StackSurat21(int size) {
        this.size = size;
        stack = new Surat21[size];
        top = -1;
    }

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

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

    public void push(Surat21 surat) {
        if (!isFull()) {
            stack[++top] = surat;
        } else {
            System.out.println("Stack penuh! Tidak bisa menerima surat lagi.");
        }
    }

    public Surat21 pop() {
        if (!isEmpty()) {
            return stack[top--];
        } else {
            System.out.println("Stack kosong! Tidak ada surat untuk diproses.");
            return null;
        }
    }

    public Surat21 peek() {
        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong! Tidak ada surat terakhir.");
            return null;
        }
    }
}
```

```
public boolean cariSurat(String namaMahasiswa) {
    for (int i = 0; i <= top; i++) {
        if (stack[i].namaMahasiswa.equalsIgnoreCase(namaMahasiswa)) {
            return true;
        }
    }
    return false;
}

public void printAll() {
    for (int i = top; i >= 0; i--) {
        Surat21 s = stack[i];
        System.out.printf("%s - %s - %s - %c - %d hari\n",
            s.idSurat, s.namaMahasiswa, s.kelas, s.jenisIzin, s.durasi);
    }
}
}
```



### 3. Hasil penulisan kode program class SuratDemo21.java

```
package Praktikum_ASD.Jobsheet9;

import java.util.Scanner;

public class SuratDemo21 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StackSurat21 stackSurat = new StackSurat21(10);

        int pilihan;
        do {
            System.out.println("\nMenu Surat Izin:");
            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 Mahasiswa");
            System.out.println("5. Tampilkan Semua Surat");
            System.out.println("0. Keluar");
            System.out.print("Pilih: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print("ID Surat: ");
                    String id = sc.nextLine();
                    System.out.print("Nama Mahasiswa: ");
                    String nama = sc.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = sc.nextLine();
                    System.out.print("Jenis Izin (S/I): ");
                    char jenis = sc.nextLine().toUpperCase().charAt(0);
                    System.out.print("Durasi Izin (hari): ");
                    int durasi = sc.nextInt();
                    sc.nextLine();

                    Surat21 suratBaru = new Surat21(id, nama, kelas, jenis, durasi);
                    stackSurat.push(suratBaru);
                    System.out.println("Surat izin berhasil diterima.");
                    break;
            }
        } while (pilihan != 0);
    }
}
```

```

        case 2:
            Surat21 diproses = stackSurat.pop();
            if (diproses != null) {
                System.out.printf("Memproses surat dari %s (%s) - %c selama %d
hari\n",
                                diproses.namaMahasiswa, diproses.kelas, diproses.jenisIzin,
diproses.durasi);
            }
            break;

        case 3:
            Surat21 terakhir = stackSurat.peek();
            if (terakhir != null) {
                System.out.printf("Surat terakhir dari %s (%s) - %c selama %d
hari\n",
                                terakhir.namaMahasiswa, terakhir.kelas, terakhir.jenisIzin,
terakhir.durasi);
            }
            break;

        case 4:
            System.out.print("Masukkan nama mahasiswa yang dicari: ");
            String namaCari = sc.nextLine();
            boolean ditemukan = stackSurat.cariSurat(namaCari);
            if (ditemukan) {
                System.out.println("Surat ditemukan.");
            } else {
                System.out.println("Surat tidak ditemukan.");
            }
            break;

        case 5:
            System.out.println("Daftar Semua Surat Izin:");
            stackSurat.printAll();
            break;

        case 0:
            System.out.println("Terima kasih telah menggunakan sistem surat izin.");
            break;

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

    } while (pilihan != 0);

    sc.close();
}
}

```